Hacker Highschool

SECURITY AWARENESS FOR TEENS



LESSON 6
HACKING MALWARE

HACKING IS LEARNING
www.hackerhighschool.org

ISECOM

WARNING

The Hacker Highschool Project is a learning tool and as with any learning tool there are dangers. Some lessons if abused may result in physical injury. Some additional dangers may also exist where there is not enough research on possible effects of emanations from particular technologies. Students using these lessons should be supervised yet encouraged to learn, try, and do. However ISECOM cannot accept responsibility for how any information herein is abused.

The following lessons and workbooks are open and publicly available under the following terms and conditions of ISECOM:

All works in the Hacker Highschool Project are provided for non-commercial use with elementary school students, junior high school students, and high school students whether in a public institution, private institution, or a part of home-schooling. These materials may not be reproduced for sale in any form. The provision of any class, course, training, or camp with these materials for which a fee is charged is expressly forbidden without a license including college classes, university classes, trade-school classes, summer or computer camps, and similar. To purchase a license, visit the LICENSE section of the HHS web page at http://www.hackerhighschool.org/licensing.html.

The HHS Project is an open community effort and if you find value in this project we ask that you support us through the purchase of a license, a donation, or sponsorship.

Table of Contents

Contributors

Pete Herzog, ISECOM

Marta Barceló, ISECOM

Kim Truett, ISECOM

Marco Ivaldi, ISECOM

Greg Playle, ISECOM

Bob Monroe, ISECOM

Simon Biles

Rachel Mahncke

Stephan Chenette

Fred Cohen

Monique Castillo

Introduction

It's amazing how easily **malware** exploits systems. Dr. Fred Cohen wrote his PhD dissertation on the idea of a virus back in 1984. It was published in 1985. The university found the dissertation profound and initially ridiculous until Dr. Cohen demonstrated his idea. This was around the time of the Morris worm. As soon as the academics saw the potential of a virus, it scared the hell out of them.

The school was afraid his dissertation might give bad guys some bad ideas. So they yanked the published product later in 1985, but the idea had already been planted and demonstrated even before the Cohen report.

The evolution of these products has led them to be made into weapons, for instance **Stuxnet**. The smallest replicating virus was only 90 lines long (and came out of MIT's Core Wars).

Malware can teach security professionals about social engineering, software exploits, stealth and advancements in technology that show the skills of some serious programmers. Newer forms of malicious programs can be extremely sophisticated and require teams of well-funded programmers to create. Other pieces are simple exploits cooked up in someone's bedroom that sneak past security controls and wreak havoc.

Malware didn't originally earn any money or bring any tangible gain (other than occasional ransomware) for the person who wrote the program. This changed over the years as malware creators learned to take advantage of data theft, using credit card information to get access into the global banking systems with the Zeus virus. Since then, this category has thrived.

Lots of new types of malware attempt to take advantage of you through scams, spam, bot-nets and spying. And it has created a billion dollar a year antivirus industry. Hmm, suppose there's any connection here?

When you dissect a virus, you get to see the inner workings of truly marvelous programs. Polymorphic, what an awesome idea! That's intelligent design, in our book. Why don't we have polymorphic Intrusion Detection Systems? It's tough to understand why malware writers use these wonderful techniques that giant software makers don't. Just like a real virus, we can learn how users think and how this software exploits human behavior to survive (and thrive).

Most Malware (or **mal**icious soft**ware**) is a computer program, or part of a program, that has damaging or unwanted effects on your computer. When people think of malware they think of a **virus**, but the term malware is about much more than just viruses. Our fun online friends have created **worms** and **Trojans**, **rootkits**, **logic bombs**, **spyware** and **botnets**. Malware can be any of these, or it can package several of these together. It's difficult to label malware today as simply a virus, worm or even worm/trojan. And that is why the generic term malware is more appropriate for our discussion.

Are you ready to dive in?

The AV-TEST Institute registers over 180,000,000 malware programs beginning from 1984. They add an additional 20,000 new samples per day. Check it out for yourself at http://www.av-test.org/en/statistics/malware/.

The problem is that we don't know how they categorize malware. For instance, polymorphic malware could look like lots of different virii, or the same one, or not be detectable at all. And intrusion detection systems will see different things than antivirus programs. Take all of these numbers with a big grain of salt.

Viruses (Virii)

This is what most people think of when they think of malware. Computer **viruses** came from computer science studies of artificial life – known then as cellular automata – which gradually became more "life-like," with the ability to propagate (make more of themselves), infect more hosts, become persistent, even hunt and kill each other. They resembled naturally occurring viruses in their behavior and thus the name stuck.

Viruses or **virii** are self-replicating pieces of software that, similar to biological viruses, attach themselves to another program, or, in the case of macro viruses, to another file. The virus is only run when the program is run or the file is opened. That's what makes viruses different from worms. If the program or file isn't accessed in any way, then the virus won't run and won't copy itself further.

Variants of viruses can use different trigger mechanisms like a certain time and date or a keystroke combination. These are usually designed for certain events such as remembrance of uprisings, crimes, acts of war or when the virus author's girlfriend puts out a restraining order against them.

Some malware consists of standalone programs that can appear to be software updates or pictures from someone's visit to the beach. Adobe PDF files have been a frequent launch point for many virus outbreaks, as well as Java. There are lots of reports of pirated software designed to look legitimate but that actually contains malware. That is why you need to look at the software **checksum** before downloading. Yes, **MD5 hash** values can be forged too, but we just want you be to as careful as possible when you download your legal copy of whatever you're getting.

A well-designed virus will evade detection, execute its payload and spread to other machines without the victim ever knowing what has happened until it's too late or maybe never.

Noted authority Dr. Fred Cohen lists some additional nasty ways malware can foul up your system and data:

- randomizing protection settings
- readable files become unreadable
- unreadable files become readable
- writable files become unwritable
- unwritable files become writable
- executables are not
- non-executables are
- setUID (trust level) privileges are set for untrusted programs

- when introduced into a competitor's manufacturing line it can lower the quality of products when the system is controlling a production operation (yikes!)

These days, you'll find that most malware is used as a payload delivery tool. A virus could be used to locate sensitive data in a network, open and keep open a connection for an attack, establish a DDoS, sniff financial information, or disrupt services such as manufacturing and infrastructure. Sophisticated malware will usually have several defense mechanisms, hold multiple exploits, and be written to survive and spread for as long as possible.

The Polymorphic Virus

Once we figured out what a virus was (after 1988), they were easy enough to detect. They had a certain signature to identify them, either within themselves as a method to prevent re-infection, or simply they had a specific structure which it was possible to detect such as a payload. Then along came the **polymorphic** virus, "poly" meaning multiple and "morphic" meaning shape. This new breed of viruses changed themselves each time they replicated, rearranging their code, changing encryption and making themselves look totally different. This morphing created a huge problem for detection, as instantly there were no longer good signatures to detect viruses.

One of the easiest ways to make a virus change itself is by simple encryption. All the virus author has to do is use a random key generator to change the virus and make it unrecognizable every time the virus was copied. The idea made it difficult for **antivirus** (**AV**) vendors to locate a common code string for their signature-based AV software. The source string code was different every time because of the encryption.

The AV vendors decided to look at which parts of a polymorphic virus could not change, which would be the encryption/decryption portions of the malware. As you might expect, the virus writers thought of methods to alter their decryption functions and make those as random as the rest of the program. Rogue programmers added alternating dates, random clocks, different algorithms, operations and all sorts of techniques to make the entire polymorphic virus as undetectable as possible.

Virus creators turned to other methods of hiding malware like breaking the code into several segments. The first virus segment would be a harmless PDF but inside the PDF was a scripting call to perform a download of some more of the virus. The second portion of the virus is encrypted so malware detectors don't notice the install.

The creators think of ways to make a virus look anything but like a virus. Since antivirus programs look for files, events, behavior, or suspicious activities that might be a virus, the polymorphic authors decided to mimic functions of the operating system, peripherals and users.

In some cases, the virus replaces the real system files with their own variations. Sweet: every time you, say, open Notepad, the virus replicates.

The Macro Virus

The **macro viru**s makes use of the built-in ability of a number of programs to execute code. Programs such as Word and Excel have limited, but very powerful, versions of the Visual Basic programming language. This allows for the automation of repetitive tasks and the automatic configuration of specific settings. These macro languages can be exploited to attach viral code to documents, which will automatically copy itself on to other documents and propagate.

Since Word and Excel were built to operate as part of a suite of programs (Microsoft Office), a macro virus could take advantage of those special operating system privileges to spread its payload across entire corporate networks with ease. Office programs are allowed to use special (undocumented) developer calls and scripts within

the operating system for increased productivity, which also gives macro viruses access to protected areas of the operating system and network.

In most email clients, you can preview an attachment without opening the email. This is where a macro virus will attack, since a mini program is opening the attachment. That preview will activate the attachment even if the file is labeled "cutepuppy.jpg." File labels can be spoofed. Go check out **Lesson Nine: Hacking Email** for more information.

You can expect to find macro-type malware wherever there are client-side executable scripts, code, forms or sub-routines. This often occurs with HTML5, Java, Javascript and other add-ons that are part of browsers. You can check out more on this in **Lesson 10: Web Security**.

Exercises

Research these questions:

6.1 What was the first virus? Don't trust the first answer you find. Check multiple sources. Five extra points for each classmate you can prove wrong.

6.2 Now: what was the first virus to be *released into the wild?* How did it propagate?

6.3 The **Klez** virus is well known for **spoofing**. What is spoofing, and how does Klez use it? Assume your computer is infected with Klez. How do you remove it? How do you find out?

6.4 Can a virus be useful or serve a useful function besides being evil? Think of the actual purpose of a virus before you make your decision.

6.5 What was the purpose of the Stuxnet virus? Based on what you read, did the virus achieve its goal(s)?

6.6 You just received an email with the following Subject: "Warning about your email account." The body of the message explains that your inappropriate use of email will result in your losing Internet privileges and that you should see the attachment for details. But you haven't done anything weird with email as far as you know. Are you suspicious? You should be. Research this information and determine what virus is attached to this message. (HINT: When you start thinking of breakfast – you're correct.)

Game On: The Malware Teacher

The technology classroom stank like old fish and maybe rat fur but was at least laid out in orderly rows. Each desk had a sleeping computer monitor resting on it. Fly-encrusted florescent lights flickered against the sunlight from the row of windows. One student in front yawned and the rest of the class caught it, spreading towards the back rows. The teacher, Mr. Tri, hunched over the screen on his desk, scowling.

If the room hadn't smelled so awful, people would have started eating their lunch, chewing gum or chatting while their teacher struggled with basic computer competence. But in the stench they pinched their noses or breathed through a sleeve. Talking, eating and chewing were the last thing anyone in the room wanted to do. Several already had eyes glued to the clock: fifty-two minutes to go.

Eight minutes late, Jace snuck in the room's backdoor as quietly as she could. The smell made her gag, a loud gag. Mr. Tri detected the sudden noise and the change in air pressure from the door opening. He spun around quick enough to see Jace before she could duck underneath a desk. "Jace. You decided to join us today. What a wonderful surprise."

The teen looked around the room and saw eyes telling her to run while she had the chance, make a break for freedom. Run, Jace, run. Save yourself!

She made eye contact with Mr. Tri and replied, "I apologize for my tardiness. I was sick in the bathroom." It was a lame excuse but the teacher was preoccupied with something even lamer. The hacker lowered her backpack and headed for her assigned desk. Pinching her nose, she asked one of her classmates what was going on. He replied, "You'll find out soon enough."

"Jace, since you missed this morning's introduction, I'll bring you up to date. Someone, one of you students, installed a flu or a cold on these computers," He said accusing everyone in the room."Now, until one of you fess up to the crime, you all will sit here enjoying this wonderful odor I brought in," Mr. Tri said. Jace looked around and saw the offending fur coat laced with sardines in the pockets, lying in the middle of the room.

"Ms. Jace do you know anything about this computer illness," he said as he pointed at the screen. Jace stood up and moved towards Mr. Tri as if she were approaching a hungry skunk. When she saw the text on the monitor and moved quickly to the keyboard and typed a few commands to see how the machine would respond.

"Hmm," she said to the screen. In a single motion, Jace reached into her backpack, pulled out her eyeglass case, opened it, and selected a USB thumb drive, thinking *gunslinger*. Her right hand was already typing while the left plugged in the drive. "Has anyone been updating the software on these computers?" The long silence had her imagining Mr. Tri's expression.

"What do you mean *update*?" he asked.

"Never mind."

First off, the browser was two versions old, and the Soda HTML extensions were famously exploitable by zero day vulnerabilities. Next, she saw another HTML5 product called Teepee that was two years old, at least. Clicking through folders of tools on the USB drive, Jace found and started Nmap to see what ports the computer was listening on. Nmap came back with a long list of open ports. Jace frowned.

Wireshark showed tons of TCP/IP traffic coming and going from five ports on the

machine. To fix that, she simply unplugged the network connection. The five ports continued to send SYN packets, even without a network connection. She turned her attention to the operating system's start-up files.

Jace wasn't aware that she said "hmmm" all the time when she was examining a computer, but the rest of the class noticed it. They began to gather around her in a semi-circle of curiosity. "Hmmm," Jace said, spotting several unusual programs in the boot loading process and system start-up.

She scanned through each file directory looking for unusual folders and their files' last-access dates. Again, this brought up a list of highly suspicious programs, directories and files.

Every bit of the bad file stuff was under one user name account name. Jace slapped a hand over her mouth just in time, then slowly turned to her right, lowered her hand and said softly to him, "Mr. Tri, it looks like the one who downloaded all this malware is a user named Super Tri." It wasn't until the laughter started that she looked over her shoulder and saw the whole class gathered behind her.

Oops.

Game Over

Worms

Worms are similar to viruses in that they propagate, but they use network services to move around. But they don't rely on someone running or accessing a file to trigger the self-replicating code; it executes on its own as soon as it can find a vulnerable host.

So a worm is a standalone program that, after it has been started, replicates without any need for human intervention. It will move from host to host, taking advantage of an unprotected network or service. Worms have overloaded severs and entire networks because of they're all about multiplying. Depending on how the worm was designed, a worm may not have a specific end point or target.

Worms have been used to map networks, to dig into hidden areas and to report their findings at predetermined connection points. This type of malware can be autonomous or work within a command and control structure.

There are several instances of worms in major systems in which nobody has been able to remove them, determine their purpose or keep taps on their location. Worms are excellent for reconnaissance because they don't normally have a payload and use covert channels for communication if the communicate at all. If the worm never communicates, it's impossible to tell where it is and what it is supposed to do.

The good news for worms is that they usually only infect a system once. The bad news is about trying to locate that infection to remove the worm: Good luck.

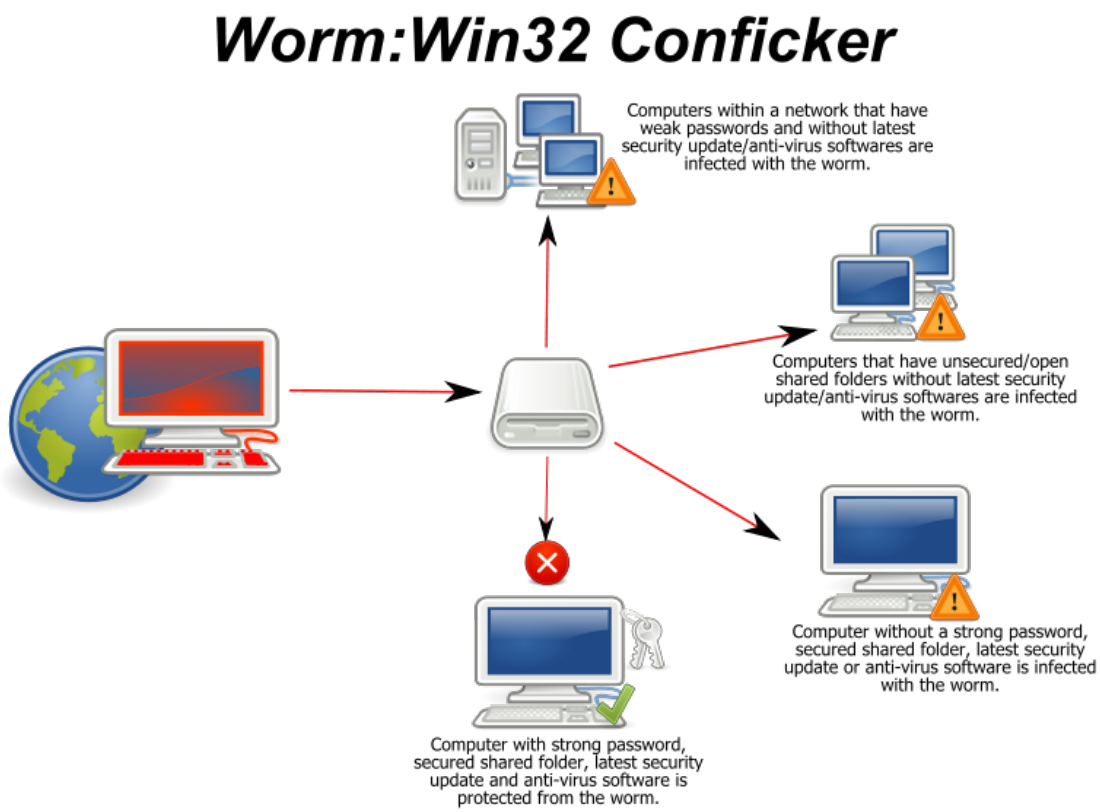

Figure 1. How the **Conflicker** Worm Spread. Photo courtesy of Wikipedia under Creative Commons. http://en.wikipedia.org/wiki/File:Conficker.svg

Exercises

6.7 Which OSs were vulnerable to the first worm to spread over the Internet? Find the source code. Yes, really. No, it will not set your computer on fire. You can open the files in your browser and see how much fun the author was having. Curse him or envy him, depending.

6.8 Search for videos that show you how to use a worm hack tool. (Hint: use exactly that last phrase.) DO NOT CLICK ON THE LINKS. Considering that worms propagate through infected media like videos, why wouldn't you necessarily trust those videos? Or even those links?

6.9 How can you make a good decision to trust or not to trust these videos? For a start, do an Internet search and find the article, "Ten Tricks to Make Anyone Trust You (Temporarily)." Are any of these tricks being used on you?

Trojans and Spyware

The bread and butter of malware falls into the category of spam. This is **trojan** and **spyware** with a touch of **adware** for an extra kick. The original **Trojan Horse** was created by the Greeks several thousand years ago. (Think about the film "Troy" if you've seen it). The basic concept is that you offer something that appears useful or benign to sneak something nasty into an otherwise secure computer. Examples include: game trailers; e-mail promising naked pictures of your favorite celebrity; a program, tool or utility; a file, such as a PDF; or pirated videos. You will often find them loaded into so-called **freeware** games. The concept of freeware is not to fill a free product with advertising or junk, but somehow that idea got mixed up.

Trojans are pieces of malware which masquerade as something either useful or tasty in order to get you to run them. There are at least two types of trojans. The first breed of

trojan malware pretends to be a useful program, picture, music, movie, or is an attachment within a program. The second type is a fake program that replaces the legitimate one on your system. Once they are inside a system they may do something unpleasant to your computer such as install a backdoor or rootkit, or – even worse – turn you machine into a zombie. Cue scary music in the background.

Your first clue that a trojan has been installed on your computer might be a massive slowdown and loss of resources. Your computer that is, not you. If you have a massive slowdown of your body and/or loss of resources then you have the flu. Go get a shot from your doc. Your computer is not so easy to fix. You are going to need your strength.

You might notice some applications won't load, or programs load that shouldn't be running at all. Don't expect your antivirus software to help because it didn't stop that trojan from installing; why should it work now?

Nope, this is all your mess and if you have one trojan on your computer, expect to find more. Since trojans are just vessels for dumping junk on your computer, you will need to figure out where the trojan came from. What was the last thing you downloaded, opened up or viewed from a friend?

To be fair to your friend, even very large organizations have given their employees, customers, and each other trojans before. Sony did it. Valve did it, twice. Microsoft may have done it but they keep calling them "undocumented features."

We are going to talk about destroying malware later.

Rootkits and Backdoors

Often when a computer has been compromised by an attacker, they'll want to get back into the machine. There are many variations on this, some of which have become quite famous – have a look on the Internet for "**Back Orifice**."

Rootkits and backdoors are pieces of malware that create methods to keep access to a machine or network. They could range from the simple (a program listening on a port) to the very complex (programs which will hide processes in memory, modify log files, and listen to a port). Both the Sobig and MyDoom viruses install backdoors as part of their payload.

Both hardware and software manufactures have been accused of installing backdoors in products. Some of this is state-sponsored hacking, while some is just nosy companies. Sony installed spyware on users devices to enforce **Digital Rights Management** (**DRM**). China has been charged with installing secret code in routers, hubs, and other products built in their country. These tactics have destroyed consumer trust in brands and products made in certain countries.

When you are dealing with rootkits, expect to lose your **master boot record** (**MBR**), the *software* that boots your OS. Rootkits need to load themselves into memory before the operating system. They do this by hiding portions of themselves in the MBR. This means that successful removal of the rootkit will also damage your MBR.

To fix the MBR you will need to get to the command prompt through system recovery. At the command prompt, enter the following command and then hit enter:

```
bootrec.exe /FixMbr
```

If successful, you should be greeted with the message "**The operation completed successfully.** The Master Boot Record has been repaired."

While the above command does fix the MBR, there still might be an error with the system partition's **boot sector** and **Boot Configuration Data** (**BCD**), which is to say there might be a *physical* issue to fix. This can happen if you install another operating system

alongside Windows 7, such as Windows XP. To write a new boot sector, try the following command:

```
bootrec.exe /FixBoot
```

Exercises

6.10 Research Back Orifice. What exactly does it do? Who created it?

6.11 Research Windows Remote Desktop. What exactly does it do? Compare it to Back Orifice: how are they different?

6.12 Let's say you have a computer you want to make your computer dual-bootable, able to start two different versions of Windows. There's a trick to it, though (as usual). In what order must you install the Windows versions for this to work?

Logic Bombs and Time Bombs

Logic bombs and time bombs are malware programs that sit quietly until some condition is met – perhaps a particular bit of data or a certain date. They do not normally propagate. For example: a program could be created that, should the administrator fail to log in for more than three weeks, would start to delete random bits of data from disks.

This occurred in a well-known case involving a programmer at a company called General Dynamics in 1992. He created a logic bomb that would delete critical data, set to be activated after he was gone. He expected that the company would then pay him significant amounts to come back and fix the problem. However, another programmer found the logic bomb before it went off, and the malicious programmer was convicted of the crime and fined $5,000 US dollars. The judge was merciful – the charges the man faced in court carried fines of up to $500,000 US dollars, plus jail time.

In 2009, a terminated employee at the mortgage loan giant Fannie May placed a logic bomb that was set to wipe out their 4,000 servers. Luckily, the malware was spotted before it activated. It wasn't such a lucky break for the ex-employee, though.

A logic bomb/time bomb is usually an insider attack committed by a disgruntled employee, contractor, or fired employee with access to the network. Prevention is the best method to stop this kind of threat. Enforce separation of duty so that no person has too much power over a system. Make sure every employee takes a vacation each year so that an evil doer can't keep covering their tracks.

And best of all, if your company fires somebody, take way their network access immediately. Don't let the employee finish up the day or check a few emails. Get them out of the office building, right after you take away their building access keys (codes). Place their network account in a special folder but remove all privileges for user access, especially remote access. That ought to help out some (as long as they don't know other employee's passwords).

Exercises

6.13 What reasonable (and legal) uses might there be for time bomb and logic bomb coding?

6.14 How can you detect such a program on your system?

Malware Today

Malware usually gives the attacker access to files or data on your computer, network, tablet or smartphone. Yes, your mobile phone can get malware too. **No computer system is immune from malware – including all personal electronics.**

Your mobile phone, or smartphone, is just a physically small computer. If you are surfing the web, using Facebook or opening email attachments, then you are vulnerable to malware on your phone. Malware may even come preinstalled. The issues are the same as with your computer; for example, you risk having your passwords hacked. More likely, the malware will wait for you to do some online banking and either clean out your bank account or steal your banking credentials and send them to the attacker.

Internet TV is also here. Now you can watch television and surf the Internet all at the same time. You can connect things up and have a "smart" home. Again, you'll have the same issues as you have on your computer. Researchers have hacked into Internet enabled TVs, onboard computers on cars and even refrigerators. Pretty much anything with an onboard computer can be attacked. Be aware that criminals can infiltrate your home, a private space where you feel secure, through your online interactions.

You may think you have nothing of value on your computer or smartphone, but your identity can be exploited. That is, an attacker could take information about you from your computer or phone together with publicly available information about you, say your Facebook photo, and there may be enough information to build a detailed profile. The attacker could try and open credit cards, or take out bank loans, in your name. This is known as **identity theft**. Creditors will then expect you to pay them back for the things the *attacker* bought. It can take years to prove you didn't spend the money and to clear your good name. It could delay you getting a loan to buy that "fast and furious" car you've been dreaming of.

We are digitally connected almost 24 hours a day and we expect our devices to remain part of the Internet even when we are not using them. Malware creators like that. Our phones are synched to our tablets which are synched to our computers which are synched to our cloud accounts. All of this information is at our finger tips and we want access to our music, files, movies, and personal data everywhere we go. Malware creators like that, too.

Currently, a lot of malware targets mobile devices. These devices have the least amount of security yet have the same accessibility to your data as a computer. On your computer, you probably have a firewall, antivirus software, and anti-spyware software installed. Your mobile devices probably don't have any of these protective measures. That needs to change.

Malware creators may be changing tactics away from ransom and denial of service attacks towards complete destruction of an organization's network data. Sony was attacked in October 2014 in a multipronged effort to release incriminating evidence, while destroying vital data in the background. This cyber assault used sophisticated malware against Sony to disrupt daily operations and render critical data useless.

Feed Your Head: Malware Flavors

According to Kaspersky in 2013 the top 20 malicious programs were:

1.	Malicious URL	93.01%
2.	Trojan.Script.Generic	3.37%
3.	AdWare.Win32.MegaSearch.am	0.91%
4.	Trojan.Script.Iframer	0.88%
5.	Exploit.Script.Blocker	0.49%
6.	Trojan.Win32.Generic	0.28%
7.	Trojan-Downloader.Script.Generic	0.22%
8.	Trojan-Downloader.Win32.Generic	0.10%
9.	Hoax.SWF.FakeAntivirus.i	0.09%
10.	Exploit.Java.Generic	0.08%
11.	Exploit.Script.Blocker.u	0.08%
12.	Exploit.Script.Generic	0.07%
13.	Trojan.JS.Iframe.aeq	0.06%
14.	Packed.Multi.MultiPacked.gen	0.05%
15.	AdWare.Win32.Agent.aece	0.04%
16.	WebToolbar.Win32.MyWebSearch.rh	0.04%
17.	AdWare.Win32.Agent.aeph	0.03%
18.	Hoax.HTML.FraudLoad.i	0.02%
19.	AdWare.Win32.Ibryte.heur	0.02%
20.	Trojan-Downloader.HTML.Iframe.ahs	0.02%

Exercises

6.15 Know the latest malware threats. That is, what new malware threats have emerged today? Go to the website of an antivirus software company and search for their threat monitor. Do an Internet search on "threat research and response."

6.16 Are there any threats that affect social networking sites? Look at different antivirus websites. Do they agree on what the latest threat is? How often do malware threats change (how many new ones are released each day)? How often should you update your anti-malware?

6.17 What are the issues relating to malware that could arise when you bring your own device (BYOD), say a laptop or smartphone, and connect it to a network at a friend's house, or at your work? What about coffee houses or restaurants?

Attackers have various motivations, but for the modern malware writer it is usually financial gain: to rob people of their money. They don't have to break into your house any more. They may empty your bank account, or spend big in your name. The other way that malware makes money is using your computer to distribute spam or phishing emails. Attackers can make a lot of money this way. That is, until your ISP blocks you.

In an even more ironic turn, crackers have come out in the open, offering malware as a service. An easy search will find you a botnet for rent or a cracker for hire to write

custom malware. Could you trust a malware writer? Can they leave themselves some sort of backdoor into *your* computer?

Mobile Malware

Attackers used to focus on the network, but now they can easily circumvent the network defenses by targeting business mobile devices instead. In these next exercises, we will look at building malware for Android tablets and phones. Since almost all of these devices connect to a network in one way or another, there is a pretty good chance that they will log onto the company network at some point. Of course, many of the networks offer remote access but only for isolated segments of that network. This isn't true for web based services though and has become a weak spot for many companies' security.

Android runs as a virtual machine (VM) with a rebranded Linux flavor, designed for small devices but built for speed. The VM is called "Dalvik" in case you were wondering, which is a Java VM with much less overhead (demand for resources). The OS is built in C++ as are all the libraries included in the Android Software Development Kit (SDK). This means that there is a Linux kernel underneath all that GUI fluff. What this also means is that Android can run Java applications inside browsers and as standalone programs.

Third party programs can run native APIs to access built-in functions of Android like the Resource Manager, Telephone Manager and other main controls. This is a major vulnerability since there aren't many reasons for a game to have access to your location, photos, text messages or other private data. The third party applications are often written in Java while system applications are written in C++ (complied for the processor in use).

Exercises

6.18 Explore your own device's Android APKs (applications). Head over to http://developer.sonymobile.com/knowledge-base/tools/ and look for APKAnalyser. This free tool will show you how that APK works and which APIs are called. It will also show you a very nice graphic of how that app works as a flowchart.

6.19 What are some of the ways we can determine where the device owner is?

6.20 Head over to http://www.xray.io/#vulnerabilities and take at look at known vulnerabilities for Android operating on an Arm processor. If you were writing malware, which of the listed issues would you try first? Remember that they are listed in alphabetical order, not by popularity. Most phones run on an Arm core.

An Apple a Day

Now it's Apple's turn to be inspected. Apple has always marketed itself as being safe from malware because of the closed operating system and advanced security features. In truth, security in iOS for all Apple mobile devices depends on users only obtaining software from the official Apple Store. For jailbroken devices, this security feature can be bypassed, which means the Apple Store isn't an effective method to protect those devices. If the company relied solely on the protection offered by users purchasing software through official channels, than it isn't much of a practical plan at all.

Users like to share photos, attachments, messages, links and all kinds of other data. The shared data becomes an entry point for malware infections, just as with any operating system. Part of the reason we haven't seen much malware for iOS is because it is a relatively new popular platform. As iPhones and iPads become more attractive, they also become a larger target for malicious hackers. Now Apple products are a large

player in the mobile community so they are getting much more attention from malware writers.

One of the first prime-time malicious programs for the iPhone is called **Wirelurker**. This application spreads using the enterprise provisioning system, which is a function that allows a company to install custom applications without having to go through Apple Store approval. Luckily the malware doesn't do much beyond loading up a comic book unless the phone is jailbroken. Those phones will have payment information slurped up and sent back to a command and control server. Other proof of concept programs have been demonstrated in the past and shrugged off by Apple as "impossible." Yet the whole idea behind a proof of concept is to show that it is possible.

Anything that connects to the Internet is susceptible to getting hacked by way of malicious links, click-jacking, redirects, Java exploits plus a ton of other vulnerabilities. Apple products are no different.

Exercises

6.21 Wirelurker is thought to compromise up to 800 million Apple users by using desktop to USB infection of iPhone and iPads. Why do you think such a powerful program uses a simple payload of installing a comic book application when it could install more dangerous software?

6.22 Look up exploit CVE-2014-4377 to see which operating systems and/or devices may be impacted. How would this exploit work if the user doesn't have Internet access? Safari will open up a rogue PDF even without Internet connectivity. Because this is an issue with the way Safari tags PDFs as images, multiple PDFs could be loaded up without the users knowledge, thus causing a buffer overflow that could be exploited.

6.23 The web page http://www.exploit-db.com/platform/?p=ios lists a collection of known vulnerabilities in iOS, which affect iPhone and iPad devices. Many of the items listed in the database involve multiple vulnerabilities ranging from Wi-Fi access to camera control. The database for iOS vulnerabilities only goes back to 2010. Which year has the largest collection of documented exploits and what do you think is the reason?

Botnets

A **botnet** is usually several hundred up to millions of computers that have been attacked, compromised, and have a **rootkit** and **backdoor** installed without the owners' knowledge. They are unwitting hosts to malware, or **zombies**. The attacker (**bot master** or **bot herder**) can remotely command those machines to do anything he wants, from sending spam, to DDoS attacks, to stealing financial information.

If your computer is infected with a bot, it could be used in an attack. An infected computer could be responsible for an attack on the police servers. Legally, you are responsible for the behavior of your computer, just like you are for your cat or dog. What if your computer was involved in an attack on critical infrastructure in your country, like power or water supply plants?

Those kinds of attacks are called **cyberwar**, though that's a dangerous word, because what the police do is very different from what armies do.

Who is behind botnets? Sometimes individuals, but usually organized crime gangs. You certainly don't want to mess with that lot! It is said that the next war (on Earth, not in the galaxy) will be fought in cyberspace.

Would you like to be a botnet hunter? There are a few dedicated individuals who do this. The problem is that, in order to hunt down and bring down a botnet, it's possible

that you'll break some laws in the process. We'd best leave this to the professionals. They need to conduct their investigations within the constraints of the law.

Botnets are also used for **denial of service** attacks (**DoS**). Some of the recent DoS attacks have demanded money in exchange for calling off the attack. In the past, most DoS attacks relied on overwhelming servers with data requests in order to shut the servers down or force a reboot of the system. Some bot-nets have owned tens of thousands of machines which direct a single attack against another network to disrupt communication and thus business.

The bot-nets are individual computers located throughout the world but are controlled by one or more **command and control** (**C&C**) servers. Each machine is controlled by the C&C servers and told what and where to attack. The C&C servers are themselves controlled by another server called the mothership. By having layers of separate communications between the hackers and the attacking machines, locating the criminals behind the attacks is difficult.

Free Stuff

People want their favorite music, TV shows, movies and more for free. Think again! How do you think an attacker might spread their malware? An effective way to spread malware (and build a botnet) would be to attach it to something that everyone wants for free. If you use an unsafe OS, don't turn off your antivirus software to speed things up.

What is antivirus software and what other countermeasures could help avoid getting malware? Each year vendors advertise their products as better than anyone else when it comes to virus detection. The largest antivirus manufacturers, Norton, McAfee, AVG, and Kasperskey hire research organizations (magazines, news outlets, social media) to endorse their products. In reality, some of the best software is open source and therefore free. The key is to do your own research to locate tools that meet your needs, not someone else's.

Delivery Techniques

Very few people would purposely put malicious software on their own system so malware creators need a way to install their products without a user's knowledge. There are several techniques that have proven themselves effective ways to install programs without the owners knowing it. Some of the best methods are repackaging, updates, and attachments (SMS, email, web links, and other malicious URLs).

Repackaging involves the use of real programs that are offered as legitimate software through distribution systems. Malware builders would take that software and add their malware to it or recompile the code to include their payload. Google Play has had a difficult time controlling legitimate programs for Android devices. Since most users don't pay attention to the original size of the correct program, it is fairly simple to replace a good copy with a malicious copy.

Software updates are another area where malware creators can fool users into installing their programs. The malicious programmer is able to persuade a user into downloading an update for some software on the user's computer. The update advisory looks legitimate and may even point the initial URL to an actual software patch. The URL or update link is actually loading up malware while telling the user that their program is being updated. This technique has been done with Adobe, Microsoft, Java, and several other well known vendors.

We've already discussed attachments as a method for distributing malware. Yet, web links are still the easiest way for malicious software to appear of a system. Browser plug-ins for running Javascript, Ajax, PDF openers, PHP, Flash and other programs allow

malware to sneak in from malicious web pages. This means that you have to be on the look out for every possible access point when you are behind your keyboard.

One of the more interesting methods of installing malware is the use of multi-stage insertion. The malicious code is moved onto a user's system in sections to avoid detection. For example, a user might come across a link on a web site or a corrupt execution call inside the web browser from that web address. That single event allows a small program to run in the background of the machine. The program would make a minor adjustment to the system that opens a port or bypasses some security feature. Once that step is complete, another program is loaded that might just be the payload or could be a second-stage attack.

This process of multiple stage program loading can go on for as long as it is needed in order to install the malware and carry out an attack. Usually multi-stage malware is sophisticated enough that it is gathering information from large data sets, such as financial institutions or credit card database information. The massive attack on the American merchandise firm Target in late 2013 used a multi-stage attack that was updated at least five different times during the course of the breach.

Besides your typical file delivery mechanisms, some malware programmers are using communication channels built into computers to propagate. One particular program called **Flame** could use the system's own Bluetooth radio to transmit the malware to other machines nearby. Wi-Fi is also being used to transmit malicious code across airwaves. There has been rumors about some malware using high frequency audio tones to send bits to other devices. This technique has been proven to work in laboratory settings with super quiet environments. Don't expect this to work very well around your house though. You talk too loud and snore.

Countermeasures

There are a number of ways that you can detect, remove and prevent malware. Some of these are common sense, others are technical alternatives. This section provides a brief explanation and examples.

Antivirus Software

Antivirus software is available in many commercial and open source versions. These all use similar methods. They each have a database of known viruses, and they match the signatures of these against the files on the system to see if there are any infections (this is often called the **blacklist** approach). Often though, with modern viruses, these signatures are very small, and there may be false positives, things that appear to be viruses but are not.

Some virus scanners employ a technique known as **heuristics**, which means that they have a concept of how a virus behaves and try to determine if an unknown application matches these criteria. More recently, antivirus software has also crossed the boundary into **Host-based Intrusion Detection**, by keeping a list of files and checksums in order to increase the speed of scanning.

And yes, Apple Macs get malware too. Current estimates show 5,000 different types of malware for Apple products. There are exploit kits specifically designed to attack Macs. There are now numerous antivirus software programs for Macs. Search online for antivirus for Mac.

Do you use an antivirus on your iPhone or iPad? On your Android phone or tablet? On your Internet-connected TV or disc player? On your Linux box? Why don't you? Is antivirus software mandatory?

There's a list of free software and processes to fix malware problems located at https://www.soldierx.com/tutorials/Malware-Removal-Guide.

Removing Unwanted Guests

Some malware is easier to remove than others. If you come in contact with some nasty virus, trojans or ransomware most antivirus software can remove the threat in a few seconds. There are other types of malware that don't go away very easy and require some research and work to remove, like rootkits. Some types of malware are almost impossible to remove without doing some damage to the data on that system.

One of the first steps to remove unwanted software to to figure out what it is: you need to identify the malicious software. Most antivirus software scans will provide you with a name and it's up to you to research that type of software. The key is to have several different antivirus scanners on hand because one is never enough. Once you know the name of the malware, head over to http://www.malwareremovalguides.info and see what they recommend for removing that threat.

Each type of malware may need to be treated differently so do your research before you start deleting files off of a machine.

More times than not, if you have one virus you will also have several more hiding elsewhere. It's not uncommon to locate several dozen different strains of trojans sitting alongside adware or rootkits. Deal with each, working on the worst infections first.

Malware Analysis

Imagine you work for an antivirus company and you discover new malware code that has never been detected before. You will need to assess the damage it could cause and what its intentions are, document and catalog the new malware, and most importantly, name it after yourself. Imagine that!

It would be a really bad idea to run malware on your own computer, or a shared computer connected to a network, for obvious reasons. If malware analysis seriously interests you, there is a lot more you need to know and you are going to need a test system exclusively for this purpose. It is pretty easy to write your own simple malware code or search online for virus code. Please be careful when you're getting into the "dark side" of the Internet. Malware writers are actual people, often with malicious and criminal intent; you don't want to hang out with them or invite them into your home.

With static analysis, it is possible to study a program without actually executing it. Tools of the trade are **disassemblers**, **decompilers**, and **source code analyzers.** Program disassembly involves converting a program file into a listing of machine language instructions; program decompilation is converting machine language instructions into the equivalent higher-level language source code; and static analysis is examining a program without actually executing it.

What if the malware is encrypted? If the code is encrypted, your job just got a bit harder but not impossible. Malicious code that is encrypted is usually a sign that the program is a multi-stage application. The stage of code that decrypts the program could already be somewhere else on the computer. You will need to look around for any script or application that downloaded around the same time that the malware was delivered.

For regular malware the usual procedure is to install and run it in a virtual machine. Depending on the type of malware, whether it is a stand alone executable, a Java app, a script or something else, you will need to decompile the program inside a

sandbox on the virtual machine. This is not for the faint of heart. Most malware has already been decompiled and cataloged by other researchers. You can save yourself time and effort by looking up this information and following it like a roadmap.

Two sites for loading up malware are http://virusscan.jotti.org/en or https://www.virustotal.com/ . These sites run the code past well-known antivirus software and tell you the results. There are several items that you will want to pay close attention to. They are:

- **Propagation:** How the malware spreads
- **Infection**: How it installs, and remains installed despite disinfection attempts
- **Self-Defense**: How it conceals its presence and resists analysis
- **Capabilities**: Software functionality available to malware owner

Hint: Never trust or click on a pop up screen offering you free antivirus software if it says your computer is infected. This is almost always malware!

Keep in mind that the file extension JAR is a compressed Java file. If you ever want to inspect a Java element, take a look at http://en.wikipedia.org/wiki/Decompiler or the JAD project at http://varaneckas.com/jad/.

Exercises

6.24 Go online and find **Sandboxie**. (It never hurts to try a search along the lines of "like Sandboxie" too.) What OS is this used on? How does it work? What applications would you use it with?

6.25 Do you use a free antivirus software? No problem, but let's check it out. Go to the vendor's website and search for a comparison of the free software versus the commercial version (it's highly likely there is one). What's the difference between the two? What would you get for your money if you bought the full version?

6.26 Search in your favorite search engine for "compare antivirus software." Choose a current review (from this year). Which antivirus product is rated number one? What makes them different?

6.27 Now test your antivirus software to see if it has detected all the threats on your computer. First, go to Bitdefender's free online malware detector available at http://quickscan.bitdefender.com . Run the online scan. This may take some time so use your time wisely while you wait. Did Bitdefender detect any malware your antivirus software missed? If so, why didn't your antivirus detect it?

6.28 Test your AV software using a fake virus file. Go to http://www.eicar.org/anti_virus_test_file.htm and read the "Antivirus or Anti-malware test file" information carefully. The file you will be testing is not an actual virus but rather designed to look like a virus to your antivirus software. Download the file. Wait to see what happens. What does your antivirus software do? Close your antivirus message, if you get one, and complete the download procedure.

6.29 Now click eicar_com.zip. This compressed zip file contains a fake virus. What happened when you attempted to open the file? Is your antivirus program detecting this file as malicious?

6.30 Using the Internet, find an example of a trojan and of spyware. See eicar.com again.

6.31 Search the Internet for examples of rootkits and backdoors.

6.32 Now consider: could you sandbox your web browser so that anything it downloads is also trapped in the sandbox? Is this an effective alternative to antivirus?

NIDS/NIPS

Network intrusion detection systems (**NIDS**) is similar to antivirus software. It looks for a particular signature or behavior from a worm or virus. It can then either alert the user (as an **IDS**, or **Intrusion Detection System**), or automatically stop the network traffic carrying the malware (as an **IPS**, or **Intrusion Prevention System**).

HIDS/HIPS

Host-based Intrusion Detection systems (**HIDS**), such as **Tripwire**, are capable of detecting changes made to files. It is reasonable to expect that an application, once it is installed, shouldn't change unless it's updated, so watching file information, such as its size, last modification date and checksum, makes it instantly obvious when something is wrong.

Firewalls

Worms propagate across the network by exploiting vulnerabilities on each host. Apart from ensuring that vulnerable services aren't running, the next best thing is to ensure that your firewall doesn't allow connections. Many modern firewalls will provide some form of packet filtering similar to a HIPS, and will drop packets matching a certain signature.

Sandboxes

The concept of a sandbox is simple. Your application has its own little world to play in and can't do anything to the rest of your computer. This is implemented as standard in the Java programming language, and can also be implemented through other utilities such as **chroot** in Linux. This restricts the damage that any malware can do to the host operating system by simply denying it the access required. In many OSs this kind of restriction is always built in. In at least one it is not. (Go on, guess which.)

Another option is to run a full machine inside a machine using a virtual machine product like XEN or VirtualBox. This isolates the virtual machine from the host operating system, only allowing access as defined by the user.

Patch, Patch, Patch, Back-up

That's what most vendors will tell you: apply every patch, apply all patches, let us install all patches, automatically, all the time! That'll make you safe!

Except that:

1. Lots of the patches vendors push out don't apply to your particular system.
2. Every patch you install is yet more buggy, malware-prone code on your computer.
3. Patches break things just about as often as they fix them.
4. Patches can crash or destroy other stable software on your desktop machine – or your server.

Which is to say, the patch game is not the cure-all that the preachers say it is. The appropriate patches are usually beneficial, though they can cause problems. But allowing automatic updates (many, many server admins have learned the hard way) is actually very dangerous. (Microsoft has done everyone a big favor by teaching us this, over and over again.)

The key is to keep your software updated on a regular basis but test the patch before you install it on critical machines. If you can't test the patch, make sure you can reverse the patch installation. Cloud storage is getting cheaper each day and you can set up

automatic back-ups between your computer and your online account. Don't forget about local incremental back-ups. Your data is valuable, keep it safe.

Scramble

Full disk encryption is another good idea to protect your data and your system from malware. Free software is capable of providing excellent encryption while still making your computer user friendly. One of the cool tricks of using disk encryption is that it replaces your boot sector with its own bootstrap. This keeps your risk of rootkit and boot sector malware infections lower.

Malicious code cannot attack something that it cannot see. Encrypted files keep sensitive information under your control so it would not be sent to the malware operator. This limits the malware's ability to capture useable information from you such as passwords, account details, those photos of you shooting milk out of your nose and your latest report card.

Don't just encrypt your hard drive. Encrypt all media and your phone.

Exercises

6.33 Do a search on the term "automatic update causes." How many things did automatic updates cause? Or at least, how many results do you get?

6.34 Research antivirus software for mobile phones. Also search for anti-malware for tablets (e.g. iPad and Android). Are these tools effective? Who uses them?

6.35 Research Stuxnet, Duqu and Flame. For each:

- What systems did it affect?
- What was its payload?
- What made if different from any other malware?
- How do you remove them from a system?

6.36 Matching Game: Research each of the following and match it to the type of countermeasure that it is:

http://www.virtualbox.org	NIDS/NIPS
http://www.tripwire.org	Antivirus
http://www.snort.org	Firewalls
http://www.checkpoint.com	Sandboxes
http://www.clamav.net	HIDS/HIPS

6.37 Research how NIDS/NIPS and HIDS/HIPS work.

6.38 Research Firewall solutions on the net. You can analyze your firewall logs and submit your logs to the SANS Internet Storm Center, DShield at http://isc.sans.edu/howto.html .

6.39 Look up **chroot** on the internet. Read about this type of "jail" or "sandbox."

6.40 Draw an **Attack Tree**. Do an advanced search on "site:www.schneier.com."

6.41 Malware is never good for anyone, other than those trying to profit from it and risking being caught. Everyone wants to avoid being infected. Take a look at this flowchart to see exactly what you're avoiding when you protect yourself from malware: Inside the business of malware http://www.computerschool.org/computers/malware/ .

6.42 Computerschool.org has an infographic on the Business of Malware. Find it.

Conclusion

Any good immunity from malware comes with a strong understanding of malware. While we can't cover every possible type of malware (because by the time you read this there will be new ones), we've exposed you to some critical points. For instance, you can barely trust the shortcuts on your computer's desktop, and you can't trust at all any file that comes to you unrequested. The key issue is trust, which involves a keen awareness of how vulnerable you become when you give trust.

We don't want you to become deeply mistrustful of everything; that's an attitude that will lock you out of lots of opportunities. Instead, be very clear in your mind when you give anyone access to you, that you're giving them trust. The same principles that make networks secure can make you secure too. Network segmentation that allows only tightly controlled visibility, for instance, is a good practice in both networking and real life.

We also aren't encouraging you to build malware and unleash it on the world or on your acquaintances. Now, probably more than ever in human history, actions have consequences. Don't kid yourself that we couldn't find you if we tried, and we're not nearly as scary as some governments and law-enforcement agencies.

Instead, we want you to see how malware works and become sensitive to the scams in uses. That makes you not just safer from malware, but safer in your whole world.

Use this power only for good, young padawan.

Today's teens are in a world with major communication and productivity channels open to them and they don't have the knowledge to defend themselves against the fraud, identity theft, privacy leaks and other attacks made against them just for using the Internet. This is the reason for Hacker Highschool.

The Hacker Highschool project is the development of security and privacy awareness learning materials for junior high and high school students.

Hacker Highschool is a set of lessons and a practical means of making hackers. Beyond just providing cybersecurity awareness and critical Internet skills, we need to teach the young people of today how to be resourceful, creative, and logical, traits synonymous with hackers. The program contains free security and privacy awareness teaching materials and back-end support for teachers of accredited junior high, high schools, and home schooling. There are multiple workbooks available in multiple languages. These are lessons that challenge teens to be as resourceful as hackers, including safe Internet use, web privacy, researching on the internet, avoiding viruses and Trojans, legalities and ethics, and more.

The HHS program is developed by ISECOM, a non-profit, open-source research group focused on security awareness and professional security development and accreditation.

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM