Blue's Clue: Tool Against USB Device Intrusion

Yu Xiang Ng, Han Zhong Tan, Kian Yun Tan, Yi Ren Tan
Infocomm Technology Cluster
Singapore Institute of Technology
Singapore
{2101575, 2103027, 2103012, 2100824}@sit.singaporetech.edu.sg

***Abstract*—This paper presents the anti-forensic tool that the team has developed to counter Universal Serial Bus (USB) based flash drives that may be inserted into the users' Windows-based system without the users' knowledge or consent. These ubiquitous USB flash drives can be used for forensic purposes such as retrieving and analysing data or nefarious purposes such as spreading malwares. The report will cover how Windows operating systems interact with USB flash drives and also highlight the functionalities that our tool will have and how each function would work together to successfully prevent unauthorised USB flash drive intrusion.**

***Keyword*—Universal Serial Bus, USB, Windows, forensic, anti-forensic, flash drive**

I. Introduction

Universal Serial Bus (USB) technology was developed in the mid 1990s and the flash drives were subsequently introduced during the turn of the century, these early flash drives were only 128 megabytes (MB) in size [1]. Today, these flash drives have become ubiquitous in our daily life and have gotten cheaper, smaller in physical size and larger in terms of storage size. These characteristics have made flash drives to be the ideal choice for attackers to infiltrate computing devices or forensic investigators to retrieve data from these devices.

As cybercrime rate rises over the years, law enforcement agencies have been admitting electronic devices as evidence in courts, thus it is essential for malicious actors to guard their devices from being accessed by forensics investigators. For anti-forensics purposes, our team proposes developing a tool that tries to divert the investigator's attention. Our tool is named Blue's Clue, a reference to the kids' show where a dog will leave clues for hosts to figure out the plan for the day. The tool's function is to covertly delay the investigation as long as possible. We have come across existing tools such as usbkill and silk-guardian, that immediately try to wipe the device's entire memory when it detects a rogue USB drive.

II. Background Research

A. Literature Review

In a research into evidence analysis on USB flash drives [2], the authors gave an overview on the usb drive and its usage by suspects. It gave a detailed technical explanation on its interaction with Windows registry and retrieving vital information about the USB drive. It also mentioned the tools used by investigators to bypass the default security configurations. The authors concluded that the USB analysis tool will be of importance toward USB drive analysis in digital forensics.

When it comes to identifying the USB flash drive, studies conducted on USB flash drives show that unique identity trails were left behind in Windows registry settings when a USB flash drive is inserted into a Windows PC [3]. These encompass information such as Device Descriptor, Configuration Descriptor, Interface Descriptor, Endpoint Descriptor and HID Descriptor. These descriptors can be used to whitelist flash drives possessed by the user, and any other flash drive not whitelisted would be a rogue device.

In another paper regarding the device "USB Killer" [4], the authors examine the usage of USB drives, ranging from storing sensitive information to deploying malicious programs. The USB Killer is often used to distribute malicious payload under the pretence of testing purposes. The author concludes that USB-based attack is one of the most effective methods that malicious actors could implement to destroy numerous devices efficiently.

B. Existing Tools and Solutions

Some of the tools we have identified are for destructive purposes. Firstly, USBkill [5], a program designed for Linux and macOS would perform actions such as shutting down the machine or performing RAM and swap wiping when irregular USB activity is detected. Another notable tool is Silk Guardian [6], which is disguised as a Linux kernel driver. Thirdly, xxUSBSentinel [7], which runs on windows with functionality such as logging USB events and creating custom USB keys for fast shutdown. Lastly, usbdeath, which is written in bash, will power off the computer based on the rule set [8].

One attack method is the keystroke injection attack [9]. Notably, Rubber Ducky uses the DuckyScript programming language to execute a prewritten keyboard sequence. Moreover, the open-source toolkit, Duck Toolkit NG can be utilised to create payloads for exploitation purposes on various Operating Systems, such as Windows, Linux, and Mac OSX [10]. Users can also select pre-built payloads from the toolkit.

Another attack method is the keyboard and mouse attack. Evilduino [11], is a hacking tool that uses Arduino microcontrollers that masquerade as a mouse and keyboard to perform cursor movements on the host device based on a pre-written script. Some of the malicious actions it is able to perform are enabling a remote desktop then adding a firewall policy to allow RDP and creating a new account with admin privileges.

C. Background Technologies

The team has chosen Python as the programming language to develop the solution due to the vast amount of modules and libraries available. Libraries such as 'time' and ‘datetime’ help the team in managing different aspects of time. ‘Time’ library was used to manipulate the system's clock while datetime was used to easily modify and display time [12].

Windows Management Instrumentation (WMI) is Microsoft’s way of providing a Common Information Model (CIM) for getting information about the computer system. We would be using it to gather the amount of disk drives that are in the computer, as well as their serial number to identify them. Fortunately, Python has a WMI module,WMI, to reduce the complexity of communication between Python and the WMI application programming interface (API) [13].

PyCryptodome is a python module that enables the usage of cryptographic functions. The symmetric-key algorithm, Advanced Encryption Standard (AES) was employed in our tools for file encryption. PyCryptodome was used to generate the necessary password salt and cipher key [14].

Filedate is a python library that allows manipulation of a file’s date which includes copying file dates from one file to another, keeping file dates and setting file dates based on the file name or manually selecting the file that the user wishes to edit. This library would be used to edit the date of files that are created randomly [15].

Faker is a Python module that generates fake data from a set of data that comes along with its installation. It is used to fill in fake text to make a file content legitimate. We would use it to create a fake incriminating file that would be used to lure the investigator [16].

III. Real World Cases

There have been cases of cyber criminals arrested on the spot with their computers still turned on, which enables the police on-site to apprehend them with the computer displaying the information to incriminate them. Depending on the law of the nation where they were arrested, they could refuse to surrender their computer password, unless the severity of the crime committed could void that privilege. A notable example would be the case of Ross Ulbricht, the founder of “Silk Road”, an online black market site on the dark web that provides illegal goods and services to buyers. During his arrest, the agents from U.S Drug Enforcement Administration were able to split him from his laptop, while another agent immediately inserted a USB drive into it. The USB drive has a software that copies key files automatically [17].

A computer worm discovered in 2010, Stuxnet, was widely touted to be the first cyberweapon in the world. Stuxnet was a computer worm designed by an unknown state actor that was used to target Iranian nuclear facilities. The facilities were air-gapped, which meant it was not connected to the internet. Thus it was believed that the medium of attack of which the computer worm spread was via an infected flash drive that was smuggled in by bad actors [18].

Alternatively, the possibility of social engineering attacks such as a malicious actor attempting to steal information in your computer by either adding or copying documents to or from your computer via flash drive while you might not be looking. An example would be your close friend attempting to get your file for an assignment due in a few days from your computer without your permission.

IV. Proposed Solution

Our proposed solution will allow users to whitelist their own USB devices. If any unlisted device were detected, a program will immediately generate and encrypt random files. This is to divert the attention of forensics investigators, making the forensics procedure a time-consuming and uphill task.

Our tool differs from existing iterations, which mostly utilises the USB-on-a-lanyard technique, where a whitelisted USB is plugged into the machine at all times, acting as a key. When this whitelisted USB is forcibly removed, a program will be executed to destroy the machine. Another existing solution is regarding the planting of malware via USB drive. In contrast, our proposed solution focuses on anti-forensics activities and also includes destructive purposes such as RAM wiping and program melting etc, making digital forensics intractable.

Our project will be utilising Python as our main programming language. Currently, all the solutions we have researched are mostly written in Python, except for xxUSBSentinel which is written in C++. Python has multiple frameworks and libraries that we could refer to.

V. Detailed Solution Explanation

Our Solution would be to be designed for Windows Operating System (OS), it is a Python script that is running behind the scene when the computer startup. In order for the user’s desired USB drive to be whitelisted, it would have to be plugged in before the computer startup.

Once the user is on the home screen, they would be free to remove it as the script that is running behind has taken the drive’s serial number as part of the whitelisted drive. The script would be polling every one second to detect if there is any new drive being plug-in. In the event that the user was to insert the USB drive that was previously whitelisted or try to take the whitelisted USB drive out, the script would just continue to run without taking any action.

However, if there was a USB drive that was not whitelisted, the script would immediately trigger the following actions in sequence. Firstly, the script would look for the file that the user stated, then encrypt its content. If it could not find the file, it would generate a text file with random text on it and encrypt the content. Secondly, the encrypted file would be copied to a subdirectory and have its name, file extension and metadata changed. The actions mentioned above would also be done for the original file to hide its contents and act as false evidence.

Last but not least, the PC would then be forced into the Blue Screen Of Death (BSOD) after a short while, which would require the user to restart the Computer. Initially, ‘SetShutdownPrivilege’ is granted to control system power state. Thereafter, the function ‘NtRaiseHardError’ is called to invoke hard error. A hard error often happens when a catastrophic event such as hardware failure occurs, which in turn causes a system crash or BSOD. This is done without any external program or administrator privilege as the BSOD follows the same methodology used by malware creators to trigger such BSOD [19].

VI. Solution Reasoning

The inspiration for our solution was spurred by the arrest of Ross Ulbricht as mentioned previously, where his laptop was taken away from him and the agent uses a purpose-built USB drive that contains a software that extracts key files when inserted. As there was no public information on how the software works in the flash drive, the team assumed that by stopping the flash drive, the data retrieval process would be effectively terminated.

Therefore, the team has designed our solution around this event and with the information that is available online. There would be assumptions involved due to the scarcity of information. Additionally, it might not be applicable in many real-world cases due to the procedures the investigating body employs or the law against such action.

The reason behind our decision to have our solution to only work on Windows OS was due to its large market share as a computer OS worldwide, which would infer that there are a lot of users using this OS as their preferred daily use. This tool will be easily accessible as Windows OS is not as complicated to set up compared to other available OS in the market [20]. Another reason is due to the existence of other similar tools that were only operable in other OS.

We have to assume that the key file that the software extract would most probably be files that fit the following criteria. First, files with extension types such as .docx or .txt, these files could be used to document sensitive information. Secondly, files that fit in a certain keyword criteria in its file name. Thirdly, files that have certain metadata changes such as date modified or date created. Lastly, files that are encrypted or zip files with passwords. We designed our Python script to go with all three possibilities and have the files encrypted and be copied over with the changes made to its name, file extension and metadata. Assuming the investigating team was to be able to get hold of those "suspicious" files and attempt to decrypt it, which would be time-consuming, they would not be able to yield any result as it is acting as a facade.

Our Python script would intentionally trigger BSOD as BSOD will cause Windows to crash and stop working, which would halt any ongoing process. This would also lead to data losses as programs were unable to save their data before this state. The computer can be restarted but would require re-authentication as the user might have set up a password for secure login. Depending on the nation's law and the severity of the alleged crime, the user has the right not to give up the password or assist the investigator[21], but there are countries where the law enforcement can access without the user's right, an example would be Singapore[22].

VII. Solution Requirements

The first requirement is that the user's computer must be running on the Windows operating system. The second requirement is that Python must be installed, as the solution uses Python code to achieve our desired result. It also uses base Python libraries that come with Python when you install it on your computer., There are only three external libraries that will be needed for us to execute the script function. Firstly, the WMI module, which is a lightweight wrapper built on top of pywin32 extensions. It would be used to talk to the WMI (Windows Management Instrumentation) API, which would assist us in getting all the drives' serial number that is in the computer. Secondly, the filedate module, which would be used to change the file metadata date information. Thirdly, the Faker module, which would generate random text to the newly created text file if the user stated a file that does not exist. Lastly, the PyCryptodomex module, a self-contained Python package of low-level cryptographic primitives, which would be used to perform encryption on the file.

We will then have to create a specific function for each action to run at an appropriate sequence without any error.

VIII. Algorithm Pseudo Code

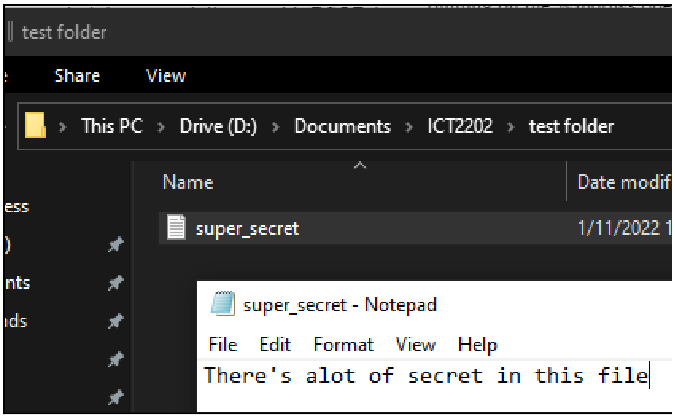
The Pseudo code shown below demonstrates the main function that would perform the following task, file encryption, filename and file extension scrambling, copy to another directory and paste in random subdirectories and lastly followed by triggering BSOD:

```
driveList = list of drive
encExtList = list of possible file extension
encNameList = list of possible file name
srcPath = original file source
password = encryption password
newPath= directory where new file should be located
ntdll= manipulate C data type
setShutDownPriviledge = privilege code


Function watch_drives():
  While condition is true:
    drives  = current list of drive gather every second
    If (drives is same as drivelist):
      Do nothing
    Else if ( drive has lesser than drivelist):
      Check if drive is a subset of drivelist:
        Do nothing
    Else:
      Stop this process from repeating:
        File encryption with password and scrambling with
encExtList and encNameList  at srcPath, follow by copying
over the newPath
        Pause for a short time to let previous process finish
        Trigger BSOD with ntdll and setShutDownPriviledge
   Pause for every second
```

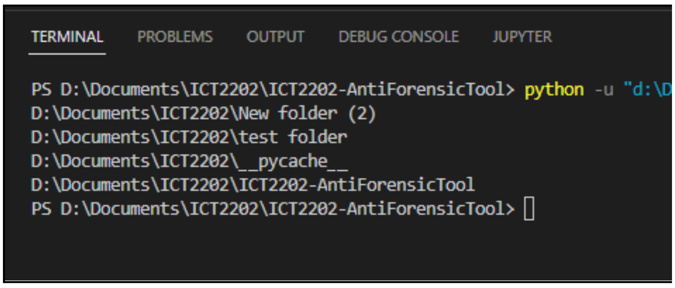
IX. Solution Result

To demonstrate that our solution does work as we have intended, we have created a text file named super_secret in the following directory shown in figure 1.



```
There's alot of secret in this file
```

Fig 1 - The target file that the script shall execute its function on



```
PS D:\Documents\ICT2202\ICT2202-AntiForensicTool> python -u "d:\D
D:\Documents\ICT2202\New folder (2)
D:\Documents\ICT2202\test folder
D:\Documents\ICT2202\__pycache__
D:\Documents\ICT2202\ICT2202-AntiForensicTool
PS D:\Documents\ICT2202\ICT2202-AntiForensicTool> ▯
```

Fig 2 - The subdirectories that was randomly generated to copy the new file toward

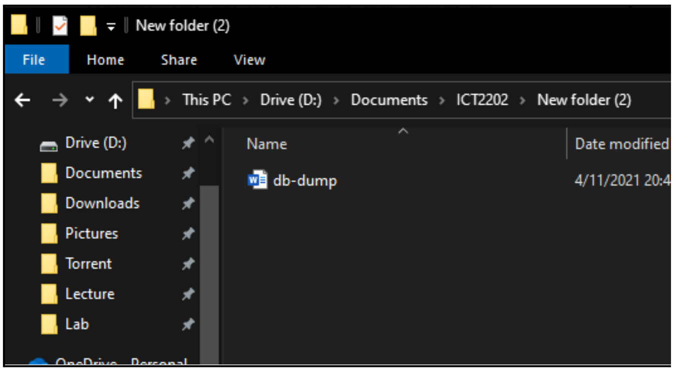
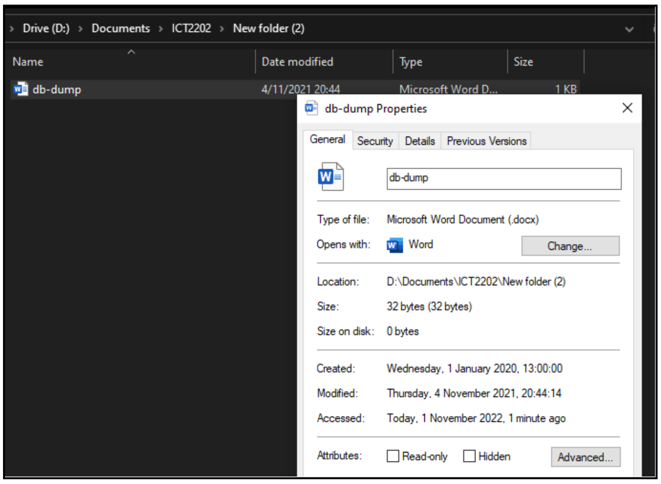


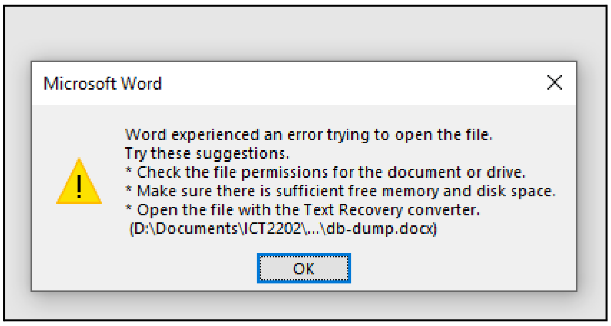
Fig 3 - The new file has been created and copied toward to one of the subdirectories

Next, the file that was shown previously in figure 1 has its content being encrypted. As shown in figure 2 and figure 3, multiple copies were made from it and placed in random subdirectories, followed by having their name and file extension changed to a name and file extension that was randomly selected from a predefined list.



	db-dump
Type of file:	Microsoft Word Document (.docx)
Opens with:	Word
Location:	D:\Documents\ICT2202\New folder (2)
Size:	32 bytes (32 bytes)
Size on disk:	0 bytes
Created:	Wednesday, 1 January 2020, 13:00:00
Modified:	Thursday, 4 November 2021, 20:44:14
Accessed:	Today, 1 November 2022, 1 minute ago
Attributes:	□ Read-only □ Hidden

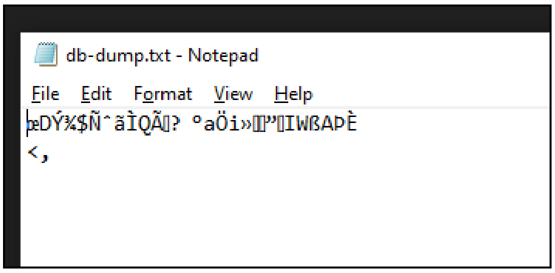
Fig 4 - The property of the file has been has been changed, along with its file name and file extension



Word experienced an error trying to open the file.
Try these suggestions.
* Check the file permissions for the document or drive.
* Make sure there is sufficient free memory and disk space.
* Open the file with the Text Recovery converter.
(D:\Documents\ICT2202\...\db-dump.docx)

Fig 5 - The file is unable to open as it was never the right extension

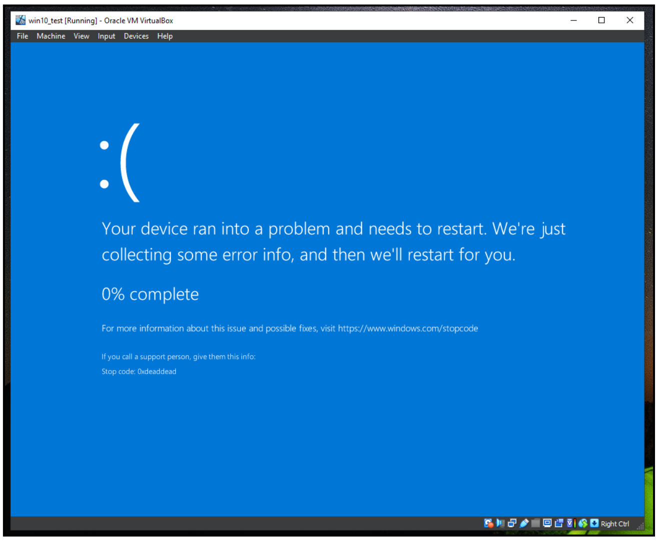
As shown in figure 4, the Metadata of those files has been changed into a random date. Figure 5 shows one of the files, which was unable to open due to an error, this was due to it being in an incorrect file type. Other files might be able to open if the file type is compatible with the file extension it has saved as.



```
þDÝ¾$Ñˆ ãÌQÃ▯? °aÖi»▯▯Þ'▯IWßAÞÈ
<,
```

Fig 6 - The content of the file has been encrypted and unable to open as it was never the right extension

Figure 6 shows that the same file from Figure 3 has been changed to the correct file extension, the content of it would be unreadable as it has been encrypted.



:(

Your device ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

0% complete

For more information about this issue and possible fixes, visit https://www.windows.com/stopcode

If you call a support person, give them this info:
Stop code: 0xdeaddead

Fig 7 - BSOD on the PC after a few second

As shown in figure 7, the machine would just crash without a valid status code. This would occur shortly after the previous procedure has been executed.

X. Solution Comparison

A. USBkill

Blue's clue differs from the existing tool USBkill such that USBkill prompts the user to create a whitelist for the desired devices to be allowed to be connected to the computer. Whereas, Blue's clue does not prompt the user upon installation, however it runs in the background while the computer starts up where the user has to make sure that the desired devices have been plugged in before starting the computer to be whitelisted.

The key difference lies in the main feature of USBkill, which involves functionality such as computer shutdown, erasing data from the RAM and swapping files by default while Blue's clue will plant encrypted false evidence randomly in the computer attempting to lure the digital forensics investigator during an investigation. Ultimately the main intent of Blue's clue is to frustrate the digital forensics investigator by creating files masquerading as evidence [5].

B. Silk-guardian

Silk-guardian took inspiration from USBkill, but was implemented as a Linux kernel driver. The difference compared to our solution would be its requirement to be built manually if any changes to the source code prior to being built [23]. It is also able to whitelist devices that are not usb drives such as plug-and-play devices such as a keyboard.

Additionally, Silk-guardian can be made to run by unplugging a recognised USB device from the device that has not yet been whitelisted. However, for this to work, the recognised USB device will have to be plugged into the device before loading the Silk-guardian kernel module. This basically employs the USB-on-a-lanyard technique. The disadvantage of this technique is that it is relatively more obvious to law enforcers who might be aware of such anti-forensics tactics. Hence, by designing our tool to detect foreign drives instead will ensure our anti-forensics intentions are less conspicuous.

C. xxUSBSentinel

xxUSBSentinel comes with more features compared to our solution. It is able to monitor and log the USB devices disconnected and connected events, as well as export those logs. It detects USB devices with their Vendor ID and product ID, while our solution uses the serial number of the drive instead. It uses a GUI for its control (Graphical User Interface) while our solution requires the user to view through the Python code and modify it.

The key function of xxUSBSentinel was the fast shutdown switch. xxUSBSentinel can set a designated flash drive to be the fast shutdown switch and shutdown will happen when the designated flash drive is removed. This is in contrast to our solution which works by detecting unrecognised USB flash drives. As mentioned previously, this will ensure our anti-forensics intentions are relatively less obvious to law enforcement officers.

Additionally, unlike our tool, the primary anti-forensics method the xxUSBSentinel employs is to shutdown the computer when a recognised USB device is disconnected [24]. Whereas our tools consist of deception and disruption techniques, making law enforcement personnel waste unnecessary and significant time and effort to conduct forensics.

D. Usbdeath

Usbdeath is a script that was inspired by Usbkill. It is written in bash, which is only applicable in linux systems, while our tool is for Windows only. It's not run as a background process, rather a file manipulation script, opposite to how we implement our solution. It uses more identification value compared to our tool [8].

XI. Conclusion

There are various solutions on the internet that perform similar functions as our solution. But solutions such as USBkill and Silk-guardian are aggressive as their purpose is to get rid of as much trail as possible, while xxUSBsentinel and Usbdeathis more of safeguarding the computer.

Our solution took a slightly more covert solution with its ability to create random files that serve as a roadblock to an ongoing forensic and the BSOD crash further disrupts any intrusion to the system by forcing the system to shut down, thus effectively locking the computer.

Our tool is also designed for Windows machines, which will make it applicable to many more users who only use the Windows OS.

Our solution is not perfect by any means as there are ways to get around it. An example would be to plug in a foreign usb drive before the computer startup, which will enable it to gain access. However, it is unlikely for the user of this tool to leave his computer turned off, seeing how he or she is already aware of such vulnerability. In conclusion, We have shown that it is possible for any person with basic programming language skills to create an anti-forensic tool. This does not mean that such a tool should be used for anti-forensic purposes, but also as an alternative for safeguarding.

XII. Reference

[1] Vultaggio, N. (2012) *History of the USB*, *Gorilla Marketing Promo*. Gorilla Marketing Promo. Available at: https://gorillamarketing.net/blogs/gorilla-marketing-blog/gorilla-blog-history-of-usb (Accessed: November 5, 2022).

[2] Keun-Gi Lee; Hye-Won Lee; Chang-Wook Park; Je-Wan Bang; Kwon-youp Kim; Sangjin Lee (2008) “USB PassOn: Secure USB Thumb Drive Forensic Toolkit”, Online: https://ieeexplore.ieee.org/document/4734222 (Accessed: September 15 2022)

[3] P. Thomas and A. Morris, "An Investigation into the Development of an Anti-forensic Tool to Obscure USB Flash Drive Device Information on a Windows XP Platform," 2008 Third International Annual Workshop on Digital Forensics and Incident Analysis, 2008.

[4] O.Angelopoulou, S.Pourmoafi, A.Jones, G.Sharma (2019) “Killing Your Device via Your USB Port”, Online: http://wrap.warwick.ac.uk/137908/1/WRAP-killing-your-device-via-USB-port-Angelopoulou-2020.pdf (Accessed on: 15/09/2022)

[5] hephaest0s (2015) *USBKILL*, *GitHub*. Available at: https://github.com/hephaest0s/usbkill (Accessed: November 5, 2022).

[6] NateBrune (2015) *silk-guardian*, *GitHub*. Available at: https://github.com/NateBrune/silk-guardian (Accessed: November 5, 2022).

[7] Thereisnotime (2018) *xxusbsentinel*, *GitHub*. Available at: https://github.com/thereisnotime/xxUSBSentinel (Accessed: November 5, 2022).

[8] Trpt (2016) *USBDEATH*, *GitHub*. Available at: https://github.com/trpt/usbdeath (Accessed: November 5, 2022).

[9] Nissim, N., Yahalom, R., & Elovici, Y. (2017). USB-based attacks. Computers & Security, 70, 675–688. https://doi.org/10.1016/j.cose.2017.08.002

[10] Cannols, B. and Ghafarian, A. (n.d.). Hacking Experiment by Using USB Rubber Ducky Scripting. [online] Available at: https://www.iiisci.org/journal/PDV/sci/pdfs/ZA340MX17.pdf (Accessed: November 5, 2022).

[11] g33kyrash (2015) *evilduino*, *GitHub*. Available at: https://github.com/g33kyrash/EvilDuino (Accessed: November 5, 2022).

[12] *Python time and datetime tutorial with examples* (2022) *Software Testing Help*. Available at: https://www.softwaretestinghelp.com/python-datetime/#The_DateTime_module (Accessed: November 5, 2022).

[13] Golden, T. (no date) *WMI - Windows Management instrumentation¶*, *WMI - Windows Management Instrumentation - WMI v1.4.9 documentation*. Available at: http://timgolden.me.uk/python/wmi/index.html (Accessed: November 6, 2022).

[14] *Pycryptodomex* (no date) *PyPI*. Available at: https://pypi.org/project/pycryptodomex/ (Accessed: November 5, 2022).

[15] kubinka0505 (2022) *filedate*, *GitHub*. Available at: https://github.com/kubinka0505/filedate (Accessed: November 5, 2022).

[16] joke2k (2012) *faker*, *GitHub*. Available at: https://github.com/joke2k/faker (Accessed: November 5, 2022).

[17] Bertrand, N. (2015) *The FBI staged a lovers' fight to catch the kingpin of the web's biggest illegal drug marketplace*, *Business Insider*. Business Insider. Available at: https://www.businessinsider.com/ross-ulbricht-will-be-sentenced-soon--heres-how-he-was-arrested-2015-5 (Accessed: October 26, 2022).

[18] *What is stuxnet?* (no date) *Malwarebytes*. Available at: https://www.malwarebytes.com/stuxnet (Accessed: November 5, 2022).

[19] Itzsten (2022) *Hallucinate*, *GitHub*. Available at: https://github.com/Itzsten/Hallucinate (Accessed: November 5, 2022).

[20] *Operating system market share worldwide* (no date) *StatCounter Global Stats*. Available at: https://gs.statcounter.com/os-market-share (Accessed: October 31, 2022).

[21] Whittaker, Z. (2019) *Another US Court says police cannot force suspects to turn over their passwords*, *TechCrunch*. Available at: https://techcrunch.com/2019/11/21/court-police-suspects-passwords/ (Accessed: November 5, 2022).

[22] *Power to access computer* (2018) *Singapore Statutes Online*. Available at: https://sso.agc.gov.sg/Act/CPC2010?ProvIds=pr39-#pr39- (Accessed: November 5, 2022).

[23] Gentoo.org. (2019). Silk Guardian - Gentoo Wiki. [online] Available at: https://wiki.gentoo.org/wiki/Silk_Guardian (Accessed: 4 Nov. 2022).

[24] Suleman, M. (2019). Automatically Shutdown PC When Specific USB Device Disconnects. [online] I Love Free Software. Available at: https://www.ilovefreesoftware.com/03/windows-10/automatically-shutdown-pc-when-specific-usb-device-disconnects.html (Accessed: 4 Nov. 2022).