**Encoding RTP Traffic Using Audio Steganography for Covert Communication**

**Computer Networks Spring 2022**

**Randolph Bushman**

**Abstract:** This paper outlines a new method of covert communication that exfiltrates information from a victim host to a Command and Control (C2) Server in a fast and covert fashion. We take advantage of the large quantities of internet traffic generated by modern Voice Over IP (VoIP) applications. We use techniques in audio steganography to disguise the information in the packets transmitted from the victim host to a victim VoIP server to a C2 Server. We exploit the unencrypted Real-Time Transmission Protocol (RTP) traffic and encode packets to carry audio segments that sound unchanged to humans. When processed, each encoded packet carries one byte of data that we reconstruct on the C2 server. We implemented two demo executable files and performed experiments in virtual and physical environments. Initial impressions indicate that VoIP-based covert communication possesses the potential for fast and covert data exfiltration in VoIP networks.

**Introduction**

In computer networking, covert channels provide methods of data exchange in ways not intended with the aim to stay secret [1]. In cybersecurity, attackers use covert channels to exfiltrate information from victim hosts or networks to receiver C2 servers in a private, non-detectable manner. Attackers rely on covert communication to hide data exchanges from network analysts, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS). Due to the modern analytical techniques that they employ, we design our channel to prioritize stealth.

Time channels and storage channels represent the two classifications of covert channels. Time channels communicate with system resources based on timing, while storage channels write data to storage locations and allow those with lower clearance to obtain access [2]. For this paper, we shall design and implement a storage channel to alter network traffic to exfiltrate data from victim hosts.

Due to the recent need for applications like Zoom, Microsoft Teams, and Webex and the large volumes of internet packets generated by these applications, we designed our covert channel to exploit VoIP traffic. VoIP represents a phone technology that allows clients to make and receive calls via the internet instead of telephony infrastructure [3]. We employ simple audio steganography techniques to exploit this traffic to mask our information. As a result, the intercepted audio traffic sounds similar enough to the original traffic and raises slight suspicion to those in a VoIP session.

We organize this paper into sections titled Background, Covert Channel Design, Results and Discussion, and Conclusion. This paper assumes that attackers possess remote access to the target hosts and the necessary administrative execution privileges on the target hosts.

**Background**

# The Open Systems Interconnection Model

The Open Systems Interconnection (OSI) Model represents a model in computing that characterizes and standardizes the communication functions of computer systems. The OSI model consists of seven layers, each of which draws attention to covert channel designers. Common storage covert channels operate in the OSI model's application, transport, and network layers.

# VoIP Protocols

VoIP refers to a set of protocols operating in several layers in the OSI model. These include RTP, Real-time Transport Control Protocol (RTCP), Session Description Protocol (SDP), and Session Initiation Protocol (SIP) [4]. RTP constitutes the protocol that delivers audio and video over computer networks [5]. Applications use RTP to communicate stream media like telephony video teleconferences. Most RTP implementations rely on the User Datagram Protocol (UDP) to transport information. RTP also runs parallel with RTCP, which monitors transmission statistics that aid stream synchronization. It exists at both layers 5 and 6 of the OSI model.

VoIP servers exist on computer networks as intermediate, centralized locations where clients exchange session data. A client transmits data to the VoIP server and then processes and forwards that data to the other clients in the same session. Typical VoIP server applications allow for multiple clients in VoIP sessions. We show this below in **Figure 1**.

Diagram

Description automatically generated

Figure 1: A representation of a VoIP server and clients on a network where clients 1, 2, and 4 exist in one session, and clients 3, 5, and 6 exist in another at the same time.

To improve RTP communication, the designers introduced the concept of state in the RTP header. We use this to provide the reliability of the Transmission Control Protocol (TCP) with the speed of UDP. RTP implements this with the sequence number and timestamp fields. Applications initiate RTP sessions with SIP and SDP. SDP carries critical information like protocol version number, session name, and encryption keys [6].

**G.711 Audio Codec**

G.711 refers to the audio encoding that Asterisk employs by default in audio transmission in RTP. It represents a narrowband audio encoding designed for telephony [7]. G.711 transmits frequencies that range from 300Hz to 3,400HZ with a sample rate of 8,000 samples per second. We intercept the raw G.711 payload in RTP packets and replace the first byte with a byte from the message we wish to transmit. Due to the poor quality of G.711, the replacement of the bytes sounds unsuspicious.

**Audio Steganography**

Audio steganography refers to the obfuscation technique employed on audio samples that mask the existence of secret information [8]. In Figure 2, we show an audio sample encoded with images that the attacker converted to sound.

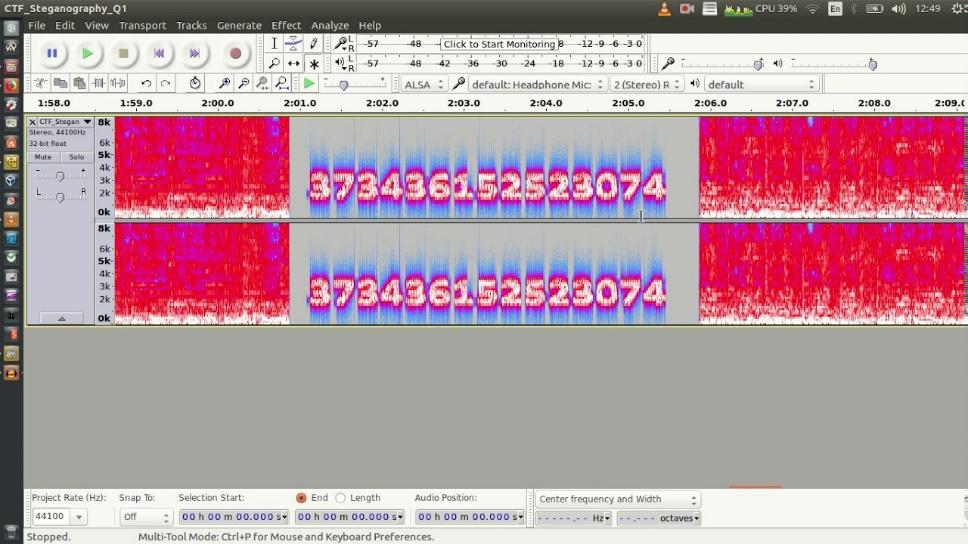


Figure 2: Hidden images placed in an audio sample spectrogram [9].

This paper demonstrates a simpler technique that maintains high performance and ease of implementation. We instead exchange the first byte of the transmitted audio within an intercepted RTP packet with a byte from the message we wish to transmit. While this introduces light static and popping sounds to the audio, we assume the session clients mistake this for poor audio quality. To further combat this problem, we also provide the attacker with the option to specify the number of packets to skip to reduce the frequency of static and pop sounds.

**Covert Channel Design**

**Tools and Techniques**

To implement our executables, we use the Python programming language and the Scapy library. While there exist faster languages, Python provides us with a simple foundation for our demos. We utilized Wireshark and Audacity to assist with the initial packet analysis as shown in **Figure 3**. We use Asterisk for our VoIP server software as it represents a common, unencrypted solution for Linux servers.

Graphical user interface, text

Description automatically generated

Figure 3: Analysis of a voicemail RTP stream in Wireshark.

Three possible executables exist for this RTP Covert Channel. We refer to them as the Victim Host Interceptor, Server Forwarder, and C2 Listener. The Victim Host Interceptor intercepts RTP packets from the session and encodes them with message segments. The modified packets then route to the VoIP server and the Server Forwarder. The Server Forwarder then recovers a copy of the packets and routes them to the C2 Listener. The C2 Listener decodes the packets and saves the information to the standard output console or a specified file location. We simplify our implementation with one-way communication to the C2 Listener. We show the general layout of the proposed network in **Figure 4.**

Diagram

Description automatically generated

Figure 4: The layout of a VoIP network with covert channel executables in Client 1, VoIP Server, and C2 Server. We encode the audio at Client 1 that forwards to the clients in the session. The other clients transmit normal, unaltered audio.

**RTP Traffic**

TCP refers to the common transport layer protocol used in the design of covert channels. Designers rely on TCP as it guarantees packet delivery and integrity. UDP represents a less common approach for covert channel implementation than TCP, as it ensures neither delivery nor integrity. While the speed of transmission increases, the reliability of UDP plummets.

RTP aims for the best of TCP and UDP with the introduction of state and RTCP. We use this protocol to carry control data that includes the number of bytes sent, packets lost, lost packets, and round-trip delay [10]. RTCP possesses five different modes that report statistical data. These include receiver report, sender report, source description items, end of participation, and application-specific functions [10]. This information helps the future design and implementation of this covert channel to improve reliability and integrity.

**Implementation**

The initial implementation for our covert channel consists of two executables instead of the proposed three. We treat the VoIP server as both a forwarder and a victim host at the same time to accomplish this. This eliminates the need for the difficult implementation of packet interception and simplifies our initial tests. We show the new proposed network diagram in **Figure 5**.

Diagram

Description automatically generated

Figure 5: The layout of our proposed solution for simpler implementation where the VoIP server acts as the victim host.

We include optional and required parameters for both executables. They allow the user to specify network address information, stealth values, and verbosity. We show the parameters of both executables below in Figure 6.

Text

Description automatically generated

Figure : The defined parameters for both executable files.

To simplify our implementation, we limit the traffic we forward to the C2 receiver to RTP traffic without the other VoIP protocols. These additional protocols shall assist in future implementations to carry packet state information for increased reliability and integrity. The addition of the other VoIP protocols also reduces the suspicion of network monitors as they disguise the C2 as a regular client. This assumes the network monitors limit their analysis to just packet capture logs. For our initial implementation, however, we shall ignore them.

**Results and Discussion**

**Perfect Conditions - Virtual Network Experiment**

We use a virtual network that consists of two virtual hosts for our initial tests. The first host acts as a regular client and the C2 receiver. The second acts as both an Asterisk server and victim host. This model serves as a compressed representation of a real VoIP network environment with multiple roles assigned to the hosts.

We run virtual network experiments to observe how well the channel operates in perfect conditions. We run version 16.16.1of Asterisk on the virtual server, while the clients run version 3.20.7 of MicroSip and version 2.10.17.3 of Zoiper5. As expected, the packets arrived with zero errors in packet arrival and integrity. We sent 10,000 packets that arrived in 337.1839s. This yields an upper bound transmission rate of 29.6574 packets per second. SSH, in comparison, took 0.0780s to download the same file. While SSH outperforms our covert channel in terms of speed, secrecy remains the top priority metric in our design.

**Normal Conditions - Physical Network Experiments**

We test our covert channel in a physical environment that consists of three separate hosts. The first host acts as a regular client. The second acts as another client and C2 server.The third acts as the VoIP server on a Raspberry Pi 4 with version 16.16.1of Asterisk. There exists one router on this network with wired speeds that average 750-940mbps download / 750-880mbps upload.

To test the reliability and integrity of our channel, we created a file that contains one of the same characters 10,000 times in a row. Next, we transmit the file contents to the forwarder on the VoIP server. We conducted five experiments with this file to test if the packets arrived on time, unaltered. To test the integrity of each packet, we compare each arrived character to the repeated ones in the test file. If a discrepancy exists, we conclude that the error comes from some packet corruption. For this experiment, we use the stealth value one and forward the data from one host in the session. We show the results in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment** | **Arrived Packets of 10,000** | **Number of Incorrect Packets** | **Transmission Time (s)** |
| 1 | 9,998 | 0 | 437.6420 |
| 2 | 9,995 | 0 | 438.4635 |
| 3 | 9,990 | 0 | 438.4051 |
| 4 | 9,993 | 0 | 438.9757 |
| 5 | 9,996 | 0 | 442.1016 |

The experiment above proves that RTP, without the use of the other VoIP protocols, transmits messages with an average error rate of less than 0.06%. The inexistence of incorrect packets likely comes from firewall filters that discard packets with an invalid checksum.

**Audio Analysis**

Due to the poor quality of G.711, the encoded audio from our channel sounds similar enough to the original that it removes suspicion from session hosts. Pop sounds from the audio remain a problem, however. We counter this with the introduction of the stealth parameter at the execution time. This value exists as an integer greater than zero and defaults to one. It represents the number of packets to skip in transmission before we encode the next. This shortens the frequency of pop sounds but slows the speed of communication. **Figure 7** shows the original audio from a 2.7825-second voicemail message from the Asterisk server.

Chart

Description automatically generated

Figure : Original audio signal from a voicemail recording.

We inject this audio with the message, “Hello, World! I am in the audio!”. We show the encoded signal in Figure 8.

Chart

Description automatically generated

Figure : Encoded audio signal from the same voicemail recording.

The encoded signal, while similar to the original, carries some discrepancies. These discrepancies, however, resemble the sounds of poor audio quality and create little suspicion. Suspicion may increase, however, if an analyst captures the original audio and compares it to the encoded signal. For example, we show the difference between the two signals below in Figure 9.

Chart

Description automatically generated

Figure : The difference between the original and encoded signals.

Once an analyst compares the signals, they may put together our pattern. As mentioned earlier, they employ a simple method of audio steganography. However, different techniques exist that return fewer visible patterns in the signals. This remains the largest issue with our channel that we may investigate in future work.

**Conclusion**

We conclude in this paper that covert communication with the use of VoIP protocols provides reliable data exfiltration in virtual network environments and accurate results for local physical networks. We also demonstrated that while our channel possesses a slow transmission speed even in perfect conditions, it provides a layer of obfuscation that introduces a safe level of suspicion in VoIP sessions. The large quantities of network traffic generated by VoIP server applications like Asterisk provide the perfect vector for exploitation.

Future work for this channel holds significant potential for research in covert communication. To address the major issue with the signal comparison, we ensure the victim host transmits the encoded audio alone. Future versions of our channel may include the implementation of robust packet interception techniques to compensate.

Implementation of two-way communication between VoIP and C2 also reduces suspicions as this represents normal behavior in a VoIP session. We also must implement the transmission of the other VoIP protocols in the session to carry session statistics if errors occur. This introduces the ability to recover lost information from the server and increases the reliability and integrity of our channel.

We shall also investigate robust steganography methods to carry additional information with less error. One implementation involves the attacker encoding the least significant bits of certain parts of the RTP payload. For example, if each RTP packet carries 172 bytes of data and encodes each byte at the least significant bit, we increase our transmission speed from 1 byte per packet to 21.5. Since we encode the least significant bits alone, this may also reduce the pops heard in the audio.

**References**

|  |  |
| --- | --- |
| [1] | "Covert Channels," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Covert\_channel. [Accessed 2022 February 2022]. |
| [2] | "Covert Channels Transcription," SKILLSET, [Online]. Available: https://www.skillset.com/certifications/topic/Covert%20Channels#:~:text=There%20are%20two%20different%20types,resources%20based%20on%20different%20timing. [Accessed 8 February 2022]. |
| [3] | "Beginner's Guide: What is VoIP and How Does it Work?," wpbeginner, 1 January 2022. [Online]. Available: https://www.wpbeginner.com/beginners-guide/beginners-guide-what-is-voip-and-how-does-it-work-explained/. [Accessed 25 March 2022]. |
| [4] | "Voice Over IP," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Voice\_over\_IP. [Accessed 22 March 2022]. |
| [5] | "Real-Time Transport Protocol," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Real-time\_Transport\_Protocol. [Accessed 22 March 22]. |
| [6] | "Sessison Description Protocol," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Session\_Description\_Protocol. [Accessed 22 March 2022]. |
| [7] | "G.711," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/G.711. [Accessed 22 April 2022]. |
| [8] | "Audio Steganography with Embedded Text," IOP Science, 6 May 2017. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/226/1/012084#:~:text=Audio%20steganography%20is%20about%20hiding,if%20the%20message%20is%20encrypted. [Accessed 22 March 2022]. |
| [9] | "HOW TO HIDE SECRET MESSAGES IN MUSIC FILES?," IICybersecurity, [Online]. Available: https://www.iicybersecurity.com/audio-steganography.html. [Accessed 25 March 2022]. |
| [10] | "Real-Time Transport Control Protocol," TechTarget, [Online]. Available: https://www.techtarget.com/searchunifiedcommunications/definition/Real-Time-Transport-Control-Protocol-RTCP#:~:text=RTCP%20is%20used%20in%20voice,round%20trip%20delay%20between%20endpoints.. [Accessed 3 April 2022]. |