

Dissertation Title
Intrusion Detection System

Name of the Group members

Bhaskar Sharma
Sneha Irukuvajjula
Vedansh Vachani

Under the guidance of

Dr. Parag Kaveri

Submitted in partial fulfilment of undergraduate Degree
Bachelor of Computer Application
To



SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH
CONSTITUTENT OF SYMBIOSIS INTERNATIONAL DEEMED UNIVERSITY, PUNE
September 2018

List of figures

<u>Serial Number</u>	<u>Title</u>
1.	Acknowledgement
2.	Abstract of the project
3.	Introduction
4.	Literature Review
5.	Feasibility Study
6.	Use Case Diagram
7.	Software Architectural Design
8.	User Interface Design
9.	Test Plan and Test Cases
10.	Drawbacks and Future Enhancements

11.	Annexures and Snapshots
12.	Summary
13.	Bibliography

Acknowledgements

We would like to thank Symbiosis Institute Of Computer Studies And Research for providing us an opportunity to do this wonderful project.

We owe our deepest gratitude to our project guide, Mr. Parag Kaveri who took keen interest in our project work and guided us all way long, till the completion of the project work by providing us with all the necessary information for developing a good system.

Last but not the least, we would also like thank a few of our friends for their encouragement and without whom the project wouldn't be as it is now.

Abstract:

With rapidly increasing technology, there is a need to secure all the systems to secure the confidential data. Intrusion Detection System is a new safeguard technology for system security after the traditional technologies. The purpose of this study is to define what IDS is the types of IDS, attacks, different tools and techniques implemented, challenges so as to safeguard the client's system and make it as secure as possible.

What is IDS?

Intrusion Detection System is a system that detects any kind of intrusion or malicious activity in the system or server and alerts the user when there has been an unauthorized attempt or access.

What is new in our system?

- Port Scan Attack Detector (PSAD).
- Log Management.

Introduction:

Intrusion Detection Systems help information systems prepare for, and deal with attacks. They accomplish this by collecting information from a variety of systems and network sources, and then analyzing the information for possible security problems. An intrusion detection system (IDS) examines system or network activity to find possible intrusions or attacks. Intrusion detection systems are either network-based or host-based. Network-based intrusion detection systems are most common, and examine passing network traffic for signs of intrusion. Host-based systems look at user and process activity on the local machine for signs of intrusion. Since each type has specific strengths and weaknesses.

There are two main components to the Intrusion Detection Systems.

-Network Intrusion Detection System (NIDS):

It performs analysis of the traffic that is passed from the network to a specific host.

Example of the NIDS would be, installing it on VPN device, to examine the traffic once it was decrypted. This way you can see if someone is trying to break into your VPN device.

- Host based Intrusion Detection System (HIDS):

Takes a snap of your existing system files and matches it to the previous snapshot. If the critical system files were modified or deleted, the alert is sent to the administrator to investigate.

Example of HIDS can be seen on the mission critical machines that are not expected to change their configuration.

Need:

Information Systems and Networks are subject to electronic attacks. Attempts to breach information security are rising every day, along with the availability of the Vulnerability Assessment tools that widely available on the Internet, for free, as well as for a commercial use. Tools such as Subseven, BackOrifice, Nmap, LOftCrack, can all be used to scan, identify, probe, and penetrate your systems. Firewalls are put in place to prevent unauthorized access to the Enterprise Networks.

Let's, however, ask ourselves: Are firewall enough to prevent any intrusion?

Let's understand this with the help of an example:

Imagine that you have just purchased a Home Theatre System. Everyone who knows anything about electronics, have an idea about how much it costs. After installing it, you decided that you might need to install new locks on all the doors of your house, because the old ones do not use the up-to-date secure mechanisms. You call the locksmith, and in about 2 months (if you are lucky) you have your new locks on your doors, and you are the only one who have the keys (well, maybe your mother has another pair). You plan to go on a holiday trip.

As you come back a week later, you find that the Entertainment room looks different. After careful examination, you realized that your Home Theatre System, that you were dwelling over for the last year, is missing. You find there are shoe stains on the carpet and the window is broken.

With all the evidences, you believe that someone broke into your house, stole, and vandalized a lot of your prized possessions. You suddenly begin to vaguely remember the brochure that you got, about a burglar alarm installation in your neighbourhood that you threw it away just a week before.

The installation and monitoring would have cost you 19.95k a month with this promotional offer. Neglecting to install the system, is a secret that you would have to leave with for the rest of your life could you have prevented it from happening, were you to install an alarm? May be not completely, but the damage would have been much less.

The real life example above is the exact same analogy of what might happen to your network. What's worth is that the thief may be on your network for a long time, and you might not even know it. Firewalls are doing a good job guarding your front doors but they do not have a possibility to alert you in case there is a backdoor or a hole in the infrastructure.

Script kiddies are constantly scanning the Internet for known bugs in the system, including constant scans by subnets. More experienced crackers may be hired by your competitors, to target your network specifically, in order to gain competitive advantage. The list of threats can go on.

Existing Software

In this section we will discuss some of the current generation intrusion detection system.

Network Intrusion Detection System:

The network IDS usually has two logical components: the sensor and the management station. The sensor is placed on a network segment, monitoring it for suspicious traffic. The management station receives alarms from the sensor(s) and displays them to an operator.

The sensors are usually dedicated systems that exist only to monitor the network. They have a network interface in promiscuous mode, which means they receive all network traffic not just that destined for their IP address and they capture passing network traffic for analysis. If they detect something that looks unusual, they pass it back to analysis station.

Host Intrusion Detection System:

The host-based IDS look for signs of intrusion on the local host system. These frequently use the host system's audit and logging mechanism as a source of

information for analysis.

They look for unusual activity that is confined to the local host such as login, improper file access, unapproved privilege escalation, or alterations on system privileges. This IDS architecture generally uses rule-based engines for analyzing activity; an example of such a rule might be, “super user privilege can only be attained through the su command.” Therefore successive login attempts to the root account might be considered an attack.

1) **SNORT** :

Snort's an open source network-based intrusion detection system (NIDS) has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching, and content matching. These basic services have many purposes including application-aware triggered quality of service, to de-prioritize bulk traffic when latency-sensitive applications are in use.

The program can also be used to detect probes or attacks, including, but not limited to operating system fingerprinting attempts, common gateway interface, buffer overflows, server message block probes and stealth port scans. Snort can be configured in three main modes: sniffer, packet logger and network intrusion detection. In sniffer mode, the program reads the network packets and displays them on the console. In packet logger mode, the program logs packets to the disk. In intrusion detection mode, the program monitors the network traffic and analyzes it against a rule set defined by the user program and then performs a specific action based on what has been identified.

Snort Performance

We obtained a result of Snort (version 2.6.1.5) on Intel 2.0 Ghz processor on Debian Linux 2.6.18 using a GNU gprof (III profiler (version 2.16). Snort was configured with five pre-processing components (Stream, frag HTTP Inspect, Telnet decode and sfport scan) and 6565 rules.

The network traces used to obtain the profiling result was obtained from MITLincotii Lab, 1998 DARPA Intrusion Detection Evaluation project file (tcpdump file format) Table 1 shows the percentage of the total execution time used by each component (Results depend on Snort configuration and test data).

2) Tiger

Tiger is a security tool that can be used as both, a security audit and an intrusion detection system. It supports multiple UNIX platforms, is free and is provided under a GPL license. Unlike other tools, Tiger needs only of POSIX tools and is written entirely in shell language.

Tiger has some interesting features that merit its resurrection, inducting a modular design that is easy to expand. And its double edge, it can be used as an audit tool and a host intrusion detection system tool. Free Software intrusion detection is currently going many ways, from network IDS (with Snort), to the kernel (LIDS, or SNARE for Linux and Systrace for OpenBSD, for example) not mentioning file integrity checkers (many of these: aide, integritsamhain, tripwire) and logcheckers (even more of these, check the Log Analysis pages).

But few of them focus on the host-side of intrusion detection fully. Tiger complements these tools and also provides a framework in which all of them can work together. Tiger is neither a logchecker nor focuses on integrity analysis. It checks the system configuration and status.

Literature Review

Introduction:

With the rapid progress in the internet based technology new application areas for computer network have emerged. In instances, the fields like business, financial, industry, security and healthcare sectors the LAN and WAN applications have progressed. All of these application areas made the network an attractive target for the abuse and a big vulnerability for the community [6].

An Intrusion Detection System (IDS) inspects the activities in a system for suspicious behaviour or patterns that may indicate system attack or misuse. Malicious users or hackers use the organization's internal systems to collect information's and cause vulnerabilities like Software bugs, Lapse in administration, leaving systems to default configuration. IDS are used in network related activities, medical applications, credit card frauds, Insurance agency [3]. In addition to the hacking, new entities like worms, Trojans and viruses introduced more panic into the net-worked society. As the current situation is a relatively new phenomenon, network defences are weak [4].

The purpose of IDS is to detect both external attacks on, and internal misuse of computer and network resources, or information residing in these resources. Intrusions are most often thought of as originating from outside a trusted network. Unauthorized access from without may be achieved by traversing a leaky firewall, exploiting a security flaw, tunnelling in through "benign" protocols, or entirely subverting security measures through an unprotected link to an external system [5]. The limitation of IDS is they cannot resolve network attacks; it passes in network for only watches network traffic like packet sniffing [7].

There are two main categories of intrusion detection techniques; Anomaly detection [1] and Misuse detection. The former analyses the information gathered and compares it to a defined baseline of what is seen as "normal" service behaviour, so it has the ability to learn how to detect network attacks that are currently unknown. Misuse Detection is based on signatures for known attacks, so it is only as good as the database of attack signatures that it uses for comparison. Misuse detection has low false positive rate, but cannot detect novel attacks. However, anomaly detection can detect unknown attacks, but has high false positive rate 2.

History:

Securing data has been a prominent issue ever since the inception of computers and their enormous applications. The studies of Intrusion detection has been active field of research for about more than three decades now. It started with the publication of John Anderson's Computer Security threat monitoring and surveillance in 1980, which is one of the earliest research papers on this field. Dorothy Denning's seminal paper, "An Intrusion Detection Model" published in 1987 provided a methodological framework that inspired a number of researchers.

After that, for the past two decades, despite of substantial research and huge commercial investments, Intrusion Detection technology is immature and ineffective. In the early days of computers, hackers rarely used automated tools to break into systems. They were intelligent with high level of expertise and followed their own methodology to perform such actions.

The recent scenario is quite different now. A wide number of intrusion tools and applications are available now that can be used to exploit scripts that capitalize on widely known vulnerabilities. Before the development of modern IDS, intrusion detection consisted of a manual search for anomalies. Due to the availability of adequate processing speed it now became possible not only to look for attack patterns after the event had occurred, but also to monitor in "real-time" and trigger alerts if intrusions were detected .[12]

IDS TECHNIQUES

ANOMALY BASED INTRUSION DETECTION

Anomaly is indicated as an outlier, peculiarities or exceptions are the data pattern which performs abnormally. Anomaly detection technique is designed to uncover the patterns that are far from the normal and others are flagged as an intrusion. Anomaly detection is useful for finding attacks like misuse of protocol and service ports, DoS based on crafted payloads, DoS based on volume (DDoS), buffer overflow and other application payload anomaly. An anomaly-based IDS, attempts to detect behaviour that is inconsistent with “normal” behaviour. Thus, these systems are sometimes called behaviour-based. An anomaly-based IDS detects hundreds of login attempts within a few seconds, it might generate an alert of suspicious activity.

SIGNATURE BASED INTRUSION DETECTION

A signature-based IDS attempts to detect patterns in network traffic that it is already aware of. The terms “knowledge-based” and “misuse-based” are synonyms for “signature-based.” This is a similar concept to anti-virus software on a PC that scans files and memory for known patterns of a computer virus. It will only detect previously known attacks.

TARGET MONITORING

Target monitoring is a technique which is used to report if any changes or modifications happen in the system. This is usually done through cryptographic algorithm which computes a crypto checksum for each targeted file. If any changes happens in crypto checksum they are reported by IDS. Tripwire checksum is an integrity checker which checks for the changes or modification in the files .

Network Intrusion Detection System:

Network based IDS systems collect information from the network itself

rather than from each separate host . The NIDS audits the network attacks while packets moving across the network. The network sensors come equipped with attack signatures that are rules on what will constitute an attack and most network-based systems allow advanced users to define their own signatures . Attack on the sensor is based on signature and they are from the previous attacks and the operation of the monitors will be transparent to the users and this is also significant.

Advantages of Network based Intrusion Detection Systems:

- Lower Cost of Ownership
- Easier to deploy
- Detect network based attacks
- Retaining evidence
- Real Time detection and quick response.
- Detection of failed attacks.

Host Based Intrusion Detection System:

Host based IDS views the sign of intrusion in the local system. For analysis they use host system's logging and other information. Host based handler is referred as sensor. Other sources, from which a host-based sensor can obtain data, include system logs and other logs generated by operating system processes and contents of objects not reflected in standard operating system audit and logging mechanisms. Host based system trust strongly on audit trail. The information allows the intrusion detection system to spot subtle patterns of misuse that would not be visible at a higher level of abstraction. The elementary principle in IDS including Network Based Intrusion Detection System (NIDS) originated from anomaly HIDS research based on Denning's pioneering work. A host-based IDS provides much more relevant information than Network-based IDS. HIDS are used efficiently for analyzing the network attacks, for example, it can sometimes tell exactly what the attacker did, which commands he used, what files he opened, rather than just a vague accusation and there is an attempt to execute a dangerous command. It is less risky to configure.

Feasibility Study:

1) Feasibility:

A) Technical Feasibility: We can say that our system would be technically because we are not getting any difficult related to resource of development and maintenance of the project. We are not planning to re-invent the wheel and are using already developed software tools which are commonly available and easy to get on the internet.

B) Economically Feasibility: The project that we will be building would be economical as it wouldn't be taking any extra tools other than our required tools for development and is available for free. We need not spend any money for the development of the system.

C) Schedule Feasibility: It is defined as the state of being probable and completed within the scheduled time. Our project has been completed within the specified time.

D) Operational Feasibility: It is related to the measurement of the performance of the system for which purpose it was developed. We can say that our project is operationally feasible as we are using Command-Line Interface and that consumes less of the system resources, CPU time and memory yielding to high performance with respect to speed and efficiency.

2) Modules:

We divided our whole project into three modules :-

A) Log Management:

This contains all the log history of the user's activities. We divided this into 3 sub-modules.

i) Checking Last Permission, Size and Files Modification:

This will include:

- Check what all permissions have been changed.
- Unexplained changes in the size of the files.
- What all modifications have been made.
- Check for all hidden files.

ii) **Hidden Files:**

This includes all the hidden files in the directory specified.

iii) **System Boot Logs:**

This will include:

- The last time user logged in
- Last boot time
- Last reboot time
- Last shutdown time
- Creating a full reboot and shutdown report.

B) Port Flood Detection:

Hackers could send a large number of packets to random ports on a host machine. Hence, our software would sniff for such large number of packets, alert the user and take a corrective action against it.

C) Building the UI of our software:

We shall mostly be concentrating on Command Line Interface.

3) Requirements:

A) Software Requirements:

-System Requirements:

Linux Operating System (Preferably any penetration testing OS like Backtrack, Kali, Parrot, Cyborg Hawk).

- Tools:

- Tshark (Command Line Packet Sniffing Tool)
- Figlet
- OS Module (Python)
- Crontab
- FTP server
- Python

-Programming Languages:

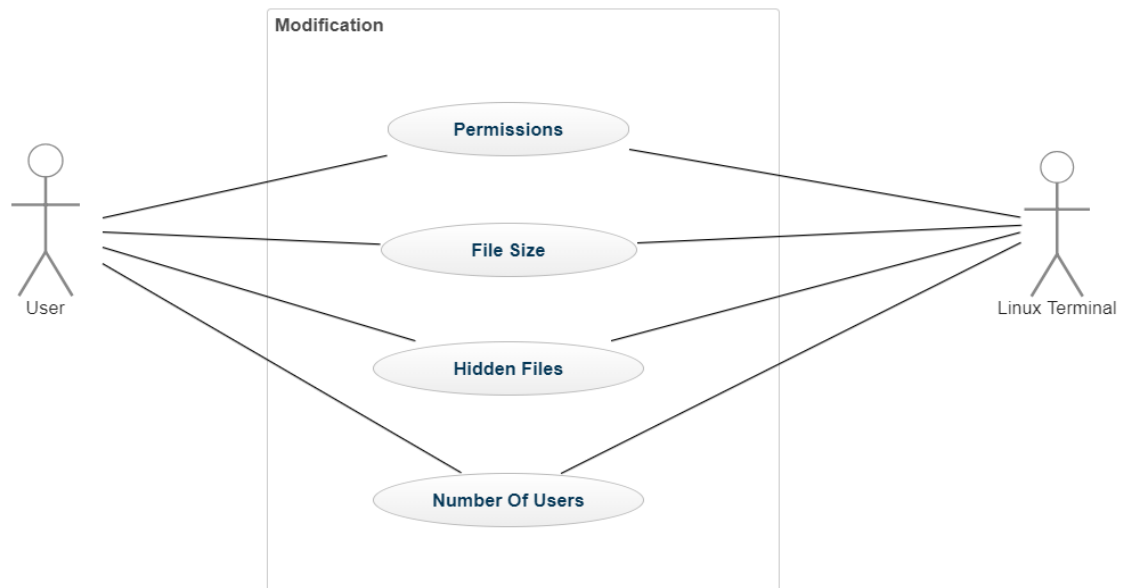
- Python
- Bash Scripting.

B) Hardware Requirements:

- A PC with Linux installed
- Internet connection.

USE -CASE DIAGRAM

Module 1: Log Management



1) Permissions:

Pre-condition: User must enter a valid path.

Flow Of Events: i) If the file permission's have been changed, the changed permissions are displayed.

Alternate Flow: i) The user does not enter a valid path.
ii) The file permissions haven't been changed.

2) File Size:

Pre-condition: User must enter a valid path

Flow Of Events: i) The list of files with their sizes are displayed.

Alternate Flow: i) User does not enter a valid path

3) Hidden Files:

Pre-condition: User must enter a valid path

Flow Of Events: i) The list of files that are hidden are displayed in the specified directory.

Alternate Flow: i) User does not enter a valid path

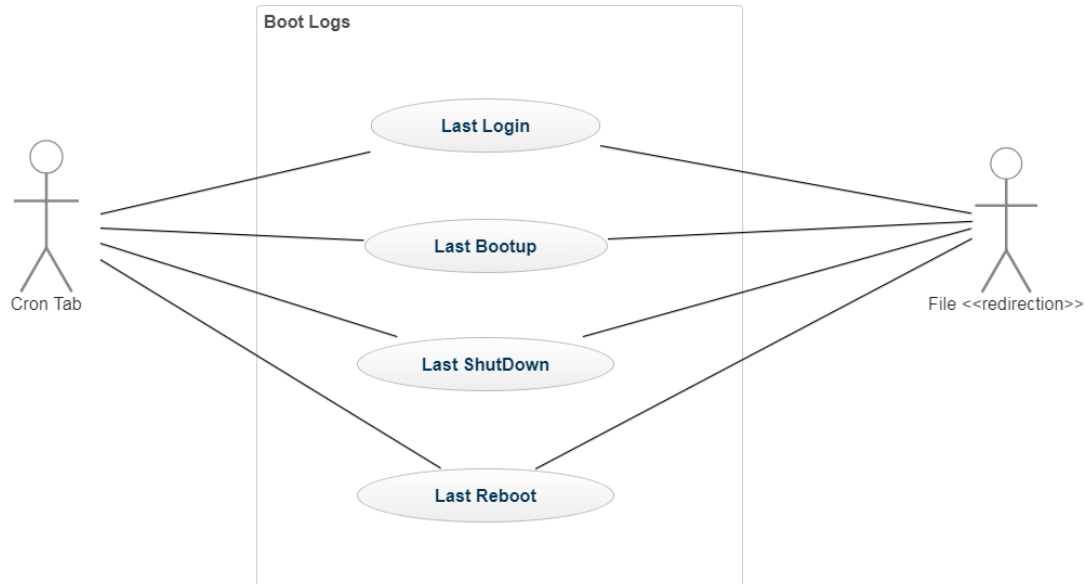
ii) No hidden files are present.

4) Number Of Users:

Pre-condition: -NIL-

Flow Of Events: i) Print all the users with /home directory.

Alternate Flow: -NIL-



1) Last Login:

Pre-condition: -NIL-

Flow Of Events: i) Cron Job is executed.

ii) The last login time is displayed.

iii) Saved in a file in the directory.

Alternate Flow: -NIL-

2) Last Bootup:

Pre-condition: -NIL-

Flow Of Events: i) Cron Job is executed.

ii) The last bootup time is displayed.

iii) Saved in a file in the directory.

Alternate Flow: -NIL-

3) Last Shutdown:

Pre-condition: -NIL-

Flow Of Events: i) Cron Job is executed.
ii) The last shutdown time is displayed.
iii) Saved in a file in the directory.

Alternate Flow: -NIL-

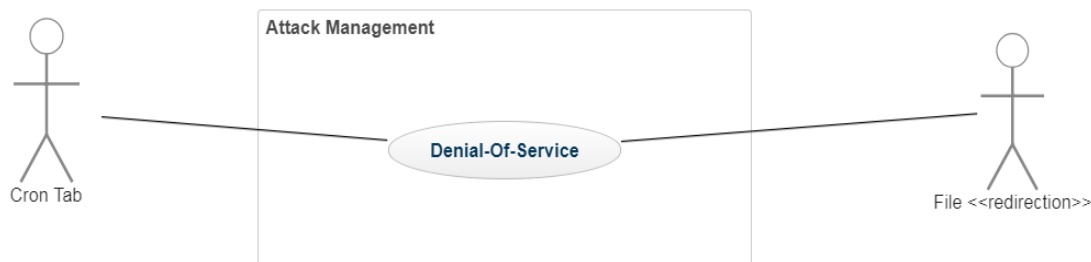
4) Last Reboot:

Pre-condition: -NIL-

Flow Of Events: i) Cron Job is executed.
ii) The last reboot time is displayed.
iii) Saved in a file in the directory.

Alternate Flow: -NIL-

MODULE 2: ATTACK MANAGEMENT



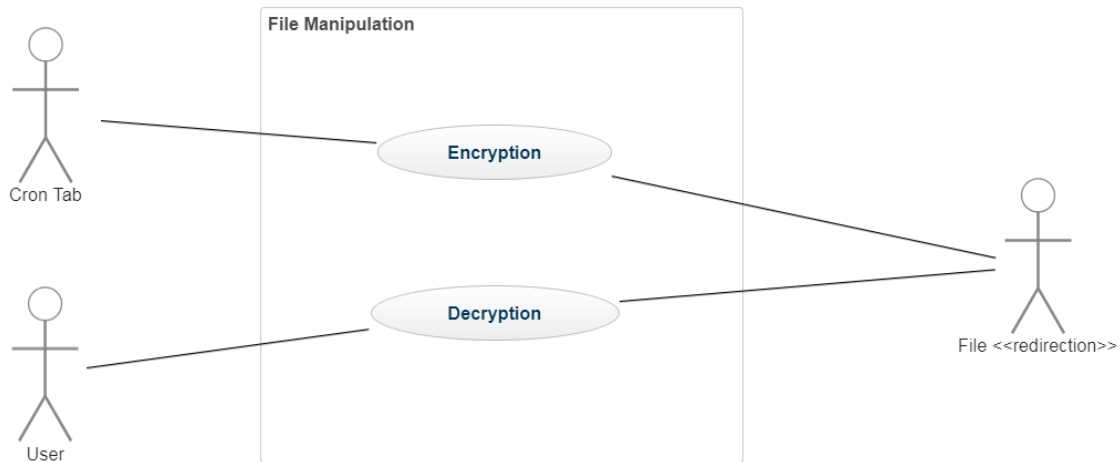
1) Denial-Of-Service:

Pre-Condition: i) The attack should be in progress.

Flow Of Events: i) All the IP Addresses in the range are scanned.
ii) Search for SYN Packets in the packets.
iii) If found, the IP Address is blocked.

Alternate Flow: i) There is no attack.

MODULE 3: SECURITY MANAGEMENT



1) Encryption:

Pre-condition: List of files that need to be encrypted should be mentioned.

Flow Of Events: i) The mentioned files are encrypted.

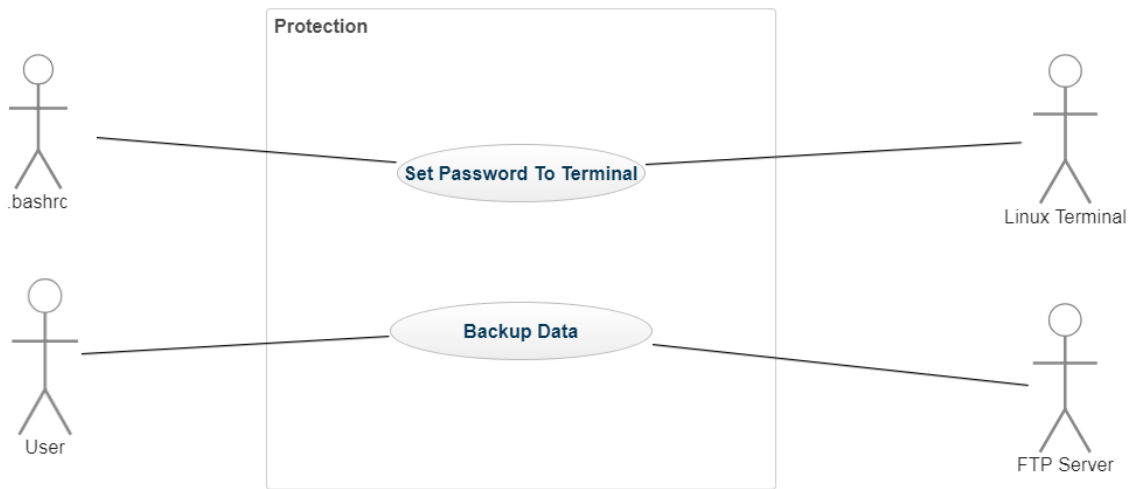
Alternate Flow: i) The files aren't found.

2) Decryption:

Pre-condition: List of all encrypted files should be mentioned.

Flow Of Events: i) The mentioned files are decrypted.

Alternate Flow: i) The files aren't found.



1) **Set Password To Terminal:**

Pre-condition: -NIL-

Flow Of Events: i) Set the password to the .bashrc file.

Alternate Flow: -NIL-

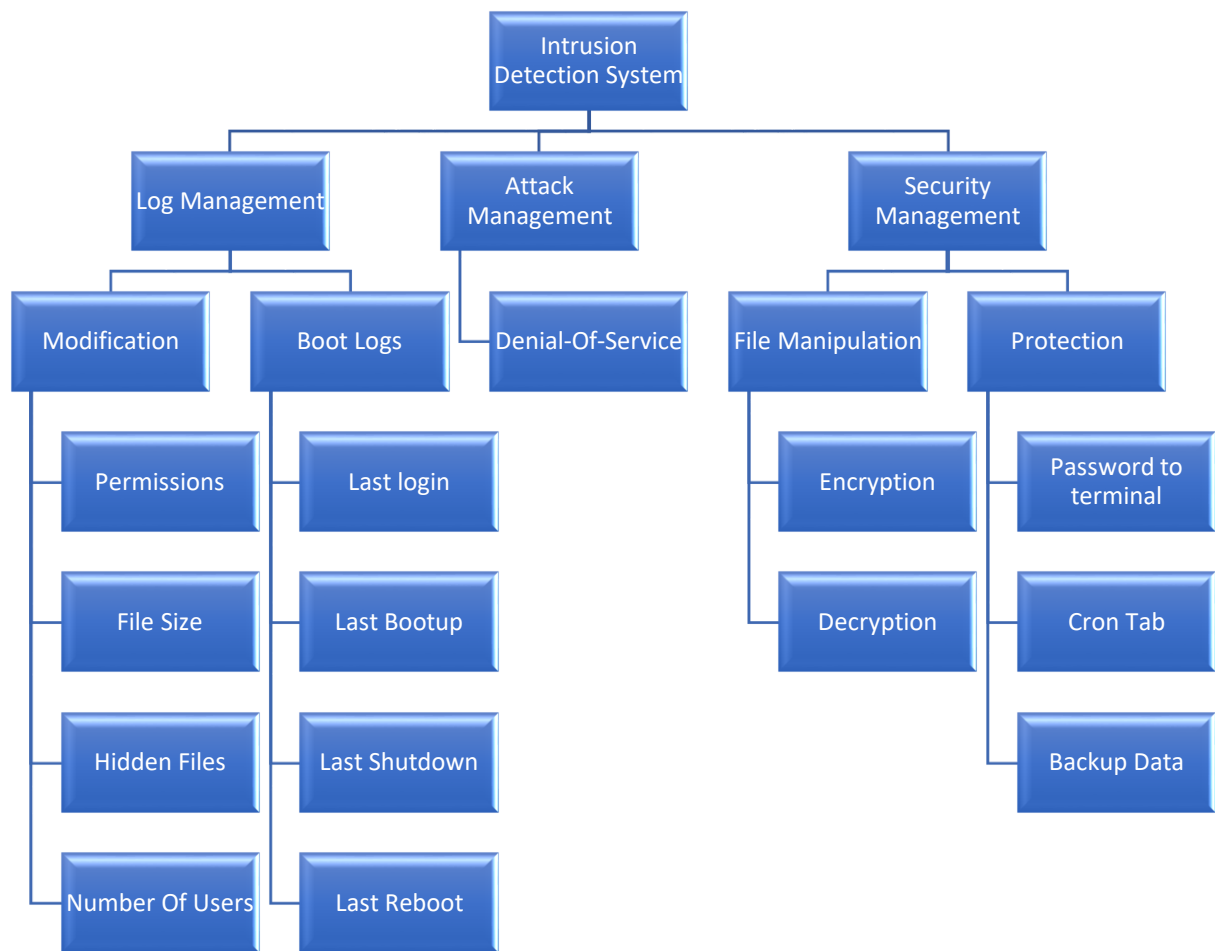
2) **Backup Data:**

Pre-condition: User must enter a valid path.

Flow Of Events: i) The files are backed up on the FTP Server.

Alternate Flow: i) The user does not enter a valid path.

SOFTWARE ARCHITECTURAL DESIGN



UI Design

We chose Command-Line Interface as our User Interface due to the following reasons:

- 1) **Speed:** Since only keyboard navigation is used unlike GUIs, it results in faster performance.
- 2) **Resources:** Since our application is already using a lot of resources, CLI consumes very less resources.
- 3) **CPU Processing time:** CLI also does not use much CPU Processing time as compared to GUI.
- 4) **Low Specification Systems:** Low specifications systems can also use our application due to CLI because it doesn't require high-end graphics, RAM, etc.

Also, Command-Line interfaces help to directly communicate with the Operating System functionalities, it is easy to modify any code and patch any bugs.

Test Plan and Test Cases

Purpose

This test plan describes the testing approach and overall framework that will drive the testing of Shaktimaan, our Intrusion Detection System. The document introduces:

- **Test Strategy:** rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions), description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform).
- **Test Management:** The test cases, inputs and outputs, what the expected output is supposed to be and what the actual output is, the action/pre-requisites performed, whether the test case is passed/failed.

Test Strategy

Test Objective: The objective is to verify all the functionalities of our project, Shaktimaan- The Intrusion Detection System and verify if all the modules work according to the specifications.

The test cases will also specify the severity of the functionalities and/or any defects/bugs in it.

Test Case ID	TC01				
Test Case Description	Checking if the permissions of a file has been changed				
Pre-Requisite	The path entered is a valid path				
	Test Priority	Low			
	Post-Requisite	None			
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 Accept the Path from User	/root/Desktop	Displays Modified Permissions	Displayed Modified Permissions	PASS
	2 Accept the Path from User	/root/De	Displays Invalid Path	Displayed Invalid Path	PASS

Test Case ID	TC03				
Test Case Description	Displays all the hidden files				
Pre-Requisite	The path entered is a valid path				
	Test Priority	Low			
	Post-Requisite	None			
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 Accept the Path from User	/root/Desktop	Displays all the hidden files	Displayed all the hidden files	PASS
	2 Accept the Path from User	/root/De	Displays Invalid Path	Displayed Invalid Path	PASS

Test Case ID	TC02				
Test Case Description	List all the files with their sizes				
Pre-Requisite	The path entered should be valid				
	Test Priority	Low			
	Post-Requisite	None			
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 Accept the Path from User	/root/Desktop	Displays Files with file size	Displays Files with file size	PASS
	2 Accept the Path from User	/root/De	Displays Invalid Path	Displayed Invalid Path	PASS

Test Case ID	TC04				
Test Case Description	Displays all the users with /home directories.				
Pre-Requisite	None				
	Test Priority	Medium			
	Post-Requisite	None			
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 User enters the value for checking all the users.	Value for checking users	List of all users with /home is displayed	List of all users with /home is displayed	PASS

Test Case ID	TC05				
Test Case Description	Check the last login of the user		Test Priority	Low	
Pre-Requisite	System should be logged in at least once		Post-Requisite	None	
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 Executing through CRON	NIL	Status of last login by the user	Status of last login by the user is displayed	PASS

Test Case ID	TC06				
Test Case Description	Check for the last bootup time		Test Priority	Low	
Pre-Requisite	Crontab should be installed.		Post-Requisite	None	
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 Executing through CRON	None	Displays the last bootup time	Displayed the last bootup time	PASS

Test Case ID	TC07				
Test Case Description	Check for the Last Shutdown time		Test Priority	Low	
Pre-Requisite	Should be Booted up at least once		Post-Requisite	None	
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 Executes through CRON	NIL	Displays last ShutDown time	Displays last ShutDown time	PASS

Test Case ID	TC08				
Test Case Description	Check for the last reboot time		Test Priority	Low	
Pre-Requisite	Cron tab should be installed.		Post-Requisite	None	
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 Executing through CRON	None	Displays the last Reboot time	Displayed the last Reboot time	PASS

Test Case ID	TC09				
Test Case Description	Prevention Of Denial Of Service Attack		Test Priority	Critical	
Pre-Requisite	Should be connected in a network		Post-Requisite	None	
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 Monitoring traffic on various IP's	None	Monitoring the traffic	Monitoring the traffic	PASS
	2 Scanning SYN Packets	None	Scans for SYN Packets	Scans for SYN Packets	PASS
	3 Blocking IP	SYN Packets - Sender	Blocks the IP of SYN Packets - Sender	Blocks the IP of SYN Packets - Sender	PASS
	4 If IP address is Spoofed	None	Block the IP of SYN Packets - Sender	Does not identify the SYN Packets	FAIL




Test Case ID	TC10				
Test Case Description	Encryption of Files		Test Priority	High	
Pre-Requisite	The path entered should be a valid path		Post-Requisite	None	
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 Executes through CRON	NIL	Encrypts the files	Encryption of files	PASS

Test Case ID	TC11				
Test Case Description	Decryption of the files		Test Priority	Medium	
Pre-Requisite	File should exist		Post-Requisite	None	
S.No	Action	Input	Expected Output	Actual Output	Test Result
	1 Accept the Username & Password	test,test	Login successful	Login successful	PASS
	2 Accept the Username & Password	admin,test	Failed to Login	Failed to login	PASS
	3 Accept the Username & Password	test,admin	Failed to Login	Failed to login	PASS
	4 Accept the Username & Password	abc,abc	Failed to Login	Failed to login	PASS

Test Case ID	TC12				
Test Case Description	Set Password to Terminal				
Pre-Requisite	Open the terminal	Test Priority	Medium		
		Post-Requisite	None		
S.No	Action	Input	Expected Output	Actual Output	Test Result
1	On opening the terminal prompt for username and password	None	Prompting the username and password	Prompting the username and password	PASS
2	Input the username and password	test,test	Successfully logged in	Successfully logged in	PASS
3	Input the username and password	test,admin	Unsuccessfull login	Unsuccessfull login	PASS
4	Input the username and password	admin,test	Unsuccessfull login	Unsuccessfull login	PASS
5	Input the username and password	admin,admin	Unsuccessfull login	Unsuccessfull login	PASS

Test Case ID	TC13				
Test Case Description	Backup over ftp				
Pre-Requisite	Connection to ftp	Test Priority	High		
		Post-Requisite	None		
S.No	Action	Input	Expected Output	Actual Output	Test Result
1	Accept the Path from User	/root/Desktop	Creates a backup on ftp server	Creates the backup on ftp server	PASS
2	Accept the Path from User	/root/De	Displays Invalid Path	Displayed Invalid Path	PASS
3	Connection to ftp	Network Req	Connected to FTP Server	Connected to FTP Server	PASS
4	Connection to ftp	No Network	Backup Failed	Backup Failed	PASS

Execution:

Exit Criteria	Status
100% Test Scripts executed	
96.6% pass rate of Test Scripts	
All expected and actual results are captured and documented with the test script	

Testing Strategy:

1) Unit Testing:

A Unit testing is a Level of Testing where smallest part of individual unit / component (called unit) is tested to determine if they are fit for use. The main intention of this activity is to check whether units are working as per design and handling error and exception more neatly.

We divided our whole project into 3 main modules, namely, Log Management, Attacks Management and Security Management. These modules are further divided into various sub-modules for easy understanding, efficiency of code and for easy testing.

During testing, we individually tested each and every component of the sub-module.

Let's take an example of a component of the sub-module, named, "Denial-Of-Service Prevention". In this, we scanned for each and every IP Address and MAC Addresses, and check if SYN packets are received. This is our checkpoint. We then tested if the SYN packets are being captured by our code. After that, we blocked the IP address by putting the captured IP Address in the IP Tables.

We tested this sub-module by attacking a system and by manually running our code to check if the IP Address of the attacker is being captured and is being added in the IP Tables and blocked.

We tested like this for each and every component individually.

2) Integration Testing:

Integration Testing is a level of software testing where individual units are combined and tested as a group.

The purpose of this level of testing is to expose faults in the interaction between integrated units.

We integrated all the sub-modules, and its components first together to see if it's working together properly and then integrating all the main modules into our UI. Say, our third module, "Security Management" has 3 main sub-modules, "Encryption and Decryption", "Backup Data" and "Cron Tab".

After testing each of the three sub-modules, we first combined "Cron Tab" with "Encryption and Decryption". On switching on the system, the encryption code is executed with the help of cron tab. The same was done with "Backup Data".

After this, the whole module was appended to our UI and tested again with all the other modules that have been tested in the same way.

Project Plan:

<u>Task Name</u>	<u>Duration</u>	<u>Start</u>	<u>End</u>
Abstract and Study	3 days	26 June 2018	28 June 2018
Requirement Analysis	11 days	1 July 2018	11 July
- Requirement Gathering	3 days	1 July 2018	3 July 2018
-Group Interaction	3 days	5 July 2018	7 July 2018
-Analysis	4 days	8 July 2018	11 July 2018
Review 1	1 day	30 July 2018	30 July 2018
Procedure & Methodology	4 days	3 August 2018	6 August 2018
Review 2	3 days	17 August 2018	19 August 2018
Coding	66 days	9 September 2018	12 February 2019
-Log Management	15 days	9 September 2018	23 September 2018
-Attack Management	26 days	1 October 2018	26 October 2018
-Security Management	25 days	17 November 2019	12 December 2019
UI Design	4 days	20 January 2019	23 January 2019
Testing	20 days	25 January 2019	14 February 2019
-Unit Testing	10 days	25 January 2019	4 February 2019
-Integration Testing	10days	6 February 2019	16February 2019

Roles and Responsibilites:

Task	In-Charge
Requirement Analysis, review and gathering	Vedansh Vachani
Attack Management	Bhaskar Sharma
Security Management	Sneha Irukuvajjula
Log Management	Vedansh Vachani
UI Design	Bhaskar Sharma
Testing	Sneha Irukuvajjula

Drawbacks and future enhancements

1) IP Spoofing in Denial Of Service Attack:

Introduction:

As we have tested in the second module, i.e, the Attack Management Module, we have seen how our Intrusion Detection System sniffs the attacker and blocks the attacker.

Our IDS first scans for all the IP Addresses that are connected in the network and sniffs for all the packets being transferred.

If there are any SYN packets being transferred, the IP Address of the attacker is obtained and sent to the IP Tables and block the attacker's IP Address to prevent any other malicious packets to transfer.

The Loophole:

Now let's say, the attacker uses some tool for Denial-Of Service Attack and attacks in the first place and somehow identifies that our system is able to detect the traffic and blocks the attacker's IP Address.

The next time, the attacker comes with a solution to spoof his/her IP Address so that our system does not recognise the IP address and it cannot block the real attacker. In IP Spoofing, multiple packets are sent through multiple IP Addresses, thus flooding the victim's IP Address resulting in a system/server crash (which is more likely a DDOS (Distributed Denial Of Service)).

The Problem :

It is pretty difficult to detect a spoofed IP Address as IP spoofing is a concept in which the IP packets are transferred using a fake IP address in the beginning.

Probable Solution:

These are the probable solutions we can come up with **if and only if** these two conditions are satisfied:

- i) We have the list of all static IP Addresses assigned to all the devices and,
- ii) We have the list of all MAC addresses in the network.

Now, if the attacker spoofs his/her IP address and try to do a Denial-Of-Service attack, we can easily detect those network packets using Wireshark and filtering the MAC addresses.

And if the attacker spoofs his/her MAC address, we can easily detect the IP packets through Wireshark by filtering out the IP addresses.

2) Sudoers accessing the data

Introduction:

As we have seen in the third module, i.e, Security Management, that we have set a password each and every time the terminal is opened. This is a good feature or in other words, a good add-on to the security of the system, as even if a hacker somehow gets to know the root/super user password, there is another password to crack! It is more like a two-step security.

The Loophole:

Let's say, the user logs in as root and opens the terminal. The user forgets his/her password of terminal. There's no option to open the terminal now!

Probable Solution:

The probable solution we can come up with this is to add a facility of a two-step verification, wherein the user is prompted for the credentials like root/superuser password, security question, etc.

SAMPLE CODE

1) Hidden Files:

```
#!/usr/bin/python
import os
#Code for searching the hidden files in the given directory
path=raw_input("Enter the path : ")
print("-----")
def hiddenFile(path):
    for root,dirs,files in os.walk(path,topdown=False): #looping through the directory '.'
        means current working directory
        for dirname in dirs:
            if(dirname[0]=="."): #if the directory name starts with .
                print (os.path.abspath(os.path.join(root)))
                print(dirname) #print the directory name
                print("-----")
        for filename in files: #looping through the files in the directory
            if(filename[0]=="."): #if the file starts with '.'
                print (os.path.abspath(os.path.join(root)))
                print(filename) #print file name
                print("-----")
if(os.path.exists(path)):
    hiddenFile(path)
else:
    print "Wrong path"
```

2) Cron Tab:

```
#!/bin/bash
#write out current crontab
crontab -l > mycron #lists out all the cron jobs
#echo new cron into cron file
echo "@reboot sleep 10; /etc/shaktimaan/Reports.sh >>
/etc/shaktimaan/logs/BootReports" >> mycron
echo "@reboot sleep 10; /etc/shaktimaan/ShutDownreport.sh >>
/etc/shaktimaan/logs/ShutdownReport" >> mycron
echo "@reboot sleep 10; /etc/shaktimaan/RebootReport.sh >>
/etc/shaktimaan/logs/RebootReport" >> mycron
echo "*/3 * * * * /etc/shaktimaan/dos.sh" >> mycron
echo "@reboot sleep 10; /etc/shaktimaan/file_encrypt" >> mycron
crontab mycron #install new cron file
rm mycron
```

3) Denial-Of-Service:

```
#!/bin/bash
first="$(echo `hostname -I | awk '{print $1}'` | cut -d'.' -f1)"
second="$(echo `hostname -I | awk '{print $1}'` | cut -d'.' -f2)"
third="$(echo `hostname -I | awk '{print $1}'` | cut -d'.' -f3)"
fourth="$(echo `hostname -I | awk '{print $1}'` | cut -d'.' -f4)"
if [[ $first -ge 1 && $first -le 126 ]]
then
    ip_add=$first".0.0.0/24"
elif [[ $first -ge 128 && $first -le 191 ]]
then
    ip_add=$first"."$second".0.0/24"
elif [[ $first -ge 192 && $first -le 223 ]]
then
    ip_add=$first"."$second"."$third".0/24"
fi
ip_array=$(sudo nmap -sP $ip_add | grep -E -o '[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+'))
#Scanning IP Addresses and storing in array
mac_array=$(sudo nmap -sP $ip_add | grep "MAC Address" | awk '{print $3}' )
#Scanning MAC Addresses and storing in array
ip_len=${#ip_array[@]}
for (( i=1; i<$ip_len; i++ ))
do
    if [ ${ip_array[$i]} != `hostname -I | awk '{print $1}'` ]
    then
        echo ${ip_array[$i]}
        sudo timeout 10 tshark -i wlan0 -Y "ip.src == `hostname -I | awk '{print $1}'` &&
ip.dst == ${ip_array[$i]}" >> /etc/shaktiman/logs/DOS #Scanning each IP for 10
seconds
        grep -a SYN log1
        if [ $? -eq 0 ] #If SYN packets are found
        then
            echo ${ip_array[$i]} "Is the attacker will block this "
            sudo iptables -A INPUT -s ${ip_array[$i]} -j DROP #IP of the attacker is
blocked
        else
            echo "You are safe and sound"
        fi
    fi
done
```

4) Installation script:

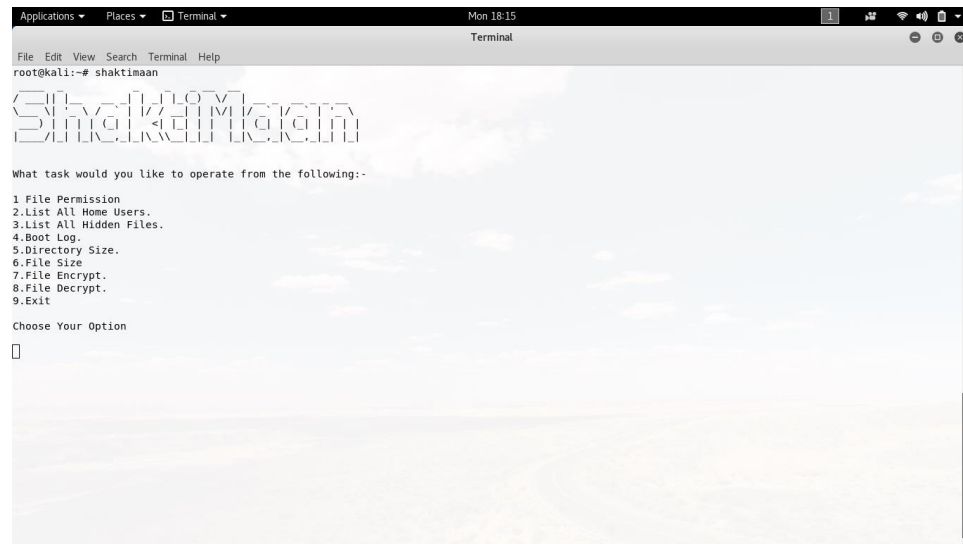
```
#!/bin/bash
sudo apt-get install tshark -y && sudo apt-get install cron && sudo apt-get install figlet
sudo apt-get install python2.7
echo "Enter the username for the terminal: "
read uname
echo $uname >> fakeshadow
echo "Enter the password for the terminal (Note: You cannot change the password again!): "
read pwd
sudo echo $pwd >> fakeshadow
sudo ./crontabsc.sh
sudo mv ~/Downloads/shaktimaan /etc
sudo cp -r /etc/shaktimaan/bashmenu/* /usr/bin/
sudo cat bashrcscript >> ~/.bashrc
```

5) Script that will run on opening of the terminal:

```
uname=`cat /etc/shaktimaan/fakeshadow | awk -F"| " 'NR==1'{print $1}`
pwd=`cat /etc/shaktimaan/fakeshadow | awk -F"| " 'NR==2'{print $1}`
echo "Enter username: "
read user
echo "Enter password: "
read -s password
if [ $user != $uname ]
then
    echo "Incorrect username"
    exit 0
else
    if [ $password == $pwd ]
    then
        echo "successful"
    else
        echo "Incorrect password"
        exit 0
    fi
fi
./file_decrypt
```

ANNEXURES AND SNAPSHOTS

Bash Menu



```
Applications Places Terminal
Mon 18:15
Terminal
File Edit View Search Terminal Help
root@kali:~# shaktimaan

S H A K T I M A A N
S H A K T I M A A N

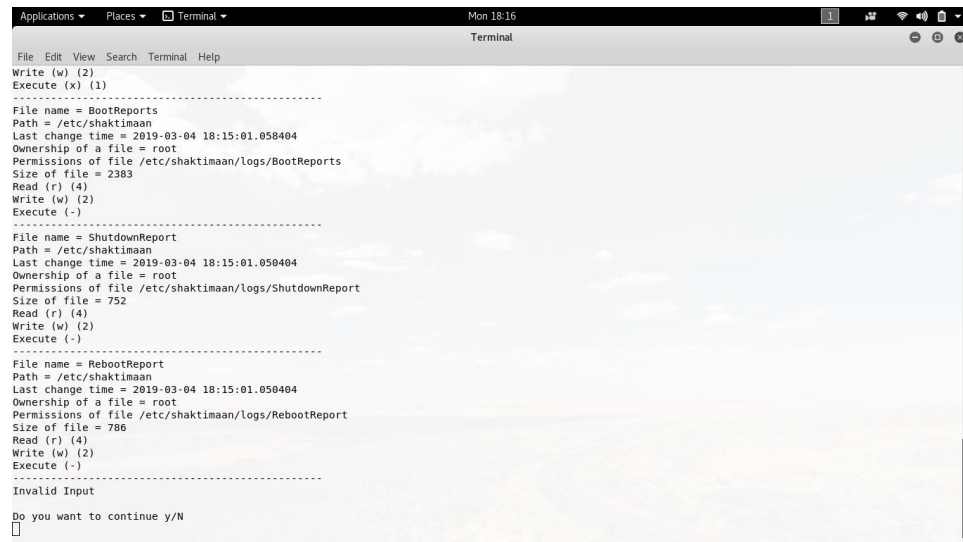
What task would you like to operate from the following:-

1.File Permission
2.List All Home Users.
3.List All Hidden Files.
4.Boot Log.
5.Directory Size.
6.File Size
7.File Encrypt.
8.File Decrypt.
9.Exit

Choose Your Option

```

Output of file permissions

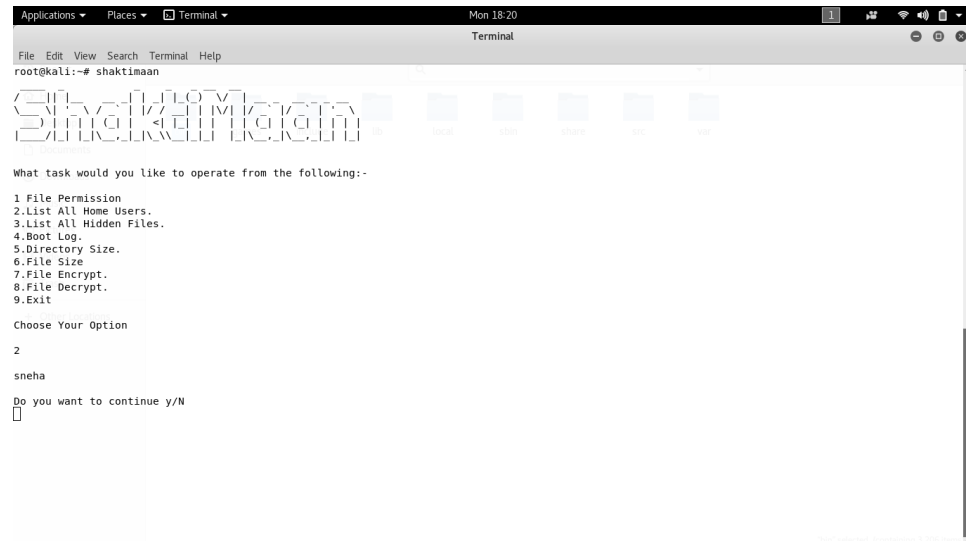


```
Applications Places Terminal
Mon 18:16
Terminal
File Edit View Search Terminal Help
Write (w) (2)
Execute (x) (1)
.....
File name = BootReports
Path = /etc/shaktimaan
Last change time = 2019-03-04 18:15:01.058404
Ownership of a file = root
Permissions of file /etc/shaktimaan/logs/BootReports
Size of file = 2383
Read (r) (4)
Write (w) (2)
Execute (-)
.....
File name = ShutdownReport
Path = /etc/shaktimaan
Last change time = 2019-03-04 18:15:01.058404
Ownership of a file = root
Permissions of file /etc/shaktimaan/logs/ShutdownReport
Size of file = 752
Read (r) (4)
Write (w) (2)
Execute (-)
.....
File name = RebootReport
Path = /etc/shaktimaan
Last change time = 2019-03-04 18:15:01.058404
Ownership of a file = root
Permissions of file /etc/shaktimaan/logs/RebootReport
Size of file = 786
Read (r) (4)
Write (w) (2)
Execute (-)
.....
Invalid Input

Do you want to continue y/N

```

Printing home users



```
Applications Places Terminal Mon 18:20
Terminal
File Edit View Search Terminal Help
root@kali:~# shaktimaan

SHAKTILOMKOTI

What task would you like to operate from the following:-

1.File Permission
2.List All Home Users.
3.List All Hidden Files.
4.Boot Log.
5.Directory Size.
6.File Size
7.File Encrypt.
8.File Decrypt.
9.Exit

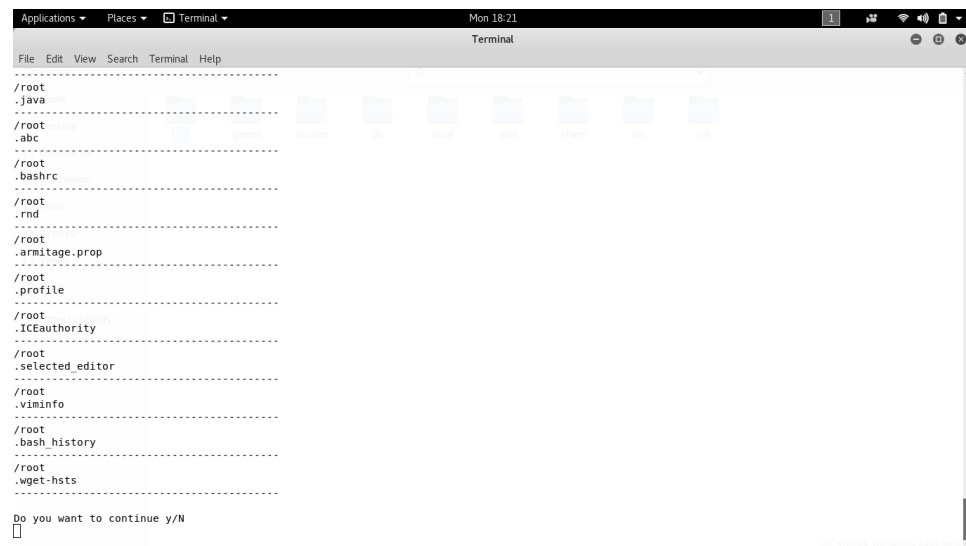
Choose Your Option
2

sneha

Do you want to continue y/N

```

Hidden Files



```
Applications Places Terminal Mon 18:21
Terminal
File Edit View Search Terminal Help
.....
/root
.java
.....
/root
.....
/root
abc
.....
/root
.bashrc
.....
/root
.rnc
.....
/root
.armitage.prop
.....
/root
.profile
.....
/root
.ICEauthority
.....
/root
.selected_editor
.....
/root
.viminfo
.....
/root
.bash_history
.....
/root
.wget-hsts
.....
Do you want to continue y/N

```

Boot Logs

[illegible]

Login history of user

```

Applications ▾ Places ▾ Terminal ▾ Mon 18:23
Terminal
File Edit View Search Terminal Help
Press 3 to check last reboot time
Press 4 to check last shutdown time
Press 5 to get the full reboot report
Press 6 to get the full shutdown report
Press 7 to exit

Your input:
1
Press 1 to check the login history of a specific user
Press 2 to check the login history of all users
Your input:
1
Enter the user name:
sneha

Login Report of
User:
sneha

7 20:31:40 2019

Do you want to continue y/N

```

Directory Size

```
Applications ▾ Places ▾ Terminal ▾ Mon 18:23
Terminal
File Edit View Search Terminal Help
┌───┴───┐
7 20:31:40 2019
Do you want to continue y/N
Y
What task would you like to operate from the following:-
1.File Permission
2.List All Home Users.
3.List All Hidden Files.
4.Boot Log.
5.Directory Size.
6.File Size
7.File Encrypt.
8.File Decrypt.
9.Exit
Choose Your Option
5
enter the directories name with full path = /etc/shaktimaan
-----
bashmenu
/etc/shaktimaan/bashmenu
4096
-----
logs
/etc/shaktimaan/logs
4096
-----
Do you want to continue y/N

```

```
Applications ▾ Places ▾ Terminal ▾ Mon 18:23
Terminal
File Edit View Search Terminal Help
┌───┴───┐
1883
-----
dirsSize
/etc/shaktimaan/bashmenu/dirsSize
553
-----
bootLog
/etc/shaktimaan/bashmenu/bootlog
1424
-----
ListUsers
/etc/shaktimaan/bashmenu/ListUsers
44
-----
shaktimaan
/etc/shaktimaan/bashmenu/shaktimaan
1043
-----
filesSize
/etc/shaktimaan/bashmenu/filesSize
573
-----
menu_encrypt
/etc/shaktimaan/bashmenu/menu_encrypt
1052
-----
menu_decrypt
/etc/shaktimaan/bashmenu/menu_decrypt
859
-----
HiddenPython
/etc/shaktimaan/bashmenu/HiddenPython
893
-----
Do you want to continue y/N

```

Summary:

Intrusion Detection Systems help information systems prepare for, and deal with attacks. They accomplish this by collecting information from a variety of systems and network sources, and then analyzing the information for possible security problems. An Intrusion Detection Systems (IDS) examines system or network activity to find possible intrusions or attacks. Intrusion Detection Systems are either network-based or host-based.

Network based Intrusion Detection Systems are most common, and examine passing network traffic for signs of intrusion.

Host-based systems look at user and process activity on the local machine for signs of intrusion. Since each type has specific strengths and weaknesses.

If you know the enemy and know yourself, you need not fear the result of a hundred battles.

Bibliography:

- [1] R.Durst, T.Champion, B.Witten, E.Miller, And L.Spagnuolo, "Testing And Evaluating Computer Intrusion DetectionSystem"Communications Of Acn, Vol.2, No.7, Pp 53-61, 1999.
- [2] Mrutyunjaya Panda And ManasRanjanPatra, "Network Intrusion Detection Using Naïve Bayes", Ijcsns International Journal Of Computer Science And Network Security, Vol 7 No.12 , 2007.
- [3] Christopher Low –“Understanding Wireless Attacks &Detection “-Giac Security Essentials Certification (Gsec) Practical Assignment 13 April 2005
- [4] PeymanKabiri And Ali A. Ghorbani- “Research On Intrusion Detection And Response:A Survey”-University Of New Brunswick, 2, Sep. 2005
- [5] G. A. Fink, B. L. Chappell, T. G. Turner, And K. F. O’donoghue- “Metrics-Based Approach To Intrusion Detection System Evaluation For Distributed Real-Time Systems”-Information Transfer Technology Group, Code B35, Naval Surface Warfare Center, Dahlgren Division
- [6] Dr. S.Vijayarani1 And Ms. Maria Sylviaa.S - “Intrusion Detection System – A Study”- International Journal Of Security, Privacy And Trust Management (Ijsptm) Vol , No 1, February 2015.
- [7] Asmaa Shaker Ashoor, Prof.Sharad Gore - “Intrusion Detection System (Ids) &Intrusion Prevention System (Ips): Case Study” International Journal Of Scientific & Engineering Research Volume 2, Issue 7, July-2011.
- [8] Vikram Kothari, ManaliRaut , SairajSamant , “Secure Gateway Defender-A Network Intrusion Detection System”,2017
- [9] M. Ali Aydın *, A. Halim Zaim, K. GökhanCeylan- “A Hybrid IntrusionDetection System Design For Computer Network Security” Computers And Electrical Engineering 35 (2009) 517–526 .
- [10] Dr. S.Vijayarani1 And Ms. Maria Sylviaa.S - “Intrusion Detection System – A Study”- International Journal Of Security, Privacy And Trust Management (Ijsptm) Vol 4, No 1, February 2015.
- [11] Asmaa Shaker Ashoor, Prof.Sharad Gore - “Intrusion Detection System (Ids) &Intrusion Prevention System (Ips): Case Study” International Journal Of Scientific & Engineering Research Volume 2, Issue 7, July-2011.

