

2

Threat Management & Cybersecurity Resources

In the previous section you learned the basic but important concept that attackers can use vulnerabilities to launch attacks and steal data. In this section you'll learn how professionals use something called frameworks to ensure that all aspects of security are addressed, the definitions of vulnerabilities and threats, about the threat management processes of vulnerability assessment and pen testing, which are used by cyber security professionals to find vulnerabilities and fix them before they can be used by attackers, and about the databases of vulnerabilities used in threat management.

Objectives

The specific things you should be able to do at the end of this section are:

1. Describe the purpose of a cyber security framework.
2. Identify common cyber security frameworks.
3. List the major steps in a cyber security framework.
4. Describe how vulnerabilities are defined.
5. List the two main sites with vulnerability libraries.
6. List the steps required to ensure any testing or scanning is performed in a legal manner.
7. Perform a basic vulnerability scan.
8. Use the results of a vulnerability scan to mitigate any vulnerabilities.

9. Explain what a penetration test is.
10. Identify the rules of engagement and how to perform a pen test.
11. Compare and contrast vulnerability scans and pen tests.
12. Define the relationship between updates and patches, and mitigating vulnerabilities.
13. Describe the different methods for ensuring patches and updates are installed.
14. Check an OS or application for update/patch status.
15. Apply what you've learned to your home computers, tablets, phones, routers, and other devices.
16. [Optional Advanced] Describe how TCP/IP addresses and ports are used.
17. [Optional Advanced] Identify commonly used port numbers.
18. [Optional Advanced] Perform a network scan and identify open ports on a device.

Section Content

Introduction

In the previous section you learned that people called attackers try to steal data from computer systems, and that one of the reasons this is possible is because computers have vulnerabilities. If there weren't any attackers or if computers didn't have vulnerabilities there'd be no need for cyber security. But since there are attackers and vulnerabilities something needs to be done to protect data.

But what exactly should be done? I'm sure you're aware of several different aspects of cybersecurity such as using strong passwords and not opening suspicious emails but there are many things that need to be done or considered, and it's difficult for anyone to know and remember everything that needs to be addressed. This is especially true for anyone new to the

field, who may not even be aware that some of these critical items exist and need to be addressed. Luckily, we're standing on the shoulders of giants, and those that have gone before us have built something called frameworks, which are like check lists containing all the items that need to be considered. In this section you'll first learn about frameworks, then about threat management, which is one of the main processes to secure systems against attackers, and finally you'll learn about the databases of vulnerabilities used in threat management.

Frameworks

When you start working in cybersecurity, you'll find that it's different from most other areas of Computer Science or Information Technology in that you won't be given a specific task to work on, you'll just be told you're responsible for security. But what does being in charge of cybersecurity mean on a day to day basis or hour to hour basis? It's not like programming where you need to write a program to meet specific needs, or network administration where you'll be tasked with configuring and maintaining computers and network devices. You won't be given a specific set of tasks, but you'll be the person responsible for ensuring that the organization's data and devices are protected. And as you'll learn in this class there are dozens of tasks you could work on, such as checking for attacks, making sure backups are being created, providing user training, performing vulnerability analysis, etc. Which of these should you do first? And how do you know if you're taking care of everything that needs to be done?

The answer to these questions is to use a resource called a framework, which is what you'll learn about in this section. You can think of a framework as being like an outline which provides a list of everything you need to do but doesn't specify the exact process to be used for any step. It's like saying that a car framework will specify that a car has an engine, a body, wheels, seats, etc. without specifying exactly what type of engine or what type of body. If you build a car using the car framework, you'll at least be sure that you haven't left out any of the necessary car components. Similarly, a cyber security framework will specify everything you need to address such as anti-virus programs, passwords, backups, etc. It won't tell you exactly what you need to do for any of the components, but if you follow the framework, you can be assured you haven't overlooked anything.

There are several cybersecurity and information assurance frameworks available, with a few that are being widely used in industry. These commonly used frameworks include the National Institute of Standards and Technology (NIST) Cybersecurity Framework (CSF), the International Organization for Standardization (ISO) 27001 framework, and the Center for Internet Security (CIS) Controls. All US Federal government agencies are required to use the NIST framework, while the ISO framework is used by much of the rest of the world since the ISO is an international standards group. There are also frameworks that are specific to different types of industries or businesses such as PCI-DSS, which is essential for organizations handling credit card transactions, and HITRUST, which provides a comprehensive set of controls tailored to the healthcare industry.

When you first start in cyber security it can be difficult to keep track of the various frameworks, but for now all you need to know is that all the frameworks serve the same general purpose which is to provide an organized approach to managing all aspects of cyber security. The cyber security frameworks typically contain categories that cover components such as the following:

- Risk/Threat Management - This involves identifying potential threats, assessing vulnerabilities, and determining the potential impact of these risks on the organization. By prioritizing risks based on their severity, organizations can allocate resources more effectively and focus on protecting their most critical assets. Identifying and fixing the most critical problems first forms the backbone of any cybersecurity strategy.
- Security Controls - These are the specific measures or systems an organization implements to mitigate identified risks. These controls can be preventive, detective or corrective. Preventative measures are those designed to stop an attack before it occurs, detective measures are those meant to identify an attack in progress, and corrective measures are those aimed at restoring systems after an attack has occurred. Common examples of security controls an organization might implement include firewalls, intrusion detection systems, and encryption protocols.

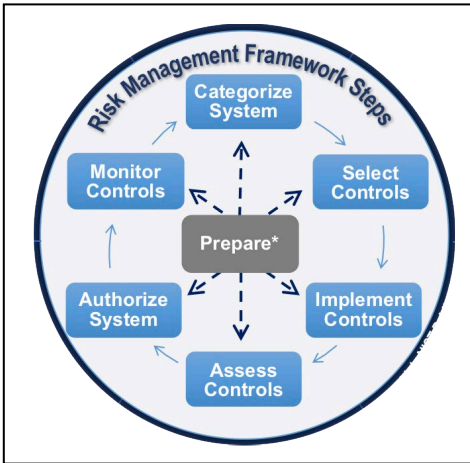
- Policies and Procedures - These include the security rules or policies an organization might use, and ways for implementing those rules. For example, an organization needs password policies that specify password complexity, the number of acceptable failed login attempts, password expiration, etc., along with procedures that include the steps for implementing the password policies.

Note that these are just some of the things that the cyber security frameworks should address, and each framework may use different names for the categories. And like the car framework, they tell you the items you need to consider, but don't tell you exactly what to do. For example, all frameworks specify that threat management needs to be done, but they won't say exactly which applications should be used.

The great thing about using a framework is that, regardless of what each component is called, the framework gives you a list of things that must be addressed to have comprehensive security.

Another great thing about using a framework is that utilizing a framework allows you to move from a reactive position to a proactive position. That is, without a framework you'll probably end up running around, responding to problems as they pop up. You'll be reacting to problems as they occur, instead of taking an organized approach to stop the problems before they occur which is what a framework is designed to do.

The last introductory concept about frameworks is that using one is a repetitive process, as shown in the following figures. It's not something that you do once and say you're finished.



The specific steps defined in each of the frameworks illustrated above have different names or labels, but they basically do the same things. Initially you go through the framework and address anything you may have missed. Everything else you'll learn about in this class, such as anti-malware, password policies and systems, incident response, etc. will be part of any framework.

After checking that everything is covered, the process followed in most frameworks is to define the vulnerabilities and risks for a system, and then mitigate or fix any vulnerabilities that you

find starting with the most critical issues. You start by mitigating the most critical risks, but after they're addressed you repeat the process for any remaining risks or check for new risks.

It's kind of like going to the doctor to check your health, and the doctor finds you have a broken leg, you have acne, and you fall asleep when you watch the videos I've made for you. The doctor will triage your symptoms and treat your leg first, since that's the most serious ailment. (Just make sure and go to the doctor and not my Grandpa Dee. Grandpa Dee is an old school veterinarian and if you break your leg you get to "go live on the farm" with your old dogs Allie and Ellie.) After your leg is fixed, the doctor will repeat the process and treat the acne, which is your next most important issue. In other words, it's a cyclical process where you address the biggest problems first, but continually look for and address any problems that may crop up. The important thing to understand is that cyber security isn't a one-and-done job, it's a process of continuous evaluation and improvement, and the frameworks provide a defined way to perform the process.

Getting back to the questions that were asked at the start of this section, about what steps should you be taking at any time as a cyber security professional, using a framework provides you with a clear roadmap. The framework will ensure you haven't missed any items, and once you have all the items in the checklist covered, the framework will provide a method for constantly reviewing and improving your security posture. And on the other hand, I've seen highly skilled individuals work without a framework. These were people that were obviously technically competent, but they always seemed to be putting out fires, rushing to fix problems as they occurred. This prevented them from making any overall progress and resulted in a lot of security issues that would have been preventable if they had used a framework.

Summary

In this section you learned about frameworks, which are used by cyber security professionals to implement all aspects of cybersecurity and information assurance for an organization and ensure that every important factor has been addressed. Each framework uses a cyclical process to evaluate an organization's current security posture, and then address any weaknesses. If

implemented properly this allows you to continually improve and strengthen an organization's security posture as time goes by and move from a reactive situation to a proactive model.

The Pledge

Before we get move on and talk about threat management, we're going to go over one super critical factor about cyber security and the things you'll learn in this class and in other cyber security classes which is that you're going to learn how to do some hacking which can get you in serious trouble if you aren't careful. You'll learn more about the specific laws covering computer crimes in other classes, but for now it should suffice to say that running vulnerability scans or pen tests on computers or networks where you do not have prior authorization is a violation of the law. The programs for doing these types of scans and assessments are easy to find and use, but you should only use them to test systems after you have written permission from the proper authority. If you run these types of scans against a system or network it will almost certainly be detected and logged, and this information can be used to easily trace the attack back to your computer. If you do a scan or pen test with the company's permission, it's called an assessment and you get paid to do it. If you do a scan or pen test without permission, it's called an attack and you get to go to jail.

When we were first putting the cybersecurity degree program together, we checked out the degree programs at other colleges and universities. There weren't many, but we noticed that most of the programs required students to pass a security background check before they were accepted into the program and allowed to start classes, and individuals who had certain types of felony convictions would not be allowed into the program. This was because students would be taught how to do things that could be illegal, such as hack into systems, crack passwords, etc. We also checked with the local employers who said they would not hire anyone convicted of certain crimes, and that it would be very difficult for someone with a criminal background to start a career in cyber security.

We took this by our college administration with the recommendation that CBC also require background checks. Their position was that even though a background check was prudent for the student's sake, they would not do it. The explanation was they previously had a similar requirement for entry into the nursing program but had been sued by a student who felt that regardless of the implications for employment, they were being denied the chance to pursue their career of choice. Because they didn't want to take the chance of facing another lawsuit, the administration decided to take the route that, while not best for the students or the program, was easiest for them. They suggested that we simply tell anyone entering the program about the problems felony convictions could cause. They also suggested that we ask/require the students to take a pledge, to only use the things they were being taught in a legal manner and to not do anything illegal. I initially thought they were joking about the pledge, but it turns out they were serious.

So ... at this point in the class you need to take the cyber security pledge. To take the pledge follow these steps:

1. Stand up.
2. Place your right hand over your heart.
3. State the following:

I (state your name) promise to use my powers only for good, and not for evil.

I know this may sound like a joke but it's very serious. You should do this even if you're at home alone. I'm going to repeat myself to ensure you know that this is serious business. Never do things like vulnerability analysis, or pen tests, or password cracking, anything you learn in this class or in this program against a system or network unless you are the system owner or have

prior written authorization from the system owner or one of their legal designees. This means you must not practice or use what you learn on computers owned by your friends or family members without their permission, and you must not practice or use what you learn on computers at the college unless you have been given express permission by the instructor.

And just so you understand how serious this can be, and the possible consequences, here are two cases that have happened at CBC. These happened before we started the cyber security program, but they involved computer science students and should show you that the college and law enforcement treat cybercrimes very seriously. The first involved a student who had already been convicted of breaking into the computers at their high school, which they apparently did to change their grades. Due to their age, they were given probation instead of jail time, with the stipulation that they stay out of further trouble. After graduating from high school, they started taking classes at CBC and soon tried breaking into the network servers at the college. Our network administrators monitor the college networks closely and have alarms that will automatically trigger when attacks like this person tried are detected. The attacks were noticed very quickly which made it easy to identify the person performing the attack. The student tried to claim that they had their instructor's permission to justify their actions, but a short investigation showed that this was false. The information about the attack was turned over to local law enforcement, the student was kicked out of the college, lost their probation, and ended up doing some time in prison.

The second case involved a student who was working at Walmart, back when it was located on Canal Drive in Kennewick. Walmart was moving to a new location which was one of their current stores on 27th in Kennewick. For some reason this upset the student, and they sent death threats to one of the Walmart vice presidents back in Bentonville Arkansas. The Walmart VP reported the threat to their local law enforcement, who were able to trace the source of the email containing the threat to an account created from a computer at CBC. The Bentonville Sherriff's department contacted CBC and told us they needed to confiscate all the computers in one of our labs as evidence. We initially thought this was a joke as the Sherriff that called had a classically stereotypical southern name, something like Sherriff Bubba Tillis. But it turns out the

Sherriff and his request were legitimate. Luckily CBC was able to check that computer's logs and found that a specific student was using the computer at the time the email account was created and the email was sent. I'm not sure of the final outcome for the student, but I was told they ended up facing federal charges as this involved multiple states.

The point of these stories is to ensure that you realize that the college treats computer crime seriously. Our network administrators have systems in place that constantly look out for any types of attacks or abnormal activity and can quickly identify the system and user behind the attack. Whether it's an attack against the college, or if the college computers are used in crimes against other systems, it will be taken seriously and passed on to law enforcement. As a cleaned-up version of the saying goes "Mess around and find out ..." Or in this case, remember your pledge and don't mess around!

Threat Management Terminology

In this section you're going to return to our basic concepts of attackers and vulnerabilities and learn about one of the first steps specified in most cybersecurity framework processes, which is to figure out what type of security is currently in place and what holes may exist so they can be fixed before an attacker uses them. In technical terms this step determines an organization's or a system's current security profile, and having the profile makes it possible to determine what steps to take next so any threats can be mitigated.

The formal name for this process is Threat Management which the book defines as "taking the appropriate steps needed to minimize hostile cyber actions", which is pretty vague. Another definition for threat management is that it consists of performing vulnerability scans or vulnerability assessments, and penetration tests. But unless you already know what those terms mean, this definition is probably equally unhelpful.

To understand what threat management is and what it's meant to accomplish, the first thing to do is to learn a few new terms including vulnerabilities, threats, risks, and mitigation.

Vulnerability

Let's start with the term vulnerability. In cybersecurity a vulnerability is a weakness or flaw in a system or program that could be used by an attacker to gain unauthorized access, disrupt services, or steal data. Vulnerabilities can exist in hardware, software, networks, or wetware which is a way to refer to the human computer users.

Hardware Vulnerabilities

Hardware vulnerabilities are flaws or weaknesses in the CPUs, memory, or other physical devices that make up a computer or network system. There aren't a lot of these, but some examples are Meltdown and Spectre which are attacks that take advantage of flaws in CPU architecture.

Normally the CPU and OS work in tandem to protect sensitive data, such as encryption keys or passwords, used by the OS. When the Operating System loads it tells the CPU that certain areas of memory that it uses should be off-limits to any other program. If another program tries to access the protected memory the CPU will deny the request. But it's been found that under certain conditions some CPUs could be tricked into giving access to the protected areas of memory allowing programs run by a user to read the sensitive data.

To explain in non-technical terms this let's use an analogy of a bank and bank guard. The bank has a vault where it stores large amounts of money that it uses to operate, and also has safety deposit boxes that the customers can use to store their own valuables. One of the bank guard's jobs is to protect the main vault by keeping the customers out and ensuring only the bank employees are allowed in. Now let's say that you discover that all bank guards have a flaw and can't see the color purple. Anyone dressed in all purple and wearing a purple mask will be invisible to the guard, allowing them to access the main vault and take whatever they want.

It's not critical that you understand how all the technical details behind these hardware vulnerabilities, but you should understand that they are only possible because of flaws in the

CPU design, and the CPU is hardware. You should also note that hardware vulnerabilities are relatively rare.

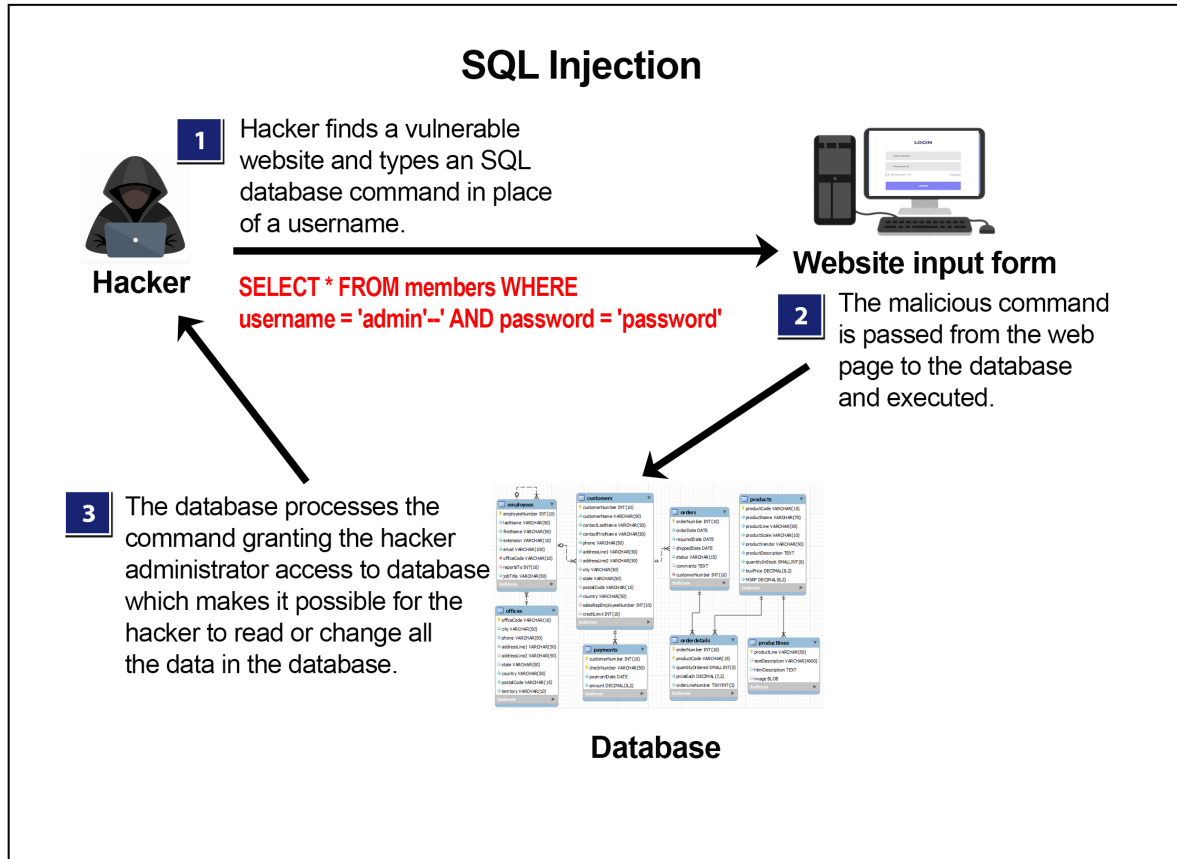
Software Vulnerabilities

Software vulnerabilities arise from flaws in the code of software applications or operating systems. These can be caused by mistakes in the program code left by unwitting or naive programmers, or they can be purposefully built-in by an attacker with plans to take advantage of the hole once the program is loaded on the victim's computer. Understanding some of the software vulnerabilities requires specialized knowledge in programming or specific applications. There are thousands of different software vulnerabilities, and you'll go into more depth on them in the chapter on Threats and Attacks. For now, you just need to know they're caused by flaws in programming code. To give you some perspective, here are a few examples of some of the easier to understand software vulnerabilities.

The first simple software vulnerability is called a SQL Injection attack. This is a common attack method where the attacker tries to take advantage of mistakes made by website and database programmers. SQL injection attacks can easily be prevented with a few programming commands, but new programmers often don't realize they should check for the "bad" commands, creating the vulnerability.

When an attacker finds a website with this vulnerability, they can trick the site's database into giving them administrator access, revealing protected data or allowing the attacker to make unauthorized alterations. This is done by inputting malicious database commands into a form on a web site. For example, say a computer manufacturer's website, like most web sites, has a page where customers can enter their username and password to login to their existing accounts. Instead of typing a username in the username text box, the attacker types the database command to allow them to login as the administrator and gain full access to all the data in the database. If the website isn't built securely, the database will process the command and give the attacker full access to the database. This allows the attacker to view or download

the customer account database tables which include the customer names and credit card details, or change or delete any of the information in the database.



When a programmer intentionally adds a software vulnerability into a program it's called a backdoor. An example of a backdoor was discovered in early 2024 in the open-source data compression software XZ Utils. Data compression takes several files and combines them into a single file to make them easy to store and transport. On Windows systems these files often have a .zip extension. The XZ Utils utility was widely used on Linux systems and the backdoor was luckily discovered by Andres Freund, a Microsoft software engineer, who noticed unusual network behavior when using the XZ Utils program. Freund found that a backdoor in XZ Utils could allow unauthorized remote access to systems running the compromised version of XZ Utils, posing a significant security risk to users and organizations relying on this software. It turns out that the backdoor done by an attacker who infiltrated the XZ Utils project by

contributing code and building trust within the community supporting the open-source code over an extended period.

[Optional] Additional Software Vulnerabilities

This section contains some examples from the “old days” which might be easier to understand than most modern vulnerabilities. In the early days of computing and networks, software and systems were often wide open by default. Most application and OS developers assumed that users would want access to most, if not all the available functionality on a computer immediately without having to do any complex setup.

For example, early versions of Microsoft Windows had many network services such as file sharing enabled by default. If file sharing is enabled other users on the network can see and access the files on the system which is an easy way to share files between computers in a home network or small office network. But since having file sharing enabled makes it easy for anyone to access a computer’s files, it also makes it easy for an attacker to read the files. Imagine the risk to the information stored on your computer if anyone connected to the same network could access every file on your device. The important concept to note is that unrestricted file sharing is a vulnerability.

An example on the UNIX/Linux side was that by default most systems had a guest account that anyone could use to login to the system. This made it easy to share files with lots of users, as you didn’t need to create a new account for each user, they could just login to the system using the guest account and access the file. But the ability to logon to a system without an account also helps attackers a huge advantage as it provides an easy to access starting point into the system and saved them hours of trying to crack into the system using more difficult methods.

If a user didn’t want certain features, they would have to manually opt-out or disable them. If any of the features made a program less secure or left security holes this opt-out system placed the burden on the user to secure or streamline the software. In other words, developers were more focused on providing maximum functionality out-of-the-box, with security or privacy settings often being secondary concerns.

The way to mitigate these vulnerabilities was to disable the services unless you needed them. While this was often a simple fix, it did require the system owner to take some action and while the steps for implementing a fix may seem easy, they were often difficult for non-technical users.

As security became a larger issue, there was a broad shift industry regarding the initial security settings and there was an evolution in the approach to how much control users are given over configurations and how much is enabled or disabled by default. Over the years, software design has shifted from a more permissive, open configuration (opt-out) to a more restrictive, secure-first model (opt-in) where software and systems are closed or locked down by default and action must be taken to start most services. This means that most features, especially those related to networking, data sharing, and potentially risky services, are disabled by default and must be manually enabled by the user.

While this shift from opt-out to opt-in made systems more secure by mitigating many potential vulnerabilities, it didn't eliminate all vulnerabilities. It took away the easy to find security holes, but attackers started looking for, and finding other holes. The vulnerabilities have become increasingly difficult to understand and patch due to several factors which include the growing complexity of systems, advanced attack methods, and the sheer number of interconnected devices. Modern software and hardware environments are more sophisticated than ever, with layers of abstraction, dependencies, and integration points. This has introduced a new range of challenges for security professionals and developers when identifying, understanding, and addressing vulnerabilities.

Modern applications and systems contain millions of lines of code, developed by large teams across various frameworks and languages. With so many components, identifying where a vulnerability lies can be a daunting task. If you've done any programming, you've probably used built-in methods or functions such as a square root function or other math functions, or different functions for opening files, etc. These functions may call other functions, and some of the code being used in these functions and libraries may have been written decades ago when security wasn't a priority. Vulnerabilities can be deeply embedded in obscure parts of the code,

making them harder to spot. Many applications rely on third-party libraries, open-source components, and external APIs. If one of these dependencies has a vulnerability, it can be difficult to find, much less fix, especially if the library is maintained by a different entity or organization.

[OPTIONAL] A Modern Software Vulnerability

Here's an example of a modern vulnerability which demonstrates how sophisticated the attack is, and how it's hard to even understand what causes the vulnerability. The name of the vulnerability is Log4Shell (CVE-2021-44228) and it affects systems running the Apache web server, which is the web server used on most Linux systems. Apache the web server used by most of the Internet during the early 2000s and still powers ~20-25% of web sites, so the vulnerability was a significant issue. The vulnerability was in a Java-based logging utility called Log4j that helped create and process the web server logs. It was used as a logging utility in many third-party libraries and frameworks, and developers often didn't realize that their applications even depended on it.

The vulnerability is called a remote code execution (RCE) vulnerability and it allows attackers to take control of affected systems by tricking Log4j into executing arbitrary code. The vulnerability occurs when Log4j logs certain types of untrusted data, such as strings that are passed into logs from user inputs. This input could contain special characters that trigger a JNDI (Java Naming and Directory Interface) lookup to a malicious server which makes it possible for an attacker to execute malicious code remotely on the server using Log4j. The vulnerability exploited the obscure interaction between Log4j, user input, and JNDI lookups, which were not immediately obvious to developers.

Don't worry if you don't understand much about the Log4j vulnerability. Understanding how the JNDI exploit works and its potential to execute malicious code requires detailed knowledge of programming and of Java's internals. The concept that this is meant to illustrate is that modern vulnerabilities can be complex and hard to understand. Modern vulnerabilities like the Log4Shell exploit intricate interactions between software components, making them harder for

developers to fully understand and address. The growing use of complex libraries, APIs, and middleware increases the chances of unexpected vulnerabilities.

So even though developers have learned their lessons and tightened up security in the programs they distribute, new vulnerabilities are still being found. The problem for you as a cybersecurity professional, is keeping track of all the different vulnerabilities on all the different systems as every device and every OS has a specific list of vulnerabilities, and new vulnerabilities are found every day. You don't necessarily need to understand the details behind every vulnerability, you just need to be able to know if the systems you're protecting have any, and if any are found what you need to do to mitigate or fix them.

Network Vulnerabilities

Network vulnerabilities are caused by weaknesses in the infrastructure or protocols that enable communication between devices on a network. While it's very simple to connect to and use the Internet there's a lot of going on behind the scenes. For example, any data you transmit on the Internet won't go directly from your device to the recipient device. Instead, the network data will typically be inspected and forwarded by dozens of routers. Each router will look at any the network data it receives, read the destination address, and then forward the data on the next hop towards its final destination. Note that every router that handles your transmission can read the data being transmitted, which is why all sensitive network transmissions should be encrypted. If there are problems with the routers or encryption protocols used in the Internet infrastructure it is considered a network vulnerability.

Wetware (Human) Vulnerabilities

Wetware or human vulnerabilities are those that are caused by the humans that use computers and systems. It turns out that these types of vulnerabilities are the easiest to understand but they are also the most difficult to deal with.

Users can make errors in judgment or poor choices and do things like select weak passwords or leave laptops where they can be stolen. Or they can be manipulated in social engineering

attacks like phishing schemes or catfishing (fake romances). The remedy for these types of attacks comes down to knowledge and training, or the use of things like multi-factor authentication (MFA). But since users are human, it means they can be lazy, greedy, scared, or they can make honest mistakes and it's difficult, if not impossible, to ensure that users won't do something that will help an attacker.

In my opinion this shows that there's something intrinsically wrong with computers and computer security. The way systems and networks are designed, the attackers have every advantage and all the pressure and responsibility is on the users to protect their data. That is, an attacker can easily use a fake identity to try and fool users, and all the pressure is on the user to decide whether an email, phone call, letter, or website is legitimate or not. Security would be much easier if the attackers did not have the ability to remain anonymous or create new fake identities with the press of a button.

In addition to users, you should also be aware that administrators should be considered as vulnerabilities, maybe even a greater threat than the users. They're a greater threat because they are the ones that know about and control the security systems, which means they also know about any holes in the security and how to take advantage of them.

The ways to prevent or catch insider attacks performed by an administrator are to ensure there's adequate monitoring and scrutiny. This can be done by logging and auditing all actions taken by administrators and having more than one administrator on each system.

Threat and Risk

The next two terms to define are threat and risk. A threat in cybersecurity refers to any potential danger or event that could exploit a vulnerability in a system or network with a resulting negative impact such as unauthorized access to confidential data, loss of integrity in the system or its data, or loss of availability to authorized users. We usually think of threats as originating with hackers or disgruntled employees, but they can also originate from natural events such as a fire or earthquake, or accidental actions.

A risk in cybersecurity refers to the potential for loss or damage to an organization's assets, data, or reputation due to a threat exploiting a vulnerability. In other words, if an attack succeeds what's the damage going to be. The key components of any risk are how probable is it that a threat will be successful in exploiting a vulnerability and the severity of consequences if the threat succeeds.

The following formula can be used to rank and compare risks:

$$\text{Risk} = \text{Threat Success Probability} \times \text{Impact}$$

For example, assume that having unlocked doors is a vulnerability. When a security consultant checks an electronics store they find that all the doors are locked except the door between the storage room and a back alley and the door to the janitor's closet which is located in the basement. The storage room is used to hold things like the computers, cameras, and phones sold in the store, while the janitors closet holds cleaning supplies. In this case the risk from the unlocked door in the storage room will be greater than the risk from the unlocked closet because both the possibility of someone using the alley door to get into the storage room is greater than the possibility of someone getting into the basement and the janitors closet, and if someone steals something like phones or cameras it will have a much greater financial impact on the business than if someone steals a broom or paper towels from the janitors closet.

The relationship between threat and risk is that a threat is the cause of potential harm while a risk is the likelihood and impact of the harm happening.

Threat Management

Now that you know about vulnerabilities and threats let's get back to discussing what threat management means. In plain English, threat management consists of checking any device or system for potential vulnerabilities or holes, and then mitigating or fixing those vulnerabilities so they can't be used by an attacker. While there are dozens of actions that can be taken to accomplish these tasks, in cyber security threat management refers to two specific processes

called vulnerability assessment/scan and penetration testing, which is commonly called pen testing.

Vulnerability scans or vulnerability assessments can be used to check a system for vulnerabilities in hardware and software, but they aren't used to check for network or wetware vulnerabilities. This is because they use an automated scanning process that checks the hardware and software on a system against a database of known vulnerabilities, and there's no way to include all the things a user may do or all the things that may happen on a network in the automated scan.

One thing to note before we move on is that the terms *Vulnerability Scan* and *Vulnerability Assessment* refer to two slightly different things. Most lay people don't make the distinction between vulnerability scan and vulnerability assessment, and the terms are often used interchangeably. However, *Vulnerability Scan* refers to the process that will build a list of vulnerabilities, while *Vulnerability Assessment* refers to doing a scan, and then assessing the results of the scan and deciding what to do about any vulnerabilities found in the scan. In reality, and in the workplace, anyone can run a scan as the only thing it requires is running a program, but performing an assessment requires a cyber security professional who knows how to interpret the results of the scan, decide whether any vulnerabilities that were found are important or not, and what to do if something needs to be done. Performing a vulnerability assessment takes more experience and knowledge than just running a scan, as you need to be able to recognize how critical each vulnerability may be and what they mean to your organization.

Penetration tests can be used to check for all the categories of vulnerabilities, including network vulnerabilities and wetware vulnerabilities. Most pen testers start with a vulnerability scan to check for hardware and software vulnerabilities, but then continue and try to leverage the same network and human vulnerabilities that an attacker would try. This means that a pen tester may do things like testing an organization's users with phishing emails and phone calls. The extents that pen testers can go through when trying to trick users are called the rules of engagement. These should be negotiated before starting the tests to ensure the pen testers

don't go out of bounds and to ensure the pen testers are legally protected for performing acts that would otherwise be illegal. Performing pen tests requires extensive knowledge and experience as dealing with users can't be fully scripted.

Mitigating or Addressing Vulnerabilities

The last term to learn about is mitigate or mitigation. Mitigation in cybersecurity refers to the process of reducing or eliminating the likelihood that a vulnerability can be exploited as well as lowering or minimizing the impact if it is exploited. The processes used in threat management, vulnerability scans and pen tests, allow cyber security professionals to take a proactive approach by finding and mitigating vulnerabilities before they can be used by attackers.

The actual steps used to mitigate vulnerabilities will vary for each category of vulnerabilities. In general terms these include the following:

Hardware - Regularly update firmware and use devices with built-in security features.

Software - Install updates and patches promptly, only download and install programs from trusted sources, and use secure coding practices during development.

Network – Use devices like firewalls, intrusion detection systems and intrusion prevention systems to monitor and screen network traffic, close unused ports, and use encrypted protocols like HTTPS or VPNs.

Wetware - Educate users on cybersecurity best practices and implement multi-factor authentication (MFA), and ensure that the actions taken by administrators are logged and audited.

Vulnerability Scans/Analysis and Pen Tests (Threat Management)

Introduction

In this section you're going to learn more about vulnerability scans/analysis and penetration tests, the two processes used in threat management. Remember that these processes allow you to work proactively and check your systems the same way an attacker would and take care of any weaknesses you might find.

Both processes start by scanning a system for hardware and software vulnerabilities. This scan has a few steps because there are hundreds of thousands of vulnerabilities, maybe millions, with new ones being found every day, and the vulnerabilities are platform specific. The first step in threat management is actually building the database of vulnerabilities, which is the first thing you'll learn about in this section. After that you'll learn about the vulnerability analysis programs that use the database to check a specific system for hardware and software vulnerabilities. This is followed by penetration testing, which provides a way to check for network and wetware vulnerabilities.

To help provide perspective on threat management we'll first use a couple of analogies with things we do in the real world like checking a car for defects and providing security to an apartment building. These should give you a good perspective on what happens during vulnerability assessments and penetration tests, and how they differ from each other. At the end of this section, you should be able to describe what happens during vulnerability assessments and penetration tests and be able to list the organizations that provide the giant databases that are used to track vulnerabilities.

One thing to note, as always in this class, you'll learn a lot of new terminology. You'll also get a brief glimpse at the programs and processes used vulnerability scans/assessment, and penetration testing. These are all very interesting subjects but once again there's not enough time in the class to do much more than give you a small taste of each. If you find them interesting, you're in luck as you can look forward to learning more about them in later classes.

Analogies

Before we jump into the details of vulnerability assessment and penetration testing, let's look at two analogies which should help to provide some perspective and differentiate between the two processes. The first analogy is looking at car repair, and the second analogy is checking security for something like an apartment building or a shopping mall.

Let's start by looking at car repair and car maintenance and how this could be done by a mechanic. When you take your car in for maintenance how does the mechanic know what to do? There are thousands of possible things that could be broken or not working right, way too many to check them all. One of the ways to decide what to do is to use a database containing a list of maintenance items.

If you take your car in for repair one of the first things the shop will do is check for scheduled maintenance items like such as oil changes, air filter cleaning or replacement, brake checks, etc. The shop will also check for any recall or special maintenance notices. Each auto manufacturer builds, maintains, and distributes a database that contains scheduled maintenance items notices and special maintenance items. Each item is linked to a specific type of vehicle so when a car comes into the shop the database can quickly generate a list of items to check or repair. This list makes it easy for any mechanic working on the car as they can just check the items on the list.

One of the main components in this analogy is the database of maintenance items. It should be noted that:

1. The database exists at all.
2. The maintenance items are each associated with a specific vehicle.
3. The database lists the possible problem along with the actions that should be taken, especially those actions required to fix special maintenance problems or recall issues.

In cyber security, you'll learn about two databases that contain lists of vulnerabilities and provide a similar service as the car maintenance database. These are called the Common Vulnerabilities and Exposures (CVE) and National Vulnerability Database (NVD) and they contain a list of platform specific vulnerabilities along with actions that should be taken to fix the problem(s).

The process of checking a computer for vulnerabilities against the CVE or NVD database is called a vulnerability scan or vulnerability assessment. The vulnerability assessment process is similar to checking the car maintenance list where the database will report different results for each different car as each vulnerability assessment looks at a specific platform, or specific combination of computer hardware and software.

One thing to note is that the mechanic can use a database to check for problems with a car, but it doesn't check for problems with the roads and bridges the car is driven on, and it doesn't check for the cause of most car problems which is the person driving the car. This same limitation holds true with vulnerability scans/analysis. That is, the CVE and VND databases hold information about vulnerabilities in hardware and software, but they don't hold information about vulnerabilities in network infrastructure or the cause of most problems, which is the computer users.

The second analogy we'll use is looking at how someone in charge of physical security for a large property like an apartment building might proceed. The building might have layers of security, like security cameras around the outside of the property, guards or key card access to the garage and lobby, and locks on each apartment door. If you oversaw security, how would you know if the building is secure? There's not going to be a database of possible problems like there are with cars.

In this case one of the ways to check how secure the building is, is to actually try breaking into the property. That is, you'd test the building's security by doing the same things you'd expect someone trying to break into the building would do. You might try something as simple as "doorknob rattling" attack where you manually check each individual door to ensure they're

locked, and the locks are secure. You can check all the hardware like the locks, windows, and fences, but you can also do more sophisticated and complicated attacks where you test the city's infrastructure and the people managing or using the building.

To test the infrastructure, you might see if you can gain access to the building through the sewer system, or maybe disguising yourself as someone who works for the power company and seeing if you can gain access to the building. To test the people, you could try making a fake company ID or pretend to be a food delivery driver and see if you can walk into secure areas of the building without being challenged. Since most people have a desire to be helpful you may even get people to open doors for you if you have your hands full of pizza boxes.

Similarly in pen tests you may try to gain access to a device by first utilizing attacks on the network infrastructure, or by checking the organization's users to see if they'll fall for schemes designed to trick them into giving you access to a system or information. The important thing to note is that pen tests go beyond scanning for hardware and software vulnerabilities and check for network and human vulnerabilities.

Hopefully these analogies give you some perspective on the components of threat management, which are the CVE and NVD databases, vulnerability assessments, and pen testing.

The Databases Used in Vulnerability Scans/Assessments

One of the main components of threat management are the databases that are used to track vulnerabilities. There are hundreds of thousands of vulnerabilities, over 240,000 in early 2024, with new ones being found every day. The sheer number of vulnerabilities makes it impossible for any one person to keep track of them all. It would be like trying to keep track of every video game that could be played on any platform, from PC games to VR games, to handheld games, or trying to keep track of every movie and TV show ever made, or trying to memorize every athlete who ever played college or professional sports.

Luckily there are a few groups that track vulnerabilities and have built databases that you can use to check for problems on a specific system. One of these is the Common Vulnerabilities and Exposures (CVE) database run by the Mitre Corporation, and another is the National Vulnerability Database (NVD) run by the NIST. You can think of the CVE and NVD databases as being like Wikipedia in the sense that they're huge central silos of information that anyone can access. But in this case, they're only used to track vulnerabilities, and unlike Wikipedia not just anyone can add or edit the database entries.

A key point about the CVE and NVD databases is that while they are closely related, they serve different purposes and store distinct types of data. Here's a quick look at how these databases function, the data each holds, and how they relate to each other.

The CVE database is designed to provide a standardized system for identifying and cataloging known vulnerabilities. Each CVE entry offer minimal technical details, focusing on identification and standardization rather than deep technical analysis or remediation advice. Each vulnerability in the CVE database is assigned a unique identifier called a CVE ID, e.g., CVE-2024-12345. The CVE ID acts as a reference point across security tools, advisories, and with the NVD database. In other words, the CVE ID is the key or index number used to track vulnerabilities across multiple databases and multiple database tables.

The data fields held in the CVE includes the following:

- CVE ID: A unique alphanumeric identifier assigned to each vulnerability.
- Description: A brief summary of the vulnerability, including what software or system is affected and the nature of the vulnerability.
- References: Links to external advisories, blogs, or other reports that provide additional information about the vulnerability.
- Status: Indicates whether the CVE entry is in Reserved, Published, or Rejected status.

The following figure shows the CVE database entry for a vulnerability in an application called Geyser that's associated Minecraft. Note that the data includes who added the record along

with references that explain why it was added. And while it doesn't contain information about how to mitigate the vulnerability it includes a link to the NVD database entry which holds the mitigation instructions.

CVE-ID	
CVE-2021-39177	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	
Geysers is a bridge between Minecraft: Bedrock Edition and Minecraft: Java Edition. Versions of Geysers prior to 1.4.2-SNAPSHOT allow anyone that can connect to the server to forge a LoginPacket with manipulated JWT token allowing impersonation as any user. Version 1.4.2-SNAPSHOT contains a patch for the issue. There are no known workarounds aside from upgrading.	
References	
Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.	
<ul style="list-style-type: none"> • CONFIRM:https://github.com/GeysersMC/Geysers/security/advisories/GHSA-h77f-xxx7-4858 • URL:https://github.com/GeysersMC/Geysers/security/advisories/GHSA-h77f-xxx7-4858 • MISC:https://github.com/GeysersMC/Geysers/commit/b9541505af68ac7b7c093206ac7b1ba88957a5a6 • URL:https://github.com/GeysersMC/Geysers/commit/b9541505af68ac7b7c093206ac7b1ba88957a5a6 • MISC:https://updates.playhive.com/weekend-maintenance-disclosure-2kJMaY • URL:https://updates.playhive.com/weekend-maintenance-disclosure-2kJMaY 	
Assigning CNA	
GitHub (maintainer security advisories)	
Date Record Created	
20210816	Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.
Phase (Legacy)	
Assigned (20210816)	
Votes (Legacy)	

The NVD, which is run by NIST, builds upon the CVE database by adding detailed technical information, including metrics that help assess the severity and impact of a vulnerability. The NVD pulls in every published CVE and adds extra context, such as vulnerability severity ratings, impact scores, and remediation suggestions. This means that all vulnerabilities in NVD come directly from the CVE database, but the NVD adds critical information such as the severity of the vulnerability, its exploitability, its potential impact, and how to mitigate it, providing a more comprehensive picture than the CVE alone.

Each NVD entry contains the following information:

- **CVSS Scores:** The Common Vulnerability Scoring System (CVSS) score, which provides a standardized way to assess the severity of a vulnerability, ranging from 0 (low) to 10

(critical). This helps organizations prioritize which vulnerabilities to address first based on their risk profiles.

- Base Score: Measures the fundamental characteristics of a vulnerability.
- Temporal and Environmental Scores: These adjust the base score based on time-based factors or specific use-case environments.
- Impact Metrics: NVD entries detail the impact of a vulnerability on confidentiality, integrity, and availability.
- Fix Information: NVD provides links to patches, workarounds, or advisories that suggest how to mitigate the vulnerability.
- References: Similar to CVE, NVD includes links to vendor advisories, security blogs, or tools that address the vulnerability.

The following figure shows an example of the database entry for a single NVD entry. This is CVE-2021-39177, the same one shown in the CVE example above. Note that the NVD entry says what it affects, and how to mitigate or fix the issue, in this case by upgrading the application. It also includes a score, which indicates the severity of the vulnerability.

CVE-2021-39177 Detail

Current Description



Geyser is a bridge between Minecraft: Bedrock Edition and Minecraft: Java Edition. Versions of Geyser prior to 1.4.2-SNAPSHOT allow anyone that can connect to the server to forge a LoginPacket with manipulated JWT token allowing impersonation as any user. Version 1.4.2-SNAPSHOT contains a patch for the issue. There are no known workarounds aside from upgrading.

[+View Analysis Description](#)

Severity

CVSS Version 3.x CVSS Version 2.0

CVSS 3.x Severity and Metrics:

	NIST: NVD	Base Score: 9.8 CRITICAL
Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H		
	CNA: GitHub, Inc.	Base Score: 7.4 HIGH
Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N		

Maintaining the CVE and NVD Databases

Another key point about the CVE and NVD databases is how new entries are created. It's important that you see there may some delay, up to several weeks, between a new vulnerability being found and an entry being made for the vulnerability. The process of discovering, reporting, verifying, and analyzing vulnerabilities involves multiple steps and stakeholders. This process can introduce delays between when a vulnerability is discovered and when a fix is issued, but some of the extra steps are used to help prevent false positives from entering the system.

Here's a quick look at the process. The CVE system relies on CVE Numbering Authorities (CNAs), which are organizations authorized to assign CVE IDs. When a vulnerability is reported, the CNA must assess whether the vulnerability meets the criteria for a CVE entry. This requires a thorough analysis of the issue to ensure it is valid, not a duplicate, and that it poses a real security threat. Some vulnerabilities are straightforward, while others are complex and require

in-depth investigation to confirm. The CNAs often work closely with the vendors of the affected software to validate the vulnerability. This coordination can take time, especially if multiple vendors are involved or if the vendor is slow to respond or is engaged in fixing the issue before public disclosure. In some cases, a vulnerability may be held back from publication until a patch is developed or until the vendor and researcher reach an agreement on the details of the vulnerability.

In addition to the time it takes to process a new vulnerability, another factor that causes delays is that there's a large volume of submissions. With the growing number of reported vulnerabilities, it can take time for the limited teams at CNAs or NIST (which manages the NVD) to process each submission. High-priority vulnerabilities may be fast-tracked, while lower-priority ones could face delays.

While both the CVE and NVD databases strive to provide timely information, the complexity of vulnerability verification, analysis, and coordination between stakeholders can introduce delays in publishing entries. The thoroughness of the processes ensures that vulnerabilities are properly validated, assessed, and the methods for fixing the vulnerability have also been tested before being publicly disclosed. For this type of database and information it's more important to be correct than it is to be fast. If the information in the database isn't correct then the entire system will become unreliable, leaving a huge gap in knowledge for cyber security professionals.

CVE and NVD Summary

Both the CVE and NVD databases have been around for many years and contain hundreds of thousands of entries, with new ones being added as new vulnerabilities are found, which means almost constantly. The databases are very complete and contain vulnerabilities for all kinds of devices and Operating Systems including computers, mobile devices, networking devices such as routers, even vehicles. You may not realize it at the moment, but the fact that this data has all been captured and is free for use is nothing short of amazing. Can you imagine how much work it would be if you had to create and maintain your own list of vulnerabilities for

every device you might have to support. It would be impossible. And the fact that these databases are free to use is even more amazing, although in reality they are supported by the US government, so they are paid for using our tax dollars.

You'll take a closer look into these databases in an exercise for this chapter, but for now the important thing to take away from this is that when we say we're checking a device for vulnerabilities we mean that we're checking it against the vulnerabilities stored in these databases. If you want to know more about the CVE and NVD database entries, you can also try the following links. If these links don't work you should try searching on your own, or you can check out the videos I made for you.

<https://www.cvedetails.com/browse-by-date.php> - You can use this if you want to see what a CVE looks like, and how many are issued.

<https://cve.mitre.org/> - This is where you would go to download the entire CVE database, or report new vulnerabilities.

<https://nvd.nist.gov/vuln-metrics/cvss#> - NIST database of CVEs. This page tells you how to read the CVSS scores which range from 1-10.

Vulnerability Scans and Assessments

The CVE (Common Vulnerabilities and Exposures) and NVD (National Vulnerability Database) databases play crucial roles in cybersecurity, but they are not typically accessed directly by cybersecurity specialists during day-to-day operations. Instead, these databases are integrated into vulnerability scanners and other security tools that automatically use their data to identify known vulnerabilities in systems and networks.

When someone says they're doing a vulnerability scan they mean that they're running one of these applications. The application will do what you'd do manually, which is build a list of all the hardware and software on a computer and then check the CVE and NVD databases for any known vulnerabilities. It's also important to note that a vulnerability scan is much more than simply checking that all OS and browser updates are installed. Checking the OS and browser for updates are a good start, but a vulnerability scan also checks for issues with the computer's hardware as well as checking for vulnerabilities in all the installed applications.

While it's theoretically possible to perform a vulnerability scan manually, without using a tool, performing the individual steps manually requires far too much time and effort. Luckily for us, there are programs that automate the process. In this section you'll learn about Nessus and OpenVAS, two of the most popular vulnerability scan/analysis programs, and then walk through the general steps in performing a vulnerability analysis.

Tools

Currently there are several good vulnerability scanning programs available, with two of the most popular being NESSUS or OpenVAS.

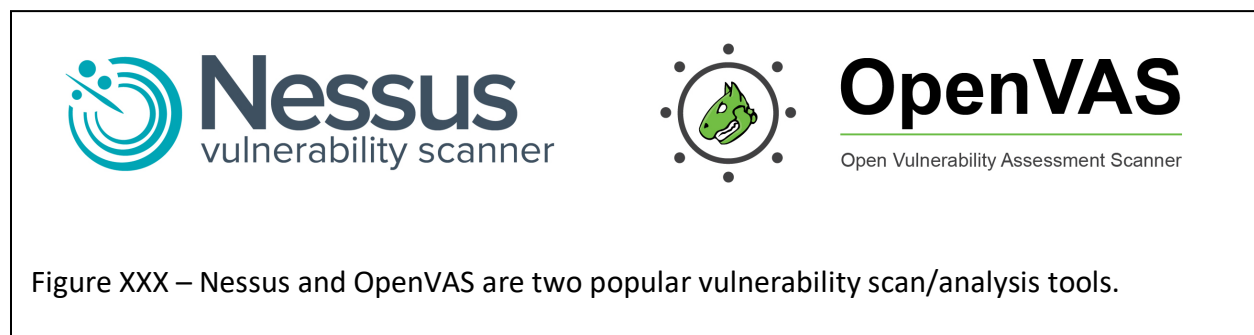


Figure XXX – Nessus and OpenVAS are two popular vulnerability scan/analysis tools.

Nessus is a proprietary vulnerability scanner developed by Tenable. It started as an open-source project but eventually moved to a proprietary commercial model. Nessus is available on a wide range of platforms including Windows, Linux, and the macOS and is widely used for vulnerability assessments, configuration audits, and compliance checks. It scans for known vulnerabilities in operating systems, software, and network devices.

Nessus uses the CVE/NVD databases plus Tenable maintains its own vulnerability database, which is supplemented by Tenable Research. This database includes proprietary plugins, vulnerability checks, and emerging threat information that may not be immediately available in public sources. Tenable updates its vulnerability feed regularly, and users with a subscription receive frequent updates for new plugins and vulnerability signatures ensuring that the latest vulnerabilities are covered.

Nessus has graphical user interface (GUI) which makes it easier to use, although configuring and using it does require some knowledge of cybersecurity, Operating Systems, and networks. It is also highly customizable and includes scanning policies tailored to specific environments or compliance requirements.

As far as pricing goes Nessus has a free, limited version called Nessus Essentials, while the full featured version called Nessus Professional operates under a subscription-based model and costs several thousands of dollars per year.

OpenVAS (Open Vulnerability Assessment System) is an open-source vulnerability scanner that is part of the Greenbone Vulnerability Management (GVM) framework. It provides comprehensive vulnerability scanning capabilities, offering a wide range of features similar to those found in commercial tools like Nessus. OpenVAS is community-driven and designed to be freely accessible to all. Like Nessus, OpenVAS uses the CVE system and pulls vulnerability information from the NVD. In addition, OpenVAS includes its own vulnerability feed known as the Greenbone Vulnerability Database (GVDB). This database is maintained by Greenbone Networks and includes vulnerability checks, detection signatures, and updates. The GVDB is community-driven, but users of the free version can sometimes experience slower updates compared to Nessus.

OpenVAS is primarily a Linux based tool and can't be run directly on Windows systems. Using OpenVAS is considered to be more difficult than using Nessus. Setting it up can be complex, requiring manual installation and configuration of components like the Greenbone Vulnerability Management (GVM) framework, which can be a significant barrier for non-technical users. And

while OpenVAS does have a web-based user interface, it is not as polished or easy to navigate as Nessus. The interface may feel overwhelming to non-technical users, especially when configuring scans or interpreting scan results.

Because OpenVAS is open source, it's freely available for anyone to use, making it highly attractive to individuals or organizations with limited budgets. But being open source also has a downside which is that users must rely primarily on the community for support, which may not be as immediate or comprehensive as Tenable's professional support for Nessus. This community support might not be sufficient for non-technical users looking for timely and specific help. While there are paid support options through Greenbone for enterprises, they are primarily geared toward technical teams.

A third option is a program called Nmap (Network Mapper) which isn't a true vulnerability scanner, but it can be used to perform a quick scan without the complicated setup required by Nessus and OpenVAS. Nmap is primarily designed for finding all the devices connected to a network and checking what network programs each device is running. While Nmap is best known for these network scanning capabilities, it can also be used as a vulnerability scanner thanks to its Nmap Scripting Engine (NSE). The NSE allows contributors to write scripts that Nmap can use to check for different things, like individual vulnerabilities. For example, here are three Nmap scripts and the vulnerabilities they check:

- `http-vuln-cve2017-5638` - Detects Apache Struts vulnerability (CVE-2017-5638)
- `smb-vuln-ms08-067` - Detects a critical SMB vulnerability related to CVE-2008-4250
- `ssl-heartbleed` - Detects Heartbleed vulnerability (CVE-2014-0160)

Nmap doesn't directly use the CVE and NVD databases and there isn't a script that checks for all vulnerabilities. Each vulnerability requires an individual script, and since Nmap is community supported the only vulnerabilities that will be checked will be the ones where some good person has taken the time to create and test the script. This means that Nmap's vulnerability scanning is much less comprehensive than Nessus or OpenVAS as it only checks for a subset of vulnerabilities in the CVE and NVD databases.

Nmap is an open-source tool, which means it's free, and it runs on Windows, Linux, and macOS platforms. Nmap is truly open source, even for support which means it has community-driven support via forums, documentation, and GitHub. While this support is free there is no dedicated commercial support.

Another interesting thing to note about Nmap is that it's a movie star. It's been featured in movies like The Matrix, Snowden, Dredd, Elysium and others as it seems to be the "go to" program used in movies to display hacking.¹

The following table provides a quick comparison between Nessus, OpenVAS, and Nmap.

	Nessus	OpenVAS	Nmap
Cost	~\$4000 per year (2024)	Free	Free
Ease of Use	Easy, with an intuitive GUI	More complex, steeper learning curve	Easy to run, hard to interpret
Vulnerability Database	Extensive, frequently updated	Large, but fewer plugins than Nessus	Small
Platforms	Windows, Linux, macOS	Linux	Windows, Linux, macOS
CVE/NVD Usage	Fully integrated	Fully integrated	Script dependant
Support	Professional, paid support	Community-driven, commercial support available	Community support only
Effectiveness	Very Comprehensive	Advanced	Basic

You don't need to know the details of these tools at this point in your career, but you should know the following:

¹ <https://nmap.org/movies/>

1. Nessus and OpenVAS are used for vulnerability scans and vulnerability analysis. Nmap will check some vulnerabilities, but its main purpose is network detection and scanning, not vulnerability analysis.
2. Neither Nessus nor OpenVAS is ideally suited for non-technical users.
3. Nessus is more expensive and is best suited for enterprises that need a polished, reliable tool with strong support, an easy-to-use interface, and comprehensive vulnerability coverage.
4. OpenVAS is ideal for those looking for a free, open-source solution. It is more challenging to set up and use but offers a powerful set of tools for those who can handle the complexity.
5. Nmap is easiest to install and run, but it will only check for a subset of vulnerabilities.

While NESSUS and OpenVAS are popular and commonly used there are several other programs available. The following sites have list of tools. If these links no longer work, you can find similar sites by searching for something like “vulnerability scan tools”.

<https://www.esecurityplanet.com/networks/vulnerability-scanning-tools/>

<https://www.softwaretestinghelp.com/vulnerability-assessment-tools/>

General Steps

In this section you’ll look at the general steps in installing and running a vulnerability scan application. A lot of this will be the normal download and install process, but there are a couple steps that are specific to vulnerability scan programs. Before you configure and run a scan, I want to remind you that performing certain actions, like running a vulnerability or network scan without proper authorization and permission can be considered criminal acts. If you decide to try this on your own make sure you only do it on your home network or on devices where you have written authorization.

1. Download, and install. This is just like downloading and installing any program.

<https://www.tenable.com/tenable-for-education/nessus-essentials?edu=true>

2. Configure the application. With Nessus and OpenVAS this step is different and more difficult than most programs as they require you to create an account. This is because when you run the program it will connect to a server that will provide the most current list of CVEs and NVDs as well as the most current custom list of vulnerabilities to check. This is done because new vulnerabilities are being found every day and this way you'll always be scanning against the most current list.
3. Configure a scan. This will tell the program things like what device(s) you want to scan, and what types of things you want to check or not check, such as connected USB drives, etc. This typically requires knowing the IP address(es) of the device(s) you want to scan. There are several different ways to discover IP addresses, but here is one of the quickest methods.
 - a. The first is to open a Command Prompt window and type the command ipconfig. To do this, go to the Windows Search box and type CMD, then click the Command Prompt application. When the program starts, type the command:

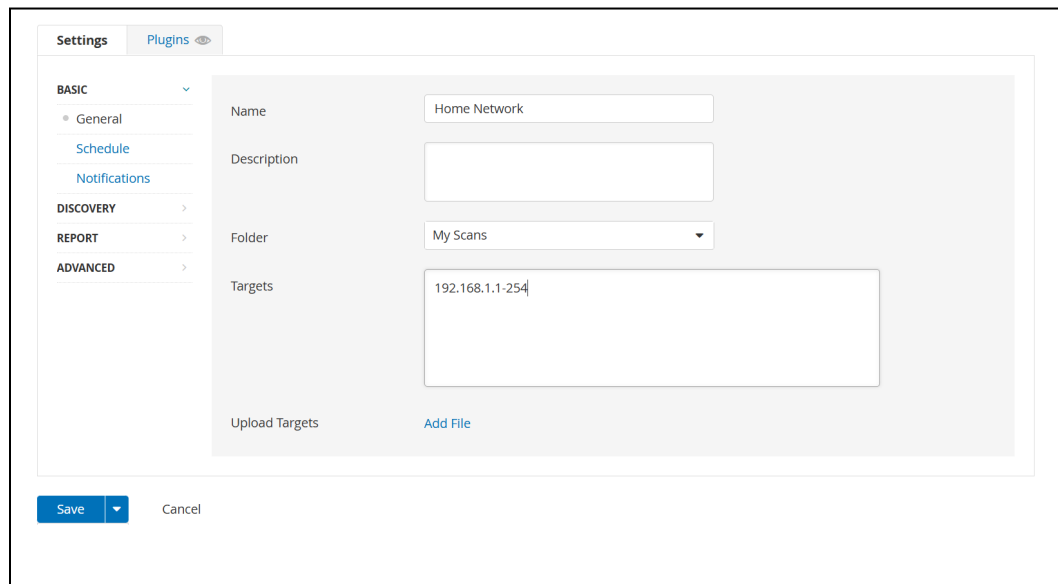
```
ipconfig
```

- b. This command will display the network settings for the device. The setting that shows the device's IP address will be labelled IPv4 Address. In the following figure the IP address is 192.168.1.20.

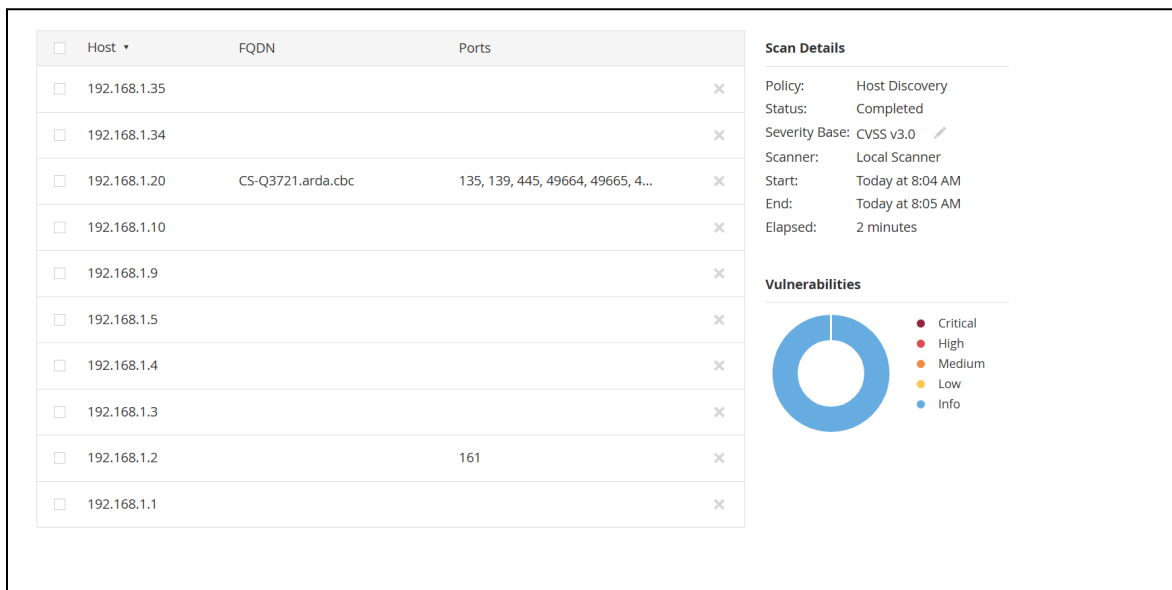
```
C:\Users\btrucks> ipconfig
Wireless LAN adapter Wi-Fi:
    Connection-specific DNS Suffix . . :
    Link-local IPv6 Address . . . . . : fe80::f382:7927:c284:153f%6
    IPv4 Address. . . . . : 192.168.1.20
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
```

- c. To run a vulnerability scan on this device you would use this IP address. If you want to scan other devices on the network, you'll need to know their IP addresses. You can find these by running a network discovery scan, which will show the IP addresses of all the devices connected to the same network as the device running the vulnerability scanner. Most vulnerability scanners have this capability built-in, and in Nessus it's called Host Discovery.

The Host Discovery will need a range of IP addresses to check. To check the entire network you can use the network number followed by host numbers 1-254. In this example, and on *most* home networks, the network number is 192.168.1. To check for all the possible hosts on this network you would ask the scanner to check IP addresses 192.168.1.1-254.



Note that this isn't checking these devices for vulnerabilities, it's just finding out which devices are on a network and what IP addresses they're using. When the host discovery is finished it will display a list of any devices found on the network and their IP addresses. In this example, Nessus found 10 devices. These IP addresses can now be used in the next step.



- Run the scan. The only thing about this step that may be out of the ordinary is how long it can take to complete. Of course, this depends on the speed of the system being checked and the network connection, but typically there are a lot of vulnerabilities to check, and it can take several minutes to finish. When the vulnerability scan runs it uses the data from CVE and NVD to compare the system's hardware, OS software, application software, and system settings against the signatures for known vulnerabilities. If a match is found, indicating that a system is susceptible to a specific CVE, the scanner flags it as a vulnerability and generates a report.
- Interpret the results. A vulnerability scan will produce a list of any vulnerabilities found during the scan. Nessus and OpenVAS will use the information from the NVD database to let you know how critical the vulnerability is, and whether you need to deal with it right away, or if it may not be a big deal. But once again it takes some experience to even understand what the NVD information means. The following figure shows an example of some the information generated by Nessus. As you can see the information in the Description and the Solution is fairly technical.

<p>MEDIUM SMB Signing not required</p> <p>Description Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.</p> <p>Solution Enforce message signing in the host's configuration. On Windows, this is found in the policy setting 'Microsoft network server: Digitally sign communications (always)'. On Samba, the setting is called 'server signing'. See the 'see also' links for further details.</p> <p>See Also http://www.nessus.org/u?df39b8b3 http://technet.microsoft.com/en-us/library/cc731957.aspx http://www.nessus.org/u?74b80723 https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html http://www.nessus.org/u?a3cac4ea</p>	<p>Plugin Details</p> <p>Severity: Medium ID: 57608 Version: 1.20 Type: remote Family: Misc. Published: January 19, 2012 Modified: October 5, 2022</p> <p>Risk Information</p> <p>Risk Factor: Medium CVSS v3.0 Base Score: 5.3 CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/ CVSS v3.0 Temporal Vector: CVSS:3.0/E:U/RL:O/RC:C</p>
---	--

The Nmap reports are even harder to decipher as they include a minimal amount of information which means it will take even more experience to know how to interpret them. Once again this is because Nmap isn't really a vulnerability scanner. The following figure shows the Nmap report for the same vulnerability shown in the Nessus report. As you can see the Nmap report is even more cryptic than the Nessus report.

```
C:\Windows\System32>nmap --script vuln 192.168.1.20
Starting Nmap 7.94 ( https://nmap.org ) at 2024-12-25 09:40 Pacific Standard Time
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_ Hosts are all up (not vulnerable).
Nmap scan report for 192.168.1.20
Host is up (0.00046s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
2701/tcp  open  sms-rcinfo
3389/tcp  open  ms-wbt-server
16992/tcp open  amt-soap-http

Host script results:
|_ smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_ smb-vuln-ms10-054: false
|_ samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

Nmap done: 1 IP address (1 host up) scanned in 70.24 seconds
```

Vulnerability scanners often use the CVSS (Common Vulnerability Scoring System), which is provided by the NVD, to rate the severity of each vulnerability. This score helps prioritize mitigation efforts based on the potential impact. For example, a critical vulnerability with a high CVSS score (9.0–10.0) will be flagged as needing urgent attention, while low-risk vulnerabilities with lower scores may not require immediate action.

In many cases the cybersecurity specialist may need to dig deeper into the specifics of the vulnerability, especially if the automated report doesn't provide enough detail or if this is a new vulnerability. After all, it's impossible for any one person to know how to fix every possible problem. In these cases, the CVE or NVD databases provide great starting points for additional research and direction. The NVD can provide additional insights, such as a detailed descriptions of the attack vector (whether it's remote or requires local access), information on whether there are available patches, and links to advisories or mitigations from the software vendor, and exploitability information, including whether exploits exist in the wild or if this vulnerability is currently being actively targeted by attackers.

The NVD can also be a good resource, or starting point, for determining what actions to take to mitigate the vulnerability. The NVD provides links to external advisories or updates, such as security bulletins from software vendors or open-source communities. This information can help determine the most appropriate course of action (e.g., applying a patch, disabling a vulnerable service, or implementing temporary mitigations).

Using Nessus or OpenVAS to run a scan on a typical home computer might seem like it should be a relatively simple and quick process, but it's actually a little complicated, complicated enough that it's not something a typical home user is going to do. Home users can use Nmap but remember that Nmap isn't really a vulnerability scanner. Instead of running a vulnerability scan the typical home user can do things like check to make sure the OS and browser are updated but checking for all the possible vulnerabilities in the hardware, OS, and applications

requires installing and running a vulnerability scanner like Nessus or OpenVAS and interpreting the results, which is beyond the skill set of the typical home user.

While Nessus and OpenVAS can take some time to install, configure, and learn how to use they can be worth the effort for anyone looking for a career in Cybersecurity or Computer Science. One of the things to keep in mind is that we need to start changing mindsets and stop thinking about taking care of a single computer to taking care of hundreds or even thousands of computers. This is where vulnerability scanners like Nessus and OpenVAS really shine, as they can be used to scan all the computers in an organization. And they can be configured to perform remote scans from a single computer. This means you don't have to install the program on every computer or physically sit at each device and run the scan which can save an incredible amount of time and effort.

Demo of scanning a single computer

If you want to see the complete process for installing a vulnerability scan program, and running the scan on a Windows Home computer using a program called Nessus there's one in the following YouTube video. It's not necessary that you watch this, as you aren't required to do a scan in this class, and you'll get plenty of experience in a later class. But watching the video or even trying it yourself should give you a better idea of what a vulnerability scan is and what it can show you.

<https://www.youtube.com/watch?v=H2ajE4KoqL4>

[Optional] Scanning and Entire Network or Organization

Now that you have a better idea of what a vulnerability scan is, and how to perform it on a single computer, it's time to look at scanning all the devices in an entire organization. You could do this by going to each device in the organization and performing a scan, but this would not only be time consuming, but it would also mean you'd need to know the location of every device. The developers behind the vulnerability scanning programs realized that that this would be a problem and found ways to automate the process. That is, they've designed their

programs so you can first use them to discover every device connected to a network, scan each of those devices, and build a report for each device.

You might think that this is only applicable to a business or large organization, but you can also use it on your home network. You might think the only thing connected to your network is your computer, but it's also possible that your network is being used by your printers, your phone(s) and tablet, game consoles, smart appliances, etc. And, if you haven't set up good security on your home router, it's even possible that your neighbors are leeching off your home network.

I'm going to explain how this works, but this is an optional subject. Understanding the process requires learning some background on TCP/IP, which is advanced enough that you'll spend the better portion of an entire class to learn about it. I'm going to try and explain it in a few paragraphs, but you don't need to know all the details for this class. For this class you just need to know that most programs you'll use for vulnerability scanning are capable of scanning all the devices on a network.

1. Background on TCP/IP
2. Discovering devices – network mapping
3. Discovering open ports on devices – port mapping
4. Configuring the scan
5. Run the scan
6. Generate report and interpret the results

Background on TCP/IP

The first network concept to learn about is how computers communicate on the Internet. This is accomplished by the following:

- The information is placed in something called a packet, which is like a package. You can think of the packet like a snail mail letter. The packet has the address of the computer the package is being sent to (destination), the address of the computer sending the package (source), and the actual information (data) stuffed inside the packet.

- The source and destination addresses are called IP addresses. (IP stands for Internet Protocol, but no one ever spells it out. It's always just IP.) Every computer or device on the Internet has a unique IP address. The IP addresses perform the same general function as a phone number as they uniquely identify a computer. IP addresses consist of 4 numbers separated by a dot or period. For example, 192.168.2.14. And like phone numbers, part of each IP address determines which network the device is on, and part of it identifies the specific device on the network.

Let's illustrate this with the phone number 1-509-547-0511. The leading 1 is the country code for the United States, 509 is the area code for Eastern Washington state, 547 is a prefix used in Pasco Washington, and 0511 identifies the specific phone. The first numbers in the IP address tell us which network the number is associated with, and the last numbers tell us a specific computer on a network. For example, in the IP number 212.69.159.108 the numbers 212.69.159 identify the CBC network, and the number 108 identifies the specific computer on the CBC network. In this case the 212.69.159 is called the network portion of the IP address and the 108 is called the host portion.

- A network packet sent to an IP address will get it to the correct computer.

Ok ... I know that's a large chunk of information to process if it's the first time you've been exposed to it. But the main thing to take away is that network packets sent to an IP address will get the packets to the correct computer on the Internet. (If you know about TCP/IP networking you might know that I'm simplifying how things work a bit. But I want to keep this brief while still providing enough information, so you understand the basics of network vulnerability scans. If you don't know a lot about TCP/IP, don't worry about learning more at this point. You'll learn all about TCP/IP in your networking classes.)

Network Ports and Port Mapping

The next network concept to learn about is how computers determine which program should be given an incoming network packet. This is necessary because a single computer can be running several different programs that could be expecting a delivery of network packets. Sending network packets to an IP address gets the packets to the correct computer, but we need something more to get the packets to the correct program on that computer. For example, you could be running a web browser, a music player like Spotify or Pandora etc., and maybe even hosting your own Minecraft server. And most companies run things like mail servers, database servers, web servers etc.; and they could all be running on the same computer. Network packets sent to a computer's IP address will get those packets to the computer, but now the computer needs to decide which program should receive the packets. This is accomplished by the following:

- Each program on a computer that sends or receives network packets is assigned something called a port number. You can think of the port number as being like a phone extension. That is, large organizations such as CBC use a single phone number, (509) 547-0511, even though there are thousands of phones at the college. Calling the main number will get your phone call to CBC, just like sending a packet to an IP address will get it to a computer. But once your call is received at CBC it has to be routed to the correct extension to get it to the correct person. This is what port numbers do on a computer, provide a means to get the incoming packets to the correct program.
- When a network packet is sent, the port number of the program on the destination computer is included along with destination IP address. This way when the packet arrives at the destination it will be passed to the correct program.
- There are lists of standard port numbers that are almost always used for services like web servers or email servers. When packets are sent to standard services, like a web server or an email server, they will be sent to something called a "well-known" service.

For example, HTTP packets are almost always sent to port 80 and HTTPS packets are almost always sent to port 443. If you set up your own server for something like Minecraft you'll have to assign it a port number, and then let anyone you want to connect to the server know what the port number is. Using the phone number extension analogy, this is like saying there's a set of standard extensions that almost everyone uses. For example, the operator will probably use extension 0, and the HR Department will use extension 666 LOL. We don't actually use these standard extensions for phones, other than 0 for the operator, but hopefully you get the idea.

The next concept to understand is what port-mapping is, and what it accomplishes.

- Port mapping consists of two steps. The first step is building a map of a network, or specifically building a list of all the computers or devices connected to a network. Technically this is called network mapping, and it is done by checking all of the IP addresses on a network to see if that address is currently in use. Going back to the phone analogy, this would be like calling an entire set of phone numbers to see if anyone answers. If someone answers then you know the phone is being used. For example, an attacker might call all 10,000 phone numbers that start with (509) 547. That is, they will start by calling (509) 547-0000, then call (509) 547-0001, (509) 547-0002 ... and end by calling (509) 547-9999.
- Once you have a map showing all of the active IP addresses, the next step is to determine which ports are open at each IP active IP address. This is done by sending packets to different ports and see if there is any response. For example, if you send packets to port 80 and get a response then you know there's a good chance the device at that IP address is running a web server. When this is complete, you will have a map, or list of all the devices on the network, and what network programs each device is running.

- After the port mapping is complete, you can begin the network vulnerability scan. When a vulnerability scan is run it's set to target each computer or device on the network, and check all of the vulnerabilities for that type of computer or device, and for each open port found in the port mapping step. For example, if the port mapping reveals that the computer at an IP address is a Windows based computer running a web server and an email server, the vulnerability scan program will check all the vulnerabilities in Windows web servers and Windows email servers. This process is fairly automated as you can just point the vulnerability analysis program at a computer or device and let it run. When the program is finished it will produce a report that contains a list of open vulnerabilities on the device that was scanned.

- After you've finished running the vulnerability scan on all of your devices you'll have a list of all the open vulnerabilities. At this point you need to do the part of the process that can't be automated as it requires analyzing the findings of the vulnerability scan. The vulnerability assessment or analysis phase requires looking at all list of problems returned by the vulnerability scan and deciding:
 - A. Whether everything on the list really is a problem. The vulnerability scanning programs are very easy to run, but it takes some experience to decipher what the actual vulnerabilities are, what they mean, and whether or not you actually need to deal with each vulnerability. For example, the scan might find problems with a web server on one of your computers, but you know that the computer in question is only being used by other security analysts at the company as a testbed, and that they require the vulnerability to remain open to do their work.

 - B. What to do about each vulnerability. It might be as simple as installing a patch, but it might be as complicated as writing new database code to sanitize queries or writing new program code.

- C. How to prioritize any actions you decide to take. Typically, you want to mitigate the most serious problems first, and then deal with any minor issues. But it takes experience with both cyber security and with the specific business you're working with to decide what makes a problem more or less serious. Going back to the house analogy, you might find an exterior door that doesn't lock. If this door is the front door to your home you should get it fixed immediately. But if this is the door to your garden shed, and all you store in the shed is an old shovel and a rake, then this probably isn't your most pressing issue.

Pen Testing

What is Penetration Testing?

The other threat management process to learn about is penetration testing, or pen testing as it's usually called. Penetration testing is a cybersecurity practice where ethical hackers simulate real-world attacks on a computer system, network, or application to identify vulnerabilities that could be exploited by malicious attackers. Pen testing is meant to provide a more complete picture of a company's security than can be shown with a vulnerability scan as pen testing checks for network and human vulnerabilities as well as hardware and software vulnerabilities. Pen tests typically include a vulnerability analysis, but the pen tests go much further. While a vulnerability assessment will report potential problems, the person running the vulnerability assessment doesn't actually try to break in. In a Pen Test, the security analyst may try to make use of any vulnerabilities and break into a system.

The main goals of pen testing are to identify vulnerabilities and detect security flaws that could be exploited, demonstrate the risk by exploiting vulnerabilities in a controlled manner, and report findings providing recommendations to help the organization fix any issues.

Legal and Ethical Considerations

Hopefully it's obvious that performing a penetration test without authorization is a criminal act. The relevant cybersecurity laws covering this are the Computer Fraud and Abuse Act (CFAA) in the United States, and General Data Protection Regulation (GDPR) in Europe. There have been cases of people who after being caught running pen tests without permission have tried to claim they weren't being malicious and planned to report their findings, but this defense hasn't stood up in court.

Pen testers also need to have clear ethical guidelines regarding what type of tests they will run, what type of information they'll gather before running tests, what type of attacks they'll run, and what they'll do with any information they might discover during the tests. A real attacker may do things like try to damage the data in an organization's database, or blackmail an employee into giving up their login credentials using information they found on social media, or even go as far as kidnapping the family member of one of the company's executives to try and gain access to the company's systems; all things a pen tester would never do because they clearly cross ethical guidelines. But there are also gray areas regarding the types of attacks and information that can be gathered and used, and these should all be agreed upon before any testing begins as part of what are called the rules of engagement.

Pen Testing Methodologies and Frameworks

You should note that there's not a simple all-in-one Pen Test program you can run. The actions pen testers use can go in many different directions depending on the type of testing they're performing, and the organization's structure. Good pen testers will have several different tools in their toolbox, like vulnerability scanners, password crackers, phishing tools, etc. that can be used at different stages of the process. While there's not a single pen testing program, there are frameworks that can be used to generally define and guide a pen tester through the main steps in the process. In this section you'll learn about two frameworks for general pen testing and the general steps they define, and one framework for web application pen testing.

Penetration Testing Frameworks

PTES: The Penetration Testing Execution Standard

The Penetration Testing Execution Standard (PTES) is an open, community-driven framework for penetration testing. It is not formally published by a single organization or governing body but rather developed and maintained collaboratively by industry experts, penetration testers, and cybersecurity professionals. The PTES framework provides best practices and methodologies for conducting penetration tests. It covers the full lifecycle of a penetration test, from pre-engagement interactions to post-testing analysis and is widely adopted by professional security teams due to its real-world focus and in-depth guidance on exploitation techniques.

NIST Penetration Testing Guide:

Another framework is the National Institute of Standards and Technology (NIST) Special Publication 800-115, titled "Technical Guide to Information Security Testing and Assessment." This guide provides a methodology for performing penetration testing and other security assessments in a structured and repeatable manner. The guide outlines different testing techniques such as network scanning, vulnerability scanning, password cracking, and social engineering. The NIST Guide is especially useful for government agencies and organizations that require formal, standardized security assessments and penetration tests to show compliance.

OWASP

There's also one pen testing framework, called the Open Worldwide Application Security Project (OWASP) Testing Guide, which is dedicated to testing web servers and web applications. This is worth knowing about and using because web applications for so many purposes, including interfacing with databases that store our most sensitive personal and financial information. The OWASP guide is a comprehensive manual that provides a framework for testing the security of web applications, offering techniques for detecting vulnerabilities and reducing risks. OWASP also publishes a widely used open-source penetration testing tool called ZAP (Zed Attack Proxy) that helps security testers find vulnerabilities during the development phase.

Hopefully it's obvious, but OWASP is web specific and not a general pen testing framework like the NIST and PTES frameworks. The OWASP pen test framework and tools may be used as part of a larger pen test, or they may be used to just test an organization's web servers and applications.

	PTES	NIST SP 800-115	OWASP
Focus	Comprehensive (network, apps, physical, social engineering)	Compliance with government and industry standards	Web centric - applications and APIs
Methodology	Risk-based, attacker-centric including social engineering	Systematic, compliance-driven	Vulnerability-centric
Phases	Seven, detailed steps	General assessment framework	Application testing methodology
Tools	Any (flexible recommendations)	Any (flexible recommendations)	OWASP ZAP, manual methods
Cost	Free guidelines; tools may cost	Free guidelines, tools may cost	Free tools and resources

Penetration Testing Phases

Pen testing typically follows a structured process, which includes the following phases. Each framework may have different ways of organizing and naming the steps, but they all follow

NIST	PTES
1 Planning	1 Pre-Engagement Interactions
	2 Intelligence Gathering
	3 Threat Modeling
2 Execution	4 Vulnerability Analysis
	5 Exploitation
3 Post-Exploitation	6 Post-Exploitation
4 Reporting	7 Reporting

these same general steps. The following diagram shows the steps, and how they're organized in the PTES and NIST frameworks.

Here's a quick description of what's done in each step of the PTES framework.

1. **Pre-Engagement Interactions** - Decide what will be tested, as well as what won't be tested, agree on deliverables, timelines, and how reports will be delivered, set rules of engagement, and obtain necessary permissions to conduct the test.
2. **Intelligence Gathering** – Gather information about the systems and people to be tested. For systems this should include things like the OS type and version, network addresses, services being run etc. For people this can include things like email addresses and phone numbers, position in the organization, and possibly social media accounts. The rules of engagement will specify the information that can be used as well as what types of information may be out of bounds.
3. **Threat Modeling** – Identify the types of attacks and tests to focus on by assessing the organization's industry, assets, and business model, and identifying critical assets that attackers may target, such as databases or customer information.
4. **Vulnerability Analysis** - Perform vulnerability analysis on the target systems to identify any vulnerabilities that can possibly be exploited.
5. **Exploitation** Test the vulnerabilities identified to determine their real-world impact. This includes hardware and software vulnerabilities identified in the vulnerability analysis step, as well as network vulnerabilities, and human vulnerabilities. Exploit vulnerabilities to gain unauthorized access, escalate privileges, or exfiltrate data. Avoid causing damage to production systems during exploitation.

6. **Post-Exploitation** – Assess the impact of successful exploitation and determine further actions an attacker could take. Test to see if you can move laterally and connect to other systems, escalate privileges, install backdoors or other ways to let you reconnect, check to see if you can change any logs to cover your tracks, and test to see if you can extract any data. Analyze the business implications of any exploited vulnerabilities.
7. **Reporting** – Deliver a detailed report with findings, risks, and remediation recommendations. Provide technical details of vulnerabilities, exploit methods, and evidence of exploitation. Offer actionable insights for remediation and risk mitigation. Include an executive summary for non-technical stakeholders.

Pen testing comes in various forms based on the information available to the person performing the testing. These are called black-box testing where the tester has no prior knowledge of the system, mimicking an attack from an external hacker, white-box testing where the tester has full access to the system's architecture and internal information, focusing on vulnerabilities that may be overlooked during development, and gray-box testing where the tester has limited knowledge, representing an attack by an insider or someone with restricted access.

Summary

Hopefully this all makes sense, but I realize it's a lot to take in. And once again, don't worry about learning all the details or even becoming an expert at running vulnerability scans or pen tests. If you get the BAS in Cybersecurity you'll have entire classes devoted to these subjects.

For now, here are some of the main points you should take away:

1. A vulnerability analysis provides you with a list of vulnerabilities that should be addressed. It can be run against a single device, multiple devices, or all the devices on a network. These are the same vulnerabilities that attackers use, which means you need

to mitigate or fix any problems before an attacker is able to exploit it. Running a scan is relative simple, but interpreting the results and knowing how to mitigate any issues requires more knowledge and experience.

2. Running a vulnerability analysis requires a database of vulnerabilities. Luckily the CVE and NVD databases exist and are available.
3. In addition to just checking for missing updates and patches to hardware and software, a pen test will also check for network and human vulnerabilities. Rather than just supplying a list of vulnerabilities, the pen test tries to attack a device or network by exploiting weaknesses in hardware, software, and possibly with the company's employees. The pen test will also provide a list of problems that need to be mitigated.
4. Vulnerability assessments and pen tests provide you with perspective on the overall security of a device, network, or organization, which allows you to prioritize the work to be done moving forward.

Applying What You've Learned to Your Home System(s)

Background & Basic Security Steps

One of the things I like to do in this class is to take the things you learn about protecting devices and networks for an organization and show you how to apply them to your home devices and network. I think that this will not only help with absorb new terms and processes, but it may also help you improve your own security. With that in mind, in this section of videos you'll learn how to apply what you've learned about using a checklist for security, and the basic process for checking for and mitigating vulnerabilities to your home system(s).

The first thing you can do is use a checklist of basic security items that have been developed by security professionals. There aren't as comprehensive as the cyber security frameworks, but they do contain lists of the basic, fundamental things all cyber security experts agree you can do that will help prevent most malware and attacks, and possibly save you if you fall for a social engineering attack and your computer is encrypted by ransomware. These are considered basic because they're relatively simple to do, and if done, will protect you against a vast majority of security issues. If you're not doing these things, then you're asking for trouble. It would be like walking around town, pulling a wagon full of money, and wearing a sign that says "Rob Me".

The basic things you can do include:

1. Run an Anti-Virus program and keep the definitions up to date.
2. Install updates for the OS and any applications.
3. Create regular backups.
4. Use strong passwords.

Hopefully these seem super basic to you, and I also hope that you're already doing all of them.

Now let's look in more go back to what you just learned about doing a vulnerability scan and using it to check for problems that may need your attention. You can do a vulnerability scan on the devices on your home network, but this can be a little difficult and time consuming, and probably not worth the effort for most home users. Instead, you could just make sure you're your OS and applications are current with the latest patches and updates. Since most vulnerabilities are fixed with a patch, your system should be safe if you've installed the latest updates. And, if you haven't been installing updates there's a good chance that your system will have some vulnerabilities. So, instead of running an actual vulnerability scan, what I suggest is that you ensure that everything on your system is up to date, which is what you're going to learn about in this section. You'll first learn how to check the Windows OS to ensure it's up to date, and then learn how to check your other applications.

Checking Windows Updates

Here's the process for checking Windows to ensure it's current with security patches and updates. If it's not up to date, I also suggest that you configure the OS so that it automatically installs updates when they become available. The process is explained in good detail at the following web site. The web site says Windows 7 in the URL, but it also has information for later versions including Windows 10. If this web page no longer exists If this URL no longer works you can do your own search for something like "How to update Windows".

<https://www.howtogeek.com/howto/5529/how-to-keep-your-new-windows-7-computer-updated-and-secure/>

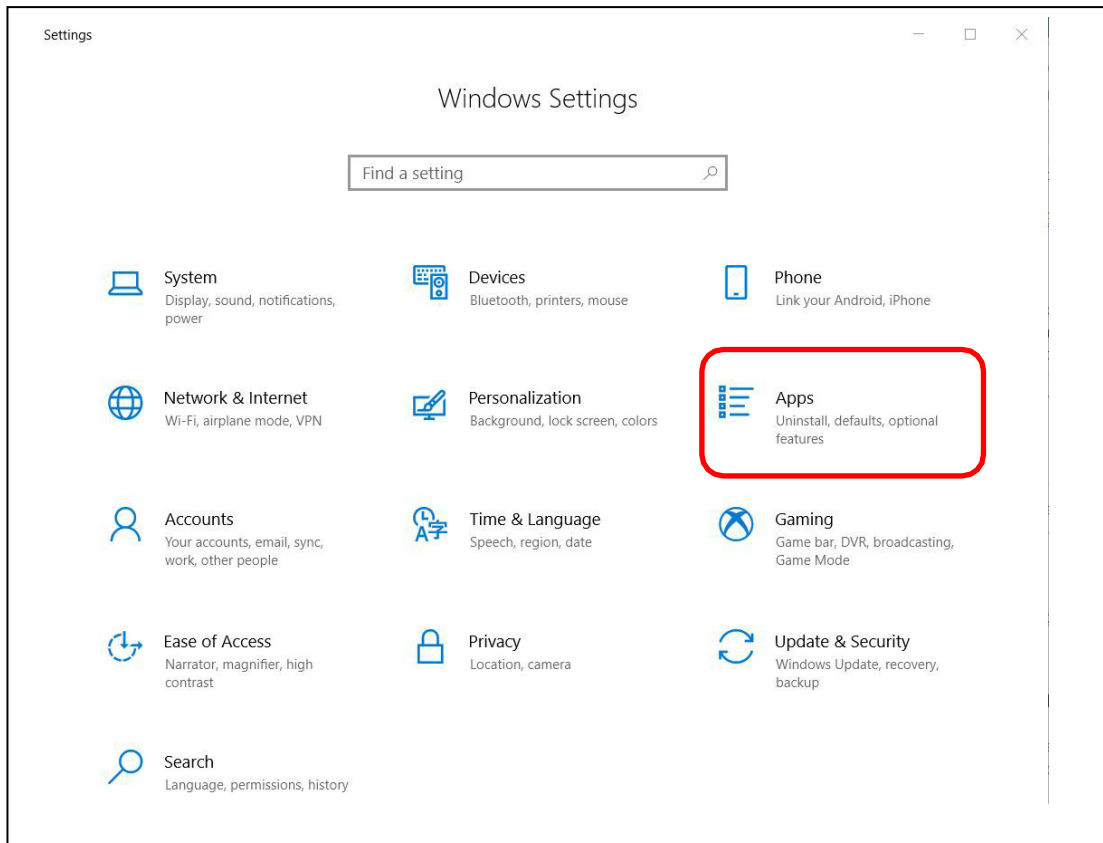
Application Patches and Updates

In addition to keeping the OS up to date, you also need to keep your applications and programs up to date. This is a little harder, as each program has to be updated separately, and they are different ways that the upgrades are performed. Here's a link to a video on checking your applications for updates.

<https://tonysako.com/home/cs150-introduction-to-computer-security/cs150-checking-all-applications-for-updates/> - Ensuring application programs are up to date.

Or you can read the following text. The first thing you should do is take inventory of your applications. You can find a list of the major installed programs on your system by opening **Windows Settings**, and then selecting **Apps**. You should go through your applications and

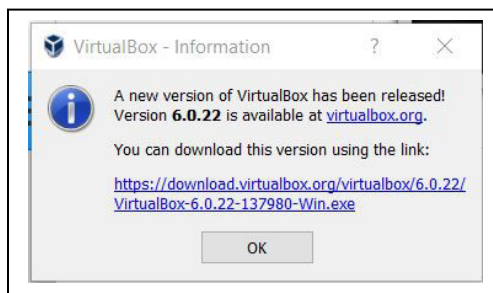
ensure that it's running the latest version. This list will contain a lot of applications, but typically some of them will be part of a suite of applications that can all be updated at one time.



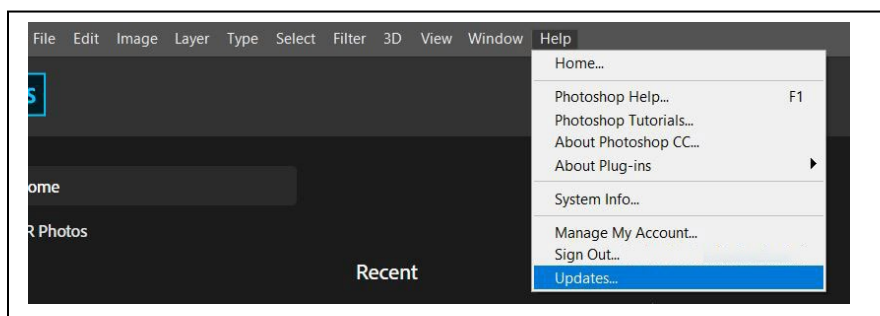
Oh ... there will probably be several programs listed that you don't recognize. You may be tempted to delete them, but be very, very careful before doing this. There are typically several applications installed to run things like the computer's sound or video, or handle the keyboard and mouse, or the touchpad. These often have strange sounding names, and you might think that they're some type of malware since you never installed them. But they are critical for running your system, and if you delete them, you may "break" your computer to the point that you need to completely reinstall Windows. So, if you see a program that you don't recognize, do an Internet search, and determine what it is before deleting the application.

Once you have a list of your applications, the next step is to ensure that you have the latest version of each application or installing any patches. This can be handled in a few different ways including:

- A. Some applications include an “updater”, which is a small program that always runs, and automatically checks for updates for you. For example, if you install most Microsoft, Apple or Adobe applications, you’ll also install a program that checks for updates for you. You can configure these applications so when an update is found they either install it automatically or prompt you to do the install. Some applications are built so they will check for updates each time you start the program. They will either ask if you want to check for updates or check for updates automatically and prompt you if an update is found.



- B. Some applications make you manually check for updates. Many programs have an option for doing this that can be accessed once the program is started. To do this, start the program, then select the **Help** menu, and then look for something like **Updates**. Some applications don't provide any help with upgrades. If that's the case you'll need to figure out what version of the program you're currently running, and then manually go to the applications web site and check to see if a new version is available.



C. A third way is to use an application to check for you. These applications all do what you would do manually, figure out which applications are installed and then see if any of the applications are out of date, but all you need to do is install and start the program. While there are some good options available, I haven't found one I really like yet. The problem as I see it is that there's no way for any single program to track updates and current versions for every other program you might possibly load onto your computer. Most the applications I've tried check against a specific list of the most popular applications, but I've never found one that was able to provide information for all of my installed programs. But in any case, I've made videos that show you how to use a couple of these programs in case you're interested.

Scanning Your Home Network and Ports

Another thing you can do to check your home security is to scan your home network. If you're going to run a full vulnerability scan on your network using Nessus the network and port scan can be done as part of that process. But if you don't to take the time and effort to install and run Nessus you can run a separate network scan to discover the devices connected to your home network. Oh ... and this is another one of the things you don't have to do, but you can do it if you choose. Plus, you'll learn how to do it in the Wireless security class, and by that time you'll have learned more about networking that will make it easier to do and understand. The reason I'm showing you this now is because doing the scans yourself, instead of just reading about them, should help you put what've learned about network and vulnerability scans into context.

If you do scan your network, there hopefully won't be any surprises. But who knows ... if you haven't secured your wireless router, you might find that a neighbor or a stranger connected to your network, sucking up your precious bandwidth.

To do this you'll first learn how to use a program called nmap to perform a port scan. There's a lot you need to know to make sense of what's going on with nmap, so in addition to the Lecture Notes I've made a couple of videos to provide some background and help you. Remember that this is all optional, but if you want to try it yourself, I strongly suggest that you watch the videos on nmap before starting as they provide an explanation of what you'll be doing and how to interpret the results of the port scan.

<https://tonysako.com/home/cs150-introduction-to-computer-security/cs150-network-ports-background-for-nmap/> - nmap – Background on Network Ports

<https://tonysako.com/home/cs150-introduction-to-computer-security/cs150-using-nmap/> - Tips for using Nmap

Scanning Your Home Router for Ports Open to the Internet

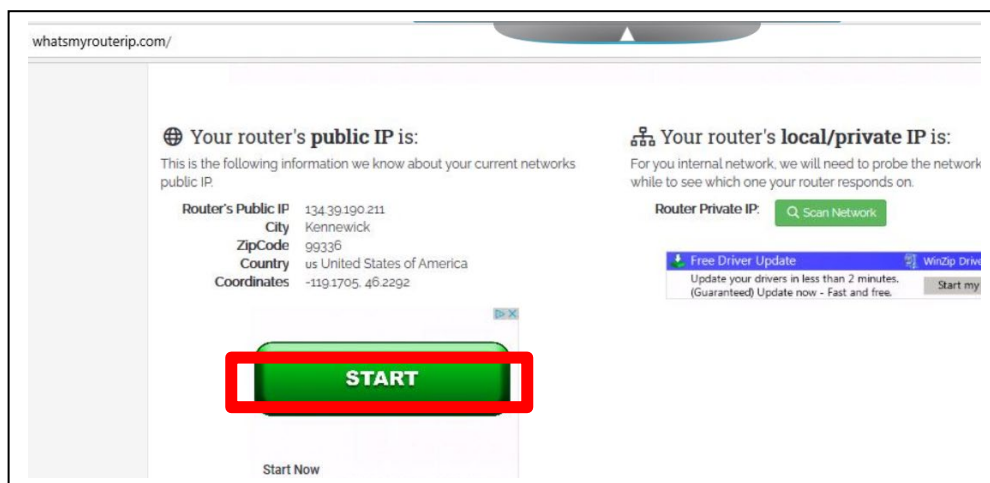
Running nmap on your home network shows you the ports that are open on the side of your router that serves your home network, not the side that's exposed to the Internet. In most cases having open ports and services on the private side of your network isn't an issue as the only devices that will be able to access them must be connected to your home network. So, if you have your home network password protected, you probably don't have to worry about an attacker getting into your router from your home network. But your router is most likely also connected to the larger Internet, and running nmap from inside your home network won't show you what ports are open and exposed to the outside world. If you want to check for open ports and vulnerabilities on the public side of the router that anyone on the Internet can access, you can use the following process. Once again, this is totally optional.

A. The first step in checking your router is to determine the outside IP address. This is typically assigned to the router by your ISP, unless you've paid extra to receive what's called a static IP address. Your IP address can also change periodically, so even if you've checked this before it may be different at this time. In any case, there are a few different ways to find your router's outside IP address including the following. Note that you only need to do one of these, as they all should have the same result.

- The simplest way is to connect to one of the following web sites:

<http://whatsmyrouterip.com/> - this site also has an explanation of the difference between your public or outer facing IP address, and your private or internal network IP address. It also contains other instructions for a few other methods for finding your router's IP in the off chance this doesn't work.

<https://www.whatismyip.com/>



- Connect to your router through the web interface. To do this you'll need to know your router's internal IP address, which is typically 192.168.1.1, but may vary.

- Use one of the manual methods described at the following web page. If this page is no longer available try searching for something like “how to find my router’s ip address”.

<https://www.businessinsider.com/how-to-find-ip-address-of-router>

- B. After you have the Public IP address for your router, you can run a port scan against it. The following links will take you to two sites that will do this scan for you. If these links no longer work you can do a search using a term like “scan home router”.

<http://www.t1shopper.com/tools/port-scan/result/>

<https://pentest-tools.com/network-vulnerability-scanning/network-security-scanner-online-openvas#>

If you use the t1shopper site, it will load what it thinks your router’s IP is for you. Be aware that scanning the ports can take a long time. It really doesn’t take all that long, but they’re slowed the process down for two reasons. The first is that they don’t want attackers to use their site to perform their port mapping. By slowing the process way down t1shopper makes they’re portmapper much less attractive to an attacker who would be better off running their own port mapper. The second reason for the slowdown is that it prevents the site being scanned from going into panic alert mode and blocking all of the traffic from the scanner. Network firewalls and routers can detect when they’re being scanned, and automatically block all of the network traffic from the

suspect computer, which essentially blocks the scan. But if the scanning is done slowly, it's much harder to see the pattern of scans and determine that an attack is underway.

- C. The hope is that the scan will not show any open ports, unless you've opened some yourself. We often open ports on our home router for my sons, when they want to run something like a Minecraft server or Garry's Mod server, and play games with their friends. Or maybe you're running a VPN, which will also require an open port. If you do find some open ports you should do some detective work and see what service is running and why it's open.

Ways to Check Your Comprehension

The test over this chapter won't be for a few weeks, so I suggest that you use the review questions at the end of the chapter to check your comprehension. I don't have the review questions loaded in Canvas, so you'll have to just read them and figure out your answers on your own. Or you could try and connect with some other students in the class and drill each other using these questions.

The material in this section will be included in Test 1, which isn't due for a few weeks. If you look in the Test 1 Canvas Module you'll see it contains a link to a Practice Test. You can take the Practice Test as few or as many times as you want. You're not required to take the practice test, but it's also a good way to check your comprehension and prepare for the "real" test.

The Activities for This Section

There are two sets of activities for this section, the Hands-On Projects and the Case Projects (writing assignments).

Hands-On Projects

The required homework for this section includes two Hands-On Projects from the book, **2-1 Exploring Common Vulnerabilities and Exposures (CVE)** and **2-2 Exploring the National Vulnerability Database**. Make sure and complete both of these assignments. Read the following notes carefully, as they explain exactly what you need to submit to receive credit.

Before submitting your work, add all of the information for your submission into a single document. Make sure that this document has the proper header information (your name, project number, date) and well as the project and step number for each item in the document. That is, if you are submitting a screen shot for Project 2-1, step 16, make sure and add some text that says "Project 2-1 #16", or something to that effect.

What to submit for Project 2-1:

Well, somebody moved your cheese again. That is, the CVE web site has web site has changed significantly since the book was published and the instructions in the book won't work. **You'll need to view the following videos on CVEs in Canvas for help completing the project.**

Viewing the old CVE web site (Video)

Viewing the new CVE web site (Video)

These videos will show you how to access the CVE web site so you can complete the project.

Watch the videos and then do the following:

- Answer the questions for #4, 6, 7, 8, and 11

- Create a screen shot after steps #11 and 13. To get a screen shot use the Windows Snipping Tool or hit the <ALT> + <Print Screen> keys at the same time. The screen shot will then be on your clipboard and you can paste it into your word document. If you need help creating a screen shot there are many videos on Youtube that will provide further instruction and details. Do NOT take a picture of your screen with your camera/phone. If you submit a photo you score will be 0 for the assignment.

What to submit for Project 2-2:

- Answer the questions for #5, 8, 9, and 16
- Create a screen shot after steps #16 and 24.

Case Project (Writing Assignment)

For the writing assignment for this section you need to the following **Case Project** which is located at the end of the chapter. Note that your book makes a distinction between "*Projects*" and "*Case Projects*". For this assignment you want to be working on the *Case Projects*.

Case Project 2-1 False Positives & False Negatives

If you haven't already read the [Guidelines for Writing Assignments](#), now would be a good time. The document explains what you'll need to do for the paper and provides details on how your paper/report will be graded. Also, remember if you need help creating your references the Citation Machine web site will be your new best friend.

When your project is complete, turn it in by opening the Assignment in Canvas, and clicking the **Submit Assignment** button.

Also, remember to check your TurnItIn score. If it is higher than 30% your submission will NOT be graded. You will need to either edit your material and put more of it in your own words, or add more original material. Once you have made your changes, you can resubmit your work. There is no way to check your TurnItIn score prior to submitting your work. This is because your work must be submitted for the TurnItIn program to be able to access it and check it. Don't worry about making multiple submissions, as it has no impact on your grade..

<https://tonysako.com/home/cs150-introduction-to-computer-security/cs150-checking-your-turnitin-score/> - How to check your TurnItIn score

Optional Extra Credit Activities

The following exercises are optional, but if you do choose to do them you can earn up to 10 extra credit points. Note that you can only earn a maximum of 10 points, but there are more than 10 points available. This means that if you complete the Nessus scan, which is worth 10 points, you won't get any additional points for completing any of the other items.

1. [10 Points] Use Nessus to run a vulnerability scan your home computer or the class VM. The steps and instructions for doing this can be found above. Note that this will take some time and effort on your part, and you will be required to create an account at Nessus.

To get the extra credit points you must create the following screenshots and email them to me through Canvas. The subject of the email must be Extra Credit Nessus.

- A screenshot showing the list of vulnerabilities found by Nessus. There may not be any urgent issues, but there should be a long list of things that Nessus checked.
 - A screenshot showing the details of one of the items in the Nessus report.
2. [5 Points] Use nmap to run a scan on your home network and display all of the connected devices. The steps and instructions for doing this can be found above. Note that you must do this on your home network, not the class VM. The class VM network won't show you anything other than the VM itself, so it's pretty boring and not too helpful. To get the extra credit points you must create a screenshot showing the network map or list of devices found by nmap, and email the screenshot to me through Canvas. The subject of the email must be Extra Credit nmap.
 3. [5 Points] Run a scan on your home router and display the ports that are open to the Internet. The steps and instructions for doing this can be found above. Note that you must do this on your home network, not the class VM. To get the extra credit points you must create a screenshot showing the website you used to perform the port scan and a list of any open ports, and email the screenshot to me through Canvas. The subject of the email must be Extra Credit Router Ports.