UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

CONTROL DE ACCESO A RED INALÁMBRICA (WNAC) UTILIZANDO SOFTWARE LIBRE Y BIOMETRÍA

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRA EN INGENIERÍA

P R E S E N T A:

CINTHYA ENETZY OLMOS ROA

DIRECTOR DE LA TESIS: DR. ENRIQUE DALTABUIT GODAS

MÉXICO, D.F. **2010.**

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A Íker, que es la inspiración de mi vida y mi motivo de seguir creciendo.

A Omar, por creer en mí y alentarme a continuar.

A mi madre Andrea, que es mi ejemplo a seguir y a mis tías Ana y Elvia que siempre están ahí cuando las necesito.

A Yuri, por ser incondicional y estar a mi lado.

Al Dr. Enrique Daltabuit, por todos los conocimientos que me brindó y por mostrarme el fascinante mundo de la Seguridad de la Información.

A Sergio Castro, por su gran apoyo moral y académico en el transcurso de este proyecto.

Tabla de contenido

1. Introducción

1.1. Motivación

Dentro de la historia de la computación se han tenido grandes avances hasta llegar a nuestros días, donde la interconexión de computadoras es indispensable. Cada vez que se hace uso de una red, ya sea una red local o Internet, formamos parte de un gran sistema teniendo grandes beneficios, pero sin darnos cuenta que tan vulnerables se vuelven nuestras computadoras. Hasta que oímos de ataques famosos donde se perdió información, donde vaciaron cuentas bancarias usando solamente la red, donde se difama algún gobierno o alguna celebridad. Y a pesar de saber que no estamos seguros al utilizar una red, no se ha dado la importancia suficiente en nuestro país, ya que se sigue pensando que nuestras redes no son un objetivo interesante para los atacantes (1).

La Seguridad de la Información toma un papel importante en lo que respecta al uso de la computación y su campo ha crecido y evolucionado considerablemente en los últimos años, convirtiéndose en una carrera acreditada a nivel mundial, ofreciendo muchas áreas de especialización. Es por ello importante desarrollar temas de investigación en dicho campo, para que nuestro país no se quede rezagado.

La Seguridad de la Información tiene como objetivo reducir los riesgos en aspectos del comportamiento (administrativos, organizacionales, reglamentos, leyes), como en los dispositivos que manejan la información (redes, líneas de transmisión, conexiones, discos) y en los lugares físicos donde se mantiene o procesa la información (edificios, laboratorios).

Los principios básicos que la Seguridad de la Información defiende son Confidencialidad, Integridad, Disponibilidad y No repudio; y para poder garantizar estos principios es necesario conocer quién tiene acceso a nuestros sistemas, para saber quién puede conocer o no la información, quién puede cambiar o borrar la información, por cuánto tiempo se puede acceder a los recursos y verificar que sea quién dice ser.

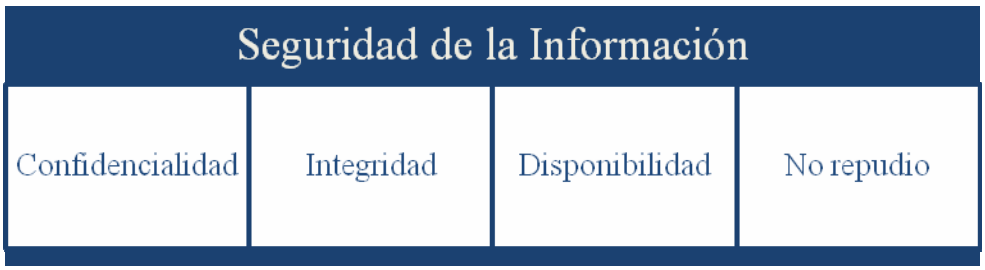


Fig. 1. Principios básicos de la Seguridad de la Información

Por lo anterior se puede decir que el control de acceso es el primer paso para la Seguridad de la Información, de ahí que se deba conocer quién se conecta para decidir a qué servicios puede acceder.

Actualmente se ha incrementado el uso de redes, y el control de acceso a dichas redes normalmente se basa solamente en el identificador de acceso a la red (dirección MAC) o nombre de usuario y contraseña. Sin embargo, este esquema no es completamente seguro.

Es necesario saber si los dispositivos que se conectan cumplen con las políticas de la empresa, si tienen los programas de seguridad necesarios, como antivirus, o que no tengan instaladas ciertas aplicaciones. Ya dentro de la red, un usuario autenticado puede cambiar la configuración de los dispositivos y de esta manera ya no adecuarse a las políticas de la empresa, haciendo posible que éste lance algún ataque interno o realice actividades que no le correspondan. También existe el problema de los ataques de Día Cero (Zero Day), donde no se tienen parches contra vulnerabilidades o los antivirus todavía no tienen la firma del virus, por lo que es muy difícil prevenirlos.

Es importante conocer la identidad del usuario para poder asociarla con su actividad dentro de la red, ya que el control de acceso se basa en quién eres para decidir a qué puedes acceder.

1.2. Objetivo

El objetivo principal de este proyecto es aumentar el nivel de seguridad del manejo de usuarios dentro de la red, a través de la implementación de la arquitectura TNC, junto con el manejo de datos biométricos dentro del proceso de autenticación, utilizando herramientas de software libre, bajo el sistema operativo Linux, y el manejo de VLANs.

Para alcanzar este objetivo es necesario tener en cuenta que uno de los problemas en el control de acceso a una red, tanto cableada como inalámbrica, es cuánto se confía en un dispositivo y quién es el usuario que desea conectarse. Es muy probable que dicho dispositivo se encuentre infectado con virus o programas maliciosos por conectarse directamente a Internet sin ninguna protección o que el usuario no sea quien dice ser.

Este problema intenta solucionarse con lo que se denomina Network Access Control, NAC, el cual se basa en la integridad de los elementos activos, donde se verifica el estado de seguridad del dispositivo antes de dejarlo conectarse a la red, además de autenticar al usuario y monitorear su sesión durante el período de tiempo que se mantenga conectado. Los pasos principales para el proceso de NAC son: Evaluación, Aislamiento, Remedio, Monitoreo.

Existen varios fabricantes y desarrolladores de software que han entrado en el mercado de NAC. Sin embargo, el grupo Trusted Computing Group (1) bajo el subgrupo Trusted Network Connect, ha desarrollado una especificación pública que no se basa en ningún producto en especial.

La arquitectura de este grupo toma varios conceptos y protocolos de seguridad ya existentes, haciendo más fácil su integración con la infraestructura. Por ejemplo, es posible utilizar los protocolos IEEE 802.1X, EAP y RADIUS para la autenticación de los usuarios.

Una de las ventajas de NAC es que no solamente se basa en quién es el usuario para decidir a qué puede acceder, esto implica considerar tanto la identidad del usuario como la evaluación del estado de seguridad del dispositivo y el ambiente de red.

Esto nos lleva a aumentar el nivel de seguridad en la autenticación del usuario, por lo que se recomienda utilizar al menos dos tipos de credenciales para verificar su identidad, como son la contraseña y algún dato biométrico. Es por eso que en este proyecto se hará uso de la biometría en el proceso de autenticación del usuario (3).

2 Antecedentes

2.1. Tecnologías a utilizar en la arquitectura

Las tecnologías de seguridad principales que se utilizan dentro de la arquitectura, en la que se basa este proyecto, son el método IEEE 802.1X, junto con el protocolo EAP y el protocolo RADIUS, que a continuación se describen brevemente.

2.1.1. 802.1X

Uno de los métodos en el campo de seguridad de la información que se utiliza en NAC, es el conocido como IEEE 802.1X, el cual es un método general para el control de acceso a redes basado en puertos, definido sobre la Capa 2 del modelo OSI. Dicho método permite autenticar a un cliente cuando se conecta inicialmente a la LAN antes de obtener una dirección IP o alguna otra configuración; define un estándar para transmitir un protocolo de autenticación, como EAP (Extensible Authentication Protocol), sobre una red LAN 802 cableada o inalámbrica, empaquetando esos mensajes en tramas Ethernet. IEEE 802.1X también es conocido como EAPOL, refiriéndose a una encapsulación EAP sobre LANs (EAP encapsulation Over LANs).

En un esquema IEEE 802.1X, cuando un nodo hace una petición de conexión, se piden las credenciales del usuario a través del autenticador (ya sea un switch o un punto de acceso AP). Mientras tanto, el puerto del switch donde llega la petición está cerrado y solamente es permitido el tráfico EAP hasta que el usuario es autenticado. Después de que las credenciales del usuario han sido enviadas, el proceso de autenticación comienza a través del protocolo EAPOL, utilizado entre suplicante y autenticador. El autenticador re-encapsula los mensajes EAP en un formato RADIUS y los envía al servidor de autenticación. Cuando el proceso de autenticación termina, el servidor de autenticación envía un mensaje de éxito, es entonces cuando el autenticador abre el puerto al suplicante.

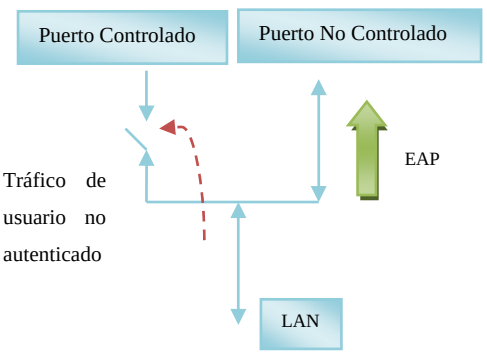


Fig. 2. Tráfico no autenticado

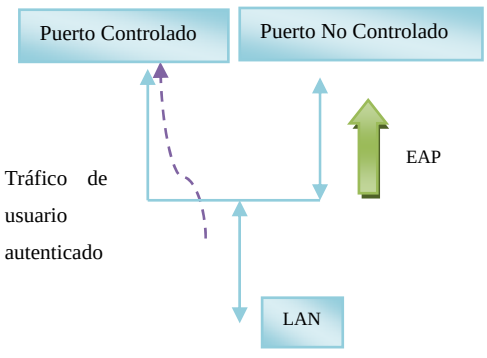


Fig. 3. Tráfico autenticado

En este proceso, el autenticador maneja sus puertos como controlados y no controlados, los cuales son entidades lógicas pero que utilizan la misma conexión física.

Antes de la autenticación, solamente el puerto No controlado está abierto y es permitido únicamente el tráfico EAPOL (ver Fig.2). Después de que se autentica al usuario, el puerto Controlado se abre y se permite el acceso a los recursos de la red (Ver Fig.3). Es decir, durante el proceso de autenticación se crean dos puntos distintos de acceso:

- Controlado: Permite el intercambio de marcos solamente si es un usuario autorizado.
- No controlado: Permite solamente intercambio de mensajes de autenticación EAP, en la capa física solamente se permite tráfico 802.1X

Si el autenticador es un punto de acceso inalámbrico, es importante proteger el envío de las credenciales del suplicante ya que viajan sobre una red inalámbrica, por lo que pueden ser interceptadas fácilmente. Ya que 802.1X está basado en EAP, se pueden utilizar varios métodos de protección de la autenticación. Los más comunes están basados en el estándar IETF TLS Transport Layer Security: Tunneled Transport Layer Security (TTLS) y Protected EAP (PEAP). Al utilizar 802.1X en redes inalámbricas se manda un tipo adicional de mensaje: EAPOL-Key, el cual transporta las claves desde el autenticador al suplicante.

2.1.2. EAP

El Protocolo Punto a Punto (Point to Point Protocol, PPP) fue diseñado para conectar computadoras con un enlace físico punto a punto, que permitiera autenticación del usuario. Principalmente se tenían dos protocolos de autenticación: el protocolo de autenticación de contraseña (PAP, Password Authentication Protocol) y el protocolo de autenticación por desafío (CHAP, Challenge Handshake Authentication Protocol) (2).

El primero, PAP, transmite el nombre de usuario y contraseña sin cifrar y el sistema comprueba la validez de la identificación. En el segundo, CHAP, el servidor de autenticación reta al cliente y el cliente prueba que tiene el secreto compartido respondiendo dicho reto; en este protocolo la contraseña nunca se envía por línea.

Posteriormente se hizo una extensión del protocolo PPP, conocido como Extensible Authentication Protocol o EAP, que proporciona un mecanismo estándar para aceptar métodos de autenticación adicionales junto con PPP.

EAP se diseñó originalmente para transportar información de autenticación en el contexto PPP/Dial-up, pero ahora se utiliza ampliamente en 802.1X. EAP provee de la metodología de autenticación, mientras que 802.1X provee el control de acceso basado en puertos.

EAP es un protocolo de autenticación flexible de la capa 2 del modelo OSI, el cual contribuye con un mecanismo genérico de transmisión de datos de autenticación que puede ser implementado en distintos subprotocolos. El cliente y el servidor pueden autenticarse mutuamente, y este proceso puede ser especializado para ajustarse a implementaciones particulares, lo cual es el motivo principal del porqué existen tantos tipos de EAP.

EAP en sí no es un mecanismo de autenticación. Tiene funciones comunes y negociaciones para los mecanismos elegidos, como podrían ser EAP-TLS, EAP-MD5 (no es recomendado para redes inalámbricas ya que solamente se usa una contraseña), LEAP (propietario de Cisco basado en nombre de usuario), PEAP y EAP-TTLS (combinan certificados del lado del cliente con alguna otra autenticación como contraseñas, ampliamente utilizados en redes inalámbricas).

EAP tiene ventajas con respecto a PAP o CHAP porque soporta más factores de autenticación, tales como contraseñas, certificados, biometría, etc. EAP no requiere de instalación de ningún código en el autenticador; sin embargo, hay que tener en mente que no todos los suplicantes, servidores RADIUS y autenticadores soportan todos los métodos de autenticación EAP.

A continuación se describen brevemente los mecanismos más conocidos de EAP:

EAP-TLS

Microsoft desarrolló EAP-TLS (EAP-Transport Level Security), basado en el protocolo Secure Socket Layer (SSL) que sirve para asegurar el tráfico Web. El uso de EAP-TLS es más apropiado cuando se cuenta con una arquitectura PKI (Public Key Infraestructure), ya que se requiere una Autoridad Certificadora, Autoridad de Registro, Sistema de manejo de certificados, etc. Si se tiene dicha infraestructura PKI, EAP-TLS permite proveer de autenticación mutua entre suplicante y servidor de autenticación, es decir, se usan certificados digitales para autenticar al servidor de autenticación y además es necesario autenticar al suplicante frente al servidor de autenticación igualmente.

Un segmento de la clave creada en el inicio de sesión TLS es enviado al autenticador y al suplicante, los cuales ya conocen la clave establecida, y el autenticador usa esa llave para el cifrado WEP.

EAP-TTLS

Otro mecanismo EAP es conocido como EAP-TTLS (EAP-Tunneled TLS), el cual es un estándar de IETF y es una extensión de la funcionalidad del protocolo de autenticación EAP-TLS; elimina el requisito de certificados digitales por lado del suplicante, permitiendo tener solamente un certificado del servidor de autenticación. Esto hace que este tipo de EAP sea seguro pero más fácil de administrar.

TTLS usa el canal TLS para intercambiar Attribute Value Pairs (AVPs), los cuales son validados en el servidor TTLS con cualquier tipo de mecanismo de autenticación. A su vez, el servidor de autenticación es autenticado con su certificado digital y el canal cifrado se establece entre el suplicante y el servidor de autenticación durante la fase inicial de autenticación de EAP-TTLS. Las credenciales de autenticación del suplicante, como contraseña o biometría, son enviados al servidor de autenticación a través del túnel y autenticado usando el algoritmo elegido, tal como MS-CHAPv2, MS-CHAP, CHAP, PAP, EAP-MD5, EAP-TNC, etc.

EAP-TTLS consta de dos etapas:

Etapa 1. Construcción del túnel TLS (túnel exterior)

1. El autenticador envía un paquete EAP-Request/ Identity al suplicante cuando éste se asocia al punto de acceso (ver Fig.4).

```
Wireless event: new AP: 00:40:00:00:f5:33
State: ASSOCIATING -> ASSOCIATED
[…]
EAP: EAP entering state IDENTITY
CTRL-EVENT-EAP-STARTED EAP authentication started
EAP: EAP-Request Identity data - hexdump_ascii(len=54):
     00 6e 65 74 77 6f 72 6b 69 64 3d 74 73 75 6e 61   _networkid=switc
[…]
EAP: using real identity - hexdump_ascii(len=3):
     65 6e 65                                          ene
```

Fig. 3. Ejemplo de EAP-Request Identity

2. El suplicante envía un paquete EAP-Response/Identity con su identidad. El autenticador lo re-empaqueta dentro del protocolo RADIUS y lo envía al servidor RADIUS (ver Fig.5).

```
EAP: EAP entering state SEND_RESPONSE
EAP: EAP entering state IDLE
EAPOL: SUPP_BE entering state RESPONSE
EAPOL: txSuppRsp
TX EAPOL: dst=00:40:96:38:f5:33
TX EAPOL - hexdump(len=12): 01 00 00 08 02 0c 00 08 01 65 6e 65
```

Fig. 4. Ejemplo de envío de respuesta mediante EAPOL

3. El autenticador envía un paquete EAP-Request/ EAP-Packet del tipo de autenticación deseado. Éste es típicamente escrito como un EAP-Request/TTLS. (Ver Fig. 6)

```
EAP: EAP entering state RECEIVED
EAP: Received EAP-Request id=13 method=21 vendor=0 vendorMethod=0
EAP: EAP entering state GET_METHOD
EAP: Initialize selected EAP method: vendor 0 method 21 (TTLS)
EAP-TTLS: Phase2 type: EAP
TLS: Phase2 EAP types - hexdump(len=48): 00 00 00 00 04 00 00 00 00
[…]
```

Fig. 5. Paquete EAP-TTLS

4. El servidor de autenticación y suplicante intercambian claves y construyen la capa TLS para cifrar la comunicación EAP subsecuente (Ver Fig.7).

```
EAP: EAP entering state METHOD
SSL: Received packet(len=6) - Flags 0x20
EAP-TTLS: Start (server ver=0, own ver=0)
TLS: using phase1 config options
TLS: Trusted root certificate(s) loaded
EAP-TTLS: Start
SSL: (where=0x10 ret=0x1)
SSL: (where=0x1001 ret=0x1)
SSL: SSL_connect:before/connect initialization
SSL: (where=0x1001 ret=0x1)
SSL: SSL_connect:SSLv3 write client hello A
SSL: (where=0x1002 ret=0xffffffff)
[…]
```

Fig. 6. Capa TLS- SSL

Etapa 2. Uso del túnel TLS para intercambiar información

1. El servidor de autenticación regresa un reto cifrado al autenticador. El autenticador lo desempaqueta de RADIUS y lo re-empaqueta en EAPOL y lo envía al suplicante (Ver Fig.8).

```
TX EAPOL - hexdump(len=103): 01 00 00 63 02 0d 00 63 15 00 16 03 […]
EAPOL: SUPP_BE entering state RECEIVE
RX EAPOL from 00:40:96:38:f5:33
```

Fig. 7. Paquetes de transmisión y recepción EAPOL

2. El suplicante responde al reto mediante el autenticador, el cual envía la respuesta al servidor de autenticación. Se realiza también el envío del certificado digital del servidor y se termina la etapa 2 (ver Fig.9).

```
EAP: EAP entering state METHOD
SSL: Received packet(len=715) - Flags 0x80
SSL: TLS Message Length: 2733
SSL: (where=0x1001 ret=0x1)
SSL: SSL_connect:SSLv3 read server hello A
TLS: tls_verify_cb - preverify_ok=1 err=0 (ok) depth=1
buf='/C=FR/ST=Radius/L=Somewhere/O=Example
Inc./emailAddress=admin@example.com/CN=Example Certificate Authority'
TLS: tls_verify_cb - preverify_ok=1 err=0 (ok) depth=0
buf='/C=FR/ST=Radius/O=Example Inc./CN=Example Server
Certificate/emailAddress=admin@example.com'
SSL: (where=0x1001 ret=0x1)
SSL: SSL_connect:SSLv3 read server certificate A
SSL: (where=0x1001 ret=0x1)
SSL: SSL_connect:SSLv3 read server key exchange A
SSL: (where=0x1001 ret=0x1)
SSL: SSL_connect:SSLv3 read server done A
SSL: (where=0x1001 ret=0x1)
SSL: SSL_connect:SSLv3 write client key exchange A
SSL: (where=0x1001 ret=0x1)
SSL: SSL_connect:SSLv3 write change cipher spec A
SSL: (where=0x1001 ret=0x1)
SSL: SSL_connect:SSLv3 write finished A
SSL: (where=0x1001 ret=0x1)
SSL: SSL_connect:SSLv3 flush data
[…]EAP: EAP entering state SEND_RESPONSE
[…]
EAP-TTLS: TLS done, proceed to Phase 2
EAP-TTLS: Derived key - hexdump(len=64): [REMOVED]
```

Fig. 8. Envío y verificación de certificado digital del servidor

3. Si el suplicante da las credenciales necesarias, el servidor de autenticación responde con un mensaje de éxito (ver Fig.10) y el autenticador permite el acceso a la red, con las restricciones enviadas por el servidor de autenticación. Se puede restringir con una cierta VLAN.

```
TNC: Recommendation = allow
EAP-TNC: TNCS done - reply with an empty ACK message
EAP-TTLS: AVP encapsulate EAP Response - hexdump(len=6): 02 03 00 06
26 01
EAP-TTLS: Encrypting Phase 2 data - hexdump(len=16): [REMOVED]
SSL: 90 bytes left to be sent out (of total 90 bytes)
EAP-TTLS: Authentication completed successfully (MAY_CONT)
```

Fig. 9. Mensaje de autenticación exitosa

Toda la información estará cifrada entre el suplicante y el servidor de autenticación. Si el cliente es configurado correctamente, la identidad del usuario no podrá ser vista, ya que al comienzo del protocolo se podrá usar una identidad anónima y hasta después de generar el túnel, enviar la identidad real. EAP-TTLS lleva a cabo esta característica al empaquetar otro protocolo de autenticación dentro del túnel TLS; a este protocolo se le conoce como protocolo interno. Con TTLS se puede usar un protocolo simple de autenticación, como una contraseña en claro, contraseñas en respuesta a un reto o técnicas más avanzadas, como autenticación basadas en tokens u otros métodos EAP.

Métodos de autenticación internos

- PAP Password Authentication Protocol. Transmite el nombre de usuario y contraseña sin cifrar. Soportado por TTLS.
- CHAP Challenge Handshake Authentication Protocol. El servidor de autenticación reta al cliente y el cliente prueba que tiene el secreto compartido respondiendo dicho reto. Soportado por TTLS.
- MS-CHAP Microsoft CHAP. No requiere que la clave esté en claro, utiliza una función hash para almacenar las contraseñas en el servidor. Soportado por TTLS.
- MS-CHAP-v2. Elimina el cifrado débil de las contraseñas. Es usado con PEAP y TTLS.
- EAP-MD5. Su estructura básica es similar a CHAP. Puede ser usado con TTLS o PEAP. También es utilizado en ambientes no inalámbricos fuera de túnel.

- EAP-TNC. Es un tipo de EAP que se basa en la especificación de Trusted Network Connect. Permite realizar verificaciones de integridad de los dispositivos en el proceso de autenticación. Este tipo de EAP se utilizará en este proyecto (Ver Fig. 11).

```
EAP-TTLS: received Phase 2: code=1 identifier=2 length=790
EAP-TTLS: Phase 2 EAP Request: type=38
EAP-TNC: Received packet: Flags 0x1 Message Length 0
TNC: Received IF-TNCCS BatchId=2
TNC: IMC-IMV-Message Type 0x80ab30
TNC: Message to IMC(s) - hexdump_ascii(len=137):
[…]
```

Fig. 10. Mensajes EAP-TNC

La estructura del paquete EAP-TNC se encuentra dentro de la carga real de datos (payload) de un paquete EAPOL en IEEE802.1X como se muestra en las siguientes figuras:

Ethernet

Dirección de Destino	
Dirección de Destino	Dirección de Origen
Dirección de Origen	

EAPOL

Tipo Ethernet	Versión de Protocolo	Tipo de Paquete
Longitud de Paquete		

EAP-TNC

Código	Identificador							Longitud
Tipo	Banderas/ Versión							Longitud de datos
	L	M	S	R	R	V	V	V	
Longitud de datos								Datos

Fig. 11. Paquete EAP-TNC

La comunicación EAP-TNC comienza con un mensaje EAP-TNC-Start, vacío, solamente con la bandera de inicio encendida (S). La carga real (payload) está en el campo de Datos. Si es un mensaje muy largo, es posible fragmentar el paquete, encendiendo la bandera de Más-fragmentos (M) y la bandera de Longitud (L).

2.1.3. RADIUS

Cuando se realiza una implementación de seguridad, la autenticación de usuario es un componente crucial, por lo que se tienen varias soluciones tales como Kerberos, RADIUS (Remote Authentication Dial-In User Service) y LDAP.

Para este proyecto se controlará el acceso de los usuarios con un servidor RADIUS, ya que los servidores RADIUS son robustos y escalables, y permiten autenticación, autorización y contabilidad (Authentication – Authorization – Accounting). Este tipo de servidores que cuentan con dichas características se conocen como servidores AAA.

En el esquema de un servidor AAA, existen tres tipos de secuencias de autorización (3):

1. AGENT. El servidor AAA actúa como intermediario entre el equipo del servicio y el usuario final.

2. PULL. El usuario final se conecta directamente con el equipo del servicio, el cual verifica en el servidor AAA si se da acceso o no. Este es el esquema a utilizar dentro del proyecto.

3. PUSH. El usuario actúa como agente entre el servidor AAA y el equipo de servicio. El servidor AAA distribuye algún recibo de autenticación al usuario, lo cual es utilizado dentro de la petición de acceso.

El servidor AAA que se utilizará en este proyecto implementa RADIUS, el cual es un protocolo sencillo, eficiente y fácil de implementar. Se envían el nombre de usuario y sus credenciales, por medio del Network Access Server (NAS) al servidor RADIUS. Este último verifica con su base de datos, la cual puede ser interna o externa, que la información sea correcta. El acceso al sistema es autorizado de acuerdo con las reglas configuradas por el administrador de red, por medio de políticas dentro del servidor RADIUS.

Las características principales de RADIUS (Remote Authentication Dial in User Service) son:

- Modelo Cliente-Servidor
- Las transacciones entre el cliente y el servidor RADIUS son autenticadas a través del uso de un secreto compartido, el cual nunca es enviado sobre la red
- Permite utilizar varios mecanismos de autenticación flexibles
- Es un protocolo extensible, ya que se pueden añadir nuevos valores de atributos

- Utiliza los puertos 1812 y 1813 (iniciales, depende de las peticiones y pueden ser cambiados)
- Capacidad de manejo de sesiones
- Utiliza paquetes UDP
- Base de datos de usuarios comúnmente es SQL, Kerberos, LDAP o Active Directory

El protocolo RADIUS está definido en el RFC 2865 (authentication and authorization) y el RFC 2866 (accounting).

Cuando un servidor RADIUS decide que se tuvo una autenticación exitosa envía ciertos atributos hacia el suplicante dentro de la respuesta de aceptación; dichos atributos podrán ser la dirección IP asignada o el conjunto de direcciones IP donde se podría elegir, el tiempo máximo de conexión, una lista de acceso u otras restricciones, parámetros de VLANs, parámetros de calidad de servicio (QoS).

La contabilidad (accounting) dentro de un servidor RADIUS permite reconocer cuando un usuario comienza o termina una conexión y adquirir datos de toda su sesión. Además se puede tener un historial de actividad de cada usuario. Cuando se concede el acceso a la red por el punto de acceso, se manda una petición Accounting Start por el punto de acceso hacia el servidor RADIUS. Normalmente se guarda la identificación del usuario, dirección de red, punto de acceso y un identificador único de sesión. Periódicamente, los registros de Accounting son enviados por el autenticador al servidor RADIUS, para actualizar el estado de una sesión activa, registrando la duración de la sesión y otros datos. Cuando se termina la sesión, el autenticador deberá emitir un registro Accounting Stop al servidor RADIUS, incluyendo la información de tiempo de uso, paquetes transmitidos, datos transmitidos, razón de desconexión, etc.

Con respecto a la seguridad del protocolo RADIUS, se puede señalar que no transmite contraseñas en claro entre el autenticador y el servidor RADIUS. Utiliza un secreto compartido junto con MD5 para proteger las contraseñas. Sin embargo, se recomienda utilizar protección adicional para cifrar el tráfico, ya que algunos atributos pueden ser considerados como información privada.

Paquete RADIUS

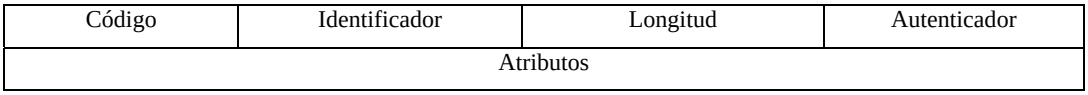


Fig. 12. Paquete RADIUS

1. Código (1 byte). Identifica el tipo de paquete
2. Identificador (1 byte). Diferencia peticiones duplicadas cuando son dos paquetes conteniendo la misma dirección IP y puerto en el lado del cliente. Necesario para tener correspondencia entre petición y respuesta.
3. Longitud (2 bytes). Longitud completa del paquete, que será de 20 a 4096 bytes.
4. Autenticador (16 bytes). Utilizado para autenticar la respuesta del servidor RADIUS y ocultar la contraseña. Permite verificar integridad del mensaje. Puede ser autenticador de petición o de respuesta.
5. Atributos

2.1.4. Biometría

La autenticación de un usuario dentro el control de acceso es fundamental. A lo largo de la historia se han utilizado varios mecanismos que ayuden a decidir si un usuario es quien dice ser. Principalmente existen cuatro grupos de mecanismos de autenticación, que pueden ser combinados para crear una solución más robusta:

1. Algo que se conoce (Ej. Usuario/contraseña)
2. Algo que se tiene (Ej. Token)
3. Algo que caracteriza (Ej. Huella dactilar)
4. Algo que determina su posición (Ej. Ubicación satelital)

La biometría forma parte del tercer grupo, ya que se utiliza alguna característica humana para identificar: se basa en la medición de la fisonomía del humano, para después utilizar descriptores matemáticos de los rasgos medidos. Cada tipo de biometría tiene sus fortalezas y debilidades, y hay que reconocer que ninguno es cien por ciento seguro. Para poder hacer uso de la biometría es necesario contar con sistemas biométricos, los cuales consisten de hardware y software; el hardware captura la característica del ser humano, y el software interpreta los datos.

La biometría nunca es exacta, por lo que no es posible utilizar funciones hash (por ejemplo, MD5) para comparar datos biométricos, ya que nunca serán iguales aunque sea el mismo dedo mostrado en el sistema. Las razones de variaciones en las mediciones de los rasgos pueden ser:

- presentación inconsistente
- presentación irreproducible
- adquisición imperfecta de la representación

La operación de un sistema biométrico es la siguiente:

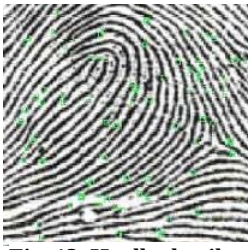


Fig. 13. Huella dactilar

1. Registro

- Uso de sistema de detección de señales con arquitectura de reconocimiento de patrones (lector de huella dactilar)
- Detección de una señal biométrica cruda
- Procesamiento de dicha señal para extraer ciertas características (minucias)
- Almacenamiento de los descriptores matemáticos de dichas características en una base de datos

2. Identificación o Autenticación

- Uso del mismo sistema de detección de señales (o uno compatible)
- Detección de una señal biométrica cruda
- Procesamiento de dicha señal para extraer ciertas características (minucias)
- Comparación de esas características con un conjunto de características que se encuentran en la base de datos
- Validación de la identidad

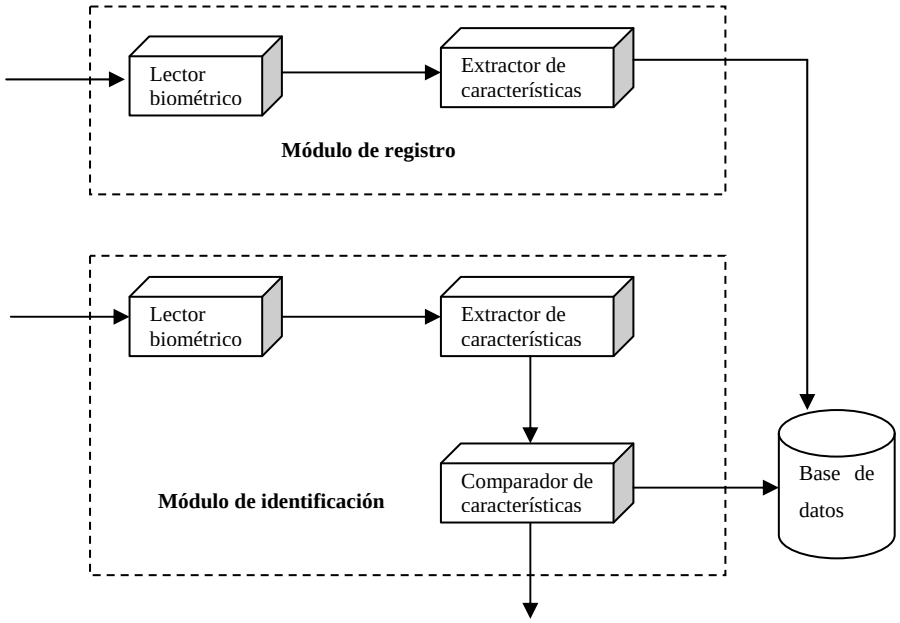


Fig. 14. Sistema biométrico

En el proceso de identificación es necesario comparar la característica medida entre las características de todos los sujetos registrados, mientras que en el proceso de autenticación se conoce la identidad del sujeto y solamente es necesario hacer una comparación.

La exactitud de un sistema biométrico se refiere a la tasa de falsos positivos y falsos negativos. La tasa de falsos positivos es el porcentaje de usuarios que no están autorizados pero que sí se les permite el acceso, y la tasa de falsos negativos es el porcentaje de usuarios autorizados a los que no se les permite el acceso. Los falsos positivos se consideran más graves en sistemas biométricos, pues permiten el acceso de intrusos. Cuando ambos son iguales se tiene la tasa de error "equiprobable", la cual es la principal medida de precisión de un sistema biométrico.

La rapidez de un sistema biométrico depende el sistema de cómputo y del sensor, e indica el tiempo necesario para anunciar una decisión (aproximadamente 5 segundos). La tasa de operación se mide desde que el usuario se acerca al sensor hasta que el usuario logra acceder al sistema (aproximadamente 6 segundos por usuario).

Para el desarrollo de este proyecto se utilizará un sistema biométrico de huella dactilar, ya que tiene un bajo costo y el sistema no manejará más de cien usuarios. Los sistemas de huella digital son exactos, pero pueden ser afectados por los cambios en la huella digital (quemaduras, las cicatrices, etcétera) y por la suciedad y otros factores que modifiquen la imagen. El esquema de representación que utiliza el sistema biométrico de huella dactilar es la Distribución de Minucias,

y el algoritmo de concordancia es la igualación de cadenas; necesita tener un módulo de procesamiento de imágenes y un lector específico. Los factores de este tipo de biometría son los siguientes:

Tabla 1. Factores de biometría de huella dactilar

Factor	**Descripción**	**Grado (Bajo, Medio, Alto)**
Universalidad	Qué tan usado es	M
Distintivo	Facilidad de distinguir característica	A
Permanencia	Característica utilizada que no cambia con el tiempo drásticamente	A
Desempeño	Facilidad de empleo y procesamiento	A
Aceptabilidad	Percibido como no intrusivo	A
Vulnerabilidad al fraude	Cómo es probable que se viole su seguridad	M

Las tecnologías de hardware principales que se utilizan para obtener las imágenes son la óptica y la capacitiva. En un sistema óptico, el dedo se coloca en una superficie de cristal, y una fuente de luz interna destaca los pliegues. El dispositivo de captura utiliza típicamente un sensor basado en un CCD (dispositivo acoplado de carga eléctrica). El problema es que los dedos pueden ensuciar el área de detección, dejando una impresión fantasma llamada imagen latente. En un cierto plazo, las imágenes latentes pueden degradar la capacidad del dispositivo de capturar una impresión exacta. El lector biométrico utilizado para este proyecto es de tipo óptico.

Los sistemas capacitivos utilizan un sensor, un arreglo de circuitos que crean una imagen de la huella midiendo el campo eléctrico que la rodea. Este tipo de sistema es muy exacto, pero el contacto directo de los dedos puede dañar el dispositivo a largo plazo.

Ataques a un sistema biométrico de huella dactilar

En un estudio hecho por la universidad del Estado de Michigan **(2)**, en Estados Unidos, se detectaron varios tipos de ataques a los cuales son vulnerables los sistemas biométricos basados en huella dactilar. Estos ataques dependen del lugar donde se realizan dentro del sistema biométrico:

1. Ataques en los que se presenta una falsa biometría ante el sensor (Ej. Dedo falso, huella de grasa, etc.).
2. Ataques en los que presenta una biometría interceptada anteriormente.

3. Ataques en los que el módulo extractor es comprometido, produciendo valores seleccionados por el atacante.
4. Ataques en los que los valores genuinos son reemplazados por los seleccionados por el atacante.
5. Ataques donde el evaluador es modificado para aceptar el ingreso al sistema.
6. Ataques en donde se compromete la plantilla de la base de datos.
7. Ataques producidos entre la base de datos y el comparador.
8. Ataques en donde el resultado del comparador es sobrescrito por el atacante.

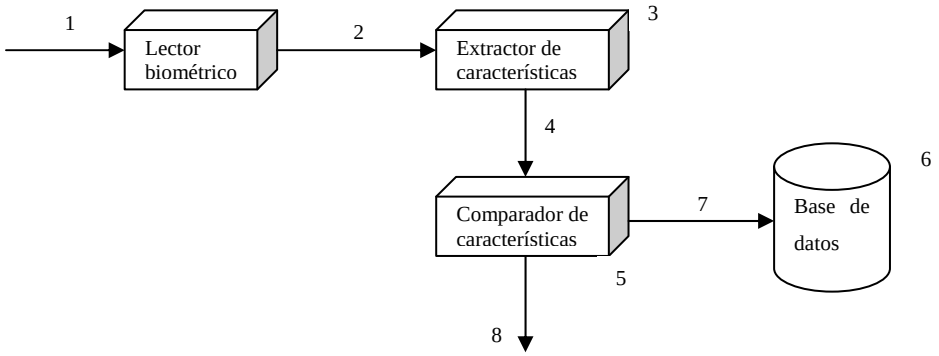


Fig. 15. Ataques en sistema biométrico

También existen otros problemas en lo que respecta a un sistema biométrico:

- Negación de Servicio (DoS). Cuando se logra que el sistema no esté disponible.
- Elusión. Cuando se logra acceder al sistema por otro acceso sin pasar por la autenticación.
- Repudio. Cuando el atacante niega el acceso a los usuarios.
- Contaminación. Cuando se roban datos biométricos de usuarios legítimos.
- Colusión. Cuando el atacante es un usuario legítimo con privilegios de acceso.
- Coerción. Cuando se obliga a un usuario legítimo acceder al sistema.

3 Control de acceso

3.1. Definición de control de acceso

El control de acceso es una actividad principal de la seguridad informática. Es la combinación del establecimiento de perímetros y de mecanismos de autenticación, que evita que usuarios sin permisos puedan acceder a una red o sistema, basándose en la definición de alguna política de seguridad. Las principales etapas que debe tener el control de acceso son: registro, identificación, autenticación y autorización.

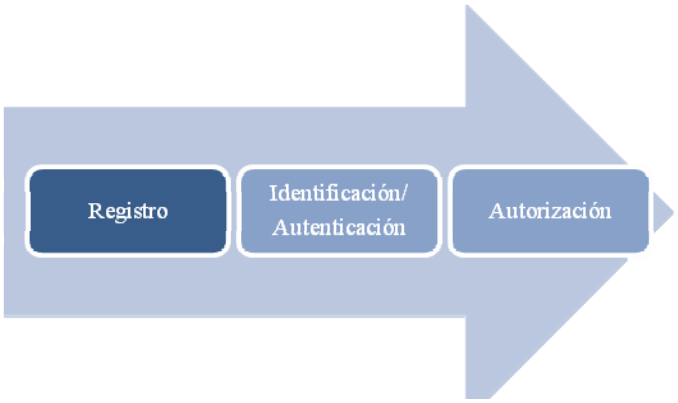


Fig. 1. Etapas del control de acceso

Es importante tener un sistema acotado para poder tener un buen control de acceso, por lo que es necesario saber en todo momento cuáles son todos los componentes y todos los usuarios (tener un inventario), y saber cómo están interactuando en un momento dado.

Una vez que un usuario muestra un autenticador ante un sistema de control de acceso, si satisface las políticas, logra convertirse en un usuario activo del sistema de información a través de una sesión; a partir de ese momento todas sus acciones están ligadas a su identidad. La fase que sigue es evitar que el usuario ejecute alguna acción de la cual no tiene permiso. El control de acceso también debe cuidar que un usuario no pueda robar la identidad a otro para adquirir sus derechos.

El control de acceso se basa en las políticas para tomar la decisión de aceptación o rechazo, por lo que las políticas, que especifican los accesos autorizados, deben estar diseñadas para proteger la confidencialidad, privacidad, autenticidad y disponibilidad. Además es necesario proteger la información de las políticas contra modificaciones no autorizadas.

3.2. Network Access Control NAC

El control de acceso que es objeto de estudio en este proyecto es NAC, el cual se puede resumir en tres aspectos: autenticación, evaluación de seguridad e información ambiental de la red.

3.2.1. Introducción a NAC

Las organizaciones actuales tienen varios mecanismos para asegurar el perímetro de la red, tales como firewalls, dispositivos VPNs, IDS, antivirus, antispam, monitoreo y filtrado Web, etc. Dichos dispositivos requieren diferentes políticas, diferente administración y recursos, haciendo esto un esquema ineficiente, por lo que se hace necesario tener un esquema más general e integral, como lo es Network Access Control –NAC–. NAC incluye políticas de seguridad de pre admisión y control después de la admisión, verificando en cada momento los usuarios e invitados que ingresan y trabajan sobre la red, permitiendo centralizar las políticas de los mecanismos para asegurar el perímetro. Es decir, NAC se enfoca en las políticas de seguridad (decisión y aplicación) y restricción de tráficos prohibidos, identificando y deteniendo a usuarios que no cumplan con las políticas, teniendo una zona segura de cuarentena y remedio.

Network Access Control es una forma diferente de asegurar el perímetro de la red. Se deben autenticar tanto el usuario como la máquina con la que accede a la red (endpoint) (3)

El control de acceso a la red NAC debe cumplir:

- Definición de políticas. Posibilidad de tener y mantener una variedad de políticas de seguridad para todo tipo de usuarios, con facilidad de modificarlas desde una consola de administración central.

- Detección. Posibilidad de detectar cualquier conexión.

- Verificación de seguridad. Posibilidad de escanear el endpoint y determinar si cumple las políticas. Este escaneo debe darse antes de dar acceso a la red, pero también se deben permitir otras revisiones después del acceso.

- Mejoras. Las políticas determinan qué recursos de red deben ser protegidos. Si no se cumplen las políticas, deberá ser posible mitigar faltas de software, o actualizaciones. Si no es posible mejorar el estado de seguridad será necesario crear cuarentena de recursos o rehusar el acceso completamente.

- Soluciones. Si el usuario que se quiere conectar no cumple con lo establecido, deberá ser posible realizar actualizaciones, aplicación de parches u otras medidas de mitigación para que pueda conectarse.

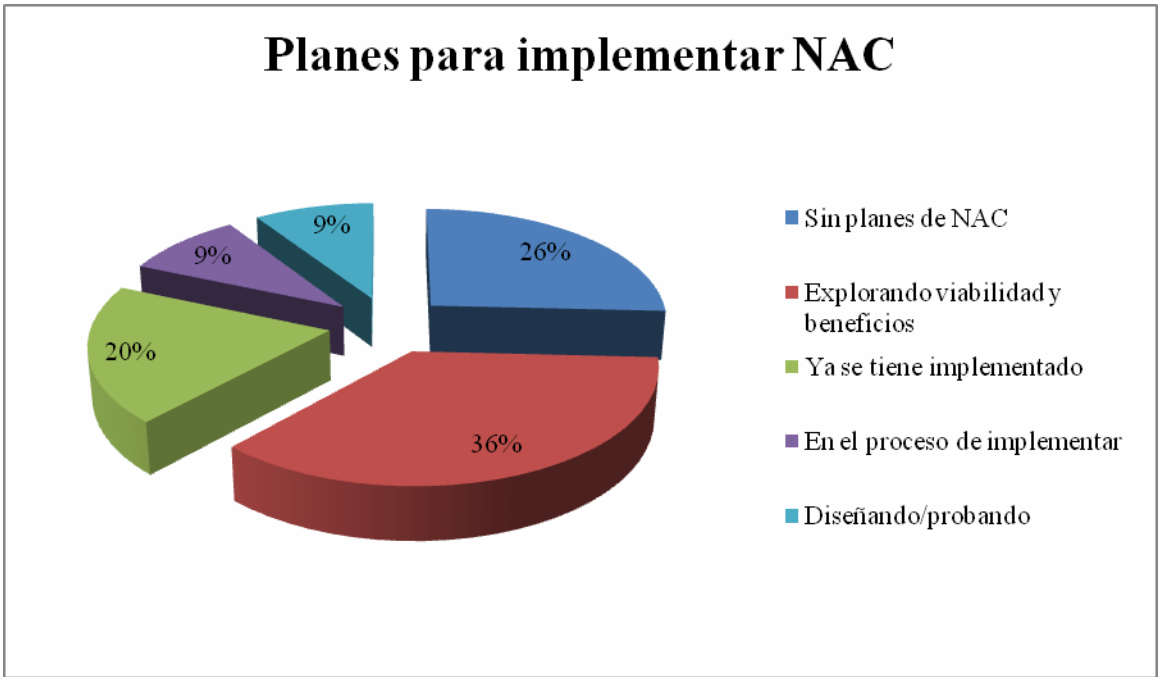


Fig. 2. Planes de implementación de NAC

(Basado en *http://www.ins.com/knowledge/surveys/industrySurvey.asp*)

Es importante recalcar que NAC también permite el control de la seguridad interna. Es decir, no solamente protegerá de accesos externos no permitidos, sino también de ataques internos, con computadoras que forman parte de la red, con ayuda del monitoreo interno de cada uno de los usuarios.

También hay que considerar que algunos usuarios o invitados serán mentirosos, es decir, que premeditadamente intentarán burlar los mecanismos de seguridad y podrán arrojar datos falsos para acceder a la red, lo cual NAC no es capaz de detectar. Para esto es necesario utilizar hardware confiable, por lo que el grupo TCG trabaja en un módulo externo (TPM) pero compatible con NAC que verifica que los dispositivos no den datos falsos.

Con esto se observa que no es posible eliminar todas las amenazas con cualquiera que sea la solución, pero lo que puede hacer una organización de manera realista es evaluar y disminuir vulnerabilidades que comprometen a sus sistemas.

El objetivo principal de NAC es tener un mejor control de los dispositivos que quieren acceder a una red y los recursos que tienen derecho a utilizar. Actualmente se ha comenzado a considerar el control de acceso a invitados o dispositivos no administrados como parte de una estrategia integral de administración de la red y no solamente como una preocupación de seguridad. Esto hace que la solución NAC esté siendo aceptada en el mercado, tanto que la empresa Gartner hizo varias encuestas, en 2008, y reportó un gran crecimiento en los desarrollos NAC en las grandes compañías en comparación del 2006 (4). Se observó que algunas compañías han comenzado a implementar NAC, tanto en infraestructura como en operación, con algunos departamentos pilotos, debido principalmente al elevado costo de desarrollar la solución completa. (3)

NAC comenzó a tomar auge en 2003, cuando apareció el gusano Blaster. Aún no se tiene un estándar robusto y se ha demostrado que es más difícil de implementar de lo que parecía; sin embargo, se ha visto que es un componente crítico para tener seguridad eficiente y no solamente para compañías grandes.

Para comenzar con NAC se requiere definir los escenarios que requieren control de acceso, analizando qué tipo de identificación se usará, las posturas de evaluación y aplicación; además, de las prioridades y limitaciones (4). El desarrollo de NAC es complejo, por lo que se recomienda primero monitorear la red, analizar su tráfico y entonces establecer las políticas necesarias.

3.2.2. Categorías principales de NAC

Existen tres categorías principales de NAC:

1. Productos basados en infraestructura
2. Dispositivos que filtran
3. Software en endpoints

Estas diferentes categorías se pueden utilizar en conjunto para lograr una solución más completa. Normalmente los productos que se basan en infraestructura y software son más robustos pero se requiere más inversión y son más complicados de configurar.

3.2.2.1. *Infraestructura NAC*

Algunas compañías necesitan instalar NAC a través de su infraestructura para tener políticas consistentes en su LAN, VPN y WLAN. Este enfoque es el más exhaustivo y caro, ya que se requiere tener una integración compleja y se tendrán que reemplazar los equipos obsoletos por equipos compatibles con NAC.

Existen varios estándares para este tipo de infraestructura, pero CISCO domina el mercado. Microsoft también intenta entrar al mercado con Network Access Protection (5).

3.2.2.2. *Dispositivos NAC*

Hay dos esquemas dentro de los dispositivos NAC:

1. En línea (in-line)

   Tendrán acceso a la red sólo los usuarios autorizados, parecido al funcionamiento de una VPN en acceso remoto.

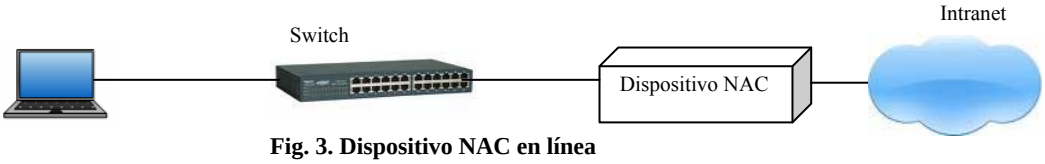


Fig. 3. Dispositivo NAC en línea

2. Fuera de banda (out of band)

   Un nodo de control recibe las peticiones de los endpoints, los escanea y monitorea. Los endpoints que no cumplan son aislados usando componentes in-band (routers, switches) y se les soluciona.

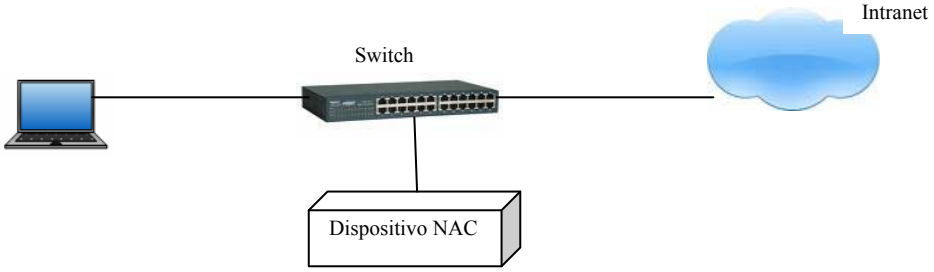


Fig. 4. Dispositivo NAC fuera de banda

3.2.2.3. Software en endpoints

Los productos basados en software normalmente son instalados en las computadoras de la misma empresa para asegurarse que estén actualizadas, que tengan los últimos parches, etc. Sin embargo, para los visitantes no será tan fácil instalar un software por lo que normalmente se utiliza un software de escaneo que se ejecuta a través del navegador Web, aunque no es tan exhaustivo.

Existen algunas restricciones en el uso de esta categoría, como el problema de que los usuarios no puedan cumplir con los requisitos de software para instalar o ejecutar la aplicación, o que se tenga algún sistema operativo no soportado. Por lo que normalmente esta solución se utiliza en conjunto con otros productos NAC, basándose en la información recabada por el software.

Agentes

Se utilizan agentes de software para escanear y analizar un dispositivo, para decidir qué acciones tomar y qué soluciones darle.

- Agente permanente (Thick agent). Este tipo de agentes son un archivo ejecutable instalado permanentemente en cada endpoint, teniendo como consecuencia que sea difícil de desarrollar dicho agente para todo tipo de hardware. También existe la opción de que ejecute tareas que solucionen los problemas que se encuentren en el dispositivo o que genere un reporte. El problema serio que se enfrenta con este agente es que los invitados no querrán fácilmente instalar un agente de esta índole.

- Agente dinámico (On-demand agent). No persiste más allá del período que el dispositivo está conectado a la red. Normalmente son controles Java o ActiveX, cargados vía una sesión Web, por lo que pueden desinstalarse al terminar la sesión (disolubles). Para implementarlos será necesario considerar su tiempo de descarga y cuánto tiempo estará conectado el usuario para determinar si es la opción correcta.

- Sin agente (Agentless solution). No se instala ningún software en el endpoint, pero se opera con alguno que exista en el dispositivo o se escanea remotamente. Tiene capacidades limitadas, por lo que no es muy recomendable. Se puede utilizar escáneres de vulnerabilidades, como Nessus.

De acuerdo a las encuestas realizadas por BT INS se tiene la siguiente gráfica de los usos de las diferentes categorías de NAC.

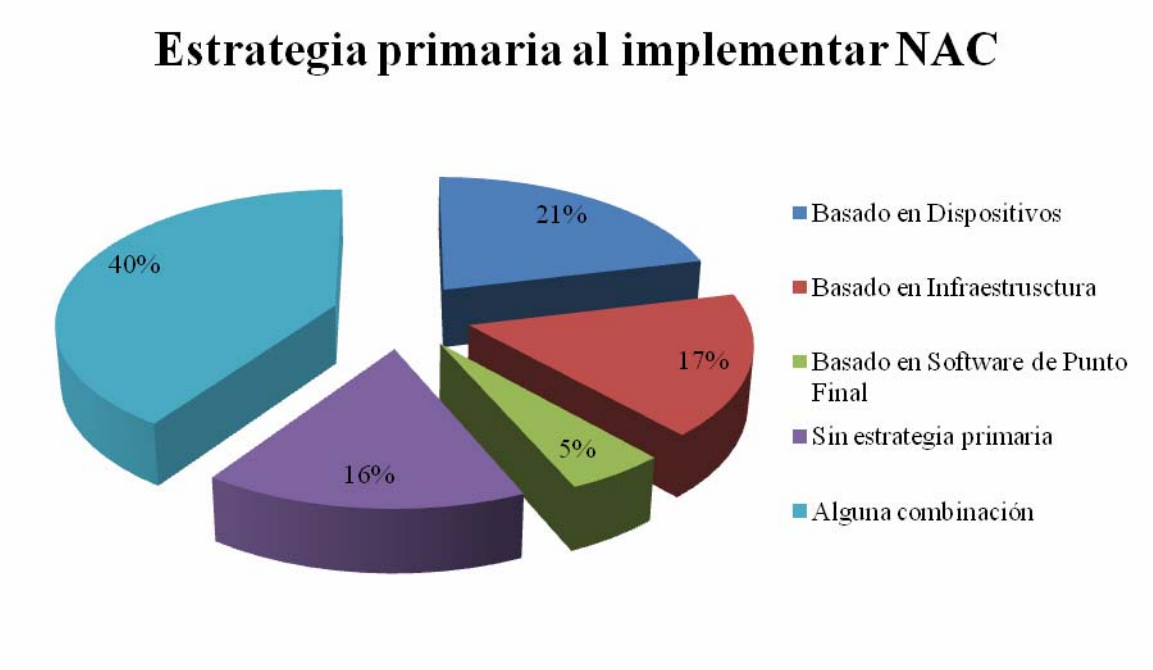


Fig. 5. Estrategia primaria al implementar NAC

(Basado en *http://www.ins.com/knowledge/surveys/industrySurvey.asp*)

3.3. Trusted Computing Group: Trusted Network Connect TNC

La seguridad de la información no solamente abarca el control de acceso a una red sino que existen otros intereses y preocupaciones, por lo que se han formado varios grupos que generan soluciones, estándares o líneas de investigación. Uno de dichos grupos que está interesado en definir estándares, para llegar a tener un cómputo confiable, se conoce como Trusted Computing Group (TCG). Este grupo trabaja en varias líneas de investigación, una de las cuales es el subgrupo Trusted Network Connect (TNC) que se enfoca en NAC; su objetivo principal es crear una arquitectura abierta alternativa a iniciativas privadas (por ejemplo, Microsoft y Cisco). Dicha arquitectura está pensada para asegurar interoperabilidad entre múltiples vendedores, teniendo como meta crear un estándar (o varios) para NAC, aunque todavía es una tecnología emergente.

Las diferentes líneas de investigación de Trusted Computing Group son las siguientes:

1. Hard copy.
   Este grupo trabaja en especificaciones para máquinas copiadoras, que usarán componentes TCG.
2. Infraestructure.
   Este grupo define el marco de la arquitectura, interfaces y metadatos necesarios para poder unir diferentes arquitecturas.
3. Mobile.
   Este grupo se enfoca en el trabajo para dispositivos móviles (laptops, teléfonos celulares y PDAs).
4. PC Client
   Este grupo provee de funcionalidad común, interfaces y un conjunto de requerimientos de seguridad y privacidad a PCs que usan componentes TCG.
5. Server
   Este grupo trabaja sobre definiciones, especificaciones, pautas y requerimientos técnicos para implementar tecnología TCG en servidores.
6. Software Stack
   Este grupo trabaja sobre un conjunto de estándares de APIs para quienes hacen aplicaciones y quieran utilizar TPM.
7. Storage
   Este grupo contribuye a las tecnologías TCG existentes y se enfocan en los estándares para la seguridad de sistemas de almacenamiento.
8. Trusted Network Connect
   Este grupo se enfoca en asegurar endpoints conforme a políticas de integridad durante y después de la conexión a red.
9. Trusted Platform Module (TPM)
   Este grupo trabajó en la especificación de TPM. Este es la base para los otros grupos de TCG.

Dado que este proyecto se enfoca en NAC, se explicará con más detalle el trabajo del subgrupo Trusted Network Connect (TNC). Éste ha definido y liberado una arquitectura libre y un conjunto de estándares para asegurar la integridad de los dispositivos que los sigan. Los estándares que

proponen permiten la interoperabilidad entre dispositivos de varios fabricantes, aunque utilicen diferentes tecnologías y políticas.

La arquitectura TNC tiene soporte para los protocolos de autenticación IEEE 802.1X, para control de acceso a redes. Los access points y switches que soportan 802.1X no necesitan ser modificados para soportar las verificaciones necesarias por TNC; simplemente se pasa la información al servidor de autenticación (RADIUS). En cambio, los servidores de autenticación necesitan estar actualizados para manejar información adicional de TNC.

3.3.1. Arquitectura TNC

Uno de los objetivos de la arquitectura TNC es lograr la interoperabilidad de soluciones NAC y el uso del cómputo confiable. Para lograr este objetivo hace uso de medidas de integridad para catalogar el estado o postura de seguridad del endpoint.

La arquitectura TNC tiene los siguientes actores (8):

- Suplicante de Acceso (Access Requestor AR)
    1. Suplicante de Acceso a la Red (Network Access Requestor NAR)
    2. Cliente TNC (TNC Client TNCC)
    3. Colectores de Medidas de Integridad (Integrity Measurement Collectors IMCs)
- Punto de Cumplimiento de Políticas (Policy Enforcement Point PEP)
- Punto de Decisión de Políticas (Policy Decision Point PDP)
    1. Autoridad de Acceso a la Red (Network Access AuthorityNAA)
    2. Servidor TNC (TNC Server TNCS)
    3. Verificadores de Medidas de Integridad (Integrity Measurement Verifiers IMVs)
- Punto de Acceso a Metadatos (Metadata Access Point MAP)
- Controladores de flujo y Sensores (Flow Controllers and Sensors)
- Aplicación de Remedio y Proveedores (Provisioning & Remediation Application PRA) [1]
- Recursos de Remedio y Proveedores (Provisioning & Remediation Resources PRR)

[1] PRA y PRR no forman parte de la arquitectura TNC, sin embargo, son considerados elementos importantes de NAC en su fase de Remedio.

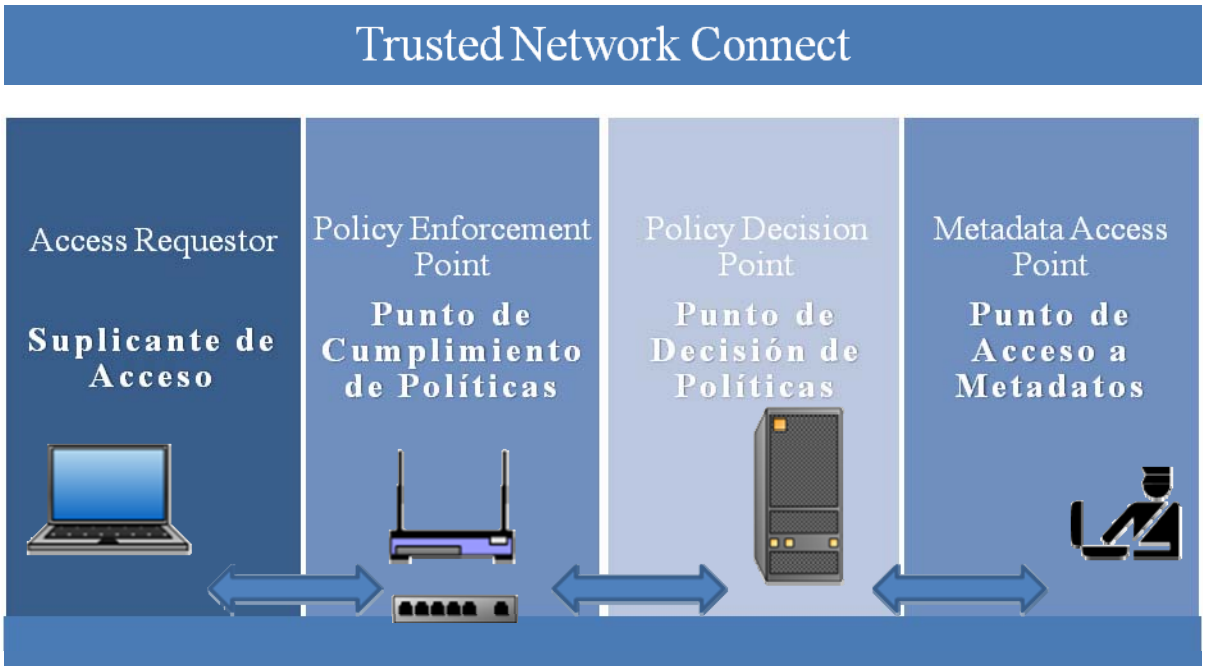


Fig. 6. Arquitectura TNC

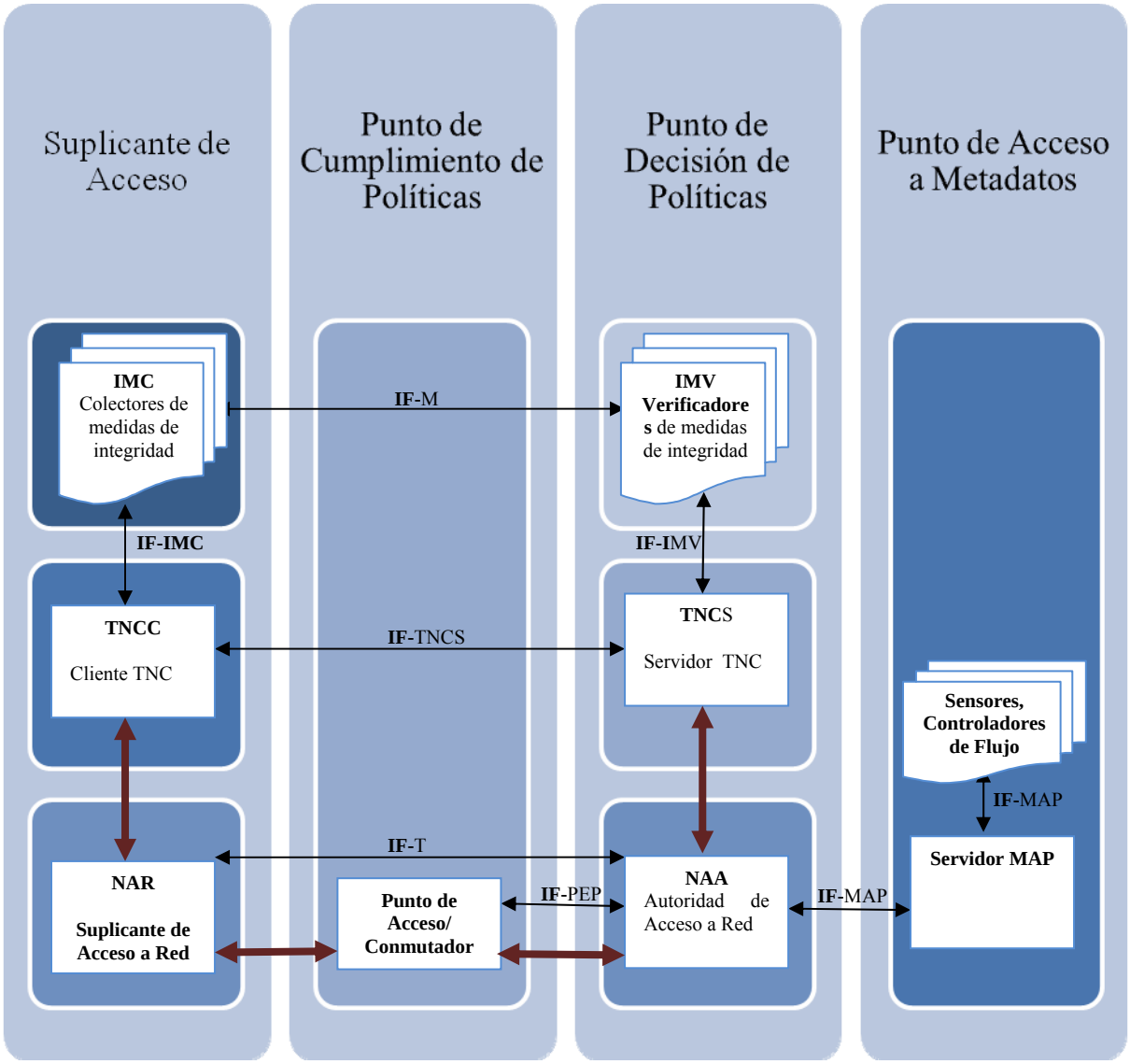


Fig. 7. Arquitectura TNC del grupo TCG (Fuente:[8] TNC_Architecture _v1_3_r6.pdf)[2]

[2] [8] TNC Architecture for Interoperability. Specification Version 1.3, Revision 6. 28 April 2008. Pág. 12.

El comportamiento común dentro de esta arquitectura es el siguiente: el PDP compara las credenciales del AR y la información de su estado de seguridad, con ciertas políticas de acceso a la red y toma la decisión de rechazar o aceptar. Si existe el PEP, el PDP le comunica su decisión, para que la ejecute. Los controladores de flujo y sensores son opcionales; pueden coordinar el monitoreo y cumplimiento de las políticas establecidas, recibiendo y compartiendo información a través de MAP.

Componentes del Access Requestor

1. Network Access Requestor NAR. *Suplicante 802.1X*
2. TNC Client *TNCC. Componente de software que se ejecuta en AR*
3. Integrity Measurement Collector IMCs. *Mide parámetros de antivirus, estado del firewall personal, versiones de software, etc.*

Policy Enforcement Point

1. PEP. *Autenticador en el esquema 802.1X, el cual es frecuentemente implementado dentro del switch.*

Controladores de Flujo

1. Elementos que toman acciones sobre flujos de red. *Firewalls internos, limitadores de tasas de transferencia, etc.*

Sensores

1. Elementos que dan información de las actividades en la red. *IDS, monitores de tráfico capa 3, escáneres de tráfico, etc.*

Metadata Access Point

1. Permite a los componentes contar con las relaciones entre endpoints, usuarios, capacidades, roles, actividades, etc.

Componentes de Policy Decision Point

1. Network Access Authority NAA. *Componente dentro del servidor AAA que decide si un AR tiene acceso, consultando al TNC Server.*

2. TNC Server TNCS. *Administra el flujo de mensajes entre IMV e IMC, juntando las Acciones Recomendadas de los IMVs para enviarlas como TNCS Action-Recommendation hacia el NAA.*

3. Integrity Measurement Verifier IMV. *Verifica los datos enviados por los IMCs contra las políticas establecidas.*

3.3.2. Interfaces TNC

Las interfaces que se establecen en la arquitectura TNC definen relaciones entre los componentes, además de protocolos y mensajes intercambiados.

Trusted Network Connect (TNC) IF-MAP

Metadata Access Protocol. Elementos que pueden compartir y correlacionar datos de la ejecución como relación entre endpoints, usuarios, capacidades, roles y atributos. Su objetivo es crear una visión general de todo el sistema de control de acceso.

Trusted Network Connect (TNC) IF-IMC

Integrity Measurement Collector Interface (API). Reúne las medidas de integridad de los IMCs hacia el TNC Client.

Trusted Network Connect (TNC) IF-IMV

Integrity Measurement Verifier Interface (API). Interfaz entre IMVs y el TNC Server, que permite dar las recomendaciones al TNCS.

Trusted Network Connect (TNC) IF-PEP

Policy Enforcement Point Interface. Permite comunicar el PDP y el PEP, llevando instrucciones para aislar al AR durante el remedio y otorgar acceso al terminar. Normalmente se implementa con el protocolo RADIUS.

Trusted Network Connect (TNC) IF-T

Network Authorization Transport Protocol. Transporta mensajes entre AR como entidad y el PDP como entidad. Se pueden enviar mensajes entre NAR y NAA. Normalmente se implementa con EAP sobre 802.1X.

IF-T permite transportar los mensajes IF-TNCCS entre NAR y NAA, que permiten la comunicación entre IMCs e IMVs.

IF-T puede mapearse a métodos EAP con túnel. Un método EAP con túnel es el que provee una capa protegida criptográficamente dentro de la cual otros protocolos pueden ser enviados.

Trusted Network Connect (TNC) IF-TNCCS

TNC Client-Server Interface. Interacción entre TNC Client y TNC Server cuando intercambian datos de medidas de integridad. Manejo de mensajes de IMCs a IMVs, como paquetes de medidas de integridad; mensajes de IMVs a IMCs, como peticiones de medidas adicionales o instrucciones de remedio); mensajes de manejo de sesión y sincronización entre Cliente y Servidor.

Trusted Network Connect (TNC) IF-M

Vendor-Specific IMC-IMV Messages. Protocolo donde se especifican mensajes específicos, transportados sobre IF-TNCCS.

Trusted Network Connect (TNC) IF-PTS

Platform Trust Services Interface. Esta interfaz adicional todavía está en desarrollo, donde se define cómo utilizar TPM con la arquitectura TNC.

3.3.3. Evaluación, aislamiento y remedio

La arquitectura TNC actual no abarca el manejo del remedio, aunque sí se contempla en una capa superior: Provisioning & Remediation Layer.

Evaluación

Los IMVs ejecutan la verificación del AR siguiendo las políticas establecidas por el administrador de red y si es necesario envía instrucciones de remedio a los IMCs.

Aislamiento

Si el AR ha sido autenticado y tiene algunos privilegios, pero no pasó la verificación de integridad por el IMV, el PDP debería regresar instrucciones al PEP para redirigir al AR a un ambiente aislado, donde el AR puede obtener actualizaciones relacionadas a la integridad. Se

pueden utilizar VLANs, teniendo acceso limitado para acceder a dichos recursos. RADIUS permite esa contención con VLAN usando Tunnel Private Group ID (RFC 3580).

Remedio

Es el proceso del AR para obtener correcciones a su configuración y otros parámetros de políticas específicas para que cumpla con los requisitos del PDP. Cuando se completa, los IMCs pueden iniciar otra fase de evaluación, teniendo en consideración la posibilidad de hacerlo más corta si se mandan solamente los datos que cambiaron.

Se tienen dos entidades relevantes para el proceso de Remedio.

1. Provisioning & Remediation Application (PRA)

Puede ser implementado como parte del AR, tal vez en un IMC. Se comunica con IMC y le da tipos específicos de información de integridad. Puede ser un antivirus.

2. Provisioning & Remediation Resources (PRR)

Representa las fuentes de información de integridad necesarias para actualizar al AR. Pueden ser servidores FTP, discos con parámetros de actualización o services packs (SP).

3.3.4. Desventajas de TNC

Una de las desventajas de la arquitectura TNC es que no se pueden detectar los endpoints mentirosos, es decir, los dispositivos que están configurados para no contestar honestamente acerca de sus capacidades de antivirus y políticas. Este problema se puede mitigar al monitorear todas las actividades de los endpoints durante y después de su conexión. Sin embargo, los sistemas infectados se detectan ya que comenzaron a esparcir la infección o a realizar actividad maliciosa; además, no todo el malware se puede detectar simplemente, tales como rootkits o keyloggers.

Un módulo adicional y opcional a la arquitectura TNC es el TPM Trusted Platform Module. Es una solución basada en hardware para evitar dicho problema de dispositivos mentirosos, la cual calcula funciones hash durante el arranque (booting), de todo el software crítico y los componentes de firmware antes de cargarlos: BIOS, kernel SO, boot loader, etc. Dichas medidas pueden ser enviadas al TNCS durante el handshake de la arquitectura TNC.

Las características principales de una plataforma con TPM son las siguientes:

a) Capacidades Protegidas. Conjunto de comandos con permisos exclusivos para acceder a posiciones blindadas (Shielded Locations: memoria, registros, etc.), donde es seguro operar con datos sensibles. TPM almacena llaves criptográficas usadas para autenticar las medidas reportadas.

b) Medidas de integridad y almacenamiento. Es el proceso de obtener métricas de las características de la plataforma que afectan la integridad de una plataforma; almacenar dichas métricas y obtener su hash.

c) Reporte de integridad. Es el proceso de avalar los contenidos de almacenamiento de integridad. El reporte es firmado usando la llave privada que se encuentra dentro de una posición blindada. El resultado es puesto en un log y su hash se añade a un registro para descubrir alguna alteración en el log.

d) Evaluaciones. Proceso de garantizar la precisión de la información.

3.3.5. Tecnologías utilizadas en TNC

Acceso a red

- **802.1X**

  Estándar para el Control de Acceso basado en Puertos (PBAC). Provee de un mecanismo de autenticación a los dispositivos de red. Es usado por la mayoría de switches y access points 802.11 y se basa en el protocolo EAP (Extensible Authentication Protocol).
- **VPN Virtual Private Network**

  Acceso remoto basado en VPNs que utiliza el protocolo IKE (Internet Key Exchange). Fuera del objetivo del trabajo.
- **PPP Point-to-Point Protocol**

  Método estándar para transportar datagramas multiprotocolo. Es la base para el acceso dial-up a Internet sobre PSTN (Public Switched Telephone Network). Normalmente utiliza EAP. Fuera del objetivo del trabajo.

Transporte de mensajes

- **EAP**
  Extensible Authentication Protocol

- **TLS y HTTPS**
  HTTP es usado para el transporte de mensajes relacionados a la aplicación, como en servicios Web. El protocolo TLS puede ser extendido para transportar reportes de integridad. Es posible tener una página web de login sobre HTTPS para autenticar a los usuarios.

Policy Decision Point

- **RADIUS**
  EAP permite varios métodos de autenticación para ser usados entre el AR (cliente o suplicante) y PDP (Servidor de autenticación). Las extensiones están definidas en RFC 3579; el objetivo de las extensiones es usar a RADIUS para transportar paquetes EAP encapsulados entre el AR (o PEP) y el PDP. Es necesario tener EAP-Message y Message –Authenticator.
- **DIAMETER**
  Mejora algunas deficiencias del protocolo RADIUS, como la fiabilidad de transporte, capacidades de negociación y soporte de roaming. RFC 3588.

3.3.6. Consideraciones de seguridad y privacidad

Es necesario tener un canal seguro entre AR y PDP para que el PEP no pueda tener acceso al contenido de este canal. Su implementación es dependiente del área de aplicación y configuración de la red, por ejemplo, un canal establecido a través de EAP con sus variantes PEAP o TTLS, en el contexto de configuración 802.1X.

Un TNCC deberá comunicarse solamente con IMCs autorizados. Un TNCS deberá comunicarse solamente con IMVs autorizados. La habilidad de un TNCC de descubrir los IMCs tiene beneficios pero también riesgos de seguridad, ya que TNCC debe tener suficientes privilegios para acceder a la información en IMCs. Por tal motivo el diseño e implementación de las interfaces deben prever spoofing, DoS y tampering ilegal.

Además es recomendable cuidar la integridad de AR y PDP, ya que deben estar protegidos en contra de ataques que modifiquen ilegalmente sus configuraciones.

Con respecto a la seguridad de soluciones de remedio, si se requiere que el AR se comunique con un servidor de remedio (RS) para obtener las actualizaciones, es necesario utilizar firmas u otra protección que asegure su integridad.

La autenticación del usuario no es requerida para ejecutar un handshake bajo la arquitectura TNC. En escenarios donde se requiere la protección de la identidad del usuario, el acceso anónimo a la red es soportado donde se requiere la protección de la identidad del usuario.

La arquitectura (8) permite la negociación del tipo de medidas necesarias para tomar decisiones de acceso, es decir, la plataforma tiene el control de las políticas, pero el cliente puede decidir cuándo abortar la petición de conexión para preservar su privacidad. El usuario puede determinar cuáles IMCs pueden ser instalados y/o cargados por el TNCC basándose en la evaluación de la habilidad del IMC para proteger la privacidad. Los que implementan IMCs pueden emplear filtros en los flujos de salida para bloquear, reemplazar o modificar los reportes de integridad, por lo tanto, el TNCC no es un buen lugar para aplicar controles de privacidad.

Es indispensable utilizar un cifrado de la información sensible, si el cliente decide proporcionar cierta información para obtener acceso a la red, se debe proteger para que no pueda ser accedida por otros.

3.4. Metodología Gartner

Gartner (4) presenta un modelo de Network Access Control, el cual señala que dicho control deberá ser cíclico, es decir, debe presentarse antes de la conexión y después de la conexión, basándose en un Baseline y en las Políticas de cada empresa.

Cuando se presenta la conexión, deben tenerse acciones de mitigación o solución de reglas no cumplidas. Además es indispensable continuar monitoreando al elemento suplicante para que no tenga comportamientos fuera de las políticas.

El modelo de Gartner señala que se deben seguir los siguientes pasos, divididos en dos categorías:

Categoría Pre-connect

- Políticas (Politics). Tienen períodos establecidos, se tienen políticas básicas y dependiendo de cada usuario se incrementarán las peticiones. Aquí se toman decisiones

como por ejemplo si deben tener firewalls personales, qué parches son necesarios, si deben usar firmas, etc.

- Implementación de políticas (Baseline). Tiene como fin determinar el cómo implementar las políticas. Se pueden utilizar agentes en los endpoints o agentes externos para verificar que se siga el baseline.
- Control de Acceso (Access Control). El control de acceso en sí. Determina a qué usuario poner en cuarentena o no dejarlo pasar definitivamente.
- Mitigación (Mitigation). En este punto se tiene la cuarentena para endpoints, parcheo y remedios necesarios para que un usuario cumpla con las políticas.

Categoría Post-connect

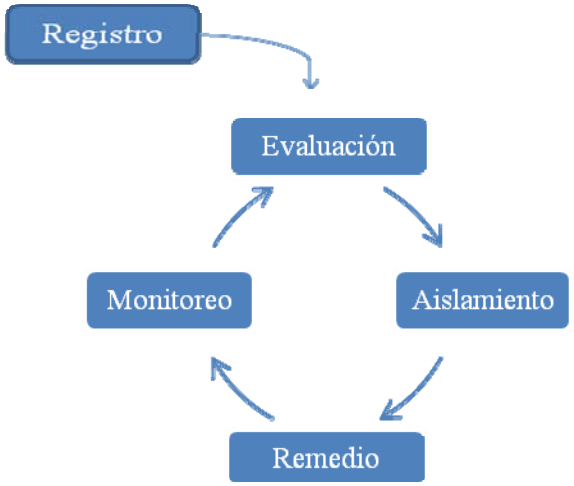


Fig. 8. Etapas de NAC

- Monitoreo (Monitoring & containing). Se deben considerar dos aspectos:
    1. Monitoreo de los endpoints, verificar su comportamiento.
    2. Monitoreo de lo que está pasando en la red, análisis de tráfico, detección de anomalías.
- Mantenimiento (Maintain). Es importante verificar que se mantengan los requerimientos operacionales de la red, es decir, que no se afecten los procesos de día a día.

Gartner ha trabajado en el tema de NAC, dando su propia definición y aumentando las capacidades necesarias para un sistema de control de acceso. Es importante que se tengan contemplados varios métodos de acceso, por ejemplo. VPNs, dial-up, SSL, inalámbrico, LAN, DHCP, 802.1X, acceso Web, etc. También es importante tener un DRP (Disaster Recovery Plan) e incluir redundancia en el sistema para asegurar que el sistema esté seguro y sea escalable.

Es indispensable el manejo eficiente de las políticas de seguridad, desde un punto de vista central, ya que en algunas organizaciones se tienen diferentes roles con diferentes configuraciones de seguridad; además debe ser fácil configurar nuevas políticas o editar las que ya se tienen.

Uno de los retos más grandes de NAC es el manejo de los invitados a la red. Normalmente los invitados no querrán instalar un agente permanente en sus equipos, por lo que normalmente se usa un agente on-demand (también conocido como "disoluble" o dinámico).

3.5. NEA-IETF

El grupo Internet Engineering Task Force (IETF) se enfoca en los protocolos de Internet, y ahora se interesó en el tema de Network Access Control, iniciando un grupo de trabajo: Network Endpoint Assessment (NEA) Working Group (9).

La meta de IETF NEA es publicar estándares para NAC. Ya existen varios protocolos y propuestas en esta área, por lo que ha comenzado con un documento de requerimientos. Se basa en protocolos cliente-servidor, evitando definir APIs. Para tener un alcance razonable han excluido la detección de dispositivos "mentirosos", remediación, aplicación y desarrollos no empresariales, por lo que NEA solamente tratará con la evaluación del endpoint y no se dedicará a la propagación de los resultados de dichas evaluaciones hacia los dispositivos de aplicación de políticas (Policy Enforcement). IETF NEA no reemplazará el trabajo de Trusted Computing Group TNC.

Su modelo es similar a las arquitecturas NAC existentes, pero no tiene la intención de basarse en alguna en específico. Incluye dos entidades:

- NEA Client
    - Posture Collectors. Recaba información acerca de la seguridad del cliente.
    - Posture Broker Client. Recaba la información de los colectores y la envía al NEA Server.
    - Posture Transport Client. Transportan dicha información.

- NEA Server
    - Posture Broker Server. Recibe la información acerca de los puntos finales y la envía hacia los validadores
    - Posture Transport Servers. Transportan dicha información.
    - Posture Validators. Determinan el cumplimiento de las políticas.

Protocolos

IETF NEA ha identificado tres protocolos como candidatos de estandarización:

1. PA. Protocolo para mensajes enviados entre Posture Collectors y Posture Validators.
2. PB. Protocolo para mensajes enviados entre Posture Broker Client y Posture Broker Server.
3. PT. Protocolo de transporte

Dichos protocolos pueden estar encapsulados, es decir, PA esta dentro de PB, el cual es transportado por PT.

3.6. Productos NAC

CISCO: CNAC Framework (Network Admission Control)

Normalmente se espera que el switch pueda manejar el concepto de NAC, por lo que CISCO se ha enfocado a dicho dispositivo. Sin embargo, no sigue los estándares principales de NAC. Utiliza un agente persistente, pero no tiene la habilidad de implementar mitigación (10).

No es una solución para todas las configuraciones, se basa en 4 puntos:

- NAC L3 IP routers
- NAC L2 802.1X (también soportado en switch)
- NAC L2 IP (requiere switch capa 3)
- NAC Dispositivos sin agentes

Ofrece los dispositivos NAC Guest Server y NAC Profiler. El primero se comunica con varios dispositivos NAC y controladores inalámbricos también Cisco y da las capacidades para la administración y reporte de la actividad de invitados. El NAC Profiler permite descubrir y monitorear dispositivos no autenticados. También introdujeron un módulo para routers, para facilitar la implementación de NAC en oficinas remotas. Puede desarrollarse en in-band u out-of-band, con lo que se puede mejorar la escalabilidad de los dispositivos NAC, pero está limitado a

usar switch Catalyst. La parte de los puntos finales se maneja con un agente (permanente o disoluble) o utilizando un escáner utilizando firmas Nessus. Los dispositivos NAC de Cisco son caros, por lo que si no se tiene una infraestructura Cisco, no será ideal utilizar esta solución. Una ventaja es que la solución Cisco tiene la posibilidad de trabajar en conjunto con Microsoft, ya que tienen interoperabilidad CNAC/MNAP.

Microsoft Network Access Protection. MNAP

El esfuerzo de Microsoft para el control de acceso a redes se conoce como MNAP. Tiene la limitante que será necesario contar con Vista Desktop y Longhorn Server, además de tener el Quarantine Agent de Microsoft para seguir baselines. Se enfoca en la identidad, es decir, permite la integración con Active Directory, basándose en Group Policy. Usa IPSec para autenticar todas las comunicaciones (11). Algunas de las desventajas que tiene MNAP es que no cuenta con un dispositivo específico para NAC y que se debe tener toda la infraestructura Windows para tener un control de acceso confiable.

Symantec NAC

Symantec Corporation tiene una solución NAC: Symantec Network Access Control. Permite tener control de acceso y seguimiento de políticas para puntos finales administrados, usuarios invitados y dispositivos no administrados.

Los puntos finales no administrados que intentan conectarse a la red pueden tener el software de protección y seguridad necesario gracias a que se tiene un cliente bajo demanda, que realiza una revisión predefinida para garantizar que el software antivirus, antispyware, firewall, así como los paquetes de servicio estén instalados y actualizados (12).

Lo manejan como una expansión a sus herramientas de control de acceso, a través de la integración entre Symantec Endpoint Protection.

Mediante un nuevo acceso a la Web que se puede habilitar como parte del proceso de descarga del cliente bajo demanda, Symantec Network Access Control también soporta la autenticación y control de acceso de identidad para usuarios invitados, quienes pueden autenticarse o firmar con claves de acceso almacenadas en ActiveDirectory, LDAP, RADIUS o con contraseñas guardadas localmente (13).

Tiene la desventaja que no maneja la detección de malware después de la conexión y funciones de contención.

FreeNAC

FreeNAC provee una solución transparente para la administración dinámica de redes virtuales, a la vez que restringe la conectividad a la red. Detecta dispositivos desconocidos que están tratando de obtener acceso a través de un conector de red Ethernet abierto, negando el acceso (y registrando el evento). Dispositivos conocidos y registrados son colocados a la red virtual que les corresponde.

Los invitados pueden opcionalmente tener acceso a una zona de redes virtuales por defecto o para invitados. Esto puede ser útil, por ejemplo, para organizaciones que desean permitir a sus visitantes acceso Web/VPN a Internet, pero restringir el acceso a las redes internas.

FreeNAC tiene dos modos de operación:

- **VMPS**
- **802.1X**

VMPS (VLAN Management Policy Server) es un método para asignar puertos de un switch a redes virtuales específicas de acuerdo a la dirección MAC del dispositivo que busca acceso a la red. En modo VMPS, un switch compatible con VMPS detecta un suplicante y crea una petición VMPS pidiendo autorización de FreeNAC, el cual revisa en su base de datos y permite o niega el acceso a la red basándose en la dirección MAC. El switch se encarga de respaldar la decisión tomada por FreeNAC y niega acceso o en caso contrario, coloca el dispositivo de manera dinámica en su red virtual por defecto.

En modo 802.1X, FreeNAC verifica las credenciales de los usuarios (a través del uso de un servidor de autentificación externo) y usa la dirección MAC del dispositivo que se conecta para asignarlo a una red virtual. Esto crea un par nombre de usuario/dispositivo que es único para cada cliente que se conecta. Para un usuario malicioso no basta con saber únicamente la dirección MAC, sino que también debe de obtener credenciales válidas, lo cual hace más difícil el obtener acceso a la red (14).

4. Propuesta

De acuerdo al contexto de las soluciones NAC y la arquitectura TNC que se va seguir se han hecho algunas consideraciones y acotaciones a este proyecto. Con respecto a la identidad del usuario se relacionará con dos tipos de autenticadores, algo que conoce (contraseña) y algo que lo caracteriza (huella dactilar). Se enviarán ambas credenciales mediante mensajes EAP.

Como ya se mencionó se seguirá la arquitectura Trusted Network Connect para la propuesta de implementación de un sistema NAC, con herramientas de software libre, sobre la plataforma Linux. La clasificación de NAC que se aborda es Basado en software, por lo que se utilizará el esquema de agentes permanentes o ”thick agents” por ser el más confiable.

Todos los usuarios, incluyendo a los invitados, deberán realizar un registro (en el lugar donde se encuentra la infraestructura), donde deberán proporcionar su nombre de usuario, contraseña y descriptor de huella dactilar, siguiendo las políticas establecidas. Es decir, para este proyecto se propone que todos los usuarios deberán ejecutar el agente para poder ingresar a la red. Dicho agente será entregado en el momento del registro.

El agente será el encargado de pedir los datos del usuario, incluyendo los datos biométricos. Dichos datos serán enviados dentro del protocolo de autenticación EAP-TTLS/EAP-TNC, permitiendo la autenticación del usuario utilizando biometría antes de permitir flujos normales de información.

Se tendrán sensores para el monitoreo de las conexiones e identificación de anomalías, con lo que se minimizará el daño causado por ataques internos y problemas de día cero. Esto es posible al tener considerado poner en cuarentena a cualquier usuario ya autenticado, si no cumple con las políticas establecidas.

4.1. Políticas

Una política de seguridad informática es una forma de comunicarse con los usuarios, ya que las mismas establecen un canal formal de actuación del personal, en relación con los recursos y servicios informáticos de la organización. Si no se tienen políticas consistentes y bien definidas no se puede tener un buen control de acceso, por lo que es recomendable especificar quién y cómo puede acceder a la red. Una de las formas más sencillas es tener lo que se denomina matriz de conectividad, donde se exponen los tipos de usuarios junto con las características necesarias para ser aceptados y en acceso que obtendrán.

4.1.1. Matriz de conectividad

Tabla 1. Matriz de conectividad

Id usuario	Estado de Seguridad	Control de Acceso
Usuario	Biometría y contraseña aceptadas Puertos autorizados SO parchado	Acceso total
Invitado	Biometría y contraseña aceptadas Puertos autorizados SO parchado	Acceso total
Invitado	Biometría y contraseña aceptadas Puertos no autorizados SO parchado	Acceso a la red de remedio
Invitado	Biometría y contraseña aceptadas Puertos autorizados SO no parchado	Acceso a la red de remedio
No autenticado	Desconocido	Sin acceso

Los usuarios son los empleados o personas que tienen un acceso regular dentro de la red. Deben estar registrados dentro del servidor de autenticación, dando de alta su nombre de usuario, contraseña y biometría de huella dactilar. Deben tener instalado un agente en el equipo con el que se conectan a la red. Se da por hecho que los usuarios no pueden conectar los dispositivos a otra red y que no pueden salir de las instalaciones, donde están bajo supervisión de un administrador que los mantiene actualizados y correctamente configurados.

Para este trabajo no se consideran los usuarios con acceso remoto por conexiones utilizando VPNs.

Los invitados deberán registrarse antes de ingresar a la red. Deberán de dar de alta los mismos datos que un usuario regular y deberán instalar el agente que se les proporcione en el momento del registro. No se deberán cambiar los archivos de configuración del servidor de autenticación

4.1.2. Sistema operativo

El sistema operativo que se verificará en este trabajo es el siguiente:

Tabla 2. Sistema operativo

Sistema Operativo	Actualizaciones	Parches
Linux kernel 2.4.X o mayor	Apt-get update	Todos los disponibles

Se tendrán las actualizaciones listas en el servidor, para que no sea necesaria la conexión a Internet.

4.1.3. Puertos autorizados

En este proyecto se verificará el estado de solamente algunos puertos del endpoint, ya que solamente se quiere demostrar la funcionalidad. Con esta tabla se demuestra que se puede verificar cualquier puerto. Se eligieron los siguientes puertos, permitiendo que solamente esté abierto el servicio de HTTP: [1]

Tabla 3. Puertos autorizados

Puerto	Estado	Puerto	Estado
TCP 80	Open	TCP 443	Close
TCP 20	Close	TCP 8080	Close
TCP 21	Close	TCP 5223	Close
TCP 22	Close	UDP 53	Close
TCP 23	Close	UDP 67	Close
TCP 25	Close	UDP 68	Close
TCP 587	Close	UDP 4444	Close
TCP 110	Close		
TCP 995	Close		

[1] Ver anexo con número de puertos y sus servicios

4.2. Arquitectura

4.2.1. Elementos del entorno de seguridad

Tabla 4. Mapeo de la arquitectura TNC con elementos utilizados

Elemento	Descripción	Acciones
AR	Usuario que requiere conexión (laptop con tarjeta de red Ethernet)	Registrarse ante el servidor de autenticación Instalar agente permanente Pedir acceso a la red
1. NAR	Suplicante 802.1X	Iniciar la petición de conexión
2. TNCC	Programa C++	Descubrir y cargar IMCs Manejar mensajes entre IMCs e IMVs Enviar recomendación de acción
3. IMC	Programa C++	Envío de datos biométricos Verificar puertos autorizados
PEP	Autenticador: Switch con soporte 802.1X (CISCO Catalyst 2950)	Asignación dinámica de VLAN, mediante atributos RADIUS
Sensor	IDS Snort	Reportar anomalías Monitorear tráfico en red
PDP	Servidor de Autenticación	Registrar usuarios Decidir acceso Mandar instrucciones a PEP
1. NAA	Programa C++	Decidir si se permite el acceso Llevar registro de sesiones (accounting) Asociar identidad de usuario con sesión
2. TNCS	Programa C++	Manejar flujo de mensajes ente IMC e IMV
3. IMV	Programa C++ con conexión a base de datos de autenticación	Verificar identidad biométrica y credenciales Comparar medidas de integridad de usuario con políticas
MAP	Servidor con base de datos	Correlacionar información de nessus y snort
(PRA)	Antivirus	Comunicar con IMCs para informar acerca de remedio de integridad
(PRR)	Paquetes para seguridad de Ubuntu	Fuentes de información de integridad

4.2.2. Interfaces y protocolos

La arquitectura TNC cuenta con siete interfaces, a través de las cuales los elementos se comunican entre sí. Algunas de las interfaces son protocolos ya conocidos como RADIUS y EAP, y otros protocolos son interfaces definidas internamente dentro de la arquitectura, como se muestra en la siguiente tabla.

Tabla 5. Interfaces entre los elementos definidas por la arquitectura TNC

Interfaz	Protocolo
IF-IMC	Inferfaz C++ (API)
IF-IMV	Interfaz C++ (API)
IF-M	Protocolo transportado sobre IF-TNCCS
IF-MAP	Interfaz (Peticiones a base de metadatos)
IF-PEP	Protocolo RADIUS
IF-T	EAP sobre 802.1X (EAP-TTLS/EAP-TNC)
IF-TNCCS	Protocolo y formato para transportar mensajes

Las interfaces IF-IMC e IF-IMV definen el formato de los mensajes enviados entre los IMCs e IMVs y las funciones necesarias para comunicarse.

La interfaz IF-T está definida mediante el protocolo de autenticación IEEE 802.1X usando cualquier tipo de EAP. Para este proyecto se utilizó EAP-TTLS/EAP-TNC, pero se puede elegir cualquier otro.

4.2.3. Flujos

4.2.3.1. Autenticación exitosa sin remedio

Para el flujo que se presenta en una autenticación exitosa sin remedio se siguen los siguientes pasos; se considera que se tiene una petición de conexión, sin necesidad de remedio:

0. Antes de comenzar una conexión, el TNCC debe detectar y cargar cada IMC, verificando su integridad y una conexión segura. Similarmente, el TNCS debe reconocer y cargar cada IMV.

1. Cuando se inicia un intento de conexión, el NAR inicia una petición de conexión en las capas de enlace y red.

2. Cuando el PEP recibe la petición, manda una petición de decisión de acceso a la red hacia el NAA. Se configura el NAA para ejecutar autenticación de usuario (User Authentication), autenticación de credenciales de plataforma (Platform Credential Authentication) y verificación de integridad (Integrity Check Handshake). La autenticación de usuario se da entre NAA y AR; las otras dos entre AR y TNCS.

3. Si tiene éxito la autenticación del usuario, el NAA informa al TNCS la petición de conexión.

4. Entonces TNCS ejecuta la autenticación (mutua) de las credenciales de la plataforma con el TNCC.

5. Si tiene éxito esa autenticación entre TNCS y TNCC, el TNCS indica a los IMVs que hay una nueva petición de conexión y que se necesita llevar a cabo la verificación de integridad. Similarmente el TNCC se los indica a los IMCs.

6A. Para que ocurra la verificación de integridad (Integrity Check Handshake), el TNCS y el TNCC comienzan el intercambio de mensajes de verificación de integridad. Esos mensajes se transmiten a través del NAR, PEP y NAN, y continúan hasta que el TNCS decide que es aceptable el estado de integridad del AR.

6B. El TNCS envía cada mensaje de los IMCs a los correspondientes IMVs. Cada IMV analiza dichos mensajes. El IMV necesita avisar al TNCS si necesita comunicarse aún más con un IMC o ya está listo para decidir su recomendación (IMV Action-Recommendation) y su resultado de la evaluación (IMV Evaluation-Result).

6C. Similarmente, el TNCC reenviará mensajes del TNCS a los IMCs correspondientes.

7. Cuando el TNCS completa el ICH con el TNCC, manda su recomendación (TNCS Action-Recommendation) al NAA, aunque el NAA todavía tiene la opción de no permitir el acceso.

8. El NAA manda su decisión al PEP para su ejecución. También debe indicar su decisión al TNCS, la cual será enviada al TNCC. Normalmente el PEP indica su ejecución de la decisión al NAR. Envía identificador de VLAN a donde conectarse y su dirección IP.

4.2.3.2. Autenticación exitosa con remedio

Se considera que se tiene una petición de conexión, con necesidad de remedio. Se siguen los pasos:

0. Antes de comenzar una conexión, el TNCC debe detectar y cargar cada IMC, verificando su integridad y una conexión segura. Similarmente, el TNCS debe reconocer y cargar cada IMV.
1. Cuando se inicia un intento de conexión, el NAR inicia una petición de conexión en las capas de enlace y red.

2. Cuando recibe la petición el PEP, manda una petición de decisión de acceso a la red hacia el NAA. Se configura el NAA para ejecutar autenticación de usuario (User Authentication), autenticación de credenciales de plataforma (Platform Credential Authentication) y verificación de integridad (Integrity Check Handshake). La autenticación de usuario se da entre NAA y AR; las otras dos entre AR y TNCS.

3. Si tiene éxito la autenticación del usuario, el NAA informa al TNCS la petición de conexión.

4. Entonces TNCS ejecuta la autenticación (mutua) de las credenciales de la plataforma con el TNCC.

5. Si tiene éxito esa autenticación entre TNCS y TNCC, el TNCS indica a los IMVs que hay una nueva petición de conexión y que se necesita llevar a cabo la verificación de integridad. Similarmente el TNCC se los indica a los IMCs.

6A. Para que ocurra la verificación de integridad (Integrity Check Handshake), el TNCS y el TNCC comienzan el intercambio de mensajes de verificación de integridad. Esos mensajes se transmiten a través del NAR, PEP y NAN, y continúan hasta que el TNCS decide que es aceptable el estado de integridad del AR.

6B. El TNCS envía cada mensaje de los IMCs a los correspondientes IMVs. Cada IMV analiza dichos mensajes. El IMV necesita avisar al TNCS si necesita comunicarse aún más con un IMC o ya está listo para decidir su recomendación (IMV Action-Recommendation) y su resultado de la evaluación (IMV Evaluation-Result).

6C. Similarmente, el TNCC reenviará mensajes del TNCS a los IMCs correspondientes.

7. Cuando el TNCS completa el ICH con el TNCC, manda su recomendación (TNCS Action-Recommendation) al NAA, aunque el NAA todavía tiene la opción de no permitir el acceso.

8. El NAA manda su decisión al PEP para su ejecución. También debe indicar su decisión al TNCS, la cual será enviada al TNCC. Normalmente el PEP indica su ejecución de la decisión al NAR. Envía identificador de VLAN a donde conectarse y su dirección IP.

9. La VLAN donde se conecta es la de Remedio, donde se tendrá que actualizar dependiendo del resultado de la recomendación de TNCS.

10. Después de permanecer en remedio, deberá comenzar de nuevo el proceso de autenticación.

4.2.3.3. Autenticación no exitosa

Se considera que se tiene una petición de conexión, fallando la autenticación de usuario. Se siguen los pasos:

0. Antes de comenzar una conexión, el TNCC debe detectar y cargar cada IMC, verificando su integridad y una conexión segura. Similarmente, el TNCS debe reconocer y cargar cada IMV.
1. Cuando se inicia un intento de conexión, el NAR inicia una petición de conexión en las capas de enlace y red.

2. Cuando recibe la petición el PEP, manda una petición de decisión de acceso a la red hacia el NAA. Se configura el NAA para ejecutar autenticación de usuario (User Authentication), autenticación de credenciales de plataforma (Platform Credential Authentication) y verificación de integridad (Integrity Check Handshake). La autenticación de usuario se da entre NAA y AR; las otras dos entre AR y TNCS.

3. Como no tiene éxito la autenticación del usuario, el NAA no informa al TNCS la petición de conexión. Por lo que termina el proceso.

4.2.3.4. Autenticación exitosa con proceso de cuarentena

Se considera que se tiene una autenticación exitosa pero ya estando dentro de la red, se presenta incumplimiento de políticas. Se siguen los pasos:

0. Antes de comenzar una conexión, el TNCC debe detectar y cargar cada IMC, verificando su integridad y una conexión segura. Similarmente, el TNCS debe reconocer y cargar cada IMV.
1. Cuando se inicia un intento de conexión, el NAR inicia una petición de conexión en las capas de enlace y red.

2. Cuando recibe la petición el PEP, manda una petición de decisión de acceso a la red hacia el NAA. Se configura el NAA para ejecutar autenticación de usuario (User Authentication), autenticación de credenciales de plataforma (Platform Credential Authentication) y verificación de integridad (Integrity Check Handshake). La autenticación de usuario se da entre NAA y AR; las otras dos entre AR y TNCS.

3. Si tiene éxito la autenticación del usuario, el NAA informa al TNCS la petición de conexión.

4. Entonces TNCS ejecuta la autenticación (mutua) de las credenciales de la plataforma con el TNCC.

5. Si tiene éxito esa autenticación entre TNCS y TNCC, el TNCS indica a los IMVs que hay una nueva petición de conexión y que se necesita llevar a cabo la verificación de integridad. Similarmente el TNCC se los indica a los IMCs.

6A. Para que ocurra la verificación de integridad (Integrity Check Handshake), el TNCS y el TNCC comienzan el intercambio de mensajes de verificación de integridad. Esos mensajes se transmiten a través del NAR, PEP y NAN, y continúan hasta que el TNCS decide que es aceptable el estado de integridad del AR.

6B. El TNCS envía cada mensaje de los IMCs a los correspondientes IMVs. Cada IMV analiza dichos mensajes. El IMV necesita avisar al TNCS si necesita comunicarse aún más con un IMC o ya está listo para decidir su recomendación (IMV Action-Recommendation) y su resultado de la evaluación (IMV Evaluation-Result).

6C. Similarmente, el TNCC reenviará mensajes del TNCS a los IMCs correspondientes.

7. Cuando el TNCS completa el ICH con el TNCC, manda su recomendación (TNCS Action-Recommendation) al NAA, aunque el NAA todavía tiene la opción de no permitir el acceso.

8. El NAA manda su decisión al PEP para su ejecución. También debe indicar su decisión al TNCS, la cual será enviada al TNCC. Normalmente el PEP indica su ejecución de la decisión al NAR. Envía identificador de VLAN a donde conectarse y su dirección IP.

9. El proceso de monitoreo interno de la red, alerta que el usuario no cumple con al menos una política, se desconecta de su VLAN y se envía a la VLAN de Cuarentena.

10. Después de estar en cuarentena y mantener un proceso de limpieza, se debe comenzar de nuevo el proceso completo de autenticación.

4.3. Software

tnc@fhh

El grupo de investigación Trust@FHH de la Universidad de Hanover (15), Alemania, está trabajando en el área de Cómputo Confiable, siguiendo la línea de Trusted Computing Group. Se enfoca principalmente en la arquitectura Trusted Network Connect.

El proyecto específico que desarrolla el grupo Trust@FHH, se denomina tnc@fhh [2], el cual es una implementación de la arquitectura TNC basada en software libre; contiene las capas y componentes básicos de TNC, así como las interfaces entre ellos. El servidor TNC se ejecuta como una extensión de FreeRADIUS, permite tener varios pares IMC/IMV, manejo de políticas básicas y está implementado con C++.

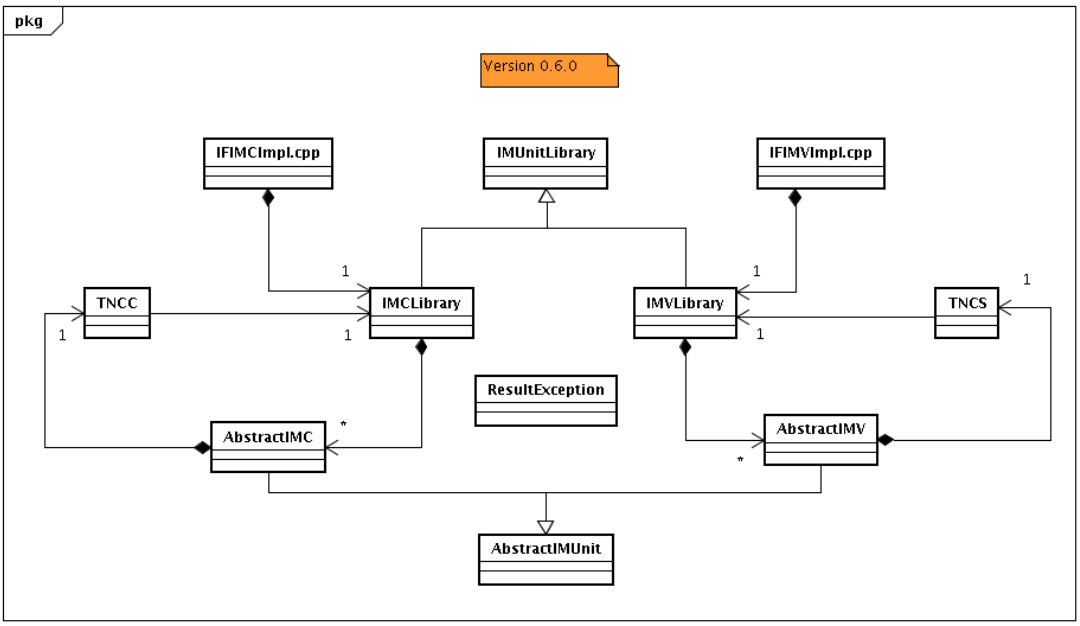


Fig. 1. Diagrama de clases del proyecto tnc@fhh

[2] http://trust.inform.fh-hannover.de/joomla/index.php/projects/tncfhh

Cuenta con un paquete para desarrolladores que facilita la creación de pares IMC/IMV; se denomina imunit. Tiene clases abstractas, instancias y funciones específicas de TNCC y TNCS, basadas en los protocolos dados en TNC. En este trabajo de tesis se utiliza dicho paquete para agregar la funcionalidad de enviar datos biométricos dentro del protocolo de autenticación, mediante el túnel seguro de comunicación EAP-TTLS/EAP-TNC.

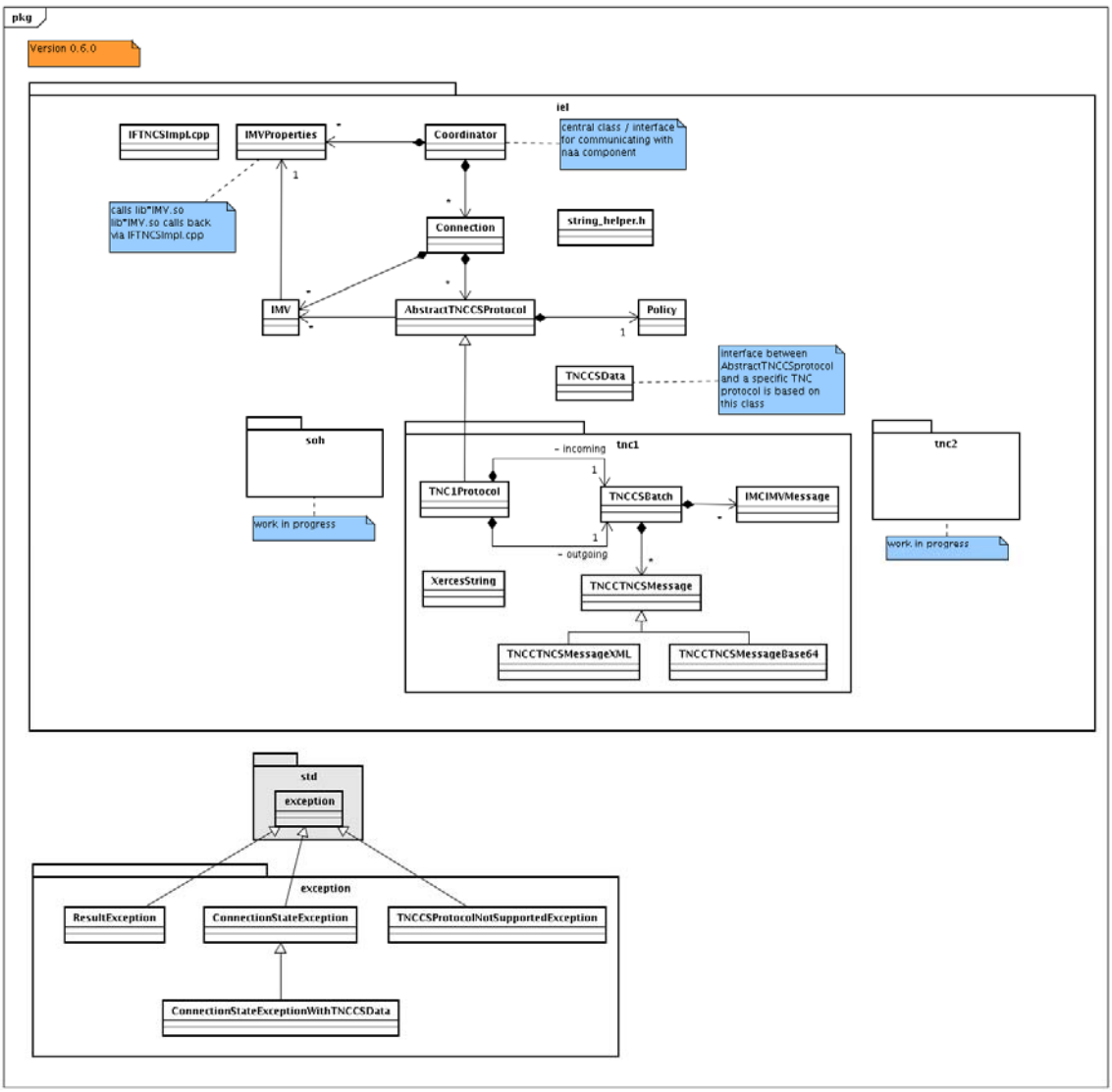


Fig. 2. Diagrama de clase (Figura obtenida de http://trust.inform.fh-hannover.de/)

Además se añaden nuevas medidas de integridad para revisar al cliente mediante la creación de un nuevo para IMC/IMV.

Libfprint

Libfprint es una biblioteca de software libre diseñado para hacer aplicaciones con soporte para lectores de huella dactilar bajo licencia GPL. Ofrece un API para desarrolladores para tener soporte de dispositivos lectores de huella dactilar, procesamiento de imagen de huella, proceso de correlación o empate de imágenes y registro. Dicha biblioteca se maneja como un objeto DSO (Dynamic Shared Object), con lo que se liga la biblioteca en tiempo de compilación y el ligador (linker) completa la liga dinámica en tiempo de ejecución. Depende de libusb y glib de la plataforma Linux.

El procesamiento de la imagen la realiza por medio de MINDTCT, que es un detector de minucias de una imagen de huella dactilar. Las minucias entonces son almacenadas bajo un formato especial de NIST, en los campos 5-12 de un registro Tipo 9.

Es importante señalar que no todos los lectores biométricos son soportados por la librería libfprint. Verificar en http://reactivated.net/fprint/wiki/Supported_devices.

FreeRADIUS

FreeRADIUS incluye una implementación de un servidor RADIUS, la cual tiene un módulo que permite el manejo de los paquetes EAP. Existe un parche especial para EAP-TNC.

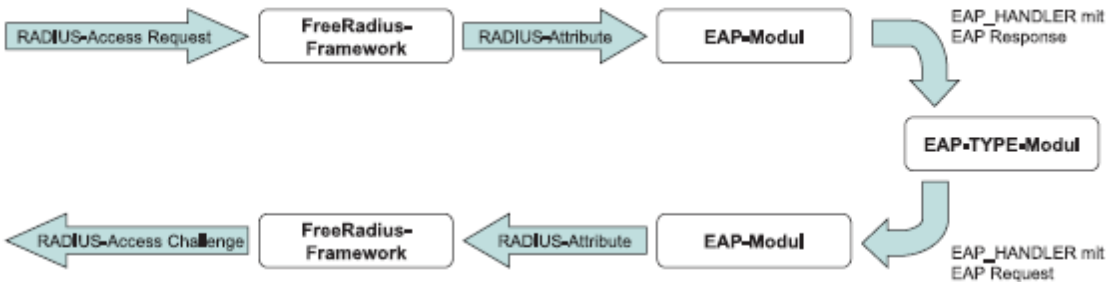


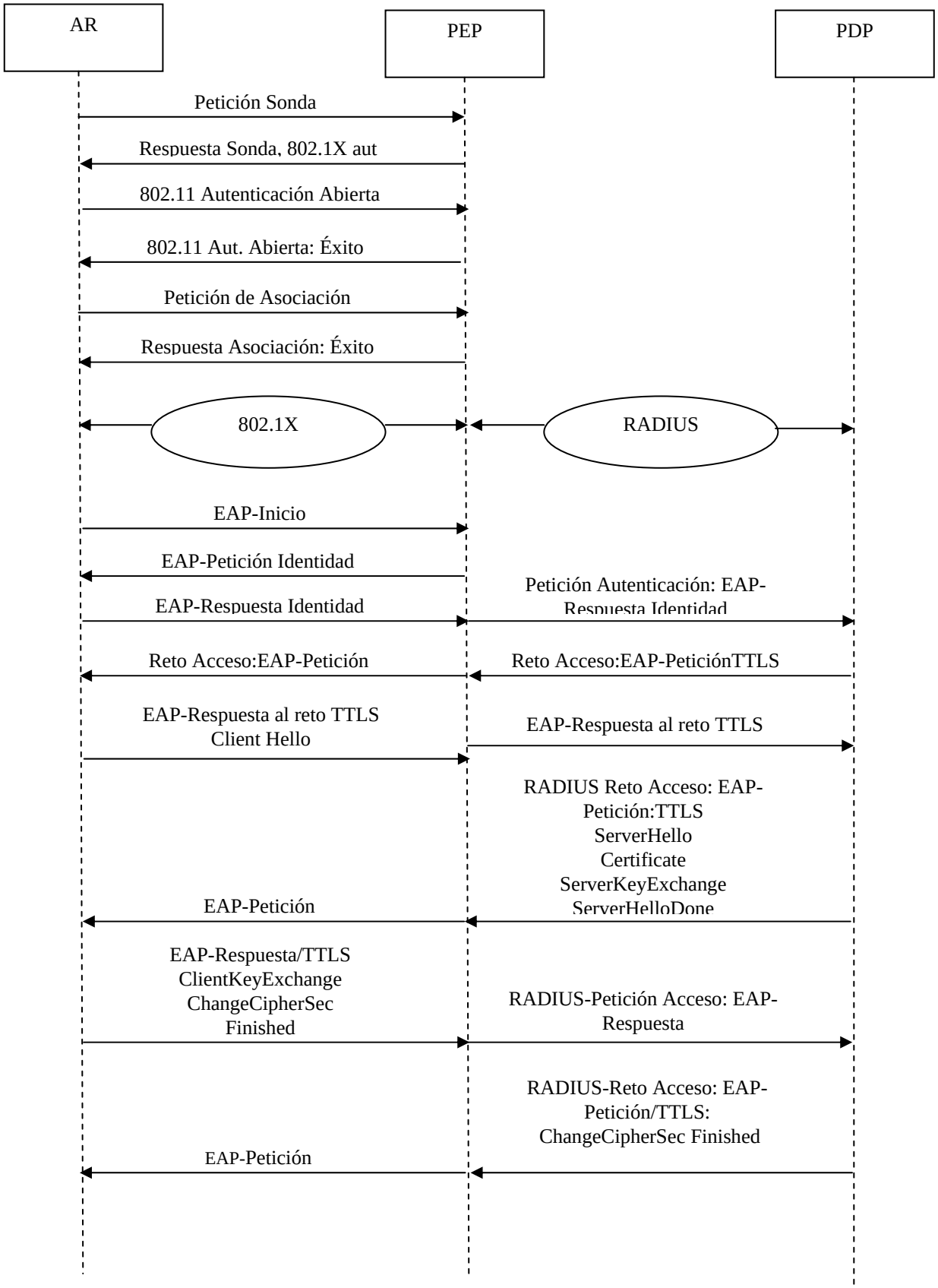
Fig. 3. Módulo EAP para FreeRADIUS

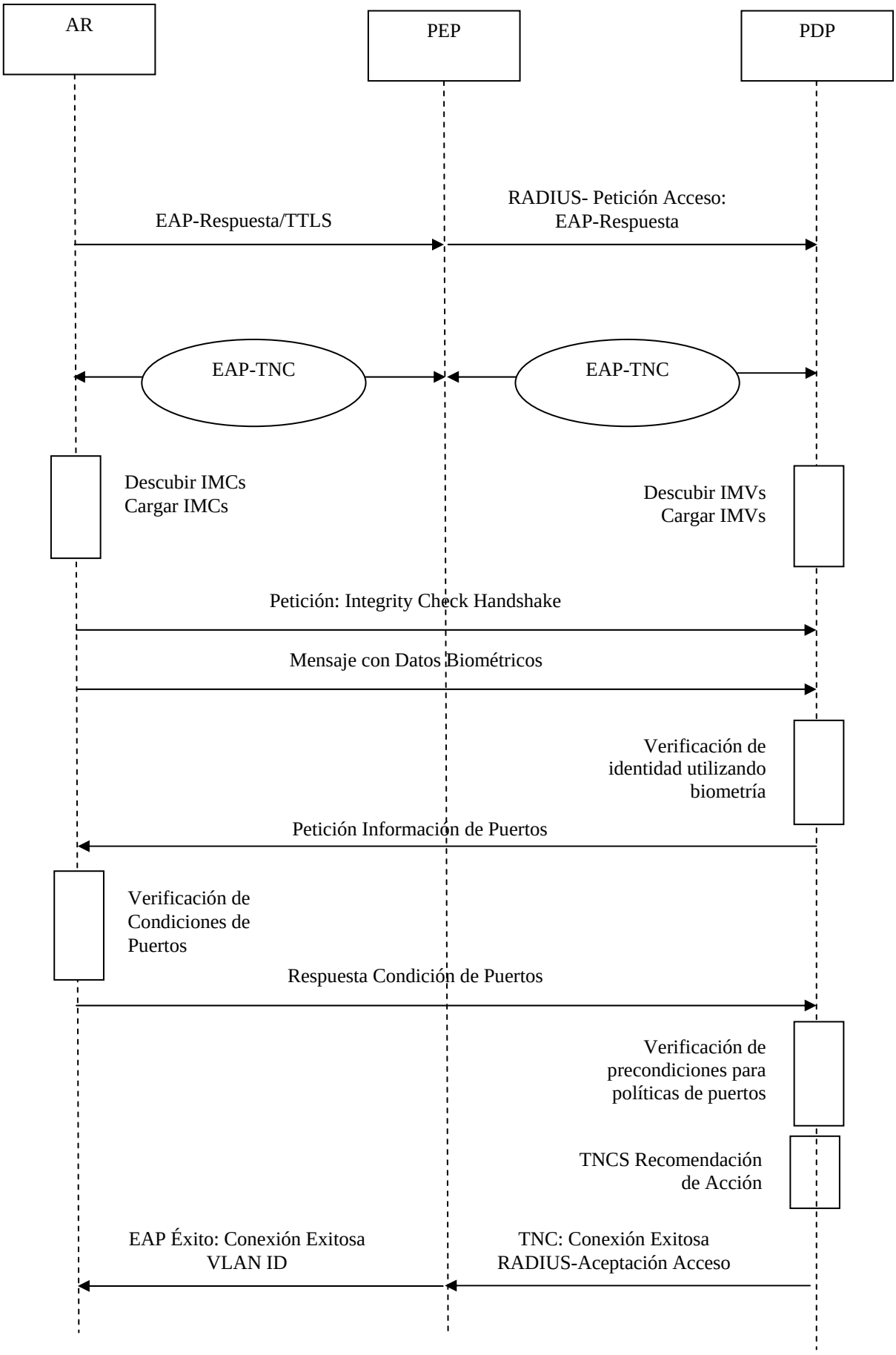
5. Desarrollo

Este proyecto es el desarrollo de un sistema para tener la infraestructura básica para la prueba de concepto de NAC y el envío de datos biométricos dentro de un protocolo de autenticación. Es una implementación de la arquitectura TNC, comprobando la factibilidad de utilizar software libre, sin la necesidad de seguir la solución de un solo vendedor.

5.1. Protocolo general

El siguiente esquema muestra los mensajes enviados entre los tres actores principales de la arquitectura TNC. Se comienza con petición de conexión y petición de asociación, con sus respectivas respuestas por parte del PEP. Después comienza el protocolo 802.1X entre el AR y el PEP y del protocolo RADIUS entre el PEP y el PDP, donde se inicia el protocolo EAP, con petición de identidad, petición de autenticación y envío de reto. Cuando se ha identificado al usuario comienza el protocolo EAP-TNC, ya dentro del túnel formado de EAP-TTLS, donde de hace la verificación de seguridad del dispositivo y el envío de datos biométricos, hasta que finalmente se da la respuesta de rechazo o aceptación, con el identificador de VLAN correspondiente.


AR
PEP
PDP
Petición Sonda
Respuesta Sonda, 802.1X aut
802.11 Autenticación Abierta
802.11 Aut. Abierta: Éxito
Petición de Asociación
Respuesta Asociación: Éxito
802.1X
RADIUS
EAP-Inicio
EAP-Petición Identidad
EAP-Respuesta Identidad
Petición Autenticación: EAP-Respuesta Identidad
Reto Acceso:EAP-Petición
Reto Acceso:EAP-PeticiónTTLS
EAP-Respuesta al reto TTLS
Client Hello
EAP-Respuesta al reto TTLS
RADIUS Reto Acceso: EAP-Petición:TTLS
ServerHello
Certificate
ServerKeyExchange
ServerHelloDone
EAP-Petición
EAP-Respuesta/TTLS
ClientKeyExchange
ChangeCipherSec
Finished
RADIUS-Petición Acceso: EAP-Respuesta
RADIUS-Reto Acceso: EAP-Petición/TTLS:
ChangeCipherSec Finished
EAP-Petición


AR
PEP
PDP
EAP-Respuesta/TTLS
RADIUS- Petición Acceso:
EAP-Respuesta
EAP-TNC
EAP-TNC
Descubir IMCs
Cargar IMCs
Descubir IMVs
Cargar IMVs
Petición: Integrity Check Handshake
Mensaje con Datos Biométricos
Verificación de
identidad utilizando
biometría
Petición Información de Puertos
Verificación de
Condiciones de
Puertos
Respuesta Condición de Puertos
Verificación de
precondiciones para
políticas de puertos
TNCS Recomendación
de Acción
EAP Éxito: Conexión Exitosa
VLAN ID
TNC: Conexión Exitosa
RADIUS-Aceptación Acceso

5.2. Implementación

A continuación se describe cómo se implementó cada uno de los elementos de la arquitectura TNC utilizada para este proyecto. Se incluyen las modificaciones necesarias en las configuraciones de los paquetes de software utilizados, para que cumplieran con los requisitos de la arquitectura.

AR

Los tres elementos que conformaron al Access Requestor, son los siguientes:

1. Wpa_supplicant. Se necesita realizar un cambio en el archivo de configuración .config para tener soporte para EAP-TNC (Ver Fig. 28).

```
# EAP-TNC and related Trusted Network Connect support (experimental)
CONFIG_EAP_TNC=y
```

Fig. 1. Cambio necesario en configuración de wpa_supplicant

2. Cliente TNC. Se utilizan las librerías del proyecto tnc@fhh.

3. IMCs. El objeto SO generado para cada IMC debe ser instalado dentro de /usr/local/lib y colocar su definición dentro del archivo /etc/tnc_config (Ver Fig.29).

```
###IMC que envía datos biométricos dentro de EAP
IMC "BIO_IMC" /usr/local/lib/libBIO_IMC.so
###IMC que verifica estado de puertos del dispositivo
IMC "HostScanner" /usr/local/lib/libHostScannerIMC.so
```

Fig. 2. Definición de IMCs

Se tienen dos pares IMC/IMV, uno que verifica la identidad del usuario y otro que verifica el estado de los puertos del dispositivo.

PEP

El PEP solamente está conformado por un switch.

1. Configuración Switch Cisco Catalyst 2950. Soporte para VLANs (Ver Fig. 30).

```
aaa                                                    new-model
aaa       group       server       radius       freeradius
server    10.0.0.10    auth-port    1812    acct-port    1813
dot1x system-auth-control
vlan                                                      96
name                                                  usuario
mtu 1400
vlan                                                      97
name                                                cuarentena
mtu 1400    […]
```

Fig. 3. Configuración básica para switch

PDP

Los elementos que conforman al servidor PDP son los siguientes:

1. Servidor FreeRADIUS. Se necesita la instalación de un parche especial para convertirse en PDP, con la utilización de las librerías TNC y EAP-TNC (Ver Fig. 31 y 32)

```
eap {
        # ...
        default_eap_type = ttls  # ...
        tnc {
        }                        # ...
        ttls {
                default_eap_type = tnc          #...
                use_tunneled_reply = yes
                virtual_server = "inner-tunnel-tnc"
        }# ...
    }
```

Fig. 4. Cambios en configuración de FreeRADIUS

```
#usr/local/etc/raddb/sites-enabled/inner-tunnel-tnc

# change the following line
# server inner-tunnel {
# to
server inner-tunnel-tnc {

post-auth {
[…]
if (control:TNC-Status == "Access") {
         update reply {
                  Tunnel-Type := VLAN
                  Tunnel-Medium-Type := IEEE-802
                  Tunnel-Private-Group-ID := 96
         }
}   # Se hace un elsif para cada VLAN creada
}
[…]
}
```

Fig. 5. Cambios en configuración FreeRADIUS (2)

2. Servidor TNC. Se utilizan las librerías del proyecto tnc@fhh.
3. IMVs. El objeto SO generado para cada IMV debe ser instalado dentro de /usr/local/lib y colocar su definición dentro del archivo /etc/tnc_config (Ver Fig.33).

```
###IMV que recibe datos biométricos desde EAP
IMV "BIO_IMV" /usr/local/lib/libBIO_IMV.so
###IMV que verifica estado de puertos del dispositivo
IMV "HostScanner" /usr/local/lib/libHostScannerIMV.so
```

Fig. 6. Definición de IMVs

Se tiene un archivo con las políticas defnidas anteriormente para verificar el estado de los puertos del dispositivo, que lee el IMV de HostScanner.

5.3. Dispositivos físicos

Los dispositivos utilizados para el desarrollo de este proyecto son una laptop, con el sistema operativo Ubuntu Intrepid, un lector biométrico, un switch y una computadora de escritorio, con el sistema operativo Ubuntu Hardy.

En la siguiente tabla se describen las características de cada dispositivo:

Tabla 1. Dispositivos físicos

AR • Laptop Dell Studio 1535 • Ubuntu Intrepid • Lector biométrico de huella dactilar Microsoft USB	 
PEP • Conmutador Cisco Catalyst 2950 **PDP**	
• PD Dell Optiplex • Ubuntu Hardy	

5.4. Base de datos

La base de datos de los usuarios registrados estará dentro del servidor PDP. Se registra nombre de usuario, contraseña y vector con el descriptor de la huella dactilar de un solo dedo (esto es por simplicidad del sistema para este proyecto).

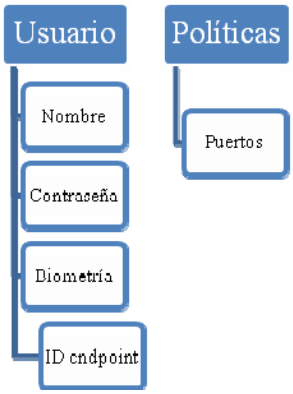


Fig. 7. Base de datos

Las políticas utilizadas para la verificación de integridad de los dispositivos se almacenan dentro del servidor PDP, por ejemplo, el archivo donde se encuentran definidos los estados de los puertos a verificar.

6. Resultados

El resultado esperado en este proyecto es poder realizar la autenticación del usuario utilizando un mecanismo de biometría, dentro del protocolo EAP-TTLS. Fue posible enviar el vector completo del descriptor de la huella dactilar dentro del túnel cifrado, encapsulado en un paquete EAP-TNC, como si fuera un mensaje. Esto permite autenticar al usuario antes de que se le asigne a una VLAN y que tenga acceso a los recursos de la red. Es decir, la autenticación biométrica se realiza utilizando la comunicación con el puerto no controlado, bajo flujo EAP.

Hasta el momento, no existe un protocolo EAP que envíe datos biométricos, permitiendo la autenticación del usuario.

6.1. Suplicante de Acceso

A continuación se muestra la comunicación del lado del cliente (AR), bajo el log de wpa_supplicant. Se observan las fases principales del protocolo seguido en la arquitectura TNC.
Por facilidad de lectura se omiten algunos mensajes, cuando se escribe […], y se acortan los valores hexadecimales enviados en este protocolo.

```
Initializing interface 'eth0' conf
'/etc/wpa_supplicant/TNC_wpa_supplicant_TTLS.conf' driver 'wired'
ctrl_interface 'N/A' bridge 'N/A'
Configuration file '/etc/wpa_supplicant/TNC_wpa_supplicant_TTLS.conf' -> […]
```

Fig.35. Inicialización de wpa_supplicant

```
Line: 4 - start of a new network block
ssid - hexdump_ascii(len=7):
     53 77 69 74 63 68 4e                              SwitchN
key_mgmt: 0x8
eap methods - hexdump(len=16): 00 00 00 00 15 00 00 00 00 00 00 00 00 00 00 00
identity - hexdump_ascii(len=3):
     65 6e 65                                          ene
password - hexdump_ascii(len=3): [REMOVED]
ca_cert - hexdump_ascii(len=31):
     2f 68 6f 6d 65 2f 65 6e 65 74 7a 79 6f 2f 45 73   /home/enetzyo/Es
     63 72 69 74 6f 72 69 6f 2f 63 61 2e 70 65 6d      critorio/ca.pem
id_str - hexdump_ascii(len=0):
eapol_flags=0 (0x0)
Priority group 0
   id=0 ssid='SwitchN'
Initializing interface (2) 'eth0'    […]
```

Fig.36. Reconocimiento de autenticador y su certificado digital

```
State: DISCONNECTED -> ASSOCIATED
Associated to a new BSS: BSSID=01:80:c2:00:00:03
[…]
EAPOL: SUPP_BE entering state REQUEST
EAPOL: getSuppRsp
EAP: EAP entering state RECEIVED
EAP: Received EAP-Request id=0 method=1 vendor=0 vendorMethod=0
EAP: EAP entering state IDENTITY
```

Fig.37. Asociación e inicio de EAP

```
EAP: EAP-Request Identity data - hexdump_ascii(len=0):
EAP: using real identity - hexdump_ascii(len=3):
     65 6e 65                                          ene
EAP: EAP entering state SEND_RESPONSE
[…]
 EAP: Initialize selected EAP method: vendor 0 method 21 (TTLS)
EAP-TTLS: Phase2 type: EAP
TLS: Trusted root certificate(s) loaded
EAP-TTLS: Start
SSL: SSL_connect:before/connect initialization
```

Fig.38. Envío de identidad del usuario e inicio de EAP-TTLS

Ya que se tiene el túnel TTLS, que será el túnel exterior, se inicia con la parte que se añade al protocolo, que es la utilización del túnel interno EAP-TNC. Se tienen varios pasos como se puede observar a continuación.

a. Se encuentra la ruta de cada uno de los IMCs (Ver. Fig. 39).

```
TNC: Configured IMC: "HostScanner" /usr/local/lib/libHostScannerIMC.so
TNC: Name: 'HostScanner'
TNC: IMC file: '/usr/local/lib/libHostScannerIMC.so'
TNC: Configured IMC: "BIO_IMC" /usr/local/lib/libBIO_IMC.so
TNC: Name: 'BIO_IMC'
TNC: IMC file: '/usr/local/lib/libBIO_IMC.so'
```

Fig.39. Reconocimiento de IMCs

b. Se inicializa cada IMC junto con sus funciones definidas (Ver Fig. 40):

```
TNC: Opening IMC: HostScanner (/usr/local/lib/libHostScannerIMC.so)
0 [0xb7d498c0] DEBUG IMUnit.IMUnitLibrary null - no log4cxx
configuration file. using basic configuration
0 [0xb7d498c0] INFO IMUnit.IMUnitLibrary null - Load imunit-Library
version 0.6.0
0 [0xb7d498c0] INFO
IMUnit.IMUnitLibrary.IMCLibrary.HostScannerIMCLibrary null - Load
HostScannerIMC library version 0.6.0
TNC: Calling TNC_IMC_Initialize for IMC 'HostScanner'
0 [0xb7d498c0] INFO IMUnit.IMUnitLibrary null –

HostScannerIMC::initialize(0, 1, 1)
TNC: TNC_IMC_Initialize: res=0 imc_ver=1
TNC: Calling TNC_IMC_ProvideBindFunction for IMC 'HostScanner'
0 [0xb7d498c0] DEBUG IMUnit.IMUnitLibrary.IMCLibrary null -
HostScannerIMC::provideBindFunction(0, 1)
TNC: TNC_TNCC_BindFunction(imcID=0,
functionName='TNC_TNCC_ReportMessageTypes')
TNC: TNC_TNCC_BindFunction(imcID=0,
functionName='TNC_TNCC_RequestHandshakeRetry')
TNC: TNC_TNCC_BindFunction(imcID=0,
functionName='TNC_TNCC_SendMessage')
TNC: TNC_TNCC_ReportMessageTypes(imcID=0 typeCount=1)
TNC: supportedTypes[0] = 8432432
TNC: TNC_IMC_ProvideBindFunction: res=0   [...]
EAP-TNC: Received packet: Flags 0x21 Message Length 0
```

Fig. 40. Inicialización del IMC HostScanner

c. Se inicia el handshake para cada IMC (Ver Fig. 41).

```
HostScannerIMC::notifyConnectionChange(0, 0, 1)
TNC: TNC_IMC_NotifyConnectionChange: 0
TNC: Calling TNC_IMC_BeginHandshake for IMC 'HostScanner'
1 [0xb7d498c0] DEBUG IMUnit.IMUnitLibrary.IMCLibrary null -
HostScannerIMC::beginHandshake(0, 0)
1 [0xb7d498c0] DEBUG IMUnit.AbstractIMUnit.AbstractIMC.HostScannerIMC
null - Send Message: HostScannerIMC active
TNC: TNC_TNCC_SendMessage(imcID=0 connectionID=0 messageType=8432432)
TNC: TNC_TNCC_SendMessage - hexdump_ascii(len=21):
     48 6f 73 74 53 63 61 6e 6e 65 72 49 4d 43 20 61   HostScannerIMC a
     63 74 69 76 65                                    ctive
TNC: TNC_IMC_BeginHandshake: 0
```

Fig.41. BeginHandshake para IMC HostScanner

d. Se manda el primer conjunto de mensajes TNCCS, utilizando una plantilla XML, donde se envía el tipo de mensajes que se utilizaran en la comunicación.
Se mandan los mensajes a través del túnel TTLS (Ver Fig.42).

```
EAP-TNC: Response - hexdump_ascii(len=568):
     3c 3f 78 6d 6c 20 76 65 72 73 69 6f 6e 3d 22 31   <?xml version="1
     2e 30 22 3f 3e 0a 3c 54 4e 43 43 53 2d 42 61 74   .0"?>_<TNCCS-Bat
     63 68 20 42 61 74 63 68 49 64 3d 22 31 22 20 52   ch BatchId="1" R
     65 63 69 70 69 65 6e 74 3d 22 54 4e 43 53 22 20   ecipient="TNCS"
     […]
     42 61 73 65 36 34 3e 3c 2f 49 4d 43 2d 49 4d 56   Base64></IMC-IMV
     2d 4d 65 73 73 61 67 65 3e 0a 3c 2f 54 4e 43 43   -Message>_</TNCC
     53 2d 42 61 74 63 68 3e                           S-Batch>
EAP-TNC: Generating Response
EAP-TNC: Sending out 568 bytes (message sent completely)
EAP-TTLS: AVP encapsulate EAP Response - hexdump(len=574): 02 01 02 3e
[…]
```

Fig.42. Envío de mensaje a través de XML

e. Se envía el dato biométrico dentro de un mensaje de BIO_IMC (Ver Fig.43). Es importante recalcar que se envían aproximadamente 2KB de datos biométricos dentro de un paquete EAP por lo que se utiliza la opción de fragmentación de paquetes, manejado por EAP-TTLS. También se hicieron pruebas al enviar el dato biométrico comprimido utilizando la herramienta GZIP, obteniendo el mismo resultado de autenticación al descomprimirlo en el servidor PDP pero disminuyendo el número de paquetes.

```
TNC: Message to IMC(s) - hexdump_ascii(len=29):
     45 78 61 6d 70 6c 65 20 6d 65 73 73 61 67 65 20   Example message
     66 72 6f 6d 20 42 49 4f 5f 49 4d 56 00            from BIO_IMV_
TNC: Call ReceiveMessage for IMC 'BIO_IMC'
27 [0xb7c888c0] DEBUG IMUnit.IMUnitLibrary.IMCLibrary null -
BIO_IMC::receiveMessage(imcID=0, conID=0, messageBuffer=0x0x81efb88,
messageLength=1d, messageType=0x000abcfe)
27 [0xb7c888c0] DEBUG IMUnit.AbstractIMUnit.AbstractIMC.BIO_IMC null -
receiveMessage round 1
28 [0xb7c888c0] DEBUG IMUnit.AbstractIMUnit.AbstractIMC.BIO_IMC null -
Send Message ENVIANDO ARCHIVO: Length = 2414 Bytes.
TNC: TNC_TNCC_SendMessage(imcID=0 connectionID=0 messageType=703742)
TNC: TNC_TNCC_SendMessage - hexdump_ascii(len=2414):
     46 50 31 02 00 00 00 00 00 01 3f 00 00 00 13 00   FP1_______?_____
     00 00 39 00 00 00 3e 00 00 00 41 00 00 00 4e 00   __9___>___A___N_

     […] (Aquí se envían 2KB del descriptor matemático de la huella)

     00 00 05 00 00 00 05 00 00 00 05 00 00 00         ______________
TNC: ReceiveMessage: 0
```

Fig.43. Envío de dato biométrico como mensaje EAP-TNC

f. Se envía la petición del estado de los puertos del dispositivo mediante el IMC HostScanner (Ver Fig. 44).

```
TNC: IMC-IMV-Message Type 0x80ab30
TNC: Message to IMC(s) - hexdump_ascii(len=137):
     54 43 50 20 32 30 0a 54 43 50 20 32 31 0a 54 43   TCP 20_TCP 21_TC
     50 20 32 32 0a 54 43 50 20 32 33 0a 54 43 50 20   P 22_TCP 23_TCP
     32 35 0a 54 43 50 20 35 38 37 0a 54 43 50 20 31   25_TCP 587_TCP 1
     31 30 0a 54 43 50 20 39 39 35 0a 55 44 50 20 35   10_TCP 995_UDP 5
     33 0a 54 43 50 20 35 33 0a 55 44 50 20 36 37 0a   3_TCP 53_UDP 67_
     55 44 50 20 36 38 0a 54 43 50 20 38 30 0a 54 43   UDP 68_TCP 80_TC
     50 20 34 34 33 0a 54 43 50 20 38 30 38 30 0a 54   P 443_TCP 8080_T
     43 50 20 35 32 32 33 0a 55 44 50 20 34 34 34 34   CP 5223_UDP 4444
     0a 55 44 50 20 36 33 31 0a                        _UDP 631_
TNC: Call ReceiveMessage for IMC 'HostScanner'
```

Fig.44. Envío de petición del estado de puertos

Después de enviar el número de los puertos requeridos, se revisa el estado de cada uno de los puertos de la solicitud en el endpoint (Ver Fig. 45).

```
TNC: Call ReceiveMessage for IMC 'HostScanner'
29 [0xb7c888c0] DEBUG IMUnit.IMUnitLibrary.IMCLibrary null -
HostScannerIMC::receiveMessage(imcID=1, conID=0, messageBuffer=0x0x81f5cc0,
messageLength=89, messageType=0x0080ab30)
29 [0xb7c888c0] DEBUG IMUnit.AbstractIMUnit.AbstractIMC.HostScannerIMC null
- Receive Message. Length = 137 Bytes
30 [0xb7c888c0] DEBUG IMUnit.AbstractIMUnit.AbstractIMC.HostScannerIMC null
- TCP Port 20(ftp-data) close
31 [0xb7c888c0] DEBUG IMUnit.AbstractIMUnit.AbstractIMC.HostScannerIMC null
- TCP Port 21(ftp) close
31 [0xb7c888c0] DEBUG IMUnit.AbstractIMUnit.AbstractIMC.HostScannerIMC null
- TCP Port 22(ssh) close
         […]
```

Fig.45. Reporte de estado de puertos dentro del endpoint

Se forma el mensaje con este reporte de los puertos para poder enviarlo hacia el IMV (Ver Fig.46).

```
TNC: TNC_TNCC_SendMessage(imcID=1 connectionID=0 messageType=8432432)
TNC: TNC_TNCC_SendMessage - hexdump_ascii(len=280):
     54 43 50 20 32 30 20 3d 20 63 6c 6f 73 65 0a 54   TCP 20 = close_T
     43 50 20 32 31 20 3d 20 63 6c 6f 73 65 0a 54 43   CP 21 = close_TC
     50 20 32 32 20 3d 20 63 6c 6f 73 65 0a 54 43 50   P 22 = close_TCP
     20 32 33 20 3d 20 63 6c 6f 73 65 0a 54 43 50 20    23 = close_TCP
     32 35 20 3d 20 63 6c 6f 73 65 0a 54 43 50 20 35   25 = close_TCP 5
     […]
TNC: ReceiveMessage: 0
```

Fig.46. Mensaje con el reporte de puertos

g. Finalmente se envía la Recomendación TNC (Ver Fig.47), en este caso se observa ALLOW, por lo que EAP manda un mensaje de autenticación exitosa (Ver Fig. 48).

```
TNC: TNCC-TNCS-Message Type 0x1
TNC: TNCC-TNCS-Message XML - hexdump_ascii(len=48):
     0a 20 20 20 20 20 20 3c 54 4e 43 43 53 2d 52 65   _      <TNCCS-Re
     63 6f 6d 6d 65 6e 64 61 74 69 6f 6e 20 74 79 70   commendation typ
     65 3d 22 61 6c 6c 6f 77 22 2f 3e 0a 20 20 20 20   e="allow"/>_
TNC: TNCCS-Recommendation: 'allow'
TNC: Calling TNC_IMC_NotifyConnectionChange(2) for IMC 'HostScanner'
61 [0xb7d498c0] DEBUG IMUnit.IMUnitLibrary.IMCLibrary null -
HostScannerIMC::notifyConnectionChange(0, 0, 2)
TNC: TNC_IMC_NotifyConnectionChange: 0
TNC: Calling TNC_IMC_NotifyConnectionChange(2) for IMC 'BIO_IMC'
61 [0xb7d498c0] DEBUG IMUnit.IMUnitLibrary.IMCLibrary null -
BIO_IMC::notifyConnectionChange(1, 0, 2)
TNC: TNC_IMC_NotifyConnectionChange: 0
TNC: Recommendation = allow
```

Fig.47. Mensaje de recomendación TNC

```
EAP-TNC: TNCS done - reply with an empty ACK message
EAP-TTLS: AVP encapsulate EAP Response - hexdump(len=6): 02 03 00 06
26 01
EAP-TTLS: Encrypting Phase 2 data - hexdump(len=16): [REMOVED]
SSL: 90 bytes left to be sent out (of total 90 bytes)
EAP-TTLS: Authentication completed successfully (MAY_CONT) […]
[…]
EAPOL: IEEE 802.1X for plaintext connection; no EAPOL-Key frames
required
WPA: EAPOL processing complete
Cancelling authentication timeout
State: ASSOCIATED -> COMPLETED
CTRL-EVENT-CONNECTED - Connection to 01:80:c2:00:00:03 completed
(auth) [id=0 id_str=]
EAPOL: SUPP_PAE entering state AUTHENTICATED
EAPOL: SUPP_BE entering state RECEIVE
EAPOL: SUPP_BE entering state SUCCESS
EAPOL: SUPP_BE entering state IDLE
EAPOL authentication completed successfully
EAPOL: startWhen --> 0
```

Fig.48. Mensajes EAP de autenticación exitosa

Cuando se tiene el mensaje EAP de autenticación exitosa, ya es posible obtener una dirección IP de acuerdo a la VLAN que sea asignada.

Finalmente, cuando se termina la comunicación entre AR y PDP es necesario eliminar de memoria las librerías de los IMCs (Ver Fig.49).

```
TNC: Calling TNC_IMC_Terminate for IMC 'HostScanner'
10168 [0xb7d498c0] DEBUG IMUnit.IMUnitLibrary.IMCLibrary null -
HostScannerIMC::terminate(0)
TNC: TNC_IMC_Terminate: 0
TNC: Calling TNC_IMC_Terminate for IMC 'BIO_IMC'
10168 [0xb7d498c0] DEBUG IMUnit.IMUnitLibrary.IMCLibrary null -
BIO_IMC::terminate(1)
TNC: TNC_IMC_Terminate: 0
10169 [0xb7d498c0] INFO IMUnit.IMUnitLibrary null - Unload Library
HostScannerIMC
10169 [0xb7d498c0] INFO IMUnit.IMUnitLibrary null - Unload Library BIO_IMC
```

Fig.49. Eliminar de memoria las librerías IMCs

6.2. Punto de Decisión de Políticas

El Punto de Decisión de Políticas en este proyecto fue implementado con FreeRADIUS Versión 2.1.6, para la arquitectura i686-pc-linux-gnu. Los resultados en este punto se observan gracias al log de FreeRADIUS como se muestran a continuación.

a. Se cargan a memoria los módulos necesarios, como por ejemplo el módulo EAP (Ver Fig. 50).

```
Module: Linked to module rlm_eap
 Module: Instantiating eap
  eap {
       default_eap_type = "ttls"
       timer_expire = 60
       ignore_unknown_eap_types = no
       cisco_accounting_username_bug = no
       max_sessions = 2048
  }
```

Fig.50. Carga de módulo EAP

b. Se inicializa cada uno de los IMVs y sus funciones (Ver Fig. 51).

```
0 [0xb7c3bac0] DEBUG TNCS.Coordinator null - Loading tncs version 0.6.0
 Module: Linked to sub-module rlm_eap_tnc
 Module: Instantiating eap-tnc
TNC-ATTACH initializing NAA-EAP
1 [0xb7c3bac0] INFO NAA-EAP.naaeap null - initialize naaeap version 0.6.0 with
default tnc_config file.
1 [0xb7c3bac0] INFO TNCS.Coordinator null - Initialize tncs version 0.6.0 ...
2 [0xb7c3bac0] DEBUG TNCS.IMVProperties null - Create IMVProperties
0:"HostScanner" file:"/usr/local/lib/libHostScannerIMV.so"
[…]
3 [0xb7c3bac0] INFO IMUnit.IMUnitLibrary null - HostScannerIMV::initialize(0,
1, 1)
3 [0xb7c3bac0] INFO TNCS.IMVProperties null - HostScanner initialized complete.
3 [0xb7c3bac0] DEBUG IMUnit.IMUnitLibrary.IMVLibrary null -
HostScannerIMV::provideBindFunction(0, 1)
3 [0xb7c3bac0] DEBUG IF-IMV.TNCS null - TNC_TNCS_BindFunction
[…]
4 [0xb7c3bac0] DEBUG IF-IMV.TNCS null - TNC_TNCS_ReportMessageTypes
4 [0xb7c3bac0] DEBUG TNCS.IMVProperties null - Create IMVProperties

1:"BIO_IMV" file:"/usr/local/lib/libBIO_IMV.so" […]
```

Fig.51. Carga de IMV HostScanner

Después de cargar los IMVs se especifican los módulos necesarios y se abren los puertos del servidor, definidos para recibir peticiones.

c. Después de que los puertos para FreeRADIUS están en espera, llega la petición de autenticación del cliente (Ver Fig.52).

```
rad_recv: Access-Request packet from host 10.0.0.1 port 1043, id=19,
length=153
       User-Name = "ene"
       Cisco-AVPair = "ssid=switchN"
       NAS-IP-Address = 10.0.0.1              […]
```

Fig.52. Mensaje de petición de autenticación

d. Comienza el handshake de TTLS y la comunicación entre cliente y servidor (Ver Fig.53).

```
[eap] EAP/ttls
[eap] processing type ttls
[ttls] Authenticate
[ttls] processing EAP-TLS
[ttls] eaptls_verify returned 7
[ttls] Done initial handshake
[ttls]     (other): before/accept initialization
[ttls]     TLS_accept: before/accept initialization
[ttls] <<< TLS 1.0 Handshake [length 0058], ClientHello
[ttls]     TLS_accept: SSLv3 read client hello A
[…]
Sending Access-Challenge of id 20 to 10.0.0.1 port 1044
```

Fig. 53. Handshake TTLS y envío de reto

e. Cuando se tiene el túnel exterior TTSL, se inicia la comunicación mediante el túnel interior EAP-TNC (Ver Fig.54).

```
+- entering group authenticate {...}
[eap] EAP Identity
[eap] processing type tnc
tnc_initiate: 1257880853
NAS scr ip = A000001
NAS scr port = 37
NAS scr port type = 19
TNC-INITIATE getting connection from NAA-EAP
11802 [0xb7bbbac0] INFO NAA-EAP.naaeap null - getConnection input is NAS
Port: 37 NAS IP: 1.0.0.10 NAS_PORT_TYPE: 19
11802 [0xb7bbbac0] DEBUG TNCS.Coordinator null - incoming connection
11802 [0xb7bbbac0] DEBUG TNCS.Coordinator null - create a new connection 0
```

Fig.54. Inicio de comunicación EAP-TNC

f. Ya que se tiene la comunicación EAP-TNC se crea una instancia de cada IMV (Ver Fig. 55).

```
11802 [0xb7bbbac0] DEBUG TNCS.Connection null - Create an IMV HostScanner
for connection 0
11802 [0xb7bbbac0] DEBUG TNCS.IMV null - Creating IMV instance for
IMVProperties HostScanner for connection 0
11802 [0xb7bbbac0] DEBUG IMUnit.IMUnitLibrary.IMVLibrary null -
HostScannerIMV::notifyConnectionChange(0, 0, 0)
11802 [0xb7bbbac0] DEBUG IMUnit.IMUnitLibrary.IMVLibrary null - Library
HostScannerIMV creating IMV instance for conID 0
```

Fig.55. Creación de instancias de IMV HostScanner

g. Dependiendo de cada IMV cargado se realizan diversas acciones, por ejemplo, con el IMV HostScanner se lee el archivo de políticas necesarias para la revisión de puertos (Ver Fig.56). Se envían y reciben los mensajes indispensables para que finalmente se tome una decisión (Ver Fig. 57)

```
IMUnit.AbstractIMUnit.AbstractIMV.HostScannerIMV null - Read Key:"TCP 20"
Value:"whatever"
11803 [0xb7bbbac0] DEBUG IMUnit.AbstractIMUnit.AbstractIMV.HostScannerIMV
null - Read Key:"TCP 21" Value:"close"
11803 [0xb7bbbac0] DEBUG IMUnit.AbstractIMUnit.AbstractIMV.HostScannerIMV
null - Read Key:"TCP 22" Value:"close"
       [...]
```

Fig.56. Lectura de políticas de puertos

```
TNC-AUTHENTICATE is starting now for connection ID 0 !
11830 [0xb7bbbac0] INFO NAA-EAP.naaeap null - processEAPTNCData for
connection 0
[…]
11835 [0xb7bbbac0] DEBUG IMUnit.IMUnitLibrary.IMVLibrary null -
HostScannerIMV::receiveMessage(imvID=0, conID=0, messageBuffer=0x0x82775c0,
messageLength=15, messageType=0x0080ab30)
11835 [0xb7bbbac0] DEBUG IMUnit.AbstractIMUnit.AbstractIMV.HostScannerIMV
null - Receive Message: Length = 21 Bytes.
[…]

BIO_IMV::receiveMessage(imvID=1, conID=0, messageBuffer=0x0x8277598,
messageLength=23, messageType=0x000abcfe)
11835 [0xb7bbbac0] INFO IMUnit.AbstractIMUnit.AbstractIMC.BIO_IMV null -
Received Message: Recibiendo archivo biometrico!!from BIO_IMC
[…]
HostScannerIMV::batchEnding(0, 0)
[…]
BIO_IMV::batchEnding(1, 0)
```

Fig.57. Envío y recepción de mensajes IMV

h. Dependiendo de los mensajes enviados y recibidos, cada IMV hace una recomendación. Dicha recomendación se envía por medio de TNCS, de acuerdo al valor de la variable TNC-Status que manipulan los IMVs (Ver Fig.58).

```
++[eap] returns ok
+- entering group post-auth {...}
++[exec] returns noop
++? if (control:TNC-Status == "Access")
? Evaluating (control:TNC-Status == "Access") -> TRUE
++? if (control:TNC-Status == "Access") -> TRUE
++- entering if (control:TNC-Status == "Access") {...}
+++[reply] returns noop
++- if (control:TNC-Status == "Access") returns noop
```

Fig. 58. Evaluación de TNC-Status para recomendación TNC

i. Finalmente se llega a la etapa donde se asigna la VLAN de acuerdo a la recomendación obtenida, por ejemplo, cuando es una autenticación exitosa se envía el mensaje EAP de aceptación al cliente (Ver. Fig. 59).

```
[ttls] Got tunneled reply code 2
       EAP-Message = 0x03030004
       Message-Authenticator = 0x00000000000000000000000000000000
       User-Name = "ene"
       Tunnel-Type:0 = VLAN
       Tunnel-Medium-Type:0 = IEEE-802
       Tunnel-Private-Group-Id:0 = "96"
[ttls] Got tunneled Access-Accept
       [...]


++[eap] returns ok
+- entering group post-auth {...}
Sending Access-Accept of id 27 to 10.0.0.1 port 1051
       Message-Authenticator = 0x00000000000000000000000000000000
       User-Name = "ene"
       Tunnel-Type:0 = VLAN
       Tunnel-Medium-Type:0 = IEEE-802
       Tunnel-Private-Group-Id:0 = "96"
[...]
```

Fig.59. Asignación de VLAN y mensaje de aceptación

6.3. Análisis de resultados

- Se cumplió el objetivo principal del proyecto que es aumentar el nivel de seguridad en el proceso de autenticación del usuario, ya que se utilizaron dos niveles de credenciales, contraseña y huella dactilar, para la identificación del usuario.
- Se aumentó el nivel de seguridad en el acceso a la red, realizando la autenticación biométrica en varias capas antes a lo que se hace normalmente (capa 7 – aplicación- del modelo OSI), es decir, dicha autenticación se realizó en la capa 2 –enlace de datos-. Esto permite tener un mayor control de quién accede a la red, ya que si no se autentica al usuario en esta capa 2, no se abren los puertos para otro tipo de tráfico.
- El envío de datos biométricos mediante paquetes EAP permite que se autentique al usuario antes de formar parte de la red; actualmente no existe un protocolo EAP que envíe datos biométricos.
- Se hicieron pruebas de envío del archivo completo con el vector del descriptor matemático de la huella dactilar, que aproximadamente mide 2KB, obteniendo algunos mensajes de error de fragmentación de los paquetes EAP. Este error de fragmentación se reportó a los investigadores alemanes a cargo del proyecto tnc@fhh, y lo corrigieron. Después de la corrección fue posible enviar el archivo completo de datos biométricos.
- Se hicieron pruebas de envío del archivo con el vector del descriptor comprimido, utilizando la herramienta de software libre GZIP, evitando el error de fragmentación y obteniendo una autenticación exitosa, es decir, no se pierden las características de los datos biométricos a pesar de la compresión.
- El monitoreo de los usuarios ya conectados en cada VLAN no se pudo realizar prácticamente, ya que el equipo con el que se contó en el desarrollo del proyecto no tuvo las características técnicas necesarias. Sin embargo, es importante recalcar que el monitoreo es un componente muy importante dentro del esquema NAC.
- Al inicio del desarrollo de este proyecto se trabajó utilizando un Access Point, para que el acceso a la red fuera inalámbrico; sin embargo, el equipo era obsoleto para los propósitos de NAC, ya que no contaba con soporte para la asignación dinámica de VLANs. Se decidió realizar el proyecto sin este componente, ya que serían solamente pequeñas modificaciones para lograr que fuera inalámbrico, puesto que se eligió EAP-TTLS como túnel interno el cual es una buena opción para redes inalámbricas.
- El trabajo realizado nos sugiere que el proyecto NAC es muy amplio y ambicioso, que no es posible abarcar con un solo trabajo de investigación, por lo que se sugiere que se establezca un grupo de investigación en el área de Seguridad de la Información, que va más allá del estudio de la red, ya que existen varias ramas muy interesantes y fértiles.
- Algunos objetivos a futuro de este tema de tesis serían:
    - probar el concepto de monitoreo central de todas las VLAN
    - desarrollar la parte de MAP para utilizar el monitoreo y la correlación de datos de los sensores.
    - realizar las modificaciones necesarias para lograr el acceso inalámbrico utilizando el Access Point adecuado.

7. Conclusiones

Es posible desarrollar mediante herramientas del dominio público un sistema de Network Access Control NAC (Control de acceso a red) que incluye políticas de seguridad de pre admisión y control después de la admisión, verificando en cada momento los usuarios e invitados que ingresan y trabajan sobre la red.

El trabajo permite ingresar datos biométricos dentro del protocolo de autenticación, EAP-TTLS/EAP-TNC. Esto permite verificar la identidad del usuario con dos mecanismos de autenticación antes de establecerse la conexión exitosa.

En el trabajo desarrollado se autenticó tanto al usuario como a la máquina con la que se accede a la red, verificando su estado de seguridad y el cumplimiento de las políticas. Si no se cumplían con dichas características se mandaba a una VLAN diferente para el manejo de la remediación.

En el proyecto se manejó un acceso cableado al switch, sin embargo el manejo del control de acceso a una red inalámbrica también está incluido en el concepto NAC y la arquitectura TNC, teniendo solamente la diferencia la negociación inicial con el punto de acceso y el mecanismo de autenticación elegido, que en este caso fue EAP-TTLS.

Es indispensable realizar pruebas de concepto e implementaciones de los estándares emergentes para demostrar que es mejor seguir una arquitectura libre que limitarse a una solución privada, permitiendo así la interoperabilidad de dispositivos y vendedores.

Para tener cómputo confiable es necesario trabajar coordinadamente con la seguridad en clientes, servidores, almacenamiento y redes. No es suficiente tener un buen control de acceso, pero es el primer paso para tener un nivel de seguridad aceptable.

Glosario

AAA Authentication, Authorization & Accounting

Access point Punto de acceso a una red inalámbrica que se encarga de recibir información de diferentes estaciones mediante un sistema de radio frecuencia (RF) para conectarlas a una red de estructura principal con cables.

AP Veáse Access Point

AR Access Requestor

Baseline Línea que marca la base o punto de referencia para el seguimiento de políticas.

Booting Proceso que inicia un sistema operativo cuando es encendida la computadora.

DIAMETER Protocolo para de autenticación (AAA) sucesor a RADIUS, definido en el RFC 3588.

DoS (Denial of Service). Negación de servicio a usuarios legítimos, cuando se pierde la disponibilidad de un sistema o servicio.

EAP (Extensible Authentication Protocol). Extensión del Protocolo punto a punto (PPP). Proporciona un mecanismo estándar para aceptar métodos de autenticación adicionales junto con PPP. Al utilizar EAP, se pueden agregar varios esquemas de autenticación, entre los que se incluyen tarjetas de identificación, contraseñas de un sólo uso, autenticación por clave pública mediante tarjetas inteligentes, certificados y otros.

Endpoint Dispositivo que es el punto final, donde se tiene al AR; la conexión física se realiza por medio de su tarjeta de red.

Handshake Protocolo de control para establecer comunicación.

Hash También conocido como "message digest", es un número generado a partir de una cadena de texto, utilizando una fórmula de tal forma que sea poco probable que algún otro texto produzca el mismo valor. En la seguridad se emplean para asegurar que los mensajes transmitidos no han sido manipulados. El emisor genera un hash del mensaje, lo cifra y lo

envía con el propio mensaje. El receptor luego decodifica ambos, produce otro hash del mensaje recibido y compara los dos hashes, si coinciden, existe una probabilidad muy elevada de que el mensaje recibido no haya sufrido cambios desde su origen.

IDS Intrusion Detection System, sistema de detección de intrusos.

IETF Internet Engineering Task Force, grupo dedicado a crear nuevas especificaciones estándares para Internet.

IMC Integrity Measurement Collector

IMV Integrity Measurement Verifier

MAP Metadata Access Point

MD5 Algoritmo de función hash, creado por Ron Rivest del MIT Laboratory for Computer Science and RSA Security, y es utilizado para crear firmas digitales. Emplea funciones hash unidireccionales, es decir, que toma un mensaje y lo convierte en una cadena fija de dígitos.

MINDTCT Método detector de minucias de una imagen de huella dactilar (MINutia DeTeCTion).

NEA Network Endpoint Assessment

NAA Network Access Authority

NAC Network Access Control, es un enfoque de la seguridad en redes que intenta unificar la seguridad del dispositivo, la autenticación del usuario y la seguridad de la red.

NAR Network Access Requestor

NIST **(**National Institute of Standards and Technology), Instituto Nacional de Estándares y tecnología, es una agencia federal de los Estados Unidos de América. La misión de este instituto es promover la innovación y la competencia industrial mediante avances en normas y tecnología.

OSI El modelo Open System Interconnection, es un modelo de referencia desarrollado por ISO, que consiste en siete capas las cuales especifican funciones particulares de la red.

Payload Área de datos de un paquete de red.

PDP Policy Decision Point

PEP Policy Enforcement Point

PPP Point-to-Point Protocol, es un protocolo que provee conexiones router-to-router y host-to-network sobre circuitos síncronos y asíncronos.

QoS Quality of Service, calidad de servicio.

RADIUS Remote Authentication Dial-In User Server, es un protocolo de autenticación y autorización para aplicaciones de acceso a la red.

Roaming En redes inalámbricas se refiere a la capacidad de moverse desde un área cubierta por un Punto de Acceso a otra sin interrumpir el servicio o pérdida de conectividad

Router Dispositivo que opera en la capa 3 (nivel de red)del modelo OSI que puede decidir por cual ruta en la red el tráfico permanece con una métrica óptima. Envía los paquetes de una red a otra, basándose en la información de la capa de red.

SO Sistema Operativo, es un programa informático que actúa de interfaz entre los dispositivos de hardware y el usuario.

Spoofing Técnica basada en la creación de tramas TCP/IP utilizando una dirección IP falsa; desde su equipo, un atacante simula la identidad de otra máquina de la red (que previamente ha obtenido por diversos métodos) para conseguir acceso a recursos de un tercer sistema que ha establecido algún tipo de confianza basada en el nombre o la dirección IP del host suplantado.

Switch Es un dispositivo digital de lógica de interconexión de redes que opera en la capa 2 (nivel de enlace de datos) del modelo OSI. Su función es interconectar dos o más segmentos de red, pasando datos de un segmento a otro de acuerdo con la dirección MAC de destino de las tramas en la red.

Tampering Falsificación o alteración de algún recurso o sistema.

TCG Trusted Computing Group

TLS Transport Layer Security, es un protocolo independiente que permite que protocolos de niveles superiores se sitúen por encima de él de manera transparente. Basado en SSL de Netscape 3.0.

TNC Trusted Network Connect

TNCC Trusted Network Connect Client

TNCS Trusted Network Connect Server

Token En sistemas de seguridad, un pequeño dispositivo del tamaño de una tarjeta de crédito que muestra un código ID que cambia constantemente (cada x minutos). El usuario primero introduce una clave y luego la tarjeta muestra un ID que puede ser utilizado para acceder a la red.

VLAN (Virtual Local Area Network), red de área local virtual, es un método para crear redes lógicamente independientes dentro de una misma red física, que ayudan a facilitar la administración de la red.

VPN (Virtual Private Network), red privada que se configura dentro de una red pública. Para establecer este tipo de red, la integridad de los datos y la confidencialidad se protegen mediante la autentificación y el cifrado.

Vulnerabilidad En seguridad informática, la palabra vulnerabilidad hace referencia a una debilidad en un sistema permitiendo a un atacante violar la confidencialidad, integridad, disponibilidad, control de acceso y consistencia del sistema o de sus datos y aplicaciones.

XML Extensible Markup Language, es un metalenguaje extensible de etiquetas que permite definir la gramática de lenguajes específicos. Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.

Anexos

A. Puertos y servicios

Puerto	Estado	Puerto	Estado
TCP 20	FTP-data	TCP 80	HTTP
TCP 21	FTP-data	TCP 443	HTTPS/SSL
TCP 22	SSH	TCP 8080	Web
TCP 23	Telnet	TCP 5223	hpvirtgrp
TCP 25	SNMP	UDP 53	Dns-lookup
TCP 587	submission	UDP 67	bootps
TCP 110	Pop3	UDP 68	bootps
TCP 995	Pop3s	UDP 4444	krb524

(18). **McClure, S; Scambray, J; Kurtz, G.** Hackers. Secretos y soluciones

B. Archivo de configuración de wpa_supplicant

El archivo de configuración que utiliza el cliente, para utilizar wpa_supplicant es el siguiente:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
ap_scan=0
network={
      ssid="SwitchN"
      key_mgmt=IEEE8021X
      eap=TTLS
      identity="XXXXX"
      password="YYYYY"
      ca_cert="/dir/ca.pem"
      id_str=""
      eapol_flags=0
}
```

Índice de ilustraciones

Índice de Tablas

Referencias

1. **McClure, S, Scambray, J y Kurtz, G.** *Hackers. Secretos y soluciones para la seguridad de redes.* Madrid : McGraw-Hill, 2002.

2. **Trusted Computing Group.** Trusted Computing Group. [En línea] 2008. http://www.trustedcomputinggroup.org/.

3. *Control de acceso biométrico a red inalámbrica (WNAC) utilizando software libre.* **Olmos Roa, Cinthya Enetzy.** Morelos, México : CIINDET, 2009. VII Congreso Internacional en Innovación y Desarrollo Tecnológico.

4. **Forouzan, B.** *Transmisión de datos y redes de comunicaciones.* Madrid : McGraw-Hill, 2001. p. 442-445.

5. **García Navarro, J.F.** *Autenticación de usuarios en redes inalámbricas mediante el uso de un protocolo robusto para el manejo de contraseñas débiles.* Maestría en Ciencia e Ingeniería de la Computación. México : UNAM, 2005. Tesis.

6. *Attacks on Biometric Systems: A Case Study in Fingerprints.* **Uludag, Umut y Jain, Anil.** Department of Computer Science and Engineering : s.n., 2004, Security, Steganography and Watermarking of Multimedia Contents VI, Vol. 5306, págs. 622-633.

7. **Tucci, Linda.** Network access control: A hybrid approach. *SearchCIO.com.* [En línea] 27 de 01 de 2009. [Citado el: 08 de 02 de 2009.] http://searchcio.techtarget.com/news/article/0,289142,sid182_gci1346107,00.html .

8. **Orans, Lawrence y Nicolett, Mark.** *Gartner's Network Access Control Model.* s.l. : Gartner, 2008.

9. **Clancy, Heather.** Tech Watch: Interest in NAC rising. *SearchITChannel.com.* [En línea] 01 de 04 de 2008. [Citado el: 22 de 01 de 2009.] http://searchitchannel.techtarget.com/news/article/0,289142,sid96_gci1307889,00.html 22 de enero de 2009.

10. **Tucci, Linda.** 2009: The year of Network Access Control. [En línea] 03 de 02 de 2009. [Citado el: 08 de 02 de 2009.] http://searchsecurity.techtarget.com.au/articles/29045-2-9-The-year-of-Network-Acces-Control-.

11. **Shavit, Yuval.** Implementing NAC products. *SearchSecurityChannel.com.* [En línea] 17 de 01 de 2008. [Citado el: 22 de 01 de 2009.] http://searchsecuritychannel.techtarget.com/generic/0,295582,sid97_gci1294505,00.html .

12. TNC Architecture for Interoperability. Specification Version 1.3, Revision 6. . [En línea] 2008 de Abril de 28. http://www.trustedcomputinggroup.org/files/resource_files/8CB439DF-1D09-3519-ADC5A15A7A9DE2D9/TNC_Architecture_v1_3_r6.pdf. Pág. 12.

13. **Sangsters, P.** *Network Endpoint Assessment (NEA): Overview and requirements* . Network Working Group : IETF, 2008.

14. **Cisco.** Network Admission Control (NAC) Framework. [En línea] Cisco, 2008. http://www.cisco.com/en/US/netsol/ns617/networking_solutions_sub_solution_home.html.

15. **Microsoft TechNet.** Network Access Protection Design Guide. [En línea] Microsoft, 2008. http://technet.microsoft.com/es-mx/library/dd125338(WS.10).aspx.

16. **Langston, Richard.** Network Access Control Technologies and Symantec Compliance on Contract. s.l. : Symantec, 2006.

17. **Symantec.** Symantec NAC. *diarioti.com.* 20 de agosto de 2008.

18. **FreeNAC Core Team.** FreeNAC. [En línea] 2006. http://www.freenac.net/es.

19. **Hannover, University.** Project FHH. *tnc@fhh.* [En línea] 2009. http://trust.inform.fh-hannover.de/joomla/index.php/projects/tncfhh.

20. **Tucci, Linda.** Network Access Control: Pointers for getting the knack of NAC. *Security for the Midmarket.* [En línea] 02 de 02 de 2009. [Citado el: 08 de 02 de 2009.] http://searchcio-midmarket.techtarget.com/tip/0,289483,sid183_gci1346628_mem1,00.html.

21. **Rittinghouse, John W. y Ransome, James F.** *Wireless Operational Security.* Oxford : Elsevier Digital Press, 2004.

22. **Riley, Steve.** Mitigating the Threats of Rogue Machines-802.1X or IPSec? 2005.

23. **Phifer, Lisa.** How to compartmentalize WLAN traffic using an existing VLAN. [En línea] 12 de 03 de 2006. [Citado el: 15 de 12 de 2008.] http://searchsecuritychannel.techtarget.com/tip/0,289483,sid97_gci1231873,00.html.

24. NAC: Managing unauthorized computers. s.l. : Sophos, Abril de 2007.