TLS for SIP and RTP

OpenWest Conference
May 9, 2014

Corey Edwards
tensai@zmonkey.org
@heytensai

v2.0

zmonkey.org

Why TLS?

zmonkey.org

Why TLS?

zmonkey.org

Why TLS?

zmonkey.org

Why TLS?

- Authenticity (man-in-the-middle)
- WebRTC requirement
- Authentication... sort of
- Privacy
  - Metadata
  - DTMF

zmonkey.org

SIP

- SIP is a plain text protocol

INVITE sip:alice@example.com SIP/2.0
Via: SIP/2.0/UDP 192.0.2.110:5060;branch=z9hG4bK742a0872;rport
Max-Forwards: 70
From: <sip:bob@example.com>;tag=as30c330ef
To: <sip:alice@example.com>
Contact: <sip:bob@192.0.2.110:40958>
Call-ID: 2f1496093878509da1074b153b880f9@192.0.2.110:5060
Cseq: 102 INVITE
User-Agent: Some Device v1.0

zmonkey.org

SIP

- SIP is multi-protocol
  - UDP is the original. Has issues with datagram size overflow. Retransmission due to loss.
  - TCP seems like an odd choice, but adds reliability. Plays nicer with NAT.
  - SCTP is a hybrid. Datagram based, with 4-way handshake. Not widely supported.

SIP

- SIP is signaling only
  - Call setup, tear down, progress
  - Messaging, routing
  - No media
  - Session Description Protocol ties to the next protocol, typically Realtime Protocol

zmonkey.org

SIP

- SIP has NAT traversal issues
  - NAT sucks
  - IPs and ports are embedded in SIP, SDP
  - Keep-alives are required
  - Some devices try to “help” by being “smarter” than the endpoints (aka ALG). These are evil.
  - NAT sucks and should die in a fire

zmonkey.org

SDP

- SDP is an offer/accept protocol

```
v=0
o=root 779795551 779795551 IN IP4 192.0.2.1
s=Asterisk PBX 13.13.13
c=IN IP4 192.0.2.1
t=0 0
m=audio 10108 RTP/SAVP 9 0 18 101
a=rtpmap:9 G722/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendrecv
```

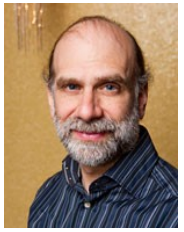
zmonkey.org

RTP

- Realtime Protocol
  - Independent protocol from SIP
  - Supports wide variety of media types, e.g. audio, video, DTMF
  - Also used by H.323, others
  - Has a companion protocol named RTCP which gives feedback reports to the far side

TLS

- TLS started life as SSL
- Developed at Netscape to secure HTTP
- SSLv2 (1995) had some major flaws
  - “Anyone, from the most clueless amateur to the best cryptographer, can create an algorithm that he himself can't break” -Bruce Schneier



- SSLv3 released in 1996

zmonkey.org

TLS

- SSL was turned over to the IETF, became TLS
- Version 1.0 released in 1999, 1.1 in 2006, 1.2 in 2008
- TLS used to downgrade itself to SSL, but this is now considered harmful
- Surprisingly, many sites still use SSLv2

Certificate Authorities

- Biggest problem with encryption is key management
- TLS uses Certificate Authorities (CA)
- Certificate Authorities
  - Sign certificates
  - Are supposed to verify before issuing
  - Can revoke certificates

zmonkey.org

Certificate Authorities

- Client ship with many well-known CA public keys
- Anybody can create their own CA. This can be used in an enterprise, for example.
- Or, skip the whole thing and have a certificate sign itself

zmonkey.org

TLS + SIP

- SIP over TLS requires either TCP or SCTP
- Simple TLS wrapper around the SIP protocol, just like HTTPS
- No STARTTLS

zmonkey.org

TLS + RTP

- RTP data can be optionally encrypted
- Encryption is easy. Hard part is key negotiation.

zmonkey.org

TLS + RTP

- RTP data can be optionally encrypted
- Encryption is easy. Hard part is key negotiation. 4 options:
  - SDES
  - DTLS
  - MIKEY
  - ZRTP

SDES

- Key is sent in plain text by presumed secure channel
- Fantastically simple
- Each SIP hop must independently negotiate key, do encrypt/decrypt
- No end-to-end security

a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:JmrRpWso6K1EBGw9Q3ua+nTbM86f+0WYZI25fC3x

zmonkey.org

DTLS

- Datagram TLS
- Modifications of TLS to make it datagram friendly
- Adapts to packet loss, out of order delivery
- 4-way handshake with cookie
- Stateless from datagram to datagram
- Keying is end-to-end

zmonkey.org

DTLS Example

- OpenSSL can do DTLS

$ **openssl s_server -dtls1 -accept 5555 -cert server.pem -key server.key**
Using default temp DH parameters
Using default temp ECDH parameters
ACCEPT

$ **openssl s_client -dtls1 -connect localhost:5555**
CONNECTED(00000003)

zmonkey.org

MIKEY

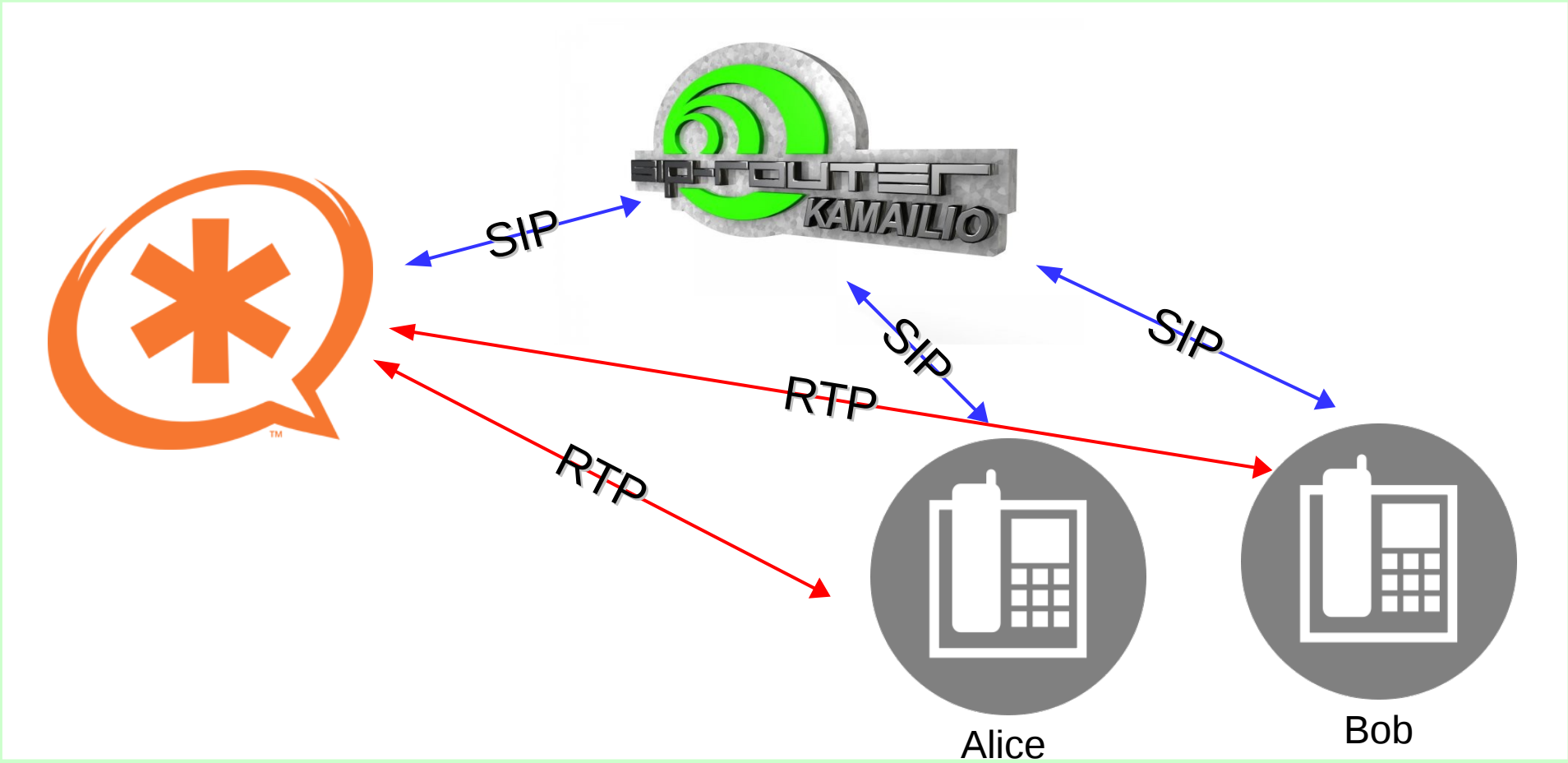
- I don't know anything about MIKEY, thus it's not worth knowing about

zmonkey.org

ZRTP

- Developed by Phil Zimmerman of PGP
- End-to-end opportunistic encryption
- Requires no certificates, authorities
- Vulnerable to man-in-the-middle, but provides authentication hash
- Hash cache increases security over time

Secure VoIP Architecture



zmonkey.org

Create a Certificate

- openssl req -out certreq.csr -new -nodes -days 9999 -keyout key.pem
- Certificate signing request (CSR) stored in “certreq.csr”
- Private key in “key.pem”
- Have CSR signed by CA, or create your own
- openssl x509 -req -days 9999 -in certreq.csr -signkey key.pem -out cert.pem

zmonkey.org

Asterisk

- cat key.pem >>cert.pem
- sip.conf

```
[general]
tlsenable=yes
tlscertfile=/path/to/your/cert.pem
tlscafile=/path/to/your/ca.pem
;tlsbindaddr=0.0.0.0:5071
;bindport=5070
```

zmonkey.org

Asterisk

- sip.conf

```
[peer]
...
transport=tls
encryption=yes
```

zmonkey.org

Kamailio

- kamailio.cfg

```
enable_tls=yes
loadmodule “tls.so”
modparam(“tls”, “config”, “/etc/kamailio/tls.cfg”)

;listen=tls:0.0.0.0:5061
```

zmonkey.org

Kamailio

- tls.cfg

```
method = TLSv1
verify_certificate = no
require_certificate = no
private_key = /etc/kamailio/key.pem
certificate = /etc/kamailio/cert.pem
ca_list = /etc/kamailio/ca.pem

[client:default]
verify_certificate = yes
require_certificate = yes
```

zmonkey.org

Jitsi

- Account
  - SIP ID openwest4XX@ip.address
  - Password soopersecret
- Connection
  - Registrar ip.address
  - Port 5061
  - Proxy ip.address
  - Port 5061
  - Preferred Transport TLS
- Security
  - SDES
  - RTP/SAVP indication MandatoryClick to add Text

zmonkey.org

Limitations

- Gateways can switch to unencrypted
  - sips: URI scheme conveys desire that the entire stream should be encrypted. YMMV.
- PSTN will probably never be encrypted
- Keying protocols are a headache
- Encryption adds latency

zmonkey.org

The End

Questions?

Corey Edwards
tensai@zmonkey.org
@heytensai

TLS for SIP and RTP

OpenWest Conference
May 2014

Corey Edwards
tensai@zmonkey.org
@heytensai