

How to detect lateral movement in the Windows operating system?

MSc Research Project
MSc in Cybersecurity

Nikhil Somashekarappa
Student ID: 20179162

School of Computing
National College of Ireland

Supervisor: Imran Khan

National College of Ireland
MSc Project Submission Sheet



School of Computing

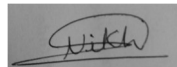
Student Name: Nikhil Somashekarappa
Student ID: 20179162
Programme: MSc in Cybersecurity **Year:** 2021-2022
Module: Research project
Supervisor: Imran Khan
Submission Due Date: 16-Dec-2021
Project Title: How to detect lateral movement in the Windows operating system?

Word Count: 4177 **Page Count:** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:



Date: 15-Dec-2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

How to detect lateral movement in the windows operating system?

Nikhil Somashekarappa
20179162

Abstract

In advance persistent threat (APT) attacks on businesses, lateral movement (LM) is one of the most significant phases. The attackers employ lateral movement to get network access while remaining undetected. If not recognized early enough, lateral movement attacks can be highly dangerous. The purpose of this research is to look at lateral movement detection techniques in the Windows operating system. To identify the lateral movement, the windows event log monitoring and analysis methods are utilized. The following research project examines the two most common lateral movement attack techniques: pass-the-hash and pass-the-ticket. Windows server security event logs are used as the input source for the application. The project proposes an alert triggering system by monitoring windows logs in case of lateral movement. The monitoring system proposed has achieved the desired efficiency and output.

Keywords: Lateral movement, Cyber security, intrusion, monitoring, alert triggering, security-auditing

1 Introduction

Lateral movement is the technique used to take control of the system and gain access to other systems on the network. The focus of the attackers gets achieved by the reconnaissance stage where they scan the network and identify the vulnerable devices. Attackers may utilize custom remote access to get lateral movement, or they may breach the victim machine's credentials and use operating system tools. When compared to other external tools, using native operating system tools allows the attacker to camouflage more effectively.

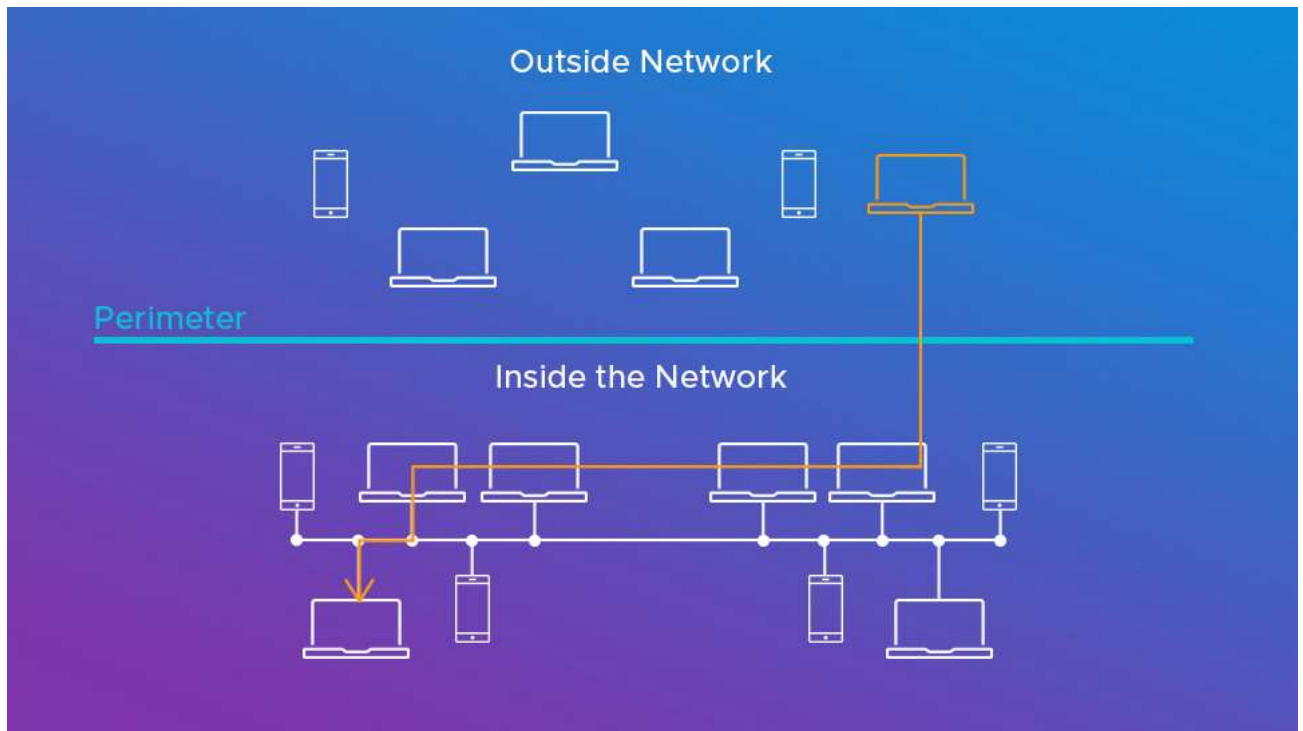


Figure 1: Lateral movement

The perimeter line, as shown in Figure 1, can be a horizontal line that divides the exterior and interior networks. The attacker will begin by moving vertically from outside to within the network. However, after they've joined the network, they'll be able to travel laterally (horizontally) to attain their goal. The horizontal flow is known as "east-west traffic," whereas the vertical movement is known as "north-south traffic."

Lateral movement enables the attacker to stay invisible and continue to retain access even if the attack was detected on the first system. After acquiring access to the endpoint with the help of malware infection or phishing attack, the attacker mimics a genuine user and goes around the network through multiple systems until the goal is achieved (crowdstrike.com, 2021).

This research is focusing on the detection of lateral movement at the Windows operating system level. The detection of the lateral movement attack is important, as failing to detect it in the early stage can lead to the attacker gaining access to the entire network. The attacker can use various techniques of lateral movement to stay hidden inside the network. The detection of the attacker can help to eliminate the threat and can protect the network. If the attacker stays within the network for a longer time, then they can compromise other systems by moving laterally with the help of pass-the-ticket and pass-the-hash techniques.

2 Related Work

For the literature review, the related 4 research papers were chosen to understand the background and need of the research question solution. The following subsection will give a brief overview of all the four papers considered.

2.1 Literature review

The graph-based metric (Purvine, Johnson, and Lo, 2016) analysing research paper mainly focuses on creating the dynamic reachability graph of the network where the attacker exploits different types of vulnerabilities. The reachability graph was used here to develop impact scores at the network level and machine level. The lateral movement mitigation strategies have also been analyzed with the help of the impact scores. The impact scores were generated dynamically as the network gets changed over time and these scores were used to mitigate lateral movement by comparing against a set threshold.

The lateral movement detection proposed known as ‘Latte’ (Liu et al., 2018) uses the graph-based approach to identify the potential lateral movement paths in a network. Here for the network graph, the computers were considered as nodes, and the user logon events were considered as edges. The logon events considered here were focused on the Kerberos service ticket requests which deal with the Pass the ticket lateral movement technique.

The framework (Chen et al., 2019) was proposed to model the lateral movement attacks and the solution to design the enterprise network to fight against the attacks. A tripartite user-host-application graph was created for performance research using event logs and network traffic obtained from a big organization. The suggested edge hardening approach was compared to the NetMelt algorithm, which is a well-known edge removal method for containing information dispersion on a homogeneous graph. Because NetMelt does not exploit the cyber system's heterogeneity, the proposed method (with or without score recalculation) could limit reachability to roughly 10% by hardening less than 1.5 percent of edges, whereas the proposed method (with or without score recalculation) could limit reachability to roughly 10% by hardening less than 1.5 percent of edges.

The machine learning approach (Bai et al., 2019) research based on the RDP session logs was suggested to detect lateral movement in 2019. RDP (Remote Desktop Protocol) is a way used by LM to successfully log on to an unauthorized computer, leaving traces in both the host and network logs. An anomaly-based technique for identifying fraudulent RDP sessions was proposed in this study. The study analyzed various machine learning techniques and concluded that LogitBoost (LB) technique gave better accuracy. For identifying fraudulent Windows RDP sessions, LB demonstrated good results. The limitation in this study was training machine learning models from scratch which will be computationally hard, expensive, and time-consuming.

The multi-detector approach (Bohara et al., 2017) to detect lateral movement attacks uses larger patterns of interactions with the system that an attacker must have after getting initial access. Using basic statistical characteristics linked to Command and Control and Lateral Movement indications, the suggested technique could identify compromised hosts. The findings suggested that the proposed method may accurately detect infected hosts with a low false-positive rate.

The following research (Bian et al., 2021) focuses on using authentication logs to detect lateral movement. Advanced persistent threats (APTs) have drastically increased the frequency of network attacks in recent years. The basic purpose of an APT is to gain unauthorized access to network assets, compromise systems, and steal data. APTs can go unnoticed for lengthy periods due to their stealthy nature, making detection difficult. Machine learning (ML) techniques are utilized in this research to identify potential APT attack targets. The Los Alamos National Lab (LANL) authentication log collection was used to investigate the graph-based characteristics. The researchers were unable to utilize data from numerous sources when locating the target assets due to inadequate flow data supplied in the LANL dataset (TA). Because the volume of data in a corporate network grows rapidly, online learning would be advantageous in terms of efficiency.

2.2 Literature review summary

Table 1: Literature review matrix

Research paper	Source of input	Result format
(Purvine, Johnson and Lo, 2016)	Authentication logs	Impact scores generated with the help of a reachability graph
(Liu et al., 2018)	Windows event logs	The network connection graph
(Chen et al., 2019)	Network traffic and event logs	Tripartite graph
(Bai et al., 2019)	RDP session logs	Precision recall curve
(Bohara et al., 2017)	Features extracted from internal and external communication traffic	True positive rates and False positive rates
(Bian et al., 2021)	Authentication and flow logs	Impact scores

3 Research Methodology

The intrusion detection system (IDS) (Barracuda, 2011) is usually comprised of a system or software which will be monitoring the network traffic for any unusual events which might be the source of the attack or threat. These unusual events will be centrally managed or collected with the help of a log monitoring system. The IDS sitting within the network will monitor the event and trigger an alert to the administrator or the monitoring team to act.

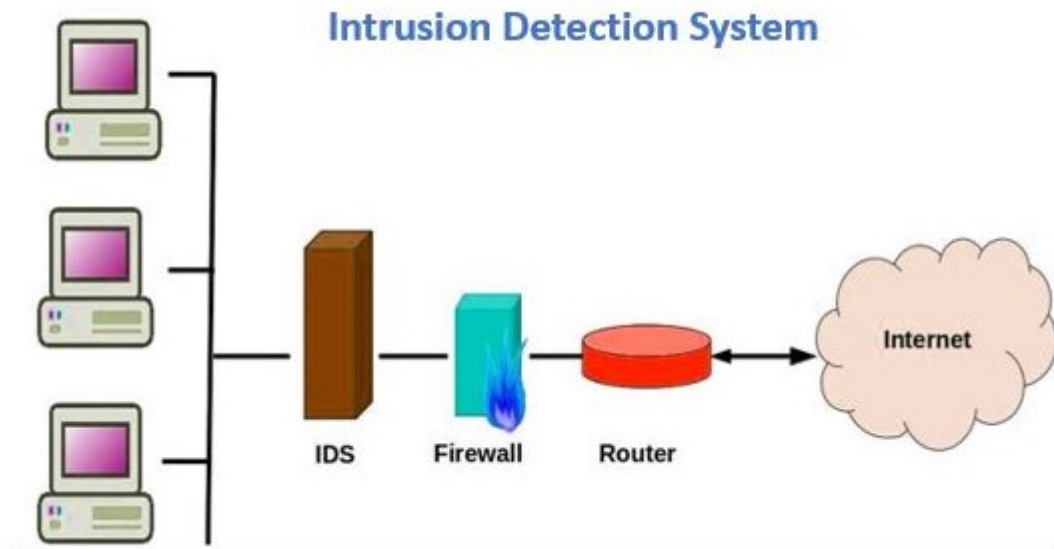


Figure 2: IDS architecture

As we can see in the general structure of an intrusion detection system, the IDS will be placed within the network at the entry after the firewall. The IDS will monitor the traffic from the firewall and the network traffic in the internal computers. The IDS will trigger an alert in case of a security breach.

3.1 Proposed research methodology

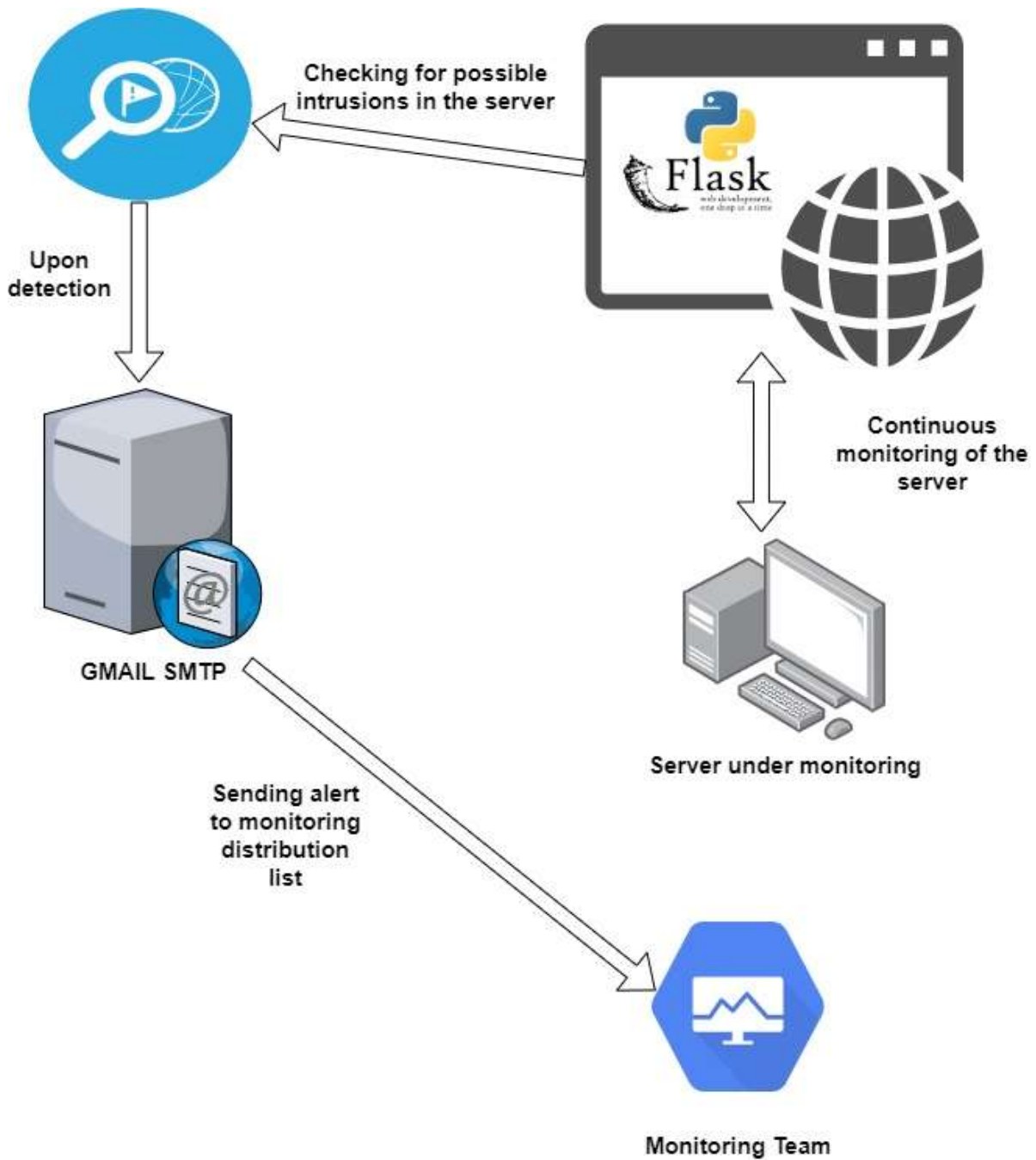


Figure 3: Proposed architecture diagram

The web application will be installed on the server which is under monitoring. The application will be continuously monitoring the server for malicious event logs. Once the application notices the event ID 4648, 4776, 4769, 4624, it will trigger an alert message for the intrusion. Once the alert is generated, the admin can send the email alert to the monitoring team. The monitoring team can look at the alert details along with the attached event dataset to take appropriate actions and mitigate the impact.

The event IDs are under monitoring and the details of the events (Dansimp, 2021) are as follows:

4648: When a process tries to log in to an account by explicitly supplying the account's credentials, this event is triggered.



Figure 4: Event 4648 details

4776: This event is triggered whenever NTLM authentication is used to validate credentials.

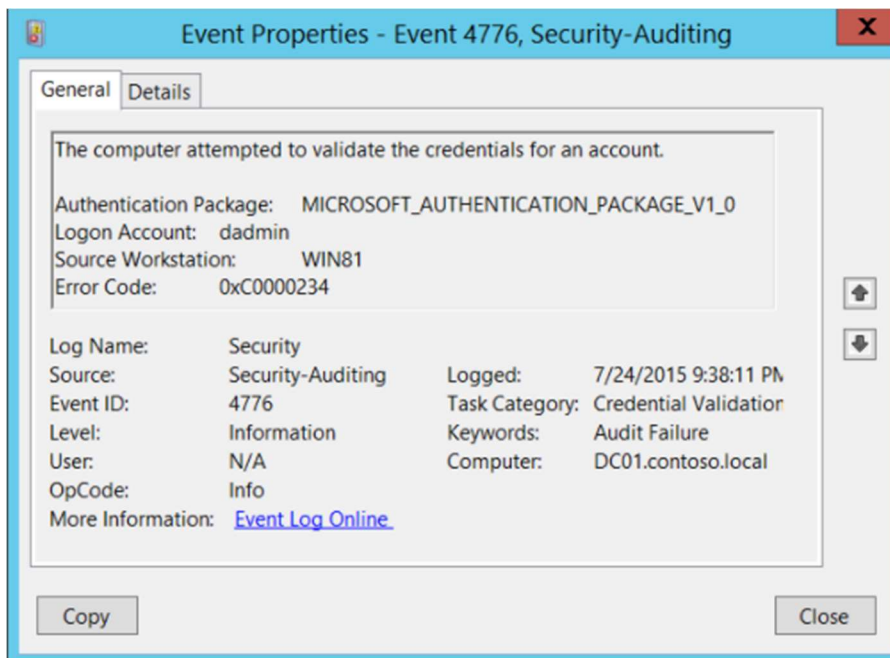


Figure 5: Event 4776 details

4769: When Key Distribution Centre receives a Kerberos Ticket Granting Service (TGS) ticket request, this event is triggered.

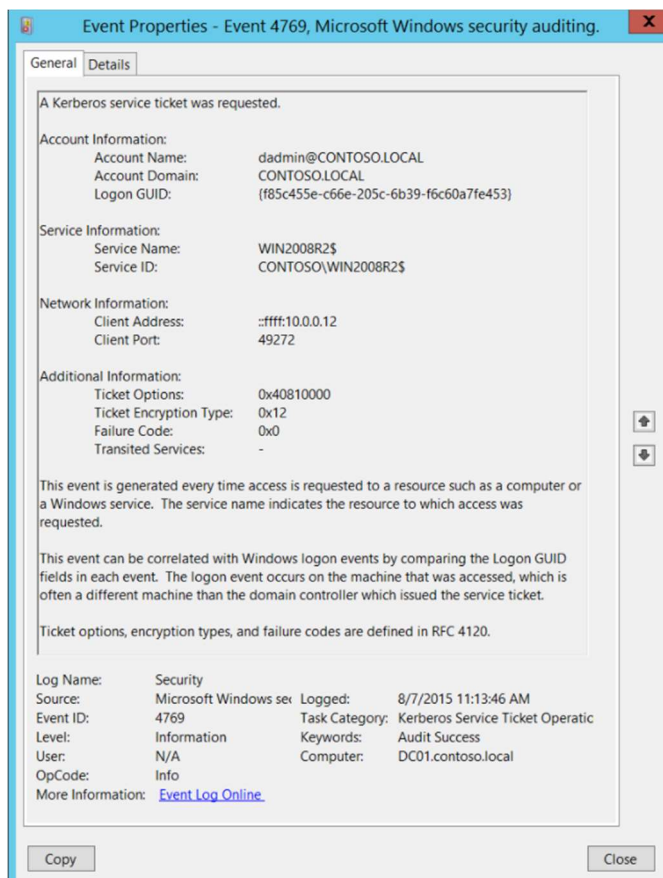


Figure 6: Event 4769 details

4624: When a login session is formed, this event is triggered (on the target machine). It is generated on the machine that was used to access the session.

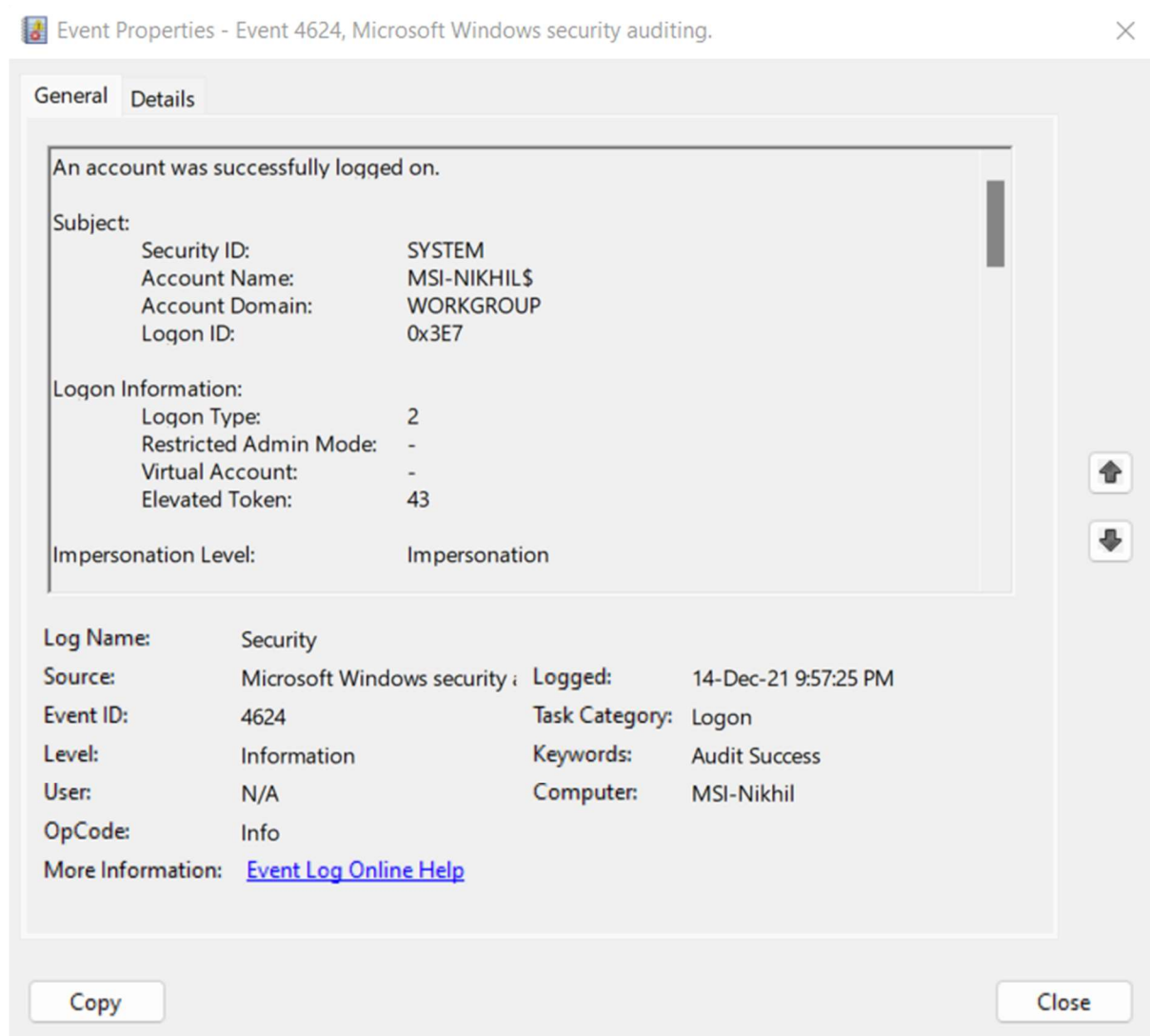


Figure 7: Event 4624 details

Types of attacks monitored:

1. Pass the hash lateral movement attack can be identified when the NTLM credentials were used.
The chronology of the events to confirm this attack is 4648 → 4776 → 4776.
2. Pass the ticket lateral movement attack can be identified when the Kerberos authentication was used.
The chronology of the events to confirm this attack is 4768 → 4769 → 4624.

The input source used for the application is windows server security event logs. The security logs consist of 5 columns – Keywords, Date and Time, Source, Event ID, and Task category.

Keywords – Gives the event type details

Date and Time – Timestamp of the event generated

Source – Source information of the event log

Event ID – Unique number given to the event type

Task category – The events associated with the certain task type

4 Design Specification

The solution for the lateral movement detection was approached here with the help of an automatic alert triggering software. The software is a web-based application built with the help of Python and flask for the back end. The software will monitor the real-time generated windows server security event logs. When certain event IDs get noticed from the software, the system detects the lateral movement and triggers the alert in the dashboard of the application. The application will then allow the admin to send an email alert also to the monitoring team.

The following are the prerequisites and the libraries used to achieve lateral movement detection.

1) Python

Python (Python.org, 2019) is an open-source programming language that is simple and user-friendly. The features include cross-platform language, an extensive collection of standard libraries, object-oriented language, and more.

Here the python was chosen to keep the application structure simple and to make the best use of all the standard libraries to enhance efficiency.

2) Flask

The web framework is written in Python (Tutorialspoint.com, 2019). There are no tools or libraries required for this web framework.

3) Flask-SQLAlchemy

Flask-SQLAlchemy (flask-sqlalchemy.palletsprojects.com, 2021) is a Flask plugin that adds SQLAlchemy functionality to your application. It seeks to make using SQLAlchemy with Flask simpler by providing useful defaults and other utilities that make common tasks easier.

4) Flask-Login

Flask-Login (Readthedocs.io, 2019) is a module that manages user sessions in Flask. It takes care of the usual responsibilities of logging in, logging out, and remembering users' sessions over time.

This library was used in our web application to provide the login feature for the admin.

- 5) Werkzeug
Werkzeug (werkzeug.palletsprojects.com, 2007) is a WSGI web application library with a lot of features. Werkzeug is a German word for “tool”. It started as a basic collection of WSGI application utilities and has evolved into one of the most powerful WSGI utility libraries. There are no dependencies enforced by Werkzeug.
- 6) Pandas
Built on top of the Python programming language, Pandas (Pandas, 2018) are a quick, powerful, versatile, and easy-to-use open-source data analysis and manipulation tool. In our application, the Pandas library is used to convert the windows security event logs into the data frame format.
- 7) Smtplib
The `smtplib` (Python.org, 2020) module creates an SMTP client session object that may be used to send email to any computer on the internet that has an SMTP or ESMTP listener daemon.
In the application, `smtplib` was used to trigger the intrusion alert email to the monitoring team.
- 8) Email
The `email` (docs.python.org, 2021) package is an email message management library. The library `email` was used to structure the body, subject line, and receiver of the alert email.
- 9) CSS
The language that we use to style an HTML document is CSS. CSS (W3schools.com, 2019) specifies how HTML components should appear.
CSS was used to give a better look and style for the application.
- 10) Bootstrap
Bootstrap (Otto, 2000) is a free and open-source CSS framework for front-end web development that is responsive and mobile-first. It includes design templates for typography, forms, buttons, navigation, and other interface elements that are based on CSS and JavaScript.
The bootstrap has been used to make the front-end of the web application.
- 11) SQLite
DB Browser for SQLite (DB4S) is an open-source, high-quality visual tool for creating, designing and editing SQLite database files.
SQLite is used in the application to store the login credentials of the admin.

5 Implementation

The monitoring team in any organization will be working 24*7 to act on any threats which cause an impact on normal operations. The team will try to resolve the things or will quickly divert the threat to the respected team to act. The intrusion detection system has become necessary in any organization to handle cyber security threats. As the name suggests the system in place will provide an alert whenever an intrusion or the unexpected even occurs in the network traffic and communication.

The solution implementation for the research question was achieved with the help of a web application that continuously monitors the security event logs. The following section will discuss more on the functionality of the application.

1. The main.py is the main application file that should be run with the command “python3 main.py” to start the application.
2. Now the terminal output will provide the URL to access the application.
3. The application is hosted on the localhost and the port number 5000.
4. The URL will look like <http://127.0.0.1:5000/>
5. By accessing the URL, the application will prompt the admin to enter the username and password.
6. By successful authentication, the admin can access the application dashboard and use the below-mentioned functionalities.

5.1 Implementation solution

The web application proposed in this research contains 3 main functionalities.

5.1.1 View logs

Here the windows server security event logs can be viewed and tracked directly from the web application dashboard.

The event logs were initially taken from the CSV format and stored into the data frame with the help of the Pandas library. The ‘read_csv’ method was used to read the log contents into the Pandas data frame.

5.1.2 Check intrusion

This functionality allows the admin to check if there is any intrusion by using the lateral attack movement technique. The lateral movement attack type is decided by the unique event ID occurrence in the logs. The two types of attack are identified here based on the below Event ID occurrence:

1. Pass the hash attack
4648 – 1st event ID
4776 - 2nd event ID

4776 – 3rd event ID

The above lateral movement attack will make use of the NTLM credentials to get access to the other accounts to move laterally within the network.

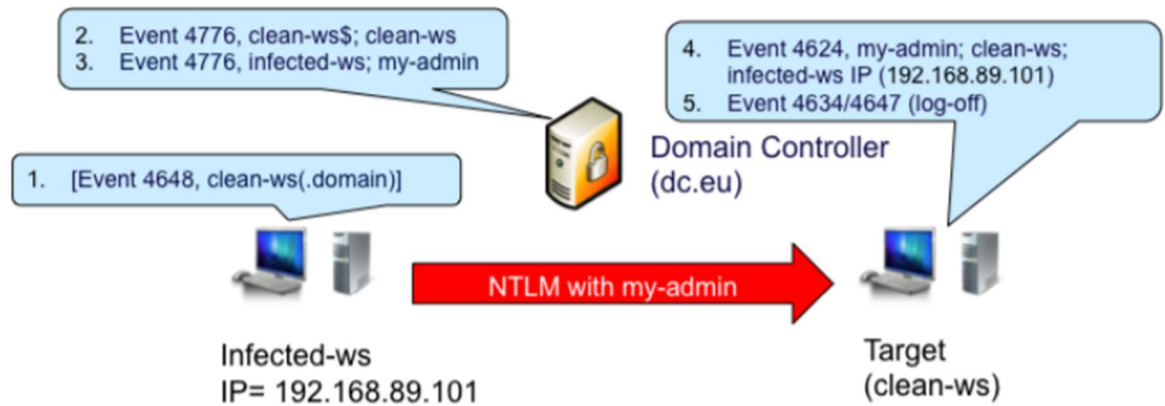


Figure 8: NTLM credential related logs

2. Pass the ticket attack

4768 – 1st event ID

4769 - 2nd event ID

4624 – 3rd event ID

The above lateral movement attack will make use of the Kerberos authentication ticket and Kerberos service ticket to get access to the other accounts to move laterally within the network.

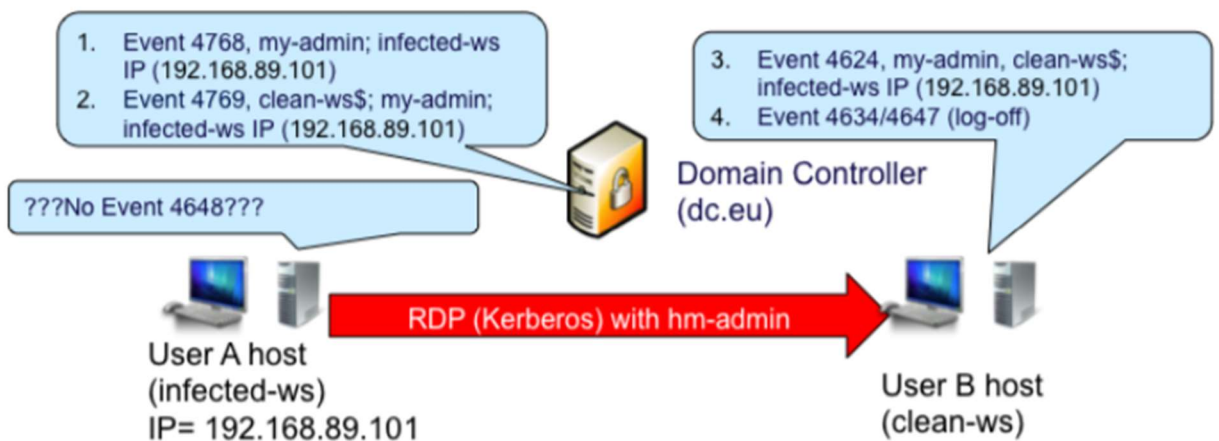


Figure 9: Kerberos authentication related logs

5.1.3 Send Mail

This feature allows the admin to send the alert email to the monitoring team to take appropriate actions and mitigate the impact. The alert email will be attached with the intrusion information along with the security event logs available then. This automatic email triggering will help the monitoring team to action without any delay or downtime. HTTP GET and POST methods were used with the ‘email’ python library to incorporate this feature into the web application. The google smtp server was used to trigger the email into the target recipient.

6 Evaluation

The proposed system will scan the stored security event logs for any intrusion which has been made using lateral movement type of attack. The system will mainly focus on two types of lateral movement attack – pass the hash and pass the ticket. The system will send an intrusion alert email to the monitoring team upon the intrusion detection. The monitoring team will notice the alert email along with event log details and take immediate action to mitigate the impact.

The following sections represent the different experiments conducted along with the results obtained.

6.1 Experiment 1 – View logs

This experiment shows the first functionality of the proposed solution. Using the “view logs” option the security event logs generated can be monitored. All the events will be listed under this option as shown in the following figure. The event logs displayed will have all the details including keywords, date and time, source, event ID and task category.

The below figure displays the output of security event logs in the web application dashboard.

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	24-11-21 15:10	Microsoft-Windows-Security-Auditing	4798	User Account Management
Audit Success	24-11-21 15:10	Microsoft-Windows-Security-Auditing	4798	User Account Management
Audit Success	24-11-21 15:10	Microsoft-Windows-Security-Auditing	4648	Logon
Audit Success	24-11-21 15:10	Microsoft-Windows-Security-Auditing	4648	User Account Management
Audit Success	24-11-21 15:10	Microsoft-Windows-Security-Auditing	4798	User Account Management
Audit Failure	24-11-21 15:10	Security-Auditing	4776	Credential Validation
Audit Success	24-11-21 15:10	Microsoft-Windows-Security-Auditing	4798	User Account Management
Audit Success	24-11-21 15:10	Microsoft-Windows-Security-Auditing	4798	User Account Management
Audit Success	24-11-21 15:08	Microsoft-Windows-Security-Auditing	4798	User Account Management
Audit Failure	24-11-21 15:10	Security-Auditing	4776	Credential Validation
Audit Success	24-11-21 15:08	Microsoft-Windows-Security-Auditing	4798	User Account Management
Audit Success	24-11-21 15:08	Microsoft-Windows-Security-Auditing	4769	Kerberos Service Ticket Operations
Audit Success	24-11-21 15:08	Microsoft-Windows-Security-Auditing	4798	User Account Management

Figure 10: Security event logs in the dashboard

6.2 Experiment 2: Check intrusion

This experiment shows the second functionality of the proposed solution. Using the “check intrusion” option the generated event logs can be checked to detect any intrusion using lateral movement type of attack. By clicking the “check intrusion” option, the security logs will be scanned to check the possible intrusion and provide the confirmation as shown in the following figure.

The below figure displays the status of the intrusion detection in the available logs.



Figure 11: Intrusion detection results on the application dashboard

6.3 Experiment 3: Send mail

This experiment shows the third functionality of the proposed solution. Using the “send mail” option the alert mail can be sent to the monitoring team. By clicking the “send mail” option, the alert mail will be sent to the monitoring team to take the necessary actions to mitigate the impact as shown in the following figure.

The below figure shows the resultant alert mail received after the intrusion gets detected.

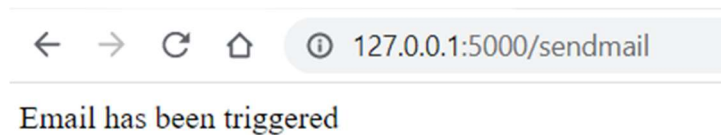


Figure 12: The email triggered message on the application dashboard

The below figure displays the alert email received by the monitoring team along with the security event log dataset for reference.

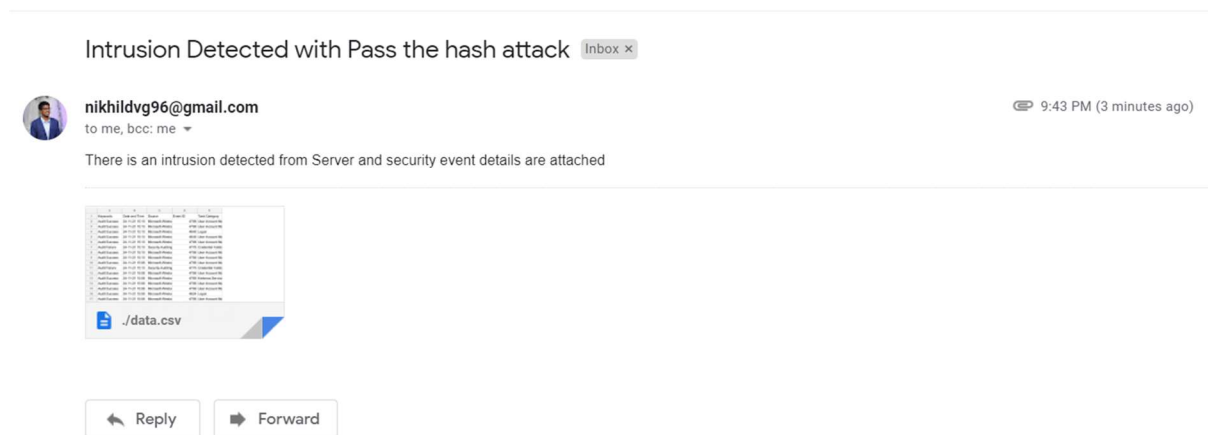


Figure 13: Pass the hash attack alert received by the monitoring team

Intrusion Detected with Pass the ticket Attack Inbox ×



nikhildvg96@gmail.com
to me, bcc: me

9:43 PM (4 minutes ago)

There is an intrusion detected from Server and security event details are attached



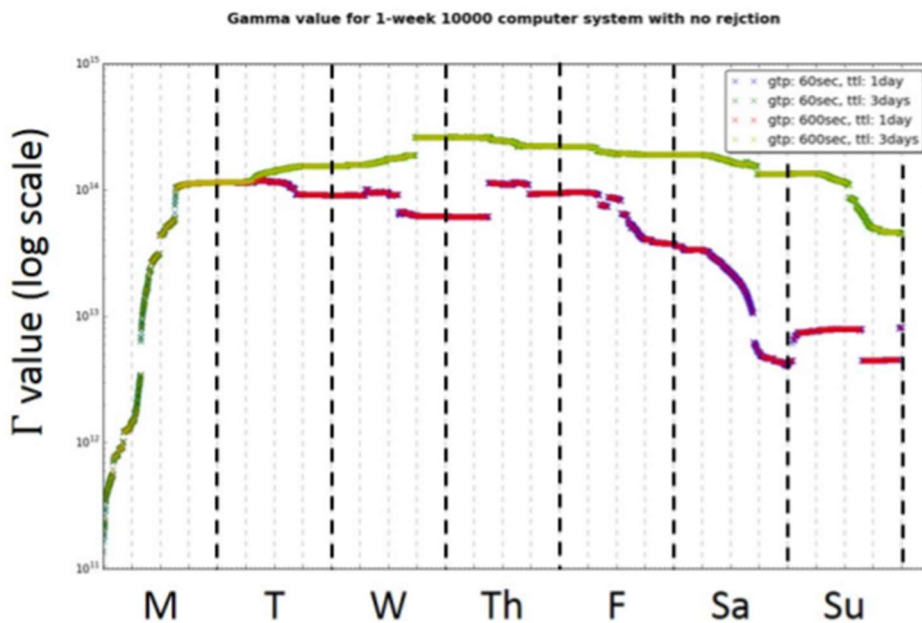
[Reply](#) [Forward](#)

Figure 14: Pass the ticket attack alert received by the monitoring team

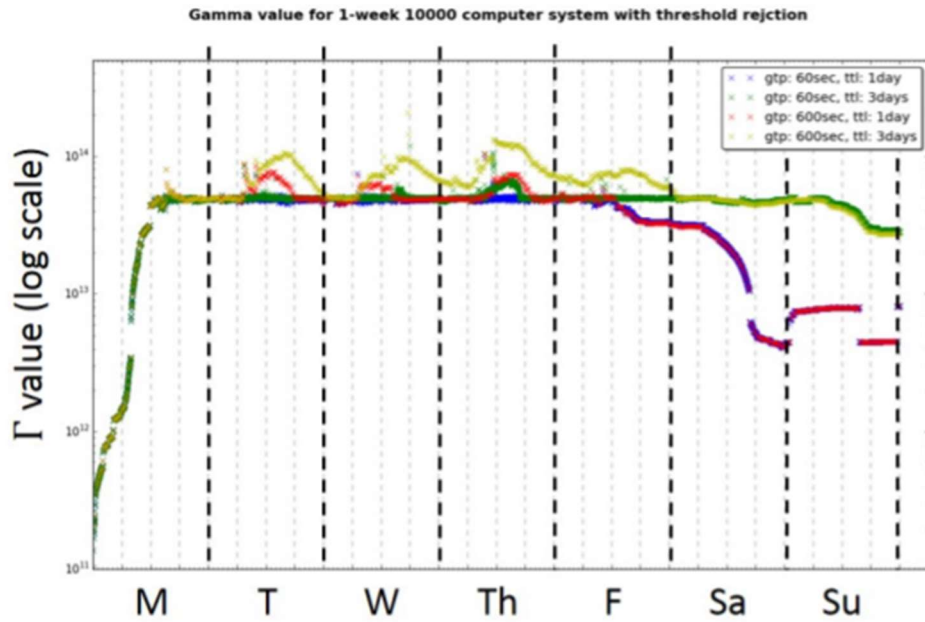
6.4 Discussion

The results obtained using the experiments carried out are provided with application snapshots have been provided under the evaluation section.

The existing approach in one of the research papers (Purvine, Johnson and Lo, 2016) proposes a graph-based impact metric system to mitigate the lateral movement attacks by setting a threshold. The approach uses evolution of impact metric Γ with 10,000 computer systems for a week. The following figure shows the obtained results, before and after setting the threshold for detection.



(a) No threshold



(b) Threshold $\theta = 5 \times 10^{13}$.

Figure 15: Evolution of impact metric Γt over one week of time. Each color indicates a different Γt calculation frequency (f) and credential time to live (ttl). Blue: $f = 1$ minute, $ttl = 1$ day. Green: $f = 1$ minute, $ttl = 3$ days. Red: $f = 10$ minutes, $ttl = 1$ day. Yellow: $f = 10$ minutes, $ttl = 3$ days.

In the proposed approach, the results are focused on providing an immediate way to detect a real-time lateral movement attack and take action to mitigate the impact at the earliest. The event logs are checked to detect the two types of lateral movement attacks – pass the hash and pass the ticket attack. The results are obtained in the form of alert generation in the web application built and the email alert sent to the monitoring team along with the event log details.

As the above experiment results show that all the functionalities of the web application are working as expected. The intrusion checking functionality successfully identified the lateral movement attack. The program is efficient and fast enough to process a small set of event logs. The application might take more time if the dataset size increases with the number of security events. As the application is designed and hosted in the local system, intrusion detection is possible for one server now.

7 Conclusion and Future Work

The lateral movement detection in the windows operating system was approached in this research project with the help of Windows server security event logs. The proposed solution for the research area was to create an automatic alert system to detect lateral movement type of attack and notify the admin. The results obtained after the design and implementation of the project have proved that the desired output was achieved. The application built here used the logs to monitor and detect the attack type based on the unique windows event IDs. The current solution proposed is currently processing the single server logs in the local environment. In the future, I'll try to implement a system on the cloud to centrally monitor the logs of multiple servers. I'll also explore the different operating systems like Mac OS, Linux, etc.

References

Bai, T., Bian, H., Daya, A.A., Salahuddin, M.A., Limam, N. and Boutaba, R. (2019). *A Machine Learning Approach for RDP-based Lateral Movement Detection*. [online] IEEE Xplore. Available at: <https://ieeexplore.ieee.org/document/8990853>.

Barracuda (2011). *What is an Intrusion Detection System?*. [online] Barracuda.com. Available at: <https://www.barracuda.com/glossary/intrusion-detection-system>.

Bian, H., Bai, T., Salahuddin, M.A., Limam, N., Daya, A.A. and Boutaba, R. (2021). *Uncovering Lateral Movement Using Authentication Logs*. IEEE Transactions on Network and Service Management, [online] 18(1), pp.1049–1063. Available at: <https://ieeexplore.ieee.org/document/9335647>.

Bohara, A., Nouredine, M.A., Fawaz, A. and Sanders, W.H. (2017). *An Unsupervised Multi-Detector Approach for Identifying Malicious Lateral Movement*. [online] IEEE Xplore. Available at: <https://ieeexplore.ieee.org/document/8069085>.

Chen, P.-Y., Choudhury, S., Rodriguez, L., Hero, A. and Ray, I. (2019). *Enterprise Cyber Resiliency Against Lateral Movement: A Graph Theoretic Approach*. arXiv:1905.01002 [cs]. [online] Available at: <https://arxiv.org/abs/1905.01002>.

crowdstrike.com. (2021). *Lateral Movement Explained | What is Lateral Movement?* [online] Available at: <https://www.crowdstrike.com/cybersecurity-101/lateral-movement/>.

Dansimp (2021). *Advanced security audit policies (Windows 10) - Windows security*. [online] docs.microsoft.com. Available at: <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/advanced-security-auditing> [Accessed 14 Dec. 2021].

docs.python.org. (2021). *email — An email and MIME handling package — Python 3.8.2 documentation*. [online] Available at: <https://docs.python.org/3/library/email.html>.

flask-sqlalchemy.palletsprojects.com. (2021). *Flask-SQLAlchemy — Flask-SQLAlchemy Documentation (2.x)*. [online] Available at: <https://flask-sqlalchemy.palletsprojects.com/en/2.x/>.

Liu, Q., Stokes, J.W., Mead, R., Burrell, T., Hellen, I., Lambert, J., Marochko, A. and Cui, W. (2018). *Latte: Large-Scale Lateral Movement Detection*. [online] IEEE Xplore. Available at: <https://ieeexplore.ieee.org/document/8599748>.

Otto, M. (2000). *Bootstrap*. [online] Getbootstrap.com. Available at: <https://getbootstrap.com/>.

Pandas (2018). *Python Data Analysis Library — pandas: Python Data Analysis Library*. [online] Pydata.org. Available at: <https://pandas.pydata.org/>.

Purvine, E., Johnson, J.R. and Lo, C. (2016). *A Graph-Based Impact Metric for Mitigating Lateral Movement Cyber Attacks*. Proceedings of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense - SafeConfig'16.

Python.org. (2019). *Welcome to Python.org*. [online] Available at: <https://www.python.org/about/>.

Python.org. (2020). *smtplib — SMTP protocol client — Python 3.8.1 documentation*. [online] Available at: <https://docs.python.org/3/library/smtplib.html>.

Readthedocs.io. (2019). *Flask-Login — Flask-Login 0.4.1 documentation*. [online] Available at: <https://flask-login.readthedocs.io/en/latest/>.

Tutorialspoint.com. (2019). *Flask Tutorial - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/flask/index.htm>.

W3schools.com. (2019). *CSS Tutorial*. [online] Available at: <https://www.w3schools.com/css/>.

werkzeug.palletsprojects.com. (2007). *Werkzeug — Werkzeug Documentation (2.0.x)*. [online] Available at: <https://werkzeug.palletsprojects.com/en/2.0.x/>.