Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide

Chapter No. 8
"Bypassing Firewalls and Avoiding Detection"
In this package, you will find:
A Biography of the author of the book
A preview chapter from the book, Chapter NO.8 "Bypassing Firewalls and
Avoiding Detection"
A synopsis of the book’s content
Information on where to buy this book

About the Author

Lee Allen is currently the Vulnerability Management Program Lead for one of the Fortune 500. Among many other responsibilities, he performs security assessments and penetration testing.

Lee is very passionate and driven about the subject of penetration testing and security research. His journey into the exciting world of security began back in the 80s while visiting BBS's with his trusty Commodore 64 and a room carpeted with 5.25-inch diskettes. Throughout the years, he has continued his attempts at remaining up-to-date with the latest and greatest in the security industry and the community.

He has several industry certifications including the OSWP and has been working in the IT industry for over 15 years. His hobbies and obsessions include validating and reviewing proof of concept exploit code, programming, security research, attending security conferences, discussing technology, writing, 3D Game development, and skiing.

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
I would like to thank my wife Kellie for always being supportive and my children Heather, Kristina, Natalie, Mason, Alyssa, and Seth for helping me perfect the art of multitasking. I would also like to thank my son-in-law Justin Willis for his service to our country. In addition, I would like to thank Kartikey Pandey and Michelle Quadros for their help and guidance throughout the writing process. A special thanks goes to Steven McElrea and Aaron M. Woody for taking the time to work through all of the examples and labs in the book and to point out my errors, it's people like you that make the security community awesome and fun!
Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide

Penetration testers are faced with a combination of firewalls, intrusion detection systems, host-based protection, hardened systems, and teams of knowledgeable analysts that pour over data collected by their security information management systems. In an environment such as this, simply running automated tools will typically yield few results. The false sense of this security can easily result in the loss of critical data and resources.

*Advanced Penetration Testing for Highly Secured Environments* provides guidance on going beyond the basic automated scan. It will provide you with a stepping stone which can be used to take on the complex and daunting task of effectively measuring the entire attack surface of a traditionally secured environment.

*Advanced Penetration Testing for Highly Secured Environments* uses only freely available tools and resources to teach these concepts. One of the tools we will be using is the well-known penetration testing platform BackTrack. BackTrack's amazing team of developers continuously update the platform to provide some of the best security tools available. Most of the tools we will use for simulating a penetration test are contained on the most recent version of BackTrack.

The Penetration Testing Execution Standard (PTES), http://www.penteststandard.org, is used as a guideline for many of our stages. Although not everything within the standard will be addressed, we will attempt to align the knowledge in this book with the basic principles of the standard when possible.

*Advanced Penetration Testing for Highly Secured Environments* provides step-by-step instructions on how to emulate a highly secured environment on your own equipment using VirtualBox, pfSense, snort, and similar technologies. This enables you to practice what you have learned throughout the book in a safe environment. You will also get a chance to witness what security response teams may see on their side of the penetration test while you are performing your testing!

*Advanced Penetration Testing for Highly Secured Environments* wraps up by presenting a challenge in which you will use your virtual lab to simulate an entire penetration test from beginning to end. Penetration testers need to be able to explain mitigation tactics with their clients; with this in mind we will be addressing various mitigation strategies that will address the attacks listed throughout the chapters.

What This Book Covers

Chapter 1, Planning and Scoping for a Successful Penetration Test, introduces you to the anatomy of a penetration test. You will learn how to effectively determine the scope of the penetration test as well as where to place your limits, such as when dealing with third-party vendor equipment or environments. Prioritization techniques will also be discussed.

Chapter 2, Advanced Reconnaissance Techniques, will guide you through methods of data collection that will typically avoid setting off alerts. We will focus on various reconnaissance strategies including digging into the deep web and specialty sites to find information about your target.

Chapter 3, Enumeration: Choosing Your Targets Wisely, provides a thorough description of the methods used to perform system footprinting and network enumeration. The goal is to enumerate the environment and to explain what to look for when selecting your targets. This chapter touches upon mid to advanced Nmap techniques and using PBNJ to detect changes on the network. The chapter closes with tips on how to avoid enumeration attempts as well as methods of trying to confuse an attacker (to buy time for the blue team).

Chapter 4, Remote Exploitation, will delve into the Metasploit® framework. We will also describe team based testing with Armitage. We take a look at proof of concept exploit code from Exploit-DB.com which we will rewrite and compile; we also take a look at THC Hydra and John the Ripper for password attacks.

Chapter 5, Web Application Exploitation, has a focus on web application attacks. We will begin by providing step-by-step instructions on how to build a web application exploitation lab and then move toward detailing the usage of w3af and WebScarab. Load balancing is discussed in detail as many environments now have these features. We introduce you to methods of detecting web application firewalls and load balancing with hands-on examples. We finish this chapter with an introduction to the Mantra browser.

Chapter 6, Exploits and Client-Side Attacks, discusses bypassing AV signatures, details the more advanced features of the Social Engineering Toolkit, and goes over the details of buffer overflows and fuzzing.

Chapter 7, Post-Exploitation, describes the activities performed after a successful attack has been completed. We will cover privilege escalation, advanced meterpreter functionality, setting up privileged accounts on different OS types, and cleaning up afterwards to leave a pristine system behind.

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
Chapter 8, *Bypassing Firewalls and Avoiding Detections*, covers methods that can be used to attempt to bypass detection while testing. This includes avoiding intrusion detection systems and advanced evasion techniques. We also discuss methods of increasing the detectability of malicious users or applications.

Chapter 9, *Data Collection Tools and Reporting*, will help you create reports and statistics from all of the data that you have gathered throughout this testing. You will learn how to collect all of the testing data and how to validate results. You will also be walked through generating your report.

Chapter 10, *Setting Up Virtual Testing Lab Environments*, walks you through setting up a test environment that mimics a corporation that has a multitier DMZ environment using IDS and "some" hardened systems and apps. This includes setting up VBOX, BackTrack, virtual firewalls, IDS and Monitoring.

Chapter 11, *Take the Challenge – Putting It All Together*, will allow you to gain hands-on experience using the skills you have learned throughout the book. We will set challenges for you that require you to perform a penetration test on your testing environment from start to finish. We will offer step-by-step solutions to the challenges to ensure that the material has been fully absorbed.

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
The type and scope of the penetration test will determine the need for being stealthy during a penetration test. The reasons to avoid detection while testing are varied; one of the benefits would include testing the equipment that is supposedly protecting the network, another could be that your client would like to know just how long it would take the Information Technology team to respond to a targeted attack on the environment. Not only will you need to be wary of the administrators and other observers on the target network, you will also need to understand the automated methods of detection such as web application, network, and host-based intrusion detection systems that are in place to avoid triggering alerts.

When presented with the most opportune target, take the time to validate that it is not some sort of honeypot that has been set up to trigger alerts when abnormal traffic or activity is detected! No sense in walking into a trap set by a clever administrator. Note that if you do find a system like this it is still very important to ensure it is set up properly and not inadvertently allowing access to critical internal assets due to a configuration error!

In this chapter, we will review the following:

- Pentesting firewalled environments
- Sliding in under the IDS
- Setting up shop internally
- Reviewing network traffic
- Using standard credentials
- Cleaning up compromised systems

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
Lab preparation
To follow along with the examples in this chapter a bit of lab preparation will be necessary.

Throughout this book there has been a strong focus on being able to emulate a target network. This is critical to being able to learn and practice the latest and greatest techniques as the excellent minds in the security research field continue to surprise us with new vulnerabilities and possible attack vectors. This book cannot cover every possible method of testing a network, but building the labs is an attempt at adding long lasting value that will hopefully lead to a lifetime of the "hacker mentality". If you continue to build out your personal lab and increase the difficulty of the practice challenges that you set for yourself you will quickly become comfortable with testing any sort of environment.

BackTrack, pfSense, and Ubuntu virtual machines should be configured in the following manner:

```
<table>
<thead>
<tr>
<th>BackTrack</th>
<th>192.168.75.0/24[Vlan1]</th>
</tr>
</thead>
</table>
| pfSense (Vlan1,Vlan2)   | 192.168.101.0/24[Vlan2]| Ubuntu
```

Certain configuration changes need to occur:

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
BackTrack guest machine

This machine will need to be connected to the 192.168.75.0/24 subnet. In the Oracle VM VirtualBox Manager console highlight the BackTrack instance and select the Settings option from the top navigation bar. Ensure that only one network adapter is enabled. The adapter should use the Vlan1 internal network option.

As previously described in Chapter 3, Enumeration: Choosing Your Targets Wisely we can assign the IP address (192.168.75.10 in this case) to an Ethernet adapter (eth0) from within BackTrack by typing the following command into a terminal:

```bash
# ifconfig eth0 192.168.75.10 netmask 255.255.255.0 broadcast 192.168.75.255 promisc
```

As the pfSense machine will need to be our router as well, we need to set it up as the default gateway. This can be accomplished as follows:

```bash
# route add default gw 192.168.75.1
```
Ubuntu guest machine
The Ubuntu machine will be used as the target. It needs to be configured to connect to VLAN2, which is a new internal network we have not used before. To create an internal network you will need to manually type VLAN2 into the network configuration screen in the Oracle VM VirtualBox Manager. Your settings should be similar to the following:

```
Network
```

<table>
<thead>
<tr>
<th>General</th>
<th>System</th>
<th>Display</th>
<th>Storage</th>
<th>Audio</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter 1</td>
<td>Adapter 2</td>
<td>Adapter 3</td>
<td>Adapter 4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Enable Network Adapter
- Attached to: Internal Network
- Name: VLAN2
- Advanced
- Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
- Promiscuous Mode: Allow Vmcs
- Mac Address: 080027543BC7
- Cable connected
- Port Forwarding

pfSense guest machine configuration
Configuring our firewall is a bit more work. It needs to be able to route restrictive traffic from the VLAN1 network to the VLAN2 subnet. There are several configuration changes we will need to make to ensure this works properly.

pfSense offers the option to reset to factory defaults from the configurations menu. Be aware that the adapters will have to be reconfigured if this option is chosen. This is not difficult, but all previous settings will be lost. Be sure to make a copy/snapshot of your pfSense machine if concerned with losing the previous configuration.

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
pfSense network setup

Our firewall guest machine will use two network adapters. One will be used for the VLAN1 segment and the other for the VLAN2 segment. VLAN1 will be treated as an untrusted wide area network for the examples within this chapter. Network Adapter 1 should resemble the following screenshot:

![Network Adapter 1 Screenshot]

Network Adapter 2 should be similar to the following:

![Network Adapter 2 Screenshot]

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
WAN IP configuration

The remaining networking setup will need to be performed from within the guest machine.

1. Boot up your pfSense virtual instance. There may be an additional delay as pfSense attempts to configure the WAN adapter. Allow it to fully load until you see the following menu:

```
--- Welcome to pfSense 2.0-RELEASE-pfSense (i386) on pfSense ---
   WAN (wan) 1e0 1e1
   LAN (lan) 0e0 1e1
0) Logout (SSH only) 9) Shell
1) Assign Interfaces 10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurer
3) Reset webConfigurer password 12) pfSense Developer Shell
4) Reset to factory defaults 13) Upgrade from console
5) Reboot system 14) Enable Secure Shell (sshd)
6) Halt system
7) Ping host
```

2. The WAN and LAN interfaces will need to be configured properly. Select option 2) Set interface(s) IP address.

3. Select option 1 – WAN.

```
Enter an option: 2
Available interfaces:
1 - WAN
2 - LAN
```

4. When asked to configure the WAN interface via DHCP type n for no.

5. The IP for the WAN adapter should be 192.168.75.1.
6. Subnet bit count should be set to 24. Type 24 and press Enter.

```
Configure WAN interface via DHCP? [yn] y
Enter the new WAN IPv4 address. Press <ENTER> for none:
> 192.168.75.1
Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
 e.g. 255.255.255.0 = 24
      255.255.0.0 = 16
      255.0.0.0 = 8
Enter the new WAN IPv4 subnet bit count:
> 24
Disabling DHCP... Done
Please wait while the changes are saved to WAN... Reloading filter...
DHCP...
The IPv4 WAN address has been set to 192.168.75.1/24
You can now access the webConfigurator by opening the following URL in your web
browser: http://192.168.75.1/
Press <ENTER> to continue.]
```

7. Press Enter to return to the configuration menu.

**LAN IP configuration**

We can set up the LAN IP information from the configuration menu as well. One benefit of configuring the LAN here is that we can have a DHCP server configured for VLAN2 at the same time.

1. Select option 2 from the configuration menu to start the LAN IP Configuration module.
2. Choose the LAN interface (Option 2).
3. When prompted to enter the IP address type 192.168.101.1.
4. The bit count should be set to 24.
5. When asked if you would like a DHCP server to be enabled on LAN choose y for yes.
6. DHCP Client IP range start will be 192.168.101.100.
7. DHCP Client IP range stop will be 192.168.101.110.
8. Press Enter.

```
Do you want to enable the DHCP server on LAN? [y/n] y
Enter the start address of the client address range: 192.168.101.100
Enter the end address of the client address range: 192.168.101.110
Please wait while the changes are saved to LAN... Reloading filter... DHCP...

The IP4 LAN address has been set to 192.168.101.1/24
You can now access the webConfigurator by opening the following URL in your web browser:
http://192.168.101.1/
Press <ENTER> to continue.
```

9. Press Enter again to return to the configuration menu.

Your LAN and WAN IP ranges should match the following:

```
--- Welcome to pfSense 2.0-RELEASE-pfSense (130b) on pfSense ---
WAN (wan)  ->  1e0  ->  192.168.75.1
LAN (lan)   ->  1e1  ->  192.168.101.1
```

**Firewall configuration**

pfSense can be configured using its intuitive web interface. Boot up the Ubuntu machine, open a terminal and perform a `sudo dhclient` to pick up an address from the pfSense DHCP server on VLAN2 (192.168.101.0/24). In a web browser on the Ubuntu machine type `http://192.168.101.1/` to access the configuration panel. If you have reset to factory defaults you will need to step through the wizard to get to the standard console.

The default username and password combination for pfSense is:

```
admin/pfsense
```

To view the current firewall rules choose **Firewall | Rules** and review the current configuration. By default the WAN interface should be blocked from connecting internally as there are not preestablished rules that allow any traffic through.

For More Information:

www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
For testing purpose, we will enable ports 80, 443, 21, and allow ICMP. Add the rules as follows:

1. Click on the **add a new rule** button displayed in the preceding screenshot.
2. Use the following rule settings to enable ICMP pass-through:
   - **Action**: Pass
   - **Interface**: WAN
   - **Protocol**: ICMP
   - **All others**: Defaults
3. Click on the **Save** button at the bottom of the screen.
4. Click on the **Apply Changes** button at the top of the screen.
5. Use the **Interface** | **WAN** navigation menu to enter the WAN interface configuration menu and uncheck **Block private networks**. Apply the changes and return to **Firewall** | **Rules**.

6. Click on the **add new rule** button.

---

For More Information:

7. Use the following rule settings to enable HTTP pass-through.
   - Action: Pass
   - Interface: WAN
   - Protocol: TCP
   - Destination port range: HTTP

8. Continue adding ports until the configuration matches the following:

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Protocol</th>
<th>Source</th>
<th>Port</th>
<th>Destination</th>
<th>Port</th>
<th>Gateway</th>
<th>Queue</th>
<th>Schedule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ICMP</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td>Block bogon networks</td>
</tr>
<tr>
<td>2</td>
<td>TCP</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td>80 (HTTP)</td>
</tr>
<tr>
<td>3</td>
<td>TCP</td>
<td>*</td>
<td>*</td>
<td>443</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>443 (HTTPS)</td>
</tr>
<tr>
<td>4</td>
<td>TCP</td>
<td>*</td>
<td>*</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21 (FTP)</td>
</tr>
</tbody>
</table>
```

At this point any machine connected to VLAN1 can communicate through the open ports as well as ping machines on the VLAN2 segment as can be seen in the following screenshot (this system running the scan is at 192.168.75.10):

```
root@ctt:# ping 192.168.101.100
PING 192.168.101.100 (192.168.101.100) 56(84) bytes of data.
64 bytes from 192.168.101.100: icmp_seq=1 ttl=64 time=1.31 ms
64 bytes from 192.168.101.100: icmp_seq=2 ttl=63 time=0.423 ms

root@ctt:# nmap -sS -T5 192.168.101.100
Starting Nmap 5.61 TEST4 ( http://nmap.org ) at 2012-01-20 14:19 EST
Nmap scan report for 192.168.101.100
Host is up (0.00057s latency).
Not shown: 997 filtered ports
PORT   STATE SERVICE
21/tcp open  ftp
80/tcp open  http
443/tcp open  https
```

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
Stealth scanning through the firewall
In this day and age, the most common security mechanism in place will be some sort of firewall. Firewalls are a great security mechanism when used in conjunction with other security controls; however, they must be properly maintained and monitored to be truly effective. There are several mechanisms that can be used to attempt to bypass these devices.

Finding the ports
It is important to know where you are being blocked when scanning. When testing through a firewall it may become difficult to prepare a stealthy attack if you do not have all of the information. Remember that tools such as Firewalker or Hping can assist with determining where the block occurs and if the port is truly available or just closed. Although this may seem trivial, knowing if there is a firewall in the first place is fairly important as well.

Traceroute to find out if there is a firewall
Sometimes we can use traceroute to see the path to the target system. Let’s take a look at a open traceroute from VLAN2 to VLAN1:

```
student@Phobos:~$ traceroute 192.168.75.10
traceroute to 192.168.75.10 (192.168.75.10), 30 hops max, 60 byte packets
  1  pfSense.localdomain (192.168.101.1)  0.248 ms  0.166 ms  0.117 ms
  2  192.168.75.10 (192.168.75.10)  1.351 ms  1.243 ms  1.188 ms
```

Looking at this result we can see that the first hop goes through our gateway at 192.168.101.1 before being routed to the host. Now we will try the reverse from the BackTrack machine:

```
root@bt:~# traceroute 192.168.101.1
traceroute to 192.168.101.1 (192.168.101.1), 30 hops max, 60 byte packets
  1  * * *
  2  * * *
  [Truncated...]
  30  * * *
```

Something is blocking us from receiving the path information (it’s the pfSense firewall configuration). This technique is not always useful, but definitely good to know about.
Bypassing Firewalls and Avoiding Detection

Finding out if the firewall is blocking certain ports

There is a firewall; now what? The next step is to determine which ports are being blocked by the firewall, or more importantly which are open.

Hping

Hping2 and Hping3 are included as part of the BackTrack 5 distribution. It can be accessed via the GUI navigation bar Applications | BackTrack | Information Gathering | Network Analysis | Identify Live Hosts | Hping2. It can also be invoked at the command line by simply typing: hping2. Hping2 is a powerful tool that can be used for various security testing tasks. The following syntax can be used to find open ports while remaining fully in control of your scan:

```
root@bt:/pentest# hping2 -S 192.168.101.100 -c 80 -p ++1
```

```
HPING 192.168.101.100 (eth0 192.168.101.100): S set, 40 headers + 0 data bytes
len=46 ip=192.168.101.100 ttl=63 DF id=0 sport=21 flags=SA seq=20 win=5840 rtt=0.6 ms
len=46 ip=192.168.101.100 ttl=63 DF id=0 sport=80 flags=SA seq=79 win=5840 rtt=0.6 ms

--- 192.168.101.100 hping statistic ---
80 packets tramitted, 2 packets received, 98% packet lossound-trip min/avg/max = 0.6/0.6/0.6 ms
```

This command allowed us to perform a SYN scan starting at port 1 and incrementing for 80 steps.

```
CTRL + Z is used to manually increment ports. Start low and work your way up manually. Start an Hping2 scan and give it a try!
```

Depending on the firewall configuration it may also be possible to send spoofed packets. During a test it is beneficial to ensure that the configuration does not allow for this behavior to occur. Hping is perfectly suited for this task. The following is an example of how you may test if the firewall allows this traffic to pass:

```
hping2 -c10 -S --spoof 192.168.101.101 -p 80 192.168.101.100
```

This command will spoof 10 packets from 192.168.101.101 to port 80 on 192.168.101.100. This is the basis for an idle scan and if successful would allow you to hping the 192.168.101.101 machine to look for an increase in the IP sequence number. In this case we could enable monitoring on the pfSense machine to emulate what this traffic looks like to a network administrator reviewing the logs.

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
Challenge yourself to create and monitor different packets and uses of Hping so that you can gain a good understanding of the traffic flow. The best means of remaining undetected while testing is to fully understand the technology that is being used.

Take a look at the logs generated from a successful scan and keep in mind that due to the amount of traffic involved even secured networks will sometimes only log and trigger events based on denied traffic.

Logging per rule will need to be enabled on the firewall to see allowed traffic. Not logging permitted traffic is fairly standard practice as it reduces the firewall log size. Educate your clients that proactively monitoring allowed traffic can also be beneficial when attempting to truly secure a network.

The granular control of hping2 in combination with the scripting capabilities of hping3 makes the Hping tool an invaluable addition to every pentesters toolbox.

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
Bypassing Firewalls and Avoiding Detection

Further information and tutorials about how to effectively use Hping2 and Hping3 can be found at the Hping wiki: http://wiki.hping.org/.

**Nmap firewalk script**

One of the easiest methods to test open ports on a firewall is to simply use the firewalking script for Nmap. To test the open firewall ports you will need a host behind the firewall as the target:

```
nmap --script=firewalk --traceroute 192.168.101.100
```

The command sequence is straightforward and familiar: we invoke `nmap`, use the `script` option, and choose the `firewalk` script. We then provide the input that `firewalk` needs by performing a traceroute to `192.168.101.100` which we know is behind our target firewall.

```
root@bt:/pentest# nmap --script=firewalk --traceroute 192.168.101.100
Starting Nmap 5.61TET4 ( http://nmap.org ) at 2012-01-20 20:03 EST
Nmap scan report for 192.168.101.100
Host is up (0.00100s latency).
Not shown: 997 filtered ports
PORT   STATE SERVICE             VERSION
21/tcp open  ftp
80/tcp open  http
443/tcp open  https
```

```
Host script results:
<table>
<thead>
<tr>
<th>firewall:</th>
<th>HOP HOST</th>
<th>PROTOCOL</th>
<th>BLOCKED PORTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>192.168.75.10</td>
<td>tcp</td>
<td>1,3-4,6-7,9,13,17,19-20</td>
</tr>
</tbody>
</table>
```

```
TRACEROUTE (using port 443/tcp)
HOP RTT ADDRESS
1 0.40 ms 192.168.75.1
2 1.22 ms 192.168.101.100
```

Although we were able to determine which ports on the firewall were open (21, 80, and 443), if you take a look at the firewall denies it quickly becomes apparent that this is not a quiet test and should only be used when stealth is not needed. What this boils down to is that stealth requires patience and a well made plan of action. It may be easier to manually verify if there are any common ports open on the firewall and then try to scan using one of the well-known ports.

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
To effectively emulate proper firewalking or port probing with Hping the network would need to have a gateway behind the firewall. This can be accomplished in a lab when replicating a production environment but is beyond the scope of this chapter. The commands remain the same; the information gained can increase dramatically. These tools use TTL to determine if a port is open or not and as our gateway is on the same machine as our firewall and router the results are varied and obscured.

All in all, idle scans remain the best method of determining what is behind a properly locked down firewall. The flavor of the moment is SYN Cache Idle scanning and a great paper about this subject titled *Idle Port Scanning and Non-interference Analysis of Network Protocol Stacks Using Model Checking* written by Roya Ensafi, Jong Chun Park, Deepak Kapur, and Jedidiah R. Crandall, University of New Mexico can be found at: http://www.usenix.org/events/sec10/tech/.

### Now you see me, now you don't — Avoiding IDS

In a secured environment you can count on running into IDS and IPS. Properly configured and used as part of a true defense in depth model increases their effectiveness tremendously. This means that the IDS will need to be properly updated, monitored, and used in the proper locations. A penetration tester will be expected to verify that the IDS's are working properly in conjunction with all other security controls to properly protect the environment.

For More Information:  
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
The primary method of bypassing any IDS is to avoid signatures that are created to look for specific patterns. These signatures must be fine-tuned to find only positively malicious behavior and should not be so restrictive that alerts are triggered for normal traffic patterns. Over the years, the maturity level of these signatures has increased significantly, but a penetration tester or knowledgeable attacker will be able to use various means to bypass even the most carefully crafted signatures. In this section, we review some of the methods that have been used by attackers in the wild.

**Canonicalization**

Canonicalization refers to the act of substituting various inputs for the canonical name of a file or path. This practice can be as simple as substituting hexadecimal representations ASCII text values. Here is an example of an equivalent string:

- **String A in text**: "This is a string"
- **String A in ASCII**: "084 104 105 115 032 105 115 032 097 032 115 116 114 105 110 103"

By taking advantage of the fact there are sometimes literally thousands of combinations possible for a single URL. To put this into perspective, let's take a look at the address we can use to get from our browser to our local Ubuntu Apache server:

http://2130706433/

Luckily, this address confuses our Apache server and we receive the following message:

![Bad Request](attachment:image)

For More Information:

www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
The previous request attempted to load the local page at 127.0.0.1. Let's see what occurs when we try to load the remote pfSense administration console in the same manner:

http://3232254721/

Here we are warned by the web server hosting the pfSense administrative console that a potential DNS Rebind attack occurred:

Let's try something else that actually works properly:

In the console, ping one of the addresses we listed above:

PING 3232254721 (192.168.75.1) 56(84) bytes of data.
64 bytes from 192.168.75.1: icmp_seq=1 ttl=64 time=9.34 ms
64 bytes from 192.168.75.1: icmp_seq=2 ttl=64 time=0.265 ms
64 bytes from 192.168.75.1: icmp_seq=3 ttl=64 time=0.292 ms

As we can see, the IP address resolved properly and we receive our replies as expected. This very same concept is key when trying to bypass an IDS rule. If the type of IDS can be determined, then it should be possible to get the signatures. When reviewing these signatures you would look for opportunities to obscure the URLs, filenames, or other path information enough that it is able to bypass the existing ruleset.

Try this out with commonly found websites. Many web servers will properly interpret these URLs and serve the page. This can be interesting when used in combination with social engineering campaigns as well. Obscuring a URL in a phishing e-mail will lead to more clicks from users who are not properly trained.
Timing is everything

In previous chapters we have already reviewed that timing can be critical when performing a network scan on a secured environment. Using Nmap we can adjust the number of packets that are sent in a given timeframe. IDS signatures look for patterns, and sending packets out to many machines in a short timeframe is a definite pattern.

When attempting to bypass these mechanisms it is important to understand the logic behind the devices and how they work. If your traffic does not match what is normally seen on a network there is good possibility that you will be blocked before there is a chance to gain much information. This can be frustrating at best and lead to a failed assessment at worst. Take your time and plan out the stages needed for a successful test. It is better to start off slow and determine which type of security mechanisms are in place than to rush in and hit every possible port in the world and get your testing IP ranges auto-banned.

Nmap and many other tools have the granularity and ability to restrict the timing of your scans. It may even be advisable to begin with some manual controlled network enumeration of specific ports that are suspected to be open rather than starting with an automated scan.

Blending in

Launching attacks internally can be both satisfying and rewarding. You will no longer be restricted by the protected outer shell of the network and can traverse at will. Be careful that the tools used do not give you away.

By understanding what an administrator would see under certain conditions a penetration tester is more likely to perform well thought-out work that is in line with the final goal of the test as described in the rules of engagement contract.

Here we have a connection from a BackTrack machine to a Kioptrix level 1 machine. Take a look at the strange traffic being logged by the firewall:
Now if we were to quickly log into the system and set up or escalate privilege of a user account to allow us SSH capability we could merge with the existing traffic on the network. Let’s take a look at the difference when we are logged into SSH now while running the `tree` command in the SSH session:

```
bash-2.05# tree | head
```

```
.  
  |-- X11R6
  |   |-- bin
  |       |   |-- fslsfonts
  |       |   |-- fstobdf
  |       |   |-- mkfontdir
  |       |   |-- xfs
  |       |     `-- xfsinfo
  |       |-- include
  |       |-- lib |
  [Output Truncated...]
  |       |-- i686
  |           |   `-- noarch
  |           |-- SOURCES
```
Bypassing Firewalls and Avoiding Detection

```
|-- SPECS
|-- SRPMS
  `-- tmp -> ../var/tmp
```

2093 directories, 33808 files

bash-2.05#

While this command is passing back the entire directory structure of the Linux box we see the following in the firewall logs:

![Firewall Logs](image)

Note that there are no entries for the SSH traffic. It is minimal compared to the previous port 139 traffic. With proper scripting the work that is done via post exploitation modules can be emulated from within an SSH connection as well, and this traffic is completely encrypted and likely to be used by various administrators throughout the network being tested.

### Looking at traffic patterns

Network sniffing can be a huge time saver. It is more difficult to use remote Windows machines to perform this task for you as the network card needs to be in promiscuous mode, but it can be done. Ideally, you will find a Unix or Linux host that can be turned into a listening station with little to no effort.

For More Information:

www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
Here we look at a compromised Linux host on the 192.168.101.0/24 subnet. Our attacking machine resides on 192.168.75.0/24 and cannot see the same traffic that the Linux machine does. We will use `tcpdump` which is readily available to many Linux distributions:

```
tcpdump -i eth0 -c 100 -n
```

Here we invoke `tcpdump` on the remote Kioptrix machine we have SSH'd into using the games account we set up during the post exploitation chapter. We use the `-i` option to specify that we would like to use `eth0` as our listening adapter. We then tell the adapter to only capture the next 100 packets. The `-n` switch is used to avoid DNS lookups and will display IP numbers rather than hostnames. The output from this command will provide us with unfiltered packet information that is primarily related to our SSH connection.

What is more interesting is to see what else is traversing that segment. Using a simple filter for `icmp` for instance we can see the following:

```bash
bash-2.05# tcpdump icmp
tcpdump: listening on eth0
00:43:52.370998 192.168.101.199 > pfSense.localdomain: icmp: echo request (DF)
00:43:52.370998 192.168.101.199 > pfSense.localdomain: icmp: echo reply (DF)
00:43:53.370998 192.168.101.199 > pfSense.localdomain: icmp: echo request (DF)
00:43:55.370998 192.168.101.199 > pfSense.localdomain: icmp: echo reply (DF)
00:45:56.370998 192.168.101.199 > pfSense.localdomain: icmp: echo request (DF)
00:45:57.370998 192.168.101.199 > pfSense.localdomain: icmp: echo reply (DF)
00:45:58.370998 192.168.101.199 > pfSense.localdomain: icmp: echo request (DF)
```

Looking at the preceding screenshot we can determine that there are additional units on this subnet. The great part about using `tcpdump` in this manner is that we are not interfering with traffic and simply sifting through information as it passes on the wire.
Cleaning up compromised hosts

When dealing with a small network it is easy to underestimate the time and effort it can take to clean up your compromised hosts. This task is critical in both avoiding detection and in leaving the network in pristine condition once your testing has been completed. The last thing anyone wants is to overlook a compromised host that has a meterpreter backdoor installed and waiting for the next person to come along and take advantage of! The key is to take meticulous notes and keep accurate record of not only what was done while testing, but also if the things that were done could possibly persist after testing.

Think about what we did in the post exploitation chapter; just how easy do you think it would be to forget that we enabled the games account to be used for SSH login—and with root privilege and a weak password at that! It seems the only thing worse would be to accidently send the wrong report to a client and give away someone’s confidential information. It may seem that people would never do either of these things, but there is a small chance that either could happen if proper planning and organization is not used. When dealing with one, two, or even five machines going back and cleaning up may not be a big concern or worry. What happens when you have 1000 machines on 40 different subnets though?

Using a checklist

If you have not scripted the full exploitation and post-exploitation process then make sure you are keeping a checklist for all actions that must be undone. This is above and beyond creating notes and logging commands for your final report. We are talking about the guide that will be used to ensure that nothing is left to chance and ALL changes are reversed properly – something as small as adding a temporary file to a world writable directory so that you could test your blind SQL injection. If you cannot remove the file yourself, have something ready for the administrator to remind them to remove the files for you. The job of a penetration tester is to assist in verifying the security of an environment, not to make it more vulnerable.

When to clean up

It is never too early to begin the cleanup process. Not only will this assist in remaining undetected, but it also ensures that a systematic approach is used throughout the entire penetration test.

There is no need to have 300 open shells to the same subnet. Pick a target that allows you to set up a proper pivot and then remove the other shells from your list. The fewer machines you have to touch, the easier the cleanup will be. You will need the additional time for reporting and verifying results anyhow!
Local log files

It is critical to have a good understanding of where the log files are stored, what they capture, and how they report the data back to the administrator. Take the time to learn about the various log files for at least the most widely used operating systems such as popular Linux distributions and Windows Servers. If attempting to avoid detection, simply erasing the logs will probably not help achieve the desired result. It would be akin to taking someone's ice cream cone, eating the ice cream and returning the cone back to the freezer. Someone is going to notice. Instead use techniques that allow you to edit portions of the log files or escalate privilege to an account that is not monitored. Many of the tasks needed to enumerate an internal network do not require administrative privileges; maybe it would be better to use a restricted account for those activities in hopes that only admin actions are being logged and monitored?

Administrators that actually review logs are not going to look for the standard traffic. They will be looking for anomalies. In order to avoid detection your traffic and actions must be able to merge with those of an average user.

Miscellaneous evasion techniques

The level of detection avoidance that can be accomplished varies from network to network. When performing the test keep in mind that in this day and age, resources are usually very limited and administrators are overworked and underappreciated. Focus on bypassing the automated detection methodologies and you are unlikely to be found by an active and eager admin unless your traffic and behavior patterns are drastically different from those of the average power user. When sniffing traffic and looking at network connections and activity you should be able to get an idea of what is considered normal traffic on the network.

Divide and conquer

When performing scans it may be a good idea to use multiple sources to originate the scan from. This is more likely to be possible in large networks after a few people have clicked the links to your social engineering campaign page. Once you have several machines under your control it is not advisable to scan from a single machine. Use the tools to break the scans into chunks and to reduce the scan times. Take advantage of idle scans, especially when there are network enabled printers available.

For More Information:
www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
Hiding out (on controlled units)

If any of the systems you have control of start to be cleaned, reimaged, or otherwise remediated before the actual penetration test has been completed, slow down or at minimum cease all aggressive testing until it can be determined who or what is taking control of remediating the systems. There may be a third party involved in which case it will become extremely important that your traffic and efforts are not confused with those of the third party, especially if that person or group turns out to be malicious in nature and are trying to ensure they do not lose control of "their" owned systems to a rival group or person. In a perfect world this would not be the case and instead there is just a very good security and administrative group taking care of business and eliminating threats as they occur.

File integrity monitoring

One security measure that we did not discuss often in this book is the usage of File Integrity Monitoring. Proper usage of this control can be devastating to an attacker and penetration tester alike. It is very simple for an administrator to use these tools to let them know when key files or directories have changed. Keep this in mind when running into those wide open systems that are just waiting to be completely pillaged. One improper change and the administrator and possibly security group will go into overdrive and start to look for the smallest anomalies on the network. This will guarantee that your job just got much more difficult.

FIM can usually be avoided by sticking to non-intrusive means of post exploitation and enumeration. Some directories and files, particularly those dealing with databases or temporary files, will not be scanned for changes due to the high rate of false positives. Ensure that any files you modify or drop are in those directories, and stay away from attempts at changing key system files. (Log files may be included in this!) Once again, think like an administrator and avoid any action that could easily be scripted to alert.

Using common network management tools to do the deed

Last but not least: Use the tools at hand to perform enumeration and further exploitation. If the targeted system has a compiler installed, use it to compile your own network scanner instead of going to some random website from the machine and downloading one. Windows machines in particular have a broad range of Net commands and shell commands that make many enumeration and pillaging tasks a breeze. Use these tools to their fullest extent when performing your testing and you will probably not be detected by the administrators.

For More Information:

www.packtpub.com/advanced-penetration-testing-for-highly-secured-environments/book
Summary
In this chapter, we learned how to set up firewall rules in pfSense and monitor our traffic so that we can learn what type of activity is loud and which type is not. We also discussed how an IDS works and how we can take advantage of that knowledge to avoid detection when performing our scans, starting social engineering campaigns, or simply assessing a web application.

We discussed traffic patterns and how attempting to match the traffic will assist in avoiding detection; after all, if all of the information looks the same how can anyone determine what is legitimate and what is not.

Also discussed were various strategies of how detection avoidance may be possible if testing in a strategic and well thought-out manner. In closing, the mindset necessary to effectively and efficiently avoid detection was touched upon as well.

In the next chapter, we will take a look at data collection tools and reporting. This is an important aspect of penetration testing and as such should not be overlooked. We take a look at generating a final report as well as providing a quick overview of effectively using tools such as vim, nano, NoteCase, and Dradis to keep track of your testing efforts.
Where to buy this book


Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.