

Bezpieczeństwo Bezprzewodowych Sieci Komputerowych – Projekt

Przegląd i test wybranych rozwiązań WIDS (Wireless Intrusion Detection System)

Skład zespołu: Michał Dziarkowski, Wojciech Matuszyński, Aleksandra Rząca, Szymon Szkarłat

Spis treści

Wprowadzenie do projektu.....	3
Snort.....	4
Wprowadzenie teoretyczne	4
Funkcjonalności Snort jako WIDS.....	4
Zalety użycia Snort jako WIDS	5
Konfiguracja	5
Przygotowanie hostów i sieci	5
Proces instalacji	6
Suricata.....	8
Wprowadzenie.....	8
Główne funkcje Suricaty	8
Cele stosowania Suricaty	9
Konfiguracja Suricaty.....	9
Instalacja Suricaty	9
Konfiguracja interfejsów sieciowych	10
Konfiguracja reguł.....	10
Uruchamianie Suricaty	11
Generowanie ruchu sieciowego.....	11
Weryfikacja logów i alertów	12
Przeprowadzanie ataków w celu generacji logów	12
Dodatkowe możliwości Suricaty.....	13
Ekstrakcja plików	13
Konfiguracja ekstrakcji plików	13
Wykorzystanie ekstrakcji plików	13
Tworzenie plików pcap.....	14
Podsumowanie.....	15

<i>Dodatek: PulledPork dla Snort i Suricata</i>	16
<i>Splunk Enterprise + Splunk Stream</i>	17
Wprowadzenie teoretyczne	17
Możliwości Splunk Stream	17
Wykorzystany sprzęt	18
Omówienie architektury Splunk	18
Instalacja i konfiguracja	18
Prezentacja zbieranych danych	24
Podsumowanie	29
<i>Kismet</i>	30
Wprowadzenie teoretyczne	30
Główne funkcje Kismet:.....	30
Cele stosowania Kismet.....	30
Część praktyczna	32
Konfiguracja sieci WLAN, czyli konfiguracja Access Pointa	32
Konfiguracja Kismet	34
Testowanie działania alertów, w tym przeprowadzanie ataków	37
Podsumowanie narzędzia Kismet	45
<i>Podsumowanie i wnioski</i>	46

Wprowadzenie do projektu

Wireless Intrusion Detection System (WIDS) to system zaprojektowany do wykrywania intruzów w sieciach bezprzewodowych. Jego głównym zadaniem jest monitorowanie ruchu w sieci bezprzewodowej i identyfikowanie podejrzanych aktywności, które mogą wskazywać na próby nieautoryzowanego dostępu lub ataków. Jego działanie opiera się na pasywnym monitorowaniu ruchu w sieci bezprzewodowej, co oznacza, że nie zakłóca działania sieci, ale jednocześnie analizuje wszystkie przesyłane dane.

System taki zazwyczaj składa się z kilku kluczowych komponentów, jak:

- **czujniki** rozmieszczone w strategicznych punktach sieci bezprzewodowej,
- **silniki analizy**, które przetwarzają dane analizując wzorce ruchu i identyfikując anomalie,
- **bazy sygnatur** lub reguł zawierające definicje znanych ataków i zagrożeń lub ogólnych schematów ruchu które powinny być raportowane,
- **system powiadomień i raportowania**.

Celem niniejszego projektu jest przegląd, porównanie i ocena poszczególnych systemów WIDS, oraz dostarczenie wskazówek dotyczących wyboru najlepszego rozwiązania dla różnych potrzeb użytkownika. W dużych organizacjach i infrastrukturach wyżej wymienione komponenty mogą być rozproszone na kilka lub kilkanaście maszyn, jednak ze względu na fizyczne możliwości, analizowane przez nas systemy będą łączyły w sobie wszystkie z wymienionych wyżej funkcjonalności w obrębie jednego oprogramowania/hosta.

Dokonyamy kompleksowej analizy, a następnie porównania czterech popularnych systemów WIDS, którymi są:

- Snort,
- Suricata,
- Splunk Enterprise + Splunk Stream,
- Kismet.

Celem tego porównania jest ocena ich zdolności do monitorowania i zabezpieczania sieci bezprzewodowych, a także wybranie najlepszego rozwiązania dostosowanego do specyficznych potrzeb użytkowników.

W ramach projektu skupimy się na kilku kluczowych aspektach każdego z analizowanych systemów:

1. **Dostępność:** Sprawdzimy, czy dany system jest dostępny jako oprogramowanie open source, komercyjne, czy też oferuje darmową wersję z ograniczeniami.
2. **Łatwość Instalacji i Konfiguracji:** Dokonyamy oceny procesu instalacji każdego systemu. Sprawdzimy, na ile proces ten jest przyjazny dla użytkownika oraz jakie są dostępne zasoby wspomagające w konfiguracji i użytkowaniu systemu.
3. **Skuteczność:** Dokonyamy próby zbadania, jak skutecznie każdy system wykrywa i reaguje na zagrożenia w sieciach bezprzewodowych. Niestety, nie we wszystkich przypadkach będzie to w pełni możliwe.
4. **Zalety i Wady:** Przedstawimy i porównamy główne zalety każdego z rozwiązań, uwzględniając ich unikalne cechy, różnice między nimi, ograniczenia oraz potencjalne obszary zastosowań.

Snort

Wprowadzenie teoretyczne

Snort to otwartoźródłowe narzędzie z kategorii Network Intrusion Detection System (NIDS) oraz Intrusion Prevention System (IPS) opracowany przez firmę Sourcefire, obecnie należącą do Cisco Systems. Narzędzie to jest używane do monitorowania ruchu sieciowego w czasie rzeczywistym, analizowania danych pakietów i identyfikowania podejrzanych wzorców, które mogą wskazywać na złośliwe działania, takie jak ataki, exploity lub próby nieautoryzowanego dostępu.

Snort może być używany jako Wireless Intrusion Detection System (WIDS), co jest głównym tematu niniejszego projektu. W tej roli Snort monitoruje ruch sieci bezprzewodowej, aby identyfikować i reagować na podejrzane działania. Tak jak w przypadku sieci przewodowych, w sieciach bezprzewodowych Snort identyfikuje zagrożenia za pomocą zestawu reguł. Reguły te można dostosować do specyficznych potrzeb i warunków sieci.

Funkcjonalności Snort jako WIDS

- 1. Monitorowanie w czasie rzeczywistym**
Snort zapewnia analizę ruchu w sieci bezprzewodowej w czasie rzeczywistym, co umożliwia natychmiastowe wykrywanie podejrzanych aktywności.
- 2. Podśluchiwanie i analiza pakietów**
Snort może przechwytywać i analizować pakiety przesyłane przez sieć bezprzewodową.
- 3. Wykrywanie anomalii oparte na sygnaturach**
Wykorzystuje predefiniowane reguły do wykrywania znanych zagrożeń poprzez dopasowywanie ruchu sieciowego do wzorców wskazujących na złośliwe działania.
- 4. Alarmowanie i logowanie**
Generuje alerty i logi, gdy wykryje podejrzane działania, co pomaga na szybką reakcję.
- 5. Wykrywanie nieautoryzowanych urządzeń**
Monitoruje pod kątem nieautoryzowanych punktów dostępowych (AP) i klientów, identyfikując nieuprawnione urządzenia próbujące połączyć się z siecią.
- 6. Wtyczki/pluginy**
Obsługuje różne preprocesory, które mogą wykonywać specyficzne zadania, takie jak normalizacja pakietów, defragmentacja i analiza protokołów, co zwiększa dokładność wykrywania.

Zalety użycia Snort jako WIDS

1. **Efektywność kosztowa:** Jako rozwiązanie open-source, Snort oferuje solidne możliwości WIDS bez konieczności inwestowania w kosztowne oprogramowanie komercyjne.
2. **Elastyczność i możliwość dostosowania:** Jest wysoce konfigurowalny, co pozwala użytkownikom tworzyć własne reguły i preprocesory dostosowane do specyficznych potrzeb i zagrożeń.
3. **Wsparcie społeczności i dostawców:** Korzysta z dużej społeczności użytkowników i współtwórców, a także z wsparcia i kontroli jakości zapewnianych przez Cisco, co gwarantuje wykrywanie aktualnych zagrożeń.
4. **Kompleksowe wykrywanie zagrożeń:** Łączy metody wykrywania oparte na sygnaturach i anomaliach, zapewniając kompleksową obronę przed szerokim zakresem zagrożeń.
5. **Skalowalność:** Może być wdrażany w różnych środowiskach sieciowych, od małych sieci biurowych po duże przedsiębiorstwa, zapewniając skalowalne rozwiązania bezpieczeństwa.
6. **Integracja z innymi narzędziami:** Może być zintegrowany z innymi systemami bezpieczeństwa, takimi jak rozwiązania SIEM, aby zapewnić kompleksową ochronę i scentralizowane zarządzanie incydentami.
7. **Szczegółowe raportowanie i analiza:** Oferuje rozbudowane możliwości logowania i raportowania wspomagając analizę po fakcie, oraz wpływając na poprawę przyszłych wykryć incydentów.

Konfiguracja

Przetestowanie funkcjonalności Snort jako WIDS może zostać przeprowadzone przy użyciu przygotowanych stacji lub odpowiednio skonfigurowanych maszyn wirtualnych, do tego celu jednak wymagane są dodatkowe interfejsy sieciowe pozwalające na połączenie Wifi. Poniżej przedstawiona została instrukcja krok po kroku, jak skonfigurować takie środowisko:

Przygotowanie hostów i sieci

- Stacja / maszyna wirtualna służąca za WIDS na której uruchomiony zostanie Snort
- Stacja / maszyna wirtualna symulująca atakującego (Kali Linux)
- Stacja / maszyna wirtualna będąca użytkownikiem sieci bezprzewodowej jak i jednocześnie ofiarą

Proces instalacji

1. **Instalacja Snort:** Może być to zrobione przez jeden z package managers (apt-get, yum) lub przez kompilację ze źródła (AUR packages w systemach bazowanych na Arch Linux). Należy pamiętać o instalacji zależnych pakietów (dependencies).
2. **Konfiguracja Snort:** Reguły stosowane w Snort definiowane są poprzez plik `local.rules` (`/etc/snort/rules/local.rules`) Program oferuje zestaw domyślnych reguł, ale mogą być one definiowane przez użytkownika lub pobrane z oficjalnej strony:

```
wget https://www.snort.org/downloads/community/community-rules.tar.gz
```

```
tar -xzf community-rules.tar.gz -C /etc/snort/rules
```

3. Edycja pliku konfiguracyjnego:

- Edytując plik `snort.conf` można modyfikować używane rule sets, należy upewnić się, że poprzednio pobrane reguły są uwzględnione w tym pliku. Zrobić to można przez odkomentowanie linii: `include $RULE_PATH/local.rules`
- Dodatkowo, aby dostosować Snort do ruchu bezprzewodowego, należy dodać zmienną `HOME_NET` do której przypisany będzie zakres adresów, na których będzie on działać. Przykładowo: `var HOME_NET 192.168.1.0/24`
- Należy określić też interfejs, który posłuży do monitorowania. Przykładowo: `config interface: wlan0mon`
- Następnie należy dołączyć plik z regułami `wireless.rules`: `include $RULE_PATH/wireless.rules`

Przykładowy config:

```
GNU nano 7.2 /etc/snort/snort.conf *
# Set the network interface
config interface: wlan0mon

# Set the HOME_NET variable to the IP range you are monitoring
# Since it's wireless, you may want to set this to 'any' or specific ranges if known
var HOME_NET any

# Set the EXTERNAL_NET variable
var EXTERNAL_NET !$HOME_NET

# Include additional rulesets, e.g.,
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/trojan.rules
include $RULE_PATH/virus.rules
```

4. **Ustawienie interfejsów sieciowych:** Za pomocą narzędzia `aircrack-ng` należy zlokalizować interfejs bezprzewodowy i ustawić go w tryb monitor. Przykładowo:
`sudo ifconfig wlan0 down; sudo iwconfig wlan0 mode monitor; sudo ifconfig wlan0 up`

5. **Uruchomienie monitorowania interfejsu:** Można skonfigurować airodump-ng tak, aby monitorowanie odbywało się wraz z zapisem do pliku pcap: *sudo airodump-ng -w /tmp/wireless_capture --output-format pcap wlan0*
6. **Integracja interfejsu ze Snort:** W końcu można uruchomić Snort, mówiąc mu który interfejs jest monitorowany, lub który plik pcap będzie analizowany: *sudo snort -i wlan0mon -c /etc/snort/snort.conf -A console* lub *sudo snort -r /tmp/wireless_capture-01.pcap -c /etc/snort/snort.conf*
7. **Analiza logów:** Wszystkie alerty wykryte przez Snort znajdują się w pliku */var/log/snort/alert*. Można też edytować plik *snort.conf* w celu zmiany docelowego katalogu logów: *output alert_fast: /var/log/snort/alert*

Suricata

Wprowadzenie

Suricata to zaawansowany system wykrywania włamań (IDS), który może działać również jako system zapobiegania włamaniom (IPS). Jest to narzędzie open-source, które analizuje ruch sieciowy w czasie rzeczywistym i jest wykorzystywane do zapewnienia bezpieczeństwa sieci.

Główne funkcje Suricata

1. Wykrywanie włamań (IDS/IPS):

- Suricata monitoruje ruch sieciowy w czasie rzeczywistym, wykrywając podejrzane aktywności i ataki. Dzięki napisanym algorytmom i regułom, Suricata może identyfikować różne typy zagrożeń, takie jak ataki typu DoS, skanowanie portów czy próby eksploatacji luk w zabezpieczeniach. Jej możliwości są ograniczone głównie zasadami w formacie YAML, które jej nałożymy.
- W trybie IDS, Suricata działa w trybie monitorowania, analizując ruch sieciowy i generując alerty, gdy wykryje podejrzane zdarzenia.
- W trybie IPS, Suricata działa bardziej proaktywnie, nie tylko wykrywając zagrożenia, ale również blokując podejrzane pakiety, co pozwala na natychmiastowe zatrzymanie ataków.

2. Logowanie i raportowanie:

- Suricata generuje szczegółowe logi i raporty dotyczące wykrytych zdarzeń i zagrożeń. Logi te mogą zawierać informacje o czasie zdarzenia, źródłowym i docelowym adresie IP, używanych protokołach, oraz szczegółowe dane na temat wykrytego zagrożenia.
- Dzięki bogatemu systemowi logowania, administratorzy sieci mogą dokładnie śledzić aktywność w sieci i analizować incydenty bezpieczeństwa.
- Suricata oferuje różne formaty logów, w tym JSON, co ułatwia integrację z systemami analizy logów i zarządzania incydentami (SIEM).
- Raportowanie może być dostosowane do specyficznych potrzeb organizacji, umożliwiając generowanie raportów na żądanie lub na podstawie zdefiniowanych harmonogramów.

3. Integracja z innymi narzędziami:

- Suricata może być zintegrowana z narzędziami takimi jak ELK stack (Elasticsearch, Logstash, Kibana) do zaawansowanej analizy i wizualizacji danych. Dzięki tej integracji, administratorzy mogą tworzyć interaktywne dashboardy i wizualizacje, które ułatwiają monitorowanie stanu bezpieczeństwa sieci.

- Integracja z narzędziami do zarządzania logami i SIEM pozwala na bardziej kompleksowe podejście do analizy bezpieczeństwa, umożliwiając korelację zdarzeń z różnych źródeł i szybsze reagowanie na incydenty.
- Suricata wspiera również integrację z systemami orkiestracji i automatyzacji (SOAR), co pozwala na automatyzację reakcji na wykryte zagrożenia.

4. Wydajność i skalowalność:

- Suricata jest optymalizowana do pracy w dużych i złożonych środowiskach sieciowych. Dzięki zaawansowanej architekturze, Suricata może przetwarzać duże ilości danych w czasie rzeczywistym, co jest kluczowe w dużych przedsiębiorstwach i centrach danych.
- Wsparcie dla wielowątkowości i rozproszonego przetwarzania pozwala na skalowanie Suricaty w celu obsługi rosnącego ruchu sieciowego bez utraty wydajności.
- Suricata oferuje również możliwość korzystania z akceleracji sprzętowej, takiej jak DPDK (Data Plane Development Kit), co dodatkowo zwiększa jej wydajność w środowiskach o dużej przepustowości. Jest to dlatego jeden z wiodących rozwiązań IDS w środowiskach przemysłowych

Cele stosowania Suricaty

Suricata jest preferowanym narzędziem wśród specjalistów ds. bezpieczeństwa sieci ze względu na:

- **Monitorowanie ruchu sieciowego:** Szczegółowe logi generowane przez Suricatę dają dokładny wgląd w ruch sieciowy, pozwalający na identyfikację potencjalnych anomalii.
- **Zapobieganie włamaniom:** Może działać jako IPS, blokując podejrzane pakiety i ataki.
- **Elastyczność i konfigurowalność:** Może być dostosowana do specyficznych potrzeb sieciowych i bezpieczeństwa. Inżynierowie bezpieczeństwa mogą tworzyć własne reguły identyfikujące ruch jako zagrożenie, a także własne reakcje na zdefiniowane incydenty

Konfiguracja Suricaty

Instalacja Suricaty

Na systemach Debian/Ubuntu instalacja polega na wpisaniu odpowiednio:

```
sudo apt-get install software-properties-common sudo add-apt-repository ppa:oisf/suricata-stable sudo apt update sudo apt install suricata jq.
```

Konfiguracja interfejsów sieciowych

W tym celu sprawdzamy dostępne interfejsy sieciowe zarówno dla komputera z zainstalowaną Suricata, jak i dla komputera atakującego za pomocą komendy:

```
ip a
```

W tej części mamy do czynienia z adresami IP:

- Maszyna Host - 192.168.1.100
- Maszyna Wirtualna z Suricata - 192.168.1.140
- Maszyna Wirtualna Atakująca - 192.168.1.107

Edytujemy plik konfiguracyjny /etc/suricata/suricata.yaml:

HOMENET: "[192.168.1.0/24]"

af - packet :

-interface : eth0

cluster - id : 99

cluster - type : cluster_flow

defrag : yes

use - mmap : yes

tpacket - v3 : yes

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

Konfiguracja reguł

Suricata używa reguł do wykrywania zagrożeń. Reguły można tworzyć samemu, lub przygotowane paczki pobrać z zewnętrznych źródeł. Oficjalne repozytorium zasad wspieranych przez Suricatę można pobrać (lub zaktualizować) wpisując komendę:

```
sudo suricata-update
```

Reguły przechowywane są w /var/lib/suricata/rules. Te konkretnie pobrane przez suricata-update znajdują się w tym katalogu w pliku suricata.rules

```

root@kali:~/home/kali
# cat /var/lib/suricata/rules/suricata.rules | head
alert ip any any → any any (msg:"SURICATA Applayer Mismatch protocol both directions"; flow:established; app-layer-event:applayer_mismatch_protocol_both_di
rections; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode; sid:2260000; rev:1;)
alert ip any any → any any (msg:"SURICATA Applayer Wrong direction first Data"; flow:established; app-layer-event:applayer_wrong_direction_first_data; flow
int:applayer.anomaly.count,+,1; classtype:protocol-command-decode; sid:2260001; rev:1;)
alert ip any any → any any (msg:"SURICATA Applayer Detect protocol only one direction"; flow:established; app-layer-event:applayer_detect_protocol_only_one
_direction; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode; sid:2260002; rev:1;)
alert ip any any → any any (msg:"SURICATA Applayer Protocol detection skipped"; flow:established; app-layer-event:applayer_proto_detection_skipped; flowint
:applayer.anomaly.count,+,1; classtype:protocol-command-decode; sid:2260003; rev:1;)
alert tcp any any → any any (msg:"SURICATA Applayer No TLS after STARTTLS"; flow:established; app-layer-event:applayer_no_tls_after_starttls; flowint:appla
yer.anomaly.count,+,1; classtype:protocol-command-decode; sid:2260004; rev:2;)
alert tcp any any → any any (msg:"SURICATA Applayer Unexpected protocol"; flow:established; app-layer-event:applayer_unexpected_protocol; flowint:applayer.
anomaly.count,+,1; classtype:protocol-command-decode; sid:2260005; rev:1;)
alert pkthdr any any → any any (msg:"SURICATA IPv4 packet too small"; decode-event:ipv4.pkt_too_small; classtype:protocol-command-decode; sid:2200000; rev:
2;)
alert pkthdr any any → any any (msg:"SURICATA IPv4 header size too small"; decode-event:ipv4.hlen_too_small; classtype:protocol-command-decode; sid:2200001
; rev:2;)
alert pkthdr any any → any any (msg:"SURICATA IPv4 total length smaller than header size"; decode-event:ipv4.iplen_smaller_than_hlen; classtype:protocol-co
mmand-decode; sid:2200002; rev:2;)
alert pkthdr any any → any any (msg:"SURICATA IPv4 truncated packet"; decode-event:ipv4.trunc_pkt; classtype:protocol-command-decode; sid:2200003; rev:2;)

```

Uruchamianie Suricata

Aby uruchomić Suricatę jako usługę należy wpisać komendy:

sudo systemctl start suricata # system uruchomi Suricatę w tym momencie

sudo systemctl enable suricata #Suricata będzie uruchamiana przy uruchomieniu systemu

Generowanie ruchu sieciowego

Możemy użyć narzędzi takich jak nmap do generowania ruchu i testowania reakcji

Suricata: *nmap -A 192.168.1.140*

Korzystamy z flagi -A, gdyż jest ona najbardziej dogłębna, a co za tym idzie, najgłośniejsza.

```

(kali@kali)-[~]
$ nmap -A 192.168.1.140
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-07 09:26 EDT
Nmap scan report for 192.168.1.140
Host is up (0.00039s latency). The 1.140 are in ignored states.
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh      OpenSSH 9.6p1 Debian 4 (protocol 2.0)
| ssh-hostkey:
|_ 256 4d:66:b9:77:2f:30:52:dd:54:08:7d:e8:97:f1:cf:20 (ECDSA)
|_ 256 9c:c3:fc:3f:9d:ad:2b:28:9b:58:ef:e5:0f:48:3f:53 (ED25519)
8080/tcp   open  http     SimpleHTTPServer 0.6 (Python 3.11.8)
|_ http-server-header: SimpleHTTP/0.6 Python/3.11.8
|_ http-title: Directory listing for /
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

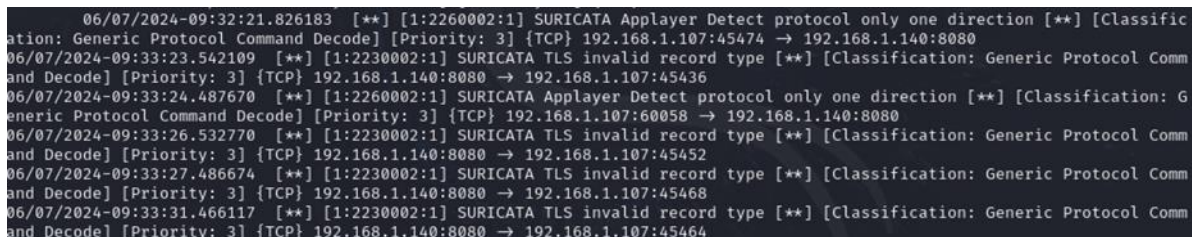
```

Figure 3: Efekt naszego ataku

Weryfikacja logów i alertów

Logi Suricata są zwykle zapisywane w `/var/log/suricata/`. Możemy przeglądać logi alertów, aby zobaczyć wykryte zdarzenia:

```
tail -f /var/log/suricata/fast.log
```



```
06/07/2024-09:32:21.826183  [**] [1:2260002:1] SURICATA Applayer Detect protocol only one direction [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.1.107:45474 → 192.168.1.140:8080
06/07/2024-09:33:23.542109  [**] [1:2230002:1] SURICATA TLS invalid record type [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.1.140:8080 → 192.168.1.107:45436
06/07/2024-09:33:24.487670  [**] [1:2260002:1] SURICATA Applayer Detect protocol only one direction [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.1.107:60058 → 192.168.1.140:8080
06/07/2024-09:33:26.532770  [**] [1:2230002:1] SURICATA TLS invalid record type [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.1.140:8080 → 192.168.1.107:45452
06/07/2024-09:33:27.486674  [**] [1:2230002:1] SURICATA TLS invalid record type [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.1.140:8080 → 192.168.1.107:45468
06/07/2024-09:33:31.466117  [**] [1:2230002:1] SURICATA TLS invalid record type [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.1.140:8080 → 192.168.1.107:45464
```

Figure 4: Pierwsze 10 linii pliku fast.log

Przeprowadzanie ataków w celu generacji logów

W celu przetestowania działania programu, potrzebujemy aby Suricata wygenerowała nam trochę logów bezpieczeństwa. Z powodu braku odpowiedniego sprzętu nie jestem w stanie przeprowadzić ataków bezprzewodowych, lecz same funkcjonalności agregacji i wizualizacji logów nie różnią się ze względu na ich rodzaj, dzięki czemu wciąż możemy je w pełni sprawdzić.

W celu generacji logów przeprowadzimy kilka ataków:

- Skanowanie SYN

```
nmap -sS 192.168.1.140
```

- Skanowanie wszystkich portów

```
nmap -p- 192.168.1.140
```

- Atak Dos

```
sudo hping3 -S -flood -p 80 192.168.1.140
```

- Atak brute-force na usługę ssh

```
hydra -l root -P .txt
```

```
ssh://192.168.1.140
```

- Atak na usługę HTTP

```
slowloris 192.168.1.140
```

Dodatkowe możliwości Suricata

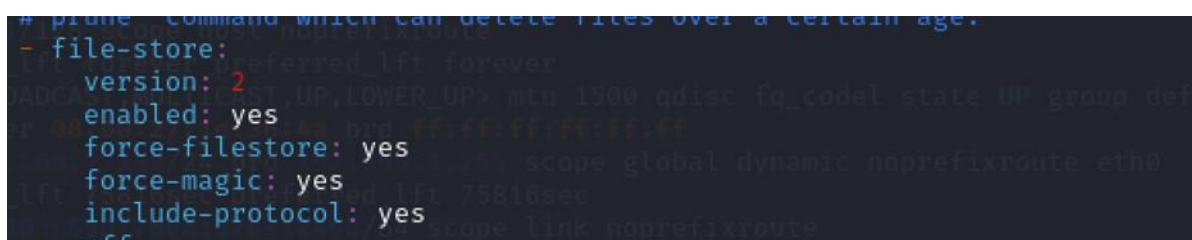
Ekstrakcja plików

Suricata oferuje możliwość ekstrakcji plików z ruchu sieciowego, co jest szczególnie przydatne w przypadku podejrzanych aktywności. Może to być wykorzystywane do analizy zawartości przesyłanych plików, w celu wykrycia potencjalnych zagrożeń.

Konfiguracja ekstrakcji plików

Aby skonfigurować ekstrakcję plików w Suricacie, należy edytować plik konfiguracyjny `suricata.yaml` i ustawić odpowiednie parametry. Poniżej znajduje się przykładowa konfiguracja:

file-store: enabled: yes force-filestore: yes force-magic: yes include-protocol: yes



```
# prune command which can delete files over a certain age.
- file-store:
  version: 2
  enabled: yes
  force-filestore: yes
  force-magic: yes
  include-protocol: yes
  xff:
```

Figure 5: Poprawnie skonfigurowany `suricata.yaml` pod względem ekstrakcji plików

- Parametr **enabled** określa, czy funkcja ekstrakcji plików jest włączona. Jeśli wartość jest ustawiona na `yes`, Suricata będzie przechwytywać i zapisywać pliki z ruchu sieciowego.
- Parametr **force-filestore** zmusza Suricatę do zapisywania wszystkich przechwyconych plików, niezależnie od innych ustawień reguł.
- Parametr **force-magic** sprawia, że Suricata używa biblioteki `libmagic` do określenia typu pliku. Jest to przydatne, gdy plik nie ma rozszerzenia lub gdy jego rozszerzenie jest mylące.
- Parametr **include-protocol** dodaje informację o protokole użytym do przesyłania pliku. Dzięki temu administratorzy mogą łatwiej analizować, w jaki sposób plik został przesłany przez sieć.

Te parametry pozwalają na szczegółową konfigurację ekstrakcji plików, umożliwiając administratorom sieci dokładne monitorowanie i analizowanie przesyłanych plików, co jest kluczowe dla utrzymania bezpieczeństwa sieci.

Wykorzystanie ekstrakcji plików

Aby skorzystać z funkcji ekstrakcji plików, należy uruchomić Suricatę z odpowiednio skonfigurowanym interfejsem sieciowym i zacząć monitorować ruch sieciowy. Gdy Suricata wykryje podejrzany ruch i przechwyci plik, będzie on automatycznie zapisywany w określonym folderze.

Tworzenie plików pcap

Suricata umożliwia również tworzenie plików pcap, które są zrzutem pakietów przechwyconych przez system. Pliki pcap są przydatne do analizy ruchu sieciowego za pomocą zewnętrznych narzędzi, takich jak Wireshark.

Konfiguracja tworzenia plików pcap

Konfiguracja tworzenia plików pcap w Suricacie odbywa się poprzez edycję pliku `suricata.yaml`. Oto przykładowa konfiguracja:

pcap: enabled: yes

filename: suricata.pcap

Parametr `enabled` określa, czy funkcja tworzenia plików pcap jest włączona. Ustawienie tego parametru na `yes` powoduje, że Suricata będzie zapisywać przechwycone pakiety do pliku pcap. Parametr `filename` definiuje nazwę pliku, do którego będą zapisywane pakiety. Można podać pełną ścieżkę do pliku, aby określić miejsce zapisu plików pcap.

Dodatkowe opcje konfiguracji plików pcap

Suricata oferuje także dodatkowe opcje konfiguracji dla tworzenia plików pcap, które mogą być przydatne w bardziej zaawansowanych scenariuszach:

- **limit-size:** Określa maksymalny rozmiar pliku pcap. Gdy plik osiągnie ten rozmiar, zostanie zamknięty, a nowy plik zostanie utworzony.
- **max-files:** Ustawia maksymalną liczbę plików pcap. Gdy liczba plików przekroczy tę wartość, najstarsze pliki będą usuwane.
- **compression:** Pozwala na kompresję plików pcap w czasie rzeczywistym, aby zaoszczędzić miejsce na dysku. Możliwe wartości to `none`, `gzip`, `lz4`.

Przykładowa bardziej zaawansowana konfiguracja może wyglądać tak:

pcap: enabled: yes filename: /var/log/suricata/suricata-limit-size: 100mb compression: gzip

```
# 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
```

Wykorzystanie plików pcap

Aby tworzyć pliki pcap za pomocą Suricata, należy uruchomić Suricatę z odpowiednio skonfigurowanym interfejsem sieciowym i monitorować ruch. Suricata będzie automatycznie zapisywać przechwycone pakiety do wskazanego pliku pcap. Pliki pcap można następnie analizować za pomocą narzędzi takich jak Wireshark, tcpdump lub inne oprogramowanie do analizy ruchu sieciowego.

Na przykład, aby otworzyć plik pcap w Wireshark, można użyć poniższej komendy w terminalu: `wireshark /var/log/suricata/suricata-2024-06-01-18:30:10.pcap`

Pliki pcap są niezwykle przydatne do szczegółowej analizy ruchu sieciowego, identyfikacji wzorców ataków, diagnozowania problemów z siecią oraz przeprowadzania retrospektywnej analizy incydentów bezpieczeństwa.

Podsumowanie

Główne funkcje Suricata obejmują:

- **Wykrywanie włamań (IDS/IPS):** Suricata potrafi wykrywać szeroki zakres zagrożeń, takich jak ataki typu DoS, skanowanie portów, czy próby eksploatacji luk w zabezpieczeniach. W trybie IPS umożliwia natychmiastowe blokowanie podejrzanych pakietów.
- **Logowanie i raportowanie:** Generuje szczegółowe logi i raporty, co ułatwia analizę incydentów oraz monitorowanie stanu bezpieczeństwa sieci. Obsługa różnych formatów logów, w tym JSON, ułatwia integrację z systemami SIEM.
- **Integracja z innymi narzędziami:** Możliwość integracji z ELK stack, systemami SIEM oraz narzędziami do orkiestracji i automatyzacji (SOAR) pozwala na zaawansowaną analizę i szybsze reagowanie na incydenty.
- **Wydajność i skalowalność:** Optymalizowana do pracy w dużych i złożonych środowiskach, Suricata wspiera wielowątkowość oraz akcelerację sprzętową, co pozwala na obsługę dużych ilości danych bez utraty wydajności.

Suricata wyróżnia się także dodatkowymi możliwościami, takimi jak ekstrakcja plików z ruchu sieciowego oraz tworzenie plików pcap do szczegółowej analizy. Te funkcje są niezwykle przydatne dla administratorów sieci, którzy potrzebują zaawansowanych narzędzi do monitorowania i analizowania ruchu sieciowego. Wszystkie te funkcjonalności dostępne są out-of-the-box, wystarczy odpowiednio zmienić plik konfiguracyjny.

Dzięki elastyczności i konfigurowalności, Suricata może być dostosowana do specyficznych potrzeb organizacji, co czyni ją idealnym rozwiązaniem dla różnorodnych środowisk sieciowych. Jej otwartość na integrację z innymi narzędziami oraz zdolność do przetwarzania dużych ilości danych w czasie rzeczywistym sprawiają, że jest jednym z najważniejszych narzędzi w arsenale specjalistów ds. bezpieczeństwa sieci.

Dodatek: PulledPork dla Snort i Suricata

PulledPork to narzędzie dedykowane dla systemów Snort i Suricata. Zostało stworzone w celu automatyzacji procesu zarządzania regułami dla tych programów. Obejmuje automatyzację procesów pobierania, aktualizacji i zarządzania regułami IDS. Tak jak zostało to dokładnie opisane wcześniej, reguły te są kluczowe dla działania tych programów, ponieważ definiują wzorce ruchu sieciowego, które mają być monitorowane i potencjalnie zgłaszane jako zagrożenia.

Główne funkcje PulledPork:

- **Pobieranie reguł:** PulledPork może automatycznie pobierać najnowsze reguły z różnych źródeł, takich jak Snort.org czy inne repozytoria.
- **Aktualizacja reguł:** Narzędzie porównuje aktualnie zainstalowane reguły z nowymi wersjami i wprowadza aktualizacje, co zapewnia, że system IDS jest zawsze na bieżąco z najnowszymi zagrożeniami.
- **Zarządzanie regułami:** PulledPork pozwala na selektywne włączanie lub wyłączanie reguł na podstawie określonych kryteriów, takich jak kategorie zagrożeń czy specyficzne identyfikatory.
- **Konwersja reguł:** Narzędzie jest w stanie konwertować reguły z formatu Snort do formatu Suricata, co umożliwia wykorzystanie tych samych reguł w obu systemach.
- **Generowanie list SID:** PulledPork może generować listy identyfikatorów (SID) reguł, które są pomocne w zarządzaniu wyjątkami i modyfikacjami reguł.

Konfiguracja i użytkowanie:

- **Plik konfiguracyjny:** PulledPork korzysta z pliku konfiguracyjnego (domyślnie *pulledpork.conf*), w którym użytkownik definiuje źródła reguł, ścieżki do plików wyjściowych, preferencje dotyczące kategorii reguł, itp.
- **Wymagania:** Narzędzie wymaga Perl oraz kilku modułów Perla, które są niezbędne do działania skryptu.
- **Kompatybilność:** PulledPork jest kompatybilny zarówno ze Snortem, jak i Suricata, co czyni go uniwersalnym narzędziem dla administratorów IDS.

Splunk Enterprise + Splunk Stream

Wprowadzenie teoretyczne

W tym rozdziale omówimy komercyjne rozwiązanie do monitorowania ruchu sieciowego jakim jest obecnie produkt firmy Cisco – Splunk Enterprise. Jest to wysoce wydajna platforma klasy big-data do przetwarzania ogromnych ilości danych będących na ogół logami z całych infrastruktur. W naszym przypadku Splunk posłuży do analizy logów typowo sieciowych, będących podsłuchanym ruchem sieciowym na hoście.

Jako sniffera użyjemy również rozwiązania Splunkowego, a konkretnie Splunk Stream. Splunk Stream to dodatek do Splunk, który specjalizuje się w zbieraniu i analizie ruchu sieciowego. Splunk Stream umożliwia przechwytywanie, filtrowanie, indeksowanie i analizowanie strumieni danych o zdarzeniach sieciowych. "Strumień" to grupa zdarzeń zdefiniowanych przez określony protokół sieciowy i zestaw pól. W połączeniu z dziennikami, metrykami i innymi informacjami, strumienie przechwycone za pomocą Splunk Stream mogą zapewnić cenny wgląd w działania i podejrzane zachowania w całej infrastrukturze sieciowej.

Możliwości Splunk Stream

Splunk Stream idealnie nadaje się do:

- Pasywnego przechwytywania strumieni danych o zdarzeniach sieciowych na żywo
- Przechwytywania metadanych i pełnych strumieni pakietów dla wielu protokołów sieciowych
- Zbierania danych protokołu NetFlow
- Stosować metody agregacji do statystycznej analizy danych zdarzeń
- Stosowanie filtrów w celu zminimalizowania wymagań indeksera
- Wyodrębnianie zawartości z ciągów znaków i generowanie skrótów
- Wyodrębnianie plików z ruchu sieciowego
- Monitorowanie trendów sieciowych i wydajności aplikacji na gotowych pulpitych nawigacyjnych
- Wdrażanie Niezależnego Stream Forwardera do przechwytywania danych na zdalnych maszynach linuxowych
- Szybkie i dyskretne skalowanie bez potrzeby ręcznego tagowania

Wykorzystany sprzęt

Na potrzeby eksperymentów skorzystamy z systemu operacyjnego Ubuntu Desktop 22.04 LTS. System zainstalowany jest na laptopie z wbudowaną bezprzewodową kartą sieciową w standardzie WiFi 4. Karta obsługuje zarówno pasmo 2,4 GHz jak i 5 GHz oraz 802.11a/b/g/n. Dokładna specyfikacja karty sieciowej: [Intel® Centrino® Advanced-N 6205, Dual Band](#)

Omówienie architektury Splunk

Na potrzeby przeprowadzenia eksperymentów, omówiony wyżej laptop posłuży nam za serwer zarówno pod Splunk Enterprise jak i osobny komponent Splunka do sniffowania ruchu, czyli Independent Splunk Forwarder (ISF). W przypadku Splunk Enterprise będzie to instalacja w modelu AiO (all in one), co oznacza, że wszystkie komponenty Splunka są na jednej maszynie. Najważniejszymi komponentami jest SH (Search Head) oraz INDX (Indexer). Do tego pierwszego logują się użytkownicy i służy on do przeszukiwania danych poprzez uruchamianie zapytań w języku SPL (Splunk Search Processing Language) oraz agreguje wyniki zapytań z klastrów indeksatorów w środowiskach rozproszonych. Indeksator natomiast jest miejscem, do którego trafiają ostatecznie wszelkie logi. To na nim są indeksowane (następuje m.in. ekstrakcja pól np. na podstawie regexów) i zapisywane na dysku. Innym komponentem jest jeszcze DS (Deployment Server), który w większych środowiskach pozwala na instalację aplikacji i add-onów na Splunkowe Forwardery służące przekazywaniu logów/danych z przeróżnych systemów na indeksery. My jednak na nasze potrzeby nie będziemy korzystać z funkcjonalności DS, lecz ręcznie zainstalujemy ISF na ten samej maszynie, co Splunk AiO.

/opt/splunk – tutaj będzie zainstalowany Splunk Enterprise (posiadający web GUI i umożliwiający monitoring aktywności)

/opt/streamfwd – tutaj będzie zainstalowany Independent Splunk Forwarder (sniffujący ruch i przesyłający logi do Splunka korzystając z HEC)

HEC – HTTP Event Collector, czyli transfer logów przy użyciu protokołu HTTP na (domyślnie) port 8088; w ten sposób ISF wysyła generowane logi do Splunka (więcej o tym w kolejnej części)

Instalacja i konfiguracja

Na początku instalujemy Ubuntu wraz z wykonaniem wszelkich aktualizacji, a następnie instalujemy Splunka używając pakiet deb, który automatycznie tworzy w systemie użytkownika 'splunk'.

```

wojtek@bbsk:~$ wget "https://download.splunk.com/products/splunk/releases/9.2.1/linux/splunk-9.2.1-78803f08aabb-linux-2.6-amd64.deb"
--2024-06-09 21:26:04-- https://download.splunk.com/products/splunk/releases/9.2.1/linux/splunk-9.2.1-78803f08aabb-linux-2.6-amd64.deb
Resolving download.splunk.com (download.splunk.com)... 108.138.51.88, 108.138.51.13, 108.138.51.70, ...
Connecting to download.splunk.com (download.splunk.com)|108.138.51.88|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 545652596 (520M) [binary/octet-stream]
Saving to: 'splunk-9.2.1-78803f08aabb-linux-2.6-amd64.deb'

splunk-9.2.1-78803f08aabb-linux-2.6-amd64 100%[=====] 520,37M  65,6MB/s   in 7,9s

2024-06-09 21:26:13 (66,0 MB/s) - 'splunk-9.2.1-78803f08aabb-linux-2.6-amd64.deb' saved [545652596/545652596]

wojtek@bbsk:~$ ls
Desktop  Music  snap          test-01.cap      test-01.kismet.netxml  test-02.csv      test-02.log.csv
Documents Pictures splunk-9.2.1-78803f08aabb-linux-2.6-amd64.deb  test-01.csv      test-01.kismet.netxml  test-02.kismet.csv  Videos
Downloads Public  Templates      test-01.kismet.csv  test-02.cap      test-02.kismet.netxml

wojtek@bbsk:~$
wojtek@bbsk:~$
wojtek@bbsk:~$ sudo dpkg -i splunk-9.2.1-78803f08aabb-linux-2.6-amd64.deb
Selecting previously unselected package splunk.
(Reading database ... 182411 files and directories currently installed.)
Preparing to unpack splunk-9.2.1-78803f08aabb-linux-2.6-amd64.deb ...
Unpacking splunk (9.2.1+78803f08aabb) ...
Setting up splunk (9.2.1+78803f08aabb) ...
/var/lib/dpkg/info/splunk.postinst: line 60: curl: command not found
complete

```

Ręcznie doinstalowujemy narzędzie curl – przyda nam się potem, na razie jego brak nic nie zmienia.

Po pomyślnej instalacji przelogowujemy się na ‘splunk’a i po raz pierwszy uruchamiamy usługę akceptując regulamin i tworząc użytkownika administracyjnego.

```

wojtek@bbsk:~$ sudo su - splunk
splunk@bbsk:~$ pwd
/opt/splunk
splunk@bbsk:~$ ls
bin      copyright.txt  ftr      lib      openssl  quarantined_files  share      swldtag
cnake    etc            include  license-eula.txt  opt      README-splunk.txt  splunk-9.2.1-78803f08aabb-linux-2.6-x86_64-manifest  var
splunk@bbsk:~$ cd bin
splunk@bbsk:~/bin$ ./splunk start --accept-license

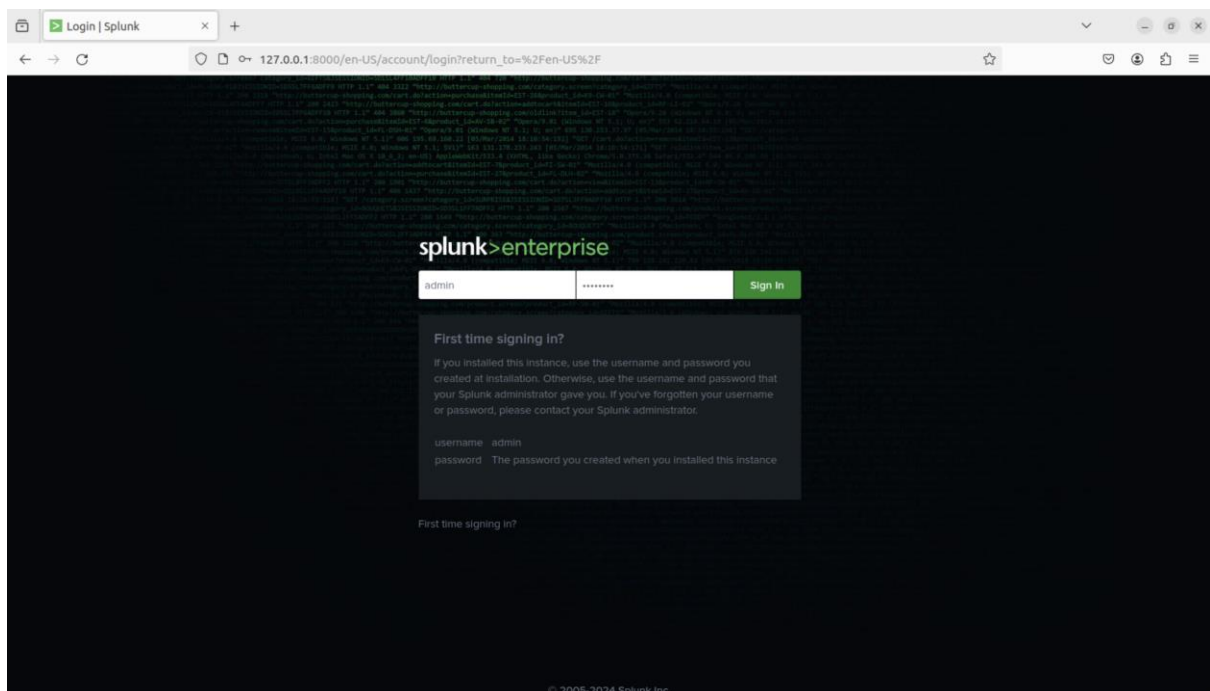
This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: admin
Password must contain at least:
* 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:

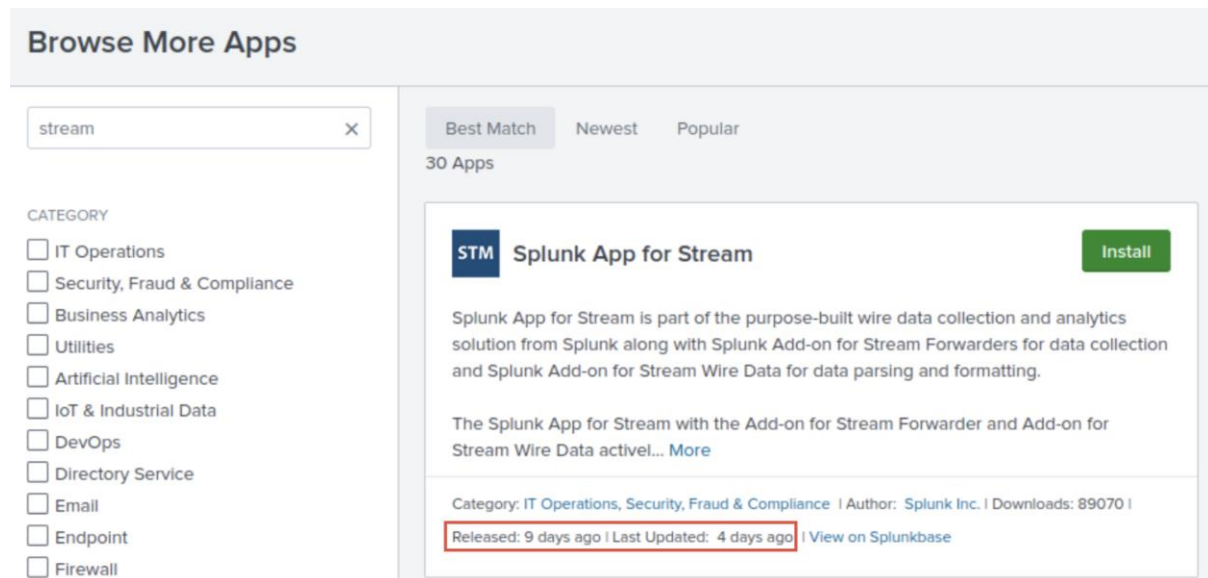
```

Możemy już zalogować się do interfejsu graficznego Splunk Enterprise (przy pomocy użytkownika, którego utworzyliśmy i nadaliśmy hasło w poprzednim kroku instalacji z CLI):

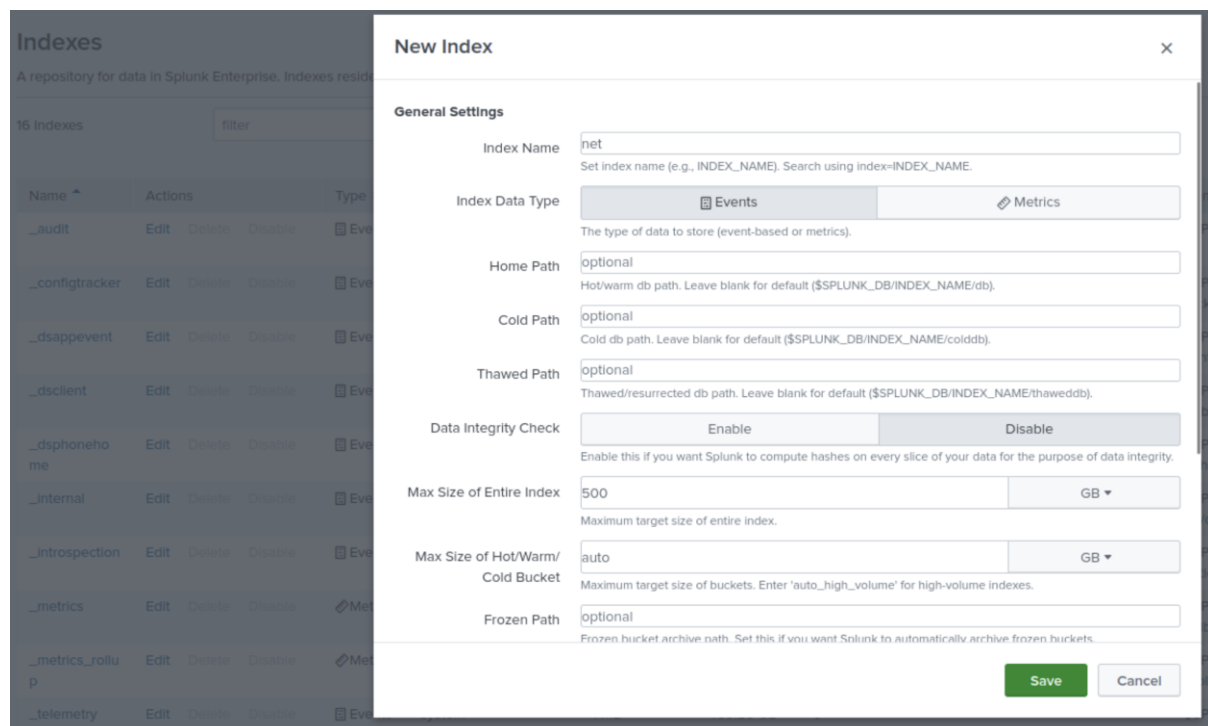


Podczas tego projektu nie będziemy się skupiać na przedstawieniu niezliczonej funkcjonalności platformy, lecz opiszemy konfigurację oraz instalację pakietu pozwalającego nam stworzyć system IDS/WIDS.

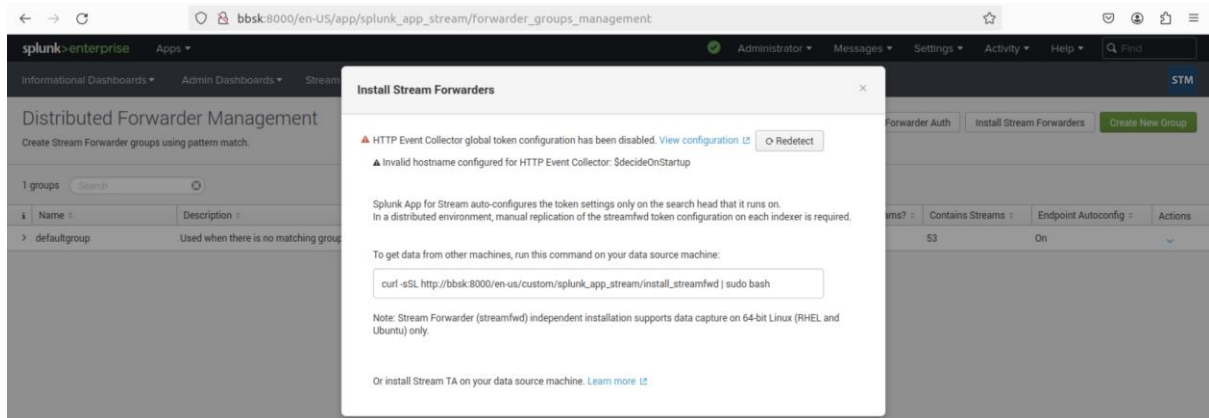
W obrębie Splunk instalujemy aplikację „Splunk App for Stream”. Jej zadanie to przedwstępna konfiguracja (wygenerowanie tokenu HEC i dodanie wpisu umożliwiającego odbiór logów za jego pomocą) oraz późniejsze monitorowanie ruchu sieciowego dzięki wbudowanym dashboardom. Warto zwrócić uwagę, że korzystamy z wersji sprzed niespełna tygodnia, co pokazuje, że rozwiązanie jest cały czas utrzymywane i aktywnie aktualizowane.



W Splunku podstawowym bytem przechowywującym zaindeksowane logi są indeksy. Pozwalają przykładowo rozdzielić logi sieciowe od tych, pozyskanych z systemów operacyjnych. Na potrzeby naszego WIDSa utworzyliśmy indeks o nazwie „net”:



Przyszła czas na instalację ISF. W aplikacji Stream mamy możliwość wygenerowania gotowego skryptu wraz z komendą instalującą ISF odwołującą się do aplikacji Splunk Stream:



Komendę kopiujemy i wklejamy w konsolkę Ubuntu (to tutaj przyda się wcześniej doinstalowany curl):

```
wojtek@bbsk:~$ curl -sSL http://bbsk:8000/en-us/custom/splunk_app_stream/install_streamfwd | sudo bash
This script will download and install Splunk Stream Forwarder 8.1.3; do you want to continue (yes/no)? [yes]yes
downloading splunkstreamfwd-8.1.3-35dbcae2.linux64.tar.bz2 package from http://bbsk:8000/en-us/custom/splunk_app_stream/install_streamfwd/linux64 ..
% Total    % Received % Xferd  Average Speed   Time    Time     Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 43.2M 100 43.2M    0     0  177M      0 --:--:-- --:--:-- --:--:-- 177M
splunkstreamfwd-8.1.3/
splunkstreamfwd-8.1.3/default/
splunkstreamfwd-8.1.3/default/streamfwd.conf
splunkstreamfwd-8.1.3/default/streamfwdlog.conf
splunkstreamfwd-8.1.3/default/vocabularies/
splunkstreamfwd-8.1.3/default/vocabularies/aton.xml
splunkstreamfwd-8.1.3/default/vocabularies/phonetic/ff1ow_vml
Copying package files to /opt/streamfwd directory...
Removing temp files...
Setting up Splunk Stream Forwarder 8.1.3 config...
Creating /opt/streamfwd/local/inputs.conf
Configuring streamfwd service
setting capabilities for streamfwd - linux 64 bit version
setting setuid for streamfwd-rhel6 - linux 64 bit version
initssystem = systemd
kernel.core_pattern = /opt/corefiles/core.%e.%t.%p
kernel.core_pipe_limit = 0
kernel.core_uses_pid = 0
fs.suid_dumpable = 1
Do you want to start Splunk Stream Forwarder 8.1.3 service (streamfwd) (yes/no)? [yes]yes
Starting streamfwd service..
Created symlink /etc/systemd/system/multi-user.target.wants/streamfwd.service → /etc/systemd/system/streamfwd.service.
Splunk Stream Forwarder 8.1.3 installation complete.
wojtek@bbsk:~$
```

```
wojtek@bbsk:~$ cd /opt
wojtek@bbsk:/opt$ ls -la
total 20
drwxr-xr-x  5 root      root      4096 cze 10 18:19 .
drwxr-xr-x 20 root      root      4096 cze  9 19:14 ..
drwxrwxrwx  2 root      root      4096 cze 10 18:19 corefiles
drwxr-xr-x 14 splunk    splunk    4096 cze 10 18:14 splunk
drwxr-x--- 11 streamfwd streamfwd 4096 cze 10 18:19 streamfwd
wojtek@bbsk:/opt$ cd streamfwd/
bash: cd: streamfwd/: Permission denied
wojtek@bbsk:/opt$ cd splunk/
wojtek@bbsk:/opt/splunk$
```

ISF jest instalowany na użytkownika streamfwd, taki użytkownik jest w systemie oczywiście tworzony, lecz nie mamy możliwości przelogowania się na niego. Wszelkie zmiany dotyczące ISF wykonujemy jednorazowo za pomocą konta root. W przypadku środowisk produkcyjnych (o których wspomniałem we wprowadzeniu) zamiast konfigurować każdy system, z którego chcemy sniffować ruch, posługujemy się również wcześniej wspomnianym DS i automatycznie dystrybuujemy gotowe paczki aplikacji na wszystkie systemy na raz.

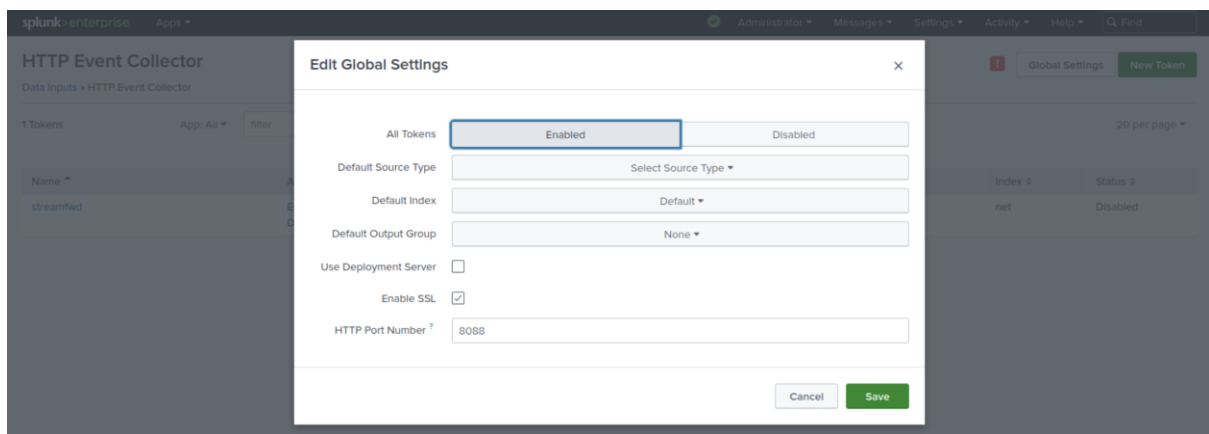
Tworzymy plik streamfwd.conf w lokalizacji /opt/streamfwd/local, aby wskazać adres indeksera oraz wklejamy token skopiowany wcześniej ze Splunk Enterprise.

```
root@bbsk: /opt/streamfwd/local

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

~

"streamfwd.conf" [New] 3L, 108B written
```



Powyższy zrzut ekranu koresponduje z plikiem /opt/splunk/etc/apps/splunk_httpinput/local/inputs.conf, w którym zdefiniowany jest „input” (sposób w jaki dany komponent Splunka przyjmuje wysyłane mu logi, w naszym przypadku HEC wraz z korespondującym tokenem – to są wpisy stworzone automatycznie dzięki aplikacji Splunk Stream) po stronie indeksera:

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

Po wprowadzeniu zmian w ISF, należy tę usługę zrestartować:

```
wojtek@bbsk:/opt/splunk$ sudo systemctl restart streamfwd
```

Sprawdzając jednak jej status otrzymujemy następujący błąd:

```
2024-06-10 18:37:28 WARN [129713913697856] (HTTPRequestSender.cpp:1478) stream.SplunkSenderHTTPEventCollector - (#5) Recovery attempt failed
2024-06-10 18:37:28 WARN [129713913697856] (HTTPRequestSender.cpp:1485) stream.SplunkSenderHTTPEventCollector - (#6) DNS lookup failed for "$decideOnStartup": Host not found (authoritative)
```

Co ciekawe, jest to swego rodzaju błąd Splunka, ponieważ fraza \$decideOnStartup widniejąca w jednym z plików konfiguracyjnych dodaje jako nazwę serwera Splunkowego hostname systemowy do innego pliku na stałe. Independent Splunk Forwarder opiera się jednak o pierwotny plik nie mogąc tym samym poprawnie działać. Niedoskonałość korygujemy ręcznie tworząc plik inputs.conf (tym razem po stronie indeksera) w lokalizacji /opt/splunk/etc/system/local zawierający wyłącznie stanzę default wraz z wpisem definiującym nazwę hosta:

* plik zawierający inicjalizacyjną wartość znajduję się w ../default a w hierarchii budowania runtime'owej konfiguracji Splunka pliki w folderach local mają wyższy priorytet niż default.

```
[default]
host = bbsk
```

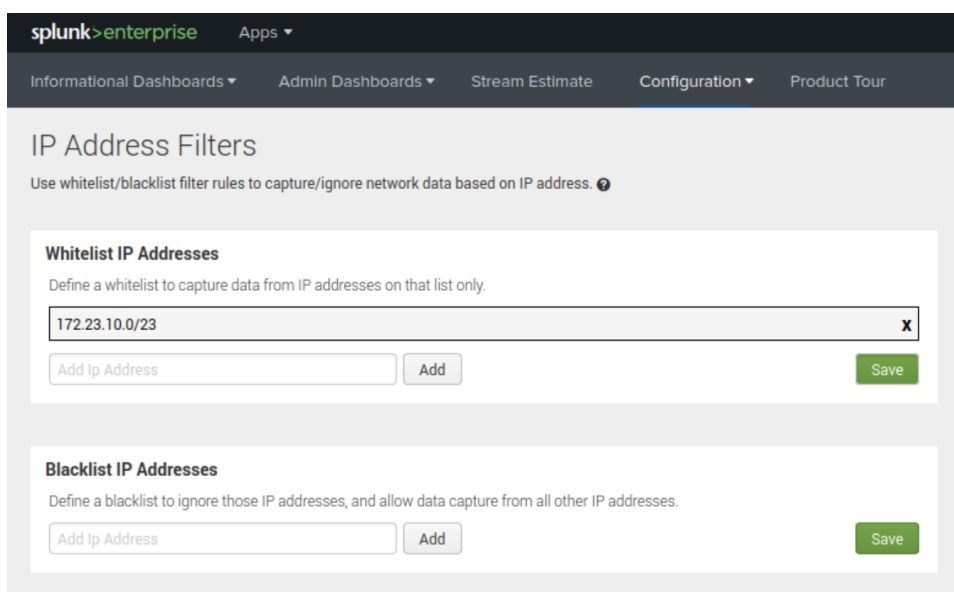
Po wprowadzeniu zmiany restartujemy usługę splunkd (czyli Splunk Enterprise) a następnie ponownie usługę splunkfwd (czyli Independent Splunk Forwarder).

```
root@bbsk:/opt/splunk# systemctl restart streamfwd
root@bbsk:/opt/splunk# systemctl status streamfwd
● streamfwd.service - Splunk Stream Forwarder 8.1.3
   Loaded: loaded (/etc/systemd/system/streamfwd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-06-10 18:46:17 CEST; 4s ago
     Process: 46270 ExecStartPre=/bin/sh -c ulimit -s 512 (code=exited, status=0/SUCCESS)
    Main PID: 46272 (streamfwd)
       Tasks: 7 (limit: 18977)
      Memory: 4.3M
         CPU: 28ms
       CGroup: /system.slice/streamfwd.service
              └─46272 /opt/streamfwd/bin/streamfwd -D

cze 10 18:46:17 bbsk systemd[1]: Starting Splunk Stream Forwarder 8.1.3...
cze 10 18:46:17 bbsk systemd[1]: Started Splunk Stream Forwarder 8.1.3.
```

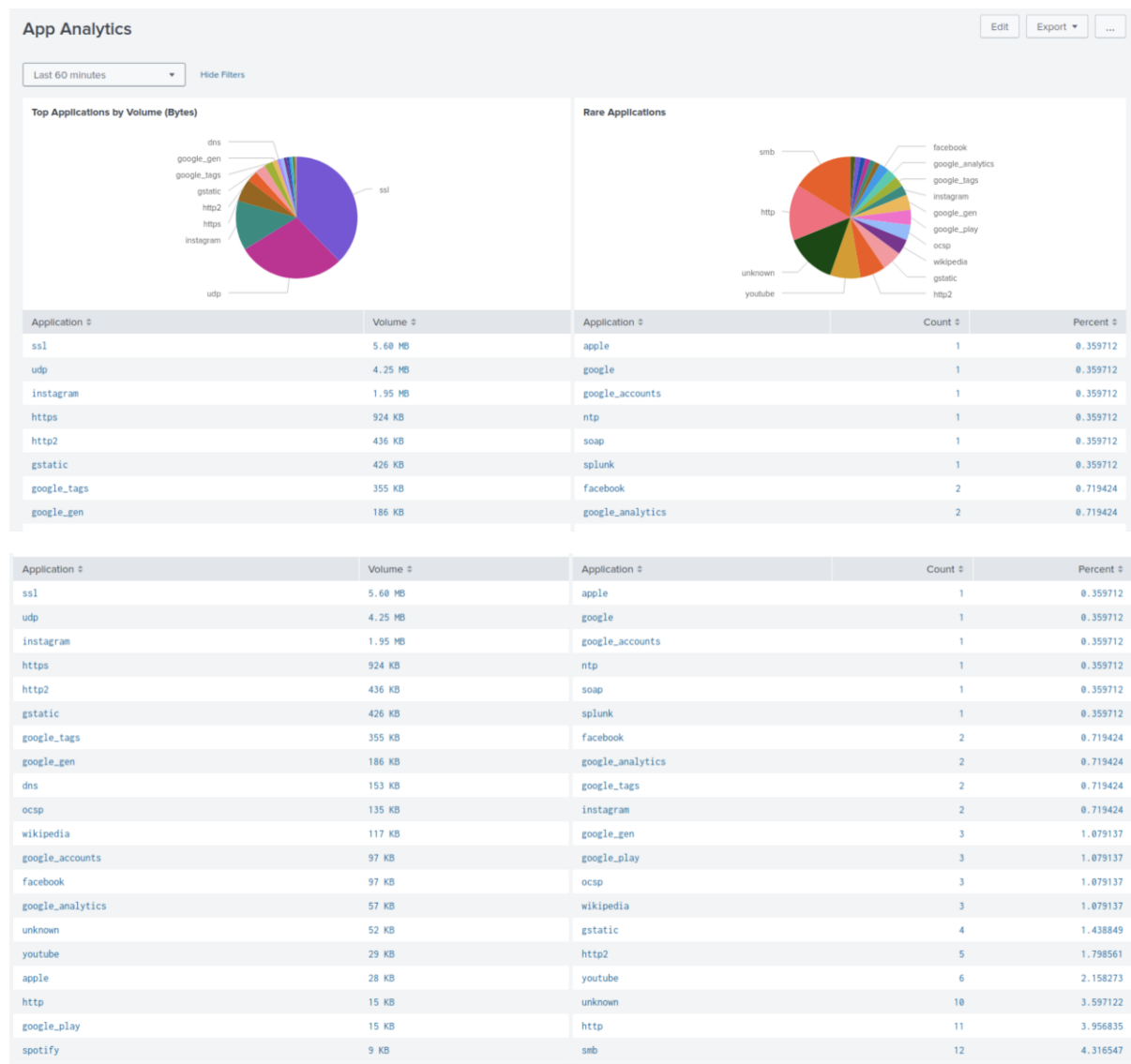
Obecnie wszystko już jest w porządku a journald nie zgłasza żadnych wpisów o jakichkolwiek błędach.

Po stronie Splunk App for Stream dodajemy adresy IP, z/do których ruch ma być monitorowany i gotowe.



The screenshot shows the 'IP Address Filters' configuration page in the Splunk Enterprise web interface. The page has a dark header with the 'splunk>enterprise' logo and a navigation bar with links: 'Informational Dashboards', 'Admin Dashboards', 'Stream Estimate', 'Configuration', and 'Product Tour'. The main content area is titled 'IP Address Filters' and includes a subtitle: 'Use whitelist/blacklist filter rules to capture/ignore network data based on IP address.' There are two sections: 'Whitelist IP Addresses' and 'Blacklist IP Addresses'. The 'Whitelist' section has a text input field containing '172.23.10.0/23', an 'Add' button, and a 'Save' button. The 'Blacklist' section has an empty text input field, an 'Add' button, and a 'Save' button.

Prezentacja zbieranych danych



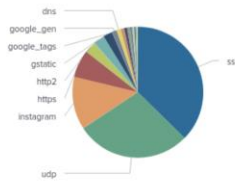
Już tutaj mamy pewnego rodzaju ciekawostkę. W tabeli widnieje ruch do serwisu zidentyfikowanego przez Splunk jako spotify czy google_play. Nie otwierałem stron żadnego z tych serwisów, ale jak widać inne otwarte (np. testowo Facebook, Instagram, Wikipedia, YouTube) się do nich odwołują...

Analytics Overview

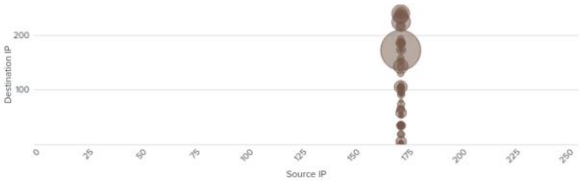
Last 24 hours

App Analytics

Top Applications by Volume (Bytes)



Flow Visualization



Web Analytics

Web Traffic Overview

Domain	Bytes In Over Time	Bytes Out Over Time	Event Count
ocsp.digicert.com			31
o.pki.goog			27
connectivity-check.ubuntu.com			10
ocsp.pki.goog			7
ocsp.sectigo.com			4
ocsp.globalsign.com			3
ocsp.r2m01.amazontrust.com			1

Client Errors Breakdown

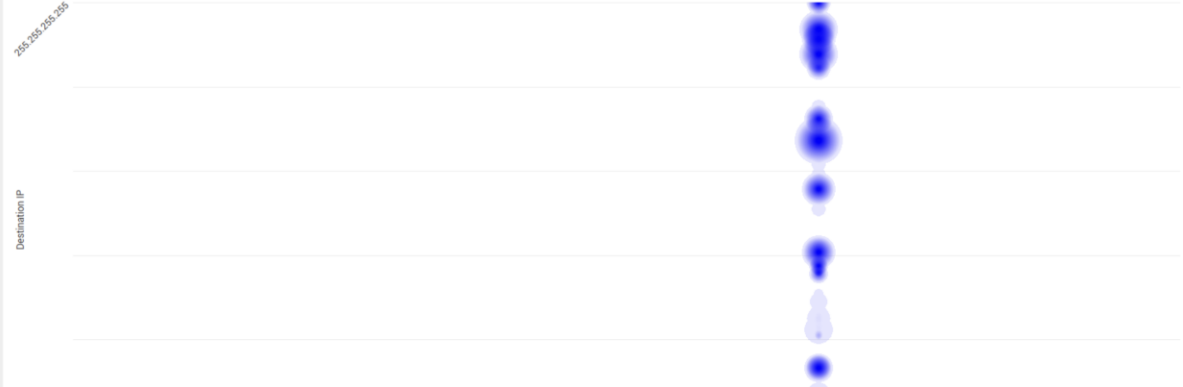
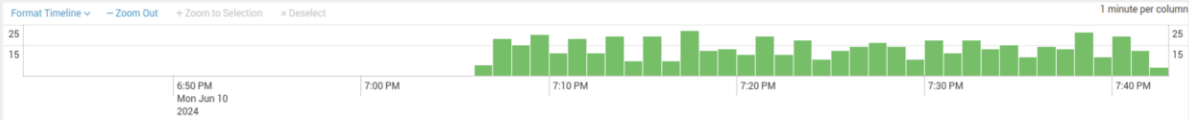


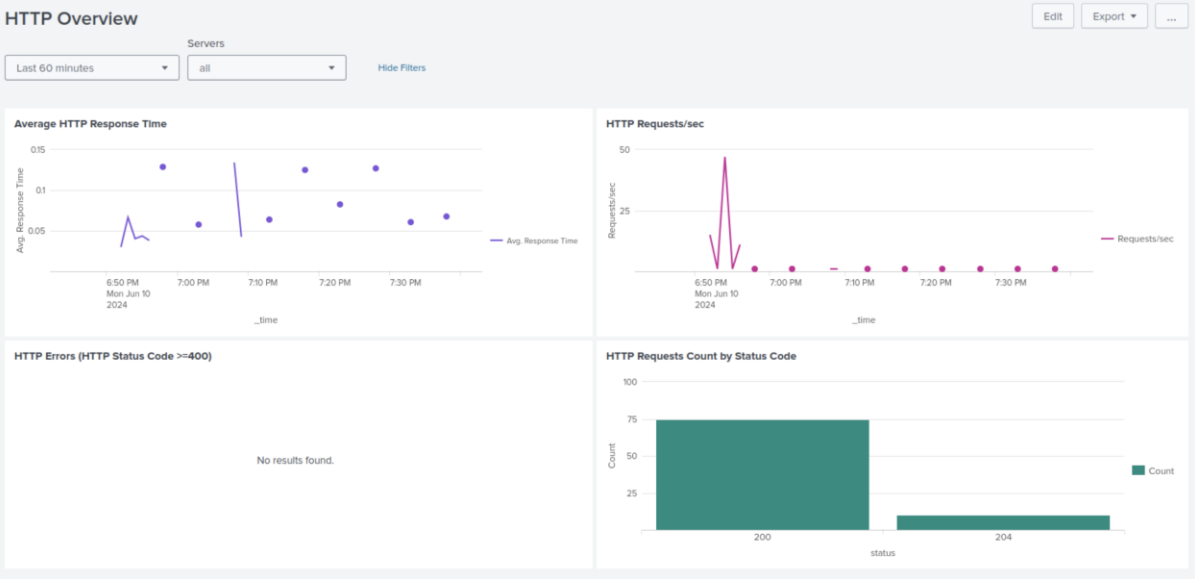
Server Errors Breakdown

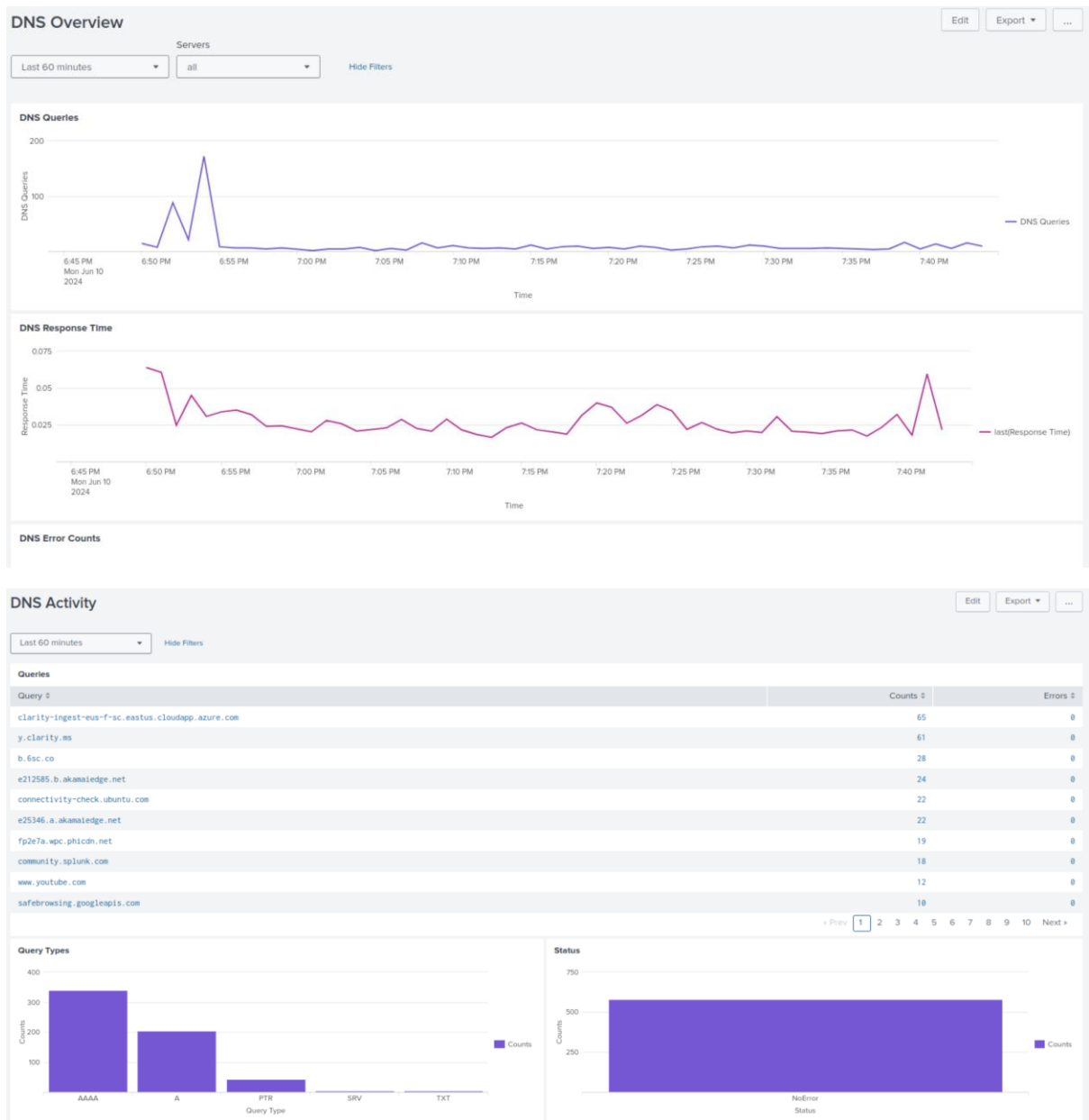


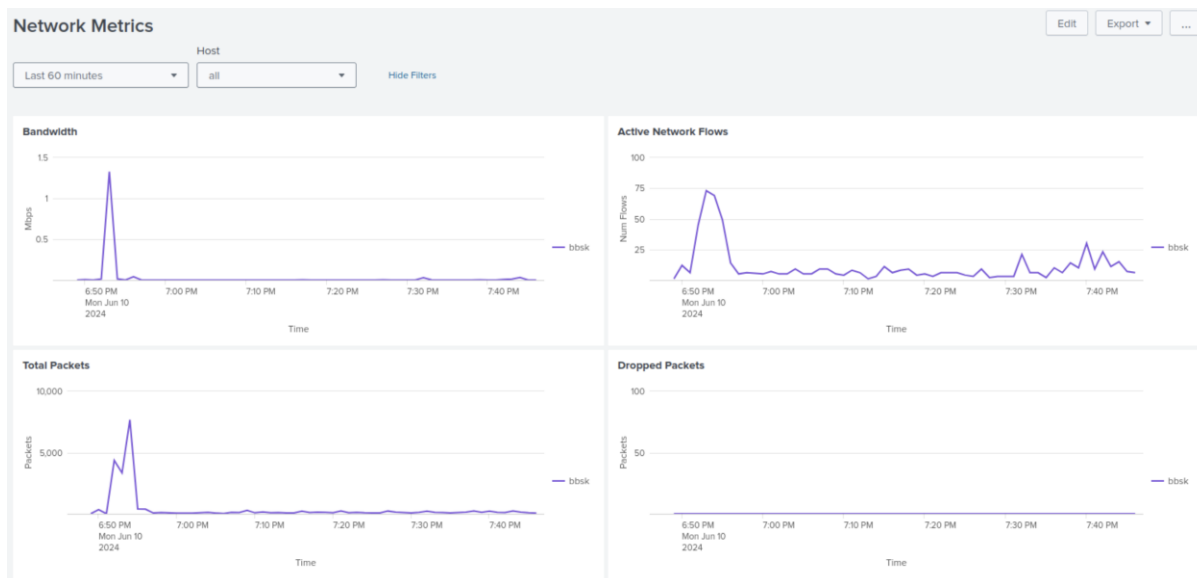
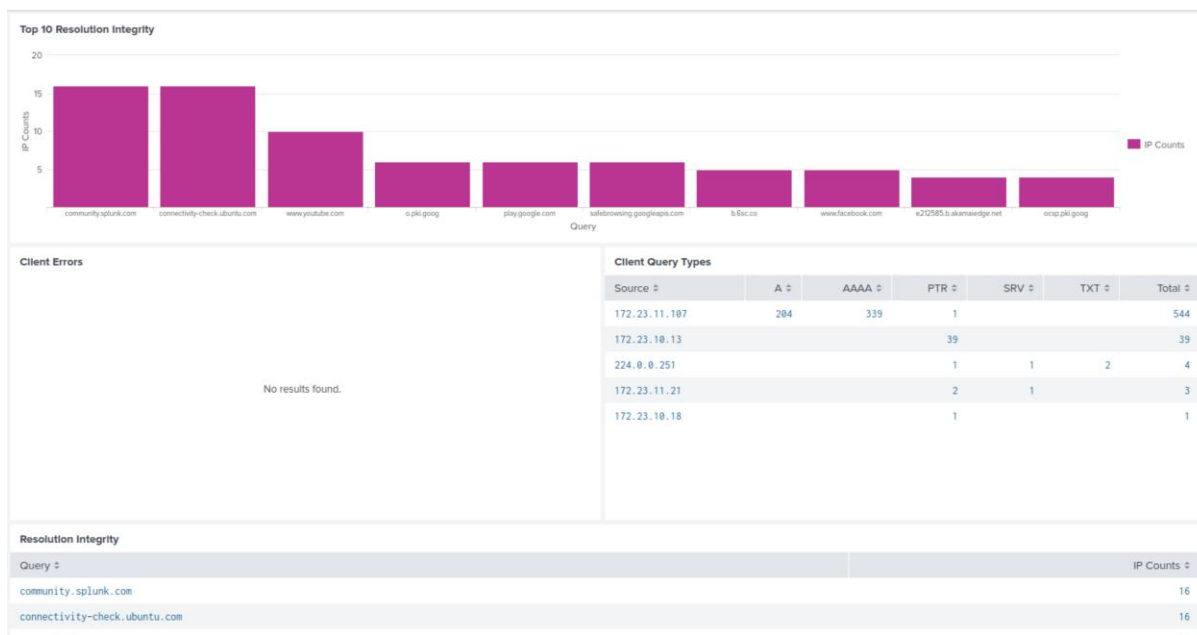
Flow Visualization

Time: 1 hour window Metric: flows

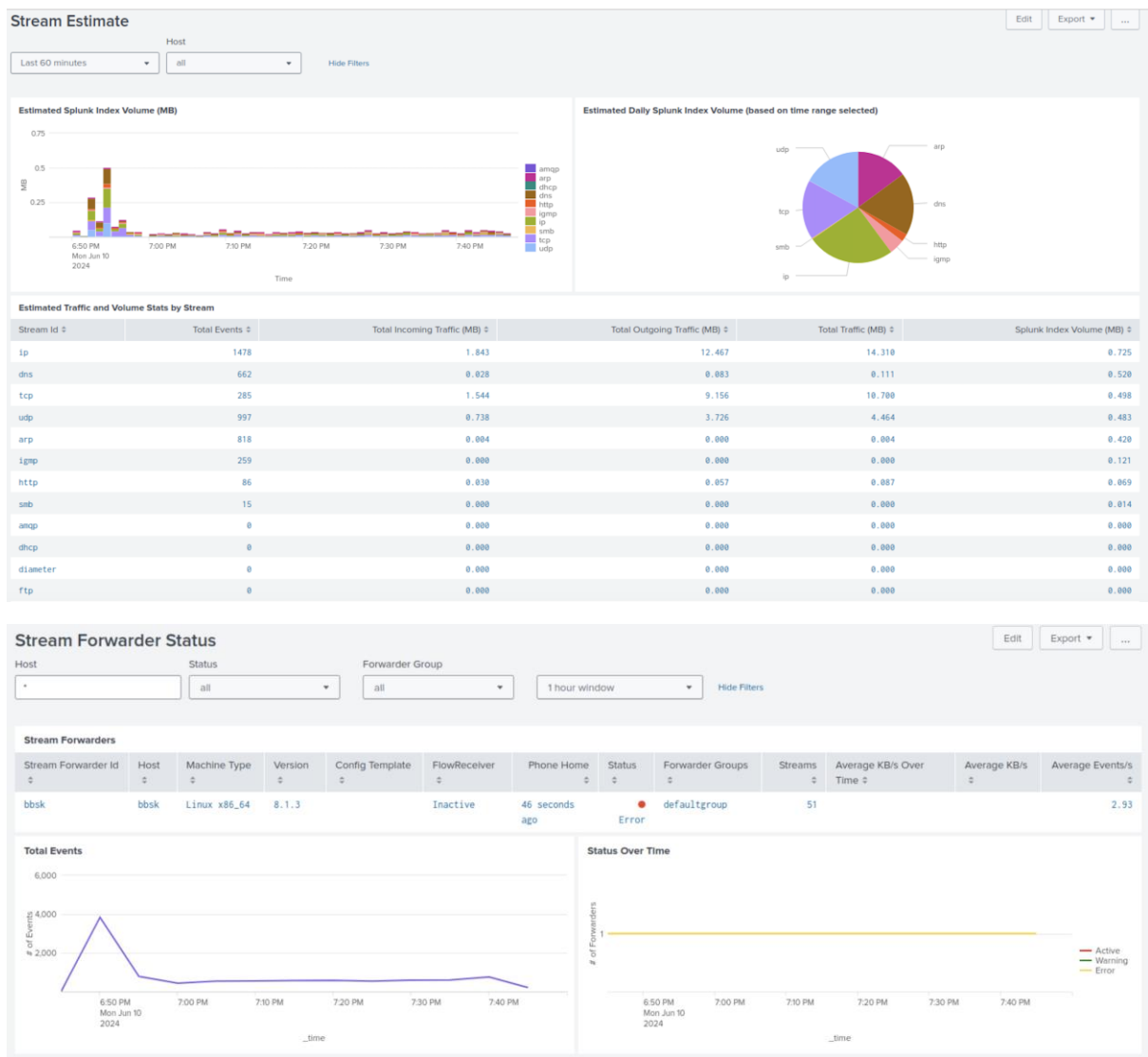








Peaki na początku to aktywne korzystanie z laptopa, dalsza część wykresu reprezentuje głównie bezczynność człowieka, ale i tak dobrze widać, że ruch sieciowy nigdy nie ustaje.



W przypadku wdrożenia wielu Forwarderów mamy dedykowany panel do monitorowania ich aktywności i stanu. Niestety jak widać powyżej status Error jest najczęstszy nawet mimo poprawnego funkcjonowania – jest to kwestia zapytania przekładającego się na rysowanie wykresu i tabeli.

Podsumowanie

Splunk Enterprise to system big-data o niezliczonej ilości funkcjonalności, na co się składają setki wiarygodnych aplikacji i add-onów do doinstalowania rozszerzającego jego możliwości. Popularny jest również Splunk Enterprise Security, który z systemu klasy big-data robi typowy SIEM. Mając zebrane odpowiednie logi, wyłącznie kreatywność i potrzeba analityków ogranicza możliwości ich wykorzystania. Logi zebrane w ramach naszego WIDS bardzo łatwo korelować z innymi danymi tworząc tym samym szeregi powiązań pomiędzy generowanymi zdarzeniami i lepiej rozumieć naszą siećową topologię co z automatu przekłada się na możliwość wdrożenia wyższych standardów jej zabezpieczenia.

Kismet

Wprowadzenie teoretyczne

Kismet to zaawansowane narzędzie do monitorowania sieci bezprzewodowej, które jest szeroko stosowane w celach zapewnienia odpowiedniego bezpieczeństwa sieci bezprzewodowych. Narzędzie to działa jako sniffer sieciowy, a często również poprzez odpowiednią konfigurację działać może jako WIDS (ang. *Wireless Intrusion Detection System*). Jest to przydatne narzędzie do badania bezprzewodowych fal radiowych wokół nas, celem znalezienia docelowych bezprzewodowych sieci LAN do złamania.

Główne funkcje Kismet:

1. Monitorowanie i sniffer Wi-Fi

Kismet może przechwytywać pakiety z sieci bezprzewodowych, analizować je i zapisywać do plików. Pakietu te można później przeglądać i analizować za pomocą innych narzędzi, chociażby takich jak Wireshark.

2. Wykrywanie sieci

Narzędzie automatycznie wykrywa wszystkie dostępne sieci Wi-Fi w zasięgu, w tym te ukryte (nierozgłaszające swojego SSID, czyli unikatowej nazwy sieci).

3. Wykrywanie intruzów

Kismet monitoruje ruch sieciowy w celu wykrywania podejrzanej aktywności, takiej jak ataki: typu deauthentication, spoofing, czy próby włamania. Wykorzystuje w tym celu specjalnie skonfigurowane alerty.

4. Obsługa wielu rodzajów sieci

Oprócz sieci Wi-Fi, Kismet obsługuje również sieci Bluetooth, Zigbee i inne typy bezprzewodowych technologii.

5. Integracja z innymi narzędziami

Dzięki REST API, Kismet może być zintegrowany z innymi narzędziami do analizy i monitorowania, takimi jak Elasticsearch i Grafana, co pozwala na bardziej zaawansowane wizualizacje i analizy danych w czasie rzeczywistym. Ponadto umożliwia to zbieranie logów systemowych, generowanych przez poszczególne urządzenia sieciowe działające w ramach jednej WLAN.

Cele stosowania Kismet

Kismet jest preferowanym narzędziem wśród specjalistów ds. bezpieczeństwa sieci ze względu na swoją wszechstronność, otwartość i zaawansowane funkcje. Jest idealnym narzędziem do:

- Audytów bezpieczeństwa sieci bezprzewodowych.
- Przeprowadzania testów penetracyjnych oraz testowania zabezpieczeń sieci.
- Monitorowania sieci w celu wykrywania anomalii i nieautoryzowanej aktywności.
- Edukacji i badań nad bezpieczeństwem sieci bezprzewodowych.

Dzięki swojej otwartości i elastyczności, Kismet może być dostosowany do różnych scenariuszy i potrzeb użytkowników, co czyni go ważnym narzędziem w arsenale każdego specjalisty ds. bezpieczeństwa sieci.

Alerty w Kismet

Kismet wykorzystuje alerty do komunikowania zdarzeń włamań do sieci bezprzewodowej i krytycznych zdarzeń, które mają miejsce w systemie.

Wyróżniamy kilka poziomów alertów:

0 – INFO – alerty informacyjne, takie jak: datasources errors, Kismet state changes itp.

5 – LOW – zdarzenia o niskim poziomie ryzyka takie jak probe fingerprints

10 – MEDIUM – zdarzenia o średnim poziomie ryzyka, mowa tu między innymi o ataku typu denial of service

15 – HIGH – zdarzenie o wysokim poziomie ryzyka, np. fingerprinted watched devices, denial of service attacks

20 – CRITICAL – błędy krytyczne, takie jak fingerprinted known exploits

Typy alertów w Kismet

DENIAL – prawdopodobnie ataki typu denial of service (DoS, DDoS)

EXPLOIT – Znana próba wykorzystania luki w zabezpieczeniach za pomocą fingerprinted exploit

OTHER – Ogólna kategoria alertów, które nie mieszczą się w istniejącym zasobniku

PROBE

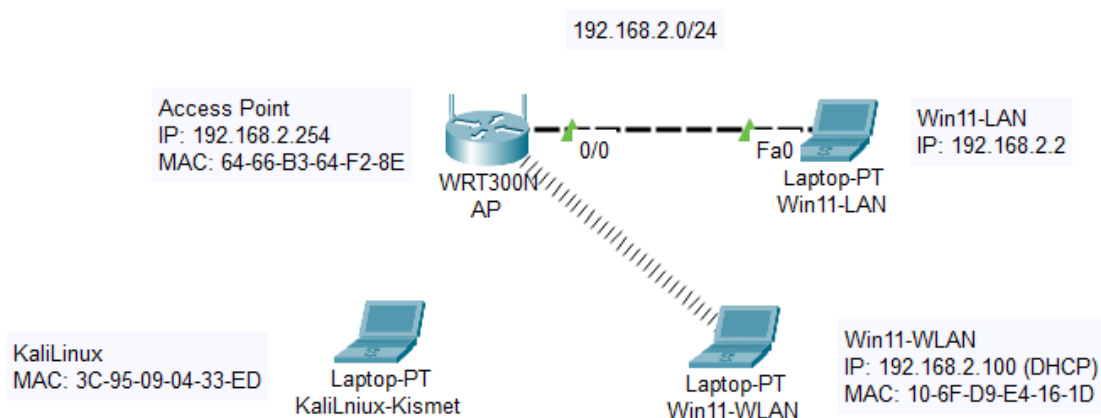
SPOOF – Próba podszycia się pod istniejące urządzenie

SYSTEM – Zdarzenia systemowe, takie jak zmiany w logach, błędy źródła danych itp.

Część praktyczna

Konfiguracja sieci WLAN, czyli konfiguracja Access Pointa

Rysunek przedstawiający konfigurację, wraz z potrzebnymi adresami IP i adresami MAC poszczególnych urządzeń w sieci.



Win11-WLAN – klient końcowy.

Win11-LAN – klient konfigurujący AP.

KaliLinux-Kismet – urządzenie odpowiedzialne za przeglądanie sieci, z tego urządzenia również przeprowadzane są trzy testowe ataki.

AP – Access Point

1. Najpierw należy fizycznie podłączyć się rutera, za pomocą kabla Ethernetowego. Ruter oczywiście podłączamy do zasilania.

2. Kolejnym krokiem jest uruchomienie przeglądarki, wpisujemy adres `192.168.0.1` (domyślny adres ustawiony na AP). Naszym oczom ukazuje się adres logowania, wprowadzamy domyślny login i hasło, dwa razy *admin*.

3. Przechodzimy do konfiguracji sieci. Wybieramy zakładkę *Wireless->Wireless Settings*.

Wireless Settings

Wireless Network Name: (Also called the SSID)

Region:

Warning: Ensure you select a correct country to conform local law. Incorrect settings may cause interference.

Channel:

Mode:

Channel Width:

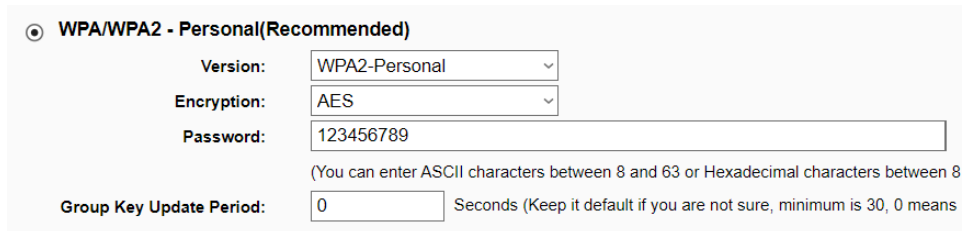
☒ Enable Wireless Radio

☐ Enable SSID Broadcast

☐ Enable WDS Bridging

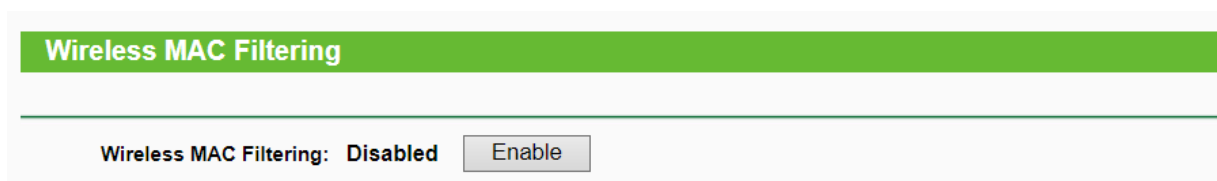
Ważne: po zakończonej konfiguracji musimy kliknąć opcję **Save**, aby wprowadzone przez nas zmiany zostały zapisane.

4. Dalej zakładka *Wireless->Wireless Security*.



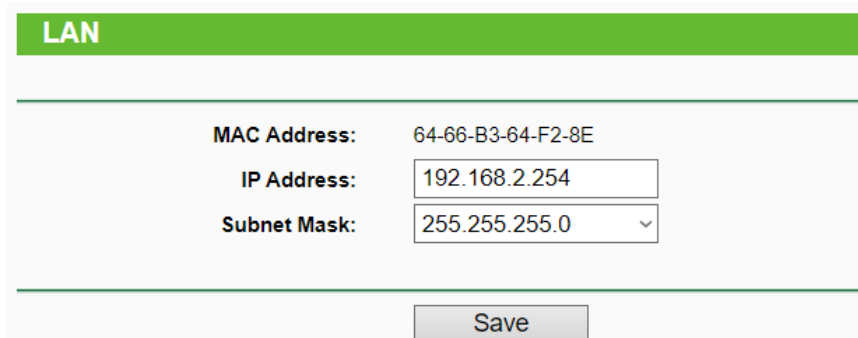
The image shows the 'Wireless Security' configuration page. It features a radio button selection for 'WPA/WPA2 - Personal(Recommended)'. Below this, there are three fields: 'Version' set to 'WPA2-Personal', 'Encryption' set to 'AES', and 'Password' set to '123456789'. A note below the password field states: '(You can enter ASCII characters between 8 and 63 or Hexadecimal characters between 8 and 31)'. At the bottom, there is a 'Group Key Update Period' field set to '0' with a label 'Seconds (Keep it default if you are not sure, minimum is 30, 0 means disabled)'.

5. W zakładce *Wireless->Wireless MAC Filtering* opcja *Wireless MAC Filtering* musi być ustawiona na *Disabled*.



The image shows the 'Wireless MAC Filtering' configuration page. It has a green header with the title 'Wireless MAC Filtering'. Below the header, there is a section with the text 'Wireless MAC Filtering: Disabled' and an 'Enable' button.

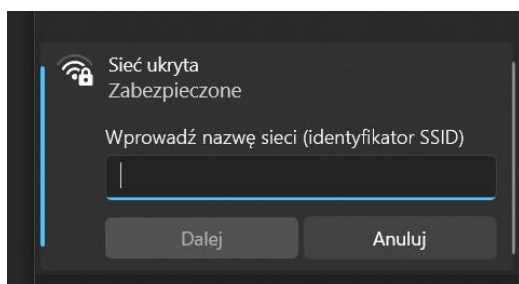
6. W konfiguracji AP ustawiamy statyczny adres 192.168.2.254 z maską /24.



The image shows the 'LAN' configuration page. It has a green header with the title 'LAN'. Below the header, there are three fields: 'MAC Address' set to '64-66-B3-64-F2-8E', 'IP Address' set to '192.168.2.254', and 'Subnet Mask' set to '255.255.255.0'. At the bottom, there is a 'Save' button.

Tyle ustawień wystarczy, możemy zatwierdzić zmiany i wykonać *Reboot*.

7. Łączymy się z siecią, z racji, że nasza sieć jest ukryta musimy wybrać opcję *Sieć ukryta*. Wprowadzamy SSID: lab1 oraz hasło: 123456789



The image shows a Windows network connection dialog box. It has a title bar with a Wi-Fi icon and the text 'Sieć ukryta Zabezpieczone'. Below the title bar, there is a text field with the placeholder 'Wprowadź nazwę sieci (identyfikator SSID)'. At the bottom, there are two buttons: 'Dalej' and 'Anuluj'.

Został przydzielony adres naszej stacji roboczej (Windows 11)

```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix  . : 
Link-local IPv6 Address . . . . . : fe80::d30b:4ada:aff9:19f4%16
IPv4 Address. . . . . : 192.168.2.100
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.2.254
```

8. Za pomocą polecenia ping sprawdzamy połączenie z AP.

```
C:\Windows\System32>ping 192.168.2.254

Pinging 192.168.2.254 with 32 bytes of data:
Reply from 192.168.2.254: bytes=32 time=8ms TTL=64
Reply from 192.168.2.254: bytes=32 time=5ms TTL=64
Reply from 192.168.2.254: bytes=32 time=5ms TTL=64
Reply from 192.168.2.254: bytes=32 time=3ms TTL=64

Ping statistics for 192.168.2.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 8ms, Average = 5ms

C:\Windows\System32>
```

Działa, możemy przejść do konfiguracji stacji Kali Linux, oraz konfiguracji narzędzia Kismet.

Konfiguracja Kismet

1. Najpierw musimy zainstalować narzędzie Kismet, działamy na oprogramowaniu Kali Linux. Robimy to za pomocą komendy *sudo apt-get install kismet*.

2. Następnie konfigurujemy interfejs bezprzewodowy, za pomocą komend przedstawionych na poniższym screenie.

```
(root@kali)~[/home/kali]
# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:"UPC0399566"
Mode:Managed  Frequency:2.412 GHz  Access Point: 38:43:7D:A4:82:CC
Bit Rate=72.2 Mb/s   Tx-Power=20 dBm
Retry short limit:7   RTS thr=2347 B   Fragment thr:off
Encryption key:off
Power Management:on
Link Quality=50/70   Signal level=-60 dBm
Rx invalid nwid:0   Rx invalid crypt:0   Rx invalid frag:0
Tx excessive retries:0 Invalid misc:5   Missed beacon:0

(root@kali)~[/home/kali]
# ifconfig wlan0 down

(root@kali)~[/home/kali]
# iwconfig wlan0 mode Monitor channel 1

(root@kali)~[/home/kali]
# ifconfig wlan0 up

(root@kali)~[/home/kali]
# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  Mode:Monitor  Frequency:2.412 GHz  Tx-Power=20 dBm
Retry short limit:7   RTS thr=2347 B   Fragment thr:off
Power Management:on
```

3. Sprawdzamy dostępność interfejsów bezprzewodowych za pomocą polecenia: *sudo airmon-ng*.

```
(root@kali)-[/home/kali]
# airmon-ng start wlan0 1

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
643 NetworkManager
758 wpa_supplicant

PHY Interface Driver Chipset
phy0 wlan0 rtl8723be Realtek Semiconductor Co., Ltd. RTL8723BE PCIe Wireless Network Adapter
(mac80211 monitor mode already enabled for [phy0]wlan0 on [phy0]1)

(kali@kali)-[/home/kali]
# iwconfig
lo no wireless extensions.

eth0 no wireless extensions.

wlan0 IEEE 802.11 ESSID:off/any
Mode:Managed Frequency:2.412 GHz Access Point: Not-Associated
Tx-Power=20 dBm
Retry short limit:7 RTS thr=2347 B Fragment thr:off
Encryption key:off
Power Management:on

wlan0mon IEEE 802.11 Mode:Monitor Frequency:2.412 GHz Tx-Power=20 dBm
Retry short limit:7 RTS thr=2347 B Fragment thr:off
Power Management:on
```

Udało się nam utworzyć interfejs bezprzewodowy w trybie Monitor, pozwoli nam to dalej na śledzenie i skanowanie sieci bezprzewodowej (WLAN).

UWAGA! Aby z powodzeniem utworzyć interfejs sieciowy w naszym przypadku konieczne było uruchomienie narzędzia kismet na tym etapie.

5. Dalej przechodzimy już do konfiguracji Kismet. Otwieranie konfiguracji: *sudo vim /etc/kismet/kismet.conf* Ustawiamy: *source=wlan0mon*

```
(kali@kali)-[/etc/kismet]
$ cat kismet.conf
# Kismet config file

enablert=YES

# This master config file loads the other configuration files; to learn more about
# the Kismet configuration options, see the comments in the config files, and
# the documentation at:
# https://www.kismetwireless.net/docs/readme/config_files/
#
# You can edit the values in these files, but to make it much easier to update Kismet
# in the future, you should put your changes in the kismet_site.conf override file.
# You can learn more about the override config at:
# https://www.kismetwireless.net/docs/readme/config_files/#configuration-override-files—kismet_siteconf
```

Ponadto musimy aktywować plik z alertami za pomocą wpisu: *enablert=YES*.

6. Konfigurowanie alertów w Kismet, w tym celu zmodyfikować należy plik *kismet_alerts.conf*. Oto przykładowe alerty, które wprowadziłem:

```
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
```

Powyższa konfiguracja alertów, oznacza, że dla danego alertu dopuszczalna ich liczba na sekundę to 1, a na minutę to 5.

W powyższym przykładzie skonfigurowano kilka alertów, które wykrywają różne rodzaje zdarzeń w sieci bezprzewodowej:

- cryptojam: Wykrywa zakłócenia w szyfrowaniu;
- dullsec: Wykrywa nieudane próby połączeń;
- probe: Wykrywa próby połączeń z różnymi sieciami;
- broadcastdiscon: Wykrywa rozgłoszeniowe pakiety deauthentication;
- netstumbler: Wykrywa aktywność narzędzia NetStumbler;
- ssidthresh: Wykrywa zmiany w identyfikatorach BSSID;
- beaconrate: Wykrywa nieprawidłowe tempo ramek beacon.

Trzy ostatnie alerty umieszczone są domyślnie w pliku konfiguracyjnym.

W pliku znajduje się więcej domyślnych alertów. Poniżej prezentuje próbkę tego co domyślnie znajduje się w pliku *kismet_alerts.conf*.

```
# Deprecated
alert=LUCENTTEST,0/min,0/sec
alert=LONGSSID,5/min,1/sec
alert=MSFBCOMSSID,5/min,1/sec
alert=MSFDLINKRATE,5/min,1/sec
alert=MSFNETGEARBEACON,5/min,1/sec
alert=MALFORMMGMT,5/min,1/sec
# Deprecated
alert=NETSTUMBLER,5/min,1/sec
alert=NOCLIENTMFP,10/min,1/sec
alert=NONCEDEGRADE,0/min,0/sec
alert=NONCEREUSE,0/min,0/sec
alert=NULLPROBERESP,5/min,1/sec
alert=OVERPOWERED,0/min,0/sec
alert=PROBECHAN,5/min,1/sec
alert=QCOMEXTENDED,0/min,0/sec
alert=RSNLOOP,5/min,1/sec
alert=RTL8195VD1406,5/min,1/sec
alert=RTLWIFIP2P,5/min,1/sec
alert=VD00202027301,5/min,1/sec
alert=VD00202027302,5/min,1/sec
alert=WPSBRUTE,5/min,1/sec
alert=WMMOVERFLOW,10/min,1/sec
alert=WMMTSPEC,10/min,1/sec
```

Niektóre alerty można wyłączyć poprzez ustawienie poszczególnych wartości na 0.

7. Monitorowanie alertów, wcześniej należy uruchomić kismet: `sudo kismet`
Zebrane logi można przeanalizować między innymi za pomocą Wiresharka. Logi alertów zawierają szczegółowe informacje o każdym wykrytym zdarzeniu. Te szczegółowo informacje to: czas, typ alertu, identyfikator BSSID, oraz inne istotne dane.

8. Uruchamiamy narzędzie do monitorowania sieci WLAN – KISMET. Wystarczy w terminalu wpisać komendę `kismet`.

```
root@kali: /home/kali
KISMET - Point your browser to http://localhost:2501 (or the address of this system) for the Kismet UI
INFO: Registered PHY handler 'Bluetooth' as ID 3
INFO: Registered PHY handler 'UAV' as ID 4
INFO: Registered PHY handler 'NrfMousejack' as ID 5
INFO: Using default rates of 10/min, 1/sec for alert 'BLEEDINGTOOTH'
INFO: Registered PHY handler 'BTLE' as ID 6
INFO: Registered PHY handler 'METER' as ID 7
INFO: Indexing ADSB ICAO db
INFO: Completed indexing ADSB ICAO db, 325554 lines 6512 indexes
INFO: Registered PHY handler 'ADSB' as ID 8
INFO: Registered PHY handler '802.15.4' as ID 9
INFO: Registered PHY handler 'RADIATION' as ID 10
INFO: (HTTPD) Could not read session data file, skipping loading saved sessions.
INFO: Serving static file content from /usr/share/kismet/httpd/
INFO: Enabling channel hopping by default on sources which support channel control.
INFO: Setting default channel hop rate to 5/sec
INFO: Enabling channel list splitting on sources which share the same list of channels
INFO: Enabling channel list shuffling to optimize overlaps
INFO: Sources will be sorted if they support it
```

Narzędzi Kismet jest dostępne z poziomu przeglądarki, pod adresem <http://localhost:2501>.

Po konfiguracji AP, sieci WLAN oraz narzędzia KISMET możemy przejść do przeprowadzenia kilku ataków.

Testowanie działania alertów, w tym przeprowadzanie ataków

1. Najpierw przeprowadzamy atak *typu deauthentication*.

```
(root@kali)-[/home/kali]
# mdk3 wlan0mon d -c 1
```

2. Próbuje się połączyć z naszą siecią (lab1).

Stacja otrzymuje normalnie adres (DHCP).

```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : 
Link-local IPv6 Address . . . . . : fe80::d30b:4ada:aff9:19f4%14
IPv4 Address. . . . . : 192.168.2.100
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.2.254
```

3. Jednak po chwili traci dostęp do sieci oraz przydzielony adres IP (całą konfigurację).

```
Wireless LAN adapter Wi-Fi:
```

```
Media State . . . . . : Media disconnected  
Connection-specific DNS Suffix  . :
```

Nie możemy dostać się na Access Point.



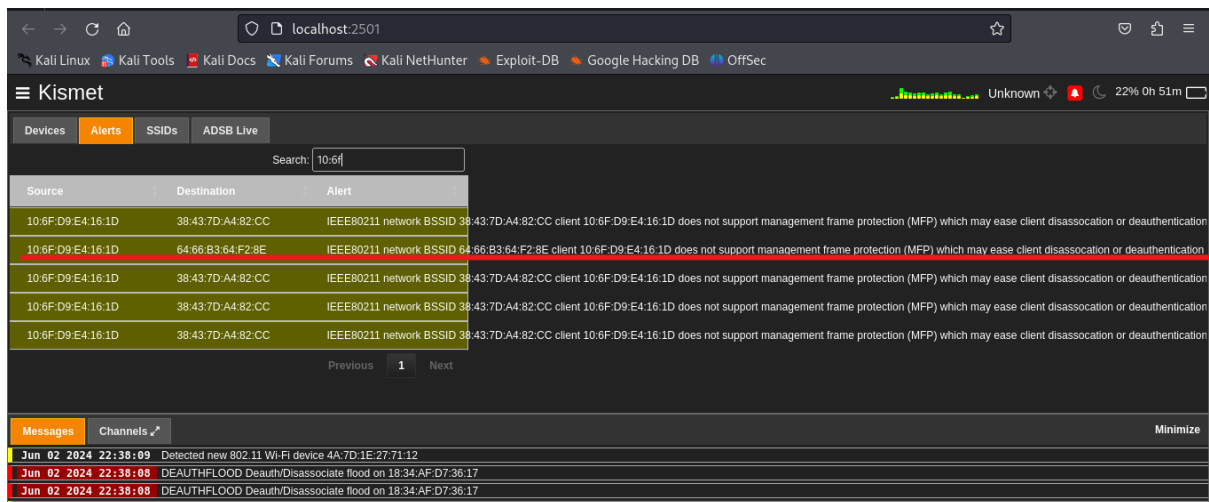
Ta witryna jest nieosiągalna

Strona **http://192.168.2.254/** jest nieosiągalna.

✓ [Uruchom Diagnostykę sieci systemu Windows.](#)

ERR_ADDRESS_UNREACHABLE

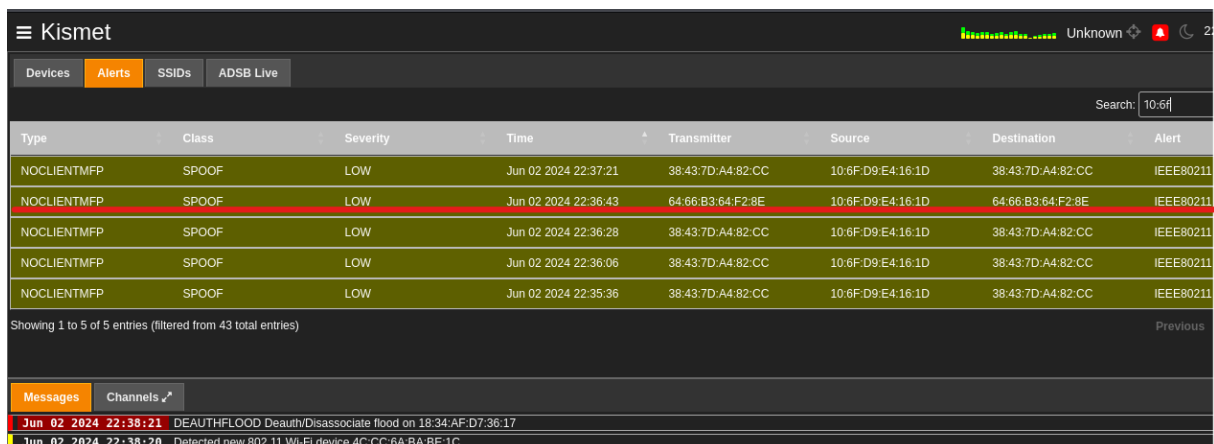
4. Wykryliśmy alert w Kismet.



The screenshot shows the Kismet web interface with the 'Alerts' tab selected. A search filter '10:6f' is applied. The table below shows the alerts:

Source	Destination	Alert
10:6F:D9:E4:16:1D	38:43:7D:A4:82:CC	IEEE80211 network BSSID 38:43:7D:A4:82:CC client 10:6F:D9:E4:16:1D does not support management frame protection (MFP) which may ease client disassociation or deauthentication
10:6F:D9:E4:16:1D	64:66:B3:64:F2:8E	IEEE80211 network BSSID 64:66:B3:64:F2:8E client 10:6F:D9:E4:16:1D does not support management frame protection (MFP) which may ease client disassociation or deauthentication
10:6F:D9:E4:16:1D	38:43:7D:A4:82:CC	IEEE80211 network BSSID 38:43:7D:A4:82:CC client 10:6F:D9:E4:16:1D does not support management frame protection (MFP) which may ease client disassociation or deauthentication
10:6F:D9:E4:16:1D	38:43:7D:A4:82:CC	IEEE80211 network BSSID 38:43:7D:A4:82:CC client 10:6F:D9:E4:16:1D does not support management frame protection (MFP) which may ease client disassociation or deauthentication
10:6F:D9:E4:16:1D	38:43:7D:A4:82:CC	IEEE80211 network BSSID 38:43:7D:A4:82:CC client 10:6F:D9:E4:16:1D does not support management frame protection (MFP) which may ease client disassociation or deauthentication

Jest to alert typu: NOCLIENTMFP. Komunikat brzmi, że nasz AP o BSSID równym 64-66-B3-64-F2-8E nie wspiera MFP, czyli standardu ochrony klientów i punktów dostępu przed sfałszowanymi ramkami zarządzania, dla klienta o adresie MAC (10-6F-D9-E4-16-1D, czyli nasza stacja Win11). Zwiększa to zatem prawdopodobieństwo przeprowadzenia skutecznego ataku deauthentication lub ataku DoS (Denial of Service).



The screenshot shows the Kismet web interface with the 'Alerts' tab selected. A search filter '10:6f' is applied. The table below shows the alerts:

Type	Class	Severity	Time	Transmitter	Source	Destination	Alert
NOCLIENTMFP	SPOOF	LOW	Jun 02 2024 22:37:21	38:43:7D:A4:82:CC	10:6F:D9:E4:16:1D	38:43:7D:A4:82:CC	IEEE80211
NOCLIENTMFP	SPOOF	LOW	Jun 02 2024 22:36:43	64:66:B3:64:F2:8E	10:6F:D9:E4:16:1D	64:66:B3:64:F2:8E	IEEE80211
NOCLIENTMFP	SPOOF	LOW	Jun 02 2024 22:36:28	38:43:7D:A4:82:CC	10:6F:D9:E4:16:1D	38:43:7D:A4:82:CC	IEEE80211
NOCLIENTMFP	SPOOF	LOW	Jun 02 2024 22:36:06	38:43:7D:A4:82:CC	10:6F:D9:E4:16:1D	38:43:7D:A4:82:CC	IEEE80211
NOCLIENTMFP	SPOOF	LOW	Jun 02 2024 22:35:36	38:43:7D:A4:82:CC	10:6F:D9:E4:16:1D	38:43:7D:A4:82:CC	IEEE80211

Showing 1 to 5 of 5 entries (filtered from 43 total entries)

Kolejny przechwycony, przy pomocy Kismet, alert jest typu: DEAUTHFLOOD / BCASTDISCON. Alert ten świadczy o przeprowadzania ataku deauthentication. Na atak ten są podatne w szczególności urządzenia, które nie posiadają wsparcia dla standardu MFP.

Zwykle niewielkie ilości pakietów cofających uwierzytelnienie lub odłączających są normalną częścią Wi-Fi, ale wiele narzędzi typu „odmowa usługi” wysyła zalew tych pakietów, aby uniemożliwić klientowi ponowne nawiązanie połączenia.

Kismet

Unknown

Charging 10%

Devices

Alerts

SSIDs

ADSB Live

Search:

Type	Class	Severity	Time	Transmitter	Source	Destination	Alert
DEAUTHFLOOD	DENIAL	MEDIUM	Jun 02 2024 19:43:16	34:2C:C4:B5:3F:E3	01:00:5E:00:00:FB	34:2C:C4:B5:3F:E3	Deauth/Disassociate flood on
BCASTDISCON	DENIAL	MEDIUM	Jun 02 2024 19:43:20	E8:D2:FF:5E:29:B4	E8:D2:FF:5E:29:B4	all	IEEE80211 Access Point BSS
BCASTDISCON	DENIAL	MEDIUM	Jun 02 2024 19:43:20	E8:D2:FF:5E:29:B4	E8:D2:FF:5E:29:B4	all	IEEE80211 Access Point BSS
DEAUTHFLOOD	DENIAL	MEDIUM	Jun 02 2024 19:43:20	E8:D2:FF:5E:29:B4	E8:D2:FF:5E:29:B4	all	Deauth/Disassociate flood on
BCASTDISCON	DENIAL	MEDIUM	Jun 02 2024 19:43:25	18:34:AF:A6:12:DE	18:34:AF:A6:12:DE	all	IEEE80211 Access Point BSS
BCASTDISCON	DENIAL	MEDIUM	Jun 02 2024 19:43:25	18:34:AF:A6:12:DE	18:34:AF:A6:12:DE	all	IEEE80211 Access Point BSS

Showing 1 to 6 of 50 entries

Previous

1

2

3

4

5

...

9

Next

Messages

Channels

Minimize

Jun 02 2024 20:04:32	Detected new 802.11 Wi-Fi device A6:F5:1F:C9:4D:CC
Jun 02 2024 20:04:30	Detected new 802.11 Wi-Fi device A6:69:DF:D7:29:EF
Jun 02 2024 20:04:28	Detected new 802.11 Wi-Fi device 3A:04:D8:5A:45:D8
Jun 02 2024 20:04:26	Detected new 802.11 Wi-Fi device 78:8B:2A:A6:84:52
Jun 02 2024 20:04:24	Detected new 802.11 Wi-Fi device 12:E1:B9:0F:93:C6
Jun 02 2024 20:04:23	Detected new 802.11 Wi-Fi device 38:68:A4:D0:C9:FC
Jun 02 2024 20:04:19	Detected new 802.11 Wi-Fi device 56:2C:7F:53:A9:58
Jun 02 2024 20:04:13	Detected new 802.11 Wi-Fi device 0F:CC:42:9A:8A:6D

Powered by many OSS components, see the [credits page](#)

5. Kolejnym atakiem jest podszycie się pod Access Point (Adres MAC rutera to: 64:66:B3:64:F2:8E).

```
(root@kali)-[/home/kali]
# mdk3 wlan0mon a -a 64:66:B3:64:F2:8E

AP 64:66:B3:64:F2:8E is responding!

Device is still responding with 65500 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 66000 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 66500 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 67000 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 67500 clients connected!
```

Do rutera przyłączani są fikcyjni klienci. Przez co pamięć RAM rutera jest sukcesywnie zapychana.

W wyniku tego stacji Win11 traci przydzielony adres IP, przez co nie może komunikować się z AP.

```
Wireless LAN adapter Wi-Fi:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
```

Nie można wejść na stronę AP poprzez wpisanie w przeglądarkę jego adresu: 192.168.2.254.



Ta witryna jest nieosiągalna

Strona <http://192.168.2.254/> jest nieosiągalna.

✓ [Uruchom Diagnostykę sieci systemu Windows.](#)

ERR_ADDRESS_UNREACHABLE

Sprawdzamy zatem poleceniem ping połączenie pomiędzy AP a stacją.

Sieć jest nieosiągalna.

```
C:\Windows\System32>ping 192.168.2.254 -t

Pinging 192.168.2.254 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Reply from 192.168.0.59: Destination host unreachable.
Request timed out.
```

```
Reply from 192.168.0.59: Destination host unreachable.
Reply from 192.168.0.59: Destination host unreachable.
Reply from 192.168.0.59: Destination host unreachable.
Reply from 192.168.0.59: Destination host unreachable.
Reply from 192.168.0.59: Destination host unreachable.
Reply from 192.168.0.59: Destination host unreachable.
Reply from 192.168.0.59: Destination host unreachable.
Reply from 192.168.0.59: Destination host unreachable.
Reply from 192.168.0.59: Destination host unreachable.
Reply from 192.168.0.59: Destination host unreachable.
Reply from 192.168.0.59: Destination host unreachable.
```

Bardzo duża ilość przydzielonych klientów spowodowała zaprzestanie pracy AP, z uwagi na zapchanie pamięci RAM punktu dostępowego.

```
Device is still responding with 24500 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 25000 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 25500 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 26000 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 26500 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 27000 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 27500 clients connected!
AP 64:66:B3:64:F2:8E seems to be INVULNERABLE!
Device is still responding with 28000 clients connected!
```

Informacja o przyłączanych klientach, które wykrywa Kismet.

```
INFO: Detected new 802.11 Wi-Fi device D0:16:B4:52:EB:B2
INFO: Detected new 802.11 Wi-Fi device FC:03:9F:E2:94:87
INFO: Detected new 802.11 Wi-Fi device 46:A9:E3:AC:42:39
INFO: Detected new 802.11 Wi-Fi device 20:3D:BD:9C:A1:78
INFO: Detected new 802.11 Wi-Fi device 7E:8E:23:6D:5B:07
INFO: Detected new 802.11 Wi-Fi device 7A:C2:50:09:70:15
INFO: Detected new 802.11 Wi-Fi device 9A:DE:1E:37:2F:BA
INFO: Detected new 802.11 Wi-Fi device EA:C1:AA:8D:AB:DF
INFO: Detected new 802.11 Wi-Fi device F2:CC:C7:3C:60:9D
INFO: Detected new 802.11 Wi-Fi device AE:37:B6:6D:F1:3C
INFO: Detected new 802.11 Wi-Fi device 06:2A:5C:5F:15:ED
INFO: Detected new 802.11 Wi-Fi device 6E:74:E5:18:FC:E1
INFO: Detected new 802.11 Wi-Fi device 72:AB:1E:15:1D:08
INFO: Detected new 802.11 Wi-Fi device EE:EB:85:A2:05:AF
INFO: Detected new 802.11 Wi-Fi device CA:35:6B:61:8C:2D
INFO: Detected new 802.11 Wi-Fi device FE:19:80:74:C7:B8
INFO: Detected new 802.11 Wi-Fi device 16:B6:38:49:0E:F7
INFO: Detected new 802.11 Wi-Fi device 5A:42:C0:F2:50:69
INFO: Detected new 802.11 Wi-Fi device CA:55:72:85:39:B4
INFO: Detected new 802.11 Wi-Fi device 2A:9C:ED:F9:97:39
INFO: Detected new 802.11 Wi-Fi device EE:9A:B4:F5:DA:8F
INFO: Detected new 802.11 Wi-Fi device E6:4D:2F:2B:D5:44
INFO: Detected new 802.11 Wi-Fi device 1A:92:B1:45:7F:41
INFO: Detected new 802.11 Wi-Fi device C8:12:0B:3F:72:B8
INFO: Detected new 802.11 Wi-Fi device 5E:46:A1:74:21:33
INFO: Detected new 802.11 Wi-Fi device F2:B7:43:AE:27:0D
INFO: Detected new 802.11 Wi-Fi device FA:84:ED:FE:5F:36
INFO: Detected new 802.11 Wi-Fi device A6:61:54:54:8E:0D
INFO: Detected new 802.11 Wi-Fi device 6A:E9:3C:61:4B:35
INFO: Detected new 802.11 Wi-Fi device 72:7A:84:61:2D:11
INFO: Detected new 802.11 Wi-Fi device 9E:2B:57:64:26:E5
```

6. Wykryte alerty w Kismet

Przechwycony atak typu: ADVCRYPTCHANGE. Jak możemy przeczytać w opisie tego alertu w dokumentacji, alert ten występuje w przypadku ataku polegającego na podszywaniu się pod Access Point, np. *evil twin attack*, który nie skopiował w pełni atrybutów. Ponadto alert ten może dotyczyć również zwykłej, na pierwszy rzut oka zmiany konfiguracji AP.

Tak było w tym przypadku.

ALERT: ADVCRYPTCHANGE

Alert

Alert ADVCRYPTCHANGE

Class SPOOF

Severity HIGH

Time Jun 07 2024 22:04:01

Alert content IEEE80211 Access Point BSSID 64:66:B3:64:F2:8E SSID "lab1" changed advertised encryption from WPA2 WPA2-PSK AES-CCMP to Open which may indicate AP spoofing/impersonation

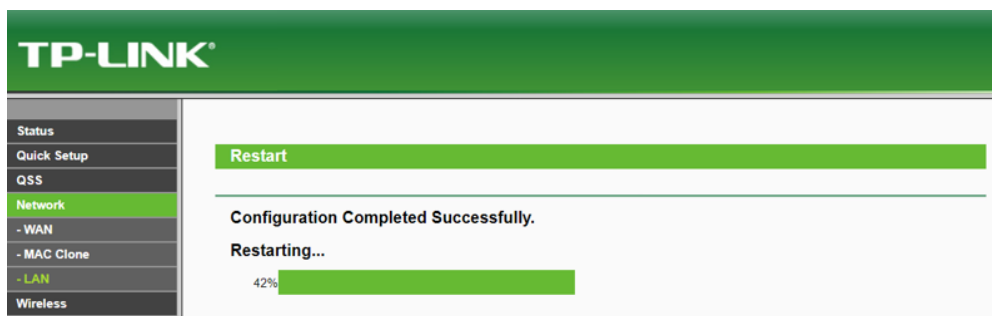
Addresses

Source 64:66:B3:64:F2:8E

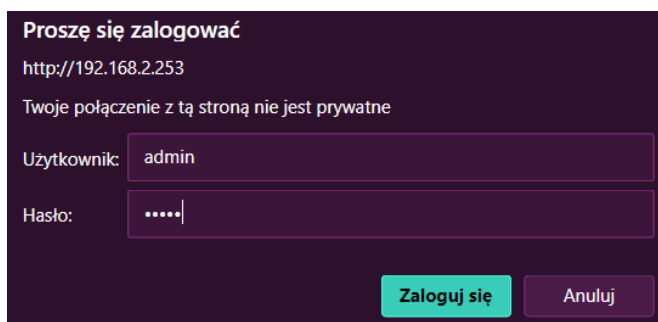
Destination 10:6F:D9:E4:16:1D

Transmitter	Source
64:66:B3:64:F2:8E	64:66:B3:64:F2:8E
64:66:B3:64:F2:8E	64:66:B3:64:F2:8E
64:66:B3:64:F2:8E	64:66:B3:64:F2:8E
44:1C:12:FF:C4:F9	44:1C:12:FF:C4:F9
44:1C:12:FF:C4:F9	44:1C:12:FF:C4:F9
18:34:AF:A6:12:DE	18:34:AF:A6:12:DE

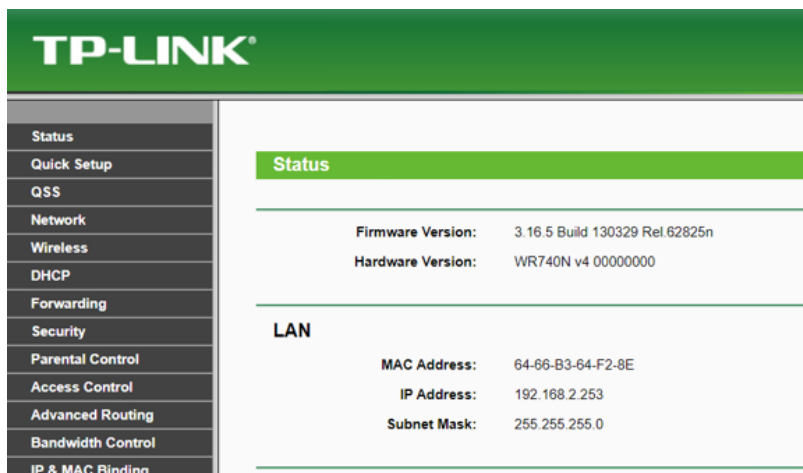
Otóż zmieniliśmy adres IP Access Pointa na 192.168.2.253.



Po odświeżeniu przeglądarki możemy zalogować się ponownie na AP, ze stacji Win11. Stacja jest stale połączona z siecią lab1.

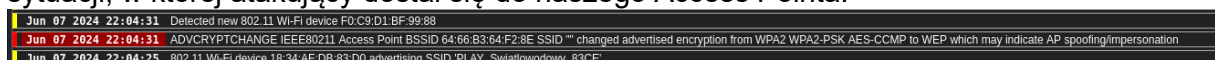


Jak przedstawiono poniżej panel konfiguracyjny jest normlanie dostępny, ale pod innym adresem.



Nasze działania zostały wykryte przez Kismet, czyli mowa tu o zmianie konfiguracji Access Pointa, a konkretnie jego adresu IPv4.

Zmiana konfiguracji może stanowić duże niebezpieczeństwo, gdyż świadczyć może o sytuacji, w której atakujący dostał się do naszego Access Pointa.



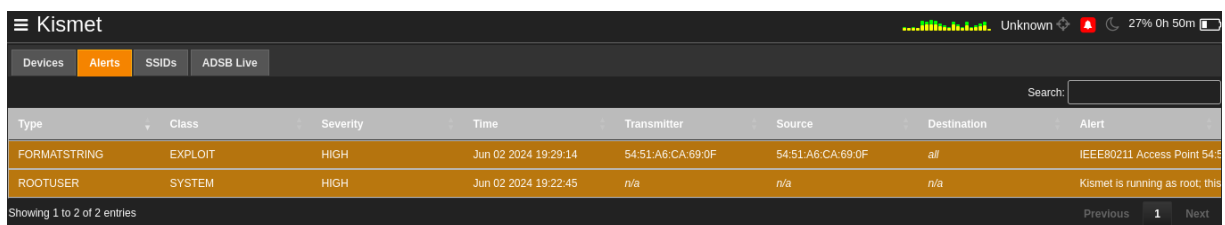
```
ALERT: ADVCRYPTCHANGE IEEE80211 Access Point BSSID 64:66:B3:64:F2:8E SSID
"" changed advertised encryption from WPA2 WPA2-PSK AES-CCMP to WEP
which may indicate AP spoofing/impersonation
INFO: Detected new 802.11 Wi-Fi device F0:C9:D1:BF:99:88
INFO: Detected new 802.11 Wi-Fi device DA:C5:0D:2A:00:B2
ALERT: ADVCRYPTCHANGE IEEE80211 Access Point BSSID 64:66:B3:64:F2:8E SSID
"" changed advertised encryption from WEP to WPA2 WPA2-PSK AES-CCMP
```

7. Beacon Flooding Attack

Uruchomienie ataku.

```
(root@kali)-[/home/kali]
# mdk3 wlan0mon b -c 1 -s 10000
```

Informacje wykryte w Kismet

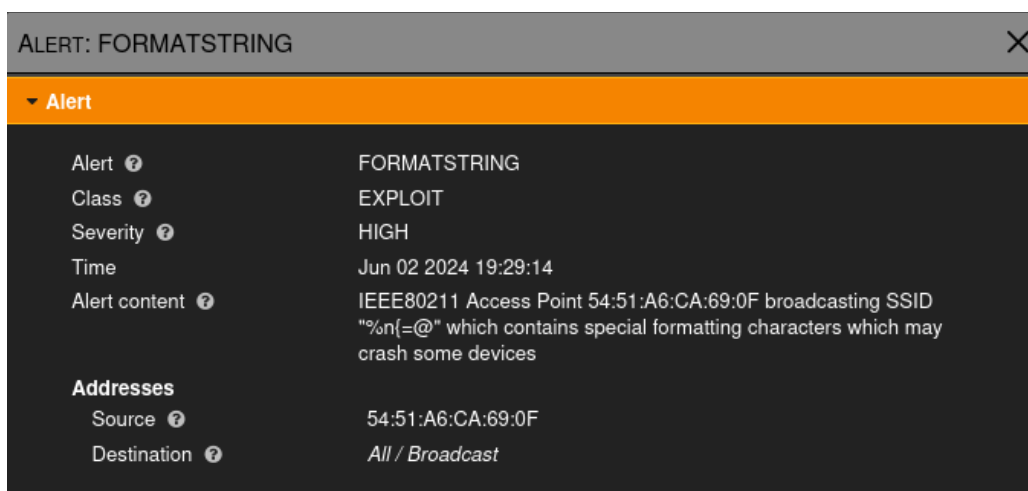


The screenshot shows the Kismet application window. At the top, there's a status bar with 'Unknown' and a battery icon. Below it, a tabbed interface has 'Alerts' selected. A table displays two alerts:

Type	Class	Severity	Time	Transmitter	Source	Destination	Alert
FORMATSTRING	EXPLOIT	HIGH	Jun 02 2024 19:29:14	54:51:A6:CA:69:0F	54:51:A6:CA:69:0F	all	IEEE80211 Access Point 54:51:A6:CA:69:0F broadcasting SSID "" which contains special formatting characters which may crash some devices
ROOTUSER	SYSTEM	HIGH	Jun 02 2024 19:22:45	n/a	n/a	n/a	Kismet is running as root; this

At the bottom, it says 'Showing 1 to 2 of 2 entries' with 'Previous' and 'Next' buttons.

Szczegółowe informacje potwierdzają wykrycie alertu dotyczącego przeprowadzania *Beacon Flooding Attack*. Standard 802.11 nie nakłada bowiem szczególnych ograniczeń związanych z nazwą sieci (SSID), jednak wykorzystywanie znaków specjalnych do nazywania sieci WLAN może budzić podejrzenia i powinno być alarmowane, co też ma miejsce.



The screenshot shows a detailed view of an alert titled 'ALERT: FORMATSTRING'. It has a close button (X) in the top right corner. Below the title is a section labeled 'Alert' with a dropdown arrow. The details are as follows:

Alert ?	FORMATSTRING
Class ?	EXPLOIT
Severity ?	HIGH
Time	Jun 02 2024 19:29:14
Alert content ?	IEEE80211 Access Point 54:51:A6:CA:69:0F broadcasting SSID "" which contains special formatting characters which may crash some devices
Addresses	
Source ?	54:51:A6:CA:69:0F
Destination ?	All / Broadcast

W Kismet mamy również informacje o utworzonych nowych sieciach z dziwnymi i podejrzanymi SSID. Nazwy tych sieci jednoznacznie wskazują na przeprowadzenie *Beacon*.

Flooding Attack.

Jun 07 2024 21:52:17	Detected new 802.11 Wi-Fi access point 8E:03:39:C4:28:8F
Jun 07 2024 21:52:16	802.11 Wi-Fi device F6:C0:42:F1:E6:09 advertising SSID 'v6iugv%TqX1i38>t' E 2Bf _Znu'THL'
Jun 07 2024 21:52:16	Detected new 802.11 Wi-Fi access point F6:C0:42:F1:E6:09
Jun 07 2024 21:52:16	802.11 Wi-Fi device 0C:B4:A3:97:DE:1E advertising SSID ',+J#UM2JO)P-.Q_9[uXkSG.jn6[LM]'

Powered by many OSS components, see the [credits page](#)

Podsumowanie narzędzia Kismet

Narzędzie Kismet, którego używaliśmy podczas naszego projektu świetnie sprawdziło się jako Wireless Intrusion Detection System (WIDS). Samo przeprowadzenie ataków oraz ich wykrycie nie stanowiło większego problemu, ze względu na dosyć intuicyjną konfigurację. Oto podsumowanie głównych zalet i wad Kismet jako WIDS:

Zalety Kismet jako WIDS:

- **Otwartość i dostępność:** Kismet jest narzędziem open-source, co oznacza, że jest dostępne za darmo i można je modyfikować według własnych potrzeb. To sprawia, że jest to ekonomiczne rozwiązanie dla wielu organizacji.
- **Intuicyjna konfiguracja:** Kismet oferuje przyjazny interfejs i dokumentację, co sprawia, że konfiguracja i użytkowanie systemu są stosunkowo proste, nawet dla mniej doświadczonych użytkowników.
- **Szerokie możliwości monitoringu:** Kismet obsługuje wiele protokołów i kanałów bezprzewodowych, co umożliwia szerokie spektrum monitoringu sieci bezprzewodowych, wykrywanie wielu rodzajów ataków oraz zbieranie szczegółowych danych o sieciach.

Wady Kismet jako WIDS:

- **Wysokie wymagania sprzętowe:** Pełne wykorzystanie możliwości Kismet może wymagać zaawansowanego sprzętu, szczególnie przy monitorowaniu dużych lub złożonych sieci bezprzewodowych.
- **Brak wsparcia dla niektórych urządzeń:** Niektóre starsze lub mniej popularne karty sieciowe mogą nie być w pełni kompatybilne z Kismet, co może ograniczyć jego skuteczność w niektórych środowiskach.
- **Brak działania prewencyjnego:** Kismet skupia się głównie na detekcji, a nie na zapobieganiu atakom. W związku z tym, aby w pełni chronić sieć, konieczne jest zastosowanie dodatkowych narzędzi lub systemów, które będą mogły reagować na wykryte zagrożenia.

Podsumowując, Kismet jest potężnym narzędziem WIDS, które oferuje szeroki zakres funkcji detekcji i monitoringu sieci bezprzewodowych. Jego otwartość, intuicyjność i wszechstronność czynią go wartościowym wyborem, mimo pewnych ograniczeń sprzętowych i braku działania prewencyjnego w kontekście wykrywanych ataków czy anomalii.

Podsumowanie i wnioski