

Manuscript version: Author's Accepted Manuscript

The version presented in WRAP is the author's accepted manuscript and may differ from the published version or Version of Record.

Persistent WRAP URL:

<http://wrap.warwick.ac.uk/162311>

How to cite:

Please refer to published version for the most recent bibliographic citation information.

Copyright and reuse:

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions.

Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher's statement:

Please refer to the repository item page, publisher's statement section, for further information.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk.

APIVADS: A Novel Privacy-Preserving Pivot Attack Detection Scheme Based On Statistical Pattern Recognition

Rafael Salema Marques¹, Haider Al-Khateeb¹, *Member, IEEE*, Gregory Epiphaniou², *Member, IEEE*, and Carsten Maple², *Fellow, IEEE*

[1]Wolverhampton Cyber Research Institute (WCRI), University of Wolverhampton, WV1 1LY, UK (e-mail: R.S.Marques@wlv.ac.uk; H.Al-Khateeb@wlv.ac.uk)

[2]Warwick Manufacturing Group (WMG), University of Warwick, Coventry, CV4 7AL, UK (email: Gregory.Epiphaniou@warwick.ac.uk; CM@warwick.ac.uk)

Abstract—Advanced cyber attackers often “pivot” through several devices in such complex infrastructure to obfuscate their footprints and overcome connectivity restrictions. However, prior pivot attack detection strategies present concerning limitations. This paper addresses an improvement of cyber defence with APIVADS, a novel adaptive pivoting detection scheme based on traffic flows to determine cyber adversaries’ presence based on their pivoting behaviour in simple and complex interconnected networks. Additionally, APIVADS is agnostic regarding transport and application protocols. The scheme is optimized and tested to cover remotely connected locations beyond a corporate campus’s perimeters. The scheme considers a hybrid approach between decentralized host-based detection of pivot attacks and a centralized approach to aggregate the results to achieve scalability. Empirical results from our experiments show the proposed scheme is efficient and feasible. For example, a 98.54% detection accuracy near real-time is achievable by APIVADS differentiating ongoing pivot attacks from regular enterprise traffic as TLS, HTTPS, DNS and P2P over the internet.

Index Terms—APT, pivot attack, privacy-preserving, lateral movement, network flow.

I. INTRODUCTION

IN the last decades, the number of advanced persistent threats (APTs) groups increased over the years, developing new attack vectors as well as the complexity regarding Techniques, Tactics and Procedures (TTP), capable of evading detection patterns used by defence solutions [1]. APT refers to a set of systematic continuous hacking processes targeting an entity to recover and exfiltrate high-value information. Experience shows that APT groups are incredibly efficient in achieving their objectives, tending to be systematic within offensive actions. According to recent studies, it is essential to point out that major nation-state threat actors will continue their efforts in the next years [1].

Identifying an attack in its initial stages, when the opponent has not yet achieved its goals, is essential for an effective defence strategy. To better prepare and reduce the threats detection time, the organisations use Cyber Threat Intelligence (CTI) to identify, understand, predict, and adapt to malicious actors’ behaviours. A technique widely used by cyber adversaries to enable connectivity to the final target is known as a

pivot. This kind of attack aims to achieve connectivity from a normally non-routable network. It expands the restricted access of a compromised device to reach the main target using traffic routing techniques.

When a pivot technique is used during the attack to overcome defences or create connectivity with the target, it generates traffic anomalies that can be identified with flow analysis and statistical methods. In this scenario, the network flow is a valuable data source to identify an ongoing attack and infer indicators of attack (IoA) within assets communication patterns. Flows contain header information about network connections between two endpoint assets. A flow is an aggregation of transmitted network packets which shares the same source IP address, source port, destination IP address, destination port and transport protocol within a time window [2].

APT campaigns can use several pivot attacks and lateral movement to expand their presence and persistence within the network for long periods before reaching their objectives. Therefore the capability to detect this type of attack on time is vital. Additionally, pivot attacks are widely used by APT campaigns [3]. They provide an attractive capability to adversaries because the initial access typically does not correspond to the actual target [4]. Moreover, differentiating pivot attacks from regular traffic is a challenging research problem due to the diversity and high volume of traffic produced by enterprise networks.

To the best of our knowledge, APIVADS is the first pivot detection scheme that does not have restrictions to local networks and is agnostic regarding transport and application protocols. Besides, it is capable of identifying pivot attacks even if it is conducted over complex interconnected networks. Table I present a summary of the main contributions of this paper.

In the remaining part of this paper, background and related work are covered in Section II. The proposed pivot attack detection scheme is then represented in Section III. The methodology including evaluation metrics and testing scenarios are presented in Section IV. Section V discusses the main results and analysis from our experiments, while Section VI concludes this study.

TABLE I
APIVADS CONTRIBUTIONS

Contribution(s)	Description
1	A novel adaptative pivot attack detection scheme (APIVADS) based on flow analysis and statistical methods agnostic regarding transport or application layer with no restrictions regarding complex interconnected networks
2	An efficient data reduction strategy based on the exclusion of traffic flows that are not compatible with a pivot attack regarding APIVADS traffic forwarding pattern recognition model
3	A parametric optimisation mechanism that can improve the detection rates based on the traffic frequency and volume perceived within a pivot tunnel

II. BACKGROUND AND RELATED WORK

A. APT threat models

We can find in the literature a wide variety of APT attack models. The Cyber Kill Chain [5] developed by Lockheed Martin, is a well-known model adopted by the National Institute of Standards and Technology (NIST), pointed by the industry and academia as reference. It enumerates and describes seven steps required by adversaries to achieve their goals. This model has been criticised in the last years due to the traditional perimeter-focused approach and malware dependency [6].

Another important Kill Chain model is the Mandiant Attack Lifecycle [7]. This model presents evolutions regarding the attacker internal network activities contemplating recursive internal reconnaissance and lateral movement. However, it still lends itself to interpretation within indicators attribution to action groups, leading to inconsistent data analysis and less efficiency regarding security personnel workflows [8].

Bryant and Saiedian [8] proposed modifications to the conventional kill-chain models to improve data aggregation and correlation resulting in more detailed alarms to security analysts.

Milajerdi et al. [9] presented HOLMES, a detection system that aims to produce a signal that indicates an APT campaign's malicious coordinated activities. However, due to the need to go unnoticed in their actions, APT modus operandi changes over time. This change in the attacker's behaviour can divert from the presented models, leading to a lack of awareness.

Alminshid and Omar [10] summarised several APT attack models and proposed one that merges the typical attack stages generally present in APT attacks.

The threat landscape evolved regarding APT modus operandi, and the criticism is well-founded because the attacks can emerge from internal adversaries without using a single malware. Since the original model's publication in 2011, modifications have been proposed by scientific authors and cybersecurity professionals over the years.

Some of the most successful detection approaches seek out malicious patterns by monitoring essential environmental changes [11] to create a specific attack condition. For instance,

a remote attacker needs to generate outbound traffic to exfiltrate data when using the internet to support the C&C channel.

In conclusion, even with differences between models, APT attacks share some similarities regarding TTP and attack phases. Furthermore, the capability to identify near real-time offensive actions and infer attack stages is essential to develop a solid cyber awareness for enterprise networks.

B. Privacy-preserving traffic analysis approaches

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 are criticized when applied in fast enterprise networks because they cannot work with end-to-end encryption. However, a recent arising trend partially mitigates the cited drawbacks, addressing a privacy-preserving DPI approach. Authors in [12], [13], [14] proposed a cloud-based provider to support middlebox outsourcing packet inspection while preserving the client's confidentiality when sharing information. To achieve a privacy-preserving model, traffic and detection rules provided to third party middleboxes typically are encrypted [15]. However, all the cited privacy-preserving DPI models use the signature-based paradigm and consequently inherits its issues and limitations, which are well documented in the literature. A signature-based or rule-based detection scheme tries to identify attacks by comparing incoming events with their stored signatures [16]. A signature is a kind of description to represent a known attack using some features. In order to comply with privacy-preserving requirements, the detection is achieved by comparing encrypted payloads with a preexisting encrypted signature. Suppose the adversary implement the polymorphic blending technique (PBT) to protect the traffic [17] or use actively evolving threats techniques to morph the traffic [18]. In that case, we face a scenario where a signature-based detection strategy will fail to detect the malicious traffic. Authors in [14] addressed privacy issues related to DPI techniques for outsourced middleboxes, their proposal prevents a new ruleset in the system to be linked to a previous inspection results. The authors stated that the strategy slightly increased the resilience making adaptive attacks less effective. Although, the signature-based method is limited to a knowledge repository, which is unsuitable for detecting unknown attacks.

Despite the significant improvement regarding detection rates achieved by the adaptive signature-based schemes, this type of solution still present difficulties identifying advanced techniques as PBT [17]. The PBT uses polymorphic data obfuscation techniques to bypass signature-based Intrusion Detection Systems (SIDS) and blending to evade anomaly-based Intrusion Detection System (AIDS). According to the authors in [17], AIDS is capable to detect simple polymorphic attacks because their byte frequency differs from the one seen in the legitimate traffic. Therefore, PBT collects raw packets to create a traffic profile and adjust the payload byte frequency

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

There are some detection approaches used by SIDS to defeat simple polymorphic attacks [19]. However, when polymorphic attacks are combined with different techniques as PBT, the chances of evasion increases because of difficulties in modelling complex systems. Additionally, cyber adversaries are evolving their techniques constantly, and already are exploiting the knowledge of the machine learning detection algorithms to evade defences [20]. Besides, experience shows that it is a matter of time before attackers adapt their TTP to new defence strategies.

According to [21], the privacy-preserving concept applies to scenarios where third-party entities process sensitive information. Therefore, host-based approaches are by default privacy-preserving because they do not share sensitive information. The flow-based analysis paradigm does not inspect packets payload, which is good from the privacy point of view. It does not have restrictions regarding end-to-end encryption or proprietary malware ciphered traffic. Unlike DPI approaches, it's mechanism extracts packet header attributes to create flows, which are used as input to algorithms without spending computational efforts regarding payload inspection and storage resources.

C. Flow-based traffic analysis

According to [22], 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 in a measurement point, process it and generate alerts. Among passive traffic analysis strategies, the most common approaches found in the related cybersecurity literature are flow-based and DPI approaches.

Typical flow attribute of unidirectional NetFlow data is presented by [23]. Those attributes are extracted from the set of packets that share the same flow. A flow is defined in [22] 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.”

A Bidirectional flow (biflow or conversation) is a flow composed of packets sent in both directions between two endpoints [24].

NetFlow-like analysis systems have been used for network monitoring, planning, and billing [24]. However, flow-based analysis attracted attention by security researchers, emerging as a fundamental approach to be explored in the field of cybersecurity [25], [26], [27].

D. 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. Apruzzese et al. [28] 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 [29], 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.

E. Related works

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 II 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.

TABLE II
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. [30]		✓			✓			✓			✓
Bai et al. [31]	✓		✓	✓					✓		
Apruzzese et al. [28]		✓			✓			✓			

To the best of our knowledge, [28] is the first paper that specifically addresses pivoting by introducing an attack detection method. However, authors in [30] stated that the approach is not feasible to detect pivot attacks in enterprise networks when considering internal and external connections. Another issue stated by [30] 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. [30] evolve the first pivot detection algorithm proposed by [28]. Unlike earlier research, their work combines human expertise with machine learning techniques to address pivot detection when dealing with 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 [30] 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 [32]) 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 [30] and [28] with APIVADS, the approaches present similar functionalities and detection results in simple scenarios. However, the algorithm created by [28] 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 [28] affects the complexity of the algorithm drastically. The pivot length size, which can increase the complexity of the theoretical worst-case scenario of [28] 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 [31] 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 [31] 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 [30] concerning [28] 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.

III. APIVADS: ADAPTIVE PIVOTING DETECTION SCHEME

A. Scheme's design objectives and scope

The scheme offers a novel privacy-preserving detection scheme to determine ongoing pivot attacks near real-time. To address scalability and cybersecurity situational awareness, a distributed agent-based approach was used to achieve detection and a centralised strategy to aggregate the results.

APIVADS agents are restricted to the local device traffic. This fact limits the agent perception to biflows that evolves the local host. However, this natural limitation is desirable to address privacy regarding sensitive data processing. Likewise, this restriction means that a single APIVADS agent cannot achieve an enterprise network cybersecurity awareness about pivot attacks. However, suppose 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

the distributed detection and a holistic view regarding ongoing pivot attacks within the devices monitored.

APIVADS pivot attack detection capacity is not affected by traffic that presents encrypted header information in most cases. An exception is made by the header attributes used to aggregate packets and generate flows: Transport protocol, source IP, destination IP, source port and destination port. The main functionality of a Pivot node is to forward traffic between two endpoints creating connectivity between the attacker and the target. Therefore, this is not practical to create a pivot tunnel without access to the packet headers attributes information used by APIVADS.

APIVADS can infer pivot tunnels when dealing with anonymous traffic as the onion routing circuit used by TOR [33]. However, identifying the real origin of anonymous traffic is not in the scope of this work. When processing traffic forwarded by anonymous routing techniques, APIVADS will identify the endpoint that is forwarding the anonymous traffic (relay node) and supporting the pivot attack, not the original IP that generated the traffic.

Unlike other detection algorithms such as [28], our detection scheme is not restricted to internal network communications. The Pivot node can be identified even if the nodes are spread over the internet. However, it is assumed that the adversary is not subverting the traffic perception.

Additionally, APIVADS does not require previous training or a knowledge repository. The data used by the detection scheme is the actual traffic perceived in the device that is collected and processed by the APIVADS algorithms continuously.

Finally, the APIVADS agent must process the pivot attack incoming and outgoing traffic biflows to achieve detection.

B. Pivot attack

A pivot scenario is illustrated in Figure 1. The bullets represent packet flows between nodes from one to four. Generally, the pivot technique is used when the attacker node cannot exchange information directly with the Target node (bullets 5 and 6). To achieve connectivity with the Target node, the attacker node needs to gain access to other network assets (Pivot nodes), which is used to forward traffic between the attacker node and the Target node.

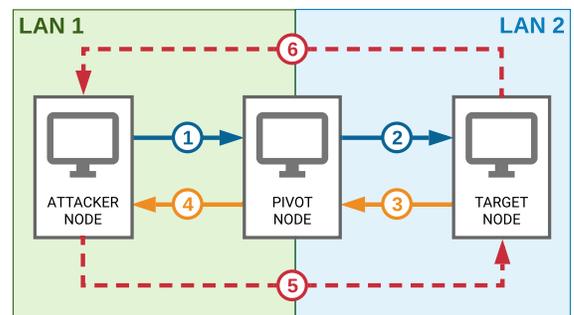


Fig. 1. Pivot attack scenario

A pivot attack must respect a logical sequence regarding communication flux. For instance, the flow corresponding to bullet 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 will exist after flow 3 arrives at the Pivot node when the Target node sends traffic to the attacker node. Another intrinsic pivot characteristic observed in the communication flow is that flows 1 and 2 will always happen before 3 and 4, with a small-time difference between them regarding packets perception by the Pivot node.

C. Privacy-preserving characteristics and description

Privacy is crucial for the success of a defence solution. Differently from DPI approaches, APIVADS adopt flow-based analysis not requiring payload inspection, and consequently compatible with end-to-end encryption simply disregard packet payloads. It just aggregate specific packet headers attributes to create flows. We proposed a specific flow attributes structure (See Table IV), that is detailed in Section III-H which is used as input to the detection algorithms. Our approach does not need a knowledge repository, eliminating the constant update dependency of signature-based strategies and privacy concerns regarding detection rules. Cryptography is the standard approach to preserve privacy when sharing information among different parties [34]. However, the privacy-preserving concept just is applicable in scenarios where sensitive information is shared with third-party entities. Therefore, because APIVADS agents data processing is strictly related to its host traffic flows attributes and no sensitive information is shared with third parties, our detection scheme is by default privacy-preserving and does not demand anonymization or extra privacy requirements regarding data processing.

D. Detection scheme and flow-based pattern recognition model

The traffic perception of APIVADS agents installed at Pivot nodes is the basis of our 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 can infer biflows between itself and other endpoints. Based on the perceived biflows attributes, our APIVADS agent performs statistical calculations to find pivot attacks patterns between biflows. Consequently, the agent installed in the Pivot node will be capable of inferring a pivot attack scenario transforming the perceived traffic into APIVADS flows (defined in Table IV) and applying the detection filters described next in this Section.

In Figure 1, we have two biflows between three hosts: The former is represented by flows one and four (communication between the attacker node and the Pivot node), and the latter is identified by flows two and three (traffic between the Pivot node and the Target node). To detect a pivot scenario using the assumptions presented above, we need to find a correlation between biflows. Let B be a biflow formed by 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 specific 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 1 as an example, if the Pivot node can identify similarities regarding specific patterns between the conversations $B(1, 4)$ and $B(3, 2)$, we can infer a pivot scenario between the attacker node and the Target node supported by the Pivot node.

Biflows within a pivot tunnel present similar duration time (time difference from the first packet perceived and the last one). The exclusion of biflow pairs that do not fit this premise is an efficient data reduction measure because legit biflows with similar duration time are unusual. Based on the assumption already stated that packets within a pivot attack occur chronologically in an interspersed way, using statistical methods is possible to measure and compare the degree of packets alternation regarding arrival time between biflows to identify traffic similarities.

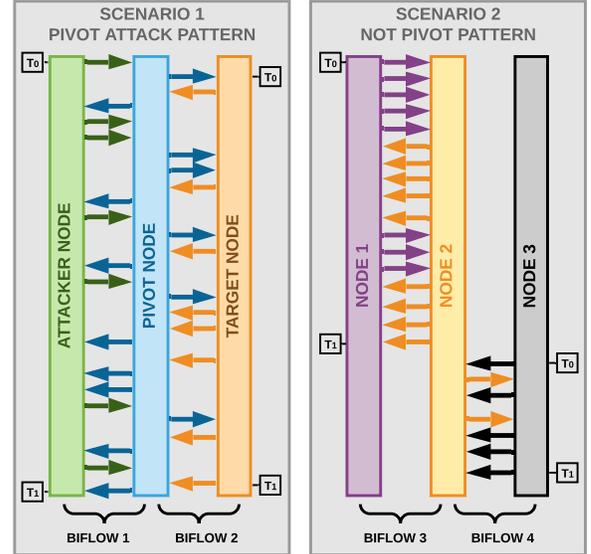


Fig. 2. Pivot and not pivot traffic patterns comparison

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

Let D_t be the absolute duration time difference between two biflows. The calculation of D_t is trivial and essential to address similarities between biflows regarding duration time which correspond to an IoA of a pivot attack. The lower is D_t result

between two biflows the similar are biflows duration time. To compute D_t between biflow 1 (B_1) and biflow 2 (B_2), we calculate the absolute duration 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) that is the maximum value of D_t to identify biflows with similar duration time. If the D_t result is more significant than pD_t , the evolved biflows do not have the necessary similarity regarding duration time. A biflow pair (BP) is composed of two biflows B_i and B_j that present similar duration time, compatibility regarding packet alternation and plausible endpoints IP addresses correlation to be considered part of a pivot tunnel. Section III-I introduces the three filter description and pseudocode algorithms responsible for reducing data, identifying biflow pairs, and consequently detecting pivot tunnels.

Scenario 1 shows a high alternated traffic between the three assets with similar D_t in a time window characterising a pivot attack. However, in scenario 2, we can observe two different concentrations of packets, the first between nodes 1 and 2 and the latter between nodes 2 and 3, not respecting feasible pivot traffic forward logic between nodes 1 and 3.

A BP presents a similar start time in a time window. In scenario 2, Let D_s be the maximum start time difference between biflow 3 (B_3) and biflow 4 (B_4). It can be calculated by the absolute difference between the biflows start times $D_s = |B_3(T_0) - B_4(T_0)|$. Similarly with D_t , the small is D_s result the similar is the biflows regarding start time. For instance, in scenario 2, the biflows start time does not share the same time window. Consequently, they are discarded as a candidate to be considered a BP by the Duration filter algorithm (Algorithm 1).

Because the Pivot node just forward the traffic between endpoints, related biflows tend to present a similar number of total packets perceived N . Therefore, we defined the parameter pN , which correspond to the maximum result concerning the ratio computation between biflows N values.

Finally, the identification of a BP is a strong IoA of a pivot attack. The following criteria are used by the APIVADS detection scheme 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 bigger than pD_f . (3) The D_s result regarding B_i and B_j must be lower than pD_s . (4) The computation of the N ratio between two biflows must be lower than pN . (5) Packets arrival time must have alternation between biflows. (Further details regarding packets alternation pattern are provided in Section III-I).

E. APIVADS data processing phases and threat model overview

The detection strategy is composed of two distinct data processing phases: detection and aggregation. Firstly, we use a host-based approach to address the detection. In this phase, the APIVADS agent collects the device perceived traffic headers, update the set of biflows and process them using data reduction and statistical techniques to infer a pivot attack. Additionally, when a new packet is perceived the extracted attributes are

never stored on disk, they are automatically aggregated to the set of biflows to save storage resources. Secondly, a distributed approach is used to aggregate the pivot attacks detection information. When an APIVADS agent detects a pivot attack, it reports the event to a third-party CTI Framework, which aggregates all pivot attack events to identify connections among the messages and infer the complete pivot tunnel. Figure 3 illustrates the two detection phases that will be detailed next in this Section.

F. APIVADS modules interaction

APIVADS uses a distributed strategy based on agents installed in network assets to identify pivots tunnels of any length. Figure 3 illustrates the data processing steps and interaction among the four APIVADS agent modules. The Data collection module receives traffic information from the device network interfaces and continuously collects packets header attributes of interest from new traffic. The Data extraction module aggregates the collected header attributes, updating a set of APIVADS flows whose structure is presented in Table IV. Therefore, APIVADS flows are clustered in biflows forwarded to the Detection filter module responsible for reducing data and inferring biflows pairs part 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 alert messages and actionable information [35], [36]. All modules and interactions among APIVADS entities will be explained in detail next.

Parameters are envisioned in APIVADS to optimise our detection scheme providing control and balance over detection metrics. Table III presents a parameter list 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 influences directly the number of packets processed by the algorithm within a biflow. Small values of T_w can restrict the amount of data sample while large values demand time and processing power. Because third-party outsourcing is not in the scope of this work, it is required that the time spent to process data and execute the algorithms (T_p) be smaller than the division of T_w by E to avoid data processing resource exhaustion. Finally, pR influence the algorithm detection accuracy. A restricted value of pR can increase false negative results, while a tolerant value probably will lead to a false positive scenario. All parameters are detailed by the the pseudocode of the algorithm as described in Section III-I.

G. Data collection module

This module is responsible for collecting and aggregating the perceived packets' metadata, preserving the temporality. Our flow-based approach uses biflows and passive network monitoring (See Section II-C), evaluating the most recent packets' arrival time within biflows. The number of packets considered in a biflow by the detection scheme is defined by

TABLE III
DETECTION SCHEME ALGORITHMS' PARAMETERS

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

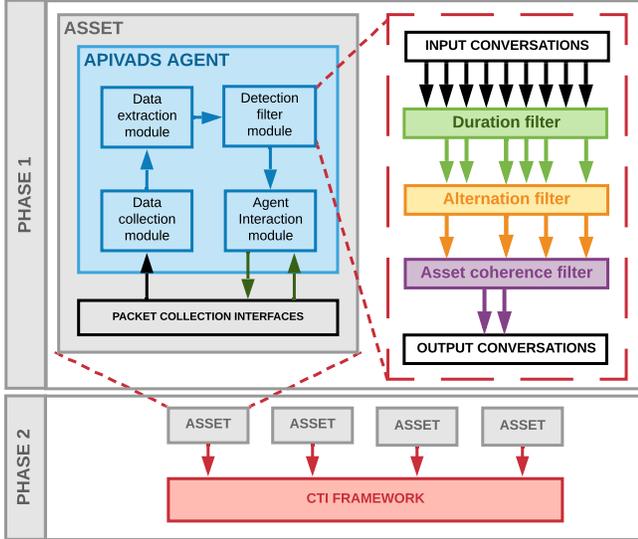


Fig. 3. APIVADS threat model diagram

the L parameter. This variable provides a mechanism to control the biflow sample size.

To measure the biflow duration time we adopted the following life span expiration rules:

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

Additionally, the pD_f parameter corresponds to the mini-

imum biflow duration time value considered by the detection algorithm. This parameter helps improve data reduction, discard irrelevant biflows due to the nature of the attack we intend to detect.

We do not use FIN or RST flags available in TCP packets' attributes because UDP traffic does not contemplate it and our approach is agnostic regarding the transport layer. Besides, the adversary can bridge data from different transport layers protocols (e.g. TCP to UDP) to bypass defence mechanisms.

H. Data extraction module

The packets gathered in the data collection module are transformed into flows that are processed to become 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. Besides, some protocol-specific attributes are not used by our detection scheme since we address an agnostic approach regarding the transport and application layers. The selected flow feature attributes and data structure are shown in Table IV.

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

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

I. Detection filter module

This module performs data reduction and statistical pattern recognition using three filter algorithms. Each filter receives input as a set of biflows. The first filter is the Duration filter algorithm (Algorithm 1). It is responsible for reducing the set of biflows output from the Data extraction module, which does not fit in a pivot attack pattern. Initially, biflows with a duration time bigger than the predefined parameter pD_f are discarded

to avoid ephemeral connections. This module verifies the degree of similarity between two biflows regarding the data reduction parameters. For example, If the resultant D_t is lower or equal to pD_t , it means that the biflows present a degree of similarity compatible with a pivot attack regarding duration time, and the evolved biflows (B_1 and B_2) are selected as candidates to become a new biflow pair $BP(B_1, B_2)$.

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 shares 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 with the result of D_t . It corresponds to the maximum limit of D_t computation between biflows to create a BP regarding duration time.

Parameter: pD_f is a predefined parameter that is compared with every biflow D_f value. If D_f is lower than pD_f the biflow is discarded.

Parameter: pD_s is a predefined parameter that is compared with the result of D_s . It corresponds to the maximum limit of D_s computation between biflows to create a BP regarding absolute start time difference.

Parameter: pN is a predefined parameter that is compared with the computation of the total bytes traffic ratio between two biflows.

Output : An array of biflow pairs BP

- 1 Compare each biflow D_f value with pD_f . If D_f is lower than pD_f the biflow is discarded.
- 2 The remaining biflows in B are compared to each other. For the sake of simplicity, let name the biflows to be compared as 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 result of B_i and B_j computation is lower than pD_s
- 5 **and** pN is bigger than the ratio of B_i and B_j regarding N
- 6 **then**
- 7 | Append the biflow pair to the result array $BP(B_i, B_j)$
- 8 **else**
- 9 | Select the next biflow candidates until all the remaining eligible biflows are tested on each other.
- 10 **end**
- 11 **return** BP

The biflow pairs created in the Duration filter are submitted as input to the Alternation filter algorithm (Algorithm 2). This algorithm is responsible for checking if the biflows that compose a BP present alternation regarding packets arrival time in the Pivot node. The date-time reference attribute array of the flows that compose a BP are merged chronologically and ordered, preserving the flow identification. Next, the array is processed by the algorithm to compute the R value, which is the maximum packet's sequence of the same flow in the merged array. The achieved value of R is compared to a predefined parameter pR , that corresponds to the maximum sequence of packet in the same flow. Therefore, an R value bigger than pR are discarded from the set of BP received from the previous filter.

The last filter is the Asset coherence filter algorithm (Algorithm 3). It receives the remaining BP that meets the previous filter's requirements: biflows pairs with duration time

ALGORITHM 2: Alternation filter algorithm

Input : A set of biflows 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 with the result of R computation. If R is bigger than pR , the biflow is discarded.

Output : An array of biflow pairs BP

- 1 Merge the two biflows that compose a BP in a temporary array preserving the flow identification and the packet arrival time chronology.
- 2 The temporary array size is limited by the the L value and is filled with the most recent packets arrival time reference attribute.
- 3 The algorithm searches the temporary array for the biggest sequence of packets within the same biflow R .
- 4 **if** R result is bigger 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

bigger than pD_f , a similar D_t , D_s less than pD_s , N ratio computation less than pN and present alternation between packets regarding different flows. This module checks for a plausible correlation of the BP set of IPs H with the endpoints, discarding inconsistent pairs (e.g. biflows with the same source and destination IP and ports). Additionally, the merged array of chronologically ordered packet timestamps are split into quarters. Each quartier must contain all possible flows within the biflows that compose a BP validating the alternation between flows over the traffic sample. For example, a BP formed by the biflows validating the alternation between flows over 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.

J. Agent interaction module

This module is responsible for interacting with CTI Frameworks. When the detection filter module identifies a pivot attack, an alert message is generated and forwarded to the CTI Framework, which has a holistic view regarding all alert messages received from APIVADS agents.

Our detection scheme does not need external information to identify an asset serving as a Pivot node providing connectivity between two other assets. However, it is not able to perceive the complete length of the pivot tunnel. To mitigate this limitation, all alert messages must be merged by the CTI Framework to identify connections between Pivot nodes and determine the complete pivot tunnel. Due to the distributed pivot detection strategy, our detection scheme achieves scalability and the possibility to be used in complex networks.

Every alert message worst-case scenario identifies a Pivot node and two more assets involved in the attack. Table V presents two alert messages samples received by the CTI framework from Pivot nodes.

ALGORITHM 3: Asset coherence filter algorithm

Input : A set of biflows 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 a local asset IP address.

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 contains one IP present in the H array
- 3 Merge the two biflows that compose a BP entry in a temporary array preserving 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 that compose the biflows B_i and B_j of BP (See Section III-D) 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

TABLE V
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 Table V information based on an acceptable time difference and the same endpoints attributes shared between alert messages is trivial to infer connection among Pivot nodes. For instance, lines 2 and 3 share the same attributes with a reasonable temporal difference, implying a pivot tunnel length of two. The reasoning between alert messages in Table V can be interpreted as a pivot tunnel diagram (see Figure 4).

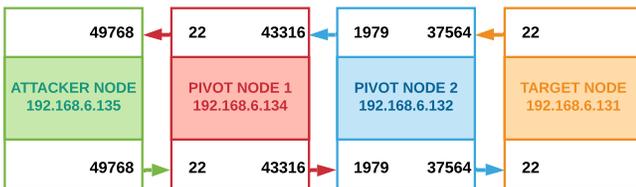


Fig. 4. Pivot tunnel representation of Table V alert messages sample

Some advanced pivot attack tools are capable of bridging

traffic between transport layer protocols. Our detection scheme can infer a pivot attack even if the opponent uses such techniques as indicated by the second alert message, where the Pivot node 2 receives TCP and forward UDP traffic.

Our approach does not depend on CTI frameworks to detect pivot attacks or to define the pivot length. The connection between messages is trivial, and could be done in several ways, not requiring significant computational effort. However, the distributed pivot results must be merged somehow. We choose to send alert messages to CTI frameworks based on the understanding that this kind of threat information is vital to proactively identify APT actors.

IV. METHODOLOGY

Experiments are paramount to evaluate the efficiency and accuracy of the APIVADS scheme. This includes optimisation tests to improve parameters and achieve better detection rates. Initially, a virtual network scenario was used to conduct APIVADS validation experiments, which have been complemented with real networks 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 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 next in this section.

A. Evaluation metrics

In theory, 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 evaluation metrics presented in Table VI to measure and compare results.

TABLE VI
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, $TPR = TP / (TP + FN)$.

B. Parameter optimisation tests

Pursuing the objective to achieve the best detection rates in our experiments, we performed several tests to identify the best

parameters combination regarding 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 common enterprise network traffic and pivot attacks.

C. 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) that impersonate a real environment generating different types of standard enterprise traffic. Figure 5 presents the virtual network experiment diagram. Boxes represent the network hosts that are differentiated by colours and letters. Information regarding IP addresses and network interface is next to every host with the correspondent colour code. Every numbered red arrow represents a *BP*.

Regarding the pivot attack identified by bullets 1 and 2, Host A is the 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 located 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”). Additionally, hosts D and E are not involved with pivot attacks, being used in the experiment to generate regular enterprise network traffic and validate APIVADS regarding false positive results.

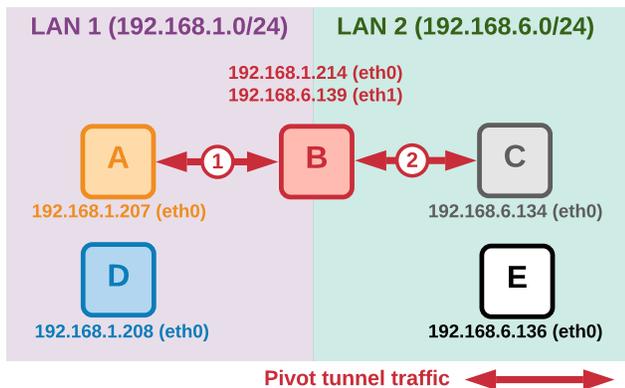


Fig. 5. Virtual network experiment diagram

We seek to validate our implementation during the virtual network experiments and identify the influence of the parameters described in Table III in the detection scheme. To determine the ideal parameters combination regarding the pivot tunnel traffic volume and frequency, we used the evaluation metrics presented in table VI.

D. Real network experiments scenario

The main objective of the experiment conducted in the real environment was to check APIVADS behaviour when exposed

to common connectivity challenges such as latency and packet loss. Additionally, this scenario is useful to identify the impact of the cited connectivity drawbacks regarding pivot attacks 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 VI to evaluate the parameter combination results regarding the ongoing pivot tunnel traffic volume and frequency.

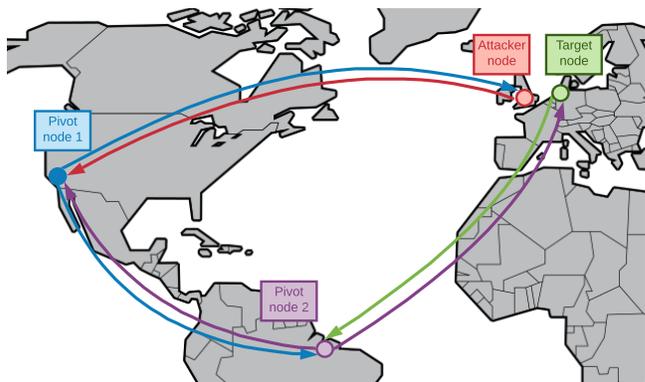


Fig. 6. Pivot attack experiment conducted over the internet

The pivot attack scenario illustrated by Figure 6 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 present in experiments is similar to standard protocols used by enterprise networks (SMTP, IMAP, HTTP, HTTPS, and DNS). Differently from 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 will simulate different attacks, and consequently can change volume, frequency and payload according to the experiment objective.

E. Evasive pivot techniques detection

Skilled adversaries can utilize 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 [28]. To determine if our pivot detection scheme can detect evasive pivot attacks, an experiment using the same scenario described in Section IV-D 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 aims to observe if our detection scheme is resilient to intentional propagation delays and observe if a parameter change is necessary to achieve detection. The delays were applied to the incoming and outgoing malicious traffic at Pivot node 1 and 2 hosts.

V. RESULTS AND DISCUSSION

A. Virtual network experiment results

During this set of experiments described in Section IV-C, our initial objective was to find an adequate parameter combination regarding detection metrics to spend less computational resources as possible. 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 collect traffic. And if we combine it with a considerable L value greater than the number of packets perceived within the biflow, the detection will not happen due to the lack of packet samples. Our approach to determining the best parameters combination was based on the amount of PPS (Packets per second) of the pivot tunnel. Let P_{tot} be the total traffic imposed to the host, which affects the time to process the algorithm (T_p) and can cause the malfunction. And P_{piv} the traffic within the pivot tunnel, which is used as traffic pattern 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 send and receive 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 to 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 ideal 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 . This 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 1, which were coined as Pivot Balance Equation (PBE).

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

Data reduction parameters were defined to discard biflows that are not compatible with the pivot attack traffic pattern. Therefore, we defined pD_f as 4 seconds to discard ephemeral biflows, pD_t as 0.01 seconds due to the virtual environment free from intrinsic network delays, pD_s was defined as 1 second because some terminal commands can demand time to generate output, and a pR of 5 sequential 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.

With the initial parameters defined, we conduct an experiment to observe the impact of detection metrics with T_w variations. Additionally, this experiment was used to verify the possibility to address near real-time detection. Figure 7 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 VII.

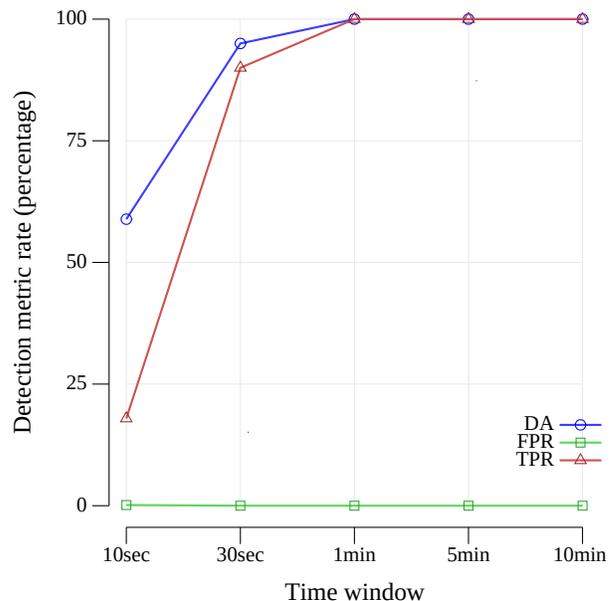


Fig. 7. Experiments result in the function of T_w

TABLE VII
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%

According to PBE computation, a fair value of T_w must be bigger 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 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 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. This fact indicates that PBE is an adequate reference to define APIVADS parameters in function of the P_{piv} we intent to detect. Additionally, even in experiments with restricted T_w values, the FPR results were almost insignificant. This 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 XI. It can be noted that Apruzzese et al. [28] has 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 [28] 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 great or equal to 60 seconds as shown in Figure 7. Note that unlike APIVADS host-based approach, the detection strategy proposed by [28] uses a network-based approach which does not address the performance challenges imposed by the near real-time detection. Additionally, [28] 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. [30] 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 automatically 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, [30] 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 [30] results will be provided in Section V-B.

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 8 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 this scenario, we must set E with a value bigger than T_p and less than T_w . Table VIII provide the detailed results represented in Figure 8. The resiliency of the algorithms regarding FPR was confirmed when respecting the PBE conditions. It achieved the worst-case scenario of 1.25%

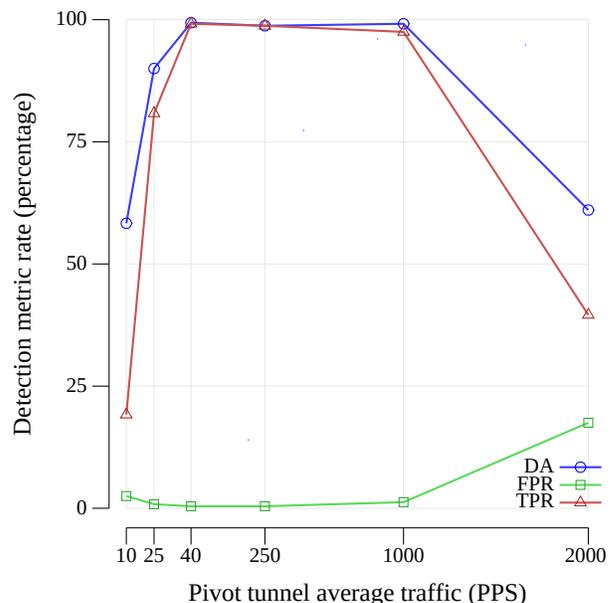


Fig. 8. P_{piv} influence in detection results

of FPR with 1000 PPS. During the experiment with P_{piv} values of 10 and 25, the amount of traffic was not sufficient 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.

TABLE VIII
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%

Figure 9 represents P_{piv} traffic in the function of time. Ambar bars over the x-axis illustrates 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 interval. When a new time window begins, the collected data is discarded. Moreover, we reduce the necessary computational power to execute the detection algorithms focusing on the recent data. The parameters used in the set of experiments illustrated by Figure 9 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 bigger than E and 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. In this experiment, we intentionally disrespect PBE first condition with which demand 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 interesting to address different types of pivot attacks in the pivot traffic volume and frequency function.

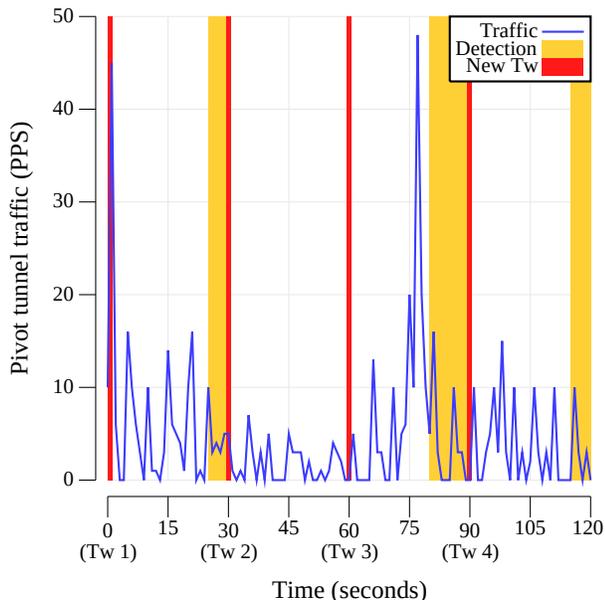


Fig. 9. Detection of pivot attacks with T_w equal to 30 seconds

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 computation, this combination of parameters requires a P_{piv} of 5 PPS. Figure 10 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 real networks experiments.

B. Real network experiment results

Moving towards real networks experiments described in Section IV-D, we aimed to validate if our implementation can identify pivot attacks over the internet. We installed an APIVADS agent in four hosts located 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 discover the parameters that must be changed

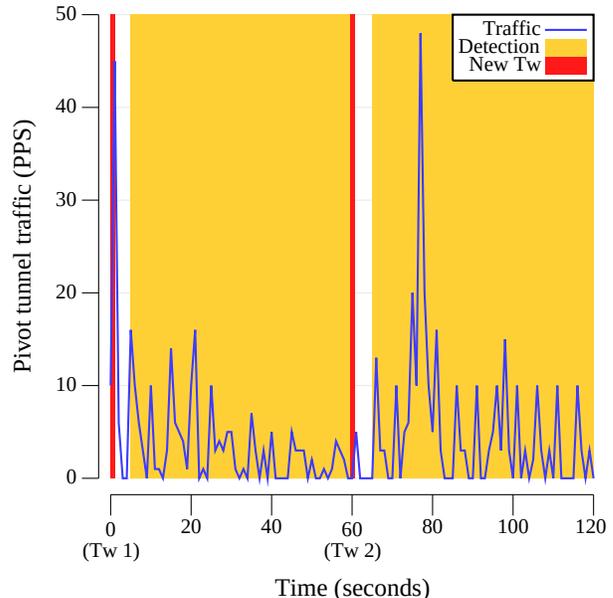


Fig. 10. Detection of pivot attacks over the internet with T_w equal to 60 seconds

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 III) 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 to do not discard biflows which are part of the pivot attack over the internet.

Besides details already described in Section IV-D, 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 was able to identify the ongoing pivot attack with excellent detection metrics rates in both Pivot nodes, which is comparable to the results conducted in the virtual environment when respecting PBE. The detailed detection metrics achieved with this experiment is presented in Table IX. The detection metrics high rates achieved with this experiment validates APIVADS regarding detecting pivot attacks without restrictions to the local network only.

TABLE IX
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 [30] when dealing with BitTorrent and other p2p protocols, it was included in the regular Pivot nodes traffic already presented in Section IV-D during the next experiments. Authors

in [30] observed a high rate of FP caused by the BitTorrent protocol behaviour regarding frequent connections initiation and reception with a small 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. We observed that besides the other data reduction criteria, biflows part of the same pivot tunnel tends to have a similar number of total bytes transferred between endpoints. Therefore, we created the pN parameter to define an acceptable ratio limit. Including this condition at the Duration filter algorithm proved efficient to discard 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 X.

TABLE X
BITTORRENT PROTOCOL EXPERIMENT DETECTION METRIC RATES

Host	TPR	FPR	DA
Pivot node 1	99.58%	2.08%	98.75%
Pivot node 2	98.75%	1.66%	98.54%

According to the detection rates achieved in previous work, Table XI helps to verify that our scheme has either outperformed the overall results reported by studies such as [30] in a real network environment or in the case of [28] was comparable while our study offers host-based detection along side other features as discussed earlier.

TABLE XI
COMPARISON WITH OTHER DETECTION ALGORITHMS

Detection approach	TPR	FPR	DA
APIVADS	99.17%	1.87%	98.65%
Husak et al. [30]	53.84%	4.51%	91.78%
Bai et al. [31]	-	-	99.98%
Apruzzese et al. [28]	100%	0.00%	100%

APIVADS results presented in Table XI are composed of Pivot node 1 and 2 detection metrics average when the BitTorrent traffic was included in the real network experiment. Regarding [31], the best results among different classifiers have been achieved with the stand-alone LogitBoost classifier (LB). With regards to the missing values in Table XI, the authors in [31] 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 [30] couldn't provide an efficient detection algorithm when exposed to protocols that

present similar patterns to pivoting, hence achieving an FPR of 99.99%. However, for the sake of completeness, we included in Table XI 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 [30], or limited to specific operating system events such as [31]. It is also not limited to private networks when compared to [28] as explained by its authors in [30]. 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.

Next, we address the possibility of identifying different APT attack stages based on P_{piv} traffic frequency and volume changes. This information is useful to predict the actual adversary objectives and possible next steps. Therefore, Table XII shows 3 parameter templates optimized 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 stages changes based on specific patterns of P_{piv} . We used the same set of parameters of the previous experiment just changing the specific parameters of each setup presented in Table XII.

TABLE XII
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

To validate the parameter setups provided by Table XII, we created a pivot attack with the correspondent P_{piv} for each setup using the same scenario described in Section IV-D.

Regarding Command & Control attack stage detection, we initially had the hypothesis that APIVADS could not be practical to address near real-time detection depending on the necessary processing power and storage resources when dealing with big values of T_w . 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 of proportion between the perceived packets 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 is 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 to APIVADS a restricted set of parameters, even respecting PBE we can have an unacceptable

value of FN. For instance, based on Table XII 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.

To mitigate this drawback, we address the premises defined by Equation 1, conditions 2 and 3. The usage 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 for future work an automatic selection of predefined parameters based on BP total packets number. This mechanism will provide APIVADS with the capacity to adapt the detection parameters based on the observed P_{piv} of the candidate BP .

Based on the presented results, we demonstrate that our detection scheme can enable consistent classification of pivot attacks cyber threat events that feed into Cyber Threat Intelligence (CTI) frameworks with relevant information.

C. Evasive pivot techniques experiments results

According to the experiment described in Section IV-E, it was observed that the T_w , L and D_t parameters have a direct influence on the results and must be adjusted to classify evasive pivot attacks weaponized 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 bigger than L to achieve detection. Therefore, setting a small T_w value with a low P_{piv} PPS can lead to FN results, and this fact 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 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 Section IV-B) to increase the proportion of T_w in the function of Z , resulting in the following variation of Equation 1 to address intentional delay techniques:

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

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 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 are more effective to detect pivot attacks with intentional delays during the preliminary experiments. To conduct the evasive pivot attack detection experiments, we define the following set of 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 XIII.

TABLE XIII
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%

According to the results achieved, we could verify that APIVADS can identify delay-based evasion techniques. However, the detection rates slightly decreased when compared with the prior tested pivot attacks. Intentional delays impose a bigger T_w due to the necessity of more samples that naturally take more time to arrive at the Pivot node.

APIVADS and the algorithm proposed by Apruzzese et al. [28] are the only pivot attack detection approaches that address intentional propagation delays to the best of our knowledge. 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 [28] cannot address pivot attacks with origin at the internet and is restricted regarding specific protocols.

D. Algorithm complexity

The detection algorithm's complexity must be compatible with the input data length and available computational processing power to be practical in real scenarios. According to the presented algorithms in Section III-I, our algorithm's complexity is exponential $O(n^2)$. This fact could lead to inconvenient large processing time imposing restrictions regarding expensive computational demands. However, due to the efficient data reduction already stated in Section V achieved by the filters and the algorithm parameters, the complexity is not a problem when facing real scenarios.

VI. CONCLUSION

This paper proposed a new scheme to identify traffic patterns related to the pivot attack. Our classification scheme addresses the problem with a flow-based approach and statistical techniques using just 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 the requirement of extra information, which is interesting in terms of scalability. Additionally, the detection scheme can define specific parameters to sense specific pivot traffic changes, which is helpful to

infer APT change of attack stage. It is also important to note that the approach is useful to contribute to the cybersecurity situational awareness of a computer network, identifying the nodes involved in a pivot attack while it is happening, even if the attack is conducted over the internet. We 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.

REFERENCES

- [1] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications surveys and tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019.
- [2] M. Ring, D. Schlör, D. Landes, and A. Hotho, "Flow-based network traffic generation using generative adversarial networks," *Computers security*, vol. 82, pp. 156–172, 2019.
- [3] R. Vera, A. F. Shehu, T. Dargahi, and A. 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*, vol. 3, no. 3, pp. 221–243, 2019. [Online]. Available: <https://www.inderscienceonline.com/doi/abs/10.1504/IJMIS.2019.104786>
- [4] A. Greco, G. Pecoraro, A. Caponi, and G. Bianchi, "Advanced widespread behavioral probes against lateral movements," *Int J Inf Secur Res*, vol. 6, no. 2, pp. 651–659, 2016.
- [5] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, p. 80, 2011.
- [6] W. Zeng and V. Germanos, "Modelling hybrid cyber kill chain." in *PNSE@ Petri Nets/ACSD*, 2019, pp. 143–160.
- [7] Mandiant, "APT1 Exposing One of China's Cyber Espionage Units," http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf, 2013. [Online]. Available: http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf
- [8] B. D. Bryant and H. Saiedian, "A novel kill-chain framework for remote security log analysis with siem software," *Comput. Secur.*, vol. 67, pp. 198–210, 2017.
- [9] S. M. Milajerdj, R. Gjomemo, B. Eshete, R. Sekar, and V. Venkatakrishnan, "Holmes: Real-time apt detection through correlation of suspicious information flows," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1137–1152.
- [10] K. Alminshid and M. Omar, "A framework of apt detection based on packets analysis and host destination," *Iraqi Journal of Science*, vol. 61, pp. 215–222, 01 2020.
- [11] G. Ahmadi-Assalemi, H. M. al Khateeb, C. Maple, G. Epiphaniou, M. Hammoudeh, H. Jahankhani, and P. Pillai, "Optimising driver profiling through behaviour modelling of in-car sensor and global positioning system data," *Computers Electrical Engineering*, vol. 91, p. 107047, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045790621000653>
- [12] C. Lan, J. Sherry, R. A. Popa, S. Ratnasamy, and Z. Liu, "Embark: Securely outsourcing middleboxes to the cloud," in *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation*, ser. NSDI'16. USA: USENIX Association, 2016, p. 255–273.
- [13] H. Sun, J. Su, X. Wang, R. Chen, Y. Liu, and Q. Hu, "Primal: Cloud-based privacy-preserving malware detection," in *Information Security and Privacy*, ser. Lecture Notes in Computer Science, vol. 10343. Cham: Springer International Publishing, 2017, pp. 153–172.
- [14] Y. Guo, C. Wang, and X. Jia, "Enabling secure and dynamic deep packet inspection in outsourced middleboxes," in *Proceedings of the 6th International Workshop on Security in Cloud Computing*, ser. SCC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 49–55. [Online]. Available: <https://doi.org/10.1145/3201595.3201601>
- [15] X. Yuan, X. Wang, J. Lin, and C. Wang, "Privacy-preserving deep packet inspection in outsourced middleboxes," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.
- [16] Y. Meng, W. Li, L.-F. Kwok, and Y. 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*, 2013, pp. 181–188.
- [17] M. Casenove, "Exfiltrations using polymorphic blending techniques: Analysis and countermeasures," in *2015 7th International Conference on Cyber Conflict: Architectures in Cyberspace*. NATO CCD COE, 2015, pp. 217–230.
- [18] A. A. Jillepalli, D. C. de Leon, and J. Alves-Foss, "Operational characteristics of modern malware: Pco threats," in *Proceedings of the Fifth Cybersecurity Symposium*, ser. CyberSec '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: <https://doi-org.ezproxy.wlv.ac.uk/10.1145/3212687.3212864>
- [19] S. A. Aljawarneh, R. A. Moftah, and A. M. Maatuk, "Investigations of automatic methods for detecting the polymorphic worms signatures," *Future generation computer systems*, vol. 60, pp. 67–77, 2016.
- [20] Z. Wang, M. Qin, M. Chen, C. Jia, and Y. Ma, "A learning evasive email-based p2p-like botnet," *China Communications*, vol. 15, no. 2, pp. 15–24, 2018.
- [21] F. Mannhardt, A. Koschmider, N. Baracaldo, M. Weidlich, and J. Michael, "Privacy-preserving process mining: Differential privacy for event logs," *Business information systems engineering*, vol. 61, no. 5, pp. 595–614, 2019.
- [22] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow monitoring explained: From packet capture to data analysis with netflow and ipfix," *IEEE Communications surveys and tutorials*, vol. 16, no. 4, pp. 2037–2064, 2014.
- [23] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954 (Informational), Internet Engineering Task Force, October 2004. [Online]. Available: <http://www.ietf.org/rfc/rfc3954.txt>
- [24] B. Trammell and E. Boschi, "Bidirectional Flow Export Using IP Flow Information Export (IPFIX)," RFC 5103 (Proposed Standard), RFC Editor, Fremont, CA, USA, pp. 1–24, Jan. 2008. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc5103.txt>
- [25] G. A. Ajaeiya, N. Adalian, I. H. Elhaji, A. Kayssi, and A. Chehab, "Flow-based intrusion detection system for sdn," in *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2017, pp. 787–793.
- [26] F. Beer and U. Buhler, "Feature selection for flow-based intrusion detection using rough set theory," in *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, 2017, pp. 617–624.
- [27] S. Zwane, P. Tarwireyi, and M. Adigun, "Ensemble learning approach for flow-based intrusion detection system," in *2019 IEEE AFRICON*. IEEE, 2019, pp. 1–8.
- [28] G. Apruzzese, F. Pierazzi, M. Colajanni, and M. Marchetti, "Detection and threat prioritization of pivoting attacks in large networks," *IEEE transactions on emerging topics in computing*, vol. 8, no. 2, pp. 404–415, 2020.
- [29] Y. Shi, X. Chang, R. J. Rodríguez, Z. Zhang, and K. S. Trivedi, "Quantitative security analysis of a dynamic network system under lateral movement-based attacks," *Reliability engineering system safety*, vol. 183, pp. 213–225, 2019.
- [30] M. Husak, G. Apruzzese, S. J. Yang, and G. Werner, "Towards an efficient detection of pivoting activity," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IFIP, 2021, pp. 980–985.
- [31] T. Bai, H. Bian, M. A. Salahuddin, A. Abou Daya, N. Limam, and R. Boutaba, "Rdp-based lateral movement detection using machine learning," *Computer communications*, vol. 165, pp. 9–19, 2021.
- [32] E. Konduru and J. Petree, "Udp to tcp bridge," American Megatrends, Inc, 2011.
- [33] B. Collier, "The power to structure: exploring social worlds of privacy, technology and power in the tor project," *Information, Communication & Society*, vol. 0, no. 0, pp. 1–17, 2020. [Online]. Available: <https://doi.org/10.1080/1369118X.2020.1732440>
- [34] P. C. Weeraddana, G. Athanasiou, C. Fischione, and J. S. Baras, "Perse privacy preserving solution methods based on optimization," in *Proceedings of the 52nd IEEE Conference on Decision and Control (CDC)*, 2013, pp. 206–211.
- [35] I. Vakiliinia, S. Cheung, and S. Sengupta, "Sharing susceptible passwords as cyber threat intelligence feed," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, 2018, pp. 1–6.
- [36] J. Thom, Y. Shah, and S. Sengupta, "Correlation of cyber threat intelligence data across global honeypots," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 2021, pp. 0766–0772.