Computer and Information Science; Vol. 16, No. 4; 2023
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Malware Investigation and Analysis for Cyber Threat Intelligence: *A Case Study of Flubot Malware*

Uchenna J. Nzenwata[1], Frank Uchendu[2], Haruna Ismail[3], Eluwa M. Jumoke[2], & Himikaiye O. Johnson[1]

[1] School of Computing and Engineering Sciences, Babcock University, Ogun State, Nigeria

[2] School of Computing and Information Security Studies, Salford University, Manchester, UK

[3]Silesian University of Technology, Gliwice, Poland, Europe

Correspondence: Uchenna J. Nzenwata, School of Computing and Engineering Sciences, Babcock University, Ogun State, Nigeria.

Received: September 19, 2023 Accepted: November 10, 2023 Online Published: November 29, 2023

doi:10.5539/cis.v16n4p47 URL: https://doi.org/10.5539/cis.v16n4p47

Abstract

Android operating systems have swiftly outpaced other operating systems (OS) in popularity, making them vulnerable to assaults since hackers are continuously looking for flaws to exploit. This is why several organisations have long been plagued by various types of mobile security threats. Utilizing a cyber-threat intelligence tool to evaluate, track, and prevent planned attacks is one crucial strategy to combat this effect. This paper discusses and investigates the FluBot malware, using the Dagah tool and Android Studio to phish, harvest and exploit malicious applications over SMS on Android devices. The Capability Maturity Model (CMM) was adopted and used for the investigation. The methodology adopted describes the operation of the FluBot malware through a cloned website, and demonstrates how FluBot is used to share a malicious link through the short message service (SMS), which is then used to grab a victim's credentials. The outcome of the study displayed the information on the FluBot malware, including its source, domain, and destination. Similar malware analysis and assessments of cyber threat intelligence may be conducted using the techniques used in this study.

Keywords: low case, comma, paper template, abstract, keywords, introduction

1. Introduction

Advanced Persistent Threats (APTs) are becoming more frequent in today's world and it is getting harder to secure wireless networks and private files as hackers always come up with new ways to steal data. Since Android is the most used smartphone operating system worldwide, it is the mobile operating system that gets targeted the most (Garg & Baliyan, 2021). Short Message Services (SMS) are delivered to iPhones and Android smartphones by a malicious program named Flubot (Salsabila, Mardhiyah & Hadiprakoso, 2022). The Flubot SMS messages come in a broad range of formats, and scammers regularly change them, (Blázquez & Tapiador, 2023). To find out what kind of malware the Flubot is, questions are being posed. Chapin, Piscitello and Strutt (2022), found that Flubot is an Android malware actively spreading over SMS, collecting passwords, online banking information, and other sensitive information from affected smartphones world-wide. The primary mode of Flubot dissemination, according to the study in (Van Haastrecht et al., 2021), is text message notifications. These notifications urge consumers to download a security update or an app. The program asks for various permissions (such as SMS, call, contact permissions, and many more) during installation that essentially gives it power over the device.

1.1 Android OS

The Android Operating System (OS) was created by Andy Rublin, Rich Miner, Nick Sears, and Chris White for Android Inc. in October 2003 (Callaham, 2018). The OS, based on Linux-Kernel and created for smartphones and tablets, is open source and source code released by Google under an Apache licence. Its system architecture is made up of four main layers which includes: Application Layer, Framework Layer, Middleware Layer and Kernel Core layer (Meng et al., 2018). Each of the layers contains the specifics as illustrated in Figure 1.

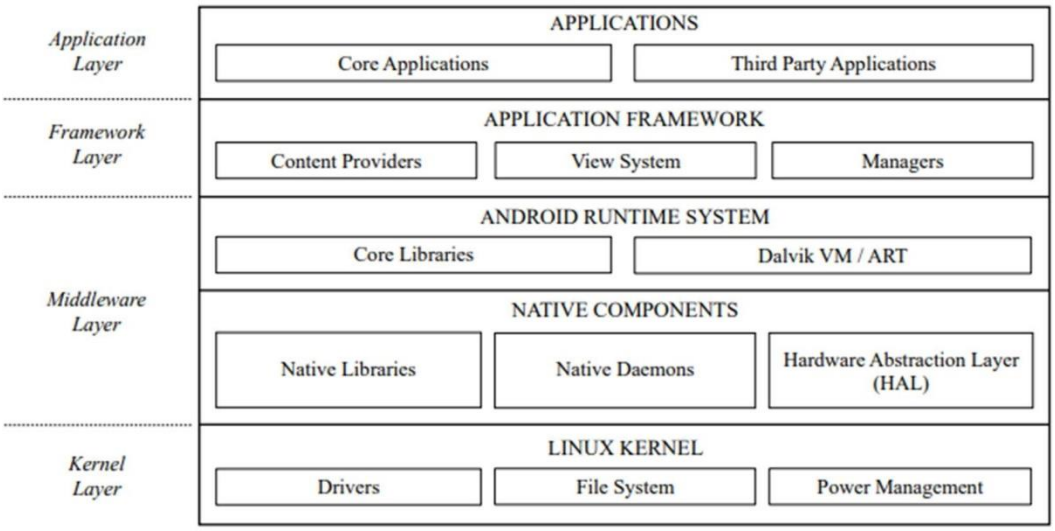

Figure 1. A Layered System Architecture of the Android OS (Meng et al., 2018)

Android quickly surpassed other operating systems in popularity, rendering it vulnerable to attacks because hackers are constantly on the lookout for weaknesses to exploit. The fact that several suppliers offer services that are marketed without well-established security measures makes defending the Android OS the most challenging issue.

1.1.1 Vulnerabilities And Security Issues Associated with The Android OS

Some of the known vulnerabilities and security issues associated with the Android OS includes the following as stated in (Özdemir and Zaim, 2021), which is summarized in Figure 2.

i. Denial of Service (DOS): This prohibits users from accessing the target system and prevents the target system from offering services.

ii. Code Execution: This happens when an attacker inserts malicious code into a string or file that is then used by the software to perform its operations.

iii. Overflow: Sequential data of the int and char types are stored in memory by buffers. When the variables of a program made up of flawed functions store more data than they can hold, it results in a buffer overflow.

iv. Gain Information: This happens when useful data about the target system is obtained during the attack phase and made easily accessible if it is in the public domain. The majority of it is completed using a tool for information gathering.

v. Gain Privilege: This is the process where the attacker searches for vulnerabilities discovered while gaining information and then exploits those vulnerabilities to get user rights.

Attack type/ using vulnerability	Android Vulnerabilities					
Attack types	DOS	Code Execution	Overflow	Bypass	Gain information	Gain privileges
Remote Attacks	✓				✓	✓
Client-Side Attacks		✓			✓	✓
Attacks Using Malicious Apps	✓	✓	✓	✓	✓	✓
Mobile Post Exploits	✓	✓	✓	✓	✓	✓

Figure 2. Android vulnerabilities/security issues with its attack type (Özdemir and Zaim, 2021)

Other related security concerns of the Android OS include: Version fragmentation, Rooting, Google Play malware, insecure apps, lack of hardware data encryption, spyware, data leaks and SMShing.

1.2 Flubot

Flubot is thought to have originated from Spain and was first discovered in December 2020 as shown in Figure 3. (Threatfabric, n.d). A report by a cybersecurity firm ThreatFabric, claims the malware is disseminated through phishing assaults, in which attackers send messages (smishing) to potential victims that contain dangerous links (Threatfabric, n.d). Clicking this link compromises the device thus, grabbing the credentials and other personal identifiable information (PII) of the victim by the attacker. Flubot's agents have a variety of motives, including monetary gains, development of botnets, undercover activities, information gathering and social engineering. There have been a number of fraudulent Short Message Service (SMS) campaigns between the end of 2020 and the beginning of 2021 that announced the arrival of a package while posing as different logistics companies, such as FedEx, DHL, or Correos. Recipients were invited to download an app on their mobile device in order to find out where the package is (Liu et al., 2021). In terms of the malicious code's functionality, once the user installs the application on their device, it begins to track the identifiers of all the applications it starts and is capable of injecting superimposed pages when it detects a session log-in in one of the target applications, so the user believes they are entering their credentials on the original website when, in reality, they are sending them to the command-and-control server (C2) controlled by the attacker. To avoid detection and analysis, the malware employs code injection, code obfuscation, and encryption. It poses a serious threat to Android users since it can spread to other devices via SMS messaging (Mayrhofer et al., 2021). Figure 4, shows the FluBot propagation pattern.

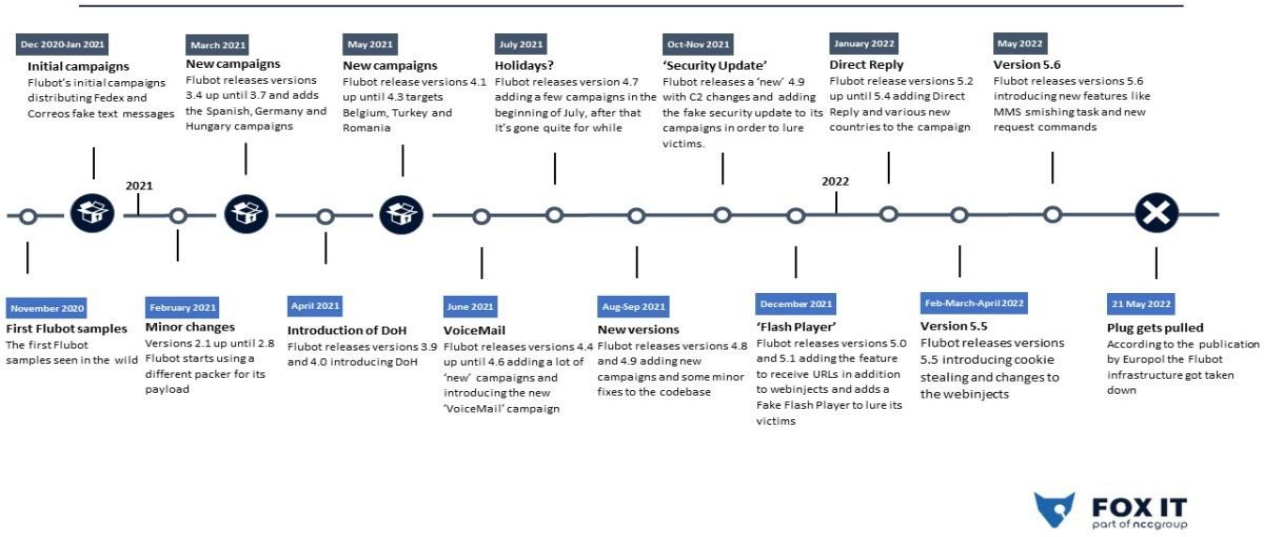

Figure 3. Evolution of the FluBot Malware (Fernick, 2022)

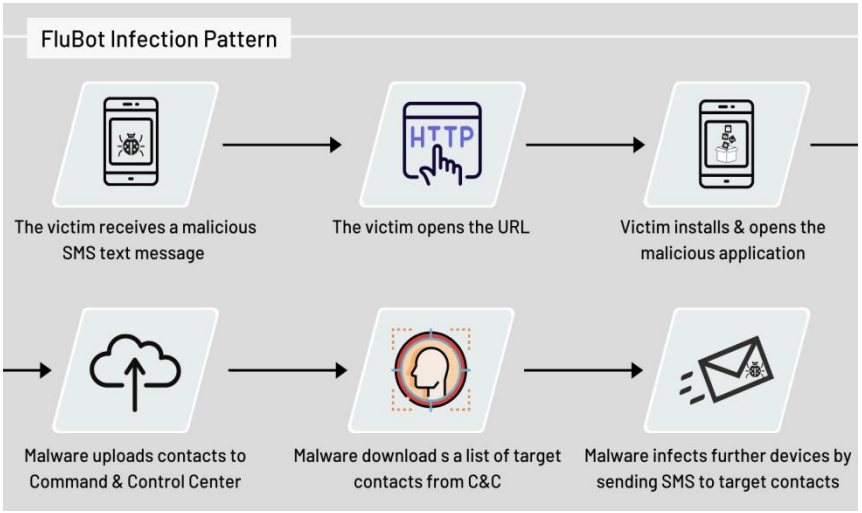

Figure 4. FluBot infection and propagation pattern (Gibbs, 2021)

2. Literature Review

FluBot has been the subject of some studies, including malware comprehension and analysis. A study was carried out in (Garcá-Teodoro, Gómez-Hernández and Abellán-Galera, 2022), where three unknown malware samples were analysed. It was identified from the codes that these samples are FluBot malware. From the study in (Garcá-Teodoro et al., 2022) and shown in Figure 5, FluBot was also referred as Fedex Banker or Cabassous. According to the studies done by the Swiss company PRODAFT (Mogicat & Zermin, n.d), it is possible that FluBot infected over 60,000 terminals and listed over eleven million phone numbers, which is equal to 25% of the population of Spain.

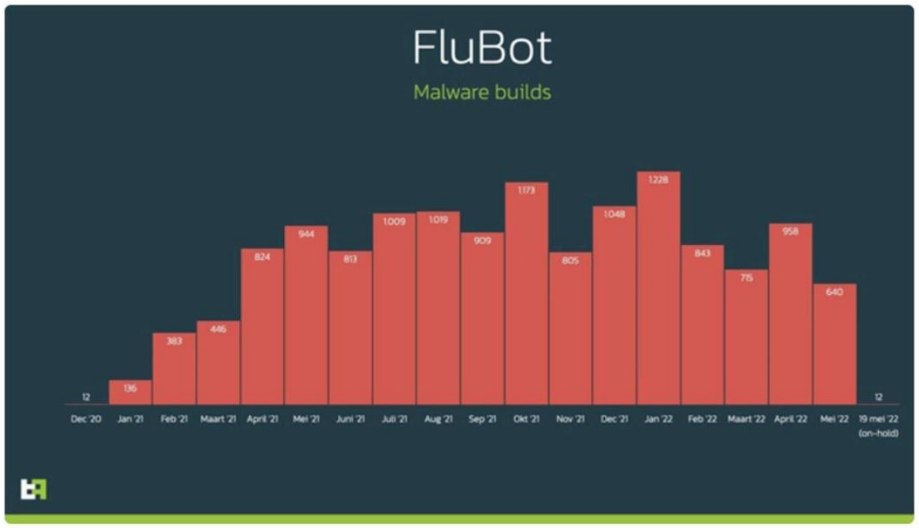

Figure 5. Graphical representation of FluBot malware spread from December 2020 to May 2022 (Threatfabric, n.d).

Android is now the most popular mobile operating system, accounting for 43.43% of the market (Riasat, Batool and Iqbal, 2022) and 70.93% of the global market share worldwide as at March 2023 by (StatCounter, 2022).

The ability to simply build and submit programmes to the official store (Google Play) not only attracts developers, but it also boosts the number of new users of this platform. Because of its popularity and market dominance, Android is frequently attacked by rogue applications. While Google claims to have eliminated up to 1.2 million dangerous apps security experts and threat intelligence firms continue to discover malicious malware disguised as legitimate programs.

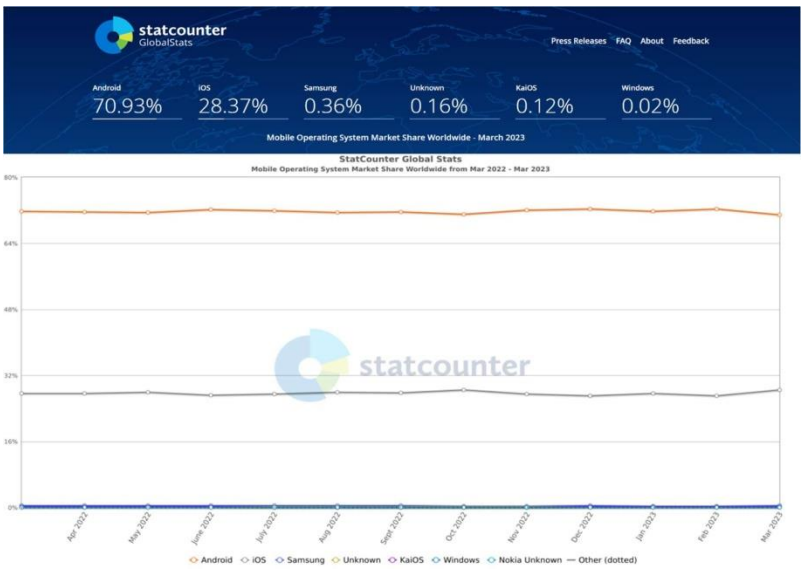

Figure 6. Graphical representation of Mobile OS Market Share Worldwide from April 2022 to March 2023 (StatCounter, 2022).

2.1 Scope and Limitation

The investigation of a mobile security threat at ABC organisation using Dagah for exploitation and Android Studio for Android device simulation, as well as carrying out a threat intelligence assessment to protect data leakage, secure wireless network communication, malware, and malicious programme propagation, is the focus of this paper. The study does not consider all types of mobile security threats; instead, it concentrates on a specific Trojan for Android devices named FluBot. The two observed limitations of this study are the usage of an android emulator in place of an actual android smartphone and Bitly's refusal to shorten URLs even after we successfully generated our access token.

2.2 Related Tools

The present ecosystem of Android tools contains various frameworks aside the Dagah tool that are intended to carry out further specialised analytic tasks. The DroidBox (Chaurasia, 2015) is used to perform dynamic analysis of Android. Another tool is the ConDroid (Schütte, Fedler and Titze, 2015), which is used execute specific code locations with no app manual interaction. For the Network analysis, the Wireshark (Ndatinya et al., 2015) is a good dynamic tool.

3. Methodology

The model considered and implemented for this investigation is the Capability Maturity Model (CMM). There are two levels of CMM and its implementation to this investigation: Threat intelligence collection capability and threat intelligence integration and dissemination. The CMM was used in this study because it has a well-defined and efficient processes, which are crucial for detecting, analyzing, and mitigating threats effectively.is a good tool for malware analysis.

3.1 Level 1: Threat Intelligence Collection Capability

This is the first phase of the model where requisite data and Indicators of Compromise (IOC) are gathered and filtered by the tactical intelligence team for threat intelligence operations. The following elements are the indicators of compromise identified as illustrated in Table 1.

Table 1. Indicators of Compromise (IOF)

Indicators of Compromise	Details
Name	FluBot
Attack family	Malware (banking Trojans)
Type of attack	Mobile Malware Attack
Target OS	Android
Country of origin	Spain
Attack Vector	SMS messaging
Resource materials	URLs, journals and books
Year of inception before propagation	2020
Risk and impact	Critical

Accordingly, in order to ascertain and understand critical information, attack and motives of the FluBot malware, Alien Vault was considered and used.

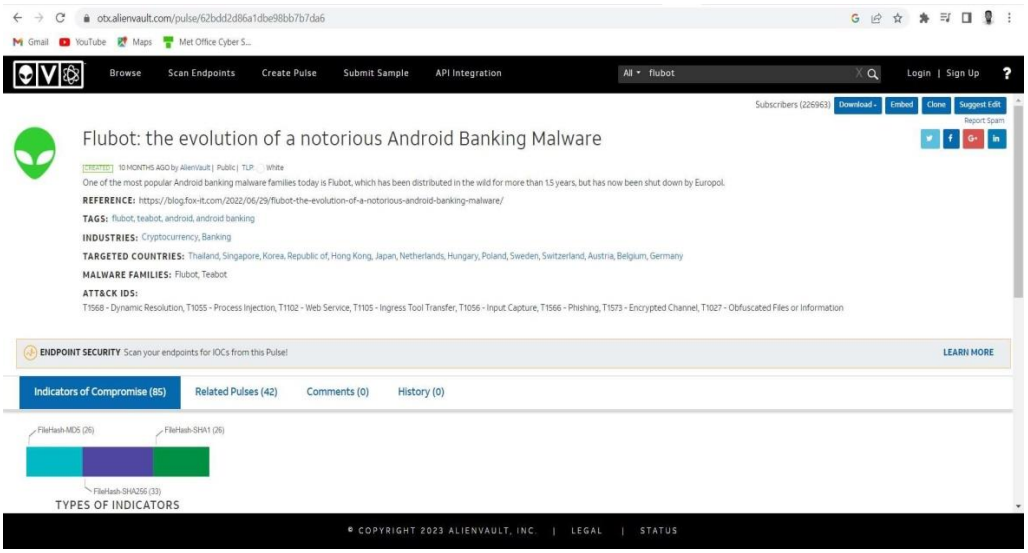

Figure 7. Research of critical information about FluBot using Alien Vault (Exchange, 2020)

3.2 Level 2: Threat Intelligence Integration and Dissemination

This is the second phase of the model where actions are taken based on the identified data or indicators of compromise collected from Level 1 to respond to the attack/threat (FluBot).

3.3 Investigation and Analysis

In investigating the FluBot Android Malware that has been a major global mobile security concern, the 5 steps of OPSEC were also considered. This includes: identification of critical information about the APT, FluBot; analysis of the APT, analysis of possible vulnerabilities; risk assessment; and use of applicable countermeasures. A static analysis was conducted on VirusTotal to generate basic metadata about FluBot.

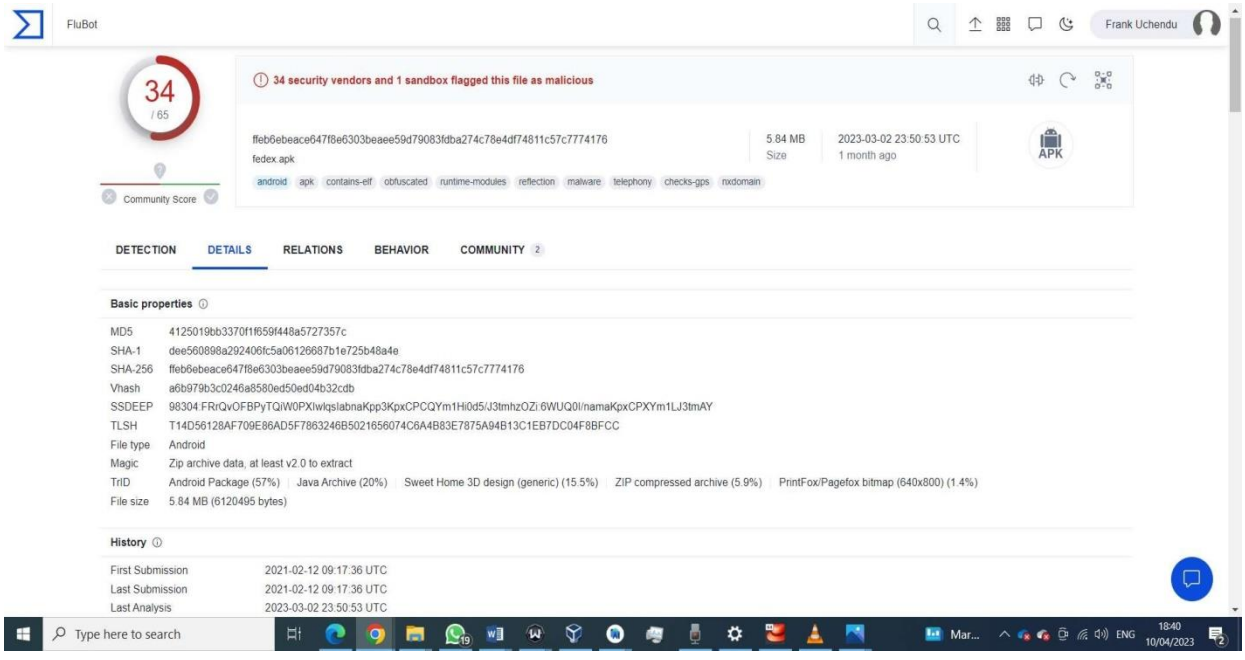

Figure 8. Basic metadata of the FluBot Android Malware (VirusTotal, n.d)

Accordingly, in order to analyze the malware dynamically, we must first understand its tactics and indicators of compromise (IP address, bad domain, OS) through the Capability Maturity Model (CMM) phases of threat intelligence analysis.

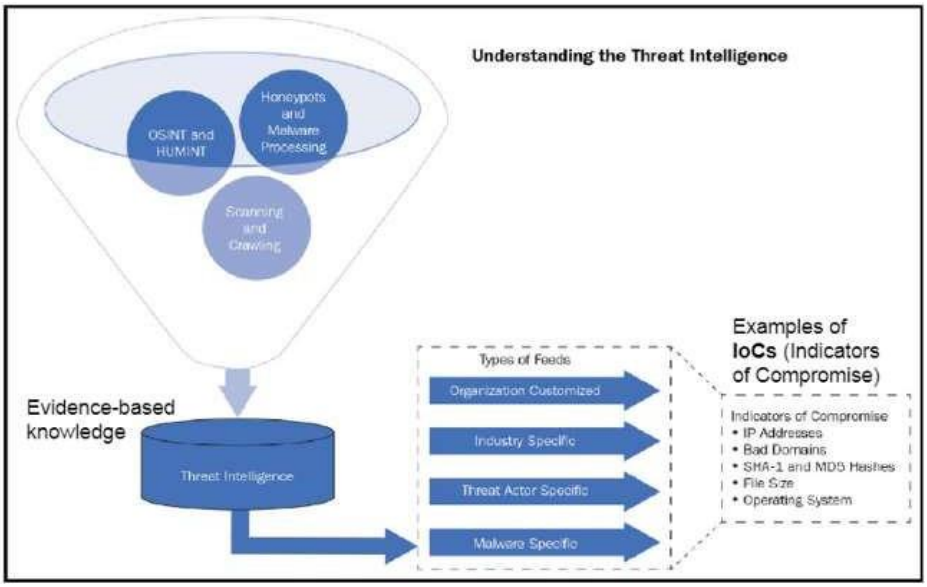

Figure 9. Depicts the Threat Intelligence Maturity Model (Abouzakhar, 2023)

Moving forward, the tools used to develop the security scenario (FluBot) in this investigation were downloaded, set up and configured respectively. They include: Dagah**,** which was installed on a virtual environment (Virtual Box) for designing and launching of attacks against Android Emulator**,** which are the simulated targets.

3.3.1 Dagah Environment Set-up

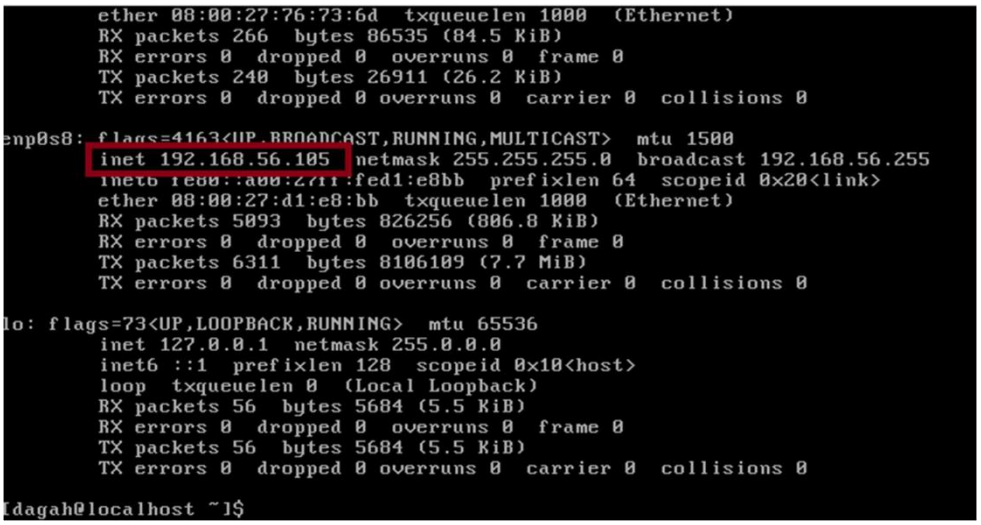

Figure 10. The IP address to use it to log on the Dagah Web interface via HTTP

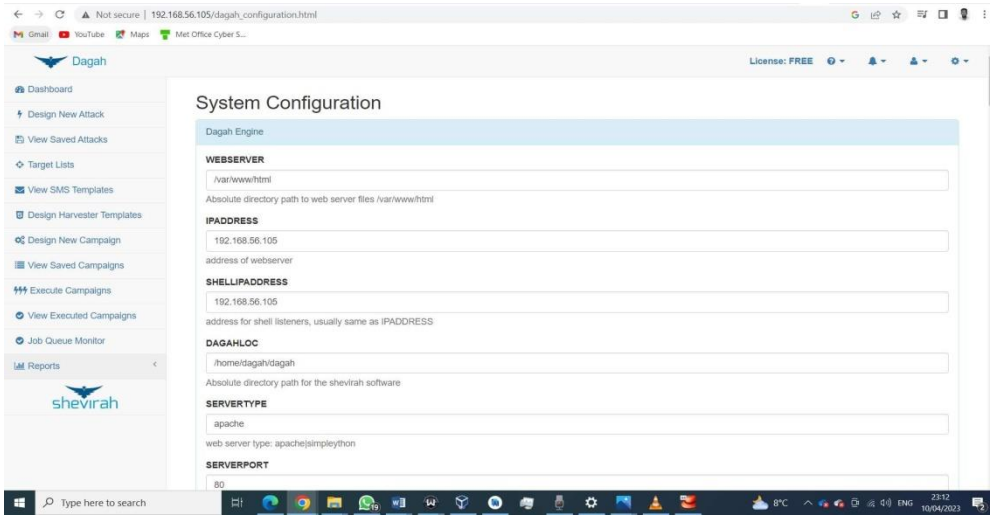

Figure 11. Configuration of the system by setting the IP address to 192.168.56.105

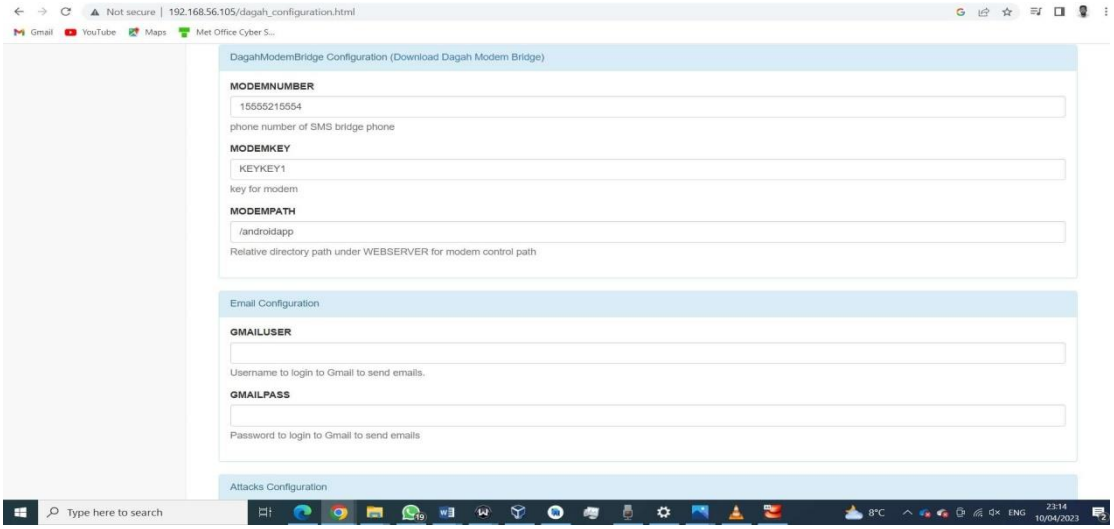

Figure 12. Configuration of the Dagah modem bridge on the GUI

we downloaded and installed the Dagah modem bridge App on the Attack Android simulator. The Dagah Modem App ensures that SMS attacks are sent from the Dagah GUI to the android device

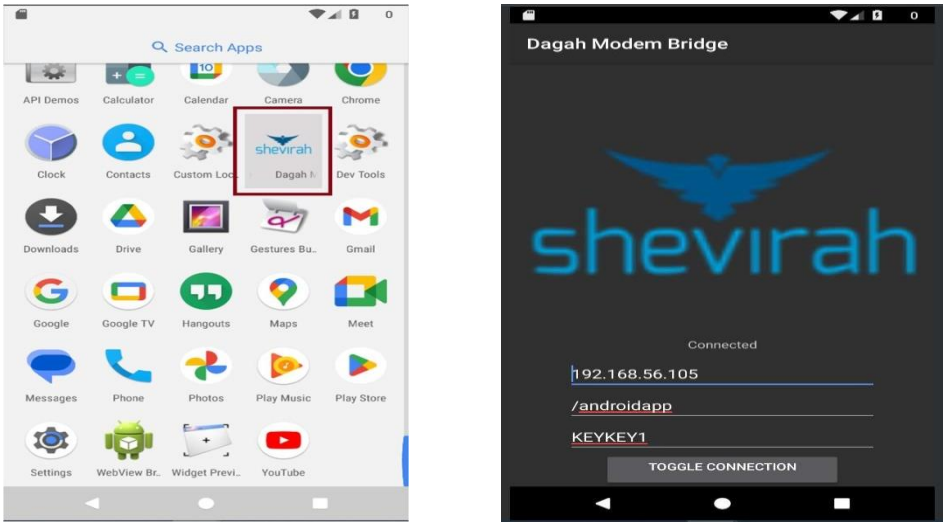

Figure 13. App on the Attackers' device, setting of the IP address, path and key on the Dagah Modem Bridge

3.3.2 Android Studio Setup

Furthermore, we installed the Android studio and created two Android Nexus 5X virtual devices for emulation of the operating system, as shown in Figure 14.

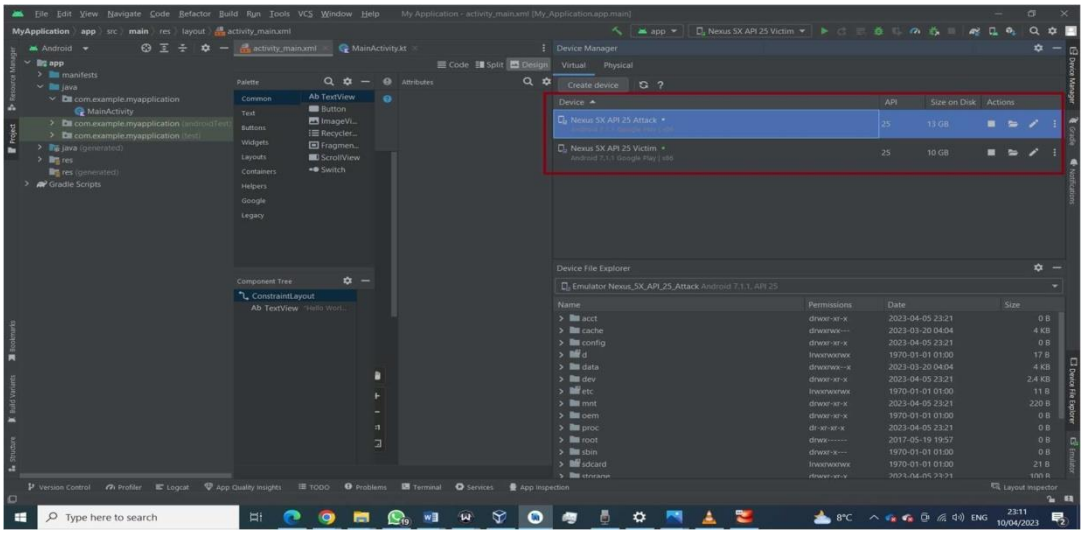

Figure 14. Installation of Android studio and creation of the devices

To ascertain that the respective devices are functional and connected, we dialed the Victim's device using that of the Attacker's as shown in Figure 15.

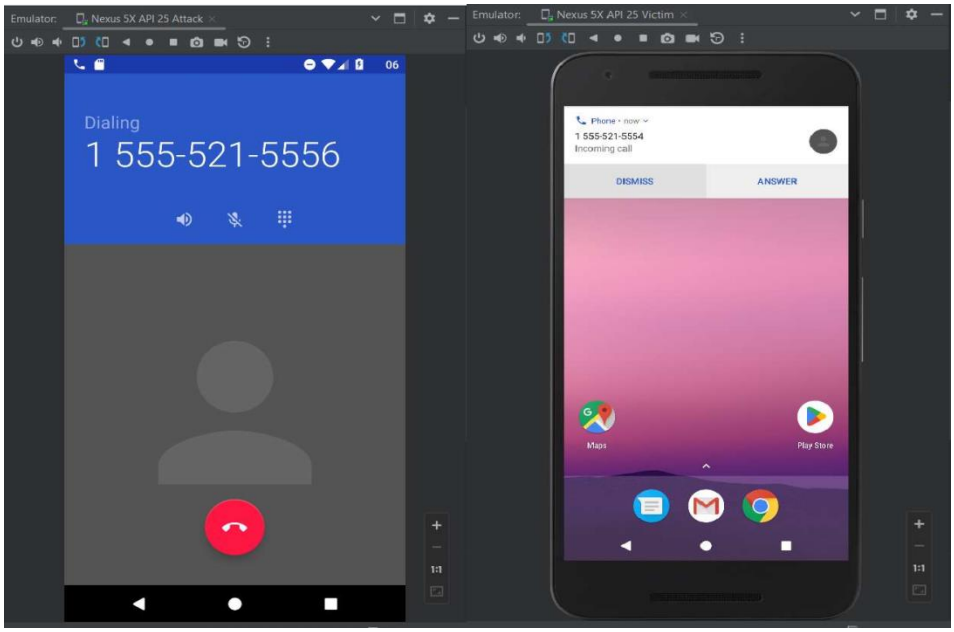

Figure 15. The Attacker's device calling the Victim's

3.3.3 Practical Experiment

FluBot is distributed through phishing attacks using SMS as its mode of transmission. It propagates by harvesting users' credentials through deceptive links and for this experiment, we conducted two harvester phishing types of attack:

i. An email phishing using the built-in Gmail template on Dagah GUI to harvest the users' Gmail log-in credentials; and

ii. we designed a harvester template by cloning and editing a website (https://gradintel.com), then harvested the credentials submitted to the website.

For both attacks, the victims' Android simulating mobile device receives an SMS from the attacker's device. When the victim clicks on the malicious link and inputs his credentials, the attacker grabs and stores the credentials in the campaign results of the Dagah GUI, as shown in Figure 16.

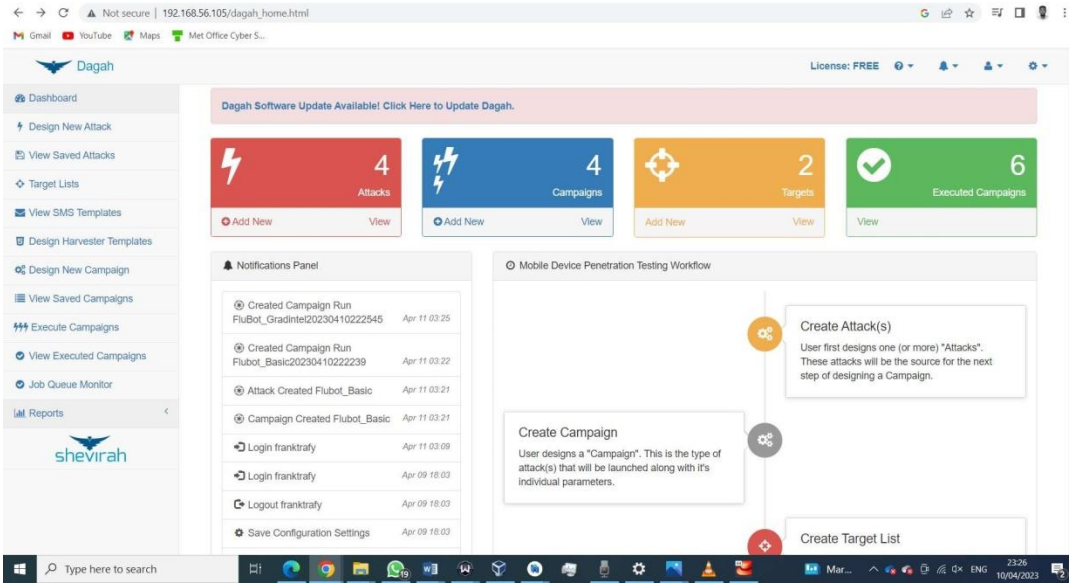

Figure 16. The Dagah GUI dashboard showing designed attacks, campaigns, target lists and the executed campaigns

For the built-in Gmail template harvester phishing attack, we designed a new attack with the harvester type of attack selected, delivery method set to SMS and harvester template (gmail.com) selected, created our target list with the Android Victim's phone number and designed a campaign before executing.

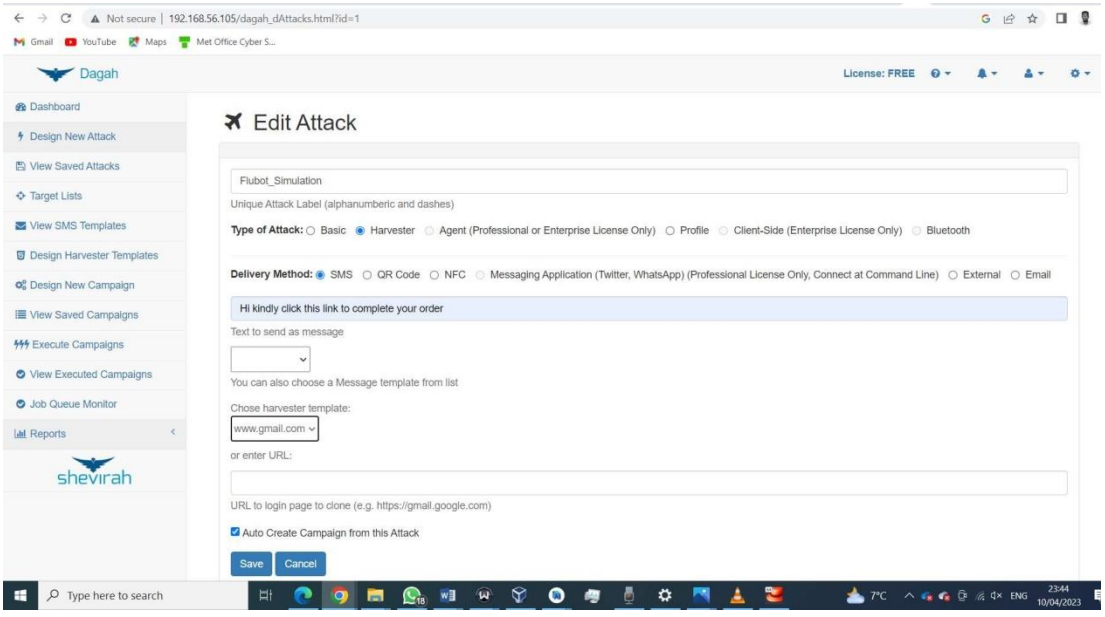

Figure 17. Designing a new Gmail harvester phishing attack on Dagah GUI

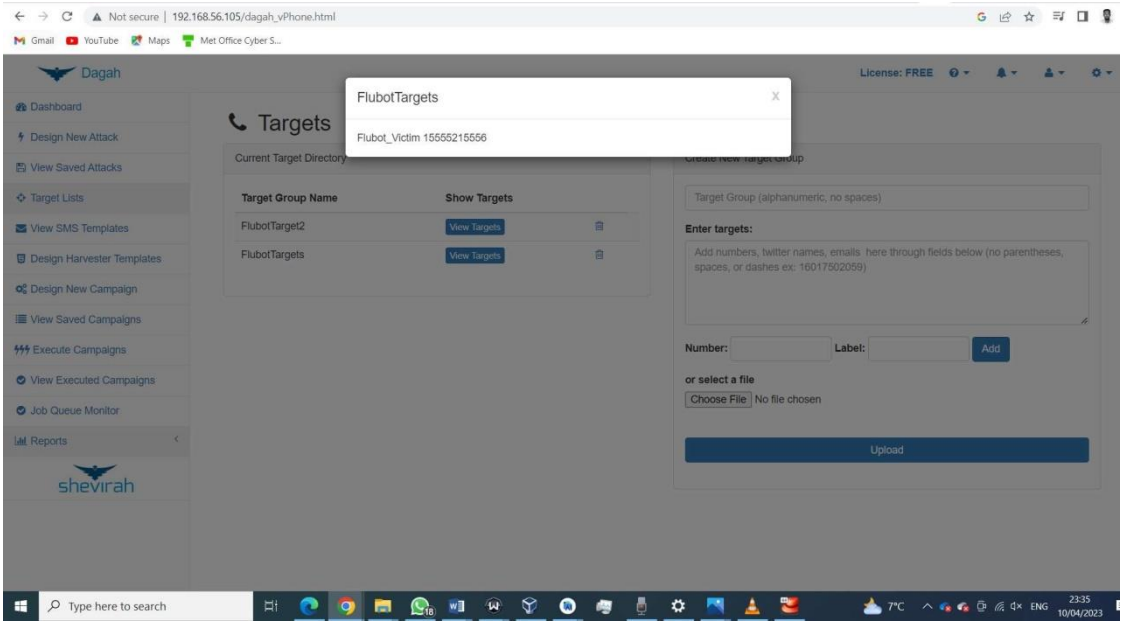

Figure 18. Target list created with the Victim's number

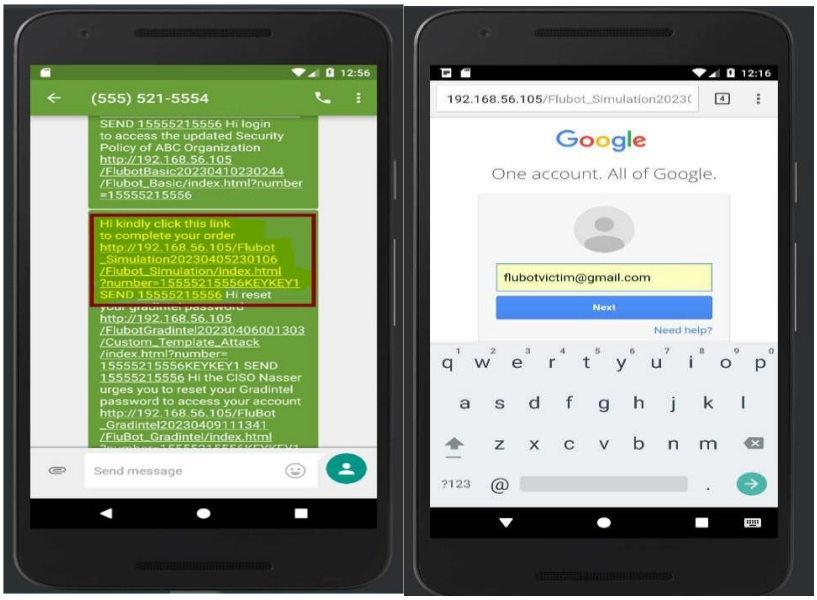

Figure 19. Depicts the victim's device receiving the malicious link and opening the phony Gmail login page

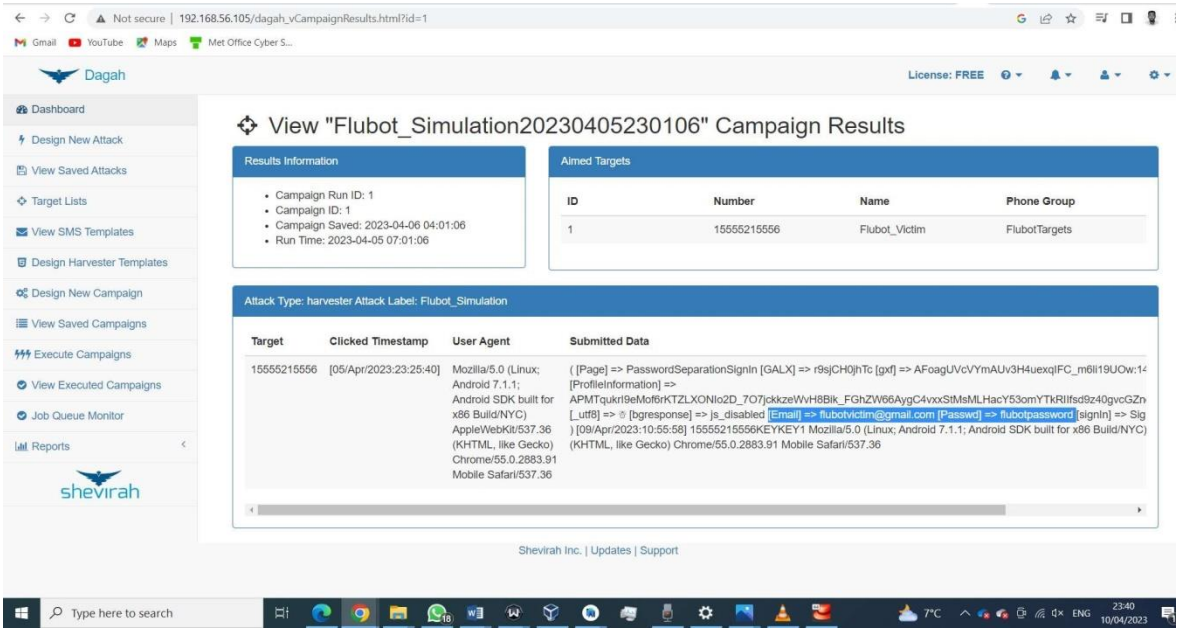

Figure 20. Results of the executed campaign showing the grabbed credentials of the Victim

For the cloned and edited Website template harvester phishing attack, we clicked on the Design Harvester Template on the Dagah GUI, clicked on Add new Harvester Template, entered the new template name (gradintel.com) and saved it. Then we removed the client-side validation script and updated the form method's POST action in line 252 of HTML code to './post.php', as shown in Figure 21.

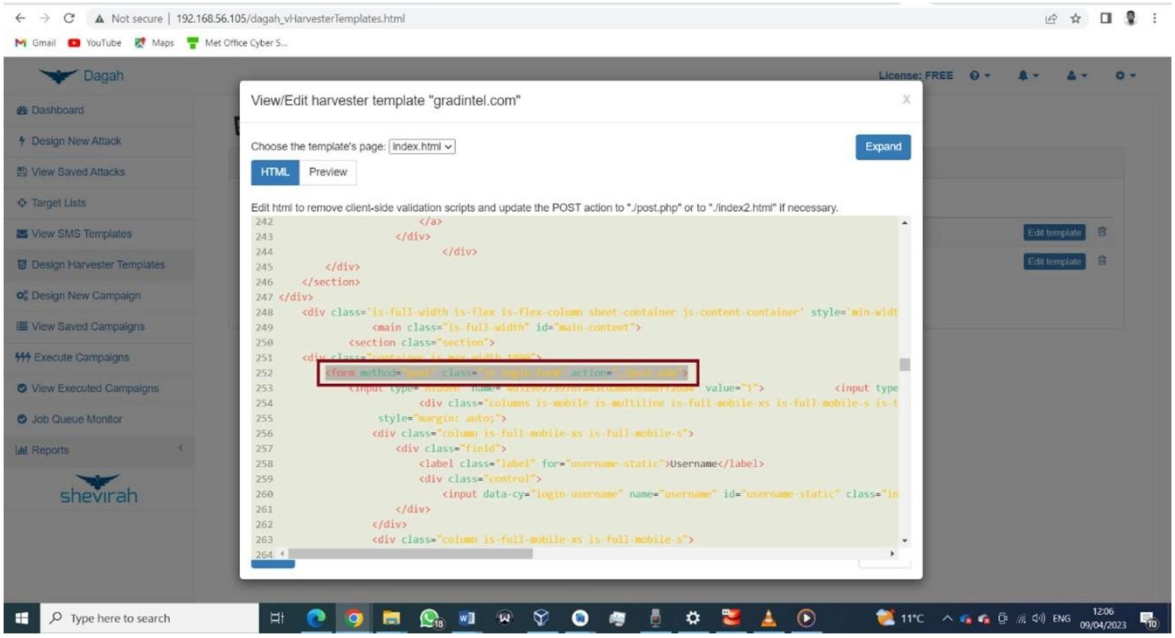

Figure 21. Editing the HTML code to clone the website

we then created a new attack, choosing the *harvester* attack type, setting the delivery mechanism to *SMS*, choosing the harvester template (*gradintel.com*), choosing our target list containing the Android Victim's phone number, and creating a campaign before launching it as depicted in figures 22, 23, 24 and 25.

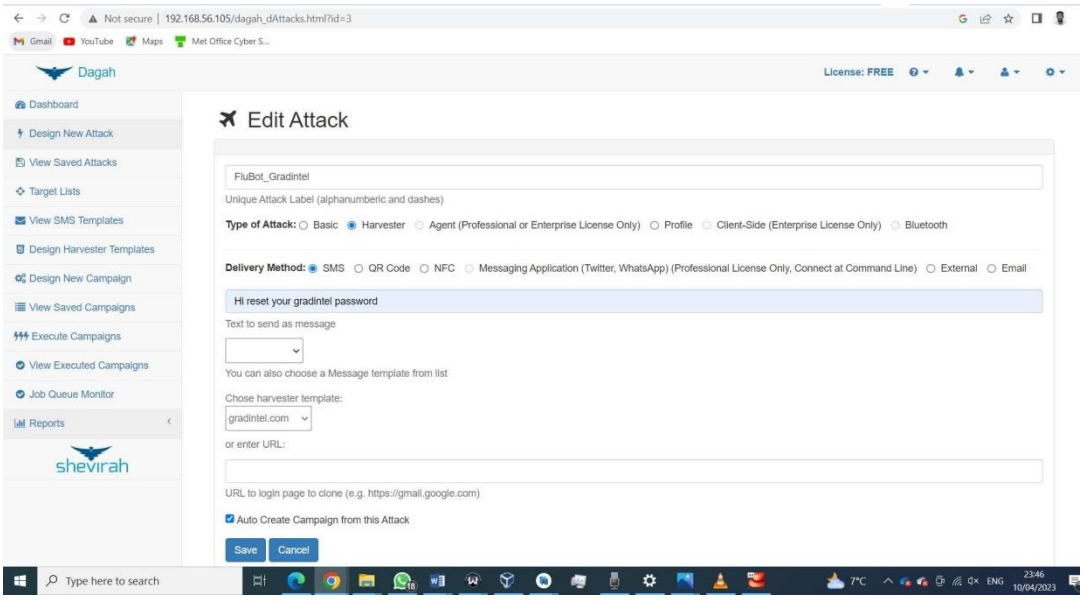

Figure 22. Designing a new website (gradintel.com) harvester phishing attack on Dagah GUI

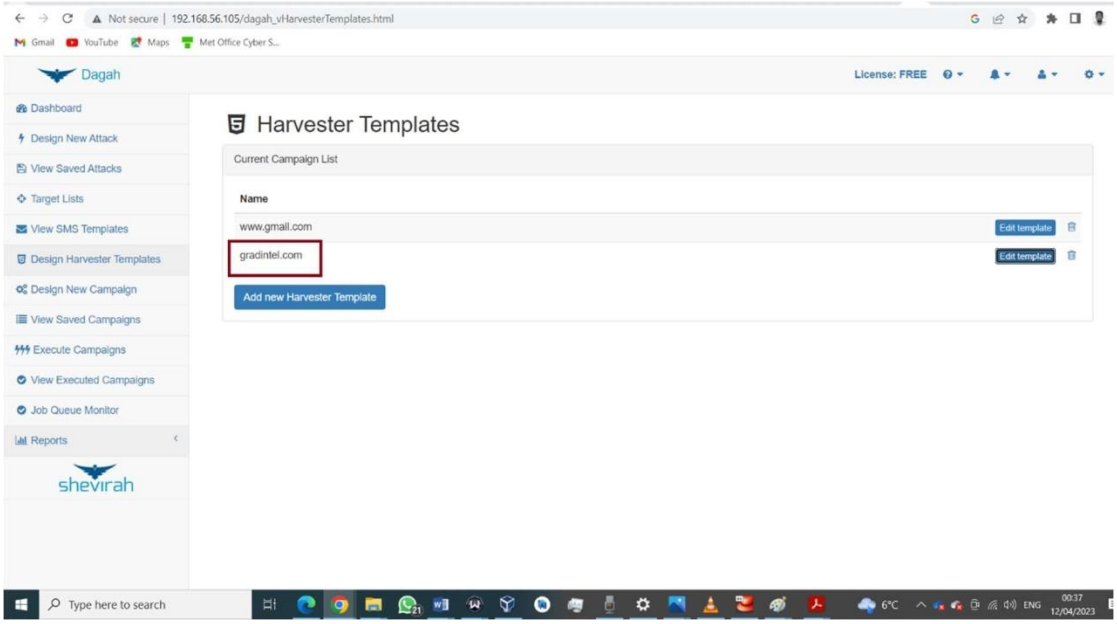

Figure 23. Choosing the newly created ‘gradintel.com’ harvester template

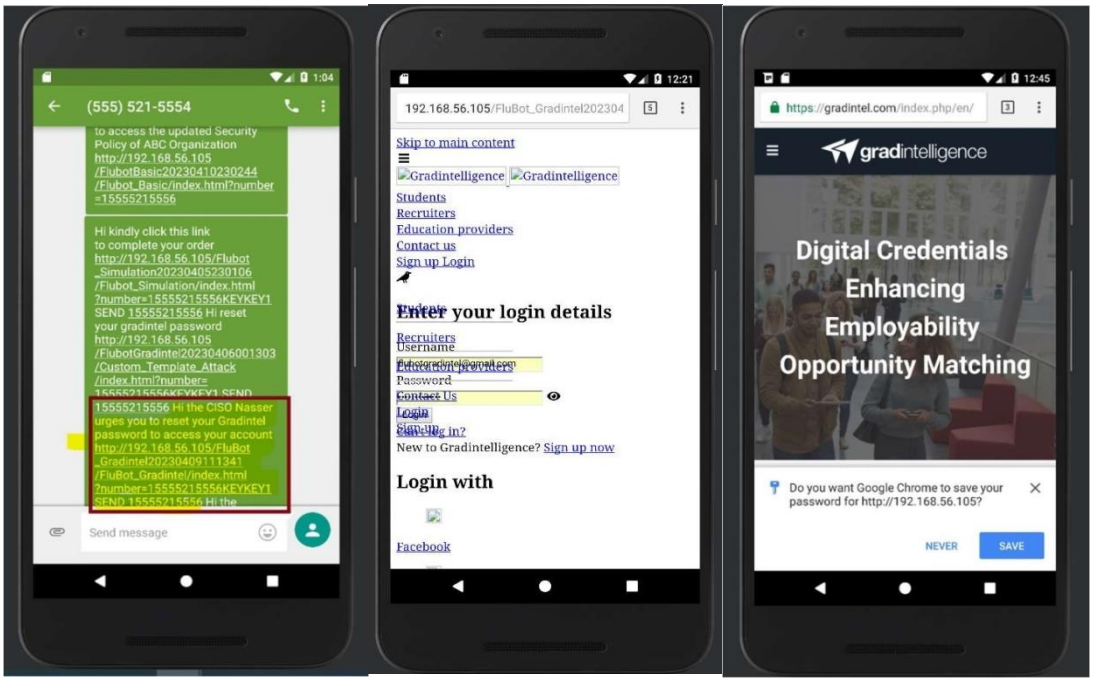

Figure 24. Victim's device receiving the malicious link, opening the phony Gradintel login page and landing page of the main Website

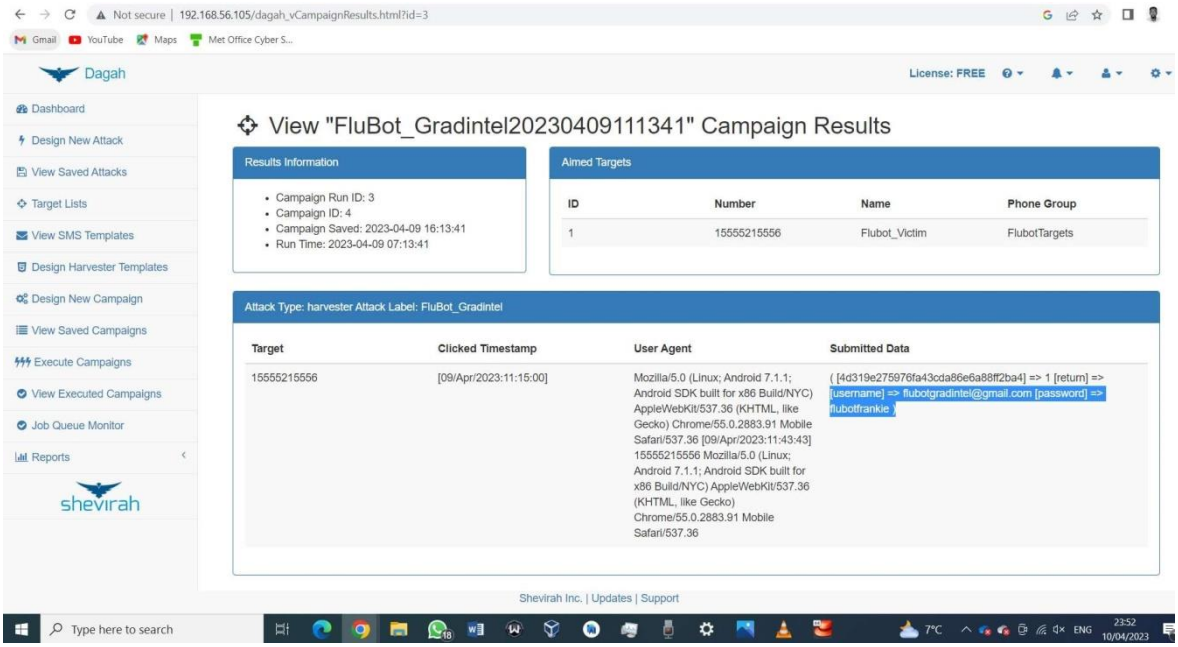

Figure 25. Results of the executed campaign showing the grabbed credentials of the Victim

3.4 Dissemination

3.4.1 Techniques and Procedures

Based on our research and the analysis conducted, we observed the following to be the techniques and procedures instigated by FluBot to obstruct mobile security:

i. String Encryption: FluBot uses a unique encryption method to encrypt all relevant strings. Each class has a function in charge of encrypting any dubious strings it comes across.

ii. MultiDex (Multi Davik Executable): APK files contain DEX (Dalvik Executable) which are executable codes that ensures the running of your Android app. When more than one DEX is generated

to run your app, it is called MultiDex. FluBot conceals its harmful code from reversers and static analyzers by using MultiDex.

iii. DEX Decryption: FluBot employs a decrypted and loaded encrypted dex file from the assets to carry out its malicious behaviour.

iv. Domain Generation Algorithm (DGA): This is an algorithm used in generating domain names. FluBot uses this algorithm to locate and communicate with the C2 server in order to bypass security safeguards.

v. DNS Tunneling over HTTPS: Flubot resolves the IP addresses of DGAs after generation, then communicates with the C2 server through DNS Tunneling over HTTPS port 443.

vi. Error Logging: The C2 records any application errors that are not noticed. This enables the attackers to update and fix the FluBot's code.

3.4.2 Appropriate Intelligence and Cost-Effective Solutions/Countermeasures

This study proposed some appropriate intelligence and cost-effective solutions/countermeasures to protect and improve systems from the FluBot mobile security threat. They include but are not limited to:

i. The internal team (e.g., IT, legal, communications, internal audit, risk management, etc.) should be trained by the organisation using tabletop exercises or other briefings intended to test and enhance incident response function.

ii. A complete system reset or safe boot of the android devices will get rid of the malware and all current settings, including stored data.

iii. For organisations and individuals, ensure you stay informed on phishing tactics and social engineering techniques through system awareness campaigns, workshops and education;

iv. Enable two-factor authentication (2FA) or multi-factor authentication (MFA) on your accounts to provide an extra layer of protection and prevent unauthorised access to your device;

v. Obtain APKs from legitimate vendors rather than unauthorised ones, and avoid installing add-on programmes as they might include the malware Flubot;

vi. Avoid opening attachments from unreliable sources or clicking on suspicious links as they can include Flubot or other malicious programs that has tendencies of compromising your device.

vii. Formulate, review and implement when needed, an intrusion detection system (IDS) to monitor network traffics for suspicious activities and signal alerts when noticed, a Business Impact Analysis and Business Continuity and Disaster Recovery Plan for contingencies.

3.4.3 Unsolved Problem

It is known that Flubot spreads using SMS messages that entice recipients to click on harmful links. Users continue to fall prey to these phishing assaults, spreading infection, despite awareness campaigns and security precautions. It remains a challenge to stop users from clicking on these links and falling for social engineering tricks.

3.5 Utilization

3.5.1 Legal and Ethical Issues

In the event of a breach in the confidentiality, integrity, or availability of an organization's system, it is necessary to have more robust governance structures, as well as legal and ethical obligations to protect and prioritise organisational assets. When there is a legal crisis or APT of any type, managing legal privileges can become a severe problem. Security concerns must be considered when drafting legal agreements with partners, suppliers, and customers. This will enable better containment, communication, and analysis of the technical and legal dangers posed by the attack.

4. Conclusion

This survey report provided a thorough analysis and explanation of the FluBot Android malware, to evaluate it as a danger to mobile security. we were able to identify critical information of the FluBot APT using VirusTotal and Alien Vault, and we were also able to develop and execute a FluBot-like attack against simulated targets using Dagah and Android Simulator. As a result, we were able to:

i. Identify FluBot-instigated tactics and procedures to undermine mobile security;

ii. Make some pertinent intelligence, recommendations and cost-effective fixes and countermeasures to ensure mobile organisational security; and

iii. Talk about moral and legal concerns to safeguard assets.

5. Recommendation

At the end of the investigation, we came up with the following recommendations:

i. Block unknown senders or enable SMS filtering in the device's settings. By doing this, you might be able to prevent harmful SMS messages from reaching your smartphone and potentially propagating the FluBot malware.

ii. Update your operating system, programs, and security updates on a regular basis to keep your devices safe from known vulnerabilities;

iii. Ensure you periodically back up your data to a secure location in order not to lose vital information;

iv. Ensure to employ a reliable antivirus program on your Android device.

v. Avoid jailbreaking your device. This could severely reduce its security and expose gaps in protection.

Acknowledgments

We are grateful to Dr. Ernest E. Onuiri for his useful advice and for giving us some possibilities that made the work better than it was before, and we are grateful to every team member who took the time to participate in this study.

Authors contributions

Dr. Uchenna J. Nzenwata was responsible for android environment implementation, the flubot malware analysis and its documentation. Frank Uchendu was responsible for the implementation using the Dagah tool and the documentation of the Dagah tool analysis. Haruna Ismail and Eluwa M. Jumoke contributed in gathering the literature sources and the documentation. Himikaiye O. Johnson was responsible for the final proof reading and the collation of the article sections. All the authors ensured that the documentation was thoroughly proof read and ascertain equal right to the contribution of the final state of the work.

Funding

Not Applicable

Competing interests

Not Applicable

Informed consent

Obtained.

Ethics approval

The Publication Ethics Committee of the Canadian Center of Science and Education.

The journal’s policies adhere to the Core Practices established by the Committee on Publication Ethics (COPE).

Provenance and peer review

Not commissioned; externally double-blind peer reviewed.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Data sharing statement

No additional data are available.

References

"FluBot Android Spyware Taken Down in Global Law Enforcement Operation," The HackerNews. Retrieved Apr. 09, 2023, from https://thehackernews.com/2022/06/flubot-android-spyware-taken-down-by.html

"VirusTotal," www.virustotal.com, Apr. 10, 2023.

Abouzakhar, N. (2023, March 06). Lecture on Advanced Persistent Attacks (APTs), Active Cyber Defence (ACD) and Threat Intelligence and Operations [PowerPoint slides]. Available: https://blackboard.salford.ac.uk/

Blázquez, E., & Tapiador, J. (2023). Kunai: A static analysis framework for Android apps. *SoftwareX, 22*, 101370. https://doi.org/10.1016/j.softx.2023.101370

Callaham, J. (2019). The history of Android OS: its name, origin and more. *Android Authority, 18.*

Chapin, L., Piscitello, D., & Strutt, C. (2022). *Malware Landscape 2022.*

Chaurasia, P. (2015). *Dynamic analysis of Android malware using DroidBox* (Doctoral dissertation, Tennessee State University).

Exchange, A. O. T. (2020). AlienVault Open Threat Exchange.

Fernick, J. (2022). *Flubot: the evolution of a notorious Android Banking Malware*. NCC Group Research, Retrieved July 5, 2022, from https://research.nccgroup.com/2022/07/05/flubot-the-evolution-of-a-notorious-android-banking-malware/

FluBot, "Partners-in-crime: Medusa and Cabassous attack banks side-by-side — ThreatFabric," Retrieved Apr. 09, 2023, from https://www.threatfabric.com/blogs/partners-in-crime-medusa-cabassous.html

García-Teodoro, P., Gómez-Hernández, J. A., & Abellán-Galera, A. (2022). Multi-labeling of complex, multi-behavioral malware samples. *Computers & Security, 121*, 102845. https://doi.org/10.1016/j.cose.2022.102845

Garg, S., & Baliyan, N. (2021). Comparative analysis of Android and iOS from security viewpoint. *Computer Science Review, 40*, 100372. https://doi.org/10.1016/j.cosrev.2021.100372

Gibbs, S. "About the flubot virus," Queensland Tech, Aug. 28, 2021. Retrieved April 10, 2023, from https://queenslandtech.com.au/flubot/

Liu, M., Zhang, Y., Liu, B., Li, Z., Duan, H., & Sun, D. (2021, December). Detecting and characterizing SMS spearphishing attacks. In Annual Computer Security Applications Conference (pp. 930-943). https://doi.org/10.1145/3485832.3488012

Mayrhofer, R., Stoep, J. V., Brubaker, C., & Kralevich, N. (2021). The android platform security model. *ACM Transactions on Privacy and Security (TOPS), 24*(3), 1-35. https://doi.org/10.1145/3448609

Meng, H., Thing, V. L., Cheng, Y., Dai, Z., & Zhang, L. (2018). A survey of Android exploits in the wild. *Computers & Security, 76*, 71-91. https://doi.org/10.1016/j.cose.2018.02.019

Mogicato, R., & Zermin, A. Design and Implementation of a Collaborative, Lightweight Malware Analysis Sandbox using Container Virtualization.

Ndatinya, V., Xiao, Z., Manepalli, V. R., Meng, K., & Xiao, Y. (2015). Network forensics analysis using Wireshark. *International Journal of Security and Networks, 10*(2), 91-106. https://doi.org/10.1504/IJSN.2015.070421

Özdemir, D., & Zaim, H. Ç. (2021). Investigation Of Attack Types in Android Operating System. *Journal of Scientific Reports-A, 46*, 34-58.

"Top 10 Android Security Risks | eSecurity Planet," eSecurityPlanet, Retrieved March 18, 2011, from https://www.esecurityplanet.com/trends/android-security-risks/

Riasat, H., Batool, T., & Iqbal, S. (2022). *Review and Comparative Studies on Mobile Operating System* (No. 8848). EasyChair.

Salsabila, H., Mardhiyah, S., & Hadiprakoso, R. B. (2022, November). *Flubot Malware Hybrid Analysis on Android Operating System.* In 2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS) (pp. 202-206). IEEE. https://doi.org/10.1109/ICIMCIS56303.2022.10017486

Schütte, J., Fedler, R., & Titze, D. (2015, March). *Condroid: Targeted dynamic analysis of android applications.*

In 2015 IEEE 29th International Conference on Advanced Information Networking and Applications (pp. 571-578). IEEE. https://doi.org/10.1109/AINA.2015.238

StatCounter, “Mobile Operating System Market Share Worldwide,” StatCounter Global Stats, 2022. Retrieved from https://gs.statcounter.com/os-market-share/mobile/worldwide

Van Haastrecht, M., Golpur, G., Tzismadia, G., Kab, R., Priboi, C., David, D., ... Spruit, M. (2021). A shared cyber threat intelligence solution for smes. *Electronics*, *10*(23), 2913. https://doi.org/10.3390/electronics10232913