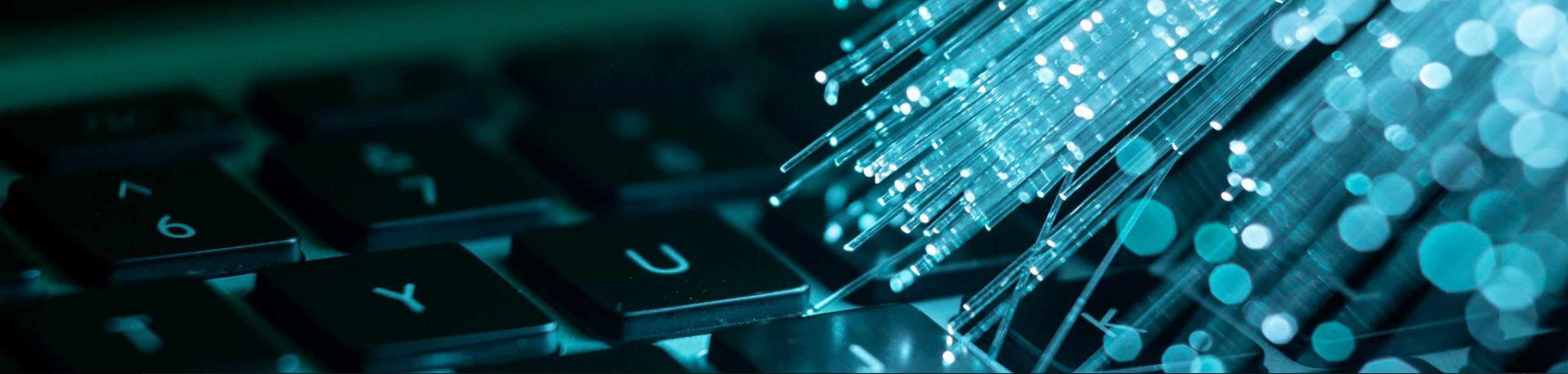


Técnicas de Perceção de Redes DNS Tunneling

João Viegas - 98372
Tiago Marques - 98459

Janeiro 2023 | 2nd Presentation

The Problem

DNS Tunneling presents a serious threat to business and security as a whole:

1. Is a very flexible attack, can be used for C&C, exfiltration and others

2. Is hard to be detected because of the DNS protocol nature

Our focus

- We will monitor the network traffic patterns of known good users to understand how they behave.
- Using that behaviour we will try to establish a profile.
- Based on that profile we will try to distinguish the normal traffic from anomalous traffic

Used Datasets

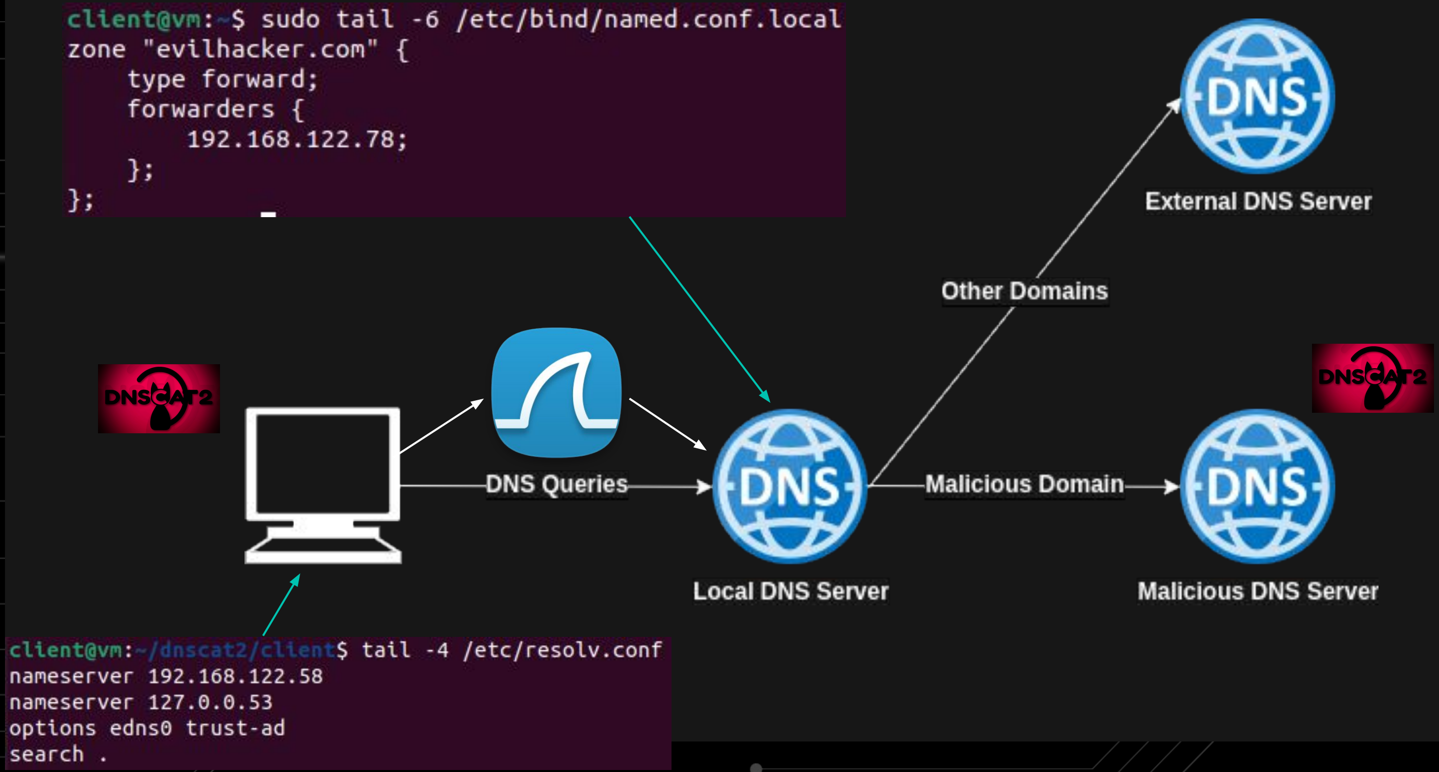
Not malicious Source

The non-malicious datasets should be obtained from an IEEE dataset w/10 days of DNS traffic put it has some heird communications.

Malicious Source

Our malicious dataset will be generated in-house through the use of virtualization (VMs), DNScat2 (DNS Tunneling Software) and bind9(DNS software).

Test Scenario



```
client@vm:~$ sudo tail -6 /etc/bind/named.conf.local
zone "evilhacker.com" {
    type forward;
    forwarders {
        192.168.122.78;
    };
};
```

```
client@vm:~/dnscat2/client$ tail -4 /etc/resolv.conf
nameserver 192.168.122.58
nameserver 127.0.0.53
options edns0 trust-ad
search .
```

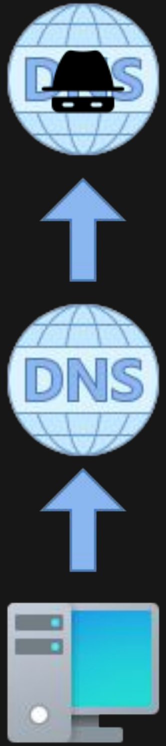
General approach of our attack

1 º - Attacker establishes a DNS connection with a client.

2 º - Attacker opens a shell session through DNS

3 º - Commands like pwd, ls, cat, echo are made to search, read and modify files in the client.

4 º - A selected file is exfiltrated.



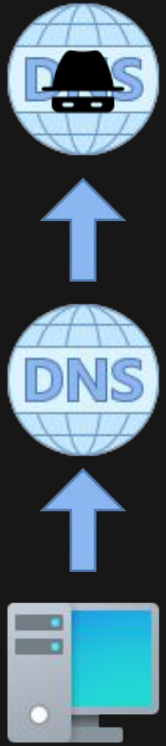
Our Types of Attacks

Our attackers used DNS tunneling to send commands via DNS.

- Attack 1 uses DNS tunneling with 3 seconds delay.
- Attack 2 uses DNS tunneling with 3 seconds steady delay.
- Attack 3 uses DNS tunneling with 5 seconds delay.
- Attack 4 uses DNS tunneling with 5 seconds steady delay.

Delay -> Maximum delay between packets

Steady -> The system consistently waits for the specified delay before transmitting the next message



Data Processing

- Collect raw packet data with a sampling period of 5 and 10 seconds.
- Filter data to allow only DNS or Secure DNS packets.
- Detect anomalous user behaviour

- Observation Windows :
  - Size of and 10 minutes
  - Sliding 1 every minute

Extracted Metrics

- To extract these metrics a custom application was written using python and pyshark library

- Metrics:

  - Number of DNS Query packets
  - Number of upload bytes
  - Number of DNS Reply Packets
  - Number of upload bytes
  - Sum time between a DNS Reply and the last DNS Query packet
  - Sum time between two sequential DNS Query packets
  - Sum time between two sequential DNS packets
  - Min time between two sequential DNS packets
  - Max time between two sequential DNS packets

Extracted Features

- To extract these metrics a custom application was written using python and numpy library

- Number of DNS Query / DNS Reply packets:
  - Mean, Median, Standard Deviation, Variance
  - 90th, 95th, 98th, 99th percentiles
- Ratio Upload Bytes/DNS Query and Download Bytes/DNS Reply:
  - Mean, Median, Standard Deviation, Variance
  - 90th, 95th, 98th, 99th percentiles
- Silence periods DNS Query/Reply (threshold = 4)
  - Mean, Median, Standard Deviation, Variance
  - 90th, 95th, 98th, 99th percentiles

Extracted Features

- **Sum of time between DNS response time / DNS Queries / DNS Packets:**
  - **Mean, Median, Standard Deviation, Variance**
  - **90th, 95th, 98th, 99th percentiles**
- **Min/Max Time between DNS Packets:**
  - **Mean, Median, Standard Deviation, Variance**
  - **90th, 95th, 98th, 99th percentiles**
- **Periodicity:**
  - **Sum time between DNS Queries / DNS Reply**
  - **Sum time between DNS Queries**
  - **Sum time between DNS packets**
  - **Min time between DNS packets**
  - **Max time between DNS packets**

Extracted Features

- Covariance:
    - DNS Query and Upload bytes
    - DNS Reply and Upload bytes
    - Min and Max time between DNS packets
    - Sum of time between 2 DNS Queries and Max time between DNS packets

Features Processing

- Features are split into 2 dataset
  - Training, containing 75% of all normal features
  - Testing, containing the other 25% of all normal features and all anomaly features

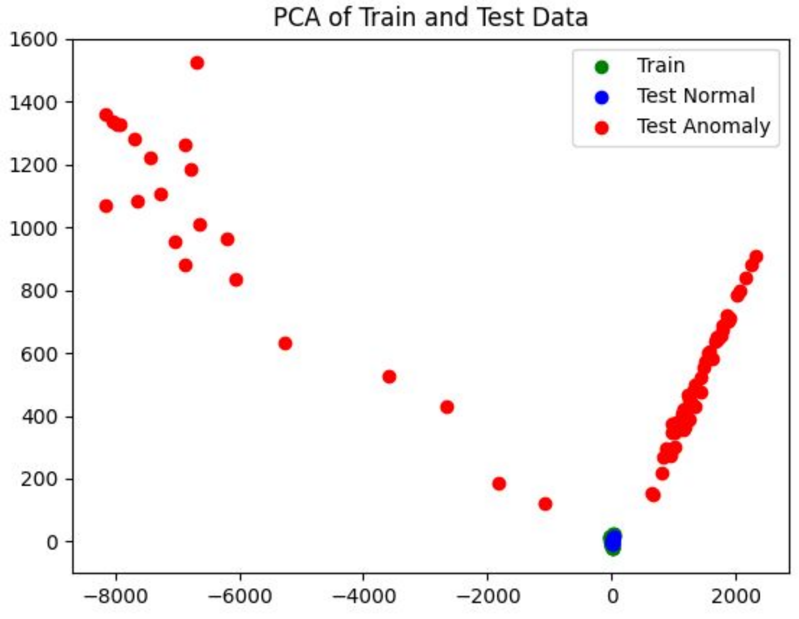
- The data is scaled using a standard scaler
  - Fitted on the training data
  - It is recommended for support vector machines

- PCA is performed to reduce the number of features

DNS Attack 1

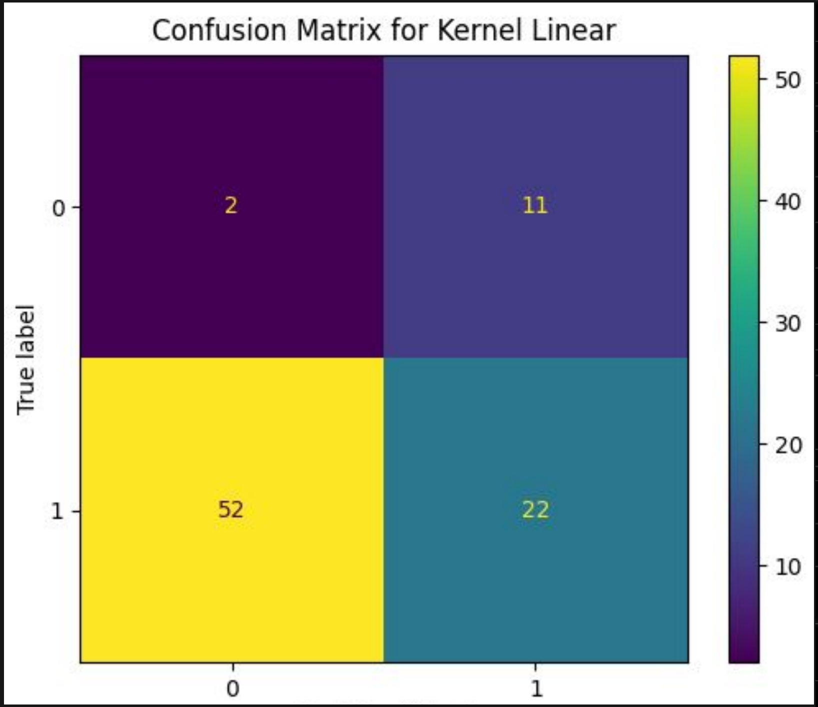
Through DNS this attacker sends responses with injected sh commands with steady non active for 3 seconds.

- Window size : 10 minutes
- Sliding window every : 1 minute
- Sampling period : 5 seconds



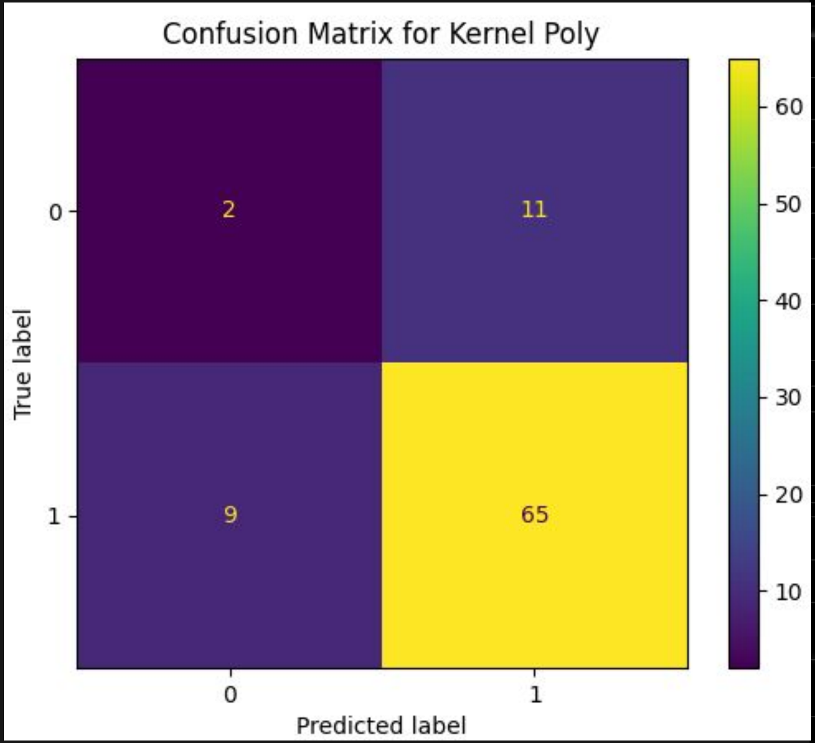
DNS Attack 1 – Kernel Linear results

- Accuracy: 25.58%
- Precision: 66.66%
- Recall: 29.7%
- F-1: 41.1%



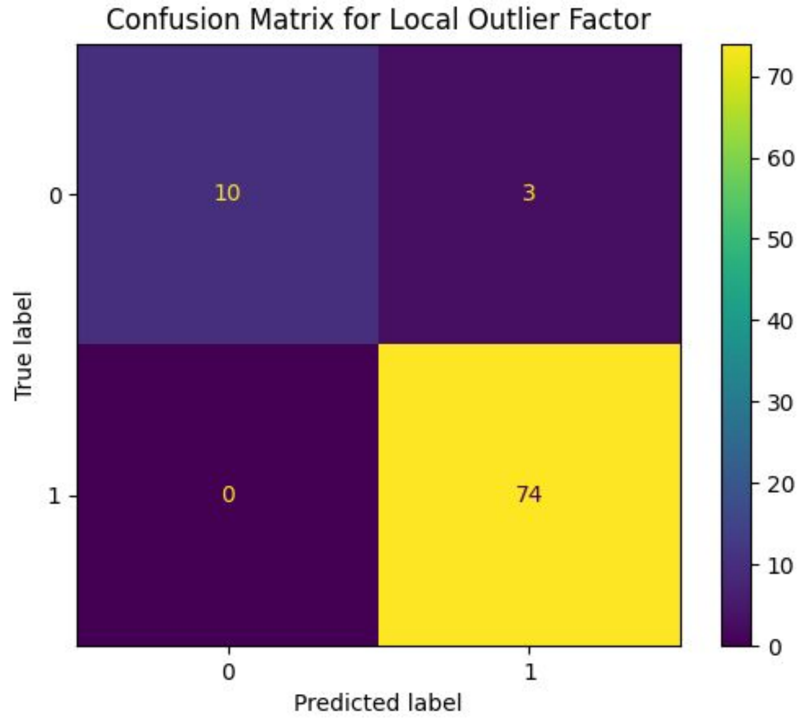
DNS Attack 1 – Kernel Poly results

- **Accuracy: 77.01%**
- **Precision: 85.52%**
- **Recall: 87.83%**
- **F-1: 86.66%**



DNS Attack 1 – LocalOutlierFactor results

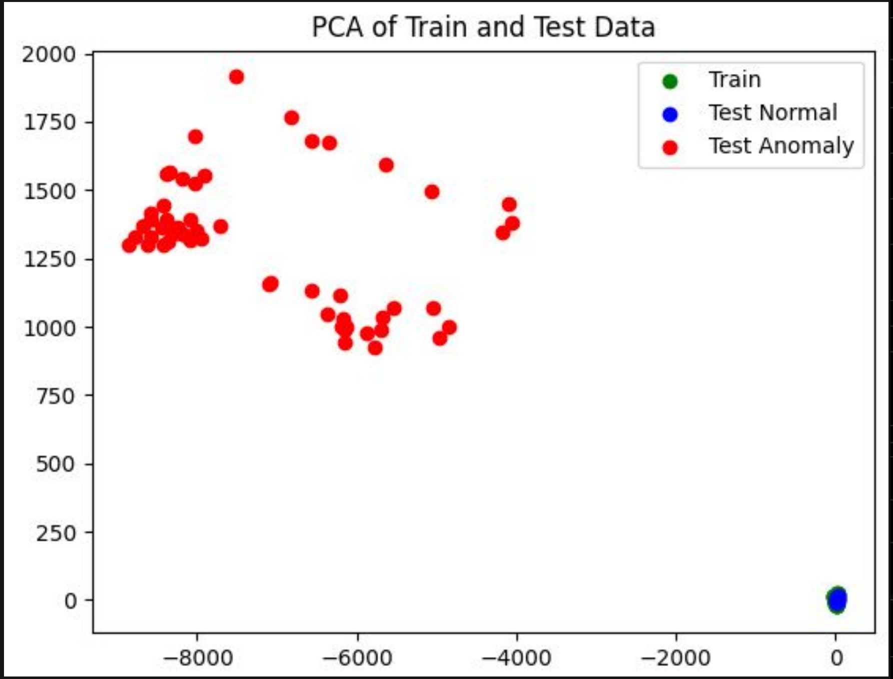
- **Accuracy: 96.55%**
- **Precision: 96.10%**
- **Recall: 100%**
- **F-1: 98.01%**



DNS Attack 2

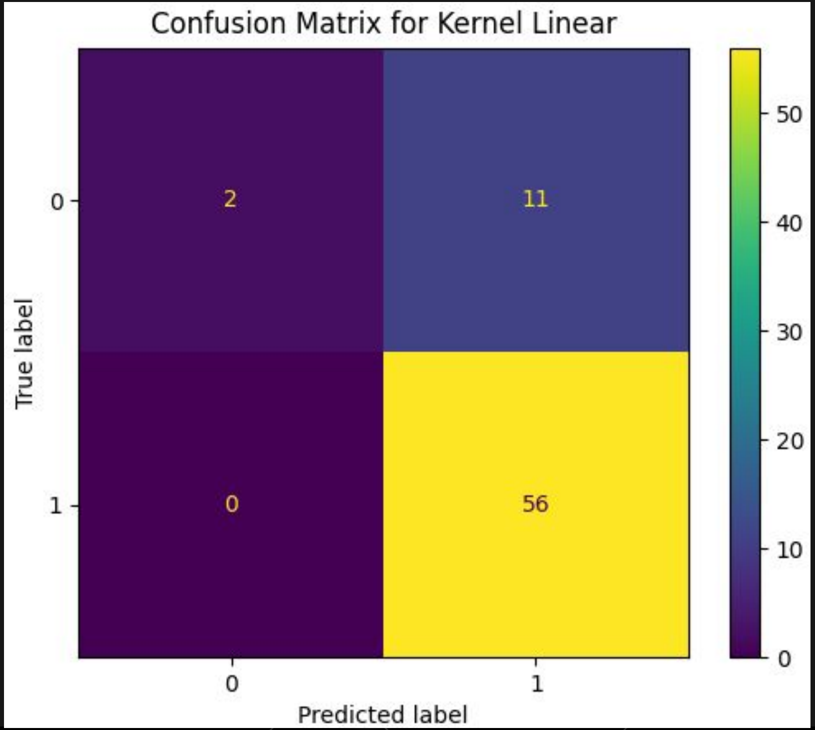
Through DNS this attacker sends responses with injected sh commands with steady active for 3 seconds.

- Window size : 10 minutes
- Sliding window every : 1 minute
- Sampling period : 5 seconds



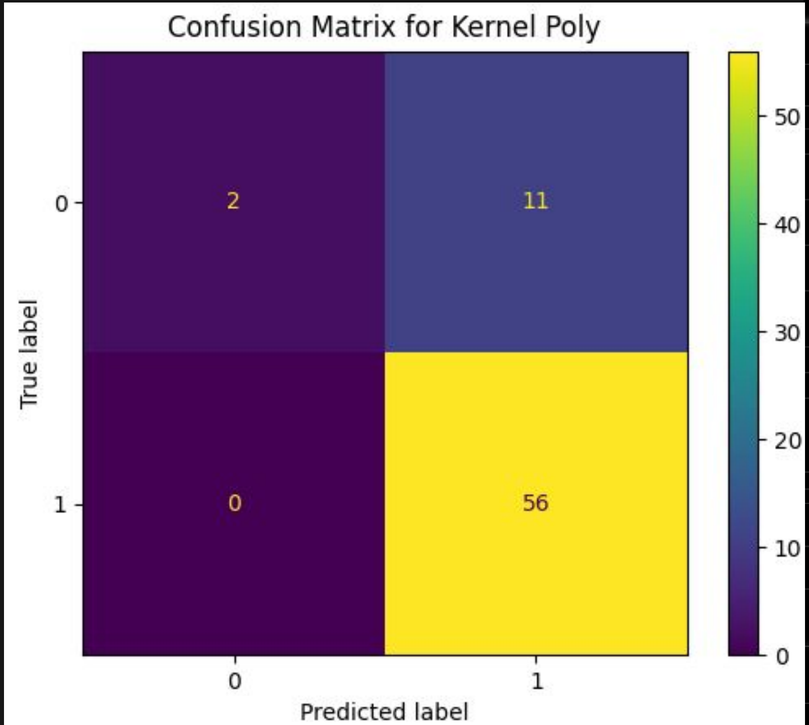
DNS Attack 2 – Kernel Linear results

- **Accuracy: 84.05%**
- **Precision: 83.58%**
- **Recall: 100%**
- **F-1: 91.05%**



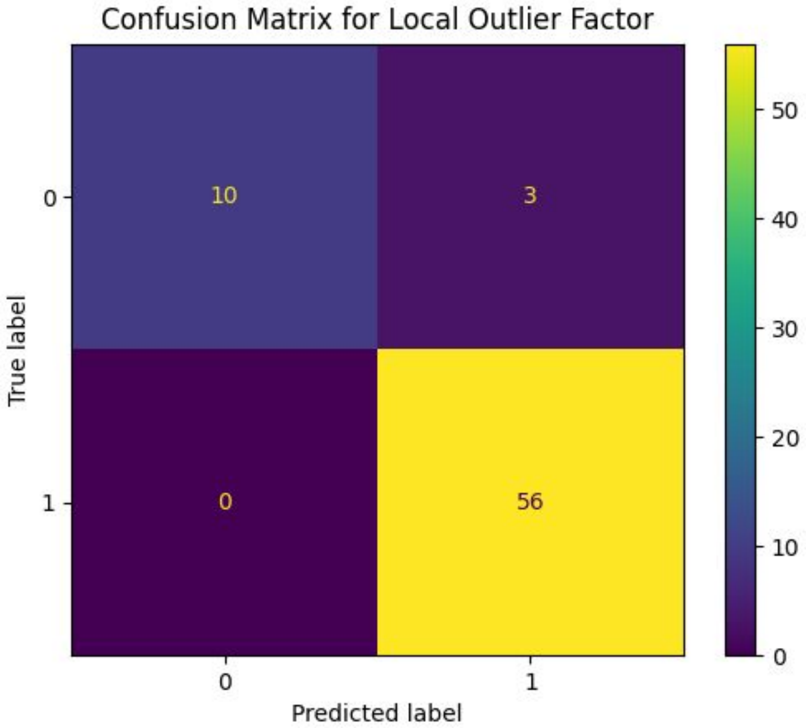
DNS Attack 2 – Kernel Poly results

- Accuracy: 84.05%
- Precision: 83.58%
- Recall: 100%
- F-1: 91.05%



DNS Attack 2 – LocalOutlierFactor results

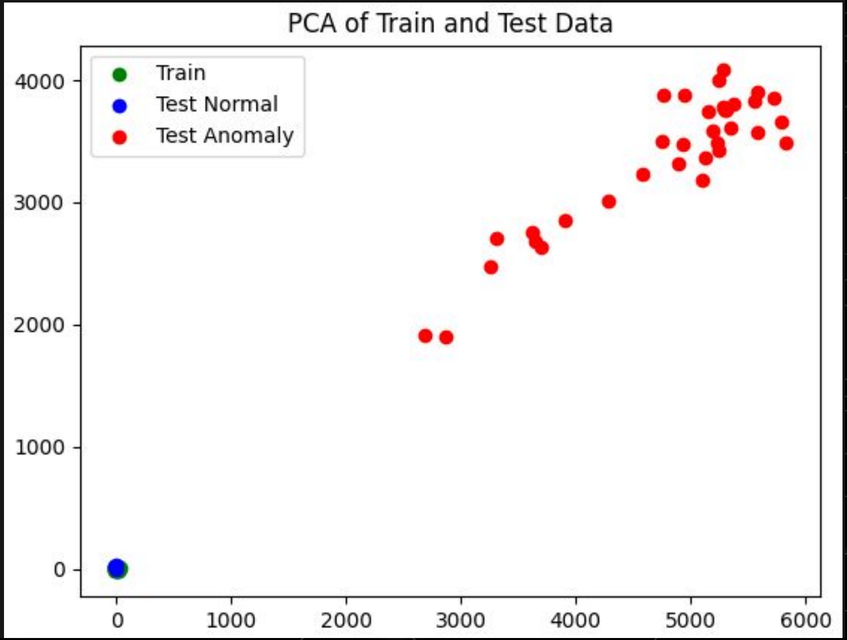
- Accuracy: 95.65%
- Precision: 94.91%
- Recall: 100%
- F-1: 97.39%



DNS Attack 3

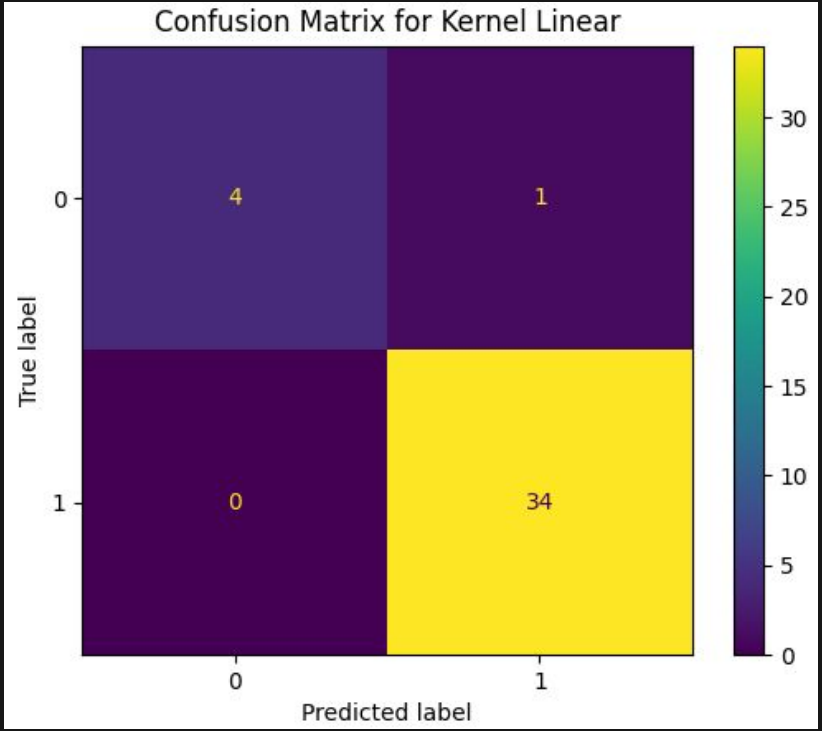
Through DNS this attacker sends responses with injected sh commands with steady non active for 5 seconds.

- Window size : 10 minutes
- Sliding window every : 1 minute
- Sampling period : 10 seconds



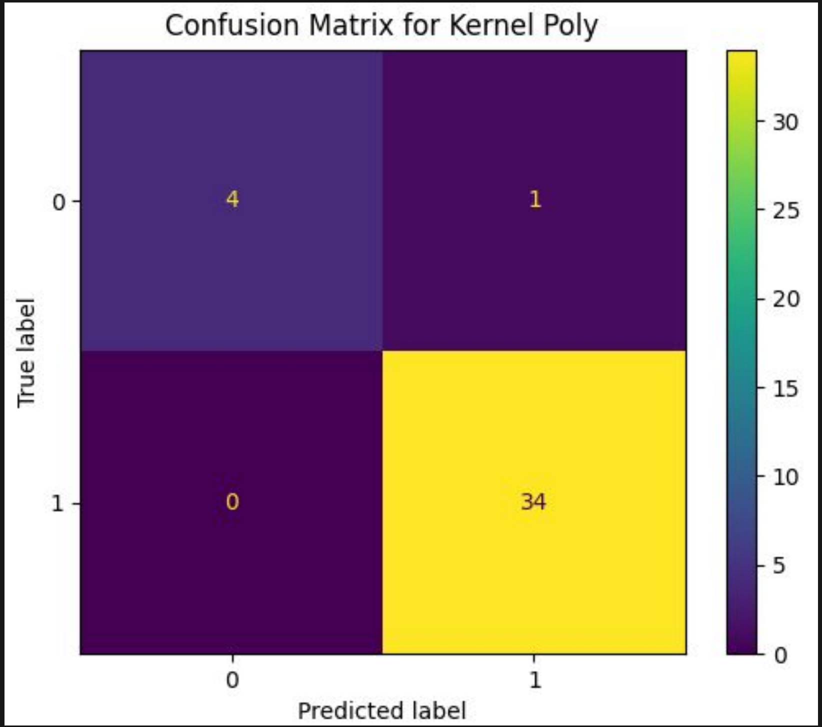
DNS Attack 3 – Kernel Linear results

- Accuracy: 97.43%
- Precision: 97.14%
- Recall: 100%
- F-1: 98.55%



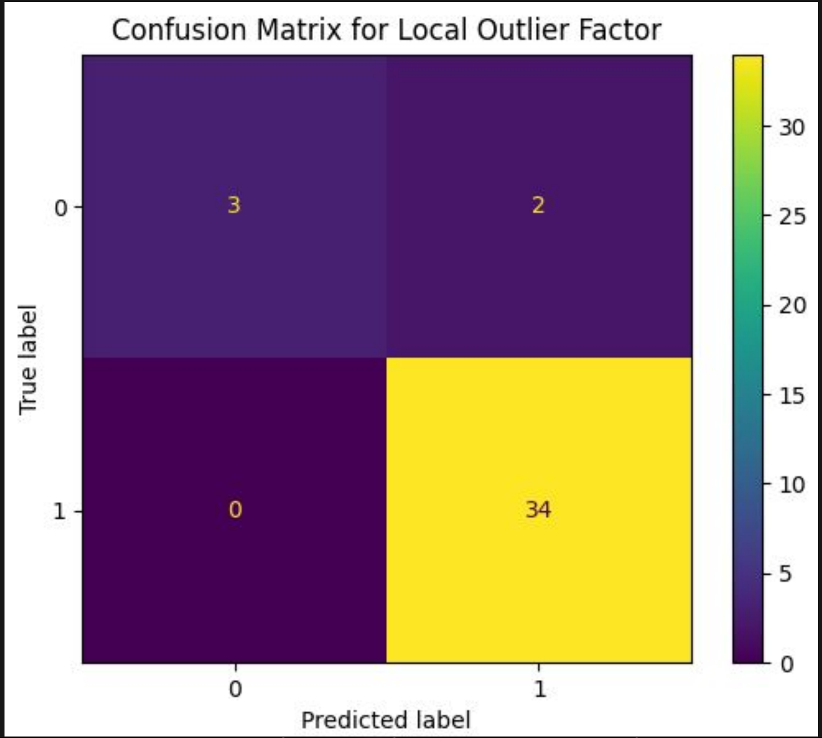
DNS Attack 3 – Kernel Poly results

- Accuracy: 97.43%
- Precision: 97.14%
- Recall: 100%
- F-1: 98.55%



DNS Attack 3 - LocalOutlierFactor results

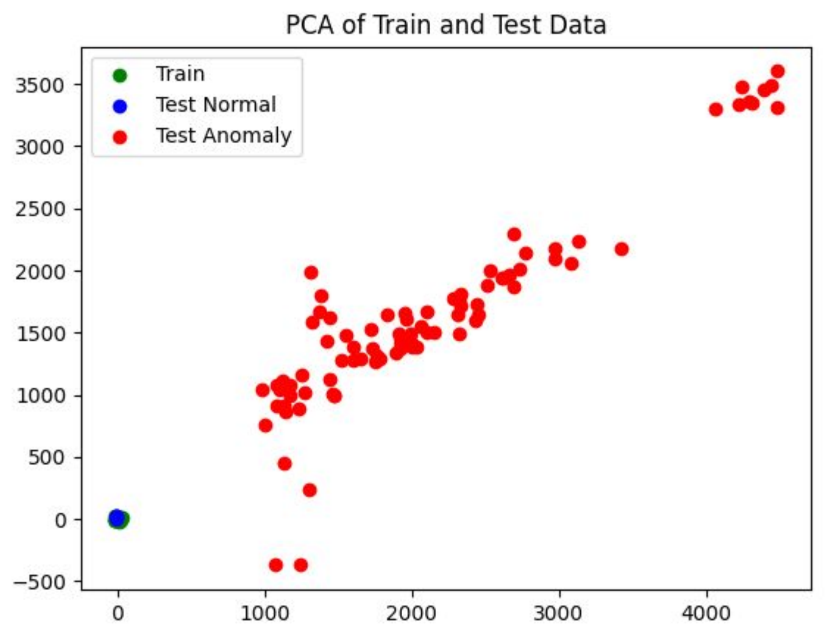
- Accuracy: 94,87%
- Precision: 94.44%
- Recall: 100%
- F-1: 97.14%



DNS Attack 4

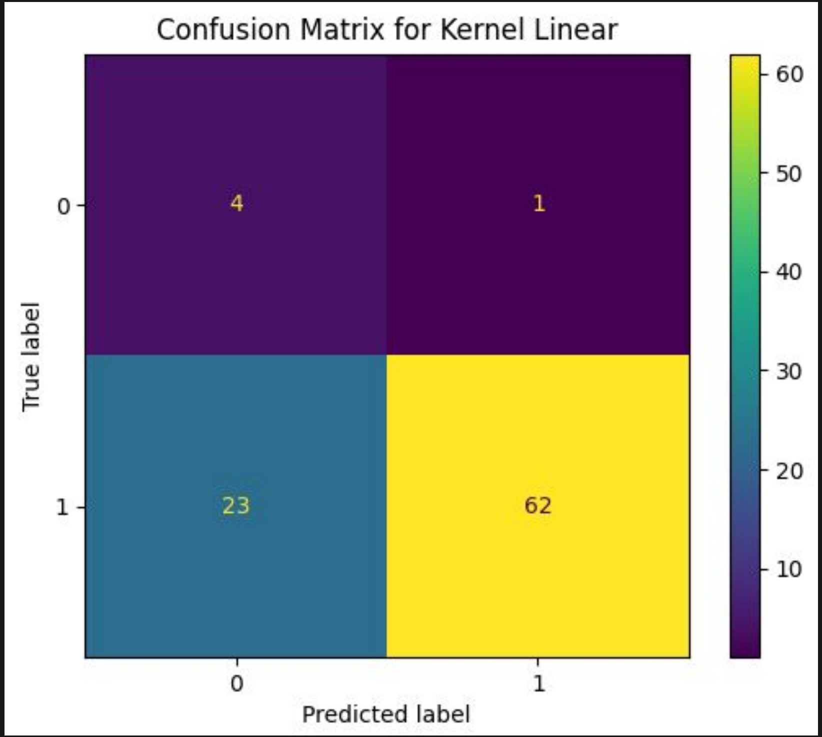
Through DNS this attacker sends responses with injected sh commands with steady non active for 5 seconds.

- Window size : 10 minutes
- Sliding window every : 1 minute
- Sampling period : 10 seconds



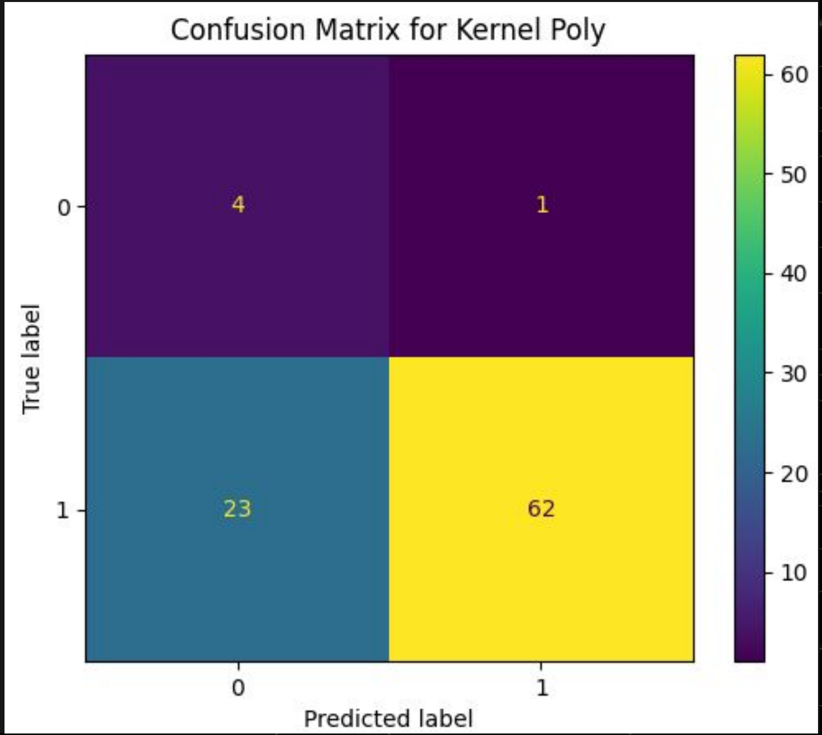
DNS Attack 4 – Kernel Linear results

- Accuracy: 73.33%
- Precision: 98.41%
- Recall: 72.94%
- F-1: 83.78%



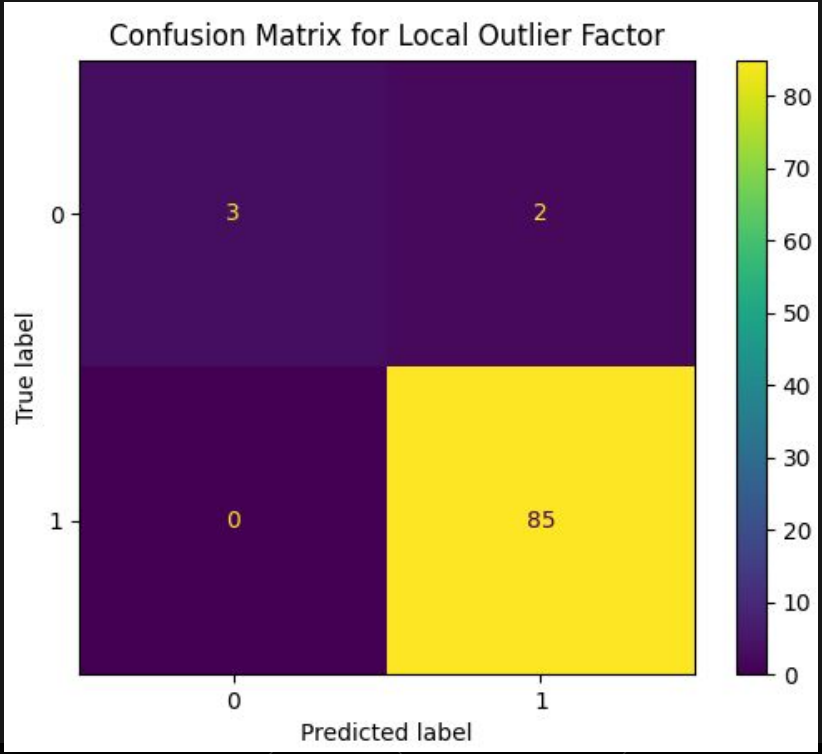
DNS Attack 4 – Kernel Poly results

- Accuracy: 73.33%
- Precision: 98.41%
- Recall: 72.94%
- F-1: 83.78%



DNS Attack 4 – LocalOutlierFactor results

- **Accuracy: 97.77%**
- **Precision: 97.70%**
- **Recall: 100%**
- **F-1: 98.83%**



Any questions?

References

https://ieee-dataport.org/documents/ti-2016-dns-dataset
https://scikit-learn.org/stable/modules/classes.html
https://github.com/KimiNewt/pyshark
https://numpy.org/doc/stable/reference/index.html#reference

https://github.com/Tiagura/TPR_Project -> project code