MUNI

Unraveling Network-based Pivoting Maneuvers: Empirical Insights and Challenges

14th EAI International Conference on Digital Forensics & Cyber Crime

Martin Husák[1,3] **(husakm@ics.muni.cz)**
Shanchieh Jay Yang[2], **Joseph Khoury**[3,4], **Đorđe Klisura**[3,4], **Elias Bou-Harb**[3,4]

[1] Institute of Computer Science, Masaryk University, Czech Republic
[2] The Cyber Center for Security and Analytics, The University of Texas at San Antonio, USA
[3] Department of Computer Engineering, Rochester Institute of Technology, USA
[4] Division of Computer Science and Engineering, Louisiana State University, USA

December 1, 2023

Presenter's Biography

RDNr. Martin Husák, Ph.D.

- Researcher at Institute of Computer Science, Masaryk University, Czech Republic
- Member of Masaryk University's incident response team CSIRT-MU (https://csirt.muni.cz/)
- Currently a visiting researcher at The Cyber Center for Security and Analytics, The University of Texas at San Antonio, USA.
- This research was supported by OP JAK "MSCAfellow5_MUNI" (No. CZ.02.01.01/00/22_010/0003229).

Introduction

Motivation

- Lateral movement has become a major research topic in network security
- *Pivoting, island hopping, stepping stone attack, command propagation, …*
- Pivoting is no longer an advanced attack technique reserved for APTs but is more and more adopted by malware!

Specific Problems

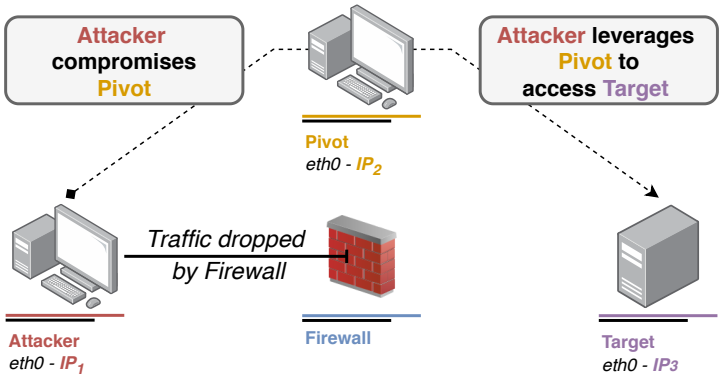
1. Lack of network-based detection methods
   - Existing approaches are mostly host-based – low network coverage
   - A typical pivot is not a well-secured server, but a forgotten IoT or unpatched desktop
2. Existing approaches are evaluated on dataset with not enough background traffic
   - We know very little about possible false positives!
   - Numerous benign pivoting and pivoting-like patterns in the network traffic

Introduction

Contributions

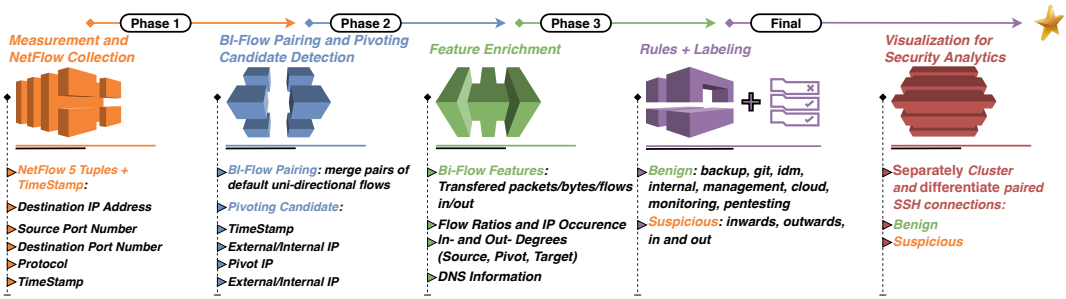
1. We employ a detection algorithm to detect pivoting and pivoting-like events in the campus network, focusing on SSH protocol
2. We empirically analyze the measurement results, identify true and false positives, and investigate the benignity or maliciousness of the detected events
3. We perform PCA and clustering to identify the most meaningful features to design a (semi-)automated pivoting detection tool not relying on local knowledge

Pivoting



An illustrative depiction of a pivoting maneuver through SSH

Pivoting Candidate Detection: Pipeline



- Pivoting detection pipeline – from NetFlow measurement to visualization
- Two-phase detection – detecting *candidates* first, then reasoning about them
- *Candidate* conforms to the signature, but can be malicious, benign, or false positive

Pivoting Candidate Detection

Experiment setup

- Measurements took place in the campus network of Masaryk University
- 36,000+ users, 15,000+ active network devices in /16 IPv4 range
- Precise NetFlow monitoring using Flowmon probes at multiple locations, no sampling, 30 s active time-out
- 10 days of measurement, most of the actors are well known

Limitations

- SSH network traffic only (filtered as *dst.port = 22*)
- RDP and Telnet traffic is heavily regulated – negligible amount of samples
- Other protocols are rare or used only by certain malware (e.g., printing protocols)
- Protocol-agnostic algorithm would explode in complexity

Pivoting Candidate Detection: Algorithm

```
1: f ← list of flows on the input
2: ϵ ← 30
3: len ← size of f
4: for i in [0, len] do
5:     for j in [i+1, len] do
6:         if f_i.dstIP == f_2.srcIP then
7:             if f_1.ts < f_2.ts < f_1.ts + ϵ then
8:                 candidates ← (f_i, f_j)
9:             end if
10:        end if
11:    end for
12: end for
```

- Algorithm inspired by the work of Apruzzese et al., IEEE TETC, 2017

Measurement Artifacts	Min	Max	Total
Biflows	*3,416,328*	*6,412,670*	*39,399,832*
Candidates	*17,026*	*75,116*	*313,193*
Unique Sources (S)	*297*	*646*	*3,410*
Unique Pivots (P)	*64*	*112*	*238*
Unique Targets (T)	*76*	*227*	*468*
Unique Triplets (S, P, T)	*695*	*6,956*	*22,655*
Pivoting Graph Components	*12*	*21*	*14*

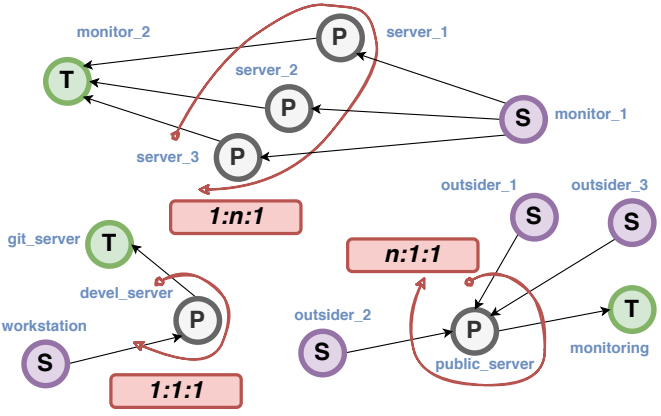
- Pivoting candidate detection, results of 10 day measurement
- *Graph components* to be explained later

Manual Pivoting Candidate Analysis: Pivoting Graph

Pivoting graph

- Visual aid for manual analysis
- Construction via algorithm:

```
1: G ← new empty directed graph
2: for each candidate do
3:     for N in S, P, T do
4:         if N not in G then:
5:             insert node X
6:         end if
7:         if (S,P) not in G then:
8:             insert edge (S,P)
9:         end if
10:        if (P,T) not in G then:
11:            insert edge (P,T)
12:        end if
13:    end for
14: end for
```

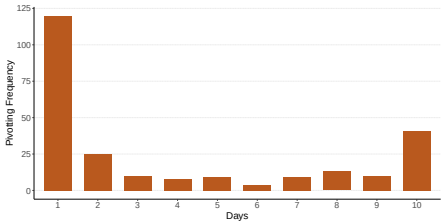


- An excerpt displaying three common (FP) patterns:
- *1:n:1* - monitoring by the tools like Icinga or Nagios
- *1:1:1* - command propagation, often seen with git
- *n:1:1* - frequently scanned SSH server initiates connection

Manual Pivoting Candidate Analysis: Results

Class	Rule	Candidates
Benign and False Positives	Monitoring	*288,161*
	(Anonymized Services)	*15,761*
	Git & Backup	*5,404*
	Management & Cloud	*1,288*
	Pentesting	*1,627*
Unclassified and Suspicious	Internal	*29*
	Inwards	*338*
	Outwards	*19*
	In and Out	*566*
Total	-	*313,193*

Rule-based annotation of pivoting candidates



Temporal analysis of pivot presence

Manual Pivoting Candidate Analysis: Discussion

Pivoting candidates

- Large number of candidates detected
- Candidate detection algorithm is fast and simple, even with large data
- Candidates most often appear only once or regularly

Candidate classification

- Vast majority of candidates is benign or FP
- Automated tools stand behind most of the FPs
  - They can be clearly identified by checking domain names (e.g., *nagios*, *.github.com*)
- No outright malicious activity was observed, although many are suspicious, such as:
  - legitimate users working from home via SSH pivot (instead of VPN)
  - unusual communication between short-lived hosts in different clouds
- Pivoting graph is a highly useful visual aid

Towards Automated Candidate Filtering

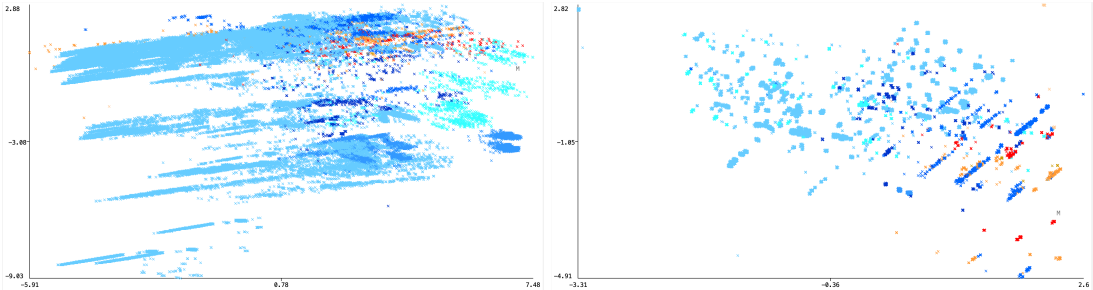
Can we automate the pivoting candidate classification?

- If yes, which features are the most important?
- Principle Component Analysis (PCA) and Clustering

Feature set – 39 in total

- 18 numerical NetFlow-based features
  - Duration and transferred packet and bytes in both connections
  - Biflows distinguish directions
  - Ratios of features between S-P and P-T connection
- 21 contextual categorical contextual features
  - 3 locations of actors (*external, public, private*)
  - 7 location combinations (e.g., Source and Target)
  - 4 in- and out-degrees of the pivoting graph
  - 7 indicators if the combination was seen the day before

Towards Automated Candidate Filtering



- Clustering analysis: The left figure shows clustering with all features, the right figure shows clustering with contextual features only
- Colors are assigned as follows: blue for benign and false positive candidates, orange for in-and-out and outwards scenarios, red for inwards scenarios
- These figures are the most compelling, but still not sufficient to cluster the candidates effectively

Discussion

Limitations

- This work focused on SSH traffic only, other protocols would require similar analysis
- We do not reflect the situation, in which the attacker uses two different IP addresses on a pivot (e.g., public and private)
- Lack of ground truth and significant imbalance of the data

Security implications

- Potential attacker would be detected using the proposed method
- The defender needs to process large amounts of alerts or automate the procedures
- Attacker with good knowledge of local environment may hide the activity

Conclusion

Summary

- SotA pivoting detection algorithm was deployed in campus network for 10 days
- No clear attacks were found, but in-depth analysis of FPs was conducted
- Proper classification of results heavily depends on contextual features

Recommendations for future work

- Signature detection is not enough, classification of the results is needed
- Checking local environment creating a whitelist or a list of filtering rules is advised
- Locations are interesting features, more fine-grained *zones* could be useful, too
- There is a need to reduce the number of results to approx. less than 10 per day, so that they can be investigated manually

MASARYK
UNIVERSITY