# Pentest+3

**Networks, Topologies, and Network Traffic** At the beginning of a black box penetration test, you will know very little about the net- works, their layout and design, and what traffic they may carry. As you learn more about the target's network or networks, you can start to lay out a network topology or logical design. Knowing how a network is laid out and what subnets, network devices, and secu- rity zones exist on the network can be crucial to the success of a penetration test.

Network Topology Understanding the topology, or layout, of a network helps a penetration tester design their scanning and attack process.

A topology map can provide information about what systems and devices are likely to be accessible, thus helping you make decisions about when to pivot to a different target to bypass security controls. **Topology diagrams** can be generated using tools like the **Zenmap GUI for Nmap** as well as purpose-built network topology mapping programs. While a Zenmap topology diagram as shown in Figure 3.13 isn't always com- pletely accurate, it can be very helpful when you are trying to picture a network. Using scanning data to create a topological diagram has a number of limi- tations. Since you are using the time-to-live information and response to scans to determine what the network looks like, firewalls and other net- work devices can mean that your topology will not match reality. Always remember that an **Nmap scan will only show you the hosts that respond and that other hosts and networks may exist! Eavesdropping and Packet Capture In addition to actively scanning for hosts and gathering topology information, penetra- tion testers will also gather information using eavesdropping with packet capture or sniffer tools.** Tools like Wireshark are often used to passively gather information about a network, including IP addresses, MAC addresses, time to live for packets, and even data about ser- vices and the content of traffic when it is unencrypted

Capturing network traffic from wireless networks can be done with Wireshark, but dedicated wireless capture tools like Kismet are also popular.

**Kismet** provides additional features that can be useful when sniffing wireless networks, including the ability to find hidden SSIDs, passive association of wireless clients and access points, and a variety of tools that help to decrypt encrypted traffic. *It is worth noting that some organizations use non-WiFi wireless networks, including Bluetooth communications, proprietary protocols, and other communication methods based on RF (radio frequency).* A**s you might imagine, Bluetooth is the most common non-WiFi wireless implementation that most penetration testers encounter, and its short range can make it challenging to intercept without getting close to your target. Fortunately, Bluetooth is often relatively insecure, making information gathering easier if you can get within range or gain access to a system that can provide that access. If your client or target uses a communication method outside of those typically in scope for a penetration test, like Ethernet and WiFi networks, you will need to make sure you have the right tools, software, and**

**knowledge to capture and interpret that traffic, and that traffic is either in or out of scope as appropriate. SNMP Sweeps Another method for gathering information about network devices is to conduct an SNMP sweep.**

*This usually requires internal access to a network and thus may not be in the first round of your active reconnaissance activities, but it can be very valuable once you have penetrated the exterior defenses of an organization.A***ctive Reconnaissance and Enumeration**

83 **Conducting an SNMP sweep** in most networks requires you to acquire the commu- nity string used by the network devices, and a lack of a response from a system does not mean there isn't a system at that IP address. In fact, there are four possible reasons a lack of response may occur: you may have the wrong community string, the system may be unreachable (fi rewalled or offl ine), the SNMP server service may not be running, or the fact that SNMP uses UDP is working against you and the response wasn't received yet—and may never be received at all! None of this means that you shouldn't attempt an SNMP scan of a network to gather information. It simply means that you may need more preparation before using a scanningtool. Once you have the information you need, SNMP scans can greatly help improve your network topology map and device discovery.

**The HighOn.Coffee Penetration Testing Tools Cheat Sheet** is a great resource for specific commands, sorted by the penetration testing phase and type of enumeration or other effort. You can find it at https:// highon.coffee/blog/penetration-testing-tools-cheat-sheet/. Specific cheat sheets for other tools and techniques like nbtscan, reverse shells, and others are also available on the same site.

If you'd like a book to work from, the Red Team Field Manual (or RTFM ) by Ben Clark is a wonderful resource.

Packet Crafting and Inspection In addition to packet capture and network scanning, penetration testers sometimes need to interact with packets and traffic directly to gather the information that they need. Manual or tool-assisted packet creation can allow you to send packets that otherwise wouldn't exist or to modify legitimate packets with your own payloads. **There are four typical tasks that packet crafting and inspection may involve:**

■✓**Packet review and decoding**

■✓Assembling packets from scratch

■✓Editing existing packets to modify their content

■✓Replaying packets While Wireshark is very useful for packet analysis, penetration testers often use other tools for packet crafting. Hping is popular because it allows you to create custom packets easily. For example, sending SYN packets to a remote system using hping can be done using the following command:**hping -S -V targetsite.com -p 8080 In this example, hping would send SYN packets to targetsite.com on TCP port 8080 and provide verbose output.** While you may not always know the flags that a command uses, many flags can be guessed—a handy trick to remember for the exam! In

addition to hping, other popular tools include Scapy, Yersina, and even NETCAT, but most penetration testers are likely to start with hping for day to day use

. Many penetration testers capture most if not all of the traffic that they generate during their penetration testing efforts. If something goes wrong, the logged traffic can be used to document what occurred and when.

Packet captures can also be useful if you think you missed some- thing or cannot get a response to reoccur. Enumeration Building the list of potential targets for a penetration test can be a massive task. If the scope and rules of engagement allow you to, you may enumerate network devices, systems,users, groups, shares, applications, and many other possible targets. Over the next few pages, we will look at some **common methods of enumerating each of these targets.** As you review each target type, bear in mind that there are both **technical and social engineering methods** to obtain this data and that the technical methods we discuss here are not the only possible methods you may encounter. Users In the past, you could often enumerate users from Linux systems via services like fi nger and rwho. Now, user enumeration requires more work. The most common means of enumerat- ing users through exposed services are **SMB and SNMP user enumeration,** but once you gain access to systems, you can also directly enumerate users from u**ser fi les, directories, and sometimes via directory services.** In many organizations, **user accounts are the same as email accounts,** making email user enumeration a very important technique. **Email Gathering valid email addresses commonly occurs prior to a phishing campaign or other penetration testing activity. In addition to more manual options, theHarvester is a program designed to gather emails, employee names, subdomains, and host information, as well as open ports and banners from search engines (including Shodan) and other sources. As you might expect, Metasploit also includes similar functionality. A search using Metasploit's email harvesting tool of the Wiley.com domain (our publisher) using Google and limited to 500 results returned 11 email addresses, 14 hostnames that were found in the search engine, and an empty result set for Shodan. Doing the same work manually would be quite slow, so using tools like Metasploit and theHarvester can be a useful way to quickly develop an initial list of targets. Remember that this type of scan is a passive scan from the target's per- spective. We're using a search engine, and these addresses are publicly exposed via that search engine. That means you can select a company that you are familiar with to practice search engine–based harvesting against. Just don't use active techniques against an organizati on without permission.**

## Harvesting emails using Metasploit

```
msf > use auxiliary/gather/search_email_collector
msf auxiliary(search_email_collector) > set domain wiley.com
domain => wiley.com
msf auxiliary(search_email_collector) > set outfile wiley-list.txt
outfile => wiley-list.txt
msf auxiliary(search_email_collector) > exploit

[*] Harvesting emails .....
[*] Searching Google for email addresses from wiley.com
[*] Extracting emails from Google search results...
[*] Searching Bing email addresses from wiley.com
[*] Extracting emails from Bing search results...
[*] Searching Yahoo for email addresses from wiley.com
[*] Extracting emails from Yahoo search results...
[*] Located 5 email addresses for wiley.com
[*]     amcslatin@wiley.com
[*]     fmcdermo@wiley.com
[*]     permissions@wiley.com
[*]     societypublishing@wiley.com
[*]     swheat@wiley.com
[*] Writing email address list to wiley-list.txt...
[*] Auxiliary module execution completed_
```

**Metasploit also includes a harvesting engine,** shown in Figure 3.14. We will dive into Metasploit usage more in future chapters, but for now, you should know that the /auxiliary/ gather/search_email_collector tool also provides an easy-to-use email address gather- ing tool. Penetration testers may also purchase commercial email address lists, search through lists of emails from compromised website account lists, or use any of a multitude of other sources for email addresses. Social Networking Sites Social media enumeration focuses on identifying all of an individual's or organization's social media accounts. These are sometimes targeted in the exploit phase for password attacks, social engineering attacks, or attempts to leverage password resets or other com- promised accounts to gain access. **Groups Groups c**ome in many forms, from Active Directory groups in an AD domain to group management tools built into identity management suites. Groups also exist in applications and service management interfaces. As a penetration tester, you need to understand both which groups exist and what rights, roles, or permissions they may be associated with. Penetration testers often target group management **interfaces and tools because add- ing an un-privileged user to a privileged group can provide an easy way to gain additional privileges without having the user directly monitored. If your target supports SNMP,** and you have the appropriate community string, you can use **snmpwalk to enumerate users as shown below using public as the community string and 10.0.0.1 as the target host. The grep and cut commands that the snmpwalk output is piped into will provide the user with information from the overall snmpwalk output. snmpwalk public -v1 10.0.0.1 1 | grep 77.1.2.25 | cut -d "" -f**

**snmpwalk public -v1 10.0.0.1 1 | grep 77.1.2.25 | cut -d "" -f4**

amba users can also be gathered using a tool like samrdump (https://github.com/CoreSecurity/impacket/blob/impacket_0_9_15/examples/samrdump.py), which com- municates with the Security Account Manager Remote interface to list user accounts and shares.

Core Security's Impacket Python libraries provide quite a few useful tools for penetration testers, including SMB tools, NTLM and Kerberos authen- tication capabilities, and a host of other useful tools. You can find a listing with descriptions at https://www.coresecurity.com/corelabs-research/ open-source-tools/impacket.

**Relationships**

Understanding how users relate to each other can be very useful when attempting to understand an organization. Fortunately, tools like the MIT Media Lab's Immersion too

(https://immersion.media.mit.edu/) can help you figure out which users connect fre- quently with others. Other relationship visualization tools are starting to become widely available, making big data techniques approachable for penetration testers.

**Shares**

Enumerating Samba (SMB) shares seeks to fi nd all available shares, which are readable and writeable, and any additional information about the shares that can be gathered. SMB scanners are built into a variety of vulnerability scanning tools, and there are also purpose-built SMB scanners like SMBMap. Nmap includes the smb-enum-shares and smb-enum-users NSE scripts as well.

**Web Pages and Servers**

Web pages and servers can be crawled and enumerated using a variety of tools. Dedicated web application assessment tools like w3af, Burp Suite, and many others can make this easier once you have identified web servers.

Many devices provide embedded web interfaces, so you may fi nd a multitude of web servers during an active scan of a larger organization. One of the fi rst tasks a penetration tester must perform is to narrow down the list of targets to a set of useful initial targets. To do this, it helps to understand the applications and sites that the servers may be hosting and fi ngerprint them to gain enough information to do so.

**Applications**

Enumerating all of an organization's applications can be challenging, particularly in a secure environment. Often, penetration testers can only connect to public applications in the early phases of a penetration test and then must continually reassess what applications and services may be accessible to them as they penetrate deeper into the organization. This

**Fingerprinting**

Application assessments rely on knowing information about the applications, such as the name, version number, underlying web server and application stack, host operating system, and any other details that can be gathered. This information is sometimes known as a fi n-gerprint. Fingerprinting applications typically starts with banner grabbing. Fortunately, NETCAT is up to the task. In Figure 3.15, we connect to a remote host using NETCAT and then issue an HTTP GET command to retrieve banner information. This tells us that the remote host is running Apache 2.2.8.

t4 banner grabbing

As you have probably already guessed, Nmap can provide the same sort of answers using the -**sV service identification flag.** In many cases, you may also want to connect with a vul-nerability scanner or web application security tool to gather more detailed information, like cookies.

The PenTest+ exam objectives mention **token enumeration**, but capturing and using tokens is typically more aligned with exploit activities. Tokens, including session tokens for privileged accounts in Windows, are often used after a service account is compromised. For a complete example of a scenario using token manipulation, you can read more at https://pentestlab.blog/tag/token-manipulation/ and as part of Metasploit's exploit capabilities at https://www.offensive-security.com/metasploit-unleashed/ fun-incognito/.

**API and Interface Enumeration**

While the PenTest+ exam objectives don't currently list APIs and other service-level inter-faces, a penetration tester should be aware that exposed APIs can be just as valuable as exposed applications. You may need API documentation to fully exploit them, but an API paired with either open access or captured API keys or other authentication and authoriza-tion tokens can provide access to all sorts of useful functions and data.88

have listed the most common OpenSSL commands and their usage.

## General OpenSSL Commands

These commands allow you to generate CSRs, Certificates, Private Keys and do other miscellaneous tasks.

- Generate a new private key and Certificate Signing Request

```
openssl req -out CSR.csr -new -newkey rsa:2048 -nodes -keyout privateKey.key
```

- Generate a self-signed certificate (see How to Create and Install an Apache Self Signed Certificate for more info)

```
openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout privateKey.key -out certific
```

- Generate a certificate signing request (CSR) for an existing private key

```
openssl req -out CSR.csr -key privateKey.key -new
```

- Generate a certificate signing request based on an existing certificate

```
openssl x509 -x509toreq -in certificate.crt -out CSR.csr -signkey privateKey.key
```

- Remove a passphrase from a private key

```
openssl rsa -in privateKey.pem -out newPrivateKey.pem
```

## Checking Using OpenSSL

If you need to check the information within a Certificate, CSR or Private Key, use these commands. You can also check CSRs and check certificates using our online tools.

- Check a Certificate Signing Request (CSR)

```
openssl req -text -noout -verify -in CSR.csr
```

- Check a private key

```
openssl rsa -in privateKey.key -check
```

- Check a certificate

```
openssl x509 -in certificate.crt -text -noout
```

- Check a PKCS#12 file (.pfx or .p12)

```
openssl pkcs12 -info -in keyStore.p12
```

## Debugging Using OpenSSL

If you are receiving an error that the private doesn't match the certificate or that a certificate that you installed to a site is not trusted, try one of these commands. If you are trying to verify that an SSL certificate is installed correctly, be sure to check out the SSL Checker.

# Debugging Using OpenSSL

If you are receiving an error that the private doesn't match the certificate or that a certificate that you installed to a site is not trusted, try one of these commands. If you are trying to verify that an SSL certificate is installed correctly, be sure to check out the SSL Checker.

- **Check an MD5 hash of the public key to ensure that it matches with what is in a CSR or private key**

```
openssl x509 -noout -modulus -in certificate.crt | openssl md5
openssl rsa -noout -modulus -in privateKey.key | openssl md5
openssl req -noout -modulus -in CSR.csr | openssl md5
```

- **Check an SSL connection. All the certificates (including Intermediates) should be displayed**

```
openssl s_client -connect www.paypal.com:443
```

# Converting Using OpenSSL

These commands allow you to convert certificates and keys to different formats to make them compatible with specific types of servers or software. For example, you can convert a normal PEM file that would work with Apache to a PFX (PKCS#12) file and use it with Tomcat or IIS. Use our SSL Converter to convert certificates without messing with OpenSSL.

- **Convert a DER file (.crt .cer .der) to PEM**

```
openssl x509 -inform der -in certificate.cer -out certificate.pem
```

- **Convert a PEM file to DER**

```
openssl x509 -outform der -in certificate.pem -out certificate.der
```

- **Convert a PKCS#12 file (.pfx .p12) containing a private key and certificates to PEM**

```
openssl pkcs12 -in keyStore.pfx -out keyStore.pem -nodes
```

You can add -nocerts to only output the private key or add -nokeys to only output the certificates.

- **Convert a PEM certificate file and a private key to PKCS#12 (.pfx .p12)**

```
openssl pkcs12 -export -out certificate.pfx -inkey privateKey.key -in certificate.crt -certfile
```



Originally posted on Sun Jan 13, 2008

# Comments

**Robert** (2014-12-13)

_

**Certificate Enumeration and Inspection**

**The certificates that an organization's websites present can be enumerated as part of an information-gathering effort**. Nmap can gather certificate information using the ssl-cert NSE script, *and all major vulnerability scanners have the ability to grab and validate certificate information*. **As you might expect, web application vulnerability scanners also specifically build in this capability. Knowing what certificates are in use, and if they are expired or otherwise problematic, can be useful to a penetration tester because out-of-date certificates often point to other administrative or support issues that may be exploited. Certificates are also used for users and services and may be acquired during later stages of a penetration test. User and service certificates and keys are typically tracked as they are acquired rather than directly enumerated.**

Information Gathering and Code

The source code, scripts, and even compiled code that underlie an organization  systems, services, and infrastructure are also very useful targets for a penetration tester. Analyzing code as part of an enumeration and information-gathering exercise can sometimes be forgotten because it requires a different skill set than port scanning and other active information gathering.

As a penetration tester, you should remain aware that code often contains useful information about targets, ranging from usernames and passwords embedded in scripts to details of how applications connect and how data is organized in databases in web application calls.

**Scripts and Interpreted Code**

The most accessible information in code is often found in scripts and other interpreted code (that is, code that is run directly instead of compiled). Most scripts and interpreted code may not be accessible during the initial active reconnaissance of an organization, but once you have bypassed outer layers of security, you are likely to recover code that you will need to analyze.

You can review code like this in Chapter 11, where we discuss scripting for penetration testing.

**Decompilation**

Compiled code, such as that found in many program binaries, requires another step before you can review it. That means you'll need a decompiler, which will pull apart the compiled code and provide readable source code. Decompilers exist for many common programming languages, so you will need to identify your specific need before matching it with an appropriate tool.Information Gathering and Defenses

A shortcut that can provide some useful information without decompiling is to use the **Linux strings utility,** which recovers text strings from compiled code. Strings is often useful during **malware analysis** once malware has been decoded from various packing methods that attempt to obfuscate the code, but for most common compiled binaries, you can simply run strings against the fi le to gather information. Figure 3.16 shows part of the strings output for NETCAT. If you'd like to try the same command, you can fi nd nc in /bin/nc on Kali Linux.

f I G u r e 3 .1 6

Excerpt of strings run on the NETCAT binary

**Debugging**

If you have the source code for a program, you can also use a debugger to review it. As with decompilation, you are unlikely to tackle much work with a debugger in the early phases, but the PenTest+ exam outline includes it in information-gathering techniques because analyzing source code is a common means of gathering additional information, and a debugger that can open programs and allow you to review them can be very useful. Fortunately, debuggers are built into the same tools you are likely to use for manual code review, like Eclipse, Visual Studio, and other integrated development environments (IDEs).

The PenTest+ exam objectives don't include manual code analysis in the Information Gathering and Vulnerability Identification objective, but reviewing scripts, HTML, and other examples of code is part of the overall exam objectives. Remember that you may be able to gather useful information from almost any data you gather from a target, including scripts and code.

**Information Gathering and Defenses**

Throughout this chapter we have discussed methods for gathering information about an organization through both passive and active methods. While you are gathering information, you need to remain aware of the defensive mechanisms that your target may have in place.90

**Defenses Against Active Reconnaissance**

Defenses against active reconnaissance primarily rely on network defenses, but reconnaissance cannot be completely stopped if any services are provided to the outside world. Active reconnaissance prevention typically relies on a few common defenses:

■■

■■

■■

Limiting external exposure of services to those that absolutely must be exposed
Using an IPS or similar defensive technology that can limit or stop probes to prevent scanning
Using monitoring and alerting systems to alarm on events that continue despite these preventative measures

Most organizations will prioritize detecting active reconnaissance on their internal networks, and organizations with a strong security policy prohibit and monitor the use of scanning tools. Active defenses may block or disconnect systems or network ports that conduct active reconnaissance activities, so monitoring your own efforts for signs of detection is critical.

**Preventing Passive Information Gathering**

Organizations have a much harder time preventing passive information gathering, as it

relies on controlling the information that they release. *Each passive information-gathering technique we reviewed has its own set of controls that can be applied. For example, DNS anti-harvesting techniques used by domain registrars can help prevent misuse. Other DNS protection techniques include these:*

■■Blacklisting systems or networks that abuse the service

■■**Using CAPTCHAs to prevent bots.**

■Providing privacy services that use third-party registration information instead of the actual person or organization registering the domain.

Implementing rate limiting to ensure that lookups are not done at high speeds.

Not publishing zone files if possible, but gTLDs are required to publish their zone files, meaning this only works for some ccTLDs.

**Other types of passive information gathering require a thorough review of exposed data and organization decisions about what should (or must) be exposed and what can be limited by either technical or administrative means.**

Gathering information about an organization is critical to penetration tests. Testers will typically be required to identify domains, hosts, users, services, and a multitude of other *elements to successfully provide complete black and gray box tests.*

*Cybersecurity teams have a wide variety of tools at their disposal to identify vulnerabilities in operating systems, plat- forms, and applications. Automated vulnerability scanners are capable of rapidly scanning systems and entire networks in an effort to seek out and detect previously unidentified vulnerabilities using a series of tests. Vulnerability management programs seek to identify, prioritize, and remediate these vulnerabilities before an attacker exploits them to undermine the confidentiality, integrity, or availability of enterprise information assets. Effective vulnerability management pro- grams use an organized approach to scanning enterprise assets for vulnerabilities, using a defined workflow to remediate those vulnerabilities and performing continuous assess- ment to provide technologists and managers with insight into the current state of enterprise cybersecurity. Penetration testers (and hackers!) leverage these same tools to develop a sense of an organization's security posture and identify potential targets for more in-depth probing and exploitation.*

**Real World Scenario**

### Developing a Vulnerability Scanning Plan

Let's revisit the penetration test of MCDS, LLC that you began in Chapter 3. When we left off, you conducted an Nmap scan to determine the active hosts and services on the network ranges used by MCDS.

As you read through this chapter, develop a plan for using vulnerability scanning to continue the information gathering that you already began. Answer the following questions:

- How would you scope a vulnerability scan for the MCDS networks?
- What limitations would you impose on the scan? Would you limit the scan to services that you suspect are running on MCDS hosts from your Nmap results or would you conduct full scans?
- Will you attempt to run your scans in a stealthy manner to avoid detection by the MCDS cybersecurity team?

## Identifying Vulnerability Management Requirements

By their nature, the vulnerability scanning tools used by enterprise cybersecurity teams for continuous monitoring and those used by penetration testers have significant overlap. In many cases, penetration testers leverage the same instances of those tools to achieve both time savings and cost reduction. If an enterprise has a robust vulnerability management program, that program can serve as a valuable information source for penetration testers. Therefore, we'll begin this chapter by exploring the process of creating a vulnerability management program for an enterprise and then expand into the specific uses of these tools for penetration testing.

As an organization begins developing a vulnerability management program, it should first undertake the identification of any internal or external requirements for vulnerability scanning. These requirements may come from the regulatory environment(s) in which the organization operates or they may come from internal policy-driven requirements.

### Regulatory Environment

Many organizations find themselves bound by laws and regulations that govern the ways they store, process, and transmit information. **This is especially true when the organization handles sensitive personal information or information belonging to government agencies.** Many of these laws are not overly prescriptive and do not specifically address the implementation of a vulnerability management program. For example, the Health Insurance Portability and Accountability Act (HIPAA) regulates the ways that healthcare providers, insurance companies, and their business associates handle protected health information. Similarly, the Gramm-Leach-Bliley Act (GLBA) governs how financial institutions may handle customer financial records. Neither of these laws specifically requires that covered organizations conduct vulnerability scanning.

Two regulatory schemes, however, do specifically mandate the implementation of a vulnerability management program==: the Payment Card Industry Data Security Standard==

(PCI DSS) and the Federal Information Security Management Act (FISMA).Identifying Vulnerability Management Requirements

**Payment Card Industry Data Security Standard (PCI DSS)**

PCI DSS prescribes specific security controls for merchants who handle credit card transactions and service providers who assist merchants with these transactions. This standard includes what are arguably the most specific requirements for vulnerability scanning of any standard.

*Contrary to what some believe, **PCI DSS is not a law.** The standard is maintained by an industry group known as the Payment Card Industry Security Standards Council (PCI SSC), which is funded by the industry to maintain the requirements. Organizations are subject to PCI DSS because of contractual requirements rather than legal requirements.*

*PCI DSS prescribes many of the details of vulnerability scans:*

■✓

■✓

■✓

■✓

■✓

Organizations must run scans on at least a **quarterly basis** and **"after any significant change in the network** (such as new system component installations, changes in network topology, firewall rule modifications, product upgrades)" (PCI DSS requirement 11.2).

Internal scans must be conducted by qualified personnel (PCI DSS requirement 11.2.1).

Organizations must remediate any high-risk vulnerabilities and repeat scans to confirm that they are resolved until they receive a "clean" scan report (PCI DSS requirement 11.2.1).

External scans must be conducted by an Approved Scanning Vendor (ASV) authorized by PCI SSC (PCI DSS requirement 11.2.2).

**Vulnerability scanning for PCI DSS compliance is a thriving and competitive industry,** and many security consulting fi rms specialize in these scans. Many organizations choose to conduct their own scans fi rst to assure themselves that they will achieve a passing result before requesting an official scan from an ASV.

Y**ou should never conduct vulnerability scans unless you have explicit permission to do so.** Running scans without permission can be a serious violation of an organization's security policy and may also be a crime.

Federal Information Security Management Act (FISMA)

The Federal Information Security Management Act of 2002 (FISMA) requires that government agencies and other organizations operating systems on behalf of government agencies comply with a series of security standards. **The specific controls required by these standards depend on whether the government designates the system as low impact, moderate impact, or high impact, according to the defi nitions shown in Figure 4.1. Further guidance on system classification is found in Federal Information Processing Standard (FIPS) 199: Standards for Security Categorization of Federal Information and Information Systems.104**

**FIPS 199 Standards**

**POTENTIAL IMPACT**

~Security ObjectiveLOWMODERATEHIGH

~Confidentiality

~Preserving authorized

~restrictions on information

~access and disclosure,

i'including means for

protecting personal

privacy and proprietary

information.'

[**44 U.S.C., SEC. 3542]**The unauthorized

disclosure of information

could be expected to have

a limited adverse effect on

organizational operations,

organizational assets, or

individuals.The unauthorized

disclosure of information

could be expected to have

a serious adverse effect on

organizational operations,

organizational assets, or

individuals.The unauthorized

disclosure of information

could be expected to have

a severe or catastrophic

adverse effect on

organizational operations,

organizational assets, or

individuals.

**Integrity**

Guarding against improper

information modification

or destruction, and

includes ensuring

information non-

repudiation and

authenticity.

[44 U.S.C., SEC. 3542]The unauthorized

modification or

destruction of information

could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.The unauthorized modification or destruction of information could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.The unauthorized modification or destruction of information could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.

**Availability**

Ensuring timely and reliable access to and use of information.

[44 U.S.C., SEC. 3542]The disruption of access to or use of information or an information system could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.The disruption of access to or use of information or an information system could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.The disruption of access to or use of information or an information system could be expected to have a severe or catastrophic

adverse effect on
organizational operations,
organizational assets, or
individuals.
Source: National Institute of Standards and Technology (NIST). Federal Information Processing Standards
(FIPS) PUB 199: Standards for Security Categorization of Federal Information and Information Systems. Febru-
ary 2004. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.199.pdf
In 2014, President Obama signed the Federal Information Security Modern-
ization Act (yes, also confusingly abbreviated FISMA!) into law. The 2014
FISMA updated the 2002 FISMA requirements to provide strong cyberde-
fense in a changing threat environment. M**ost people use the term FISMA to refer to the combined effect of both of these laws.**
All federal information systems, regardless of their impact categorization, must meet the
basic requirements for vulnerability scanning found in NIST Special Publication 800-53,
Security and Privacy Controls for Federal Information Systems and Organizations. EachIdentifying Vulnerability Management Requirements
105

organization subject to FISMA must meet the following requirements, described in the sec-
tion "**Control Description" (https://nvd.nist.gov/800-53/Rev4/control/RA-5):**
a.**Scans for vulnerabilities in the information system and hosted applica-
tions and when new vulnerabilities potentially affecting the system/
application are identified and reported;**
b.Employs vulnerability scanning tools and techniques that facilitate
interoperability among tools and automate parts of the vulnerability
management process by using standards for:
1.*Enumerating platforms, software flaws, and improper configurations;*
*2.Formatting checklists and test procedures; and*
*3.Measuring vulnerability impact;*
*c.Analyzes vulnerability scan reports and results from security control assessments;*
d.*Remediates legitimate vulnerabilities in accordance with an organiza-
tional assessment of risk; and*
e.*Shares information obtained from the vulnerability scanning process and security control assessments to help eliminate similar vulnerabilities in other information systems* (i.e. systemic weaknesses or deficiencies)
These requirements establish a baseline for all federal information systems. NIST
800-53 then describes eight control enhancements that may be required depending on the
circumstances:
1.The organization employs vulnerability scanning tools that include the capa-
bility to readily update the information system vulnerabilities to be scanned.

2.The organization updates the information system vulnerabilities scanned prior to a new scan (and/or) when new vulnerabilities are identified and reported.
3.The organization employs vulnerability scanning procedures that can identify the breadth and depth of coverage (i.e., information system components scanned and vulnerabilities checked).
4.The organization determines what information about the information system is discoverable by adversaries and subsequently takes organization-defined corrective actions.
5.The information system implements privileged access authorization to information system components for selected vulnerability scanning activities.
6.The organization employs automated mechanisms to compare the results of vulnerability scans over time to determine trends in information system vulnerabilities.
7.(Withdrawn by NIST)

8.The organization reviews historic audit logs to determine if a vulnerability identified in the information system has been previously exploited.
9.(Withdrawn by NIST)
10. The organization correlates the output from vulnerability scanning tools to determine the presence of multi-vulnerability/multi-hop attack vectors
Note that requirements 7 and 9 were control enhancements that were previously included in the standard but were later withdrawn.
In cases where a federal agency determines that an information system falls into the moderate impact category, it must implement control enhancements 1, 2, and 5, at a minimum. If the agency determines a system has high impact, it must implement at least control enhancements 1, 2, 4, and 5.

**Corporate Policy**

**The prescriptive security requirements of PCI DSS and FISMA cover organizations involved in processing retail transactions and operating government systems, but those two categories constitute only a fraction of all enterprises. Cybersecurity professionals widely agree that vulnerability management is a critical component of any information security program, and for this reason, many organizations mandate vulnerability scanning in corporate policy, even if that is not a regulatory requirement.**

**Support for Penetration Testing**

While penetration testers often draw upon the vulnerability scans that organizations conduct for other purposes, they may also have specialized scanning requirements in support of specific penetration testing efforts.
If a penetration testing team plans to conduct a test of a specific network or environment, they may conduct an in-depth scan of that environment as one of the first steps in their information-gathering phase. Similarly, if the team plans to target a specific service, they may design and execute scans that focus on that service. For example, an organization might decide

to conduct a penetration test focused on a newly deployed Internet of Things (IoT) environment. In that case, the penetration testers may conduct vulnerability scans that focus on networks containing those devices and using tests that are focused on known IoT vulnerabilities.

**Identifying Scan Targets**

Once an organization decides to conduct vulnerability scanning and determines which, if any, regulatory requirements apply to its scans, it moves on to the more detailed phases of the planning process. The next step is to identify the systems that will be covered by the vulnerability scans. Some organizations choose to cover all systems in their scanning process, whereas others scan systems differently (or not at all) depending on the answers to questions such as these:

■■

■■

**What is the data classification of the information stored, processed, or transmitted by the system?**

Is the system exposed to the Internet or other public or semipublic networks?Identifying Vulnerability Management Requirements

■■What services are offered by the system?

■■Is the system a production, test, or development system?

107

Organizations also use automated techniques to identify the systems that may be covered by a scan. Cybersecurity professionals use scanning tools to conduct discovery scans that search the network for connected systems, whether they were previously known or unknown, and build an asset inventory. Figure 4.2 shows an example of an asset map developed using the QualysGuard asset inventory functionality.

QUALYS GUARD ASSET MAPS?>

**Many factors influence how often an organization decides to conduct vulnerability scans against its systems:**

■■

■■

■■

■■

■■

■■

The organization’s risk appetite is its willingness to tolerate risk within the environment. If an organization is extremely risk averse, it may choose to conduct scans more frequently to minimize the amount of time between when a vulnerability comes into existence and when it is detected by a scan.

**Regulatory requirements, such as PCI DSS or FISMA, may dictate a minimum frequency for vulnerability scans.** These requirements may also come from corporate policies.

Technical constraints may limit the frequency of scanning. **For example, the scanning system may only be capable of performing a certain number of scans per day and organizations may need to adjust scan frequency to ensure that all scans complete successfully.**

Business constraints may prevent the organization from conducting resource-intensive vulnerability scans during periods of high business activity to

Licensing limitations may curtail the bandwidth consumed by the scanner or the number of scans that may be conducted simultaneously.

avoid disruption of critical processes.

Licensing limitations may curtail the bandwidth consumed by the scanner or the number of scans that may be conducted simultaneously.

*Operational constraints may limit the ability of the cybersecurity team to monitor and react to scan results promptly.*

Cybersecurity professionals must balance all of these considerations when planning a vulnerability scanning program. It is usually wise to begin small and slowly expand the scope and frequency of vulnerability scans over time to avoid overwhelming the scanning infrastructure or enterprise systems.

Penetration testers must understand the trade-off decisions that were made when the organization designed its existing vulnerability management program. These limitations may point to areas where penetration testers should supplement the organization's existing scans with customized scans designed specifically for the purposes of penetration testing.

## Configuring and Executing Vulnerability Scans

Whether scans are being performed by cybersecurity analysts focused on building a lasting vulnerability management program or penetration testers conducting a one-off scan as part of a test, administrations must configure vulnerability management tools to perform scans according to the requirements-based scan specifications. These tasks include identifying the appropriate scope for each scan, configuring scans to meet the organization's requirements, and maintaining the currency of the vulnerability scanning tool.110

### Scoping Vulnerability Scans

The scope of a vulnerability scan describes the extent of the scan, including answers to the following questions:

■✓

What systems, networks, services, applications, and protocols will be included in the vulnerability scan?

■✓What technical measures will be used to test whether systems are present on the network?

■✓*What tests will be performed against systems discovered by a vulnerability scan?*

*When designing vulnerability scans as part of an ongoing program, administrators should fi rst answer these questions in a general sense and ensure that they have consensus from technical staff and management that the scans are appropriate and unlikely to cause disruption to the business. Once they've determined that the scans are well designed and unlikely to cause serious issues, they may then move on to configuring the scans within the vulnerability management tool.*

When scans are taking place as part of a penetration test, penetration testers should still avoid business disruption to the extent possible. However, the invasiveness of the testing and the degree of coordination with management should be guided by the agreed-upon statement of work (SOW) for the penetration test. If the penetration testers have carte blanche to use whatever techniques are available to them without prior coordination, it is not necessary to consult with management. Testers must, however, always stay within the agreed-upon scope of their SOWs.

By this point, the fact that penetration testers must take pains to stay within the defined parameters of their SOWs should not be news to you. Keep this fact top-of-mind as you take the PenTest+ exam. If you see questions asking you whether a decision is appropriate, your first reaction should be to consult the SOW.

Scoping Compliance Scans

Scoping is an important tool in the cybersecurity toolkit because it allows analysts to reduce problems to manageable size. For example, an organization that processes credit cards may face the seemingly insurmountable task of achieving PCI DSS compliance across its entire network that consists of thousands of systems.

Through judicious use of network segmentation and other techniques, administrators may isolate the handful of systems actually involved in credit card processing, segregating them from the vast majority of systems on the organization's network. WhenConfiguring and Executing Vulnerability Scans

111

done properly, this segmentation reduces the scope of PCI DSS compliance to the much smaller isolated network that is dedicated to payment card processing.

When the organization is able to reduce the scope of the PCI DSS network, it also reduces the scope of many of the required PCI DSS controls, including vulnerability scanning. Instead of contracting with an approved scanning vendor to conduct quarterly compliance scans of the organization's entire network, they may reduce the scope of that scan to those systems that actually engage in card processing. This will dramatically reduce the cost of the scanning engagement and the remediation workload facing cybersecurity professionals after the scan completes.

Configuring Vulnerability Scans

Vulnerability management solutions provide the ability to configure many different parameters related to scans. In addition to scheduling automated scans and producing reports, administrators may customize the types of checks performed by the scanner, provide **credentials to access target servers, in**stall scanning agents on target servers, and conduct scans from a variety of network perspectives.

The examples in this chapter use the popular Nessus and QualysGuard vulnerability scanning tools. These are commercial products. Organizations looking for an open-source solution may wish to consider the OpenVAS project, available at http://www.openvas.org/.

Scan Sensitivity Levels

**Cybersecurity professionals configuring vulnerability scans should pay careful attention to the configuration settings related to the scan sensitivity level.** While it may be appropriate in some cases to conduct full scans using all available vulnerability tests, it is usually more productive to adjust the scan settings to the specific needs of the assessment or penetration test that is underway.

Scan sensitivity settings determine the types of checks that the scanner will perform and should be customized to ensure that the scan meets its objectives while minimizing the possibility of disrupting the target environment.

**Typically, administrators create a new scan by beginning with a template.** This may be a template provided by the vulnerability management vendor and built into the product, such as the Nessus templates shown in Figure 4.5, or it may be a custom-developed template created for use within the organization. As administrators create their own scan confi gurations, they should consider saving common confi guration settings in

**Administrators may also improve the efficiency of their scans by configuring the specific plug-ins that will run during each scan. Each plug-in performs a check for a specific vulnerability, and these plug-ins are often grouped into families based on the operating system, application, or device that they involve. Disabling unnecessary plug-ins improves the speed of the scan by bypassing unnecessary checks and also may reduce the number of false positive results detected by the scanner.***nessus check**

???\

For example, an organization that does not use the *Amazon Linux* operating system may choose to disable all checks related to Amazon Linux in its scanning template. Figure 4.6 shows an example of disabling these plug-ins in Nessus. Similarly, an organization that blocks the use of some protocols at the network firewall may not wish to consume time performing external scans using those protocols.Configuring and Executing Vulnerability Scans

Figure 4.6

**Disabling unused plug-ins**

Scanning Fragile Systems

Some plug-in scan tools perform tests that may actually disrupt activity on a fragile production system or, in the worst case, damage content on those systems. These plug-ins present a tricky situation. Administrators want to run the scans because they may identify problems that could be exploited by a malicious source. At the same time, cybersecurity professionals clearly don’t want to cause problems on the organization’s network!

These concerns are heightened on networks containing nontraditional computing assets, such as networks containing industrial control systems (ICSs), Internet of Things (IoT)

devices, specialized medical equipment, or other potentially fragile systems. While penetration tests should uncover deficiencies in these systems, it is not desirable to disrupt production activity with poorly configured scans if at all avoidable.
113114

One way around this problem is to maintain a test environment containing copies of the same systems running on the production network and running scans against those test systems first. If the scans detect problems in the test environment, administrators may correct the underlying causes on both test and production networks before running scans on the production network.
During penetration tests, testers may wish to configure their scans to run as stealth scans, which go to great lengths to avoid using tests that might attract attention. This is especially true if the organization's cybersecurity team is not aware that a penetration test is underway. Service disruptions, error messages, and log entries caused by scans may attract attention from the cybersecurity team that causes them to adjust defenses in a manner that obstructs the penetration test. Using stealth scans better approximates the activity of a skilled attacker, resulting in a more realistic penetration test.
Supplementing Network Scans
**Basic vulnerability scans run over a network, probing a system from a distance. T**his provides a realistic view of the system's security by simulating what an attacker might see from another network vantage point. However, the firewalls, intrusion prevention systems, and other security controls that exist on the path between the scanner and the target server may affect the scan results, providing an inaccurate view of the server's security independent of those controls.
Additionally, many security vulnerabilities are difficult to confirm using only a remote scan. V**ulnerability scans that run over the network may detect the possibility that a vulnerability exists but be unable to confirm it with confidence, causing a false positive result that requires time-consuming administrator investigation.**
**Virtualization and Container Security**
Many IT organizations embrace virtualization and container technology as a means to improve the efficiency of their resource utilization. Virtualization approaches allow administrators to run many virtual "guest" operating systems on a single physical "host" system. This allows the guests to share CPUs, memory, storage, and other resources. It also allows administrators to quickly reallocate resources as needs shift.
Containerization takes virtualization technology a step higher up in the stack. Instead of merely running on shared hardware, as is the case with virtual machines, containers run on a shared operating system but still provide the portability and dynamic allocation capabilities of virtualization.Configuring and Executing Vulnerability Scans

Administrators and penetration testers working in both virtualized and containerized environments should pay careful attention to how the interactions between components in those environments may affect the results of vulnerability scans. For example, network communications between virtual machines or containerized applications may take place

entirely within the confines of the virtualization or containerization environment using virtual networks that exist in memory on a host. Services exposed only within those environments may not be detectable by traditional network-based vulnerability scanning. Agent-based scans may work in a more effective manner in these environments. Many vulnerability management tools are also now virtualization- and containerization-aware, allowing them to process configuration and vulnerability information for components contained within these environments.

Modern vulnerability management solutions can supplement these remote scans with trusted information about server configurations. This information may be gathered in two ways. First, administrators can provide the scanner with credentials that allow the scanner to connect to the target server and retrieve configuration information. This information can then be used to determine whether a vulnerability exists, improving the scan's accuracy over noncredentialed alternatives. For example, if a vulnerability scan detects a potential issue that can be corrected by an operating system service pack, the credentialed scan can check whether the service pack is installed on the system before reporting a vulnerability.

Credentialed scans are widely used in enterprise vulnerability management programs, and it may be fair game to use them in penetration tests as well. However, this depends upon the parameters of the penetration test and whether the testing team is supposed to have "white box" access to internal information as they conduct their work. If a penetration test is intended to be a "black box" exercise, providing the team with results of credentialed vulnerability scans would normally be outside the bounds of the test. As always, if questions exist about what is or is not appropriate during a penetration test, consult the agreed-upon SOW.

Figure 4.7 shows an example of the credentialed scanning options available within QualysGuard. Credentialed scans may access operating systems, databases, and applications, among other sources.

**Configuring authenticated scanning116**

Credentialed scans typically only retrieve information from target servers and do not make changes to the server itself. Therefore, administrators should enforce the principle of least privilege by providing the scanner with a read-only account on the server. This reduces the likelihood of a security incident related to the scanner's credentialed access.

In addition to credentialed scanning, some scanners supplement the traditional server-based approach to vulnerability scanning with a complementary agent-based approach. In this approach, administrators install small software agents on each target server. These agents conduct scans of the server configuration, providing an "inside-out" vulnerability scan, and then report information back to the vulnerability management platform for analysis and reporting.

System administrators are typically wary of installing agents on the servers that they manage for fear that the agent will cause performance or stability issues. If you choose to use an agent-based approach to scanning, you should approach this concept conservatively, beginning with a small

pilot deployment that builds confidence in the agent before proceeding with a more widespread deployment.

Scan Perspective

**Comprehensive vulnerability management programs provide the ability to conduct scans from a variety of scan perspectives.** Each scan perspective conducts the scan from a different location on the network, providing a different view into vulnerabilities. Penetration testers must be keenly aware of the network topology of the environments undergoing testing and how the location of their tools on the network may affect scan results.

For example, **an external scan is run from the Internet, giving administrators a view of what an attacker located outside the organization would see as potential vulnerabilities. Internal scans might run from a scanner on the general corporate network, providing the view that a malicious insider might encounter.** Finally, scanners located inside the data center and agents located on the servers offer the most accurate view of the real state of the server by showing vulnerabilities that might be blocked by other security controls on the network.

The internal and external scans required by PCI DSS are a good example of scans performed from different perspectives. The organization may conduct its own internal scans but must supplement them with external scans conducted by an approved scanning vendor.

Vulnerability management platforms have the ability to manage different scanners and provide a consolidated view of scan results, compiling data from different sources. Figure 4.8 shows an example of how the administrator may select the scanner for a newly configured scan using QualysGuard.