A Framework for Characterizing the Evolution of Cyber Attacker-Victim Relation Graphs

Richard B. Garcia-Lebron*, Kristin M. Schweitzer†, Raymond M. Bateman† and Shouhuai Xu*
*Department of Computer Science
University of Texas at San Antonio
†U.S. Army Research Laboratory South - Cyber

***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 tackling this problem, in this paper we propose a novel, graph-theoretic abstraction, dubbed the *evolution of attacker-victim relation graphs*, for characterizing cyber attackers' reconnaissance behaviors. The framework is centered at 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.**

***Index Terms*—Cybersecurity data analytics, security modeling, graph time series, bipartite graphs**

I. 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 use 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 explore the various kinds of methods that can be adopted to characterize the evolution of such similarities. We also conduct 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 that 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.

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 are or are not 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 II presents the framework. Section III describes the case study and results. Section IV reviews related prior studies. Section V concludes the paper.

II. The Framework

Figure 1 highlights the framework, which consists of five components: data collection and preprocessing, graph-theoretic representations, lower-dimension representations (with or without using embedding), similarity-based time series representations, and temporal analysis.

A. 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., [1], [2], [3]). 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 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

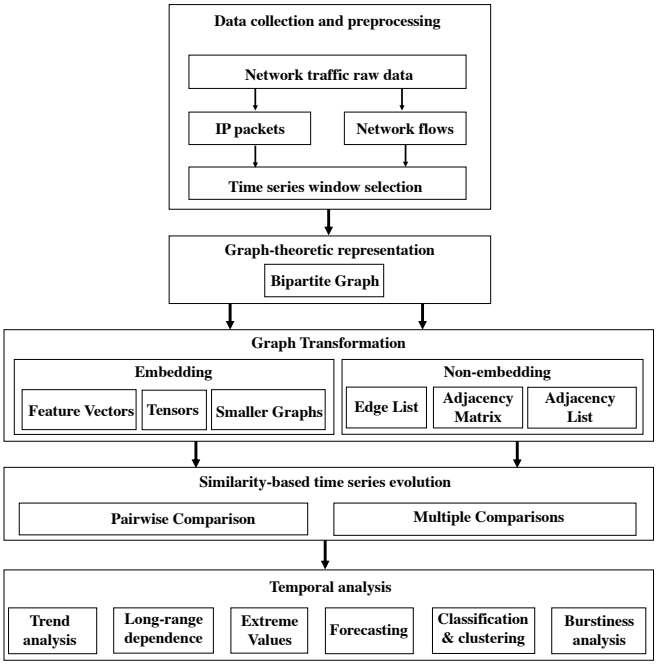


Fig. 1: The framework

when the communication between the source and 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 T belongs to time interval I_i if and only if $t_i \leq T < t_i + \Delta$. Packet-based preprocessing is similar except that there is no need to assemble packet(s) into flows.

B. Graph-Theoretic Representations

After dividing the data life-cycle into time windows of length Δ, we naturally obtain a time series of attack-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 attacks from a particular attacker to a particular victim in interval I_i). In many, but not all, cases, the attack-victim relation graphs are bipartite graphs.

C. 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 [4], [5]. 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 is to represent a graph 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.

D. Similarity-based Time Series Representations

Regardless of the specific graph-transformation method, we can define some kinds of *similarity* 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 leads to a new *time series of similarities*, which is the target for actual analysis in the next step.

E. 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.

III. Case Study and Results

A. Case Study

1) Data Collection and Preprocessing: We analyze a dataset collected at a honeypot, by transforming it to flows.

Dataset. Figure 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 are attacking some victims) and some victims are active (i.e. if are been under attack) while others no. Since the honeypot offers no legitimate Internet services, the traffic is considered malicious (see, e.g., [1], [2], [3]).

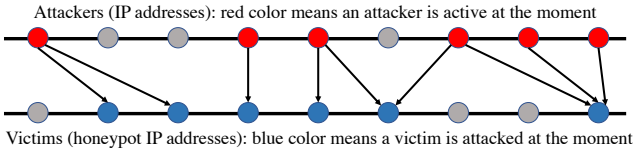


Fig. 2: A snapshot of attacks at a moment in time, in red the active attacker, in blue the active victims.

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* [6] and *Nepenthes* [7] programs. The dataset contains 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-process the PCAP data into IPFIX network flows using an idle time of 60 seconds and lifetime of 300 seconds. For this converting, we use the Yet Another Flowmeter (YAF) and the *super_mediator* tools of the Computer Emergency Response Taskforce (CERT) [8]. The dataset leads to 92,477,692 TCP flows (or attacks).

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

Table I presents the simple statistics and standard deviation (σ) of 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 observe that many flows contain only a single packets, suggesting scanning activities or initial reconnaissance efforts.

2) Graph-Theoretical Representation: Figure 2 suggests that the dataset can be naturally represented by the evolution of bipartite graphs. In order to generate a time series of bipartite graphs, we need to select the unit of time window, 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 $\Delta = 0.5, 1, 2, 9, 12, 30, 60, 90, 120, 180, 360, 720$ (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 $W_i : E_i \to \mathbb{I}^+$ 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).

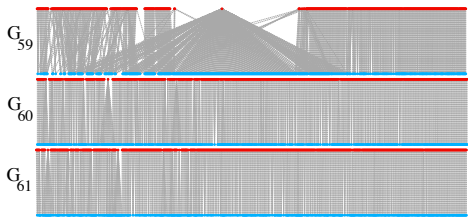


Fig. 3: Bipartite graphs G_{59}, G_{60}, and G_{61} with $\Delta = 12$.

Figure 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 SSH server for possibly launching password brute-forcing attacks.

3) Graph Transformations: In the present paper, 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 it's 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 II 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 $C_i = \{c_1, c_2, \ldots, c_m\}$, were a WCC size is define as $z_k = |c_k|$ for $1 \leq k \leq m$ and define the set $Z_i = \{z_1, z_2, \ldots z_m\}$ as the set of WCC sizes.

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 $\mathsf{stats}(D_{in})$

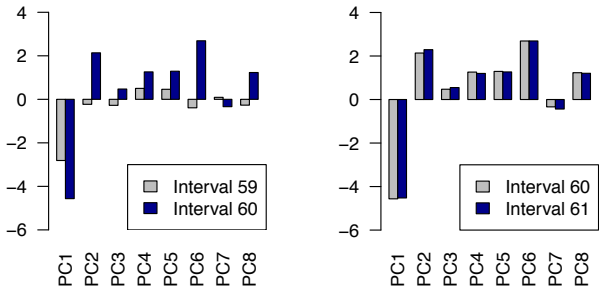
TABLE II: Features for the feature vector embedding $\mathsf{Embed}(G_i) = F_i$ representing bipartite graph G_i

We define the feature of *weighted out-degree* for attackers. This feature reflects the number of attacks 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 define the feature of *weighted in-degree* of victims in V_i, representing the number of attacks 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 [9], 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 4 plots the refined feature representation of the aforementioned G_{59}, G_{60}, and G_{61}. Figure 4a shows that G_{59} and G_{60} are very different, while Figure 4b shows that G_{60}



(a) F_{59} vs F_{60}. (b) F_{60} vs F_{61}

Fig. 4: Refined feature vectors of G_{59}, G_{60} and G_{61}.

and G_{61} are very similar. This is consistent with a visual examination of Figure 3.

4) 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:

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

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 $S(G_i, G_{i+1}) \in [0, 1]$.

5) 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 [10] if the rate of the auto-correlation function decays slowly. Formally, if

$$r(h) = Cor(X_t, X_{t+h}) \sim h^{-\beta} L(h), \quad h \to \infty \quad (2)$$

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 [11], which can be estimated using the *fArma* package in R [12],

B. Results

1) Bipartite Graph Feature Analysis: Table III 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.

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}	✓	✓	✓	✓	✓	✓	✓	✓	✓

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

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 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 1: 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.

2) Evolution Trends Analysis: Figure 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 $\Delta \in \{30, 60, 90, 120, 180, 360, 720\}$ are very similar, and (iii) the trends for $\Delta \in \{0.5, 1, 2, 9, 12\}$ are quite different from the trends for $\Delta \in \{30, 60, 90, 120, 180, 360, 720\}$.

Figure 6 presents the correlation matrix between the trends with different Δ's, and confirms that the trends within each

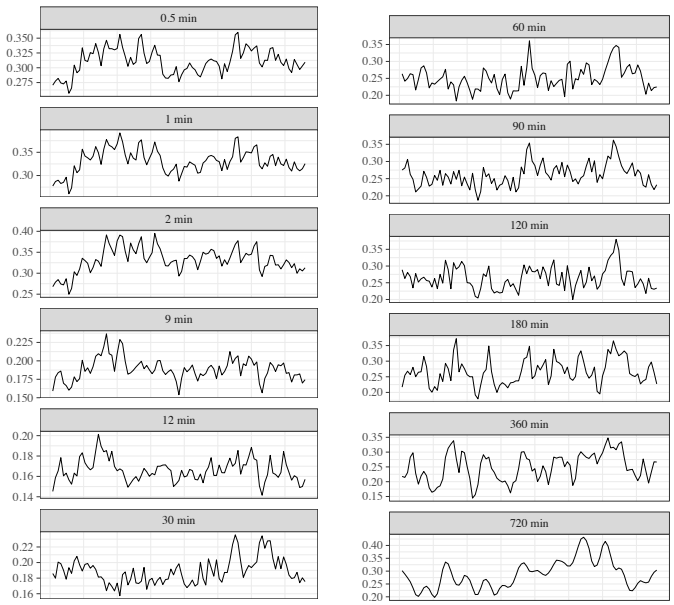


Fig. 5: Time series trend analysis with different Δ's, which are indicated on the top of each sub-figure, where the x-axis represents time and the y-axis is the trend of similarity scores $S(F_i, F_{i+1})$.

group of Δ's are highly correlated with each other, but different groups are little correlated with each other.

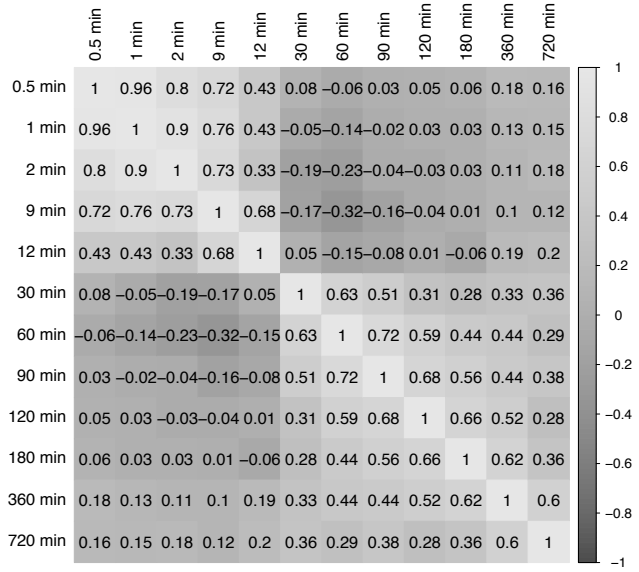


Fig. 6: Correlation matrix of daily frequency trends.

Summarizing the preceding discussion, we draw:

Insight 2: 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.

3) LRD Analysis: Table IV 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 two methods are all greater than 0.5, indicating LRD. We further use the

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 IV: The Hurst parameters with different Δ's.

Δ	$\hat{H}$	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

TABLE V: Test results for spurious LRD with different Δ's.

Smoothly Varying Trend test [13] to test whether the times series exhibit *spurious* LRD or not. Table V 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:

Insight 3: The time window size affects whether the time series exhibits LRD. Because LRD implies that a time series can be accurately predicted [1], [2], [14], [15], [3], the defender needs to be conscious in selecting Δ.

IV. 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 [16], [17], [1], [2], [14], [15], [3], [18], [19]. In contrast to previous studies on cybersecurity data analytics that focus on univariate [1], [20], [2], [14], [15], [21] or multivariate time series [18], [3], 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 [22]; characterizing attack probing activities [23]; clustering attacks [24], [25], [26], [27]; modeling attack inter-arrival times [28], [29]; predicting/forecasting attack rates [1], [2], [14], [15], [3]; detecting cyber attacks (e.g., malware, botnets) [30], [31], [32], [33], [34], [35], [36], [37].

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., [38], [39], [20], [18]), 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 [40], [41].

V. 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.

References

[1] Z. Zhan, M. Xu, and S. Xu, "Characterizing honeypot-captured cyber attacks: Statistical framework and case study," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1775–1789, 2013.

[2] Z. Zhan, M. Xu, and S. Xu, "Predicting cyber attack rates with extreme values," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1666–1677, 2015.

[3] C. Peng, M. Xu, S. Xu, and T. Hu, "Modeling multivariate cybersecurity risks," *Journal of Applied Statistics*, vol. 0, no. 0, pp. 1–23, 2018.

[4] S. Banerjee, M. Jenamani, and D. K. Pratihar, "Properties of a projected network of a bipartite network," *CoRR*, vol. abs/1707.00912, 2017.

[5] D. Koutra, N. Shah, J. T. Vogelstein, B. Gallagher, and C. Faloutsos, "Deltacon: Principled massive-graph similarity function with attribution," *ACM Trans. Knowl. Discov. Data*, vol. 10, pp. 28:1–28:43, Feb. 2016.

[6] N. Provos, "A virtual honeypot framework," in *Proc. USENIX Security Symposium*, 2004.

[7] E. Balas and C. H. Viecco, "Towards a third generation data capture architecture for honeynets," *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, pp. 21–28, 2005.

[8] C. Inacio and B. Trammell, "Yaf: Yet another flowmeter," in *LISA*, 2010.

[9] M. Kuhn, "The caret package," 2009.

[10] G. Samorodnitsky, "Long range dependence," *Found. Trends. Stoch. Sys.*, vol. 1, pp. 163–257, Jan. 2007.

[11] W. D. Ray and J. Beran, "Statistics for Long-Memory Processes.," *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, vol. 159, no. 1, p. 180, 1996.

[12] D. Wuertz, T. Setz, and Y. Chalabi, "Modelling arma time series processes: The farma package," 2017.

[13] Z. Qu, "A test against spurious long memory," *Journal of Business and Economic Statistics*, vol. 29, no. 3, pp. 423–438, 2011.

[14] Y.-Z. Chen, Z.-G. Huang, S. Xu, and Y.-C. Lai, "Spatiotemporal patterns and predictability of cyberattacks," *PLoS One*, vol. 10, p. e0124472, 05 2015.

[15] C. Peng, M. Xu, S. Xu, and T. Hu, "Modeling and predicting extreme cyber attack rates via marked point processes," *Journal of Applied Statistics*, vol. 0, no. 0, pp. 1–30, 2016.

[16] S. Xu, "Cybersecurity dynamics," in *Proc. Symposium and Bootcamp on the Science of Security (HotSoS'14)*, pp. 14:1–14:2, 2014.

[17] S. Xu, "Emergent behavior in cybersecurity," in *Proceedings of the 2014 Symposium and Bootcamp on the Science of Security (HotSoS'14)*, pp. 13:1–13:2, 2014.

[18] M. Xu, L. Hua, and S. Xu, "A vine copula model for predicting the effectiveness of cyber defense early-warning," *Technometrics*, vol. 0, no. 0, pp. 1–13, 2016.

[19] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Comput. Surv.*, vol. 49, pp. 62:1–62:35, Dec. 2016.

[20] Z. Zhan, M. Xu, and S. Xu, "A characterization of cybersecurity posture from network telescope data," in *Proc. of the 6th International Conference on Trustworthy Systems (InTrust'14)*, pp. 105–126, 2014.

[21] M. Xu, K. M. Schweitzer, R. M. Bateman, and S. Xu, "Modeling and predicting cyber hacking breaches," *IEEE Trans. Information Forensics and Security*, vol. 13, no. 11, pp. 2856–2871, 2018.

[22] A. Herrero, U. Zurutuza, and E. Corchado, "A neural-visualization ids for honeynet data," *Int. J. Neural Syst.*, vol. 22, no. 2, 2012.

[23] 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*, vol. 6, pp. 175–188, march 2011.

[24] 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*, pp. 7–13, 2009.

[25] 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*, pp. 147–154, 2008.

[26] 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*, pp. 79–87, 2007.

[27] 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*, pp. 45–54, 2004.

[28] 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*, pp. 79–91, 2006.

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

[30] 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.

[31] 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)*, pp. 39–58, 2004.

[32] 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)*, pp. 967–974, 2006.

[33] V. Pham and M. Dacier, "Honeypot trace forensics: The observation viewpoint matters," *Future Generation Comp. Syst.*, vol. 27, no. 5, pp. 539–546, 2011.

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

[35] C. Kreibich and J. Crowcroft, "Honeycomb: creating intrusion detection signatures using honeypots," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 51–56, 2004.

[36] G. Portokalidis and H. Bos, "Sweetbait: Zero-hour worm detection and containment using low- and high-interaction honeypots," *Comput. Netw.*, vol. 51, no. 5, 2007.

[37] 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.

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

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

[40] J. Z. Bakdash, S. Hutchinson, E. G. Zaroukian, L. R. Marusich, S. Thirumuruganathan, C. Sample, B. Hoffman, and G. Das, "Malware in the future? forecasting analyst detection of cyber events," *CoRR*, vol. abs/1707.03243, 2017.

[41] R. E. Harang and A. Kott, "Burstiness of intrusion detection process: Empirical evidence and a modeling approach," *IEEE Trans. Information Forensics and Security*, vol. 12, no. 10, pp. 2348–2359, 2017.