OneStopLogs

Tay Qi Wen Jordan   
1902114 *1902114@sit.singaporetech.edu.sg*

Lee Khai Liang Eugene   
1902155  
1902155@sit.singaporetech.edu.sg  
Poh Hong Yi  
1902148  
*1902148@sit.singaporetech.edu.sg*

Poh Qi Xian Bryan  
1902197  
*1902197@sit.singaporetech.edu.sg*

# **introduction**

OneStopLogs is tool that performs both Log Extraction and Log Visualisation. OneStopLogs consists of a plug and play thumb drive script that helps Incident Responders extract out the logs and a web application to help analyse and visualize the logs while flagging out important events that happen within the logs.. The aim of OneStopLogs is to help an Incident Responder be able to easily filter various kinds of information from different logs. This is done to help reduce the time that incident responders need to spend on analysing all these logs compared to having to manually look through the logs. Manually looking through the logs may lead to incident responders missing out vital and critical information amongst the many log entries. Our goal for OneStopLogs is to be able to quickly script out the more important logs and to visualise them more efficiently.

# **background research**

Event logs are files that holds details on usage and operations of the devices, operating systems or applications. Several recent studies[1,5,9,17] have suggested that event logs can provide useful insights to the system's history which can assist system administrator in understanding how the attacker might enter and compromise the system by showing sequence of action which led to an attack on the system. For example, if an attacker was brute forcing the system and succeeded in gaining access, the event logs would show multiple failed login entries followed by a successful login entry. In addition, it would also show which user account was compromised allowing the system administrator to take the appropriate actions on the correct user account.

Event logs can also show what an attacker might be doing after compromising the system, which could potentially reveal connections to other systems, thus identifying the attacker or locating other breaches in the system. For instance, after gaining access to the system an attacker might want to retrieve some important files. If he accessed the system remotely, he needs to connect back to his system and the logs will capture his IP address. While if he accessed the system physically, he needs to insert a USB and the logs will have a timestamp of when he is in the compound.

The advancement of technology and the increasing number of end-point devices meant that attacks involving normal users are also increasing. As event logs are like an audit trail as it records user events, it aids the forensic investigator during a digital forensic investigation. It can provide timestamp of events which helps investigator in establishing a timeline and narrow down potential suspects. However, as event logs records systems operations in addition to user events, this results in a large amount of information being recorded, thus its challenging to find relevant information from event logs quickly. According to[1], it is suggested that good visualization aids in giving meaning to the logs extracted as well as gaining better understanding of system operations.

Researches[6,22] shows that although there is an increase in demand for IT industries, there is a lack of qualified personal to satisfy the hiring needs. Therefore, there is a need for tools to be efficient and effective while remaining easy and intuitive to use so as to bridge the existing technology skill gap while not sacrificing results.

# **Analysis of the issue and challenges faced**

Given the importance of logs, there is a lack of free open source software that does both gathering of logs and proper visualization of the events in them that use on both Windows and Linux. For Windows there is Event Viewer application to display all the logs but lack of proper graphical visualizations which make it difficult for user to understand it. For most user, it just a huge amount of meaningless data without any form of analysing with a lack of user-friendly interface. Many of the tools in the market is either only for Windows or for Linux and only focusing on collection or visualization of logs. For example, one Windows side there is Kiwi Syslog Server while on Linux point there is Graylog. From the common nature of logs, there is a large volume of them collected overtime leading to a massive speed issue when processing the proper information of them.

# **solutions & tools availiable**

There are a variety of solutions and tools available for log extractions and log visualisations. However, as mentioned in our analysis, there are no tools that is free that performs both log extractions and log visualisations. Below are a list of solutions and tools that are mostly free and performs either log extraction or log visualisation.

## FTK imager (Open Source)

FTK imager is a imaging tool that helps Incident Responders assess Electronic evidence to view. It can generate copies of the evidence, open all types of evidence regardless of what Operating System the evidence come from, view and extract all the files and folders in the evidence, extract logs from the folders, see and recover deleted files. It can do all these while ensuring that the integrity of the original source stays the same. It is also a very fast imaging tool.

## Sans Sift Workstation (Open Source)

Sans Sift Workstation is a group of open-source incident response and forensic tools designed to perform digital forensic examinations. It can match any current incident response and forensic toolkit. It can take a log file and make it into a timeline csv by using a tool called Log2time.

## FireEye RedLine(Open Source)

FireEye RedLine is an open-source security tool. It provides users with investigative capabilities to find signs of malicious activity through memory and file analysis and the development of a threat assessment profile. This tool is able to collect from memory file-system meta data, registry data, event logs, network information, services, tasks and web history. It is also able to view the contents of all these data. It is able streamline memory analysis for analysing malware based on priority. Can perform Indicators of Compromise analysis.

## GoAccess (Open Source)

GoAccess is a fast and terminal based log analyser. It is able to quickly analyse and view web statistics in real-time without needing to use your browser. It is just for analysis and visualization of Web logs.

## Log2Timeline (Open Source)

Log2Timeline is a tool for generating forensic timelines from digital evidence such as disk images and event logs. It uses the logs from other tools to generate a document for timeline analysis, can have a lot of formats.

## SpectX (Open Source)

SpectX is a parser and query engine for analysing logs across multiple sources. It will create a structed virtual view from the logs files and enables analysts to get quick statistics insights. It can also parse, enrich and join results from multiple sources.

## Log Parser 2.2 (Open Source)

Log parser is a tool that provide universal query access to text-based data such as log files, XML files, CSV files, as well as key data sources on Windows Operating System such as the Registry, the file system, and active directory. Your query can be custom-formatted in text base output or changed to more speciality targets such as SQL and SYSLOG.

## Autopsy (Open Source)

Autopsy is a digital forensic platform and graphical interface to The Sleuth Kit and other digital forensic tools. It has modules that performs timeline analysis, keyword search, finding web artifacts, data carving, extracting EXIF from pictures and find indicators of compromise.

## ELK Stack (Paid)

ELK Stack is 3 tools together, Elasticsearch, Logstash and Kibana. Elasticsearch is for searching for actions of a specific IP address, analysing a spike in transaction request. This tool enables us to store, search and analyse with ease. Logstash is a processing pipeline that ingest data from a multitude of sources. Transforms it and send it to a stash. Kibana then uses the data to generate visualization such as waffle charts and heat maps.

## Splunk (Paid)

Splunk makes it simple to collect, analyse and act upon big data, giving u the insights to drive operational performance and business results. It can collect, index logs and machine data from source. It then combines the machine data with data in your relational databases, data warehouses and Hadoop and NoSQL data stores.

# **Onestoplogs’s technical solution**

OneStopLogs is a free and open source that is able to perform Log Extraction, Log Analysis and Log Visualization.

## Log Extraction

**Windows Environment**

We decided to select the three main default logs that store information, namely Security Logs, Application Logs and System Logs. We also decided to add four more components to our extraction of the log files. Local Session Manager logs, WLAN-auto config logs, Prefetch and Recent

1. Security Event Logs

Information that relate to login attempts, elevated privileges, and other audited events. It also contains security-related events specified by the system’s audit policy. It is one of the primary tools Investigators view to detect and investigate attempted and successful unauthorized activity.

1. Application Event Logs

Information logged by applications and Services hosted on the local machine. Any application that start-ups, shutdown or have run-time error will all be logged in by Application Logs. The same holds for events written by Windows Services as well.

1. System Event Logs

Messages that are generated by Windows Operating System. Usually they are records of events written by device drivers. This Log will contain Service Control Manager events, Windows Update Service events and run-time error events. It will also show the severity of the event being logged.

1. Local Session manager

Log files of system components that creates, destroy and manages sessions. It can view information, control and configure session running on the terminal server. It can also view the enumeration information related to sessions running on a terminal server.

1. WLAN-auto config

Log files that contain messages from the WLAN service that handles the core functions of wireless network in Windows. This Log will contain successful login to a WLAN and wireless network, failure to login to a WLAN or wireless network, the network around you and any fail attempts to connect to a wireless due to incorrect credentials.

1. Prefetch

A Windows feature implemented to speed up the loading of programs. Information by default is stored as an individual file in Prefetch with a pf extension. We can extract this list to identify details about file execution such as file name, timestamp, resource consumed as file executes. We choose this to extract due to Prefetch being able to view for past 7 days what application the user has run/executed or write into.

1. Recent

A Windows file Directory that contains all of your most recently accessed things. It can be documents, folder, device manager, VMs and the list goes on. We decide to extract this because we felt that this give us more insight on what was the user recently accessing before the incident happen, to be able to pinpoint the source of the problem.

**Linux Environment**

For Linux, we decided to take the 4 main logs, Auth.logs, kern.logs, message.logs and sys.logs. We have also decided to look at lastlog, last, lastb and history files from Linux.

1. Auth.logs

Authentication related events in Debian and Ubuntu server logged in it. We can use it to view anything involving user authorization in it. Basically, using it to investigating failed login attempts, brute-force attacks and other vulnerabilities related to user authorization mechanism.

1. Kern.logs

Information logged by the kernel, which is the kernel information and events on your system. It also contains warning data. This log is to troubleshoot kernel related errors and warning, helpful in troubleshooting custom-built kernel, and useful in debugging hardware and connectivity issues.

1. Message.logs

Contains generic system activity logs. It is mainly used to store informational and non-critical system messages. We use this log to track non-kernel boot errors, application-related service errors and the messages that are logged during system start-up. This is the first log file we check if something goes wrong.

1. Sys.logs

Stores all global system activity data, including startup messages. We use this log to also track for errors and status messages of all applications. It also shows system error messages, system startups and shutdowns, and change in network configuration.

1. Lastlog

Reports the most recent login of remote users. It will format and print the contents of the last login log in the lastLog file. It will print the login-name, port and the last login time.

1. Last

Searches through the /var/log/wtmp and displays a list of all users logged in (and out) since that file is created. Basically, we are able to view local and remote connection.

1. Lastb

Same as Last, except that it shows a log of all bad login attempts. We use this file to view all bad login attempts into the system.

1. History

A File that will show us the last five hundred commands the user have entered into the system. We use this file to view if there is a bad command that the users typed into the machine that cause the problem/breech to happen.

## Log Analysis

**Windows Environment**

1. Security Event Logs

The logs highlighted to the user are for: service installed, login events, user account management, computer account manager, security group management, object access events, cryptographic action attempt, network share object, IPsec, code integrity, replay attack and unprotection of protected data attempted.

As to why these logs was decided to be collected, user account management, security account management and computer account management are important for the user as a common occurrence for a threat to the system is to gain access to the system is by resetting account password, creating and enable account. If the threat goal of distributing the system, accounts will be locked, disabled, or even deleted.

Login logs are monitored and brought up to the user for cases of any brute force attempt to the account, out of place logins and superuser logins when not in use. Service installed event is taken note of in case of any authorized service being installed.

Security Group Management is also monitored to track of any security group changes in the environment that can lead to any permission being compromised. Object access event is retrieved if the user wants to enable logging of the object for monitoring.

Network shared object in the event of any medication and deleting of the object, which the user can see into it for any changes or missing network object files not done by the user for suspicious activity.

Attempted cryptographic context and function is investigated to highlight to the user there is a cryptographic event could lead to a file crypted to be hidden away from the user.

IPsec settings modification is being visualization to the user as IPsec provides confidentiality, integrity, and authentication for network layer, which can lead to any compromise packets due to unauthorized changes to the settings for that layer.

Code integrity log is important as it alert when hash of file is not valid, leading to authentication issues. Replay attack is a log user must take note as a man-in-the-middle attack may have occurred.

Lastly, unprotection of protected data attempted could means an encryption by a different user is being decrypt by a different user, possibly means that an unauthorized decryption is materialize.

1. Application Event logs

As application logs does not have industry standards for managing its log events, events are inconsistent in terms of event IDs used and description of events. However, there is still a need to analyse application logs as it is critical to identify unavailable services, data corruption and application faults. Thus, we decided to filter the logs by its type - Information, Warning, Error and Critical. By doing so, allows the user to quickly focus the more important events, i.e., Error - Application/services stopped.

1. System Event logs

The logs highlighted are possible CVEs, services installed. We also have a Pie chart that shows the severity of the logs grouped by warning, error and critical.

For the Pie Chart, we grouped all the logs into the respective severity level and displayed it to them. Then we will show a table for the warning logs first. If they click any part of the pie chart, we will change the table to the respective corresponding severity level so that they can view what they need.

Services installed by the user is a key event as they can affect your entire system. If there are malware or data breeches that just recently happen, we will want to look at what types of services was just installed in the computer so that we can see if the user may have accidentally installed some malware. This is also due to the fact that humans are always the weakest link in the security chain, thus may unknowingly execute any forms of virus.

Possible CVEs are logged whenever an exploit attempt is made against a vulnerability that has been patched. This is key in identifying what are the possible attacks that have happened during the timeframe of the incident. This can lead investigators to see if attackers may have tried some old exploits before moving to more advanced attack.

1. Local Session Manager logs

We have displayed a table to show local session log in and outs. Investigator can look at this table to see if at the time of the incident, what user was logged in. This can narrow down the search scope a lot of just view what the user has been doing up to the point of the incident and after to see if maybe he has been running malicious applications.

1. WLAN-auto config logs

We have displayed a table to view which WIFI the user is connected to, what time the user connected to it, and what time it stopped connecting to it, succeed or not, and modes. Investigators can look at this file to find out what WIFI the company is using, what mode of authentication they are using. If it is not WPA2-Personal Enterprise, it is very easily hackable and could be the source of the data breech.

1. Prefetch

We choose to look at prefetch as it contains details of recently executed files. Investigators will want to look at this file as when an incident happen, usually what the user has run for within the most recent 7 days is highly suspicious and it can be that the user may have accidently run some malware that caused the incident.

1. Recent

We look at this file as it can show us what the user has most recently accessed. Investigators will want to look at this file as when an incident occurs, we can see what has the users has accessed, run or execute for the past 7 days leading up until the time of the incident. This is very key in finding out what exactly the user has been doing and running that may have caused the incident.

**Linux Environment**

1. Auth.logs

We focused on who used the SU command and who the user SU to. This is because if the user doesn’t have sufficient permission, it is highly unlikely that an employee of your company will use SU to privilege escalate himself to access files that he/she is not supposed to. So, if there is a malware or data loss that just happen, this is one key factor to look at.

1. Kern.logs,

We focused on newly installed USB as it is highly suspicious if a malware or data loss just happen to the company within 7 days of installation of a new USB device. We will extract the date, the manufacturer, product and serial number of the USB.

1. Sys.logs

We focused on the warning, errors and critical errors. These are all indicators of any suspicious activities that could cause the problem/data loss that just happen to the company.

1. Lastlog command

We took everything that is inside as it displays all users that have logged in remotely, what port they came in, from where they login from and the time of their last login.

1. Last command

We took everything that is inside as it displays all users that have logged in, locally or remotely, and we want to see what time they accessed, how long was their session, hostname, how the user was connected, and where the user is connected.

1. Lastb command

We mainly took the username of the host, how the user was connected, where the user was connected and the time it happened. This to see all the bad logins that was attempted.

1. History command

We just take everything in the file as the pass 500 commands that the user type may or may not have triggered and event that cause the incident to happen, thus we need to look carefully at what the user typed in case the problem came from user’s own commands.

# **Comparison with other tools**

There is no shortage of event logs analysis tools out there, but not all are suited for users with little to no knowledge. OneStopLogs is created and designed to accommodate these users. OneStopLogs comes with plug and play log collection script for the collection of Windows or Linux Event Logs - tested on both Windows and Linux operating system, user can run the script from a USB and the script will handle the rest. Logs collected by the OneStopLogs log collection script can be visualized and display through OneStopLogs very own web-based user interface which runs on python3(Flask). OneStopLogs will auto display top few events - events that we felt users should pay more attention to - in graphs and table format. In the event where the user wishes to have more information, they can also choose to view the raw logs. OneStopLogs provides a search bar where users can use to filter their raw logs through the use of keywords.

# **technologies used by onestoplogs**

## Batch Scipt

Batch script is a list of commands that are processed by the command prompt and performed in sequence. A batch file is normally used to run multiple processes, perform repetitive tasks or help load programs without any user input or intervention. Batch scripts are often used to save time for the user.

## Powershell

PowerShell is a task automation and configuration management framework from Microsoft. PowerShell is built on the .NET Core and comes with its own scripting language. PowerShell also has specialized commands called cmdlets that performs specific functions.

## Python

Python is an object-oriented, high-level programming language. Python is simple and has easy to learn syntax which makes picking up the language fast. The edit-test-debug test is very fast as well, which makes it attractive for rapid application development. Python also comes with extensive libraries and open source codes from other developers.

## Flask Python Framework

Flask python is a framework for web applications. Flask provides tools and libraries that allows a user to set a web application easily. Flask is considered a lite framework, which means that it has little dependency on external libraries.

## HTML

HTML is the standard markup language for creating web pages, it uses element tags to display the structure and content of the web page.

## Javascript

JavaScript is a programming language that helps to implement complex features on a webpage. JavaScript can also be used to display various charts and visualization graphics.

## Bootstrap

Bootstrap isa CSS framework which helps to style the webpage and make the webpage responsive.

## Jinja

Jinja is a templating language used to communicate between Python and Flask. Through Jinja, we are able to pass in variables from Python into the Html webpage.

## Werkzeug

Werkzeug is a WSGI web application library. Werkzeug is used to provide error handling functions in the Flask framework

# **Architecture & engineering of onestoplogs**

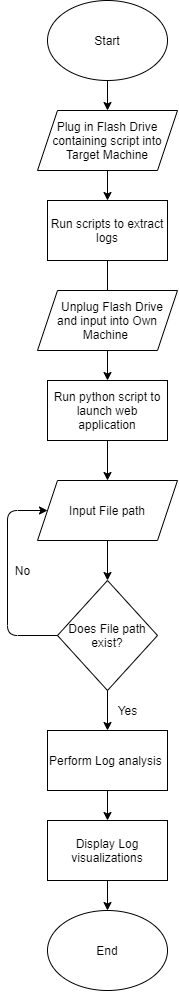
OneStopLogs is separated into three different phrases, Log extraction, log analysis and log visualization.

With the help of Batch Scripts and PowerShell scripts, OneStopLogs is able to detect and perform log extraction from different types of Operating Systems. Due to time constraints, OneStopLogs will specifically target only Windows and Linux (Ubuntu and Kali). OneStopLogs will also only extract logs that we feel are the most important that Incident Responders look out for first.

With the help of Python, OneStopLogs will also provide Log analysis where we look at the logs, filter and categorize the logs. OneStopLogs will only utilize entries that we deem important.

With the help of Flask and Jinja, OneStopLogs performs log visualization and transforms the logs into various charts and showcase it on a html web application for the Incident Responders. However, with the limited timeframe given, we are unable to provide very in-depth analysis of all the logs extracted, therefore, the visualizations done are minimal.

Figure 1 below depicts the process flow of our application



# **test setup and results**

## Log Extraction

#### Windows Environment

As shown in Figure 2, the USB device starts with only the log extraction scripts and web application inside. Double click the OneStopLogs.bat script to execute the script. As a temp/logs/windows folder cannot be found, the script will create a directory as shown in Figure 3. the logs will then be extracted into <Drive:>/temp/logs/windows/<Date> as shown in Figure 4. Note that the script extracts the logs into the directory that the script resides in. This effectively demonstrates the portability and ease of use of OneStopLogs.

**\***Windows Operating System automatically blocks scripts downloaded from the internet to help protect the computer. However, we have found a workaround and are able to unblock the script from bash.

**\*\***By default, Windows Operating System will set a user’s Execution-Policy to be restricted. This causes a problem for us as we are unable to run scripts on the machine. However, there is no need to manually change the Execution-Policy of the current user as we have found a way to bypass and elevate the Execution-Policy from restricted to remote access. After the execution of our scripts, we will demote the Execution-Policy of the user back to restricted.

## Linux Environment

OneStopLogs works in a Linux Operating System as well. As shown in Figure 5, the USB starts with only the log extraction scripts and web application inside. We are also able to see the windows logs that we have just extracted.   
Likewise, fire up a terminal, give the scripts appropriate permission for execution, and execute the script. Similarly, to Windows, extracted logs will be saved into <Media Directory>/temp/logs/linux/<Date> as shown in Figure 6.

## Log Analysis

## Start up the python web application by running the webapp.py file. The index page will allow you to input a file path. Insert in one of the file path for the previously extracted logs <Drive:>/temp/logs/<OS>/<Date> as shown in Figure 7.

## Refer to V. ONESTOPLOGS’S TECHNICAL SOLUTION Part B. for reasons why we chose specific event entries to analyse.

## Log Visualisations

After the analysis and filtering of the logs, we display the filtered logs in the form of graphs and tables.

**Windows**

Overview Page: Figure 8

Security Page: Figure 9-12

Application Page: Figure 13

System Page: Figure 14

Timeline Analysis Page: Figure 15-16

Raw logs Page: Figure 17

**Linux**

Overview Page: Figure 18

User activity: Figure 19

# **References**

[1] Aharon M., Barash G., Cohen I., Mordechai E. (2009) One Graph Is Worth a Thousand Logs: Uncovering Hidden Structures in Massive System Event Logs. In: Buntine W., Grobelnik M., Mladenić D., Shawe-Taylor J. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2009. Lecture Notes in Computer Science, vol 5781. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-04180-8_32>

[2] B. Dysert, “What is the Purpose of the Application Event Log?,” *tips.net*, 22-May-2017. [Online]. Available: https://windows.tips.net/T012886\_What\_is\_the\_Purpose\_of\_the\_Application\_Event\_Log.html. [Accessed: 30-Oct-2020].

[3] “Batch Script Tutorial,” *Tutorialspoint*, 2020. [Online]. Available: https://www.tutorialspoint.com/batch\_script/index.htm. [Accessed: 30-Oct-2020].

[4] “Community: Downloads,” *SIFT Workstation Download*, 2008. [Online]. Available: https://digital-forensics.sans.org/community/downloads. [Accessed: 30-Oct-2020].

[5] E. Kenneally, *Digital logs - Proof matters*, Jun. 2004.

“FTK® Imager,” *AccessData*. [Online]. Available: https://accessdata.com/products-services/forensic-toolkit-ftk/ftkimager. [Accessed: 30-Oct-2020].

[6] J. Voogt, O. Erstad, C. Dede, and P. Mishra, “Challenges to learning and schooling in the digital networked world of the 21st century,” *Journal of Computer Assisted Learning*, Jun. 2013.

[7] JBrown, “Extracting Windows Prefetch Files,” *Medium*, 12AD. [Online]. Available: https://medium.com/@stdout\_/extracting-windows-prefetch-files-41a6aac63291. [Accessed: 30-Oct-2020].

[7] “Jinja¶,” *Jinja*, 2007. [Online]. Available: https://jinja.palletsprojects.com/en/2.11.x/. [Accessed: 30-Oct-2020].

[9] K. Dashora, D. S. Tomar, and J. L. Rana, *A Practical Approach for Evidence Gathering in Windows Environment*, Aug. 2010.

[10] “lastlog - Unix, Linux Command,” *Tutorialspoint*. [Online]. Available: https://www.tutorialspoint.com/unix\_commands/lastlog.htm. [Accessed: 30-Oct-2020].

[11] “Log Parser 2.2,” *Log Parser 2.2 : The Official Microsoft IIS Site*, 08-Apr-2010. [Online]. Available: https://www.iis.net/downloads/community/2010/04/log-parser-22. [Accessed: 30-Oct-2020].

[12] “Log2Timeline Tutorial,” *Medium*, 14-May-2020. [Online]. Available: https://medium.com/@cloudyforensics/log2timeline-tutorial-d769994c3570. [Accessed: 30-Oct-2020].

[13] M. Adrian, R. Armin, D. Lord, and U. Markus, “Werkzeug,” *Pallets*, 2015. [Online]. Available: https://palletsprojects.com/p/werkzeug/. [Accessed: 30-Oct-2020].

[14] M. Blodgett, A. M. Gorzelany, C. Detels, D. Mabee, N. Schonning, K. Ainapure, G. Lindsay, L. Poggemeyer, and D. Simpson, “Advanced Troubleshooting Wireless Network Connectivity - Windows Client Management,” *Advanced Troubleshooting Wireless Network Connectivity - Windows Client Management | Microsoft Docs*, 20-Aug-2020. [Online]. Available: https://docs.microsoft.com/en-us/windows/client-management/advanced-troubleshooting-wireless-network-connectivity. [Accessed: 30-Oct-2020].

[15] M. Maruthamuthu, “How To Check User Login History In Linux?,” *2daygeek.com*, 03-Apr-2019. [Online]. Available: https://www.2daygeek.com/how-to-check-user-login-history-in-linux-using-last-lastb-lastlog-command/. [Accessed: 30-Oct-2020].

[16] Marcel, “12 Critical Linux Log Files You Must be Monitoring,” *12 Critical Linux Log Files You Must be Monitoring Comments*, 19-Apr-2018. [Online]. Available: https://www.eurovps.com/blog/important-linux-log-files-you-must-be-monitoring/. [Accessed: 30-Oct-2020].

[17] P. K. Sahoo, *Research Issues on Windows Event Log*, Mar. 2012.

[18] “Python Introduction,” *Introduction to Python*. [Online]. Available: https://www.w3schools.com/python/python\_intro.asp. [Accessed: 30-Oct-2020].

[19] “Redline: Free Security Software,” *FireEye*, 2018. [Online]. Available: https://www.fireeye.com/services/freeware/redline.html. [Accessed: 30-Oct-2020].

[20] “Run Queries on Raw Log Files. Instantly.,” *SpectX*, 2015. [Online]. Available: https://www.spectx.com/. [Accessed: 30-Oct-2020].

[21] “Splunk Enterprise,” *Splunk*. [Online]. Available: https://www.splunk.com/en\_us/download/splunk-enterprise.html?utm\_source=capterra. [Accessed: 30-Oct-2020].

[22] T. Berger and C. B. Frey, “TECHNOLOGY, GLOBALISATION AND THE FUTURE OF WORK IN EUROPE ESSAYS ON EMPLOYMENT IN A DIGITISED ECONOMY,” *BRIDGING THE SKILLS GAP*, Mar. 2015.

[23] “Using the history command,” *Using the history command | Media Temple Community*. [Online]. Available: https://mediatemple.net/community/products/dv/204404624/using-the-history-command. [Accessed: 30-Oct-2020].

[24] “Visual Web Log Analyzer,” *GoAccess*, 2010. [Online]. Available: https://goaccess.io/. [Accessed: 30-Oct-2020].

[25] W. H. Nicholas, “An introduction to the Flask Python web app framework,” *Opensource.com*, 02-Apr-2018. [Online]. Available: https://opensource.com/article/18/4/flask. [Accessed: 30-Oct-2020].

[26] W. Sean and K. Shawn, “What is PowerShell? - PowerShell,” *PowerShell | Microsoft Docs*, 22-May-2020. [Online]. Available: https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7. [Accessed: 30-Oct-2020].

[27] “What is HTML?,” *What is HTML*. [Online]. Available: https://www.w3schools.com/whatis/whatis\_html.asp. [Accessed: 30-Oct-2020].

[28] “What is JavaScript?,” *MDN Web Docs*, 2005. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\_steps/What\_is\_JavaScript. [Accessed: 30-Oct-2020].

[29] “What is the ELK Stack?,” *Elastic*, 2020. [Online]. Available: https://www.elastic.co/what-is/elk-stack. [Accessed: 30-Oct-2020].

[30] “Windows Logging Basics - The Ultimate Guide To Logging,” *Log Analysis | Log Monitoring by Loggly*, 05-Aug-2019. [Online]. Available: https://www.loggly.com/ultimate-guide/windows-logging-basics/. [Accessed: 30-Oct-2020].

[31] “Windows Security Log,” *Wikipedia*, 26-Aug-2017. [Online]. Available: https://en.wikipedia.org/wiki/Windows\_Security\_Log. [Accessed: 30-Oct-2020].

[32] “[MS-TSTS]: Overview,” *[MS-TSTS]: Overview | Microsoft Docs*, 15-Feb-2019. [Online]. Available: https://docs.microsoft.com/en-us/openspecs/windows\_protocols/ms-tsts/0fb8dffb-3d82-43ca-8c09-6f62e36cfc9a. [Accessed: 30-Oct-2020].