

**ZAKŁAD SYSTEMÓW ZŁOŻONYCH**  
Politechnika Rzeszowska

**BEZPIECZEŃSTWO SYSTEMÓW I SIECI KOMPUTEROWYCH**

Laboratorium zdalne

***Implementacja IDS w oparciu o system SNORT***

## **1. Wprowadzenie**

Snort jest darmowym, wydawanym na warunkach wolnej licencji sieciowym systemem wykrywania ataków (ang. Network Intrusion Detection System, NIDS). Oprogramowanie stworzone pierwotnie przez M. Roesch'a w 1998, a następnie rozwijane przez Sourcefire oferuje szeroki wachlarz narzędzi, które pozwalają na wykrycie ataków oraz co istotne w czasie rzeczywistym przeprowadzić analizę ruchu oraz rejestrować pakiety (ang. packet logging) w sieciach opartych o protokoły IP/TCP/UDP/ICMP. Snort posiada spory zakres dostępnych mechanizmów, dzięki czemu może wyszukiwać czy dopasować podejrzane treści, dokonać analizy strumieni pakietów oraz wykryć dużą ilość ataków oraz anomalii występujących w sieci min.: przepełnienie buforów, skanowanie portów.

Przed przystąpieniem do realizacji laboratorium należy poznać kilka podstawowych koncepcji związanych z Snort. Oprogramowanie może być skonfigurowane do jednego z trzech trybów:

- *Sniffer mode* - w którym przechwytuje pakiety spełniające określone kryteria, a następnie wyświetla zebrane pakiety w konsoli w sposób ciągły.
- *Packet Logger mode* - logowanie pakietów na dysk.
- *Network Intrusion Detection System (NIDS) mode* - przeprowadzenie wykrywania i analizy ruchu sieciowego. Tryb ten jest najbardziej złożony i wymaga najwięcej konfiguracji.

### **1.1. Definiowanie nagłówka reguły**

Podstawa funkcjonowania systemu Snort, dzięki której możliwe jest wykrywanie włamań jest język tworzenie reguł, który pozwala na stworzenie skomplikowanych wytycznych. Na podstawie ich system IDS może decydować czy pewne wydarzenie potraktować jako atak bądź jako zwyczajny proces funkcjonujący w sieci komputerowej.

Poszczególną regułę w Snort można podzielić na dwie logiczne fragmenty: nagłówek oraz opcje reguły. Nagłówek reguły zawiera akcje jakie ma wykonać, sprawdzany protokół oraz źródłowy i docelowy adres IP (oraz opcjonalnie dla protokołów L4 numer portu). Część w nawiasach okrągłych (opcje reguły) zawiera zestaw opcji użytych dla tej reguły, pozwalając na sprawdzenie innych atrybutów pakietu. Sposób zapisywania reguły w Snort:

```
action proto src_ip src_port direction dst_ip dst_port (options)
```

### **Parametr *Action***

Kiedy pakiet zostanie odebrany przez system, źródłowy oraz docelowy adres IP oraz numery portów są sprawdzane z włączonymi regułami (zasadami). Jeśli którakolwiek z nich odpowiada pakietowi, następnie opcje dodatkowe są sprawdzane z pakietem. Jeśli wszystkie porównania zwrócą wynik pozytywny, określona akcja jest wykonywana.

Snort oferuje kilka różnych akcji (ang. action) jakie można wykorzystać podczas tworzenia reguł:

- *alert* - generowanie alarmu wykorzystując wybraną metodę, a następnie loguj pakiet,
- *log* - loguj pakiet,
- *pass* - zignoruj pakiet,
- *activate* - wywołaj alarm i uruchom dynamiczną regułę,
- *dynamic* - pozostaj bezczynny, dopóki nie zostanie aktywowany przez regułę *activate*, a następnie pracuj jako reguła *log*,
- *drop* - blokuj (odrzucaj) i loguj pakiet
- *reject* - blokuj pakiet, loguj go, a następnie wyślij TCP Reset bądź wiadomość *ICMP Port Unreachable* (jeśli protokół to UDP),
- *sdrop* - blokuj pakiet, ale nie loguj go.

### **Parametr *proto***

Kolejne z pól podczas definiowania reguły określa protokół. Na chwilę obecną, Snort analizuje podejrzane zachowanie protokołów TCP, UDP, ICMP oraz IP. Deweloper ma w planach wdrożyć w najbliższej przyszłości obsługę protokołów routingu min. RIP, OSPF, EIGRP, tunelowania czy innych protokołów warstwy sieciowej.

### **Parametr *src\_ip***

System Snort nie posiada mechanizmu, który pozwoliłby na translację nazw, dlatego też podaje się wyłącznie źródłowy adres IP, dodając opcjonalnie maskę sieciową podaną w notacji CIDR. Przykładowo wprowadzając zapis 10.0.30.0/24 sprawdzane będą adresy z zakresu 10.0.30.1 do 10.0.30.255 przez tą regułę. Poza określeniem podsieci, Snort może brać pod uwagę również pewną listę adresów IP, w tym przypadku adresy należy oddzielić

przecinkiem, a całą listę adresów umieszcza się w kwadratowych nawiasach (kolejność wpisów nie ma znaczenia):

```
[192.168.1.1,192.168.1.45,10.1.1.24]
```

Snort pozwala na wykorzystanie zakresu CIDR w liście adresów, oraz na wykorzystaniu logicznego operatora NOT oraz wyrażenia *any* informujący, że pod uwagę przez regułę brany jest dowolny adres IP.

### **Parametr *src\_port***

Numery portów mogą być określone na kilka sposobów, włączając w to określenie *any* (brany pod uwagę jest dowolny nr portu) czy statyczną definicją zakresu portów, które identyfikuje się poprzez operator dwukropka ":"

Definicja	Znaczenie
1:1024	Wszystkie porty z zakresu <1..1024>
:6000	Wszystkie porty mniejsze bądź równe wartości 6000
500:	Wszystkie porty większe bądź równe wartości 500

Również w przypadku definiowania portów, istnieje możliwość wykorzystania operatora logicznego NOT. Przykładowo chcąc wywołać akcję logowania, dla dowolnego ruchu przychodzącego do hostów z zakresu 192.168.1.0/24 i portów NIE mieszczących się w zakresie 6000 do 6010 (włącznie) należy wydać polecenie:

```
log tcp any any -> 192.168.1.0/24 !6000:6010
```

### **Parametr *direction***

Operator *direction* określa orientację bądź kierunek, dla którego ruch będzie sprawdzany. Wprowadzona wartość "->" określa, że należy sprawdzić ruch sieciowy przychodzący z podanego źródłowego adresu w kierunku docelowym. W Snort istnieje również dwukierunkowy operator "<>" informujący, że para adresów IP/portów może być sprawdzana zarówno w kierunku przychodzącym jak i wychodzącym. Należy zwrócić uwagę na fakt, że operator <-został usunięty z obecnej wersji Snort.

## **1.2. Definiowanie opcji reguł**

Dzięki możliwości sprawdzeniu pakietu pod znacznie dokładniejszych zdefiniowanych wytycznych, możliwe jest stworzenie systemu IDS, który z dużą dokładnością będzie wykrywał niebezpieczne działania mające miejsce w sieci. Każda opcja reguły jest oddzielona od siebie znakiem średnika, natomiast znak dwukropka służy do oddzielenia argumentu od nazwy reguły. W wprowadzeniu do laboratorium scharakteryzowane zostały jedynie wykorzystywane w przykładach opcje reguł, dokładniejszą

informację o pozostałych opcjach znajduje się w dokumentacji oprogramowania Snort. Istnieją cztery główne kategorie reguł:

- **general** - Opcje zapewniają informacje na temat reguły, ale nie wpływają w żaden sposób na wykrycie zagrożenia.
- **payload** - Opcje dotyczą wyszukiwania danych wewnątrz pakietu i mogą być ze sobą powiązane.
- **non-payload** - Snort posiada możliwość zbadania wartości znajdujących się w polu nagłówka protokołów IP, ICMP oraz TCP.
- **post-detection** - Opcje powiązane z wyzwalanym działaniem, kiedy pakiet zostanie pomyślnie porównany z regułą.

#### *General Rule Option:*

- Opcja **msg** w laboratorium zostanie wykorzystana do zapisywania wiadomości do reguły, która została wywołana przez pewną czynność w sieci (np. symulacja ataku na sieć), dzięki czemu wpisy w logach są bardziej czytelne. Format `msg:"<message text>";`
- Opcja **gid** (generator id) jest używana do identyfikacji, która część oprogramowania Snort jest odpowiedzialna za wygenerowanie komunikatu. Konkretnie wartości są zdefiniowane w dokumentacji, aby uniknąć potencjalnych konfliktów zaleca się wykorzystać wartość większą niż 1mln, bądź nie używać jej z racji, iż jest to opcjonalne pole. Format `gid:<generator id>;`
- Opcja **sid** jest użyta do unikalnego zidentyfikowania reguły w Snort, dzięki tej informacji zewnętrzne (dodatkowe) narzędzia mogła łatwo ją zidentyfikować. Format `sid:<snort rules id>;` Wartość opcji powinna być jasno określona:
  - <100 zarezerwowana
  - 100-999,9999 zasady wykorzystane w dystrybucji Snort
  - >=1 mln używane jako lokalne reguły

#### *Payload Detection Rule:*

- a) Opcja **content** jest jedną z najważniejszych opcji reguły dostępnych w Snort. Pozwala na sprawdzenie określonej zawartości w ładunku pakietu. Jeżeli dane zawarte w opcji są identyczne jak w pakiecie, następuje dalsze sprawdzanie opcji dla reguły. Dane zawarte w opcji reguły mogą zawierać binarne dane umieszczone pomiędzy znakami kreski poziomej (|) reprezentującej kod binarny (jako postać liczby heksadecymalnej). Dodatkowo możliwe jest wprowadzenie danych w postaci wartości typu *string*. Opcje

posiada kilkanaście dodatkowych podopcji, które pozwalają bardziej szczegółowo określić wytyczne jakie dane mają być wyszukiwane w pakiecie.

*Non-Payload Detection Rule Options:*

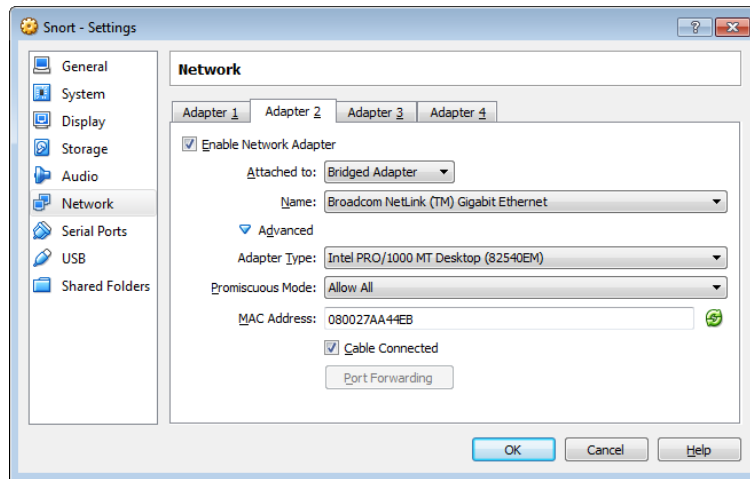
Snort dokonuje sprawdzenia dogłębnego nagłówka protokołów IP, TCP czy ICMP. Dostępna lista opcji jest na tyle obszerna, że zaleca się zapoznanie z dokumentacją systemu Snort. Zgodnie z nią, istnieje 20 różnych opcji zbadania nagłówka wspomnianych protokołów.

Tabela 1 Dostępna lista opcji pozwalających sprawdzić pola w nagłówku protokołów L3 oraz L4.

Keyword	Description
fragoffset	The fragoffset keyword allows one to compare the IP fragment offset field against a decimal value.
ttl	The ttl keyword is used to check the IP time-to-live value.
tos	The tos keyword is used to check the IP TOS field for a specific value.
id	The id keyword is used to check the IP ID field for a specific value.
ipopts	The ipopts keyword is used to check if a specific IP option is present.
fragbits	The fragbits keyword is used to check if fragmentation and reserved bits are set in the IP header.
dsize	The dsize keyword is used to test the packet payload size.
flags	The flags keyword is used to check if specific TCP flag bits are present.
flow	The flow keyword allows rules to only apply to certain directions of the traffic flow.
flowbits	The flowbits keyword allows rules to track states during a transport protocol session.
seq	The seq keyword is used to check for a specific TCP sequence number.
ack	The ack keyword is used to check for a specific TCP acknowledge number.
window	The window keyword is used to check for a specific TCP window size.
itype	The itype keyword is used to check for a specific ICMP type value.
icode	The icode keyword is used to check for a specific ICMP code value.
icmp_id	The icmp_id keyword is used to check for a specific ICMP ID value.
icmp_seq	The icmp_seq keyword is used to check for a specific ICMP sequence value.
rpc	The rpc keyword is used to check for a RPC application, version, and procedure numbers in SUNRPC CALL requests.
ip_proto	The ip_proto keyword allows checks against the IP protocol header.
sameip	The sameip keyword allows rules to check if the source ip is the same as the destination IP.

## 2. Wykorzystana topologia do celów laboratoryjnych

Przeprowadzenie laboratorium wymaga uzyskania połączenia z siecią globalną w celu pobrania i zainstalowania niezbędnego oprogramowania. W związku z tym, dla maszyny wirtualnej, wskazanej przez prowadzącego należy upewnić się, że istnieją skonfigurowane dwie karty sieciowe. Należy sprawdzić ich konfigurację – pierwsza z kart skonfigurowana jest w trybie NAT, natomiast druga – zbridgowana z interfejsem Gigabit Ethernet hosta macierzystego. Konfiguracja drugiej karty przedstawiona jest poniżej:



Po uruchomieniu maszyny wirtualnej, należy zalogować się do niej wprowadzając jako login *root* oraz hasło *P@ssw0rd*. Interfejs *eth0* po uruchomieniu powinien być domyślnie włączony, oraz posiadać adresację z podsieci 10.0.2.0/24. Interfejs *eth1*, który będzie wykorzystywany do sprawdzania ataków na sieć powinien być domyślnie, po uruchomieniu maszyny wyłączony. Należy ten interfejs włączyć, oraz wprowadzić odpowiednią adresację wykorzystując wskazany przez prowadzącego adres IP z podsieci 192.168.5.0/24; przykładowa komenda konfiguracyjna:

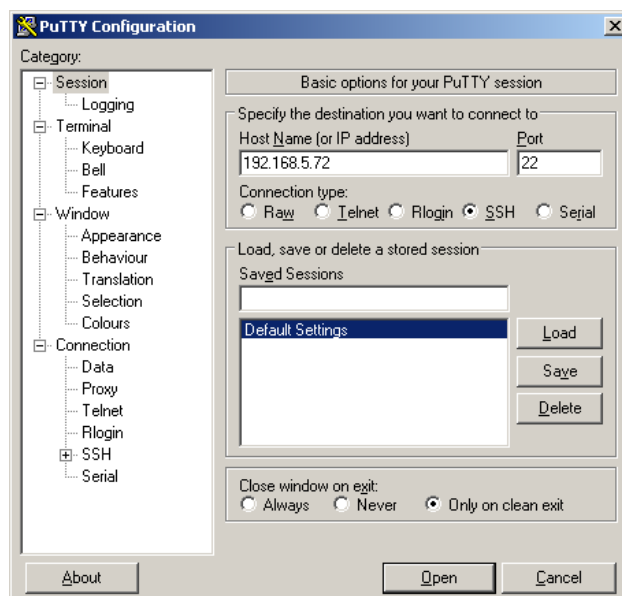
```
ifconfig eth1 192.168.5.91 netmask 255.255.255.0 up
```

Dodatkowo należy sprawdzić, czy interfejsy są aktywne i czy posiadają odpowiednią adresację IP:

```
192.168.5.72 - PuTTY
root@Snort:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:b7:7d:7c
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feb7:7d7c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2584 (2.5 KiB)  TX bytes:1938 (1.8 KiB)

root@Snort:~# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:fb:e6:d7
          inet addr:192.168.5.72  Bcast:192.168.5.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feb:e6d7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:757 errors:0 dropped:0 overruns:0 frame:0
          TX packets:194 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:76265 (74.4 KiB)  TX bytes:30230 (29.5 KiB)
```

W celu ułatwienia wprowadzenia komend do systemu Debian wykorzystaj narzędzie Putty, bądź inne oprogramowanie klienckie SSH:



Dodatkowo z racji, iż w sieci laboratoryjnej wykorzystuje się serwer pośredniczący należy odpowiednio skonfigurować proxy (wprowadzone zmiany istnieją jedynie do kolejnego restartu maszyny wirtualnej):

```
export http_proxy=http://w3cache.prz.edu.pl:8080
export https_proxy=$http_proxy
```

W celu sprawdzenia, czy poprawnie podstawowe komendy konfiguracyjne zostały wprowadzone należy korzystając z narzędzia wget pobrać dowolną stronę WWW; przykładowo należy wprowadzić:

```
wget http://google.pl
```

### 3. Instalacja niezbędnych pakietów

Po uruchomieniu maszyny wirtualnej z systemem Debian 7.7 należy sprawdzić czy są dostępne aktualizacje do zainstalowanego oprogramowania, a następnie w przypadku, gdyby istniały należy je zainstalować. W przypadku, gdyby w repozytorium znajdowały się starsze pakiety, można zedytować plik `/etc/apt/sources.list` wykorzystując do tego celu edytor tekstowy `nano` bądź `vi` wprowadzając inne serwery lustrzane.

```
apt-get update && apt-get upgrade
```

Zainstaluj dotdeb GnuPG key:

```
cd /usr/src && wget http://www.dotdeb.org/dotdeb.gpg
cat dotdeb.gpg | apt-key add -
```

Zainstaluj wszystkie niezbędne pakiety, w przypadku instalacji MySQL podaj hasło, które zapamiętasz (w celu ułatwienia procesu instalacji, nawiąż sesję SSH z hostem korzystając z Putty, a następnie skopiuj niezbędny fragment kodu):

```
apt-get install apache2 apache2-doc autoconf automake bison ca-certificates
ethtool flex g++ gcc gcc-4.4 libapache2-mod-php5 libcrypt-ssleay-perl
libmysqlclient-dev libnet1 libnet1-dev libpcre3 libpcre3-dev libphp-adodb
libssl-dev libtool libwww-perl make mysql-client mysql-common mysql-server
```

```
ntp php5-cli php5-gd php5-mysql php-pear sendmail sendmail-bin sysstat
usbmount
```

Zainstaluj bibliotekę C/C++ libpcap, dzięki której możliwe będzie przechwytywanie pakietów po stronie użytkownika:

```
cd /usr/src
wget http://www.tcpdump.org/release/libpcap-1.6.2.tar.gz
tar -zxf libpcap-1.6.2.tar.gz
cd libpcap-1.6.2
./configure --prefix=/usr
make
make install
```

Zainstaluj libdnet - bibliotekę, która zapewnia uproszczony, przenośny interfejs do niektórych niskopoziomowych funkcji sieciowych (min. manipulację adresami sieciowymi, przeglądanie i modyfikowanie pamięci podręcznej czy firewall'ing):

```
cd /usr/src && wget http://libdnet.googlecode.com/files/libdnet-1.12.tgz
tar -zxf libdnet-1.12.tgz
cd libdnet-1.12
./configure --prefix=/usr --enable-shared
make
make install
```

Zainstaluj bibliotekę DAQ, która zastępuje bezpośrednie wywołanie do funkcji libpcap. Biblioteka ułatwia pracę na wielu interfejsach sprzętowych oraz programowych bez konieczności wprowadzenia zmian do systemu Snort.

```
cd /usr/src
wget https://www.snort.org/downloads/snort/daq-2.0.4.tar.gz
tar xvfz daq-2.0.4.tar.gz
cd daq-2.0.4
./configure
make
make install
```

Wyłącz funkcje *Large Receive Offload* oraz *Generic Receive Offload* na interfejsie, na którym ruch będzie zbierany. Dzięki wyłączeniu funkcji zezwolimy na to, aby karta sieciowa nie wykonała defragmentacji pakietów zanim zostaną przetworzone przez jądro systemu:

```
ethtool --offload eth1 rx off tx off
ethtool -K eth1 gso off
ethtool -K eth1 gro off
```

Zmodyfikuj shared library path:

```
echo >> /etc/ld.so.conf /usr/lib
echo >> /etc/ld.so.conf /usr/local/lib && ldconfig
```

Zainstaluj oprogramowanie Snort, pobierając źródła z oficjalnej strony producenta:

```
cd /usr/src
wget https://www.snort.org/downloads/snort/snort-2.9.7.0.tar.gz
```



```
tar xvfz snort-2.9.7.0.tar.gz
cd snort-2.9.7.0
./configure --enable-sourcefire
make
make install
```

Utwórz foldery, w których będą umieszczane pliki konfiguracyjne oraz logi:

```
mkdir /usr/local/etc/snort
mkdir /usr/local/etc/snort/rules
mkdir /var/log/snort
mkdir /var/log/barnyard2
mkdir /usr/local/lib/snort_dynamicrules
touch /usr/local/etc/snort/rules/white_list.rules
touch /usr/local/etc/snort/rules/black_list.rules
touch /usr/local/etc/snort/sid-msg.map
```

Utwórz grupę oraz użytkownika o nazwie snort:

```
groupadd snort && useradd -g snort snort
chsh -s /bin/bash snort
```

W przypadku, gdyby takowy użytkownik istniał, dodaj go do grupy snort:

```
usermod -G snort snort
```

Zmień właściciela do pliku logów:

```
chown snort:snort /var/log/snort /var/log/barnyard2
```

Skopiuj przykładowe pliki konfiguracyjne do odpowiednich lokalizacji:

```
cp /usr/src/snort-2.9.7.0/etc/*.conf* /usr/local/etc/snort
cp /usr/src/snort-2.9.7.0/etc/*.map /usr/local/etc/snort
cp /usr/src/snort-2.9.7.0/etc/snort.conf /usr/local/etc/snort
```

#### 4. Modyfikacja pliku konfiguracyjnego snort.cfg

Korzystając z narzędzia *vi*, przejdź do edycji pliku konfiguracyjnego *snort.conf* znajdującego się w lokalizacji: */usr/local/etc/snort/*. Zmodyfikuj linię 45, która określi podsieć LAN stosowaną do komunikacji w laboratorium

Domyślna wartość:	Nowa wartość:
<b>ipvar HOME_NET any</b>	<b>ipvar HOME_NET 192.168.5.0/24</b>

Zmodyfikuj linię 48, która określa zewnętrzną podsieć:

Domyślna wartość:	Nowa wartość:
<b>ipvar EXTERNAL_NET any</b>	<b>ipvar EXTERNAL_NET !\$HOME_NET</b>

Zmodyfikuj linie #104, #109, #110, aby wskazać poprawną ścieżkę do zdefiniowanych zasad:

```
var RULE_PATH ./rules
var WHITE_LIST_PATH ./rules
var BLACK_LIST_PATH ./rules
```

Przejdź do linii #293, a następnie zmodyfikuj linię, dodając na końcu *max\_gzip\_mem 104857600*:

```
preprocessor http_inspect: global iis_unicode_map unicode.map 1252
```

```
compress_depth 65535 decompress_depth 65535 max_gzip_mem 104857600
```

Przejdź do sekcji *Configure output plugins* (linia #510):

```
#####  
# Step #6: Configure output plugins  
# For more information, see Snort Manual, Configuring Snort - Output Modules  
#####
```

Następnie dodaj w nowej linii:

```
output unified2: filename snort.log, limit 128
```

Przejdź do sekcji *Customize your rule set* (linia #540):

```
#####  
# Step #7: Customize your rule set  
# For more information, see Snort Manual, Writing Snort Rules  
#  
# NOTE: All categories are enabled in this conf file  
#####
```

Następnie zakomentuj (znacznikiem hasz) wszystkie linie w tej sekcji zaczynające się od wyrażenia *include \$RULE\_PATH/app-detect.rules* pozostawiając jedynie aktywną linię *include \$RULE\_PATH/local.rules*. Celem tej modyfikacji jest zapewnienie prostego środowiska testowania działania systemu SNORT:

```
#####  
# Step #7: Customize your rule set  
# For more information, see Snort Manual, Writing Snort Rules  
#  
# NOTE: All categories are enabled in this conf file  
#####  
  
# site specific rules  
include $RULE_PATH/local.rules  
#####  
  
#include $RULE_PATH/app-detect.rules  
#include $RULE_PATH/attack-responses.rules  
#include $RULE_PATH/backdoor.rules  
#include $RULE_PATH/bad-traffic.rules  
#include $RULE_PATH/blacklist.rules  
#include $RULE_PATH/botnet-cnc.rules  
#include $RULE_PATH/browser-chrome.rules
```

Zapisz wprowadzone zmiany (edytor nano CTRL+O, edytor vi :wq), a następnie otwórz plik */usr/local/etc/snort/rules/local.rules*, w celu wprowadzenia własnej prostej reguły w celu sprawdzenia poprawności funkcjonowania SNORT:

```
nano /usr/local/etc/snort/rules/local.rules  
alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:1;)
```

Po zapisaniu zmian w pliku uruchom SNORT wprowadzając w 1 lini komendę:

```
/usr/local/bin/snort -A console -q -u snort -g snort -c  
/usr/local/etc/snort/snort.conf -i eth1
```

Sprawdź komunikację z systemem, na którym uruchomiono SNORT, wykorzystując narzędzie ping; powinny pojawić się odpowiednie wpisy w oknie konsoli:

```
192.168.5.220 - PuTTY
root@Snort:~# /usr/local/bin/snort -A console -q -u snort -g snort -c /usr/local/etc/snort/snort.conf -i eth1
01/14-14:46:07.108295  [**] [1:1000001:1] ICMP test [**] [Priority: 0] {ICMP} 192.168.5.1 -> 192.168.5.220
01/14-14:46:07.108378  [**] [1:1000001:1] ICMP test [**] [Priority: 0] {ICMP} 192.168.5.220 -> 192.168.5.1
01/14-14:46:08.110698  [**] [1:1000001:1] ICMP test [**] [Priority: 0] {ICMP} 192.168.5.1 -> 192.168.5.220
01/14-14:46:08.110723  [**] [1:1000001:1] ICMP test [**] [Priority: 0] {ICMP} 192.168.5.220 -> 192.168.5.1
```

## 5. Konfiguracja Barnyard2

Barnyard2 w pewien sposób jest dodatkiem do oprogramowania Snort, który pozwala na zapisywanie logów oraz alertów w bardzo szybki sposób do postaci binarnych plików, które następnie Barnyard może odczytać i wprowadzić do bazy danych MySQL.

```
cd /usr/src && wget https://github.com/firnsy/barnyard2/archive/master.tar.gz
tar -zxf master.tar.gz
cd barnyard2-master/
autoreconf -fvi -I ./m4
```

W zależności of wersji systemu wprowadź jedną z dwóch komend:

```
./configure --with-mysql --with-mysql-libraries=/usr/lib/x86_64-linux-gnu/
./configure --with-mysql --with-mysql-libraries=/usr/lib/i386-linux-gnu
```

Wprowadź kolejne komendy w celu kompilacji, a następnie instalacji Barnyard2:

```
make
make install
mv /usr/local/etc/barnyard2.conf /usr/local/etc/snort
cp schemas/create_mysql /usr/src
```

Otwórz plik `/usr/local/etc/snort/barnyard2.conf` w edytorze tekstowym w celu zmodyfikowania linii #27 -30:

```
config reference_file: /usr/local/etc/snort/reference.config
config classification_file: /usr/local/etc/snort/classification.config
config gen_file: /usr/local/etc/snort/gen-msg.map
config sid_file: /usr/local/etc/snort/sid-msg.map
```

Linie #227 zmodyfikuj do postaci:

```
output alert_fast
```

Przejdź do końca pliku, a następnie wprowadź linię, która określi sposób komunikacji z Bazą Danych. Do konfiguracji poniżej linii podaj hasło wprowadzone przy instalacji MySQL:

```
output database: log, mysql, user=snort password=P@ssw0rd dbname=snort host=localhost
```

## 6. Konfiguracja MySQL oraz Apache2

Zaloguj się do serwera MySQL wprowadzając komendę:

```
mysql -u root -p
```

Utwórz tabelę snort, zwiększ uprawnienia użytkownikowi snort, ustaw hasło dla tego konta, oraz zaimportuj bazę wykorzystując plik z kopią BD. Sprawdź poprawność zaimportowania tabel:

```
create database snort;
grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to snort@localhost;
SET PASSWORD FOR snort@localhost=PASSWORD('P@ssw0rd');
use snort;
source /usr/src/create_mysql
show tables;
exit
```

Wprowadź poniższe komendy, aby uruchomić snort oraz barnyard2 (dwie komendy):

```
/usr/local/bin/snort -q -u snort -g snort -c /usr/local/etc/snort/snort.conf -i eth1 &
/usr/local/bin/barnyard2 -c /usr/local/etc/snort/barnyard2.conf -d /var/log/snort -f snort.log -
w /usr/local/etc/snort/bylog.waldo -C /usr/local/etc/snort/classification.config &
```

Sprawdź komunikację z systemem, na którym uruchomiono SNORT, wykorzystując narzędzie ping. Następnie po zalogowaniu się do bazy danych sprawdź czy pojawiły się wpisy:

```
mysql -u root -p -D snort
select * from event;
```

```
mysql> select * from event;
+-----+-----+-----+-----+
| sid | cid | signature | timestamp |
+-----+-----+-----+-----+
| 1 | 1 | 507 | 2014-12-22 17:34:54 |
| 1 | 2 | 507 | 2014-12-22 17:34:54 |
| 1 | 3 | 507 | 2014-12-22 17:34:55 |
| 1 | 4 | 507 | 2014-12-22 17:34:55 |
+-----+-----+-----+-----+
```

Skopiuj domyślnie utworzony certyfikat SSL do odpowiedniej lokalizacji:

```
cp /etc/apache2/sites-available/default-ssl /etc/apache2/sites-enabled
```

Otwórz plik `/etc/php5/apache2/php.ini`, na następnie zmodyfikuj linię #454 do postaci:

```
error_reporting = E_ALL & ~E_NOTICE
```

Wprowadź kolejne komendy:

```
a2enmod ssl
pear config-set preferred_state alpha
/etc/init.d/apache2 restart
```

## 7. Zadania do samodzielnego zrealizowania

Bazując na przeprowadzonej instalacji oraz modyfikacji plików systemu Snort należy przygotować własne reguły, które pozwolą wykryć konkretne działania występujące w sieci. Wszelkie zmiany przeprowadź na pliku `/usr/local/etc/snort/rules/local.rules`, co ułatwi sprawdzenie poprawności funkcjonowania systemu Snort oraz samych reguł. Dla każdej kolejnej reguły zwiększaj wartość `sid` o jeden.

Uwaga: W celu przyspieszenia procesu testowania reguł, po dodaniu nowej w pliku `local.rules`, uruchom Snort wprowadzając polecenie:

```
/usr/local/bin/snort -A console -q -u snort -g snort -c /usr/local/etc/snort/snort.conf -i eth1
```

- a) Sprawdź korzystając z polecenia `netstat -ant`, jakie porty są aktywne. W obecnej wersji systemu Debian, protokół Telnet jest domyślnie wyłączony. Napisz prostą regułę

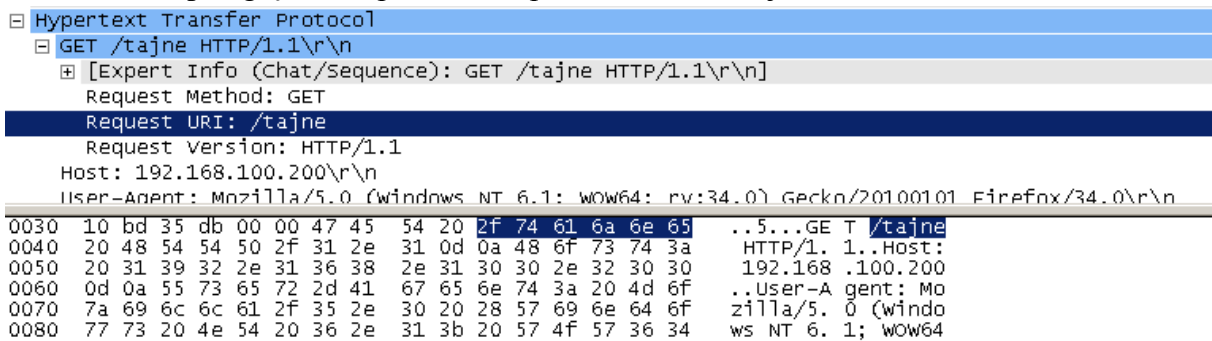
wedle, której pojawi się komunikat "Próba nawiązania połączenia Telnet", w sytuacji próby nawiązania połączenia TCP z dowolnego portu na adres systemu snort na porcie 23.

Wskazówka: Wykorzystaj opcję *flags: S+*, która sprawdza czy istnieje flaga SYN w połączeniu TCP.

```
alert tcp any any -> $HOME_NET 23 (msg:"Próba nawiązania połączenia Telnet"; flags: S+; sid:10000002; rev:1;)
```

- b) Napisz regułę wedle której będzie sprawdzana komunikacja do serwera Snort na porcie 80, w której atakujący żąda (GET) dostępu do podstron umieszczonych w lokalizacji */tajne*. Sklasyfikuj typ działania za pomocą wyrażenia *classtype: web-application-activity*;

Wskazówka: Sprawdź za pomocą Wireshark jakie dane zawiera żądanie GET do zasobu */tajne* na serwerze Apache2. Wykorzystaj opcję *nocase*; dla opcji *content:""*; dzięki czemu sprawdzane będą zarówno duże jak i małe znaki. W celu wygenerowania ataku, w oknie przeglądarki wprowadź `http://192.168.5.IP/tajne`



The screenshot shows a packet capture in Wireshark. The selected packet is an HTTP GET request. The details pane shows the following information:

- Request Method: GET
- Request URI: /tajne
- Request Version: HTTP/1.1
- Host: 192.168.100.200\r\n
- User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0\r\n

The packet bytes pane shows the raw data of the request, including the GET method and the URI.

```
alert tcp any any -> 192.168.5.0/24 80 (msg:"Próba uzyskania dostępu do /tajne"; content:"GET /tajne"; nocase; classtype: web-application-activity; sid:10000003; rev:1;)
```

- c) Napisz regułę wywołującą akcję *alert* wykorzystując opcję *pcre*, wykorzystującą wyrażenie regularne, dzięki któremu będą wyszukiwane żądanie GET do podstron htm bądź html:

```
alert tcp any any -> 192.168.5.0/24 80 (msg:"Test Logowania"; pcre:"/GET.*\.htm/i"; classtype: web-application-activity; sid:10000004; rev:1;)
```

- d) Napisz regułę wywołującą *alert*, w sytuacji, kiedy na adres serwera Snort bądź podsieć \$HOME\_NET zostanie przeprowadzony atak *Ping of Death*. Skorzystaj z opcji *dsize* określając, że atak tego typu występuje gdy rozmiar pakietu jest większy niż 10KB.

```
alert icmp any any -> 192.168.5.0/24 any (msg:"Ping of Death"; dsize: >10000; sid:10000005; rev:1;)
```

Po napisaniu wszystkich reguł należy sprawdzić czy pojawiają się kolejne rekordy w bazie danych snort w tabeli events. Przed uruchomieniem systemu Snort oraz Barnyard2, należy

usunąć pliki logów oraz wyczyścić zawartość pliku *bylog.waldo*, aby móc wygenerować rekordy do Bazy Danych. W tym celu należy wprowadzić następujące komendy:

```
pskill snort && pkill barnyard2
rm -rf /var/log/snort/* /var/log/barnyard2/*
rm /usr/local/etc/snort/bylog.waldo
touch /usr/local/etc/snort/bylog.waldo
```

W celu uruchomienia systemu Snort oraz Barnyard2 należy wprowadzić poniższe komendy:

```
/usr/local/bin/snort -q -u snort -g snort -c /usr/local/etc/snort/snort.conf -i eth1 &
/usr/local/bin/barnyard2 -c /usr/local/etc/snort/barnyard2.conf -d /var/log/snort -f
snort.log -w /usr/local/etc/snort/bylog.waldo -C /usr/local/etc/snort/classification.config &
```

Po uruchomieniu system Snort z Barnyard2 i po przeprowadzeniu symulacja taku pojawia się monit informujący o dodaniu sygnatury:

```
INFO [dbProcessSignatureInformation()]: [Event: 5] with [gid: 1] [sid: 1000005]
[rev: 1] [classification: 0] [priority: 0]
was not found in barnyard2 signature cache, this could lead to display
inconsistency.
To prevent this warning, make sure that your sid-msg.map and gen-msg.ma
p file are up to date with the snort process logging to the spool file.
The new inserted signature will not have its information present in the
sig_reference table.
Note that the message inserted in the signature table will be snort def
ault message "Snort Alert [gid:sid:revision]"
You can always update the message via a SQL query if you want it to be
displayed correctly by your favorite interface
```

Sprawdź jakie dokładne ataki zostały wykryte i zapisane do tabeli events BD snort:

```
mysql> SELECT DISTINCT signature from event;
+-----+
| signature |
+-----+
| 507 |
| 508 |
| 509 |
| 510 |
| 511 |
+-----+
5 rows in set (0.00 sec)
```

## 8. Konfiguracja BASE

Dzięki systemowi BASE można w prosty sposób przeglądać informacje zebrane przez Snort i zapisane w bazie MySQL.

```
cd /usr/src
wget http://sourceforge.net/projects/secureideas/files/BASE/base-1.4.5/base-1.4.5.tar.gz
tar -zxf base-1.4.5.tar.gz
cp -r base-1.4.5 /var/www/base
chmod 777 /var/www/base
```

Wprowadź w oknie przeglądarki <https://192.168.100.91/base/> (adres IP odpowiednio zamień).

## Basic Analysis and Security Engine (BASE) Setup Program

The following pages will prompt you for set up information to finish the install of BASE.  
If any of the options below are red, there will be a description of what you need to do below the chart.

Settings	
Config Writable:	Yes
PHP Version:	5.4.35-0+deb7u2
PHP Logging Level:	[ERROR][WARNING][PARSE]

[Continue](#)

Uzupełnił kolejno pokazujące się formularze zgodnie z wytycznymi. W oknie pierwszym w polu *Pick a Language* wybierz **english** bądź **polih**, w polu *Path to ADODB* wprowadź **/usr/share/php/adodb**

Step 1 of 5	
Pick a Language:	english ▾ [?]
Path to ADODB:	/usr/share/php/adodb [?]
<a href="#">Continue</a>	

W oknie, w którym konfigurujemy Bazę Danych uzupełnij pola następująco (pozostałe pozostaw puste):

- *Pick a Database type* **MySQL**
- *Database Name* **snort**
- *Database User Name* **snort**
- *Database Password* **P@ssw0rd**

Step 2 of 5	
Pick a Database type:	MySQL ▾ [?]
Database Name:	snort
Database Host:	localhost
Database Port: Leave blank for default!	
Database User Name:	snort
Database Password:	.....

W oknie trzecim, wprowadź ustawienia dostępu do systemu BASE:

Step 3 of 5	
<input type="checkbox"/> Use Authentication System [?]	
Admin User Name:	root
Password:	.....
Full Name:	Konto Administratora
<a href="#">Continue</a>	

W oknie czwartym kliknij "**Create BASE AG**". Upewnij się, że system zwrócił wyłącznie komunikaty *Successfully*, a następnie kliknij w odnośnik *step 5*.

Step 4 of 5		
Operation	Description	Status
BASE tables	Adds tables to extend the Snort DB to support the BASE functionality	Create BASE AG
	• snort	

Po poprawnym procesie konfiguracji powinno pojawić się okno systemu BASE:

Utwórz plik `/etc/init.d/snortbarn` korzystając z edytora tekstu nano w konsoli SSH, a następnie skopiuj poniższy kod (opcjonalnie, skopiuj plik przekazany przez Prowadzącego w oparciu o WinSCP):

```
#!/bin/sh
#
### BEGIN INIT INFO
# Provides: snortbarn
# Required-Start: $remote fs $syslog mysql
# Required-Stop: $remote fs $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# X-Interactive: true
# Short-Description: Start Snort and Barnyard
### END INIT INFO
. /lib/init/vars.sh
. /lib/lsb/init-functions
ifconfig eth1 192.168.5.91 netmask 255.255.255.0 up
export http proxy=http://w3cache.prz.edu.pl:8080
export https proxy=$http proxy
mysqld get param() {
/usr/sbin/mysqld --print-defaults | tr " " "\n" | grep -- "--$1" | tail -n 1 | cut -d= -f2
}

do start()
{
log_daemon_msg "Starting Snort and Barnyard" ""
# Make sure mysql has finished starting
ps_alive=0
while [ $ps_alive -lt 1 ];
do
pidfile=`mysqld get param pid-file`
if [ -f "$pidfile" ] && ps `cat $pidfile` >/dev/null 2>&1; then ps_alive=1; fi
sleep 1
done
/sbin/ifconfig eth1 up
/usr/local/bin/snort -q -u snort -g snort -c /usr/local/etc/snort/snort.conf -i eth1 &
```



```

/usr/local/bin/barnyard2 -c /usr/local/etc/snort/barnyard2.conf -d /var/log/snort -f snort.log -w
/usr/local/etc/snort/bylog.waldo -C /usr/local/etc/snort/classification.config 2> /dev/null &
log end msg 0
return 0
}
do_stop()
{
    log daemon msg "Stopping Snort and Barnyard" ""
    kill $(pidof snort) 2> /dev/null
    kill $(pidof barnyard2) 2> /dev/null
    log_end_msg 0
    return 0
}
case "$1" in
start)
do start
;;
stop)
do stop
;;
restart)
do_stop
do_start
;;
*)
echo "Usage: snort-barn {start|stop|restart}" >&2
exit 3
;;
esac
exit 0

```

Dodaj uprawnienie do uruchamiania do wcześniej utworzonego pliku, oraz utwórz dowiązanie symboliczne, aby system *Snort* oraz *Barnyard2* uruchamiał się automatycznie po starcie systemu Debian:

```

chmod +x /etc/init.d/snortbarn
inserv -f -v snortbarn
update-rc.d snortbarn defaults

```

Po przeładowaniu maszyny poleceniem *init 6* możesz sprawdzić czy niezbędne usługi załadowały się prawidłowo, do tego celu wprowadź komendę:

```
ps aux | grep snort
```

## 9. Pobieranie aktualnych reguł dzięki Puledpork

Wykorzystując Puledpork, możliwe jest pobranie aktualnych reguł udostępnianych przez producenta, a następnie wykorzystanie ich przez Snort do zwiększenia poziomu wykrywania zagrożeń.

```

cd /usr/src
wget https://pulledpork.googlecode.com/files/pulledpork-0.7.0.tar.gz
tar -zxf pulledpork-0.7.0.tar.gz
cd pulledpork-0.7.0
cp pulledpork.pl /usr/local/bin
cp etc/*.conf /usr/local/etc/snort

```

Aby wykorzystać *Sourcefire VRT Certified Rules*, należy posiadać zarejestrowane konto na stronie producenta systemu snort.org, a następnie wygenerować *oinkcode*, bądź wykorzystać

wartość wskazaną przez Prowadzącego zajęcia. Korzystając z edytora vi zedytuj plik `/usr/local/etc/snort/pulledpork.conf`

Linia	Wymagana zmiana:
#19	Wprowadź swoją wartość oinkcode, w miejscu <code>&lt;oinkcode&gt;</code> <pre>rule_url=https://www.snort.org/reg-rules/ snortrules-snapshot.tar.gz 13fdc7b993d891090aa022be78d14798b815e58f</pre>
#21	Zakomentuj linię dotyczącą reguł <code>community</code> : <pre># NEW Community ruleset: #rule_url=https://s3.amazonaws.com/snort-org/www/rules/community/ community-rules.tar.gz Community</pre>
#26	Wprowadź swoją wartość oinkcode, w miejscu <code>&lt;oinkcode&gt;</code> <pre>rule_url=https://www.snort.org/reg-rules/ opensource.gz 13fdc7b993d891090aa022be78d14798b815e58f</pre>
#27	Odkomentuj tą linię, aby wykorzystać <code>Emerging Rules</code>
#131	Zmień wartość <code>distro</code> na <code>distro=Debian-6-0</code> <pre># Define your distro, this is for the precompiled shared object libs! # Valid Distro Types: # Debian-5-0, Debian-6-0, # Ubuntu-8.04, Ubuntu-10-4 # Centos-4-8, Centos-5-4 # FC-12, FC-14, RHEL-5-5, RHEL-6-0 # FreeBSD-7-3, FreeBSD-8-1 # OpenBSD-4-8 # Slackware-13-1 distro=Debian-6-0</pre>
#139	Zakomentuj <code>BlackListe</code> <pre>#black_list=/usr/local/etc/snort/rules/iplists/default.blacklist</pre>
#194 ->197	Odkomentuj 3 linie: <pre>enablesid=/usr/local/etc/snort/enablesid.conf # dropsid=/usr/local/etc/snort/dropsid.conf disablesid=/usr/local/etc/snort/disablesid.conf modifysid=/usr/local/etc/snort/modifysid.conf</pre>

Wyłącz wszystkie reguły FWSAM:

```
echo pcre:fwsam >> /usr/local/etc/snort/disablesid.conf
```

Uruchom `PulledPork` (do pobrania jest kilkadziesiąt MB danych):

```
chmod +x /usr/local/bin/pulledpork.pl
/usr/local/bin/pulledpork.pl -c /usr/local/etc/snort/pulledpork.conf -T -l
```

```

1921685220 - PuTTY
root@Snort:/usr/src/pulledpork-0.7.0# /usr/local/bin/pulledpork.pl -c /usr/local/etc/snort/pulledpork.conf -T -l
http://code.google.com/p/pulledpork/
-----
PulledPork v0.7.0 - Swine Flu!
-----
Copyright (C) 2009-2013 JJ Cummings
cummingsj@gmail.com
Rules give me wings!
-----
Checking latest MDS for snortrules-snapshot-2970.tar.gz...
They Match
Done!
IP Blacklist download of http://labs.snort.org/feeds/ip-filter.blf...
Reading IP List...
Checking latest MDS for opensource.gz...
Rules tarball download of opensource.gz...
They Match
Done!
Checking latest MDS for emerging.rules.tar.gz...
Rules tarball download of emerging.rules.tar.gz...
They Match
Done!
Prepping rules from opensource.gz for work...
Done!
Prepping rules from emerging.rules.tar.gz for work...
Done!
Prepping rules from snortrules-snapshot-2970.tar.gz for work...
Done!
Reading rules...
Modifying Sids...
Done!
Processing /usr/local/etc/snort/enablesid.conf...
Modified 0 rules
Done
Processing /usr/local/etc/snort/disablesid.conf...
Modified 0 rules
Done
Setting Flowbit State...
Enabled 62 flowbits

```

Po pobraniu niezbędnych plików powinna pojawić się informacja o nowych regułach:

Rule Stats...

```
New:-----44023
Deleted:---0
Enabled Rules:----23552
Dropped Rules:----0
Disabled Rules:---20470
Total Rules:-----44022
No IP Blacklist Changes
Done
Please review /var/log/sid_changes.log for additional details
Fly Piggy Fly!
```

Sprawdź, czy zostały pobrane reguły oraz sprawdź ich składnię. W związku z dużym obciążeniem systemu przy aktywnych kilkudziesięciu tysiącach reguł nie uruchamiaj ich w testowanej maszynie wirtualnej.

```
cd /usr/local/etc/snort/rules/
ls -l
nano snort.rules
```

Dodatkowo istnieje możliwość dodania reguł do CRON, aby regularnie w zdefiniowanych odstępach pobierać bazę reguł, co pozwoli szybciej reagować na zagrożenia występujące w sieci.

## 10. Pytania kontrolne

- Z jakimi głównymi akcjami powiązane są utworzone reguły generowane na potrzeby systemu Snort.
- Czy istnieje mechanizm pozwalający pobierać możliwie aktualne reguły?
- Jaki jest główny problem badania zawartości ruchu sieciowego generowanego przez HTTPS?