Chapter 12
Vulnerability Assessment and Management

Chuck Easttom

Introduction

Awareness of vulnerabilities is a critical issue in cybersecurity. There are well-known and documented vulnerabilities that can be readily addressed, but only if one is aware of them. Vulnerabilities are an issue of what you don't know can hurt you. Fortunately, there are a number of tools and resources that can aid you in understanding vulnerabilities in your system. You can then integrate that information into your security process.

The first step is to become aware of vulnerabilities in your network. As will be seen in this chapter, there is a wide array of tools and websites to assist you in this. But then that information must be integrated into your cybersecurity plans in order to mitigate the vulnerabilities. The latter part of this chapter will address that issue. This chapter is closely related to the material you saw in Chap. 10 and there is some minimal overlap.

Tools

In this section, we will examine some widely used tools for finding vulnerabilities. Some of these are general vulnerability scanners. Others check specific vulnerability issues. We will also look at both open source and commercial tools. The criteria for a tool being mentioned in this section is simply its popularity.

I. Alsmadi et al., *The NICE Cyber Security Framework*,
https://doi.org/10.1007/978-3-030-41987-5_12

Shodan

Shodan is a well-known tool for vulnerability scanning. The website https://www.shodan.io/ is essentially a search engine for vulnerabilities. You need to sign up for a free account to use it, but there is no spam or other issued with the website. It is also popular with attackers, thus defenders should also use this site. You can also be sure that attackers use this site as well. You can see the website in Fig. 12.1.

The issue with Shodan is using the proper search terms. There are many you can use, a few will be discussed and one demonstrated here. Here are a few basic searches:

Search for default passwords with a specific filter

default password country:US
default password hostname:chuckeasttom.com
default password city:Dallas
default password state:AR

Find Apache servers

apache city:Dallas.

Find Webcams

webcamxp city:Houston
OLD IIS
"iis/6.0"

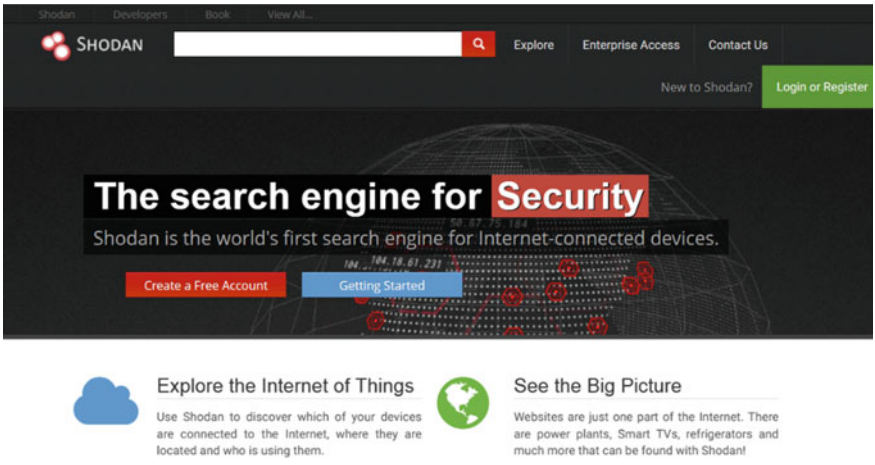

Fig. 12.1 Shodan.io

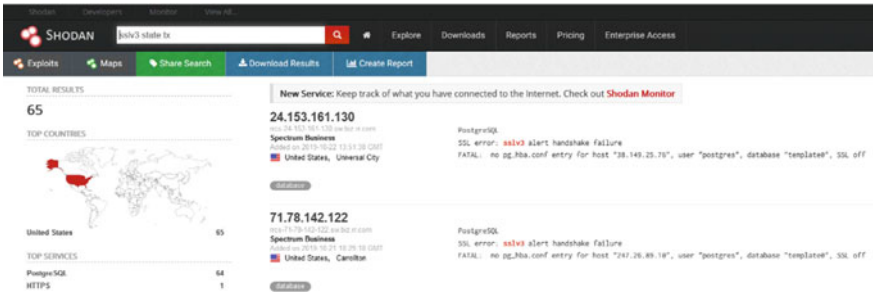

Fig. 12.2 Shodan search results

The preceding list are examples of search terms, the filters you can use include

- city: find devices in a specific city
- country: find devices in a specific country
- geo: you can pass it coordinates (i.e. latitude and longitude)
- hostname: find values that match a specific hostname
- net: search based on an IP or/x CIDR
- os: search based on operating system
- port: find particular ports that are open
- before/after: find results within a timeframe.

As an example, Fig. 12.2 shows the results for my search sslv3 state:tx

Of most use to cybersecurity professionals will be searching for vulnerabilities and filtering the results to a specific domain. That domain would be your organizations domain. Shodan is a very effective starting point for your vulnerability assessment. It will show you well-known vulnerabilities that are publicly accessible. This will be the first items you should address.

Maltego

Maltego is generally considered an open-source intelligence tool, rather than a vulnerability scanner. There are several versions of the product, some are free versions others are not. The website is https://www.paterva.com/web7/downloads.php#tab-3. The community version is free.

Results are well represented in a variety of easy to understand views In concert with its graphing libraries, Maltego identifies key relationships between data sets and identifies previously unknown relationships between them. Figure 12.3 shows the main screen of Maltego.

Maltego is often used for open-source intelligence. But it can be useful in understanding certain vulnerabilities. For example, if a suspect email is received, the website or email address can be searched with Maltego. Maltego is primarily used

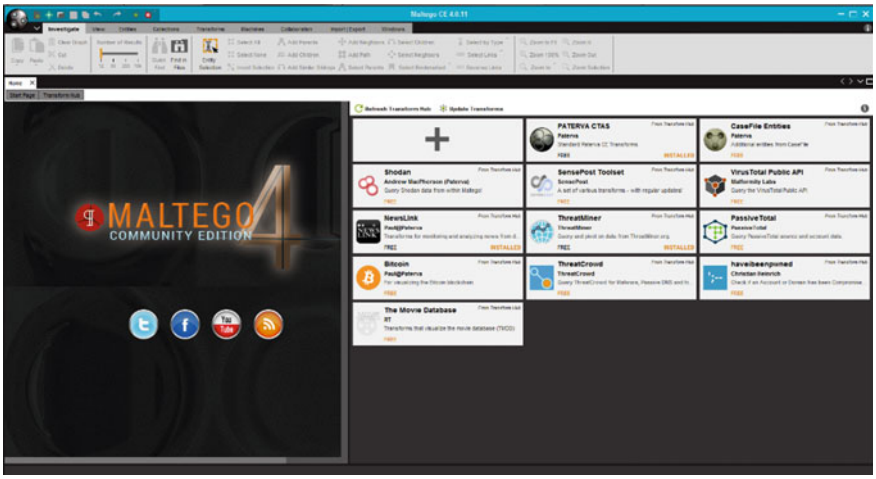

Fig. 12.3 Maltego

by working with entities and transforms. You select some entity such as an email address or website and select a transform for that entity. Once you have selected something to graph, be it a person, email address, website or other items, the relationships between that entity and other entities are shown as a graph. This can be seen in Fig. 12.4.

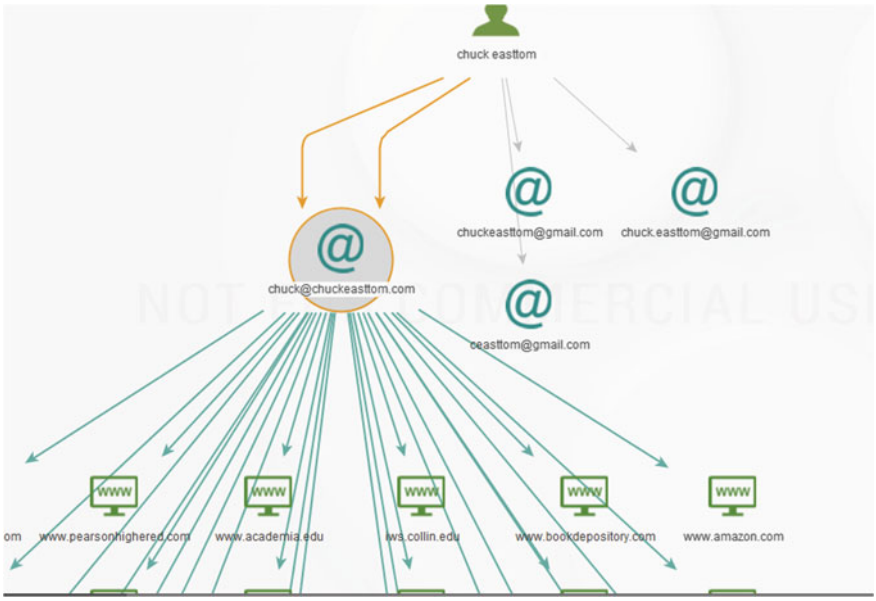

Fig. 12.4 Maltego graph

Maltego is more complex than some of the other tools we have discussed in this chapter. However, there are tutorials on the web to help you master this tool.

https://www.paterva.com/web7/docs/documentation.php
https://null-byte.wonderhowto.com/how-to/hack-like-pro-use-maltego-do-network-reconnaissance-0158464/

Spending some time learning the details of Maltego can be very worthwhile. It is not the most critical tool for vulnerability scanning but can be useful in analyzing vulnerabilities.

Nessus

Nessus (www.Nessus.org) is the most widely known commercial network vulnerability scanner. Tin the past there was a free version for personal use and a commercial version. It is now only available for a license cost. This is perhaps the most widely used vulnerability scanner available today. In this section, we will briefly explore the basic functionality. If you have an interest in learning more about Nessus, then it is recommended that you consult the documentation available at the Nessus website.

Nessus is a well-known vulnerability scanner. It has been used for many years. The license is currently over $2100 per year and can be obtained from https://www.tenable.com. Its price has been a barrier for small organizations with limited budgets. The primary advantage of Nessus is that the vendor is constantly updating the vulnerabilities it can scan for. Nessus also has a very easy-to-use web interface, as shown in Fig. 12.5.

If you select **New Scan**, you are given a number of options, as shown in Fig. 12.6.

You can select **Basic Network Scan** to see a number of intuitive basic settings. You have to name your scan and select a range of IP addresses, as shown in Fig. 12.7.

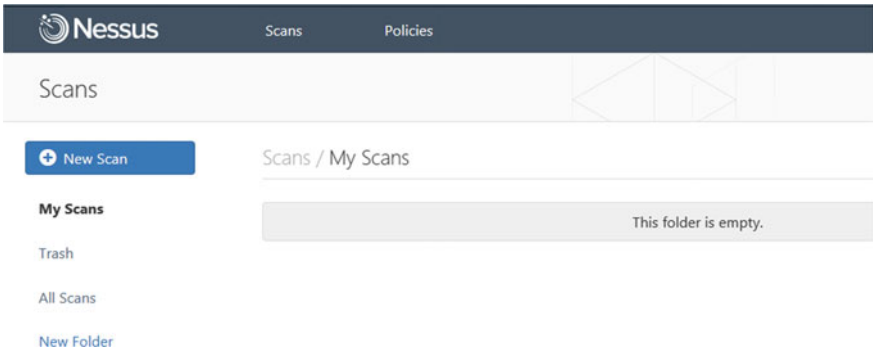

Fig. 12.5 Nessus new scan

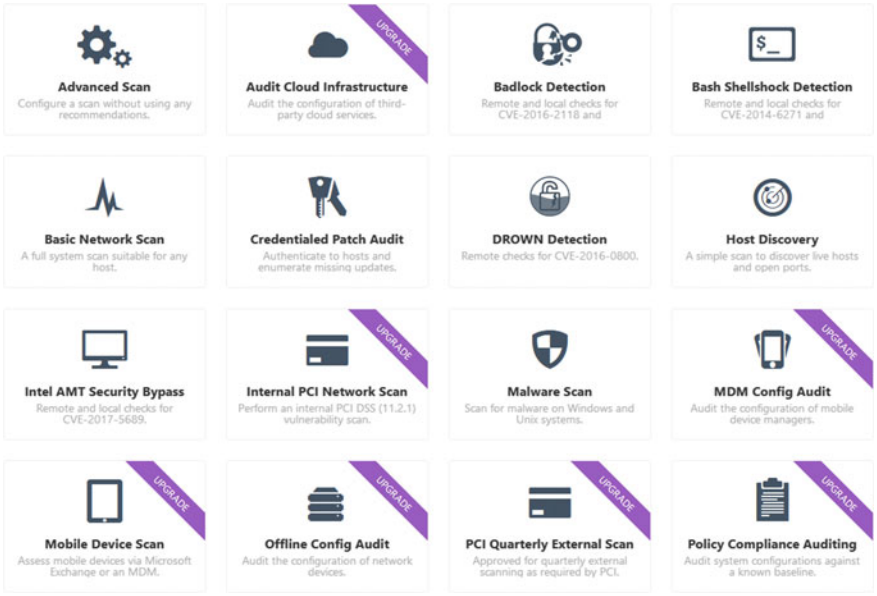

Fig. 12.6 Nessus options

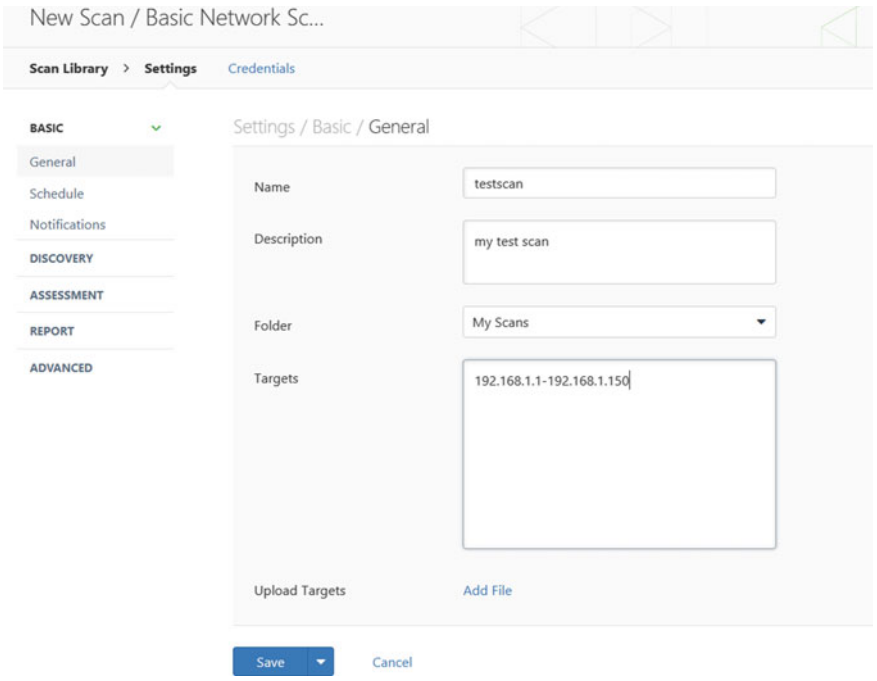

Fig. 12.7 Nessus network scan

Then you can either schedule the scan to run later or launch it right away. Nessus scans can take some time to run because they are quite thorough. The results are presented in a very organized screen that is quite intuitive.

OWASP Zap

The Open Web Application Security Project (OWASP) is the standard for web application vulnerability. OWASP offers a free vulnerability scanner called the Zed Attack Proxy, commonly known as OWASP ZAP. You can download it from https://github.com/zaproxy/zaproxy/wiki/Downloads. The interface, shown in Fig. 12.8, is very easy to use.

The results are displayed in an easy to navigate format. One can simply double click on any specific result to get more details. The results can be seen in Fig. 12.9.

OWASP ZAP is a very easy-to-use tool. The basics can be mastered in a few minutes. And given that OWASP is the organization that tracks web application vulnerabilities, it is a very good source for testing the vulnerabilities of a website.

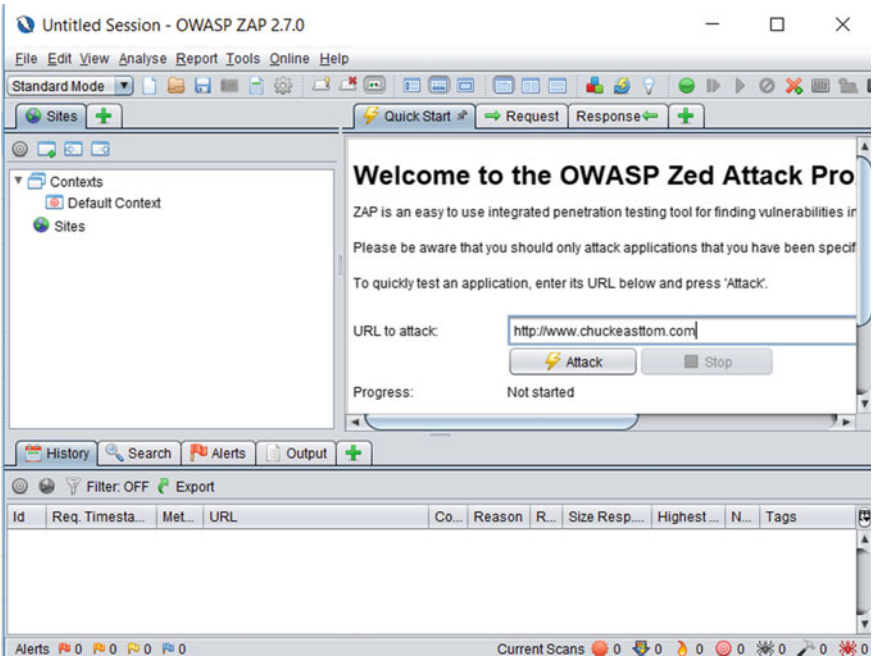

Fig. 12.8 OWASP ZAP main screen

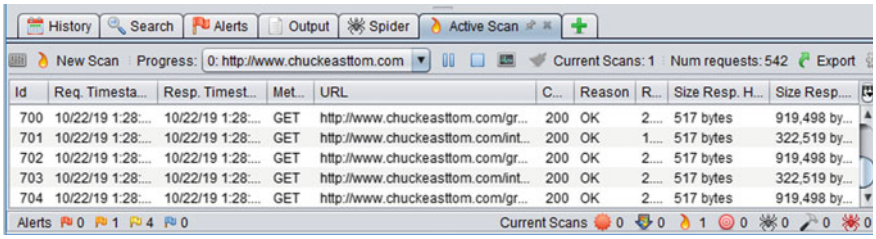

Fig. 12.9 OWASP ZAP results

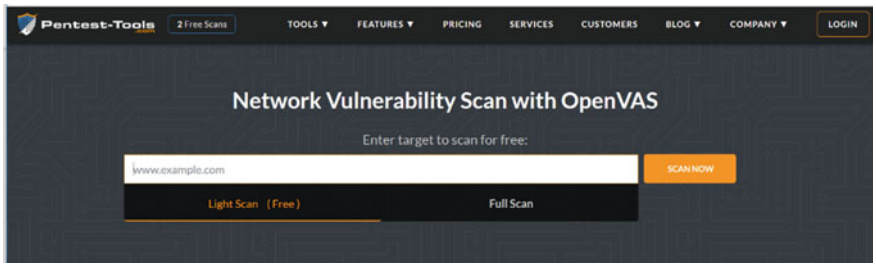

Fig. 12.10 OpenVAS online scan

OpenVAS

OpenVAS is one of the most widely used open-source vulnerability scanners. You can download OpenVAS, or you can use their online vulnerability scan. https://pentest-tools.com/network-vulnerability-scanning/network-security-scanner-online-openvas.

The website can be seen in Fig. 12.10.

The results are easy to see. Now on the result screen, you will be prompted to purchase the full version by upgrading to a pro account. However, you are also welcome to continue using the free version. This is shown in Fig. 12.11.

OpenVAS is an effective and easy-to-use tool. Even if you have other tools you are accustomed to, it would be advisable to include OpenVAS in your suite of vulnerability scanners.

Testing Specific Issues

In addition to general-purpose vulnerability scanners, there are websites that will scan for specific items. For example, the following site will provide you a report on the SSL/TLS security of any public website.

SSLLabs https://www.ssllabs.com/ssltest/analyze.html.

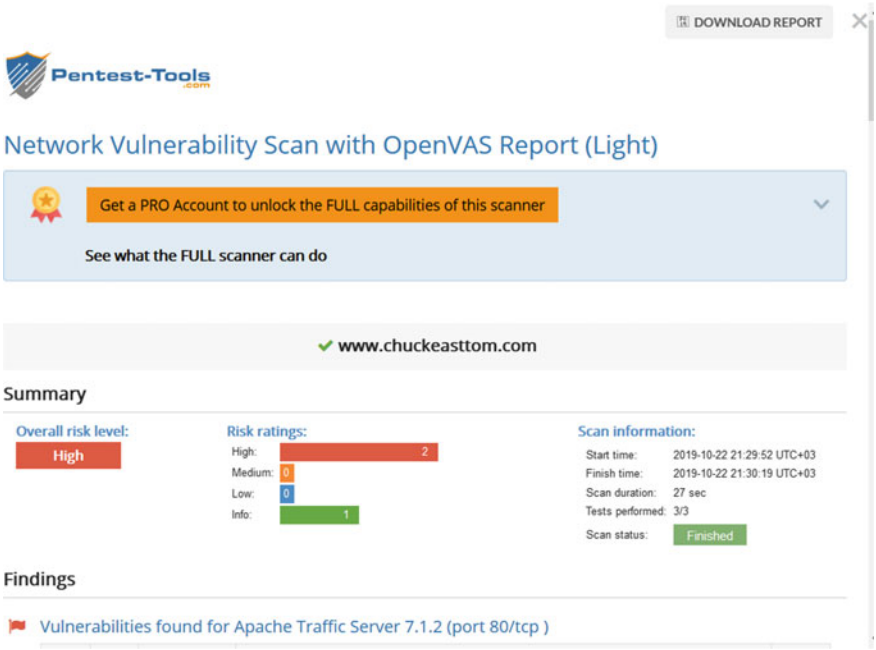

Fig. 12.11 OpenVAS results

You can see the results it provides in Fig. 12.12 .

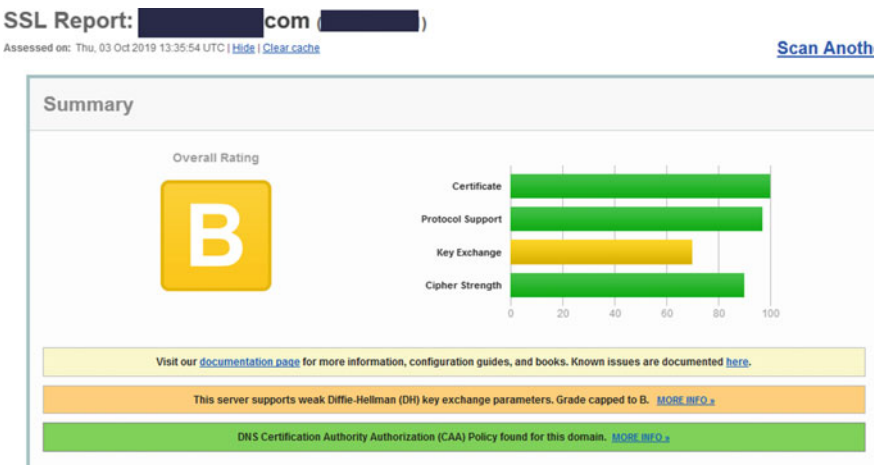

Fig. 12.12 SSL report

Another page that helps check SSL/TLS is Digicert. Digicert is also one of the major certificates authorities. You can see that page in Fig. 12.13. https://www.digicert.com/help/.

There are sites that also allow you to investigate specific threats. For example, Threatcrowd is a very popular thereat intelligence site https://www.threatcrowd.org. You can see that website in Fig. 12.14.

The Sans Internet Storm Center is another good resource for understanding current threats and vulnerabilities. https://isc.sans.edu. This site provides you a good resource for current events in threat research. You can see that website in Fig. 12.15.

Fig. 12.13 DigiCert report

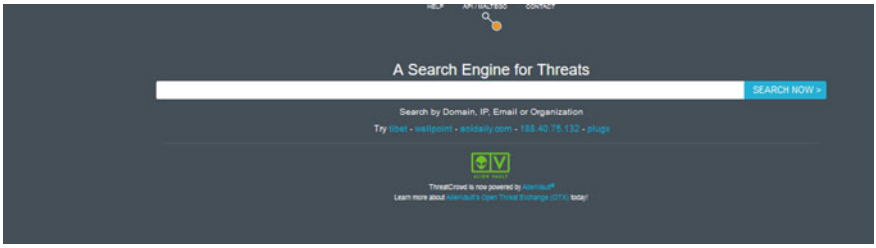

Fig. 12.14 Threatcrowd

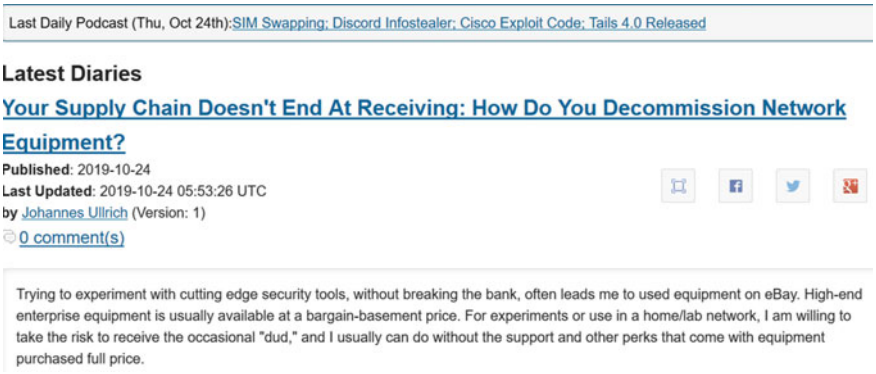

Fig. 12.15 Sans internet storm center

The Sans institute is well-known for publishing cybersecurity research. The storm center aggregates cybersecurity news. It could be argued that a cybersecurity professional should frequently reference the storm center (or some similar source) to keep current with ongoing threats.

Recon-Ng

This tool is an open-source Linux tool. It can be downloaded to any Linux machine. However, it comes with Kali Linux. You will see Kali later in this chapter in reference to Metasploit. This tool is a web reconnaissance tool written in Python. It has several different modules one can load and scan for specific vulnerabilities. There are literally scores of vulnerabilities that can be scanned for. You can see Recon-ng in Fig. 12.16.

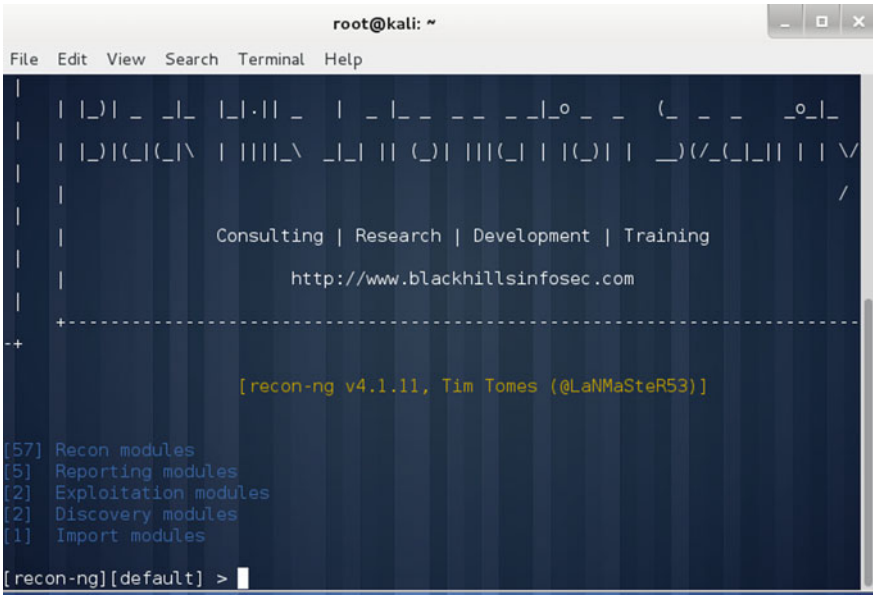

Fig. 12.16 Recon-ng

Metasploit

This tool is important enough to warrant a large separate section to itself. This tool can be downloaded separately, but ships with Kali Linux. Kali is a free Linux distribution. Metasploit is a tool often associated with penetration testing. It is most often described in the context of creating exploits to be delivered to a target. However, Metasploit has a large number of scanners built into it. These can be very useful in vulnerability assessment.

Much of Metasploit can be divided into four types of objects you will work with:

- Exploits: These are pieces of code that will attack a specific vulnerability.
- Payload: This is the code you actually send to the target. It is what actually does the dirty work on that target machine, once the exploit gets you in.
- Auxiliary: These modules provide some extra functionality. For example, scanning. For the purposes of this chapter, we will focus on these Auxiliary modules.
- Encoders: These embed exploits into other files like PDF, AVI, etc. We will see those in the next chapter.

It may be helpful to consider a quote from Rapid 7, the company that distributes Metasploit:

> A vulnerability is a security hole in a piece of software, hardware or operating system that provides a potential angle to attack the system. A vulnerability can be as simple as weak passwords or as complex as buffer overflows or SQL injection vulnerabilities
>
> "To take advantage of a vulnerability, you often need an exploit, a small and highly specialized computer program whose only reason of being is to take advantage of a specific vulnerability and to provide access to a computer system. Exploits often deliver a payload to the target system to grant the attacker access to the system.
>
> The Metasploit Project host the world's largest public database of quality-assured exploits. Have a look at our exploit database – it's right here on the site" (https://community.rapid7.com/docs/DOC-2248)

Even if you don't use Metasploit as a tool for scanning vulnerabilities, visiting the website can keep you updated on existing security issues. In the coming sub-sections, some of the specific Metasploit scanners will be examined briefly.

SMB Scanner

Scanning for SMB is a very important SMB, Server Message Block, is used by Windows Active Directory. When you are scanning for this, you are checking to see if the target is a Windows computer that has SMB running. Now SMB is not a vulnerability per se. In fact, it is necessary for Windows Active Directory. However, there have been numerous exploits of flaws in SMB. The Wannacry virus exploited an SMB vulnerability. The scan is easy:

```
use scanner/smb/smb_version
set RHOSTS 192.168.1.177
set THREADS 4
run
```

Of course, you should replace the IP address 192.168.1.177 with the IP address of the target you are scanning. You can see the results in Fig. 12.17.

While this is simple, it has a lot of information in it. First is simply loading the specific scanner:

```
use scanner/smb/smb_version
```

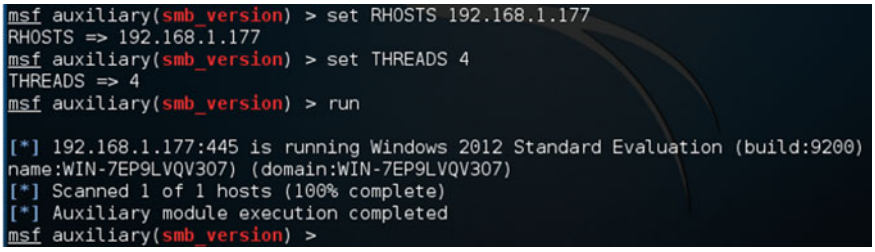

Fig. 12.17 SMB scanner

This is essentially how one loads any module in Metasploit. This is saying that you intend to use a specific module. And it gives the path to that module. Notice the first part of the path is *scanner*. This particular directory has a number of scanners you can use. The next line is:

```
set RHOSTS 192.168.1.177
```

First, notice the RHOSTS. This is the IP address for the remote host(s) you are scanning. Some modules will have RHOST, indicating you can only scan one target, others will have RHOSTS, indicating you can scan several targets if you wish. All scanners will use an RHOST or RHOSTS, but not all exploit modules will Anytime a module has RHOSTS rather than RHOST, you could scan a range of IP addresses. Just modify the command to say:

```
set RHOSTS 192.168.1.177 192.168.1.215
```

Then we have

```
set THREADS 4
```

This is telling Metasploit how many threads to use to run this module. There is no specific rule on this, other than don't select too high a number or your own machines CPU may not be able to handle it. When in doubt, just go with 1 thread. Finally, we have:

```
run
```

Every module om Metasploit ends with either *run* or *exploit*. This just tells Metasploit to go and do whatever you have just setup. If the target does not have the vulnerability you are scanning for, then you will get no results.

SQL Server Scan

This is another scanner that is not inherently a vulnerability. Having SQL Server is an important part of many Microsoft networks. However, being aware that the machines are running and may require patching can be useful. If you carefully studied the SMB scan, then the commands here will be obvious. You type in:

```
use auxiliary/scanner/mssql/mssql_ping
set RHOSTS 192.168.1.177
Set THREADS 1
Set USE_WINDOWS_AUTHENT false
run
```

There is only one new item, which is USE_WINDOWS_AUTHENT false. This is just telling Metasploit that you don't have any login credentials for SQL Server, so don't attempt to login The results can be seen in Fig. 12.18.

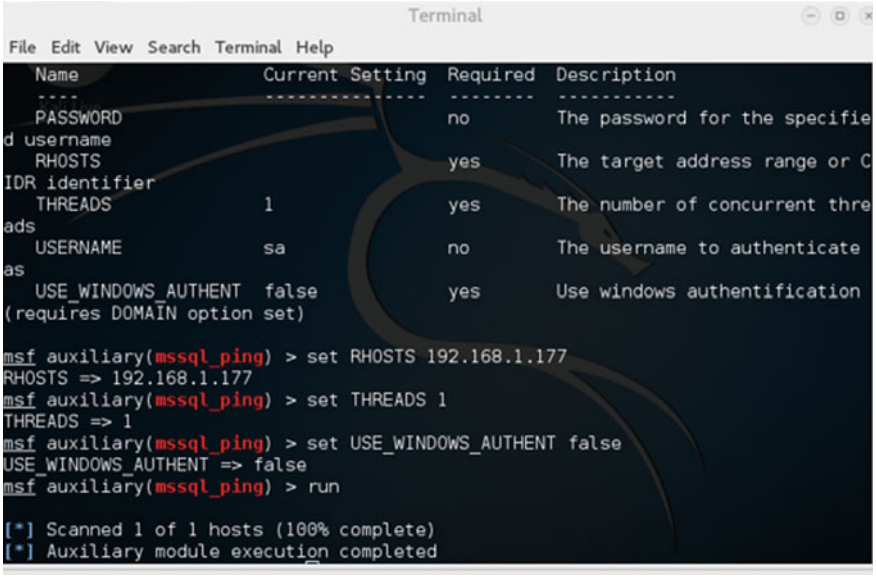

Fig. 12.18 SQL server scan

This is a very simple scanner to run. As we examine other scanners, you will probably notice some commonalities in all the various scanners.

SSH Server Scan

This is a scan to detect SSH (Secure shell) servers on the target. SSH is a secure remote access protocol. So, you may wonder why it would be scanned for. Certainly, you want administrators to utilize SSH rather than options such as telnet. The issue is unauthorized SSH communication. Since SSH is encrypted, it is also an excellent way for a malicious insider to exfiltrate data. That makes it important that you know of any SSH processes on your network. The commands are very similar to what you have already seen:

use scanner/ssh/ssh_version
set RHOSTS 192.168.1.177
Set THREADS 1
Set USE_WINDOWS_AUTHENT false
Run

Since all of these commands have already been explained, no further explanation is needed. You can see the results in Fig. 12.19.

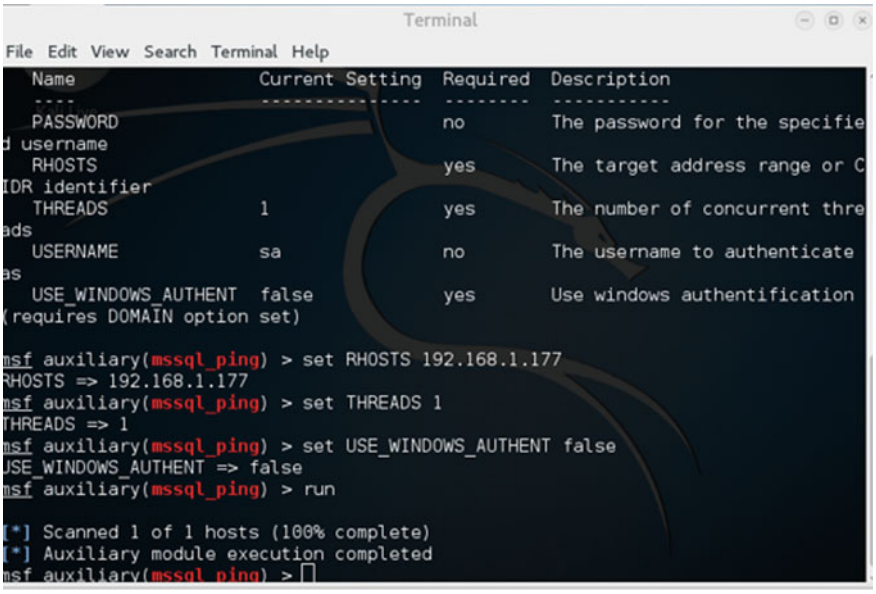

Fig. 12.19 SSH scan

Anonymous FTP Servers

As you might guess, this scans for FTP servers that allow anonymous login. This is a significant vulnerability. Many FTP servers use anonymous logins by default. You need to be aware of when such servers are running on your network,

use auxiliary/scanner/ftp/anonymous
set RHOSTS 192.168.1.177
Set RPORT 21
Set THREADS 1
Set USE_WINDOWS_AUTHENT false
run

You might notice that some scans are in the/scanner/directory, but many are in the/auxiliary/scanner/directory. You can see the results of this scan in Fig. 12.20.

Other Tools

Metasploit also integrates with other tools. For example, nmap is integrated into Metasploit. It is also possible to integrate OpenVAS into Metasploit. These can both be executed without Metasploit. However, integrating these tools along with

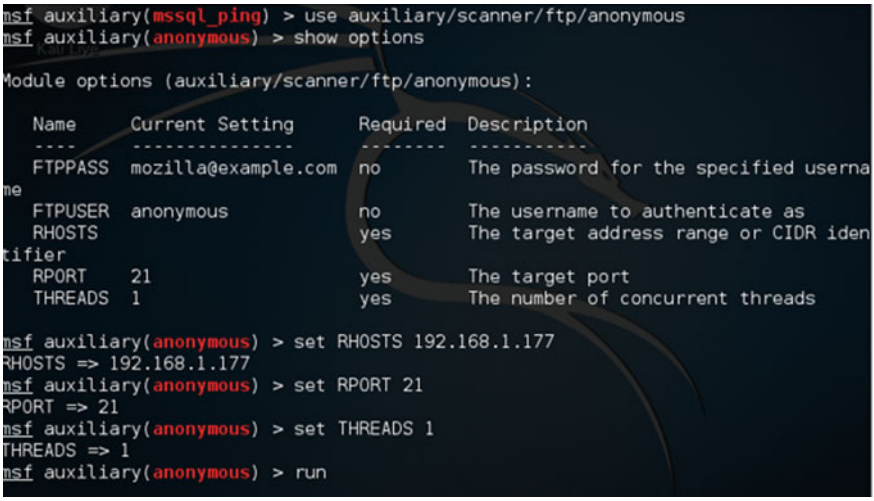

Fig. 12.20 Anonymous FTP scanner

Metasploit gives a single location to run scans from. Also, one can setup Metasploit to record all scans into a database. This allows for reporting of all these scans by querying that database.

Responding to Vulnerability

Now that you have a range of tools to find vulnerabilities, the issue becomes how to respond. This will depend on the nature of the vulnerability. In the case that the issue is simply a missing patch, such as an outdated version of SMB, then the obvious response is to first test the patch, then roll it out to your entire network.

Some issues, however, require a great deal of resources to remediate. It may seem odd, but then one has to evaluate whether not to remediate the issue. There is a basic formula one works though. One quantifies the severity of the issue and the probability of it occurring. The combination of severity and probability gives you an indication of how substantial a risk the vulnerability presents. Chap. 10 discussed CVSS in some detail. Then you compare that with the resources required to address the issue. If the resources required to address an issue then there has to be a decision made as to whether or not to expend the resources.

There are always four possible responses to any issue:

Mitigation: Take steps to reduce the impact or reduce the probability of the event occurring. Anti-virus software is a classic example. It reduces the risk of a virus infection occurring.

Avoidance: This is eliminating the risk. Often this is simply not possible.

Transfer: Transferring the risk means that someone else is now responsible should the event occur. This is often seen as cyber breach insurance. Normally, the insurance carrier will require certain mitigating controls to be in place.
Acceptance: This is very risky. This is essentially stating that addressing the issue costs more than the impact of the risk is realized, or that the probability of it is so remote that it can be ignored. Acceptance should only be embraced after careful threats.

The most common response to any vulnerability or risk is mitigation. How much resources are allocated to mitigation is contingent upon the likelihood of the risk being realized and the impact should it be realized.

Risk and vulnerability response begins by ranking each risk based on probability and impact. Then those vulnerabilities are ordered based on their ranking. The highest-ranking vulnerabilities should be addressed first. Addressing vulnerabilities is much like any other process in cybersecurity. It should be carefully planned and thought out. It is not an ad hoc addressing vulnerability with no order or plan.

Conclusions

In this chapter, you have seen a wide range of tools and techniques for vulnerability scanning. It is important that all security professionals be very aware of the vulnerabilities in their systems. Each of the tools described in this chapter can be part of your vulnerability scanning approach. We also covered briefly how to respond once you find a vulnerability.