VoIP Network Covert Channels to Enhance Privacy and Information Sharing

Jens Saenger[2], Wojciech Mazurczyk[1], Jörg Keller[2], Luca Caviglione[3]

Abstract

Information hiding is increasingly used to implement covert channels, to exfiltrate data or to perform attacks in a stealthy manner. Another important usage deals with privacy, for instance, to bypass limitations imposed by a regime, to prevent censorship or to share information in sensitive scenarios such as those dealing with cyber defense. In this perspective, the paper investigates how VoIP communications can be used as a methodology to enhance privacy. Specifically, we propose to hide traffic into VoIP conversations in order to prevent the disclosure, exposure and revelation to an attacker or blocking the ongoing exchange of information. To this aim, we exploit the voice activity detection feature available in many client interfaces to produce fake silence packets, which can be used as the carrier where to hide data. Results indicate that the proposed approach can be suitable to enforce the privacy in real use cases, especially for file transfers. As interactive services (e.g., web browsing) may experience too many delays due to the limited bandwidth, some form of optimization or content scaling may be advisable for such scenarios.

Keywords: information hiding, network covert channel, VoIP, IP telephony, security & privacy.

1. Introduction

Enforcing the privacy of end users and organizations requires to hide information from an external observer, which can be a censor, a firewall deployed by a regime as well as an attacker trying to collect data for social engineering, or prevent collaborative cyber defense campaigns. This usually entails the adoption of techniques to guarantee secrecy, anonymity and non-interference [1]. Even if information hiding approaches, including covert channels, have been primarily adopted by malware to implement anti-forensics mechanisms or abusive communication services, they are increasingly used to engineer file systems able to preserve the privacy, bypass limitations imposed by a regime, prevent censorship or protect sources in journalism [2].

In general, Privacy Enhancing Technology (PET) is an umbrella definition enclosing a variety of tools. Encryption is one of the most popular mechanisms and it assures protection by making the original data unreadable without the proper key. Modern encryption techniques offer a very high degree of protection but do not prevent to identify that two parties are communicating, which could be a valuable information for a censor or an attacker [3]. In this perspective, network covert channels are crucial not only to protect data, but also to hide that a conversation had taken place [4]. Applying some form of information hiding to a traffic flow can be used to create a covert communication path by injecting secret data into different features of the network traffic. Possible carriers are unused fields in the header of a packet or statistics characterizing the flow, such as the throughput, the packet error rate, and the jitter [5, 6]. Recent works also investigate methods targeting emerging network scenarios, for instance those embedding data within IPv6 traffic or in IPv4/IPv6 transitional mechanisms [7].

To remain unnoticed, the overt traffic flow, i.e., the traffic in which the covert message is hidden, should not appear as an anomaly. Thus, its popularity plays a major role [5, 8] and Voice over IP (VoIP) applications are an excellent choice, as they are ubiquitously available, produce vast volumes of data, and the conversation is intrinsically bidirectional and balanced in terms of traffic volumes [9]. To create a covert channel within a VoIP flow, we present a method taking advantage and largely extending the information hiding technique originally presented in [10, 11]. In essence, it exploits the Voice Activity Detection (VAD) feature used to save bandwidth by suspending the transmission of packets during speech pauses. Secrets are then injected into fake Real-time Transport Protocol (RTP) packets generated during silence intervals acting as the carrier. The feasibility of using information hiding techniques for PET purposes has been already partially investigated (see, e.g., [12]). Unfortunately, the "arm race" between cyber criminals and security experts imposes to consider new approaches for supporting collaborative cyber defense campaigns. In this vein, novel and advanced techniques to enforce privacy are an important building block, especially to mitigate the impact of counterattacks [13] as well as to prevent the poisoning of data exchanged by Law Enforcement Agencies (LEAs) or generated by distributed security frameworks [14]. However, at the best of our knowledge, this is the first work evaluating the performances of network covert channels to enforce the privacy of users exchanging data through the Internet.

Compared to our previous works [10, 11], in this paper we introduce a framework to implement a steganographic communication service exposed via a generic, virtual network inter-

[1]Warsaw University of Technology.
[2]FernUniversität in Hagen.
[3]Institute for Applied Mathematics and Information Technologies.
Corresponding Author: luca.caviglione@ge.imati.cnr.it

face, which can be used to tunnel other protocols. To effectively implement a PET and support collaborative cyber defense, hidden channels should deliver a bi-directional communication path. Thus, differently from [10, 11], we extend the original mechanism to support full-duplex communications. Another important difference with [10, 11] concerns the analysis of the impact of embedding data in silence/talkspurt patterns in terms of delay and resource availability. In this paper we also assess collaborative cyber defense, which is a totally different scope compared to our previous works.

Summing up, the main contributions of this paper are: *i*) the design of a PET-capable framework leveraging VoIP traffic by providing a virtual network interface for tunneling protocols of the TCP/IP suite, *ii*) the extension of the original injection technique presented in [10, 11] to support full-duplex paths often used for Machine-to-Machine (M2M) communications and collaborative defense frameworks, *iii*) an assessment of the performances experienced by the different functional layers using the covert channel, and *iv*) an increased understanding of security requirements of collaborative cyber defense efforts, which still lack of successful models [13].

The remainder of the paper is structured as follows. Section 2 deals with the related work, while Section 3 discusses VoIP basics and its use for information hiding purposes. Section 4 introduces the threat model and the design and implementation choices of our approach. Section 5 showcases the performance evaluation and Section 6 assess the quality experienced by endpoints secretly exchanging data. Lastly, Section 7 concludes the work.

2. Related Work

In essence, the goal of privacy enhancing methodologies is to protect individuals by using techniques to enforce anonymity, pseudo-anonymity, unlinkability, and unobservability of their data [12, 15]. This can be done by acting on different layers, for instance, on network traffic, user identity, data storage as well as by deploying mechanisms to encrypt and separate the information [12, 16]. Even if the literature lacks of PET technologies specifically tailored for sharing and collaborating for cyber defense purposes, it offers many general solutions [12, 15, 16] targeting different parts of the protocol stack. However, in this work, we concentrate on PET methodologies for protecting endpoints communicating through the network.

In general, the most notable example is Tor [16], which widely demonstrated its effectiveness to enhance the privacy of users. That is why some states are developing countermeasures to prevent its usage. For instance, Chinese Tor users are blocked by the Great Firewall that drops the SYN/ACK sent by a Tor bridge in response to a connection request from a client [17]. Therefore, obfuscating the usage of Tor through a proper Pluggable Transport (PT) is of paramount importance [17]. A possible idea is proposed in [18], where Tor traffic is morphed to resemble a Skype video call for bypassing blocks or firewall rules. However, as discussed in [19], trying to achieve unobservability by mimicking another protocol is substantially a flawed approach as the censor only needs to find few discrepancies, which are very difficult to avoid. Moreover, the wide amount of different implementations accounts for a multitude of bugs, quirks and peculiar behaviors that have to be replicated to make a convincing duplicate of a specific protocol. Consequently, our approach of embedding data in a real network flow generated via a publicly available client interface aims at preventing such issues and increases the chances that the secret conversation will remain unspotted.

Concerning the use of VoIP for building covert channels, the literature showcases several techniques. As possible examples, [20] and [21] survey data hiding methods targeting IP telephony in a wider sense, e.g., algorithms for embedding secrets in signaling or metadata. Instead, [6] and [22] focus on techniques to inject information in audio/video streams. Analogously, [23] and [24] demonstrate how different fields of the Real-time Transport Control Protocol (RTCP) can be manipulated to encode arbitrary information. A similar approach is presented in [25], which shows how to inject secrets by adjusting the timestamps used for synchronizing the streams. As regards works targeting the voice, in [26] authors compare four different hiding techniques, such as low-bit coding, spread spectrum, echo data hiding and phase coding. In [27] the voice data of two unaware parties is compressed to make room for secrets, whereas in [28] a bidirectional covert channel is set up by using the least significant bit of frames generated by the G.711 codec. Similarly, the work of [29] exploits unused bits of G.723.1 frames. A different approach is proposed in [30], where the F0 parameter used by Speex to control the pitch of the vocal signal is altered to free bits for secret data. Lastly, [31] and [32] discuss more specific techniques as they propose methodologies that are built for Skype. The former manipulates the inter-packet delay for selected protocol data units, whereas the latter embeds secrets by replacing the payload of packets containing silence.

In contrast to the works above, the framework that is proposed in this paper leaves the voice information unaltered but adds fake RTP packets during periods of silence to transport the secret data. Also, it not only implements a network covert channel in VoIP communications but extends this approach to provide a virtual network interface through which other protocols can be tunneled. This path can then be used to orchestrate devices implementing firewalls or probes, as well as nodes collecting data to feed machine-learning-capable detection tools.

We point out that covert channels have been previously used both to improve privacy [33] or as a threat to privacy [34]. Moreover, a recent trend deals with the creation of air-gapped covert channels allowing to secretly exchange data between two endpoints without network connectivity (see, e.g., [35] or [36] and [37]). However, to the best of our knowledge, there is not any previous study presenting a privacy-enhancing network interface that uses a network covert channel especially when information has to be shared to counteract or prevent attacks.

As regards countermeasures, literature proposes two different approaches: detect and block a covert channel in a specific communication, or modify all communications to disrupt the potential covert channel or at least restrict its bandwidth

[21, 38]. Unfortunately, both approaches are tightly coupled with the used injection mechanism. For instance, removing data hidden in timing statics of traffic could require to act at a packet level, whereas sanitizing an HTTP conversation may lead to inspecting protocol internals or statistical indicators describing the tags composing the hypertext [6]. Moreover, developing a middlebox for blocking a covert channel is not a simple task. In fact, inspecting the traffic may lead to scalability issues and "sanitizing" all the flows (e.g., by buffering mechanisms or by overwriting ambiguous/unused fields of a specific protocol) which means penalizing also clean flows. For the specific case of VoIP communications, the main approach aims at finding signatures or distortions in the VoIP stream (see, e.g., [39] for a technique using the Mel-cepstrum technique). However, as it will be detailed in Section 6, our approach does not lead to appreciable distortions on the audio quality.

3. VoIP fundamentals and its data hiding potential

With the VoIP hypernym, we refer to the family of services enabling voice conversations through the Internet. The advantages of using VoIP traffic for embedding the secret information are: it is ubiquitously available and its presence is not perceived as an anomaly; it is widely used leading to a volume of data difficult to block, inspect or manipulate; it requires real-time or quasi real-time constraints, thus invasive monitoring methodologies, such as deep packet inspection could be inapplicable or easily detected.

3.1. VoIP Basics

In essence, VoIP enables remote peers, each one implementing a User Agent (UA), to communicate by converting the voice from analog to digital samples delivered through a suitable media protocol. A popular solution to manage the call is the Session Initiation Protocol (SIP), which offers various methods mimicking operations performed in legacy telephony (e.g., ringing) as well as parameters for initializing the media stream [40]. To transfer voice data, the UA typically uses two protocols. The RTP actually transports the voice data through the network, while the RTCP provides information for synchronizing the streams. Both flows are transported via UDP [10].

For the purpose of creating a covert channel, in this paper, we exploit the VAD optimization available in many UAs [10, 11]. Put briefly, VAD allows the sender UA to stop the transmission during speech pauses of the talker [41]. This can lead to relevant bandwidth savings, as typical VoIP conversations are characterized by silence periods ranging from 35% to 70% [42, 43]. Figure 1 depicts the RTP streams generated by the UA with or without VAD.

Implementing an efficient VAD strategy could be challenging, as it should not reduce the quality of the conversation by causing additional distortions or delays [41]. In fact, aggressive interruptions of the voice flow can lead to cropping of speech samples (and result in quality issues) and a total lack of noise can confuse the talker, i.e., it gives the idea that the conversation has been finished. In this case, ad-hoc synthetic comfort noise should be transmitted or generated locally by the receiving UA [44].

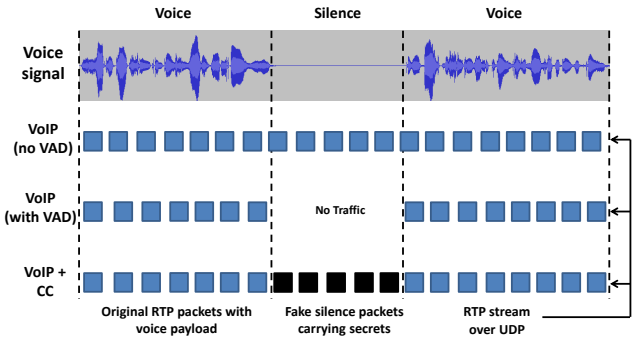

Figure 1: The different RTP traffic streams used for creating the covert channel.

3.2. Covert Channel with Voice Activity Detection

As previously mentioned, the method originally proposed in [10, 11] creates fake silence packets when the transmission of RTP data is stopped by the VAD. The resulting overt stream is composed of the VoIP traffic plus the covert channel, which is embedded in fake silence packets. In other words, this method hides information to a third party controlling the network, by transforming a VAD-enabled VoIP conversation into a non-VAD one, as depicted in Figure 1. During this process, the resulting overt RTP flow should not be disrupted and the quality of the conversation should remain unaltered. In general, fluctuations in the voice (including additional noise) are due to the impact of the network, for instance in terms of lost packets or late delivery of samples, which are discarded by the codec due to the expiration of suitable timeouts. However, deviating from an expected outcome can trigger some additional investigations (see, e.g., [19] and the references therein) and an external observer could reveal the presence of the hidden information. Consequently, the injection/extraction of secrets is crucial as well as creating fake silence packets in adherence to a recognizable, well-known template [10, 11]. Therefore, the ultimate goal of the used method is to preserve all the features of the entire VoIP conversation, even by paying something in terms of throughput of the secret data.

To better understand the performance of the proposed PET-enabled framework, we briefly summarize how the data is injected and extracted from the VoIP flow. Preliminarily, the information to be transmitted through the covert channel is dynamically subdivided into chunks to fit into Protocol Data Units (PDUs) generated by the RTP layer. The fake silence packets embedding data are identified at the receiving side by using a 4-byte long hash placed in the payload as a preamble. To avoid the disruption of the RTP stream and the VoIP conversation, the secret sender spoofs the `Synchronization Source Identifier` (SSRC ID) and the `Payload Type` (PT) fields. It also adjusts the values of the `Timestamp` and the `Sequence Number`. To not reveal the presence of VAD, the mark bit `M` is set to zero. The secret receiver probes every incoming PDU for the hash. If found, the secret is extracted and the RTP packet

is dropped. For regular packets, altered RTP fields are changed before they are delivered to the UA of the callee. Indeed, alterations of the stream could be revealed by inspecting counters and statistics provided by the RTCP. Hence, the method used to create the covert channel intercepts RTCP messages and spoofs Sender Reports to adjust the `Packet Count`, `Octet Count`, as well as the current `Timestamp`. This allows to compensate alterations in terms of volumes and delays due to the additional non-VAD RTP packets.

To guarantee the stealthiness of the covert channel, the information embedded in fake silence PDUs should not represent an anomaly. To this aim, the used VAD injection technique can infer the speech codec used by means of a heuristics considering timestamps, data patterns, and the delay and jitter of packets. Then, fake silence packets not carrying secrets are filled with suitable, comfort noise patterns (e.g., samples encoded with G722.1) [10].

The bandwidth of a covert channel implemented taking advantage of talkspurts characterizing VoIP conversations mainly depends on the number of silence packets filled with secrets. Even if a detailed discussion on tuning such parameter is outside the scope of this work, we point out that the higher the number of silence packets filled with secrets, the larger the amount of covert information that can be sent per time unit. However, this would lead to a non-negligible alteration of the spectrum of silence/comfort noise. In addition, the "aggressiveness" in stopping the transmission of voice data (e.g., due to reduced overhangs) may lead to additional disruption in the conversation quality, hence reducing the stealthiness of the covert channel [10]. According to [45], such parameters are governed by a "magic triangle" rule, i.e., increasing the bandwidth of the covert channel reduces its stealthiness and vice versa.

4. Threat Model, Design and Implementation

In this section, we first describe the considered threat model followed by the design and implementation details.

4.1. Threat Model

To design the proposed PET-capable framework, we assume a threat model where two peers want to exchange information to establish a collaborative cyber defense efforts. As the attacker, we consider a malicious entity able to control the network infrastructure as well as to capture, drop or modify communications of the users, for instance by using suitable traffic policies, filtering or firewall rules or heuristics. Even if many attacks are launched by state-wide or well-financed teams, we consider as a worst case, an attacker able to perform relevant damages even in the presence of modest equipment. Thus, to consider a realistic case, the attacker has only limited computational power and/or storage resources. Hence, he/she may possess capabilities like deep-packet inspection, but they could be applied to network traffic at a "full" wire speed only for short observation windows. Also, the captured network traces are very space-consuming, thus they cannot be stored for long periods and sophisticated traffic analysis could be unfeasible with cost-effective hardware

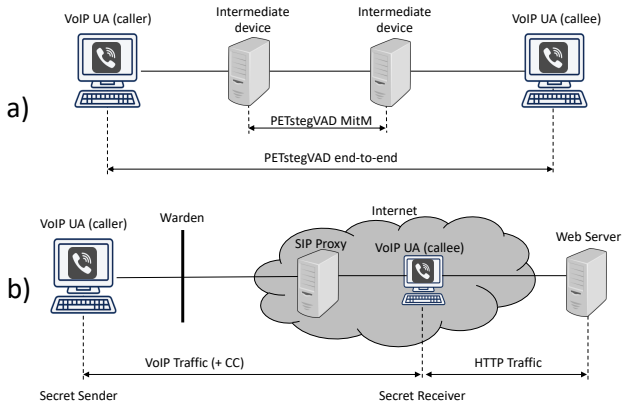

Figure 2: Reference threat model considered for creating a covert channel with PETVAD and a typical use case.

or without paying relevant monetary costs. Lastly, we assume that the attacker has no interest in blocking access of legitimate users to standard Internet services like VoIP communications. As discussed in [2, 19], such assumptions allow to model two different kinds of privacy threats: *i*) a state-level oblivious adversary acting as a censor [2] and *ii*) an attacker wanting to block alarms, poison samples to be used for training AI-based countermeasures or launch a DoS against collaborative security frameworks [14].

In the following we refer to the proposed framework as PETVAD. Moreover, with the *secret sender* and the *secret receiver* we denote endpoints wanting to exchange data through the covert channel. For instance, they can be a client wanting to access a remote server blocked by a firewall, or a source needing to send a picture to a journalist via an insecure network path. According to the literature on information hiding, a *warden* is an external observer able to inspect the overt traffic flow for discovering and blocking the hidden data exchange [5, 6].

Figure 2a depicts the reference threat model, originally introduced in [10]. We point out that the warden can be both a censor blocking communication between two individuals or an attacker wanting to gather the information exchanged between entities cooperating for security-related duties. Despite the scenario, different covert channels can be built by using PETVAD. Specifically:

- End-to-End (E2E): the secret sender and the secret receiver are co-located within the hosts running the UA of the caller and the callee, respectively. The overt network traffic is the RTP stream carrying the VoIP conversation, and the covert data is embedded in the flow of fake silence packets.
- Man-in-the-Middle (MitM): the VoIP conversation between two UAs is hijacked by intermediate devices, acting as the secret sender and secret receiver, respectively. In this case, the covert channel has a limited scope, and the remainder of the overt traffic does not contain any hidden information.

Figure 2b depicts a typical use case for PETVAD, which was not considered in our previous works [10, 11]. Let us consider a user (the secret sender) wanting to access a blocked or tracked website (the secret receiver). Hence, PETVAD can be used to "tunnel" the HTTP traffic within a VoIP conversation (the overt channel) as to impede the warden to spot or block the communication. Concerning the protocol architecture, if PETVAD is not used on an end-to-end basis, the intermediate node hosting the UA of the callee should also act as a sort of "information hiding proxy" by properly encapsulating the HTTP communication or by retrieving the webpage. The VoIP traffic containing the covert channel can be generated by PETVAD via artificial/synthetical VoIP conversations, thus the VAD/non-VAD transformation and the embedding of secrets could be merged. Alternatively, a VAD-enabled VoIP stream between two unaware UAs has to be hijacked.

4.2. *Implementation Details*

Figure 3 depicts the reference software architecture of the proposed framework. In essence, it is composed of:

- Secret sender and secret receiver: they are the processes needing to communicate through the PET-enhanced, end-to-end covert channel. From the viewpoint of the protocol architecture, PETVAD is made available as a network interface, thus, both the secret sender and receiver are reduced to the high layers of the TCP/IP stack.
- PETVAD: it implements the bidirectional covert channel by wrapping two channels generated via the injection of fake VAD packets. From the viewpoint of a warden, the two PETVAD endpoints appear as the VoIP UA of the caller and the callee, respectively.
- VoIP conversation: it represents the overt traffic sent on the network. It can be implemented via two VoIP UAs, which can be part of the framework (e.g., instrumented client interfaces or software libraries) or unaware endpoints.

The secret endpoints and the PETVAD layer communicate through a TUN interface. In general, RTP packets of the VoIP UA are smaller than the PDUs produced by the protocol stack of the secret sender or receiver. For instance, the typical maximum transfer unit of the TUN interface is 1500 bytes, whereas an RTP packet generated with the G.711 codec has only 160 bytes available for the payload. As a consequence, packets flowing through the TUN have to be properly buffered and fragmented. Buffering is crucial since transmissions can be done only in silence periods, thus the bandwidth of the covert channel during talkspurt periods is zero.

The mechanism implemented in [10, 11] has to be adapted to properly work with a SIP proxy, as depicted in Figure 2. For instance, in our trials we had a situation where a SIP proxy accounted for the creation of two independent connections (from/to the caller and the callee, respectively), thus it acted as an RTP relay. This may cause a "misalignment" in the hashes used to recognize fake RTP packets embedding secrets. In fact, the

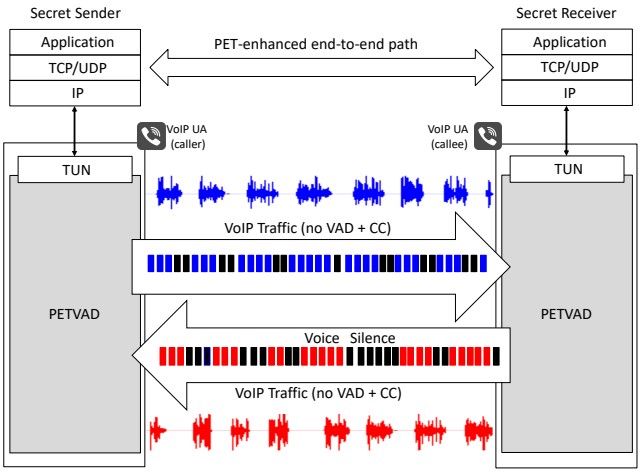

Figure 3: Reference architecture of PETVAD when implementing a privacy-enhanced bidirectional path. Blue, red: packets carrying voice samples. Black: packets carrying silence.

original implementation generates the hash by using a predefined secret and the `Sequence Number` value from the RTP header [10]. However, as mentioned, the SIP proxy breaks the end-to-end semantic, hence each flow has its own `Sequence Number` resulting into different hashes. Obviously, it could be possible to compute the hash by only using the pre-shared secret. Unfortunately, this may significantly reduce the stealthiness of the channel due to a bias in the entropy of the flow (the first four bytes of the payload of modified packets will have exactly the same value). A possible alternative considers other fields available in the protocols of the TCP/IP stack, e.g., the `ID` field within the IPv4 header or the `Timestamp` within the RTP. It is possible that the aforementioned fields would be "normalized" by the RTP relay, too. In this case, as a workaround, the secret sender and secret receiver may set a counter for each VoIP flow and assign it a value of the pre-shared secret when the call begins. Then, at each transmitted RTP packet, the counter is incremented by one and the hash is calculated according to the predefined secret and the current value of the counter. Such an approach randomizes the first four bytes of the payload of each covert packet thus making the hidden communication harder to be revealed even if the packets still can be successfully identified at the receiver side.

As regards the implementation of the proposed approach to enforce privacy requirements when sharing the information, the software layers of PETVAD have been written in C and by borrowing some code from previous works [10, 28]. The code has been optimized to limit overheads and delays to capture, spoof and inject traffic as to prevent the disruption of the conversation. To intercept and capture SIP/RTP/RTCP packets, we generated programmatically `iptables` rules. The UAs have been implemented by using the Pjsua[4] client interface. It has been selected since it uses the VAD engine of Speex and implements constant silence patterns for many codecs, including G.711(a), G.711(μ),

[4]http://www.pjsip.org. Last accessed December 2019.

Table 1: Download times and data rates achieved with `scp` using different communication paths.

File Size		10 kB		100 kB		1 MB	
		time [s]	rate [kB/s]	time [s]	rate [kB/s]	time [s]	rate [kB/s]
direct	mean	0.59	17.50	0.55	186.11	0.64	1,645.60
	std. dev.	0.07	1.98	0.02	6.42	0.02	47.77
PETVAD	mean	11.78	0.937	43.16	2.52	424.01	2.48
	std. dev.	3.11	0.30	10.73	0.72	26.54	0.17

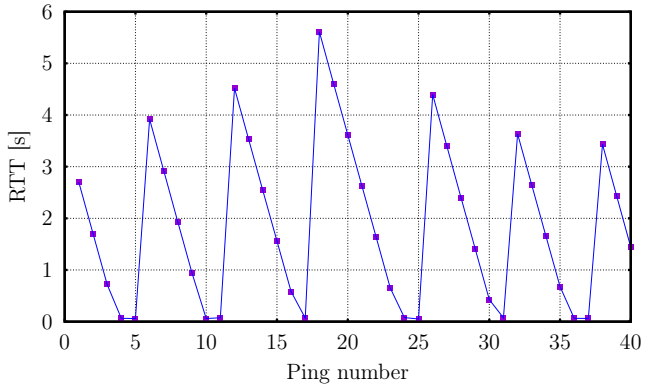

Figure 4: Behavior of the RTT characterizing a PETVAD covert channel.

GSM, iLBC, and Speex. The silence RTP packets are generated with the smallest possible payload [46] and silence descriptors are sent using Speex in the VAD mode. The Pjsua clients have been integrated into the overall framework via ad-hoc scripts.

5. Performance Evaluation

To evaluate the behavior of PETVAD in realistic setups, we performed a thorough measurement campaign. Two network configurations have been considered: *i)* a LAN testbed to assess the impact of network impairments and *ii)* a WAN scenario with a DSL line to evaluate the uncontrollable behavior of the Internet. To collect the produced traffic, we used a Netgear ProSafe GS105E switch with the traffic mirrored towards an ad-hoc machine running `tcpdump` and `Wireshark`. The two secret endpoints have been implemented on Linux Ubuntu 16.04 running on Intel Core i5-3230 and AMD E2-1800 hosts, both with 8 GB of RAM. Each trial has been repeated 10 times as to have the proper statistical relevance. Numerical results have been obtained with Matlab R2017b and bash scripts.

The VoIP stream has been produced with the G.711(μ) codec leading to an exploitable payload for silence RTP packets of 156 bytes (note that it is because the 4 bytes of 160 bytes payload is used to transfer the identification hash), and the entire silence periods have been used to embed the secret information. This may cause a reduced stealthiness but allows to consider a worst case scenario. To implement a realistic talker, we used traces from the TIMIT speech corpus [47] and we prepared representative VoIP conversations for both male and female english individuals. To assess performances in fair conditions, we considered conversations with talk periods interleaved with silence according to a 50:50 ratio, as it happens in real-world cases [10, 11, 42]. In the rest of the paper, with *direct* we denote users accessing the network without using PETVAD. We point out that investigating the sensitivity of the method proposed in [10, 11] as a "standalone" covert channel is outside of the scope of this paper (see [10] for a thorough evaluation). Rather, we focus on its performances when used as a PET for key network services.

5.1. Network-level Analysis

As any other communication path, a covert channel can be also considered as a network pipe characterized by bandwidth and delay [48]. For the case of PETVAD, both parameters tightly depend on the used codec and the "duty cycle" of talk-silence periods as they determine the available payload and the number of silence RTP packets. As a consequence, characterizing PETVAD in terms of Round Trip Time (RTT) allows to understand how the higher layers of the secret endpoints behave, especially when connection-oriented protocols are used.

Figure 4 depicts the RTT experienced by the secret sender to `ping` the secret receiver through a PETVAD channel. As shown, the RTT exhibits a "bursty" behavior. In fact, when the talker is active, the covert channel is unavailable since PETVAD cannot generate fake silence RTP packets. Therefore, the data produced by the secret sender has to be buffered until the talkspurt stops and data can be injected again. As regards the shape of the RTT, ping packets are generated with a default interval of 1 s and can be sent during silence periods according to the rate of RTP packets (i.e., 20 ms in our setup). Concerning the values of RTT, they are tightly coupled with the behavior of the talker. In our case, the talk periods of the created TIMIT-based VoIP conversations are in the range of 3 to 5 seconds. Obviously, the higher the variability in the behavior of the talker, the more the jitter of the covert channel.

As it will be detailed later, end users wanting to exchange data through PETVAD may experience high and variable delays. This behavior is similar to the one characterizing Tor, even if in this case the main reason has to be ascribed to overheads introduced by the overlay routing [49]. Even if the network behavior of PETVAD may discourage the use of time-sensitive and interactive applications, the feasibility of the proposed approach remains unaltered. In fact, PETVAD has to be considered a last resort to a complete block of Internet-like communications. The impact of the network layer is somewhat mitigated when in the presence of time-insensitive services like file transfers. Luckily, bulk downloads are the majority of traffic ob-

served on the Tor network [50], hence PETVAD performances are adequate to guarantee its deployment in many real-world scenarios. Concerning the case of M2M communications, e.g., lightweight signaling among tools exploited by LEAs or cyber defense deployments cooperating to exchange alarms or data, PETVAD offers satisfactory performances but at the price of some decay in the quality perceived by real-time traffic due to buffering during talk periods.

5.2. Transport-level Analysis

In this section, we focus on the behavior of the TCP, which allows to create a reliable stream between the PETVAD communicating endpoints. Another possible option concerns the use of UDP. However, it could require to implement proper error recovery and flow-control mechanisms in the upper layers that may break the semantic of pre-existent services, as well as introduce statistical signatures in the resulting traffic flows [19]. Besides, applications needing to interact with impaired channels, like those created via overlaying or information hiding mechanisms, could exploit novel and application-specific protocols like Quick UDP Internet Connections (QUIC) allowing to transmit HTTP over UDP [51]. However, they are still not widely adopted, thus their evaluation is left as a future development.

In general, the performances of the transport layer highly influence the feasibility of using PETVAD in real-world scenarios. Unfortunately, large bandwidth-delay products may lead to poor performances due to the slow-start and congestion control algorithms of TCP-like protocols [52]. To this aim, we evaluate PETVAD when used to transfer files with the de-facto standard SSH. Specifically, we used `scp`, which can be split into the following basic operations: establish a TCP connection, exchange a session key and complete login operations, transfer the file, and terminate the TCP connection. We tested performances with different file sizes, i.e., 10 kB, 100 kB, and 1 MB. We denote by kB and MB 10^3 and 10^6 bytes, respectively.

Table 1 reports the collected results. As shown, when using a direct connection, the TCP saturates all the available bandwidth and achieves high data rates leading to short transfer times. Instead, when the transport connection is tunneled through the PETVAD channel, rates reduce causing higher transfer times. This can be ascribed to two overlapping causes: higher RTTs impair congestion control algorithms of the TCP and intermittent channel availability due to the talkspurt-silence process may cause time outs. For the sake of completeness, Figure 5 shows the bursty behavior of TCP segments sent via the covert channel.

Lastly, to compare the performances of PETVAD with other mechanisms implementing a covert channel in VoIP streams, we computed the average bitrate experienced by the transport layer, which is ~19.2 kbit/s. In general, the channel capacity of PETVAD is higher than the one provided by other methods targeting de-facto standard applications like Skype. As possible examples, the technique discussed in [31] allows a capacity of ~2.8 kbit/s, while the one in [53] of ~0.6 kbit/s. Our approach also outperforms mechanisms based on RTP. In fact, the technique modulating the NTP Timestamp field within the

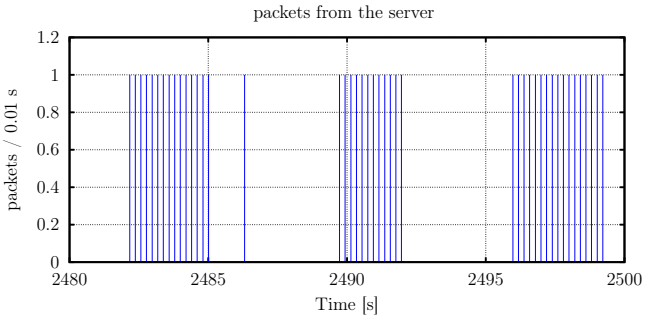

Figure 5: Bursty behavior of the segments sent through the covert channel.

RTP header presented in [24] only guarantees a steganographic bandwidth of ~0.05 kbit/s. Instead, the work in [54] accounts for a higher capacity, i.e., ~32 kbit/s, but at the price of performing a resource-consuming transcoding of the data, possibly reducing the stealthiness of the covert channel.

5.3. Application-level Analysis

To evaluate the impact of PETVAD on the application layer, we tested the performances of the Hyper Text Transfer Protocol (HTTP) for retrieving various contents in many network conditions. We consider the HTTP a good synecdoche for a large variety of scenarios, as it is used for streaming, access Software-as-a-Service frameworks, interact with a plethora of Web 2.0 applications including online security tools, IoT nodes and devices to exchange data and controls as well as services for threat intelligence [52, 55, 56, 57].

For the first round of tests, we performed trials in a local setting and we used `netem` to emulate various network conditions. Specifically, we considered different values for the delay, i.e., no delay, 20 ms, and 100 ms, and different values for the packet loss, i.e., no loss, 1%, and 5%. We point out that, "no delay" and "no loss" mean that no additional impairments have been artificially added, i.e., only those characterizing the LAN have been considered. For the second round of tests, we investigated a WAN scenario with flows affected by uncontrollable impairments happening in the wild. In this case, PETVAD is used to retrieve a content from the Internet via the proxy configuration depicted in Figure 2. For both rounds of experiments, we used three web pages of 1 MB composed of different inline objects. Each page contains 1, 10, and 100 inline objects of 1 MB, 100 kB and 10 kB of size, respectively. This permits to enlighten interactions between the covert channel and the pipelining architecture of HTTP [52, 58].

Table 2 summarizes the results obtained for the first round of experiments. As expected, PETVAD is slower than a direct access since the HTTP is influenced by the poor performances of the TCP when in the presence of high delays. Besides, the download of a unique inline object is quicker than retrieving many smaller contents. This can be ascribed to the multiple TCP connections that are penalized by the delays and the size of the inline objects preventing to saturate the available bandwidth. Concerning the impact of the packet loss, when in the presence of high values (i.e., 5%), jointly with the high delays

Table 2: Time and rate statistics for HTTP over plain connectivity and PETVAD in different network environments.

packet loss		0%					
	delay	0 ms		20 ms		100 ms	
		time [s]	rate [kB/s]	time [s]	rate [kB/s]	time [s]	rate [kB/s]
direct							
1x 1M	mean	0.10	11,062.39	0.82	1,252.30	3.61	283.43
	std. dev.	0.01	78.88	0.10	122.25	0.01	0.59
10x 100k	mean	0.10	10,213.04	1.02	1,005.07	4.98	205.80
	std. dev.	0.01	135.81	0.01	0.74	0.10	4.05
100x 10k	mean	0.17	5,927.01	4.50	227.43	21.65	47.31
	std. dev.	0.01	209.65	0.01	0.16	0.01	0.01
PETVAD							
1x 1M	mean	417.29	2.45	418.28	2.45	422.21	2.43
	std. dev.	12.68	0.07	13.29	0.08	10.28	0.06
10x 100k	mean	434.06	2.36	436.88	2.34	442.53	2.31
	std. dev.	12.54	0.07	12.17	0.06	13.81	0.07
100x 10k	mean	585.24	1.75	583.47	1.75	660.53	1.55
	std. dev.	9.90	0.03	13.96	0.04	15.21	0.04

delay		0ms					
	packet loss	0%		1%		5%	
		time [s]	rate [kB/s]	time [s]	rate [kB/s]	time [s]	rate [kB/s]
direct							
1x 1M	mean	0.09	11,062.39	0.092	11,074.35	0.61	16,72.18
	std. dev.	0.01	78.88	0.01	213.25	0.49	4330.85
10x 100k	mean	0.10	10,213.04	0.14	7,249.60	0.43	2,375.07
	std. dev.	0.01	135.81	0.08	2,719.13	0.31	3144.88
100x 10k	mean	0.17	5,927.01	0.73	1,397.25	3.55	288.69
	std. dev.	0.01	209.65	0.84	1,789.26	1.94	277.54
PETVAD							
1x 1M	mean	417.29	2.45	476.05	2.15		
	std. dev.	12.68	0.07	33.25	0.13		
10x 100k	mean	434.06	2.36	500.77	2.05	all transfers	
	std. dev.	12.54	0.07	35.71	0.14	stopped	
100x 10k	mean	585.24	1.750	736.71	1.39		
	std. dev.	9.90	0.03	50.13	0.10		

and floating "capacity" of PETVAD, the HTTP fails to retrieve the page. This is due to the additional fragility caused by multiple TCP connections "tunneled" within a single transport flow causing timeouts [55].

Table 3 contains results for the second round of tests. As reported, performances are similar to the previous case and the slightly increased transfer times are mainly due to the proxy routing data through additional layers. Besides, the proxy partially isolates the LAN from the Internet acting like a sort of protocol enhancing proxy [52].

Table 3: Time and rate statistics for HTTP over plain connectivity and PETVAD in the Internet.

		time [s]	rate [kB/s]
direct			
1x 1M	mean	1.41	725.12
	std. dev.	0.04	20.57
10x 100k	mean	2.36	434.57
	std. dev.	0.04	7.69
100x 10k	mean	16.96	60.38
	std. dev.	0.99	3.67
PETVAD			
1x 1M	mean	426.25	2.40
	std. dev.	34.25	0.15
10x 100k	mean	438.37	2.34
	std. dev.	14.39	0.07
100x 10k	mean	598.27	1.71
	std. dev.	6.13	0.02

6. User-side Analysis

To precisely understand if PETVAD can be used in the wild, we also investigate the Quality of Experience (QoE) perceived by end users both in terms of voice and application-related metrics.

6.1. Quality of Application-related Metrics

As discussed in Section 5, the conversation between the caller and the callee influences the bandwidth of the covert chan-

nel. In fact, the statistical properties of silence rule the number of fake RTP packets that can be used as a carrier. To avoid that the VoIP conversation will reveal the covert channel by appearing as an anomaly, the two UAs should behave according to realistic conversation models. This can be achieved in two manners: by highjacking real VoIP conversations or by using synthetic talkers implementing accurate vocal corpus. Since injecting data in preexistent traffic is outside of the scope of this paper, we concentrate on VoIP conversations generated via the TIMIT speech corpus [47]. In general, the performance of bidirectional channels is partially influenced by how the UAs are synchronized. In fact, realistic conversations are made of alternating talkers. Thus, to have a sort of "stealthiness bounds" we considered the performance of PETVAD for two different conditions. A realistic one, where talkers perfectly (naturally) alternate, and a worst case, where they completely overlap. Table 4 contains the collected results when PETVAD is used to retrieve various web pages. As shown, when talkers alternate, the times needed to download the page are higher. This is due to delays experienced by TCP segments containing acknowledgments. In fact, when in the presence of aligned conversations, the segments contained in fake RTP packets can be acknowledged "instantaneously" as also the other side has a silence phase making the covert channel full-duplex. Instead, when the two endpoints converse in a realistic manner, segments in fly have to wait for the next silence phase before they are acknowledged, i.e., the covert channel appears as temporarily half-duplex.

For the sake of completeness, we evaluated also if the covert communications may cause suspicions due to audible artifacts in the audio conversation. Similarly to [11], the sound quality perceived both by the caller and the callee is not deteriorated by the presence of PETVAD. In fact, a clean VoIP call is characterized by PESQ MOS values [59] equal to 4.356 and 4.349, for the caller and the callee, respectively. With PETVAD, the resulting PESQ MOS scores are 4.173 and 4.346. This does not offer an effective indicator for suspecting that a covert communication is taking place. To recap, PETVAD exhibits the same detectability of the original injection technique, which has been already assessed in [10].

6.2. QoE-related Metrics

As shown in Section 5.2, transferring data through a covert channel leads to performance losses, which can be tolerated while transferring files. Instead, when using interactive applications (e.g., web browsing) the perceived experience may be very poor. Therefore, this round of experiments has been made to assess the behavior of PETVAD in terms of QoE. To this aim, we considered a user wanting to browse the Internet in a stealthy manner. We point out that this case can model both individuals wanting to send data without being spotted by a hostile regime as well as security experts retrieving data or updating intelligence reports. To have a realistic testbed, we access popular websites by using PETVAD through a DSL connection with the UA of the callee (serving as a proxy implementing the needed information-hiding-related operations) deployed in the Internet, as depicted in Figure 2. To have proper statistical relevance, each trial has been repeated 20 times. The considered

Table 4: Performances of PETVAD when retrieving a web page with alternating and overlapping talkers.

		time [s]	rate [kB/s]
PETVAD - simultaneous talkers			
1x 1M	mean	417.29	2.45
	std. dev.	12.68	0.07
10x 100k	mean	434.06	2.36
	std. dev.	12.54	0.07
100x 10k	mean	585.24	1.75
	std. dev.	9.90	0.03
PETVAD - alternate talkers			
1x 1M	mean	439.42	2.33
	std. dev.	21.09	0.11
10x 100k	mean	507.28	2.02
	std. dev.	20.53	0.08
100x 10k	mean	1,098.54	0.93
	std. dev.	39.47	0.03

Table 5: Used sites with number of inline objects and sizes according to the used browser.

site	N. inline objects	size - regular [kB]	size - text [kB]
google.com	4	242.98	1.04
youtube.com	105	2,793.57	27.12
baidu.com	5	59.17	0.51
en.wikipedia.org	65	805.32	15.69
qq.com	167	2,706.11	38.96

websites have been selected according to popularity[5], size and complexity criteria [55]. Table 5 reports the sample sites and their complexity in terms of inline objects. Each destination has been accessed in two manners:

- regular: all the inline objects composing the page are retrieved via a standard browser (i.e., Mozilla Firefox in our trials);
- text: the Links text-based browser is used to remove multimedia contents and to enhance performances in terms of page loading time in bandwidth-scarce environments.

Table 6 presents the performances achieved both in terms of transfer times and data rates. Clearly, accessing the site in textual form reduces the amount of data to be retrieved, thus, leading to the smaller transfer times. However, this may be not enough to saturate the available capacity due to the high RTT (see Section 5.1). Therefore, richer pages with many multimedia contents (e.g., Youtube and QQ) require a transfer time in the range of several minutes. This can be acceptable for offline reading, but it actually impedes any form of real-time browsing. Luckily, accessing pages in a textual form can save time, but it is not always possible since some destinations do not have the core information in the textual form (e.g., YouTube). In our trials, a notable exception is given by Baidu for which the server reacted differently when accessed through Links. We point out

[5] See "The top 500 sites on the Web" ranking by Alexa and available at the URL `https://www.alexa.com/topsites`. Last accessed December 2019.

Table 6: Direct web browsing and using PETVAD for different destinations.

	Browser	Regular		Text	
		time [s]	rate [kB/s]	time [s]	rate [kB/s]
direct					
google.com	mean	0.94	258.04	0.53	1.97
	std. dev.	0.05	37.43	0.01	0.04
youtube.com	mean	13.93	200.60	1.38	19.70
	std. dev.	0.18	2.54	0.12	1.22
baidu.com	mean	5.16	11.48	1.85	0.27
	std. dev.	0.31	0.63	0.06	0.01
en.wikipedia.org	mean	4.16	193.57	0.34	46.24
	std. dev.	0.04	1.94	0.01	1.84
qq.com	mean	112.51	24.05	0.35	110.49
	std. dev.	10.53	2.35	0.03	11.77
PETVAD					
google.com	mean	96.46	2.61	36.40	0.03
	std. dev.	10.81	0.29	6.46	0.01
youtube.com	mean	1217.58	2.31	29.80	0.92
	std. dev.	15.97	0.02	6.22	0.20
baidu.com	mean	28.24	2.10	51.38	0.01
	std. dev.	5.83	0.42	8.56	0.01
en.wikipedia.org	mean	293.15	2.74	15.51	1.01
	std. dev.	16.71	0.14	4.53	0.31
qq.com	mean	1230.30	2.18	23.47	1.70
	std. dev.	25.30	0.04	5.53	0.39

that if the time needed to retrieve a content becomes too high (for instance, it approaches a duration in the range of hours) this can be acceptable for offline reading or to retrieve documentation from an investigative source, but it could reduce the stealthiness of PETVAD. In fact, a long lasting covert channel requires a VoIP conversation (the overt channel) that persists an equal amount of time. This can represent an anomaly, as the average duration of a conversation is in the range of minutes [60].

The information reported in Table 6 does not give a precise idea on the "quality" perceived by the users. In fact, if the objects arrive quickly or close in time, user would be able to start reading the content, even if partial. In other words, incomplete pages could be readable according to how inline objects are retrieved. To this aim, Figure 6 depicts the Cumulative Distribution Functions (CDFs) of transfer times. As it is visible, the larger the size of the website, the higher the variations experienced by the users when retrieving a page. Due to its dependence on the talkspurt-to-silence behavior, PETVAD downloads have a high degree of variability. Such a phenomena is exacerbated for content-rich pages, and using the text-based browser can be a palliative solution. Hence, PETVAD can be used to retrieve contents within the size up to 1MB before deteriorating too much the user experience. However, if the multimedia portion of the page can be sacrificed, the text-only version could suffice for a variety of web contents.

Lastly, Figure 7 portraits the waterfall diagram depicting how the inline objects are retrieved through PETVAD. For the sake of brevity, we only report results for the "synthetic" website introduced in Section 5, but similar results have been collected for the other cases. As shown, the presence of the covert channel mainly impacts on how "quick" the pipeline of HTTP can be fed, thus causing many inline objects to be blocked. Therefore, a proper caching or optimized scheduling should be deployed as to counteract decays in the QoE [61].

7. Conclusions and Future Work

In this paper we have presented a VoIP-based network covert channel framework for enhancing the privacy both of end users and entities cooperating for cyber defense purposes. In more detail, we developed a framework for embedding data in fake silence packets generated during silence periods of VAD-capable VoIP conversations. Results have indicated that the proposed approach can be used in real-world settings, but with some limitations. For instance, content-rich web browsing may require further optimizations, such as operating in a text-only fashion.

Future works aim at overcoming some limitations of PETVAD. A relevant part of our ongoing research deals with the design of synthetic talkers making realistic VoIP conversations without recurring to a static corpus. In this perspective, we aim at exploring machine-learning or AI-based agents to produce vocal patterns, possibly to have a talk-to-silence ratio adaptively matching the offered load of secret information. Another important research topic deals with application-related optimizations. Specifically, for the case of HTTP, the "proxy" placed at the UA of the caller in our case may implement some form of compression or intelligent scheduling of inline objects as to reduce the page loading time or to prioritize the retrieval of more informative contents. Another investigation will deal with the use of novel transport mechanisms like QUIC possibly with some form of bandwidth reservation mechanism between the two communicating endpoints, for instance by using

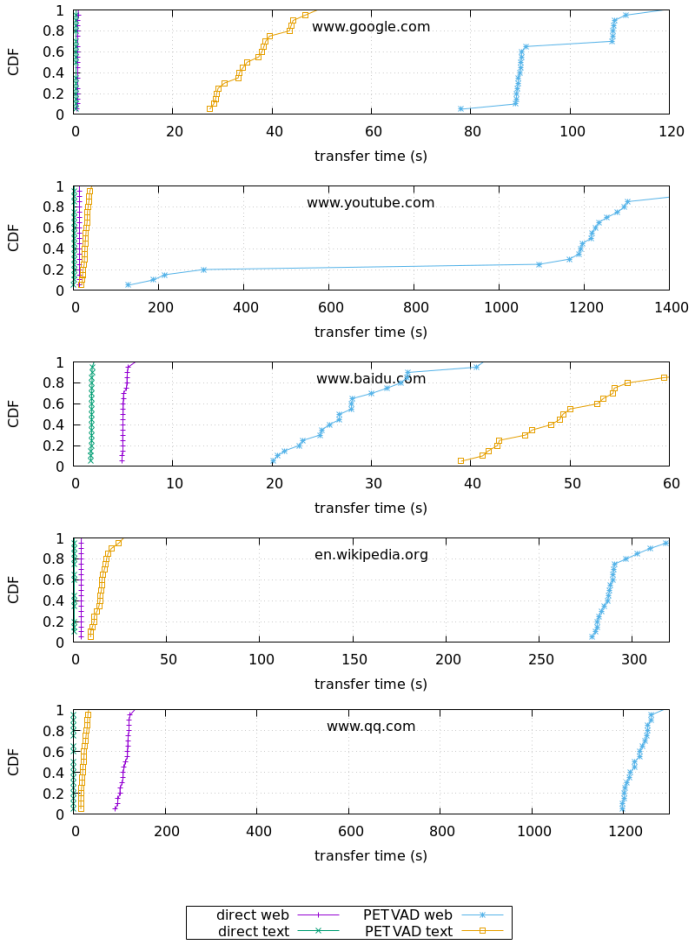

Figure 6: CDF values for the download times when retrieving websites for all the considered environments.

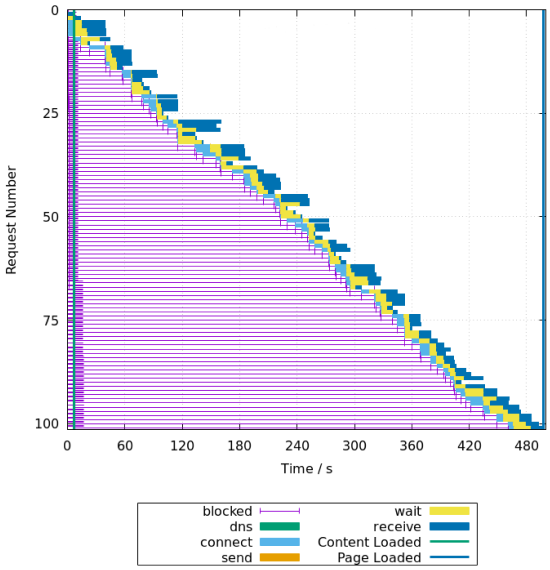

Figure 7: Waterfall diagram of the PETVAD used to retrieve 100 inline objects.

a technology-independent scheme [62]. Besides, future implementations of the PETVAD will support MitM scenarios. We also plan to port the PETVAD to Tor, especially by exporting the covert channel as a pluggable transport, i.e., a PT [17].

Lastly, a relevant part of our ongoing research deals with the investigation of countermeasures. On one hand, this allows to understand how to protect the proposed framework against blocking attempts. On the other hand, countermeasures can be useful to prevent malicious activities leveraging VoIP-based information hiding. To this aim, we envisage different approaches. The first deals with the identification of timing anomalies of RTP packets generated by PETVAD in comparison to the RTP packets generated by the same UA without a covert channel. A similar methodology can be also applied to the lower layers of the protocol stack, e.g., to timing/volume statistics of datagrams containing the VoIP flow [10, 11]. The second one instead considers the VoIP conversation as a sort of black box. In this case, the warden can be engineered to: detect when the silence periods occur and randomly normalize/remove packets sent during this time to blindly disrupt the covert channel (this is an improvement over research presented in [63, 64]), or spot VoIP communications not exhibiting a realistic evolution. We point out that, in order to have needed overt traffic, the covert endpoints may be synthetically generating a call, e.g., with artificial sound samples or vocal corpus. This can be used as a signature. Another possible indicator is the anomalous activity of the talker, which can be inferred even in the presence of encrypted traffic flows (see, [65] for a statistical attack to Skype flows able to reveal the language and the gender of the talker).

Acknowledgments

This work has been supported by the EU Project SIMARGL, Grant Agreement No 833042.

References

[1] D. Hughes, V. Shmatikov, Information hiding, anonymity and privacy: a modular approach, Journal of Computer Security 12 (1) (2004) 3–36.

[2] S. Wendzel, W. Mazurczyk, L. Caviglione, M. Meier, Hidden and uncontrolled–on the emergence of network steganographic threats, in: ISSE 2014 Securing Electronic Business Processes, Springer, 2014, pp. 123–133.

[3] J. Justin, S. Manimurugan, A survey on various encryption techniques, International Journal of Soft Computing and Engineering 2231 (2012) 2307.

[4] W. Mazurczyk, L. Caviglione, Information hiding as a challenge for malware detection, IEEE Security & Privacy 13 (2) (2015) 89–93.

[5] S. Zander, G. Armitage, P. Branch, A survey of covert channels and countermeasures in computer network protocols, IEEE Communications Surveys & Tutorials 9 (3) (2007) 44–57.

[6] W. Mazurczyk, L. Caviglione, Steganography in modern smartphones and mitigation techniques, IEEE Communications Surveys & Tutorials 17 (1) (2015) 334–357.

[7] W. Mazurczyk, K. Powójski, L. Caviglione, IPv6 covert channels in the wild, in: Proceedings of the Third Central European Cybersecurity Conference, ACM, 2019, p. 10.

[8] B. Carrara, C. Adams, Out-of-band covert channels—a survey, ACM Computing Surveys 49 (2) (2016) 23.

[9] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, Detailed analysis of Skype traffic, IEEE Transactions on Multimedia 11 (1) (2009) 117–127.

[10] S. Schmidt, W. Mazurczyk, R. Kulesza, J. Keller, L. Caviglione, Exploiting IP telephony with silence suppression for hidden data transfers, Computers & Security 79 (2018) 17–32.

[11] S. Schmidt, W. Mazurczyk, J. Keller, L. Caviglione, A new data-hiding approach for IP telephony applications with silence suppression, in: Proceedings of the 12th International Conference on Availability, Reliability and Security, ACM, 2017.

[12] J. Heurix, P. Zimmermann, T. Neubauer, S. Fenz, A taxonomy for privacy enhancing technologies, Computers & Security 53 (2015) 1–17.

[13] D. F. Vázquez, O. P. Acosta, C. Spirito, S. Brown, E. Reid, Conceptual framework for cyber defense information sharing within trust relationships, in: 2012 4th International Conference on Cyber Conflict, IEEE, 2012, pp. 1–17.

[14] G. Oikonomou, J. Mirkovic, P. Reiher, M. Robinson, A framework for a collaborative DDoS defense, in: 2006 22nd Annual Computer Security Applications Conference, IEEE, 2006, pp. 33–42.

[15] A. Pfitzmann, M. Hansen, A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, online: `http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf`.

[16] P. Syverson, G. Tsudik, M. Reed, C. Landwehr, Towards an analysis of onion routing security, in: Designing Privacy Enhancing Technologies, Springer, 2001, pp. 96–114.

[17] P. Winter, S. Lindskog, How the Great Firewall of China is blocking Tor, in: Free and Open Communications on the Internet, ACM, 2012.

[18] H. Mohajeri Moghaddam, B. Li, M. Derakhshani, I. Goldberg, Skypemorph: Protocol obfuscation for TOR bridges, in: ACM conference on Computer and Communications Security, ACM, 2012, pp. 97–108.

[19] A. Houmansadr, C. Brubaker, V. Shmatikov, The parrot is dead: Observing unobservable network communications, in: 2013 IEEE Symposium on Security and Privacy, IEEE, 2013, pp. 65–79.

[20] W. Mazurczyk, K. Szczypiorski, Steganography of VoIP streams, in: OTM Confederated International Conferences on the Move to Meaningful Internet Systems, Springer, 2008, pp. 1001–1018.

[21] W. Mazurczyk, VoIP steganography and its detection — a survey, ACM Computing Surveys 46 (2) (2013) 20.

[22] E. Zielińska, W. Mazurczyk, K. Szczypiorski, Trends in steganography, Communications of the ACM 57 (3) (2014) 86–95.

[23] L. Y. Bai, Y. Huang, G. Hou, B. Xiao, Covert channels based on jitter field of the RTCP header, in: International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2008, pp. 1388–1391.

[24] Y. Huang, J. Yuan, M. Chen, B. Xiao, Key distribution over the covert communication based on VoIP, Chinese Journal of Electronics 20 (2) (2011) 357–360.

[25] C. R. Forbes, A new covert channel over RTP, Thesis/Dissertation Collections.

[26] T. Takahashi, W. Lee, An assessment of VoIP covert channel threats, in: Third International Conference on Security and Privacy in Communications Networks, IEEE, 2007, pp. 371–380.

[27] A. Janicki, W. Mazurczyk, K. Szczypiorski, Steganalysis of transcoding steganography, Annals of Telecommunications - Annales des Télécommunications 69 (7) (2014) 449–460.

[28] C. Forbes, B. Yuan, D. Johnson, P. Lutz, A covert channel in RTP protocol, in: Computational Intelligence: Foundations and Applications, World Scientific, 2010, pp. 813–819.

[29] Y. Huang, S. Tang, C. Bao, Y. J. Yip, Steganalysis of compressed speech to detect covert voice over Internet protocol channels, IET Information Security 5 (1) (2011) 26–32.

[30] A. Janicki, Pitch-based steganography for Speex voice codec, Security and Communication Networks 9 (15) (2016) 2923–2933.

[31] W. Mazurczyk, M. KaraÅ, K. Szczypiorski, Skyde: a skype-based steganographic method, International Journal of Computers Communications & Control 8 (3) (2013) 432–443. `doi:10.15837/ijccc.2013.3.469`.
URL `http://univagora.ro/jour/index.php/ijccc/article/view/469`

[32] X. Wang, S. Chen, S. Jajodia, Tracking anonymous peer-to-peer VoIP calls on the Internet, in: Proceedings of the 12th ACM Conference on Computer and Communications Security, ACM, 2005, pp. 81–91.

[33] M. H. Almeshekah, M. J. Atallah, E. H. Spafford, Enhancing passwords security using deceptive covert communication, in: H. Federrath, D. Gollmann (Eds.), ICT Systems Security and Privacy Protection, Cham, 2015, pp. 159–173.

[34] D. V. Bailey, D. Boneh, E. Goh, A. Juels, Covert channels in privacy-preserving identification systems, in: Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007, 2007, pp. 297–306.

[35] M. Hanspach, M. Goetz, On covert acoustical mesh networks in air, JCM 8 (11) (2013) 758–767. `doi:10.12720/jcm.8.11.758-767`.
URL `https://doi.org/10.12720/jcm.8.11.758-767`

[36] M. Guri, B. Zadov, D. Bykhovsky, Y. Elovici, Powerhammer: Exfiltrating data from air-gapped computers through power lines, IEEE Transactions on Information Forensics and Security.

[37] M. Guri, B. Zadov, Y. Elovici, Odini: Escaping sensitive data from faraday-caged, air-gapped computers via magnetic fields, IEEE Transactions on Information Forensics and Security 15 (2019) 1190–1203.

[38] I. Ghafir, V. Prenosil, J. Svoboda, M. Hammoudeh, A survey on network security monitoring systems, in: Proc. IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), 2016, pp. 77–82. `doi:10.1109/W-FiCloud.2016.30`.

[39] C. Kraetzer, J. Dittmann, Mel-cepstrum based steganalysis for voip steganography, in: Security, steganography, and watermarking of multimedia contents IX, Vol. 6505, International Society for Optics and Photonics, 2007, p. 650505.

[40] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, SIP: session initiation protocol, Tech. rep. (2002).

[41] A. Sangwan, M. Chiranth, H. Jamadagni, R. Sah, R. V. Prasad, V. Gaurav, VAD techniques for real-time speech transmission on the Internet, in: 5th IEEE International Conference on High Speed Networks and Multimedia Communications, IEEE, 2002, pp. 46–50.

[42] J. Berger, A. Hellenbart, R. Ullmann, B. Weiss, S. Moller, J. Gustafsson, G. Heikkila, Estimation of 'quality per call' in modelled telephone conversations, in: IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2008, pp. 4809–4812.

[43] V. Prasad, A. Sangwan, H. Jamadagni, M. Chiranth, R. Sah, V. Gaurav, Comparison of voice activity detection algorithms for VoIP, in: Seventh International Symposium on Computers and Communications, IEEE, 2002, pp. 530–535.

[44] R. Zopf, Real-time transport protocol (RTP) payload for comfort noise (CN), Tech. rep. (2002).

[45] J. Fridrich, Steganography in digital media: principles, algorithms, and applications, Cambridge University Press, 2009.

[46] A. Moizard, G. Herlein, J.-M. Valin, A. E. Heggestad, RTP payload format for the Speex codec, RFC 5574.

[47] V. Zue, S. Seneff, J. Glass, Speech database development at MIT: TIMIT and beyond, Speech Communication 9 (4) (1990) 351–356.

[48] L. Caviglione, M. Podolski, W. Mazurczyk, M. Ianigro, Covert channels in personal cloud storage services: The case of Dropbox, IEEE Transactions on Industrial Informatics 13 (4) (2017) 1921–1931.

[49] P. Dhungel, M. Steiner, I. Rimac, V. Hilt, K. W. Ross, Waiting for anonymity: Understanding delays in the Tor overlay, in: 2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P), IEEE, 2010, pp. 1–4.

[50] D. Gopal, N. Heninger, Torchestra: Reducing interactive traffic delays over Tor, in: Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society, ACM, 2012, pp. 31–42.

[51] G. Carlucci, L. De Cicco, S. Mascolo, HTTP over UDP: an experimental investigation of QUIC, in: Proceedings of the 30th Annual ACM Symposium on Applied Computing, ACM, 2015, pp. 609–614.

[52] L. Caviglione, Can satellites face trends? The case of Web 2.0, in: Satellite and Space Communications, 2009. IWSSC 2009. International Workshop on, IEEE, 2009, pp. 446–450.

[53] W. Mazurczyk, Lost audio packets steganography: the first practical evaluation, Security and Communication Networks 5 (12) (2012) 1394–1403. `arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/sec.502`, `doi:10.1002/sec.502`.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.502`

[54] W. Mazurczyk, P. Szaga, K. Szczypiorski, Using transcoding for hidden communication in ip telephony, Multimedia Tools Appl. 70 (3) (2014) 21392165. `doi:10.1007/s11042-012-1224-8`.

URL https://doi.org/10.1007/s11042-012-1224-8

[55] A. Cardaci, L. Caviglione, A. Gotta, N. Tonellotto, Performance evaluation of SPDY over high latency satellite channels, in: International Conference on Personal Satellite Services, Springer, 2013, pp. 123–134.

[56] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, D. Estrin, A first look at traffic on smartphones, in: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, ACM, 2010, pp. 281–287.

[57] S. K. Datta, C. Bonnet, N. Nikaein, An IoT gateway centric architecture to provide novel M2M services, in: 2014 IEEE World Forum on Internet of Things, IEEE, 2014, pp. 514–519.

[58] J. Heidemann, K. Obraczka, J. Touch, Modeling the performance of http over several transport protocols, IEEE/ACM Transactions on Networking 5 (5) (1997) 616–630.

[59] T. Falk, W.-Y. Chan, Performance study of objective speech quality measurement for modern wireless-VoIP communications, EURASIP Journal on Audio, Speech, and Music Processing 2009 (1) (2009) 104382.

[60] S. Guha, N. Daswani, An experimental study of the Skype peer-to-peer VoIP system, Tech. rep., Cornell University (2005).

[61] A. Cardaci, L. Caviglione, E. Ferro, A. Gotta, Using spdy to improve web 2.0 over satellite links, International Journal of Satellite Communications and Networking 35 (4) (2017) 307–321.

[62] L. Caviglione, F. Davoli, Peer-to-peer middleware for bandwidth allocation in sensor networks, IEEE communications letters 9 (3) (2005) 285–287.

[63] G. Fisk, M. Fisk, C. Papadopoulos, J. Neil, Eliminating steganography in internet traffic with active wardens, in: F. A. P. Petitcolas (Ed.), Information Hiding, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 18–35.

[64] W. Mazurczyk, S. Wendzel, M. Chourib, J. Keller, Countering adaptive network covert communication with dynamic wardens, Future Generation Computer Systems 94 (2019) 712 – 725. doi:https://doi.org/10.1016/j.future.2018.12.047.
URL http://www.sciencedirect.com/science/article/pii/S0167739X18316133

[65] C. V. Wright, L. Ballard, F. Monrose, G. M. Masson, Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob?, in: USENIX Security Symposium, Vol. 3, 2007, pp. 43–54.

Jens Saenger graduated at FernUniversität Hagen in Informatics. His main research interests are network security and privacy.

Wojciech Mazurczyk is an university professor at the Institute of Computer Science at Warsaw University of Technology (WUT). His research interests include network security, information hiding, and network forensics. Mazurczyk received PhD and DSc degrees in telecommunications from WUT. He is also an associate editor for IEEE Transactions on Information Forensics and Security and Mobile Communications and Networks Series Editor for the IEEE Communications Magazine.

Jörg Keller is a professor of Parallelism & VLSI at the Faculty of Mathematics and Computer Science of FernUniversität in Hagen, Germany. His research interests include, among others, energy-efficient parallel computing, security and cryptography and fault tolerant computing. Prof. Keller is a member of Editorial Board for Journal of Universal Computer Science and Journal of Cyber Security and Mobility, as well as Program Committee member of several conferences/workshops.

Luca Caviglione is a research scientist with the Institute for Applied Mathematics and Information Technologies at the National Research Council of Italy. His research interests include networking with emphasis on wireless communications, cloud architectures, and network security. He received a PhD in electronics and computer engineering from the University of Genoa.