Volume 26 Issue 01, 2023

ISSN: 1005-3026

https://dbdxxb.cn/

ANALYSIS OF VOIP SECURITY THROUGH VAPT AND NETWORK PACKET ANALYSIS

Indrajeet Singh

School of Doctoral Studies and Research, National Forensic Sciences University, Sector 9, Gandhinagar-382007, Gujarat, India

*Corresponding author Dr. Naveen Kumar Chaudhary

School of Cyber Security and Digital Forensics, National Forensic Sciences University, Sector 9, Gandhinagar-382007, Gujarat, India

*Corresponding author Muhammad Zubair Dahir

School of Cyber Security and Digital Forensics, National Forensic Sciences University, Sector 9, Gandhinagar-382007, Gujarat, India

Biographical notes: Indrajeet Singh is a research scholar (Cyber Security) in the School of Doctoral Studies and Research, World's First of Its Kind, National Forensic Sciences University, Gandhinagar, Gujarat, Government of India. He received his MTech in Computer Science and Engineering from the Jagannath University (Haryana) and BE in Computer Engineering from the Guru Jambheshwar University of Science and Technology (Haryana).

Dr. Naveen Kumar Chaudhary is a Professor and the Dean at the National Forensic Sciences University, Gandhinagar, Gujarat, India. He is also the Head of the NFSU Cyber Defence Centre which is the first ISO 27001-certified CDC Lab in India. His research interest areas are network forensics, drone forensic, and next-generation networks.

Muhammad Zubair Dahir is pursuing MSc (Cyber Security) in the School of Cyber Security and Digital Forensics, the world's First of its kind, at the National Forensic Sciences University, Gandhinagar, Gujarat, Government of India.

Abstract

The rapid growth of technology such as computers or mobile phone applications that interacts with the internet poses a security risk. Vulnerability Assessment and Pen-Testing (VAPT) is an offensive method for safeguarding an organization with security assets. It is a big challenge for an organization to protect its data from growing vulnerabilities. In this research, various automated tools are used to identify the target system security breaches in the VoIP infrastructure posture as a part of vulnerability assessment. If the attacker identifies these security breaches, it can result in significant data loss. Pen-Testing is a method to determine whether the security measures are performing effectively. Voice Over Internet Protocol (VoIP) is one of the effective solutions employed by skilled criminals to conceal their identity from ordinary communication mechanisms. The method for VAPT of VoIP infrastructure is discussed and implemented in this research to combat similar types of attacks.

Keywords: Vulnerability Assessment and Penetration Testing (VAPT), VAPT tools, Security Assessment tools, Session Initiation Protocol (SIP), Real-Time Protocol (RTP)

1. Introduction

Voice over Internet Protocol is a cutting-edge innovation that has changed the mode of communication services, which are conveyed over IP networks. It offers better performance and discounted costs to its clients, making it more well-known than the past Public Switched Telephone Network. Despite its developing prominence, this innovation turned into an objective for various attacks [1]. The new improvement of voice over IP (VoIP) has caused phone communication to be a modern medium, to be specific the internet, making it more advantageous and less expensive for individuals to reach one another. These advantages have been generally received by the customary clients, yet additionally by cyber criminals, who can now decrease the expense and functional intricacy [2].

In comparison with different attacks for example email spam, voice spam calls are altogether more upsetting because they require immediate attention. When there is an incoming call, it is very hard to decide whether the incoming call is valid or spam by just looking at the caller ID, which is frequently "unknown" or even spoofed [3].

Two sorts of protocols are utilized in fundamental VoIP communication, (i) signaling protocol helps create, change, and make an end to the ongoing session, (ii) Media Transport Protocol, which transports the information after two entities lay out an association. Usually, SIP is utilized [4]. RTP (Real-time Transport Protocol) is indeed the standard signaling and media transport protocol [5].

Due to the lack of a specification for encrypting the original data in the RTP protocol, an attacker can easily grab the packets using different network sniffing tools and get the data of the user easily. Furthermore, experts have devised SRTP (Secure Real-Time Transport Protocol), which identifies and mitigates this problem by encrypting the payload field [6].

A couple of services that makes VoIP so well-known are, messaging, audio-video conferencing call, and voice mail. Besides, in light of the fact that these services are involving the internet as a mode of communication, they are more presented with security concerns, as VoIP technology has inherited Internet weakness. VoIP security is especially critical since phone conversations are sent in plaintext over the Internet, making it simple for an assailant to gain admittance to the correspondence channel because of the SIP's feeble confirmation [7].

According to security measures, SIP vulnerabilities are the primary factor of attacks against VoIP technologies [8,9]. Sip was created without any security concerns so it is disposed to different types of attacks like, snooping, message altering, and registration seizing. Users' trust in VoIP rather than PSTN is eroded as a result of such attacks. The third-generation partnership project (3GPP) officially adopted SIP as a signaling standard in 2000. The IETF announced SIP in RFC-3261 in 2002 [10].

Security hazards exist in all Internet-facing systems and apps. VAPT is the strategy utilized by security professionals all around the world to address these security issues [11].

Security is perhaps the main test data framework. Due to the rising interconnections of computers through the internet, expanded extensibility, and the unrestrained development of the size of the intricacy framework. The security of the system has become more of a challenge than in the past. Furthermore, it is an organization's responsibility to secure its cyber assets appropriately by utilizing a thorough and systematic method to protect against the threats that an organization may face [12].

Vulnerability analysis is the most common way of recognizing a subsection information space where a threatening client can take advantage of the coherent imperfections in a framework to make it defenseless [13].

Penetration testing, often known as ethical hacking or pen-testing, is a method for assessing risks in a network system before an attacker exploits them [14].

To resolve this issue, organizations teamed up to make Common Vulnerability Exposure (CVE) and Common Weakness Enumeration (CWE), two of which are supported by the MITRE Corporation [15]. Focusing on high-risk problems is one strategy to address these weaknesses. OWASP [16] and SANS [17] are two such communities.

The increase in network resource users leads to the creation of a significant amount of communicated data, as well as the challenge of dependable, secure data transmission, or information security. The utilization of encryption methods, an OS server, design, and settings of the Asterisks programming firewall, are the solution to this issue [18].

Since transmission dormancy with high communication quality is unsuitable, huge packets estimated altogether diminish network productivity. Small packets, then again, increment the portion above as they have a set length and the packet size diminishes [19]. The number of packets in which messages are divided increments quickly, bringing about a critical expansion in network traffic. As a result, the ideal length of transmitted packets is required to achieve maximum efficiency and better service quality for network users [20].

Since ideal parameters are subjected to various elements, a few of them which are continually changing the network capabilities, it is hard to decide them with high exactness. For instance, while sending packets over the network, the sum and kind of traffic, as well as the number of deferrals, but while fostering a network, designers utilize a cut-off inside which the packet length, or rather its information field, can be found, though the header is generally fixed length [21].

x- Lite is the softphone that is utilized for VoIP calls. It is a free softphone that provides a high-quality output to the user. Because of cost savings and ease of installation, many firms are switching from traditional communication media to VoIP systems [22]. It will be the most important component of next-generation networks. A codec (coder-decoder) is a device that turns an audio stream into a compressed digital form for transmission and vice versa for replay [23, 24].

The unwavering quality and security threats for confidentiality, integrity, and availability. It is seen that acquiring a higher level of reliability and security is as yet a significant challenge in the VoIP communication framework [25].

We undertake the Vulnerability Assessment and Penetration Testing activity of this VoIP technology as an infrastructure configuration in this research, as well as discuss some relevant definitions and problems. This research additionally depicts the weakness associated with VoIP that are utilized in VoIP security tools to find the shortcomings that make the threats and issues in the security of VoIP. We also go over how to install VoIP,

including an overview of all the protocols needed to use the Internet Telephony idea. Finally, we will discuss a vulnerability that might compromise the entire VoIP system.

2. Methodology

The conventional traditional method of the VoIP infrastructures Vulnerability Assessment and Pen-Testing (VAPT) activity is used in this research. The VAPT lifecycle is depicted in the diagram below:



The life cycle is as follows:

Fig: 1 Life Series of Vulnerability Assessment and Penetration Testing

We are going to initiate the activity by the respective flow chart mentioned below.

Journal of Northeastern University Volume 26 Issue 01, 2023



Fig: 2 VAPT Flow Chart for VoIP server (Asterisks)

Elaborating it further,

Our research was conducted in the following steps by building our lab setup and environment to conduct and analyze the VOIP Server and testing scenarios.

These steps to be performed and executed are as follows

Step 1: Creating Parent Virtual Machine Instances (Setup) PBX Server in the Ubuntu Machine



Copyright © 2023. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/



Step 2: Host and Child Virtual Instances for connecting VOIP Calls, it can be a flexible OS, we are taking windows in our cases. Host Machine 1Virtual Machine 1



Fig: 4 Host and Child Virtual Machine Instance for Connecting the VoIP Call

Step 3: Configuration of Asterisk server.



Fig: 5 Installation and configuration of Asterisk Server

Step 4: After successful configuration and server integration. The call is initiated from the Host machine using X-Lite to the VMware Windows machine using MicroSIP softphones.



Fig: 6 Call initiation from the Child Virtual Machine

3. Vulnerability Assessment of VoIP Infrastructure (PBX Server, Asterisks)

Know we need to focus on our traditional VAPT testing activity, as discussed above. Scope – Info Gathering – VA Activity – Pen Testing – Result Analysis – Report

Step 1: Operating System-Level Vulnerability Assessment, (UBUNTU)

Starting with the Scope and information gathering,

Scope of IP - Machine on which asterisk server installed and Nmap test result - Outcome

• Port state service: 2000/tcp open cisco-sccp: This vulnerability can be used to exploit by sending specially crafted packets to the SCCP (2000/tcp) port, a DoS attack can be completed by a far-off aggressor.

• TLS 1.2 not vulnerable to heartbleed: In the open SSL cryptographic software library heartbleed is a critical vulnerability. This allows the compromising of the private keys used to identify service providers and encrypt data, as well as the identity and password of the users. An attacker can leverage this flaw to listen in on a conversation, which is known as an Eavesdrop attack.

```
ubuntu@ubuntu:~$ nmap 192.168.110.141

Starting Nmap 7.60 ( https://nmap.org ) at 2022-04-09 05:08 PDT

Nmap scan report for ubuntu (192.168.110.141)

Host is up (0.00031s latency).

Not shown: 999 closed ports

PORT STATE SERVICE

2000/tcp open cisco-sccp

Nmap done: 1 IP address (1 host up) scanned in 2.14 seconds
```

| Testing SSL server 192.168.110.141 on port 2000 | |
|--|--|
| TLS renegotiation: Session renegotiation not supported | |
| TLS Compression: OpenSSL version does not support compression | |
| Rebuild with zllbig-dev package for zlib support | |
| Heartbleed: | |
| TLS 1.2 not vulnerable to heartbleed | |
| TLS 1.1 not vulnerable to heartbleed | |
| TLS 1.0 not vulnerable to heartbleed | |

Fig: 7 Nmap test (Information Gathering)

Step 2: Application-Level Vulnerability Assessment, Nessus activity that has been performed on the Respective OS (UBUNTU). The vulnerable application (Firefox) which is installed in the operating system is shown in fig 8 below. It indicates the critical vulnerabilities of the application which can cause the failure of the VoIP infrastructure.

• Ubuntu 18.04 LTS/20.04 LTS/21.10: Vulnerabilities of Firefox (USN-5314-1), (USN-5284-1), (USN-5131-1): This vulnerability was discovered when the XSLT parameter was removed in some circumstances. The users were tricked to open a specially crafted website. This blemish can be utilized to do a DoS attack or execute arbitrary code.

| CRITICAL | 9.8 | 158646 | Ubuntu 18.04 LTS / 20.04 LTS / 21.10 : Firefox vulnerabilities (USN-5314-1) |
|----------|------|--------|---|
| CRITICAL | 9.8 | 158502 | Ubuntu 18.04 LTS / 20.04 LTS / 21.10 : GNU C Library vulnerabilities (USN-5310-1) |
| CRITICAL | 9.8 | 158053 | Ubuntu 18.04 LTS / 20.04 LTS : Firefox vulnerabilities (USN-5284-1) |
| CRITICAL | 9.1 | 155687 | Ubuntu 18.04 LTS / 20.04 LTS / 21.04 / 21.10 : BlueZ vulnerabilities (USN-5155-1) |
| CRITICAL | 9.0 | 153583 | Apache < 2.4.49 Multiple Vulnerabilities |
| CRITICAL | 10.0 | 154883 | Ubuntu 18.04 LTS / 20.04 LTS / 21.04 / 21.10 : Firefox vulnerabilities (USN-5131-1) |
| HIGH | 8.8 | 157457 | Ubuntu 16.04 LTS / 18.04 LTS / 20.04 LTS / 21.10 : BlueZ vulnerability (USN-5275-1) |
| нісн | 8.8 | 155970 | Ubuntu 18.04 LTS / 20.04 LTS / 21.04 / 21.10 : Firefox vulnerabilities (USN-5186-1) |
| HIGH | 8.8 | 153925 | Ubuntu 18.04 LTS / 20.04 LTS / 21.04 : Firefox vulnerabilities (USN-5107-1) |
| HIGH | 8.8 | 158817 | Ubuntu 18.04 LTS / 20.04 LTS / 21.10 : Firefox vulnerabilities (USN-5321-1) |
| HIGH | 8.8 | 159022 | Ubuntu 18.04 LTS / 20.04 LTS / 21.10 : Firefox vulnerabilities (USN-5321-2) |

Fig: 8 Application-Level Vulnerability Assessment

Step 3: Communication Level Vulnerability Assessment, Nessus activity that has been performed on the Respective OS (UBUNTU)

• OpenSSL 1.0<1.0.1u multiple vulnerabilities (SWEET32): The SWEET32 vulnerability allows attackers to exploit the block cipher collisions. The attacker can use a 64-bit block cipher to compromise HTTPS connections.

• OpenSSL 1.0.1<1.0.1s several vulnerabilities: When parsing the malformed DSA private keys, this vulnerability occurs. This exploit can be utilized by a distant assailant to ruin memory, bringing about a DoS attack.

| CRITICAL | 9.8 | 89081 | OpenSSL 1.0.1 < 1.0.1s Multiple Vulnerabilities (DROWN) |
|----------|------|--------|---|
| CRITICAL | 9.8 | 93814 | OpenSSL 1.0.1 < 1.0.1 u Multiple Vulnerabilities (SWEET32) |
| CRITICAL | 9.8 | 85299 | PHP 5.5.x < 5.5.28 Multiple Vulnerabilities |
| CRITICAL | 9.8 | 88693 | PHP 5.5.x < 5.5.32 Multiple Vulnerabilities |
| CRITICAL | 9.8 | 90007 | PHP 5.5.x < 5.5.33 Multiple Vulnerabilities |
| GRITICAL | 9.8 | 90360 | PHP 5.5.x < 5.5.34 Multiple Vulnerabilities |
| GRITICAL | 9.8 | 90920 | PHP 5.5.x < 5.5.35 Multiple Vulnerabilities |
| CRITICAL | 9.8 | 91897 | PHP 5.5.x < 5.5.37 Multiple Vulnerabilities |
| CRITICAL | 9.8 | 92554 | PHP 5.5.x < 5.5.38 Multiple Vulnerabilities (httpoxy) |
| CRITICAL | 9.1 | 101788 | Apache 2.4.x < 2.4.27 Multiple Vulnerabilities |
| CRITICAL | 9.1 | 128033 | Apache 2.4.x < 2.4.41 Multiple Vulnerabilities |
| CRITICAL | 9.1 | 88679 | PHP prior to 5.5.x < 5.5.31 / 5.6.x < 5.6.17 Multiple Vulnerabilities |
| CRITICAL | 9.0 | 153583 | Apache < 2.4.49 Multiple Vulnerabilities |
| CRITICAL | 10.0 | 78555 | OpenSSL Unsupported |

Fig: 9 Communication Level Vulnerability Assessment

Step 4: Network Level Vulnerability Assessment, Devices (Router)

• IP Forwarding Enabled: This exploit allows the attacker to route the packet through the host and potentially bypass various firewalls and routers.

• DNS Server Cache Snooping: This vulnerability reveals the information about the owner of the DNS server such as vendor details. This method could be used to gather statistical information such as the time when the owner typically accesses the confidential data. The cached DNS records the remaining Time-to-live (TTL) values which provide accurate data.

| Severity | CVSS v3.0 | Plugin | Name |
|----------|-----------|--------|---|
| MEDIUM | 6.5 | 50686 | IP Forwarding Enabled |
| MEDIUM | 5.3 | 12217 | DNS Server Cache Snooping Remote Information Disclosure |

Fig:10 Network Level Vulnerability Assessment

4. Penetration Testing of the VoIP Infrastructure (PBX Server, Asterisks) Step 1: Identification of Known Exploit found in Vulnerability Assessment are as follows: In this paper, according to the vulnerability assessment, we shall perform two attacks on the VoIP system are as follows:

- Eavesdropping
- Denial of Service

Step 1: Manual Fuzz/ Attacking the system/ Packet Analysis

Eavesdropping VoIP calls with Wireshark

Eavesdropping, often known as snooping or listening, occurs when a computer, smartphone, or other connected device intercepts data being transmitted across a network. The attacker exploits an insecure network communication to obtain access to data transmitted or received

by the user.

| | | | | | _ | | × |
|--|--|--|-------------------|--|----|------------|------|
| Cristing propp | napog | No. Marken Marken | | | | | ^ |
| hie Edit View Go | Capture Analyze Stat | stics lelephony Wireles | i icols Pi | eb. | | | |
| 🗶 🗏 🖉 🔜 🚍 🚍 | X 🖸 🤇 + + 😫 | 11 🖬 🚍 📾 🍕 🍕 | e, 👥 | | | | |
| udp | | | | | 83 | — • | + |
| io. Time | Source | Cestination | Protocol | Length 3xfs | | _ | |
| 1 0.000000 | 192.168.48.128 | 192.168.48.2 | DNS | 76 Standard query 0xced4 A dns.msftncs1.com | | | |
| 2 1.918890 | 192.168.48.129 | 192.168.48.2 | DNS | 87 Standard query ExceSe PTR 129.48.168.192.in-addr.arpa | | | |
| 3 1.930544 | 192.168.48.2 | 192.168.48.129 | UDP | 42 53 + 50838 Len=0 | | | |
| 4 2.722455 | 192.168.48.1 | 192.168.48.129 | SIP/SOP | 1459 Request: INVITE sip:70028192.168.48.1 | | | |
| 5 2.722987 | 192.168.48.129 | 192.168.48.1 | SIP | 610 Status: 401 Unauthorized | | - | _ |
| 6 2.724499 | 192.168.48.1 | 192.168.48.129 | SIP | 362 Request: ACK sip:70028192.168.48.1 | | | |
| 8 2.843858 | 192.168.48.1 | 192.168.48.129 | SIP/SOP | 138 Request: INVITE sip:70028192.168.48.1 | | | |
| 9 2.845104 | 192.168.48.129 | 192.168.48.1 | SIP | 554 Status: 100 Trying | | | |
| 10 2.846422 | 192.168.48.129 | 192.168.48.1 | SIP/SOP | 861 Status: 200 OK (INVITE) | | | |
| 11 2.849327 | 192.168.48.1 | 192.168.48.129 | RTCP | 174 Receiver Report Source description | | | |
| | | | | | | | |
| 0000 00 0c 29 53 c 0010 05 a5 1b 33 0 0020 30 81 ac 8e 1 | c8 61 00 50 56 c0 00 10 00 80 11 38 42 c0 13 c4 05 91 ad ee 49 | 01 00 00 45 00 ···)5 a8 30 01 c0 a8 ···3 4e 56 49 54 45 0··· | a P V 88 0 | (· | | | |
| 1030 20 73 69 70 3 | 3a 37 30 30 32 40 31 | 39 32 2e 31 36 sip | 700 28192. | 16 | | | |
| 1010 38 2e 34 38 2 | 2e 31 20 53 49 50 2f | 32 2e 30 0d 0a 8.48 | 1 5 1P/2.4 | | | | |
| 1050 20 31 39 32 2 | 20 33 49 30 21 32 28 | 2e 31 3a 34 34 192 | 168 .48.1 | 44 | | | |
| 0070 31 37 34 35 6 | 62 72 61 6e 63 68 3d | 7a 39 68 47 34 174; | bran ch+298 | 64 | | | |
| 1000 62 4b 2d 64 3 | 38 37 35 34 33 2d 36 | 61 34 31 63 36 bK-d | 1754 3-6843 | let . | | | |
| 090 35 30 34 39 3 | 33 35 35 33 36 62 2d | 31 2d 2d 64 38 5049 | 3553 6b-1 | 48 | | | |
| 00+0 37 35 34 33 2 | 23 35 72 78 67 72 74 | ed ea 4d 61 78 7543 | -jep ort-1 | ax . | | | |
| 10c0 6f 6e 74 61 6 | 13 74 3a 20 3c 73 69 | 70 3a 37 30 30 onta | tt (sipi) | NO | | | |
| O Z User Datagram P | Protocol: Protocol | | | Packets: 4725 - Daplayed: 4674 (98,9%) - Dropped: 0 (0,0%) | 1 | Profile: | Defa |
| and the standy and | | | _ | II | | | - |

Fig: 11 Host machine sending an invite request

The host machine (192.168.48.1) sent an invite request to the asterisk server(192.168.48.129)

| Lavesdropping.pcap. | pcapng | tion Tolenham Minders | Tests Male | - B |
|---|--|--|---|-----|
| Eat view co | Tapture Analyze Statis | T I P P A A | icos mep | |
| ■ A2 ♥ B 8 | NN YFFE | 1 T D E d d | 1 II | (M) |
| dp | | | | |
| Time | Source | Destination | Protocol Length Info | |
| 14 2.867914 | 192.168.48.1 | 192.168.48.129 | RTP 214 PT+ITU-T 6.711 POW, SSRC+0x8CAAE9E2, Seq=7391, Time=2492500, Mark | |
| 15 2.868467 | 192.168.48.128 | 192.168.48.129 | SIP 534 Status: 180 Ringing | |
| 16 2.887779 | 192.168.48.1 | 192.168.48.129 | RTP 214 PT+ITU-T 6.711 POW, SSRC+0x8CAAE9E2, Seq=7392, Time=2492660 | |
| 17 2.907753 | 192.168.48.1 | 192.168.48.129 | RTP 214 PT+ITU-T 6.711 PCHU, SSRC+0x8CAAE9E2, Seq+7393, Time+2492820 | |
| 18 2.927694 | 192.168.48.1 | 192.168.48.129 | RTP 214 PT=ITU-T 0.711 PCHU, SSRC=0x8CAAE9E2, Seq=7394, Time=2492988 | |
| 19 2.947649 | 192.168.48.1 | 192.168.48.129 | RTP 214 PT+ITU-T 6.711 PCHU, SSRC-0x8CAAE962, Seq=7395, Time=2493140 | |
| 20 2.967582 | 192.168.48.1 | 192.168.48.129 | RTP 214 PT+ITU-T 6.711 POW, SSRC+0x8CAAE9E2, Seq=7396, Time=2493300 | |
| 21 2.988520 | 192.168.48.1 | 192.168.48.129 | RTP 214 PT=ITU-T 0.711 PCHU, SSRC=0x8CAAE9E2, Seq=7397, Time=2493468 | |
| 22 3.008454 | 192.168.48.1 | 192.168.48.129 | RTP 214 PT-ITU-T 6.711 PCHU, SSRC-0x8CAAE9E2, Seq=7398, Time=2493628 | |
| 23 3.028411 | 192.168.48.1 | 192.168.48.129 | RTP 214 PT-ITU-T 6.711 PC/U, SSRC-0x8CAAE9E2, Seq=7399, Time=2493788 | |
| rame 15: 534 by | tes on wire (4272 bits) |), 534 bytes captured | 4272 bits) on interface \Device\NPF (11CAEFD4-8073-4150-9844-2088C2E46DE9), id 0 | |
| User Datagram Pr Session Initiati | rotocol, Src Port: 5818: ion Protocol (180) | 1, Dst Port: 5060 | 6.45.129 | |
| iser Datagram Pr ession Initiati | rotocol, Src Port: 5818 ion Protocol (180) | 1, Ost Port: 5060 | 6.46.129 | |
| User Datagram Pr Session Initiati 0 00 0c 29 53 0 | rotocol, Src Port: 5818 ion Protocol (180) c8 61 00 0c 29 96 e8 0 | 10.40.120, 0511 1221 1, Ost Port: 5060 | | |
| User Datagram Pr Session Initiati 00 00 0c 29 53 10 02 05 85 ef 0 | rotocol, Src Port: 5418 [on Protocol (180) c8 61 00 0c 29 96 e8 5 80 00 80 11 60 a3 c0 a | 10.40.120, Dit: 12/1 1, Dit Port: 5060 19 08 00 45 00 ···)5 | €. €. | |
| User Datagram Pr Session Initiati 00 00 0c 29 53 10 02 08 85 ef 20 30 81 e3 45 | retocol, Src Port: 5418 ion Protocol (180) c8 61 00 0c 29 96 e8 0 00 00 80 11 40 a3 c0 a 10 c4 01 44 63 77 534 | 10.40.120, 051 12/1 1, 05t Port: 5060 10 05 06 45 00 | 4.129 | |
| User Datagram Pr Session Initiati 0 00 0c 29 53 0 0 02 08 85 ef 0 30 81 8 4 5 0 30 81 8 4 5 0 30 81 8 4 5 | rotocol, Src Port: 5418 ion Protocol (180) c6 61 00 0c 29 96 e8 0 40 00 80 11 40 a3 c0 a 13 c4 01 f4 63 77 53 c 13 c4 62 64 63 77 53 c | 240-40-120, UNT 192-1 1, UNT Port: 5000 09 00 00 45 00 | | |
| User Datagram Pr Session Initiati 0 00 0c 29 53 0 02 05 55 ef 0 30 81 e3 45 0 30 20 31 38 0 69 61 3a 20 1 1 30 30 20 | retocal, Src Port: 5418 ion Protocal (180) c6 51 60 6c 25 96 e6 1 50 66 88 81 46 69 83 56 8 50 26 82 81 46 67 75 35 30 26 92 66 56 67 76 9 30 36 95 97 32 26 96 66 79 | 10-0-110, 011 13-1 1, Dit Port: 5000 10-00 00 45 00 35- 8 30 00 45 00 35- 8 30 00 45 00 35- 16 47 04 8 50 15- 16 47 04 8 50 15- 16 47 51 56 20 15- 16 55 56 20 - | 6.46.129 €. €.<td></td> | |
| User Datagram Pr Session Initiati 00 00 0c 29 53 10 02 00 85 ef 10 30 81 e3 45 10 30 20 31 34 10 69 61 3a 20 10 31 39 32 2e 10 30 32 32 | c6 61 00 0c 29 96 e8 0 00 00 00 11 00 0c 29 96 e8 0 00 00 80 11 00 a3 c0 a 13 c4 01 f4 63 77 53 4 30 20 52 66 6e 67 69 6 53 49 50 47 52 2e 10 2 31 36 38 2e 34 38 2e 3 27 26 6f 22 76 46 32 | 200-00.120, 00t1 192.1 1, Dat Port: 5000 00 00 00 45 00 | 4.41.129 | |
| User Datagram Pr Session Initiati 00 00 0c 29 53 10 02 08 85 ef 0 10 30 20 14 53 10 06 01 3a 20 13 13 9 32 20 10 31 39 32 20 10 35 33 53 50 15 56 35 55 | ctocol, Src Port: 5018 ion Protocol (180) c6 61 00 0c 22 96 e8 8 00 00 80 11 40 a3 c6 0 13 c6 01 46 43 77 53 4 30 20 52 65 46 e7 60 6 53 49 50 27 32 20 10 13 16 38 20 43 88 20 4 27 20 6f 72 74 34 35 3 76 65 64 43 31 39 32 5 | 10 - 0 - 1 - 0 - 1 - 1 - 1 - 1 - 1 - 1 - | 6.46.129 €. | |
| User Datagram Pr Session Initiati 0 00 0c 29 53 10 02 06 85 ef 3 28 12 45 10 29 21 33 10 12 29 13 38 10 29 13 39 13 28 22 28 10 39 28 28 13 39 22 28 10 39 26 35 28 10 39 26 36 28 31 10 43 48 28 31 | c6 61 00 0c 29 96 e8 0 00 00 00 11 00 0c 29 96 e8 0 00 00 80 11 00 83 00 30 40 14 63 77 534 30 20 52 69 6e 67 69 6 31 36 32 2e 49 6e 67 69 6 31 36 38 2e 34 38 2e 3 76 65 64 36 31 39 52 2 29 39 30 62 72 61 6e 6 | 200-00.120, 0011 192-1 1, Dit Port: 5000 09 00 00 45 00 | a. 46, 129 b | |
| Dier Datagram Pr Session Initiati 0 00 0c 29 53 0 20 85 5 ef 0 30 20 51 36 0 39 20 51 36 0 39 20 51 36 0 31 30 52 20 0 31 30 52 20 0 34 35 20 0 34 36 25 36 0 34 30 22 31 0 56 51 55 69 0 34 36 25 31 0 56 41 35 24 31 0 56 41 35 25 31 0 56 41 35 25 31 0 56 41 35 25 31 0 56 41 35 31 0 56 41 35 31 0 56 41 35 31 0 56 41 35 31 0 56 56 56 56 31 0 56 56 56 56 56 56 0 56 56 56 56 56 56 0 56 56 56 56 56 56 0 56 56 56 56 56 56 56 56 56 0 56 56 56 56 56 56 56 56 56 56 56 56 56 | ctocol, Src Port: 5428 ion Protocol (180) c6 51 00 0c 29 96 e8 0 40 00 80 11 40 a3 c6 a 13 c4 01 f4 63 77 53 4 30 20 52 69 6e 67 69 6 53 49 50 2f 32 2e 30 2 31 53 8 24 38 26 34 38 26 3 72 70 4f 72 74 34 35 3 72 6 55 64 31 39 52 32 32 93 56 26 6 49 51 30 31 29 52 16 56 6 | 100-00-120, OHT 192-1 1, Ost Port: 5000 100 00 00 45 00>5- 100 00 00 45 00>5- 100 00 00 00 00 00 100 00 00 00 00 100 00 00 00 100 00 00 100 00 00 100 00 00 100 00 10 | € | |
| User Datagram Pr Session Initiati 00 00 0c 29 53 30 81 20 85 61 30 81 20 45 30 81 20 45 30 81 20 45 30 81 20 45 30 30 20 31 30 69 61 3a 20 30 36 36 26 31 39 52 26 53 55 60 30 36 36 26 31 30 52 31 30 43 32 21 30 43 42 31 00 44 7 34 62 31 66 56 56 56 31 66 56 35 31 66 56 56 56 31 66 56 56 56 31 66 56 56 56 56 31 66 56 56 56 56 56 31 66 56 56 56 56 56 56 56 56 56 56 56 56 | c6 61 00 0c 29 96 e8 0 00 00 01 10 00 0c 29 96 e8 0 00 00 00 11 00 00 0c 00 30 20 00 01 10 00 00 30 20 00 00 11 00 00 0c 30 20 00 0c 00 0c 00 31 36 32 20 6 6e 67 69 6 33 40 52 69 6e 67 69 6 33 40 52 69 6e 67 69 6 31 36 38 2e 32 2e 38 2e 3 76 65 64 3d 31 39 52 2 32 93 3b 62 72 61 6e 64 49 31 30 31 31 65 64 3 36 44 33 80 23 23 33 46 | 280.40.100, 0911 192.1 1, Dat Port: 5660 19 08 00 45 00 | a. 44, 129 b | |
| User Datagram Pr Session Initiati 0 00 0c 29 53 18 02 06 85 ef 19 20 20 13 20 19 20 21 30 19 20 21 30 19 30 22 20 19 30 20 10 30 10 40 40 10 40 10 10 40 10 10 40 10 10 10 10 10 10 10 10 10 10 10 10 10 | ct 61 00 0C 29 96 e8 0 00 00 0C 1800 00 00 00 11 00 0C 1800 00 00 0C 1800 00 0C 1800 00 0C 1800 00 0C 1800 00 0C 1800 0C 18000 0C 18000 0C 1800 0C 18000 0C 18000 0C 1800 | 100 0.011 192.1 1, Dist Port: 5600 100 0.01 192.1 100 0.01 192.1 100 0.01 192.1 100 0.01 192.1 100 0.01 192.1 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 1000 | a. 44.129 b. 46.200 c. co519(2, 100) b. 49.12915 c. 49.12915 c. to 5406(2) d. 49.12915 c. to 5406(2) d. 49.12915 d. 49.12915 | |
| User Datagram Pr Session Initiati 40 00 0c 20 53 42 00 85 ef 30 20 85 ef 30 20 31 38 40 66 cl 3a 20 31 39 22 0 31 39 32 20 41 43 2c 45 2c 31 39 32 2c 41 43 2c 41 41 45 4c 41 41 41 45 4c 41 41 41 41 41 41 41 41 41 41 41 41 41 41 4 | etocol, Src Port: 5418 ion Protocol (180) c6 51 00 0c 29 96 e8 1 00 00 80 11 40 83 c0 8 10 c4 81 46 377 534 10 20 48 21 46 377 534 10 20 52 69 6e 67 69 6 31 36 32 69 6e 67 69 6 31 36 36 24 34 36 2e 3 72 70 66 62 32 43 95 32 72 70 66 62 32 43 95 32 72 89 64 62 33 30 22 31 54 0 64 63 51 21 54 52 31 54 0 64 63 61 63 61 63 41 32 35 | 100 0.01 19.01 1 0.01 Port: 5668 10 0.01 Port: 5668 11 2.01 Port: 5668 11 2.01 Port: 5668 11 Port: 5668 Port: 5668 11 Po | - 4.4.129 | |

Fig:12 Asterisk acknowledged the invite request

The asterisk server (192.168.48.129) acknowledged the invite request from the host machine (192.168.48.1) and a call was established between the windows host machine (192.168.48.129) and the windows virtual machine (192.168.48.128)

VoIP calls are a built-in feature of Wireshark that can decode RTP data into a playable audio format.



Fig: 13 RTP Player (Listening to the Conversation of the Clients)

Copyright © 2023. Journal of Northeastern University. Licensed under the Creative Commons Attribution Noncommercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/ Voice data was captured through the use of the function called Telephony in Wireshark. The voice data is then played using the RTP player in Wireshark and we managed to listen to the entire conversation.

Step 2: Exploit using Metasploit/ other tools (Known as CVE Exploit)

Denial of Service (DoS) attack on VoIP server (Asterisk)

A DoS attack plans to stop a machine or organization, leaving it unusable to its expected clients. DoS attack force the objective to crash by flooding it with traffic or conveying its information. This attack deprives real users of services or resources, such as workers, associates, or account holders, of the service or resources they expected. We used an SYN flood DoS attack against the asterisk server in this study.

A DoS assault plans to stop a machine or organization, leaving it unusable to its expected clients. DoS assault force the objective to crash by flooding it with traffic or conveying its information

The DoS attack on the asterisk server, which was running on an Ubuntu host, was carried out using Kali Linux. Hping3, a free TCP/IP packet generator, and analyzer built by Salvatore Sanfilippo (also known as Antirez) and included in Kali Linux is comparable to the ping software but has more capabilities than merely making an ICMP request. Hping can be used to deliver a massive amount of TCP traffic to a target while concealing the source IP address and making it appear random or even from legitimate sources. This function in Kali Linux was used to capture and display the amount of traffic being transferred to the server using Wireshark.



Fig: 14 Starting hping3 using Kali Linux

Performing the Syn flag DOS attack on the server

A TCP SYN flood is a sort of DoS attack in which a tremendous quantity of SYN entreaty is shipped off a server trying to overpower it with open connections.

On the Asterisk server, we used the following command to launch an SYN flag DoS attack. sudo hping3 -S --flood -V -p 80 192.168.14.129



Fig: 15 SYN flag dos attack

Wireshark was used to capture the packets that were being sent to the server (192.168.48.129) by Kali (192.168.48.130) using the hping3 tool. The above screenshot illustrates the severity of the packets that were being sent at a time and thereby denying the server even time to respond.

| | AN ATTE | 1 7 2 2 6 d d | e, 11 | | - |
|---|--|--|--|--|---|
| Apply a display filter | c@#1-/> | | | | |
| Time | Source | Destnation | Protocol | Length 3rfs | _ |
| 1 0.000000 | 192.168.48.129 | 192.168.48.130 | TCP | 60 80 + 44585 [RST, ACK] Seq=1 Ack+1 Win+0 Len+0 | - |
| 2.0.000247 | 192.168.48.129 | 192.168.48.150 | TOP | 60 80 * 44586 [RST, ACK] Seq=1 Ack+1 Win=0 Len=0 | |
| 3 0.000325 | 192.168.48.129 | 192.168.48.139 | TOP | 60 00 + 44587 [K57, ACK] Seq=1 Ack=1 kin=0 Len=0 | |
| 4 0.000339 | 192,168,48,129 | 192.168.48.150 | TOP | 60 80 * 44588 [RST, ACK] Seg+1 Ack+1 Win+0 Len+0 | |
| 5 0.000795 | 192.108.48.129 | 192.158.48.139 | TOP | 60 00 + 44107 [K17, ACK] 549+1 ACK+1 Min+0 Len+0 | |
| 5 0.000409 | 192.108.48.129 | 192,108,48,158 | TOP | 60 00 + 44590 [K57, ACK] Seg=1 ACK+1 Min+0 Len+0 | |
| 7 0.000482 | 192.108.48.129 | 192.108.48.139 | TOP | 60 00 + 44391 [KST, ACK] Seq=1 ACK=1 MINHO Lenno | 5 |
| 0 0.000490 | 192.100.40.129 | 192.100.40.190 | 100 | CO DD = 44392 [C31, ACK] Segul Ackel Mine Lenne | |
| 9 0.000343 | 192.108.48.129 | 192.100.40.130 | TOP | to be - 44393 [KS1, ACK] Seg-1 ACK-1 Mine Lene | |
| 11 0 000000 | 192.100.40.129 | 102.100.40.100 | TCR | to be - Added [Dit, Ack] Segra Acket Mine Lenne | - |
| 11 0.000000 | 192.100.40.129 | 192.100.40.100 | 100 | the of a state first, see a segre state to be a | |
| 11.0.000015 | 102 168 48 120 | 192 168 48 158 | TCP | the state of the s | |
| 14.0.000677 | 102.168.48.138 | 192,168,48,129 | TCP | (6 4711 - 6 (SV)) Seed (165) and a seed | |
| 15.0.000446 | 192,168,48,129 | 192,168,48,158 | TCP | ca sh a 44501 [bit] A/2] Sant A/41 Ulawh Lened | |
| 16.0.000604 | 192,168,48,138 | 192,168,48,129 | TCP | 60 44752 + 80 (SVN) 500+0 W(0+522 L0++0 | |
| 17 0.000704 | 192,168,48,130 | 192.168.48.129 | TCP | 60 44733 + 00 [SYW] Segre Win-512 Lene0 | |
| Tame 351 60 byte thernet II, Src Internet Protocol Transmission Cont | es on wire (400 bits); : V%ware_53:c8:61 (00: 1 Version 4, Src: 192. trol Protocol, Src Por | (4) (0:29:53:08:61), Dst: 168.48.129, Dst: 192.1 (1:80, Dst Port: 4460) | V%ware_ef: 168.48.130 1, Seq: 1, | Linterisce UdviceOver_[11:LUCrOM-00/3-4130-3004-2000(2040003), 10 0 16:90 (00:00:29:ef1f6:90) Ack: 1, Len: 0 | |
| 0 00 0c 29 ef f | 6 9a 00 0c 29 53 c8 10 00 40 05 58 7c c0 | 61 08 00 45 📉 ···)·· a5 30 81 c0 a5 ··(··) |)5 a | 4 | |
| 30 82 00 50 4 | H 39 00 00 00 00 75 | 56 60 27 50 14 0···P | 9 ····WK' | r. | |
| 00 00 20 75 0 | 0 00 00 00 00 00 00 | 60 | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Fig: 16 Packets capturing using Wireshark (from Ubuntu to Kali)

The above screenshot illustrates the server (192.168.48.129) trying to respond to the requests being made by Kali (192.168.48.130). The server is now dedicating most of its resources to Kali thereby denying legitimate clients to acquire the resources.

5. Results

Upon evaluation, VoIP systems are vulnerable and because of those vulnerabilities, they are prone to several attacks like Eavesdropping and Denial of Service.

5.1: Result of Eavesdropping Attack

| | Time | Source | Destination | Protocol | Length Info | | | | | | | | |
|-------|-------------|-----------------------|----------------------|---|---------------|---|----------|----------|---------|-----------|----------------|---|---|
| 1 | 0.000000 | 192.168.48.128 | 192.168.48.2 | 2 ONS | 76 Sta | ndard query 0xced4 A dns.msft | ncsi.com | | | | | | |
| 2 | 1.918890 | 192 168 48 129 | 192 168 48 2 | 005 | 27.Sta | Mard duery Breese PTE 179 48 | 1AR_193 | _10+A55r | AFOA | | | | - |
| 3 | 2 733455 | Wireshark - SIP Flow: | s - Eavesdropping.pc | ap.pcapng | | | | | | | | - | u |
| 3 | 2.722987 | | | | | - | | - | | | | | |
| 6 | 2.724499 | Start Time Stop Time | Initial Speaker F | rom | | 10 | Protocol | Duration | Packets | State | Comments | | |
| 7 | 2.843858 | 2.722455 31.934538 | 192.168.48.1 " | Client1" <sip:7003@192< td=""><td>.168.48.1></td><td>"7002" <sip:7002@192.168.48.1></sip:7002@192.168.48.1></td><td>SIP</td><td>00:00:29</td><td>9</td><td>COMPLETED</td><td>INVITE 401 200</td><td></td><td></td></sip:7003@192<> | .168.48.1> | "7002" <sip:7002@192.168.48.1></sip:7002@192.168.48.1> | SIP | 00:00:29 | 9 | COMPLETED | INVITE 401 200 | | |
| 8 | 2.843858 | 2.853543 31.765252 | 192.168.48.129 | Client1" <sip:7003@19< td=""><td>2.168,48,129></td><td><sip:7002@192.168.48.12858181;ob></sip:7002@192.168.48.12858181;ob></td><td>SIP</td><td>00:00:28</td><td>7</td><td>COMPLETED</td><td>INVITE 200</td><td></td><td></td></sip:7003@19<> | 2.168,48,129> | <sip:7002@192.168.48.12858181;ob></sip:7002@192.168.48.12858181;ob> | SIP | 00:00:28 | 7 | COMPLETED | INVITE 200 | | |
| . 9 | 2.845104 | 28.049042 26.2/21/5 | 192,106,45,1 | Client1 <sip: 03@192<="" td=""><td>.108.48.1></td><td>"Client1"<sip: 03@192.168.48.1=""></sip:></td><td>295</td><td>00:00:00</td><td>4</td><td>REJECTED</td><td>SUBSCRIBE 401</td><td></td><td></td></sip:> | .108.48.1> | "Client1" <sip: 03@192.168.48.1=""></sip:> | 295 | 00:00:00 | 4 | REJECTED | SUBSCRIBE 401 | | |
| 10 | 2.846422 | | | | | | | | | | | | |
| | 3 840237 | | | | | | | | | | | | |
| ane | 1: 76 bytes | | | | | | | | | | | | |
| hern | et II, Src: | | | | | | | | | | | | |
| tern | et Protocol | | | | | | | | | | | | |
| er u | Name System | | | | | | | | | | | | |
| - G L | name system | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| _ | | | | | | | | | | | | | |
| 00 | 50 56 ea 10 | | | | | | | | | | | | |
| | 3e ec 3a 00 | | | | | | | | | | | | |
| 00 | AD | | | | | | | | | | | | |

Fig: 17 Session Initiation Protocol (SIP) Flow Diagram

The SIP flows are shown in the screenshot above. A SIP User Agent (UA) client initiates a session with the called endpoint User Agent Server (UAS) by sending a request to the called endpoint UASSIP, Uniform Resource Locator (URL). Assuming that the User Account Control (UAC) knows the UAS IP address, the request can be submitted. In the event that the client can't be found, the UAC sends the request to an intermediary or side-tracks it to the server. Until the client is found, the request might be steered to a few servers. After the SIP address is set out to an IP address, the request is directed off the UAS. Assuming the client acknowledges the call, the client's abilities are conceded, and the conversation starts.

The SIP flows allowed us to listen to the conversation that took place between the Windows host machine (192.168.48.1) using X-Lite and the Windows VMware machine using MicroSIP softphones (192.168.48.128)



Fig: 18 RTP Player (Listening to the Conversation of the Clients)

Through the RTP Player, we were able to play SIP flows and listened to the conversation that took place between the Windows host machine (192.168.48.1) using X-Lite and the windows VMware machine (192.168.48.128).

5.2: Result of Denial-of-Service Attack

In the figure below, we can see that Kali Linux (192.168.48.130) managed to perform a DOS attack on the asterisk server (192.168.48.129). Immediately the attacker creates a server connection but does not complete it. The server will have to devote more resources to waiting for half-opened connections, which may cause legitimate traffic to become unresponsive.

Journal of Northeastern University Volume 26 Issue 01, 2023

| DOS.pcap.pcapng | | | | | | | - | σ | 1 |
|-------------------------------|------------------------|--------------------------|---------------|----------------------------------|---------------------|-----------------|---|-----|---|
| le Edit View Go | Capture Analyze Stat | istics Telephony Wireles | s Tools He | 2 | | | | | |
| Interior a decision fiber - c | | 1 T 🗶 🗮 d d | - 4 12 | | | | | - | , |
| Time | Source | Ownership | Pentroval | and his | | | | - | É |
| 25.0.000767 | 192,168,48,130 | 192,168,48,129 | TCP | 60 44739 + 80 [SVN] Secol Mine | 512 Leneb | | | | C |
| 26 0.000784 | 192.168.48.130 | 192.168.48.129 | TCP | 60 44740 + 80 [SYN] Seare Mine | 512 Len=0 | | | | |
| 27 0.000797 | 192,168,48,130 | 192.168.48.129 | TCP | 60 44741 + 80 [SYN] Seg=0 Win- | 512 Len=0 | | | | E |
| 28 0.000807 | 192.168.48.130 | 192.168.48.129 | TCP | 60 44742 + 80 [SYN] Seg=0 Win- | 512 Len=0 | | | | Г |
| 29 0.000817 | 192.168.48.130 | 192.168.48.129 | TCP | 60 44743 + 80 [SYN] Seq=0 Win- | 512 Len=0 | | | | Ξ |
| 30 0.000827 | 192.168.48.130 | 192.168.48.129 | TCP | 60 44744 + 80 [SYN] Seq=0 Win- | 512 Len=0 | | | | F |
| 31 0.000835 | 192.168.48.130 | 192.168.48.129 | TCP | 60 44745 + 80 [SYN] Seq=0 Win- | -512 Len=0 | | | | |
| 32 0.000844 | 192.168.48.130 | 192.168.48.129 | TCP | 60 44746 + 80 [SYN] Seq=0 Win- | -512 Len=0 | | | | Ē |
| 33 0.000853 | 192.168.48.130 | 192.168.48.129 | TCP | 60 44747 + 80 [SYN] Seq+0 Win+ | 512 Len=0 | | | - 0 | - |
| 34 0.000862 | 192.168.48.130 | 192.168.48.129 | TCP | 60 44748 + 80 [SYN] Seq=0 Win- | 512 Len=0 | | | | |
| Frame 25: 60 byte | s on wire (400 bits) | , 60 bytes captured (4 | 180 bits) on | interface \Device\NPf_{11CAEFD4- | 8873-4158-9844-2888 | C2E46DE9}, id 0 | | | Î |
| Ethernet II, Src: | Vibuare_ef:f6:9a (00 | :0c:29:ef:f6:9a), Dst: | : Where_53: | :8:61 (00:0c:29:53:c8:61) | | | | | |
| Internet Protocol | Version 4, Src: 192 | .168.48.130, Dst: 192. | 168.48.129 | | | | | | |
| Transmission Cont | rol Protocol, Src Po | rt: 44739, Dst Port: 8 | 10, Seq: 0, | en: 0 | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| 00 00 0c 29 53 cl | 8 61 00 0c 29 ef f6 | 98 08 00 45 00)5 |) | | | | | | |
| 10 00 28 69 97 0 | 0 00 40 05 2e e5 c0 | al 30 82 c0 al .(1. | | - | | | | | |
| 20 30 81 ae c3 0 | 0 50 15 26 Se 17 28 | 1c 4a e5 50 02 0 ···· | ·P-8 ··(·)· | | | | | | |
| 10 02 00 06 3c 0 | 0 00 00 00 00 00 00 00 | 60 | | | | | | | |

Fig: 19 SYN flood attack packets

The Figure below illustrates that the server (192.168.48.129) is resetting connections initiated by the attacker (192.68.48.130). This limits the server's time in responding to legitimate requests and thereby ends up wasting resources on the attacker.

| Description Description Publical Largett Definition 0. Ten Source Description Source Description Description 0. Ten Source Description Source Description Description 0. Description Source Description Description Description Description 0. Description Description Description Description Description Description Description 0. Description Description Description Description Description Description Description 0. Description Description Description Description Descriptin Description < | |
|---|---|
| The Source Centration Patical Langth 100 1 0:000000 120:1616.48.122 152:161.48.1120 TCP 60 80 + 44555 [ES7, ACC] Seq1 Ack+1 Min+0 Len+0 2 0:000247 132:161.48.1120 152:161.48.1120 TCP 60 80 + 44555 [ES7, ACC] Seq1 Ack+1 Min+0 Len+0 3 0:000375 132:161.48.1120 152:161.48.1100 TCP 60 80 + 44555 [ES7, ACC] Seq1 Ack+1 Min+0 Len+0 4 0:000375 132:161.48.1120 152:161.48.1100 TCP 60 80 + 44558 [ES7, ACC] Seq1 Ack+1 Min+0 Len+0 5 0:000375 132:161.44.122 132:161.44.130 TCP 60 80 + 44558 [ES7, ACC] Seq1 Ack+1 Min+0 Len+0 6 0:000449 132:161.44.122 132:161.44.130 TCP 60 80 + 44551 [ES7, ACC] Seq1 Ack+1 Min+0 Len+0 7 0:000452 132:161.44.120 132:161.44.130 TCP 60 80 + 44552 [ES7, ACC] Seq1 Ack+1 Min+0 Len+0 9 0:000555 132:161.44.120 132:161.44.130 TCP 60 80 + 44552 [ES7, ACC] Seq1 Ack+1 Min+0 Len+0 10 0:00055 | |
| 1 8,000000 192,164,41,122 192,164,41,120 TCP 60 89 4455 [K57, ACS Seq-1 Adx3 Michael Lende 3 8,000325 192,164,41,122 192,164,41,100 TCP 60 89 44555 [K57, ACS Seq-1 Adx3 Michael Lende 4 8,000335 192,164,41,122 192,164,41,100 TCP 60 89 44555 [K57, ACS Seq-1 Adx3 Michael Lende 5 8,000395 192,164,41,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 6 8,000401 192,164,41,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 7 8,000452 192,164,41,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 8 8,000451 192,164,41,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 8 8,000451 192,164,41,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 9 8,000455 192,164,41,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000455 192,164,41,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000455 192,164,41,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000454 192,166,44,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000454 192,166,44,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000454 192,166,44,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000454 192,166,44,122 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000454 192,166,44,129 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000454 192,166,44,129 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000454 192,166,44,129 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000454 192,166,44,129 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 18 8,000454 192,164,44,129 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 19 8,000454 192,164,44,129 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Michael Lende 19 8,000454 192,164,44,129 192,164,41,100 TCP 60 89 44559 [K57, ACS Seq-1 Adx3 Mic | |
| 2 8,000247 192,164,44,122 192,164,44,129 17C* 66 80 + 44555 [857, ACS Seq-1 ACx1 Min+0 Len+0 4 8,000335 192,164,44,122 192,164,44,120 17C* 66 80 + 44555 [857, ACS Seq-1 ACx1 Min+0 Len+0 5 8,000355 192,164,44,122 192,164,44,130 17C* 66 80 + 44585 [857, ACS Seq-1 ACx1 Min+0 Len+0 6 8,000480 192,166,44,122 192,164,44,130 17C* 66 80 + 44598 [857, ACS Seq-1 ACx1 Min+0 Len+0 6 8,000481 192,166,44,122 192,164,44,130 17C* 66 80 + 44598 [857, ACS Seq-1 ACx1 Min+0 Len+0 8 8,000482 192,166,44,122 192,164,44,130 17C* 66 80 + 44598 [857, ACS Seq-1 ACx1 Min+0 Len+0 9 8,000545 192,166,44,122 192,164,44,130 17C* 66 80 + 44591 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00055 192,164,44,122 192,164,44,130 17C* 66 80 + 44591 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00055 192,164,44,122 192,164,44,130 17C* 66 80 + 44591 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00055 192,164,44,122 192,164,44,130 17C* 66 80 + 44591 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,122 192,164,44,130 17C* 66 80 + 44595 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,122 192,164,44,130 17C* 66 80 + 44595 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,122 192,164,44,130 17C* 66 80 + 44595 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,122 192,164,44,130 17C* 66 80 + 44595 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,122 192,164,44,130 17C* 66 80 + 44597 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,122 192,164,44,130 17C* 66 80 + 44597 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,122 192,164,44,130 17C* 66 80 + 44597 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,122 192,164,44,130 17C* 66 80 + 44597 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,120 192,164,44,130 17C* 66 80 + 44597 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,120 192,164,44,130 17C* 66 80 + 44597 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,120 192,164,44,130 17C* 66 80 + 44597 [857, ACS Seq-1 ACx1 Min+0 Len+0 10 8,00056 192,164,44,120 192,164,44,130 17C* 66 80 + 44597 [857, ACS Se | |
| 3 0.00023 352.166.46.127 32.166.46.128 32.166.46.128 102 107 60 80 44557 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 6 0.00035 352.166.46.129 322.166.46.128 102.166.46.128 107 60 80 44558 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 6 0.00042 132.166.46.129 132.166.46.129 132.166.46.129 107 60 80 44558 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 8 0.00045 132.164.41.29 132.166.44.129 132.166.44.130 TCP 60 80 44592 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 8 0.00045 132.164.41.29 132.166.44.129 132.166.44.130 TCP 60 80 44592 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 9 0.00045 132.164.41.29 132.166.44.130 TCP 60 80 44592 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 10 0.00044 132.164.41.29 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 11 0.00004 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000051 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000054 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000054 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000054 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000054 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000054 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000054 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000054 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000054 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000054 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 13 0.000057 132.166.46.129 132.166.44.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 14 0.000057 132.166.46.129 132.166.46.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 15 0.000057 132.166.46.129 132.166.46.130 TCP 60 80 44595 [\$57, ACK] Seq:1 ACk=1 Min+0 Len+0 14 0.000057 1 | |
| 4 0.000239 312-108.44.129 32-108.44.130 TCP 60 80 + 44588 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 6 0.000409 312-108.44.129 32-108.44.130 TCP 60 80 + 44598 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 6 0.000409 312-108.44.129 32-108.44.130 TCP 60 80 + 44598 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 8 0.000408 312-108.44.129 32-108.44.130 TCP 60 80 + 44592 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 9 0.000408 312-108.44.129 32-108.44.130 TCP 60 80 + 44592 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 10 0.000408 312-108.44.129 32-108.44.130 TCP 60 80 + 44592 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 10 0.000408 312-108.44.129 32-108.44.130 TCP 60 80 + 44592 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 11 0.000041 312-108.44.129 32-108.44.130 TCP 60 80 + 44595 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 11 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 + 44595 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 13 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 + 44595 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 13 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 + 44597 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 13 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 + 44597 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 13 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 + 44597 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 13 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 + 44597 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 14 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 + 44597 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 14 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 + 44597 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 14 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 + 44597 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 14 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 + 44597 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 14 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 44597 [\$7], ACK [\$54-0 ACk+1 Kin+0 Len+0 14 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 44597 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 35 0.000056 312-108.44.129 32-108.44.130 TCP 60 80 44597 [\$7], ACK [\$94-1 ACk+1 Kin+0 Len+0 14 0.000056 312-108.41.129 32-108.44.130 TCP 60 80 44597 [\$7], | |
| 3 0.000279 392.164.46.1279 132.164.46.138 TCP 60 80 44559 [857, ACC] Seq1 ALX=1 Kin-0 Len-0 7 0.000482 392.164.46.129 132.164.46.129 132.164.46.129 132.164.46.129 132.164.46.129 8 0.000482 132.164.46.129 132.164.46.129 132.164.46.129 132.164.46.129 132.164.46.129 9 0.000485 132.164.46.129 132.164.46.129 132.164.46.129 132.164.46.129 132.164.46.129 10 0.00045 132.164.46.129 132.164.46.129 132.164.46.129 132.164.46.129 132.164.46.129 10 0.000564 132.164.46.129 132.164.46.139 TCP 60 80 44595 1857, ACX Seq1 ALX=1 March Marc | |
| 6 0.000000 122.161.40.129 122.164.41.29 127.164.41.20 TCP 60 80 + 44592 (557, ACX) Seq1 At-1 Mine Leme 7 0.000012 122.164.41.29 127.164.41.29 127.164.41.19 TCP 60 80 + 44592 (557, ACX) Seq1 At-1 Mine Leme 9 0.000545 122.164.41.29 127.164.41.19 TCP 60 80 + 44592 (557, ACX) Seq1 At-1 Mine Leme 9 0.000545 122.164.41.29 127.164.41.19 TCP 60 80 + 44592 (557, ACX) Seq1 At-1 Mine Leme 10 0.000555 122.164.41.29 127.164.41.19 TCP 60 80 + 44592 (557, ACX) Seq1 At-1 Mine Leme 11 0.000054 122.164.41.29 127.164.41.19 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000056 122.164.41.29 127.164.41.19 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000056 122.164.41.29 127.164.41.19 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000056 122.164.41.29 127.164.41.19 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000056 122.164.41.29 127.164.41.19 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000056 122.164.41.29 127.164.41.19 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000057 122.164.41.29 127.164.41.19 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000056 122.164.41.29 127.164.41.19 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000056 127.164.41.29 127.164.41.19 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000057 122.164.41.29 TCP 121.64.41.10 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000057 122.164.41.29 TCP 121.64.41.10 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 11 0.000057 122.164.41.29 TCP 121.64.41.10 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 12 0.000057 122.164.41.29 TCP 121.64.41.10 TCP 60 80 + 44595 (557, ACX) Seq1 At-1 Mine Leme 13 0.000057 122.164.41.29 TCP 121.64.41.10 TCP 60 80 + 44597 (557, ACX) Seq1 At-1 Mine Leme | |
| 2 0.000402 1392.164.44.129 132.164 | |
| 8 0.000005 132.163.43.125 122.163.43.126 TCP 00 80 + 44932 [537, ACX] Seqt1 Atk-1 Mine Leme 9 0.000555 132.163.41.125 132.164.41.136 TCP 00 80 + 44932 [537, ACX] Seqt1 Atk-1 Mine Leme 10 0.000565 132.163.41.125 132.164.41.136 TCP 00 80 + 44595 [537, ACX] Seqt1 Atk-1 Mine Leme 11 0.000561 132.163.41.125 132.164.41.136 TCP 00 80 + 44595 [537, ACX] Seqt1 Atk-1 Mine Leme 12 0.000561 132.163.41.125 132.164.41.136 TCP 00 80 + 44595 [537, ACX] Seqt1 Atk-1 Mine Leme 13 0.000561 132.163.41.125 132.164.41.136 TCP 00 80 + 44595 [537, ACX] Seqt1 Atk-1 Mine Leme 13 0.000561 132.163.41.125 132.164.41.136 TCP 00 80 + 44595 [537, ACX] Seqt1 Atk-1 Mine Leme 14 0.00057 132.163.41.125 132.164.41.136 TCP 00 80 + 44597 [537, ACX] Seqt1 Atk-1 Mine Leme 14 0.00057 132.163.41.125 132.164.41.136 TCP 00 80 + 44597 [537, ACX] Seqt1 Atk-1 Mine Leme | |
| 9 0.000553 1292.163.45.1229 1322.163.45.126 TCP 60 00 + 45939 [557, ACS] Seq21 ACA-1 Minde Lemed 10 0.000555 1292.163.45.1229 1322.163.45.126 TCP 60 00 + 45939 [557, ACS] Seq21 ACA-1 Minde Lemed 12 0.000564 1292.163.45.1229 1322.163.45.130 TCP 60 00 + 45935 [557, ACS] Seq21 ACA-1 Minde Lemed 12 0.000563 1322.163.45.1229 1322.163.45.130 TCP 60 00 + 45955 [557, ACS] Seq21 ACA-1 Minde Lemed 13 0.000563 1322.163.45.123 1322.163.45.130 TCP 60 00 + 45957 [557, ACS] Seq21 ACA-1 Minde Lemed 14 0.000577 1322.163.45.133 1322.163.45.130 TCP 60 00 + 45957 [557, ACS] Seq21 ACA-1 Minde Lemed 15 0.000565 1321.163.45.123 1322.163.45.130 TCP 60 00 + 45957 [557, ACS] Seq21 ACA-1 Minde Lemed 14 0.000577 1322.163.45.133 1322.163.45.130 TCP 60 00 + 45957 [557, ACS] Seq21 ACA-1 Minde Lemed | |
| 10 0.000575 127.168.4.1.127 127.168.4.1.126 107 0.01 | |
| 11 0.000004 132.108.48.129 132.108.48.139 107 60 00 + 44995 [K57, KK1] Seq24 JKL+1 Min-0 Len-0 12 0.000051 132.108.48.139 107.018.48.130 107.018.48.130 107.018.48.130 13 0.000563 132.108.48.139 102.108.48.130 107.018.019 107.018.019 13 0.000563 132.108.48.139 102.108.48.130 107.0109 60 00 + 44592 657, JKC1 Seq24 JKL+1 Min-0 Len-0 14 0.000577 132.108.48.130 107.0109 60 4071 ± 60 [S57, JKC3 Seq24 JKL+1 Min-0 Len-0 15 0.000564 132.108.48.129 102.108.48.129 107.0109 60 4071 ± 60 [S57, JKC3 Seq24 JKL+1 Min-0 Len-0 15 0.000564 132.108.48.129 107.0109 60 4071 ± 60 [S57, JKC3 Seq24 JKL+1 Min-0 Len-0 | |
| 12 θ.000053 152.13.64.41.279 152.13.64.41.19 1CP 60 80 + 44995 [C57, AC] [Seq2] A2C+1 [sin=0] [sin=0] 13 θ.000057 152.13.64.41.29 152.14.64.41.9 1CP 60 80 + 44995 [C57, AC] [Seq2] A2C+1 [sin=0] 14 θ.000057 152.13.64.41.19 152.13.64.41.19 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+1 [sin=0] 15 θ.000056 152.13.64.41.19 152.13.64.41.19 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+10 1CP 60 84731 ± 60 [S57, AC] [Seq2] A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] [A2C+10 1CP 60 80 + 44995 [S57, AC] [Seq2] [A2C+10 1CP 1CP 1CP 1CP <t< td=""><td></td></t<> | |
| 13 * 0000000 192-180-081-127 192-186-48-128 (CF 60 00 * 00 * 00 * 000 * | |
| 24 4000077 2527200-901200 1527200-90120 1527200-90120 100 00 4000 4000 5800 4000 5800 4000 4000 | _ |
| The second and the second states of the second state for a second state for a second state second states and second st | _ |
| 16.0.000/04 102.168.48.150 102.168.48.120 T/P 60.44712 a 80 (SVII) Secul VisuS12 Lenut | _ |
| 19 a debite 1 162 163 153 153 153 155 156 159 66 44731 - 86 [216] serve bland blands | |
| A ANALYSI DECOMPANY DECOMPANY IN THE STATE OF STATES | |
| rame 35: 60 bytes on wire (400 bits), 60 bytes captured (400 bits) on interface \Device\MPF_[liCAF04-B073-4150-9544-2005C2E46019], 18 0 | |
| thernet II, Srci Whare_SJcGidi (00:00:129:5JcGidi), Dut: Whare_efif6:9a (00:00:129:efif6:9a) | |
| internet Protocol Werston 4, 5rc: 192.106.40.129, 031: 192.106.46.130 | |
| Transmission Control Protocol, Src Port: 80, Dst Port: 44601, Seq: 1, Ack: 1, Len: 0 | |

Fig:20 SYN flood attack server packets captured

6. Analysis

The goal of this research is to examine the vulnerability assessment and pen-testing of the VoIP infrastructure. That is, determining how the scenario is generated and what occurs to the VoIP infrastructure as a result of such attacks. We were able to hear the voice communication between the two clients during the eavesdropping attack, indicating that the RTP protocol is not secure because it allows a third party to hear the conversation word for word. There is no

security in the RTP protocol. The server was subjected to an SYN flood DOS attack, which prevented the server from allocating resources to appropriate clients due to the lack of a security mechanism in

place to identify and prevent such attacks. Pen-testing is a very important process in such scenarios as through this process we come to know about the security and weakness of our VoIP call infrastructure. As described above after testing our VoIP system we can prevent the system from malicious users/intruders to compromise our personal information.

7. Conclusion& Future Work

The goal of this research is to examine the VoIP infrastructure's vulnerability and conduct a Pen-Testing. As we have seen in this study, eavesdropping on a phone call conversation is simple and rapid when utilizing a MITM attack tool like Wireshark to sniff the data. In VoIP evaluation, pen-testers ought to attempt to complete this attack to check whether listening in is feasible. To combat this type of attack associations ought to execute the Secure Real-Time Protocol, a safe protocol that encodes information being traded, making it hard for an assailant to interpret the information and stand by listening to the discussion regardless of whether it is blocked. We also used the hping3 tool in Kali Linux to perform an SYN flood DoS attack against the server, knowing the server's IP address made it very easy to carry out the attack.

In VoIP assessments, pen-testers ought to attempt to complete this attack to check whether listening in is feasible. To combat this type of attack associations ought to execute the Secure Real-Time Protocol, a safe protocol that encodes information being traded, making it hard for an assailant to interpret the information and stand by listening to the discussion regardless of whether it is blocked.

As a result, a few prevention methods must be implemented to prevent the VoIP server's attack. To detect unusual traffic patterns, an Intrusion Detection System (IDS) should be installed. If your onsite firewall supports it, set up an SYN attack threshold and SYN flood defense system. Installing rate-limiting networking equipment and commercial solutions to gain network visibility and the capacity to monitor and analyze traffic across different network segments. The outcome of our research can be used to troubleshoot information in cases of protection vulnerabilities impacting VoIP systems. They can also help with a better understanding of VoIP vulnerabilities and threats, which will lead to better network architecture in the future.

References

1. Rehman UU, Abbasi AG. Security analysis of VoIP architecture for identifying SIP vulnerabilities. In2014 International Conference on Emerging Technologies (ICET) 2014 Dec 8 (pp. 87-93). IEEE.

2. Carrillo-Mondéjar, J., Martinez, J. L., & Suarez-Tangil, G. (2022). On how VoIP attacks foster the malicious call ecosystem. Computers & Security, 102758.

3. 28th USENIX Security Symposium (USENIX Security 19), USENIX Association, Santa Clara, CA (2019), pp. 1327-1340

4. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "Sip: Session Initiation Protocol", RFC3261: Internet Engineering Task Force (IETF), June 2002.

5. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC3550: Internet Engineering Task Force (IETF), July 2003.

6. Secure Real-time Transport Protocol (SRTP), https://datatracker.ietf.org/doc/rfc3711/

7. Y. M. Koh, K. H. Kwon, "A New Lightweight Protection Method against Impersonation Attack on SIP", Advances in Computer Science and its Applications CSA 2013, Vol. 279, 2014.

8. A. D. Keromytis, "A Look at VoIP Vulnerabilities", Usenix Security Article, Vol. 35, No. 1, February 2010.

9. A. D. Keromytis, "A Comprehensive Survey of Voice over IP Security Research", IEEE Communications Surveys & Tutorials Vol. 14, No.2, May 2012.

10. RFC3261: Internet Engineering Task Force (IETF), June 2002.

11. Shah S, Mehtre BM. An overview of vulnerability assessment and penetration testing techniques. Journal of Computer Virology and Hacking Techniques. 2015 Feb;11(1):27-49.

12. e Canadian Institute of Chartered Accountants Information Technology Advisory Committee, Using an Ethical Hacking Technique to assess Information security Risk, Toronto. http://www.cica.ca/research- and-guidance/documents/ it-advisory-committee/item12038.pdf. Accessed 03 Oct 2013.

13. parks, S., Embleton, S., Cunningham, R., Zou, C.: Automated vulnerability analysis: leveraging control flow for evolutionary input crafting. In: IEEE 23rd Annual Computer Security Applications Conference (2007)

14. Yaqoob I, Hussain SA, Mamoon S, Naseer N, Akram J, ur Rehman A. Penetration testing and vulnerability assessment. Journal of Network Communications and Emerging Technologies (JNCET) www.jncet. org. 2017 Aug;7(8).

15. The MITRE Corporation, Common Weakness Enumeration. http:// www.cwe.mitre.org/. Accessed 03 Oct 2013

16. Open Web Application Security Project. OWASP Top 10 Project. http://www.owasp.org/index.php/category:OWASP_Top_Ten_Project. Accessed 03 Oct 2013
17. SANS Institute. SANS Top 25 Software Errors. http://www.sans. org/top25-softwareerrors/. Accessed 03 Oct 2013

18. Manankova OA, Yakubov BM, SERIKOV T, YAKUBOVA M, MUKASHEVA A. Analysis and Research of the Security of a Wireless Telecommunications Network Based on the IP PBX Asterisk in an OPNET Environment. Journal of Theoretical and Applied Information Technology. 2021 Jul 31;99(14):3617-30.

19. K. Salah, A. Alkhoraidly, "An Opnet based simulation approach for deploying VoIP", International Journal of Network Management, John Wiley & Sons Ltd. V.16 issue 3, 2006, pp. 159-183.

20. A.-B.R. Suleiman, A. Hameed, "Simulation of SIP-Based VoIP for Mosul University Communication Network", College of Electronics Eng., University of Mosul, Mosul, Iraq, 2012.

21. M.Z. Yakubova, "Calculation of Traffic Volume and Type in Asterisk IP PBX Network", Modern Challenges and Decisions of Globalizations. International Conference. New York, USA, 2013

22. H. Al-Saadawi and A. Varol, "Voice over IP forensic approaches: A review," May 2017. doi: 10.1109/ISDFS.2017.7916507.

23. H. Kim, H. Lee, and H. Lim, "Performance of Packet Analysis between Observer and WireShark," in 2020 22nd International Conference on Advanced Communication Technology (ICACT), 2020, pp. 268–271. doi: 10.23919/ICACT48636.2020.9061452

24. I. I. Androulidakis, "SPRINGER BRIEFS IN ELECTRICAL AND COMPUTER ENGINEERING VoIP and PBX Security and Forensics A Practical Approach Second Edition." [Online]. Available: http://www.springer.com/series/10059 25.

Methods Eng 29, 1591-1610 (2022). https://doi.org/10.1007/s11831-021-09631-5