4 Endpoint and Application Security

In this section you'll learn about the proactive things that are being done to try and find and fix new vulnerabilities before systems are released, and methods that developers and engineers can use when creating new systems to ensure they're secure.

Objectives

At the end of this module, you will be able to:

1. Explain how new threats are found, solutions discovered, and how this information communicated.
2. Explain why secure coding is critical to security.
3. List the challenges involved with writing secure code.
4. Explain the problem of using legacy code.
5. Describe different threat intelligence sources.
6. List the steps for securing an endpoint.
7. Explain the importance of applying updates and patches.
8. Explain how to create and deploy SecDevOps.
9. Compare and contrast popular anti-malware suites for computers and mobile devices.
10. Explain how Virtual Machines can be used for compartmentalization, and how this can be used to increase security.

Section Content

Introduction

All the defense processes you've learned about up to this point have been reactive or ways to patch existing vulnerabilities or react to attacks after they occur. In this section you'll learn about the proactive things that are being done to try and find and fix new vulnerabilities before systems are released, and methods that developers and engineers can use when creating new systems to ensure they're secure. If you're new to computing some of the descriptions may be hard to follow as you need to understand something about programming to understand buffer overflow attacks, or understand databases to understand SQL injection, but try and remember that this is just an introduction, and you won't be expected to have any more than a surface understanding of each attack category. Even if you don't understand the technical tips for writing secure code and building secure systems, you should take away the concept that security should be considered at the very beginning of any project instead of being an afterthought and only being addressed after the system has been built.

You will also learn about the three main tasks associated with securing computers which are ensuring they boot and start properly, securing them from attacks, and hardening or increasing their protection to make them even more secure.

What this section is about – what you should learn

In the previous sections you've learned about the various threats and vulnerabilities that have already been uncovered, and how to check systems to see if they're vulnerable. If we were looking at a timeline all these things would be items that occurred before the current point in time. In this section you'll learn about things that are happening today or should happen today. These include discovering new threats and hopefully finding a way to mitigate them, protecting endpoints from both old and new threats, and trying to prevent new threats by educating

programmers and developers so they make security a priority as they develop new systems, instead of thinking about it after the fact.

Here are the three main subjects you'll learn about in this section:

1. How are new threats found, solutions discovered, and this information communicated
2. How to secure your endpoint devices
3. How to design secure applications

How to design secure applications

I'm going to start with the first subject last, designing secure applications, because this is one of those concepts that frequently gets lost in the constant avalanche of cyber security information, even after you graduate and start a career in cybersecurity. This is a super important concept, because finding a solution for this issue could mitigate almost all the threats you've learned about so far.

To start, let's take a big step back, maybe two big steps, and not concentrate on specific threats and attacks or ways to protect systems. Instead, let's look at the big picture and one of the common sources of almost all the security problems. This source is poorly designed code … Sure, there are lots of problems caused by wetware (people), things like users that choose weak passwords, or people that fall for phishing attacks, but almost every other vulnerability is caused by a flaw in a system's hardware or software. I know this seems super basic, but stick with me and I think you'll see the point.

The two main categories of software on a computer are the OS and the applications. Both can have security issues, but any security issues in either the OS or applications will be caused by bad programming code. When we talk about vulnerabilities in hardware what we're typically looking at are problems with the firmware that controls the hardware. That is, each hardware device like a memory chip, SSD, motherboard, CPU, etc. have some code "burned" into their system, and this code is called firmware. The big point though, is that hardware won't have security issues

because they have a bad piece of copper wire or a bad magnetic spot on a drive, their security issues are caused by programming code that does things in an insecure fashion.

If we summarize this, we end up with these main points:

- The main components of a device are the hardware and the software
- The two main categories of software are the OS and other applications or programs
- The main source of hardware security issues can be traced back to problems in the programming code in the firmware
- The main source of OS and application security issues can be traced back to problems in the programming code

Which means that most computer and endpoint security issues can be traced back to problems in programming code!

Now the questions are (1) why are there so many programs with security issues, can't programmers write code that doesn't cause security issues and (2) what can be done about this problem?

The answer to first question has a few different components including:

1. Perspective and priority.
2. Knowledge required to implement encryption algorithms
3. Ease of use versus security

Perspective and Priority

The perspective issue is that from a developer's or programmer's point of view they're building products that are going to solve a problem, most don't write code thinking of ways it could be used maliciously. For example, when Tim Berners-Lee came up with the idea for the web and wrote the code for the first web server and web client, he was thinking about ways to help

scientists share information. He wasn't thinking about all the ways that his code could be subverted or misused. Or how about the first person to make a wireless router. I'm sure they were thinking it would be great to network computers without pulling a bunch of wires. And they actually did give security some thought and ended up requiring a password to connect to the router. But, security wasn't one of their primary considerations and the encryption algorithm they selected was implemented poorly, making it easy to crack the password to the router. The important point from these examples is that most programmers don't start a project thinking about ways their work could be used for positive purposes. They don't start a project thinking about ways it could be subverted and used for evil. Of course, this doesn't hold true all programmers. For example, the programmers that make password cracking tools or Distributed Denial of Service (DDoS) tools expect their tools to be used to attack other systems.

But the vast majority of programmers don't think about security when they write code, they're thinking about writing code that solves some problem. Solving that problem is their main priority and if they do consider security it's usually an afterthought, or they think that cyber security is another person's job, separate from what they do.

While it's true that most organizations will have a cybersecurity specialist, everyone in the organization will play a part. Just as every employee needs to be trained to use strong passwords and not fall for phishing attacks, programmers also need to be trained to write code that isn't susceptible to attack. This is easier than it may sound, the training part at least, because many of the problems have a known "fix" that can be easily implemented while the program is being written. For example, we know how to prevent buffer overflow attacks, SQL injection attacks, and XSS attacks. All of these can be prevented by simply checking any input provided by a user and throwing out anything that's unacceptable. Buffer overflow attacks can be stopped by checking to make sure that any user input is no longer than a maximum length, and rejecting any unacceptable input. SQL and XSS attacks can be stopped by rejecting any user input to ensure it doesn't contain characters such as **'='** or **<script>** respectively.

I'm going to get up on my soapbox for a minute and say that this emphasis on problem solving

over security shows up almost immediately in programming. If you've taken any programming class, think back on what you learned. You obviously learned how to do things like use variables, if statements, and loops, but did you learn anything about writing secure code? Of course, this isn't quite fair, as there's lots to learn when you start programming. But the point remains that many security issues could be easily avoided if programmers were trained to use code that avoids basic mistakes.

Knowledge required to implement encryption algorithms

Another common issue facing programmers, beyond basic input checking, comes when they implement code to handle securing data or transmissions using encryption. When we're talking about programming and security, this is an area of programming that is of obvious interest and concern as attackers will hone in on this portion of any program, looking for any weakness. Just like locks are used to protect physical items like our vehicles and our homes, cryptology is used to protect everything in the digital world. In the real world, good security relies on choosing the best lock for the job and installing them correctly. You might choose a great lock for the front door of your house, but if you install it backwards it won't provide much protection.

The good news is that there are very secure encryption algorithms available on almost every platform. The bad news is that actually using one of these algorithms requires some in depth knowledge in cryptography. Implementing an encryption algorithm in a program requires setting some parameters to control the algorithm. Good parameter choices will make the algorithm strong, while poor choices will make the encryption easy to break. For example, Elliptic Curve Cryptography (ECC), which can be used to encrypt web traffic uses the formula:

$y2 = x3 + ax + b$

When you implement the ECC formula you need to choose values for **a** and **b** constants in the formula. Selecting "good" values for **a** and **b** will protect any data you encrypt, while selecting "bad" values for **a** and **b** will make it easy for an attacker to crack the encryption and read your data. I actually know quite a bit about math and cryptography, and I can't tell you how to select good values. But I can tell you that there have been implementations of ECC that have used bad values, although in the most famous case the poor values were chosen to purposefully weaken the encryption[1][2]. The solution for this, is to make sure that you have someone who knows what they're doing when they implement any encryption code.

And once again, don't worry about understanding the ECC formula. You'll learn the details about ECC and other encryption formulas in the Cryptology class. What you need to take away from this is that cryptology and encryption are the main way data and systems are protected, so proper use and implementation are critical. Improper implementation will result in a huge vulnerability.

Ease of use versus security

Another factor that makes it harder to design secure code is finding a balance between ease of use and security. For example, code that required passwords with a minimum length of 12 characters and the Capcha that required finding the pictures with a bus in them would be very secure, but not the easiest to use. It would be like building a home with concrete walls, no windows, and one door with 4 locks. This home would be very secure against break-ins, but hard to get in, even for the people living in the home.

We also often see this with smart devices and wireless routers, where every device is configured to use the same default password for the administrator account. This makes it easy for the manufacturer to configure the devices at the factory and print the setup instructions, but makes it very easy for hackers to access a device that still uses the default password. It only takes a

[1] https://www.wired.com/2013/09/nsa-backdoor/
[2] https://eprint.iacr.org/2017/554.pdf

quick web search to find the default passwords for nanny cams[3] or wireless routers[4]

The solution for this is more difficult, as it requires finding and striking the correct balance between ease of use and security, keeping in mind that most people are not tech savvy. That is, you and I might not think it's a big deal to change the default admin password on our home router, but apparently this is a big challenge for many people. If you were the owner of a company that makes a smart device, say a nanny cam, how would you handle this?

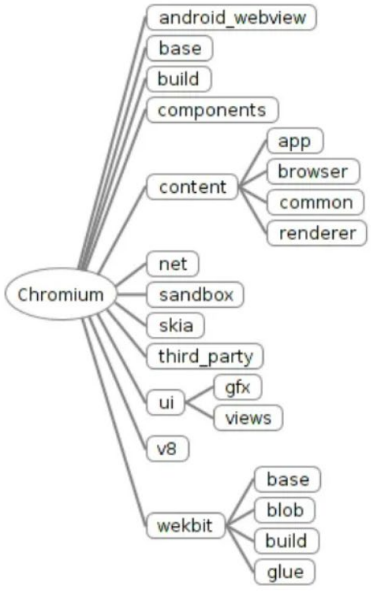
Code Complexity and Legacy Code Use

The last thing I want to talk about regarding designing secure code, which isn't discussed directly in the book, is how complex computer programs can be, and the use of legacy code or code modules. If you've done any programming you know that even a simple task can require quite a few lines of code. And even if you're an experienced programmer it can be tough to look at a given piece of programming code and determine what it does. This can be true even if you were the original author of the code as it may have been several months or even years since you wrote the code. Now assume that you've started a career as a programmer and become part of a team supporting a larger project with hundreds of thousands or even millions of lines of code. For example, say you go to work at Google and work on supporting Chrome which is estimated to have over 40 million lines of code[5]. You'll only be responsible for a small portion of the code, but you'll still need to know how your code interacts with other sections of the Chrome code, and this can take some time to figure out.

The good news is that while it may take some time, you can eventually figure out what each line of code does. The bad news is that some of your code will undoubtably rely on other code that you can't access and don't have control over. This other code exists in the form of code

[3] https://www.ispyconnect.com/userguide-default-passwords.aspx
[4] https://proprivacy.com/router/guides/default-router-login-details
[5] https://www.youtube.com/watch?v=nDtfWq_LNmk&t=298s – trying to calculate the number of lines of code in modern browsers. Kai Hendry

"libraries", or libraries of objects and functions that you can access from your program. As a simple example, say you want to calculate the square root of a number. You can write the code yourself, but more likely you'll use the built-in square root function. As a more complex example, say you want to animate a 3D character on the screen. If you had a few dozen years to spare you could write all of the code to do this yourself, or you could learn how to use a library of 3D code such as Direct3D or OpenGL. In either of these cases the program you ship will be using code developed and maintained by someone else. For example, the following diagram shows a partial listing of the different code libraries used by the Chrome browser.



Project	Description
android_webview	Integration of /content into android.
base	Common code shared between all sub-projects.
build	Build Configurations shared by all projects
cc	chrome Compositor implementations
components	components for Content Module layer
content	Core code for multi-process sandboxed browser
net	Network libraries for Chromium
sandbox	Sandbox for preventing hacked renderer from modified system.
skia	skia Graphics library, some classes in ui/gfx wrap skia
third_party	External libaries such as blink, decoders, compression, ...
ui	• /gfx: Shared graphics lib, base of Chromium UI grp. • /views: frameworks for UI; rendering, layout, event handling. ○ ALSO CHECK! chrome/browser/ui/view
v8	V8 JavaScript Engine
webkit	WebKit glues & some web features

Using other code libraries wouldn't be an issue if all libraries contained secure code. But some of these libraries have been around for years and years, and like the pyramids in Egypt their origins have been lost in the sands of time. Ok, the pyramids are a bad comparison because the code might be old and dusty, but it's not that old. A better analogy would be some medical research study that was run once years ago, and the results of that study have been taken as gospel ever since. For example, most of us have been taught that to lose 1 lb of body weight we need to burn 3500 calories. This is based on a single study done in 1958 by medical researcher Max

Wishnofsky where he burned a pound of fat and calculated the amount of energy it released[6]. The conclusion from this study wasn't correct. Yes, burning one pound of fat outside of the body will generate 3500 calories of energy, but this weight loss in humans is nowhere near this simple and depends on many other factors. The 3500 calories per pound rule isn't even good as an estimate, it's more likely to be 7000 calories per pound[7][8], but this value was accepted without question for almost 50 years, and is still widely accepted and used by dietitians, exercise devices, etc. The fact that it's not correct that you will lose a pound of body weight by burning 3500 calories during exercise or by reducing your caloric intake by 3500 calories isn't a security issue, it's just meant to show that we often use other people's work assuming that it's correct, only to find out later that there was a problem.

Attackers know that this older code is still in use, and on many occasions have found vulnerabilities in code that had been used for years. For example, there was a bug called the Kaminsky Bug[9] found in DNS in 2008, even though DNS had been is use for 20 years. DNS is an essential part of the Internet as it's the system used by the Internet to translate human friendly names like www.columbiabasin.edu to the IP address required to actually open a network connection. When this threat was discovered researchers went into a panic, as it could've been disastrous for the entire Internet. They were able to develop and implement a solution, working in secrecy as long as they could to keep knowledge of the threat away from attackers.

Another more recent example was a set of vulnerabilities found in the code used to play certain audio files. These vulnerabilities could allow an attacker to gain unauthorized access to all the audio and video on an Android phone, so they could listen to your conversations or access your camera at any time. The flaw was in some code in the Apple Lossless Audio Codec (ALAC) audio decoders in the Qualcomm and MediaTek chips that are used in almost every Android device so it affected hundreds of millions of devices. The Apple Lossless code had been used for years and

[6] https://pubmed.ncbi.nlm.nih.gov/13594881/
[7] https://lifehacker.com/you-need-to-burn-7-000-calories-to-lose-a-pound-not-3-1719560948
[8] https://www.nih.gov/news-events/news-releases/nih-body-weight-planner-added-usda-supertracker-food-activity-tool
[9] https://duo.com/blog/the-great-dns-vulnerability-of-2008-by-dan-kaminsky

hadn't been updated since 2011, but the flaw wasn't discovered until 2022.

You don't need to know the details of the Kaminsky Bug or the ALAC bug. Information about these bugs has been presented to provide examples of how the use of legacy code in modern systems can cause security issues.

Summary – Secure Coding

Here are the important points to take away from this discussion about secure coding. There are several things that can cause security issues in programming code:

1. Programmers not knowing how to write secure code. This can be remedied through training.
2. Secure code can make a program or device harder to use.
3. Programs make use of code libraries which may contain their own bugs.

Finding New Threats

Now that you have an idea of how security bugs can find their way into programs, you can see what a great challenge is can be for programmers to write code and create programs without any security bugs. Most of the big bugs will be found during development and testing, before an application is put into production, but there always seem to be some new issue or problem, which leads us back to the first item addressed by the book which is finding new threats. The book does a good job explaining how new threats are discovered and communicated, so the only thing I have to add is to ask you to tie this information back to the lists of vulnerabilities tracked by the CVEs and NVDs. That is, where do you think the CVE and NVD data originated?

How to secure your endpoint devices

The last big thing the book discusses is how to secure your endpoint devices. This is the promised follow-up to the previous chapter where you learned about all the threats and attacks that could occur. There's a lot to cover on this particular subject, and like most subjects in the book, the book does a good job summarizing what it can in a few pages, but there's a lot more to know.

In the old days, back in the late 1990s and early 2000s, the main suggestion was to simply use an anti-virus program to detect any virus or malware while the file was first being loaded onto the computer. The security software would inspect every file being loaded on the computer, whether it was a file download, email attachment, etc. and look for specific patterns of bits indicating the file contained malware. You can think of this like running facial recognition on every employee and customer coming in to the mall. The mall security can use this to prevent known problems from entering the mall. The anti-virus programs were, and still are, able to recognize and catch lots of attacks. But threats have evolved and now there are ways for a virus to evade detection, plus other types of attacks that anti-virus software won't even recognize. This is like someone wearing a mask to get past facial recognition, or an attacker holding everyone in a store for ransom.

The point is that computer and network systems need additional protection to stop things like an insider stealing data or ransomware from encrypting systems. To address all the various threats endpoint security systems have evolved from just by anti-virus programs to suites of programs that use several additional components. Some of the components are code based, automated systems, such as good anti-virus and anti-malware, but some of the components require some live monitoring to look for unusual situations. Using the shopping mall analogy, you might have facial recognition scanners at all entrances, which should stop any known thieves from entering the mall. But to stop even more crime you also have cameras throughout the mall and someone watching the monitors for any unusual activity. The cameras can catch things like an employee leaving a service door open so thieves can slip in, or a gang of thieves robbing the Cinnabon.

Today an endpoint security system will include the following components:[10]

- Machine-learning classification to detect zero-day threats in near real time

[10] From - https://www.crowdstrike.com/cybersecurity-101/endpoint-security/

- Advanced anti-malware and anti-virus protection to protect, detect, and correct malware across multiple endpoint devices and operating systems
- Proactive web security to ensure safe browsing on the web
- Data classification and data loss prevention to prevent data loss and exfiltration
- Integrated firewall to block hostile network attacks
- Email gateway to block phishing and social engineering attempts targeting your employees
- Actionable threat forensics to allow administrators to quickly isolate infections
- Insider threat protection to safeguard against unintentional and malicious actions
- Centralized endpoint management platform to improve visibility and simplify operations
- Endpoint, email and disk encryption to prevent data exfiltration

I'm not going to add a little to the book's descriptions of the different tools and methods, but I am going to give you a different way to organize them or look at them using the following categories:

1. Threats Caused by Known Vulnerabilities
    a. Not caused by users
    b. Caused by users
2. New Threats
3. System Wide Monitoring

Threats Caused by Known Vulnerabilities

As you learned in previous chapters, there are millions of threats caused by known vulnerabilities. These might be threats that are associated with a problem in the OS or a problem in an application, and aren't directly caused by users. This means that they can typically be mitigated with a patch or update.

The other set of threats in this category are those caused by users and their actions. For example, when a user falls for a phishing attack and gives away a password, or when a user opens an email attachment containing malware or ransomware. There's no way to fix these

types of problems with an update or a patch. And even user training isn't effective at fully stopping these threats. Training has been shown to slightly reduce the number of incidents, but users, and people in general, constantly fall for the same tricks. The thing that is typically done to help protect systems from these types of threats is to run anti-malware which watches what users are doing and tries to prevent them from doing something stupid. For example, most anti-malware will filter incoming email and block any messages or attachments containing a known virus. Think of this as someone who sits and watches you as you use your computer, let's call them your personal computer trainer or PCT. Any time the PCT sees you start to do something stupid they will intervene. You try and open an email attachment in a message from UPS-delivery.com about your package being delayed; your PCT knocks the mouse out of your hand. You need a book for one of your classes and start to download a PDF of the book from a website in Eastern Europe that has free copies of college textbooks, and your PCT knocks the mouse out of your hand. You get an email from the IRS instructing you to make a payment immediately or face criminal charges, so you start to enter your credit card information, then your PCT knocks the mouse out of your hand. You get hungry and start to bite into a donut, your PCT knocks the donut out of your hand and hands you an apple.

While the anti-malware has grown more sophisticated, it can't prevent every stupid user action, so another thing most endpoint protection packages include is a backup component. Backups will save you if you fall for ransomware, or if your system gets infected with malware that deletes or alters your files.

New Threats

One of the challenges in cybersecurity is that new threats are constantly being found. The book does a good job describing zero day threats and how they're discovered, but knowing how they're discovered and knowing how to protect against them are two different things. These threats will eventually be added to the CVE and NVD databases, but you may need to protect your systems before then.

In most cases a fix can be found to the problematic program, and an update or patch can be

issued. The advisories about the problem and the patches themselves will be issued with varying levels of urgency. Some threats aren't seen as immediate, so you might be able to take a few days to run the updates. Others may be extremely urgent, and the advisories will suggest you implement that patch immediately. In the worst case scenario, the advisories may suggest turning off your devices or disconnecting them from the Internet until they've been patched.

In any case, regardless of the urgency, the way that newly discovered threats are handled is by installing a patch or update.

If we summarize the different ways to protect endpoints we end up with:

- Existing Threats
    - System with no users – Install patches and updates
    - System with users – Use anti-malware, and backup important data on a regular basis

- New Threats
    - Install updates and patches as advised

The main steps to perform to protect an endpoint aren't anything complicated. You just need to do things that you should be doing on your home computer such as:

1. Install patches and updates
2. Use anti-malware
3. Backup important data on a regular basis

System Wide Monitoring

One of the things that makes protecting endpoints seem more complicated is the discussion of Host Intrusion Detection System (HIDS) Host Intrusion Preventions Systems (HIPS), and Endpoint Detection and Response (EDR) tools. These are just tools that automate the process of checking all the systems on an entire network, or the network itself for out of the ordinary events.

What's an out of ordinary event? There are a couple of methods used to identify events that are out of the ordinary. The first is to set up a system of trip wires or alarms, that will be set off when certain files or programs are accessed. Another is to what ordinary behavior looks like, which provides a baseline that can be used to identify out of the ordinary behaviors. Going back to our shopping mall analogy, we might count the number of people in the mall jewelry store and find that there are typically no more than 10 customers at any time. If we check the jewelry store and see there are 100 customers, it would be out of the ordinary and worth checking. An example for a computer might be something like the amount of data retrieved from a database. On an ordinary day we might see that about 2 GBytes of data are retrieved throughout the day, typically in chunks that are 2MB or smaller. If we see 500GB of data being retrieved, all in 2 transfers, it would be flagged as out of the ordinary, an alarm would be raised to alert someone that this should be checked.

Rather than have human beings sit and watch every computer on a network, or the network itself for out of ordinary events, the HIDS, HIPS, and EDR tools are systems that use agent programs to do the monitoring. The agent programs are deployed to the systems being monitored, and once configured, can report unusual behavior or alarms back to a central server. Like any automation, these programs are great because unlike a human they will never sleep and never look away, and you don't have to pay them.

If you would like a better idea of how one of these programs works you can watch the following video which provides an overview of a system called Splunk. If this link no longer works you can do your own search from something like "Splunk Demonstration".
https://www.youtube.com/watch?v=OgKdLvwSIYY

Artificial Intelligence

As the book points out, many endpoint security systems are using AI or working on adding more AI to replace the human monitoring component with software that looks for situations that are out of the ordinary. But do you think that the AI will ever be as effective as a human? You don't

have to answer that. It's just something for you to think about.

The last thing I'm going to tell you before we move on is a stupid Terminator joke. Hopefully you've seen the Terminator movies. If not, your homework for this week is to watch at least the first 3. When I read about using AI to protect systems I get worried. After all, the biggest threat to any computer system comes from people. Either an outside attacker, or more likely an insider, or an employee that falls for a phishing attack, or visits a fake web site an infects the system with ransomware. So, if the AI is supposed to protect the system, and the biggest threat comes from people … all I have to say is welcome to Skynet. Although we're way past 1997 and Cyberdyne hasn't invented the neural net yet, so you can probably relax for a few more years.

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

As is the case throughout this class, you've learned about a myriad of different items in this section. You'll want to try and get a general sense of what the various items are, and start to learn the terminology. You should also be starting to pick up on the fact that a large part of security is simply applying patches and updates, and running a decent anti-virus or anti-malware application. It's kind of like securing a car. That is, leaving your car unlocked with keys in it makes a car very easy to steal. And on the flip side, doing basic things like locking your doors and not leaving the keys in the car will protect against this risk. There are additional things you could do, like disconnecting the battery every time you park the car, but just doing the basic actions of taking the keys and locking the doors will prevent 99% of the problems.

When we start to apply what you've learned in this chapter to your home security it might feel a bit overwhelming as there's a lot of material in the book. But really, there are only a handful of basic, fundamental things that cyber security experts agree you should be doing to protect your home personal devices. These tasks are relatively simple to perform and if done will help prevent

most malware and attacks, and possibly save you if you fall for a social engineering attack or your computer is encrypted by ransomware. The basic things you should do are:

a. Install patches and updates
b. Use anti-malware
c. Backup important data on a regular basis
d. Use strong passwords

There are plenty of other things you could do, but you if you do these three things you're going to protect your devices from a vast majority of threats. I've said that doing these three things are relatively simple, but I guess I should qualify that statement. They're relatively simple compared to some of the advanced things you could do like using full disk encryption or using virtual machines, but performing these tasks does take a little bit of knowledge and experience and they may be difficult for a novice user or non-technical user to perform. So, when I say these tasks are simple, I'm assuming that you have the basic technical skills required to do something like update software or create a backup. In any case, in this section we'll discuss these three basic steps, plus a few advanced steps.

Updates and Patches

I'm going to say (type) something I'll probably say 8 or 9 times in this class. One of the super basic things you can do to secure your computer is to ensure that all of the patches and updates are up to date, not just for Windows, but for all your applications. Most people are familiar with running Windows updates, but relatively few know if all their programs and applications are up to date, or how to check. If you're sure everything on your system is up to date you can skip the rest of this section. If not, you should view this material and the decide whether you want to take action on your devices.

A. Make sure that OS on your computer is up to date. I also suggest that you configure the OS so that it automatically installs updates when they become available. You may have done this already, if you followed the checklist provided above. If not, here's another web site

that shows you how to do this, but concentrates specifically on Windows Updates. It says Windows 7 in the URL, but it also has information for later versions including Windows 10. If this URL no longer works you can do your own search for something like “How to update Windows”.

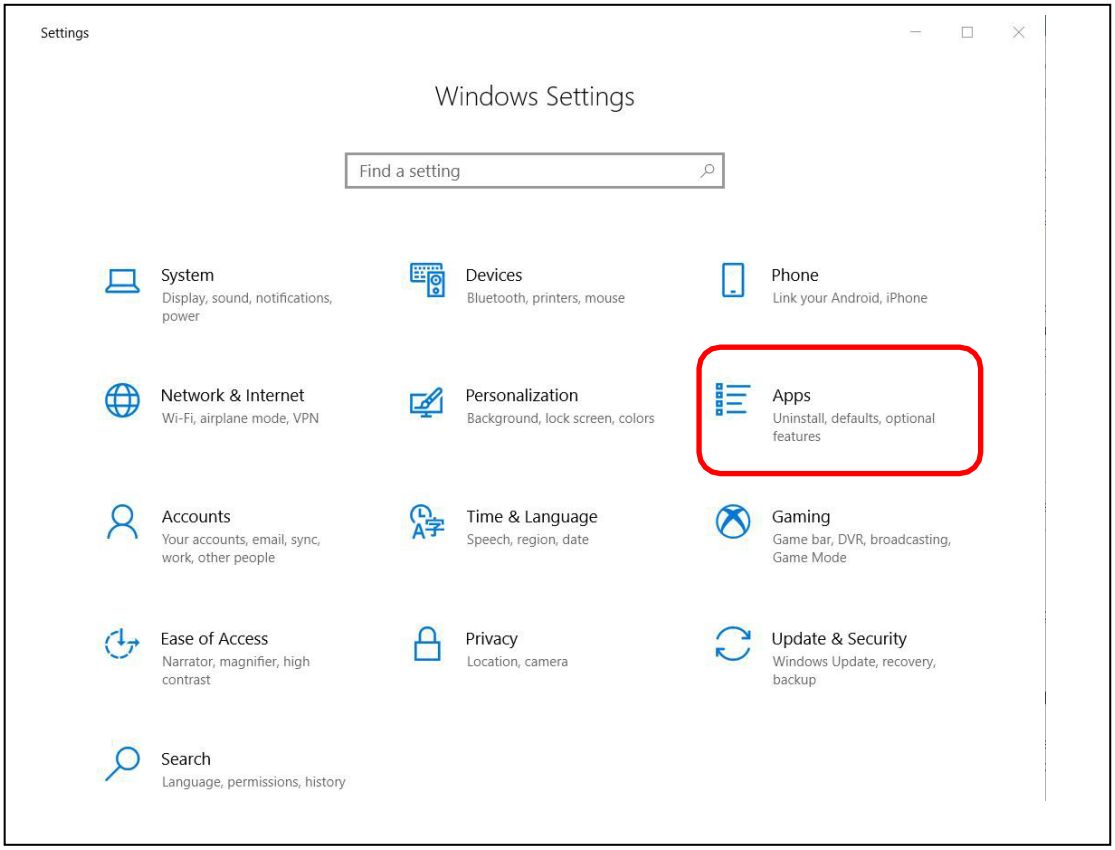
https://www.howtogeek.com/howto/5529/how-to-keep-your-new-windows-7-computer-updated-and-secure/

B. 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. I’ve made a few videos that demonstrate different ways to check your applications to see if you have the latest version or if updates are available.

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, which shows you a manual way to check your programs and applications.

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.

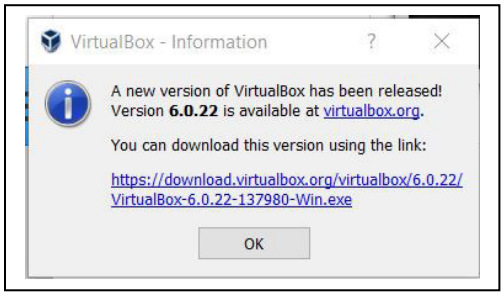


Oh … there will probably be several programs listed that you don't recognize. You may be tempted to delete them, but be very, very careful before doing this. There are typically several applications installed to run things like you sound or video, or handle your keyboard and mouse, or your touchpad. 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.

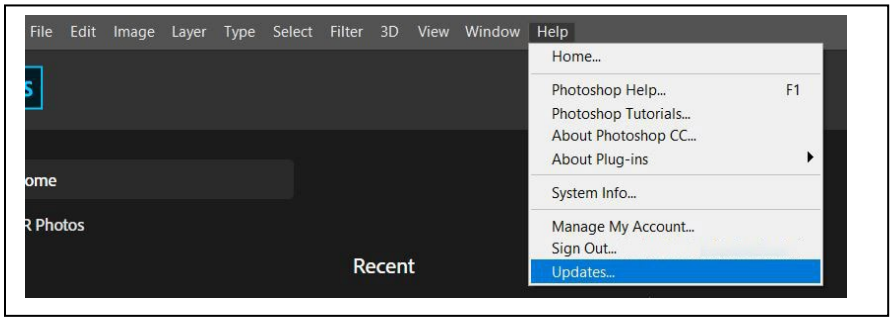
Once you have a list of your applications, the next step is to ensure that you have the latest version of each application, or installing any patches. This is handled in one of three ways:

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. The 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.



Once again, checking your programs and applications for updates is completely optional.

But I suggest you do it for any programs that you use extensively.

Checking Browser Plug-ins for Updates and Patches

In addition to the applications you just checked, there's another set that can be easy to overlook. These are any browser plug-ins or extensions. These apps won't show up on the main list in Applications in Windows Settings, but if you have any ensuring they are up to date is critical to your overall security.

I've made a couple videos explaining what the browser extensions are, and how to check for updates. Note, when we used an older version of the book checking your browser extensions was a class assignment. It's no longer an assignment but the first video will talk about using a site called Qualsys to complete an assignment. You can ignore this information, and even skip the first video.

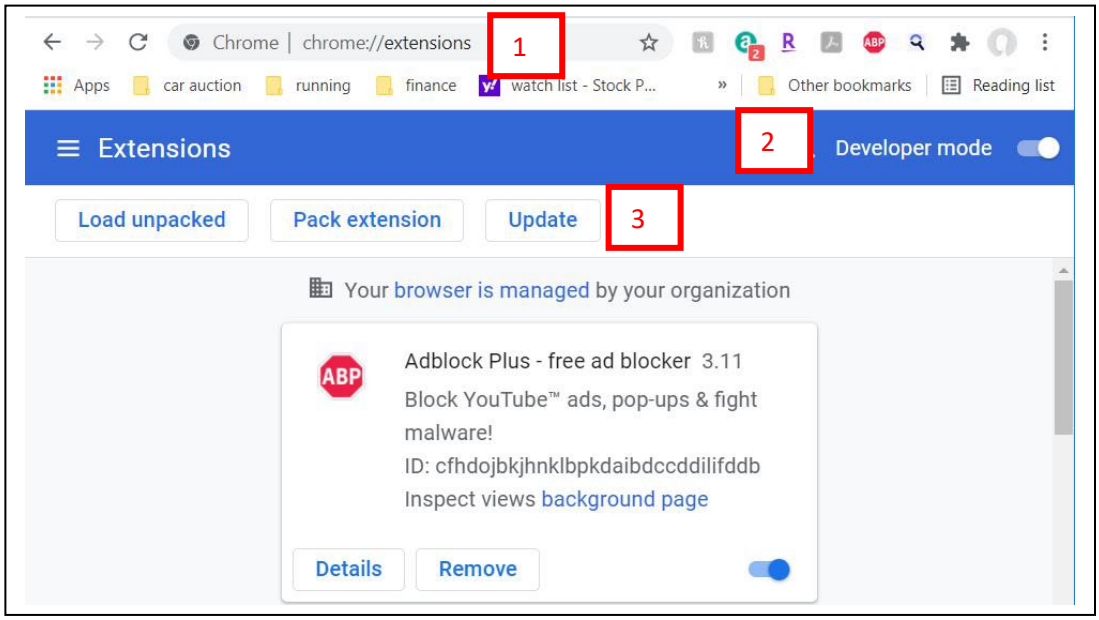
[OPTIONAL] https://tonysako.com/home/cs150-introduction-to-computer-security/cs150-checking-browser-add-on-security/ - This video explains the problem with Hands-On assignment 5-1 in the 5th and 6th editions of the book, and how to deal with it. If you're using the 7th edition or higher you may want to skip this video.

https://tonysako.com/cs150-various-methods-for-checking-browser-plugin-extensions-for-updates/ - Methods for checking your browser's plugins/add-ons to see if they need updating. You're not required to do this for the class, but I suggest that you do something like this to keep your personal devices up to date.

In addition to the methods listed in the video, here's one more method for checking Chrome for updates.

1. Start Chrome and type **chrome://extensions** in the location dialog box
2. Turn **Develop mode** on

3. Click the **Update** button



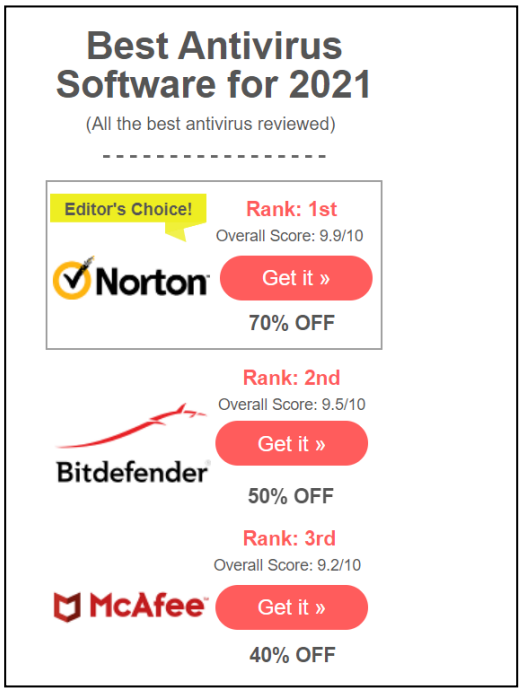
Anti-Virus & Anti-Malware – Home Computers

Hopefully you're running anti-virus and anti-malware on your home computer. If you don't know if you are or not, you need to figure this out right away. If you're running Windows then there's a good chance you're running Microsoft's Windows Defender, which comes with Windows and doesn't cost you anything. Although, you may have been given a free 6 month or 1 year subscription to something like Norton or McAfee, and that free subscription could be expired. In any case, make sure you know if your computer is running some type of anti-malware.

There are also dozens of 3rd party anti-virus/anti-malware applications available, and the best provide more components of endpoint security such as online backups, phishing protection, identity protection, built-in VPNs, password managers, etc. There are plenty of web sites you can use to compare the features and costs of the various anti-virus and endpoint security packages, so I'm not going to go into details here. For example, the following site runs comprehensive tests and publishes the results for free. (If this page no longer exists, try doing your own search for something like "endpoint security comparison" or "anti-virus comparison".)

https://www.av-comparatives.org/tests/malware-protection-test-march-2022/

If you do your own search, practice caution as many of the sites might look legitimate, but are really trying to get you to buy a security package from their site instead of directly from the developer. They'll list one of the packages as the best solution, and offer a "bargain" price, but you may end up downloading a package that itself is infected with malware.



I also want to point out that there's no perfect anti-malware solution. When the packages are tested to see if they can spot known threats, most of them are able to detect and block 98%-99% of the viruses and malware in the test group. But, there's no package that finds everything.

And if you're really new to this, I also want to point out that an endpoint security program may take a good portion of your system resources to run. The application is always running, watching everything you do to try and protect you, which will be a drag on your computer's overall performance. Or in very plain terms, it will make your computer seem slower. This isn't a huge issue on mid to higher end CPUs, but it may cause a noticeable drag on older systems or systems with very low-end CPUs. I'm not saying that this means you shouldn't run an anti-virus or endpoint package, I'm just trying to say that you need to take this into consideration if you're shopping for an endpoint security package, and make sure and compare the performance of each package as well as the features.

The other thing to note is that even though none of the packages are able to block 100% of the threats you will NOT get better protection by running multiple endpoint security packages. Yes, people have tried this. They make the perfectly reasonable assumption that if they run more than

one package, the combination of packages should be able to block 100% of the threats. This may be true (or it may not) but running multiple security packages will cause a noticeable drag on your system. You only need one.

Backups and Ransomware

Now let's talk about backups. Backing up your data may seem like an unnecessary chore, and maybe even a little boring. But … this is the one thing that will allow you to laugh in the face of ransomware. Having backups of your data can't prevent ransomware from encrypting your device, but if this does happen all you need to do is reinstall the OS and restore your data Ok, you might not be laughing if this happens, as restoring your computer can take a few hours. But having your data backed up means you won't need to pay an attacker.

Microsoft has attempted to build some protection against ransomware into Windows[11][12], however I wouldn't rely on it. Plus, even if it does work it only protects against ransomware it won't do you any good if your drive has a hardware error. Hopefully your disks and SSD drives never fail or go bad. I've had many fail, and while it's upsetting and takes some time to buy a new drive and reinstall everything I've always had backups that have made it possible to restore my most precious data, things like family photos and videos, or the books I've written.

My question for you is are you making regular backups of your data? If not, there are several good reasons, in addition to ransomware, to start and to start today. For example, hardware does fail and your drive(s) can go bad at any time, or your device may get lost, or it may get stolen, or it might get damaged. Besides, creating a backup is not expensive or hard to do[13]. You can back up just your important data, or you can make an image of your entire drive, which makes restoring your computer even easier to do. And the prices for backup devices such as external USB drives

[11] https://support.microsoft.com/en-us/windows/protect-your-pc-from-ransomware-08ed68a7-939f-726c-7e84-a72ba92c01c3
[12] https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/controlled-folders?view=o365-worldwide
[13] https://www.pcmag.com/news/the-beginners-guide-to-pc-backup

are ridiculously cheap, or you can back up some of your data to the cloud for free.

The best defense against ransomware is to prevent attacks by taking all the usual precautions and staying away from rogue web sites, not opening email or attachments from people you don't know, etc. But the best way to recover from a ransomware attack is to simply recover any lost data from a backup.

Use strong passwords

Another component of basic security is to use strong passwords. If your only personal device is a desktop computer that's always physically protected inside your home this isn't a huge issue. But if you use a laptop, cell phone, or tablet or other mobile device using strong passwords and 2 factor authentication is extremely important. I'm not going to say any more about what makes a strong password at this point as it will be covered later in this class. You also have a hands-on exercise in this chapter where you learn how to set password rules for Windows computers.

A Complete Computer Security Checklist

While installing updates, creating backups, and using strong passwords are considered basic steps for home devices, there are other advanced things you can do. The following web site has a good list of items that you can do, along with decent explanations. You might take a look at the page in the link and see how many of the items you are already doing.

https://www.tutorialspoint.com/computer_security/computer_security_securing_os.htm

If you decide to do anything on this page, make sure you understand what you're doing. Don't just blindly follow their instructions without a decent understanding of what it is being done. Some of the steps, such an encrypting your entire drive with Bitlocker can cause your data to become inaccessible under some circumstances. That is, you can't interrupt the encryption process once it starts, or all of your data will be lost. And you must remember and provide your Bitlocker password every time you start your system. If you forget the password you won't be able to access any of your data. (You'll learn all about Bitlocker in the Forensics class.)

If you have questions about any of the items in the web page, you can always do more research on your own. Or ask me and I can point you at some additional resources.

Use of Virtual Machines

The last advanced suggestion I have for you is different than using an anti-virus program, backing up your data, and using strong passwords. Those suggestions are things I think you definitely should be doing no matter what. If you're not doing these three things you're just asking for trouble, like trying to drive to Seattle with a car that burns oil and has two nearly flat tires, or heading out for a 50 mile hike through grizzly country in the middle of summer, carrying 10 lbs of raw burger and no water.

This last suggestion is more of a way to enhance your security than something you absolutely should be doing. The suggestion is to use different computers, one for anything that requires sensitive information, like online banking, and another for general web browsing and email, and games if you have the time to play them. Of course, who can afford to buy two computers, take the time and effort required to maintain two computers. I don't know about you, but taking care of one computer is more than enough work for me. So … my real suggestion is to create and use two different virtual machines and use compartmentalization to protect your computer(s). This process is described about the 5:30 mark in the video:

https://tonysako.com/home/virtual-machine-benefits/

When I say that I suggest you do this, I realize that not everyone is going to be able to actually make a VM.

The first obstacle you might face is that creating VMs does require some technical knowledge, mainly about computer hardware. It isn't too technically difficult, but this is not something I would suggest you attempt if you're a new computer user. If you've built your own computer or have your A+ certification you should be able to do this easily. If you're not sure if you have the knowledge and experience, I suggest watching a few YouTube videos on building VMs, and if see if

you understand what they're saying. If so, then you could try it, but if any of it sounds confusing or foreign, I would wait until you get more experience. Here are a couple of links to videos and articles on creating VMs using two different hypervisors, Windows Hyper-V and Oracle's VirtualBox. If these links no longer work, you can find plenty of resources by doing a search for something like "how to create a windows vm".

https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/quick-start/quick-create-virtual-machine - Using Microsoft's Hyper-V

https://www.youtube.com/watch?v=scIDUbbNv6s – Using Oracle VirtualBox

The second obstacle you might face are hardware related. If you want to create your own Windows VMs you'll need to have the Windows install media, your Windows installation code, and a computer that has enough free storage space and a CPU that supports virtualization. You probably don't have the Windows install media, but you can download the ISO file from Microsoft.

Ways to Check Your Comprehension

The test over this chapter, Test 2, won't happen until after you finish three more chapters. If you want check your comprehension you can use the review questions at the end of the chapter or the Practice Test. 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.

If you want to use the Practice Test, look in the Test 2 Canvas Module for the link to a Practice Test 2. 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 required hands-on assignments and the required writing project.

Required Hands-On Projects Homework - Understanding Your Computer's Security Settings

(Note that this exercise is not in the book and you must use the following to complete the assignment.) The goal of this assignment is to familiarize you with some of the basic security measures implemented on your personal computer. In this exercise you will document the anti-virus program in use, identify the firewall protecting your system, and detail the firewall settings. Anti-virus programs and firewalls are essential tools that safeguard your system from malicious attacks, unauthorized access, and other security threats. These controls are key to providing security so it's important that you know how to verify their proper configuration. To complete this assignment, use the following steps. Note that you can do this on your VM or your own computer. If you have a mac and want to do this on your own computer you will have to discover the steps for yourself.

1. Identify Your Anti-Virus Program

    a. Open the Start Menu by clicking the Windows icon in the bottom-left corner of the screen.
    b. Type "Windows Security" in the search bar and press Enter.
    c. In the Windows Security window, click on "Virus & threat protection."
    d. Under the "Virus & threat protection settings" section, look for the name of the anti-virus program protecting your system (e.g., Windows Defender or a third-party anti-virus like Norton, McAfee, etc.) Create a screenshot showing the anti-virus program being used by the system.

2. Identify the Firewall Being Used

a. Return to the Windows Security main window.
b. Click on "Firewall & network protection."
c. Check the current status of the firewall for each network type (Domain network, Private network, Public network). Ensure that the firewall is turned on. If a third-party firewall is in use, its details should appear here. Document the name of the firewall protecting your system.
d. Create a screenshot showing the firewall being used for Public networks

3. Document Your Firewall Settings. Note that the steps provided are for the Windows firewall. If your computer uses a third-party anti-virus or firewall, you may need to refer to the documentation provided by the software vendor to locate detailed settings.

a. In the "Firewall & network protection" window, click on "Advanced settings". (This requires administrative privileges.)
b. In the Windows Defender Firewall with Advanced Security window, create a screenshot showing the following:
   - Inbound Rules: Are there any custom inbound rules configured? What are their purposes?
   - Outbound Rules: Are there any custom outbound rules configured? What do they restrict or allow?
   - Profiles: Confirm whether the firewall settings for Domain, Private, and Public profiles are enabled and configured appropriately.

Required Case Project Homework (Writing Assignment)

The writing assignment for this section requires you to do some research and write a paper. You can select either of the subjects described in Case Project 4-2 or 4-6. You only need to write one paper, but it must be on one of these subjects. All of the papers require you to do some research, so make sure and keep track of the papers or web sites you use for research, and include them as references in your paper.

Hopefully you remember how your paper must be formatted, and the other guidelines for writing papers. But if not, you can refer to the Written Project Guidelines document for details on how your paper/report will be graded. You can find the document at:

https://tonysako.com/writingprojectguidelines2021/