A FRAMEWORK FOR CHARACTERIZING CYBER ATTACK RECONNAISSANCE BEHAVIORS

Richard B. Garcia-Lebron

Department of Computer Science, University of Texas at San Antonio

A FRAMEWORK FOR CHARACTERIZING CYBER ATTACK RECONNAISSANCE BEHAVIORS

APPROVED BY SUPERVISING COMMITTEE:

Shouhuai Xu, Ph.D.

Rajendra Boppana, Ph.D.

Greg B. White, Ph.D.

Weining Zhang, Ph.D.

Wenbo Wu, Ph.D.

A FRAMEWORK FOR CHARACTERIZING CYBER ATTACK RECONNAISSANCE BEHAVIORS

by

RICHARD B. GARCIA-LEBRON, MSc.

DISSERTATION
Presented to the Graduate Faculty of
The University of Texas at San Antonio
In Partial Fulfillment
Of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

COMMITTEE MEMBERS:
Shouhuai Xu, Ph.D., Co-Chair
Rajendra Boppana, Ph.D.
Greg B. White, Ph.D.
Weining Zhang, Ph.D.
Wenbo Wu, Ph.D.

THE UNIVERSITY OF TEXAS AT SAN ANTONIO
College of Sciences
Department of Computer Science
August 2019

DEDICATION

To my wife Wilnelia Antuna-Camacho and her family for their unconditional support.
To my mother Angie I. Lebron-Milian and grandparents: Eugenio Lebron-Negron (1930–2009) & Carmen L. Milian De Leon (1934–2016) for teaching me values.
To my sister Nelian A. Cordero-Lebron and my cousins for inspiring me to become their role model.

ACKNOWLEDGEMENTS

I want to give thanks to my advisors Dr. Shouhuai Xu, Dr. Wenbo Wu, and the members of the committee Dr. Rajendra Boppana, Dr. Greg B. White, and Dr. Weining Zhang for their guidance and commitment with excellence.

I wish to express my gratitude to my lab colleagues Dr. Marcus Pendleton, Jose Mireles, Eric Ficke, Huashan Chen, and Zheyuan Sun for the help and comrade during this journey.

In addition, I want to express my gratitude to Dr. Nicole Beebe, Dr. Greg B. White and Dr. Rajendra Boppana for their dedication to mentor and lead the National Science Foundation (NSF) Cyber Corbs. Scholarship for Service (SFS) at the University of Texas at San Antonio.

I want to express my gratitude to Dr. Raymond M. Bateman and Dr. Kristin M. Schweitzer from ARMY Research Laboratory South-Cyber for their mentorship during this journey.

I would also like to thank the ARMY Research Laboratory for supporting and sponsoring this work. I would like to extend my gratitude to the Department of Computer Science for giving me the opportunity to become a doctoral student.

I would like to thanks my lovely wife Wilnelia Antuna-Camacho for the support, encouragement, love and for always believing in me during this journey. Last, but not least, I want to thank my mother Angie I. Lebron-Milian for her sacrifices to help me become the human being I am today.

The research presented in the Dissertation was supported in part by an Army Research Laboratory grant #W911NF-17-2-0127, and by a National Science Foundation Scholarship For Service grant.

This Masters Thesis/Recital Document or Doctoral Dissertation was produced in accordance with guidelines which permit the inclusion as part of the Masters Thesis/Recital Document or Doctoral Dissertation the text of an original paper, or papers, submitted for publication. The Masters Thesis/Recital Document or Doctoral Dissertation must still conform to all other requirements explained in the Guide for the Preparation of a Masters Thesis/Recital Document or Doctoral Dissertation at The University of Texas at San Antonio. It must include a comprehensive abstract, a full introduction and literature review, and a final overall conclusion. Additional material (procedural and design data as well as descriptions of equipment) must be provided in sufficient detail to *allow a clear and precise judgment to be made of the importance and originality of the research reported.*

It is acceptable for this Masters Thesis/Recital Document or Doctoral Dissertation to include as chapters authentic copies of papers already published, provided these meet type size, margin, and legibility requirements. In such cases, connecting texts, which provide logical bridges between different manuscripts, are mandatory. Where the student is not the sole author of a manuscript, the student is required to make an explicit statement in the introductory material to that manuscript describing the students contribution to the work and acknowledging the contribution of the other author(s). The signatures of the Supervising Committee which precede all other material in the Masters Thesis/Recital Document or Doctoral Dissertation attest to the accuracy of this statement.

August 2019

A FRAMEWORK FOR CHARACTERIZING CYBER ATTACK RECONNAISSANCE BEHAVIORS

Richard B. Garcia-Lebron, Ph.D.
The University of Texas at San Antonio, 2019

Supervising Professor: Shouhuai Xu, Ph.D.

Sophisticated cyber attacks often start with a reconnaissance phase, which may expose useful information about the attacks that will be waged later. It is therefore important to systematically understand and characterize cyber attack reconnaissance behaviors. However, little research on this matter has been reported in the literature. The present dissertation aims to fill the void by proposing and investigating the first systematic framework for characterizing cyber attack reconnaissance behaviors. The framework consists of three levels of abstractions: *macroscopic*, *mesoscopic*, and *microscopic*. Correspondingly, the dissertation makes the following three contributions.

First, in order to characterize cyber attack reconnaissance behaviors at the macroscopic level, we propose a novel abstraction, dubbed *dynamic attacker-victim relation graphs*, to represent cyber attack reconnaissance behaviors. This abstraction leads to a time series of graphs and allows us to characterize the evolution of the attacker-victim relation over time. We present a case study with a focus on identifying the number of time resolutions that need to be considered in order to obtain a comprehensive characterization of these dynamic attacker-victim relation graphs.

Second, in order to characterize cyber attack reconnaissance behaviors at the mesoscopic level, we propose clustering cyber attackers based on their reconnaissance behaviors over time. We propose a novel abstraction, dubbed *multi-resolution clustering*, to characterize the evolution of attackers' reconnaissance behaviors in adjacent time windows as well as the evolution of persistent attackers' reconnaissance behaviors over multiple adjacent time windows.

Third, in order to characterize cyber attack reconnaissance behaviors at the microscopic level, we propose the novel notion of *attacker reconnaissance trajectory hierarchy tree* for representing temporal and spatial behaviors of cyber attack reconnaissance.

TABLE OF CONTENTS

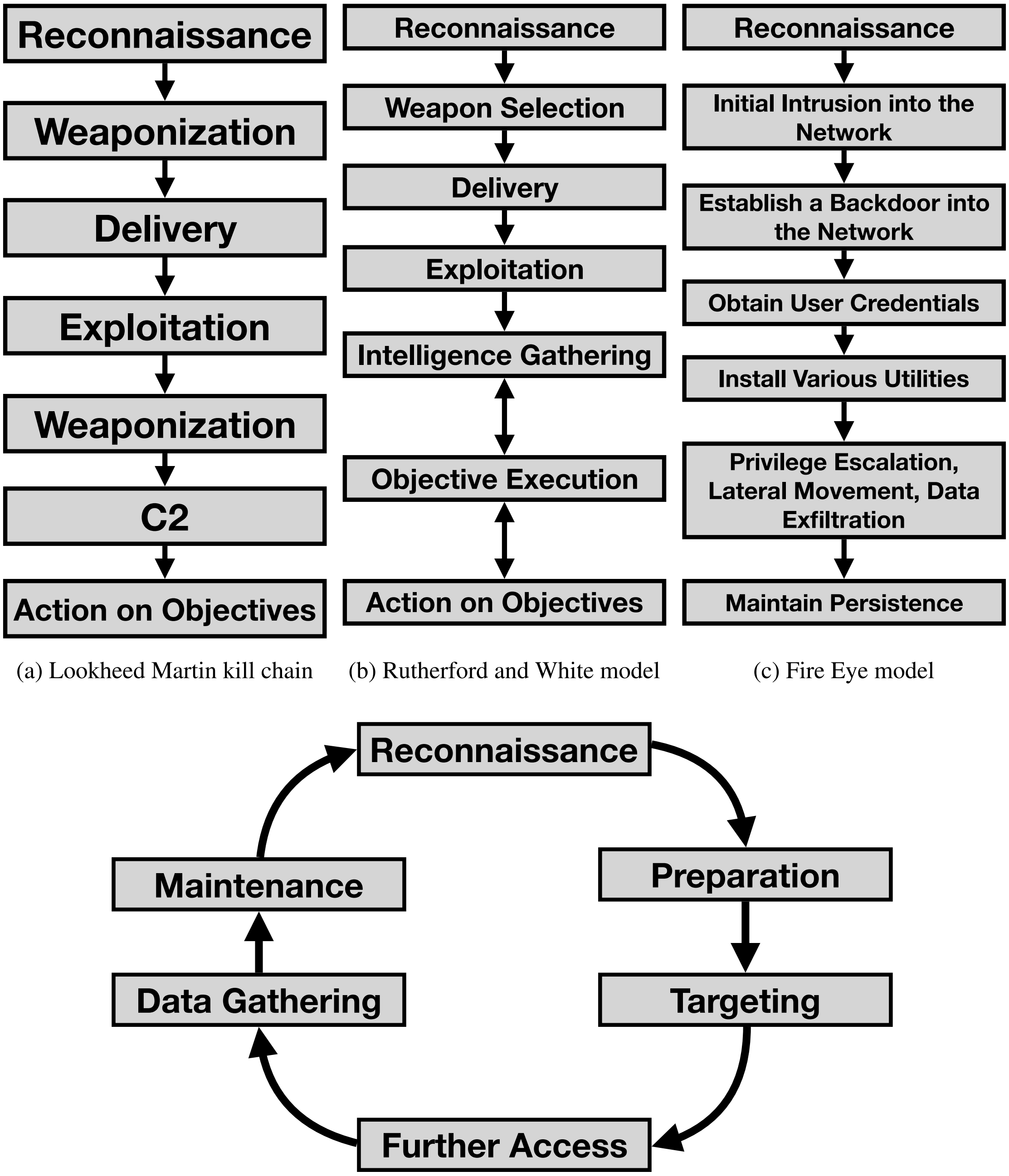
LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

The scale and complexity of cyber threats are becoming increasingly overwhelming. As a consequence, many entities in various sectors have become victims of cyber attacks. For example, data breaches have caused an average loss of \$3.8 millions per incident and an average loss of \$148.0 per breached data record [15]. The prevalence of cyber attack incidents indicates that cyber defense lags largely behind cyber attacks. These incidents can be largely attributed to the following *asymmetry*: Cyber attackers can successfully break into a system by exploiting a single vulnerability. In contrast, the defender must adequately protect the entire vulnerability surface cutting across hosts and networks as well as hardware and software. The urgency and importance of the problem has motivated the development and deployment of many cyber defense tools and sensors for monitoring the situation and collect data, which can be used to various purposes. As cyber attacks get more sophisticated, researchers have built various models to describe and understand sophisticated attacks.

On one hand, there have been a number of efforts at *qualitatively* modeling cyber attacks. As highlighted in Figure 1.1a, researchers at Lockheed Martin introduced the notion of cyber attack kill chain [46, 60, 66, 141], which describes sophisticated attacks, such as Advanced Persistent Threats (APTs), in seven steps: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and act on objective. Mireles *et al.* [81] introduced the notion of *attack narratives* by leveraging the cyber kill chain. Kim *et. al.* [65] extended the the cyber kill chain for multimedia environments and Hahn *et. al.* [56] extended the anatomy to include the cyber-physical layer. As highlighted in Figure 1.1b, Rutherford and White [111] proposed a variant that includes intelligence gathering, objective execution, and exfiltration of information. As highlighted in Figure 1.1c, researchers at Fire Eye introduced a related model for describing APTs [16], which consist of eight stages: initial recon, initial compromise, establish foothold, escalate privilege, internal recon, move laterally, maintain presence, and complete mission. As highlighted in Figure 1.1d, researchers at Command Five Pty Ltd introduced a similar model.



(a) Lookheed Martin kill chain (b) Rutherford and White model (c) Fire Eye model

(d) Command Five model

Figure 1.1: Models for describing cyber attacks: (a) Lookheed Martin cyber kill chain; (b) the Rutherford and White variant of the cyber kill chain model; (c) Fire Eye model; and (d) the Command Five Pty Ltd attack model

.

On the other hand, there have been efforts at *quantitatively* modeling cyber attacks from a holistic perspective. In particular, Xu [134, 136] has pioneered a systematic framework for modeling and reasoning cyber security from a holistic or whole-system perspective, dubbed *Cybersecurity Dynamics*. One pillar underlying the Cybersecurity Dynamics framework is dubbed *first-principle models*, which aims to model and characterize the evolution of the global cyber security state of a network in question. The evolution is caused by cyber attack-defense interactions. Several families of first-principle Cybersecurity Dynamics models have been investigated, including preventive and reactive defense dynamics [30, 31, 52, 72, 73, 78, 80, 129, 133, 138, 140, 147], proactive defense dynamics [58], adaptive defense dynamics [43, 139], and active defense dynamics [137, 146]. These theoretical studies led to many deep insights into the laws that govern the evolution of the global cyber security state.

The present dissertation focuses on the first step of sophisticated cyber attacks, namely *cyber reconnaissance*, which is the process of gathering information on the victims, such as the hosted services, open port, software versions, and operating system [26]. Additionally, this dissertation focuses on understanding and characterizing cyber attacks reconnaissance behaviors.

1.1 Dissertation Research Motivation and Problem Statement

Despite the clear importance of understanding cyber reconnaissance behaviors, there are no systematic studies on this topic. For example, we are not aware of any previous PhD dissertation that focuses on this topic. The main goal of this dissertation is to investigate cyber reconnaissance behaviors, such that the findings can be used in effective cyber defense practice, such as the following aspects:

- *Enabling quantitative understanding of cyber threat situation.* The defender can use cyber reconnaissance behaviors to cluster the extremely large cyber attackers into a smaller number of attackers families, enabling more effective analysis of cyber threat situation in real time. It may not be feasible to cope with the large number of cyber attackers individually.

- *Enabling proactive cyber defense.* It is important to understand how to structure cyber reconnaissance data into time series (e.g., at what time resolution) such that the time series exhibit the desirable statistical properties (e.g., Long-Range Dependence or LRD) that can be leveraged to forecast cyber reconnaissance behaviors [143, 145]. Being able to forecast incoming cyber threats is one of the fundamental capabilities for enabling effective cyber defense.

- *Enabling deceptive defense.* Once the defender understood the families of cyber attackers in the wild, the defender can selectively and intentionally let some attackers of interest get through so as to monitor their behaviors in the next stages of their sophisticated attack process. This defense approach allows the defender to discover new attack strategies and tactics. This capability is critical to reduce the gap between the attacker and the defender.

While achieving the ultimate goals mentioned above, the present dissertation is also focused on answering the following research problems:

- What are the relevant levels of abstractions that we should use to deepen our understanding of cyber reconnaissance behaviors?

- What are the representations of cyber reconnaissance data that can enable us to achieve the ultimate goals mentioned above?

1.2 Dissertation Contributions

The *conceptual* contribution of the dissertation is to formulate a systematic and quantitative way of understanding cyber reconnaissance behaviors. As highlighted in Figures 1.2, we propose using three levels of abstractions to structure cyber reconnaissance data, including a *macroscopic* level, a *mesoscopic* level, and a *microscopic* level. This treatment is inspired by the aforementioned Cybersecurity Dynamics framework [134, 136].

The *technical* contribution of the dissertation is to investigate a systematic framework or methodology for exploring each of the three levels of abstractions mentioned above, leading to the following three contributions:

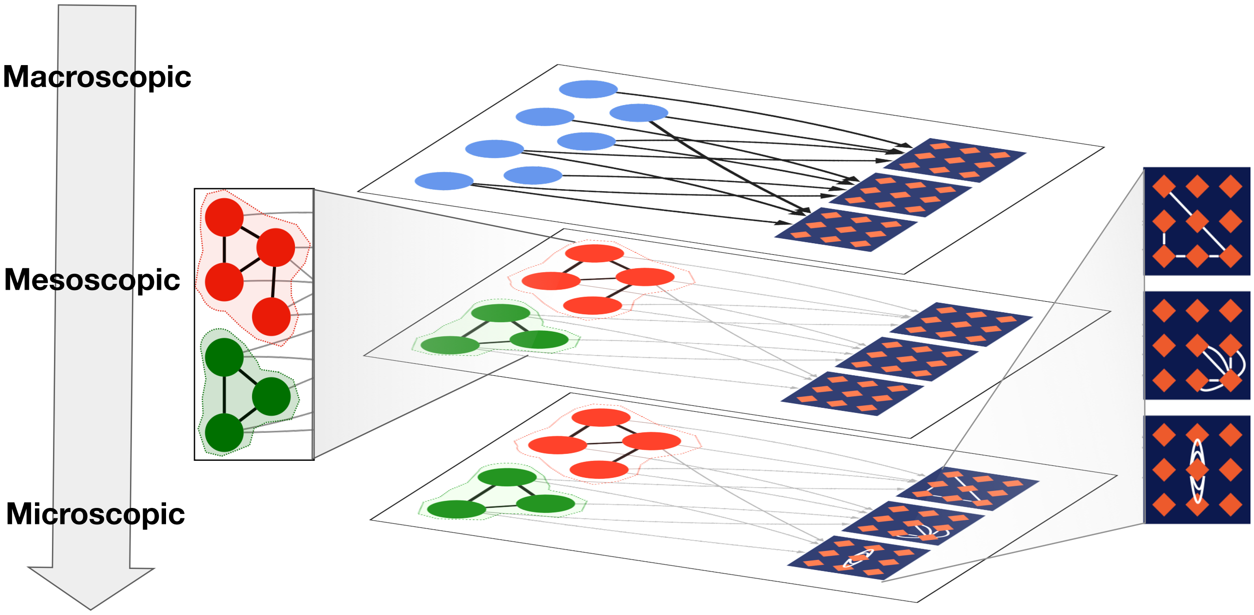


Figure 1.2: The three levels of abstractions we propose for quantitatively modeling cyber reconnaissance behaviors.

- At the *macroscopic* level, we propose a novel, graph-theoretic abstraction, dubbed the *evolution of attacker-victim relation graphs*. Specifically, we use time series of attack-victim relation graphs to describe the reconnaissance behaviors of cyber attackers. Our contribution describes the similarity between two bipartite graphs at adjacent time windows of a certain time resolution (e.g., per second vs. per minute). We explore the various kinds of methods that can be adopted to characterize the evolution of such similarities. Furthermore, we provide a case study that focuses on an important problem: how many time resolutions have to be considered in order to obtain a comprehensive understanding of the evolution of the attack-victim bipartite graphs? This problem is important because under different time resolutions, the time series may exhibit different temporal characteristics, all of which may be important.

- At the *mesoscopic* level, we propose clustering cyber attackers according to their reconnaissance behaviors via graph-based two-stage community detection. We conduct a case study on the evolution of high-activity attackers, the evolution of persistent attackers, and the evolution of the attackers families.

- At the *microscopic* level, we introduce the novel notion of *attacker reconnaissance trajectories hierarchy trees* to represent cyber attackers' reconnaissance behaviors. This notion allows us to cluster cyber attackers according to their microscopic-level tactics. We develop a top-down approach to find the hierarchy of attacker reconnaissance-trajectories and implement a graph-theoretic clustering to find the families of attackers given the similarity between the attackers trajectories hierarchy trees.

It is worth mentioning that the present dissertation falls into the intersection of the fields known as Cybersecurity Data Analytics and Cybersecurity Metrics, which are the two other pillars (other than First-Pinciple modeling) of the Cybersecrurity Dynamics framework [134, 136]. Cybersecurity Data Analytics aims to take full advantage of a given cyber security dataset by statistically using statistical, machine learning, and artificial intelligence methods [32, 33, 47, 48, 53, 70, 71, 75–77, 92, 94, 95, 113, 114, 126–128, 131, 142–145]. Cybersecurity Metrics aims to systematically define and measure the important security, resilience, and agility metrics [29–31, 34, 35, 45, 82, 84, 85, 91, 104].

1.3 Dissertation Organization

Chapter 2 presents a *macroscopic*-level characterization of cyber attack reconnaissance behaviors. Chapter 3 presents a *mesoscopic*-level characterization of cyber attack reconnaissance behaviors. Chapter 4 presents a *microscopic*-level characterization of cyber attack reconnaissance behaviors. Chapter 5 concludes the dissertation.

CHAPTER 2: CHARACTERIZING THE EVOLUTION OF CYBER ATTACKER-VICTIM RELATION GRAPHS

This chapter was published at the *IEEE Military Communications Conference* in 2018 (MILCOM 2018) with co-authors: Kristin M Schweitzer, Raymond M Bateman, and Shouhuai Xu.

Chapter Abstract

Understanding and characterizing the reconnaissance behaviors of cyber attackers is an important problem that has yet to be tackled. As a first step towards approaching this problem, in this chapter we propose a novel, graph-theoretic abstraction, dubbed the *evolution of attacker-victim relation graphs*, for characterizing cyber attackers' reconnaissance behaviors. The framework is focused on describing the similarity between two graphs at adjacent time windows of a certain resolution (e.g., per second vs. per minute). We also conduct a case study focusing on the number of time resolutions that need to be considered in order to obtain a comprehensive understanding of the evolution of attack-victim relation graphs.

2.1 Introduction

Understanding, characterizing, and even predicting the reconnaissance behaviors of cyber attackers is an important problem that has yet to be tackled. This problem is important because it can help defenders detect and recognize different reconnaissance behaviors, and therefore help defenders respond to anticipated attacks effectively (e.g., using deception to force an attacker to expose its intent rather than simply dropping the attacker's traffic). Despite its clear importance, this problem has not been investigated in the literature.

In this paper, we make a first step towards tackling this problem, by proposing a novel, graph-theoretic abstraction, dubbed the *evolution of attacker-victim relation graphs*. In this framework, we used a time series of attack-victim relation graphs to describe the reconnaissance behaviors of cyber attackers. Given such a time series, the framework is centered at describing the similarity between two bipartite graphs at adjacent time windows of a certain time resolution (e.g., per second vs. per minute). We explored the various kinds of methods that can be adopted to characterize the evolution of such similarities. We also conducted a case study using a real-world dataset of honeypot-captured time series of cyber attacker-victim relation graphs, which are naturally modeled by bipartite graphs. The case study focuses on an important problem: how many time resolutions have to be considered in order to obtain a comprehensive understanding of the evolution of the attack-victim bipartite graphs? This problem is important because under different time resolutions, the time series may exhibit different temporal characteristics, all of which may be of interest.

Our contributions. We make the following contributions. First, we initiate the study of understanding and characterizing cyber attackers' reconnaissance behaviors via the time series of attack-victim relation graphs. This graph-theoretic abstraction allows us to formulate various questions that can be answered by leveraging a range of existing tools. Second, in order to characterize the evolution of the attacker-victim relation graphs, we propose using features to represent these graphs and using similarities between such graphs corresponding to different time windows. Moreover, we define the notions of *effective features* (i.e., features that may or may not be useful in characterizing the evolution of attacker-victim bipartite graphs) and *robust features* (i.e., features that are effective across time resolutions). Third, we use a dataset that was collected at a honeypot to conduct a case study to investigate the time resolutions that need to be considered in order to characterize the evolution of the attacker-victim bipartite graphs as comprehensive as possible. Experimental results show that only a couple of time resolutions need to be considered.

Paper outline. Section 2.2 presents the framework. Section 2.3 describes the case study and results. Section 2.4 reviews prior studies. Section 2.5 concludes the paper.

2.2 The Framework

Figure 2.1 highlights the framework, which consists of five components: data collection and preprocessing, graph-theoretic representations, lower-dimension representations (with or without us-

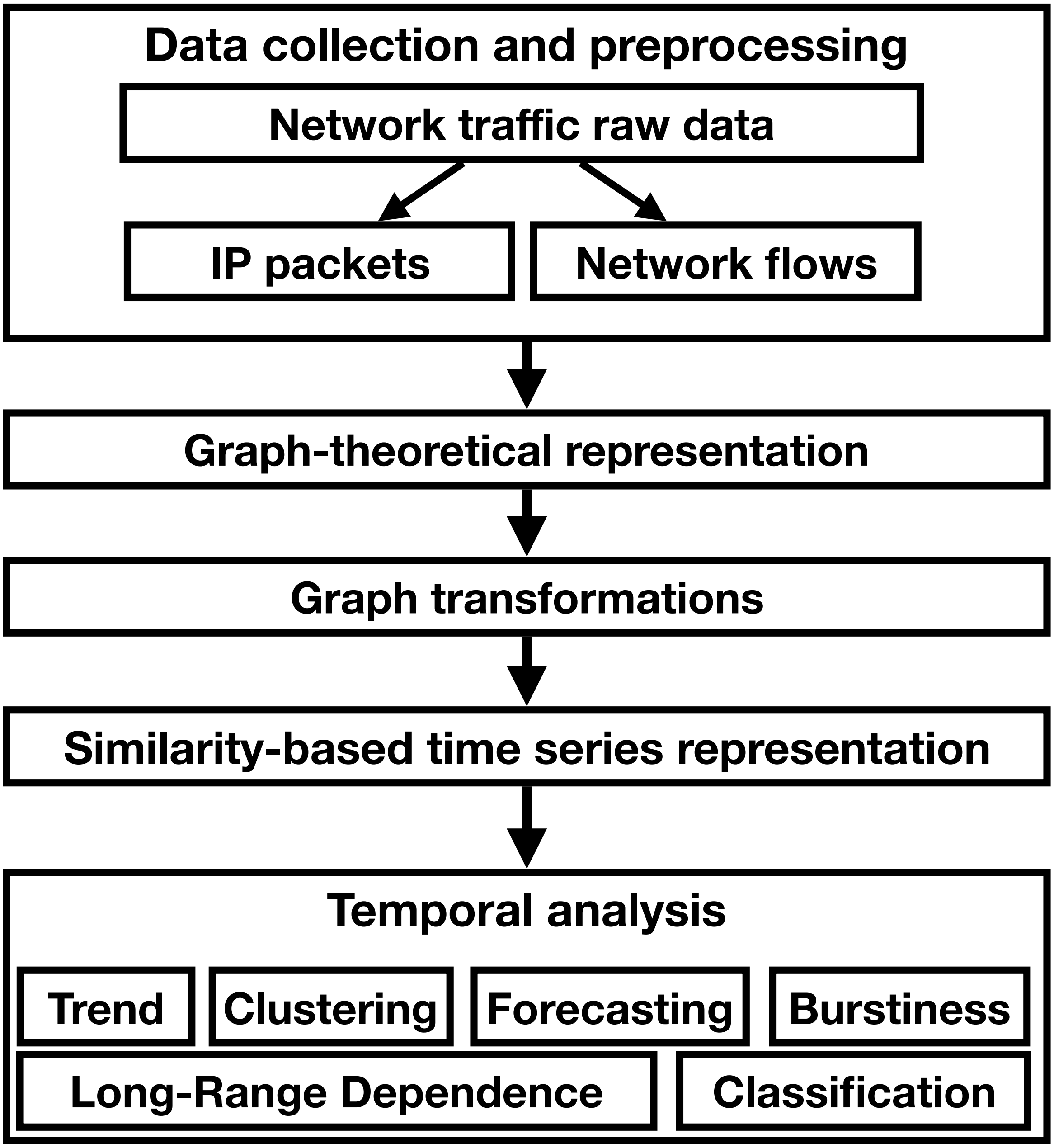


Figure 2.1: The framework

ing embedding), similarity-based time series representations, and temporal analysis.

2.2.1 Data Collection and Preprocessing

In general, network data are often collected in the raw Packet Capture Data (PCAP) format, which may be turned into IP packets or flows. A flow contains one or multiple packets and it is a common practice to treat each flow as an attack (see, e.g., [96, 143, 145]). A flow is a tuple of five fields: *source IP address*, *destination IP address*, *source port*, *destination port*, and *protocol*. Each flow has a start time and an end time. For flow-based analysis, we need to specify two extra parameters: the *idle time* and *lifetime* of a flow. The *idle time* is used to terminate a flow when the communication between the source and destination has become idle (i.e., no packets exchanges) for longer than the idle time parameter. On the other hand, a flow is terminated and a new flow is created when the communication between the source and the destination exceeds the *lifetime* parameter. The time resolution parameter, denoted by Δ, is selected. The life-cycle of a dataset is divided into intervals $I_0, I_1, \ldots$, where $I_i = [t_i, t_i + \Delta)$, such that a flow with a start time s belongs to time interval I_i if and only if $t_i \leq s < t_i + \Delta$. Packet-based preprocessing is similar except that there is no need to assemble packet(s) into flows.

2.2.2 Graph-Theoretic Representations

After dividing the data life-cycle into time windows of length Δ, we obtain a time series of attacker-victim relation graphs as follows. For each time interval I_i, we transform the flows in interval I_i to a directed and weighted graph, denoted by $G_i = (A_i, V_i, E_i, W_i)$, where A_i is the set of attackers (i.e., attacker IP addresses), V_i is the set of victims (i.e., victim IP addresses), E_i is the set of edges indicating the existence of IP packet(s) or flow(s) from an attacker to a victim, and $W_i : E_i \rightarrow \mathbb{I}^+$ is the weight function (i.e., the number of probes from a particular attacker to a particular victim in interval I_i). In many, but not all cases, the attacker-victim relation graphs are bipartite graphs.

2.2.3 Graph Transformations

In order to analyze the time series of graphs, we often need to transform to lower-dimension representations. For this purpose, there are two general approaches.

- Embedding: This approach is to embed G_i into another space. For example, we can embed attacker nodes into a smaller graph, where an edge in the embedded graph reflects how similar a pair of attackers are. Alternatively, we can embed victim nodes into a smaller graph, where an edge reflects the similarity between a pair of victims in terms of the common attackers against them [25, 67]. Yet another alternative is to embed the time series into a tensor of adjacency matrices of the G_i's. Let us denote the embedded graph of G_i by $\mathsf{Embed}(G_i)$.

- Non-embedding: This approach represents a graph by using any of the following data structures: the graph adjacency matrix, the graph adjacency list, or the graph edge list. Moreover, a feature vector may be defined to represent the graphs.

2.2.4 Similarity-based Time Series Representations

Regardless of the specific graph-transformation method, we can define some kinds of *similarities* to describe the relation between the embedded graphs $\mathsf{Embed}(G_i)$ and $\mathsf{Embed}(G_{i+1})$, or between the non-embedded graphs G_i and G_{i+1} or their feature representations. This comparison leads to a new *time series of similarities*, which is the target for actual analysis in the next step.

2.2.5 Temporal analysis

Given the time series of similarities between two consecutive embedded graphs $\mathsf{Embed}(G_i)$ and $\mathsf{Embed}(G_{i+1})$ or non-embedded graphs G_i and G_{i+1}, we can analyze the temporal characteristics to understand the evolution of the time series of the attacker-victim relation graphs. Some examples of temporal analysis are: trend analysis, long-range dependence (LRD), anomaly detection, forecasting, burstiness analysis, classification, and clustering.

2.3 Case Study and Results

2.3.1 Case Study

Data Collection and Preprocessing

We analyzed a dataset collected at a honeypot, by transforming it to flows.

Dataset. Figure 2.2 illustrates the kind of data captured by honeypots, where each dot represents an IP address. Specifically, victims are the honeypot IP addresses that can be attacked by IP addresses outside of the honeypot (i.e., attackers). At a particular moment of observation, some attackers are active (i.e., if they are attacking some victims) and some victims are active (i.e. if they are being under attack) while others no. Since the honeypot offers no legitimate Internet services, the traffic is considered malicious (see, e.g., [96, 143, 145]).

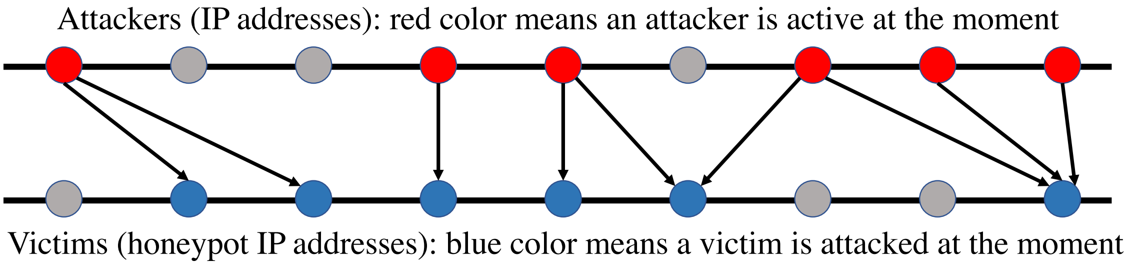

Attackers (IP addresses): red color means an attacker is active at the moment
Victims (honeypot IP addresses): blue color means a victim is attacked at the moment

Figure 2.2: A snapshot of attacks at a moment in time, the active attacker in red, the active victims in blue.

The network traffic was collected by a honeypot of 1024 IP addresses from 2/2/2014 to 5/9/2014. The honeypot is a low-interaction honeypot based on the *Honeyd* [100] and *Nepenthes* [24] programs. The dataset contained 6,403 raw packet captures (PCAP) files for a total of 597GB of data.

Converting network traffic into network flows. Since the honeypot outbound traffic is limited to five minutes (for institutional regulation), we pre-processed the PCAP data into IPFIX network flows using an idle time of 60 seconds and lifetime of 300 seconds. For this conversion, we used the Yet Another Flowmeter (YAF) and the *super_mediator* tools of the Computer Emergency Response Taskforce (CERT) [62]. The dataset led to 92,477,692 TCP flows (or attacks).

Table 2.1: Simple statistics and standard deviation of flow duration, # of packets per flow, and # of bytes per flow.

	min	25%	50%	μ	75%	max	σ
flow duration	0.001	0.001	0.003	7.5	0.6	300	23.3
# of packets	1.0	1.0	1.0	2.0	2.0	550	2.5
# of bytes	40	48	52	125.3	113.0	47125	266.6

Table 2.1 presents the simple statistics and standard deviation (σ) of the flow duration in seconds (i.e., the interval between the time at which a flow starts and the time at which a flow ends), the number of packets per flow, and the number of bytes per flow (i.e., the length of the content in a flow). We observed that many flows contain only a single packet, suggesting scanning activities or initial reconnaissance efforts.

Graph-Theoretical Representation

Figure 2.2 suggests that the dataset can be represented by the evolution of bipartite graphs. To generate a time series of bipartite graphs, we need to select the time window unit denoted by Δ as shown in the framework. In order to see the impact of time resolution, we consider a range of Δ's, namely $\Delta \in \{0.5, 1, 2, 9, 12, 30, 60, 90, 120, 180, 360, 720\}$ (unit: minute). Then, the dataset is divided into intervals $I_0, I_1, \ldots$, where $I_i = [t_i, t_i + \Delta)$. For each time interval I_i, we transform the flows in interval I_i to a directed and weighted bipartite graph, namely $G_i = (A_i, V_i, E_i, W_i)$ as shown in the framework, where A_i is the set of attackers (i.e., attacker IP addresses), V_i is the set of victims (i.e., honeypot IP addresses), E_i is the set of edges indicating the existence of one or more flows from an attacker to a victim, and $W_i : E_i \rightarrow \mathbb{I}^+$ is the weights on the edges in E_i (i.e., the number of flows from a attacker to a victim in interval I_i).

Figure 2.3 plots G_{59}, G_{60}, and G_{61} with $\Delta = 12$ minutes. We observe that an attacker in G_{59} launched 495 attacks against 495 different victims with destination port # 22, indicating that the attacker is trying to find a SSH server for possibly launching password brute-forcing attacks.

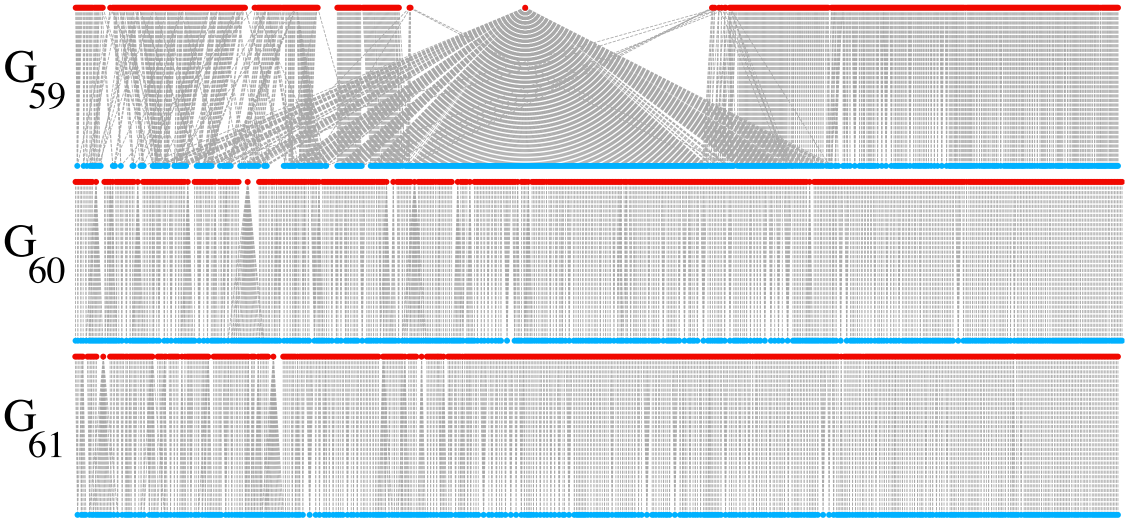


Figure 2.3: Bipartite graphs G_{59}, G_{60}, and G_{61} with $\Delta = 12$.

Graph Transformations

In this chapter, we focus on transforming attacker-victim bipartite graphs to their feature vector representations, meaning that G_i is represented by a feature vector F_i. Given that some features may or may not be *effective* for the purpose of characterizing the evolution of the attacker-victim relation graphs, we define the following concepts:

Definition 1. *(effective feature) Corresponding to a given time resolution Δ, a feature is effective if (i) its standard deviation over time is significantly greater than zero, meaning that its values substantially change over time and therefore the feature offers a discrimination power; and (ii) it does not linearly depend on other features (i.e., not redundant).*

Definition 2. *(robust feature) A feature is robust if it is effective with respect to any time resolution Δ.*

Table 2.2 describes the 28 features we define, dubbed $f_1, \ldots, f_{28}$. In addition to some self-explaining features, we also consider the *weak connected component* (WCC) feature, which is defined as the maximal *connected* subgraph $c_j = (\alpha_j, \nu_j, \epsilon_j)$ such that $c_j \subseteq G_i$, $\alpha_j \subseteq A_i$, $\nu_j \subseteq V_i$, $\epsilon_j \subseteq E_i$, where for any two c_j and c_h the following holds true: $\alpha_j \bigcup \alpha_h = \emptyset$, $\nu_j \bigcup \nu_h = \emptyset$ and $\epsilon_j \bigcup \epsilon_h = \emptyset$. We denote the set of WCC in G_i by $C_i = \{c_1, c_2, \ldots, c_m\}$, where the WCC size is defined as $z_k = |c_k|$ for $1 \leq k \leq m$ and let $Z_i = \{z_1, z_2, \dots z_m\}$ be the set of WCC sizes.

Table 2.2: Features for the feature vector embedding $\mathsf{Embed}(G_i) = F_i$ representing bipartite graph G_i

f_1	Number of attackers, namely $\lvert A_i \rvert$
f_2	Number of victims, namely $\lvert V_i \rvert$
f_3	Number of edges, namely $\lvert E_i \rvert$
f_4	Number of WCC, namely $\lvert C_i \rvert$
f_{5-10}	Statistical summary of Z_i, or $\mathsf{stats}(Z_i)$
f_{11-16}	Statistical summary of W_i, or $\mathsf{stats}(W_i)$
f_{17-22}	Statistical summary of D_{out}, or $\mathsf{stats}(D_{out})$
f_{23-28}	Statistical summary of D_{in}, or $\mathsf{stats}(D_{in})$

We consider the feature of *weighted out-degree* for attackers. This feature reflects the number of probes launched by the attacker within time Δ. For attackers in A_i, the set of attackers' weighted out-degrees are denoted by $D_{out} = \{\sum_{v \in V_i} W_i(a, v) | a \in A_i\}$. Similarly, we consider the feature of *weighted in-degree* of victims in V_i, representing the number of probes against a victim, denoted by $D_{in} = \{\sum_{a \in A_i} W_i(a, v) | v \in V_i\}$.

Since some features are not *effective* for characterizing the evolution of bipartite graphs, we should remove them. For this purpose, we use the Classification and Training (caret) Package in R [69], which has the following steps:

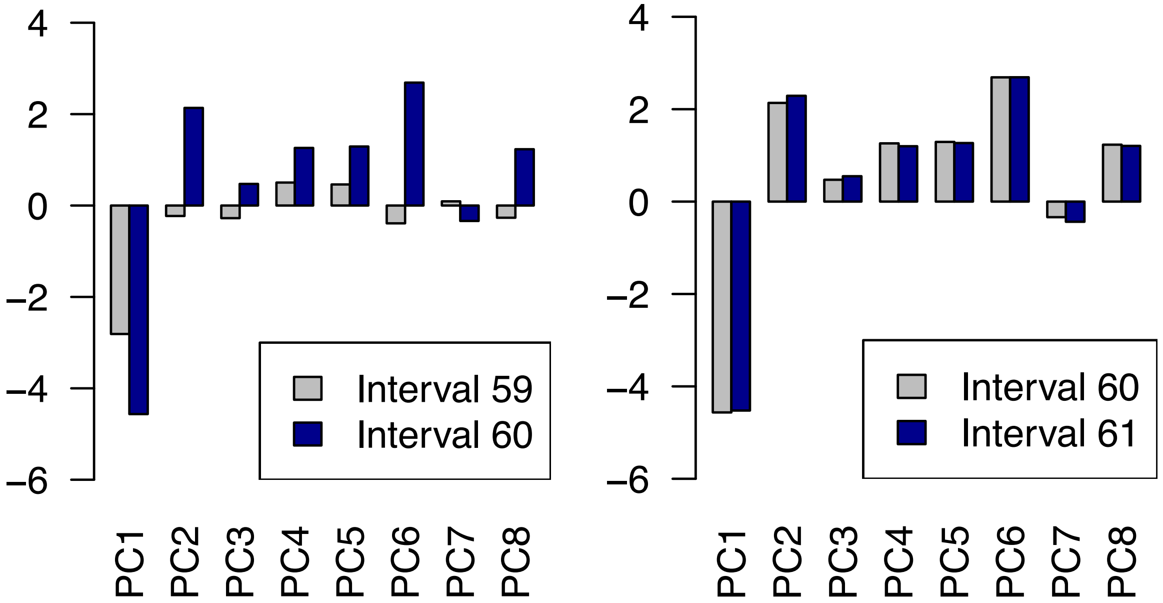
1. Identify and remove 0-variance features.

2. Find and remove linearly dependent features.

3. Apply a Box and Cox transformation to fix the skewness of the remaining features.

4. Normalize the remaining feature vector and perform a Principal Component Analysis (PCA) to reduce the size of the remaining feature vector.

Figure 2.4 plots the refined feature representation of the aforementioned G_{59}, G_{60}, and G_{61}. Figure 2.4a shows that G_{59} and G_{60} are very different, while Figure 2.4b shows that G_{60} and G_{61} are very similar. This is consistent with a visual examination of Figure 2.3.



(a) F_{59} vs F_{60}.

(b) F_{60} vs F_{61}

Figure 2.4: Refined feature vectors of G_{59}, G_{60} and G_{61}.

Similarity-based representations of pairs of bipartite graphs

In order to analyze the evolution of the similarity between an adjacent pair of bipartite graphs, we define:

Definition 3. *(similarity) The similarity between a pair of bipartite graphs, G_i and G_{i+1}, is defined as:*

$$S(G_i, G_{i+1}) = \frac{1}{1 + \delta(F_i, F_{i+1})}, \tag{2.1}$$

where $\delta(F_i, F_{i+1})$ is the Euclidean distance between the feature vectors, which are also assured to have the same dimensions after the PCA treatment. Note that $S(G_i, G_{i+1}) \in [0, 1]$.

Temporal Analysis

We conduct two kinds of temporal analysis for the similarity time series $\{S(G_i, G_{i+1})\}_{i=0,1,...}$. The first analysis is to decompose its trend, seasonality and residual. This can be done using the *stl* function from the *netlib* package in R. The second analysis is to analyze whether there is a long-range dependence (LRD). A time series $\{X_t : t \geq 0\}$ is said to possess LRD [115] if the rate of the auto-correlation function decays slowly. Formally, if

$$r(h) = Cor(X_t, X_{t+h}) \sim h^{-\beta} L(h), \;\; h \to \infty \tag{2.2}$$

for $0 < \beta < 1$, where h is the lag and $L(\cdot)$ is a slowly varying function such that $\lim_{x\to\infty} \frac{L(ix)}{L(x)} = 1$ for all $i > 0$. The degree of LRD can be quantified by the Hurst parameter [105], which can be estimated using the *fArma* package in R [124],

2.3.2 Results

Bipartite Graph Feature Analysis

Table 2.3 lists the features that are kept by the PCA, namely those marked with a ✓, with respect to different time resolution Δ's (i.e., the columns). We observe that the minimum edge weight (f_{11}), the 25% percentile edge weight (f_{12}), the 75% percentile edge weights (f_{15}), the minimum out-degree (f_{17}), the first quantile out-degree (f_{18}), and the third quantile out-degree (f_{21}) are *ineffective* features. This is because these features almost always have 0-variance regardless of the Δ, which can be attributed to the following fact: (i) for 75% of the bipartite graphs, 75% of the edges correspond to less than four attacks; and (ii) 25% of the attackers launch a single attacks. These observations support that the attacker-victim interactions in the dataset correspond to reconnaissance efforts or scan activities. In contrast, the number of attackers (f_1), the number of edges (f_3), the average size of connected components (f_{14}), the median out-degree (f_{19}), the average out-degree (f_{20}), the maximum out-degree (f_{22}), the median in-degree (f_{25}), the average in-degree (f_{26}), and the maximum in-degree (f_{28}) are *robust* features. This is because, according to Definition 2, these features always have a non-zero variance and are not linearly dependent on any of the other features, regardless of the Δ.

For $\Delta \in \{30, 60, 90, 120, 180, 360, 720\}$, the time window is large enough such that (almost) every victim is attacked at least once in a time window, explaining why the number of victims (f_2) is removed (i.e., it does not provide any discrimination power). This also causes the removal of

Table 2.3: Features that are kept (marked by ✓) vs. removed with respect to different Δ's. Note that the outcome is the same for $\Delta = 120, 180, 360, 720$.

Δ	0.5	1	2	9	12	30	60	90	120, 180, 360, 720
f_1	✓	✓	✓	✓	✓	✓	✓	✓	✓
f_2	✓	✓	✓	✓	✓				
f_3	✓	✓	✓	✓	✓	✓	✓	✓	✓
f_4	✓	✓	✓	✓	✓				
f_5						✓	✓	✓	✓
f_6						✓	✓	✓	
f_7					✓	✓	✓	✓	
f_8	✓	✓	✓	✓	✓	✓	✓	✓	
f_9				✓	✓	✓	✓	✓	
f_{10}	✓	✓	✓	✓	✓	✓	✓	✓	
f_{11}, f_{12}									
f_{13}	✓	✓	✓	✓	✓	✓			
f_{14}	✓	✓	✓	✓	✓	✓	✓	✓	✓
f_{15}									
f_{16}	✓	✓	✓	✓	✓			✓	✓
f_{17}, f_{18}									
f_{19}, f_{20}	✓	✓	✓	✓	✓	✓	✓	✓	✓
f_{21}									
f_{22}	✓	✓	✓	✓	✓	✓	✓	✓	✓
f_{23}, f_{24}					✓	✓	✓	✓	✓
f_{25}, f_{26}	✓	✓	✓	✓	✓	✓	✓	✓	✓
f_{27}				✓	✓	✓	✓	✓	✓
f_{28}	✓	✓	✓	✓	✓	✓	✓	✓	✓

the number of WCC (f_4) because it is the same for those Δ's. For $\Delta \in \{0.5, 1, 2, 9, 12\}$, the time window is small enough such that the minimum size of WCC (f_5) is always 1, the 25% percentile of the WCC size is always 1 (f_6), and the minimum in-degree (f_{23}) is alway 1, explaining why these features are removed.

Summarizing the preceding discussion, we conclude with:

Insight 4. *Under different time resolution (i.e., time windows), different sets of features should be used to characterize the evolution of the attack-victim bipartite graphs.*

Evolution Trends Analysis

Figure 2.5 shows the trend of similarity scores with respect to different Δ's. We observe that (i) the trends for $\Delta \in \{0.5, 1, 2, 9, 12\}$ are very similar, (ii) the trends for $\Delta \in \{0.5, 1, 2, 9, 12\}$ are quite different from the trends for $\Delta \in \{30, 60, 90, 120, 180, 360, 720\}$, and (iii) the trends for $\Delta \in \{30, 60, 90, 120, 180, 360, 720\}$ are very similar.

Figure 2.6 presents the correlation matrix between the trends with different Δ's, and confirms that the trends within each group of Δ's are highly correlated with each other, but different groups are little correlated with each other.

Summarizing the preceding discussion, we draw:

Insight 5. *In order to fully characterize the evolution of the attack-victim bipartite graphs, the defender only needs to consider a couple of time resolutions: a small time window (e.g., $\Delta = 12$ minutes) and a large time window ($\Delta = 90$ minutes), where the specific window size may depend on the size of the honeypot.*

LRD Analysis

Table 2.4 presents three Hurt parameters: the average variant method (RS), the difference aggregate variance method (AGV), and the Peng's method (Peng), which are obtained by using estimator *fArma* with respect to different Δ's. We observe that the time series exhibit LRD, except that the Hurst parameter based on the RS method is 0.4, but the Hurst parameters estimated by the other

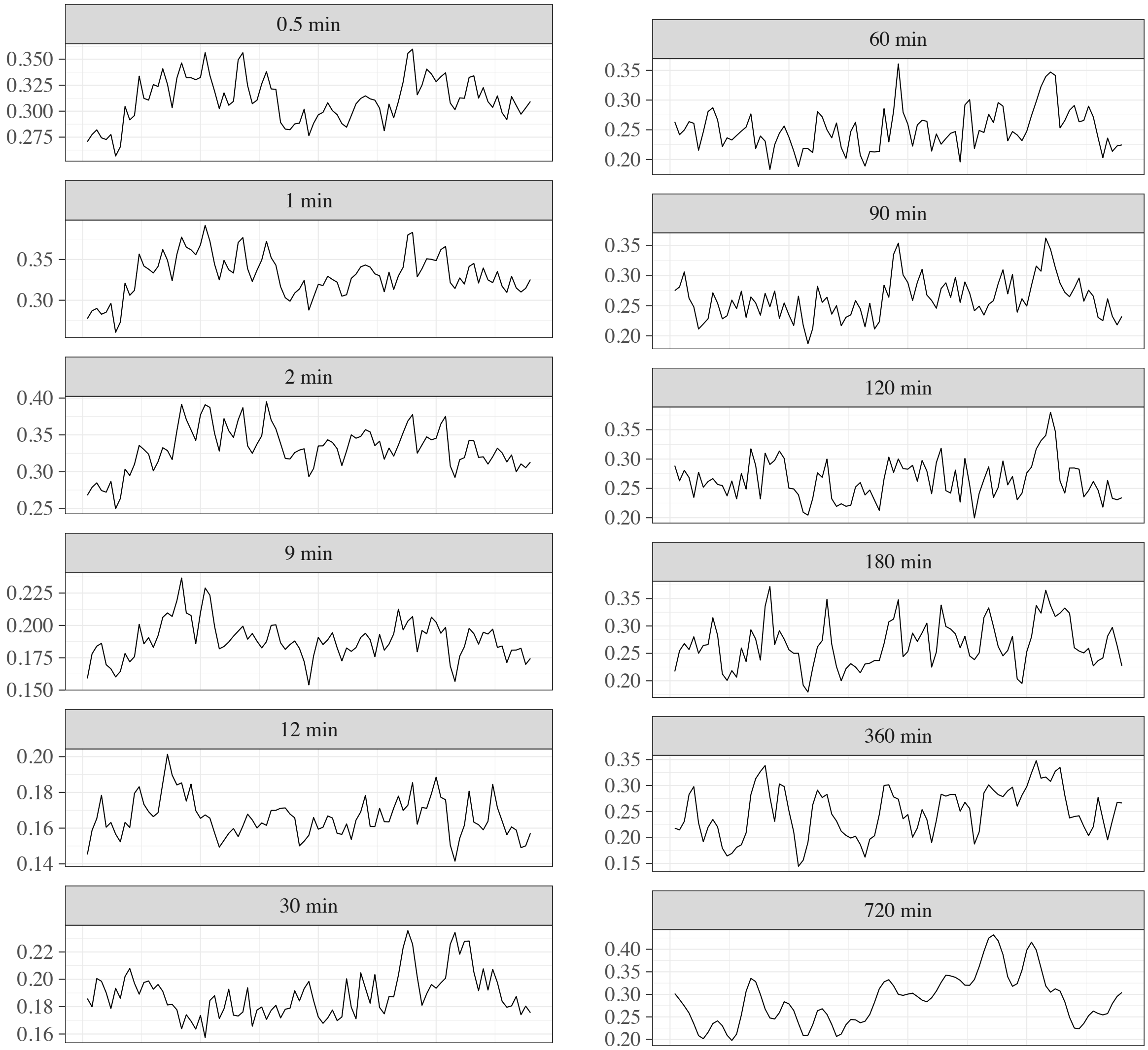
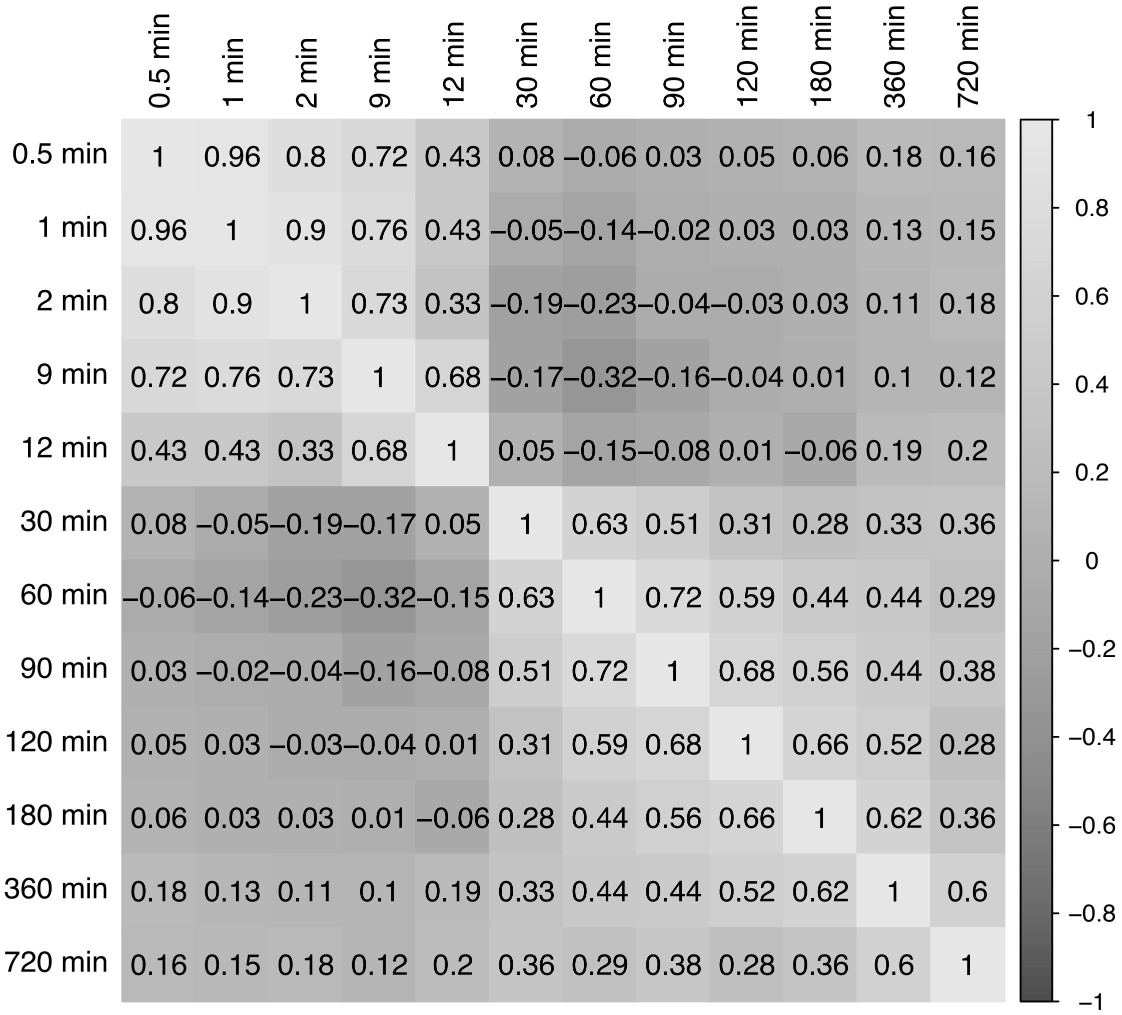


Figure 2.5: Time series trend analysis with different Δ's, which are indicated on the top of each sub-figure, where the x-axis represents time and the y-axis is the trend of similarity scores $S(F_i, F_{i+1})$.



	0.5 min	1 min	2 min	9 min	12 min	30 min	60 min	90 min	120 min	180 min	360 min	720 min
0.5 min	1	0.96	0.8	0.72	0.43	0.08	−0.06	0.03	0.05	0.06	0.18	0.16
1 min	0.96	1	0.9	0.76	0.43	−0.05	−0.14	−0.02	0.03	0.03	0.13	0.15
2 min	0.8	0.9	1	0.73	0.33	−0.19	−0.23	−0.04	−0.03	0.03	0.11	0.18
9 min	0.72	0.76	0.73	1	0.68	−0.17	−0.32	−0.16	−0.04	0.01	0.1	0.12
12 min	0.43	0.43	0.33	0.68	1	0.05	−0.15	−0.08	0.01	−0.06	0.19	0.2
30 min	0.08	−0.05	−0.19	−0.17	0.05	1	0.63	0.51	0.31	0.28	0.33	0.36
60 min	−0.06	−0.14	−0.23	−0.32	−0.15	0.63	1	0.72	0.59	0.44	0.44	0.29
90 min	0.03	−0.02	−0.04	−0.16	−0.08	0.51	0.72	1	0.68	0.56	0.44	0.38
120 min	0.05	0.03	−0.03	−0.04	0.01	0.31	0.59	0.68	1	0.66	0.52	0.28
180 min	0.06	0.03	0.03	0.01	−0.06	0.28	0.44	0.56	0.66	1	0.62	0.36
360 min	0.18	0.13	0.11	0.1	0.19	0.33	0.44	0.44	0.52	0.62	1	0.6
720 min	0.16	0.15	0.18	0.12	0.2	0.36	0.29	0.38	0.28	0.36	0.6	1

Figure 2.6: Correlation matrix of daily frequency trends.

two methods are all greater than 0.5, indicating LRD. We further use the Smoothly Varying Trend test [101] to test whether the times series exhibit *spurious* LRD or not. Table 2.5 summarizes the results, where $Z_1 > 1.517$ and $Z_2 > 1.426$ means the null hypothesis H_0 is true (i.e., the time series exhibits spurious LRD). In summary, we draw:

Table 2.4: The Hurst parameters with different Δ's.

Δ	0.5	1	2	9	12	30
RS	0.70	0.71	0.70	0.61	0.56	0.64
AGV	0.71	0.74	0.77	0.77	0.77	0.91
Peng	0.62	0.61	0.60	0.56	0.56	0.53

Δ	60	90	120	180	360	720
RS	**0.40**	0.67	0.66	0.71	0.94	0.68
AGV	0.80	0.82	0.97	0.84	0.75	0.69
Peng	0.60	0.60	0.59	0.61	0.65	0.55

Table 2.5: Test results for spurious LRD with different Δ's.

Δ	$\widehat{H}$	Z_1	H_0	Z_2	H_a
0.5	**0.743**	**6.540**	**true**	**6.344**	**true**
1	**0.707**	**5.297**	**true**	**4.680**	**true**
2	**0.682**	**4.625**	**true**	**4.178**	**true**
9	0.820	0.467	false	0.380	false
12	0.793	0.743	false	0.448	false
30	0.542	1.423	false	1.086	false
60	**0.599**	**1.693**	**true**	**1.693**	**true**
90	0.606	1.327	false	1.326	false
120	0.602	0.555	false	0.555	false
180	0.566	1.325	false	1.325	false
360	0.714	0.486	false	0.486	false
720	0.548	1.239	false	1.124	false

Insight 6. *The time window size affects whether the time series exhibits LRD. Because LRD implies that a time series can be accurately predicted [33, 93, 96, 143, 145], the defender needs to be conscious in selecting Δ.*

2.4 Related Work

The present study falls into the field of cybersecurity data analytics, which is an indispensable pillar in the broader framework of Cybersecurity Dynamics [33,91,93,96,130,134,135,143,145]. In contrast to previous studies on cybersecurity data analytics that focus on univariate [33,93,132,143–145] or multivariate time series [96,130], the framework focuses on analyzing the evolution of the attacker-victim relation graphs, which are bipartite graphs in the real-world dataset. Honeypot-captured datasets have analyzed from other perspectives, such as: visualizing the ports that are observed in honeypot datasets [61]; characterizing attack probing activities [74]; clustering attacks [19–21,39]; modeling attack inter-arrival times [18,63]; predicting/forecasting attack rates [33,93,96,143,145]; detecting cyber attacks (e.g., malware, botnets) [22,44,51,68,79,97–99].

Two other kinds of datasets have been analyzed in the literature as well, although none of these studies analyzed the evolution of the attacker-victim (bipartite) graphs. On one hand, there have been studies on analyzing blackhole-captured cyber attacks (e.g., [88,125,130,144]), but not on the evolution of the attack-victim relation graphs. On the other hand, datasets collected at enterprise networks (i.e., neither honeypots nor telescopes) have been analyzed in [23,59].

2.5 Conclusion

We presented a framework for characterizing the evolution of attacker-victim relation graphs, as a first step towards understanding and characterizing cyber attackers' reconnaissance behaviors. The framework is centered at describing the similarity between two bipartite graphs at adjacent time windows. We also conducted a case study with emphasis on identifying the number of time resolutions to characterize the evolution of the evolution of attacker-victim relation graphs.

The framework represents our first step towards a thorough understanding of cyber attack reconnaissance behaviors.

Acknowledgement. This work was supported in part by ARL grant #W911NF-17-2-0127.

CHAPTER 3: CHARACTERIZING CYBER ATTACK RECONNAISSANCE BEHAVIORS

Chapter Abstract

Cyber attack reconnaissance is the initial step preluding cyber attacks. Understanding and characterizing cyber attack reconnaissance behaviors has a potential in helping and guiding defenders in preparing and orchestrating their defense. In this chapter we investigate cyber attack reconnaissance behaviors through a clustering analysis. We propose clustering cyber attackers based on their reconnaissance behaviors over time. This prompts us to come up with a novel abstraction, dubbed *multi-resolution clustering*, to characterize the evolution of attackers' reconnaissance behaviors in adjacent time windows. This further allow us to characterize the evolution of persistent attackers' reconnaissance behaviors over multiple adjacent time windows. Moreover, we present a case study on identifying suitable parameter combinations for these characterization studies.

3.1 Introduction

Understanding and clustering cyber attackers based on their reconnaissance behaviors is an essential problem. This problem is significant because it can help defenders detect and recognize different families of reconnaissance behaviors, and therefore guide defenders in preparing effective defense (e.g., the defenders may only need to spend more resources on coping with a smaller number of representative attackers). Despite it importance, this problem is little understood or investigated.

In this chapter, we make a first step towards solving the aforementioned problem. Specifically, we make the following contributions. First, we initiate the study of clustering cyber attackers using their reconnaissance behaviors, which are modeled as time series of reconnaissance activities. Second, we propose using a two-resolution methodology to characterize cyber attack reconnaissance behaviors. Third, we apply the methodology to a real-world dataset to draw useful insights. We find that a defender needs to consider multiple parameter combinations for the purposes of forecasting cyber attack reconnaissance rates and for quantifying the number of attacker families. The actual number of parameter combinations may be specific to the dataset in question, but identifying a general guiding principle (e.g., what kinds of data would demand what number of parameter combinations) is left for future research.

The rest of the paper is organized as follows. Section 3.2 presents the framework. Section 3.3 reports a case study on applying the framework to a data set collected by a low-interaction honeypot. Section 3.4 discusses related prior studies. Section 3.5 concludes the present paper.

3.2 Framework

At a high level, we propose characterizing cyber attack reconnaissance behaviors from the following three perspectives.

- Attacker similarity within a time window with respect to the coarse-grained time resolution.
- Attacker similarity across $q > 1$ time windows with respect to the coarse-grained time resolution.
- Attacker cluster similarity across $q > 1$ time windows with respect to the coarse-grained time resolution.

As highlighted in Figure 3.1, the framework has four components: (i) data collection and reprocessing, (ii) attackers filtering, time series construction and local window attacker clustering, (iii) construction of the attacker behaviors time series, (iv) evolution temporal analysis, and (v) final attacker clustering.

3.2.1 Data collection and preprocessing

Cyber attack reconnaissance behaviors can be reflected by the corresponding network traffic, which can be captured as raw Packet Capture (PCAP) data from a relevant network instrument, such as honeypot and telescope. PCAP data can be reassembled into IP packets or flows. A flow is a five field tuple: *source IP address*, *destination IP address*, *source port*, *destination port*, and *protocol*

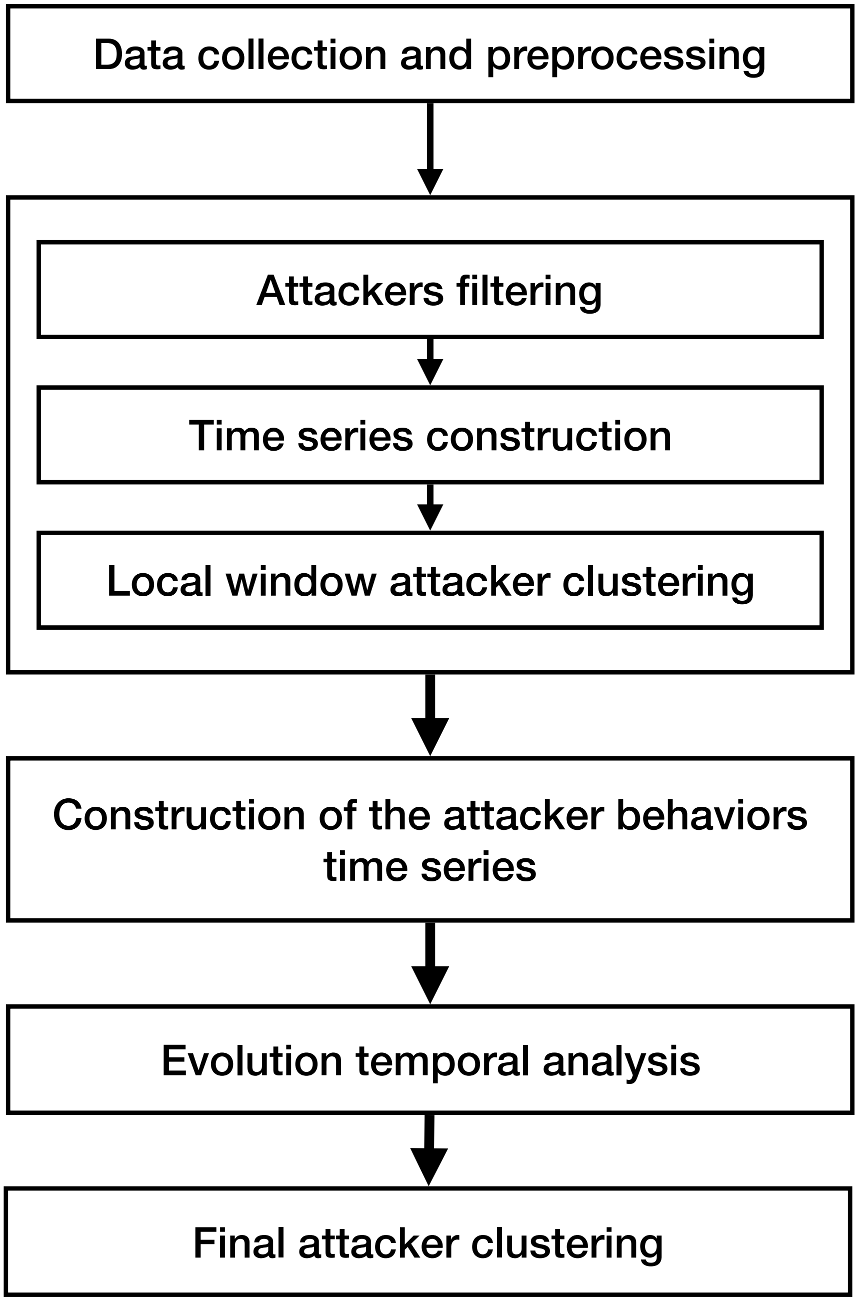

Data collection and preprocessing
Attackers filtering
Time series construction
Local window attacker clustering
Construction of the attacker behaviors time series
Evolution temporal analysis
Final attacker clustering

Figure 3.1: The framework for characterizing cyber attack reconnaissance behaviors.

[36]. There are standard routines for reassembling PCAP data into IP packets or flows [54]. For reassembling flows, two parameters are needed: *idle time*, which specifies when a flow should be terminated after observing no communication activities between a source and a destination; and *lifetime*, which specifies when a flow should be terminated and a new flow may be started. In this paper, we propose using on flow-based representation to characterize cyber attack reconnaissance behaviors.

3.2.2 Building representations of reconnaissance behaviors

The notion of two-resolution method

Denote by $[0, T)$ the life span of a cyber attack reconnaissance behavior dataset. Let Ω be the universe of attacker identities (i.e., IP addresses). We propose modeling cyber attack reconnaissance behaviors via the following *two-resolution* method.

- The first resolution is a coarse-grained time resolution δ (e.g., day or hour), which divides $[0, T)$ into δ-length time windows, denoted by $W_k = [T_k, T_{k+1})$, where $T_{k+1} - T_k = \delta$ for $k = 1, \ldots, K$, $T_1 = 0$ and $K = T/\delta$ (assuming $\delta | T$; otherwise, the last time window may be omitted). Denote by $\Omega_k \subseteq \Omega$ the set of attackers that are observed (i.e., the attacker waged at least one reconnaissance activity) during the k-th time window W_k. Each attacker $u \in \Omega$ in time window W_k incurs $X_{u,k}$ reconnaissance activities, leading to a time series of $X_{u,k}$ over $k = 1, \ldots, K$.

- The second resolution is a fine-grained time resolution ρ (e.g., second or minute), which divides a coarse-grained time window T_k into $n_k = \delta/\rho$ smaller time windows (assuming $\rho | \delta$). Correspondingly, denote by $X_{u,k,t}$ the number of reconnaissance activities by attacker $u \in \Omega$ in the t-th smaller time window of the k-th coarse-grained time window T_k, where $t = 1, \ldots, n_k$. Note that $X_{u,k} = \sum_{t=1}^{n_k} X_{u,k,t}$.

Figure 3.2 illustrates the ideas discussed above, where each dot represents a reconnaissance activity of an attacker u, v or w. Figure 3.2a illustrates the pre-processed data when no time resolution is considered. Figure 3.2b illustrates the coarse-grained representation of the pre-processed data, where each time window has a time length δ. For example, we have $X_{u,1} = 9$ because attacker u waged 9 reconnaissance activities during the first coarse-grained time window $W_1 = [0, \delta)$; we have $\Omega_1 = \{u, v, w\}$ because these three attackers are observed in the coarse-grained time window $W_1 = [0, \delta)$; and we have $\Omega_2 = \{v, w\}$ because only attackers v and w are observed during the coarse-grained time window $W_2 = [\delta, 2\delta)$. Figure 3.2c illustrates how the coarse-grained time windows are further divided into smaller time windows, each of which has a time length ρ. For example, we have $X_{u,1,1} = 1$ and $X_{u,1,2} = 2$ because attacker u waged 1 and 2 reconnaissance activities during the first fine-grained time window $[0, \rho)$ and the second fine-grained time window $[\rho, 2\rho)$, respectively.

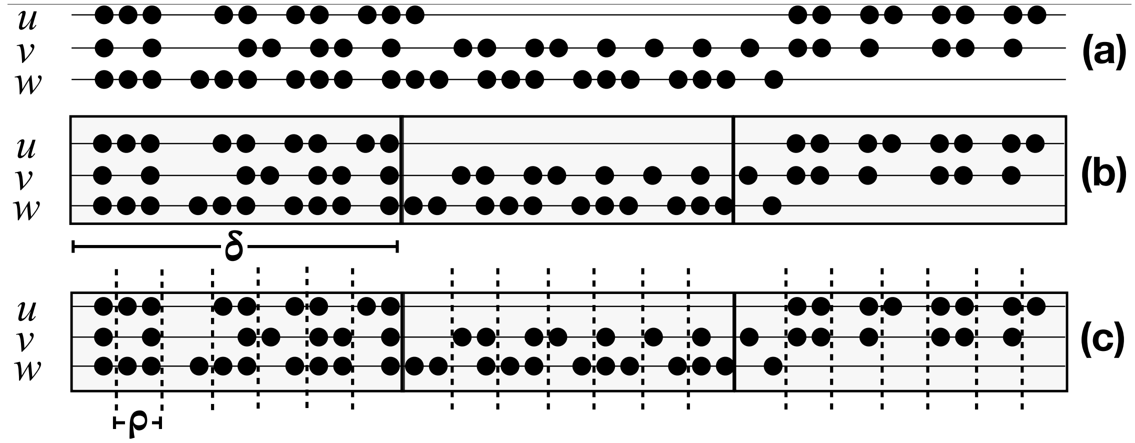


Figure 3.2: Illustration of the two-resolution model for describing cyber attack reconnaissance activities of three attackers, denoted by $u, v, w \in \Omega$.

High-activity vs. low-activity attackers

The basic reconnaissance behavior of attacker $u \in \Omega$ with respect to a coarse-grained time window W_k can be described by the following time series of the number of reconnaissance activities incurred by attacker u during time window W_k:

Time Series 1 (single-window reconnaissance rate)**.** *The time series of the number of reconnaissance activities incurred by attacker $u \in \Omega$ with respect to a coarse-grained time window W_k is defined as $\mathbf{X}_{u,k} = \{X_{u,k,1}, \ldots, X_{u,k,n_k}\}$.*

Since some attackers may incur very few reconnaissance activities, we propose focusing on the *high-activity* attackers, which are identified according to a threshold parameter α with respect to a fine-grained time window of length ρ (and therefore a coarse-grained time window W_k as shown below).

Definition 7 (high-activity vs. low-activity attackers with respect to a coarse-grained time window W_k)**.** *An attacker $u \in \Omega$ is an high-activity attacker in a coarse-grained time window W_k if it wages $X_{u,k} = \sum_{t=1}^{n_k} X_{u,k,t} > \alpha \times n_k$ reconnaissance activities. We propose focusing on the set of high-activity attackers in time window W_k, denoted by Ω_k^H, where $\Omega_k^H = \{u \in \Omega_k \,|X_{u,k} > \alpha \cdot n_k\}$. Denote the set of low-activity attackers by $\Omega_k^L = \Omega_k \setminus \Omega_k^H$.*

Definition 7 leads to the following time series of interest:

Time Series 2 (high-activity attackers over the time horizon)**.** *The time series of the set of high-activity attackers over the time horizon is defined as $\mathbf{A} = \{|\Omega_1^H|, \ldots, |\Omega_K^H|\}$.*

Time series $\mathbf{A}$ is interesting because (i) it serves as a starting point for characterizing the persistent reconnaissance behavior of high-activity attackers and (ii) it can be leveraged to understand the impact of parameter combinations (δ, ρ, α) on the resulting characteristics. Note that the notion of high-activity attackers is defined with respect to a single time series, and that an attacker being highly active in one coarse-grained time window is not necessarily highly active in another time window. This motivates us to identify and characterize the attackers that exhibit persistent high-activity over a number of consecutive coarse-grained time windows, which leads to the following definition:

Persistent high-activity attacker over multiple time windows

Definition 8 (persistent high-activity attackers in q consecutive coarse-grained time windows)**.** *An attacker $u \in \Omega_k$ is said to be a persistent high-activity attacker if u is a high-activity attacker during the q-consecutive time windows $W_{k-q+1}, \ldots, W_k$, where $q \leq k \leq K$. Note that $u \in \bigcap_{j=k-q+1}^{k} \Omega_j^H$.*

Definition 8 leads to the following time series of interest:

Time Series 3 (fraction of persistent high-activity attackers over the time horizon)**.** *The persistent high-activity attackers can be described by their fraction among the high-activity attackers over time, denoted by* $\mathbf{P}_q = \{P_{q,q}, \dots, P_{q,K}\}$*, where*

$$P_{q,k} = \frac{|\bigcap_{j=k-q+1}^{k} \Omega_j^H|}{|\Omega_k^H|} \tag{3.1}$$

for $q \leq k \leq K$*.*

Time series $\mathbf{P}_q$ is interesting because it allows us to (i) characterize the evolution of persistent high-activity attackers and (ii) investigate the impact of parameter q on the resulting characteristics of such attackers. While time series $\mathbf{P}_q$ allows us to characterize persistent attackers from a *temporal* perspective, it focuses on the reconnaissance behaviors of *individual* attackers. This motivates us to further consider their *spatial* behaviors in terms of their similarity, which can be measured in many different ways (as shown by some examples below) but always leads to the notion of *attacker similarity graph* as defined below.

Pair-wise vs. group-wise attackers similarity over a single coarse-grained time window

Definition 9 (attacker similarity graph)**.** *Consider a coarse-grained time window* W_k*, corresponding to which we can define an undirected but weighted graph* $G_k = (\Omega_k^H, E_k, N_k)$*, where the node or vertex set* Ω_k^H *is the set of high-activity attackers in coarse-grained time window* W_k *as defined above,* $E_k = \Omega_k^H \times \Omega_k^H$ *is the edge set, and* $N_k : E_k \rightarrow [0, 1]$ *is a weight of edge* (u, v) *such that* $N_k(u, v)$ *denotes the similarity between attackers* u *and* v *according to an appropriate definition (such as the one described bellow).*

There are many ways to define $N_k(u, v)$. In what follows we give a concrete example, in which we leverage existing time series clustering methods such that the more clustering methods assign two attackers to a same cluster, the higher the similarity between the two attackers. The reason for leveraging multiple clustering methods to define attacker similarity is that we want to make the attacker similarity metric robust.

Definition 10 (example definition of pair-wise attackers similarity based on a set of clustering methods)**.** *Let $\mathbb{M}$ denote a set of clustering methods. For a coarse-grained time window W_k where $1 \leq k \leq K$, suppose a clustering method $m \in \mathbb{M}$ leads to $r_{k,m}$ clusters according to the time series of high-activity attackers observed during time window W_k, namely $\mathbf{X}_{u,k} = \{X_{u,k,1}, \ldots, X_{u,k,n_k}\}$ for $u \in \Omega_k^H$. Let $c_{k,m}(u) \in \{1, \ldots, r_{k,m}\}$ denote the cluster identity to which attacker u belongs according to clustering method m. This leads to the following definition of attacker similarity:*

$$N_k(u,v) = \frac{1}{|\mathbb{M}|} \sum_{m \in \mathbb{M}} \mathbf{1}\left\{c_{k,m}(u), c_{k,m}(v)\right\} \tag{3.2}$$

where

$$\mathbf{1}\{x,y\} = \begin{cases} 1 & \textit{if } x = y, \\ 0 & \textit{otherwise}. \end{cases} \tag{3.3}$$

There are many clustering methods $m \in \mathbb{M}$ that can be used for our purposes, such as those reviewed in [17].

For example, there are a family of closeness-based time series clustering methods, including: Dynamic Time Warping Distance [83, 112, 120], Global Alignment Kernels [41], and Soft-DTW Distance [42, 116].

Given a pair-wise attacker similarity graph G_k, we propose using an appropriate graph clustering method to derive the group-wise membership of attacker u. An example method for this purpose is a community detection algorithm, such as those discussed in [27, 37, 103, 108, 109], while noting that different methods may exhibit different characteristics (e.g., the multi-level community detection algorithm [27] offers a fast execution time in the worst-case scenario, which is important when G_k is large).

Suppose that there are R_k clusters that are derived from an appropriate graph clustering method (e.g., a community detection method mentioned above). Let

$$C_k(u) \in \begin{cases} \{1, \ldots, R_k\} & \text{if } u \in \Omega_k^H \\ \{0\} & \text{otherwise} \end{cases} \tag{3.4}$$

denotes the cluster identity of the attacker u corresponding to the coarse-grained time window W_k. We denote the set of clusters of high-activity attackers with respect to coarse-grained time window W_k as: $\mathbb{C}_k = \{C_k(u) | u \in \Omega^H\}$. In order to understand the evolution of cyber attack reconnaissance behaviors over time, we propose using the similarity between the $\mathbb{C}_k$'s over consecutive time windows.

Definition 11 (Attackers' group-wise similarity over q-consecutive window of time)**.** *Given a number q of consecutive time windows, we define attackers' cluster similarity over q-consecutive time windows, denoted by $S_{q,k}$ where $q \leq k \leq K$, as*

$$S_{q,k} = \frac{1}{q} \sum_{j=k-q}^{k-1} \zeta(\mathbb{C}_j, \mathbb{C}_k) \tag{3.5}$$

where $\mathbb{C}_k$ represent the attackers' cluster labels in time window k and $\zeta(\mathbb{C}_j, \mathbb{C}_k)$ is an appropriate definition of similarity between the cluster assignments in time window j and the cluster assignments in time window k (see example below).

Example 12. *In order to help understand Definition 11, let us consider the following toy example with two time windows, W_1 and W_2, where $\Omega_1^H = \{u_1, u_2, u_3, u_4\}$, $\Omega_2^H = \{u_3, u_4, u_5, u_6\}$, and $\Omega^H = \{u_1, u_2, u_3, u_4, u_5, u_6\}$. Suppose the attackers in Ω_1^H are clustered into two groups, meaning $R_1 = 2$, and the attackers in Ω_2^H are clustered into three groups, meaning $R_2 = 3$. For W_1, we have $C_1(u) \in \{1, 2\}$ for any $u \in \Omega_1^H$ can have following assignments and $C_1(u) \in \{0\}$ for any $u \in \Omega^H \setminus \Omega_1^H$; suppose the clustering assignments for all of the attackers in Ω^H with respect to time window W_1 is $\mathbb{C}_1 = \{1, 1, 2, 2, 0, 0\}$, meaning that u_1 and u_2 belong to the same cluster, u_3 and u_4 belong to the same cluster, and u_5 and u_6 belong to the cluster of no activities. For W_2,* *we have $C_2(u) \in \{1,2,3\}$ for any $u \in \Omega_2^H$ and $C_1(u) \in \{0\}$ for any $u \in \Omega^H \setminus \Omega_2^H$; suppose the clustering assignment for all of the attackers $u \in \Omega^H$ with respect to time window W_2 is $\mathbb{C}_2 = \{0,0,3,2,2,1\}$.*

There are many ways for defining $\zeta(\mathbb{C}_j, \mathbb{C}_k)$, such as: the Normalize Mutual Information score (NMI) [117], the Adjusted Rand Score score [122], the Fowlkes-Mallows Index (FMI) [49], the Homogeneity score [107], the Completeness score [107], and the V-Measure (vM) score [107]. For example, the vM score is the harmonic mean of the homogeneity and the completeness and is equivalent to the NMI when the harmonic mean is replaced with the arithmetic mean. More specifically, let us substitute vM(C_j, C_k) for $\zeta(\mathbb{C}_j, \mathbb{C}_k)$ when we consider the vM score between two cluster assignments C_j and C_k. Then, we have [107]:

$$\text{vM}(\mathbb{C}_j, \mathbb{C}_k) = 2\left(\frac{\hat{\mathbf{h}}(\mathbb{C}_j, \mathbb{C}_k) \times \hat{\mathbf{c}}(\mathbb{C}_j, \mathbb{C}_k)}{\hat{\mathbf{h}}(\mathbb{C}_j, \mathbb{C}_k) + \hat{\mathbf{c}}(\mathbb{C}_j, \mathbb{C}_k)}\right), \tag{3.6}$$

where the *homogeneity* score $\hat{\mathbf{h}}(\mathbb{C}_j, \mathbb{C}_k)$ measures intuitively to what extent the clusters in $\mathbb{C}_k$ belong to one or another cluster in $\mathbb{C}_j$ and the *completeness* score $\hat{\mathbf{c}}(\mathbb{C}_j, \mathbb{C}_k)$ measures the members of a class in $\mathbb{C}_k$ are assigned to the same cluster in the cluster assignment $\mathbb{C}_j$. Note that both the homogeneity measure and the completeness measure are asymmetric, meaning that $\hat{\mathbf{h}}(\mathbb{C}_j, \mathbb{C}_k) \neq \hat{\mathbf{h}}(\mathbb{C}_k, \mathbb{C}_j)$ and $\hat{\mathbf{c}}(\mathbb{C}_j, \mathbb{C}_k) \neq \hat{\mathbf{c}}(\mathbb{C}_k, \mathbb{C}_j)$. However, the V-Measure score (vM) is a symmetric measure where vM$(\mathbb{C}_j, \mathbb{C}_k) =$ vM$(\mathbb{C}_k, \mathbb{C}_j)$.

Example 13 (vM measure of homogeneity and completeness between two cluster assignments)**.** *Consider two cluster assignments $\mathbb{C}_1 = \{0,0,1,1,1,1\}$ and $\mathbb{C}_2 = \{0,0,1,1,2,2\}$. On one hand, we have $\hat{\mathbf{h}}(\mathbb{C}_1, \mathbb{C}_2) = 1$ because when comparing $\mathbb{C}_2$ against $\mathbb{C}_1$, we see each cluster in $\mathbb{C}_2$ also belongs to a single cluster in $\mathbb{C}_1$; on the other hand, we have $\hat{\mathbf{h}}(\mathbb{C}_2, \mathbb{C}_1) = 0.58$ because one cluster in $\mathbb{C}_1$ is split into two clusters in $\mathbb{C}_2$. In terms of the completeness measure, we have $\hat{\mathbf{c}}(\mathbb{C}_2, \mathbb{C}_1) = 1$ because the members of the clusters in $\mathbb{C}_1$ are assigned to a same cluster in $\mathbb{C}_2$ meaning that the cluster $\{1,1\} \in \mathbb{C}_2$ are also the cluster together in $\mathbb{C}_1$ and the cluster $\{2,2\} \in \mathbb{C}_2$ are also the cluster together in $\mathbb{C}_1$. However $\hat{\mathbf{c}}(\mathbb{C}_1, \mathbb{C}_2) = 0.58$ because the members of the clusters $\{1,1\}$ and*

$\{2,2\}$ *in* $\mathbb{C}_2$ *have the same cluster assignment in* $\mathbb{C}_1$*, meaning the cluster where different in* $\mathbb{C}_2$ *but the same in same class in* $\mathbb{C}_1$*. Putting the pieces together, we have* vM *score*

$$\begin{aligned}\text{vM}(\mathbb{C}_1,\mathbb{C}_2) &= 2\left(\frac{\hat{\mathbf{h}}(\mathbb{C}_1,\mathbb{C}_1)\times\hat{\mathbf{c}}(\mathbb{C}_1,\mathbb{C}_2)}{\hat{\mathbf{h}}(\mathbb{C}_1,\mathbb{C}_2)+\hat{\mathbf{c}}(\mathbb{C}_1,\mathbb{C}_2)}\right)\\ &= 2\left(\frac{1\times 0.58}{1+0.58}\right)=0.73\end{aligned}$$

and

$$\text{vM}(\mathbb{C}_2,\mathbb{C}_1) = 2\left(\frac{0.58\times 1}{0.58+1}\right)=0.73$$

Building on top of Definition 11, we can define a time series of group-wise attacker similarity over the time horizon (Time Series 4) and the notion of *group-wise attacker similarity over the time horizon.*

Time Series 4 (group-wise attacker similarity over the time horizon)**.** *The attackers' group similarity can be described by the time series* $\mathbf{C}_q = \{S_{q,q}, S_{q,q+1}, \dots, S_{q,K}\}$ *for* $1 < q < K$.

Time series $\mathbf{C}_q$ is interesting because it allow us (i) to characterize the evolution of the attackers clusters similarity and (ii) to investigate how the choice of parameter q influence the resulting characteristics.

Definition 14 (pair-wise attacker similarity over the time horizon)**.** *Given the single-window pair-wise attacker similarity graphs* G_k *for all* $k \in [1,K]$*, we propose creating a new graph to reflect the pair-wise attacker similarity over the time horizon, namely*

$$\mathcal{G} = (\Omega^H, \mathcal{E}, \mathcal{N}),$$

where the vertex set $\Omega^H = \bigcup_{k=1}^K \Omega_k^H$ *and the edge set* $\mathcal{E} = \bigcup_{k=1}^K E_k$ *with an edge* (u,v) *annotated with a weight* $\mathcal{N}(u,v) = \frac{1}{K}\sum_{k=1}^K N_k(u,v)$ *(i.e., the average pair-wise attacker similarity, where the average is taken over the time windows).*

Pair-wise attacker similarity over the time horizon

Definition 14 allows us to cluster attackers according to their *average* reconnaissance behaviors over the time horizon.

Definition 15 (attacker clusters according to their average reconnaissance behaviors over the time horizon)**.** *Given $\mathcal{G} = (\Omega^H, \mathcal{E}, \mathcal{N})$, the high-activity attackers can be clustered into R families (e.g., via an appropriate community detection algorithm). Let $Q(u) \in \{1, \ldots, R\}$ denote the cluster assignment of attacker $u \in \Omega^H$. Then, the i-th family of attackers can be defined as*

$$\mathbb{Q}_i = \{u | Q(u) = i \wedge u \in \Omega^H\}$$

for $1 \leq i \leq R$, meaning that the attackers in a same family exhibits a similar reconnaissance behavior on average over the time horizon.

On the relationship between the definitions

Figure 3.3 highlights the relationships between the concepts defined above. A solid arrow pointing from A to B means that concept B refines concept A from a specific perspective. For example, the concept *persistent high-activity attackers*, which is defined over q coarse-grained time windows, refines the concept of *high-activity attackers*, which is defined over a single coarse-grained time window, from the perspective of the period of time during which an attacker is highly active. A dashed arrow pointing from C to D means that concept C uses concept D as a building-block. For example, the concept of *pair-wise attacker similarity over the time horizon* is used as a building-block in defining the concept of *attacker clusters over the time horizon* because the latter group together multiple attackers that exhibit a large pair-wise similarity.

3.2.3 Research Questions

Building on top of the preceding descriptive model, we investigate the following research questions (RQs):

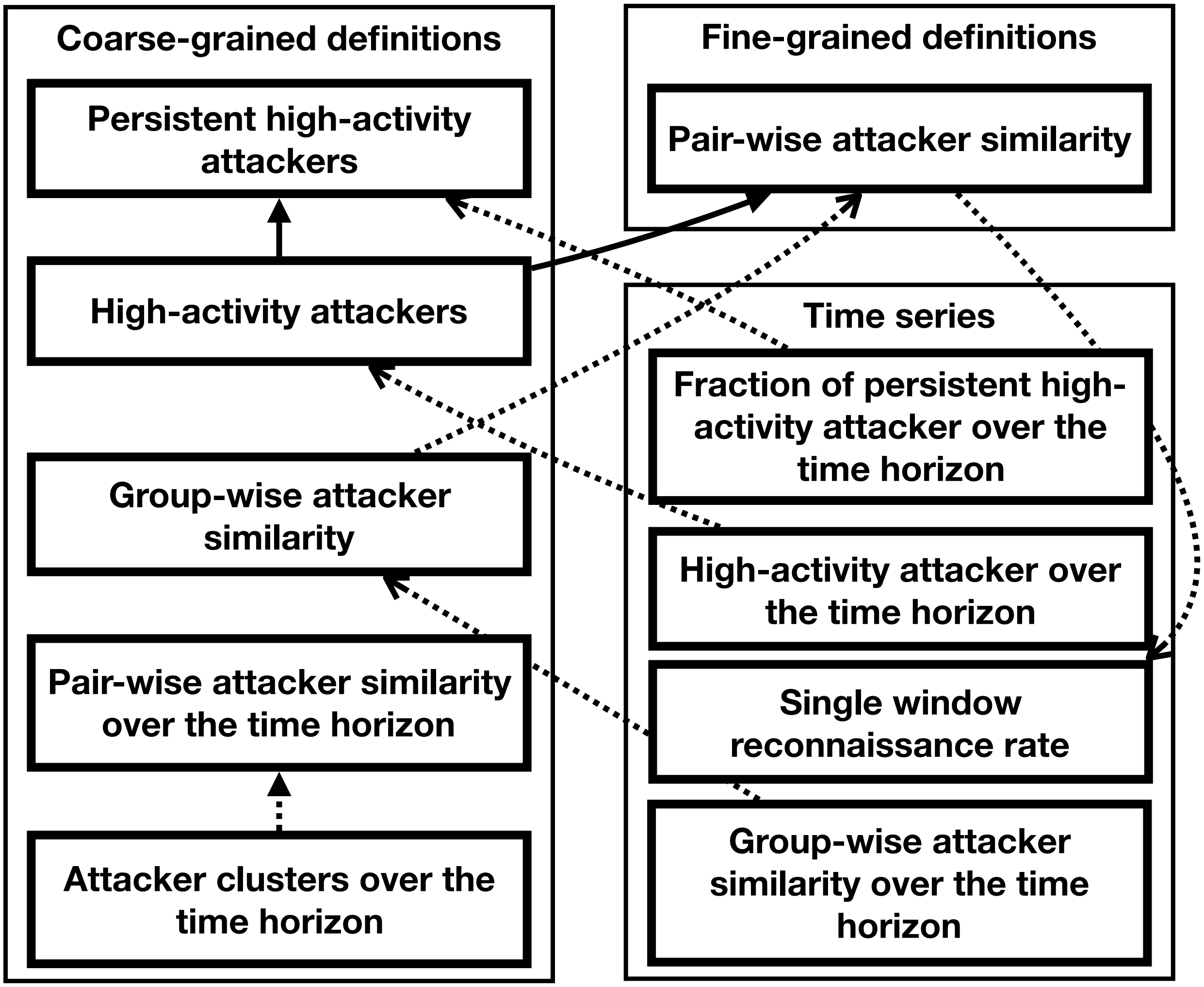

Coarse-grained definitions
Persistent high-activity attackers
High-activity attackers
Group-wise attacker similarity
Pair-wise attacker similarity over the time horizon
Attacker clusters over the time horizon
Fine-grained definitions
Pair-wise attacker similarity
Time series
Fraction of persistent high-activity attacker over the time horizon
High-activity attacker over the time horizon
Single window reconnaissance rate
Group-wise attacker similarity over the time horizon

Figure 3.3: The relationship between the concepts in coarse-grained time resolution vs. fine-grained time resolution.

RQ1: How does the choice of parameter q influences the properties of the applicable times series, namely: $\mathbf{P}_q$ and $\mathbf{C}_q$? This matter is important because the defender only needs to consider the q's that make $\mathbf{P}_q$ and $\mathbf{C}_q$ exhibit different characteristics.

RQ2: For a fixed q, how does the parameter combination (δ, ρ, α) influence the properties of the applicable time series $\mathbf{P}_q$ and $\mathbf{C}_q$ from the Long Range Dependence (LRD) perspective? This matter is important because it is known that LRD can lead to a high prediction accuracy [33, 93, 96, 143, 145].

RQ3: When do the time series $\mathbf{A}$, $\mathbf{P}_q$ and $\mathbf{C}_q$ exhibit LRD and a similar trend? By grouping the parameter settings according which the time series $\mathbf{A}$, $\mathbf{P}_q$ and $\mathbf{C}_q$ a similar trend, the defender can reduce the number of parameter settings that should be considered in order to have a comprehensive understanding of cyber attack reconnaissance behaviors.

RQ4: How many families of attacker reconnaissance behaviors during the entire life-span? This matter is clearly important because it substantially reduces the number of attackers the defender would have to cope with.

3.3 Case Study and Results

In this section we present a case study and report the results.

3.3.1 Data collection and pre-processing

Our case study is based on a dataset collected at a low-interaction honeypot consisting of 1,024 Internet Protocol (IP) addresses, during the period of time between 2/6/2014 and 5/8/2014 (i.e., 96 days in total). Although the dataset is several years old, it is sufficient for our purpose of demonstrating the usefulness of our framework. The honeypot ran the low-interaction honeypot programs known as *Honeyd* [100] and *Nepenthes* [24]. Since a honeypot offers no legitimate services, the incoming traffic can bee deemed as malicious, which is a widely-adopted practice (see, e.g., [22, 44, 51, 68, 79, 96–99, 143, 145]). We convert the honeypot-collected raw PCAP

Table 3.1: Description of papers notations and symbols

Notation	Description
$[0, T)$	The dataset life-spam.
K	Number of non overlapping windows.
$W_k = [T_k, T_{k+1})$	The k-th window for $k = 1, \dots, K$. Where T_k is the start time of the window and T_{k+1} is the end time of the window.
$\delta = T_{k+1} - T_k$	The window length and coarse-grained resolution.
Ω	All the attacker in the data set.
Ω_k	Domain of all attacker that launch at least one attack at window W_k.
ρ	The fine-grained resolution
$X_{u,k,t}$	The attacker $u\ in \Omega_k$ number reconnaissance activities at time t for the fine-grained resolution where $t = 1, \dots, n_k$ and $n_k = \frac{\delta}{\rho}$.
$X_{u,k} = \sum_{t=1}^{n_k} X_{u,k,t}$	The total reconnaissance activities at the coarse-grained resolution.
α	Threshold to split the attacker domain Ω_k into high-activity and low-activity attackers.
Ω_k^H	Domain of high-active attackers at window W_k.
Ω_k^L	Domain of low-active attackers at window W_k.
$\mathbf{A}$	The time series of high-activity attacker over the time horizon
$P_{q,k}$	The fraction of persistent high-activity attacker at window W_k.
$\mathbf{P}_q$	The time series for the fraction of persistent high-activity attacker over the time horizon.
$G_k = (\Omega_k^H, E_k, N_k)$	Attacker similarity graph at the local window W_k.
$N_k(u, v)$	Weight for the edges in the attacker similarity graph G_k.
$\mathbb{M}$	Set of clustering methods.
$c_{k,m}(u)$	The attacker u clustering identity given the clustering method mß$\mathbb{M}$ at the local window W_k.
$C_k(u)$	The attacker u cluster identity at the coarse-grained time window W_k.
$\mathbb{C}_k$	The set of cluster of high-activity attacker with respect to a coarse-grained time window W_k.
$S_{q,k}$	Attacker cluster similarity over q consecutive time windows.
$\zeta(\cdot, \cdot)$	Similarity between the set of clustering identities.
$\mathbf{C}_q$	The time series for the group-wise attacker similarity over the time horizon.
$\mathcal{G} = (\Omega^H, \mathcal{E}, \mathcal{N})$	Pair-wise attacker similarity graph over the time horizon.
$Q(u)$	The attacker $u \in \Omega^H$ cluster identity over the time horizon.
$\mathbb{Q}_i$	The i-th family of attacker over the time horizon.
$\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbb{N})$	The graph of similar parameter settings.
$\mathbb{N}(\cdot, \cdot)$	Similarity of parameter settings.

dataset into the IPFIX network flows by using the tools known as Yet Another Flowmeter (YAF) and super_mediator of the Computer Emergency Response Taskforce (CERT) [62]. As in many previous studies, we set the flow idle time to be 60 seconds and set the flow lifetime to be 300 seconds (see, for example, [96, 143, 145]).

Table 3.2: Basic statistics of reconnaissance activities, where μ is the average and σ the standard deviation.

		Basic Statistics				
		min	μ	median	max	σ
Flows with non-zero duration time	flow duration	0.001	16.8	0.69	300	1185.6
	# of packets	2	3.7	3	550	12.5
	# of bytes	56	248	167	41,220	157,195
Flows with zero duration time	# of packets	1	1	1	65	0.2
	# of bytes	28	51	40	2,600	1,552.3

The dataset contains 42,734,422 TCP flows, each of which is treated as a reconnaissance activity. Among these flows, 74% of them have zero duration time, meaning that they correspond to single packet reconnaissance activities. Among the 26% of the flows with non-zero duration time (indicating multiple-packet reconnaissance activities), 50% of them have a duration time that is less than 0.7 seconds, indicating that most reconnaissance behaviors are essentially scanning/probing. Table 3.2 summarizes the basic statistics of the flows in two groups (i.e., non-zero vs. zero duration time), where "# of packets (bytes)" means the number of packets (bytes) of an individual flow.

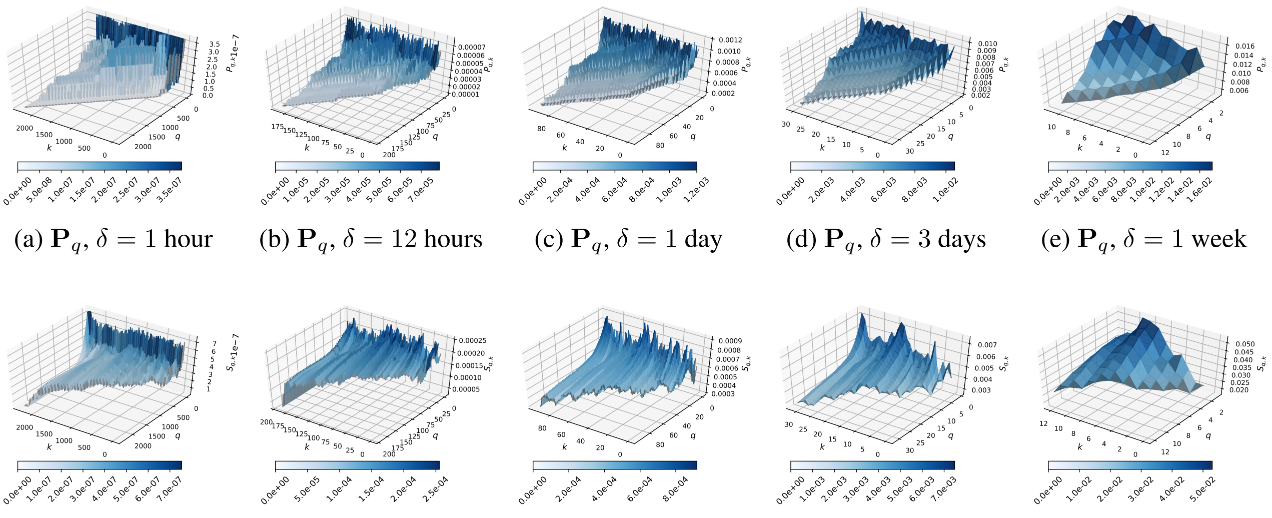
3.3.2 Experiments settings

In our experiments, we consider the following parameter combinations of (δ, ρ, α), while noting that an appropriate q will be searched. First, for the coarse-grained time resolution δ, we consider five resolutions: one week, three days, one day, twelve hours, and one hour. For the fine-grained time resolution ρ, we test ten resolutions: two hours, one hour, thirty minutes, fifteen minutes, ten minutes, five minutes, one minute, thirty seconds, fifteen seconds, and ten seconds. Nevertheless, when the coarse-grained time resolution δ is low (e.g. twelve hours or one hour), it is not meaningful to consider a low fine-grained time resolution (e.g., two hours or one hour) because a fine-grained time series within a coarse-grained time window does not have enough data points that are adequate for a statistical analysis. Therefore, in total we consider 20 parameter combinations of (δ, ρ), which are listed in the first two columns of Table 3.3. Second, for the threshold parameter α (according to which we differentiate high-activity attackers from low-activity attackers), we consider three values: 0.25, 0.5 and 0.75.

In our experiments, we consider a bagging clustering approach. For the clustering methods in the bagging approach we combine three distance measures with nine clustering algorithms resulting in the $|\mathbb{M}| = 27$ pair-wise time series clustering methods. The three distance metrics are: the Dynamic Time Warping Distance [83, 112, 120], the Global Alignment Kernels [41], and the Soft-DTW Distance [42] (i.e., the R library *dtwclust* [116]). Over a distance matrix, we apply the following clustering algorithms: k-Means, PAM, AGNES single linkage, AGNES average linkage and ANGNES complete linkage from the R library *cluster* and the community detection algorithms Fast greedy community (FGC) detection introduced by [37], Multi level community (MLC) detection introduce by [27], Label propagation community (LPC) detection [103], and Information map community (IMC) detection introduced by [108, 109] from the *igraph* library [40]. For the cluster similarity measure in Equation 3.5, we used the V-Measure Score to compute $\zeta(\mathbb{C}_j, \mathbb{C}_k)$. We choose the V-Measure score because: (i) it is equivalence to the arithmetic normalize mutual information score (ii) the metrics is symmetric compared with the homogeneity and completeness scores [107], (iii) is equivalent the harmonic mean of the homogeneity and the completeness scores [107].

3.3.3 Results

Now we report our experimental results with respect to the four RQs mentioned above.



(a) $\mathbf{P}_q$, $\delta = 1$ hour (b) $\mathbf{P}_q$, $\delta = 12$ hours (c) $\mathbf{P}_q$, $\delta = 1$ day (d) $\mathbf{P}_q$, $\delta = 3$ days (e) $\mathbf{P}_q$, $\delta = 1$ week

(f) $\mathbf{C}_q$, $\delta = 1$ hour (g) $\mathbf{C}_q$, $\delta = 12$ hours (h) $\mathbf{C}_q$, $\delta = 1$ day (i) $\mathbf{C}_q$, $\delta = 3$ days (j) $\mathbf{C}_q$, $\delta = 1$ week

Figure 3.4: (a)-(e) The average evolution of time series $\mathbf{P}_q$, (f)-(j) the average evolution of time series $\mathbf{C}_q$ or $1 < q \leq k$, $1 < k \leq K$, between parameters settings with similar δ and different ρ and α.

Experimental Results with respect to RQ1

Figure 3.4 plots the average evolution of time series $\mathbf{P}_q$ and $\mathbf{C}_q$ for $1 \leq q < k$ and $1 < k < K$ with different combinations of parameters (δ, ρ, α), where the average is for a fix values of δ over the parameters ρ and α. For example, for Figure 3.4a and Figure 3.4f present the average $\mathbf{P}_q$ and average $\mathbf{C}_q$ respectively for $\delta = 1$ hour by averaging values of $\rho \in \{1$ min., 30 sec., 15 sec., 10 sec.$\}$ and $\alpha \in \{0.25, 0.5, 0.75\}$.

As we expect that both the persistence and the similarity diminish as q increases. We also observe that the attacking behaviors are mostly similar for $q = 1$. Therefore, we proceed with $q = 1$ and assess the time series $\mathbf{P}_1$ and $\mathbf{C}_1$. We also observe that $\mathbf{P}_1$ and $\mathbf{C}_1$ often exhibit quite different characteristics because they reflect different aspects of high-activity attackers.

Insight 16. *Both persistency and similarity diminish as q increases, and reconnaissance behaviors are mostly similar within $q = 1$ coarse-grained time window. This suggests us to consider $q = 1$ only.*

Experimental Results with respect to RQ2

For this analysis we propose estimating the Hurst-Kolmogorov parameter H to test which parameters combination leads to time series with $H \geq 0.5$, indicating the presence of LRD [121]. For a time series with a Hurst parameter $H \geq 0.5$, we propose applying the Smoothly Varying Trend test [102] to verify if the LRD is spurious or not. Since not all of the parameter combinations lead to LRD, we propose using the following rankings:

- Corresponding to a q, a parameter setting is *suitable* if the time series $\mathbf{A}$, $\mathbf{P}_q$ and $\mathbf{C}_q$ exhibit the LRD property.

- Corresponding to a q, a parameter setting is *robust* if the time series $\mathbf{A}$, $\mathbf{P}_q$ and $\mathbf{C}_q$ exhibit the LRD property for any three values of $\alpha \in \{0.25, 0.5, 0.75\}$.

Note that a robust parameter setting implies the parameter setting is also suitable.

Table 3.3 lists the parameter settings and time series that exhibit LRD, spurious-LRD, or doesn't exhibit LRD. We observe that when δ is too small (e.g., smaller that 12 hours) or too large (e.g., larger than 3 days), the time series do not exhibit LRD or exhibit spurious LRD, which hints that cyber attack reconnaissance behaviors are not rich within a small time window but are diminishing within a large time window. The parameter settings of (i) $\delta = 1$ day and $\rho = 5$ min; (ii) $\delta = 1$ day and $\rho = 30$ min; and (iii) $\delta = 12$ hours and $\rho = 1$ min are *robust* parameters because all of the time series exhibit LRD regardless of the value of α. For each of the time series, the ✓s in columns $\mathbf{A}$, $\mathbf{P}_1$, and $\mathbf{C}_1$ represent the parameter settings that better characterize the evolution of the number of high-activity attackers, the evolution of attacker persistence, and the evolution of the attacker cluster similarity, respectively.

Insight 17. *The defender needs to identify the right combinations of parameter values in order to make the time series exhibit LRD and forecast cyber attack reconnaissance activities.*

Table 3.3: LRD Analysis and Trend Correlation results: The × mark indicate that a time series exhibits spurious LRD, the ✓indicates that a time series exhibits true LRD, and we leave it blank for the parameter settings that do not exhibit LRD ($H \leq 0.5$). The "Cluster Id" column indicates the trend correlation cluster id for the suitable parameter settings.

δ	ρ	**A**			$\mathbf{P}_1$			$\mathbf{C}_1$			Cluster Id		
		0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75
week	2 hours									✓			
week	1 hour	✓	✓	✓	✓			✓	✓	✓	0		
week	30 min		✓						✓	✓			
week	15 min	✓	✓	✓				✓	✓	✓			
3 days	1 hour	✓	✓	✓	✓			✓	✓	✓	0		
3 days	30 min	✓	✓	✓				✓	✓	✓			
3 days	15 min			✓						✓			
3 days	10 min	✓	✓						✓	✓			
1 day	**30 min**	✓	✓	✓	✓	✓	✓	✓	✓	✓	0	0	0
1 day	15 min	✓	✓	✓				✓	✓	✓			
1 day	10 min	✓	✓	✓	✓			✓	✓	✓			
1 day	**5 min**	✓	✓	✓	✓	✓	✓	✓	✓	✓	1	1	0
12 hours	10 min	✓	✓	✓	✓	×	✓	✓	✓	✓		0	0
12 hours	5 min	×	×	×	✓	×	✓	✓	✓	×			
12 hours	**1 min**	✓	✓	✓	✓	✓	✓	✓	✓	✓	0	0	1
12 hours	30 sec	✓	✓	✓	×	✓	✓	✓	✓	✓		1	1
1 hour	1 min	×	×	×	×	×	×	×	×	×			
1 hour	30 sec	×	×	×	×	×	×	✓	×	×			
1 hour	15 sec	×	×	×	×	×	✓	✓	×	✓			
1 hour	10 sec	×	×	×	×	×	×	×	×	✓			

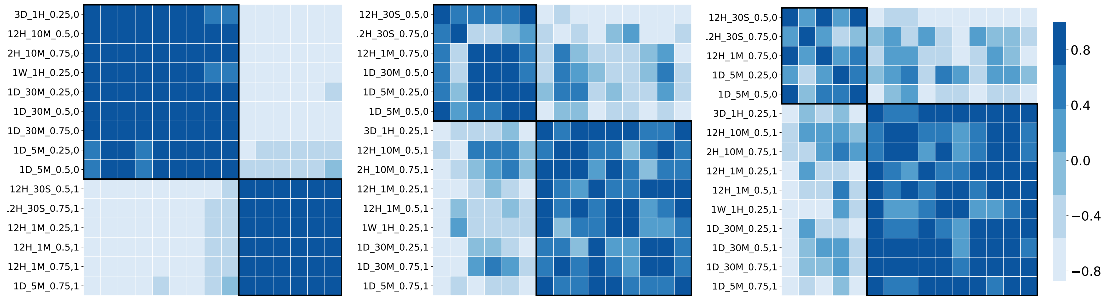
Experimental Results with respect to RQ3

Table 3.3 shows that certain parameter settings are more suitable than others. Within the suitable parameter settings, we assess whether or not some of the parameter settings lead to similar attacker reconnaissance behavior patterns. In this analysis, we extract the trend of the evolution: $\mathbf{A}$, $\mathbf{P}_q$, and $\mathbf{C}_q$ using a frequency of two weeks, then we compare the the correlation between the trends between different parameter settings, and compute a correlation matrix for each time series. Different window lengths δ produce time series with different number of observations or length. In order to assess the correlation between the time series with different parameter settings, we need to converge the different length time series into time series of same length or same number of observations. To achieve the goal, we take the average of the observations in the time series to match the number of observations of the time series with least number of observations. Assume that the time series with the least number of observations have a weekly observation, to converge the daily time series to match the time series with weekly observations, we take the average of the seven daily time series within the same time window. The other times-series with observations every 3-day, 12-hour, 1-hour, etc. are converged similarly.

After converting the time series with different parameters into a same frequency, we are able to obtain a correlation matrix for each time series. Based on the pairwise correlation, which can be viewed as a similarity measure between those time series, we apply the agglomerative cluster using ward linkage [57, 123] to group parameter settings into different clusters. We choose the number of cluster that optimize the silhouette score [57, 110].

In addition, to the suitable and robust parameter settings rankings the results of the parameter clustering suggest a third ranking for the parameter settings that are robust and produce time series with similar trends. Corresponding to the number of consecutive windows q, we say a parameter setting is *ideal* if the time series $\mathbf{A}$, $\mathbf{P}_q$ and $\mathbf{C}_q$ are *robust* and the evolution trends are similar for $\alpha \in \{0.25, 0.5, 0.75\}$.

Figure 3.5a-3.5c present the correlation matrix clustering results for the time series $\mathbf{A}$, $\mathbf{P}_1$, and $\mathbf{C}_1$ respectively. We set $q = 1$ based on our previous conclusion that a shorter parameter q leads



(a) (b) (c)

Figure 3.5: (a) correlation matrix for the $\mathbf{A}$ trends, (b) correlation matrix for the $\mathbf{P}_1$ trends, (c) correlation matrix for the $\mathbf{C}_1$ trends. The row label shows the parameter settings where W is short for week, D is short for day, H is short for hour, M is short for minute and S is short for second.

to a more persistent attacking behaviors. In general, the suitable parameter settings are grouped into two clusters based on the correlation between the time series constructed based on them. However, these two clusters may be different for different time series. Hence, to further understand the similarity of different parameter settings, we combine the clustering results of the correlation matrices for $\mathbf{A}$, $\mathbf{P}_1$ and $\mathbf{C}_1$ to create a complete weighted and undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbb{N})$. Where the vertex set $\mathbb{V}$ correspond to the suitable parameter settings, and the edge set $\mathbb{E} = \mathbb{V} \times \mathbb{V}$. To assess the weigh of the edges we define the edge weigh function $\mathbb{N} : \mathbb{E} \to [0, 1]$ where for parameter settings $(p, p') \in \mathbb{V}$:

$$\mathbb{N}(p, p') = \frac{1}{3} \sum_{R \in \{\mathbf{A}, \mathbf{P}_1, \mathbf{C}_1\}} \mathbf{1}_R(p, p') \tag{3.7}$$

and

$$\mathbf{1}_R(p, p') = \begin{cases} 1 & \text{if parameter settings } p \text{ and } p' \text{ are clustered} \\ & \text{together when using correlation matrix } R \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

and find the communities in the graph $\mathbb{G}$. Figure 3.6 show the two group of parameter settings given by the communities in the similarity graph and the “Cluster Id” column in Table 3.3 show the two groups of parameter settings.

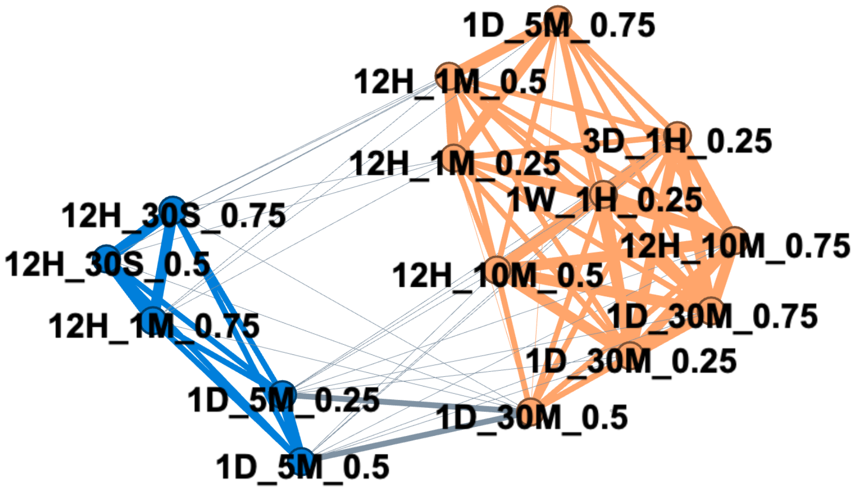


Figure 3.6: suitable parameter settings affinity graph. Communities are differentiate by the color. The width of the edge is proportional to the number of time series for which the pair of parameters have correlated trend. Notation: W is short for week, D is short for day, H is short for hour, M is short for minute and S is short for second.

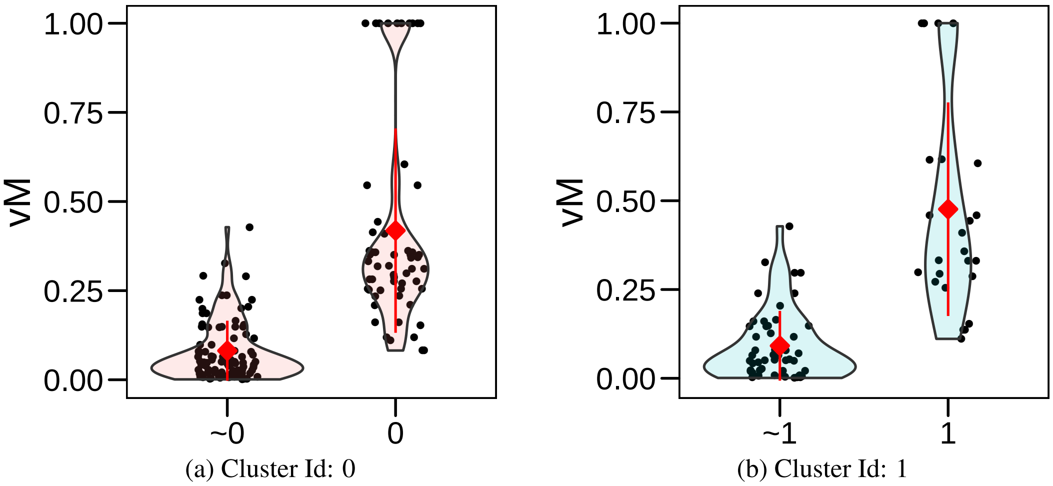
The parameters settings δ =1 day and ρ =30 min is an *ideal* parameters settings because is *robust* and the trend of the time series $\mathbf{A}$, $\mathbf{P}_1$ and $\mathbf{C}_1$ fall under the same group.

Insight 18. *Depending the parameter setting a data set might exhibit different evolution's, leading into multiple cases. Defenders need to be conscious on how many parameter settings are required to have a holistic description of the evolution's.*

Experimental Results with respect to RQ4

Given the similarity graphs corresponding to each coarse-grained time window, we compute graph $\mathcal{G}$ as described in the framework. In order to find communities in $\mathcal{G}$, we adopt the MLC algorithm because the number of edges in $\mathcal{G}$ is $\mathcal{O}(|\mathcal{V}|^2)$ and the MLC algorithm has a better complexity than the other algorithms, namely $\mathcal{O}(n\log^2(n))$, where $n = |\Omega_k^H|$ [86].

As discussed above, the parameter settings can be grouped into two clusters, namely Cluster Id 0 and Cluster Id 1. On average both clusters of similar parameter settings cluster the attackers in 4 families of reconnaissance behaviors. We identify the final clustering of attackers in each parameter setting. Figure 3.7 shows that clustering results with respect to the parameter settings in the same group are similar, with an average vM score of 0.5. However, when comparing the clustering results with the parameter settings that belong to different groups, the average vM is below 0.15. This indicates that similar parameter settings indeed lead to similar clustering results.



(a) Cluster Id: 0

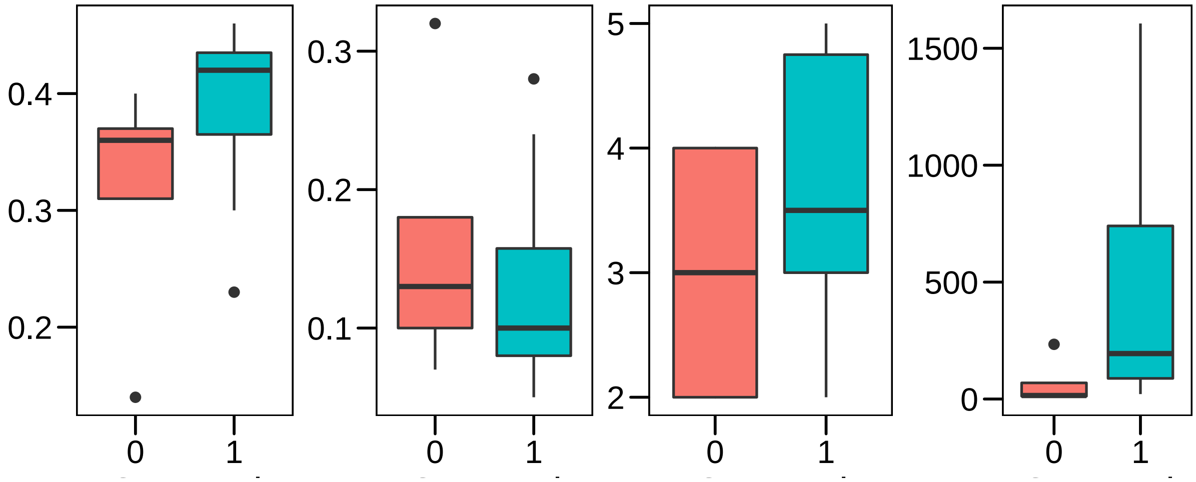
(b) Cluster Id: 1

Figure 3.7: V-Measure score comparison between the final clusters in the experiments witting the same cluster id h, and the V-Measure score of the clustering form experiments in group h with the clustering of the experiments not in group h, label as $\sim h$. We can observer the comparison between the experiment witting the same group have an higher vM score while the comparison with experiment outside the group have a lower vM score.

Figure 3.8 presents the box-plots of characteristics of the attacker families in each cluster of parameter settings. To understand the topology of the families in each parameter settings cluster we measure the sparsity and density using the Gini-Index (GI) generalization for graphs sparsity [55] and the Coleman graph density score (D) estimation [38]. Both GI and D are between zero and one, where the graph sparsity is higher when the GI value is closer to 1 and the D value is closer to 0. In Figure 3.8a and Figure 3.8b, the values of GI and D show that cluster one generates more sparse communities than cluster zero does. The aforementioned result present two complementary results depending the context the families of attacker that are more sparse can be more suitable or vise-versa. Therefore both clusters are necessarily to provide a holistic description of the attacker reconnaissance families. Because each cluster offers a different characterization of the attacker reconnaissance families, both clusters of suitable parameter settings are needed to offer a more comprehensive analysis. In Figure 3.8c, we observe that both clusters has from two to five families, while noting that the number of communities is much higher than the number of families.

Figure 3.8d indicates that there many attackers that are isolated, meaning that that those attacker have their own reconnaissance behaviors.

Insight 19. *The defender may have to consider multiple parameter combinations for quantifying the number of cyber attack reconnaissance behaviors, perhaps because the reconnaissance behaviors do not contain enough information to uniquely cluster cyber attack reconnaissance behaviors.*



(a) (b) (c) (d)

Figure 3.8: (a) Gini-Index sparsity score, (b) Density score, (c) the number of families and (d) number of communities for the ensemble graph $\mathcal{G}$.

3.4 Related Work

From a conceptual point of view, the present study falls into the field of cybersecurity data analytics, which has been investigated in many contexts, such as: univariate time series forecasting [33, 93, 132, 143–145], multivariate time series forecasting [96, 130], and graph time series of attacker-victim relations [54]. The present study investigates a new aspect of multivariate time series, namely the *temporal* and *spatial* behaviors of cyber attack reconnaissance behaviors.

From a datasets point of view, low-interaction honeypot data has been analyzed from the following aspects: the visualization of the five flow field protocol, source and destination IP-Addresses and ports [61]; the characterization of attack inter-arrival times [18, 63]; the prediction or forecasting of attack rates [33, 93, 96, 143, 145]; the detection of cyber attacks such as malware and botnets [22, 44, 51, 68, 74, 79, 97–99]; the clustering attacks [19–21, 39]. When compared with these previous studies, we focus on a different aspect, namely cyber attack reconnaissance behaviors with one ultimate goal of understanding the dynamic evolution of attackers in the wild purely based on their reconnaissance behaviors. This aspect could be integrated with the others investigated in the literature to enrich our understanding to cyber attacks. Two other kinds of datasets have been analyzed in the literature as well, although none of these studies analyzed the evolution of cyber attack reconnaissance behaviors. There have been analyses on datasets that are collected at enterprise networks (i.e., neither honeypots nor telescopes) [23, 59]. Thonnard et. al. [119] cluster attacks with the goal of discovering attribution of the attack source. There have been studies on analyzing blackhole-captured cyber attacks (e.g., [88, 125, 130, 144]), but not on the evolution of cyber attack reconnaissance behaviors. On the other hand, Katipally et. al. uses a multi-stage attack detection system to cluster attacker based on their behaviors [64], while leveraging attack payloads, which are not available for cyber attack reconnaissance studies.

3.5 Conclusion

We proposed a methodology for characterizing the evolution of cyber attack reconnaissance behaviors. We observe that for both forecasting cyber attack reconnaissance rates and for quantifying the number of attacker families, the defender needs to consider multiple parameter combinations. The actual number of parameter combinations may be specific to the dataset in question, but identifying a general guiding principle (e.g., what kinds of data would demand what number of parameter combinations) is left for future research.

CHAPTER 4: CHARACTERIZING CYBER ATTACKS RECONNAISSANCE TRAJECTORIES

Chapter Abstract

In this chapter, we characterize cyber attack reconnaissance behaviors through the lens of individual attackers' *reconnaissance trajectory*, which is a novel concept introduced in the present paper. In order to characterize reconnaissance trajectories, we propose a technique called *reconnaissance trajectory hierarchy trees*, which may be of independent value. Experimental results based on some real-world datasets show that on average 3.7K attackers can be grouped into 29 families according to their cyber reconnaissance trajectories. Furthermore, these 29 families can be further divided into four classes: single country and single target service, single country and multiple target services, multiple countries and single target service, and multiple countries and multiple target services. We also identify 44 attackers that exhibit the same attack reconnaissance trajectory hierarchy tree during a five-year time span.

4.1 Introduction

It is an important, yet difficult, task to understand and characterize cyber attacker behaviors from the limited information exposed in the stage of cyber attack reconnaissance. The previous chapter uses temporal information to characterize and cluster attacker-victim behaviors exhibited at the reconnaissance stage. However, in this chapter we move a step further to understand to what extent cyber attackers' temporal-spatial behaviors exhibited at their reconnaissance stage can be used to characterize them, despite the fact that no information about their intent, attack vectors (or payload), or attack tactics is given. Leveraging both temporal and spatial information of cyber reconnaissance behaviors allows us to better understand the families of attack behaviors, leading to richer characterization of cyber threat situational awareness. To the best of our knowledge, the problem of clustering attacker reconnaissance temporal-spatial behaviors is little understood.

We make the following three contributions. First, we initiate the study of characterizing and clustering the temporal-spatial behaviors of cyber attack reconnaissance. Second, we propose a framework for characterizing and clustering the temporal-spatial behaviors. The framework introduces a novel abstraction for the attackers reconnaissance behaviors dubbed: attacker reconnaissance-trajectories. It also provide a visual representation for the cluster of attacker reconnaissance-trajectories dubbed: attacker reconnaissance-trajectories hierarchy trees. In addition, our framework proposes to use the most influence attacker in the families as the families representatives. Third, in order to show the usefulness of the framework, we report a case study of characterizing and clustering the temporal-spatial behaviors based on two low interaction honeypot datasets from the years 2016 and 2019. Some of our findings are highlighted as follows:

- On average 3.7K attackers are divided in twenty-nine families of attacks reconnaissance.
- Attacks reconnaissance families representatives can be organize in four classes: single country and single target service,single country and multiple target service, multiple country and multiple targeted services, and multiple countries and single target service,
- Forty-four attacker exhibit the same attacks reconnaissance trajectory hierarchy tree after five years, suggesting their reconnaissance strategies didn't evolve during that period of time.

Our contributions are summarize in four steps: (i) for each attacker find their reconnaissance-trajectories, (ii) summarize the attacker reconnaissance patterns into reconnaissance-trajectories hierarchy tree, (iii) used the hierarchy tress to compare how similar are a pair attackers and implement a graph base clustering to find the families of attacker given the similarity between the attackers reconnaissance-trajectory hierarchy trees and (iv) for each family of attacker behavior find the most influence attackers in the family as use them as the family representatives.

4.2 Framework

Figures 4.1 highlight the framework which consist in four components: data collection and preprocessing for each heavy hitter attacker, building heavy hitter attacker reconnaissance trajectories, build hierarchy tree to represent the reconnaissance trajectories, and discover the attacker families.

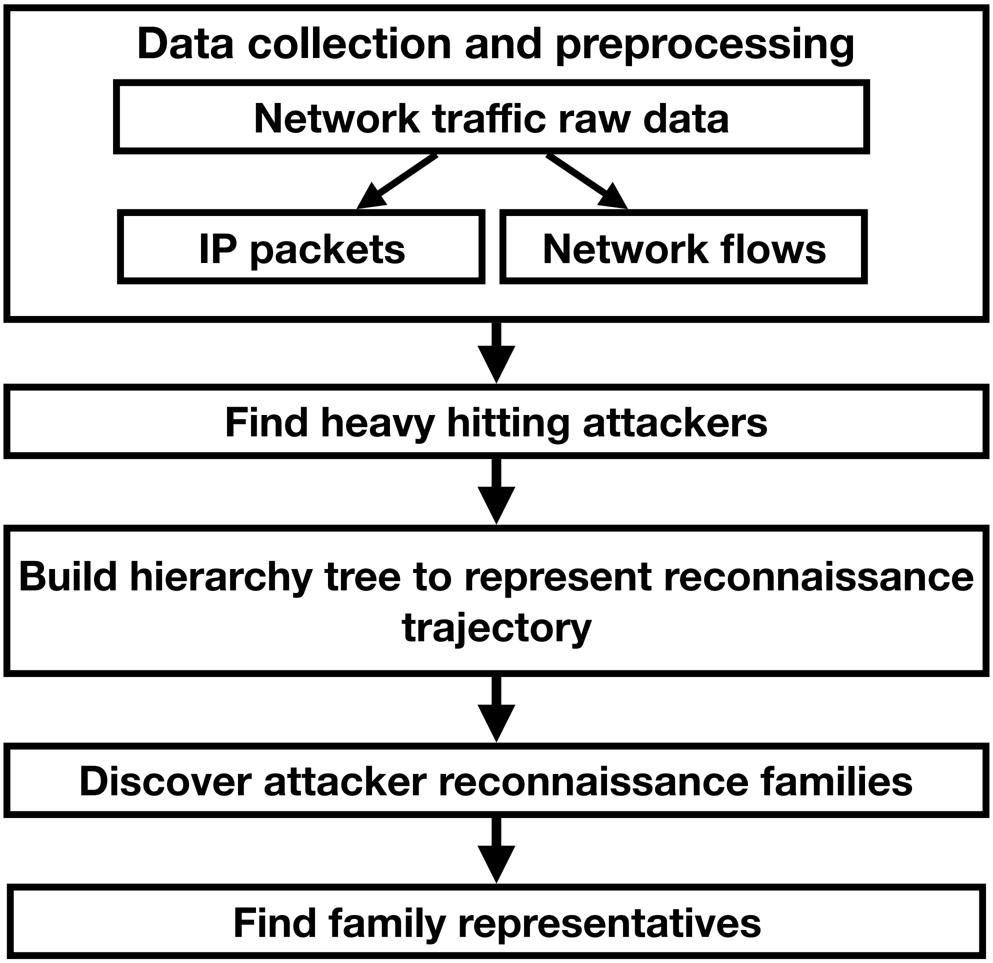

Data collection and preprocessing
Network traffic raw data
IP packets
Network flows
Find heavy hitting attackers
Build hierarchy tree to represent reconnaissance trajectory
Discover attacker reconnaissance families
Find family representatives

Figure 4.1: Framework for characterizing cyber attack reconnaissance trajectories.

4.2.1 Data collection and pre-processing

Is common to capture network traffic as raw Packet Capture Data (PCAP), which can be further converted into IP packets or flows. A flow is commonly defined as a tuple of five fields: *source IP address*, *destination IP address*, *source port*, *destination port*, and *protocol* [36]. Each flow has a start time and end time. There is a standard routines to ensemble PCAP into network flows or IP Packets [54]. For reassembling flows, two parameters are needed the *idle time* and *lifetime*. The *idle time* specifies when a flow should be terminated after observing no communication activities between a source and a destination. The *lifetime* specifies when a flow should be terminated and a new flow may be started. In this chapter, we propose using on flow-based representation to cluster the cyber attack reconnaissance trajectories.

4.2.2 Heavy-hitter attackers

Let $L = [0, t)$ denote the lifespan of a data set. Define Ω be the ID domain of all observed attackers during the time interval L. In theory the cardinality $|\Omega|$ can be as large as 2^{32}, while in our study we observe $|\Omega| \approx 4.4$ millions. We propose to model the attacker reconnaissance tactics as the sequence of ordered flows dubbed, the trajectory.

Definition 20 (Flow)**.** *A flow is defined as the tuple* $f = \langle s, a_s, a_t, p_s, p_t, \rho, \tau, e\rangle$ *where* $f[s]$ *denote the start time of the flow,* $f[a_s]$ *the source IP-Address,* $f[a_t]$ *the target IP-Address,* $f[p_s]$ *the source port,* $f(p_t)$ *the target port,* $f[\rho]$ *the protocol,* $f[\tau]$ *the TCP flags, and* $f[e]$ *the end time of the flow.*

Definition 21 (an attackers' reconnaissances attempts)**.** *Let* v *be an arbitrary attacker in* Ω*. All the attacker* v *reconnaissance attempts are denoted as a sequence of flows* $F_v = \{f_1, f_2, \ldots, f_n\}$ *where* $f_i[a_s] = v$*,* $f_i[s] \in L$ *for all* $f_i \in F_v$*. In addition, for all* $f_i[s] < f_j[s]$ *for any* $f_i, f_j \in F_v$ *and* $i < j$*.*

Let $|F_v|$, the cardinality of the attacker v reconnaissance tour, denote the number of flows from v during the time interval L. Because some attacker might have a very short reconnaissance, we only focus in the heavy hitter attackers which are defined as follows.

Definition 22 (Heavy-hitter attacker). *For a given number of flows α, an attacker $v \in \Omega$ is a heavy hitter attacker if $|F_v| > \alpha$.*

We denote the set of heavy hitter attackers Ω^H, where

$$\Omega^H = \{v | v \in \Omega \wedge |F_v| > \alpha\}$$

4.2.3 Building hierarchy trees for representing reconnaissance trajectories

For a heavy hitter attacker $v \in \Omega^H$ with reconnaissance tour F_v, we construct the attacker reconnaissance trajectories as follows.

Definition 23 (Attacker reconnaissance trajectories). *For an attacker v, a reconnaissance trajectory $T_j \subseteq F_v$ is defined as a set of K_j consecutive flows $T_j = \{f_i, \ldots, f_{i+K_j}\}$ such that $\forall f_i \in T_j \setminus f_{i+K_j}$,*

$$f_i[s] - f_{i+1}[s] \leq \mu_v^F,$$

where μ_v^F is the average idle time between any consecutive flows in F_v. An adjacent trajectory T_{j+1} after T_j with K_{j+1} flows is $T_{j+1} = \{f_{i+K_{j+1}}, \ldots, f_{i+K_j+K_{j+1}}\}$ such that

$$f_i[s] - f_{i+1}[s] > \mu_v^F.$$

Finally, the set of reconnaissance trajectories for attacker v is denoted by $\mathcal{T}_v = \{T_1, \ldots, T_m\}$ where $T_i \cap T_j = \emptyset$ for any $T_i, T_j \in \mathcal{T}_v$.

Attacker reconnaissance trajectory hierarchy tree nodes description

For each heavy hitter attacker $v \in \Omega^H$, we abstracted the reconnaissance trajectories hierarchy through a tree with five hierarchy node levels: the root (Level-0), the child nodes for the target ports (Level-1), the child nodes for the protocol (Level-2), the child nodes for the `tcp` flags (Level-3) and the leaves (Level-4). Each node contains a subset of the attacker reconnaissance trajectories. Nodes are annotated with a node type (*e.g.* root, target ports, protocol, `tcp` flags and leaf nodes), a label, a set of attacker reconnaissance trajectories and a parent node. These nodes are denoted by $\mathcal{T}_v^{\text{RN}}$, $\mathcal{T}_v^{\text{TP}}$, $\mathcal{T}_v^{\text{P}}$, $\mathcal{T}_v^{\text{T}}$, and $\mathcal{T}_v^{\text{L}}$, respectively are defined as follows.

Definition 24 (Root node)**.** *The root node of the tree contains all the reconnaissance trajectories in $\mathcal{T}_v$ and is denoted by $\mathcal{T}_v^{RN}$.*

Definition 25 (Target ports nodes)**.** *The target ports nodes split the reconnaissance trajectories according to the target ports such that the flows of the reconnaissance trajectories in each child node of the target ports nodes belongs to the same set of target ports. For an attacker v with reconnaissance trajectories set of $\mathcal{T}_v$, the target ports node splits trajectories of v into N_{TP} child nodes: $\mathcal{T}_v^{TP}(1), \ldots, \mathcal{T}_v^{TP}(N_{TP})$. For each child node $\mathcal{T}_v^{TP}(k)$, for $k = 1, \ldots, N_{TP} - 1$, define*

$$\mathbb{F}_v^{TP}(k) \equiv \bigcup_{T_j \in \mathcal{T}_v^{TP}(k)} T_j, \quad \text{and} \quad \mathbb{S}_v^{TP}(k) \equiv \bigcup_{f_i \in \mathbb{F}_v^{TP}(k)} f_i(p_t)$$

to be the set that contains all flows in the reconnaissance trajectories and the set of appeared target ports in $\mathcal{T}_v^{TP}(k)$. Then, each node $\mathcal{T}_v^{TP}(k)$ can be labeled by $\mathbb{S}_v^{TP}(k)$. For $k = N_{TP}$, let

$$\mathbb{S}_v^{TP}(N_{TP}) = \left\{ T_j | T_j \in \left(\mathcal{T}_v^{RN} \setminus \bigcup_{k=1}^{N_{TP}-1} \mathcal{T}_v^{TP}(k) \right) \right\}$$

and label it as MIX_v^{TP}.

Definition 26 (Protocol nodes)**.** *The protocol ports nodes, children of the target ports nodes, split the reconnaissance trajectories according to the protocol ports such that the flows of the reconnaissance trajectories in each child node of the target ports nodes belongs to the same set of protocol ports and target ports. The protocol ports node further split trajectories of an attacker v in each target ports node $\mathcal{T}_v^{TP}(k)$ into N_P child nodes: $\mathcal{T}_v^{TP,P}(k, 1), \ldots, \mathcal{T}_v^{TP,P}(k, N_P)$. For each child node $\mathcal{T}_v^{TP,P}(k, l)$, for $l = 1, \ldots, N_P - 1$, define*

$$\mathbb{F}_v^{P}(k,l) \equiv \bigcup_{T_j \in \mathcal{T}_v^{TP,P}(k,l)} T_j, \quad \text{and} \quad \mathbb{S}_v^{P}(k,l) \equiv \bigcup_{f_i \in \mathbb{F}_v^{P}(k,l)} f_i(\rho)$$

to be the set that contains all flows in the reconnaissance trajectories and the set of appeared protocols in $\mathcal{T}_v^{TP,P}(k,l)$. Then, each node $\mathcal{T}_v^{TP,P}(k,l)$ can be labeled by $\mathbb{S}_v^P(k,l)$. For $l = N_{TP}$, let

$$\mathbb{S}_v^{TP}(k, N_P) = \left\{ T_j | T_j \in \left(\mathcal{T}_v^{TP}(k) \setminus \bigcup_{l=1}^{N_P - 1} \mathcal{T}_v^{TP,P}(k,l) \right) \right\}$$

and label the node $\mathcal{T}_v^{TP,P}(k, N_P)$ as MIX$_v^P$.

Definition 27 (The `tcp` flag nodes)**.** *The `tcp` flag nodes, children of the protocol ports nodes, split the reconnaissance trajectories according to the `tcp` flags such that the flows of the reconnaissance trajectories in each child node of the target ports nodes belongs to the same set of `tcp` flags, protocol ports, and target ports. The `tcp` flag nodes further split trajectories of an attacker v in each protocol ports node $\mathcal{T}_v^{TP,P}(k,l)$ into N_T child nodes: $\mathcal{T}_v^{TP,P,T}(k,l,1), \ldots, \mathcal{T}_v^{TP,P,T}(k,l,N_T)$. For each child node $\mathcal{T}_v^{TP,P,T}(k,l,s)$, for $s = 1, \ldots, N_T - 1$, define*

$$\mathbb{F}_v^T(k,l,s) \equiv \bigcup_{T_j \in \mathcal{T}_v^{TP,P,T}(k,l,s)} T_j, \textit{ and } \mathbb{S}_v^T(k,l,s) \equiv \bigcup_{f_i \in \mathbb{F}_v^P(k,l,s)} f_i(\rho)$$

to be the set that contains all flows in the reconnaissance trajectories and the set of appeared `tcp` flags in $\mathcal{T}_v^{TP,P,T}(k,l,s)$. Then, each node $\mathcal{T}_v^{TP,P,T}(k,l,s)$ can be labeled by $\mathbb{S}_v^T(k,l,s)$. For $s = N_T$, let

$$\mathbb{S}_v^T(k,l,N_T) = \left\{ T_j | T_j \in \left(\mathcal{T}_v^{TP,P}(k,l) \setminus \bigcup_{s=1}^{N_T - 1} \mathcal{T}_v^{TP,P,T}(k,l,s) \right) \right\}$$

and label the node $\mathcal{T}_v^{TP,P,T}(k,l,N_T)$ as MIX$_v^T$.

Following the aforementioned splitting rules, the reconnaissance trajectories are grouped together according to their target ports, protocols, and `tcp` flags. To understand the spatial patterns of the attacks in the network space (or IP address space), we further cluster the reconnaissance trajectories in each `tcp` node based on their target IP address (victim), target port, and flow starting time.

Definition 28 (Reconnaissance trajectory similarity graph)**.** *For each `tcp` flag node $\mathcal{T}_v^{TP,P,T}(k,l,s)$, we define a weighted and complete graph $G_v(k,l,s) = \{\Omega(k,l,s), E, N\}$ where $\Omega(k,l,s)$ is the* *vertex set containing all the reconnaissance trajectories in* $\mathcal{T}_v^{TP,P,T}(k,l,s)$*,* $E = T_i \times T_j$ *is the edge set, and* $N : E \to [0,1]$ *is weight such that* $N(i,j)$ *denote the similarity between the reconnaissance trajectories* T_i *and* T_j *according to an appropriate definition.*

Due to the fact that there are many ways to define $N(i,j)$, we use an ensemble clustering approach to combine multiple existent clustering methods. Let $\mathbb{M}$ denote a set of clustering methods, where $|\mathbb{M}| \geq 1$. Suppose $m \in \mathbb{M}$ discover r_m clusters of reconnaissance trajectories in $\mathcal{T}_v^{\text{TP,P,T}}(k,l,s)$. Let $c_m(T_i) \in \{1,\ldots,r_m\}$ denote the clustering assignment for the trajectory T_i. For a pair of trajectories (T_i, T_j), we define their similarity as the fraction of the same clusters to which they belong, namely:

$$N(i,j) = \frac{1}{\mathbb{M}} \sum_{m\in\mathbb{M}} \mathbf{1}\{c_m(T_i), c_m(T_j)\} \tag{4.1}$$

where

$$\mathbf{1}\{x,y\} = \begin{cases} 1 & \text{if } x = y; \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

In order to get the final clustering of the reconnaissance trajectories in $\mathcal{T}_v^{\text{TP,P,T}}(k,l,s)$, we apply a community detection algorithm [27, 37, 103, 108, 109] based on the weighted graph $G_v(k,l,s)$. Assume that there are $R(k,l,s)$ number of clusters retrieved by the community detection algorithm, and let $C(T_i) \in \{1,\ldots,R(k,l,s)\}$ denote the cluster identity of the reconnaissance trajectory T_i.

Definition 29 (Leaf nodes)**.** *The leaf nodes contain the reconnaissance trajectories based on the clustering results of reconnaissance trajectories in the* `tcp` *nodes using their target IP address (victim) and flow starting time such that for each leaf node* $\mathcal{T}_v^{TP,P,T,L}(k,l,s,q)$

$$C(T_j) = q, \text{ for all } T_j \in \mathcal{T}_v^{TP,P,T,L}(k,l,s,q),$$

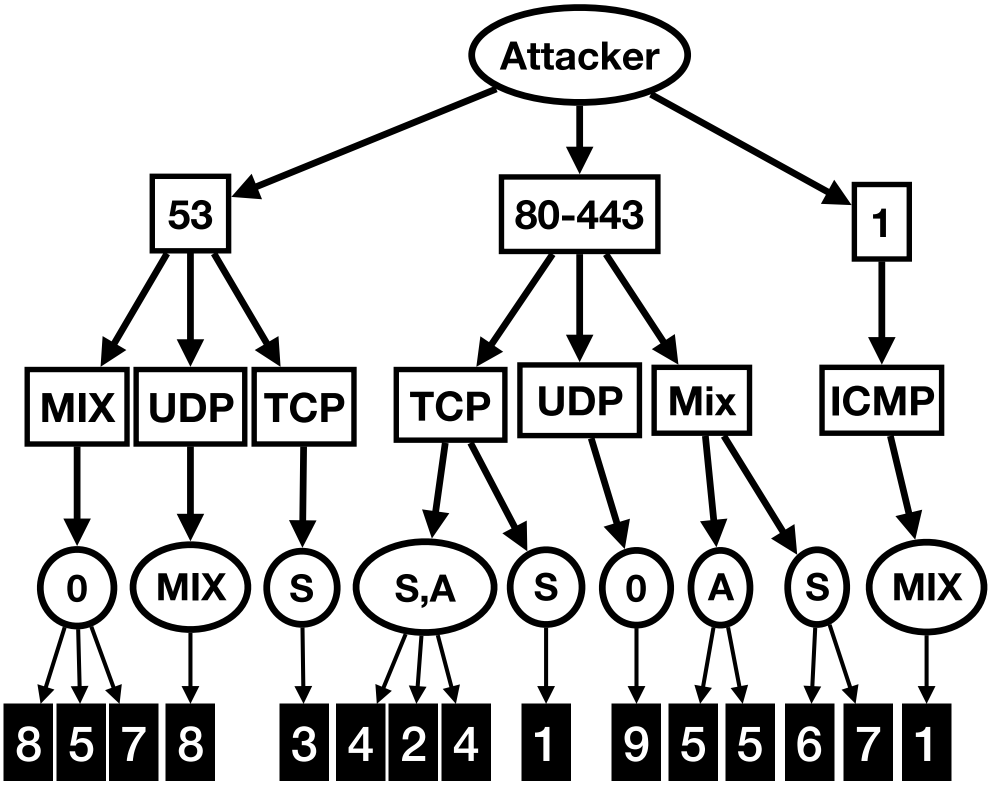


Figure 4.2: Example diagram for the attacker hierarchy tree. The bottom level represents the clustering of the reconnaissance trajectories. Each leaf represent a cluster. The leaf label represents the number of trajectories in the cluster e.g. the node with label “S,A” have three leaves representing three clusters with 4, 2 and 4 trajectories respectively.

and is labeled by q where $q \in \{1, \ldots, R(k,l,s)\}$.

In Figure 4.2 we show a diagram of the reconnaissance trajectories hierarchy tree.

Attacker reconnaissance trajectory hierarchy tree description

Using the attacker reconnaissance trajectory hierarchy tree nodes descriptions we create a directed graph where the different hierarchy tree nodes compose the vertex set and the edge set are the connections between the parent and children nodes.

Definition 30 (Attacker reconnaissance trajectory hierarchy tree)**.** *Define the directed graph $\mathcal{H}_v = \{\mathbb{V}, \mathbb{E}\}$ with the vertex set containing all the nodes in the hierarchy tree:*

$$\mathbb{V} = \{\mathcal{T}_v^{RT}, \mathcal{T}_v^{TP}(k), \mathcal{T}_v^{TP,P}(k,l), \mathcal{T}_v^{TP,P,T}(k,l,s), \mathcal{T}_v^{TP,P,T,L}(k,l,s,q)\}$$

for $1 \leq k \leq N_{TP}$, $1 \leq l \leq N_P$, $1 \leq s \leq N_T$, *and* $1 \leq q \leq R(k,l,s)$*; and the edge set* $\mathbb{E}$ *containing edges connecting the parent and child nodes in* $\mathbb{V}$.

4.2.4 Attacker families and families representatives

Attacker families

We propose to use a graph based clustering approach to discover the attacker families. We define a undirected and weighted graph $\mathcal{G} = \{\Omega^H, \mathcal{E}, \mathcal{N}\}$ using the k-nearest neighbor approach [106]. Let Ω^H be the set of heavy hitter attacker or vertices and $\mathcal{E} \subset \Omega^H \times \Omega^H$ the set of edges where $(v, u) \in \Omega^H$ if $\mathcal{H}_u$ is in the k-most similar reconnaissance trajectory hierarchy tree to $\mathcal{H}_v$ and the weight of the edges are annotated with the normalized ([0,1] range) similarity between the attacker hierarchy trees:

$$\mathcal{N}(u, v) = \frac{1}{1 + \delta(\mathcal{H}_v, \mathcal{H}_u)} \tag{4.3}$$

where $\delta(\mathcal{H}_v, \mathcal{H}_u)$ defines the distance between the reconnaissance trajectory hierarchy trees $\mathcal{H}_v$ and $\mathcal{H}_u$ according to a proper definition, for our case study we select the APTED [89, 90] tree distance because its fast execution time.

Definition 31 (Attacker trajectory reconnaissance family). *Let* Q *be the number of communities in* $\mathcal{G}$ *detected by a community detection algorithm (e.g. [37]), define* $\mathcal{C}(v) \in \{1, \ldots, Q\}$ *the community assignment of the attacker* $v \in \Omega^H$*. Then the* i*-th attacker reconnaissance family is:*

$$\mathbb{C}_i = \{v | \mathcal{C}(v) = i \wedge v \in \Omega^H\}.$$

Attacker families representatives

For each attacker trajectory reconnaissance family $\mathbb{C}_i$, for $1 \leq i \leq Q$, we select four family representatives based on degree, betweenness score [50], closeness score [50] and pagerank score [87]. The betweenness score [50] for vertex $v \in \mathbb{C}_i$ measures the number of shortest path from any pair of nodes (different than v) that passes through node v. The closeness score [50] for vertex $v \in \mathbb{C}_i$

measures the average length of the shortest path from v to any other vertex $v' \in \mathbb{C}_i$. While the page-rank score [87] of $v \in \mathbb{C}_i$ intuitively measure the importance or relevance of v.

For each attacker trajectory reconnaissance family $\mathbb{C}_i$, the four family representatives are selected by the attackers' reconnaissance trajectory hierarchy trees that maximize each of the four measures respectively. For our case study we used the implementations of the python `iGraph` library [40]. Intuitively the aforementioned four metrics define the influence of the attacker to trajectory reconnaissance family. Because the attacker influence is associated to the similarity between the hierarchy tree of the attacker and its neighbors, attackers with the optimal influence score are likely to better describe the rest of the attacker in the family.

4.2.5 Research Questions

Building on top of the preceding descriptive model, we investigate the following research questions (RQs):

RQ1: How many families of attacker reconnaissance trajectories in a given dataset? What are their geographical characteristics?

RQ2: What are the characteristics of each family?

RQ3: Are there any attackers with the same reconnaissance behaviors in year 2014 and 2019?

4.3 Case study and Results

4.3.1 Data collection and pre-processing

Our case study is based on two datasets collected at a low-interaction honeypot, which has 1,024 IP addresses. First dataset (D1) collected data between 2/6/2014 to 5/8/2014 and the second dataset (D2) collect data from 12/31/2018 to 4/30/2019. Although D1 is several years old, it is sufficient for demonstrating the usefulness of our framework. The honeypot runs the *Honeyd* [100] and *Nepenthes* [24] programs. Since a honeypot offers no legitimate services, the traffic coming to a honeypot is deemed as malicious (see, e.g., [22, 44, 51, 68, 79, 96–99, 143, 145]). We convert the raw PCAP dataset into IPFIX network flows by using the tools known as Yet Another Flowmeter (YAF) and super_mediator of the Computer Emergency Response Taskforce (CERT) [62]. As in many previous studies, the *idle time* is set to 60 seconds and the flow *lifetime* is set to 300 seconds (see, for example, [96, 143, 145]).

Dataset D1

The dataset led to 92,477,692 flows each of which is treated as a reconnaissance activity. Among them 74% of the flows have zero duration time (meaning that on average they correspond to single packet reconnaissance activities). Among the 26% of the flows with non-zero duration time (indicating multiple-packet reconnaissance activities), 50% of them have a duration time that is less than 0.7 seconds, indicating that most reconnaissance behaviors are essentially scanning/probing activities. Table 4.1 summarizes the basic statistics of the flows in two groups (i.e., non-zero vs. zero duration time), where "# of packets (bytes)" means the number of packets (bytes) of an individual flow.

Table 4.1: Basic statistics of reconnaissance activities of dataset D1, where μ is the average and σ the standard deviation.

		Basic Statistics				
		min	μ	median	max	σ
Flows with non-zero duration time	flow duration	0.001	16.8	0.69	300	1185.6
	# of packets	2	3.7	3	550	12.5
	# of bytes	56	248	167	41,220	157,195
Flows with zero duration time	# of packets	1	1	1	65	0.2
	# of bytes	28	51	40	2,600	1,552.3

Dataset D2

The dataset led to 455,465,974 flows each of which is treated as a reconnaissance activity. Among them 74% of the flows have zero duration time (meaning that on average they correspond to single packet reconnaissance activities). Among the 26% of the flows with non-zero duration time (indicating multiple-packet reconnaissance activities), 50% of them have a duration time that is less than 0.7 seconds, indicating that most reconnaissance behaviors are essentially scanning/probing activities. Table 4.2 summarizes the basic statistics of the flows in two groups (i.e., non-zero vs. zero duration time), where “# of packets (bytes)” means the number of packets (bytes) of an individual flow.

Table 4.2: Basic statistics of reconnaissance activities of dataset D2, where μ is the average and σ the standard deviation.

		Basic Statistics				
		min	μ	median	max	σ
Flows with non-zero duration time	flow duration	0.001	146.5	256.9	300	15616.9
	# of packets	2	5.8	6	7895	22.9
	# of bytes	56	621.7	342	5,746,064	188,049.1
Flows with zero duration time	# of packets	1	1	1	14	0.02
	# of bytes	28	69	57	20,028	4,442.3

4.3.2 Find heavy-hitter attackers

To find the heavy hitter attacker we considered $\alpha = 100$ for the dataset D1 and we denoted the set of heavy hitter attackers in D1 as Ω^H_{D1} where $|\Omega^H_{D1}| = 3,089$. We used $\alpha = 5,000$ for dataset D2 and denote the set of heavy hitter attacker in D2 as Ω^H_{D2}, where $|\Omega^H_{D2}| = 4,437$.

4.3.3 Building Attacker reconnaissance trajectories hierarchy tree and Attacker Families

We used Definition 23 to build the attacker reconnaissance trajectories and Definition 30 to build the reconnaissance trajectories hierarchy trees for each attacker $v \in \Omega^H$. To generate the hierarchy tree leaves we represent the trajectories as a three-dimensional trajectory using the flow start time, target port and destination IP address. Then we define the closeness between a pair trajectories using the trajectory distance metrics: Symmetric Hausdorff distance [118], the Dynamic Time Warping (DTW) [83, 112, 120], and the Soft-DTW [42]. We combine the distance metrics with the clustering algorithms: k-Means, Agglomerative Clustering algorithm (AGNES) single linkage, AGNES average linkage, AGNES complete linkage and the Affinity Propagation clustering algorithm. To choose the clustering algorithm parameters, we select the parameter that optimize the silhouette score. A total of fifteen clustering methods ($|\mathbb{M}| = 15$) where used for Equation 4.1. To find the communities of the similarity graph we used the multi level communities detection algorithm [27].

4.3.4 Find Attacker Families

To compute the similarity between a pair of attackers we used Equation 4.3 where we let $\delta(\mathcal{H}_v, \mathcal{H}_u)$ be the APTED tree edit distance implementation [89, 90]. The APTED tree edit distance required tree penalty cost: (i) penalty of adding (or removing) a node, (ii) penalty of adding (or removing) an edge, and (iii) the penalty of renaming (or re-labeling) a node. For the penalty of adding node or edges we used the default cost of one. However we customized the rename penalty as follow:

Example 32 (Node rename penalty cost)**.** *For node a_v in hierarchy tree $\mathcal{H}_v$ and node a_u in hierarchy tree $\mathcal{H}_u$ denote the cost of renaming a_v and a_u as $rc(a_v, a_u)$ define as:*

$$rc(a_v, a_u) = \begin{cases} \phi(a_v, a_u) & \textit{if } a_v \equiv \mathcal{T}_v^{\textit{RN}} \wedge a_u \equiv \mathcal{T}_u^{\textit{RN}} \\ \psi(a_v, a_u) & \textit{if } a_v \equiv \mathcal{T}_v^{\textit{TP}}(k) \wedge a_u \equiv \mathcal{T}_u^{\textit{TP}}(k) \\ \psi(a_v, a_u) & \textit{if } a_v \equiv \mathcal{T}_v^{\textit{TP,P}}(k) \wedge a_u \equiv \mathcal{T}_u^{\textit{TP,P}}(k) \\ \psi(a_v, a_u) & \textit{if } a_v \equiv \mathcal{T}_v^{\textit{TP,P,T}}(k) \wedge a_u \equiv \mathcal{T}_u^{\textit{TP,P,T}}(k) \\ \eta(a_v, a_u) & \textit{if } a_v \equiv \mathcal{T}_v^{\textit{TP,P,T,L}}(k) \wedge a_u \equiv \mathcal{T}_u^{\textit{TP,P,T,L}}(k) \\ 1 & \textit{otherwise} \end{cases} \tag{4.4}$$

where

$$\psi(x, y) = \frac{|x \cup y| - |x \cap y|}{|x \cup y|} \textit{ and } \eta(x, y) = \frac{|x - y|}{\max(\{x, y\})}$$

and the cost of renaming the root nodes is:

$$\phi(x, y) = \frac{\texttt{gdist}(x, y) + \texttt{idist}(x, y)}{2}$$

where `gdist` is the normalized geographical distance and `idist` is IP-space normalized distance of the attacker IP addresses.

Table 4.3: Dataset D1 statistical summary of the families number of members (#MEMB) and number of unique attacker countries (#UCTS). Where μ represents the average and σ the standard deviation.

	μ	σ	min	25%	50%	75%	max
#MEMB	99.65	73.27	16	37	106	126	358
#UCTS	27.52	17.48	1	11	31	42	53

To choose the k parameter for the k-nearest neighbor graph $\mathcal{G}$ we used the theorem by Brito *et al.* [28] and used $k = 2\ln(|\Omega^H|)$.

For each family of attacker trajectory reconnaissance we computed its four representatives. For some families the topology of the hierarchy tree are same among some of the family representatives. Therefore, in the results we choose to illustrate only the unique hierarchy tree topology among the four family representatives.

4.3.5 Dataset D1 Results

Experimental Results with respect to RQ1

We found that the 3,089 attackers can be divide in 31 families. Table 4.3 shows the basic statistics of the number of attackers in each family and the number of unique attacker countries in each family. The attacker reconnaissance families in average has 73 members from 17 different countries. The largest family has 358 members and the family geographically more diversified contains members from 53 different countries.

Experimental Results with respect to RQ2

On average, families are represented with 100 attackers from 28 countries. However, we found five families in where all the attacker are from the same country. The countries of this five families are Brazil, Colombia, India, Turkey, and United States. Brazil with 17 attackers and two unique hierarchy tree family representatives shown in Figure 4.3a and Figure 4.3b. Colombia with 16 attackers and one unique hierarchy tree family representative show in Figure 4.3c. India with 19 attackers and one unique hierarchy tree family representative shown in Figure 4.3d. Turkey with 17 attackers

Table 4.4: Dataset D2 statistical summary of the families number of members (#MEMB) and number of unique attacker countries (#UCTS). Where μ represent the average and σ the standard deviation.

	μ	σ	min	25%	50%	75%	max
#MEMB	164.33	120.19	18.0	79.5	140.0	197.5	479.0
#UCTS	28.19	20.45	1	14.0	25.0	43.5	68.0

and one unique hierarchy tree family representative shown in Figure 4.3e and United States with 22 attackers and three unique hierarchy tree family representatives shown in Figure 4.3f, Figure 4.3g and Figure 4.3h. Among the family representatives the common reconnaissance target were on: the file share service SMB (target port 445), the remote login service SSH (target port 22), HyperText Transfer service HTTP (target port 80), and remote desktop service RDP (target port 3389). Furthermore, we also observed families with attackers from multiple countries that also target a single service including SMB, SSH, HTTP, RDP, also the remote administration service (radmin) in port 4899 and the virtual network computing (VNC) service in port 5900. Two families target multiple target port combinations: attackers in the first family are from China, Germany and Russia targeting 65 port combinations including common reconnaissance strategies such as probing ports 80 and 8080, 23 and 2323, 22 and 2222. The second family attackers geo-location source is shown in Figure 4.5 where the family representatives did reconnaissance in 14 ports combinations shown in Figure 4.4 associated with memory leaks vulnerabilities on Cisco, SMTP worms, multiple trojans including: Antigen, Barok, BSE, Gip, Laocoon, Magic Horse, MBT, Nimda, Shtirlitz, Stukach, Tapiras, WinPC IOS; denial of service vulnerabilities in Point-to-Point Tunneling Protocol Virtual Private Networking (PPTP) and data exfiltration vulnerabilities in the Symantec Endpoint Protection Manager and Symantec Backup Exec System Recovery Manager [13].

4.3.6 Dataset D2 Results

Experimental Results with respect to RQ1

The 4,437 attackers in D2 are divide in 27 families, Table 4.4 shows the basic statistics of the number of attacker and the number of unique countries in each family. The attacker reconnaissance

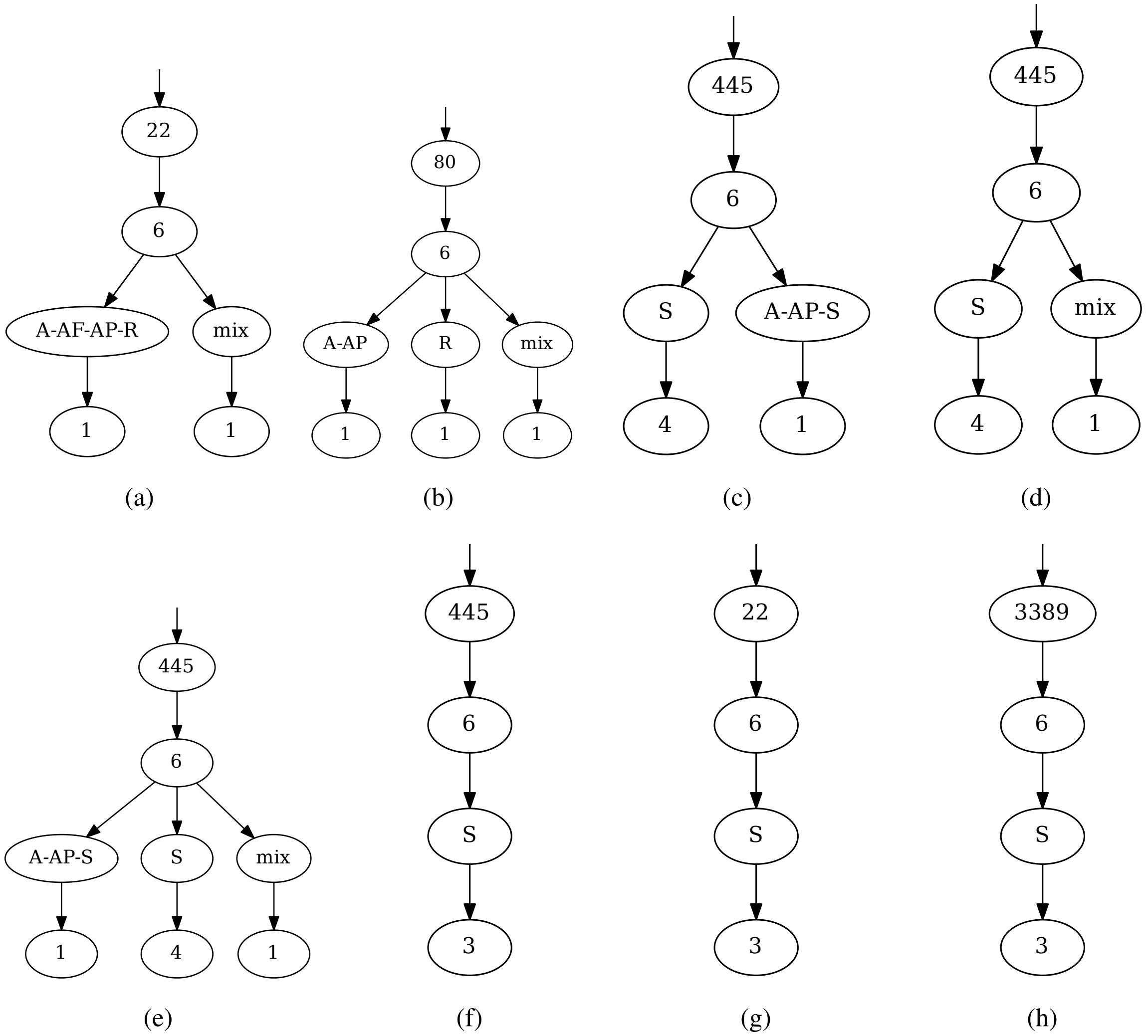


Figure 4.3: Unique hierarchy tree for family representatives. (a)-(b) The representatives for the family of attackers from Brazil, (c) the representative for the family of attackers from Colombia, (d) the representative for the family of attackers from India, (e) the representative for the family of attackers form Turkey and (f)-(h) the representatives from the family of attackers form the United States.

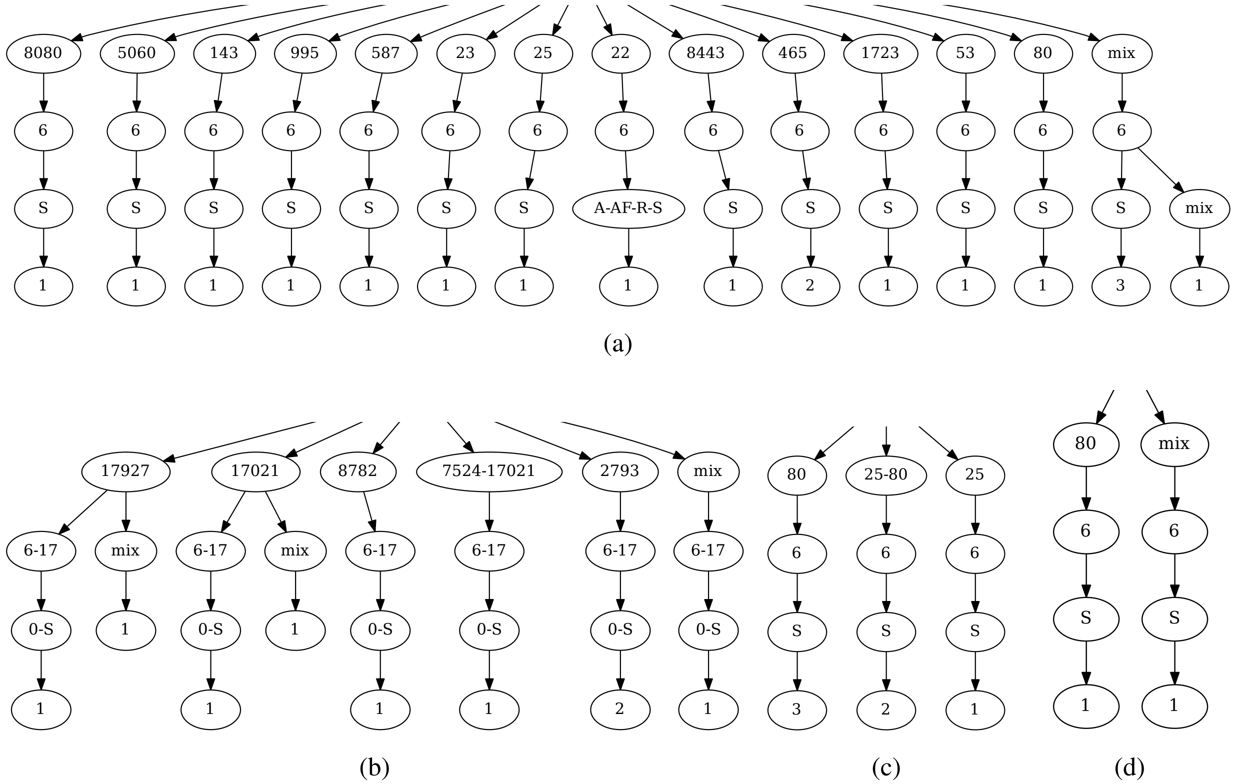


Figure 4.4: Family representatives hierarchy trees: (a) optimal betweenness score, (b) optimal closeness score, (c) largest degree and (d) optimal pagerank score representative.

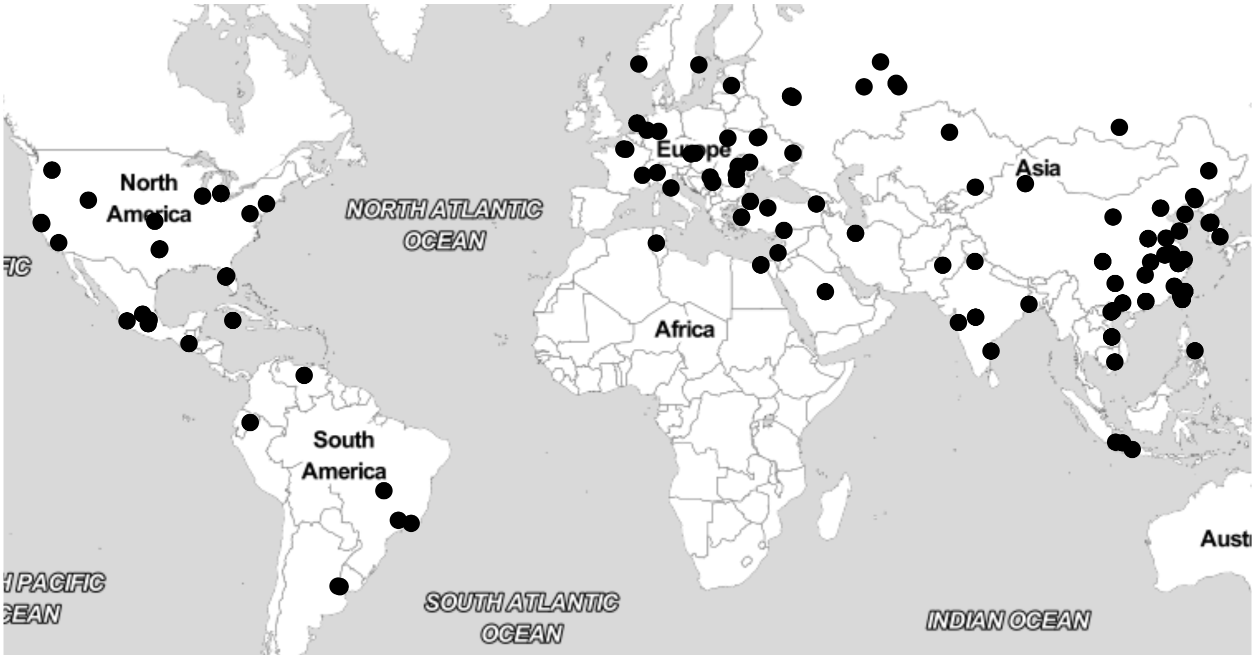


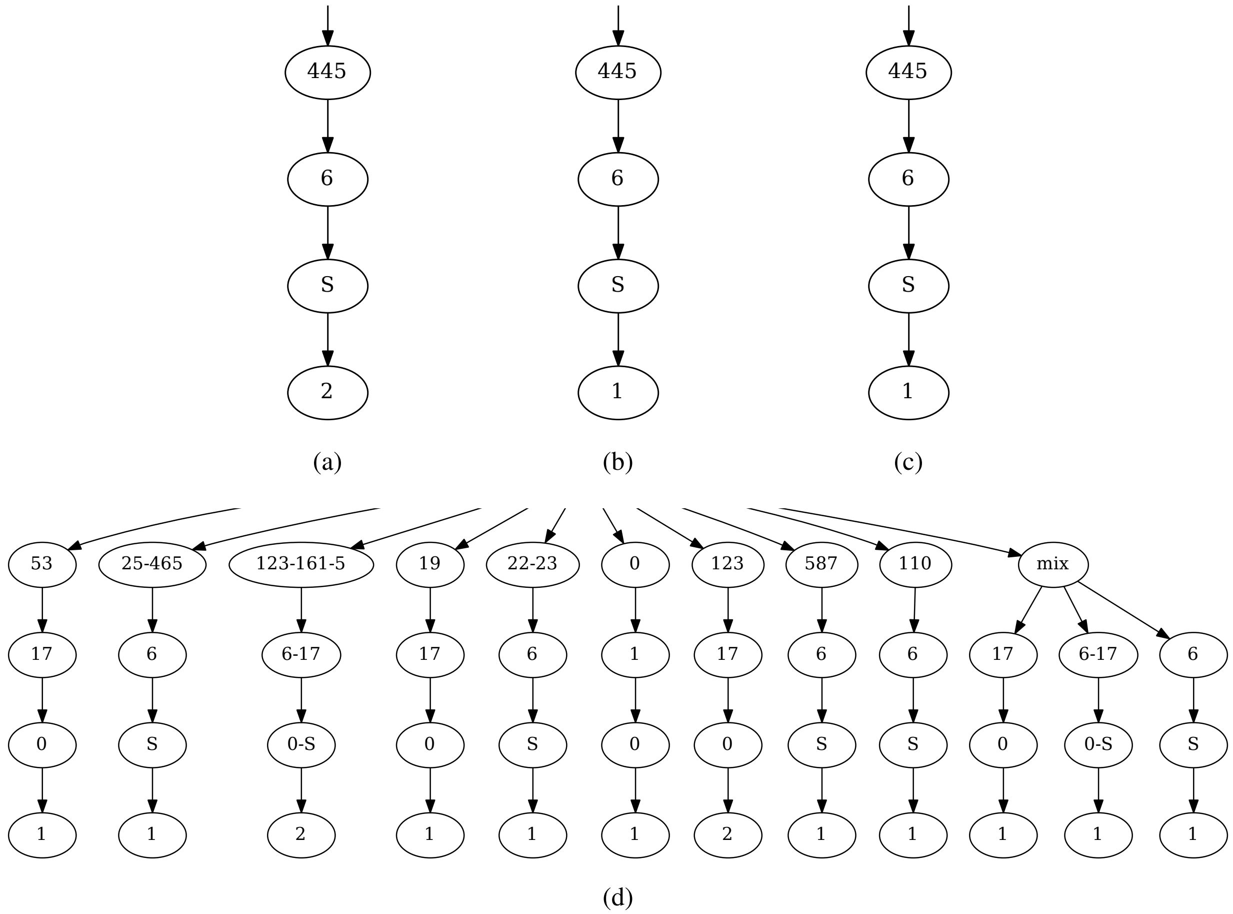
Figure 4.5: Attackers geo-locations.

families in average has 120 members from 20 different countries. The largest family has 479 members and the family geographically more diversified contained members from 68 different countries.

Experimental Results with respect to RQ2

On average there are 164 attackers from 28 different countries in each family. There are four families where all the attackers are from the same countries: (i) a family of 18 attacker from Iran with an unique hierarchy tree family representative shown in Figure 4.6a, (ii) a family with 48 attackers from India with an unique hierarchy tree family representative shown in Figure 4.6b, (iii) a family with 76 attackers from China with an unique hierarchy tree family representative shown in Figure 4.6d and (iv) a family with 33 attackers from Thailand with an unique hierarchy tree family representative shown in Figure 4.6c. The single country families with attacker from Iran, India and Thailand focus the reconnaissance to the SMB service. The single country family with attackers from China targets multiple services including: SMB, SSH, Telnet (target port 23), domain name service (DNS, target port 53), simple mail transfer protocol (SMTP, target port 25, 465 for outgoing SSL encryption and 587 for outgoing TLS encryption), network time protocol (NTP, target port 123), simple network management protocol (SNMP, target port 161), remote job entry target port 5 (also known to be exploit by trojans Incoming Routing Redirect Bomb and yoyo [1,13]) Chargen service target port 19 (also known to be used by the trojan Skun [1,13]) and post office protocol (POP3) on target port 110 (also known to be exploitable by the ADM worm, ProMail trojan, Bancos and Civcat exploits [1,13]).

Among the families with attackers from multiple countries, six families did reconnaissance only in the SMB service, two families did reconnaissance only in the TELNET service targeting target ports 23 and its commonly used variant 2323, five families did reconnaissance only for the HTTP service, two families did reconnaissance only for the SSH service targeting target port 22 and its commonly used variant port 2222. Additionally six families were known to focus their reconnaissance on target ports that are known to be associated with trojan and threats. A family of



(a) (b) (c)

(d)

Figure 4.6: Single country families representatives: (a) Iran, (b) India, (c) Thailand and (d) China.

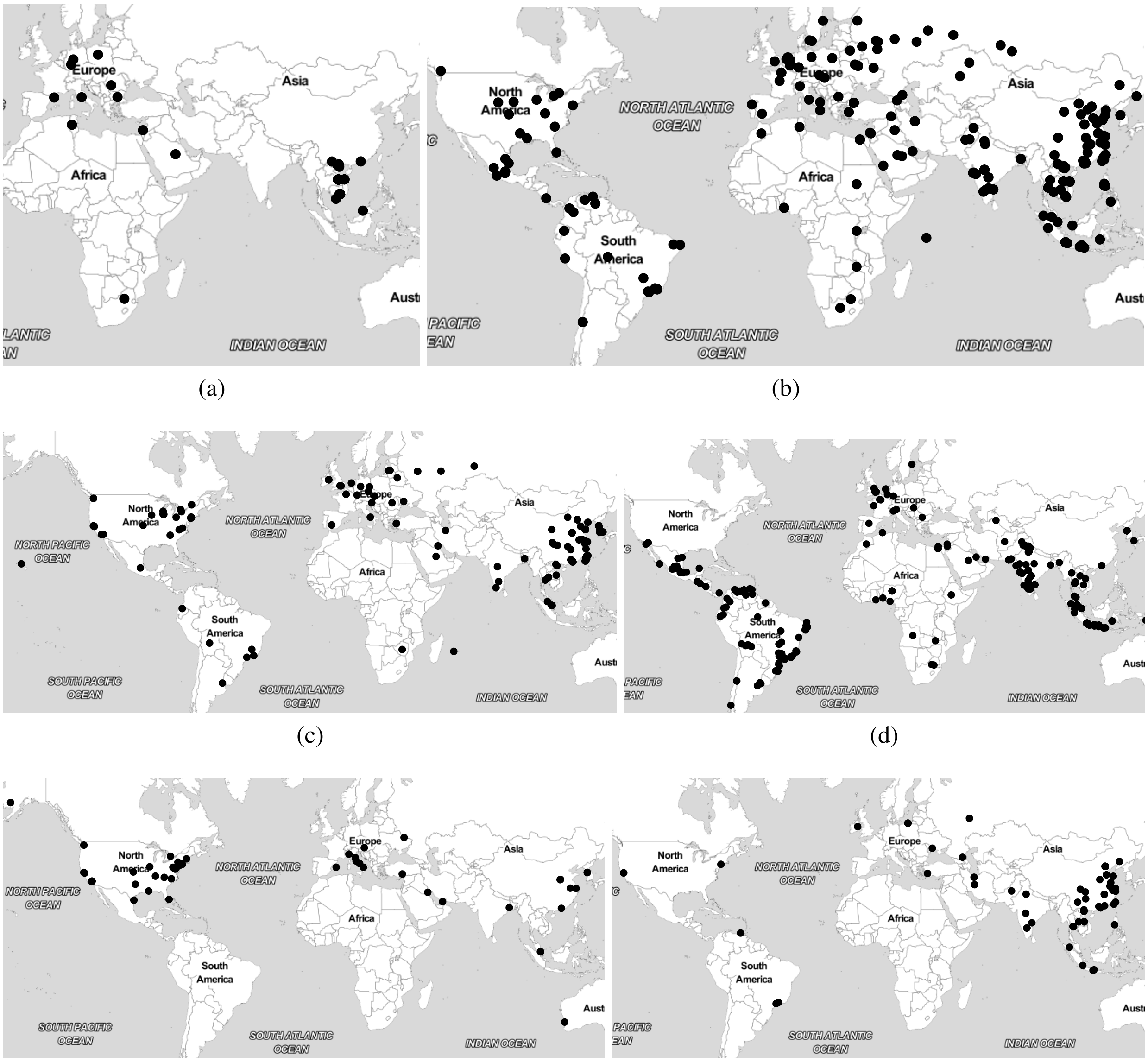
140 attackers from 13 different countries shown in Figure 4.7a did reconnaissance on target port 8033 associated with the RingZero trojan [12, 13]. A family of 312 attackers from 62 different countries shown in Figure 4.7b did reconnaissance on target port 7001 associated with trojans Freak2k, Freak88, and NetSnooper Gold, also with an exploit on the WLS Security component in Oracle WebLogic Server [8,13]. A family of 158 attackers from 40 countries shown in Figure 4.7c did reconnaissance on the target port 8800 associated to the apple address book, the W32.Noomy worm and the arbitrator code execution vulnerability on the Sun Java System Web Server [6,13,14]. A family of 328 attacker from 49 countries shown in Figure 4.7d did reconnaissance on target port 3390 associated with the Backdoor.Dawcun trojan, vulnerabilities on Voice Over IP Phones, and the Unidata Shell [2,4,5,13]. A family of 52 attackers from 15 countries shown in Figure 4.7e did reconnaissance on target port 37215 associated with vulnerabilities in Huawei HG532 routers [9, 11, 13], and a family of 83 attacker from 17 countries shown in Figure 4.7f did reconnaissance on target port 3306 associated two MySQL vulnerabilities, the W32.Spybot.IVQ worm, and the Nemog trojan [3,7,10,13].

Insight 33. *By characterizing the representatives of attacker reconnaissance trajectory families, the defender can reduce analysis load by 96% on average.*

4.3.7 Experimental Results with respect to RQ3

We found that dataset D1 and D2 have two attackers in common from Spain and South Korea Figure 4.8 show the attacker reconnaissance trajectories hierarchy tree in year 2014 targeting the SMB and the radmin services. Figure 4.9 show the attackers trajectory trees in year 2019 where both attacker did reconnaissance in the same service they did in year 2014.

The fact that some attackers show the same reconnaissance trajectory hierarchy tree in different time spams motivate us to explore the intersection of heavy hitter attacker form datasets D1 and D2. For the aforementioned task we compute a second set of heavy hitter attacker in D2 using $\alpha = 100$ and denote it $\Omega^H_{D2.2}$. We found the intersection set of heavy hitter attacker $\Omega^H_I = \Omega^H_{D1} \cap \Omega^H_{D2.2}$, where $|\Omega^H_I| = 45$. Where 44 of the attackers $u \in \Omega^H_I$ have identical reconnaissance trajectory



(a)

(b)

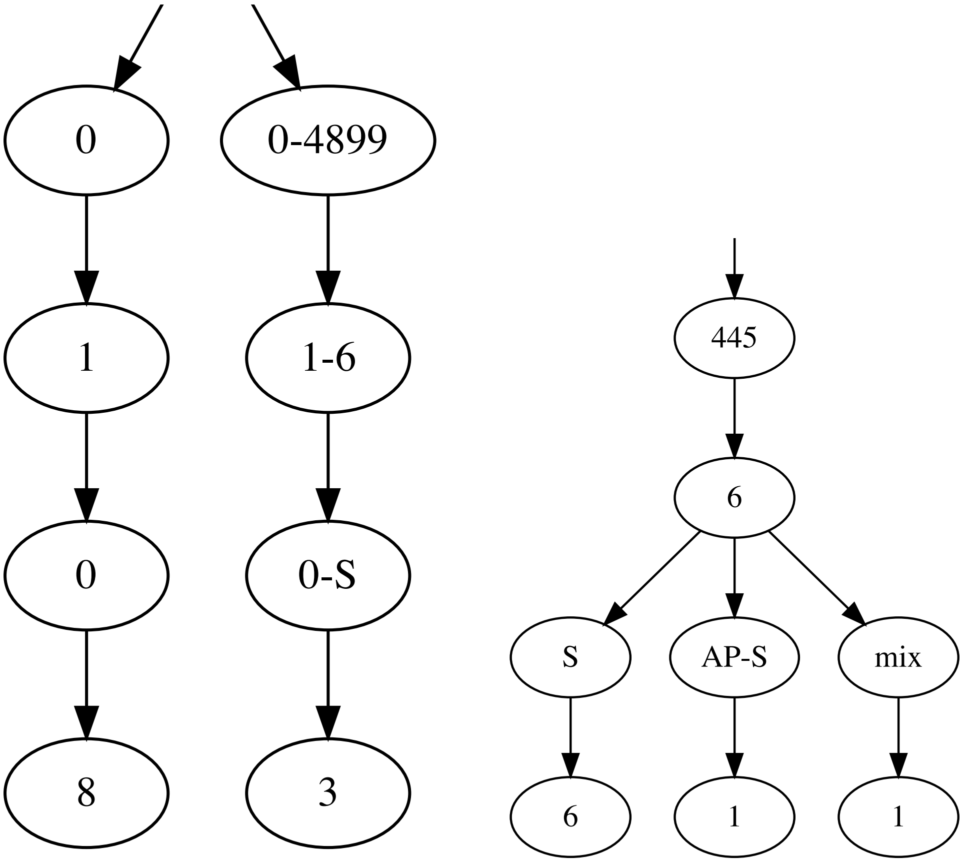
(c)

(d)

(e)

(f)

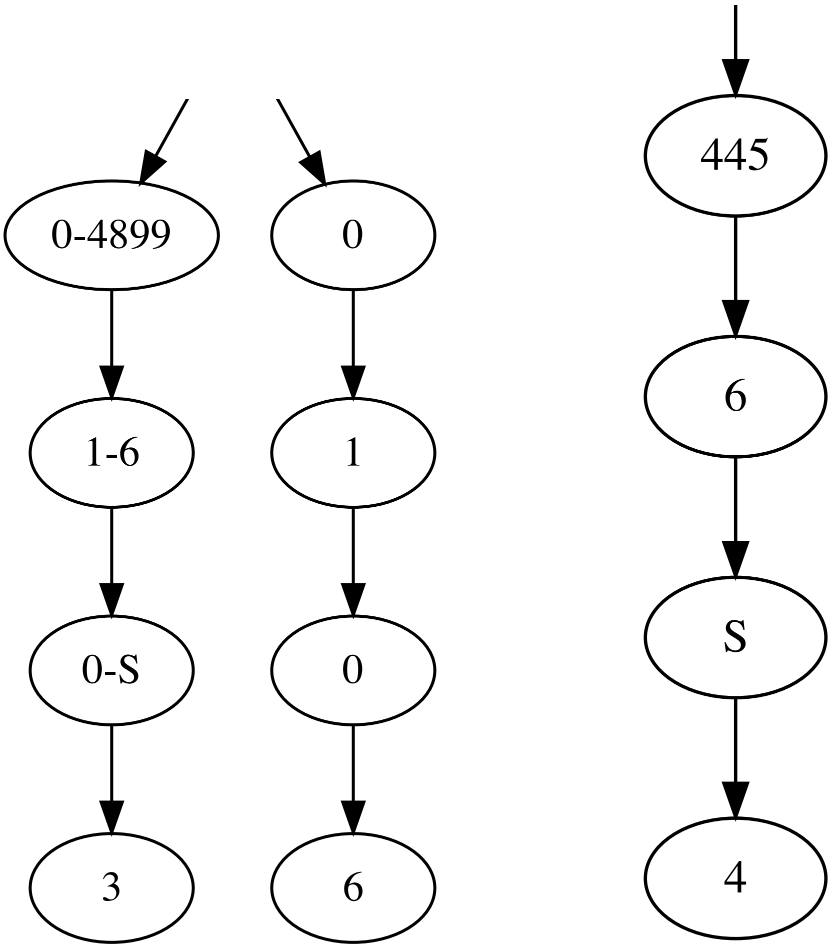
Figure 4.7: Attacker geo-location for families of attacker associated with: (a) the RingZero trojan [12,13] (b) trojans Freak2k, Freak88 and NetSnooper Gold and an exploit on the WLS Security component in Oracle WebLogic Server [8,13], (c) the W32.Noomy worm, and the arbitrator code execution vulnerability on the Sun Java System Web Server [6,13,14], (d) Backdoor.Dawcun trojan, vulnerabilities on Voice Over IP Phones and the Unidata Shell [2,4,5,13] (e) with vulnerabilities in Huawei HG532 routers [9,11,13] and (f) two MySQL vulnerabilities, the W32.Spybot.IVQ worm, and the Nemog trojan [3,7,10,13]



(a)

(b)

Figure 4.8: Attackers reconnaissance trajectory hierarchy trees in dataset D1, year 2014: (a) attacker form South Korea and (b) attacker from Spain.



(a)

(b)

Figure 4.9: Attackers reconnaissance trajectory hierarchy tress in dataset D2, year 2019: (a) attacker form South Korea and (b) attacker from Spain.

Table 4.5: Summary of the number of attackers (#MEMB) and the number of unique attacker countries (#UCTS) for each family in Ω_I^H.

Family Id	#UCTS	#MEMB
1	7	13
2	6	10
3	6	11
4	5	10

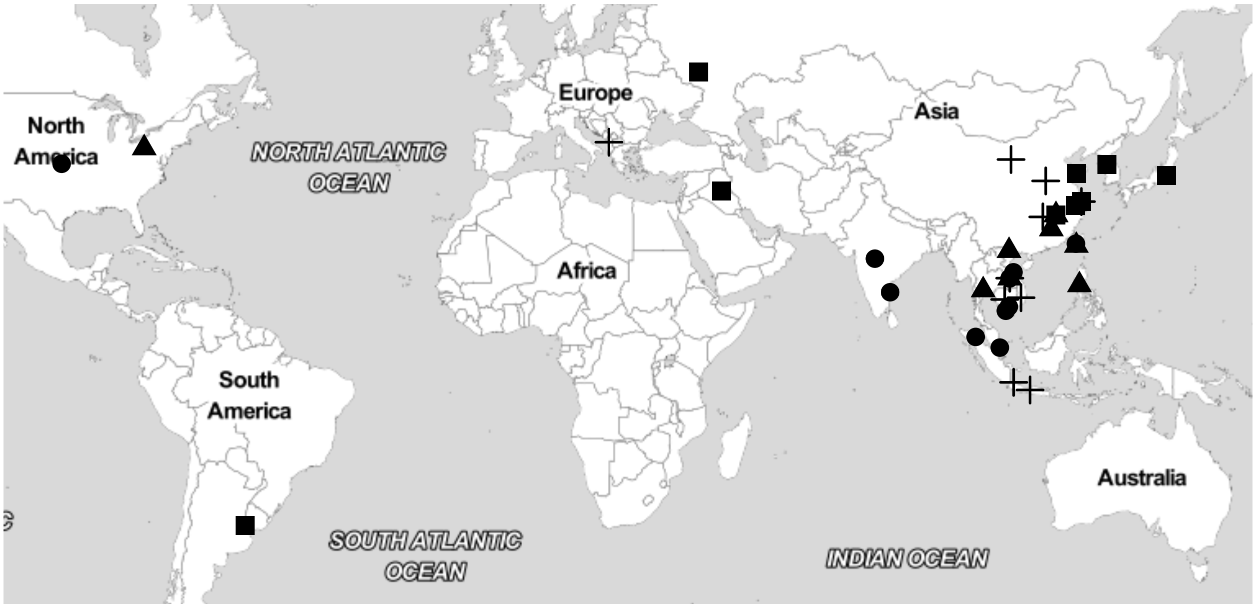
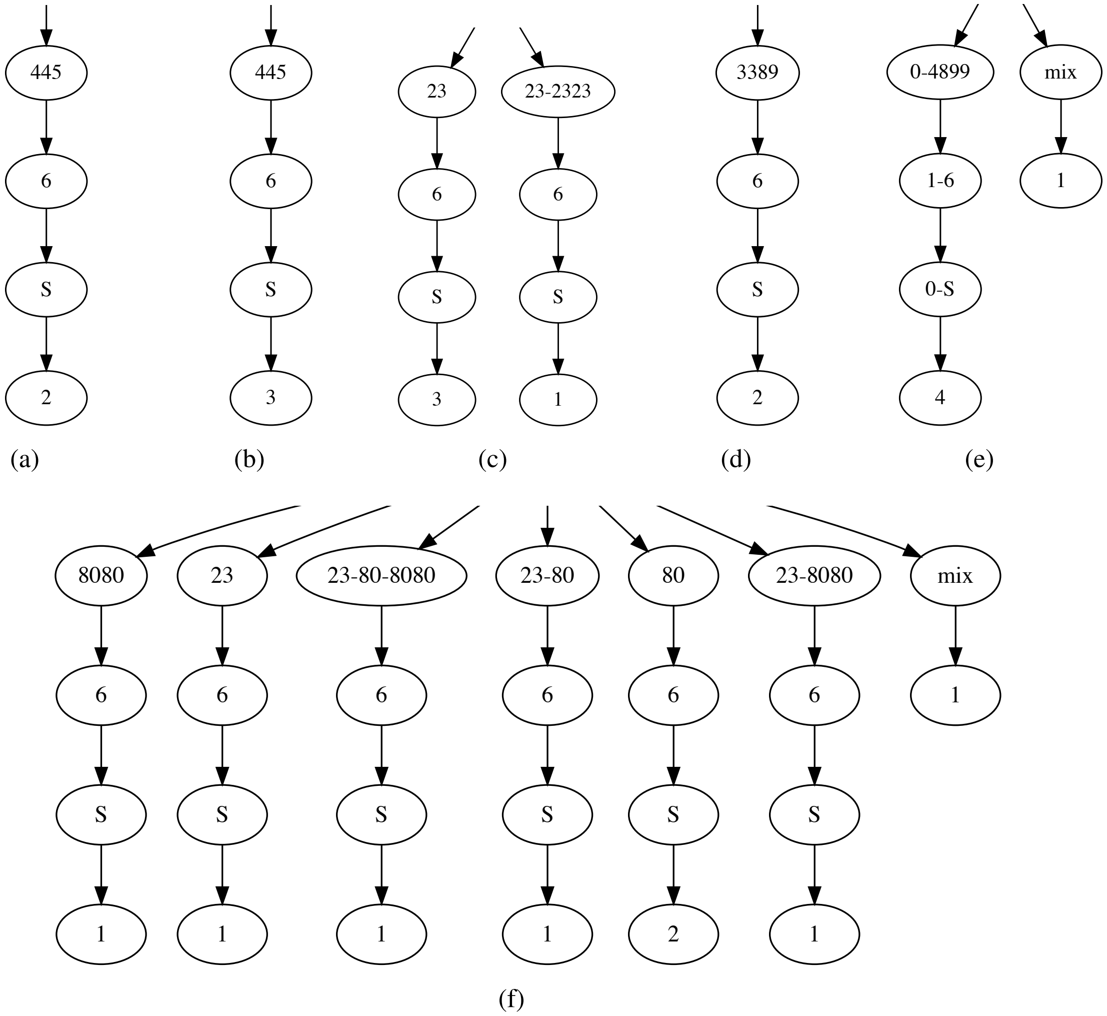


Figure 4.10: Geo-location of attackers for the families in Ω_I^H. Family one label as •, family two label as ▲, family three label as ■ and family four label as +.

hierarchy tree in both period of time. We found four families of attacker in Ω_I^H, Table 4.5 shows the number of attackers in each families and the number of unique attackers countries of each family. In Figure 4.10 we show the geographical information, and Figure 4.11 shows the representative of each family of attacker in Ω_I^H, where the representatives of family one do reconnaissance on the SMB service only, family two representatives target SMB and TELNET services, family three target the RDP services and family four representative focus in HTTP and FTP services.

Insight 34. *Families of attacker reconnaissance trajectories with similar hierarchy trees in 2014 and 2019 are mostly from countries in Asia.*



(a) (b) (c) (d) (e)

(f)

Figure 4.11: Attacker reconnaissance trajectory hierarchy tree family representative for attackers in Ω_I^H. Unique representatives for: (a) family one , (b)-(c) family two, (d) family three and (e)-(f) family four.

4.4 Related Work

The present study contributes to one of the pillars in the Cybersecurity Dynamics framework [33, 91, 93, 96, 130, 134, 135, 143, 145], the field of cybersecurity data analytics. Previous studies in this field include: univariate time series forecasting [33,93,132,143–145], multivariate time series forecasting [96, 130], and graph time series of attacker-victim relations [54]. The present study investigates a new aspect of multivariate time series, namely the *temporal* and *spatial* behaviors of cyber attacker reconnaissance through the lens of trajectories.

From the datasets perspective, low-interaction honeypot data has been analyzed from the following perspectives: the visualization of the attacks source, destination IP addresses and ports [61]; the analysis of attack inter-arrival times [18,63]; the forecasting of attack rates [33,93,96,143,145]; the detection of malware and botnets [22, 44, 51, 68, 74, 79, 97–99]; the clustering attacks [19–21, 39]. When compared with these previous studies, we focus on a different aspect, namely the characterization of cyber attacker reconnaissance trajectories with one ultimate goal of understanding the families of attackers in the wild, purely based on their reconnaissance trajectories trees. This aspect could be integrated with the others investigated in the literature to enrich our understanding of cyber attacks.

Two other kinds of datasets have been analyzed in the literature as well, although none of these studies analyzed the families of attacker reconnaissance trajectories. Datasets collected at enterprise networks have been analyzed in [23, 59]. Thonnard *et. al* [119] cluster attacks feature with the goal of discovering attribution. There has been studies on analyzing blackhole-captured cyber attacks (e.g., [88,125,130,144]), but not on the attackers reconnaissance trajectories tree. On the other hand, Katipally *et. al* uses a multi-stage attack detection system to cluster attackers base on their behavior [64], focusing in dividing the attacker in seven groups. To define the attackers behaviors they use the alerts from the intrusion detection system SNORT, this alert heavily required the attack payload while for our study we focus in the attacker reconnaissance effort *e.g.* probing rates, without knowing the attack payload. In contrast to the aforementioned work we abstract the attacker behavior via trajectory reconnaissance and the attacker trajectory reconnaissance tree which depend solely on the attacks protocols, TCP flags, destination IP address and port and the temporal and spatial patter of the trajectories.

4.5 Conclusion

We presented the novel abstraction of reconnaissance trajectory for characterizing cyber attack reconnaissance behaviors. We further presented on how to organize and represent reconnaissance trajectories into a hierarchy tree, and how to use such trees to cluster attacker reconnaissance behaviors. We applied the methodology to two datasets and found that thousands of attacker can be respectively divide into 31 and 27 families according to their reconnaissance trajectories. We also draw some useful insights from the empirical studies.

CHAPTER 5: CONCLUSION

5.1 Summary

The present dissertation initiated a systematical study of cyber attack reconnaissance behaviors at three levels of abstraction: macroscopic, mesoscopic and microscopic.

- At the macroscopic level, we presented a framework for characterizing the evolution of attacker-victim relation graphs. We also conducted a case study with emphasis on identifying the number of time resolutions to characterize the evolution of attacker-victim relation graphs.
- At the mesoscopic level, we proposed a two-resolution clustering approach to identifying the families of cyber attack reconnaissance behaviors based on cyber attack reconnaissance rates. We investigated the evolution of the attacker families and found the parameter combinations that led to the LRD property. Our case study showed that two sets of parameters are sufficient to provide a comprehensive analysis of cyber attack reconnaissance behaviors.
- At the microscopic level, we proposed a novel abstraction for characterizing cyber attack reconnaissance trajectories. For our case study, we used two datasets.

5.2 Future Work

The present dissertation represents the first step towards ultimately tackling a problem of fundamental importance. Open problems are abundant, such as:

- How many levels of abstractions are necessary and sufficient for obtaining a holistic understanding of cyber attack reconnaissance behaviors?
- What are the properties or factors that determine the number of levels of abstractions for obtaining a holistic understanding of cyber attack reconnaissance behaviors?

- For a given level of abstraction, how many parameter combinations are necessary and sufficient for obtaining a comprehensive cyber attack reconnaissance behaviors?
- For a given level of abstraction, what are the properties or factors that determine the number of parameter combinations that are are necessary and sufficient for obtaining a comprehensive cyber attack reconnaissance behaviors?
- What cyber attacker information can or cannot be derived from cyber attack reconnaissance behaviors given that reconnaissance behaviors do not contain attack payload?

BIBLIOGRAPHY

[1] Admin Subnet security scan. http://www.adminsub.net/tcp-udp-port-finder/trojan. Accessed: 2019-06-10.

[2] Backdoor.Dawcunis trojan. https://www.symantec.com/security-center/writeup/2010-040116-0914-99. Accessed: 2019-06-10.

[3] Backdoor.Nemog trojan. https://www.symantec.com/security-center/writeup/2004-081610-2414-99. Accessed: 2019-06-10.

[4] CVE-2005-3722 vulnerability. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3722. Accessed: 2019-06-10.

[5] CVE-2005-3723 vulnerability. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3723. Accessed: 2019-06-10.

[6] CVE-2010-0388 vulnerability. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0388. Accessed: 2019-06-10.

[7] CVE-2011-5049 vulnerability. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-5049. Accessed: 2019-06-10.

[8] CVE-2015-4852 vulnerability. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4852. Accessed: 2019-06-10.

[9] CVE-2015-7254 vulnerability. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-7254. Accessed: 2019-06-10.

[10] CVE-2016-6531 vulnerability. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-6531. Accessed: 2019-06-10.

[11] CVE-2017-17215 vulnerability. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-17215. Accessed: 2019-06-10.

[12] RingZero trojan. https://www.symantec.com/security-center/writeup/2000-121809-3414-99. Accessed: 2019-06-10.

[13] Speed Guide security scan port 5. https://www.speedguide.net/port.php?port=5. Accessed: 2019-06-10.

[14] W32.Noomy worm. https://www.symantec.com/security-center/writeup/2004-092711-1953-99. Accessed: 2019-06-10.

[15] 2018 cost of a data breach study:global overview, July 2018.

[16] Anatomy of an apt attack: Step by step approach, September 2018.

[17] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review. *Information Systems*, 53:16–38, 2015.

[18] E. Alata, M. Dacier, Y. Deswarte, M. Kaaâniche, K. Kortchinsky, V. Nicomette, V. Pham, and F. Pouget. Collection and analysis of attack data based on honeypots deployed on the internet. In *Proc. Quality of Protection - Security Measurements and Metrics*, pages 79–91, 2006.

[19] S. Almotairi, A. Clark, M. Dacier, C. Leita, G. Mohay, V. Pham, O. Thonnard, and J. Zimmermann. Extracting inter-arrival time based behaviour from honeypot traffic using cliques. In *5th Australian Digital Forensics Conference*, pages 79–87, 2007.

[20] S. Almotairi, A. Clark, G. Mohay, and J. Zimmermann. Characterization of attackers' activities in honeypot traffic using principal component analysis. In *Proc. IFIP International Conference on Network and Parallel Computing*, pages 147–154, 2008.

[21] S. Almotairi, A. Clark, G. Mohay, and J. Zimmermann. A technique for detecting new attacks in low-interaction honeypot traffic. In *Proc. International Conference on Internet Monitoring and Protection*, pages 7–13, 2009.

[22] K. Anagnostakis, S. Sidiroglou, P. Akritidis, K. Xinidis, E. Markatos, and A. Keromytis. Detecting targeted attacks using shadow honeypots. In *Proc. USENIX Security Symposium*, 2005.

[23] Jonathan Z. Bakdash, Steve Hutchinson, Erin G. Zaroukian, Laura R. Marusich, Saravanan Thirumuruganathan, Char Sample, Blaine Hoffman, and Gautam Das. Malware in the future? forecasting analyst detection of cyber events. *CoRR*, abs/1707.03243, 2017.

[24] Egon Balas and Camilo H. Viecco. Towards a third generation data capture architecture for honeynets. *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, pages 21–28, 2005.

[25] Suman Banerjee, Mamata Jenamani, and Dilip Kumar Pratihar. Properties of a projected network of a bipartite network. *CoRR*, abs/1707.00912, 2017.

[26] Sai Bhamidipati. The art of reconnaissance - simple techniques, August 2001.

[27] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, oct 2008.

[28] M. R. Brito, E. L. Chávez, A. J. Quiroz, and J. E. Yukich. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics & Probability Letters*, 35(1):33–42, August 1997.

[29] John Charlton, Pang Du, Jin-Hee Cho, and Shouhuai Xu. Measuring relative accuracy of malware detectors in the absence of ground truth. In *IEEE Military Communication Conference (MILCOM 2018)*, 2018.

[30] Huashan Chen, Jin-Hee Cho, and Shouhuai Xu. Quantifying the security effectiveness of firewalls and dmzs. In *Proceedings of the 5th Annual Symposium on Hot Topics in the Science of Security (HoTSoS'2018)*, pages 9:1–9:11, 2018.

[31] Huashan Chen, Jin-Hee Cho, and Shouhuai Xu. Quantifying the security effectiveness of network diversity: poster. In *Proceedings of the 5th Annual Symposium on Hot Topics in the Science of Security (HoTSoS'2018)*, page 24:1, 2018.

[32] Lingwei Chen, Shifu Hou, Yanfang Ye, and Shouhuai Xu. Droideye: Fortifying security of learning-based classifier against adversarial android malware attacks. In *FOSINT-SI'2018*, pages 253–262, 2018.

[33] Yu-Zhong Chen, Zi-Gang Huang, Shouhuai Xu, and Ying-Cheng Lai. Spatiotemporal patterns and predictability of cyberattacks. *PLoS One*, 10(5):e0124472, 05 2015.

[34] Jin-Hee Cho, Packtrick Hurley, and Shouhuai Xu. Metrics and measurement of trustworthy systems. In *IEEE Military Communication Conference (MILCOM 2016)*, 2016.

[35] Jin-Hee Cho, Shouhuai Xu, Patrick M. Hurley, Matthew Mackay, Trevor Benjamin, and Mark Beaumont. Stram: Measuring the trustworthiness of computer-based systems. *ACM Comput. Surv.*, 51(6):128:1–128:47, 2019.

[36] Kimberly C. Claffy, H-W Braun, and George C. Polyzos. A parameterizable methodology for internet traffic flow profiling. *IEEE Journal on selected areas in communications*, 13(8):1481–1494, 1995.

[37] Aaron Clauset, M E J Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 70:066111, 01 2005.

[38] Thomas F Coleman and Jorge J Moré. Estimation of sparse jacobian matrices and graph coloring blems. *SIAM journal on Numerical Analysis*, 20(1):187–209, 1983.

[39] G. Conti and K. Abdullah. Passive visual fingerprinting of network attack tools. In *Proc. 2004 ACM workshop on Visualization and data mining for computer security*, pages 45–54, 2004.

[40] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[41] Marco Cuturi. Fast global alignment kernels. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 929–936, USA, 2011. Omnipress.

[42] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 894–903. JMLR. org, 2017.

[43] Gaofeng Da, Maochao Xu, and Shouhuai Xu. A new approach to modeling and analyzing security of networked systems. In *Proceedings of the 2014 Symposium on the Science of Security (HotSoS'14)*, pages 6:1–6:12, 2014.

[44] D. Dagon, X. Qin, G. Gu, W. Lee, J. Grizzard, J. Levine, and H. Owen. Honeystat: Local worm detection using honeypots. In *Proc. Recent Advances in Intrusion Detection (RAID'04)*, pages 39–58, 2004.

[45] Pang Du, Zheyuan Sun, Huashan Chen, Jin-Hee Cho, and Shouhuai Xu. Statistical estimation of malware detection metrics in the absence of ground truth. *IEEE Trans. Information Forensics and Security*, 13(12):2965–2980, 2018.

[46] Giora Engel. Deconstructing the cyber kill chain, November 2014.

[47] Xing Fang, Maochao Xu, Shouhuai Xu, and Peng Zhao. A deep learning framework for predicting cyber attacks rates. *EURASIP J. Information Security*, 2019:5, 2019.

[48] Eric Ficke, Kristin M. Schweitzer, Raymond M. Bateman, and Shouhuai Xu. Characterizing the effectiveness of network-based intrusion detection systems. In *2018 IEEE Military Communications Conference, MILCOM 2018, Los Angeles, CA, USA, October 29-31, 2018*, pages 76–81, 2018.

[49] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[50] Linton C Freeman. conceptual clarification." social networks. *"Centrality in social networks*, 1(3):215–239, 1978.

[51] Y. Gao, Z. Li, and Y. Chen. A dos resilient flow-level intrusion detection approach for high-speed networks. In *Proc. IEEE ICDCS'06*, 2006.

[52] Richard Garcia-Lebron, David J. Myers, Shouhuai Xu, and Jie Sun. Node diversification in complex networks by decentralized coloring. Journal of Complex Networks, 2018.

[53] Richard Garcia-Lebron, Kristin Schweitzer, Raymond Bateman, and Shouhuai Xu. A framework for characterizing the evolution of cyber attacker-victim relation graphs. In *IEEE Milcom'2018*. 2018.

[54] Richard B Garcia-Lebron, Kristin M Schweitzer, Raymond M Bateman, and Shouhuai Xu. A framework for characterizing the evolution of cyber attacker-victim relation graphs. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 70–75. IEEE, 2018.

[55] Swati Goswami, CA Murthy, and Asit K Das. Sparsity measure of a network graph: Gini index. *Information Sciences*, 462:16–39, 2018.

[56] Adam Hahn, Roshan K Thomas, Ivan Lozano, and Alvaro Cardenas. A multi-layered and kill-chain based security analysis framework for cyber-physical systems. *International Journal of Critical Infrastructure Protection*, 11:39–50, 2015.

[57] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[58] Yujuan Han, Wnelian Lu, and Shouhuai Xu. Characterizing the power of moving target defense via cyber epidemic dynamics. In *Proc. 2014 Symposium on the Science of Security (HotSoS'14)*, pages 10:1–10:12, 2014.

[59] Richard E. Harang and Alexander Kott. Burstiness of intrusion detection process: Empirical evidence and a modeling approach. *IEEE Trans. Information Forensics and Security*, 12(10):2348–2359, 2017.

[60] Kristin E Heckman, Frank J Stech, Roshan K Thomas, Ben Schmoker, and Alexander W Tsow. *Cyber denial, deception and counter deception*. Springer, 2015.

[61] A. Herrero, U. Zurutuza, and E. Corchado. A neural-visualization ids for honeynet data. *Int. J. Neural Syst.*, 22(2), 2012.

[62] Christopher Inacio and Brian Trammell. Yaf: Yet another flowmeter. In *LISA*, 2010.

[63] M. Kaâniche, Y. Deswarte, E. Alata, M. Dacier, and V. Nicomette. Empirical analysis and statistical modeling of attack processes based on honeypots. *CoRR*, abs/0704.0861, 2007.

[64] Rajeshwar Katipally, Li Yang, and Anyi Liu. Attacker behavior analysis in multi-stage attack detection system. In Frederick T. Sheldon, Robert K. Abercrombie, and Axel W. Krings, editors, *CSIIRW*, page 63. ACM, 2011.

[65] Hyeob Kim, HyukJun Kwon, and Kyung Kyu Kim. Modified cyber kill chain model for multimedia service environments. *Multimedia Tools and Applications*, 78(3):3153–3170, 2019.

[66] Dennis Kiwia, Ali Dehghantanha, Kim-Kwang Raymond Choo, and Jim Slaughter. A cyber kill chain based taxonomy of banking trojans for evolutionary computational intelligence. *Journal of computational science*, 27:394–409, 2018.

[67] Danai Koutra, Neil Shah, Joshua T. Vogelstein, Brian Gallagher, and Christos Faloutsos. Deltacon: Principled massive-graph similarity function with attribution. *ACM Trans. Knowl. Discov. Data*, 10(3):28:1–28:43, February 2016.

[68] C. Kreibich and J. Crowcroft. Honeycomb: creating intrusion detection signatures using honeypots. *SIGCOMM Comput. Commun. Rev.*, 34(1):51–56, 2004.

[69] Max Kuhn. The caret package, 2009.

[70] D. Li, Q. Li, Y. Ye, and S. Xu. Enhancing robustness of deep neural networks against adversarial malware samples: Principles, framework, and aics'2019 challenge. In *AAAI-2019 Workshop on Artificial Intelligence for Cyber Security (AICS'2019)*, 2019.

[71] Deqiang Li, Ramesh Baral, Tao Li, Han Wang, Qianmu Li, and Shouhuai Xu. Hashtran-dnn: A framework for enhancing robustness of deep neural networks against adversarial malware samples. *CoRR*, abs/1809.06498, 2018.

[72] X. Li, T. Parker, and S. Xu. Towards quantifying the (in)security of networked systems. In *Proc. of IEEE International Conference on Advanced Information Networking and Applications (AINA'07)*, pages 420–427, 2007.

[73] Xiaohu Li, Paul Parker, and Shouhuai Xu. A stochastic model for quantitative security analyses of networked systems. *IEEE Transactions on Dependable and Secure Computing*, 8(1):28–43, 2011.

[74] Z. Li, A. Goyal, Y. Chen, and V. Paxson. Towards situational awareness of large-scale botnet probing events. *Information Forensics and Security, IEEE Transactions on*, 6(1):175–188, march 2011.

[75] Zhen Li, Deqing Zou, Shouhuai Xu, Hai Jin, Hanchao Qi, and Jie Hu. Vulpecker: an automated vulnerability detection system based on code similarity analysis. In *Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC 2016, Los Angeles, CA, USA, December 5-9, 2016*, pages 201–213, 2016.

[76] Zhen Li, Deqing Zou, Shouhuai Xu, Hai Jin, Yawei Zhu, Zhaoxuan Chen, Sujuan Wang, and Jialai Wang. Sysevr: A framework for using deep learning to detect software vulnerabilities. *CoRR*, abs/1807.06756, 2018.

[77] Zhen Li, Deqing Zou, Shouhuai Xu, Xinyu Ou, Hai Jin, Sujuan Wang, Zhijun Deng, and Yuyi Zhong. Vuldeepecker: A deep learning-based system for vulnerability detection. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*, 2018.

[78] Zongzong Lin, Wenlian Lu, and Shouhuai Xu. Unified preventive and reactive cyber defense dynamics is still globally convergent. *IEEE/ACM Trans. Netw.*, 27(3):1098–1111, 2019.

[79] C. Livadas, R. Walsh, D. Lapsley, and W. Strayer. Using machine learning techniques to identify botnet traffic. In *Proc. IEEE LCN Workshop on Network Security (WoNS'2006)*, pages 967–974, 2006.

[80] Wenlian Lu, Shouhuai Xu, and Xinlei Yi. Optimizing active cyber defense dynamics. In *Proceedings of the 4th International Conference on Decision and Game Theory for Security (GameSec'13)*, pages 206–225, 2013.

[81] Jose David Mireles, Jin-Hee Cho, and Shouhuai Xu. Extracting attack narratives from traffic datasets. In *2016 International Conference on Cyber Conflict, CyCon U.S. 2016, Washington, DC, USA, October 21-23, 2016*, pages 118–123, 2016.

[82] Jose David Mireles, Eric Ficke, Jin-Hee Cho, Patrick Hurley, and Shouhuai Xu. Metrics towards measuring cyber agility. IEEE Transaction on Information Forensics & Security, 2019 (accepted for publication).

[83] M Müller. *Dynamic Time Warping*, pages 69–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[84] David M. Nicol, William H. Sanders, and Kishor S. Trivedi. Model-based evaluation: From dependability to security. *IEEE Trans. Dependable Sec. Comput.*, 1(1):48–65, 2004.

[85] Steven Noel and Sushil Jajodia. *A Suite of Metrics for Network Attack Graph Analytics*, pages 141–176. Springer International Publishing, Cham, 2017.

[86] Günce Keziban Orman, Vincent Labatut, and Hocine Cherifi. Comparative evaluation of community detection algorithms: a topological approach. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(08):P08001, aug 2012.

[87] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[88] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of internet background radiation. In *Proc. ACM Internet Measurement Conference (IMC'04)*, pages 27–40, 2004.

[89] Mateusz Pawlik and Nikolaus Augsten. Efficient computation of the tree edit distance. *ACM Transactions on Database Systems (TODS)*, 40(1):3, 2015.

[90] Mateusz Pawlik and Nikolaus Augsten. Tree edit distance: Robust and memory-efficient. *Information Systems*, 56:157–173, 2016.

[91] Marcus Pendleton, Richard Garcia-Lebron, Jin-Hee Cho, and Shouhuai Xu. A survey on systems security metrics. *ACM Comput. Surv.*, 49(4):62:1–62:35, December 2016.

[92] Marcus Pendleton and Shouhuai Xu. A dataset generator for next generation system call host intrusion detection systems. In *2017 IEEE Military Communications Conference, MILCOM 2017, Baltimore, MD, USA, October 23-25, 2017*, pages 231–236, 2017.

[93] Chen Peng, Maochao Xu, Shouhuai Xu, and Taizhong Hu. Modeling and predicting extreme cyber attack rates via marked point processes. *Journal of Applied Statistics*, 0(0):1–30, 2016.

[94] Chen Peng, Maochao Xu, Shouhuai Xu, and Taizhong Hu. Modeling and predicting extreme cyber attack rates via marked point processes. *Journal of Applied Statistics*, 44(14):2534–2563, 2017.

[95] Chen Peng, Maochao Xu, Shouhuai Xu, and Taizhong Hu. Modeling multivariate cybersecurity risks. *Journal of Applied Statistics*, 0(0):1–23, 2018.

[96] Chen Peng, Maochao Xu, Shouhuai Xu, and Taizhong Hu. Modeling multivariate cybersecurity risks. *Journal of Applied Statistics*, 0(0):1–23, 2018.

[97] V. Pham and M. Dacier. Honeypot trace forensics: The observation viewpoint matters. *Future Generation Comp. Syst.*, 27(5):539–546, 2011.

[98] I. Polakis, T. Petsas, E. Markatos, and S. Antonatos. A systematic characterization of im threats using honeypots. In *NDSS*, 2010.

[99] G. Portokalidis and H. Bos. Sweetbait: Zero-hour worm detection and containment using low- and high-interaction honeypots. *Comput. Netw.*, 51(5), 2007.

[100] Niels Provos. A virtual honeypot framework. In *Proc. USENIX Security Symposium*, 2004.

[101] Zhongjun Qu. A test against spurious long memory. *Journal of Business and Economic Statistics*, 29(3):423–438, 2011.

[102] Zhongjun Qu. A test against spurious long memory. *Journal of Business and Economic Statistics*, 29(3):423–438, 2011.

[103] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76:036106, Sep 2007.

[104] A. Ramos, M. Lazar, R. H. Filho, and J. J. P. C. Rodrigues. Model-based quantitative network security metrics: A survey. *IEEE Communications Surveys Tutorials*, 19(4):2704–2734, 2017.

[105] W D Ray and J Beran. Statistics for Long-Memory Processes. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 159(1):180, 1996.

[106] P RESNICK. Anopen architecture for collaborative filterring of netnews. In *Proc CSCW'94*, 1994.

[107] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, 2007.

[108] M. Rosvall, D. Axelsson, and C. T. Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, Nov 2009.

[109] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[110] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[111] J. R. Rutherford and G. B. White. Using an improved cybersecurity kill chain to develop an improved honey community. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 2624–2632, Jan 2016.

[112] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, February 1978.

[113] Moustafa Saleh, Tao Li, and Shouhuai Xu. Multi-context features for detecting malicious programs. *J. Computer Virology and Hacking Techniques*, 14(2):181–193, 2018.

[114] Moustafa Saleh, E. Paul Ratazzi, and Shouhuai Xu. A control flow graph-based signature for packer identification. In *2017 IEEE Military Communications Conference*, pages 683–688, 2017.

[115] Gennady Samorodnitsky. Long range dependence. *Found. Trends. Stoch. Sys.*, 1(3):163–257, January 2007.

[116] Alexis Sardá-Espinosa. Comparing time-series clustering algorithms in r using the dtwclust package. *Vienna: R Development Core Team*, 2017.

[117] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, March 2003.

[118] Abdel Aziz Taha and Allan Hanbury. An efficient algorithm for calculating the exact hausdorff distance. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2153–2163, 2015.

[119] Olivier Thonnard, Wim Mees, and Marc Dacier. On a multicriteria clustering approach for attack attribution. *SIGKDD Explorations*, 12(1):11–20, 2010.

[120] Paolo Tormene, Toni Giorgino, Silvana Quaglini, and Mario Stefanelli. Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. *Artificial Intelligence in Medicine*, 45(1):11 – 34, 2009.

[121] Hristos Tyralis and Demetris Koutsoyiannis. Simultaneous estimation of the parameters of the hurst–kolmogorov stochastic process. *Stochastic Environmental Research and Risk Assessment*, 25(1):21–33, Jan 2011.

[122] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1073–1080, New York, NY, USA, 2009. ACM.

[123] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[124] Diethelm Wuertz, Tobias Setz, and Yohan Chalabi. Modelling arma time series processes: The farma package, 2017.

[125] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Huston. Internet background radiation revisited. In *Proc. ACM Internet Measurement Conference (IMC'10)*, pages 62–74, 2010.

[126] Li Xu, Zhenxin Zhan, Shouhuai Xu, and Keying Ye. Cross-layer detection of malicious websites. In *Third ACM Conference on Data and Application Security and Privacy (ACM CODASPY'13)*, pages 141–152, 2013.

[127] Li Xu, Zhenxin Zhan, Shouhuai Xu, and Keying Ye. An evasion and counter-evasion study in malicious websites detection. In *IEEE Conference on Communications and Network Security (CNS'14)*, pages 265–273, 2014.

[128] M. Xu, K. M. Schweitzer, R. M. Bateman, and S. Xu. Modeling and predicting cyber hacking breaches. *IEEE Transactions on Information Forensics and Security*, 13(11):2856–2871, Nov 2018.

[129] Maochao Xu, Gaofeng Da, and Shouhuai Xu. Cyber epidemic models with dependences. *Internet Mathematics*, 11(1):62–92, 2015.

[130] Maochao Xu, Lei Hua, and Shouhuai Xu. A vine copula model for predicting the effectiveness of cyber defense early-warning. *Technometrics*, 0(0):1–13, 2016.

[131] Maochao Xu, Lei Hua, and Shouhuai Xu. A vine copula model for predicting the effectiveness of cyber defense early-warning. *Technometrics*, 59(4):508–520, 2017.

[132] Maochao Xu, Kristin M. Schweitzer, Raymond M. Bateman, and Shouhuai Xu. Modeling and predicting cyber hacking breaches. *IEEE Trans. Information Forensics and Security*, 13(11):2856–2871, 2018.

[133] Maochao Xu and Shouhuai Xu. An extended stochastic model for quantitative security analysis of networked systems. *Internet Mathematics*, 8(3):288–320, 2012.

[134] Shouhuai Xu. Cybersecurity dynamics. In *Proc. Symposium and Bootcamp on the Science of Security (HotSoS'14)*, pages 14:1–14:2, 2014.

[135] Shouhuai Xu. Emergent behavior in cybersecurity. In *Proceedings of the 2014 Symposium and Bootcamp on the Science of Security (HotSoS'14)*, pages 13:1–13:2, 2014.

[136] Shouhuai Xu. Cybersecurity dynamics: A foundation for the science of cybersecurity. In Zhuo Lu and Cliff Wang, editors, *Proactive and Dynamic Network Defense*. Springer New York, 2019.

[137] Shouhuai Xu, Wenlian Lu, and Hualun Li. A stochastic model of active cyber defense dynamics. *Internet Mathematics*, 11(1):23–61, 2015.

[138] Shouhuai Xu, Wenlian Lu, and Li Xu. Push- and pull-based epidemic spreading in arbitrary networks: Thresholds and deeper insights. *ACM Transactions on Autonomous and Adaptive Systems (ACM TAAS)*, 7(3):32:1–32:26, 2012.

[139] Shouhuai Xu, Wenlian Lu, Li Xu, and Zhenxin Zhan. Adaptive epidemic dynamics in networks: Thresholds and control. *ACM Transactions on Autonomous and Adaptive Systems (ACM TAAS)*, 8(4):19, 2014.

[140] Shouhuai Xu, Wenlian Lu, and Zhenxin Zhan. A stochastic model of multivirus dynamics. *IEEE Transactions on Dependable and Secure Computing*, 9(1):30–45, 2012.

[141] Tarun Yadav and Arvind Mallari Rao. Technical aspects of cyber kill chain. In *International Symposium on Security in Computing and Communication*, pages 438–452. Springer, 2015.

[142] Yanfang Ye, Shifu Hou, Lingwei Chen, Xin Li, Liang Zhao, Shouhuai Xu, Jiabin Wang, and Qi Xiong. Icsd: An automatic system for insecure code snippet detection in stack overflow over heterogeneous information network. In *Annual Computer Security Applications Conference (ACSAC'2018)*, 2018.

[143] Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. Characterizing honeypot-captured cyber attacks: Statistical framework and case study. *IEEE Transactions on Information Forensics and Security*, 8(11):1775–1789, 2013.

[144] Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. A characterization of cybersecurity posture from network telescope data. In *Proc. of the 6th International Conference on Trustworthy Systems (InTrust'14)*, pages 105–126, 2014.

[145] Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. Predicting cyber attack rates with extreme values. *IEEE Transactions on Information Forensics and Security*, 10(8):1666–1677, 2015.

[146] Ren Zheng, Wenlian Lu, and Shouhuai Xu. Active cyber defense dynamics exhibiting rich phenomena. In *Proc. 2015 Symposium on the Science of Security (HotSoS'15)*, pages 2:1–2:12, 2015.

[147] Ren Zheng, Wenlian Lu, and Shouhuai Xu. Preventive and reactive cyber defense dynamics is globally stable. *IEEE Trans. Network Science and Engineering*, 5(2):156–170, 2018.

VITA

Richard Bryan Garcia-Lebron was born in Puerto Rico. He did his undergraduate work at the University of Puerto Rico, Rio Piedras campus, at San Juan, Puerto Rico. He received his Bachelor of Science in Computer Science in 2011. After his computer science degree he continue his graduate studies at the University of Puerto Rico, Rio Piedras campus. He received his Master of Science in Mathematics in 2014. After his master degree he began his Doctor of Philosophy in Computer Science in The University of Texas at San Antonio with the purpose of combining his interests in: Computer Science, Mathematics, and Cyber Security. He received his Doctor of Philosophy in Computer Science in 2019.