Human Factors in Webserver Log File Analysis: A Controlled Experiment on Investigating Malicious Activity

Lucas Layman
Fraunhofer CESE
College Park, MD, USA
llayman@fc-md.umd.edu

Sylvain David Diffo
Karlsruhe U. of Applied Sciences
Karlsruhe, Germany
sylvaindiffo@gmail.com

Nico Zazworka
Elsevier Information Systems GmbH
Frankfurt am Main, Germany
zazworka@gmail.com

ABSTRACT

While automated methods are the first line of defense for detecting attacks on webservers, a human agent is required to understand the attacker's intent and the attack process. The goal of this research is to understand the value of various log fields and the cognitive processes by which log information is grouped, searched, and correlated. Such knowledge will enable the development of human-focused log file investigation technologies. We performed controlled experiments with 65 subjects (IT professionals and novices) who investigated excerpts from six webserver log files. Quantitative and qualitative data were gathered to: 1) analyze subject accuracy in identifying malicious activity; 2) identify the most useful pieces of log file information; and 3) understand the techniques and strategies used by subjects to process the information. Statistically significant effects were observed in the accuracy of identifying attacks and time taken depending on the type of attack. Systematic differences were also observed in the log fields used by high-performing and low-performing groups. The findings include: 1) new insights into how specific log data fields are used to effectively assess potentially malicious activity; 2) obfuscating factors in log data from a human cognitive perspective; and 3) practical implications for tools to support log file investigations.

Categories and Subject Descriptors

K.6.m [**Management of Computing and Information Systems**]: Miscellaneous—*security*

General Terms

Security

Keywords

security, science of security, log files, human factors

1. INTRODUCTION

Network cybersecurity attacks take on many forms, from network breaches and stealing of sensitive information to flooding a server with so much information that it becomes unresponsive. A study by Verizon and the U.S. Secret Service found that 98% of data theft took place on network servers, and that 86% of victims had evidence of the breach in server or application *log files* [17]. Despite the presence of such evidence, most victims did not discover the breach until weeks or months later (Figure 1).

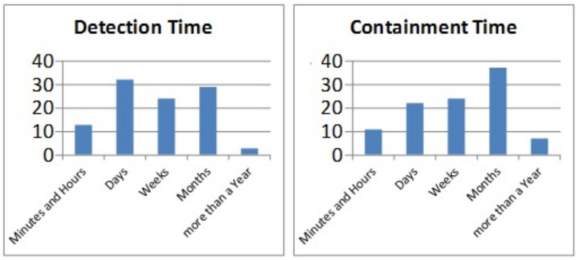


Figure 1: Typical times to detect and contain a security attack. [17]

The volume of network and log file information available has necessitated automated solutions for detecting cybersecurity attacks. These automated systems are useful for processing large amounts of esoteric information, but, like many predictive systems, generate a high number of false positives and false negatives [12] [16]. Human agents, e.g., an IT administrator, must often step in to perform "cyber forensics" – verify the attack, identify its origin, determine any losses, and take corrective action. The most commonly-used tools to support cyber forensics of log files are relatively primitive, often no more sophisticated than a text editor or Microsoft Excel coupled with word search. Scientific research is sparse on human-based cyber forensics and the technologies which could better support this activity.

1.1 The Science – Investigating Human Processses for Log File Investigation

The goal of this research is to deepen the scientific understanding of how human agents use the information in webserver log files to investigate potential attacks. Such research will generate findings and hypotheses regarding: 1) the type of information that is useful for identifying various classes of attack; 2) effective and ineffective strategies used to investigate attacks; 3) common discriminators between easy and difficult to find attack patterns; and 4) distinctions

HotSoS'14 April 08 - 09 2014, Raleigh, NC, USA
Copyright 2014 ACM 978-1-4503-2907-1/14/04 ...$15.00.
http://dx.doi.org/10.1145/2600176.2600185.

in the types of attacks discoverable by different classes of users (e.g., experts vs novices).

To address this goal, we conducted two controlled experiments with 65 cybersecurity experts and novices. The subjects examined excerpts from webserver log files that contained malicious behavior and normal activity. The subjects identified which log file fields were most useful for evaluating each log file for malicious activity. They also described their processes for analyzing the log files. We collected demographic information, performance data (accuracy and time taken), and post-mortem questionnaire responses. We performed a combination of quantitative and qualitative analyses to explore how the subjects investigated the log files and to uncover any interesting interactions between the type of attack, the expertise of the subjects, and the methods used to investigate the log files. Due to space limits, we focus on the quantitative analysis in this conference paper; a full discussion of the qualitative analysis may be found in [7].

The remainder of this paper is organized as follows: §2 provides background and related work; §3 describes our experimental design; §4 presents quantitative analysis findings; and we conclude in §5.

2. BACKGROUND AND RELATED WORK

Intrusion detection systems match known attack signatures and/or identify anomalous patterns that depart from "typical" behavior [4]. Human analysts, when exploring log data, employ this same strategy through inferential analysis and pattern recognition [1]. The most common challenge in investigating log files is the volume of data. A number of approaches have been proposed to address this issue, including correlating anomalous activities using multiple sources [16], data clustering and machine learning [5,15], and data visualization [6,14]. The presence of noise, misleading, or corrupted information in log files hinders the forensic process [3] of both humans and machines. One goal of this research is to better understand the confounding factors that inhibit log file analysis by human investigators.

There are potentially many parallels between log file investigation and the software engineering concept of *program comprehension*. Brooks' theory of comprehension of computer programs [2] observes that developers look for "beacons" in the code that appear relevant to the programming task, similar to IT security administrators searching for patterns indicative of anomalous behavior in a log file [1]. This a specific instance of "information foraging" [13]: the processes by which describes how humans allocate time and resources, identify relevant environmental queues, and select and pursue information in an environment where the information available adapts over time. This approach is susceptible to "anchoring" [11], wherein initial estimates of the desired finding incorrectly dominate the results. By better understanding the processes by which humans analyze log file data, we hope to provide guidance for tool development and investigative methodologies that support information foraging in log files while helping to avoid cognitive pitfalls in analysis.

3. CONTROLLED EXPERIMENT DESIGN

We conducted two experiments to identify which log file fields are most useful to users when investigating potential attacks. We also gathered data on the investigative processes and strategies used by subjects while analyzing the log files. We focus on web server log files as these are among the most prevalent and accessible sources of forensic data on computer networks. The format of log file data is rarely human-friendly, yet, buried in these files is often the best information to indicate potentially malicious activity. Figure 2 shows an excerpt from an IIS 7.0 log file from the Fraunhofer CESE webserver. Figure 3 illustrates the individual fields of a log entry. The last two lines of Figure 2 show an attacker trying to exploit a potential vulnerability in the webserver.

3.1 Observational study

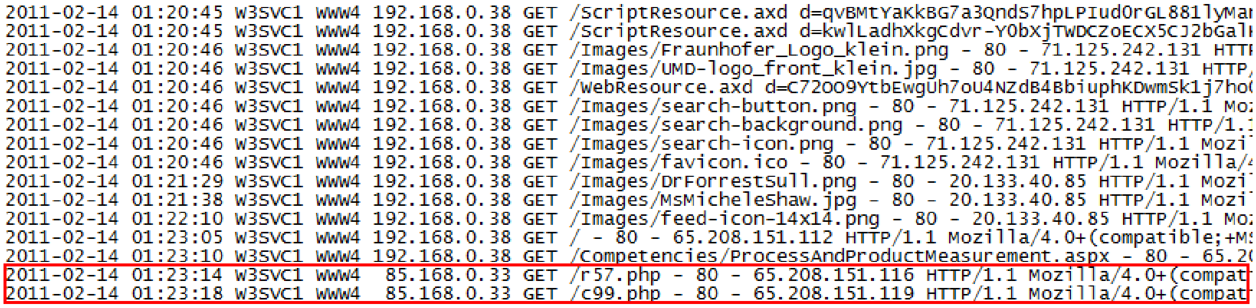
The first study was conducted with 14 self-selected subjects from Fraunhofer CESE and the University of Maryland (UMD) computer science department. UMD subjects were provided Amazon gift cards as compensation. In total, 8 students, 5 researchers, and 1 IT/Network administrator participated. The individual sessions took place in conference rooms at Fraunhofer and UMD with the first author present to administer the study. At the start of each study session, the subject completed a questionnaire on their experiences in IT computing, cybersecurity, and log file analysis (the questionnaire may be found in Appendix B of [7]).

To perform the experiment, the facilitator provided the subjects with a laptop. The laptop was connected to the Internet. During the log file analysis, the subjects' actions were recorded using Camtasia software. A microphone was placed on the table to capture audio along with the video recording.

The subjects were provided with 50-line excerpts from six log files of real web activity on Fraunhofer CESE webserver. Four of the six log files contained malicious behavior. A summary of the log files is provided in Table 1, and the full log files can be found in Appendix D of [7]. The log files were converted to Excel workbooks. Each log file contained the following log fields:

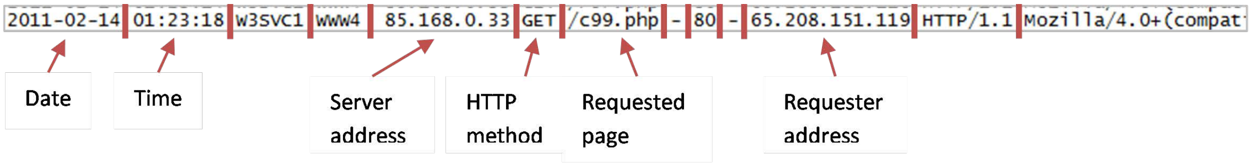
- a line number
- the date and time on which the event occurred.
- the HTTP method – i.e. GET or POST
- the IP address of the client who issued the request to the Fraunhofer web server.
- the client host name – the server or location name of the client IP address. This field was blank when IP could not be resolved to a hostname.
- the used protocol – i.e. HTTP
- response status code, e.g., 200 or 404
- the size of the requested resource – the number of bytes the server sent in response to the request
- the name of the requested resource (URL), i.e., the resource the user requested from the webserver

The subjects were first shown an example of a log file in Excel (which contained no malicious activity), and the facilitator described the different data fields. The subjects were told their task was to determine whether the log files shown to them had *normal, suspicious, or malicious activity* and provide a confidence rating from 1 (lowest) to 5 (highest). Subjects were instructed to think aloud to describe what they



```
2011-02-14 01:20:45 W3SVC1 WWW4 192.168.0.38 GET /ScriptResource.axd d=qvBMtYaKkBG7a3Qnd57hpLPIud0rGL881lyMa
2011-02-14 01:20:45 W3SVC1 WWW4 192.168.0.38 GET /ScriptResource.axd d=kwlLadhXkgCdvr-Y0bXjTWDCZoECX5CJ2bGal
2011-02-14 01:20:46 W3SVC1 WWW4 192.168.0.38 GET /Images/Fraunhofer_Logo_klein.png - 80 - 71.125.242.131 HTT
2011-02-14 01:20:46 W3SVC1 WWW4 192.168.0.38 GET /Images/UMD-logo_front_klein.jpg - 80 - 71.125.242.131 HTTP
2011-02-14 01:20:46 W3SVC1 WWW4 192.168.0.38 GET /WebResource.axd d=C72OO9YtbEwgUh7oU4NZdB4BbiuphKDwmSk1j7ho
2011-02-14 01:20:46 W3SVC1 WWW4 192.168.0.38 GET /Images/search-button.png - 80 - 71.125.242.131 HTTP/1.1 Mo
2011-02-14 01:20:46 W3SVC1 WWW4 192.168.0.38 GET /Images/search-background.png - 80 - 71.125.242.131 HTTP/1.
2011-02-14 01:20:46 W3SVC1 WWW4 192.168.0.38 GET /Images/search-icon.png - 80 - 71.125.242.131 HTTP/1.1 Mozi
2011-02-14 01:20:46 W3SVC1 WWW4 192.168.0.38 GET /Images/favicon.ico - 80 - 71.125.242.131 HTTP/1.1 Mozilla/
2011-02-14 01:21:29 W3SVC1 WWW4 192.168.0.38 GET /Images/DrForrestSull.png - 80 - 20.133.40.85 HTTP/1.1 Mozi
2011-02-14 01:21:38 W3SVC1 WWW4 192.168.0.38 GET /Images/MsMicheleShaw.jpg - 80 - 20.133.40.85 HTTP/1.1 Mozi
2011-02-14 01:22:10 W3SVC1 WWW4 192.168.0.38 GET /Images/feed-icon-14x14.png - 80 - 20.133.40.85 HTTP/1.1 Mo
2011-02-14 01:23:05 W3SVC1 WWW4 192.168.0.38 GET / - 80 - 65.208.151.112 HTTP/1.1 Mozilla/4.0+(compatible;+M
2011-02-14 01:23:10 W3SVC1 WWW4 192.168.0.38 GET /Competencies/ProcessAndProductMeasurement.aspx - 80 - 65.2
2011-02-14 01:23:14 W3SVC1 WWW4  85.168.0.33 GET /r57.php - 80 - 65.208.151.116 HTTP/1.1 Mozilla/4.0+(compat
2011-02-14 01:23:18 W3SVC1 WWW4  85.168.0.33 GET /c99.php - 80 - 65.208.151.119 HTTP/1.1 Mozilla/4.0+(compat
```

Figure 2: Excerpt from a Fraunhofer CESE webserver log file (Log D) showing r57.php and c99.php attacks.



```
2011-02-14 01:23:18 W3SVC1 WWW4  85.168.0.33 GET /c99.php - 80 - 65.208.151.119 HTTP/1.1 Mozilla/4.0+(compat
```

Figure 3: Example log file fields from a single request.

Table 1: Descriptions of log file samples used in the studies

Identifier	Malicious Activity	# attack lines	Description	OWASP Top 10 [10]
Log A	phpMyAdmin backdoor	6	A client scans for unsecured phpMyAdmin admin pages, which could give the client an attacker control over the server's database.	A4
Log B	SQL injection	1	A client attempts to use the website search box to execute SQL code to collect user logins, passwords, and email addresses.	A1
Log C	None	0	Normal website browsing activity by clients	n/a
Log D	r57 and c99 shells	2	A client attempts to access two known shell scripts that, if present, can give an attacker control over a system.	none
Log E	JavaScript injection	8	A client attempts to post JavaScript to the website that redirects website visitors to a spam site (cross-site scripting).	A3
Log F	None	0	Several benign search spiders and other bots access the site, but no overtly malicious activity.	n/a

were looking at, clicking on, and reasoning about during the log file analysis. Subjects were instructed that they could use any tool at their disposal to assist in the analysis. They were told they could use Excel's built-in functions (e.g., sorting, filtering), that they could modify the files, and use any other capability on the laptop they wanted. Subjects were told they could also use the Internet as a resource.

A log file was loaded into Excel by the researcher. Subjects were limited to 10 minutes on each log file to keep the total experimental session to close to 60 minutes for scheduling reasons, but they were not explicitly informed of this time limit. The facilitator noted the time taken by the subject on each log file and whether or not the subject found any attacks. After analyzing each log file, the subjects were asked to explain their reasoning for identifying a log file as suspicious or malicious. Subjects were asked to identify the three log file fields that were the most useful in their analysis. This data was entered into a pre-made form (see Appendix C of [7]. This process was repeated for all six webserver log files.

The order in which log files were presented was randomly generated without replacement for the subject sample set. After analyzing all six log files, the subjects were asked debriefing questions regarding what their own processes were for analyzing the files. Finally, the subjects were thanked for their time and the audio and video recording stopped. Between each subject, the recordings were transferred to a server and deleted from the laptop. Fresh copies of the log files were loaded, and the cache and browsing history of the laptop's web browsers were cleared.

For analysis, the researchers examined the facilitator's notes and the audio/video recordings to measure:

1. The log files the subject correctly classified as malicious or not

2. The number of times a log field was clicked on in the Excel sheet

3. The number of times a log field was mentioned by the subjects

4. The actions users took, e.g., browsing the web, filtering in Excel, highlighting a line of text

5. The time taken to complete each log file

6. The top 3 most useful log fields

In this paper, only items 1, 5, and 6 are analyzed.

3.2 Amazon Mechanical Turk study

The second study replicated the observational study on the web with a broader sample in terms of subject experience. The study materials, including instructions, random ordering of log files, demographic questionnaire, and debriefing questions, were all converted into a series of ASP.NET web forms. Because of the move to a web-based study, no think-aloud protocol or audio/video capture could be used. The study was anonymous and no personally-identifiable information was collected, although subjects could provide an email address if they were interested in the results.

A few changes were made to the study protocol. The demographic questions were clarified and expanded to be appropriate for a wider audience. Additional demographic questions were added to determine how participants had heard about our study and to provide feedback about the study website. The other significant change was to the classification mechanism for log file activity changed from Malicious, Suspicious, or Normal with a confidence rating to a continuous scale of: *Definitely normal* → *Probably normal* → *Don't know/need more info* → *Probably malicious* → *Definitely malicious*.

We solicited participants through Amazon's Mechanical Turk (http://www.mturk.com). The Mechanical Turk provides mechanisms for *requesters* to post tasks online where interested *workers* can complete those tasks in exchange for compensation. Requesters deposit money into an Amazon account and create a Human-Intelligence Task (HIT) using Amazon's web-based services. The requester also determines how much he/she will pay a worker to complete a HIT. The HIT is then posted on a searchable Amazon site, and workers can preview and/or begin work on the HIT.

Users found and completed our study through Amazon's Mechanical Turk interface. We established an initial beta testing period with $30 compensation to eliminate bugs in the test instrument – data from this period are excluded from analysis. Following test, we deployed the experiment to the Mechanical Turk and offered $15 in compensation. The data submitted to our HIT (i.e. questionnaire responses and log file analysis results) were reviewed by a researcher for completeness and approved before workers received compensation.

We implemented a 2.5 minute, 10 question pretest to ensure that participants had the requisite knowledge to analyze the log files with some degree of accuracy. If a worker missed more than two questions on the pretest, they were not able to begin our study. A discussion of the adaptations and challenges incurred in migrating the observational study to the Mechanical Turk environment may be found in [8].

We received 60 participants in a three-week period. Participants found the study by searching on the Amazon site and through social networking on cloud worker forums. The time to complete the HIT was set to 90 minutes to discourage outside collaboration (no subjects exceeded this time limit). The experimental data were collected and stored on a Fraunhofer webserver. Five subjects did not complete the experiment due to technical errors in the online test instrument, two subjects' data were rejected because they clicked through the website quickly without participating in earnest, and one subject's data was rejected because he took the experiment twice (his first set of data was kept).

3.3 Subject demographics

The subjects classified themselves into one of five groups according to their main job responsibilities:

- IT and Network Administrators
- Researchers in Computer Science or a related field
- Software Engineering-related jobs, such as developer, tester, project manager
- Students, graduate or undergraduate
- Other, including Educators, IT Hobbyists, Bloggers, and Unemployed engineers.

Table 2: Study participants' job roles

Role	Obs. study	MTurk study	Total
IT/Net Admin	1	10	11
Other	0	8	8
Researcher	5	2	7
Soft. Eng.	0	21	21
Student	8	11	19
Total	14	51	65

The breakdown of subjects according to their self-reported job roles is shown in Table 2. The professional (i.e. non-student) subjects reported a range of experiences in their respective fields, with 41% having 1-5 years, 23% having 5-10 years, and 31% having 10+ years of experience.

We also collected data to approximate the subjects' expertise and experience in cybersecurity. We asked subjects "How familiar are you with Internet cybersecurity attacks?" Subjects could select one or more of the following:

- None or very little
- Read about attack mechanics
- Attacked or defended in a controlled setting
- Defended or investigated attacks on a public network
- Engineered software to attack or defend

We also asked the subjects ”Have you ever analyzed webserver log files (or similar log files”, to which they could select one or more of:

- No
- In class or Internet examples
- On a private network
- On a public network.

Subjects could select multiple answers for both questions. A complete breakdown of subject responses according to job role may be found in §2.3 of [7].

Across all subjects in both studies, 84.6% indicated they had some experience either reading about cyber attacks or defending against them, and 33.9% of subjects had investigated or defended against an attack on a public or corporate network. Furthermore, 83.1% of the subjects had experience analyzing webserver log files. Given these figures, we assert that: 1) most of our study subjects had at least a rudimentary understanding of cybersecurity threats; and 2) most of our subjects were qualified to understand the contents of the log files in our experiment.

3.3.1 Subject clustering according to cyber attack and analysis experience

We wanted to analyze differences between experienced and inexperienced groups in terms of their performance, time taken, and useful log data fields. To identify these groups, we used the unsupervised EM (Expectation-Maximization) algorithm to automatically cluster subjects according to their responses to three questions pertaining to expertise in cybersecurity and log file analysis. The clusters are characterized by the highest weighted response (e.g., ”I have engineered attacks”) in the cluster. The EM algorithm produced four clusters, which we interpret as follows:

- Educated and trained: Individuals who have read about attacks, defended against them in a controlled setting, and have some experience defending public networks.
- Controlled experiences: Individuals who have read about attacks and have some experience defending against them in a controlled setting only.
- On-the-job experience: Individuals who may have read about cyber attacks, but most of their experience is defending public networks and analyzing public network log files.
- No experience: Individuals with little to no experience with cyber attacks or log files.

The distribution of subjects among these clusters is shown in Table 3.

4. ANALYSIS AND FINDINGS

Our analysis focuses on quantitative analysis of subject accuracy in correctly identifying a log activity as malicious or not, the time taken to analyze the log files, and the most useful log fields identified by the subjects. We analyze these data according to subject role (i.e. job description), experience cluster, and accuracy. In a subsequent publication, we will provide the complementary qualitative analysis of subject strategies for searching, collating, and interpreting log file data.

Table 3: Clusters of users based on cybersecurity and log file analysis experience

Cluster	# of subjects	%
Educated and trained	15	23%
Controlled experiences	23	35%
On-the-job experience	12	19%
No experience	15	23%

4.1 Log file activity assessment accuracy

Of the six log files used in the study, four contained malicious activity (see Table 1). Overall, 16 subjects correctly assessed all 6 log files, 18 subjects were correct on 5/6, 19 were correct on 4/6, 9 were correct on 3/6, and 3 were correct on 1/6. The correctness of subject responses is defined in Table 4.

4.1.1 Accuracy by log file

The aggregated results of the subjects’ assessments of whether the log file activity was malicious or not for both the observational and Mechanical Turk studies are shown in Table 5.

Approximately 75% of the subjects’ assessments were correct. The outlier is Log D, where nearly half of the subjects were incorrect. The accuracy for Log D was significantly lower than all other log files (McNemar’s χ^2, $p<0.05$). The malicious activity in Log D is on two lines that show requests to two different files, `r57.php` and `c99.php` (see Figure 2 above), which are backdoor scripts that provide full control over the system. In Log D, the attacker was determining whether these script files existed on the webserver.

The data suggest two contributors to the 45% error rate in classifying this log file. First, the attack only appears on two lines out of 50. Second, the two lines result in 404 errors, but these two lines are surrounded by other 404 errors from a search crawler requests (hostname is `new-search.umd.edu` in Figure 2). In post-analysis questioning, six individuals mentioned the surrounding search crawler, but only two mentioned the attack lines. This leads to **Observation 1: the failed requests containing the two attack lines are obscured by the surrounded surrounding 404 noise** despite the distinctively short URLs, the `.php` extension, and the `.ru` originating hostname.

4.1.2 Accuracy by role

We examined the accuracy of the subjects’ assessments according to their job role. Overall, researchers were the most accurate, while software engineers were the least accurate (Table 6).

Log D assessments were significantly less accurate (two-sided Fisher’s exact test, 95% CI). The significant difference is due to 71% of Software Engineering-related job roles and 45% of IT/Network Admins missing the attack. A combined 76% of Students and Researchers correctly classified Log D. The majority of students and researchers in our study were associated with the University of Maryland, and thus may have inferred the `new-search.umd.edu` activity was legitimate because the domain was familiar to them. This leads to **Hypothesis 1: familiarity with the network on which a server operates, and who normally visits a webserver, is useful when determining whether a**

Table 4: Correctness table for subject assessments of log file activity

	Actual	Correct response	Incorrect
Obs. study	Malicious	Malicious | Suspicious	Normal
	Normal	Normal	Malicious | Suspicious
MTurk study	Malicious	Probably malicious | Definitely malicious	Undecided | Probably normal | Definitely normal
	Normal	Probably normal | Definitely normal	Undecided | Probably malicious | Definitely malicious

Table 5: Summary of log file assessment accuracy

	Log A		Log B		Log C		Log D		Log E		Log F		Total	
	#	%	#	%	#	%	#	%	#	%	#	%	#	%
Correct	58	89.2%	50	76.9%	51	78.5%	36	55.4%	47	72.3%	50	76.9%	292	74.9%
Incorrect	7	10.8%	15	23.1%	14	21.5%	29	44.6%	18	27.7%	15	23.1%	98	25.1%

Table 6: Summary of log file assessment accuracy (p-value for 2-sided Fisher's exact test)

Role	Log A		Log B		Log C		Log D		Log E		Log F		Total	
	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.
IT/Net Admin	9	2	11	0	9	2	6	5	8	3	9	2	2	14
Other	7	1	4	4	6	2	5	3	4	4	6	2	32	16
Researcher	7	0	5	2	5	2	6	1	6	1	5	2	34	8
Soft. Eng.	18	3	17	4	19	2	6	15	15	6	18	3	93	33
Student	17	1	13	5	12	6	13	5	14	4	12	6	81	27
Total	58	7	50	15	51	14	36	29	47	18	50	15	292	97
%	89.2	10.8	76.9	23.1	78.5	21.5	55.4	44.6	72.3	27.7	76.9	23.1	74.9	25.1
p-value		0.742		0.096		0.416		**0.027**		0.622		0.679		

user's intentions are malicious or not. These findings suggest that log file analysts would benefit from automation that helps to reduce or filter out innocuous noise and helps the user to become familiar and ask questions about the network ecosystem.

4.1.3 Accuracy by experience cluster

The assessment accuracy according to cybersecurity experience is shown in Table 7. The only statistically significant difference is in Log C, where 47% of those with no cybersecurity experience and 26% of those with experience only in controlled settings misclassified this normal log file as malicious. Log F was also misclassified as malicious but without a statistically significant effect. These misclassifications may be due to inexperience, and, conversely, the experience of the Educated and On-the-job group may help to identify normal behavior. The false positive classifications of Log C and F may have been induced by the Hawthorne Effect wherein the subjects are *expecting* to find attacks due to the nature of the study.

When subjects were asked to explain why they thought activity was malicious, many of the inaccurate subjects expressed concern over the requests from client hostnames containing terms like ”crawler” and ”spider”. We note that the crawlers and spiders in Logs C and F were from unfamiliar domains (i.e. not Google, Yahoo, or Bing). Some subjects were concerned by the number of failed requests (return status 404) generated by the search spiders. For the experts, this was apparently not a concern. As one expert put it, ”Spiders issuing requests that return 404s is not abnormal. I tend to only worry about 404s when I think that a human is on the other side of the request.” Thus, we reach **Hypothesis 2: some non-experts may be overly suspicious of bots and proxies simply because of their names.**

4.2 Analysis of time spent

A 10 minute time limit was imposed for each log file in the observational study, and the Mechanical Turk study was limited to 90 minutes total. Five data points are excluded from time analysis (from three students and one researcher) from the observational study where subjects were stopped at the 10 minute limit.

Time data for the Mechanical Turk study has an unknown error factor since there was no guarantee that the subjects were always focused on the task. The distribution of total time taken to perform the experiment is shown in Figure 4; the distribution is non-normal (Shapiro-Wilk, $p<0.001$) and positively skewed. The median time spent by subjects to complete analysis of all six log files was 1393s (23:13), with a minimum of 409s (6:49), and a maximum of 4484s (1:14:44).

4.2.1 Analysis of time spent per log file

The median time taken to analyze a log file was 213s (3:33). The distributions of time taken for each log file were non-normal with a positive skew (Shapiro-Wilk, $p<0.001$ for each

Table 7: Summary of log file assessment accuracy (p-value for 2-sided Fisher's exact test)

	Log A		Log B		Log C		Log D		Log E		Log F		Total	
	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.
Trained	12	3	14	1	14	1	7	8	9	6	14	1	70	20
Controlled	20	3	15	8	17	6	14	9	16	7	15	8	91	41
On-the-job	12	0	11	1	12	0	5	7	10	2	11	1	61	11
None	14	1	10	5	8	7	10	5	12	3	10	5	64	26
p-value	0.465		0.093		**0.009**		0.524		0.548		0.093			

log file and for total time). Figure 5 shows the time taken per log file. Log E (JavaScript injection) had the highest median analysis time with 320s (5:20), and Log C (normal activity) had the lowest median time with 155s (2:53). The differences in time taken for each log file were statistically significant (Friedman's ANOVA $p<0.001$).

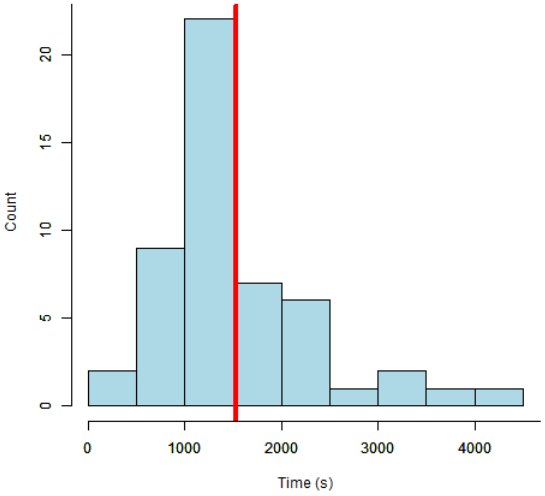


Figure 4: Total analysis time taken across subjects

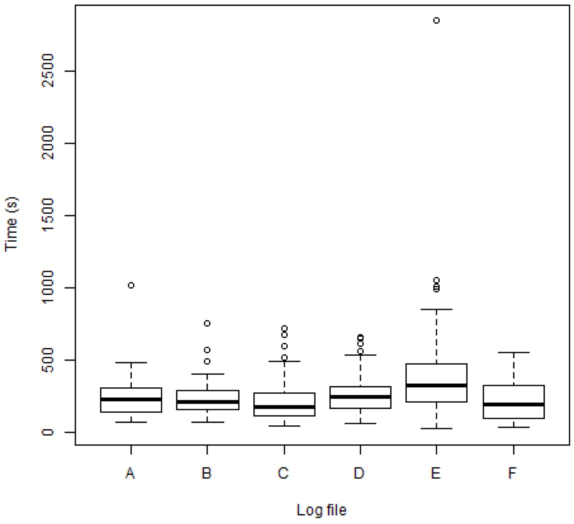


Figure 5: Time taken per log file

Subjects took significantly longer to analyze Log E than Logs A, B, C, and F (Tukey's HSD test significant at $p<0.05$), yet the subjects' overall accuracy in assessing Log E was not statistically different than the other log files (Table 6). A sample of two of the eight attack lines in Log E are shown in Figure 6. The subjects likely spotted all eight attack lines quickly as the "Requested Page" field contained noticeably longer text than other lines. We hypothesize that the longer times spent are due to subjects: 1) attempting to parse the hexidecimal encoded strings; and 2) verifying that the site embedded in the request was malicious. This leads to **Observation 2: log file analysts may benefit from automation that helps them to parse hexadecimal characters, and to look up information about a particular website or domain**.

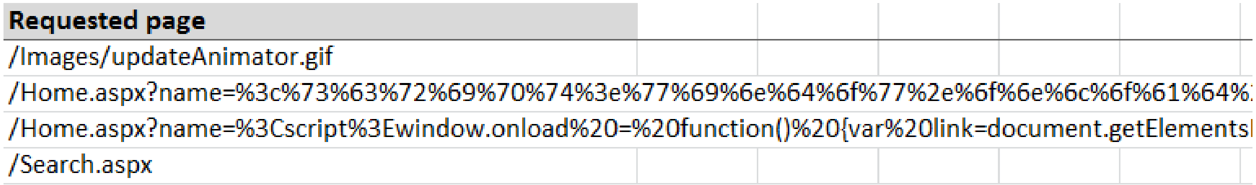
4.2.2 Analysis of time spent per job role

The job groups had similar median total times analyzing each of the log files – the Kruskal-Wallis rank sum test for the total log file analysis time by role (χ^2 = 1.505, df=4, p=0.829) indicates no significant differences among the job roles at the 95% confidence interval. For individual log files, analysis shows no statistically significant differences at the 95% confidence interval (Kruskal-Wallis, $p>0.05$) for all log files by role.

Let us revisit our earlier finding in §4.1.2 that Software Engineers and IT/Network Admins were less accurate in assessing Log D. When creating one group with Software Engineers + IT/Network Admins and another group with all other roles, the differences in time taken on Log D are significant at the 95% confidence interval (Kruskal Wallis, χ^2=3.953, df=1, p=0.047). The Software Engineering and IT/Network Admin group took significantly less time to analyze Log D, which may be a contributing factor to their lower accuracy. This group was also significantly faster when analyzing Log C (Kruskal Wallis, χ^2=5.273, df=1, p=0.022), which contains normal (non-malicious) behavior. Yet, there is no significant difference in the accuracy of the two groups with respect to Log C. In this case, perhaps the IT/Networking and Software Engineering group is more experienced at "whitelisting" favorable data or are more confident in their ability to quickly assess a log file for possible threats.

4.2.3 Analysis of time spent per experience cluster

A summary of the total time taken to complete log file analysis by the groups clustered by cybersecurity experience is shown in Figure 7. The difference in the distribution of time taken between groups is statistically significant for Log C (χ^2=11.068, df=3, p=0.011) and Log F (χ^2=10.51, df=3, p=0.0147) at the 95% confidence interval; all other logs are



Requested page
/Images/updateAnimator.gif
/Home.aspx?name=%3c%73%63%72%69%70%74%3e%77%69%6e%64%6f%77%2e%6f%6e%6c%6f%61%64%
/Home.aspx?name=%3Cscript%3Ewindow.onload%20=%20function()%20{var%20link=document.getElements
/Search.aspx

Figure 6: Excerpt of cross-site scripting attack lines from Log E

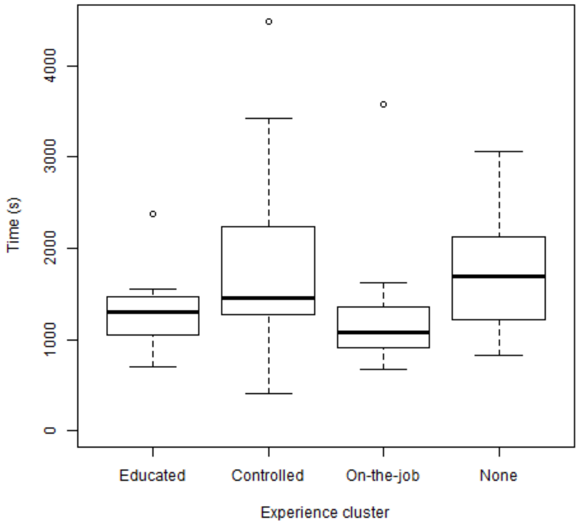


Figure 7: Total analysis time by experience cluster

non-significant. Logs C and F are those containing normal (non-malicious) behavior. For both log files, subjects in the Controlled and None groups took longer than subjects in the Educated and On-the-job groups. This increase in time may be due to the less experienced groups taking longer to inspect suspicious activity, whereas the more experienced groups were relatively quick to white-list such activity as normal. As discussed in §4.1.3, these less experienced groups were more likely to incorrectly identify these files as malicious.

4.3 Most useful log file fields

For each log file, the subjects were asked to identify the most useful, 2nd most useful, and 3rd most useful log fields in assessing whether the log activity was malicious or normal. Choosing the 2nd and 3rd most useful field was optional. The frequency of each log field's ranking was tabulated for each log file, e.g. "Requested Page" was selected as most useful for Log A by 46 participants, "Status" was second most useful for Log A by 23 participants, etc. The raw data may be found in §2.6 of [7].

By a wide margin, *the field most frequently identified as "the most useful" was the Requested Page* (ranked the most useful in 65.4% of cases). The requested page is the most direct indicator of a client's intentions and desires. The requested page can contain strings indicative of common attacks, such as SQL injection, Javascript injection, and access to backdoor shells or unsecured applications on the webserver.

The *four fields were most frequently identified as "2nd most useful": Hostname (29.2%), Client IP (20.2%), Requested Page (18.6%), and Status (17.5%).* These fields are used in conjunction with the most useful field to make an assessment. The Client IP and Hostname are useful for determining if a request is from a malicious (or innocuous) source, whereas the Status code helps to quickly identify failed requests, which may be linked to malicious resources. The 3rd most useful field is evenly distributed among most of the fields.

It is worth noting what was *not* deemed a useful field. The Protocol and the Size of Response were deemed less useful by the subjects (ranked as most useful, 2nd most useful, or 3rd most useful in less than 5% of cases). This suggests **Observation 3: the protocol and size of the response could be hidden when displaying log information on HTTP webservers to reduce the amount of information presented to users.**

The distributions of the useful fields vary according to log file. For example, the 2nd and 3rd most useful fields for Log E contain higher proportions for the Method field than in the other log files due to the POSTing of JavaScript code (Figure 6). The subjects likely had different "beacons" [2] that raised suspicions in each log file. The processes and strategies used for individual log files are beyond the scope this paper; a discussion of how the fields are used, and the strategies employed by our subjects, may be found in [7].

4.3.1 Most useful log file fields according to individual log file accuracy

We examined the most useful fields of those subjects who correctly assessed a log file's activity versus those who were incorrect. In all log files, the Requested Page was the most useful field among both correct and incorrect respondents. Table 8 shows the most useful field for each log file with the Requested Page removed to gain insight on how other fields influence the performance of the subjects.

The focus on Status in Log D by the incorrect subjects is interesting. Log D contains many 404 (file not found) status codes that are unrelated to the actual attack. As discussed in §4.1.1, it could be the case that the incorrect subjects became *desensitized to 404 status codes in this log file, thus making it more difficult to identify the two attack lines* (Figure 2).

The focus on Date/Time in Log F is also noteworthy. Log F contains several instances where a client issues five or more requests in less than a second. Log F also contains activity from a Russian search spider that accesses the website approximately once per hour. Though these activities are not malicious, we offer **Hypothesis 4: the timing of requests may lead less experienced respondents to incorrectly identify activities as malicious**.

Table 9 shows the 2nd most useful field per log file based on assessment correctness. While the correct respondents focused on the Hostname, the incorrect respondents tended to focus on the Client IP. The hostname field may contain semantically meaningful information about the geographic

Table 8: Most useful field per log file (Requested Page removed)

	Log A	Log B	Log C	Log D	Log E	Log F
Correct	status	hostname	hostname status	hostname	client IP hostname	status
Incorrect	client IP	hostname	hostname	status	hostname	date/time

Table 9: Second useful field per log file

	Log A	Log B	Log C	Log D	Log E	Log F
Correct	status	hostname	hostname	hostname	hostname	hostname
Incorrect	client IP	client IP	client IP	hostname	hostname status requested page status	client IP hostname

origin of a request, e.g., a hostname ending in `.ru` originates from Russia. While a savvy user can gather the origin of a Client IP, less experienced users may not. Based on additional qualitative data [7], we suggest **Hypothesis 5: frequency and repetition of requests from the same Client IP may lead to incorrectly labeling innocuous activities as malicious**. As discussed in the previous paragraph, this seems to indicate a focus on *request timing* by the inexperienced subjects. Furthermore, in their debriefing questions, some respondents are distrustful of non-human agents. Such agents (e.g., crawlers) are likely to request multiple resources in a short amount of time, which does not necessarily indicate malicious intent.

4.3.2 Most useful log file fields according to overall subject performance

Table 10: Performance groups

Overall accuracy	# subjects	Group
6/6	16	Perfect
5/6	18	High
4/6	19	Middle
3 or less	12	Low

We grouped the subjects into similarly sized groups according to their overall accuracy at log file analysis (Table 10). We then analyzed the most useful fields according to these groups.

Table 11 shows the 2nd most useful field per log file according to the performance group. Both the Perfect performers and the Low performers look at the Client IP for Log A and Log B. This lends weight to our earlier discussion that the more savvy users may know how to elicit meaningful information from the Client IP, whereas the low performers may become distracted by the frequency of requests. Importantly, we note that the Hostname is *not* a part of the default log file content – the Hostname field in our log files was generated using a script to reverse lookup the hostname based on the Client IP. Based on the overall usefulness of the Hostname, we offer **Observation 4: logging technologies and post-mortem analysis tools provide support for looking up the hostname of a given IP**.

Among the Perfect performers, the appearance of Method in Log E is also noteworthy. Log E is the only log file that contains HTTP POST methods, which implies that the client is attempting to send data to the server. POSTs occur, for example, when a client uses a search box on a website (the search terms inputted by the user are POSTed to the website). POST methods are a common way to perform injection attacks, such as the malicious activity in Log E (Figure 6). The Perfect-performers identified these POST methods as suspicious, whereas the other groups utilized other data fields ahead of the method. This implies **Observation 4: POST methods may indicate potentially malicious activity on a website that typically does not accept user inputs**.

4.4 Threats to Validity

Internal validity – The subjects in both studies were self-selected in exchange for compensation, and thus were not drawn from a truly random population of technically qualified individuals. In the observational study, we observed a maturation effect where the median time taken to analyze the first log file was approximately 2.5 minutes longer than the last log file. To minimize this effect on our results, the order in which log files were presented was randomized (without replacement) across subjects. The log file assessment instrument was changed between the two studies, and the results were combined using the schema in Table 4. In the observational study, 20/84 log files were marked ”suspicious”, which we assert is equivalent to ”probably malicious” given the subjects' think aloud statements and the definition of the word ”suspicious”. In the Mechanical Turk study, 17/306 (5.5%) of responses were marked ”Don't know/need more info”, which was always counted incorrect – the sensitivity of our analyses to these responses has not been tested.

External validity – The controlled experiments introduced constraints that limit generalizability, including the presence of an observer in the observational study, time constraints, and a limited toolset for investigating the log files. Subjects in the Mechanical Turk study may have reacted to the pretest by focusing on log file fields mentioned in the pretest. A Hawthorne effect was likely present and may have increased the rate of false positives in Logs C and F

Table 11: Second useful fields by performance group

	Log A	Log B	Log C	Log D	Log E	Log F
Perfect	Client IP Status	Client IP	Hostname	Hostname	Method	Status
High	Hostname	Hostname	Client IP Requested Page	Hostname	Status	Status
Middle	Status	Status Client IP	Hostname Status	Requested page	Hostname	Hostname
Low	Client IP	Client IP	Hostname Status	Requested Page	Hostname	Status

for inexperienced subjects. The log file samples were from the Fraunhofer CESE IIS webserver, which received 8-10K requests per day. The log samples represent a constrained set of attacks, two of which were seeded by the researchers (Logs B and E); the rest of the traffic data in the log files (including the attacks in Logs A and D) are real data.

The log files samples are limited to 50 lines, but in a complete log file, factors affecting fatigue, noise, and search will undoubtedly influence attack investigation. Further, sophisticated attacks will likely be spread out over multiple systems and applications. Thus, the data focused on by investigators may change under more realistic conditions and warrants further study.

Conclusion validity – The reported statistical results use mostly non-parametric tests whose assumptions were met. The greatest threat to conclusion validity is combining the data from the two studies. The prevalence of less-experienced students in the observational study and professional engineers in the Mechanical Turk study manifested in a number of significant differences discussed throughout §4. However, the overall performance in the two studies nearly the same: 73.8% accurate in the observational study and 75.2% accurate in the Mechanical Turk study. Regarding time data, five data points from the observational study exceeded the 10 minute time limit - these points (and thus the artificial constraint) were removed from all time analyses.

5. CONCLUSION

This paper presents results from two controlled studies conducted with a total of 65 IT professionals and novices who were investigating potentially malicious activity in webserver log files. Our goal was to understand which log fields were most valuable and how the fields were used by the subjects. Analysis of the quantitative data yielded a number of statistically significant effects, and the qualitative feedback from participants provided key insights into the subjects' thought processes. The findings include:

1. 404 webserver responses are useful indicators of attacks, but are *easily obfuscated by 404 noise*
2. *familiarity with the network ecosystem* (e.g. typical visitors and requests) is important for ”white-listing” acceptable behavior
3. non-experts may *erroneously fixate on the timing of requests* as an indicator of malicious activity
4. non-experts may be *overly suspicious of bots*
5. the *Requested Page* is most useful log file field for investigations by a wide margin
6. the presence of *POST methods* can be very useful for identifying certain classes of attack

Many security technologies, out of necessity, focus on extracting potential attack information from a large volume of data. However, human investigators performing network monitoring or cyberforensics will benefit from tools and technologies that are attuned to human problem solving processes. In addition to the implications of the findings listed above, our research has identified specific recommendations for tools to support human log file analysis, e.g.:

1. Provide support to convert hexidecimal to printable characters.
2. Enable hiding or removing less useful fields, such a Protocol and Response Size.
3. Provide support to resolve IP addresses to hostnames.

We have operationalized these concepts in the InViz tool [9] (http://www.fc-md.umd.edu/inviz), which provides real-time visualization of webserver events. This research is part of ongoing work to better understand the human factors that influence attack detection and investigation. In future publications, we will provide qualitative analysis of the processes and strategies used by the subjects in our studies to group, search, and correlate log file activities when investigating potential attacks. Future work will include replication of this experiment to determine if the hypotheses hold under varying conditions.

Acknowledgement

This work was funded by a Fraunhofer USA Special Funds grant. The authors would like to acknowledge Ulrike Becker-Kornstaedt, Davide Falessi, Forrest Shull, and Marcel Schwarzmann for their help in designing and analyzing these studies. The authors would also like to thank the participants in the study for contributing their valuable data, insights, and experiences.

6. REFERENCES

[1] D. Botta, R. Werlinger, A. Gagné, K. Beznosov, L. Iverson, S. Fels, and B. Fisher. Towards understanding IT security professionals and their tools. In *Proceedings of the 3rd symposium on Usable privacy*

and security - SOUPS '07, page 100, New York, New York, USA, July 2007. ACM Press.

[2] R. Brooks. Towards a theory of the comprehension of computer programs. *International Journal of Man-Machine Studies*, 18:543–554, 1983.

[3] E. Casey. Investigating sophisticated security breaches. *Communications of the ACM*, 49(2):48, Feb. 2006.

[4] D. Denning. An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, Feb. 1987.

[5] K. A. Garcia, R. Monroy, L. A. Trejo, C. Mex-Perera, and E. Aguirre. Analyzing Log Files for Postmortem Intrusion Detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1690–1704, Nov. 2012.

[6] T. Jankun-Kelly. Detecting Flaws and Intruders with Visual Data Analysis. *IEEE Computer Graphics and Applications*, 24(5):27–35, Sept. 2004.

[7] L. Layman. Information and Processes Used When Investigating Web Server Log Files for Malicious Activity. Technical report, Fraunhofer Center for Experimental Software Engineering, College Park, MD, 2013.

[8] L. Layman and G. Sigurdsson. Using Amazon's Mechanical Turk for User Studies: Eight Things You Need to Know. In *Proceedings of the 7th International Symposium on Empirical Software Engineering and Measurement (ESEM 2013)*, pages 275–278, Baltimore, Maryland, USA, 2013.

[9] L. Layman and N. Zazworka. InViz: Instant Visualization of Cyber Attacks. In *Proc. of the 2014 Symposium and Bootcamp on the Science of Security (HotSoS '14)*, page to appear, Raleigh, NC, 2014.

[10] OWASP. Top 10 2013, http://goo.gl/umM0Br, 2013.

[11] J. Parsons and C. Saunders. Cognitive Heuristics in Software Engineering: Applying and Extending Anchoring and Adjustment to Artifact Reuse. *IEEE Transactions on Software Engineering*, 30(12):873–888, 2004.

[12] T. Pietraszek. Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection. In E. Jonsson, A. Valdes, and M. Almgren, editors, *Recent Advances in Instrusion Detection*, volume 3224 of *Lecture Notes in Computer Science*, pages 102–124. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[13] P. Pirolli and S. Card. Information Foraging. *Psychological Review*, 106(4):643–675, 1999.

[14] H. Shiravi, A. Shiravi, and A. Ghorbani. A Survey of Visualization Systems for Network Security. *Visualization and Computer Graphics, IEEE Transactions on*, PP(99):1, 2012.

[15] R. Vaarandi. A data clustering algorithm for mining patterns from event logs. In *Proceedings of the 3rd IEEE Workshop on IP Operations & Management (IPOM 2003) (IEEE Cat. No.03EX764)*, pages 119–126. IEEE, 2003.

[16] F. Valeur, G. Vigna, C. Kruegel, and R. Kemmerer. Comprehensive approach to intrusion detection alert correlation. *IEEE Transactions on Dependable and Secure Computing*, 1(3):146–169, July 2004.

[17] Verizon. 2010 Data Breach Investigations Report, http://goo.gl/28pPGM. Technical report, Verizon, 2010.