Characterizing Transition Behaviors in Internet Attack Sequences

Haitao Du and Shanchieh Jay Yang
Department of Computer Engineering
Rochester Institute of Technology, Rochester, New York 14623

***Abstract*—Cyber attacks from the Internet often span over multiple ports and multiple hosts. This work hypothesizes that there are distinct sequential patterns revealing hacking behavior. A feature called Attack Transition Action (ATA) is defined to represent the changes on attacked destinations and ports over time. The simplicity of the feature enables the development of a probabilistic model, revealing higher order transitions hidden within the attack sequences. The model trained with a real-world attack dataset uncovers several natural clusters of Internet attack behaviors. The discovered behavior patterns are explained with representative hacking strategies. Our systematic modeling and analysis provides an effective means to characterize classes of Internet attacks.**

I. Introduction

The increasing vulnerabilities and freely distributed cyber attack tools have led to significant volume of malicious activities from the Internet to penetrate enterprise networks. One of the common tasks in cyber attack analysis is effectively analyzing attacks that share similar hacking strategies. There are different criteria to perform such classification. On a high level, the category proposed by Hansman and Hunt [1] includes *viruses, worms, buffer overflows, Denial of Services (DoS) attacks, etc.*. On the other hand, the penetration may take several stages due to the increasing complexity and sophistication of network defenses [2]. In an attempt to track and predict attack penetrations, Yang *et.al.* [3] categorize attacks according to their penetration stages, including *Reconnaissance*, *Privilege Escalation*, *Intrusion*, and *Goal*.

Among the different attacks, viruses, worms and DoS attacks have attracted much attention, *e.g.,* [4] [5] [6], and various reconnaissance techniques have been well studied [7] [8]. For multi-stage attacks, simulation models have been proposed [9] and machine learning techniques have been utilized to predict future attack actions [10] [11].

While the exiting taxonomies [12] [1] [3] [13] serve well as reference guides, there is little development on systematic methods to characterize Internet attacks with multi-purposed hacking strategies. Analyzing specific vulnerabilities and packet exchanges can give detail insights on whether the hacker is performing *Host Discovery*, *Services Scanning*, *Vulnerability Attempts* or a mixture of them. However, they do not provide a large-scale understanding of the (potentially changing) strategies employed by the hackers. One key challenge of deducing hacker strategy is the complexity involved with the attack surface, typically composed of a large set of hosts and services. This complexity, at first glance, precludes the development of analytical models that are scalable to treat a significant number of observables.

This work proposes to examine Internet attack sequences defined with *Attack Transition Action (ATA)*, a feature that represents transitions in targeted host IP and port but agnostic to their specific values. Using ATA sequences originated from heterogeneous attack sources, we use Finite Order Markov Model to discover and analyze the patterns and strategies. A real-world attack dataset, University of California San Diego (UCSD) Network Telescope dataset [14], is used for analysis. The dataset is collected in November 2008 by monitoring malicious traffic towards a Class-A network, presumably representing 1/256 of global attacks [15]. Our results reveal both expected and surprising hacking behaviors that can be effectively identified based on the ATA patterns.

II. Sequential Attack Action Modeling

A. Internet Attacks

Cyber attacks from the Internet can be generally categorized into *Host Discovery*, *Services Scanning* and *Vulnerability Attempts*. We first examine how packet level observables may be associated with these behaviors.

Several techniques can be used for host discovery. The simplest way to identify live hosts is using ICMP echo request (*i.e.,* Ping). However, because it is widely misused, network administrator usually block ICMP messages to protect hosts from malicious probing. Alternatively, attackers may utilize TCP and UDP protocols. For example, one may send a TCP packet with SYN flag set on any port. No matter receiving ACK or RST packets, the attacker will know the target is live. This is typically referred to as *Half Open Scan*. Half open connections are often not recorded by applications, but could be seen by Intrusion Detection Systems, especially when the attack source is probing well known ports, such as #21 (FTP) or #80 (HTTP). Therefore, hackers may send packets to undefined ports [1]. Similar techniques also works for UDP, while the difference is that hacker would be waiting for *port unreachable* instead of RST packet. In summary, probing packets on undefined or random ports could be indicative of the behavior of host discovery.

Service scanning usually targets on few specific and well-known ports, such as #21 (FTP), #23 (Telnet), #80 (HTTP).

[1] In this paper, the terms *defined port* and *undefined port* refer to the registration of Internet Assigned Numbers Authority (IANA).

978-1-4577-0638-7 /11/$26.00 ©2011 IEEE

It can be done by sending a sequence of requests, and the feedbacks often provide the attacker with detailed information on the targeted services. There are countless service scanning techniques by constructing different types of packets for stealthy attacks. However, two essential characters exist in all scanning techniques: (1) the set of probing ports and (2) the order over which the ports are probed. Scanning tools such as NMAP [16] gives options including *fast scan* and *comprehensive scan*. Fast scan only probes few well-known ports. On the other hand, comprehensive scan exams almost every defined port to get the complete information of running services. All of them can be performed on different ports of a single host, or sweeping across different hosts. In addition, the attacker can specify the order, or perform a random scanning. In summary, probing packets on defined ports are more likely to imply service scanning.

Service scanning can provide attackers with information to identify possible vulnerabilities. No matter what the exact vulnerabilities are, from mis-configuration (*e.g.,* weak passwords) to program bugs (*e.g.,* buffer overflow), exploiting vulnerabilities would require sending repetitive malicious trails to specific services (ports). Such attack behavior is called vulnerability attempts. For example, one popular attack on HTTP server is sending various types of GET request to probe directory structure and weakly protected configuration files. Therefore, vulnerability attempts are associated with repeating packets targeted on the same host and port.

Although there are common traits in identifying these attacks, characterizing them in general is not an easy task because of the various combinations of techniques. An attacker may change the strategy based on the feedback from the target or even subjective factors, such as his/her personality. The following section proposes a feature set that is agnostic to specific IP addresses and port numbers and, thus, effectively model the attack behavior transitions.

B. Attack Transition Action (ATA)

Destination IP and destination port are two key factors to understand attacking behavior. However, it may not be desirable to define a random variable on the exact values of IP and port numbers. There are two reasons: first, there would have been a large number of possible values for the random variables[2]; second, it is not necessary to differentiate the exact destination IP and port. For example, probing the 254 IP addresses in two different subnets can be treated as the same type of behavior. Similarly, sending packets to undefined ports (*e.g.,* port 38743 vs. port 38744) implies the same behavior, regardless the exact port numbers. Therefore, our analysis focuses on whether there is a change in destination IP and port between two consecutive observables regardless the specific values. The order of the actions within the sequence is determined based on their time stamps. A random variable called *Attack Transition Action (ATA)*, $X \in \{0, 1, 2, 3\}$ is defined to denote that, comparing to the previous observable originated by the same source, the current attack action shows: 1) no change on IP and port, 2) only change in port, 3) only change in IP, and 4) changes in both IP and port.

Recalling the discussions in Section II-A, we conjecture that attack behaviors can be described using ATA defined above. For example, sending packets to the same target IP and port gives a sequence of ATA-0, and implies vulnerability attempt. A sequence of packets targeting on different ports of the same host, *i.e.,* a sequence of ATA-1, indicates single host service scanning. Sending a series of packet to different destinations but the same port indicates scanning certain service across hosts, and can be captured by a sequence of ATA-2. A sequence of ATA-3 could make sense if the destination IP are randomly chosen and ports are undefined. Such scenario may indicate host discovery.

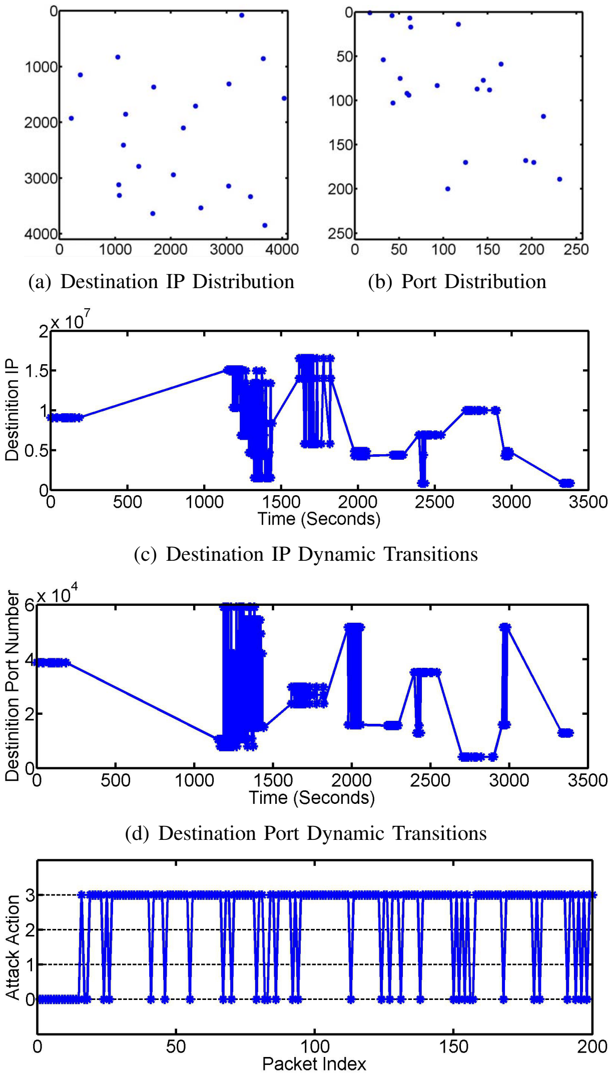
The usage of ATA is effective in representing more complex attack behavior. Consider two attacking sources in the UCSD dataset. Figures 1 and 2 represent the two attacks from three different viewpoints. Subfigures (a) and (b) show the distribution of targeted IPs and ports in a 2^{12} by 2^{12} IP space and a 2^8 by 2^8 port space, respectively. Subfigures (c) and (d) plots the targeted IPs and ports over time. Finally, Subfigure (e) shows the sequence of the attack defined by the ATA feature.

Figure 1 shows an attack originated from 77.195.130.171. Figures 1(a) and 1(b) show that it targeted randomly on various hosts and ports. Figures 1(c) and 1(d) show the dynamic transitions in the targeted IPs and ports. It can be seen that there are nine sub-attacks within the sequence. Different IPs and ports were targeted in each sub-attack and a short break is observed between the sub-attacks. Although there are nine sub-attacks, the overall attack exhibits consistent behavior, as shown in Fig.1(e). The ATA sequence shown contains the first 200 out of approximately 500 actions (1,652 seconds) for better visualization. Other than the first 20 or so ATAs, which are all 0's, the remaining ATAs are either 3's or 0's. This means that the attack begins with a single vulnerability attempt and then attempted host discovery with random IP and port choices.

Figure 2 shows an attack sequence originated from 78.152.29.21, which exhibits a very different attack behavior from that shown in Figure 1. First, Figures 2(a) and 2(b) shows that the targeted IPs and ports are not randomly chosen. Specifically, the attack targeted on four subnets (0.241.65.x, 0.242.90.x, 0.229.1.x , 0.159.253.x)[3] and did a comprehensive scan for all possible addresses on each subnet. Only two Windows file sharing services, port #139 and #445, are attacked in approximately 5,000 actions. As shown in Figure 2(c), the attack stays in one subnet for a period of time before switching to another. Figure 2(d) shows the switching between the two aforementioned ports. The attack behavior can be summarized by the ATA sequence shown in Figure 2(e). The use of ATA-2 and occasional switchings to ATA-3 suggests the cross-host

[2] IPv4 packet has 2^{32} possible values for destination IP and 2^{16} possible values for ports

[3] For anonymity reason, the first byte of IP address for Network Telescope is masked with 0.



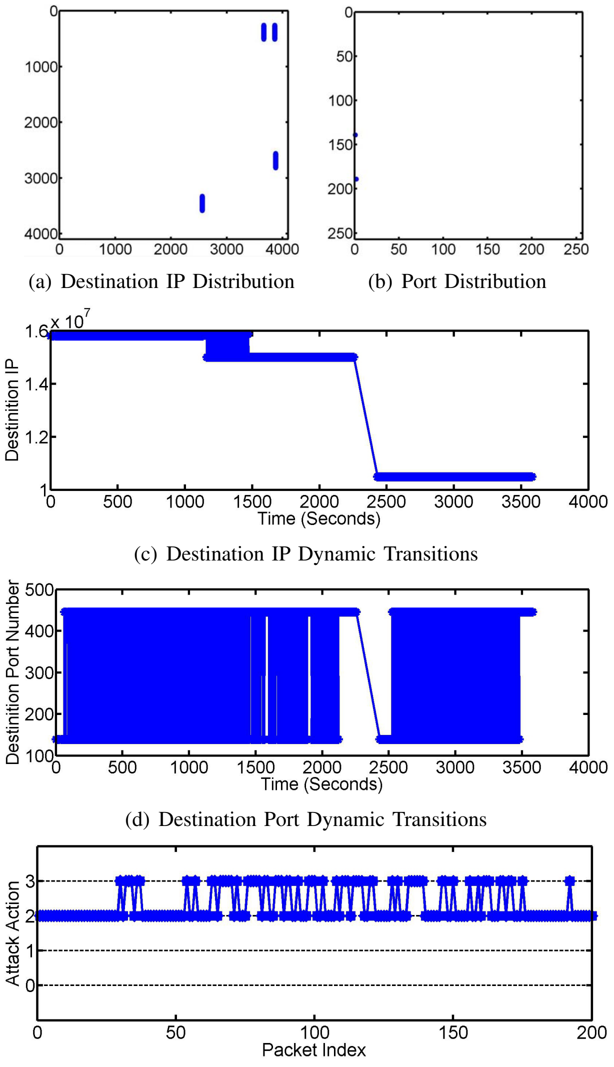
(a) Destination IP Distribution (b) Port Distribution

(c) Destination IP Dynamic Transitions

(d) Destination Port Dynamic Transitions

(e) ATA Sequence

Fig. 1. Attack Visualization for 77.195.130.171



(a) Destination IP Distribution (b) Port Distribution

(c) Destination IP Dynamic Transitions

(d) Destination Port Dynamic Transitions

(e) ATA Sequence

Fig. 2. Attack Visualization for 78.152.29.21

vulnerability scanning attack.

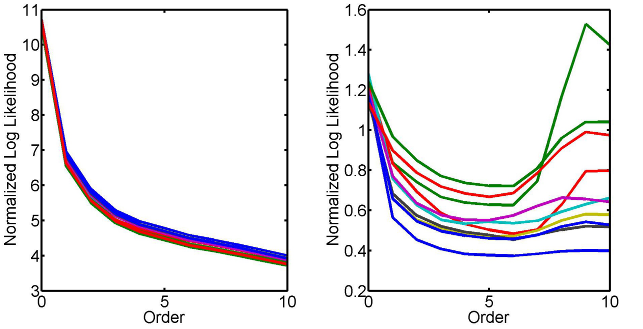
The two examples shown above illustrate the effectiveness of using ATA sequences to differentiate attack behaviors. Because of the simplicity of the feature set, a formal model is developed to systematically capture attack patterns in ATA sequences. The model can then be used to effectively discover and characterize different attack behaviors.

C. ATA Sequence Model Selection

Suppose $(X_1, X_2, X_3, \cdots, X_N)$ is an ATA sequence of length N, where X_i denotes the i^{th} action. The ideal model for such a sequence is the joint distribution $P(X_1, X_2, \cdots, X_N)$, which gives a full description of all possible events and any marginal or conditional probabilities. However, the length of an attack can vary and be very large - the attack sequences from the UCSD dataset can be as long as 10^6. The number of parameters for such model would grow exponentially with respect to N. *Finite Order Markov Model* has been used in many settings to reduce the model complexity by applying conditional independence assumption to past events that are less or not relevant to the current. For o^{th} order model, action X_i is conditionally dependent on the previous o random variables. The assumption may be reasonable because Internet attacks are results of an interactive process, where even scripted attacks require interrupts to determine the next set of actions.

Identifying the order of a model is a general problem of model selection. The principle of *Bias-Variance Tradeoff* [17] suggests how to search for the model with optimal complexity. Fitting a model that is too simple, would lead to a high *Bias*. The capability of the model is limited to describe the data. In this case, no matter how to optimize the parameters (*i.e.,* the probability distributions) within the model, the performance would not be desirable. On the other hand, fitting a model that is too complex would lead to a high *Variance*, which is *overfitting* on the training data. In this case, the model preforms perfectly on training data but fails to generalize on unseen data. This work uses *K-ford Cross Validation* to reveal the optimal order o for the Finite Order Markov Model.

For *K-fold Cross Validation*, the dataset is divided into K chunks and perform training and testing K times. In each time, one chunk of data is used for testing and the others for training. The averaged performance on test error is used to select the optimal model. Once the model complexity is determined, all data is used to train and obtain the final model.



(a) Training Error

(b) Testing Error

Fig. 3. Cross Validation Curve for Model Selection

The first hour (*i.e.,* 1/48) of the UCSD Network Telescope 2-day dataset is used to select the optimal model using 10-fold Cross Validation. Figure 3 shows the training and testing errors of the ten repetitions, one line for each. The x-axis denotes the model order and the y-axis denotes the value of the loss function, for which negative likelihood is chosen to fit a joint distribution with unsupervised learning. As expected, the training error (Fig. 3(a)) goes down as o increases, and the testing error (Fig. 3(b)) exhibits a ”U” shape pattern and suggests that the optimal $o = 6$. As a result, sixth-order model tells us that the joint distribution of length N sequence can be expressed as a product of conditional probabilities shown in Eq. (1).

$$P(X_1, X_2, \cdots, X_N) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2) \cdots P(X_6|X_1, \cdots, X_5)\prod_{i=7}^{N} P(X_i|X_{i-1}, \cdots, X_{i-6}) \qquad (1)$$

III. Attack Transition Action Sequences Analysis

A. ATA Sequence Model

The resulting sixth-order Markov Model can be used to analyze Internet attacks. First, we focus on analyzing ATA patterns of length 7 since Cross Validation suggests that an ATA best depends on the previous six ATAs. Let $(x_1\ x_2 \cdots x_7)$, $x_i \in \{0, 1, 2, 3\}$ denotes a sequence of seven ATA events, and refer each combination as an *ATA Pattern*. There are $4^7 = 16,384$ possible ATA patterns, but only $5,021$ of them occurred in the dataset. Figure 4 shows the probability of occurrence (in log scale) for the 5,021 patterns, from the smallest probability to the largest. More than half of the patterns have a probability less than 10^{-5} and about 150 patterns have a probability over 10^{-3}. The top 3 occurring patterns are (0000000) with the probability of 0.1700, (2222222) with 0.1693, and (3333333) with 0.1125. These three ATA patterns accounted for about 45% of the data.

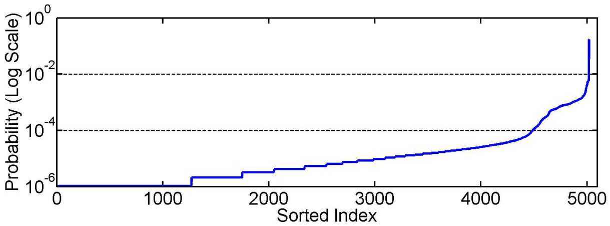


Fig. 4. Probability of ATA Patterns

The non-uniform distribution of patterns can be explained well from cyber security perspective. First, not all ATA patterns are reasonable hacking behaviors. Consider the pattern (1112111) as an example. The ATA-1's in the beginning suggest services scanning across different ports. Having ATA-2 following a series of ATA-1 is not a reasonable (at least not a common) hacking behavior, since it means that that the hacker keeps the same port when switching hosts for service scanning. In fact, it is more common with scripting attacks where the same order of scans happen on a different host, which will lead to an ATA pattern such as (1113111).

B. ATA Pattern Sets

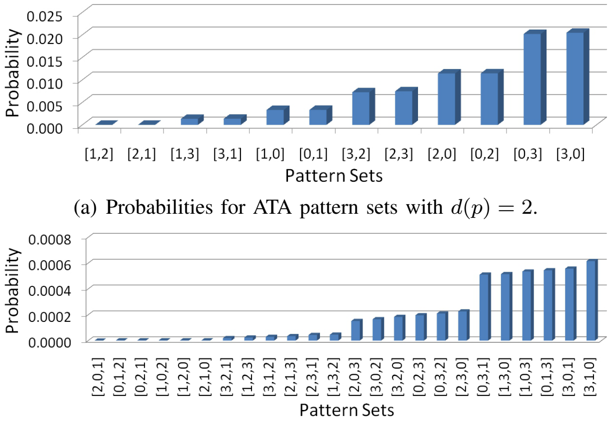
As discussed earlier, the dominant attacks are (0000000), (3333333) and (2222222), which indicate vulnerability attempts, host discovery, and services scanning, respectively. We now turn our attention to more complex patterns. Let $d(p)$ denote the number of unique values in an ATA pattern p. The three dominant patterns has $d(p) = 1$. We omit the discussion for patterns with $d(p) = 4$, as they are special cases (0.07% of the dataset), which require more in-depth analysis on packet level details. This paper focuses on performing systematic analysis for $d(p) = 2$ and $d(p) = 3$.

Notice that in the 7-digit ATA patterns, the specific times an ATA repeats itself may not be very important. For example, the numbers of ATA-0 and ATA-3 in (0033333) versus (0003333) do not add values in differentiating between the two. Therefore, we define *pattern sets* in the subsequent analysis, where a pattern set $[x, y]$ has $d(p) = 2$ and represents all 7-digit ATA patterns that contain at least one ATA-x followed by at least one ATA-y, and likewise for $d(p) = 3$ pattern sets $[x, y, z]$. There are a total of 12 pattern sets with $d(p) = 2$. With this definition, the probability of having a pattern set $[x, y]$ can be written as

$$\begin{aligned} P([x, y]) =& P(x, y, \cdots, y) + P(x, x, y, \cdots, y) \\ &+ \cdots + P(x, \cdots x, y) \end{aligned}$$

Similarly, there are 24 pattern sets with $d(p) = 3$. The probability of seeing a pattern set $[x, y, z]$ is a sum of the probabilities of occurrence for all 15 patterns in the set.

Figure 5 gives the probability distribution for the pattern sets with $d(p) = 2$ and $d(p) = 3$. Interestingly, there are natural grouping in both cases. For the case of $d(p) = 2$, the most frequent group includes pattern sets $[0, 3]$ and $[3, 0]$, the next group contains $[0, 2]$ and $[2, 0]$, and so on. Each group contains same ATAs but in different permutations. Similarly, for the case of $d(p) = 3$, the most frequent pattern set contains all the permutations of ATA-0, ATA-1, and ATA-3, the second group contains permutations of ATA-0, ATA-2, and ATA-3, and so on.



(a) Probabilities for ATA pattern sets with $d(p) = 2$.

(b) Probabilities for ATA pattern sets with $d(p) = 3$.

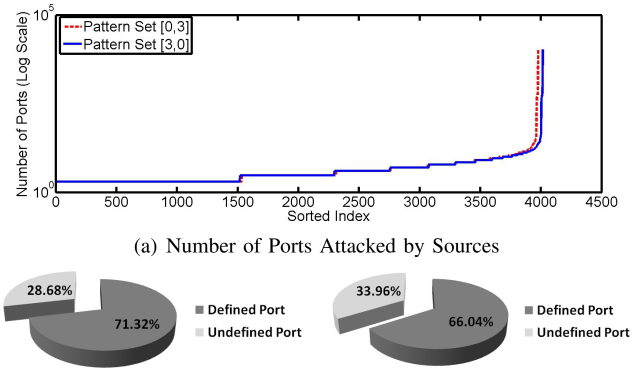
Fig. 5. Probabilities for Different Pattern Sets

The statistics tells us that, host discovery and vulnerability attempts (*i.e.,* combination of ATA-0 and ATA-3) are more frequent than scanning across the hosts (*i.e.,* ATA-2), and services scanning across ports is the least one (*i.e.,* ATA-1). Such information is intuitive on the passive dataset [15]: because the attack sources receive no feedback from the targets, the hackers would likely attack a large number of hosts instead of comprehensive scanning on one target.

C. Orders of ATA in Pattern Set

An interesting observation from Figure 5 is the natural grouping, where the order of ATAs in the pattern sets does not seem to make much difference in their likeliness to occur. This raises a question on whether the attack sources in these different pattern sets actual behave similarly. We examine several metrics for the sources in each group, and the results suggest a good match in each group. In particular, we consider the number of distinct ports attacked by the sources and whether the attacked ports are defined or undefined ports. We choose these two metrics since they suggest how diverse the attacks are and whether they focus on specific service scanning or random discovery. Two examples are given below.

Consider the sources from the pattern sets $[0, 3]$ and and those from $[3, 0]$. Figure 6(a) plots the number of ports (in log scale) attacked by each source in the two sets. The sources are sorted based on the number of ports they attacked, from lowest to highest. The overlaping lines suggest the overall behavior in $[0, 3]$ closely matches to that in $[3, 0]$. For both cases, approximately $1,500$ out of $4,000$ sources attack only 1 port. Figures 6(b) and 6(c) give the distribution of defined versus undefined ports for the two cases. The resemblance between the two pie charts again suggests the similar behaviors between the two.



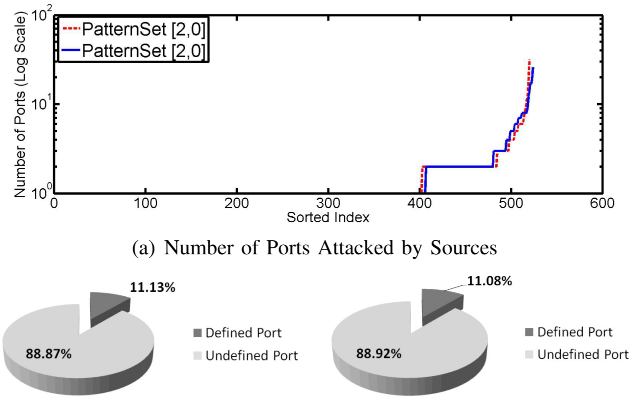
(a) Number of Ports Attacked by Sources

(b) Port Distribution in Set [0,3]

(c) Port Distribution in Set [3,0]

Fig. 6. Source Behavior from Pattern Sets $[0, 3]$ and $[3, 0]$

The same analysis is performed on the group containing pattern set $[0, 2]$ and $[2, 0]$, and the results are shown in Figure 7. There is again clear resemblance between the source behaviors from the two pattern sets. In addition, the behaviors shown in Figures 6 and 7 are clearly different. That is, sources from the first group behave very differently as compare to those in the second group. In particular, the sources in pattern sets $[0, 3]$ and $[3, 0]$ are more likely to target on undefined port (about 70%), which suggests for host discovery. On the contrary, the sources in pattern sets $[0, 2]$ and $[2, 0]$ mainly target on defined port (about 88%), suggesting specific services were targeted.



(a) Number of Ports Attacked by Sources

(b) Port Distribution in Set [0,2]

(c) Port Distribution in Set [2,0]

Fig. 7. Source Behavior on Pattern Set $[0, 2]$ and $[2, 0]$

Similar analyses were performed on the other groups and the results all suggested matching source behavior within each group. This finding prescribes that the order over which an ATA is in a pattern set is a non-factor in differentiating attack behavior. Given this property, we further define a new notation $\{x, y\}$ to represent both the pattern sets $[x, y]$ and $[y, x]$. Similarly, $\{x, y, z\}$ represent all pattern sets containing ATA-x, ATA-y, and ATA-z. Using this notation, we summarize in

Table I how the ATA patterns can be use to identify the Internet attack behaviors. Note that a single pattern may suggest for multiple behaviors as there are multi-purpose attacks.

TABLE I
INTERNET ATTACKS AND CORRESPONDING ATA PATTERNS

Internet Attacks	ATA Patterns
Host Discovery	{3}, {0,3},{1,3}, {0,1,3}
Services Scanning	{1}, {2}, {0,1}, {0,2}, {1,2} {0,1,2}, {1,2,3},{0,2,3}
Vulnerability Attempts	{0}, {0,3}, {1,3}

D. ATA Inference

In addition to its simplicity, our analysis has suggested three key properties associated with ATA sequences. First, the cross-validation finds that the optimal finite order for the probabilistic model, which is six for the dataset considered in this paper. Second and third, neither the exact number of times an ATA repeats nor the order over which an ATA occurs in an ATA pattern is significant for differentiating attacks . Because of these properties, the probabilistic models can be used to answer questions such as the following. If a hacker does at least one ATA-0 and one ATA-1, how likely will ATA-3 also show up? That is

$$P(\{z\}|\{x,y,-\}) \;=\; \frac{P(\{z,-\})}{P(\{x,y,-\})}$$

where $\{x,y,-\}$ denotes the ATA patterns contain at least one ATA-x (and one ATA-y) with or without any other ATAs. Table II gives the results of these conditional probabilities.

TABLE II
CONDITIONAL PROBABILITIES FOR ATA SEQUENCES

Conditional Prob.	Value	Conditional Prob.	Value
$P(\{0\}\|\{1,2,-\})$	0.7780	$P(\{2\}\|\{0,1,-\})$	0.0684
$P(\{0\}\|\{1,3,-\})$	0.5808	$P(\{2\}\|\{0,3,-\})$	0.0403
$P(\{0\}\|\{2,3,-\})$	0.1104	$P(\{2\}\|\{1,3,-\})$	0.1325
$P(\{1\}\|\{0,2,-\})$	0.0271	$P(\{3\}\|\{0,1,-\})$	0.3752
$P(\{1\}\|\{0,3,-\})$	0.0872	$P(\{3\}\|\{0,2,-\})$	0.0747
$P(\{1\}\|\{2,3,-\})$	0.0544	$P(\{3\}\|\{1,2,-\})$	0.9740

Several interesting observations can be made from the results shown in Table II. Recall how the ATA pattern sets match to attack behaviors in Table I. The high value for $P(\{3\}|\{1,2,-\}) = 0.974$ suggests that a service scanning attack almost always goes with host discovery, but not vice versa (since $P(\{2\}|\{1,3,-\})$ and $P(\{1\}|\{2,3,-\})$ are very low). On the other hand, $P(\{0\}|\{1,2,-\}) = 0.778$ tells us that if the scanning is across both IPs and ports, the hacker is more likely to also perform vulnerability attempts. Finally, host discovery attacks have a good chance to try the same vulnerability after discovering the target.

IV. CONCLUSION

Going beyond examining detail packet contents over time, this work proposes to use ATA sequences to characterize Internet attack behavior based on how hackers choose the target IPs and ports over time. Because of its simplicity, ATA sequences can be trained and form Finite Order Markov Models for analytical analysis. Optimal model complexity was found via Cross Validation. Analyzing the model identifies intriguing properties of Internet Attacks, and allows differentiating between host discovery, service scanning and vulnerability attempts. In addition, our analysis with the attack dataset has suggested that ATA pattern sets have a natural grouping where the order over which the ATAs occur in a pattern is not significant for differentiating attack behaviors. The proposed framework provides a systematic and scalable solution to characterize Internet attack strategies that can be multi-purposed and changing over time.

ACKNOWLEDGMENT

The authors would like to thank Prof. Justin Domke for valuable discussions on sequence probabilistic modeling.

REFERENCES

[1] S. Hansman and R. Hunt, “A taxonomy of network and computer attacks,” *Computers & Security*, vol. 24, no. 1, pp. 31–43, 2005.

[2] A. Fuchsberger, “Intrusion detection systems and intrusion prevention systems,” *Information Security Technical Report*, vol. 10, no. 3, pp. 134–139, 2005.

[3] S. J. Yang, A. Stotz *et al.*, “High level information fusion for tracking and projection of multistage cyber attacks,” *Information Fusion*, vol. 10, no. 1, pp. 107–121, 2009.

[4] D. Moore, V. Paxson, S. others, and N. Weaver, “Inside the slammer worm,” *IEEE Security & Privacy*, vol. 1, no. 4, pp. 33–39, 2003.

[5] D. Moore, C. Shannon *et al.*, “Inferring internet denial-of-service activity,” *ACM Transactions on Computer Systems (TOCS)*, vol. 24, no. 2, pp. 115–139, 2006.

[6] H. Wang, D. Zhang, and K. G. Shin, “Detecting syn flooding attacks,” in *Proceedings of IEEE INFOCOM’02*, vol. 3, June 2002, pp. 1530–1539.

[7] M. Allman *et al.*, “A Brief History of Scanning,” in *Proceedings of ACM SIGCOMM’07*, 2007, p. 82.

[8] Yegneswaran *et al.*, “Internet intrusions: Global characteristics and prevalence,” in *Proceedings of the international conference on Measurement and modeling of computer systems*, 2003, p. 147.

[9] M. Kuhl, J. Kistner *et al.*, “Cyber attack modeling and simulation for network security analysis,” in *Proceedings of Winter Simulation Conference*, Dec. 2007, pp. 1180–1188.

[10] D. Fava, S. Byers, and S. Yang, “Projecting cyberattacks through variable-length markov models,” *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 3, pp. 359–369, Sept. 2008.

[11] H. Du, D. Liu, and ohters, “Toward ensemble characterization and projection of multistage cyber attacks,” in *Proceedings of IEEE ICCCN’10*, 2010, pp. 1–8.

[12] F. Cohen, “Simulating cyber attacks, defences, and consequences,” *Computers & Security*, vol. 18, no. 6, pp. 479–518, 1999.

[13] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[14] E. Aben *et al.*, “The CAIDA UCSD Network Telescope Two Days in November 2008 Dataset, http://www.caida.org/data/passive/telescope-2days-2008_dataset.xml.”

[15] D. Moore, C. Shannon, G. Voelker, and S. Savage, “Network telescopes: Technical report,” *CAIDA*, 2004.

[16] G. Lyon, *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure, USA, 2009.

[17] T. Hastie, R. Tibshirani *et al.*, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2001.