Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: Crossref, ROAD & Google Scholar

25

A Review on Assessing Extent of Malware Damage and Basic Countermeasures

Omkar Mandar Pradhan[*], Harshita Kansara[**], Durvankur Sanjay Balkawade[**]

[*] M.S.c Computer Science

[**] School of Computer Science, MIT World Peace University

Abstract

In today's era of a digital world, Malware is a threat that looms above our heads. Attackers find a way to infect our systems to access our data and possibly disrupt our lives. Malware comes in many different forms and factors. The Malware could be harmful or a simple inconvenience, and it could cause a system-wide breakdown or even attempt to set up a backdoor. We chose the topic to get hands-on experience on at least some kinds of feasibly accessible Malwares. Learning about different Malware and testing a few of them out was a great way of understanding them clearly. In this research paper, We will be discussing kinds of Malware, their possible impacts, and some countermeasures. As Important as countermeasures against Malware is, understanding different types of malware threats is just as important.

We will be testing some malware threats in a virtual environment using two different virtual machines configured to be vulnerable enough to get a good look at the damage done. We are using virtual environments supporting snapshots to revert the system to a working state if some malware affects it a way that leaves it unusable.

We tested Malware that can establish a reverse TCP connection to give the attacker access to the system. We also tried some minor malware, which can kill random processes, delete arbitrary files, copy the largest files, mess with extensions, mess with environment variables and insert keyloggers.

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: Crossref, ROAD & Google Scholar

1. Introduction

Malware is a term given for a collection of software that intends to cause harm to the system. Malware is shorthand for malicious software, which contains malicious code developed by cyber attackers to either gain unauthorized access to a network or damage it (Cyber Edu, n.d.). A wide variety of Malware exists that all work in different ways to achieve other goals. I will be discussing the different types of Malwares found and their possible purposes. Malware like Trojans are carriers that act beneficial but end up infecting your system. Malware has evolved since its clear conception in the early 1970s.

Malware has gotten out of hand lately, even with the presence of antimalware and antivirus software. Developers have been able to bypass security systems to infect their targets. Users need to know what Malware is, and they need to protect themselves against Malware. Cyberattacks these days are multi-faceted as they don't utilize a single attack vector, and attackers aim to get multiple points of entry into a system to gain access to data. We attempt to discuss Malware and test some basic malware on a vulnerable virtual machine to study their effects and develop precautionary measures to protect against these attacks.

We play the role of both an attacker and a defender to check the extent of damage done by Malware and some possible countermeasures to cull the damage.

2. Literature Review

As Malware is a term used to classify any harmful software, we have to look into the categories of Malware that have set goals and outcomes. Malware is a broad term, and to understand what Malware (Kaspersky, 2023)is, we will look at some of the categories (Baker, 2021).

- **Trojans:** A Trojan is how most systems are infected, based on the ancient Greek tale of the trojan horse in which Greek soldiers used a giant wooden horse to infiltrate the walled city of troy and destroy it from within. A Trojan in the cyber world works similarly, as a file carries a malicious code script executed by the system once the victim opens the file. Lately, attackers have come up with a way that utilizes macros to perform their ulterior motive. People use macros to make their lives easier. Attackers exploit this by embedding malicious code which the victim cannot see, the

victim ends up permitting macros on the system, and the Malware executes its ulterior motive. Trojans are dangerous as a user could get an infected file off the internet without even realizing it.

- **Worms:** A worm is a type of Malware that spreads across a network by itself. The worm is a special kind of Malware that doesn't require a software attachment or interaction. The worm is especially dangerous as it feeds off the system's resources and keeps scanning for newer machines to infect. A worm could aggressively infect multiple devices causing a network-wide infection that could damage the resources considerably. The worm's prime directive is to stay active as long as possible on the target system and then spread itself to as many systems as possible. Detection of a worm is possible if the worm starts consuming resources extensively.
- **Virus:** A computer virus is a malicious code that attaches itself to an application while execution. A virus is also self-replicating, but its intentions are not to spread extensively over a network automatically. Instead, it intends to replicate and damage the host system to prevent regular functioning. A worm can automatically replicate itself over the network, whereas a virus needs some human interaction to transfer itself over a network. A virus could be hazardous to the system as some viruses cause damage by deleting files, applications or even flooding the network. Detecting viruses without dedicated software could be tricky. The user might notice a decrease in performance, loss of data, spam, and frequent crashes.
- **Rootkits:** A rootkit is a program that aims to provide elevated access to a computer while reaming unnoticed. The rootkit gives an attacker remote access to your system, which the attacker can steal data. Rootkits are challenging to detect as all they do is provide escalated access remotely. A rootkit can stay on the target system for a long time without being detected, making it dangerous. Several types of rootkits exist that are segregated based on the installation location. An attacker could send a Trojan carrying a rootkit to a victim to gain access, or the rootkit could be already present in the hardware. Some organizations install rootkits on their devices as a backup plan in case of device theft or loss.

- **KeyLoggers:** Keystroke loggers or keyloggers are a form of Malware that monitors a victim's keyboard activity. Key loggers work by capturing every keystroke made by an unsuspecting victim and storing it for later retrieval or sending the captured data over a covert channel. Key loggers are extremely difficult to detect when implemented as they never directly interact with the victim. Keyloggers are a form of spyware implemented to monitor only a tiny domain of activity. However, dangerous key loggers have an actual use case scenario in monitoring employee activity or children's activity. A simple countermeasure is to use virtual or on-screen keyboards if possible, as not all keyloggers are capable enough to capture pixel-based strokes.
- **Mobile Malware:** As the name suggests, mobile Malware specifically targets handheld or mobile devices. Mobile malware targets devices running mobile operating systems that work on a different architecture. Mobile Malware is becoming an imminent threat due to the increase of BYOD[Bring Your Own Devices] policies in organizations. Attackers develop Malware to exploit vulnerabilities present in mobile operating systems. Mobile Malware is again a broad term that encapsulates all the previously discussed types of Malware modified to work on a mobile operating system. Everyone currently uses mobile devices for everything from emails to actual banking, and it makes sense for an attacker to compromise a mobile device with appropriate Malware.

3. Methods

Creation of Virtual environments

As testing malware or exploits on existing systems is not possible or feasible, we will be utilizing virtual environments to test some basic kinds of Malware. we used Oracle VM virtual box (Version 6.1.18 r142142 (Qt5.6.2)), an open-source virtualization software, to set up our testing environment. Virtual machines were my best option to test out Malware due to the availability of technology like snapshots. Snapshots allow the user to bring a virtual machine back to a saved state to revert any damage or changes made during the operation. We isolated the virtual machines on a virtual network to prevent any malware we tested from affecting our regular desktop system on accident.

We created two virtual machines to create a simple testing environment. The first was an officially acquired virtual machine running Microsoft Windows 2007 (Microsoft), and the other virtual machine was an officially acquired Kali Linux virtual machine (OWASP). Kali Linux is an operating system developed by OWASP as a primary penetration testing operating system.

We disabled the Windows firewall in the virtual machine for the initial testing to make it vulnerable to the basic malware and discovery techniques. Kali Linux comes packed with valuable tools for penetration testing. We set up a shared folder to transfer files quickly between the two created virtual machines, the sahred folder was on a standard desktop environment that allowed direct communication with the 2 virtual machines.

Using NMap and ZenMap for basic reconnasiance

We used nmap and its gui tool zenmap to perform discovery techniques to check if the machine is discoverable using basic, intense and port scan(specific and generalized) methods.

Scans from L1 – L2 were used.

Nmap CLI was used to perform a basic banner grabbing reconnaiance to gain onformation about the target system.

Using Armitage and Metasploit to create payloads

Metasploit is a penetration testing framework used by attackers and defenders alike to understand hacking techniques and methods. It is a more accessible alternative to manually scripting a payload and executing an attack (Porup, 2019).

Armitage is a java-based GUI developed for the Metasploit framework. Attackers and defenders use Armitage to understand better and initiate different kinds of attacks, and it can even scan for targets on the network.

Most of the Malware we created utilizes the reverse TCP connection module present in the Metasploit framework. The reverse TCP module initiates a connection with the victim system, and an attacker can use this connection to implement other exploits or control the system.

Metasploit is a command-line-based interface that an attacker uses to initiate attacks or scans. The reverse TCP module's payloads were created and transported to the target system via the established shared folder.

We used Armitage to scan the network for devices using Nmap and after detecting devices. We then used Armitage to create reverse TCP payloads that we transferred between the virtual machines via the shared folder.

Creating a simple script-based malware

Developers use scripting to write executable scripts to perform tasks on the command shell/prompt. The creator meticulously scripts most Malware to achieve their goals without being noticed by the target or security systems. As we just wanted to test out the damage a well-crafted script could do to a vulnerable system, we made simple scripts that we could execute through a web shell or a reverse shell. A packaging tool can package created scripts with useful software or files to make it even more inconspicuous.

- Script, one enumerates files in the current directory and stores the output in a text file
- Script two copies and pastes a file from the guide in the same place
- Script three looks for the largest file and deletes it

We transferred these scripts to the Kali machine and executed them using a web shell on the windows target machine. The scripts are not inherently dangerous with the correct access controls activated, the scripts could be merged and modified to make them more dangerous.

Using Fat rat to create Trojans

The fat rat is a Trojan creation software used by attackers to make payloads and backdoors easily. The fat rat is a shell-based utility that allows a user to create Trojans or payloads. Rat is an acronym for a Remote access Trojan, and a rat is used to access an infected system remotely to steal data or cause damage. The tools combine popular payloads like Metasploit-based payloads and MSFvenom payloads. The tool can create a payload contained in a word document to execute a script or a command after opening.

We used fat rat to create reverse TCP-based Trojans for a windows machine, an android

device and tested the different delivery methods included in the tool. Fat Rat is capable of inserting backdoors in preexisting android APK files as well.

We shared the payloads created using a shared folder and executed the files on the target machine. As the fat rat supports multiple Trojan creation techniques, we made a simple payload, a shell, a word-based, and an android payload.

Using Beast to create Malware

Beast is a windows-based tool used to create Trojans and backdoor access utilities. Beast contains a file binder that combines multiple files to change the signature of the final package to bypass security systems.

We used Beast to compromise a second windows machine that was the clone of the first vulnerable windows machine created. We used the beast server creation utility to create a server and give it a different name and icon. We also set it up to inject its process into explorer.exe to try and hide it even better.

We shared the file created by Beast using the shared folder and executed it on the target machine.

Creating a simple keylogger in C

A key logger is a software that can track and monitor keystrokes made by a user. Key loggers can even be hardware-based which are physically installed in a system to track keystrokes. Depending on the complexity and the security access, a key logger can either store the recorded data locally or send it over the network. The idea behind storing recorded information locally on the system is that the system is accessible either remotely or physically with little to no effort.

We wrote a simple keylogger using C that logs the keystrokes once activated and stores the recorded keystrokes in a text file near the executable. The key logger is virtually untraceable by regular users. Shutting the key logger down requires terminating its process using a task manager or a similar method.

A developer could create a similar keylogger using python and the OS library. An even more straightforward keylogger utilizes a web shell to execute a simple echo command that echos

all the keystrokes entered by a user into a file or on the console itself.

Using a wrapper to mask Malware

A wrapper is a tool used to mask malicious files to try and get past security systems. An attacker can use a custom packer to alter a file's digital signature to change what it looks like to the security system entirely. The wrapper may even support encryption functionality to mask the file even further. The wrappers are a type of glue ware that binds different software together.

We used IExpress wizard to try and get past malware scanners by wrapping them. Instead of creating Malware, we utilized available malware files to verify if the security system put in place worked. We packaged the malware file into an EXE file containing road rash that executes the script if the user opens the game.

Using JPS virus maker to create a simple virus

JPS virus maker is a GUI-based tool used to create executable virus files for windows machines. It has many inbuilt scripts that an attacker can use to damage a target system. Creating viruses has become pretty easy lately due to utilities like these.

We created a simple virus that disables the start button and disrupts user mouse control. We transferred the virus using the shared folder made for the virtual machines. As we played the role of both the attackers and defenders, it was easy for me to test out the impact of Malware on the different systems.

Using internet worm maker thing to create a simple worm

Internet worm maker thing is software used by attackers to create worms. The tool has multiple choices and options for the user to create a worm. It even has the functionality to create a worm with a custom code or batch script.

We used the functionality to embed the basic script malware we created into a worm. As we isolated the virtual environments from the network, we protected our actual systems from the worm.

Using Metasploit to compromise an Android device

We used an old mobile device to test out android based attack vectors and attack techniques.

The Metasploit framework contains payloads and packages written by a developer to compromise the security of an android device. Attackers package MSF venom payloads into sound or preexisting APK files that the victim can install on the android device to infect the machine.

We created a simple reverse TCP module-based malware using Metasploit to compromise my android device. As android is a Linux-based operating system, Metasploit is used to access the shell and gain root access. The root access can then be used to establish ftp connections or modify user details and reroute system logs.

Activating the firewall to test security:

Testing out Malware without any security measure is an excellent way to check the impact of the Malware. We wanted to check out the subterfuge capabilities of some programs by testing them against a very basic firewall provided by windows.

We added new rules in the firewall list to deny TCP/FTP requests and disabled software installation without admin access.

4. Findings

Effect of ZenMap (Nmap) discovery:

ZenMap Discovery modes: We used Zenmap to attempt to find the virtual machine on the created virtual network with varting intensities of scans before and after enabling the basic firewall. Zenmap was also used to perform banner grabbing to learn more about the target system.

With the fireball enabled Zenmap was not able to detect the virtual machine using a basic scan or and intense L1 scan. Using intense L2 scan Zenmap was able to detect the machine and list out open ports.

Banner Grabbing proved inconclusive with the firewall enabled.

With the firewall disabled zenmap was able to detect the windows virtual machine immediately using even the basic scan, an intense scan listed out all the open ports and services.

Banner grabbing listed out the SMB service version which allowed us to check it using CVE

to learn about the vulnerability.

Effect of Malware on the vulnerable Windows machine:

- **Armitage Reverse TCP Payload:** We used Armitage to create a reverse TCP payload to infect the vulnerable windows system. Initially, Armitage detected the system and performed an intense Nmap scan to determine open ports and services. We used the reverse TCP payload shell to create a payload and initiate a listener on the system. We used the shared folder to transfer the executable payload and start it on the windows system.

  The Metasploit listener was immediately able to establish a connection with the system. We used the established connection to execute shellcodes and gain remote access to the system.

- **Script-Based Malware:** We created three simple executable bat files to test the impact of elementary script attacks.

  The first file enumerated the directory it was run in and saved the output onto a text file. We can then recover the text file using any possible means.

  The second script file copied the first subdirectory it found and pasted it inside the folder. The script aimed to consume disk space which over time would reduce the performance.

  We designed the third script file to seek out the largest individual file in the current directory and delete it permanently—the script aimed to delete the user's sensitive data. In windows the C drive genrally contains system files and is more often than not the largest directory so a possibility of deleteion poses a threat.

- **Fat Rat Trojans:** We used Fat Rat to create reverse TCP payloads that we saved using three techniques.

  We created a simple EXE payload with the reverse TCP shell to access the system remotely using Metasploit. Executing the file establishes a connection with the listener.

  The second method we utilized was a simple bat file containing a reverse TCP shell payload. Executing the file in a shell environment establishes a connection with the listener created using Metasploit.

The third method we utilized was to create a word file-based Malware that contained a reverse TCP shell. Opening the infected file on the system established a connection with the Metasploit listener on my attack machine.

- **Beast Trojan:** We used beast Trojan to create a server file for a vulnerable windows machine and transferred it using the shared folder. We started the server on the target machine.

  We used a different windows machine to start the beast client that gave me access to the vulnerable system.

- **Keylogger created using C:** We transferred the keylogger executable file to the target system using the shared folder and started the keylogger. We terminated the keylogger process after making some keystrokes entering login credentials on a simple HTML form we created.

  We accessed the log file created by the keylogger and verified the keystrokes we entered. The simple keylogger was able to register all keystrokes except the Esc key as the event generated by Esc cant be decoded without os libraries.

- **Malware packaged using wrapper:** We used the online service provided by Virus Total to verify malware files. Using the web-based file checker, we were able to confirm that a script had a previously used signature.

  We used the wrapper to try and mask the Malware inside different files. After wrapping the malicious script, only a few antivirus software detected the malware file's signature.

- **A virus created using JPS:** We used JPS virus maker to craft a simple virus that removes the start button and disrupts mouse control for the infected system.

  After executing the file, it disrupted mouse control and removed the start button successfully. Even after a restart the effects didn't seem to g away so the system had to be restored using a fresh snapshot.

- **Worm created with a custom script:** We used the Internet worm maker thing to make a worm that executed a custom script that copied files into a subdirectory.

  The worm successfully executed the file copying script and even reached the second windows virtual machine that was created and enabled just to test the ability of the worm.

Effect of Malware on windows machine after activating firewall:

- **Armitage Reverse TCP Payload:** Armitage could not detect the system using a Nmap scan after starting the firewall. We had to manually add the machine and set up a listener. As we added the firewall rule to prevent TCP connections, the file could not connect with the listener.
- **Script-Based Malware:** As the scipt based malware was executed in a non protected folder the scripts were able to achieve the set outcome. Attempting to execute the script in a protected folder environment like C drive didn't let the scripts perform their jobs and required manual confirmation.
- **Fat Rat Trojans:** Although created using different methods, the fat rat Trojans utilized a reverse TCP connection to connect with the listener. As the firewall rule blocked TCP connections, the Malware could not establish a successful connection with the listener.
- **Beast Trojan:** As the beast Trojan utilizes an open port to connect with the client, we turned the configured port off. The Trojan was unable to establish a connection with the client.
- **Keylogger created using C:** The keylogger designed was almost undetectable, so the presence of a firewall didn't impede its function. We was able to recover logged keystrokes after a dummy session by accessing the directory manualy.
- **A virus created using JPS:** The virus created using JPS was not particularly harmful to the system, but the older windows defender could not detect the virus and prevent execution. The virus was able to remove the start button and disrupt mouse movement. The system had to be restored to last stable state to continue further testing.
- **Worm created with a custom script:** The worm created using the tool carried the custom script to copy and paste files into a subdirectory. The older version of windows defender was unable to detect and stop the execution of the worm. Adding a second layer of security to the cloned windows machine prevented the worm from copying itself to the other device.

Effect of crafted Malware Android Device:

The android device we used was an older device without security patches.WeI transferred the

modified APK onto the device using a data cable and executed them to connect with the listener setup using Metasploit.

After a successful connection, we issued simple commands on the device shell to list directories and processes.

Effect of crafted Malware on android device with antivirus:

We used the same machine and payloads for the tests. We added Kaspersky mobile antivirus to the device to check the subterfuge capability of the crafted Trojans. The antivirus software was able to identify the infected file after a single scan hence alerting us before installing the file and compromising the device.

5. Discussion

As discussed, Malware has spread to a great extent. Users unknowingly download and execute malware scripts on their unprotected systems leading to data loss and theft. Protection against cyber-attacks comes in many shapes and forms. The process is not as straightforward as it should be. Protecting against the plethora of malware types is not easy. Sometimes a malware attack is multi-faceted, which means it could have multiple attack vectors or end goals. Protecting against a single plan or attack vector can leave clinks in your armor.

Following some practices can help safeguard the user against a malware threat. I will be discussing some methods to defend against Malware. These are standard practices that users could follow to prevent malware infections.

1. **Use antivirus software:** Antivirus software has evolved to be more accurate at detecting Malware. Having layers of security is essential to protect data and user privacy in this digital era. Freely available antivirus software or default antivirus software is better than keeping your system unguarded. Antivirus databases are updated regularly to keep newer signatures in check. Antivirus software also contains a sandbox environment implementation to allow the user access to the possible threat in a safe environment. Antivirus software can schedule scans to detect and clean any malware found on the system.
2. **Use a firewall:** System administrators implement firewalls to block access to a

private network. Firewalls can help protect against Malware that slips by antivirus and attempts to connect with the attacker. Administrators can add rules to firewalls to deny requests made by unauthorized software. A firewall can protect against any cyber threat that attempts to connect to the network after installation or execution, effectively blocking access to backdoors.

3. **Avoid using untrustworthy websites:** Users download all kinds of files from the network. Attackers can infect downloadable files with some malware that can damage the user's system. A user must remain careful while downloading any file from the web. Some browsers have basic implementations to warn the user if a file or website seems suspicious. It is advisable to follow the browser warning to prevent malware infection.
4. **Avoid opening attachments from unreliable senders:** Emails are a large part of today's conversation methods, preferred by many. Reputable email clients can scan files for Malware and warn the user about it. Some email clients cannot check encrypted documents to verify security. Avoiding opening emails from unreliable senders is a good practice.
5. **Keep offsite backups if possible:** Threats like ransomware go after the victim's data, ransomware encrypts data and demands a price for a decryption key. Keeping offsite backups helps prevent dependency on one system. Some viruses and worms are also known to destroy crucial data. Regular backups let the user return their system to a previous state which is functional and noninfected.
6. **Close unused services and ports:** Some malware can initiate a conversation with a central server using open ports on a system or even utilize available services to execute their goals. Closing inactive services can help lower the number of possible attack surfaces. Open ports must be protected using some authentication, and unsued ports should be closed to deter any communication attempt.
7. **Prevent Script execution if possible:** Script execution is an essential part of many systems and is functional. Attackers can use scripts to damage the user system and perform malicious code. Disabling unauthorized script execution can prevent Malware that executes scripts from achieving their final goal—setting up

authorization before script execution can warn the user about an imminent attack.

8. **Disable macros if possible:** Macros are essential while working on extensive data. Attackers can use macros to execute malicious code and scripts. Disabling macros can help the user prevent macro attacks or Malware based on macros. Microsoft Office allows the users to open downloaded files securely by disabling any access other than reading, and this could help protect users from document-based Malware.

6. Conclusion

Malware is a threat that looms over every user in this digital era, and it threatens not only our data but also our privacy as users. Attackers manage to find a way to defeat our security measures and disrupt our lives. Malware has many types, and each of them serves a different purpose.

We used a virtual environment to test the impact of some malware and tried to understand the level of threat we face. Testing different malware helped me understand the field a bit better. We tested the Malware in a vulnerable environment by disabling the firewalls and safety precautions and then tried the same using primary security countermeasures.

We were able to get an idea about the impact of basic Malware on a system and was also able to create a script that can cause some damage. We used a few different Trojan methods to package or wrap the Malware. A service like Virus Total is essential in detecting malware signatures.

We were able to test out some basic countermeasures to protect our systems against Malware. Antivirus software is essential to protect our devices from Malware that can damage our systems. We need to be vigilant about the resources we get from the web, as attackers can modify Malware to such an extent that it is difficult to detect.

References

1 Baker, K. (2021, August 19). THE 11 MOST COMMON TYPES OF MALWARE. Retrieved from crowdstrike: https://www.crowdstrike.com/cybersecurity-101/-malware/types-of-malware/

2 Cyber Edu. (n.d.). What is Malware? Retrieved from Forcepint: https://www.forcepoint.com/cyber-edu/malware

3 Kaspersky. (2023). kaspersky>home security> resource center> threats. Retrieved from kaspersky: https://www.kaspersky.com/resource-center/threats/types-of-malware

4 Microsoft. (n.d.). Microsoft Edge Developer. Test IE11 and Microsoft Edge Legacy using free Windows 10 virtual machines you download and manage locally. Microsoft. Retrieved from https://developer.microsoft.com/en-us/microsoft-edge/-tools/vms/

5 OWASP. (n.d.). Kali Linux. Virtual Machines. Retrieved from https://www.kali.org/-get-kali/#kali-virtual-machines

6 Porup, J. (2019, March 25). What is Metasploit? And how to use this popular hacking tool. Retrieved from CSO India: https://www.csoonline.com/article/3379117/what-is-metasploit-and-how-to-use-this-popular-hacking-tool.html

7 screetesec. (n.d.). Git Repo. Retrieved from FatRat Git Repo: https://github.com/-screetsec/TheFatRat