Malware Analysis

MALWARE BEHAVIOUR

COMPILED BY: DHARMESH DAVE | ASST. PROF. | NATIONAL FORENSIC SCIENCES UNIVERSITY

Downloaders and Launchers

- Downloaders simply download another piece of malware from the Internet and execute it on the local system. Downloaders are often packaged with an exploit.
- Downloaders commonly use the Windows API **URLDownloadtoFileA**, followed by a **call** to **WinExec** to download and execute new malware.
- A launcher (also known as a loader) is any executable that installs malware for immediate or future covert execution. Launchers often contain the malware that they are designed to load.

Backdoors

- A backdoor is a type of malware that provides an attacker with remote access to a victim’s machine.
- Backdoor code often implements a full set of capabilities, so when using a backdoor attackers typically don’t need to download additional malware or code.
- Backdoors communicate over the Internet in numerous ways, but a common method is over port 80 or 443 using the **HTTP or HTTPS** protocol.
- Backdoors come with a common set of functionality, such as the ability to manipulate registry keys, enumerate display windows, create directories, search files, and so on.

Reverse Shell

- A reverse shell is a connection that originates from an infected machine and provides attackers shell access to that machine. Reverse shells are found as both stand-alone malware and as components of more sophisticated backdoors.
- Once in a reverse shell, attackers can execute commands as if they were on the local system.
- **Windows Reverse Shells**
- Two implementation methods using cmd.exe:
  - Basic
  - Multithreaded

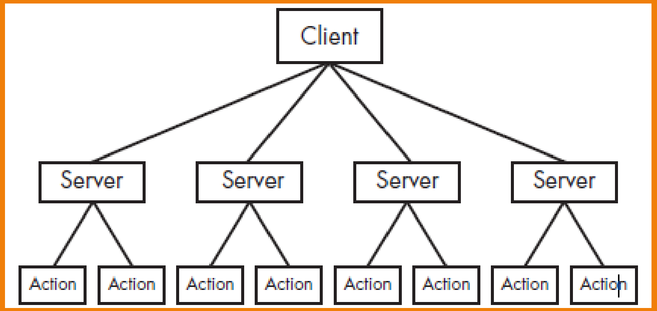
Reverse Shell

- **Basic**
- It involves a call to CreateProcess and the manipulation of the STARTUPINFO structure that is passed to CreateProcess.
- First, a socket is created and a connection to a remote server is established. That socket is then tied to the standard streams (standard input, standard output, and standard error) for cmd.exe.
- CreateProcess runs cmd.exe with its window suppressed, to hide it from the victim.

Reverse Shell

- **Multithreaded**
- The multithreaded version of a Windows reverse shell involves the creation of a socket, two pipes, and two threads.
- This method is sometimes used by malware authors as part of a strategy to manipulate or encode the data coming in or going out over the socket.
- **CreatePipe** can be used to tie together read and write ends to a pipe, such as standard input (stdin) and standard output (stdout).
- The CreateProcess method can be used to tie the standard streams to pipes instead of directly to the sockets. After CreateProcess is called, the malware will spawn two threads: one for reading from the stdin pipe and writing to the socket, and the other for reading the socket and writing to the stdout pipe.

RATs



- A remote administration tool (RAT) is used to remotely manage a computer or computers. RATs are often used in targeted attacks with specific goals, such as stealing information or moving laterally across a network.
- The server is running on a victim host implanted with malware. The client is running remotely as the command and control unit operated by the attacker.
- The servers beacon to the client to start a connection, and they are controlled by the client. RAT communication is typically over common ports like 80 and 443.

Botnet

- A botnet is a collection of compromised hosts, known as zombies, that are controlled by a single entity, usually through the use of a server known as a botnet controller.
- The goal of a **botnet** is to compromise as many hosts as possible in order to create a large network of zombies that the botnet uses to spread additional malware or spam, or perform a distributed denial-of-service (DDoS) attack.
- Botnets can take a website offline by having all of the zombies attack the website at the same time.

RAT vs Botnet

- There are a few key differences between botnets and RATs:
- Botnets have been known to infect and control millions of hosts. RATs typically control far fewer hosts.
- All botnets are controlled at once. RATs are controlled on a per-victim basis because the attacker is interacting with the host at a much more intimate level.
- RATs are used in targeted attacks. Botnets are used in mass attacks.

Credential Stealers

- Attackers often go to great lengths to steal credentials, primarily with three types of malware:
- Programs that wait for a user to log in in order to steal their credentials
- Programs that dump information stored in Windows, such as password hashes, to be used directly or cracked offline
- Programs that log keystrokes

Credential Stealers

- **Hash Dumping**
- Attackers try to grab these hashes in order to crack them offline or to use them in a pass-the-hash attack. A pass-the-hash attack uses LM and NTLM hashes to authenticate to a remote host (using NTLM authentication) without needing to decrypt or crack the hashes to obtain the plaintext password to log in.
- Pwdump and the Pass-the-Hash (PSH) Toolkit are freely available packages that provide hash dumping.

Credential Stealers

- **Hash Dumping**
- Pwdump is a set of programs that outputs the LM and NTLM password hashes of local user accounts from the Security Account Manager (SAM). Pwdump works by performing DLL injection inside the Local Security Authority Subsystem Service (LSASS) process.
- Standard pwdump uses the DLL lsaext.dll. Once it is running inside lsass.exe, **pwdump calls GetHash**, which is exported by lsaext.dll in order to perform the hash extraction.
- This extraction uses undocumented Windows function calls to enumerate the users on a system and get the password hashes in unencrypted form for each user.

Credential Stealers

- **Keystroke Logging**
- Keylogging is a classic form of credential stealing. When keylogging, malware records keystrokes so that an attacker can observe typed data like usernames and passwords.
- Windows user-space keyloggers typically use the Windows API and are usually implemented with either **hooking** or **polling**.
- Hooking uses the Windows API to notify the malware each time a key is pressed, typically with the **SetWindowsHookEx** function.
- Polling uses the Windows API to constantly poll the state of the keys, typically using the **GetAsyncKeyState** and **GetForegroundWindow** functions.

Credential Stealers

- **Keystroke Logging**
- The GetAsyncKeyState function identifies whether a key is pressed or depressed, and whether the key was pressed after the most recent call to GetAsyncKeyState (program checks the SHIFT and CAPS LOCK keys).
- The GetForegroundWindow function identifies the foreground window—the one that has focus—which tells the keylogger which application is being used for keyboard entry (Notepad or Internet Explorer, for example).

Credential Stealers

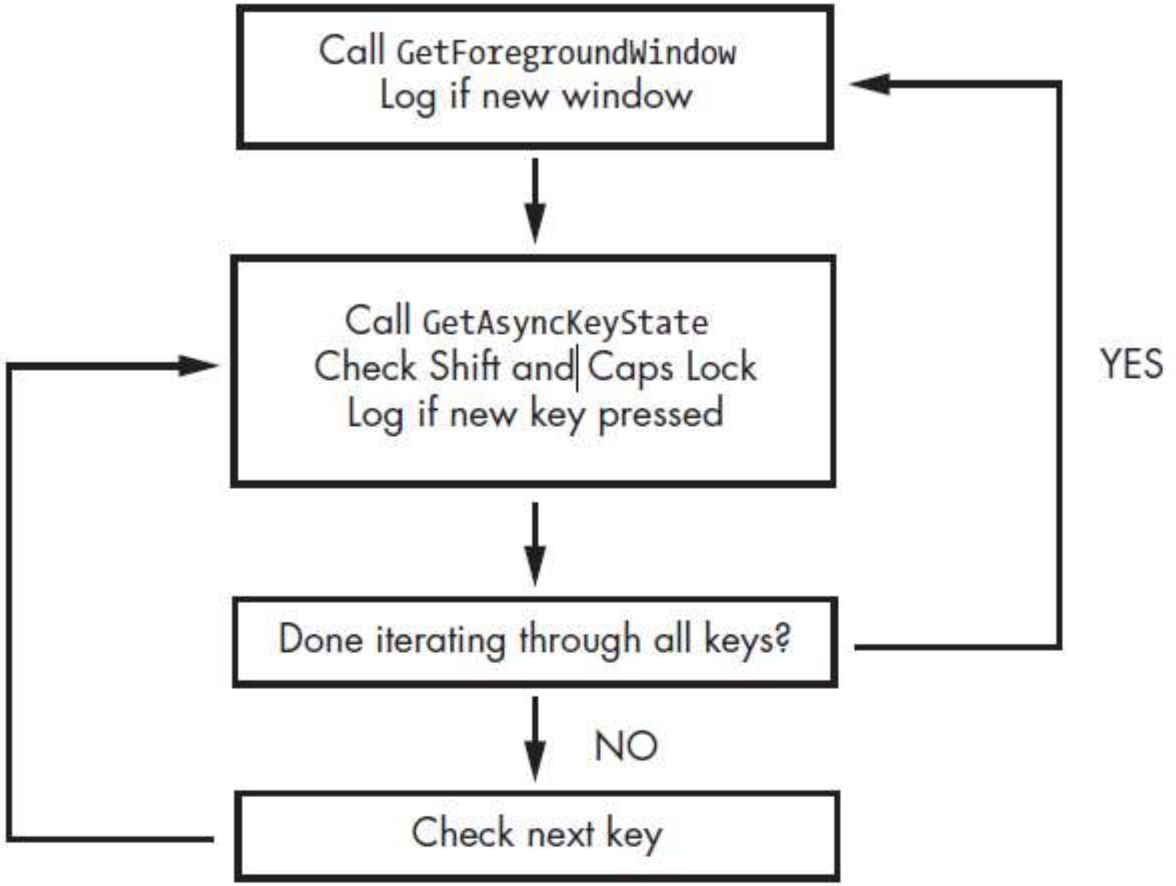


Figure 11-3: Loop structure of `GetAsyncKeyState` *and* `GetForegroundWindow` *keylogger*

Persistence Mechanisms

- **Windows Registry**
- RunKey (Autoruns):
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- AppInit_DLLs: AppInit_DLLs are loaded into every process that loads User32.dll, and a simple insertion into the registry will make AppInit_DLLs persistent.
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows
- Winlogon Notify: Malware authors can hook malware to a particular Winlogon event, such as logon, logoff, startup, shutdown, and lock screen. This can even allow the malware to load in safe mode.
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\

Persistence Mechanisms

- **Windows Registry**
- SvcHost DLLs: Installing malware for persistence as an svchost.exe DLL makes the malware blend into the process list and the registry better than a standard service.
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost
- **DLL Load-Order Hijacking**
- DLL load-order hijacking is a simple, covert technique that allows malware authors to create persistent, malicious DLLs without the need for a registry entry.
- **Example:** If a malicious DLL named *ntshrui.dll* is placed in /Windows, it will be loaded in place of the legitimate DLL. The malicious DLL can then load the real DLL to ensure that the system continues to run properly.

Persistence Mechanisms

- **DLL Load-Order Hijacking**
- The default search order for loading DLLs on Windows XP is as follows:
  - *1. The directory from which the application loaded*
  - *2. The current directory*
  - *3. The system directory (the GetSystemDirectory function is used to get the*
  - *path, such as …/Windows/System32/)*
  - *4. The 16-bit system directory (such as …/Windows/System/)*
  - *5. The Windows directory (the GetWindowsDirectory function is used to get*
  - *the path, such as …/Windows/)*
  - *6. The directories listed in the PATH environment variable*

References

- Practical Malware Analysis by Michael Sikorski