Security analysis and experiments for Voice over IP RTP media streams

Christian Wieser, Juha Röning
University of Oulu
Department of Electrical Engineering
Computer Engineering Laboratory, OUSPG
Linnanmaa PO BOX 4500
FIN-90014 University of Oulu, Finland
Email: ouspg@ee.oulu.fi

Ari Takanen
Codenomicon Ltd.
Kaitovayla 1
FI-90570 University of Oulu, Finland
Email: ari.takanen@codenomicon.com

***Abstract*— The Real-Time Transport Protocol (RTP), a transport protocol for real-time applications is the standard for transmitting encoded voice and video in IP telephony, including networks built using elements depending on Session Initiation Protocol (SIP) and H.323 protocol family. Different security requirements were analyzed, potential vulnerabilities were identified, and means of attacking them were built. As a case study, we assessed six implementations using the found attack methods trying to compromise the classical information security principles: confidentiality, integrity and availability. All of the implementations available for evaluation failed to perform in a secure manner under the test. We managed to eavesdrop the media stream and to inject a third party voice into an ongoing call. Finally, we successfully performed Denial-of-Service (DoS) attacks.**

I. Introduction

Voice over IP (VoIP), also known as IP telephony, has become widely deployed, threatening to replace the traditional telephony networks not only in the fixed-line environment but also in mobile networks. The VoIP technology and market develops rapidly, releasing new products to the market at high pace. Both enterprises and consumers are connecting their IP telephones to the public Internet.

From security perspective, this new transition of telephony to IP leads to new opportunities in improving the security of traditional telephony with security technologies widely used in other Internet protocols. But the openness of the Internet also brings to telephony the same threats that other open communication networks have had to endure. Real-Time Transport Protocol (RTP) and the RTP Control Protocol (RTCP) are examples of the open interfaces that can be used to attack VoIP systems.

The purpose of this study is to investigate potential RTP-related vulnerabilities in the VoIP implementations that enable the possibility of attacking the VoIP systems.

To keep this simple, we limited the study to the basic triangle of security requirements, or security principles, namely confidentiality (or secrecy), integrity and availability. The confidentiality requirement for us is that the data streams cannot be read in transit. The integrity requirement here is that messages cannot be altered in transit and third party modifications to the media streams are discovered and discarded. The availability requirement is that the offered service is reachable, reliabile and robust.

Our contribution in this paper is:

- We explain the necessary background to understand RTP security implications
- We study real-life threats and attacks against RTP implementations
- We describe the results using a case study approach to keep the findings realistic

The rest of the paper is organised as follows. In Section II we describe some of the available related work and further reading for security analysis of VoIP systems. The attacks studied here and their execution are described in Section III. An analysis of the test results after trying out the attack scenarios is given in Section IV. Section V discusses the work in a wider perspective and Section VI concludes.

II. Related Work

Industry standards exist with aims to prevent security problems in VoIP networks. One example is RFC3711 [1], which describes a profile to provide confidentiality, message authentication, and replay protection to RTP. Also implementation guidelines exist. NIST [2] gives a general introduction to security of IP telephony, and its security implications and recommendations. How these standards and guidelines are implemented still requires further study.

A generic overview of RTP security has been given by Ville Hallivuori [3] from Helsinki University of Technology. Several network analyzer tools such as Vomit [4] include the feature of converting the media streams seen in the network into audio files, provided that there is access to the media path. Although implementation level vulnerabilities related to RTP have not been publicly studied before this article, some test results are available for the accompanying control protocols, disclosed by the PROTOS researchers from University of Oulu [5]. These robustness tests or fuzzing tests were conducted for both SIP and H.323 protocols. Commercial tools are also nowadays available for finding these crash-level robustness flaws in SIP, H.323 and RTP/RTCP implementations.

III. Potential vulnerabilities

In IP telephony systems we can typically find different protocols for handling the setup and teardown of a connection (control protocol) and for the exchange of encoded voice and video data (media transport protocol). The most commonly used control protocols are the Session Initiation Protocol (SIP) [6] and the H.323 protocol family. Both standards use RTP to transmit the actual encoded voice. [7]

Although not required by the specification, a RTP stack is typically build up as follows: RTP uses the User Datagram Protocol (UDP) as the transport layer protocol, and the Internet Protocol (IP) acts as the networking protocol, which interfaces lower protocol layers.

RTP is described detailed in RFC3550 [8], but for the following discussion it might be worth to have a look at certain fields of the 12 byte long protocol header, presented in Figure 1:

> **Sequence number** - increases by one for each packet sent for loss detection and reordering.
> **Timestamp** - reflects the time of the first sample. This can be used to compensate for jitter.
> **SSRC** - acts as an identifier for the packet.

After the RTP header the encoded voice is added. RTP profiles [9] define the encoding, sampling rate, number of channels and the amount of samples per payload.

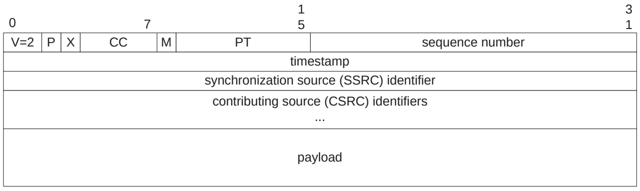
0
7
1
5
3
1
V=2
P
X
CC
M
PT
sequence number
timestamp
synchronization source (SSRC) identifier
contributing source (CSRC) identifiers
...
payload

Fig. 1. RTP packet, taken from RFC3550 [8]

The accompanying RTCP is outside the scope of this paper. It proved irrelevant for the attacks described here. RTCP-related attacks would require additional research.

A. The test bed

We chose the classical naming scheme to describe our attack as shown in Figure 2. Alice initiates a call to Bob, and Eve tries to interfere.

Alice and Bob use six different IP telephony endpoints: two H.323 endpoints and four SIP user agents - three commercial and three open-source implementations. Alice and Bob communicate without the usage of intermediate servers.

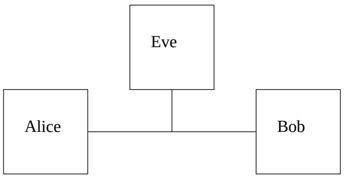

Fig. 2. Test bed

TABLE I
Used default codec

Implementation	Codec
001	G.723
002	G.711 PCMA
003	G.711 PCMU + comfort noise
004	G.711 PCMA
005	G.711 PCMU
006	G.711 PCMU

Eve uses an off-the-shelf PC running the Linux operating system. Eve had access to all information exchanged by Alice and Bob. She tries to penetrate the ongoing call by

1) eavesdropping it.
2) speaking into the ongoing call by adding her voice to.
3) injecting voice without eavesdropping.
4) starting a Denial-of-Service (DoS) attack.

In the description of attack results we have omitted the actual product names and replaced them with a three-digit number. These found issues are not specific to this random sample of six implementations only, and it would not be fair to publish the product names with the results. Also, we did not communicate our findings with the vendors and security coordinators, as is described in best-practise guidelines for handling multi-vendor vulnerability disclosures [10]. The reason were: we did not manage to package the test material easy deployable (for example, providing platform-independent link-layer network access); also, we tested only a small amount of implementations.

B. Security primitive: confidentiality

Due to the lack of secrecy mechanisms in the used VoIP endpoints, it is straight-forward for Eve to eavesdrop to the ongoing conversation using any VoIP enabled network analyzators. Table I shows the codecs used for encoding of the voice. Only one of the tested implementations supports encryption of the media stream, if manually enabled.

C. Security primitive: integrity

Eve tries to inject RTP data into the ongoing connection (also known as a replay-attack). She tries two attacks with samples of different duration (one and ten seconds), which are encoded using the same codec as the ongoing connection. Because each RTP packet includes an identifier, a sequence number and a timestamp, she adapts her data to be slightly ahead of the eavesdropped data (by one to ten packets).

Table II shows the results of the voice injection. We gave the verdict “good” when we could hear Eves voice in a quality that was comparable to the ongoing connection, “understandable” when we heard interferences and “poor” when the voice was heavily distorted. We could perform the attack successfully on all tested applications.

In the next step Eve simplifies her attack and does not adopt identifiers of the RTP header. This simplified version of the voice injection attack is still working with most implementations. In Table III we gave the verdict “disregard” when Eve

TABLE II

Comprehensibility of RTP voice injection by a third party

Implementation	1s sample	10s duration sample
001	good	good
002	understandable	understandable
003	poor	poor
004	good	good
005	understandable	understandable
006	good	good

TABLE III

Evaluation of the RTP data fields

Implementation	SSRC	Timestamp	Sequence number
001	disregard	partly	partly
002	disregard	disregard	disregard
003	evaluating	partly	partly
004	disregard	disregard	disregard
005	disregard	disregard	disregard
006	disregard	disregard	disregard

could set the data field to a random value with the attack is still working, “partly” when it had to be in the area of the expected value and “evaluating”, if it had to match.

One further thing hinders Eve: The transport layer dependence to find out the source/destination IP-address and the UDP source/destination port number. Note also that it could be possible to use the broadcast address as the destination address. By adapting her attack she finds out that some implementations do not evaluate transport/network layer information. Table IV shows whether the attack worked when Eve sends voice on the broadcast address or with incorrect source-IP address or source-UDP port. However, one value that has to be correct: the UDP-destination port. As Table V shows, applications use either fixed or easily guessable UDP port number values, further simplifying the attack.

D. Security primitive: availability

There are two common methods for attacking the availability. First method consists of flooding the end-point with traffic so that the valid trafic is disturbed or even rejected, effectively denying service. The second method consists of finding single invalid inputs that would crash the product, testing the robustness and reliability of the product.

TABLE IV

Transport layer evaluation

Implementation	Broadcast dst-IP address	Incorrect src-IP address	Incorrect source-UDP port
001	working	working	working
002	working	working	working
003	no	no	no
004	working	working	working
005	working	working	working
006	no	no	no

TABLE V

UDP destination port selection

Implementation	Start up	Next call
001	fixed (49608)	$newPort = oldPort - 2$
002	fixed (5004)	$fixed$
003	fixed (5000)	$newPort = oldPort + 2$
004	fixed (49152)	$newPort = oldPort + 2$
005	fixed (5000)	$newPort = oldPort + 4$
006	fixed (32782)	$fixed$

First the attacker (Eve) tries to flood one of the end-points (Bob) with arbitrary RTP packets to hinder or completely prevent the reception of Alice’s voice. This is successful against all implementations, jamming the reception enough that the valid voice is not understandable.

Using robustness testing (fuzzing) to discover crash-level flaws in the tested implementations was conducted using both internal tools and commercial tools. Some flaws were found, and this proves that there is still some quality assurance improvements needed in the product development practices.

IV. Analysis of the test results

All three primitives of information security are breached. We expected to be successful to a certain extend – due to the connection-less character of the transport layer protocol UDP – but were astonished that we could simplify the attacks considerably, requiring almost no knowledge of the system under attack. Although many of these vulnerabilities have been known for a long time, the VoIP products do not protect themselves from the attacks.

The tested RTP implementations did not provide confidentiality of the media stream by default or at all. Although specifications are available, encryption is still not widely adopted in VoIP products.

When audio stream was injected into the call by an attacker, all implementations decoded and played the attackers voice, effectively breaking the integrity criteria. Half of the tested implementations give the impression of not performing any checks on the RTP protocol fields before playing out the encoded voice. Four implementations did not perform any transport layer checks. Although the UDP ports for RTP can be chosen freely, implementations used fixed or easy guessable ports, further simplifying an attack.

Finally, we performed a Denial-of-Service attacks successfully. Disturbing of the connection by flooding RTP traffic was possible. Also invalid RTP packets could be used to find quality problems in the products.

V. Discussion

The described attacks have further potential when used to transmit unsolicited VoIP messages (i.e. Spam over Internet Telephony, or SPIT). A combination of easily guessable UDP-destination port and missing evaluation of RTP/UDP-packet header data could lead to a situation where the message – when sent to the broadcast address – is heard on all connected telephones.

The current security level of VoIP can be compared to that of IP-based network software five to ten years ago. For example, back then, it as accepted practice to use “telnet” to access servers. Today it is considered bad practice due to the lack of mechanisms to ensure confidentiality and integrity. Further, implementing network applications must be done with great care to not introduce vulnerabilities (which are nevertheless found on a daily basis), see e.g. [11]. These lessons learned from the past should be proactively applied to VoIP development and deployment.

We have deliberately not mentioned any of the proposed encryption methods, for two reasons. Firstly, only one of the implementations tested supported encryption, and secondly, we were looking at application layer attacks against RTP. A separate study would be required for methods providing security by lower layer protocols (IPsec) or within the media streams (SRTP).

We hope to contribute to a realistic risk assessment in VoIP deployment and wish that vendors adopt and improve their implementations.

VI. Summary

The tested RTP implementations provided no confidentiality of the media stream. All implementations were susceptible to integrity attacks, with the attacker requiring varying level of information of the ongoing call. Furthermore, Denial-of-Service attacks were performed successfully. We managed to penetrate the information security of the calls completely. Unprotected RTP media streams should be considered highly vulnerable.

References

[1] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, “The Secure Real-time Transport Protocol (SRTP),” RFC 3711, Mar. 2004.

[2] D. Kuhn, T. Walsh, and S. Fries, Eds., *Security Considerations for Voice Over IP Systems*. Gaithersburg, Maryland: NIST, Jan. 2005.

[3] V. Hallivuori, “Real-time transport protocol (rtp) security,” 2000. [Online]. Available: http://kotiweb.kotiportti.fi/vhallivu/files/rtp˙security.pdf

[4] (2004) vomit - voice over misconfigured internet telephones. [Online]. Available: http://vomit.xtdnet.nl/

[5] C. Wieser, M. Laakso, and H. Schulzrinne, “SIP robustness testing for large-scale use,” in *SOQUA/TECOS*, 2004, pp. 165–178.

[6] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP: Session Initiation Protocol,” RFC 3261, June 2002.

[7] H. Schulzrinne and J. Rosenberg, “Internet Telephony: architecture and protocols - an IETF perspective,” *Computer Networks (Amsterdam, Netherlands: 1999)*.

[8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 3550, July 2003.

[9] H. Schulzrinne and S. Casner, “RTP Profile for Audio and Video Conferences with Minimal Control,” RFC 3551, July 2003.

[10] M. Laakso, A. Takanen, and J. Röning, “Introducing constructive vulnerability disclosures,” in *Proc. 13th Annual FIRST Conference on Computer Security Incident Handling*, Toulouse, France, June 2001. [Online]. Available: http://www.ee.oulu.fi/research/ouspg/protos/sota/FIRST2001-disclosures/

[11] (2005) CERT coordination center. [Online]. Available: http://www.cert.org/