

draft-ietf-sip-rfc2543bis-07

Review of section 22 “Security Considerations”

DRAFT v0.1

To be used only within sip@ietf.org mailing list

Ofir Arkin
ofir@atstake.com

At stake Limited
6-8 James St.
London, England
www.atstake.com

Abstract

This paper is a review of the security features of the proposed draft-ietf-sip-rfc2543bis-07 which describes the next version of the **Session Initiation Protocol's (SIP)** RFC.

This document is still a work in progress document, and provided as is.

The document is provided to the sip@ietf.org mailing list members and is intended to be used by this mailing list members only (for the time being until future, more mature, versions will be provided).

22 Security Considerations [Page 113]

First paragraph

Starting the security section with a statement that “*SIP is not an easy protocol to secure*” is not reassuring. The goal is to make people understand the security problems related to SIP (and to VoIP) and not to use the fear factor.

The description that follows gives a better understanding of the problem or at least some reasons for the trouble – many usage scenarios that are different in the participants, architecture and trust relations.

We are not to blame

The second paragraph reminds us an important point – the fact that SIP Signaling has nothing to do with other protocols participating in a VoIP based solution, and their security hazards.

The harsh statement literally suggests that we are on our own with other protocols (quite understandable).

But a major thing is being forgotten here, there are different protocols that combine, together, a complete VoIP based solution and not only SIP Signaling.

Not associating even the MIME portion (body) of SIP is not serious even. I am not familiar with any protocol what so ever that will state that a part of its packet is “not of his responsibility regarding its security”. This is although MIME security is being mentioned. It is better to state that SIP bodies which are MIME types can be secured using S/MIME and to link them with section 21.

It is true that SIP is not responsible for RTP but it is worth mentioning that securing RTP or any other Voice sample transport protocol is an important task in the overall security of a complete VoIP solution based on SIP. In fact some of the solutions in securing SIP will be invaluable when other protocols of the VoIP solution we will be using, will be compromised and abused.

22.1 Attacks and Threat Models

SIP will also be used in closed environments such as a corporate LAN where the threat model will be the same. The threat of someone sniffing the wire is not only an Internet centric thing... and I bet that the aim is more towards the usage of SIP for carriers and in LANs than for it's usage with the public Internet.

The selection of security related hazards is narrowed to only registration hijacking, impersonating a server, and the mention of denial of service attacks (some types not all).

I must say that I do not agree with some of the scenarios presented with the draft.

22.1.1 Registration Hijacking

“The FROM header field of a SIP request, however, can be modified arbitrarily by the owner of the UA”

This scenario occurs when the UA is an IP Phone (or a softswitch) and the UA owner is playing with the configuration of the device.

BUT the RFC writer totally forgets that in the security world there is no such thing as “*Client Security*”, or even trusting a client. What if the client will use its own UA that allows him to change what ever field he wishes? – Just remember *not to trust a client* (the problem is even more emphasized when intelligence is at the end points).

Authentication with Registrars

This is an interesting point because a user needs to enter his user name and password at least at the first time it registers with a registrar.

- What will you do when the registration expires?
- What will happen when a UA needs to re-register after your registration expired?
- Will the UA prompt the user again?
- Is this means that when a user will need to use E911 services and his UA is not registered he will be required to enter his credentials before calling the police or the emergency services?
- Is the UA going to use cached information again and again?

22.1.2 Impersonating a Server

Several venues for a UA's request to be intercepted by some other party:

- Access to the Wire at some point of the routing of SIP Signaling
- DNS poisoning
- SIP Proxy, SIP Redirect, SIP Registrar cracking/poisoning

Problem: What will happen with the Chicago.com and Biloxi.com scenario if both will answer but the Chicago.com's answer will be first to arrive back? There are no mechanisms, that I am aware of, *that will alert the end point that something is wrong and it got an answer from two DIFFERENT redirect servers...*

This will also put fraud detection mechanisms in a hard position.

Where the intelligence is?

I have a problem with the scenario that is being presented in this section:

"This family of threats has a vast membership, many of which are critical. As a converse to the registration hijacking threat, consider the case in which a registration sent to biloxi.com is intercepted by chicago.com, which replies to the intercepted registration with a forged 301(Moved Permanently) response. This response might seem to come from biloxi.com yet designate chicago.com as the appropriate registrar. All future REGISTER requests from the originating UA would then go to chicago.com".

Interesting enough, a UA needs to REGISTER to his SIP Registrar server and the SIP Registrar informs him it had moved...

Is everybody seeing the irony here?

How is the registrar itself telling me it had moved...?

Isn't this my Location Service job rather than the end-point in this particular scenario?

In my opinion there is a difference between an end point which generates a 30x moved for some reason and a SIP entity server.

We need something that will give us non-repudiation...

22.1.4 Tearing Down Sessions

An interesting side note – if the parameters an attacker is introducing with his BYE request are not the same as of the corresponding party the attacker will receive a 481 response back from the UA "Call/Transaction does not exist".

This is not good – it gives an attacker the ability to brute force (guess by mistakes) the correct parameters he needs if the signaling is encrypted or not visible to him.

Since the "To" and "From" fields are predictable (in a lot of cases there is simply nothing to do about this) it leaves an attacker with the Cseq and with the Call-ID field values to brute force/guess.

Another note: if we are to encrypt signaling than any signaling pertaining to come from one of the sides not encrypted should be dropped (*where is logging in all of this?*)

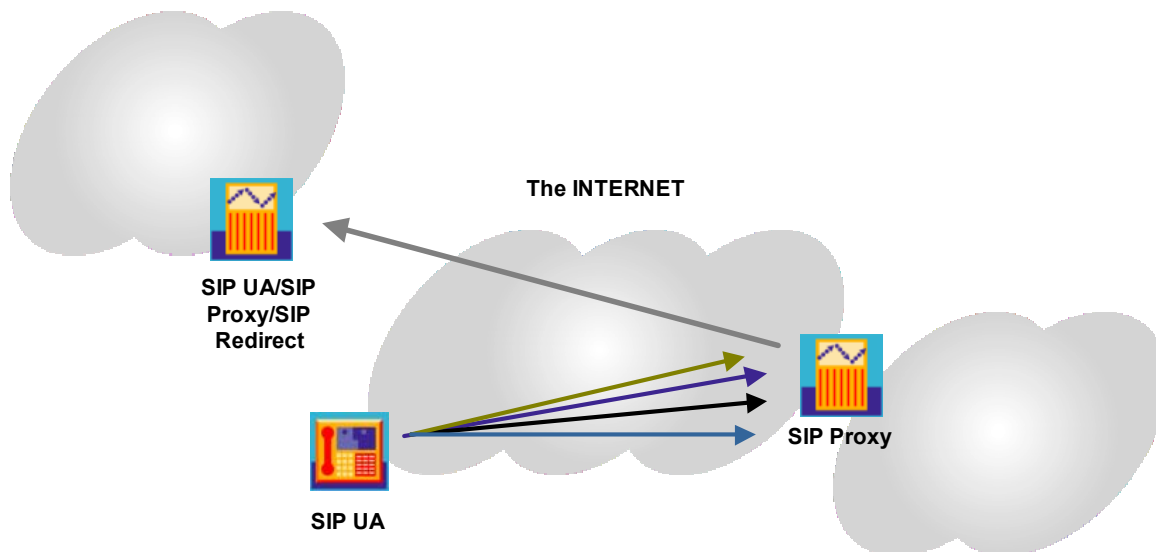
22.1.5 Denial of Service and Amplification

“Usually by directing an excessive amount of network traffic...”

Sometimes, and even in many cases, you do not need that excessive amount of traffic to create a denial of service condition.

The first method of denial of service described with this section talks about a case where an attacker spoof requests coming from the same host or several spoofed hosts through a certain route (VIA header). What will happen is when the response will be sent or a message will be forwarded through the route, that device will be flooded with a lot of messages and a denial of service condition will occur (dodgy, only if the SIP Proxy forks the requests it will actually produce more traffic. If the Proxy does not fork, then we might try the target directly. If we are not able or do not wish to send the flooding messages directly, then using a SIP Proxy might knock off the SIP Proxy itself...)

If the source for the spoofed messages is one source then it is more than easy to put a certain threshold within a SIP Proxy server saying for example that 20 simultaneous requests from the same initiator is our threshold to be met. If the threshold is met then further requests will be dropped. The threshold's aim is to eliminate the possibility of the usage of one spoofed source IP to flood purposes (or other weird things).



False INVITEs coming from the SAME SIP UA (Attacker) with spoofed source IP address and spoofed information in the correct places (such as the VIA header).

The SIP Proxy receives them all and forward them to the End-Point which in turn being flooded. The End device can be SIP UA/SIP Proxy/SIP Redirect/Any other device etc.

What will be the case if the same source uses multiple spoofed IP addresses in order to flood a SIP Proxy/UA/Redirect/Innocent victim?

Than this is a trickier situation, but again if we set a threshold on the number of requests going to a certain end point per second/minute we will be able to handle this. Again, our main problem is that the intelligence is at the end points. Hence, the IP Phone itself needs to generate a response that will suggest that it is engaged. The down side is if this is a denial of service aimed at a SIP

Proxy/Redirect which is on the path to a spoofed target (VIA header) a threshold is not a proper solution.

Some ideas:

- We can check the domain these requests are coming from and block any relay (like an SMTP server).
- We can limit the number of requests per route we make.
- We can limit the number of requests per second we wish to handle (for the same target; or from the same source)

I feel that more efforts with these scenarios need to be made. And I am willing to investigate this further.

Attacks utilizing Registrars

With some of the examples in this section I do not agree.

For example – “attackers could de-register some or all users in an administrative domain, thereby preventing these users from being invited to new sessions”.

But isn't this a fraud kind of a thing?

Are we going to have dumb devices that only perform their dumb duties but lack certain intelligence? If a registrar is not able to understand that a portion of its registered devices are registered with the same IP, this is bad...

If we talk about mobility than different [host@user](#) addresses has different IPs (and not the same IPs). Usually ALIASES will be made at a DNS level and not at the location service level if we need to associate different [user@host](#) descriptions with one another.

Call forwarding will be made using the end-devices and not using the location service... so I do not see any reason other than mistakes that the same IP address will be used for different users or phone numbers (especially in the same domain).

Another example: “attackers might also attempt to deplete available memory and disk resources of a registrar by registering huge numbers of binding”.

If I have registrar for the domain @STAKE for example, why should I allow registrations from different domains?

Suggestions:

- We can also limit the number of requests per second we process requests from one source
- We can limit the number of forked requests/responses by a forking SIP Proxy

22.2 Security Mechanisms

“...Proxy servers must therefore be trusted, to some degree, by SIP UAs. To this purpose, low-layer security mechanisms for SIP are recommended, which encrypt the entire SIP requests or responses on the wire on a hop-by-hop basis, and which allow end points to verify the identity of proxy servers to whom they send request”.

Two solutions are offered which are TLS and IPSEC.

With **IPSEC** we need to have the UA to use pre-sharing keys with the first hop proxy...
Do I need to state why this is bad?

TLS is using TCP. This will produce all sorts of problems when the transport protocol will be TCP and not UDP. This also means that if you wish to use UDP you will not have the ability to use encryption... I do not need to rewrite all of the problems regarding TCP we have with VoIP based protocols and architectures. They hold true for signaling protocols as well.

It seems to me that things were bent in order to have some sort of solution. What if this solution will introduce more problems than gain?

SIP was designed as a lightweight protocol not as a full blown heavy weight protocol...?

Usage of certificate – how are we going to validate the certificates?
Who is going to produce the certificates?

22.2.1 Transport and Network Layer Security

RFC Supported Ciphers:

- Must support TLS_RSA_WITH_AES_128_CBC_SHA
- Should support TLS_RSA_WITH_3DES_EDE_CBC_SHA (backward compatibility)
- Implementers MAY support any other cipher suite

Why giving the Server side implementing other cipher suites instead of requiring them to use AES 128bit encryption only? By giving the server side the ability to use weaker ciphers you are opening yourself to different sorts of cryptography based attacks (especially when some of the cipher text represents known text or it is VERY easy to understand what it will contain at least for the first few lines).

Do you remember who used 40bit DES encryption and how easy it was to break it?

The SIP Servers should be required to use a strong cipher and to enforce the client to use a strong cipher. Giving the choice to developers / companies regarding what to implement will not gain any good.

22.3.1 Requirements for Implementers of SIP

There are a lot of problems with this part. First only SIP Redirects, SIP Proxies and SIP Registrars **MUST** implement TLS and **MUST** support both mutual and one-way authentication.

Why it is only **RECOMMENDED** that a UA will be able to only initiate a TLS based connection?
And why is UAs are only **MAY** be capable of acting as TLS servers?
Should this be **REQUIERED** instead?

What the RFC suggests is that UAs will only verify certificates they receive from the different server entities... and this is up to the implementors to knock their heads on the wall and get with the implementation.

How this verification of certificates is going to take place? This is an important thing!
Is this going to be like the way it is implemented today with web browsers?

With S/MIME we have the problem of key distribution...

22.3.2 Security solutions

These are only suggestions nothing more...

It is understood that the Registrar provides his own certificate to the UA in order to validate himself for the UA. It is also understood that the UA is required to check that the domain within the certificate is the domain it wishes to register with...

1. UA establish TLS based communication with the Registrar
2. Registrar needs to send its certificate [UA needs to cut off communications if a certificate is not being provided or the certificate is revoked or the certificate cannot be validated or is not validated]
3. After the TLS tunnel is created (9 moves) the UA needs to create a REGISTER request to the Registrar.
4. The Registrar needs to ask for authentication from the UA. Therefore it is using the 401 or 407 responses for this. The RFC also specify that the REALM inside the Proxy-Authenticate needs to correspond to the domain given by the certificate (this is to validate the UA).
5. When receiving an Authentication request the UA can do two things:
 - a. Prompt the User for credentials, or
 - b. Take an appropriate credential from a keyring corresponding to the realm parameter in the challenge (Oh my god!)
6. Register resubmission with credentials
7. The TLS session is being maintained if the Registrar is also the SIP Proxy (preventing spoofed connections)

All we need is to have a correct username and password and a working UA and we are on. No one will require us to provide a certificate...

Keyring storing on the IP Phone is a bad thing to have. The IP Phone can be cloned and therefore another user within the domain can use the same credentials without even being prompted for username and password. Are we preaching for no real authentication here?

Where is the UA's certificate? Are we not missing a core thing here which will allow us to bind the UA to the Voice Network as well? Or does the problem of storing a certificate on the UA drives the non inclusion of this requirement within the RFC?

When a server will need to talk to the UA the Server will not be able to verify the UA's identify without getting a certificate (and extra authentication as well).

This is all good and well but it only covers the usage internally and not covers the problem with someone external to this domain spoofing requests. TLS sessions opened with the SIP Proxy server will put a burden on a very busy SIP Proxy. Oh well...

BUT

Why isn't it written that the registrar needs also to verify that the UA registers a host@domain which corresponds with the domain the registrar is representing?

Why we are not binding the UA to the network/domain?

Injecting public key false certificates to IP Phones...

Targeting the IP Phones and not the servers...

Requests and Transitive Trust

It is so nice that the connection from Biloxi.com to bob will be already opened...

What will happen if this connection was teared down somehow? This means that Bob will always be the initiation of the encrypted TLS tunnel. But what will happen if this tunnel was not recreated and not this INVITE needs to be delivered to bob?

Are we going to have here another denial-of-service?

G

Peer-to-Peer requests

You must be kidding me right?

There is no authentication... and this is wide open.

What about UA to UA? Have we forgotten something?

Other Interesting Points

- *All of this WILL NOT prevent a situation were one can mangles routing.* In such a situation having a pre-tailored SIP Proxy with a valid certificate will give an attacker the ability to perform a man-in-the-middle attack, and will open a venue for all the other attacks that are "prevented" if TLS is being used.
- *Time out problems with the TLS connection* (this will actually be interesting to watch).
- Inability to understand or control the voice channel after the creation. This means that if we shut down the RTP and RTCP channels there is no notification given to SIP.
- Bandwidth allocation (missing as well).

Impersonating to any SIP Entity

- SIP Proxy
- SIP Location Service
- SIP Registrar
- SIP UA

More ideas to follow