

A novel flow-based statistical pattern recognition architecture to detect and classify pivot attacks

Item Type	Thesis or dissertation
Authors	Marques, Rafael Salema
Citation	Marques, R.S. (2022) A novel flow-based statistical pattern recognition architecture to detect and classify pivot attacks. Wolverhampton: University of Wolverhampton. http://hdl.handle.net/2436/625109
Publisher	University of Wolverhampton
Rights	Attribution-NonCommercial-NoDerivatives 4.0 International
Download date	2026-05-16 02:19:21
License	http://creativecommons.org/licenses/by-nc-nd/4.0/
Link to Item	http://hdl.handle.net/2436/625109

A NOVEL FLOW-BASED STATISTICAL PATTERN
RECOGNITION ARCHITECTURE TO DETECT AND
CLASSIFY PIVOT ATTACKS

RAFAEL SALEMA MARQUES

A thesis submitted in partial fulfilment of the requirements of the
University of Wolverhampton for the degree of Doctor of Philosophy

2022

This work or any part thereof has not previously been presented in any form to the University or to any other body whether for the purposes of assessment, publication or for any other purpose (unless otherwise indicated). Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

The right of Rafael Salema Marques to be identified as author of this work is asserted in accordance with ss.77 and 78 of the Copyright, Designs and Patents Act 1988. At this date copyright is owned by the author.

Signature:

Date:

ABSTRACT

Pivot attack or pivoting is a well-known technique used by threat actors to cover their tracks and overcome connectivity restrictions imposed by the network defences or topology. Therefore, detecting ongoing pivot attacks while the opponent has not yet achieved their goals is essential for a solid defence strategy. However, recognising and classifying this technique in large corporate networks is a complex task. The literature presents limited studies regarding pivot attacks, and mitigation strategies have severe constraints to date. For example, related work still focuses on specific protocol restrictions techniques scoped at internal network assets only. This approach is inefficient since opponents commonly create pivot tunnels across the internet.

This thesis introduces and evaluates APIVADS, a novel flow-based detection scheme to identify compromised assets supporting pivot attacks. Moreover, APIVADS outperforms previous approaches regarding features and capacities. To the best of our knowledge, this is the first protocol and cryptographic primitives agnostic, privacy-preserving approach capable of detecting pivot attacks over the internet. For example, Its efficient data reduction technique can achieve near real-time detection accuracy of 99.37% by distinguishing ongoing pivot attacks from regular enterprise traffic such as TLS, HTTPS, DNS and P2P over the internet. Additionally, this thesis proposes

APCA, an automatic pivot attack classifier algorithm based on perceived indicators of attack (IoA) generated by APIVADS, to determine the level of connectivity achieved by the adversary. APCA can distinguish between different types of pivoting and contribute to the threat intelligence capabilities regarding the adversary modus operandi. The architecture composed by APIVADS and APCA considers a hybrid approach between decentralised pivoting host-based detection and a centralised approach to aggregate results and achieve scalability. Empirical results from our experiments show that even when the adversary uses evasive pivoting techniques, the proposed architecture is efficient and feasible regarding classification and detection, achieving high accuracy of 98.54% and low false positives.

ACKNOWLEDGEMENTS

Countless people have supported me in writing this thesis, and without their patience and support, this task would not have been possible. First and foremost, I thank my supervisors, Dr Haider Al-Khateeb and Gregory Epiphaniou. They read my numerous revisions and helped me sort out my ideas and steer them in the right direction. I thank them for their patience, guidance and support throughout my studies.

I recognise the opportunity and thank the Brazilian Air Force for providing the necessary funding to complete this project. Moreover, I thank my lovely wife Juliana for the sacrifices she made for me to pursue my PhD far from home. To my daughters Julia and Marina, who moved abroad and endured this long process with us, always supporting and loving us. I thank God every day for the opportunity to be their father.

CONTENTS

Abstract	ii
Acknowledgements	v
List of Tables	xi
List of Figures	xiii
1 Introduction	1
1.1 Rationale	3
1.2 Problem Statement	4
1.3 Aim, Research Questions and Objectives	5
1.3.1 Aim	5
1.3.2 Research Questions	5
1.3.3 Objectives	6
1.4 Scope	6
1.5 Constraints	8
1.6 Contributions	9
1.7 Thesis overview	11
2 Literature review	13
2.1 Advanced Persistent Threat (APT)	13
2.2 APT threat models	15
2.3 Privacy-preserving traffic analysis	22
2.4 Flow-based traffic analysis	25

2.5	Traffic profiling and clustering	28
2.6	Compromise Detection	29
2.7	Indicators of Compromise (IoC)	30
2.8	Indicators of Attack (IoA)	30
2.9	Cyber Threat Intelligence frameworks (CTI)	31
2.10	Pivoting	32
2.11	Summary	36
3	Methodology	37
3.1	Instruments	37
3.2	Dataset creation	38
3.3	Evaluation metrics	39
3.4	Experiments	40
3.4.1	Parameter optimisation tests	41
3.4.2	Virtual network experiments scenario	41
3.4.3	Real network experiments scenario	43
3.4.4	Evasive pivot techniques detection	44
3.4.5	APT attack stages inference	45
3.5	Summary	46
4	APIVADS: Adaptive Pivoting Detection Scheme	47
4.1	Pivot attack	47
4.2	Privacy-preserving characteristics and description	48
4.3	Detection scheme and flow-based pattern recognition model	49
4.4	APIVADS data processing phases and threat model overview	53

4.5	APIVADS modules interaction	54
4.6	Data collection module	58
4.7	Data extraction module	59
4.8	Detection filter module	60
4.9	Agent interaction module	64
4.10	Summary	67
5	Pivot attack classification	69
5.1	Classification criteria	69
5.2	Semantic Network Models (SNM)	73
5.2.1	Single interface semantic model	75
5.2.2	Dual interface semantic model	76
5.3	Automatic Pivot Classifier Algorithm (APCA)	77
5.4	APCA Advantages and drawbacks	82
5.5	Offensive and defensive pivot metrics	83
5.6	Summary	86
6	Results and discussion	89
6.1	Virtual network experiment results	89
6.2	Real network experiment results	99
6.3	APT attack stages inference results	103
6.4	Evasive pivot techniques experiments results	106
6.5	Algorithms complexity	108
6.6	Summary	110
7	Conclusions and Future Work	113

Bibliography	116
A Dataset examples (snapshots)	147

LIST OF TABLES

2.1	Kill chain stages from attacker’s perspective [1]	15
2.2	Attackers actions and related challenges in camouflage sub-phases [2]	18
2.3	APT attack stages	21
2.4	Pivot detection approaches comparison	34
3.1	Evaluation metrics	40
3.2	Attack stage inference parameters in function of P_{piv}	45
4.1	Detection scheme algorithms’ parameters	56
4.2	APIVADS flow attributes structure	59
4.3	Pivot Attack Alert Messages sample	65
5.1	Pivot attack classes	72
5.2	APIVADS Alert messages sample [3]	78
5.3	Class V pivot attack alert messages scenario	80
6.1	Detailed experiments result in function of T_w	92
6.2	Detailed experiments result in the function of PPS	96
6.3	Real networks experiments detailed results	100
6.4	BitTorrent protocol experiment detection metric rates	101
6.5	Comparison with other detection algorithms	102

6.6	Intentional propagation delays experiment results	108
6.7	Algorithms usability comparison regarding processing time . .	109

LIST OF FIGURES

2.1	Cyber kill chain 6 stages attack progression [4]	15
2.2	DOTMUG Threat Model [5]	17
2.3	Circular Cyber kill chain model [6]	19
2.4	Proposed revised cyber kill chain model [1]	20
2.5	Simple pivot scenario	32
3.1	Virtual network experiment diagram	42
3.2	Pivot attack experiment conducted over the internet	44
4.1	Pivot attack scenario	48
4.2	Pivot and not pivot traffic patterns comparison	51
4.3	APIVADS threat model diagram	57
4.4	Pivot tunnel representation of Table 4.3 alert messages sample	66
5.1	Pivot attack classification scenarios	73
5.2	Single interface semantic network model	75
5.3	Dual interface semantic network model	76
5.4	Class V pivot scenario diagram	79
5.5	Pivoting offensive and defensive capability metrics example	83
6.1	Experiments result in the function of T_w	92
6.2	P_{piv} influence in detection results	95

6.3	Detection of pivot attacks with T_w equal to 30 seconds	97
6.4	Detection of pivot attacks over the internet with T_w equal to 60 seconds	98
6.5	Change of behaviour detection based on pivot traffic variation	105
A.1	APIVADS PAAM generated events	147
A.2	Pivot attack in the Command & Control attack stage pcap files dataset	148
A.3	Biflows statistical attributes	149

CHAPTER 1

INTRODUCTION

The internet's complex infrastructure supports several critical communications. Governments, industry, academic and financial institutions heavily rely on the rapid exchange of information and facilities over this network. However, it is also a high-value target and fertile environment for attackers to steal confidential data and intellectual property.

The cybersecurity threat landscape is continuously evolving. The number of targeted attacks conducted by systematic and skilled operators referred to as Advanced Persistent Threats (APT) have increased significantly in the last couple of years [7]. According to recent threat intelligence reports [8, 9], APT funded by nation-states typically conduct intelligence operations expressing military power in the cyber domain [10] and a new type of APT is emerging: independent mercenary groups. Those adversaries share similar tactics, techniques and procedures (TTP) with state-sponsored APT groups by offering hacking services or acting as information brokers.

Leaking sensitive information to financial competitors is highly lucrative, attracting talented hackers driving the cybercrime APT ecosystem. The constant change within the modus operandi of APT groups to remain undetected

is forcing the evolution of defence mechanisms to develop new techniques to timely identify intruders and prevent damage. APT groups are incredibly efficient in achieving their objectives. However, the necessary interaction with the target network to accomplish their goals creates evidence of presence within the target network [11], generating indicators of compromise (IoC) and indicators of attack (IoA).

Identifying an attack in its early stages is essential for an effective defence strategy. Moreover, It is critical to identify compromised network resources to neutralise the opponent. Organisations use Cyber Threat Intelligence (CTI) to identify, understand, predict and adapt to malicious actor behaviour to reduce threat detection time [12].

A technique widely used by APT to extend access to the targeted network and permeate various stages of the attack, from initial exploitation to data exfiltration, is the pivot attack or pivoting [13, 14]. The attacker uses pivot tunnels to bypass route restrictions between assets of interest or bypass firewalls to achieve connectivity to the final target.

Over the years, various approaches to APT detection have been explored. However, few studies address the pivot technique, despite being a technique widely used by APT actors to expand their access to the target network. I selected the paradigm of flow-based network monitoring combined with statistical techniques to achieve lightweight pivot attack detection near real-time. The detection architecture has demonstrated precise results in identifying pivot nodes, providing high detection accuracy rates and minimal false

positives.

1.1 Rationale

According to the authors in [15], identifying pivot tunnels in near real-time is a complex and challenging research problem due to the following facts: It is immune to signature-based approaches and computationally expensive given the massive amount of traffic processed by modern networks. The cited authors conducted a recent study suggesting that it is possible to obtain good pivot detection results using the flow-based monitoring paradigm. However, Husak et al. [16] state that this approach is not suitable for detecting pivot attacks in real scenarios, as most pivot attacks originate on the internet, and the detection scheme is restricted to internal networks connections only. The last cited authors also proposed a detection approach considering internal and external connections. The approach taken by [16] solved the constraint regarding complex interconnected networks. However, it is still incomplete as it cannot cope with P2P protocols like BitTorrent, does not provide good detection metrics and disregards well-known variations of pivot attacks.

A variety of approaches to APT detection methods and models were proposed. In recent years, researchers have typically characterised the APT attack life cycle into stages [17] composed of TTPs, tools and methods used by APT actors to conduct offensive actions systematically. Although, pivot attacks permeate various APT attack stages and are a commonly used technique by APT actors to expand their access to the target network.

There is a literature gap in detecting and classifying pivot attacks. The capability to identify pivot tunnels and classify them with a criterion that provides the necessary granularity is primarily unexplored and relevant as a defensive resource. It can contribute to identifying compromised assets, inferring APT attack stages and providing relevant information to support cyber threat attribution.

1.2 Problem Statement

Pivot attack detection is crucial because it is a widely used technique employed in various APT attack stages. Effective detection of pivot attacks is essential for developing a solid defence strategy. Identifying ongoing APT attacks in their initial stages is helpful when the opponent has not yet achieved its goals. Additionally, the literature presents limited studies regarding pivot attacks, since the detection and classification strategies to date have severe constraints.

To our knowledge, the pivot attack detection and classification problem have not been addressed with consistent results. For example, current work has restrictions on specific protocols for internal network assets only. Therefore, the current pivot attack detection strategies are inefficient since opponents commonly create pivot tunnels across the internet with multiple hops to hide the attacker's origin and reduce the attribution probability.

1.3 Aim, Research Questions and Objectives

1.3.1 Aim

This research aims to improve state-of-the-art pivot attack detection and classification. The focus was on providing a scalable and lightweight approach to detect and classify pivot attacks in high-speed networks in near real-time. Additionally, the detection should be effective regardless of the network topology and the type of traffic, as well as the transport and application layers. Finally, the solution for effectively detecting pivot attacks should have high accuracy and a low number of false positives.

1.3.2 Research Questions

In light of the previous considerations, the main objectives and contributions of this thesis are to answer the following research questions (RQ):

RQ1: Is the flow-based approach efficient in detecting compromised devices supporting pivot attacks in simple and complex interconnected networks under high traffic conditions?

RQ2: How to infer the current APT attack stage based on pivot tunnel traffic information using the flow-based network monitoring paradigm?

RQ3: How viable is a pivot attack classification based on perceived IoA to identify the degree of connectivity achieved by the adversary?

1.3.3 Objectives

Specific objectives (O) are identified to address the research problem as follows:

- O1:** To model and evaluate network-level IoA event to correlate traffic patterns with pivot attacks.
- O2:** To research and develop a flow-based detection scheme which can identify compromised devices supporting pivot attacks in near real-time and infer APT attack stages regardless of the network topology.
- O3:** To investigate and model a pivot attack classification criterion to infer the degree of connectivity achieved by the adversary based on perceived IoA.
- O4:** To perform a critical evaluation of the proposed architecture with other approaches existing in the literature regarding pivot attacks.
- O5:** To create a dataset of pivot attacks traffic TTP related to different APT attack stages.

1.4 Scope

Unlike other pivot attack detection approaches such as [15], APIVADS can identify pivot tunnels in simple and complex interconnected networks with high traffic.

APIVADS offers a novel privacy-preserving detection scheme to detect ongoing pivot attacks in near real-time. Therefore, a distributed agent-based approach for detection and a centralised strategy for aggregating and classifying the results with APCA was used to achieve scalability and cybersecurity situational awareness.

The architecture is composed of a pivot attack detection scheme (APIVADS) enriched with useful information regarding pivot attack classification provided by an automatic pivot attack classifier algorithm (APCA). The APCA uses as input the IoA produced by APIVADS. Therefore, the classification mechanism depends on APIVADS detection.

The host-based agents are restricted to the local device, limiting the agents' perception strictly to local host traffic. However, this natural limitation is desirable to ensure privacy when processing sensitive data. Likewise, this restriction means that a single APIVADS agent cannot ensure the cybersecurity of an enterprise network concerning pivot attacks. Nevertheless, suppose that a third-party CTI framework aggregates the pivot attack events generated by the APIVADS agents. In that case, it is possible to achieve scalability due to distributed detection and a holistic view of ongoing pivot attacks within the monitored devices.

The pivot attack detection capability is not affected by traffic that contains an encrypted payload. The detection algorithm uses packet header attributes as input data to generate flows and derives new attributes. This detection strategy has proven efficient because a pivot node's primary func-

tion is to forward traffic between two endpoints creating an indirect connection between the attacker and the target. Therefore, it is not practical to transfer data using the TCP/IP protocol and consequently create a pivot tunnel without access to the packet header attribute information used by APIVADS presented in Chapter 4. Furthermore, the architecture detection and classification mechanisms do not require any previous training or knowledge repository.

1.5 Constraints

APIVADS can infer pivot tunnels when dealing with anonymous traffic, such as the onion routing circuit used by TOR [18]. However, identifying the actual origin of anonymous traffic is beyond the scope of this work. When processing traffic forwarded by anonymous routing techniques, APIVADS identifies the endpoint forwarding the anonymous traffic (relay node) and supporting the pivot attack rather than the original IP that generated the traffic.

The input data used by APIVADS is the current traffic perceived in the device, which is continuously collected and processed by the algorithms. Therefore, it is assumed that the adversary is not subverting the traffic perception. Finally, the agents must process the pivot attack incoming and outgoing traffic biflows to achieve detection and classification.

1.6 Contributions

The main contributions of this thesis are:

1. The development of a novel flow-based detection scheme to identify compromised assets supporting pivot attacks in near real-time, using an agnostic approach concerning protocols and cryptographic primitives. To the best of our knowledge, this is the first pivoting detection scheme with no network topology constraints. Hence, it can detect pivot attacks originating from the internet and targets within an enterprise network.
2. An efficient data reduction technique to discard biflows that are not involved in pivot attack activities.
3. A dataset of pivot attacks traffic TTP related to different APT attack stages to provide reproducibility and accurate comparison with future works.
4. Based on the pivot detection algorithm parameters, we model typical pivot traffic pattern as a function of volume and frequency capable to infer APT attack stages.
5. The development of network-level IoA event to feed CTI frameworks, correlate traffic patterns with pivot attacks and infer pivot attacks of any length.

1.6. CONTRIBUTIONS

6. Provided a pivot attack classification criterion based on the perceived IoA events that can be used to infer the connectivity achieved by the attacker within the target network.

Most of the results and contributions of the thesis were published in the following journals:

- Rafael Salema Marques, Gregory Epiphaniou, Haider Al-Khateeb, Carsten Maple, Mohammad Hammoudeh, Paulo André Lima De Castro, Ali Dehghantanha, and Kim Kwang Raymond Choo. A flow-based multi-agent data exfiltration detection architecture for ultra-low latency networks. *ACM Transactions on Internet Technology*, 21(4), jul 2021., DOI: 10.1145/3419103.
- Rafael Salema Marques, Haider Al-Khateeb, Gregory Epiphaniou, and Carsten Maple. APIVADS: A novel privacy-preserving pivot attack detection scheme based on statistical pattern recognition. *IEEE transactions on information forensics and security*, 17:700–715, 2022., DOI: 10.1109/TIFS.2022.3146076.
- Rafael Salema Marques, Haider Al-Khateeb, Gregory Epiphaniou and Carsten Maple. Pivot attack classification for cyber threat intelligence. *Journal of Information Security and Cybercrimes Research*, (Accepted).

The paper “A flow-based multi-agent data exfiltration detection architecture for ultra-low latency networks” was presented at the Faculty of Science

and Engineering Festival of Research 2021.

1.7 Thesis overview

The remainder of the thesis is structured into chapters as follows:

Chapter 2 provides an overview of the background knowledge required to conduct this research. In addition, it explores previous research findings and state-of-the-art topics of interest.

Chapter 3 presents the methodology used to design the architecture and explains in detail the choices made to solve the research problem. This chapter permeates several objectives defining the dataset that will generate input for APIVADS algorithms addressing *Objective 5*, evaluation metrics, optimisation tests and experiments scenarios. Therefore, partially addressing *Objectives 2 and 4*.

Chapter 4 addresses *Objective 2* presenting APIVADS modules, filters, algorithms and the data processing solution to detect compromised devices supporting pivot tunnels in simple and complex networks. In order to address the *Objective 1*, this chapter introduces the Pivot Attack Alert Message (PAAM), which corresponds to an IoA event generated by APIVADS.

Chapter 5 provides details on the classification criterion of the pivot attacks proposed in this work used by APCA to fully address *Objective 3* and partially addressing *Objective 1* providing details regarding PAAM processing to infer pivot classes.

Chapter 6 presents the achieved results and compares the proposed solu-

1.7. THESIS OVERVIEW

tion with other approaches found in the literature addressing *Objective 4*.

Finally, Chapter 7 concludes the results of the thesis and provides suggestions for future research.

CHAPTER 2

LITERATURE REVIEW

This chapter provides an overview of important concepts and previous strategies concerning pivoting and flow-based approaches to detect compromised devices performing malicious activities. First, the Advanced Persistent Threat (APT) attack is defined, and central APT attack models are presented. Afterwards, traffic analysis approaches are presented and followed by the concept of compromise detection, indicators of compromise and attack. Finally, Cyber Threat Intelligence Frameworks (CTI) are introduced.

2.1 Advanced Persistent Threat (APT)

The United States Air Force (USAF) formalised the nomenclature for this type of threat to allow military teams to discuss the characteristics of the attack with civilian partners, as information about the attacks had not yet been released to the public. This concept emerged in the context of cyber defence after Google publicly announced that it had been the victim of a sophisticated and personalised attack that likely originated in China. However, the attack was not unique to Google. More than 30 strategic North American companies associated with the technology and defence industries were

2.1. ADVANCED PERSISTENT THREAT (APT)

also attacked, resulting in significant damage due to the loss of intellectual property. After the public announcement, the term was described in a US patent published in 2008 [19], defining APT as an offensive state-sponsored cyber actor characterised by greater sophistication, stealth and skill.

According to [20], by the end of 2010, the term APT was not restricted to state-sponsored operations. It was also used to refer to sophisticated TTP in long-term hacking campaigns conducted by well-resourced adversaries to meet the demands of powerful entities [21].

APT attacks are persistent, targeted at a specific organisation, systematic, related to sporadic events [22, 23] and performed in several steps [24, 25]. Additionally, tend to present a long dwell-time [26, 27], which in the context of APT attacks, is a metric that corresponds to the time that an APT actor stays undetected within a target network. Therefore, long dwell-time gives the attackers significant time to go through the attack cycle, propagate, and accomplish tasks with success [28].

The multi-stage characteristic increases the complexity of defence strategy against this type of attack. Experience shows that APT groups are incredibly efficient in achieving their objectives [29, 30, 31]. Several large-scale security breaches are attributed to APT actors [32, 33, 34, 35, 36] and according to recent studies, it is relevant to point out that prominent nation-state threat actors and powerful entities are likely to continue their efforts in the following years [37]. After all, the main goal of an APT attack typically involves cyber espionage and sabotage [38].

2.2 APT threat models

The literature contains a wide variety of APT attack models. The Cyber Kill Chain [39] developed by Lockheed Martin is a well-known model adopted by the National Institute of Standards and Technology (NIST) and referenced by the industry and the academic community. Initially, this model contemplates 6 sequential stages, from reconnaissance to Exfiltration [4] as illustrated in Figure 2.1.

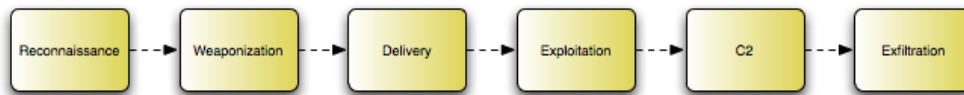


Figure 2.1: Cyber kill chain 6 stages attack progression [4]

However, in a subsequent study, this model was updated to 7 kill chain stages [40] as presented in Table 2.1.

Table 2.1: Kill chain stages from attacker’s perspective [1]

Stages	Content
Stage 1 (Reconnaissance)	Collect information such as e-mail addresses and conference information.
Stage 2 (Weaponization)	Combine exploits and backdoors to insert payload.
Stage 3 (Delivery)	Transfer the weaponized file to the victim system through an e-mail, web, or USB.
Stage 4 (Exploitation)	Use vulnerabilities in order to run the code in the victim system.
Stage 5 (Installation)	Install a malicious program in the attack target’s asset.
Stage 6 (Command & Control)	Open a channel to remotely control the victim system and command accordingly.
Stage 7 (Action on Objectives)	When access that is equivalent to handling the actual keyboard becomes possible, the invader has achieved their purpose.

2.2. APT THREAT MODELS

This famous model was modified by different researches with some variations and differences [41, 42, 43, 44].

Another important kill chain model is the Mandiant Attack Life cycle [45]. This model illustrates the evolution of the attacker's internal network activities, considering recursive internal reconnaissance and lateral movement. However, it still requires much interpretation when assigning indicators to action groups, leading to inconsistent data analysis and less efficiency in security personnel workflows [46].

The [40] model and all the other that present similarities regarding to its traditional perimeter-focused approach and malware dependency has been criticised in the last few years [47]. The threat landscape evolved in APT modus operandi, and the criticism is well-founded, as attacks can originate from internal adversaries without using a single piece of malware. Since the original model was published in 2011, modifications have been proposed by academic authors and cybersecurity professionals over the years.

Bryant and Saiedian [46] proposed modifications to conventional kill chain models to improve data aggregation and correlation, resulting in more detailed alarms to security analysts.

Pham et al. [31] introduced a quantitative framework to model the APT threat based on APT actors' reasoning about incentives and deterrents when evaluating the nodes to compromise and persist within a network. This model considers the balance regarding the value of a target against the possibility of detection.

Charan et al. [5] proposed DOTMUG, a threat model to address the misusing of legitimate services. Figure 2.2 illustrates the threat model scenario that uses Google Teachable Machine for malicious purposes to establish a foothold, lateral movement, and data exfiltration phases of the APT life cycle. The authors conducted five experiments to validate the threat model and discussed an emerging TTP concerning APT actors named Living Off the Land Binaries (LoLBins) [48, 49]. This technique abuses legitimate binaries part of the operating system to perform malicious tasks without leaving any artefacts behind.

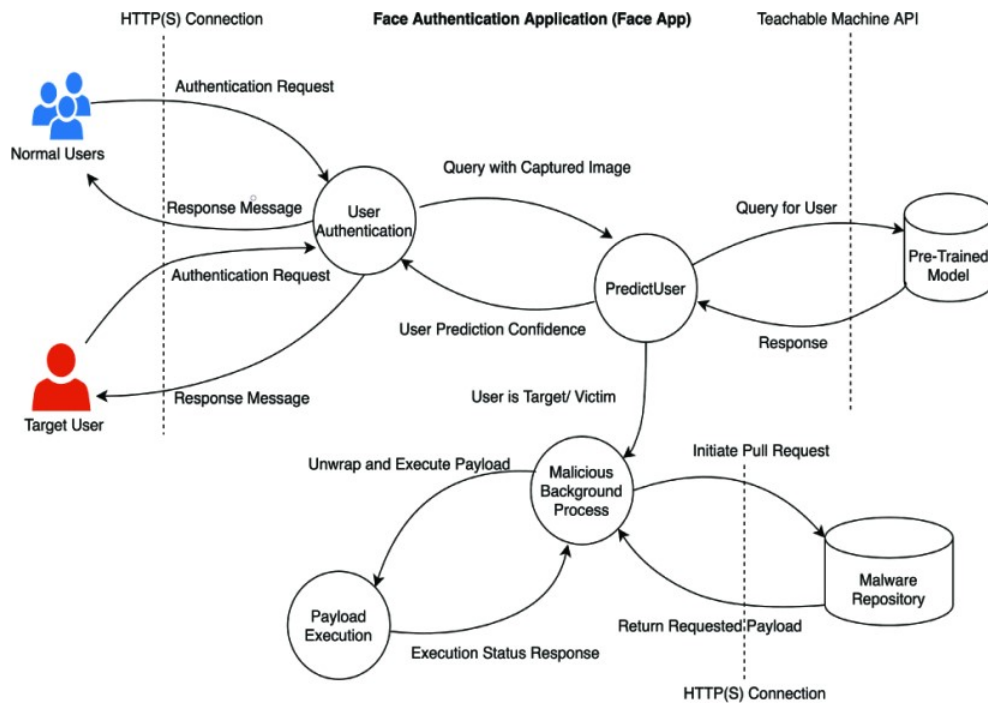


Figure 2.2: DOTMUG Threat Model [5]

Messaoud et al. [2] presented an APT life cycle focused on attackers’

2.2. APT THREAT MODELS

objectives rather than their actions pointing out six APT phases. The authors explicitly append the “camouflage” term to most APT life cycle model phases. In each phase, camouflage refers to the activities taken by the attacker to be undetected during the attack progress as reproduced in Table 2.2. Additionally, the authors point out the challenges and opportunities regarding the camouflage.

Table 2.2: Attackers actions and related challenges in camouflage sub-phases [2]

APT Phase	Attackers actions	Challenges
Phase 1: Reconnaissance & Camouflage	Collecting information outside organisation perimeter. Using social engineering with hidden identity.	Control of information exposed outside the organisation. Raising awareness of the employees.
Phase 2: Gaining access & Camouflage	Exploiting vulnerabilities permitting access without generating logs. Using trusted domain or most frequently site web in infection campaign.	Review and control security levels of trusted domains. use of behavioural anomaly detection techniques.
Phase 3: Lateral movement & Camouflage	Hiding traffic in normal and trusted organisation traffic. Using custom encryption and applicative hidden tunnels to exchange with the attacker.	Change and integrity monitoring. Privileged accounts monitoring. Network traffic analyses. Phase 4: Gathering information & Camouflage
Using custom encryption and applicative hidden tunnels to gather information	Asset and user behaviour monitoring. Phase 5: Actions on Objective & Camouflage	Encrypting and defragmenting transferred files. Using applicative hidden tunnels to transfer data.
Data volume and contents monitoring. Asset and user behaviour monitoring. Phase 6: Cleaning	Covering tracks without generating traces.	Change and integrity monitoring. Logs outsourcing.

Marchetti et al. [50] proposed a framework for categorising internal hosts that are probably involved in APT activities by observing high volumes of network traffic. It combines heuristics, behavioural and statistical models capable of identifying suspicious activities on large networks. The approach

compares individual hosts statistics based on their past behaviour and against other hosts of the organization.

The authors in [6] introduced a non linear model differently from the majority of the APT life cycles. The circular kill chain illustrated in Figure 2.3 indicates a repetitive pattern.

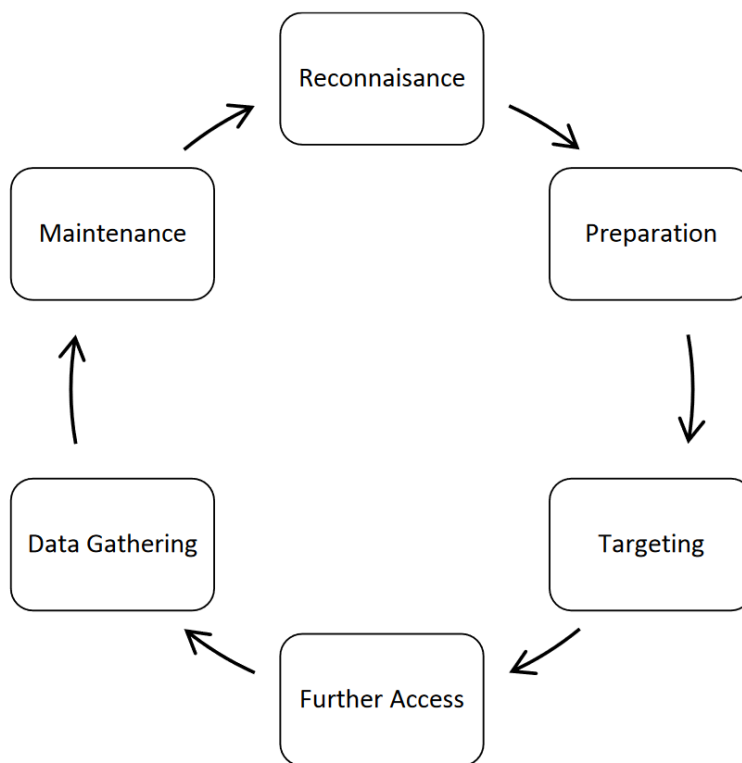


Figure 2.3: Circular Cyber kill chain model [6]

However, the threats that occur outside the organization and the cycle of threats from inside have different threat attributes, and the circular model cannot explain these differences.

This model presents logical advantages since APT actors tend to repeat

2.2. APT THREAT MODELS

actions to maintain their presence within the target network. Eventually, the cybersecurity analysts and the automatic set of network defence tools will detect and mitigate the attacker’s TTP, forcing a new attack cycle. However, the threats that arise outside the organization and the internal threats have different attributes, and the circular model cannot explain these differences. Therefore, [1] presented a model that intent adequately to express each stage of internal and external threats that must be addressed differently, as illustrated in Figure 2.4.

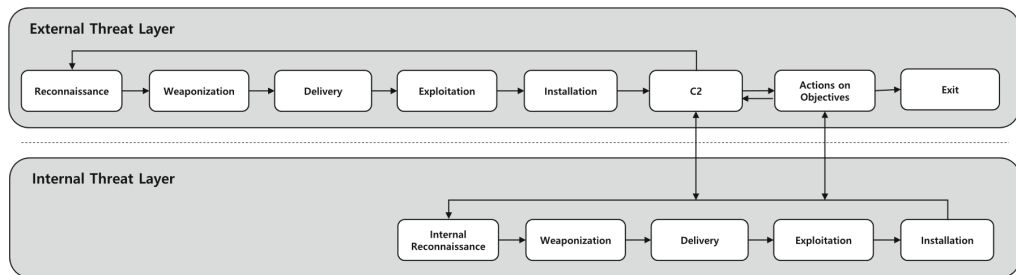


Figure 2.4: Proposed revised cyber kill chain model [1]

Milajerdi et al. [51] presented HOLMES, a detection system that aims to produce a signal indicating the malicious coordinated activities of an APT campaign. However, due to the need to go unnoticed in their actions, the APT’s modus operandi changes over time. This change in attacker behaviour can deviate from the presented models leading to a lack of cybersecurity awareness.

Ferazza [52] compared the three most well-known and widely utilised models, Lockheed Martin’s Cyber Kill Chain [39], MITRE’s ATT&CK frame-

work [53], and the Diamond Model [54], based on the dissimilarities between their objectives, designs, and levels of abstraction.

Some of the most successful detection approaches seek to identify negative patterns by monitoring environmental changes [55, 56] to infer a specific attack condition. For instance, an attacker must generate outbound traffic when using the internet to exfiltrate data.

Alminshid and Omar [57] summarise several APT attack models and propose a model that combines the typical attack stages generally encountered in APT attacks, as shown in Table 2.3:

Table 2.3: APT attack stages

Attack stages	Description
Intelligence gathering	Identification of defence solutions and potential targets using intelligence disciplines (e.g. OSINT, (see [58]) and, based on the information gathered, planning a customised attack to infiltrate the target network.
Initial exploitation	The objective of this phase is to achieve initial access to the target network (foothold). This is typically achieved using social engineering techniques to induce the user to compromise the computer system inadvertently. Given that email is a widely used resource in companies for dealing with people outside the organisation, this is a widespread modus operandi to deliver the initial vector (e.g., an email with a malicious document attached).
Command and Control	Assuming the attacker does not have physical access to the target's premises, it is necessary to establish a Command and Control (C&C) channel to send and receive commands to the malware inside the target network [59]. Once the C&C channel is persistent, the attacker will discover and exploit other vulnerable network assets, expand access laterally within the network, harvest credentials and exfiltrate data.
Privilege escalation	The privilege escalation goal is to achieve administration privileges within the network [60], meaning manipulating files, accessing restricted resources or installing unwanted programs such as malware. After obtaining system administrator privileges, the attacker can subvert the operating system to effectively cover their tracks and remain unnoticed for as long as possible.
Data exfiltration	Unauthorised transmission of sensitive information from the victim's network to external locations under the attacker's control [61, 62, 63]. However, normally, exfiltration activities conducted by APT groups involve moving information out of the victim network in small amounts over a long period of time. [64, 65]

In conclusion, despite the differences between the models, APT attacks share some similarities in terms of TTP and attack phases. Furthermore, identifying attack actions in near real-time and deriving attack phases is essential for developing solid cyber awareness in corporate networks.

2.3 Privacy-preserving traffic analysis

Network traffic metadata has value to attackers because it contains sensitive information. Likewise, third-party vendors should not access unencrypted traffic due to data privacy concerns, including compliance with data protection laws. Packets payload inspection can lead to privacy problems and requires expensive hardware for storage and processing. Furthermore, Deep Packet Inspection (DPI) approaches [66, 67, 68, 69] are criticised when applied in fast enterprise networks because they cannot work with end-to-end encryption [70, 71] which is essential in the context of modern secure communication protocols. However, a recent trend partially mitigates the drawbacks by adopting a privacy-preserving DPI approach. The authors in [72, 73, 74] propose a cloud-based provider that supports middlebox outsourcing packet inspection while preserving the client’s confidentiality when sharing information. Traffic and detection rules sent to third-party middleboxes are typically encrypted due to privacy [75, 76, 77]. However, all the privacy-preserving DPI models cited use the signature-based paradigm and consequently suffer from its issues and limitations, which are well documented in the literature. A signature or rule-based detection scheme attempts to identify attacks by

comparing incoming events with their stored signatures [78, 79]. A *signature* is a description that represents a known attack based on some characteristics. Some works focused on solving the manual setting of signatures when the DPI tools identify new protocols [80, 81]. In order to comply with privacy requirements, detection is achieved by comparing encrypted payloads with an existing encrypted signature. Suppose the adversary employs the Polymorphic Blending Technique (PBT) to protect the traffic [82] or actively uses evolving threat techniques to alter the traffic [83]. In this case, we face a scenario where a signature-based detection strategy fails to detect the malicious traffic. The authors in [74] addressed privacy issues related to DPI techniques for outsourced middleboxes. Their proposal prevents a new rule set in the system from being linked to previous inspection results. The authors stated that the strategy slightly increases resilience and makes adaptive attacks less effective. However, the signature-based method is limited to a knowledge repository, which is unsuitable for detecting unknown attacks.

Despite the significant improvement in detection rates achieved by adaptive signature-based schemes, this type of solution still presents difficulties in identifying advanced techniques such as PBT [82]. The PBT uses polymorphic data obfuscation techniques to bypass Signature-based Intrusion Detection Systems (SIDS) and blending to evade an Anomaly-based Intrusion Detection System (AIDS). According to the authors in [82], AIDS can detect simple polymorphic attacks because their byte frequency differs from that in legitimate traffic. Therefore, PBT collects raw packets to create a

2.3. PRIVACY-PRESERVING TRAFFIC ANALYSIS

traffic profile and adjusts the payload byte frequency to bypass the AIDS detection mechanism by impersonating legitimate traffic with the expected byte frequency. Additionally, PBT uses a byte substitution technique to obfuscate data which can be considered polymorphic as it changes with each communication according to the traffic profile expected by AIDS.

SIDS uses some detection approaches to defeat simple polymorphic attacks [84]. However, when polymorphic attacks are combined with different techniques such as PBT, the chance of evasion increases due to the difficulties in modelling complex systems. Additionally, cyber adversaries are constantly evolving their techniques and exploiting the knowledge of machine learning detection algorithms to evade defences [85]. Moreover, experience shows that it is only a matter of time before attackers adapt their TTP to new defence strategies.

According to [86], the concept of privacy applies to scenarios in which third-party entities process sensitive information. Therefore, host-based approaches are privacy-preserving by default, as they do not share sensitive information. The flow-based analysis paradigm does not inspect the packet payload, which is good from a privacy point of view. It has no restrictions regarding end-to-end encryption or proprietary malware ciphered traffic. Unlike DPI approaches, its mechanism extracts packet header attributes to create flows used as input to algorithms without requiring computational effort for inspecting payloads and storage resources.

2.4 Flow-based traffic analysis

Various studies have been conducted to create cyber awareness regarding attackers' actions in computer networks using the flow-based [87, 88, 89] and conversation-based approach [90]. This relatively new research field is gaining importance among researchers due to its advantages over the traditional Deep Packet Inspection (DPI) approach for identifying malicious traffic, because this type of analysis uses a scalable method to reduce the traffic volume into flows [91]. The main drawbacks associated with DPI compared to the flow-based approach are its reduced network speed, and its impossibility of being used in an encrypted communication scenario where the cypher is secure and unknown [88, 92]. On the other hand, since the flow-based approach does not inspect the packet payload, it has limitations regarding the amount and variety of information extracted from the observed traffic. Our literature review focuses on peer-reviewed papers presenting novelties of the flow-based approach to provide cybersecurity awareness in recent years. We chose to follow the trend of the flow-based technique due to the success already achieved in solving internet traffic classification problems [93, 94, 95]. Additionally, it is necessary to adapt the original approach to use the DPI approach in fast enterprise networks. Some authors proposed signature-based solutions using middlebox outsourcing packet inspection [73, 72, 96, 97]. Besides the drawbacks already stated, the signature-based paradigm presents critical limitations regarding polymorphic data obfuscation [82].

Narang et al. [98] introduced the usage of 2-tuple (source IP, destination IP) instead of the traditional flow-based 5-tuple (source IP, source port, destination IP, destination port, protocol) to differentiate legitimate P2P traffic from a malicious one. In this way, they created PeerShark, whose strategy is focused on observing the different communication between peers. PeerShark can categorise P2P traffic with more than 95% accuracy.

According to [99], there are two main approaches to network monitoring: active and passive. Active techniques inject traffic into a network to perform measurements (e.g. ping and traceroute). Passive strategies observe the generated traffic at a measurement point, process it and generate alerts. Flow-based and DPI approaches are the most common passive traffic analysis strategies in the related cybersecurity literature.

The typical flow attribute of unidirectional NetFlow data is presented by [100]. These attributes are extracted from the set of packets that share the same flow [101]. A flow is defined in [102] as “a set of IP packets passing an observation point in the network during a certain time interval, such that all packets belonging to a particular Flow have a set of common properties”. Therefore, a bidirectional flow (biflow or conversation) is composed of packets sent in both directions between two endpoints [103]. Finally, NetFlow-like analysis systems have been used for network monitoring, planning, and accounting [103]. NetFlow is considered a novel approach to be explored in the field of cybersecurity [104, 105, 106].

Authors in [107] proposed an internet traffic classification based on flows’

statistical properties with machine learning techniques. Initially, the authors extract flows from a traffic capture. Then, select relevant statistical properties of these flows to use as input in an unsupervised learning mechanism to group flows by similarities. The authors stated a better result for bidirectional flows than the unidirectional counterparts when applied to the K-means algorithm.

Tayal et al. [108] proposed a scheme that finds recurring and regular time interval traffic. The approach still uses flows to find similarities between multiple instances of flows and infer communication patterns whereas the authors in [109] proposed a behavioural learning flow-based model. The authors developed BASTA (behavioural Analytics System using Timed Automata), which uses probabilistic deterministic real-time automata (PDR-TAs) to detect infected hosts and identify unseen infections in networks.

Wang et al. [110] introduced BotMark, a hybrid botnet detection approach that analyses traffic behaviour using flow and graph-based strategies. Moreover, Botmark is agnostic concerning the botnet C&C protocol and structure, demands no previous knowledge, and can be employed in complex networking environments. However, BotMark discards biflows to legitimate servers (e.g., Google, Youtube). This assumption can be dangerous since threat actors commonly use well-known services to employ C&C servers [111, 112, 113] to hide their activities in plain sight.

2.5 Traffic profiling and clustering

Traffic profiling is essential in modern network contexts to understand user behaviour and support decisions in traffic optimisation and capacity planning [93, 114, 115]. Therefore, it is widely used to identify network traffic patterns in the cybersecurity research field, using clustering methods to derive traffic profiles based on characteristics and behaviours within malicious activities. For example, Priyanka and Dave [116] presented PeerFox, a two-tier detection scheme to identify P2P botnet activities in their waiting stage. The authors considered two basic behaviours to profile traffic and achieve detection: long-lived peers and the intensity of search requests. Furthermore, the authors in [117] proposed an automated network application profiling framework based on Traffic Causality Graphs (TCGs), achieving high accuracy in application identification for P2P traffic even when the program uses random ports and encryption to protect the communication.

Mai and Park [118] combined unsupervised learning to cluster the network traffic in groups with similar features and apply classification in order to identify botnet activities. Additionally, the authors compared some clustering method metrics to identify the best results in detecting botnets.

Vance [119] presented an alternative algorithm derived from flow-based attributes deployed in the detection of APT. Their results indicate that statistical modelling of APT communications can successfully develop deterministic characteristics for detection. Burghouwt et al. [120] proposed a novel

approach to real-time detection of C2 channels based on trust of traffic destinations. In the approach, a destination can become trusted by transitivity, if its origin can be evaluated by another trusted entity. The main contribution of [121] was an algorithm that performs DNS traffic monitoring in large networks based on the extension of standard flows. The authors in [122] presented NEMEA, a modular framework for network traffic analysis at Layer 7 that uses the stream-wise concept, i. e. data is analysed continuously in the memory with minimal data storage required. Finally, Berkay Celik et al. [123] modelled a flow-based framework that uses tamper-resistant features at the transport layer to protect against rootkits.

2.6 Compromise Detection

The main objective of the compromise detection approach is to identify devices subverted by cyber adversaries without permission to support unauthorised activities. Therefore, identifying compromised devices on time is vital to achieving cybersecurity awareness and mitigating attacks as it increases the opponent's capabilities and presence inside the network perimeter.

Most compromise detection schemes address the problem using the paradigm of node misbehaviour detection [124]. Therefore, authors in [99] proposed a compromise detection technique that recognise the attack tool behaviour instead of checking for traffic deviations, which was proven to reduce false positives significantly. Also, Hellemons et al. [125] proposed SSHCure, a flow-based intrusion detection system for SSH attacks that can identify com-

promised hosts using flow data. Finally, enterprise networks receive several different types of attacks constantly. However, only a few attacks result in compromise [126], increasing the complexity of identifying compromised devices.

2.7 Indicators of Compromise (IoC)

Indicators of Compromise (IoC) are observable artefacts produced by the adversary TTP used in an attack and other associated activities [127, 128, 129]. The term is initially used in the digital forensic and incident response area, but it is slowly extending to cybersecurity solutions [130]. In the context of CTI, a compromise indicator can be defined as a piece of information used to identify a potentially compromised system [131]. This information can be a simple IP address to a complex set of tactics, techniques and procedures. Therefore, the primary purpose of the IoC is to provide helpful information to identify a potentially compromised system.

2.8 Indicators of Attack (IoA)

While an IoC is suitable for identifying pieces of evidence left behind when a breach occurred (e.g. presence of malware), indicators of attack (IoA) is a new concept to address instantaneous measurements. An IoA is useful to describe a suspicious activity [132] that can provide real-time visibility regarding signs of an ongoing attack, such as code execution, pivot tunnels, covert channels, and lateral movement within a network [133]. Therefore,

IoA addresses actions and steps that reveal the adversary's intentions and can be used to predict their objectives.

2.9 Cyber Threat Intelligence frameworks (CTI)

Threat intelligence is threat information aggregated, transformed, analysed, interpreted or enriched to provide the necessary context for decision-making processes and information sharing with other organisations [134, 135, 136, 137]. CTI frameworks have gained relevance in recent years due to the capability to aggregate, classify and correlate events [138, 139, 140] and threat information from various sources, prevent attacks, forecast potential threats and reduce the dwell-time [141, 142]. However, with multiple and eclectic sources of information, the semantic meaning and product standardization can lead to misinterpretation of the information. Also, conflicts of interest among distinct organizations can lead to knowledge share barriers [137].

Additionally, CTI frameworks encapsulate intelligence regarding TTPs and events, providing relevant information while hiding unnecessary details from CTI decision-making [143]. Therefore, reducing the cybersecurity analyst workload. Finally, despite the cited drawbacks, CTI frameworks provide a better understanding of the adversary modus operandi [144], contributing to the attribution and detection of malicious actors.

2.10 Pivoting

To expand the control over the target network, APT typically conducts enterprise reconnaissance and lateral movement to identify vulnerable assets of interest, holding sensitive information. A common technique used by APT to overcome connectivity restrictions imposed by firewalls or to access different network segments is the Pivot attack. They provide an attractive capability to adversaries because the initial access typically does not correspond to the actual target [145, 146]. Apruzzese et al. [15] described the first flow-based pivoting detection algorithm, which uses temporal graph-analytics techniques to detect the attacks and prioritise detection results based on a scoring system. The same authors defined the pivot attack as a command propagation tunnel created among three or more internal hosts to control a specific target. According to [147], Lateral movement-based attacks usually happen in the C&C attack phase to gather internal system structure information, achieve persistence and expand control over the target network.

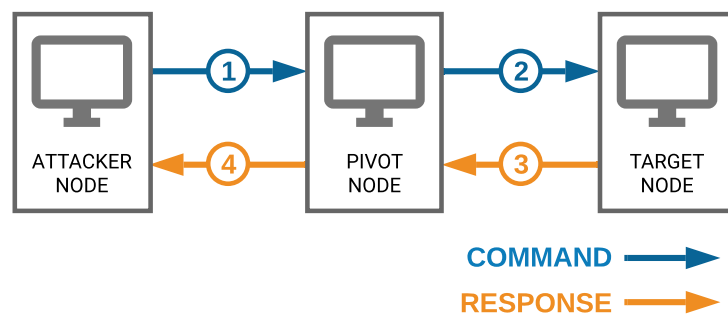


Figure 2.5: Simple pivot scenario

Figure 2.5 illustrates a simple pivot scenario where the attacker cannot directly access the target node due to the absence of a route between networks or lack of connectivity through defence mechanisms like firewalls. However, the pivot node is accessible from the attacker node. It can connect to the target node, being able to forward both outgoing and incoming traffic between the attacker and the target node. From the attacker node, outgoing traffic is represented by numbers 1 and 2 (command), while incoming traffic is represented by numbers 3 and 4 (response).

There are few studies focusing on the development of detection schemes for pivot attacks. However, it is essential that we compare our detection scheme with prior research effort in this area. Table 2.4 presents a comparison among the approaches regarding detection results, capacities, and restrictions based on the authors' statements. Each column represents an algorithm feature, and when present, it is identified with a checkmark.

To the best of our knowledge, [15] is the first paper that specifically addresses pivoting by introducing an attack detection method. However, authors in [16] stated that the approach is not feasible to detect pivot attacks in enterprise networks when considering internal and external connections. Another issue stated by [16] is regarding a high number of FP when their detection strategy is applied to p2p traffic (e.g. BitTorrent) or gaming protocols. Husak et al. [16] evolve the first pivot detection algorithm proposed by [15]. Unlike earlier research, their work combines human expertise with machine learning techniques to address pivot detection when dealing with

Table 2.4: Pivot detection approaches comparison

	Host-based approach	Network-based approach	Distributed processing	Privacy-preserving	Any length pivot tunnel	Application layer agnostic	Transport layer agnostic	Intentional delays resilient	Requires training phase	Near real-time detection	Complex networks
APIVADS	✓		✓	✓	✓	✓	✓	✓		✓	✓
Husak et al. [16]		✓			✓			✓			✓
Bai et al. [148]	✓		✓	✓					✓		
Apruzzese et al. [15]		✓			✓			✓			

internal and external hosts. However, results shown 99.99% of false positives when applied to a real network environment. We understand that some assumptions in [16] are not accurate. Firstly, the authors assume that protocols and destination ports are the same for both pivot candidate biflows. In reality, adversaries can bridge traffic at the transport layer (e.g. UDP to TCP bridge [149]) or using software at the application layer to send commands in a specific protocol and plan to receive the response via different service or port. Therefore, according to our understanding, the detection scheme should be agnostic regarding protocols and ports to address unconventional techniques. Therefore, we understand the approach disregards important pivot attacks scenarios.

When comparing [16] and [15] with APIVADS, the approaches present

similar functionalities and detection results in simple scenarios. However, the algorithm created by [15] is not applicable within complex networks. This is a significant limitation because a real-world adversary commonly uses the internet to conduct malicious activities, and consequently, the attacker node is located outside the enterprise network. The centralised processing adopted by [15] affects the complexity of the algorithm drastically. The pivot length size, which can increase the complexity of the theoretical worst-case scenario of [15] does not affect our approach in the same way due to the distributed processing strategy, where each asset is responsible for identifying and processing part of the problem, merging the result in CTI Frameworks.

Authors in [148] propose a Machine Learning (ML) approach to detect anomalous RDP sessions based on the extraction of features from host event logs and system calls. Although APIVADS and [148] use different data as input, the paper targets lateral movement attacks that can share similar characteristics with pivot attacks in several ways. For instance, an attacker can use the internet to access a remote desktop inside an enterprise network and use it to access other devices. In this case, the RDP host is serving as a Pivot node. Besides the excellent result of DA and TPR outperforming APIVADS, the authors tolerate a higher number of FP in exchange for a lower incidence of FN and this fall in the same problem stated by [16] concerning [15] work already mentioned.

Finally, based on the comparisons provided, APIVADS outperforms other detection approaches with regard to features and capacities. To the best of

our knowledge, this is the first transport and application protocols agnostic privacy-preserving approach, capable to detect pivot attacks with complex network scenarios in near real-time.

2.11 Summary

This chapter presents an overview of Advanced Persistent Threat (APT), Privacy-preserving, the flow-based traffic analysis approach, Traffic profiling techniques, Compromise detection, Indicators of Compromise (IoC), Indicators of Attack (IoA), the CTI Framework concept and Pivoting.

Section 2.1 introduces the APT definition and the term evolution over the years. Section 2.2 presents various APT attack models pointing out differences, advantages and drawbacks. In Section 2.3 we address the advantages of privacy-preserving traffic analysis strategies in corporate networks and the challenges concerning different approaches. The emerging approach of flow-based traffic analysis is shown in Section 2.4 which was used in our work due to the success already achieved in solving traffic classification problems. Section 2.5 provides the importance of traffic profiling and clustering to understand behaviours and support decisions in a modern network administration. The definitions of IoC and IoA which are directly related to the Compromise Detection concept are addressed in Sections 2.6, 2.7 and 2.8 respectively. Section 2.9 presents CTI Frameworks' features and concepts. Finally, Section 2.10 presents the state-of-art of pivot attack detection in the context of cybersecurity.

CHAPTER 3

METHODOLOGY

This chapter provides an overview of the methods used to achieve the objectives of this research. Information concerning the instruments used to implement a proof of concept prototype of the pivot attack detection and classification algorithms (APIVADS and APCA) to generate pivot attacks and legit traffic are presented and followed by the evaluation metrics utilised to measure and compare results. Next, parameters optimisation tests details are provided. Also, the experiment scenarios conducted in virtual and real environments are described. Since the threat actors can apply intentional propagation delays, evasive pivot experiments scenarios were envisioned to test our algorithms' implementation. Finally, APIVADS parameters are provided to address APT attack stages inference based on the pivot tunnel traffic information.

3.1 Instruments

The instruments used in our experiments consist of the Golang programming language [150] to code the detection algorithms', qBittorrent [151], wget [152], and Firefox [153] to impersonate regular user traffic. OpenSSH [154]

was used to create the pivot tunnels, and Linux (Ubuntu) [155] was used as virtual OS machines.

3.2 Dataset creation

The network flow is a valuable data source to identify an ongoing attack and infer IoA within assets communication patterns. APIVADS detection algorithms process the traffic headers related to its agents' perceived traffic. The implementation developed to validate this thesis uses Tshark [156] to collect traffic into PCAP Capture File Format [157] and generate biflows. Tshark is a mature, well-documented, and lightweight tool capable of generating biflows traffic statistics. Therefore, the biflows attribute statistics related to the PCAP file of interest are processed by APIVADS detection algorithms to create IoA events when a pivot attack pattern is identified. In Appendix A Figure A.3 we present the biflows statistics generated by Tshark using the WireShark network protocol analyzer [158, 159] to illustrate it graphically. The biflows inside the red rectangle are part of a pivot attack since they present patterns that indicate a pivoting IoA. The cited patterns are presented in Chapter 4.

To achieve the capability to conduct reproducible experiments and identify the best APIVADS parameters to detect pivot attacks, a dataset was created for each APT attack stage defined in Table 3.2. Therefore, the following method was used:

1. A pivot attack technique and an APT attack stage are selected.

2. Based on the selection of the previous item, a scenario to support the experiment concerning software, hardware and network topology is created in a controlled environment.
3. APIVADS agents are deployed in the pivot nodes.
4. Based on APIVADS parameters and the experiment objectives, we simulate a real attack to generate traffic in the pivot tunnel.
5. All the traffic perceived by the pivot node is processed by the APIVADS agents while it is collected and stored into PCAP files.
6. All the APIVADS agents detection logs are collected and stored with the experiment PCAP files.

The dataset file names are composed of a numeric sequence followed by the first packet date time information. Additionally, the PCAP files contain packets within a time window. For instance, Appendix A Figure A.2 illustrates the set of PCAP files referent to the Command & Control attack stage, where each file corresponds to a time window of 60 seconds.

3.3 Evaluation metrics

Theoretically, a perfect classifier must not generate False positive (FP) or False negative (FN) errors. To evaluate our scheme's feasibility and effectiveness, we reference the metrics presented in Table 3.1 to measure and compare results.

3.4. EXPERIMENTS

Table 3.1: Evaluation metrics

Metric	Description
True Positive (TP)	The number of conversation pairs correctly identified as pivot tunnel.
True Negative (TN)	The number of conversation pairs correctly identified as not pivot tunnel.
False Positive (FP)	The number of conversation pairs wrongly identified as pivot tunnel.
False Negative (FN)	The number of conversation pairs wrongly identified as not pivot tunnel.
Detection Accuracy (DA)	Percentage of correctly identified conversation pairs $(TP+TN) / (TP+TN+FP+FN)$.
True Negative Rate (TNR)	Percentage of correctly identified conversation pairs as not pivot, $TNR = TN / (TN + FP)$.
True Positive Rate (TPR)	Percentage of correctly identified conversation pairs as pivot, $TPR = TP / (TP + FN)$.

3.4 Experiments

Experiments are paramount to evaluating the efficiency and accuracy of the APIVADS scheme, including optimisation tests to improve parameters and achieve better detection rates. Initially, a virtual network scenario was used to conduct APIVADS validation experiments. Next, we complemented it with real network scenarios to evaluate it against real-world connectivity challenges such as intentional propagation delays imposed by attackers, latency and packet loss.

APIVADS agents collect packets and aggregate them in biflows near real-

time. During the experiments, regular and malicious traffic is generated to simulate a typical workstation of an enterprise network. APIVADS agents were exposed to various scenarios with different protocols and services presented in this section.

3.4.1 Parameter optimisation tests

Pursuing the objective of achieving the best detection rates in our experiments, we performed several tests to identify the best parameter combinations for detection. The tests were conducted initially in a virtual network environment and later in complex network scenarios with real-world traffic propagation problems like latency and packet loss. Experiments have been performed to identify the maximum, minimum, and average values observed within standard enterprise network traffic and pivot attacks.

3.4.2 Virtual network experiments scenario

A virtual network environment infrastructure was built to carry out initial experimentation. It consists of five Linux virtual machines (Ubuntu 19.10 64 bits with 2GB RAM) impersonating a real environment and generating different types of standard enterprise traffic. Figure 3.1 presents the virtual network experiment diagram. Boxes represent the network hosts that are differentiated by colours and letters. IP addresses and network interface information is next to every host with the corresponding colour code. Every numbered red arrow represents a *BP*.

Regarding the pivot attack identified by bullets 1 and 2, Host A is the

3.4. EXPERIMENTS

attacker node, C is the target node, and B is the Pivot node. B provides the traffic forward between the network interfaces eth0 (LAN 1) and eth1 (LAN 2), supporting communication between A and C in different networks. We used SSH connections to create the pivot attack scenario. It corresponds to the propagation of Linux terminal commands between A and B, typically used by attackers (ex: “netstat”, “ifconfig”, “whois”, “whoami” and “ps aux”). Also, hosts D and E are not involved with pivot attacks, used in the experiment to generate regular enterprise network traffic and validate APIVADS regarding false-positive results.

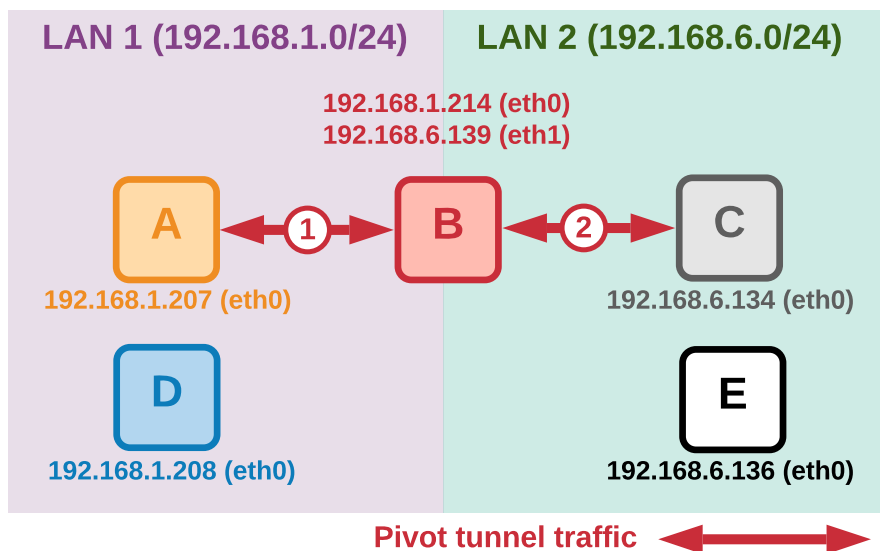


Figure 3.1: Virtual network experiment diagram

We seek to validate our implementation during the virtual network experiments and identify the influence of the parameters described in Table 4.1 in the detection scheme. To determine the ideal parameters combination re-

garding the pivot tunnel traffic volume and frequency, we used the evaluation metrics presented in Table 3.1.

3.4.3 Real network experiments scenario

The experiment's main objective in the real environment was to check APIVADS behaviour when exposed to common connectivity challenges such as latency and packet loss. Additionally, this scenario helps identify the impact of the cited connectivity drawbacks regarding pivot attack detection. An update regarding parameter analysis optimisation is conducted in this round of experiments to improve APIVADS detection results when dealing with complex scenarios in real environments. As performed in the virtual experiments scenario, we used the evaluation metrics presented in table 3.1 to evaluate the parameter combination results regarding the ongoing pivot tunnel traffic volume and frequency.

The pivot attack scenario illustrated by Figure 3.2 is created using the protocol SSH to support a two jump pivot tunnel over the internet (Pivot nodes 1 and 2). The pivot propagates malicious commands between the attacker and the target nodes. The assets used in the experiment are owned and remotely controlled by the authors and are located in different countries, WAN and IP ranges: attacker node is in the United Kingdom, Pivot node 1 in the United States, Pivot node 2 in Brazil and Target node in the Netherlands.

The type of traffic in experiments is similar to standard protocols used by enterprise networks (SMTP, IMAP, HTTP, HTTPS, and DNS). However,

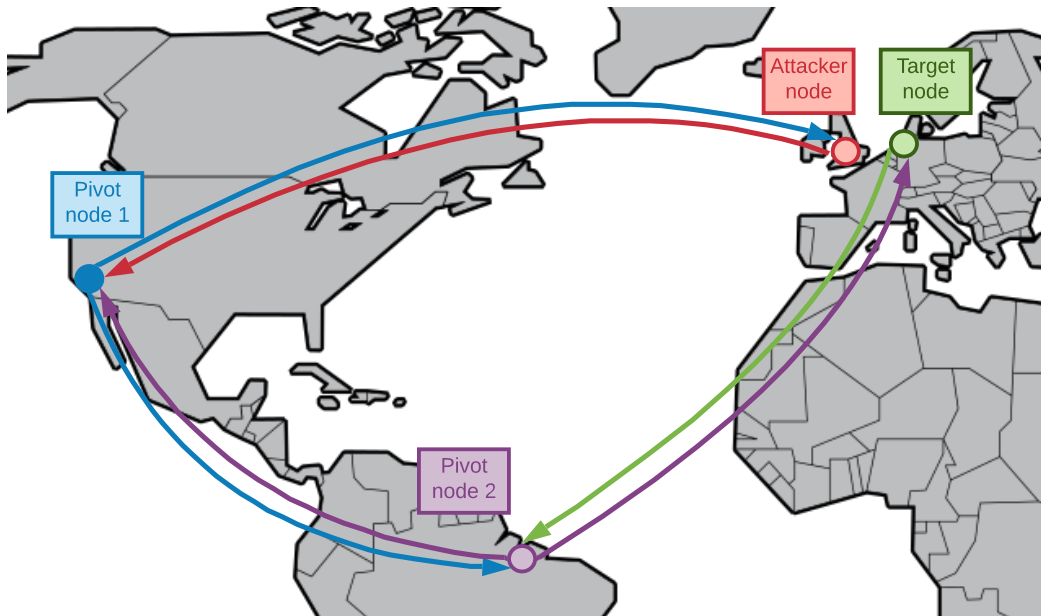


Figure 3.2: Pivot attack experiment conducted over the internet

unlike the virtual network experiments scenario, which is less complex and addresses different validations, the malicious traffic sent throughout the SSH pivot tunnel in the real network scenario simulates different attacks. Therefore, volume, frequency and payload variations are expected according to the experiment objective.

3.4.4 Evasive pivot techniques detection

Skilled adversaries can utilise techniques to manipulate the pivot tunnel traffic to evade detection. A known technique to avoid the correct classification from detection algorithms is to apply intentional propagation delay to the pivot traffic [15]. To determine if our pivot detection architecture can detect evasive pivot attacks, an experiment using the same scenario described in Subsection 3.4.3 regarding regular traffic and pivot tunnel was envisioned.

We applied intentional propagation delays to simulate a pivot attack conducted by an advanced opponent. This experiment observes if our detection architecture is resilient to intentional propagation delays and if a parameter change is necessary to achieve detection. The delays were applied to the incoming and outgoing malicious traffic at Pivot nodes 1 and 2.

3.4.5 APT attack stages inference

We address the possibility of identifying different APT attack stages based on P_{piv} traffic frequency and volume changes. This information helps predict the actual adversary objectives and possible next steps. Therefore, Table 3.2 shows three parameter templates optimised to detect pivot tunnels supporting different APT attack stages. Our primary purpose in this set of experiments is to verify if APIVADS can identify attack stage changes based on specific patterns of P_{piv} . Therefore, we used the same set of parameters of the previous experiment, just changing the specific parameters of each setup presented in Table 3.2.

Table 3.2: Attack stage inference parameters in function of P_{piv}

Setup	T_w	L	E	P_{piv}
Initial Exploitation	30s	80 packets	10s	5 PPS
Command & Control	1h	25 packets	60s	0.02 PPS
Data Exfiltration	10s	1000 packets	5s	1000 PPS

3.5 Summary

This chapter presents the APIVADS methodology to detect and classify pivot attacks. First, Section 3.3 defines the evaluation metrics used in this work to compare with other algorithms and measure efficiency. Next, to facilitate understanding of the results presented in Chapter 6, the following sections in this chapter provide details regarding the experiment scenarios conducted in this work.

CHAPTER 4

APIVADS: ADAPTIVE PIVOTING DETECTION SCHEME

This chapter presents APIVADS, an Adaptive Pivoting Detection Scheme. Initially, the pivot attack scenario we intend to detect is defined, and a brief description of privacy-preserving aspects of APIVADS is provided. Following this, the detection scheme is presented, and a detailed description of flow-based pattern recognition about pivot attacks is given. Finally, we explain APIVADS data processing phases, algorithms and details on the interaction between the scheme modules and entities.

4.1 Pivot attack

A pivot scenario is illustrated in Figure 4.1. The numbers 1-4 represent packet flows between nodes. Generally, the pivot technique is used when the attacker node cannot exchange information directly with the target node (numbers 5 and 6). Instead, to establish a connection with the target node, the attacker node needs to gain access to other network assets (pivot nodes), which are used to forward traffic between the attacker node and the target node.

A pivot attack must follow a logical sequence of communication flow. For

4.2. PRIVACY-PRESERVING CHARACTERISTICS AND DESCRIPTION

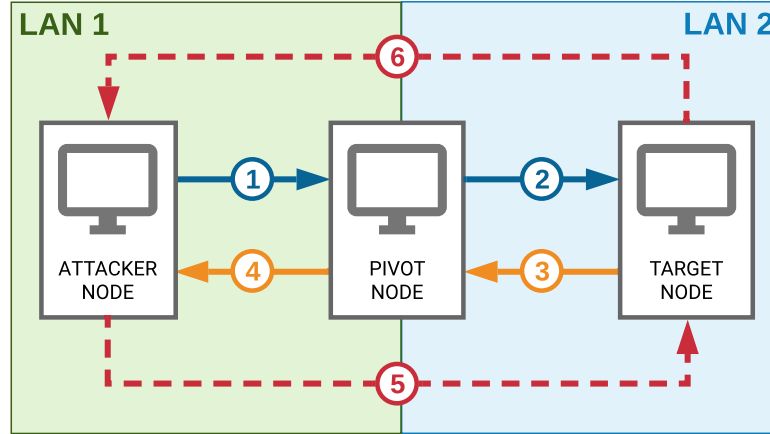


Figure 4.1: Pivot attack scenario

instance, the flow corresponding to number 2 (pivot node outgoing traffic and target node incoming traffic) will only exist when flow 1 reaches the pivot node. Using the same reasoning, we can assume that flow 4 exists after flow 3 arrives at the pivot node when the target node sends traffic to the attacker node. Another intrinsic pivot characteristic in the communication flow is that flows 1 and 2 always arrive before flows 3 and 4, with a slight time difference between them in terms of packet perception by the pivot node.

4.2 Privacy-preserving characteristics and description

The protection of privacy is crucial to the success of a defence solution. Unlike DPI approaches, APIVADS adopts flow-based analysis without requiring payload inspection, thus achieving compatibility with end-to-end encryption while disregarding packet payloads. It merely aggregates specific packet

header attributes to create flows. We proposed a specific flow attributes structure (see Table 4.2), explained in detail in Section 4.7, used as input to the detection algorithms. The approach does not need a knowledge repository, eliminating the constant update dependency of signature-based strategies and privacy concerns of detection rules. Cryptography is the standard approach to privacy preservation when sharing information among different parties [160]. However, the privacy-preserving concept only applies when sensitive information is shared with third parties. Therefore, the detection scheme is by default privacy-preserving, as the APIVADS agents' data processing is solely related to its host's traffic flow attributes and no sensitive information is shared with third parties. It does not demand anonymisation or additional privacy requirements regarding data processing.

4.3 Detection scheme and flow-based pattern recognition model

The traffic perception of APIVADS agents installed at pivot nodes forms the basis of the detection strategy. This node is responsible for forwarding traffic between the attacker node and the target node. An agent installed on the pivot node can perceive incoming and outgoing network traffic and infer biflows between itself and other endpoints. Based on the perceived biflow attributes, the APIVADS agent performs statistical calculations to find pivot attack patterns between biflows. Consequently, the agent installed in the pivot node is capable of inferring a pivot attack scenario by transforming the

4.3. DETECTION SCHEME AND FLOW-BASED PATTERN RECOGNITION MODEL

perceived traffic into APIVADS flows (defined in Table 4.2) and applying the detection filters described next in this chapter.

In Figure 4.1 we have two biflows between three hosts: The first one is represented by flows 1 and 4 (communication between the attacker and the pivot node). In contrast, the second one is illustrated by flows 2 and 3 (traffic between the pivot and the target node). To detect a pivot scenario under the assumptions presented above, we need to find a correlation between biflows. Let B be a biflow consisting of a set of incoming flows $F_i, i = \{1, 2, 3, \dots, n\}$, and a set of outgoing flows $F_o, o = \{1, 2, 3, \dots, n\}$ between certain endpoints. The direction of the flow (incoming and outgoing) is characterised by the observer (pivot node), and a biflow can be defined in the function of incoming and outgoing flows $B(F_i, F_o)$. Using Figure 4.1 as an example, if the pivot node can identify similarities for specific patterns between the conversations $B(1, 4)$ and $B(3, 2)$, a pivot scenario can be inferred between the attacker node and the target node supported by the pivot node.

Biflows within a pivot tunnel show a similar duration (time difference between the first perceived packet and the last). Excluding biflow pairs that do not match this premise is an efficient data reduction measure, as legit biflows with similar duration are unusual. Based on the assumption that packets within a pivot attack occur scattered in time, it is possible to use statistical methods to measure and compare the degree of alternation in packet arrival time between biflows to identify similarities in traffic.

Figure 4.2 illustrates two scenarios: a pivot attack pattern and a traffic

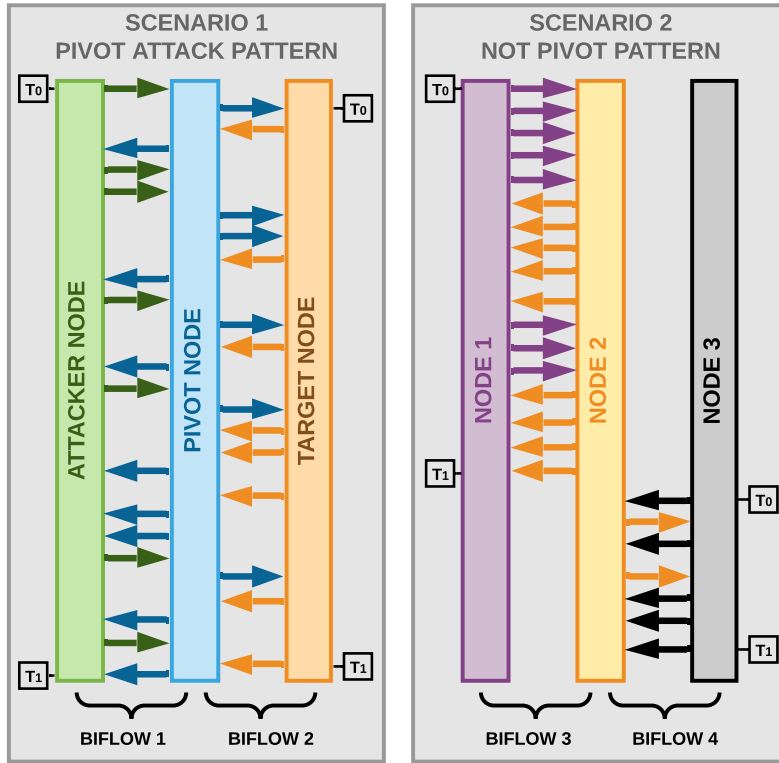


Figure 4.2: Pivot and not pivot traffic patterns comparison

pattern that does not correspond to a pivot attack (scenarios 1 and 2, respectively). The oriented arrows represent packets ordered chronologically from T_0 to T_1 , with the direction indicating origin and destination, T_0 representing the arrival time of the first packet and T_1 representing the last packet perceived within a biflow in a time window. For example, in scenario 1, biflow one is represented by the green and blue arrows between the attacker node and the pivot node. The first and last packet arrival times of the biflows are represented by T_0 and T_1 . Additionally, D_f corresponds to the biflow duration, which is the difference of T_1 minus T_0 .

4.3. DETECTION SCHEME AND FLOW-BASED PATTERN RECOGNITION MODEL

Let D_t be the absolute time difference between two biflows. The calculation of D_t is trivial and essential to take into account similarities between biflows in terms of duration, which correspond to an IoA of a pivot attack. The lower D_t is between two biflows, the similar their durations are. To compute D_t between biflow 1 (B_1) and biflow 2 (B_2), we calculate the absolute time difference between B_1 and B_2 . The D_t calculation can be expressed with the following equation: $D_t = |B_1(T_1 - T_0) - B_2(T_1 - T_0)|$. The D_t result is compared with a predefined parameter (pD_t), which is the maximum value of D_t , to identify biflows with similar duration. If the D_t result is more significant than pD_t , the evolved biflows do not have the necessary similarity for the duration. A biflow pair (BP) is composed of two biflows B_i and B_j that have similar duration, compatibility regarding packet alternation and plausible endpoint IP address correlation to be considered as part of a pivot tunnel. Section 4.8 introduces the three filter description and pseudocode algorithms responsible for data reduction, biflow pair identification and pivot tunnel detection.

Scenario 1 shows a high level of alternating traffic between the three assets with similar D_t in a time window that characterises a pivot attack. However, in scenario 2, two different concentrations of packets are observed. The first is between nodes 1 and 2 and the second between nodes 2 and 3, which does not correspond to the logic of a possible forwarding of pivot traffic between nodes 1 and 3.

A BP has a similar start time in a time window. Let D_s be the maximum

start time difference between two biflows. It can be calculated by the absolute difference between the biflows start times $D_s = |B_3(T_0) - B_4(T_0)|$. Similarly to D_t , the smaller D_s is, the similar the biflows are for the start time. For instance, comparing biflow 3 (B_3) and biflow 4 (B_4) for the D_s calculation in scenario 2, the start time of the biflows is not in the same time window. Consequently, they are discarded as candidates for a BP by the duration filter algorithm presented in Algorithm 1.

Assuming that in a classic pivot attack, the pivot node only forwards traffic between endpoints, related biflows tend to present a similar number of total packets N . Therefore, we defined the parameter pN , corresponding to the maximum result in the ratio computation between the biflows' N values.

Finally, identifying a BP is a strong IoA for a pivot attack. The APIVADS detection scheme uses the following criteria to infer a pivot attack: (1) The D_t result regarding B_i and B_j must be lower than pD_t . (2) Both biflows must have D_f greater than pD_f . (3) The D_s result with respect to B_i and B_j must be lower than pD_s . (4) The calculation of the N ratio between two biflows must be lower than pN . (5) The arrival time of the packets must alternate between the biflows.

4.4 APIVADS data processing phases and threat model overview

The detection strategy comprises two distinct data processing phases: detection and aggregation. First, we use a host-based approach to address the

detection. In this phase, the APIVADS agent collects the traffic headers perceived by the device, updates the set of biflows and processes them using data reduction and statistical techniques to infer a pivot attack. When a new packet is perceived, the extracted attributes in a production implementation should never be stored on disk; they should automatically be aggregated to the set of biflows to save storage resources. Second, a distributed approach aggregates the agents' pivot attack detection information. When an APIVADS agent detects a pivot attack, it reports the event to a third-party CTI Framework that aggregates all pivot attack events to identify connections among messages and infer the complete pivot tunnel. Figure 4.3 illustrates the two detection phases that will be explained in more detail in this chapter.

4.5 APIVADS modules interaction

APIVADS uses a distributed strategy based on agents installed in network assets to identify pivot tunnels of any length. Figure 4.3 illustrates the data processing steps and interaction among the four APIVADS agent modules. The data collection module receives traffic information from the device's network interfaces and continuously collects packet header attributes of interest from new traffic. The data extraction module aggregates the collected header attributes and updates a set of APIVADS flows whose structure is presented in Table 4.2. Therefore, APIVADS flows are clustered in biflows that are passed to the detection filter module, responsible for data reduction and

CHAPTER 4. APIVADS: ADAPTIVE PIVOTING DETECTION SCHEME

derivation of biflow pairs in the context of a pivot attack. Finally, the agent interaction module is responsible for interacting with CTI frameworks as a Threat Intelligence Feed (TIF) to integrate the proposed scheme with other defence solutions providing alerts, and actionable information [161, 162]. All modules and interactions among APIVADS entities are explained in detail hereafter.

Parameters are envisioned in APIVADS to optimise the detection scheme and provide control and balance of detection metrics. Table 4.1 contains a list of parameters used in APIVADS detection filter algorithms.

APIVADS data reduction parameters (pD_t , pD_f , pD_s and pN) are used by the detection filter module algorithms to discard *BP* candidates based on biflows similarities and specific characteristics. L , T_w and E are performance parameters. L directly influences the number of packets processed by the algorithm within a biflow. Small values of T_w can restrict the amount of data sampled, while large values demand time and processing power. The E parameter defines the algorithm's detection execution frequency. Since third-party outsourcing is beyond the scope of this work, the time spent on data processing and algorithm execution (T_p) must be less than dividing T_w by E , to avoid exhausting data processing resources. Finally, pR influences the algorithm's detection accuracy. A restricted value of pR may increase false negatives, while a tolerant value will likely lead to a false positive scenario. All parameters are described in detail by the algorithm's pseudocode as described in Section 4.8.

4.5. APIVADS MODULES INTERACTION

Table 4.1: Detection scheme algorithms' parameters

Parameter	Description
pD_t	The maximum value of D_t computation between two biflows is compatible with a BP pattern total duration.
pD_f	The minimum biflow duration value considered by the detection scheme.
pD_s	The maximum value of D_s calculation between two biflows to be considered compatible with a BP pattern regarding absolute start time difference.
pN	The maximum value of the N computation between two biflows to be considered compatible with a BP pattern in terms of total bytes traffic ratio.
L	The number of most recent packets to be considered within a flow.
pR	The maximum value of R calculation between two biflows to be considered compatible with a BP pattern regarding sequences of packets of the same flow.
T_w	A parameter restricting the algorithm to process biflows within a specific time interval (detection time window).
T_p	Corresponds to the necessary time to process the data and execute the detection algorithms.
E	A parameter to specify the execution of the detection algorithms (detection execution frequency).

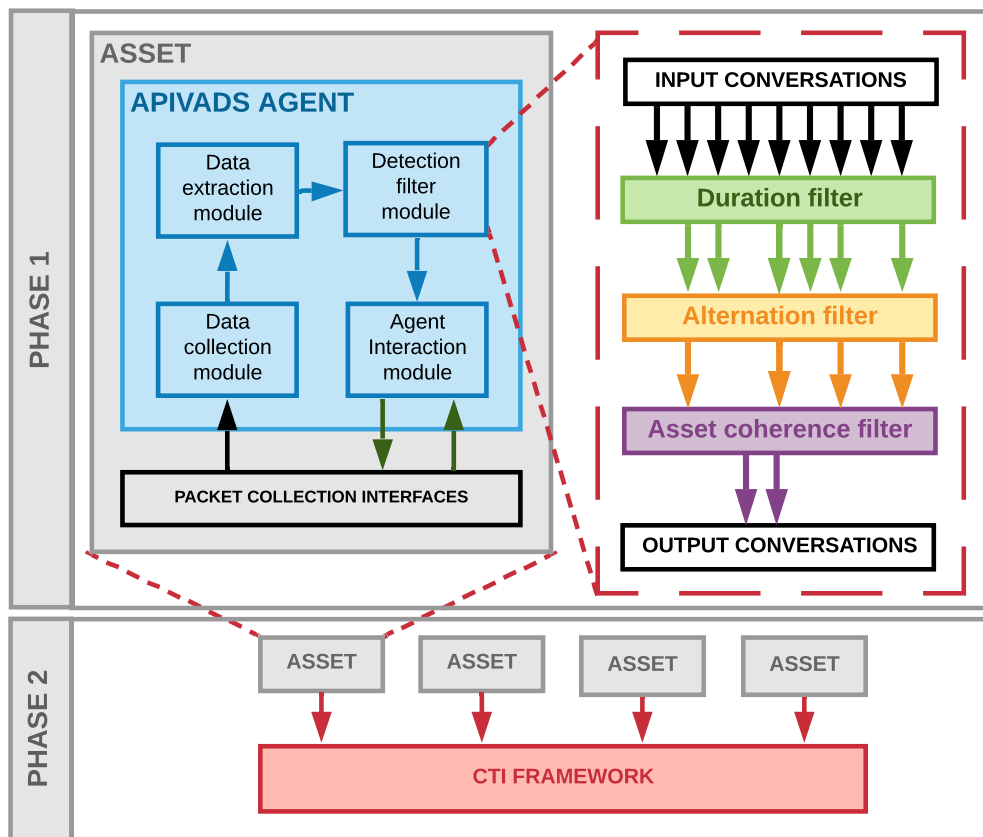


Figure 4.3: APIVADS threat model diagram

4.6 Data collection module

This module is responsible for collecting and aggregating the perceived packets' metadata while preserving temporality. The flow-based approach uses biflows and passive network monitoring (see Section 2.4), evaluating the arrival time of the most recent packets within biflows. In addition, the detection scheme controls the number of packets considered in a biflow using the L parameter. In other words, this variable provides a mechanism to control the sample size for biflows.

To measure the biflow duration, we adopted the following lifespan rules:

1. If no packets belonging to a specific biflow are perceived within 60 seconds (inactive timeout).
2. If the biflow duration time reaches 1 hour (active timeout).

Additionally, the pD_f parameter corresponds to the minimum biflow duration value considered by the detection algorithm. This parameter helps improve data reduction and discard irrelevant biflows due to the type of attack we intend to detect.

We do not use FIN or RST flags included in TCP packet attributes because UDP and other protocols does not contemplate it and APIVADS is agnostic regarding the transport layer. Besides, the adversary can bridge data from different transport layer protocols (e.g. TCP to UDP bridge [149]) to bypass defence mechanisms.

4.7 Data extraction module

The packets gathered in the data collection module are transformed into flows that are processed into conversations, which are forwarded as input to the detection filter module. Some common flow attributes are discarded in this module to avoid unnecessary processing, reduce storage requirements and achieve a lightweight detection scheme. In addition, the detection scheme does not use protocol-specific attributes since we adopt an agnostic approach regarding the transport and application layers. The selected flow feature attributes, and data structure are depicted in Table 4.2.

Table 4.2: APIVADS flow attributes structure

Attribute	Type	Example
Flow identification	hash	0xBABF4E7C
Date-time reference	timestamp array	[2018-03-13 12:22:10.353, 2018-03-13 12:22:11.642, ... 2018-03-13 12:22:20.134]
Transport protocol	categorical	TCP
Source IP address	categorical	192.168.0.5
Source port	categorical	52128
Destination IP address	categorical	192.168.0.7
Destination port	categorical	8080
Total bytes	numeric	120

In the detection scheme, a flow can be represented by the following 8-tuple: $F = \{I, T, T_r, S, S_p, D, D_p, N\}$. Let I be the flow identification, T

be an array of packet arrival timestamps $T = \{t_1, t_2, t_3 \dots t_n\}$, T_r be the transport protocol, S be the source IP address, S_p be the source port, D corresponds to the destination IP address, D_p be the destination port and N be the total number of bytes within the flow.

Then we create biflows based on flows that include packets sent in both directions and share the same endpoints. Finally, we infer new attributes to the biflows based on the merged flows (total number of bytes, relative start and duration).

4.8 Detection filter module

This module performs data reduction and statistical pattern recognition using three filter algorithms that receive a set of biflows as input. The first filter is the duration filter algorithm (Algorithm 1). It is responsible for reducing the biflows output from the data extraction module, which does not fit in a pivot attack pattern. Initially, biflows with a duration lower than the predefined parameter pD_f are discarded to avoid ephemeral connections. This module verifies the level of similarity between two biflows for the data reduction parameters. For example, suppose the result D_t is less than or equal to pD_t . In that case, the biflows show a degree of similarity in terms of duration that is consistent with a pivot attack. The resulting biflows (B_1 and B_2) are selected as candidates to form a new biflow pair $BP(B_1, B_2)$.

The biflow pairs created in the duration filter are used as input to the alternation filter algorithm (Algorithm 2). This algorithm checks whether the

CHAPTER 4. APIVADS: ADAPTIVE PIVOTING DETECTION
SCHEME

Algorithm 1: Duration filter algorithm

- Input** : A set of biflows $B = \{B_1, B_2, B_3, \dots, B_n\}$, where each element is composed of flows that share the same source IP, destination IP, source port and destination port within a time window.
- Parameter:** pD_t is a predefined parameter that is compared to the result of D_t . It corresponds to the maximum limit of the D_t calculation between biflows to create a BP in terms of duration.
- Parameter:** pD_f is a predefined parameter that is compared to each biflow D_f value. If D_f is less than pD_f , the biflow is discarded.
- Parameter:** pD_s is a predefined parameter that is compared to the result of D_s . It corresponds to the maximum limit of the D_s computation between biflows to create a BP regarding the absolute start time difference.
- Parameter:** pN is a predefined parameter compared to the calculation of the total byte traffic ratio between two biflows.
- Output** : An array of biflow pairs BP

- 1 Compares each biflow D_f value with pD_f . If D_f is less than pD_f , the biflow is discarded.
 - 2 The remaining biflows in B are compared with one another. For the sake of simplicity, we name the biflows to be compared B_i and B_j .
 - 3 **if** the D_t result of B_i and B_j computation is lower than pD_t
 - 4 **and** D_s the result of B_i and B_j computation is lower than pD_s
 - 5 **and** pN is greater than the ratio of B_i and B_j with respect to N
then
 - 6 | Append the biflow pair to the result array $BP(B_i, B_j)$
 - 7 **else**
 - 8 | Select the next biflow candidates until all remaining eligible
| biflows are tested against each other.
 - 9 **end**
 - 10 **return** BP
-

biflows that compose a BP show an alternation about the packet arrival time in the pivot node. First, the date and time reference attribute array of the

4.8. DETECTION FILTER MODULE

flows that compose a BP is merged chronologically and ordered, preserving the flow identification. The algorithm then processes the array to calculate the R value, which is the maximum packet sequence of the same flow in the merged array. Finally, the achieved value of R is compared to a predefined parameter pR , which corresponds to the maximum packet sequence in the same flow. Therefore, an R value more significant than pR is discarded from the set of BP received from the previous filter.

Algorithm 2: Alternation filter algorithm

Input : A set of biflow pairs $BP = \{BP_1, BP_2, BP_3, \dots, BP_n\}$

Parameter: L corresponds to the number of most recent packets to be considered within a flow.

Parameter: pR is a predefined parameter compared to the result of R computation. If R is greater than pR , the biflow is discarded.

Output : An array of biflow pairs BP

- 1 The two biflows that make up a BP are merged into a temporary array, preserving the flow identification and packet arrival time chronology.
 - 2 The temporary array size is limited by the L value and is filled with the most recent packets arrival time attribute.
 - 3 The algorithm searches the temporary array for the biggest sequence of packets within the same biflow R .
 - 4 **if** R result is greater than pR **then**
 - 5 | Exclude the biflow pair from the BP array
 - 6 **else**
 - 7 | Process the next BP element until the last entry
 - 8 **end**
 - 9 **return** BP
-

The last filter is the asset coherence filter algorithm (Algorithm 3). It

CHAPTER 4. APIVADS: ADAPTIVE PIVOTING DETECTION SCHEME

receives the remaining BP that meet the requirements of the previous filter: biflows pairs with duration time greater than pD_f , a similar D_t , D_s less than pD_s , N ratio calculation less than pN and present alternation between packets regarding different flows. This module checks for reasonable correlation of the BP set of IPs H with the endpoints and discards inconsistent pairs (e.g. biflows with the same source and destination IP and ports). Additionally, the merged set of chronologically ordered packet timestamps is split into quarters. Each quarter must contain all possible flows within the biflows, forming a BP that validates the alternation between flows across the traffic sample. For example, a BP composed by the biflows B_1 and B_2 have four flows $B_1(F_i)$, $B_1(F_o)$, $B_2(F_i)$ and $B_2(F_o)$ that must be present in all quarters.

Algorithm 3: Asset coherence filter algorithm

Input : A set of biflow pairs $BP = \{BP_1, BP_2, BP_3, \dots, BP_n\}$ **Input** : A set of IPs $H = \{IP_1, IP_2, IP_3, \dots, IP_n\}$ where each element corresponds to the local asset**Parameter:** L is a predefined parameter corresponding to the number of most recent BP packets considered by the algorithm**Output** : An array of biflow pairs BP

- 1 Compare the source and destination IP attributes of the biflows that compose BP (B_i and B_j biflows) with the device set of IP addresses H
 - 2 Check if B_i and B_j source or destination IP attributes contain an IP in the H array
 - 3 Merge the two biflows that compose a BP entry into a temporary array while maintaining the flow identification and the packet arrival time chronology.
 - 4 Split the temporary array data into quarters: Q_1, Q_2, Q_3 and Q_4 .
 - 5 **if** *All flows forming the B_i and B_j biflows of BP (see Section 4.3) are not present in all quarters (Q_1, Q_2, Q_3 and Q_4)* **then**
 - 6 | Exclude the biflow pair entry from BP
 - 7 **else**
 - 8 | Process the next BP element until the last entry
 - 9 **end**
 - 10 **return** BP
-

4.9 Agent interaction module

This module is responsible for interacting with CTI frameworks. For example, when the detection filter module identifies a pivot attack, an alert is generated and forwarded to the CTI framework, which has a holistic view of all alerts received from APIVADS agents.

The detection scheme does not require any external information to identify an asset as a pivot node and to establish a connection between two

CHAPTER 4. APIVADS: ADAPTIVE PIVOTING DETECTION SCHEME

other assets. However, it cannot detect the complete length of the pivot tunnel. Therefore, the CTI framework must aggregate all alerts with similar attributes to identify connections between pivot nodes and determine the entire pivot tunnel. Through the distributed pivot detection strategy, the detection scheme achieves scalability and the possibility of being deployed in complex networks.

Every Pivot Attack Alert Message (PAAM) identifies a pivot node and two other devices involved in the attack. Table 4.3 presents two PAAM samples received by the CTI framework from pivot nodes.

Table 4.3: Pivot Attack Alert Messages sample

ID	Date-time	Transp	SrcIP	SPort	DstIP	DPort
#1	2021/02/25 11:13:41	TCP	192.168.6.135	49768	192.168.6.134	22
#1	2021/02/25 11:13:41	TCP	192.168.6.134	43316	192.168.6.132	1979
#2	2021/02/25 11:13:42	TCP	192.168.6.134	43316	192.168.6.132	1979
#2	2021/02/25 11:13:42	UDP	192.168.6.132	37564	192.168.6.131	22

Analysing the information in Table 4.3 based on an acceptable time difference and the same endpoint attributes shared between PAAMs is trivial to infer a connection among the pivot nodes. For instance, rows 2 and 3 share the same attributes with a reasonable time difference, implying a pivot tunnel length of two. The inferences from the alerts in Table 4.3 can be

4.9. AGENT INTERACTION MODULE

interpreted as a pivot tunnel diagram (see Figure 4.4).

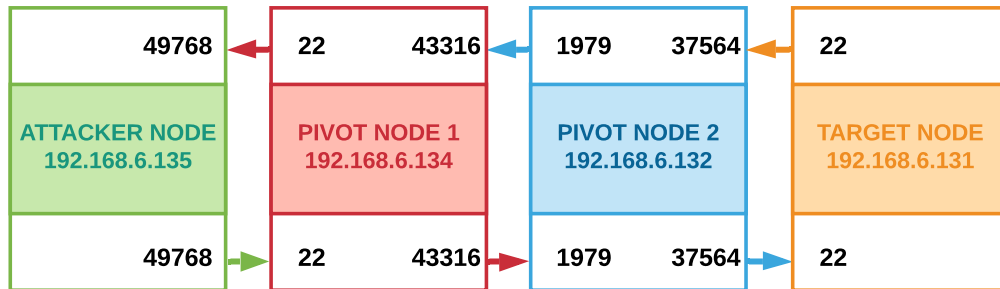


Figure 4.4: Pivot tunnel representation of Table 4.3 alert messages sample

Some advanced pivot attack tools are capable of bridging traffic between transport layer protocols. The detection scheme can infer a pivot attack even if the adversary uses such techniques, as illustrated by the second alert where pivot node 2 receives TCP and forward UDP traffic.

The approach does not rely on CTI frameworks to detect pivot attacks or define the pivot length. The connection between messages is trivial and could be done in several ways without requiring much computational effort. However, the distributed pivot results need to be merged somehow. We chose to send alerts to CTI frameworks based on the assumption that this kind of threat information is vital for the proactive identification of APT actors and attribution.

4.10 Summary

This chapter presents the Adaptive Pivoting Detection Scheme (APIVADS). Initially, we provide details regarding the APIVADS detection scheme, modules and algorithms. Next, is presented the distributed data processing strategy of interaction among agents. Finally, the CTI framework interaction to address pivot tunnels of any length is presented to the reader.

CHAPTER 5

PIVOT ATTACK CLASSIFICATION

Besides the detection scheme provided by APIVADS, we propose in this thesis a pivot attack classification criteria based on the connectivity achieved by the attacker within the target network. Pivot attacks can manifest differently because the attacker must adapt the pivoting TTP according to the network defences and topology. A high degree of connectivity provides more possibilities to the opponent regarding TTP usage compared to restricted scenarios. The criteria presented in this thesis deliver helpful information to comprehend the adversary’s capabilities and modus operandi. Additionally, the proposed classification can aggregate value to threat intelligence models providing pivot attacks IoA details.

5.1 Classification criteria

According to [163], pivoting techniques can manifest in two ways: “Proxy” and “VPN” (Virtual Private Network) pivoting. The proxy pivoting is characterised by a bidirectional traffic tunnel between the Attacker Node (AN) and the Target Node (TN) supported by proxies or port forwarders installed in the Pivot Nodes (PN). The main objective of a proxy is to relay application

5.1. CLASSIFICATION CRITERIA

data between clients and servers that may not have direct IP connectivity [164]. Therefore, a proxy pivoting inherits a proxy service's features and limitations, typically restricted to specific TCP and UDP ports. A VPN represents a temporary extension of the corporate network [165]. It uses a virtual network interface that provides layer-2 access to the target's network. This technique allows attackers to route traffic through the PN to a different network, providing transparent connectivity within the target. Therefore, it is a desirable scenario from the attacker's point of view, providing more possibilities concerning TTP when compared with proxy pivoting.

The binary pivot attack classification criterion proposed by [163] is adopted by a few offensive cyber security products [166, 167]. In addition, one can find this criterion in various informal sources of knowledge such as blog posts and tutorials [168, 169]. However, it is limited in terms of definitions to address complex pivot attack scenarios and TTPs.

We argue that a binary classification of pivot attacks is simplistic and does not provide the required granularity to express pivoting correctly. The opponent can achieve different possibilities regarding distinct degrees of connectivity. Therefore, the current definition can lead to confusion because some variations of pivot attacks can achieve full network access regarding specific protocols and ports (e.g. TCP) over the target network using a transparent proxy. For example, the Sshuttle Project [170] provides a tool which cannot be classified as a VPN nor Proxy pivoting. While it presents VPN characteristics since it can forward every port of a specific protocol on

an entire network, on the other hand, it uses the ssh protocol to forward traffic. According to the tool authors, “Sshuttle assembles the TCP stream locally, multiplexes it statefully over an ssh session, and disassembles it back into packets at the other end”.

Another pivoting method is proposed by Chisel [171]. It is a fast TCP/UDP tunnel transported over HTTP and secured with SSH. The tool provides several possibilities regarding connectivity, such as SSH over HTTP, reverse proxy, multiple tunnel endpoints over one TCP connection, and compatibility with SOCKS or HTTP CONNECT proxies. Chisel is mainly used to bypass firewalls and allow access to multiple protocol services and ports over the target network.

Both cited pivoting solutions cannot be classified as VPN or proxy pivoting, indicating an evident lack of classification granularity. Therefore, providing an accurate description of the pivot attack is necessary to increase the range of classification possibilities. Table 5.1 proposes a new nomenclature based on the OSI model [172] and on different degrees of connectivity achieved by the pivot tunnel.

To exemplify the proposed pivot attack classification, Figure 5.1 illustrates a scenario where the opponent compromises a device inside a Demilitarized Zone (DMZ), and the attacker node is located on the internet. A DMZ is a physical or logical subnetwork that exposes services to untrusted networks. It provides an additional layer of security to an organisation’s Local Area Network (LAN) and denies the attacker’s direct access to Net-

5.1. CLASSIFICATION CRITERIA

Table 5.1: Pivot attack classes

Pivot class	Description
Class I	A pivot scenario where the adversary achieves connectivity to a single host and is limited to a specific network protocol and transport layer (IP and port).
Class II	Refers to a pivot scenario where the opponent achieves connectivity to a single host and is limited to a specific network protocol and IP. However, the attacker can access different ports regarding the transport layer.
Class III	A pivot scenario where the opponent achieves unrestricted connectivity to a single host regarding the network and transport layer.
Class IV	A pivot scenario where the adversary can connect to different hosts but is restricted to the same network protocol layer (e.g. TCP) with no restrictions regarding the transport layer.
Class V	A pivot attack where the opponent has unrestricted network access on the targeted network.

work 1. Differently from predictions techniques and algorithms used to infer knowledge as [173, 174], our classification criteria, the pivot attack is characterised by the degree of connection achieved by the pivot node to the target nodes. The combination of red numbers corresponds to distinct degrees of connectivity. Let number 1 be the bifold that connects the attacker node to the pivot node, and numbers 2, 3, 4 and 5 biflows from the pivot node to the target nodes.

To characterise the simplest expression of a pivot attack (Class I), numbers 1 and 2 are sufficient because they fit the requirement of a connection to a single host limited to a specific network protocol, IP and port. To infer a Class II pivot attack is necessary to identify connectivity to a single host in

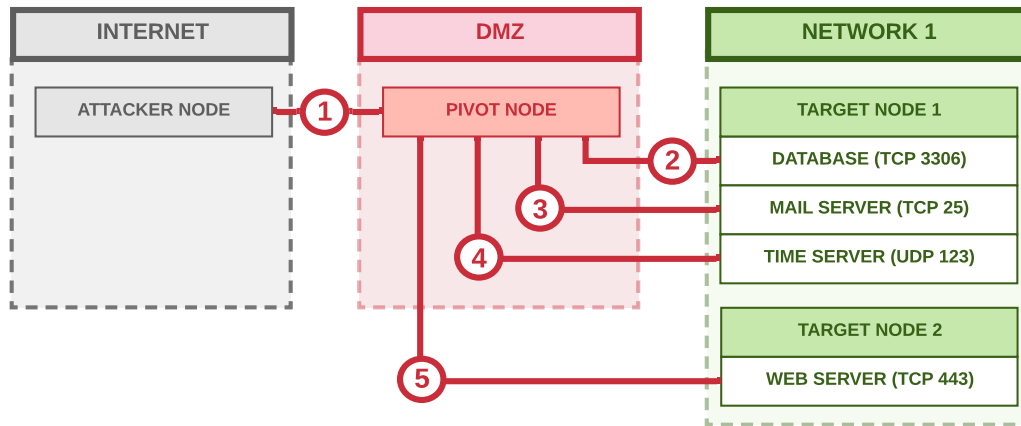


Figure 5.1: Pivot attack classification scenarios

the same network protocol but in different ports. This scenario can be characterised by combining numbers 1, 2 and 3, for instance. A Class III pivot tunnel is defined when the opponent achieves connectivity to a single host in different network protocols and ports (numbers 1, 2 and 4). A Class IV pivot tunnel requires connectivity to different hosts in the same network protocol (e.g. TCP) with no restrictions regarding ports. This scenario is exemplified by the combination of numbers 1, 2, 3 and 5. Finally, a Class V pivot tunnel is characterised when the opponent achieves unrestricted network access on the targeted network with connections to different protocols and ports with multiple hosts.

5.2 Semantic Network Models (SNM)

This Section presents the Semantic Network Models (SNM) to complement the pivot attack classification criteria described in Section 5.1. The main

objective of a pivot attack is to achieve bidirectional communication with a device of interest when a direct connection impossible. The attacker can use various techniques and tools to conduct a pivot attack. Due to the infinity of logical and physical network configurations and TTP variations, it is unfeasible to address a model that fits all possible pivot attack schemes. However, we can create generic SNM to express the pivot attack interactions and traffic flow based on the number of network interfaces evolved within the pivot node.

Every pivot attack class this thesis addresses can manifest using a single or dual network interfaces traffic forwarding strategy. The adversary TTP selection between single or dual network interfaces will depend on the credentials achieved (privileged or not privileged user) and connectivity obstacles to overcome. For example, a single network interface pivot tunnel typically is used in simple scenarios where the adversary does not need to forward traffic between different networks and does not have privileged credentials to change interface configuration and routing rules. Figure 5.1 numbers 1 and 2 illustrate the single pivot network interface scenario described when the adversary needs to achieve connectivity with one port on the target node, for example. On the other hand, the dual network interface pivot tunnels usually are applied when the opponent faces a complex scenario that imposes traffic forwarding between different networks and the adversary achieved privileged credentials in the pivot node. Both types of semantic models are presented in detail next.

5.2.1 Single interface semantic model

A single interface pivot tunnel is commonly used in the early stages of an attack when the pivoting techniques that provide better results regarding connectivity are not feasible. In addition, a single interface pivot attack typically does not require privileged access to forwarding traffic between endpoints.

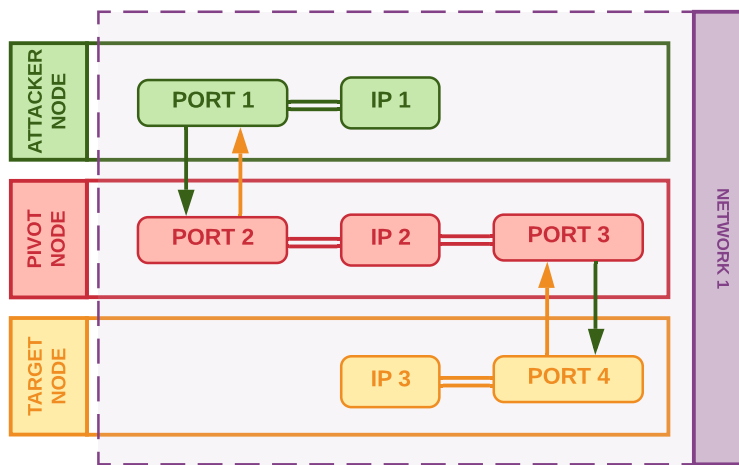


Figure 5.2: Single interface semantic network model

Figure 5.2 illustrates a single network interface pivot semantic model, where the attacker uses the pivot node to forward traffic between the attacker node and the target node. In this scenario, the pivot node forwards the traffic bridging ports 2 and 3 using operating system native commands or specific applications (e.g. malware and network utility software). Regarding network-level IoA, the pivot node typically uses one IP address (IP2) to support the attack, forwarding traffic between the attacker node and the target node when direct access to the resource of interest is not possible due

to connectivity restrictions.

5.2.2 Dual interface semantic model

The TTP used to achieve unrestricted connectivity within the devices of a different network usually requires route manipulation, elevated privileges and more than one network interface. Figure 5.3 represents a dual interface semantic model, where the pivot node uses two different IP addresses (IPs 2 and 3) to route the traffic from network 1 to network 2. Pivot attacks that require a dual interface provide full network connectivity between the attacker and the target node. Concerning network-level IoA, the pivot node must forward traffic between two different networks, requiring two IP addresses.

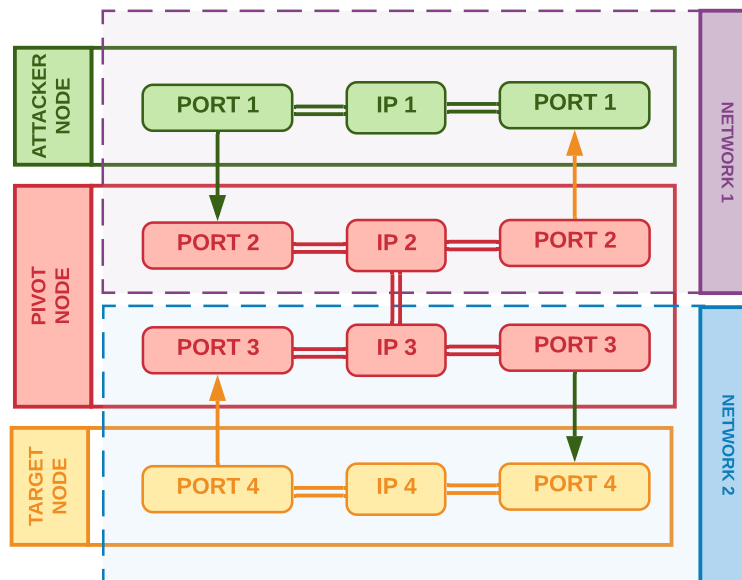


Figure 5.3: Dual interface semantic network model

5.3 Automatic Pivot Classifier Algorithm (APCA)

A PAAM P is generated when the APIVADS agent finds a pivot attack traffic pattern. The new message is forwarded to the CTI framework that process it to infer correlation among the set of PAAMs received from the agents. Suppose the cited algorithm identifies a correlation among previous PAAMs being part of the same attack. In that cases, it creates a group of PAAMs G which can be represented as the following expression: $G = \{P_1, P_2, \dots, P_n\}$. When a new PAAM P_{n+1} is processed and identified as part of an observed pivot attack, APIVADS insert the new message into the correspondent group of pivot attack messages.

APIVADS detection scheme can infer a pivot tunnel of any length correlating PAAMs from different agents (See [3] for more details), which are concentrated into a CTI framework. Appendix A Figure A.1 present this thesis implementation of APIVADS PAAM IoA events. When the PAAMs are processed and the detection scheme infers correlation among messages, a group G of PAAM is created. The group formation is based on an acceptable time difference and endpoint attributes shared among PAAMs. For instance, lines 2 and 3 of Table 5.2 correspond to the second biflow of message ID #1 $P_1(B_2)$ and first biflow of message ID #2 $P_2(B_1)$ respectively, indicating a connection between the cited messages because $P_1(B_2)$ and $P_2(B_1)$ share the same attributes except for the date time reference. Therefore, the detection scheme creates a group of PAAM G composed by the messages identified

5.3. AUTOMATIC PIVOT CLASSIFIER ALGORITHM (APCA)

with ID #1 and #2, which can be expressed as $G = \{P_1, P_2\}$.

Table 5.2: APIVADS Alert messages sample [3]

ID	Date time	Transp	SrcIP	SPort	DstIP	DPort
#1	2021/02/25 11:13:41	TCP	192.168.6.135	49768	192.168.6.134	22
#1	2021/02/25 11:13:41	TCP	192.168.6.134	43316	192.168.6.132	1979
#2	2021/02/25 11:13:42	TCP	192.168.6.134	43316	192.168.6.132	1979
#2	2021/02/25 11:13:42	UDP	192.168.6.132	37564	192.168.6.131	22

The APCA initially classifies every group of pivot tunnels as Class I. However, based on the attributes observed in the PAAMs of the group, the algorithm can infer different classes and update the classification based on evidence that the attacker achieved a more significant level of connectivity within the target network.

Figure 5.4 illustrates Table 5.3 representing four PAAMs (P_1, P_2, P_3 and P_4) related to a Class V pivot tunnel. LAN1 and LAN2 are local area networks containing squares identified with a single letter inside. Each square represents a device with one or more IP addresses attached to network interfaces. The red arrows with numbers symbolise pivot traffic between endpoints. Host B is the pivot node, which has two network interfaces (eth0 and eth1) and can route the traffic between LAN 1 and LAN 2, supporting the pivot tunnel from the attacker node (host A) to the target nodes (C, D and

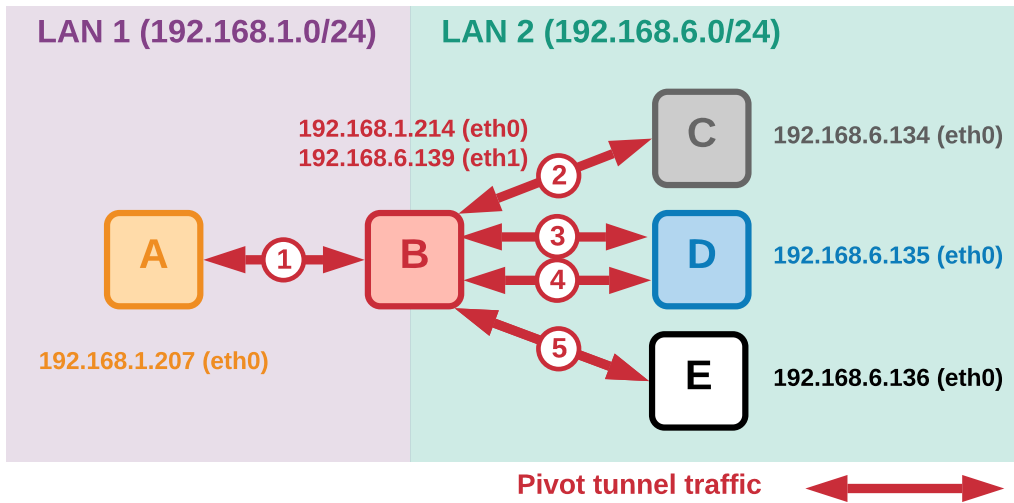


Figure 5.4: Class V pivot scenario diagram

E).

In Figure 5.4 the attacker node (host A) used the pivot node (host B) to achieve connectivity with the target nodes in LAN 2 (hosts C, D and E). A typical pattern regarding flow-based IoA within a set of PAAMs to characterise a Class V pivot attack is a repetitive biflow within different messages. For example, number 1 (pivot traffic from A to B) of Figure 5.4 corresponds to the first biflow of the messages with ID #1, #2, #3 and #4 of Table 5.3, presenting the same attributes except for a time difference. However, the second biflow of each PAAM present differences regarding several attributes. This pattern indicates that the attacker can connect from LAN 1 to different devices in different protocols, ports and services at LAN 2.

Considering just the PAAMs with IDs 1 and 2, we can infer a Class IV pivot because the adversary was observed connecting to different hosts using

5.3. AUTOMATIC PIVOT CLASSIFIER ALGORITHM (APCA)

the same network protocol (TCP) but in different transport layer ports (80 and 22). On the other hand, When the message with ID 3 arrived, the algorithm should reclassify the pivot attack to a Class V because the opponent achieved connectivity to different hosts in different network protocols (TCP and UDP) and transport layers (80, 22 and 444).

Table 5.3: Class V pivot attack alert messages scenario

ID	Date time	Transp	SrcIP	SPort	DstIP	DPort
#1	2021/05/23 11:09:39	TCP	192.168.1.207	42474	192.168.1.214	22
#1	2021/05/23 11:09:39	TCP	192.168.6.139	41486	192.168.6.134	80
#2	2021/05/23 11:14:31	TCP	192.168.1.207	42474	192.168.1.214	22
#2	2021/05/23 11:14:31	TCP	192.168.6.139	39450	192.168.6.135	22
#3	2021/05/26 11:21:53	TCP	192.168.1.207	42474	192.168.1.214	22
#3	2021/05/26 11:21:53	UDP	192.168.6.139	59742	192.168.6.135	4444
#4	2021/05/26 11:24:37	TCP	192.168.1.207	42474	192.168.1.214	22
#4	2021/05/26 11:24:37	UDP	192.168.6.139	52124	192.168.6.136	53

Algorithm 4 represent the pseudocode to achieve automatic pivot attack classification based on APIVADS PAAMs. Let G be a group of PAAMs received from APIVADS. Every message P is composed by two biflows (B_1 and B_2), Therefore, $P = (B_1, B_2)$. Since all groups are pre-classified as Class

I pivot because it is the simplest pivot attack scenario, our algorithm will reclassify the group if necessary by analysing G set of PAAMs, which can be expressed according to the following expression: $G = \{P_1, P_2, P_3 \cdots P_n\}$.

Algorithm 4: Automatic pivot classifier algorithm

Input : A group of PAAMs G .
Output : A classification C (According to Table 5.1) related to the input group G .

```

1 if  $G$  present different DPort attribute regarding the target nodes
   then
2   |  $C == \text{Class2}$ ;
3 else if  $G$  present different DPort and Transp attributes regarding the
   target nodes then
4   |  $C == \text{Class3}$ ;
5 else if  $G$  present different DPort and DstIP attributes regarding the
   target nodes then
6   |  $C == \text{Class4}$ ;
7 else if  $G$  present different values for DPort, DstIP and Transp
   attributes regarding the target nodes then
8   |  $C == \text{Class5}$ ;
9 else
10  |  $C == \text{Class1}$ ;
11 end
12 return  $C$ 

```

The automatic pivot classifier algorithm can reclassify a group of PAAMs G . Therefore, when a new message is included in the group, the algorithm process G again, updating the actual classification if the analysis result indicates a more significant degree of connectivity achieved by the adversary.

5.4 APCA Advantages and drawbacks

This section compares APCA with similar classification approaches related to cybersecurity. Since our approach to classifying pivot attacks uses the PAAM received by the CTI framework as input, we consider advantages and drawbacks related to classification and not the necessary steps to extract the input data.

Some authors proposed machine learning techniques to classify malicious traffic [175] and documents [176]. We understand that our solution brings convenient advantages in this specific case because it is lightweight, simple to implement, and does not require training, differently from the machine learning approaches that requires large data sets and time to train models.

The classification criteria presented in this thesis provide extra details on different pivot attack detection strategies. Also, enriching the pivoting detection with additional attributes can increase the situation awareness concerning cyber threats. Additionally, the classification criterion provided in this thesis can provide visibility concerning the level of connectivity achieved by the adversary inferring TTP possibilities. However, it is reactive because it depends on observing pivot attack events. Therefore, it is not helpful in preventing or predicting future attacks.

5.5 Offensive and defensive pivot metrics

In order to measure the success when carrying out some activity, it is essential to identify the objectives based on standards and efficiency metrics concerning what is sought to be achieved. Therefore, this classification proposal is relevant to supporting metrics on both the pivot attacks' defensive and offensive aspects.

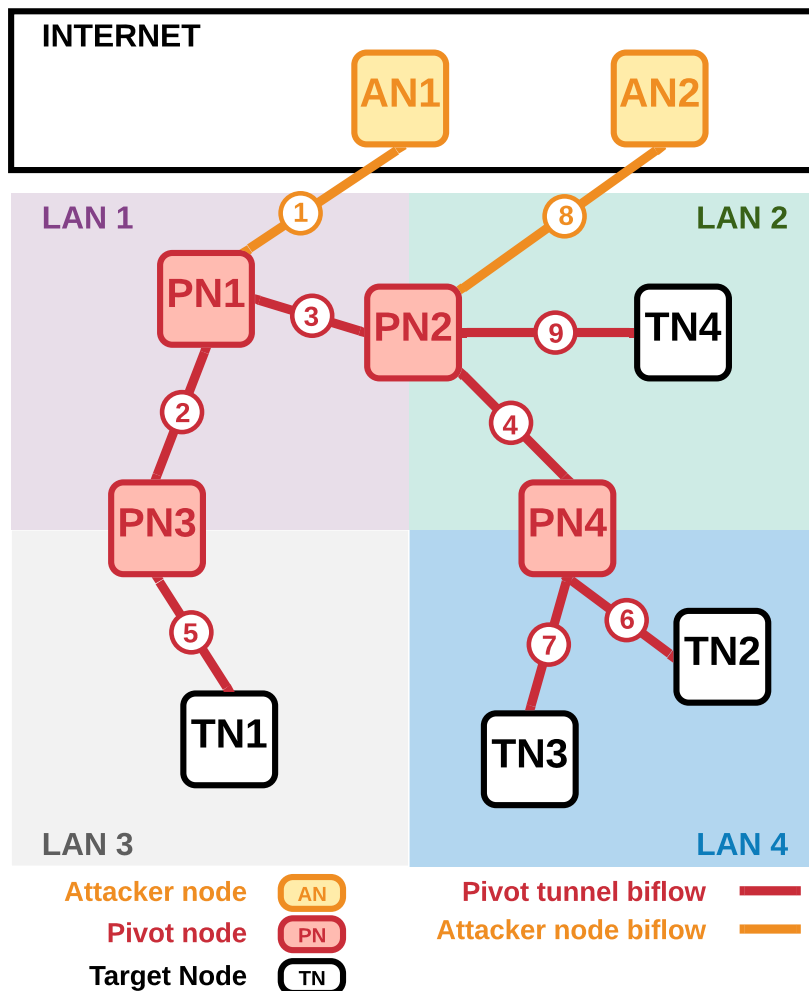


Figure 5.5: Pivoting offensive and defensive capability metrics example

5.5. OFFENSIVE AND DEFENSIVE PIVOT METRICS

The pivot attack classification can provide requirements regarding the necessary connectivity to achieve an objective. For example, an adversary simulation assessment can define a minimum efficiency standard for the aggressors based on the pivoting classification. Additionally, it can be used to organize and measure the degree of penetration and persistence in the target network. Figure 5.5 illustrates an enterprise network segmented into four Local Area Networks (LAN 1, 2, 3 and 4). Suppose the adversary achieved access (Foothold) into two different hosts named Pivot Node 1 (PN1) and Pivot Node 2 (PN2). The PN1 is placed at LAN 1, and number 1 represents the biflow connecting it to the Attacker Node (AN1) on the internet. The second access achieved by the adversary is represented by number 8, which corresponds to a biflow that connects the Attacker Node 2 (AN2) located on the internet to the Pivot Node 2 (PN2), which can access the LAN1 and LAN 2. T1, T2, T3 and T4 represent the target nodes, which correspond to assets that contain sensitive information the adversary intends to access and exfiltrate to AN1 or AN2. Since the initial access and the targets typically are located in different network segments, it is necessary to create pivot tunnels using other devices (pivot nodes) to connect AN1 and AN2 to the targets. For example, the host PN3 presents connectivity with PN1 and can access a specific service in T1, a target node. Therefore, to access T1 at LAN 3, the adversary creates a pivot tunnel using PN1 and PN3. Numbers 1, 2 and 5 represent the biflows part of the pivot attack to exfiltrate T1 sensitive information. Suppose the adversary's objective is to access a single

service provided by T1. In this case, a pivot attack Class I is sufficient to succeed. Regarding targets T2 and T3, consider that a web server provides the information of interest in T2 and a mail server in T3. Both targeted services are supported by the TCP protocol and are located at LAN 4. The pivot nodes PN1, PN2 and PN4 can be used to create a pivot tunnel between AN1 and the targets (T2 and T3) to overcome connectivity issues. Another option is to create a pivot tunnel using PN2 and PN4 from AN2 to access the targets T2 and T3. However, a Pivot Class I is insufficient to achieve data exfiltration between AN1 or AN2 and the targets T2 and T3. The host PN4 must access different hosts in different ports that use the same network-level protocol. According to the pivot attack classification proposed in this thesis, this scenario requires at least a pivot Class IV to be successful.

Regarding pivoting using TN4, two possibilities are feasible. The first option is using AN1 to create a tunnel using PN1 and PN2 (numbers 1, 3 and 9) or using PN2 to forward the traffic between AN2 and TN4 (numbers 8 and 9). For completeness, suppose the attacker needs to access different services in TN4. In this case, a Class II pivot provides the minimum connectivity to achieve success for the adversary.

Another possibility regarding the offensive aspect of pivoting is related to the adversary's presence within the target network. Eventually, the defence identifies the adversary's actions, and the malicious access to the network is lost. Therefore, the attacker typically creates backup access to guarantee the presence; consequently, as significant is the number of access nodes available,

as resilient is the attacker's presence in the target network. For instance, the attacker can define a minimum efficiency standard of at least two different accesses to the target network to decrease the chances of losing access within the target network.

Defensive pivot attack metrics can be helpful to support defence requirements regarding pivoting prevention and network segmentation. In other words, a pivot attack classification can provide tangible security requirements to reduce the attack surface and support network segmentation criteria based on the connectivity achievable by the opponent in specific scenarios. For example, using Figure 5.5 as a reference, assume LAN 4 is physically segmented from LAN 1 and 3. In this case, the defensive plan assumes that pivot attacks will originate from LAN 2. Therefore, AN1 cannot be used regarding pivot attacks to access TN2, TN3 and TN4. The restrictions imposed by the segmentation consequently result in cost reduction, permitting the concentration of defensive resources and efforts.

5.6 Summary

In this chapter, we propose a pivot attack classification criteria based on the connectivity achieved by the attacker. Section 5.1 presents our classification criteria and points out the issues with the binary classification adopted by security products that, in our opinion, are limited in terms of granularity to express complex pivoting scenarios and techniques. Next, we introduce two pivot attack Semantic Network Models (SNM) in Section 5.2 to complement

the pivoting classification criteria and provide a better understanding of the attack. Section 5.3 presents the automatic pivot classifier algorithm, which was created to classify the pivot attacks according to the criteria proposed in this thesis based on the pivot attack alert messages (PAAM) generated by APIVADS agents. Finally, Section 5.5 presents supporting metrics on both the pivot attacks' defensive and offensive aspects.

CHAPTER 6

RESULTS AND DISCUSSION

This chapter shows the achieved results of APIVADS concerning the set of experiments described in Chapter 3 and provides a critical evaluation of the proposed detection scheme with other approaches existing in the literature.

6.1 Virtual network experiment results

Our initial objective was to find an adequate parameter combination regarding detection metrics and to spend as few computational resources as possible. First, however, it is necessary to find equilibrium among parameters to ensure the proper functioning of the algorithms. For example, a small value of T_w imposes a temporal limit to collecting traffic. Moreover, suppose we combine it with a considerable L value more significant than the number of packets perceived within the biflow. In that case, the detection will not happen due to the lack of packet samples. Our approach to determining the best parameters combination was based on the pivot tunnel's amount of PPS (Packets per Second). Let P_{tot} be the total traffic imposed on the host, which affects the time to process the algorithm (T_p) and can cause the malfunction. Moreover, P_{piv} the traffic within the pivot tunnel is used as a traffic pattern

6.1. VIRTUAL NETWORK EXPERIMENT RESULTS

reference in the experiments to identify ideal parameters to detect the pivot attack. During the initial experiments, we imposed a 10 PPS to P_{piv} because it corresponds to a typical Command and Control stage when the attacker sends and receives terminal commands to the target.

The T_w and L parameters strongly correlate with P_{tot} because the unbalance between the former variables can impose restrictions on E based on the required time to execute APIVADS algorithms. We assume that the minimum value of P_{piv} to collect an adequate number of packets must be greater or equal to the result of the division of L by T_w . To increase our chances of achieving detection in the first third of the time window, we defined that the excellent value of P_{piv} can be calculated by the division of L by T_w and multiplied by 3 (Condition 1). Therefore, T_p must be small than the computation of T_w divided by E to achieve near real-time detection in every execution (Condition 2). Finally, E must be bigger than T_p and smaller than T_w . It is necessary because the algorithms must process data before the subsequent execution to avoid malfunctions generated by processing power exhaustion (Condition 3). Those assumptions and conditions resulted in Equation 6.1, coined as Pivot Balance Equation (PBE).

$$\left\{ \begin{array}{l} (1) \quad 3 \times \left(\frac{L}{T_w} \right) \leq P_{piv} \\ (2) \quad \frac{T_w}{E} > T_p \\ (3) \quad T_w > E > T_p \end{array} \right. \quad (6.1)$$

Data reduction parameters were defined to discard biflows incompatible with the pivot attack traffic pattern. Therefore, we defined pD_f as 4 seconds to discard ephemeral biflows and pD_t as 0.01 seconds due to the virtual environment being free of intrinsic network delays. pD_s was defined as 1 second because some terminal commands may take time to generate output and a pR of 5 consecutive packets observed within the same flow. We defined a fixed value of E as 5 seconds, L equal to 200 packets and an average of 10 PPS of pivot traffic for this set of experiments. The selection of initial parameters was based on observation of a real pivot attack pattern to achieve detection.

With the initial parameters defined, we experimented to observe the impact of detection metrics with T_w variations. Additionally, this experiment was used to verify the possibility of addressing near real-time detection. Figure 6.1 presents the experiment results graphically. The y-axis indicates the achieved detection metric rate values, and the x-axis corresponds to the values of the T_w parameter used in the experiment. Detailed values of the experiments are presented in Table 6.1.

According to PBE computation, a fair value of T_w in this experiment must be greater or equal to 1 minute to increase the chances of detection. Therefore, we expected degradation of TPR and DA with values below 1 minute, and our experiments confirmed it. It becomes clear that the L parameter must be compatible with the number of packets collected in a time window. Otherwise, the detection algorithms will disregard the biflow until

6.1. VIRTUAL NETWORK EXPERIMENT RESULTS

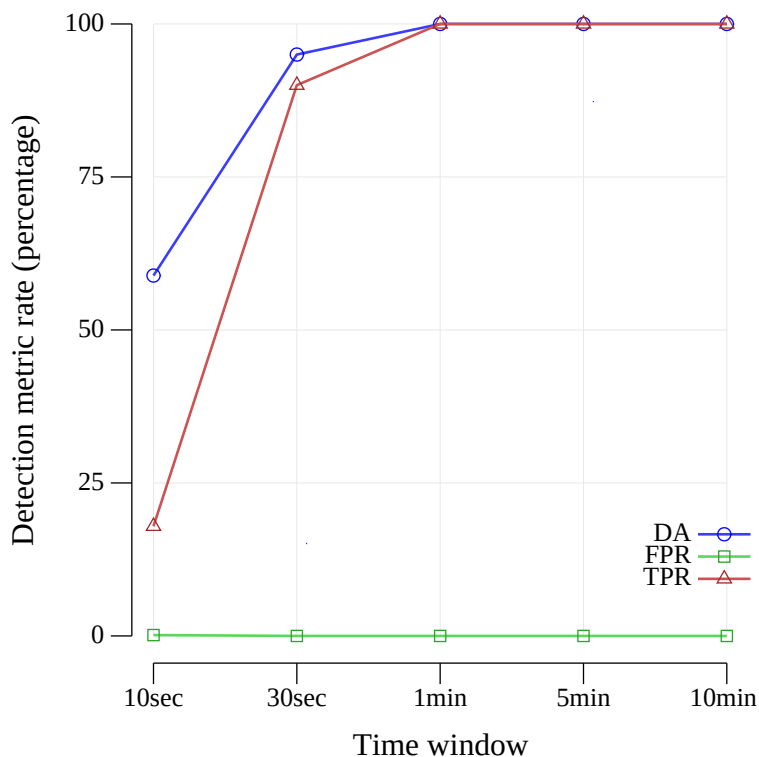


Figure 6.1: Experiments result in the function of T_w

Table 6.1: Detailed experiments result in function of T_w

Time window	TPR	FPR	DA
10 seconds	17.92%	0.14%	58.89%
30 seconds	90.00%	0.00%	95.00%
1 minute	100.00%	0.00%	100.00%
5 minutes	100.00%	0.00%	100.00%
10 minutes	100.00%	0.00%	100.00%

the number of perceived packets is bigger or equal to L . The lack of packets sample will be reflected in all detection metrics, especially regarding FN. A more accurate result is expected as more significant the number of packets

perceived within a biflow while respecting the Equation 6.1 conditions.

Because hosts used in this experiment have sufficient resources to process the amount of traffic collected between executions, APIVADS agents successfully detected all pivot attacks when T_w was defined with values superior to 1 minute. It indicates that PBE is an adequate reference to define APIVADS parameters in function of the P_{piv} we intend to detect. Additionally, even in experiments with restricted T_w values, the FPR results were almost insignificant. It occurs because the duration filter algorithm only selects biflows with similar duration time, which is uncommon between unrelated traffic.

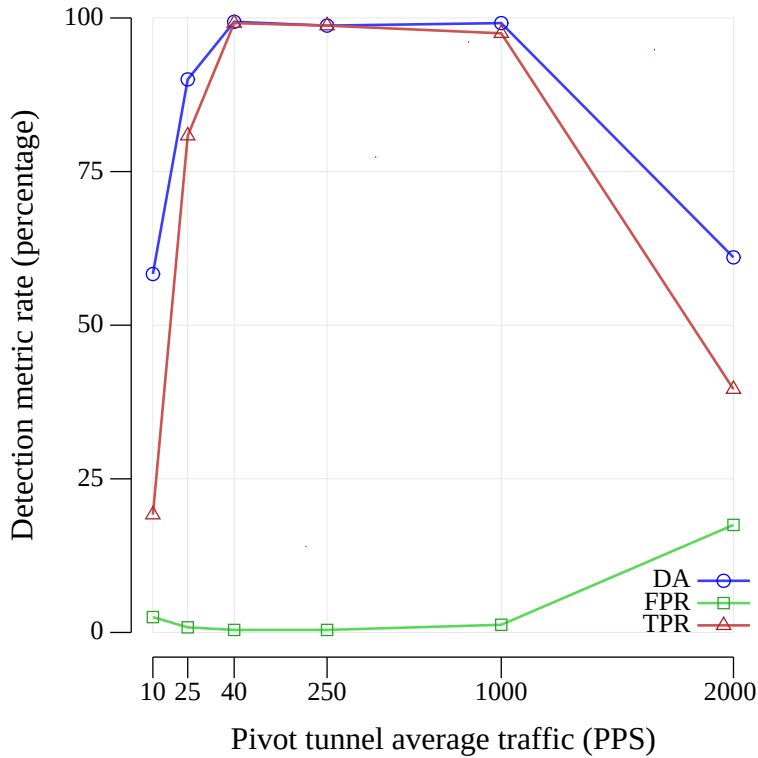
To compare results from our APIVADS testbed with other approaches, we have created Table 6.5. It can be noted that Apruzzese et al. [15] have stated some results when using T_w defined as 60 minutes without intentional propagation delays, which can be compared with our experiment. However, the authors in [15] reported an Accuracy of 100% with the cited parameters without providing any other metric for comparison purposes. With APIVADS, we have achieved the same result with T_w greater or equal to 60 seconds, as shown in Figure 6.1. Unlike the APIVADS host-based approach, the detection strategy proposed by [15] uses a network-based approach that does not address the performance challenges imposed by the near real-time detection. Additionally, [15] work is limited to internal network pivot attacks, underperforming the detection performance and capabilities of APIVADS since most real-life pivot attacks originate from the internet.

Husak et al. [16] can address external network pivot attacks, which would

provide a good source of comparison with APIVADS results. However, the authors stated a high false positives value of 99.99% because the detection algorithm could not differentiate between common protocol traffic patterns and pivot attacks. Therefore, the authors applied the Principal Component Analysis (PCA) machine-learning algorithm to infer the true pivoting features providing relationships among groups of attributes. However, the authors do not present results that can be directly compared with our approach. Regarding PCA, [16] was limited to SSH traffic, and the study did not provide details about the implementation making it challenging to compare the algorithms' performance accurately. Further details concerning [16] results will be provided in Section 6.2.

In the next set of experiments, we gradually increased the P_{piv} PPS to stress APIVADS data processing and verify the impact of detection metric rates. We defined L as 200 packets, a fixed T_w value of 15 seconds and E equal to 5 seconds. According to PBE, this parameter combination requires a minimum P_{piv} of 40 PPS to achieve high detection rates.

Figure 6.2 shows that the detection rates improve as P_{piv} PPS increases. However, while the PBE conditions were respected, we observed excellent detection metric results. Although, with the gradual increase of traffic, the APIVADS agent could not execute the algorithms before a new detection routine starts when dealing with more than 1000 PPS. This behaviour was expected because, eventually, the host will not have resources available to execute the APIVADS algorithms every 5 seconds as defined by E . To avoid

Figure 6.2: P_{piv} influence in detection results

this scenario, we must set E with a value bigger than T_p and less than T_w . Table 6.2 provide the detailed results represented in Figure 6.2. The resiliency of the algorithms regarding FPR was confirmed when respecting the PBE conditions. It achieved the worst-case scenario of 1.25% of FPR with 1000 PPS. During the experiment with P_{piv} values of 10 and 25, the amount of traffic was insufficient to feed the algorithm in the defined time window, resulting in insufficient data sample error, confirming the adequacy of PBE again. Finally, as big is the PPS within P_{piv} as fast the detection will occur while respecting the PBE.

Figure 6.3 represents P_{piv} traffic in the function of time. Ambar bars over

6.1. VIRTUAL NETWORK EXPERIMENT RESULTS

Table 6.2: Detailed experiments result in the function of PPS

P_{piv}	TPR	FPR	DA
10 PPS	19.16%	2.50%	58.33%
25 PPS	80.83%	0.83%	90.00%
40 PPS	99.16%	0.41%	99.37%
250 PPS	98.75%	0.41%	98.75%
1000 PPS	97.50%	1.25%	99.16%
2000 PPS	39.58%	17.50%	61.04%

the x-axis illustrate when detection occurs within the time window, while the red bars indicate a new time window ($T_w1, T_w2 \dots T_wn$). The blue line corresponds to the P_{piv} perceived within the pivot tunnel.

Our APIVADS implementation updates and creates new biflows as it perceives new packets in a time window. When a new time window begins, the collected data is discarded. Moreover, we reduce the computational power to execute the detection algorithms focusing on the recent data. The parameters used in the set of experiments illustrated by Figure 6.3 are the same as the previous, except for T_w , which was set as 30 seconds, L equal to 150 packets, E as 5 seconds, and an average P_{piv} of 10 PPS. Additionally, the T_p value was not greater than E , consequently not affecting the APIVADS data processing performance. In the second time window (T_w2), we can observe that APIVADS could not identify the ongoing pivot attack because the minimum requirement of 150 packet samples imposed by L was not reached in the time window. We intentionally disrespect the PBE first condition in this

experiment, which demands a P_{piv} of 15 PPS. Therefore, to reduce FN incidence due to lack of packet samples is necessary to increase T_w or decrease L observing PBE conditions. This flexibility is of interest when considering different types of pivot attacks in the pivot traffic volume and frequency function.

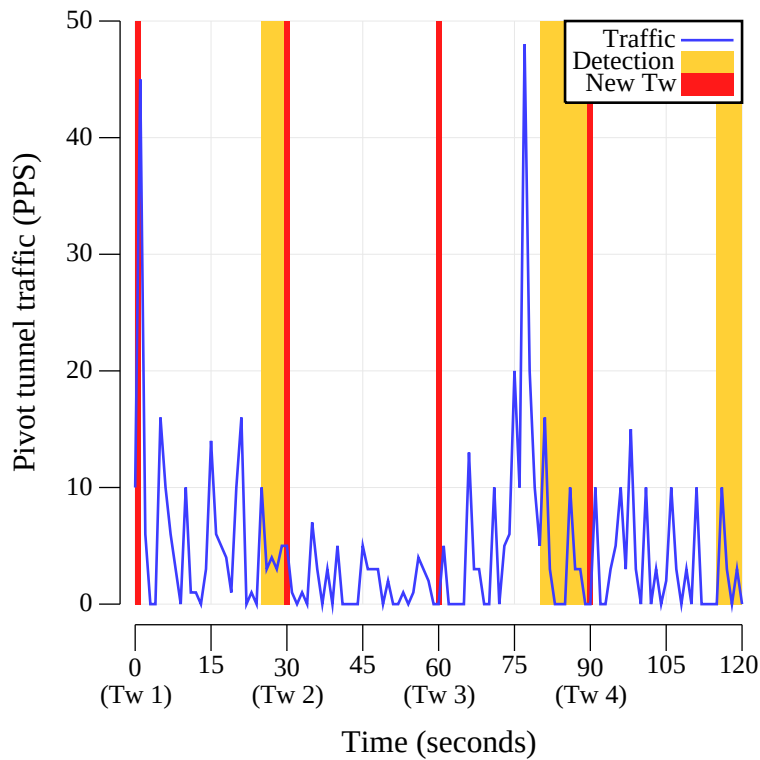


Figure 6.3: Detection of pivot attacks with T_w equal to 30 seconds

New execution of the same experiment using the same traffic was conducted to compare results with the previous experiment. However, this time we respected the PBE to increase APIVADS detection chances. We set T_w as 60 seconds and decreased L to 100 packets. According to PBE compu-

6.1. VIRTUAL NETWORK EXPERIMENT RESULTS

tation, this combination of parameters requires a P_{piv} of 5 PPS. Figure 6.4 presents an entirely different detection result achieved when respecting PBE. It indicates that the combination of balanced parameters can be helpful to adapt the detection mechanism regarding P_{piv} PPS of interest, improving the detection results. Additionally, the capability to sense specific variations regarding P_{piv} can be used to infer APT attack stages, which will be discussed in detail in actual network experiments.

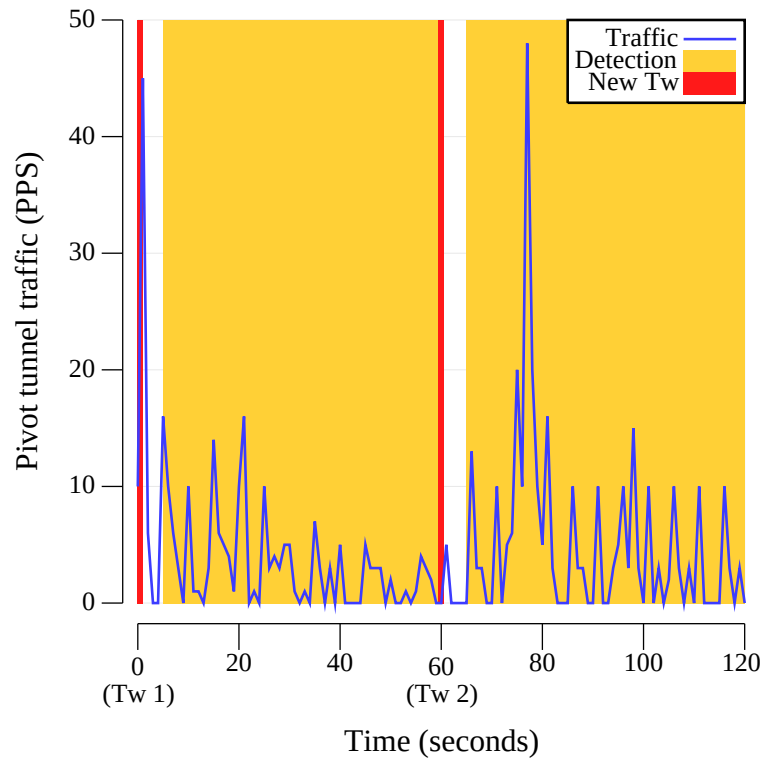


Figure 6.4: Detection of pivot attacks over the internet with T_w equal to 60 seconds

6.2 Real network experiment results

Moving towards real network experiments described in Subsection 3.4.3, we aimed to validate if our implementation can identify pivot attacks over the internet. We installed an APIVADS agent in four hosts in different countries to conduct the experiments. Initially, APIVADS was not detecting the pivot attack with the same parameters used in the virtual environment experiments. As expected, we had to adjust APIVADS data reduction parameters to the new environment in a real network with typical connectivity issues. Our approach to discovering the parameters that must be changed was to observe the pivot tunnel biflows' characteristics. We recognised that we could not achieve detection due to the restricted value of 0.01 pD_t (see Table 4.1) used in the virtual experiments. When dealing with real network scenarios, the computation of D_t between biflows part of a pivot tunnel presented an average of 0.08 seconds. This difference of D_t values observed from virtual to real networks was caused by latency. Therefore, increasing pD_t to 0.1 seconds was sufficient not to discard biflows which are part of the pivot attack over the internet.

Besides details already described in Subsection 3.4.3, we defined T_w as 60 seconds, L equal to 50 packets, and a P_{piv} of 2.5 PPS. With APIVADS parameters adjusted to the real environment, we created a pivot tunnel with two jumps. It was observed that APIVADS could identify the ongoing pivot attack with excellent detection metrics rates in both pivot nodes, comparable

6.2. REAL NETWORK EXPERIMENT RESULTS

to the results conducted in the virtual environment when respecting PBE. The detailed detection metrics achieved with this experiment are illustrated in Table 6.3. The detection metrics' high rates achieved with this experiment validate APIVADS regarding detecting pivot attacks without restrictions to the local network only.

Table 6.3: Real networks experiments detailed results

Host	TPR	FPR	DA
Pivot node 1	98.33%	0.83%	98.75%
Pivot node 2	99.16%	0.41%	99.37%

Next, motivated by the unacceptable rate of FP stated by [16] when dealing with BitTorrent and other p2p protocols, it was included in the regular pivot nodes traffic already presented in Subsection 3.4.3 during the following experiments. Authors in [16] observed a high rate of FP caused by the BitTorrent protocol behaviour regarding frequent connections initiation and reception with a slight time difference.

To address this detection issue, we included in the APIVADS flow attributes structure the number of total bytes observed within a biflow. Our strategy to differentiate biflows related to BitTorrent traffic from biflows part of a pivot attack is based on the computation of the total bytes ratio between them. Besides the other data reduction criteria, we observed that biflows part of the same pivot tunnel tends to have a similar number of total bytes trans-

ferred between endpoints. Therefore, we created the pN parameter to define an acceptable ratio limit. Including this condition in the duration filter algorithm proved efficient in discarding BitTorrent biflows unrelated to pivot attacks.

We set T_w as 60 seconds, L to 100 packets and E as 5 seconds. The average T_p observed was near 0.5 seconds to process approximately 200 biflows, complying with PBE. We used the SSH protocol to create a pivot tunnel with a P_{piv} of 5 PPS. Detection metric results are presented in Table 6.4.

Table 6.4: BitTorrent protocol experiment detection metric rates

Host	TPR	FPR	DA
Pivot node 1	99.58%	2.08%	98.75%
Pivot node 2	98.83%	1.59%	98.32%

According to the detection rates achieved in previous work, table 6.5 helps to verify that our method outperformed the overall results of studies such as [16] in a real network environment. Nevertheless, our results are also comparable with the study conducted by [15]. In contrast, our study offers host-based detection alongside other features, as discussed earlier.

APIVADS results presented in Table 6.5 are composed of pivot node 1 and 2 detection metrics average when the BitTorrent traffic was included in the real network experiment. Regarding [148], the best results among different classifiers have been achieved with the stand-alone LogitBoost classifier (LB).

6.2. REAL NETWORK EXPERIMENT RESULTS

Table 6.5: Comparison with other detection algorithms

Detection approach	TPR	FPR	DA
APIVADS	99.17%	1.87%	98.65%
Husak et al. [16]	53.84%	4.51%	91.78%
Bai et al. [148]	-	-	99.98%
Apruzzese et al. [15]	100%	0.00%	100%

With regards to the missing values in Table 6.5, the authors in [148] stated the following detection metrics: Precision (99.87%), Recall (99.47%) and F_1 (0.992). Therefore, we can estimate slightly better results than our approach based on the provided metrics despite not having the exact metric values to calculate FPR and TPR values.

As already stated, authors in [16] could not provide an efficient detection algorithm when exposed to protocols that present similar patterns to pivoting, hence achieving an FPR of 99.99%. However, for completeness, we included Table 6.5 the PCA experiment metrics results that disregard other protocols different from SSH.

Unlike other approaches, APIVADS consider near real-time detection. Additionally, it does not present protocol restrictions such as [16], or is limited to specific operating system events such as [148]. It is also not limited to private networks when compared to [15] as explained by its authors in [16]. Overall, our approach presents high accurate pivot attack detection rates in complex interconnected networks (the internet) and overcomes the previously cited approaches regarding limitations and functionalities.

6.3 APT attack stages inference results

To validate the parameters provided by Table 3.2, we created a pivot attack with the correspondent P_{piv} for each setup using the same scenario described in Subsection 3.4.3. Being able to infer APT attack stages is useful to predict the actual adversary objectives and possible next steps.

Regarding Command & Control attack stage detection, we initially hypothesised that APIVADS could not be practical to address near real-time detection depending on the necessary processing power and storage resources when dealing with large values of T_w . However, because APIVADS does not require any other information than a set of biflows to execute the detection algorithms, once the packets are perceived and aggregated, they can be discarded. We verify a considerable difference between the perceived packet number and the aggregated version of APIVADS biflows. For example, in this experiment, we collected approximately an average of 212.000 packets from pivot node 1 and 205.000 from pivot node 2 in one hour. Due to the constant and effective data reduction strategy adopted by APIVADS, while the experiment was executed, the traffic was gradually transformed into 310 biflows for the pivot node 1 and 260 for the pivot node 2. Therefore, storage and processing power exhaustion tend to be feasible, with most scenarios demanding an acceptable amount of resources. Additionally, if we aim to identify a specific pattern of P_{piv} and provide APIVADS with a restricted set of parameters, we can have an unacceptable value of FN. For instance, based

6.3. APT ATTACK STAGES INFERENCE RESULTS

on Table 3.2 setups, suppose we define the Command & Control set of parameters to detect Data Exfiltration activities supported by a pivot tunnel. If the attacker does not exceed 1000 packets in 5 seconds, the detection will fail due lack of data samples.

To mitigate this drawback, we address the premises defined by Equation 6.1, conditions 2 and 3. Using of large T_w values with a reduced E value will result in a fast detection while PBE is respected. The selection of T_w and E can be made dynamically based on the computation of T_p . We plan an automatic selection of predefined parameters based on BP total packet number for future work. This mechanism will allow APIVADS to adapt the detection parameters based on the observed P_{piv} of the candidate BP .

Figure 6.5 presents a real environment experiment that intends to observe the sensibility of P_{piv} variations regarding frequency and volume to infer APT attack stages. Initially, in the former minute time window, the traffic imposed was related to simple shell commands sent every 5 seconds, equivalent to 2.5 PPS average, achieving near 25 seconds. We exfiltrated a file using the pivot tunnel for approximately 40 seconds in the second minute. In the next 20 seconds, in the second minute, we reduced the traffic to shelling commands like the one generated in the first minute. During the file upload, the average PPS raised to 2,464.81. In Figure 6.5 we can see that the traffic variation regarding frequency and volume is easy to identify. It confirms that observing P_{piv} can be used to achieve sensibility regarding the amount of traffic of interest and infer changes in TTP and APT attack stages. Therefore, a high

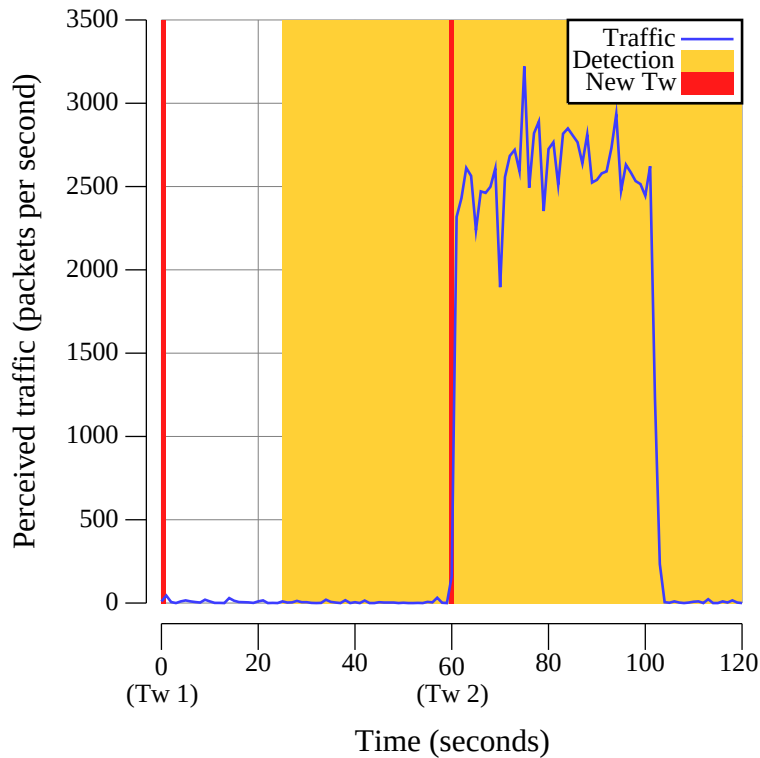


Figure 6.5: Change of behaviour detection based on pivot traffic variation

T_w with reduced L is adequate to detect shell commands or beacons, while small T_w values with large values of L are suitable to detect data exfiltration. Brute force attacks or other malicious activities that generate a large amount of traffic suggest a change of APT attack stages. Moreover, alternating between different APIVADS parameters setups over algorithms execution can be useful to identify different pivot tunnel traffic and consequently identify different APT attack stages in the same time window. However, increasing the algorithms execution frequency will increase the required processing power. Finally, based on the presented results, we demonstrate that APIVADS can consistently identify APT attack stage changes by monitoring the pivot tun-

nel traffic variations regarding frequency and volume.

6.4 Evasive pivot techniques experiments results

As described in Subsection 3.4.4, the T_w , L and D_t parameters have a direct influence on the results. They must be adjusted to classify evasive pivot attacks weaponised with intentional propagation delays. We observed that the cited parameters must be balanced with the delay size applied by the opponent and the amount of P_{piv} PPS perceived.

As already stated, the number of packets within a pivot tunnel observed in a time window must be greater than L to achieve detection. Therefore, setting a small T_w value with a low P_{piv} PPS can lead to FN results, which is aggravated when facing pivots scenarios that apply intentional delays.

Intentional delays Z could affect the detection mechanism if the minimum number of packets is not perceived in a time window. To minimise the packet delay effects regarding the Duration filter algorithm, we set pD_t and pD_s parameters equal to or bigger than the applied delay to achieve compatibility with the worst-case scenario and avoid false negative results. Since Z impacts all BP four flows, T_w and L must be compensated. We adapted PBE (See Subsection 3.4.1) to increase the proportion of T_w in the function of Z , resulting in the following variation of Equation 6.1 to address intentional delay techniques:

$$\left\{ \begin{array}{l}
 (1) \ 3 \times \left(\frac{L + (L \times 0.1 \times Z)}{T_w - (4 \times Z)} \right) \leq P_{piv} \\
 (2) \ \frac{T_w}{E} > T_p \\
 (3) \ T_w > E > T_p \\
 (4) \ Z < T_w + T_p
 \end{array} \right. \quad (6.2)$$

A fourth condition was included to address intentional delays. The imposed delay to P_{piv} can not be bigger than the sum of T_w and T_p . Otherwise, the number of FN will increase while the algorithm detection routine is executed.

Equation 6.2 was used as a reference in the evasion experiments to improve detection rates and overcome additional classification challenges imposed by intentional delays.

We identified that bigger T_w and L values effectively detect pivot attacks with intentional delays during the preliminary experiments. To conduct the evasive pivot attack detection experiments, we define the following APIVADS parameters: L was set as 400 packets, T_w as 10 minutes, E and Z defined with 10 seconds. To create the pivot tunnel, we used the SSH protocol and imposed a P_{piv} of 5 PPS. Detection results are presented in Table 6.6.

According to the results achieved, we could verify that APIVADS can identify delay-based evasion techniques. However, the detection rates slightly decreased compared to the prior tested pivot attacks. It is because intentional delays impose a bigger T_w due to the necessity of more samples that naturally

Table 6.6: Intentional propagation delays experiment results

Host	TPR	FPR	DA
Pivot node 1	94.58%	2.50%	96.04%
Pivot node 2	98.75%	1.66%	98.54%

take more time to arrive at the pivot node.

APIVADS and the algorithm proposed by Apruzzese et al. [15] are the only pivot attack detection approaches that address intentional propagation delays to the best of our knowledge. Furthermore, Apruzzese et al. stated optimum detection metrics, achieving 100% of recall and precision without mentioning any performance decrease, differently from APIVADS, which presents a slight decrease in performance regarding detection metrics. However, as already stated, the detection strategy proposed by the authors in [15] cannot address pivot attacks with origin on the internet and is restricted regarding specific protocols.

6.5 Algorithms complexity

The detection and classification algorithm's complexity must be compatible with the input data length and available computational processing power to be practical in real scenarios. Therefore, for evaluation purposes, we used the Big-O notation [177]. According to the presented algorithms in Section 4.8, our detection algorithm's complexity is exponential $O(n^2)$. This fact could lead to considerable processing time imposing restrictions on expensive com-

putational demands. However, due to the efficient data reduction achieved by the filters and the algorithm parameters already stated, the complexity is not a problem when facing real scenarios.

The APCA requires some time to process. Therefore, it is essential to identify the impact of the classifier algorithm during the whole processing time compared with the APIVADS Detection Algorithm complexity.

The APCA compares the arrived PAAM with a group of clustered PAAM already received by the CTI framework, presenting a linear complexity $O(n)$. Therefore, the Automatic Pivot Classifier algorithm's processing delay has no relevant impact when compared with APIVADS because the latter presents a greater magnitude of complexity. Finally, since a pivot attack is an outlier event, even in an enterprise network, the amount of data to be processed tends to be insignificant to a linear complexity algorithm regarding processing time.

To confirm our hypothesis that the processing time of APCA does not produce a significant delay when compared with APIVADS processing time, Table 6.7 presents the comparison between the cited algorithms.

Table 6.7: Algorithms usability comparison regarding processing time

Algorithm	10^1	10^2	10^3	10^4	10^5
APCA	545ns	$1\mu s$	$5\mu s$	$56\mu s$	$740\mu s$
APIVADS	$9\mu s$	$747\mu s$	139ms	8s	12min

Both algorithms were submitted to a progressive and similar increase of input data, from 10^1 to 10^5 entries. In the worst-case scenario, all PAAM received by the CTI framework can correspond to a different pivot attack (linear complexity). The number of PAAM messages above 10^2 entries is not a feasible scenario in a real corporate network since pivot attacks tend to be an outlier event. However, for completeness, subsequent tests with larger input values were performed to show that APCA processing time impact is insignificant compared to APIVADS asymptotically speaking.

6.6 Summary

This chapter presents the evaluation of APIVADS when submitted to the experiments described in Chapter 3. Initially, in Subsection 3.4.2, we addressed experiments in a virtual network scenario to validate the implementation and to find a good parameter combination regarding detection metrics. The experiments provided evidence to identify parameter balance conditions that resulted in the Pivot Balance Equation (PBE), which is used to increase the chances of pivot attack detection. It was observed that APIVADS detection algorithms showed resiliency regarding FPR when respecting PBE conditions, achieving the worst-case scenario of 1.25% of FPR.

Moving toward the real network experiment scenarios described in Subsection 3.4.3, the main objective was to validate our APIVADS implementation regarding the detection of pivot attacks over the internet. Due to real-world connectivity challenges such as latency and packet loss, the parameters used

in this experiment were changed to increase the detection metrics. Therefore, an average of 98.74% of TPR, 0.62% of FPR and 99.06% of DA were achieved. When addressing BitTorrent traffic in the tests to identify if APIVADS could differentiate from pivoting traffic, achieving 99.16% of TPR, 1.87% of FPR and 98.64% of DA. Additionally, Table 6.5 provides a comparison with other detection algorithms regarding the detection metrics defined in Section 3.3.

Subsection 3.4.5 present the APT attack stages inference experiment results. We concluded that APIVADS could infer APT attack stages based on observing variations of P_{piv} regarding frequency and volume.

In Section 6.4 we show the results related to the evasive pivot attack technique experiments. It was observed that T_w , L and D_t parameters directly influence the results when the pivot attack is weaponised with intentional propagation delays, requiring changes to address this type of pivoting. Therefore, the PBE equation had to be adjusted, and a fourth condition was included to detect intentional delays according to Equation 6.2 to avoid an increase in FN. The final result of this type of experiment indicates that a bigger T_w is essential to collect more samples and observed a slight decrease in performance when compared with the previous experiments regarding the detection metrics, achieving an average of 96.66% of TPR, 1.08% of FPR and a DA of 97.29%.

Section 6.5 presents the results concerning APIVADS and APCA algorithms complexity. The exponential complexity could lead to significant processing times and constraints on computational requirements. However, we

6.6. SUMMARY

could also find that the efficient data reduction used by APIVADS (section 4.8) makes the amount of biflows to be processed in a host part of a real enterprise network irrelevant. Additionally, comparison tests with large input values were performed to show that APCA processing time impact is insignificant compared to APIVADS asymptotically speaking.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This thesis proposes a new architecture to classify and identify traffic patterns related to pivot attacks. Our detection scheme addresses the problem with a flow-based approach and statistical techniques, using exclusively the information extracted from the packet headers collected in the asset to perform the detection. This way, each device can monitor its traffic and identify patterns that indicate a pivot attack without requiring extra information, which is interesting in terms of scalability. Additionally, the detection scheme can define a set of parameters to sense specific pivoting traffic changes, which helps infer changes in the APT attack stage. Moreover, it is essential to mention that the approach contributes to the situational awareness of a computer network's cybersecurity by identifying the nodes supporting a pivot attack as it occurs, even if it is carried out over the internet.

Additionally, we proposed a pivot attack classification based on the connectivity achieved by an adversary and a pivot attack SNM model that provides additional classification attributes considering the number of network interfaces evolved in the pivot attack. Therefore, an automatic pivot classifier algorithm was created as a proof of concept and is applied to the APIVADS

detection algorithms. The follow-up studies may use the proposed pivot attack classification to improve cyber risk analysis frameworks' accuracy since adversaries widely use pivot attacks in offensive campaigns. However, due to the lack of classification and granularity of pivoting, the risk of this attack is usually not considered in existing risk analysis systems.

Concerning the research objectives achievement, Objective 1 was addressed with the modelling and evaluation of the PAAM to express a pivot attack IoA event and infer pivot classes. Objective 2 was achieved with the research and development of APIVADS, which achieved high detection metric rates during the experiments where our implementation was exposed even facing p2p protocols like BitTorrent that present a similar behaviour to a pivot attack. Objective 3 was addressed with the investigation and modelling of APCA, which can infer the degree of connectivity achieved by the adversary based on perceived IoA provided by PAAM processing to infer pivot classes. Objective 4 was achieved based on the experiment results and critical evaluation of the proposed architecture with other approaches found in the literature. Finally, creating a pivot attack dataset related to different APT attack stages addresses Objective 5.

To the best of our knowledge, the pivot attack detection and classification in scenarios that include p2p protocols was an open problem solved by the proposed architecture. The implementation of APIVADS achieved high detection metric rates during the experiments that it was exposed. Additionally, the architecture provides the necessary scalability and flexibility to

address fast enterprise networks even if the attack originates from the internet.

For future work, some improvements within the architecture could be made. The possibility of automatically selecting predefined parameters based on the perceived pivot tunnel traffic is envisioned. Therefore, an increase in performance concerning detection metrics is expected. Another suggestion is the inclusion of third-party outsourcing. If this is implemented, APIVADS could process the traffic from devices that do not have available processing power (e.g. simple IoT devices) or when installing APIVADS agents is not feasible. In that case, it is essential to model a data exchange scheme to keep the privacy-preserving characteristic of the architecture presented in this thesis.

BIBLIOGRAPHY

- [1] Hyeob Kim, Hyuk jun Kwon, and Kyung-Kyu Kim. Modified cyber kill chain model for multimedia service environments. *Multimedia Tools and Applications*, 78:3153–3170, 2018.
- [2] Brahim I. D. Messaoud, Karim Guennoun, Mohamed Wahbi, and Mohamed Sadik. Advanced persistent threat: New analysis driven by life cycle phases and their challenges. *2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS)*, pages 1–6, 2016.
- [3] Rafael Salema Marques, Haider Al-Khateeb, Gregory Epiphaniou, and Carsten Maple. Apivads: A novel privacy-preserving pivot attack detection scheme based on statistical pattern recognition. *IEEE transactions on information forensics and security*, 17:700–715, 2022.
- [4] Mike Cloppert. Security intelligence: Attacking the cyber kill chain. SANS Computer Forensics Blog, October 2009. <https://www.sans.org/blog/security-intelligence-attacking-the-cyber-kill-chain/>.
- [5] P.V. Sai Charan, P. Mohan Anand, Sandeep K Shukla, Naveen Selvan, and Hrushikesh Chunduri. Dotmug: A threat model for target specific

BIBLIOGRAPHY

- apt attacks—misusing google teachable machine. In *2022 10th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–8, 2022.
- [6] Command Five Pty Ltd. Advanced Persistent Threats: A Decade in Review. https://paper.seebug.org/papers/APT/APT_CyberCriminal_Campagin/2011/C5_APT_ADecadeInReview.pdf, 2011.
- [7] Ming Li, Qiang Li, Guangzhe Xuan, and Dong Guo. Identifying compromised hosts under apt using dns request sequences. *Journal of parallel and distributed computing*, 152:67–78, 2021.
- [8] Kaspersky Lab. APT trends report q2 2020. <https://securelist.com/apt-trends-report-q2-2020/97937/>, November 2020.
- [9] Kaspersky Lab. APT trends report q3 2020. <https://securelist.com/apt-trends-report-q3-2020/99204/>, November 2020.
- [10] Boguslaw Olszewski. Advanced persistent threats as a manifestation of states’ military activity in cyber space. *Scientific Journal of the Military University of Land Forces*, 189(3):57–71, 2018.
- [11] Joseph Sexton, Curtis Storlie, and Joshua Neil. Attack chain detection. *Statistical analysis and data mining*, 8(5-6):353–363, 2015.

BIBLIOGRAPHY

- [12] Thomas D. Wagner, Khaled Mahbub, Esther Palomar, and Ali E. Abdallah. Cyber threat intelligence sharing: Survey and research directions. *Computers & Security*, 87:101589, 2019.
- [13] Rafael Salema Marques, Gregory Epiphaniou, Haider Al-Khateeb, Carsten Maple, Mohammad Hammoudeh, Paulo André Lima De Castro, Ali Dehghantanha, and Kim Kwang Raymond Choo. A flow-based multi-agent data exfiltration detection architecture for ultra-low latency networks. *ACM Trans. Internet Technol.*, 21(4), jul 2021.
- [14] Raúl Vera, Amina F. Shehu, Tooska Dargahi, and Ali Dehghantanha. Cyber defence triage for multimedia data intelligence: Hellsing, desert falcons and lotus blossom apt campaigns as case studies. *International Journal of Multimedia Intelligence and Security*, 3(3):221–243, 2019.
- [15] Giovanni Apruzzese, Fabio Pierazzi, Michele Colajanni, and Mirco Marchetti. Detection and threat prioritization of pivoting attacks in large networks. *IEEE transactions on emerging topics in computing*, 8(2):404–415, 2020.
- [16] Martin Husak, Giovanni Apruzzese, Shanchieh Jay Yang, and Gordon Werner. Towards an efficient detection of pivoting activity. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 980–985. IFIP, 2021.

BIBLIOGRAPHY

- [17] Santiago Quintero-Bonilla and Angel Martín del Rey. A new proposal on the advanced persistent threat: A survey. *Applied sciences*, 10(11):3874–, 2020.
- [18] Ben Collier. The power to structure: exploring social worlds of privacy, technology and power in the tor project. *Information, Communication & Society*, 0(0):1–17, 2020.
- [19] Robert Schmidt, Gregory J Rattray, and Christopher J Fogle. Methods and apparatus for developing cyber defense processes and a cadre of expertise, July 10 2008. US Patent App. 11/947,655.
- [20] Atif Ahmad, Jeb Webb, Kevin C. Desouza, and James Boorman. Strategically-motivated advanced persistent threat: Definition, process, tactics and a disinformation model of counterattack. *Computers & Security*, 86:402–418, 2019.
- [21] Colin Tankard. Advanced persistent threats and how to monitor and deter them. *Network security*, 2011(8):16–19, 2011.
- [22] Stefan Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)*, 3(3):186–205, 2000.
- [23] Peng Zhou, Gongyan Zhou, Dakui Wu, and Minrui Fei. Detecting multi-stage attacks using sequence-to-sequence model. *Computers & Security*, 105:102203, 2021.

BIBLIOGRAPHY

- [24] Ibrahim Ghafir, Mohammad Hammoudeh, Vaclav Prenosil, Liangxiu Han, Robert Hegarty, Khaled Rabie, and Francisco J Aparicio-Navarro. Detection of advanced persistent threat using machine-learning correlation analysis. *Future generation computer systems*, 89:349–359, 2018.
- [25] Ruchika Mehresh and Shambhu Upadhyaya. Surviving advanced persistent threats in a distributed environment – architecture and analysis. *Information systems frontiers*, 17(5):987–995, 2015.
- [26] Dingyu Yan, Feng Liu, and Kun Jia. Modeling an information-based advanced persistent threat attack on the internal network. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2019.
- [27] Tyler Wrightson. *Advanced persistent threat hacking : the art and science of hacking any organization*. McGraw-Hill Education, New York, 1st edition edition, 2015.
- [28] Jason Axelrod. Attacker dwell time: Ransomware’s most important metric. *The American City County*, Sep 30 2020. Copyright - Copyright Penton Media, Inc., Penton Business Media, Inc. Sep 30, 2020; Última atualização em - 2021-09-10.
- [29] Eric. Cole. *Advanced persistent threat understanding the danger and how to protect your organization*. Syngress, Boston, 1st edition edition, 2013.

BIBLIOGRAPHY

- [30] Saranya Chandran, Hrudya P, and Prabaharan Poornachandran. An efficient classification model for detecting advanced persistent threat. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2001–2009, 2015.
- [31] Luan Huy Pham, Massimiliano Albanese, and Benjamin W. Priest. A quantitative framework to model advanced persistent threats. In *ICETE*, 2018.
- [32] Thomas M Chen. Stuxnet, the real start of cyber warfare? [editor’s note]. *IEEE network*, 24(6):2–3, 2010.
- [33] Sean Bodmer. *Reverse deception : organized cyber threat counter-exploitation*. McGraw-Hill, New York, 1st edition edition, 2012.
- [34] Sachin Kumar Sahu, Abhishek Anand, Aseem Sharma, and Nidhi Nautiyal. A review: Outrageous cyber warfare. In *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, pages 70–74, 2016.
- [35] Henry Mwiki, Tooska Dargahi, Ali Dehghantanha, and Kim-Kwang Raymond Choo. Analysis and triage of advanced hacking groups targeting western countries critical national infrastructure: Apt28, red october, and regin. *Advanced Sciences and Technologies for Security Applications*, 2019.

BIBLIOGRAPHY

- [36] Zsolt Bederna and Tamás Szádeczky. Cyber espionage through botnets. *Security Journal*, 33:43–62, 2019.
- [37] Adel Alshamrani, Sowmya Myneni, Ankur Chowdhary, and Dijiang Huang. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications surveys and tutorials*, 21(2):1851–1877, 2019.
- [38] S. Sibi Chakkaravarthy, D. Sangeetha, and V. Vaidehi. A survey on malware analysis and mitigation techniques. *Computer Science Review*, 32:1 – 23, 2019.
- [39] Eric M Hutchins, Michael J Cloppert, and Rohan M Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1:80, 2011.
- [40] Lockheed Martin. The cyber kill chain. (Accessed: Apr. 20, 2021).
- [41] Will Gragido and John Pirc. *Cybercrime and Espionage: An Analysis of Subversive Multi-Vector Threats*. Elsevier Science Technology Books, Saint Louis, 2011.
- [42] Przemek Shem Radzikowski. Cybersecurity: Expanded look at the apt life cycle and mitigation. 2016.
- [43] James R Rutherford and Gregory B White. Using an improved cybersecurity kill chain to develop an improved honey community. In *2016*

BIBLIOGRAPHY

- 49th Hawaii International Conference on System Sciences (HICSS)*, pages 2624–2632. IEEE, 2016.
- [44] P.V. Sai Charan, P. Mohan Anand, and Sandeep K. Shukla. Dmapt: Study of data mining and machine learning techniques in advanced persistent threat attribution and detection. In Ciza Thomas, editor, *Data Mining*, chapter 5. IntechOpen, Rijeka, 2021.
- [45] Mandiant. APT1 Exposing One of China’s Cyber Espionage Units. <https://www.mandiant.com/sites/default/files/2021-09/mandiant-apt1-report.pdf>, 2013.
- [46] Blake D. Bryant and H. Saiedian. A novel kill-chain framework for remote security log analysis with siem software. *Comput. Secur.*, 67:198–210, 2017.
- [47] Wen Zeng and Vasileios Germanos. Modelling hybrid cyber kill chain. In *PNSE@ Petri Nets/ACSD*, pages 143–160, 2019.
- [48] Marcus J. Carey and Jennifer Jin. *Oddvar Moe*, pages 164–168. 2019.
- [49] Radah Tarek, Saadi Chaimae, and Chaoui Habiba. Runtime api signature for fileless malware detection. In Kohei Arai, Supriya Kapoor, and Rahul Bhatia, editors, *Advances in Information and Communication*, pages 645–654, Cham, 2020. Springer International Publishing.
- [50] Mirco Marchetti, Fabio Pierazzi, Michele Colajanni, and Alessandro Guido. Analysis of high volumes of network traffic for advanced per-

BIBLIOGRAPHY

- sistent threat detection. *Computer networks (Amsterdam, Netherlands : 1999)*, 109:127–141, 2016.
- [51] Sadegh M Milajerdi, Rigel Gjomemo, Birhanu Eshete, R Sekar, and V.N Venkatakrishnan. Holmes: Real-time apt detection through correlation of suspicious information flows. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1137–1152. IEEE, 2019.
- [52] Francesco Maria Ferazza. Cyber kill chain, mitre att&ck, and the diamond model: a comparison of cyber intrusion analysis models. 2022.
- [53] MITRE. *Adversarial tactics, techniques and common knowledge*, 2020.
- [54] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz. *The Diamond Model of Intrusion Analysis*. 2013.
- [55] Yujie Fan, Yanfang Ye, and Lifei Chen. Malicious sequential pattern mining for automatic malware detection. *Expert systems with applications*, 52:16–25, 2016.
- [56] Manel Jerbi, Zaineb Chelly Dagdia, Slim Bechikh, and Lamjed Ben Said. On the use of artificial malicious patterns for android malware detection, 2020.
- [57] Khalid Alminshid and Mohd Omar. A framework of apt detection based on packets analysis and host destination. *Iraqi Journal of Science*, 61:215–222, 01 2020.

BIBLIOGRAPHY

- [58] Robert M Clark. *Intelligence collection*. 2014.
- [59] Haleh Shahzad, Abdul Rahman Sattar, and Janahan Skandaraniyam. Dga domain detection using deep learning. In *2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP)*, pages 139–143, 2021.
- [60] Fehmi Jaafar, Gabriela Nicolescu, and Christian Richard. A systematic approach for privilege escalation prevention. In *2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 101–108, 2016.
- [61] Mohammad Ahmad Abu Allawi, Ali Hadi, and Arafat Awajan. Mlde: Multi-layer data exfiltration detection system. In *2015 Fourth International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec)*, pages 107–112, 2015.
- [62] Tracey Caldwell. The eagle has landed: part one. *Computer Fraud Security*, 2015(12):5–9, 2015.
- [63] Faheem Ullah, Matthew Edwards, Rajiv Ramdhany, Ruzanna Chitchyan, M. Ali Babar, and Awais Rashid. Data exfiltration: A review of external attack vectors and countermeasures. *Journal of Network and Computer Applications*, 101:18–54, 2018.

BIBLIOGRAPHY

- [64] Si-Jung Kim, Do-Eun Cho, and Sang-Soo Yeo. Secure model against apt in m-connected scada network. *International Journal of Distributed Sensor Networks*, 10(6):594652, 2014.
- [65] Colin Tankard. New rules for combating new threats. *Computer Fraud Security*, 2014(4):14–16, 2014.
- [66] Rafael Antonello, Stenio Fernandes, Carlos Kamienski, Djamel Sadok, Judith Kelner, István Gódor, Géza Szabó, and Tord Westholm. Deep packet inspection tools and techniques in commodity platforms: Challenges and trends. *Journal of Network and Computer Applications*, 35(6):1863–1878, 2012.
- [67] Michael Finsterbusch, Chris Richter, Eduardo Rocha, Jean-Alexander Muller, and Klaus Hanssgen. A survey of payload-based traffic classification approaches. *IEEE Communications Surveys Tutorials*, 16(2):1135–1156, 2014.
- [68] Tomasz Bujlow, Valentín Carela-Español, and Pere Barlet-Ros. Independent comparison of popular dpi tools for traffic classification. *Computer Networks*, 76:75–89, 2015.
- [69] Neminath Hubballi and Pratibha Khandait. Keyclass: Efficient keyword matching for network traffic classification. *Computer Communications*, 185:79–91, 2022.

BIBLIOGRAPHY

- [70] Britta Hale and Chelsea Komlo. On end-to-end encryption. *Cryptology ePrint Archive*, 2022.
- [71] Aakash Bharadawaj Srinivasan, Hemalatha S , and Ramathmika Ramathmika. A table-based end to end encryption technique without key exchange. *Engineered Science*, 19 (September 2022) In Progress:279–284, 2022.
- [72] Chang Lan, Justine Sherry, Raluca Ada Popa, Sylvia Ratnasamy, and Zhi Liu. Embark: Securely outsourcing middleboxes to the cloud. In *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation*, NSDI'16, page 255–273, USA, 2016. USENIX Association.
- [73] Hao Sun, Jinshu Su, Xiaofeng Wang, Rongmao Chen, Yujing Liu, and Qiaolin Hu. Primal: Cloud-based privacy-preserving malware detection. In *Information Security and Privacy*, volume 10343 of *Lecture Notes in Computer Science*, pages 153–172, Cham, 2017. Springer International Publishing.
- [74] Yu Guo, Cong Wang, and Xiaohua Jia. Enabling secure and dynamic deep packet inspection in outsourced middleboxes. In *Proceedings of the 6th International Workshop on Security in Cloud Computing*, SCC '18, page 49–55, New York, NY, USA, 2018. Association for Computing Machinery.

BIBLIOGRAPHY

- [75] Xingliang Yuan, Xinyu Wang, Jianxiong Lin, and Cong Wang. Privacy-preserving deep packet inspection in outsourced middleboxes. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.
- [76] Liron Schiff and Stefan Schmid. Pri: Privacy preserving inspection of encrypted network traffic. In *2016 IEEE Security and Privacy Workshops (SPW)*, pages 296–303, May 2016.
- [77] Hao Ren, Hongwei Litt, Dongxiao Liu, and Xuemin Sherman Shen. Toward efficient and secure deep packet inspection for outsourced middlebox. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.
- [78] Yuxin Meng, Wenjuan Li, Lam-For Kwok, and Yang Xiang. Towards designing privacy-preserving signature-based ids as a service: A study and practice. In *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, pages 181–188, 2013.
- [79] James Scott. Signature based malware detection is dead. *Institute for Critical Infrastructure Technology*, 2017.
- [80] Chengcheng Xu, Shuhui Chen, Jinshu Su, Siu-Ming Yiu, and Lucas CK Hui. A survey on regular expression matching for deep packet inspection: Applications, algorithms, and hardware platforms. *IEEE Communications Surveys & Tutorials*, 18(4):2991–3029, 2016.

BIBLIOGRAPHY

- [81] Alok Tongaonkar, Ruben Torres, Marios Iliofotou, Ram Keralapura, and Antonio Nucci. Towards self adaptive network traffic classification. *Computer Communications*, 56:35–46, 2015.
- [82] Matteo Casenove. Exfiltrations using polymorphic blending techniques: Analysis and countermeasures. In *2015 7th International Conference on Cyber Conflict: Architectures in Cyberspace*, pages 217–230. NATO CCD COE, 2015.
- [83] Ananth A. Jillepalli, Daniel Conte de Leon, and Jim Alves-Foss. Operational characteristics of modern malware: Pco threats. In *Proceedings of the Fifth Cybersecurity Symposium, CyberSec '18*, New York, NY, USA, 2018. Association for Computing Machinery.
- [84] Shadi A Aljawarneh, Raja A Moftah, and Abdelsalam M Maatuk. Investigations of automatic methods for detecting the polymorphic worms signatures. *Future generation computer systems*, 60:67–77, 2016.
- [85] Zhi Wang, Meilin Qin, Mengqi Chen, Chunfu Jia, and Yong Ma. A learning evasive email-based p2p-like botnet. *China Communications*, 15(2):15–24, 2018.
- [86] Felix Mannhardt, Agnes Koschmider, Nathalie Baracaldo, Matthias Weidlich, and Judith Michael. Privacy-preserving process mining: Differential privacy for event logs. *Business information systems engineering*, 61(5):595–614, 2019.

BIBLIOGRAPHY

- [87] Anna Sperotto, Gregor Schaffrath, Ramin Sadre, Cristian Morariu, Aiko Pras, and Burkhard Stiller. An overview of ip flow-based intrusion detection. *IEEE Communications surveys and tutorials*, 12(3):343–356, 2010.
- [88] Muhammad Fahad Umer, Muhammad Sher, and Yaxin Bi. Flow-based intrusion detection: Techniques and challenges. *Computers Security*, 70:238–254, 2017.
- [89] Markus Ring, Daniel Schlör, Dieter Landes, and Andreas Hotho. Flow-based network traffic generation using generative adversarial networks. *Computers & security*, 82:156–172, 2019.
- [90] Ruidong Chen, Weina Niu, Xiaosong Zhang, Zhongliu Zhuo, and Feng-mao Lv. An effective conversation-based botnet detection method. *Mathematical problems in engineering*, 2017:1–9, 2017.
- [91] Zahra Jadidi, Vallipuram Muthukkumarasamy, Elankayer Sithiraseenan, and Kalvinder Singh. A probabilistic sampling method for efficient flow-based analysis. *Journal of Communications and Networks*, 18(5):818–825, 2016.
- [92] Johan Garcia, Topi Korhonen, Ricky Andersson, and Filip Västlund. Towards video flow classification at a million encrypted flows per second. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 358–365, 2018.

BIBLIOGRAPHY

- [93] Taimur Bakhshi and Bogdan Ghita. Traffic profiling: Evaluating stability in multi-device user environments. In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 731–736. IEEE, 2016.
- [94] Camila F. T. Pontes, Manuela M. C. de Souza, João J. C. Gondim, Matt Bishop, and Marcelo Antonio Marotta. A new method for flow-based network intrusion detection using the inverse potts model. *IEEE Transactions on Network and Service Management*, 18(2):1125–1136, 2021.
- [95] Brian Schmidt, Ala Al-Fuqaha, Ajay Gupta, and Dionysios Kountanis. Optimizing an artificial immune system algorithm in support of flow-based internet traffic classification. *Applied Soft Computing*, 54:1–22, 2017.
- [96] Hao Ren, Hongwei Li, Dongxiao Liu, Guowen Xu, Nan Cheng, and Xuemin Shen. Privacy-preserving efficient verifiable deep packet inspection for cloud-assisted middlebox. *IEEE Transactions on Cloud Computing*, 10(2):1052–1064, 2022.
- [97] Hao Ren, Hongwei Li, Dongxiao Liu, Guowen Xu, and Xuemin Sherman Shen. Enabling secure and versatile packet inspection with probable cause privacy for outsourced middlebox. *IEEE Transactions on Cloud Computing*, pages 1–1, 2021.

BIBLIOGRAPHY

- [98] Pratik Narang, Chittaranjan Hota, and VN Venkatakrishnan. Peer-shark: flow-clustering and conversation-generation for malicious peer-to-peer traffic identification. *EURASIP Journal on Information Security*, 2014(1):1–12, 2014.
- [99] Rick Hofstede, Pavel Celeda, Brian Trammell, Idilio Drago, Ramin Sadre, Anna Sperotto, and Aiko Pras. Flow monitoring explained: From packet capture to data analysis with netflow and ipfix. *IEEE Communications surveys and tutorials*, 16(4):2037–2064, 2014.
- [100] B. Claise. Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational), October 2004.
- [101] B. Claise, B. Trammell, and P. Aitken. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. RFC 7011 (Internet Standard), September 2013.
- [102] Rick Hofstede, Mattijs Jonker, Anna Sperotto, and Aiko Pras. Flow-based web application brute-force attack and compromise detection. *Journal of network and systems management*, 25(4):735–758, 2017.
- [103] B. Trammell and E. Boschi. Bidirectional Flow Export Using IP Flow Information Export (IPFIX). RFC 5103 (Proposed Standard), January 2008.
- [104] Georgi A Ajaeiya, Nareg Adalian, Imad H Elhajj, Ayman Kayssi, and Ali Chehab. Flow-based intrusion detection system for sdn. In *2017*

BIBLIOGRAPHY

- IEEE Symposium on Computers and Communications (ISCC)*, pages 787–793. IEEE, 2017.
- [105] Frank Beer and Ulrich Buhler. Feature selection for flow-based intrusion detection using rough set theory. In *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, pages 617–624. IEEE, 2017.
- [106] Skhumbuzo Zwane, Paul Tarwireyi, and Matthew Adigun. Ensemble learning approach for flow-based intrusion detection system. In *2019 IEEE AFRICON*, pages 1–8. IEEE, 2019.
- [107] Alina Vlăduțu, Dragoș Comăneci, and Ciprian Dobre. Internet traffic classification based on flows’ statistical properties with machine learning: Internet traffic classification based on flows’ statistical properties. *International journal of network management*, 27(3):e1929–, 2017.
- [108] A. Tayal, N. Hubballi, and N. Tripathi. Communication recurrence and similarity detection in network flows. In *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6, Dec 2017.
- [109] G. Pellegrino, Q. Lin, C. Hammerschmidt, and S. Verwer. Learning behavioral fingerprints from netflows using timed automata. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 308–316, May 2017.

BIBLIOGRAPHY

- [110] Wei Wang, Yaoyao Shang, Yongzhong He, Yidong Li, and Jiqiang Liu. Botmark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. *Information Sciences*, 511:284–296, 2020.
- [111] Matheus Bernardes. GTRS - Google Translator Reverse Shell. (Accessed: Jun. 03, 2022).
- [112] Paolo Passeri. Cloud Threats Memo: Preventing the Exploitation of Dropbox as a Command and Control. (Accessed: Jun. 03, 2022).
- [113] Himanshu Sharma. *Hands-on red team tactics : a practical guide to mastering red team operations*. Packt Publishing Ltd, Birmingham, UK, 1st edition edition, 2018.
- [114] Esam Sharafuddin, Nan Jiang, Yu Jin, and Zhi-Li Zhang. Hospital: Host and network system profiler and internet traffic analyzer. In *2010 IEEE Globecom Workshops*, pages 420–424. IEEE, 2010.
- [115] Kuai Xu, Zhi-Li Zhang, and Supratik Bhattacharyya. Internet traffic behavior profiling for network security monitoring. *IEEE/ACM Transactions on Networking*, 16:1241–1252, 2008.
- [116] Priyanka and Mayank Dave. Peerfox: Detecting parasite p2p botnets in their waiting stage. In *2015 International Conference on Signal Processing, Computing and Control (ISPCC)*, pages 350–355. IEEE, 2015.

BIBLIOGRAPHY

- [117] Hirochika Asai, Kensuke Fukuda, Patrice Abry, Pierre Borgnat, and Hiroshi Esaki. Network application profiling with traffic causality graphs. *International Journal of Network Management*, 24(4):289–303, 2014.
- [118] Long Mai and Minh Park. A comparison of clustering algorithms for botnet detection based on network flow. In *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 667–669, 2016.
- [119] A. Vance. Flow based analysis of advanced persistent threats detecting targeted attacks in cloud computing. In *2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology*, pages 173–176, Oct 2014.
- [120] Pieter Burghouwt, Marcel Spruit, and Henk Sips. Detection of botnet command and control traffic by the identification of untrusted destinations. In Jing Tian, Jiwu Jing, and Mudhakar Srivatsa, editors, *International Conference on Security and Privacy in Communication Networks*, pages 174–182, Cham, 2015. Springer International Publishing.
- [121] Milan Čermák, Pavel Čeleda, and Jan Vykopal. Detection of dns traffic anomalies in large networks. In Yvon Kermerrec, editor, *Advances in Communication Networking*, pages 215–226, Cham, 2014. Springer International Publishing.

BIBLIOGRAPHY

- [122] T. Cejka, V. Bartos, M. Svepes, Z. Rosa, and H. Kubatova. Nemea: A framework for network traffic analysis. In *2016 12th International Conference on Network and Service Management (CNSM)*, pages 195–201, Oct 2016.
- [123] Z. Berkay Celik, R. J. Walls, P. McDaniel, and A. Swami. Malware traffic detection using tamper resistant features. In *MILCOM 2015 - 2015 IEEE Military Communications Conference*, pages 330–335, Oct 2015.
- [124] Hui Song, Liang Xie, Sencun Zhu, and Guohong Cao. Sensor node compromise detection: The location perspective. In *Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing, IWCMC '07*, page 242–247, New York, NY, USA, 2007. Association for Computing Machinery.
- [125] Laurens Hellemons, Luuk Hendriks, Rick Hofstede, Anna Sperotto, Ramin Sadre, and Aiko Pras. Sshcure: A flow-based ssh intrusion detection system. In Ramin Sadre, Jiří Novotný, Pavel Čeleda, Martin Waldburger, and Burkhard Stiller, editors, *Dependable Networks and Services*, pages 86–97, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [126] Rick Hofstede, Aiko Pras, Anna Sperotto, and Gabi Dreo Rodosek. Flow-based compromise detection: Lessons learned. *IEEE Security*

BIBLIOGRAPHY

- Privacy*, 16(1):82–89, 2018.
- [127] Kirsty P and Ollie Whitehouse. Indicators of Compromise (IoCs) and Their Role in Attack Defence. Internet-Draft draft-paine-smart-indicators-of-compromise-00, Internet Engineering Task Force, March 2020. Work in Progress.
- [128] Yuma Kurogome, Yuto Otsuki, Yuhei Kawakoya, Makoto Iwamura, Syogo Hayashi, Tatsuya Mori, and Koushik Sen. Eiger: automated ioc generation for accurate and interpretable endpoint malware detection. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 687–701, 2019.
- [129] Yuta Kazato, Yoshihide Nakagawa, and Yuichi Nakatani. Improving maliciousness estimation of indicator of compromise using graph convolutional networks. In *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, pages 1–7, 2020.
- [130] Hyeisun Cho, Seulgi Lee, Nakhyun Kim, Byungik Kim, and Junhyung Park. Method of quantification of cyber threat based on indicator of compromise. In *2018 International Conference on Platform Technology and Service (PlatCon)*, pages 1–6, 2018.
- [131] Antonio Villalón-Huerta, Ismael Ripoll-Ripoll, and Hector Marco-Gisbert. Key requirements for the detection and sharing of behavioral indicators of compromise. *Electronics (Basel)*, 11(416):416–, 2022.

BIBLIOGRAPHY

- [132] Kaspersky Lab. Using indicators of compromise (IOC) and attack (IOA) for Threat Hunting. Kaspersky lab, Jul 31, 2022. [Online].
- [133] CrowdStrike white paper: indicators of attack versus indicators of compromise. CrowdStrike, Apr 29, 2021. [Online].
- [134] Vasileios Mavroeidis and Siri Bromander. Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 91–98, 2017.
- [135] Christopher S. Johnson, Mark Lee Badger, David A. Waltermire, Julie Snyder, and Clem Skorupka. Guide to cyber threat information sharing. Technical report, 2016.
- [136] Adil Atifi and Elias Bou-Harb. On correlating network traffic for cyber threat intelligence: A bloom filter approach. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 384–389, 2017.
- [137] Sonali Chandel, Mengdi Yan, Shaojun Chen, Huan Jiang, and Tian-Yi Ni. Threat intelligence sharing community: A countermeasure against advanced persistent threat. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 353–359, 2019.
- [138] Daegeon Kim, JiYoung Woo, and Huy Kang Kim. “i know what you did before”: General framework for correlation analysis of cyber threat

BIBLIOGRAPHY

- incidents. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 782–787, 2016.
- [139] Marina Danchofsky Ibrishimova. Cyber incident classification: issues and challenges. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 469–477. Springer, 2018.
- [140] Cláudio Martins and Ibéria Medeiros. Generating threat intelligence by classification and association of security events. In *DSN Workshop on Data-Centric Dependability and Security*, 2019.
- [141] Wiem Tounsi and Helmi Rais. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers Security*, 72:212–233, 2018.
- [142] Tianyi Wang and Kam Pui Chow. Automatic tagging of cyber threat intelligence unstructured data using semantics extraction. In *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 197–199, 2019.
- [143] Kris Oosthoek and Christian Doerr. Inside the matrix: Cti frameworks as partial abstractions of complex threats. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 2136–2143, 2021.
- [144] Manisha Parmar and Alberto Domingo. On the use of cyber threat intelligence (cti) in support of developing the commander’s understand-

BIBLIOGRAPHY

- ing of the adversary. In *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*, pages 1–6, 2019.
- [145] Alessandro Greco, Giovanni Pecoraro, Alberto Caponi, and Giuseppe Bianchi. Advanced widespread behavioral probes against lateral movements. *Int J Inf Secur Res*, 6(2):651–659, 2016.
- [146] Ping Chen, Lieven Desmet, and Christophe Huygens. A study on advanced persistent threats. In *Communications and Multimedia Security*, 2014.
- [147] Yu Shi, Xiaolin Chang, Ricardo J Rodríguez, Zhenjiang Zhang, and Kishor S Trivedi. Quantitative security analysis of a dynamic network system under lateral movement-based attacks. *Reliability engineering system safety*, 183:213–225, 2019.
- [148] Tim Bai, Haibo Bian, Mohammad A Salahuddin, Abbas Abou Daya, Noura Limam, and Raouf Boutaba. Rdp-based lateral movement detection using machine learning. *Computer communications*, 165:9–19, 2021.
- [149] Eswar Konduru and Jerry Petree. Udp to tcp bridge, 2011.
- [150] Go Authors. The go programming language. (Accessed: Apr. 20, 2021).
- [151] qBittorrent official website. (Accessed: Apr. 20, 2021).
- [152] GNU Wget. (Accessed: Apr. 20, 2021).

BIBLIOGRAPHY

- [153] Mozilla firefox. (Accessed: Apr. 20, 2021).
- [154] OpenBSD Foundation. OpenSSH. (Accessed: Apr. 20, 2021).
- [155] Canonical Ltd. Ubuntu desktop. (Accessed: Apr. 20, 2021).
- [156] Combs Gerald. Tshark—dump and analyze network traffic. (Accessed: Apr. 20, 2021).
- [157] Guy Harris and Michael Richardson. PCAP Capture File Format. Internet-Draft draft-gharris-opsawg-pcap-02, Internet Engineering Task Force, June 2021. Work in Progress.
- [158] Vinit Jain. *Wireshark Fundamentals: A Network Engineer's Handbook to Analyzing Network Traffic*. Apress L. P, Berkeley, CA, 2022.
- [159] Chris Sanders. *Practical packet analysis : using Wireshark to solve real-world network problems*. No Starch Press, San Francisco, third edition, 2017.
- [160] Pradeep Chathuranga Weeraddana, Georgios Athanasiou, Carlo Fischione, and John S Baras. Per-se privacy preserving solution methods based on optimization. In *Proceedings of the 52nd IEEE Conference on Decision and Control (CDC)*, pages 206–211, 2013.
- [161] Iman Vakilinia, Sui Cheung, and Shamik Sengupta. Sharing susceptible passwords as cyber threat intelligence feed. In *MILCOM 2018 - 2018*

BIBLIOGRAPHY

- IEEE Military Communications Conference (MILCOM)*, pages 1–6, 2018.
- [162] Jay Thom, Yash Shah, and Shamik Sengupta. Correlation of cyber threat intelligence data across global honeypots. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0766–0772, 2021.
- [163] Nash Haynes. *Cyber Crime E-Book*. ED-TECH PRESS, 1 edition, January 2018.
- [164] Marc Chatel. Classical versus Transparent IP Proxies. RFC 1919, March 1996.
- [165] Perry B Gentry. What is a vpn? *Information Security Technical Report*, 6(1):15–22, 2001.
- [166] Setting up a test environment for VPN Pivoting with Metasploit Pro. Rapid7, Jul 28, 2021. [Online].
- [167] How VPN Pivoting Works (with source code). HelpSystems, Jul 28, 2021. [Online].
- [168] VPN Pivoting. Infosec, Jul 28, 2021. [Online].
- [169] VPN Pivoting. EyesOpen IT Security, Jul 28, 2021. [Online].
- [170] Combs Gerald. sshuttle: where transparent proxy meets VPN meets ssh. (Accessed: Apr. 29, 2021).

BIBLIOGRAPHY

- [171] Jaime Pillora. A fast TCP/UDP tunnel over HTTP. Github, Aug 05, 2021. [Online].
- [172] H Zimmermann. Osi reference model - the iso model of architecture for open systems interconnection. *IEEE transactions on communications*, 28(4):425–432, 1980.
- [173] Kavita Sahu, F.A. Al-Zahrani, R.K Srivastava, and Rajeev Kumar. Evaluating the impact of prediction techniques: Software reliability perspective. *Computers, Materials and Continua*, 67:1471–1488, 02 2021.
- [174] Abdulaziz Attaallah, Hassan Alsuhabi, Sarita Shukla, Rajeev Kumar, Bineet Gupta, and Prof. Raees Khan. Analyzing the big data security through a unified decision-making approach. *Intelligent Automation Soft Computing*, 32:1071–1088, 01 2022.
- [175] Enrico Bocchi, Luigi Grimaudo, Marco Mellia, Elena Baralis, Sabyasachi Saha, Stanislav Miskovic, Gaspar Modelo-Howard, and Sung-Ju Lee. Magma network behavior classifier for malware traffic. *Computer networks (Amsterdam, Netherlands : 1999)*, 109:142–156, 2016.
- [176] Yuanzhang Li, Xinxin Wang, Zhiwei Shi, Ruyun Zhang, Jingfeng Xue, and Zhi Wang. Boosting training for pdf malware classifier via active

BIBLIOGRAPHY

learning. *International journal of intelligent systems*, 37(4):2803–2821, 2022.

- [177] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

APPENDIX A

DATASET EXAMPLES (SNAPSHOTS)

```
NEW SESSION: 2021/07/15 16:57:40
-----
NEW SESSION: 2021/07/15 16:57:40 OS: linux amd64
NEW SESSION: 2021/07/15 16:57:40 Host: bob
NEW SESSION: 2021/07/15 16:57:40 IPs: [192.168.6.132 ]
NEW SESSION: 2021/07/15 16:57:40 Date: 2021-07-15 16:57:40.190579466 +0100 BST m=+0.031781720
NEW SESSION: 2021/07/15 16:57:40
-----
PIVOT: 2021/07/15 16:58:00 *** PIVOT DETECTED: 0.000413 => {IPv4 tcp 192.168.6.131 50102 192.168.6.132 1979 20692 2.618353 10.0837} {IPv4 tcp 192.168.6.132 45810 192.168.6.134 22 22436 2.61784 10.0836}
PIVOT: 2021/07/15 16:58:00 RECORDED AT FILE: /home/madex/go/src/madex1/pcaps/pivot_00001_20210715165746.pcap
PIVOT: 2021/07/15 16:58:20 *** PIVOT DETECTED: 0.000513 => {IPv4 tcp 192.168.6.132 45810 192.168.6.134 22 54432 2.61784 30.1962} {IPv4 tcp 192.168.6.131 50102 192.168.6.132 1979 50136 2.618353 30.1962}
PIVOT: 2021/07/15 16:58:20 RECORDED AT FILE: /home/madex/go/src/madex1/pcaps/pivot_00001_20210715165746.pcap
PIVOT: 2021/07/15 16:58:20 *** PIVOT DETECTED: 0.000513 => {IPv4 tcp 192.168.6.131 50102 192.168.6.132 1979 50136 2.618353 30.1962} {IPv4 tcp 192.168.6.132 45810 192.168.6.134 22 54432 2.61784 30.1962}
PIVOT: 2021/07/15 16:58:20 RECORDED AT FILE: /home/madex/go/src/madex1/pcaps/pivot_00001_20210715165746.pcap
PIVOT: 2021/07/15 16:58:21 *** PIVOT DETECTED: 0.000513 => {IPv4 tcp 192.168.6.132 45810 192.168.6.134 22 54432 2.61784 30.1962} {IPv4 tcp 192.168.6.131 50102 192.168.6.132 1979 50136 2.618353 30.1962}
PIVOT: 2021/07/15 16:58:21 RECORDED AT FILE: /home/madex/go/src/madex1/pcaps/pivot_00001_20210715165746.pcap
```

Figure A.1: APIVADS PAAM generated events

<< pcaps > shell_commands

Pesquisar em shell_commands




















Nome	Data de modificação	Tipo	Tamanho
 pivot_00001_20210208225649.pcap	08/02/2021 19:57	Wireshark capture file	475 KB
 pivot_00002_20210208225749.pcap	08/02/2021 19:58	Wireshark capture file	367 KB
 pivot_00003_20210208225849.pcap	08/02/2021 19:59	Wireshark capture file	491 KB
 pivot_00004_20210208225949.pcap	08/02/2021 20:00	Wireshark capture file	416 KB
 pivot_00005_20210208230049.pcap	08/02/2021 20:01	Wireshark capture file	698 KB
 pivot_00006_20210208230149.pcap	08/02/2021 20:02	Wireshark capture file	494 KB
 pivot_00007_20210208230249.pcap	08/02/2021 20:03	Wireshark capture file	499 KB
 pivot_00008_20210208230349.pcap	08/02/2021 20:04	Wireshark capture file	327 KB
 pivot_00009_20210208230449.pcap	08/02/2021 20:05	Wireshark capture file	369 KB
 pivot_00010_20210208230549.pcap	08/02/2021 20:06	Wireshark capture file	1.212 KB
 pivot_00011_20210208230649.pcap	08/02/2021 20:07	Wireshark capture file	357 KB
 pivot_00012_20210208230749.pcap	08/02/2021 20:08	Wireshark capture file	497 KB
 pivot_00013_20210208230849.pcap	08/02/2021 20:09	Wireshark capture file	465 KB
 pivot_00014_20210208230949.pcap	08/02/2021 20:10	Wireshark capture file	412 KB
 pivot_00015_20210208231049.pcap	08/02/2021 20:11	Wireshark capture file	464 KB
 pivot_00016_20210208231149.pcap	08/02/2021 20:12	Wireshark capture file	473 KB
 pivot_00017_20210208231249.pcap	08/02/2021 20:13	Wireshark capture file	1.092 KB
 pivot_00018_20210208231349.pcap	08/02/2021 20:14	Wireshark capture file	656 KB
 pivot_00019_20210208231449.pcap	08/02/2021 20:15	Wireshark capture file	438 KB

Figure A.2: Pivot attack in the Command & Control attack stage pcap files dataset

Wireshark · Conversations · pivot_00001_20210209171525.pcap

Ethernet		IPv4 · 17		IPv6 · 10		TCP · 333		UDP · 9					
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
2.31.248.86	57189	185.52.0.113	1979	187	14 k	94	5264	93	9300	0.000000	59.5363	707	1249
2.31.248.86	57182	185.52.0.113	1979	265	131 k	79	4532	186	126 k	0.440206	58.4651	620	17 k
172.18.0.4	52620	172.18.0.3	27017	140	19 k	84	7448	56	11 k	0.683834	55.0103	1083	1742
2.31.248.86	57171	185.52.0.113	1979	98	34 k	45	3204	53	31 k	4.379969	54.3629	471	4667
185.52.0.113	37080	168.235.94.254	1979	208	39 k	108	7668	100	32 k	4.380078	54.1749	1132	4742
172.18.0.4	52512	172.18.0.3	27017	36	6720	24	2256	12	4464	5.599851	50.0401	360	713
172.18.0.4	52440	172.18.0.3	27017	36	6720	24	2256	12	4464	2.153817	50.0032	360	714
185.52.0.113	42248	2.31.248.86	8080	6	456	6	456	0	0	9.483309	31.0734	117	0
185.52.0.113	42230	2.31.248.86	8080	6	456	6	456	0	0	6.483067	31.0656	117	0
185.52.0.113	58004	168.235.94.254	8080	6	456	6	456	0	0	3.482817	31.0579	117	0
185.52.0.113	42146	2.31.248.86	8080	6	456	6	456	0	0	0.482455	31.0502	117	0
92.63.197.18	54807	185.52.0.113	3357	3	168	3	168	0	0	31.995989	6.8302	196	0
92.63.197.18	54807	185.52.0.113	3309	3	168	3	168	0	0	29.491097	6.8184	197	0
185.191.171.13	16504	185.52.0.113	443	27	8714	17	1919	10	6795	0.890408	1.0221	15 k	53 k
185.191.171.26	38270	185.52.0.113	80	12	3650	7	845	5	2805	8.126320	0.3633	18 k	61 k
185.191.171.19	15924	185.52.0.113	80	11	3582	6	777	5	2805	29.810500	0.1650	37 k	135 k
89.190.156.35	51252	185.52.0.113	80	9	711	5	321	4	390	14.998940	0.0085	303 k	368 k

Name resolution
 Limit to display filter
 Absolute start time
Conversation Types ▼

Copy ▼
Follow Stream...
Graph...
Close
Help

Figure A.3: Biflows statistical attributes