HEISON CHAK

VoIP watch: security

Heison Chak is a system and network administrator at SOMA Networks. He focuses on network management and performance analysis of data and voice networks. Heison has been an active member of the Asterisk community since 2003.

heison@chak.ca

AS I WAS WAITING AT TORONTO Pearson Airport to board my flight, my early evening nap was interrupted by a familar sound—the default ring tone of a Cisco phone. Pearson is one of the airports that have taken the step to convert most (if not all) telephone communication to VoIP about three years ago. Today, Cisco IP handsets can be seen just about everywhere throughout the airport, from airline counters to information kiosks.

Although most VoIP implementations focus on voice quality, latency, and interoperability, the first question that comes to my mind is, How is security handled at this scale of deployment? In other words, how are confidentiality, availability, and integrity issues being addressed?

These state-of-the-art telephony systems promise to cut communication costs by carrying more voice calls than traditional switched circuit networks and enable enhanced services such as unified communications. However, as with traditional telephony, vulnerability to theft of service, denial of service attacks, and eavesdropping are all concerns for organizations deploying VoIP, and the consequences can be far more serious.

Confidentiality

As with data networks, VoIP security needs to be handled in a similar context, which may involve properly locking down servers and placing them behind firewalls, patching against vulnerabilities, and monitoring activities with intrusion-detection systems. Call detail records contain identity of callers and call patterns and should be treated with the same level of sensitivity as the actual content of a communication channel. Since voice travels in packets over IP networks, hackers can use data-sniffing and other hacking tools to carry out unauthorized wiretapping. It is possible to identify, modify, and play back voice traffic traversing such networks. For example, the vomit utility converts a conversation of a Cisco IP phone in G.711 (a codec) to a wave file that can be played back with a sound player.

```
$ vomit -r phone.dump | waveplay -S8000 -B16 -C1
```

Break-ins of a call manager host or soft switch that is directly accessible from the Internet or an

open network on a university campus could result in loss or compromise of sensitive data. Credit card numbers, social security numbers, and other important PINs entered during a phone call may end up in the wrong hands, allowing identity theft. A compromised gateway could turn into financial damages as a result of theft of use.

Availability

When designing VoIP networks, one should be aware that a VoIP packet stream exhibits behavior different from that of data packets. Although VoIP packets are small, they come in at a higher rate than do most data packets. Consider a regular data switch deployed in a VoIP network trying to handle tens or hundreds of VoIP devices communicating at 20-ms packetization interval (voice media separated into 20-ms frames for transmission); the switch can easily grind to a halt with high packet rates while utilization is still low. Buffers, echo canceller, and interface queues on routers and switches may also introduce additional delay, contributing to unpleasant conversations. Reducing hop count and increasing bandwidth may ease some of these delay issues. In the back office, when VoIP equipment is deployed alongside data equipment, one must size UPS and HVAC accordingly. Provisioning additional UPS power and runtime for soft switches and PoE capability will avoid an embarrassing situation should UPS power be overdrawn in a failover situation. The airflow in a small riser room may no longer be adequate for the VoIP PBX system. One of the biggest challenges in VoIP is providing telephony-like system uptime with general-purpose computer hardware and software. The discrete network elements like to advertise 4 or 5 9s of availability, but the ITU Telcordia estimates overall PSTN availability to be 99.94%. This metric also implies that 99.94% is the end-to-end requirement for VoIP to achieve PSTN equivalence.

Integrity

Voice packets should not be altered, callerID should reflect the true identity of a caller, and call detail records should be guarded with care, so that billing reports can be generated accurately. These requirements may sound reasonable and simple, but the fact is that they may be technically difficult to achieve. With NAT (Network Address Translation) and some widely used Layer 4 protocols (e.g., SIP and H.323), Layer 3 addresses can often be found in the wrong layer, making them difficult to deal with. For example, the "contact" address of a SIP packet originated from a host with an RFC1918 address behind a NAT firewall is not reachable from the Internet. An ALG (application layer gateway) and the middlebox solution are designed to disassemble the packet and replace the Layer 4 SDP (session description protocol) contact address of a SIP packet with a routable address of the edge router, such that return packets can be routed. It is obvious that such techniques violate the integrity of these VoIP packets, and it will continue to happen as long as the dominant VoIP protocol breaks NAT. There are many workarounds, yet the permanent fix is to avoid using NAT altogether and may be to go to IPv6 or use a protocol, such as IAX, that works well with NAT.

As with callerID, which was never really trusted in the PSTN world, emerging to VoIP makes it even easier to forge. The following Asterisk dial plan demonstrates how easy it is to alter callerID information. It sends a call to a SIP provider with callerID set to "Bill G" (you may be surprised to find out how many telephone companies actually pass the callerID

onward). In the extensions.conf configuration file used by Asterisk, changing the callerID is as simple as including a couple of lines:

```
_9.,1,SetCallerID(Bill G)
_9.,n,Dial(SIP/${provider}/${EXTEN:1})
```

Now What?

Some suggest signing, encrypting, and tunneling VoIP packets to ensure authenticity of callers, protecting all voice stream and touch-tone keystrokes, and working around the NAT problem. Since VoIP is susceptible to delay, having to sign every single packet and crypto overhead may introduce further delay to a VoIP packet in transit. Tunneling can also impact throughput, as additional header is required, worsening the header versus payload ratio (especially for efficient codecs, such as G.729). Header compression can ease the pain but may require custom work.

Typically, the one-way delay of a PSTN phone call is less than 150 ms. To maintain similar quality of voice over an IP network, there need to be algorithms that can perform tasks of signing and encrypting packets in a speedy fashion.

Delay Source (G.729)	*On-Net Budget (ms)*
Device sample capture	0.1
Encoding delay (Alg delay + processing)	17.5
Packetization/depacketization delay	20
Move to output queue/queue delay	0.5
Access uplink transmission delay	10
Backbone network transmission delay	latency
Access downlink transmission delay	10
Input queue to application	0.5
Jitter buffer	60
Decoder processing delay	2
Device playout delay	0.5
Total (one-way)	121.1+latency

Until such algorithms become available, we may need to weigh confidentiality against usability. To match the latency in PSTN of 150 ms, when the typical VoIP latency is already 121.1 ms (ignoring network latency), any algorithm used for encryption must be so fast as to be insignificant. Perhaps we should continue to rely on our own ears to authenticate a caller's voice until standards and the required infrastructure for authentication exist.