Monitorovanie sieťových prostredí

Prerekvizity

Štandardné monitorovanie sieťových prostredí pozostáva zo zberu rôznych štatistických informácií a záznamov (angl. log) o vzniknutých udalostiach. Monitorovanými zariadeniami sú primárne servery a sieťové zariadenia ako smerovače a prepínače. Na týchto zariadeniach je vhodné monitorovať aktivity súvisiace s:

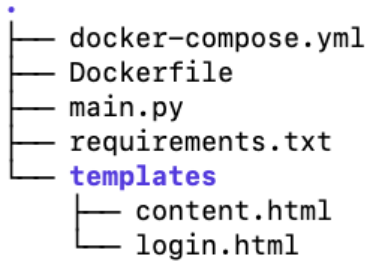
1. Prihlasovaním na služby:
   - SSH: vzdialené pripojenie za účelom konfigurácie
   - HTTP: prihlásenie sa na webové stránky
2. Prenosom dát:
   - IPFIX: Štatistické informácie o prenášaných dátach
3. Záznamy o rôznych udalostiach:
   - Syslog: zber log. udalostí

V rámci tohto vzdelávacieho materiálu budú ukázané scenáre, ako využitím nástroja Splunk zbierať rôzne typy záznamov a ich následná analýza pre účely odhalenia bezpečnostných incidentov. Monitorované bude:

1. Virtuálny smerovač CSR1kv
   a. Syslog správy - detekcia neštandardného správania
   b. Pokusy o SSH prihlásenie - detekcia útoku hrubou silou na prihlásenie
   c. Štatistiky prenosu dát - protokol IPFIX - detekcia skenovania portov, DoS,
2. Server (Ubuntu VM)
   a. Pokusy o SSH prihlásenie - detekcia útoku hrubou silou na prihlásenie
      i. /var/log/auth.log
   b. Snort hlásenia
      i. /var/log/snort/snort.alert.fast
3. Webová aplikácia
   a. Pokusy o HTTP prihlásenie - detekcia útoku hrubou silou na prihlásenie
      i. /var/log/nginx/access.log
      ii. /var/log/nginx/error.log

Webová aplikácia

Webová aplikácia je napísaná v Jazyku Python využitím knižnice Flask. Prihlásenie do aplikácie je zabezpečené použitím Cookies, ktoré obsahujú základné informácie o používateľovi [logged_in, username, ID]. Pre jednoduchosť bude aplikácia nasadzovaná využitím technológie docker. Celková štruktúra projektu vyzerá nasledovne:



docker-compose.yml

```
version: "3.9"

services:
  web-app:
     build: .
     ports:
     - "5000:5001"
```

Aplikácia bude po spustení dostupná na: http://<server_ip>:5000

Dockerfile

```
FROM python:3.13-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
CMD ["python", "main.py"]
```

main.py

```
from flask import Flask, render_template, request, session,
redirect, url_for, abort, Response

app = Flask(__name__)
app.secret_key = 'your_secret_key'  # used to sign cookie
# cookie is saved in browser Storage-Cookies encoded using base64
# session:
eyJsb2dnZWRfaW4iOnRydWUsInVzZXJfaWQiOjEzOCwidXNlcm5hbWUiOiJDTkwifQ.
Z2c0Cw.hv8USmren-swoXIzOHdMpWwr9ZI
# session: {"logged_in": true,"user_id": 138,"username": "CNL"}
```

```
@app.route('/')
def login():
      if 'logged_in' in session:
      return redirect(url_for('content'))
      return render_template('login.html')

@app.route('/login', methods=['POST'])
def login_post():
      username = request.form['username']
      password = request.form['password']

      if username == 'CNL' and password == 'cnlpass':
      session['logged_in'] = True
      session['username'] = username
      session['user_id'] = 138
      # session: {"logged_in": true,"user_id": 138,"username":
"CNL"}
      return redirect(url_for('content'))
      else:
      return Response('Invalid username or password', status=401)
      return redirect(url_for('login'))

@app.route('/content')
def content():
      if 'logged_in' in session:
      return render_template('content.html')
      else:
      return redirect(url_for('login'))

@app.route('/logout')
def logout():
      session.pop('logged_in', None)
      session.pop('username', None)
      session.pop('user_id', None)
      return redirect(url_for('login'))  # route to url associated
with login() function


if __name__ == '__main__':
      app.run(host='0.0.0.0', port=5001, debug=True)
```

requirements.txt

```
Flask
```

content.html

```
<!DOCTYPE html>
```

```
<html>
<head>
	<title>Welcome</title>
	<style>
	body {
		font-family: sans-serif;
		display: flex;
		justify-content: center;
		align-items: center;
		min-height: 100vh;
		background-color: #f0f0f0;
	}

	.container {
		background-color: #fff;
		padding: 30px;
		border-radius: 5px;
		box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
		text-align: center;
	}

	h1 {
		margin-bottom: 20px;
	}

	a {
		text-decoration: none;
		color: #4CAF50;
	}
	</style>
</head>
<body>
	<div class="container">
	<h1>Welcome {{ session.username }}!</h1>
	<p>You are successfully logged in.</p>
	<p><a href="{{ url_for('logout') }}">Logout</a></p>
	</div>
</body>
</html>
```

login.html

```
<html>
<head>
	<title>Login</title>
	<style>
	body {
		font-family: sans-serif;
```

```
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
            background-color: #f0f0f0;
      }

      .container {
            background-color: #fff;
            padding: 30px;
            border-radius: 5px;
            box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
      }

      h1 {
            text-align: center;
            margin-bottom: 20px;
      }

      label {
            display: block;
            margin-bottom: 10px;
      }

      input[type="text"],
      input[type="password"] {
            width: 100%;
            padding: 10px;
            margin-bottom: 15px;
            border: 1px solid #ccc;
            border-radius: 3px;
      }

      button {
            background-color: #4CAF50;
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 3px;
            cursor: pointer;
      }
      </style>
</head>
<body>
      <div class="container">
      <h1>Login</h1>
      <form method="POST" action="/login">
            <label for="username">Username:</label>
            <input type="text" id="username" name="username"
required>
```

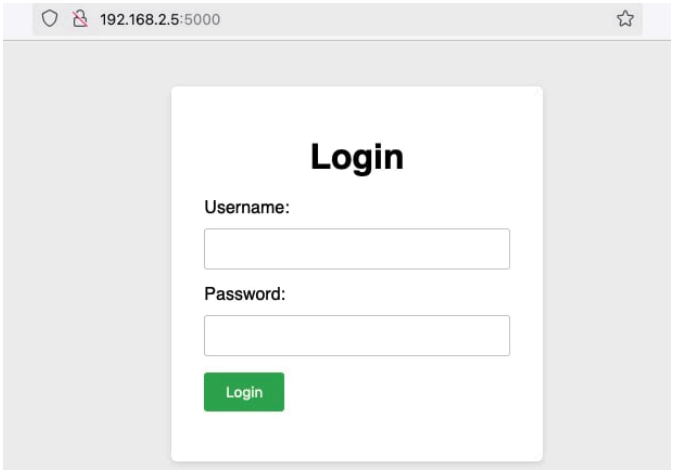
```
            <label for="password">Password:</label>
            <input type="password" id="password" name="password"
required>

            <button type="submit">Login</button>
      </form>
      </div>
</body>
</html>
```

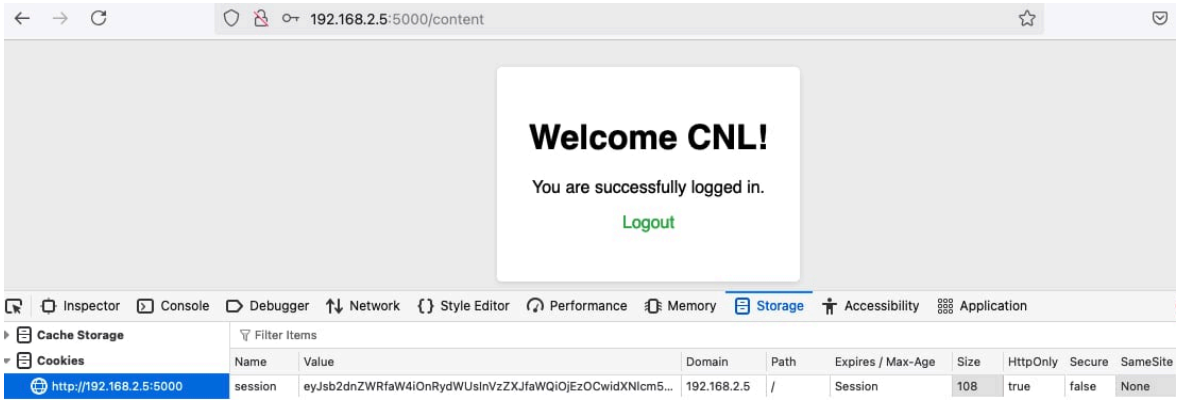
Používateľské rozhranie webovej aplikácie

Spustenie docker kontajnera webovej aplikácie je možné príkazom:

```
docker compose up -d
```

Webová aplikácia obsahuje dve stránky a to Login stránku a obsahovú stránku. Obsahová stránka je dostupná len prihlásenému používateľovi. Prihlásenie je implementované v podobe generovania Cookies (base64 hodnota obsahujúca stav prihlásenia, meno používateľa, ID a digitálny podpis), ktoré slúžia na overenie identity používateľa.





Nginx server

Webové aplikácie sa štandardne nachádzajú za tzv. reverse proxy technológiou, ktorá umožňuje monitorovanie prístupu a množstvo ďalších nastavení webového servera. Pre simuláciu reálnej interakcie s webovou aplikáciou bude táto aplikácia dostupná pod doménou web-app.local. Mapovanie prístupu z danej domény na webovú aplikáciu pracujúcu na porte 5000 bude zabezpečené Nginx serverom.

Inštalácia Nginx

```
apt install nginx

systemctl status nginx.service
systemctl stop nginx.service
systemctl start nginx
```

Nastavenie reverse proxy funkcionality

```
nano /etc/nginx/sites-available/web-app.local
```

```
server {
       listen 80;
       listen [::]:80;

       server_name web-app.local www.web-app.local;

       location / {
                  proxy_pass http://127.0.0.1:5000;
                  proxy_set_header Host $http_host;
                  proxy_set_header X-Real-IP $remote_addr;
                  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                  proxy_set_header X-Forwarded-Proto $scheme;
       }
}
```

```
ln -s /etc/nginx/sites-available/web-app.local /etc/nginx/sites-enabled/

systemctl restart nginx.service
```

Zobrazenie prístupu na webovú službu a vzniknuté chybové hlášky je možné vidieť v súboroch:

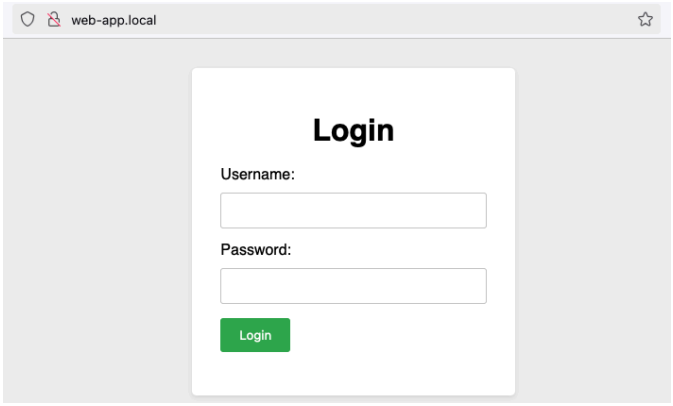
```
cat /var/log/nginx/access.log

sudo cat /var/log/nginx/error.log
```

Overenie prístupu cez doménový názov

Na zariadení, z ktorého sa bude pristupovať k vytvorenej webovej aplikácii je potrebné nastaviť preklad domény na IP adresu servera. Najjednoduchšie je pridať statické mapovanie do súboru [**/etc/hosts**]:

```
192.168.2.5      web-app.local
```

Ako je možné vidieť na nasledujúcom obrázku, prístup k webovej aplikácii je možný využitím doménového názvu.



Vďaka využitiu Nginx servera je možné zobraziť kompletnú komunikáciu s webovou aplikáciou a to zobrazením obsahu súboru [/var/log/nginx/access.log]

```
192.168.2.4 - - [22/Dec/2024:16:59:25 +0100] "GET /favicon.ico HTTP/1.1" 404 184
"http://web-app.local/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:133.0) Gecko/20100101
Firefox/133.0"

192.168.2.4 - - [22/Dec/2024:17:01:39 +0100] "POST /login HTTP/1.1" 401 28
"http://web-app.local/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:133.0) Gecko/20100101
Firefox/133.0"

192.168.2.4 - - [22/Dec/2024:17:01:39 +0100] "GET /favicon.ico HTTP/1.1" 499 0
"http://web-app.local/login" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:133.0)
Gecko/20100101 Firefox/133.0"

192.168.2.4 - - [22/Dec/2024:17:01:48 +0100] "POST /login HTTP/1.1" 401 28 "http://web-app.local/"
"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:133.0) Gecko/20100101 Firefox/133.0"

192.168.2.4 - - [22/Dec/2024:17:02:00 +0100] "POST /login HTTP/1.1" 302 203
"http://web-app.local/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:133.0) Gecko/20100101
Firefox/133.0"

192.168.2.4 - - [22/Dec/2024:17:02:00 +0100] "GET /content HTTP/1.1" 200 438
"http://web-app.local/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:133.0) Gecko/20100101
```

```
Firefox/133.0"

192.168.2.4 - - [22/Dec/2024:17:02:03 +0100] "GET /logout HTTP/1.1" 302 189
"http://web-app.local/content" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:133.0)
Gecko/20100101 Firefox/133.0"

192.168.2.4 - - [22/Dec/2024:17:02:03 +0100] "GET / HTTP/1.1" 200 608
"http://web-app.local/content" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:133.0)
Gecko/20100101 Firefox/133.0"
```

Z uvedených LOG záznamov, je možné vidieť kto pristupuje k webovej aplikácii [192.168.2.4], akými metódami [POST, GET], na aké URL [http://web-app.local], na aké endpointy [/login, /content, /logout] a aký status kód bol po vykonanej akcii vrátený používateľovi [200, 302, 401].

Nasadenie Splunk servera

Splunk predstavuje nástroj pre zber a analýzu dát za účelom detekcie bezpečnostných incidentov. Splunk je možné nasadiť manuálne (priamo na server), alebo využitím docker obrazu. Pre jednoduchosť bude použitá technológia docker.

Spustenie docker kontajnera

Pre spustenie splunk servera je možné využiť oficiálny docker obraz, akceptovať licenciu a nastaviť heslo pre admin prístup. Následne je potrebné povoliť všetky porty, ktoré budú použité pre prenos monitorovaných záznamov na Splunk server [používateľské rozhranie, syslog, Netflow a HTTP]. Pre uchovanie nastavení a prístup k vybraným log. súborom z lokálneho súborového systému je tiež potrebné využitím Volumes mapovať prístup k týmto súborom.

```
nano docker-compose.yml
```

```
version: "3.9"

services:
  splunk-server:
       image: splunk/splunk:latest
       container_name: splunk-free
       environment:
       - SPLUNK_START_ARGS=--accept-license
       - SPLUNK_PASSWORD=P4ssw0rd123!
       - SPLUNK_HOST=0.0.0.0
       ports:
       - 8000:8000   # GUI
       - 9997:9997/udp   # UDP syslog port
       - 514:514/udp   # Common UDP syslog port (optional)
       - 2055:2055/udp   # NetFlow port
       - 8088:8088   #HTTP Event Collector
       volumes:
```

```
      - ./splunk_data:/opt/splunk/var/lib/splunk
      -
/var/log/nginx/access.log:/opt/splunk/etc/system/local/inputs/nginx_access.log
      - /var/log/nginx/error.log:/opt/splunk/etc/system/local/inputs/nginx_error.log
      - /var/log/auth.log:/opt/splunk/etc/system/local/inputs/auth.log
      -
/var/log/snort/snort.alert.fast:/opt/splunk/etc/system/local/inputs/snort.log
```

Nastavenie Nginx servera

Pre sprístupnenie Splunk servera pod jeho doménovým názvom je potrebné vykonať nasledujúcu úpravu v konfigurácii Nginx servera:

```
nano /etc/nginx/sites-available/splunk.local
```

```
server {
      listen 80;
      listen [::]:80;

      server_name splunk.local www.splunk.local;

      location / {
                  proxy_pass http://127.0.0.1:8000;
                  proxy_set_header Host $http_host;
                  proxy_set_header X-Real-IP $remote_addr;
                  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                  proxy_set_header X-Forwarded-Proto $scheme;
      }
}
```

```
ln -s /etc/nginx/sites-available/splunk.local /etc/nginx/sites-enabled/

systemctl restart nginx.service
```

Nastavenie prekladu doménového názvu na IP adresu

Na zariadení, z ktorého sa bude pristupovať k Splunk serveru je potrebné nastaviť preklad domény na IP adresu servera. Najjednoduchšie je pridať statické mapovanie do súboru [**/etc/hosts**]:

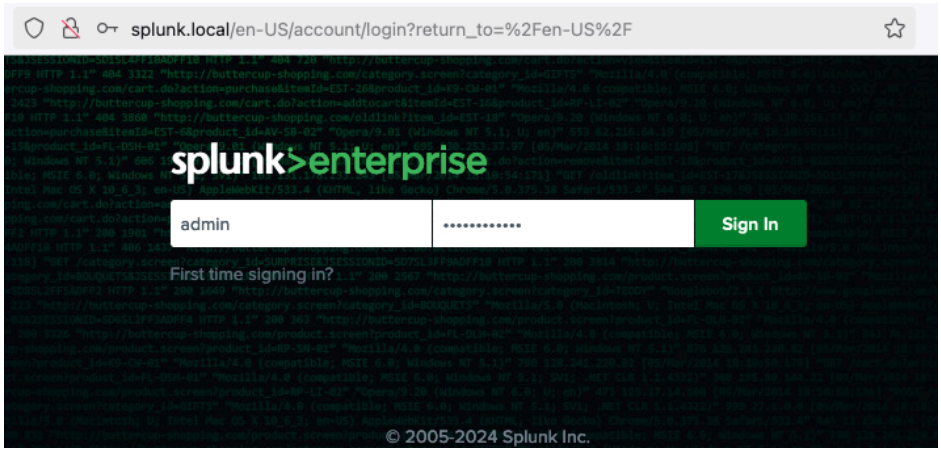
```
192.168.2.5      splunk.local
```

Prvotné spustenie prihlásenie a nastavenie Free licencie

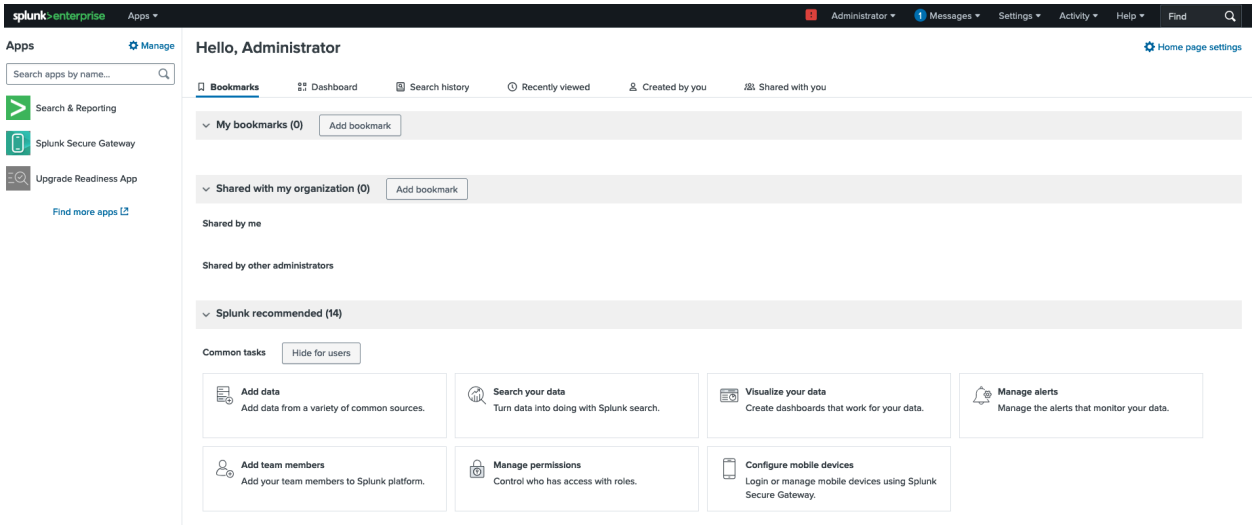
Spustenie docker kontajnera Splunk servera je možné príkazom:

```
docker compose up -d
```

Prvotné prihlásenie je možné využitím používateľa **admin** a hesla definovaného v docker-compose súbore.



Základné používateľské rozhranie Splunk servera vyzerá nasledovne:



Po prvotnom spustení je aktivovaná Trial licencia. V rámci tohto vzdelávacieho materiálu sa bude pracovať s Free verziou, ktorú je možné nastaviť nasledovne:

1. Settings - SYSTEM - Licensing
2. Change license group
3. Free licence
4. Save
5. Restart Now (tento krok je vhodné vykonať aj na úrovni docker kontajnera)

Free Licencia neposkytuje autentifikáciu, manažment používateľov a v rámci jedného dňa umožňuje indexovať maximálne 500MB.

Centralizované ukladanie dát nástrojom Splunk

Splunk si je možné predstaviť ako platformu pre centralizovaný zber dát rôzneho typu.

Nastavenie zdroja dát

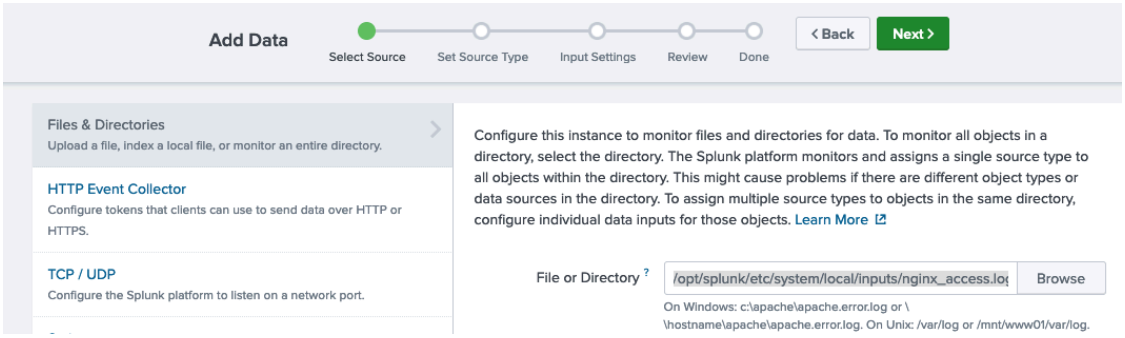
Pre vloženie akýchkoľvek dát do Splunk servera je potrebné vykonať nasledovnú postupnosť krokov:

1. Settings - Add Data
2. Upload (manuálne vloženie .log súboru) / Monitor (príjem dynamicky získaných dát)
3. Voľba zdroja dát - najčastejšie:
    a. Files & Directories (lokálny súborový systém)
    b. HTTP Event Collector (príjem dát v HTTP požiadavkách)
    c. TCP/UDP (otvorenie portu pre príjem dát zo štandardných protokolov - Syslog = 514/udp)

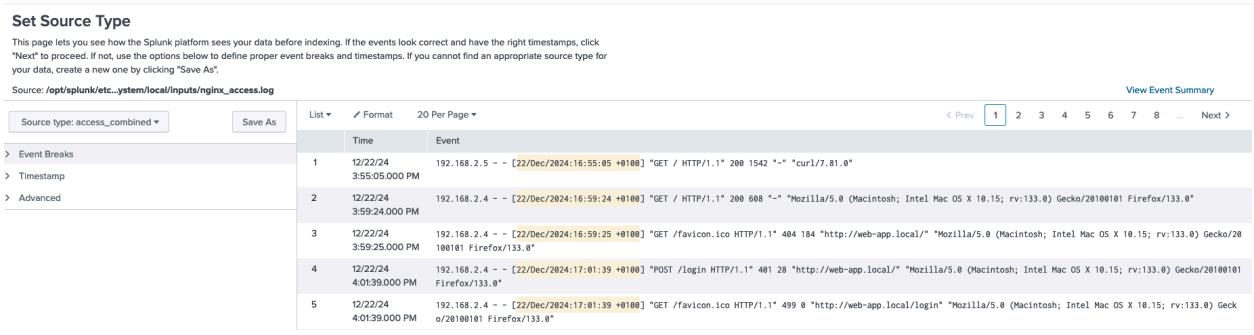
Zber log. správ z lokálneho súborového systému

Pre tento účel je potrebné zvoliť ako zdroj dát **Files & Directories** a vykonať nasledujúce nastavenia:

1. Zvoliť cestu k danému súboru (cesta k súboru vo vnútri Docker kontajnera): /opt/splunk/etc/system/local/inputs/nginx_access.log



2. Nastaviť Typ zdroja dát (access_combined)

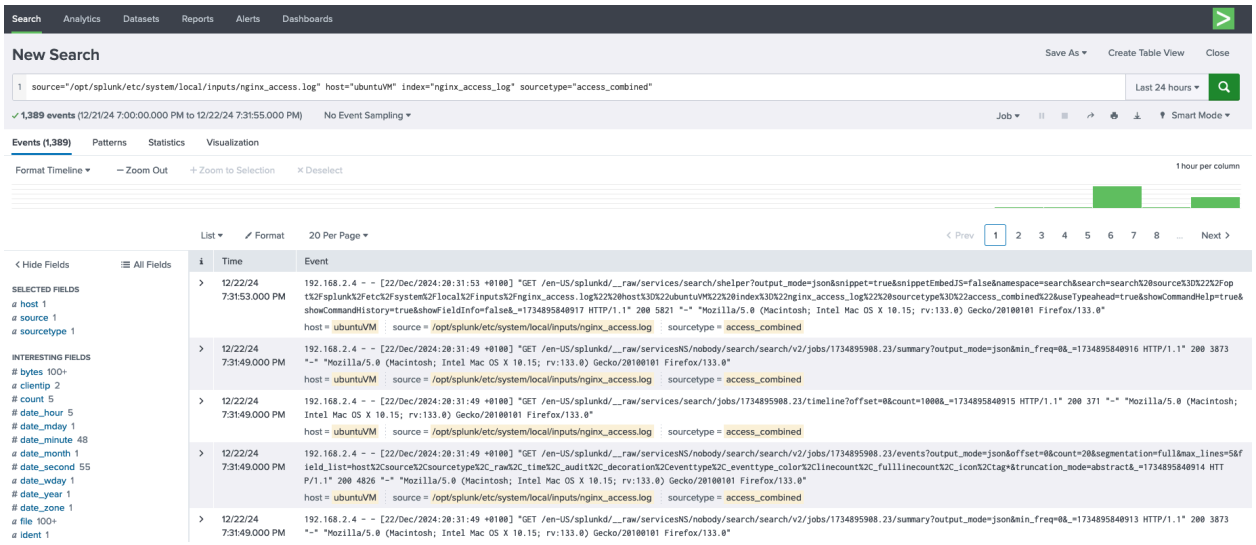


3. Nastavenie názvu zariadenia, pod ktorým budú dáta v rámci Splunk servera identifikované
4. Nastavenie názvu indexu, ktorý bude predstavovať získané dáta.
5. Overenie nastavenej konfigurácie

Po pridaní daného zdroja Splunk automaticky vygeneruje reťazec (Query) pre identifikáciu uložených dát. V tomto prípade sa jedná o:

```
source="/opt/splunk/etc/system/local/inputs/nginx_access.log"
host="ubuntuVM" index="nginx_access_log"
sourcetype="access_combined"
```

Uložené dáta je možné prehľadávať po kliknutí na **Splunk>enterprise - Search & Reporting**.



Pre prístup k SSH autentifikačnému log súboru je postupnosť krokov rovnaká. Možný tvar požiadavky pre získanie daných dát je nasledovný:

```
source="/opt/splunk/etc/system/local/inputs/auth.log"
host="ubuntuVM" index="ssh_auth_log" sourcetype="SSH_auth"
```

Zber Syslog správ zo sieťových zariadení

Prerekvizitou je vzájomná dostupnosť sieťového zariadenia a servera, na ktorom je spustený Splunk.

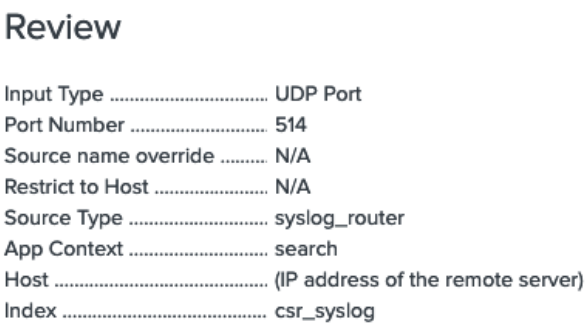
Konfigurácia CSR1kv

```
logging host 192.168.2.5 !ip address Splunk servera
! (optional) logging host 192.168.2.5 transport tcp
! (optional) logging host 192.168.2.5 transport udp port 514
session-id string csr1kv
```

Konfigurácia Splunk servera

Pre priamy príjem Syslog dát je potrebné spustiť odchytávanie dát na Syslog porte (514).

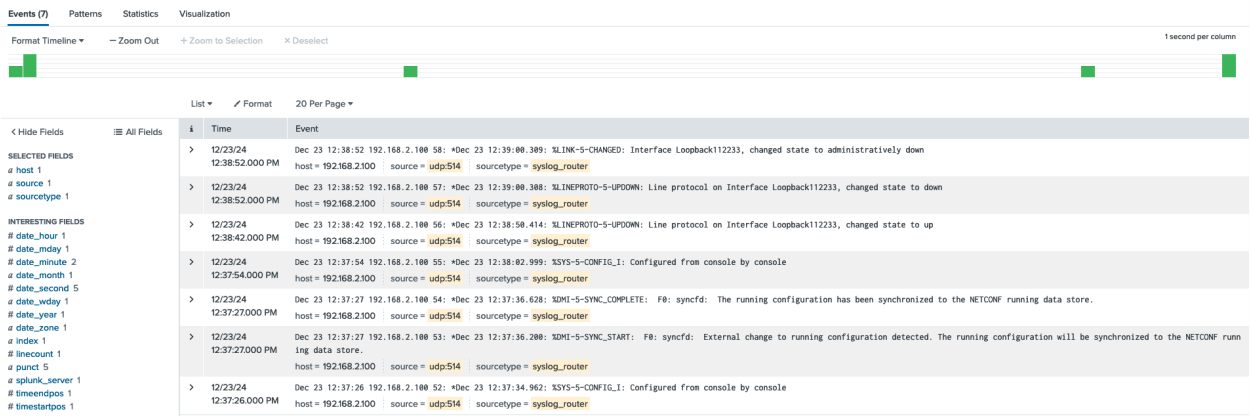
- Source type: syslog_router - custom
- App context: Search & Reporting
- Host: IP (slúži na filtrovanie podľa zdroja dát)
- Index: csr_syslog



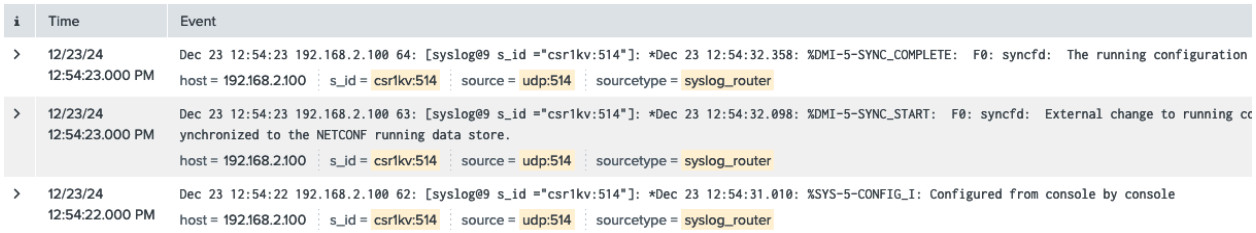
Možný tvar požiadavky pre získanie daných dát je nasledovný:

```
source="udp:514" index="csr_syslog" sourcetype="syslog_router"
```

Po vygenerovaní syslog udalostí, je možné, dané log. správy vidieť priamo v Splunk rozhraní.



Ak je potrebný ďalší atribút na filtrovanie dát, je ho možné pridať v rámci konfigurácie smerovača atribútom [`session-id string csr1kv`]. V Splunk rozhraní je daná hodnota uložená do premennej [s_id]:



i	Time	Event
>	12/23/24 12:54:23.000 PM	Dec 23 12:54:23 192.168.2.100 64: [syslog@9 s_id ="csr1kv:514"]: *Dec 23 12:54:32.358: %DMI-5-SYNC_COMPLETE: F0: syncfd: The running configuration host = 192.168.2.100 s_id = csr1kv:514 source = udp:514 sourcetype = syslog_router
>	12/23/24 12:54:23.000 PM	Dec 23 12:54:23 192.168.2.100 63: [syslog@9 s_id ="csr1kv:514"]: *Dec 23 12:54:32.098: %DMI-5-SYNC_START: F0: syncfd: External change to running cc ynchronized to the NETCONF running data store. host = 192.168.2.100 s_id = csr1kv:514 source = udp:514 sourcetype = syslog_router
>	12/23/24 12:54:22.000 PM	Dec 23 12:54:22 192.168.2.100 62: [syslog@9 s_id ="csr1kv:514"]: *Dec 23 12:54:31.010: %SYS-5-CONFIG_I: Configured from console by console host = 192.168.2.100 s_id = csr1kv:514 source = udp:514 sourcetype = syslog_router

Možný tvar požiadavky pre získanie dát od špecifického zdroja (s_id) je nasledovný:

```
source="udp:514" index="csr_syslog" sourcetype="syslog_router"
s_id="csr1kv:514"
```

Zber Netflow (IPFix) dát zo sieťových zariadení

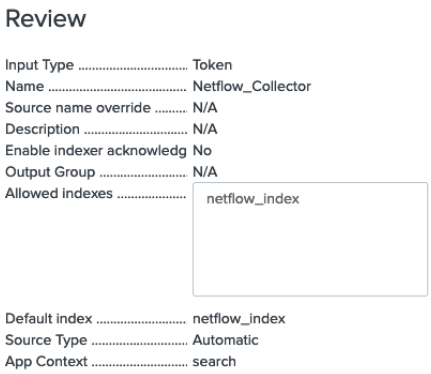
Konfigurácia CSR1kv

```
flow record rec
 match ipv4 source address
 match ipv4 destination address
 collect counter packets
 collect datalink mac source address input
 collect counter bytes
!
!
flow exporter exp
 destination 192.168.2.5
 source GigabitEthernet1
 transport udp 2055
!
!
flow monitor mon
 exporter exp
 cache timeout inactive 5
 cache timeout active 10
 record rec
!
interface gi1
  ip flow monitor mon input
```

Konfigurácia Splunk servera

Príjem NetFlow dát, vzhľadom k ich komplexnej štruktúre vyžaduje ich prvotné predspracovanie, napr. nástrojom Logstash a ich následné odoslanie na Splunk server využitím HTTP protokolu. Splunk poskytuje **HTTP Event Collector**, ktorý dokáže prijať HTTP správu a indexovať prijatý obsah.

- Name: Netflow_Collector
- Source type: Automatic
- Index: netflow_index



Vygenerovaný token má podobu: **b904950d-9ea1-4c6e-bf61-b16a20ba247a** (pri každom generovaní sa reťazec mení).

Možný tvar požiadavky pre získanie dát o dátových tokoch je nasledovný:

```
source="http:Netflow_Collector" index="netflow_index"
```

Splunk je pripravený pre príjem Netflow dát, ktoré mu budú doručené v obsahu HTTP (Settings - Data Inputs - HTTP Event Collector - Global Settings - Enable SSL bez zaškrtnutia) požiadaviek v podobe:

- URL:
  - http://192.168.2.5:8088/services/collector/raw
  - optional: http://192.168.2.5:8088/services/collector/event
- Authorization
  - API Key
    - Key: Authorization
    - Value: Splunk b904950d-9ea1-4c6e-bf61-b16a20ba247a
- Payload
  - /raw = akékoľvek raw data
  - /event =

```
{
  "sourcetype": "test",
  "event": {
     "ipv4_src_addr": "192.168.2.3",
     "in_src_mac": "44:f0:9e:a8:b5:d1",
     "in_bytes": 84
  }
}
```

Konfigurácia Logstash nástroja

Logstash je nástroj na spracovanie dát. Z pohľadu funkčnosti obsahuje nasledujúce tri komponenty:

1. Príjem dát
    - syslog, súbory, databázy, akýkoľvek port (napr. 2055 pre netflow dáta)
2. Transformácia dát
    - grok, mutate, … Slúži na filtrovanie a úpravu prijatých dát
3. Odoslanie dát
    - súbor (csv, json), databáza (elasticsearch), HTTP endpoint, syslog, …

Logstash je možné spustiť ako docker kontajner, pričom všetky konfiguračné nastavenie sa definujú v súbore logstash.conf. Adresár pre ukladanie logstash dát je potrebné predom vytvoriť a dať práva na zapisovanie.

docker-compose.yml

```
version: "3.8"
services:
  logstash:
      image: logstash:7.9.1
      container_name: logstash
      ports:
      - "5044:5044"
      - "9600:9600"
      - "2055:2055/udp"
      volumes:
      - ./logstash.conf:/usr/share/logstash/pipeline/logstash.conf
      - ./logstash.yml:/usr/share/logstash/config/logstash.yml
      - ./ls_data:/usr/share/logstash/data
      networks:
      - elk

networks:
  elk:
      driver: bridge
```

logstash.conf

```
input {
  udp {
      port => 2055
      type => "netflow"
      codec => netflow {
      versions => [9]
      }
      tags => ["csr_netflow_tag"]
  }
}
```

```
filter {
  if [type] == "netflow" {
     mutate {
     rename => { "host" => "netflow_host" }
     }
  }
}

output {
  if [type] == "netflow" {
   http {
     content_type => "application/json"
     http_method => "post"
     url => "http://192.168.2.5:8088/services/collector/raw"
     headers => ['Authorization', 'Splunk
b904950d-9ea1-4c6e-bf61-b16a20ba247a']
     }

   stdout {
   }
 }
}
```

logstash.yml

```
http.host: 0.0.0.0
```

V uvedenom konfiguračnom súbore je možné vidieť spustenie odchytávania netflow dát na porte 2055, úpravu ich obsahu (zmena atribútu host na netflow_host - potrebné lebo Logstash odosiela metadáta s názvom host, ktoré obsahujú IP adresu logstash servera) a odoslanie HTTP protokolom na Splunk endpoint.

Po spustení bude chvíľu logstash hlásiť chybu:

```
Can't (yet) decode flowset id 256 from source id 256, because no
template to decode it with has been received. This message will
usually go away after 1 minute.
```

Dôvod je v tom, že Netflow protokol posiela štatistické dáta a k nim šablónu (posielaná +- raz za minútu). Logstash musí najskôr prijať šablónu, aby vedel dekódovať význam štatistických dát a až následne dekódované dáta bude posielať na Splunk server.

Zobrazenie Netflow dát

Po vygenerovaní komunikácie prechádzajúcej cez rozhranie GigabitEthernet1 na smerovači CSR1kv, je možné, dané Štatistické informácie o dátových tokoch vidieť priamo v Splunk rozhraní. Nasledujúci obrázok znázorňuje formát prijatých dát zobrazených v používateľskom rozhraní Splunk servera.

Zber .log súborov z externých serverov

Pre získavanie obsahu .log súborov z externých serverov je potrebné mať na externých serveroch nainštalovaný nástroj Filebeat, ktorý dokáže odosielať obsah súborov do nástroja Logstash.

Inštalácia nástroja Filebeat

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
sudo apt-key add -
sudo apt-get install apt-transport-https
echo "deb https://artifacts.elastic.co/packages/8.x/apt stable
main" | sudo tee -a /etc/apt/sources.list.d/elastic-8.x.list

sudo apt-get update && sudo apt-get install filebeat
sudo systemctl start filebeat
sudo systemctl enable filebeat
```

Konfigurácia nástroja Filebeat

Nastavenie nástroja Filebeat umožňuje definovať monitorované súbory (paths), pridať k nim riadiace atribúty (fields - neskôr použité na filtrovanie) a nastaviť cieľ, kam obsah súboru bude odosielaný (logstash).

/etc/filebeat/filebeat.yml

```
- type: log
  enabled: true
  paths:
     - /var/log/nginx/access.log
  fields:
```

```
    log_type: nginx
    log_src: ubuntuVM

# Comment elasticsearch output and enable logstash
output.logstash:
  hosts: ["localhost:5044"]
```

Nastavenie Logstash - prijatie filebeat dát a odosielanie na HTTP endpoint

logstash.conf

```
input {
  beats{
   port => 5044
   type => "filebeats"
   }
}

output {
if [type] == "filebeats" {
      http {
         content_type => "application/json"
         http_method => "post"
         url => "http://192.168.2.5:8088/services/collector/raw"
         headers => ['Authorization', 'Splunk
b904950d-9ea1-4c6e-bf61-b16a20ba247a']
         }
     }
}
```

Formát prijatých dát v Splunk rozhraní je znázornený na nasledujúcom obrázku:

Zber IDS oznámení o incidentoch z nástroja snort

Snort je IDS systém pre sledovanie rôznych udalostí a generovanie hlásení o potenciálnych bezpečnostných incidentoch.

Snort inštalácia

```
sudo apt-get update
sudo apt-get install snort -y

snort --version

cd /etc/snort
sudo cp snort.conf snort.conf.backup
```

Nastavenie Snort služby

```
sudo nano /lib/systemd/system/snort.service
[Unit]
Description=Snort NIDS Daemon
After=syslog.target network.target

[Service]
Type=simple
ExecStart=/usr/sbin/snort -q -D -u snort -g snort -c
/etc/snort/snort.conf -i enp0s25

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl daemon-reload
sudo systemctl start snort
sudo systemctl status snort
```

Nastavenie snort pravidiel

V rámci pravidiel je možné nastaviť rozsah pozorovaných IP adries, umiestnenie pravidiel pre kontrolu a výstupných súborov.

```
sudo nano /etc/snort/snort.conf
ipvar HOME_NET 192.168.2.5/32
# include ...
include $RULE_PATH/local.rules

output log_tcpdump: tcpdump.log
```

```
output alert_fast: snort.alert.fast
output alert_unified2: filename snort.alert, limit 128, nostamp
output unified2: filename snort.log, limit 128, nostamp,
mpls_event_types, vlan_event_types
```

V rámci konfiguračného súboru je možné nastaviť, aké pravidlá budú používané pri IDS funkcionalite. Pravidlá, ktoré aktuálne nie sú potrebné je vhodné zakomentovať.

V rámci nástroja Snort je možné používať komunitné pravidlá (stiahnuteľné zdarma), registrované pravidlá (vyžadujú registráciu) alebo pravidlá vytvorené administrátorom. Vytvorenie vlastných pravidiel vyzerá nasledovne:

```
sudo nano /etc/snort/rules/local.rules
alert icmp any any -> $HOME_NET any (msg:"ICMP detected";
sid:10000001; rev:001;)
alert tcp any any -> $HOME_NET 22 (msg:"SSH Detected"; sid:
1000002; rev:1;)
```

Uvedené pravidlá budú vytvárať hlásenie po prijatí ICMP alebo SSH komunikácie.

Snort hlásenia a ich odosielanie na Splunk

Snort hlásenia je možné zobraziť nasledovne:

```
sudo tail -f /var/log/snort/snort.alert.fast
```

```
12/24-23:07:54.003724  [**] [1:10000001:1] ICMP detected [**] [Priority: 0] {ICMP} 192.168.2.3 -> 192.168.2.5
12/24-23:07:54.500818  [**] [1:1000002:1] SSH Detected [**] [Priority: 0] {TCP} 192.168.2.3:63762 -> 192.168.2.5:22
12/24-23:07:55.005131  [**] [1:10000001:1] ICMP detected [**] [Priority: 0] {ICMP} 192.168.2.3 -> 192.168.2.5
12/24-23:07:55.525001  [**] [1:1000002:1] SSH Detected [**] [Priority: 0] {TCP} 192.168.2.3:63762 -> 192.168.2.5:22
```

Obsah daného súboru je možné využitím nástroja Filebeat odosielať na Logstash server, ktorý ich následne odošle na Splunk server.

/etc/filebeat/filebeat.yml

```
- type: log
  enabled: true
  paths:
     - /var/log/snort/snort.alert.fast
  fields:
    log_type: snort
    log_src: ubuntuVM

output.logstash:
  hosts: ["localhost:5044"]
```

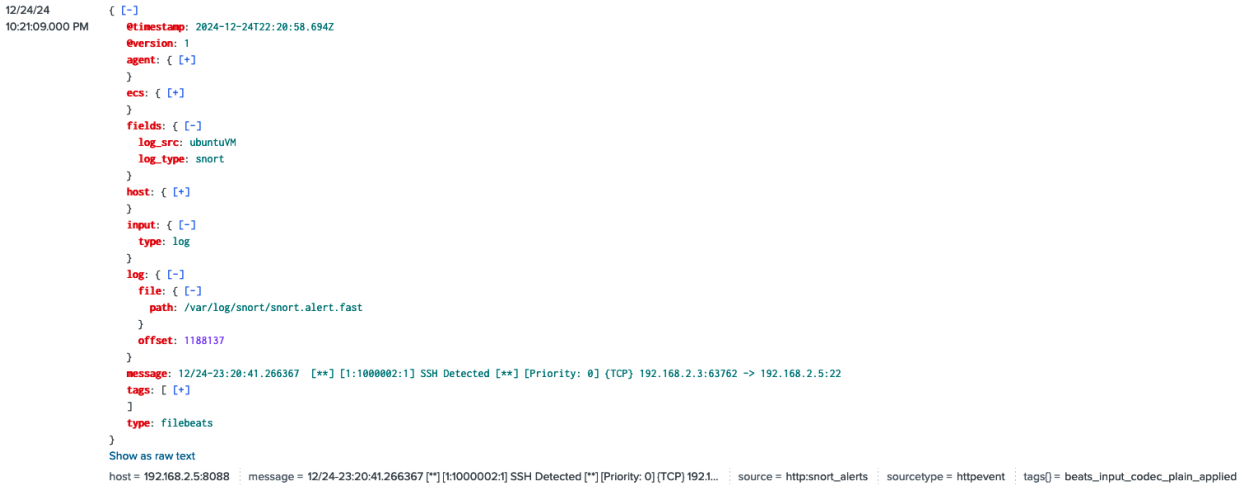
Pred odoslaním Snort hlásení na Splunk server je vhodné na serveri vytvoriť nový HTTP token a index. Vygenerovaný token je: 10191a7c-9047-4b81-9233-9c8433a66a5e. Požiadavka na zobrazenie dát vyzerá nasledovne:

```
source="http:snort_alerts" (index="snort_alerts")
```

logstash.conf

```
input {
  beats{
   port => 5044
   type => "filebeats"
   }
}

output {
if [type] == "filebeats" {
      http {
        content_type => "application/json"
        http_method => "post"
        url => "http://192.168.2.5:8088/services/collector/raw"
        headers => ['Authorization', 'Splunk
10191a7c-9047-4b81-9233-9c8433a66a5e']
        }
    }
}
```

Formát dát zobrazených v Splunk rozhraní vyzerá nasledovne:



Na zobrazenie len tých log. správ, ktoré v sebe obsahujú reťazec “ICMP” je možné použiť požiadavku:

```
( index="snort_alerts" ) message="*ICMP*"
```

Všeobecný prístup k zberu dát

Use Case 1: Syslog dáta zo spravovaného prostredia

- Z každého sieťového zariadenia je možné priamo odosielať syslog dáta na Splunk server (logy rozoznateľné cez `session-id`).
- Z každého servera je možné priamo odosielať syslog dáta na Splunk server.
- Syslog dáta je tiež možné odosielať z akéhokoľvek zariadenia na Logstash server, ktorý ich následne odošle cez HTTP protokol na Splunk server.

Use Case 2: Netflow dáta zo spravovaného prostredia

- Netflow dáta (rozoznateľné podľa host atribútu - pridáva sieťové zariadenie) je potrebné, z každého sieťového zariadenia odosielať na logstash server, ktorý ich následne spracuje a prepošle cez HTTP protokol na Splunk server.

Use Case 3: Serverové logy spustených služieb

- Monitorovanie obsahu .log súborov z lokálneho súborového systému do Splunk servera je možné priamo vyhľadaním daných súborov z lokálneho disku.
- Monitorovanie .log súborov z externých serverov je možné kombináciou FileBeat nástroja (definuje zdrojové .log súbory a odošle ich do Logstash nástroja) a Logstash nástroja (prijaté .log súbory priamo prepošle na HTTP endpoint Splunk servera). Oba nástroje umožňujú pridať ID odosielajúceho zariadenia.

Use Case 4: Aplikačné logy z aplikácii

- V každej aplikácii je možné naprogramovať odosielanie všetkých log. správ na Splunk endpoint
- Ak aplikácia ukladá logy do špecifického súboru, tak jeho obsah je možné nástrojmi Filebeat alebo Logstash preposlať na Splunk server.

Simulácia útokov

Útok hrubou silou na prihlasovací formulár webovej stránky

Zdrojový kód skúšajúci prihlásenie na stránku využitím rôznych mien a hesiel môže vyzerať nasledovne:

```
import requests

def test_login(username, password):
       url = "http://localhost:5000/login"
       data = "username={}&password={}".format(username, password)
       headers = {
       "Content-Type": "application/x-www-form-urlencoded"
       }

       response = requests.request("POST", url, headers=headers, data=data)
       print("Status code:", response.status_code)
       if response.status_code == 200:
       print("Username: {}, Password: {}".format(username, password))
```

```
def simulated_dos(attempts, u_name, u_pass):
	for i in range(0, attempts):
	for name in u_name:
		for passw in u_pass:
		test_login(name, passw)

if __name__ == "__main__":
	u_name = ["admin", "user", "test", "CNL", "root", "cisco"]
	u_pass = ["adminpass", "userpass", "testpass", "cnlpass", "rootpass", "class"]

	simulated_dos(3, u_name, u_pass)
```

Útok hrubou silou na službu SSH

Útok hrubou silou na pripojenie sa na službu SSH je možné vykonať využitím nástroja **Hydra**.
Inštalácia nástroja je nasledovná:

```
sudo apt update
sudo apt install hydra
```

Použitie uvedeného nástroja vyžaduje nasledujúce nastavenia:

1. Vytvorenie súboru obsahujúceho používateľské mená:

```
nano users.txt

admin
cisco
test
ssh
user
```

2. Vytvorenie súboru obsahujúceho používateľské heslá:

```
nano password.txt


cisco
cisco123
cisco123!
jahoda
test
```

3. Spustenie nástroja Hydra je možné príkazom:

```
hydra -L users.txt -P password.txt ssh://192.168.2.5
```

Výstupom programu je nájdená kombinácia prihlasovacích údajov v tvare meno:heslo.

DoS útok využitím protokolu ICMP

Nástroj ping je možné použiť na generovanie istého počtu ICMP správ. Zároveň je možné nastaviť frekvenciu odosielania dát. Možný tvar terminálového príkazu je:

```
sudo ping -c 10 -f 192.168.2.5
```

Prepínač -f spôsobí odoslanie ICMP správy 100x za sekundu.

Prieskumnícky útok skenovaním portov

Skenovanie otvorených portov je možné vykonať nástrojom nmap nasledovne:

```
nmap -p 1-1024 192.168.2.5
nmap -sT 192.168.2.5
```

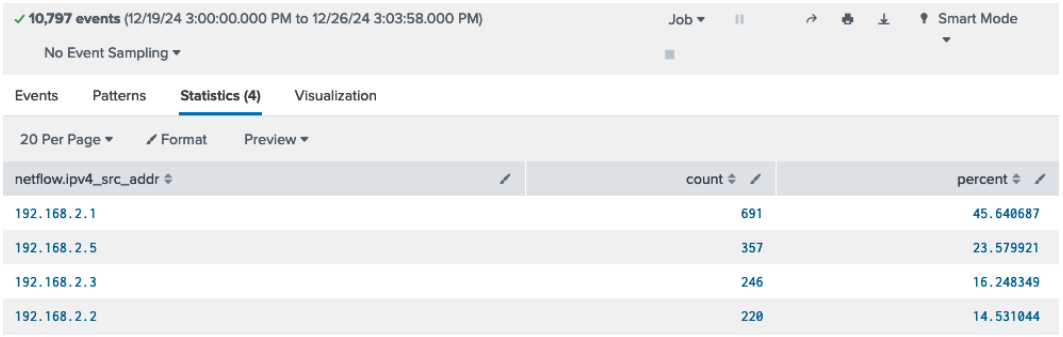
Vyhľadávanie v Splunk-u

Na základe zozbieraných dát je možné vykonať vyhľadávanie rôznych štatistík a údajov za účelom identifikácie komunikujúcich strán, analýzy prenášanej prevádzky a bezpečnostných incidentov.

Identifikácia TOP komunikujúcich strán

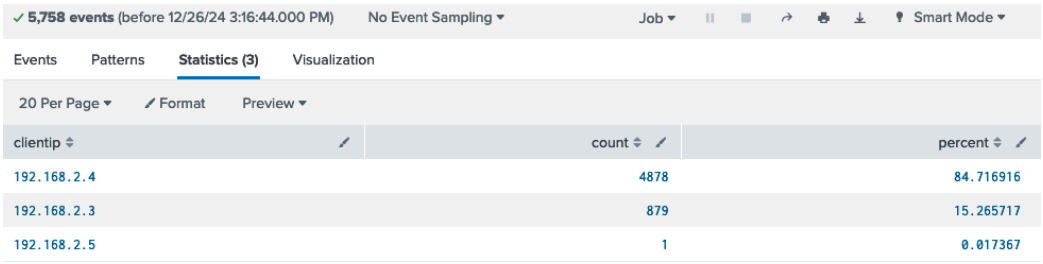
- Netflow
  - Identifikácia najčastejšie vyskytujúcich sa zdrojových a cieľových adries

```
index="netflow_index" | top limit=10
netflow.ipv4_src_addr
index="netflow_index" | top limit=10
netflow.ipv4_dst_addr
```



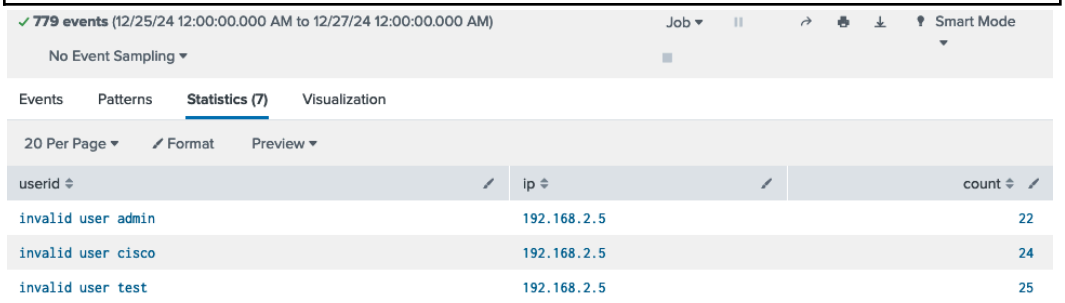
- Nginx
  - Identifikácia zdrojových adries pre webovú komunikáciu

```
index="nginx_access_log" | top limit=10 clientip
```



- SSH access log
  - Identifikácia používateľských mien s chybným prihlásením:

```
index="ssh_auth_log" | rex "Failed password for
(?<userid>.*) from (?<ip>.*) port"
| stats count by userid ip
```



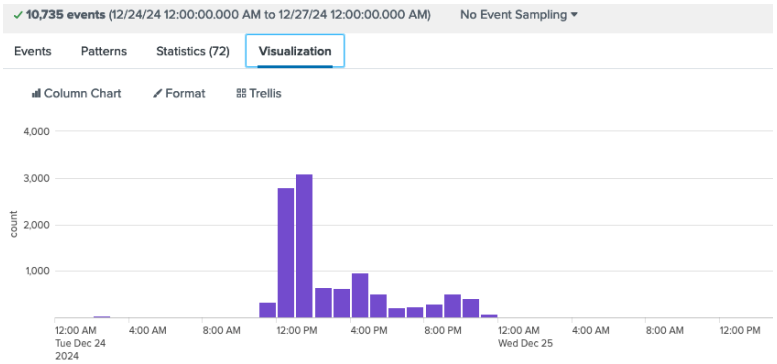
  - Len neexistujúci používatelia:

```
index="ssh_auth_log" | rex "Failed password for invalid
user (?<userid>.*) from" | stats count by userid
```

Analýza sieťovej prevádzky a bezpečnostných incidentov

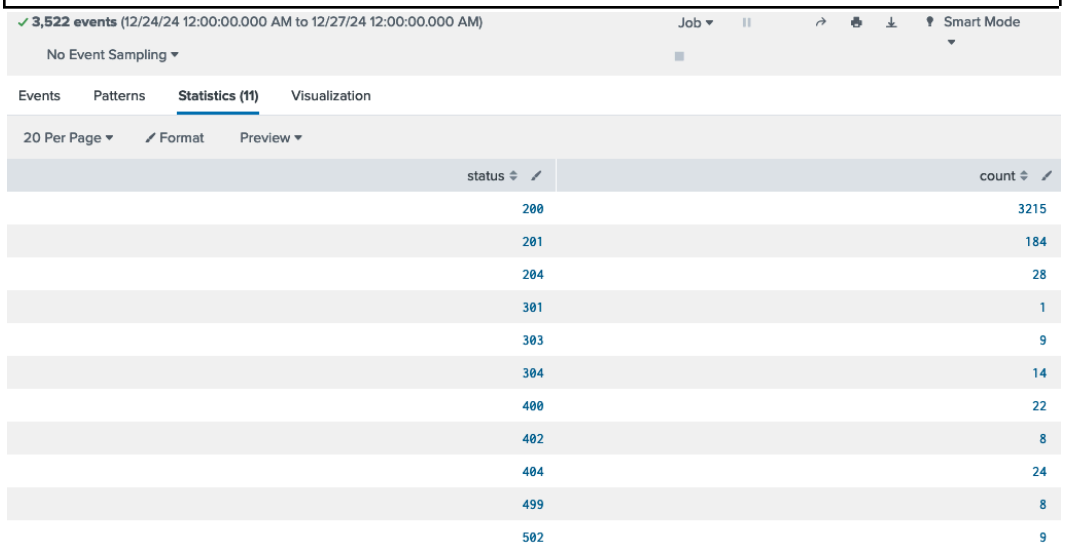
- Netflow
  - Vizualizácia vzorov v komunikácii v rámci zvoleného časového intervalu využitím čiarového diagramu (Štatistiky prijatých dát za každú hodinu):

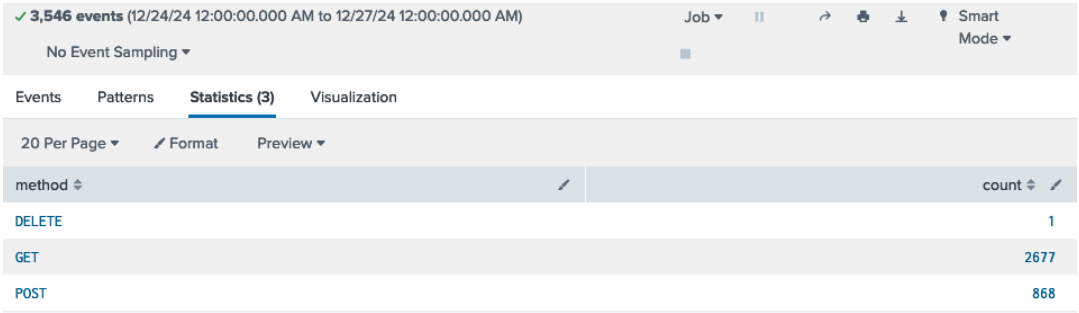
```
index="netflow_index" | timechart span=1h count
```



- Nginx
  - Analýza HTTP požiadaviek podľa použitej návratovej hodnoty a metódy:

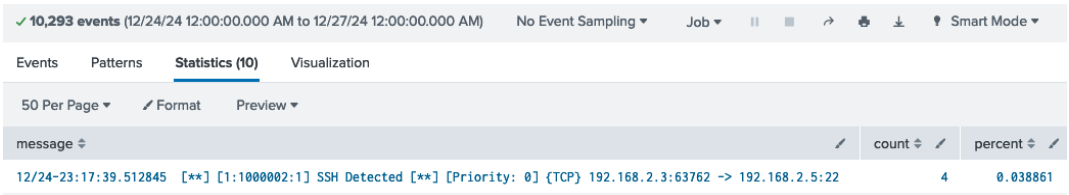
```
index="nginx_access_log" | stats count by status
index="nginx_access_log" | stats count by method
```





- Snort
  - Identifikácia najčastejšieho typu hlásení / vzorov

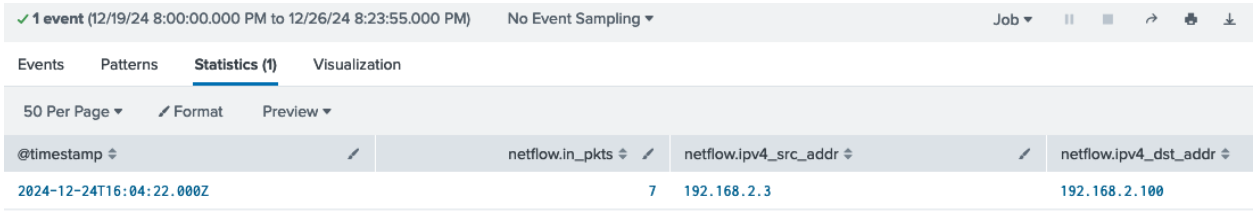
```
index=snort | top limit=10 message
```



Korelácia udalostí

Vyhľadávanie použitím viacerých tabuliek umožní koreláciu udalostí. Napr. je možné zobraziť len také štatistické informácie o dátových tokoch, z Netflow indexu, ktorých cieľová adresa je náš smerovač a zdrojová adresa sa zároveň nachádza ako zdrojová adresa v Nginx indexe. Takýto filter ukáže používateľa, ktorý pristupoval na webovú službu a zároveň pristupoval na sieťové zariadenie. Požiadavka pre zobrazenie časovej značky, množstva paketov, zdrojovej a cieľovej IPv4 adresy vyzerá nasledovne:

```
index=netflow_index netflow.ipv4_dst_addr=192.168.2.100 [search
index="nginx_access_log" | top limit=10 clientip | table clientip |
rename clientip as  netflow.ipv4_src_addr]
| table @timestamp, netflow.in_pkts, netflow.ipv4_src_addr,
netflow.ipv4_dst_addr
```



Vysvetlenie požiadavky

- `index=netflow_index netflow.ipv4_dst_addr=192.168.2.100:`

Tento výraz vyberá len tie záznamy z indexu "netflow_index", kde je cieľová IP adresa ("netflow.ipv4_dst_addr") rovná "192.168.2.100". To znamená, že sa zameriava na všetky sieťové toky smerujúce na túto konkrétnu IP adresu.

- `[search index="nginx_access_log" | top limit=10 clientip | table clientip | rename clientip as netflow.ipv4_src_addr]`

Táto podpožiadavka vykonáva nasledovné:.

- `index="nginx_access_log"`: Vyberá záznamy z indexu "nginx_access_log".
- `| top limit=10 clientip`: Vyberá 10 najčastejších IP adries klientov.
- `| table clientip`: Zobrazí iba pole "clientip" (IP adresa klienta).
- `| rename clientip as  netflow.ipv4_src_addr`: Pomenuje pole "clientip" na "netflow.ipv4_src_addr" pre konzistenciu s hlavným vyhľadávaním.
- `| table @timestamp, netflow.in_pkts, netflow.ipv4_src_addr, netflow.ipv4_dst_addr:` Tento výraz zobrazuje požadované polia v prehľadnej tabuľke:

Tvorba Dashboard-ov v Splunk-u

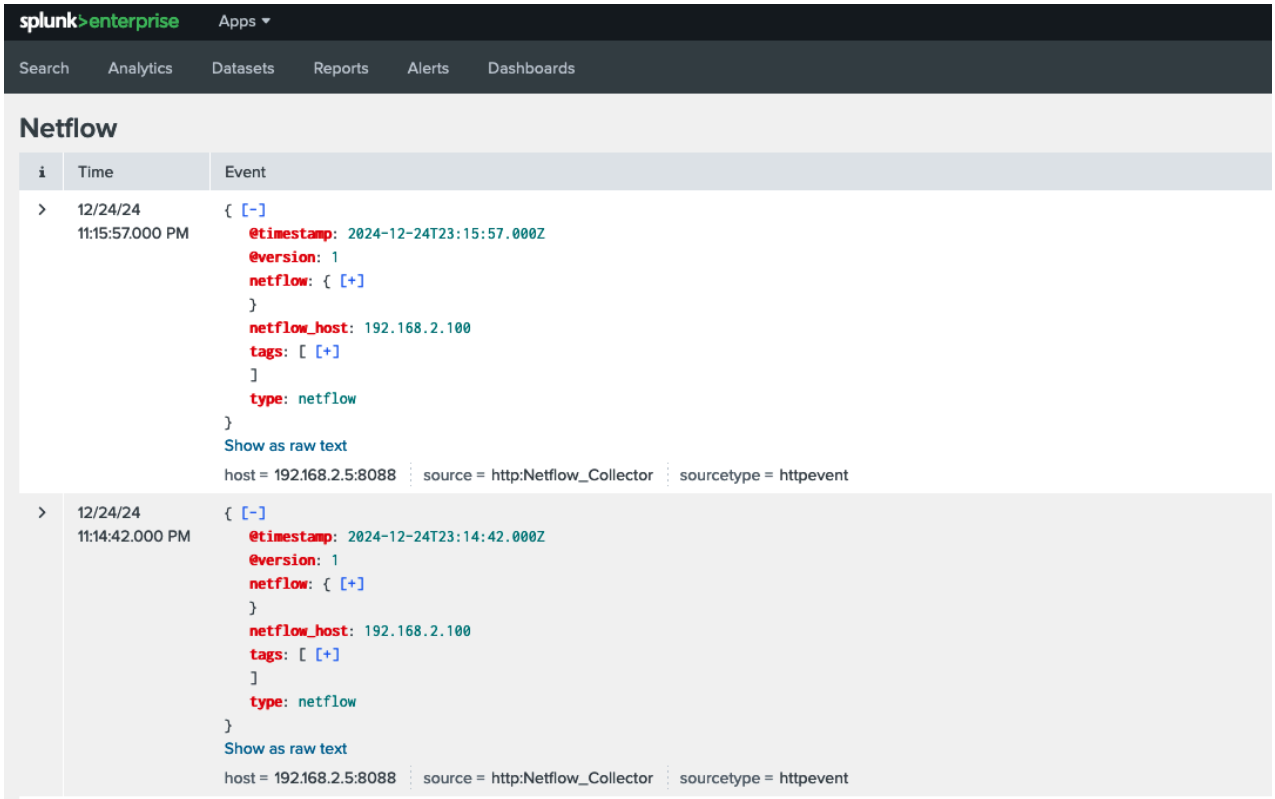
Pre rýchle zobrazenie dát, ktoré je potrebne často kontrolovať je vhodné vytvárať riadiace panely (angl. Dashboards). V rámci Splunk-u je možné riadiaci panel vytvoriť v časti:

- **Search&Reporting - Dashboards - Create New Dashboard**
  - **Dashboard Title: Netflow**
  - **Classic Dashboard**

Po vyplnení potrebných polí sa zobrazí pracovná plocha, do ktorej je možné tlačidlom **Add Panel** vkladať elementy umožňujúce zobrazenie potrebných informácii. Podľa typu požadovanej vizualizácie je možné zvoliť z:

- Events
- Statistics Table
- Line Chart
- Single Value
- …

Následne po voľbe elementu už len stačí zvoliť časový interval a požiadavku pre vyhľadávanie. Uloženie daného elementu sa vykonáva stlačením tlačidla **Add to Dashboard**. Event dashboard zobrazujúci výsledok požiadavky [index=netflow_index netflow.ipv4_src_addr = 192.168.2.3]:

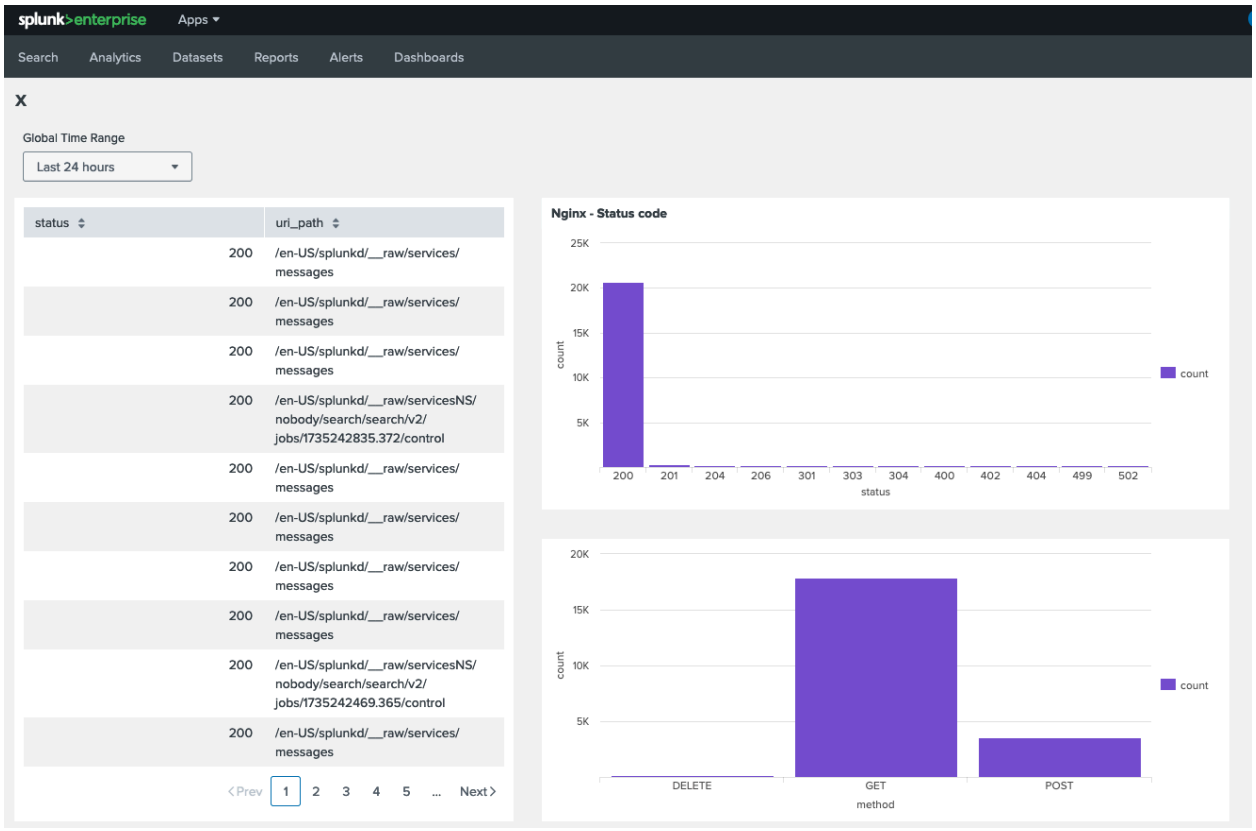


Pre vytváranie riadiacich panelov, ktoré budú obsahovať viacero elementov je potrebné zvoliť:

- Dashboard Studiu - Absolute

Nasledujúci riadiaci panel bol vytvorený dotazmi:

1. index="nginx_access_log" | table status, uri_path
2. index="nginx_access_log" | stats count by status
3. index="nginx_access_log" | stats count by method



Vytvorené riadiace panely je možné stiahnuť vo formátoch CSV, PDF, PNG.