Global Information Assurance Certification Paper

Firewall Fingerprinting:

Using default TCP/UDP port combinations and Nmap to identify firewall types in a network

By Charles Hamby

GIAC Security Essentials Certification
Practical Assignment
Version 1.4b
14 February 2003

Abstract

The basic purpose of a firewall, whether corporate or SOHO is to prevent unauthorized access to or from a private network **(Webopedia)**. Firewalls do this by, among other things, controlling which TCP and UDP ports traffic is allowed in and out of. Historically, firewalls have been very difficult to install, configure, and maintain. In an attempt to rectify this problem, and in order to add new features to their products, such as remote management and VPN support, firewall vendors have added in additional user-friendly features to make maintenance and management easier as well as to provide these features. In many cases the side-effect of this has been to open up additional TCP and/or UDP ports in the firewall to accommodate these features. In many cases the port numbers associated with these services are unusual enough that they can be recognized and associated with the particular vendor. This opens up the possibility of "fingerprinting" a firewall based on the results of a port scan of the firewall.

Introduction

There are currently a wide variety of firewalls and firewall-type products on the market: everything from software-based desktop firewalls such as Zone Alarm all the way to larger, more expensive firewalls such as the Cisco PIX series of firewalls. What many of these firewalls seem to have in common is an increasing trend towards ease of use and ease of administration. While this trend is not necessarily a bad thing (nobody wants firewalls to be harder to manage, after all), there is at least one possible ramification to this trend towards ease of management.

Several firewalls currently on the market open various TCP and/or UDP ports by default for a wide variety of purposes. As a general rule these ports are generally open for purposes such as remote management, inter-device communication, or for VPNs. While not a bad thing in and of itself, it does present a rather unique situation: The corporate firewall, which is supposed to be monitoring and locking down access to these ports, can in some cases be opening up new ones for its own use.

This fact opens up the intriguing possibility of fingerprinting firewalls using known TCP/UDP port combinations in much the same manner that Nmap uses malformed packets to fingerprint operating systems. If a hacker is aware that a specific port or a combination of ports is associated with a certain firewall (a Watchguard Firebox II, for example), he would be able to use a port scanner to identify the firewall with a reasonable degree of accuracy. While not an exploit in and of itself, this would still be a potentially valuable reconnaissance technique, as it would allow the hacker to determine the manufacturer (and possibly the model in some cases) of the firewall being run by an organization. The hacker could then simply then search for known vulnerabilities for the identified firewall type and proceed to try the exploit(s) against the target firewall.

It should be noted at the outset however, that not all firewalls that implement remote management or other features will immediately be vulnerable to this type of enumeration. Through the judicious use of port scanners and tools such as Fyodor's Nmap, it can be relatively easy to identify whether or not a firewall or firewall-like device is leaking information of this type.

Of course, the best defense against this problem is simply to close down the offending port(s) if at all possible. If not, it may be necessary to block the port(s) at a perimeter device located further out in your network (such as a border router) in order to prevent the identification of your firewall. Changing the default port number is also a valid method to deter fingerprinting; unfortunately some firewalls do not allow you to change the default port numbers. The use of a Network Intrusion Detection System (NIDS) ruleset to detect scans against known ports is also advisable.

Problem Discussion

With the implementation of an ever-increasing feature set in the modern firewall, it has become almost inevitable that some feature would or could wind up being used to thwart the purpose for which the firewall was put in place.

The complexity of the modern firewall is such that ease of use features are almost a must as a selling point. Unfortunately, all this ease of use must come at a price. That price is small, but it is a price. Some of those features, such as remote management listen on a port on the firewall. This may not seem like a problem at first, but when you consider that the purpose of a firewall is to tightly control the flow of traffic between the LAN side and the WAN side of the network by restricting which ports are open, closed, and filtered, this can potentially be an issue, depending on the security restrictions of your institution. Compounding the problem is the fact that many firewall vendors use "odd" port numbers for management (apparently in an attempt to keep them from conflicting with other, known services). The net effect of all of this is a company firewall essentially announcing its identity to every passing port scanner. At first glance it's hard to see the problem here. What is the harm in someone knowing you have, say a Checkpoint-1 or a Cisco PIX firewall? The answer is simple. Would you want every scanning script kiddie to know your company had a Checkpoint-1 if a new vulnerability had just been discovered for it?

The question that needs to be answered at this point is how to tell the difference between a particular type of firewall and a different type of firewall masquerading as that brand. For example, how does one tell the difference between a Watchguard Firebox and a Checkpoint-1 that just happens to have TCP 4100 open? Since TCP 4100 is a rarely used port, you could rely on the fact that it isn't generally used by anything other than Watchguard and assume that when

you see it, you have found a Watchguard firewall, but there might be a better way.

Probably one of the best known security tools currently out there is Fyodor's Nmap. Nmap can be used for a wide variety of purposes, from port scanning to remote OS detection. The remote OS detection function is typically used against workstations, servers, printers, etc. in order to help you determine the operating system of the target to "adjust your shellcode accordingly" **(Fyodor)**.

While the built-in OS detection of Nmap does not currently know about or understand some of the firewall operating systems, nevertheless, it can still be used to fingerprint them. In order to successfully fingerprint a firewall using Nmap, one of two conditions must be met:

1) The firewall must be running an operating system that Nmap recognizes and can fingerprint successfully.
2) There must be a firewall of the type available in a laboratory setting available to test (a known quantity) to determine the fingerprint.

The advantage to this method is that once a fingerprint for the firewall has been established with a reasonable to high degree of certainty not only does it reduce or preclude the possibility of firewall masquerading, but even if all the ports on the firewall are filtered Nmap may still be able to provide a TCP/IP fingerprint, thus giving you enough to fingerprint the firewall.

Case Study 1: Watchguard Firebox Firewalls

Probably one of the best documented examples of a port that can be easily matched to a particular type of firewall is TCP port 4100. In August 2000, a denial of service (DoS) vulnerability was announced in the authentication applet of the Watchguard firebox firewalls. This vulnerability would allow a series of malformed URLs to be sent to port 4100, thus causing the DoS. The firewall would then need to be reset **(ISS)**. Watchguard quickly released a patch to correct the problem, but this had the net effect of announcing to the whole world the location of what was once a rather obscure authentication port. It is easy to point out that this information had already been freely available to the public already **(Watchguard Technologies, p. 19),** but this simply made it that much easier to find. Building on this information, it is a simple matter to add a specific line to the port list setup of a port scanner (Superscan 3.0 in this case) to recognize TCP 4100 as a Watchguard Firebox Firewall (see figure 1).

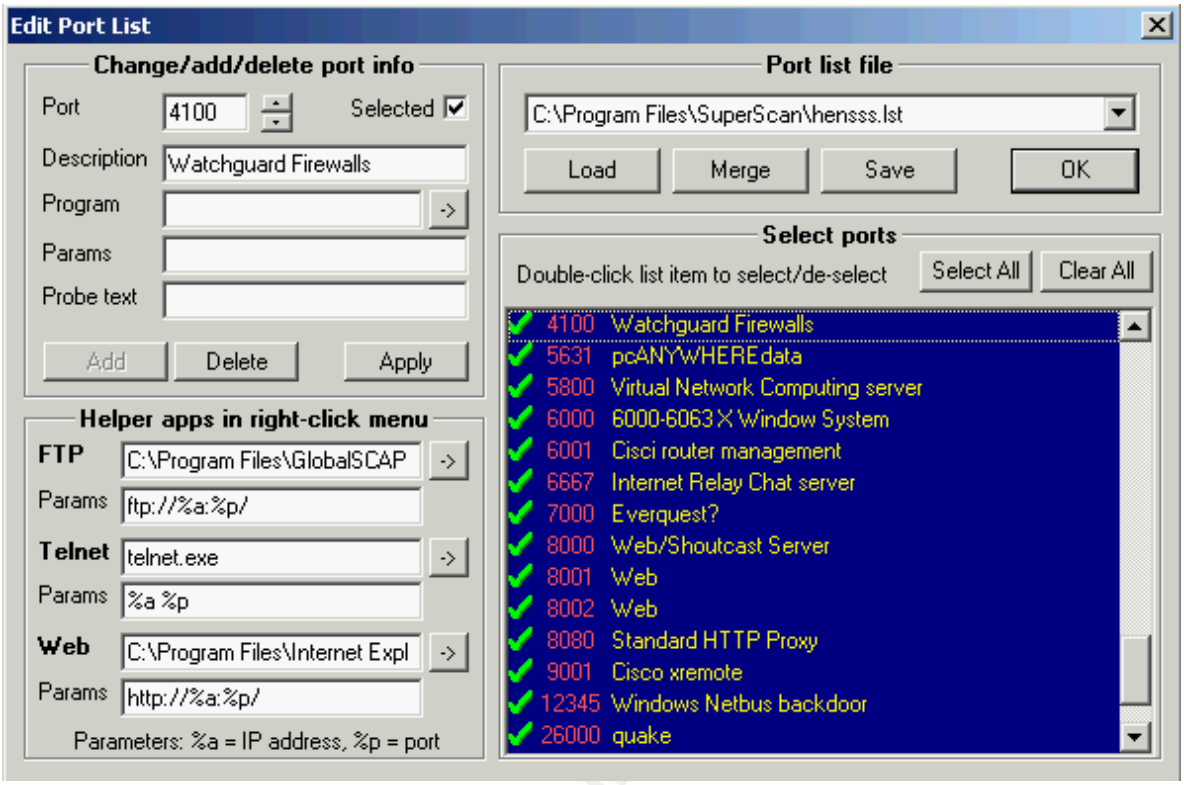

Figure 1: Adding TCP 4100 to Superscan

A review of the current IANA port assignments shows that the only other application using TCP 4100 is a secure instant messenger called Igo-incognito **(IANA)**. A search of Google also shows that Sybase uses TCP 4100, but it seems unlikely that anyone would leave their Sybase server exposed to the Internet.

Since TCP 4100 is an easily recognizable port number, the next step is to attempt to determine if a fingerprint can be established using Nmap. According to the Watchguard website, the Firebox II uses a hardened Linux 2.0 kernel as its base operating system **(Watchguard Technologies)**. For the purposes of this cases study, multiple Watchguard Firebox systems have been selected for testing at various remote locations in order to accurately simulate a typical Nmap scan. Another non-Watchguard device has been configured with TCP port 4100 open to see if it can fool Nmap. After experimenting with various types of Nmap scans, it was determined that the most useful scan was a SYN scan. In this particular instance, the use of a FIN scan (-sF) returned identical results while the use of an ACK scan, which "is generally used to map out firewall rulesets" **(Fyodor)** returned no useful information at all. TCP port 4100 was specifically scanned for since it was the only port that definitely belonged to the firewall and not to a server on the trusted side of the firewall. The results of the scans are shown below:

```
  UW PICO(tm) 4.2                                          File: 10001log

# nmap (V. 3.00) scan initiated Mon Jan 20 14:39:00 2003 as: nmap -sS -v -O -P0 -p 4100 -oN 10001log 10.0.0.1
Warning:  OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on  (10.0.0.1):
Port       State       Service
4100/tcp   open        unknown
Remote operating system guess: Linux 2.0.32-34
TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)
IPID Sequence Generation: Incremental

# Nmap run completed at Mon Jan 20 14:39:04 2003 -- 1 IP address (1 host up) scanned in 4 seconds
```

Figure 2: Initial scan from inside the perimeter of the network

Figure 2 shows the result of an Nmap scan against a Watchguard Firebox firewall from the LAN side of the network. From the information previously presented (TCP port 4100 Linux 2.0 kernel) it is quite easy to see that this is in fact a Watchguard Firebox.

```
  UW PICO(tm) 4.2                                          File:          log

# nmap (V. 3.00) scan initiated                          as: nmap -sS -v -oN          log -P0 -O -p 4100
Warning:  OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on                  net                :
Port       State       Service
4100/tcp   open        unknown
Remote operating system guess: Linux 2.0.32-34
TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)
IPID Sequence Generation: Incremental

# Nmap run completed at                                  -- 1 IP address (1 host up) scanned in 5 seconds
```

Figure 3: Ideal Scan from Outside the Perimeter of the Network

Figure 3 shows the results of an Nmap scan under what Nmap considers ideal conditions. Once again, Nmap is able to correctly identify the OS as Linux 2.0.32-34 and this can be called a Watchguard Firebox with a high degree of certainty.

```
  UW PICO(tm) 4.2                                          File:          log

# nmap (V. 3.00) scan initiated                          as: nmap -sS -v -oN          log -P0 -O -p 4100
Warning:  OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on                  .net               ):
Port       State       Service
4100/tcp   open        unknown
No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SInfo(V=3.00%P=i686-pc-linux-gnu%D=1/23%Time=3E30B018%O=4100%C=-1)
TSeq(Class=TR%IPID=I%TS=U)
T1(Resp=Y%DF=N%W=3FE0%ACK=S++%Flags=BAS%Ops=ME)
T2(Resp=N)
T3(Resp=Y%DF=N%W=3FE0%ACK=S++%Flags=ASF%Ops=ME)
T4(Resp=N)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=F%UCK=F%ULEN=134%DAT=E)


TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)
IPID Sequence Generation: Incremental

# Nmap run completed at                                  -- 1 IP address (1 host up) scanned in 17 seconds
```

Figures 4-6: Non-ideal Nmap Scan results ("partial fingerprints")

Figures four through six give us a different picture. Here Nmap was not able to give us a definitive answer as to the remote operating system. Nevertheless, the information it has provided us is interesting. Each scan taken by itself isn't particularly useful, but taken as a whole it becomes interesting in its uniformity, particularly when compared with our bogus device:

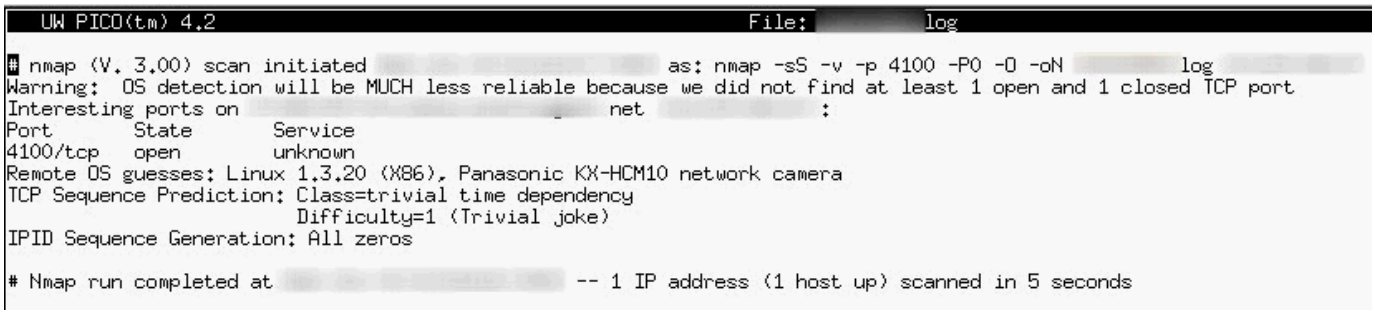

Figure 7: Non-Watchguard Device with TCP 4100 Open

The consistency of this "partial fingerprint" returned by Nmap might be used in non-ideal test conditions in lieu of complete identification by Nmap to successfully identify Watchguard Firebox firewalls on the Internet.

Case Study 2: An Unknown Device

The second case study deals with an unknown device that was identified by the method outlined above and will focus on the transcripts of a fingerprint of a firewall that occurred around August 2002 after an individual identified only by the initials "TB" penetrated and hacked a mail server at an unknown network.

The introductory text of the first message is shown below along with scan results of the unknown network device:

```
Hi list,

Recently I came against a weird network infrastructure. Can you help me
identify the types of devices used?

I have this:

[Internet]
----------------------
[An unknown device]
---------------------------------------
[Box 1-Webserver] [Box 2-Mailserver]

I was able to compromise the mailserver using web exploits.
Portscans of each devices from either the internet or the mailserver
yields
different results as shown above.

------------------
From the internet:
------------------
Unknown device:
     Open ports: 21,23,25,53,80,109,110,442
     Closed ports: 49400,54320,61439,61440,61441,65301
     Filtered: Everything else
     Replies to echo requests: No
```

```
    Nmap tcp fingerprint: FreeBSD 2.2.1 - 4.1, FreeBSD 4.1.1 - 4.3
(X86)
```
("TB")

Ignoring for a moment the bragging on a pen-test list about hacking a mail server, a closer inspection of the characteristics of the unknown device is warranted. Most of the open ports are fairly common and would be expected on a network of medium to large size (SMTP, POP3, Telnet, etc.). However, there are a couple of unusual ports there. TCP 109 is typically associated with POP2 and 442 is typically associated with cvc_hostd. Given the network diagram provided by "TB", the mystery device almost certainly must be a firewall. The next logical step is then to look for a firewall vendor that uses TCP 442 or TCP 109.

A search of Google using the keywords TCP 109 firewall yields nothing. A search of Google using the keywords TCP 442 firewall instantly brings back the ICSA labs firewall lab report on Borderware Technologies Inc. Firewall Server version 6.5 Feature Pack A **(ICSA Labs)**.

From this it is possible to make a preliminary guess that the unknown device in question is a Borderware firewall. It is still necessary to get confirmation, however. A search of the Borderware website (www.borderware.com) using the search function for the keywords "firewall" and "FreeBSD" show that, in fact, as of June 2001, the underlying firewall OS was slated to be upgraded to FreeBSD 4.x as part of the Version 6.5 package. This, along with the CPU (a Celeron 600 per the ICSA labs web page) is identical to the Nmap fingerprint.

From this information, it is possible to say with a reasonably high degree of certainty that the firewall in question is a Borderware Technologies Inc. firewall, at least Version 6.5.

The responses to "TB"'s request for information are similar to my analysis. In one post, Javier Fernández-Sanguino Peña posts the following:

```
Maybe you have a Borderware firewall there (BTW, it's pretty uncommon).
It seems to be an application-level (proxy) firewall so it fits with
some of the things you have found.  More info at
http://www.borderware.com/products/fw/fwserver.html

It seems that it runs on hardened OS (based on BSD 4.4) on Intel so it
does fit your fingerprinting. You might want to read the Security
Target, it's certified EAL4 and EAL5 so it might be a tough one :)


Bordeware provides a name server. Which adds greater confidence on my
guess :) Also, it doesn't seem to be ISC's bind. Also, due to proxying
I'd gather that the OS fingerprinting done to the webserver and
mailserver are in fact results realted to the firewall.
```

My guess (after some Google research): you have a Borderware firewall.

Regards

Javi

(Fernandez-Sanguino Pena)

One interesting (or slightly disturbing) note about this test: In "TB"s e-mail he/she specifically noted that TCP 442 was open from the Internet. In the ICSA Labs Firewall Lab Report, the following is found:

Service Port Listening	Service Identified	Expected?	Available To
TCP 442	BWClient Administrative Interface	Yes	private

Figure 8: ICSA Firewall Lab Report Table Showing TCP 442 Only Available to the LAN Interface

(ICSA Labs)

From this we can determine one of three possible things: the Borderware firewall in question is a Borderware Firewall Server 6.5 prior to feature pack A and TCP 442 could be seen on the public interface, this particular firewall was badly misconfigured, or there is a bug in the Borderware firewall software which allows TCP 442 to be seen on the public interface when it is not supposed to.

Case Study 3: A Completely Filtered Device

Some firewalls (or firewall like-devices) will be completely filtered. That is to say, the device will accept no unsolicited inbound connection attempts. Unfortunately this is the hardest type of firewall to attempt to fingerprint simply because it won't respond to attempts to talk to it. Take the following example from a local small business:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host                                                    appears to be up ...
 good.
Initiating SYN Stealth Scan against
)
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.
Warning:  OS detection will be MUCH less reliable because we did not find at lea
st 1 open and 1 closed TCP port
All 1601 scanned ports on                                                are: filtered
No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SInfo(V=3.00%P=i686-pc-linux-gnu%D=1/23%Time=3E30E58F%O=-1%C=-1)
T5(Resp=N)
T6(Resp=N)
T7(Resp=Y%DF=N%W=400%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=F%UCK=F%ULEN=134%DAT=E)


Nmap run completed -- 1 IP address (1 host up) scanned in 1942 seconds
```

Figure 9: A completely filtered perimeter device

From the information shown above it is impossible to tell what this device is. All the ports are filtered and Nmap was not able to determine anything. Further scanning is therefore going to be counterproductive. Therefore, other techniques are called for.

A useful technique in this instance is to go *behind* the firewall. Enumerating the company associated with the firewall you are targeting is the first step. Get as much information as you can: Names, addresses, numbers, everything. The objective is to narrow it down and then use social engineering or some other method to place yourself into an area that is on the company network *behind* the firewall in question. It does not have to even be in the same building, just behind the firewall. Obviously if your target firewall is in Brazil this method has some drawbacks, but for local (within the same city or state) firewalls this method can be quite effective. Other options include such tactics as war dialing to find back doors into the network that can be exploited to gain access.

Depending on your target, you may not need to do any sort of social engineering at all. Some universities, for example, tend to be extremely lax about people coming on to their campuses and plugging a laptop into their campus network. From their it's child's play to initiate a scan against the LAN side of their firewall, which is generally much more willing to be enumerated than the WAN side.

In the example shown above, the back door was even bigger: an unencrypted wireless network. Connecting to the company LAN and obtaining a DHCP IP address was a straightforward process. After that, obtaining the identity of the network device was simple, and did not even require the use of Nmap, as simple HTTP banner enumeration yielded the necessary information readily.

```
* + 192.168.1.1   [Unknown]
        |___    80  World Wide Web HTTP
                |___ HTTP/1.1 401 Authorization Required..WWW-Authenticate:
Basic realm="Linksys BEFSR41/BEFSR11/BEFSRU31"..Content-type: text/html.
```

Figure 10 HTTP Banner Enumeration Using Foundstone's Superscan 3.0

Defenses

The ideal defense to this technique is to filter the port or ports that could uniquely identify the firewall. The problem with this is that these are also the ports that allow such functions as remote management, so in many cases this may not be practical. Similarly, another defense is to change the port number. Unfortunately in many cases the vendors did not allow this option. It is ironic that in some of

the lower end products that you change the default remote management port number but that in some of the higher end firewalls you cannot.

Instead of taking such drastic actions, it might be advisable to block known ports at further out perimeter devices, such as border routers. This would provide a measure of security from scanning from the Internet while still allowing local access to the basic functions of the firewall from the LAN side for administrative purposes.

This is not an ideal solution, however, because it does not address the problem of LAN side scanning for known firewall ports. Once again, if possible, changing the default management ports is definitely the best solution to defeat this problem. If, on the other hand, changing the default port is not possible, another solution would be to setup a network intrusion detection system (NIDS) ruleset to detect scans for particular firewall ports (e.g. TCP 442 if one has a Borderware firewall). Since vendors such as Watchguard and Borderware tend to use uncommon ports this, when combined with filtering at the perimeter, would allow for a method of fingerprint scan detection for scans originating from the LAN.

Conclusion

While not a method of penetration itself, firewall fingerprinting nevertheless represents a valid method of network enumeration and, as such, needs to be defended against. If an attacker can successfully fingerprint a network's firewall, he or she can then determine if there are any known exploits against the firewall (such as the afore mentioned DOS attack against the Watchguard Firebox). This is especially critical if the network administrator is not vigilant with patches and upgrades. Once the attacker has fingerprinted the firewall, the attacker can search for information about past and current vulnerabilities about the firewall and the underlying operating system and then attempt to attack the firewall based on the information based on the information gained from the fingerprint.

The three cases studies presented in this paper showed different ways that firewalls or firewall-like devices can be enumerated. In many cases, Fyodor's Nmap is an invaluable tool for enumerating a firewall, as most firewall tend to be based on a standard or hardened OS of an identifiable type, which Nmap can identify. This, when used in combination with in combination with a known or a combination of known ports that are associated with a particular firewall can lead to a firewall fingerprint.

It is also sometimes possible to fingerprint a firewall with a "partial fingerprint". When Nmap scan conditions are non-ideal, but you have a number of known firewall types that are generating a number of identical "partial fingerprints" in combination with a known or a combination of known ports, this may also be used to generate a firewall fingerprint.

Finally, in the event that a firewall is completely filtered, Nmap will be of little help. In this case, it often proves more helpful to get behind the firewall through some means (social engineering, war driving, etc.) and scan the firewall from the LAN side of the network.

There are several possible defenses against this technique, including blocking the port at the firewall, changing the default port numbers, blocking the port at a perimeter device, and/or setting up a NIDS ruleset. The judicious use of the defenses will help minimize the risks associated with this network enumeration method.

References

1. Webopedia. "firewall." 24 September, 2002.
URL: http://www.webopedia.com/TERM/f/firewall.html (14 Feb, 2003).

2. Fyodor. "Remote OS Detection via TCP/IP Stack FingerPrinting." 11 June, 2002. URL: http://www.insecure.org/nmap/nmap-fingerprinting-article.html (14 Feb 2003).

3. Internet Security Systems. "Firebox II malformed URL to port 4100 denial of service." (15 August, 2000).
URL: http://www.iss.net/security_center/static/5098.php (14 Feb, 2003).

4. Watchguard Technologies Inc. "Watchguard Firebox System Reference Guide Firebox System 4.6": Watchguard Technologies Inc., 2001. 19.

5. IANA. "Port Numbers." (14 Feb, 2003)
URL: http://www.iana.org/assignments/port-numbers (14 Feb, 2003).

6. Watchguard Technologies Inc. "Firebox II." (2003).
URL: http://www.watchguard.com/products/fireboxii.asp (14 Feb, 2003)

7. Fyodor. "Nmap network security scanner man page"
URL: http://www.insecure.org/nmap/data/nmap_manpage.html (14 Feb, 2003)

8."TB". "Device Fingerprinting." (18 Aug, 2002)
URL: http://cert.uni-stuttgart.de/archive/pen-test/2002/08/msg00045.html (14 Feb, 2003)

9. ICSA Labs. "BorderWare Technologies Inc. BorderWare Firewall Server version 6.5 Feature Pack A." (July 24. 2002).
URL:http://www.icsalabs.com/html/communities/firewalls/certification/rxvendors/borderwarefirewallserver/labreport_cid274.shtml (14 Feb, 2003)

10. Fernandez-Sanguino Pena, Javier. "Looks like a Borderware firewall (was Re: Device fingerprinting)." (21 Aug, 2002).
URL: http://cert.uni-stuttgart.de/archive/pen-test/2002/08/msg00051.html (14 Feb, 2003)

Last Updated: July 25th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event