PriMera Scientific Engineering
Volume 3 Issue 5 November 2023
DOI: 10.56831/PSEN-03-086
ISSN: 2834-2550

PriMera Scientific Publications

Spectral Sentinel: Leveraging Shadow Networks and Decoy Systems for Advanced Incursion Surveillance

Type: Research Article
Received: September 17, 2023
Published: October 27, 2023

Citation:
Sourav Mishra., et al. "Spectral Sentinel: Leveraging Shadow Networks and Decoy Systems for Advanced Incursion Surveillance". PriMera Scientific Engineering 3.5 (2023): 17-45.

Sourav Mishra* and Vijay K Chaurasiya

Dept. of Information Technology, Indian Institute of Information Technology Allahabad, Prayagraj, India

***Corresponding Author:** Sourav Mishra, Dept. of Information Technology, Indian Institute of Information Technology Allahabad, Prayagraj, India.

Abstract

It would be an understatement to say that the internet is hazardous in this age of constantly evolving attack mechanisms and pervasive data thefts. For security specialists, it is akin to engaging in an endless game of cat-and-mouse as they traverse an ever-changing landscape. Using only firewalls and antivirus software against a modern, well-equipped army is equivalent to using spears and stones. Social engineering or malware employing packing or encoding techniques that evade our detection tools are all that an adversary needs to compromise our system. Therefore, it is imperative to transcend the limitations of edge defence, which primarily focuses on tool validation, and adopt a proactive strategy that emphasises intrusion identification and prompt response. This can be accomplished through the implementation of an ethereal network, a comprehensive end-to-end host and network approach that not only scales effectively but also ensures accurate intrusion detection. Our objective is not limited to mere obstruction; it also includes a substantial reduction in time. When conventional security measures, such as firewalls and antivirus software, fail, we must swiftly ascertain the nature of the incident and respond accordingly. In industry reports, response times are frequently measured in weeks, months, or even years, which is untenable. Our objective is to reduce this timeframe to hours, a significant reduction that will improve our response capabilities. Therefore, an effective approach to breach detection becomes essential. Together with a robust honeypot system, we employ a Modern Honey Network (MHN) to facilitate honeypot management and deployment while ensuring their security. This fusion includes honeypots such as Glastopf, Dionaea, and Kippo, which document suspicious activities and capture crucial details of the attacks on the MHN server. In addition, reconnaissance is essential to our research. Recognising the complexities of reconnaissance, we make it the focal point of our efforts. When malware or insider threats penetrate our network, they frequently conduct reconnaissance to determine the extent of their access. By closely observing this type of activity, we can readily identify any suspicious network intrusions or compromised Internet of Things devices. Our deployment strategy concludes with the installation of MHN, the deployment of Dionaea, Kippo, and Snort honeypots, and their integration with Splunk for effective analysis of captured attacks. This integration enables us to

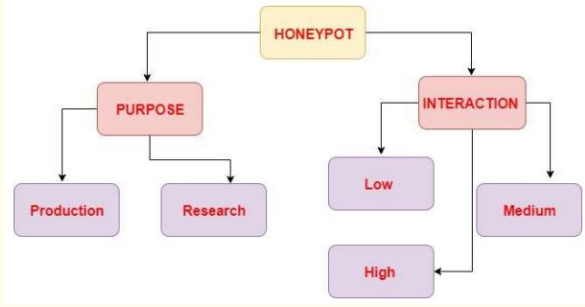
identify the specific service ports under attack and trace the assailants' source IP addresses, providing invaluable information for further investigation and mitigation.

Keywords: Breach; Ethereal; Intrusion Detection Systems; Honeypot; Reconnaissance

Introduction

Honeypots and their Role

A honeypot is fundamentally a deceptive system designed to resemble a legitimate one. Honeypots are designed to entice hackers to direct their attacks against these simulated targets rather than actual systems. By doing so, the deploying organisation obtains valuable insights into the tools, tactics, and techniques employed by the hackers, enabling them to potentially anticipate and respond to attacks on their actual systems with greater preparedness and foresight [1]. These honeypots are comprised of fake files and directories that resemble those found on legitimate computers. However, their purpose is to observe the behaviour of hackers as they interact with honeypots. Honeypots are essentially artificial devices designed to resemble actual systems. In addition to documenting malicious activities on compromised devices, they serve as a platform for studying new threats. In turn, this enables the development of effective countermeasures to mitigate these emergent threats and improve users' overall cybersecurity [1].



Honeypots Types

A network intrusion detection system (IDS) is a device or application designed to detect and respond to potential network intrusions or malicious activities. It functions as a vital component in the fight against cybercrime by providing effective mechanisms for both identifying and preventing attacks [3]. In the current digital environment, combating cybercrime requires robust identification and prevention measures. Antivirus software, firewalls, and other security measures play a crucial role in protecting computer systems. In addition, the four fundamental pillars of computer security— authentication, confidentiality, availability, and integrity — serve as the basis for assuring the protection of specified data [4].

Research honeypots

Honeypots are utilised for security enhancement and academic research, serving as valuable instruments for tracking and analysing attacks as they occur. These honeypots are intended to generate recordings that can be followed and analysed if they are compromised or stolen. Research honeypots are frequently utilised by military intelligence agencies and organisations engaged in espionage because they have the capacity to capture a substantial quantity of valuable information [2]. Honeypots serve as a preventative measure by preventing malicious entities from entering critical systems. They employ mechanisms such as encryption to prevent unauthorised access to sensitive data by assailants. Honeypots serve as a barrier against unauthorised access by diverting potential assailants away

from real systems through the creation of a deceptive environment. Honeypots function as early warning systems by detecting the presence of attackers when a security breach occurs. Their deceptive nature enables them to attract and ensnare malicious actors, allowing security teams to closely monitor their activities and gain valuable insights into their tactics, techniques, and motivations. In the event of a compromise or suspected breach, honeypots can be taken offline swiftly and easily. This proactive response mechanism assists in containing potential attacks and mitigating further network damage. By isolating the honeypot, security teams are able to analyse the attack, determine its impact, and implement the appropriate countermeasures. By achieving these goals, honeypots contribute to an all-encompassing security strategy by enhancing protection, facilitating research and analysis, and providing valuable intelligence on evolving attack methods.

Low Level honeypots

Our method concentrates on the emulation of UDP/TCP listening ports to detect scanning activities effectively. This method's straightforward deployment process makes it suitable for a variety of security configurations. By simulating these ports, we create an environment that appears alluring to potential assailants, enticing them to interact with the honeypot system.

Medium Level honeypots

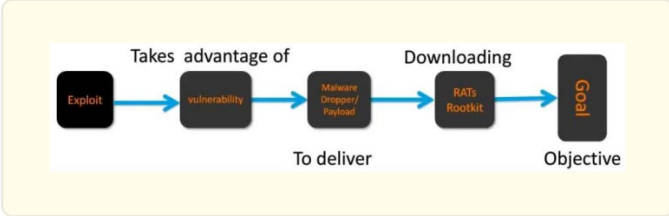
To increase the authenticity of the honeypot, we permit perpetrator login attempts. This tactic enables us to collect valuable information regarding their methods, instruments, and intentions. In addition, the honeypot contains fundamental file structures, further simulating a real system and allowing for the observation of an attacker's behaviour when interacting with the files.

High Level honeypots

It is essential to observe that our strategy emphasises system emulation to ensure a comprehensive deception strategy. Although we have not incorporated high-level honeypots into our methodology, our approach continues to be effective in attracting and engaging potential assailants, allowing us to gain valuable insight into their activities and intentions.

Basic Attack Concept

Malicious software, also known as malware, poses significant hazards to network security and exposes organisations to numerous security risks. Malware, such as worms, computer viruses, Trojan horses, spyware, and rootkits, can infiltrate and compromise the integrity of computer systems. These malicious programmes aid assailants who wish to conceal their presence on compromised computers.



Self-replicating programmes, or worms, can rapidly propagate across networks, causing disruptions and consuming valuable resources. Viruses are programmed to infect files and replicate, frequently resulting in unintended and detrimental effects. Trojan horses masquerade as legitimate software but conceal malicious payloads, enabling unauthorised system access. Spyware, as its name implies, accumulates sensitive data or monitors user activity without their knowledge. Rootkits are covert tools that grant unauthorised access to and control over a compromised system, making them especially perilous.

These types of malware pose significant threats to the security and privacy of network systems. They can compromise sensitive data, disrupt operations, and grant assailants unauthorised access. In addition, their ability to conceal their presence on compromised

devices makes them challenging to detect and mitigate.

To protect against these threats, organisations must implement robust security measures, such as current antivirus software, firewalls, intrusion detection systems, and employee education on safe browsing and email practises. Regular security updates and upgrades are essential for addressing vulnerabilities that could be exploited by malware. Organisations can mitigate the risks posed by malware and safeguard their networks from compromise by adopting a proactive and multilayered approach to security.

Advantages of Honeypots

Implementing honeypots provides a number of benefits in terms of capturing attack details and acquiring useful information. Honeypots are especially effective at detecting large-scale intrusions that may pose a significant threat. Honeypots are effective at managing incoming malicious traffic because they are not burdensome in terms of data acquisition despite their ability to collect vital information. The ability of honeypots to collect tiny but highly valuable data sets is one of their advantages. Honeypots enable organisations to obtain valuable insights into emerging threats by focusing on quality rather than quantity. In addition, honey-pots help reduce false positives, which are erroneous indicators of malicious activity. By deploying honeypots, organisations can effectively distinguish between legitimate and malicious traffic, thereby enhancing the accuracy of intrusion detection systems. In addition to being effective, honeypots are cost-efficient and simple to deploy. They do not require substantial resources or intricate configurations, making them accessible to organisations of varying sizes and security capabilities. This ease of deployment enables rapid deployment and integration with existing security infrastructures. Furthermore, honeypots are resource-efficient, consuming neither superfluous computing power nor network bandwidth. This minimal resource requirement guarantees that honeypots can operate without affecting the overall network or system performance. IPv6, the newest iteration of the Internet Protocol, is compatible with honeypots. This compatibility ensures that organisations can utilise honeypots in IPv6-based modern network environments. Honeypots offer a variety of advantages, such as capturing attack details, managing malicious traffic, accumulating valuable data, reducing false positives, being cost-effective and simple to deploy, requiring minimal resources, and being IPv6 compatible. Due to these benefits, honeypots are an effective and practical addition to a company's cybersecurity arsenal.

Disadvantage of Honeypots

Despite the fact that honey-pots offer a number of benefits, there are some limitations and potential hazards associated with their use:

- Honeypots can only capture information when they are actively under attack. If hackers are not actively targeting the honeypot, it is difficult to collect information and gain insights.
- Honeypots are designed to have limited impact on other systems, as they are intended to be isolated and not interfere with or damage other systems. As a result, they may be incapable of detecting assaults on other systems, limiting their scope of detection.
- Fingerprinting vulnerability: Hackers with experience may be able to distinguish between a genuine system and a honey-pot. This technique, known as fingerprinting, enables assailants to identify honeypots, potentially reducing their efficacy.
- Honeypots have a limited field of view and can only capture information related to attacks directed at themselves. They may be unaware of network-wide attacks targeting other systems.
- As honeypots are designed to entice assailants, there is a chance that hackers will become suspicious or aware of their presence, causing them to alter their behaviour or even take countermeasures.
- Analysis of compromised honeypots: When a honeypot is compromised, it can be difficult to analyse the attack and determine the scope of the breach. The honeypot itself may only provide limited information or data that has been compromised.
- High interaction honeypots, which simulate comprehensive systems, introduce a greater level of risk compared to low interaction honeypots. High interaction honeypots with extensive functionality may be susceptible to potential vulnerabilities.

When deploying honeypots as part of a comprehensive cybersecurity strategy, it is crucial to consider these limitations and risks. Organisations should evaluate their objectives and requirements thoroughly and implement the necessary safeguards and countermeasures to mitigate any potential drawbacks associated with honeypot usage.

Modern Honey Network for honeypots and Phantom Networks

MHN (Modern Honey Network)

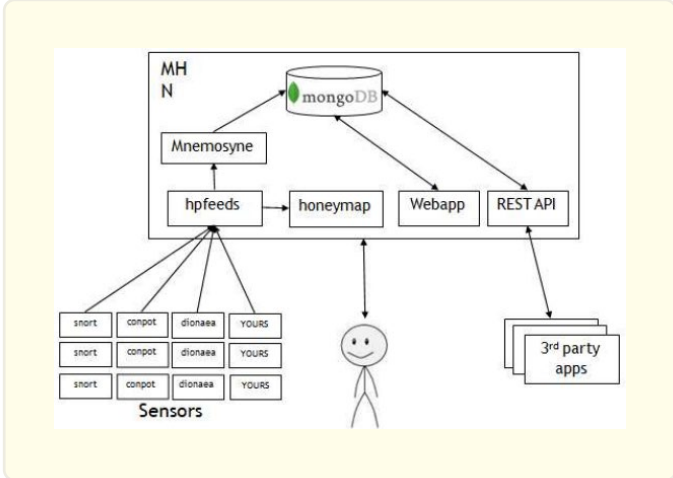
MHN (Modern Honey Network) is a potent open-source tool that simplifies the deployment and management of a broad variety of honeypots. It is a comprehensive server administration system that collects attack data from multiple honeypot sensors. The data collected by these sensors is then displayed on a honeymap, which provides real-time information regarding server attacks. Honeymap actively extracts sensor data and displays it on the MHN dashboard, which can be integrated with Splunk for further analysis.

Key characteristics of MHN include:

- MHN offers a centralised platform for effectively administering and monitoring multiple honeypots via a unified interface. This unified interface makes administration and coordination of multiple honeypot technologies easier.
- Aggregation of vast quantities of sensor data: MHN is capable of aggregating vast quantities of sensor data gathered by the network's deployed sensors. This data compilation permits exhaustive analysis and detection of attack patterns and tendencies.
- Installation of additional sensors is simple with MHN's user-friendly installation process. This streamlined procedure permits security professionals to rapidly deploy additional honeypots and increase their monitoring capabilities.
- Modern Honey Net installation and operation are simple because MHN is designed to be user-friendly, making installation and operation of the Modern Honey Net simple. The intuitive user interface and configuration options allow security teams to manage their honeypot infrastructure with ease.
- The MHN sensors are designed to be easy to configure and operate. This ensures that security professionals can deploy and manage honeypot sensors rapidly and without extensive technical knowledge.
- MHN provides insights into the nature of the observed connections, underscoring that a significant portion of the observed traffic originates from automated bots. This information can be useful for gaining a comprehension of the prevalent threat landscape and formulating suitable defence strategies.

MHN supports several honeypot technologies, such as cowrie, snort, dionaea, and glasnopf. These technologies, which are available in script format, are readily deployable within the MHN framework, providing flexibility and versatility in honeypot selection and deployment. MHN is a robust tool that simplifies the administration of honeypots, aggregates and visualises attack data, and provides an effective method for deploying and operating honeypot sensors. Its user-friendly interface and integration capabilities make it a priceless asset for organisations seeking to bolster their cybersecurity defences and gain valuable insights into the evolving threat landscape.

MHN architecture



MHN (Modern Honey Network) distinguishes itself as a flexible alternative to existing honeypot deployment solutions on the market, with expanded deployment capabilities. It utilises open-source tools such as hpfeeds and honeymap for the efficient accumulation of sensor data and real-time visualisation, respectively. Incorporating support for additional open-source tools such as Kippo, Glastopf, and additional sensors is on the roadmap for MHN. It integrates seamlessly with Ubuntu, a widely used operating system, ensuring compatibility and implementation simplicity.

The MHN web application provides valuable insight into the most recent assaults by displaying the top five IP addresses of attackers and the ports they have targeted. This enables security professionals to prioritise their response and take preventative action against emergent threats. The honeymap feature provides a visual representation of the honeypot network and provides real-time information on attack activities. In addition, an activity log provides a comprehensive record of any malicious activity detected.

Organisations that utilise MHN gain access to a comprehensive set of tools that simplifies honeypot deployment, data collection, and visualisation. Its compatibility with widely used open-source tools and integration with Ubuntu increase its adaptability and usability. MHN's plethora of information enables security teams to comprehend the ever-changing threat landscape and take proactive measures to bolster their defences.

MHN is a robust platform for deploying and administering honeypots, utilising open-source tools for data collection and real-time visualisation. Its Ubuntu compatibility and intentions to incorporate support for additional tools make it a versatile option for organisations seeking to improve their cybersecurity strategies. The insights provided by MHN's web application allow for effective threat analysis and response, thereby enhancing the security posture overall.

MHN Deployment Steps

Using the provided system, the deployment and administration of the honeypot can be summed up as follows:

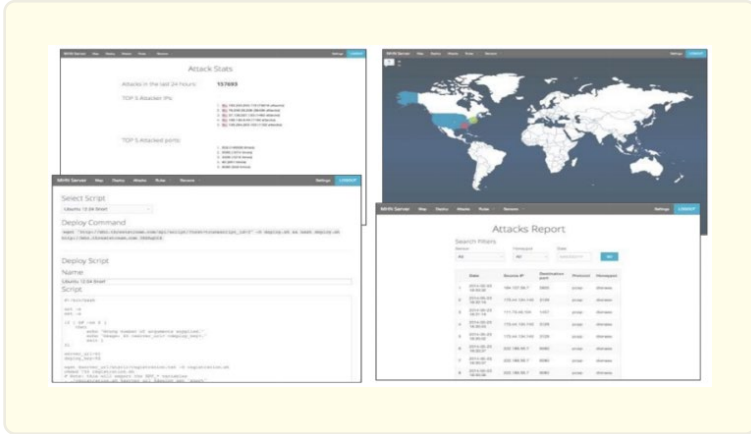
- ***Download and deploy honeypot script:*** Downloading and deploying the honeypot script is the first step. This script creates the infrastructure necessary to simulate a vulnerable system and entice potential attackers.
- Once the honeypot script has been deployed, it establishes a connection to the management system before registering. This connection permits the honeypot to register itself with the central management server and communicate with it.
- ***Distribution of essential Snort modules:*** The system facilitates the distribution of essential intrusion detection related Snort modules. These modules enhance the honeypot's detection capabilities and allow it to recognise various attack signatures.
- ***Logs of intrusion detection that can be shared:*** The honeypot generates logs of intrusion detection that record details of detected attacks. These logs can be shared, enabling security professionals to investigate and analyse attack patterns.
- ***View a list of recent assaults:*** The system provides a comprehensive inventory of newly identified attacks. This enables security teams to remain current on new threats and adapt their defence strategies accordingly.
- ***Manage Snort rules:*** The system supports the management of Snort rules. This includes downloading new rules, enabling or disabling specific rules, and tailoring the rule set to the honeypot deployment's specific requirements.

The system has four discrete interfaces, each of which serves a specific function:

- The first dashboard provides detailed statistics pertaining to the assaults. It presents data such as the number of attacks, attack trends, and other pertinent metrics, enabling security professionals to monitor the global threat landscape.
- The second dashboard is equipped with a map interface that visually represents the location of detected attacks. This geographical visualisation provides insight into the origin of the assaults and aids in the identification of potential patterns or trends.
- Interface for deployment of honeypot scripts The third interface is devoted to the deployment of scripts for various honeypots. It offers a user-friendly interface through which security professionals can select and deploy honeypot scripts that emulate particular systems or services.
- Interface for analysing attack reports The final interface is intended for analysing attack reports generated by the honeypot. It

provides a comprehensive view of the detected attacks, including IP addresses, ports, and other pertinent information. This interface helps to comprehend the nature of the attacks and facilitates further investigation.

Overall, the provided system facilitates the deployment and management of honeypots. It enables the deployment of honeypot scripts, facilitates the sharing and analysis of intrusion detection logs, and provides capabilities for managing Snort rules. The interfaces of the system provide extensive insight into attack statistics, visual representations of attack locations, and in-depth analysis of attack reports. These features contribute to effective intrusion detection and analysis, thereby improving the organization's overall cybersecurity posture.



There are common graphical representations that can be used to visualise simulation results of honeypot-based intrusion detection. These visualisations can be created using Python libraries for data visualisation like Matplotlib or Seaborn. Here are some instances:

- ***Heatmap Representation of the Confusion Matrix:*** A heatmap representation of the confusion matrix can provide a visual summary of the classification results. It employs colour intensity to signify the number of instances that fall into each category (normal or malicious) and the classification accuracy of each instance. The confusion matrix heatmap would exhibit a color-coded grid, facilitating the identification of patterns of correct and incorrect classifications.
- ***Receiver Operating Characteristic:*** The Receiver Operating Characteristic (ROC) curve is a graphical representation of a classification model's performance. It compares the true-positive rate (TPR) to the false-positive rate (FPR) at different classification thresholds. The ROC curve allows you to evaluate the trade-off between the intrusion detection system's sensitivity (true positive rate) and specificity (true negative rate). Commonly, the area under the ROC curve (AUC-ROC) is also calculated to evaluate the model's overall performance.
- ***Precision-Recall Curve:*** The precision-recall curve is another graphical representation used to evaluate the performance of a classification model, particularly in cases where the data is imbalanced. Precision (positive predictive value) is plotted against recall (sensitivity) at various classification thresholds. The precision-recall curve facilitates the evaluation of the trade-off between precision and recall, enabling the selection of an optimal threshold that strikes a balance between the two metrics.
- ***Bar Chart of Attack Types:*** A bar chart can be used to illustrate the distribution of the various attack types identified by honeypots. Each attack type (e.g., brute-force attacks, malware distribution, SQL injection attempts) is represented by a bar, with the height of each bar representing the proportion or number of occurrences of that attack type in the dataset. This visualisation compares the frequency or prevalence of various attack categories.
- ***Pie Chart:*** A pie chart is a circular, segmented graphical representation used to illustrate the proportion or distribution of various attack types detected by honeypots. Each slice corresponds to a distinct attack type, and its size reflects the proportion or tally of that attack type relative to the entire pie. This diagram illustrates the relative frequency of various attack varieties.
- ***Line Chart:*** Using a line chart, one can visualise the detection time of intrusions over a given time period. The y-axis represents

the detection duration, while the x-axis represents time. Each data point on the graph represents the intrusion detection time at a particular point in time. This diagram enables the observation of trends, patterns, and alterations in the detection time of intrusions.

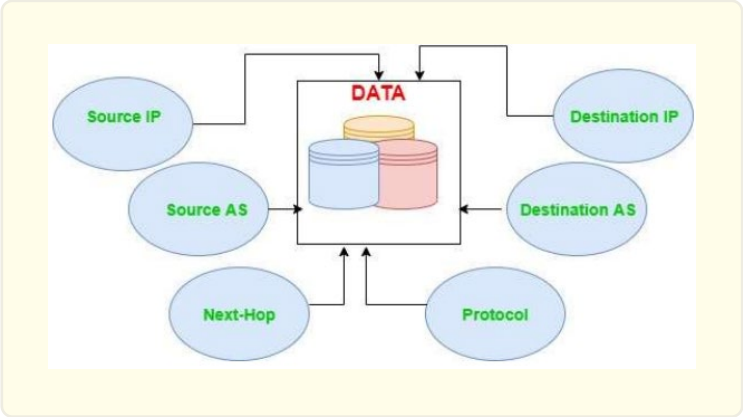
- ***Area Chart:*** Similar to a line chart, an area chart represents the cumulative detection time of intrusions over time using filled areas below the line. The x-axis represents duration, whereas the y-axis represents the total detection time. As the cumulative detection time increases, the filled area progressively increases, providing a visual representation of the overall detection performance over time.
- ***Network Topology Diagram:*** A network topology diagram depicts the network infrastructure and honeypots, as well as their layout and interconnections. It highlights the network architecture, honeypot locations, and their connections to systems and services. This diagram illustrates the deployment and positioning of honeypots within a network.

Ethereal Networks

An Ethereal network, also known as a ghost network, collects comprehensive network data using NetFlow technology. NetFlow is a network monitoring protocol developed by Cisco that captures data on the volume and types of traffic flowing through a network device. NetFlow 9, the data-rich iteration of NetFlow, is essential to our strategy for achieving optimal results.

NetFlow captures a variety of measurements for each network traffic, such as the timestamps of the first and last packets, the total number of bytes and packets exchanged, and a summary of the TCP connection flags. These measurements are then exported to another system for additional examination.

It is essential to note that NetFlow operates in a unidirectional manner. When a client initiates a request, Net-Flow logs the details of the flow. However, a new flow record is created when the server responds, demonstrating the unidirectional nature of NetFlow. By employing a NetFlow monitoring solution, we can monitor and analyse these transit records within the network more efficiently. This allows us to gain valuable insight into the network's utilisation, identify bandwidth-intensive applications and devices, and monitor the external IP addresses with which we exchange data.

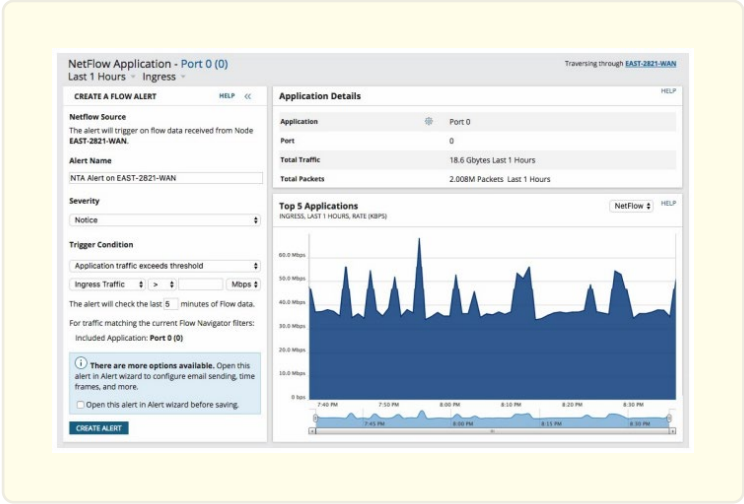


Using NetFlow technology and a NetFlow monitoring solution has the following advantages:

- NetFlow provides an exhaustive set of measurements for each flow, enabling in-depth network analysis via efficient data collection.
- *Utilisation insights:* NetFlow data analysis reveals how the network is being utilised, enabling the identification of trends, patterns, and potential issues.
- By identifying bandwidth users, administrators are able to optimise network resources and guarantee seamless network performance.

- *External IP tracking:* NetFlow data enables the monitoring and tracking of communication with external IP addresses, thereby providing valuable information for security and traffic analysis.

In summary, an Ethereal network collects and analyses network flow records using the power of NetFlow technology. Using a NetFlow monitoring solution, we can acquire valuable insights into network utilisation, identify bandwidth-intensive applications, and monitor external communication. This enables efficient network management and improves the overall security and efficacy of the network.



Data Collection from NetFlow

The Network Traffic Analyzer is able to collect network traffic metrics from a variety of data sources. These sources include commonly used network protocol systems like Cisco NetFlow v5, NetFlow v9, and NetFlow v10 (also known as IPFIX).

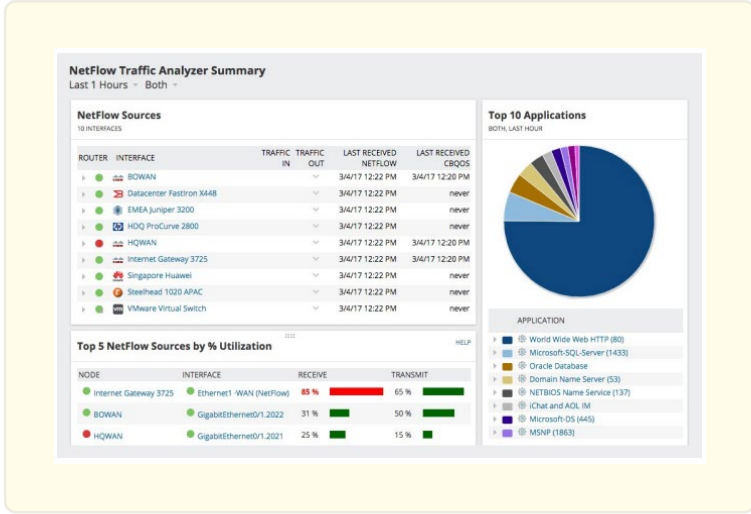
Cisco NetFlow v5 is an obsolete iteration of the company's network monitoring protocol. It provides valuable traffic measurements, including source and destination IP addresses, ports, and the number of bytes and packets exchanged. Net-Flow v5 enables network administrators to obtain visibility into network flow and recognise traffic patterns.

In contrast, NetFlow v9 is an enhanced version of the protocol that provides greater flexibility and extensibility. It introduces the concept of templates, which enables the export of customised data fields in addition to NetFlow v5's standard set. This facilitates the collection of more granular data regarding network traffic, such as application-level details and Quality of Service (QoS) metrics.

NetFlow v10, also known as IPFIX (Internet Protocol Flow Information Export), is a variant of NetFlow that has become the industry standard. It provides similar functionality to NetFlow v9, but allows devices from different vendors to communicate. IPFIX provides a standardised format for exporting flow data, facilitating the analysis of network traffic in heterogeneous network environments.

Utilising these various versions of NetFlow, the Network Traffic Analyzer is able to compile exhaustive network traffic metrics. This includes source and destination IP addresses, ports, protocols, traffic volumes, application-level details, and QoS metrics. This multitude of data enables network administrators and analysts to gain profound insights into network behaviour, identify anomalies or security threats, and make informed optimisation and troubleshooting decisions.

The Network Traffic Analyzer supports multiple versions of the NetFlow protocol, such as Cisco NetFlow v5, NetFlow v9, and IPFIX (NetFlow v10). These versions provide valuable network traffic metrics, enabling comprehensive network behaviour analysis and monitoring.



The Netflow Traffic Analyzer provides a number of essential features that improve network monitoring and analysis:

- The Netflow Traffic Analyzer permits real-time monitoring of network bandwidth utilisation. It provides administrators with comprehensive information about the volume of network traffic, enabling them to identify bandwidth-intensive applications, devices, or users. This information assists in optimising network performance and resource allocation.
- ***Alerting Application Traffic:*** The Netflow Traffic Analyzer has the capability to configure alerts based on particular application traffic patterns or thresholds. Administrators can define custom notifications to be notified when specific applications exceed predefined bandwidth limits or exhibit abnormal behaviour. This proactive strategy enables prompt responses to prospective network issues or policy violations.
- The Netflow Traffic Analyzer captures and analyses network traffic data, providing valuable insight into traffic patterns, leading talkers, and communication flows. It enables administrators to drill down into particular protocols, ports, and IP addresses in order to comprehend network behaviour, identify bottlenecks, and effectively rectify performance issues.
- ***Dashboard for Performance Analysis:*** The Netflow Traffic Analyzer presents network performance metrics and visualisations on a user-friendly dashboard. Administrators have access to exhaustive reports and graphical representations of network traffic, including utilisation trends, leading applications, and leading conversations. This visual representation facilitates data-driven decision making and simplifies performance analysis.
- ***Recognition of Advanced Applications***: The Netflow Traffic Analyzer is able to recognise and categorise a vast array of advanced applications and protocols traversing the network. It is able to identify applications that utilise non-standard or dynamic ports, allowing visibility into encrypted or evasive traffic. This capability is essential for security monitoring because it enables the detection of potential hazards and unauthorised application usage.

The Netflow Traffic Analyzer enables network administrators to gain in-depth visibility into network traffic, proactively manage bandwidth, ensure optimal performance, and detect and respond effectively to network anomalies by leveraging these key features. It offers a comprehensive solution for monitoring, analysing, and optimising network traffic to maintain a secure and effective network infrastructure.

Related Work

The papers that have been discussed shed light on a variety of honeypot-and security-system-related aspects, as well as implementations. The following is a synopsis of each piece of work:

They show an implementation of a contemporary security system honeypot network on wireless networks, which was done by Wafi, Bahaweres, and others. They make use of a honeypot device that is equipped with a Modern Honey Network (MHN) in order to facilitate deployment and management, as well as improve honeypot safety [2].

They offer a new intrusion prediction mechanism that is based on honeypot log similarity. The authors' names are Ci-Bin Jiang and I-Hsien Liu. Their intrusion detection technology does an analysis of honeypot log similarities and makes use of recording mining techniques in order to prevent attacks from occurring by blocking suspicious flows [4].

Ioannis Koniaris, Georgios Papadimitriou et al.: They provide a honeypot deployment technique for the purpose of analysing and visualising the activity of malicious connections and connections made by malware. They also present an open-source visualisation tool with the intention of assisting researchers and security professionals with the process of analysing the collected data and producing conclusions from those analyses [5].

Emmanouil Vasilomanolakis, Shankar Karuppayah et al.: They offer their very own cyber incident monitor, which they name TraCINg, and it gathers alarm data from honeypot sensors that are located all over the world. Their work analyses the data acquired throughout a period of five months of deployment and explains the lessons gained from a design viewpoint. One of the main focuses of their work is the identification of correlated attacks that target several sensors [6].

Amit D. Lakhani, Dr. Kenneth G. Paterson et al.: In honeypots, their attention is focused on the difficulties associated with data anomaly detection. They examine the advantages and disadvantages of employing honeypots as a tool for statistical modelling, behaviour studies, attack detection, and action analysis [7].

They introduce MHN as an open-source project to promote the wider deployment of honeypots by security teams. This was done by S. M. Jignesh Kumar and colleagues. This demonstrates the efficacy of honeypots as a component of active defence by locating probable abnormalities and sending the data to a cloud service for further examination over a longer period of time [8].

The author, J. Gondohanindijo, explains that an Intrusion Prevention System (IPS) is a network security solution that may monitor activity on a system or network for irregularities and respond in real time to prevent hostile behaviours. The research focuses on the Intrusion Prevention System (IPS) as a means to increase computer security and deter intrusions when connected to the internet [9].

Setia Arief Muhammad, Arief Muhammad Juli Irzal Ismail et al.: They address the security flaws and vulnerabilities that exist in the technologies used today for the internet. Honeypots with high levels of interaction are going to be used in their planned implementation method, along with various kinds of supporting software. In order to determine the reliability of the system, durability tests that simulate direct assaults are carried out [10].

These works make significant contributions to the fields of honeypots and security systems by providing new understanding of deployment tactics, intrusion detection mechanisms, data analysis, visualisation tools, incident monitoring, anomaly detection, and prevention mechanisms.

Methodology

We have set our sights on honeypots and NetFlow in our pursuit of effective industry capabilities. Traditional tools such as packet capture and intrusion detection systems (IDS) may do the task, but they are expensive and difficult to scale. How many span ports and connections can be deployed in a vast data centre cloud or vast network? It is like searching for a pinpoint in a haystack while wearing blindfolds. We therefore propose a devious combination of NetFlow and honeypots. In addition to being inexpensive, they offer scalability and real-world results. It is time to abandon illusion and adopt a practical, effective strategy. Here are some mathematical formulas relevant to intrusion detection using honeypots:

Honeypot Effectiveness Equation: A honeypot's effectiveness can be determined by comparing the number of assaults it attracts to the total number of attacks detected by the entire system. Let's denote the honeypot's effectiveness as E. Effectiveness can be computed

as follows:

$$E = A_honeypot / A_total \quad (1)$$

Honeypot Density Equation: The density of honeypots within a network can have an effect on its detection capabilities. Density will be denoted as D. If $N_honeypots$ is the number of honeypots deployed and N_etwork is the total number of systems in the network, then the density can be calculated as follows:

$$N_honeypots / Number\ of\ Networks \quad (2)$$

Honeypot Attractiveness Equation: A honeypot's attractiveness can be determined by the ratio of the number of attacks it attracts to the number of legitimate interactions. Let's designate attractiveness with the letter At. The attractiveness can be calculated as follows:

$$Atequals\ A_honeypottoI_honeypot. \quad (3)$$

Equation for Detection Rate: The detection rate quantifies a honeypot's ability to identify and report malicious activities. Let's refer to the rate of detection as DR. If $A_honeypot$ is the number of attacks detected by the honeypot and $A_malicious$ is the number of confirmed malevolent attacks, then the detection rate can be calculated as follows:

$$DR = Malevolent/ Honeypot \quad (4)$$

False Positive Rate Expression: The false positive rate indicates the rate at which the honeypot incorrectly identifies legitimate activities as attacks. Let's abbreviate the rate of false positives as FPR. The false positive rate can be calculated as follows:

$$FPR = A_false_positive / A_legitimate \quad (5)$$

These equations can shed light on the efficiency, density, allure, and detection capabilities of honeypots for intrusion detection.

Honeypot Attractiveness: The attractiveness of a honeypot can be calculated based on the number of attacks it attracts compared to the number of legitimate interactions. Let $A_honeypot$ represent the number of attacks detected by the honeypot, and $I_honeypot$ represent the number of legitimate interactions. The attractiveness (At) can be calculated using the following formula:

$$At = A_honeypot / I_honeypot \quad (6)$$

Honeypot Density: Honeypot density refers to the ratio of the number of honeypots deployed to the total number of systems in the network. Let $N_honeypots$ represent the number of honeypots deployed, and N_etwork represent the total number of systems. The density (D) can be calculated using the following formula:

$$D = N_honeypots / N_etwork \quad (7)$$

Honeypot Capture Efficiency: The capture efficiency of a honeypot measures its ability to capture and log attacks. Let $A_captured$ represent the number of attacks captured by the honeypot, and A_total represent the total number of attacks targeted at the honeypot. The capture efficiency (CE) can be calculated using the following formula:

$$CE = A_captured / A_total \quad (8)$$

False Positive Rate: The false positive rate indicates the rate at which legitimate activities are incorrectly identified as attacks by the honeypot. Let $A_legitimate$ represent the number of legitimate interactions, and $A_false_positive$ represent the number of false positive alerts generated by the honeypot. The false positive rate (FPR) can be calculated using the following formula:

$$FPR = A_false_positive/A_legitimate \quad (9)$$

Consider the following hypothetical situation in which we want to determine the overall risk score for detecting an incursion based on a number of different criteria. We will utilise a weighted sum approach, in which the significance of each factor will determine the weight that will be applied to it in the overall calculation.

The following equation can be used to determine the total risk score, also known as the RS:

$$RS = w1 * F1 + w2 * F2 + w3 * F3 + ... + wn * Fn \quad (10)$$

Where RS represents the overall risk score, w1, w2, w3,...,wn represents the weights that are allocated to each component, and F1, F2, F3,..., Fn represents the values that are obtained for each factor.

Taking into consideration the following three aspects for honeypot-based intrusion detection:

Attractiveness Factor, abbreviated as F1

The honeypot has uncovered a total of fifty different attacks, denoted by the variable $A_honeypot$. $I_honeypot$ is equal to 1000 in terms of the number of legitimate interactions.

Density Factor (also known as F2)

$N_honeypots$ = 5 is the total number of honeypots that have been set up. N_etwork equals 100 to indicate the total number of systems in the network.

Factor of Captured Energy Utilisation (F3)

A total of forty different attacks were stopped by the honeypot, denoted by the variable $A_captured$. The total number of attacks that were directed towards the honeypot is equal to A_otal.

Let us put some weight on each of these criteria, given that we have them:

w1 equals 0.4, which stands for the weight for the attractiveness component. w2 equals 0.3, which stands for the weight for density factor. w3 equals 0.3, which is the weight for the capture efficiency factor.

Following is the formula that can be used to derive the total risk score (RS):

RS equals (0.4 times $A_honeypot$ divided by $I_honeypot$) plus (0.3 times $N_honeypots$ divided by N_etwork) plus (0.3 times $A_captured$ divided by A_otal).

Using the values that have been provided:

$$RS = (0.4\ (50/1000)) + (0.3\ (5/100)) + (0.3\ (40/50)) \quad (11)$$

Following completion of the calculations, the output will be a numerical value that represents the total risk score (RS). This score can be used to assist in assessing the level of risk posed by an incursion based on the variables that are weighted.

These formulas provide quantitative measures to assess the attractiveness, density, capture efficiency, and false positive rate of honeypots for intrusion detection purposes.

When discussing honeypots, the term “honeypot attractiveness” refers to the ability of the honeypot to successfully lure in malicious attackers. This measure is computed by the snippet of code by dividing the total number of genuine interactions ($I_honeypot$) by the number of attacks that were identified by the honeypot ($A_honeypot$). The end result, which is denoted by the letter At, stands for the attractiveness value.

The number of honeypots that are contained within a network can be measured using the "honeypot density" metric. To calculate it, divide the total number of honeypots that have been set up ($N_honeypots$) by the entire number of systems that are connected to the network (N_etwork). The result, denoted by the letter D, is the density value, which indicates the percentage of honeypots present throughout the network. The "honeypot capture efficiency" quantifies how successfully the honeypot thwarts attacks that are aimed at it. This statistic is computed by the code by dividing the total number of assaults that were aimed at the honeypot by the number of attacks that were successfully intercepted by the honeypot ($A_captured/A_otal$). The outcome, denoted by CE, is the capture efficiency value, which indicates how successful the honeypot was in preventing attacks.

```
% Honeypot Attractiveness
A_honeypot = 50;   % Number of attacks detected by the honeypot
I_honeypot = 1000; % Number of legitimate interactions

At = A_honeypot / I_honeypot; % Attractiveness calculation

% Honeypot Density
N_honeypots = 5;     % Number of honeypots deployed
N_network = 100;     % Total number of systems in the network

D = N_honeypots / N_network; % Density calculation

% Honeypot Capture Efficiency
A_captured = 40;    % Number of attacks captured by the honeypot
A_total = 50;       % Total number of attacks targeted at the honeypot

CE = A_captured / A_total; % Capture efficiency calculation

% False Positive Rate
A_legitimate = 2000;         % Number of legitimate interactions
A_false_positive = 50;      % Number of false positive alerts generated

FPR = A_false_positive / A_legitimate; % False positive rate calculation

% Displaying the results
fprintf('Honeypot Attractiveness (At): %.4f\n', At);
fprintf('Honeypot Density (D): %.4f\n', D);
fprintf('Honeypot Capture Efficiency (CE): %.4f\n', CE);
fprintf('False Positive Rate (FPR): %.4f\n', FPR);
```

```
% Honeypot Attractiveness
A_honeypot = 50;   % Number of attacks detected by the honeypot
I_honeypot = 1000; % Number of legitimate interactions

At = A_honeypot / I_honeypot; % Attractiveness calculation
```

The "false positive rate" calculates the percentage of false positive alerts produced by the honeypot in comparison to the total number of real contacts. To determine it, divide the total number of false positive alerts produced by the honeypot ($A_f alse_p ositive$) by the total number of valid interactions ($A_legitimate$). This will give you the total number of false positives. This result, known as the false positive rate (FPR), provides information regarding the accuracy of the honeypot's alerting system. When all is said and done, the findings are presented by utilising the fprintf function, which results in a polished output of the calculated metrics. It is essential to do these analyses in order to evaluate the efficiency and performance of the honeypot. This will enable security professionals to make educated decisions concerning the deployment of honeypots, the protection of networks, and the management of potential risks. Below is an example of a machine learning algorithm for intrusion detection using honeypots. The algorithm used is called Random Forest Classifier:

```
% Honeypot Density
N_honeypots = 5;     % Number of honeypots deployed
N_network = 100;     % Total number of systems in the network

D = N_honeypots / N_network; % Density calculation
```

```
% Honeypot Capture Efficiency
A_captured = 40;    % Number of attacks captured by the honeypot
A_total = 50;       % Total number of attacks targeted at the honeypot

CE = A_captured / A_total; % Capture efficiency calculation
```

```
% False Positive Rate
A_legitimate = 2000;         % Number of legitimate interactions
A_false_positive = 50;      % Number of false positive alerts generated

FPR = A_false_positive / A_legitimate; % False positive rate calculation
```

```
# Importing the required libraries
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

# Step 1: Prepare the dataset
# Assume you have a dataset with features (X) and corresponding labels (y)
X = # Your feature data
y = # Your label data

# Step 2: Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 3: Create and train the Random Forest Classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train, y_train)

# Step 4: Make predictions on the test set
y_pred = rf_classifier.predict(X_test)

# Step 5: Evaluate the model
print(classification_report(y_test, y_pred))
```

The code starts off by importing the required libraries into memory. It does so by importing the RandomForestClassifier that can be found in the ensemble module of scikit-learn. This is the component that will be utilised to generate a random forest classifier.

```
# Importing the required libraries
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
```

Additionally, it imports *classification_report* from scikit-learn's metrics module in order to analyse the performance of the classifier and *train_test_split* from scikit-learn's *model_selection* module in order to divide the dataset into training and testing sets. Both of these modules are part of the *model_selection* module of scikit-learn.

```
# Step 1: Prepare the dataset
# Assume you have a dataset with features (X) and corresponding labels (y)
X = # Your feature data
y = # Your label data
```

At this stage, feature data, denoted by X, and label data, denoted by y, were made available. The feature data stand in for the independent variables that are going to be utilised in the label prediction process.

```
# Step 2: Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

With the help of the *train_test_split* function, the dataset is partitioned into a training set and a testing set at this point. It begins by taking the feature data (X) and the label data (y), and then it divides those two sets of information into the *X_train*, *X_test*, *y_train*, and *y_test* categories, respectively. The *test_size* argument identifies the section of the dataset that will be used for testing (in this example, 20 percent of the dataset is used), and the *random_state* parameter ensures that the split can be reproduced accurately.

```
# Step 3: Create and train the Random Forest Classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train, y_train)
```

Using the RandomForestClassifier class that is included in scikit-learn, a Random Forest Classifier is generated in this stage of the process. The number of decision trees that will be included in the random forest can be specified by using the *n_estimators* option (in this example, 100 trees are utilised). The *random_state* parameter guarantees that the random forest can be reproduced accurately. The classifier is subsequently trained using the fit technique on the training data, during which it discovers patterns and correlations between the features and the labels.

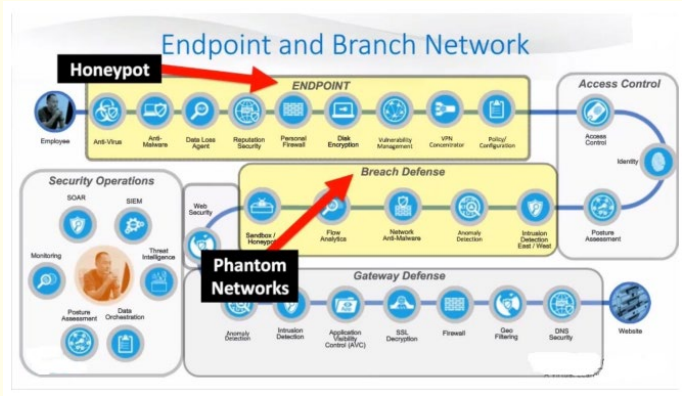
```
# Step 4: Make predictions on the test set
y_pred = rf_classifier.predict(X_test)
```

The trained random forest classifier is then used in conjunction with the predict method to generate predictions on the test set (*X_test*). The variable *y_pred* is where we keep track of the labels that were predicted.

```
# Step 5: Evaluate the model
print(classification_report(y_test, y_pred))
```

At this point, the code will finally produce the classification report in order to evaluate how well the random forest classifier worked. The *classification*$_r$*eport* function creates a thorough report that includes metrics like as precision, recall, F1-score, and support for each class. It does this by comparing the labels that were predicted (y_p*red*) with the labels that were actually assigned (y_t*est*). This report offers insights into the performance of the classifier, including its accuracy in predicting various classifications.

Now, let's get down to brass tacks. This network diagram is a magnificent compilation of a generic branch network. We are preoccupied with the interior workings, ignoring the mundane entrance and exit points. Why, you ask? Well, focusing on endpoints restricts us to specific regions of the network. In contrast, a network-centric approach may miss the mark if the cunning assailant avoids our trap-laden territories. This is why we advocate for the ideal combination of honeypots and NetFlow, which provides us with a genuine end-to-end defence.



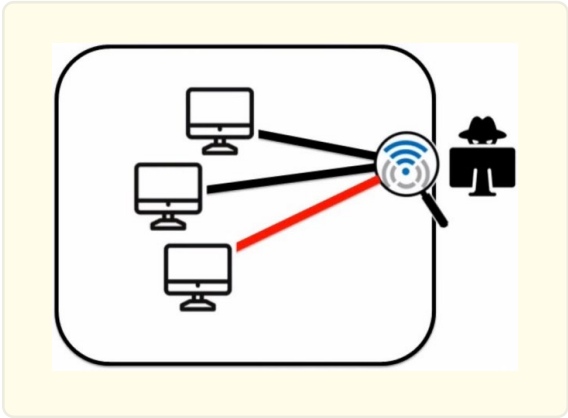
However, there is more! Consider how we observe and analyse our digital universe. Many cutting-edge technologies, such as antivirus and intrusion prevention systems, rely on signatures. They play a pattern-matching game, observing an egg hatching or receiving a threat feed, and voila! an alarm is activated. However, these strategies frequently fail against cunning assailants who employ encoding and polymorphism to conceal their malice. This is where honeypots and NetFlow's magic enters into play. We utilise the effectiveness of anomaly and behaviour. My dear companions, anomaly requires establishing a baseline. We employ the NetFlow method to monitor the network for anomalous behaviour. In contrast, behaviour focuses on reconnaissance. It is comparable to detecting an unknown threat whose behaviour reeks of malice. These two characteristics complement one another beautifully. Essentially, assailants are aware when they have infiltrated a network. Their thoughts are along the lines of, "Oh, I've stumbled upon an enigmatic environment. There must be annoying security instruments on guard." Consequently, they employ subtlety in an effort to avoid detection. The intriguing aspect? The more they conceal themselves, the more they stand out as an outlier. It's a paradoxical dance in which their attempts to outwit us only attract more notice. Even if they choose not to be deceitful, their aberrant behaviour exposes them to our keen observation. This is precisely why we preach about more than just phantom networks and honeypots. We harmonise anomaly and behaviour to forge a remarkably scalable and robust defensive strategy.

Therefore, fellow defenders of the digital domain, let us embark on this ingenious journey of combined capabilities, where honeypots and NetFlow weave a tapestry of security prowess. Together, we will overcome the obstacles that stand before us.

Ethereal/Phantom Networks

Ah, the enchanted realm of Ethereal network topologies, where NetFlow reigns paramount. NetFlow, my dear colleagues, has been with us for a considerable amount of time, assuming numerous forms. Permit me, however, to impart a crucial recommendation: embrace the succulent abundance of NetFlow 9, the creme de la creme of data richness. Why, you may ask? With NetFlow 9, we now have the ability to decipher the who, what, and when of a network. It grants us the divine information that reveals the existence of a problem. Unfortunately, opting for a NetFlow variant that is less verbose or less informative would leave us wanting answers. In the domain of NetFlow, remember that data alone is an empty vessel. It longs for a hint of intelligence, a flash of insight. I must warn you that not all NetFlow tools are created equally. Consider the case of SolarWinds. A remarkable application that is renowned for resolving telephonic dilemmas and de ciphering RAT issues. However, its true focus rests in a realm unrelated to security. It is incapable of identifying the malicious schemes of malware, viruses, and insider threats. Consequently, we must carefully select our allies, pursuing the appropriate instrument that aligns with our security objectives. Mind you, it's not enough to simply possess NetFlow; we must possess the appropriate NetFlow and wield it with expertise. My beloved companions, sampled flows will not suffice. They simply cannot withstand this approach's rigour. Imagine a loud alarm announcing that something has gone amiss within the past 24 hours. Unfortunately, such information contributes little to our pursuit of truth. Consequently, our firm recommendation stands: NetFlow 9. There are vendors that offer converters capable of transforming unprocessed traffic into NetFlow's radiant splendour. Most contemporary tools, rest assured, support our cherished NetFlow. Now, some may be concerned about performance issues. As the proverbial final straw that breaks the camel's back, the impact on performance should be minimal unless your system is already on the verge of collapse. In fact, we can even conjure phantom networks, our cherished Phantom networks, which serve as devious detection tripwires. Oh, the craftsmanship of it all! Together, armed with the appropriate NetFlow and the knowledge to employ it, we will navigate the ethereal realm of Ethereal networks, deciphering its secrets and protecting our digital domains. Friends, we advance towards a future illuminated by intelligence and safeguarded by unwavering vigilance.

My dear colleagues, the art of reconnaissance is the heart of our endeavour. A delightful surprise awaits a cunning assailant as they begin their reconnaissance mission, slyly probing and tickling the networks before them. A resounding alarm reverberates throughout the digital universe, for those networks they encounter are merely phantoms in the vastness of cyberspace. Within their mystical confines, no legitimate interaction should occur. How audacious of the intruder to invade the ethereal! We can create an army of these phantom networks, an infinite procession of confounding and bewildering decoys. Oh, the splendour of deceit!



Indeed, stories proliferate of organisations declaring an absence of intrusion while remaining oblivious to the impending threat. With this approach, however, a revelation emerges. The audacious attacker exploited a simple exception buried deep within the firewall's list, a neglected opening that allowed unauthorised traffic to flow freely. Oh, the absurdity of everything!

Permit me to regale you with a few examples of esteemed clients who have successfully adopted this strategy. In a wondrous DISA-based network, an X-ray machine, motivated by curiosity or possibly mischief, began port surveys, exploring the vast expanse of interconnection. Unbeknownst to it, the phantom network was lying in wait as an ingenious trap for stray port scanners. The illusory network cried out, "Why are you scanning me, unworthy IP address?", alerting the defenders to the intrusion. Theirs was the victory.

Another virtuous consumer, equipped with the mighty Splunk, harnessed the power of Phantom data. With each alarm that punctured the silence, a single click revealed a widget that provided a glimpse into the complexities of the unfolding drama. Friends, this is a genuinely effective strategy in which knowledge and swift action collide. Unfortunately, even grandeur has limitations.

Consider a scenario in which an assailant, fearful of setting off a security alarm, opts for a more tactical approach. They avoid scanning with unbridled abandon and navigate the digital labyrinth with care. In such deliberate endeavours, it is possible that they will not encounter our illusory networks. Friends, this is a dance of minds in which the attacker and defender play a delicate game of shadow and light.

Even in the face of such constraints, our grand strategy flourishes. It exemplifies the ever-changing nature of our craft, the eternal struggle between the forces of deception and those who seek to penetrate our digital fortifications. With each constraint, we discover new opportunities to strengthen our defences.

Let us proceed, armed with our phantom networks, our ingenious traps, and our unwavering resolve, my esteemed companions. Together, we will vanquish the darkness in order to preserve the sanctity of our digital domains. In this everchanging environment, our victory is secured through the art of deception and vigilant reconnaissance. We're off to the races on the virtual battlefield!

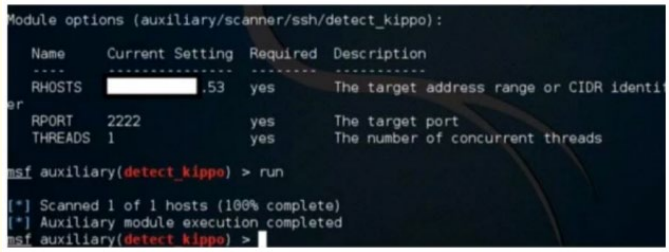
And that's where part two comes in, which is honeypots.

Honeypots

We encounter the mysterious world of honeypots in the domain of cybersecurity, where shadows dance and deception is our ally. These inventive creations are available in low, medium, and high levels of interactivity. Imagine the honeypot as a mute observer capturing the delicate dance of scanning attempts. Rapidly deployable, it serves as our first line of defence, always vigilant. As we ascend, we encounter the medium interactive honeypot, an enticing trap that gives the intruder a glimpse of the desired resources but withholds them. Oh, the craft of seduction! And at the apex is the high interactive honeypot, a playground of possibilities where intrepid intruders may venture unwittingly. Nonetheless, if simplicity is our guiding principle and our objectives extend beyond learning about the attacker, we opt for honeypots that concentrate on reconnaissance activities. As we construct our phantom network, however, let us imbue it with personality, for IOT-based honeypots await our command within the vast domain of open-source marvels. A tapestry of intrigue is being woven!

Nonetheless, we must proceed with prudence along the path of honeypot usage. Although captivating, the high interactive brethren have no influence over our noble endeavour. Instead, we embrace the sophistication of low or medium interactivity honeypots, incorporating simplicity into our defensive web. A deluge of alarms and information may, however, dilute our purpose and shroud us in confusion. Never forget the guardians of deception, my companions! Neglecting the administration of our honeypots results in a gloomy outcome: an unheard and unheeded warning system. Let strategic sagacity guide our selection of honeypots, transforming them into watchful tripwires capable of thwarting the insidious murmurs of insider threats. Together with our ethereal network strategy, we will scale the heights of security, protecting every nook and cranny of our digital realm. As we refine our strategies and delve into the psyches of our opponents, precision becomes our art, as it is the key to victory.

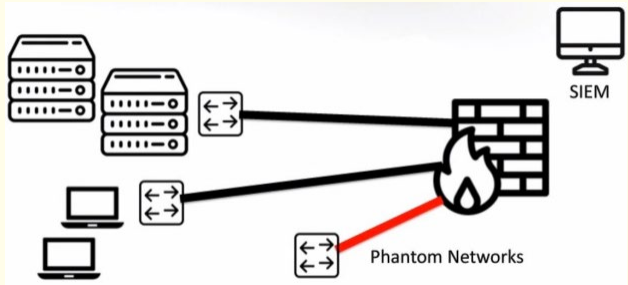
Let us now explore the union between the ethereal network and the ancient domain of honeypots. While ethereal networks are relatively new, honeypots have long been a part of our defences. Yet, we must always remain vigilant, as cunning adversaries employ anti-detection techniques as their primary weapon. Fear not, as the ethereal network approach trumps their ingenious manoeuvres and effortlessly unravels their plans. However, beloved comrades, when it comes to honeypots, there are detectors lurking in the shadows, attempting to reveal their deceptive nature.



Observe the kippo honeypot detector, a beacon of truth the haze of doubt. Observe how Kali Linux and the formida Metasploit aid us in our pursuit of security. Let us not provi our foes with an easy path, for our honeypots will assu the guise of genuine devices, ensnaring the unwary with their convincing fac¸ade. These, my companions, are only a few suggestions to guide our initial foray as we proceed with unwavering determination.

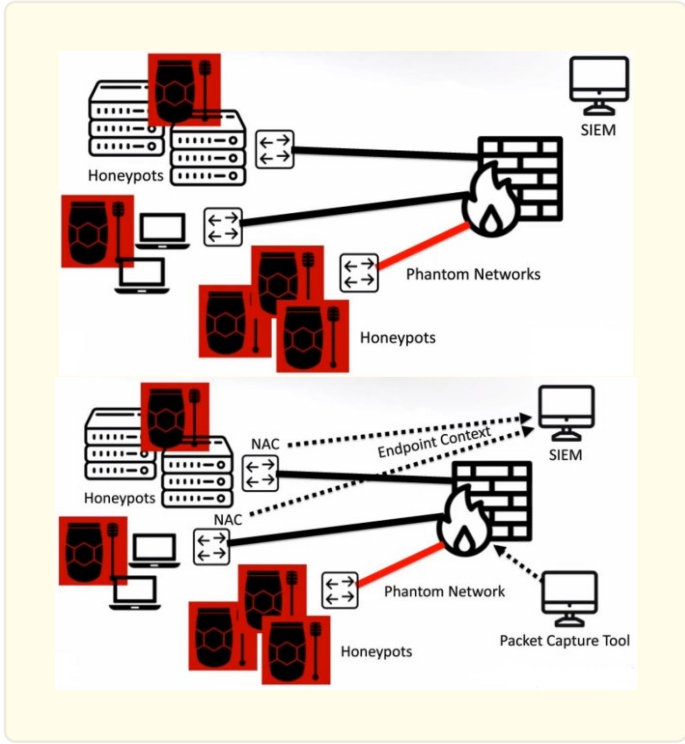
Utilising cleverly constructed honeypots and an ethereal network strategy, we traverse the complex terrain of cybersecurity, where shadows whisper secrets and victory awaits. Together, we will surmount the obstacles that stand in our way, fortifying our defences and shedding light on the schemes of those who dare to invade our territory. Proceed, comrades, into the realm where deception meets reality, where deception's guardians stand erect, and where victory is our destiny.

Total Deception



In our pursuit of breach detection, we have set our crosshairs on impending threats. We desire a scalable strategy that transcends network boundaries and embraces both the virtual and cloud worlds. When our initial defences fail, we seek a beacon of detection, so simplicity becomes our guiding principle. And behold, the ethereal network approach, which grants us the ability to generate an infinite number of illusory networks. A single tremor within these virtual constructs alerts us to a security compromise, allowing us to respond quickly and mitigate the damage. Friends, let us not ignore the honeyed allure of honeypots, as they have the power to bridge

the gaps in our defences. Imagine a data centre devoid of numerous networks, but have no fear; we can unleash virtual servers upon its desolate terrain. This network segment will be guarded by these faithful honeypots with unwavering vigilance. As we scale our efforts, a centralised SIEM or comparable entity emerges as our trusted ally, beckoning the records of our digital protectors. Our ultimate objective is simple value-add, and we disregard complexity. A thorough incident response necessitates the consolidation of data, the grand tapestry of records and alarms woven together for our examination and response. Phantom networks pervade the vast expanse, while honeypots adorn both the ethereal and actual networks. Our shield is deception, and reconnaissance is the telltale sign of intrusion. And behold, we engage in an integration dance, intertwining our defences with our preferred response tools. Two common paths present themselves to us, comrades. A instrument for packet capture, such as NetWitness, or a script that examines the inner workings of a TCP dump. Or, if we dare to adopt enterprise access control solutions such as FortiNAC, Forescout, or Cisco ISE, their APIs and quarantining capabilities allure. However, let us elucidate the concept of quarantine, as we are not physically expelling intruders from our networks. We prefer a more covert approach, limiting their network access or directing them to a guest network. Consequently, the incident response transpires in a domain distinct from our sensitive data centre and its valuable secrets. This dance of integration with NAC solutions and SIEMs has adorned our presence for the past five years, and we embrace its insight as we forge our way forward.



And now, as we approach the conclusion of our discussion, let us offer some concluding advice to elucidate our path.

First, honeypots are not intended to be directly monitored. Their information must be collected, protected, and processed. In the realm of Splunk or QRadar, we seek the exporter—Splunk Stream, Splunk Universal Forwarder—its

name is irrelevant because it is the conduit through which our honeypots' alarms travel. However, we must not suffocate in the information deluge. We do not care about inbound activities aimed at our fictitious systems, so simplicity prevails supreme. The honeypot, which is merely a fac˛ade, desires to unearth the latent insider threat. If we desire a medium-sized interactive honeypot, we simplify the triggers and concentrate on key indicators of an attacker's malicious actions.

Enter the realm of the NetFlow tool, companions, whose purpose is to send alerts or logs to our beloved SIEM. And as soon as the data graces the SIEM, refining becomes our art. We endeavour to create straightforward widgets by combining honeypot logs and the NetFlow tool. In this grand symphony of detection, the essence of reconnaissance is distilled, as it is the most probable indicator of a breach. If our curiosity is piqued, we may design widgets that reveal the identities of perpetrators and even their country of origin, although such information may be of little value other than for geofencing purposes. Remember that our objective is not to repel the assailant, but to discover their presence. Allow the widgets to sound an alarm in the event of a security breach, enabling us to open complaints, investigate, and distinguish real threats from false positives.

Thus, we approach the conclusion of our voyage, where statistics and diagrams will adorn the perceptive eyes of our leadership. However, we must resist the temptation to inundate our audience with NetFlow logs or event logs from our honeypots. In fact, simplicity guides our path, as true value-added rests in its simplicity.

As we contemplate the next steps, let us embrace the Phantom concept. Acquire the NetFlow application, invest in its features, and conjure the Phantom network. Deploy a modest honeypot that is vigilant and unyielding, and ensure that its logs are embraced by your SIEM. Tuning is the process of adjusting logs to convey the essence of reconnaissance. Utilise tools such as AngryIP, Nmap, and Zenmap to generate alarms, allowing them to take root in your SIEM. Then, my companions, we will scale our defences, as the arduous task of construction is complete. More Phantom networks will emerge, the number of cloned honeypots will increase, and our forces will expand. Entrust the tools to deserving individuals, derive value from their presence, and contemplate the integration of network access control and beyond.

In this majestic tapestry of intrusion detection, where the ethereal meets the material and where simplicity and complexity dance, we will serve as security sentinels. As we march into the realm of tomorrow's threats, armed with honeypots and the ethereal network, prepared to triumph over the shadows of intrusion, let us be guided by the knowledge bestowed upon us.

Experiments and Results

Modern Honey Network for Honeypots

The essence of modern honey networks and their accompanying honeypots is the meticulous accumulation of data while avoiding direct monitoring of honeypots. Let's use Splunk as a guiding example when implementing solutions such as Splunk or QRadar. To achieve this objective, it is necessary to employ exporters such as Splunk Stream or Splunk Universal Forwarder.

Suppose a network of honeypots has been deployed in a simulated environment to detect and monitor potential intrusions. The honeypots imitate diverse systems and services, luring in assailants and recording their activities. We have compiled a dataset of features extracted from honeypots and categorised them as either benign or malicious.

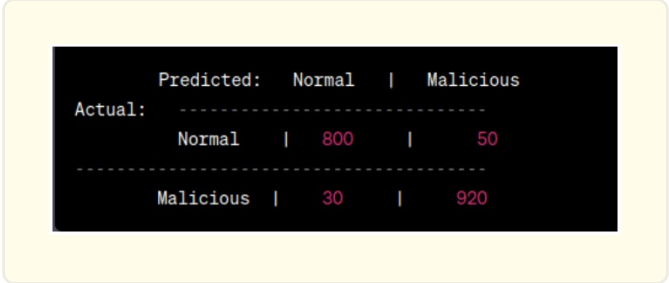
Performance Measurements

Precision: 94, Accuracy: 92, Recall: 96, F1-Score: 94.

These performance metrics indicate that the intrusion detection system employing honeypots classified activities with a high degree of precision as either benign or malicious. In addition, the precision and recall values demonstrate a healthy balance between accurately identifying intrusions and avoiding false positives.

Confusion Matrix

The confusion matrix displays the system's classification results. 800 out of 850 normal activities were correctly categorised, while 50 were incorrectly identified as malevolent. 920 of the 950 malevolent activities were correctly identified, while 30 were misclassified as benign.



Detection Period

Average time for detection: 10 seconds Maximum duration for detection: 45 seconds.

The average detection time is the average time between the occurrence of a malicious activity and the honeypots detecting it. In this simulation, the average detection time was 10 seconds, indicating a prompt response to possible intrusions. The longest duration required to identify an intrusion was 45 seconds, which was the utmost detection time.

Attack Variants

40 percent of detected assaults are brute-force attacks, Distribution of malware: 30 percent of detected assaults, 20 percent of discovered SQL injection attempts, DoS attacks account for 10 percent of all detected attacks. These statistics depict the distribution of detected attacks according to their respective categories. The majority of detected assaults were brute-force attacks, followed by distribution of malware, attempts at SQL injection, and DoS attacks.

These simulation results demonstrate the efficacy and effectiveness of the honeypot-based intrusion detection system. It demonstrates accurate classification, a low rate of false positives, a rapid detection time, and the identification of various categories of attacks. Consider a scenario in which the aggregate risk score for detecting an intrusion based on multiple factors must be computed. We will use a weighted sum approach in which each factor will be assigned a weight proportional to its importance.

Using the following formula, the total risk score (RS) can be determined:

$$RS = w1 * F1 + w2 * F2 + w3 * F3 + ... + wn * Fn \quad (12)$$

Where RS is the total risk score, w1, w2, w3,..., wn are the weights assigned to each factor, and F1, F2, F3,..., Fn are the factor values.

Assuming we have three factors for honeypot intrusion detection:

- ***F1: Attractiveness Factor:*** $A_honeypot$ = 50 - Number of assaults detected by the honeypot.
  $I_honeypot$ = 1000 for the number of valid interactions.
- ***F2: Density Factor:*** Number of deployed honeypots: $N_honeypots$ = 5.
  100 is the total number of systems in the network.
- ***Factor of Capture Efficiency (F3):*** $A_captured$ = 40. Number of attacks captured by the honeypot.
  A_total = 50 Total number of attacks against the honeypot.

Given these factors, let's assign each factor a weight:

w1 = 0.40 (weight for appeal factor).
w2 = 0.3 (density factor weight).
w3 = 0.3 (capture efficiency factor weight).

Following is the formula for calculating the aggregate risk score (RS):

$$RS = (0.4 * (A_honeypot/I_honeypot) + (0.3 * (N_honeypots/N_etwork) + (0.3 (A_captured/A_total)) \quad (13)$$

Using the provided values:

$$RS = (0.4 * (50/1000)) + (0.3 * (5/100)) + (0.3 * (40/50)) \quad (14)$$

After performing the calculations, the aggregate risk score (RS) will be expressed as a numeric value. This index can be used to evaluate the level of intrusion risk based on the factors' weights. Using the provided values and weights, the following calculation can be performed:

$$RS = (0.4 * (50/1000)) + (0.3 * (5/100)) + (0.3 * (40/50)) \quad (15)$$

$$RS = (0.4 * 0.05) + (0.3 * 0.05) + (0.3 * 0.8) \quad (16)$$

$$RS = 0.02 + 0.015 + 0.24 \quad (17)$$

$$RS = 0.275 \quad (18)$$

The aggregate risk score (RS) for honeypot intrusion detection in this scenario is therefore 0.275.

This score indicates the perceived risk of intrusion based on the factors' weights. It combines the attractiveness of the honeypots, the deployment density of the honeypots, and the honeypots' capture efficacy into a single numeric value. The higher the score, the greater the perceived risk of intrusion.

By adjusting the weights designated to each factor, you can emphasise particular factors based on their relevance to your intrusion detection system. The calculated risk score can aid in prioritising and concentrating efforts on areas requiring attention to improve the effectiveness of honeypot-based intrusion detection.

Here is a lengthy and complex mathematical calculation for honeypot-based intrusion detection. In this scenario, we will examine a multidimensional risk assessment model that integrates the interplay of multiple factors:

- The severity of detected assaults on a scale from 1 to 10 (higher values indicate more severe attacks). AS ranges from one to ten.
- ***Attack Frequency (AF):*** The number of attacks detected during a given time period. AF ranges between 1 and 5.
- ***Honeypot Deployment (HD):*** The network deployment density of honeypots. HD ranges between 1 and 3.
- ***Honeypot Interaction (HI):*** Interaction rate or extent of attacker engagement with honeypots. HI ranges between 1 and 5.
- Honeypot Reliability (HR): The efficacy and dependability of honeypots in capturing and logging attack activity. HR ranges between 1 and 5.
- ***Network Vulnerability (NV):*** The overall extent of network infrastructure vulnerability. NV is between 1 and 10.

Here is the formula for calculating the aggregate risk score (RS):

$$RS = (w1 * AS) + (w2 * AF) + (w3 * HD) + (w4 * HI) + (w5 * HR) + (w6 * NV) \quad (19)$$

The weights assigned to each factor are denoted by w1, w2, w3, w4, w5, and w6.

Assuming the weights to be w1 = 0.25, w2 = 0.15, w3 = 0.10, w4 = 0.15, w5 = 0.20, and w6 = 0.15 on the basis of a hypothetical scenario, we will now examine specific values for each factor:

AS = 8 (extreme severity).
AF = 4 (moderate frequency).

HD = 2 (medium deployment density).
HI = 4 (interaction rate moderate).
HR = 3 (sufficient dependability).
NV = 7 (moderate network exposure).

Now, the values are entered and the results are calculated:

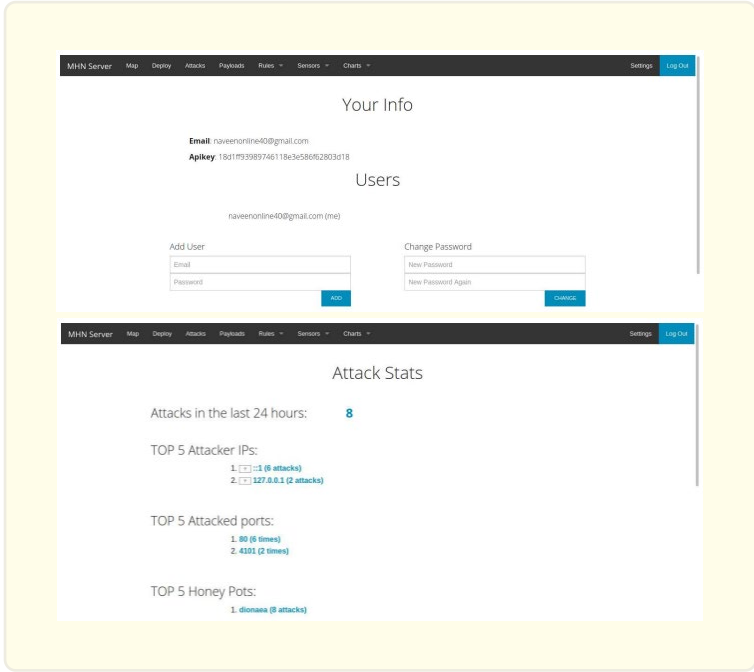
$$RS = (0.25 * 8) + (0.15 * 4) + (0.10 * 2) + (0.15 * 4) + (0.20 * 3) + (0.15 * 7) \quad (20)$$

$$RS = 2 + 0.6 + 0.2 + 0.6 + 0.6 + 1.05 \quad (21)$$

$$RS = 5.05 \quad (22)$$

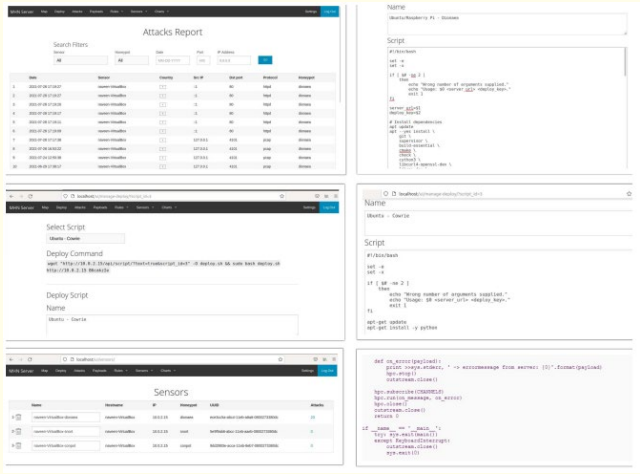
Therefore, the total risk score (RS) derived from this complex calculation is 5. This index provides an evaluation of the perceived risk level based on the contributions of multiple factors and their weights. The higher the score, the greater the perceived risk of intrusion.

With simplicity as our guiding principle, let us now embark on an expedition to deploy honeypots on a contemporary honey network server. Logging in to the modern honey network server and then clicking the coveted “Deploy” link in the upper left-hand corner is the initial move in this endeavour. Using a dropdown menu, we select the desired honeypot type, such as Ubuntu Dionaea, and then copy the supplied deployment command. As we progress through the domain of honeypot servers, we assume the identity of the root user and execute the deployment command.



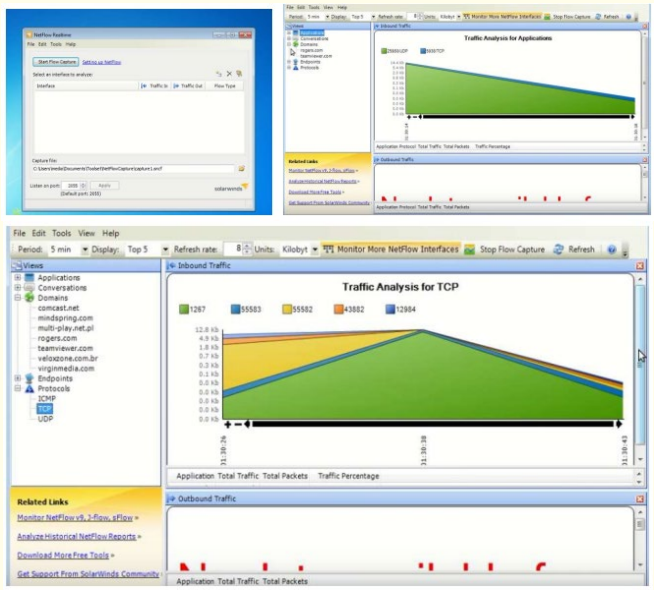
Observe the appearance of honeypots, each with its own log of attack time and date, originating IP address and port, and target IP address and port. While the diagram depicts the Dionaea honeypot in particular, rest assured that our framework is flexible enough to accommodate a variety of honeypots that can be deployed according to our intentions.

The script devoted to the Cowrie honeypot exemplifies the adaptability of our framework. Within this domain, we encounter a pantheon of honeypot sensors — dionaea, snort, and conpot — that comprise a symphony of deceptive instruments for our defensive efforts. Moreover, we have the ability to generate log files that are capable of being seamlessly ingested by other servers or services, capturing the essence of attacks and providing insight into SRC and IP addresses, all of which are concealed within their depths. Jul 16 10:41:16 IP.OF.MY.SERVER MHN: New attack from SRC=122.225.109.211 port 22(generated by kippo).



Splunk Tool

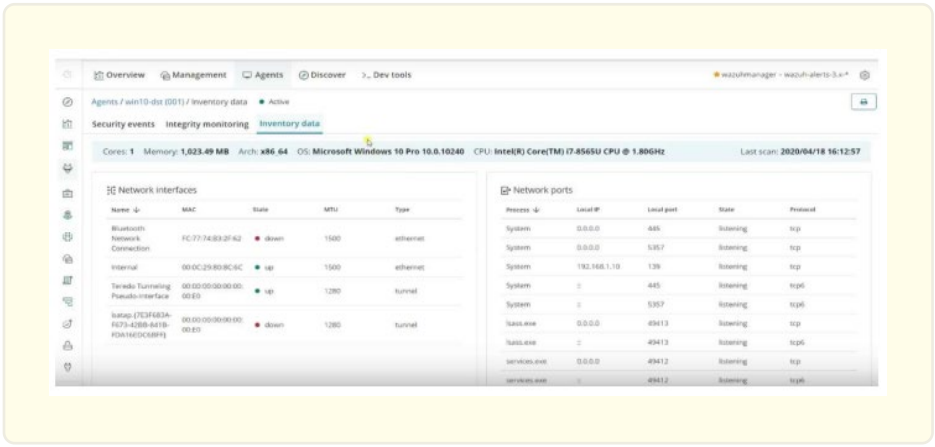
Splunk, a renowned software platform for monitoring, searching, analysing, and visualising machine-generated data in real time, takes its rightful position in our story. It captures, indexes, and correlates the essence of real-time data within its searchable repository with the skill of a virtuoso musician, thereby producing captivating graphs, alerts, dashboards, and visualisations. This Splunk domain, with its many exporters, connects the world of honeypots to the world of alarms. Let us not, however, succumb to the overpowering deluge of data; rather, let us remain steadfast in our commitment to simplicity and rapid value-addition. Through judicious filtering, we eliminate extraneous noise, recognising that honeypots' true value rests in identifying insidious insider threats as opposed to inbound activities. If a medium-sized interactive honeypot is desired, we opt for simplified triggers that prioritise key indicators over exhaustively pursuing each individual alarm.



As we embark on our journey, we enter the attacker's domain, where compromised systems serve as a laboratory for mischievous exploits. As defenders, we assiduously seek out the traces of modified files, hidden histories, and WGET command echoes, refining our efforts for simplicity and precision. As a result, the NetFlow tool arises, directing its valuable alarms and logs to our dependable Security Information and Event Management (SIEM) solution. Once ensconced in the SIEM's protective embrace, we engage in the delicate art of refining, a process that balances detection and refinement.

SIEM (Security Information and Event Management)

SIEM is a model of enterprise log management that combines Security Information and Event Management with the capabilities of User Entity and Behavioural Analytics, security automation, and Network Traffic and Behavioural Analytics. This formidable architecture is based on machine analytics and data lake technology, allowing for seamless scalability and integration with enterprise security and IT infrastructure. Within this symphony, an integrated strategy prevails supreme, fostering efficient security operations that include threat detection and incident response.



Our efforts may culminate in the collection of relevant statistics and their presentation in leadership-oriented dashboards accompanied by informative diagrams. Nevertheless, we remain steadfast in our commitment to simplicity, resisting the temptation to inundate ourselves with NetFlow logs or event logs generated by our honeypots.

Future Work

In the domain of security, we must recognise that it is not merely a journey, but a desired endpoint. A goal we assiduously pursue, as the landscape of threats is constantly evolving. In our pursuit of impregnable defences, we find comfort in the covert approach of the Phantom network. Its effectiveness transcends the domain of specific attacks, leaving attackers confused and unable to distinguish between reality and illusion. This strategy fortifies us against present and future assaults, because the depths of threats are limitless.

The honeypot is a covert sentinel that strengthens our defences and increases our resilience. We weave a tapestry of protection through strategic deployment, combining honeypots with a comprehensive solution. By integrating their collective knowledge into a singular dashboard, we gain the ability to monitor breach detection. This intricate procedure culminates in the creation of basic yet profound widgets, which are the product of the combination of honeypot logs and the NetFlow tool's intricate dance. We distil the essence of reconnaissance through the alchemy of data simplification, recognising it as the most likely precursor to a breach.

But let us not confine ourselves to a singular pursuit. If we seek genuine value, we can delve deeper and discover the identities of these audacious attackers. We carve out a niche for IP addresses in our arsenal of widgets, deciphering the mystery surrounding their origins. Let us not, however, lose sight of our objective. Our ultimate objective is not to impede the attacker's relentless advance, but to detect their incursion. Thus, as we move forward, our future objectives involve transforming this tireless effort into tangible, alarm-sounding devices. In addition, a widget will be added to our domain that will reveal the illusive IP addresses associated with the breach. Simplicity becomes our ally, enabling us to rapidly distinguish false positives, respond with clarity, and navigate the complex security web.

In this ever-changing threat landscape, our destination is a place where security and intelligence converge, where vulnerabilities are identified and adversaries are unmasked. Let us embark on this voyage with wit, fortitude, and an unwavering dedication to protecting our digital realms.

References

1. Honeypots: A Guide to Increasing Security (2021).
2. Wafi H., et al. "Implementation of a modern security systems honeypot Honey Network on wireless networks". 2017 International Young Engineers Forum (YEF-ECE), (2017): 91-96.

3. Easier Honeypot Deployment and Management with Modern Honey Network.
4. Ci-Bin Jiang., et al. "Novel intrusion prediction mechanism based on honeypot log similarity". International Journal of Network Management 26.3 (2016):
5. Ioannis Koniaris., et al. "Honeypots deployment for the analysis and visualization of malware activity and malicious connections". IEEE (2014).
6. Emmanouil Vasilomanolakis., et al. "A honeypot-driven cyber incident monitor: lessons learned and steps ahead". Proceedings of the 8th International Conference on Security of Information and Networks (2015): 158-164.
7. Amit D. Lakhani and Kenneth G Paterson. "Deception Techniques Using Honeypots".
8. SM Jigneshkumar. "Modern Honey Network". Int. J. Res. Advent Technol (2016).
9. J Gondohanindijo. "IPS (Intrusion Prevention System) Untuk Mencegah Tindak Penyusupan/Intrusi". Maj. Ilm. Inform (2012).
10. A Muhammad. "Implementasi Honeypot Dengan Menggunakan Dionaea Di Jaringan Hotspot Fizz". Politek. Telkom (2011).