Host Based Intrusion Detection System Using Audit Trails

Harsh Jha[a], Renu Mishra[b] and Amit Kumar Upadhyay[c]

[a,b,c] Sharda School of Engineering and Technology ,Computer Science and Engineering,
Sharda University,Greater Noida ,UP,India

Introduction

Intrusions are any unauthorised activities that damage an information system. This implies that any attack that might jeopardise the availability, secrecy, or purity of the information will be regarded as an incursion. For instance, breaches are activities that would prohibit legitimate users from using computer services. An intrusion detection system, also known as an IDS, is a component of hardware or software that monitors computer networks for malicious behaviour in order to maintain system security. The goal of an IDS is to identify different types of malicious network activity and computer activity that a regular firewall is unable to identify. Every network and host should always be equipped with IDS. For networks, we have network-based intrusion detection systems (NIDS), and for hosts, we have host-based intrusion detection systems (HIDS). Based on the detection pattern, our systems can be roughly divided into two groups: SIDS (Signature-based Intrusion Detection Systems) and HIDS (Host-based Intrusion Detection Systems) (AIDS).

Intrusion data sources

We discussed IDS in the previous section and how they may be categorised depending on the data sources utilised as input to find anomalous activity. Host-based IDS (HIDS) and Network-based IDS are the two main types of IDS technology in terms of data sources (NIDS). The host system's operating system, window server logs, firewall logs, application system audits, and database logs are just a few examples of the audit sources that HIDS examine. The use of a pendrive or direct access to the device are examples of assaults that HIDS can identify without the need of network data. Packet capture, NetFlow, and other network data sources are used by NIDS to monitor network traffic that has been collected from a network. IDS on the network may be used to keep track of a large number of machines connected to a network. NIDS has limited ability to inspect all data in a high bandwidth network due to the volume of data passing through contemporary high-speed communication networks. NIDS is able to monitor the external malicious activities that could be initiated from an external threat at an earlier phase, before the threats spread to another computer system. Together with HIDS and firewalls, NIDS installed at various points within a specific network architecture can offer a robust, multi-tier defence against outside and internal threats. summarises the contrasts between HIDS and NIDS. A HIDS approach using discontinuous system call patterns was proposed by Creech et al., in an effort to reduce false alarm rates while increasing detection rates. The fundamental concept is to employ a HIDS that will use advance audit data provided by an application, and this advance data will then be used to generate alerts and inform the admin about the intrusion as well as his/her activities after the intrusion and his/her identification through photo taking. In order to improve detection performance, a variety of AIDS systems have been integrated into Network Intrusion Detection System (NIDS) and Host Intrusion Detection System (HIDS). These systems employ statistical, machine learning, and knowledge-based methods. Data sources can help distinguish between regular behaviour and invasive behaviour.

Needs of IDS

In the previous part, we talked about IDS and how they can be classified based on the data sources used as input to detect unusual behaviour. In terms of data sources, the two primary kindsz of IDS technology are host-based IDS (HIDS) and network-based IDS (NIDS). A few examples of the audit sources that HIDS examines include the operating system of the host system, window server logs, firewall logs, application system audits, and database logs. Without the need for network data, HIDS can recognise certain types of attacks, such as the use of a pendrive or direct access to the device. Network intrusion detection systems (NIDS) use packet capture, NetFlow, and other network data sources to analyse network activity that has been gathered from a network. IDS on the network can be used to monitor numerous devices that are linked to the network. Due to the amount of data moving through modern high-speed communication networks, NIDS has a restricted ability to examine all data in a high bandwidth network. Before the threats propagate to another computer system, NIDS is able to watch the external harmful actions that could be started from an external threat at an early stage. A strong, multi-tier defence against external and internal dangers can be provided by NIDS placed at different places within a particular network design in conjunction with HIDS and routers. highlights the differences between HIDS and NIDS. Creech et al. suggested a HIDS strategy based on irregular system call patterns in an attempt to lower false alert rates while raising detection rates. The basic idea is to use a HIDS that will use advance audit data provided by an application, and this advance data will then be used to generate alerts and inform the administrator about the intrusion as well as his or her activities after the intrusion and his or her identification through photo taking. A number of AIDS systems have been incorporated into Network Intrusion Detection System (NIDS) and Host Intrusion Detection System in order to enhance monitoring efficiency (HIDS). These systems use knowledge-based, statistical, and machine learning techniques. Info sources can aid in separating out normal behaviour from intrusive behaviour.

Detection methodologies:-

1. **Signature-based Detection (SD)**: Misuse Detection or Knowledge Based Detection are other names for SD. A pattern or string that is associated with a known threat or attack is called a signature. To identify potential intrusions, SD compares patterns to events that have been captured. utilising the information learned from certain attacks and system flaws.

2. **Anomaly-based Detection (AD):** An anomaly is a departure from a known behaviour, and profiles are the typical or anticipated behaviours discovered by long-term observation of routine activities, network connections, hosts, or users. A few examples of AD include Trojan horses, DoS attacks, attempted break-ins, spoofing, genuine users infiltrating systems, and more.

3. **Stateful Protocol Analysis (SPA):** The stateful in SPA signifies that an IDS may be able to track and know the protocol states (e.g.pairing requests with replies). Although the SPA process resembles an AD, they are fundamentally distinct. The network protocol models in SPA are often based on protocol standards from international standard bodies, including the IETF. Another name for SPA is specification-based detection. Hybrid-Most IDSs employ a variety of techniques to offer more thorough and precise detection. For instance, SD and AD are complimentary strategies since the former addresses specific risks and assaults and the latter concentrates on unidentified dangers.

Motivations for Quantitative Evaluations

A diverse range of potential clients are keen in the outcomes of quantitative evaluations of IDS accuracy. It was difficult for even the author of this paper to deal with the situation where someone tried to break into his laptop while he was out of the room. This could happen to anyone in any public place, or an intruder could be a malicious person who has access to your device. As a result, such data is necessary for regular users to save their data or improve their system security. Security analysts who examine the output of IDSs want to know how probable it is that alerts will be issued when particular sorts of assaults are initiated. When analysing audit data, we need make sure that It sould be very simple to reach the intruder or identify one by looking at IDS output, so, finally, R&D programme managers need to be aware of the benefits and drawbacks of the systems that are currently in use in order to properly focus on research, put effort into system improvement, and monitor their progress.

Signature-based (knowledge-based)	Anomaly-based (behavior-based)	Stateful protocol analysis (specification-based)
Pros		
● Simplest and effective method to detect known attacks. Detail contextual analysis.	● Effective to detect new and unforeseen vulnerabilities. Less dependent on Os. Facilitate detections of privilege abuse.	● Know and trace the protocol states. Distinguish unexpected sequences of commands.
Cons		

● Ineffective to detect unknown attacks, evasion attacks, and variants of known attacks. Little understanding to states and protocols. Hard to keep signatures/patterns up to date. Time consuming to maintain the knowledge	● Weak profiles accuracy due to observed events being constantly changed. Unavailable during rebuilding of behavior profiles. Difficult to trigger alerts in right time.	● Resource consuming to protocol state tracing and examination. Unable to inspect attacks looking like benign protocol behaviors. Might incompatible to dedicated OSs or APs.

Table 1. Pros and Cons of IDS.

Comparisons of IDS technology types.

Item	Technology			
	HIDS	NIDS	WIDS	NBA
Components[a]	Agent: software (inline passive)MS: $1\sim n$ DS: $1\sim n$ (option)	Sensor: n (inline/passive) MS: $1\sim n$ DS: $1\sim n$ (option)	Sensor: n (passive) MS: $1\sim n$ DS: $1\sim n$ (option)	Sensor: n (most MS: $1\sim n$ (option) DS: optional
Detection scope of sensor/agent	Single host	Network subnet: n Host: n	WLAN: n WLAN client: n	Network subnet: n Host: n
Architecture[b]	MN or SN	MN	MN or SN	MN or SN
Strengths	Only HIDS can analyze end-to- end encrypted communications'activity.	Capable to analyze the broadest scopes of AP protocols	WIDS is more accurate due to its narrow focus. Only WIDS can supervise wireless protocol activity.	Superior detection powers at reconnaissance scanning, reconstruct malware infections and DoS attacks
Technology limitations[c]	• More challenging in detection accuracy due to a lack of context knowledge Delays in • alert generation and centralized reporting • Consume host resources • Conflict with existing security controls	• Cannot monitor wireless protocols • High false positive and falsenegative rates • Cannot detect attacks within encrypted traffic No • full analysis support under high loads.	• Cannot monitor AL, TL andNL protocol activities. • Cannot avoid evasiontechniques. • Sensors are susceptible to physical jamming attacks. Cannot • compensate for insecure wireless protocols	• The major limitation is the delay in detection attacks, caused by transferring flow data to NBA in batches, but not in real time.
Security capabilities				
Information gathering APs,	Network traffic, system calls, file system activity.	Hosts, OSs, APs, network traffic.	WLAN, devices (e.g., clients).	Hosts, OS, services (IP, TCP, UDP, etc).
Logging	Reference (Stavroulakis andStamp, 2010)	Reference (Stavroulakis andStamp, 2010)	Reference (Stavroulakis andStamp, 2010)	Reference (Stavroulakis andStamp, 2010)
Detection methodology[d]	SD and AD (combined)	SD (major), AD and SPA	AD (major), SD and SPA	AD (major), SPA
Type of suspicious eventsdetected	AL, TL and NL network traffic, event logs (e.g., application activities, file system activities),	AL, TL, NL and HW reconnaissance and attacks, unexpected AP services, policy	Wireless protocol activity, insecure WLAN and devices, DoS attacks, network scanning,	AL, TL, NL anomalous traffic flows (DoS attacks, malware) unexpected AP service

Technology types

1. Host-based IDS (HIDS)
2. Network based IDS (NIDS)

We have now provided a summary of IDS detection methodologies, tactics, and systems. People should be careful while choosing the approaches because each has benefits and limitations. When it comes to inspecting known attacks, pattern-based IDS is both simple to use and very successful. However, the method struggles to identify novel attacks, assaults masked by defensive measures, and many variations of known attacks. Additionally, a number of rule-dependent techniques for identifying unanticipated assaults have previously been put forth.

However, using such methods could make it difficult to create and update the information needed for specific assaults.

Heuristic-based approaches also have the advantage of requiring no prior knowledge of assaults, but they struggle in real-time programming because to their high computing cost. Therefore, before practical applications, having a thorough understanding of IDSs and application needs is essential. We also suggest a much more thorough analysis of IDSs. The tables and statistics we mentioned make it easier to understand the big picture. We also provide a quick introduction to two well-known open source IDS learning tools. However, as it is extensively used in cloud systems, virtualization technology is becoming much more and more significant.. The VM is the first digital component that interacts directly with people, hence we also research a number of IDS problems on VMs.

Probability of Detection

This statistic establishes the frequency of attacks accurately picked up by an IDS in a specific environment during a specific period of time. Because the set of assaults utilised during the test heavily influences an IDS's ability to detect threats, evaluating the detection rate can be challenging. Additionally, an IDS can be set up or calibrated to favour either the capacity to detect attacks or the reduction of false positives because the probability of detection correlates with the false positive rate (see section 3.2 for an explanation of this). When testing for false positives and hit rates, one must be sure to use the same configuration. Additionally, subtle variants of attacks can avoid an NIDS. When an attack is launched in a clear, straightforward manner, an NIDS may be able to identify it, but not when even straightforward methods of stealthiness are used. Attacks can be made inconspicuous by fragmenting packets, utilising several forms of data encoding, odd TCP flags, encrypting attack packets, distributing them across several network sessions, and launching them from different sources.

Ability to Identify an Attack

By giving each attack a common name, a name for a vulnerability, or by classifying the attack, this statistic shows how well an IDS can identify the attack that it has detected.

Ability to Determine Attack Success

This test shows whether the IDS can identify successful remote site attacks that grant the attacker higher-level access to the system being targeted. Many remote privilege-gaining attacks (or probes) now used in network contexts are unsuccessful and do not harm the machine being targeted. However, many IDSs do not distinguish between unsuccessful and successful attacks. Some DSs and some IDS can only identify the signature of attack actions for the same attack, while others can detect the proof of damages (whether the attack was successful) (with no indication whether attack succeeded or not). The ability to determine attack success is essential for the analysis of attack correlation and attack scenario; it also greatly simplifies an analyst's work by differentiating between more significant successful attacks and the typically less damaging unsuccessful attacks. Information on both successful and unsuccessful attacks is needed to measure this capability.

Proposed Work

HIDS monitors a variety of host events and activities in order to detect malicious code and intrusion activities in host systems such as desktops, mail servers, DNS servers, webservers, database servers, and so on. When HIDS detects malicious code or activity, as well as unexpected behaviour such as buffer overflow and file-system access, it prevents it from being executed. HIDS identifies intrusions on the host system by collecting information such as file systems accessed, activity logs based on mousestrokes and snapshots, and so on. When an intrusion is detected, it alerts the host's administrator with the entire audit data. Algorithm for taking an intruder's photo. Algorithm for capturing mousestrokes. Data selection and binding Algorithm for Screen Capture. Our suggested framework includes the following components: CommandCam Image Grabber, Data Sending Algorithm, Remote Code Checking, and System System Administrator.

Modelling Host Based Intrusion detection system using Audit Data

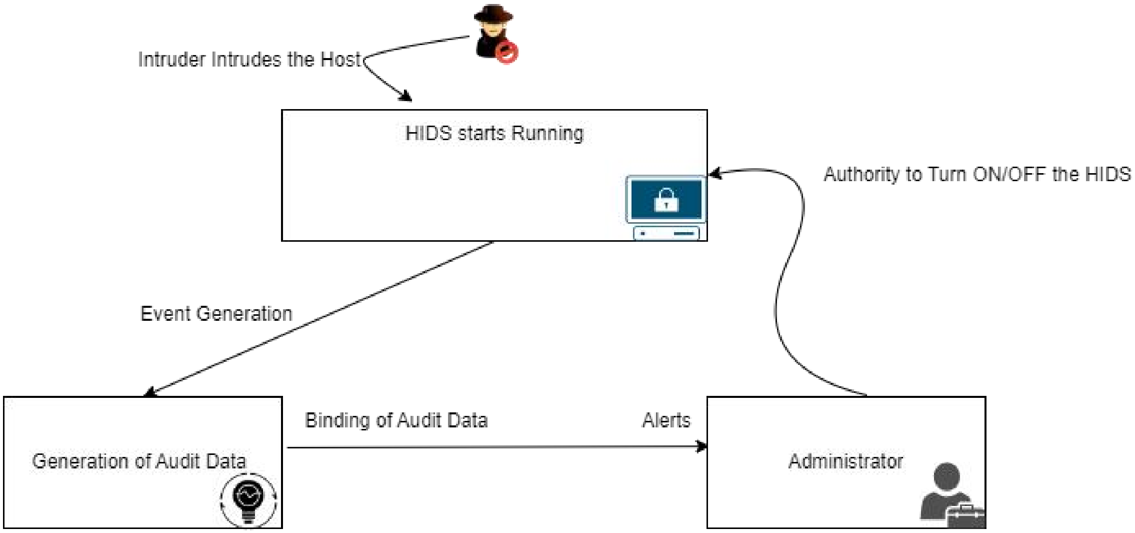
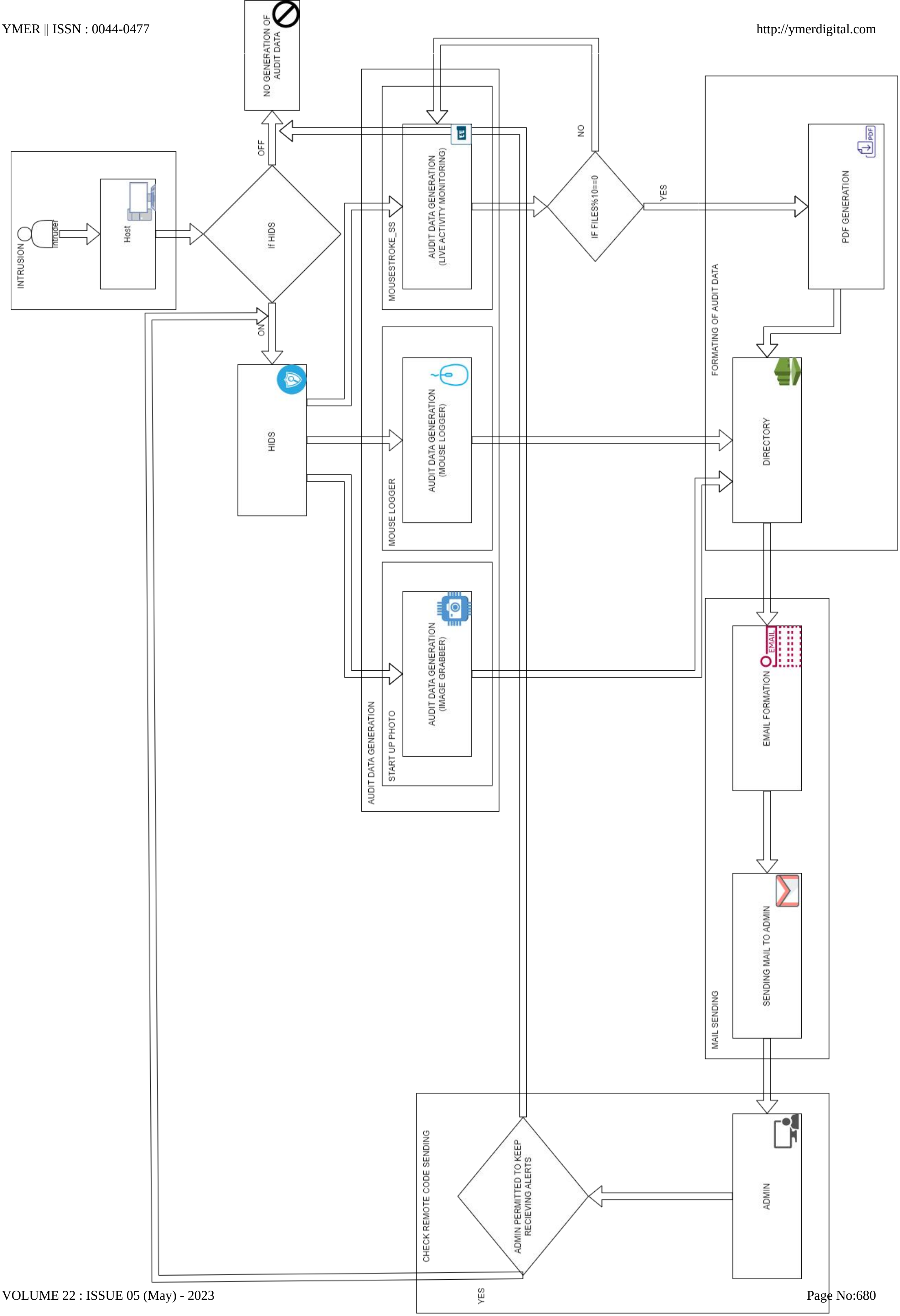


Fig 5. Our Model Of HIDS using Audit Data

Components of Our Framework :

1. Intruder snapshot capturing Algorithm.
2. Mousestroke capturing Algorithm.
3. Selection of data and Binding.
4. Screen Capturing Algorithm.
5. CommandCam Image Grabber.
6. Data Sending Algorithm.
7. Remote Code Checking .
8. System System Administrator – (S)He will takes the appropriate action based on the Display Warning , Report Generated and Tracing the Attacks/Intruders activities.

Architecture:-


INTRUSION
Intruder
Host
If HIDS
OFF
NO GENERATION OF AUDIT DATA
ON
HIDS
AUDIT DATA GENERATION
MOUSESTROKE_SS
AUDIT DATA GENERATION (LIVE ACTIVITY MONITORING)
MOUSE LOGGER
AUDIT DATA GENERATION (MOUSE LOGGER)
START UP PHOTO
AUDIT DATA GENERATION (IMAGE GRABBER)
IF FILES%10==0
NO
YES
FORMATING OF AUDIT DATA
PDF GENERATION
PDF
DIRECTORY
MAIL SENDING
EMAIL FORMATION
EMAIL
SENDING MAIL TO ADMIN
CHECK REMOTE CODE SENDING
ADMIN
ADMIN PERMITTED TO KEEP RECIEVING ALERTS
YES

ALGORITHM 1:MOUSESTROKE_PDF.PY

```
Input :Mouse strokes on the Mousepad.
Output: Audit files containing details of mousestroke(Updated in the Directory)
Procedure:
        Create a new python file and save it with a .py file extension. You will first want to
            import Listener from pynput.mouse or from pynput.mouse import Listener
                If Listener is imported
                    Setup the listener by creating an instance in a with statement and using
                        it's .join() method to join it to the main thread.with Listener() as
                listener:   listener.join()
                    If Setup Is completed
                        Create three methods;
                        on_move, on_click and on_scroll with the  parameters as shown:
                            def on_move(x, y):
                                pass
                            def on_click(x, y, button, pressed):
                                pass
                            def on_scroll(x, y, dx, dy):
                                pass
        Link these methods to the listener instance with the function names as the args; I have named
    the methods as they are defined in the listener class. Now when an action occurs, one of these methods
    will be run.
                If Linking done
            with Listener(on_move=on_move, on_click=on_click, on_scroll=on_scroll) as listener:
        Create an email and attach these audit data and send it to the admins mail.
    END PROCEDURE
```

The aforementioned algorithm is crucial for producing audit data that can be used for intrusion detection; to do this, we import and create a few methods such as on scroll, on move, and on click, and then pass a few arguments to them. After linking these functions together with linker, we then use Listener with the three functions' arguments and functions.The pynput module will be used to monitor mouse events and then atleast all these mouse logs will be binded with other audit data & then sent ot admin , Use the cmd command pip install pynput to install this module. The output will indicate when the module has been installed correctly, so check it to make sure there haven't been any mistakes.

A new Python file should be created and saved with the.py file extension. The first thing you should do is import Listener from pynput.mouse. Make the following three methods with the specified parameters: on move, on click, and on scroll. I have named the methods according to how they are defined in the listener class; link these methods to the listener instance with the function names as the args. Now, one of these procedures will be executed when an action takes place. if you desire the background execution of this script.Using the.pyw file extension, select File > Save As. There won't be a console window or a running appearance if it is now run outside of IDLE. But we must first delete the print statements in order to prevent the console from appearing.

Set up the fundamental settings as I've shown below and import logs. Change all print statements to logging after that info.

ALGORITHM 2: StartUpPhoto.py

```
Input : Intrusion by the Intruder
Output: Captured Image sent to admin(Updated in the Directory)
Procedure:
        Import all the packages required for grabbing date,setting up server for sending mails
        and for image grabber.
        If packages imported
              Start the image grabber using startfile command
              If image grabber started
                    Setup emails to send and receive alerts
                    If email setup is done
                         Define mail headings,body.
                        Use imported packages for setting up mail server.
                                   Attach the captured image as mail attachment
          using MIMEText  and MIMEBase start the server for sending mails
      If no error found
             Send mail to target admin email
            Close the server
      Close the Continous Attachment of files to mail.
      Remove the image from directory.
End Procedure
```

CommandCam is a straightforward and user-friendly command line webcam image grabber for Windows, and it will be used in the aforementioned method. A single webcam picture is recorded and saved as a bitmap file. For usage in batch files and other circumstances where you need a very straightforward method to automate picture capture, CommandCam is perfect. Therefore, in our algorithm, we are using this command cam to capture in person intruders' images using the front camera of your laptop and we are doing this when the intruder will intrude into the system, using input of algorithm first. CommandCam uses Microsoft's DirectShow API to access webcams, so it should work with most USB cameras.After the images are taken, the data generated by the algorithm first I.e mouse logs will be sent along with all of the images to the host's administrator via email, allowing them to determine who the intruder is.

ALGORITHM 3: CheckRemoteCodeSending.py

```
Input : Unique Code for Controlling HIDS
Output: Control over On and Off of HIDS
Procedure:
        Create a python file then import the packages for like
        Set Email and Access Token of the respected gmail account of admin.
        Create three methods search,get_emails,get_body and pass the respected parameters.
         def get_body(msg):
         def search(key, value, con):
         def get_emails(result_bytes):
         Keep checking for email from the desired accouont using get_emails,get_body and
search
        if code==predefinedValueforstoppingIds
             Stop the process
        Else
              Pass
```

END PROCEDURE

In our third algorithm, we have set up access to our mails through our application so that when the admin receives alerts for intrusion activity, S(he) should be able to at least control her/his device, so that further intrusion can be stopped.For this, we will be using the Google API.To use the Gmail API, we need a token to connect to Gmail's API.We first enable the Google mail API, and now all w e re good to go. gather all email and look for specific mail in the inbox and as soon as we will be finding our specific mail from a specific defined gmail account ,we will look and authenticate our code found in the email with the one we are looking for and if found ,we will stop the ids .In future on the basis of this mail we can also control our host ,we can turn off and on our host and also we can lock it temporarily.

Competetive Analysis

S.N	Technology type	Performance	Type of source	Comments
[1]	Network based	Average	User profiles,usage of disk and memory.	Simple but less accuracy
[2]	Network based	Average	Network packets.	Real-time and active .
[3]	Host&network based	High	Network traffic. User profiles system's events or incidents, log events.	Optimal statistical (probabilistic) model
[4]	Network based	High	Audit records, rule patterns from user profiles and policy	Self-study, control is poor
[5]	Network based	Average	Knowledge base for association rule discovery.	Automatically generated models
[6]	Host&network based	Average	Network packets	Varied modeling / profiling
[7]	Network based	High	Profiles limited sample data, binary data.	Lower false positive rate,
[8]	Network based	High	State-transition diagram of known attacks.	High accuracy
[9]	Host based	High	Audit records, user profiles.	High-level task pattern
[10]	Network based	Average	Audit data, sequence of system calls or commands.	Probabilistic Self-training
[11]	Network based	Poor	Log files.	Low false positive rate, Less effective
[12]	Network based	Average	Sequence of commands predict events audit records, network traffic (tcp/udp/icmp)	Self-learning Fault tolerant

Proposed Work	***Host Based***	***High***	***Advance audit data(audit trails) ,binded multimedia files ,mouse logs.***	***Very Simple to execute ,less complex , detection of the intruder can also be done .***

Finding/Result Analysis:-

Numerous classification metrics exist for Intrusion Detection Systems (IDS), which may also be referred to by multiple names. Table 4 elucidates the confusion matrix of a two-class classifier, which serves as a valuable tool in evaluating the performance of an IDS. It is important to note that each column of the matrix denotes the instances within a predicted class, while each row represents the instances in an actual class. By utilizing such a classification matrix, one can effectively assess the precision, recall, accuracy, and other performance measures of an IDS.

Actual Class	Predicted Class		
	Class	Normal	Attack
	Normal	True negative (TN)	False Positive (FP)
	Attack	False Negative (FN)	True positive (TP)

Intrusion Detection Systems (IDS) are evaluated based on several performance measures to determine their effectiveness in detecting and preventing intrusions. Some of the standard performance measures used for evaluating IDS are:

- **True Positive Rate (TPR):** True Positive Rate (TPR) is an important performance metric for Intrusion Detection Systems (IDS) and is calculated as the ratio between the number of correctly predicted attacks and the total number of attacks. A TPR of 1 indicates that all intrusions are detected, which is extremely rare for an IDS. TPR is also known as the Detection Rate (DR) or the Sensitivity. Mathematically, TPR can be expressed as:

$$TPR = \frac{TP}{TP + FN}$$

- **False Positive Rate (FPR):**FPR measures the rate at which the IDS raises an alarm for benign activities. It is calculated by dividing the number of normal instances that are incorrectly classified as attacks by the total number of normal instances. It can be expressed mathematically as:

$$FPR = \frac{FP}{FP + TN}$$

- **False Negative Rate (FNR):** False negative means when a detector fails to identify an anomaly and classifies it as normal. The FNR can be expressed mathematically as:

$$FNR = \frac{FN}{FN + TP}$$

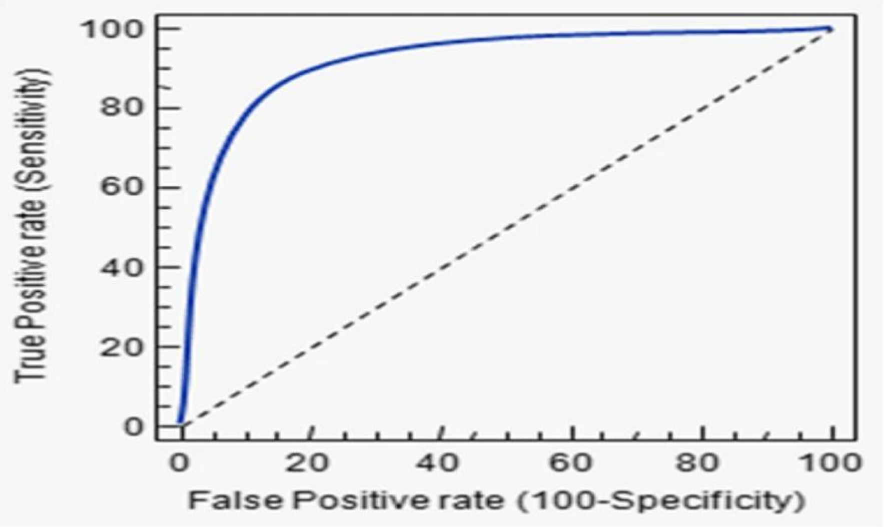
- **Classification rate (CR) or Accuracy:** TCR stands for Detection Rate or True Positive Rate (TPR). It is the ratio of the number of correctly predicted positive instances (i.e., instances of anomalous traffic behavior correctly detected by the IDS) to the total number of positive instances (i.e., all instances of anomalous traffic behavior in the dataset).

$$ACCURACY = \frac{TP + TN}{TP + TN + FP + FN}$$

The ROC curve is a graphical representation of the performance of a binary classifier system as the discrimination threshold is varied. It plots the True Positive Rate (TPR) on the y-axis and the False Positive Rate (FPR) on the x-axis.

Each point on the ROC curve represents a specific trade-off between the TPR (also known as sensitivity) and the FPR (1-specificity) at a particular classification threshold. The closer the curve is to the upper left corner of the graph, the better the performance of the classifier, with 100% sensitivity (i.e., all true positives are correctly identified) and 100% specificity (i.e., no false positives are identified).

The ROC curve is a useful tool to compare the performance of different classifiers or to optimize the classification threshold of a single classifier. The area under the curve (AUC) can be used as a single metric to summarize the overall performance of the classifier, with values ranging from 0.5 (random performance) to 1.0 (perfect performance).



The suggested architecture will safely protect the host computer systems and their data from invasions and attacks by unauthorised individuals. The audittrials data and logs generated by the programmes and operating system are used to detect intrusions and alert the administrator so that appropriate action can be done. HIDS using audit trial data can use both the technology of Intrusion Detection System and intrusion prevention system,but in this paper we have attempted to execute intrusion prevention system only,HIDS here will make use of audit data generated by the framework we are creating and using this audit data it will detect intrusion has occurred and will notify the admin, who can then take further action for Prevention of intrusive activities, malicious Behaviour. Intruder snapshot capturing Algorithm employs multiple Python programmes to capture an intruder's image while utilising a pre-built c++ image grabber. The mousestroke capture algorithm will produce a number of mousestroke trails. Screen Capturing Algorithm will take screenshots of screen activity. Data Sending Algorithm, Data Selection, and Binding will be done to make sure that the admin of the PC is notified of intrusions with the right audit data. Remote Code Checking will be done by the System System Administrator to gain control over its host.

CONCLUSION:

The first thing that we did was conduct a survey of the most recent and cutting-edge technology trends on HIDS. After that, we identified the flaws in the already existing intrusion detection system and did research on the most effective techniques and algorithms for detecting intrusions. Finally, we built a model that we anticipate will offer high promising levels of security, performance, and accuracy. The field of HIDS is very active; recent research areas offer a hundred percent security on computer systems and Information Systems that can detect and prevent all types of intrusions and malicious activities in real time, without creating any false alarms and without any human intervention; however, there has been a need of HIDS using audit data. This HIDS chooses the audit data as the data source for its creation of alerts to the admin, and once alerts have been generated for administrating the system, they are sent to the administrator.

REFERENCES

[1] A. Khraisat and A. Alazab, “A critical review of intrusion detection systems in the Internet of Things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges”, Cyber Security 4(18), 2021, DOI: 10.1186/s42400-021-00077-7

[2] A. Colakovi and M. Hadziali, “Internet of Things (IoT): A Review of Enabling Technologies, Challenges, and Open Research Issues”, Computer Networks, 2018, DOI: 10.1016/j.comnet.2018.07.017.

[3] E. C. Ugwuabonyi and E.Z. Orji, “Issues and Challenges in Security and Privacy of Internet of Things (IoT)”, International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), 7(12), 2018, ISSN 2278-2540.

[4] B. B. Zarpaelo, R.S. Miani, C.T. Kawakani and S. C. Alverenga, “A Survey of Intrusion Detection in Internet of Things”, Journal of Network and Computer Applications, 2017, DOI: 10.1016/j.jnca.2017.02.009.

[5] A. Mayzaud, R. Badonnel and I. Chrisment, “A Taxonomy of Attacks in RPL-based Internet of Things”, International Journal of Network Security”, ACEEE a Division of Engineers Network, 18 (3), pp.459-473, 2016, DOI:10.6633/IJNS.201605.18(3), hal-01207859.

[6] T. A. Tchakoucht and M. Ezziyyani, “Building A Fast Intrusion Detection System For High-Speed Networks: Probe and DoS Attacks Detection”, Procedia Computer Science, 127, pp. 521–530, 2018.

[7] K.K. Patel and S.M. Patel, “Internet of Things-IoT: Definition, Characteristics, Architecture, Enabling Technologies, Application and Future Challenges”, International Journal of Engineering Science and Computing, 6(5), ISSN 2321- 3361, 2016, DOI: 10.4010/2016.1482.

[8] V. Adat and B. B. Gupta, “Security in Internet of Things: issues, challenges, taxonomy, and architecture”, Telecommunication System, 2017, DOI: 10.1007/s11235-017-0345-9.

[9] A. Tewari and B.B. Gupta, “Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework”, Future Generation Computer Systems, 108, ISSN: 0167-739X, pp: 909-920, 2020, DOI: 10.1016/j.future.2018.04.027

[10] R. Sahay, G. Geethakumari and K. Modugu, “Attack Graph based Vulnerability Assessment of Rank property in RPL-6LowPAN in IoT”,
IEEE Explore, 2018, DOI: 10.1109/WF-IoT.2018.8355171

[11] J. Deogirikar and A. Vidhate, “Security Attacks in IoT: A Survey. International Conference on IoT in Social, Mobile, Analytical and Cloud”, I-SMAC- 2017, IEEE, 2017.

[12] A. R. Sfar, E. Natalizio, Y. Challal and Z. Chtourou, “A Roadmap for Security Challenges in the Internet of Things”, Digital Communications and Networks, 4, pp.118-137, 2018.

[13] E E. Hemdan and D.H. Manjaiah, “Cybercrimes Investigation and Intrusion Detection in Internet of Things based on Data Science Methods”, Cognitive Computing for Big Data Systems over IoT, 2018, DOI: 10.1007/978-3-319-70688-7_2.

[14] Y. Fu, C. Yan, J. Cao, O. Kore and X. Cao, “An Automata based Intrusion Detection method for Internet of Things”, Mobile Information Systems, Hindawi Publications, 2017(1750637), 2017, DOI: 10.1155/2017/1750637.

[15] S. Raza, L. Wallgren and T. Voigt, “SVELTE: real-time intrusion detection in the Internet of Things”, Ad Hoc Network, 11(8), ISSN: 2660, 2018.

[16] H. Qu, L. Lei, X. Tang and W. Ping, “A Lightweight Intrusion Detection Method Based on Fuzzy Clustering Algorithm for Wireless Sensor Networks”, Advances in Fuzzy Systems, Article ID: 4071851, 2018, DOI: 10.1155/2018/407185.

[17] S. K. Biswas, “Intrusion Detection Using Machine Learning: A Comparison Study”, International Journal of pure and Applied Mathematics, 118 (19), pp.101-114, ISSN: 1311-8080 (print); ISSN: 1314-3395 (online), 2018.

[18] N. Moustafa, B. Turnbull and K. R. Choo, “An Ensemble Intrusion Detection Technique based on

proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things", IEEE Internet of Things Journal, 2018, DOI:10.1109/JIOT.2018.2871719

[19] S. U. Jan, S. Ahmed, V. Shakov and I. Koo, "Toward a Lightweight Intrusion Detection System for the Internet of Things", IEEE Access, 2019, DOI: 10.1109/ACCESS.2019.2907965.

[20] M. Eskandari, Z. H. Janjua, M. Vecchio and F. Antonell, "Passban IDS: An Intelligent Anomaly Based Intrusion Detection System for IoT Edge Devices", IEEE Internet of Things Journal, pp. (99):1-1, 2020, DOI: 10.1109/JIOT.2020.2970501.

[21] O. Alkadi, N. Moustafa, B. Turnbull and K. R. Choo, "A Deep Blockchain Framework-enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks", IEEE Internet of Things Journal, 2020, DOI:10.1109/JIOT.2020.2996590.

[22] M. A. Cheema, H. K. Qureshi, C. Chrysostomou and M. Lestas, "Utilizing Blockchain for Distributed Machine Learning based Intrusion Detection in Internet of Things", 16th International Conference on Distributed Computing in Sensor Systems (DCOSS- 2020), IEEE Xplore, 2020, DOI: 10.1109/DCOSS49796.2020.00074.

[23] G. D. L. T. Parra, P. Rad, K. R. Choo and N. Beebe, "Detecting Internet of Things Attacks using Distributed Deep Learning", Journal of Network and Computer Applications, 163(102662), ScienceDirect, 2020, DOI: 10.1016/j.jnca.2020.102662.

[24] M. A. Alsoufi, S. Razak, M. M. Siraj, I. Nafea, F. A. Ghaleb, F. Saeed and M. Nasser, "Anomaly-based Intrusion Detection Systems in IoT using Deep Learning", Applied Sciences, 11(18), 8383, 2021,DOI:10.3390/app11188383.

[25] P. Kumar, G. P Gupta and R. Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the Internet of Things networks", Journal of Ambient Intelligence and Humanized Computing, 12, pp. 9555–9572, 2020, DOI:10.1007/s12652-020-02696-3

[26] L. Santos, R. Gonçalves, C. Rabadao and J. Martins, "A flow-based intrusion detection framework for internet of things networks", Cluster Computing, Springer, 2021, DOI: 10.1007/s10586-021-03238-y

[27] E. Benkhelifa, T. Welsh and W. Hamouda, "A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Towards Universal and Resilient Systems", IEEE, 2018, DOI:10.1109/COMST.2018.2844742.

[28] D. Oh, D. Kim and W. W. Ro, "A Malicious Pattern Detection Engine for Embedded Security Systems in the Internet of Things", Sensors, 14 (12), ISSN: 24188–24211, 2014, DOI: 10.3390/s141224188.

[29] T. H. Lee, T. H. Wen, L. H. Chang, H. S. Chiang and M.C. Hsieh, "A lightweight Intrusion Detection Scheme based on Energy Consumption Analysis in 6LowPAN", Advanced Technologies, Embedded and Multimedia for Human-centric Computing, Lecture Notes in Electrical Engineering 260, Springer Netherlands, pp. 1205–1213, 2014.

[30] A. Mehmood, M. Mukherjee, S. H. Ahmed, H. Song and M. M. Malik, "NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks", Journal of Supercomputers, Springer Science+Business Media, LLC, Springer Nature, 2018, DOI:10.1007/s11227-018-2413-7

[31] C. Cervantes, D. Poplade, M. Nogueira and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things", IFIP/IEEE International Symposium on Integrated Network Management (IM), pp.606–611, 2015.

[32] A. Sforzin and M. Conti, "RpiDS: Raspberry Pi IDS-A fruitful Intrusion Detection System for IoT", International IEEE Conference on Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart World Congress, 2016, DOI:10.1109/UIC-ATC-Scalcom-CBDCom-IOP-SmartWorld.2016.114.

[33] D. Midi, A. Rullo, A. Mudgerikar and E. Bertino, "KALIS: A system for knowledge-driven adaptable intrusion detection for the Internet of Things", Proceedings of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS'17), 2017.

[34] A. Wani and S. Revathi, “Analyzing Threats of IoT Networks Using SDN Based Intrusion Detection System (SDIoT-IDS)”, Smart and Innovative Trends in Next Generation Computing Technologies (NGCT-2017), Springer, CCIS 828, pp. 536–542, 2018.

[35] J. Amaral, L. Oliveira, J. Rodrigues, G. Han and L. Shu, “Policy and Network-based Intrusion Detection System for IPv6-enabled Wireless Sensor Networks”, IEEE International Conference on Communications (ICC-2014), pp. 1796–1801, 2014.

[36] N. K. Thanigaivelan, E. Nigussie, S. Virtanen and J. Isoaho, “Hybrid Internal Anomaly Detection System for IoT: Reactive Nodes with Cross-Layer Operation”, Security and Communication Networks, Article ID: 3672698, 2018, DOI: 10.1155/2018/3672698.