NetFlow Based Botnet Detection

SEYEDALIREZA VAZIRI – RIPE 75

About Me

SeyedAlireza Vaziri

- Network/System Engineer since 2007
- Security Administrator since 2016
- Machine Learning newbie

Agenda

- Botnets, Usage, History
- Modern Botnets
- Botnet detection and countermeasure
- Netflow based detection
- Machine learning classification
- Questions

Bot

Vulnerable and unattended Devices:

- Computers
- Smartphones
- IoT (e.g. CCTV, xDSL Modem)

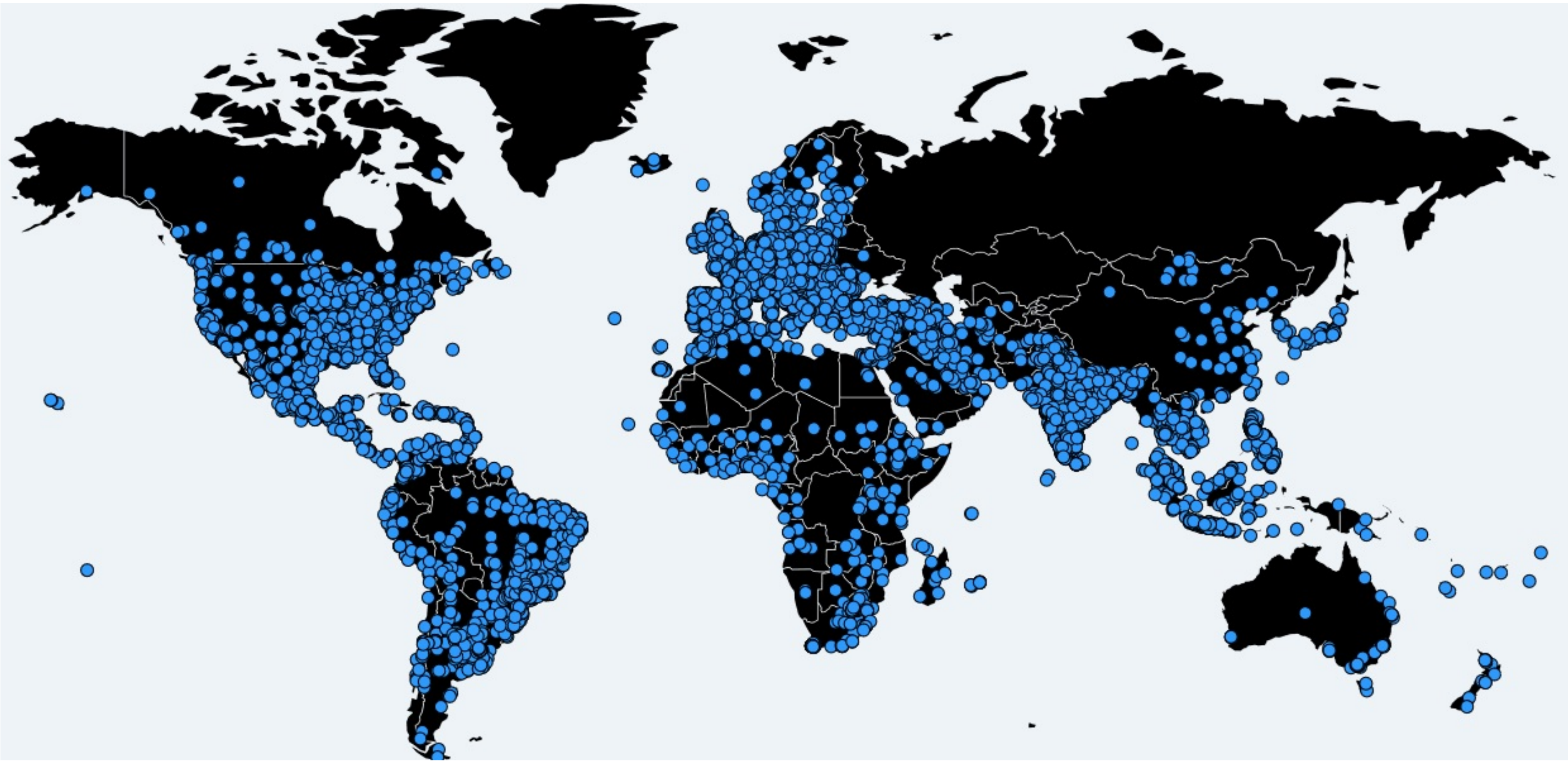
Botnet Usage

Network of bots is named Botnet and being used for:

- Spams
- DDoS
- Malware Distribution

Botnet History

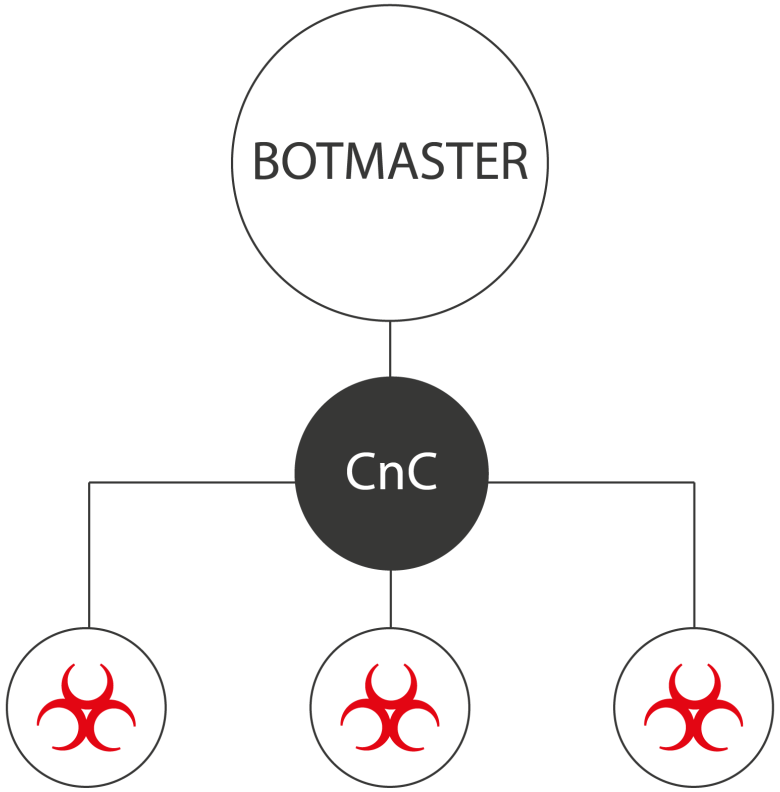
- Marina
- Zeus
- Cutwail
- Mirai



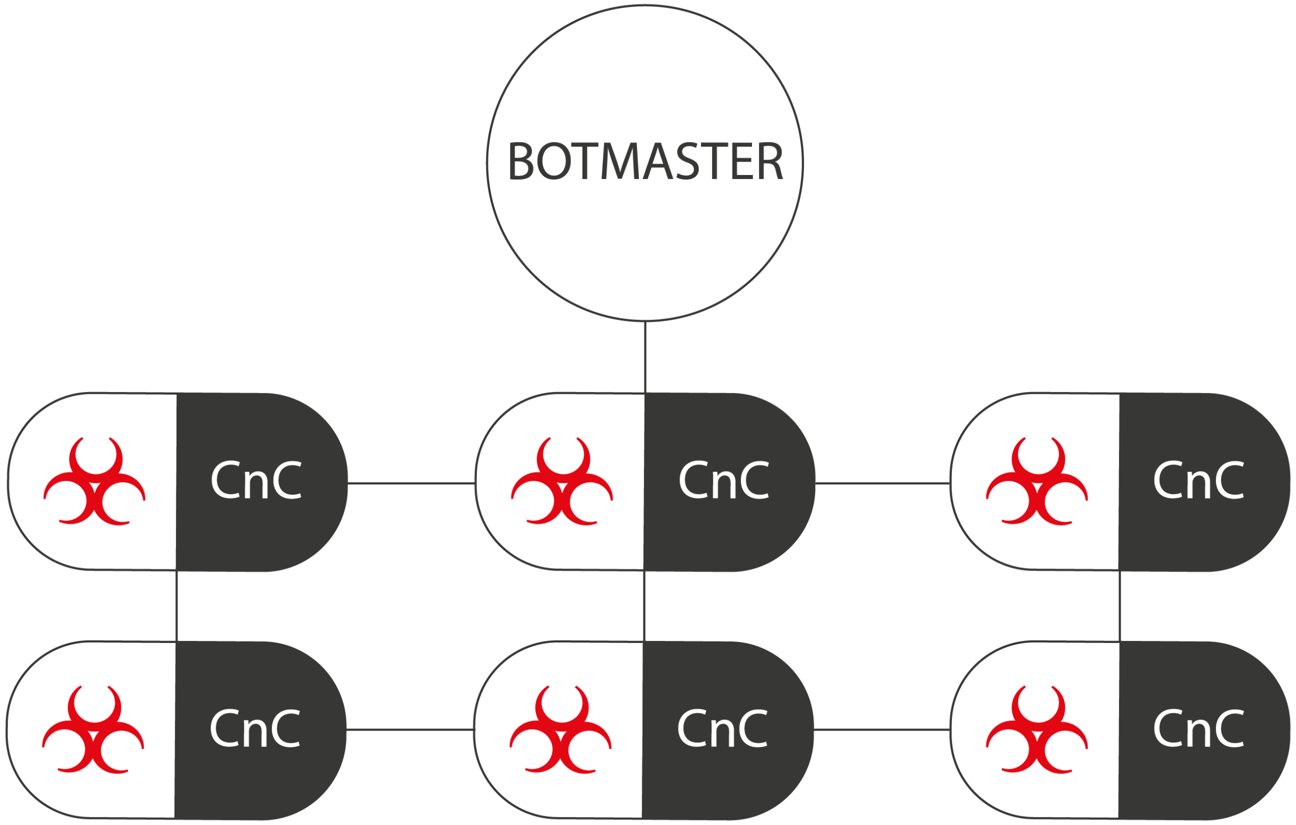
Botnet Dictionary

- Bot
- Botnet
- CnC (Command and Control)
- Botmaster

Botnet Diagram



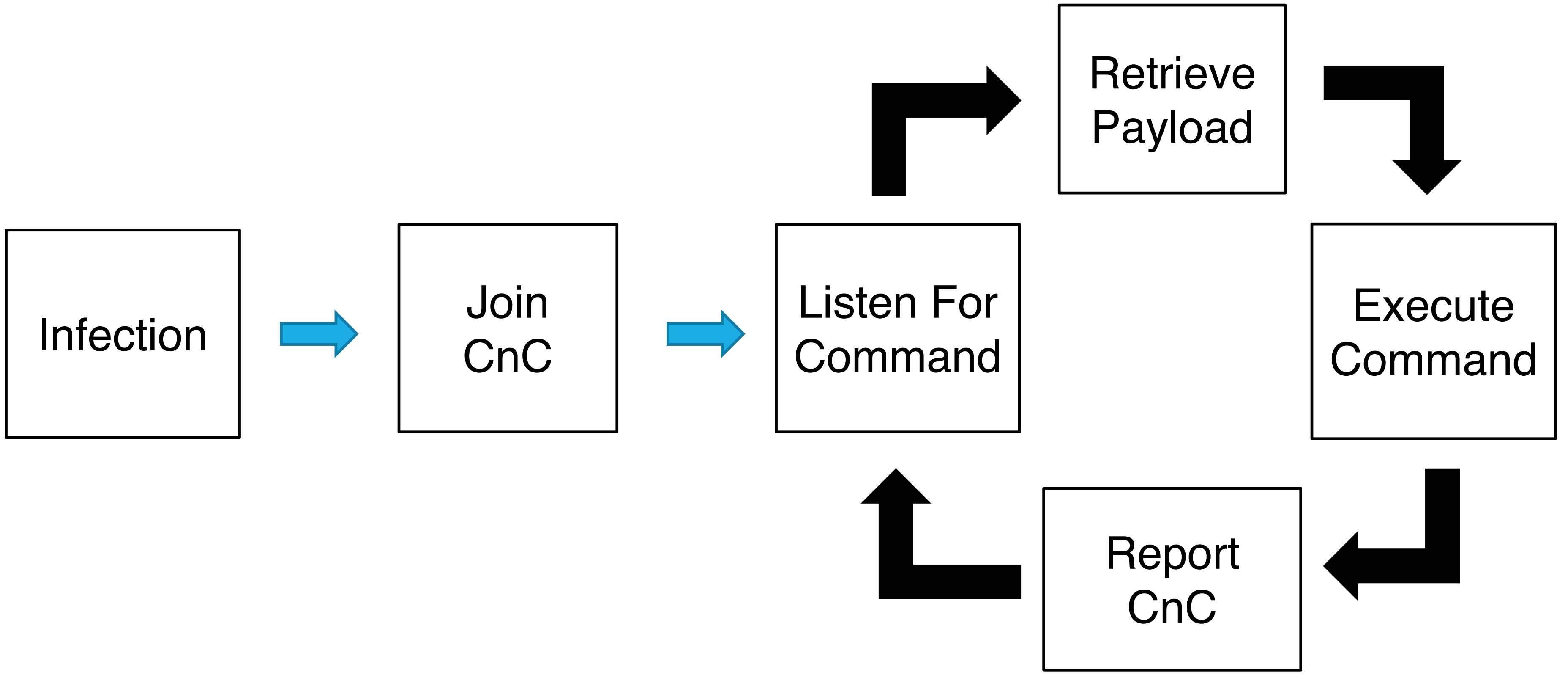
Modern Botnet Diagram



Modern Botnet

- P2P Communication
- No SPOF (Single Point of Failure)
- Encryption
- Randomness
- Obfuscation

Bot lifecycle



Botnet Detection

Current methods:

- IDPS
- DPI
- Signature Based, Anomaly Based

Dealing with Botnets

Internal

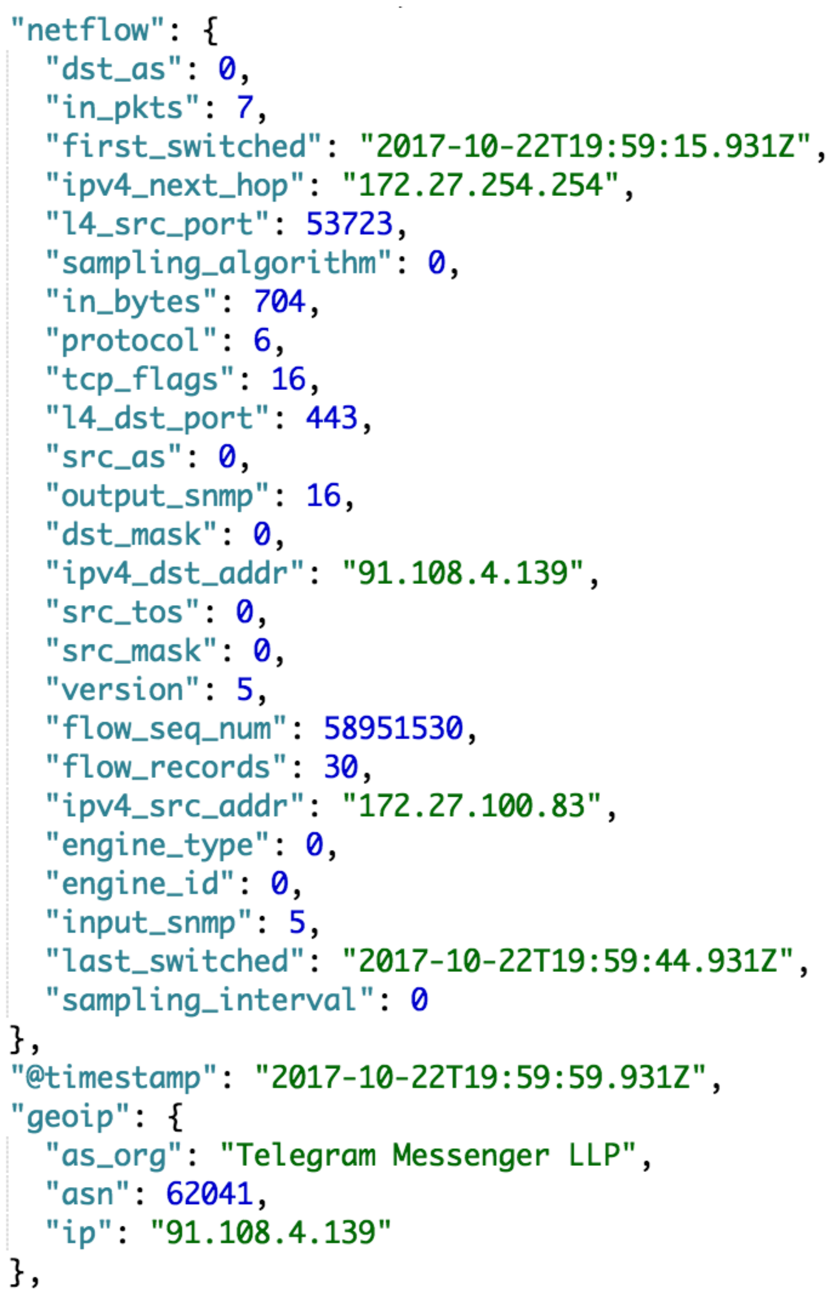
We are attacking others

External

Others attacking us

NetFlow/S-Flow/IPFIX

- src/dst IP/Port
- Packet
- Bytes
- ASN
- Duration



```
"netflow": {
  "dst_as": 0,
  "in_pkts": 7,
  "first_switched": "2017-10-22T19:59:15.931Z",
  "ipv4_next_hop": "172.27.254.254",
  "l4_src_port": 53723,
  "sampling_algorithm": 0,
  "in_bytes": 704,
  "protocol": 6,
  "tcp_flags": 16,
  "l4_dst_port": 443,
  "src_as": 0,
  "output_snmp": 16,
  "dst_mask": 0,
  "ipv4_dst_addr": "91.108.4.139",
  "src_tos": 0,
  "src_mask": 0,
  "version": 5,
  "flow_seq_num": 58951530,
  "flow_records": 30,
  "ipv4_src_addr": "172.27.100.83",
  "engine_type": 0,
  "engine_id": 0,
  "input_snmp": 5,
  "last_switched": "2017-10-22T19:59:44.931Z",
  "sampling_interval": 0
},
"@timestamp": "2017-10-22T19:59:59.931Z",
"geoip": {
  "as_org": "Telegram Messenger LLP",
  "asn": 62041,
  "ip": "91.108.4.139"
},
```

Blacklist

Lists of CnC IP addresses:

- ISC
- CYMRU
- Spamhaus
- Many more

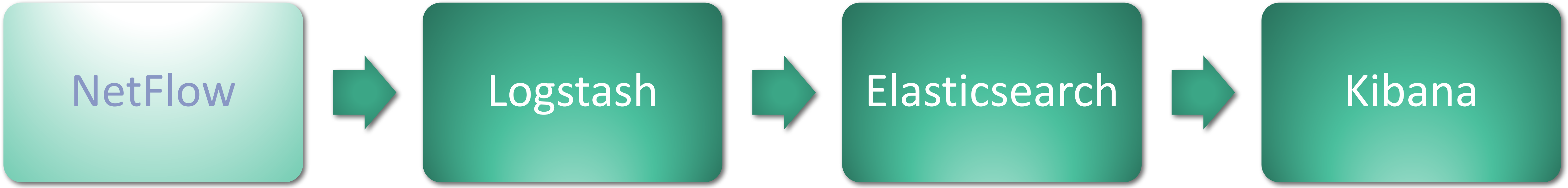
ELK Stack

Powerfull Search Engine:

- Elasticsearch, Logstash, Kibana
- Open Source
- Handle millions of records with ease
- Scalable

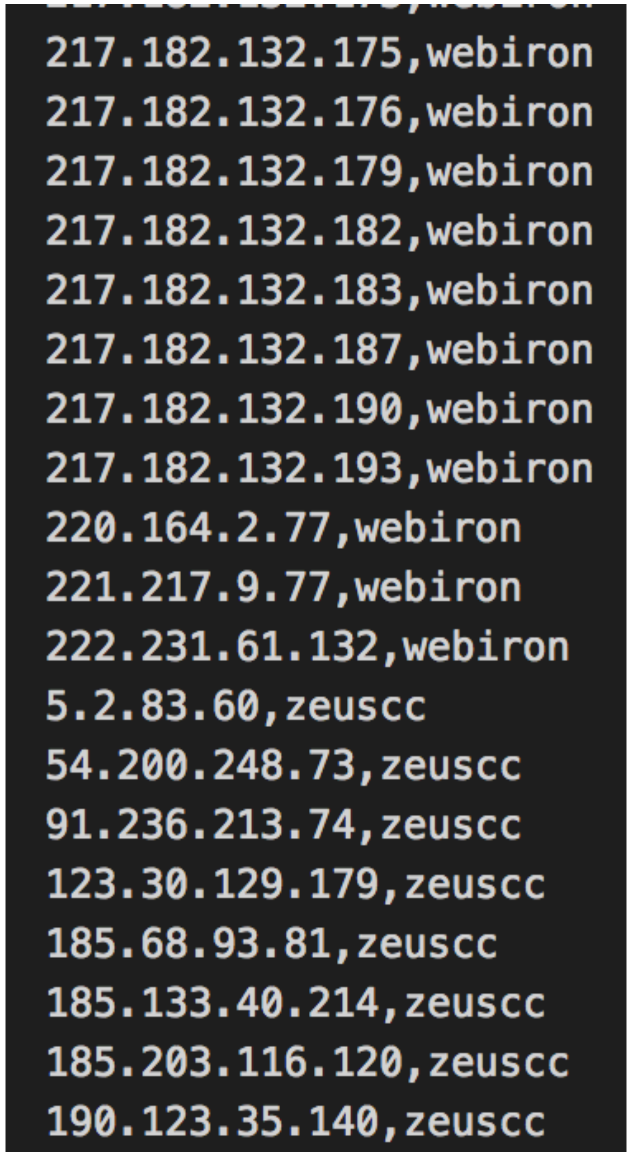
elastic

Netflow to ELK



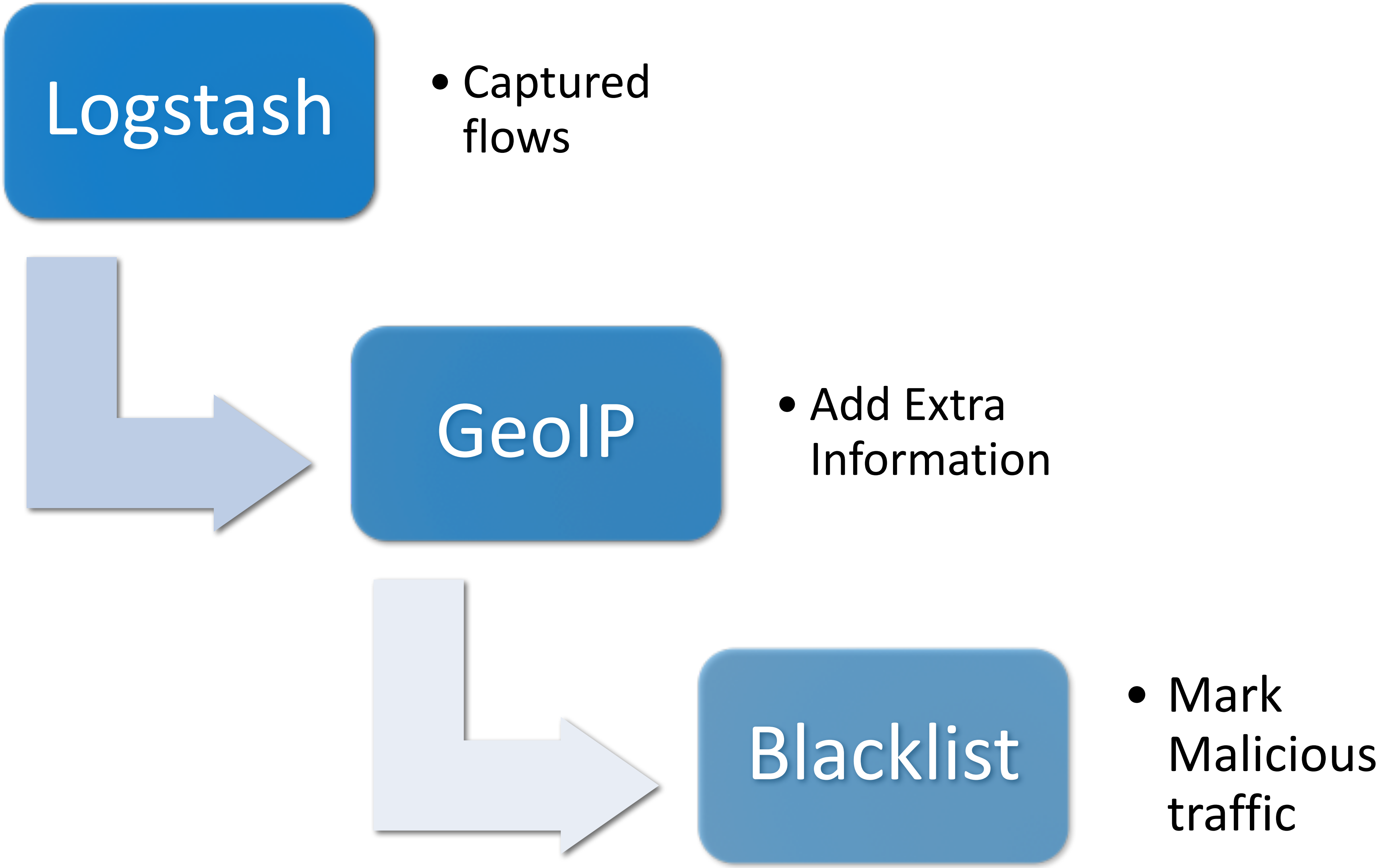
Logstash Filtering

- Blacklist IP Dictionaries
- Marking malicious traffic
- GeoIP translation



```
217.182.132.175,webiron
217.182.132.176,webiron
217.182.132.179,webiron
217.182.132.182,webiron
217.182.132.183,webiron
217.182.132.187,webiron
217.182.132.190,webiron
217.182.132.193,webiron
220.164.2.77,webiron
221.217.9.77,webiron
222.231.61.132,webiron
5.2.83.60,zeuscc
54.200.248.73,zeuscc
91.236.213.74,zeuscc
123.30.129.179,zeuscc
185.68.93.81,zeuscc
185.133.40.214,zeuscc
185.203.116.120,zeuscc
190.123.35.140,zeuscc
```

Logstash Diagram



Logstash

- Captured flows

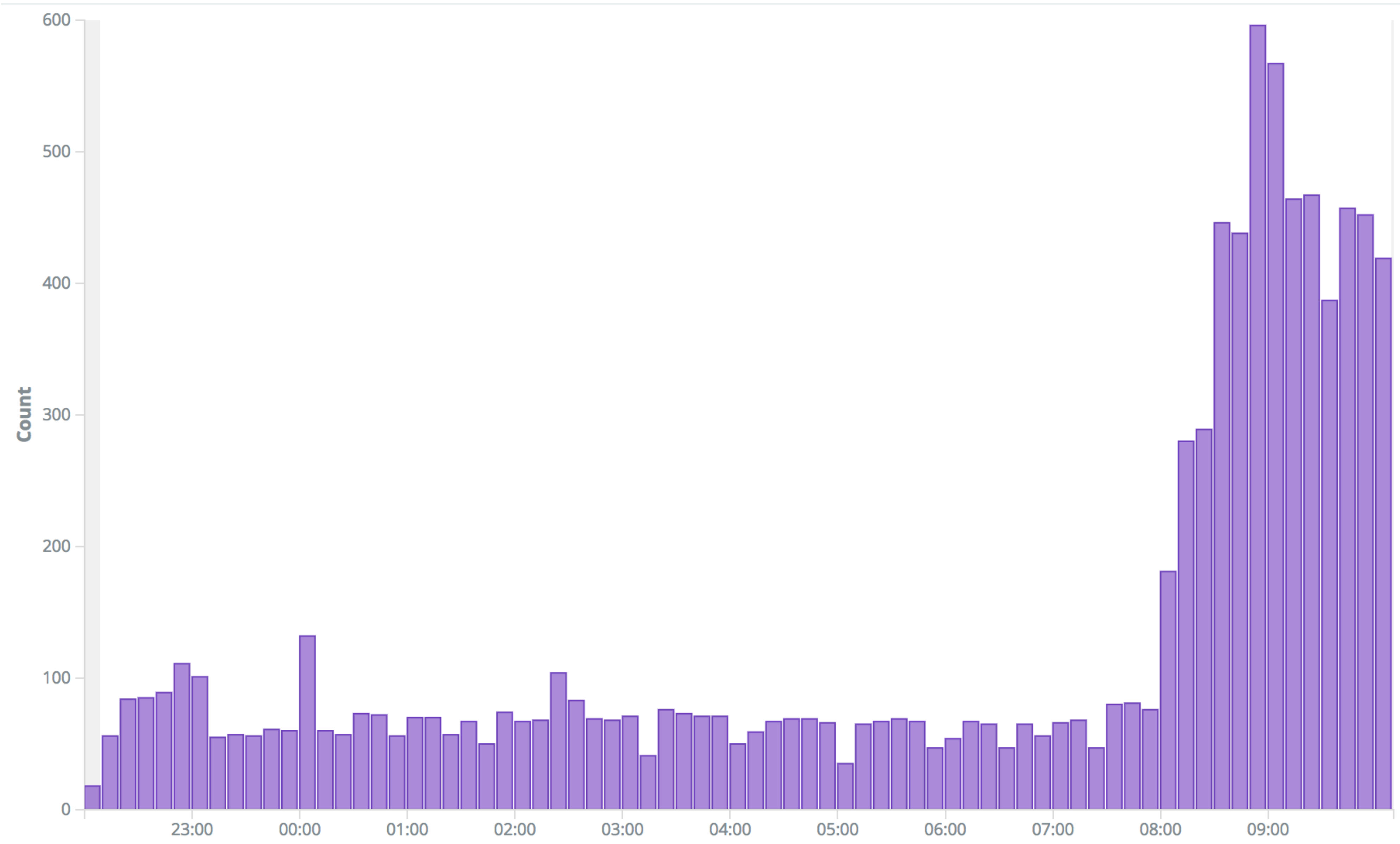
GeoIP

- Add Extra Information

Blacklist

- Mark Malicious traffic

Corporate Malicious Traffic



Machine Learning

Finding Similar Flows

- Supervised Learning
- Infected Flows as Train/Test data
- Classify flows based on learned data

Features for ML

- Malicious marked traffic
  - SRC IP
  - DST port
  - SRC port
  - Byte
  - Packets
  - Duration
  - ASN

Targets for ML

- Malicious Flows
  - Zeus
  - Mirai
  - any other malicious flow

Reduce False Positives

- Trusted Flows
  - DNS
  - HTTP
  - HTTPS
  - …

Scikit Learn

- Python based ML library
- Easy to use

Zeus (UDP) Case Study

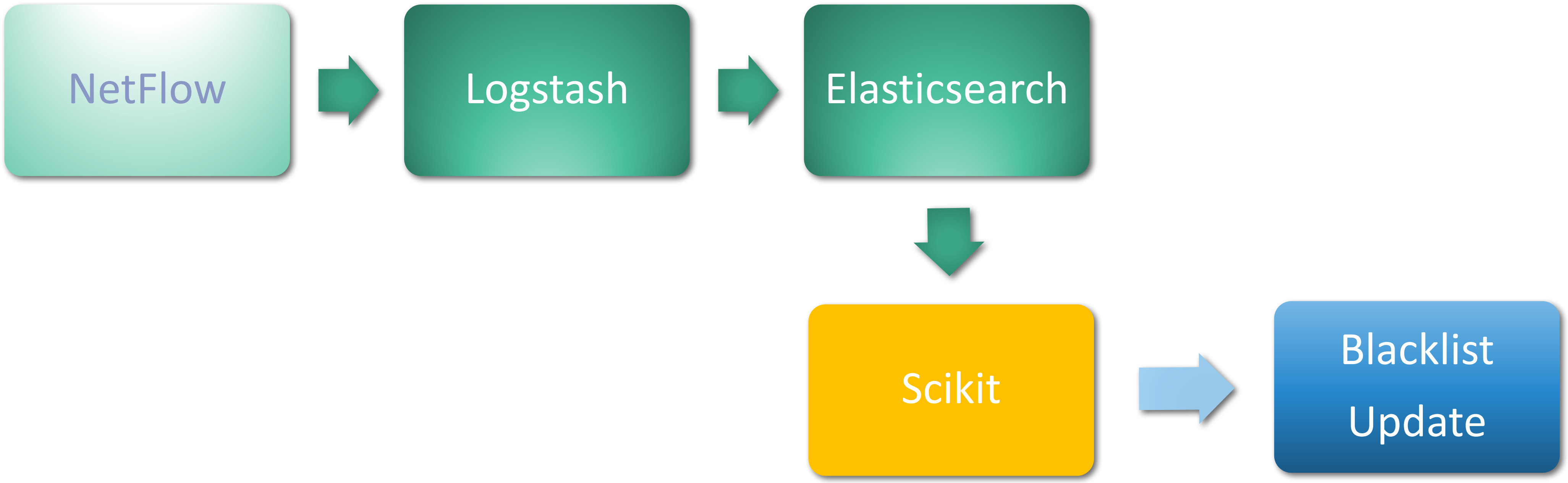
Classifier	Dataset	Train/Test	Accuracy
KNN – K=7	60000	50/50	82.9%
KNN – K=7	80000	50/50	86.8%
KNN – K=7	100000	50/50	89.3%

More data beats better algorithm!

Why not 100%

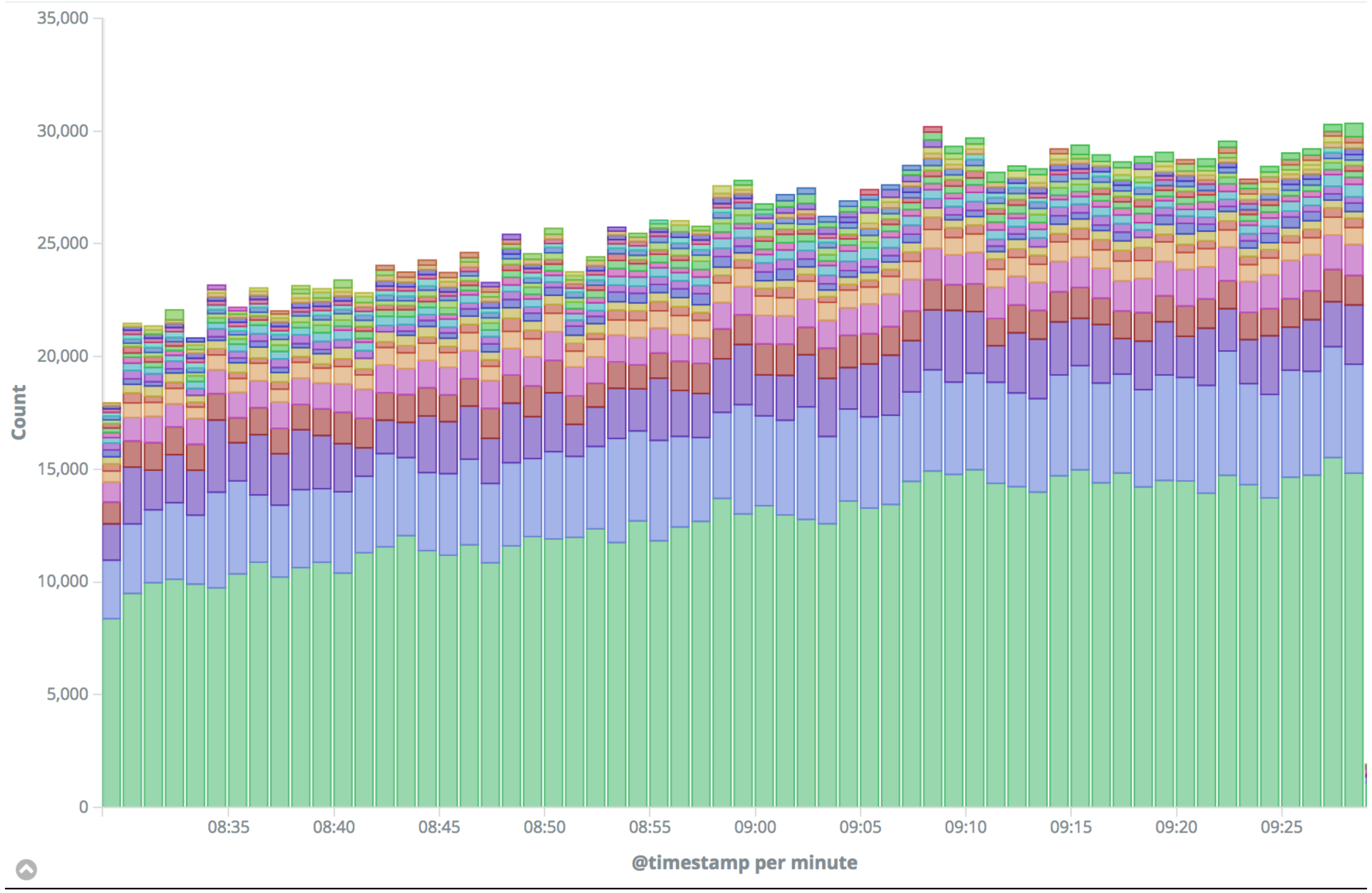
- Flows are unidirectional
- Flows are not classified into lifecycle steps
- Timeouts and retry
- Speed and Bandwidth
- Different versions of Zeus

Final Diagram



ASN whitelist

- Google
- Facebook
- Akamai
- Telegram



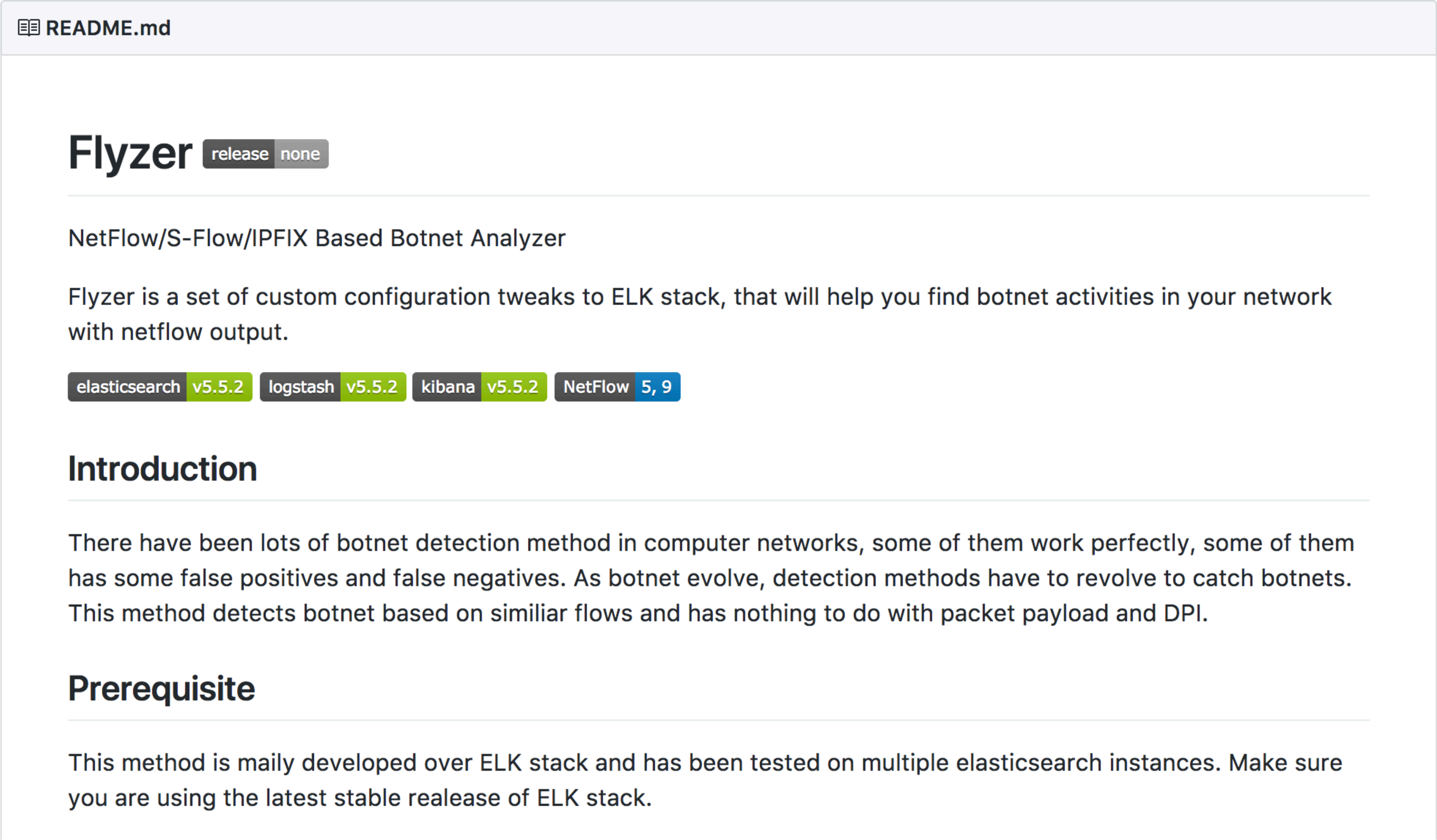
ToDo

- Bidirectional and related Flows
- ASN/Prefix reputation/anomaly
- Actions for detected botnets

Final words

- Netflow is cheap and handy
- Machine learning is amazing
- ML is the tool that will rescue us from internet threats

aliereza/flyzer



README.md

Flyzer release none

NetFlow/S-Flow/IPFIX Based Botnet Analyzer

Flyzer is a set of custom configuration tweaks to ELK stack, that will help you find botnet activities in your network with netflow output.

elasticsearch v5.5.2 logstash v5.5.2 kibana v5.5.2 NetFlow 5, 9

Introduction

There have been lots of botnet detection method in computer networks, some of them work perfectly, some of them has some false positives and false negatives. As botnet evolve, detection methods have to revolve to catch botnets. This method detects botnet based on similiar flows and has nothing to do with packet payload and DPI.

Prerequisite

This method is maily developed over ELK stack and has been tested on multiple elasticsearch instances. Make sure you are using the latest stable realease of ELK stack.

Questions
Comments