A Proactive Approach to Network Security

A thesis
submitted in partial fulfilment of
the requirements for the award of the degree of

Master of Technology
in
Computer Science and Engineering
by

ROHAN P

Department of Computer Engineering
National Institute of Technology, Calicut
Kerala - 673601
2004

Certificate

*This is to certify that the thesis entitled "*__A Proactive Approach to Network Security__*" is a bonafide record of the work done by* **ROHAN P**, *(Roll no. Y2P106) under our supervision and guidance. The thesis has been submitted to the* **Department of Computer Engineering** *of* **National Institute of Technology, Calicut** *in partial fulfilment of the award of the Degree of* **Master of Technology** *in* **Computer Science and Engineering**.

Prof. Govindan
Professor and
Head
Dept.of Computer Engineering
NIT Calicut

Vinod Pathari
Project Guide
Senior Lecturer
Dept.of Computer Engineering
NIT Calicut

Abstract

This Project aims at developing a proactive based approach towards Network Security.The project goes in three phases. In the first phase we demonstrate a Vulnerability Scanner which will scan for various known vulnerabilties in computers and generates a report showing the possible vulnerabilities on the scanned host. In the second phase we show various types of attacks that are possible on networks.In the third phase we give a detailed description of defending techniques and tools that are available.

Acknowledgements

A successful project is a fruitful culmination of efforts by many people, some directly involved and some others indirectly, by providing support and encouragement. I express my sincere gratitude to everyone who had helped me in completing the undertaken project successfully.

I would like to thank **Dr. V. K. Govindan**, Professor & Head of the Department, CSED, NITC, for his constant support and encouragement throughout the project.

I would like to thank **Dr. M. P. Sebastian**, Assistant Professor & Staff incharge, Main Computer Center, CSED, NITC, for his immense help in providing all the resources and facilities.

With a profound sense of gratitude, I would like to express my heartfelt thanks to my project guide **Mr. Vinod Pathari**, Senior Lecturer, CSED, NITC, for his expert guidance, cooperation and immense encouragement. I am highly indebted to him for his commendable support and evincing keen interest in this project. Without his support, I could have not moved a step forward.

Last but not least, I also extend my thanks to the entire faculty and staff of the Department of Computer Engineering, NITC, who has encouraged me throughout the course of my Masters' Degree.

I also express my thanks to my friends, for their support and encouragement in the successful completion of this project work.

ROHAN P

Contents

Figures

Figure

Chapter 1

Introduction

Computer security[3] is the process of preventing and detecting unauthorized use of computer. Prevention measures help us to stop unauthorized users from accessing any part of our computer system. Detection helps us to determine whether or not someone attempted to break into our system. It is remarkably easy to gain unauthorized access to information in an insecure networked environment, and it is hard to catch the intruders. Even if users have nothing stored on their computer that they consider important, that computer can be a weak link, allowing unauthorized access to the organization's systems and information.

Three basic security concepts important to information on the Internet are confidentiality, integrity, and availability. Concepts relating to the people who use that information are authentication, authorization, and nonrepudiation.

When information is read or copied by someone not authorized to do so, the result is known as loss of confidentiality.

Information can be corrupted when it is available on an insecure network. When information is modified in unexpected ways, the result is known as loss of integrity. This means that unauthorized changes are made to information, whether by human error or intentional tampering.

Information can be erased or become inaccessible, resulting in loss of availability. This means that people who are authorized to get information cannot get what they need.

When a user cannot get access to the network or specific services provided on the network, they experience a denial of service.

Security is not a click and go process. One cannot install a software and think that his computer is safe from intruders.Computer networks undergo various types of attacks from various types of attackers.The attackers usually exploit a vulnerability present in computers and try to exploit the vulnerability to get access to the target system.It is important for a network administrator to periodically scan for vulnerabilities present and apply necessary patches.Vulnerabilities are present in computer systems because of various flaws in design and implementations of softwares and protocols and weak configuration of systems.

In the face of the vulnerabilities and incident trends , a robust defense requires a flexible strategy that allows adaptation to the changing environment, well-defined policies and procedures, the use of robust tools, and constant vigilance.

This paper illustrates a detailed mechanism for making computer networks more reliable and secure. The second chapter descibes about various vulnerabilities present in network protocols , operating systems and a GUI based tool that scans hosts for various vulnerable software that are installed and also scan for various ports for installed trojans. The third chapter describes about various types of attacks that are possible on networks and a tools that can actively attack networks to make the target vulnerable to denail-of-service. In the fourth chapter we discuss about various prevention mechanisms like firewalls ,Intrusion detection systems.

Chapter 2

Vulnerability Scanning

A vulnerability is a weakness that a person can exploit to accomplish something that is not authorized or intended as legitimate use of a network or system. When a vulnerability is exploited to compromise the security of systems or information on those systems, the result is a security incident. Vulnerabilities may be caused by engineering or design errors, or faulty implementation.

Many early network protocols that now form part of the Internet infrastructure were designed without security in mind. Without a fundamentally secure infrastructure, network defense becomes more difficult. Furthermore, the Internet is an extremely dynamic environment, in terms of both topology and emerging technology. Because of the inherent openness of the Internet and the original design of the protocols, Internet attacks in general are quick, easy, inexpensive, and may be hard to detect or trace. An attacker does not have to be physically present to carry out the attack. In fact, many attacks can be launched readily from anywhere in the world - and the location of the attacker can easily be hidden. Nor is it always necessary to "break in" to a site (gain privileges on it) to compromise confidentiality, integrity, or availability of its information or service.

Another factor that contributes to the vulnerability of the Internet is the rapid growth and use of the network, accompanied by rapid deployment of network services involving complex applications. Often, these services are not designed, configured, or maintained securely. In the rush to get new products to market,

developers do not adequately ensure that they do not repeat previous mistakes or introduce new vulnerabilities.

2.1 Types of Technical Vulnerabilities

The following taxonomy is useful in understanding the technical causes behind successful intrusion techniques, and helps experts identify general solutions for addressing each type of problem.

Flaws in Software or Protocol Designs

Protocols define the rules and conventions for computers to communicate on a network. If a protocol has a fundamental design flaw, it is vulnerable to exploitation no matter how well it is implemented. An example of this is the Network File System (NFS), which allows systems to share files. This protocol does not include a provision for authentication; that is, there is no way of verifying that a person logging in really is whom he or she claims to be. NFS servers are targets for the intruder community.

When software is designed or specified, often security is left out of the initial description and is later "added on" to the system. Because the additional components were not part of the original design, the software may not behave as planned and unexpected vulnerabilities may be present.

Weaknesses in the Implementation of Protocols and Software

Even when a protocol is well designed, it can be vulnerable because of the way it is implemented. For example, a protocol for electronic mail may be implemented in a way that permits intruders to connect to the mail port of the victim's machine and fool the machine into performing a task not intended by the service. If intruders supply certain data for the "To:" field instead of a correct E-mail address, they may be able to fool the machine into sending them user and password information or granting them access to the victim's machine with privileges to

read protected files or run programs on the system. This type of vulnerability enables intruders to attack the victim's machine from remote sites without access to an account on the victim's system. This type of attack often is just a first step, leading to the exploitation of flaws in system or application software.

Software may be vulnerable because of flaws that were not identified before the software was released. This type of vulnerability has a wide range of subclasses, which intruders often exploit using their own attack tools.The following examples of subclasses are included:

- race conditions in file access
- non-existent checking of data content and size
- non-existent checking for success or failure
- inability to adapt to resource exhaustion
- incomplete checking of operating environment
- inappropriate use of system calls
- re-use of software modules for purposes other than their intended ones

By exploiting program weaknesses, intruders at a remote site can gain access to a victim's system. Even if they have access to a nonprivileged user account on the victim's system, they can often gain additional, unauthorized privileges.

Weaknesses in System and Network Configurations

Vulnerabilities in the category of system and network configurations are not caused by problems inherent in protocols or software programs. Rather, the vulnerabilities are a result of the way these components are set up and used. Products may be delivered with default settings that intruders can exploit. System administrators and users may neglect to change the default settings, or they may simply set up their system to operate in a way that leaves the network vulnerable.

An example of a faulty configuration that has been exploited is anonymous File Transfer Protocol (FTP) service. Secure configuration guidelines for this service stress the need to ensure that the password file, archive tree, and ancillary software are separate from the rest of the operating system, and that the operating system cannot be reached from this staging area. When sites misconfigure their anonymous FTP archives, unauthorized users can get authentication information and use it to compromise the system.

With all the types of various vulnerabilties present in protocols and operating systems it is very important for an adminstrator to actively scan his network periodically to check for vulnerabilities.

Techniques to Exploit Vulnerabilities

As intruders become more sophisticated, they identify new and increasingly complex methods of attack. For example, intruders develop sophisticated techniques to monitor the Internet for new connections. Newly connected systems are often not fully configured from a security perspective and are, therefore, vulnerable to attacks.

The most widely publicized of the newer types of intrusion is the use of the packet sniffers described in the section above on packet sniffers. Other tools are used to construct packets with forged addresses; one use of these tools is to mount a denial-of-service attack in a way that obscures the source of the attack. Intruders also "spoof" computer addresses, masking their real identity and successfully making connections that would not otherwise be permitted. In this way, they exploit trust relationships between computers.

With their sophisticated technical knowledge and understanding of the network, intruders are increasingly exploit network interconnections. They move through the Internet infrastructure, attacking areas on which many people and systems depend. Infrastructure attacks are even more threatening because legitimate

network managers and administrators typically think about protecting systems and parts of the infrastructure rather than the infrastructure as a whole.

2.1.1 Viruses,Worms and Trojan Horses

Viruses[4], worms, and Trojan Horses are malicious programs that can cause damage to computers and information on computers, slow down the Internet, and use them to spread themselves to the rest of the Web.

virus A virus is a piece of computer code that attaches itself to a program or file so it can spread from computer to computer, infecting as it travels. Viruses can damage software, hardware, and files.

worms A worm, like a virus, is designed to copy itself from one computer to another, but it does so automatically by taking control of features on the computer that can transport files or information. Once a worm in the system it can travel alone.

Trojan Horses Trojan Horses are computer programs that appear to be useful software, but instead they compromise security and cause a lot of damage.

2.2 Scanner

The GUI based network scanner developed as a part of this project will actively check for various known vulnerabilities present in various operating systems and services.It will also scan for various number of ports for installed worms and trojans.The scanner will generate a report showing all the possible vulnerabilties and gives references to various patches available on the internet.

The tool first scans for various ports[9] open on the target host and save the necessary information in a special file. Then we check for various services running on the target like rpc, bind, statd, ssh and various vulnerable versions of ftp services Serv-U, Pro-ftpd, glftpd and irc services. It also scans for the webserver

for various vulnerabilities in their configuration and vulnerable CGI scripts. It scans for various vulnerable versions of mail services.

All the data from the scan is logged into a seperate file which can be viewed for later analysis. The report suggest various patches available on the internet for upgrade. It also scans for various known trojans running on the target host. The scanner detects trojans such as Deep Throat, Dm-setup, Ini killer, Dark shadow, Doly, Subseven, Doom, win crash etc.

We used perl and C for the backend and Qt Designer[8] for the frontend to develop the scanner. One thing that should be remembered is the vulnerabiltiy scanner does not actually exploit the vulnerability to check for it but checks for various known vulnerable services.

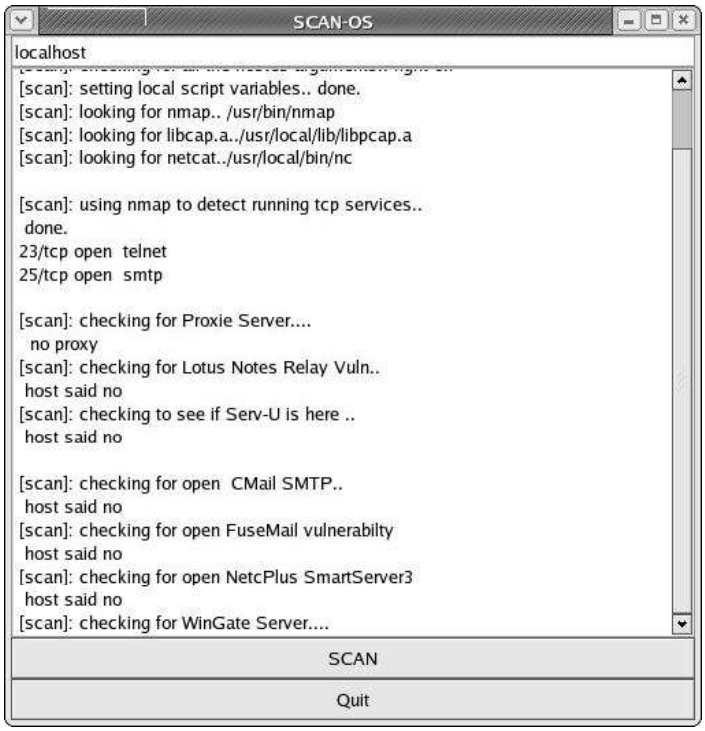

Figure 2.1: Vulnerability Scanner

Chapter 3

Network Security Incident

A network security incident is any network-related activity with negative security implications. This usually means that the activity violates an explicit or implicit security policy . Incidents come in all shapes and sizes. They can come from anywhere on the Internet, although some attacks must be launched from specific systems or networks and some require access to special accounts. An intrusion may be a comparatively minor event involving a single site or a major event in which tens of thousands of sites are compromised.

A typical attack pattern consists of gaining access to a user's account, gaining privileged access, and using the victim's system as a launch platform for attacks on other sites.

3.1 Types of Incidents

Incidents can be broadly classified into several kinds: the probe, scan, account compromise, root compromise, packet sniffer, denial of service, exploitation of trust, malicious code, and Internet infrastructure attacks.

Probe

A probe is characterized by unusual attempts to gain access to a system or to discover information about the system. One example is an attempt to log in to an unused account. Probing is the electronic equivalent of testing doorknobs

to find an unlocked door for easy entry. Probes are sometimes followed by a more serious security event, but they are often the result of curiosity or confusion.

Scan

A scan is simply a large number of probes done using an automated tool. Scans can sometimes be the result of a misconfiguration or other error, but they are often a prelude to a more directed attack on systems that the intruder has found to be vulnerable.

Account Compromise

An account compromise is the unauthorized use of a computer account by someone other than the account owner, without involving system-level or root-level privileges (privileges a system administrator or network manager has). An account compromise might expose the victim to serious data loss, data theft, or theft of services. The lack of root-level access means that the damage can usually be contained, but a user-level account is often an entry point for greater access to the system.

Root Compromise

A root compromise is similar to an account compromise, except that the account that has been compromised has special privileges on the system. The term root is derived from an account on UNIX systems that typically has unlimited, or "superuser", privileges. Intruders who succeed in a root compromise can do just about anything on the victim's system, including run their own programs, change how the system works, and hide traces of their intrusion.

Packet Sniffer

A packet sniffer is a program that captures data from information packets as they travel over the network. That data may include user names, passwords, and proprietary information that travels over the network in clear text. With perhaps hundreds or thousands of passwords captured by the sniffer, intruders can launch

widespread attacks on systems. Installing a packet sniffer does not necessarily require privileged access. For most multi-user systems, however, the presence of a packet sniffer implies there has been a root compromise.

Denial of Service

The goal of denial-of-service attacks is not to gain unauthorized access to machines or data, but to prevent legitimate users of a service from using it. A denial-of-service attack can come in many forms. Attackers may "flood" a network with large volumes of data or deliberately consume a scarce or limited resource, such as process control blocks or pending network connections. They may also disrupt physical components of the network or manipulate data in transit, including encrypted data.

Exploitation of Trust

Computers on networks often have trust relationships with one another. For example, before executing some commands, the computer checks a set of files that specify which other computers on the network are permitted to use those commands. If attackers can forge their identity, appearing to be using the trusted computer, they may be able to gain unauthorized access to other computers.

Malicious Code

Malicious code is a general term for programs that, when executed, would cause undesired results on a system. Users of the system usually are not aware of the program until they discover the damage. Malicious code includes Trojan horses, viruses, and worms. Trojan horses and viruses are usually hidden in legitimate programs or files that attackers have altered to do more than what is expected. Worms are self-replicating programs that spread with no human intervention after they are started. Viruses are also self-replicating programs, but usually require some action on the part of the user to spread inadvertently to other programs or systems. These sorts of programs can lead to serious data loss, downtime, denial

of service, and other types of security incidents.

Internet Infrastructure Attacks

These rare but serious attacks involve key components of the Internet infrastructure rather than specific systems on the Internet. Examples are network name servers, network access providers, and large archive sites on which many users depend. Widespread automated attacks can also threaten the infrastructure. Infrastructure attacks affect a large portion of the Internet and can seriously hinder the day-to-day operation of many sites.

3.2 Denial of Service Attacks

Moving up to the Internet protocol suite, the fundamental problem is : there is no real authenticity or confidentiality protection in most mechanisms. This is particularly manifest at the lower-level TCP/IP protocols. Consider, for example, the three-way handshake . This protocol can be exploited in a surprising number of different ways.

The following are a few exploits

SYN Flooding The SYN flood attack is, simply, to send a large number of SYN packets and never acknowledge any of the replies. This leads the recipient to accumulate more records of SYN packets than his software can handle.

Smurfing

Another common way of bringing down a host is known as smurfing. This exploits the Internet Control Message Protocol (ICMP), which enables users to send an echo packet to a remote host to check whether it's alive. The problem arises with broadcast addresses that are shared by a number of hosts. Some implementations of the Internet protocols respond to pings to both the broadcast address and their local address (the idea was to test a LAN to see what's alive). So the protocol allows both sorts of behavior in routers. A collection of hosts at a

broadcast address that responds in this way is called a smurf amplifier. The attack is to construct a packet with the source address forged to be that of the victim, and send it to a number of smurf amplifiers. The machines there will each respond (if alive) by sending a packet to the target, and this can swamp the target with more packets than it can cope with.

Distributed Denial-of-service Attacks

Rather than just exploiting a common misconfiguration as in smurfing, an attacker subverts a large number of machines over a period of time, and installs custom attack software in them. At a predetermined time, or on a given signal, these machines all start to bombard the target site with messages. So far, DDoS attacks have been launched at a number of high-profile Web sites, including Amazon and Yahoo. They could be even more disruptive, as they could target services such as DNS and thus take down the entire Internet. Such an attack might be expected in the event of information warfare; it might also be an act of vandalism by an individual.

Spoofing Spoofing is the creation of TCP/IP packets using somebody else's IP address. Routers use the "destination IP" address in order to forward packets through the Internet, but ignore the "source IP" address. That address is only used by the destination machine when it responds back to the source.

3.3 Attack Tool

The attack tool developed will attack the target host or network. implenentations of SYN Flooding , UDP Flooding ,Smurf Flooding have been successful on the target hosts.

The tool can be used as a penetration testing tool to actively test against

networks for denial of service. This is different from the other types of tests in the sense that we can test the target network as if we are attacking really disregarding the host even if it breaks down.

SYN Flooding : In this attack we create TCP packets [7] with spoofed source address and and sent to the target address. Flooding is done infintely until the target host is crashed or the attacker stops it. The attack is successful against many systems and the target was unable to serve the specified service. The target port can also be specified to disable a particular service like webserver or ftp services.

Smurf Attack : In this attack we created TCP packets with spoofed source address and we use a broadcast file in which we give various known broadcast addresses. Then we take each address as target and send icmp echo request packets with the spoofed source address. The broadcast addresses will forward the packets to all the systems under the broadcast address. All the systems responded with a icmp echo reply packets and the spoofed target was flooded with a sequence of icmp packets and clogged the target network using all the bandwidth.

UDP Flooding : In this attack we created UDP packets and sent infinite number of UDP packets to a udp service running on the target host and denying the corresponding service.

FTP Trojan : A trojaned ftp program with inbuilt keylogger has been implemented. This ftp program which once installed on a computer will automatically installs a keylogger and sends a mail periodically containing the keystrokes pressed.

The main thing we would like to suggest to defend such type of attacks is to use programs that are certified by certification authorities. And use rootkit detectors to check for modified programs and use latest versions of daemons like syslogd.

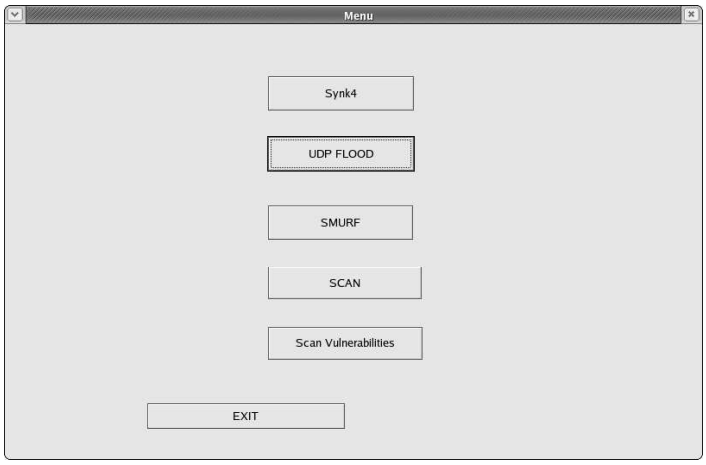

Figure 3.1: Attack Tool

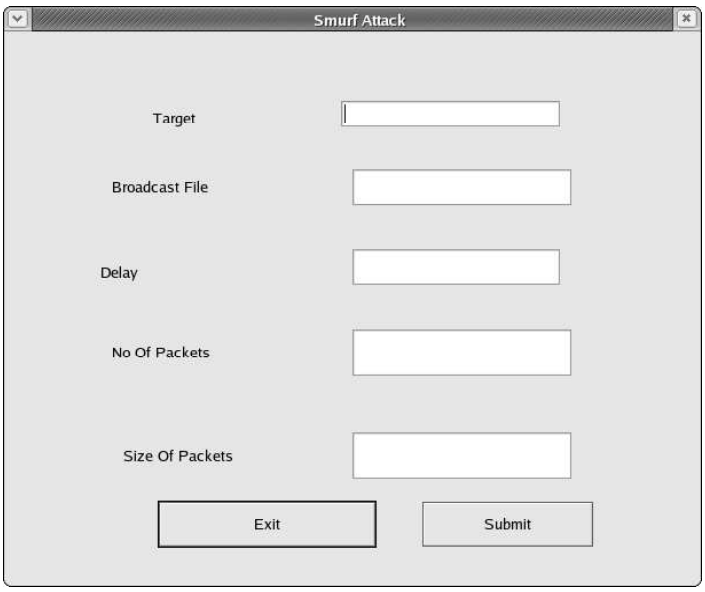

Figure 3.2: Smurf Attack

Chapter 4

Defences against Network Attacks

It might seen reasonable to hope that most attacks, at least those launched by script kiddies can be thwarted by a system administrator who diligently monitors the security bulletins and applies all the vendors' patches promptly to his software. This is part of the broader topic of configuration management.

4.1 Configuration Management

Tight configuration management is the most critical aspect of a secure network. One should be sure that all the machines in your organization are running up-to-date copies of the operating system, that all patches are applied as they're shipped, that the service and configuration files don't have any serious holes (such as world-writeable password files), that known default passwords are removed from products as they're installed, and that all this is backed up by suitable organizational discipline, then one can deal with nine and a half of the top ten attacks. (one should still have to take care with application code vulnerabilities such as CGI scripts, but by not running them with administrator privileges you can greatly limit the harm that they might do.) Configuration management is at least as important as having a reasonable firewall; in fact, given the choice of one of the two, one should forget the firewall. However, it's the harder option for many companies, because it takes real effort as opposed to buying and installing

an off-the-shelf product. Doing configuration management by numbers can even make things worse. Several tools are available to help the systems administrator keep things tight. Some enable to do centralized version control, so that patches can be applied overnight, and everything can be kept in synch; others, will try to break into the machines on the network by using a set of common vulnerabilities. Some familiarity with these penetration tools is a very good idea, as they can also be used by the opposition to try to hack you. What is appropriate is to say that adhering to a philosophy of having system administrators stop all vulnerabilities at the source requires skill and care; even diligent organizations may find that it is just too expensive to fix all the security holes that were tolerable on a local network but not with an Internet connection. Another problem is that, often, an organisation's most critical applications run on the least secure machines, as administrators have not dared to apply operating system upgrades and patches for fear of losing service. This leads us to the use of firewalls.

4.2 Firewalls

The most widely sold solution to the problems of Internet security is the firewall[10]. This is a machine that stands between a local network and the Internet, and filters out traffic that might be harmful. The idea of a "solution in a box" has great appeal to many organizations, and is now so widely accepted that it's seen as an essential part of corporate due diligence. Firewalls come in basically three flavors, depending on whether they filter at the IP packet level, at the TCP session level, or at the application level.

4.2.1 Packet Filtering

The simplest kind of firewall merely filters packet addresses and port numbers. This functionality is also available in routers and in Linux. It can block the kind of IP spoofing attack discussed earlier by ensuring that no packet that appears to come from a host on the local network is allowed to enter from outside. It can also stop denialof-service attacks in which malformed packets are sent to a host, or the host is persuaded to connect to itself (both of which can be a problem for people still running Windows 95). Basic packet filtering is available as standard in Linux, but, as far as incoming attacks are concerned, it can be defeated by a number of tricks. For example, a packet can be fragmented in such a way that the initial fragment (which passes the firewall's inspection) is overwritten by a subsequent fragment, thereby replacing an address with one that violates the firewall's security policy.

4.2.2 Ingress versus Egress Filtering

Egress filtering is the simple process of filtering of outbound traffic from the network. The purpose behind egress filtering is to prevent any packets with invalid or incorrect addresses from leaving one's site. Egress filtering generally occurs at the edge of a network, at the firewalls and border routers. At no time should your network send out any packets with addresses not legally assigned to you to do so means either your firewall may be misconfigured to show the world your internal address space, or worse, that you are the home of one or more DDOS attack agents. There should be very little effect or loss of functionality to the network when implementing egress filtering , all legitimate traffic requires legal addresses, so blocking anything else will only break things that should not be sent in the first place. If the site has already been compromised, connection to the

Internet degrade as firewalls and routers struggle to stop the traffic. Given the possibility of being brought into a lawsuit if the site is involved in a DDOS attack against another, finding and stopping bogus traffic from leaving the network at the cost of performance until we can clean up the DDOS agents is a small price to pay.

Ingress filtering[2] manages the flow of traffic as it enters a network under your administrative control. Servers are typically the only machines that need to accept inbound traffic from the public Internet. In the network usage policy of many sites, there are few reasons for external hosts to initiate inbound traffic to machines that provide no public services. Thus, ingress filtering should be performed at the border to prohibit externally initiated inbound traffic to non-authorized services.

4.2.3 Combinations

At really paranoid sites, multiple firewalls may be used. There may be a choke, or packet filter, connecting the outside world to a screened subnet, also known as a demilitarized zone (DMZ), which contains a number of application servers or proxies to filter mail and other services. The DMZ may then be connected to the internal network via a further filter that does network address translation. Within the organization, there may be further boundary control devices, including pumps to separate departments, or networks operating at different clearance levels to ensure that classified information doesn't escape either outward or downward. Such elaborate installations can impose significant operational costs, as many routine messages need to be inspected and passed by hand. This can get in the way so much that people install unauthorized back doors, such as dial-up standalone machines, to get their work done. And if your main controls are aimed at preventing information leaking outward, there may be little to stop a virus get-

ting in. Once in a place it wasn't expected, it can cause serious havoc.

4.2.4 Strengths and Limitations of Firewalls

Since firewalls do only a small number of things, it's possible to make them very simple, and to remove many of the complex components from the underlying operating system (such as the RPC and sendmail facilities in Unix). This eliminates a lot of vulnerabilities and sources of error. A firewall can only be as good as its configuration, and many organizations don't learn enough to do this properly. They hope that by getting the thing out of the box and plugged it in, the problem will be solved. It won't be. It may not require as much effort to manage a firewall as to configure every machine on your network properly in the first place, but it still needs some.

The big trade-off remains security versus performance. It is up to us to determine wheter to install a simple filtering router, which won't need much maintenance, or to go for a full-fledged set of application relays on a DMZ, which not only will need constant reconfiguration–as our users demand lots of new services that must pass through it, but will also act as a bottleneck.

Another issue with firewalls (and boundary control devices in general) is that they get in the way of what people want to do, and so ways are found round them. As most firewalls will pass traffic that appears to be Web pages and requests (typically because it's for port 80), more and more applications use port 80, as it's the way to get things to work through the firewall. Where this isn't possible, the solution is for whole services to be reimplemented as Web services (webmail being a good example). These pressures continually erode the effectiveness of firewalls. Finally, it's worth going back down the list of top ten attacks and asking how many of them a firewall can stop.

4.3 Intrusion Detection Systems

Intrusion Detection is the process and methodology of inspecting data for malicious, inaccurate or anomalous activity. At the most basic levels there are two forms of Intrusion Detection Systems that will encounter: Host and Network based. ntrusion Detection Systems can assist the administrator with notification when malicious or suspicious activity occurs. Host based can use logs as its data source, while Network based will use network traffic as its data source.

Host Based:

Host based Intrusion Detection Systems role is to identify tampering or malicious activity occurring on the system. This is achieved by monitoring log files, users, and the file system. Host based can use system logs, application logs, host traffic, and in some instances firewall logs as its data source. Some of activities that Host based can monitor include: Ability to monitor user specific actions:

Host based can monitor the file system for file permission changes, privilege escalation, and watch certain users. Any changes that can happen would be notified right away, some even have the ability to prevent these attacks from ever occurring.

Access to system log files, running processes, and files system:

Host based ID systems have the capability to watch system log files and search for certain strings/patterns and generate an alarm.

Ability to determine the success/failure of an attack:

Since Operating Systems log what events have occurred, it makes it extremely easier to determine the success/failure of an attack. This makes the rate of false positives reduced.

Attacks that use NIDS evasion techniques:

Many tools used by intruders come with NIDS evasion techniques built-in.

Network Intrusion Detection Systems that aren't updated regularly might miss the new method of NIDS evasion. Host based will be able to log the attack, log either failure or success of the attack.

Host based intrusion detection go hand in hand with Network based. The more information that is gathered during/after an attack can greatly increase the administrator's chances of identifying the source, type of attack, and hopefully thwarting further attacks.

Network Based:

Network based Intrusion Detection Systems (NIDS) can monitor both ingress and egress traffic. There are two forms of NIDS, Pattern Matching and Anomaly based. NIDS use network traffic as its daa source; monitoring network traffic in real time, and alerting in near real time.

Live Network Traffic:

The capability to use live network traffic as the data source reduces the chances of tampering, ensuring that what is captured is what is seen on the wire.

Detection of Attacks at Time of Occurrence:

Unlike their counterparts, network based can notify you the instant an attack was noticed. The faster an administrator is notified, faster they can respond. With faster notification, it can reduce the damage caused if the attack was successful.

Detect Unsuccessful Attacks:

The capability to detect unsuccessful attacks can bring light to malicious intent. The discovery of unsuccessful attacks can lead to notification of further attempts from the attacker.

As discussed above the basic level of intrusion detection systems have two forms: host and network based. Network based itself has two1 in two forms: Pattern Matching and Anomaly Baed, which are briefly discussed below.

Pattern Matching:

Most intrusion detection systems are pattern matching based systems. The intrusion detection system contains prior information about specific attacks and vulnerabilities. It appliesthis to ingress and egress traffic by inspecting each packet against its signature database. When such a condition is met, an alarm is triggered and the administrator is notified. The accuracy of a Knowledge based system relies on its signature databases.

Passive and Reactive Systems:

Host and Network based systems can either be passive systems or reactive based systems. Most network-based systems are passive with reactive capabilities. Passive systems detect possible attacks, log the information and issue an alert. Reactive systems attempt to react in some way to the malicious content it has spotted. Though reactive systems implement nice defensive mechanisms, they are still prone to false positives. One of the most widely used Network Intrusion Detection System is SNORT[6]. Snort has the ability to actively monitor and log all the events and produce alerts. Snort rules have to be updated periodically to detect new attacks. As a part of the project we have setup and configured a snort monitor. Configuring snort to fit our requirements is the most crucial part in setting up the IDS. Configuration mainly depends on the level of security we intend. If we are running a

Chapter 5

The Future

Research and development efforts are underway to allow critical applications to operate in the future in a more secure environment than exists today.

5.1 Internetworking Protocols

The IETF (Internet Engineering Task Force) Proposed Standard for the Next Generation Internet Protocol (IPng) is being designed to cope with the vastly increased addressing and routing needs associated with the exponential growth of the Internet. IPng provides integral support for authenticating hosts and protecting the integrity and confidentiality of data.

The first release of IPng is officially termed IPv6 (Internet Protocol version 6). Since it is impractical to replace the existing protocol instantly and simultaneously throughout the Internet, IPv6 is designed to coexist with the current version of IP, allowing for a gradual transition over the course of years. Implementations of IPv6 for many routers and host operating systems are underway.

In the future, authentication protocols will increasingly be supported by technology that authenticates individuals (in the context of their organizational or personal roles) through the use of smart cards, fingerprint readers, voice recognition, retina scans, and so forth.

Protocol design, analysis, and implementation will be the subject of continued research. A primary goal is 100% verifiably secure protocols (that is, pro-

tocols as provably secure as the cryptographic algorithms supporting them), but researchers are nowhere near attaining this goal.

5.2 Intrusion Detection

Research is underway to improve the ability of networked systems and their managers to determine that they are, or have been, under attack. Intrusion detection is recognized as a problematic area of research that is still in its infancy. There are two major areas of research in intrusion detection: anomaly detection and pattern recognition.

Research in anomaly detection is based on determining patterns of "normal" behavior for networks, hosts, and users and then detecting behavior that is significantly different (anomalous). Patterns of normal behavior are frequently determined through data collection over a period of time sufficient to obtain a good sample of the typical behavior of authorized users and processes. The basic difficulty facing researchers is that normal behavior is highly variable based on a wide variety of innocuous factors. Many of the activities of intruders are indistinguishable from the benign actions of an authorized user.

The second major area of intrusion detection research is pattern recognition. The goal here is to detect patterns of network, host, and user activity that match known intruder attack scenarios. One problem with this approach is the variability that is possible within a single overall attack strategy. A second problem is that new attacks, with new attack patterns, cannot be detected by this approach.

Finally, to support the needs of the future Internet, intrusion detection tools and techniques that can identify coordinated distributed attacks are critically needed, as are better protocols to support traceability.

5.3 Software Engineering and System Survivability

Current software engineering methods and practice have had only limited success in managing the intellectual complexity of designing and implementing software. Moreover, in the design of software systems, security concerns are typically an afterthought (addressed through add-ons and software patches) rather than being an integral part of the overall design. This means that software systems of any significant size and complexity are likely to have exploitable security flaws. Because managing the intellectual complexity of software is difficult, upfront security design in products is rare, and detailed knowledge about systems is widespread, systems will be breached in spite of our best efforts to make them invulnerable. Therefore, the concept of information systems security must encompass the specification of systems that exhibit behaviors that contribute to survivability in spite of intrusions. Only then can systems be developed that are robust in the presence of attack and are able to survive attacks that cannot be completely repelled.

System survivability is the capacity of a system to continue performing critical functions in a timely manner even if significant portions of the system are incapacitated by attack or accident. We use the term system in the broadest possible sense, which includes networks and large-scale “systems of systems”.

5.4 Web-Related Programming and Scripting Languages

Downloading interesting, informative, or entertaining ”content” from a remote site to a user’s local machine is central to the activity of Web browsing (or ”net surfing”). The content getting the most attention from Web users and the greatest concern from security experts is executable content, code to be executed on the local machine on download. This executable content may provide live audio of a conference in progress, a jazz tune, three-dimensional (3-D) animation effects, or

hostile code that destroys the local file system. Executable code is authored using one or more Web-related programming or scripting languages designed specifically for the production of platform-independent executable content. Languages in this category include JAVA and ActiveX. Executable content is called an "applet" in JAVA and a "control panel" in ActiveX.

Web-related programming languages pose new security challenges and concerns because code is downloaded, installed, and run on a user's machine without a review of source code (the recommended practice for secure use of publicly available software). These activities can be triggered by following any hypertext link or opening any page while browsing. A user may not even be aware that code has been downloaded and executed. Some Web-related programming languages, most notably JAVA, have built-in security features, but security experts are concerned about the adequacy of these features.

As executable content makes Web browsing even more alluring, further research in software engineering and greater user awareness will be necessary to counter security risks. Presently, the security of executable content depends upon the correctness of multiple vendors' implementations, the inherent security of platform-independent "virtual machines," and the safety of the source code that is executed. In the foreseeable future, users need to be educated about the risks so they can make informed choices about where to place their trust.

5.5 Intelligent Autonomous Agents - A New Computing Paradigm

The future Internet environment is likely to be increasingly dependent on an agent-based model of computing, with significant implications for Internet security. Agents are executable software objects with executions that are not tied to any specific host or computing resource or to any geographical or logical network location. Agents perform computation and communication defined by a user,

but the execution platforms are typically outside the user's administrative control (and outside the administrative control of the user's organization). The conceptual model of agent operation is one in which an intelligent agent, at the request of a user, goes to one or more remote hosts to perform a computation or gather information and then returns to the user with the result. An agent's mode of operation may range from partially to fully autonomous, and the degree to which an agent is autonomous may vary throughout the life of that agent.

Chapter 6

Conclusion

Preventing and detecting attacks that are launched over networks, and particularly over the Internet, is probably the most newsworthy aspect of security engineering. The problem is unlikely to be solved any time soon, as so many different kinds of vulnerability contribute to the attacker's toolkit. Ideally, people would run carefully written code on secure platforms; in real life, this won't always happen. But there is some hope that firewalls can keep out the worst of the attacks, that careful configuration management can block most of the rest, and that intrusion detection can catch most of the residue that make it through. Because hacking techniques depend so heavily on the opportunistic exploitation of vulnerabilities introduced accidentally by the major software vendors, they are constantly changing.

We suggest a step by step to attain a secure network.

- Use good backups for recovery purposes
- Use penetration testing tools as mentioned in section three prior to connecting to internet with in the LAN and check for possible denial of service attacks
- Periodically scan the network for various vulnerabilities and trojans using scanners as mentioned in section two

- Install latest anti-virus and anti trojan software
- Disable as many services as possible
- Upgrade the software with latest versions or patches
- Install a good firewall in front of the network
- Based on the level of security one can install an IDS to log various attacks and trace the actual attacker

Bibliography

[1] Comer D.E. Internetworking with TCP/IP 3 volumes. Prentice Hall of India Private Limited, May 1993.

[2] p.and D.Senie Ferguson. Network ingress filtering: Defeating denial of service attacks which ip source address spoofing.

[3] IETF. Site Security Policy Handbook RFC 1281.

[4] Denning P.J. Computers under attack: Intruders, worms, and viruses. 1990.

[5] Security information. [online] Available: http ://www.securityfocus.org.

[6] Snort ids. [Online]Available: http ://www.snort.org.

[7] Richard Stevens. Unix Network Programming. Prentice Hall of India Private Limited, May 1999.

[8] TrollTrench. Qt Assistant, October 2000.

[9] Vulnerable ports. [Online]Available: http://www.cert.org/current/serviceports.html, november 2003.

[10] Cheswik W.R. Firewalls and Internet Security:Repelling the Wily Hacker. Addison-Wesley, 1994.