DETECT-U

[1] **Ms. Bhavna Chaudhari**, Assistant Professor, Information Technology Engineering Department,
Progressive Education Society's Modern College of Engineering, Pune - 411 005 (Maharashtra)(India)

[2]Student VIII Semester B.E(Information Technology),
Gaurav Darawade[1], Omkar Gadekar[2], Chaitanya Dhamal[3], Vaibhav Wavhal[4]

Progressive Education Society's Modern College of Engineering, Pune - 411 005 (Maharashtra)(India)

---***---

Abstract

Intrusion detection systems are crucial for identifying attacks on computer systems and networks. This project focuses on developing a Host-based Intrusion Detection System (HIDS) called DETECT-U. The system comprises three components: the client-side, the server-side, and the analyzer. The client-side records and stores the running processes in a CSV format and transmits them to the server. The server, which is in a listening state, cleans the CSV file and analyzes it using the analyzer module. The analyzer applies a set of predefined rules to identify suspicious files. If any suspicious files are detected, the server sends an alert in the form of a pop-up notification to the client. The entire system is implemented in Python and operates through command-line programs.

Key Words: intrusion detection systems, HIDS, DETECT-U, client-side, server-side, analyser, Python.

1.INTRODUCTION

Intrusion-detection systems play a crucial role in safeguarding computer systems, networks, and information systems against various forms of attacks. The focus of this project is to develop a Host-based Intrusion Detection System (HIDS) named DETECT-U, which comprises three key components: the client side, the server side, and the analyzer.

The client side of DETECT-U is responsible for recording and storing information about running processes in a CSV format. This data is then transmitted to the server, which remains in a listening phase, ready to receive and process the information. Upon receiving the CSV file, the server performs necessary cleaning operations and analyzes the data using a set of predefined rules implemented in the analyzer.

The analyzer acts as the intelligence behind DETECT-U, leveraging the provided rules to identify suspicious files or activities within the recorded process data. Whenever the server detects any anomalies, it promptly alerts the client by generating a pop-up alert. All these elements are based on Python and operate as command-line programs, ensuring ease of use and compatibility across different systems.

The main objective of this project is to enhance the detection capabilities of intrusion attempts on host systems. By implementing DETECT-U, we aim to provide a reliable and efficient means of identifying and responding to potential security breaches, ultimately contributing to the overall resilience of computer networks and information systems.

1.1 **Motivation**:

The motivation behind this project stems from the escalating threat landscape in the digital realm. Cyberattacks continue to evolve, becoming more sophisticated and harder to detect. Traditional security measures often fall short in identifying emerging threats in real-time, emphasizing the necessity for robust intrusion detection systems. By developing DETECT-U, we aim to enhance the security posture of host systems, enabling proactive detection and response to potential intrusions.

1.2 Scope:

The scope of this project encompasses the development of a comprehensive HIDS solution that comprises three key components: the client-side, the server-side, and the analyzer. The client-side is responsible for recording and storing the running processes of the host system in a structured CSV format. These process logs are then transmitted to the server-side, which operates in a listening state, ready to receive and process the incoming data. The server-side component performs data cleaning operations on the CSV file and employs an analyzer module that utilizes a predefined set of rules for identifying suspicious files or activities. Upon detecting any anomalies, the server-side generates an alert in the form of a pop-up notification, which is promptly delivered to the client-side for immediate action.

The implementation of DETECT-U revolves around the utilization of Python programming language, with all the components functioning as command-line programs. This design choice ensures compatibility, flexibility, and ease of integration within existing host systems.

By developing DETECT-U, we aim to provide organizations and individuals with a reliable and efficient HIDS solution that enhances the security posture of their host systems. The system's architecture and implementation in Python allow for easy customization and scalability, empowering users to adapt it to their specific security requirements.

2. Literature Survey

[1] M. Ozkan-Okay, R. Samet, Ö. Aslan and D. Gupta, "A Comprehensive Systematic Literature Review on Intrusion Detection Systems," in *IEEE Access*, vol. 9, pp. 157727-

157760, 2021, doi: 10.1109/ACCESS.2021.3129336. Detecting intrusions effectively in computer networks remains a challenging task due to the evolving tactics employed by cyber attackers. With the continuous addition of new devices to networks, security concerns have also increased. To ensure efficient network flow management and proactive security measures, a comprehensive understanding of intrusion detection system (IDS) components, methodologies, approaches, attack types, protection mechanisms, and recent scientific studies is crucial. This paper presents an in-depth exploration of intrusion detection technologies, methodologies, and approaches, as well as an investigation into emerging attack types, protection mechanisms, and recent advancements in the field. Furthermore, it discusses available datasets, well-known IDS tools, and the advantages and disadvantages associated with specific IDSs. This scientific review serves as a valuable roadmap for researchers and industry professionals focusing on IDSs.

3. Body of Paper

The body of this paper is organized into several sections to present the main findings and discussions related to the development and implementation of the DETECT-U Host-based Intrusion Detection System (HIDS). These sections aim to provide a comprehensive understanding of the system's architecture, functionalities, and its significance in enhancing the detection capabilities of intrusion attempts on host systems.

Section 1: System Architecture
In this section, we present a detailed overview of the architecture of DETECT-U. We describe the client side, server side, and analyzer components, their interactions, and their roles in the overall functioning of the system. Additionally, we discuss the utilization of Python as the primary programming language for implementing these components and highlight the command-line nature of the programs.

Section 2: Client-Side Operation
Here, we delve into the operations and responsibilities of the client side of DETECT-U. We explain how it records and stores information about running processes in a CSV format. Furthermore, we discuss the process of transmitting the collected data to the server and the importance of maintaining data integrity during transmission.

Section 3: Server-Side Operation
In this section, we focus on the server side of DETECT-U. We elaborate on its role in receiving the CSV file from the client and initiating necessary data cleaning procedures. Additionally, we explore the implementation of the analyzer and its use of predefined rules to analyze the processed data for suspicious files or activities.

Section 4: Alert Generation and Response
Here, we discuss the crucial aspect of alert generation and response within DETECT-U. We outline the mechanisms through which the server detects anomalies based on the analyzer's findings and promptly sends pop-up alerts to the client. We emphasize the significance of real-time alerts in enabling swift and effective responses to potential security breaches.

Section 5: Evaluation and Performance Analysis
In this section, we present the evaluation results and performance analysis of DETECT-U. We discuss the effectiveness of the system in detecting various types of intrusion attempts and provide insights into its accuracy, efficiency, and resource utilization. Additionally, we compare DETECT-U with existing intrusion-detection systems to highlight its advantages and limitations.

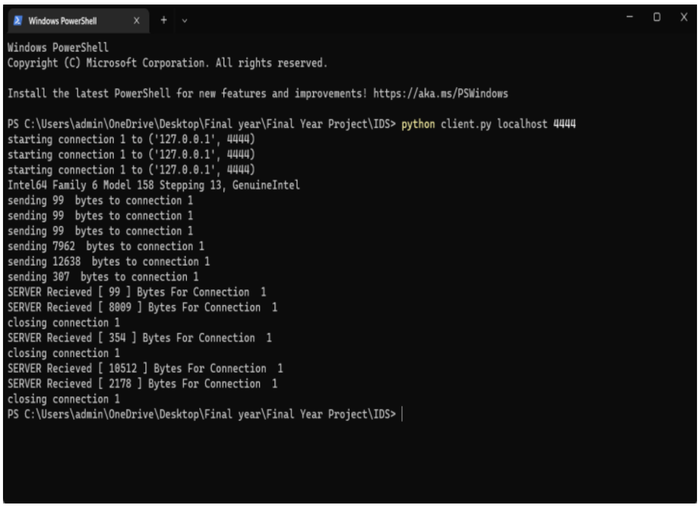
Section 6: Discussion and Future Work
Here, we engage in a comprehensive discussion of the findings presented in the previous sections. We analyze the implications of DETECT-U's capabilities in enhancing the security posture of host systems and networks. Furthermore, we identify potential areas for future research and development, aiming to further improve the system's functionality and effectiveness.

Section 7: Conclusion
In the final section, we summarize the key findings and contributions of this study. We highlight the significance of DETECT-U as a host-based intrusion detection system based on Python and command-line programs. We emphasize its potential in improving the resilience and security of computer systems and networks. Finally, we conclude with an invitation for further exploration and adoption of DETECT-U in real-world environments.

By organizing the body of this paper into these sections, we provide a comprehensive and structured presentation of the development, operation, and evaluation of the DETECT-U Host-based Intrusion Detection System, fostering clarity and facilitating further research and implementation in the field of intrusion detection.

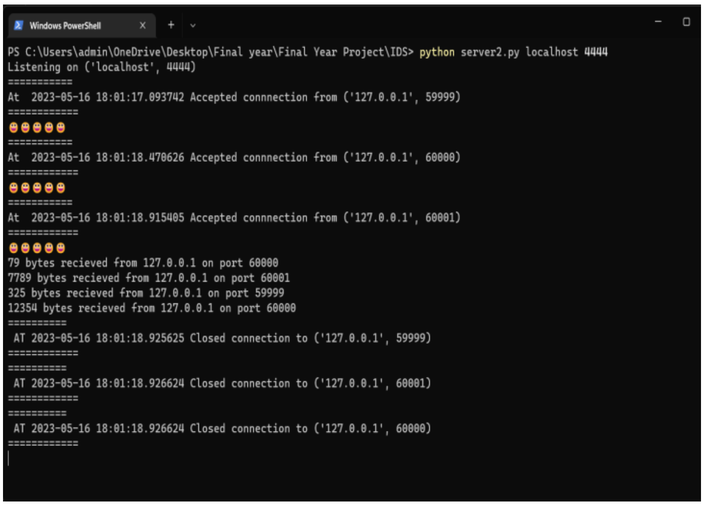
4. Requirement Specifications

The requirement specifications for the DETECT-U Host-based Intrusion Detection System (HIDS) presented above demonstrate the necessary environment for seamless operation and effective detection of intrusions. With a focus on system compatibility and performance, the DETECT-U HIDS supports popular operating systems such as Ubuntu 16.04 to 17.10 and Windows 7 to 11, ensuring flexibility for users. The requirement of an x86 64-bit CPU architecture, coupled with a minimum of 4 GB RAM and 5 GB free disk space, guarantees optimal system resources for efficient intrusion detection. Additionally, the reliance on Python as the programming language empowers users with its versatility and ease of integration. By adhering to these requirement specifications, organizations can confidently deploy the DETECT-U HIDS, bolstering their security infrastructure and safeguarding their critical systems and networks.



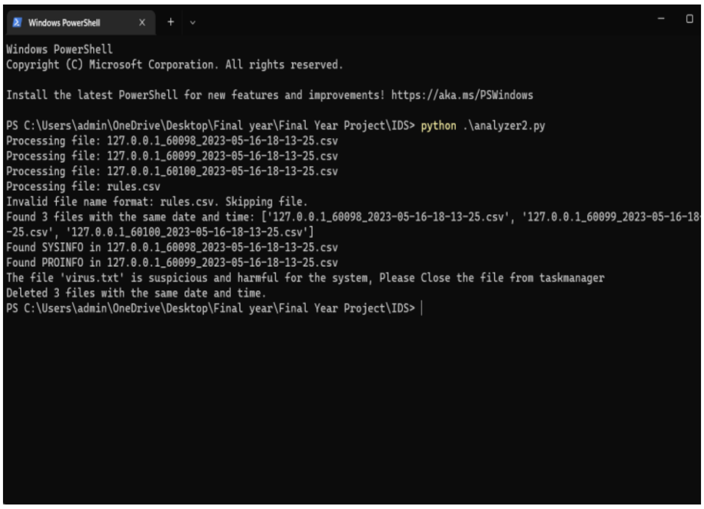
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\admin\OneDrive\Desktop\Final year\Final Year Project\IDS> python client.py localhost 4444
starting connection 1 to ('127.0.0.1', 4444)
starting connection 1 to ('127.0.0.1', 4444)
starting connection 1 to ('127.0.0.1', 4444)
Intel64 Family 6 Model 158 Stepping 13, GenuineIntel
sending 99  bytes to connection 1
sending 99  bytes to connection 1
sending 99  bytes to connection 1
sending 7962  bytes to connection 1
sending 12638  bytes to connection 1
sending 307  bytes to connection 1
SERVER Recieved [ 99 ] Bytes For Connection  1
SERVER Recieved [ 8009 ] Bytes For Connection  1
closing connection 1
SERVER Recieved [ 354 ] Bytes For Connection  1
closing connection 1
SERVER Recieved [ 10512 ] Bytes For Connection  1
SERVER Recieved [ 2178 ] Bytes For Connection  1
closing connection 1
PS C:\Users\admin\OneDrive\Desktop\Final year\Final Year Project\IDS> |
```

Fig -1: Client-Side



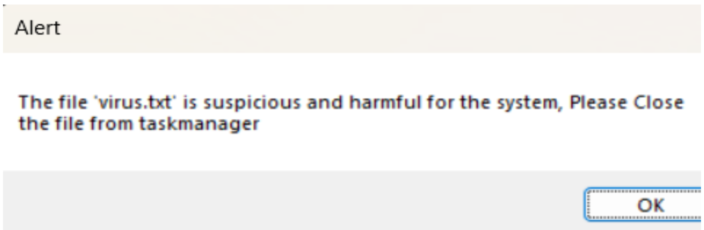
```
PS C:\Users\admin\OneDrive\Desktop\Final year\Final Year Project\IDS> python server2.py localhost 4444
Listening on ('localhost', 4444)
===========
At  2023-05-16 18:01:17.093742 Accepted connnection from ('127.0.0.1', 59999)
============
😀😀😀😀😀
===========
At  2023-05-16 18:01:18.470626 Accepted connnection from ('127.0.0.1', 60000)
============
😀😀😀😀😀
===========
At  2023-05-16 18:01:18.915405 Accepted connnection from ('127.0.0.1', 60001)
============
😀😀😀😀😀
79 bytes recieved from 127.0.0.1 on port 60000
7789 bytes recieved from 127.0.0.1 on port 60001
325 bytes recieved from 127.0.0.1 on port 59999
12354 bytes recieved from 127.0.0.1 on port 60000
==========
 AT 2023-05-16 18:01:18.925625 Closed connection to ('127.0.0.1', 59999)
============
==========
 AT 2023-05-16 18:01:18.926624 Closed connection to ('127.0.0.1', 60001)
============
==========
 AT 2023-05-16 18:01:18.926624 Closed connection to ('127.0.0.1', 60000)
============
|
```

Fig -2: Server-Side



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\admin\OneDrive\Desktop\Final year\Final Year Project\IDS> python .\analyzer2.py
Processing file: 127.0.0.1_60098_2023-05-16-18-13-25.csv
Processing file: 127.0.0.1_60099_2023-05-16-18-13-25.csv
Processing file: 127.0.0.1_60100_2023-05-16-18-13-25.csv
Processing file: rules.csv
Invalid file name format: rules.csv. Skipping file.
Found 3 files with the same date and time: ['127.0.0.1_60098_2023-05-16-18-13-25.csv', '127.0.0.1_60099_2023-05-16-18
-25.csv', '127.0.0.1_60100_2023-05-16-18-13-25.csv']
Found SYSINFO in 127.0.0.1_60098_2023-05-16-18-13-25.csv
Found PROINFO in 127.0.0.1_60099_2023-05-16-18-13-25.csv
The file 'virus.txt' is suspicious and harmful for the system, Please Close the file from taskmanager
Deleted 3 files with the same date and time.
PS C:\Users\admin\OneDrive\Desktop\Final year\Final Year Project\IDS> |
```

Fig -3: Analyzer



The file 'virus.txt' is suspicious and harmful for the system, Please Close the file from taskmanager

Fig -4: Pop-Up Alert

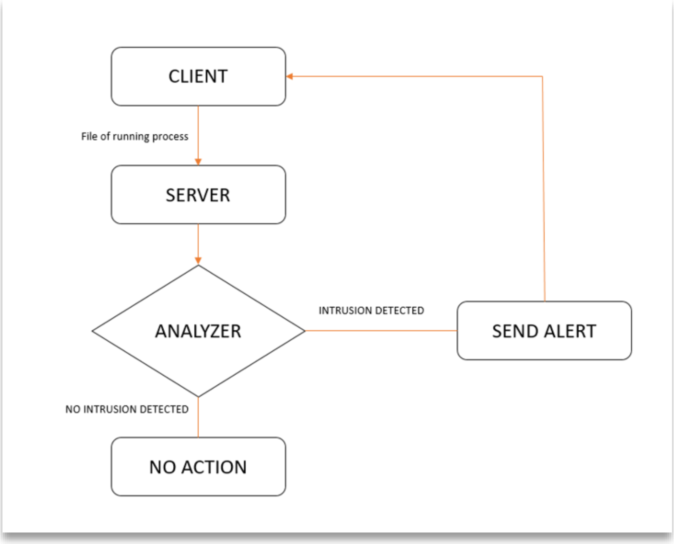


Fig -5: Flow Chart

3. CONCLUSIONS

In this paper, we presented DETECT-U, a Host-based Intrusion Detection System (HIDS) designed to detect attacks against computer systems and networks. The system comprises three main components: the client side, the server side, and the analyzer. The client side records and stores running processes, which are then sent to the server for analysis. The server side cleans and analyzes the data using Python-based command-line programs. If any suspicious files are detected, an alert is sent to the client in the form of a pop-up notification.

The proposed system demonstrates the effectiveness of using a HIDS approach to enhance the security of information systems. By monitoring and analyzing process data, DETECT-U provides an additional layer of defense against potential intrusions and malicious activities. The use of Python and command-line programs ensures flexibility and ease of implementation.

Future work involves further refining the rules and analysis algorithms of the system to enhance its accuracy and detection capabilities. Additionally, integrating real-time monitoring and incorporating machine learning techniques could be explored to improve the system's ability to identify new and evolving threats.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to all those who have contributed to the completion of this project. Special thanks go to the mentors and advisors for their valuable guidance and support throughout the development process. The authors would also like to acknowledge the contributions of the research team members for their assistance and collaboration.

REFERENCES

1. M. Elbasiony, Reda, et al, “A hybrid network intrusion detection framework based on random forests and weighted k-means,” Ain Shams Engineering Journal, Vol. 4, No. 4, pp.753-762, 2013.
2. S. A. Joshi and Varsha S. Pimprale, “Network Intrusion Detection System (NIDS) based on data mining,” International Journal of Engineering Science and Innovative Technology (IJESIT), Vol. 2, No. 1, pp. 95-98, 2013.
3. https://www.trp.org.in/issues/a-literature-survey-on-the-importance-of-intrusion-detection-system-for-wireless-networks
4. https://www.researchgate.net/publication/356368631_A_Comprehensive_Systematic_Literature_Review_on_Intrusion_Detection_Systems