JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

MULTI-SERVICE HONEYPOT: COMPREHENSIVE NETWORK SECURITY MONITORING AND DECEPTION FRAMEWORK

K. Sharath Kumar[1], Reddyvari Venkateswara Reddy[2], Bathula Chandana[3], Shrutika Shrikhande[4], Tadicherla Deva Kumar[5],

[1]Assistant Professor, Department of CSE (Cyber Security), CMR College Of Engineering & Technology, Hyderabad, Telangana, India.
[2] Associate Professor, Department of CSE (Cyber Security), CMR College Of Engineering & Technology, Hyderabad, Telangana, India.
[3,4,5]B. Tech Student, Department of CSE (Cybersecurity), CMR College Of Engineering & Technology, Hyderabad, Telangana, India.

Abstract: The field of cybersecurity faces a never-ending barrage of new attacks. Defenders usually concentrate on strengthening their perimeter defenses and don't pay attention to the attackers' evolving strategies. Examining current approaches for using honeynets to detect and prevent cyberattacks is the primary objective of this field's research. There are two that are especially fascinating techniques that focus on demonstrating empirically that honeypots—more precisely, Multi-service Honeypots—offer a proactive defense. The idea behind honeypots and their variations, such as Multi-service Honeypot, is to mimic a controlled environment to be able to entice an attacker and provide important details regarding any security holes in the system. Security experts can strengthen the defenses defending their company's IT systems by keeping an eye on the tactics, tools, and procedures (TTPs) that attackers utilize in the regulated honeypot configuration and networks from possible online dangers. Under the pretense of purposeful deceit, honeypot's function. They give the impression of being trustworthy systems, complete with information or weaknesses that draw in bad actors. But in reality, they have a secret goal: to learn more about threat actors that interact with the honeypot. When threat actors use the honeypot in this way, they unintentionally give the defenders recorded interactions that provide crucial details regarding the attack tactics, tools, and procedures they employ.

***IndexTerms* – Attacks, defenders, perimeter defenses, honeynets, honeypots, multi-service honeypots, tactics, tools, and procedures (TTPs), deceit, threat actors.**

I. INTRODUCTION

Data security and network system security have recently drawn increased attention. Network systems must be secured against hackers since they house crucial data and resources. Honeypots and honeynets are commonly used by security professionals to safeguard network infrastructures. Honeypots are another tool used by security professionals to pick up new hacking skills from fresh attackers and intruders. A honeypot: what is it? The Honeynet Project is credited to Spitzner, who described a honeypot as "A protection asset whose worth depends on being examined, assaulted, or destroyed" in 2002. "A data system resource that is valuable only when used illegally or uninvited source," was the more expansive definition provided by Spitzner a year later. To put it practically, a device known as a honeypot is meticulously laid out to feast on a connection to draw in an uninvited attacker.

The world of parked automobiles and trench coats has given way to a network and program playground where competing countries and companies scour the landscape in search of that fleeting opening. Too often marketed as "security solutions." The objective is straightforward and admirable: to cease operations and, most importantly, to get rid of the requirements for deception stratagem appellations. One, cunning is cunning, a term that originates from a project carried out in the early 1990s by a collection of astute researchers who were eager to put an opponent in an artificially confined and constrained space, with their opening countered by force, regardless of the ultimate result, which they refused to acknowledge in the air quotes: decoy, act, bait; whatever it took to create evidentiary fact, in the form of a certification of the enemy's strategy and execution.

This study looks at the honeypot space. The fundamental ideas of the subject are examined, in addition to the characteristics that set it apart and several security-related study topics that could be used as models for honeypot studies. The dirty system finds many appealing targets in complex systems. Security analysts benefit much from the examination of attacker interactions with honeypots. They can gather intelligence on emerging threats, zero-day vulnerabilities, and the greatest recent attacker Tactics, Techniques, and Procedures (TTPs) by closely analyzing attacker activity.

The landscape of honeypots varies greatly in complexity. Low-interaction honeypots provide a targeted view of attacker activity restricted to a single service. They are expertly designed to mimic specific services (like a web server or secure shell login). High-interaction honeypots, conversely, offer a wider perspective. They entice the attacker in by seeming like whole operating systems, which reveals their entire toolkit of techniques and goals. This article will go over the benefits and drawbacks of each type of honeypot in detail to assist security professionals in choosing the best instrument for their unique security objectives. The setup and planning of a honeypot deployment must be just as thorough.

II. LITERATURE REVIEW

According to Paul. A.J. et. al.'s (2007), research work primarily utilizes this infrastructure as a service for cloud computing security [1]. They concentrate on key skills and present chances for capital cost offerings. This research aims to apply Honey Pot as a cloud environment security technique.[2] "Intrusion detection and prevention using honeypot," Rajbhar, Vivekanand. An adaptive honeypot setup, deployment, and maintenance technique were published by Fraunholz, Daniel, Hans D. Schotten, and Marc Zimmermann in the International Journal of Advanced Research in Computer Science 9.4 (2018) [3].

The International Conference on Advanced Communication Technology (ICACT) is on its nineteenth anniversary, in 2017. The article "Flexible Honeypot Deployment for Invasion Detection" by K. R. Sekar et al. In 2018, IEEE hosted the Second International Conference on Electronics and Informatica (ICOEI) [4].

"An IDPS system using honeypot strategy in sensor networks "IEEE,2018 International Conference on Electrical, Electronics, Computer, and Control Engineering (ICCCEEE). This research reports on the effectiveness of a honeypot Uniform Resource Locator (URL) system inspired by the Human Interaction device (HID) system that was built employing Python to access a URL. Its goal is to offer a more rapid and effective substitute for the HID system-developed honeypot URL mechanism for identifying harmful web URLs [5]. It is significant to note that an analyst needs several hours or even days to respond to the HID system because the phony URL system can only be used by humans.

However, the developed URL system is more efficient because it allows an analyst to respond in a matter of seconds. They are using a honeypot system to find harmful web pages. In [5], the authors designed a honeypot method to identify malicious web URLs. Python is used to build the system on the client side. A client-side crawler assembles the URL addresses that are subsequently utilized to view websites after joining.

The signature-based intrusion detection system sets off a trigger when these URL addresses are malevolent or contain vulnerabilities. Security is therefore provided because the dangerous URL addresses are kept on the blacklist [6].

Koniaris et al. only use honeypot systems for the study and visualization of malicious activities and connections. Two alternate search honeypots have been established in their applied practice. In a different work, Song Li et al. discussed the mixed interaction honeypot-based intrusion detection system they have a stable network and security More to enhance were developed [7]. The primary type is classified based on their self-propagation—adversarial software for their purposes. The second type is honeypots that are supervised to reach systems traps for malicious activities [8].

A distributed honeypot system has been proposed by Chawda et al. to look for new vulnerabilities. Low interaction honeypot systems were employed in their system to expose it to additional vulnerabilities as a frontend content filter [9]. Experts like Xiangfeng Suo et al. and colleagues have discussed the application of honeypot technology in intrusion detection systems. They have proposed using honeypot systems in research to eliminate intrusion detection system issues [10].

A computer network security signature generator based on honeypots has been implemented by Paul et al. In particular, the created approach has been utilized to defend against attacks by polymorphic worms. Additionally, the created system is capable of isolating suspicious traffic and gathering valuable information regarding hostile traffic and worm attacks [11].

In their study, Riboldi et al. created a low-interaction honeypot system to track illicit activity on Voice Over Internet Protocol (VoIP) systems. About Session Initiation Protocol (SIP) 3502-port number, over 92 days on the mechanism, whose effectiveness has been questioned tracked things that have been compiled. They have understood their system to be accessible, much like firewalls and VoIP environments with intrusion detection systems [12].

III. INTERACTION LEVELS

Honeypots are essential elements of the cybersecurity environment that fall into three categories: low, medium, and high interaction with a combined type of honeypot i.e. Hybrid honeypot. These levels indicate how involved and efficient the honeypot is in communicating with potential attackers.

1. Low-Level interaction Honeypot: Minimal engagement with the attacker is provided by low-interaction honeypots. Low-interaction honeypots only mimic the services and protocols that are most commonly targeted by attackers, such as Telnet, File Transfer Protocol (FTP), Secure Shell (SSH), Post Office Protocol (POP3), Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), Simple Mail Transfer Protocol (SMTP), Virtual Network Computing(VNC) protocol. Installing, maintaining, and growing low-interaction honeypots is typically simpler than installing, managing, and growing middle- and high-interaction honeypots, It makes them accessible choices for businesses with limited funding. High-interaction honeypots, however, might provide more details and insight into the attacker's activities. The Simple and Lightweight Honeypot system Honeytrap Honeypot, which mimics a single susceptible service or protocol, is an illustration of a low-interaction honeypot. Usually, honeytrap is used in a restricted and targeted way, concentrating on particular protocols or services like SSH, FTP, or Telnet.

2. Medium-Level interaction Honeypot: Medium-interaction honeypots are not static; instead, they attempt to mimic particular services or protocols to provide potential attackers with something "useful." Even so, they pose a greater risk than low-interaction honeypots, they offer more comprehensive gathering and monitoring capabilities that help businesses understand not just what attackers are attempting to accomplish, but also their motivations. A nice illustration of a medium-interaction honeypot is the Cowrie Honeypot. It is an open-source, modular honeypot system that mimics FreeBSD and Linux. Among other things, it can record file uploads, command histories, and keystrokes. Put simply, there's no excuse for not keeping a watch on it.

3. High-level interaction Honeypot: The most thorough and accurate replica of a target system is offered by a high-level honeypot, making it challenging for the adversary to distinguish them from real production systems. These honeypots are created to be installed throughout the network architecture of an organization to be able to gather a variety of threat intelligence. High-level honeypots are more dangerous than medium- and low-level honeypots, even when they are usually placed in a controlled, isolated area to lessen the possibility that an adversary will access the real systems. The versatile and potent Honeyd Honeypot is an illustration of a high-level honeypot; it can simulate a vast array of operating systems, services, and network topologies. Because Honeyd can simulate intricate network settings, it's helpful for companies looking to improve their security posture.

4. Hybrid Honeypot: A hybrid honeypot mimics the appearance and feel of a real production network or system. Because it uses real data and assets—such as user accounts, files, databases, etc.—it is almost precisely like the actual item. The main goal of a pure honeypot is to attract invasive behavior to observe its instruments, techniques, and tactics without putting any real production systems in jeopardy. Typically, pure honeypots are located in a carefully regulated environment, isolated from the main network. Any behavior detected within the honeypot is thus quarantined and kept apart from any operational systems. They usually contain a variety of sensors and monitoring devices to gather data on any behavior of the honeypot. such as system records, network traffic, and attempted breaches. Better incident response plans, intrusion systems, and security measures can be made with the usage of this information. However, maintaining a pure honey pot can be difficult and resource-intensive because it needs ongoing care and attention to be effective.

IV. Limitation of existing solutions

It's imperative that you consider these limitations and use a range of honeypot techniques and security measures to be able to enhance your overall protection strategy. These limitations show how important it is to combine user-friendliness and accurate functionality to get the best security results:

1. Pre-defined Template Constraints: Certainly, Pre-configured templates fail to comply with many specialized security needs. Implementing some complex honeypot scenarios that need more detailed configurations, or require interactions that are not accessible as part of the templates themselves, could prove challenging.
2. Limited-service Scope: The pre-defined template may concentrate on services that are frequently targeted, such as SSH and web servers. These online resources are beneficial for broad security monitoring, but they might not be appropriate for companies looking to honeypot less popular services or particular industry-specific apps.
3. Scalability Challenges: Scaling a user-centric honeypot platform may serve as a tricky process. While there are distributed deployment options with some platforms, managing multiple honeypots across a network -- and analyzing the data -- can be unwieldy compared to simpler setups.
4. False Positives and Workload: Additionally, false positives from integrated analysis technologies can necessitate human research to distinguish between genuine threats and benign activity. For a security team, that may get incredibly labor-intensive, especially particularly when paired with everything else data that a traditional honeypot might produce.
5. Dependency on Third-Party Tools: The honeypot relies on third-party tools such as Network Mapper (Nmap) for port scanning and WEBrick for (Hypertext Transfer Protocol) HTTP server emulation. Any issues or limitations with these tools could potentially impact the overall functionality and effectiveness of the honeypot.

V. Methodology

1. Planning

Strategic planning is the first step in putting this website architecture into action. The following things happen at this phase: Select whether to install the honeypot on the website's host server or another server, and whether to employ a hybrid, high-interaction, or low-interaction honeypot. They are splitting the CPU, RAM, storage, and network bandwidth required for the honeypot and choosing a suitable tool for monitoring and analysis to enable efficient activity observation and assessment within the honeypot.

2. Deployment

Installing and setting up the honeypot on the server of the hosting website or, according to the host's requirements on an alternative server. Next, just configure the resources that are needed to keep the honeypot operational. Next, test the project to guarantee that it performs as needed and can capture the attacker or harmful behavior without interfering with the legitimate users' activities.

3. Monitoring

During this phase, the deployed honeypot must be constantly observed. Regular observation of the group honeypot is necessary; if any activity occurs, it must be investigated and given attention. Using a range of monitoring technologies in place of manual monitoring to keep a watch on the honeypot's activity in real-time, such as log analysis tools, Intrusion Detection systems (IDS), Security Information and Event Management (SIEM) systems, etc. The monitoring and analysis tools that were selected and configured during the planning phase must be installed and configured to properly track and assess honeypot activities. To guarantee that you are informed right away of any suspicious or malicious activity that the honeypot detects, set up alerts and notifications. Set up the rules and monitoring tools appropriately.

4. Analysis

Understanding the type and extent of the observed activity as well as locating the channels via which attackers communicate with the honeypot are facilitated by parsing the collected data and logs from the device. It is feasible to identify specific threat actors or current attack operations by looking for patterns of the attacker. Indicators Of compromise (IOCs), such as(Internet Protocol) IP addresses, Hashes, and invasive payloads can also be obtained by examining the analyzed data. Threat information may then be supported and proactive security measures directed with additionally these IOCs.

5. Response

Depending on the application of the honeypot being used, the way one responds to its behavior should change. It would be good to just block the attacker's IP address if a low-interaction honeypot is being deployed. However, it might be able to communicate with the attacker and learn their objective if a high-interaction honeypot is being employed.

VI. Implementation

In this paper, architecture had three main types of options they are geolocation finder, website cloning, and camera phishing. Here the users can opt/select an option for further process. It is a completely web-based project that offers different types and also provides customization for various factors.

1. Honeypot Installation: The Honeypot object is created at the start of the code and acts as the main hub for controlling various features. It contains instructions for setting up, activating, and monitoring many facets of Honeypot.

2. Menu System: The menu system has been designed to offer an interface that is easy to use allowing the user to select what to do by inputting the corresponding numeric option. It is modular allowing it to seamlessly engage with the various features. This object is used as a central management point and encapsulates methods for configuring, starting, and managing different characteristics of the honeypot.

3. Selection for Specific Configuration: The layout demonstrates how simple it is to use each component individually. It offers "several configurations, e.g., Fast Auto-Config, Manual Configuration. Fast Auto-Configuration allows for real-time setup that applies predefined configurations, specially designed for fast deployment. Any settings that are likely to remain constant, such as port configurations, response messages, and logging (etc.), may be defined to save time at setup. Manual Config lets advanced users customize many settings to taste."

4. Examine service emulation functionalities: Accepted protocols of the framework are (Hypertext Transfer Protocol) HTTP Server, (Transmission Control Protocol)TCP Server, HTTP Emulation, and (Secure Shell) SSH Emulation. Utilizing the WEBrick library and the HTTP Server option, you can build a basic web server that is always open on port 8000. Making efforts to entice possible attackers and monitor their activities, primarily shows how honeypots can be configured to mimic internet services. In addition, using the Transmission Control Protocol (TCP) Server option causes a TCP server to start listening on port 20, accept connections, and reply with a brief message. However, in contrast, the HTTP Finally, an emulated SSH server is launched via SSH Emulation, listening on port 22 and displaying a 2.8 OpenSSH fake banner. To track unwanted login attempts, offer information about possible intrusion attempts, and identify structures in exploit usage, this was arranged to fake SSH services.

5. Investigate gathered Information: Once the Honeypot system is up and running, the gathered data may be viewed through the menu system. This is all available for further analysis and Some threat intelligence. By using this extensive framework, you will have access to a variety of features to improve your network security and threat detection skills while customizing and deploying the Honeypot system to your exact requirements.

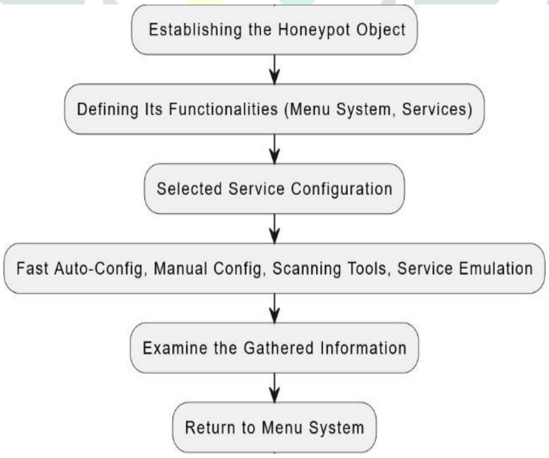
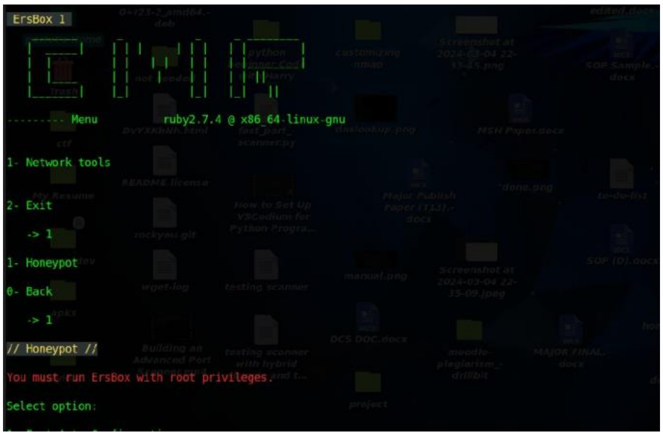

Establishing the Honeypot Object
Defining Its Functionalities (Menu System, Services)
Selected Service Configuration
Fast Auto-Config, Manual Config, Scanning Tools, Service Emulation
Examine the Gathered Information
Return to Menu System

Fig.1 Process of Multi-Service Honeypot

VII. Result and discussion

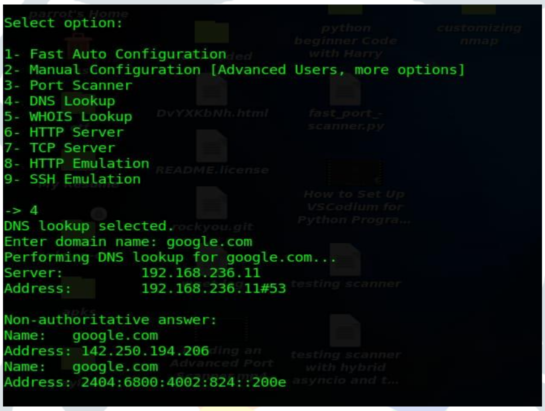
By providing insights into attacker behavior and offering a mechanism for early threat detection, honeypots can significantly enhance your overall security posture.

The proposed modular honeypot platform represents a significant step forward in the field of deception-based security solutions. It greatly facilitates deployment, builds in analysis tools, and enables security teams to quickly and easily identify attacker patterns. This deeper understanding of attacker behavior is what will finally enable security teams to get one step ahead of cyber threats rather than remain in a reactive mode. Further work may involve integrating the platform with existing security systems, advanced threat intelligence feeds, and machine learning-fed anomaly detection to broaden the system’s overall situational awareness and threat analysis capabilities. This crucial tool will continue to be relevant in the ongoing fight against cybercrime thanks to its ongoing evolution.



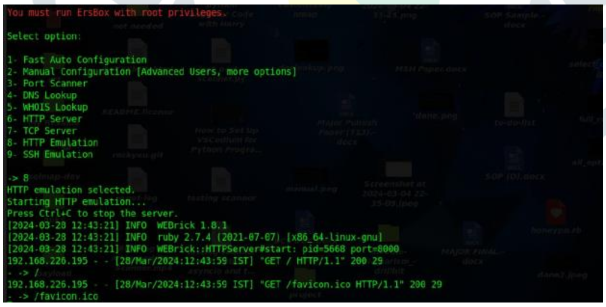
```
ErsBox 1

--------- Menu          ruby2.7.4 @ x86_64-linux-gnu

1- Network tools

2- Exit

   -> 1

1- Honeypot

0- Back

   -> 1

// Honeypot //

You must run ErsBox with root privileges.

Select option:
```

Fig.2 Multi-Service Honeypot Interface



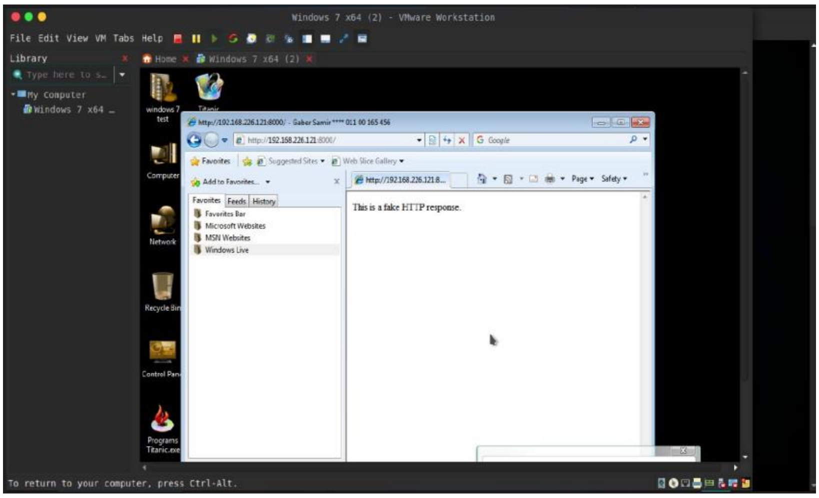
```
Select option:

1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]
3- Port Scanner
4- DNS Lookup
5- WHOIS Lookup
6- HTTP Server
7- TCP Server
8- HTTP Emulation
9- SSH Emulation

-> 4
DNS lookup selected.
Enter domain name: google.com
Performing DNS lookup for google.com...
Server:         192.168.236.11
Address:        192.168.236.11#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.194.206
Name:   google.com
Address: 2404:6800:4002:824::200e
```

Fig.3 Specific Configurations



```
You must run ErsBox with root privileges.

Select option:

1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]
3- Port Scanner
4- DNS Lookup
5- WHOIS Lookup
6- HTTP Server
7- TCP Server
8- HTTP Emulation
9- SSH Emulation

-> 8
HTTP emulation selected.
Starting HTTP emulation...
Press Ctrl+C to stop the server.
[2024-03-28 12:43:21] INFO  WEBrick 1.8.1
[2024-03-28 12:43:21] INFO  ruby 2.7.4 (2021-07-07) [x86_64-linux-gnu]
[2024-03-28 12:43:21] INFO  WEBrick::HTTPServer#start: pid=5668 port=8000
192.168.226.195 - - [28/Mar/2024:12:43:59 IST] "GET / HTTP/1.1" 200 29
- -> /
192.168.226.195 - - [28/Mar/2024:12:43:59 IST] "GET /favicon.ico HTTP/1.1" 200 29
- -> /favicon.ico
```

Fig3. HTTP Emulation from Honeypot Side



This is a fake HTTP response.

Fig.4 HTTP Emulation from Attacker Side

VIII. Acknowledgment

The Author is grateful to the CMR College of Engineering & Technology for providing better facilities and practical requirements.

References

[1] Mohammadzad M. et al. published a paper titled "Using Rootkits Hiding Techniques to Conceal Honeypot Functionality" in the Network and Computer Applications Journal, 2023.

[2] Angelo Dell’Aera, 2022 Angelo Dell’Aera M., libemu - x86 emulation and shellcode detection, 2022.

[3] M. Umer Altaf and M. Shafique, “Honeypots: Concepts, mechanisms, and potential”, Computers & Security, vol. 103, pp. 17-34, 2021.

[4] Mariconti, E., Onaolapo, J., Andriotis, P., Kastania, A., & Stringhini, G. (2017). It's a trap: Emperor Honeypot strikes back. USENIX Security Symposium (pp. 665-682).

[5] Iuga, C., Nurse, J. R. C., and Erola, A. (2016). Baiting the hook: factors impacting susceptibility to phishing attacks. Hum. Cent. Comput. Inf. Sci. 6, 8. doi:10.1186/s13673-016-0065-2.

[6] Wilson, T., Maimon, D., Sobesto, B., & Cukier, M. (2015). The effect of a surveillance banner in an attacked computer system: Additional evidence for the relevance of restrictive deterrence in cyberspace. Journal of Research in Crime and Delinquency.

[7] Virvilis, N., Serrano, O. S., & Vanautgaerden, B. (2014). Modifying the scenario: The skill of tricking crafty assailants. NATO CCD COE Publications.

[8] Kaur, T., Malhotra, V., & Singh, D. D. (2014). Comparison of various instruments for network security. honeypot and firewall intrusion detection system.

[9] Stiawan, D., Abdullah, A. H., & Idris, M. Y. (2011). Characterizing network intrusion prevention system. The International Journal of Computer Applications (0975–8887).

[10] Akkaya & Thalgott, F. (2010). Honeypots in network security.

[11] L. Spitzner, “Honeypots: Catching the Insider Threat”, IEEE Security & Privacy, vol. 1, no. 2, pp. 15-23, 2003.

[12] N. Provos, “A Virtual Honeypot Framework”, The actions of the 1st Workshop on Virtualization in Dependable Systems, VIDS ’05, pp. 1-10, 2005.

[13] Weiler, N. (2002). Honeypots for distributed denial of service attacks. IEEE Computer Society,109-114.