Final Project
Comp 8505
Report

Aadi Bisht
December 5, 2023

Contents

Purpose

This assignment is to create a rootkit that has the functionality of a starting and stopping the keylogger, transferring the keylog file generated by the keylogger, watching a file and directory, stopping the file and directory watching, transferring files to and from the victim, running a program on the victim's machine and uninstalling the files and disconnecting the commander from the victim.

All the communication between the victim and commander uses encryption and a covert channel. We used the ID field of the IP Header as the covert placeholder.

Keylogger and Rootkits

A keylogger, short for "keystroke logger," is a type of malicious software or hardware device designed to record and monitor the keystrokes typed on a computer or mobile device. Keyloggers are typically used for nefarious purposes, such as stealing sensitive information like passwords, credit card numbers, or personal messages.

A rootkit is a type of malicious software or set of tools that is designed to gain unauthorized access to a computer or system while hiding its presence and activities from users and security software. Rootkits are named after the "root" superuser account in Unix-based operating systems, which has complete control over the system, and they are particularly adept at hiding themselves deep within a computer's operating system.

File and Directory Watching

File and Directory watching are techniques used in computer programming and operating systems to keep track of changes to files and directories in a file system. These techniques are commonly used in various applications and services to respond to changes in files and directories, such as updating content, triggering events, or automating tasks.

What is a Covert Channel?

A covert channel is a communication channel that is used for transferring information in a way that is hidden or concealed from normal security mechanisms or protocols. These channels are typically exploited by malicious actors to bypass security controls and transmit information in a manner that is not intended or authorized.

Covert channels can take various forms, but they generally involve manipulating or abusing existing resources or mechanisms within a computer system or network.

Some common examples of covert channels include:

Timing Channels: These channels exploit variations in the timing of system events to convey information. For example, the time it takes to perform a specific operation might be manipulated to encode binary data.

Storage Channels: These channels use shared resources, such as files, memory, or disk space, to transmit information. By altering the state of these resources, data can be hidden and retrieved.

Network Channels: Covert network channels use unauthorized methods to transmit data over a network, often bypassing network security measures. For instance, using unused or low-priority network protocols to transmit hidden data.

Side-Channel Attacks: These channels focus on exploiting unintentional emissions from a system, such as electromagnetic radiation, power consumption, or even acoustic signals, to infer information about its operation.

Uses of Covert Channels

The uses of covert channels can be both malicious and non-malicious:

Malicious Uses:

Data Exfiltration: Attackers may use covert channels to exfiltrate sensitive or confidential data from a compromised system, evading detection and security controls.

Evasion of Security: Covert channels can be used to bypass security mechanisms and access restricted resources or systems.

Command and Control: Malicious software (e.g., malware) may use covert channels to establish communication with a command-and-control server, allowing attackers to remotely control compromised systems.

Non-Malicious Uses:

Research and Testing: Security professionals and researchers may use covert channels to assess the vulnerabilities and weaknesses of systems, networks, and applications.

Educational Purposes: Covert channels can be used in educational settings to teach students about security and ethical hacking by demonstrating how these channels work and how to defend against them.

What Fields to use for Covert Channels

Best Fields:

IPv4:

Identification: Often unused or predictable, can encode data.

Time to Live (TTL): May carry data and be modified within limits.

IPv6:

Flow Label: Typically, unused and can carry arbitrary data.

Traffic Class: Bits can be modified to encode covert data.

TCP:

Sequence Number: Can be manipulated without disrupting the flow.

Window Size: Can be adjusted without disrupting the connection.

UDP:

Length: May be altered without affecting the message.

ICMP:

Type/Code: Some codes may be unused or carry hidden data.

Worst Fields that may not be able to manipulated to carry data:

IPv4/IPv6:

Source/Destination Address: Easily detected, and their changes disrupt routing.

Protocol Number: Critical for proper protocol identification.

TCP:

Flags (e.g., SYN, ACK): Critical for proper connection establishment.

Source/Destination Port: Easily detected, and changes disrupt connectivity.

UDP:

Source/Destination Port: Easily detected, and changes disrupt connectivity.

ICMP:

Identifier: Typically used for error message pairing.

Sequence Number: Used to order and pair responses with requests.

Checksum: Altered payload can be complemented in checksum.

Requirements

Commander

Task	Status
Able to start the keylogger	Completed
Able to stop the keylogger	Completed
Able to transfer the key log file	Completed
Able to disconnect from the victim	Completed
Able to connect again to the victim	Completed
Able to start the file/directory watcher	Completed
Able to stop the file/directory watcher	Completed
Able to transfer the modifications made in the file/directory	Completed
Able to move files to deleted directory if the file/directory being watched is deleted	Completed
Able to create the directory with the IP address such that the path is: $/downloads/< ip\_addr >$	Completed
Able to create the delete directory when a file/directory is deleted	Completed
Files are overwritten when transferred from the same IP address	Completed
Able send command to run a program on the victim	Completed
Able to transfer a file to the victim	Completed
Able to receive a file from the victim	Completed
Able to send command to uninstall everything on the victim	Completed
Able to encrypt and decrypt data when transferring over from the sender to the receiver	Completed
Able to send port knock packets to the victim	Completed
Able to use covert channel for all the transmissions	Completed

Victim

Task	Status
Able to run forever until timed out/signal interrupt	Completed
A new file is created when the keylogger is started	Completed
Able to transfer the file to the commander and deletes the file after transfer is completed	Completed
Able to run as root	Completed
The process's name is changed to a common name (PROGRAMMATICALLY)	Completed
Keylogger can log each keystroke made and also capture special characters like "!"	Completed
Key log File is closed when the keylogger is stopped	Completed
A new file is created when the keylogger is started	Completed
Able to transfer the file to the commander and deletes the file after transfer is completed	Completed
Able to watch for events: creation, deletion, modification of the file/directory	Completed
Able to transfer the file/directory to the commander when an event is triggered	Completed

Able to stop the file/directory watcher	Completed
Able to run the command sent	Completed
Able to receive a file transferred from the commander and store it in the root of working directory	Completed
Able to send a file to the commander	Completed
Able to uninstall everything and exit out when received a command to uninstall	Completed

Platform

Linux based OS (Fedora, Kali, etc)

Languages

Python