Czech Technical University in Prague

Faculty of Electrical Engineering

MASTER'S THESIS

2021 Bc. Akshay Sharma

Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Telecommunication Engineering

A Honeypot for Research on VoIP Systems Security

Ing. Program: Electronics and communication

Specialization: Communication System and Network

Aug. 2021

Author: Bc. Akshay Sharma

Supervisor: Ing. Pavel Troller, Csc.,

Declaration

I hereby declare that this master's thesis is altogether my work and that I used only the cited sources following the Methodical instruction about the observance of ethical principles of preparation of final university projects.

Prague

……..

Signature

Acknowledgments

I am deeply grateful to my supervisor: Ing. Pavel Troller, Csc., Faculty of Telecommunications at Czech Technical University in Prague, for his encouragement and invaluable guidance throughout the thesis, the door from Mr. Pavel Troller was always open for me whenever I ran into some problem and queries about my thesis project. Furthermore, he drove me in the right direction whenever he thought I needed it. Also, I would like to thank Ing. Umang, Arpan Tyagi, Swati Kushwaha for providing me with unfailing support throughout my project. This accomplishment would not have been possible without them.

Many thanks to my family for their support, love, and understanding in this pandemic situation.

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Sharma Akshay** Personal ID number: **489464**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Telecommunications Engineering**

Study program: **Electronics and Communications**

Specialisation: **Communication Systems and Networks**

II. Master's thesis details

Master's thesis title in English:

A Honeypot for Research on VoIP Systems Security

Master's thesis title in Czech:

Honeypot pro výzkum zabezpečení VoIP systémů

Guidelines:

Based on free open-source software products, create a testing environment with standard VoIP exchange functions, usable for monitoring and analysis of fraudulent attacking traffic and suitable for development of tools for blocking it.

Bibliography / sources:

[1] Spitzer L.: Honeypots: Tracking Hackers. Addison-Wesley Professional (September 20, 2002). ISBN 978-0321108951
[2] Akkaya D., Thalgott F.: Honeypots in Network Security. LAP LAMBERT Academic Publishing (April 29, 2012). ISBN 978-3659113529

Name and workplace of master's thesis supervisor:

Ing. Pavel Troller, CSc., Department of Telecommunications Engineering, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **12.02.2021** Deadline for master's thesis submission: **13.08.2021**

Assignment valid until: **30.09.2022**

Ing. Pavel Troller, CSc. Supervisor's signature	Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt	Student's signature

Abstract

Voice over Internet Protocol (VoIP) technology has been developed, and its importance in the broadband network is rapidly increasing. *VoIP* phones call on the same route used for network and internet traffic and face the same cyber threat that plagues the data network today. These threats included denial-of-service attacks, viruses, data breaching. The security issue or concern of the Ip telephony network pales compared to the technology boom and how IP telephony equipment manufacture brings technology to the mass area. In the past, we already have seen the many developed technologies face the real challenge only after these technologies become a boom in the market and widely adopted by the companies.

This paper explains the security risk, problems associated with the VoIP network, and different types of attacks faced by the VoIP; it also outlines the steps for helping to secure an organization's VoIP network.

Contents

FIGURE

TABLE

1.1 Introduction

What is VoIP? It is an acronym for Voice over Internet Protocol or, in more familiar terms, phone service via the Internet; it worked by taking analog voice signals and convert them into a digital signal and send these signals as data over the Internet. The voice service flow over the general-purpose packet-switched network instead of the then dedicated circuit-switched transmission line. The thought of VoIP is the result of telephone and internet technology started approx. In the 1870s by alexander Grahm bell and Elisa gray [1].

The First voice transmission uses a ring-down circuit, which means there is no ringing of a headset, no dialing of numbers; instead, a physical wire connected two devices. The Internet era started in 1968 when the first Internet was developed by ARPANET(Advance Research project Agency Network). The invention of TCP/IP begins around 1972 by Dr. Vint Cerf. And finally, in Hallock 2004, VocalTec[1] released the first-ever Internet VoIP application in Feb 1995. The product was Internet Phone, allowing one user to call another using their computer with a mini microphone and speakers. Later VocalTec worked on adding Internet voice mail applications and paired their Internet phone software with Microsoft NetMeeting[2].

Protocol used in VoIP is commonly referred to as VoIP protocol. Currently, three protocols are widely being used which are, H.323 family of Protocol, SIP, which is Session Initiation Protocol; and MGCP, which is Media Gateway Controller Protocol. From these three protocols, SIP is considered as more intelligent at the endpoint security system. MGCP think of as competent in network components, and finally, H.323 is intelligent universally.

With the advancement of *VoIP*, it has become clear that technology is going to the third aspect and the first and second aspects are cellular network and PSTN. However, the call can be made to any of these three aspects of devices from anywhere in the world. With Increasing popularity, is also attract different invaders, Security issue in VoIP is quite complex Each component in VoIP draw various vulnerabilities for attackers to exploit. Many vulnerabilities are similar to those of the PSTN system; for example, in PSTN, eavesdropping can be achieved by physically placing the

[1] Company was founded in 1985 by Alon Cohen and Lior Haramaty, who patented the first Voice over IP audio transceiver.

[2] Microsoft NetMeeting is a discontinued VoIP and multi-point videoconferencing client included in many versions of Microsoft Windows. It uses the H.323 protocol for videoconferencing -- 1996

listening device on the phone line, access to the *TCP/IP* network can also allow to recognize of a listing device, i.e., *Packet analyzer(Wireshark)* on the network and intercept phone conversation.[26]

For this purpose of preventing different *VoIP* attacks, one needs to understand it first; for this, they do need a system that can be used as a bet for the attackers. Setting up Honeypot on the network has been gain popularity since the 2000s. The primary purpose of the Honeypot system is intended for no other purpose other than collecting data from attacks. They gather information about attackers, their sessions while allowing them to work on Honeypot.

1.2 Session initiation protocol.

SIP is used to initiate, modify, and terminate a two-way interactive user session involving multimedia elements such as video, voice, instant messaging, online games, and virtual reality. [2]

SIP is combined with other related IETF protocols (such as SIP, SDP, and MGCP) to provide a broader range of VoIP services. The SIP architecture is similar to HTTP (Client-Server Protocol). It contains the request sent from the SIP user's client to the SIP server. The server processes the request and responds to the client. The request message and the associated response message execute the SIP transaction together. SIP is one of the most popular VoIP protocols used today. The Internet community invented it as a generic protocol for enhancing web experience with multimedia interaction. Later it was found as suitable for pure VoIP application, and a new round of development started to improve its initially naïve and insufficient signaling to make it more compatible with other signaling systems of the current telco industry; it resulted in a big heap of various recommendation specifying extensions and improvements of the original SIP. For many years, the SIP version is stuck at 2.0 and does not increase regardless of much progress, added since v2.0 was established. The original "ROOT" document specifying SIP 2.0 is RFC 3261[3] .

The SIP architecture (Figure 1) describes two components, and the first one is the User-Agent (UA) second is SIP Server.

[3] https://tools.ietf.org/html/rfc3261

1.3 SIP Components

User Agent (UA) is available at the SIP endpoints and has been used to create or receive SIP messages and create a SIP session, user agent itself contains two components.

> User-agent client (UAC) is a function that performs client functionality. That generates a request, which in turn is service by UAS (user agent servers). In other words, UAC is a Session Initiation Protocol or Voice over Internet Protocol application that serves as a peer-to-peer (p2p) communication gateway and generates distributed network service requests.

User-agent Server (UAS) responds to User Agent Client (UAC) service requests based on input.

SIP Server, sip server is described in three categories.

A redirect server is a user agent that generates a 3xx[4] response to the request it receives, directing the client to contact an alternate URI set. It also allows users to change geographical location and still be reachable through the same SIP address.

A *proxy server* is an entity that acts as both a server and a client to request on behalf of other clients. It is often responsible for a domain that a client is registered.

Register Server accepts *REGISTER* requests and places the information it receives in those requests into the location service for the domain it handles.

[4] 3xx—Redirection Responses.

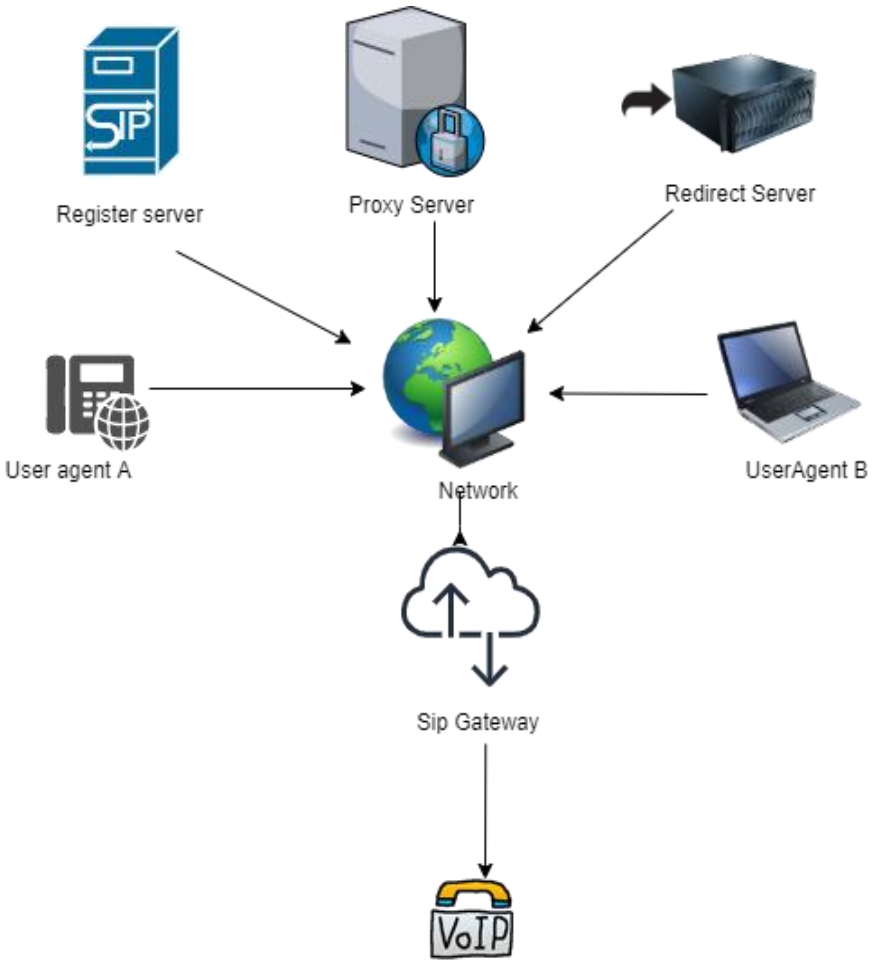


Figure 1. Sip Communication

After receiving the SIP request, it will determine the next-hop server and return the address of the next-hop server to the client instead of forwarding the request to the next-hop server. The server that converts the destination number to an IP address. This information will then be passed back to the original endpoint, responsible for sending the message directly to the destination. One of the security advantages of SIP is that it uses only one port.

1.4 Sip Core Method

INVITE is used to initiate a session with the user agent, or we can say an INVITE method is helping to establish a session between user agents

- INVITE message to contain the media information of the caller.
- The session can be considered established when the INVITE message has received a successful response (2xx) or sent an ACK.

ACK Confirm that the client has received a final response to an invite request.

BYE is used to terminate an established session.

- A proxy server cannot send it.
- It generally routes between two ends.

CANCEL any pending searches but does not terminate a call that has already been accepted.

- It can be sent either by UA or the proxy server.
- CANCEL is a hop-by-Hop request that means it goes through between user agents.

OPTIONS—Queries the capabilities of servers.

REGISTER—Registers the address listed in the header field with a SIP server. A user agent sends this request to a registration server.

- It carried the address of the Record in the TO header of the user that is being registered.
- The request contains 3600 sec.
- One UA can send a Register request on behalf of another user agent, and this type of registration is known as third-party registration. Here the form tag contains the URI of the party submitting the registration on behalf of the party identifier in the header.

1.5 SIP Response Code

SIP 1xx—Informational Responses, continuing to process the request. The informational response, also known as a Provisional response (1xx), may contain message bodies, including session descriptions.

SIP 2xx—Successful Responses – The action was successfully received, understood, and accepted.

SIP 3xx—Redirection Responses – Action needs to be taken to complete the request.

SIP 4xx—Client Failure Responses – Request has the wrong syntax or cannot fulfill at this server.

SIP 5xx – Server Error -- the server failed to fulfill a valid request.

SIP 6XX- Global Failure — the request cannot be fulfilled at any server.

2.1 VoIP Security

There is much information on VoIP and how we can secure them mean all about VoIP security, but there is no standard measure about VoIP security, and there is no general solution. Here in this thesis, we tried to evaluate the problem and make the possible solution.

IP telephony, also commonly known as VoIP, is nowadays being used to transport or, say, carry the voice signal Ofir Arkin states in this article [3] mentioned that the security issue with VoIP is much greater than the standard PSTN regular telephone network. He bases this declaration on the reality that VoIP uses the IP protocol to switch signaling and voice and that the identical community may be used for each statistic and voice. There is the technology for setting apart statistics and voice networks indeed with digital LANs (VLANs). The identical community components are but used, making it susceptible to DoS or comparable attacks. VoIP vulnerabilities also are much more likely to be posted than relative weaknesses in PSTN as it is for a more on-hand generation for maximum people. Arfin additionally states that VoIP customers include more intelligence than standard PSTN phones. This can make those customers more likely to be compromised with the aid of using an attacker. One needs to additionally anticipate that a VoIP name is transferred thru extraordinary networks managed with the assistance of using outstanding entities, making it tough to decide the safety degree from one quit to another.

Accessibility is an integral part of VoIP. The availability of the conventional public switched telephone network is relatively high, and people who migrate to VoIP expect to decrease. However, this is not easy to achieve because availability depends on many components on the network, and many components are not required. There is no redundancy. Of course, redundancy makes the solution more expensive.

It involves no wonder that maximum contemporary-day corporations nowadays are already using telephony technology, VoIP. These virtual phones, with their superior capabilities and included networks, can shop each time and money. However, each corporation must comprehend that even though VoIP gives many productiveness and monetary benefits, precautions want to be taken to live blanketed from safety risks.

Securing a VoIP system may be more intricate than simply securing a straight information network, which will produce some vulnerability for the system. However, nobody ought to ever let security vulnerabilities detour them from exploiting an

information processing communication system. Since one in all the most significant security threats, eavesdropping or a "tapped" telephone line, can happen on each IP phone system and a POTS (Plain old telephone service). Luckily, knowledge is one of the most effective strategies of prevention, especially during this case. Here are a number of the highest VoIP security risks.

A VoIP implementation is exposed to various threats from various network levels and trusted areas within the network [4]. For instance, an attacker can attack VoIP in different ways and use a Denial of service attack, exploit a vulnerability to the SIP, or try to hijack VoIP calls through traditional hijacking, UDP spoofing, etc. categorized the attacks as follow. (Figure 2)

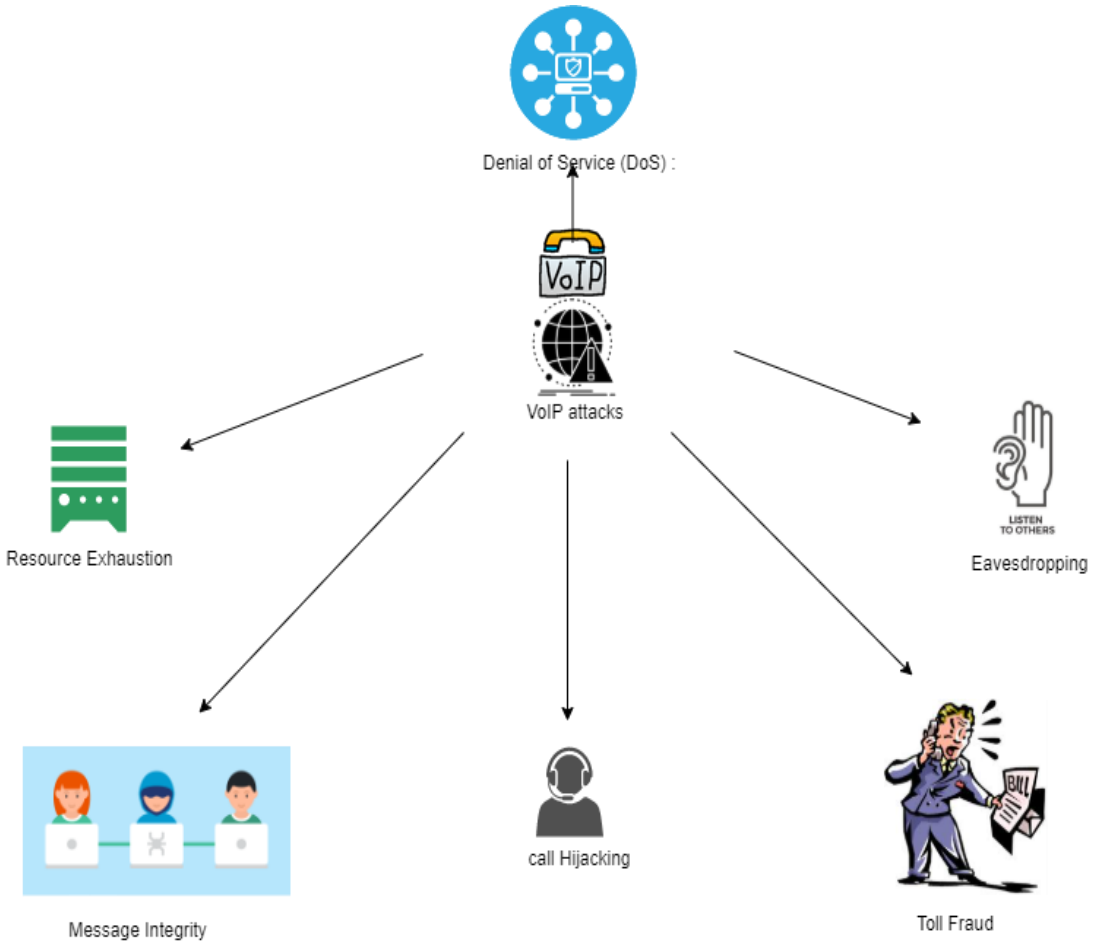


Figure 2. VoIP security threats

2.2 Call Hijacking: call hijacking in which attackers can spoof in SIP response or message (e.g., 301 Moved Temporarily) to hijack the existing running call towards poxy/endpoint doing so, attackers need to make a proper SIP header replication.

Registration hijacking occurs when an attacker gives a registrar a valid UA and replaces the legitimate registration with their address. This attack causes all incoming calls to be sent to the User-Agent registered by the attacker. Figure 3 illustrates registration hijacking.

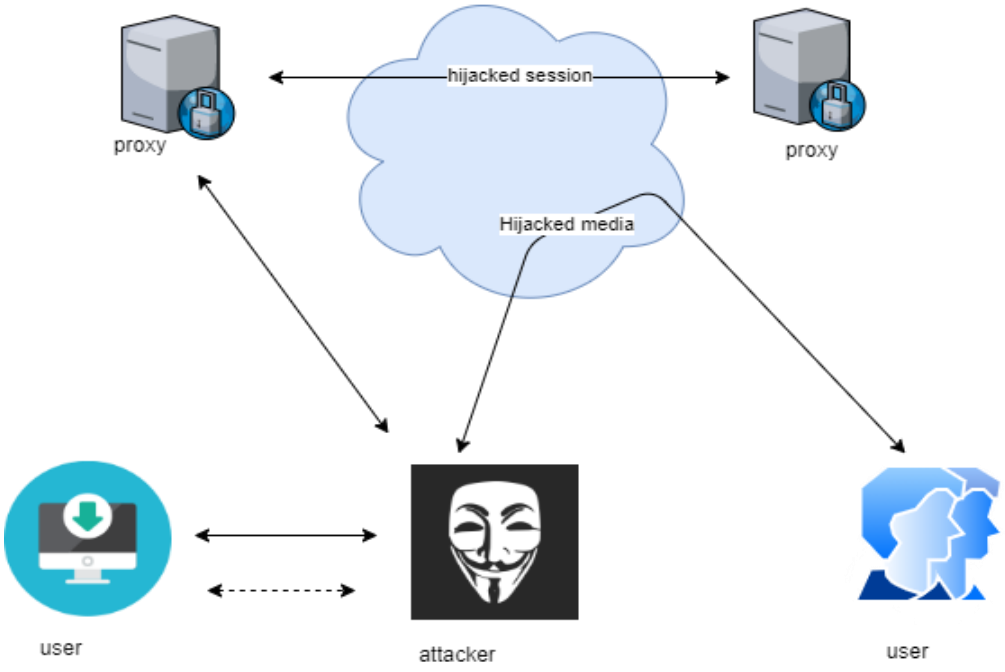


Figure 3. Registration hijacking process

Registration is usually done via UDP, which makes it easy to forge requests. No authentication is required, and it is generally insecure when it comes to authentication (only username and password). According to RFC3261, only registrars are "suggested" to object to registration requests. Query, or you need a simple query Username/password that may be attacked by dictionary. In a dictionary attack, the attacker will obtain one of your usernames and then view a list of possible passwords based on your company's knowledge. External attackers can create directories by

scanning your registered USER AGENT address. You can create an extended list and use the SIP message "OPTIONS" to create a user's hidden guide. In the case of some companies, they put very week id and password or auto-generated password some time which is just extension with an alphanumeric.

In such cases, attackers can guess your password or use a brute force attack to know the password, and the failed registration is not always logged. Your SIP proxy usually does not detect directory scan attempts or registry hack attempts. Registry hacking may cause the loss of calls to legitimate UAs, which may be one of the user's mobile phones or a vital resource, and this Unauthorized UA can also perform MIML attacks (man-in-the-middle), in which there is transparency between the calling UA and the called UA and signaling and media can be collected and changed. Another type of MIML attack is to redirect incoming calls to the bearer. Toll fraud gateway.

2.3 Resource Exhaustion Is also known as memory exhaustion, When the SIP server receives the Flood INVITE, its memory starts to use it, and it has saved a copy of the Flood message in its internal buffer, so it can process the news until it receives BYE and OK to complete the call. Messages used for further processing, such as Used for digest authentication[5]. In addition, some SIP servers are configured as stateful servers. For this reason, the content of the previous message and the current message must be retained throughout the session, for example, If the communication channel includes firewall or NAT traversal [5]. The SIP message is different, and call confirmation is usually kept between one second to a few seconds; this may cause problems with the server and cause insufficient memory exhaustion.

2.4 Eavesdropping is an act of deliberate eavesdropping in a conversation, usually without the best intentions. Although the law now includes VoIP phone systems, this is not the current trend, as demonstrated by the SIPtap attack in 2007 and the Peskyspy Trojan in 2009. Cybercriminals have closely monitored VoIP from the beginning. This attacker can perform VOMIT (voice-over misconfigured Internet

[5] Digest authentication - **Authentication-Info** mutual authentication with HTTP Digest. a UAS may include this header field in a 2xx response to a request that was successfully authenticated using digest based on the Authorization header field.

telephones) with local access to the VoIP LAN may sniff the network traffic and decrypt the voice communication.

2.5 Message Integrity the attackers may conduct a man-in-middle attack and later the original communication between two callers. Message integrities refer to the prevention of any unauthorized modification in voice packets.

2.6 Toll Fraud an attacker can hack into the system and used it for long calls, and these calls are paid for which provider or the attacked system user must pay the price.

2.7 Denial of Service (DoS) DoS is a Denial-of-Service attack that means shut down a machine or a network and making it inaccessible to users. DoS attack was mainly driven by flooding with traffic. There are generally two methods by which Dos attack works, surging or crashing, and the most common DoS attacks are – malformed some time know *as fuzzy SIP messages, flooding attack*

2.7.1 Malformed Message or Fuzzy Message

Malformed messages are a specific type of message that does not contain a mandatory field. Although some SIP devices can handle these types of attacks nowadays, some cannot manage such a message and lead to a system crash.
Error messages are log messages for syntax errors. The attacker changed the correct SIP message, causing the server to crash when the server tried to process the message.

Before the attackers attempted a Fuzzy attack, they discovered the function of SIP by sending REGISTER and OPTIONS responses. As we know, these transport protocols provide information about SIP user agent functions. respect. [6] In this case, the attacker sent the OPTION Figure () message sent to the target SIP server. The target server replies with an OPTIONS message. In this way, the attacker can detect the implemented SIP target method—for example, the provider's SIP return function and the version of the potential SIP destination.

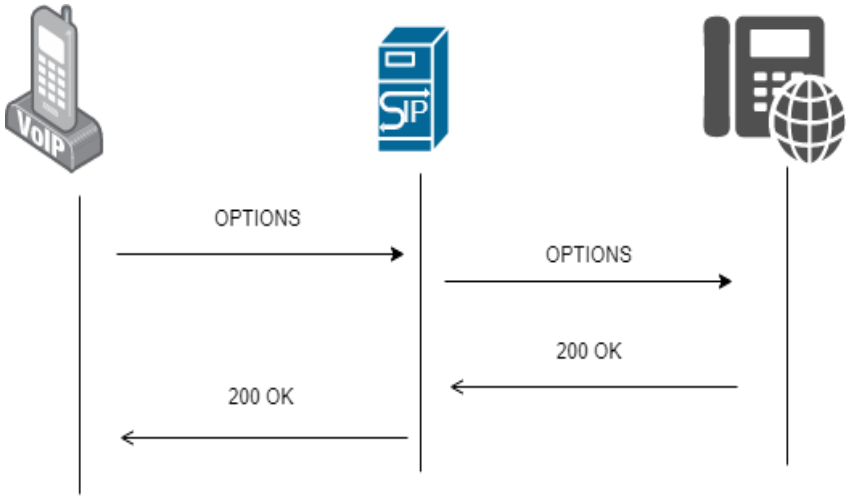


Figure 4. Discovering User Agent server "Capabilities."

Figure 5 and Figure 6 is an example of malformed message found in *INVITE* *sip: null* and sip 'or' this means called party address is missing in this case server sends a lousy request indication and reject the message. Still, some time under some situations, the server failed to respond to the awful request.

```
Session Initiation Protocol (INVITE)
  Request-Line: INVITE sip:+0046842002734@195.39.17.246 SIP/2.0
  Message Header
    Via: SIP/2.0/UDP 193.46.255.78:61921;branch=z9hG4bK526367103
        Transport: UDP
        Sent-by Address: 193.46.255.78
        Sent-by port: 61921
        Branch: z9hG4bK526367103
    Max-Forwards: 70
    From: <sip:'or''='@195.39.17.246>;tag=518514814
    To: <sip:+0046842002734@195.39.17.246>
    Call-ID: 1533541714-494635054-1342103608
    [Generated Call-ID: 1533541714-494635054-1342103608]
    CSeq: 1 INVITE
    Contact: <sip:'or''='@193.46.255.78:61921>
    Content-Type: application/sdp
    Content-Length: 211
    Allow: ACK, BYE, CANCEL, INFO, INVITE, MESSAGE, NOTIFY, OPTIONS, PRACK, REFER, REGISTER, SUBSCRIBE, UPDATE, PUBLISH
    User-Agent: Linksys-SPA942
  Message Body
```

Figure 5. An example of Fuzzy Sip message

Figure 6. An example of Fuzzy Sip message

The sequence of formatted string/ANSI-escape char/invalid UTF-8 can also cause a buffer overflow in the SIP server. Figure 7 shows the example of SIP with unwanted characters.

Figure 7. An example for the wrong formatting, where the most significant number of unnecessary characters has append

The last example in Figure 8 of fuzzy attack is SQL injection, where attackers inject malicious code for gaining access to the SIP database.

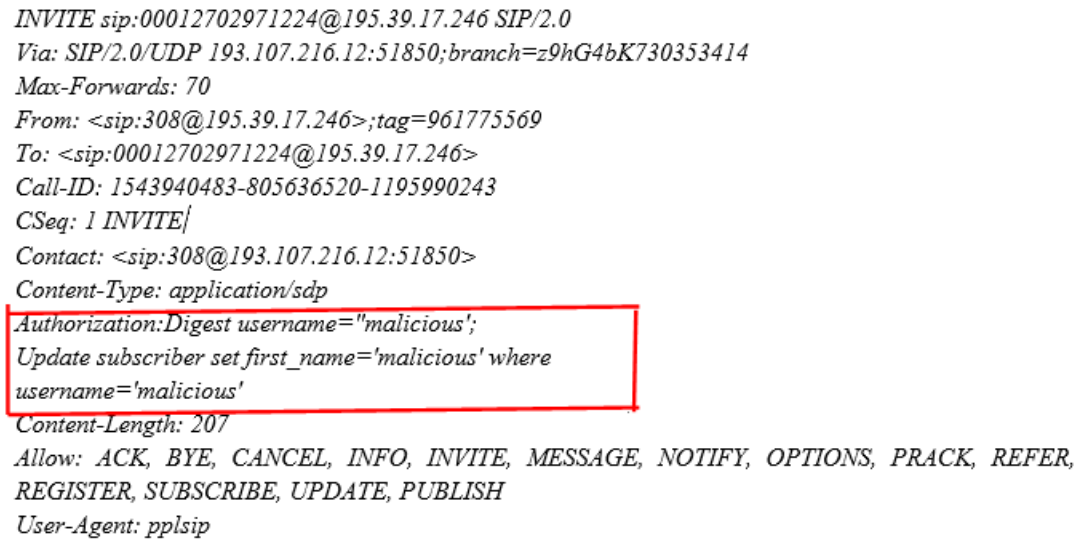


Figure 8. an example of a SIP message infected by SQL injection.

We can approach specific approaches in this situation of attack. The first approach is to use a set of predefined malformed messages, and the other is to use a "malformed engine" that can pre-defined the messaged as per the definition of the system.

2.7.2 DoS Flood Attack.

The principle of these types of attacks is elementary; the primary purpose of these attacks is to exhaust the system. The most often targets are the CPU and memory; attackers generate a considerable number of SIP messages. It can either be INVITE, REGISTER, OPTION. It does not matter, and these messages are being sent periodically to the target server. When it reaches the target server, the server tries to act it as a standard message and start acting upon request, but due to the flood, it is soon out of resource and cannot perform further action and soon may break down.

Creating a SIP invite flood is easy; one needs a generator of SIP messages; in the market, there are many sip generators available freely are SIPp and SIPr

Generally, a flood attack is an attacker that sends a large amount of traffic to a specific server or system to slow down its speed by slowing down the rate of memory, CPU, and bandwidth. Keeping the target SIP server busy and unable to process legitimate call requests makes it difficult for ordinary users to use the SIP server. In this case, valid data packets will be discarded or processed slowly, making VoIP service impossible. Generally, SIP can only handle several requests at the same time. However, in many invitations, the device will be overloaded due to memory usage, and the processing processor will be empty [7]. When the memory and processor are used up, the communication channel will be overloaded, which can be a problem for the server because it makes the system vulnerable to attacks. Here, we only discuss one of them, such as INVITE Flood. In figure (9), the attacker generates an INVITE FLOOD attack by sending many SIP invitation messages to the server. The SIP server is responsible for processing these false messages, which makes the server busy. When an attacker sends a stream of notification messages, the SIP device will be overloaded and consume memory. In addition, the communication channel between SIP and ordinary users is overloaded, which leads to DoS between users and agents.

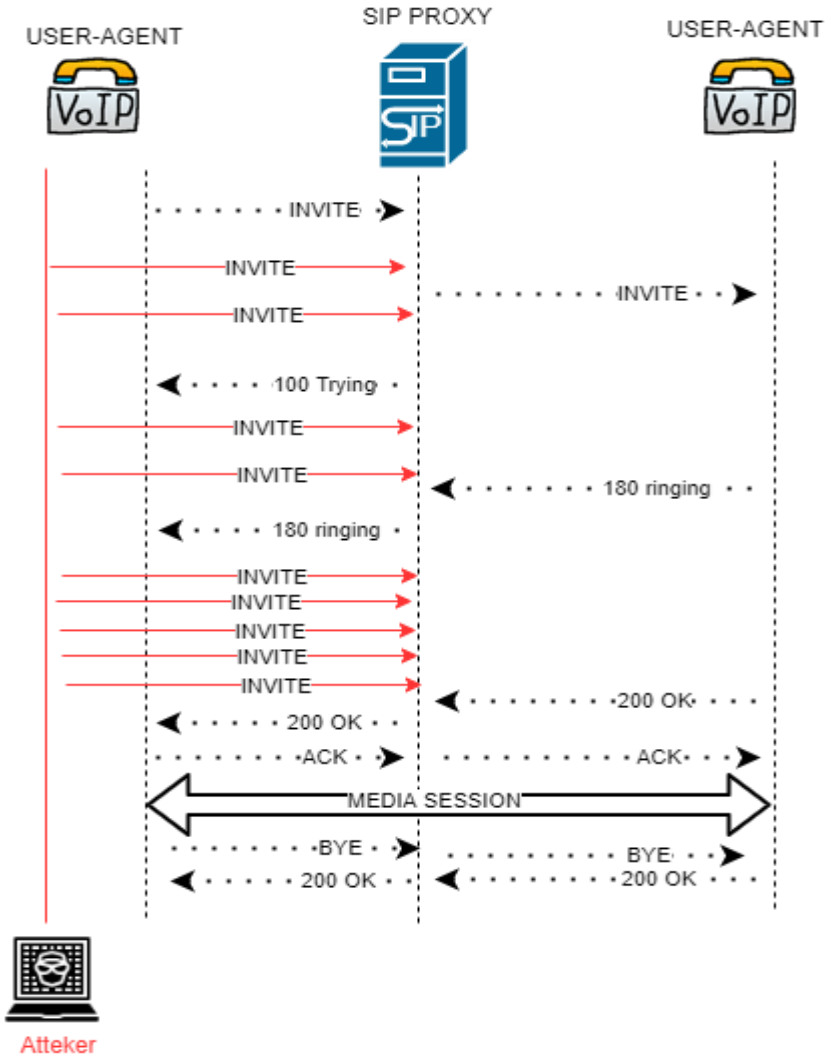
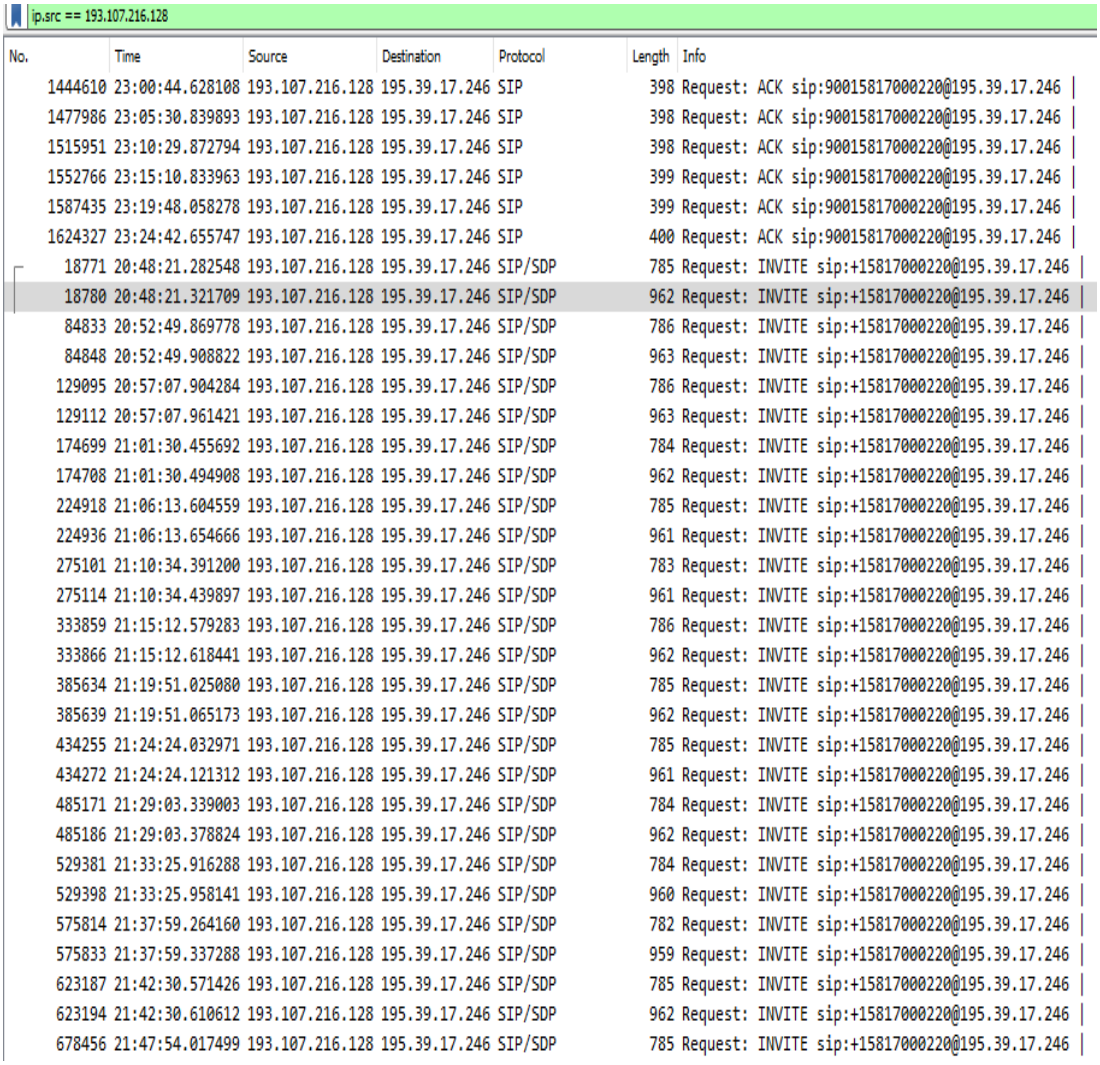


Figure 9. an example of SIP INVITE flood

Two significant impacts are resulting from a SIP flooding attack which is Memory exhaustion and CPU exhaustion.



ip.src == 193.107.216.128

No.	Time	Source	Destination	Protocol	Length	Info
1444610	23:00:44.628108	193.107.216.128	195.39.17.246	SIP	398	Request: ACK sip:90015817000220@195.39.17.246
1477986	23:05:30.839893	193.107.216.128	195.39.17.246	SIP	398	Request: ACK sip:90015817000220@195.39.17.246
1515951	23:10:29.872794	193.107.216.128	195.39.17.246	SIP	398	Request: ACK sip:90015817000220@195.39.17.246
1552766	23:15:10.833963	193.107.216.128	195.39.17.246	SIP	399	Request: ACK sip:90015817000220@195.39.17.246
1587435	23:19:48.058278	193.107.216.128	195.39.17.246	SIP	399	Request: ACK sip:90015817000220@195.39.17.246
1624327	23:24:42.655747	193.107.216.128	195.39.17.246	SIP	400	Request: ACK sip:90015817000220@195.39.17.246
18771	20:48:21.282548	193.107.216.128	195.39.17.246	SIP/SDP	785	Request: INVITE sip:+15817000220@195.39.17.246
18780	20:48:21.321709	193.107.216.128	195.39.17.246	SIP/SDP	962	Request: INVITE sip:+15817000220@195.39.17.246
84833	20:52:49.869778	193.107.216.128	195.39.17.246	SIP/SDP	786	Request: INVITE sip:+15817000220@195.39.17.246
84848	20:52:49.908822	193.107.216.128	195.39.17.246	SIP/SDP	963	Request: INVITE sip:+15817000220@195.39.17.246
129095	20:57:07.904284	193.107.216.128	195.39.17.246	SIP/SDP	786	Request: INVITE sip:+15817000220@195.39.17.246
129112	20:57:07.961421	193.107.216.128	195.39.17.246	SIP/SDP	963	Request: INVITE sip:+15817000220@195.39.17.246
174699	21:01:30.455692	193.107.216.128	195.39.17.246	SIP/SDP	784	Request: INVITE sip:+15817000220@195.39.17.246
174708	21:01:30.494908	193.107.216.128	195.39.17.246	SIP/SDP	962	Request: INVITE sip:+15817000220@195.39.17.246
224918	21:06:13.604559	193.107.216.128	195.39.17.246	SIP/SDP	785	Request: INVITE sip:+15817000220@195.39.17.246
224936	21:06:13.654666	193.107.216.128	195.39.17.246	SIP/SDP	961	Request: INVITE sip:+15817000220@195.39.17.246
275101	21:10:34.391200	193.107.216.128	195.39.17.246	SIP/SDP	783	Request: INVITE sip:+15817000220@195.39.17.246
275114	21:10:34.439897	193.107.216.128	195.39.17.246	SIP/SDP	961	Request: INVITE sip:+15817000220@195.39.17.246
333859	21:15:12.579283	193.107.216.128	195.39.17.246	SIP/SDP	786	Request: INVITE sip:+15817000220@195.39.17.246
333866	21:15:12.618441	193.107.216.128	195.39.17.246	SIP/SDP	962	Request: INVITE sip:+15817000220@195.39.17.246
385634	21:19:51.025080	193.107.216.128	195.39.17.246	SIP/SDP	785	Request: INVITE sip:+15817000220@195.39.17.246
385639	21:19:51.065173	193.107.216.128	195.39.17.246	SIP/SDP	962	Request: INVITE sip:+15817000220@195.39.17.246
434255	21:24:24.032971	193.107.216.128	195.39.17.246	SIP/SDP	785	Request: INVITE sip:+15817000220@195.39.17.246
434272	21:24:24.121312	193.107.216.128	195.39.17.246	SIP/SDP	961	Request: INVITE sip:+15817000220@195.39.17.246
485171	21:29:03.339003	193.107.216.128	195.39.17.246	SIP/SDP	784	Request: INVITE sip:+15817000220@195.39.17.246
485186	21:29:03.378824	193.107.216.128	195.39.17.246	SIP/SDP	962	Request: INVITE sip:+15817000220@195.39.17.246
529381	21:33:25.916288	193.107.216.128	195.39.17.246	SIP/SDP	784	Request: INVITE sip:+15817000220@195.39.17.246
529398	21:33:25.958141	193.107.216.128	195.39.17.246	SIP/SDP	960	Request: INVITE sip:+15817000220@195.39.17.246
575814	21:37:59.264160	193.107.216.128	195.39.17.246	SIP/SDP	782	Request: INVITE sip:+15817000220@195.39.17.246
575833	21:37:59.337288	193.107.216.128	195.39.17.246	SIP/SDP	959	Request: INVITE sip:+15817000220@195.39.17.246
623187	21:42:30.571426	193.107.216.128	195.39.17.246	SIP/SDP	785	Request: INVITE sip:+15817000220@195.39.17.246
623194	21:42:30.610612	193.107.216.128	195.39.17.246	SIP/SDP	962	Request: INVITE sip:+15817000220@195.39.17.246
678456	21:47:54.017499	193.107.216.128	195.39.17.246	SIP/SDP	785	Request: INVITE sip:+15817000220@195.39.17.246

Figure 10. SIP Invite DDoS attack

But there is a difference in the case of REGISTER flood; in a normal DoS attack, the server's memory and CPU are exhausted, but in REGISTER, it gets affect to nonces value.

During communication, SIP authentication guarantees that the sender of a SIP message is authenticated means they have an identity that they are claiming. As the SIP server receives the INVITE message instead of the INVITE request, the server sends a 407 AUTHENTICATION response to the caller. The 407 message contains a "nonce" value, a random string generated by the server for the caller. Both sip server

and UAC share a secret password in between. The caller uses the nonce value to create a unique response.

When the UAC sends the request to connect again with the unique response value, which the server will use to authenticate the recommendation using this method, the password cannot be seen as a text format, and MD5 is used to create the unique text response.

There are two different forms of failure to authenticate 4xx challenges response, i.e., 401 or 407

If the origin server does not accept the credentials sent with a request, it should return 401 (unauthorized) responses. The response must include a WWW-Authenticate header field containing at least one challenge applicable to the requested resources. If a proxy does not accept the credentials sent with a request, it should return 407 (proxy authentication required). The response must include a proxy-authenticate header field containing a challenge applicable to the proxy for the requested resource.

If the user uses a valid id in case of REGISTER flood, it leads to the exhaustion of nonces.

The SIP registration looks as follow

1. The calling party send a REGISTER request to the server (without a digest)
2. The server responds with 401, and this means server does not accept the credentials sent with a request.
3. The caller made its authentication digest and sent a new REGISTER request with this time valid digest.
4. The server sends 200 OK and places this information into the system,

For this reason, it gets exhausted since the server will generate the nonce value until and unless the registration process continues or until the end of life.

The formula for generating the response is complex.

HA1 = MD5(username:realm:password)

HA2 = MD5(method:digestURI)

Response = MD5(HA1:nonce:HA2)

Example of the nonce generator.

In this example, *user = patrol-cz*, a password is "*patrol-czpassword*" and let generate a response to the server.

Here the user name is "*patrol-cz*," the realm which is "*sip.voiparound.com*," digest URI is *sip:sip.voiparound.com , the* method is Register and the password which is *patrol-czpassword.*

Now generate the first path HA1 by the formula

HA1 = MD5(username:realm:password)

HA1 = MD5 (*patrol-cz: sip.voiparound.com: patrol-czpassword*)

Now, if we put the HA1 value in the MD5 generator tool, here, in this case, I used the first tool found on the website, and we got *HA1= 39c700680ffe3c8e97fc779c628d3810*

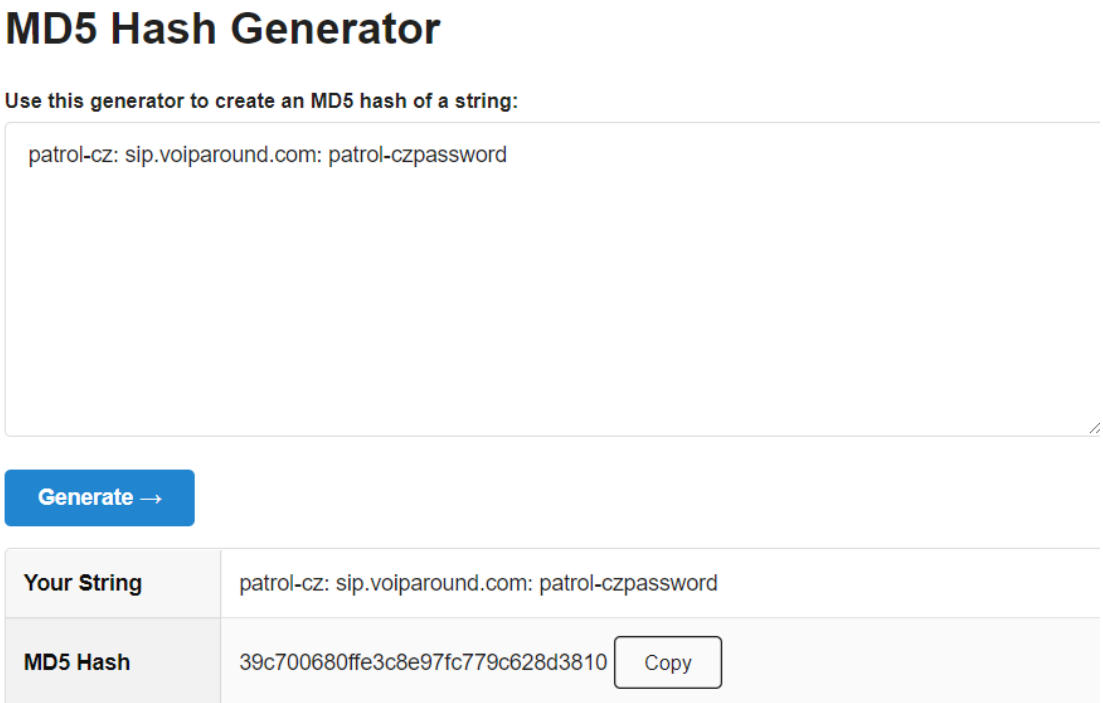


Figure 11. MD5 generation for HA1

Now for HA2, we know the method is Register, and our digest URI is *sip:sip.voiparound.com*

HA2= MD5(method:digestURI)
HA2 = MD5(REGISTER: *sip:sip.voiparound.com*)

Now again if we put this value in MD5 generator, the output is e5704bb95989a86bfeb15fdaa8d0a86a

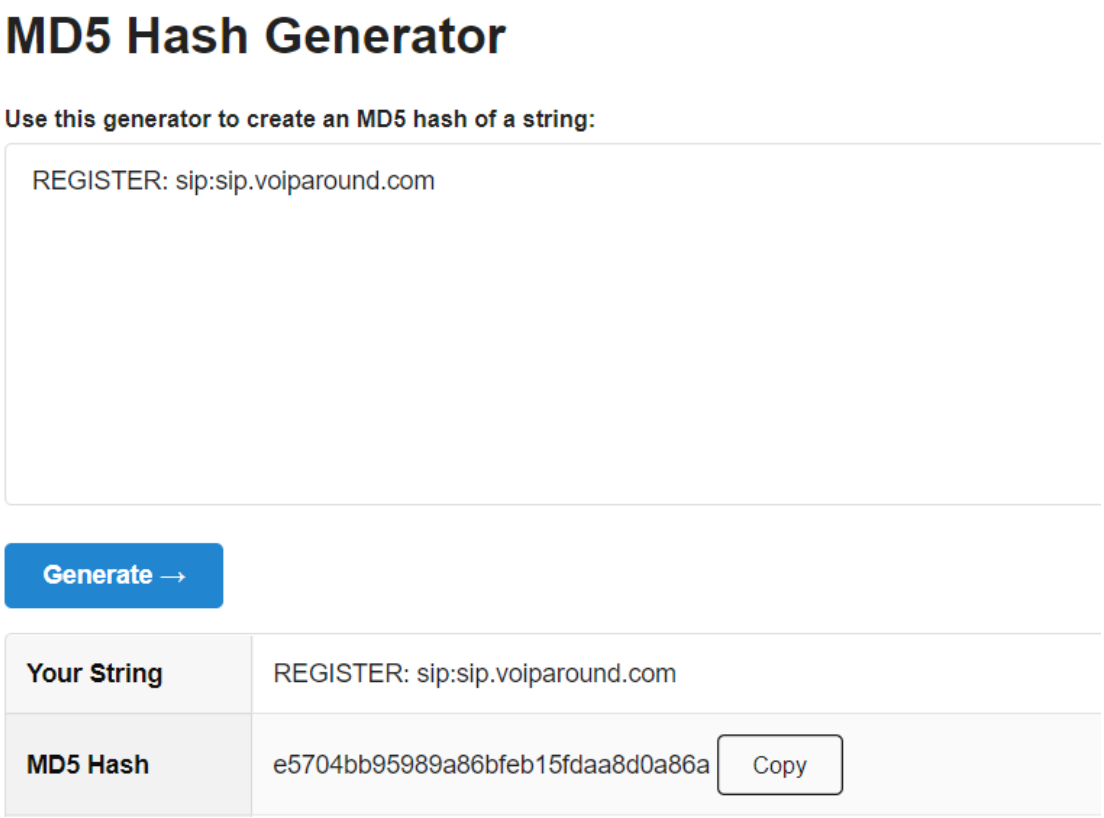


Figure 12. MD5 generator for HA2

Now, if your join HA1+HA2 in one string and has it, you will get the response
Response = MD5 (HA1:nonce:HA2)
Response = MD5 (*39c700680ffe3c8e97fc779c628d3810: 4206259671:* e5704bb95989a86bfeb15fdaa8d0a86a)

Which will give the final response

Response = MD5 (64ab5b3cce4353e3d8baa491beb60a7f)

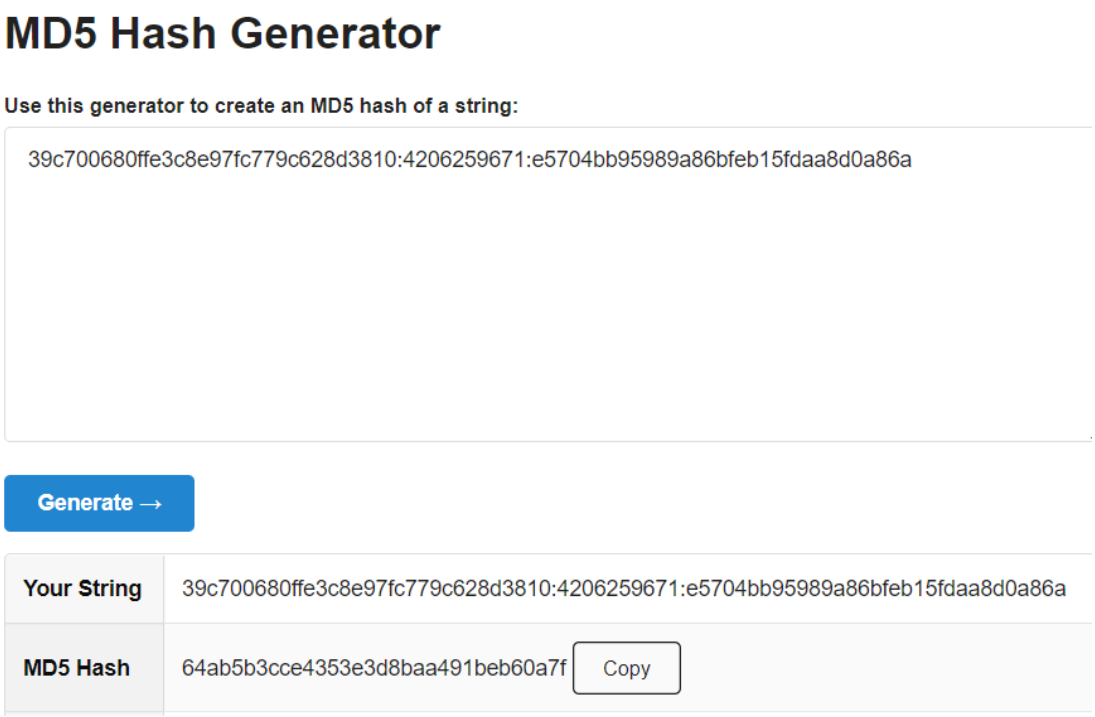


Figure 13. MD5 for Final Response (HA1:nonce:HA2)

So, when the user sends this response to the SIP server, he will get a 200 OK response and hence user is registered on the server.

We have seen many flood attacks during my work, a few of them in the following chapters. But as we talk about the mitigation of these attacks, I have created my script, which can be helpful for these flood attacks.

3.1 Network Intrusion Detection Systems

The goal of an Intrusion detection system (IDS) is to "identify, preferable in real-time, unauthorized access, misuse and abuse of computer buy both system insider and external penetration" [9].
An IDS is used as a substitute for building a shield around the network. The shielding effect can be decent
In several ways, including failure to prevent attack from an insider.
However, while IDS technology progresses, methods to circumvent IDSs are becoming more prevalent [10]. For example, Wagner and Soto develop a class of mimicry attacks that mimic the application's actual behavior [11]. Considering these attacks and the rising frequency of encrypted communication, alternatives such as honeypots have become more popular.

3.2 Honeypot.

The exact definition of a honeypot is debatable. However, some of the writers and magazines put their intentions as follows.

As per the book *Honeypot: Tracking Attackers* by *Lance Spitzner,* the Honeypot is defined as
"A honeypot is a security resource whose value lies in being probed, attacked, or compromised"

A more practical but limiting definition is given by pcmag.com.
"*A server that is configured to detect an intruder by mirroring a real production system. It appears as an ordinary server doing work, but all the data and transactions are phony. Located either in or outside the Firewall, the Honeypot is used to learn about an intruder's techniques as well as determine vulnerabilities in the entire system*".

In its simplest form, A honeypot is a system designed to attract attackers, and this information allows security researchers and network defenders to analyze network-based attacks better. Honeypot has no production value beyond research.

Honeypots are far different from most traditional security tools in that they can take on various manifestations. Most of the security devices or technologies are dedicated only to any specific problems. Let us have an example of a security solution, Firewall; this is often used to protect your organization by controlling what traffic flow you want to input or access your employees. It is generally deployed around an organization's perimeter to block unauthorized activities. The second example is Network Intrusion Detection systems, designed to detect attacks by monitoring either system or network activities. they are also used to block unauthorized activities.

But Honeypots are different in that they are not limited to solving the problem if you are; they are not meant to solve the problem; they develop a way to identify the problem. These tools or software or system, whatever you say, it is up to the development are highly flexible tools that can be applied to various situations. Therefore, we cannot define the definition of Honeypot because of its extensive uses. For example, a honeypot can be used with Firewall and network intrusion system by sharing the attack pattern, target system, and fingerprint; it also captures and analyzes automated attacks, such as worm, DDoS, and act as early as possible to capture and analyze and warned. However, the Honeypot is all up to the person making it or the industry targeting it to use it and in which environment. However, all the action-reaction share only a common feature: their value lies in being prob, attacked, and conceding.

3.3 How Honeypot works.

Honeypot work behind the Firewall. Honeypot's primary job is to simulate the services we need to protect and induce various attacks. When anyone or unauthorized user tries to enter the system with a fake identity with the help of the honeypot system, they do not directly see the actual design but instead attack replicating the procedure. According to the Web Application Security Project (OWSP), some top attacks recorded were SQL injection and broken Authentication [12]. When someone tries to enter, a log is generated about all the incoming entries. Even though the attackers can enter the system and capture the data from the database, we can fool them by providing fake or manipulated data, which the Honeypot does. We can save, and most importantly, we can know the behavior of attack, type of attack, patter, and

fingerprint, and by this, we can create a system that maps these patterns and avoids being hacked.

There is always a misconception about the honeypots that are the device used to attract the attack, but this is not actual. Honeypot does not advertise himself to attracting or to hacked, instead of that most honeypot capture the data/traffic or any activity that is within the range of this. Some of them are designed to be the same as the server's configuration found in any industry.

4.1 Type of Honeypot.

Honeypots can be classified based on their purpose (*production, research*) and level of interaction *(low, medium, and high*).

4.2 Based on the purpose

Production honeypot is the most common type of Honeypot used to collect cybersecurity-related information within a business or organization's production network. These are implemented within your organization because they can secure your atmosphere, such as detecting attacks. We can also say that production honeypot real deal is to find bad guys. Generally, these are placed inside the production network with other production servers by an organization to improve their security.

Typically, production honeypot is considered low interaction honeypots, which are easy to deploy on the network. Because they require less functionality than research Honeypot, they are generally easier to create and implement. Although they determined the attack method, they provided less information about the attacker than the research Honeypot. You can find out which system the attacker came from and what attacks are being launched, but you may not know who they are, how they are organized, or what tools are used.

Research Honeypot A research honeypot is designed to gain information about the Blackhat community and not directly value an organization [14]. They gather information about common threats that businesses may face to better protect themselves from these threats. Its principal function is to study the attackers' progress

and establish their lines of attack. It helps to understand their motives, behavior, and organization. Examine honey pots. They are complex to implement, maintain, and collect large amounts of data. Research honeypot is typically used by organizations such as universities, government, militaries, or large corporations interested in learning more about threats research.

Research honeypots add tremendous value to research by providing a platform for investigating cyber threats. You can observe the behavior of intruders and gradually record them as they attack and compromise the system. This intelligence is one of the most unique and exciting characteristics of Honeypot [13]. Valuable tools for developing analytical and forensic skills. Sometimes they can even help discover new worms.

4.3 Based on interaction.

High Interaction honeypot – in high interaction honeypot, the attackers interact with the device or system as they would in real or regular, any command or application an end-user would be expected to be installed, and there is almost no restriction placed on what the attackers can do once they are inside the system. But the only difference is that these devices are used to analyze data, and harm cannot be done in the virtual environment. Such interaction aims to get the input or behavior of hacking to target the data they need about their fingerprint.

Low Interaction honeypot – A low-interaction honeypot simulates only services that cannot be exploited to gain total access to the Honeypot [15]; these types of honeypots are easy to maintain. Since they only consume few resources, multiple virtual machines can quickly be done on a single system, short response timeless code is required. An example of low interaction honeypot is Honeyd[6].

[6] https://github.com/DataSoft/Honeyd
http://www.honeyd.org/

5.1 honeypot role in security: detection, analysis, and mitigation

Now we have come across the type based on purpose and interaction of Honeypot. now this is the right time to understand the security concern related to a honeypot; we will break down the category into two parts

Detection and response and more detail and specific details are defined in Bruce Schneier's Secrets and Lie digital security in a networked world [16]

5.1 Detection

Detection in security plays the most crucial part; suppose you have some intrusion detection software that works on logs, and if the software cannot detect the intruders, it would be difficult for the owner to be hacked. Since attackers enter the system layer by layer and catch it on early-stage harm, it will be less. But if we incorrectly attack the Honeypot, it may introduce risk, providing an attacker a window into an organization.

In network security, we have many challenges because sooner or later, a supportive system will fail, and attackers will get access inside. There can be a variety of reasons for yielding. Some of them are Firewall is configured poorly, users are using a very poor password, the window is without updated patching there for this is not proper to be saying that we can decrease the attacking into 0%. Still, instead of that, we can detect, analyses, and avoid them.

Numerous systems in the market help detect the malware, but how can a honeypot help see suspicious activities. While Honeypot is limited to prevention, but it adds an extra layer in detection.

Detection has always been an arduous task for the industry, either with detections are false positive false negative or data aggregation. [13]

False-positive: according to Wikipedia, A *false positive* is an error in binary classification in which a test result incorrectly indicates the presence of a condition such as a disease when the disease is not present

False-negative: is the opposite error where the test result incorrectly fails to indicate the presence of a condition when it is present. Or when an organization fail to detect the attack

Data aggregation: centrally collecting all the data used for detection and then corroborating that data into valuable information [13]

5.2 Analysis

Once the detection is successful, the next step is to analyze the data we capture during the detection step. Whenever the attackers break into the system, in 90% of cases, they leave the footmark of attacking. This footmark is helping to analyze how they get into it, what they did after going inside the system, and this is the information that is very difficult to capture. Without it, an organization cannot defend or respond quickly.

However, even if attackers hide their actions and modify the log file, these cations can be tracked. Each file maintains its last modified date/time on most operating systems, but sometimes these can very quickly be polluted and worthless. And here, the Honeypot comes as a rescue party; if the Honeypot gets hacked, the only thing compromised is that the actual activity on the system is the attackers' activity. Honeypot can quickly be taken offline in case of attack because the attackers will no longer pollute it for further investigation. Also, Honeypot provides no production service, and organizations can quickly be taken down for further study or analysis.

5.3 Mitigation

One of the most significant challenges in the network security community is mitigating or tackling these attacks. Attacks are never on the same side; it is constantly branching out. The industry is spending billions of dollars just for gating information and ease them, but some time in security, we deal with only very little knowledge of the attack.

The main problem of not having ease of attack is data, traditionally, we are taking information whenever some hacking incident happed, and they leave the fingerprint behind it. Still, sometimes as we already said earlier, it is not 100% good chances are they pollute the data, and we have nothing in our hands. But we also should say these

techniques are helpful; you can find how hacking process and methods. Honeypots can help get data without having a security issue to the central server; someone can collect data, analyze it, and take proper steps to stop them.

The benefit of having honeypot data is that you can predict how and when the attack will happen. This can be done by deploying multiple honeypots in-network, and by using these data, you can ease the attack.

6.1 Honeypot Deployment strategies.

Deployment of Honeypot can have many strategies, from installing a single honeypot to creating a whole network of honeypots known as a honeynet. The deployment depends on the amount the data required and the investment.

One of the most command deployments for Honeypot is facing the Internet. This scenario is typically used if a honeynet is set up for research purposes, captures malware samples for further analysis, tracks network worm activity, or studies an attacker's behavior. [19] this includes the study of malicious activities with the chance of learning new activities about vulnerabilities and exploits. For this, the Honeypot should be accessible from the outside or inside the DMZ figure (14). Another deployment can be placed in the production network segment in the case of compromised systems and learn about internal infections [19]

Other, Honeypot is deployed inside the production environment to detect the compromised system and study their behavior and sometimes call it an insider threat. for this purpose, the Honeypot can be placed inside In a different LAN segment and assigned the previously unallocated IP address. putting Honeypot inside the production environment can lead to the compromised system and can help to study or monitor closely about the if other methods are also affected or compromised in the background, and so-called case Advance persistent threats (APTs)[7]

6.2 Virtual/physical Honeypot

To gather the intelligence about the attackers in a domain, network admin or system admin decoy systems known as a honeypot, which is used to deceive the unauthorized access into the system.

A honey pot can be placed in multiple locations on a network, including the DMZ. To deceive an attacker and allow them to interact with the system, a honeypot needs to be the mirror image of the original host.

[7] advanced persistent attack (APT) uses continuous, clandestine, and sophisticated hacking techniques to gain access to a system and remain inside for a prolonged period of time, with potentially destructive consequences.

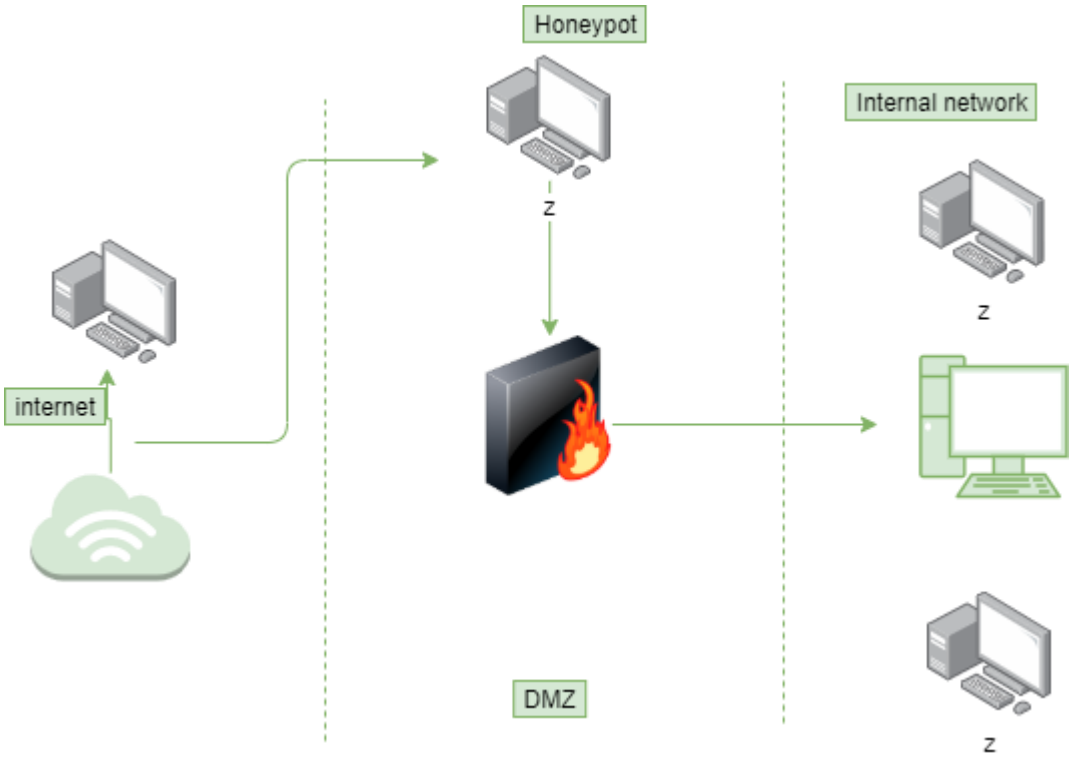


Figure 14. honeypots deploy locations.[17]

Developing your Honeypot is not as complicated as it looks. One only needs a server with PBX if it is asterisk configuration and Internet and some basic coding for finding fingerprints. But at the end of the day, it all depends on what you require for your hands-on practice of the technology you are comfortable with.

Honeypot can be classified according to whether it is in a virtual environment or a physical environment. Cheating in a virtual environment has several advantages: downloading the ISO file and starting the computer with the virtual operating system. The advantage of running virtual software is that it is a snapshot. The administrator can take a shot of the running state of the computer and roll it back in a short time. It is also helpful if an attacker can break out of the VM; they can then determine what environment they are running in and proceed with caution [16].

Using a physical Honeypot can be very beneficial because it can make the attacker believe that you are on an actual device. Generally, if we are satisfied with this, virtual honeypots have limited memory and processing power and are found more frequently than physical honeypots. Servers other than the physical server can be

backed up regularly to track what is happening on the attacked server. However, compared with virtual servers, the construction cost of physical servers is higher.

6.3 Our Honeypot and system configuration.

Our Honeypot is a virtual machine running on a physical server located in one of the server rooms in the Department of Telecommunications in the main FEE building in Prague dejvice, technika 2.

This Honeypot is a virtual and physical host and runs on Sinux distribution (Professor Mr. Pavel Troller own Linux distribution), and Virtualization is made by qemu/KVM. KVM is a kernel-based machine, an open-source virtualization technology built into Linux. Specifically, KVM let you turn Linux into a hypervisor that allows a host machine to run multiple, isolated virtual environment called guest or virtual machine. KVM uses a combination of security-enhanced Linux (SELinux) and secure Virtualization (sVirt) for enhanced VM security and isolation. SELinux establishes security boundaries around VMs. sVirt extends SELinux's capabilities, allowing Mandatory Access Control (MAC) security to be applied to guest VMs. And preventing manual labeling errors. [18]. The Asterisk distribution, a software implementation of a private branch exchange (PBX), replaces my Masterisk (Professor Mr. Pavel Troller own modified asterisk system known as Masterisk), which runs on a virtual machine primary application. The virtual machine is directly connected to the DTT technological intranet. It has only a private IP address, and the public IP address is tunneled to it from the server in T-Mobile Premises.

7.1 PBX (Asterisk)

A PBX stands for "*private branch exchange*," is a telephony system to have internal switching.

A PBX is a phone switch located at the customer premise, and it comes in different sizes, from being a 2 line to thousand lines. It acts as the central switching system for phone calls within a business. IP PBX system handles internal traffic between stations and acts as the gatekeeper to the outside world.

A Hosted PBX, sometimes call it as Virtual PBX, offers similar functionality and features. Still, the only difference is that the switching is placed in a central location, but the phone is at customer premises. PBX is in the customer premises example, a business office, and a desktop phone is connected to it. Initially, it was a complex command line, but as technology grew and made life more accessible, the web technologies also improved. More PBXs entered the segment with GUI (graphical user interface). User devices can regularly change IP. Therefore, a device needs to register with a SIP server periodically. This can be to a PBX to reconfirm where to send invites (for receiving or making a call) [20]

7.2 Asterisk as an alternate for traditional PBX

If you are thinking of making your PBX costly for you, you need specialized switches and hardware and a proprietary system that doesn't come cheap and might not even offer all the telephony features you are looking for.

But there is always an alternative; as thousands of business and network administrators have found, the open-source Asterisk PBX has gained tremendous popularity, offering surprisingly powerful telephony features on less expensive or can say inexpensive hardware, when we compare with traditional PBX. It not only has been saving the companies money but has also been able to integrate telephony with the network applications in a way that previously might be not possible [21]

Let's start with basic what *Asterisk* is: open-source PBX software that runs on various operating systems that include Windows, Linux, Mac OS, OpenBSD, FreeBSD, and Sun Solaris. It can run on inexpensive off-the-shelf hardware. It includes high-end

features such as interactive voice response, voice mail, conference calling, and automatic call distribution and routing that have only been available with traditional PBX.

It is also exceedingly flexible. The new function can be created by writing scripts in Asterisk's language, by writing a module in C and particular importance is that it can handle VoIP calls and work with various VoIP protocols, including Session Initiation Protocol (SIP) and H.323, and act as a gateway between IP phones and the public switched telephone network.

7.3 Asterisk Introduction.

Asterisk is an open-source software PBX, created by Digium, Inc., and has a continuously growing user and developer base. Digium invests in both developing the Asterisk source code and low-cost telephony hardware that works with Asterisk. Asterisk runs on Linux and other Unix platforms with OR without hardware that connects your server to the traditional global telephony network, the PSTN [22].

Asterisk channels are used for various connections, both to VoIP protocol like SIP, H.323 and the hardware that connects to the PSTN like Zaptel, PRI, and other devices. Asterisk supports many VoIP protocols, including signaling protocols such as H.323 and SIP and media transmission protocols like RTP. Each channel supports one or more protocols. The media stream, the authentic voice on the network, can be encoded with many different codecs, from (G.711) to GSM and ILBC.

7.4 Configuration

For configuration of Asterisk for your device, you need a few requirements, e.g., the dial plan, the accepted prefix for the calling party, etc.; the dial plan is stored in a text file, the configuration file *extension. Conf*. each of the wings belongs to a context, either the default context or a specific context or specific as per your need, for example, incoming sip calls, long-distance outbound PSTN calls, and inter-office calls. All users connected to Asterisk belong to the context (specified in the channel configuration file). Asterisk is looking for advice on handling calls from that user, checking access to expensive lines with different rules for local users. And contacts called from outside.

In the dial plan, your installation all movements and conditions that the PBX ought to handle. You can configure contexts that paintings best all through a part of the day or night-time. You can consist of context from a different context and simplify or make a complex dial plan.

Here are some examples of what you can perform with the dial plan.

- If the user does not answer the primary or secondary call within 20 seconds, the call is connected to voice mail.
- Join a call in a multi-party conference.
- Transfer the call to another Asterisk PBX.
- Block calls from unknown or unwanted callers.
- Find data in the database based on the caller ID query and decide which agents answer the call.
- Create call queues and allow a group of agents to handle incoming calls.

Extensions. conf -The Dial plan

The configuration file "extensions. conf" contains the dial plan, master plan, or sequence of operations from Asterisk. Control the processing and routing of incoming and outgoing calls. The behavior of all connections through the PBX is configured here. The content of extensions. Conf is divided into sections that can be applied to static definition and configuration or executable dial plan components; in this case, they are called contexts. The configuration part is universal and global, and the system administrator ultimately defines the context name. The dial plan variables and their initial values are described in the extension. conf with the field begin with

[globals]

Asterisk global variables are usually used as constants rather than variables. They typically have only one place in your dial plan, where you can enter values. If you change the phone system configuration, you may want to change these values in the future.

Let us take an example for a global variable (from VoIP-info) [22]

[globals]

Here is an example that shows the global variable used in the dial plan.

; extension ring on incoming call

INCOMING => Extension-name/3& Extension-name/4

; ringing time before it is going to voicemail

RINGTIME=> number

; sound you need to plan on a ring

VMZNNOUNCE=> sound/my-vm-announce

; for outgoing call

OUTGING => Extension-name/1& Extension-name/2

These are a few examples you need to configure for four dial plans, and it is totally up to you to configure the desired value and modify it according to your need and make a dial plan the way you want to use it.

These are other configuration parts in Asterisk that you need to set up as per the guidelines for info; you can refer to VoIP-info.org, the second most crucial file that needs to configure for Asterisk to communicate with the outer world is sip. conf

Sip.conf

In sip.conf under [general] add a register definition:

Format:

register => user[:secret[:authuser]]@host[:port][/extension]

or

register => fromuser@fromdomain:secret@host

or

register => fromuser@fromdomain:secret:authuser@host:port/extension

- User is the user id for the SIP server ex 12345
- Althusser is the optional authorization user for the SIP server
- The secret is the user's password
- The host is the domain or hostname for the SIP server.
- Port sends the registration request to this port at host default for is 5060
- /1234 is the asterisk contract extension.

Sometimes the internet connection is not pretty reliable; it can go up and down. Because of this, you keep on losing your sip registry; you can add register attempts and register timeout setting to the general section. Setting register attempts=0 will force Asterisk to reregister until it can (the default is ten tries). register timeout sets the length of time in seconds between registration attempts (the default is 20 seconds). [22]

While doing the configuration, things come up in mind that where to start it; it has been a problem that everyone is facing while installing running scripting of anything. However, for the asterisk configuration, one can follow the asterisk homepage https://www.asterisk.org/ and form there can navigate the process. Also, I found another website that might help https://wiki.asterisk.org/wik, and this page is for the documentation of the asterisk file.

8.1 Primary role

The primary goal of implementing a successful honeypot Is to gather as much information as possible. To achieve this, data control, data capture, and data analysis need to be satisfied.[23].

Getting data is hard to implement since the attacker expects the system to perform as an actual host. If the Honeypot excessively restricts the actions and commands an attacker can take, they may be skeptical. However, too little control prevents an attacker from communicating with other systems and using techniques such as SSH tunneling to attack other targets. When implementing Honeypot in the network, consider the privileges and freedom the attacker has. Data capture is the second data requirement that needs to be addressed. Communications on certain services can be encrypted, for example, with SSH [23]. Because the attacker uses an encryption protocol, it becomes more challenging to obtain session information. In addition, the collected data should be recorded for future analysis. If these logs are stored in the smokescreen itself, the attacker can delete them from the system. External decoys are essential and contribute to data integrity.

Lastly, once the captured data is stored in a separate location, it needs to be analyzed [23]. Exploring honeypot data to identify new trends, threats and methods helps to understand the motivations and methods of attackers.

8.2 Risk of implementing Honeypot.

Although decoys on a host network can help protect the web, some risks are associated with lures. As mentioned above, an attacker who penetrates a spoofed system will be granted extended access to the system as if it were your system. Attack methods such as SSH tunneling or allowing the Honeypot to operate as part of a botnet are enabled and attributed to the Honeypot [23],[24]. In particular, honeypots bring risks to the internal network as they promote "an aggressive atmosphere" [24]

8.3 Detecting and fingerprinting

A Honeypot is most effective when it completely misleads users into believing that the whole system has been hacked; therefore, attackers often use fingerprints to determine whether they are in the Honeypot. Different versions of software usually handle inputs in different ways under other operating systems. These differences can help an attacker determine what software and design are used on a honeypot [24]. This can be done by testing various inputs, analyzing source code, and writing scripts to identify the software. By evoking a response that any available operating system does not have.

Once a honeypot is identified, attackers would do one of two things. First, they may not waste their time further on the Honeypot as they know it is not a natural system. Second, they may give the Honeypot invalid information to degrade the data the Honeypot is collecting [23]; studying such corrupted information might now no longer cause any sizable findings. In both cases, the Honeypot is no longer beneficial to that attacker and possibly others with whom the attacker communicates.

Many Honeypot implementations are open source, so a common source of deception needs to be provided so that legitimate users can test the decoy and collect data. Over time, attackers can thoroughly analyze the source code and find errors or vulnerabilities. The analysis is inevitable. The attacker can also dynamically explore the Honeypot and make educated guesses based on its behavior to determine the environment.

As interactions increase, honeypot fingerprint identification becomes more complex because low-interaction Honey generally has a higher chance of being detected. While looking at this, as some of the Honeypots are available in the market, their source code is not unique, and attackers have grown and overlooked all these codes.

9.1 My Honeypot script.

By reading the different articles and seen different approaches for Honeypot, I decided to make my script; I developed a honeypot script; Python, we all know, is a high-level, general-purpose programming language that is easy to use and implement. We used a different approach to analyze attacks in a SIP-based network with various features for attack analysis. According to collected data, most attacks came from the SIP (port 5060) protocol

During my next section, I will be continuing my work on Honeypot; in the first part, I collected the information on what type of attacker from user agents we are getting, then the number of attackers we all also try to get the contact number from where they are calling and to whom they are calling with the certainty of the country also. At last, I also made a web interface with the help of Python too to display my result.

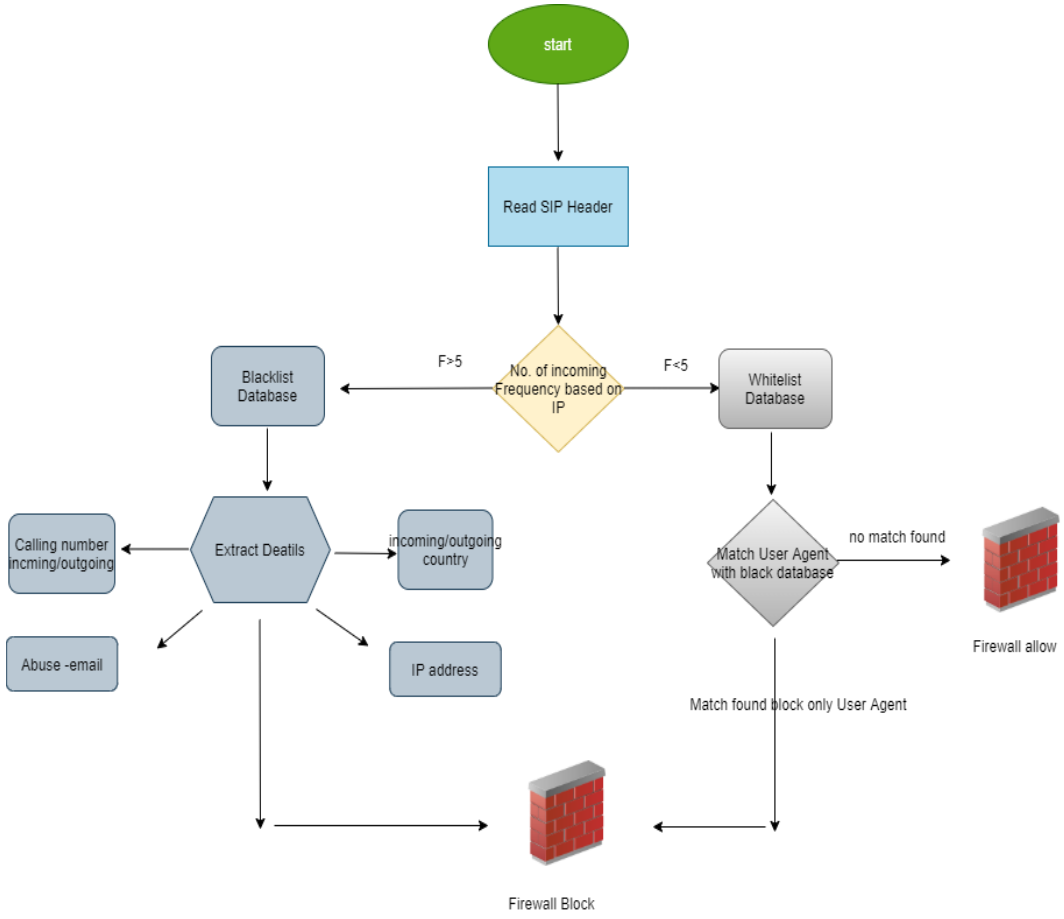


Figure 15. Figure flow chart for the proposed mechanism

The flow, as mentioned above chart is the proposed mechanism, how I started my work. In the first step, I counted the frequency of incoming Ip addresses associated with the user agent. If the frequency is less than 5, I counted them in the whitelist database and more than 5 in the blacklist database. For the database calculation, I used SQLite with pandas. The second step goes in two ways. First, if the frequency is less than 5, it will go to the white database, as we mentioned earlier. However, sometimes attackers can use the same user agent with a different IP address; in that case, I choose to compare the user agent name with the blacklisted items. If it is matched, we will block that only user agent, not the Ip, and this is because sometimes the Ip address can be for personal users, not the attackers, so it is better to block the User-agent instead of the real IP. In the second case, I decided to check based on incoming frequency; if the frequency is more than 5, it automatically goes to the blacklist database.

From the blacklist database, I managed to extract the details from blacklisted IP address example, calling number where the attackers are trying to reach and from where he is calling with the country name for which IPWhois helped me; also, we managed to get the abuse email and sent few emails also to let them know that attack is coming from their domain

Part 1: Analyzing the data

We explore the data in various ways; we check the attack in real-time figures [13]. Secondly, I downloaded the sip data in txt format for my script.

Figure 16. real-time data for SIP message

Here, in the above figure, is just an example of a real-time SIP message format; here, I ran the command which is *path location -I eth0 -s 1500 UDP port 5060 -v,* which is taking the ethernet port, and 1500 is for bringing message size under 1500 byte (which is a typical default value for the MTU size.) Since usual SIP proxies (like the load-balancer) only interfere with the content of SIP messages where it's necessary for the SIP routing, but otherwise leave the message intact as received from the endpoints. In contrast, the B2BUA creates a new call leg with a new SIP message from scratch towards the called party, SIP message sizes are reduced significantly by the B2BUA[8] , and MTU stands for Maximum Transmission Unit, which is typically a maximum size of the packet that can be transmitted from the network interface all the device including server and switches/routers involved in communication, should have the same MTU size and the default size in most of the Ethernet network is 1500 bytes. *UDP port 5060* uses the Datagram Protocol, a communication protocol for the Internet network, transport, and session layers. This protocol used over *port 5060* makes it possible to transmit a datagram message from one computer to another. Unlike *TCP port 5060, UDP port 5060* is connectionless and does not guarantee reliable communication; it is up to the application to receive the message on port 5060 to process any error and verify correct delivery.

From the approach, counting the frequency and outing them into a white/blacklist database, I conclude all the attacks on different methods, given in the tables below. I found that attackers are using different user-agents to attack our system and different approach. Below mention table is an example of user agents that were in attack on our system. In this table, we also found unique and other user agent names. That is funny because user agent names are generally device names, but the attackers change their names accordingly.

[8] A back-to-back user agent (**B2BUA**) is a logical network element in Session Initiation Protocol (SIP) applications. SIP is a signaling protocol for managing multimedia Voice over Internet Protocol (VoIP) telephone calls.

Method INVITE

User Agent	Number of Count
User-Agent: zdfvcvvhgjmn,jlj,vv	12471
User-Agent: Linksys-SPA942	8348
User-Agent: (Very nice Sip Registrar/Proxy Server)	2862
User-Agent: axzxccbvcnbxcbzxvssmnnsbscbs	2833
User-Agent: Linksys/SPA942	2250
User-Agent: vxcvbchfghffyyyyyyyyyyyyyyyy	1641
User-Agent: qwserrr@!#$%	1596
User-Agent: vxffhfghgjhrrttttt	1548
User-Agent: sipcli/v1.8	629
User-Agent: Cisco-SIPGateway	443
User-Agent: PBX	413
User-Agent: MGKsip release 1110	262
User-Agent: asaxzxccbvcnbxcbzxvssmnnsbscbsds	220
User-Agent: SDaxzxccbvcnbxcbzxvssmnnsbscbs	217
User-Agent: FPBX	173
User-Agent: Friendly Scanner	83
User-Agent: pplsip	67
User-Agent: ABTO SIP	50
User-Agent: a'or'3=3--	26
User-Agent: Cisco	11
User-Agent: Z 3.14.38765 rv2.8.3	11
User-Agent: FreePBX 1.8	6
User-Agent: TS-v4.5.1-20g	1
User-Agent: Trixbox	1

Table 1 Invite attack with User-Agent name and attacking frequency

e.g., User-Agent: SDaxzxccbvcnbxcbzxvssmnnsbscbs, which is a unique name with strange string characters.

If we again look into the table mentioned above, we realize that some user agents are botnets. According to the definition, Botnet is a network of computers devices that

work together; you can imagine Botnet as zombies, they have the physical device, but they are working all together on command . that term botnet is formed from "ro**bot**" and "**net**work." sometimes it refers as a tool to automate mass attacks, for example, server crashing, malware distribution.

Here in our case from table 1, User-Agent: Friendly Scanner is a type of Botnet; generally, it scan IP range for SIP servers example, PBX, which uses 5060 port as communication, and if they find port open, it attempts the brute force and way into the SIP server by checking anonymously with the combination of password and account.

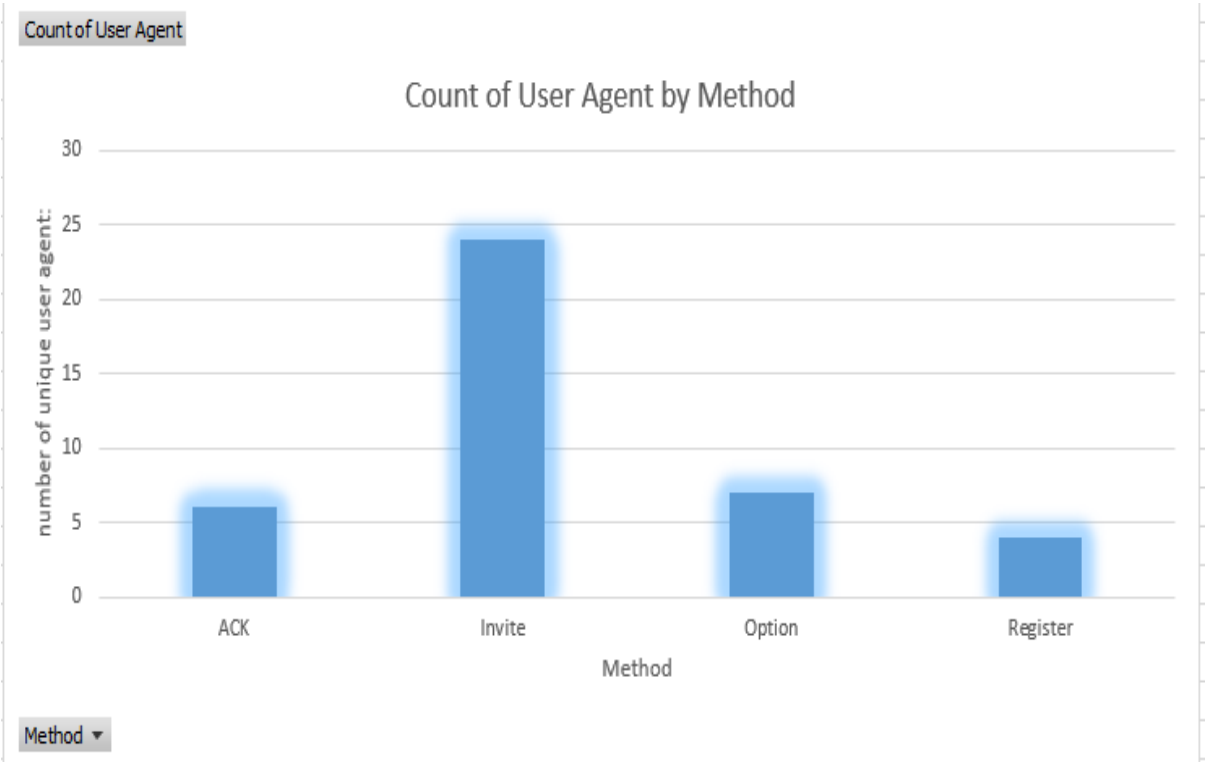


Figure 17. comparison of different methods in SIP message

The following figure shows the activity by different user agents in 3 days of the trial period. As we can see, most of them are interested in the *Invite* message.

Part 2 distribution of IP address as per frequency.

As we discussed previously that our approach is to list the given IP addresses according to their input/attack on our Honeypot; for this, we decided to map the IP address according to their input frequency; if the frequency is greater the 5, it directly goes to blacklist database, and if it is less then five it goes to white list, but here also we are checking the name of user agent by which IP address is associated. If it is the same as our database blacklist, we block this.

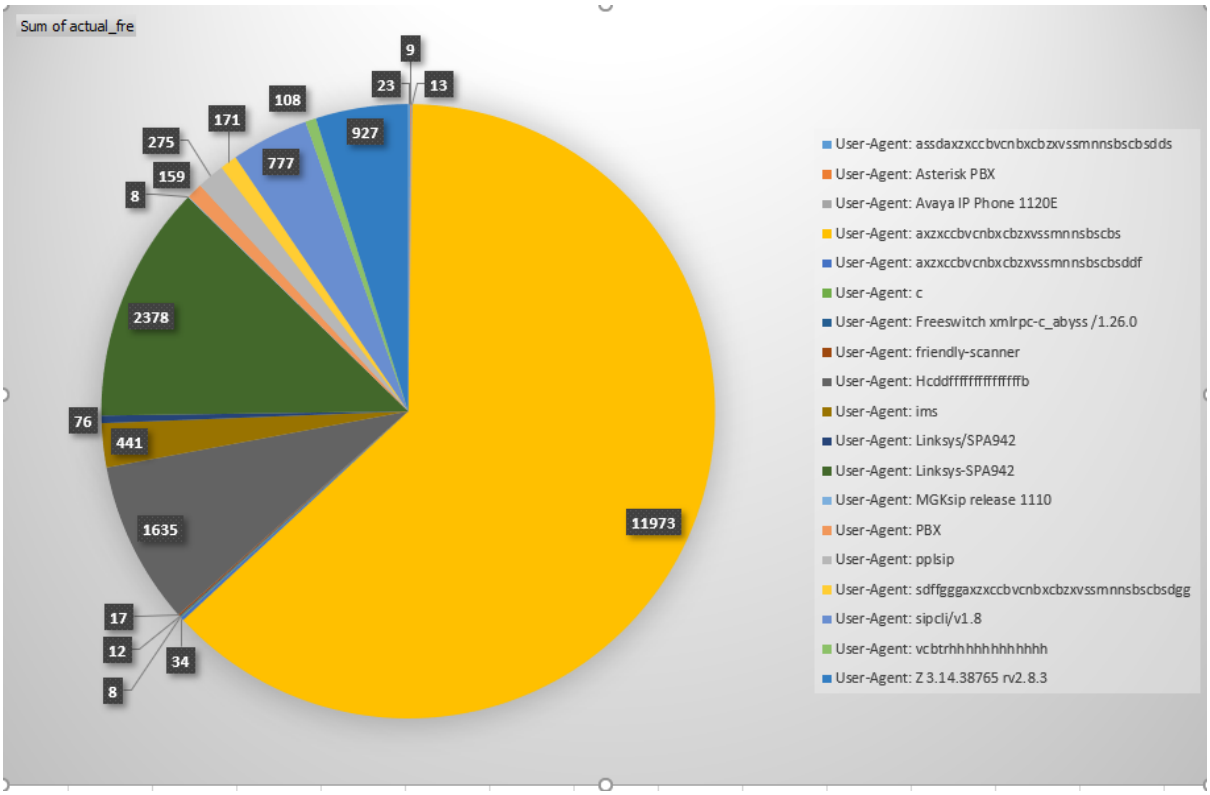


Figure 18. showing the attacking percentage of the user agent in the blacklist database

The figure mentioned above is the percentage count of blacklisted user agents and their input frequency; all listed user agents are more than 5 in counts. And they are blocked by our firewall agent.

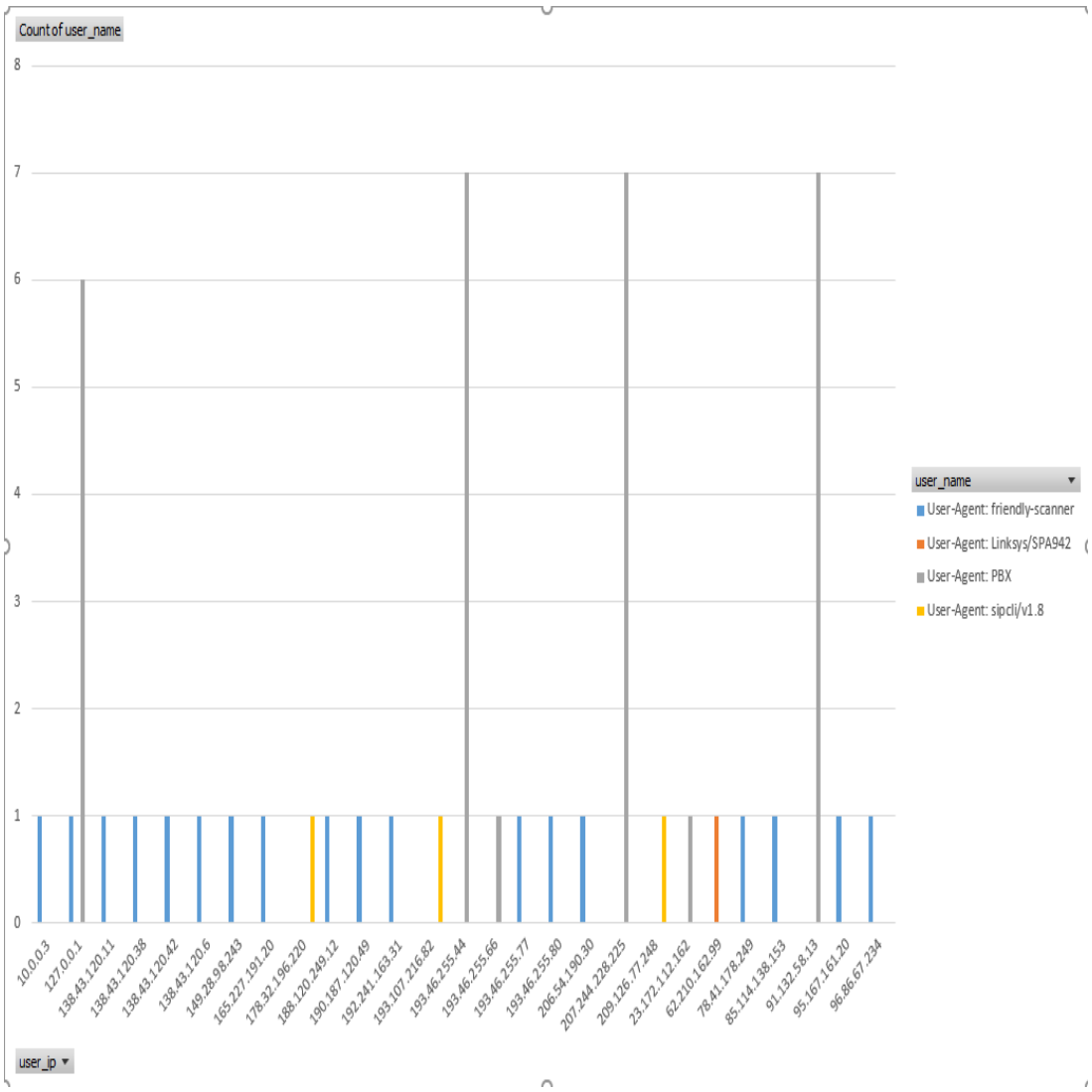


Figure 19. whitelist database with blacklist

The above figure (19) displays the whitelisted user agent which is lying in the blacklisted database. The above two figures can also analyze their attacking behavior; they frequently change their IP address to gain access.

Also, blocking an only IP address is not a solution, sinch with the help of this experiment, we can analyze that they are changing their IP address for gaining access, so from this, we also find a new conclusion to block the User-agent name along with the IP address, this can prevent the attacking, but prevention of 100 % attack is not possible. Since we agreed that blocking IP addresses is not a solution because

sometimes, we never know that an IP address can be from a not legitimate user; that user is also blocked. From this point of view, we only secure the few IP addresses which are very, very frequent, but our focus was to ensure the user agent; by this means, I created an intelligent script which is counting on the frequency and stopped the user agent also it is maintaining the history, suppose a blocked user agent is attacking your system, so it will not count that agent directly plugged.

Part 3 extracting the attacking details

Now we come to the point where we are trying to extract the data from capture IP addresses example, where the attackers are calling, which country they are trying to contact from, and which country they are targeting with their domain email address, the abuse email address we will send the mail to the email address that the attack is coming from your domain.

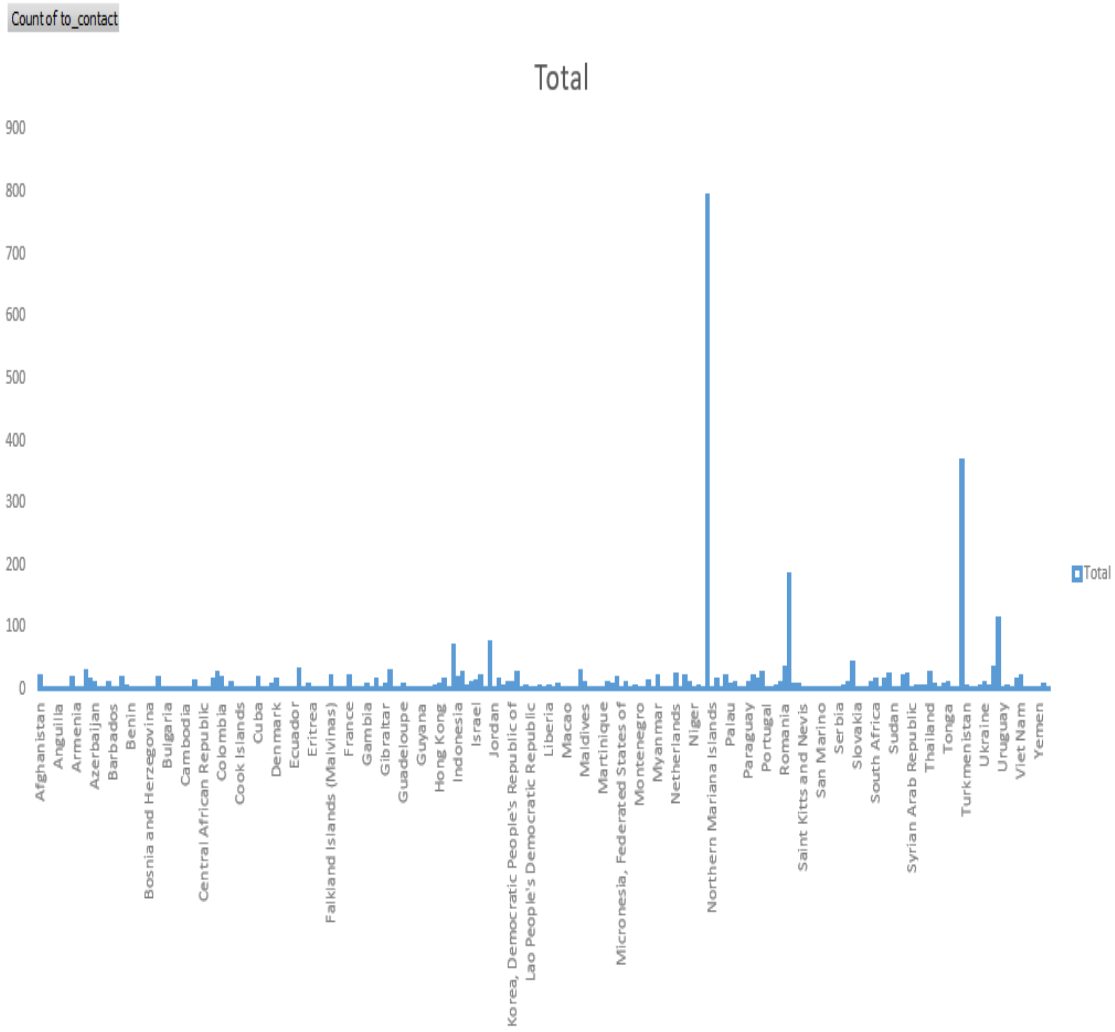


Figure 20. Contacted country from the attacker

In figure 20 and table 2, we extracted the numbers the attacker's target to contact via our Honeypot; according to data processed, the most active attackers they want to call are *Turkey, Russia, and the USA*. These statistics and results come from the various attacks created in our Honeypot; this program generates a CSV file names contact.csv.

This program also takes the country code from the python extension pycountry with phone_iso3166.country, which helps determine the country's name.

Country	Number of times called
Turkey	367
Russian Federation	184
United States	114
Japan	76
India	69
Singapore	43
Romania	35
United Kingdom	33
Egypt	32
Greece	30
Malaysia	29
Australia	29
Korea, Republic of	27
Iran, Islamic Republic of	27
Thailand	26
Poland	25
China	25
Netherlands	24

Table 2 number of times attackers try to call the number associated with the country

Country name	Count of country_incoming	Abuse email
France	2	['abuse@online.net']
Netherlands	22	['abuse@squitter.eu'],['report@peenq.nl']
United States	68	['noc@quickserver.co']

Table 3 top countries for attackers trying to get access to our Honeypot.

We also combined all Ip addresses, gathered the information based on the country, and found that the United state and Netherlands were responsible for most attacks coming to our Honeypot.

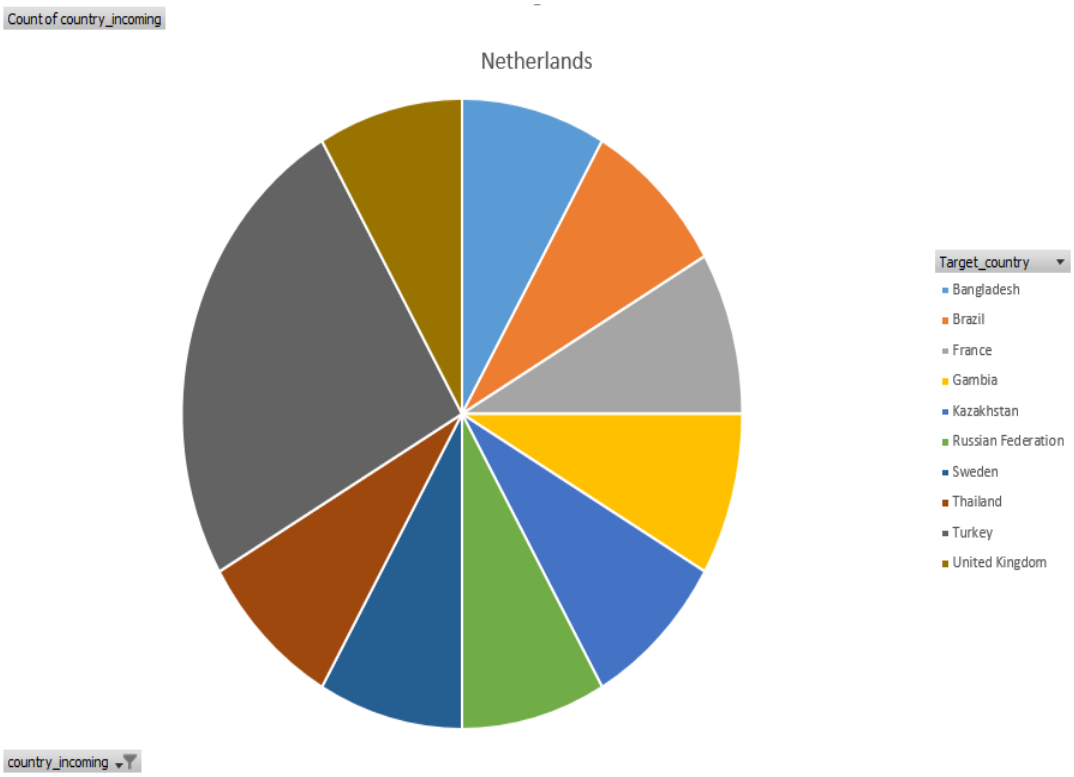


Figure 21. target countries by the targeted attackers for Netherland

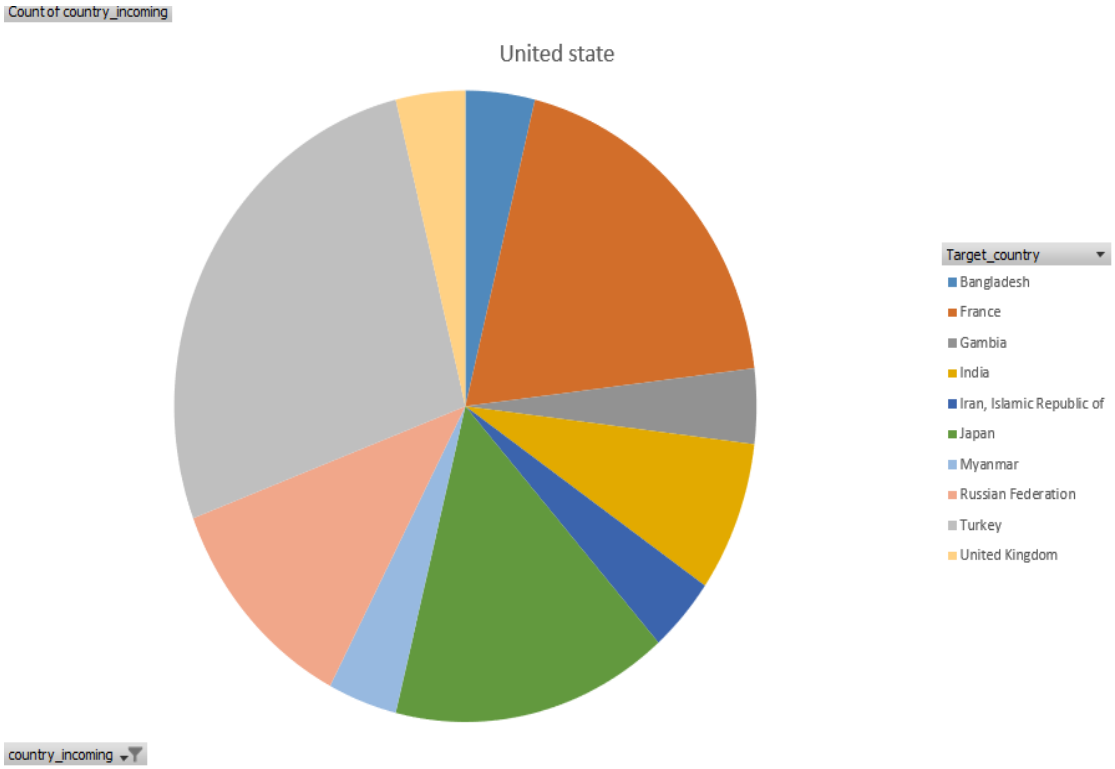


Figure 22. target countries by the targeted attackers for United state

ip	to_contact	user_agent	Target_country	country_incoming	email
37.49.230.13	9.00482E+13	User-Agent: Linksys-SPA942	Turkey	Netherlands	['abuse@squitter.eu']
193.46.255.52	9.00468E+13	User-Agent: Linksys-SPA942	Turkey	Netherlands	['report@peenq.nl']
193.46.255.52	7.00468E+13	User-Agent: Linksys-SPA942	Russian Federation	Netherlands	['report@peenq.nl']
193.46.255.52	46843737856	User-Agent: Linksys-SPA942	Sweden	Netherlands	['report@peenq.nl']
193.46.255.52	9.00047E+14	User-Agent: Linksys-SPA942	Turkey	Netherlands	['report@peenq.nl']
193.46.255.52	8.80047E+14	User-Agent: Linksys-SPA942	Bangladesh	Netherlands	['report@peenq.nl']
193.46.255.52	7.70047E+1	User-Agent:	Kazakhstan	Netherlands	['report@peenq.nl']

	4	Linksys-SPA942			
193.46.255.52	6.60047E+14	User-Agent: Linksys-SPA942	Thailand	Netherlands	['report@peenq.nl']
193.46.255.52	5.50047E+14	User-Agent: Linksys-SPA942	Brazil	Netherlands	['report@peenq.nl']
193.46.255.52	4.40047E+14	User-Agent: Linksys-SPA942	United Kingdom	Netherlands	['report@peenq.nl']
193.46.255.52	3.30047E+14	User-Agent: Linksys-SPA942	France	Netherlands	['report@peenq.nl']
193.46.255.52	2.20047E+14	User-Agent: Linksys-SPA942	Gambia	Netherlands	['report@peenq.nl']
23.148.145.228	9.00114E+16	User-Agent: Linksys-SPA942	Turkey	United States	['noc@quickserver.co']
23.148.145.228	9.00011E+17	User-Agent: Linksys-SPA942	Turkey	United States	['noc@quickserver.co']
23.148.145.228	9.527E+19	User-Agent: Linksys-SPA942	Myanmar	United States	['noc@quickserver.co']
23.148.145.228	2.20011E+17	User-Agent: Linksys-SPA942	Gambia	United States	['noc@quickserver.co']
23.148.145.228	3.30011E+17	User-Agent: Linksys-SPA942	France	United States	['noc@quickserver.co']
23.148.145.228	4.40011E+17	User-Agent: Linksys-SPA942	United Kingdom	United States	['noc@quickserver.co']
23.148.145.228	9.00184E+16	User-Agent: Linksys-SPA942	Turkey	United States	['noc@quickserver.co']
23.148.145.228	8.80019E+17	User-Agent: Linksys-SPA942	Bangladesh	United States	['noc@quickserver.co']
23.148.145.228	9.00014E+17	User-Agent: Linksys-SPA942	Turkey	United States	['noc@quickserver.co']

23.148.145.228	9.81044E+15	User-Agent: Linksys-SPA942	Iran, Islamic Republic of	United States	['noc@quickserver.co ']
23.148.145.228	8.10442E+14	User-Agent: Linksys-SPA942	Japan	United States	['noc@quickserver.co ']
23.148.145.228	7.98104E+16	User-Agent: Linksys-SPA942	Russian Federation	United States	['noc@quickserver.co ']
23.148.145.228	8.10442E+14	User-Agent: Linksys-SPA942	Japan	United States	['noc@quickserver.co ']
23.148.145.228	9.0081E+17	User-Agent: Linksys-SPA942	Turkey	United States	['noc@quickserver.co ']
23.148.145.228	9.10104E+16	User-Agent: Linksys-SPA942	India	United States	['noc@quickserver.co ']
23.148.145.228	8.10104E+16	User-Agent: Linksys-SPA942	Japan	United States	['noc@quickserver.co ']
23.148.145.228	7.1001E+17	User-Agent: Linksys-SPA942	Russian Federation	United States	['noc@quickserver.co ']
23.148.145.228	8.1001E+17	User-Agent: Linksys-SPA942	Japan	United States	['noc@quickserver.co ']
23.148.145.228	9.1001E+17	User-Agent: Linksys-SPA942	India	United States	['noc@quickserver.co ']
23.148.145.228	9.01442E+14	User-Agent: Linksys-SPA942	Turkey	United States	['noc@quickserver.co ']
23.148.145.228	9.01144E+15	User-Agent: Linksys-SPA942	Turkey	United States	['noc@quickserver.co ']
23.148.145.228	7.01144E+15	User-Agent: Linksys-SPA942	Russian Federation	United States	['noc@quickserver.co ']
23.148.145.228	3.37004E+16	User-Agent: Linksys-SPA942	France	United States	['noc@quickserver.co ']
23.148.145.228	3.35004E+16	User-Agent: Linksys-	France	United States	['noc@quickserver.co ']

		SPA942			
23.148.145.228	3.377E+17	User-Agent: Linksys-SPA942	France	United States	['noc@quickserver.co ']
23.148.145.228	3.355E+17	User-Agent: Linksys-SPA942	France	United States	['noc@quickserver.co ']

Table 4 Final Output

In the final part of our script, we make IP address as a reference and, based on that, aggregated the user agent, country, and abuse mail; since data was for more than 1,00,000 IP addresses, I extracted a long list and made it available for above table 4 result. During working, some of the IP details were not available Since they were private IPs, and some of the contact addresses were also unavailable.

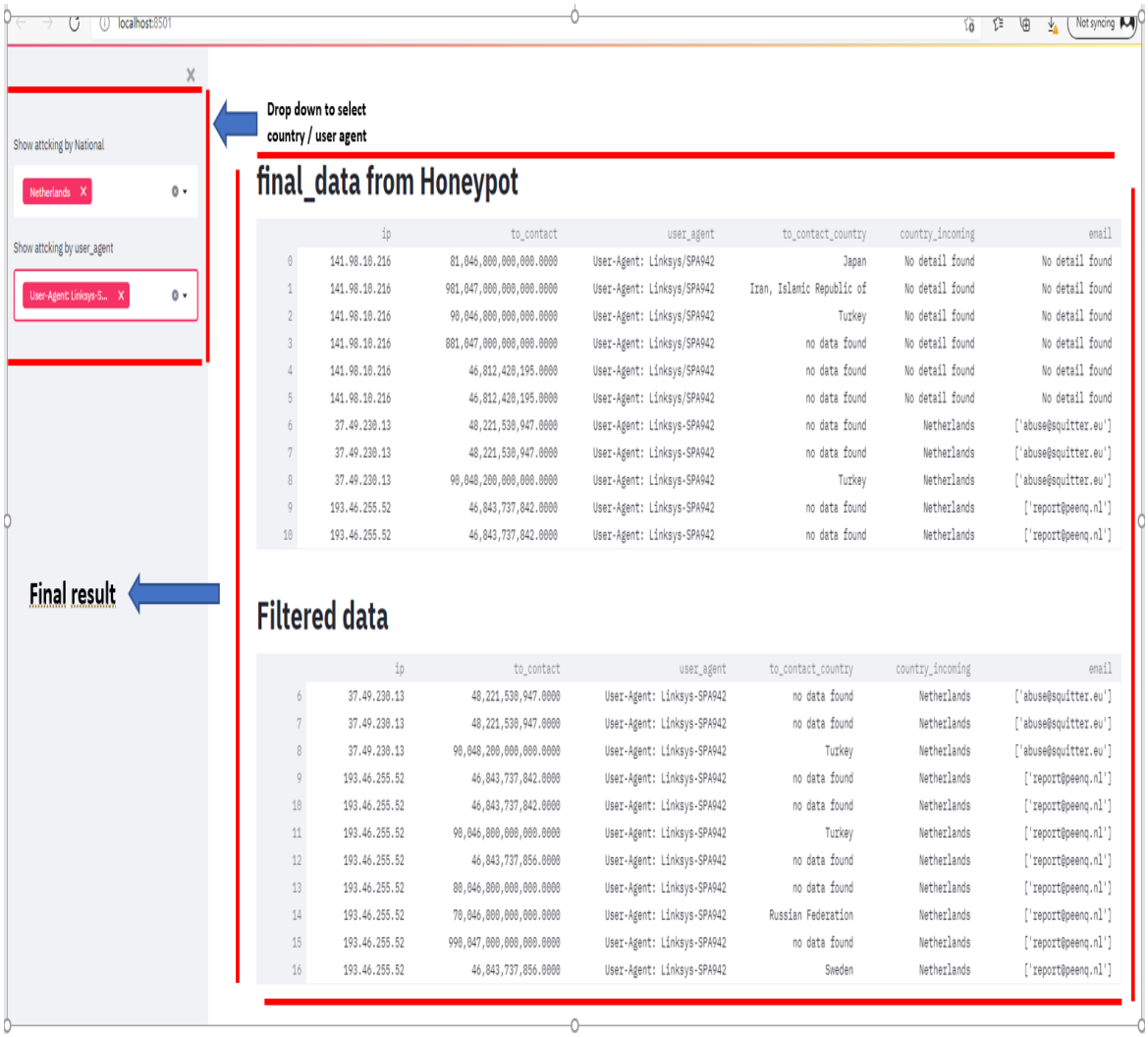


Figure 23 web interface format for table 4 data

In Figure 23 , displayed data is from table four in the form of web interface , it is created with the help of Python web interface that is streamlit and working on local host 8501 . from it you can clearly filter out the desired result with the dropdown option of “National” & “User Agent”.

10.1 Conclusion

This thesis project focused on integrating a honeypot with an existing system. Earlier, the honeypots are acting as a decoy system. The honeypots are unconventional tools. Their primary purpose is to collect data and subsequently analyze them to gain more information about attackers and their method, device they are using, type of attacks, and objective. This thesis paper mainly outlines the attacking types, their incoming frequency, and we put them on Firewall.

There are few things that we need to be sure of before analyzing the Honeypot. First is the data, because diagnosing of attacking pattern is based on data you are receiving, sometimes data can be false, and it misleads your whole project usage of secure communication channel is sufficient only for low-interaction and medium honeypots and for high-integration it is needed to use another method of data collection. Collectors residing on virtualized networks based on operating-system-level Virtualization are a solution to this type of decoy. The second important topic is to analyze the data from the Honeypot. Either it is your own made script or script from the web, but analyzing data is very important; while performing a task, you will get much information but extracting the valuable items in your study. In this thesis paper, we received a million of data but getting the right one was necessary. I pulled practical details like country name, contact number, and user-agent (device) for that track. We can also get the attackers' location, but I am not sure about its reliability. Because sometimes they are attacking using VPN and by this, their location is changed.

10.2 Future work

Prediction of future work is the most challenging combination because you never know what the attacking pattern could be, but we can make some assumptions and prepare our device. According to our research, the most valuable thing for future prospective could be an automatic firewall for user-agent, though further work can be possible. Part form improving this one can also attract more attackers to Honeypot using a "network funnel" that would route traffic destined to various unused parts of

the administered network to honeypots. We also suggest deploying other honeypots recommended in [25] to extend a portfolio of efficient traps

The effective display of a large amount of network data accelerates the development of detection methods and promotes the operability of the developed methods.

References

Hallock, J., 2004. A brief history of VoIP. *Evolution*.[1]

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E., 2002. RFC3261: SIP: session initiation protocol.[2]

Arkin, O., 2002. Why ET Can't Phone Home?: Security Risk Factors with IP Telephony based Networks. *Sys-Security Group*. [3]

Dhamankar, R., 2005. Intrusion Prevention: The Future of VoIP Security. *White paper. Tipping Point*. [4]

Egevang, K. and Francis, P., 1994. *The IP network address translator (NAT)* (pp. 1-10). RFC 1631, may. [5]

Geneiatakis, D., Kambourakis, G., Dagiuklas, T., Lambrinoudakis, C. and Gritzalis, S., 2005, September. A framework for detecting malformed messages in SIP networks. In *2005 14th IEEE Workshop on Local & Metropolitan Area Networks* (pp. 5-pp). IEEE. [6]

Luo, M., Peng, T. and Leckie, C., 2008, April. CPU-based DoS attacks against SIP servers. In *NOMS 2008-2008 IEEE Network Operations and Management Symposium* (pp. 41-48). IEEE. [7]

McGrew, R., 2006, January. Experiences with honeypot systems: Development, deployment, and analysis. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)* (Vol. 9, pp. 220a-220a). IEEE. [8]

Mukherjee, B., Heberlein, L.T. and Levitt, K.N., 1994. Network intrusion detection. *IEEE network*, *8*(3), pp.26-41. [9]

Ptacek, T.H. and Newsham, T.N., 1998. *Insertion, evasion, and denial of service: Eluding network intrusion detection*. Secure Networks inc Calgary Alberta. [10]

Wagner, D. and Soto, P., 2002, November. Mimicry attacks on host-based intrusion detection systems. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (pp. 255-264). [11]

Wichers, D., 2013. Owasp top-10 2013. *OWASP Foundation, February*. [12]

Spitzner, L., 2003. *Honeypots: tracking attackers* (Vol. 1). Reading: Addison-Wesley. [13]

Sadasivam, K., Samudrala, B. and Yang, T.A., 2005. Design of network security projects using honeypots. *Journal of Computing Sciences in Colleges*, *20*(4), pp.282-293. [14]

Schneier, B., 2015. *Secrets and lies: digital security in a networked world*. John Wiley & Sons.[15]

LM Security.,2016. Feb., How malware detects virtualized environment and its countermeasures [16]

McCaughey, R.J., 2017. *Deception using an ssh honeypot*. Naval Postgraduate School Monterey United States. [17]

Anonymous, What is KVM? Red Hat - We make open source technologies for the enterprise. Available at: https://www.redhat.com/en/topics/virtualization/what-is-KVM [Accessed May 12, 2021]. [18]

Kijewski, P. and Pawliński, P., 2014. Proactive detection and automated exchange of network security incidents. *Abgerufen am*, *20*. [19]

Davidson, J., Peters, J. and Bhatia, M., 2006. *Voice over IP fundamentals*. Cisco press. [20]

Gralla, P., 2006. Throw away your PBX: Why Asterisk may be the VoIP future of your network. Computerworld. Available at: https://www.computerworld.com/article/2547632/throw-away-your-pbx--why-asterisk-may-be-the-VoIP-future-of-your-network.html. [21]

Anonymous, 2020. Asterisk introduction - VoIP-Info. VoIP. Available at: https://www.VoIP-info.org/asterisk-introduction/ [Accessed 2021]. [22]

Anonymous, Introduction. DECEPTION USING AN SSH HONEYPOT. Available at: https://faculty.nps.edu/ncrowe/oldstudents/17Sep_McCaughey.htm [Accessed 2021]. [23]

Joshi, R.C. & Sardana, A., 2011. Honeypots: a new paradigm to information security, Science Publishers, CRC Press. (pp. 1-37) [24]

Anon, 2016. Proactive detection of security incidents II - Honeypots. ENISA. Available at: https://www.enisa.europa.eu/publications/proactive-detection-of-security-incidents-II-honeypots [Accessed May 11, 2021]. [25]

D. Butcher, X. Li and J. Guo, "Security Challenge and Defense in VoIP Infrastructures," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 6, pp. 1152-1162, Nov. 2007, doi: 10.1109/TSMCC.2007.905853. [26]