Detection of Trojan horse by Analysis of System Behavior and Data Packets

Prof. Abuzneid Abdelshakour, Vamshi Krishna Gudipati, Varun Kumar, Aayush Vetwal, Anjorin Adeniyi

Department of Computer Science, University of Bridgeport, CT.

Abstract

Trojan horse is said to be one of the most serious threats to computer security. A Trojan horse is an executable file in the windows operating system. These executable files will have certain static and runtime characteristics. Multiple windows system process will be called whenever a Trojan horse tries to execute any operation on the system. In this paper a new Trojan horse detection method by using windows dynamic link libraries to identify system calls from a Trojan horses is explicated. Process explorer is used to identify the malicious executable and to determine whether it is a Trojan or not. Further, an attempt is made to study the network behavior after a Trojan horse is executed using wire shark.

Introduction

Problem Statement

Due to the increased numbers of internet users across the globe and the number of network connections and facilities available now, there is increased risks to internet networks in form of malwares, worms, trojan horses that are unknown to them. The internet is flexible in such a way that it provides as much information that users need as accurate as possible.

The rate of attacks have risen progressively in the last few years as so many attackers attempt to send trusted looking files over the network which looks so enticing to the users and once installed is a rootkit which sends personal information of the users to the different attackers unknown to the users and this files or trojan horses are smart enough to bypass the antivirus which the user relies on to detect any form of suspicious files or applications installed.

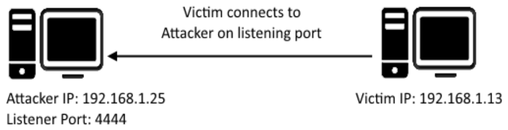
A more flexible method of detecting trojan horses other than the static method which takes different kinds of trojan horses as input in a database which contains intrusion signatures of different trojan horses, the downsize of this is that new crafted scripts or trojan horses which are not available on the database are not regarded as trojan horse and thereby outsmarts the antivirus. The new method that will be proposed in this paper is to dynamically detect trojan horses without having to check a collections of signatures on the database.

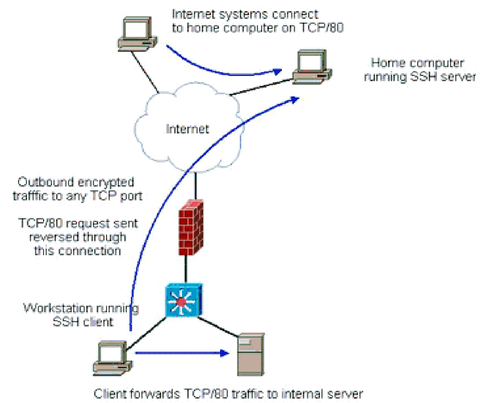
The Trojans have very distinct run time characteristics when compared to traditional executables. These characteristics can be used to determine whether the program is Trojan or not. Different Trojans have different type of characteristics depending on the motive of the creator. If the hacker just want monitor the data flow on the client or victims computer then there will not be any sever effect on the victim computer. So victim may never know that a Trojan is running on his system as it is not interfering with his operations. Some of the Trojan may send some secure information to its creator over network. This may choke the network bandwidth, so these Trojans can be detected by monitoring the network packets. Although this is not a simple task, there is definite chance to find the Trojan. Some of the Trojans will have immediate effect on the system performance, so that they can be detected instantly. But these type of Trojan are very rare as no one wants to get caught immediately until and unless their purpose is solved.

Trojan can be described as a simple executable file in windows operating system. But it has some of its properties very different from the general executable files. We can use these properties to detect the presence of a Trojan horse. The Trojans are always not active on the client system. For a Trojan to work, the client should run it at least once on his system. From then the Trojan starts doing the work for the exploiter like sending the data to the listener, and providing remote access.

Trojan Horse Creation

All these attacks can be done using different payloads and different approaches. Here, we show you all the characteristics of the Trojan as stated above and eventually get complete access over the remote computer[5]. So we first considered to exploit a remote computer using the Backtrack r3 operating system, an Ubuntu Linux distribution that focuses on security aimed at digital forensics and penetration testing. With the basic Commands on the Metasploit Framework (Command line Interface), we have exploited a remote computer. BackTrack is based on Linux environment. It is a penetration testing platform that supports penetration testers, bug hunters, and security professionals to perform assessments in a purely native environment dedicated to hacking. Irrespective of how it is being using BackTrack, one may install BackTrack, boot it from a Live DVD, flash drive, the penetration distribution has been customized down to kernel configuration, every package, every assessment tool, scripts used for exploitation and patch solely for the purpose of the penetration tester. BackTrack is intended for all kinds of users from the most savvy security professionals to early rookies to the information security field. It promotes a robust and efficient way to find and update the largest database of security tools collection to-date. The user community range from highly skilled penetration testers in the field of information security field, government entities, information technology, security enthusiasts, and individuals who are very new to the security community. Whilst it is so easy to use and carry out exploitation, it has been a blind folded job for anyone with the minimum knowledge on how to use it and carry out an exploit. We have created a backdoor by injecting a reverse meterpreter payload onto an application that we want to use for exploitation. Here, an exe file is used to exploit into the target computer. This is a reverse TCP protocol for creation of the Trojan horse. Below are the steps we used to establish a connection with the remote computer.

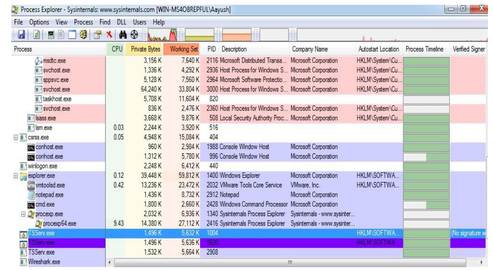




Detection by Process Explorer

We have proposed several methods to detect a Trojan in the system. Whatever the process is, the principal goal is to segregate a suspicious process or program out of several others, based on the behaviors that Trojan or suspicious file relatively shows. Before analyzing or detecting a Trojan horse, it is necessary to figure out the objects of Trojan horse operation. Trojan horses usually operate on registry, file, port, process, system service and other I/O interfaces like keyboard, webcam etc. Based on these objects that Trojans act upon, we now know where to monitor the activity.

The basic implementation is to code a function that is invoked every time a certain system process in windows is started. Hooks were distributed by Microsoft predominantly to help programmers to straighten out the errors of their applications, but they can be put to use in many different ways[6]. However, using API hooking and DLL injection to detect what a certain foreign harmful process is doing in our system is a complex matter, because every time we inject dll to a process, we are inflicting with the system memory that is otherwise always reserved for that particular process. This could bring several problems while using the system simultaneously.



Packet analysis by Wire Shark

Wireshark is a graphical user interface based, packet analysis tool which goes through the phase of collecting, converting and analyzing of captured data from the network.

Collection Phase: In this phase, the packet analysis tool assembles the raw binary data from the wire. Generally, this is carried out by switching the selected network interface into promiscuous mode. In this mode, the network card can listen to all the network segment, not only the traffic that is addressed to it.

Conversion Phase: During this phase, the captured binary data is converted into a readable form. This is where the most advanced command-line packet sniffers stop. At this point, the network data is in a form that can be interpreted only on a very basic level, leaving the majority of the analysis to the end user

Analysis Phase: This is the third and the final phase which implicates the analysis of the readable form data. This is by far the most important phase which helps in better understanding of the network activities. The packet analyzer takes the captured network data, verifies its protocol based on the material extracted and begins to analyze the protocol's specific features in accordance with the filters that are applied in the analysis.

Conclusion

We considered to detect software trojan horses in operating system like windows, mac, linux flavors etc, thereby solving a major problem in networks and a lot of methods were randomly looked into. More work has been done on detecting malwares, worms, and hardware trojan horses

In this paper, we made use of project explorer which is really useful in windows operating system as windows is generally believed to be less secured compared to other operating systems like mas os x, linux, unix etc.