Published on *Revista .Seguridad* (https://revista.seguridad.unam.mx)



# Frameworks para monitoreo, forense y auditoría de tráfico de red-II (POC)

Javier Ulises Santillán Arenas

auditoría
forense
Frameworks
monitoreo
Passive Network Audit Framework
PNAF
tráfico de red
numero-25

Frameworks

En el artículo anterior se presentó una introducción general sobre tres *frameworks* de monitoreo de tráfico de red: NSM (Network Security Monitoring), SIEM (Security Information and Event Management) y PNA (Passive Network Audit). Recapitulando, tanto el objetivo del análisis como los recursos técnicos (software y hardware) con que se cuente, definen el modelo de análisis que puede ser más conveniente para el analista de tráfico. Para mayor referencia se puede consultar la Figura 1. “Panorama General de los modelos de monitoreo y análisis” del artículo anterior.

El actual trabajo tiene como objetivo presentar una prueba de concepto de Passive Network Audit Framework (PNAF), implementación de un *framework* basado en PNA el cual puede ser utilizado como herramienta de análisis pasivo de tráfico de red, aprovechando ventajas de otras herramientas. El alcance de este artículo incluye la instalación, configuración y modelos de ejecución para extracción de datos e interpretación de información. Para una mejor referencia del modelo teórico, puede consultarse el artículo anterior y la fuente original[1] de dicho *framework.*

## Modelo general de PNAF

El modelo general de PNAF (Figura 1) define el funcionamiento y flujo de datos a través del cual PNAF decodifica, filtra e interpreta información a partir del tráfico de red.

Modelo de análisis de Passive Network Audit Framework (PNAF)
Figura 1. Modelo de análisis de Passive Network Audit Framework (PNAF)

## Modos de instalación

PNAF incluye una serie de herramientas[2] para captura y análisis de tráfico de red. Debido a sus características y capacidades, algunas de estas herramientas por sí mismas pueden involucrar complejos procesos de instalación y configuración. Por esta razón, PNAF está diseñado para proveer modos de instalación que faciliten y automaticen el proceso de manera que el analista pueda hacer uso del *framework* sin mayor problema. Existen cuatro modos de instalación explicados a continuación.

#### Instalación mediante instalador

PNAF incluye un instalador que automatiza la descarga, compilación y configuración de todas las herramientas. Este instalador incluye un *wizard* (asistente de instalación) basado en *dialog*. Para poder utilizar este modo de instalación es necesario cumplir los siguientes requerimientos:

- Sistema *Debian GNU/Linux* 7.x o superior o *Gentoo stage* 3.x.
- Conexión a Internet durante la instalación (el instalador descarga e instala algunas dependencias por medio de *Apt* o *Emerge*) así como algunos módulos de Perl a través de *CPAN*.

*Es posible su funcionamiento en Ubuntu u otra distribución basada en Debian, sin embargo implica verificar las equivalencias en paquetes instalados por Apt. Asimismo, es posible usar el instalador en otros sistemas no basados en Debian siempre y cuando se instalen manualmente todas las dependencias necesarias para la compilación de las herramientas. Para esto se puede consultar el archivo README e instalar las dependencias equivalentes de la lista de Apt/emerge.*

Debido a que PNAF instala una gran cantidad de dependencias, se recomienda instalar el *framework* en un ambiente *chroot* para evitar cualquier problema de compatibilidad de dependencias en el sistema nativo. En esta prueba de concepto el proceso se hará de esa manera y se explicará a continuación.

Asumiendo que se cuenta con un sistema *Debian* 8 de 64 bits (*amd64*):

1. Creación del ambiente *chroot* (vía *debootstrap*):

```
# aptitude install debootstrap
# debootstrap --arch amd64 jessie chroot_pnaf
http://ftp.mx.debian.org/debian
# mount -t sysfs sysfs chroot_pnaf/sys
# mount -t proc proc chroot_pnaf/proc
# mount -o bind /dev chroot_pnaf/dev
# mount -o bind /dev/pts chroot_pnaf/dev/pts
```

Ahora se cambia al ambiente *chroot*:

```
# chroot chroot_pnaf          # cd ~
```

2. Descarga de PNAF

Opción 1: Desde el repositorio oficial del proyecto:

```
# aptitude install git          # git clone
https://dev.honeynet.org.mx/traffic-analysis/pnaf.git
```

Opción 2: Desde el *mirror* en github:

```
# git clone https://github.com/jusafing/pnaf
```

Una vez descargado, ingresar al directorio y ejecutar el instalador:

```
# cd pnaf            # ./install.sh
```

Esto ejecutará el asistente. Primero se deberá confirmar que se desea instalar PNAF. Posteriormente se pueden seleccionar las herramientas que se incluirán en el *framework*. A pesar de que no todas las herramientas se utilizan en esta versión de PNAF (v0.1.2) se recomienda seleccionar todas para utilizarlas de manera independiente. Visto de este modo, PNAF es también un asistente para la instalación de herramientas de análisis de tráfico de red.

Selección de herramientas en la instalación de PNAF

Image not found or type unknown

Figura 2. Selección de herramientas en la instalación de PNAF

Si es la primera vez que se instala PNAF se debe seleccionar una instalación limpia, "*clean installation*". En caso de que exista una instalación previa, la instalación limpia eliminará cualquier herramienta y archivos instalados por PNAF. Si sólo se desea agregar o reinstalar ciertas herramientas, se debe seleccionar "NO" en este paso.

Selección del tipo de instalación

Image not found or type unknown

Figura 3. Selección del tipo de instalación

A partir de ese momento el instalador comenzará a compilar y configurar todas las herramientas. Este proceso puede durar aproximadamente 30 minutos, dependiendo de las capacidades del equipo. En caso de que exista un error en el proceso, el instalador mostrará el mensaje correspondiente y se podrán verificar los archivos *install.log* e *install.log.exec* para identificar el problema. De lo contrario, si el proceso terminó correctamente, PNAF habrá quedado instalado y listo para usarse.

Como recomendación, para actualizar las variables de ambiente hay que salir del ambiente *chroot* y volver a entrar. Asimismo, para identificar cuándo se esté dentro del directorio de *chroot* es conveniente agregar una etiqueta al *shell*:

```
# echo 'PS1="(PNAF) $PS1"' >> ~/.bashrc
# exit       # chroot chroot_pnaf
```

Para verificar que PNAF se ha instalado correctamente:

```
# pnaf_auditor --help
```

Opciones de ejecución de PNAF

Image not found or type unknown

Figura 4. Opciones de ejecución de PNAF

#### Instalación mediante Debian *chroot* preconfigurado

El segundo modo de instalación corresponde al uso de un directorio *chroot* con todas las herramientas precompiladas y preconfiguradas. Esta alternativa básicamente evita todo el proceso de compilación y creación del directorio raíz con *debootstrap* (herramienta presentada en el modo anterior). Por otro lado, facilita tener una plantilla con un directorio preparado para usarse con *chroot*. Con este modo sólo es necesario descargar el archivo empaquetado *.tar.bz2* (aproximadamente 1.3 GB) y desempaquetarlo en el sistema de archivos local.

*Es importante mencionar que este modo funciona sólo si el sistema local en el que se planea instalar es Debian 8 amd64 (la compilación de todas las herramientas depende de la arquitectura y de las versiones de las dependencias usadas por Apt).*

```
# wget http://pnaf.honeynet.org.mx/download/chroot_pnaf.tar.bz2
# tar -jxvf chroot_pnaf.tar.bz2
# mount -t sysfs sysfs   chroot_pnaf/sys
# mount -t proc proc     chroot_pnaf/proc
# mount -o bind /dev     chroot_pnaf/dev
# mount -o bind /dev/pts chroot_pnaf/dev/pts
# chroot chroot_pnaf
```

Igualmente se puede verificar que PNAF se ha instalado correctamente:

```
# pnaf_auditor –help
```

#### Instalación (uso) mediante una máquina virtual

En este modo de instalación es necesario descargar una imagen de máquina virtual en formato OVA para ser importando con VirtualBox o Vmware.

1. Descargar la imagen en http://pnaf.honeynet.org.mx/download/pnaf-0.1.2.ova

2. En VirtualBox:

En el menú *File/Import appliance/* seleccionar el archivo OVA y crear la máquina virtual.

Esta máquina virtual tiene instalado PNAF con todas las opciones mostradas en el instalador. Las credenciales de acceso para el usuario *root* se muestran en el mensaje de bienvenida una vez que se inicia la máquina virtual.

Máquina virtual preconfigurada con PNAF

Image not found or type unknown

Figura 5. Máquina virtual preconfigurada con PNAF

Una vez dentro de la máquina virtual se debe ingresar nuevamente al directorio con *chroot*.

```
# mount -t sysfs sysfs   chroot_pnaf/sys
# mount -t proc proc     chroot_pnaf/proc
# mount -o bind /dev     chroot_pnaf/dev
# mount -o bind /dev/pts chroot_pnaf/dev/pts
# chroot /root/chroot_pnaf
```

**Instalación mediante módulo de Perl (instalación independiente)**

Esta opción incluye la instalación del  núcleo (*core*) de PNAF, es decir, instalación independiente del módulo de Perl. Este modo se puede utilizar cuando se desee usar PNAF con una instalación propia de las herramientas. Sin embargo, no es recomendable ya que se necesita configurar una gran cantidad de opciones, incluyendo la ruta de los archivos binarios de cada una de las herramientas, archivos de configuración, *logs*, etcétera. Para esto, una vez descargado PNAF, editar la configuración de cada una de las herramientas:

```
# cd pnaf
# vim build/pnaf/Pnaf/lib/Pnaf/Core.pm (establecer rutas)
# cd build/pnaf/Pnaf
# perl Makefile.PL
# make
# make test       # make install
```

## Configuración

La mayoría de las opciones de configuración se definen directamente como argumentos al momento de la ejecución del auditor de PNAF (*pnaf_auditor*).

Para visualizar las herramientas incluidas en el *framework* se puede ejecutar:

```
# pnaf_auditor –-version
```

Para mayor información sobre las opciones disponibles en PNAF 0.1.2:

```
# pnaf_auditor –-help
```

En caso de necesitar una configuración específica, por ejemplo, agregar firmas del IDS Suricata, se pueden modificar los archivos de configuración dentro del directorio */pnaf/etc*.

## PoC: análisis de capturas de tráfico de red

A continuación se analizarán tres archivos de captura en formato PCAP. La flexibilidad de las herramientas usadas por PNAF permite extraer e interpretar la información de maneras diferentes. El análisis en esta prueba de concepto no representa la totalidad de la información que se puede obtener. Así, la PoC incluye un análisis general los archivos de captura de manera que se obtenga la siguiente información y cuyo propósito se explica en la siguiente tabla:

Análisis de información de la PoC

Image not found or type unknown

Tabla 1. Análisis de información de la PoC

**Ejecución inicial**

***Procesamiento general***

Teniendo el archivo de captura *test1.cap*, se ejecuta *pnaf_auditor* de la siguiente manera:

```
# pnaf_auditor –-cap test1.cap –log_dir /pnaf/www/test1
```

Esto ejecutará una serie de herramientas y procesará la información almacenando los resultados en el directorio */pnaf/www/test1*.

Ejecución de PNAF

Image not found or type unknown

Figura 6. Ejecución de PNAF

***Procesamiento específico***

Ahora, asumiendo que se desea obtener un filtrado específico, se puede ejecutar *pnaf_auditor* con las siguientes opciones:

```
# pnaf_auditor –-cap test2.cap –log_dir /pnaf/www/test2 --home_net 192.168.1.0/24
```

En esta ejecución se indica que se analizará el archivo de captura *test2.cap* y que la red de la organización “ *home_net*” es el segmento 192.168.1.0/24 (se pueden indicar más segmentos en formato *CIDR**[3]*** separados por comas). Asimismo, se indica que se desea extraer el *payload* en caso de identificar una alerta de IDS (útil para análisis a fondo y verificación de falsos positivos).

***Análisis e interpretación de la información***

La interpretación de la información se puede llevar a cabo en distintas etapas.

#### *1.* Logs *de línea de comandos*

El *log* generado durante la ejecución muestra el resumen de los resultados. Esta fase es importante porque se obtiene información de las herramientas utilizadas así como del panorama general de los datos obtenidos. La siguiente figura muestra la explicación del *log* generado por PNAF.

Log de ejecución y resultados generales de PNAF

Image not found or type unknown
Figura 7. Log de ejecución y resultados generales de PNAF

#### *2.* Logs *en interfaz web*

PNAF permite una visualización de los resultados a través de una interfaz web básica. Esta interfaz permite listar:

- Archivos de *log* "*crudos*" generados por cada una de las herramientas
- Archivos de *log* preprocesados en formato JSON[4]
- Archivos de *log* con resultados de auditoría en formato JSON y visualizados en forma de árbol

Para poder utilizar la interfaz web es necesario activar el servidor web apache incluido en PNAF.

```
# apachectl start
```

Usando la configuración predeterminada, todos los directorios de resultados que se almacenen en */pnaf/www* e indicados en la opción --*log_dir* podrán ser visualizados en la web usando http://localhost o la dirección IP específica donde se ejecute PNAF.

Visualización web básica de PNAF

Image not found or type unknown
Figura 8. Visualización web básica de PNAF

Dentro de este directorio se tiene la siguiente estructura y se muestra un ejemplo en la Figura 9:

Archivos generados para la visualización web de PNAF

Image not found or type unknown

Archivos generados para la visualización web de PNAF

Image not found or type unknown
Figura 9. Archivos generados para la visualización web de PNAF

El análisis en esta PoC va de lo general a lo particular. Primero se puede dar un vistazo general en el contenido del archivo *json/summary/dataset.html*. Aquí, el analista puede acceder a la información clasificada de las herramientas. Cada herramienta contiene dos categorías principales, Summary (resumen de los datos) y Tracking (información por activos -IP, *servers*, etcétera-). Según la necesidad del analista y los hallazgos encontrados se pueden extraer datos a profundidad por cada herramienta y filtrar la información mediante el cuadro de búsqueda en cada árbol de herramienta.

Conjunto de datos (datasets) de las herramientas usadas en PNAF

Image not found or type unknown
Figura 10. Conjunto de datos (*datasets*) de las herramientas usadas en PNAF

Continuando con el análisis, ahora se visualiza el filtrado y preprocesamiento de datos generados por PNAF, el cual lleva a cabo una correlación básica y conjunta de la información, creando diferentes categorías. Para ello se accede al archivo */json/summary/auditSummary.html* y es aquí donde el analista puede extraer información detallada de la actividad de cada uno de los activos que intervienen en el tráfico de red. Por ejemplo, se puede obtener información sobre URL, certificados SSL, alertas IDS, archivos transferidos, software utilizado, etcétera. De la misma manera, si la auditoría se prefiere hacer tomando como clasificación general a los activos, entonces se puede visualizar el archivo */json/summary/auditTracking.html*.

Resumen de auditoría. Clasificación general

Image not found or type unknown
Figura 11. Resumen de auditoría. Clasificación general

Resumen de auditoría. Clasificación por activos

Image not found or type unknown
Figura 12. Resumen de auditoría. Clasificación por activos

Finalmente, el archivo */json/summary/auditOutput.html* muestra el resultado del análisis de vulnerabilidades basadas en CVE[5] y versiones de software encontradas. Este análisis incluye un sistema de puntaje (*score*) que indica el posible impacto de las vulnerabilidades identificadas. Asimismo, se muestra la lista de equipos identificados en listas negras de dominios o IP, o cuya interacción estuvo involucrada con equipos de la red.

Resumen de auditoría. Análisis de vulnerabilidad de software basado en CVE

Image not found or type unknown
Figura 13. Resumen de auditoría. Análisis de vulnerabilidad de software basado en CVE

Conjuntando toda la información recopilada y filtrada tanto por PNAF como por el mismo analista, es posible determinar el estado general y características de la red. La información detallada dependerá del tipo de problema o necesidad que se desea resolver.

Finalmente, es importante hacer énfasis en el hecho de que PNAF es susceptible a falsos positivos debido a la naturaleza misma de PNA, en donde, al no contar con la información completa y obtener datos a partir de una interpretación del tráfico de red, cierta información podría representar hechos erróneos. Así, una de las tareas del analista implica la identificación y verificación de información certera.

Actualmente PNAF se encuentra en desarrollo en la versión 0.2. Futuras versiones incluirán cambios significativos en la interfaz web y estabilidad del *framework*. Para actualizaciones consultar las páginas del proyecto.

**Si quieres saber más consulta:**

- The Honeynet Project Map
- Blog del proyecto Honeynet
- Honeynet UNAM-Chapter
- PNAF en el proyecto Honeynet

## Referencias


- Javier Ulises Santillán Arenas. (2015). "Frameworks para monitoreo, forense y auditoría de tráfico de red I" en Revista .Seguridad número 24,
- http://revista.seguridad.unam.mx/numero24/frameworks-para-monitoreo-forense-y-auditor-de-tr-fico-de-red-i
- Javier Santillan. (2014). *Passive Network Audit Framework, Master thesis*. The Netherlands : Eindhoven University of Technology.


---

[1] Javier Santillan. (2014). *Passive Network Audit Framework, Master thesis*. The Netherlands: Eindhoven University of Technology.

[2] Tabla 1. “Herramientas de análisis de tráfico de red” del artículo anterior.

[3] https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing

[4] https://en.wikipedia.org/wiki/JSON

[5] https://cve.mitre.org/

---