

# FlowHacker – An Approach for Detecting Unknown Network Attacks With Flows

Miguel Pupo Correia

joint work with Luís Sacramento

European Security Conference, Lisboa, June 2017



FCT Fundação para a Ciência e a Tecnologia  
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR



## Outline

- Motivation
- Approach
- Evaluation
- Conclusion

## Outline

- **Motivation**
- Approach
- Evaluation
- Conclusion

3

## Motivation

- Scenario:
  - Large national telco: mobile commun., Internet, TV,...
  - Connected to its own provider
  - Huge amount of traffic in/out, much is encrypted
  - Possibly new attacks / new variants



4

## Motivation

- Compromised **hosts** do attacks such as:
  - Distributed denial of service attacks
  - Exfiltrating confidential data
  - Sending spam
  - Mapping the network
  - Contact bot command&control centers
  - etc.

5

## Network Intrusion Detection Systems

- Traditional NIDSs:
- **Knowledge-based**: require signatures of attacks
  - Not good for new attacks
- **Behavior-based**: require clean traffic for training
  - Where to get it with our scenario?
- Most do **deep packet inspection**, unfeasible with too much traffic

6

## Outline

- Motivation
- **Approach**
- Evaluation
- Conclusion

7

## FlowHacker approach

- Detection framework to detect malicious hosts based on network traffic
- Not **knowledge-based**, to avoid need for signatures
- Not **behavior-based**, as no training traffic exists
- No **deep packet inspection**, as it is slow
- Detects hosts doing new attacks or new variants

8

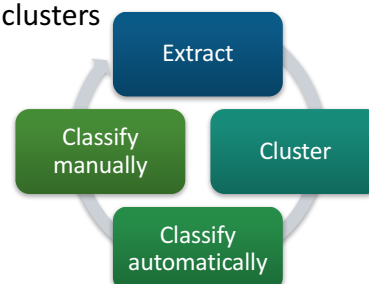
## Key ideas

- Collect traffic data summarized as **network flows**
- Extract data about **hosts** from flows
- Use **unsupervised machine learning / clustering**
  - to get information that humans can understand without previous knowledge about attacks
- Use **supervised machine learning / classifier**
  - to automatically assign clusters to classes/categories
  - ex: web servers, hosts doing distributed denial of service,...
- Manually label new clusters

9

## FlowHacker approach

- Loop:
  - Collect **flows** for a period of time (e.g., 1 day)
  - Extract from the flows data about **hosts** with MapReduce
  - Use **clustering** to create groups of hosts
  - Use **classifier** to automatically classify hosts
  - Manually label remaining clusters
  - Repeat for next period



10

## FlowHacker approach

- Loop:
  - Collect flows for a period of time (e.g., 1 day)

11

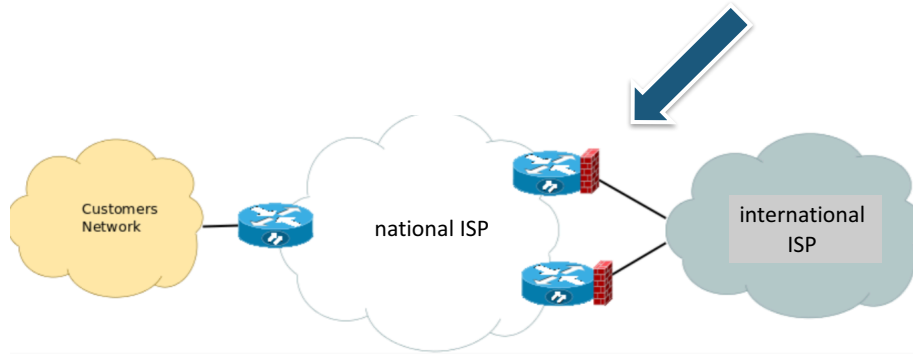
## Flows

- **Flow**: sequence of related packets observed during an interval of time
  - A **flow** is defined in terms of a subset of src IP, dest IP, protocol, src port, dest port; ex: *(\*, 1.2.3.4, TCP, \*, 80)*
- **Netflow**: monitoring approach created by Cisco
  - Idea is to capture data about network flows
  - Data: begin/end of flow timestamps, n. packets, n. bytes
  - Variants: IPFIX (standard based on Netflow 9), sFlow,...

12

## Flow collection

- Flows collected on NetFlow-enabled **border routers**



13

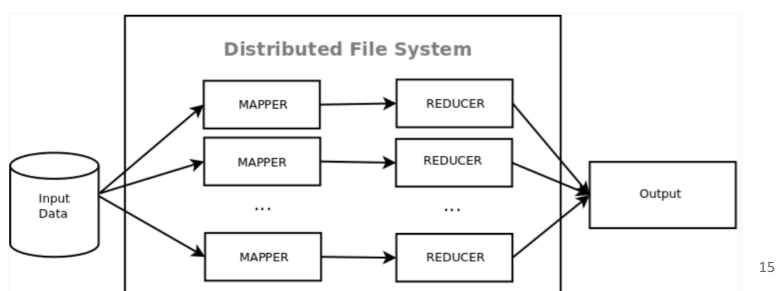
## FlowHacker approach

- Loop:
  - Collect **flows** for a period of time (e.g., 1 day)
  - Extract from the flows data about **hosts** with MapReduce

14

## Host data extraction

- Flow format:  
<Source IP, Destination IP, Source Port, Destination Port, Protocol, TCP Flags, #Bytes, #Packets, Duration>
- Use MapReduce for extracting data per **host (IP)**
  - aggregated by source or destination IP address



## Host data extraction

- Host **features** (data) extracted by MapReduce:

Feature	Description
<b>Aggregation Key</b>	The IP address that will be used as an identifier, to which the below features relate to
<b>NumSIPs / NumDIPs</b>	The number of IP addresses contacted
<b>NumSports</b>	The number of different source ports contacted
<b>NumDport</b>	The number of different destination ports contacted
<b>textbfNumHTTP</b>	The number of packets to/from port 80 (HTTP)
<b>NumIRC</b>	The number of packets to/from ports 194 or 6667 (IRC)
<b>NumSMTP</b>	The number of packets to/from port 25 (SMTP)
<b>NumSSH</b>	The number of packets to/from port 22 (SSH)
<b>TotalNumPkts</b>	The total number of packets exchanged
<b>PktRate</b>	The ratio of the number of packets sent and its duration
<b>ICMPRate</b>	The ratio of ICMP packets, and total number of packets
<b>SynRate</b>	The ratio of packets with a SYN flag and the total number of packets
<b>TotalNumBytes</b>	The overall sum of bytes
<b>AvgPktSize</b>	The average packet size
<b>BadSubnet</b>	This field expresses whether the IP address belongs to a blacklisted subnet
<b>MaliciousIP</b>	This field expresses whether the IP address is blacklisted
<b>OpenVaultBlacklistedIP</b>	Same as the above, but checked from a trusted and well know threat database
<b>MaliciousASN</b>	This field shows if the IP address belongs to a blacklisted ASN
<b>LocationCode</b>	Code for the country associated with the address

extracted from  
the flows directly

based on threat  
intelligence



## FlowHacker approach

- Loop:
  - Collect **flows** for a period of time (e.g., 1 day)
  - Extract from the flows data about **hosts** with MapReduce
  - Use **clustering** to create groups of hosts

17

## Unsupervised ML / clustering

- Idea: group similar hosts in clusters (sets)
- Why? Humans can understand and classify a few clusters, not zillions of hosts
- How?
  - Normalize every feature into range [0,1]
  - Run clustering algorithm, e.g., **K-Means**, to get **k** clusters
  - **k** can be defined, e.g., with the **elbow method** (finds the “elbow”, i.e., when adding more clusters does not improve the modelling of the data)

18

## FlowHacker approach

- Loop:
  - Collect **flows** for a period of time (e.g., 1 day)
  - Extract from the flows data about **hosts** with MapReduce
  - Use **clustering** to create groups of hosts
  - Use **classifier** to automatically classify hosts
  - Manually label remaining clusters

19

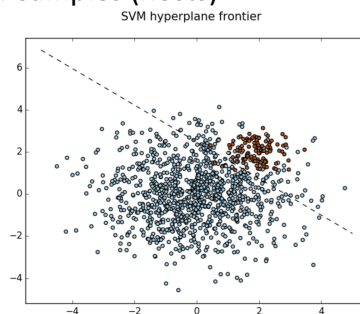
## Intrusion detection with flows

- Each cluster contains hosts with similar behavior
  - ex: web servers, mail servers, hosts sending spam, hosts doing denial of service,...
- What to do with them? (at cruise speed)
- Already seen? Use classifier to classify automatically
- Never seen?
  - Label manually, with help of the features' values
  - Focus attention on smaller clusters with odd feature distribution; often malicious
  - Retrain classifier

20

## Supervised ML / classification

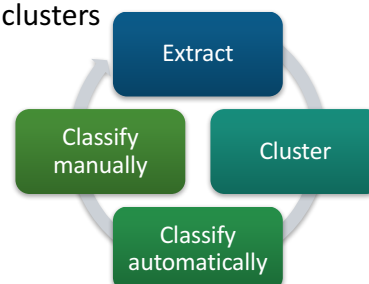
- Naïve solution: use labelled hosts to train a Binary Support Vector Machine (SVM) classifier
  - Samples/hosts classified as benign or malicious
  - Finds an hyperplane that separates samples
  - Classifies new samples (hosts)



21

## FlowHacker approach

- Loop:
  - Collect **flows** for a period of time (e.g., 1 day)
  - Extract from the flows data about **hosts** with MapReduce
  - Use **clustering** to create groups of hosts
  - Use **classifier** to automatically classify hosts
  - Manually label remaining clusters
  - Repeat for next period



22

## Outline

- Motivation
- Approach
- **Evaluation**
- Conclusion

23

## FlowHacker Tool – interface

```

Welcome to the NIDS of the Future!

First of all, which dataset will you analyze?
Source - 1
Destination - 2
$ 1

=====
||          SOURCE AGGREGATION          ||
||          ANALYSIS                    ||
=====

Please choose wich clustering technique you wish to apply
MiniBatch - 1
K-Means - 2
$ 2

You are analyzing Monday, which contains 448 entries

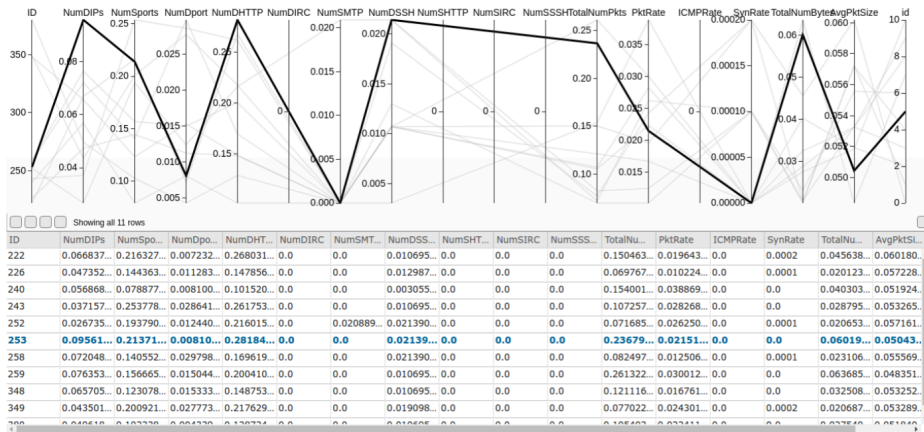
You are now in the clustering section. What would you like to do?
1. See description of the dataset
2. Choose the optimal number of clusters
3. Visualize the clusters
4. Analyze clusters
5. Open web interface
6. Validate attacks (from database ground truth)
7. Plot two features
8. Change dataset
9. Change clustering type
0. Exit

What will it be?
$ █

```

24

## Tool – interactive visualiz. of cluster



25

## Evaluation

- Two parts:
  - Synthetic dataset (ISCX)
    - Designed for IDSs
    - Flows are labelled
    - Allows validating the approach
  - Real dataset collected at the telco
    - No ground truth

26

## ISCX dataset evaluation

Clusters	1	2	3	4	5	6	7	8	9	10
# Entries	1	5	1	1	5	5	16	8	3	1
Features										
1	-	-	-	-	-	-	-	-	-	-
2	-	0.125; 0.023	-	1.0	0.243; 0.064	-	-	0.200; 0.016	0.277; 0.076	0.115
3	-	-	-	-	-	-	-	-	-	0.227
4	-	0.131; 0.023	-	-	0.418; 0.56	-	-	0.213; 0.031	-	-
5	-	-	-	-	-	-	-	-	-	-
6	-	-	-	0.325	-	-	-	-	-	-
7	-	-	1.0	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	0.305	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	0.128; 0.005	0.184

Avg ; StdDev

Low values

Saturday clusters

- Brute-Force SSH attack found during this day (cluster 3)
  - Maximum for SSH connections (and high, not seen in table)

27

## Telco dataset evaluation

Cluster #	1	2	3	4	5	6	7	8	9	10	11	12	13	14
# Hosts	1605	51773	6485	13305	529	1730	1729	21507	8523	8522	1498	4686	10	5653
Features														
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	0.368; 0.138	-
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	0.61; 0.208	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Cluster data with source aggregation key (i.e., aggregated by IP inside the telco) cluster data – 1<sup>st</sup> part

28

## Telco dataset evaluation

Cluster #	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
# Hosts	1	824	5	4606	1864	1676	12	107	13	2233	2233	8091	10	13897	16843	35
Features																
1	1.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	1.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	0.667	-	-	-	-	-	-	-	0.384; 0.184	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.237; 0.158
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	1.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	0.542; 0.18	-	-	-	-	-	-	-	-	-
10	-	-	0.626; 0.21	-	-	-	-	-	-	-	-	-	-	-	-	-
11	1.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	0.843	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	0.261; 0.06	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Cluster data with source aggregation key – 2<sup>nd</sup> part

29

## Source aggregation key – cluster 15

i.e., host(s) in the telco's network

Cluster #	15	Cluster #	15
# Hosts	1	# Hosts	1
Features		Features	
1	1.0	9	-
2	-	10	-
3	1.0	11	1.0
4	-	12	-
5	0.667	13	-
6	-	14	-
7	-	15	0.843
8	0.843	16	-

- Spammer or denial of service (?)
  - High connectivity to various users, many ports, receiving communication on IRC port, communication through HTTP, high number of packets sent, high number of bytes

30

## Source aggregation - Cluster 21

Cluster #	21	Cluster #	21	
# Hosts	12	# Hosts	12	
Features		Features		
1	-	9	0.541 ; 0.181	High IRC communication
2	-	10	-	
3	-	11	-	
4	-	12	-	
5	-	13	-	
6	-	14	-	
7	-	15	-	
8	-	16	0.261 ; 0.026	High average packet size

- Bot communicating with C&C server
  - Confirmed by accessing the IP of the C&C server

31

## Telco dataset evaluation summary

Cluster #	Aggregation Key	Highlighted Features	Type of Attack
15	Source	1, 3, 5, 8, 11, 15	Spam / DoS
16	Destination	1, 3, 6	DoS
17	Source	10	Brute-Force SSH
20	Destination	1, 2, 15	Network Scan
21	Source	9, 16	Botnet Communication
22	Destination	1, 3, 8, 15	Web Application Probing
27	Source	1, 2, 5, 8, 11, 15	DDoS IRC Botnet
29	Destination	1, 2, 4, 11, 15	DDoS Botnet

32



## Outline

- Motivation
- Approach
- Evaluation
- **Conclusion**

33

## Conclusion

- FlowHacker: Network Intrusion Detection for identifying malicious hosts using flows
- ...without having to say how entities misbehave
- Use clustering (unsupervised ML) to reduce the size of the problem and
- a classifier (supervised ML) to automatize classification
- Keep humans in the loop; mandatory w/evolving threats
- Detects attacks involving many packets, not low traffic attacks like buffer overflows or SQL injection

34

## Thanks! Questions?

Daniel Gonçalves, João Bota, Miguel Correia. **Big Data Analytics for Detecting Host Misbehavior in Large Logs.** TrustCom 2015

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 (INESC-ID)

**FCT** Fundação para a Ciência e a Tecnologia  
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

