International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue V, May 2019- Available at www.ijraset.com

Sandbox Evasive Remote Access Trojan

Kamlesh Tukaral[1], Dr. Ravi K Sheth[2]
[1]*Student Master in Technology Cyber Security,* [2]*Assistant Professor Raksha Shakti University*
[1]*Department of Information Technology and Telecommunication, Raksha Shakti University, Ahmedabad, India*

Abstract: ***A Remote Access Trojan(RAT) is remote control software which installed on a system, will allow a remote computer to take control of it. A RAT allows an attacker to remotely control a computing system and usually consists of a server unnoticeably running and listening to definite TCP/UDP ports on the victim's system as well as a client acting as the interface between the server and the attacker. The most common means of infection is through installing games, email attachments or downloading applications from unverified third party mediums. The developer of the malware usually uses various fully undetectable techniques in order to make the malware unsuspicious to the users. These often implanted in lawful programs through RAT-endorsed procedures. They are stealthily planted and help gain access of victim's system, through patches, games, E-mail attachments, or even in legitimate-looking binaries. Once installed, RATs perform their unpredicted or even unauthorized operations and use an array of techniques to hide their traces to remain undetectable and stay on victim systems for the long haul. To counter such advance Trojans modern security solutions use method like sandboxing. Sandboxing is an automated technology for malware detection that's widely used by traditional antivirus programs and other security applications. By placing a malware into a controlled virtualized environment where it can't cause any harm, security software can analyse the behaviour of the malware and develop protection against it. Cuckoo sandbox is the leading open source automated malware analysis system. You can throw any suspicious file at it and in a matter of minutes Cuckoo will provide a detailed report outlining the behaviour of the file when executed inside a realistic but isolated environment. In this paper, our objective is to know more about the sandbox and potential methods that can be used by a RAT to evade the sandboxing method.***

Keywords: ***Sandbox, Evade AV, Advance RAT, Remote Access Trojan, Trojan, third party applications, email attachments, malicious, compromised system***

I. INTRODUCTION

Malware or malicious software is any program or file that is harmful either to a computer software or user. In Spanish, "mal" is a prefix that means "bad" overall making the term "bad ware". Malware includes worms, computer viruses, Trojan horses and spyware which are few examples of malware. These malicious programs can carry out a variety of functions, ranging from stealing, encrypting or deleting sensitive data, they might also alter or hijack core computing functions and monitor users' computer activity without users' permission or knowledge. One of the most common but dangerous Trojans is called Backdoor Trojans (also known as Remote Access Trojans or RATs). A computer with a sophisticated backdoor Trojan installed may also be referred to as a zombie or bot. These threats are almost invisible to the user and if they succeed in entering the system, the intruder has remote access to the system. The intruder/attacker, with the help of the Trojan, is now in control of the computer. Few abilities of backdoor Trojans are to gather information, terminate or run a task or process, download and upload files, reports to the attacker's machine, change critical setting in Windows, restart or shutdown the PC, and perform Denial of Service (DoS) attacks.

Sandboxes are automated behavior-based malware analysis systems that are at the core of most network security solutions today. The deployment of sandboxes to detect advanced threats began over a decade ago. Back then, malware authors had already found ways to evade traditional antivirus solutions, which rely on static analysis, by using techniques such as polymorphism, metamorphism, encryption, obfuscation and anti-reversing protection. Malware analysis sandboxes are now considered the last line of defense against advanced threats. Cuckoo sandbox is most widely used automated malware analysis system with infinite application opportunities. Moreover, it is open source and compatible with Windows, OS X, Linux and Android.

II. REMOTE ACCESS TROJAN

There are two programs required for this tool. A client program runs on the Attacker's computer, it listens for the server program on the specified port to make connection, implements a GUI and the attacker can send through various commands to carry out the attack. The server program runs in the background of the victim's machine, hidden from the user. It makes connection with client program whenever it's online and uses it to receive commands from the attacker and carries out the required function. A simple Network Program consists of 2 parts, a server and a client. The server program must be started first and waits or listens for the client

©IJRASET: All Rights are Reserved

program to connect. However, it is also possible to have the server connect to the client as in the case of Reverse-Connection-RATs that are used to bypass firewall or router limitations. The server program will usually be on one computer while the client program will be on another computer. Both can be on the same Local Area Network, or, on the Internet. After connection is established, the client will send a command to the server. Upon receiving the command, the server will execute it.

Remote Access Tool is a piece of software used to remotely access or control a device. This tool can be used legitimately by system administrators for accessing the client computers. Remote Access tools, when used for malicious purposes, are known as a Remote Access Trojan (RAT). They can be used by a malicious user to control the system without the knowledge of the victim. Most of the popular RATs are capable of performing key logging, screen and camera capture, file access, code execution, registry management, password sniffing etc.

A. Damage by Remote Access Trojan

Because a RAT enables administrative control, it makes it possible for the intruder to do about anything on the targeted victim's computer, including:

1) Observing user behavior through key loggers or other spyware.
2) Accessing confidential information, such as credit card.
3) Starting a system's webcam and recording video.
4) Taking screenshots.
5) Distributing viruses and other malware.
6) Formatting drives.
7) Deleting, downloading or modifying files and file systems.

B. Trojan Usage

1) Dark Comet was in use by the Syrian government to spy on its citizens. The general population had taken to employing VPNs and secure chat applications to block government surveillance, so the spyware features of DarkComet enabled the Syrian government to avoid those security measures.
2) It can also be used to spy on suspicious targets by the government.
3) Intelligence agencies in Germany use malware to track computers of people under suspicion. The Trojan is able to track user chats and conversations on smart phones and PCs.
4) German agencies have mandatory authority to place Trojan horses on the hard drives of suspected criminals using email that would install key loggers, record webcams and microphones and scan infected hard drives for documents, diagrams and photography.
5) Chinese covert intelligence bodies have been associated with Trojan horse activity against both other governments and private industry
6) NSA was spying on everyone living in US which were later disclosed by snowden.
7) A government Trojan is installed on a computer or network by a law enforcement agency for the purpose of capturing information relevant to a criminal investigation, it acts as a spyware. Depending on the prerequisite, government Trojan horses may interrupt email or VoIP traffic; scan hard drives for applicable digital media or even record conversations and video conferences. This type of software captures data and then sends it back to a central server for processing and analysis without a user's knowledge; it is generally called as a back door Trojan horse virus.
8) Swiss government agencies have reported to be working with Internet service providers to record speech on an infected PC's microphone, as opposed to of intercepting encrypted voice packets.

Government Trojans represent a step in turning the tables on cybercriminals by using a proven mechanism for capturing data secretly

III. SANDBOXES

Sandboxes are automated behavior-based malware analysis systems that are at the core of most network security solutions today. The deployment of sandboxes to detect advanced threats began over a decade ago. Back then, malware authors had already found ways to evade traditional antivirus solutions, which rely on static analysis, by using techniques such as polymorphism, metamorphism, encryption, obfuscation and anti-reversing protection. Malware analysis sandboxes are now considered the last line of defense against advanced threats.

©IJRASET: All Rights are Reserved

The operating principle of a sandbox is simple. It determines if a file is malicious based on the observed behavior of the file in a controlled environment over a defined analysis period.

It does this by recording all the actions performed by the file and determining if any of these represent malicious behavior patterns. Since detection is not based on static signatures, sandboxes can even detect zero-day and targeted malware, previously unknown to security researchers or analysts.

The success of behavior-based malware detection hinges on the behavior exhibited by the file during analysis. Thus, the objective of any sandbox evasion technique is to conceal the real behavior of the malicious file, thereby evading detection. Malware authors are always looking for new, innovative ways to elude sandboxes.

IV. POTENTIAL METHOD TO DETECT SANDBOX

After reading so much about the sandboxing method one can say that the core principle of this method is behavioural analysis of the file. It determines if a file is malicious based on the observed behaviour of the file in a controlled environment over a defined analysis period. It does this by recording all the actions performed by the file and determining if any of these represent malicious behaviour patterns. This malicious behaviour patterns includes:

1) Decrypting itself for execution
2) Trying to copy itself in startup programs
3) Making changes in registry
4) Trying to get a remote connection
5) Trying to disable security features

As we can notice that the real analysis starts when the RAT gets executed. So we can say that the best method to avoid the sandbox is to not get executed in the sandbox. For this purpose, first we need to detect whether our RAT is running in a sandbox or not. Below are some methods which can be used to detect a sandboxing environment.

a) We can try to detect the system configuration of the system before executing our RAT. For example, no. of CPU cores, RAM size, hard disk size, etc. Because no normal system would have 1 GB RAM and below 250 GB HDD in present scenario.
b) We can try to detect user interactions like scrolling of a document or clicking event of mouse.
c) We can even keep a check box pop up for user interaction before executing the malicious payload.
d) We can try to check the environment of the system for particular processes related to sandboxes. Also we can try to identify VM artifacts.

By using any of the above methods we would be able to evade the normal sandboxing methods use by antivirus solutions. For increasing the chances of evasion I suggest to collectively use more than one method to detect the sandbox environment.

V. PROPOSED METHOD: SANDBOX EVASIVE RAT

Phase one includes coding the two programs required for the project, namely- Client and Server programs. The Server program has functionalities which include listening for connection for active clients, relaying the appropriate commands to the client and implementing a Graphical User Interface for the attacker's convenience. The Client program's functionalities include trying to establish connection with the server whenever the mobile phone is connected with the network; carry out the necessary functions as per the user's commands.

In second phase, having finished the client and the server programs, we now test the connection between the two programs, the victim's device and the attacker's.

The client, which runs in the background of the victim's device, sends a connection request to the attacker, whenever the victim's machine is online and connected with the network.

The server (attacker) is constantly online, listening for active clients. The phase is successfully completed when a connection is established between the attacker and the victim; the attacker is notified of the victim's presence and is successfully able to send through commands which are executed on the victim's device.

Now in third phase we will insert some extra code in our client program which will help it to detect the sandbox environment. I will be using three different checks to detect if my RAT is running in a sandbox environment or not.

The first method I'll be using is to check if RAM is more than 1 GB or not.

©IJRASET: All Rights are Reserved

```
import ctypes

class MEMORYSTATUSEX(ctypes.Structure):
    _fields_ = [
        ("dwLength", ctypes.c_ulong),
        ("dwMemoryLoad", ctypes.c_ulong),
        ("ullTotalPhys", ctypes.c_ulonglong),
        ("ullAvailPhys", ctypes.c_ulonglong),
        ("ullTotalPageFile", ctypes.c_ulonglong),
        ("ullAvailPageFile", ctypes.c_ulonglong),
        ("ullTotalVirtual", ctypes.c_ulonglong),
        ("ullAvailVirtual", ctypes.c_ulonglong),
        ("sullAvailExtendedVirtual", ctypes.c_ulonglong),
    ]

memoryStatus = MEMORYSTATUSEX()
memoryStatus.dwLength = ctypes.sizeof(MEMORYSTATUSEX)
ctypes.windll.kernel32.GlobalMemoryStatusEx(ctypes.byref(memoryStatus))

if memoryStatus.ullTotalPhys/1073741824 > 1:
    print("The RAM of this host is at least 1 GB in size. Proceed!\n")
else:
    print("Less than 1 GB of RAM exists on this system. Do not proceed.\n")
```

Fig. 1 RAM check

The second check would be of how many times it has used usb. Normally sandboxes don't use them.

```
from winreg import *
import sys

MinimumUSBHistory = 2

if len(sys.argv) == 2:
    MinimumUSBHistory = int(sys.argv[1])

HKLM = ConnectRegistry(None,HKEY_LOCAL_MACHINE)
Opened_HKLM_Key = OpenKey(HKLM, r'SYSTEM\ControlSet001\Enum\USBSTOR')

if QueryInfoKey(Opened_HKLM_Key)[0] >= MinimumUSBHistory:
    print("Proceed!")
else:
    print("Number of USB devices ever mounted: " + str(MinimumUSBHistory))
```

Fig. 2 USB history check

And the last check would be of human interaction.

```
import ctypes
import sys

dialogBoxTitle = "check human interaction";
dialogBoxMessage = "This is a sample dialog box to ensure user activity!"

if len(sys.argv) == 3:
    dialogBoxTitle = sys.argv[1]
    dialogBoxMessage = sys.argv[2]

MessageBox = ctypes.windll.user32.MessageBoxW
MessageBox(None, dialogBoxMessage, dialogBoxTitle, 0)

print("Now that the user has clicked \"OK\" or closed the dialog box, we will proceed with malware execution!")
```

Fig. 3 human interaction check

These three methods would be used to detect the sandbox environment. Once these three tests are passed our RAT will execute its malicious payload, till then it stays hidden. The above code is written in python and is for detecting windows sandbox environment.

©IJRASET: All Rights are Reserved

VI. CONCLUSION

New technology is emerging every day, both cyber security experts and hackers are using latest technology to get an edge over each other. Sandboxing technology is widely used for malware detection and prevention, so hackers search for ways to teach their malware to stay inactive in the sandbox. In this way, sandbox-evading malware can bypass protection and execute its malicious code without being detected by modern cybersecurity solutions. We have just described a method to design an advance malware which can be used by the law enforcements to spy and access the computer systems of suspicious individuals or companies.

REFERENCES

[1] Common Malware Types: Cybersecurity 101. URL: https://www.veracode.com/blog/2012/10/common-malware-types-cybersecurity-101
[2] Malware (or malicious software). URL: https://searchsecurity.techtarget.com/definition/malware
[3] Prakhar Ahlawat, Sushant Dhar, Samruddha Wagh, Amit Koppad. "Remote Access Tool Using Metasploit". International Journal on Recent and Innovation Trends in Computing and Communication, Volumne:5 Issue:4, April 2017
[4] Malware. URL: https://en.wikipedia.org/wiki/Malware
[5] Cuckoo. URL: https://cuckoosandbox.org/
[6] Malware hiding and evasion techniques. URL : https://www.andreafortuna.org/cybersecurity/malware-hiding-and-evasion-techniques/
[7] Malware VM detection techniques evolving: an analysis of GravityRAT. URL:https://www.andreafortuna.org/dfir/malware-analysis/malware-vm-detection-techniques-evolving-an-analysis-of-gravityrat/
[8] Basic Sandbox Evasion with Python. URL : https://medium.com/@AntiSec_Inc/basic-sandbox-evasion-with-python-573da582ef3
[9] Dridex Code Breaking – Modify the Malware to Bypass the VM Bypass. URL:https://cofense.com/dridex-code-breaking-modify-the-malware-to-bypass-the-vm-bypass/
[10] Dangerous malware powered by artificial intelligence. URL:http://www.securitynewspaper.com/2018/08/10/dangerous-malware-powered-by-artificial-intelligence/
[11] Most common sandbox evasion techniques. URL: https://www.apriorit.com/dev-blog/545-sandbox-evading-malware
[12] Distribution of malware under windows. URL: https://www.av-test.org/fileadmin/_processed_/a/f/csm_AV-TEST-_Distribution_of_malware_under_Windows_2018_3462cfad3f.png

©IJRASET: All Rights are Reserved