SSH Compromise Detection using NetFlow/IPFIX

Rick Hofstede, Luuk Hendriks

UNIVERSITY OF TWENTE.

REDSOCKS
malware threat defender

“51 percent of respondents admitted that their organizations have already been impacted by an SSH key-related compromise in the last 24 months.”

–Ponemon 2014 SSH Security Vulnerability Report

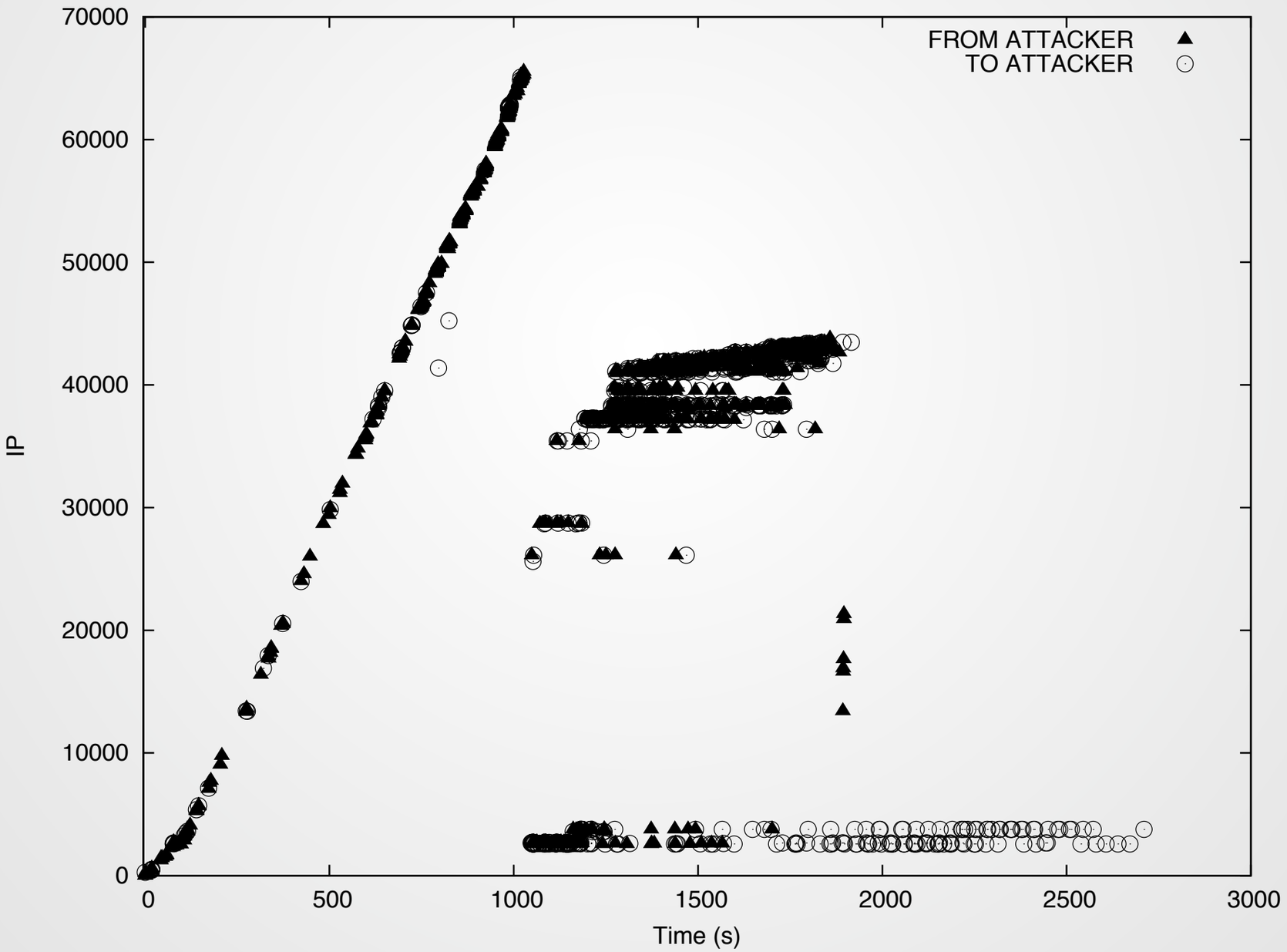
SSH Compromise Detection using NetFlow/IPFIX

Rick Hofstede, Luuk Hendriks

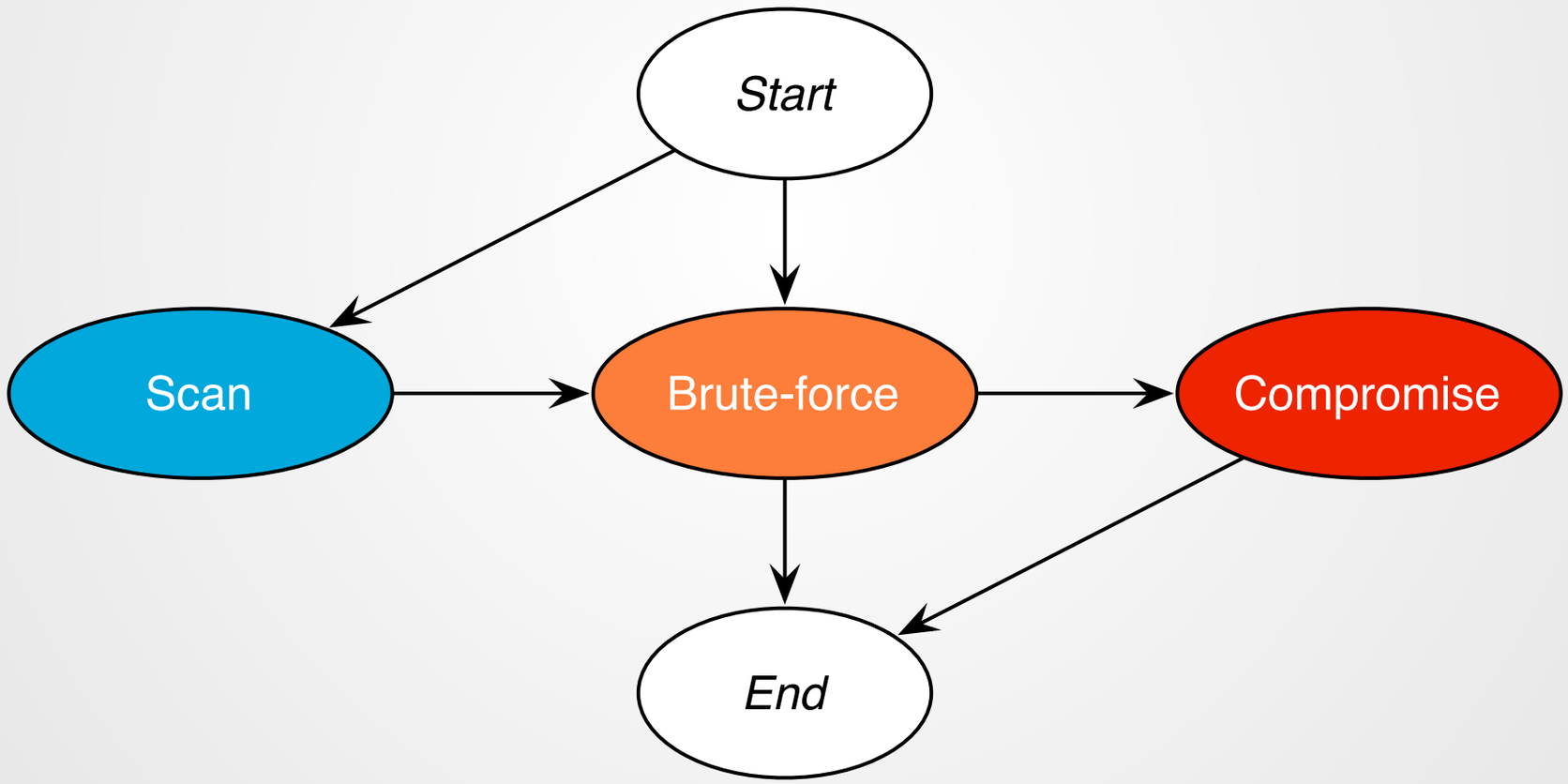
UNIVERSITY OF TWENTE.

REDSOCKS
malware threat defender

SSH attacks


FROM ATTACKER
TO ATTACKER
IP
Time (s)
0
10000
20000
30000
40000
50000
60000
70000
0
500
1000
1500
2000
2500
3000

SSH attacks



SSH attacks

- SSH intrusion detection on end hosts is hardly scalable
- Network-based approaches exist, but only inform security operators about the presence of attacks

We perform compromise detection.

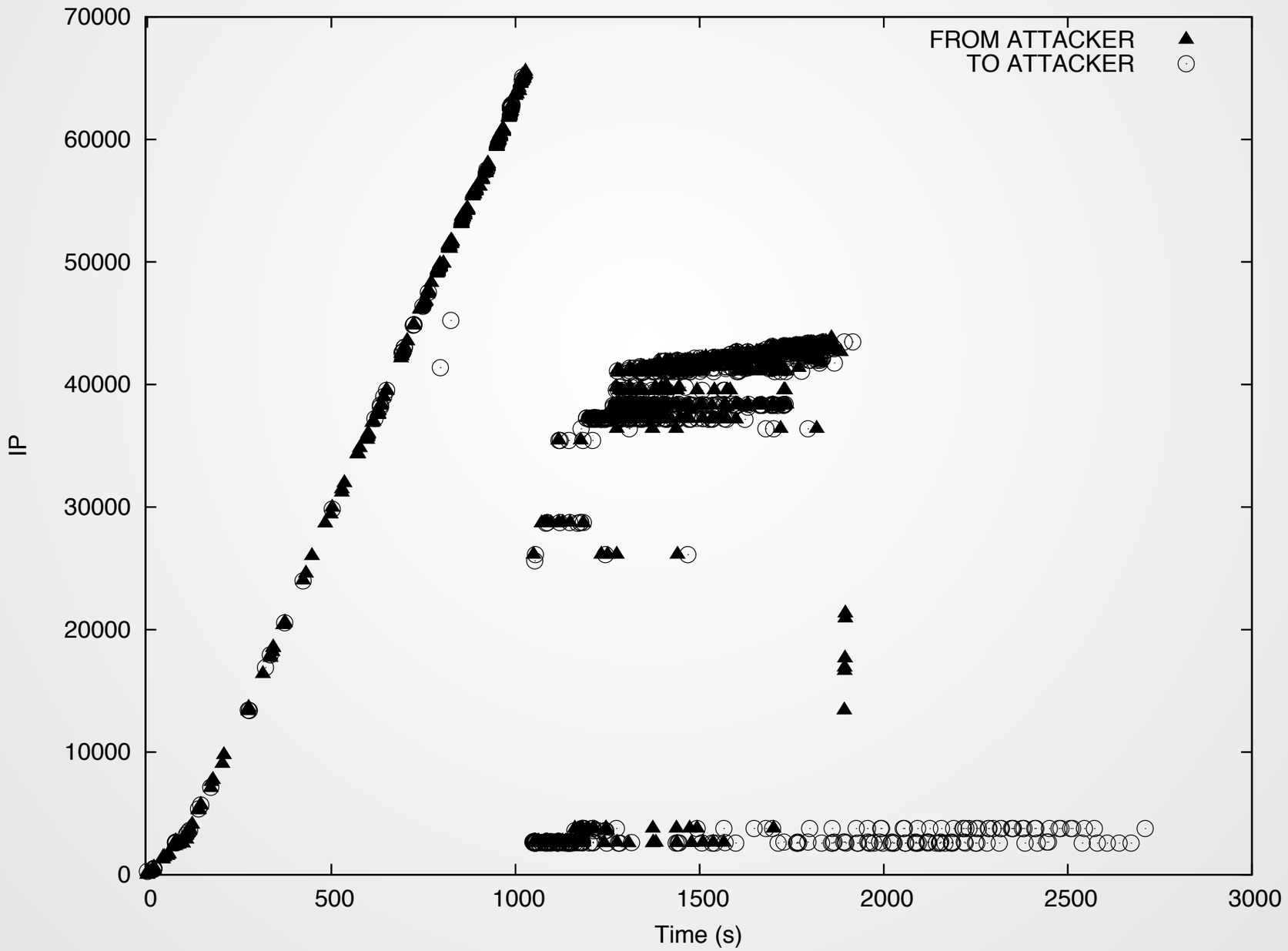
UNIVERSITY OF TWENTE.

REDSOCKS
malware threat defender

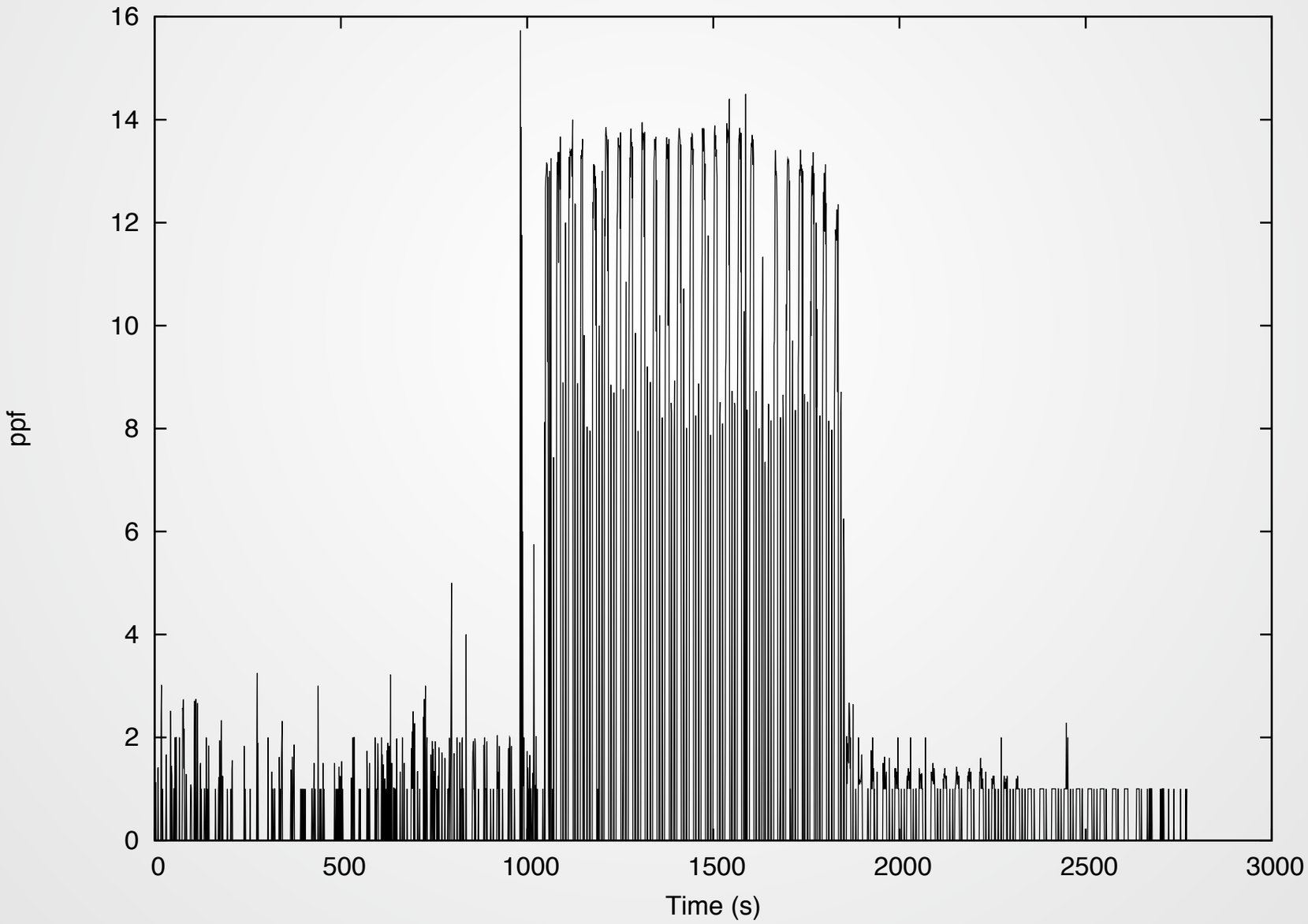
We perform compromise detection.

All flow-based.

SSH attacks


FROM ATTACKER
TO ATTACKER
IP
Time (s)
0
10000
20000
30000
40000
50000
60000
70000
0
500
1000
1500
2000
2500
3000

SSH attacks

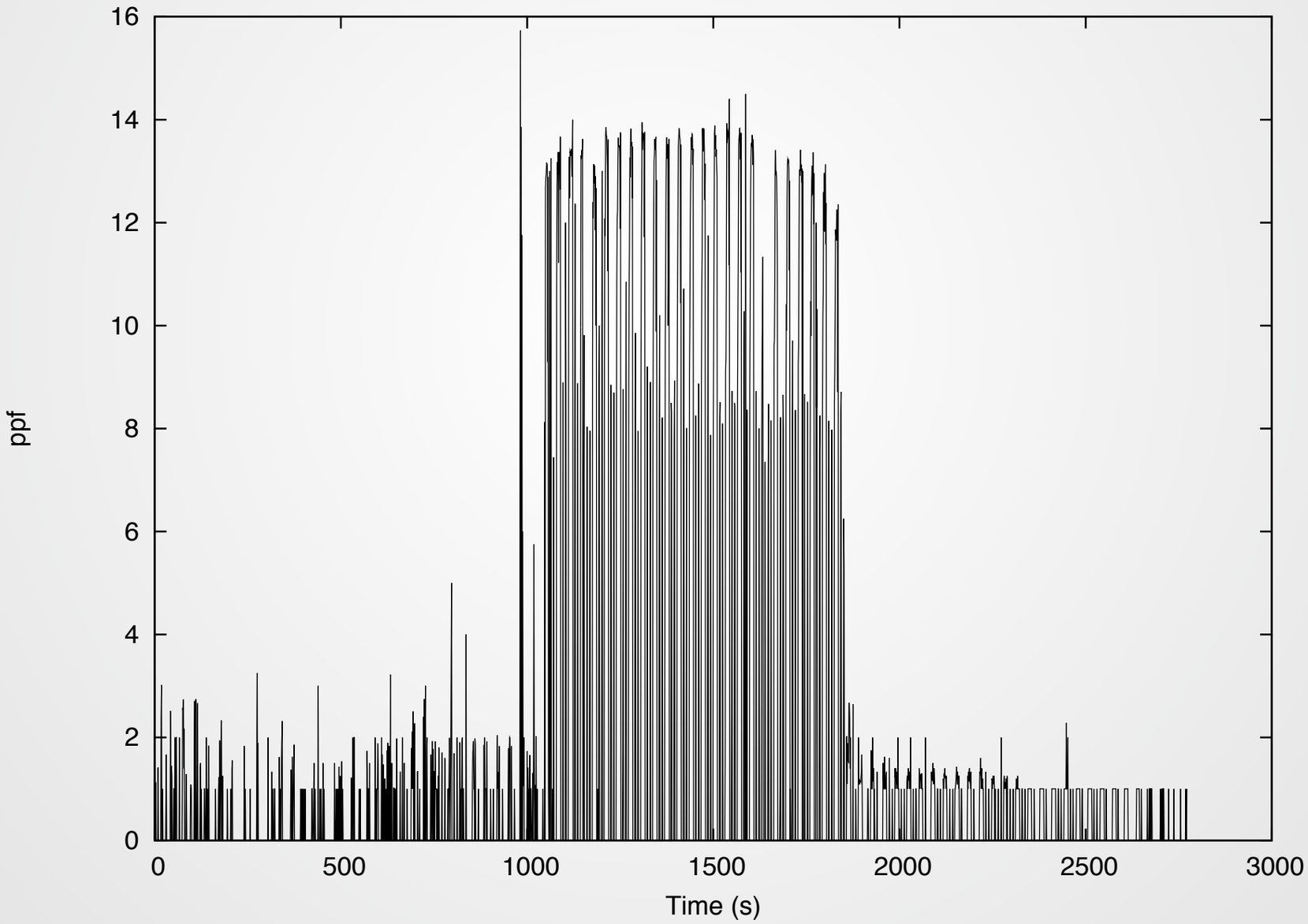


A bit of history…

A bit of history…

- SSHCure 1.0 (June ’12):
  - Purely deviation-based compromise detection
- SSHCure 2.0 (May ’13):
  - Notifications, database maintenance, performance profiling, …

A bit of history…



A bit of history…

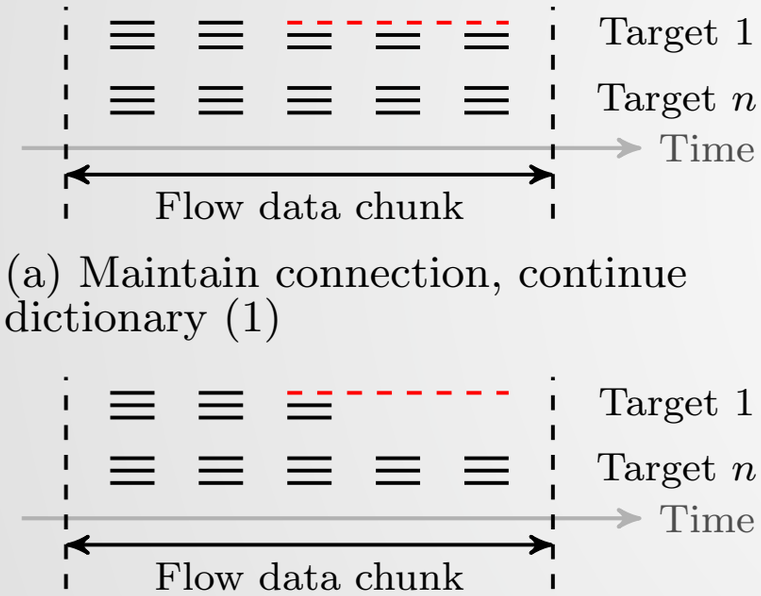
- SSHCure 1.0 (June ’12):
  - Purely deviation-based compromise detection
- SSHCure 2.0 (May ’13):
  - Notifications, database maintenance, performance profiling, …

Recent/upcoming releases

Recent/upcoming releases

- SSHCure 2.4 (July ’14):
  - New compromise detection algorithm (CCR paper release), based on ‘action upon compromise’
- SSHCure 3.0 (January ’14):
  - New frontend, ingress vs. egress attacks

Recent/upcoming releases

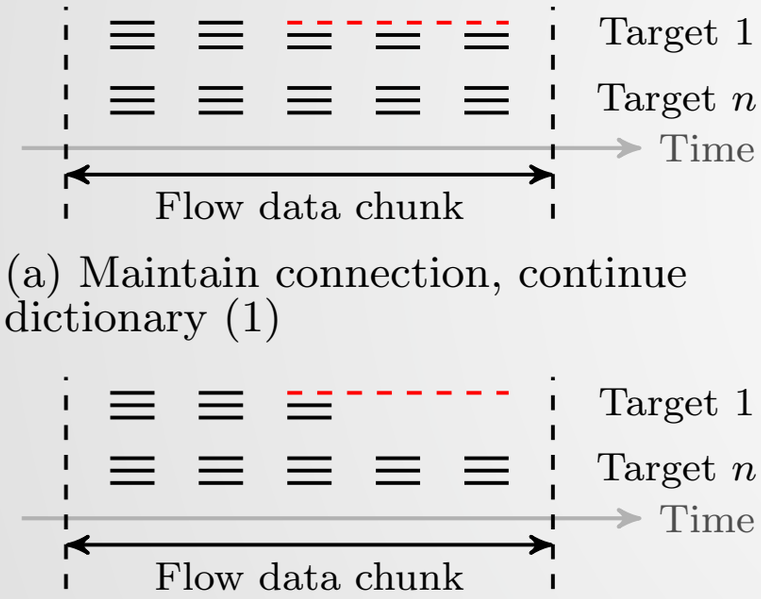


(a) Maintain connection, continue dictionary (1)

(d) Maintain connection, abort dictionary (1)

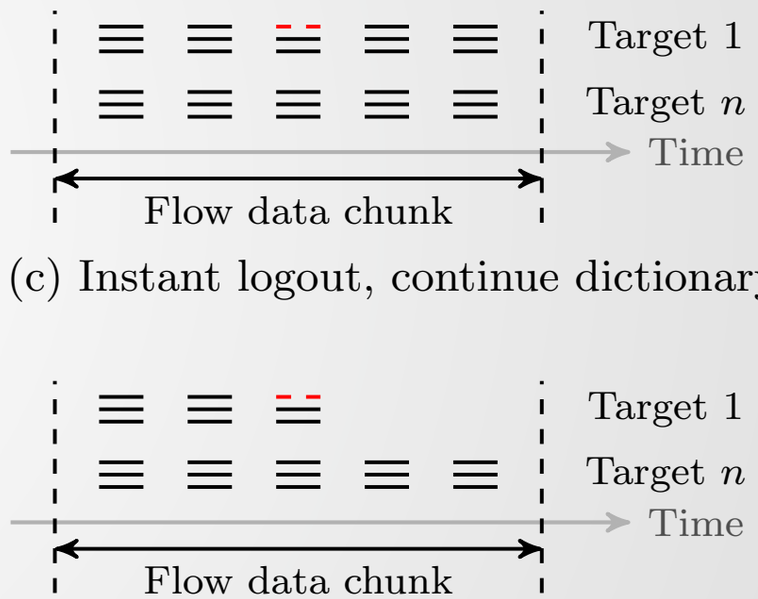
SSH Compromise Detection using NetFlow/IPFIX.
In: ACM SIGCOMM Computer Communication Review, October 2014

Recent/upcoming releases



(a) Maintain connection, continue dictionary (1)

(d) Maintain connection, abort dictionary (1)

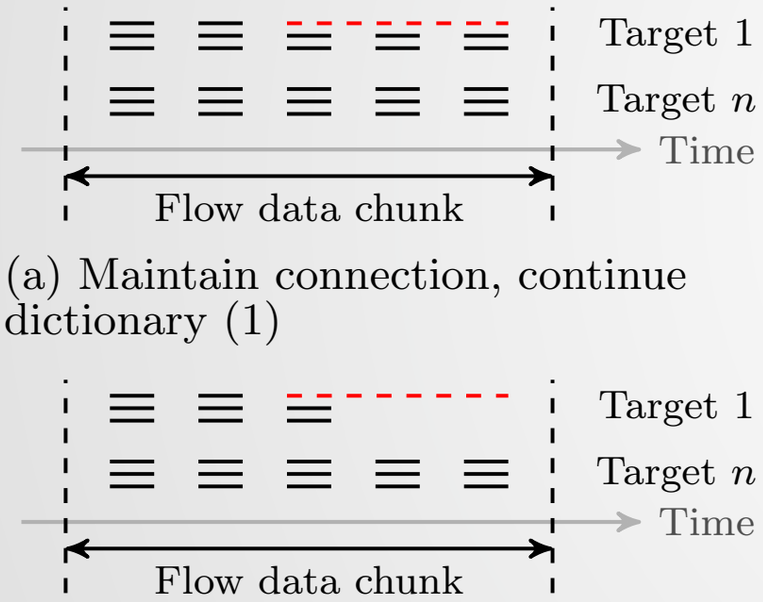


(c) Instant logout, continue dictionary

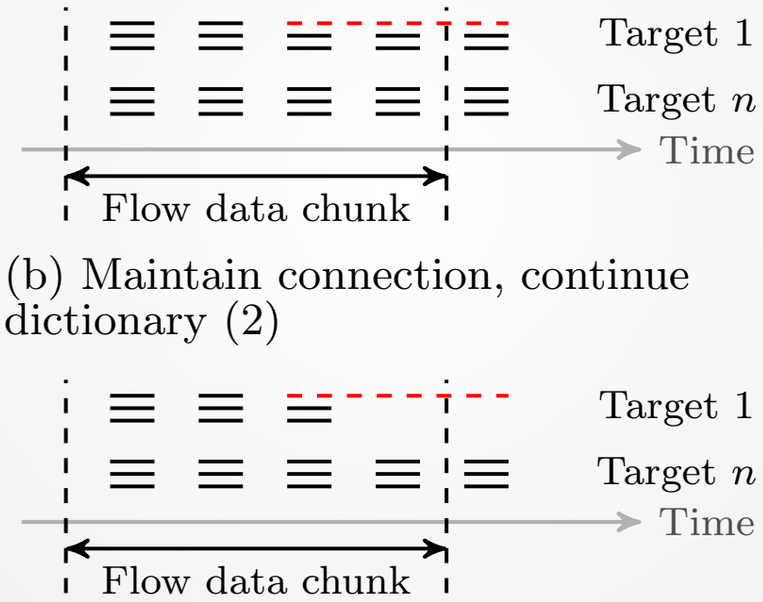
(f) Instant logout, abort dictionary

SSH Compromise Detection using NetFlow/IPFIX.
In: ACM SIGCOMM Computer Communication Review, October 2014

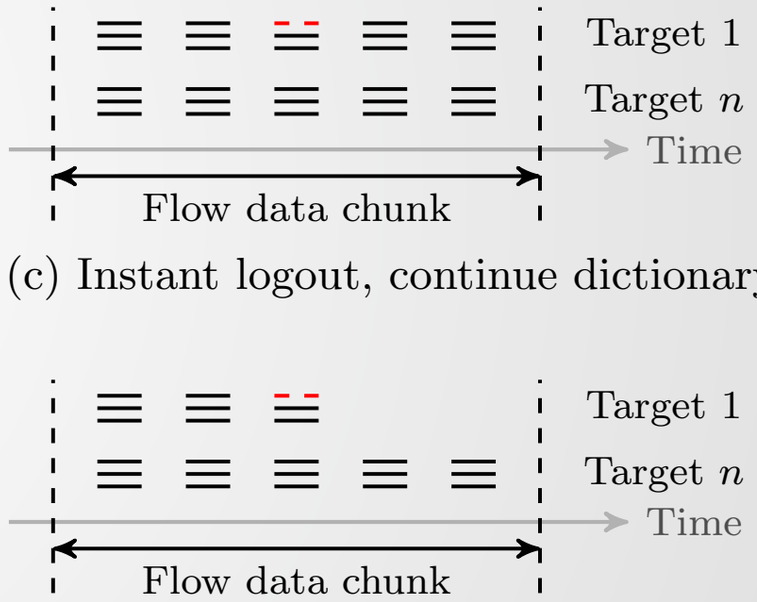
Recent/upcoming releases



(a) Maintain connection, continue dictionary (1)



(b) Maintain connection, continue dictionary (2)



(c) Instant logout, continue dictionary

(d) Maintain connection, abort dictionary (1)

(e) Maintain connection, abort dictionary (2)

(f) Instant logout, abort dictionary

SSH Compromise Detection using NetFlow/IPFIX.
In: ACM SIGCOMM Computer Communication Review, October 2014

Recent/upcoming releases

- SSHCure 2.4 (July ’14):
  - New compromise detection algorithm (CCR paper release), based on ‘action upon compromise’
- SSHCure 3.0 (January ’14):
  - New frontend, ingress vs. egress attacks

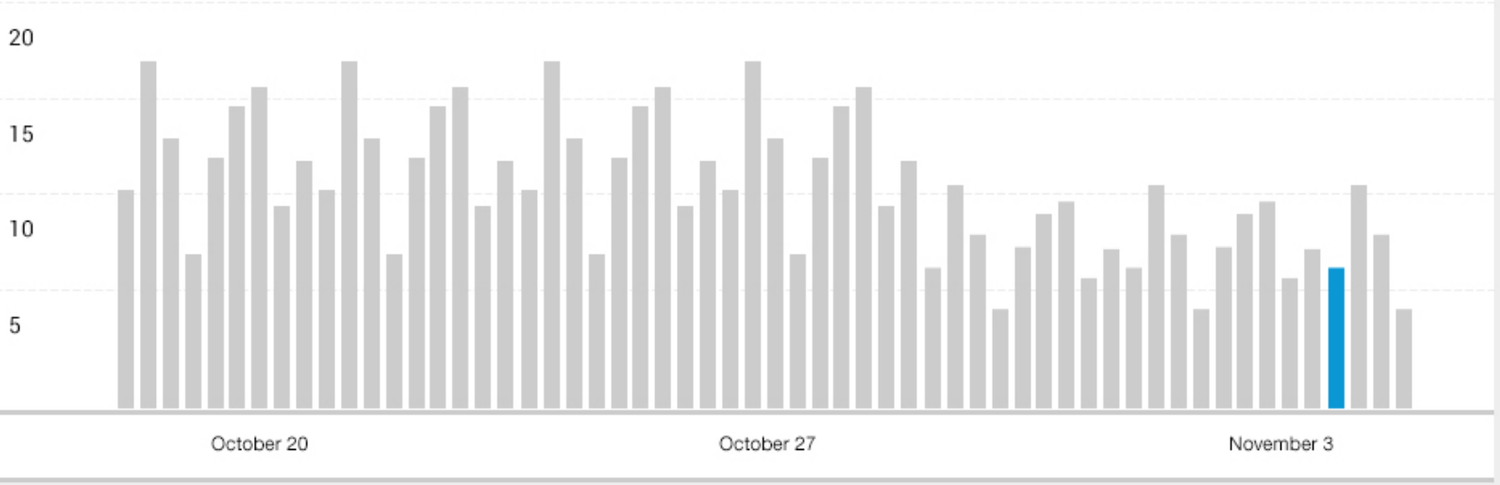
SSHCURE a flow-based intrusion detection system for NfSen. UNIVERSITY OF TWENTE.

- Dashboard
- Incoming
- Outgoing
- Hosts
- Search
- Status
- Help
- Settings

Incoming attacks

Visits — Scan — Brute force — Compromise

Day | Week | Month



Incoming attacks

Phases	Active	Attacker	Date	Targets
	⚡	123.123.123.123	Mon. Jun 30, 2014 19:57	12
		123.123.123.123	Mon. Jun 30, 2014 19:57	456
		130.89.148.136	Mon. Jun 30, 2014 19:57	32
	⚡	123.123.123.123	Mon. Jun 30, 2014 19:57	7455
		123.123.123.123	Mon. Jun 30, 2014 19:57	64

Top targets - Compromise

Target	Attacks	Compromise
123.123.123.123	12	2
123.123.123.123	456	3
130.89.148.136	32	5
123.123.123.123	7455	64
123.123.123.123	64	78

Outgoing attacks

Phases	Active	Attacker	Date	Targets
	⚡	123.123.123.123	Mon. Jun 30, 2014 19:57	12
		123.123.123.123	Mon. Jun 30, 2014 19:57	456
		130.89.148.136	Mon. Jun 30, 2014 19:57	32
	⚡	123.123.123.123	Mon. Jun 30, 2014 19:57	7455
		123.123.123.123	Mon. Jun 30, 2014 19:57	64

Top targets - Brute Force

Target	Attacks	Compromise
123.123.123.123	12	2
123.123.123.123	456	3
130.89.148.136	32	5
123.123.123.123	7455	64
123.123.123.123	64	78

SSHCure

Validation approach

- Ground truth: `sshd` logs from 93 honeypots, servers and workstations, divided over two datasets:
  - Dataset 1 — easy targets
  - Dataset 2 — more difficult targets

	Honeypots	**Servers**	**Workstations**	**Attacks**
Dataset 1	13	0	0	636
Dataset 2	0	76	4	10353

SSHCure

Validation results

- Evaluation metrics:
  - TP / FP — correct / false identification of incident
  - TN / FN — correct / false identification of non-incident
- Detection accuracy close to 100%

	TPR	**TNR**	**FPR**	**FNR**	***Acc***
Dataset 1	0,692	0,921	0,079	0,308	0,839
Dataset 2	—	0,997	0,003	—	0,997

SSHCure

Deployment

- SSHCure is open-source and actively developed
  - Download counter SourceForge (Dec. ’14): 3k
  - Recently moved to GitHub (summer ’14)
- Tested in several nation-wide backbone networks
- Many successful deployments already:
  - Web hosting companies
  - Campus networks
  - National Research and Education Networks (NRENs)
  - Governmental CSIRTs/CERTs

Lessons learned

Lessons learned

Lessons learned

- Ease-of-use is key

Lessons learned

- Ease-of-use is key
  - Many potential SSHCure users (e.g., CSIRTs) are less-skilled than we are

Lessons learned

- Ease-of-use is key
  - Many potential SSHCure users (e.g., CSIRTs) are less-skilled than we are
- Installation scripts are important

Lessons learned

- Ease-of-use is key
  - Many potential SSHCure users (e.g., CSIRTs) are less-skilled than we are
  - Installation scripts are important
- Use of NfSen:

Lessons learned

- Ease-of-use is key
  - Many potential SSHCure users (e.g., CSIRTs) are less-skilled than we are
  - Installation scripts are important
- Use of NfSen:
  - Widely used in (European) NREN community

Lessons learned

- Ease-of-use is key
  - Many potential SSHCure users (e.g., CSIRTs) are less-skilled than we are
  - Installation scripts are important
- Use of NfSen:
  - Widely used in (European) NREN community
  - Experience with SURFmap [1]

[1] http://surfmap.sf.net/

Lessons learned

Lessons learned

- Ingress vs. egress attacks

Lessons learned

- Ingress vs. egress attacks
- Initial focus mainly on ingress attacks

Lessons learned

- Ingress vs. egress attacks
- Initial focus mainly on ingress attacks
- CSIRTs are becoming more responsible *towards* the Internet: Keep it clean!

Lessons learned

Lessons learned

- Integration into workflow is important

Lessons learned

- Integration into workflow is important
  - Yet another tool is hard to integrate into CSIRT workflow

Lessons learned

- Integration into workflow is important
- Yet another tool is hard to integrate into CSIRT workflow
- Integration with existing systems is necessary: IODEF, X-ARF, QuarantaineNet, …

Lessons learned

Lessons learned

- Advertizing is important

Lessons learned

- Advertizing is important
  - People don’t spot your cool project by themselves

Lessons learned

- Advertizing is important
  - People don't spot your cool project by themselves
- Visit meetings & conferences (FloCon, TERENA TNC, RIPE, etc.)

Lessons learned

- Advertizing is important
  - People don't spot your cool project by themselves
  - Visit meetings & conferences (FloCon, TERENA TNC, RIPE, etc.)
  - GitHub vs. SourceForge

Lessons learned

Lessons learned

- 1:1 sampling is hardly used by non-academia

Lessons learned

- 1:1 sampling is hardly used by non-academia
- Problem for our algorithms

Lessons learned

- 1:1 sampling is hardly used by non-academia
- Problem for our algorithms
- Admins are ‘afraid’ of increasing sampling rates

Lessons learned

Lessons learned

- Input data quality is hard to predict

Lessons learned

- Input data quality is hard to predict
  - Algorithms should be as resilient to various data sources as possible

Lessons learned

- Input data quality is hard to predict
  - Algorithms should be as resilient to various data sources as possible
  - Examples:

Lessons learned

- Input data quality is hard to predict
  - Algorithms should be as resilient to various data sources as possible
  - Examples:
    - Availability of TCP flags

Lessons learned

- Input data quality is hard to predict
  - Algorithms should be as resilient to various data sources as possible
  - Examples:
    - Availability of TCP flags
    - Assumptions on flow cache entry expiration

Thanks!

https://nl.linkedin.com/in/rhofstede/

www http://rickhofstede.nl

@ r.j.hofstede@utwente.nl,
rick.hofstede@redsocks.nl

http://nl.linkedin.com/in/luukhendriks

www https://luukhendriks.eu

@ luuk.hendriks@utwente.nl

Questions?

https://github.com/sshcure/sshcure