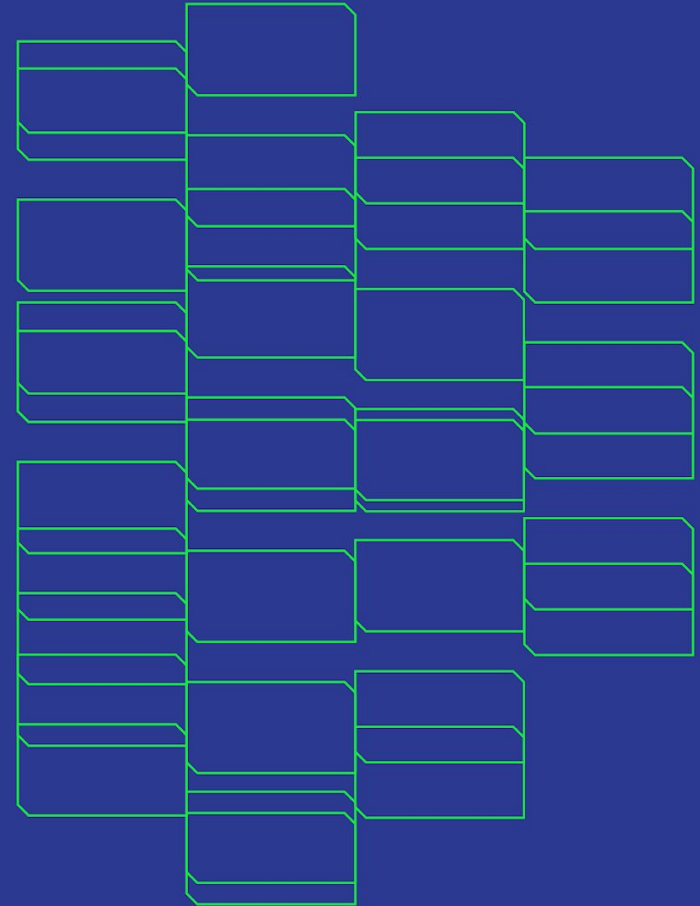


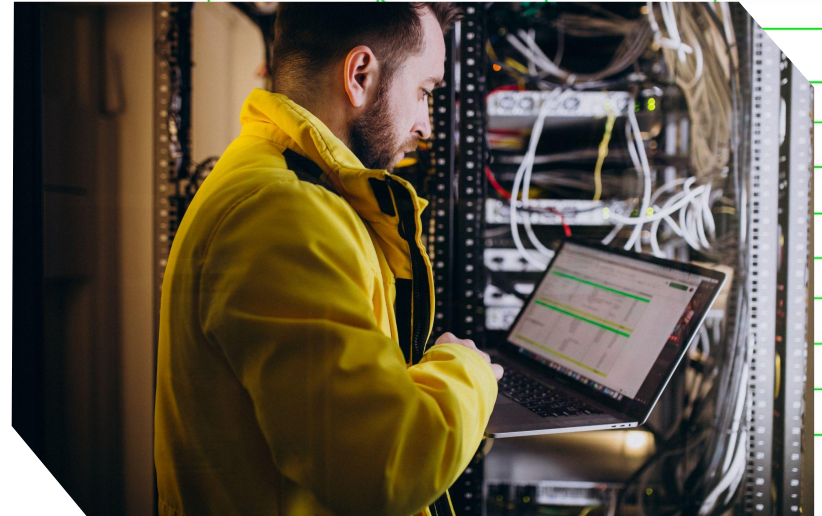
Exploring machine learning for DDoS mitigation

Jakub Man
CESNET



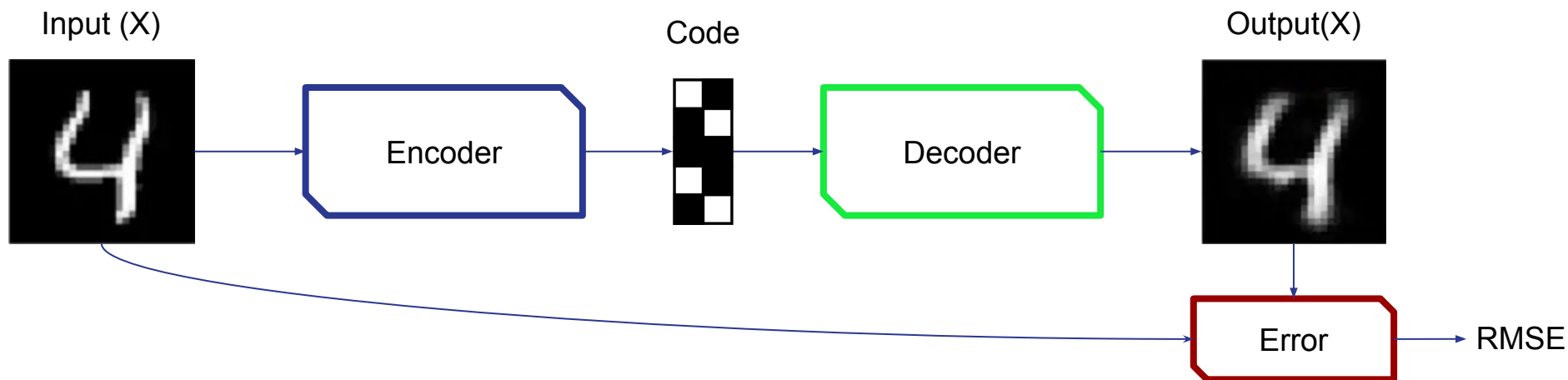
Requirements on ML

- Fast response to changing attacks
- Low false-positives
- Good performance



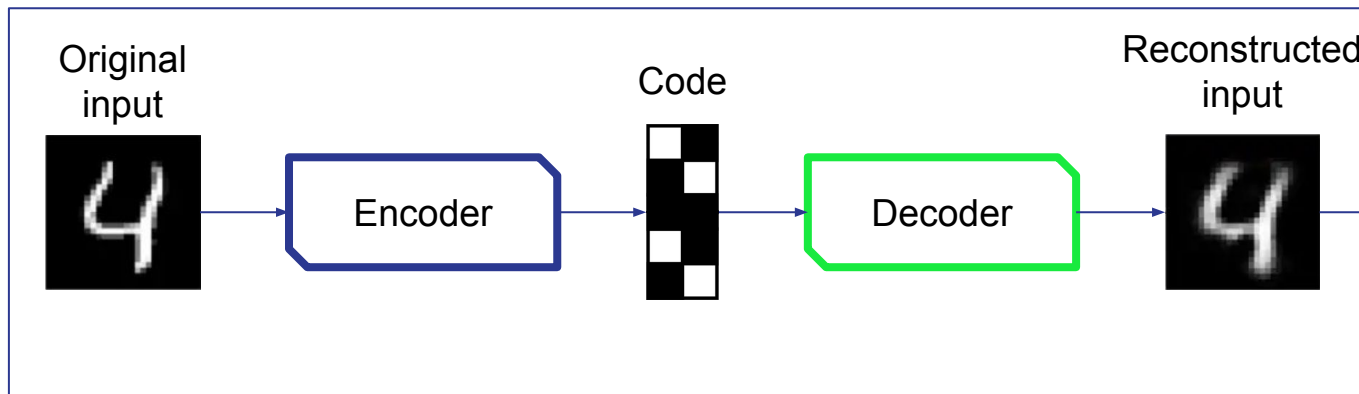
Autoencoder

- Artificial neural network
- Reconstructs its input on output ($X = Y$)
- Minimizes reconstruction error $RMSE(X, Y)$
- Trained in unsupervised manner (only legitimate traffic)



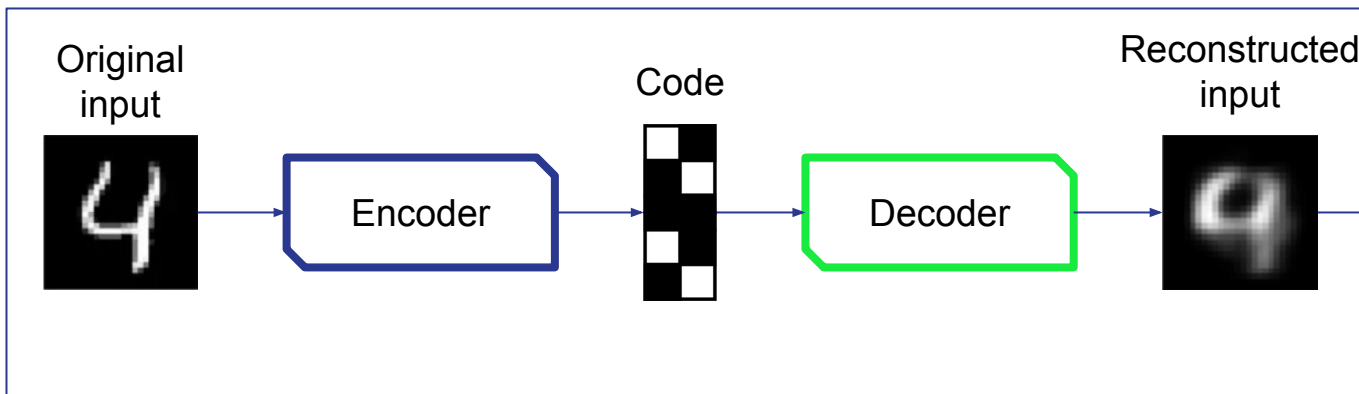
Autoencoder - anomaly detection in packets

Benign packets



Small reconstruction error

Malicious packets



Large reconstruction error

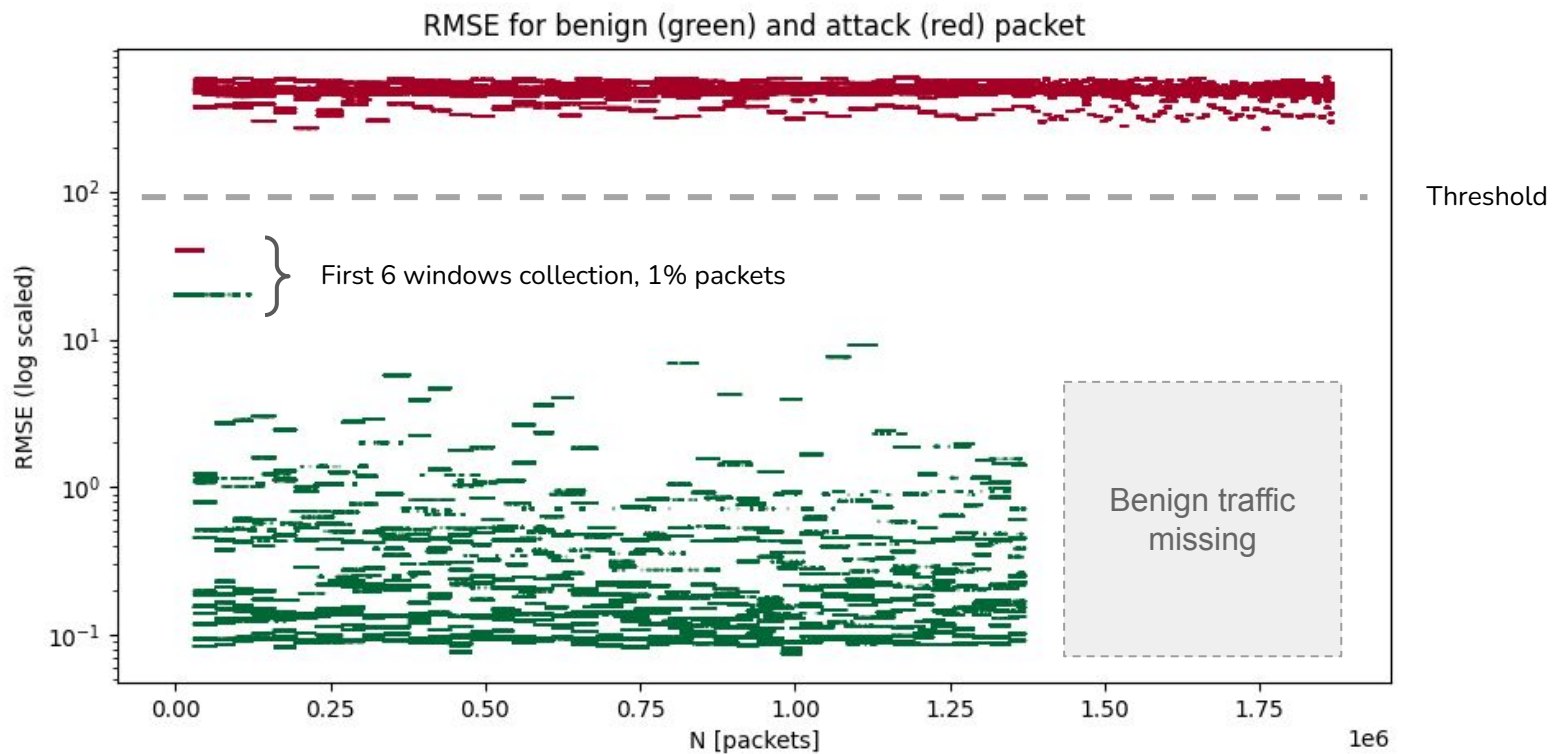
Experimenting with autoencoders

- Uses sFlow (sampled packets)
- Aggregates “windows” of traffic by source IPs
- Features typical for attacks are extracted from the windows
 - Timestamp, L4 ports, payload length, ...
- Result is a list of malicious IPs

```
----- Per-Source IP Communication Statistics -----
```

	detected_after	detections_pos	detections_neg	pkts_allowed	pkts_denied	label
.126	0	0	50	78946	0	Benign
.231	0	0	50	105784	0	Benign
.190	0	0	50	134751	0	Benign
.233	0	0	49	85529	0	Benign
.59	0	0	50	146605	0	Benign
.42	0	0	50	121405	0	Benign
32	0	0	50	85362	0	Benign
.34	0	0	50	138748	0	Benign
.219	0	0	50	149444	0	Benign
.244	0	0	50	80100	0	Benign
3.215	0	0	50	93319	0	Benign
.132	0	0	50	85493	0	Benign
254	0	0	50	126989	0	Benign
.188	0	0	50	114494	0	Benign
118	0	0	50	122068	0	Benign
38	6	50	0	1749	87661	Benign
.120	0	0	49	113594	0	Benign
3.138	6	97	0	340	45844	Attack
.76	6	96	0	444	44993	Attack
3.246	6	96	0	487	45777	Attack
.196	6	97	0	458	55951	Attack
3.200	6	97	0	495	46725	Attack
3.55	6	97	0	509	48287	Attack
217	6	96	0	456	45142	Attack
.116	6	96	0	460	45094	Attack
39	6	96	0	495	44990	Attack
7.180	6	97	0	520	46630	Attack
3.73	6	95	0	515	45812	Attack
1.44	6	97	0	446	45611	Attack
.42	6	96	0	458	45682	Attack
3.79	6	96	0	492	47730	Attack
21	6	96	0	596	46220	Attack
2.166	6	91	0	509	45294	Attack
26	6	96	0	474	46067	Attack
.12	6	96	0	453	47535	Attack
.229	6	96	0	527	47879	Attack
167	6	96	0	498	47869	Attack

Autoencoder test results



Threshold: 100; Recall: 0.99; Precision: 1.00; Accuracy: 0.99

How do we help networking professionals to configure this correctly?

```
[windower]
window_length = 1
history_min = 6
history_size = 0
history_timeout = 900
packets_min = 0
samples_size = 40
keep_frag = false

[kitnet]
max_autoencoder_size = 10
feature_mapper_learn_period = 50
learning_rate = 0.1
hidden_ratio = 0.75
```

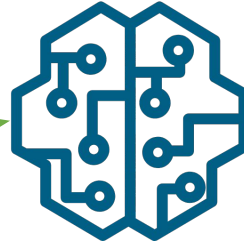
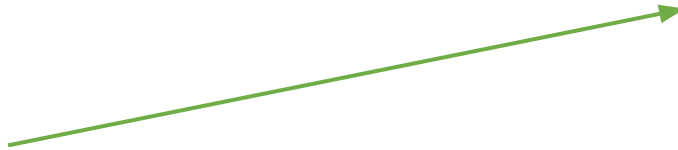
Configuration



Administrator

Configuration

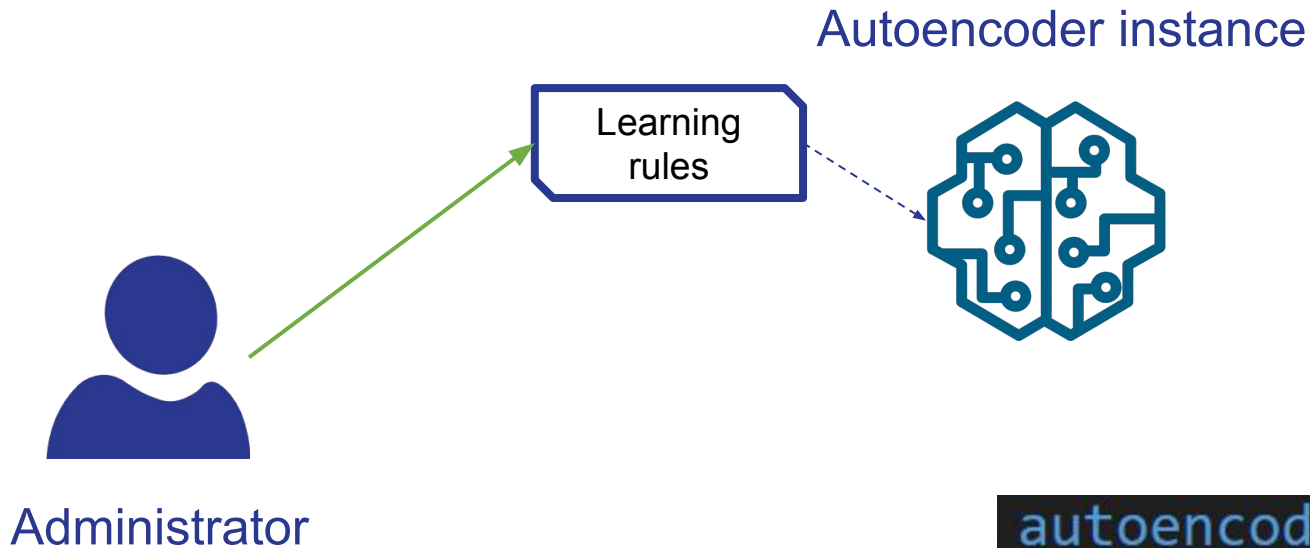
Autoencoder instance



Administrator

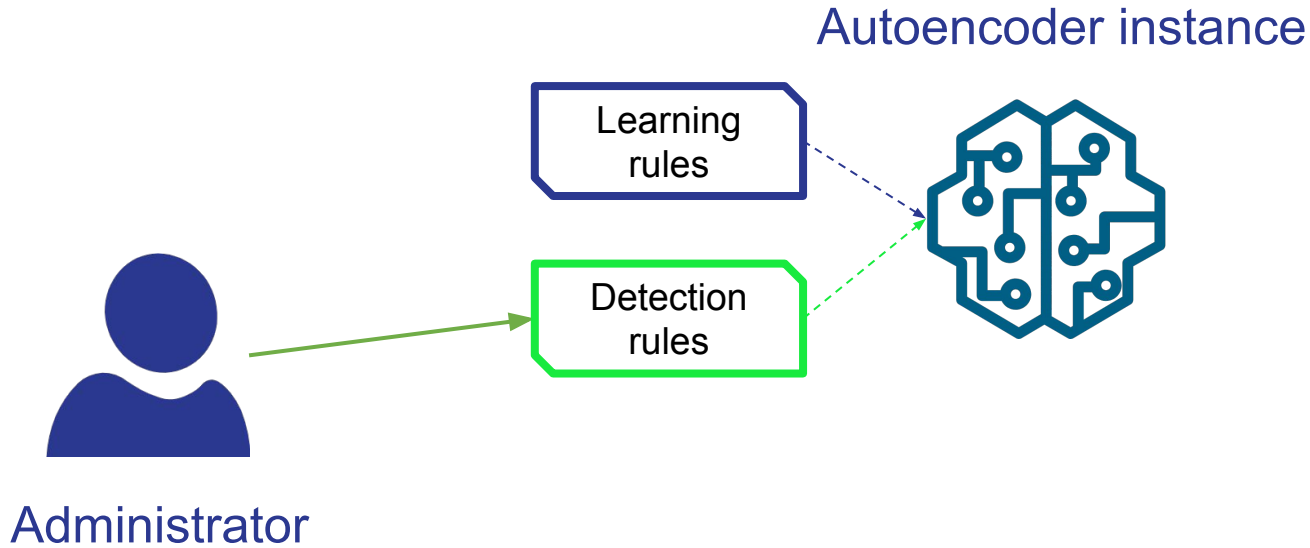
```
autoencoder_instance:  
  detection_threshold: 50  
  name: "Primary instance"  
  ip_set_id: 1
```

Configuration



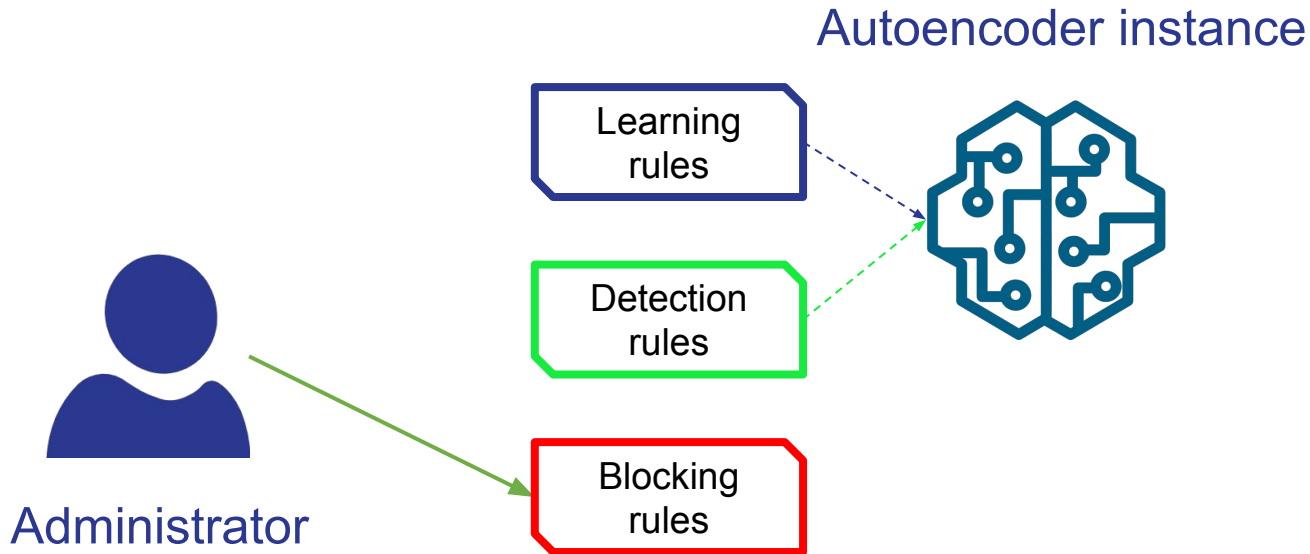
```
autoencoder:  
  mode: learn  
  ip_src: ['192.168.0.0/16']  
  vlan: 100  
  instance: "Primary instance"
```

Configuration



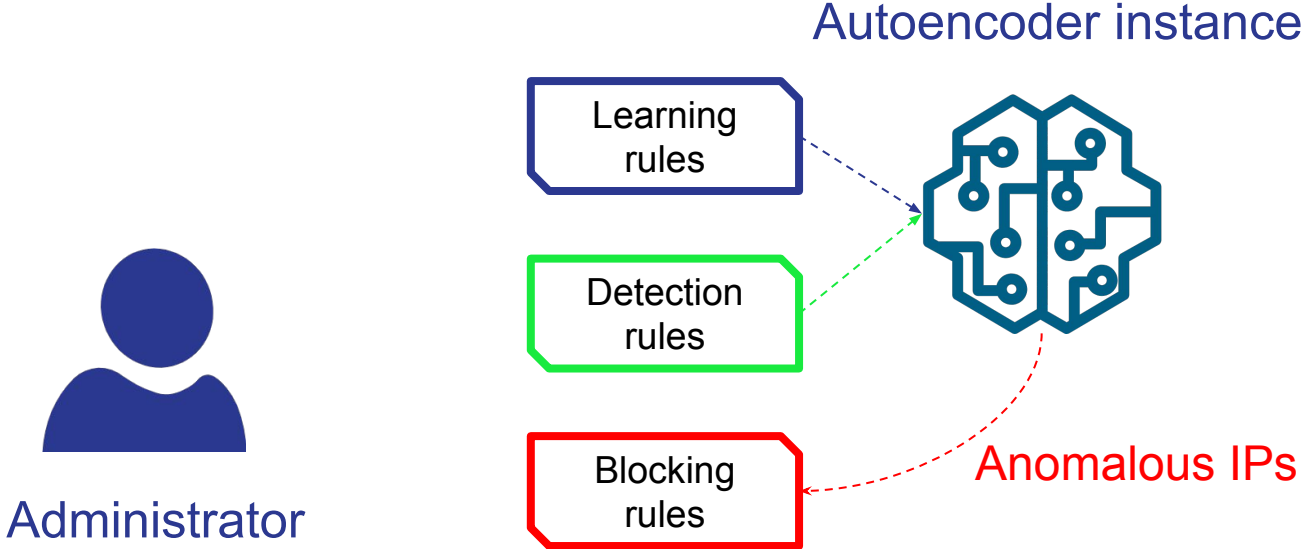
```
autoencoder:  
  mode: detect  
  ip_src: ['10.50.0.0/16']  
  instance: "Primary instance"
```

Configuration



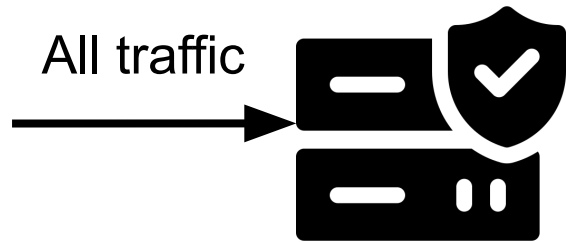
```
exact_match:  
  ip_set_id: 1  
  ip_dst: ['10.60.0.0/16']
```

Configuration

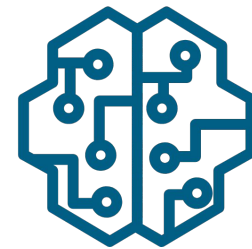


Integration into our system

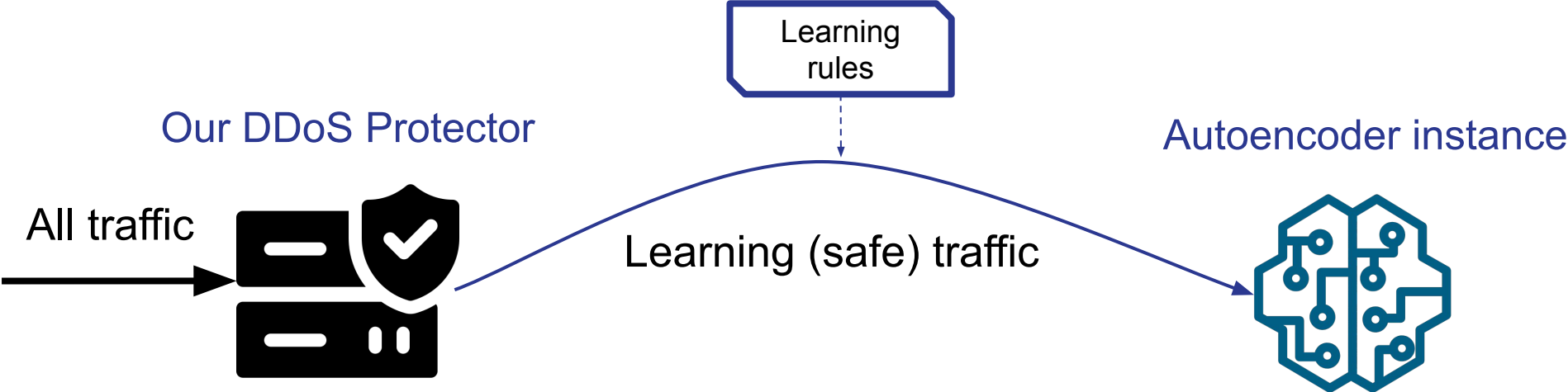
Our DDoS Protector



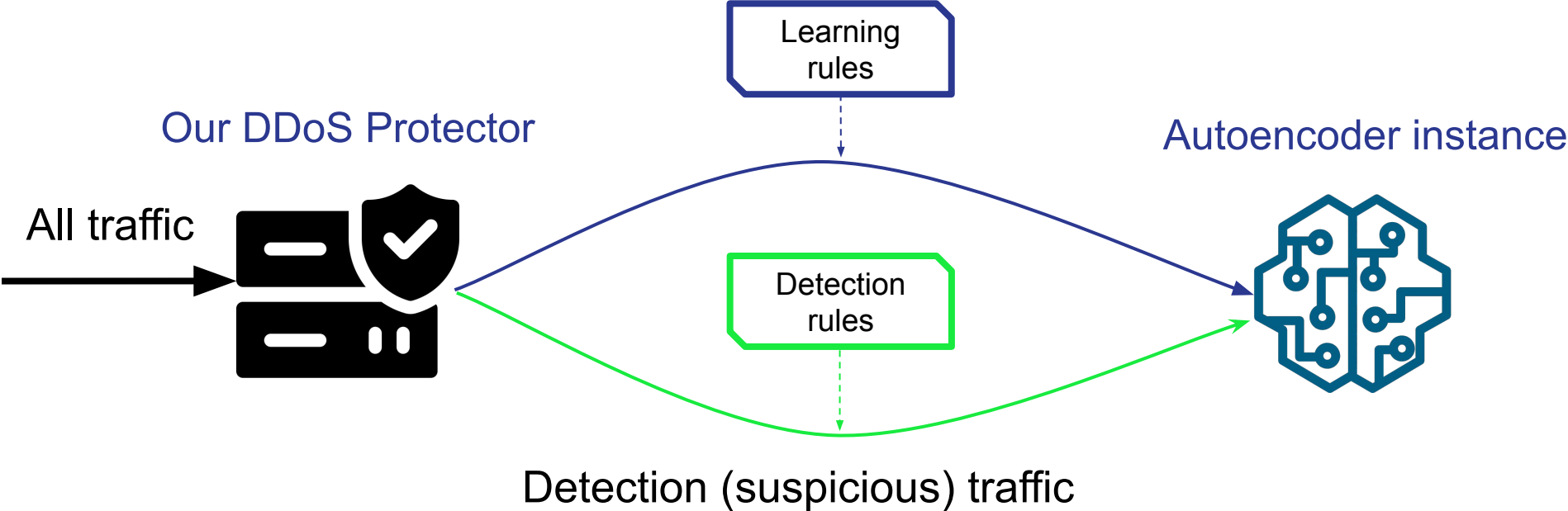
Autoencoder instance



Integration into our system



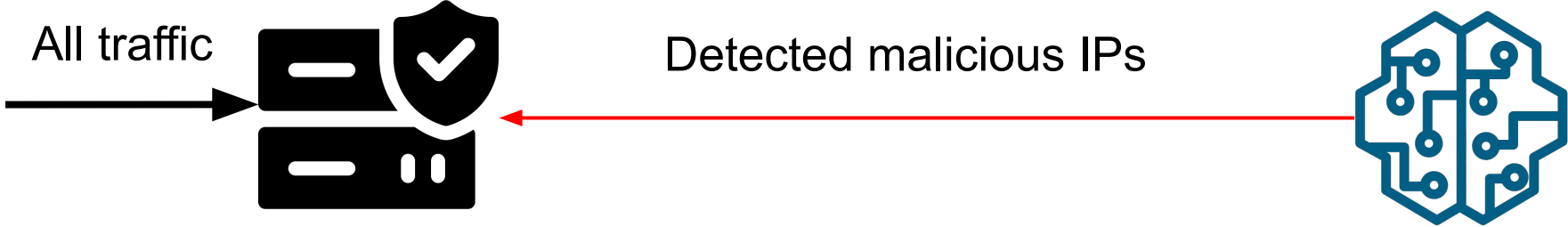
Integration into our system



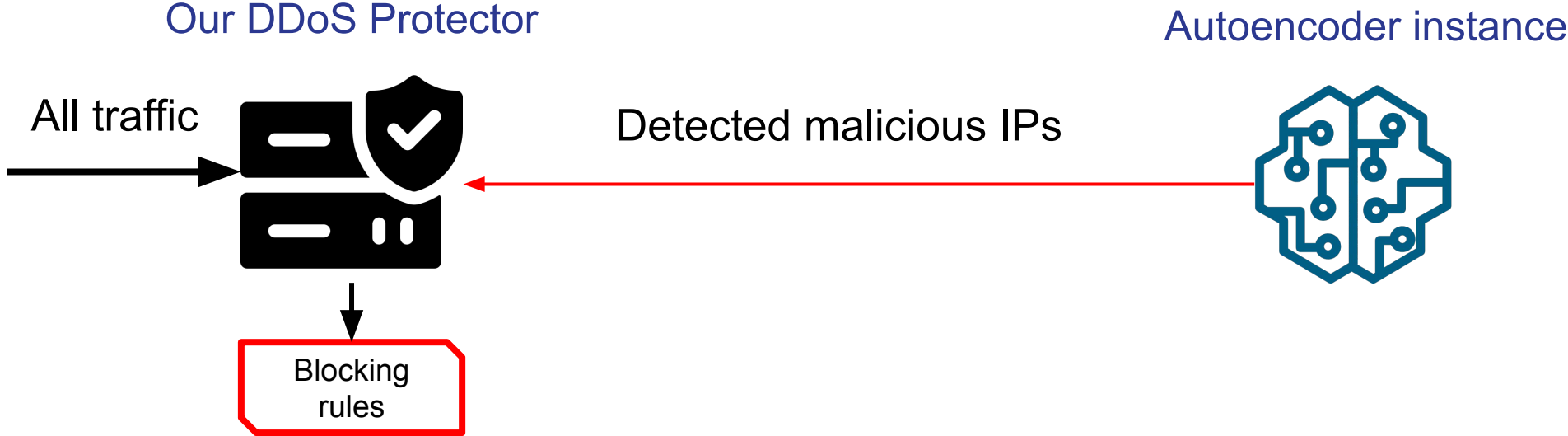
Integration into our system

Our DDoS Protector

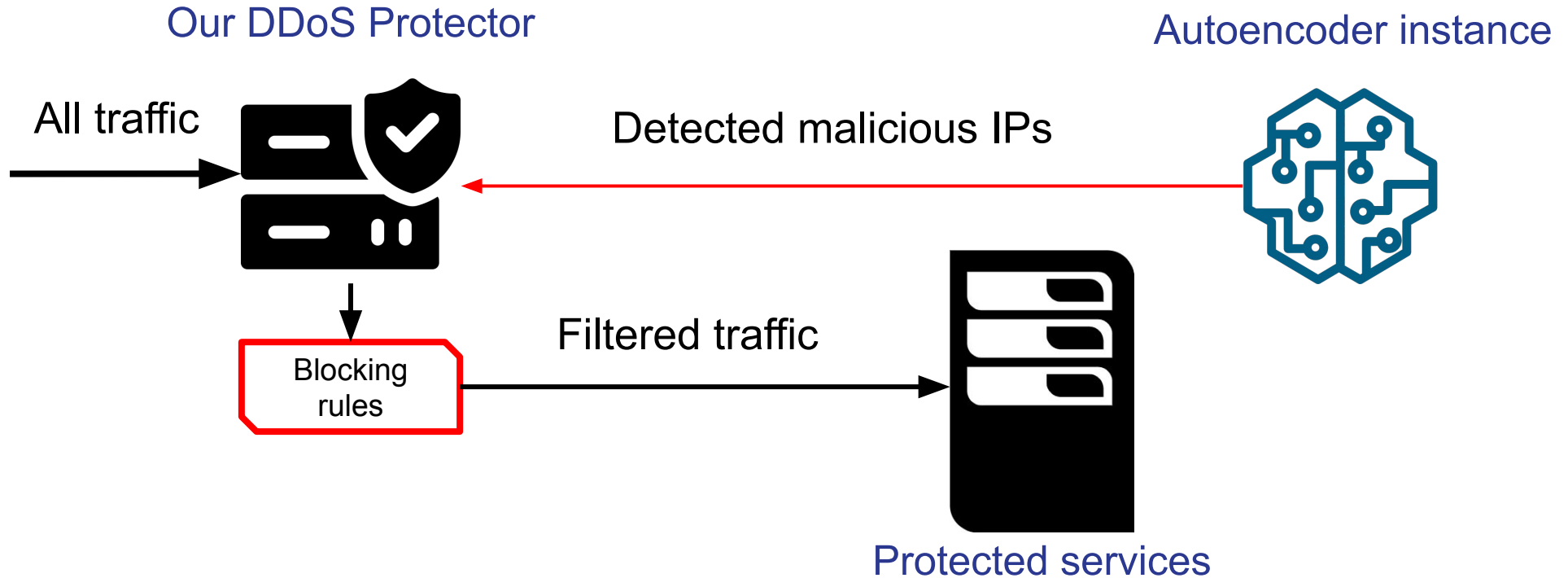
Autoencoder instance



Integration into our system



Integration into our system



Experimental integration

The screenshot shows the ExaFS web interface. A modal window titled "IP set Main instance" is open, displaying a list of IP addresses. The modal includes a search bar, a "View raw data" button, and a table of IP addresses with "whois" and "NERD" buttons for each. The background shows the "Active DDoS Protector rules" section with a table of rules and their status.

ID	Source IP ranges	Source ports
6	Any IPv4 Any IPv6	Any port
5	Any IPv4 Any IPv6	Any port
4	Any IPv4 Any IPv6	Any port
3	Any IPv4 Any IPv6	Any port
2	Any IPv4 Any IPv6	Any port
1	Any IPv4 Any IPv6	Any port

IP Address	whois	NERD
10.44.80.1	whois	NERD
10.44.80.2	whois	NERD
10.44.80.3	whois	NERD
10.44.80.4	whois	NERD
10.44.80.5	whois	NERD
10.44.80.6	whois	NERD
10.44.80.7	whois	NERD
10.44.80.8	whois	NERD
10.44.80.9	whois	NERD
10.44.80.10	whois	NERD

Conclusion

- Traditional methods are still needed as a pre-filter
- More complex attacks can be handled by machine learning
- Performance needs to be improved for production



Thank you

Any questions?

Jakub.Man@cesnet.cz
ddp@cesnet.cz



Co-funded by
the European Union



Security .Days