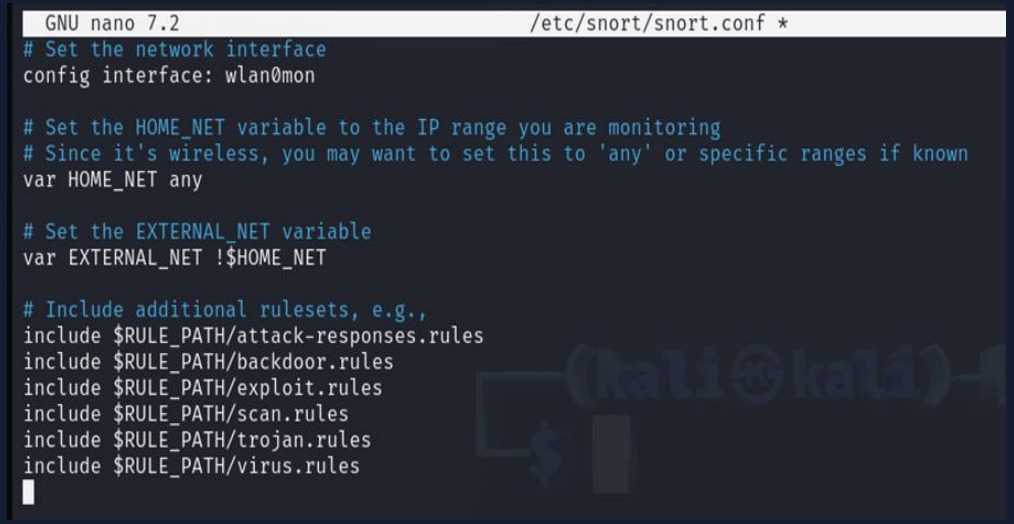
Przegląd i test wybranych rozwiązań WIDS

(Wireless Intrusion Detection System)

Michał Dziarkowski, Wojciech Matuszyński, Aleksandra Rząca, Szymon Szkarłat

Snort

- Jedno ze starszych narzędzi IDS open-source, opracowane przez Sourcefire (obecnie należy do Cisco), możliwe do dostosowania konkretnie pod WIDS.
- Wykorzystuje predefiniowane reguły do wykrywania zagrożeń. Umożliwia samodzielne definiowanie tych reguł lub pobieranie ich z zewnątrz.
- Obsługuje różne wtyczki i pluginy, które mogą wykonywać specyficzne zadania. Oprogramowanie otwarte na modyfikacje.
- Duża społeczność użytkowników, a także wsparcie od strony Cisco



Konfiguracja odbywa się niemal w całości przez dostosowanie pliku snort.conf, a także przez załączanie plików .rules definiujących reguły.

Do uruchomienia programu potrzebne jest dodatkowe narzędzie np. aircrack-ng, które przełączy docelowy interfejs do Snorta w tryb monitor.

Suricata

- Otwarte rozwiązanie IDS rozwijane przez Open Information Security Foundation.
- Umożliwia łatwą integrację z większością dużych rozwiązań SIEM, takimi jak ELK Stack lub Splunk.
- Poza zwyczajnym przechwytywaniem nagłówków pakietów w sieci, oferuje też możliwość analizy całych pakietów, nagrywanie ruchu w postaci .pcap, oraz ekstrakcję plików przesyłanych w sieci.
- Możliwość tworzenia własnych reguł detekcji i reakcji pozwalają na wykorzystanie Suricaty zarówno jako IDS, jak i IPS.
- Suricata została zaprojektowana z myślą o wykorzystaniu nowoczesnych wielordzeniowych procesorów. Dzięki temu może równocześnie przetwarzać wiele pakietów sieciowych, co znacznie zwiększa jej wydajność i pozwala na lepszą obsługę ruchu w dużych sieciach.

- Aby zacząć przechwytywać ruch za pomocą Suricaty, wystarczy zmodyfikować plik /etc/suricata/suricata.yaml, a następnie uruchomić usługę.

```
# Linux high speed capture support
af-packet:
  - interface: eth0
    cluster-id: 99
    cluster-type: cluster_flow
    defrag: yes
    use-mmap: yes
    tpacket-v3: yes
```

- Włączenie ekstrakcji plików czy nagrywania ruchu jest tak samo proste!

```
- file-store:
    version: 2
    enabled: yes
    force-filestore: yes
    force-magic: yes
    include-protocol: yes
```

```
# Cross platform libpcap capture support
pcap:
  - enabled: yes
  - filename: /var/log/suricata/suricata-%Y-%m-%d-%H:%M:%S.pcap
  - limit-size: 100mb
  - compression: gzip
  # Settings for reading pcap files
pcap-file:
  # Possible values are:
  #  - yes: checksum validation is forced
  #  - no: checksum validation is disabled
  #  - auto: Suricata uses a statistical approach to detect when
  #  checksum off-loading is used. (default)
  # Warning: 'checksum-validation' must be set to yes to have checksum tested
  checksum-checks: auto
```

alert.rules

```
alert tcp any any -> any any (msg:"rer"; content:"rer B"; content:"rer C"; flow:established,to_client; sid:1; rev:1;)
#alert http any any -> any any (msg:"next try"; sid:2; content:"devine"; content: "qui"; rev:2;)
alert http any any -> any any (msg:"next try"; sid:2; content:"loir"; content:"devine"; content: "qui"; rev:2;)
alert ip any any -> any any (msg:"rer"; sid:5; rev:3; flow:established,to_server; http.useragent; content:"toto";)
alert rdp any any -> any any (msg:"rdp test"; sid:2; rev: 4;)
alert http any any -> any any (msg:"test http"; http.host; content:"TOTO"; sid:10;)
```

Przykładowe zasady w Suricacie służące do wykrywania wyszczególnionego ruchu

Pulledpork

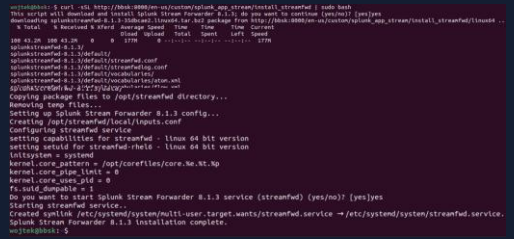
Narzędzie dedykowane dla Snorta i Suricaty. Stworzone w celu automatyzacji procesu zarządzania regułami tj. pobierania, aktualizacji i konfigurowania ich.

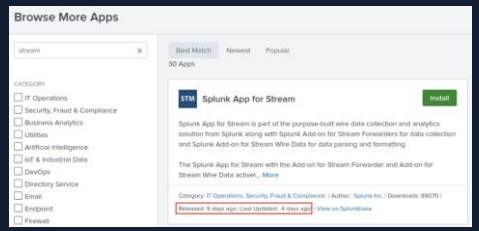
Znacznie ułatwia pracę z tymi systemami, ponieważ to właśnie odpowiednio zdefiniowane i utrzymywane reguły są kluczem do ich skutecznego działania.

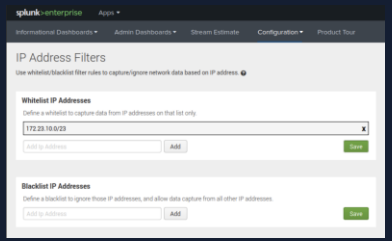
Splunk Enterprise + Splunk Stream

Splunk Enterprise to rozwiązanie komercyjne klasy big-data zdolne do przetwarzania ogromnej ilości danych. Splunk Stream to sniffer sieciowy, który wykorzystaliśmy do stworzenia ze Splunka systemu WIDS.

Rozwiązanie to w zastosowanej przez nas architekturze wymaga ręcznego zainstalowania komponentu będącego snifferem, czyli Splunk Stream Forwarded (ISF - Independent Splunk Forwarder; komunikuje się z indexerem poprzez HEC - protokół oparty na HTTP):



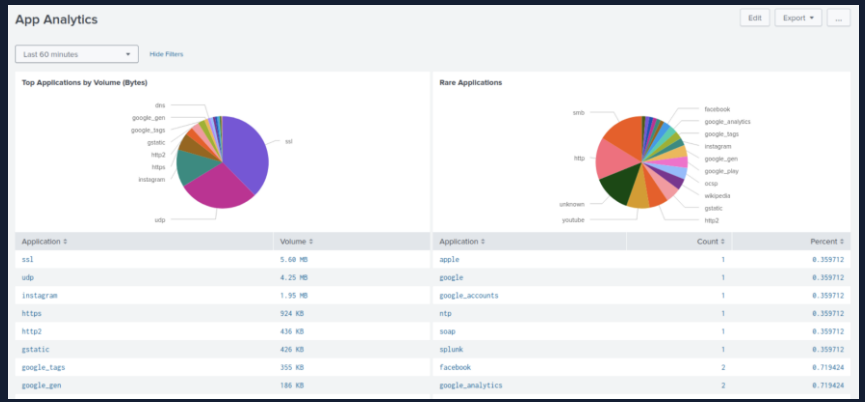


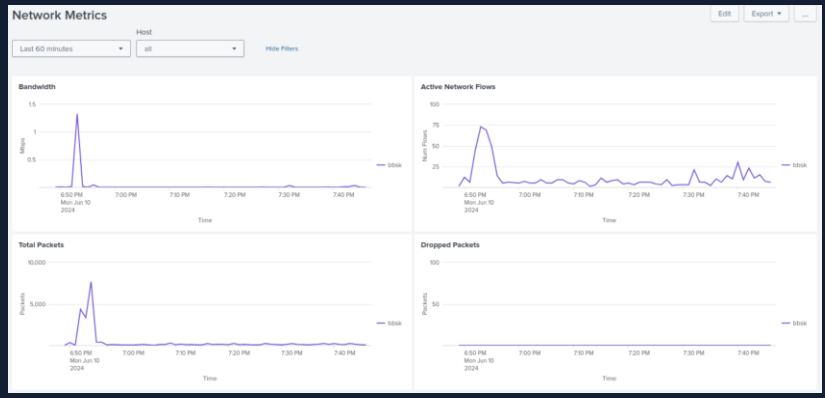


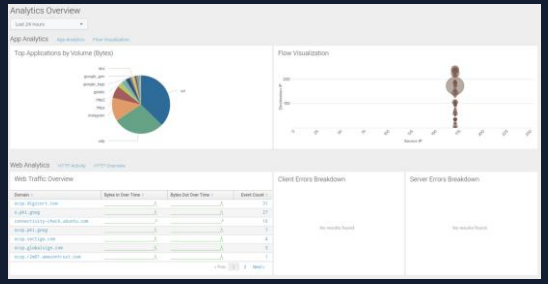

```
[streamfwd]
httpEventCollectorToken = 66cee548-d364-464c-b851-a59b61abb4bd
indexer.0.url = http://bbsk:8088
```

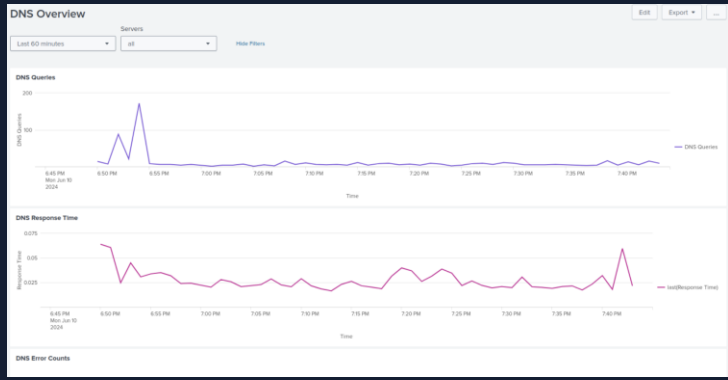
```
splunk@bbsk:~/etc/apps/splunk_httpinput/local$ cat inputs.conf
[http://streamfwd]
disabled = 0
token = 66cee548-d364-464c-b851-a59b61abb4bd
index = net
```

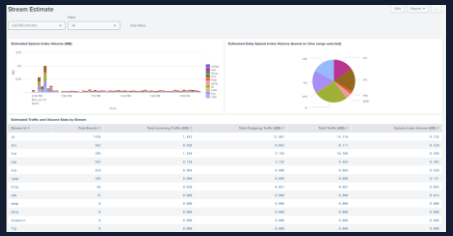
Prezentacja zebranych danych sieciowych

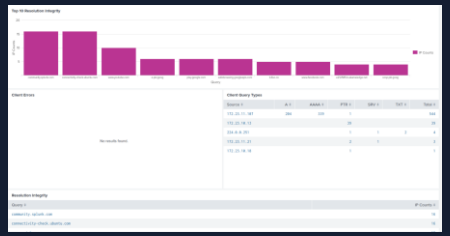


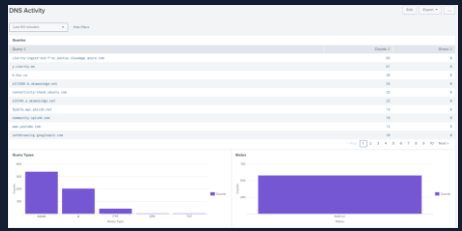












Kismet

Narzędzie Kismet, wykorzystywane przez nas podczas projektu, świetnie sprawdziło się jako **WIDS**.

- Jest to narzędzie w pełni darmowe (rozwiązanie typu *open source*), z którego korzystać mogliśmy również podczas zajęć laboratoryjnych.
- Choć, co prawda konfiguracja alertów odbywa się przez edycję pliku tekstowego (*kismet_alerts.config*), to poza tym narzędzie Kismet oferuje przejrzysty i intuicyjny interfejs graficzny, umożliwiający przyjemną analizę przechwyconych alertów (przeważnie są to próby ataków na sieć WLAN, ale nie tylko).
- Do przeprowadzania tej części projektu posłużył nam zwykły domowy AP (bardzo zbliżony parametrami do tego czym dysponowaliśmy chociażby w labie) oraz trzy laptopy: pierwszy z systemem Kali Linux (przeprowadzenie ataków oraz uruchomienie narzędzia Kismet), kolejny laptop - stacja z Win11 - umożliwił on konfigurację przewodową AP, trzecia stacja, to stacja końcowa - laptop z systemem Win11.
- Jedną z głównych wad Kismet jest fakt, że nie może on podejmować żadnych akcji związanych z przechwyconymi alertami. Działa po prostu jak **WIDS** ;)
- Natomiast jako ciekawostkę warto zaznaczyć, że można połączyć go między innymi z **Elasticsearch**.

Alerty w Kismet

- Alerty konfigurujemy w pliku *kismet_alerts.config* (fragment poniżej).

```
alert=cryptojam,5/min,1/sec
alert=dullsec,5/min,1/sec
alert=probe,5/min,1/sec
alert=broadcastdiscon,5/min,1/sec
alert=bssidthresh,5/min,1/sec
```

- Udało nam się przeprowadzić ataki: *Beacon Flooding Attack*, *Deauthentication Attack* oraz *atak z podszyciem się pod AP.*

- Przechwycone alerty: *NOCLIENTMFP*, *DEAUTHFLOOD*, *BCASTDISCON*, *ADVCRYPTCHANGE* oraz *FORMATSTRING*