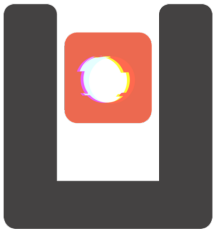

# Network Security – Laboratorio

Karina Chichifoi (@TryKatChup)

April 3, 2022

Per rilevare gli attacchi che arrivano alla nostra rete si guarda il traffico in ingresso e in uscita. Occorre quindi un sistema che:

- esamini tutto il traffico senza rallentarlo
- generi pochissimi falsi allarmi (quindi pochi falsi positivi)
- non lasci sfuggire attacchi reali (falsi negativi nulli)

- IDS (Intrusion Detection System): rilevano attività inappropriate, errate o anomale in una rete e le segnalano.
- IPS (Intrusion Protection System): è una componente hardware o software il cui scopo è quello di prevenire tentativi di attacco per gli host che fanno parte della rete.

Ad esempio un IPS può effettuare il drop dei pacchetti, il reset di una sessione o bloccare e blacklistare un host che sta eseguendo un attacco alla rete.

Esistono due tipologie di IDS:

- NIDS (Network Intrusion Detection System): i sistemi esaminano il traffico nella rete e monitorano più host per identificare le intrusioni;
- HIDS (Host Intrusion Detection System): analizzano i dati e le attività di un singolo host, come ad esempio i registri di sistema, i log, modifiche al file system e così via.

La rilevazione di attacchi che giungono via rete viene svolta analizzando il traffico in entrata/uscita. Problema essenziale:

- esaminare tutto il traffico senza rallentarlo
- generando pochissimi falsi allarmi
- senza lasciar sfuggire attacchi reali

Due approcci:

- Signature based: rileva flussi con caratteristiche notoriamente malevole
- Anomaly based: rileva flussi che si discostano dalla normalità

Tra i NIDS più diffusi abbiamo

- Snort
- Suricata
- Bro

- Software open source creato da OISF nel 2009, fork di Snort.
- Si può definire come IDPS, agisce sia da IDS che da IPS.
- Multipiattaforma.

- Supporto al multithreading
- Uso di accelerazione grafica per l'analisi tramite tecnologia nVidia CUDA
- Trigger di script in linguaggio LUA

Installazione: seguire i passi del sito della doc di Suricata
L'anno scorso avevamo una VM con Suricata con sopra Ubuntu.
Lo script di installazione è presente qua

Nel file `suricata.yaml` vengono definiti vari parametri di funzionamento per il software e i percorsi di sistema per caricare le regole o salvare i file di log, oltre che per specificare determinati comportamenti.
Solitamente si trova in `/etc/suricata/suricata.yaml`

`pid-file`, se decommentata, definisce un percorso di default per il file contenente il pid di Suricata, utilizzato in assenza del comando `--pidfile <file>`

```
# Default pid file.
# Will use this file if no --pidfile in command options.
# pid-file: /var/run/suricata.pid
```

`default-log-dir` specifica in quale directory Suricata andrà a scrivere i propri file di log.

```
default-log-dir: /var/log/suricata/
```

`default-rule-path` indica il percorso predefinito dove Suricata si aspetta di trovare i file `.rules`

```
# Set the default rule path here to search for the files.
# if not set, it will look at the current working dir
default-rule-path: /etc/suricata/rules
```

`rule-files` precede la lista di tutti i file di regole che Suricata caricherà all'avvio. I file qui indicati devono essere presenti nella directory specificata da `default-rule-path`. Qualora si volessero caricare regole esterne alla directory di default è possibile utilizzare il parametro `-s` quando si digita il comando Suricata.

```
[...]
- emerging-trojan.rules
- emerging-user_agents.rules
# - emerging-virus.rules
- emerging-voip.rules
- emerging-web_client.rules
- emerging-web_server.rules
[...]
```

Consente di utilizzare Suricata come IPS.

```
netmap:
  - interface: wg-vulnbox
    copy-mode: ips
```

Le rules sono delle istruzioni fornite all'IDPS che definiscono come il sistema debba comportarsi al presentarsi di determinati pacchetti o situazioni. Più regole inerenti un determinato argomento di sicurezza vengono raccolte all'interno di file con estensione `.rules`.

Le regole Suricata sono divise in tre parti:

- **Azione**: è l'azione che Suricata deve intraprendere al verificarsi delle condizioni definite a seguire nella regola.
- **Intestazione**: definisce il dominio di azione della regola, rendendone possibile l'innesco (trigger) solo quando un pacchetto cade all'interno di tale dominio.
- **Opzioni**: parole chiave che permettono di specificare ulteriormente il comportamento di una regola (ad esempio `msg`, `content`).

```
# Block OR 1=1
alert http any any -> any 8080 (msg:"OR 1=1 pcre"; content: "OR 1=1";
    http_uri; nocase; sid: 100014; rev: 1;)
```

In caso di match con le regole

- `pass`: Suricata smette di esaminare il pacchetto e lo lascia passare.
- `alert`: il pacchetto viene lasciato passare, ma genererà un alert sotto forma di log. È l'azione più comune intrapresa nell'intrusion detection.
- `drop`: se Suricata agisce come IPS (modalità in-line) effettua il drop del pacchetto evitando che proceda oltre. Suricata genererà poi un alert per il pacchetto.
- `reject`: verrà inviato sia al destinatario che al mittente un pacchetto di reject (`reset-packet` per TCP, `ICMP-error` per altri protocolli). Se Suricata è disposto in modalità in-line, procederà al drop del pacchetto. Verrà generato un alert per il pacchetto.

- **Protocollo**: protocollo affinché la regola venga applicata (tcp, icmp, ip, http).
- **Sorgente e Destinazione**: Sono due coppie composte da Indirizzo IP di un host o di una rete e da una porta. Si possono usare:
  - notazione CIDR
  - parole chiave come `any`
  - `$HOME_NET` e `$EXTERNAL_NET` definite in `suricata.yaml`.
- **Direzione**:
  - `->`: `<IP, porta>` a sinistra funge da sorgente, mentre a destra si ha il destinatario
  - `<-`: il contrario di `->`
  - `<>`: entrambe le direzioni

Con questi comandi Suricata carica delle regole pronte all'uso da enti che si occupano di sicurezza.

```
sudo suricata-update
sudo suricata-update list-sources
sudo suricata-update enable-source tgreen/hunting
```

```
Name: et/open
  Vendor: Proofpoint
  Summary: Emerging Threats Open Ruleset
  License: MIT
Name: oisf/trafficid
  Vendor: OISF
  Summary: Suricata Traffic ID ruleset
  License: MIT
```

Di default, Suricata, salva i propri output e alert su file di log. I vari log sono presenti nella cartella specificata nella variabile `default-log-dir` nel file `suricata.yaml`.

Su Windows genericamente viene usata la cartella `log` all'interno della cartella di installazione di Suricata. Su Linux viene spesso utilizzata la directory `/var/log/suricata`.

Sotto la voce `outputs` del file di configurazione sono elencati i vari tipi di log che Suricata può salvare.

Sono log mono-riga, prodotto degli alert generati dalle regole di Suricata. È possibile scegliere se abilitare o meno questo tipo di log e se scrivere o meno in append al file. Questo genere di log si presentano con una specifica formattazione e contengono al loro interno quello che nelle regole è specificato dalla parola chiave `msg`.

```
12/03/2012-04:05:45.999712  [**] [1:1:1] Suricata! [**]
[Classification: Not Suspicious Traffic] [Priority: 3] {TCP}
192.168.1.2:58236 -> 91.198.174.224:80
```

Si tratta del file di log che registra le informazioni riguardo al traffico HTTP di passaggio attraverso il sensore. È possibile specificare il formato del log, se scrivere in append, e se abilitarlo o meno.

```
1/07/2012-01:55:30.337469 www.google.com [**]
/logos/Logo_25wht.gif [**] Mozilla/5.0 (X11; Ubuntu;
Linux x86_64; rv:16.0) Gecko/20100101 Firefox/16.0
[**] 90.147.13.136:46933 -> 173.194.35.177:80
64
```

Domande?