ANGELOS D. KEROMYTIS

a look at VoIP vulnerabilities

Angelos Keromytis is an associate professor with the Department of Computer Science at Columbia University and head of the Network Security Lab. He is interested in all aspects of systems and network security. He received his Ph.D. from the University of Pennsylvania and his B.Sc. from the University of Crete.

VOICE OVER IP (VOIP) AND INTERNET Multimedia Subsystem (IMS) technologies offer higher flexibility than traditional telephony infrastructures and the potential for lower cost through equipment consolidation and new business models. In this article, I examine the current state of affairs on VoIP/IMS security through a survey of all the 221 known/disclosed security vulnerabilities in the Common Vulnerabilities and Exposures (CVE) database and in IETF RFCs/drafts. My key finding is that the higher complexity of VoIP/IMS systems leads to a variety of attack vectors, many of them caused by unforeseen and unexpected component interactions. A second finding is that what people seem to worry about in VoIP (traffic interception and impersonation) bears no resemblance to the distribution of vulnerabilities actually disclosed. The article concludes with some practical suggestions for securing VoIP systems.

VoIP/IMS refers to a class of products that enable advanced communication services over data networks. While voice is a key aspect in such products, video and other capabilities (e.g., collaborative editing, whiteboard sharing, calendaring) are supported. The key advantages of VoIP are flexibility and low cost. The former derives from the (generally) open architectures and software-based implementation, while the latter is due to new business models, equipment and network-link consolidation, and ubiquitous high-speed broadband connectivity.

As a result, VoIP has seen rapid uptake in both the enterprise and consumer markets. An increasing number of enterprises are replacing their internal phone switches with VoIP-based implementations, both to introduce new features and to eliminate redundant equipment. Consumers have embraced a slew of technologies with different features and costs, including P2P calling, Internet-to-phone network bridging, and wireless VoIP. These new technologies and business models are being promoted by a new generation of startup companies that are challenging the traditional status quo in telephony and personal telecommunications. As a result, a number of PSTN providers have already completed or are in the process of transitioning from circuit-switched networks to VoIP-friendly packet-switched backbones. Finally, as the com-

mercial and consumer sectors go, so do governments and militaries due to cost reduction concerns and the general dependence on commercial off-the-shelf (COTS) equipment for the majority of their IT needs.

Higher complexity is often the price we pay for more flexibility. You can find more details about the complexity found in VoIP in the paper this article is based on [1]. In brief, several factors contribute to architectural, protocol, implementation, and operational complexity:

- The number and complexity of the various features integrated in a product are perhaps the single largest source of complexity. For example, voice and video transmission typically allow for a variety of codecs that may be used in almost-arbitrary combinations.
- Openness and modularity, generally considered desirable traits, allow for a number of independent implementations and products. Each of these comes with its own parameters and design choices. Interoperability concerns and customer feedback then lead to an ever-growing baseline of supported features for all products. A compounding factor to increasing complexity for many of the open VoIP protocols is the "design-by-committee" syndrome, which typically leads to larger, more inclusive specifications than would be the case in the closed, proprietary wireline telephony network from 20 years ago.
- Because VoIP systems are meant to operate in a variety of environments, business settings, and network conditions, they must be highly configurable, increasing complexity. Of particular concern are unforeseen feature interactions and other emergent properties. These have often led to exposed systems through misconfiguration (or poorly understood configuration), as in the case of fraudsters who broke into Internet-accessible VoIP PBXs and routed long-distance calls through them at the expense of the PBX owners; this specific instance was estimated to have cost upwards of $5 million [5, 6]. Another case, enabled by the use of default passwords, was estimated to have cost $55 million [2].
- Finally, VoIP is intended to work over a public data network such as the Internet, or an enterprise/operator network that uses the same underlying technology. As a result, there is a substantial amount of (strictly speaking) non-VoIP infrastructure that is critical for the correct operation of the system, including DHCP, DNS, TFTP/BOOTP, NAT (and NAT traversal protocols such as STUN), NTP, SNMP, routing, the Web (HTTP, TLS/SSL, etc.), and many others. Even a "perfectly secure" VoIP system can be compromised by subverting elements of this infrastructure.

Because of this complexity, manifesting both in terms of configuration options and size of the code base for VoIP implementations, VoIP systems represent a large attack surface. Over time, we should expect to encounter security problems arising from design flaws (e.g., exploitable protocol weaknesses), undesirable feature interactions (e.g., combinations of components that enable new attacks or facilitate known attacks), unforeseen dependencies (e.g., compromised paths through seemingly unrelated protocols), weak configurations, and, not least, implementation flaws.

In trying to understand the threat space against VoIP, my approach is to place known vulnerabilities within a structured framework. While a single taxonomy is not likely to be definitive, using several different viewpoints and mapping the vulnerability space along several axes may reveal trends and areas that merit further analysis. As a starting point, I use the taxonomy provided by the Voice over IP Security Alliance (VoIPSA), available at http://www.voipsa.org/. VoIPSA is a vendor-neutral, not-for-profit organization composed of VoIP and security vendors, organizations, and individuals with an interest in securing VoIP protocols, products, and installations. The classification identifies six broad areas of concern: (1) social threats, (2) traffic

eavesdropping, interception, and modification threats, (3) denial of service (DoS), (4) service abuse, (5) physical access threats, and (6) interruption of services threats. Due to the nature of the vulnerabilities discussed, only the first four categories are relevant to our discussion. I also place the surveyed vulnerabilities within the traditional threat space of confidentiality, integrity, availability (CIA), and consider whether the vulnerabilities exploit bugs in the protocol, implementation, or system configuration.

Many of the vulnerabilities center on the Session Initiation Protocol (SIP), so it is worth highlighting some of its features and discussing its overall complexity. SIP is a protocol standardized by the Internet Engineering Task Force (IETF) and is designed to support the setup of bi-directional communication sessions, including, but not limited to, VoIP calls. It is similar in some ways to HTTP in that it is text-based, has a request-response structure, and even uses a mechanism based on the HTTP Digest Authentication for user authentication. However, it is an inherently stateful protocol that supports interaction with multiple network components (e.g., middleboxes such as PSTN bridges). While its finite state machine is seemingly simple, in practice it has become quite large and complicated—an observation supported by the fact that the main SIP document is the second largest RFC ever (after the encyclopedic "Internet Security Glossary," RFC 4949). Figure 1 shows the number of SIP-related RFCs (and the number of total bytes in these) per year (until May 2009), and a size comparison of the main SIP RFC with respect to the TCP RFC, the five main MIME RFCs, the two Secure MIME (S/MIME) RFCs, and the four main IPsec RFCs. These graphs should provide a quantitative, if indirect, indication of the complexity of SIP.

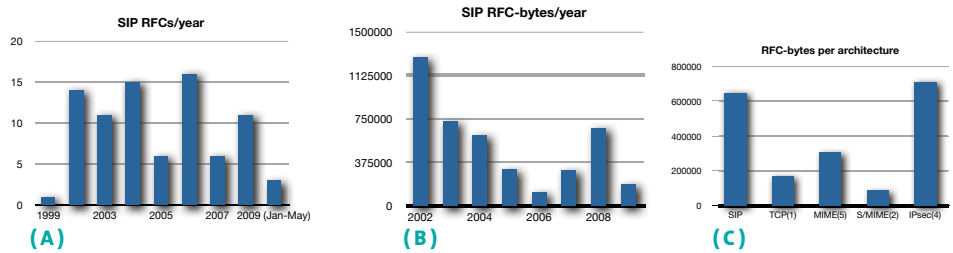

FIGURE 1: BREAKDOWN OF SIP-RELATED RFCS AND THEIR SIZES

For a complete reference to the vulnerabilities surveyed, see an online table at [3].

Overview of VoIP Vulnerabilities

Threats against VoIP system availability by exploiting implementation weaknesses are fairly common. Some implementations were shown to be vulnerable to crashes or hanging when given empty, malformed, or large numbers of SIP INVITE (or other) messages. It is worth noting that the same vulnerability may be present across similar protocols on the same platform and product due to code sharing and internal software structure, or to systems that need to understand VoIP protocols but are not nominally part of a VoIP system. The reason for the disproportionately large number of DoS vulnerabilities is due to the ease with which such failure can be diagnosed, especially when the bug is discovered through automated testing tools (e.g., fuzzers). Many of these vulnerabilities may be more serious than a simple crash and could possibly lead to remote code injection.

Unexpected interactions between different technologies used in VoIP systems can also lead to vulnerabilities. In some cases cross-site scripting (XSS) attacks were demonstrated against the administrator- and customer-facing management interface (which was Web-based) by injecting malicious

JavaScript in certain SIP messages, often through SQL injection vulnerabilities. The same vulnerability could also be used to commit toll fraud by targeting the underlying database. XSS attacks that are not Web-oriented have also been demonstrated, with one of the oldest VoIP-related vulnerabilities permitting shell command execution. Another Web-oriented attack vector is Cross Site Request Forgery (CSRF), whereby users visiting a malicious page can be induced to automatically (without user intervention, and often without any observable indications) perform some action on the Web servers (in this case, VoIP Web-based management interface) that their browser is already authenticated to.

The complexity of the SIP finite state machine has sometimes led to poor implementations. One vulnerability allowed attackers to convince a phone receiving a call to silently complete the call, which allowed the adversary to eavesdrop on the device's surroundings! The same vulnerability could be used to deny call reception at the target, since the device was already marked as busy. In other cases, it is unclear to developers what the use of a specific protocol field may be, in which case they may silently ignore it. Occasionally, such information is critical for the security of the protocol exchange, and omitting or not checking it allows adversaries to perform attacks such as man-in-the-middle or traffic interception, or to bypass authentication checks.

Since SIP devices are primarily software-driven, they are vulnerable to the same classes of vulnerabilities as other software. For example, buffer overflows are possible even against SIP "hardphones" and much more so for softphones, allowing adversaries to gain complete control of the device. Such vulnerabilities typically arise from a combination of poor (non-defensive) programming practices, insufficient testing, and the use of languages, such as C and C++, that support unsafe operations. Sometimes these vulnerabilities appear in software that is not directly used in VoIP but must be VoIP-aware, e.g., firewalls or protocol analyzers. It is also worth noting that these are not the only types of vulnerabilities that can lead to remote code execution. Other input validation failures can allow attackers to download arbitrary files from a user's machine or to place calls by supplying specially encoded URIs or other parameters.

Undocumented on-by-default features are another source of vulnerabilities. These are often remnants from testing and debugging during development that were not disabled when a product shipped. As a result, they often offer privileged access to services and data on a device that would not be otherwise available. One particularly interesting vulnerability allowed an attacker to place outgoing calls through the Web management interface.

A significant class of vulnerabilities in VoIP devices revolves around default configurations, in particular default usernames and passwords. Lists of default accounts are easy to find on the Internet via search engine. Users often do not change these settings; ironically, this seems to be particularly so for administrative accounts, which are rarely (if ever) used in the home/SOHO environment. Other default settings involve NTP servers and DNS servers.

Call interception vulnerabilities are a big concern with VoIP, given the plethora of tools for decoding video and audio streams and the ease of eavesdropping on network traffic, especially on the local subnet. Sometimes such vulnerabilities arise from strange protocol interactions and implementation decisions. For example, caching the location (address) of a VoIP phone based on the IP address used during boot time (using TFTP) seems a reasonable approach; however, since the boot and VoIP stacks are not necessarily tightly integrated, interaction with one protocol can have adverse effects (e.g., changing the perceived location of the phone) in the other protocol. Other

instances of such vulnerabilities involve improper/insufficient credential checking by the registrar or proxy or by the SNMP agent on the VoIP device, which can lead to traffic interception and user impersonation.

The integration of several capabilities in VoIP products, e.g., a Web server used for the management interface, can lead to vulnerabilities being imported to the VoIP environment that would not otherwise apply. In the specific example of an integrated Web server, directory traversal bugs or similar problems (such as lack of proper authentication in the Web interface) can allow adversaries to read arbitrary files or other information from the device. SIP components integrated with firewalls may also interact in undesirable ways. Improper handling of registration requests may allow attackers to receive messages intended for other users. Other such examples include failure to authenticate server certificates in wireless environments, enabling man-in-the-middle and eavesdropping attacks.

Some of the most serious non-implementation types of vulnerabilities are those where the specification permits exploitable behavior. For example, certain vendors permit the actual URI in a SIP INVITE call and the URI used in the Digest Authentication to differ; while arguably allowed by the spec, this enables toll fraud via credential reuse.

While rare, protocol-level vulnerabilities also exist. These represent either outright bugs in the specification or unforeseen interaction between different protocols or protocol components. For large, complicated protocols such as SIP and H.323, where components (code, messages, etc.) are semantically overloaded and reused, it is perhaps not surprising that such emergent properties exist. One good example is the relay attack in the SIP Digest Authentication [4], whereby an adversary can reuse another party's credentials to obtain unauthorized access to SIP or PSTN services (such as calling a premium or international phone line). This attack, depicted in Figure 2, is possible because authentication may be requested in response to an INVITE message at any time during a call, and the responder may issue an INVITE message during a call either automatically (because of timer expirations) or through a user action (e.g., placing the caller on hold to do a call transfer).

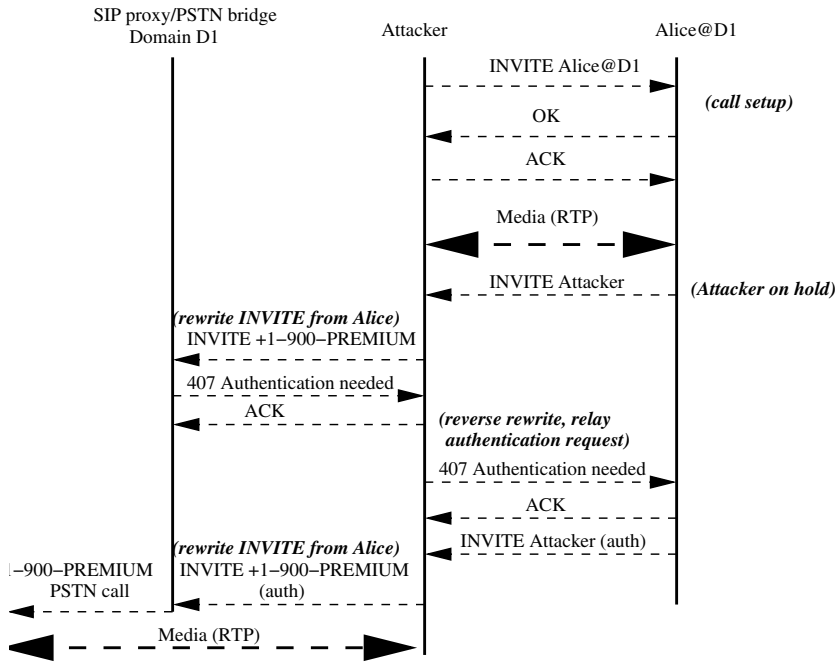

FIGURE 2: SIP RELAY ATTACK

DISCUSSION

I examined 221 vulnerabilities, 219 of which were disclosed in CVE and two as Internet drafts or RFCs. Figure 3 shows the reported number of vulnerabilities per year, up until approximately November 2009. The good news is that there appears to be a large drop in the number of *reported* vulnerabilities in the past two years. The reasons for this drop (and whether it will revert or not) are not known, which is reason enough for caution.

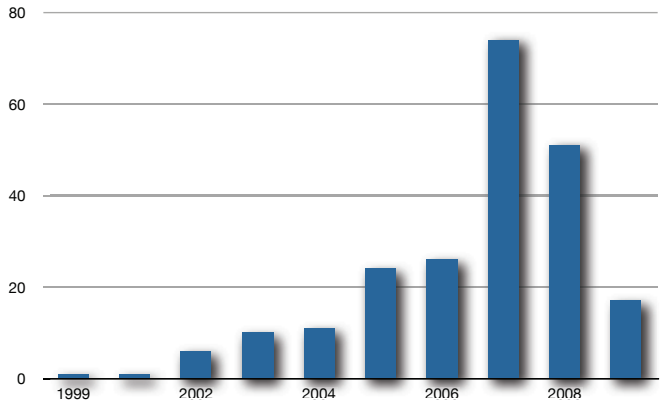

FIGURE 3: VULNERABILITIES PER YEAR

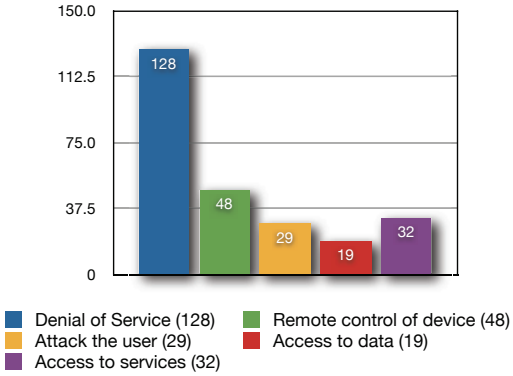

FIGURE 4: BREAKDOWN BY EFFECT

Looking at the vulnerabilities surveyed, a few patterns emerge. An informal classification of vulnerability effects is shown in Figure 4. Most categories are self-explanatory; "attack the user" refers to vulnerabilities that permit attackers to affect the user/administrator of a device, without necessarily compromising the system or getting access to its data or services. XSS attacks and traffic eavesdropping attacks fall in this category, whereas attacks that compromise state (data) resident on the system fall in the "access to data" category.

Half of the problems lead to a DoS in either an end-device (phone, softphone) or a server (proxy, registrar, etc.). This is not surprising, since DoS is easily diagnosed. In many cases, the problem was discovered by automated testing, such as protocol or software fuzzing; software failures are relatively easy to determine in such settings. Some of these vulnerabilities could in fact turn out to be more serious, e.g., a memory corruption leading to a crash could be exploitable in a code injection attack. The second largest class of vulnerabilities allow an adversary to control the device, whether by code injection, default passwords and services, or authentication failures. Note that a few of the vulnerabilities (approximately 10%) were counted more than once in this classification.

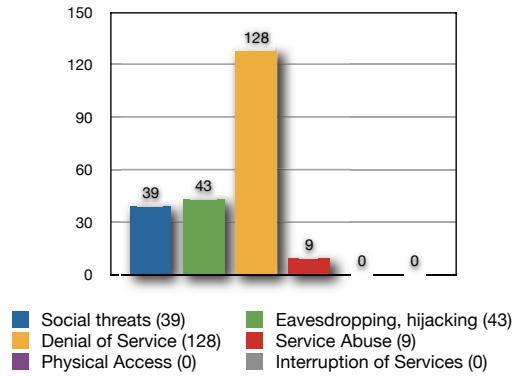

FIGURE 5: VULNERABILITY BREAKDOWN USING THE VOIPSA TAXONOMY

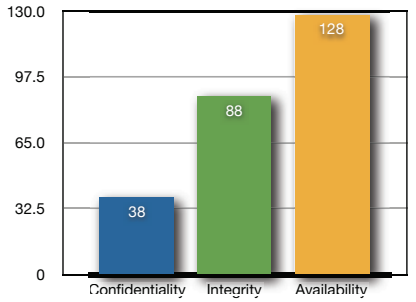

FIGURE 6: VULNERABILITY BREAKDOWN BASED ON (CONFIDENTIALITY, INTEGRITY, AVAILABILITY) CLASSIFICATION

The same pattern with respect to the predominance of DoS vulnerabilities holds when looking at the breakdown according to the VoIPSA taxonomy, shown in Figure 5. It should not be surprising that, given the nature of the vulnerabilities disclosed in CVE, there is no data on physical access and (accidental) interruption of services vulnerabilities. Furthermore, while "Access to services" was a non-negligible component in the previous breakdown, it represents only 4% here. The reason for this apparent discrepancy is in the different definitions of service: the specific element in the VoIPSA taxonomy refers to VoIP-specific abuse, whereas my informal definition covers lower-level system components which may not be usable in, for example, placing fraudulent calls. Another observation is that, while the VoIPSA taxonomy covers a broad spectrum of concerns for VoIP system designers and operators, its categories are perhaps too broad (and, in some cases, imprecise) to help characterize the types of bugs examined.

The vulnerability breakdown according to the traditional (Confidentiality, Integrity, Availability) security concerns again reflects the predominance of DoS threats against VoIP systems, as seen in Figure 6. However, Integrity violations (e.g., system compromise) are a sizable component of the threat space, while Confidentiality violations constitute only 15% of disclosed vulnerabilities. This represents an inversion of the perceived threats by users and administrators who, anecdotal evidence suggests, typically worry about such issues as call interception and eavesdropping.

Figure 7 shows the breakdown based on source of vulnerability. The overwhelming majority of reported problems arise from implementation issues, which should not be surprising given the nature of bug disclosure. Problems arising from configuration represented 11% of the total space, including such items as privileged services left on and default username/passwords. However, note that the true picture (i.e., what actually happens with deployed systems) is probably different in that configuration problems are

most likely undercounted: such problems are often site-specific and are not reported to bug-disclosure databases when discovered. On the other hand, implementation and protocol problems are prime candidates for disclosure. What is surprising is the presence of protocol vulnerabilities; one would expect that such problems would have been discovered and issued during protocol development, specification, and standardization. Their mere existence indicates high protocol complexity.

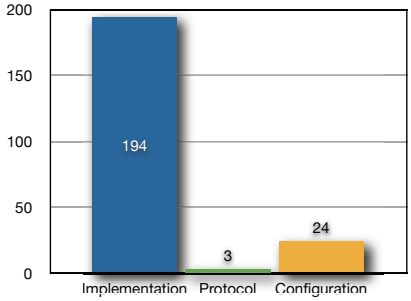

FIGURE 7: VULNERABILITY BREAKDOWN BASED ON SOURCE (IMPLEMENTATION, CONFIGURATION, PROTOCOL)

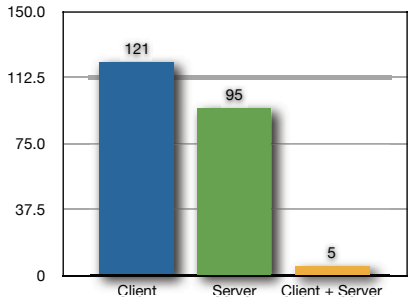

FIGURE 8: VULNERABILITIES PER PLATFORM

Finally, Figure 8 shows the breakdown of vulnerabilities based on the affected type of platform. In a few cases, typically when a bug was found in a software library, the vulnerability could be exploited in both clients and servers. Otherwise, vulnerabilities are equally distributed between the two primary types of VoIP platform. Although not shown here, the same holds when looking at specific classes of vulnerabilities (e.g., DoS).

Conclusions

The large majority of disclosed threats focused on DoS attacks based on implementation issues. While fault-tolerance techniques (such as replication) can be applied in the case of servers, it is less clear how to provide similar levels of protection at acceptable cost and usability to end-user devices. Unfortunately, the ease with which mass DoS attacks can be launched over the network against client devices means that they represent an attractive venue for attackers to achieve the same impact.

Code injection attacks in their various forms remain a problem, despite considerable progress in creating defenses. We need to do a better job at deploying and using these defenses where possible and in devising new techniques suitable for the constrained environments that some vulnerable VoIP devices represent.

Weak default configurations also present a big problem, as they do across a large class of consumer and enterprise products and software. The situation is likely to be much worse in the real world, considering the complexity of

securely configuring a system with as many components as VoIP. Vendors must make an effort to provide secure-by-default configurations, and to educate users how best to protect their systems. Administrators are in need of tools to analyze their existing configurations for vulnerabilities. While some tools dynamically test network components (e.g., firewalls), we need tools that work higher in the protocol and application stack. Furthermore, we need ways of validating configurations across multiple components and protocols.

Finally, there is simply no excuse for protocol-level vulnerabilities. While there exist techniques for analyzing and verifying security protocols, they do not seem to cope well with complexity. Aside from using such tools and continuing their development, protocol designers and standardization committees must consider the impact of their decisions on system implementers, i.e., whether a feature or aspect of the protocol is likely to be misunderstood and/or mis-implemented. Unfortunately, while simpler protocols are desirable, they seem incompatible with the trends we have observed in standardization bodies.

Network administrators can and must be proactive. Concrete steps to protect VoIP systems include but are not limited to:

- Stay current with firmware updates and security news about the devices deployed in your network. It is easy to overlook the fact that a VoIP hardphone may require a software update, just as servers and desktops do.
- Change the default/administrator authentication credentials in all devices and services! Make sure you cover all services running in each device (e.g., the Web-based management interface).
- Use any of a number of free or commercial SIP fuzzing tools, especially before initial roll-out of VoIP services, and after each firmware/software update. Do this both against end devices (hardphones and softphones) and servers.
- Make it your business to know what services each VoIP device is running. Do not trust the vendor to have produced a locked-down system—several misconfiguration-induced vulnerabilities came from leftover services running on the device. A simple port-scan will typically reveal such problems. If a service is not absolutely necessary, stop it; if that is not possible, block it at the firewall and complain to the vendor.
- Take steps to harden your VoIP servers. This may involve using obscure OS security features, or a different OS altogether. If possible, consider using a redundant server configuration with different operating systems running the same application server. (Using different application servers would be ideal, but impractical over the long run due to incompatibilities and configuration drift.) If you use server redundancy, make sure to test it periodically! There's nothing worse than discovering your secondary server is misconfigured while your primary server is compromised. (In reality, there are many things worse than this. Nonetheless, it is a very unpleasant situation.)
- Harden/protect the infrastructure on which your VoIP services rely. Specific services that merit attention include DNS, DHCP, and TFTP. This involves many of the steps mentioned above, for each of these services.
- Limit arbitrary access to VoIP devices. While this seems at odds with the basic premise of VoIP, it is possible to channel communications through media gateways. While this risks introducing some scalability problems, it also offers the opportunity to monitor traffic for abnormal behavior and to block some types of attacks against end devices. Along the same lines, you may also want to consider putting all your VoIP traffic into a different

VLAN, especially if VLAN port configurations can be frozen (admittedly a difficult proposition in many environments).

- When possible, enable TLS authentication and encryption for SIP signaling and use SRTP for media encryption. While the use of SRTP in particular is not widespread, the benefits appear to outweigh the (performance-related) drawbacks.

While there is no guarantee that the above steps will prevent a compromise (or that they are complete), they would have helped against most of the disclosed vulnerabilities we examined. The bottom line is that, while the situation with respect to VoIP security is currently bleak, there are steps you can take to protect your infrastructure today.

ACKNOWLEDGMENTS

The US National Science Foundation and the French National Research Agency supported this work under Grant CNS-09-14312 and Contract ANR-08-VERS-017, respectively.

REFERENCES

[1] A.D. Keromytis, "Voice over IP: Risks, Threats and Vulnerabilities," Cyber Infrastructure Protection (CIP) Conference, June 2009: http://www.cs.columbia.edu/~angelos/Papers/2009/cip.pdf.

[2] B. Krebs, "Security Fix: Default Passwords Led to $55 Million in Bogus Phone Charges," June 2009": http://voices.washingtonpost.com/securityfix/2009/06/default_passwords_led_to_55_mi.html.

[3] A.D. Keromytis, "Reference to Vulnerabilities Surveyed," February 2010: http://www.usenix.org/publications/login/2010-02/pdfs/keromytistables.html.

[4] R. State, O. Festor, H. Abdelanur, V. Pascual, J. Kuthan, R. Coeffic, J. Janak, and J. Floroiu, "SIP Digest Authentication Relay Attack: draft-state-sip-relay-attack-00," March 2009: http://tools.ietf.org/html/draft-state-sip-relay-attack-00.

[5] John Oates, *The Register*, "Two Charged with VoIP Fraud," June 2006: http://www.theregister.co.uk/2006/06/08/voip_fraudsters_nabbed/.

[6] Dan Goodin, *The Register*, "Fugitive VOIP Hacker Cuffed in Mexico," February 2009: http://www.theregister.co.uk/2009/02/11/fugitive_voip_hacker_arrested/.