HALİÇ UNIVERSITY

INSTITUTE OF NATURAL SCIENCES

COMPUTER ENGINEERING DEPARTMENT

INTERNET SECURITY GATEWAYS AND THE APPLICATIONS

MASTER OF SCIENCE THESIS

Prepared by:

Murat OĞUZ

Supervisor:

Prof.Dr.Ali Okatan

İstanbul, April 2008

HALİÇ UNIVERSITY

INSTITUTE OF NATURAL SCIENCES

COMPUTER ENGINEERING DEPARTMENT

INTERNET SECURITY GATEWAYS AND THE APPLICATIONS

MASTER OF SCIENCE THESIS

Prepared by:

Murat OĞUZ

Supervisor:

Prof.Dr.Ali Okatan

İstanbul, April 2008

PREFACE

This report is the result of my master thesis project. This master thesis is also the last part of my Master of Science degree at Haliç University.

I would like to thank the following persons; Prof.Dr. Ali Okatan for being my guide during all my education, Asst.Prof.Dr.Yüksel Bal for being my supervisor during the project. Also thanks to Cenk Iscan for helping me to develop the concept code. Finally, I would like to thank my family for supporting me always.

TABLE OF CONTENTS

SUMMARY

Internet is growing day by day. There are millions hosts and all of them are connected with each other. From big companies, militaries, universities to end users at home are using Internet.

In this big environment, there are much security risks and attaks. Every day, new products like antivirus, antispam, firewalls etc. are being produced to defeat the attacks to the hosts. But the important question about security for from IT administrators to home users is which points should be thought to protect their systems and which types of risks are exist?

Security can be defined as “after finding and defining the risks, using the necessary tools to protect the systems from getting real these risks as an attack”. After applying the security products onto the production systems, it is not enough to protect always. Because, the security must be considered everyday and the systems must be improved by updating the existing systems or adding new ones.

The security risks exist in any systems, if the system has any value. So, the risks must be defined in the systems. After defining the risks, Email Servers, Web Servers, Database Servers and Other Application Servers should be thought for the known attacks types and their protecting ways. In addition, physical security and network security should also be analyzed.

This paper is aimed as a guide to show the most important points which the people should know and think to protect their host systems from the attacks at the internet.

ÖZET

Internet günden güne büyümektedir. İçerisinde milyonlarca host vardır ve hepsi birbirine bu yapı üzerinde bağlıdır. Şirketlerden askeri kurumlara, üniversitelerden ev kullanıcısına dek herkes interneti kullanmaktadır.

Bu büyük ortamda, çok fazla güvenlik açığı ve atak mevcuttur. Hergün, bu ataklara karşı koymak için, yeni virus tarayıcıları, güvenlik duvarları v.s. geliştirilmektedir. Fakat, bu ortamdaki tüm kullanıcılar için önemli güvenlik sorusu şudur;ne tür riskler mevcuttur ve korunmak için hangi noktaları düşünmek gerekmektedir?

Güvenlik şu şekilde tanımlanabilir. "Sistemler için varolan risklerin bulunması ve tanımlanmasından sonra, gerekli araçların kullanıcılarak öngörülen risklerin gerçek hayatta ortaya çıkmasını önlemeye yönelik çalışmalardır". Üretim sistemlerine güvenlik ürünlerinin kurulması, korumanın tamamen sağlandığı anlamına gelmez. Çünkü, güvenlik hergün düşünülmesi ve var olan sistemlerin güncellenmesi ve yenilerinin eklenmesiyle devam eden bir süreçtir.

Bir sistemin değeri varsa, güvenlik riskleri mevcut demektir. Dolayısıyla risk sistemlere bağlı olarak tanımlanmalıdır. Riskler tanımlandıktan sonra, elektronik posta sunucuları, web sunucuları, veritabanı sunucuları ve uygulama sunucularının var olan ilgili atak tiplerine karşı korunma yollarının düşünülmesi gerekmektedir. Ek olarak, fiziksel güvenlik ve ağ güvenliğininde incelenmesi gereklidir.

Bu çalışma, internet üzerinde bilinen en yaygın ataklara karşı sistemlerini korumayı amaçlayan ilgili kişilerin düşünmesi gereken temel noktaları vurgulamayı amaçlamaktadır.

1. SECURITY BASICS

1.1. An Overview of Threat and Risk Assessment

There are many methodologies that exist today on how to perform a risk and threat assessment. There are some that are "open-source" and those that are proprietary; however, they all try to answer the following questions.

- What needs to be protected?
- Who/What are the threats and vulnerabilities?
- What are the implications if they were damaged or lost?
- What is the value to the organization?
- What can be done to minimize exposure to the loss or damage?

The outcome or objective of a threat and risk assessment is to provide recommendations that maximize the protection of confidentiality, integrity and availability while still providing functionality and usability. In order to best determine the answers to these questions a company or organization can perform a threat and risk assessment. This can be accomplished using either internal or external resources. It is important that the risk assessment be a collaborative process, without the involvement of the various organizational levels the assessment can lead to a costly and ineffective security measure.

The choice between using internal or external resources will depend on the situation at the time. The urgency of the assessment will also help in determining whether to outsource or use internal resources. The external resource should not have a vested interest in the organization and "be free from personal and external constraints which may impair his or her independence."

The core areas in a risk assessment are:

- Scope
- Data Collection

- Analysis of Policies and Procedures
- Threat Analysis
- Vulnerability Analysis
- Correlation and assessment of Risk Acceptability

1.2. Scope

Identifying the scope is probably the most important step in the process. The scope provides the analyst with what is covered and what is not covered in the assessment. It identifies what needs to be protected, the sensitivity of what is being protected.

The scope will also identify what systems and applications are included in the assessment. When investigating and determining the scope keep in mind the intended audience of the final recommendations (i.e. senior management, IT department or certifying authority). The scope should indicate the perspective from which the analysis will take place, whether it is from an internal or external perspective or both. The level of detail is directly related to the intended recipient of the final analysis.

1.3. Collecting data

This step involves collecting all policies and procedures currently in place and identifying those that are missing or undocumented. Interviews with key personnel can be conducted using questionnaires or surveys to assist in identifying assets and missing or out-of-date documentation. The systems or applications identified in the scope are enumerated and all relevant information gathered on the current state of those systems.

- Service pack levels
- Port scanning
- Services running
- Wireless leakage
- Operating system type

- Intrusion detection testing
- Network applications running
- Phone systems testing
- Physical location of the systems
- Firewall testing
- Access control permissions.
- Network Surveying

1.4. Analyze the policies and procedures

The review and analysis of the existing policies and procedures is done to gauge the compliance level within the organization. Sources for policy compliance that can be used as a base line are:

- ISO 17799
- BSI 7799
- Common Criteria – ISO 15504

It is important to identify the portions that are deemed not to be in compliance with respect to the specific industry and organization. Care must be taken not to determine. Because so many security standards exist, it is often difficult to determine which best applies to the organization. Generic standards offer the most comprehensive view, but these often require security measures that are inappropriate in one or another industry. They fail to take into account the context.

1.5. Vulnerability Analysis

The purpose of vulnerability analysis is to take what was identified in the gathering of information and test to determine the current exposure, whether current safe guards are sufficient in terms of confidentiality, integrity or availability. It will also give an indication as to whether the proposed safe guards will be sufficient. Various tools can be used to identify specific vulnerabilities in systems.

The problem faced within many organizations is the ability to effectively filter out the false positives inherent in assessment applications. The result of the various tools must be verified in order to accurately determine the reliability of the tools in use and to avoid protecting an area that in reality does not exist. False positive results can be mitigated by ensuring that the assessment applications are up to date with the latest stable signatures and patches.

The vulnerability analysis phase also includes penetration testing with the objective of obtaining something of value, such as a text file, password file, classified document etc. It is important to note that this should be pre-determined with senior management. There are two classifications of penetration testing, testing with knowledge and testing with zero-knowledge. Zero-knowledge testing is usually conducted as an external penetration test, where the tester has no knowledge of the systems involved or network architecture, in effect simulating an external attack and compromise. In a knowledge penetration test the analyst assumes the role of an employee with basic rights and privileges and has access to basic knowledge regarding systems and network topology.

The specific vulnerabilities can be graded according to the level of risk that they pose to the organization, both internally and externally. A low rating can be applied to those vulnerabilities that are low in severity and low in exposure.

1.6. Threat Analysis

Threats are described as anything that would contribute to the tampering, destruction or interruption of any service or item of value. The analysis will look at every element of risk that could conceivably happen. These threats can be split into Human and Nonhuman elements. For example:

Human;

- Hackers
- Theft (electronically and physically)

- Non-technical staff (financial/accounting)
- Accidental
- Inadequately trained IT staff
- Backup operators
- Technicians, Electricians

Non-Human;

- Floods
- Lightning strikes
- Plumbing
- Viruses
- Fire
- Electrical
- Air (dust)
- Heat control

Threats that are identified must be looked at in relation to the business environment and what affect they will have on the organization. Threats go hand in hand with vulnerabilities and can be graded in a similar manner, measured in terms of motivation and capability. For example, the internal non-technical staff may have low motivation to do something malicious; however, they have a high level of capability due to their level of access on certain systems. A hacker, on the other hand, would have a high motivation for malicious intent and could have a high level of capability to damage or interrupt the business. It is important to note that motivation does not play a part in natural occurring phenomena. A low rating can be given where the threat has little or no capability or motivation. A high rating can be given for those threats that are highly capable and highly motivated.

The use of a grading system will assist greatly in the quantification of risk. The difficulty has always been in justifying the protection of assets. Management is better

able to understand the implications of the threat and vulnerabilities when they are quantifiable and measurable.

1.7. Analysis of acceptable risks

One of the final tasks is to assess whether or not the existing policies, procedures and protection items in place are adequate. If there are no safeguards in place providing adequate protection, it can be assumed that there are vulnerabilities. A review of the existing and planned safeguards should be performed to determine if the previously known and discovered risks and threats have been mitigated. It is not the job of the analyst to determine what an acceptable risk is to an organization. The analyst's role is to use the findings from the vulnerability and risk assessment to assist in determining, along with the parties involved, what level of risk is acceptable to the organization. The results are the basis for selecting appropriate security measures to be put in place or to remove those that are ineffective. Over-protection can introduce unnecessary costs and overhead. The level of protection required and maintainable will be different for every organization. Depending on the size of the IT department they may or may not be able to maintain the recommended safeguards. This needs to be taken into account in order to effectively recommend a product or procedure.

1.8. Current security threats, vulnerabilities, and security technologies.

In order to understand the IT and network security environment, and how best to deal with it, it is necessary to define some terms, and describe the kinds of threats and security solutions that exist today. This is not intended to be an exhaustive list, but rather a "plain english" description of the most common terms.

1.8.1. Vulnerabilities

Vulnerabilities are known (or newly found) security holes that exist in software. An example is a buffer overflow, which occurs when the developer of a software product expects a certain amount of data, for example 20 bytes of information, to be sent at a particular point in the operation of a program, but fails to allow for an error

condition where the user (or malicious attacker) sends a great deal more data, or unexpected (perhaps special) characters. Vulnerabilities can exist in software running on PC's, servers, communications equipment such as routers, or almost any device running software. Not all vulnerabilities are created equal- some will cause the program affected to crash (which can lead to a denial of service condition on the affected system), or cause a reboot, or in the worst case, they can allow the attacker to gain root or administrative access to the affected system. Upon discovery of a vulnerability, the software vendor will develop a fix, or software patch, and make it available to users of the software.

1.8.2. Exploits

When vulnerabilities are found in software, the hacker community will frequently attempt to develop attack code that takes advantage of the vulnerability. This attack software is called an exploit, and exploit code is frequently shared among hackers, as they attempt to develop different sophisticated attacks.

1.8.3. Threats or attacks

One useful way to categorize security threats or attacks is to look at the intent- a directed attack is one aimed at a single company- for example a company attempting to hack into a competitors network. A mass attack is usually a virus or worm, that is launched onto the Internet, and that replicates itself to as many systems as possible, as quickly as possible. Attacks may come from outside of a company, or a company insider may carry them out.

1.8.4. Viruses

Viruses are generally carried within e-mail messages, although they are anticipated to become a security problem for instant messaging traffic as well. Users unknowingly cause the virus to execute as a program on their system when they click on an attachment that runs the virus program. Virus writers go to great lengths to disguise the fact that the attachment is in fact a virus. They also attempt to spread by using all of the e-mail addresses that they can find on an infected system to send

themselves to. An example of a well know virus is the Bagle family of viruses (there have been many versions of this virus). These viruses contain their own e-mail server, so that they can replicate by sending email to all mail addresses that they harvest from the compromised system.

1.8.5. Worms

An example of a worm is the Blaster worm, which rapidly spread through the Internet in August 2003. Blaster targeted computers running Windows operating systems, and used a vulnerability in Remote Procedure Call (RPC) code. Blaster affected computers running Windows 2003 operating system, Windows NT 4.0, Windows NT 4.0 Terminal Services Edition, Windows 2000, and Windows XP. After compromising hundreds of thousands of systems, Blaster launched a distributed denial of service attack on a Microsoft Windows update site.

1.8.6. Trojan horses

As the name implies, these are software programs that are put onto target systems (whether by a direct hack, or as the result of a virus or worm) that have a malicious intent. The Trojan can capture passwords, or provide root access to the system remotely.

1.8.7. Denial of service attacks (DoS)

A denial of service attack attempts to put the target site out of operation, frequently by flooding the site with bogus traffic, thus making it unusable. The attacker attempting to create a denial of service condition will oftentimes try to compromise many PC's, and use them to "amplify" the attack volume, and to hide his or her tracks as well. This is called a Distributed Denial of Service Attack (DDoS). Denial of service attacks have now become a popular criminal activity. In an online form of the "protection racket" (pay us some protection money or we'll ruin your business), computer criminals have taken to using denial of service attack methods to put online businesses out of business, at least temporarily, and to then demand money from the target. This sort of cyber extortion attack has been used by

hacker rings operating out of Eastern Europe, and has caused significant disruptions to online bookmakers and gambling sites. They are estimated to have cost the industry upwards of £40m ($60-$70m)8. Any business that depends on online ordering for a significant portion of its revenues is susceptible to this sort of attack. Denial of Service attacks have also been used to try and put competitors out of business. In a case that surfaced in August, 2004, a satellite TV dealer hired hackers to mount DoS attacks on the websites of his 6 primary competitors, causing them over $2M in lost revenue9. Denial of service attacks are very hard to effectively protect against.

1.8.8. Spam

Spam is not a security threat, but spam techniques are increasingly being used to deliver malicious software. Spam can also be used to launch "phishing" attacks, which attempt to elicit confidential personal information (bank account information, credit card information, etc.) as a means to steal identity, or cause financial harm.

1.8.9. Routers

Routers are perhaps not generally thought of as "security solutions", however most routers today provide packet filtering capabilities, and they can be used to enhance the security of most networks. In addition, there are certain security tasks that are best performed on the router in order to optimize the performance of the overall network, and to reduce the processing load on a firewall.

1.8.10. Firewalls

Firewalls are a fundamental network security solution. Firewalls are used to restrict inbound and outbound network access to only traffic that is allowed by the security policy of the organization. For example, an organization that does not maintain a publicly accessible webserver on their company LAN can use a firewall to define and enforce a security policy that allows outbound web access for employees, but that blocks any inbound webserver access attempts (HTTP protocol, port 80 access) at the firewall.

1.8.11. Anti-virus software

Anti-virus (AV) software is used to scan e-mail messages looking for defined viruses, which show up as known signatures that the software recognizes as a virus. AV solutions can be implemented on each desktop, or they can be implemented as a gateway or e-mail server function, where all incoming messages are scanned before being delivered to the recipient. Best practices for preventing viruses on a corporate network call for both desktop and gateway or server AV to be implemented, to ensure that laptops that plug into the LAN canot corrupt systems "behind" the AV Gateway. It is important that both types of AV software are kept up-to-date, as new viruses are found on a very frequent basis.

1.8.12. Virtual Private Networks

The ubiquity and low cost of Internet connections have created a requirement to use the Internet for private company communications, replacing more expensive private networks (frame relay, and private line networks). Virtual Private Network (VPN) technology was developed to allow the Internet to be used in a private manner, with all data between company locations or endpoints being encrypted. VPN's provide privacy for the data while it is in transit across the Internet. VPN's do not secure endpoints from other sorts of attacks, however. And from a security standpoint, VPN's actually extend the corporate network to remote locations. The notion that the network is only as secure as it's weakest link is worth bearing in mind when implementing VPN's, as the weakest link may become the executive's home PC which has a VPN connection to headquarters, or the salesperson's laptop which is equipped with a VPN connection for remote access, or the business partner's LAN that is equipped with a VPN connection to allow sharing of information. Another way to think about this is to acknowledge that the actual network perimeter to be secured extends to all systems that are provided with VPN access- not just those on the local LAN.

1.8.13. Intrusion detection/prevention systems

Intrusion detection (IDS) and intrusion prevention (IPS) systems are products that can analyze certain types of traffic, and determine whether the traffic is legitimate traffic, or if the traffic matches a known pattern indicating that it is attack traffic. An example might be web (port 80) traffic, which a firewall would hypothetically be configured to allow. An IDS system can look at the traffic, and determine that the traffic is actually a NIMDA attack, and not valid user traffic, based upon the pattern. An IDS product will alert on invalid traffic, while an IPS product will block the offending traffic. IDS/IPS products come in two configurations- they are implemented either as a network device analyzing traffic on the local LAN segment, or they are software implemented on a specific host that looks at traffic on that host only.

1.8.14. Spam filtering

Spam filtering can be implemented on the e-mail server, or on a separate appliance sitting between the Internet and the mail server. There are many techniques that can be used to try and identify Spam, and generally the goal is to eliminate as much as possible false positives (legitimate mail misclassified as Spam), while also eliminating false negatives (Spam that slips past the Spam filter). A category of Spam that is more ominous than most is what are known as “phishing” attacks. These are generally mass messages that are cleverly crafted to look like legitimate mail from a bank or online merchant, that request the recipient to verify some confidential personal information, usually including account data. Unsuspecting victims who actually respond, and provide their personal information, oftentimes end up the victim of identity theft, or some sort of financial fraud. Implementing a Spam filter will help to improve the security posture of a company, and it will also help to improve the productivity of the company.

2. AUTHENTICATION

2.1. An Overview of Different Authentication Methods and Protocols

Authentication can be accomplished in many ways. The importance of selecting an environment appropriate Authentication Method is perhaps the most crucial decision in designing secure systems. Authentication protocols are capable of simply authenticating the connecting party or authenticating the connecting party as well as authenticating itself to the connecting party. This overview will generalize several Authentication Methods and Authentication Protocols in hopes of better understanding a few options that are available when designing a security system.

2.2. Passwords

Passwords are the most widely used form of authentication. Users provide an identifier, a typed in word or phrase or perhaps a token card, along with a password. In many systems the passwords, on the host itself, are not stored as plain text but are encrypted. Password authentication does not normally require complicated or robust hardware since authentication of this type is in general simple and does not require much processing power. Password authentication has several vulnerabilities, some of the more obvious are: Password may be easy to guess. Writing the password down and placing it in a highly visible area. Discovering passwords by eavesdropping or even social engineering. The risk of eavesdropping can be managed by using digests for authentication. The connecting party sends a value, typically a hash of the client IP address, time stamp, and additional secret information. Because this hash is unique for each accessed URI, no other documents can be accessed nor can it not be used from other IP address without detection. The password is also not vulnerable to eavesdropping because of the hashing. The system is, however, vulnerable to active attacks such as the-man-in-the middle attack.

2.2.1. One-time passwords

To avoid the problems associated with password reuse, one-time passwords were developed. There are two types of one-time passwords, a challenge-response password and a password list.

The challenge-response password responds with a challenge value after receiving a user identifier. The response is then calculated from either the response value (with some electronic device) or select from a table based on the challenge. A one-time password list makes use of lists of passwords which are sequentially used by the person wanting to access a system. The values are generated so that it is very hard to calculate the next value from the previously presented values. For example, the S/Key system calculates values xi starting from initial value R: x1=f(R), x2=f(f(R)), ..., xn=f(xn-1). The f() is chosen so that f-1 is very difficult. First the xn is used, then the xn-1 is used.

It is important to keep in mind that Password systems only authenticate the connecting party. It does not provide the connecting party with any method of authenticating the system they are accessing, so it is vulnerable to spoofing or a man-in-middle attack.

2.3. Public-key cryptography

Public key cryptography is based on very complex mathematical problems that require very specialized knowledge. Public key cryptography makes use of two keys, one private and the other public. The two keys are linked together by way of an extremely complex mathematical equation. The private key is used to decrypt and also to encrypt messages between the communicating machines. Both encryption and verification of signature is accomplished with the public key.

The advantage of public-key cryptography is that the public key is readily available to the public. In fact, public-keys are often published to public directories on the Internet so that they can be easily retrieved. This simplifies key-management efforts.

2.4. Zero-knowledge proofs

Zero-knowledge proofs make it possible for a Host to convince another Host to allow access without revealing any "secret information". The hosts involved in this form of authentication usually communicate several times to finalize authentication. The client will first create a random but difficult problem to solve and then solves it using information it has. The client then commits the solution using a bit-commitment scheme and then sends the problem and commitment to the server. The server then asks the client to either prove that the problems are related or open the committed solution and prove that it is the solution. The client complies with the request. Typically, about ten successful exchanges will be required to take place before the authentication process is complete and access is granted.

The zero-knowledge proof can be made to be non-interactively. In this instance only one message from client to server is needed. This method utilizes a one-way hash function where the committing answers are based on the output of that hash function. The number of proofs needed is generally larger (64 or more), to avoid brute-force attacks. The zero-knowledge proof of identity has it share of problems. Perhaps the most vulnerable one is that while Host A thinks he is proving his identity to Host B, it is possible for Host B to simultaneously authenticate to a third party, Host C, using Host A's credentials.

2.5. Digital Signatures

In many instances it is not necessary to authenticate communicating parties; for instance when downloading application updates or patches from the Internet. From a security pointof- view, the server does not need to screen who is downloading the software. The use downloading the software does not necessarily care what particular server it is downloading from. However, the user may want to be assured that the downloadable data is genuine and not a Trojan Horse or other malicious or invalid information. In this instance a digital signature would best serve to authenticate the downloadable data. A digital signature is a digest calculated from a signed document (typically a one-way hash function) which is then signed (encrypted with private

key). The client verifies the digest signature by decrypting it with the server's public key and compares it to the digest value calculated from the message received. The signature can also be used by the server to verify data the client is sending.

2.6. Widely used Authentication Protocols

In this section we will briefly examine some of the more commonly used protocols used to address security issues within open networks. Authentication is the first and most important line of defense in a system of trusted and open networks.

2.6.1. Secure Sockets Layer

Secure Sockets Layer (SSL), developed by Netscape Communications, provides a secure method of communication for TCP connections, especially for HTTP connections. SSL work in this manner: after a TCP connection is established, the client sends a client hello message to which the server responds with a server hello message. The hello messages establish connection attributes which include the protocol version, a session identifier, the cipher suite used, and the compression method in addition to random values for both the server and the client. After the hello messages are exchanged, the server will send its certificate. W hen the server has been authenticated, depending on the cipher suite used, the server may then request a certificate from the client. After receiving the client hello, the server instructs the client to start using encryption and finishes the initial handshake. The application transfer can now take place. When the client and the server decide to resume a previous session or duplicate an existing session, only the hello messages are exchanged. If the server does not find a matching session identifier, it will assume the connection is a new one. The advantage of resuming previous session is that it saves processing time, which may have a considerable effect on server performance.

2.6.2. IP SEC

The IP Authentication header provides strong authentication and integrity for IP datagrams. Depending on the signing algorithm used, it may also provide non-

repudiation, excluding those fields that are changed during transmit, like hop count or time to live. The authentication header has fields for the next header, payload length, security parameters index (SPI: identifies security association (SA) between two hosts), sequence number, and authentication data. The authentication is transport-protocol independent, so there may be data from more than one different protocol, for instance TCP and UPD. The authentication data is calculated with a message digest algorithm.

To avoid replay attacks, the 32-bit sequence number is not allowed to wrap around; one must establish a new SA and generate new keys. This happens once in 232 packets so, if 1460 byte TCP segments are transferred one can transfer 5.7 TB of data using one SA.

2.6.3. Secure Shell

Secure Shell (SSH) is a protocol for providing secure remote login and other secure network services over an insecure network. With SSH (version 2) each host has a host key, during the connection establishment the client can verify he is talking to the right server. The server keys can be stored locally on the clients or they may be distributed by using a key distribution protocol.

After a reliable byte stream is established between the client and the server, host authentication takes place using the transport layer functions. Both ends send version identification. The key exchange begins with both the client and server sending a key exchange initialization packet. The initialization packet contains a list of algorithms for key exchange, keys, encryption, MAC, and the level of compression supported. The server and client may negotiate a different set of algorithms for each direction of data flow. For each category, the best algorithm is chosen that both the client and server support.

2.6.4. Kerberos

Kerberos authentication was developed at the Massachusetts Institute of Technology (MIT). There are two main components: a ticket, which is used for user

authentication and securing data, and an authenticator that is used to verify that the user is the same user to whom the ticket was initially granted. When a user logs into a system, the system connects to the Kerberos server where it retrieves a session key to be used between the user and the ticket granting service (TGS). This is encrypted with a key based on the user's password. If the user provides the right password the end system is able to decrypt the session key. After this is done, the user password is erased from memory to avoid being compromise. The ticket (Ticket granting ticket: TGT) expires after a set amount of time.

When a user wants to connect to a service to which he does not already have a ticket, the user connects to the TGS and gets a ticket that can only be used to access the particular service the ticket was granted for. The user can now connect through an encrypted channel to the server. After the ticket expires, the user must request a new one from the TGS.

The major issue with Kerberos is its scalability. The Kerberos server must store secret keys for each of the users and each of the TGSs. Kerberos can get very complex in enterprise implementations where trust relationship need to be in place between multiple organizations.

3. DIGITAL CERTIFICATES

3.1. Overview

Asymmetric encryption, using private keys in combination with certificates, allows users to identify themselves over an electronic network, to communicate privately, and to sign electronic documents. These functions form the basis for e-commerce, and a system that exploits this technology is known as a Public Key Infrastructure (PKI).

3.2. Internet Business demands a PKI

Asymmetric encryption, using private keys together with certificates, allows users to identify themselves over an electronic network, to communicate privately and to sign electronic documents. The administration of - and ability to use - certificates and public and private keys, provides the enabling structure for e-transactions based on this concept. This underlying structure forms a Public Key Infrastructure (PKI).

It is almost impossible to establish a working Public Key Infrastructure without a common carrier of data that is easily accessible by the general public. Or, to put it another way, without a commonly accepted method of connecting computers there is simply no need for a PKI. This is probably why large scale PKI has not made greater strides already, despite the fact that the technology has been around for more than two decades. However, the increasing use of home based computers, the expansion of the Internet, and market exposure resulting from this “new” information transport system are now providing a catalyst for innovative ways of doing business. The virtual marketplace is much less expensive to invent, and faster to develop, than its physical counterpart. Formerly accepted laws of the market are, if not set aside completely, subject to disruption. It is, perhaps, inappropriate to compare the virtual marketplace with the physical one, but nevertheless they both offer companies space to do business. Cyberspace has made it possible for newcomers in various business segments to compete with well-established larger competitors.

The new market has put great pressure on organizations that are successful, comfortable and have a stable business in the traditional marketplace, to adapt and find viable ways of doing business in the virtual arena. Without going into further analysis of the potential winners and losers in Cyberspace, there's a common factor essential to success in the virtual marketplace - the act of non-repudiation, binding customers and businesses to contracts. Traditional methods of signing agreement orders, etc. must be reproduced electronically. PKI provides the means to do this. Without the ability to create legally binding contracts between remote parties electronic commerce will be unable to reach its full potential.

3.3. Security Services

The general purpose of a PKI is to enable security across networks and to provide the means to remotely identify a user and to establish methods, which imitate - and possibly improve - the written signature. There are four security services that must be in place before a viable e-business can evolve. These services are authentication, confidentiality, integrity and non-repudiation.

3.3.1. Authentication

Verifying that a user actually is who s/he claims to be. In the physical world, this is commonly accomplished by use of a passport, driving license or ID card. (in some countries a credit card is acceptable for this purpose, although without a photograph credit cards cannot provide true authentication). From an e-commerce perspective it must be possible to verify the identity of a user remotely.

3.3.2. Confidentiality

Confidentiality means ensuring that no one other than the expected parties is able to see an ongoing dialogue. In the physical world appropriate levels of confidentiality are assured by means such as voice control, choice of location, time, etc. In the virtual world it is more difficult to know who might be listening. Thus, in the virtual world, services which offer an assurance of confidentiality take on a more crucial role.

3.3.3. Integrity

This means ensuring that a message cannot be altered in any way during transmission. There has always been a demand for integrity when two or more remote parties need to rely on a given quantity of information. In the virtual world the traditional seal has been replaced by a digital signature.

3.3.4. Non-repudiation

Non-repudiation is the act of assuring the origin and/or issuance of a transaction or action. A physical agreement is likely to be produced on a paper document of some sort; most likely the date will be written on it prior to signing the document, and the procedure will be monitored by the other party, which will then also sign the document. This procedure is then repeated, setting up two identical agreements, or one party will get a copy, allowing it to claim verification in case of a dispute. In the virtual world it is equally necessary to create statements that, firstly, state an origin and secondly, can be verified at a later stage.

3.4. PKI setup and the major players

Remember what PKI stands for, and especially the last word, infrastructure. Once the PKI is established it should serve all kinds of e-commerce, or in other words, any electronic business transaction conducted over an electronic network, either public or enterprise-wide. Once the PKI is in place the end user will probably not give much thought to the new application provider. The security routines involved in determining trust, and the procedures involved in storing a trusted server CA certificate, will become as natural as determining trust before taking the decision to buy something in a physical shop. The PKI itself is the ground upon which e-business applications are built, and through which e-commerce transactions flow.

There are four key elements within a PKI.

3.4.1. The Certification Authority

The CA is the authority that issues and revokes certificates. Providing assurance that the certified information is correct, and that the key used for signing certificates and CRLs is not compromised, are among the responsibilities of the Certification Authority. Certification Authorities are bound by a number of other regulations too, but these are probably the most important ones. A PKI smart card-based medium for private key storage and operation requires secure routines, including secure transportation from the manufacturer or supplier to the CA, as well as from the CA to the end-user. A further requirement of such a medium is that the PKI smart cards used are sourced from a trusted manufacturer. As the issuing authority, the CA must provide a reliable operation of the certificate management system and assure delivery of CRLs at scheduled occasions. The CA organisation must provide for well-developed audit capabilities without increasing the risk of exposure or public purposes.

3.4.2. The Certificate Repository

It is the function of the repository to store certificate and CRL information (CRLs are lists of revoked certificates; the issuing CA digitally signs each list). As an end-user and, possibly more importantly, as an end-entity, access is required to the repository where the relevant CA has placed certificates and CRL's. This repository is the source of the latest status information for a given certificate. It is also the place to undertake a certificate search in cases where e-mail encryption to a specific user is desired, and there is therefore a requirement for the user's public key. Within an organization, this information would probably be stored in the internal address book. To serve as many end-users and end-entities as possible the repository must provide for good capacity throughput and, not least, provide a commonly accepted interface for the requester. A common interface, and well-used protocol, is LDAP (Lightweight Directory Access Protocol); while X.500 provides a common repository (database) structure. It is probable that the repository will cater for more than simply certificate information. The important factor about certificate

information is that it is signed by the CA, making it easy for the requestor to verify data integrity (given that the issuing CA is trusted).

3.4.3. The end-user

The end-user is typically someone using PKI enabled services over the Internet from a personal computer. This service could be a relatively new one, such as e-banking, or a well-established one, such as electronic mail. Outgoing mail may be encrypted by utilising the expected receiver's public key. Given the contents of a typically S/MIME structured message, the receiver can verify the signature of the sender. The same applies to other PKI enabled services, such as electronic banking from the home or e-shopping, although the structure of the signed message may be a different one, for instance PKCS#7. The structure of the actual message is not the relevant issue; the important thing is that it is possible to create a legally binding contract between the end-user and the service provider (endentity) and vice versa.

3.4.4. The Service Provider

The service provider is the typical application service point, be it a banking application, e-mail server or any other PKI enabled application. The server is likely to be connected to a back-end system, providing the actual application database etc. Although not explicitly shown in the figure, the end-entity would most probably be equipped with a firewall to protect it from unwanted attempts to access the server. Once the end-user and end-entity have authenticated themselves the confidentiality security service is initiated. All data transport between the end-user and end-entity takes place in an encrypted format from that point on, thus reassuring both parties that data transferal is confidential during transmission.

3.5. Central processes in a PKI

3.5.1. Issuing certificate

The CA issues certificates to end-users and end-entities according to defined CA-policies. By issuing an X.509 certificate the CA binds the certified public key to a specific end-user or endentity, thus logically also binding the private key. It is vital

that information within the certificate is correct, since the CA has signed this information and an independent third party may verify that the CA issued the certificate. The end-user or end-entity will use the certificate and the keys in combination with the certificate for authentication, and possibly signature operations. It is usual to provide certificates with different extensions that define the purpose of the certificate, such as authentication, confidentiality and non-repudiation.

A certificate is typically issued for a limited time period. This period will depend on the purpose of the certificate. A certificate which identifies a user as an individual may last for several years but certificates within a single sign-on system may last only for a limited period.

3.5.2. Revoking certificates

A certificate may be revoked whenever the private key of an end-user or end-entity is lost or if there is a suspicion that the private key has become exposed. This process normally takes place after the party owning the private key directs the CA to revoke the certificate. Revoked certificates are placed on a special list signed by the CA. This list is called a Certification Revocation List (CRL). The CRL will be distributed to a predefined and well-known place on a regular basis. Once a certificate is revoked there is no longer a bond between the former owner and the keypair. It will still be possible to determine the previous owner, but the binding period ends. Thus it is still possible to verify a signature that was made before the certificate was revoked, but new authentication services or signatures should not be accepted. The binding of a keypair with a given owner also expires when the certificate expires. This is a different issue and has nothing to do with revocation, although the practical consequences are likely to be the same, i.e., after expiration the keys will not be accepted for identification or nonrepudiation.

3.5.3. Authentication / Verification

By providing a challenge that requires a response, it is possible to authenticate each of the two parties involved. Either may prove ownership of a certificate by

responding to the challenge with an encrypted response. The response is encrypted with the end-user's or end-entity's private key. The challenging party may decrypt the response using the public key within the certificate assumed to be that of the challenged party. The challenged party is considered authenticated if the decrypted response is verified to match the challenge. This procedure is performed from both sides, thus the server (end-entity) verifies the client authentication and the client (end-user) verifies the server. Both sides must have - and trust - the public key corresponding to the private key used by the CA when it issued the certificates. It's imperative, therefore, that the CA is seen to be acting transparently, and that its public key is known. This is a basic requirement of a PKI, since no party would be able to implement a trusted model without there being something to trust in the first place. Put another way, it's impossible to trust a certificate unless the user is confident it was issued by a trusted CA.

3.5.4. Non-repudiation / Verification

The certificate itself is a good example of a non-repudiation service. Any party, including a third party, can verify that a noted CA issued the certificate. The act of non-repudiation act is made possible through the use of a digital signature. The digital signature is created by encrypting given data with the private key specified for non-repudiation. It is to be expected that different applications would require different keys, as mentioned earlier. The data itself may be plain text, or squeezed into a tiny data format through the use of a special algorithm. The latter is often called a hash or a message digest.

The verifying party would basically apply the same technique as used when verifying at the time of authentication, i.e. by using the certified public key to match the expected values. This procedure would ensure non-repudiation at the time of action, since the receiving party should be able to check for certificate validity and revocation status. In order to provide for long term non-repudiation, a commonly accepted time-stamping service is necessary, and the timestamp should comprise part of the signed data, allowing it to be checked at a later stage. This could be done by a notary system.

Non-repudiation services have immense potential within electronic commerce, ranging from plain mail signatures to signing crucial agreements or business transactions. Remember that your reallife signature may be copied, but a thoroughly protected private key ensures that your digital signature is impossible to copy.

3.5.5. Protecting your key information

Security issues around network- (Internet) connected personal computers are heavily debated today. One of the most discussed issues is whether it's possible for an unauthorized person to gain access to stored data, or read and alter information that has been produced prior to being sent across the network.

Obviously it's difficult to protect against intruders without establishing a fault-proof firewall, but from a home-user perspective a firewall may not be wanted. Working through a firewall over which the user does not have personal control could limit the way in which the network can be used. Although you might have a bulletproof firewall at home, this is unlikely to be the only place from which you will conduct e-business in the future. Where is it safe to store the keys used for identification, and to sign valuable agreements, documents, and orders over the Internet? The answer is within a smart card.

3.6. The most secure smart card is the PKI card

Public key infrastructure (PKI) systems build on the uniqueness and protection offered by the users' private RSA keys. The private keys should never be exposed to anyone – not even to the user. By utilizing the power of the PKI card (a smart card with a cryptoprocessor that supports RSA) the keys may be accessed and used only within the card. Once stored in the PKI card the key value will never leave the card. It is the operating system that prevents the keys from being exposed outside the card. They can thus never be read, removed or tampered with (even by the user). User access to the card functions is via a PIN code that the user may change at any time. PKI cards are eas to use, highly portable and can be integrated with a wide range of applications. Examples of suitable applications include financial on-line services

such as home banking, secure mail, and secure web services or virtual private networks (VPNs). Remember that all smart cards are not alike - they come in many different varieties. Many cards are unable to provide support for the RSA algorithm within the card processor. And even if they do support RSA, they may be unable to handle this process very efficiently. Far too often solutions have been implemented in which the smart card is no more than a storage media for the keys. Only the PKI smart card can establish the level of security and processing speed that is required for operating in a large scale PKI.

4. INTRUTION DETECTION/INTRUTION PREVENTION

4.1. Overview

An important security product that has emerged is Intrusion Detection Systems (IDS). In order to understand IDS properly, one must first have an understanding of *intrusions*.

Intrusion is difficult to define because not everyone agrees on what is consider of an intrusion. Intrusions are defined as attempts to compromise confidentiality, integrity, or availability of data, or to bypass the security mechanisms of an IT system. An intrusion may be generally described as a sequence of related actions by a malicious adversary that results in the occurrence of unauthorized breaches to a target system or network.

4.2. What is Intrusion Detection?

The National Institute of Standards and Technology (NIST) defines intrusion detection as the process of monitoring the events occurring in an IT system and analyzing them for signs of intrusions. These intrusions are the results of attackers accessing systems from the Internet, authorized users of the systems who attempt to gain additional unauthorized privileges, and authorized users who misuse the privileges given to them. The ideal Intrusion Detection System notifies appropriate person of an attack in progress with 100% accuracy, promptly, with complete diagnosis of the attack, and recommendations on how to block it. But such ideal systems do not exist.

4.3. Why do we need Intrusion Detection System?

It is a common misunderstanding that firewalls can recognize and block intruders. A firewall is simply a fence around a network, with a couple of well-chosen gates. A fence has no capability of detecting somebody trying to break in (such as digging a hole underneath it), nor can a fence differentiate somebody coming through the gate is allowed in. A firewall simply restricts access to the designated points on the

network. Having Security cameras, motion detectors, and burger alarms can provide information about who is coming through allowed gates or if someone is digging a hole underneath, etc. These security devices can be configured to set off an alarm and notify housekeepers of any specious activity going around. Intrusion detection systems are the security cameras, motion detectors and burglar alarms. IDS can be configured to respond to predefined suspicious activities. The underlying reasons for using intrusion detection systems are relatively straightforward: protect data and maintain systems integrity. Intrusion detection takes one step further of basic measures of security mechanism such as firewalls and other access control. An Intrusion Detection System does not replace firewalls; firewalls are must in any corporate security foundation. Intrusion Detection Systems identify attacks against networks or a host that firewalls are unable to see. Having IDS to complement a firewall can provide an extra layer of protection to a system such as:

- Identifying attacks that firewall legitimately allow through (such as http attacks against web servers).
- Identifying attempts such as port scan or ping sweep.
- Notice insider hacking.
- Provide additional checks for holes/ports opened through firewalls, intentionally or unintentionally.

4.4. Types of Intrusion Detection

Now that reasons to consider having intrusion detection are defined, next issue to determine is what type of intrusion detection system best suits an organization's requirements to strengthen its network security. IDS can be viewed two different ways: how to detect, where to detect.

4.4.1. How to detect:

These are the *types* of Intrusion Detection tools. Intrusion can be detected by signature/pattern analysis, or anomaly/heuristic analysis.

4.4.1.1. Signature/pattern based IDS

It is also known as the *knowledge based* IDS. This intrusion detection system contains a database of known vulnerabilities. It monitors traffic and seeks a pattern or a signature match. IDS can be placed on a network to watch network vulnerabilities and can be placed on host.

4.4.1.2. Benefits of Signature/Pattern based IDS

- Provides very low false alarms as compare to Heuristic based IDS.
- Provides detail contextual analysis providing steps for preventive or corrective actions.

4.4.1.3. Drawbacks of Signature/Pattern based IDS

- It is difficult to gather knowledge about known attacks and keeping up-todate with new vulnerabilities.
- Signatures and corrective recommendations are generalized; thus it makes it harder to understand them.
- Knowledge about attacks is very focused, dependent on the operating system, version, platform, and application. As a result, intrusion detection tool is closely tied to a given environment. Signature/Pattern based IDS are more popular and commercially used than Heuristic/Anomaly detection based IDS. Major vendors such as ISS offer network based and host based signature detection.

4.4.1.4. Heuristic/Anomaly detection

It is also known as the *behavior based* IDS. These types of IDS tools analyze traffic patterns and infer *normal* activity. It then, applies statistical or heuristic measures to events to determine if they match the model/statistical *normal*. Events outside accepted *normal* behavior generate alerts.

4.4.1.5. Benefits of Anomaly/Heuristic based IDS

- Identify any possible attack.
- Identify attacks that we haven't seen before – Or close variants to previously-known attacks

4.4.1.6. Drawbacks of Anomaly/Heuristic based IDS

- Normal can change over time, introducing the need for periodic online retraining of the behavior profile, resulting either in unavailability of the intrusion detection system or in additional false alarms.
- Current implementations provide high false alarms.
- Requires expertise to figure out what triggered an alarm. There are many research projects in works right now with utilizing Heuristic/anomaly based IDS such as IDES (Intrusion Detection Expert System), GrIDS (Graph-based Intrusion Detection System), and Emerald (Event Monitoring Enabling Responses to Anomalous Live Disturbances).

4.4.2. Where to detect

These are *deployment techniques* of Intrusion Detection. A sensor can be placed on a network segment or on a host. They represent the products of Intrusion Detection System.

4.4.2.1. Network based Intrusion Detection Systems

It monitors the traffic on the entire network segment. Similar to a network sniffer, network based IDS tools collect raw network packets as the data source from the network or a hub/switch. However, network based IDS can reassemble packets, look at headers, determine if there are any predefined patterns or signatures match from the network traffic to generate alerts, and automatically take action based on the content of the packet. RealSecure network agents from ISS and SecureIDS from Cisco are examples of Network based IDS.

4.4.2.2. Benefits of Network based IDS

- Monitor network for port scans.
- Monitor network for malicious activity on known ports such as http port 80.
- Identify various sorts of spoofing attacks.
- Does not impact network performance.
- Increased tamper resistant.
- Operating systems independent.

4.4.2.3. Drawbacks of Network based IDS

- Packets lost on flooded networks.
- Reassemble packets incorrectly.
- No understanding of O/S specific application protocols such as SMB.
- No understanding of obsolete network protocols.
- Does not handle encrypted data.

4.4.2.4. Host-based IDS

It operates on information collected from within an individual computer system. Host-based IDSs utilize information sources such as operating system audit trails, C2 audit logs, and system logs. Operating system audit trails are usually generated at the innermost (kernel) level of the operating system, and are therefore more detailed and better protectd than system logs. System logs are collected in very compact form but contain application or system specific events. Host based IDS operate on the logs and not actual traffic. RealSecure host agents from ISS is an example of Host based IDS.

4.4.2.4.1. Benefits of Host based IDS

- Monitor events local to a host, and can detect successful or failure of attacks that cannot be seen by a network-based IDS.
- Operate in an environment in which network traffic is encrypted.
- Unaffected by switched networks and is independent of network topology.

- Monitor system specific activities such as file access, user access, etc.
- Provide thorough information gathered via logs and audit; for example Kernel logs know who the user is.
- No additional hardware is needed to implement Host based IDS solution.
- When Host-based IDSs operate on OS audit trails, they can help detect attacks that involve software integrity breaches.

4.4.2.4.2. Drawbacks of Host based IDS

- Host based IDS are harder to manage as information must be configured and managed for every host individually.
- Host based IDSs are network blind and cannot detect a network scans or other such surveillance that targets entire network.
- If the host is compromised, collected log data by the Host based IDS can be subverted.
- Disabled by certain denial-of-service attacks.
- Uses operating system audit trails as an information source. The amount of information can be immense and can require additional local storage on the system.
- Inflict performance deficiency on monitored host.

4.5. Effectively Deploying an IDS Solution

Choosing IDS is not easy as picking a technology or product or vendor. Effectively deploying an IDS solution requires planning, strategic deployment, maintenance, monitoring, responding to an incident, and handling of an incident.

4.5.1. Planning

None of the IDS products or technologies can deliver a silver bullet for solving security problems, but combined intelligently, they provide a solid solution for detecting threats to a network. Both host based and network based deployment strategies have unique benefits and strengths that compliment each other. Financial investment and budgetary limitations are major factors in deciding an IDS solution.

No single product or technology is an answer to security solution, but combining both of the IDS technologies will greatly improve any networking environment resistance to attacks and misuse.

Planning begins by establishing organization's acceptable tolerance for *Threat/Vulnerability/Risk/Impact*. In order to accomplish this tolerance, first identify the threats to a system, compare them to vulnerabilities. Secondly, determine how these systems are at risk. And then, determine if systems are compromised in any way, how it will impact these system and the business. Such analysis may be beyond technical requirement and may involve management decision. Management team has to be the decision maker in establishing an acceptable ratio and security team should design countermeasures for risks greater than what management is will to accept.

4.5.2. Strategic Deployment

Security community believes less than 15% of the intrusions to any systems are detected. A poorly deployed security solution provides a false sense of security. After determining what must be secured, other necessary requirements are listed below:

- Where IDS should be placed.
- How IDS will assist in securing systems.
- What method of intrusion to use in detection such log base detection, signature based or heuristic base detection.
- Where should the detection sensor be placed, on a network or on a host or combination as needed. Successfully deployment of an IDS solution does not make it fully secured. As Bruce Schneir put it, *"security is a process not a product."* Process has just began.

4.5.3. Maintenance

New vulnerabilities that threaten any business are being discovered regularly. As business requirements change, so will the security needs. Therefore, a security policy

will also change to accommodate these changes. Along with other security products, IDS product will also need to be updated. IDS product vendor will also provide patches and upgrades that will be needed to keep up to date. Sensors can detect only

vulnerabilities they know about; if they don't know about new vulnerabilities, they can't detect them. Having a process in place that keeps IDS up-to-date with latest knowledge base and detection definitions is a vital part of maintaining the IDS effectiveness.

4.5.4. IDS Monitoring

Counterpane Security, mentions in a white paper a crucial need for monitoring IDS by stating "*If security products were perfect, there would be no need for detection. If computer programs never had security bugs, there would be not need for monitoring.*

But protection mechanisms are not perfect, and programs have bugs. They work but they need to be monitored." If it is not monitored, Intrusion detection by itself offers a little value. Let's say a bank has security cameras. Having various cameras do not make bank secured. If no one watches these cameras, they would not be able to detect as a suspicious person coming in to the bank and robbing them. Sure everything is recorded but it could have been prevented. Just like that, monitoring IDS is very important. Network attacks can happen any time of the day not just in business hours. The monitoring goal of IDS is to positively identify real attacks from false positives and false negatives.

4.5.5. Incident Response

Security products provide protection, and that protection is primarily useful as a delaying tactic: it gives the defender time to detect the attack and respond. Why bother detecting an attack in the first place, if nothing is going to be done about it? Detection without response is useless; it's an alarm ringing with no one listening. Incident response team is needed to sort out real attacks from false positive and false negatives. Response to an incident can be automatic or manual. Automatic response

work against automated attacks but a manual response may be required for an intelligent attack. The mind behind the attack is the real enemy.

4.5.6. Incident Handling

It is very important to have a well-defined and well-documented security policy that contains incident handling procedures. Define incident handling team and their designated tasks. Improperly handled information gathering may not be admissible in court of law. Once an attack is detected, even certified incident handler may panic. Experience incident handlers have recommend following steps:

- Remain calm; don't hurry.
- Notify your organization's management.
- Provide a game plan (with options if possible).
- Apply need-to-know.
- Use out-of-band communications; avoid email and other network-based communications channels.
- Take good notes, good enough to serve as evidence in a court of law.
- Determine how the incident happened and how it was detected.
- Contain the problem; pull the network cable if needed.
- Back up the system(s), and collect evidence.
- Assess the impact and damage from the incident
- Eradicate the problem and get back in business.
- Lessons learned, apply what you have learned.

And depending on the seriousness of the attack, an organization may choose to pursue legal action against who was responsible for the attack. For such process, consulting with Network Forensic experts may become necessary.

5. APPLICATION AND DATABASE SECURITY

5.1. An approach to Application Security

One of the basic flaws in how risk is assessed and security solutions implemented is that the various components are viewed within stovepipes rather than holistically together. For instance, there are usually separate approaches and teams assessing the network, operating system, web server, database, middleware and application. Given that the applications themselves are often crafted with little oversight of security professionals and without standards of development this has created an opportunity for disaster. This paper details an approach to application security that when implemented not only brings these disparate views of risk together where they belong (within the application) but also prescribes how to involve the security professionals in the development process so that the resultant applications behaves predictably and with no surprises. Development groups are often overlooked when Security teams work within an organization. Developers are usually hired for their development or coding expertise and may have zero or minimal security knowledge. As security often tends to focus on firewalls and servers, server and network administrators often get most of the attention. Without Security involvement, applications can be developed that create major security exposures, despite heavy investment in firewalls and other technologies. Such security flaws, if discovered late in the application development life cycle, can result in applications having to be redeveloped before being deployed, or can force reliance on expensive, inflexible security solutions that can be added after the fact, often using hardware or third party applications. Security organizations should have a liaison working closely with development teams to ensure that security expertise is available as applications are being developed. In addition, security education should be mandatory for application developers, to give them the tools they need to avoid developing insecure applications.

Early websites used static pages. Gaining access to the code running on the website had the potential to alter content displayed on the site, but it was not possible to compromise the security of the platform through the code making up the site.

Solid network, platform and physical security were adequate defenses against a serious compromise. All of these areas of security are still important layers in defense of a website, but more is required today. Today's e-commerce websites are comprised of many applications interacting with one another, with back-end databases, and with other entities. Applications are objects that have privileges that can be compromised. Firewalls protect websites by only allowing certain ports to be accessible to the outside world. Typically, only port 80, HTTP, and port 443, secure HTTP, are open to the outside world. Operating system security protects against compromise from within, an intruder getting onto the internal network somehow, legally or illegally. Both of these approaches make it difficult to compromise machines, but what if a hacker accesses resources by entering through the very ports that must be open for the website to operate? By attacking the applications running on a website, hackers can potentially do just that. One way to ensure that developed applications are secure is to work with developers throughout the product development life cycle, ensuring that flexible, scaleable security is built into applications from the start. The best approach for working with development teams is to develop a security life cycle that matches the Systems Development Life Cycle (SDLC) that is in use. Security can be planned right from the concept stage of a project, allowing for application growth and easier replacement of security components as technology develops. Decisions regarding security can be made before the application architecture is completed and before code is written. This document discusses an approach to assessing application security that will work within most organizations. It first discusses some classes of threats that should be considered when designing security for applications. It then shows how to develop a simple Security Development Life Cycle to complement an organization's Systems Development Life Cycle. One approach for assessing risk in applications or systems is then discussed, with an example. Finally, some conclusions are reached about how to approach security in applications.

5.2. Classes of Threats

This section of the document discusses different types of threats that can be used to compromise an application, taking advantage of a security vulnerability. It is by no means a complete list of threats, but is included to give the reader an idea of the types of attacks to think about when designing security into web applications.

5.2.1. Privilege Elevation

Privilege elevation is a class of attacks where a hacker is able to increase his/her system privileges to a higher level than they should be. If successful, this type of attack can result in a hacker gaining privileges as high as root on a Unix System. An example of such an attack can be found at http://www.ciac.org/ciac/bulletins/m-026.shtml. In this example, authorized users of versions of OpenSSH earlier than 3.0.2 could gain the ability to run arbitrary code with superuser privileges. Once a hacker is able to run code with this level of privilege, the entire system is effectively compromised.

5.2.2. Unauthorized Data Access

One of the more popular types of attacks is gaining unauthorized access to data within an application. Data can be accessed on servers or on a network. The data can then be used for further attacks, such as using illicitly gained personal information to steal identities. Session hijacking is an example of this class of attack. HTTP is a stateless protocol, so in order to maintain the concept of logged-in session with a web application; session ids are used to tie user actions together at the web server level. Session hijacking involves guessing session ids of other users' web sessions. If session ids are not random enough, hackers can predict session ids of logged-in users and take over their sessions, thereby gaining access to any data accessible within the security context of the logged in users.

5.2.3. Denial of Service

Denial of service (DOS) attacks are currently getting a lot of attention, but the attacks that appear in the press are often network-based attacks. Applications can also be attacked in ways that render the application, and sometimes the entire machine, unusable. Poorly designed applications can sometimes be knocked offline simply by attempting logins to the userid that the application runs under enough times to lock out the userid. For an example of an application DOS attack, see http://www.securiteam.com/exploits/5XP0D1F5FM.html. In this case, a backdoor in the Kazaa and Morpheus web-based file sharing applications can be exploited to consume all available bandwidth, effectively knocking out the entire machine the application is running on. Also of interest is the fact that this attack bypasses personal firewalls setup to prevent this type of thing from happening.

5.2.4. Data Manipulation

Data Manipulation attacks involve changing data used by a website in order to gain some advantage or to embarrass the website's owners. Hackers will often gain access to HTML pages and change them to be satirical or offensive. One well-known example of a data manipulation attack is called hidden field manipulation. Data is often stored in hidden fields in a web page. This data can be viewed and changed using the 'View Source' option on the browser. If the values of the hidden fields are not verified by the application when a form page is submitted, results can be different than what is expected. For example, if prices on a shopping site are stored in hidden fields, what will happen if a hacker changes the price of an item from $100 to $1 before submitting the form and the server does not verify the price field? The hacker could receive the item for a fraction of the actual price, and the error might not be discovered until it is too late.

5.2.5. Identity Spoofing

Identity spoofing is a technique where a hacker uses the credentials of a legitimate user to gain access to an application or system. This can be a result of users being

careless with their ids and passwords, ids and passwords being transmitted over a network or stored on a server without encryption, or users setting easy to guess passwords. Once a hacker has possession of a user's credentials, he/she can login to the application with all of the privileges normally assigned to that user. This threat can be reduced or eliminated by requiring the use of strong passwords and forcing frequent password changes, by not storing or transmitting clear-text passwords, or by the use of hardware tokens. The approach to be taken depends on the value of the data protected by the id and password.

5.2.6. Cross-Site Scripting

Cross-site scripting is a relatively new approach that is being used to attack websites. It involves disguising a script as a URL variable within a URL from a legitimate site, and tricking a user into clicking on that URL, perhaps by sending it in an official looking e-mail or using some other approach. If the site does not check URL variables, the script will then execute in the user's browser. While this does not attack the website directly, it can be used to run arbitrary code in a user's browser, collect userids and passwords, or anything else that a script can be used to do. One of the most well known cross-site scripting attacks was used against Microsoft's Hotmail service and other e-mail services. See a description at http://www.whitehatsec.com/labs/advisories/WH-Security_Advisory-08152001.html.

5.3. Security and the Systems Development Life Cycle

Most large organizations that develop software applications follow a Systems Development Life Cycle or SDLC. The SDLC describes the product development methodology that takes a system from concept to reality. An SDLC is multi-phased, with different phases representing different moments in a product's life. In order to ensure that applications are developed with adequate security, a Security Development Life Cycle, or SecDLC, should be in place that complements the SDLC being used by the development teams. The SecDLC does not have to match the SDLC exactly, as long as the required security tasks can be mapped to the phases of the SDLC.

5.3.1. A Simple Security Development Life Cycle

A general Security Development Life Cycle has two phases: Risk Analysis and Test. Each phase has its own set of tasks that contributes towards designing and building a more secure product.

5.3.1.1. Risk Analysis

In the Risk Analysis phase, the security and development teams work together to ensure that security is a major consideration in the design of the system. At the end of this phase, the full costs of implementing security measures for the designed application should be known. There are three major security activities in this phase.

5.3.1.1.1. Business Requirements

As business requirements are developed, the security team needs to ensure that the organization's security standards are reflected in the business requirements documentation. At this stage, the security requirements will be at a high level. For example: ensuring that customer information to be transmitted across the Internet is encrypted. It is good practice for organizations to require that one of the areas that must sign off on business requirements is Information Security.

5.3.1.1.2. Risk Assessment

Once business requirements are developed and signed off, the development team will be working on producing technical specifications for the new application. As the design begins to take shape, the security team should be performing risk analysis of the system design, ensuring that designs that would violate security standards are rejected, or modified. The risk assessment process documents risks and threats to the application, and determines countermeasures that must be taken in order to mitigate each threat. As the Risk Assessment is, arguably, the most important security document relating to an application, the next section of this document looks at risk assessments in more detail.

5.3.1.1.3. Technical Specifications

Finally, countermeasures determined by the risk assessment need to be included in the technical specifications for the application. Again, it is good practice to have Information Security as one of the required sign-offs for this document.

5.3.1.2. Test

The Test phase is the next phase of the Security Development Life Cycle. In this phase, the security team works with the developers and the test team to ensure that countermeasures are correctly implemented and that code is developed following security best practices. There are three security activities in this phase.

5.3.1.2.1. Unit Testing

As system modules are created, developers test them in isolation from the rest of the system using test driver programs and other tools. The security team should be working with the developers at this stage, defining tests for each module to test its security behavior. For example, all buffers for a module should be checked to determine if they are susceptible to overflows. If resources permit, it is good practice to have independent code reviews at this stage also. However, for many large organizations it is impractical to review all code, but the security team may want to require that certain code with major security implications (e.g. Code that performs authentication) undergo review.

5.3.1.2.2. Integration / Quality Assurance Testing

This is the stage where the system modules are being assembled and tested as an integrated application. The security team should ensure that security testing is embedded in the overall test plan for the product at this stage. Specific tests should be in place to check authentication, authorizations and entitlements, and to test all countermeasures that were put in place as a result of the risk analysis.

5.3.1.3. Deployment

At this stage, all integration, QA and user acceptance testing has been completed and the application is deployed in production. The security team should monitor the deployment to ensure that all countermeasures that are external to the application code (e.g. Firewalls, intrusion detection, etc.) are correctly installed so that they operate as anticipated. If the application is a web application, the last step before the site is entered into DNS servers should be to conduct a full penetration, or ethical hacking, test. This final level of security testing involves professional 'hackers' attempting to penetrate the system. The testing should be done from the perspective of an outsider with no approved system access and a malicious insider who has access to the system. It is often a good idea to have this testing done by a third party who had no involvement in the development or design of the system. Once the penetration testing is complete, there will be a very good picture of any remaining vulnerabilities that may need to be corrected prior to opening the system to the public. The reason for doing penetration testing at this stage, and not during integration testing, is to ensure that the system is tested in its production state, and not in a QA or development environment, which may not completely mirror production. It is okay to do the testing in a QA environment, and many companies do, but it should be done with the awareness that, if the QA environment does not exactly mirror production there may be security vulnerabilities that are missed.

The Security Development Life Cycle that has been presented here is very general in nature. It can be adapted to fit any Systems Development Life Cycle. The approach is to ensure that most of the time Security personnel spend working on application development projects is consultative in nature, and takes place at points in the project where it is most effective. Concentrating on designing an application for security, and then testing to ensure that design goals were met, is significantly less expensive than retrofitting security to a product that is already built.

5.4. Creating the Risk Assessment

There are many different ways to perform risk assessments on applications and no one variation will work in every single environment. It is really a matter of finding an approach that works in most cases and only deviating from it when it is clearly not appropriate for the project being reviewed. Using a consistent approach whenever possible makes it easier for developers to know what to expect and allows Security personnel to follow a ‘cookie cutter’ approach to assessing risk.

6. WEB SECURITY

6.1. Overview

Securing web sites, and web servers in particular, has been the focus of many security articles and conferences over the past few years. Obviously, a web site's security level is heavily influenced by the security means, which are used by, and on, the web server. It seems obvious that the key to a secure web site is the level of security achieved from security of the web server. One might have "stumbled" over a web site's database security issues if he or she was interested in DBA chores. Database security is also a well-known subject in web site security, but it is mostly documented as a standalone issue.

Building a web site is a task that involves more then one OS and more then one kind of software. Therefore, the security of the web site is achieved from the synergy of all the factors and not from the web server alone.

6.2. Assumptions

When building a web site we must survey the risks facing the web site from all different aspects. Not all web sites face the same "threats"; many web sites are just another collection of HTML pages in the vast cyberspace of the Internet. But, web sites conducting business, containing information (considered valuable for a malicious hacker) or holding a political view, are at higher risk then others. E-commerce web sites often hold valuable information (credit card numbers or other private, personal data) and conduct business, and are thus placed at a high-risk position.

Having recognized a web site is in the high-risk zone, we must consider the different types of security hazards:

- Denial of Service (including distributed).
- Defacement (the replacement of content on a web site, indicating it has been hacked).

- Data Theft.
- Fraud (data manipulation or actual theft).

While any of these attacks might cause revenue lose, the method of defense against each is different. Since there is no global security solution that can provide the full defensive spectrum an e-commerce web site requires, it has become extremely difficult to choose the right line of defense. Security is a product that comes with a price tag. At first, this might be very obvious since products such as firewall and anti-virus have known pricing. However, the costs of on-going security, software-security updates, new web-site technologies etc, cannot be calculated during initial installation planning. Eventually the web site owner will have to decide what level of security will be provided, while considering the current risks and costs involved.

6.3. Web Sites Under Attack

Web site attacks vary significantly from site to site and from hacker to hacker, and their focus has changed as well in the passing years, shifting from network level attacks to web server hacking from within the HTTP protocol itself. DoS and DDoS attacks have become a hacker-sport and can be seen in different forms; Ranging from network based DoS such as PING flooding, to full connection HTTP requests.

6.4. DoS and DdoS

When a hacker wishes to “down” a web site, all which is needed, is a computing base that can produce a larger amount of CPU-demanding activities (for example, IP floods) then the web site is capable of handling. This is true for a fully clustered web site that is connected via a T1 connection, not only for web sites with more limited resources. The attacker needs only to generate traffic that exceeds the line capabilities, and effectively the web site will no longer be available to the Internet. Generating a large amount of traffic doesn’t require having a large connection on the attacker side. The attacker may choose to use “bots1” or amplifiers2 as the attack base. Most information regarding DoS and DDoS shows the use of network level

exploits and various methods of IP based flooding. The SANS paper on the subject "Consensus Roadmap for Defeating Distributed Denial of Service Attacks" which can be found at http://www.sans.org/ddos_roadmap.htm, reflects these methods and the possible defense. Recently, a new method of DDoS has been developed. Using bots to open full connections to the web site, and request an object on the web site. Using full connections compromises the identity and the origin of the attack, since the bots can be hard to trace back to their owner. These connections cannot be differentiated for all intents and purposes from ordinary requests of web browsers.

Currently there are no known defenses against DoS attacks implementing full connections (CDN3 is a partial and extremely expensive method that isn't feasible for most web sites). This is due to the fact that no publicly available web server or securit product can fully guarantee connection originates from a "bot" and not from a legitimate connection.

Defending your web site against the more "ordinary" DoS and DDoS attacks (namely network level attacks) is a well documented art, and consists mainly of ISP cooperation with the web site owner. Most methods of defense include rate-limit of various forms, and unwanted network traffic blocking (such as fragment blocking, UDP blocking etc).

Most of the blocks need to be performed at the ISP level, or the attacker will be able to saturate the line connecting the web site, effectively denying service to the web site.

6.5. Web Server Based Attacks

Many of the network-based attacks that create a denial of service are hard to achieve, or hold little "glory" to the attacker. This said, one must consider the fact that data theft cannot be achieved via DoS attacks. Therefore, web server attacks have become extremely popular in the past few years. Web server attacks bypass the firewall since they connect to the web site with legal network requests (i.e. TCP port 80), and are hard to trace if the web site does not employ strict log file procedures.

Web based attacks vary from web server to web server. For example: gaining control over a console on a remote MS-IIS server can be achieved using different variants of the Unicode attack, while Linux Apache server console can be controlled using a Perl test cgi attack. Other attacks and vulnerabilities through which a remote attacker can gain access to a web server while bypassing the firewall are listed in various web resources, such as www.securityfocus.com, the bugtraq mailing lists and more.

6.6. Known Web Configuration

There is no single way to install a web site that will hold all the security answers. The different ways to install and configure the different web and network components varies greatly as web sites become more complex. A few known configurations that address the security issues are:

6.6.1. Configuration 1 – Basic Disjointed

A straightforward configuration, which includes the web server as a multi-homed server with one interface connected to the world and a second interface dedicated for database communications. All communications to and from the web site are maintained by the firewall while internal communications are not monitored or filtered.

6.6.1.1. **Security considerations**:

This basic configuration provides network level security (via the firewall) and DB protection (via disjointed networks).

The load balancer (if external hardware is used) can be used as the second level network-filtering device for extra security. The use of two network cards provides low-level protection against poorly configured firewall devices (for example, fire-walking will not reveal the DB server). This configuration provides no means of application or OS level protection. The entire security architecture is based upon the filtering devices (firewall and load balancer). If the OS hardening process is not redone frequently on a per-patch basis, the web site will be vulnerable to application

and OS level hacking. In the event that the web server is hacked the database server will be fully exposed to the hacker via the web server. This is true even if the second NIC on the web server uses a different protocol. It is recommended that a basic method of filtering be used to prevent the misuse of networking protocols. The Compaq DISA6 and Microsoft DNA7 web site designs are similar and are basically modeled in this configuration. Both Compaq and Microsoft rely on the OS hardening process to provide the application level security and on the programmers' capability to produce secure code.

6.6.2. Configuration 2 – Filtered Disjointed

In this configuration, the addition of the filtering firewall, via the second "DMZ" on the main firewall provides an added level of security. Any hacking on the web servers will provide only minimal access to the database servers. Obviously the web servers can access the database server with an appropriate ODBC connector or similar means. This configuration could potentially provide a hacker (should he be able to "own" the web server machine) limited direct data access capabilities. The use of out-of-band communications means that the connection to the server is done from a different route then all other communication to and from the web site. This configuration can be achieved with a second firewall for improved performance. The firewall would be placed between the DB and the IIS servers (as suggested in the MS paper). It is not necessary to place the DB server in the corporate network.

Application business logic for the web site is based on a separate server to allow for easier scalability. This server may also be used for web management. Software such as MS Site Server or MS Application Server provides the content distribution, web statistics etc.

6.6.2.1. Security considerations:

This configuration provides network level security (via the firewall) and DB protection (via disjointed networks). It also provides low-level application protection

since core data processing is shifted from the front-end web servers to back office application servers that have no direct communications with the site's users.

If MS SQL is used, TCP 1433 should be used instead of named pipes. This will provide a higher level of filtering.

When implementing the web content distribution mechanism it is recommended not to use windows shares. FTP or MS Site Server replications are preferred. The "Filtered Disjointed" configuration provides the administrator with the tools to filter all network-based activity on the secure side of the firewall. The main idea behind this configuration is to eliminate the ability of one server to communicate directly with the other servers. Application connectivity is allowed to provide the site functionality (web servers will be allowed communications with MS SQL Server using TCP 1433), and no other protocol will be allowed. Although there's a performance penalty due to the extra network segments and filtering, should one of the web servers be compromised all network transactions can be logged, leaving an audit trail.

6.6.3. Configuration 3 – Application Protection

In the effort to protect the web site from application level hacking, we need to use a "higher level" filter. The filter would be used to examine the HTTP protocol, and if possible the HTTP GET, HEAD, POST, and PUT commands and parameters. This parameter should comply with RFC 2616 (http://www.faqs.org/rfcs/rfc2616.html) and with the restrictions of the site administrator. Such a filter can be found i n some of the commercial proxy servers or in dedicated filtering products. This approach apposes the Microsoft e-commerce strategy shown earlier in configuration 1, and in the e-commerce web site security, that all application level security should be driven from the DNA design and proper code writing.

6.6.3.1. **Security considerations**:

This configuration provides a high level of security, both network and application level.

Application filtering might require the use of out-of-band management tools, since not all proxy servers can act as routers for other non-HTTP protocols. The "Application Protection" configuration provides the administrator with multi-layer security protection. It can be used in versatile situations, and has proven itself in protecting web sites from new hazards such as Nimda and code-red (at the time of the worm release un-patched web sites using the "Application Protection" configuration would not be harmed). This protection, however, doesn't scale easily to mega-sized ecommerce sites. Monitoring tasks should be carefully planed. When monitoring a web site that has only one function that answers to HTTP requests in the client path, the monitor termination point is clear. In a configuration that holds many different components that receive HTTP requests it is imperative to monitor them separately and to assure that they are all up.

7. SECURING EMAIL AND ANTI SPAM

7.1. Overview

Email security has become a hot topic in Information Technology circles as new exploits and vulnerabilities affecting the most popular email clients and operating systems continue to make headline news on a regular basis.

It is no wonder that email security is a priority concern for many organizations. In this section, I will outline the various threats to email security, focusing on those that are of particular concern. I will then review some of the most recent advancements in the industry that are aimed at solving some of these issues.

7.2. What Does Email Security Involve?

The three main principles of Information Security involve maintaining the confidentiality, integrity, and availability of information resources. These three principles can be directly applied to the area of email security as well.

Confidentiality of email involves making sure it is protected from unauthorized access. Integrity of email involves a guarantee that it has not be modified or destroyed by an unauthorized individual. Availability of email involves ensuring that mail servers remain online and able to service the user community. A weakness in any one of these three key areas will undermine the security posture of an email system and open the door to exploitation.

7.3. What Are The Threats to Email Security?

7.3.1. Viruses

Email security is threatened by a range of issues. One of the most publicized and high risk of all the issues is viruses. Viruses are so dangerous because they often deliver extremely destructive payloads, destroying data, and bringing down entire mail systems. As a result they are a major drain on corporate IT departments and users.

According to an ICSA Labs 2003 Virus Prevalence Survey, in 2003 nine of the Top 10 reported viruses were mass mailers. Also, all of the viruses that were responsible for actual disasters during that time were either Internet worms or mass mailer viruses. To make matters worse, both of these virus types tend to stay around longer than other types, even after anti-virus products have included protection against them in their products.

The top 10 reported viruses, nine were either mailers or mass mailer viruses. The exception to this was Blaster, which was a worm that exploited a DCOM RPC vulnerability but did NOT contain any mass-mailing functionality. In the same ICSA survey, it was identified that email as the source of virus infection has been steadily increasing.

The impact of viruses on organizations is huge. The impact goes far beyond money, resources, and effort required to recover from such incidents. It also includes loss of productivity, corrupt and/or lost data, and loss of user confidence.

7.3.2. Spam

Another major threat to email security today is SPAM, often cited by organizations as being their *number one* concern. Otherwise known as junk email, SPAM is considered a security threat not only because the volume of it can affect system availability, but also because it can carry viruses, malicious code, and fraudulent solicitations for private information.

Businesses lose money when SPAM overloads network and server resources. Even with spam filtering mechanisms in place employees inevitably end up spending inordinate amounts of time sorting through messages trying to distinguish legitimate emails from SPAM.

In a survey conducted by Information Security/SearchSecurity.com on the business implications of spam, lost productivity (92 percent) and clogged email servers (62 percent) were cited as the most painful consequences of spam. However, a growing concern is the threat of virus propagation via spam. Many security

professionals fear that virus writers and spammers could get together and collaborate on more invasive ways of compromising networks and circumventing filters. That will make the line between what is a virus and what is spam more fuzzy than it has already become, when you consider the consequences of spam on resource utilization.

7.3.2.1. Well-intended SPAM?

Described by research firm Gartner as 'Friendly Fire', the volume of email being sent by well-meaning friends and family to employees is on the increase. Although the statistics available on this particular issue vary greatly, SurfControl Inc. cited in their whitepaper 'Fighting the New Face of Spam' that friendly junk email could cost a company with 500 employees nearly $750,000 each year. Although email from family and friends may pose less of an overall security risk the *volume* of it can certainly affect *availability*. And there is increased risk also if you consider that many home users sending these 'friendly' emails are sending them from less secure systems than we find on a corporate network where often virus definitions are out of date and systems are unpatched. This makes it even more important for organizations to ensure they have systems in place to protect against not only the obvious, but even the seemingly well-intentioned.

7.3.3. Phishing

Phishing, also known as identify theft, is a newer threat to email security that was relatively unheard of one year ago. Phishing is the process whereby identity thieves target customers of financial institutions and high-profile online retailers, using common spamming techniques to generate large numbers of emails with the intent of luring customers to spoofed web sites and tricking them into giving up personal information such as passwords and credit card numbers. It is a problem that has literally exploded over the last year. A study released by Gartner Research in May estimates that 76 percent of all known phishing attacks had occurred since last December. The Anti-Phishing Working Group (www.antiphishing.org), an industry association of more than 200 organizations, reported 1,125 unique phishing attacks in

April, up from 402 in March and nearly seven times the number reported in January.8 It is expected that these numbers will continue to climb drastically as security professionals struggle to find an effective solution to the problem.

Phishing has the potential to be highly lucrative for the 'Phisher', the individual or organization staging the attack. For the most part, a phishing attack is easy and cheap to engineer, is extremely hard to trace, and even if only a small percentage of recipients respond to requests for personal information – the return on investment can be very high. Although early phishing attacks were marked by misspellings, improper grammar, and less than perfect imitations of corporate logos and websites, Phishers are becoming more sophisticated in both the quality of their scams and the techniques they are using making this a growing security risk.

Gartner estimates the direct cost to companies of phishing attacks was $1.2 billion in 2003. Given the sharp rise in the number of phishing attacks so far reported in 2004, its obvious losses in 2004 will exceed last year's numbers. The impact of phishing attacks against organizations doesn't stop with direct losses. Companies are also faced with downtime during an attack, having to issue new credentials to customers who have compromised their personal information, potential liability, and damage to their corporate image. And if phishing can't be brought under reasonable control consumers are going to become extremely reluctant to do business online (therefore loss of consumer confidence).

7.4. What can we do?

There is a variety of mail security products on the market today, aimed at addressing the various threats to email security. They come in the form of special software that you can load on an existing mail server or on a dedicated mail gateway platform, or in the form of a hardware appliance that acts as an email gateway. Another option for companies is to outsource mail security to an outsourced service provider. All of these scenarios typically offer a similar feature set, although there are definite differences among competing products in terms of what they have to offer. Some of the common features in mail security products today include content

filtering services such as antivirus, antispam, HTML tag removal, script removal, block of attachments by file type, scanning of inappropriate content, confidentiality checks, and disclaimer enforcement. Antispam methods supported by most products include real-time blackhole lists (RBL), heuristics, confirmation process, Bayesian filtering, open relay protection, size and bandwidth control, and encryption. Despite all the advancements in email security products, we continue to see an increase in the number of security related issues. Virus writers are continuously looking to exploit vulnerabilities in systems and software, and make every attempt possible to cover their tracks. Spammers are constantly changing the appearance of spam and masking its source to avoid it being blocked before it reaches it's target. It is evident in both of these scenarios that one of the biggest challenges in solving the virus and spam problem is in identifying the origin of email messages. As a result the industry is crying out for radical changes to the email infrastructure that will bring these problems under control. Some of the major initiatives over the last year intended to address these ongoing issues involve *Sender Authentication*. They include the Sender Policy Framework, Caller ID for Email, the Sender ID Framework, DomainKeys, and Accreditation and Reputation Services.

7.4.1. Sender Policy Framework

One of the first technologies developed to authenticate the sender of an email message was the Sender Policy Framework (SPF). It is a technology created by Meng Wong (founder of email service firm pobox.com) that aims to identify the origin of email messages.

7.4.2. Caller ID

Another technology aimed at Sender Authentication, developed by Microsoft, is called Caller ID for Email. Similar in many ways to SPF, Caller ID specifies what is called a Purported Responsible Address (PRA) record, instead of an SPF record. The difference between the two is basically the algorithm used to determine the address that is checked for authenticity. SPF uses the visible email address of the sender, while PRA checks the record against the most recent sender of the email message.

So, PRA indicates where the email came from most recently, SPF indicates from where the email initially came. After Microsoft announced it's plan earlier this year to pursue the standardization of it's Caller ID technology,11 it ended up proposing a hybrid specification to the Internet Engineering Task Force (IETF) combining it's Caller ID technology with SPF. The hybrid solution is known as the Sender ID Framework, and also comprises a third specification called Submitter Optimization.

7.4.3. The Sender ID Framework

Sender Optimization is an optional extension to the SMTP MAIL command that would allow the receiver to check for spoofing BEFORE the message is sent across the internet. It allows the sender to declare the PRA within the SMTP protocol. If implemented, a SUBMITTER= parameter would be specified on the MAIL FROM: command, if the PRA is different from the MAIL FROM. This would be a necessary requirement for mailing list servers and mail forwarders where the MAIL FROM will almost never match the PRA. Some implementation examples of Sender Optimization where the submitter parameter would be used (taken from http://www.microsoft.com/mscorp/twc/privacy/spam_senderid.mspx):

Normal Mail Submission is like below;

S: 220 alumni.almamater.edu ESMTP server ready

C: EHLO example.com

S: 250-alumni.almamater.edu

S: 250-DSN

S: 250-AUTH

S: 250-SUBMITTER ***(SUBMITTER extension advertised in EHLO response)***

S: 250 SIZE

C: MAIL FROM:<alice@example.com> SUBMITTER=alice@example.com *(SUBMITTERparameter added to MAIL command)*

S: 250 <alice@example.com> sender ok

C: RCPT TO:<bob@alumni.almamater.edu>

S: 250 <bob@alumni.almamater.edu> recipient ok

C: DATA

S: 354 okay, send message

C: From: alice@example.com

C: (message body goes here)

C: .

S: 250 message accepted

C: QUIT

S: 221 goodbye

The Sender ID Framework has been debated by the IETF for the past several months. Among several issues with the proposal is Microsoft's attempt to patent technology used for the Caller ID component, which may end up meaning Sender ID would require users to sign a license agreement. This has angered many in the open source world, and has somewhat soured the support of some that had previously backed the technology. The Sender ID proposal was most recently dealt a setback on September 11th when the IETF reached consensus that Microsoft's patent claims should not be ignored and their insistence on keeping the technology secret was unacceptable. After the results of the IETF vote, Microsoft indicated it will continue with its plans to develop its own proposal for Caller ID. They stated however, that

they will use the Purported Responsible Address (PRA) to authenticate the source of email messages although they will continue to publish both SPF and PRA records (they will only check the PRA).

In the meantime, the proposal for the Sender ID Framework is not necessarily dead. The IETF ruling allows for negotiation, if Microsoft considers removing licensing restrictions. Given that some of the biggest email providers in the world (AOL, Microsoft, Yahoo, Comcast, Earthlink, and BT) have been promoting Sender ID, and products like Sendmail are adding support to their mail transfer agents, Sender ID is likely still to be further debated at the IETF.

7.4.4. Domain Keys

Domain Keys is a technology proposal developed by Yahoo, that provides a mechanism for verifying both the domain of each email sender and the integrity of the messages sent using DNS and an RSA public/private key method to digitally sign messages.

8. FIREWALLS

8.1. Overview

This chapter will outline the critical elements for implementing an Internet firewall in your organization, in a way that allows you to choose the solutions that best suit your needs. The options range from the cheap to the not-so-cheap, from the visible to the invisible, and from difficult to easy.

As individual requirements will vary, we cannot readily endorse any one product or approach over another, yet they all offer benefits that you may want to use. In this new market, most products are only partially finished -- yet are completely priced -- so your choices will likely end up as budgetary decisions as much as functional ones.

8.2. Firewall Basics

Almost by definition, a "firewall" provides a filter that incoming or outgoing packets must pass through. If the firewall does something beyond filtering, like checking against a restrictions list, that's great, but it's not necessarily the "definition" of a firewall's function.

Most firewalls do perform some sort of "accept" or "reject" functionality, but that's strictly a matter of implementation. The simplest firewall could just be an Ethernet bridge that you keep powered off, only to be made available when the connection is needed. This would probably work for keeping intruders off of your network, but I doubt you want the management interface much. Most firewall products offer much more in the way of actively filtering packets according to certain criteria that you establish.

These filtering firewall products can take many forms. They may be a replacement TCP/IP stack that you load on an existing system, or a software module that exclusively communicates with an existing stack. At the other end of the extreme, the product may be a completely independent operating system written explicitly with Internet security as the objective. There are also application-specific

firewall products that only offer protection for certain types of Internet connectivity, such as SMTP or HTTP. There are also hardware-based products that typically fall into the router realm, allowing you to set filters for incoming and outgoing connections. Prices range from free (bundled with the stack or app) to tens of thousands of dollars.

All of these products can rightly be called "firewalls" because essentially they trap inbound or outbound packets, analyze them, and then either send them on their way or toss them out. Any one of these products may or may not suit your needs. Once you've got a handle on the issues, however, you should be able to do your own product elimination, simply by comparing functional specifications.

At the least, almost all firewall products offer IP address filtering. These filters work by examining the header of the IP packet and making pass/fail decisions based on the source and destination IP addresses. For clarification purposes, let's look at the figure above, which shows a simple two-segment network with a firewall separating them. One segment has a UNIX host, and the other has a PC client.

When the PC client tries to Telnet to the UNIX host, the Telnet client on the PC generates a TCP packet, and hands it to the local stack for delivery. In turn, the stack places the TCP packet inside of an IP packet, and then sends to the UNIX host via the route defined in the PC client's TCP/IP stack. In our case, the PC client is sending the IP packet to the firewall for delivery to the UNIX host.

Suppose that we have told the firewall that it is not to accept any packets destined for the UNIX host, as depicted in below. Then the firewall would reject the IP packet, perhaps bothering to tell the client or perhaps not. Since no IP traffic for that destination would get forwarded, only users on the same segment would be able to access the UNIX host.

Another scenario might be that the firewall has been configured so that it simply will not accept any packets from that PC in particular. Then other systems could connect to the UNIX host, but that specific PC could not.

This type of filtering is the most basic of all. By setting accept or reject filters per IP address, these types of products can provide very basic protection mechanisms for a simple LAN. If the systems are not allowed to communicate because of source or destination IP address filters, then the packets are simply rejected.

These types of filters are commonly used in smaller shops that need to control where users can or cannot go, but beyond that they're not extremely reliable. IP addresses can be spoofed, so using these filters by themselves are not enough to stop an intruder from getting into your network. However, it is a fundamental building block of good firewall design, and is a critical component of a complete defensive infrastructure.

If you do employ IP address filters, then make sure that you use IP addresses when you create your accept and reject tables, and don't use DNS host names, since DNS can be spoofed even more easily than IP addresses can be.

8.3. TCP/UDP Port Filtering

Using simple IP address comparison to allow or reject packets is a brute method of filtering. It doesn't allow for the possibility that multiple services may be running on the destination host, some of which we may want to allow users to access. For examp le, we may not want users to Telnet into the system, but we may want them to be able to access the SMTP/POP mail server that's running on it. To enable this level of control, we have to be able to set filters according to the TCP or UDP port numbers in conjunction with the IP address filters described earlier.

For example, the default Telnet configuration calls for the server to monitor TCP port 23 for incoming connections. Therefore, if we know that we do not want to allow any Telnet connections to our UNIX host from the PC, we can simply tell the firewall to reject any IP packets going to the UNIX host that have a TCP destination port number of 23. Since the PC's Telnet client would normally generate just such a request, the service would effectively be disabled for it, as depicted above.

It should be pointed out that this type of setup -- where we are explicitly excluding a port -- is generally a bad idea. If you need to protect a system, you're better off by rejecting everything, and only accepting the TCP or UDP packets that you know you want to let through. It may seem like more work, but it's less work in reality, and has the added benefit of keeping your systems from being easily compromised. In other words, of the two approaches -- that which is not forbidden is allowed, and that which is not expressly allowed is forbidden -- the latter one is safer.

If we wanted to reverse this example, perhaps using SMTP and POP, then we would add these services to the acceptable list for the UNIX host's destination address, and reject all other packets. Therefore, any connection request bound for the UNIX host which had TCP destination port addresses of 110 or 25 would be allowed to pass, but no other packets would, since they wouldn't meet this "allow" condition. This would include Telnet, thereby providing the "exclude" condition described above, only with less work (told you so).

By combining the IP addresses and TCP/UDP port numbers, you can develop some pretty reliable filters. For example, if your internal SMTP mail server only talks to your Internet Service Provider's (ISP's) mail server, then you could implement a firewall filter that only allowed incoming SMTP connections that came from the ISP's mail server, and are destined for your internal SMTP mail server. This will keep some of the hackers from being able to exploit any SENDMAIL weaknesses that you haven't plugged.

This level of control is generally where many of the pseudo-firewall products stop. By allowing you to set your filters so that no incoming traffic from the Internet can access any ports except the ones you want, it would seem to preserve your security to a satisfactory degree. However, this is just not the case.

8.4. Clients have TCP/UDP Ports, Too

Since TCP/IP is a peer-to-peer protocol, each node has a unique address. This philosophy carries up into the applications layer as well, meaning that applications

and services also have addresses (or port numbers). Since it takes two to tango, both the client and the server must have unique port numbers on their individual systems in order for a TCP or UDP connection to be established. For example, the Telnet server listens for incoming connections on port 23. However, the Telnet client also has a port number. Without this, the client's IP stack would not know which application the packet was for.

Historically, almost all TCP/IP client applications use a randomly assigned port number above 1023 for their end of the connection. This is a legacy from TCP/IP's roots in the UNIX world. On UNIX systems, only the root account has access to ports below 1024, which are reserved for server services (like Telnet, FTP, etc.). In order to allow client applications to work, they must use port numbers above 1023.

For you to allow any sort of connection to work then, you must allow any packet with destination port numbers higher than 1023 to come into your network. If the response packets are not returned, the client will not be able to establish a connection.

In terms of Internet firewalls, this can cause some problems in design. If you are to block all incoming ports, then you have just kept all of your clients from being able to use Internet resources. The inbound packets that are the responses to their external connection requests will not survive the firewall's inbound filters.

You may think that it's okay to open up all ports above 1023. Not so. There are a lot of services that run on ports higher than 1023, such as X clients, and RPC-based services like NFS and NIS/YP. Most of the non-UNIX IP-specific products -- like NetWare/IP for example -- use port numbers that are above 1023 as well. This means that if you let any old packet that meets the above-1023 criteria into your network, then you're exposing those systems to attack, and none of them are particularly well known for their robust security mechanisms.

8.5. Additional Security Measures

One potential solution to this problem is to build bi-directional filters into your firewall. You may want to define the filters so that you only allow packets that are from well-known services into your network, and reject any packets that are not from specific applications. For example, if you know that your users will be accessing World Wide Web servers, then you could only allow packets that have a source port number of 80 into the network, as illustrated below.

Unfortunately, there are a couple of problems with this solution as well. First of all, you don't always know what port numbers the servers that you are trying to access are running on. Modern day servers such as HTTP and Gopher are completely configurabl e in this manner, allowing you to run them on any port that you want to.

If you implement this type of filter, then your users will not be able to access those sites that do not use the "standard" port numbers that you expect them to.

Another security problem comes from the fact that there is no way for you to know for sure that the packets coming into your network from port 80 are indeed response packets from a World Wide Web server. Some hacker may have compiled their own network invasion tool that runs on port 80 on their machine, thereby making their insidious data look perfectly harmless, at least to your firewall anyway. If they can get into your network just by setting their application's source port manually, then they can do whatever they want with the vulnerable systems, and your firewall will be useless.

8.6. UDP Port Filtering

Due to UDP does not have any ACK functionality, we cannot control UDP incoming ACK bits. UDP is designed for unreliable transmissions which do not require or benefit from negotiated connections. These types of services generally include broadcasts, routing protocols, and other "maintenance" packets that advertise services. Unfortunately, there are lots of other services that use UDP as well, including NFS, NTP, DNS, WINS, NetBIOS-over-TCP/IP, and NetWare/IP. These

types of services can be corrupted pretty easily, or used as launching pads for additional attacks on your network.

One possible solution to this problem is to simply not allow any incoming UDP connections. To enable this functionality, you need to be able to configure your firewall so that it will forward UDP packets received from the internal interface, but will not forward UDP packets received from the external interface. While this will certainly prevent any incoming UDP connections, it will not always be feasible.

For example, DNS name resolution requests use UDP, so if you are providing your own DNS services, then you must allow at least some internal connection requests to pass through your firewall. Also, there are client applications like Archie and IRC which use UDP, and if you want to provide them to your users you will need to let the appropriate response packets into your network.

You'll need to provide for some sort of UDP connectivity, whether it is for end-user applications or for system level services like DNS. About the only thing that you can do is to limit t hese connections to somewhat "trusted" sites. For example, you may want to filter DNS so that only UDP packets between your internal DNS server and your ISP's DNS server are allowed to cross the wire. Likewise, you could filter Archie packets so that only the UDP packets from a specific Archie server were allowed into the network. Talk, being a relatively uncontrollable application, is not easily configurable for filtering, and would likely have to be cut off altogether.

There is a risk with allowing even well-known hosts to send packets into your network, however. If a hacker can spoof the IP address of the host that you have given clearance to, then they can take advantage of your internal system, albeit to a limited degree. A new breed of packet-filtering routers are becoming available that attempt to solve this problem by "remembering" outbound UDP packets. If an incoming UDP packet matches the destination address and port number of a recent outgoing UDP packet, then it is allowed in. If no recent packet can be found in memory, the firewall will reject the packet.

Again, the fundamental problem with this approach is that there is no way of knowing for sure that the outside host generating the packet is indeed the service that the internal client is expecting to communicate with. If a hacker were to spoof the IP address of your ISP's DNS server, then they could theoretically launch an attack from the UDP port associated with DNS. However, this particular weakness would exist anyway if you were filtering UDP packets so that only DNS queries and results from your ISP were allowed into the network. Your situation is neither improved nor degenerated by deploying these dynamic packet-filtering products.

9. REFERENCES

[1] http://msdn.microsoft.com/

[2] http://technet.microsoft.com

[3] http://support.microsoft.com/kb/153119/en-us

[4] www.faqs.org/rfcs/rfc2821.html

[5] http://www.windowsitlibrary.com/

[6] http://www.emailarchitect.net/

[7] http://www.systemwebmail.com/faq/1.6.aspx

[8] http://www.codeproject.com/KB/cs/SendMailUsingGmailAccount.aspx

[9] http://www.aspheute.com/

[10] http://www.developer.com/

[11] http://www.devsource.com/

[12] http://www.daniweb.com/code/

[13] http://www.f-secure.com/

[14] http://securitylabs.websense.com/

[15] http://www.kaspersky.com/

[16] http://www.ibm.com/

[17] http://emea.trendmicro.com/emea/home/

[18] http://www.aladdin.com/

[19] http://www.verisign.com/

[20] http://www.8e6.com/internet-resources/internet-filtering-white-papers.htm

[21] http://www.sonicwall.com/

[22] http://www.itsecurity.com/

[23] http://www.sonicwall.com/

[24] http://www.firewall-servers.com/

[25] http://www.all-internet-security.com/

[26] http://www.symantec.com/

[27] http://www.ca.com/us/it-security-solutions.aspx

[28] http://www.gfi.com/

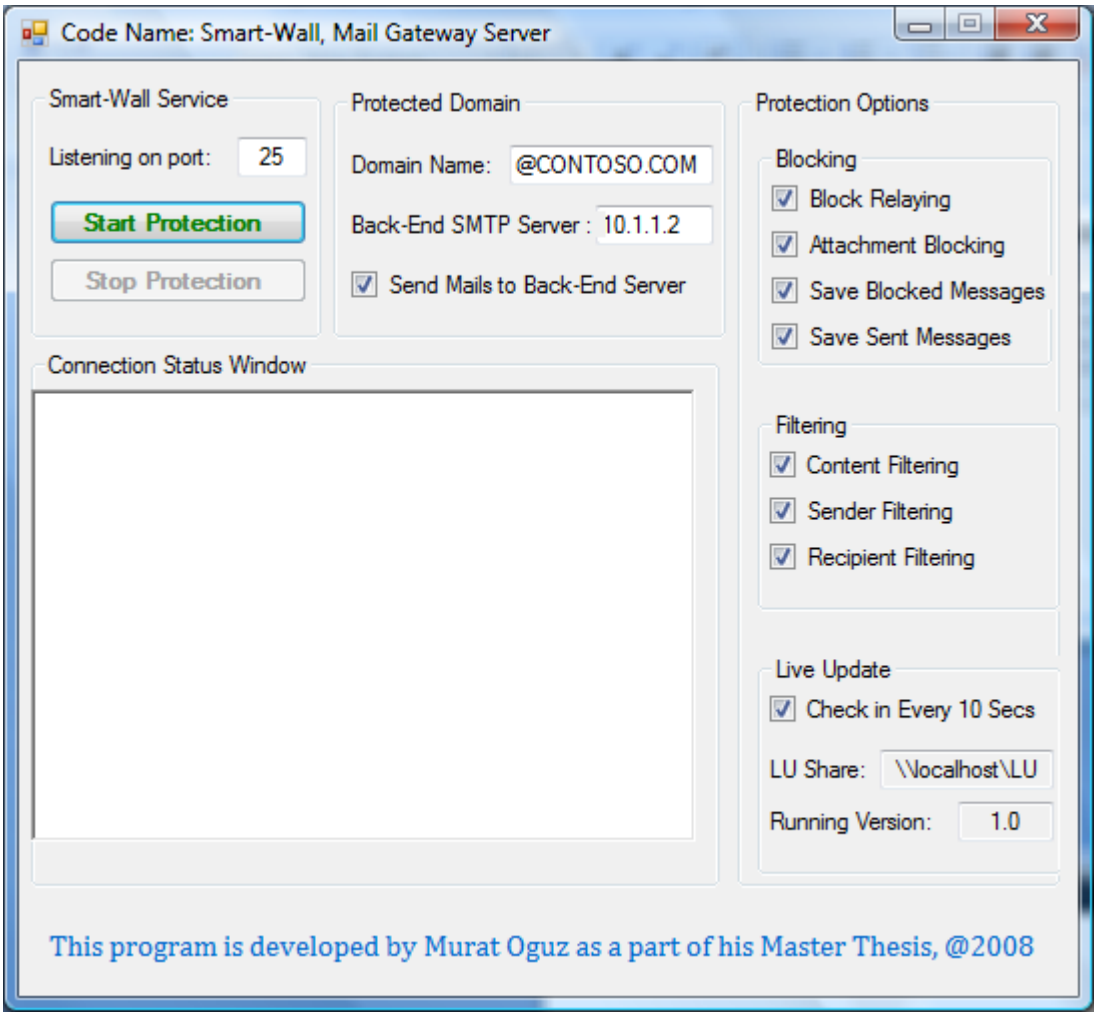
[29] http://www.pandasecurity.com/

[30] http://www.petri.co.il/test_smtp_service.htm

APPENDIX A – Mail Gateway Server, Demo Code

This appendix contains a concept program for securing SMTP mail flow for an organization.

A.1. User's Manual



Start Protection: Starts the service to listen for incoming mails.

Stop Protection: Stops the services.

Listening on port: Sets the port which the service is listening.

Domain Name: The protection will be done for this domain name.

Back-End SMTP Server: The server which the protection service will deliver the mails after scanning.

Send Mails to Back-End Server: Enables delivering the mails to back end mail server after scanning.

Block Relaying: Blockes relaying for the domain which is set in Domain Name.

Attachment Blocking: Blockes the mails which contain any attachment.

Save Blocked Messages: Saves the mails onto the disk under %Program folder%\Queue\Sent if it is blocked.

Save Sent Messages: Saves the mails onto the disk under %Program folder%\Queue\Sent after scanning.

Content Filtering: Scans and blockes a mail due to its content. Keywords for scanning can be set in the file %Program folder%\Config\Keywords.txt

Sender Filtering: Scans and blockes a mail due to the sender. Senders for scanning can be set as a mail address or a domain name in the file %Program folder%\Config\Senders.txt

Recipient Filtering: Scans and blockes a mail due to the recipients. Recipients for scanning can be set as a mail address or a domain name in the file %Program folder%\Config\Recipients.txt

Check in Every 10 Secs: The service checks for new keywords, senders and recipients files from its live update share.

LU Share: Live Update share for the new files.

Running Version: Show the running Live Update files version

A.2. Source Code

The source code is in the CD which is attached to this paper. If you also need the source code, you can directly contact with the writer via mu_og@hotmail.com.