Disabling a Computer by Exploiting Softphone Vulnerabilities: Threat and Mitigation

Ryan Farley and Xinyuan Wang

Department of Computer Science
George Mason University
Fairfax, VA 22030, USA
email: {rfarley3,xwangc}@gmu.edu

Abstract. As more and more people are using VoIP softphones in their laptop and smart phones, vulnerabilities in VoIP protocols and systems could introduce new threats to the computer that runs the VoIP softphone. In this paper, we investigate the security ramifications that VoIP softphones expose their host to and ways to mitigate such threats.
We show that crafted SIP traffic (noisy attack) can disable a Windows XP host that runs the official Vonage VoIP softphone within several minutes. While such a noisy attack can be effectively mitigated by threshold based filtering, we show that a stealthy attack could defeat the threshold based filtering and disable the targeted computer silently without ever ringing the targeted softphone.
To mitigate the stealthy attack, we have developed a limited context aware (LCA) filtering that leverages the context and SIP protocol information to ascertain the intentions of a SIP message on behalf of the client. Our experiments show that LCA filtering can effectively defeat the stealthy attack while allowing legitimate VoIP calls to go through.

Keywords: VoIP, Softphone, Security, Attack on Host, Host Defense.

1 Introduction

As VoIP is getting increasingly popular, it becomes an attractive target to attackers [8]. Many known VoIP exploits stem from vulnerabilities of the de facto signaling protocol in use, Session Initiation Protocol (SIP) [16]. Previous works [27, 12, 8, 23] have shown that SIP weaknesses make it possible for the attackers to do such things as remotely monitor a call, modify billing control signals, and even implement voice pharming attacks. As with any device on the Internet, if a VoIP phone is vulnerable and unprotected, then an attacker can exploit it from anywhere in the world.

In this paper we take an alternative point of view by focusing on the stability and security of the systems that host VoIP softphones. As with most network based applications softphones enlarge a device's attack surface, which increases the chance that an attacker can find a point of leverage and pivot to compromise the host machine. In this paper we case study the softphone provided by Vonage, which at one point had the largest US residential VoIP market share [13, 14], in order to investigate stability and security threats that a VoIP softphone can introduce to the host running it and how we can mitigate such threats.

Specifically, we present two attacks against a Windows XP host running the official Vonage softphone. These attacks use crafted SIP messages and can make the Windows XP Host completely unusable until reboot. Indirectly, these attacks also prevent the victims from receiving incoming and making outgoing calls within seconds. The first attack (the noisy attack) can remotely disable a machine running the Vonage softphone by occupying all available physical and virtual memory within minutes. The second attack (the stealthy attack) takes longer to achieve the same effect, but it never rings the softphone. These attacks illustrate that weaknesses in a VoIP softphone can introduce fatal vulnerabilities to the host system. Few people may realize that, indeed, a vulnerable application can enable a remote attacker to completely disable the computer that runs the vulnerable application.

We have investigated ways to mitigate the identified attacks from the network. We have first used threshold based filtering to detect the spikes in arrival rates of `Invite` messages, and we have found that it can effectively diminish the effects of the noisy attack by as much as 99.8%. However, threshold based filtering is not effective against the stealthy attack, which neither rings the softphone nor use abnormally high rate of SIP messages. We have designed the *limited context aware (LCA)* approach, which buffers all incoming packets in a waiting queue to determine whether they are attack related or safe legitimate traffic. Our experimental results show that our LCA method can eliminate 100% of the stealthy attack's packets without interfering with standard SIP operation.

The rest of the paper is organized as follows. In section 2 we illustrate the basis of our two attacks by presenting signal flooding techniques to disable the operation of the softphone itself. We then present our two attacks for the softphone host in section 3, and two defense mechanisms against the identified attacks in section 4. In section 5, we empirically evaluate the two attacks and the effectiveness of our proposed defenses against them. We discuss the existing related work in section 7 and the implications of the identified attacks on a softphone host in section 6. Finally we conclude in section 8.

2 Background

Session Initiation Protocol (SIP) [16], is a general purpose application layer signaling protocol used for creating, modifying, and terminating multimedia sessions, such as VoIP calls, among Internet endpoints known as User Agents (UAs). To facilitate locating UAs, all users in a SIP network are identified by a SIP Uniform Resource Identifier (URI), which typically includes an username and hostname in a format much like an email address.

Signaling between UAs is based on the request-response paradigm. A User Agent Client (UAC) sends requests to a User Agent Server (UAS) which then

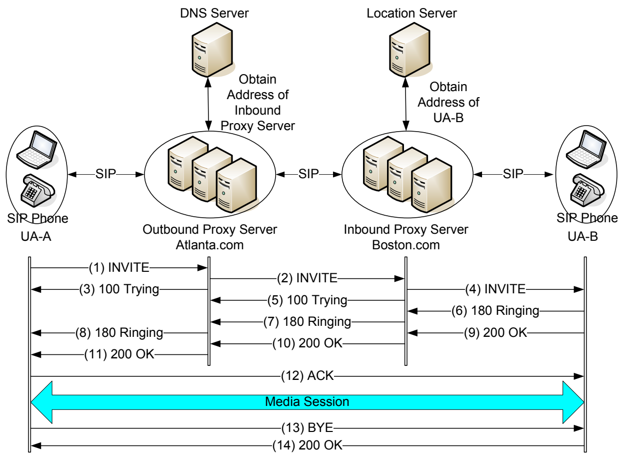


Fig. 1. An example SIP message flow for a VoIP call.

```
INVITE sip:17031234567@129.174.130.175:5060 SIP/2.0 Via:
SIP/2.0/UDP 216.115.20.41:5061 Via: SIP/2.0/UDP 216.115.20.29:5060
Via: SIP/2.0/UDP 216.115.27.11:5060;branch=z9hG4bK8AE8A3914F0
From: "GMU" <sip:17032345678@216.115.27.11>;tag=455412559 To:
<sip:17031234567@voncp.com> Call-ID:
58A8C0B-8D6F11DC-B8E18C7A-2083704C@216.115.27.11 CSeq: 101 INVITE
Contact: <sip:17032345678@216.115.20.41:5061> Max-Forwards: 13
X-Von-Relay: 216.115.27.30 Content-Type: application/sdp
Content-Length:   361

v=0 o=CiscoSystemsSIP-GW-UserAgent 5330 7344 IN IP4 216.115.27.30
s=SIP Call c=IN IP4 216.115.27.30 t=0 0 m=audio 13598 RTP/AVP 0 18
2 100 101 c=IN IP4 216.115.27.30 a=rtpmap:0 PCMU/8000 a=rtpmap:18
G729/8000 a=fmtp:18 annexb=no a=rtpmap:2 G726-32/8000 a=rtpmap:100
X-NSE/8000 a=fmtp:100 192-194 a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
```

Fig. 2. An example SIP Invite message with SDP information.

replies with both the appropriate response and a corresponding status code. An endpoint can function as both UAC and UAS at the same time.

Figure 1 illustrates the message flow of a normal SIP VoIP session between UA-A and UA-B without authentication. Initially, A only knows the URI of B. Since this does not provide the specific location information needed to complete a call, A must send the `Invite` to its outbound proxy server, atlanta.com. Once atlanta.com resolves the URI, it forwards the `Invite` to the appropriate next hop, boston.com. Next, boston.com relays the `Invite` to B and sends a `Trying` back to atlanta.com, which is relayed to A. Once B receives the `Invite`, it sends a `Ringing` to A. When B finally answers the call, it sends a `200 Ok` to A, to which A responds with an `Ack`.

During this exchange, the packets also contain a session description protocol, or SDP. This establishes the voice data parameters each client will use, such as media codec and port numbers. Figure 2 shows an example of an `Invite` message with SDP information.

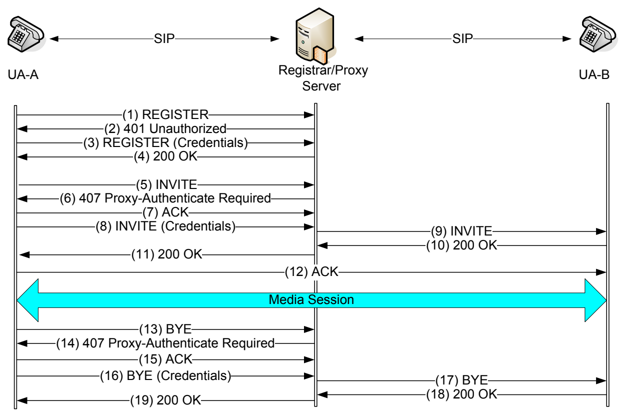


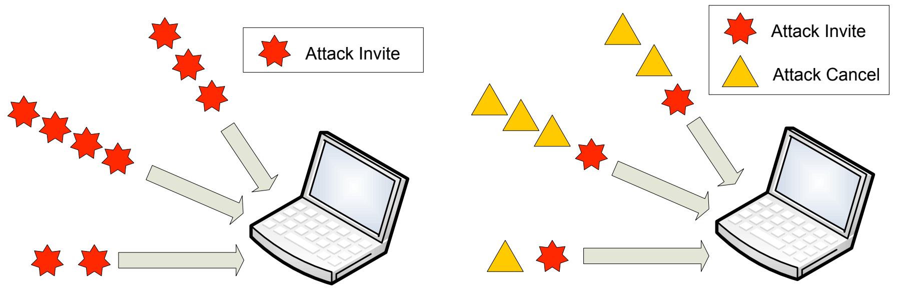
Fig. 3. Typical registration, as well as call setup and termination.

At the end of the call, the UAC that first hangs up sends a `BYE` to the other UAC. The other UAC then responds with a `200 Ok` and terminates its RTP stream. Upon receiving the `200 Ok`, the first UAC terminates its RTP stream as well.

Based on HTTP digest authentication [3], SIP provides authentication of `Invite`, `Register` and `Bye` messages from UAC to UAS. Figure 3 illustrates the message flows of typical authenticated registration, call setup and termination. Because SIP does not require the UAC to authenticate the `Invite` message from UAS, most VoIP service providers (e.g., Vonage, AT&T) do not have authentication protection of the `Invite` message sent from the SIP server to SIP phone. This enables attackers to freely spoof `Invite` message and ring any targeted SIP phone with the spoofed `Invite` message.

Once the Vonage softphone receives a spoofed `Invite`, it will keep ringing for 3 minutes unless the user picks up the softphone or a corresponding `Cancel` message is received. The SIP softphone will not ring for duplicate Call-IDs repeated within a certain period (60 seconds for the Vonage softphone). The Vonage softphone has two ports allocated for incoming RTP audio streams, creating a limit of two simultaneous phone "lines." Therefore, attackers only need to send two spoofed `Invite` messages to occupy the two lines of the targeted Vonage softphone.

In our experiments we have found that two things will happen when both lines of the Vonage softphone are ringing. First, all additional incoming call requests are given a `Busy` response. For legitimate calls routed through the client's proxy, this means that the caller is sent to voicemail. Second, since no line is free, the target can not make an outgoing call. If the target answers and hangs up on a fake `Invite`, then that line would become free until the next `Invite` arrives. For the Vonage softphone, lines become available again when audible ringing stops at three minutes. In other words, only two spoofed `Invite` messages are needed every three minutes to occupy both lines and prevent the softphone from



(a) Noisy attack. (b) Stealthy attack.

Fig. 4. Visualization of the Attack from Multiple Sources.

receiving incoming calls. In the rest of this paper, we investigate how attackers can exploit the vulnerabilities of the SIP protocol and SIP softphones to disable a Windows XP host that runs the official Vonage SIP softphone.

3 Disabling the Softphone Host

Most previous attacks on VoIP have targeted either the VoIP infrastructure (such as the proxy servers, billing systems, network usability, and fundamentals of SIP signaling) or the physical VoIP devices. It is not clear precisely how weaknesses in the softphone open the host up for attack. In the following sections we show how attackers can disable Windows XP machines running the official Vonage softphone. In section 3.1, we describe a host DoS attack that can consume a target machine's memory resources in minutes. In section 3.2, the attack is significantly refined into a slower but much stealthier form.

3.1 Noisy Attack on Softphone Host

According to the SIP specification [16], the signaling processing at the the caller's and callee's sides are inherently stateful. In other words, the end UAC needs to allocate memory for keeping the state information for every Call-ID seen in any received `Invite` so that the UAC can respond to all requests that require repeated responses. For example, if all incoming lines are busy, up to three `Busy` responses over ten seconds are sent for every `Invite` request received. This enables attackers to deplete the memory of the host running SIP softphone.

To launch such a noisy attack, the attacker only needs to make sure that each crafted `Invite` message has an unique Call-ID. The attacker can easily spoof the source IP address and launch the attack from distributed places as visualized in figure 4(a).

Specifically, the attacker can send a large number of crafted `Invite` messages at a high rate (e.g., hundreds per second) to the targeted softphone. The recipient will only hear as many simultaneous rings as they have lines (e.g., Vonage

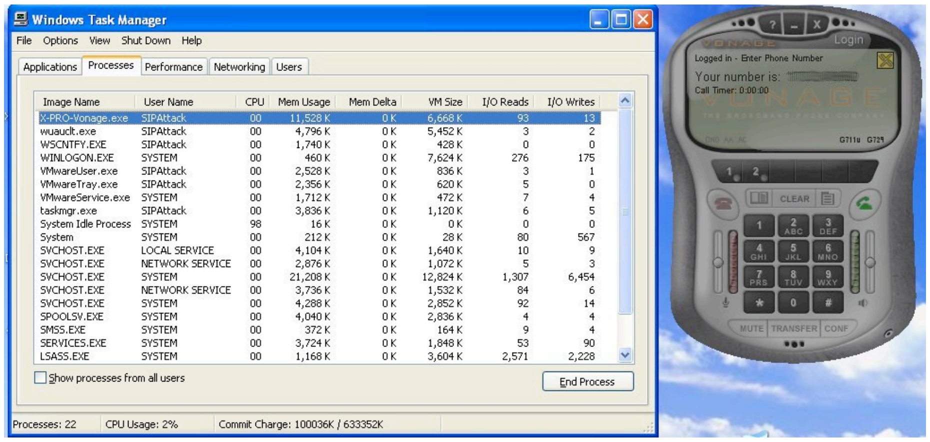

Windows Task Manager
File Options View Shut Down Help
Applications Processes Performance Networking Users
Show processes from all users
End Process
VONAGE
Login
VONAGE
THE BROADBAND PHONE COMPANY
DND AA AC
G711u G729
1 2
CLEAR
1 2 ABC 3 DEF 4 GHI 5 JKL 6 MNO 7 PRS 8 TUV 9 WXY * 0 #
MUTE TRANSFER CONF

Image Name	User Name	CPU	Mem Usage	Mem Delta	VM Size	I/O Reads	I/O Writes
X-PRO-Vonage.exe	SIPAttack	00	11,528 K	0 K	6,668 K	93	13
wuauclt.exe	SIPAttack	00	4,796 K	0 K	5,452 K	3	2
WSCNTFY.EXE	SIPAttack	00	1,740 K	0 K	428 K	0	0
WINLOGON.EXE	SYSTEM	00	460 K	0 K	7,624 K	276	175
VMwareUser.exe	SIPAttack	00	2,528 K	0 K	836 K	3	1
VMwareTray.exe	SIPAttack	00	2,356 K	0 K	620 K	5	0
VMwareService.exe	SYSTEM	00	1,712 K	0 K	472 K	7	4
taskmgr.exe	SIPAttack	00	3,836 K	0 K	1,120 K	6	5
System Idle Process	SYSTEM	98	16 K	0 K	0 K	0	0
System	SYSTEM	00	212 K	0 K	28 K	80	567
SVCHOST.EXE	LOCAL SERVICE	00	4,104 K	0 K	1,640 K	10	9
SVCHOST.EXE	NETWORK SERVICE	00	2,876 K	0 K	1,072 K	5	3
SVCHOST.EXE	SYSTEM	00	21,208 K	0 K	12,824 K	1,307	6,454
SVCHOST.EXE	NETWORK SERVICE	00	3,736 K	0 K	1,532 K	84	6
SVCHOST.EXE	SYSTEM	00	4,288 K	0 K	2,852 K	92	14
SPOOLSV.EXE	SYSTEM	00	4,040 K	0 K	2,836 K	4	4
SMSS.EXE	SYSTEM	00	372 K	0 K	164 K	9	4
SERVICES.EXE	SYSTEM	00	3,724 K	0 K	1,848 K	53	90
LSASS.EXE	SYSTEM	00	1,168 K	0 K	3,604 K	2,571	2,228

Processes: 22 CPU Usage: 2% Commit Charge: 100036K / 633352K

Logged in - Enter Phone Number
Your number is: [redacted]
Call Timer: 0:00:00

Fig. 5. Screenshot of the memory usage of the Windows XP host before the attack.

softphone has two lines). The high rate of crafted `Invite` messages will disable the softphone even if the user keeps hanging up the fake calls.

Because the softphone will allocate memory for each incoming `Invite` message even if the phone lines are busy, a high rate (e.g., hundreds messages per second) of `Invite` messages can occupy almost all free physical memory on the host within a few minutes. As the attack continues and memory usage grows, if the system does not have enough RAM to handle the allocations, then the user will receive out of memory errors and excessive virtual memory page swaps will make the host completely unresponsive.

The fundamental vulnerabilities here are: 1) the softphone needs to keep the state of every incoming `Invite` message as specified in the SIP RFC; 2) the deallocation of the memory can not keep up with the high rate of memory allocation triggered by the high rate of `Invite` messages. Therefore, even if the softphone can hang up (the fake calls) faster that the incoming `Invite` messages, the memory consumption would continue to increase and eventually lead to occupying all free memory of the host. We found these vulnerabilities are specific to the `Invite` message. Other SIP messages, such as `Cancel` and `Bye` do not have a similar effect on memory allocation.

We implemented this attack and were able to deplete almost all available free memory of a Windows XP host running a Vonage softphone in just several minutes. The host begins thrashing indefinitely and is unusable until reboot.

3.2 Stealthy Attack on Softphone Host

The noisy attack will cause the target softphone to ring if it is not used. From the attacker's point of view, it is desirable to attack the softphone host without ever ringing the softphone so that the softphone users will not be alerted.

In SIP, the call initiator is allowed to send `Cancel` messages after sending out the `Invite` message. This essentially tells the receiver to ignore any `Invite`

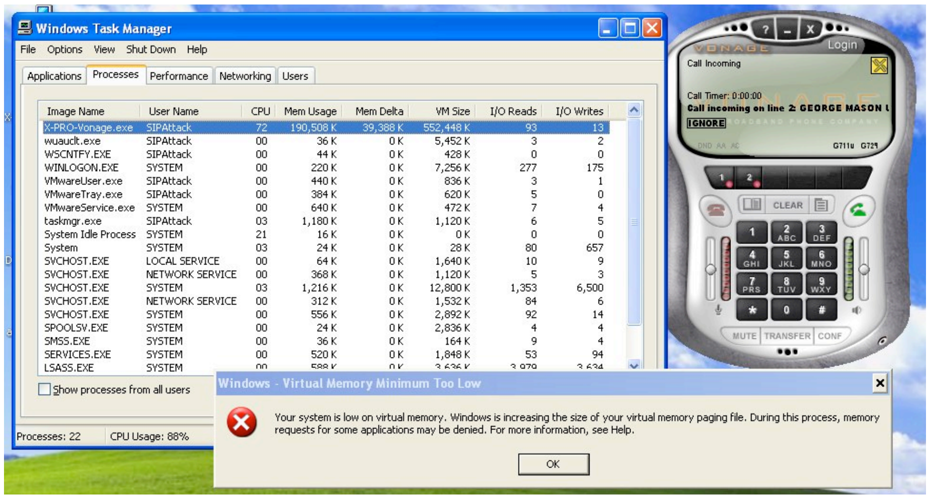

Windows Task Manager
File Options View Shut Down Help
Applications Processes Performance Networking Users
Show processes from all users
VONAGE
Login
IGNORE
DND AA AC
G711u G729
CLEAR
MUTE TRANSFER CONF

Image Name	User Name	CPU	Mem Usage	Mem Delta	VM Size	I/O Reads	I/O Writes
X-PRO-Vonage.exe	SIPAttack	72	190,508 K	39,388 K	552,448 K	93	13
wuauclt.exe	SIPAttack	00	36 K	0 K	5,452 K	3	2
WSCNTFY.EXE	SIPAttack	00	44 K	0 K	428 K	0	0
WINLOGON.EXE	SYSTEM	00	220 K	0 K	7,256 K	277	175
VMwareUser.exe	SIPAttack	00	440 K	0 K	836 K	3	1
VMwareTray.exe	SIPAttack	00	384 K	0 K	620 K	5	0
VMwareService.exe	SYSTEM	00	640 K	0 K	472 K	7	4
taskmgr.exe	SIPAttack	03	1,180 K	0 K	1,120 K	6	5
System Idle Process	SYSTEM	21	16 K	0 K	0 K	0	0
System	SYSTEM	03	24 K	0 K	28 K	80	657
SVCHOST.EXE	LOCAL SERVICE	00	64 K	0 K	1,640 K	10	9
SVCHOST.EXE	NETWORK SERVICE	00	368 K	0 K	1,120 K	5	3
SVCHOST.EXE	SYSTEM	03	1,216 K	0 K	12,800 K	1,353	6,500
SVCHOST.EXE	NETWORK SERVICE	00	312 K	0 K	1,532 K	84	6
SVCHOST.EXE	SYSTEM	00	556 K	0 K	2,892 K	92	14
SPOOLSV.EXE	SYSTEM	00	24 K	0 K	2,836 K	4	4
SMSS.EXE	SYSTEM	00	36 K	0 K	164 K	9	4
SERVICES.EXE	SYSTEM	00	520 K	0 K	1,848 K	53	94
LSASS.EXE	SYSTEM	00	588 K	0 K	3,636 K	3,979	3,634

Processes: 22 CPU Usage: 88%

Windows - Virtual Memory Minimum Too Low

Your system is low on virtual memory. Windows is increasing the size of your virtual memory paging file. During this process, memory requests for some applications may be denied. For more information, see Help.

OK

Call Incoming

Call Timer: 0:00:00

Call incoming on line 2: GEORGE MASON U

Fig. 6. Screenshot of the memory usage of the Windows host after the attack. The error message notes that the virtual memory is filled and the the swap file size needs to be increased.

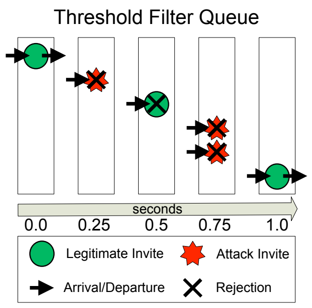
message that has the matching Call-ID. As a result, the receiver SIP phone will not ring.

As indicated in figure 4(b), the stealthy attack exploits this feature of SIP by sending a number of `Cancel` messages with the matching Call-ID immediately after every `Invite` message is sent. While only one `Cancel` message is needed to silence the receiver SIP phone, multiple `Cancel` messages are used here to make sure any sent `Invite` message will not ring the receiver SIP phone even if some `Cancel` messages are lost in the network.

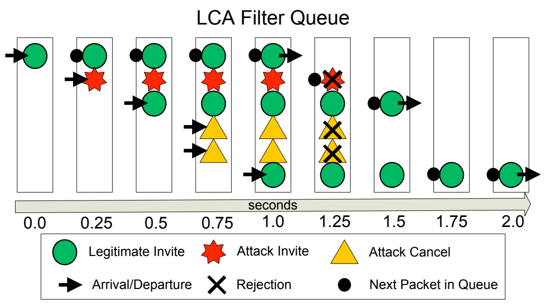
Because of the n extra `Cancel` messages per `Invite` message, the rate of `Invite` messages needs to be reduced to $\frac{1}{n+1}$ of that of noisy attack to avoid network congestion and packet loss. This slows down the memory consumption due to the attack and it takes longer to disable the softphone host. However, since the stealthy attack will never ring the softphone, it can remain undetected over a significantly longer time period. The end result is the same, almost all memory is occupied and the machine will need to be rebooted in order to be usable again.

We have implemented this stealthy attack. The empirical results show significant effects on a target system within the first half-hour. It takes about two hours depending on the rate of `Invite` messages for the stealthy attack to deplete the host memory and disable the host completely.

Figure 5 shows the memory usage and CPU utilization of the Windows XP host before the attack. The official Vonage softphone used less than 12MB memory and less than 7MB virtual memory file. The CPU utilization of the Windows XP host is only 2%.



(a) Packets are processed immediately.



(b) Packets are held for one second, then processed.

Fig. 7. Visualizations of the filter queueing systems.

As shown in Figure 6, memory usage and CPU utilization of the Windows XP host went up significantly after an attack. Specifically, the attack has not only increased the memory usage of the Vonage softphone from 11MB to 190MB, but also increased its virtual memory usage from 6MB to 552MB. This effectively depleted the available free memory of the entire Windows XP host and caused it to hang.

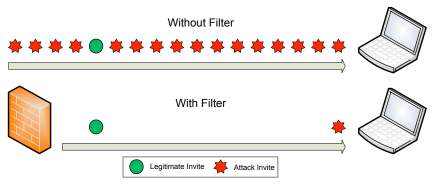
4 Defense Mechanisms

Proper defense against SIP spoofing attacks is not an easy solution without authentication or encryption. As discussed in section 2, the softphone is programmed to respond to even vaguely authentic looking signals, which makes it very difficult to distinguish between real and fake on a packet to packet basis. In order to make a decision, a grouping or series of packets must be analyzed.

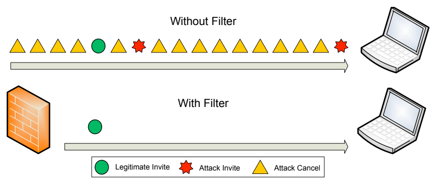
Of course, analyzing too many packets per grouping in the defense may introduce unacceptably high latency for SIP to remain operational. On the other hand, too few packets would make it hard for the defense to reliably determine whether the traffic is an attack or not. Analyzing only previous packets, such as with state machines, prevents the mechanism from blocking the beginning of attacks. This is a serious disadvantage if the attack only consists of unrelated single packets, such as our noisy DoS attack.

There are some external factors that can guide the defense mechanism though. For instance, the average person should not expect to receive more than one or two calls a second. Anything more than that is probably due to network error or a flooding attack. For another example, if a call is cancelled by the caller before the callee would have a reasonable chance to answer, then it is unnecessary to ring the callee at all. There heuristics allow us to simplify the defense by excluding the those SIP messages that would not set up meaningful calls.

Given previous work and noted limitations, we present two defense mechanisms for SIP flooding attacks. The threshold (TH) filter, in section 4.1, considers



(a) Noisy attack v. Threshold filter.



(b) Stealthy attack v. LCA filter.

Fig. 8. Attack and legitimate mixed streams with and without the filters activated.

only one factor: the rate at which `Invite` messages are arriving. The limited context aware (LCA) filter, in section 4.2, queues `Invite` messages long enough to determine if there is any associated cancelling signals. If state machine based filtering can be thought of as using the past to guide decisions, then the threshold filter uses the present, and the LCA filter uses a slice of the relative future.

4.1 Defending against Noisy Attack with the Threshold Filter

As detailed in section 3.1, the noisy attack consists of large number crafted `Invite` messages. To protected the softphone and its host, we want to filter out attack `Invite` messages while allowing legitimate `Invite` messages to pass.

Ideally the filter should be able to definitively distinguish attack packets from legitimate traffic. Given that the whole attack `Invite` message can be crafted and spoofed by the attacker, there is no meaningful content "pattern" or "signature" in the `Invite` message we can use to distinguish attack `Invite` messages from legitimate `Invite` messages. In addition, the attack `Invite` messages can spoof the source IP address of legitimate SIP servers. Therefore, we can not filter the noisy attack based on source IP either.

This leaves the arrival rate as the only usable detection factor—anything above a certain arrival rate indicates that an attack is more than likely occurring. How do we decide on that threshold level? While it can change from user to user, we can safely assume that even a heavy telephone user should not expect more than one to two legitimate phone calls per second. Any higher rate of incoming calls can not be manually handled by a human.

Figure 7(a) illustrates the filtering with threshold of one `Invite` message per second. For high rate (e.g., hundreds of attack `Invite` messages per second) noisy attack, threshold based filtering can be a very effective mitigation as it can filter out most, if not all attack `Invite` messages. However, threshold based filtering can still let very few attack `Invite` messages to reach the protected machine at the rate of no more than one attack `Invite` message per second as shown in figure 8(a). In addition, threshold based filtering has a very small chance to block legitimate `Invite` message while the high rate noisy attack is going on. This, however, is not a concern as the high rate noisy attack would have made the softphone unusable anyway.

If the noisy attack uses a rate of `Invite` messages less than one per second, it will not be effectively blocked by the threshold based filtering. However, it would take much longer time for such a low rate noisy attack to deplete the memory of the softphone host. In addition, keeping ringing the softphone during such extended period of time (e.g., several hours) would most likely alert the softphone user. Therefore, low rate noisy attack is likely to be stopped before it causes real problem to the softphone and its host.

The stealthy attack, on the other hand, would never ring the target softphone. This makes it unlikely to be noticed by the user of the targeted softphone. Since the stealthy attack has a much lower rate of `Invite` messages than the noisy attack, it can not be effectively filtered by the threshold based filter. Therefore, it is necessary to develop other defense against the stealthy attack.

4.2 Defending against Stealthy Attack with the LCA Filter

As detailed in section 3.2, the stealthy attack sends a low rate of cancelled `Invite` messages, consuming memory slowly while preventing the target softphone from ringing.

To filter such a stealthy attack with low rate `Invite` messages, we need to find some characteristics that can distinguish such stealthy attack from legitimate traffic. One unique characteristics of the stealthy attack is that every `Invite` message is followed by at least one `Cancel` message with matching Call-ID. This enables us to filter out the stealthy attack while allowing legitimate calls to go through.

Specifically, we introduce a queue to temporarily hold every incoming SIP message for a fixed period of time T. Such a queue of incoming SIP messages gives us a *limited context* from which we can ascertain the intentions of the incoming SIP messages by looking for any `Cancel`s associated with any `Invite` in the queue. As shown in figure 7(b), the queue consumer periodically checks the front packets in the queue. If the packet is an `Invite`, then the rest of the queue is searched for any `Cancel` message with a matching Call-ID). If any are found, then the `Invite` and the associated `Cancel`s are dropped. If any packet in the queue has waited for at least the required time of T, we pull it off the queue and forward it along the incoming path.

Note such a *limited context aware* filtering could indeed block legitimate `Invite` followed by legitimate `Cancel` caused by immediate hanging up after dialing the number. This is fine as the legitimate call has been cancelled by the caller already.

The period of time T for the queue needs to be long enough to catch all the associated `Cancel` that can prevent the previous `Invite` from ringing the callee's SIP phone. On the other hand, it can not be too long to interfere with legitimate SIP signaling. For Vonage, if an `Invite` from a proxy server is held for more than 1.5 seconds, then the proxy automatically sends a `Cancel`. Thus 1.5 seconds is the upper limit of T. For the lower time limit, we must consider that it is advantageous for the attacker to send `Cancel`s as soon as possible; the fewer they send and longer they wait, then the more likely that the softphone

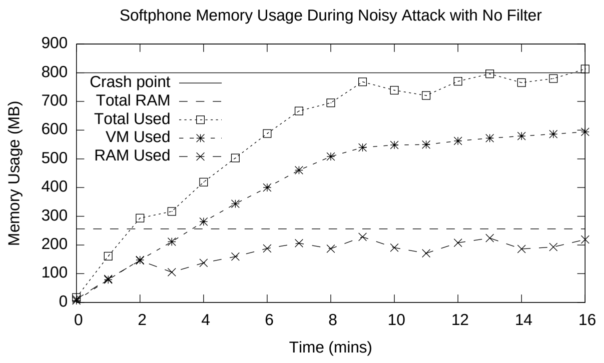


Fig. 9. Details of the escalating memory usage during the Noisy DoS attack without the Threshold filter.

might ring. In our limited context aware (LCA) filter, we use a one second wait time.

By searching for an associated `Cancel`, the LCA filter has a way to distinguish calls that should be allowed from those that shouldn't. This eliminates any dependency of the effectiveness on the arrival rate. As seen in figure 8(b), the LCA filter can effectively filter all the stealthy attack traffic while virtually never blocking legitimate traffic. The only theoretical false positive is when the (31 hexadecimal digits) Call-ID of a `Cancel` message from a stealthy attack happens to match that of a legitimate `Invite` message.

5 Experiments

To prove the concept of the attack and defense, we have implemented both the noisy and the stealthy attacks in Linux, the threshold based and LCA filters within the transparent network bridge in FreeBSD using divert sockets.

We have conducted two sets of experiments. In section 5.1 we demonstrate the effect of both noisy and stealthy attacks on the softphone host by measuring the memory usage caused by the attacks. In section 5.2 we analyze the effectiveness of the defense mechanisms, and present the overhead required to implement both filtering systems.

The target for these experiments is a virtual machine of 256 MB RAM running Windows XP and the X-PRO Vonage 2.0 softphone, release 1105x build stamp 17305. To avoid sending unnecessary outbound traffic, we have filtered out any replies related to our attack traffic.

5.1 Attacks

We created the initial `Invite` template from a PCAP trace of a legitimate call captured at the target's gateway. While this is the easiest solution given our setup, it is possible for the attacker to gather enough information through other means and create a template from scratch without a man-in-the-middle, or MITM. For instance, the proxy server IP addresses are relatively static if the target's geographical location is known and the attacker could scan the target's open ports to see which the softphone is using. This means that an attacker can execute this threat from anywhere in the world as long as they know the target's IP address.

The memory usage over the course of the noisy DoS attack is detailed in figure 9. Note that in the first minute there was a dramatic climb in the softphone's memory usage from 17MB to 161MB. Also, the processor utilization rose to 80–100% as soon as the flooding began. In nine minutes, the Vonage softphone has occupied over 500MB virtual memory and almost all 256MB RAM. By the sixteenth minute, as in shown in figure 6, the system has been resizing the swap file, the processor has become still swamped, and the UI has become completely unresponsive. The system did not return to usable state, even after several days, until rebooted.

Figure 11(b) shows the the memory usage of the Vonage softphone under the stealthy DoS attack. The memory usage growth under stealthy attack is much slower than the noisy DoS attack, and the processor utilization does not spike as soon as flooding begins. The end result is the same though, after roughly two hours, the Windows XP host will have no free memory and an UI that remains unresponsive until the system is rebooted.

For these attacks, the average memory allocated per `Invite` was 13KB. This is slightly higher, 49KB, during the first minute of the noisy attack. Considering that only the Call-ID had to be changed, and it has 31 hexadecimal digits, then this could theoretically consume even large memory systems.

5.2 Defense Mechanisms

In this section we analyze the effectiveness of the defense mechanisms against their associated attacks. Additionally, we present the maximum packet handling rate and the overhead of the filters. We have conducted experiments with various combinations of attacks and the filters: 1) threshold filter against noisy attack; 2) threshold filter against stealthy attack; 3) LCA filter against stealthy attack; and 4) combination of LCA filter and threshold filter (LCA filter first and threshold filter second) against stealthy attack. Since the LCA filter is only effective against stealthy attack, we do not test LCA filter against noisy attack.

Figure 10 shows the measured effectiveness, in term of attack packets blocked, of various filters against various attacks. Specifically, the threshold filter is able to block 99.8% of the noisy DoS attack and 95.6% of the stealthy attack at their maximum rate of `Invite`. This illustrates that the threshold filter's effectiveness decreases if the `Invite` arrival rate decreases. Note that the threshold filter can not block `Cancel`s, so only 15.2% of all attack traffic was blocked. In addition, the threshold filter has a slight chance to block legitimate `Invite` messages since it does not distinguish legitimate and illegitimate `Invite` messages. The LCA

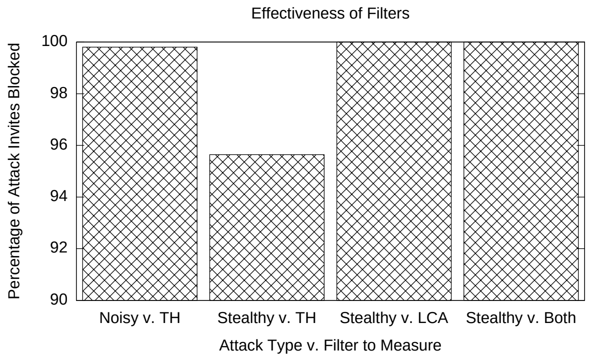
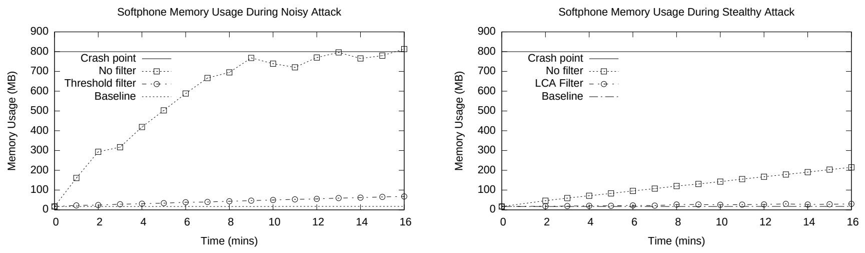


Fig. 10. Effectiveness of the filters against the various attacks measured by percent of attack INVITE messages blocked.



(a) Noisy attack with and without the Threshold filter.

(b) Stealthy attack with and without the LCA filter.

Fig. 11. Comparison of the escalating memory usage during the attacks with and without a filter.

filter, on the other hand, is able to block 100% the stealthy DoS attack at any capacity. In the LCA filter experiments, we mixed the legitimate `Invite` messages (from legitimate calls) with stealthy attack `Invite` messages. The LCA filter has never blocked any legitimate `Invite` messages mixed within the stealthy attack traffic. In summary, the threshold filter and LCA filter are complementing each other, and the combination of them are effective against both noisy and stealthy attacks.

We have measured the memory consumption caused by the noisy and stealthy attacks as well as the impact of the threshold based filtering and the LCA filtering. As seen in figure 11(a), the threshold filter effectively slows down the memory growth during a noisy attack. Without any filtering, noisy attack has consumed 813 MB of memory of the softphone host in 16 minutes. With threshold filtering, noisy attack has only occupied 67.7 MB in 16 minutes. Figure 11(b) compares the memory consumptions of the stealthy attack with and without the LCA filtering. The LCA filtering has successfully prevented the stealthy attack from occupying additional memory from the softphone host.

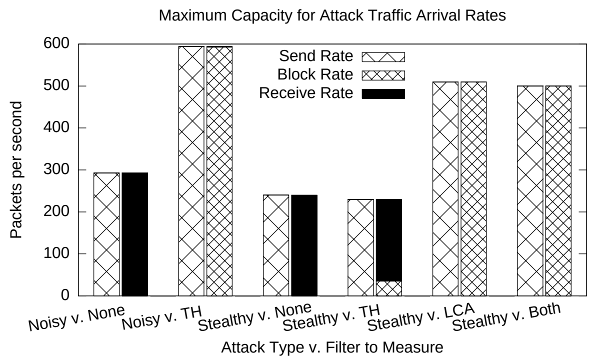


Fig. 12. An approximation of throughput for the filters under the DoS attacks.

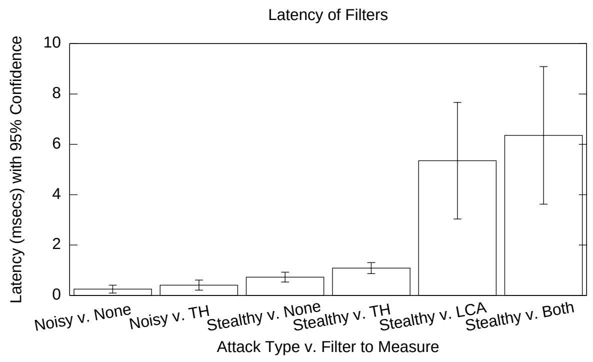


Fig. 13. Latency caused by the filters' packet processing under the DoS attacks with confidence intervals.

We have measured the maximum throughput (i.e., fastest packet rate without packet loss due to congestion) of both the noisy and stealthy attacks with and without threshold and LCA filtering. As shown in figure 12, noisy attack can achieve rate of about 300 `Invite` per second. Threshold filtering can handle rate close to 600 `Invite` per second. While maximum rate of stealthy attack is less than 250 `Invite` per second, LCA filtering can handle rate close to 500 `Invite` per second.

Per RFC 2544, we have measured the processing latency of our filters at the maximum throughput. As shown in figure 13, the threshold filter introduces no more than one millisecond processing delay when running against either the noisy or stealthy attack. The LCA filter adds less than five milliseconds processing delay in addition to one second queueing delay. Neither the one second queueing delay nor the processing delay has shown any noticeable impact on the normal VoIP signaling functionality in our experiments.

6 Discussion

The host disabling attacks we present in this paper are relatively straight forward to implement. In fact, since the softphone chooses a predictable port number, the attacker only needs to know the target's IP address. Our experience shows that when the Vonage softphone is behind a NAT enabled router, often the external mapping retains the internal port number. Therefore, to disable a targeted softphone behind NAT, the attacker only needs the public IP address of the targeted softphone, which can be easily obtained by observing any SIP traffic from the targeted softphone anywhere along its path. Since the process of spoofing SIP messages implies changing the source IP address, and optionally the caller ID information, there is an innate layer of stealth in the attack.

While host disabling attacks described in sections 3.1 and 3.2 are conducted against the Vonage softphone, they are actually exploiting features defined in the SIP protocol. Therefore, these attacks could be applicable to other softphones as well. It is one of our future works to investigate if other softphones are vulnerable to the newly identified attacks.

Besides our proposed network based mitigation, enforcing the SIP authentication on `Invite` messages from the SIP server to the SIP phone could help mitigate the newly identified attacks. To prevent the replay of authenticated `Invite` messages to the SIP phone, the SIP phone needs to challenge each authenticated `Invite` message to it. This would introduce substantial overhead to the SIP phone. To the best of our knowledge, no US residential VoIP service provider enforces the authentication of SIP messages from the SIP server to the SIP phone.

7 Related Work

Early VoIP security work primarily focused on analyzing the vulnerabilities and potential exploits of VoIP protocols and their implementations. McGann and Sicker [11] analyzed the potential security threats in SIP-based VoIP, creating an invisible listening post and modifying negotiation information on the fly. Geneiatakis et al. [4] surveyed the security vulnerabilities in SIP. Later work focused more on demonstrating exploits on current VoIP systems rather than illustrating potential exploits. Me and Verdone [12] detailed several VoIP attacks over insecure wireless networks. State [23] showed that it is possible to exploit the implementation vulnerabilities of a SIP stack in order to make any targeted GXV-3000 SIP phone accept calls from a remote attacker without ringing or user interaction. Zhang et al. [27] demonstrated that vulnerabilities in SIP can be used to launch billing attacks on currently deployed commercial VoIP services. Wang et al. [24] investigated the trust of several leading VoIP services (e.g., Vonage, AT&T) and showed that their VoIP calls can be transparently diverted and redirected—leading to voice pharming attacks on the VoIP users. It has been further detailed [26] that these call diversion attacks can be launched by a remote attacker who is not initially in the path of VoIP traffic of the target.

Lee et al. [10] showed how flooding attacks can cause constant ringing on the target UAC but did not discuss the effects on host resources. Herculea et al. [7] surveyed eight VoIP flooding attacks and arrival rates, but did not address host resource effects. Seedorf et al. [17] demonstrated single message attacks for eight SIP UAC implementations, but did not discuss remotely disabling the host. Kapravelos et al. [9] detailed arrival rates and cancelling a pending call, but did not claim that their attack is silent nor discussed its effects on the host. To the best of our knowledge, no previous work discusses how crafted SIP messages can be used to remotely disable a host that executes VoIP applications.

On the other side of the line significant work exists in securing VoIP. Geneiatakis et al. [5] surveyed SIP security mechanisms. In a later paper, Geneiatakis et al. [6] detailed memory usage of a SIP proxy under a flooding attack but did not discuss effects of such an attack on the UAC host. Additionally, they presented a bloom filter system to track call state in order to detect `Invite` floods. However, unlike our LCA filter, once a flood is detected their system can not distinguish legitimate packets. Reynolds and Ghosal [15] proposed a multi-protocol scheme to defend against flooding attacks on VoIP networks. Wu et al. [25] and Sengar et al. [18] presented using state information and cross-protocol correlation to detect denial-of-service attacks on VoIP. Sengar et al. [19] detailed a VoIP intrusion detection based interactive protocol state machine. Deng and Shore [2] proposed using nonces to protect the SIP servers from flooding attacks. Kapravelos et al. [9] presented a flow state based filter mechanism for preventing nuisance dialing but did not address false positive rates. Soupionis et al. [22] detailed an audio based anti-bot verification method and more formal methods later [20, 21]. In general, most existing VoIP defense mechanisms are designed to protect the VoIP servers, and be deployed close to them, rather than end hosts such as SIP softphones. Additionally, most existing detection methods are somewhat post-mortem and do not provide real-time flood protection while guaranteeing to keep legitimate VoIP calls alive against the attacks that we have described. To the best of our knowledge there is no existing work focused on securing against SIP messages that can remotely disable a host executing VoIP applications.

8 Conclusion

VoIP is a quickly growing dependence for modern communication needs. Unfortunately, its benefits come with an increased risk of attacks from the Internet. Most current exploits (e.g., billing attacks and remote eavesdropping attempts), however, have focused on the VoIP devices (e.g., servers, clients) or users. To the best of our knowledge, we are the first to demonstrate that an attacker can completely disable the very host that runs vulnerable VoIP applications.

We have developed two attacks that can disable a Windows XP computer running the official Vonage softphone. The noisy attack, which rings the targeted softphone, can disable the targeted Windows XP host within a few minutes. The stealthy attack, which never rings the targeted softphone, can completely disable the targeted Windows XP host within a couple of hours. Both versions can be launched from anywhere in the world as long as the target's IP address is known. Given the large subscriber base of VoIP and the ease of implementing such attacks, this is a viable threat to systems stability and security.

To mitigate the host disabling attacks, we have designed and evaluated two network based defense mechanisms. The threshold based filtering has very low overhead and can block 99.8% of the attack traffic at its maximum rate. However, threshold based filtering is not effective against slow and stealthy attacks and can block critical portions of legitimate VoIP traffic. Limited context aware (LCA) filtering can reliably filter all stealthy attack traffic while allowing virtually all legitimate traffic to pass in real-time. Therefore, the combination of threshold and LCA filters are effective against both the noisy attack and stealthy attack.

References

1. Arkko, J., Torvinen, V., Camarillo, G., Niemi, A., Haukka, T.: Security Mechanism Agreement for the SIP. RFC 3329 (Jan 2003)
2. Deng, X., Shore, M.: Advanced Flooding Attack on a SIP Server. In: Proc. of the Intl. Conf. on Availability, Reliability and Security (ARES). pp. 647–651. IEEE Computer Society (March 2009)
3. Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., Stewart, L.: HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 (Jun 1999)
4. Geneiatakis, D., Dagiouklas, A., Kambourakis, G., Lambrinoudakis, C., Gritzalis, S., Ehlert, S., Sisalem, D.: Survey of Security Vulnerabilities in Session Initiation Protocol. IEEE Commun. Surveys and Tutorials 8 (3), 68–81 (2006)
5. Geneiatakis, D., Kambourakis, G., Dagiuklas, T., Lambrinoudakis, C., Gritzalis, S.: SIP Security Mechanisms: A State-of-the-Art Review. In: Proc. of the 5th Intl. Netw. Conf. (INC). pp. 147–155. ACM (2005)
6. Geneiatakis, D., Vrakas, N., Lambrinoudakis, C.: Utilizing Bloom Filters for Detecting Flooding Attacks against SIP based Services. Computers & Security 28(7), 578 – 591 (2009)
7. Herculea, M., Blaga, T., Dobrota, V.: Evaluation of Security and Countermeasures for SIP-Based VoIP Architecture. pp. 30–34 (Aug 2008)
8. Jaques, R.: Cyber-Criminals Switch to VoIP 'Vishing', http://www.vnunet.com/vnunet/news/2160004/cyber-criminals-talk-voip
9. Kapravelos, A., Polakis, I., Athanasopoulos, E., Ioannidis, S., Markatos, E.: D(ei)aling with VoIP: Robust Prevention of DIAL Attacks. Proc. of the 15th European Symp. on Research in Computer Security (ESORICS) (2010)
10. Lee, C., Kim, H., Ko, K., Kim, J., Jeong, H.: A VoIP Traffic Monitoring System based on NetFlow v9. Intl. Journal of Advanced Science and Technology 4 (2009)
11. McGann, S., Sicker, D.C.: An Analysis of Security Threats and Tools in SIP-Based VoIP Systems. In: Proc. of the 2nd Workshop on Securing VoIP (Jun 2005)
12. Me, G., Verdone, D.: An Overview of Some Techniques to Exploit VoIP over WLAN. In: Proc. of 2006 Intl. Conf. on Digital Telecommun. (Aug 2006)
13. Moskalyuk, A.: US VoIP Market Shares (Aug 2006), http://blogs.zdnet.com/ITFacts/?p=11425
14. Now, V.: Vonage Is Still #1 In VoIP Market Share (Jul 2006), http://www.voipnow.org/2006/07/vonage_is_still.html

15. Reynolds, B., Ghosal, D.: Secure IP Telephony using Multi-layered Protection. In: Proc. of the 10th Netw. and Distrib. Syst. Security Symp. (NDSS) (Feb 2003)
16. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol. RFC 3261
17. Seedorf, J., Beckers, K., Huici, F.: Single-Message Denial-of-Service Attacks Against Voice-over-Internet Protocol Terminals. Intl. Journal of Electronic Security and Digital Forensics 2, 29–34(6) (2009)
18. Sengar, H., Wijesekera, D., Wang, H., Jajodia, S.: Denial of Service Attacks on IP Telephony. In: Proc. of the 14th IEEE Intl. Workshop on Quality of Service (IWQoS). IEEE Computer Society (Jun 2006)
19. Sengar, H., Wijesekera, D., Wang, H., Jajodia, S.: VoIP Intrusion Detection Through Interacting Protocol State Machines. In: Proc. of the Intl. Conf. on Dependable Syst. and Netw. (DSN). pp. 393–402. IEEE Computer Society (2006)
20. Soupionis, Y., Basagiannis, S., Katsaros, P., Gritzalis, D.: A Formally Verified Mechanism for Countering SPIT. In: Proc. of Conf. on Critical Information Infrastructures Security (CRITIS). pp. 128–139 (2010)
21. Soupionis, Y., Gritzalis, D.: ASPF: Adaptive anti-SPIT Policy-based Framework. In: Proc. of the Intl. Conf. on Availability, Reliability and Security (ARES). pp. 153–160 (2011)
22. Soupionis, Y., Tountas, G., Gritzalis, D.: Audio CAPTCHA for SIP-Based VoIP. In: Proc. of 24th IFIP Intl. Information Security Conf. (SEC). pp. 25–38 (2009)
23. State, R.: Remote eavesdropping with SIP Phone GXV-3000 (Aug 2007), `http://www.voipsa.org/pipermail/voipsec_voipsa.org/2007-August/002424.html`
24. Wang, X., Zhang, R., Yang, X., Jiang, X., Wijesekera, D.: Voice Pharming Attack and the Trust of VoIP. In: Proc. of the 4th Intl. Conf. on Security and Privacy in Commun. Netw. (SecureComm). pp. 1–11. ACM (2008)
25. Wu, Y.S., Bagchi, S., Garg, S., Singh, N., Tsai, T.: SCIDIVE: A Stateful and Cross Protocol Intrusion Detection Architecture for Voice-over-IP Environments. In: Proc. of the Intl. Conf. on Dependable Syst. and Netw. (DSN). pp. 433–442. IEEE Computer Society (Jul 2004)
26. Zhang, R., Wang, X., Farley, R., Yang, X., Jiang, X.: On the Feasibility of Launching the Man-in-the-Middle Attacks on VoIP from Remote Attackers. In: Proc. of the 4th Intl. Symp. on Information, Computer, and Commun. Security (ASIACCS). pp. 61–69. ACM (Mar 2009)
27. Zhang, R., Wang, X., Yang, X., Jiang., X.: Billing Attacks on SIP-Based VoIP Systems. In: Proc. of the 1st USENIX WOOT (Aug 2007)