**BASAVARAJESWARI GROUP OF INSTITUTIONS**

**Ballari Institute of Technology & Management**

**AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITYJNANA SANGAMA, BELAGAVI 590018**

**INTERNSHIP**

**Report On**

# Basic Intrusion Detection System

Submitted in partial fulfillment of the requirements for the award of degree of

**Bachelor of Engineering**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

## Submitted by

**E Harsha**

**3BR22CS049**

## Internship Carried Out By

**EZ TRAININGS & TECHNOLOGIES PVT.LTD HYDERABAD**

**Internal Guide External Guide**

**S.STEFFI NIVEDITA RAGAVAN P**

**Asst.prof,CSE Technical Trainer**

**VARADA ALEKHYA**

**Asst. prof,CSE**

### BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution\*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)**

**"Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India) Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197**

**2023-2024**

**BASAVARAJESWARI GROUP OF INSTITUTIONS**

## BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

**Autonomous institute under VISVESVARAYA TECHNOLOGICAL UNIVERSITYJNANA SANGAMA,**

**BELAGAVI 590018**

NACC Accredited Institution\*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to**

**Visvesvaraya Technological University, Belagavi)**

### "JnanaGangotri"Campus,No.873/2,Ballari-HospetRoad,Allipur,

**Ballar1-583 104 (Karnataka)(India)**

**Ph: 08392 – 237100 / 237190, Fax: 08392 –237197**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# CERTIFICATE

This is to certify that the Internship entitled **“ Basic Intrusion Detection System ”** has been successfully completed by **E.Harsha** bearing USN **3BR22CS049** a bonafide student of Ballari Institute of Technology and Management,

Ballari. For the partial fulfillment of the requirementsfor the **Bachelor’s Degree in Computer**

**Science and Engineering** of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY,

Belagavi during the academic year 2023-2024.

**Signature of Internship Signature of HOD**

**Co-ordinator**

**S.STEFFI NIVEDITA R N KULKARNI**

**Asst.prof,CSE Prof. and HOD(CSE)**

**VARADA ALEKHYA**

**Asst. prof,CSE**

**DECLARATION**

E.HARSHA **,** second year student of Computer

Science and Engineering, Ballari Institute of Technology, Ballari, declare that Internship entitled **EVENT MANAGEMENT PLATFORM** is a part of Internship Training successfully carried out by **EZ TECHNOLOGIES & TRAININGS PVT.LTD ,Hyderabad** at “**BITM,BALLARI”.** This report is submitted in partial fulfillment of the requirements for the award of the degree, Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi.

**Date : 28.09.2024 Signature of the Student**

**Place : Ballari**

**ACKNOWLEDGEMENT**

The satisfactions that a company the successful completion of my internship on “ Basic Intrusion Detection System ” would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is my privilege to express my gratitude and respect to all those who inspired me in the completion of my internship.

I am grateful to our respective coordinator **“S**.**Steffi Nivedita (Asst.prof,CSE) , Varada Alekya (Asst.prof,CSE)”** for his noble gesture, support co-ordination and valuable suggestions given to me in the completion of Internship.

I also thank **R N Kulkarni,** H.O.D. Department of **Computer science and engineering** for extending all his valuable support and encouragement.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Chapter Name** | **Page No.** |
| **1** | **Company Profile** | **01** |
| **2** | **Day to day activity(student diary extract)** | **02** |
| **3** | **Abstract** | **03** |
| **4** | **Introduction of the project** | **04** |
| **5** | **Description** | **05** |
| **6** | **Algorithm** | **06-07** |
| **7** | **Output** | **08-11** |
| **8** | **Conclusion** | **12** |
| **9** | **References** | **13** |

**INTRODUCTION:**

**What is an Intrusion Detection System?**

A system called an intrusion detection system (IDS) observes network traffic for malicious transactions and sends immediate alerts when it is observed. It is software that checks a network or system for malicious activities or policy violations. Each illegal activity or violation is often recorded either centrally using an SIEM system or notified to an administration. IDS monitors a network or system for malicious activity and protects a computer network from unauthorized access from users, including perhaps insiders. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between ‘bad connections’ (intrusion/attacks) and ‘good (normal) connections’.

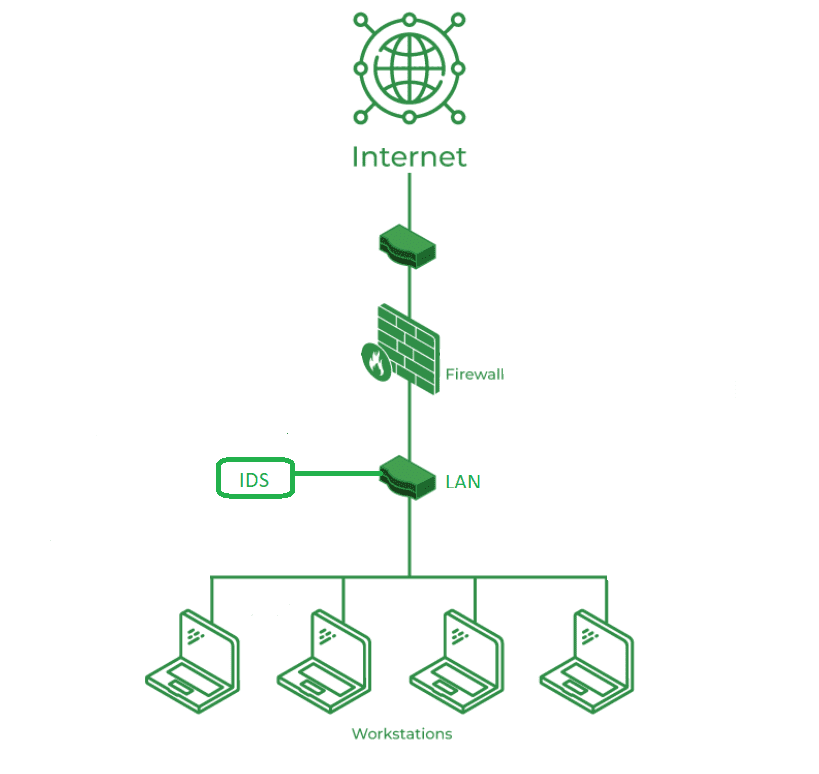
**Working of Intrusion Detection System(IDS)**

* An IDS (Intrusion Detection System) monitors the traffic on a computer network to detect any suspicious activity.
* It analyzes the data flowing through the network to look for patterns and signs of abnormal behavior.
* The IDS compares the network activity to a set of predefined rules and patterns to identify any activity that might indicate an attack or intrusion.
* If the IDS detects something that matches one of these rules or patterns, it sends an alert to the system administrator.
* The system administrator can then investigate the alert and take action to prevent any damage or further intrusion.

**Classification of Intrusion Detection System(IDS):**

 Intrusion Detection System are classified into 5 types:

* **Network Intrusion Detection System (NIDS):** Network intrusion detection systems (NIDS) are set up at a planned point within the network to examine traffic from all devices on the network. It performs an observation of passing traffic on the entire subnet and matches the traffic that is passed on the subnets to the collection of known attacks. Once an attack is identified or abnormal behavior is observed, the alert can be sent to the administrator. An example of a NIDS is installing it on the subnet where [firewalls](https://www.geeksforgeeks.org/introduction-of-firewall-in-computer-network/) are located in order to see if someone is trying to crack the [firewall](https://www.geeksforgeeks.org/introduction-of-firewall-in-computer-network/).
* **Host Intrusion Detection System (HIDS):** Host intrusion detection systems (HIDS) run on independent hosts or devices on the network. A HIDS monitors the incoming and outgoing packets from the device only and will alert the administrator if suspicious or malicious activity is detected. It takes a snapshot of existing system files and compares it with the previous snapshot. If the analytical system files were edited or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission-critical machines, which are not expected to change their layout.



*Intrusion Detection System (IDS)*

* **Protocol-Based Intrusion Detection System (PIDS):** Protocol-based intrusion detection system (PIDS) comprises a system or agent that would consistently reside at the front end of a server, controlling and interpreting the protocol between a user/device and the server. It is trying to secure the web server by regularly monitoring the [HTTPS protocol](https://www.geeksforgeeks.org/explain-working-of-https/) stream and accepting the related [HTTP protocol](https://www.geeksforgeeks.org/http-full-form/). As HTTPS is unencrypted and before instantly entering its web presentation layer then this system would need to reside in this interface, between to use the HTTPS.
* **Application Protocol-Based Intrusion Detection System (APIDS):** An application [Protocol-based Intrusion Detection System](https://www.geeksforgeeks.org/intrusion-detection-system-ids/) (APIDS) is a system or agent that generally resides within a group of servers. It identifies the intrusions by monitoring and interpreting the communication on application-specific protocols. For example, this would monitor the SQL protocol explicitly to the middleware as it transacts with the database in the web server.
* **Hybrid Intrusion Detection System:** Hybrid intrusion detection system is made by the combination of two or more approaches to the intrusion detection system. In the hybrid intrusion detection system, the host agent or system data is combined with network information to develop a complete view of the network system. The hybrid intrusion detection system is more effective in comparison to the other intrusion detection system. Prelude is an example of Hybrid IDS.

**What is an Intrusion in Cybersecurity?**

Understanding Intrusion Intrusion is when an attacker gets unauthorized access to a device, network, or system. Cyber criminals use advanced techniques to sneak into organizations without being detected. Common methods include:

* **Address Spoofing:**Hiding the source of an attack by using fake, misconfigured, or unsecured proxy servers, making it hard to identify the attacker.
* **Fragmentation**: Sending data in small pieces to slip past detection systems.
* **Pattern Evasion:** Changing attack methods to avoid detection by IDS systems that look for specific patterns.
* **Coordinated Attack:** Using multiple attackers or ports to scan a network, confusing the IDS and making it hard to see what is happening.

**Intrusion Detection System Evasion Techniques**

* **Fragmentation:**Dividing the packet into smaller packet called fragment and the process is known as [fragmentation](https://www.geeksforgeeks.org/what-is-fragmentation-in-operating-system/). This makes it impossible to identify an intrusion because there can’t be a malware signature.
* **Packet Encoding:** Encoding packets using methods like Base64 or hexadecimal can hide malicious content from signature-based IDS.
* **Traffic Obfuscation:** By making message more complicated to interpret, obfuscation can be utilised to hide an attack and avoid detection.
* **Encryption:**Several security features, such as data integrity, confidentiality, and data privacy, are provided by [encryption](https://www.geeksforgeeks.org/what-is-data-encryption/). Unfortunately, security features are used by malware developers to hide attacks and avoid detection.

**Benefits of IDS**

* **Detects Malicious Activity:** IDS can detect any suspicious activities and alert the system administrator before any significant damage is done.
* **Improves Network Performance:** IDS can identify any performance issues on the network, which can be addressed to improve network performance.
* **Compliance Requirements:** IDS can help in meeting compliance requirements by monitoring network activity and generating reports.
* **Provides Insights:** IDS generates valuable insights into network traffic, which can be used to identify any weaknesses and improve network security.

**Detection Method of IDS**

* **Signature-Based Method:** Signature-based IDS detects the attacks on the basis of the specific patterns such as the number of bytes or a number of 1s or the number of 0s in the network traffic. It also detects on the basis of the already known malicious instruction sequence that is used by the malware. The detected patterns in the IDS are known as signatures. Signature-based IDS can easily detect the attacks whose pattern (signature) already exists in the system but it is quite difficult to detect new malware attacks as their pattern (signature) is not known.
* **Anomaly-Based Method:** Anomaly-based IDS was introduced to detect unknown malware attacks as new malware is developed rapidly. In anomaly-based IDS there is the use of machine learning to create a trustful activity model and anything coming is compared with that model and it is declared suspicious if it is not found in the model. The machine learning-based method has a better-generalized property in comparison to signature-based IDS as these models can be trained according to the applications and hardware configurations.

**Comparison of IDS with Firewalls:**

IDS and firewall both are related to network security but an IDS differs from a [firewall](https://www.geeksforgeeks.org/introduction-of-firewall-in-computer-network/)as a firewall looks outwardly for intrusions in order to stop them from happening. Firewalls restrict access between networks to prevent intrusion and if an attack is from inside the network it doesn’t signal. An IDS describes a suspected intrusion once it has happened and then signals an alarm.

**Why Are Intrusion Detection Systems (IDS) Important?**

An Intrusion Detection System (IDS) adds extra protection to your cybersecurity setup, making it very important. It works with your other security tools to catch threats that get past your main defenses. So, if your main system misses something, the IDS will alert you to the threat.

**Placement of IDS**

* The most optimal and common position for an IDS to be placed is behind the firewall. Although this position varies considering the network. The ‘behind-the-firewall’ placement allows the IDS with high visibility of incoming network traffic and will not receive traffic between users and network. The edge of the network point provides the network the possibility of connecting to the extranet.
* In cases, where the IDS is positioned beyond a network’s firewall, it would be to defend against noise from internet or defend against attacks such as port scans and network mapper.An IDS in this position would monitor layers 4 through 7 of the [OSI model](https://www.geeksforgeeks.org/open-systems-interconnection-model-osi/) and would use Signature-based detection method. Showing the number of attemepted breacheds instead of actual breaches that made it through the firewall is better as it reduces the amount of false positives. It also takes less time to discover successful attacks against network.
* An advanced IDS incorporated with a firewall can be used to intercept complex attacks entering the network. Features of advanced IDS include multiple security contexts in the routing level and bridging mode. All of this in turn potentially reduces cost and operational complexity.
* Another choice for IDS placement is within the network. This choice reveals attacks or suspicious activity within the network. Not acknowledging security inside a network is detrimental as it may allow users to bring about security risk, or allow an attacker who has broken into the system to roam around freely.

**Advantages**

* **Early Threat Detection**: IDS identifies potential threats early, allowing for quicker response to prevent damage.
* **Enhanced Security**: It adds an extra layer of security, complementing other cybersecurity measures to provide comprehensive protection.
* **Network Monitoring**: Continuously monitors network traffic for unusual activities, ensuring constant vigilance.
* **Detailed Alerts**: Provides detailed alerts and logs about suspicious activities, helping IT teams investigate and respond effectively.

**Disadvantages**

* **False Alarms**: IDS can generate false positives, alerting on harmless activities and causing unnecessary concern.
* **Resource Intensive**: It can use a lot of system resources, potentially slowing down network performance.
* **Requires Maintenance**: Regular updates and tuning are needed to keep the IDS effective, which can be time-consuming.
* **Doesn’t Prevent Attacks**: IDS detects and alerts but doesn’t stop attacks, so additional measures are still needed.
* **Complex to Manage**: Setting up and managing an IDS can be complex and may require specialized knowledge.

**Conclusion:**

Intrusion Detection System (IDS) is a powerful tool that can help businesses in detecting and prevent unauthorized access to their network. By analyzing network traffic patterns, IDS can identify any suspicious activities and alert the system administrator. IDS can be a valuable addition to any organization’s security infrastructure, providing insights and improving network performance.

**Program:**

class AlertLog:

 def \_init\_(self, alert\_id, network\_id, description):

    self.alert\_id = alert\_id

    self.network\_id = network\_id

    self.description = description

class AlertLogManager:

    def \_init\_(self):

        self.alerts = {}

        self.next\_alert\_id = 1

    def create\_alert(self, network\_id, description):

        alert\_id = self.next\_alert\_id

        self.alerts[alert\_id] = AlertLog(alert\_id, network\_id, description)

        self.next\_alert\_id += 1

        return alert\_id

    def read\_alert(self, alert\_id):

        return self.alerts.get(alert\_id)

    def update\_alert(self, alert\_id, network\_id=None, description=None):

        if alert\_id in self.alerts:

            if network\_id:

                self.alerts[alert\_id].network\_id = network\_id

            if description:

                self.alerts[alert\_id].description = description

                return True

            return False

    def delete\_alert(self, alert\_id):

        if alert\_id in self.alerts:

            del self.alerts[alert\_id]

            return True

        return False

class IntrusionDetector:

    def \_init\_(self, alert\_manager):

        self.alert\_manager = alert\_manager

    def detect\_network\_intrusions(self, network\_id):

 #Simulate intrusion detection logic

        if network\_id % 2 ==0: #Examplecondition

            alert\_id = self.alert\_manager.create\_alert(network\_id, "Suspiciousactivity detected.")

            return alert\_id

        return None

    def log\_security\_alerts(self, alert\_id):

        alert = self.alert\_manager.read\_alert(alert\_id)

        if alert:

            print(f"Alert ID {alert.alert\_id}: {alert.description} on Network{alert.network\_id}")

        else:

            print("Alert not found.")

 ###2.Unit Testing using unittest

import unittest

class TestAlertLogManager(unittest.TestCase):

    def setUp(self):

        self.manager = AlertLogManager()

    def test\_alert\_crud\_operations(self):

 #Test Create

        alert\_id1 = self.manager.create\_alert(1, "Test Alert 1")

        self.assertIsNotNone(self.manager.read\_alert(alert\_id1))

#Test Read

        alert = self.manager.read\_alert(alert\_id1)

        self.assertEqual(alert.description, "Test Alert 1")

 #Test Update

        self.assertTrue(self.manager.update\_alert(alert\_id1,description="Updated Alert 1"))

        self.assertEqual(self.manager.read\_alert(alert\_id1).description,"Updated Alert 1")

 #Test Delete

        self.assertTrue(self.manager.delete\_alert(alert\_id1))

        self.assertIsNone(self.manager.read\_alert(alert\_id1))

class TestIntrusionDetector(unittest.TestCase):

    def setUp(self):

        self.manager = AlertLogManager()

        self.detector = IntrusionDetector(self.manager)

    def test\_detection\_and\_logging(self):

        network\_id = 2 #Shouldtrigger an alert

        alert\_id = self.detector.detect\_network\_intrusions(network\_id)

        self.assertIsNotNone(alert\_id)

        self.assertIsNotNone(self.manager.read\_alert(alert\_id))

        self.detector.log\_security\_alerts(alert\_id)

if \_\_name\_\_ == "\_main\_":

    unittest.main()

**Algorithm:**

1. **Class Definitions:**
   * Define AlertLog class:
     + Initialize with alert\_id, network\_id, description.
   * Define AlertLogManager class:
     + Initialize with an empty dictionary alerts and next\_alert\_id set to 1.
     + Methods:
       - create\_alert(network\_id, description):
         * Create an alert, assign an alert\_id, store it, increment next\_alert\_id.
       - read\_alert(alert\_id):
         * Return the alert with the given alert\_id.
       - update\_alert(alert\_id, network\_id, description):
         * Update network\_id and/or description if the alert exists.
       - delete\_alert(alert\_id):
         * Delete the alert if it exists.
   * Define IntrusionDetector class:
     + Initialize with alert\_manager.
     + Methods:
       - detect\_network\_intrusions(network\_id):
         * Simulate intrusion detection (example: if network\_id is even, create an alert).
       - log\_security\_alerts(alert\_id):
         * Log details of the alert by its alert\_id.
2. **Unit Testing:**
   * Create TestAlertLogManager class:
     + Test CRUD operations for alerts.
   * Create TestIntrusionDetector class:
     + Test detection and logging of alerts.
3. **Execution:**
   * Run unit tests.

**Flow Chart:**

[Start]

|

v

[Define AlertLog Class]

|

v

[Define AlertLogManager Class]

|

v

[Define IntrusionDetector Class]