2010 24th IEEE International Conference on Advanced Information Networking and Applications

Empirical Analysis of Attackers Activity on Multi-Tier Web Systems

Katerina Goseva-Popstojanova, Brandon Miller, Risto Pantev, and Ana Dimitrijevikj
Lane Department of Computer Science and Electrical Engineering
West Virginia University, Morgantown, WV, 26506-6109, USA
E-mails: Katerina.Goseva@mail.wvu.edu, {bmille24, rpantev, adimitri}@mix.wvu.edu

***Abstract*—Web-based systems commonly face unique set of vulnerabilities and security threats due to their high exposure, access by browsers, and integration with databases. In this paper we present empirical analysis of attackers activities based on data collected by two high-interaction honeypots. The contributions of our work include: (1) Classification of the malicious traffic to port scans, vulnerability scans, and attacks; (2) Conducting experiments which, in addition to attackers activities aimed at individual components, allowed us to observe and study vulnerability scans and attacks that span multiple system components; and (3) Statistical characterization of the malicious traffic.**

***Keywords*-port and vulnerability scans; attacks; Web-based systems; empirical analysis of malicious traffic; distribution fitting**

I. Introduction

Many business and everyday activities are now built as Web based applications. These applications commonly face a unique set of vulnerabilities due to the access by browsers, high exposure, and their integration with databases. SANS Institute Annual update of the top 20 security risks (http://www.sans.org/top20/) stated that almost half of the vulnerabilities discovered in 2007 were Web application vulnerabilities. Even more, Web application vulnerabilities were listed as the top server-side vulnerabilities, with the number of attempted attacks for some of the large Web hosting farms ranging from hundreds of thousands to even millions every day. Computer Security Institute reported that 92% of respondents to a survey experienced more than ten Web site incidents [7].

Finding attack attempts in a huge amount of monitored data from a Web server under regular use is a ‘needle in a haystack’ problem. Therefore, we decided to develop and deploy several honeypots that appear to be legitimate servers, but are actually collecting information on attackers’ activity. In case of some honeypots the goal is to allow adversaries to easily penetrate the system, so researchers can study attackers’ behaviors after successful exploitation [2], [11]. Our goal is different – we aim at studying the patterns and characteristics of attackers activity on typical Web based systems. Therefore, our experimental setup has the following unique features:

- We deployed high-interaction honeypots with standard off-the-shelf operating system and applications that follow typical security guidelines and do not include user accounts with nil or weak passwords.
- We built two identical honeypots. One of them was advertised and thus allowed for attacks based on search engines. The IP address of the second honeypot, which served as a control in our analysis, was not advertised anywhere on the Web. Surprisingly, honeypots from related work that included Web servers, with exception of [11], were not advertised.
- Instead of a set of independent applications typical for the honeypots in the related work, our honeypots have meaningful functionality and follow a three-tier architecture consisting of Web server, application server, and a database. In addition to capturing the network traffic as in related work, our data collection process also included application level logging which appeared to be very useful and allowed for more efficient, in-depth analysis of attackers activity.
- Web-based systems running on our honeypots allowed direct attacks to each component, as well as attacks on one component through the others. For example, a database server may be attacked directly on its port, or through a more complex attack by first accessing the Web and application servers. This aspect of attackers’ activities has not been addressed in the related work.

The main contributions of our work with respect to the empirical analysis are as follows:

- Part of our analysis consists of *descriptive statistics* aimed at classifying attackers’ activity to part scans, vulnerability scans, and attacks on different components of the Web based system. In this context, a *port scan* is used to check for open or closed ports and for used or unused services. A *vulnerability scan* is used to explore the presence of a vulnerability. Finally, an *attack* is defined as an exploit of vulnerabilities by a human or a program. In addition to vulnerability scans and attacks to individual components, we observed and studied vulnerability scans and attacks that span multiple system components.
- We carried out *formal statistical analysis* of the attackers activities, including the number of TCP connections and packets originated from unique IP sources, and characteristics of malicious TCP connections (i.e., num-

1550-445X/10 $26.00 © 2010 IEEE
DOI 10.1109/AINA.2010.138

ber of packets and bytes transferred per connection and connection duration). Unlike statistical characterization of (nonmalicious) network traffic which has a long tradition (see for example [8], [9] and references therein), it appears that there were only very few attempts to statistically model some aspects of malicious traffic, such as the distribution of the time between visits of reappearing IPs in [4], or time between two consecutive attacks at a given destination IP [10]. The statistical analysis of the malicious traffic presented in this paper is a step towards filling this gap.

The paper is organized as follows. Section II presents the related work, followed by the description of the experimental set-up given in section III. In-depth analysis of the malicious TCP traffic is presented in section IV, while the statistical characterization is given in section V. Section VI concludes the paper.

II. Related work

During the last decade several initiatives have been developed to monitor and collect real world data about malicious activities on the Internet, including deploying honeypots. One example is the data collection environment Leurre.com (http://www.leurrecom.org/) which is based on low-interaction honeypots that emulate particular operating systems and services. Analysis of frequently targeted ports, port sequences, and attack origins, based on data collected by multiple low-interaction honeypots was presented in [5], [14]. The analysis presented in [10], based on data collected from 14 low interaction honeypots, included using linear regression to model the number of attacks per unit of time as a function of attacks originating from a single country, and fitting a mixture of exponential and Pareto distributions to model the time between two consecutive attacks. Low-interaction honeypots, however, can be easily fingerprinted by the attackers. Another limitation is that attackers can only perform limited activities, without being able to scan for vulnerabilities or succeed in compromising the server.

In order to provide more realistic experience to the attackers and gather more information about attacks, high-interaction honeypots supported by the Honeynet Project (http://www.honeynet.org/) utilize actual operating systems and applications. The work presented in [12] was based on one high-interaction honeypot and two low-interaction honeypots. The analysis consisted of distribution of attacks across different ports, attacks origins, and description of two instances of successful attacks. Similar analysis based on three high interaction honeypots, each running different operating system, was presented in [6]. [13] explored whether port scans are precursors to attacks based on network traffic data collected from two high-interaction honeypots. The analysis considered only the number of packets per connection without identifying the specific types of scans and attacks. The goal of the work presented in [2] was to analyze the behavior of the attackers who succeeded in breaking into a high-interaction honeypot which had weak passwords for multiple SSH user accounts.

A recent work presented in [4] compared the data collected by Leurre.com and two high-interaction honeypots which ran several unrelated applications. The analysis was again based only on the network traffic data and included most often scanned ports, number of attacking hosts, persistence of attackers, and the distribution of the time between the first packet exchanges from reappearing IPs. Another recent paper [3], again using only network traffic, compared the events that targeted similar ports on the same day across data collected by two high-interaction honeypots and data from two global repositories. It is also worth mentioning a recent study based on analysis of firewall logs from over 1600 different networks world wide [15]. This work included analysis of dominant ports visited by attackers, identification of the worst offenders, and analysis of the worm related traffic.

III. Experimental setup

In this work we use the experimental setup shown in Figure 1 which follows the principles of the generation II high-interaction honeypots developed as a part of the Honeynet project. An integral part of a honeypot system is the honeywall which acts as a bridging firewall between the honeypot and the Internet. Any traffic going to or from the honeypots passes through the honeywall, which logs all of the packets using TCPDump and then silently forwards the traffic without modifying the hop count of the packets. The honeywall limits the outbound connections an attacker can initiate from a honeypot to 20 packets per day, which reduces the risk of malicious activities originated from a compromised honeypot. The captured network traffic is stored in a central data repository which ran on a separate physical host. We also collected information related to the system activity and various applications running on our honeypots.

We built two identical honeypots. Each honeypot had its own IP address and a hostname and ran on a VMWare virtual machine with a default installation of Ubuntu 7.04. One of the honeypots was *advertised* using a technique called 'transparent linking' which involves placing a hyperlink pointing to our honeypot on a regular, public Web page, so that the advertised honeypot is indexed by search engines

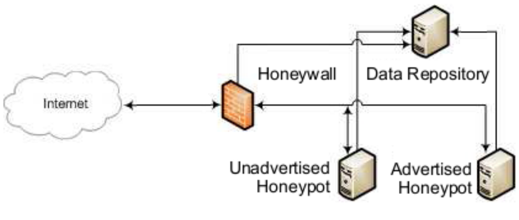

Figure 1. Experimental Setup

and Web crawlers, but cannot be accessed directly by humans. This way we allowed for attacks based on search engines (using the so called *search-based strategy* [11]) and thus have a more realistic setup. The second honeypot was not advertised anywhere on the Web. This *unadvertised* honeypot could only be reached by *IP-based strategy* when an attacker scans or attacks an IP address without (previous) involvement of search engines [11]. In our setup the unadvertised honeypot serves as a control and allows us to determine the relative contribution of search-based strategies (which only work on the advertised honeypot) to IP-based strategies (which work on both honeypts).

Instead of a collection of independent services typical for related work, each of our honeypots ran a Web based system with a three-tier architecture (i.e., Web server, an application server, and database). The particular configuration consisted of an Apache2 (version 2.2.3-3) used as a Web server. PHP5 (version 5.2.1) serves the phpMyAdmin application (version 2.9.1.1), which is the front-end of the MySQL database (version 5.0.38-0). In addition, OpenSSH server and client (version 4.3p2-8) were installed to allow for remote login, as it is typical for many Web systems. The SSH user account was set up with a strong password. The MySQL server allowed for a user login via phpMyAdmin interface. No user accounts in the MySQL server, however, were directly accessible by remote systems. The software packages installed on the honeypots are typical installations of somewhat older versions, each with a number of known vulnerabilities. Such configurations provided plenty of opportunities for compromising the honeypots, while still running applications new enough to be found on Internet.

In addition to the network traffic typically collected by honeypots, our setup included operating system and application level logging, which appeared to be very useful and often allowed for more efficient analysis of attackers' activity. Specifically, Apache, SSH, and MySQL logs were integrated into the syslogs. We used custom developed scripts to parse the network traffic capture file and application level logs.

For both the advertised and unadvertised honeypot we first removed the legitimate nonmalicious traffic which consisted of the system management traffic and legitimate Web crawlers such as Google and MSNbot. The crawlers were removed based on the IP addresses listed in iplists.com, a Web site which publishes lists of crawler's IP addresses and other similar sites and based on manual inspection of the reminding traffic. As expected, the unadvertised Web server did not receive traffic from legitimate crawlers. We decided to analyze only the incoming traffic because the outgoing traffic consisted only of responses to requests sent to the honeypots. It should be emphasized that neither of our honeypots was exploited successfully in the four months duration of the experiment.

IV. What did attackers do?

Our honeypots ran during the period of almost four months (June 2 to September 28, 2008). As expected the traffic was dominated by the TCP component. Thus, 91.25% of unique IP sources and 99.95% of the packets on the advertised honeypot, that is, 87.41% of unique IP sources and 99.95% of the packets on the unadvertised honeypot were due to TCP traffic. With respect to TCP which is connection oriented protocol, following the definition used in the area of network traffic analysis [9], we define a *connection* as a unique tuple $\{$source IP address, source port, destination IP address, destination port$\}$ with a maximum inter-arrival time between packets of 64 seconds. Advertised and unadvertised honeypots had 41,359 and 52,017 connections, respectively.

We start with discussing the distribution of the malicious TCP traffic across different ports based on the results shown in Table I.

- SSH (port 22) and MySQL (port 3306) traffic dominate the malicious TCP traffic on each honeypot, contributing over 99% of the total number of packets.
- HTTP (port 80) was the third most popular port, with significantly more traffic on the advertised than on the unadvertised honeypot, which shows that search-based strategies dominate the malicious visits on port 80.
- In addition to the six ports given in Table I, only three other ports FTP (21), HTTP ALT (8000) and SSL (443) were targeted more than once.

Next, we provide a detailed analysis of TCP traffic across HTTP, MySQL, and SSH protocols. A unique characteristic of our analysis is that we distinguish between *port scans*, *vulnerability scans*, and *attacks*. For this purpose, we first identify the port scans to ports 80, 3306, and 22 by extracting from the pcap files the TCP connections to each of these ports that did not end up in the corresponding application log. Then, we analyze the application logs to identify vulnerability scans and attacks.

A. HTTP traffic

Only 4.59% of connections to port 80 on the advertised honeypot and 23.07% of the connections to port 80 on the unadvertised honeypot were port scans to port 80. Total of 18 unique IP addresses on advertised and 15 on unadvertised honeypot scanned port 80, out of which 14 scanned both honeypots. It should be noted that four attackers on advertised honeypt and three on unadvertised honeypot (with two being common) first port scanned 80 before attacking Apache, PHP, or MySQL servers.

Based on the data extracted from Apache logs, in addition to the request level traffic, we analyzed the session level traffic, where a *session* is defined as a sequence of requests from the same source IP address to port 80, with a time between two successive request not exceeding 30 minutes [8]. For both vulnerability scans and attacks coming through

	Advertised honeypot				Unadvertised honeypot			
Port	Connections		Packets		Connections		Packets	
SSH (22)	16,908	40.88%	203,569	64.59%	28,777	55.32%	346,164	77.91%
MySQL (3306)	23,649	57.18%	100,765	31.97%	22,874	43.97%	97,163	21.87%
HTTP (80)	522	1.26%	10,301	3.27%	78	0.15%	463	0.10%
SMTP (25)	53	0.13%	53	0.02%	58	0.11%	58	0.01%
MS SQL (1433)	35	0.08%	74	0.02%	37	0.07%	76	0.02%
HTTP ALT (8080)	25	0.06%	54	0.02%	26	0.05%	52	0.01%
Other	167	0.40%	346	0.11%	166	0.32%	352	0.08%
Total	**41,359**	**100.00%**	**315,162**	**100.00%**	**52,017**	**100.00%**	**444,328**	**100.00%**

Table I
BASIC STATISTICS ABOUT TCP PORTS VISITED ON EACH HONEYPOT

the front-end Apache server, we specifically distinguished between those ending up at Apache and those spanning multiple components of the Web-based system, which is unique to our study. The following observations can be made from the results presented in Table II.

- The advertised honeypot received significantly more HTTP requests and sessions than the unadvertised honeypot.
- The number of sessions and requests due to vulnerability scans were significantly higher than the number of sessions and requests due to attacks on both honeypots.
- Unlike the advertised honeypot, vulnerability scans to multiple components, password cracking and e-mail harvesting attacks did not reach the unadvertised honeypot.

Specifically, *vulnerability scans* were distributed among the following categories:

- *DFind* is a vulenrability scanning tool utility which allows an attacker to probe whether a host is vulnerable to specific exploits or is running certain services.
- *OPTIONS* is an HTTP method which allows the client to determine the options and/or requirements associated with a resource, or the capabilities of a server.
- *CONNECT* is an HTTP method which in our case was used by an attacker to establish a connection to another server using our Web server as a proxy.
- *Fingerprinting* category subsumes all fingerprinting done to different components in our setup. *Apache server* and *phpMyAdmin* were fingerprinted by sending GET requests. Each server returned information about the corresponding installation. *Multiple components* within individual session were fingerprinted only on the advertised server, in two different ways. In 6.54% of the malicious sessions, similarly to fingerprinting of Apache and phpMyAdmin, the agent field was either missing or identified as a library used in programming language. In additional 17.29% of sessions, attackers accessed the phpMyAdmin page with a Mozilla like or Opera browser and found out the versions of the phpMyAdmin, Apache, and even the Linux distribution.

Next, we briefly describe different types of *attacks* observed on our honeypots.

- *E-mail harvesting*[1] was done by four attackers in five sessions. Each session consisted of 49 requests, repeating the following sequence several times: list the directory structure, access each directory and list the files looking for e-mail addresses to harvest.
- *CVE-2008-3906* is a CRLF injection vulnerability in Mono 2.0 and earlier. One attacker launched this attack assuming Mono was running on the server.
- *CVE-2006-6374* is related to multiple CRLF injection vulnerabilities in phpMyAdmin 2.7.0-pl2.
- *Password cracking attacks* were attempts to access the MySQL server through port 80 by accessing the phpMyAdmin application. They were observed only on the advertised server. In all password cracking sessions attackers tried at most two single username/password combinations within each session, which prevents easy detection based on long sessions.
- *MySQL* attack was trying to break into the MySQL server by searching for the 'main.php' script in different locations on both the advertised and unadvertised server. This attacker obviously used IP-based strategy to reach the servers.

The last two type of attacks, Password cracking and MySQL, which constitute 8.94% of the total HTTP sessions on the advertised honeypot, are attacks that span multiple Web system components.

B. *MySQL traffic*

The MySQL database server, which runs on port 3306, received a significant portion of the traffic to both honeypots. The port scans originated from nine unique IP addresses which visited each honeypot only once. Vulnerability scans and attacks to specific known vulnerabilities directly to the MySQL servers were not observed since the servers were configured to reject the connections to port 3306 from remote users.

99.9% of the connections and packets that came to port 3306 originated from the same source IP address. This

[1] Harvesting e-mail addresses from the Internet is the primary way spammers build their lists.

	Advertised honeypot				Unadvertised honeypot			
	Sessions		Requests		Session		Requests	
Vulnerability scans: Total	**185**	**86.44%**	**443**	**43.95%**	**30**	**88.24%**	**37**	**69.81%**
DFind	17	7.94%	17	1.69%	16	47.06%	16	30.19%
OPTIONS	13	6.07%	13	1.29%	11	32.35%	11	20.75%
CONNECT	1	0.47%	1	0.10%	1	2.94%	1	1.89%
Fingerprinting								
Apache	26	12.15%	31	3.08%	1	2.94%	1	1.89%
PHP/phpMyAdmin	77	35.98%	71	7.04%	1	2.94%	8	15.09%
Multiple components	51	23.83%	310	30.75%	0	0.00%	0	0.00%
Attacks: Total	**29**	**13.56%**	**565**	**56.05%**	**4**	**11.76%**	**16**	**30.19%**
E-mail harvesting	5	2.34%	245	24.31%	0	0.00%	0	0.00%
CVE-2008-3906	1	0.47%	14	1.39%	0	0.00%	0	0.00%
CVE-2006-6374	4	1.87%	34	3.37%	3	8.82%	4	7.55%
Password cracking	18	8.41%	260	25.79%	0	0.00%	0	0.00%
MySQL	1	0.47%	12	1.19%	1	2.94%	12	22.64%
Total	**214**	**100%**	**1008**	**100%**	**34**	**100%**	**53**	**100%**

Table II
BREAKDOWN OF VULNERABILITY SCANS AND ATTACKS OF THE HTTP APPLICATION LEVEL TRAFFIC

attacker scanned port 80 on both honeypots and then returned 20 days later launching direct attacks on both MySQL servers. This shows that there may be a temporal dependence, often long time apart, between scans and attacks. The attack on each server lasted over two hours during which the attacker generated 23,663 connections to the advertised honeypot and 22,858 connections to the unadvertised honeypot. In these connections the attacker used almost every source port between 1025 and 5000, which suggests a use of an automated script. We suspect that these attacks were password cracking attempts, although we cannot be certain since MySQL servers did not allow direct access through port 3306 and thus no login information could be exchanged.

Note that MySQL server was fingerprinted and attacked through port 80, as a part of the multiple components vulnerability scans and attacks (see Table II).

C. SSH traffic

In case of the SSH protocol, the number of ports scans was also small; there were 8 port scans on the advertised and 5 port scans on the unadvertised server, which contributed respectively to only 0.05% and 0.02% of the total number of TCP connections to port 22. One attacker on each server first completed port scans to port 22, and then attempted a password cracking attacks.

The summary of the vulnerability scans and attack analysis is presented in Table III. Unlike the malicious HTTP traffic, the SSH traffic was dominated by attacks. On the advertised honeypot 23 unique IP sources first ran a vulnerability scan consisting of only one connection with 6-9 packets and than started a password cracking attack consisting of many connections, each with typically 10-15 packets. Very similar behavior was noticed on the unadvertised honeypot.

In case of almost all password cracking attacks, when an attempt to guess a pair of username and password failed, the attacker broke down the connection and tried again in a new connection with a different source port, most likely to avoid detection. The longest sequences lasted 4 hours and 51 minutes on the advertised honeypot, and 10 hours and 4 minutes on the unadvertised honeypot.

V. STATISTICAL CHARACTERIZATION

Based on the descriptive statistical analysis presented in section IV and characteristics of the nonmalicious traffic we suspected that heavy-tailed distributions may be a good model for some characteristics of the malicious traffic.

The simplest heavy-tailed distribution is the classical Pareto distribution with a shape parameter α and location parameter b which has the cumulative distribution function (CDF) $F(x) = P[X \leq x] = 1 - (b/x)^{\alpha}$. In practical terms, a random variable that follows a heavy-tailed distribution can give rise to extremely large values with non-negligible probability.

To estimate the tail index α of a Pareto distribution we employed the *log-log complementary distribution (LLCD) plots* and *Hill estimator* [8]. LLCD plot is a plot of the complementary cumulative distribution function $P[X > x] = 1 - F(x)$ on log-log axes. Linear behavior for the upper tail is an evidence of a heavy-tailed distribution. In that case, we select value for x from the LLCD plot above which the plot appears to be linear. Then, we estimate the slope, which is equal to $-\alpha$, using least-square regression.

Hill estimator is an alternative, more robust approach for estimating the tail index α of a semiparametric Pareto type model. It estimates α as a function of the k largest elements in the data set. Thus, for each value of k we obtain an estimate of the tail index parameter $\alpha_{k,n}$. When these estimates are plotted as a function of k, if the estimator stabilizes to a constant value this provides an estimate of α. The absence of such straight line behavior is an indication that the data are not consistent with Pareto-like distribution.

Our analysis included goodness-of-fit-testing for Pareto distribution, and in cases when the test failed fitting alternative distribution(s) into the data sample.

	Advertised honeypot			Unadvertised honeypot		
	Unique IPs	Connections	Packets	Unique IPs	Connections	Packets
Vulnerability scans	7	8	37	8	10	48
Vulner. scans followed by password attacks	23	16,725	201,467	21	28,592	344,047
Password cracking attacks	1	171	2.053	1	173	2,060
Total	**31**	**16,904**	**203,557**	**30**	**28,775**	**346,155**

Table III
BREAKDOWN OF THE SSH VULNERABILITY SCANS AND ATTACKS

A. Connections and packets per unique IP

The fact that 97% of all TCP connections to the advertised honeypot were generated by only 3.8% of the unique IP sources indicates that a heavy-tailed distribution may be a good fit. Similar observations were made in [10], [15], without actually performing the distribution fitting.

From the Hill plot of the connections per unique IP address shown in Figure 2, we observe that α is approximately 0.4, which is consistent with the estimate from the LLCD plot (see Figure 3). The high value of the coefficient of determination $R^2 = 0.91$ indicates a good fit between the empirical and mathematical distributions.

Hill estimator and LLCD plot also gave consistent estimates for the heavy-tailed index α for the number of packets on the advertised honeypot (≈ 0.4), and number of connections (≈ 0.4) and packets (≈ 0.3) on the unadvertised honeypot. In all cases these random variables follow a heavy-tailed distributions with both infinite mean and variance. In practical terms, this means that extremely large number of connections (or packets) can originate from a small number of attackers with non-negligible probability. These events from the tail of the distribution, although rare, often may have the mass of the probability distribution function, that is, generate the majority of the connections (or packets) to the server. An example of this is the malicious user who attacked the MySQL server directly on port 3306 and produced 99.9% of the MySQL connections and packets on both honeypots (see section IV-B).

B. Attributes of TCP connections

In this subsection we present statistical analysis of the TCP connection attributes: connection duration, number of packets per connection, and bytes transferred per connection, which to the best of our knowledge has not been done in the past for malicious TCP traffic.

The 3D scatter plot of all TCP connections is shown in Figure 4. We explored the correlation coefficients between the pairs of connection attributes. To measure the extent of the correlation, we use the Spearman's rank correlation coefficient r_s since the connection attributes, as shown later in this section, are not normally distributed. In addition, Spearman's correlation coefficient is rather robust to outliers, which may consist up to 20% of the data sample. The results show that Number of packets and Bytes transferred per TCP connection have the highest positive correlation $r_s = 0.98$ among the pairs of attributes, which is statistically significant at significance level of 0.05. Connection duration and Number of packets and Connection duration and Bytes transferred are also positively correlated with $r_s = 0.84$ and $r_s = 0.83$, respectively. The main reason for this positive correlation is the high number of points in the body of the distributions, that are closer to the origin in Figure 4. However, we also observe some long TCP connections, mainly due to the MySQL traffic, which have very few packets and bytes transferred. These connections are less than 1% and therefore do not affect the Spearman correlation coefficient.

Table IV shows the minimum, median, and maximum values of each connection attribute, for TCP traffic to all ports, and TCP traffic to port 80 for the advertised honeypot. Since our goal was to study whether these attributes follow a heavy-tailed distributions, we again used LLCD plot and Hill estimator to estimate the heavy-tail index α. In addition, we used Anderson-Darling (A^2) test [1] to test the null hypothesis that an attribute fits a specific distribution. This test is generally much more powerful than either of better known Kolmogorov-Smirnov or χ^2 tests, particularly for detecting deviations in the tail of a distribution.

As it can be seen from Table IV, the Number of packets and Bytes transferred per TCP connection follow a Pareto distribution, both with heavy-tailed index $1 < \alpha < 2$, that is, have a finite mean and infinite variance. It is interesting to explore what are the points in the tails of these distribution. Thus, 77.50% of the 80 connections in the tail of the Number of packets per TCP connection were attacks, including 38.75% e-mail harvesting and 35.00% password cracking attacks. The remaining 22.50% were due to vulnerability scans to multiple components which accessed the Web server with a Mozilla like or Opera browser. Out of the 250 connections in the tail of the Bytes transferred per connection, 62.00% were attacks (including 15.60% e-mail harvesting and 26.40% password cracking attacks through port 80 shown in Table II).

On the other side, the hypotheses that connection duration can be modeled with either Pareto distribution or log-normal distribution failed. Instead, log-logistic distribution with CDF $F(x) = [1 + [\beta/(x - \gamma)]^{\alpha}]^{-1}$ and parameter values given in Table IV is a good fit. Note that log-logistic distribution is similar in shape to the log-normal distribution, but has heavier tail. Unlike the Number of packets and

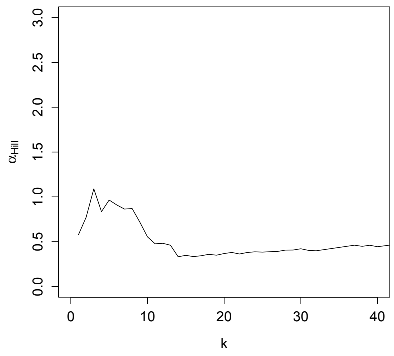

Figure 2. Hill plot

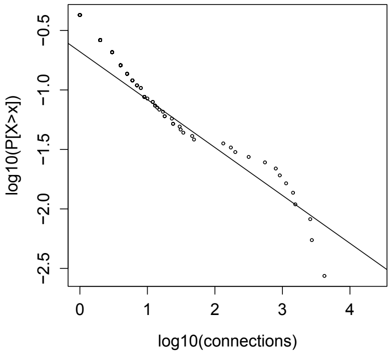

Figure 3. LLCD plot

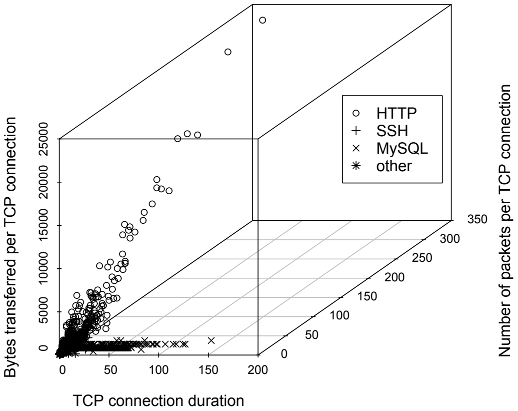

Figure 4. Malicious TCP traffic

	TCP all ports			TCP port 80		
	# of packets	Bytes transferred	Duration	# of packets	Bytes transferred	Duration
min/median/max	1/4/325	60B/242B/23.7KB	0/1/144 sec	1/14/325	66B/1.3KB/23.7KB	0/2/39 sec
Distribution Parameters	Pareto $\alpha = 1.2$ $b = 24$	Pareto $\alpha = 1.5$ $b = 1546$	Log-logistic $\alpha = 2.2092$ $\beta = 8.2387$ $\gamma = 49.321$	Pareto $\alpha = 1.1$ $b = 24$	Pareto $\alpha = 1.3$ $b = 1443$	Log-logistic $\alpha = 1.6118$ $\beta = 3.1898$ $\gamma = 14.963$

Table IV
DISTRIBUTION FUNCTIONS OF TCP CONNECTION ATTRIBUTES FOR THE ADVERTISED HONEYPOT

Bytes Transferred per TCP connection, the connections with the longest durations (approximately 1–2.4 minutes) were to destination port 3306 (i.e., password cracking attacks directly to the MySQL server).

The right part of Table IV shows the results of the same analysis, this time for the TCP connections to port 80 only. Perhaps the most interesting observation is that the heavy-tailness of the Number of packets and Bytes transferred per TCP connection are actually due to the TCP traffic to port 80 even though only 522 out of 41,359 connections were to port 80. This formally confirms the previous analysis which showed that almost all points in the tails of these two distributions were due to attacks or vulnerability scans to port 80 (i.e., HTTP traffic shown in Table II).

Figure 4 clearly explains the reasons behind this behavior. It is obvious that connections with large number of packets and bytes transferred all belong to the malicious traffic to port 80 (annotated by 'o'). MySQL traffic (annotated by 'x') had connections with variable duration, but very few packets and bytes transferred. Even more, all SSH connections were close to the origin in the 3D plot, with small duration, number of packets, and bytes transferred. (SSH connections are annotated with '+', but cannot be seen in Figure 4 because they overlap with HTTP, MySQL and other TCP connections close to the origin.)

The analysis of the TCP connection attributes for the unadvertised server is not as interesting, mainly due to the fact that this server has seen significantly less HTTP traffic. Thus, neither Number of packets nor Bytes transferred per TCP connection of the unadvertised server were heavy-tailed. Actually, these are not even skewed distribution which is obvious from the min/median/max values (1/12/14 packets and 60/1320/2640 bytes transferred). The most similar behavior to the advertised server has the Connection duration attribute, with min/median/max = 0/3/124 seconds, which is due to the fact that the longest TCP connections were due to traffic to port 3306 (MySQL). This, in addition to the results in section IV based on search-based strategies, clearly show that honypots deploying Web-based systems have to be advertised to reflect the realistic attackers activities.

VI. CONCLUSION

In this paper we presented an empirical analysis of attackers activity on typical multi-tier Web servers based on data collected by two high-interaction honeypots. We believe that it is of utmost importance to deploy honeypots that run typical configurations and fully functional systems to allow for realistic studies of attackers' activity. Although this approach led to more complex analysis, it allowed us to observe phenomena that would not have surfaced in a collection of independently running applications typically deployed on honeypots in related work. As an illustration we point out the vulnerability scans and attacks which spanned multiple components, such as fingerprinting Apache and phpMyAdmin in a single session (24% of HTTP sessions) and password cracking attacks to MySQL server through Apache and phpMyAdmin (9% of HTTP sessions). Our work also illustrates that Web-based honeypots have to be advertised to enable for search-based strategy, which appeared to be used for majority of vulnerability scans and attacks on port 80. Surprisingly, most of the honeypots from related work that included Web servers were not advertised.

An interesting observation is that the relative contributions of vulnerability scans and attacks are different for different protocols. Thus, while vulnerability scans tended to outnumber attacks for HTTP, attacks dominated the MySQL and SSH malicious traffic. Furthermore, password cracking attacks were prevalent, with some instances of attacks based on applications' vulnerabilities. The consequence of this observation is that using weak passwords may still be the weakest link in systems security, leading to many systems being compromised.

The statistical analysis showed that the number of connections and packets per unique attacker follow a heavy-tailed distribution with a small number of attackers submitting most of the malicious traffic. As illustrated by the TCP traffic directed to the MySQL server these heavy hitters drastically change the profile of the traffic, and although rare, can actually contribute to the majority of the connections and packets.

The analysis of the TCP connection attributes of the advertised server showed that the Number of packets and Bytes transferred per connection follow Pareto distribution with finite mean and infinite variance. This practically means that connections with extremely large number of packets and/or bytes transferred can happen with non-negligible probability. Perhaps the most interesting observation in this respect is that the heavy-tailness of these distributions is due to the HTTP component of the TCP traffic. On the other side, the distribution of the TCP connection duration, although skewed, is not heavy-tailed. The longest connections were not due to HTTP traffic; rather they belonged to the direct attacks to the MySQL server.

Potentially, there is a significant benefit from statistical modeling of the malicious traffic. For example, these models can be used for generating realistic malicious traffic for verification and validation of systems' security or to help the intrusion detection process. Our future work includes deployment of honeypots with different technologies and further statistical analysis of the malicious traffic.

Acknowledgments

This work is funded by the NSF CAREER grant CNS-0447715. The authors thank David Krovich and Jonathan Lynch for their support with the experimental setup.

References

[1] T. W. Anderson and D.A. Darling, "A test of goodness of fit," *J. Amer. Stat. Assn.*, vol. 49, 1954, pp. 765-769.

[2] E. Alata, V. Nicomette, M. Kaaniche, M. Dacier, and M. Herrb, "Lessons learned from the deployment of a high-interaction honeypot," *6th European Dependable Computing Conf.*, 2006, pp. 39-46.

[3] R. Berthier, D. Kormann, M. Cukier, M. Hiltunen, G. Vesdonder, and D. Sheleheda, "On the comparison of network attack datasets: An empirical analysis," *11th IEEE High Assurance Systems Eng. Symp.*, 2008, pp 39-48.

[4] R. Bloomfield, I. Gashi, A. Povyakalo, and V. Stankovic, "Comparison of emirical data from two honeynets and a distributed honeypot network," *19th Int'l Symp. Software Reliability Engineering*, 2008, pp. 219-228.

[5] P. T. Chen, C. S. Laih, F. Pouget, and M. Dacier, "Comparative survey of local honeypot sensors to assist network forensics," *1st Int'l Workshop on Systematic Approaches to Digital Forensic Eng.*, 2005, pp. 120-132.

[6] M. Dacier, F. Pouget, and H. Debar, "Honeypots: Practical means to validate malicious fault assumptions," *10th IEEE Pacific Rim Int'l Symp. Dependable Computing*, 2004, pp. 383–388.

[7] L. Gordon, M. Loeb, W.Lucyshynm, and R.Richardson, *Computer Crime and Security Survey*. Computer Security Insitute, 2006.

[8] K. Goseva-Popstojanova, F. Li, X. Wang, and A. Sangle, "A Contribution towards solving the Web workload puzzle," *36th IEEE/IFIP Int'l Conf. Dependable Systems & Networks*, 2006, pp. 505-514.

[9] N. Hohna, D. Veitch, and T. Ye, "Splitting and merging of packet traffic: Measurement and modelling," *Performance Evaluation*, vol. 62, 2005, pp. 164-177.

[10] M. Kaaniche, E. Alata, V. Nicomette, Y. Deswarte, and M. Dacier, "Empirical analysis and statistical modelling of attack processes based on honeypots," *Workshop on Empirical Evaluation of Dependability and Security*, 2006.

[11] http://www.honeynet.org/papers/webapp/

[12] R. McGrew and R. B. Vaughn, "Experiences with honeypot systems: Development, deployment, and analysis," *39th Annual Hawaii Int'l Conf. System Sciences*, 2006, p. 220a.

[13] S. Panjwani, S. Tan, K. Jarrin, and M. Cukier, "An experimental evaluation to determine if port scans are precursors to an attack," *35th IEEE/IFIP Int'l Conf. Dependable Systems & Networks*, 2005, pp. 602-611.

[14] F. Pouget, M. Dacier, and V. Hau Pham, "Leurre.com: On the advantages of deploying a large scale distributed honeypot platform," *E-Crime and Computer Conf.*, 2005.

[15] V. Yegneswaran, P. Barford, and J. Ullrich, "Internet intrusions: Global characteristics and prevalence," *ACM SIGMETRICS Int'l Conf. Measurement and Modeling of Computer Systems*, 2003, pp. 138-147.