FreeSWITCH as a Kickass SBC

Moises Silva <moy@sangoma.com>
Manager, Software Engineering

facebook.com/Sangoma
twitter.com/Sangoma
youtube.com/SangomaTechnologies
blog.sangoma.com

SANGOMA

Moises Silva <moy@sangoma.com>
Manager, Software Engineering

facebook.com/Sangoma
twitter.com/Sangoma
youtube.com/SangomaTechnologies
blog.sangoma.com

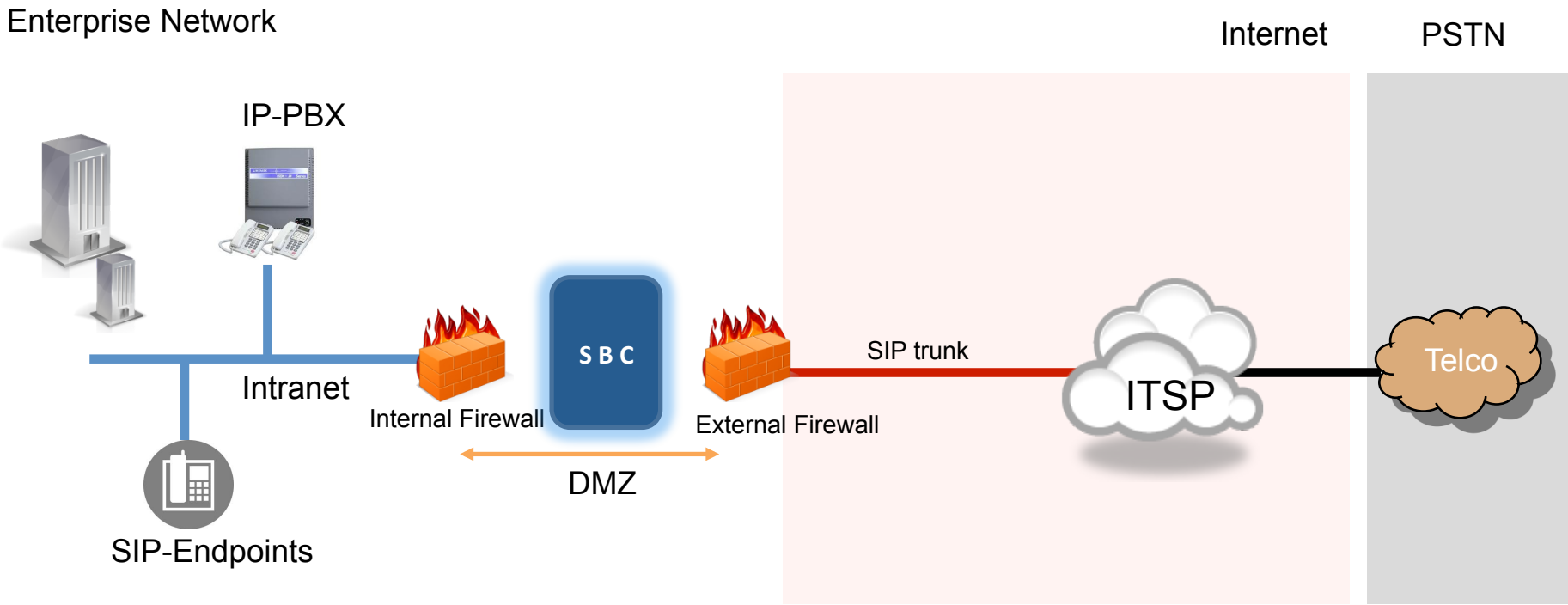
SANGOMA

SBC: The Mythical Beast

- No precise technical definition
- Not standardized anywhere
- B2BUA or Proxy?
- Handles signaling or media?
- More of a marketing term than technical?

SBC: The Mythical Beast

SBC: The Mythical Beast

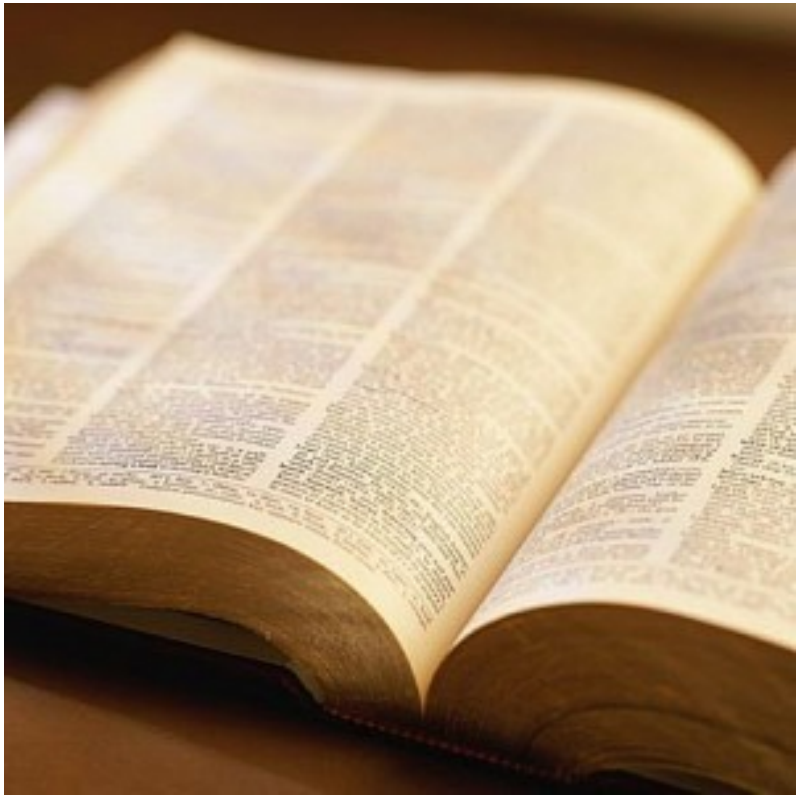


SBC: The Mythical Beast

- Carrier SBC sits at the edge of service provider networks
- Enterprise SBC sits at the edge between customer and service provider

SBC: Definition

- Recently loosely described by RFC 5853



SBC: Definition

- Topology Hiding
- Media Traffic Management
- Fix capability mismatches
- SIP NAT
- Access Control
- Protocol Repair
- Media Encryption
- Threat protection (ie: DoS, Malformed packets)

Topology Hiding

- Hide the service provider or enterprise network implementation details
- Use of RFC 3323 (privacy) not enough
- B2BUA required in most cases

Topology Hiding

- INVITE before topology hiding

```
INVITE sip:callee@u2.domain.example.com SIP/2.0
Via: SIP/2.0/UDP p3.middle.example.com;branch=z9hG4bK48jq9w174131.1
Via: SIP/2.0/UDP p2.example.com;branch=z9hG4bK18an6i9234172.1
Via: SIP/2.0/UDP p1.example.com;branch=z9hG4bK39bn2e5239289.1
Via: SIP/2.0/UDP u1.example.com;branch=z9hG4bK92fj4u7283927.1
Contact: sip:caller@u1.example.com
Record-Route: <sip:p3.middle.example.com;lr>
Record-Route: <sip:p2.example.com;lr>
Record-Route: <sip:p1.example.com;lr>
```

Topology Hiding

- INVITE after topology hiding

```
INVITE sip:callee@u2.domain.example.com SIP/2.0
Via: SIP/2.0/UDP p4.domain.example.com;branch=z9hG4bK92es3w230129.1
Contact: sip:caller@u1.example.com
Record-Route: <sip:p4.domain.example.com;lr>
```

Topology Hiding in FreeSWITCH

- FreeSWITCH is a B2BUA
- Header manipulation using SIP dialplan variables
- Do not enable SDP pass-thru (bypass_media=true)

Topology Hiding Cons

- It is a B2BUA!
- Breaks end to end security mechanisms such as RFC 4474
- Obviously no difference from a MITM attack

Media Traffic Management

- SBC may modify SDP to stay in media path
- Operators enforce the use of certain codecs
- Single point for audition (ie Lawful Interception)

Media Traffic Management

- SDP before media management

```
v=0
o=owner 2890844526 2890842807 IN IP4 192.0.2.4
c=IN IP4 192.0.2.4
m=audio 49230 RTP/AVP 96 98
a=rtpmap:96 L8/8000
a=rtpmap:98 L16/16000/2
```

Media Traffic Management

- SDP after media management

```
v=0
o=owner 2890844526 2890842807 IN IP4 192.0.2.4
c=IN IP4 192.0.2.4
m=audio 49230 RTP/AVP 96
a=rtpmap:96 L8/8000
```

- Session attribute removed enforcing a policy

Media Traffic Management

- Allows operators to use different billing model according to the media type (Voice, Video, Text)
- Fix ‘lost’ or ‘ignored’ BYE issue
- QoS based routing

Media Traffic Management in FreeSWITCH

- Flexible codec configuration in SIP profiles or dialplan
- Media timeout via SIP profile configuration or channel variable “rtp_timeout_sec”
- Flexible SDP manipulation thru variables

Media Traffic Management in FreeSWITCH

- Set codec preferences

```
<action application="set" data="absolute_codec_string=PCMU,G722"/>
<action application="bridge" data="sofia/gateway/myprovider/123456"/>
```

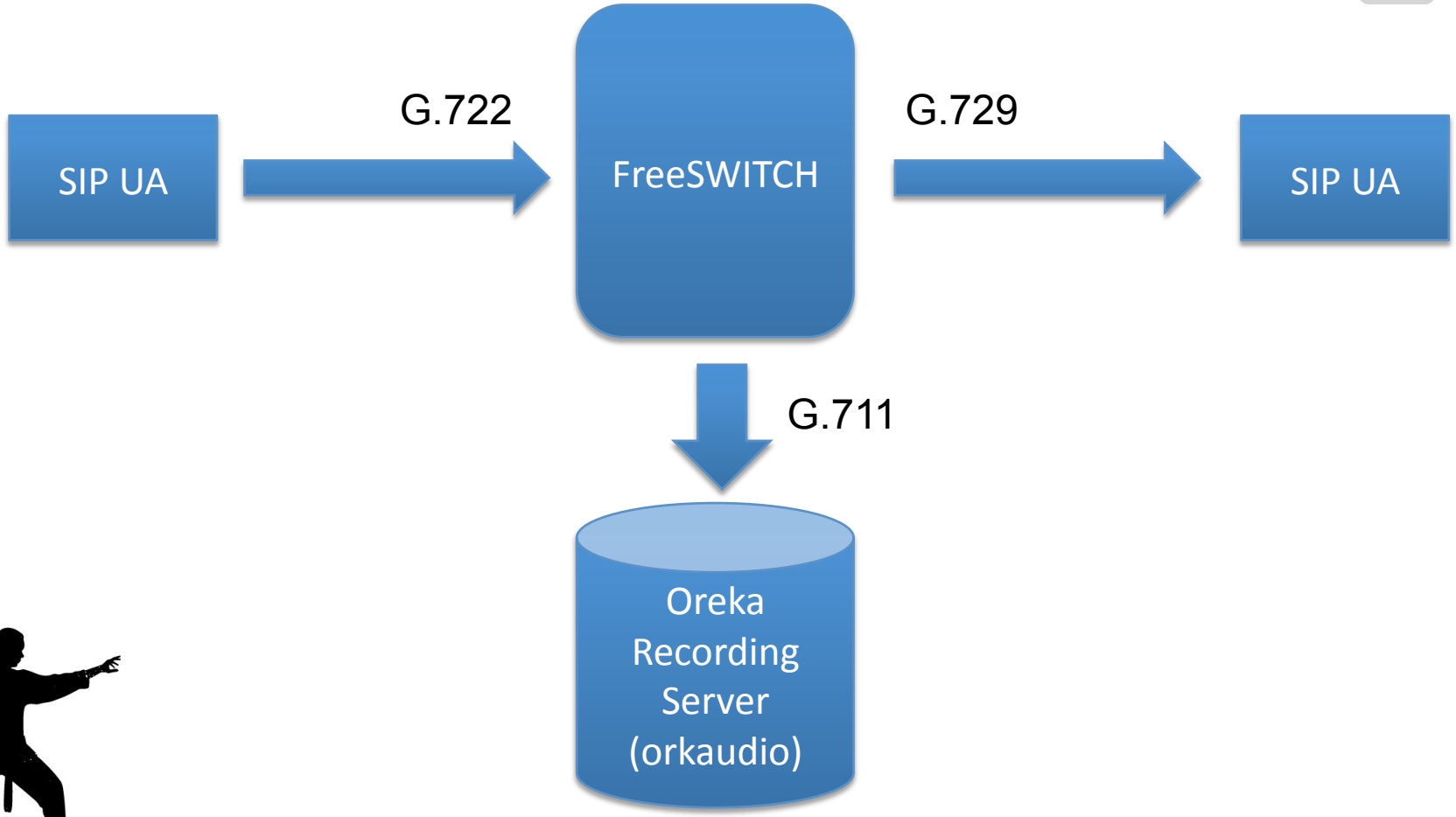
- SDP manipulation

```
<action application="export" data="sip_append_audio_sdp=a=fmtp:18 annexb=no"/>
<action application="bridge" data="sofia/gateway/myprovider/123456"/>
```

Media Traffic Management in FreeSWITCH

- Use record_session for local recordings
- Use oreka_record to send media to recording server (pending merge on official git repo)
  - git@github.com:moises-silva/freeswitch.git

Media Traffic Management in FreeSWITCH



Media Traffic Management Cons

- Increase length of media path
- SBC must be aware of all types of media, slowing adoption of new media features

Fix Capability Mismatches

- Allow interconnection of user agents with different capabilities (ie: codecs)
- IPv4 to IPv6 internetworking
- SIP over UDP to TCP etc

Fix Capability Mismatches in FreeSWITCH

- Virtually every codec in the industry
- HD voice codecs such as G.722 and Siren 7
- Hardware transcoding available

Fix Capability Mismatches in FreeSWITCH

- SIP over UDP/TCP/TLS
- IPv4 and IPv6 support

Fix Capability Mismatches

SIP NAT

- Detect when a UA is behind NAT
- Help maintain NAT bindings alive
- Provides a publicly reachable IP address

SIP NAT in FreeSWITCH

- Automatic RTP adjustment when media comes from different address than advertised on SDP
- Aggressive NAT detection
- STUN and ICE support

Access Control

- It's all about control, yes, sys admins are control freaks
- The network edge of the operator or enterprise is a convenient place to enforce policies
- All SIP and RTP traffic to/from a single point (the SBC)

Access Control in FreeSWITCH

- Flexible ACL configuration available
- mod_limit for call rate and general resource limiting per IP or ‘realm’
- Distributed limits across servers

Protocol Repair

- The world isn’t perfect, broken devices are everywhere
- Not feasible to throw all broken equipment to the garbage (that’d be too easy wouldn’t it?)

Protocol Repair in FreeSWITCH

- Allows matching and fixing of SIP header values
- Allows modifying SDP contents
- Not always possible, SIP stack may discard messages if deemed malformed (ie FUBAR)

Media Encryption

- Desirable to perform encryption on public network

- Internal network may need un-encrypted media (ie, lawful interception, endpoints without SRTP support)

Media Encryption in FreeSWITCH

- TLS / SRTP support
- ZRTP / SRTP support

Threat Protection

- DoS
- Malformed and/or malicious packets
- SIP REGISTER scans (ie: sipvicious)
- SIP INVITE scans

Threat Protection for FreeSWITCH

- DoS limitation using iptables hashlimit (Kristian's famous script)
- iptables hashlimit does not work on TLS obviously
- iptables hashlimit does not help with malformed packets

Threat Protection for FreeSWITCH

- Fail2ban for registration scans
- Depends on log line format, good enough?
- mod_fail2ban makes things easier (by kyconquers on github)

Threat Protection

- Use SIP ACLs
- Even trusted networks can become compromised
- IP spoofing on UDP traffic makes things complicated

Sofia Limits

- Extension to mod_sofia to specify SIP message limits, per host optionally (no ACL yet ☹)
- Uses mod_hash (or any limit interface) to keep track of messages
- Launch ESL event when limit is exceeded
- Malformed packets are accounted for as well

Sofia Limits

```
<profile name="external">

  <settings>
        <!-- settings here -->
  </settings>

  <limits>
        <!-- 100 requests per minute coming from any host -->
        <request-rate host="ANY" method="ANY" rate="100/60" />
        <!-- 1 REGISTER per minute coming from any host -->
        <request-rate host="ANY" method="REGISTER" rate="1/60" />
        <!-- 30 INVITEs per second coming from 10.1.1.1 -->
        <request-rate host="10.1.1.1" method="INVITE" rate="30/1" />
        <!-- 1 malformed message per minute -->
        <request-rate host="ANY" method="MALFORMED" rate="1/60" />
  </limits>

</profile>
```

Conclusion

- Yes, SBCs get in the way, and that's a useful and desirable feature in some businesses
- FreeSWITCH core foundations go a long way in implementing most common SBC features

THANK YOU.

facebook.com/SangomaTech

twitter.com/Sangoma