National College of Ireland

# Novel Technique for Detecting Unknown Threats Using Honeynet Instead of Purple Teaming in Organizations

MSc Research Project
MSc Cyber Security

Jithin Paul John
Student ID: x20254857

School of Computing
National College of Ireland

Supervisor: Imran Khan

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | | | |
|---|---|---|---|
| **Student Name:** | Jithin Paul John | | |
| **Student ID:** | x20254857 | | |
| **Programme:** | MSc Cyber Security | **Year:** | 2022-2023 |
| **Module:** | MSc Research Project | | |
| **Supervisor:** | Imran Khan | | |
| **Submission Due Date:** | 15-12-2022 | | |
| **Project Title:** | Novel technique for detecting unknown threats using honeypot instead of purple teaming in organization | | |
| **Word Count:** | 6870 | **Page Count:** | 21 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Jithin Paul John

**Date:** 15-12-2022

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

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

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

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

# Novel Technique for Detecting Unknown Threats Using Honeynet Instead of Purple Teaming in Organizations

Jithin Paul John
x20254857

**Abstract**

Unknown threats are one of the key problems facing the contemporary cybersecurity industry. New dangers and attacks appear every day, along with the development of extremely complex techniques and instruments used to employ them. It is getting harder to identify and stop these assaults because of how sophisticated they are. Organizations currently use purple teaming exercises to identify and defend against novel cyber threats using the knowledge of the red team and blue team. Setting up rules in IDS will only recognize attacks that are behaviour-based and signature-based due to limitations with the engineers' skills. Manual, irregular testing is insufficient to ensure that control gaps don't appear, sometimes undetected for days. To prevent attackers from exploiting loopholes, continuous testing and validation are required, but they are nearly unachievable without automation. To address these issues, this research suggests a novel detection technique that makes use of a web application honeypot or honey network. The construction of web-based honeypots is straightforward and devoid of complex architectural designs, and they create noticeably fewer false positives than traditional systems.

## 1 Introduction

Data breaches have happened to most organizations, and the number of victims is rising. Despite the size of the company, everyone is now a target. The primary reason is due to excessive dependence on security solutions, which are not useful against certain types of attacks. In traditional security, there are lists of known threats that may be appropriately prevented. Unfortunately, there will always be vulnerabilities that the system is blind to. The main area where most security systems fall short is in responding to unforeseen attacks. Too many threats were used over time. Malware creators started employing minute changes to make them resistant to detection systems. The deadliest dangers are those that evolve from already-existing ones.

The problem is that not all risks, that can be identified online are recognized or even acknowledged to exist. Because of the obfuscating nature of these threats antivirus software, network filters, intrusion detection/prevention systems, and other conventional defensive

methods cannot detect or block many fast-changing exploits and spread on the internet. Most unidentified threats that lurk online cannot be protected against by traditional decision-based security solutions, which rely on detecting the appearance and behaviour of malware your system is likely to encounter.

Unknown threats are defined as exploits that use novel techniques to take advantage of security holes. There is an inherent challenge with any threat detection methods that rely on statistics and signatures. Although these tactics are effective against known security risks, it has been determined that because traditional approaches rely on databases of known threats, they have relatively little ability to defend against changes in attack tactics. With unidentified attacks, the adversary can locate security loopholes, defects in the source code of the software, or network vulnerabilities and produce destructive exploits for a cyber-attack by exploiting the weakness.

By integrating members of the red team and blue team, the business launched a purple teaming exercise to detect and neutralize new threats. Purple teams, the proactive attitude they engender, and the personnel already on staff are crucial for spotting issues with an organization's security arrangement. By collaborating with the red and blue teams during the purple teaming exercises, organizations may completely comprehend the tactics, methods, and procedures (TTPs) of threat actors. By duplicating these TTPs in various red team scenarios, the blue team may set up, adjust, and improve its detection and reaction capabilities.

Monitoring occurrences in organizations IT environment and spotting actual security issues is known as threat detection. The ability to identify threats in real time is necessary for both detection and prevention, which are independent. The lengthy purple teaming activities involve a team effort to comprehend and describe a rule that must be implemented in the detection and prevention system. Manual, irregular testing is insufficient to ensure that control gaps don't appear, sometimes undetected for days. To stop attackers from taking advantage of the gaps, continuous control testing and validation are necessary, but this is very hard to do without automation.

The suggested detection techniques employ a honeypot with several services that gathers all incoming packets and sends them to a packet analyzer for analysis with the help of a customized IDS. This will aid in the detection of all incoming packets and the comprehension of any new strategies and tactics employed by adversaries. This approach will be more successful in gathering all the data while requiring fewer individuals to participate. Attackers are made to think they have access to the original software or network through deception. All incoming traffic, not only attacks with behavioral or signature-based features, is captured by this technique. Even the non-malicious packets are saved for further analysis and study.

## 1.1 Research Question

These above-mentioned issues and the need for an automated system inspire the research question: **How can organizations use honey networks along with customized IDS as an automated system for the detection and analysis of unknown threats instead of using purple teaming exercises?**

# 2 Related Work

This section discusses earlier research that is relevant to the approach proposed in this study in terms of discovering unknown threats. For a deeper understanding, investigations on honeypots and other methods of detection have also been studied. A wide range of research has been carried out, with a focus on using machine learning, and deep learning to discover unknown threats. The following is a list of research that use a honeypot or unknown threat detection system as a detection instrument.

## 2.1 Unknown/Zero-day threat Literature

The article by (Kumar and Sinha, 2021) is an effective method of detecting zero-day cyberattacks with two parts, namely the development of signatures and the assessment phase, and explores the idea of the "heavy-hitter" and the graph methodology. Using heavy-hitter and graph-based methodologies, they were able to identify attack vectors, although the precise category of attack variations was not identified.

(Radhakrishnan, Menon, and Nath, 2019) study on examining the detection and analysis techniques already in use while designing systems that can identify the most recent threats. The main objective of the project is to create new methods or improve the detection and analyzing malware. This paper's primary drawback was its exclusive focus on malware.

In a different work (Lobato et al., 2018), stream processing technology is used to process data quickly. They utilize honeypots to collect information and test different detection methods. The suggested architecture combines both unsupervised anomaly detection and online recognized threat categorization to further secure the network. Threats can be identified using the suggested approach, but signature-based IDS systems won't pick these intrusions up.

Another research by (Abri et al. 2019) investigates several machine learning techniques that may be used to identify hazards from zero-day malware. The authors of this work examined 34 machine-learning classifiers and found that forest classifiers had the highest accuracy. To have a deeper insight, the test must run with more data sets.

The work by (Innab, Alomairy, and Alsheddi's 2018) describe a novel technique using anomaly-based detection and honeypots to discover zero-day attacks. The two main methodologies of anomaly-based detection and honeypot to thwart the Zero Day assault are then compared for advantages and disadvantages. The results of this article showed higher false positive rates for anomaly-based detection and a constrained field of view for the honeypot.

Firstly, the obtained data sets are examined. Secondly, the malware links were connected using a correlation approach, allowing us to predict future malware kinds. Lastly, train them and employ malware detection methods to enable them to recognize malware and deliver the intended outcomes. (Patidar and Khandelwal, 2018) proposed utilizing machine learning approaches to identify zero-day attacks and found that it protects both organizational networks

and personal data.In the investigation made by (Blaise et al., 2020) the anomaly detection approach was proposed for botnets that use a single port. Here, the authors track the port change detection technique to spot the early appearance of botnets, recently found vulnerabilities, and other forms of attacks within a certain network.

In a different publication (Tang et al., 2020), the topic of zero-day attack detection is covered. This paper proposes ZeroWall, an unsupervised approach that may work with an already-installed WAF. The authors suggest a comprehensive architecture for supplementing current signature-based WAFs with unsupervised machine learning-based zero-day Web detection techniques. The disadvantage of this strategy is that performance suffers when there is insufficient data.

In their study (Ullah et al., 2019) employ deep learning to identify undiscovered cyber threats in IoT devices. The deep neural network is also designed to visualize binary data to recognize risky virus patterns. The research focuses mostly on malware distribution via pirated software and provides a detection approach for families of unknown malware in IoT clouds and environments.

(Shaukat et al., 2020) examined the capacities of three learning models to recognize and classify malware, spam, and incursion in a performance evaluation of various detection approaches. Using commonly used and benchmark datasets, they evaluated the evaluation results in terms of recall, accuracy, and consistency. However, a substantial number of authors have tried to highlight the drawbacks that machine learning approaches entail.

The article by (Zhang et al., 2021) addresses several detection techniques that may be applied to find undiscovered threats. It begins with intrusion detection strategies before concentrating mostly on deep learning approaches. Deep learning techniques, according to the authors, are better at identifying unidentified dangers. It hasn't been tried out or put into practice, though.

The study conducted by (Farooq and Otaibi, 2018) on many best machine learning algorithms for cyber threat identification in another publication. Analytics powered by machine learning (ML) is a great way to offer context after analyzing the normal behavioural baselines of security occurrences and producing a few false-positive security alarms. As stated by the authors, the approach must be evaluated to determine its efficacy.

## 2.2 Honeypots Literature

A study of honeypots and honeynets for the identification of unidentified threats to the industrial internet of things was completed by (Franco et al., 2021). Based on the level of interaction, flexibility, accessibility of the source code, and the anticipated IoT, IIoT, or CPS application, they offered a categorization of honeypots and honeynets. Additionally, it helps in understanding various honeypots and their properties.

The research article by (Naik and Jenkins, 2018) proposes a resource-optimized fuzzy low-interaction honeypot-based approach to detect and counter a spoofing attack. It provides a

detection strategy for the spoofing attack based on the analysis and data gathered from the honeypot. To prevent a spoofing attack, the study then offers a hazy approach for foreseeing it and giving quick warning.

A novel approach to employing a honeypot to protect resources and data in the cloud that is implemented through an application was presented by (Negi, Garg, and Lal, 2020). A honeypot frequently acts as a monitoring tool and, when required, can provide a warning. The application records the user's IP address because the functioning of a Honeypot requires quiet detection, allowing the administrator to subsequently evaluate it and locate the offending party.

The Kfsensor and HoneyD honeypots are investigated in the research study by (Bhagat and Arora, 2018). The foundational network services have undergone testing. In HoneyD, many suspicious IP addresses have been discovered. Kfsensor, on the other hand, just shows the IP of the network where it is mounted.

A system that employs a honeypot intrusion detection and prevention system (IDPS) and can animate in real-time to display network activity on servers was developed in the study by (Sethia and Jeyasekar, 2019). The high rate of false positives, which is one of the most significant shortcomings of anomaly-based IDS and is reduced by this developed approach.

A thorough investigation by (Tan et al., 2022) for a safe and reliable AIoT, threat detection, and situational awareness based on honeynets is recommended. For this strategy, a Docker-based honeynet to collect a particular quantity of attack log data using a deep learning approach was built. The findings only applied to an AIoT context, though. Despite the authors' claims to the contrary, the presented approach is very scalable.

## 2.3 Research Niche

Organizational security is strongly tied to the paradigm that is given in this study. Organizations are the main targets of unknown attacks since data theft is their main goal. Consequently, the suggested approach is quite significant in the actual world since it aids businesses in identifying fresh dangers and securing their networks. Details of the cited works are displayed in the table below.

| **Authors and year** | **Strength** | **Weakness** |
|---|---|---|
| Bhagat, N. and Arora, B. (2018) | Traffic was recorded and compared between two honeypots. | One honeypot only recorded internal network traffic, which was not what was intended to be done. |
| Abri, F. *et al.* (2019) | 99% accurate comparison of well-known traditional machine learning and deep learning algorithms | A few of the zero-day malware can sometimes be misclassified by classifiers. |

| | | |
|---|---|---|
| Sethia, V. and Jeyasekar, A. (2019) | Malware that primarily targets the SQL server 2000 for XP, SMB, and FTP protocol is caught, along with one undiscovered malware. | Only low interaction honeypots were able to handle suggested technique. |
| Ullah, F. *et al.* (2019) | Malware files has been converted to colour images for better visualization | Only detects known risks. Using this technique, unknown hazards are avoided. |
| Blaise, A. *et al.* (2020) | Ability to identify known and unidentified threats aimed at servers and related items | Detecting botnets was the proposed model's only use. |
| Tang, R. *et al.* (2020) | ZeroWall greatly outperforms previous methods with high F1-scores > 0.98. | Performance is penalized by having too little data. |
| Franco, J. *et al.* (2021) | 79 innovative honeypots and honeynets for the Internet of Things were studied to extract their common traits. | The study was unsuccessful because just a few honeypots and honeynets used ML methods. |

**Table 1: Strengths and weaknesses of related works**

# 3 Research Methodology

Any firm can use the study extensively to safeguard its network. This detection technique will act as a backup detection strategy akin to the purple team training. The major goal of the suggested strategy is to identify unknown threats by gathering all incoming network requests and classifying them as malicious or not. To do this, a packet analyzer will be used. The detection and prevention system's signature can be modified once the threat has been located Below are the major objectives of the proposed model.

1. Building a honeynet to draw in adversaries for study.
2. Detecting malicious traffic using both signature-based and heuristics-based detection techniques.
3. The discovery of new threats and the addition of their signatures to a unique user-defined repository.
4. Identifying anonymous attackers using the application.
5. Using fingerprinting, identify protocols and servers.
6. Data on attack statistics from a honeypot.
7. Dashboard that shows the country from where the attack originated.
8. Logging benign traffic separately in a database so that further in-depth analysis is possible.

In this project, we use a honeynet as a decoy network with one or more honeypots to lure threat actors into various levels of damaging conduct. A honeynet is a framework that allows you to create and maintain your own analytics repository, import data from other projects, and produce data into two dashboards with comprehensive reports that can be read by humans. Data

gathering, security system upkeep, threat detection, incident response, and other responsibilities are all components of a larger, more extensive process that necessitates strong departmental interaction. The research model's technique is described below.

## 3.1 Malicious traffic detection

Malicious traffic detection is capable of spotting communication between endpoint devices and the command-and-control servers that are employed in botnet or other malware assaults. The essential element of any information is maintaining the CIA (Confidentiality, Integrity, Availability). A cyber-intrusion is any action that aims to undermine the CIA or circumvent its cybersecurity safeguards. Security professionals frequently utilize IDS (Intrusion Detection Systems) to find security risks. It is a tool or piece of software that searches a system or network for unlawful activities or rules that have been breached. For centralized data gathering or to alert an administrator of any incursion or violation, security information and event management systems are frequently used. IDS generally examine each packet entering and exiting a specific network to look for indicators of intrusion. The characteristics of intrusions may be recognized by a competent IDS and its auxiliary features, which may then automatically react by producing security logs or issuing warnings in response. The methods below are what we use to spot malicious traffic.

### 3.1.1 Commonly suspicious trail types

- Categorical network traffic analytics.
- Analysis of passive DNS to identify malicious domains
- HTTP fingerprinting
- Tools for automatic testing
- Fingerprinting application using headers
- Monitoring DNS tunneling
- Identification of newly registered domains
- Tracking IP addresses and suspicious domains

### 3.1.2 Signature-based detection method

"Signature-based detection" creates a recognizable signature for a known danger so that it may be recognized in the future. A virus scanner could detect the hash of a known harmful file or a particular pattern of the script that is attached to a file, for example. For the signature-based detection technique, we follow the procedures below.

- Compiling static trails from many antiviruses scan reports
- User-defined custom list

### 3.1.3 Heuristic-based detection method

Instead of determining if a device is infected with malware by consulting a list of defects, heuristic detection finds files with malicious behaviour or code structure and flags them as possibly dangerous. It's carried out through dynamic and static analysis methods.

### 3.2 Non-Malicious traffic detection

It provides a summary of undiscovered traffic. For non-malicious traffic, we maintain a separate database that may be utilized for additional analysis. For real-time monitoring of undiscovered packets, a dedicated dashboard will be provided. The packets that are gathered will include the information below.

- Arrival time
- Protocol, source IP, destination IP
- Source and destination ports and fingerprint
- Data payload

## 4 Design Specification

Prevention, detection, and reaction are the three operational categories that may be used to group security principles. Preventive action is any action that stops threats and secures a system. Finding behaviours that put the systems' confidentiality, integrity, and availability at risk is the process of detection. When harmful habits are discovered, the employment of taking immediate action is referred to as a "reaction." In an ideal scenario, responding strengthens prevention while enhancing subsequent detections. Because security concepts are developed to function well in their specific application, they perform poorly in other areas. The parts of the suggested model are shown in the diagram below.

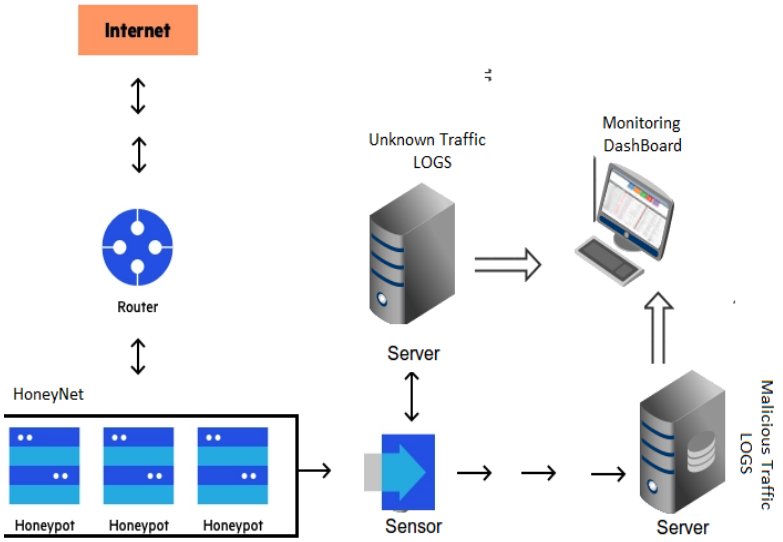


**Figure 1 Architecture of the proposed model**

## 4.1 Honeynet/Honeypot

An artificial system called a "honeypot" is placed at the network's entry to capture hackers. A honeypot serves as a company's primary network security tool. It serves as a means of diverting the attacker from the main server attack. To allow a packet to or from a local area network, the firewall often uses several laws. Therefore, incoming packets are filtered using the firewall and honeypot. Honeypot replicates a range of services based on how much contact it receives. (*What Is a Honeypot? Meaning, Types, Benefits, and More*, no date)

A honeypot's ability to operate is influenced by the level of touch it receives. They imitate various system operations and convey the idea that the system is ineffective. There are several kinds of honeypots, including pure honeypots, and honeypots with high and low levels of interaction. High-interaction honeypots imitate many services at once. On the other hand, are more expensive to deploy and maintain. Low-interaction honeypots are intended to imitate just one service. Low-interaction honeypots have several advantages over higher-interaction honeypots, but their main advantage is that they consume fewer resources and are thus easier to set up.

The three elements data control, data capture, and data collection must be included in each honeynet. To stop attackers from discovering they are within a honeynet and to make sure that the honeynet won't be utilized to attack other systems once it has been infiltrated, data control comprises controlling the flow of data. They can be installed in private deployment environments with public IP addresses, cloud computing environments, real application/production settings, and Demilitarized Zones (DMZs) within business networks. These deployment options each have advantages and disadvantages of their own. Every step and activity performed inside the honeypot is tracked as part of the data collection process. Data collection requires the capacity to securely send all the collected data to a single location. Each honeypot has a network scanner and an IP tracker.

Here we have used a Dionaea as the honeypot. It is a malware-capturing honeypot. Dionaea intends to capture malware that makes use of flaws revealed by services provided through a network that eventually gets a copy of the infection. Dionaea has a modular design and integrates Python as its scripting language to mimic protocols. It is far superior and supports IPv6 and TLS and can detect shellcodes using LibEmu. Dionaea probably has exploitable defects, just like any other piece of software. Dionaea operates without administrative rights in a confined context to lessen the effects.

## 4.2 Sensor

The sensor, a part of the monitoring nodes, is a standalone component. The independent system receives traffic from several honeypots and checks it for the blacklisted packet. If a match is found, the event data is sent to the server and saved in the relevant logging database. The sensor contains several parts that the suggested model can employ for efficient detection. They are,

- Multiprocessing: The CPU will be utilized completely if this option is chosen. To process packets, extra cores will be necessary for addition to the one core utilized for packet capture (with proper affinity, IO priority, and good level settings). A single core will be utilized for all purposes else.
- Update Period: The number of seconds between each recurring update of the footprints is set to one day using specifications from the trail's directory.
- Custom trails: This can be used by the user to provide the location of the directory containing the special log files for the recently discovered directory of logs.
- Use Heuristics: This will initiate heuristic techniques that may produce false positives, including questionable domain names, downloads, lengthy domain names, and so forth.
- Capture filter: By excluding the irrelevant packets with the network capture (tcpdump) filter, the data collection process may be made simpler. As a result, two distinct categories of occurrences may be distinguished.

## 4.3 Server

In this design, the server's major duties are to store event data and provide backend assistance for the dashboard.

- Log Server: This enables the development of fresh signatures for attacks that haven't yet been recognized, which may subsequently be applied to a particular list. All identified packets will be saved immediately in the logging directory if this option is not used.
- Logstash Server: The event data will be transmitted in JSON format to the specified location via a UDP service (like Logstash).
- Syslog Server: A UDP (for instance, Syslog) service listening at the specified IP will receive the event data in CEF (Common Event Format) format.

## 4.4 Custom IDS

An IDS may be used to assess the amount and diversity of attacks. Organizations might adjust their security policies or implement better security measures using this knowledge. An intrusion detection system can be used by organizations to identify errors or problems with the setup of their network equipment. Here, IDS examines the current signatures and discards packets from adversaries.

A custom IDS has been used as a network intrusion detection system rather than a predesigned application. Custom IDS have been designed with various rule sets which will be followed when a new packet reaches the IDS. The IDS will initially check the existing databases for signatures, if it matches the packet will be dropped. If not, it will proceed with a heuristic-based check. Any packets that bypass the signature-based and heuristic-based analysis will be a new threat and it will be logged as a new threat.

Several entry files such as suspicious domains, rouge DNS details, well-known botnets, crypto mining bot details, anonymous web proxy details, malware details, onion (TOR project) details, and DNS sinkhole concepts have been defined with the data collected from open-source resources. The TOR concept will work in a similar way to how google blocks anonymous users. DNS sinkhole is a technique that checks DNS queries against a live list of known harmful websites. By adding the fictitious entry to the DNS, it may also be used to redirect traffic to malicious URLs. Additionally, it is a technique for rerouting fraudulent internet traffic so that it may be recorded and examined. The whitelisted and blacklisted entry files help IDS to check whether the packets are malicious or not.

```
        if ".onion." in query:
            trail = re.sub(r"(\.onion)(\..*)", r"\1(\2)", query)
            _ = trail.split('(')[0]
            if _ in trails:
                result = True
                flag = 1
                log_event((sec, usec, src_ip, src_port, dst_ip, dst_port, proto,
                           TRAIL.DNS, trail, trails[_][0], trails[_][1]), packet)
```

```
# Reference: http://malwaretips.com/threads/connects-to-tor-hidden-services-through-tor2web.42274/
# Reference: http://darkspider.info/index.php?menu=list

onion.gq
onion.lt
onion.cab
onion.city
onion.direct
onion.link
onion.nu
tor2web.fi
tor2web.blutmagie.de
tor2web.org
tor2web.ru
tor-gateways.de
```

**Figure 2 Tor concept and entry file**

## 4.5 Dashboards

Dashboards are a must for any reliable detection system. The system gathers log information from several sources, normalizes it, and then prepares the information for analysis. Dashboards are used to present the analytical findings as insightful data. The dashboard enables the security team to assess analytical advice and investment alternatives graphically in one location. Dashboards are utilized in this example of the suggested model to obtain the details below.

- An information tooltip will popup when the mouse is moved over one of these symbols, showing all items possessed and all port numbers being scanned by an adversary.
- Source port, destination port, protocol, and other event data are displayed.

- When the mouse pointer remains over the threat's route for a few seconds, a frame showing the results of a query will appear.
- The threat column contains the threat's particular ID.

# 5 Implementation

The project has been implemented over an AWS EC2 instance. It helps to create and deploy apps more quickly using Amazon EC2 since there is no longer a requirement to make an upfront hardware investment. Launch as many or as few virtual servers as you require, set up networking and security settings, and control storage using Amazon EC2. An Ubuntu 22.0 operating system has been spun up initially. Python 3.8 has been installed as it's a mandatory requirement for Honeypot and custom IDS which are written in python.

In the next step, Dionaea honeypot has been implemented. Dionaea is dedicated to locating the malware. Dionaea does this by simulating a variety of network-exposed services, including SMB, HTTP, FTP, TFTP, MySQL, MSSQL, VoIP, and others. Although we could set Dionaea to listen to any protocol in the list, we will only select the ones that are most likely to result in accidents since we want our honeypot to appear realistic. Once the cloning is completed, we need to install the dependencies and build the directory. After this, we will configure ihandlers where the honeypot handles the traffic, and services, and the main config file where we remove unwanted services and configure Dionaea as a service. As an additional feature, the virus total has been configured with Dionaea to improve the performance. This will help to upload the captured binaries to the virus total, to get both help from the community and to get an AV scan of the captured packets automatically.

After implementing the honeypot, custom IDS has been installed. It is a python-based intrusion detection system that has a sensor part and a server side. Custom IDS is a major part of this project as it is configured in a way that can filter packets as malicious or not by analyzing the content. It uses the entry files associated with the sensor.py and segregates the packets considering the source IP, packet type, tools used, etc. The analyzed packets will be differentiated as unknown traffic or malicious based on the packet. The entry files are classified as below based on which the IDS distinguishes the packets.

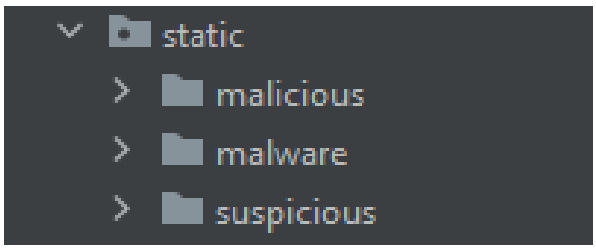


**Figure 3 Entry file classification**

Pcapy is the packet analyzer used in the custom IDS. It is a very adaptable application that may be used for a variety of purposes. In the most basic scenario, it may just be a collection of simple packet blocks that represent a raw capture of the network data.

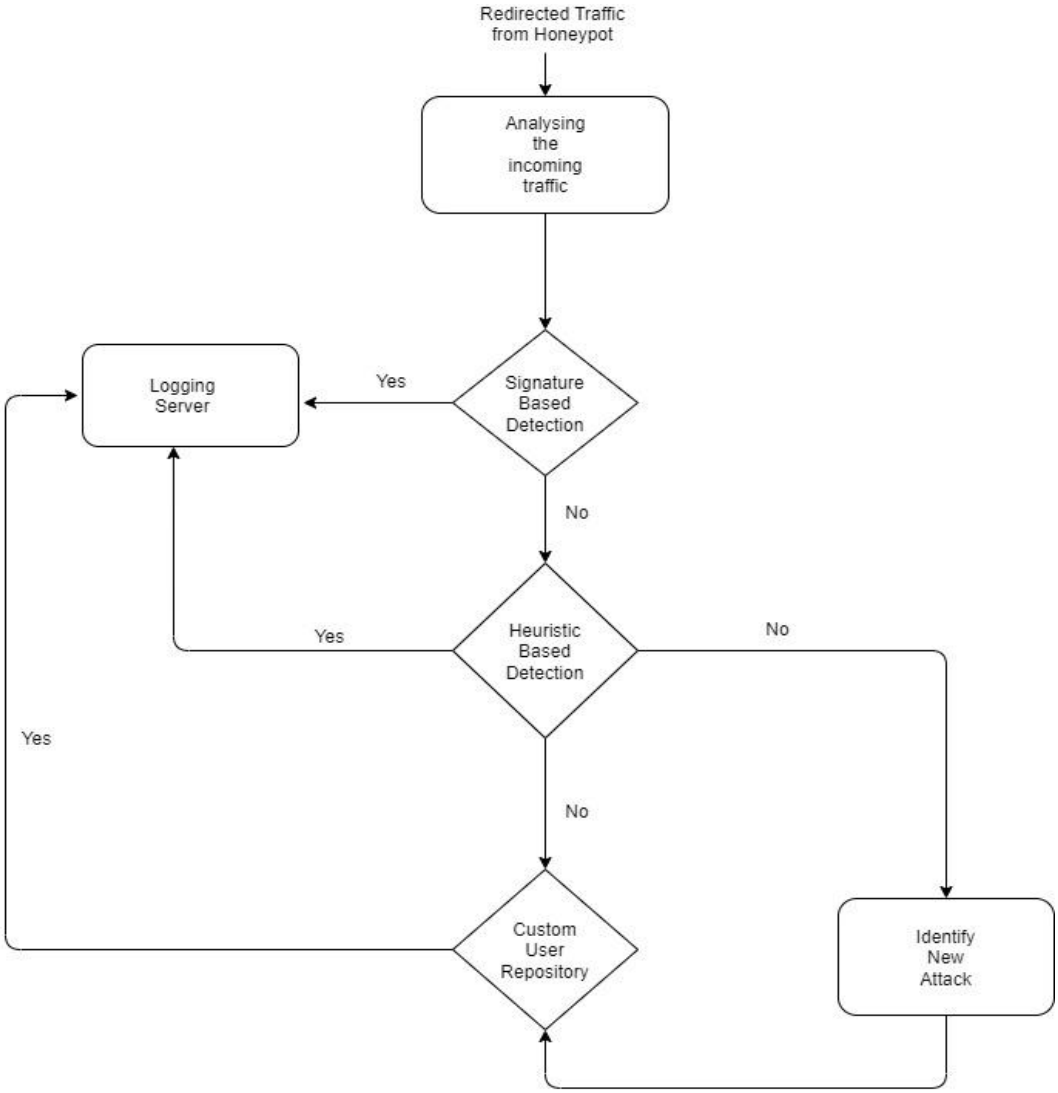


**Figure 4 Data flow diagram – Custom IDS**

The data collection and analysis process are depicted using the suggested model in the diagram above. Data captured by the honeypot will be transferred to IDS. While the custom-developed IDS is switched on, it will fetch data regarding vulnerable IPs, and other malicious data from various Talos websites and other databases. These data are used to identify whether the incoming packets and IPs are malicious, suspicious, or not.

Initially, with the help of the current signature-based detection techniques, the data will be examined. By using the predefined repository of static signatures fetched earlier, the signature-based detection system uses a deterministic pattern that helps in the easy identification of malicious domains or bytes found in the packet header. Once a new packet comes, the system starts comparing them to the known signatures in its repository. It continues to track network traffic and look for signature matches. If a match is discovered, the file is labeled as a "threat" and prevented from acting further.

The arriving packets will be checked using heuristic-based detection methods if the signature-based detection system fails to discover any signature for them with the current signature. This method mainly uses a file analysis technique to identify whether the packets are malicious or not by looking for specific strings or keywords. The packet will be sent to the packet analyser for additional investigation if the detection and prevention systems did not discover any malicious data and it will be logged. The packet will be carefully examined, and data will be recorded as a new threat if any dangerous functionality is discovered.

Other than this, an anomaly-based detection method is also used to analyze packet headers and differentiate them. Typical aspects of anomaly-based detection are used here such as comparing communication intervals for normal communication while using tools, finding client-side application banners, analyzing the response request header while protocol communicates, and checking the data payload size. These data will serve as a baseline for the detection and helps in identifying the tool used by the attacker. Along with the custom IDS, we install two different dashboards. One is for malicious traffic and the other is for normal, unknown traffic. Both dashboards show the details with the date, time of the attack, source details, destination detail, type of attack, and other information. This will help any organization distinguish the nature of packets and attack patterns.

# 6 Evaluation

Honeypot and custom IDS have been implemented and tested by simulating several attacks using Kali Linux. The below screenshots of the dashboards show details of packet capture.

## 6.1 Experiment 1

Dirbuster is a multi-threaded java application that has been used to brute force directories and file names on web/application servers. The below figure shows the attack executed and the same has been captured by custom IDS and after analysis, it has been tagged as a suspicious attack.

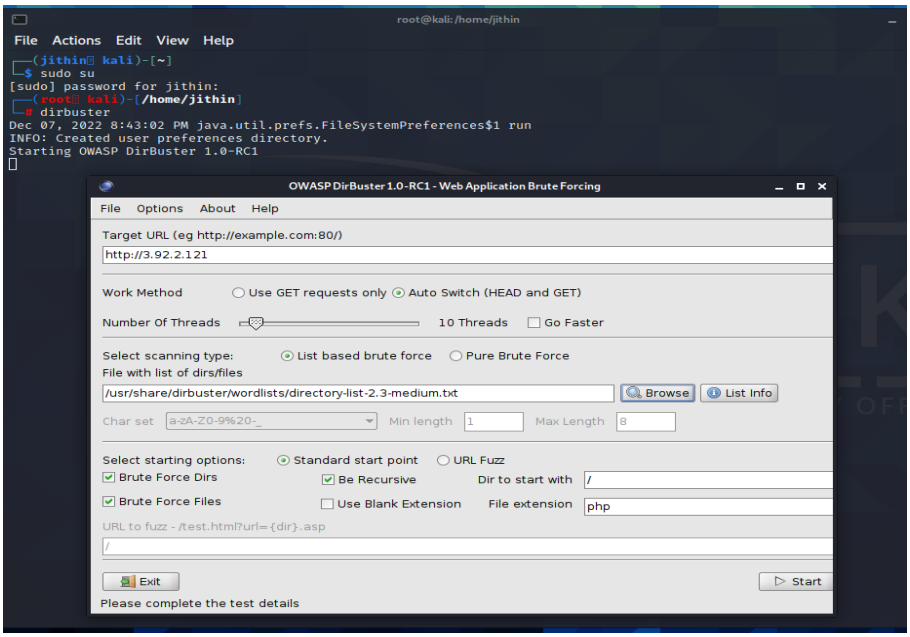


**Figure 5 Attack simulation using Dirbuster**

## 6.2 Experiment 2

Metasploit's dynamic scripting tool gives users a lot of options. It has been used to compromise a system utilizing shell-based access or meterpreter. By switching payloads based on the open ports and services imitated by the Honeypot, Metasploit made testing simple. The same has been captured by IDS and marked as a suspicious attack in the dashboard.

Figure 5 shows the dashboard for known attacks and patterns. Here we can see the various packets captured by IDS and they are marked as suspicious along with the geo-location of the source IP. As the IP of the project setup was publicly available, IDS captured malicious attacks that came from a known attackers from various parts of the world. The dashboard shows mass scanning performed by attackers using shodan.io, and by scanners hosted in the cloud like digitalocean, latisys, cari.net, etc. As checked, the source IPs are marked as malicious in Virus Total and other databases.

| Dst_port | Protocol | Type | Deduce | Info |
|---|---|---|---|---|
| 80 (http) | TCP | UA | DirBuster | user agent (suspicious) |
| 80 (http) | TCP | URL | /gate.php amazon | php script (malicious) |
| 80 (http) | TCP | URL | 3.85.135.152 amazon | potential web shell (suspicious) |
| 10443 | TCP | IP | 179.43.177.154 | known attacker |
| 80 (http) | TCP | URL | /dd.php amazon | php script (malicious) |
| 80 (http) | TCP | URL | /backdoor.php amazon | php script (malicious) |

**Figure 6 Suspicious/Malicious attacks - Dashboard**

The below figure shows the non-malicious/non-suspicious traffic. This traffic can be used for further studies.

**Figure 7 Unknow/Non-malicious traffic – Dashboard**

# 7 Discussion

The proposed model is a detection method that can be used instead of a purple teaming exercise. The project has been designed as an automated system that captures all the packets coming in and detects unknown threats or suspicious packets. The system works based on the attack and scanning policies defined in the custom IDS such as malicious domains, attack patterns, malware signatures, etc. When a packet arrives, it checks various open-source databases, whether the data in the packet has any existing signature. This is being checked in an open-source database such as Virus Total. It also compares the source IP address and other data in the packet with the help of pcapy by comparing the data defined in the entry files. As an alternative, data collected by honeypot is uploaded to Virus Total to have another check with the signatures and details in their database.

Creating a traffic regulation policy can be implemented as an improvement in the project. As the proposed system captures all the packets coming in, there are chances of traffic congestion. So, establishing one or more intrusion detection traffic control rules from all policies sections or the traffic regulation policy's view to keep an eye out for heavy network traffic that can possibly overflow your system. It helps to keep an eye out for System SSL/TLS, TCP, or UDP

connections. Also, if the system can show the signature details of the known attacks, malicious files, or packets in the dashboard, it will be helpful for the analyst to research more about the threat.

The implementation has been successful, and the custom IDS was able to analyze and segregate the packets as malicious, suspicious, or non-malicious. The attacks from existing signatures were identified and the packet was dropped. The model also shows the geo-location from where the attack came. The research produced good accuracy for each kind of attack executed. To check the performance of the suggested model, accuracy has been calculated. The accuracy is calculated using the below equation.

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$

- True Positive (TP): The number of packets precisely identified as an attack.
- True Negative (TN): The number of packets correctly classified as normal.
- False negative (FN): The number of attack packets misclassified as normal.
- False Positive (FP): The wrong classification of normal packets as an attack.

Accuracy has been calculated by dividing the number of accurate predictions by the total number of predictions to determine the proposed model's accuracy. The total packets received are classified into nine categories and accuracy has been classified in each of them.

Below table and diagram show the accuracy level of the system with various types of packets captured after simulating different types of scans, requests, and attacks. Suspicious IP requests and port scanning has 98% of accuracy. However, the rate of false positives may be high as the system tag usual packets as suspicious based on behaviour. Suspicious HTTP requests were detected with an accuracy of 97% and anonymous attackers and malware attacks show an accuracy of 94%.

**Table 2 Accuracy table for various packets captured**

| Test Case | Attack Type | Accuracy (%) |
|---|---|---|
| 1 | Mass scans | 95 |
| 2 | Anonymous attackers | 94 |
| 3 | Service attackers | 95 |
| 4 | Malware | 94 |
| 5 | Suspicious domain lookups | 96 |
| 6 | Suspicious IP request | 98 |
| 7 | Suspicious HTTP requests | 97 |
| 8 | Port scanning | 98 |
| 9 | False Positives | 98 |

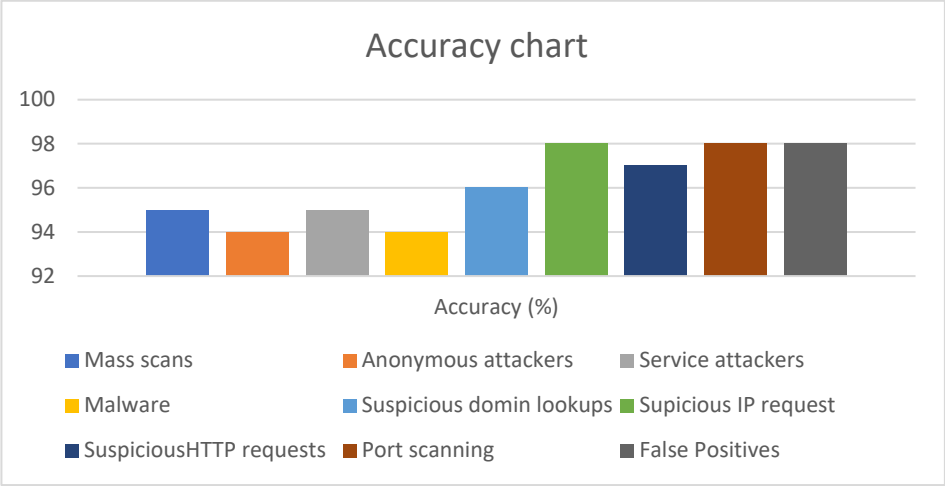


**Figure 8 Diagrammatic representation on the accuracy level**

Most of the previous studies used machine learning, deep learning, and neural network-based concepts in the intrusion detection system. The amount of network data, the level of granularity necessary to increase efficacy and accuracy, and lastly the variety of protocols and data traveling are the three key constraints of those studies. Some of the studies were focused only on certain traffic/attacks such as DoS, probing, U2R, R2L, etc. Here in this project, we were able to capture all the packets and classify them into different categories as suspicious, malicious, and non-malicious accurately. As well as Dionaea honeypot can mimic fourteen protocols that are widely in use. Thus, the proposed model can be used to analyze and learn a wide variety of threats and attack patterns.

# 8 Conclusion and Future Work

The paper suggests a new detection method for unknown threats which can be followed by any organization instead of using a long purple teaming exercise. The concept uses a honeypot with several services running on it and a custom IDS that is defined with several policies and entry files, with the data collected from open-source databases. The project was able to capture packets and classify them after checking the content. The packets that bypass signature-based and heuristic-based detection will always be a new threat that uses new tactics, techniques, and procedures (TTPs). So, this will help organizations to learn more about those threats and implement new defensive methods to keep their data safe by reducing purple teaming exercises to spot unexpected threats.

The suggested model can be linked to a malware team in the future to further investigate emerging threats that lack a signature and solve the developing issue of malware attacks. By downloading a copy of the attachment or by analyzing the packet deeply, the malware team can avoid major damage. Additionally, by employing a good dataset for improved results, artificial intelligence may be used in conjunction with the customized IDS to enhance precision by reducing false positives and thereby increasing accuracy.

# References


Abri, F. *et al.* (2019) 'Can Machine/Deep Learning Classifiers Detect Zero-Day Malware with High Accuracy?', in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 3252–3259. Available at: https://doi.org/10.1109/BigData47090.2019.9006514.

*Be careful - there are unknown threats out there! | TrustPort* (no date). Available at: https://www.trustport.com/en/info-center/blog/be-careful-there-are-unknown-threats-out-there (Accessed: 2 August 2022).

Bhagat, N. and Arora, B. (2018) 'Intrusion Detection Using Honeypots', in *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 412–417. Available at: https://doi.org/10.1109/PDGC.2018.8745761.

Blaise, A. *et al.* (2020) 'Detection of zero-day attacks: An unsupervised port-based approach', *Computer Networks*, 180, p. 107391. Available at: https://doi.org/10.1016/j.comnet.2020.107391.

Farooq, H.M. and Otaibi, N.M. (2018) 'Optimal Machine Learning Algorithms for Cyber Threat Detection', in *2018 UKSim-AMSS 20th International Conference on Computer Modelling and Simulation (UKSim)*, pp. 32–37. Available at: https://doi.org/10.1109/UKSim.2018.00018.

Franco, J. *et al.* (2021) 'A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems', *IEEE Communications Surveys & Tutorials*, 23(4), pp. 2351–2383. Available at: https://doi.org/10.1109/COMST.2021.3106669.

Hindy, H. *et al.* (2020) 'Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection', *Electronics*, 9(10), p. 1684. Available at: https://doi.org/10.3390/electronics9101684.

Innab, N., Alomairy, E. and Alsheddi, L. (2018) 'Hybrid System Between Anomaly Based Detection System and Honeypot to Detect Zero Day Attack', in *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pp. 1–5. Available at: https://doi.org/10.1109/NCG.2018.8593030.

Lee, J. *et al.* (2019) 'Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles', *IEEE Access*, 7, pp. 165607–165626. Available at: https://doi.org/10.1109/ACCESS.2019.2953095.

Liu, X. and Liu, J. (2021) 'Malicious traffic detection combined deep neural network with hierarchical attention mechanism', *Scientific Reports*, 11(1), p. 12363. Available at:

https://doi.org/10.1038/s41598-021-91805-z.

Lobato, A.G.P. *et al.* (2018) 'An Adaptive Real-Time Architecture for Zero-Day Threat Detection', in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6. Available at: https://doi.org/10.1109/ICC.2018.8422622.

Naik, N. and Jenkins, P. (2018) 'A Fuzzy Approach for Detecting and Defending Against Spoofing Attacks on Low Interaction Honeypots', in *2018 21st International Conference on Information Fusion (FUSION)*, pp. 904–910. Available at: https://doi.org/10.23919/ICIF.2018.8455555.

Negi, P.S., Garg, A. and Lal, R. (2020) 'Intrusion Detection and Prevention using Honeypot Network for Cloud Security', in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 129–132. Available at: https://doi.org/10.1109/Confluence47617.2020.9057961.

Patidar, C.P. and Khandelwal, H. (2018) 'ZERO DAY ATTACK DETECTION USING MACHINE LEARNING TECHNIQUES', 6(1), p. 4.

Radhakrishnan, K., Menon, R.R. and Nath, H.V. (2019) 'A survey of zero-day malware attacks and its detection methodology', in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pp. 533–539. Available at: https://doi.org/10.1109/TENCON.2019.8929620.

Sethia, V. and Jeyasekar, A. (2019) 'Malware Capturing and Analysis using Dionaea Honeypot', in *2019 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–4. Available at: https://doi.org/10.1109/CCST.2019.8888409.

Shaukat, K. *et al.* (2020) 'Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective', in *2020 International Conference on Cyber Warfare and Security (ICCWS)*, pp. 1–6. Available at: https://doi.org/10.1109/ICCWS48432.2020.9292388.

Tan, L. *et al.* (2022) 'Secure and Resilient Artificial Intelligence of Things: A HoneyNet Approach for Threat Detection and Situational Awareness', *IEEE Consumer Electronics Magazine*, 11(3), pp. 69–78. Available at: https://doi.org/10.1109/MCE.2021.3081874.

Tang, R. *et al.* (2020) 'ZeroWall: Detecting Zero-Day Web Attacks through Encoder-Decoder Recurrent Neural Networks', in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pp. 2479–2488. Available at: https://doi.org/10.1109/INFOCOM41043.2020.9155278.

Ullah, F. *et al.* (2019) 'Cyber Security Threats Detection in Internet of Things Using Deep Learning Approach', *IEEE Access*, 7, pp. 124379–124389. Available at: https://doi.org/10.1109/ACCESS.2019.2937347.

*What Is a Honeypot? Meaning, Types, Benefits, and More* (no date) *Fortinet*. Available at: https://www.fortinet.com/resources/cyberglossary/what-is-honeypot (Accessed: 29 July 2022).

*What is a Packet Sniffer?* (2022) *www.kaspersky.com*. Available at: https://www.kaspersky.com/resource-center/definitions/what-is-a-packet-sniffer (Accessed: 1 August 2022).

Zhang, L. *et al.* (2021) 'Research on Unknown Threat Detection Method of Information System Based on Deep Learning', *Journal of Physics: Conference Series*, 1883(1), p. 012107. Available at: https://doi.org/10.1088/1742-6596/1883/1/012107.

*Jithinpj9/HDS* (no date). Available at: https://github.com/Jithinpj9/HDS (Accessed: 24 November 2022).