Desarrollo de un conjunto de aplicaciones para detección de ataques en redes IPv4/IPv6 utilizando Python

Erick Melgar, Ing. José Patiño
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
emelgar@espol.edu.ec, jpatino@espol.edu.ec

Resumen

El complejo mundo de la seguridad informática presenta novedades día a día y requiere que el auditor de red se encuentre en constante crecimiento profesional. Los auditores de seguridad suelen disponer de un kit de herramientas o aplicaciones para ejercer sus proyectos. El siguiente trabajo comprende en realizar una herramienta para dichos auditores de red los cuales necesitan detectar los múltiples ataques producidos por personas ajenas que desean perjudicar a los usuarios obteniendo su información o manipularla para otros fines. El usuario final al utilizar dicha herramienta podrá lograr detectar ataques comunes que se encuentran en las redes IPv4/IPv6 como por ejemplo "Man In The Middle" o la Denegación de Servicios.

Palabras Claves: *Ataques en redes de datos, Man In The Middle, DoS, DNS, Python, IPv4, IPv6.*

Abstract

The complex world of informatic security presents news every day and requires the network auditor is in constant professional growth. Auditors securiy usually have a toolkit or applications to perform their projects. The following work involves in making a tool for network said auditors who need to detect multiple attacks produced by outsiders who wish to harm users getting their information or manipulate it for other purposes. The end user when using this tool may be able to detect common attacks that are in the IPv4 / IPv6 networks such as "Man In The Middle" or Denial of Service.

Keywords*: Attacks in data networks, Man In The Middle, DoS, DNS, Python, IPv4, IPv6.*

1. Introducción

Las amenazas contra la seguridad basadas en la red han provocado robos de identidad y fraude financiero generalizados. El correo no deseado, los virus y el spyware causan graves problemas a empresas y consumidores. [2]

Este proyecto apunta a obtener la experiencia técnica para defender las redes de las instituciones a las que nos encontremos evitando así problemas de seguridad de las mismas a futuro. Además se utilizara el lenguaje de programación llamado "Python" lo cual es uno de los más utilizados para el desarrollo de las aplicaciones contra seguridades de red. [6]

Además se implementara un conjunto de aplicaciones que permitirá la detección de diversos ataques que se realizan a las redes IPv4 e IPv6 para poder evitar la infiltración de usuarios ajenos a nuestra red que tenga de propósito perjudicar la información que se desea enviar a través de la misma.

2. Fundamento teórico

El documento se enfoca en cinco principales ataques que se encuentran regularmente en la redes IPv4 e IPv6 además del lenguaje de programación Python. Para esto se necesita conocer las versiones de protocolos de internet en las que se las encuentran.

2.1 IPv4

El objetivo del protocolo de internet es el de realizar la entrega de paquetes, aunque no todos los paquetes llegan en el mismo orden o ni siquiera llegan; y el de la fragmentación y armado de estos paquetes; y llevarlos de un nodo en una red hacia su nodo destino, para lograr esto, a cada paquete se lo identifica con una dirección IP. [3]

Como lo habíamos mencionado, cada paquete que viaja a través de la red, dentro de si hay un lugar en donde se encuentra la dirección IP de origen y la dirección IP de destino. Y cabe recalcar que la dirección IPv4 no es más que una dirección lógica que

posee 32 bits divididos en 4 grupos de 8 bits llamados octetos, con esa cantidad de bits, podemos sacar la dirección IP que identifica a la red, así como la dirección del host; cada grupo de 8 bits se encuentra separado por un punto; y al ser de 8 bits, significa que en formato decimal, cada casilla puede tomar el valor entre 0 y 255, es decir hay una posibilidad de 256 combinaciones. [8]

Figura 1. Ejemplo de una dirección IPv4

2.1.1. Problemas en IPv4

Aunque IPv4 ha sido muy utilizada a nivel mundial, presenta ciertos problemas, ciertas desventajas que hace que poco a poco pensemos definitivamente en cambiarnos a IPv6. Uno de los problemas que presenta las direcciones IP en su cuarta versión son: Al tener una combinación de 4 octetos, 32 bits en total, nos da un número limitado de direcciones lógicas, que cuando llegue a ese límite, como está sucediendo ahora; no se podrá asignarles direcciones IP a más equipos. [12]

2.2. IPv6

Cuando hablamos de protocolo IPv6, ya no debemos preguntarnos si sería una opción acertada hacerlo; la pregunta que nos debemos hacer es, cuando nos vamos a cambiar y como lo haremos.

Debido al inevitable fin del protocolo IPv4, y aunque de muchas maneras se ha tratado de continuar con su uso, el protocolo IPv6 cada vez comienza a tomar mayor protagonismo dentro del mundo del "networking"; aunque no es algo nuevo, y se lo viene incorporando de a poco, es algo a lo que deberíamos ponerle más énfasis, comprender cada vez más este protocolo; y disfrutar de todas las bondades que nos brinda. [7]

Una de sus principales bondades y características es que pone fin al problema de direccionamiento que sufría IPv4, ya que posee un esquema de direcciones mucho mayor. Al mismo un espacio mayor para las direcciones IP, permitirá a los servidores de internet una mejor organización y distribución. Cabe recalcar que una red basada en IPv6, trabaja exclusivamente con ICMPv6.

2.2.1. Estructura de la dirección IPv6

La estructura del protocolo IPv6 está formada por el encabezado, extensiones y el direccionamiento.

El encabezado se encuentra formado por ocho campos, lo que da lugar a un total de 40 octetos. Entre los campos tenemos a:

- La versión, está formada por 4 bits, aquí detalla la versión del paquete, sabemos que viene el número 6.
- Clase de tráfico, este ocupa 8 bits, aquí se etiqueta al paquete con un Punto de Código de Servicios Diferenciados (DSCP), en el que especifica cómo debe ser manejado.
- Etiqueta de Flujo, son 20 bits, marca la secuencia de los paquetes a través de la red, además de diferenciar a los paquetes que necesiten un tratamiento a lo largo de la trayectoria.
- Dirección Fuente, 128 bits, especifica la dirección IPv6 de la cual sale el paquete.
- Dirección Destino, 128 bits, especifica la dirección IPv6 a la cual debe llegar el paquete.

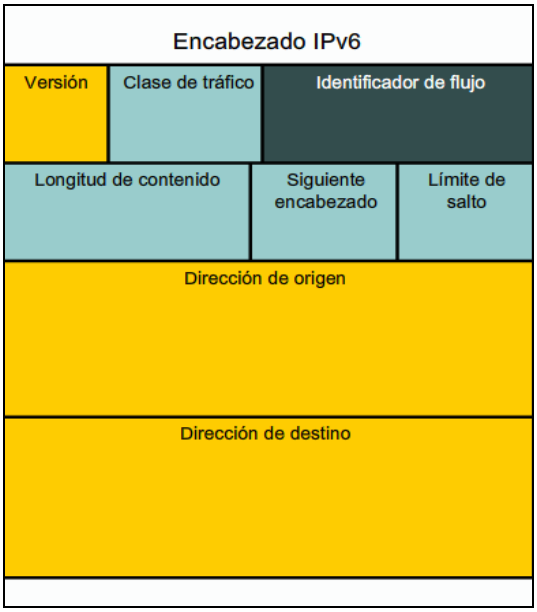

Figura 2. Encabezado IPv6

En lo que respecta al direccionamiento, este consiste en 8 sets, de 16 bits hexadecimales, cada set está separado por ":", y todos juntos forman los 128 bits, que es la longitud de una dirección de protocolo IPv6. [5]

2.3. Python

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, es un lenguaje de programación muy poderoso que nos permite realizar cualquier tipo de programa, desde aplicaciones web, o aplicaciones en Windows hasta servidores de red. Se desarrolla como un proyecto de código abierto. Es un lenguaje de programación interactivo, e interpretado, esto significa que no necesita compilar el código para poder ejecutarlo, lo que lo hace con rapidez de desarrollo y con inconvenientes a menor velocidad.

Python aunque es muy fácil de usar, es un lenguaje de programación muy poderoso, nos brinda estructuras de datos muy eficientes, de alto nivel y siempre sin perder de vista la meta que es de hacerlo de una manera sencilla. Su fácil sintaxis, su modo de interpretar los comandos y su extensa biblioteca hacen de este un lenguaje perfecto para el desarrollo de aplicaciones en diversos campos, y cabe indicar que también en diferentes plataformas. [14]

3. Ataques en redes de datos IPv4 e IPv6

3.1. Man In The Middle en IPv6 con Neighbor Advertisement Spoofing

Este ataque tiene como objetivo leer, insertar y modificar a voluntad la información que se intercambia entre dos equipos sin que éstos no se den cuenta de que el enlace entre ellos ha sido violado.

Este ataque es usado en ambos protocolos IPv4/IPv6 usando diferentes técnicas. letra de nombres, pronombres, verbos, adjetivos y adverbios; en minúsculas artículos, conjunciones o preposiciones (a menos que el título empieza con esa palabra).

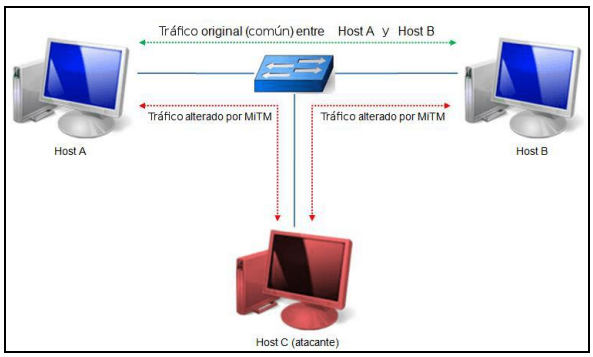

Figura 3. Ataque Man In The Middle

En este caso el ataque será para IPv6 usando Neighbor Advertisement Spoofing el cual es similar al ataque ARP Spoofing que existe para IPv4 usando el protocolo ARP, pero en IPv6 ese protocolo ya no se utiliza lo cual ahora se utiliza ICMPv6 que cumple funciones como las de ARP pero utilizando Neighbor Discovery Protocol(NDP).

El NDP consiste en un mecanismo con el cual un nodo que se incorpora a una red descubre la presencia de otros nodos en el mismo enlace y también puede ver sus direcciones IP. El funcionamiento normal de la comunicación en IPv6 debe ser que el nodo envía un mensaje Neighbor Solicitation(Solicitud de Vecino) como multicast para localizar su destino y luego de encontrarlo el destino envía un mensaje Neighbor Advertisement como unicast al origen.

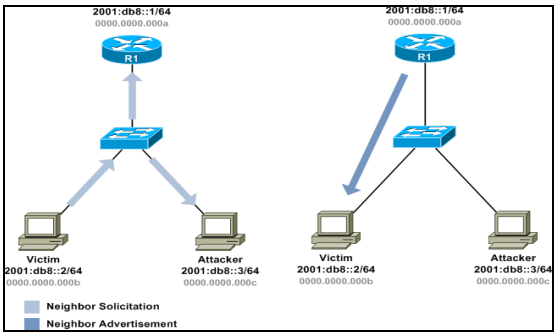

Figura 4. Envío de paquetes para detección de nodos.

El ataque se realiza spoofeando la dirección IPv6 de origen del paquete, para simular ser un mensaje que viene del otro equipo víctima, pero en ambos casos se pone la dirección MAC del atacante, para conseguir que el conmutador de comunicaciones haga llegar todos los mensajes a la máquina del hombre en medio. [1]

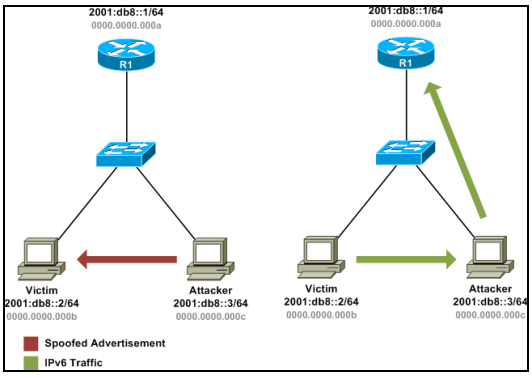

Figura 5. Ataque Neighbor Advertisement Spoofing

3.2. Man In The Middle en IPv4 con ARP Spoofing

Una de la forma más antigua de lograr un ataque Man-in-the-middle es utilizar ARP Spoofing. Con ARP Spoofing el objetivo del atacante es la capa dos del protocolo de direcciones MAC.

En una red de área local, un host se comunica con otro host, incluyendo la puerta de enlace (gateway), mediante la entrega de paquetes a una dirección MAC. Primero el protocolo ARP tiene que resolver la dirección MAC de la dirección IP de un host. No existe un procedimiento de comprobación o de autenticación del protocolo ARP.

Cuando un equipo necesita enviar información a otro host en la red el ordenador genera un ARP broadcast. El ARP broadcast es enviado a cada ordenador de la red solicitando que una dirección IP específica responda con la dirección MAC correspondiente. Cuando un equipo responde con una dirección MAC, los datos pueden ser entregados a esa dirección MAC, el problema es que la respuesta se acepta sin verificación.

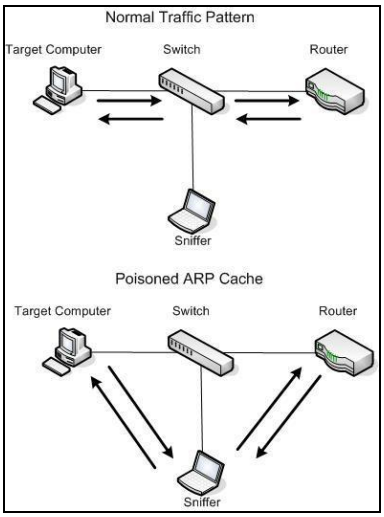

Figura 6. Ataque con ARP Spoofing.

ARP Spoofing implica el envío de información fraudulenta a los hosts atacados por lo que incorrectamente mapa la dirección MAC del atacante como pertenecientes a la dirección IP. [1]

3.3. Man In The Middle en IPv6 con SLAAC

En IPv6 existe otra forma de realizar Man In The Middle y es utilizando el SLAAC (Stateless Address Autoconfiguration) o Autoconfiguración de direcciones libres de estado.

Para realizar este ataque, el atacante debe introducir un enrutador (o algún dispositivo que actúe como tal, puede ser su propio ordenador) en la red interna con dos interfaces (virtuales o no): una de cara a la red interna, que soporte solamente IPv6 y otra con la conexión a Internet (solamente IPv4). En esos momentos existirá una red adicional IPv6, pero el atacante no controlará el tráfico.

El intruso comenzará a enviar RA (Router Advertisements, anuncios de rutas), que es una especie de DHCP para IPv6. El objetivo es que el tráfico pase a través de la interfaz IPv6 sin que los clientes noten nada y esto se consigue gracias a una especificación obsoleta.

El método es definir en el enrutador un prefijo IPv6 e incrustar en los últimos 32 bits una dirección IP versión 4, que según el ataque previsto, debe coincidir con un servidor DNS del propio atacante, situado en la interfaz IPv4 del enrutador (en Internet). Si se configura adecuadamente ese enrutador del atacante para que se encargue de traducir (a través de NAT-PT) las direcciones IPv6 de las víctimas a IPv4, se consuma el ataque, engañando al usuario para que crea que su servidor DNS es el del atacante.

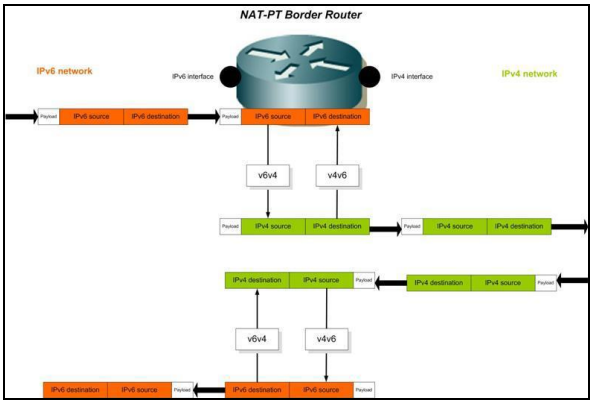

Figura 7. Ataque con SLAAC

El siguiente paso es hacer que los sistemas operativos usen la red IPv6 (y sus DNS) creada paralelamente y que lo hagan instantáneamente (si no responde a tiempo, se usaría el DNS legítimo). Esto se consigue de forma muy sencilla por dos razones: La primera es el uso de Application Layer Gateways (ALGs), que es necesario en NAT-PT para hacer NAT en protocolos "especiales" como FTP. La segunda es que los sistemas operativos modernos prefieren siempre utilizar IPv6 (se han diseñado así para, facilitar la migración).

En resumen, la víctima utiliza sin darse cuenta el DNS del atacante para resolver direcciones y, por tanto, puede ser redirigido a cualquier página (que no use certificados) de forma transparente. [4]

3.4. DoS en IPv4 con Invalid MAC Spoofing

Uno de los ataques más famosos, del que siempre estamos escuchando en los medios de comunicación es el ataque de denegación de servicio (DoS).

El ataque lo realizamos al servidor, saturándolo sus puertos con flujos de información y peticiones, llega un momento en que el servidor no puede procesar todas las peticiones, lo que hace que colapse, haciendo que no pueda proveer el servicio; a este fenómeno se lo denomina denegación.

El Invalid MAC Spoofing es una técnica que consiste en cambiar la dirección MAC en un dispositivo de red, por otro que no es el que le corresponde. Para que todas las peticiones que le deberían llegar al servidor, se pierdan en la red.

Haciendo que tanto el servidor no sea capaz de proporcionar su servicio, ya que las peticiones correctas no llegan a este, así como que los paquetes que logren llegar podrían no ser legítimos, sino paquetes alterados. [1]

3.5. DNS Hijacking

Para poder acceder a diferentes páginas en internet, acostumbramos a poner el nombre del dominio en la barra de direcciones de nuestro navegador preferido.

Pues este nombre de dominio es traducido a una dirección IP publica, que hace referencia al sitio web, que deseamos acceder, esta traducción de nombre de dominio a IP lo hace nuestro servidor DNS (Domain Name System).

Muchas veces este servidor se encuentra dentro de nuestra propia red Lan y es un blanco utilizado por los hackers, con las intenciones de redirigir o "secuestrar" las direcciones DNS, a los servidores DNS falsos; con el fin de inyectar malware en el PC; prueba de ello son las promociones de estafas de phishing, la publicidad en los sitios web de alto tráfico, y cualquier otra forma relacionada de la actividad criminal.

Una vez que el atacante logra secuestrar nuestro servidor, se transforma en un DNS falso, ya que en vez de traducir el dominio en una dirección legítima, este nos envía direcciones IP falsas, re direccionándonos a sitios web maliciosos. DNS Hijacking puede ocurrir con cualquier sitio web grande o pequeño y convertir esos sitios web en los sitios web maliciosos sin el conocimiento del internauta. [2]

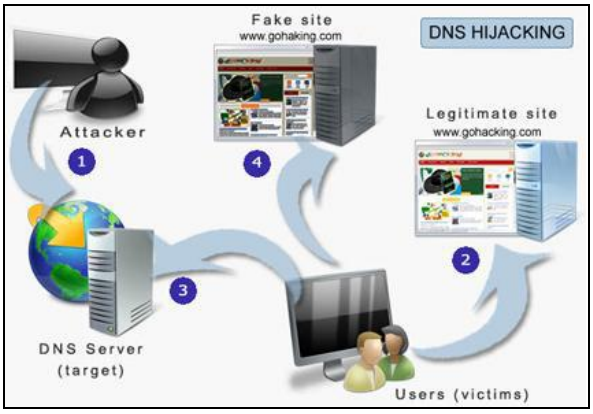

Figura 8. Ataque DNS Hijacking

Dado no solo los usuarios sino que los propietarios de sitios web, dependen de servidor DNS legítimo, que se emitió por sus proveedores de servicios de Internet (ISP), secuestradores DNS utilizan el malware en forma de un troyano para intercambiar la asignación del servidor DNS legítimos por el ISP con una asignación de servidor DNS manual desde un falso servidor DNS.

Cuando los internautas visitan los sitios web de confianza con los nombres de dominio legítimos, son secuestrados automáticamente a un sitio web malicioso que se disfraza como el legítimo. El enrutador del servidor DNS legítimo al servidor DNS falso pasa desapercibido tanto por el internauta y el propietario legítimo sitio web. Esto abre el sitio web malicioso para realizar cualquier acto criminal que el hacker desea porque el usuario piensa que son en el sitio web real.

4. Detección de Ataques

4.1. Detección de Neighbor Advertisement Spoofing

Para poder realizar la detección se debe saber cómo funciona el ataque, la victima envía un mensaje Neighbor Solicitation (NS) a una dirección multicast para buscar los nodos que se encuentran en la red y encontrar el dispositivo con el que se desea comunicar, una vez que recibe el mensaje el nodo receptor este envía un mensaje unicast de Neighbor Advertisement (NA) y almacenara en la tabla de vecinos la dirección MAC asociada con la dirección IPv6.

El atacante aprovecha esos envíos de mensajes para interceptarlos y enviar mensajes Neighbor Advertisement a ambos equipos poniendo la dirección IPv6 del otro y la dirección del atacante.
Por lo que para realizar la detección nosotros realizamos captura de paquetes con un “sniffer” de la

librería scapy, pero la efectividad se encuentra en que debemos realizar filtros para obtener los paquetes.

El primer filtro es obtener todos los paquetes IPv6, luego hay que capturar todos los nodos IPv6 que incluyen enrutadores y Neighbor Advertisement con el siguiente código que nos permite realizarlo:

Filter= "ip6 and not tcp and not udp"

Luego obtenemos los paquetes que son de Neighbor Advertisement ICMPv6 NA los cuales corresponden a la detección de este ataque, vemos si dicho paquete no tiene como destino la dirección multicast ff02::1 entonces estamos siendo atacado. Obtendremos la siguiente información:

- Atacante: pkt[IPv6].src
- Dirección MAC del Atacante: pkt[Ether].src
- Objetivo: pkt[ICMPv6ND_NA].tgt

Los nombres y afiliación del autor(es) deben estar centrados abajo del título y se imprimirán en Times tamaño 10, sin negrilla, dejando una línea de espacio a tamaño 10 después del título, tal como se indica arriba.

4.2. Detección de ARP Spoofing

Este ataque es muy común en las redes IPv4 lo cual realiza un envenenamiento a la tabla ARP simulando a las víctimas que tienen una comunicación directa entre ellos sin saber que existe un intermediario que es el atacante.

El método de detección es realizar una captura de paquetes IPv4 realizando un filtro a todo el tráfico "ARP", ahí obtenemos los datos del enrutador por lo cual si nuestra IP se encuentra en el registrado entonces lo almacenamos en una variable igual que la MAC.

Si en un futuro recibimos una IP del enrutador que no estaba almacenada en la variable anterior y a la vez no se encuentra la MAC registrada anteriormente almacenada es porque estamos siendo atacados. Por lo cual mostraremos una alerta para decir al usuario que está siendo atacado.

4.3. Detección de Man In The Middle con SLAAC

La metodología para la detección es realizar la captura de paquetes con un "sniffer" de la librería scapy, pero la efectividad se encuentra en que debemos realizar filtros para obtener los paquetes.

El primer filtro es obtener todos los paquetes IPv6, luego hay que capturar todos los nodos IPv6 que incluyen enrutadores y Neighbor Advertisement con el siguiente código que nos permite realizarlo:

Filter= "ip6 and (dst host ff02::1)"

Usamos la dirección multicast ff02::1 que representa todos los nodos que se encuentran en el segmento de red, también se lo puede filtrar por la dirección multicast ff02::1:2 que es para los servidores DHCP y los agentes de retransmisión de la red. De ahí almacenamos en una variable los mensajes RA(Router Advertisement) que son del enrutador que inicialmente las envio, luego con el "sniffer" realizado en el paso anterior filtramos los nuevos mensajes RA para compararlos y si no pertenecen al enrutador original entonces advertiremos al usuario que está siendo atacado indicando la mac del atacante con el comando:

pkt[ICMPv6NDOptSrcLLAddr].lladdr

4.4. Detección de DoS con Invalid MAC Spoofing

Este ataque es similar al ataque "Man In The Middle" con ARP Spoofing solo que en este caso lo que realiza es una denegación de servicio por lo que se detecta con el método para ARP Spoofing pero además se necesita saber si hay respuesta con el otro nodo por lo que se realiza un tiempo de espera y si no hay respuesta entonces será por negación de servicio.

4.5. Detección de DNS Hijacking

Para la detección de este ataque se necesita primero configurar dos archivos que son parte del aplicativo, los cuales son "dns_ip.conf" donde se almacenaran las direcciones ip que se desean monitorizar, y el otro archivo llamado "nombres_dns.conf" donde irán el nombre de los DNS para poder notificar. Estos son almacenados para poder verificar que no exista cambio de ip's al momento de interactuar y saber si el sitio ha sido falsificado.

5. Desarrollo Interfaz Gráfica

Para el desarrollo del conjunto de aplicaciones para la detección de ataques en redes IPv4 e IPv6, nos hemos propuesto hacerlo de una manera sencilla y de fácil manejo para el usuario.

La arquitectura en la cual se basa este conjunto de aplicaciones es el siguiente:

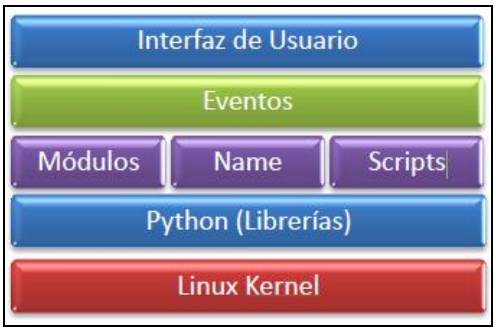

Figura 9. Arquitectura de la aplicación

Como ya lo mencionamos, la aplicación fue desarrollada en Ubuntu, por lo que esta va a correr sobre el kernel de Linux, es aquí donde Python hará su trabajo, ejecutaremos librerías muy conocidas, que darán al usuario una interfaz gráfica muy amigable y fácil de manejar

Para su interfaz gráfica hemos tratada de resumirla en una sola ventana de presentación inicial. Esta será nuestra carta de presentación. Como veremos en la ventana, tendremos los iconos correspondientes a cada ataque que la aplicación está preparada para detectar.

Figura 10. Pantalla Inicial del aplicativo

Es decir, dentro de esta ventana, si necesitamos comenzar a detectar un ataque específico, lo que tendremos que hacer, simplemente será de dar un clic con nuestro cursor, sobre el ícono del ataque que deseamos detectar. Esto hará, que el programa ejecute un nuevo script, haciendo que aparezca una nueva ventana indicando que está preparado y atento para la detección de un ataque; las ventanas para los diferentes ataques, son muy parecidas y gráficamente funcionan de la misma manera. Esto hace que cumplamos con uno de nuestros objetivos en la realización del software, que era que nuestra aplicación sea muy sencilla para el usuario.

Además de los íconos referentes a cada ataque, es importante señalar, que para detectar un posible ataque, la suite de aplicaciones debe estar corriendo, es decir, el script de la ventana principal debe estar activo, y a su vez, seleccionar el ataque que deseamos capturar. Si cerramos estas ventanas, el programa no se ejecutará en segundo plano.

Las pantallas para cada ataque son similares donde se podrá observar el botón "Start" con el cual se podrá iniciar la detección de los ataques. Cuando la imagen de la parte inferior derecha se encuentra de color verde es debido a que no se ha detectado ninguna amenaza.

Figura 11. Estado Normal

Cuando dicho icono está de color rojo significa que fue detectado alguna anomalía por lo cual aparecerá la información respectiva en el cuadro de la parte inferior izquierda.

Figura 12. Estado Advertencia

6. Implementación y Resultados

6.1. Herramientas utilizadas para la implementación

Python 2.7.9: Se utiliza esta versión de Python debido a que es más estable que la versión 3 y es compatible con las librerías a utilizar.

Scapy: Otra de las herramientas que necesitaremos es Scapy. ¿Qué es y para qué nos sirve? Bueno pues Scapy es un script desarrollado en Python. Y nos sirve para poder generar paquetes, es inclusive capaz de realizar ataques. Trabaja con los principales protocolos de comunicación, si deseamos ver con detalles cuales son los protocolos con los que trabaja, solo debemos poner el comando ls().

Tkinter: otra herramienta de la cual vamos a hacer mucho uso, no es más que el paquete GUI (Graphic User Interface) estándar de Python. No es la única librería de interfaz gráfica pero sin embargo es una de las más populares.

Evil Foca: herramienta diseñada por la empresa "Eleven Paths" que permite realizar ataques en redes IPv4 e IPv6 para lo cual se lo utiliza para poder realizar las pruebas de funcionalidad correspondientes. [13]

6.2. Análisis estadístico de resultados

Para poder obtener una aprobación de nuestro proyecto se necesita realizar un estudio estadístico mediante los resultados que se dieron en las pruebas para lo cual necesitaremos empezar calculando el número mínimo de observaciones.
La ecuación (4.1) nos permite calcular el número mínimo de observaciones que deben efectuarse; donde n es el número mínimo de observaciones, W el rendimiento mínimo esperado, Z_β poder estadístico y Z_α nivel de confianza asignado.

$$n = \frac{W - W^2\,[Z_\beta + 1.4\,(Z_\alpha)]^2}{W^2} \qquad (4.1)$$

Figura 13. Ecuación para el número de observaciones

Para este proyecto necesitamos seleccionar un valor de Z_α para lo cual lo se seleccionara de la Tabla 1 que se describe a continuación:

Tabla 1. Valores de Zα para diferentes niveles de confianza

NIVEL DE CONFIANZA (1-α)		
α	**%**	Z_α
0,050	95,0	1,960
0,025	97,5	2,240
0,010	99,0	2,576

Luego obtengo el valor de Z_β que representa el poder estadístico para lo cual usamos los datos de la Tabla 2 que se encuentra a continuación:

Tabla 2. Valores de Zβ para diferentes niveles de poder estadístico

NIVEL DE CONFIANZA (1-β)		
β	**%**	Z_β
0,20	80,0	0,842
0,15	85,0	1,036
0,10	90,0	1,282

Para nuestro proyecto hemos decidido tomar los siguientes valores: nivel de confianza (1-α) 99%, diferencia mínimo observable (W) 70% y Poder estadístico (1-β) 80%. Usando estos valores, la ecuación 2 queda de la siguiente manera:

$$Z\alpha = 2{,}576 \qquad Z\beta = 0{,}842 \qquad W = 0{,}70$$

$$n = \frac{0{,}70 - 0{,}70^2\,[0{,}842 + 1.4\,(2{,}576)]^2}{0{,}70^2} = 9$$

Tenemos como resultado que el número mínimo de observaciones debe ser 9 por tanto este será aplicado para cada una de las pruebas y posterior tabulación de datos.

Al tener esta información se empezó a realizar las respectivas pruebas tomando como datos el resultado de la detección de cada ataque descritas en el subcapítulo 4.2 para esto se tomó como referencia una respuesta positiva o negativa en la detección y estos fueron los resultados en los nueve intentos.

Tabla 3. Resultados de frecuencias en pruebas exitosas o fallidas

ATAQUES	RESULTADOS		
	EXITOSOS	FALLIDOS	TOTAL
ARP SPOOFING	9	0	9
NEIGHBOR ADVERTISEMENT SPOOFING	8	1	9
DoS INVALID MAC SPOOFING	9	0	9
MITM SLAAC	8	1	9
DNS HIJACKING	7	2	9

Una vez que hemos tomado el número de observaciones suficientes procedemos a la aplicación

de estadística descriptiva. En este caso tomaremos los datos de resultados exitosos los cuales nos interesan para poder dar por valido al proyecto.

Tabla 4. Resultados de estadística descriptiva

Media	**8,2**
Moda	**9**
Desviación estándar	**0,75**
Varianza de la muestra	**0,56**
Mínimo	**7**
Máximo	**9**

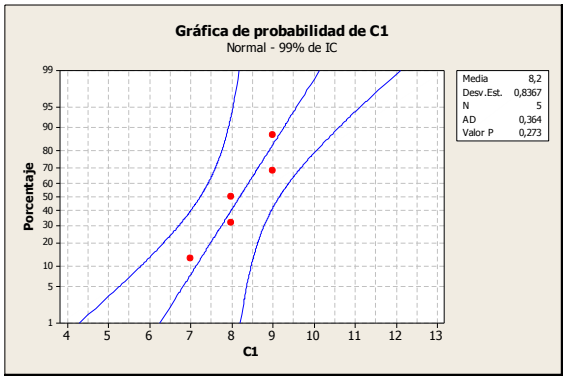

Figura 14. Gráfica de datos de la estadística descriptiva

Con los datos obtenidos podemos darnos cuenta que la media es de 8,2 siendo esto un dato favorable para la aprobación del proyecto debido a que el resto de datos no sufren grandes variaciones lo cual nos permite concluir que las pruebas fueron exitosas.

7. Conclusiones

Una vez que se ha implementado la aplicación, hemos quedado muy satisfechos, debido a que los resultados de las pruebas realizadas fueron exitosos. Hemos logrado implementar con éxito una aplicación desarrollada en Python, aprovechando sus librerías de código abierto y fácil manejo. Hemos sido capaces de detectar los cinco diferentes tipos de ataque, que anteriormente han sido especificados, entre estos estaban tres tipos diferentes de Man In The Middle, un DoS y un ataque al DNS, dentro de una red LAN basada en IPv4 como en IPv6. Se ha trabajado y se ha conseguido con éxito darle una interfaz gráfica de fácil manejo y fácil entendimiento al usuario; a fin de que tanto la ejecución del programa, como la comprensión de la detección de ataques, sean de fácil entendimiento para todos aquellos que necesiten utilizar la aplicación.

8. Referencias

[1] García Rambla, Juan Luis, Ataques en Redes de Datos en IPv4 e IPv6, España: 0xword 2nd Edition, 2013.

[2] Verdejo Álvarez, Gabriel, Seguridades en Redes IP, 1st Edition España: Universidad Autónoma de Barcelona, 2003.

[3]CISCO SYSTEMS, Seguridad en Internet, : http://www.cisco.com/web/ES/solutions/es/internet_security/index.html, fecha de consulta octubre 2014.

[4] 6DEPLOY, IPv6 Deployment and Support, http://www.6deploy.org/index.php?page=tutorials2, fecha de consulta septiembre 2014.

[5] CISCO SYSTEMS, IPv6 Basics, http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/srnd/ipv6/basics.htm, fecha de consulta septiembre 2014.

[6] Lutz, Mark, Learning PYTHON, 4th Edition United States of America: Published by O'Reilly Media, Inc., 2009.

[7] CISCO SYSTEMS, IPv6 Headers, http://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd80260042.pdf, fecha de consulta octubre 2014.

[8] ARIN, Arin IP Addressing Statistics, https://www.arin.net/knowledge/stats.pdf, fecha de consulta noviembre 2014.

[9] Seitz, Justin, Gray Hat Python, 1st Edition No Starch Press, 2009.

[10] O'Connor, TJ., Violent Python, 1st Edition United States of America: Elsevier, 2013.

[11] Van Rossum, Guido, El tutorial de Python, 1st Edition Argentina: Python Software Foundation, 2013.

[12] IPv6 MX, Fundamentos de IPv4, http://www.ipv6.mx/index.php/informacion/fundamentos/ipv4, fecha de consulta octubre 2014.

[13]Eleven Paths, Presentando Evil FOCA, http://blog.elevenpaths.com/2013/08/re-presentando-evil-foca-defcon-edition.html?q=Evil+Foca, fecha de consulta julio 2014.

[14] Bird, Steven, Natural Language Processing with Python, 1st Edition O'Reilly Media, 2009.