T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

BİLGİ SİSTEMLERİNDE LOG YÖNETİMİ VE LOGLARIN DEĞERLENDİRİLMESİ

Yüksek Lisans Tezi

Ersun BAYRAKTAROĞLU

İSTANBUL, 2009

T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİ TEKNOLOJİLERİ

BİLGİ SİSTEMLERİNDE LOG YÖNETİMİ VE LOGLARIN DEĞERLENDİRİLMESİ

Yüksek Lisans Tezi

Ersun BAYRAKTAROĞLU

Tez Danışmanı: Yrd. Doç. Dr. Orhan GÖKÇOL
Tez Yardımcı Danışmanı: Yrd. Doç. Dr. Adem KARAHOCA

İSTANBUL, 2009

T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ
ENSTİTÜ ADI
PROGRAM ADI

Tezin Adı: Bilgi Sistemlerinde Log Yönetimi Ve Logların Değerlendirilmesi
Öğrencinin Adı Soyadı: Ersun Bayraktaroğlu
Tez Savunma Tarihi: 28/5/2009

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğu Enstitümüz tarafından onaylanmıştır.

Prof. Dr. Bülent Özgüler
Enstitü Müdürü

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğunu onaylarım.

Yrd. Doç. Dr. Orhan Gökçol
Program Koordinatörü

.
Bu Tez tarafımızca okunmuş, nitelik ve içerik açısından bir Yüksek Lisans tezi olarak yeterli görülmüş ve kabul edilmiştir.

Jüri Üyeleri		İmzalar
Unvanı, Adı ve SOYADI		
Tez Danışmanı	Yrd. Doç. Dr.Orhan Gökçol	----------
Ek Danışman	Yrd. Doç. Dr. Adem Karahoca	----------
Üye	Yrd. Doç. Dr. Yalçın Çekiç	----------
Üye	Yrd. Doç. Dr. M Alper Tunga	----------

ÖNSÖZ

Tez çalışmasında, kurumsal bilgi sistem mimarilerine örnek teşkil edebilecek, çok sayıda bilişim sistemi bulunan bir ortamda, Log Yönetimi üzerinde tez çalışmasının gerçekleştirme olanağı bulunabilmiştir. Alternatifbank A.Ş. kurumuna araştırma için verilen destek için teşekkür ederim.

Çalışma boyunca yönlendirmeleri ve yorumları için Yrd. Doç. Dr. Orhan Gökçol, Yrd. Doç. Dr. Adem Karahoca, yüksek lisans eğitimim sırasında verdikleri destekten dolayı aileme teşekkür ediyorum.

Tez çalışması süresince akademik çalışma ortamında her türlü olanak sunan Bahçeşehir Üniversitesine teşekkür eder, başarıların devamını dilerim.

ÖZET

BİLGİ SİSTEMLERİNDE LOG YÖNETİMİ VE
LOGLARIN DEĞERLENDİRİLMESİ

Bayraktaroğlu, Ersun

Bilgi Teknolojileri
Tez Danışmanı: Doç.Dr. Orhan Gökçol

(Haziran, 2009), 111

Bilgi Teknolojilerini oluşturan sistemler üzerinde yürütülmekte olan işlemler ve yine bu sistemler üzerinde çalışmakta olan uygulamaları kullanmak sureti ile gerçekleştirilen kullanıcı işlemlerine ait aktiviteler, başta yasal düzenlemeler olmak üzere, kurumsal politikalar ve hatta kişisel/yönetsel politikalar uyarınca, başta güvenlik amacı olmak üzere kaydedilmektedir. Bu kayıtlardan oluşan veri kümeleri (loglar, iz kayıtları) yasal soruşturmalarda, problemlerin çözümlenmesinde, kök sebep analizlerinde veya belki de bir olay meydana gelmeden önce uyarılar oluşturmak ve birçok çeşitli amaçla kullanılabilir. Farklı sistemler tarafından, farklı şekillerde depolanan bu kayıtların, gerektiğinde bir arada ele alınarak değerlendirilmesi, kayıt sayısının çokluğu açısından yönetilmesi, hangi aşamada filtre edilebileceği gibi sorunlar, log yönetimi ve log değerlendirilmesi konusunda, standartları belirlenmemiş ve tam anlamı ile çözülememiş meseleler olarak ortaya çıkmaktadır. Log Yönetimini gerçekleştirmek üzere, öncelikle sistemlerden anlamlı bilgilerin toplanabilmesi, toplanan bu logların güvenli bir şekilde depolanabilmesi, bilgi güvenliği veya yasal nedenlerle araştırılma gereken durumlarda sonuç elde edebilmek amacı ile karşılaştırma (korelasyon) yapılabilmesi gerekir. Bu konuda yapılan çalışmaların izinden giderek, açık kaynak kodlu araçlarla, anlamlı bir log yönetim sistemi oluşturabilmek üzere yapılabilecekler, tezimizin ana amacını oluşturmuştur.

Araştırmanın birinci bölümde, akademik kitaplıklardan tarama yapılarak log yönetimi çalışmalarındaki yaklaşımlar, karşılaşılan zorluklar tespit edilmiş, tez çalışmasının yapıldığı kurumda bilgi sistem mimarisi belirlenmiş, log yönetiminin bilgi güvenliği ile ilişkisi üzerinde durulmuştur. Log sisteminin bilgi güvenliği konusunda gelişen kontrollerin çok önemli bir bölümü haline gelmekte olduğu tespit edilmiştir.

İkinci bölümde, teorik ve pratik olarak çeşitli yazılım/araçlarının, bilgi sistemlerinde log ve ilgili verileri toplamak üzere, hangi amaçlarla kullanılabileceği araştırılmış, log yönetiminin zorlukları, bu işlemi yerine getirmek için dikkat edilmesi gereken hususlar, akademik araştırmalarda bu tez çalışmasının referans edildiği sonuçlardan yararlanılarak, ilgili aşamalarda bu çalışmalarla birlikte, tez çalışması kapsamında yapılan değerlendirmeler açıklanmıştır.

Üçüncü bölümde, açık kaynak kodlu Ossim Log Yönetim araçlarından log yönetimi ve karşılaştırmalar için ne şekilde yararlanabileceği, geliştirmeler, karşılaşılan zorluklar ve yapılan çalışmaların detayları anlatılmıştır.

Sonuç bölümünde Log Yönetimi ve Logların değerlendirilmesi konusunda, incelenen diğer araştırmalardan ve yapılan tez çalışmasından yararlanarak, log yönetim sistematiğinin adım adım tanımlanmasına çalışılmıştır.

Anahtar Kelimeler: Ossim Log Yönetim, Bilgi Güvenliği, Korelasyon ve Araçlar.

SUMMARY

LOG MANAGEMENT AND EVALUATION
IN INFORMATION SYSTEMS

Bayraktaroglu, Ersun
Information Technologies

Thesis Advisor: Doç. Dr. Orhan Gökçol
(June, 2009), 111 pages

Operations on systems constituting Information Technologies and activities on user actions while using the applications running on these systems are being logged mainly for the needs of legal arrangements and also for personal and managerial policies and especially for security reasons. Data sets (logs, trace records) consisting of such records can be used for legal investigations, for analyzing problems, for root cause analyses and also, perhaps for generating early warnings before an incident occurs, and for many other purposes. Problems like, dealing and evaluating together, managing the so many records, and deciding in which phase to filter out the records which are stored by various systems in various ways are issues that are not fully resolved and lacking standards. In order to realize Log Management, first we need to collect meaningful data from systems, then to store these collected logs securely, and finally to do comparisons (correlation) in cases the we need to get results for information security or legal reasons. The main objective of this thesis is to pursue the works done on this subject and on what can be done in order to form a reasonable log management system using open source code tools.

In the first part of this research, the log management approaches and the encountered difficulties were determined by doing a thorough research on academic libraries; the information system architecture of the institution were this thesis was conducted was determined; and the relation between log management and information security was investigated. It was established that the log system is becoming a very important part of the evolving controls on information security.

In the second part, the current solutions on log and security management were investigated; difficulties that may be encountered while creating a log management system were studied; critical points that need attention were tried to be determined; and other academic researches were evaluated.

The third part explains how the open source code Ossim Log Management tool can be utilized for log management and comparisons, the difficulties encountered and the details of the work done.

In the conclusion part, the log management systematic was attempted to be defined step by step by utilizing other researches and thesis studies done on Log Management and Log evaluation topics.

Key Words: Ossim Log Management, Information Security, Correlation and Tools.

İÇİNDEKİLER

TABLOLAR

ŞEKİLLER

YAZILIM KODLARI

KISALTMALAR

Merkezi İşlem Birimi (Central Processing Unit)	:	CPU
Grafik Arayüz (Graphical User Interface)	:	GUI
İşletim Sistemi (Operating System)	:	OS
Hypertext Preprocessor	:	PHP
Güvenlik Olay Yönetimi (Security Incident Event Manager)	:	SIEM
Eklenti	:	PLUG-IN
Sanal Özel Ağ (Virtual Private Network)	:	VPN
Adres Çöüzmleme Protokolu (Address Resolution Protocol)	:	ARP
Uzaktan gözetim sistemi	:	RMON
Grafik Arayüz (Graphic Unit Interface)	:	GUI
İletişim Protokelleri Listesi (Request For Comments)	:	RFC
Bankacılık Düzenleme ve Denetleme Kurulu	:	BDDK
Düzenli İfade (Regular Expression)	:	REGEX

SEMBOLLER

Log Bilgisi Elde Edilecek Bilgi Sistemleri	:	R
Bilgi Sistemlerinde Yer Alan Cihazlar	:	D_n
Her Cihazda Üretilen Log Çeşitleri	:	B_{im}
Küme Elemanı	:	$\in$
Loglardaki Olay Çeşitleri	:	e_{ip}
Girişim Müdahale Düzeyi	:	C
Saldırı Düzeyi	:	A

İNGİLİZCE TERİMLER

Central Processing Unit (Merkezi İşlem Birimi)	:	CPU
Graphical User Interface	:	GUI
HyperText Markup Language	:	HTML
HyperText Transfer Protocol	:	HTTP
Intrusion Prevention System	:	IDP
Intrusion Detection System	:	IDS
Internet Engineering Task Force	:	IETF
Internet Information Services	:	IIS
Internet Protocol	:	IP
Local Area Network	:	LAN
Network mapper	:	NMAP
Media access control	:	MAC
Statistical Packet Anomaly Detection Engine (Snort, IDS)	:	SPADE
The National Institute of Standards and Technology	:	NIST

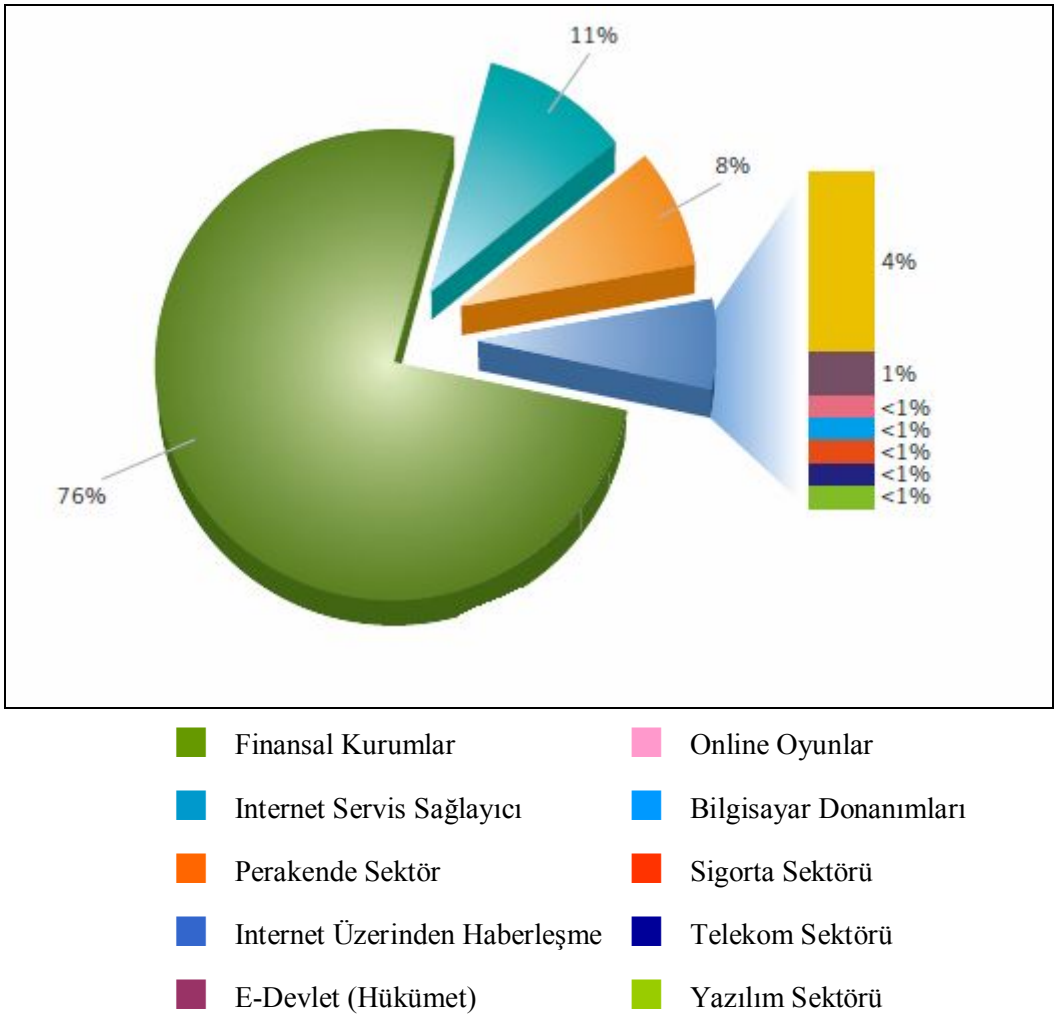
1. GİRİŞ

Bilgi sistemlerinin çok yönlü olarak kullanımı ve sistemlerin hem lokal hem de geniş ağlar içerisinde diğer sistemlerle etkileşimi, sistem yönetiminde bir çok farklı alanın kontrol edilmesi ve bu kontrollerin ayrı ayrı geliştirilmesi gereğini ortaya çıkartmaktadır.

1.1 LOG SİSTEMLERİNİN ÖNEMİ VE GELİŞTİRİLME NEDENLERİ

Son yıllarda ve oldukça hızlı gelişen bir süreçte, bilgi sistemleri yönetimi (IT Governance) akademik ve bilimsel bir araştırma konusu haline gelmiş, değişik metodolojiler ve standartların kullanımına yönelik yaklaşımlar ortaya çıkmıştır. Sistem yaşam döngüsü ve bu standartlar çerçevesinde kullanılmakta olan sistemlerin, izlenmesi gereği kararlı bir zorunluluk haline gelmiştir. Sistemleri pasif ve aktif yöntemlerle izlemek mümkündür. Örneğin ateş duvarı (firewall) kuralları ve güvenlik politikaları pasif izlemeye bir örnektir, Conti (2006, s.167). Bu kuralların birleşiminden türetilen yeni kurallar veya izlenen aktivitelerin yorumlanması sayesinde, herhangi bir girişimden önce uyarı mekanizmaları kurulabilmesi, ya da firewall tarafından izinsiz paketlerin geçirilmemesi, aktif izlemeye bir örnek olarak verilebilir. Sistem bütünü ve olaylar hakkında bilgi sahibi olmaya çalışarak, başka aktif sonuçlar elde etmek mümkündür, Pouget, Dacier (2003, s.7). Tez çalışması içerisinde de açıklanan ağ güvenliği konuları, sıklıkla pasif önlemler üzerinde kurulmuş kontrolleri işaret eder. İzleme işleminin temel amacı, güvenlik ifadesini açıklayan, bütünlük, ulaşabilirlik ve gizlilik üçlüsünü sağlam bir şekilde ayakta tutmaya yöneliktir. Her önleme karşın, bütünlüğün bozulmasına sebep olabilecek, kasıtlı yada kasıtlı olmayan girişimler, sistemsel hata veya yanlış kodlamalar nedeniyle, bir çok beklenmedik olay meydana gelebileceği açıktır.

Diğer yandan, teknolojik gelişimlere paralel olarak, çeşitli amaçlara yönelik biçimde artan bir şekilde sistemler üzerinde etkili olabilecek, bir çok yeni atak tipleri oluşmaya devam etmektedir. Bu nedenle hangi güvenlik olayının ne tür bir tehdit oluşturabileceği tekrar edilen risk değerlendirme çalışmaları ile belirlenmeli, bu riski minimize etmek için çeşitli kontroller oluştururken, değerlendirme aşamasında gerekli olacak olan hangi iz kayıtlarının gerekebileceği konusunda önceden çalışılmalıdır. Şekil 1,1'de yaygın internet tehditlerinin (phishing saldırıları gibi) hedeflediği sektörler gösterilmektedir.



Şekil 1.1 : Internet üzerinden yapılan sahtekârlıkların hedef aldığı sektörler
Kaynak : Symantech Security Threat Report, Nisan 2009.

Olasılıkları belirlerken, bilgi sistemleri güvenliğine yönelik bir çok araştırmada, sistemi kullanmakla yetkilendirilmiş kullanıcıların, dışarıdan sisteme müdahale etmek isteyebilecek girişimcilerin güvenlik ihmallerinden daha yüksek bir oranda tehdit oluşturabildiği de dikkate alınmalıdır, Gorge M. (2007).

Ülkemizde bilgi güvenliği farkındalığının artmasına paralel bir şekilde 5651 sayılı yasanın gerekliliği olarak, kurumlar ve servis sağlayıcılar tarafından, internete ulaşım kayıtlarının bir yıl boyunca saklanma zorunluluğu getirilmiştir. Bu nedenle, özellikle kurumsal yapılarda kullanılan otomatik Internet Protokol adresi dağıtan DHCP sunucu iz kayıtlarının, belli süreler ile tutulması gereksinimi ortaya çıkmıştır. Hatta iç ağınızda

barındırdığınız Web, E-Posta veya FTP sunucunuza ait trafik bilgisinin kayıtları da kanunlarda belirtilen gerekleri yerine getirebilme üzere, doğruluğunu ve bütünlüğünü bozmadan muhafaza etmeli ve gizliliği temin edilmelidir.

Teknolojinin kullanımı yaygınlaşmaya devam ettiği müddetçe, özellikle bilgi sistemleri alanında yapılan denetimlerin de yaygınlaşmakta olduğu görülmektedir. Denetim yapan sorumlular için, yapılan işlemlerin kontrolünde aranan en önemli kanıtları yine log verileri oluşturmaktadır.

Bu nedenle tez çalışması dahilinde, hangi sistemler üzerinde iz kayıtlarının tutulması gerektiği, iz kayıtlarının tutulacağı sistemler hakkında yukarıdan aşağıya doğru bir inceleme yapılmıştır. Bir kurum bünyesinde işi, işlevi hakkında deneyim sahibi olmuş uzmanlar tarafından en doğru bilgilerin alınabileceği bilinmelidir. Bu çalışmalarda, hem içeriden hem de kurum/altyapı dışından kaynaklanabilecek erişim veya teşebbüsleri dikkate almak gerekmektedir. Özetlemek istersek, log yönetiminin gerçekleştirilebilmesi için, hem servis verilen iş konusu, hem de sistemler ve teknolojileri hakkında bilgi sahibi olmak, iz kayıtları konusunda sistem yeteneklerini gerektiğinde geliştirebilmek gerekmektedir, bkz. Kod 8.1, Kod 8.2.

J. Herrerias, R. Gomez, (2007), bilişime yönelik saldırılar yada sistemde kesintiye neden olabilecek kritiklikteki hata oluşturan durumların araştırabilmesinin, bilgisayar sistemi ve iletişim ağları güvenli bir şekilde işletilirken, bu tip bir girişimin (intrusion) tespit edildiği anda, bu girişime ait aksiyonları gösterebilen kanıtların elde edilmesine bağlı olduğunu belirtmiştir. Sistemlerde meydana gelebilecek bu tip aksiyonların tespiti için kullanılabilecek gerçek ana kaynak ve bilgilerin yorumlanabileceği en etkin veri kümesi, Log dosyalarıdır.

1.2 LOGLAR VE BİLGİ GÜVENLİĞİ İLE İLİŞKİSİ

Bilgi sistemleri içerisinde çalışmakta olan tüm ağ ve güvenlik bileşenleri her gün çok sayıda log üretir. Ancak bunlara bir de sunucular ve istemcilerin logları da eklendiğinde hareketlere ve trafiğe ilişkin değerli olabilecek bir çok bilginin süzülmesi, başka hareket

ve trafik bilgileri ile ilişkilendirilmesi, analiz edilmesi, takibi ve anlamlı sonuçlar verebilecek şekilde raporlanması neredeyse olanaksız bir hale gelmektedir.

Bu tür bilgilerin yeterince etkili şekilde değerlendirilemediği durumlarda, kontrol gerçek anlamda sağlanamamakta ve kontrol edilemeyen bilgi sistemleri alt yapılarına yönelik yatırımlar da, verimli kullanılamamış olacaktır.

Log yönetiminde kullanılacak araçların çeşitli yönleri ile incelenmesi de bu araştırmanın içerisinde yer alırken, ortamın ihtiyaçlarına göre bunun hangi sistemler için gerçekleştirileceği ve hangi amaçlara hizmet edebileceği, ayrıca bunun sonuçları üzerinde nasıl bir değerlendirme yapılabileceği de araştırma kapsamına girmiş, bu konuda işleyen bir organizmanın çeşitli birimleri ile görüşülerek bilgi toplanmaya çalışılmıştır. Yasal düzenlemelere göre, bilgi sistemlerinde denetim izlerinin bir çoğu, sistemler için tutulan log kayıtlarını işaret eder. Log yönetim çalışmalarında ele alınması gereken ilk konunun, bilgi sistemlerinin hangi süreçlerinde, hangi log verilerinin log yönetimi amacı ile değerlendirilmesi gerektiğine karar verilmesidir.

Log yönetimi, bilgi güvenliği yönetiminin önemli bir bölümü veya bileşeni, bilgi güvenliği yönetimi ise ağın yönetimi ile oldukça yakın ilişkideki bir yönetim sistemidir. (Network güvenliği ve yönetimi, bilgi güvenliğinin sağlanması için gereken sistemlerden yalnızca biridir.) Ağ güvenliği yönetimi için, dikkat edilmesi gereken önemli bir mesele, network üzerindeki atakların saptanabilmesi ve bunları doğru olarak tanımlayabilmektir. Aslında bu durum çok kullanıcılı ve çok çeşitli sistem-ağ yapısına sahip ortamlarda, samanlıkta iğne aramaya benzetilebilir.

Bilgi güvenliğini sağlamak üzere ihtiyaç duyulabilecek en önemli veriler güvenlik raporlarıdır. Güvenlik Raporları birer birer sistemlerin kendisinden alınabilecek raporlardan çok, farklı veritabanlarında bulunan veriler kullanılarak; otomatik olarak oluşturuldukları takdirde, güvenlik durumuna ait daha kapsamlı genel bir görüntü sunabilecek ve farklı bakış açılarına ait verileri bir araya getirecektir. Yine bu sonucu elde etmek üzere toplanan olay bilgilerinin, log kayıtları şeklinde depolanarak kullanılması, log yönetimindeki amaçlardan biridir, Forte, Power (2008).

Tablo 1.1: Bilgi Güvenliğinde Log Verilerinden Elde Edilecek Çıktılar

Güvenlik Raporları Elde Edilebilecek Sonuçlar:
· Kaynak, hedef ve türlerine göre Güvenlik Olayları İstatistikleri
· Anormallik raporları
· Güvenlik raporları
· Erişilebilirlik raporları
· Kullanılabilirlik ve profil raporları
· Sunucu IDS raporları
· Güvenlik Açıkları raporları
· Özel metrikler
· Risk metrikleri
· "Forensic" konsol

Log yönetimi çalışmalarında, yalnızca log depolamak bir amaç olmamalı, oldukça yoğun bir ön araştırma ve sistemler konusunda bilgi gerektiren bu tip çalışmalarda, yukarıda belirtilen amaçlara ulaşılması da sağlanmalıdır, Gomez, Herrerias (2007).

Özellikle network güvenliğinde, pasif ve aktif gözleme amacıyla geliştirilen girişim tespit sistemleri (Intrusion detection systems) network üzerindeki belli olayları izleyip çeşitli vakaları saptayabilir. Girişim tespit sistemleri, bir seri network olayının (event) oluşturabileceği ve önceden bir imza veritabanında buna benzer olayların yer aldığı veriler (signature) yardımı ile bu atakların benzeştirilerek tespit edilebilmesi esasına dayanmaktadır. Ancak bir çok durumda, bu sistemlerde çok sayıda alarm oluşabilmekte ve atakların bir çoğunun aslında bir network atağı olmadığının sonradan anlaşılabileceğine dair çok sayıda örnek oluşması da kaçınılmazdır, Pouget, Dacier (2003, s.36). Bu durum yanlışlıkla alarm edilen aslında normal olan prosesler yada kısaca " false positive " olarak adlandırılır. Bunu şu şekilde örnekleyebiliriz: "Windows işletim sistemi üzerinde çalışan IIS web servis hizmetini etkileyebilecek bir atak için, Linux işletim sisteminde çalışan Apache servisini de etkileyeceği düşünülerek alarm üretilmesi" gibi bir örnek verilebilir. Ayrıca girişim tespit sistemleri (IDS, IPS) sistemleri her yeni atağın signature bilgisini içermeyebilirler. Üstelik çoğu kez ataklar son derece kompleks davranışlar sunmaktadır. Bu konu üzerinde geliştirilen Ossim gibi açık kaynak kodlu ürünler, atak ve zayıflıkların tespiti konusunda oldukça etkili servisler sunmaktadır, Gomez, Herrerias (2007, s.5). Tez araştırmasında ulaşılmak istenen hedeflerden biri de, genel olarak sistemler üzerinde, bilgi sistemlerinin ortamdaki yapısı değerlendirilerek elde edilen log kayıtlarının, girişim tespit

sistemlerini de içerecek bir biçimde, açık kaynak kodlu ürünlerle ihtiyaçları karşılaştıran bir çözümleme yapılabileceğini belirlemektir.

1.3 LOG YÖNETİMİ AMAÇLI İNCELENEN YAZILIM ARAÇLARI

Yukarıda belirtilen açıklamalara dayanarak, araştırmanın ilk aşamalarında, çeşitli log yönetim araçlarının avantajlı ve dezavantajlı yönleri, bu tez çalışmasındaki amaç dikkate alınacak şekilde incelenmiştir. İncelenen örneklerde, Log toplama, Log yönetimi ve Log karşılaştırma amaçlı araçlardan bir kısmının ticari yollarla elde edilebildiği, bir kısmının da açık kaynak kodlu araçlar olup kullanılabileceğini de belirtilmiştir. Ticari araçlar da dahil olmak üzere incelenen araçların ve kullandıkları yöntemlerin, (Tablo 1.2) log yönetimi ve log korelasyonunu konusunda, tam olarak çözüm sağlayıp sağlayamayacağı tartışmaya açık gözükmektedir. Bu durum, log yönetimi ve olay verilerinin değerlendirilmesinin zor yönetilebilir bir süreç olduğunu tekrar işaret etmiştir. Log yönetimi konusunda yapılan diğer bir araştırmada, "Gartner Group"'dan Kavangh ve Nicolette (2008) isimli araştırmacılar logların güvenliği konusunda gelecekte yapılması gerekli çalışmalara ve bu yönde kullanılabilecek teknolojilere değinmiştir. Çalışmada incelediğimiz araçlar hakkında aşağıdaki bilgilere ulaşılmıştır.

Kiwi SysLog, Knoppix-NSM, Knoppix-STD, Back Track and NST, log korelayonu olmayan araçlardır. Ancak OSSIM ve Prelude daha uygun bir yapılanma sunmaktadır. Her iki ürün hem korelasyon motoru içermekte, hem de bu konuda kendisini ispatlamış araçlar ile bütünleşmiştir. (https://wiki.edubuntu.org/UMCSecurity).

Prelude aracıda çeşitli alıcı ajanlar (sensor) kullanabilen ve IDMEF standardında (Intrusion Detection Message Exchange Format) event üretebilen, özellikle alarm üretebilme yeteneği gelişmiş bir araçtır.

Açık kaynak kodlu, log yönetim ve güvenlik aracı OSSIM, bilinen kabul görmüş bir çok aracın kolay entegre edildiği, güçlü bir korelasyon motoru içeren ve halen geliştirilmekte olan açık kaynak kodlu halen aktif bir projedir. Bu duruma, ilgili yazılım geliştiricilerinin OSSIM üzerinde hızlı eklentiler sağlayacak olması ve çeşitli sorunlar hakkında hızlı yardım bulunabilmesi anlamına gelmektedir. Ayrıca plug-in (eklenti)

yazmak için tez uygulama çalışmaları içerisinde farklı denemeler gerçekleştirilmiş ve başarılı sonuçlar elde edilmiştir. OSSIM bünyesine "Prelude"'e ait olabilecek avantajların eklenebileceği gözükmektedir.

Tablo 1.2: Log Yönetim Sistemleri Karşılaştırma Tablosu

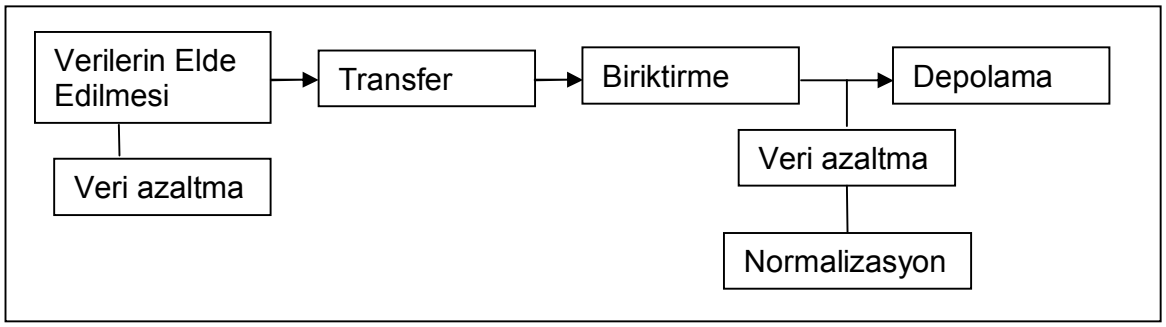
SIEM Güvenlik Araçları	**Avantajlar**	**Dezavantajlar**
Knoppix-NSM	• Detaylı Network Traffic Raporu • Güçlü İzleme "Monitoring"	• İzleme ("Monitoring") • Log Korelasyon Yok
Arc-Sight	• Merkezi Veritabanı (Gerçek Zamanlı) • Analiz Araçları	• Üretici Know-How Bağımlılığı • Ticari
Tivoli Inside Manager	• Merkezi Veritabanı (Gerçek Zamanlı) Olarak Toplayan; Bunları Birbiriyle İlişkili Süreçlere Oturtarak İzleyen, Analiz Eden Ve Raporlayan Gelişkin Bir Araçtır.	• Korelasyon Zayıf • Entegrasyon Gereği, • Ticari
Prelude	• IDMEF standard for information collection • Correlation	• Plug-in yapmak son derece güç
Backtrack	• Çok sayıda araç içerir • Bootable CD	• Korelasyon Yok
NST	• 100'ün üzerinde güvenlik aracı • Kurulum CD'si	• Korelasyon Yok
KIWI	• Performans • Kurulum CD'si	• Korelasyon Yok
OSSIM	• Güçlü Korelasyon • Yazılabilir Plug-in • Kurulum CD'si • Aktif Olarak Geliştirilmekte	• Yeni Korelasyon İhtiyaçları

2. LOG KAYITLARI, ANALİZ VE YÖNETİM

J. Herrerias, R. Gomez, (2007), "A log Correlation Model to Support the Evidence Search Process in a Forensic Investigation" isimli makalelerinde, tez çalışmasının ana fikriyle örtüşen biçimde, farklı sistemlerden log toplamaya yönelik ajanlardan oluşan bir set yapısı ile birlikte, filtreleme, normalizasyon ve karşılaştırma aşamalarını anlatmıştır.

2.1 LOG KAYITLARININ TOPLANMASI

Log kayıtları birden çok sürecin işletilebilmesi koşulu ile toplanmaktadır.



Şekil 2.1 : Log Verilerinin Toplanması Yaşam Döngüsü

Log toplama süreci aşağıda gösterilen bir formül ile ifade edilmiştir. Denklemde sistemler (domain) R ile simgelenmiş, bu sistem içindeki tüm cihazlar D ile gösterilmiştir. Denklem 2.1 log kayıtları kümesi, 2.2 bilgi sistemleri cihazlarının farklı log tiplerini, 2.3 ise her sistem cihazının farklı olay logları kümesine sahip olduğunu göstermektedir.

$$R = \{D_1\ D_{2,\ldots,}\ D_n\} \qquad \textbf{Denklem 2.1}$$

Her sistem cihazının kendi loglarının olması, B log türleri olmak üzere,

$$D_i = \{B_{i1,}\ B_{i2,\ldots,}\ B_{im}\},\ i \in [1,n] \qquad \textbf{Denklem 2.2}$$

Her indeksin bir cihazı temsil etmesi durumunda, her cihazın farklı tip olay kayıtları oluşturması durumu (Örneğin windows sistemlerinde, uygulama (application), sistem (system), güvenlik (security) loglarının ayrı ayrı olması gibi...), *e* olay tipi olmak üzere; log modellemesini formüle edebiliriz.

$$B_{ij} = \{e_{ij1}, e_{ij2,\ldots..}, e_{ip}\},\ i \in [1,n],\ j \in [1,m]$$ **Denklem 2.3**

Log kayıt kümesi, işlemlerin yürütülmekte olduğu sistemde, sistem üzerinde aksiyonlara ait kayıtların tutulması ile oluşturuldukları gibi, bir başka sisteme aktarılmasını sağlayan diğer bir aksiyon ile de (eş zamanlı olan ve olmayan) oluşturulabilir. Tüm log verilerinin aynı ortama aktarılması işlemine Log Depolaması adı verilmiştir. Ancak, sonuçlara bakıldığında, çok sayıda ve yığınlar şeklinde kaydedilen tüm bilişim olayları (events), suç amaçlı veya hatalarla ortaya çıkabilecek girişimlerin araştırılmasını, son derece karışık bir duruma getirmiştir. J. Herrerias, Gomez, (2007) araştırmalarında , log dosyalarının bölümlenmesi (amaç ve yapısına göre) ile elde edilen olay karşılaştırmaları (event correlation) için ve logları bir ajanlar kümesinden faydalanarak toplamak, filtre etmek ve normalize etmek amaçlı genel bir model oluşturmuştur. Modele ait log toplanması ile ilgili formül başlıca gelişimi ile yukarıda verilmiştir. Bu modelin filtrasyon (kayıt sayısını indirgemek) ve normalizasyon aşamaları, tez çalışmasında logların bir noktada biriktirilmesi sonrasında değil, aksine daha öncesinde, hangi sistemden hangi amaçla log toplanacağının belirlenmesi süresinde ele alınmıştır. Her durumda bu çalışmalar, girişimlerle ilgili kanıtların toplamasına yardımcı olmak üzere hedeflenmiştir. Girişim tespitinde istatistiksel veri ile eşleşmesinde bir zaafiyet oluşturmamak açısından, girişim tespit sistemlerin elde ettiği network ile ilgili loglara tez çalışması boyunca filtre uygulaması yapılmamıştır.

Log depolama işlemi öncesinde bir sistemin kendi yapısında oluşturulan log kayıt bilgilerinin farklılığı, yine sistemden sisteme değişmektedir. Örneğin güvenlik sistemlerinin en başında gelen firewall sistemleri, göreceli internet bağlantılarına ait bilgileri (zira bağlandığınız bir internet uygulaması bir çok arka plan bağlantıları da yaratabilmektedir), IP adreslerini, port numaralarını, bağlantı zamanlarını loglamaktadır. Bu işlem, işletim sisteminin bütününde, kullanıcı izinlerini kontrol etmek dahil bir çok proses ile birlikte yapılmaktadır, İncebacak (2007)

Tez ile ilgili çalışmalarının ilerleyen aşamalarında, işletim sistemleri, yada diğer uygulamaların bu işlemleri birlikte yapmasından dolayı oluşabilecek performans

darboğazları yanında, farklı sistemlerden toplanacak log hareketleri (log events), bazı önemli problemlerle bizi karşı karşıya bırakmıştır:

1. Çok yüksek sayılarda ve büyüklüklerde log kayıtları,
2. Log kayıt desenlerinin farklılığı,
3. İçeriklerin oldukça farklı olması.

Log kayıt desenlerindeki dizayn (context) ile ilgili problemin ana sebebi, hemen her uygulamanın kendi log formatına sahip olmasından kaynaklanmaktadır. Bu duruma sebep olan etkenler arasında log dosyaları analizinin uygulama alanlarının düşünüldüğü kadar geniş olmaması, log kayıtları dizayn edilirken okunabilirlik seviyesinin öncelikler arasında olmaması nedeniyle, çoğu kez bu işlevin uygulama geliştiricinin kendi inisiyatifine bırakılmış olması olarak gösterilebilir. Asıl sebep ise, bu konuda bir standart getiren bir organizasyonun olmamasıdır. Log yönetiminin önemli bir Bilgi Sistemi aktivitesi olduğu düşünüldüğünde, Tablo 2.1'de örneklenen sorumluluk modelinin, kurumlarda uygulanması gerektiği düşünülmüştür.

Log kayıtlarının içerikleri ve desen karışıklığı ile ilgili sorunlar, sistem yöneticilerinin tek bir ortamda veritabanı oluşturması ve kayıtları analiz edecek uzmanlar açısından ayrı bir zorluk oluşturmaktadır. Örneğin suça yönelik bulgu toplamaya çalışan bir araştırmacı, ortamda mevcut sistemlerin bu formatlarını öğrenmek zorundadır. Bu problem çok sayıda uygulama ve aygıtın bulunduğu geniş organizasyonlarda daha da büyük bir mesele haline gelecektir. Bu durum, BDDK düzenlemeleri gereği finans kuruluşlarında da karşılaşılan büyük bir mesele olarak karşımıza çıkmıştır.

Kimi uygulamalar için bilişim endüstrisi ile ilgili komitelerde standartlar açısından yönelik çalışmalar başlatılmış, ancak bu standartlar daha çok intrusion detection sistemlerinin log dosyaları formatı üzerinde yoğunlaşmıştır. Bu konuda çalışan komite IDMEF olarak adlandırılmıştır, H. Debar, D.Curry, B. Feinstein, (2007).

2.2 LOG KAYITLARININ ANALİZİ

Log kayıtlarının değerlendirilmesi süreci, çok sayıda event (olay-hareket) kaydının yer alacağı ortamlarda, bu konuda çalışanı verilerin ele alınması ve yönetilmesinin zorluğu

ile karşı karşıya bırakacaktır, Forte (2004), Gorge (2005). Logların okunma/gözlenme sıklığının ne olması gerektiği de, bilgi/deneyim eksikliği hissedilen diğer bir mesele olarak karşımızda çıkmıştır (Bkz. Tablo 2.2 Log Kayıtları Yaşam Döngüsü). Herhangi bir atak gerçekleştirimine ait süreç aksiyonlarını bulmaya çalışmak, çok sayıda ve farklı farklı sistemlerden gelen log kayıtlarını incelemek anlamına gelmektedir. Sadece girişim tespitlerinin değil, sistemlerdeki o ana dek farkında olmadığınız çeşitli olayların çözümlenmesi, iz kayıtlarının (log kayıtları) ve bu kayıtların analizi ile mümkün olabilir.

Log analizi, Log dosyalarından değerlendirilebilir bilgi edinmenin başlıca metotlarından biridir. Bu konuda uzun bir geçmiş olmamasına karşın, bir çok araç geliştirilmiştir. Ancak araştırmalardan elde edilen neticeler konusunda, log analiz sistemlerinin yeterliliği, kullandıkları veya içinde bulundurdukları istatistiksel bilgilerle sınırlı olabilmektedir, Forte (2004). Özellikle istatistiksel yöntemlerle elde edilen imza veritabanları, yine daha önce kaydedilmiş olay kayıtlarından yararlanarak tespit edilmiştir. Dikkat edilirse bu sonuç da, log analizi sonucu elde edilen bir çıktıdır. Kaydedilmiş olan imza bilgisi bünyesinde, önceki girişim/saldırı sürecinin nasıl oluştuğuna dair akışlarla ilgili bilgilerde yer alabilmektedir. Bu imza bilgilerinin yardımıyla, benzer bir olayın olması önceden engellenebileceği gibi, olaya neden olan sorumluyu belirlemek de mümkün olabilir. Genellikle bu kapsamdaki log analiz araçları belli sistem loglarına odaklanmaktadırlar (Router, switch, vpn, http Proxy hareketleri gibi).

Log Analizi bilişim sistemlerinde suça yönelik araştırmaların da doğruluğunu ortaya çıkarabilecek çalışmaların başında gelmektedir. Eldeki log dosyalarının bütünü bile, sistemler hakkında belli ölçülerde görünürlük sağlayabilir. Bu nedenle farklı yaklaşımların ve yorumların oluşturulması, olay kaydı (Event/Log) karşılaştırmalarında etkili olabilir. Bu yaklaşımlar sistemler ve uygulamalar ile gelişen aktiviteler hakkında yeni yorumlar elde etmek için olanak veren bir yol olarak düşünülmelidir. Bu prosesin bir girdi olarak etki edeceği alanlardan biri vaka yönetimi (Event management) olarak adlandırılmaktadır. Olay/Vaka yönetiminin amacı olay karşılaştırma işlevi ile bütünleşik bir bilgi elde etmektir.

Kaydedilen her bir olay iz kaydı (event), log kayıtları farklı yapı ve içeriklerde (farklı desenlerde) bile olsa, bir girişime ait aksiyonun analizini yapabilmek ve tarihsel bir şekilde ortaya bu aksiyonları ortaya çıkarabilmek açısından etkin olabilir. Aksine yeterince geliştirilmemiş ve bütünü içermeyen bir olay analizi ile bu olayların etkisi ve önemi açığa çıkarılamayabilir. Örneğin IDS sistemlerinde false positive olarak adlandırılan (yanlış anlaşılan) bulguların sebeplerinden biri olarak bu durum öne sürülebilir. Bunun tersine, belirli bir kayıt deseni yapısına yada içeriğine yükseltilmiş işlenmiş verilerin bulunduğu, bir veya daha çok log dosyasından elde edilen olay analizleri, sistemlerde ne tip olayların geliştiğinin daha iyi bir resmini sunabilecektir. Bu da imza veritabanlarının geliştirilmesinde etkili kullanılabilir.

Gomez, Herrerias (2005), olay bilgilerinin en basit şekli ile anlamlı olarak hem sistem, hem uygulama ve hem de güvenlik loglarından elde edilebileceğini belirtmiştir. Tez çalışmasında ise ilgili sistemlerden bu log verileri toplanırken, birlikte diğer sistemlerden alınan özelleştirilmiş log dosyalarının da depolanarak analiz çalışması için, diğer bir veri kümesi elde edebilmek üzerinde durulmuştur. Genel olarak bilgi sistemleri altyapılarında kullanımı oldukça yaygın, benzer veritabanları, uygulama ve sistemler olabileceği düşünüldüğünde, tez çalışması sırasındaki bu tür geliştirmelerin diğer ortamlarda da uygulanmasının mümkün olduğu düşünülmüştür. Bu işlemi gerçekleştirmek için farklı log dosyalama mekanizmasına sahip sistemlerinden belli ajanlar kümesinden faydalanarak log elde edilmesi ve bu olayları (event) birleştirerek filtreleme ve normalize etme hususları çalışılmıştır, (Bkz. Ek 4).

File System Auditor, OSSIM, Kiwi Syslog Deamon, Swatch, Infraskope, Manage Engine Event Log Analyzer gibi birçok log analiz programları bulunmaktadır. Bu tür yazılımlar, log kayıt bilgilerinin etkin ve kural tabanlı bir şekilde toplanmasını sağlar. Bu şekilde gereksiz log kayıtları elenir ve sadece kurumun güvenlik ilkeleriyle ilgili kritik olaylar kayıt altına alınır. Farklı kaynaklardan toplanan bu log bilgileri üzerinde tek noktadan kurumsal kayıt tutma süresi uygulanabilir.

Ayrıca USB bellek kullanımı, alınan ekran görüntüleri, önemli dokümanları yazdırma, HTTP tunneling, VPN, bluetooth, mesajlaşma uygulamaları, dial_up ya da diğer ağ ara yüzleri, Mac adres değişiklikleri ve sniffer,gibi kötü niyetli yazılımlara ait bilgiler

işletim sistemleri tarafından loglanmamaktadır. Bu olayları gözlemleyebilmek içinde log yönetim sisteminden yararlanmak gereklidir. Günümüzde Symantech firması tarafından geliştirilen log analiz yazılımıyla, kişisel bilgisayarlarınızda yüklü firewall veya IDS yazılım loglarını web üzerinden sisteme göndererek analiz etmeniz mümkündür.

Log yönetim sistemlerinde gerçek zamanlı izleme ve uyarı mesajları ile bu tür tehlikeleri tanımlamak ve harekete geçip gerekli güvenlik önlemleri alarak açığı kısa sürede kapatmak mümkündür.

2.3 LOG KAYITLARININ YÖNETİMİ

Log bilgileri, bu verilerin kaynağı olan süreç ve sistemlerdeki aksiyonlardan elde edilmektedir. Log yönetiminin sağlanabilmesi, sistemler ve bu sistemler üzerinde çalışacak tüm süreçler için yapılacak planlamalar ile ilgilidir.

Tablo 2.1: Log Yönetimi Yaklaşımları ve Sorumluluklar

NIST(2007) tarafından yayınlanan Guide to Computer Security Log Management isimli yayından derlenmiştir.

Log, bir organizasyon ve sistem içerisinde oluşan olayların kaydedilmesi olarak tanımlanır. (NIST)	**Bir organizasyon içerisinde Log Yönetimi ile ilgili sorumluluklar ve paydaşlar**	**Log Yönetimi için yapılması gereken ana fonksiyonlar**
• Log yönetimi organizasyonun ihtiyaçlarına göre belirlenmelidir. • Organizasyonlar Log Yönetimi altyapısını oluşturmalıdır • Log Yönetim sorumlulukları tüm paydaşlar ve sistem çalışanlarınca açık bir şekilde belirlenmelidir. • Log Yönetim ile ilgili standartlar oluşturulmalıdır.	• Sistem Altyapı ve Network Yöneticisi. • Sistem Güvenlik Yöneticisi. • Bilgi Güvenliği Yöneticisi • Bilgi Sistemleri Yöneticisi • Uygulama Paydaşları ve İş Birimleri • Denetleyiciler • Uygulama Geliştiriciler • Vaka Yönetimi Sorumluları (Computer Securtiy Incident Response Team)	• Log kaynaklarının konfigüre edilmesi • Log kayıtlarının filtrelenmesi, seçimi ve depolanması • Log analizleri konusunda sürecin belirlenmesi • Tanımlanabilen olaylar için aksiyonların belirlenmesi

2.3.1 Log Kayıtlarının Oluşturulmasını Zorunlu Kılan Nedenler

Çok yakın bir zamana kadar, sadece sistem sorunlarının çözmek amacıyla bilgi sistemlerinde log kayıtları dikkate alınırken, günümüzde ise hem bilgi güvenliği ve hem de standartlara uyum için loglama yapılmaktadır.

FISMA, HIBAA, SOX, COBIT, ISO 27001 gibi uluslararası standartlara göre log yönetimini zorunludur. Ülkemizde, 04.05.2007 tarihli 5651 sayılı kanunda internet suçlarını önlemeye yönelik olarak kurumların log yönetimi ile ilgili yükümlülükleri belirlemiştir.

Kredi kartlarının kontrolünde kullanılan, PCI veri güvenliği standardı da log yönetimini zorunlu kılan standartlara örnek verilebilir. PCI DSS Standardı içerisindeki 6 başlık altında 12 gereksinim ifade edilmiştir ve bunlardan biriside log yönetimidir.

Kurumların güvenlik politikasına, standartlara, kanun ve düzenlemelere göre bu konuda ortak bir noktanın henüz oluştuğunu söylemek mümkün değilken, an azından yasal gereksinimler nedeniyle, Internet'e bağlanan her bilgi sistemi cihazının hangi adresleri kullandığı, hatta hangi zamanda hangi internet kaynağı ile iletişim kurduğuna dair logların toplanması gereği yasalar ile belirlenmiştir.

Loglar, güvenlik denetimi sağlamak amacıyla merkezi olarak kaydedilmeli ve arşivlenmelidir. Bir sistemde kayıt altına alınabilecek olaylara örnek olarak:

1. Uygulamalara ait olaylar,
2. Ağ cihazlarına ait olaylar,
3. Veritabanı olayları,
4. Yedekleme,
5. Hatalar,
6. DHCP kayıtları,
7. Web aktiviteleri

sayılabilir.

Yönetimi bakımından birbirlerinden ayrılabilecek, bağımsız bir bilgi sistemi ortamında, (bunlar kurumlara veya kişilere ait ağ altyapıları olabilir) hangi veri kaynağından yada süreçten log toplanması gerektiği, standartlar, yapılan işin içeriği ve güvenlik ihtiyaçlarına göre olduğu kadar, bir araştırma sırasında, kanıt oluşturmak veya uğranılabilecek zararların tespitinde bütünlüğü sağlayacak unsurlar da dikkate alınarak belirlenebilir.

Kurumlarda Log yönetimi ve analizi bağımsız bir birim tarafından yapılmalıdır. Kurumların Bilgi Sistemleri Organizasyonlarında Log Yönetimi, genel olarak bilgi güvenliği çalışanlarının sorumluğundadır. (Bkz. Tablo 2.1)

Tez çalışmasının ilerleyen bölümlerinde de değinileceği şekilde, logların sistem yöneticisi tarafından alınması ve analiz edilmesi bir güvenlik riski oluşturur. Çünkü log kayıtları bu kademelerin içerisinde, değiştirebilir durumda kabul edilir.

2.3.2 Log Kayıtlarının Kalite Kontrol Gerekleri

Forte (2004) Log Karşılaştırma Sanatı başlıklı makalesinde, log dosyalarının korelasyonunun iki ayrı aktiviteye bağlı olduğunu belirmiştir: Girişim tespitleri ve Forensic (Suça Yönelik) network aktiviteleri. Bir kesintiyi önlemek üzere, karşılıklı ilişkide olan bu iki önemli aktivite göz ardı edilemez.

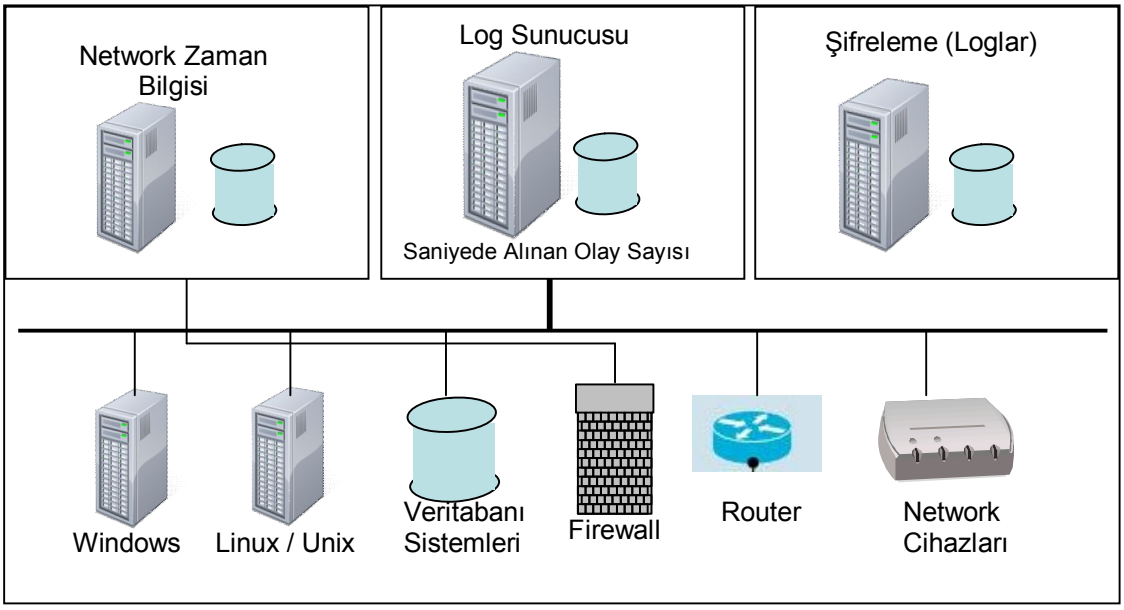
Forte (2004) her bilgi teknolojisi nesnesinin, olay günlüklerini (logları) üretme yeteneğine sahip olduğunu ancak, log kayıtlarının adli veya kanıt amacıyla kullanılacağı anda aşağıda açıklanan temel unsurlara sahip olması gerektiğini belirtmiştir:

2.3.2.1 Bütünlük (Integrity)

Log kayıtlarının herhangi bir tahrifata uğramamış, izinsiz olarak değiştirilmemiş ve belli bir filtreleme çalışması analiz edilmeden indirgenmemiş olması gerekmektedir.

2.3.2.2 Zaman damgası (Time stamping)

Log kayıtlarının olayın kaydedildiği anda, bu kaydın tarih ve saatinin, kabul edilebilecek yada zamanı garanti edecek doğrulukta alınması gerekmektedir. Bir olay veya özel durum sonrasında korelasyon yapılabilmesi açısında bu son derece gerekli bir unsurdur. Tez çalışmasının ilerleyen aşamalarında, logların olaylarla ilişkilendirmesi veya karşılaştırılması açısından anahtar olarak kullanılacak en önemli bilginin zaman damgası olduğu açık olarak ortaya çıkmıştır.



Şekil 2.2 : Log Yönetimi Kalite Sistem Şeması

2.3.2.3 Normalizasyon ve verinin anlamlı bir biçimde azaltılması

Normalizasyon ile söylenmek istenen şey, kaynak verilerdeki bütünlüğü bozmayacak şekilde, Log dosyası verilerinden uygun bir bilgi ortaya çıkarmak için, korelasyon aracının bu yeteneğe sahip olması gereğidir.

Verinin azaltılması (mantıklı bir biçimde) işlemi ise, seçilen kriterler uyarınca ve bir seri uygun olayı tanımlamak üzere veriyi filtre edebilmektir. Yine tez çalışması sırasında, veri sayısını azaltma işleminin, hem logların incelenmesi sırasında hem de logların veritabanlarına kaydedilmesi aşamasında mantıklı bir biçimde yapılabileceği, böylece, gereksiz zaman ve performans kaybının önüne geçilmesini sağlanacaktır.

2.3.3 Log Kayıtları Yaşam Döngüsü

Sistem performans değerleri ve bilginin kalite kontrolü için log kayıtlarının depolanması gerekli kılan durumlara ait bilgiler için yapılan araştırmada, NIST(2007) tarafından yayınlanan 'Guide to Computer Security Log Management' isimli kitaptan yararlanılmıştır. Log kayıtlarının yasal düzenlemeler tarafından belirlenen süreler dışında, problem kök sebep analizleri çalışması, şüpheli işlemlerin geriye dönük araştırılması ile ilgili durumları da göz önüne alınarak, belli süreler boyunca saklanması gerekmektedir.

Tablo 2.2'de Log dosyalarının ne kadar süre saklanması gerektiği, log sunucusuna aktarım süreleri ve hangi log verilerinin inkar edilmezlik açısından şifrelenmesine yönelik bilgiler yer almaktadır.

Tablo 2.2: Log Kayıtları Yaşam Döngüsü

NIST(2007) tarafından yayınlanan Guide to Computer Security Log Management isimli yayından derlenmiştir

Kategori	Kritik Olmayan Sistemler	Kritik Sistemler	Çok Kritik Sistemler
Log kayıtları hangi süre ile saklanmalıdır	1 veya 2 hafta	En az bir ay	3 ay 1 yıl, *Yasal Düzenlemelere tabi kuruluşlarda en az 1 Yıl
Log rotasyonu hangi sıklıkla yapılmalıdır (Log Sunucusuna)	Opsiyonel (Her hafta bir kez veya 25 MB)	Her 15 ile 60 dakika arasında veya her 2-5 megabyte veride	Her 5 dakikada
Loglar hangi sıklıkta analiz edilmelidir	Haftada 1 kez	Her 24 saatte bir	Her 12 saatte
Logların Şifrelenmesi	Opsiyonel	Evet	Evet

Böyle bir çalışma ortamındaki temel sorun, farklı platform, ortam ve sistemlerin üzerinde bulunan algılayıcıların birlikte çalışabilmesi olacaktır. Birlikte çalışabilirlik denildiğinde, buna yönelik standartların oluşturulması gerekeceği, açıktır. Bu doğrultuda atılan adımlardan birisi, IETF' in (Internet Engineering Task Force) "Intrusion Detection" çalışma grubu tarafından önerilen "Intrusion Detection Message Exchange Format (IDMEF)" standardıdır. IDMEF, uyarıların (alert) formatını ve bir uyarı değiş tokuş protokolünü tanımlar. Bundan başka, algılayıcıların

gördüğü(algıladığı) şeyler üzerinde ortak bir anlayışın oluşturulması gerekir. Aksi halde, farklı algılayıcılar aynı tür bir izinsiz giriş bulmaya çalıştıkları halde, birbirleri ile uyuşamayacaklar, aralarında anlaşma sağlanamayacaktır. Diğer bir olası gelişme, yazılıma dayalı "IDS" sistemlerinin donanım tabanlı algılama teknolojisi haline dönüştürülmesi olabilir. Bu şekilde analizler daha hızlı yürütülebilecektir.

Tablo 2.3'de Cyber Security Industry kurumu tarafından yayınlanan ve kişisel veya kurumsal sistemlerde kurulu güvenlik uygulamaları ve yapılarının hangi oranda kullanım yüzdesine sahip olduğuna dair bir araştırma sonucu verilmiştir.

Tablo 2.3: Güvenlik Teknolojileri Kullanım Oranları

Kaynak: Cyber Security web site (2008)

2008 Cyber Security Industry (CSI) Kullanılan Güvenlik Teknolojileri (2008)	
Anti-virus software	**97%**
Anti-spyware software	80%
Application-level firewalls	53%
Biometrics	23%
Data loss prevention / content monitoring	38%
Encryption of data in transit	71%
Encryption of data at rest (in storage)	53%
Endpoint security client software / NAC	34%
Firewalls	94%
Forensics tools	**41%**
Intrusion detection systems	69%
Intrusion prevention systems	54%
Log management software	**51%**
Public Key Infrastructure systems	36%
Server-based access control lists	50%
Smart cards and other one-time tokens	36%
Specialized wireless security systems	27%
Static account / login passwords	46%
Virtualization-specific tools	29%
Virtual Private Network (VPN)	85%
Vulnerability / patch management tools	65%
Web / URL filtering	61%
Diğerleri	3%

3. AĞ GÜVENLİĞİ VE LOG YÖNETİMİ İLİŞKİSİ

İnsanların bilgi sistemlerini kullanarak bir ağa bağlı olarak çalıştıkları süre gitgide çoğalmaktadır. Bu durum fiziksel ve mantıksal güvenlik meselelerinden daha farklı olan yeni yeni zayıflıklara da yol açabilmektedir. Teknolojik gelişim network güvenliğinin önemini her geçtiğimiz gün artırmaktadır.

3.1 AĞ GÜVENLİĞİ GEREKLERİ:

Kişisel bilgisayarların kullanımının da her geçen gün artması, birden çok bağlantının sağlanabileceği internet şebekelerinin artık hemen hemen her evden kurulabilecek kadar yaygınlaşmış olması, internete erişim için sağlanan kolaylıklar, bütün bu açıklıkların da varlığını artırmaktadır. Bu durum gerek internet gerekse lokal ağlarda hangi zaman diliminde hangi bilgisayar ile network üzerinde bağlantının kurulduğuna dair log kayıtlarını tutma gereğini de, ileride olası soruşturmalar nedeniyle gerekli kılmaktadır. Bu konuda hazırlanan yasal düzenlemeler bu gerek üzerine kurulmuştur.

Bilgisayarlarda kullanılan güvenlik sağlamaya yönelik araçlar, sorunsuz, daha şeffaf ve çok daha fazla esnek yapılara sahip olmak durumundadır.

Artan sayıdaki ağa bağlanma olanakları ve kişisel bilgisayarlar sonuç olarak sayılamayacak miktarda güvenlik riskleri oluşturmaktadır. Web üzerinde sörf ve elektronik posta alışverişleri, ağlar arasında bir çok politika belirlenerek kullanılan firewall korumaları ile bir ölçüde koruma altına alınabilmekte, bu ateş duvarları yukarıda bahsedilen nedenlerle sürekli gözlenen bir sistem olmak durumunda kalmaktadır.

Güvenlik kelimesini sadece kötü niyetli ataklara karşı korunmak şeklinde adlandırmak yerine, etkin ve iyi kontrol edilmiş sistemlerin geliştirilmesi olarak ifade etmek günün koşullarına daha uygun bir tanım olabilir.

3.2 ZAYIFLIKLAR, TEHDİTLER VE ATAKLAR (VULNERABİLİTİES, THREATS AND ATTACKS)

Ağ güvenliği için kullanılmakta olan üç adet yaygın terim bulunmaktadır: güvenlik açığı diğer anlamı ile zayıflıklar (vulnerability), tehditler (thread) ve ataklar. Güvenlik amacı ile kullanılan cihazlar dahil olmak üzere, router, switch, masaüstü bilgisayarlar, sunucular gibi hemen her ağ cihazının bir eksikliği olabilir. Üstelik yeni açıklıkları araştıran veya bu tehditlerden fayda sağlamaya yönelik araştırma yapan bir çok istekli insan olduğu bilinmektedir. Çeşitli araçlar, scriptler (program parçacıkları) ve yazılımlar ile network ve network cihazları arasında bir çok atak girişimi gerçekleştirilebilir. Bu atakların sıklıkla ulaşmaya çalıştıkları sistemler ise sunucular, kişisel bilgisayarlardır.

3.2.1 Zaafiyetler (Vulnerabilities)

Başlıca üç tip zayıflık (Vulnerabilities) bulunmaktadır.

3.2.1.1 Teknolojik Zaafiyetler (Technology Weaknesses)

Bilgisayar sistemleri ve ağlar kendi yapıları içerisinde zayıflıklar içermektedir. Bunlar arasında TCP/IP protokol demeti ve ağ ekipmanları da bulunmaktadır. Özellikle ağ ekipmanlarının fiziksel olarak güvenliği sağlanmalıdır.

3.2.1.2 Konfigürasyon hataları (Configuration weaknesses)

Network yöneticilerinin veya mühendislerinin hangi kontrol zayıflıkları olduğuna dair bilgi edinme ihtiyacı gitgide artmakta ve en doğru şekilde konfigürasyon konfigürasyonlar yapmaları gerekmektedir. Bir çok sistemde güvenlik açısından yeterli derece yapılandırma bulunmamaktadır. Örneğin unix sunucularda, direk olarak sisteme ulaşan kullanıcıların yanlış şifre denemelerinde ön tanımlarda (default configuration) bir engelleme yapılmadığı bu tez çalışmasında rastlanılan bir husustur.

3.2.1.3 Güvenlik politikalarında zayıflıklar (Security weaknesses)

Güvenlik politikalarındaki zayıflıklar önceden tahmin edilemeyen tehditlerin oluşumuna neden olabilirler. Kullanıcıların takip etmekte zorlandıkları politikalar, beklenmedik güvenlik açıkları doğurabilir. Örneğin kullanıcı kodu ve şifrenin paylaştırılması gibi.

3.2.2 Tehditler (Threats)

Başlıca 4 adet tehdit türü bulunmaktadır.

3.2.2.1 Yapısal olmayan tehditler (Unstructured threats)

Yapılandırılmamış tehditler çok deneyimsiz kişilerce, kolay yollardan bulunabilecek hacker ve cracker araçları sayesinde kolayca kullanılabilir.

3.2.2.2 Yapısal tehditler (Structured threats)

Yapısal tehditler, bu tip girişimler konusunda bir çok deneyime sahip ve bir şekilde motive olmuş girişimcilerin ürünüdür. Zayıflıklar konusunda düşünülemeyecek ölçüde bilgi sahibi olabilirler. Bu gruptaki kişiler sık sık kurumsal firmaların ağlarına sızmak isteyebilirler. Büyük dolandırıcılık ve hırsızlık vakaları bu tip girişimlerin sonucudur. Sosyal mühendislik saldırıları da bu grup içerisinde yer alır.

3.2.2.3 Harici tehditler (External threats)

Yapısal tehditler içeriden veya dışarıdan başlatılarak yönetilebilen girişimlerdir. Direk olarak dışarıdan içeriye doğru bir tehdit grubunun oluşmasının nedeni, erişim kolaylığı sağlamak üzere oluşturulmuş olan Vpn, çevirmeli ağ üzerinden erişim veya kablosuz network erişimleri üzerinden gerçekleşebilir. Unutulmuş açık bir bağlantı bu tip bir tehdide örnek olarak gösterilebilir.

3.2.2.4 Dahili tehditler (Internal threats)

Bir sunucu üzerinde hesap sahibi olan, yetkilendirilmiş olarak ağa erişebilen veya fiziksel olarak network cihazlarına ulaşabilen lokasyonlardaki her tip kullanıcı, iç tehdit unsuru olarak gösterilebilir.

Tehditlerin, yukarıda bahsettiğimiz zayıflıkları kullanarak bir girişimden sonuç alabilmesi için atak işleminin gerçekleştirilmesi gerekir. (Zaafiyet üzerine gelen tehdit unsurları, girişimler ile riski meydana getirirler.)

3.2.3 Ataklar (Girişimler)

Ataklar aşağıda anlatıldığı gibi 4 ayrı sınıfa ayrılabilir.

3.2.3.1 Keşif amaçlı ataklar (Reconnaissance)

Keşif atakları, erişim hakkı bulunulmayan sistemlere erişmek için servisler hakkında bilgi toplamayı, sistem haritasını çıkarabilmeyi, zayıflıkları tespit etmek üzere (açık bir pencere var mı?) araştırma işlemleridir. Bu durum bilgi toplama aşaması olarak da adlandırılabilir. Çoğunlukla izinsiz erişim ve servis kesintisine (Denial Of Service) neden olabilecek girişimlerdir. Sadece DOS amaçlı atak tipi de bulunmaktadır. Bu tip atakların önceden kestirimi oldukça güç bir konudur. Ağ üzerinde bilgi toplamak üzere kurulmuş bir araç ile girişimi tespit etmek üzere, önceden belirlenmiş veya kestirim yaparak alarm oluşturmaya yönelik girişim tespit etme sistemleri, toplanan log kayıtlarından yararlanmak durumundadır.

3.2.3.2 Erişim sağlamaya yönelik ataklar (Access)

Bir hesap ve şifresi olmayan girişimcinin, bir cihaz veya sisteme erişim yeteneği elde etmesine yönelik atak tipidir. Bu nedenle sistemde yetkisiz erişim iz kayıtlarının tutuluyor olması bu tip girişimlerin tespiti açısından yararlı olacaktır. Bloke edilmemiş hesaplar üzerinden sayısız şifre denemesi yapabilen atak araçları bulunmaktadır. Bu durum en çok fonksiyonel kimlikler (bir yazılımın veritabanına ulaşmak için kullandığı gömülü veya şifreli hesaplar) ile sistem yöneticisi şifreleri için geçerli olabilmektedir. İş riskleri nedeni ile bloke etme özelliği konulmamış hesaplar bu tip tehditlere maruz kalabilmektedir.

3.2.3.3 Servisleri kesintiye uğratmaya yönelik ataklar (Denial of Service (DoS))

Servis kesintisi atakları (Denial of service) bir ağı kullanılmaz hale getirmek, bozmak veya kasıtlı olarak belli kullanıcıları servis kesintine uğratmak amaçlıdır. Bu tip ataklarda servis performanslarının düşmesi ile neticelenen pek çok durum da yaşanmaktadır. Bir çok sistem daemon veya servis olarak adlandırılan, sürekli olarak talep bekleyen bir bilgi sistemi mimarisi barındırır. Bu nedenle bir şekilde bu talebi internet üzerinden genel kullanıma açık veya sınırlandırılmış olarak erişilecek şekilde kullanıcılar yapmaktadır. Bir şekilde bu talebi yapabilecek noktada olan girişimci, bir çok yöntemle servis kesintisi atakları gerçekleştirebilir. Servis kesintisine sebep olan

kaynak noktanın tespit edilmesi önlem almak açısından gerekebilir. Ancak yukarda anlatılan nedenlerden dolayı tehdit olasılığı yüksek olan bir atak tipidir.

3.2.3.4 Solucanlar, virüsler ve Truva atları (Worms, Viruses and Trojan Horses)

Kötü niyetli yazılımlar sistemlere zarar vermek, bozulmalarını sağlamak, girişimciye her tür bilgiyi transfer edebilmek, servis kesintisi yaratabilmek amaçlıdır. Solucanlar, virusler ve Truva atları olarak adlandırılmaktadır. Hemen her kötü niyetli yazılımın çalışma şekline, bilgisayar üzerinde kapladığı alan ve yerleştirdiği servis ile yaptığı işe göre oluşan bir imzası (signature) bulunmaktadır. Bu da network üzerinde dinleme ve log toplama faaliyetlerinin gereğini ortaya çıkartır. Zira bu imzalar sayesinde, iz kayıtlarından yararlanarak bu tip atakları daha fazla zaman geçirmeden tespit etmek mümkün olabilir.

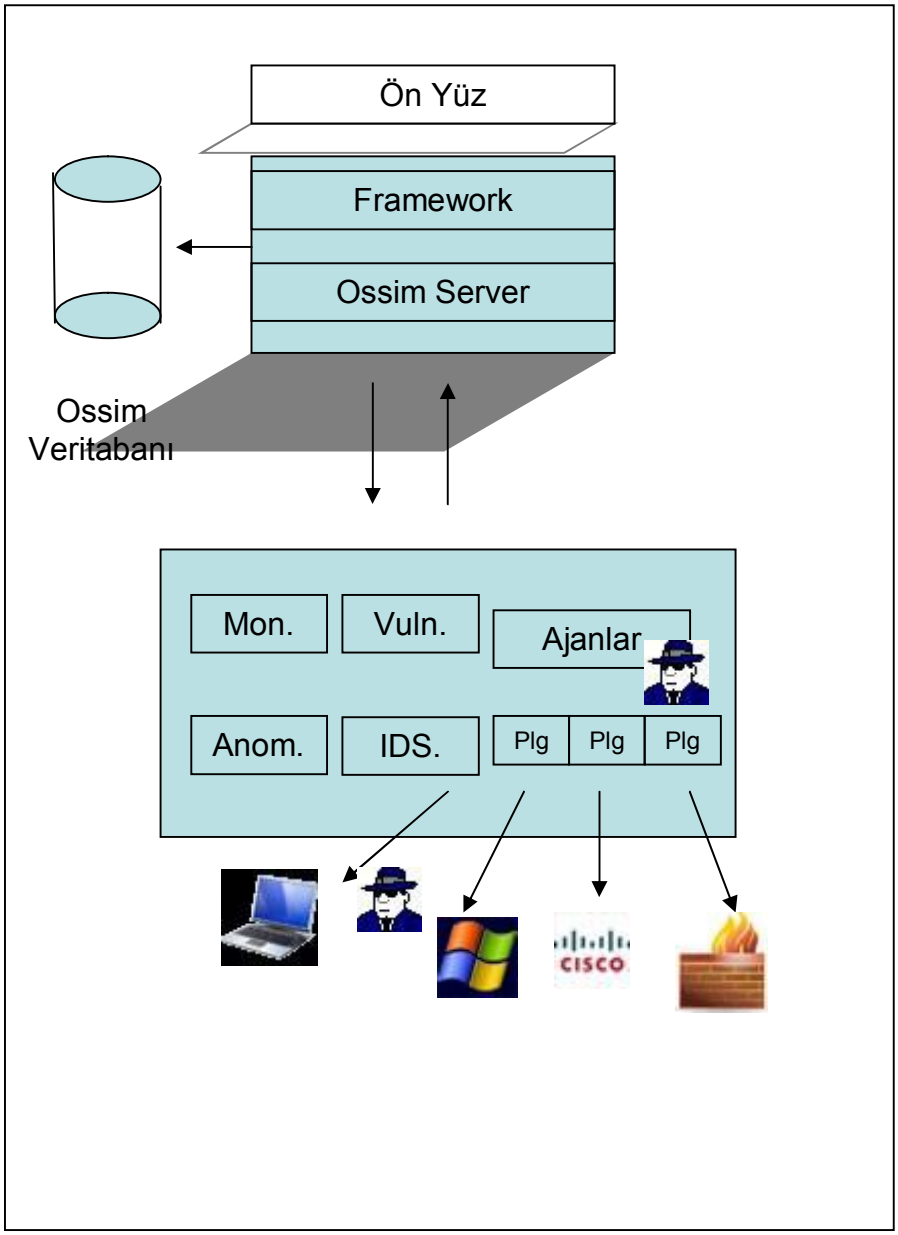
4. OSSİM AÇIK KAYNAK KODLU BİLGİ GÜVENLİĞİ VE LOG YÖNETİM SİSTEMİ

Ossim ana bileşenleri ile bilgi güvenliği yönetim sistemi olarak tanımlanabilir. Tez çalışmasında esas alınan özelliklerini kullanarak bir log depolama ve log yönetim aracı olarak kullanım amacına hizmet edebileceği gösterilmiştir.

4.1 OSSİM MİMARİSİ

Client/server (istemci/sunucu) mimarisinde çalışır. Ossim ajanlarının ve merkezi veritabanına veri göndermek isteyen sistemleri istemciler, sunucuyu ise Ossim servisinin çalıştığı çatı (framework) ve servis olarak tanımlayabiliriz. Ossim sunucu yapısı (framework daemon) bilginin bir bütün olarak değerlendirilmesi için verilerin gösteriminin yapıldığı bir ön yüz içerir. Bu ön yüz web tabanlı kullanıcı ara yüzüdür.

Minimal düzeyde Ossim altyapısı bir ajan, bir sunucu, bir servis çatısı ve bir veritabanından oluşur. Bu bileşenler farklı sistemler üzerinde yer alabileceği gibi tek bir sunucuda da yer alabilirler.

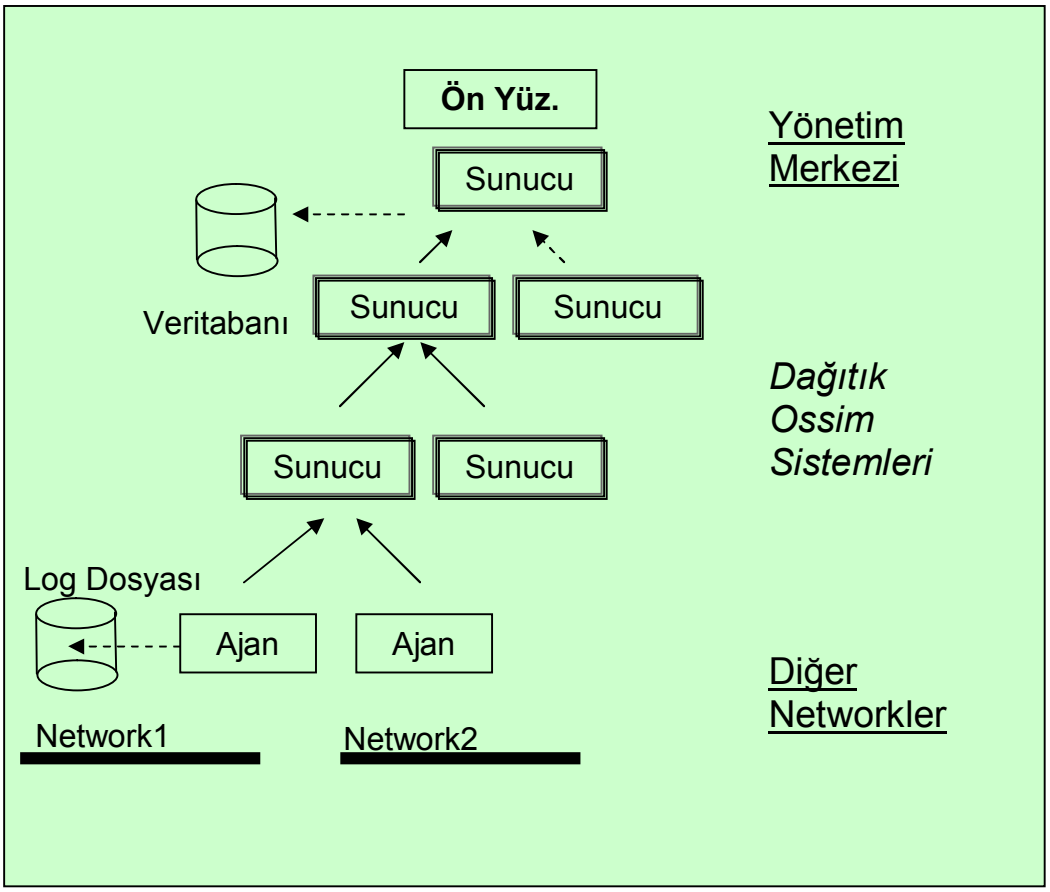


Şekil 4.1: Ossim Mimarisi

Plg. : Plugin (Eklentiler)
Anom. : Anonymous (Etherhet trafiğinde rasgele dinlenebilen paketler)
Mon. : Monitor (Gözlem)
Vuln. : Zayıflıklar (Vulnerability)

4.2 OSSİM HİYERARŞİK YAPISI

OSSIM aracı ile bir ağaç yapısı şeklinde, çeşitli sunucu ve ağ sistemlerinde veri toplamak mümkündür. Şube, bölge yapısına sahip işletmelerde kullanım mümkündür.



Şekil 4.2: Ossim Hiyerarşisi

Araştırmamızda aynı anda çok kullanıcılı ve çok sayıda sistem bulunan bir bilgi sistemi alt yapısında, network üzerinden geçen paketlerin yakalanarak log sunucusuna depolanması konusunda performans sıkıntıları yaşanmıştır. Bu nedenle yönetsel anlamda ve network aktif cihazları üzerinde tanımlamalar yaparak, bölümlemeler sağlanabildiği takdirde birden çok Ossim sunucusu kullanarak hem gerekli sistemlerin logunun alınması, hem de bu sistemlerce üretilen aktivitelere ait olay/iz kayıtlarının eksiksiz bir logunun alınması mümkün kılınmaya çalışılmıştır.

4.3 AĞ HARİTASI (MAPPING)

Network haritası, bulunulan bilgi sistem ağı üzerindeki tüm network cihazlarının araştırılması ve üzerindeki servislerin belirlenerek, çalışmakta olan topoloji tespitinin yapılması işlemlerinin bütünüdür. Ossim kurulduğu bilgi sistemi altyapısında network topolojisini araştırarak şemasını elde edebilecek bileşenlere sahiptir.

Bir ağın performanslı bir şekilde izlenebilmesi için, bu ağ haritasının çıkarılması ve bu ağa uygun doğru alt yapıların kurulması gerekmektedir. Tez çalışmasında, çalışmanın yürütüldüğü kurum içerisinde kritik olarak tespit edilmiş sunucular için, bir cisco switch üzerinde konfigürasyon yapılmış, 6 adet sunucunun izlenmesine yönelik olarak, tüm bu sunuculara giden network (ethernet) paketlerinin bir kopyasının yansıtıldığı özel bir port ile (mirror port) Ossim sunucusunda mevcut network kartlarından birine bu paketleri izlemek amacıyla direk bir bağlantı sağlanmıştır. Aynı Ossim sunucusunun diğer network bağlantısı üzerinden yönetimsel konsol işlemleri ve web tabanlı kullanıcı işlemleri gerçekleştirilmiştir.

Bu aşamada Log Yönetiminde performans elde etmek üzere araştırma süresinde karar verilen konular:

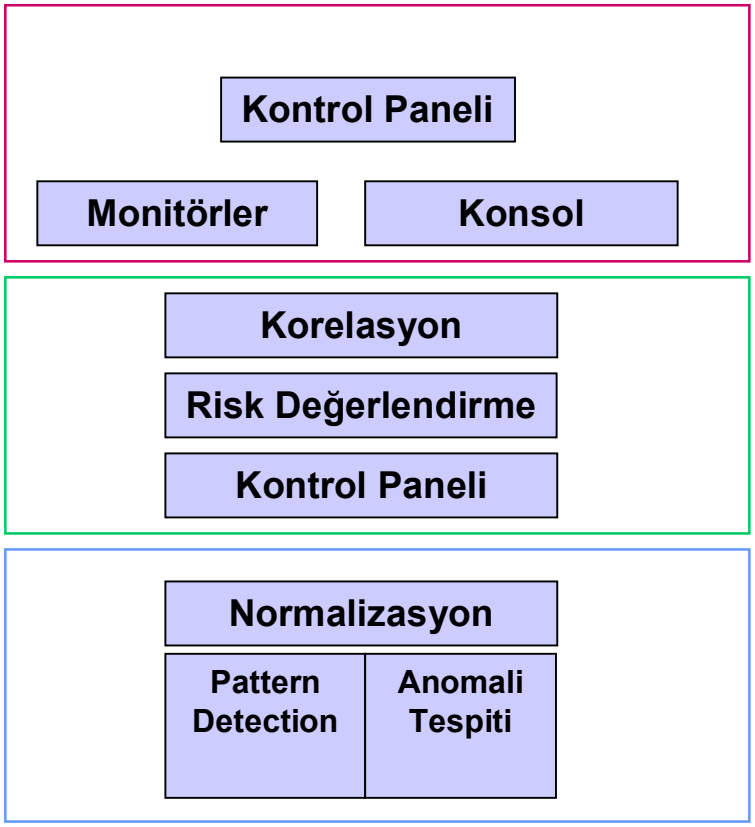
- Network haritasının oluşturulması,
- Topolojisi ortaya çıkarılan network üzerinde kritik sistemlerin belirlenmesi,
- Kritik sistemlerin network cihazları üzerinde hangi port (bağlantı) noktalarına bağlı olduklarının belirlenmesi ve/veya belli sistemlerin o topolojide toplandığı sanal ağlar yaratılması.

Ek 3'de switch konfigürasyonu nasıl yapılabileceği bir cisco switch üzerinde tanımlanmıştır.

4.4 OSSİM VERİAKIŞI

Ajan olarak isimlendirilen eleman, sunucular ve ossim çatısı, farklı katmanlarda veri işleyen üç bileşendir. Veri toplama ve normalize etme işlevi (kısmen) ajanların

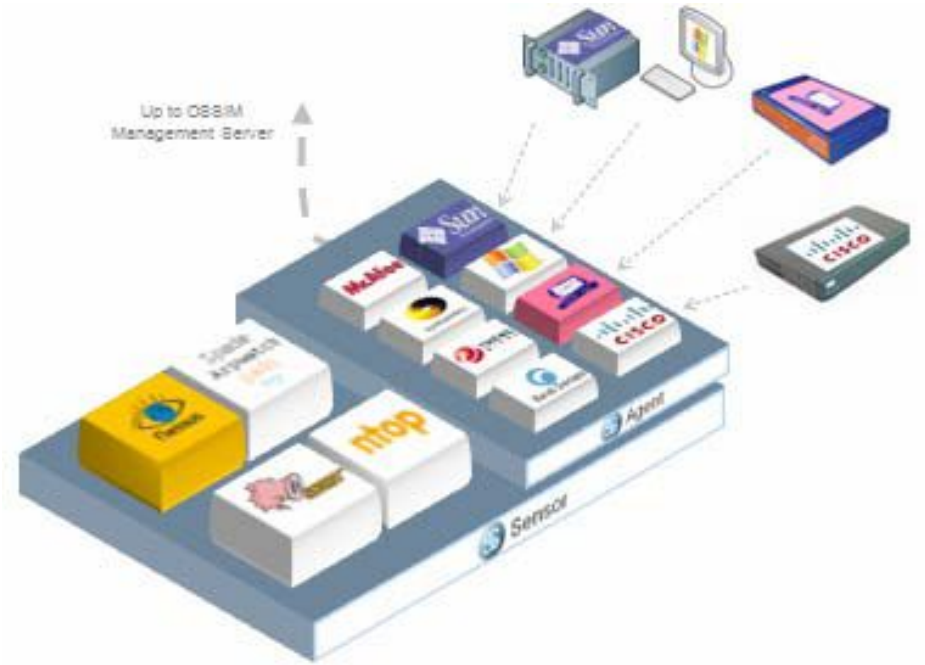
görevidir. Önceliklendirme, risk yönetimi ve korelasyon sunucu tarafında gerçekleşir. Ossim çatısı (framework) servislerin izlenmesi, konsol ve kontrol süreçlerinin sahibidir.



Şekil 4.3: Ossim Veri Akışı

Normalizasyon işlemlerini Ossim ara yüzü haricinde, veritabanına direk ulaşarak yapmak mümkündür. Esasen Ossim Log Sunucusuna ait log veritabanına dışarıdan erişim mümkün kılınmayacak şekilde ayarlanmıştır. Veritabanı konfigürasyonları buna uygun şekilde yapılanmış olarak sistem kurulmaktadır. Kuruluşun üzerine yapıldığı debian linux işletim sistemi firewall konfigürasyonları bu veritabanına erişime izin vermez. Ancak sistem yöneticisi, yönetim işlemleri için hem mysql veritabanında hem de linux işletim sisteminde ayarlamalar yaparak, kendi iş istasyonu üzerinden veritabanının bir kopyasını oluşturarak normalizasyon işlemlerini test ortamında gerçekleştirebilir ve uygun olduğunu düşündüğü takdirde, gerçek log veritabanında bu değişiklikleri yerine getirebilir. Uzak bağlantı için Mysql> GRANT ALL PRIVILEGES ON db.* TO root@'max.here.com'; komutunun girilmesi gerekmektedir.

4.5 OSSIM AJAN(AGENT) YAPISI



Şekil 4.4: Ossim Ajanları

Bir Ossim ajanı veri toplamak ve bu veriye ait deseni tek bir formata indirgemek şeklinde önemli bir (normalize) rolü gerçekleştirir. Veriler (log kayıtları) sistemde yer alan aygıtlardan (router, switch, kişisel bilgisayar, sunucu v.b.) açık kaynak kodlu araçlar ile toplanırlar.

Windows, Linux veya Unix işletim sistemi ile işletilen sunucular, ateş duvarları ve network cihazları bu ajanlara bilgi iletirler. Örneğin Snort girişim tespit sistemi de SysLog dosyasına bu ajan üzerinden bilgi taşıyabilir.

Diğer bir çok sistemden de log toplamak mümkündür. Tez çalışmasında buna ait örnekler oluşturularak, ossim sisteminden yararlanılarak özelleştirilmiş kayıtlar log sistemine dahil edilmiştir. Bu şekilde, veritabanlarından, belli bir standarda uygun olmadığı halde diğer log dosyalarından ve düz metin (text log) dosyalarından log elde etmek mümkündür.

Ossim bünyesinde yer alan açık kaynak kodlu ürünler, kötü niyetli verileri tarama, izleme ve tespit etme yeteneğine sahiptir. Bu sistemde bilgiler (veri) depolandıkları anda, hemen normalize edilirler. Örneğin açık kaynak kodlu network tarama (scan) araçlarından Snort olay imza veri tabanını referans alarak (signature) vakaları tespit etmek amaçlı çalışır.

Ossim ajanlardan gelen bilgileri (bu ajanlar iyi bilinen sistem araçları ve network cihazları için ossim açık kaynak kodu içerisinde geliştirilmiştir), yazılmış olan konfigürasyon amaçlı plug-in'ler aracılığı ile belli bir log verisi formatına getirilmektedir. Format unix-linux sistemlerde kullanılan syslog yapısına göre düzenlenmiştir. Yeni bir araç veya network cihazı için yeni bir konfigürasyon dosyası oluşturmak gerekir. Bu konfigürasyonların yapıldığı plug-in'lere ait örnek Kod 8.2'de bulunmaktadır. Konfigürasyon dosyasının iki önemli amacı şunlardır;

- Verinin hangi adresten alınacağını gösteren referans
- Bu verinin anlamlı bir syslog düzenine ne şekilde getirileceği.

Verinin belli bir formata çevrilmesi işlemi, "regular expression" (düzenli ifade) olarak adlandırılan (veri ifadesinin düzenlenmesi) temeline dayanır. Bu konuda üzerinde çalışılan Ubuntu linux sistemlerde çalışan yine açık kaynak kodlu regular expression ve windows sistemlerde çok benzer özelliklerle çalışan Kodos araçlarından tez içerisinde yararlanılmıştır.

Ossim çatısı, ağ üzerinde hangi cihazları izleyip, tarayıp, kontrol edeceğini bilmek isteyecektir. Bu sistem network "mapping" (ağın haritasının çıkarılması) amacını taşır. Bir network yapısı haritası belirlendiğinde, ağ üzerindeki ilgili sistemlerin zayıflıkları, zayıflık tarama araçları ile taranarak kontrol edilir. Ossim bu haritalama sayesinde ve zayıflıkları taramaya başlamakta, bu andan itibaren pattern üzerinde, doku ve anomali tespit işlemlerini de başlatmaktadır.

5. OSSİM SİSTEMİNİ OLUŞTURAN AÇIK KAYNAK KODLU BİLEŞENLER

5.1 NMAP

Ossim içerisinde yer alan bileşenlerden biri olan, açık kaynak kodlu Nmap, ağı izleyen bir denetim aracıdır. Araç sistemleri (hosts), açık bağlantı noktalarını (port), işletim sistemlerini ve bir çok iyi bilinen uygulamanın versiyonunu keşfedebilmektedir.

NMAP Ossim içerisinde farklı iletişim seviyelerinde çalışmaktadır. İlk aşamada, hızlı bir şekilde sistemleri keşfetmek üzere, internet protokol ping özelliği ile tarama yapar. Eğer ping isteklerine cevap alamama durumu olursa, o sistem üzerindeki açık bağlantı noktalarını TCP (Transmision Control Protocol) özelliğini kullanarak tarama işlemine devam eder. Bu işlemin elde ettiği sonuç şu şekilde açıklanabilir: Eğer TCP scan bir port erişimi sağlayabilirse, tarama hemen durdurulur ve sistem keşfedilmiş olur. İkinci bir tarama başlayarak diğer portları da keşfetmeye çalışır. Bir sistemin ağ üzerinde verdiği servisler, bir port numarası ile adreslenir ve bu tarama sırasında kullanılacak port numaraları aralığı (örnek, 1 numaraları portdan 5000 numaralı porta kadar) TCP tarama sırasında Nmap tarafından belirlenir. NMAP ayrıca, UDP hizmetleri taraması da yapmaktadır. Fakat bu tarama network üzerine ekstra bir yük getirebilmektedir.

Tablo 5.1: Komut satırından örnek bir Nmap taraması

```
1 ----------------------------------------------------
IP: 10.10.10.1
MAC(s): 0:06:25:78:20:75 (2005/06/11 12:58:11)
VENDOR: Cisco WPC11 v2.5
ICMP: Enabled
Port Service Application
80 www Unknown HTTP (HTTP/1.1)
2 ----------------------------------------------------
IP: 10.10.10.83
MAC(s): 8:00:20:A0:14:A5 (2005/06/11 12:58:11)
VENDOR: Sun Microsystems Inc.
Port Service Application
22 ssh OpenSSH 3.8.1 (Protocol 2.0)
80 www Apache/1.3.29 (Unix) PHP/4.3.10 mod_ssl/2.8.16
```

Nmap: http://insecure.org/nmap

5.2 AÇIKLIK TARAMA (VULNERABİLİTY SCANNİNG)

Açıklık tarama işlemi sistemler üzerindeki zayıflıkların tespitidir. Bir anlamda hacker (sistemi ele geçirme) çalışmasına benzetilebilir. Yetkisi olmadığı halde, yetkili bir kullanıcı gibi sisteme alternatif yollardan log-in olanakları araştırılır. Ancak açıklık bulunduğu takdirde bundan faydalanma ve deneme yolları araştırılmaz. Bu noktadan sonrası şüpheli hareketlere girecektir.

5.2.1 Nessus

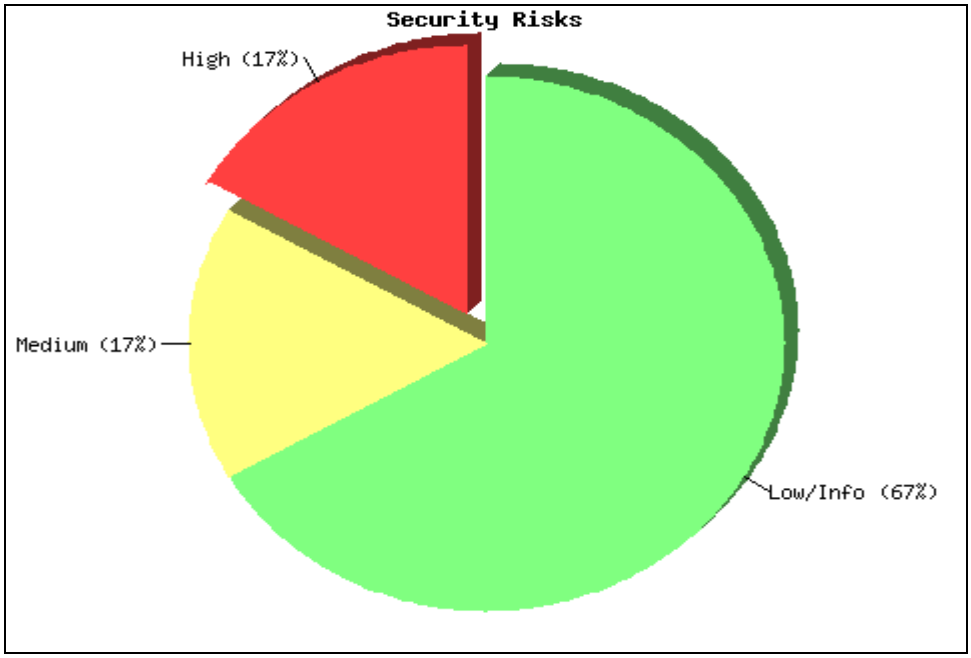
Nessus açık kaynak kodlu bir zayıflık tarama aracıdır. Tamamen kendi kendine çalışacak biçimde testler yaparak, güvenlik zaafiyetlerini ortaya çıkarabilecek şekilde dizayn edilmiştir.

Nasıl çalışır?

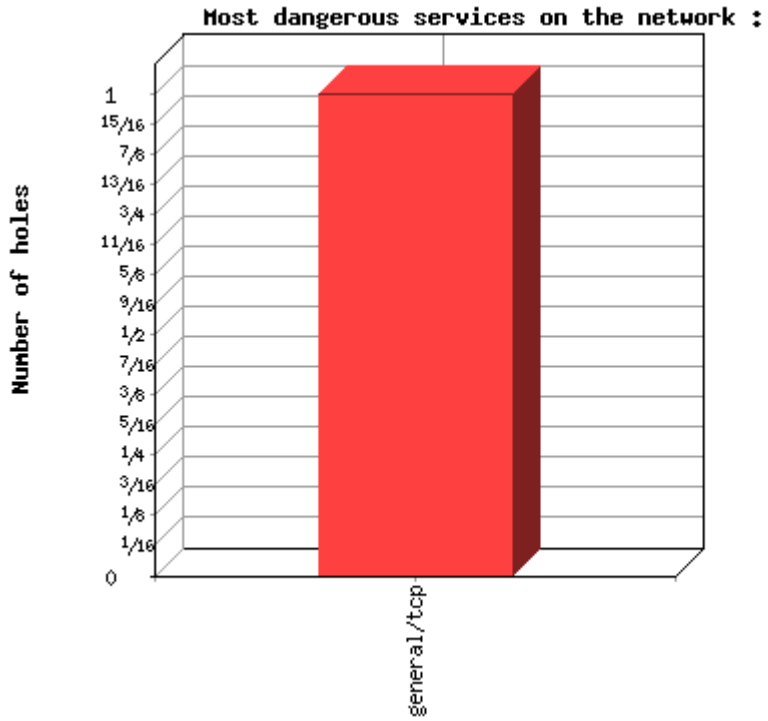
Nessus sunucu-istemci (client-server) mimarisinde çalışmaktadır. Sunucu internet bağlantısı ile elde ettiği güncel test bilgilerinden yararlanarak network üzerinde testleri gerçekleştirir. Tez çalışmasında Ossim sunucusundan bu testler gerçekleştirilmiştir. Nessus client uygulaması konfigürasyon ve raporlama için gereken bilgileri toplar. Sunucu ve istemci arasındaki iletişim, şifrelenmiş bir şekilde (SSL) gerçekleşir. Üç tür nessus istemcisi bulunmaktadır;

- Unix sistemler için grafik ara yüz (Native Unix Gui.),
- Komut satırı istemcisi,
- Windows sistemler ara yüzü.

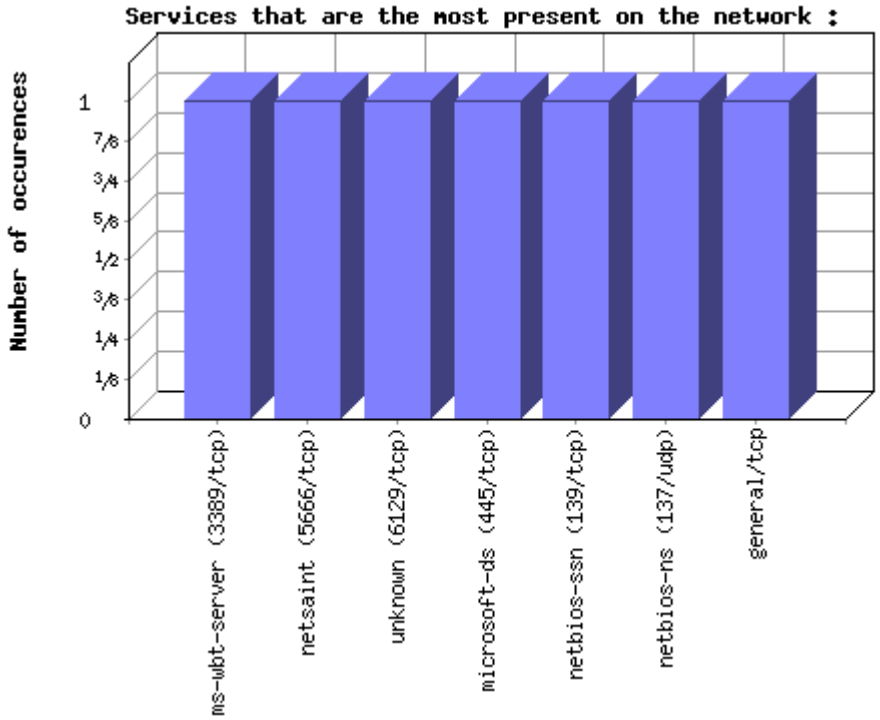
Nessus: http://www.nessus.org/nessus/



Şekil 5.2: Nessus tarama sonucu elde edilen güvenlik açıkları bilgisi
Kaynak: Tez çalışması için kurulan Ossim Sistemi



Şekil 5.3: Nessus tarama sonucu elde edilen en tehlikeli servis bilgisi grafiği
Kaynak: Tez çalışması için kurulan Ossim Sistemi



Şekil 5.4: Nessus tarama sonucu elde edilen en tehlikeli servis bilgisi detay port grafiği
Kaynak: Tez çalışması için kurulan Ossim Sistemi

Nessus taraması detay değerlendirmesi Ek 9'da görülmektedir.

5.2.2 Örüntü (Doku) Tarama (Pattern Detection)

Genel olarak girişim tespit sistemleri, bu girişimlere ait ipuçlarını işaret eden bir veritabanı kütüphanesi sahiptirler. (Signature veritabanı). Bu imzalar ağ üzerindeki trafiği kontrol ederek karşılaştırma yapılmasında kullanılır. Ağ trafiği ile bu veritabanındaki imza dokusunun eşleşmesi durumunda, olası bir atağın oluşmakta olduğu tespit edilebilir. Bu durumda bir alarm mekanizmasının çalıştırılması mümkündür. Ossim sunucusu içerisinde bir çok hazır yönerge (direktif) bulunmaktadır. Direktifler bu alarmların hangi durumlarda oluşacağını belirtmek üzere kullanılabilmektedir.

5.2.3 Snort IDS

Snort yine açık kaynak kodlu ve oldukça bilinen bir girişim tespit sistemidir. Snort dört tür modelde çalışacak biçimde konfigüre edilebilir:

5.2.3.1 Snort sniffer modu

Sniffer modu temel olarak network paketlerini okur ve sürekli bir yayın şeklinde konsol üzerinde bu paketlerin gösterimi gerçekleştirebilir. (Ekran üzerindeki görüntü)

5.2.3.2 Network paketlerini diske kaydedilmesi

Paketlerin bir logunun disk üzerine kaydedilmesi modelidir.

5.2.3.3 Ağ girişim tespit modu (Network Intrusion Detection System)

NIDS oldukça karışık, ancak konfigüre edilebilir yapıdadır. Kullanıcı tarafından oluşturulan, ön tanımlı kural setini kullanarak girişim tespit edebilen, izlediği bilgilere göre aksiyonlar gerçekleştirebilen moddur. Kısacası veri akışını analiz ederken, daha önce tanımlanmış kurallarla eşleştirme işlemi yapıp, yine ilgili kurallarda belirtilmiş aksiyonların uygulanmasını sağlayabilmektedir.

Tez çalışması sırasında burada önemli bir detay oluştuğuna karar verdik. Bir sunucu NDIS modunda, yani Network üzerinde geçen her paketi araştıran sniffer modunda çalışırken, sunucular log kayıtlarını tam olarak oluşturamama riski ile karşı karşıyadır. Bu durumda çok önemli bir veriyi kaybetme riski ile karşı karşıya kalınabilir. Bu nedenle NIDS modunu, kabloya gönderilen her bir paketi kaydetmeniz gerekmeyecek şekilde etkinleştirmek için, "snort.conf" içindeki kurallar dosyası linkini yapılandırmanız gerekir.

Snort tarafından yazılan log dosyaları, sunucuda mysql veritabanına ACID tarafından kaydedilir.

Ağ girişim tespit modu Ossim sunucusu tarafından da kullanılmaktadır.

5.2.3.4 Araya girerek yönlendirme modu, (Inline Mode)

Bu mod ağ trafiğine ait paketlerin iptables ya da IPFW üzerinden yönlendirme yapılarak snort ajanını devreye sokmasıyla, paketin değerlendirilerek yönlendirilmesi esasına dayanır. Paketler, yine daha önceden tanımlanmış kurallar (inline-specific rules) uyarınca geçirilebilir veya geçirilmesinin engellenir. (İzin verilmeyen bir erişim olduğuna karar verilen network paketi "drop" edilir). İlk amacı girişim tespit sistemi (IDS) olarak çalışan Snort, inline modunda çalıştırıldığı anda, girişim önleme sistemi (IPS) özelliği kazanmış olacaktır. Özetle bu mod sisteme sızma girişimlerini tespit etmekle kalmamakta, aynı zamanda bu girişimlere ait paketlerin drop edilmesine olanak sağlamaktadır.

Snort: http://www.snort.org

5.2.4 Anomali Tespiti (Anomaly Detection)

Anomali tespiti, yine bilgi sistemlerine istenmedik bir girişimin veya amacı dışında aksiyonun belirlenmesidir. Tespit sistem aktivitelerinin gözlenmesi, normal ve normal olmayan network trafiğinin sınıflandırılması yöntemi ile çalışmaktadır. IDS sistemlerinin kullandığı imza yapısından daha farklı olarak, normal network aktivitelerini ezberleyerek aksi durumları belirlemeye çalışır.

Pasif bir şekilde örüntü arayarak veri toplayan düşük seviyeli detektörler ve monitörler trafiği etkilememektedir.

5.2.4.1 Spade aracı

Snort girişim tespit sisteminin raporlama mekanizmasından yararlanarak, bir ön işlemci şeklinde çalışarak, umulmayan network paketleri hakkında alarm üreten bir yazılım bileşenidir. Hedef ve kullanılan portların bir değerlendirmesini yaparak, pek de alışık olunmayan bağlantıları tespit edebilme yeteneğine sahiptir.

Snort yapısı içerisine yerleştirilmiş, istatistiksel paket bozukluk saptama motoru (SPADE) bir SNORT için uyumlu bir ek olan istatistiksel anomaliyi saptama sistemidir. Gizli port taramalarının otomatik saptaması içinde kullanılır. SPADE, x saniye boyunca z kadar denemelere bakan geleneksel yaklaşımı kullanmak yerine, port taramalarını saptamak için bozukluk puanı kavramının kullanımını önerir. Network üzerinde mevcut

bir paketin “bozukluk puan”ını hesaplamak için basit bir frekans tabanlı bir yaklaşım kullanılır. Verilen paketin görünme zamanı ne kadar azalırsa bozuk olma oran puanı o kadar artar. Bir başka deyişle, geçmişte en son aktiviteye bağlı yabancılık derecesi olarak bir bozukluk oranı belirtilir.

Bir defa bozukluk puanı bir değeri geçince, paketler port taramalarını saptamak için dizayn edilen snort motoruna iletilir.

5.2.4.2 RRD aberrant-behaviour aracı

N-Top için yazılan bir eklenti olan önleyici davranış amaçlı RRD (Aberrant Behaviour plug-in), kullanım parametrelerini öğrenerek, tahmin edilmedik davranış biçimleri için alarm üretir.

Burada kullanılan yöntem, ağ ile ilgili parametreleri belli bir zaman serisi (zaman dizilimi) içerisinde, bir adım sonrasında tahmin ederek, gerçekleşen yani gözlenen parametreler ile tahmin ettiği parametreler arasındaki sapmayı ölçmektir.

Bu bilgi (alışılmadık) network parametreler hakkında bilgi sağlayarak, bütün bunlar, network üzerinde bir atağın varlığını gösterebilir.

5.3 ARPWATCH

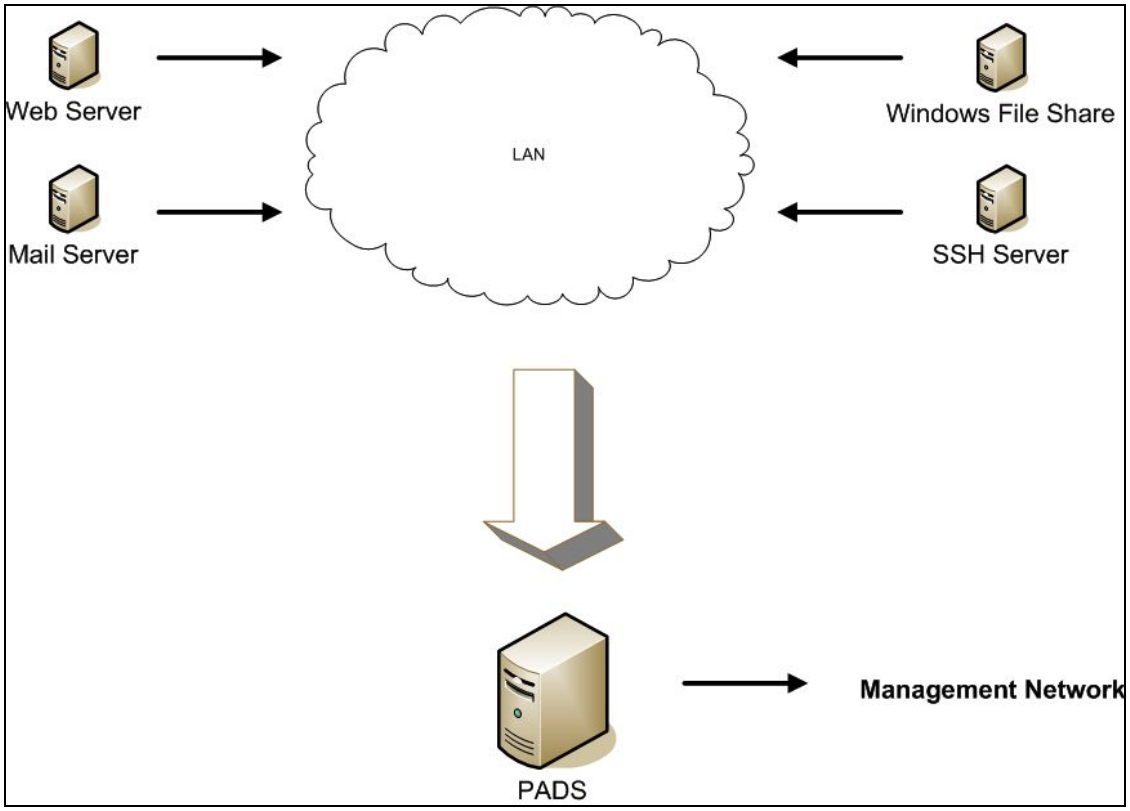
Arpwatch Adres Çözüm protokolü olan ARP aktiviteleri için, tüm ethernet trafiğini gözlemleyen bir araçtır. ARP aktiviteleri içerisinde gözlemlemek istenilebilecek en önemli aksiyon, ARP "Spoofing"’dir. ARP spoofing lokal network içerisinde, sahte ARP mesajları göndererek, network üzerinde ethernet trafik kurallarına göre aranmakta olan gerçek node yerine, başka bir node’un kendi IP adresi ile atak girişiminde bulunan girişimcinin MAC adresini birleştirerek, başkasının yerine geçme işlevi için kullanılan bir yöntemdir.

5.4 PADS

Pasif network servislerini yada cihazlarını tespit etme amaçlı bir sistemidir. (Passive Asset Detection System)

Varlık yönetimi, bilgi güvenliği ve dolayısı ile log değerlendirmeleri açısından önemli bir faktördür. Sistemde ağa bağlı tüm cihazların takibinin yapılması gerekebilmektedir. Nmap ve Nessus gibi etkin tarayıcılara rağmen, bazen de pasif bir şekilde ağ aygıtları belirlemek gereklidir. Pads ve Nmap araçları bir araya gelerek, o ana kadar rastlanmamış ve sonradan ortaya çıkmış olabilecek yeni network servislerini tespit edebilirler.

Çalışmamızda Pads'in pasif varlık belirleme sisteminden daha çok, pasif anomali tespit sistemi olarak davrandığı düşünülmüştür.



Şekil 5.5: Pasif Network Sistemlerinden Bilgi Toplama

5.5 P0f

p0f pasif işletim sistemi (Operating Systems) için bir ortam belirleyicidir. Fakat daha farklı amaçlara da hizmet edebilir. Örneğin ortamda bir NAT kullanımı (Network adress translation) olup olmadığı, bir yük dengeleyici (load balancer) çalışıp çalışmadığını, firewall varlığını, uzak sistemlere bağlantıları ve uzaktaki sistemlerin ne zamandır çalıştıklarını ve diğer node' lardaki ağları ve onların internet sağlayıcılarını da tespit edebilir. Bütün bu sistemler bir ateş duvarı ile korunuyor olsa da p0f ile tanımlanabilirler. P0f network üzerinde direk yada endirekt bir ilave trafik yaratmayacak kadar avantajlıdır. EK Ossim Ara yüzü P0f Rapor Çıktısı

p0f TCP/IP paketlerinde imza veritabanından yararlanarak eşleştirme yaparak, ilgili paketin başlık (header), windows size (bir seferde alınan veri büyüklüğü byte cinsinden) gibi bilgilerle işletim sistemini ortaya çıkarabilmektedir. EK P0f

Tablo 5.2: p0f aracı ile saptanan pasif varlık işletim sistemi tespitleri

OS (İşletim Sistemleri)	Total
Windows Server 2003	13
Windows 2000 Advanced Server	4
Windows 9x	4
Debian Linux	3
Windows 2000 – XP	3
FreebSD 5.0 RELEASE (x86)	3
Linux 2.4.xx	2
Windows 2000	2
Windows 2000 Professional	2
Windows 98 / 2000	2
Linux 2.4.22-gentoo-r5	1
Windows 2000 Running IIS Version 5	1
Windows 2003	1
Windows 98 Second Edition	1
Linux.2.4.20-web100	1

5.6 NETWORK GÖZLEMLEME (MONİTORİNG)

Network üzerinde log toplarken ve/veya ağınızı monitör ederken, anlamlı (başarılabilir) ve iyi seçilmiş noktalar belirlenmelidir.

5.6.1 Network Gözlem Amaçları

Network gözlemleri pasif bir aksiyondur. Belli bir bağlantı noktası (network geçiş noktası) üzerinden geçen tüm network trafiğini kontrol etmek network monitoring tanımını ifade eder. Örneğin bir kurumun internet bağlantısını monitör ederek, network oturumları (sessions) hakkında bilgi edinirken, ağdan dışarı çıkan ve içeri gelen paketleri de izlemek mümkündür.

Tez çalışmasında bununla ilgili bir kaç test gerçekleştirilmiştir. Ossim sunucusunun performansına göre uygun olabilecek network trafik miktarını belirlerken gerçekleştirilen testlerde, tüm network trafiğinin dinlenmesi gereken noktalarda en az 8CPU ve 2 gbps bant genişliğini sağlayabilecek network kartlarına sahip ve bu işe özel olarak ayrılmış "dedicated" tek bir sunucu olması gerektiği, özel olarak log toplanacak sunucular için yine bazı segmentasyonların gerçekleştirilmesi gerektiği belirlenmiştir. Bu konuda oluşan kısıtlar için yaptığımız denemeler sonucunda;

1- Proxy kullanıyorsanız buna "dedicated" edilmiş Ossim Sunucusu gerekmektedir
2- Windows sunucular için yaklaşık 10'ar sunucudan oluşan her gruba bir Ossim sunucu atanmalıdır.
3- VPN, DHCP, DNS gibi sunuculardan oluşan bir grup için sunucu ayrı olabilir,
4- Log toplayacağınız ve plug-in yazılmış sunucular için alınacak log verisinin sıklığına ve büyüklüğüne göre, veri alma yönteminize özel olarak sunucu sayısı belirlenmelidir.

5.6.2 Ntop

Ntop network kullanım bilgilerini göstermek üzere, network trafiğini araştıran açık kaynak kodlu bir araçtır. Komut satırından çalışır. Bir çok işletim sistemi ortamında çalışabilir. Ağın kullanımı hakkında Ntop kullanıcıları bir web ara yüzünden izleme yapabilirler. Ntop bu özelliği ile basit bir RMON ajanı görevi görmektedir.
http://www.ntop.org/

Host	Domain	Data			Packets		
		Current	Avg	Peak	Current	Avg	Peak
all-systems.mcast.net		0.0 bit/s	0.6 bit/s	4.9 bit/s	0.0 Pkt/s	0.0 Pkt/s	0.0 Pkt/s
anadolu-ersun		0.0 bit/s	5.3 Kbit/s	46.8 Kbit/s	0.0 Pkt/s	1.7 Pkt/s	14.1 Pkt/s
dsl.static8121519283.ttnet.net.tr		0.0 bit/s	5.3 Kbit/s	46.3 Kbit/s	0.0 Pkt/s	1.6 Pkt/s	13.6 Pkt/s
192.168.1.1		0.0 bit/s	1.1 bit/s	6.6 bit/s	0.0 Pkt/s	0.0 Pkt/s	0.0 Pkt/s
192.168.1.1		0.0 bit/s	34.2 bit/s	299.2 bit/s	0.0 Pkt/s	0.0 Pkt/s	0.3 Pkt/s
212.156.13.28		0.0 bit/s	9.2 bit/s	81.2 bit/s	0.0 Pkt/s	0.0 Pkt/s	0.1 Pkt/s
224.0.0.2		0.0 bit/s	0.6 bit/s	5.4 bit/s	0.0 Pkt/s	0.0 Pkt/s	0.0 Pkt/s
239.255.255.250		0.0 bit/s	6.1 bit/s	54.1 bit/s	0.0 Pkt/s	0.0 Pkt/s	0.1 Pkt/s
00:18:DE:A4:1E:01		0.0 bit/s	1.1 bit/s	6.6 bit/s	0.0 Pkt/s	0.0 Pkt/s	0.0 Pkt/s

Şekil 5.6 Ossim raporlarında Ntop ara yüzünden alınan network verimlilik değerleri (Network Throughput: All Hosts - Data Sent+Received)

Total	392.1 KBytes [999 Pkts]	
IP Traffic	391.8 KBytes [993 Pkts]	
Fragmented IP Traffic	0 [0.0%]	
Non IP Traffic	252	
Average TTL	110	
TTL <= 32	0.9%	9
32 < TTL <= 64	1.2%	12
64 < TTL <= 96	0.0%	0
96 < TTL <= 128	97.3%	972
128 < TTL <= 160	0.0%	0
160 < TTL <= 192	0.0%	0
192 < TTL <= 224	0.0%	0
224 < TTL <= 256	0.0%	0

Şekil 5.7: Ossim raporlarında Ntop ara yüzünden alınan bir network swicth cihazı için trafik raporu

5.6.3 Nagios

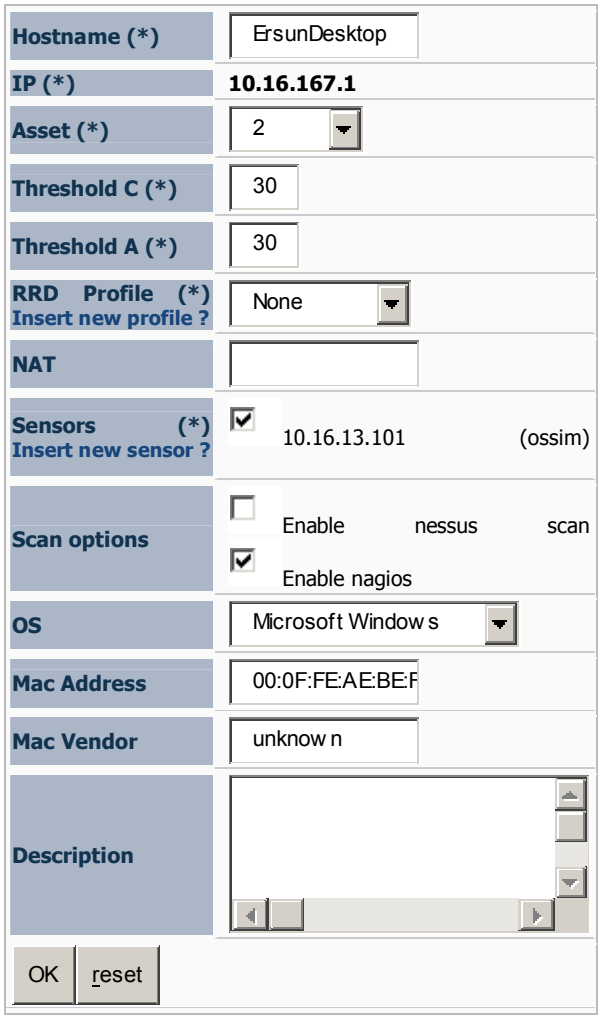
Nagios client (istemci), kişisel bilgisayar, sunucu veya sunucu üzerindeki servislerin gözlenmesi için dizayn edilmiş olan, log yöneticisine kullanıcıların farkına varmasının daha öncesinde, ağ üzerinde olan bir problemi haber verecek şekilde davranış yeteneğine sahip bir izleme aracıdır. Bu izleme servisi (monitoring daemon), belirtilebilecek harici eklentilerle Nagios'a durum bilgisini döner. Bir problem oluştuğu durumda çalışan Nagios servisi, çeşitli yollar ile (email, hızlı mesaj, short message service) bir mesaj göndererek, konfigürasyon bölümünde belirlemiş olduğunuz yöneticilere durum bildirir. Ayrıca Nagios tarafından izlenen sistemlere ait o andaki durum bilgisi, tarihsel log verileri ve raporlara bir web browser üzerinden erişilebilmektedir. Nagios da Ossim sistemi içerisinde yer almaktadır. Tez çalışması sırasında kullandığımız sürümünde GUI ara yüzü Ossim'de yer almamakta olduğundan, bir sonraki versiyon beklenmiş ve bu versiyona ilk olarak Nisan 2009'da ulaşabilmiş durumdayız. Nagios'un kendi başına çalıştığı ortamda ve/veya Ossim bütününde çalışması sırasında performans açısından çok iyi konfigüre edilmesi gerektiği anlaşılmıştır. Bu nedenle sürekli olarak Nagios özelliğini aktif etmemiş durumdayız.

Nagios ile ilgili konfigürasyonları Ossim dışında çalışan versiyonlarda web ara yüzünden harici araçlar ile yapmak mümkündür. Ossim kullanımı dahilinde ise yapılacak konfigürasyonlar aşağıdaki gibidir. Bu bilgilere yalnızca tez çalışması sırasında ulaşabildik. Ossim kullanıcı ara yüzünden, configuration tab' ına ve Host Scan bölümüne erişerek, bir nagios client ayarlaması yapılabilmektedir.

Tablo 5.5: Sistem tarama konfigürasyon örneği (Ntop)

Host	Plugin id	Plugin sid	Action
10.16.167.1	nagios (2007)	ANY	Delete
10.16.2.107	nessus (3001)	ANY	Delete
10.16.2.107	nagios (2007)	ANY	Delete
10.116.13.61	nessus (3001)	ANY	Delete
10.116.13.61	nagios (2007)	ANY	Delete
New			

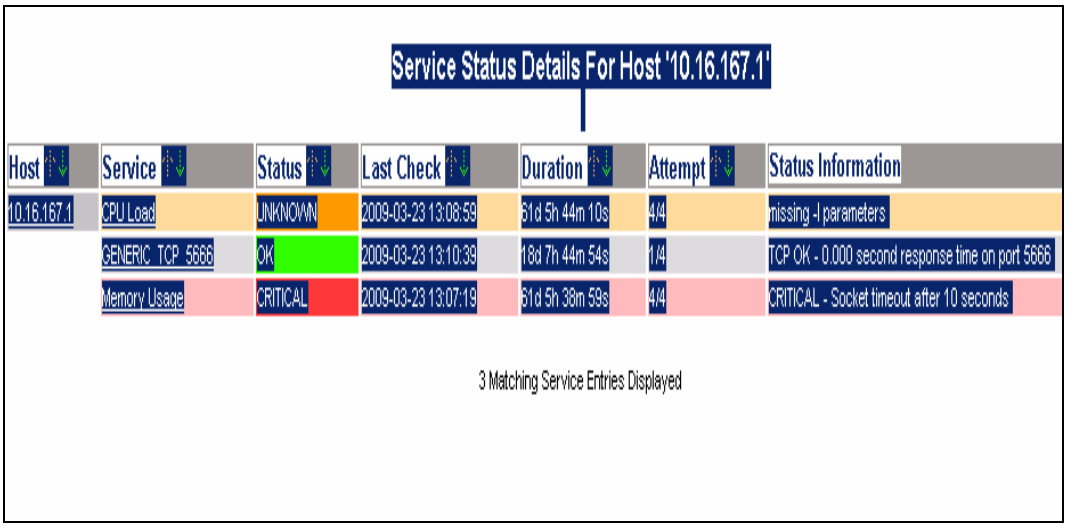
Ossim Nagios arayüzünden bir host tanımlama işlemi Şekil 5.7'de gösterildiği gibi örneklenmiştir.



Şekil 5.8: Ossim Nagios Host Tanımlaması (* Zorunlu olarak girilen alanlar)

Nagios bir host üzerinde çalışan SNMP servisi veya kendi ajanı ile servis durumu/kapasite/performans bilgisi alabilir. (Yukarıdaki örnekte bir desktop Ossim ara yüzünden tanımlanmıştır.)

Host olarak isimlendirdiğimiz network üzerindeki bir node olan kişisel bilgisayara NS Client ++ yazılımı kurularak, Ossim'de "monitors" bölümünden erişilebilirlik ("avaliability") izlemesi yapılabilir.



Şekil 5.9: Ossim Nagios Host Monitoring (İzleme)

Nagios Ayarları – Host üzerinde Nsclient dizininde "Programfiles/Nsclient/Nsc.Ini" dosyasında, Nagios ile ilgili ayarlamaları yapmak gerekmektedir. Nagios konfigürasyonlarının hem Ossim Sunucusu hem de client (istemci veya host) tarafında dikkatli bir biçimde ayarlanması gerekir. Yanlış konfigürasyonlar network üzerinde yoğun paket alışverişine sebep olabilirler. Client ve sunucu konfigürasyonlar EK XX de örneklenmiştir.

http://www.nagios.org/

NAGIOS ile yapılabilecekler;

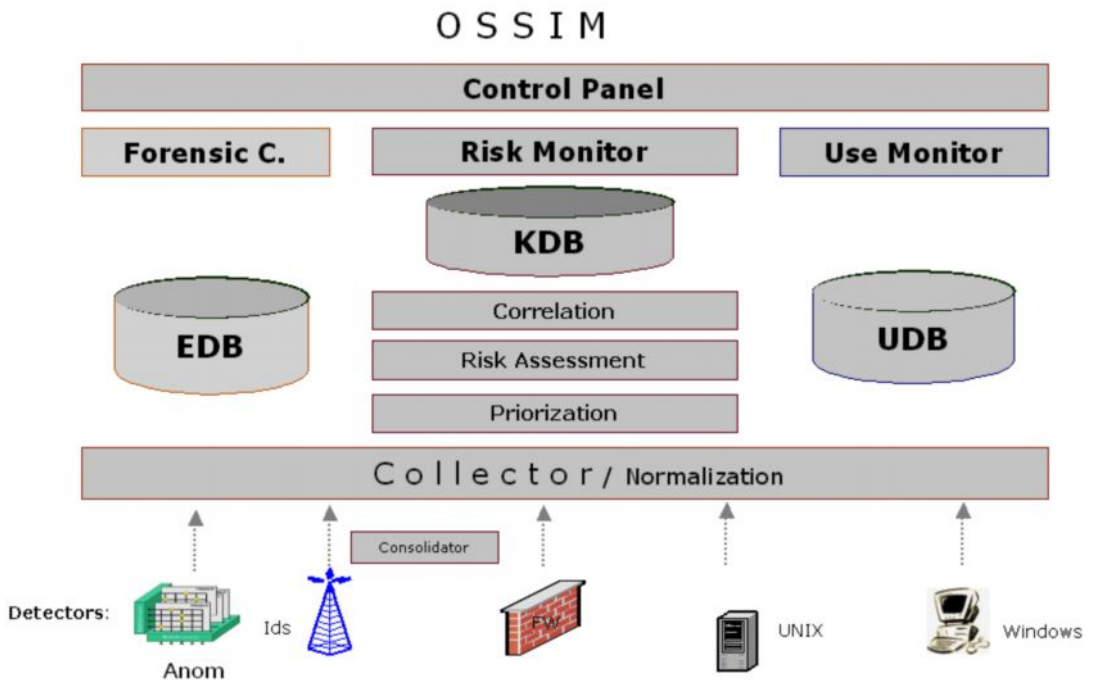
- Makine kaynakları denetleneme (disk kullanımı vs)
- Network servisleri denetleneme (SMTP, POP3, HTTP, NNTP, PING, vs)
- Otomatik log döndürme (log rotation)
- Belirtilen servisler yada makineler üzerinde belirtilen durumlarda kullanıcıya çeşitli
- Yöntemlerle uyarı verme
- Mail , SMS , Telefon , ICQ vb yöntemlerle uyarı verebilme
- Web ara yüzünden makinelerin, servislerin, logların ayrıntılı takibi olarak sıralanabilir.

6. OSSIM SUNUCU KONFİGÜRASYONLARI

Ossim sunucu sistemi korelasyon ve risk yönetim modülleri önceliklendirme ihtiyaçlarına göre konfigüre edilebilmektedir.

6.1 OSSİM KONFİGÜRASYON BİLEŞENLERİ

Ossim sunucu sistemi, çeşitli ajanlar aracılığı ile topladığı ve belli bir formata getirilmiş log bilgilerini işleyen bir sistem yapısına sahiptir. Bu sistemin log yönetimi açısından önceliklendirme, karşılaştırma ve risk değerlendirme prosesleri, Ossim tarafından sağlanabilen diğer önemli fonksiyonlarıdır. Bu amaçlarla toplanan bütün veriler kolektör vasıtası ile veritabanı içerisine depolanırlar.



Şekil 6.1: Ossim Konfigürasyon Şeması

6.2 ÖNCELİKLENDİRME (PRIORITISATION)

Önceliklendirmenin tanımladığı durum, eğer bir atak başarılı bir atak ise bunun önem yani kritiklik derecesi olarak açıklanabilir. Verilecek önceliklendirme (Priority) değeri atak olduğunda verilebileceği zararın büyüklüğünü gösterecektir. Öncelik değeri 0 ve 5 arasındaki bir değer olarak belirlenebilmektedir.

Ossim sunucusu tarafından alınmış veriler, Ossim ajanları tarafından er zaman için bir ID (Tanım numarası) ve SID (Güvenlik tanım numarası) değerleri verilerek normalize edilmişlerdir. Bu değerler snort tespit sistemi tarafından tespit edilmiş veya bir ping isteğinin tespiti ile verilmiş olay numaralarını işaret edebilir. Bir öncelik bir olay için verilmiş olacaktır. Önceliklendirme bilgisi Ossim veritabanı içerisinde bulunabilir ve yöneticinin ihtiyaçlarına göre ayarlanabilir. Bir olay için öncelik değeri atamanın bir diğer yolu bir politika (ilke) oluşturmaktır (Policy). Bir politika bir network trafiğini temsil edebilir. Bir politikaya uyan bir durum tespit edildiğinde, Ossim veritabanında önceliklendirmeye göre verinin normalize edilmesi yani, önemsiz olan veri yerine öncelikli olanın üzerine yazılması gibi bir normalizasyon yapılabilir. Aslında bu durumu bu şekilde ayarlamak, tüm verilere sahip olabilme anlamında bir risk oluşturabilir. Ancak alınan verinin hızla büyüdüğü ortamlarda, bu opsiyonu kullanmak zorunluluğunda kalınabilmektedir.

Örneğin, bir sunucuya karşı yapıldığı tespit edilen port taraması tipindeki bir girişim, bir yazıcıya yapılan girişimden daha önemli olabilir. Normalleştirme tablolarında varsayılan değerleri yöneticiler her olay için değiştirebilmektedir.

Tüm bunlarla birlikte öncelik veya tehditlerin ağ topolojisine uygun şekilde yerleştirildiği bir Politika Paneli bulunmakla birlikte bu panel mesela harici saldırılara içsel olanlardan daha yüksek değerler vermekte veya bilinen yanlış alarmlar için önceliği azaltmaktadır.

6.3 DATA TOPLAMA POLİTİKALARININ UYGULANMASI

Toplanacak olayların miktar ve tiplerini filtrelemek ve kontrol etmek için her sensor üzerinde bir Öncelik İlkesi ve korelâsyon direktifleri kurmak mümkündür.

Örnek olarak çok sayıda benzer olayı birleştirebilir ve tek bir biçimde gönderebiliriz.
Aynı zamanda sensor seviyesindeki ani riskleri değerlendirebilir ve sadece önemli olayları toplayabiliriz.

Önceliklendirme Politikaları tek bir ana sistemden uygulanabilmekte ve bu sayede de

sensorun veritabanı erişimi gerekli olmamakta ve genel Önceliklendirme İlkesi de tüm sensorler için merkezi bir server üzerinden kullanılabilmektedir.

6.4 KORELASYON (CORRELATION)

Sistemler veya network üzerinde girişim olarak anlaşılmaması gerektiği halde, bir atak zannedilebilecek girişimlerin (false positive) sayısının azaltılması açısından korelasyon önemli bir özelliktir. Korelasyonun kısa bir tanımını şu şekilde yapabiliriz: Bir olayın nasıl gerçekleşebileceğine karar vermiş olabilmek. Bu yaklaşımı sağlayabilmek üzere, gerçekleme değerine 0 ve 10 arasında bir değer verilmektedir.

Korelasyon üç farklı şekilde yapılmaktadır:

- Farklı olayların korelâsyonu (Mantıksal Korelasyon)
- Olay ve güvenlik açıklarının korelâsyonu (Çapraz Korelasyon), ve
- Olay ve işletim sistemlerinin – hizmetlerin korelâsyonu (Envanter Korelâsyonu)

Korelasyon her olayı raporlamadan önce bunların her birini kontrol ederek, 24 saatte milyonlarca olan olay sayısını bir düzinenin altına indirebilmektedir. Sistem yöneticisi tarafından bu sonuç daha iyi okunabilir. Korelasyon sistemi, soyut ve kompleks alarmları raporlayabilmesi sayesinde ayrıca Ossim Yönetici konsoluna bilgi vermektedir.

OSSIM'in mantıksal korelasyon motorunda yaklaşım her olayı kontrol etmek üzerine kurulmuştur. Bir günde milyonlarca olay kaydı oluşmasından dolayı, kontroller tamamlanıncaya kadar bu kayıtları değerlendirmek konusunda zorluk çekilmesi son derece açık bir sonuçtur. Bu nedenle olaylara yönelik olarak korelâsyon motoru da, bir saldırının gerçek mi yoksa yanlış alarm mı olduğunu belirlemek için delil veya semptomlar aramaktadır.

6.4.1 Mantıksal Korelasyon

Mantıksal Korelasyonun esas amacı bir güvenlik olayının gerçek mi yoksa yanlış alarm mı olduğunu belirlemek için deliller aramaktır. Bu, günümüzün güvenlik

sistemlerindeki esas konudur. Bir gün boyunca neredeyse çoğu yanlış alarm olan çok sayıda olayla karsılaşabiliriz. Bir saldırının gerçekten var olduğunu kontrol etmek için, bu kayıt denizinde otomatik çalışan süreçlere ihtiyaç olacaktır.

OSSIM Mantıksal Korelâsyon motoru şunları sunmaktadır:

- Hem detektörlerden örüntü girişini hem de izleyicilerden gelen gösterge girişini beraber kabul eden karma kaynak yapısı;
- Belirlediğimiz Korelasyon Direktifleri çerçevesinde olay çıktılarını ilişkilendirerek tekrarlanabilir sistemlerin analizi,
- "N" seviyedeki korelâsyon yapısını dağıtılmış bir topoloji dahilinde tanımlayabilmemiz dolayısıyla hiyerarşik dağıtımlı bir mimari analizi;
- Her direktif aşaması için esnek nesne yönelimli ve zaman aralığı tanımları;

Mantıksal Korelasyon yine bir mantıksal koşul düğümlerini ağacını uygulayan Korelasyon Direktifleri tarafından uygulanmaktadır. Bu yapı türü aynı zamanda bir "VE"/"VEYA" ağacı olarak da bilinmekte olup genellikle yapay zekâ sistemlerinde kullanılmaktadır. Bir düğüm koşulu örtüştüğünde korelasyon motoru ilkin çocuklara yönelecektir, örtüşmez ise de bir sonraki "ağabey"e yönelecektir (aynı babaya sahip yasayan olaya). Bu Y ekseni üzerinde "VE" işlemini gerçekleştirmekte veya X ekseni üzerinde "VEYA" işlemini gerçekleştirmektedir (Daha sonra verilecek bir örnek bunu daha iyi anlamımızı sağlayacaktır).

Korelâsyon motoru koşullarla örtüşen düğümler boyunca ilerledikçe güvenilirlik değişkeni de büyümektedir: daha çok delil elde ettikçe, saldırı olduğu yönündeki olasılıklar da artmaktadır.

Örneğin bir port tarama girişimi yapılmış bir ağ üzerinde, bu işlemin sonrasında çok sayıda event tespit edildiği bir durumda, bu eventlar belirli bir ağ üzerinden geldiğinin anlaşılması durumunda, bu tehlikeli bir atak olarak tanımlanabilir. Bu durum bazı kuralar tarafından belirlenmiş olmalıdır. Bu eventlar, belli bir gerçekleme ve önceliklendirme değerine sahip olup, bu şekle uyan bir durum ise, buna ait bir alarm aksiyonunun gerçekleştirilmesi gerekmektedir.

Kurallar bir XML tablo içerinde belirtilmiştir. Bu tabloların kendiniz tarafından bir xml sheet içerisinde kalacak şekilde düzenlenmesi olanağı bulunmaktadır. Mevcut bir kural setini istediğiniz bir biçimde değiştirmeniz mümkündür.

Her bir kural seti direktif (directive) olarak adlandırılmıştır. Her direktif özel nitelikli yeni bir olay türü tanımlamakta (bu da ismini Direktiften almaktadır) ve özel bir önceliğe sahip olmaktadır; zira bunlar çoğu zaman kendilerini tetikleyen olaylardan daha geniş örüntüler tanımlamaktadırlar. Bu yeni olay bir yandan herhangi bir harici ajan tarafından gelmiş gibi toplayıcıya gönderilecek olan normal bir OSSIM olayını teşkil etse de diğer yandan farklı korelâsyon seviyelerinin uygulanabileceği yeni bir yol yaratacaktır.

Bir direktif bir olayın ardışıklığı-sıralanması olarak da değerlendirilebilir. Zira bir alarm oluşmaşı durumunda, aslında bir ID, SID ve gerçeklenme ile olayın ard arda örtüşmesi söz konusudur. Bu olay veritabanına da yazılmış olacaktır. Alarm oluşturan bu olay tekrar ederse gerçekleme değerini artıracak yönde değiştirmek gerekir. Bu da risk seviyesinin artması anlamına gelecektir. Risk seviyesi hakkında ilgili bölümde bilgi verilmiştir.

6.4.2 Çapraz Korelasyon

Çapraz Korelasyon ile Detektör veya Güvenlik Açıkları Tarayıcılarından gelen bilgileri süzmek suretiyle mağduru olduğumuzu (veya olmadığımızı) bildiğimiz olaylara öncelik verir veya bunların önceliklerini düşürür.

OSSIM Çapraz Korelâsyonu her detektör için Güvenlik Açıkları Veritabanlarına ve Detektör Çapraz Tablolarına ihtiyaç duymaktadır. OSSIM, OSVDB Güvenlik Açıkları Veritabanını kullanmakta ve su anki durum ile Snort IDS – Nessus Çapraz Korelâsyon Tablosunu içermektedir.

Çapraz korelasyonun bir olayın hedefi etkileyip etkilemediğine göre kontrol edilmesi ile false positive sonuçları azalttığını söylemek mümkündür.

Bu neden çapraz korelasyonun bir destination (hedef) IP adresi ile yapılan girişim olaylarında geçerli olacağını söylememiz pek de yanlış olmayacaktır. Bu hedef yani varış yerine ait IP adresi, adrese kullanan sistemde bir zayıflık olduğunu akla getirecektir. Bu sonuç, nessus gibi bir zayıflıklar konusunda bir signature veritabanına sahip bir sistem tarafından taranarak ortaya çıkarılmış bir sonuç olacaktır. Bu atağa karşı zayıflıkları olan bir sistem var ise, atağın artmasına yönelik gerçekleme değeri artacaktır.

Eğer snort bir Ip adresi için atak (girişim) olduğunu tespit etmiş ise ve bu IP adresi zayıflıkları olma ihtimali olan bir sistem ise, gerçekleme değeri en yüksek olan 10 değerine çekilmelidir.

Mac adreslerinin değişimi, İşletim sistemleri ile ilgili bir olay olduğu için ve bir varış noktası IP adresi içermeyeceğinden, çapraz korelasyon örneği olarak gösterilemez.

6.4.3 Envanter Korelasyonu

Gerçekleştirilen saldırılar her zaman için işletim Sistemlerine ve/veya hizmetlere yöneliktir.

Envanter Korelasyonu saldırı hedefi olan makinenin bu işletim sistemi ve/veya hizmeti kullanıp kullanmadığını kontrol etmektedir. Bunları kullanıyor olması durumunda riskin varlığından emin olabiliriz fakat kullanmıyor ise bu olayın bir yanlış alarm olduğu teyit edebiliriz.

Bu tip bir korelâsyon Envanterin doğruluk düzeyine bağlıdır ve OSSIM de Otomatik Envanter ve Envanter Yönetimi bölümlerinde açıklanan Envanter özelliklerine sahiptir. Bu sayede sonuç durumunu en üst düzey doğrulukla vermektedir.

Envanter korelayosnunu, işletim sistemi, port, bağlantı protokolü, servis ismi ve servis versiyonu karşılaştırmaları olarak da düşünebiliriz. Çapraz korelasyondan başlıca farkı, hedef (varış) noktası belli olan bir IP adresi içermemesidir.

Örnek bir korelasyon olarak, beklenmedik bağlantıların ölçüldüğü, solucan tespitleri (worm detected) gösterilebilir. Çeşitli korelasyon seviyeleri oluşturarak, bunları daha dikkat çekici örneğin "istila alarmları" olarak belirleyebiliriz.

6.5 OSSİM SİSTEMİNDE RİSKLERİN DEĞERLENDİRİLMESİ

Ossim sisteminde risk değerlendirme, potansiyel kaybı ve ana makine veya ağınızda bir kayıp oluşma ihtimalini ölçer. Risk değerlendirmenin üç evresi, öncelik değişiklikleri, C&A düzeylerinin güncellenmesi ve risklerin hesaplanmasıdır.

Öncelik değişiklikleri, 6.2 bölümünde anlatılmış olan Öncelik verme işlemiyle gerçekleştirilir.

6.5.1 Ossim Sunucusunda Tehditlerin Başarı Seviyesinin Risk Açısından Değerlendirilmesi

Ossim sisteminde riskin etkisini ölçmek üzere C&A düzeyleri yöntemi uygulanmaktadır. "C" sembolü müdahale düzeyi anlamındadır. Başarılı olmuş bir saldırının kanıtını sağlar. "A" sembolü ise, sistemin maruz kaldığı saldırı düzeyidir. Yürütülmekte olan saldırıların olasılığını gösterir. Tespit edilmeleri için, bu saldırıların başarılı olma zorunluluğu da yoktur.

Her iki değişkenin önemi şu şekilde belirtilebilir: A değişkeni olası bir saldırının başlatıldığını görmek için önemlidir. Çeşitli nedenlerle, UDP gibi daha az güvenilir protokollerin sıkça kullanılabildiği sistemlerde, çok sayıda saldırının başlatılmış olması normaldir. O halde, C veya müdahalenin olup olmadığına bakılması, çalışan uzmanı tehlikeli saldırılar hakkında daha güvenilir yorum yapma şansı oluşturabilecektir. C düzeyinin, yükselmesi, makinenin saldırıya maruz kaldığını ifade edeceğinden, denetlenmesi çok önemlidir. Bir makine bir güvenlik tarayıcısı, rasgele pasif portlarda bir hizmet vs. gösterdiğinde, tuhaf C&A düzeylerine sahip olmuş olabilir. Bunun anlamı, bu makinede C&A düzeylerinin iyi ayarlanması gerektiğidir.

C&A kuralları, bir sisteme aşağıdaki kurallara göre atanır:

- Sistem 1' den sistem 2' ye gerçekleştirilen tüm olası saldırılar sistem 2'nin A'sını (yaşanan saldırıların düzeyi) ve sistem 1'in C'sini (müdahale veya normalde bir bilgisayar korsanı tarafından gerçekleştirilen şüpheli etkinliklerin düzeyi) yükseltir.
- Eğer bir olay türünün bir hedefi yoksa, olaylar o makinede içsel olduğundan (MAC olayları, OS değişim olayları, ana makine ID'leri olayları...) yalnızca kaynak makinenin C düzeyini yükseltiriz.
- Olay bir ağa özgüyse, o ağın C&A düzeyi olayla aynı şekilde güncellenir.

6.5.2 Ossim Sisteminde Riskin Hesaplanması

OSSIM iki farklı risk değeri kullanır. İlk risk, kaynak adresine bağlı olarak ve diğeri de hedef adresine göre hesaplanır. Kaynak adresi riski C düzeyi ve hedef de A düzeyi için kullanılır.

Risk, öncelik, güvenilirlik ve kıymete bağlı olarak hesaplanır. Kullanılan formül:
Risk = (Öncelik * Güvenilirlik * Değer) / 25

Öncelik işleminden 0 ila 5 arasında değişen bir öncelik elde ederiz ve korelasyondan da 0 ile 10 arasında bir güvenilirlik elde edebiliriz.

Risk değeri, 0 ila 5 arasında değişen bir sayıdır. Makinenin önemini temsil eder. Örneğin, bir sunucu, bir yazıcıdan daha önemlidir o halde, yazıcının kıymet sayısı daha düşüktür. Kıymet değerleri, yönetici tarafından, iş haritasına ve kırılganlığa göre belirlenir.

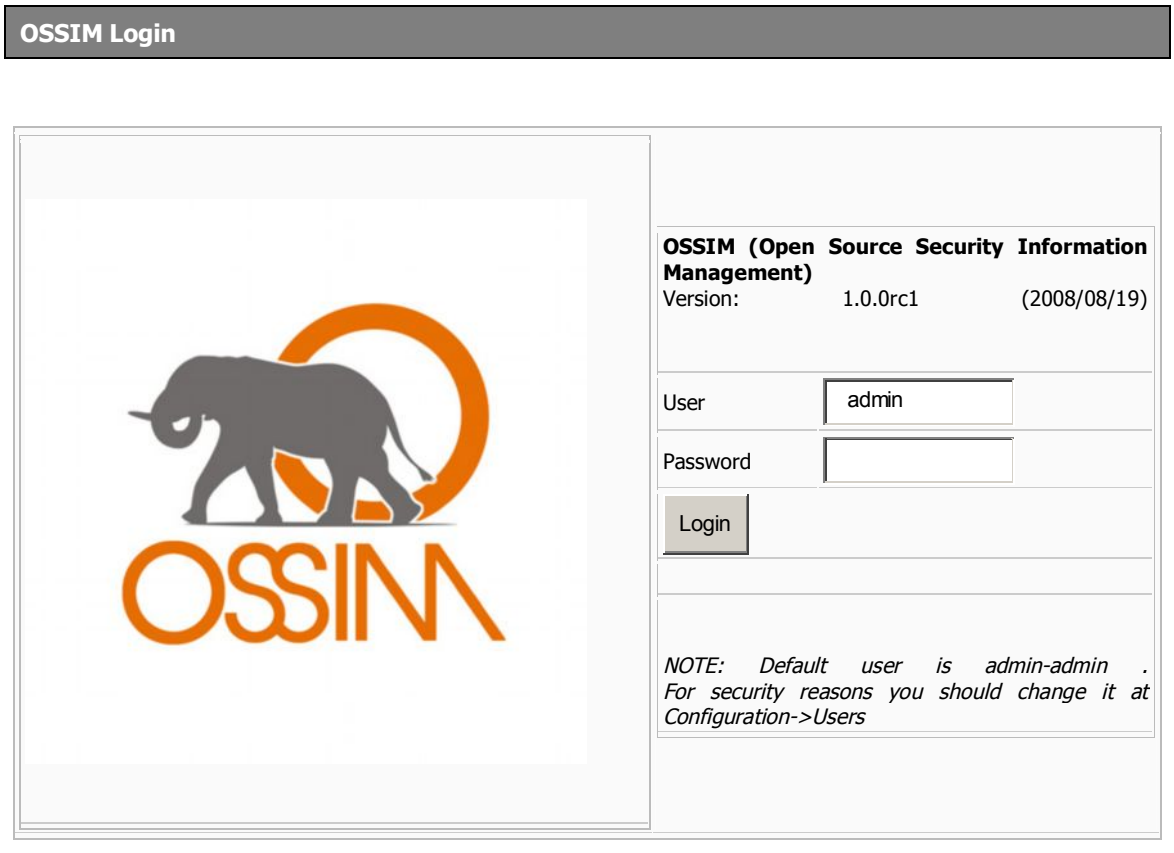
Risk değeri, 0 ila 10 arasında olabilir. Bu sayede, alarmları ve en tehlikeli olaylara veya saldırılara verilen yanıtı kolayca inceleyebiliriz. Ancak, her olay ve saldırının iki riski vardır. Riski en büyük olan olay, yöneticiye rapor edilir.

7. OSSIM ARAYÜZLERİ VE RAPORLAR

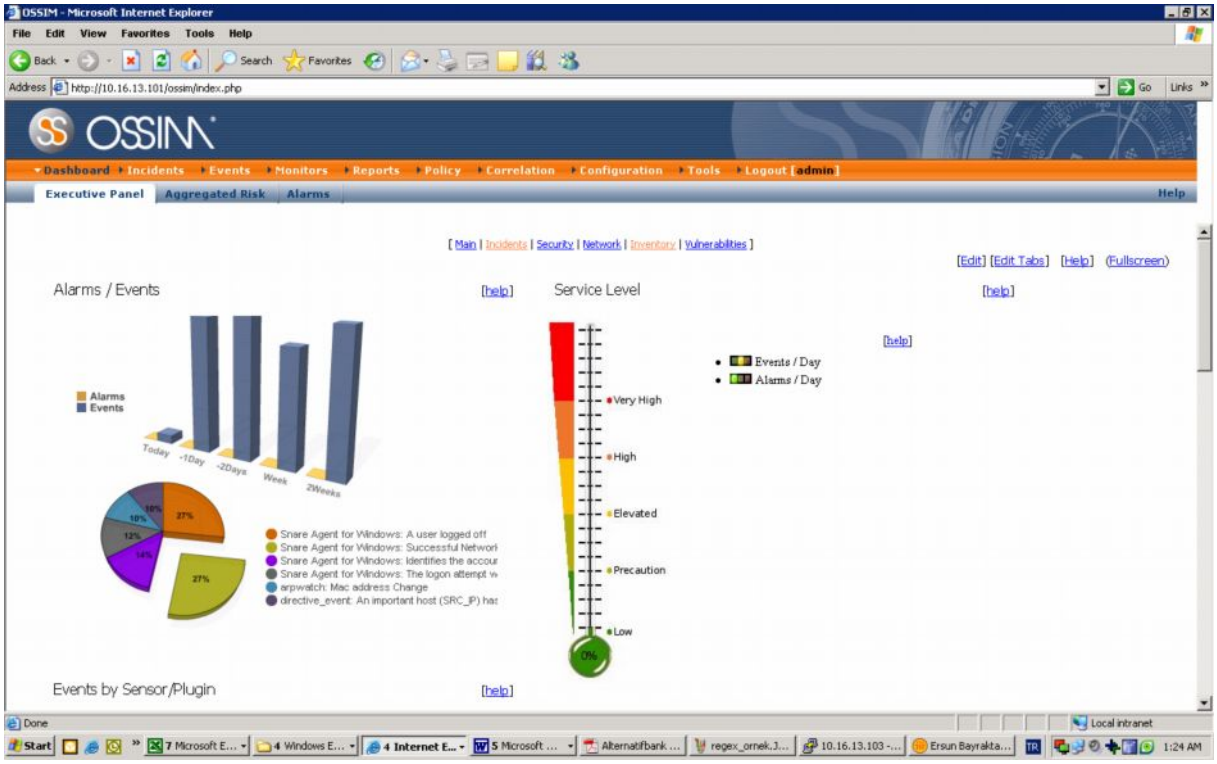
Logların izlenmesi, güvenlik yönetimi ve analiz amaçlı çok sayıda çıktının Ossim arayüzlerinde izlenmesi mümkündür ve bu ekranlar kişiselleştirilebilmektedir.

7.1 GÖSTERGE PANELİ (DASHBOARD)

Ossim'in kullanıcı grafik ara yüzünün ilk menü başlığı Dashboard'dur. Bu bölümde ölçüm metriklerinizi ve sorgulamalarınızı tanımlayabilirsiniz. Bu sorgular sizin güvenlik ve zayıflık raporlarınızı oluştururlar. Güvenlik ile ilgili verilerinize kolay bir erişim bu şekilde sağlanmış olabilir. Bu sorgulamalar ile güvenlik ve zayıflık konusunda raporlar oluşturulmaktadır. Son 10 günlük alarm bilgisi oluşturulabilir, ne miktarda zayıflık olduğunu görebilir ve ne kadar olay oluştuğunu kısaca gözlemleyebilirsiniz.



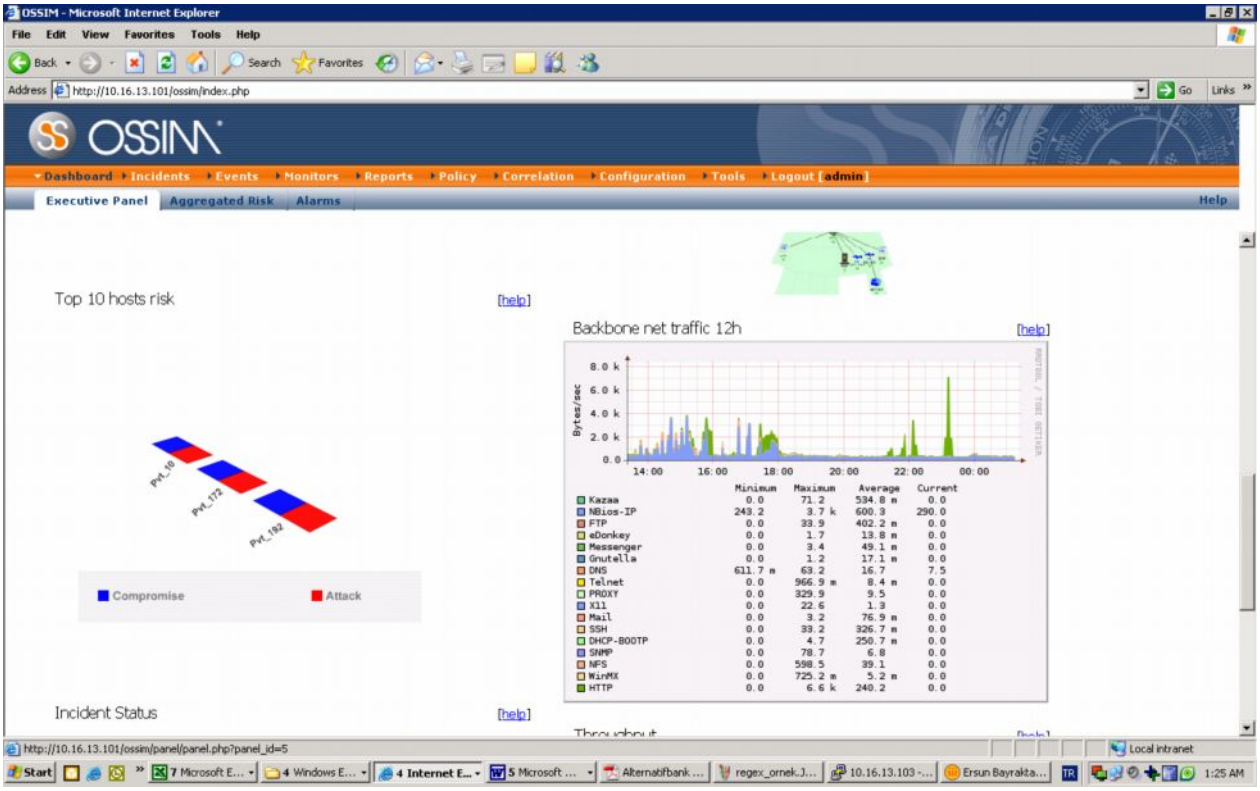
Şekil 7.1: Ossim Sistemi Web Tabanlı Giriş Ekranı



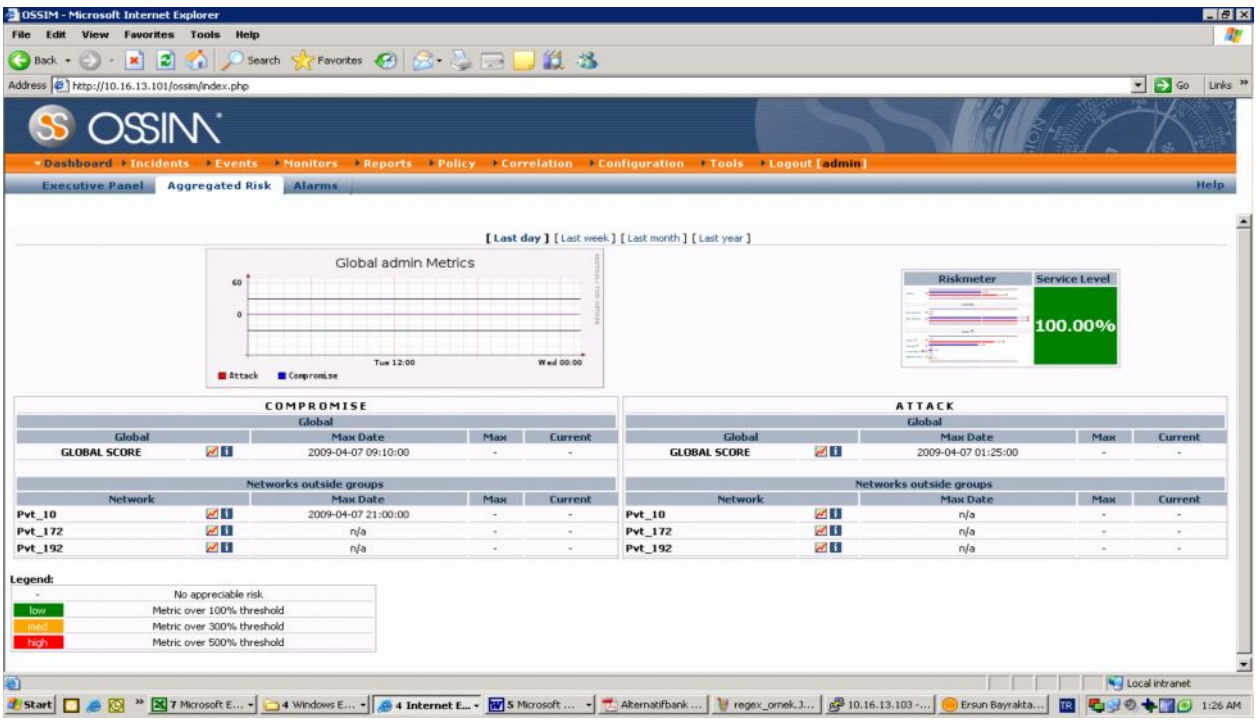
Şekil 7.2: Ossim Gösterge Paneli

7.2 METRİKLER

Ossim metrikleri risk seviyeleri durumunu (C&A) göstermektedir. Atakların grafiksel bir gösterimini ve etkili olup olmadıklarının anlaşılması için kolayca bir gözlem olanağı sunarlar. Bir sistem güvenlik/log yöneticisinin en önemli rollerinden biri bu ölçütleri gözlemlemesi olabilir. Metrikler, daha güvenli bir ortam oluşturulması yönünde önemli bir karar destek mekanizmasıdır.



Şekil 7.3: Risk seviyesi yüksek olan ve Ossim Sunucusu Tarafından Takip edilen sunuculara ait network trafiğine ait ölçüm ekranı

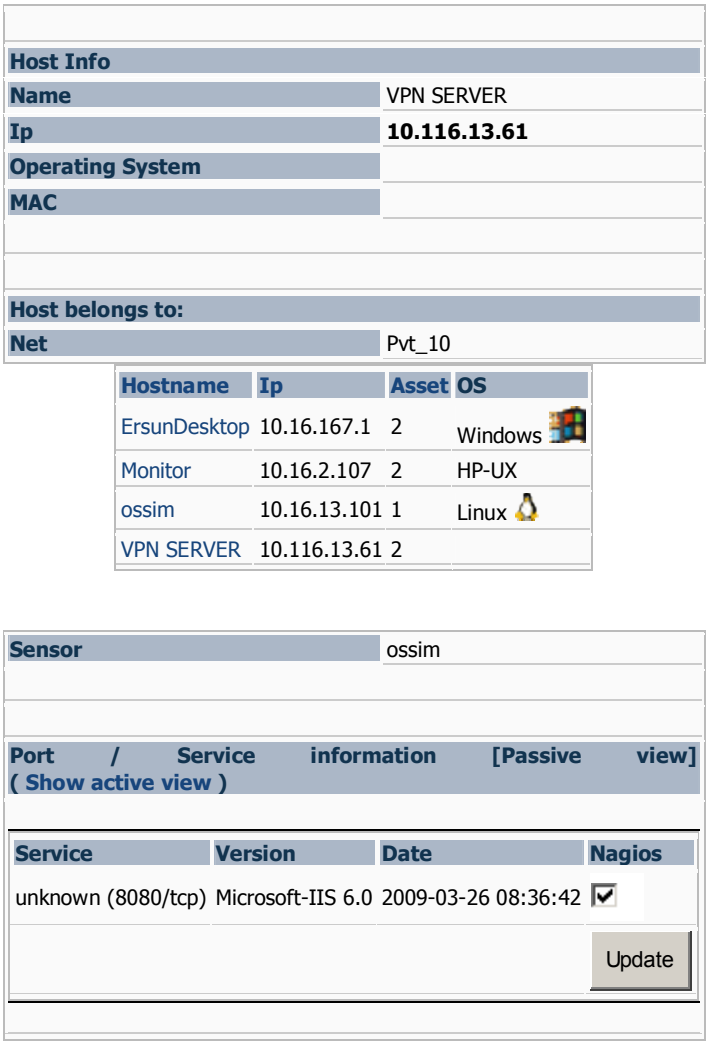


Şekil 7.4: Network trafik ölçümlerine ait detaylı risk raporu örneği

7.3 GÜVENLİK RAPORLARI (SECURİTY REPORTS)

Ossim tarafından ön ayarlamaları yapılmış güvenlik raporlarını bu ekranda sorgulamak mümkündür. Bilgisayar ağınızın güvenlik durumu ile ilgili bilgileri görebilirsiniz.

Bu grafik formatta zayıflıklar, anomaliler, network trendleri hakkında bilgiler ve host kaynaklarını görebiliriz.



Inventory - 10.116.13.61

Şekil 7.5: Güvenlik Raporu Örneği (Açılan bir Microsoft Web Servisi IIS için uyarı)

7.4 ZAYIFLIK RAPORLARI (VULNERABİLİTY REPORTS)

Zayıflık raporları Nessus aracı tarafından üretilirler. Bu raporlar oldukça önemli bir zayıflığın tespitinde ve yöneticilerin bu zayıflıklara karşın önlem almalarında önemli bir rol oynarlar. Nessus açık kaynak kodlu bir araç olduğu halde, bulundurduğu zayıflık veritabanlarını ticari yollar ile çok daha hızlı bir şekilde güncelleme olanakları sunmaktadır.

7.5 AVAİLABİLİTY REPORTS

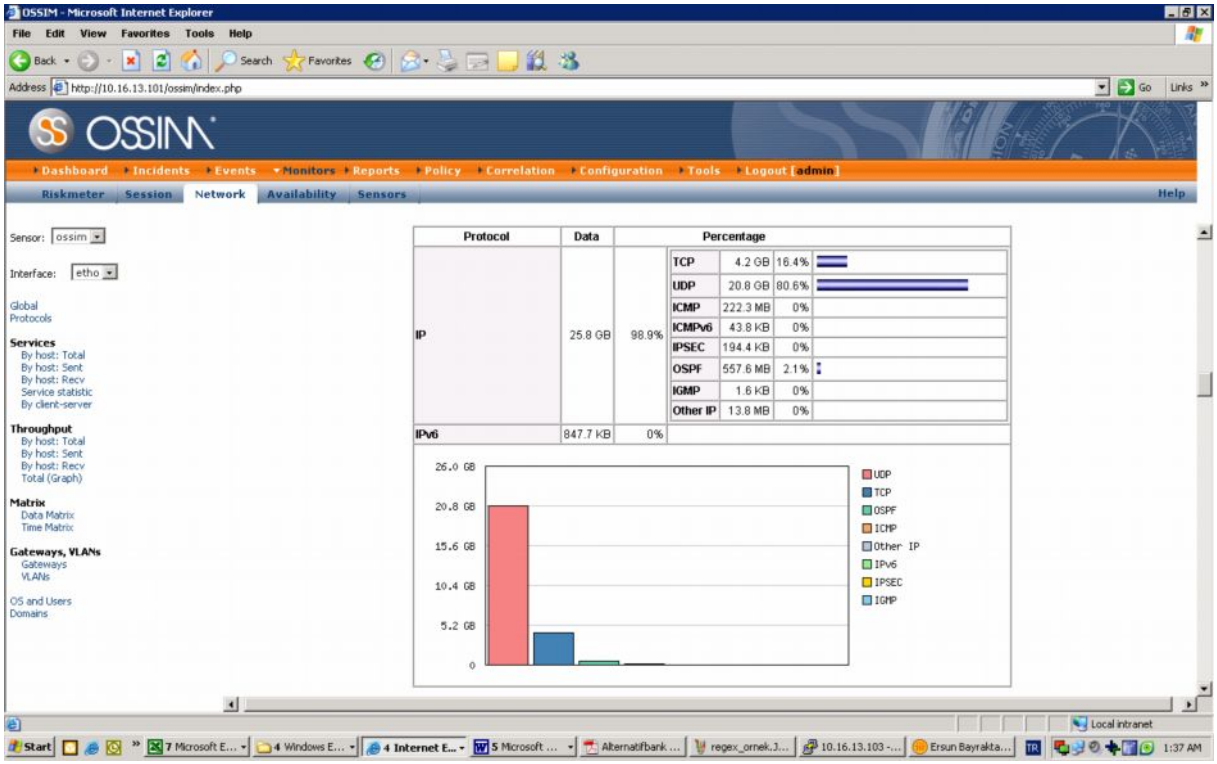
Nagios da aslında oldukça önemli bir güvenlik yönetim aracı olarak ortaya çıkmaktadır. Ossim bu aracı erişilebilirlik raporlarında kullanmaktadır. Zaten bu raporlar sayesinde kurgulama yapılan hostlar üzerinde çalışan servisler ve o hostlar için ayrıntılı bilgiler alınmaktadır..

7.6 AĞ KULLANIM RAPORU

Network raporu, network trafiğinizin anlık bir görüntüsünü sunmak üzere tasarlanmıştır. Buradaki bilgilerin bir çoğu Ntop aracı tarafından sağlanmıştır.

Remote Hosts Distance			
Network Load	**Actual**	25.5 Kbps	5.1 Pkts/sec
	Last Minute	5.5 Kbps	2.9 Pkts/sec
	Last 5 Minutes	7.8 Kbps	3.5 Pkts/sec
	Peak	10.6 Mbps	8053.4 Pkts/sec
	Average	36.2 Kbps	19.5 Pkts/sec
Historical Data	[ ]		

Şekil 7.6: Network Kullanım Raporu Detay



Şekil 7.7: Servis Bazında Network Kullanım Raporu Detay

7.7 FORENSİK ANALİZÖR (FORENSİC ANALYZER)

"Forensic" analizör oluşmuş olan olayların bilgisini gösterir. Bir alarm ile karşılaşıldığı takdirde, büyük bir ihtimal ile "forensic" analiz konsoluna yönelmeniz gerekebilecektir. Bu inceleme yapacak kişilere detaylı bilgi verebilecek bir araçtır. Bu bilgiler arasında olayın nerede oluştuğu, hedef ve kaynak bilgiler ile çeşitli bilgilere ulaşmak mümkündür.

7.7.1 ACID Girişim Veritabanı Analiz Konsolu (Analysis Console for Intrusion Databases).

"Forensic" (suça yönelik) analizlerin gerçekleştirilmesinde kullanılan ACID yazılıma ait grafik ara yüzü Ossim içerisinde yer almaktadır. PHP tabanlı bu analiz motoru veritabanındaki bilgileri analiz etmek üzere tasarlanmıştır. Bu bilgiler üzerinde proses ettiği anahtar bilgilere dikkat edecek olursak, ağınızdaki kesin bilgiler, zaman, sensor, imza veritabanına uyan durumlar, protokoller, IP adresleri, TCP/UDP Portları ve network sınıflandırmalarıdır. Bu durum OSI katmanları olan Layer-3 ve Layer-4 OSI katmanlarında da loglanmış olaylara ait bilgilerin sunulabileceğini veya değerlendirilebileceğini gösterir. EK 11 ve EK 12'de ACID "forensic" raporları görülmektedir..

ACID: http://acidlab.sourceforge.net/

7.7.2 Direktif Oluşturma (Directive Editör)

Ossim sisteminin en önemli özelliklerinden birisi, girişim tespitlerine göre imza veritabanından yararlanarak alarm üretilmesini veya aksiyon oluşturulmasını sağlayan direktif oluşturma sistemidir. Var olan direktiflere ait görüntü Şekil 7.8'de gösterilmektedir. Sisteme kendimize ait direktiflerin de girilmesinin sağlayabilmek üzere PHP kodları ile geliştirme yapılabilir. Bu konudaki araştırmaya devam edilmektedir.

Ossim sistemi tarafından oluşturulan direktif numaraları 1 ila 2999 arasında değişmektedir. Diğer direktifler özellikle Snort imza veritabanlarının kural setlerinde oluşturulmuştur. Kullanıcı tarafından oluşturulabilecek yeni direktifler için, direktif numarası 500000'den büyük olacak şekilde tanım yapmak mümkündür.

Directive Editor

Click on the left side to view a directive.
Click on the categories of directives to expand or collapse them.

Directive numbering

Category	Numbers
Generic ossim	1-2999
Attack correlation	3000-5999
Virus and Worms	6000-8999
Web attack correlation	9000-11999
DoS	12000-14999
Portscan/scan	15000-17999
Behaviour anomalies	18000-20999
Network abuse and error	21000-23999
Trojans	24000-26999
Miscellaneous	27000-34999
User contributed	500000+

Element of a directive

Şekil 7.8: Ossim Direktif Tanımları

8. LOG SUNUCUSUNA ROTASYON YAPILMASI

Log kayıtları, Ossim sunucusunda çalışan syslog sistemine aktarılabilmektedir.

8.1 SYSLOG SİSTEM LOGLARININ KULLANIMI

Linux ve Unix işletim sisteminin olay günlükleri syslog (RFC 3164, RFC 5424 ile tanımlanmıştır) servisi tarafından kaydedilmektedir. Ossim tarafında ise **syslog-ng** servisi aynı amaçla kullanılmaktadır. Forte (2004) Logların kaynak sistemlerden toplanarak Log Sunucusu olarak isimlendirilen sunucuya aktarılması işlemini Log Rotasyonu olarak adlandırmıştır. Zira kaynak makinedeki loglar her ne kadar konfigürasyonu yapılsa da, eninde sonunda eski logların üzerine yenilerinin yazılması ile sonuçlanacak bir disk alanı kısıtlaması ile karşı karşıyadır. Bu nedenle bir log sunucusuna bu aktarımın yapılması gerekir. Ancak log sunucusunun syslog dosyasına aktarılan çok sayıdaki log nedeniyle performans problemleri ile karşı karşıya gelmesi kaçınılmazdır. Syslog sistemi RFC 3164 kurallarında da belirtildiği gibi UDP protokolü ile iletişim kurar. Bunun nedeni TCP protokolündeki paket kontrollerinden arınmış olan çok daha hızlı UDP protokolünden faydalanmaktır. (* Ancak bu da paket kayıplarına neden olabilecek bir protokoldür.) Burada performans problemini oluşturan asıl problemin, çok sayıda kaynak sunucunun tek bir log sunucusuna rotasyon yapmasıdır.
Linux sistemlerde merkezi bir log sunucusunun kurulumu için aşağıdaki komut seti kullanılabilir.

 # /usr/sbin/syslogd –m 0 –r

Linux işletim sisteminin kurulumu ile birlikte gelen Syslog programının kullanımı yerine, Ossim sunucusunda da olduğu şekilde, syslog-ng uygulamasını (açık kaynak kodu) kullanarak, esnek filtreleme yetenekleri ile birlikte kontrollü bir protokol olan TCP protokolünü kullanmak mümkün olmuştur. Hatta bu noktada “stunnel” ve “ssh“ kullanılarak iki ağ arasında şifrelenmiş veri alışverişi de sağlanabilecektir.

Syslog-ng programı ile diğer bilgisayar sistemlerinden mesaj alacak şekilde bir konfigürasyonu aşağıdaki örnek komut işlemi ile sağlamak mümkündür:

Source r_src {tcp(ip(“192.168.1.3”) port(5140););

Tablo 8.1: Syslog Mesajları Örneği

Mar 1 06:25:43 server1 sshd[23170]: Accepted publickey for server2 from 172.30.128.115 port 21011 ssh2
Mar 1 07:16:42 server1 sshd[9326]: Accepted password for murugiah from 10.20.30.108 port 1070 ssh2
Mar 1 07:16:53 server1 sshd[22938]: reverse mapping checking getaddrinfo for ip10.165.nist.gov failed - POSSIBLE BREAKIN ATTEMPT!
Mar 1 07:26:28 server1 sshd[22572]: Accepted publickey for server2 from 172.30.128.115 port 30606 ssh2
Mar 1 07:28:33 server1 su: BAD SU kkent to root on /dev/ttyp2
Mar 1 07:28:41 server1 su: kkent to root on /dev/ttyp2

Syslog konfigürasyonları için Ek –XX Syslog konfigürasyon dosyası (Syslog-Ng.conf) incelenmelidir.

Ossim sunucusunun kurulu olduğu debian işletim sisteminde syslog kayıt deseni standardı aşağıdaki gibidir:

```
[syslog - datamining]
event_type=event
regexp="^(?P(\S+\s+\d+\s+\d\d:\d\d:\d\d)\s+(?P[^\s]+)\s+(?P[^\s]+)\s+(?P[^\[]*)\[(?P\d
+)\]:(?P.*))$"
sensor={resolv($sensor)}
date={normalize_date($1)}
plugin_sid=1
sensor={resolv($sensor)}
userdata1={md5sum($logline)}
userdata2={$logline}
userdata3={$generator}
userdata4={$logged_event}
userdata5={$pid}
```

8.2 PERFORMANS KRİTERLERİNE GÖRE LOG TOPLANMASINDA YÖNTEM SEÇİMİ

Ağ ortamında bir çok güvenlik cihazı zaten bulunabilmektedir. İşte bu nedenle OSSIM’e bir firewall sistemi entegre edilmesi oldukça kolay olduğu gibi, loglarını toplamayı düşündüğünüz önemli bir sistemin verilerini de değerlendirmeniz

mümkündür. Bu tezin en önemli amaçlarından biri mevcut ajanlar dışında, hedef olarak seçilen bir sistemden nasıl bir log alma mekanizması oluşturmak gerektiği ve bunun Ossim eklentileri ile syslog sistemine nasıl dahil olacağının yapılabileceğinin gösterilmesi veya bulunabilmesidir.

3. ve 4. bölümlerde belirtildiği gibi log verilerinin toplanmasına yönelik çalışmalarda, hem log sunucusunun aynı anda değerlendirebileceği network üzerindeki trafik (1 Gbps hızındaki bir network kartının takip edebileceği paket kapasitesi dikkate alınmalıdır) ve hem de diğer sistemlerden otomatik veya manüel olarak rotasyonla gelen log kayıtlarının miktarları düşünüldüğünde, özellikle büyük kurumlarda birden çok log sunucusunun kullanılması gerekmektedir.

Ossim sunucusunun varsayılan değerlerle topladığı event bilgilerinin yanı sıra, kurum ihtiyacına göre, performans kriterleri göz önüne alınarak, belli işlemler için birden çok log sunucusu kullanılmıştır. Tez çalışmasında çalışılan kurumda, Proxy ve firewall logları için bir sunucu kullanılırken, özellikle seçilen bazı sistemler için ayrı bir log sunucusu kurulmuştur. Bu sunucuda öncelikle denetim kapsamında önemli olabilecek iz kayıtlarının elde edileceği oracle kurumsal veritabanında, logların toplanmasına yönelik çalışmalar sonucunda, veritabanına direk erişim yetkisi olan kullanıcıların aktiviteleri ve veritabanı yöneticisinin aksiyonları log sunucuna aktarılmıştır. Log sunucusuna veritabanı sistem yöneticisinin aktiviteleri manüel bir yöntem ile aktarılırken, daha sonraki geliştirme ile direk olarak, veritabanı yöneticisinin herhangi bir değişikliğe izin vermeden Ossim sunucusu syslog-ng yapısına loglar eş zamanlı olarak aşağıdaki konfigürasyonlar ile aktarılmıştır. (Tez çalışmasında, yapılan kapsam araştırmasında loglarının alınması kararlaştırılan veritabanı yöneticisinin aktiviteleri için crone tab olarak isimlendirilen (batch process) süreç ile text bir dosyadan Ossim log sunucusuna aktarımı sağlanmıştır. Örnek: /bin/cat /var/log/paylas/LOG-`date +\%Y\%m\%d`.txt > /var/log/hyperion.txt. İlk aşamada bu metod ile bu aktivitelerin loglarını almaktayken, bu yığın işin çalışması anında, diğer sistemlerden gelen loglar ve bu yığın veriler nedeniyle performans sıkıntısı yaşanmıştır. Daha sonra yapılan bir çalışma ile veritabanı yöneticisinin (DBA, SYSDBA) aktivitelerini online bir şekilde log sunucusuna aktarımı direk olarak SYSLOG'a göndermek şeklinde başarılmış, böylelikle hem performans darboğazının hem de verilerin değiştirilme ihtimalinin de önüne geçilmiştir).

Kod 8.1: Oracle veritabanı üzerinden, kurumda veritabanına direk erişen kullanıcıların logunun oluşturulması (Bu politika, tetiklenen aksiyon (Trigger) yöntemi ile alınabilecek şekle getirilmiştir).

Oracle tarafından özel aksiyonların denetim izlerinin (logların) tutulması işlemine verilen isim - Fine Grained Auditing (FGA)' dir.

Oracle tarafından alınan standart loglama üzerine INSERT, UPDATE ve/veya DELETE işlemlerin logunu alacak şekilde, kullanıcı nesnesi için politika tanımlanmıştır.

Aşağıda bir veritabanı nesnesinde, ilgili kolon ve oracle sql ifadesi bazında hangi bilgilerin logunu oluşturmak istiyorsak, bunun ile ilgili veritabanı politikası yönetim seviyesinde tanıtılmıştır. Tez çalışmaında **EK 2'de** örnek çıktının SYSLOG üzerinden Log Sunucusu veritabanına aktarılan verileri gösterilmiştir.

Örnek: Oracle 10g DBMS_FGA.ADD_POLICY prosedürü :

```
dbms_fga.add_policy ( object_schema => 'PIET',  object_name => 'EMP',
policy_name => 'MYPOLICY1', audit_condition => NULL,
audit_column => 'SALARY,COMMISSION_PCT',
audit_column_opts => DBMS_FGA.ALL_COLUMNS,
audit_trail => DBMS_FGA.DB_EXTENDED,
statement_types => 'INSERT, UPDATE');


SQL> connect piet/piet
Connected.

SQL> CREATE TABLE EMP (
EMPNO        NUMBER(4) NOT NULL,
ENAME        VARCHAR2(10),
JOB          VARCHAR2(9),
MGR          NUMBER(4) CONSTRAINT EMP_SELF_KEY REFERENCES EMP (EMPNO),
HIREDATE     DATE,
SAL          NUMBER(7,2),
COMM         NUMBER(7,2),
DEPTNO       NUMBER(2) NOT NULL,
CONSTRAINT EMP_PRIMARY_KEY PRIMARY KEY (EMPNO));
Table created.

SQL> INSERT INTO EMP VALUES (7839,'KING','PRESIDENT',NULL,'17-NOV-81',5000,NU;
1 row created.
...
SQL> grant all on emp to miller;
Grant succeeded.

conn system/manager

SQL> execute sys.DBMS_FGA.ADD_POLICY(-
object_schema   => 'PIET', -
object_name     => 'EMP', -
policy_name     => 'mypolicy1', -
audit_condition => 'sal < 1000', -
audit_column    => 'comm', -
enable          => TRUE,  -
statement_types => 'INSERT');
```

```
PL/SQL procedure successfully completed.

SQL> select * from DBA_AUDIT_POLICY_COLUMNS ;

OBJECT_SCHEMA                  OBJECT_NAME
------------------------------ ------------------------------
POLICY_NAME                    POLICY_COLUMN
------------------------------ ------------------------------
PIET                           EMP
MYPOLICY1                      COMM

SQL> select OBJECT_SCHEMA, OBJECT_NAME, POLICY_NAME, POLICY_TEXT,
POLICY_COLUMN, ENABLED, SEL, INS, UPD, DEL
from DBA_AUDIT_POLICIES ;

OBJECT_SCHEMA OBJECT_NAME POLICY_NAME POLICY_TEXT POLICY_COLUMN ENA
SEL INS UPD DEL
------------- ----------- ----------- ----------- ------------- --- --- --- --- ---
PIET          EMP         MYPOLICY1   sal < 1000  COMM          YES NO  YES NO  NO

SQL> conn miller/miller
Connected.

---------------------------------
1.1 sal < 1000 ==> INSERT audited
---------------------------------

SQL> INSERT INTO PIET.EMP(EMPNO, ENAME, SAL, COMM, DEPTNO)
VALUES(1000, 'SAM', 800, 15, 10);
1 row created.

-------------------------------------
1.2 sal = 1000 ==> INSERT not audited
-------------------------------------

SQL> INSERT INTO PIET.EMP(EMPNO, ENAME, SAL, COMM, DEPTNO)
VALUES(3000, 'TOM', 20000, 1000, 20);
1 row created.

-----------------------------------------------------------
1.3 Audit column comm is not present --> INSERT not audited
-----------------------------------------------------------

SQL> INSERT INTO PIET.EMP (EMPNO, ENAME, SAL,DEPTNO)
VALUES (1111, 'RAMA', 98,30);
1 row created.

SQL> commit;
Commit complete.

conn system/manager

SQL> select DB_USER,OBJECT_SCHEMA "SCHEMA",OBJECT_NAME,
POLICY_NAME,SQL_TEXT
from dba_fga_audit_trail ;


DB_USER  SCHEMA   OBJECT POLICY_NAME
```

-------- -------- ------ ------------------------------
SQL_TEXT

MILLER PIET EMP MYPOLICY1
INSERT INTO PIET.EMP(EMPNO, ENAME, SAL, COMM, DEPTNO)
VALUES(1000, 'SAM', 800, 15, 10)

Veritabanı Loglarının veritabanı yöneticisinin değiştirmesine olanak verilmeyecek biçimde, işletim sisteminin log dosyasına direk olarak yazılması için bulunan yöntemler.
(Not: Tez çalışmasının ilk aşamasında, bu yöntem tespit edilene dek, oluşan FGA tabloları, Hyperion isimli bir raporlama aracından faydalanarak, CSV formatında Ossim Log sunucusuna aktarılmış, bir eklenti aracılığı ile sunucu veritabanına veriler basılmıştır. Daha sonra direk olarak log sunucusunun üzerine verilerin değiştirilmeden aktarımı amacı ile çalışma başlatılmış, tez çalışmasının raporlanmasının son aşamasında bu yöntemin çalıştığı teyit edilmiştir.)

1) Oracle açılış konfigürasyonunda yapılacak bir değişiklik ile **AUDIT_SYSLOG_LEVEL** parametresine girilecek değerler vasıtası ile **AUDIT_TRAIL** parametresi İşletim Sisteminin kullandığı SYSLOG mekanizmasına yönlendirilebilmektedir.

2)SYSLOG sistemine logların atılması işlemini başlatmak üzere yapılan konfigürasyonlar;

2.1) Veritabanı konfigürasyonunda denetim işinin İşletim Sistemine devredilmesi

Örnek: **SQL> ALTER SYSTEM SET AUDIT_TRAIL=OS SCOPE=SPFILE;**

2.2) **AUDIT_SYSLOG_LEVEL** parametresinin değiştirilmesi :

SQL> ALTER SYSTEM SET AUDIT_SYSLOG_LEVEL="local1.warning"
SCOPE=SPFILE;
AUDIT_SYSLOG_LEVEL parametresinin öncelik ve içerik parametrelerinin belirlenmesi. **AUDIT_SYSLOG_LEVEL=facility.priority**.

Facility: Mesajların loglanması için işletim sisteminin hangi özelliğinin kullanılacağı burada belirtilir. Kabul edilen parametreler: User, local0–local7, syslog, daemon, kern, mail, auth, lpr, news, uucp, and cron.

Priority: Mesajın öncelik bilgisi, varsayılan bırakılabilir. (Notice, info, debug, warning, err, crit, alert, and emerg).

2.3) Denetim verilerinin (mesajların) syslog konfigürasyonunda nereye gönderileceğini belirtir.

Örnek: Denetim seviyesini warning olarak ayarladığımızı var sayalım.
AUDIT_SYSLOG_LEVEL to **local1.warning**, bu durumda :

local1.warning /var/log/audit.log
parametresi ile **/var/log/audit.log** dosyasına veriler gönderilir.

2.4) Syslog sistemini işletim sisteminde aşağıdaki komut ile yeniden başlatırız.:

$/etc/rc.d/init.d/syslog restart
Syslog servisine bütün loglar atılmış olacaktır. .

8.3 OSSİM KONFİGÜRASYONUNA EKLENTİLER (OSSİM PLUGİNS)

Ossim log yönetim sisteminde osiris.cfg, ossec.cfg, ossim-agent.cfg, ossim-monitor.cfg, p0f.cfg, snare.cfg, sudo.cfg, symantec-ams.cfg, syslog.cfg, tarantella.cfg, vmware-workstation.cfg, cisco.cfg isimler ile yer alan Tablo 8.2'de belirtilen çok sayıda eklenti konfigürasyonu, günümüzde sektörde kullanılan bir çok sistemin log kayıt desenlerinin çözümlenerek ossim log sunucusu veritabanına aktarılması işlemini yürütmektedir. Alınması gereken diğer log bilgileri için, tez çalışmasında yeni eklenti dosyaları geliştirilmiştir. Eklenti dosyaları direk olarak Ossim sunucusuna bir ajan yardımı rotasyon edilemeyen log verileri için gerekli bir çözümdür. Logları oluşturan sistem çalıştığı sunucu linux veya unix ise, syslog mekanizmasına log üretemediği takdirde bu yöntem kullanılabilir. Syslog sistemine log çıkabilen araçların logları ise direk olarak işletim sistemi fonksiyonları ile Ossim sunucusunun syslog yapısına bu logları aktarabilirler.

Tablo 8.2: Ossim sisteminde hazır olan eklentiler listesi

apache.cfg	nagios.cfg	postfix.cfg
arpwatch.cfg	nessus-monitor.cfg	realsecure.cfg
cisco-ids.cfg	netgear.cfg	rrd.cfg
cisco-pix.cfg	netscreen-firewall.cfg	session-monitor.cfg
cisco-router.cfg	netscreen-manager.cfg	snare.cfg
cisco-vpn.cfg	nmap-monitor.cfg	snort.cfg
clamav.cfg	nortel-switch.cfg	snortunified.cfg
clurgmgr.cfg	ntop-monitor.cfg	sophos.cfg
dhcp.cfg	ntsyslog.cfg	spamassassin.cfg
fortigate.cfg	opennms-monitor.cfg	squid.cfg
fw1ngr60.cfg	oracletest.cfg	squid-orj.cfg
gfi.cfg	osiris.cfg	ssh.cfg
heartbeat.cfg	ossec.cfg	stonegate.cfg
iis.cfg	ossim-agent.cfg	sudo.cfg
intrushield.cfg	ossim-monitor.cfg	symantec-ams.cfg
ipfw.cfg	p0f.cfg	syslog.cfg
iphone.cfg	pads.cfg	tarantella.cfg
iptables.cfg	pam_unix.cfg	tcptrack-monitor.cfg
mwcollect.cfg	ping-monitor.cfg	test.cfg

8.3.1 Ossim ile Oracle Veritabanı Aktivelerine Ait Logların Alınması

Ossim tarafında henüz bir eklenti ve mekanizma geliştirilmemiş olan Oracle veritabanı ve diğer sistemler için, o sistemdeki loglama veya audit mekanizması çalışılarak, log rotasyonları için sistemler geliştirilmesi gerekmektedir. Bu sürece ait yöntem tez çalışmasında şu şekilde gerçekleştirilmiştir:

- Eklenti oluşturmak üzere (Plugin eklenmesi) "/etc/ossim/agent/plugins" dizininde, "vim /etc/ossim/ossim_setup.conf" dosyasında gerekli düzeltmeler yapılmalıdır. (Bkz. Kod 8.2)
- Log verisi alınacak kayıt sisteminin desen yapısı belirlenmelidir. En önemli anahtar alanlar ayrı ayrı alındıkları takdirde, girişim tespit sistemlerinde belirleyici rol oynayacaklardır. Bunlar arasında zaman damgası, kullanıcı kodu ve yapılan aktiviteler anahtar rol taşımaktadır. (Tablo xx)
- "/etc/init.d/ossim-agent restart", "/etc/init.d/ossim-servers restart" komut adımları yerine getirilerek sunucu yeniden başlatılmalıdır.

Kod 8.2: Plugin listesine yeni bir eklenti eklemek

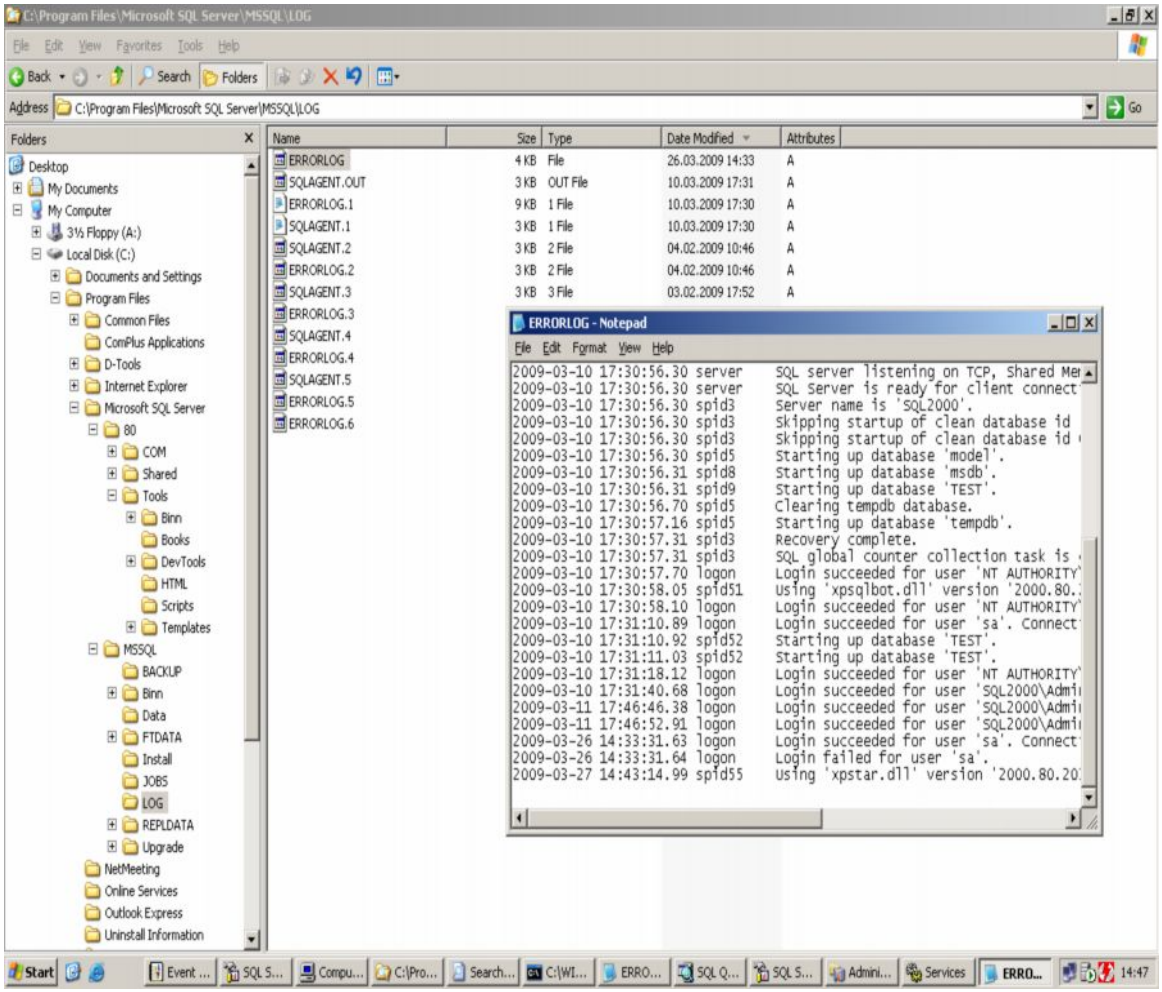
```
[output-db]
base=ossim_events
enable=True
host=localhost
pass=1234
type=mysql
user=root

[output-plain]
enable=False
file=/var/log/ossim/agent-plain.log

[output-server]
enable=True
ip=10.16.13.101
port=40001
```

```
[plugin-defaults]
date_format=%Y-%m-%d %H:%M:%S ; format, not date itself
interface=eth0
ossim_dsn=mysql::ossim:root:1qaz2wsx3edc
sensor=10.16.13.101

[plugins]
arpwatch=/etc/ossim/agent/plugins/arpwatch.cfg
dhcp=/etc/ossim/agent/plugins/dhcp.cfg
iptables=/etc/ossim/agent/plugins/iptables.cfg
nagios=/etc/ossim/agent/plugins/nagios.cfg
nmap=/etc/ossim/agent/plugins/nmap-monitor.cfg
ntop=/etc/ossim/agent/plugins/ntop-monitor.cfg
oracletest=/etc/ossim/agent/plugins/oracletest.cfg
osiris=/etc/ossim/agent/plugins/osiris.cfg
ossim-ca=/etc/ossim/agent/plugins/ossim-monitor.cfg
p0f=/etc/ossim/agent/plugins/p0f.cfg
pads=/etc/ossim/agent/plugins/pads.cfg
pam_unix=/etc/ossim/agent/plugins/pam_unix.cfg
rrd=/etc/ossim/agent/plugins/rrd.cfg
snare=/etc/ossim/agent/plugins/snare.cfg
snort=/etc/ossim/agent/plugins/snortunified.cfg
ssh=/etc/ossim/agent/plugins/ssh.cfg
sudo=/etc/ossim/agent/plugins/sudo.cfg
hyperion=/etc/ossim/agent/plugins/hyperion.cfg---TEZ
oracleDBA=/etc/ossim/agent/plugins/oracleDBA.cfg---TEZ
```

8.3.2 Ossim ile MS-SQL Veritabanı Aktivelerine Ait Logların Alınması

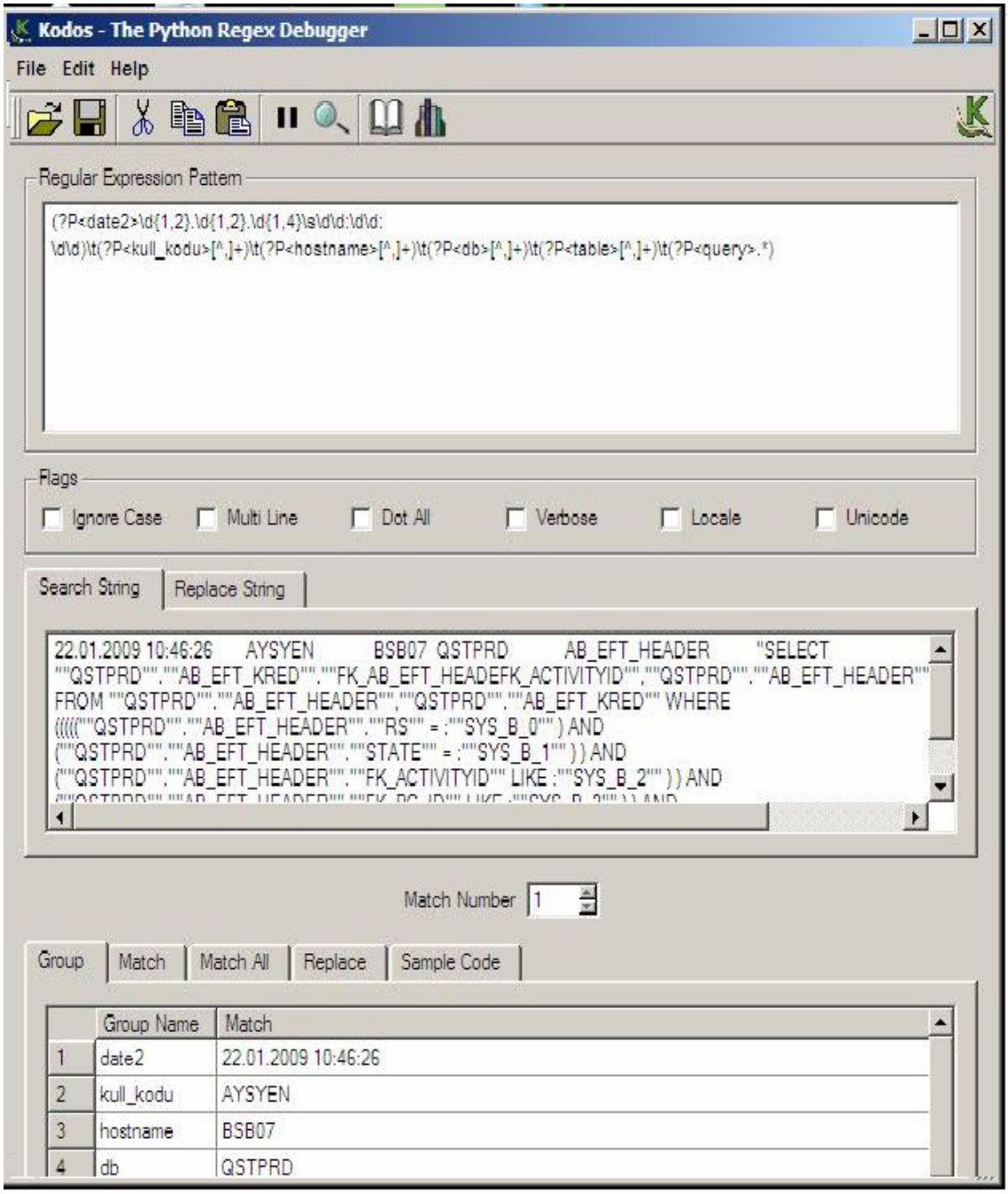
Ossim tarafında henüz bir eklenti ve mekanizma geliştirilmemiş olan diğer bir sistemde, yine sıkça kullanılmakta olan Ms-Sql veritabanıdır. 8.3.1'de anlatılan yöntemler geçerli olmak üzere, Ms-Sql sisteminde yapılması gereken işlemler çözümlenmeye çalışılmıştır.

Ms-Sql veritabanlarının kurulumunda özellikler bölümünde varsayılan denetim (audit) seçenekleri bırakıldığı takdirde, bu veritabanına başarısız erişimlere ait log kayıtları, log dosyalarına yazılmamaktadır. 1'den 6'ya kadar olan derinlik değerinin 6 olarak belirtilmesi gereklidir. (Çalışma Ms-Sql 2000 ve 2005 versiyonlarında gerçekleştirilmiştir). Şekil 8.1'de audit seviyesi 6'ya çekilen log dosyasında içerik örnekleri yer almaktadır.



Şekil 8.1: Audit seviyesi ayarlanmış olan Ms-Sql Log Örneği (Başarısız erişimler de izlenebilmektedir)

Bu logların Ossim sunucusuna rotasyonunu gerçekleştirmek üzere, log deseninin incelenerek, syslog yapısındaki uygun formata getirilmesi gerekmektedir. Bu çalışmalar için açık kaynak kodlu Regex ve benzeri araçlar kullanılabilir.



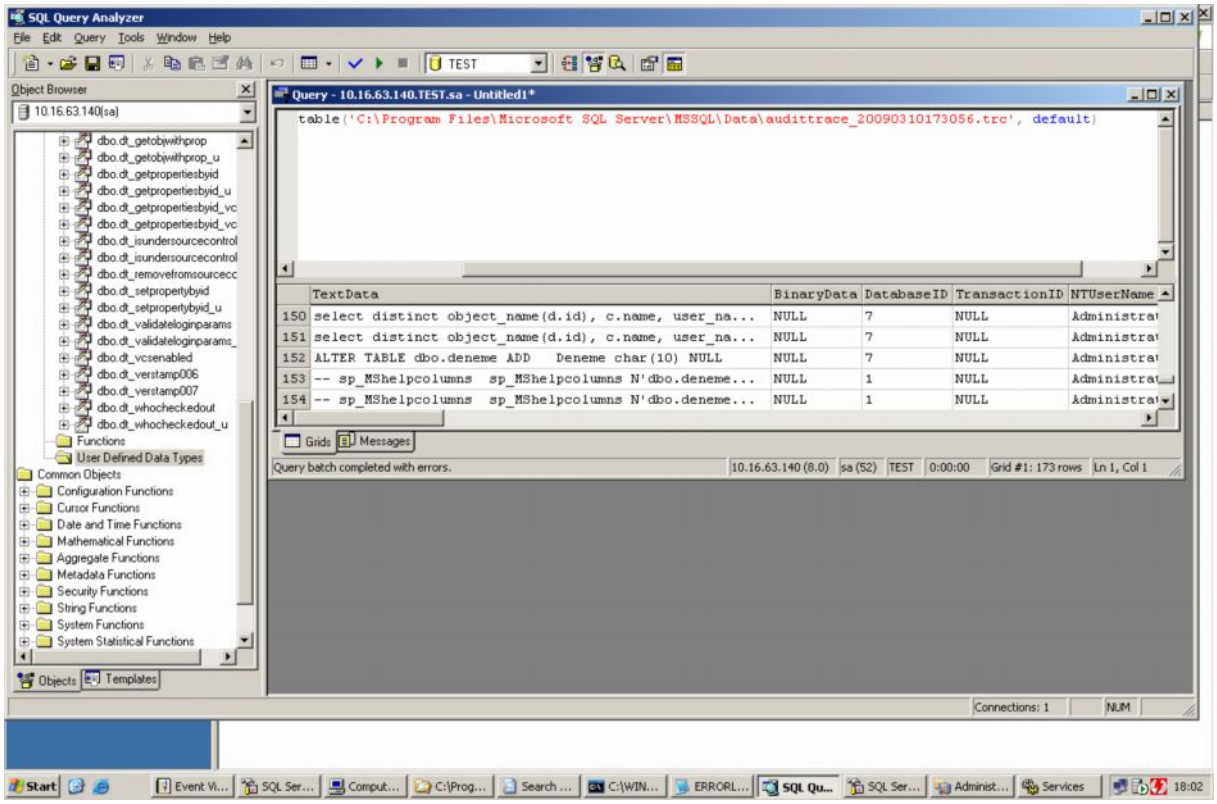
Şekil 8.2: Ms-Sql Loglarına ait kayıt desenlerinin belirlenmesi (Regex)

Herhangi bir tabloda veritabanına direk olarak erişerek bir değişiklik yapılmasına yönelik logların oluşturulması için ise, Ms-Sql veritabanlarında SQL Profiler konfigürasyonlarının yapılması gerekmektedir. Yine performans sorunları ve disk üzerinde artan yazma işlevinin minimize edilmesi için, tablolar üzerinde ön araştırma yapılmalı ve gerekli olan tablolarda log kayıtlarının tutulması sağlanmalıdır. İzlemek üzere Trace tablosunun aktivite edilmedi gerekmektedir (Şekil 8.3)

Veritabanında gerçekleşen işlemlerin detaylı log kaydının tutulması için (tablolar üzerinde gerçekleşen aktiviteler dahil) aşağıdaki kod setlerinin çalışması gerekmektedir. Bu kod setleri ile MS-Sql log dizininde trace ile ilgili yeni bir log dosyası oluşmaktadır SELECT * FROM ::fn_trace_gettable('C:\Program Files\Microsoft SQL Server\MSSQL\Data\audittrace_20040822191554.trc', default)． Bu log dosyasını veritabanı yönetim ekranından (SQL Management) görmek mümkündür.

Kod 8.3: Ms Sql trace /takip) loglarının aktive edilmesi

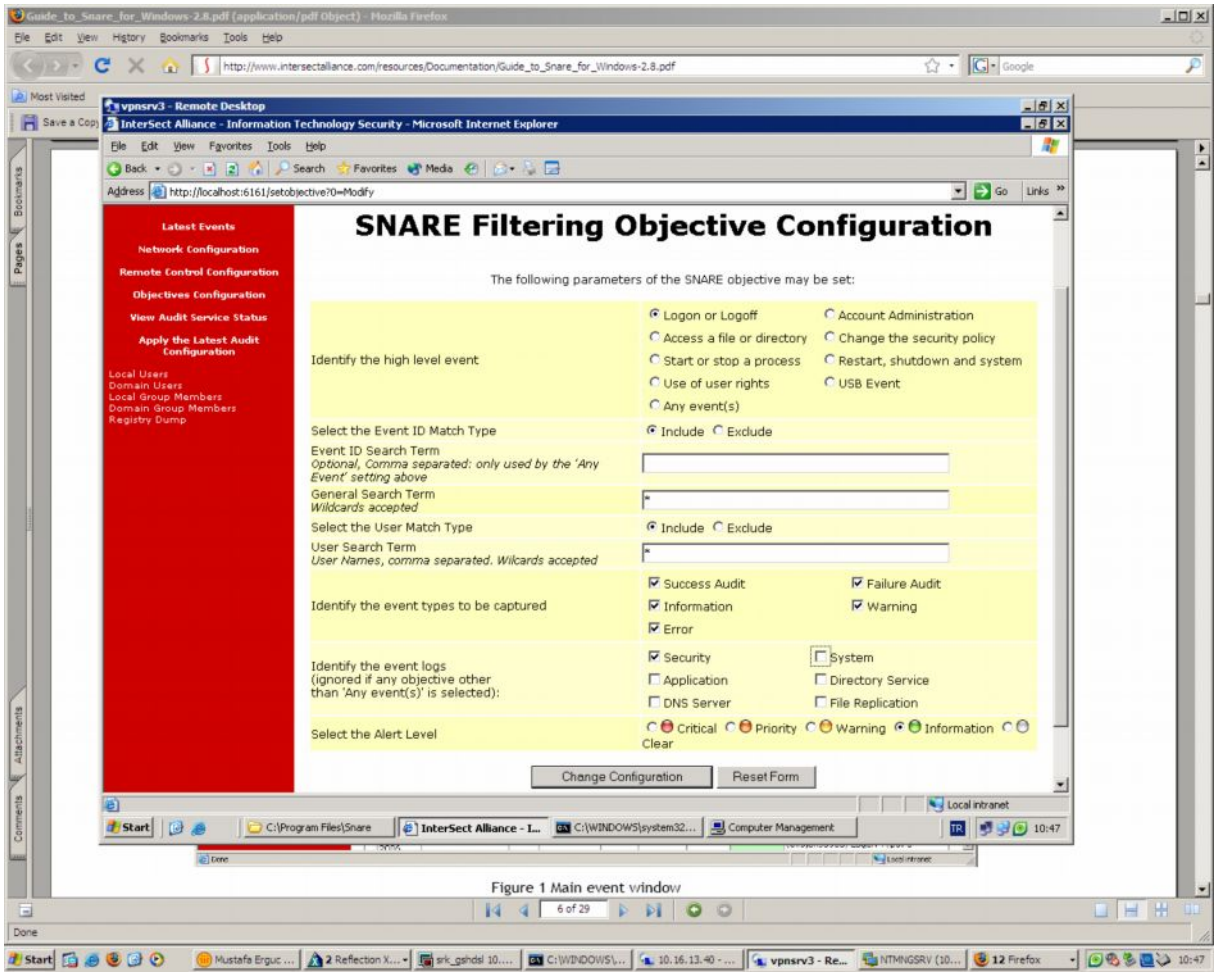
```
USE test
EXEC sp_configure 'show advanced option', '1'
RECONFIGURE

EXEC sp_configure 'c2 audit mode', 1
RECONFIGURE
```



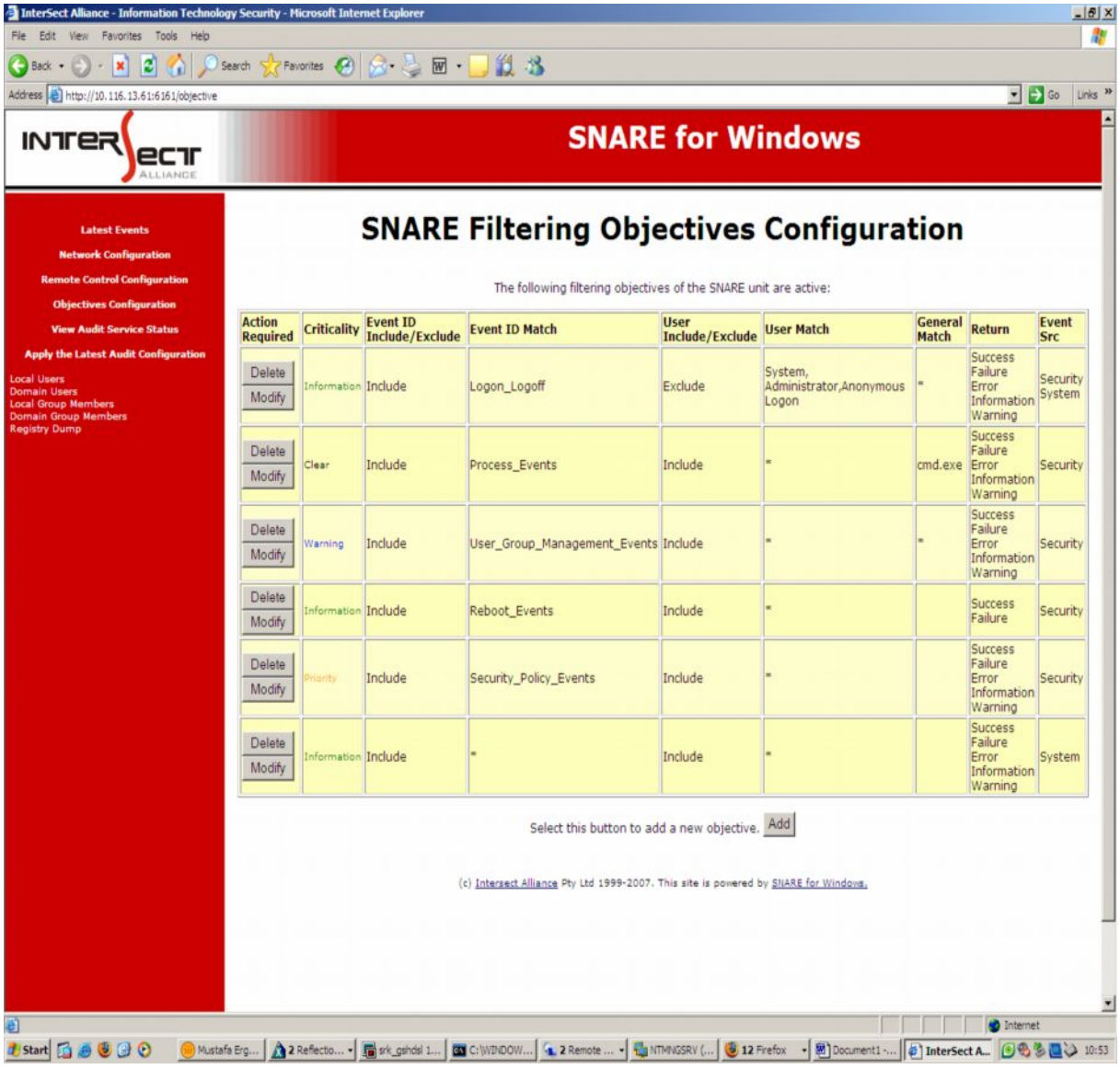
Şekil 8.3: Ms-Sql SQL Profiler yapısının aktive edilmesi

8.3.3 Açık Kaynak Kodlu Ajanlar ile Windows Sistem Loglarının Ossim Log Sunucusuna Rotasyonu

Windows sistemlerinde Güvenlik (Security), Sistem (System) ve Uygulama (Application) başlığında 3 tür log üretilmektedir. Örneğin DHCP sunucusunun logları, VPN sunucusunun logları ve diğer windows sisteminde çalışan servislerin logları da buraya toplanmaktadır. Windows sistemlerinden log almak üzere bir çok açık kaynak kodlu ajan bulunmaktadır. Bunlar arasında en iyi konfigüre edilebilir ve performans olarak değerlendirilebilir olan Snare yazılımı tez çalışmasında windows sunucularda kullanılmıştır.

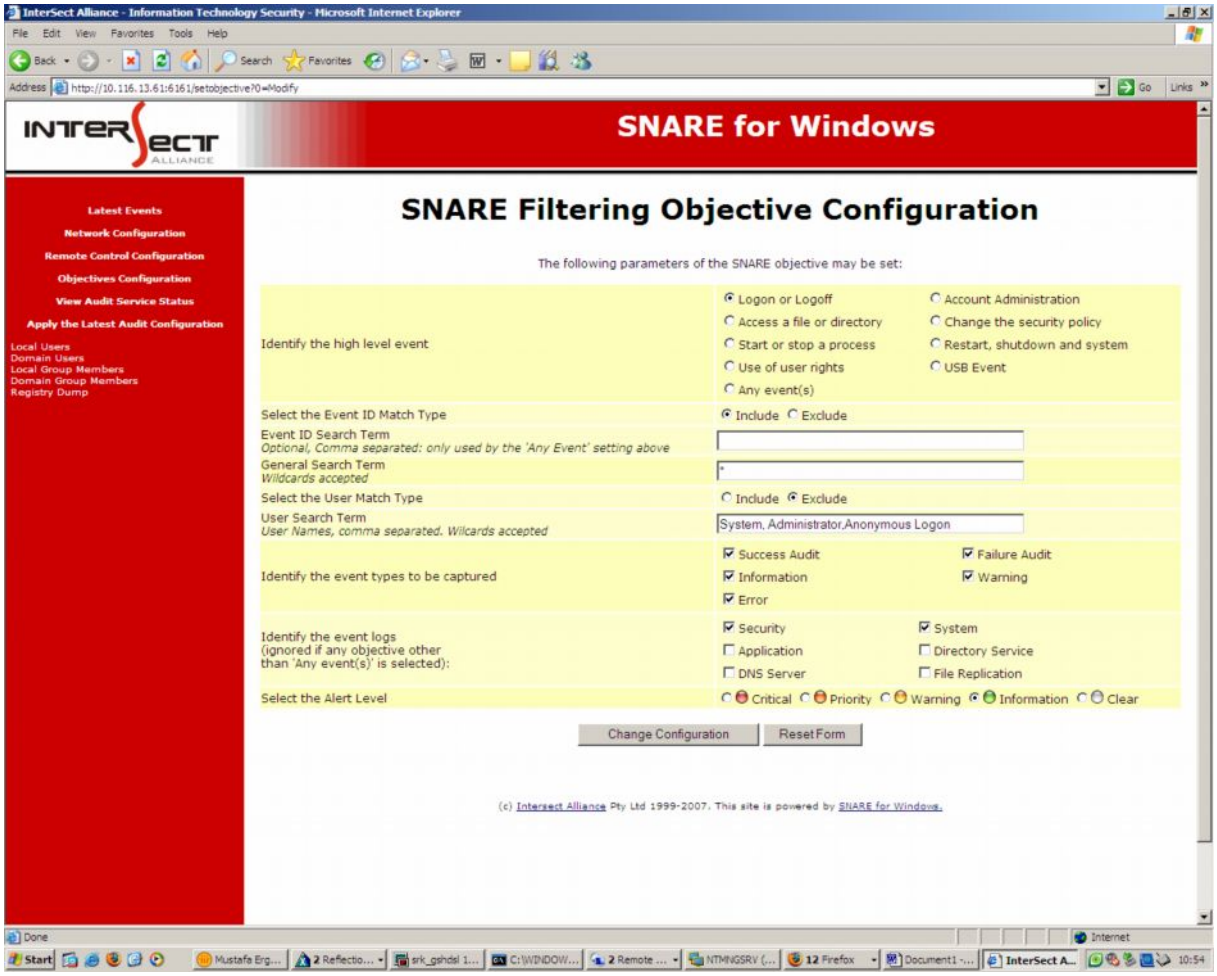


Şekil 8.4: Windows işletim sistemlerinde snare agent konfigürasyon ekranı

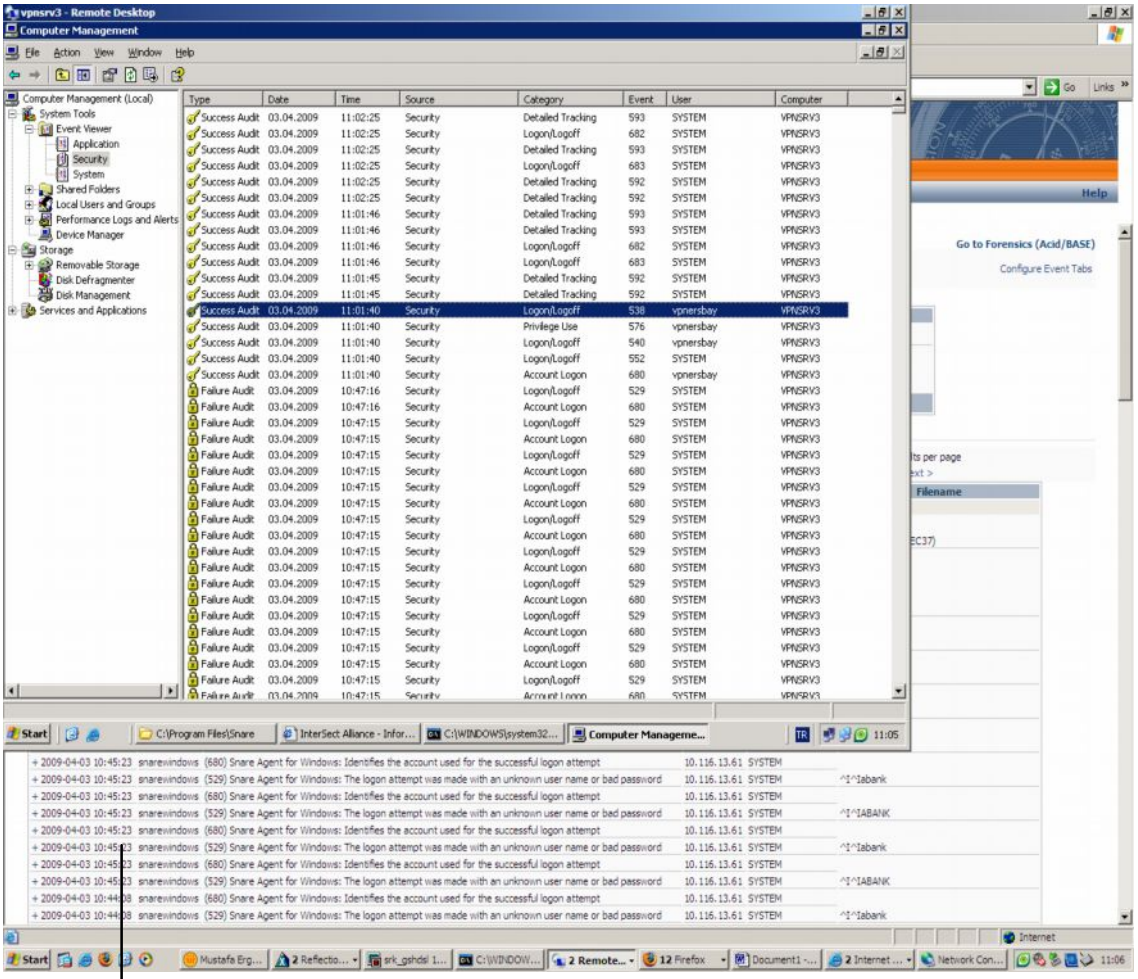


Şekil 8.5: Windows işletim sistemlerinde snare agent ile yapılabilecek log indirgeme (filtreleme) seçenekleri

Snare ajanı ile varsayılan konfigürasyonlarla logların toplanması işlemini gerçekleştirildiğinde karşımıza çok sayıda event kaydı çıkabilecektir. Bu nedenle konfigürasyonları düzenlemek üzere windows sunucusunun hangi kritik rolü üstlendiği üzerinde çalışılarak, buna ilgin log kayıtlarının toplanması ve diğer önemli loglar üzerinde durmak gerekmektedir. Örneğin VPN sunucusu windows işletim sistemi üzerinde çalışıyorsa, önemli olan aksiyonlar uzaktan bağlanan kullanıcının başarı ve başarısız denemeleri, ne zaman sisteme login olup, ne zaman sistemden ne zaman çıktığına dair log toplanması olabilir.



Şekil 8.6: Windows işletim sistemlerinde snare agent konfigürasyonları



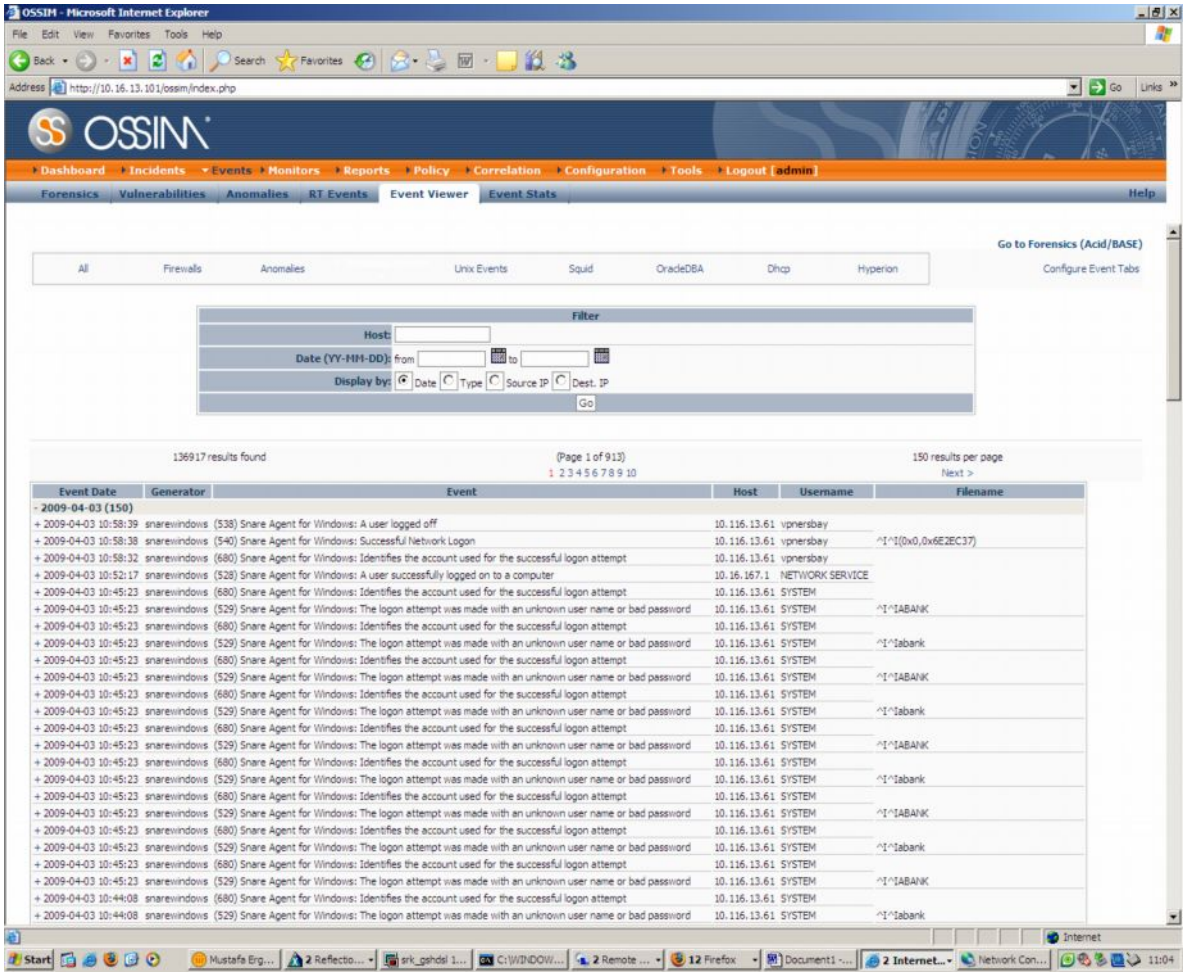
Şekil 8.7: Windows işletim sistemlerinde loglar

Plugin: Snarewindows (1518)

Plugin SID: Snare Agent For Windows: Successful Network Logon (540)

Username: Hussar

Userdata 1: Successful Network Logon: ^Iuser Name:^Ihussar ^Idomain:^I^IVPNSRV3 ^Ilogon ID:^I^I(0x0,0x349929a) ^Ilogon Type:^I3 ^Ilogon Process:^IIAS ^Iauthentication Package:^IMICROSOFT_AUTHENTICATION_PACKAGE_V1_0 ^Iworkstation Name:^I ^Ilogo



Şekil 8.8: Ossim sunucusuna rotasyon yapılmış Windows işletim sistemi logları

8.3.4 Ossim veritabanında bulunan log verilerin değiştirilmezliğinin sağlanması

Logların şifrelenmesi, güvenli bir şekilde iletimi konusunda asimetrik anahtar kullanımı ve diğer yöntemler konusunda "Design And Implementation Of A Secure And Searchable Audit Logging System" makalesinde çeşitli yöntemler anlatılmıştır (İncebacak, Davut 2007).

Tez çalışmasında ise, kullanılan sistemlerden yararlanarak basitleştirilmiş ve kullanılabilir checksum, (özet kontrol değeri) yapısı önerilmiştir. Ossim tarafından kullanılan log depolama arası Mysql veritabanından text formatında export edilebilecek dosyaların, haftalık olarak özet değerlerini çıkaran ve "c" dili ile yazılan uygulamadan tez çalışmasında yararlanılmıştır.

Kod 8.4: Checksum (Özet değer) oluşturan yazılım kodu

```
/* cksum  */
#define PROGRAM_NAME "cksum"
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#if 0
# include <sys/types.h>
# include "system.h"
# include "closeout.h"
# include "long-options.h"
# include "error.h"
#endif

// extern int errno ;

/* Number of bytes to read at once.  */
# define BUFLEN (1 << 16)

void gettime(const char *) ;

static unsigned long const crctab[256] =
{
0x0,
0x04C11DB7, 0x09823B6E, 0x0D4326D9, 0x130476DC, 0x17C56B6B,
0x1A864DB2, 0x1E475005, 0x2608EDB8, 0x22C9F00F, 0x2F8AD6D6,
0x2B4BCB61, 0x350C9B64, 0x31CD86D3, 0x3C8EA00A, 0x384FBDBD,
0x4C11DB70, 0x48D0C6C7, 0x4593E01E, 0x4152FDA9, 0x5F15ADAC,
0x5BD4B01B, 0x569796C2, 0x52568B75, 0x6A1936C8, 0x6ED82B7F,
0x639B0DA6, 0x675A1011, 0x791D4014, 0x7DDC5DA3, 0x709F7B7A,
0x745E66CD, 0x9823B6E0, 0x9CE2AB57, 0x91A18D8E, 0x95609039,
0x8B27C03C, 0x8FE6DD8B, 0x82A5FB52, 0x8664E6E5, 0xBE2B5B58,
0xBAEA46EF, 0xB7A96036, 0xB3687D81, 0xAD2F2D84, 0xA9EE3033,
0xA4AD16EA, 0xA06C0B5D, 0xD4326D90, 0xD0F37027, 0xDDB056FE,
0xD9714B49, 0xC7361B4C, 0xC3F706FB, 0xCEB42022, 0xCA753D95,
0xF23A8028, 0xF6FB9D9F, 0xFBB8BB46, 0xFF79A6F1, 0xE13EF6F4,
0xE5FFEB43, 0xE8BCCD9A, 0xEC7DD02D, 0x34867077, 0x30476DC0,
0x3D044B19, 0x39C556AE, 0x278206AB, 0x23431B1C, 0x2E003DC5,
0x2AC12072, 0x128E9DCF, 0x164F8078, 0x1B0CA6A1, 0x1FCDBB16,
0x018AEB13, 0x054BF6A4, 0x0808D07D, 0x0CC9CDCA, 0x7897AB07,
0x7C56B6B0, 0x71159069, 0x75D48DDE, 0x6B93DDDB, 0x6F52C06C,
0x6211E6B5, 0x66D0FB02, 0x5E9F46BF, 0x5A5E5B08, 0x571D7DD1,
0x53DC6066, 0x4D9B3063, 0x495A2DD4, 0x44190B0D, 0x40D816BA,
0xACA5C697, 0xA864DB20, 0xA527FDF9, 0xA1E6E04E, 0xBFA1B04B,
0xBB60ADFC, 0xB6238B25, 0xB2E29692, 0x8AAD2B2F, 0x8E6C3698,
0x832F1041, 0x87EE0DF6, 0x99A95DF3, 0x9D684044, 0x902B669D,
0x94EA7B2A, 0xE0B41DE7, 0xE4750050, 0xE9362689, 0xEDF73B3E,
0xF3B06B3B, 0xF771768C, 0xFA325055, 0xFEF34DE2, 0xC6BCF05F,
0xC27DEDE8, 0xCF3ECB31, 0xCBFFD686, 0xD5B88683, 0xD1799B34,
0xDC3ABDED, 0xD8FBA05A, 0x690CE0EE, 0x6DCDFD59, 0x608EDB80,
0x644FC637, 0x7A089632, 0x7EC98B85, 0x738AAD5C, 0x774BB0EB,
0x4F040D56, 0x4BC510E1, 0x46863638, 0x42472B8F, 0x5C007B8A,
0x58C1663D, 0x558240E4, 0x51435D53, 0x251D3B9E, 0x21DC2629,
0x2C9F00F0, 0x285E1D47, 0x36194D42, 0x32D850F5, 0x3F9B762C,
0x3B5A6B9B, 0x0315D626, 0x07D4CB91, 0x0A97ED48, 0x0E56F0FF,
0x1011A0FA, 0x14D0BD4D, 0x19939B94, 0x1D528623, 0xF12F560E,
0xF5EE4BB9, 0xF8AD6D60, 0xFC6C70D7, 0xE22B20D2, 0xE6EA3D65,
0xEBA91BBC, 0xEF68060B, 0xD727BBB6, 0xD3E6A601, 0xDEA580D8,
0xDA649D6F, 0xC423CD6A, 0xC0E2D0DD, 0xCDA1F604, 0xC960EBB3,
0xBD3E8D7E, 0xB9FF90C9, 0xB4BCB610, 0xB07DABA7, 0xAE3AFBA2,
```

```
0xAAFBE615, 0xA7B8C0CC, 0xA379DD7B, 0x9B3660C6, 0x9FF77D71,
0x92B45BA8, 0x9675461F, 0x8832161A, 0x8CF30BAD, 0x81B02D74,
0x857130C3, 0x5D8A9099, 0x594B8D2E, 0x5408ABF7, 0x50C9B640,
0x4E8EE645, 0x4A4FFBF2, 0x470CDD2B, 0x43CDC09C, 0x7B827D21,
0x7F436096, 0x7200464F, 0x76C15BF8, 0x68860BFD, 0x6C47164A,
0x61043093, 0x65C52D24, 0x119B4BE9, 0x155A565E, 0x18197087,
0x1CD86D30, 0x029F3D35, 0x065E2082, 0x0B1D065B, 0x0FDC1BEC,
0x3793A651, 0x3352BBE6, 0x3E119D3F, 0x3AD08088, 0x2497D08D,
0x2056CD3A, 0x2D15EBE3, 0x29D4F654, 0xC5A92679, 0xC1683BCE,
0xCC2B1D17, 0xC8EA00A0, 0xD6AD50A5, 0xD26C4D12, 0xDF2F6BCB,
0xDBEE767C, 0xE3A1CBC1, 0xE760D676, 0xEA23F0AF, 0xEEE2ED18,
0xF0A5BD1D, 0xF464A0AA, 0xF9278673, 0xFDE69BC4, 0x89B8FD09,
0x8D79E0BE, 0x803AC667, 0x84FBDBD0, 0x9ABC8BD5, 0x9E7D9662,
0x933EB0BB, 0x97FFAD0C, 0xAFB010B1, 0xAB710D06, 0xA6322BDF,
0xA2F33668, 0xBCB4666D, 0xB8757BDA, 0xB5365D03, 0xB1F740B4
};

unsigned long nCheckSumBuf( const char *pszBuf, long nBytes, unsigned
long nCRC )
{
while (nBytes--)
{
nCRC = (nCRC << 8) ^ crctab[((nCRC >> 24) ^ *(pszBuf++)) & 0xFF];
}
return nCRC;
}

/* Calculate and fill in the checksum and length in bytes
of file FILE.
Return 0 if successful, -1 if an error occurs. */

int  nCheckSumFile(const  char  *pszDirectory,  const  char  *pszFile,
unsigned long *pnCRC, unsigned long *pnLength )
{
unsigned char     caBuf[BUFLEN];
unsigned long     nCRC = 0;
long              nLength = 0;
long              nBytesRead;
register FILE     *fp;
// char                 caFullPath[_MAX_PATH];
char              caFullPath[BUFLEN];

caFullPath[0] = '\0';
if ( pszDirectory != NULL )
{
strcpy( caFullPath, pszDirectory );
strcat( caFullPath, "\\" );
}
strcat( caFullPath, pszFile );
fp = fopen (caFullPath, "rb");
if (fp == NULL)
{
return -1;
}

#if 0
/* Read input in BINARY mode, unless it is a console device.  */
SET_BINARY (fileno (fp));
#endif
```

```
while ((nBytesRead = fread (caBuf, 1, BUFLEN, fp)) > 0)
{
unsigned char *cp = caBuf;

nLength += nBytesRead;
nCRC  = nCheckSumBuf( cp, nBytesRead, nCRC );
}

if (ferror (fp))
{
return -2;
}

if (fclose (fp) == EOF)
{
return -3;
}

// now checksum the length
nBytesRead  = nLength;
while (nBytesRead > 0)
{
nCRC  = (nCRC << 8) ^ crctab[((nCRC >> 24) ^ nBytesRead) & 0xFF];
nBytesRead >>= 8;
}

nCRC        = ~nCRC & 0xFFFFFFFF;

*pnCRC                                      = nCRC;
*pnLength                               = nLength;

return 0;
}

int main( int argc, char** argv)
{
unsigned long nCRC;
unsigned long nLength;
char  logfile[100] ;
char  tmp_time[20] ;
int   i ;
FILE  *fp ;


for (i = 0 ; i++; i < 100 )
logfile[i]='\0' ;

sprintf(logfile, "%s%s", argv[0], ".out") ;

if (argc < 2)
{
fprintf (stdout, "cksum inputfile");
return -1;
}

if ( (fp = fopen(logfile,"a+")) == NULL )
{
/* printf("Hata! %s açılamadı, %s\n",logfile,strerror(errno)) ; */
```

```
printf("Hata! %s açılamadı\n",logfile) ;
exit(2) ;
}


if (nCheckSumFile(NULL, argv[1], &nCRC, &nLength ) == 0)
{
/* fprintf (stdout, "%u %u", nCRC, nLength); */
gettime(tmp_time);
fprintf (fp, "%s\t%s\t%u\t%u\n",tmp_time, argv[1], nLength, nCRC);
return 0;
}
else
{
return -1;
}
fclose(fp) ;

}

void gettime(const char * out_time)
{
time_t    today ;
struct tm  * time_ptr ;

time(&today) ;
time_ptr = localtime(&today) ;
sprintf(out_time, "%4.4d%2.2d%2.2d:%2.2d%2.2d%2.2d",
(2000 + (time_ptr->tm_year%100)), time_ptr->tm_mon, time_ptr->tm_mday,
time_ptr->tm_hour , time_ptr->tm_min ,time_ptr->tm_sec ) ;
```

Tablo 8.3: Log dosyalarının değiştirilmezliği için geliştirilen checksum uygulamasının çıktı örnekleri

Checksum dosyası:

20090317:165601	cksum.c	6378	1649157310
20090317:165614	cksum.c	6378	1649157310
20090317:165614	cksum.c	6378	1649157310
20090317:165615	cksum.c	6378	1649157310
20090317:165616	cksum.c	6378	1649157310
20090317:165627	cksum.exe	13926	3071514912
20090317:165628	cksum.exe	13926	3071514912

Log veritabanı arşivlenirken oluşturulacak bu checksum değerleri ile logların değiştirilemeyeceği garanti edilebilir.

9. TARTIŞMA VE SONUÇ

Log verilerinin toplanması, yalnızca girişim tespit sisteminin bir parçası olan araçlar için veri depolanması anlamına geldiği takdirde, kanıt oluşturacak diğer bilgilerin gözden kaçması söz konusudur. Güvenlik yönetimlerinin yalnızca network aktivitelerine yönelik olması, diğer girişimlerin ve bu girişimler sonucu yapılan aktivitelerin takip edilememesine neden olabilir.

Her sistemde korunması gereken en önemli risk varlığı verilerdir. Bu verilere ulaşım ve yapılan aktivitelerin iz kayıtlarının alınması son derece önemli olabilir. Suça yönelik araştırmaların yalnızca network aktivitelerini ele aldığı durumda, veritabanı ve ilgili diğer sistem loglarının ayrıca sistem yöneticileri tarafından loglarının alındığı varsayımından hareket edilmesi yeterli olmayabilir, Forte (2004). Bu bilgilerin de değiştirilmediğinden veya değişmediğinden emin olmanın bir yolu olarak, önemli ve anlamlı özellikle veri tabanı hareketlerinin yani logların, tez çalışmasında değişmez biçimde log sistemine alınması gerektiği düşünülmüştür.

Log verilerinin analiz edilmesi aşamasında, sistemlere ulaşan kullanıcı kodlarının da, zaman damgası kadar önemli olduğu varsayılabilir. Çok sayıda kullanıcı, uygulama ve sistemin yer aldığı alt yapılarda, kullanıcı kodlarının belli bir standartta oluşturulması önemli bir strateji olabilir.

Finansal Kurumların, denetim metodolojisi olarak benimsenen Cobit DS.5 (Tedarik ve Destek Kontrol Süreçleri) ve ME.2 (Gözlem ve Değerlendirme) süreçleri başta olmak üzere, ITIL Change Management (Değişim Yönetimi), ISO 27001 ve 27002 Bilgi Güvenliği yönetimi standartlarının bir çok kontrolü, Log Yönetiminin önemine odaklanmış, anahtar kontrol kriteri olarak iz kayıtlarının tutulması ve gözlemlenmesini göstermişlerdir.

Log Yönetimi çalışmalarında karşılaşılacak ilk durum, tez dokümanının önceki aşamalarında da bahsedilen sistemsel çeşitlilik ve log içeriklerindeki farklılıklarıdır. Bu nedenle log yönetimi belli bir sistematik ve yaklaşım oluşturularak başlatılmalıdır.

Örneğin değişim yönetimi kapsamında, test ortamında test edilen bir kaynak kodunun, üretim ortamına atılırken, hangi değişim nedeni ile devreye alındığının, onay mekanizmalarından geçilerek yapılması sağlanır. Burada en uygun olan yöntemlerden birisi, geliştiricinin (developer) test ortamına verdiği kaynak kodunun, bağımsız başka bir birim tarafından yeniden binary haline getirilmesi, test ortamındaki ile aynı binary dosya uzunluğunda binary çalışabilir kod elde edilmesinin garantisinin aranması olabilir. Yapılan işlemin yetki verilmiş hangi kullanıcı tarafından yapılmış olduğu ise, bir iz kaydı sonucudur ve bu sonuç bir log kayıt sisteminde tutulabilir. Bu bilginin ne kadar kritik olup olmadığına, logların oluşturulup oluşturulmayacağına, sistemin veya kurumun sahipleri tarafından yasal düzenlemeler de göz önüne alınarak karar verilmesi gerekir. Diğer yandan teknolojiyi yöneten uzmanlar, tespit edici, önleyici veya iz kaydı gerekebilecek sistemler için nasıl bir takip mekanizması oluşturacaklarını belirlemeli , hem yasal hem de iş birimlerinin ihtiyacı doğrultusunda uygun log yönetim sistemlerini oluşturmalıdırlar.

Bilgi Sistemleri süreçleri ile gerçekleştirilen işlemler sürecin kendisi iken, bu işlemlerde uygulanan kontroller ise sürece ait kontrollerdir. Finansal yönden önemli olabilecek, hatta kişisel bilgilerin de yer aldığı veriler, sadece bu bilgileri girerken doğrulama (verify) ve giriş kontrolleri ile ele alınamayacak ölçüde öneme sahiptir. Ayrıca sistemlerin kendileri de ilk olarak konfigürasyonları açısından kritik varlıklardır. En azından bu nedenle bile, sistemler üzerindeki bilgi teknoloji süreçlerine ait kontroller de ele alınmalıdır. Bilgi Güvenliği bu süreçlerin kontrollerini, veri güvenliği çalışmaları ile paralel bir şekilde, olası bir kötü niyetli aksiyonun tespit edilebilmesi, dışardan veya rakiplerden gelen teknolojik tehditlerin önlenebilmesi amacına yönelik olarak geliştirmek durumundadır. Ayrıca yasal bir soruşturma gereken durumda kanıt elde edebilmek üzere, işlemlerin kim tarafından ne şekilde yapılmış olabileceğine dair başka bir bilgiyi daha güvenilir biçimde depolamak gerekmektedir.

Bu kontrollerin oluşturulması için gereken diğer anahtar kontrol, log yönetimidir. Log (iz) kaydı toplanması, bu iz kaydının belli süreler ile güvenilir bir biçimde saklanması, teknolojinin yoğun ve yaygın kullanımı nedeniyle son derece önemli bir proses haline gelmiştir.

Tez çalışması sırasında da, özellikle sonuç odaklılık yaklaşımı, zaman darboğazı, log yönetimi gereklerinin bilgi güvenliği farkındalığı ile sağlanması konusunda eksik olduğu düşünülen yönlendirmelerden kaynaklanabilecek nedenlerle, bu konuda şimdiye kadar teknoloji ile uğraşan, yazılım üreten çalışanların yeterince özendirilmemesi veya geliştirmelerin log oluşturması yönünde bir içerik standardı oluşturamaması gösterilebilir.

Log Yönetiminin anlamlı bir biçimde gerçekleştirilebilmesi, sistemler hakkında geniş bilgi sahibi olmayı ve kişisel ve kurumsal ağlarda işlenen verilerin ne olduğunun bilinmesini gerektirir. Bilgi sistemleri için aktivitelere ait kanıt oluşturmak, hem de güvenliğe yönelik ihtiyaçları karşılamak üzere gereken niteliklerde bir log yönetimi politikasının belirlenmesi gerekir. Araştırma sonunda ulaşılan sonuçlar ve aşamalar bu politikaya örnek oluşturacağı düşünülen şekilde sıralanmıştır.

Log bilgilerinin elde edileceği süreç ve sistemlerin belirlenmesi,

- Hangi sistemlerin hangi log bilgilerini oluşturduğu belirlenmeli ve araştırılmalıdır. Bu sistemler üzerinde çalışan servislerin ayrı log mekanizmaları olabilir,
- Bu loglar arasında hangi log bilgileri bilgi sisteminin hizmet verdiği veya kontrol ettiği veriler için kritik ve değerlendirilebilir loglardır,
- Olay kayıtlarında karakteristik verilerin belirlenmesi (Amaç),
- Logların hangi sıklıkta rotasyona tabi tutulacağının belirlenmesi (Kritik bilgilerin anlık loglanması gibi.),
- Log yönetimi sorumluluklarının belirlenmesi (Logun oluşması, gözlemlenmesi aşamasına kadar olan her adım için).

Logların sunucuya aktarımı,

- Hangi tip sistemlerim loglarının direk olarak log sunucularına gönderilebileceğinin tespit edilmesi. Logların toplanması için ajan kurulup kurulmayacağı, doğal olarak direk log sunucusuna aktarım olup olmayacağı belirlenmelidir.

- Bağımsız log sunucusuna, log üreten sistem verilerinin hangilerinin log sunucusu depolama sistemindeki kayıt desenlerinde uygun olarak ayrılabileceğinin belirlenmesi (Zaman Damgası, Kullanıcı Kodu, IP Adresi gibi..),
- Gerektiği durumlarda kritik bilgilerin şifreli olarak log rotasyonuna gireceğinin belirlenmesi,
- Log bilgilerinden filtrelenecek olanların belirlenmesi, (logun indirgenmesi), normalizasyon çalışmaları.

Logların depolanması ve düzenlemeler,

- Performans ölçümlerinin yapılması, gerektiğinde birden çok log toplayan sunucunun kullanımının planlanması,
- Disk kapasitelerinin (depolama) belirlenmesi,
- Log dosyalarının tümü ve ilgili kısmı için yasal veya kurumsal gereklilik nedeniyle değişmezliğini kanıtlayıcı önlemlerin alınması,
- Depolama sisteminden gerekli ve kullanılabilir raporlar için filtreleme mekanizmaları oluşturulması.

Log Analizi,

- Hangi sıklıkta log bilgilerinin analizinin gerçekleştirileceği belirlenmelidir.
- Kimler log verilerine ulaşabilir.
- Şüpheli aktiviteler tespit edildiğinde alınacak aksiyonların belirlenmesi.
- Hangi bilgilerin analiz sırasında gerekeceğinin belirlenmesi.
- Log verileri içerisinde olası şifre v.b. bilgilere rastlanıldığında alınacak aksiyonların belirlenmesi.

KAYNAKLAR

Kitaplar

Lochart Andrew., (2006), *Ağ Güvenliği İpuçları, 100 Etkin Güvenlik Tekniği*, Açık Akademi, İstanbul

Lyon, "Fyodor", G. (2008), *NMap Network Scanning,* Insecure. Com LLC.

Süreli Yayınlar

Stephen G. Eick., Michael C. Nelson., Jeffery D. Schmidt., (1994), Graphical Analysis of Computer Log Files, *Communications of the ACM, 37*, No. 12, December 1994, pages 50-56.

Herrerias J., Gomez H., (2007), *A Log Correlation Model to Support the Evidence Search Process in a Forensic Investigation*, *Dept of Computer Science* ITESM-CEM, IEEE, Computer Security

Takahashi D., Xiao Y., (2008), *Retrieving knowledge from auditing log-files for computer and network forensics and accountability*, Department of Computer Science, The University of Alabama, U.S.A., Security Comm. Networks. 2008; 1:147–160, John Wiley & Sons, Ltd.

Casey, D., *Turning Log Files Into A Security Asset*, Network Security, Volume 2008, Issue 2, February 2008, Pages 4-7, Science Direct

Lundin E., Erland Jonsson, *Anomaly-Based Intrusion Detection: Privacy Concerns And Other Problems Computer Networks*, Volume 34, Issue 4, October 2000, Pages 623-640, Science Direct

Forte Dario V., *The "ART" Of Log Correlation: Part 1: Tools And Techniques For Correlating Events And Log Files*, Computer Fraud & Security, Volume 2004, Issue 6, June 2004, Pages 7-11, Science Direct

Forte Dario V., , *The "ART" Of Log Correlation: Part 2: Tools And Techniques For Correlating Events And Log Files*, Computer Fraud & Security, Volume 2005, Science Direct

Burnett M., *The Art Of Tracking, Stealing The Network,* 2003, Pages 235-267, Science Direct

Will Schmied, Robert J. Shimonski, Thomas W. Shinder Dr., Tony Piltzecker, *Configuring And Using Auditing And The Event Logs MCSE/MCSA (Exam 70-214)* Study Guide, 2003, Pages 607-648, Science Direct

Jay Beale Members Of The Snort Team, Andrew R. Baker, Joel Esler, Stephen Northcutt, Toby Kohlenberg, Raven Alder, Dr. Everett F. (Skip) Carter Jr, James C. Foster, Matt Jonkman, Raffael Marty, Eric Seagren, *Exploring IDS Event Analysis*, Snort Style, Snort Intrusion Detection And Prevention Toolkit, 2006, Pages 411-497, Science Direct

Seagren E., *Managing Event Logs,Secure Your Network For Free*, 2006, Pages 263-336, Science Direct

Brian Wotring, Bruce Potter, *Log Monitoring And Response, Host Integrity Monitoring Using Osiris And Samhain*, 2005, Pages 307-326, Science Direct

Brian T. Contos CISSP, William P. Crowell, Colby Derodeff GCIA, GCNA, Dan Dunkel, Dr. Eric Cole, Regis Mckenna, *Log Collection, Physical And Logical Security Convergence*, 2007, Pages 289-318, Science Direct

HP Syslog Vulnerability, Network Security, Volume 1996, Issue 2, February 1996, Page 2, Science Direct

Forte, Dario; Power, Richard., *Guaranteeing Convergence In Security Management With Consolidated Log Management.* Computer Fraud & Security, Jul2008, Vol. 2008 Issue 7, P5-6, 2p; DOI: 10.1016/S1361-3723(08)70110-6; (AN 33530713), Ebsco

Shipley, Greg., *SIEM Tools Come Up Short*. Network World, 6/30/2008, Vol. 25 Issue 26, P30-40, 6p; (AN 33023727) , Ebsco

Tullett, Jon., *OSSIM*, For IT Security Professionals, Jun2004, P44-44, 0p, 1 Color; (AN 13615241) , Ebsco

Snyder, Joel., *Providing Clarity Is Just One New Feature Bringing Network Security Into Focus.* Information Security, Jun2007, Vol. 10 Issue 6, P43-49, 5p; (AN 25529194) , Ebsco

Rapor

Kent K., Souppaya M., (2006), *Guide to Computer Security Log Management*, National Institute of Standards and Technology Special Publication 800-92, Natl. Inst. Stand. Technol. Spec. Publ. 800-92, 72 pages (September 2006)

Pouget. F., Dacier. M., (2003), White Paper: *Alert Correlation: Review of the state of the art 1,* France Institut Eurecom

Gorge M., (2007), *Making sense of log management for security purposes – an approach to best practice log collection, analysis and management*, Computer Fraud & Security (*Mathieu Gorge is the Managing Director of Vigitrust – a security consultancy based in Ireland.)*

Symantec EMEA Internet Security Threat Report Trends for 2008 Volume XIV, Published April 2009
http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_emea_internet_security_threat_report_04-2009.en-us.pdf

www.nist.org - NIST-SP800-92 – The National Institute of Standards and Technology (US) Guide to Computer Security Log Management
http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf

http://enterprisenetworksandservers.com/monthly/art.php?1501 – "Log management is the missing security performance ingredient", by Drew Robb

Karen Kent, Murugiah Souppaya, *National Institute of Standards and Technology, Guide to Computer Security Log Management*, September 2006, , Computer Security Division, Information Technology Laboratory.

H. Debar, D.Curry, B. Feinstein,*The Intrusion Detection Message Exchange Format(IDMEF)* Network Working Group, March 2007 Request for Comments: 4765, IDMEF http://www.ietf.org/rfc/rfc4765.txt

Robert Fischer Matr.Nr. 0026899, *Motivations and Challenges in Designing a Distributed Log Management Framework*, Institut f¨ur Softwaretechnik und interaktive Systeme Information & Software Engineering Group der Technischen Universit¨at Wien, W¨urthgasse 12/4, 1190 Wien, April 2007

Tez

İncebacak, D., (2007*), Design and implementatıon of a secure and searchable audıt logging system, In Partıal Fulfillment Of The Requirements For The Degree Of Master Of Science In The Department Of Informatıon Systems*, Ortadoğu Teknik Üniversitesi

John Conti, G., (2006), Counterıng Network Level Denıal Of Informatıon Attacks Usıng Informatıon Vısualızatıon*, In Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the College of Computing,* Georgia Institute of Technology

Rrd aberrant behaviour, http://cricket.sourceforge.net/aberrant/rrd_hw.htm
[Ziyaret Tarihi: Aralık 2008]

Arpwatch: http://freequaos.host.sk/arpwatch/
[Ziyaret Tarihi: Aralık 2008]

http://passive.sourceforge.net/
[Ziyaret Tarihi: Ocak 2009]

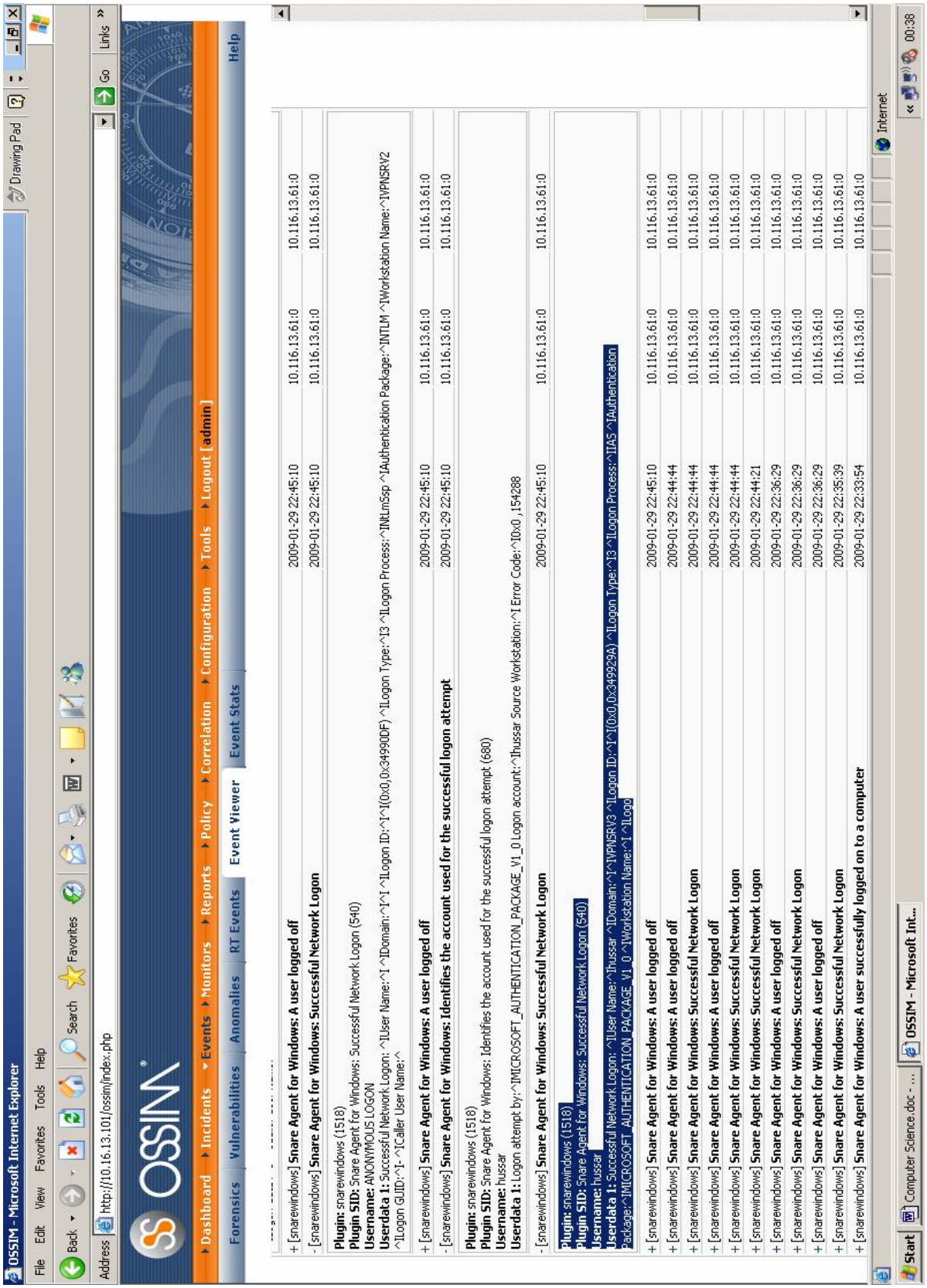
P0f, http://lcamtuf.coredump.cx/p0f.shtml
[Ziyaret Tarihi: Ekim 2008]

P0f, http://www.sans.org/resources/idfaq/p0f.php
[Ziyaret Tarihi: Kasım 2008]

http://citeseerx.ist.psu.edu/
[Ziyaret Tarihi: Şubat 2009]

EKLER

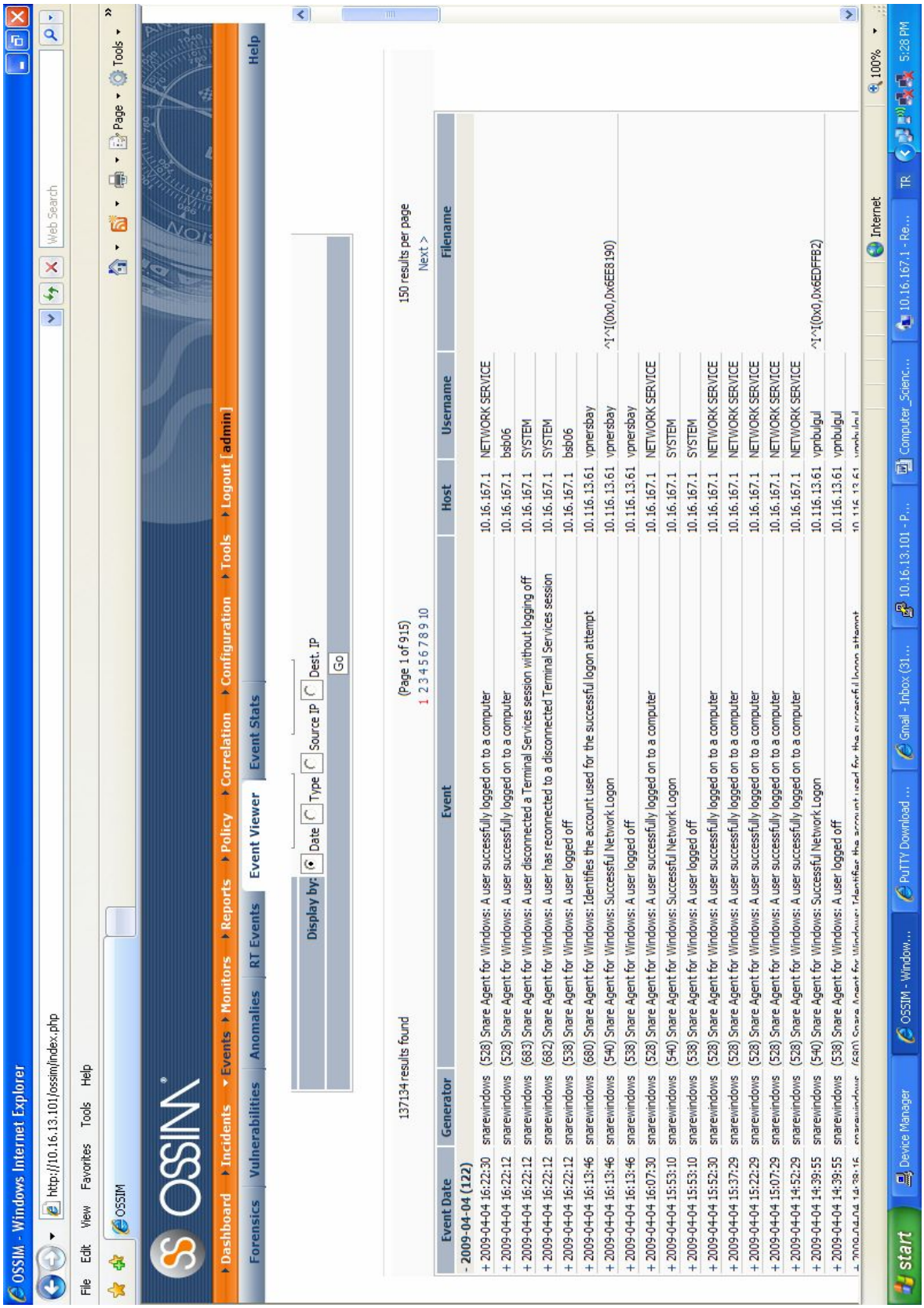
Ek 1-Ossim Olay Gözlem Ekranı - Windows Sunucular



Ek 2-Ossim Olay Gözlem Ekranı Oracle Kullanıcı Aktiviteleri

Ek 3-Performans Ölçütleri Dikkate Alınarak Ossim Sunucusu Tarafından Dinlenen Network Trafiği İçin, Network Switch Üzerinde Mirror Port Uygulaması

Ek 4-Snare Agent İle Ossim Sunucusuna Rotasyon Yapılmış Loglar



Ek 5-Ossim Veritabanı

Ek 6-Ossim Syslog-Ng Konfigürasyonu

```
destination df_daemon { file("/var/log/daemon.log"); };
destination df_kern { file("/var/log/kern.log"); };
destination df_lpr { file("/var/log/lpr.log"); };
destination df_mail { file("/var/log/mail.log"); };
destination df_user { file("/var/log/user.log"); };
destination df_uucp { file("/var/log/uucp.log"); };
# these files are meant for the mail system log files
# and provide re-usable destinations for {mail,cron,...}.info,
# {mail,cron,...}.notice, etc.
destination df_facility_dot_info { file("/var/log/$FACILITY.info"); };
destination df_facility_dot_notice { file("/var/log/$FACILITY.notice"); };
destination df_facility_dot_warn { file("/var/log/$FACILITY.warn"); };
destination df_facility_dot_err { file("/var/log/$FACILITY.err"); };
destination df_facility_dot_crit { file("/var/log/$FACILITY.crit"); };
# these files are meant for the news system, and are kept separated
# because they should be owned by "news" instead of "root"
destination df_news_dot_notice { file("/var/log/news/news.notice" owner("news")); };
destination df_news_dot_err { file("/var/log/news/news.err" owner("news")); };
destination df_news_dot_crit { file("/var/log/news/news.crit" owner("news")); };
# some more classical and useful files found in standard syslog configurations
destination df_debug { file("/var/log/debug"); };
destination df_messages { file("/var/log/messages"); };
# pipes
# a console to view log messages under X
destination dp_xconsole { pipe("/dev/xconsole"); };
# consoles
# this will send messages to everyone logged in
destination du_all { usertty("*"); };
# filters
# all messages from the auth and authpriv facilities
filter f_auth { facility(auth, authpriv); };
# all messages except from the auth and authpriv facilities
filter f_syslog { not facility(auth, authpriv); };
# respectively: messages from the cron, daemon, kern, lpr, mail, news, user,
# and uucp facilities
filter f_cron { facility(cron); };
filter f_daemon { facility(daemon); };
```

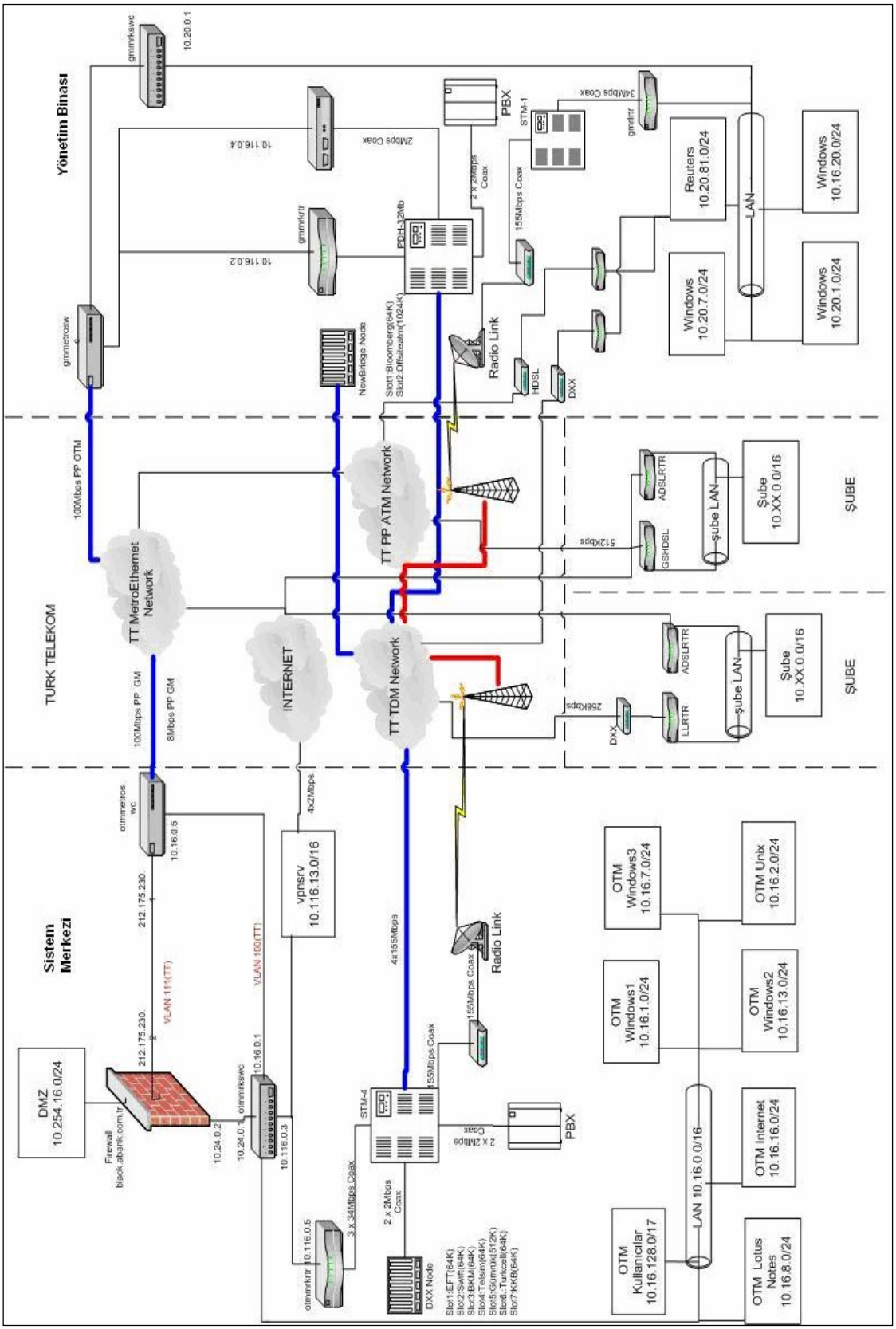
```
filter f_kern { facility(kern); };
filter f_lpr { facility(lpr); };
filter f_mail { facility(mail); };
filter f_news { facility(news); };
filter f_user { facility(user); };
filter f_uucp { facility(uucp); };
# some filters to select messages of priority greater or equal to info, warn, and err
# (equivalents of syslogd's *.info, *.warn, and *.err)
filter f_at_least_info { level(info..emerg); };
filter f_at_least_notice { level(notice..emerg); };
filter f_at_least_warn { level(warn..emerg); };
filter f_at_least_err { level(err..emerg); };
filter f_at_least_crit { level(crit..emerg); };
# all messages of priority debug not coming from the auth, authpriv, news, and mail facilities
filter f_debug { level(debug) and not facility(auth, authpriv, news, mail); };
# all messages of info, notice, or warn priority not coming form the auth,
# authpriv, cron, daemon, mail, and news facilities
filter f_messages {level(info,notice,warn) and not facility(auth,authpriv,cron,daemon,mail,news);};

# messages with priority emerg
filter f_emerg { level(emerg); };

# complex filter for messages usually sent to the xconsole
filter f_xconsole {facility(daemon,mail) or level(debug,info,notice,warn)
or (facility(news) and level(crit,err,notice));};
# logs
# order matters if you use "flags(final);" to mark the end of processing in a
# "log" statement

# these rules provide the same behavior as the commented original syslogd rules

# auth,authpriv.*                /var/log/auth.log
log { source(s_all);
filter(f_auth);
destination(df_auth); };

# *.*;auth,authpriv.none          -/var/log/syslog
log { source(s_all);
filter(f_syslog);
```

```
destination(df_syslog);};
# this is commented out in the default syslog.conf
# cron.*                        /var/log/cron.log
#log { #       source(s_all); #       filter(f_cron);
#       destination(df_cron);  #};

# daemon.*                      -/var/log/daemon.log
log { source(s_all);
filter(f_daemon);
destination(df_daemon); };

# kern.*                        -/var/log/kern.log
log { source(s_all); filter(f_kern);
destination(df_kern); };

# lpr.*                         -/var/log/lpr.log
log { source(s_all); filter(f_lpr);
destination(df_lpr); };

# mail.*                        -/var/log/mail.log
log { source(s_all); filter(f_mail);
destination(df_mail); };

# user.*                        -/var/log/user.log
log { source(s_all); filter(f_user);
destination(df_user); };

# uucp.*                        /var/log/uucp.log
log { source(s_all);
filter(f_uucp);
destination(df_uucp); };

# mail.info                     -/var/log/mail.info
log { source(s_all);
filter(f_mail);
filter(f_at_least_info);
destination(df_facility_dot_info); };
# mail.warn                     -/var/log/mail.warn
log { source(s_all);
```

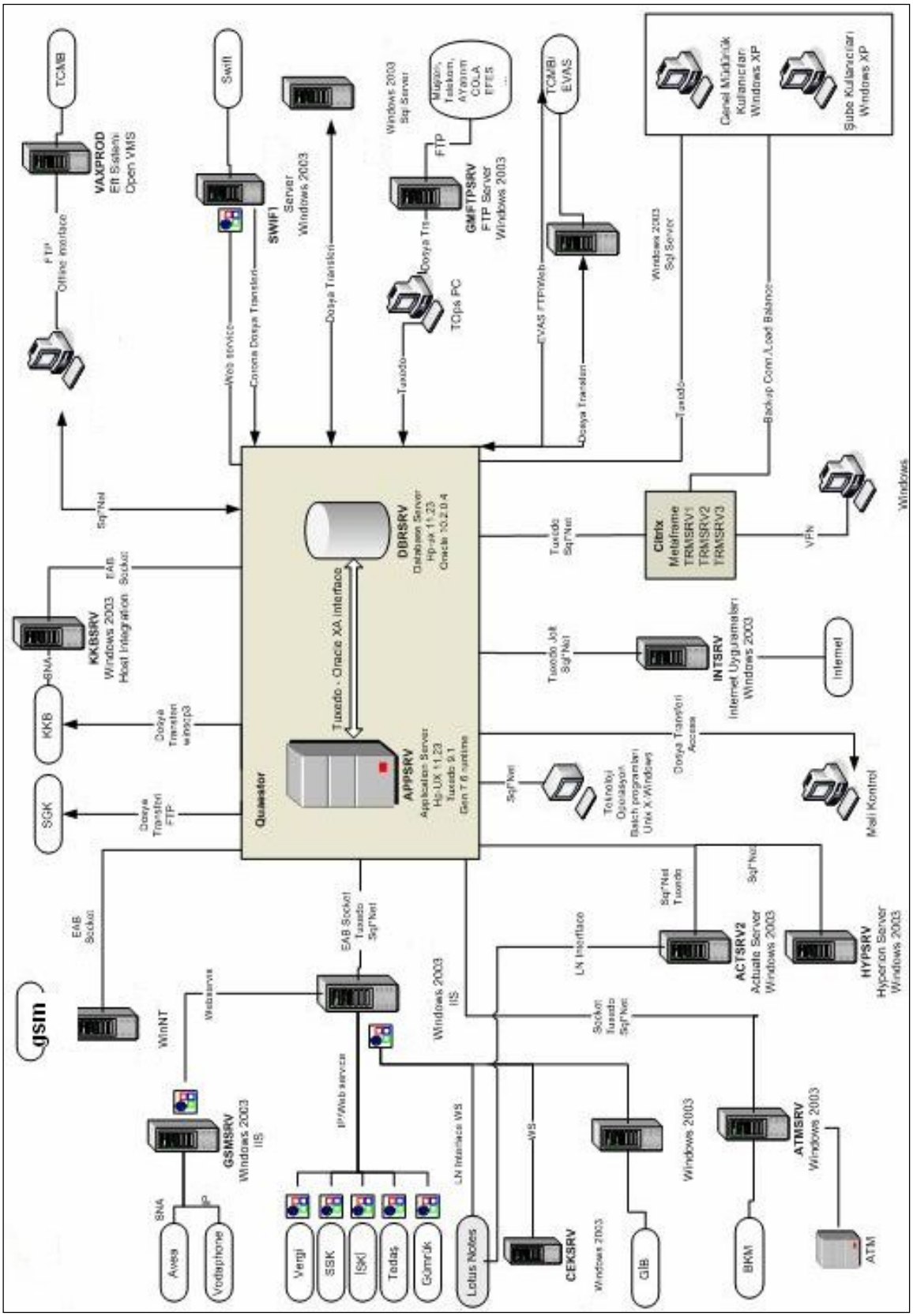
```
filter(f_mail);
filter(f_at_least_warn);
destination(df_facility_dot_warn); };

# mail.err                    /var/log/mail.err
log { source(s_all);
filter(f_mail);
filter(f_at_least_err);
destination(df_facility_dot_err); };

# news.crit                   /var/log/news/news.crit
log { source(s_all);
filter(f_news);
filter(f_at_least_crit);
destination(df_news_dot_crit); };

# news.err                    /var/log/news/news.err
log { source(s_all);
filter(f_news);
filter(f_at_least_err);
destination(df_news_dot_err); };

# news.notice                 /var/log/news/news.notice
log { source(s_all);
filter(f_news);
filter(f_at_least_notice);
destination(df_news_dot_notice); };

# *.=debug;\
#        auth,authpriv.none;\
#        news.none;mail.none     -/var/log/debug
log { source(s_all);
filter(f_debug);
destination(df_debug); };

# *.=info;*.=notice;*.=warn;\
#        auth,authpriv.none;\
#        cron,daemon.none;\
#        mail,news.none          -/var/log/messages
```

```
log { source(s_all);
filter(f_messages);
destination(df_messages); };
# *.emerg                       *
log { source(s_all);
filter(f_emerg);
destination(du_all); };

# daemon.*;mail.*;\
#       news.crit;news.err;news.notice;\
#       *.=debug;*.=info;\
#       *.=notice;*.=warn       |/dev/xconsole
log { source(s_all);
filter(f_xconsole);
destination(dp_xconsole); };

ossimpro:/etc/syslog-ng#
```

Ek 7-Kurum Network Topolojisi



Ek 8-Kurum Bilgi Sistemleri Kritik Sistem Altyapısı



Ek 9-Ossim tarafından kullanılan Zayıflık Tarama Aracı Nessus Çıktı Örneği

Vulnerability found on port general/tcp

You are running a version of Nessus which is not configured to receive
a full plugin feed. As a result, the security audit of the remote host produced
incomplete results.

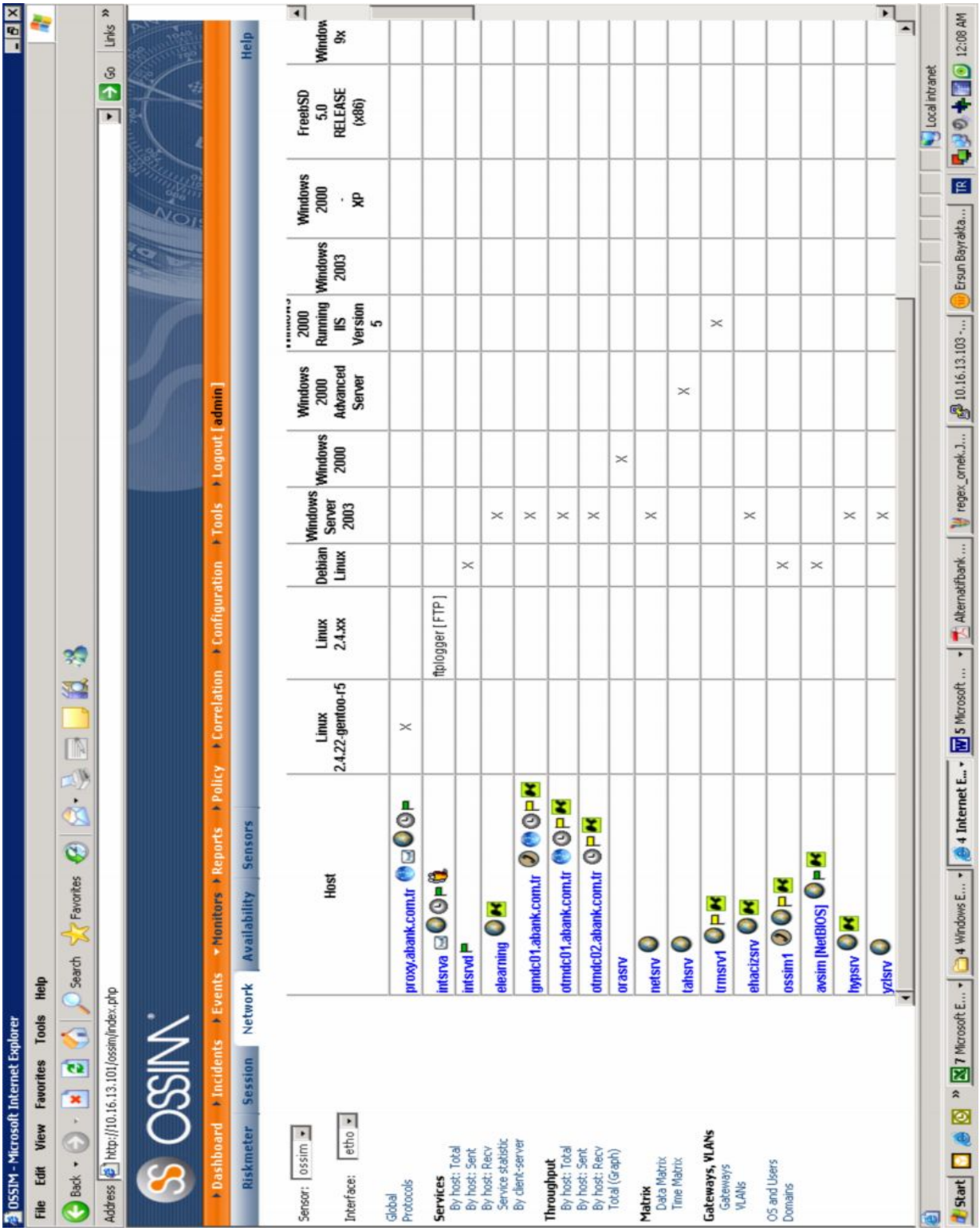
To obtain a complete plugin feed, you need to register your Nessus scanner
at http://www.nessus.org/register/ then run nessus-update-plugins to get
the full list of Nessus plugins.

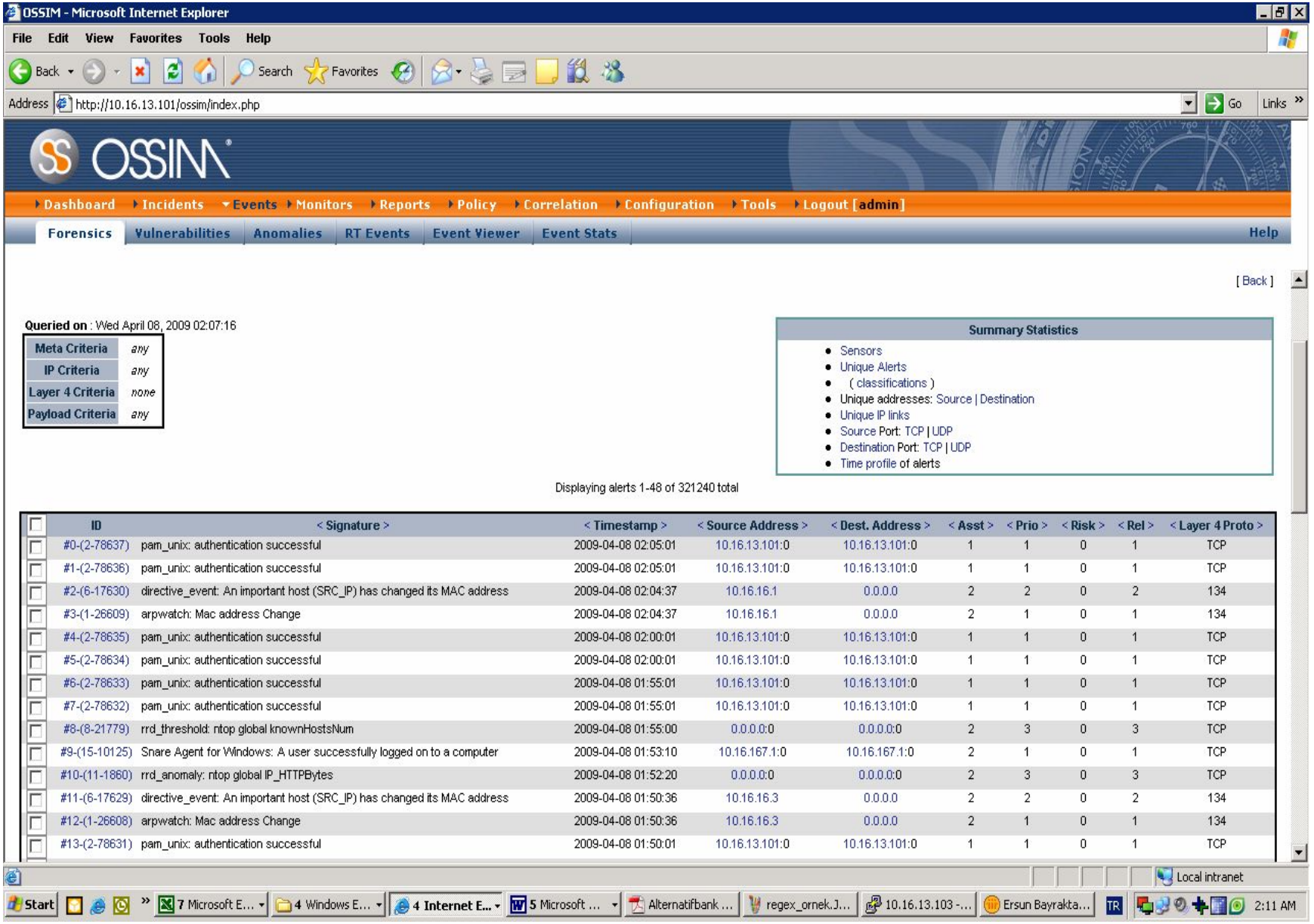
Information found on port general/tcp

Information about this scan :

Nessus version : 2.2.8
Plugin feed version : 200605221015
Type of plugin feed : GPL only
Scanner IP : 10.16.13.101
Port scanner(s) : nessus_tcp_scanner
Port range : 1-15000
Thorough tests : no
Experimental tests : no
Paranoia level : 1
Report Verbosity : 1
Safe checks : no
Max hosts : 20
Max checks : 4
Nessus ID : 19506

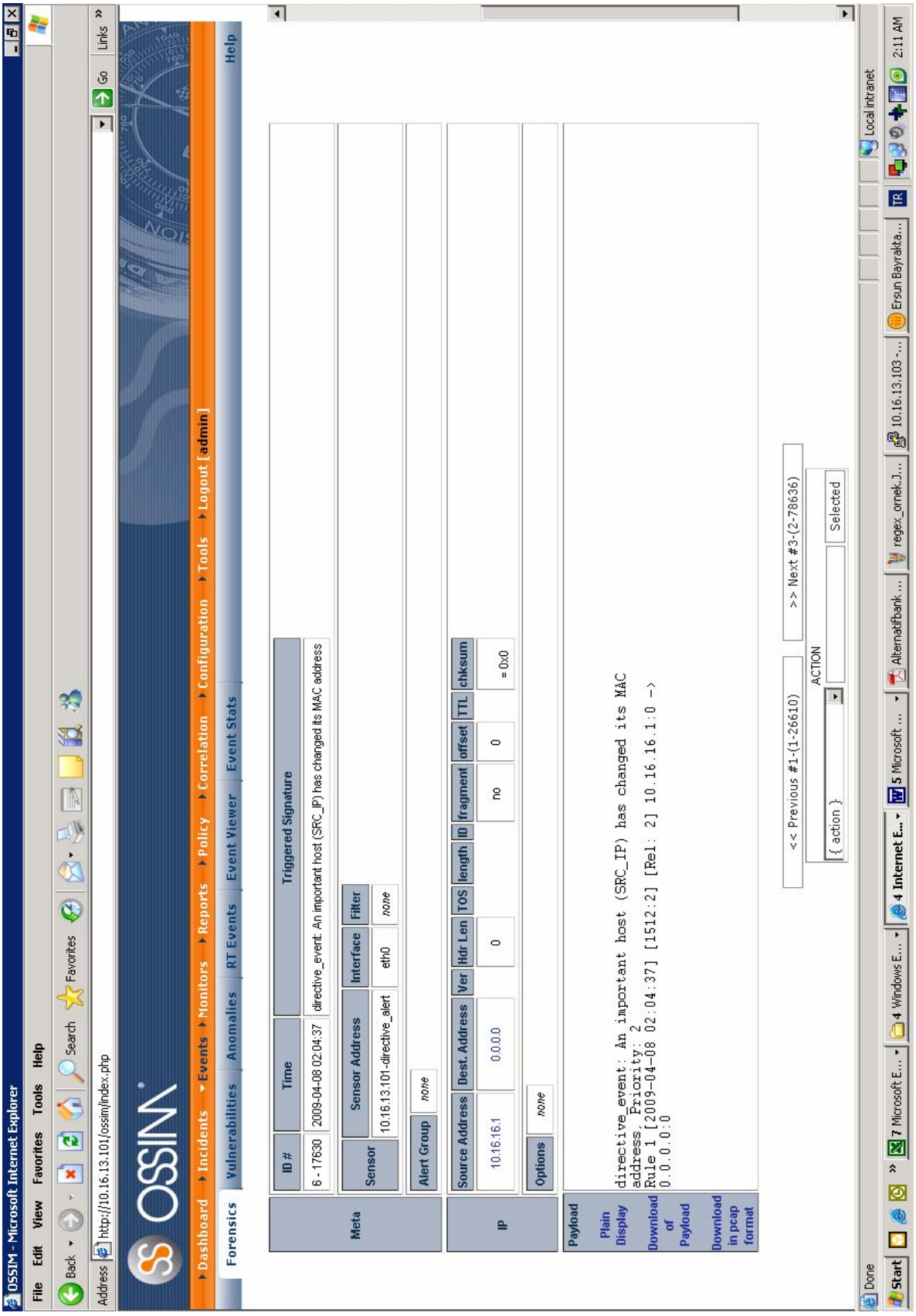
Ek 10-P0f Pasif Network Tarama Aracı Ossim Çıktısı (İşletim Sistemleri)





Ek 11-Ossim ACID Forensik Konsol

Ek 12-Ossim ACID Forensik Konsol Detay



Ek 13-Nagios Servis İzleme Ajanı Konfigürasyonu (Yüklenen Host Üzerinde)

```
.....
; Check other hosts through NRPE extreme beta and probably a bit dangerous! :)
;NRPEClient.dll
; Extreamly early beta of a task-schedule checker
;CheckTaskSched.dll

[Settings]
;# OBFUSCATED PASSWORD
;  This is the same as the password option but here you can store the password in an obfuscated manner.
;  *NOTICE* obfuscation is *NOT* the same as encryption, someone with access to this file can still figure out the
;  password. Its just a bit harder to do it at first glance.
;obfuscated_password=Jw0KAUUdXlAAUwASDAAB
;
;# PASSWORD
;  This is the password (-s) that is required to access NSClient remotely. If you leave this blank everyone will be able to access the daemon remotly.
;password=secret-password
;
;# ALLOWED HOST ADDRESSES
;  The syntax is host or ip/mask so 192.168.0.0/24 will allow anyone on that subnet access
allowed_hosts=10.16.13.101
;
;# USE THIS FILE
;  Use the INI file as opposed to the registry if this is 0 and the use_reg in the registry is set to 1
;  the registry will be used instead.
use_file=1
;
; # USE SHARED MEMORY CHANNELS
;  This is the "new" way for using the system tray based on an IPC framework on top shared memmory channels and events.
;  It is brand new and (probably has bugs) so dont enable this unless for testing!
;  If set to 1 shared channels will be created and system tray icons created and such and such...
shared_session=1
```

Ek 14- Host tarafına NSClient ajanı kurulduktan sonra, Ossim sunucusu üzerindeki nagios konfigurasyonlarında hangi servislerin ekleneceği belirlenmesi

Nagios Client konfigürasyonlarında komut satırından yapılan işlemler:

C:\Documents and Settings*computer name*>net start NSClientpp

The NSClientpp (Nagios) 0.3.6.316 2009-02-04 w32 service is starting.

The NSClientpp (Nagios) 0.3.6.316 2009-02-04 w32 service was started successfully.

C:\Documents and Settings*computer name*>netstat -an | find /i ":5666

TCP 0.0.0.0:5666 0.0.0.0:0 LISTENING

```
vim /etc/nagios2/conf.d/ossim-configs/hosts/10.16.167.1.cfg

define host{
host_name 10.16.167.1
alias ErsunDesktop
address 10.16.167.1
use generic-host
}

define service{
use                 generic-service
host_name           10.16.167.1
service_description    CPU Load
check_command       check_nt!CPULOAD!-l 5,80,90
}


define service{
use                 generic-service
host_name           10.16.167.1
service_description    Memory Usage
check_command       check_nt!MEMUSE!-w 80 -c 90
}
```

Ek 15- Ossim Konfigürasyonuna Eklentiler (Plugins)

Oracle test

plugin_id: 9001

[DEFAULT]

plugin_id=9001

[config]

type=detector

enable=yes

source=log

location=/var/log/syslog

create log file if it does not exists,

otherwise stop processing this plugin

create_file=false

process=syslogd ; -r || -u

start=yes ; launch plugin process when agent starts

stop=no ; shutdown plugin process when agent stops

startup=/etc/init.d/sysklogd start

shutdown=/etc/init.d/sysklogd stop

restart=yes ; restart plugin process after each interval

restart_interval=_CFG(watchdog,restart_interval) ; interval between each restart

list of sids (comma separated) to be excluded by the detector

#exclude_sids=200,302,404,403

[translation]

- = 0.0.0.0

. = 0.0.0.0

```
#
#
#
[oracle-log1]
#ERTCAV;15092008
01:14:37;QSTPRD;BSB12;MRKDMN\BSB12;;XRATE;P_QSTPRD_40417_ERTCA
V;"SELECT ""XRATE_DATE"" ,""XRATE_TIME"" ,""XRATE_TYPE""
,""CREATE_DATE"" ,""CREATE_TIME"" ,""XRATE_BUY""
,""XRATE_MIDDLE"" ,""XRATE_SELL"" ,""QUOTATION_UNIT"" ,""STATUS""
,""LAST_USERID"" ,""FK_CURRENCY"" ,""PARITY""  FROM
""ACTPRD"".""XRATE"" WHERE ((((""XRATE_DATE"" = TO_DATE
(:""SYS_B_0"",:""SYS_B_1"") ) AND NOT((""XRATE_TYPE"" = :""SYS_B_2"" ) ) )
AND NOT((""XRATE_TYPE"" = :""SYS_B_3"" ) ) ) AND NOT((""XRATE_TYPE""
= :""SYS_B_4"" ) ) )"

#09/15/08      NIMULU  BSB19   QSTPRD  RELATION       select * from relation

event_type=event
regexp=(?P<querydate>\d\d/\d\d/\d\d)\t(?P<dbuser>.+)\t.+\t(?P<db>\w+)\t(?P<table>\w
+)\t(?P<query>.+(INSERT).*)

#(?P<timestamp>\d{8}\s(?P<querydate>\d{1,2}:\d{1,2}:\d{1,2}).+;.+;.+;.+;.+;.+;(.*
#src_ip={resolv($host)}
#src_ip={$host}
# A lot of warnings with failed translates.
#dst_ip={resolv({translate($dst_ip)})}
#dst_ip={resolv($dst_ip)}
#dst_ip={$dst_ip}
plugin_sid=1
#log={$url}
userdata1={$querydate}
```

```
userdata2={$dbuser}
userdata3={$db}
userdata4={$table}
userdata5={$query}

[oracle-log2]
#09/15/08      NIMULU  BSB19   QSTPRD  RELATION        select * from relation

event_type=event
regexp=(?P<querydate>\d\d/\d\d/\d\d)\t(?P<dbuser>.+)\t.+\t(?P<db>\w+)\t(?P<table>\w
+)\t(?P<query>.+(SELECT).*)

#(?P<timestamp>\d{8}\s(?P<querydate>\d{1,2}:\d{1,2}:\d{1,2}).+;.+;.+;.+;.+;.+;(.*
#src_ip={resolv($host)}
#src_ip={$host}
# A lot of warnings with failed translates.
#dst_ip={resolv({translate($dst_ip)})}
#dst_ip={resolv($dst_ip)}
#dst_ip={$dst_ip}
plugin_sid=2
#log={$url}
userdata1={$querydate}
userdata2={$dbuser}
userdata3={$db}
userdata4={$table}
userdata5={$query}
```

EK 16-Tanımlamalar

IDMF: Bu format Modeli XML olarak merkezileştirilmiş ve alarm modellerini sınıflandırmıştır.

VPN: (Ortak kullanılan bir ağ üzerinde, sanal olarak diğer iletişim gruplarından kendi arasındaki iletişim şifreleme veya tanımlama yolu ile izole edilen ağ) Hacker: Bilgisayar yazılımı ve sistem uzmanı olan, uzmanlık edinmiş iyi veya kötü niyetli kişiler.

Mac Adresi: Bir bilgisayar ağına bağlanmak üzere kullanılan ağ kartlarının her biri için, üretim sırasında kaydedilen ve bir eşi olmayan bir numara verilir. Media access control address (MAC) olarak adlandırılan bu adresin uzunluğu 48 bit'tir. Tipik bir MAC adresi 00-50-05-1A-00-AF'dir. Ağ kartları bir diğer ağ kartına veri yollarken alıcıyı diğerlerinden ayırmak için bu MAC adresini kullanır. Buna göre ilk 6 rakam yani 00-50-05 üretici kodu, son 6 rakam ise bu kartın seri numarasıdır. MAC adresi bütün olarak değerlendirildiğinde dünyada üretilen her ağ kartı farklı bir MAC adresine sahip demektir. Mac adresi yazılımlar ile değiştirilebilir.

Ethernet: Bilgisayarlar arasındaki sinyal paketlerini, günün teknolojisinde kullanılan çeşitli kablo tipleri ile 10 Mbps (Milyon bit/saniye) ile N Gbps hızında taşıyan, CSMA/CD (çarpışma algılama) kullanan yaygın lokal network teknolojisi.

ARP: RFC 826 dosyasında tanımlanmış olan ARP (İngilizce Address Resolution Protocol, yani Adres Çözümleme Protokolü), bir TCP/IP ağında IP adresleri ile MAC adresleri arasındaki bağı yapmak için kullanılır. Bir makine diğer bir makinenin MAC adresini öğrendiğinde bunu daha sonra sorma ihtiyacı duymamak için ARP önbelleğine koyar, Windows, Linux ve MacOS'ta bu liste arp -a komutu ile görülebilir. > arp -a Interface 192.168.1.17 --- 0x20005 Internet Address Physical Address Type 192.168.1.1 00-07-3a-b3-22-3c dynamic

NTP Zaman Protokolü (Network Time Protocol): Bir bilgisayarın uzaktaki bir sunucu ile yerel zaman ve tarihin eşleşmesi için kullanılan protokoldür. Böylece yerel bilgisayar saati LAN üzerinde milisaniye, WAN üzerinde birkaç ondalık milisaniyeler

doğruluğunda çalışır. Bu sunucular UTC (Universal Coordinated Time) ile senkronize edilmektedir.
http://www.eecis.udel.edu/~mills/ntp.html
http://www.ntp.org/

DHCP: (İngilizce ***D**ynamic **H**ost **C**onfiguration **P**rotocol*): Bir TCP/IP ağındaki makinelere IP adresi, ağ geçidi veya DNS sunucusu gibi ayarların otomatik olarak yapılması için kullanılır. Günümüzde neredeyse tüm ev ve halka açık ağlarda kullanılmaktadır, ofis veya daha kontrollü bir bağlantı sağlanan yerlerde ise statik IP adresi tercih edilir.

Syslog: Unix işletim sistemleri tarafından standart olarak sunulan bir yazılımdır. Bu yazılım sistemde sürekli olarak çalışarak (servis=daemon) log tutma işlemini gerçekleştirir. Bu sayede sistem üzerinde çalışan birçok yazılımın verdiği hata ve/veya bilgi mesajları her zaman aynı yere yazılır. Sistem yöneticisinin işini bir ölçüde kolaylaştıran bu yazılım aynı zamanda merkezi log tutma özelliğini de taşır. Yazılımların gönderdikleri mesajlar belli önem sıralarına göre ayrılarak farklı dosyalara yazılırlar. Bu sayede sistemin kritik hataları anında algılanabilir. Linux sistemin log servisi sistemin önemli bir parçasıdır. Sistem sorunlarını önlemek ve çözmek için bir sistem yöneticisi bu logları analize edebilmelidir. Sistem loglarını konfigüre etmek ve log dosyalarını yönetebilinmelidir, özellikle sistem çok yüklenmeli çalışıyor ise bu dosyalar çok büyümektedir. Bu dosyalar sadece makul bir zaman için tutulmalıdır, bu da dosyaları incelemekte yardımcı olacaktır.Red Hat Linux sisteminde iki sistem log süreci bulunmaktadır:

1. syslogd – Sistemin temel log süreci
2. klogd – Çekirdek loglarını tutan süreç

Belirtilmiş olduğu gibi syslogd sistemin temel loglarını tutan bir süreçtir. Bu süreç sistem açılışında /etc/rc.d/init.d/syslog script'i çalıştırılarak başlatılır. Önemli bir olay meydana geldiğinde ve yazılım bu olay hakkında log tutmak istediğinde, mesajını syslogd sürecine gönderir. Genel olarak loglar /var/log/messages dosyasına yazılır. Bunun dışında syslogd sürecin çeşitli log işlemleri vardır. Örneğin syslogd olay

loglarını farklı dosyalara yazabilmektedir, ya da sistem konsoluna gönderir, ya da o an sistemde bulunan kullanıcılara vb. **/etc/syslog.conf** dosyası syslogd sürecin konfigürasyon dosyasıdır. Çekirdek mesajlarını tutan diğer önemli süreç klogd'dir. Genel olarak bu süreç gelen mesajları direk syslogd sürecine aktarır. Ama istenildiğinde bu süreç komut satırından çalıştırılarak çekirdek mesajları belirtilmiş dosyaya yazdırılabilinmektedir.
Spade: http://www.sans.org/resources/idfaq/anomaly_detection.php

Phising: Bir web sayfasının bir kopyasını yapıp kullanıcının hesap bilgilerini çalmayı amaçlayan bir internet dolandırıcılığı.

COBIT: Control Objectives for Information and Related Technology, Bilişim sistemi ve teknoloji ile ilgili hedeflerin kontrolü amaçlı, ISACA isimli denetim kökenli bir grup tarafından kurulan vakıf tarafından oluşturulan yöntem dokümanıdır. Ülkemizde bankacılık alanındaki denetimlerde bu metod baz alınmaktadır.

ÖZGEÇMİŞ

Ad Soyad : Ersun Bayraktaroğlu

Doğum Tarihi : 01.01.1965

Doğum Yeri : Ankara

Yabancı Dili : İngilizce

Eğitim Durumu

Lisansüstü : Bahçeşehir Üniversitesi – Bilgi Teknolojileri 2006 – Devam Ediyor

Ankara Üniversitesi Fen Bilimleri Enstitüsü 1985-1988

Lisans : Ankara Üniversitesi Fen Fakültesi Jeoloji Mühendisliği 1981-1985

Lise : Ankara Yenişehir Lisesi 1978 - 1981

İş Deneyimleri : Belbim A.Ş. Yazılım 1989- 1994
Koç Allianz Sigorta A.Ş. Network ve Donanım 1994-2006
Alternatifbank A.Ş. 2006-

Kullanılan Programlar : Ansi C
Delphi
PL Sql
.Net Visual Basic
.Net Visual C#
Cobol