

# Segurança em Redes de Comunicações

## Report 2

Universidade de Aveiro

Bruno Silva (97931)brunosilva16@ua.pt  
Marta Oliveira (97613) marta.alex@ua.pt



universidade  
de aveiro

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Análise do comportamento típico da rede</b>	<b>2</b>
<b>3</b>	<b>Regras SIEM e testes</b>	<b>7</b>

# Capítulo 1

## Introdução

O presente relatório tem como objetivo descrever a resolução do guião *Security in Communications Networks* desenvolvido no âmbito da unidade curricular de Segurança em Redes de Comunicações.

## Capítulo 2

# Análise do comportamento típico da rede

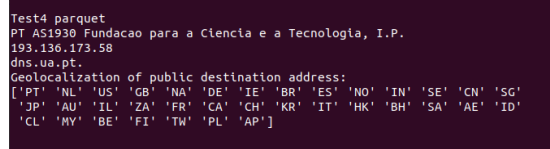
O nosso script, *Script.py*, analisa tráfego de rede de diferentes maneiras para identificar comportamentos anômalos mas para conseguir identificar é necessário em primeiro analisar o ficheiro que define o comportamento normal da rede e dos dispositivos, *data4.parquet*.

No nosso script Python realizamos uma série de operações para analisar os dados de tráfego de rede e identificar padrões de comportamento típicos.

Começamos por identificar a geolocalização de endereços de destinos públicos com a biblioteca GeoIP.

```
1 cc = data[bpublic]['dst_ip'].apply(lambda y: gi.  
    country_code_by_addr(y)).to_frame(name='cc')  
2 print('Geolocalization of public destination address:  
    ')  
3 print(cc)  
4 print("\n")
```

Esta foi a lista que observámos:



```
Test4 parquet  
PT AS1930 Fundacao para a Ciencia e a Tecnologia, I.P.  
193.136.173.58  
dns.ua.pt.  
Geolocalization of public destination address:  
['PT' 'NL' 'US' 'GB' 'NA' 'DE' 'IE' 'BR' 'ES' 'NO' 'IN' 'SE' 'CN' 'SG'  
'JP' 'AU' 'IL' 'ZA' 'FR' 'CA' 'CH' 'KR' 'IT' 'HK' 'BH' 'SA' 'AE' 'ID'  
'CL' 'MY' 'BE' 'FI' 'TW' 'PL' 'AP']
```

Figura 2.1: Geolocalizações

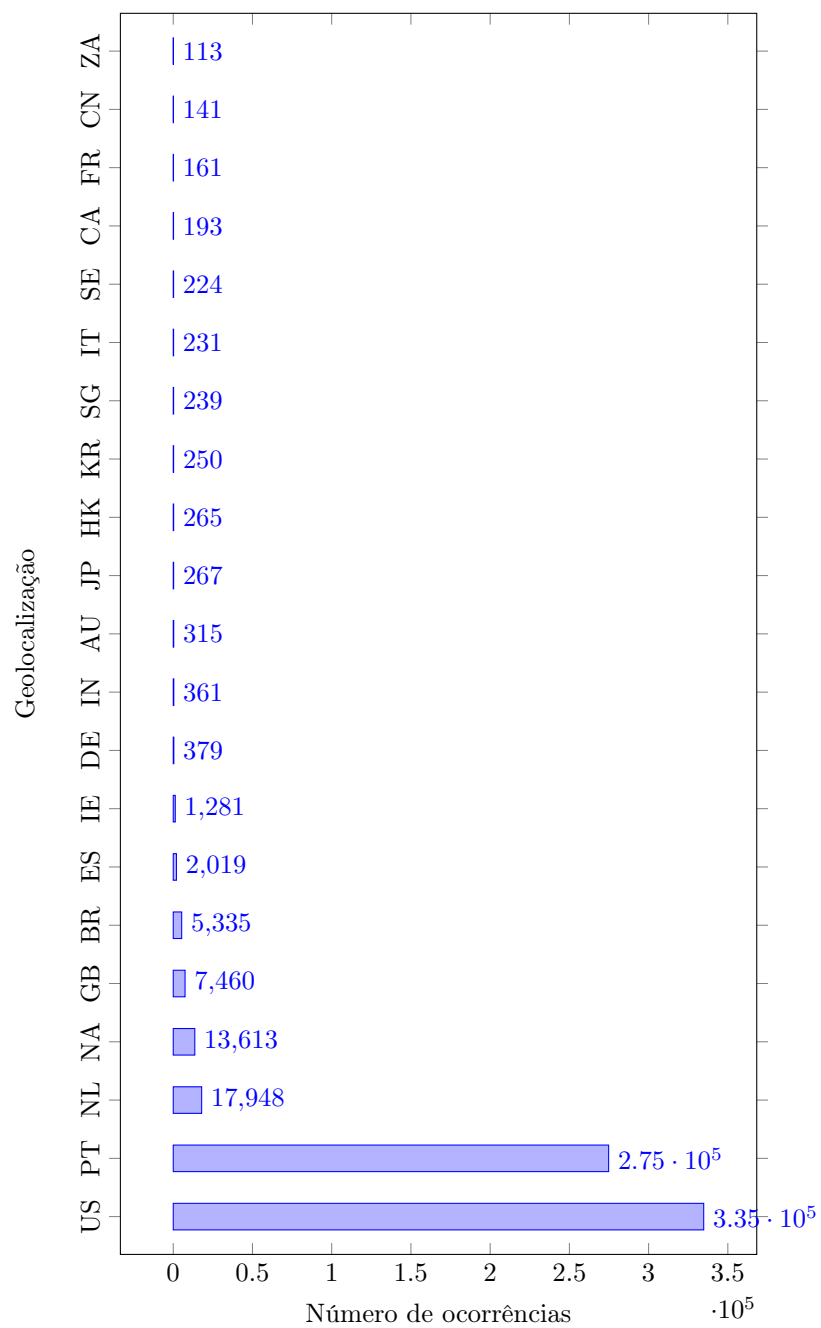


Figura 2.2: Número de ocorrências que cada geolocalização foi identificada

De seguida identificámos quais os protocolos mais usados no tráfego de rede, para isso utilizámos a função `unique()` para os identificar. Neste caso, os protocolos mais utilizados são o TCP e o UDP:

```
1 expected_protocols = data['proto'].unique()
2 print('expected protocols:', expected_protocols)
```

```
Test4 parquet
PT AS1930 Fundacao para a Ciencia e a Tecnologia, I.P.
193.136.173.58
dns.ua.pt.
expected protocols: ['udp' 'tcp']
```

Figura 2.3: Protocols used in Data4.parquet

O script também analisa quais são as portas mais usadas no tráfego de rede. Descobrimos que as portas mais utilizadas são as portas 443 e 53.

```
1 port_counts = data['port'].value_counts()
```

Uma análise igualmente importante foi a análise temporal do tráfego. Para isso, convertemos os timestamps para segundos e criámos uma nova coluna com a hora do dia correspondente. De seguida, contamos o número de eventos de tráfego por hora e visualizamos esses dados num histograma.

```
1 data['timestamp_seconds'] = data['timestamp'] / 100.0
2
3 data['hour'] = pd.to_datetime(data['timestamp_seconds'],
4                               unit='s').dt.hour
5
6 # histograma para ver a distribui o de eventos por
7   hora
8 hourly_counts = data['hour'].value_counts().sort_index()
9
10 # Visualiza o histograma
11 plt.figure(figsize=(10, 6))
12 plt.bar(hourly_counts.index, hourly_counts.values)
13 plt.xlabel('Hour of Day')
14 plt.ylabel('Number of Events')
15 plt.title('Distribution of Events in a Day')
16 plt.xticks(range(24)) # Este comando far com que o
17   eixo x mostre todas as horas do dia
18 plt.show()
```

Analisando o histograma identificamos facilmente que o tráfego é mais intenso desde as 8h às 17h:

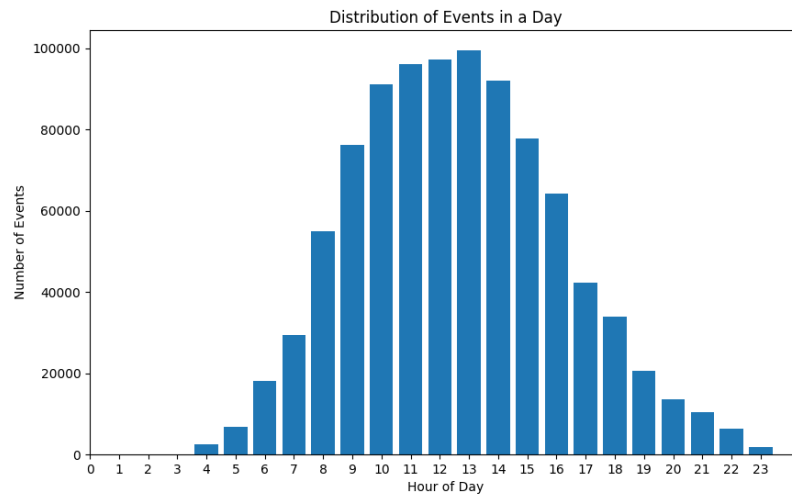


Figura 2.4: Events in a day

O script, adicionalmente, analisa os IPs de origem mais comuns na rede. Descobrimos que a maioria do tráfego tem origem na rede 192.168.104.0/24. Desta forma, assumimos que esta é a rede IPV4 privada da empresa.

```
1 # Use value_counts na coluna de IP de origem
2 ip_counts = data['src_ip'].value_counts()
3 print("\n")
4 # Imprima o resultado
5 print(ip_counts)
```

Igualmente, também verificámos os IP's de destino mais usados e as suas organizações respectivas:

```
Most common private dest ip's: dst_ip
192.168.104.224 56240
192.168.104.231 55044
192.168.104.234 41379
192.168.104.230 41001
192.168.104.222 40516
192.168.104.232 40028
Name: count, dtype: int64
```

```
Most common public destinations:
```

IP	Origin	Organization	Domain
142.250.200.68	United States	AS15169 GOOGLE	mad07s24-in-f4.1e100.net.
172.217.17.14	United States	AS15169 GOOGLE	mad07s09-in-f14.1e100.net.
157.240.212.35	Portugal	AS32934 FACEBOOK	edge-star-mini-shv-01-lis1.facebook.com.
213.13.146.142	Portugal	AS3243 Servicos De Comunicacoes E Multimedia S.A.	sapo.pt.
157.240.212.174	Portugal	AS32934 FACEBOOK	instagram-p42-shv-01-lis1.fbcdn.net.
216.58.215.131	United States	AS15169 GOOGLE	mad41s04-in-f3.1e100.net.
88.157.217.145	Portugal	AS2860 Nos Comunicacoes, S.A.	a88-157-217-145.static.cpe.netcabo.pt.
104.244.42.193	United States	AS13414 TWITTER	
193.126.240.146	Portugal	AS2860 Nos Comunicacoes, S.A.	websites.tolnegocios.com.
204.79.197.212	United States	AS8068 MICROSOFT-CORP-MSN-AS-BLOCK	a-0010.a-msedge.net.

Figura 2.5: Common IP'S

É possível observar que os domínios mais acedidos são os da Google, estando também presente o Facebook , o Twitter, entre outros.

Desta forma, conseguimos concluir algumas informações da rede, resumidamente:

1. O tráfego geralmente usa um conjunto esperado de protocolos como udp e tcp e consequentemente dos portos 443 e 53.
2. O tráfego geralmente provém da rede 192.168.104.0/24.
3. O tráfego geralmente incide mais das 8h até às 17h.
4. As geolocalizações mais comuns são a US e PT.
5. Os serviços mais acedidos são da Google.



## Capítulo 3

# Regras SIEM e testes

De seguida, vamos criar regras SIEM que são projetadas para detetar eventos de segurança relevantes, como tentativas de invasão, atividades suspeitas ou comportamentos maliciosos e com isso iremos analisar o ficheiro test4.parquet onde se encontra os dados da empresa de um dia.

### Regra 1

Nesta regra temos como objetivo de identificar a presença de um volume substancial de bytes de download e upload nos fluxos, o que pode ser indicativo de atividades relacionadas a botnets e CC (Command and Control). É aplicado um limiar que considera um acréscimo superior ao dobro da linha de base. Essa abordagem permite identificar de forma mais eficaz atividades anômalas e suspeitas relacionadas a tráfego de rede.

```
1 threshold = 2
2 test_total_traffic = test_data.groupby('src_ip').apply
   (lambda x: x['up_bytes'].sum() + x['down_bytes'].
   sum())
3 sudden_increase = monitor_traffic(test_total_traffic,
   threshold)
4 anomalies = sudden_increase[sudden_increase['Anomaly',
   ].fillna(False)]
5 if not anomalies.empty:
6     ip_list = anomalies.index.tolist()
7     print("The following IPs exceeded the expected
   volume flow:")
8     for ip in ip_list:
9         print(ip)
10 else:
11     print("None ip exceeded the expected volume flow")
```

Na captura conseguimos verificar os ip's que excedem esse limite imposto.

```
The following IPs exceeded the expected volume flow:
192.168.104.108
192.168.104.11
192.168.104.111
192.168.104.115
192.168.104.121
192.168.104.123
192.168.104.151
192.168.104.155
192.168.104.156
192.168.104.161
192.168.104.170
192.168.104.174
192.168.104.176
192.168.104.189
192.168.104.193
192.168.104.200
192.168.104.203
192.168.104.206
192.168.104.207
192.168.104.209
192.168.104.29
192.168.104.38
192.168.104.43
192.168.104.44
192.168.104.46
192.168.104.47
192.168.104.48
192.168.104.71
192.168.104.72
192.168.104.74
192.168.104.84
192.168.104.91
192.168.104.92
192.168.104.97
192.168.104.99
```

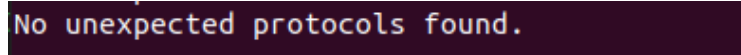
Figura 3.1: Events in a day

## Regra 2

Tem como objetivo identificar o uso de protocolos de rede que não são esperados, isto é UDP e TCP.

```
1 unexpected_protocols = test_data[~test_data['proto'].  
    isin(expected_protocols)]  
2 if not unexpected_protocols.empty:  
3     print("Anomalies due to unexpected protocols usage  
    :")  
4     for ip in unexpected_protocols['src_ip'].unique():  
5         print(f"IP: {ip}, Protocol: {  
            unexpected_protocols['proto'].unique()}")
```

No entanto, não foram encontradas anomalias nesse aspecto.



```
No unexpected protocols found.
```

Figura 3.2: Protocols

## Regra 3

Esta regra identifica qualquer tráfego que não seja proveniente da rede '192.168.104.0/24' ou que utilize portas não autorizadas é considerado uma anomalia.

Os ports que identificámos como não autorizados foram o 20 e 21 (FTP) pois é considerado um protocolo inseguro dado que não usa criptografia.

Também a porta 22, o protocolo SSH é usado usado para acesso remoto seguro a servidores e é frequentemente utilizado para ataques Ddos.

A porta 23 pois o Telnet é um protocolo que permite a comunicação interativa com outro host em uma rede. No entanto, ele é inseguro e desatualizado.

Por último, a porta 80 pois é do protocolo HTTP e o seu trafego não é seguro.

```

1 #Ip's autorizados
2 authorized_ips = [f'192.168.104.{i}' for i in range
   (256)]
3 # Cria uma lista vazia para os IPs n autorizados
4 unauthorized_ips = []
5 for ip in data['src_ip'].unique():
6     if ip not in authorized_ips:
7         unauthorized_ips.append(ip)
8 print(unauthorized_ips)
9 unauthorized_ports = [80,22,20,21,23]
10 detect_unauthorized_traffic(test_data,
    unauthorized_ips, unauthorized_ports)

```

Para esta regra usámos a seguinte a função:

```

1 def detect_unauthorized_traffic(data,
   unauthorized_destinations, unauthorized_ports):
2     unauthorized_traffic = data[(data['dst_ip'].isin(
        unauthorized_destinations)) | (data['port'].
        isin(unauthorized_ports))]
3     if not unauthorized_traffic.empty:
4         print("Anomalies due to unauthorized traffic
           to specific destinations and ports:")
5         print(unauthorized_traffic)
6     else:
7         print("No unauthorized traffic to specific
           destinations and ports found.")

```

Como já tínhamos comprovado no ponto anterior, não conseguimos encontrar nenhum uso de protocolo invulgar ou e além disso também não encontramos com IP's com origens fora da subrede da empresa.

## Regra 4

Tem como objetivo detectar fluxos de rede que ocorrem fora de uma janela de tempo normal e além disso com um fluxo maior que o suposto. Assumimos que o horário normal está compreendido entre as 8h e as 17h (como comprovado anteriormente) e dessa forma se existir tráfego fora dessa janela temporal e com uma quantidade de tráfego maior que um valor predefinido, no caso escolhemos 100000000 bytes, uma anomalia é detetada.

```

1 avgDown = test_data.groupby('src_ip')['down_bytes'].
    transform('mean')
2 avgUp = test_data.groupby('src_ip')['up_bytes'].
    transform('mean')
3 test_data['timestamp'] = pd.to_datetime(test_data['
    timestamp'], unit='s', origin='unix')
4 high_traffic_query = "((up_bytes + down_bytes) >
    100000000) and (timestamp.dt.hour < 8 or timestamp.
    dt.hour > 17)"
5 high_traffic_results = test_data.query(
    high_traffic_query)
6 high_traffic_results['timestamp'] =
    high_traffic_results['timestamp'].dt.strftime('%H:%
    M:%S')
7 print("High Network Traffic during specific hours:")
8 print(high_traffic_results)

```

Encontrámos as seguintes anomalias:

High Network Traffic during specific hours:							
	timestamp	src_ip	dst_ip	proto	port	up_bytes	down_bytes
index							
907240	21:54:43	192.168.104.43	142.250.184.244	udp	443	108881828	1736691
907241	07:14:52	192.168.104.43	142.250.184.244	udp	443	116776940	1354036
907245	20:35:01	192.168.104.43	142.250.184.244	udp	443	194498765	2875006
907246	05:53:30	192.168.104.43	142.250.184.244	udp	443	265049697	4274864
907248	00:33:54	192.168.104.43	142.250.184.244	udp	443	124569895	1287291
907266	00:25:00	192.168.104.170	142.250.184.223	udp	443	221964849	2580814
907250	19:13:06	192.168.104.43	142.250.184.244	udp	443	513558265	7794324
907251	04:34:07	192.168.104.43	142.250.184.244	udp	443	201986256	2288865
907268	19:04:35	192.168.104.170	142.250.184.223	udp	443	344843969	4111571
907277	19:44:21	192.168.104.71	13.107.42.40	tcp	443	187774177	1626949
907253	23:14:46	192.168.104.43	142.250.184.244	udp	443	151462137	2111793
907278	05:03:30	192.168.104.71	13.107.42.40	tcp	443	127370977	1847195
907271	23:07:15	192.168.104.170	142.250.184.223	udp	443	358408202	3923986
907280	23:40:57	192.168.104.71	13.107.42.40	tcp	443	212982408	2925867
907256	03:13:04	192.168.104.43	142.250.184.244	udp	443	121439716	1085526
907282	18:20:13	192.168.104.71	13.107.42.40	tcp	443	136067831	1345552
907274	03:09:58	192.168.104.170	142.250.184.223	udp	443	156821388	1621010
907258	21:52:54	192.168.104.43	142.250.184.244	udp	443	107004979	1294854
907283	03:40:13	192.168.104.71	13.107.42.40	tcp	443	104956074	1689467
907259	07:14:28	192.168.104.43	142.250.184.244	udp	443	106682799	1359412
907285	22:19:20	192.168.104.71	13.107.42.40	tcp	443	162559428	1451712
907261	01:53:16	192.168.104.43	142.250.184.244	udp	443	160789002	2806491
907286	07:38:24	192.168.104.71	13.107.42.40	tcp	443	255776026	2996007
907263	20:30:36	192.168.104.43	142.250.184.244	udp	443	282896180	2828947
907288	02:16:39	192.168.104.71	13.107.42.40	tcp	443	156524763	1883818
907290	20:58:25	192.168.104.71	13.107.42.40	tcp	443	247803023	3287943
907291	06:17:53	192.168.104.71	13.107.42.40	tcp	443	168446223	1641135
907293	00:58:05	192.168.104.71	13.107.42.40	tcp	443	111708019	1887563
907295	19:37:22	192.168.104.71	13.107.42.40	tcp	443	406749281	5212179
907298	23:40:43	192.168.104.71	13.107.42.40	tcp	443	143955474	1619672
907300	18:19:55	192.168.104.71	13.107.42.40	tcp	443	146533512	1483488
907303	22:19:39	192.168.104.71	13.107.42.40	tcp	443	117090573	1032108
907304	07:41:23	192.168.104.71	13.107.42.40	tcp	443	132003882	1591192

Figura 3.3: Unexpected flows

Podemos observar fluxos com elevado número de Upbytes e downBytes a horas pouco usuais.

## Regra 5

Esta regra tem como objetivo identificar localizações geográficas incomuns com base numa lista de localizações geográficas típicas como já mencionamos

```
1 ##Rule5
2 typical_geolocation = ['PT', 'NL', 'US', 'GB', 'NA', '
   DE', 'IE', 'BR', 'ES',
3 'NO', 'IN', 'SE', 'CN', 'SG',
4 'JP', 'AU', 'IL', 'ZA', 'FR', 'CA', 'CH', 'KR', 'IT',
   'HK', 'BH', 'SA', 'AE', 'ID', 'CL', 'MY', 'BE', 'FI',
   'TW', 'PL', 'AP']
5 unusual_geolocation=detect_unusual_geolocation(
   test_data, typical_geolocation)
```

Para fazermos essa verificação usamos esta função:

```
1 def detect_unusual_geolocation(data,
   typical_geolocation):
2     gi = pygeoip.GeoIP('./GeoIP.dat')
3     unusual_geolocation = data[~data['dst_ip']].apply(
        lambda ip: gi.country_code_by_addr(ip) in
        typical_geolocation)]
4
5     if not unusual_geolocation.empty:
6         print("Anomalies due to unusual geolocation of
           destination IP addresses:")
7         for ip in unusual_geolocation['dst_ip'].unique
           ():
8             geolocation = gi.country_code_by_addr(ip)
9             print(f"IP: {ip}, Geolocation: {
               geolocation}")
10    else:
11        print("No anomalies due to unusual geolocation
           of destination IP addresses.")
```

Obtivemos uma lista ainda exaustiva de ips em geolocalizações não normais.

```
IP: 185.19.21.200, Geolocation: RU  
IP: 91.234.80.136, Geolocation: RU  
IP: 95.141.227.37, Geolocation: RU  
IP: 78.137.109.55, Geolocation: RU  
IP: 185.182.109.217, Geolocation: RU  
IP: 178.17.168.230, Geolocation: MD  
IP: 45.125.4.79, Geolocation: MM  
IP: 37.190.49.154, Geolocation: RU  
IP: 109.108.52.62, Geolocation: RU  
IP: 176.32.178.85, Geolocation: RU  
IP: 176.126.38.8, Geolocation: GR  
IP: 185.19.202.176, Geolocation: RU  
IP: 188.130.169.37, Geolocation: RU  
IP: 217.78.187.32, Geolocation: RU  
IP: 195.95.133.178, Geolocation: RU  
IP: 116.206.137.131, Geolocation: MM  
IP: 194.87.232.133, Geolocation: RU  
IP: 94.137.187.185, Geolocation: GE  
IP: 193.105.11.7, Geolocation: RU  
IP: 193.151.226.51, Geolocation: KG  
IP: 195.200.216.6, Geolocation: RU  
IP: 185.179.84.181, Geolocation: RU  
IP: 46.252.118.63, Geolocation: RU  
IP: 78.18.188.88, Geolocation: RU
```

Figura 3.4: Unusual GeoLocalizations

Os resultados desta análise demonstram que existe comunicações com países não expectáveis, tais como Rússia, Myanmar, Grécia, entre outros.