International Journal of Recent Innovation in Engineering and Research
Scientific Journal Impact Factor - 3.605 by SJIF
e- ISSN: 2456 – 2084

OFFENSIVE SECURITY ANALYSIS WITH MODULAR TROJAN

Omkar Ukirde[1], Shivanand Shinde[2], Akshay Biradar[3], Shubham kadoo[4] and Prof. Sagar Jaikar[5]

[1,2,3,4,5] *Department of Information Technology, International Institute of Information Technology, (I2IT) , Savitribai Phule Pune University, Pune, India*

Abstract - Trojans, either helpful or damaging, are conceived by their authors for many differing reasons. Whether for academic reasons, money, fame, protection, or just an attempt to illegally gain access to unauthorized information, all computer viruses bring with them side effects that are typically damaging. By exploring these motivations and studying the impacts of various viruses, it is possible to gain insight into the not so evident reasons for their creation as well as what side effects have occurred as a result of their release [2]. A machine infected with a Trojan can be made to do a multitude of things by a remote attacker. When a victim installs the malicious program, their computer will communicate silently with an attacker whenever they are logged in. It installs the modular trojan (written in Python and packaged as an executable) onto the victim's computer. The Trojan communicates with a remote server which is controlled by the attacker. Furthermore, the victim can connect to the server from anywhere. This document contains information about modular Trojan, python and details on how to create/write modular Trojan for the sole purpose of offensive security analysis.

Keywords - Modular Trojan, python, network security, coding

I. INTRODUCTION (*WHAT IS MODULAR TROJAN?)*

A modular Trojan is the malicious lines of code which can load and execute different functional modules at run-time [3]. Next, we are going to present the different types of security models [1], the different types of attackers, the different types of attacks [7], and the terminology [7].

A. Different types of security models

Offensive security

Offensive security is a proactive and adversarial approach to protecting computer systems, networks and individuals from attacks, means hacking own system before someone else does. Offensive security measures are focused on seeking out the perpetrators and in some cases attempting to disable or at least disrupt their operations.

Defensive security

It focuses on reactive measures, such as patching software and finding and fixing system vulnerabilities.

B. Different types of attackers

White hat

The white hats are the good guys. This type of attacker is the ones that run penetration tests or hacks into a system ethically in order to find the flaws the system may have. After, they share their findings with the company or network's owner without using the exploits to harm others.

Black hat

The black hats are the bad guys. This type of attacker will find flaws and holes in the system and won't notify anyone to use for their own benefit. They will exploit it themselves to harm others.

Red hat

The red hats are those who hack the system, usually Linux/UNIX platforms, in order to find flaws and make it better without any permission from anyone through open source software.

Grey hat

The grey hats are in between black hats and white hats. These are the type of attackers who will find a flaw and let the owner of the system know. At the same time, they will let the world know by publicizing it in the web so other attackers would go for an attack to harm others.

C. Different types of attacks

ARP Poisoning

The attacker gets in between the computer and the router. This way, having physical access, the attacker continues to run a man in the middle attack from this situation.

Brute Force

Brute Force is basically deciphering a password. The process is long depending on the victim's password. This attack is very noticeable because trying out the password attempts the system can tell whether the person trying to log in is actually the victim or not. 3) Dictionary Attack A dictionary attack is when the attacker uses a text file that contains various common words for passwords and uses it to apply a brute force attack on the system, but this one is faster.

DNS Poisoning

The DNS poisoning attack is when the attacker gains physical access to the computer. Once the attacker gained physical access, it can spoof the system.

Fuzzing

This attack goes hand in hand with the social engineering attack. First, the attacker uses an exploit and a payload to access gain access into the system. The attacker fools the system because he/she is impersonating the victim once the attacker has gained the access desired. 6) Network

Mapping

This attack is basically the attacker tracing the victim's steps. The attacker will scan the network for the entire system including the services the system has. This way, the attacker gains access and commit the harmful attack. 7) Password Cracking Password cracking is when the attacker cracks the victim's password through rainbow tables. 8) Session Hacking This attack happens when the attacker uses the victim's credentials to impersonate the victim and gain access to their system. An example of this attack is password hacking. The attacker finds out the password of the victim and logs in as the victim. 9) Social Engineering This attack happens when the attacker impersonates and convinces the victim of being someone of trust. This way, the attacker gathers important information or credentials from the victim to commit harmful attacks.

SQL Injection

This happens when the attacker uses injections of malicious SQL responses or scripts to gain full access to the system.

Vulnerability Scanning

The attackers, to gain knowledge about the system, do this type of scanning. They make a penetration test after finding out how many exploits are known to the system. Then, they proceed with the attack to gain full access.

Wardriving

Wardriving is an attack where the attacker has access to an encrypted access point. From there, the attacker captures the packets that are encrypted to decrypt them. After decrypting them, they gain access to the network because they find out the key necessary.

Zero Day

Zero Day is the term they have chosen to describe a new attack that no one knows about. It presents new attacks without any solution yet.

D. Terminology

Client Side Exploit

This means doing the exploit through social engineering, on the client's side of the system. 2) Trojan A Trojan, as stated earlier, is the program doing malicious activities 3) Injection

As the name suggests, is the term used to describe the addition or injection of the payloads into the system.

Module

This is referred to a tool.

Payload: It is the information about the exploit that identifies the material's destination and source and it is sent after the exploit has already run. 6) Physical Access:

It is having the computer or system's access in location. 7) Remote Access:

It is not having the physical access to the system so they access it remotely.

8) Scanner

A scanner is a tool that scans or looks up something specific throughout the system.

Server Side Exploit

This is the opposite of client side exploit; this is when the exploit is being accessed through the server.

Shellcode

The shellcode is the code that is in a payload.

Target

The target as its definition suggest is the machine or system that the attacker is focused on harming.

Vulnerability

Vulnerability is where the system is flawed. It is the location in the system that the attackers exploit.

Having a little bit of background information about the Trojan, we now proceed to discuss how modular Trojan is different from traditional Trojan.

Modular Trojan

The working of typical Trojan is execution of predefined functionalities in a single executive file written by attacker during its development. This makes the Trojan predictive and limited. There is no way of adding functionalities after deployment. In modular Trojan, we can load and execute external modules at runtime, whenever they required. It gives us additional advantage of changing functionalities of Trojan. So there's nothing someone could say about "what it does".

Why Python?

Trojan can be developed in any sort of languages but choosing python gives us a powerful, developed community of programmers. Python [8] has clean object oriented design, enhanced process control and its strong integration, all of which contribute to the increase in its speed and productivity.

II. ARCHITECTURE

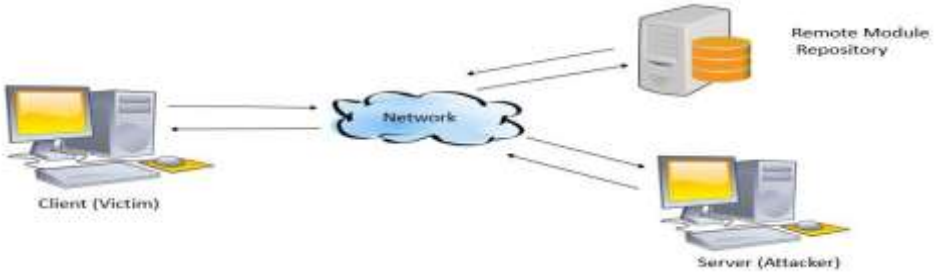

Fig 1 system architecture of modular Trojan.

III. COMPONENTS

a) Client/victim system

The target machine on which executable file will run.

```
function loadmodule(name,ip)

        moduleName = name
        remoteIP = ip
        tempfile = create tempfile
        modAddress = "http://"+remoteIP+"/moduleName"

        //Download module using python library 'urllib'
        urllib.urlretrieve(modAddress,tempfile)

        if moduleName present in cwd, then
                //load and execute downloaded module through import functionality
                __import__(moduleName)
        else
                print "module not downloaded"
```

b) Server/attacker system

The system from which attacker controls the target machine.

c) Remote module repository

The machine where modules are stored and can be called whenever required.

. **CONNECTION**

The connection between client and server is handled by socket programming. When IP address of server is unknown to the) client system, then it search for latest tweets and parse the address from the tweet.

EXECUTION

The execution of modules from remote repository can be implemented by using python import functionality. The following algorithm shows the working of the actual process.

IV. MODULES

1) Key logger

- is the action of recording (logging) the keys struck on a keyboard, typically covertly, so that the person using the keyboard is unaware that their actions are being monitored. The keylogger module is bundled with Trojan exe and it will start to capture the key strokes as soon as the Trojan is executed. The pyHook library wraps the keyboard hooks in the Windows Hooking API and stores the result in text file.

Webcam image capture

-captures user actions using webcam by recording through webcam of the system Microphone recorder

-records audio through system's microphone and stores it on local machine. It uses pyaudio and wave library functions to record sound and stores it in .wav format. Screenshot capture

-capture screenshots to know what user is doing. It uses PIL library and captures the current working screen using Imagegrab() function.

Chrome password

-displays chrome browser passwords saved by user. Chrome saves its passwords in SQLite database locally. We decrypt the data using Win32crypt library functions. Below is the screenshot of the file created containing usernames along with the passwords:

Fig 2 screenshot of chromepass.txt file Browser history

-displays chrome browser history and most visited sites. It also sorts the output according to the visit count. It uses sqlite3 library to fetch the required results.

Port scanner

-scans ports, gives output in form of open or closed. We need to provide two arguments to our custom command 'scan', ip address and ports for scanning. The result of scanning can be shown as below:

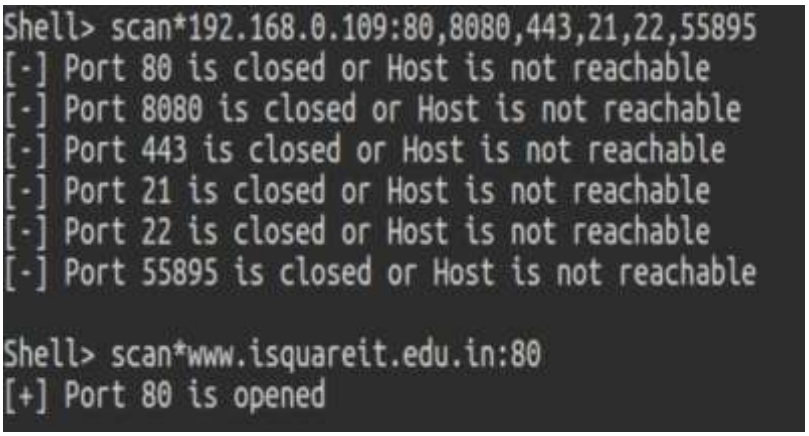

Fig 3, port scan results

Firewall management

-can disable the firewall when the module is loaded. It is capable of turning the firewall off and on based on various profiles such as private, public and domain.

Persistent

-makes program's existence permanent by creating copy and storing on local machine. It adds registry entry in windows registry to start itself at every boot time.

Admin rights check

-checks whether the user is admin of system or not. It uses ctypes python library function.

Tweet grabber

-grabs the latest tweets by given user from his twitter account. It uses python libraries like 'beautifulsoup' to parse the webpage and 're' to get the actual tweet.

Load remote module

-loads module from remote system i.e. kind of repository. The repository is the collection of different python modules. It will download the module using urllib and loads immediately. 13) File grabber

-grabs any type of file of any size from victim's

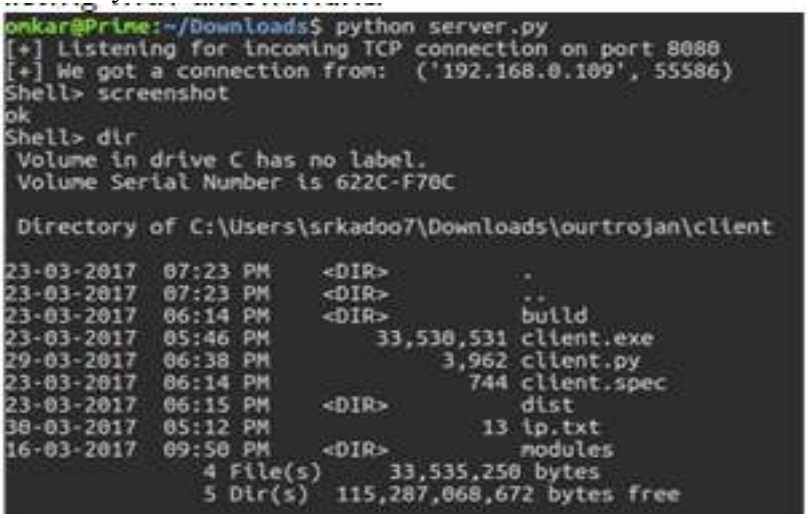

Fig 4 command output on attackers end

15) Remote Desktop protocol
-create a replica of victim's system and gives full access to do any sort of actions. The victim machine can be handled from attackers end with the capability of passing mouse clicks and keyboard strokes. [8].

V. ANALYSIS

In our approach of injecting the Trojan, we kind of social engineered the victim and asked to execute the file. Once executed, it creates copy of its files on target machine so that attacker could have entire control. This behavior leads us to think on verifying the sources of files and direct execution of programs without bothering about its developers. We use different trusted virus scanner and got positive responses too. Results are as follow:

Fig. 5, scan results from online virus scanner Every other antivirus failed to detect anything except one warning explaining about some intra-transferring of files.

Fig 6, scan results from virustotal.com

Most of the widely used antivirus failed to detect anything. Modular Trojan can be found helpful in Organization, we can monitor the usage of resources, activities performed by worker who is under suspicion of being a malicious insider by logging his activities on his workstation.

VI. CONCLUSION

There are many packages and Trojan available to help you take advantage of people's carelessness. In this case it is very handy to learn how exactly Trojans are written and how they work. You need a good understanding of system functionality and how data transfers over a network, as well a strong programming background. Once you get started writing Trojan it really becomes an easy process that allows you to take advantage of many security aspects people ignore while on internet. Since this program was a Trojan and not a virus, automatic replication was not part of the behaviour, this malicious program had to be explicitly distributed to and activated by the users of the systems. However, even though this program only infected a small target group of users, irreparable damage was left behind.

REFERENCES

[1] Hamad AL-Mohannadi, Qublai Mirza, Anitta Namanya, Irfan Awan, Andrea Cullen and Jules Disso, "Cyber-Attack Modeling Analysis Techniques: An Overview", in 2016 4th International Conference on Future Internet of Things and Cloud Workshop
[2] Patrick Carnahan, Dusty Roberts, Zack Shay, Jeff Yeary, "The motivation behind computer viruses", Georgia Institute of Technology in 2015
[3] Seth Kulakow, "NetBus 2.1,Is It Still a Trojan Horse or an Actual Valid Remote Control Administration Tool?", SANS Institute Reading Room in 2015
[4] Seifried, Kurt. "How To Hack: An Introduction." Sys Admin. November 2000 [5] Williams, Jim, "Script Kiddies: The Trojan Horse Attack" URL: http://netsecurity.about.com/compute/netsecurity , 2001
[6] Marco Giuliani and Andrea Allievi, "Carberp - a modular information stealing Trojan" in 2015
[7] Mark Bacchus, Maria A. Gutierrez, Alexander Coronado, "Creating Exploits", in Advanced Ethical Hacking Conference, 2014
[8] Mark Vasilkov, "Kivy Blueprints, chapter 5, making a remote desktop app", in 2015