Opportunistic Security in VoIP/SIP

Alexander Traud

I. Introduction

WHILE investigating Voice-over-IP phones based on SIP [RFC3261] (VoIP/SIP phones) from 38 manufacturers, several issues were found in SDES-sRTP [RFC4568], which were common across manufacturers. This White Paper gives a bird's-eye view and collects all the puzzle pieces to establish a technology like SDES-sRTP which enables Opportunistic Security in VoIP/SIP.

A. *Target Audience*

This document has three distinctive target audiences:

a) Manufacturer of VoIP/SIP Phones (Software):
   i) Usability Engineer
   ii) Product Owner
   iii) Software Engineer
   iv) Test Department
b) Administrators for Telephony
c) Security Researchers: Penetration Tester

Instead of creating three White Papers, I believe that combining the knowledge about Opportunistic Security in VoIP/SIP helps all.

B. *Scope*

The following topics are from an end-user point of view. The topics focus on the out-of-the-box experience to gain Usable Security. Therefore, the topics range from theoretic knowledge to hands-on experience (best practices). Every topic calls for action separated into three groups: Makers, Admins, and Tester. The group 'Tester' can be an IT administrator, a security researcher in the role of penetration tester, or a testing department at a phone manufacturer.

1) Makers: The individual topics are about to help you as manufacturer to review your existing implementation or upgrade your product with SIP-over-TLS and SDES-sRTP. The topics outline not only how it shall be done but also what can go wrong. I hope that you understand that – like interoperability and usability – security is not about finishing a 'project' but a continuous topic throughout the lifecycle of the product line.

2) Admins: You as administrator learn how to configure/setup your VoIP/SIP server. Furthermore, you learn what to look for in the phones. This enables you to be your own penetration tester to double-check your configuration of the server and the phones. Furthermore, the topics enable you to double-check the implementation of the phones, so you can file reports yourself (or choose another product).

3) Tester: You as tester get an introduction into SDES-sRTP. Furthermore, I describe what to look for, when it comes to deploy and use a SDES-sRTP implementation; how it was done. Therefore, you learn what was not looked at, which can be a starting point for your own ideas how to test an SDES-sRTP implementation. For example, the business environment scenarios of [RFC5359] were not looked that.

C. *Scope Limits*

1) Technology SDES-sRTP: Although this document is about Opportunistic Security in VoIP, it is specific to SIP with SDP usage for which examples are described in [RFC3665]. Therefore, this document is not about H.323 or proprietary 'Digital' phones. SDES-sRTP has several alternatives like utilizing a Virtual Private Network (VPN) instead, which avoids Deep-Packet Inspecting for example. SDES-sRTP is about endpoint-to-access-edge security (e2ae security) and not about end-to-end security like DTLS-sRTP [RFC7879] and ZRTP [RFC6189]. Although those technologies have benefits over SDES-sRTP, this document here shall help you to move from current cleartext practice to at least endpoint-to-access-edge security. Several implementations show that SDES-sRTP, DTLS-sRTP, and ZRTP can be offered in parallel; a manufacturer can offer all of them in each call.

2) Technical Personnel for Telephony: Formally, Software Usability in 'Usable Security' requires a definition of the user. The phones described in this document are often targeted solely at mid-size enterprises, who do not have their own but an external administrator. Manufacturers think, an end-user 'just' telephones and raises issues with the internal IT coordinator, who raises an issue with the external administrator, who raises the issue with the distributor, who raises the issue with the manufacturer. Such manufacturers try to push away not only the product support but also the liability to that administrator. Nevertheless, the software of these phones scale to be used in large enterprises as well. Additionally, these phones could be used by small enterprises and at home office. Finally, because the trend is towards IP telephony, all existing phones could be replaced by an IP phone nowadays. Consequently, even your mother's phone might be an IP phone in the future. Therefore, the topics raised in this document cover all, from home consumer to professional enterprises. We are talking about consumer electronics and its software. Although written from the perspective of end-users, this document focuses on the technical personnel, who provides telephony.

II. Security Management

A. Responsible Disclosure (Topic 1)

IP phones are part of the Internet of Things (IoT), which can be attacked by everyone. Because phones are sold on second-hand market, a (hopefully white-hat) hacker can get hold of such a phone easily. In such cases, there is no chain of support like manufacturer ↔ distributor ↔ administrator ↔ end-user. Therefore, hackers have to report security issues directly, although they are end-users not with thousands but only one phone.

Call for Action:

Makers	1. offer a contact channel like the E-mail security@example.com
	2. offer a secure communication channel via OpenPGP and S/MIME
	3. provide a polite employee trained in error culture
Admins	test your deployment, your manufacturer might not have done it
Tester	report security issues in a Responsible Disclosure

Security impact: When reports of hackers get lost/blocked by product support, the finding is not fixed. Furthermore some product-support channels still envision themselves as customer support rather than product support: A hacker does not need support. He wants to pass a finding so the product undergoes the required fix. You do not want to get rid of the reporter, you do not want to solve the symptoms, you have to fix the causing issue.

B. Interoperability Testing (Topic 2)

Protocols like SIP are highly flexible and contain later added extensions. Therefore, in a SIP implementation some parts were not tested or extensions were never tested in combination. This phenomena is called Protocol Ossification[1]. For example, I tested SIP Session Timers [RFC4028] and AES-256 [RFC6188 and RFC7714] with SDES-sRTP: Beside calls with fading audio, no audio, and no call establishment at all, I had to report misperceptions even in reference implementations and specifications themselves.

Call for Action:

Makers	1. go for interoperability tests like SIPit of the SIP Forum
	2. offer a contact channel like the E-mail sip@example.com
	3. provide a polite employee trained in error culture, who knows the difference between a feature request and protocol violation
Admins	test your deployment, your manufacturer might not have done it
Tester	report interoperability issues

Security impact: Administrators might not have the time to analyze the cause of an interoperability issue. Their fast and easy solution is to disable the non-working feature. Therefore, if sRTP does not work, administrators might simply disable it, leaving the end-user in an insecure environment. Consequently, interoperability is one of the puzzle pieces to achieve Opportunistic Security in VoIP/SIP.

C. Software Updates – Obligation to Provide (Topic 3)

Consumers are told to install the latest software updates to stay secure, because those updates might contain security fixes. To ease this, the manufacturer has the obligation to provide updates. Administrators have other things to do than searching in Web portals, FTP listings, or Drop boxes for the latest updates on a daily basis.

[1] https://blog.cloudflare.com/why-tls-1-3-isnt-in-browsers-yet/

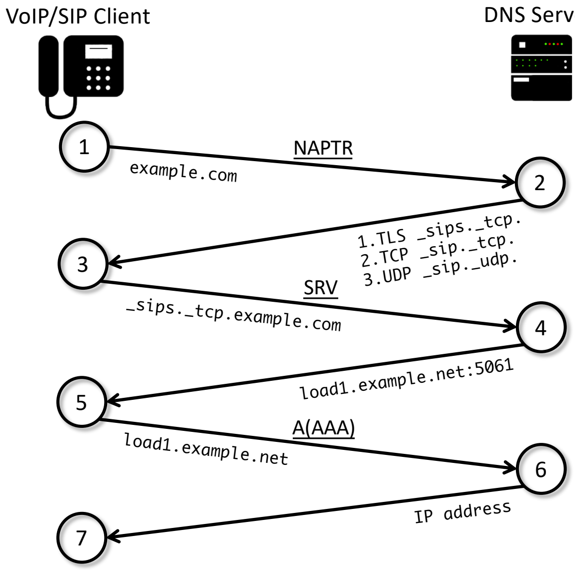

Fig. 1. RFC3263: DNS queries to locate an User-Agent Server

Call for Action:
Makers 1. offer automatic software updates, and
2. offer a newsletter just about software updates for those who prefer to update manually, and
3. offer release candidates for those who prefer to pre-test
Admins A) enable automatic software updates, or
B) if you prefer manual updates, sign up to an ‘software update newsletter’
Tester check that the automatic update actually works (URL changed, server certificate untrusted, . . .)
Security impact: This allows a phone to fetch the latest software and security enhancements out-of-the box.

III. Security Technology

A. Ideal Deployment: DNS-NAPTR (Topic 4)

When the VoIP phone is deployed, only a SIP-URI (user@example.com) and a password is required/entered. This is possible thanks to [RFC3263], which uses DNS to locate the server, transport, and port (see figure 1. In an ideal world, just a client certificate is installed (for example via USB) and the phone extracts its SIP-URI from that certificate.

Call for Action:
Makers add DNS-NAPTR, enable it on default
Admins add DNS-NAPTR and DNS-SRV to your domain
Tester is the DNS-SRV query not for `_sip._tls` but `_sips._tcp`
Security impact: DNS-NAPTR allows a phone to use SIP-over-TLS out-of-the box. SDES-sRTP requires[2] such an encrypted SIP channel.

[2] draft-ietf-sipbrandy-osrtp-07; section 4

TABLE I
SECURITY LEVELS OF OPENSSL 1.1.1B

Level	Bit Strength	RSA Keys	DH Keys	ECC Keys	SSL/TLS Version	Forward Secrecy	Certificate Signature	Cipher MAC
1	80	1024	1024	160	SSL 3.0 or newer		at least SHA-1	at least SHA-1
2	112	2048	2048	224	TLS 1.0 or newer		at least SHA-2	
3	128	3072	3072	256	TLS 1.1 or newer	required		
4	192	7680	7680	384	TLS 1.2 or newer	required		at least SHA-2
5	256	15360	15360	512	same as Level 4	required		

TABLE II
SECURITY LEVELS OF ECDHE GROUPS

Level	X25519	P-256	X448	P-521	P-384	(secp224r1)
3	✓	✓	✓	✓	✓	✗
4	✗	✗	✓	✓	✓	✗
5	✗	✗	✗	✓	✗	✗

B. Signaling Encryption: SIP-over-TLS (Topic 5)

When SIP-over-TLS is available as transport, the phone tries a TLS handshake with the remote party (SIP registrar, SIP proxy server, or callee). The idea of encryption is, that it is strong enough that nobody can crack it with all known computing power within a certain amount of time. As of today, the measurement for this is the bit strength. In the TLS handshake a cipher suite with a certain bit strength gets negotiated. If a cipher suite with Forward Secrecy was chosen, the handshake also negotiates the modulo-prime group (in case of DHE) or an elliptic-curve group (in case of ECDHE). Again, both come with a certain bit strength. Because these factors determine the overall bit strength of the connection, the open-source library OpenSSL 1.1 introduced the concept of Security Levels[3] (see table I).

Call for Action:

Makers 1. offer at least TLS in version 1.3 to prepare for ESNI[4], and
2. offer an user interface to set the minimum Security Level

Admins upgrade your server to support at least Security Level 3 (see topic 6.1)

Tester 1. check the TLS version, cipher suites, groups, and hashes (via Port Mirroring and Wireshark)
2. try downgrade attacks by not accepting TLS 1.2, TLS 1.1, and TLS 1.0

Cipher suites and (EC)DHE groups got a lot of attention since the year 2013. The OpenSSL Security Level is one approach. If you cannot update to OpenSSL 1.1 immediately, another approach was to configure suites and groups manually in source code. Suite lists found on the Internet are not recommended because they expect a specific OpenSSL version. Furthermore in OpenSSL, a suite list of `ALL` enables Export and Anonymous Cipher Suites. Even `HIGH` still enabled Anonymous Cipher Suites. For open-source projects, which do not know the underlying OpenSSL version, `HIGH:-COMPLEMENTOFDEFAULT` is recommended. For all other projects, please, consider the current list of Mozilla Firefox[5]. That Web browser has an open-discussion culture why a specific suite was enabled/disabled and why the order was chosen. Are you more clever than the community of Mozilla? Beside the suites, you have to configure/limit the groups, which is why upgrading to OpenSSL 1.1 and its Security Level is recommended. Table II shows the default order of the groups for ECDHE. Since OpenSSL 1.0.2, the default list is at Security Level 3 already but includes Brainpool groups as well. Therefore, the offered group list is a clue, which OpenSSL branch is used. secp224r1 was just included for comparism.

Security impact: With enryption, passive eavesdropping of the transmitted content gets more complicated, which makes SDES-sRTP useful. With ESNI, passive eavesdropping of the meta content gets more complicated, which improves privacy.

C. Signaling Authentication: Certificates: Trust Anchors (Topic 6.1)

Encryption secures against passive listeners. Encryption without Authentication is called Opportunistic Security [RFC7435] because passive listeners are considered attackers already [RFC7258]. If you need protection against active attackers as well, you must be sure the phone talks to the intended server. This is done by authenticating. In the TLS handshake, the message `Certificate` fulfills this. The phone has to trust that certificate, otherwise the connection is not authenticated. There are two ways to trust a certificate:

A) the phone knows the public key of the entity certificate from that server, or
B) the phone knows a trust anchor, which signed that entity certificate.

Alternative A requires that the phone knows all servers beforehand. This is feasible, if an administrator manages the phone. Alternative B requires the server to indicate the signing trust anchor. This is done by sending (all) intermediate certificates (see figure 2).

[3] https://www.openssl.org/docs/man1.1.1/man3/SSL_CTX_get_security_level.html

[4] https://blog.cloudflare.com/esni/

[5] https://www.ssllabs.com/ssltest/clients.html

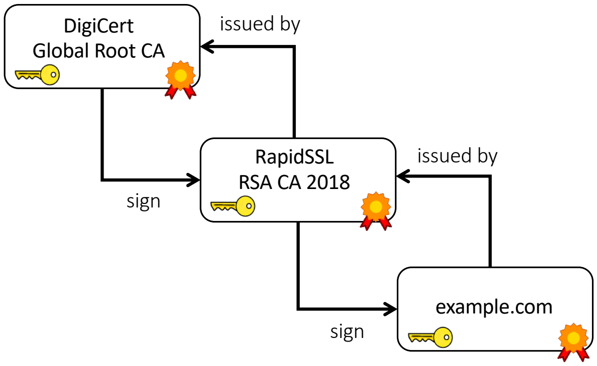

Fig. 2. You create your own private key. You request a certificate authority to sign your public key. The final certificate is then issued by that certificate authority.

The phone comes with trust-anchors built-in, like you are used from your Web browser (or operating system) and its Certificate Manager. Certificate Authorities might get dis-trusted. Therefore, the admin must be able to see the existing trust anchors and be able to disable individual ones. Because a trust anchor is just an issuer name and its public key, the hash algorithm of a root certificate does not matter. Therefore, even SHA-1 based root certificates are of no problem[6]. Like with encryption, there has to be no weak spot in authentication. Consequently, the entity certificate and all intermediate certificates have to fulfill a certain Security Level as well.

Call for Action:

Makers 1. include trust anchors out-of-the box, see <http://www.traud.de/trust-anchors/self-signed.htm>
2. display those installed anchors on all interfaces (phone and Web)
3. allow to disable individual anchors on all interfaces (phone and Web)
4. allow to add own trust anchors, put no limitation on the hash algorithm
5. warn (but allow) an administrator, when he installs a
– RSA based public key with less then 2048 bit
– ECDSA based public key with less then 224 bit

Admins 1. for RSA, go for a well-known certificate authority like GoGetSSL, Let's Encrypt, RapidSSL
2. for ECDSA, go for a well-known certificate authority like DigiCert, GlobalSign, or Sectigo
3. provide the intermediate certificates up to a common trust anchor

Tester 1. check with a self-signed certificate, whether the phone validates trust anchors at all
2. check with an Symantec issued certificate, whether the phone trusts outdated trust anchors
3. check with an MD2, MD5, or RSA 512 bit intermediate certificate (less than Security Level 1)
4. check with a SHA-1 or RSA 1024 bit intermediate certificate (less than Security Level 2)

Only a few well-known certificate authorities support Security Level 3 with RSA. Therefore, I recommend to go for an

[6] https://security.stackexchange.com/a/91918

ECDSA (ECC) based certificate. Because not all VoIP/SIP phones support ECDSA yet, consider a dual-certificate installation; you buy one RSA and one ECDSA based certificate. When an RSA-only phone connects, the server provides the RSA based certificate. With this, you reach Security Level 2. When a ECDSA capable phone connects, the server provides the ECDSA based certificate (Security Level 3).

Security impact: Trust Anchors are the first puzzle piece to support not only encryption but also authentication.

D. Signaling Authentication: Certificates: Hostname Validation (Topic 6.2)

If well-known trust anchors are accepted, an attacker could buy a certificate and still put himself as Man-in-the-Middle (MitM). Therefore, the certificate contains an identification to which server it belongs: servername. The phone validates the entered hostname (of the SIP-URI) with the servername presented in the certificate. For example with OpenSSL, this validation must be enabled manually[7].

Call for Action:

Makers validate the hostname
Admins you cannot[8] use a Wildcard certificate, instead go for
A) one certificate for each server, or
B) a Multi-Domain certificate (SAN)
Tester 1. send a certificate with a wrong hostname like ‘secure.dus.net’ for ‘proxy.dus.net’
2. point the DNS-SRV to a different domain as shown in figure 1

It is important to use not a derived domain name but the one entered by the user. DNS-SRV might redirect to another hostname, for example from `.com` to `.net` like in figure 1. The name `example.net`, derived from DNS, is not the new name for validation, because pure DNS could have been manipulated by an active attacker. The phone has to stick to the originally entered domain.

Security impact: Hostname Validation is the second puzzle piece to support authentication. Without authentication, an active attacker is able to extract the crypto key of the upcoming SDES-sRTP.

E. SSL/TLS Library Version and Defaults (Topic 7)

The TLS library within a phone has an interface to the outside world, either through the various TLS based protocols like SIP and LDAP but also through the configuration interface, which is based on a Web server and therefore on HTTPs. Everything outside accessible can be attacked. Therefore, it is important to update to the latest release within a branch of your TLS library. In OpenSSL prior to version 3, the latest release is identified by a letter, the latest branch is identified by a version number.[9]

In Software Usability, the concepts of defaults is known to reduce mistakes. However a default is not chosen always consciously. Sometimes a value/behavior simply happens to be the default. TLS libraries with branches tend to keep the unconscious defaults within a release, not to disturb the library user by a changed behavior. Therefore, to get the latest ‘good’ defaults, you have to update to the latest branch (or configure manually).

Call for Action:

Makers 1. display the version of all your ‘outside’ libraries (phone and Web)
2. update to the latest branch of your TLS library
3. if you use OpenSSL, configure it with `no-deprecated`
Admins A. update to the latest release of your TLS library, or
B. if your TLS library came with the operating system (OS), update to the latest OS version
Tester check whether the underlying operating system is still maintained
do not get confused by release numbers of the TLS library; some OS backport security fixes

Because a certificate path might contain legacy certificates to maintain compatibility with older devices (which do not support the latest trust anchors), the validation starts at the entity certificate and stops with the first known trust anchor. Manufacturers, check whether your TLS library does this! For example, OpenSSL before 1.0.2a required the trust anchor at the end of the certificate chain always. Consequently, if you use OpenSSL you have to upgrade to at least version 1.0.2 and enable the verification parameter `X509_V_FLAG_TRUSTED_FIRST`. Since OpenSSL 1.1, this parameter is set always.

Security impact: An outdated TLS library indicates an unsupported product.

F. Media Encryption: SDES-sRTP: Crypto Key (Topic 8.1)

With SDES-sRTP, a symmetric key is exchanged between the call parties on the signaling layer, to be more precise in SDP within SIP. A new key is randomly created with each call. Because SDP is a cleartext protocol, this requires SIP to be encrypted for example via SIP-over-TLS. Then, a RTP stream is created with each caller, for each media type (like voice,

[7] https://wiki.openssl.org/index.php/Hostname_validation

[8] RFC5922; section 7.2

[9] https://www.openssl.org/blog/blog/2018/11/28/version/

video, and text). That stream is encrypted with the key from the remote party. In SDP, you see that (binary) key in the line `crypto`, which is Base64 [RFC4648; section 4] encoded because SDP is not a binary but a cleartext protocol. The challenge is to create that key with enough randomness.

Call for Action:

Makers	use a maintained Random-Number Generator (RNG)
Admins	n/a
Tester	decode that Base64 encoded key and check whether it is really binary

Security impact: Protect the RTP stream against (powerful) passive eavesdropping.

G. Media Encryption: SDES-sRTP: Audio-Codec Constant Bitrate (CBR; Topic 8.2)

Research [RFC7587; section 8] has shown, that an audio-codec – which does not send with a constant bitrate (CBR) but a variable bitrate (VBR) – can be decoded even when encrypted. The suggestion is to use only CBR based audio codecs or to switch to CBR when sRTP is in use. This affects for example the audio codecs Speex and Opus.

Call for Action:

Makers	1. request via SDP only CBR 2. send via sRTP only CBR
Admins	disable Speex and Opus, if you are unsure
Tester	1. check the SDP, whether CBR was negotiated 2. check the packet size of each RTP packet, whether VBR is used

Security impact: Protect the RTP stream against passive eavesdropping.

H. Media Authentication (Topic 9)

As with signaling encryption, the received media needs to be authenticated by the phone. In this case, the problem is an active attacker, who sends broken (denial of service) or previous media packets (replay attack) to destroy the conversation. For example, the open-source library libSRTP gives an error when the packet was not authenticated (`err_status_auth_fail`) or has the wrong sequence number (`err_status_replay_old`).

Call for Action:

Makers	1. discard unauthenticated sRTP packets (silence) 2. do not reset the RTP stream in the middle of a call, which creates a new RTP-SEQ 3. do not reset the sRTP session in the middle of a call, which resets the sRTP-ROC
Admins	enable sRTP Authentication if optional
Tester	1. make calls longer than 22 minutes[10] (Roll-Over Counter; sRTP-ROC) 2. make calls longer than 30 minutes[11] (Session Timers), sometime re-INVITEs reset the sRTP-ROC 3. RFC5359 (hold/unhold and multi-call scenarios; I tested only single-call scenarios)

Security impact: Protect the RTP stream against active attackers.

IV. Pieces Come Together

A. Padlock Icon: Open (Topic 10)

With SIP-over-TLS and SDES-sRTP a lot can get wrong. However, the end-user wants to place his call. Furthermore over time, existing encryption/authentication schemes might get obsolete. Opportunistic Security allows to ignore authentication, for example if you phone a server which trust anchor is unknown to you. A phone has several user interfaces like a graphical display. Therefore, a pad-lock icon while calling can be used to identify weak, downgraded, or unauthenticated encryption. If that icon is selectable, the phone can provide an explanation why the call is deemed to be not secure. Then, the end-user can inform the administrator to debug the cause further. Sometimes, it is just a minor configuration error. Sometimes, it is an implementation error. Therefore, that shown explanation does not need to be understandable by the end-user because he hands the explanation over to an expert. However, such an icon must be selectable by the end-user. One approach is to show an open padlock icon, whenever the call does not fulfill the intended Security Level; and no icon when it does.

[10] The RTP sequence counter rolls over after 0xffff packets. When a RTP packet is sent every 20 milliseconds, this results in 21 minutes and 50.7 seconds – if Voice-Activity Detection (VAD) and Discontinuous Transmission (DTX) were disabled.

[11] After 30 minutes, sessions expire on default [RFC4028; section 4]. The refresh of that session happens earlier. Because that refresh might have happened before the sequence counter rolled over, you have to wait until the session is refreshed, while the counter had rolled over.

Call for Action:

Makers	1. show an icon, when TLS is not used
	2. show an icon, when TLS Authentication failed
	3. show an icon, when sRTP was not established
	4. make that icon selectable, describe what failed
	5. provide the overall achieved Security Level, for example via an INFO button
Admins	teach your users to watch for the correct indicators (icons)
Tester	1. test an UDP connection: Is no icon shown?
	2. test an UDP connection with sRTP: Is a misleading icon shown?
	3. test an unauthenticated TLS connection: Is a misleading icon shown?

Security impact: When a new phone (or new firmware) shows an open padlock icon, while the existing deployment does not, the end-user gets aware of a changed situation. This awareness helps to reveal outdated phones and latest developments in security research.

B. Final Words

If you find an error or you have a suggestion, please, drop me an E-mail. My address and the latest version of this document can be found at <http://www.traud.de/voip/>.

Version 1.0
March 19, 2019