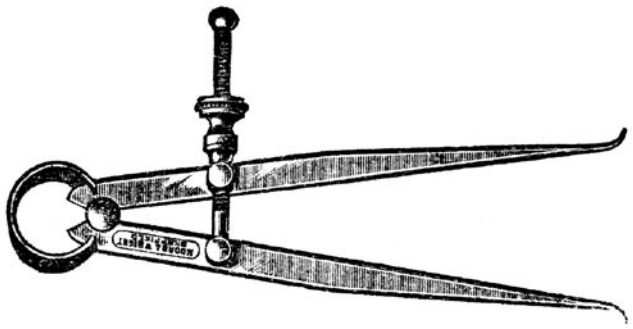
Seguridad en la Voz sobre IP – Protocolos SIP y RTP

Tobias Glemser, Reto Lorenz

Artículo publicado en el número 5/2005 de la revista *hakin9*. Os invitamos a la lectura de toda la revista.

Seguridad en la Voz sobre IP – Protocolos SIP y RTP

Tobias Glemser, Reto Lorenz



Antes del último CeBit en Marzo de 2005, la Voz sobre IP (Voice over IP, VoIP) ya era una de las palabras mágicas del mundo de las telecomunicaciones, y la gran esperanza de los proveedores y fabricantes. En los países donde hay buenas infraestructuras de red hay varias ofertas de paquetes VoIP. Las soluciones VoIP, tarde o temprano, sustituirán a la telefonía fija. Con tanta excitación, nadie se está dando cuenta de los peligros ocultos.

Hoy en día, la tecnología VoIP es una parte normal de las ofertas de acceso a Internet de banda ancha. Las llamadas gratuitas entre los usuarios de VoIP del mismo proveedor y las ofertas baratas para acceder a las redes clásicas de telefonía son otra de las razones del rápido éxito de esta tecnología. Pero no sólo en el mercado SOHO (*Small Office / Home Office*, Pequeña Oficina/ Oficina Doméstica) han descubierto las ventajas de VoIP. También las grandes empresas reconocen el potencial de estas tecnologías.

Pueden conectar las filiales de la empresa con un sólo cable que sirve tanto para voz como para datos. Los empleados, estén donde estén, son accesibles con un sólo número de teléfono, y al compartir la infraestructura, los costes de mantenimiento e inversión se reducen. Los problemas empiezan a tenerse en cuenta cuando ya se ha extendido esta tecnología (como siempre). Pero en un principio lo que presentan es una estrategia de migración brillante y unos servicios sobrevalorados.

Se ha discutido mucho sobre un caso, en el que una niña de trece años murió porque el número de emergencias norteamericano (911) no estaba operativo para la red de VoIP que utilizaba su madre. No existen regulaciones legales en torno a las llamadas de emergencia en las redes VoIP en la mayoría de los países – de hecho se están discutiendo en estos momentos.

A pesar de tales deficiencias organizativas, existen ya varios ataques contra la infraestructura técnica. Antes de examinarlos, tenemos que entender los parámetros básicos del protocolo SIP (*Session Initiation Protocol*, Protocolo de Inicio de Sesión). Nos quedaremos con SIP, ya que el desarrollo está claramente orientado hacia este protocolo en lugar de H.323. Pero olvidemos

En este artículo aprenderás...

- Conocimiento básico de los protocolos SIP,
- Algunas tácticas y posibles ataques reales contra usuarios y proveedores de VoIP.

Lo que deberías saber...

- Conocimiento básico de los protocolos de red,
- cómo realizar ataques en una red LAN utilizando el envenenamiento ARP,
- conocer la familia de los protocolos actuales de telecomunicación.

SIP – necesidades simples

Los paquetes SIP contienen parámetros para la configuración de la llamada inicial. El resto de parámetros, como los atributos para la conexión RTP – se incluyen en el Protocolo de Descripción de Sesiones (*Session Descritption Protocol*, SDP), que está integrado en el cuerpo de los mensajes SIP. Los paquetes SIP se dividen en paquetes de petición y de respuesta. Los mensajes se codifican según el estándar UTF-8. Por esa razón, pueden ser leídos directamente si no se utilizan mecanismos adicionales de seguridad.

Los mensajes SIP son muy similares a HTTP. Los campos de encabezado requeridos para cada mensaje de petición se muestran en la Tabla 1. A partir de estos elementos del protocolo, podemos ver que las definiciones del protocolo se usan para una comunicación relativa al contexto, aunque se envíe a través de un protocolo de transporte como UDP.

Tras la descripción de los componentes SIP, podemos estudiar los mensajes de petición (véase Tabla 2) – tenemos que diferenciar entre varios métodos. SIP puede ser mejorado con nuevos métodos de petición, así que sólo hablaremos de los más básicos (para ver otros más específicos, habrá que mirar los RFCs relevantes). Estos métodos y sus mensajes son una introducción. Nos muestran donde pueden producirse ataques de diversos tipos. Esta vez no vamos a estudiar los diferentes tipos de respuestas y sus usos.

Los mensajes se integran en el contexto de comunicación. Los segmentos de contexto se llaman *diálogos* y *transacciones*, y los *diálogos* pueden contener varias transacciones. Por ejemplo, una llamada VoIP es un diálogo SIP, compuesto por transacciones `INVITE`, `ACK` y `BYE`. Los agentes de usuario correspondientes deben ser capaces de almacenar el estatus de un diálogo por un período más largo de tiempo para ser capaces de generar mensajes con parámetros correctos.

A causa de los *diálogos*, hay varios parámetros, no sólo el `Call-ID`. Ya hemos hablado de *tag* y *branch*. Una discusión más profunda sobre estos parámetros no es ahora necesaria. Dejemos reflejado que las coherencias sistemáticas entre valores específicos del contexto y el comportamiento de los agentes de usuario no es tan transparente como otras definiciones SIP. Esta es la razón de que existan implementaciones poco seguras y llenas de errores, con agujeros de seguridad.

Después de una llamada exitosa a través del proxy SIP, la comunicación real de voz se hace a través de RTP. Utilizando el código intercambiado, los mensajes de voz se transfieren entre los dos sistemas (si la comunicación IP directa es posible), sólo se precisa el proxy-SIP para la finalización de la llamada.

las cosas que podemos hacer con este protocolo (véase el Recuadro *SIP – necesidades simples*). Lo importante es cómo pueden atacarse las infraestructuras VoIP y cómo podemos protegernos. La descripción de los ataques la haremos sobre un entorno VoIP basado en SIP como protocolo de señales. Los ataques se basan en métodos muy comunes. No vamos a tratar métodos contra una implementación concreta.

SIP y sus derivados

Cuando hablamos de comunicación, en el caso de VoIP tenemos que diferenciar entre diferentes protocolos, responsables del establecimiento y finalización de las llamadas. Uno ha de distinguir entre varios participantes para la señalización, transferencia de voz o mensajes de enlace. Al contrario que en la telefonía convencional, donde la comunicación se hace a través de un sólo cable desde el punto de vista del usuario, en VoIP hay una bifurcación en varios canales. Nombraremos los principales protocolos:

- para la señalización – SIP y SDP para el intercambio de propiedades de difusión,
- para el transporte – UDP, TCP, SCTP,
- para la difusión – RTP, sRTP, RTCP,
- para los enlaces – SIP, MGCP.

Estos protocolos nos dan las funciones básicas para el uso de VoIP, y se usan en cada vez más aplicaciones. Por supuesto, hay otros protocolos, pero centrarnos en todos ellos nos llevaría demasiado tiempo.

Para permitirnos evaluar mejor los ataques, tendremos que echar un vistazo a la configuración básica de las llamadas. Sólo usaremos un proxy SIP en nuestras prácticas. Este proxy forma parte de la señalización y del marcado. En la práctica, hay normalmente dos o más proxies SIP, especialmente si ambos participantes no están en el mismo entorno de red. En el caso de varios proxies, los mensajes SIP se intercambian entre ellos, así que habrá más capas de comunicación. Antes de entrar en detalles, la Figura 1 nos da una panorámica de estas características. Las capacidades de los protocolos utilizados no son especialmente novedosas. Al contrario, SIP utiliza técnicas muy conocidas – por ejemplo, utiliza elementos esenciales de HTTP. RTP fue definido hace casi 10 años, y su última actualización data del 2003.

Ataques SIP/ARP contra VoIP

Hay varios ataques, con requisitos diferentes. Vamos a aprender a usar nosotros mismos siete de los más extendidos, más efectivos y más discutidos.

La principal razón de la vulnerabilidad de VoIP, comparado con los simples teléfonos antiguos – *Plain Old Telephone Systems* (POTS), es el uso de medios compartidos. No hay una línea dedicada a la llamada, sino una red usada por múltiples usuarios y un montón de aplicaciones diferentes. Esto permite que el atacante se infiltre en nuestras comunicaciones – a través de ordenadores.

Pinchar llamadas telefónicas y reproducirlas en frente de nuestros socios es uno de los ataques más impresionantes contra VoIP. Tal y como describimos antes, la señalización se hace a través de un proxy SIP, la comunicación a través de una técnica peer-to-peer. En nuestro escenario queremos escuchar la conversación entre Alicia y Bob. Para esto, lanzaremos un ataque de *hombre en el medio* (*man in the middle* – MITM) utilizando el envene-

Tabla 1. *Campos del encabezamiento de una petición SIP*

Campo del Encabezamiento	Significado de la petición SIP
`Request-URI`	Contiene el método, el llamado `Request-URI` y la versión SIP utilizada. `Request-URI` consiste normalmente en la misma dirección, igual que el campo `To` (excepción: `REGISTER`-Method).
`To`	El destino de un mensaje y el método de enlace. El destino es un recipiente lógico, porque no está claro desde el principio que el mensaje llegue al destino. Dependiendo del contexto de comunicación puede incluirse un tag value.
`From`	Identificador lógico del origen de la petición. El campo `From` tiene que contener un tag value, elegido por el cliente.
`CSeq`	Command Sequence: permite la comprobación del orden de un mensaje dentro de una transacción. Consiste en un valor integral y un identificador del método de petición.
`Call-ID`	Un valor `Call-ID` debería ser no-recurrente, el mismo para todos los mensajes de un diálogo. Debería ser elegido por métodos criptográficos.
`Max-Forwards`	Este parámetro se utiliza para evitar situaciones de bucle. Si no hay criterios para un determinado valor, debería usarse 70.
`Via`	Este campo muestra el camino del transporte y el lugar a donde debería enviarse una respuesta. Este campo debe contener un branch value, no recurrente para el agente de usuario en cuestión. La `Branch-ID` siempre empieza con *z9hG4bK* y marca el principio de la transacción con esta petición.

Tabla 2. *Métodos del encabezamiento de una petición SIP*

Método	Instrucciones y explicación
`REGISTER`	Con este método, un cliente puede registrarse y des-registrarte desde un proxy. Estará listo y disponible para la comunicación VoIP. Para des-registrarse, el valor se pone en `0`.
`INVITE`	Es el método más importante, sin él no necesitaríamos SIP. Todos los métodos se subordinan a este método, aunque se usen en solitario. `INVITE` se usa para hacer nuevas llamadas.
`ACK`	Si una llamada (p.ej. una video-conferencia) se pone en marcha, esto se acredita por una petición `ACK` separada. Directamente tras esta petición, la conexión se inicia.
`BYE`	Este mensaje se usa para terminar una llamada de forma normal. Con él, una transacción establecida por medio de `INVITE` finaliza. Un mensaje `BYE` no es procesado sin el parámetro correcto de diálogo (`Call-ID` o *tag*).
`CANCEL`	Usando `CANCEL` una conexión establecida puede interrumpirse antes de establecer la llamada. También se usa en situaciones de error.
`OPTIONS`	Este método se usa para intercambiar los métodos de petición o el atributo de medios para la transmisión.
`NOTIFY`	Un método adicional definido en RFC 3265. Permite el intercambio de mensajes de estatus de un recurso al que se conecte el cliente. Por ejemplo, el cliente recibe notificación de nuevos mensajes de voz.

namiento ARP (véase el Recuadro *Ataque de Envenenamiento ARP*) para convencer al proxy así como a los teléfonos VoIP de Alicia y Bob, de que lo que quieren es comunicarse con nosotros y no entre ellos.

El esquema del pinchado – o *sniffing* de VoIP se muestra en la Figura 2. Primero, se prepara la llamada. Alicia envía la petición de llamada al proxy SIP. Este mensaje se intercepta y es el atacante quien lo envía. El proxy SIP trata de llegar a Bob para decirle que Alicia quiere comunicarse con él – este mensaje es interceptado y reenviado, a su vez. Tras una inicialización de llamada exitosa, la llamada (que utiliza el protocolo RTP) entre Alicia y Bob tiene lugar. Esta comunicación está siendo interceptada y reenviada por el atacante.

Si utilizas una herramienta como Ethereal para captar la comunicación, también recibirás los datos del flujo RTP. Para reproducirlos, puedes cargar los datos en un decodificador de voz como el analizador Firebird DND-323 o utilizar el propio Ethereal si las leyes G.711 U (PCMU) o G.711 A (PCMA) son las usadas como códec (estos son los estándares de codificación y decodificación de transmisiones telefónicas).

Una herramienta muy útil para la decodificación de voz, y para el envenenamiento ARP, es Cain & Abel (véase el Recuadro *En la Red*). Una vez iniciado, debes comprobar todos los hosts en tu sub-red (utilizando

Ataque de Envenenamiento ARP

El atacante envenena la tabla ARP de los sistemas a los que ataca. La tabla ARP sirve para convertir las direcciones lógicas IP en direcciones Layer 2 del modelo de referencia OSI (direcciones MAC de Ethernet). Prácticamente todos los sistemas operativos no protegidos aceptan respuestas ARP no solicitadas. El atacante rellena la tabla ARP con las direcciones IP que necesita, y pone su propia dirección MAC detrás de las IP's enviando algunas respuestas ARP. Reenvía cada paquete recibido al destinatario original, que también está envenenado. La comunicación funciona perfectamente, pero la intercepción no es reconocida por los que están hablando, a no ser que utilicen mecanismos criptográficos como TLS/SSL.

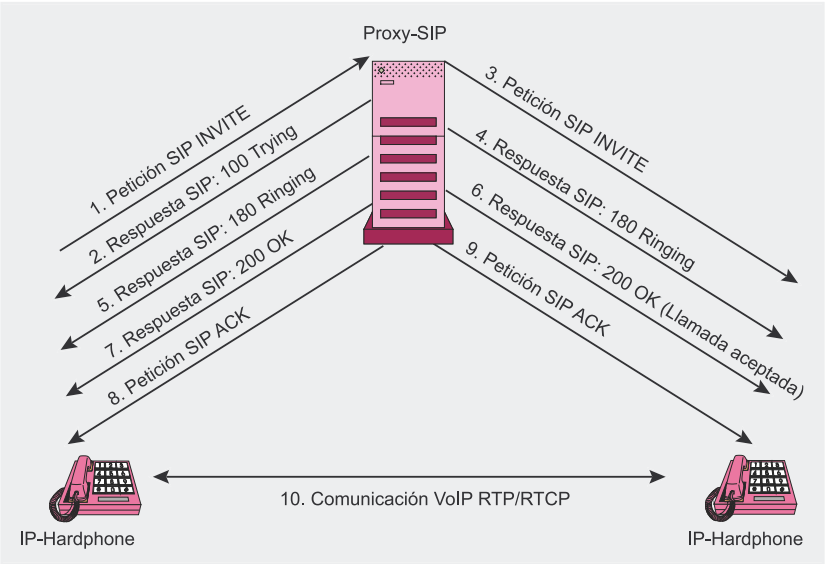


Figura 1. *Panorámica del establecimiento de una llamada utilizando SIP*

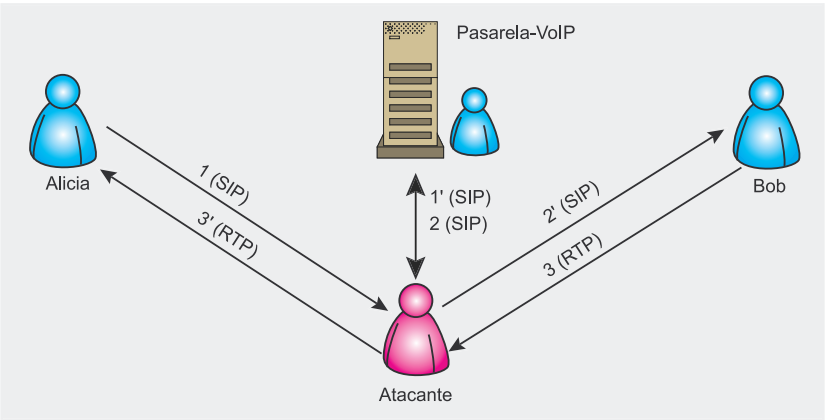


Figura 2. *Sniffing de VoIP*

peticiones ARP) haciendo click sobre el símbolo *plus*. Estos hosts pueden ser ahora vistos bajo la pestaña *Sniffer*, y pueden ser nuestras víctimas si así lo seleccionamos en la subpestaña *ARP*. En este caso, seleccionamos la dirección IP de Alicia, Bob y el proxy SIP. Luego pulsamos el botón de *Start/Stop ARP*, y el envenenamiento ARP comenzará. Sólo queda hacer una cosa – sentarnos y esperar. El resto lo hará Cain & Abel (véase Figura 3). Si se ha establecido una llamada, y esta ya ha finalizado, entre Alicia y Bob, la llamada se grabará directamente como un archivo WAV, y se mostrará en la pestaña *VoIP*. Puedes escucharlas con cualquier reproductor de audio. Por cierto: si mientras tanto se intercambia entre ellos alguna clave o password (por ejemplo POP3), podemos echarles un ojo en la pestaña *Passwords*.

Podemos ver que no hay problemas para atacar dentro de una red local, escuchar las comunicaciones y reproducirlas, ya que no hay mecanismos de seguridad adicionales establecidos.

Robo de identidades y rapto de registros

Normalmente, el registro en el proxy SIP se hace mediante un nombre de usuario y una contraseña. Los mensajes SIP, tal y como dijimos antes, no están encriptados. Si un atacante es capaz de captar el proceso de autenticación, por ejemplo mediante spoofing ARP, puede usar la combinación de usuario y contraseña recogida para entrar él mismo en el proxy SIP.

Un ataque como este no es posible ya en las implementaciones VoIP modernas. El proceso de autenticación (véase el Recuadro *Mecanismos de Seguridad en los Protocolos VoIP*), al igual que otras funciones importantes, utiliza autenticación resumida (*digest authentication*). Primero, el cliente se identifica frente al proxy SIP (véase Listado 1). El proxy rechaza la entrada enviando el mensaje de estatus *401 Unauthorized* (Listado 2) y envía una demanda al cliente para que se identifique a través de autenticación resumida. En la linea que empieza por *WWW-Authenticate* se envía el valor *nonce*, que debe ser aleatorio.

En el tercer paso (Listado 3), el cliente se re-identifica. Esta vez él también envía un mensaje *WWW-Authenticate*. Contiene su nombre de usuario, el entorno, y el valor nonce que ha enviado el servidor. Lo más importante es el valor de la respuesta. Usualmente, se trata de un *hash* MD5 construido sobre los valores de nombre de usuario, clave de acceso, el código nonce el método HTTP y el URI solicitado. El mensaje es procesado por un servidor que construye otro *hash* MD5, que consiste de los mismos parámetros. Si ambos hashes son iguales, la identificación es correcta y se certifica con un mensaje de estatus enviado por el servidor (véase Listado 4).

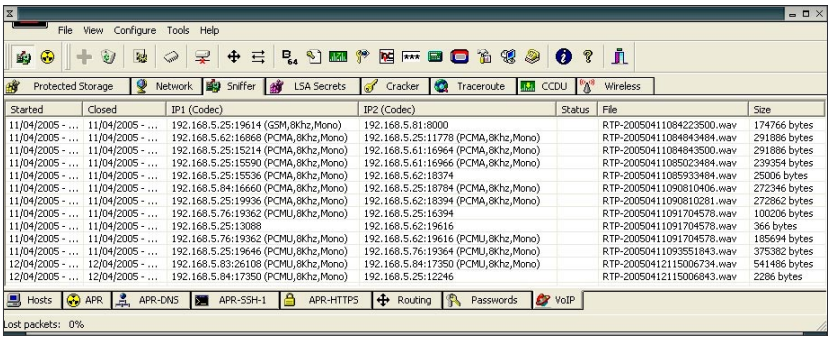


Figura 3. *Decodificación de voz con Cain & Abel*

El *hash* enviado en el paso 3 tiene dos características que evitan la autenticación falsa o la reutilización de un hash captado anteriormente: sólo es válido para el valor *nonce* generado aleatoriamente, y contiene el nombre del usuario y su clave de acceso como valores. Estos mecanismos hacen imposible que el atacante consiga descifrar la clave de acceso en un período de tiempo razonable.

DoS – Denial of Service (Negación del Servicio)

Como en cualquier sistema que ofrezca servicios, siempre puede echarse abajo el servidor si tienes suficiente ancho de banda. En el caso del proxy SIP, uno puede lanzar un ataque de *tormenta de registros* que sobrecargue el servicio. Además, pueden lanzarse ataques DoS sobre el servicio mismo, si el servicio es vulnerable. Es posible incluso acceder al servidor utilizando ataques de sobrecarga del buffer contra el servicio – este ataque era posible en 2003 con el servidor de código abierto PBX Asterisk (CAN-2003-0761). Debido a las debilidades a la hora de procesar parámetros con mensajes `MESSAGE` e `INFO`, un atacante podía lanzar comandos locales en el contexto del servicio asterisk – normalmente iniciado por el superusuario.

Derribar el servicio SIP por medio de mensajes inválidos depende de cada implementación. Si el servidor no tiene mecanismos que gestionen (o ignoren) los mensajes inválidos, puede caerse. (Para comprobar su comportamiento, existe un entorno de pruebas PROTOS basado en Java, que cada dueño de PBX (*Private Branch Exchange*) debe lanzar contra su sistema (véase el Recuadro *En la Red)*.

Un tipo diferente de ataque DoS es el llamado DoS soportado por el usuario. La Figura 4 nos muestra un mensaje UDP enviado al teléfono SIP con Login 14 e IP 192.168.5.84 desde el proxy SIP 192.168.5.25. Enviando este mensaje, el proxy (o el atacante) está señalizando que el usuario tiene nuevo correo de voz en su bandeja de entrada. Podemos reconocer esto echando un ojo al cuerpo del mensaje y al mensaje `Messages-Waiting: yes` así como a `Voice-Message: 1/0`. En lugar de un mensaje de voz, podría ser un mensaje de fax, por ejemplo. El primer dígito (`1` en este caso) indica cuantos mensajes nuevos se almacenan, el segundo (`0` en este caso) indica cuantos mensajes viejos se almacenan.

Como podéis ver, hemos editado este paquete. Esto puede hacerse fácilmente con Packetyzer (véase el Recuadro *En la Red*), una herramienta de Windows basada técnicamente en Ethereal. Cada paquete individual puede ser editado – se muestran los checksums incorrectos y pueden corregirse. Así que podemos enviar este mensaje a recipientes aleatorios; sólo necesitamos la IP y el nombre de usuario, que suele ser el mismo que el número de teléfono. Para dejar más claro que no se necesita más información ponemos todos los otros campos a `0`. Otros campos, como `User-Agent` no tienen importancia alguna.

¿Cuál es el problema de falsificar estos mensajes? ¿No contiene información sensible, o sí? La mayor parte de los teléfonos (probamos un Cisco 9750 y un Grandstream BT-100) procesan estos mensajes (además, incluso también aquellos con checksums erróneos) y los muestran al usuario. Normalmente empieza a parpadear un icono o toda la pantalla. Nuestro usuario – pensando que hay un mensaje nuevo, llamará a su buzón de mensajes. Como no se muestra ningún mensaje nuevo, el usuario pensará que es un pequeño fallo y lo ignorará. Al poco tiempo, la pantalla comenzará a parpadear de nuevo. Ahora el usuario llamará al servicio técnico, y va a ser muy divertido ver cómo buscan el fallo por todas partes, porque todo funciona

Mecanismos de Seguridad en los Protocolos VoIP

Además de los mecanismos relacionados con la comunicación relativa al contexto, hay otros mecanismos de seguridad en SIP, aunque no son obligatorios. Los mecanismos atañen principalmente a la autenticación y la seguridad criptográfica de la comunicación.

La autenticación se hace a través de varias vías: Un método común es la autenticación resumida. Es un simple método *challenge-response*, que puede usarse en todas las peticiones.

Otro método para asegurar los paquetes SIP es el uso del conocido S/MIME. Con S/MIME, el cuerpo de un mensaje SIP puede asegurarse con certificados S/MIME. El uso de S/MIME presupone un PKI establecido y la implementación de mecanismos necesarios para verificar los certificados. En el caso de SIP, S/MIME normalmente asegura los mensajes SDP. El uso de S/MIME requiere mucho tiempo y esfuerzo, si no hay estructuras establecidas.

Otros mecanismos requieren elementos adicionales de protocolo. Para SIP y RTP uno puede usar TLS. En el caso de la protección SIP, la protección se haría solo salto-a-salto, y uno no puede estar seguro de que la persona destino de la comunicación utilice un teléfono que incorpore TLS.

perfectamente. Realmente no hay ningún error.

Si el atacante empieza a mandar este tipo de mensaje a todos los usuarios de una red, tanto los usuarios como el soporte técnico van a pasar mucho tiempo buscando el error. Si manda el mensaje a muchísimos usuarios, y todo el mundo empieza a llamar a su buzón de correo, puede darse un pico de la demanda que incluso haga caer el servidor.

Interrupción de llamada

Mucha gente en otros artículos dice que el envío de un simple mensaje `BYE` a uno de los participantes en una llamada hará que ésta termine inmediatamente. No es tan fácil. En primer lugar, el atacante debe conocer el `Call-ID` del Call-Dialog. El RFC 3261 dice: *El campo de encabezamiento de Call-ID actúa como único identificador para agrupar una serie de mensajes. Debe ser el mismo para todas las peticiones y respuestas enviadas por cada agente de usuario en un diálogo.*

No hay leyes que obliguen a que el `Call-ID` tenga que ser construido a base de hashes, pero en la mayor parte de los casos, es así: `Call-IDs` aleatorios. Así, el atacante tiene que *pinchar* la inicialización de la llamada para poder terminarla usando el `Call-ID`. Si tiene que hacer esto, el contenido de la llamada será mucho más interesante que su simple finalización.

Phreaking

Lo que se conoce como phreaking, el clásico fraude de los teléfonos tradicionales, que se hacía anteriormente enviando tonos especiales del sistema desde cabinas telefónicas públicas, puede volver a renacer. Como el flujo de voz (RTP) y la señalización (SIP) vuelven a estar juntos, este escenario será posible, aunque todavía no se haya conseguido a día de hoy.

Un cliente preparado prepara una llamada a otro cliente preparado. Ambos se conectan vía un proxy SIP y se comportan de manera normal.

Listado 1. *Registro SIP Fase 1 (cliente a proxy SIP)*

```
REGISTER sip:sip.example.com SIP/2.0
Via: SIP/2.0/UDP 10.10.10.1:5060;rport; ←
   branch=z9hG4bKBA66B9816CE44C848BC1DEDF0C52F1FD
From: Tobias Glemser <sip:123456@sip.example.com>;tag=1304509056
To: Tobias Glemser <sip:123456@sip.example.com>
Contact: "Tobias Glemser" <sip:123456@10.10.10.1:5060>
Call-ID: 2FB73E1760144FC0978876D9D69AE254@sip.example.com
CSeq: 20187 REGISTER
Expires: 1800
Max-Forwards: 70
User-Agent: X-Lite
Content-Length: 0
```

Listado 2. *Registro SIP Fase 2 (proxy a cliente) – rechazo*

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 10.10.10.1:5060;rport=58949; ←
   branch=z9hG4bKBA66B9816CE44C848BC1DEDF0C52F1FD
From: Tobias Glemser <sip:123456@sip.example.com>;tag=1304509056
To: Tobias Glemser <sip:123456@sip.example.com>; ←
   tag=b11cb9bb270104b49a99a995b8c68544.a415
Call-ID: 2FB73E1760144FC0978876D9D69AE254@sip.example.com
CSeq: 20187 REGISTER
WWW-Authenticate: Digest realm="sip.example.com", ←
  nonce="42b17a71cf370bb10e0e2b42dec314e65fd2c2c0"
Server: sip.example.com ser
Content-Length: 0
```

Listado 3. *Registro SIP Fase 3 (cliente a proxy) – re-autenticación*

```
REGISTER sip:sip.example.com SIP/2.0
Via: SIP/2.0/UDP 10.10.10.1:5060;rport; ←
   branch=z9hG4bK913D93CF77A5425D9822FB1E47DF7792
From: Tobias Glemser <sip:123456@sip.example.com>;tag=1304509056
To: Tobias Glemser <sip:123456@sip.example.com>
Contact: "Tobias Glemser" <sip:123456@10.10.10.1:5060>
Call-ID: 2FB73E1760144FC0978876D9D69AE254@sipgate.de
CSeq: 20188 REGISTER
Expires: 1800
Authorization: Digest username="123456",realm="sip.example.com", ←
  nonce="42b17a71cf370bb10e0e2b42dec314e65fd2c2c0", ←
  response="bef6c7346eb181ad8b46949eba5c16b8",uri="sip:sip.example.com"
Max-Forwards: 70
User-Agent: X-Lite
Content-Length: 0
```

Listado 4. *Registro SIP Fase 4 (proxy a cliente) – éxito*

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.10.10.1:5060;rport=58949; ←
   branch=z9hG4bK913D93CF77A5425D9822FB1E47DF7792
From: Tobias Glemser <sip:123456@sip.example.com>;tag=1304509056
To: Tobias Glemser <sip:1888819@sipgate.de>; ←
   tag=b11cb9bb270104b49a99a995b8c68544.017a
Call-ID: 2FB73E1760144FC0978876D9D69AE254@sip.example.com
CSeq: 20188 REGISTER
Contact: <sip:123456@10.10.10.1:5060>;q=0.00;expires=1800
Server: sip.example.com ser
Content-Length: 0
```

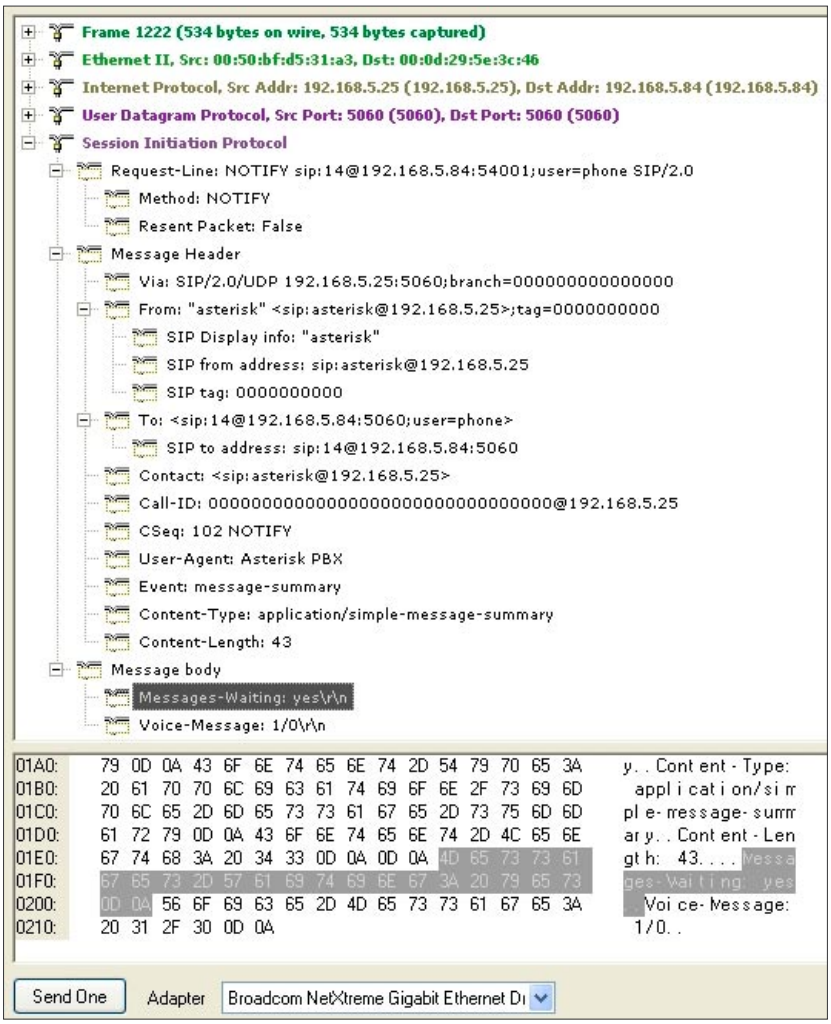


Figura 4. *Un paquete SIP modificado*

Directamente tras el establecimiento de llamada, el proxy recibe una señal para la terminación, que está certificada por los dos clientes. Pero ellos no se desconectan, la llamada no termina – y el servidor SIP no se da cuenta.

Si ambos clientes están en la misma subred, la llamada no terminará en ningún caso, por el flujo de voz P2P. Si hay una salida a través del proxy SIP (por ejemplo; Si se conecta a otra red), la comunicación RTP será enrutada por el proxy. Ahora tendrá que terminar el flujo RTP por sí mismo. El proxy tiene que darse cuenta de que vía SIP se ha enviado una señal de finalización de llamada y transfiere esta información directamente al control de las comunicaciones RTP.

Otro posible ataque phreaking depende de la implementación del proxy SIP. En algunos casos, como en la versión actual de Asterisk, se requiere una re-autenticación vía autenticación resumida como hemos visto en los listados 1–4 para prácticamente todas las comunicaciones cliente – servidor. Otras implementaciones requieren re-autenticación sólo tras un cierto período de tiempo. Así, en el siguiente escenario es posible generar gastos para el proveedor.

Un atacante envía un mensaje `INVITE` válido al proxy SIP, utilizando las credenciales de un usuario que ya se ha autenticado con éxito. El proxy SIP inicializa la llamada. Los paquetes restantes necesarios para la inicialización pueden ser enviados por el atacante – mediante un proceso automático sin saber los paquetes de respuesta del servidor. Algunos sistemas de teléfonos especiales de servicios preven increíbles cantidades para una llamada – con independencia de la duración de la misma. Así, sería posible para un atacante provocar grandes pérdidas con llamadas cortas a cargo de otros usuarios.

SPIT (Spam over IP Telephone) – Spam sobre telefonía IP

SPIT es uno de los peligros más mencionados para VoIP – un atacante envía mensajes de voz parecidos al correo basura. Al contrario de las llamadas desde robots en el mundo de POTS, las llamadas VoIP no generan costes. Al igual que el correo basura tradicional, el *spitter* utiliza la dirección de la víctima, en este caso no su dirección de correo electrónico, sino su dirección SIP. En un contexto de expansión de la telefonía IP sólo es cuestión de tiempo conseguir muchas direcciones SIP válidas, especialmente si se utilizan libretas de direcciones centrales.

El spitter llama a un número SIP, el proxy de la víctima procesa esta llamada y la víctima tiene que escuchar la basura del spitter, tal como el tamaño mínimo del miembro viril. Como el *spammer*, el spitter sólo necesita una cosa – ancho de banda. De hecho, los mensajes de voz consumen más recursos que

En la Red

- *http://www.ee.oulu.fi/research/ouspg/protos/testing/c07/sip/* – PROTOS Test Suite,
- *http://www.ethereal.com* – El sniffer de red Ethereal,
- *http://www.packetyzer.com* – Packetyzer – Sniffer TCP/IP Windows basado en Ethereal,
- *http://www.asterisk.org* – Asterisk, el PBX de código abierto,
- *http://www.oxid.it* – Cain & Abel.

Sobre los autores

Ambos autores trabajan como asesores de seguridad IT. Tobias Glemser trabaja en Tele-Consulting GmbH Alemania (*http://www.tele-consulting.com*), desde hace más de 4 años, mientras que Reto Lorenz es uno de sus directores de negocio.

los e-mails. Un mensaje de 15 segundos (ninguna víctima resistirá más de eso), tendrá un tamaño de unos 120 kBytes utilizando un códec de 64 kbit/s. Utilizando troyanos – igual que el spam, un usuario desprotegido de Internet podría sufrir el abuso de enviar SPIT a través de su ancho de banda.

Diallers o Marcadores

El uso de marcadores (o *diallers*), que cayó en picado con la popularización de tecnologías que no necesitan marcado, como DSL o cable, puede volver a ser una amenaza. Por el uso de un cliente SIP, tenemos el mismo escenario de un dialler clásico que utiliza el módem o la linea RDSI/ISDN para llamar a ciertos números premium. Por ejemplo, un dialler puede infectar un cliente SIP e instalar un número como si fuera un prefijo, o introducir un proxy nuevo, mucho más caro. Las llamadas pasarán a través de estos números caros sin que el usuario se dé cuenta, hasta que llegue la primera factura. Sólo es cuestión de tiempo que los diallers vuelvan a aparecer en la escena.

Conclusión

No hay duda de que VoIP es una de las técnicas más atractivas de los últimos años, y están apareciendo aplicaciones de uso masivo en Internet, pero también en las redes privadas de empresas. Si vemos lo que se dice en los medios sobre los problemas de seguridad de VoIP, podríamos pensar que la pareja SIP y RTP es algo realmente débil. Hay que pensar sobre los problemas de seguridad antes de instalar cualquier tecnología.

Por lo que podemos ver, se conocen muchos tipos de ataque – que no son otra cosa que ataques al protocolo IP ligeramente modificados. Los ataques contra SIP/RTP son posibles por lo general en las estructuras LAN siempre que no se encripten las comunicaciones. Por ejemplo, a través de un simple pinchado de los flujos RTP. Este ataque es idéntico al pinchado de una comunicación TCP/IP. La mayor parte de los otros ataques sólo son posibles si el proxy SIP o el agente de usuario (UAC, *User Agent Client*) no procesa correctamente el `Call-ID` o si el atacante pincha el `Call-ID`. La amenaza es mayor cuando no se requiere autenticación resumida para cada acción relevante, pero el problema real es SPIT, si hablamos de dinero, ya que podemos estar seguros de que todos estos anunciantes van a utilizarlo. ■