Detecting Distributed Attacks in SDN Based on Automatic Thresholding

Yaokai Feng@WP3

(Kyushu University)

2018.09.11@IITD

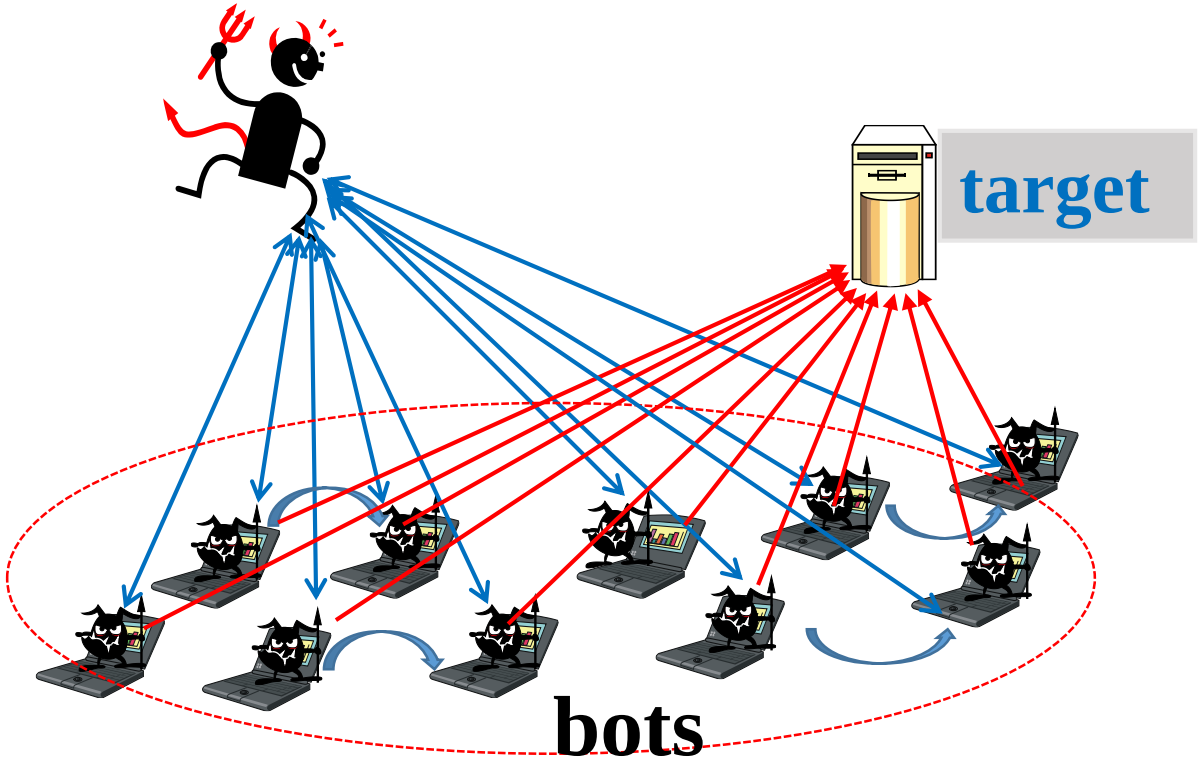
A Joint work with

Mr. KOMIYA (Master student)

Prof. Sakurai

Why **Distributed Attacks** ?

What is distributed attack?



Attacker controls a lot of hosts (called bots) to launch attacks collaboratively.

A report

Also said "caused **the largest damages** in recent years

from http://opensources.info/ddos-attack-mitigated-10-da... causes 650 gbps/

YEAR	ACCOUNT HIJACKING	TARGETED ATTACK	DISTRIBUTED ATTACK	SQL INJECTION	MAL-WARE
2016	15.1%	11.6%	11.3%	8.4%	8.0%
2015	8.8%	10.5%	9.7%	17.5%	6.4%
2014	9.8%	9.5%	9.3%	12.8%	9.4%

one of
the most sophisticated and **the most dangerous**
kinds of attacks **for many years**

Some examples

- Mar. 20, 2013
  to Spamhaus Co.
  Peak traffic: about **300Gbps (64 full-4.7GB-DVD**)

- Feb. 13, 2014
  to Cloudfare
  Peak traffic: about **400Gbps**

- Sept. 20, 2016
  to Krebs on Security
  Peak traffic: **620Gbps**
  (from 1.8M IoT devices infected by “Mirai” virus)

Some other examples

- 2000, a wave of DDoS attacks to Yahoo and Amazon, over **$1.2 billion** in damages.

- 2011, DDoS attack to Sony,
  Sony said: about **$170M** (on cle
  Many others said: **much larger** that that

Also, reputation loss

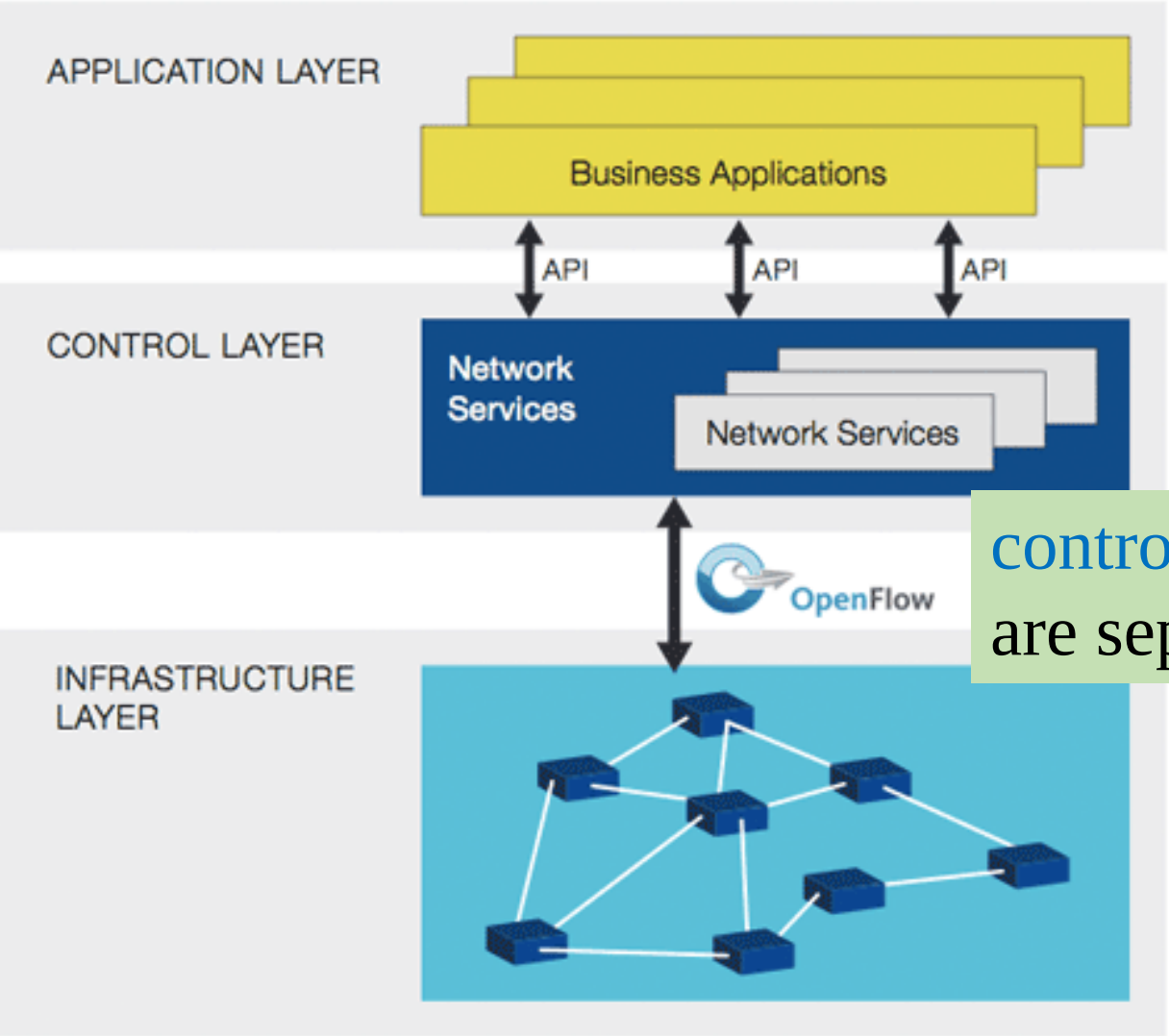
- 2011, a large botnet was smashed, more than **4,000,000** bots

- 2013 two servers in USA controlled more than 1,000 bots
  about **$500M** in consumer and business losses

- 2016, 13 root servers, around 3 hours.
  availability averaged across all root servers **down to around 50%**

Why **SDN** ?

SDN framework



control plane & data plane are separated

https://www.sdxcentral.com/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/

Cloud, SDN in IoT world

Cloud services: storage, computation, … …

the ***heart*** of the connected IoT world

a key role for smart homes/buildings/cities

SDN

- makes cloud infrastructure more efficient & flexible
- makes it easier for organizations to host applications

SDN security is critically important for cloud security in IoT world

SDN has been use

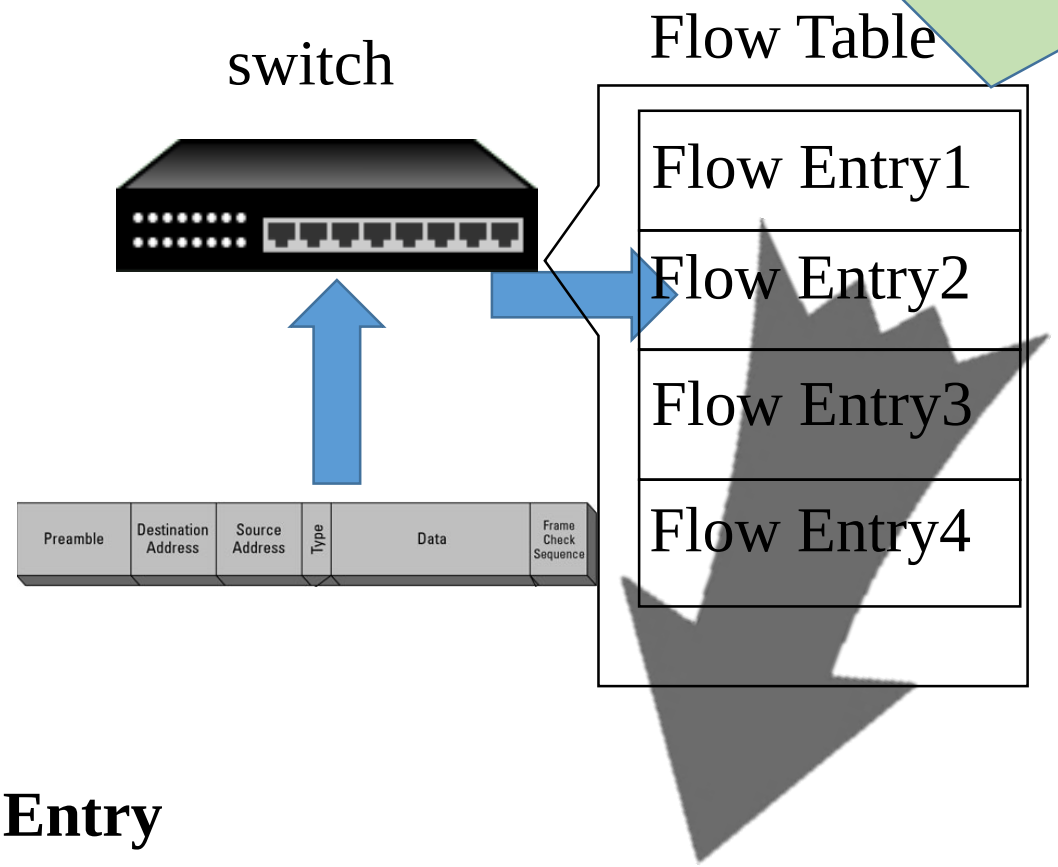
- Microsoft's cloud,
- NTT's cloud,

and many other cloud syst

WP3's Objective:
Cloud Security for IoT

Some details

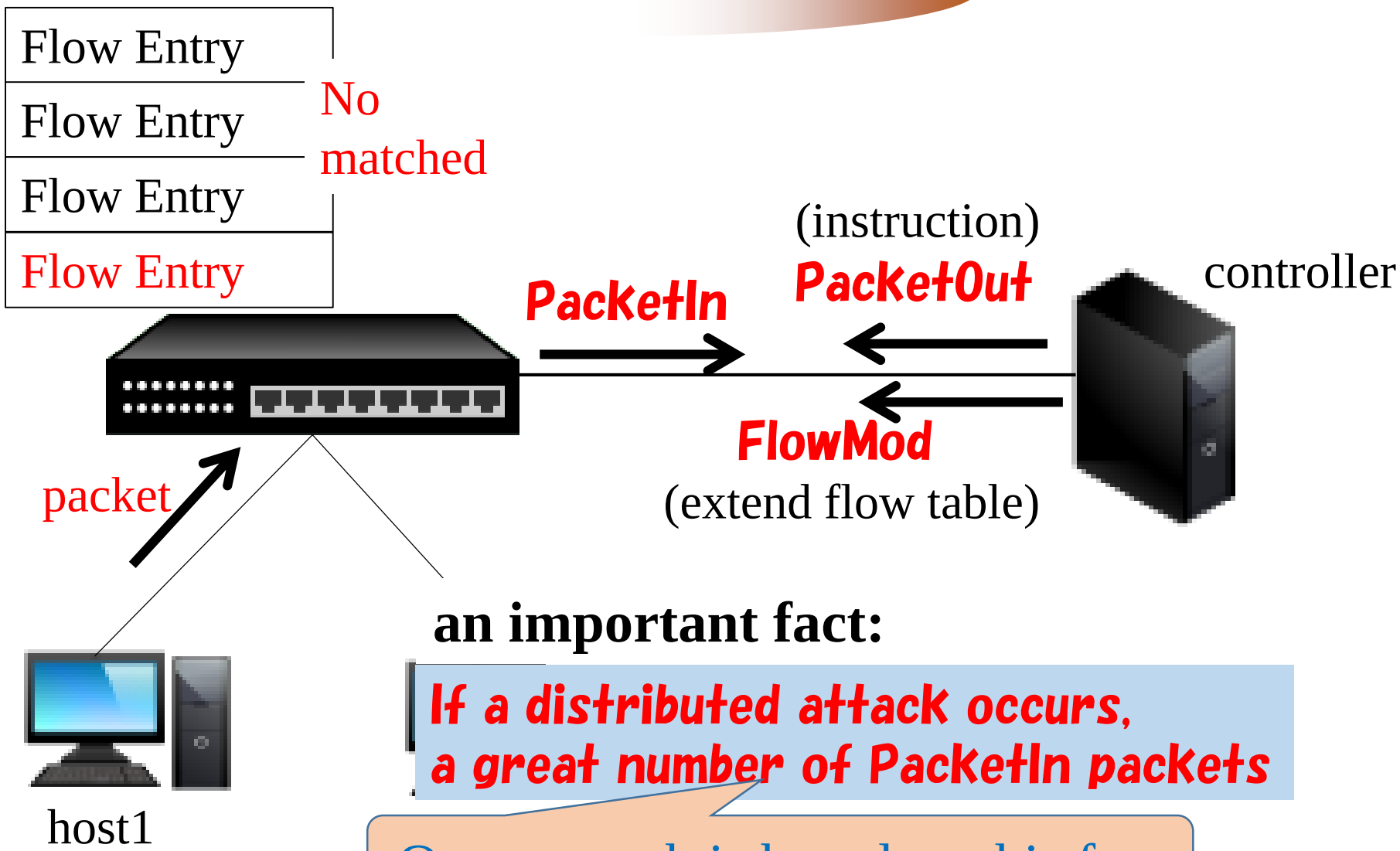
Where did these flow entries come?



Flow Entry

Priority	Match Field	Instruction	… …

PacketIn & PacketOut



an important fact:

If a distributed attack occurs, a great number of PacketIn packets

Our research is based on this fact

Why **Automatic Thresholding** ?

If a distributed attack occurs,
a great number of PacketIn packets

This fact can be used to pick out such attacks

a threshold is needed

pre-defining such a threshold

not easy!

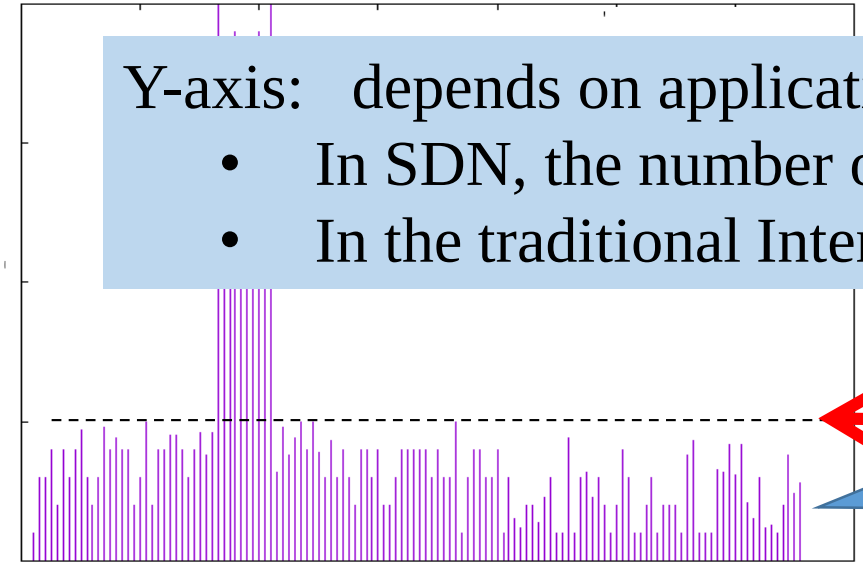
Our proposal

The threshold is extracted from the historical data

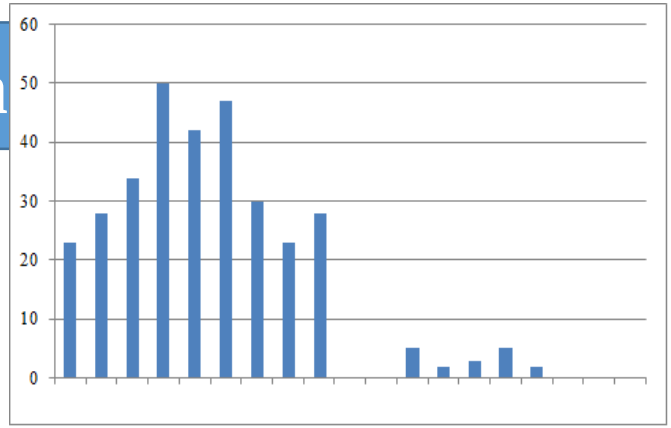
X-axis: time

Y-axis: depends on applications

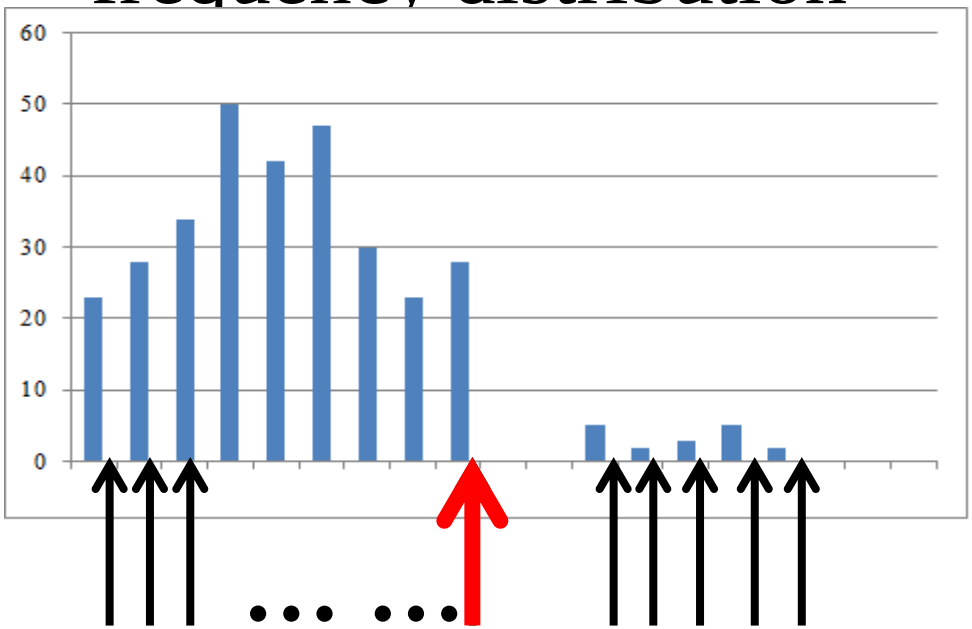
- In SDN, the number of PacketIn packets
- In the traditional Internet, the number of unique source IPs



frequency distribution



frequency distribution



Score of the position *i*

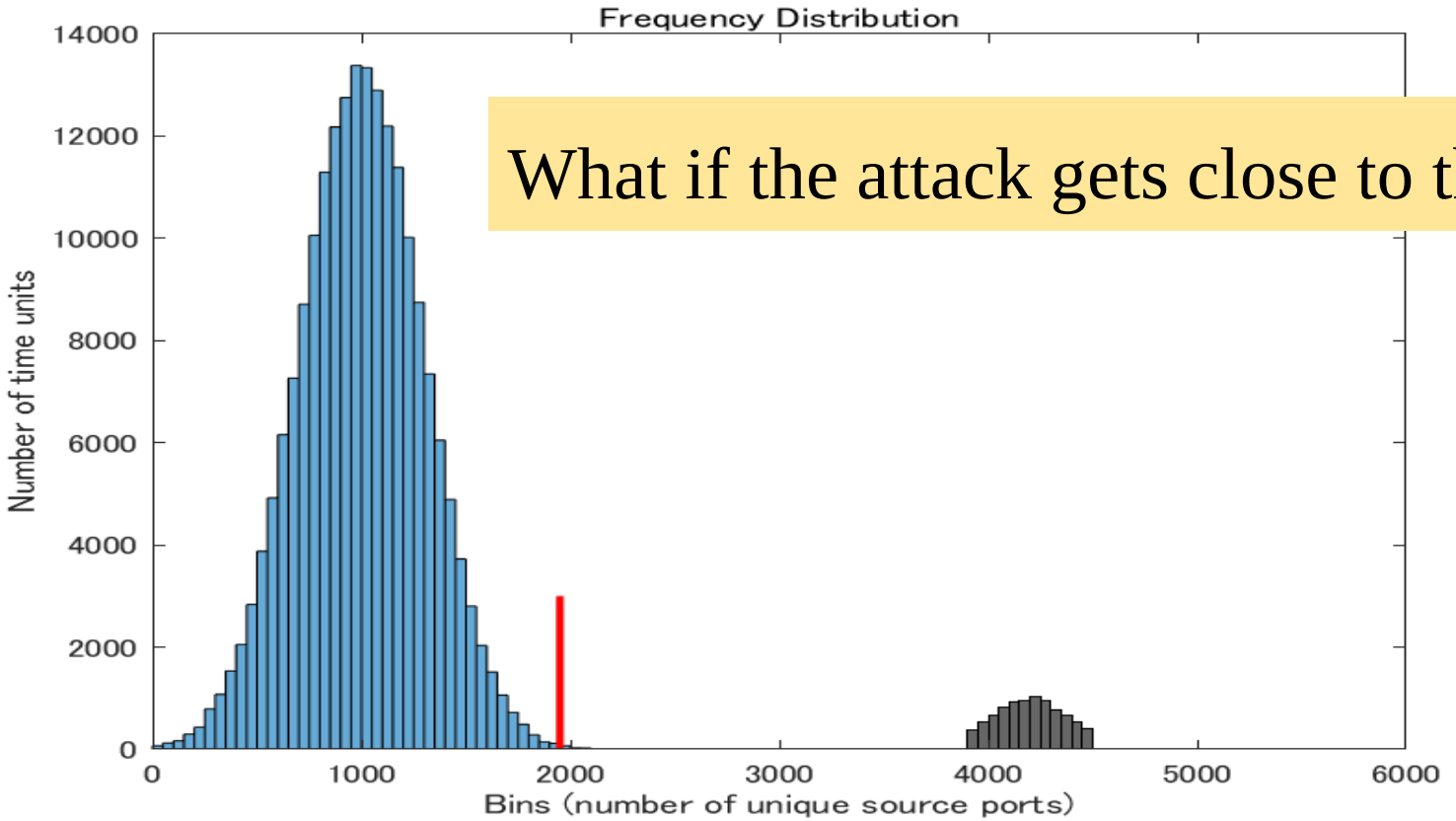
$$S_i = \frac{\sum_{q=i}^{n} \frac{f_q}{q}}{n-i}$$

$$Final\ result = argmin\ S_i$$

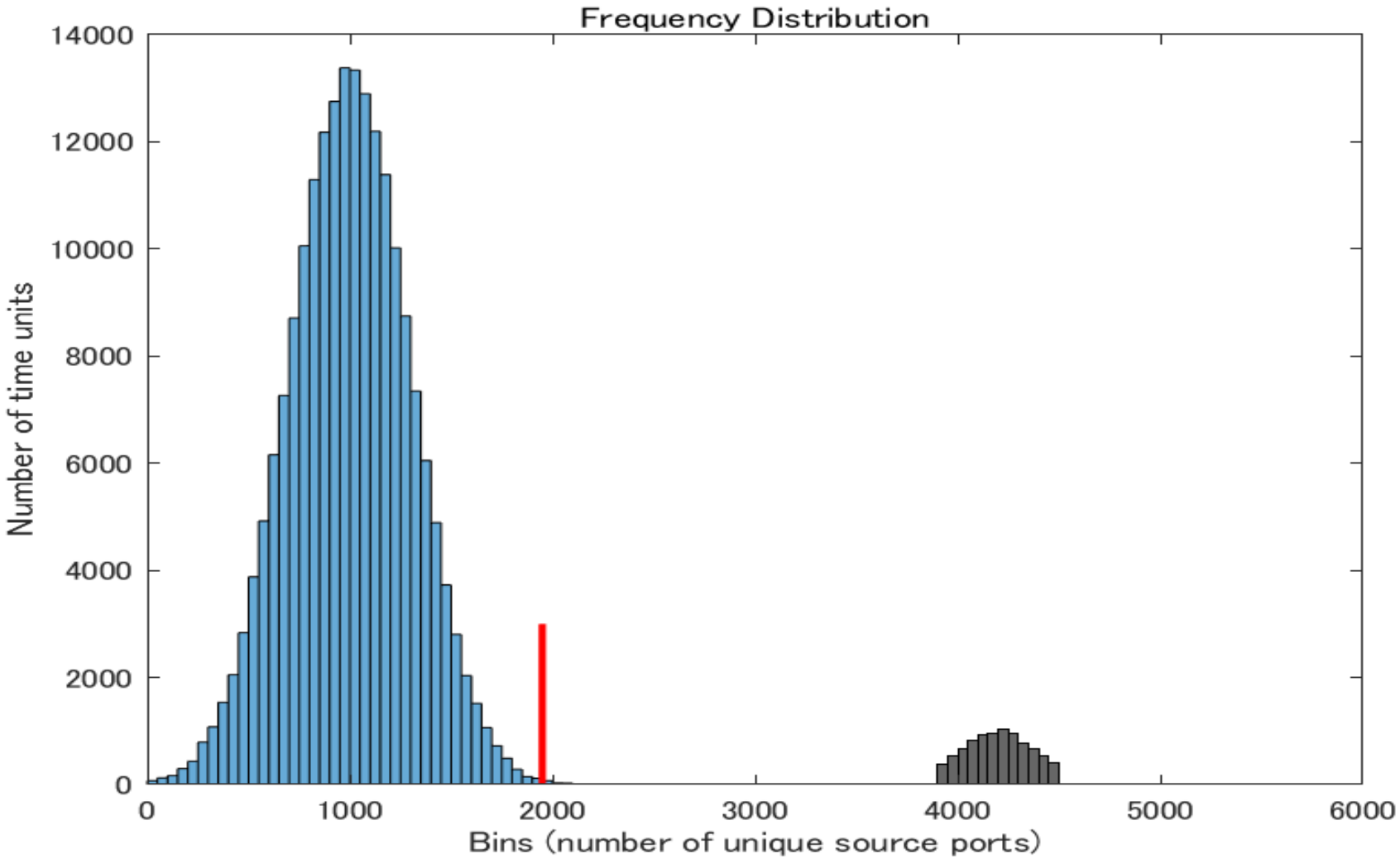
$$1 \le i < n$$

- Every *possible position* **is scored** by an ***evaluation function***
- **The final result** is decided according to those scores

Behavior test of our threshold-extraction

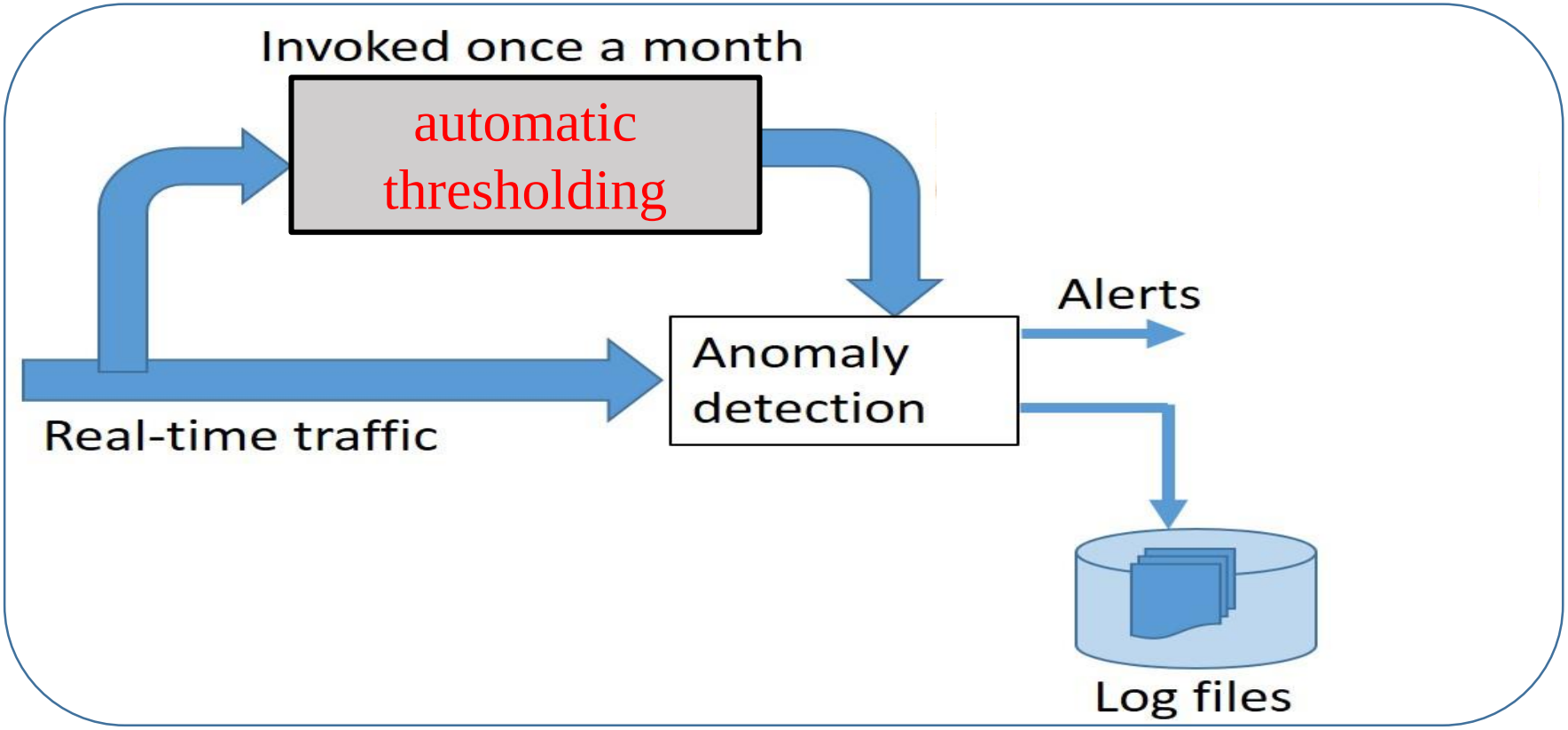


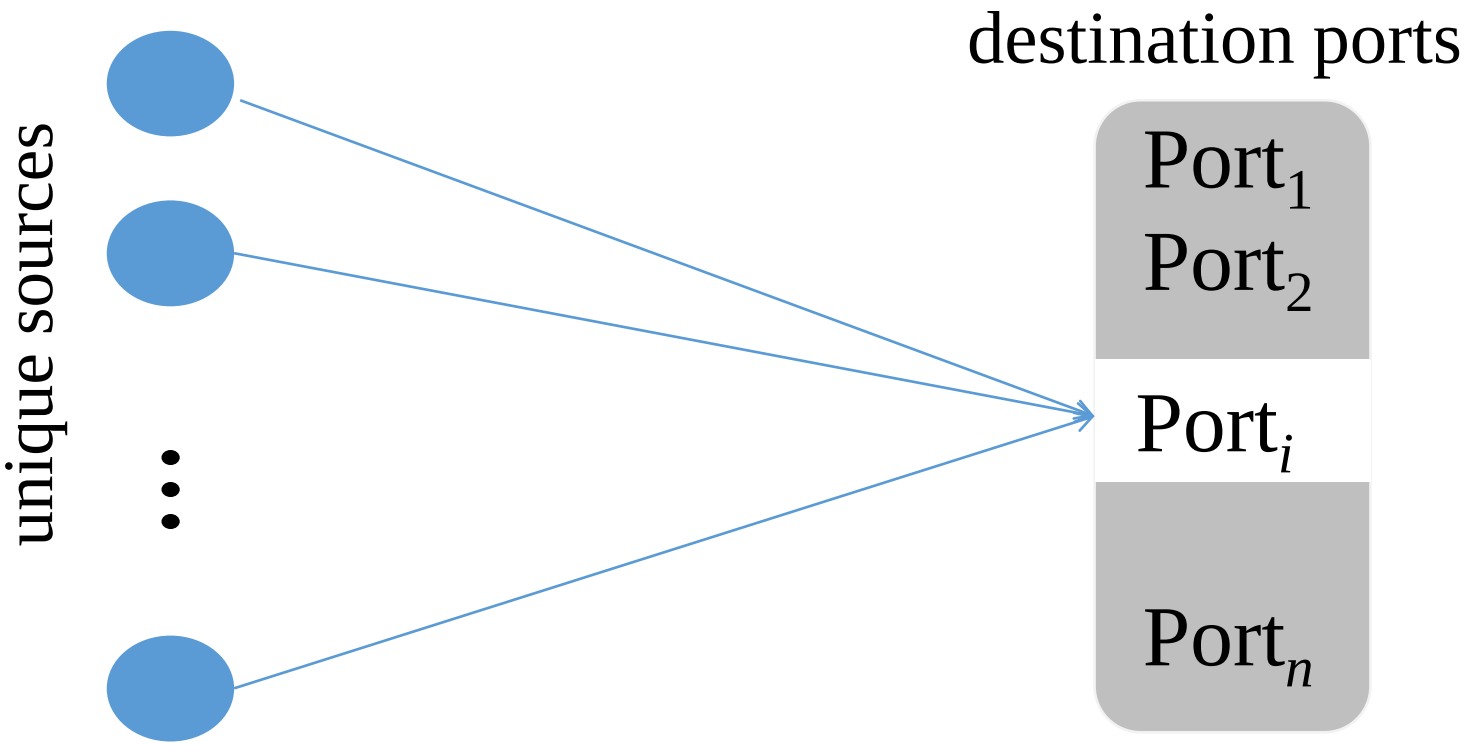
What if the attack gets close to the normal?


Frequency Distribution
Number of time units
14000
12000
10000
8000
6000
4000
2000
0
0
1000
2000
3000
4000
5000
6000
Bins (number of unique source ports)

Our automatic threshoulding works well in traditional network (our previous research)

Yaokai Feng, Yoshiaki Hori, Kouichi Sakurai, “A Detection System for Distributed DoS Attacks Based on Automatic Extraction of Normal Mode and Its Performance Evaluation”, *the 10th International Conference on Security, Privacy and Anonymity (SpaCCS2017), LNCS 10656, pp. 461-473, 2017.12*


Invoked once a month
automatic thresholding
Alerts
Anomaly detection
Real-time traffic
Log files


destination ports
unique sources
Port$_1$
Port$_2$
Port$_i$
Port$_n$

Results (examples)

alert list

```
2015/10/01 04:05:00 TCP 10001 60 15 32
2015/10/01 11:00:00 TCP 12345 90 15 177
2015/10/02 00:00:00 TCP 32075 60 5 11
2015/10/02 04:35:00 TCP 10001 60 15 36
2015/10/02 19:10:00 TCP 11238 80 5 25
2015/10/03 10:00:00 TCP 34124 60 5 14
2015/10/04 04:00:00 TCP 51817 60 5 15
2015/10/04 21:55:00 TCP 13138 60 5 11
2015/10/05 08:00:00 TCP 49115 60 10 30
2015/10/05 08:00:00 TCP 42893 60 5 15
2015/10/05 12:10:00 TCP 58455 70 60 236
2015/10/05 12:10:00 TCP 51768 70 5 17
2015/10/05 16:20:00 TCP 58470 60 5 11
2015/10/05 18:25:00 TCP 51768 70 5 16
2015/10/05 20:45:00 TCP 31337 80 5 104
2015/10/06 00:00:00 TCP 31337 90 5 206
2015/10/06 16:00:00 TCP 5555 90 15 207
2015/10/06 16:00:00 TCP 6666 90 15 203
2015/10/07 20:05:00 TCP 7022 70 10 32
2015/10/08 00:00:00 TCP 32054 80 5 23
```

source-IPs of each alert

```
SensorID=sensor053
ObservationTime=2016/01/26-04:00:00
PortID=0/TCP
AlertLevel=60
Unique SourceHosts=
*.22.36.3, *.49.35.5, *.186.168.8, *.0.180.3, *.19.4
15.6.213, *.63.22.34, *.214.134.3, *.130.140.36, *.2
 *.108.77.38,
(11)

SensorID=sensor053
ObservationTime=2016/01/26-04:05:00
PortID=0/UDP
AlertLevel=70
Unique SourceHosts=
*.94.66.33, *.236.147.43, *.127.0.56, *.142.51.74, *
5, *.148.116.29, *.249.139.56, *.63.22.63, *.30.40.6
77.3, *.110.247.58, *.110.161.63, *.192.226.37, *.11
 *.251.122.3, *.128.230.75, *.13.121.43, *.98.96.98,
(18)
```

global alerts and their source-IPs

```
2015/10/01-04:05:00
PortID:  53413/TCP
sensor053  sensor057  senso-058  sensor062
Source IP list:
*.6.167.142,  *.6.216.35,  *.6.216.53,  *.6.216.57,  *
38,  *.6.216.39,  *.6.216.55,  *.3.216.46,  *.6.216.43
216.51,  *.6.216.58,  *.6.216.40,  *.6.216.49,  *.6.21
*.6.216.59,  *.6.2'6.62,  *.6.216.53,  *.6.216.45,  *.6
7,  *.6.216.44,  *.240.192.138, *.6.135.131,  *.6.216.6
6.216.48,  *.6.216.54,  *.6.216.50,  *.6.216.60,  *.6.2
(28)

2015/10/01-08:10:00
PortID:  32764/TCP
sensor053  sensor056  senso-057  sensor058  sensor062
103
Source IP list:
*.6.216.57,  *.6.2'6.39,  *.6.216.52,  *.6.216.59,  .6
,  *.6.216.54,  *.6.2'6.46,  *.6.216.43,  *.6.216.56,
6.44,  *.6.216.62,  *.6.216.38,  36.240.192.138,  *.6.2
 *.6.216.61,  *.6.216.42,  *.6.216.40,  *.6.216.45,  *
.60,  *.6.216.49,  *.6.216.48,  *.6.216.36,  *.6.216.55
```

Detection 1 (our previous research)

Datasets:

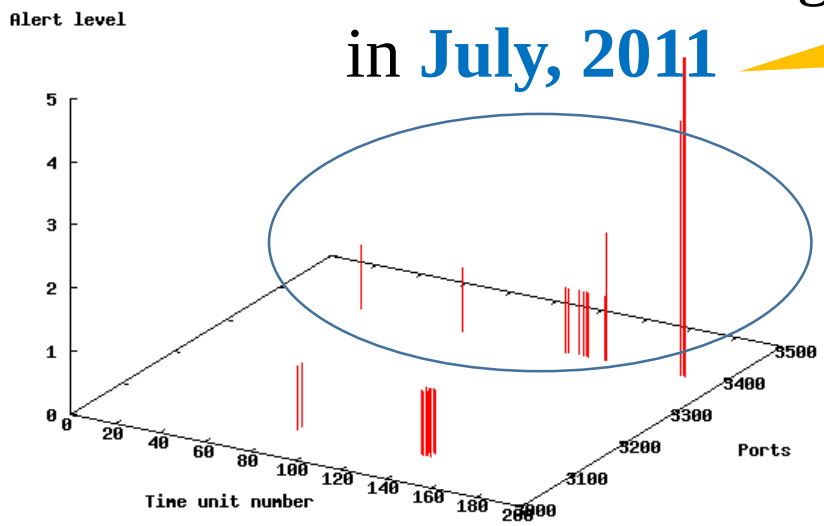
- The traffic data in June and July, 2011, of the Darknet NICTER
- NICTER is built and managed by NICT, Japan

- The traffic data in June was used as training data
- The traffic data in July was used as test data

Detection case 1

Morto worm was reported by several security companies from Aug. 10, 2011

Our detection result using da
in **July, 2011**



We captured it
at 20:00 in July 14

22	12:0 20:00
23	~ 0
24	12:00
26	0 0~12:00
31	0 0~12:00

Detection case 2

An IoT botnet was reported by National Police Agency, Japan,

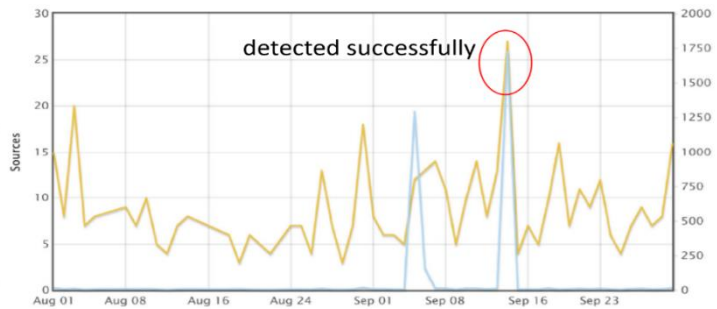
on Dec. 15, 2015,

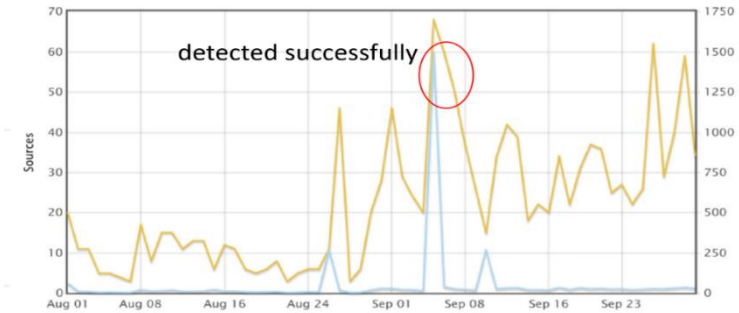
Our detection result in Nov. 2015

We captured it

#global alerts	Details
44	53413/UDP（14 times） 23/TCP（29 times） 32764/TCP （1 time, secret backdoo

Some other cases (based on SANS's observation)





How does our proposal behave in SDN ?

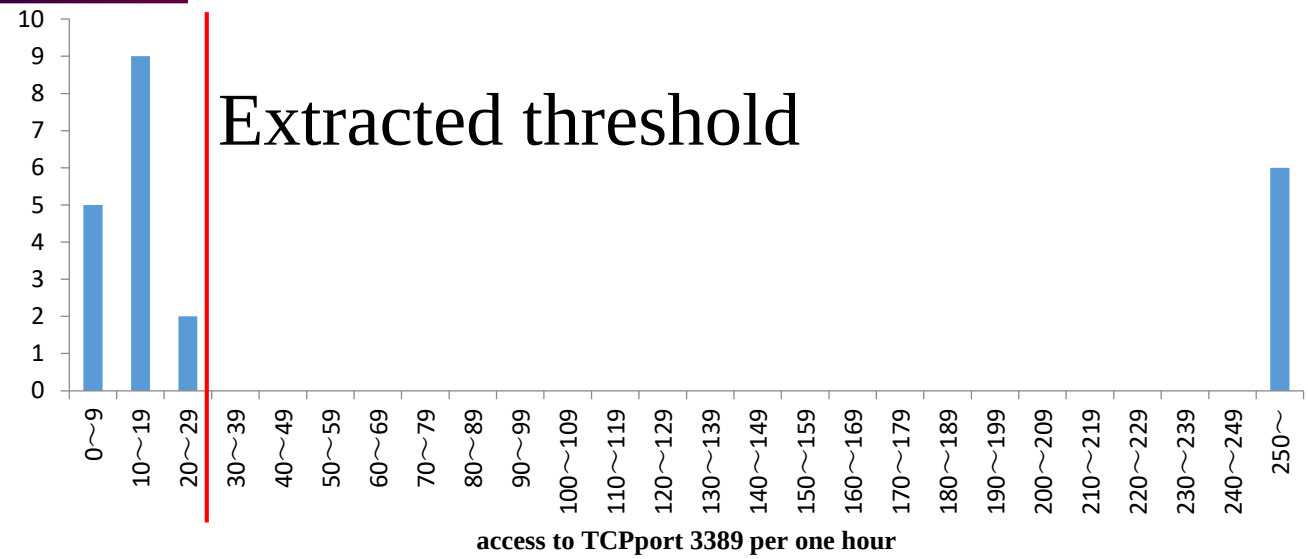
(very preliminary experiment)

Environment & datasets

- Simulated SDN environment: Mininet
- Datasets: Traffic data from the Darknet NICTER (managed by NICT, Japan)
- Training data:
  the dataset on June 30, 2011
- Test data:
  the datasets on July 14 & 30
- All the traffic data were re-produced in the SDN environment using TCPreplay

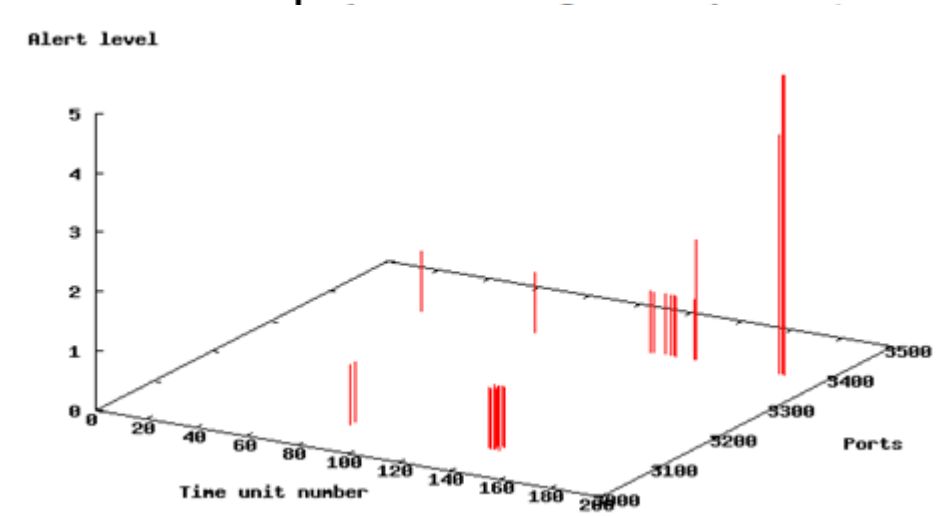
timestamps of PacketIns

- 1534998236.61
- 1534998236.61
- 1534998236.61
- 1534998236.87
- 1534998236.87
- 1534998236.87
- 1534999298.15
- 1534999298.15
- 1534999298.15
- 1534999301.07
- 1534999301.07
- 1534999301.07
- 1534999307.1
- 1534999307.1
- 1534999307.1
- 1535000432.53
- 1535000432.54
- 1535000432.54
- 1535000435.54
- 1535000435.55
- 1535000435.55
- 1535000441.57



Detection result

dataset	Total PacketIn mes	First time detection (from
7/14/2011	867	
7/31/2011	137	



Summary

Focus:

Detecting distributed attacks in SDN

Originality:

Based on automatic thresholding

Observations:

Our preliminary experiment indicates that our proposal works well

Future work

- to examine the behavior of our proposal in more detail using other datasets

- to analyze the experiment result and try to improve the threshold extraction algorithm if necessary

Thanks