**A Survey of Web Vulnerabilities**

by

Benjamin Fogel

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 10, 2015

Keywords: Web Vulnerabilities, Empirical Study



Approved by

Munawar Hafiz, Assistant Professor of Computer Science and Software Engineering
Anthony Skjellum, Professor of Computer Science and Software Engineering
Jeffrey Overbey, Assistant Professor of Computer Science and Software Engineering

Abstract

This study tracked the patching characteristics of the top 100,000 sites to three vulnerabilities: the POODLE attack, the POODLE TLS attack, and the FREAK attack. The study also carried out a survey on top server administrators asking specific questions of the POODLE attack and general questions about an administrator's decision process. The goal was to identify how the web reacts and responds to known vulnerabilities in addition to finding characteristics and tendencies of secure websites. Our research found a slow, yet steady patching rate for all vulnerabilities for most sites. Additionally, our research found little evidence that a site vulnerable to one vulnerability would be vulnerable to another. Lastly, our research found that server administrators are not able to keep with the evolving world of web vulnerabilities due to greater concerns of compatibility and server up time.

Acknowledgments

I would like to thank Dr. Munawar Hafiz for his guidance and support throughout this research. I would also like to thank Dr. Jeffrey Overbey and Dr. Anthony Skjellum for being in my committee and giving their time to review the work done.

Contents

List of Figures

List of Tables

List of Abbreviations

| | |
|---|---|
| CBC | Cipher Block Chaining |
| Freak | Factoring RSA Export Keys |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | HTTP using SSL or TLS |
| OSI | Open Systems Interconnection model |
| Poodle | Padding Oracle On Downgraded Legacy Encryption |
| RC4 | Rivest Cipher 4 |
| SSL | Secure Sockets Layer |
| TLS | Tranpsort Layer Security |

# Chapter 1

# Introduction

In 1999, less than 5% of the global population used the Internet according to the International Telecommunications Union [22]. By 2014, the ITU reported that 40% of the global population now uses the Internet [23]. The Internet has transformed how society functions through everyday life. The drastic increase in Internet usage is not hard to understand. Banking, voice calls, and driving instructions are among many things have been re-implemented using the Internet.

With an ever-expanding purpose, the Internet has become the primary interaction for many services. However, since the Internet has become a central hub for important activities, a new issue has risen: security. An IBM Study found that cyber attacks reached an all time high with a 12% increase of security incidents in 2013 [12].

Security incidents can result from a variety of vulnerabilities. While the IBM study shows that most incidents result from server misconfiguration [11], exploits found in widely used code can arguably be considered more harmful. If exploits are found within shared code or implementations, then there is a much larger percentage of the Internet susceptible to the exploit. From October of 2014 to April of 2015, there were many vulnerabilities reported affecting a significant amount of the Internet. We focus on three vulnerabilities which target weaknesses in the SSL/TLS protocol. These 3 incidents are known as the Poodle attack (October 2014), the Poodle attack against TLS (December 2014), and the Freak Attack (March 2015).

## 1.1 Research Goal

The main goal of the research found in this thesis aims to identify how the web reacts and responds to known vulnerabilities, particularly the 3 vulnerabilities previously mentioned. Furthermore, this research aims to assort the Internet in ways to identify trends and tendencies within certain categories of the Internet.

## 1.2 Thesis Statement

*On the Internet, security vulnerabilities emerge unpredictably and often undermine the mitigation effort. Even servers that see a lot of Internet traffic remain vulnerable after a vulnerability is known and fixes are available. Server administrators face a tough task to keep the servers up-to-date in terms of security vulnerabilities despite their best effort.*

*To demonstrate the difficulty of fighting Internet vulnerabilities, we demonstrate three vulnerabilities in a short period of time on the same target. We then observe how the protection mechanism are adopted at a slow rate across popular websites. Lastly, we determine the difficulties of securing a server that a server administrator encounters from a survey.*

## 1.3 Approach

In this thesis, high traffic web servers were analyzed with respect to the previously mentioned vulnerabilities. The web servers include the Top 100,000 websites defined by Alexa's Top 1 million global sites[4].

A custom web scanner, initially built by Shane Farmer, was built using a collection of tools that obtained information including available cryptographic protocols, available cipher suites, and open ports. From this information, a server could be determined vulnerable or protected against certain vulnerabilities. The scanner was run roughly on a bi-weekly interval, such that no two running scans ever overlapped.

Scanned websites were sorted into multiple categories based on site ranking, site type, and security characteristics obtained from the scanning. Comparisons were made between categories to identify categories with strong security versus categories with weak security.

Finally, a security survey was sent to a random sampling of website administrators. The survey asked general questions of the server configuration and administrator patching characteristics as well as specific questions about the Poodle attack. The results from this survey would provide insight into reasons for certain security configurations.

All this information is used to identify certain responses to web vulnerabilities. In addition, characteristics can be identified which lead to either strong security or weak security. These observations can help the community identify desirable characteristics and responses which lead to secure servers.

### 1.4 Organization of Thesis

Chapter 2 recounts the background of our research. Chapter 3 introduces works related to this study. Chapter 4 describes the experiment plan of this study. Chapter 5 analyzes and discusses the results of our observations on the Poodle attack, the Poodle TLS attack, the FREAK attack and the survey. Chapter 6 describes the conclusions and future work.

Chapter 2

Background

## 2.1 A Brief Overview of Cytographic Protocols

Most HTTP connections are secured with either Secure Sockets Layer (SSL) or Transport Layer Security (TLS). These cryptographic protocols provide the foundation of secure communication on the Internet. Either of these cryptographic protocols used in conjuction with the HTTP protocol create the HTTPS protocol. Referring to the OSI model, these protocols provide security at the transport layer. This allows HTTP or other application layer protocols to use these cryptographic protocols to ensure security [6].

In 1994, Secure Sockets Layer was invented by Netscape Communications as a response to Internet security concerns [21]. SSL communications initially perform a handshake which is shown in Figure 2.1.

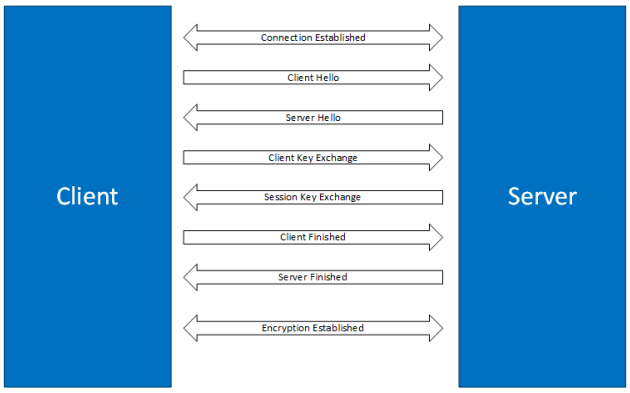


Figure 2.1: Simple SSL Handshake Sequence

At the beginning of the handshake, the client sends a "hello" message to the server. This message contains certain information about the client including the protocol number and the list of cipher suites available to the client [20].

Once the server has received the client's "hello" message, another "hello" message is sent from the server back to the client. This message contains the version number the server will support as well as the cipher suite the server will support. After the "hello" messages are exchanged, a standard key exchange is performed between the client and server until a secure communication is established.

In the years following its introduction, SSL underwent several modifications in order to improve security. There were two version released after SSL which were SSL2 and SSL3. In 1999, SSL3 underwent another improvement but was renamed by the Internet Engineering Task Force to the Transport Layer Security Protocol [9].

While TLS provided some implementation improvements, the protocol still followed the original SSL handshake mechanisms. Also while each SSL version and subsequently each TLS version improved on the prior versions, the new protocols never supplanted the old. In fact, SSL2 and SSL3 are still in use on some browsers and servers.

## 2.2 Three Recent Attacks on the SSL/TLS protocol

### 2.2.1 Poodle Attack

On October 24, 2014, a Google security team unveiled an attack known as Poodle which stands for Padding Oracle On Downgraded Legacy Encryption [16]. Adam Langley, another Google researcher, describes the Poodle attack as a fundamental design flaw in SSL/TLS where SSL/TLS authenticates before encryption. [14].

The Poodle attack exploits this design flaw by performing a downgrade. Servers will try to choose the highest transport layer security version possible when first connecting. However, in an effort to maintain an established connection, servers commonly allow clients to downgrade to lower versions if needed. A third party could also intercept message between the server and client and force a downgrade itself.

For example, a connection could be downgraded from TLS 1.0 to SSL3. Langley points out that SSL3 allows cipher suites known to be weak, specifically the RC4 and CBC based

cipher suites [14]. In CBC-based ciphers, there is extra padding added to the end of every request as well as a byte specifying the padding length. If the padding length were increased, it could reveal parts of the message meant to be encrypted. Due to the weaknesses, a single byte can be decrypted on average in 256 requests [14]. This allows the possibility of a man-in-the-middle attack who could decrypt messages between the client and the server after a downgrade.

The Google security team proposed a patch at the time of disclosure that a flag be added to SSL/TLS implementations on both clients and servers. The flag, TLS_FALLBACK_SCSV, would disallow the downgrade from TLS versions to SSL and save both clients and servers from a possible Poodle attack. An alternate approach proposes disabling SSL 3 altogether. While this hinders clients that exclusively encrypt with SSL 3 or lower, it secures against the Poodle attack.

### 2.2.2 Poodle TLS Vulnerability

On December 8, 2014, another vulnerability similar to the original Poodle attack was released [19]. This vulnerability exploits similar padding flaws without the need for downgrading. Since TLS is an upgrade of SSL3, vulnerabilities found in SSL3 founds its way into some TLS implementations [15].

Since the Poodle TLS vulnerability affects all versions of the SSL/TLS protocol, vulnerable SSL/TLS implementations need to be reimplemented to protect against the padding problems. These implementations are not normally performed by server administrators, so most sites must wait for software vendor updates to secure against this vulnerability.

### 2.2.3 Freak Attack

In the early 1990s the United States government placed restrictions on the cipher suites that were exported to other countries [6]. The restrictions specifically disallowed the export of cipher suites with a key length greater than 512 bits [6]. While the restriction has been

lifted, the weak cipher suites continue to be implemented to ensure compatibility anywhere outside the United States. These cipher suites are known as the export cipher suites.

On March 3, 2014, the FREAK attack, otherwise known as the Factoring RSA Export Keys attack, was announced by a research team led by Karthikeyan Bhargavan. The team found that "that several implementations incorrectly allow the message sequence of export ciphersuites to be used even if a non-export ciphersuite was negotiated" [5]. The implementation error allows a man-in-the-middle style attack to downgrade a secure non-export cipher suite to an RSA export ciphersuite. The team then states that the key could be extracted in only 8 hours using $100 on an Amazon EC2 instance [5]. This shows the practicality and severity of the Freak Attack.

In order to protect against the vulnerability, server administrators need to remove all RSA export cipher suites from their accepted cipher suite collection. Additionally, clients can protect themselves against the vulnerability by upgrading to a browser that does not support any RSA export ciphersuites.

# Chapter 3

# Review of Literature

Server configurations are not only important to their respective institutions but to the Internet altogether. Internet clients rely on server configuration statistics to decide which features to implement and which legacy features can be removed. It is important to keep a current understanding of server configurations to adapt against possible vulnerabilities in a timely manner. The following paragraphs discuss various studies that consider server configurations.

## 3.1 *Non-compliant and Proud: A Case Study of HTTP Compliance*

The IETF, or Internet Engineering Task Force, is responsible for specifying Internet standards that the community will implement. These responsibilities include updating the HTTP standard to match contemporary implementation needs as well as security needs. By keeping current with HTTP standards, the adopting community would inherently be more secure. In June 1999, the IETF released the HTTP/1.1 protocol to supersede the previous HTTP/1.0 protocol [2]. Adamcyzk, Hafiz, and Johnson studied the compliance of and rate of adoption of the HTTP 1.1 protocol [1].

Adamcyzk, Hafiz, and Johnson speculated that while servers are easily capable of complying with HTTP protocol, websites chose to be non-compliant. In the study, "Non-compliant and Proud: A Case Study of HTTP Compliance", the implementation and configuration of eight HTTP methods were investigated for the top 100 websites according to Alexa, as well as the top 25 computer science department websites.

The study made an important distinction between the implementation and configuration of each HTTP method. While the implementation of each HTTP method explores whether

the method accepts any signature on that method, the configuration compares the method signatures with the standards defined by HTTP/1.1. In addition, the IETF specifies a recommendation of compliance for each protocol standard of either MUST, SHOULD, or MAY [7]. The study rates the compliance of the methods for each site referring to the levels given by the IETF. The study found that while websites implemented all of the methods, most websites failed to correctly comply with HTTP/1.1 standards. In conclusion, the study proposed that "security concerns, the limited use of most HTTP methods, and HTTP-agnostic systems" could explain why HTTP/1.1 has not been correctly configured, even 8 years after the standard was released [1].

### 3.2 SSL Pulse

In 2012, SSL Labs created a project named SSL Pulse which monitors Alexa's top 1 million websites and reports general statistics about SSL and TLS implementations [13]. SSL Pulse publishes information on a monthly basis and includes Heartbleed, Poodle TLS, and supported SSL/TLS versions. This information roughly illustrates the SSL/TLS security practices of the Internet. Table 3.1 aggregates some of the information presented by SSL Pulse that is relavent to this study.

The table shows that there was a steady decrease of vulnerable sites after the disclosure of both Heartbleed and the Poodle TLS vulnerability. One of the more interesting findings shows that the BEAST attack vulnerability has increased in percentage, from 69.4% to 81.5%, in November 2, 2013 till March 4, 2015. This can be attributed to to levels of

| | | Vulnerability | | |
|---|---|---|---|---|
| | | Heartbleed | Poodle TLS | |
| Time since Disclosure (months) | 0 | - | 10.1% | |
| | 1 | 0.8% | 7.3% | |
| | 2 | 0.7% | 6.2% | |
| | 4 | 0.5% | - | |

Table 3.1: Percentage of Sites Vulnerable to a Vulnerability tracked by SSL Pulse

publicity of each vulnerability. The BEAST attack has been disclosed since 2004, while the Heartbleed and Poodle attacks are currently prominent vulnerabilities.

Although SSL Pulse provides very useful information, there are some insights missing. For example, SSL Pulse does not provide information strictly about the Poodle attack. Additionally, SSL Pulse does not provide demographics of the top one million websites. Still, SSL Pulse is an excellent resource that provides statistics of SSL/TLS information for the Internet. In fact, Ivan Ristic, author of SSL Labs, uses the information presented by SSL Pulse to create *SSL/TLS Deployment Best Practices*, an article which recommends optimal server configuration practices [18].

### 3.3 *The Matter of Heartbleed*

In April 2014, a vulnerability affecting OpenSSL was discovered known as Heartbleed. *The Matter of Heartbleed*, a study by Durumeric, et al., performed "a comprehensive, measurement based analysis of the vulnerabilitys impact" [10]. In the study, one million websites were monitored for patching rates and patching behavior.

Durumeric, et al., found that while the top 500 websites were all patched quickly, less popular websites responded less quickly and "plateaued after about two weeks" with 3% remaining vulnerable 2 months after disclosure [10]. Durumeric, et al., also hypothesized that a delayed patching response was partly due to a lack of advanced notice given to system vendors.

# Chapter 4

# Experiment Plan

The main goal of this thesis was to identify how web servers responded to web vulnerabilities. Since the study only initially tracked the poodle vulnerability, extensive information was needed to account for future vulnerabilities. Furthermore, certain vulnerabilities require additional information to be added to the data acquisition process.

The data was collected from October 2014 to March 2015 in such a way to maximize the number of observations while disallowing any 2 scans from overlapping. In addition to the vulnerability information, demographic information was collected about each observed site. The following paragraphs detail the data acquisition process as well as the subsequent analysis.

## 4.1 Data Acquisition

### 4.1.1 Scope of Study

In this study we chose to observe the top 100,000 websites as defined by Alexa's Top 1 Million Websites. By using the top 100,000 websites, this study is able to cover an excellent representation of frequently visited sites across the web. This was also a small enough number to allow more frequent scans to track minute changes and trends among the 100,000 websites.

### 4.1.2 Methodology

Initially, this study only tried to identify characteristics of the Poodle vulnerability. A tool, initially built by Shane Farmer, was built using a vulnerable version of OpenSSL toolkit

to identify which sites were still vulnerable. This study also gathered data that could be useful for future vulnerabilities. By using the same OpenSSL toolkit, information about SSL/TLS version was also collected about the sites. The last information collected was the site's ciphersuite information found by the cipherscan tool. The initial scanning code can be found in Appendix A.

Over the course of the study, the scans were performed as often as possible without any overlap between subsequent scans. Table 4.1 shows the dates the study performed each scan. Since features were added as the study progressed, scans took longer to complete towards the end of the time period.

| **Scan Dates** |
|---|
| 11/13/2014 |
| 11/22/2014 |
| 12/12/2014 |
| 12/28/2014 |
| 1/18/2014 |
| 2/12/2014 |
| 3/1/2014 |
| 3/21/2014 |

Table 4.1: Scan Dates

On December 8th, the TLS variant of the Poodle was announced. In order to start tracking a site's vulnerability to Poodle TLS, tlsprober was introduced to scans starting on December 12th, 2014. The FREAK attack was a vulnerability that demonstrated weaknesses in certain ciphers used. Since every site's ciphersuite had been recorded since the initial scan, no modifications were required to observe the FREAK attack.

Categorical information about each site was collected using Alexa analytics. Also, 68 common ports were scanned using nmap starting on March 1, 2014. An additional scan on March 21, 2014 found that these ports remained unchanged, and this study assumes port status to be static during the analysis. The category information, port information, and Alexa rank of each site was used to find similar characteristics of secure or insecure servers.

## 4.2 Tools

### 4.2.1 OpenSSL

OpenSSL is widely used open-source toolkit that provides SSL/TLS implementations in addition to crytography suites [17]. While OpenSSL is often used in servers, this research only utilizes the OpenSSL client program, s_client. The program provides a generic SSL/TLS client which provides useful information about the connection and status of the SSL/TLS protocol.

This research uses OpenSSL to collect information about server protocol implementations. A specific OpenSSL version that provides the SCSV flag option was required to gather data about the Poodle attack. Because of this reason, this study uses OpenSSL version 1.0.1j.

### 4.2.2 cipherscan

Cipherscan is an open source tool that utilizes a custom version of OpenSSL to provide ciphersuite information. Cipherscan reveals ciphersuite information of a site including ciphersuite priority, ciphersuite name and ciphersuite supported protocols. This information was ultimately useful to identify sites using export ciphersuites vulnerable to the FREAK attack.

### 4.2.3 tlsprober

TLSProber is an open source tool developed by Opera Software ASA. The tool provides a large amount of information about SSL/TLS server implementation. The tool was one of the first open source tools to provide information about whether a server was vulnerable to the Poodle TLS vulnerability.

### 4.2.4 NMap

NMap is a simple, yet powerful network scanner. This study uses nmap to perform its port scans. NMap is able to identify the state of the port, the possible service running on a port, and a possible reason of a port's state.

## 4.3 Survey

In order to better understand the decisions behind server configurations, a survey was conducted asking questions about server administrator behaviors. Table 4.2 lists the questionnaire provided in the survey.

| |
|---|
| What OS are you running? |
| What web server are you running? |
| Is your server self-hosted or cloud-hosted? |
| Can you please estimate your monthly web traffic? (visits/month) |
| Please choose the options you could describe the content hosted on your server as |
| Have you heard of the Poodle attack? |
| If yes, please describe how you have heard of the attack. |
| In general, how do you stay updated about new vulnerabilities? Please provide details or list your information sources if possible. |
| Have you considered patching your server to prevent against the Poodle Attack? |
| In general, how recently have you patched or re-configured your server for security reasons? |
| In general, how often do you patch or re-configure your server? |
| Do you prefer to patch your server as updates are distributed or do you perform multiple updates at once? |
| Can you please describe a specific security incident you were patching? |
| Can you describe the process that you followed? |
| Please list some reasons as to why an update should be skipped? |
| Can you describe an update that you chose to update at a later time? |
| Can you describe a recent update that you chose to skip? |
| Are there any potential attack vectors that you prioritize defending? |
| Do you monitor for unusual (suspicious or malicious) server activity? |
| Do you have automated detection for unusual server activity? |
| Do you redirect http traffic to https? |
| Do you remotely administer your server? |
| Do you use VPN in order to remotely control your server? |
| Is your server a dedicated web server or does it server other purposes? |
| If applicable, list other purposes served by your server. |
| Have you disabled or removed the default accounts on the server? |
| Are you also the primary developer of your web application? |
| Do you maintain a blacklist or whitelist of IPs or IP ranges that can access your server? |
| Do you run any configuration management software? |

Table 4.2: Questionnaire

Questions 1-5 ask general information about the server. Questions 6-9 inquire directly about the Poodle attack and tries to identify information sources about vulnerabilities. Questions 10-19 attempts to identify patching characteristics by the server administrator. The rest of the survey asks more specific information related to the configuration of the server.

Surveys were emailed to site administrators identified by a site's whois information. Since a large number of sites employed a privacy service protecting their email information, a large enough sampling was needed to reach an adequate number of site administrators. 2,910 surveys were sent to server administrators categorized by their site ranking, and the site's poodle vulnerability status. Table 4.3 show how many sites were chosen per category for the survey.

| | 1-100 | 101-1,000 | 1,001-10,000 | 10,001-100,000 |
|---|---|---|---|---|
| Vulnerable to Poodle | 2 | 11 | 63 | 252 |
| Protected against Poodle | 5 | 72 | 595 | 1910 |

Table 4.3: Number of Sites Receiving Survey by Rank and Vulnerability To Poodle

# Chapter 5

# Results

## 5.1 Poodle

The Poodle attack was the initial vulnerability tracked by this study. Our initial scan started on November 11th, 18 days after the Poodle attack was announced. Of the top 100,000 sites scanned, we were able to collect information for 61,288. We were unable to collect information on sites which did not have port 443 open The initial scan showed an overall poodle vulnerability rate of 20.2%.

| Top 1 - Top 100 Sites | Top 101 - Top 1000 Sites | Top 1001 - Top 10000 Sites | Top 10001 - Top 100000 Sites |
|---|---|---|---|
| 47.76 % | 25.83 % | 17.40 % | 20.42 % |

Table 5.1: Initial Vulnerability Percentage Per Rank Category

Table 5.1 shows the initial vulnerability percentage per rank category. The top 100 sites actually displayed the highest percentage of poodle vulnerability while the top 1,001 - top 10,000 sites had a lower percentage.

Table 5.2 lists the initial vulnerability percentage by Alexa category. Category information could only be retrieved from Alexa on 19,135 sites out of the 61,288 total sites. 17 out of 18 categories were below the overall poodle vulnerability rate. Additional, sites that did not have any category information had an initial poodle vulnerability rate of 21.50 %.

| Category | Vulnerability Percentage | Category Site Count |
|---|---|---|
| World | 19.00 % | 8959 |
| Regional | 14.05 % | 3595 |
| Computers | 18.38 % | 2008 |
| Business | 12.40 % | 1839 |
| Reference | 13.62 % | 1109 |
| Shopping | 11.56 % | 900 |
| Society | 19.54 % | 696 |
| Art | 19.97 % | 686 |
| Recreation | 12.29 % | 407 |
| Science | 18.26 % | 367 |
| Games | 17.35 % | 317 |
| Sports | 18.52 % | 270 |
| Health | 17.32 % | 231 |
| Home | 13.10 % | 229 |
| News | 14.10 % | 220 |
| Kids_and_Teens | 20.77 % | 207 |
| Adult | 12.63 % | 95 |

Table 5.2: Initial Vulnerability Percentage Per Alexa Category

| Port Number | "open" | "filtered" | "closed" |
|---|---|---|---|
| 20 | 10.3 | 15.54 | 29.68 |
| 21 | 30.05 | 15.44 | 20.33 |
| 22 | 30.21 | 15.76 | 22.42 |
| 23 | 12.6 | 18.8 | 26.74 |
| 25 | 33.72 | 13.05 | 20.82 |
| 53 | 37.52 | 13.91 | 21.55 |
| 79 | 11.33 | 18.98 | 26.07 |
| 80 | 20.7 | 14.79 | 14.71 |
| 110 | 33.68 | 15.06 | 20.31 |
| 123 | 9.71 | 18.93 | 26.21 |
| 143 | 34.21 | 14.97 | 20.46 |
| 161 | 10.38 | 19.05 | 25.94 |
| 194 | 10.07 | 18.97 | 26.04 |
| 389 | 55.71 | 18.21 | 26.1 |
| 443 | 20.75 | 12.04 | 5.52 |
| 445 | 7.53 | 19.26 | 26.53 |
| 465 | 35.92 | 14.73 | 21.47 |
| 500 | 10.11 | 18.94 | 26.09 |
| 512 | 9.96 | 18.95 | 26.14 |
| 513 | 9.48 | 18.99 | 26.04 |
| 520 | 10.22 | 18.97 | 26.01 |
| 587 | 33.6 | 15.57 | 22.17 |
| 636 | 56.93 | 18.18 | 26.13 |
| 993 | 34.54 | 15.15 | 20.26 |
| 995 | 34.34 | 15.25 | 20.29 |
| 1026 | 8.71 | 18.55 | 26.62 |
| 1241 | 10.44 | 18.64 | 26.39 |
| 1243 | 11.04 | 18.54 | 26.58 |
| 1433 | 5.52 | 18.62 | 26.74 |
| 1444 | 10.47 | 18.55 | 26.54 |
| 2048 | 57.49 | 17.74 | 26.57 |
| 2049 | 47.88 | 17.64 | 27.2 |
| 2525 | 16.28 | 18.59 | 26.38 |
| 3104 | 10.63 | 18.52 | 26.58 |
| 3269 | 10.41 | 18.52 | 26.62 |
| 3306 | 32.84 | 17.79 | 20.8 |
| 5000 | 54.46 | 17.73 | 26.62 |
| 5353 | 10.71 | 18.5 | 26.6 |
| 5432 | 10.59 | 18.61 | 26.51 |
| 5555 | 12.89 | 18.49 | 26.62 |
| 5900 | 9.97 | 18.49 | 26.62 |
| 5987 | 10.37 | 18.53 | 26.56 |
| 7002 | 56.89 | 17.74 | 26.56 |
| 8000 | 49.62 | 17.74 | 26.09 |
| 8080 | 12.32 | 20.09 | 27.05 |
| 8082 | 53.59 | 17.74 | 26.41 |
| 8089 | 55.71 | 17.75 | 26.48 |
| 8172 | 5.97 | 18.66 | 26.45 |
| 8443 | 15.42 | 20.02 | 25.46 |
| 8834 | 10.26 | 18.54 | 26.53 |
| 8835 | 10.17 | 18.54 | 26.52 |
| 8888 | 50.6 | 17.45 | 26.44 |
| 9090 | 52.13 | 17.72 | 26.61 |
| 9200 | 57.38 | 17.73 | 26.55 |
| 9443 | 56.39 | 17.74 | 26.55 |
| 12321 | 11.36 | 18.54 | 26.52 |
| 20000 | 55.5 | 17.65 | 26.37 |
| 32764 | 10.47 | 18.05 | 27.17 |
| 32775 | 10.37 | 18.03 | 27.22 |
| 32776 | 10.23 | 18.03 | 27.22 |
| 32777 | 10.14 | 18.03 | 27.21 |
| 32778 | 10.37 | 18.02 | 27.23 |
| 49152 | 7.29 | 18.42 | 26.84 |
| 49253 | 10.3 | 18.25 | 26.8 |
| 52731 | 10.47 | 18.46 | 26.51 |
| 53337 | 10.37 | 18.47 | 26.49 |

Table 5.3: Initial Vulnerability Percentage Per Port

Table 5.3 lists the initial vulnerability percentage categorized by ports and their respective statuses. There are 2 different scenarios for a port. Either a port with an open status has a higher vulnerability percentage or a lower vulnerability percentage than the closed port. The scenarios can be explained better with additional date points and thus will be discussed later in this section.

Another metric was also used to categorize sites. This study defines this metric as the port security limit. The port security limit is the maximum number of open ports on any given server before a server is deemed exposed. The port security limit for the poodle attack is 5 open ports. Any server that contains more than 6 open ports is categorized as exposed. Servers containing 5 or less open ports are categorized as restricted. The initial scans found an initial vulnerability percentage for restricted servers to be 14.75%, while the vulnerability percentage of exposed servers was 34.34%.

Given the initial state of vulnerability to poodle, the following information covers the state of the poodle attack across all scans. Figure 5.1 demonstrates the poodle vulnerability over time for each rank category. Some interesting observations can be identified from the figure. First, the top 100 rank category demonstrates a much higher percentage of vulnerability compared to the other categories. Additionally, after December 12th, there were no poodle patches made on the top 100 servers. But the top 100 also showed a much higher rate of patching from November 13 through December 12. The remaining categories all followed approximately the same trend, where patching rates slowed as time passed.

Figure 5.2 demonstrates the poodle vulnerability over time for all categories. We anticipated that financial related sites would be more secure. Sites categorized under Arts, Society, or World demonstrated the highest vulnerability rates while sites categorized under shopping or business had the lowest vulnerability rates. Over the course of the study all of the categories dropped approximately 3-4 percent except the reference category. The reference category fell only by 1.6 percent.

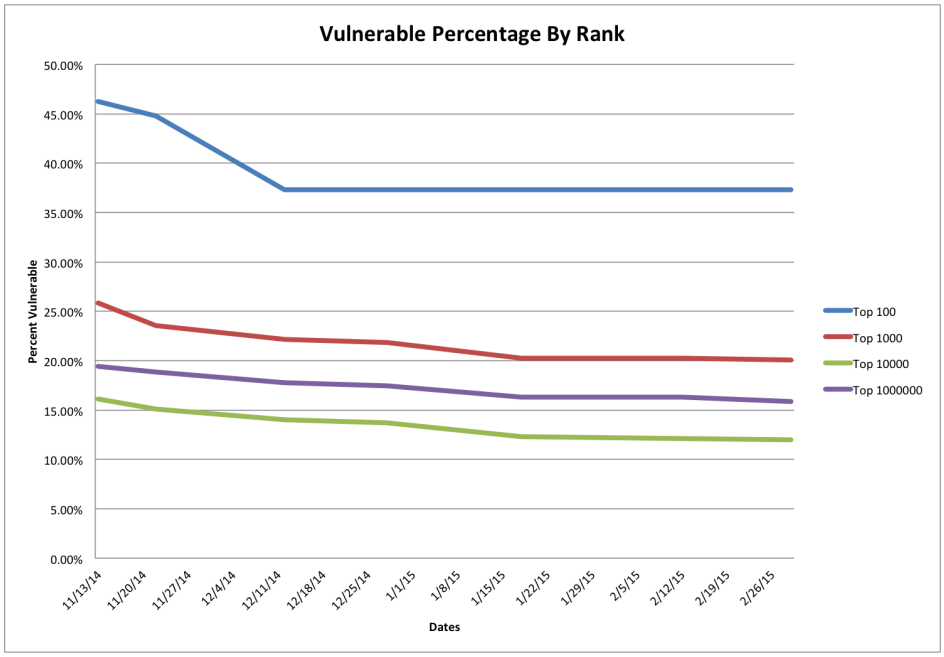


Figure 5.1: Poodle Vulnerability Percentage by Rank

Three trends were revealed when categorizing the sites by port and port status. Figure 5.3 shows the common trend. In this trend, servers with a particular port open exhibit significantly higher vulnerability percentages than servers with the port closed. Additionally, a server with a particular port status of filtered had an even lower vulnerability percentage than servers having the port closed. All 3 port statuses for port 22 had similar patching rates during the study.

Another trend is shown by Figure 5.4. In this trend, servers with a particular port closed have higher vulnerability percentages than servers with the port open. This is opposite of the previous trend. The vulnerability percentage for an open port in this trend was incredibly low. It is important to note that the poodle vulnerability percentage for closed ports were consistent across all ports. This can be seen by looking back at Table 5.3, where all ports with a closed status other than 80 and 443 are within a 10 percent variation.

Our last trend shows servers with an open port with high vulnerability percentages that exhibit a sudden drop off in vulnerability. Figure 5.5 demonstrates the characteristic of our last trend for port 389. Initially, sites with port 389 open have a vulnerability rate of 50%.

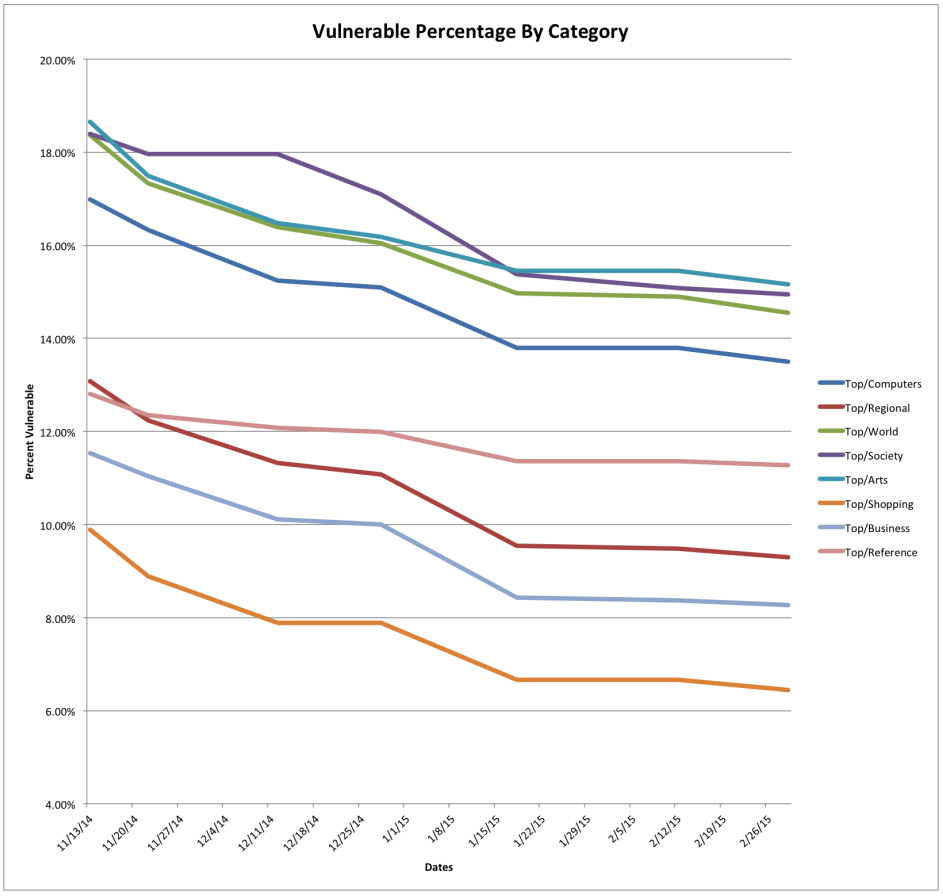


Figure 5.2: Poodle Vulnerability Percentage by Alexa Category

Then, from December 28, 2014 to January 18, 2015, there was a massive vulnerability drop off from 48% to 3.4%.

Using a 5 port security limit, a wide disparity can be seen between exposed and restricted servers. This relationship can be seen in Figure 5.6. Sites that have 5 or less open ports were significantly less vulnerable to the poodle attack than sites that had more than 6 open ports across the entire study.

Aside from identify categorization vulnerability statistics, it is important to identify how the poodle attack was fixed. There were 2 proposed methods to patch against the poodle attack, either disable ssl3 or add the SCSV flag. Table 5.4 shows the patching breakdown for all sites. It can be seen that most of the patched sites disabled SSL3 instead of added the SCSV flag.

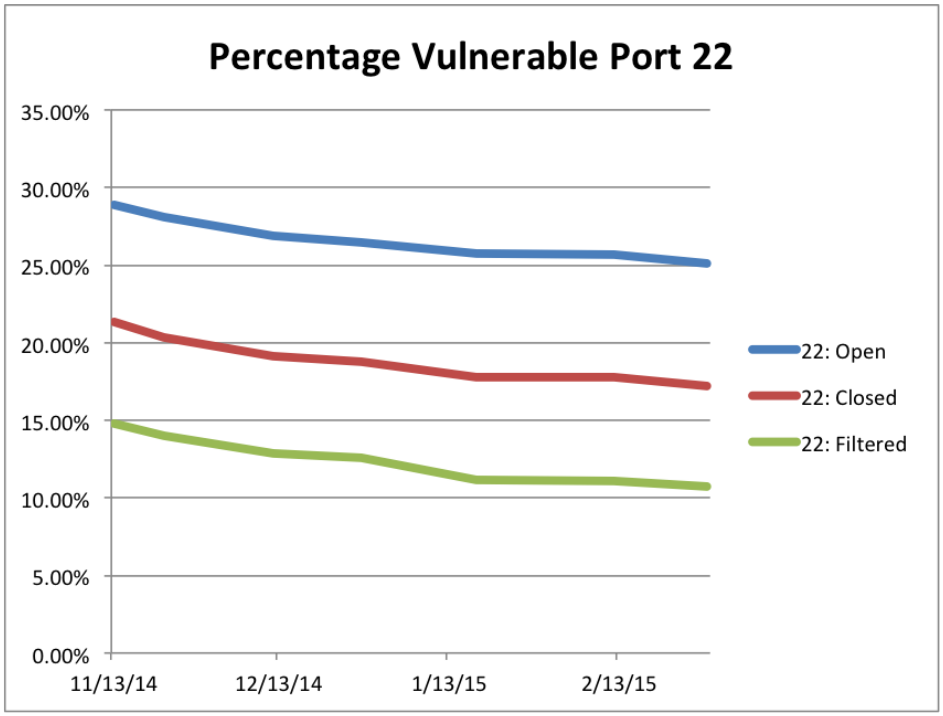


Figure 5.3: Poodle Vulnerability Percentage on Port 22

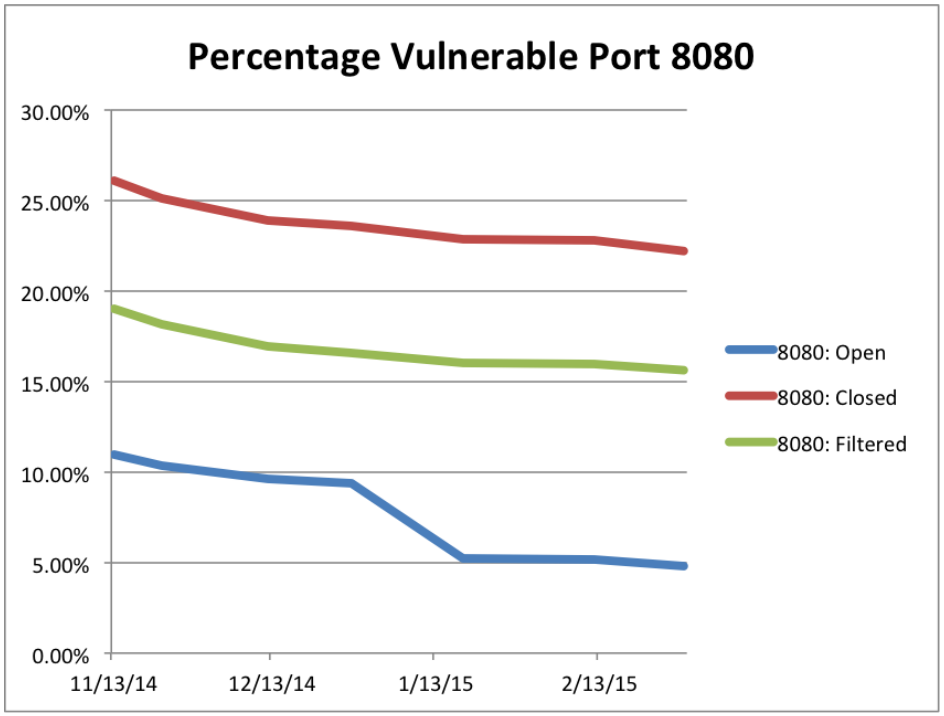


Figure 5.4: Poodle Vulnerability Percentage on Port 8080

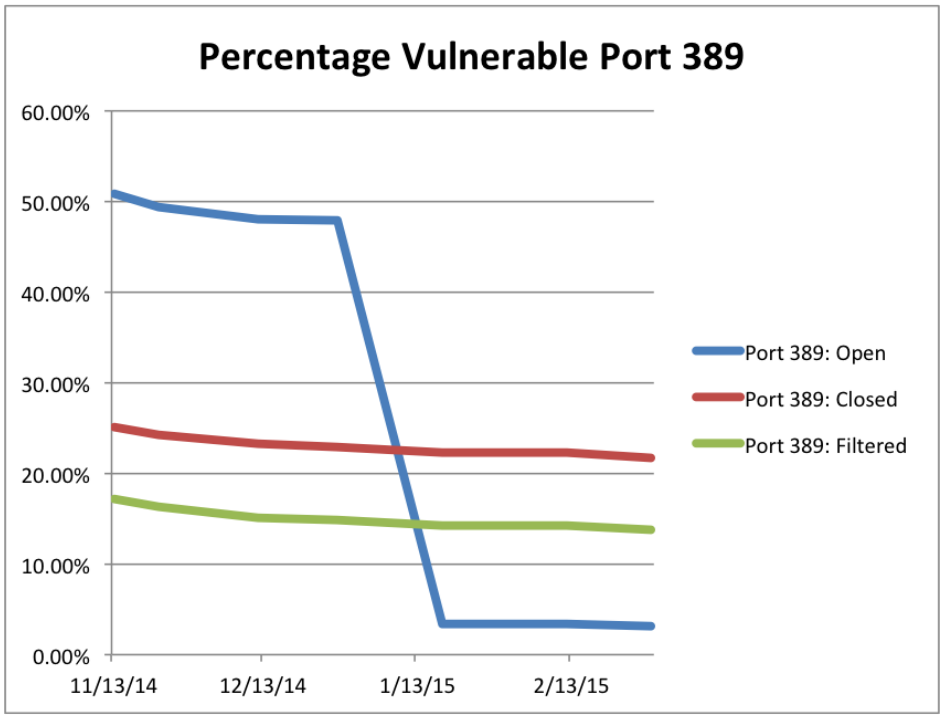


Figure 5.5: Poodle Vulnerability Percentage on Port 389

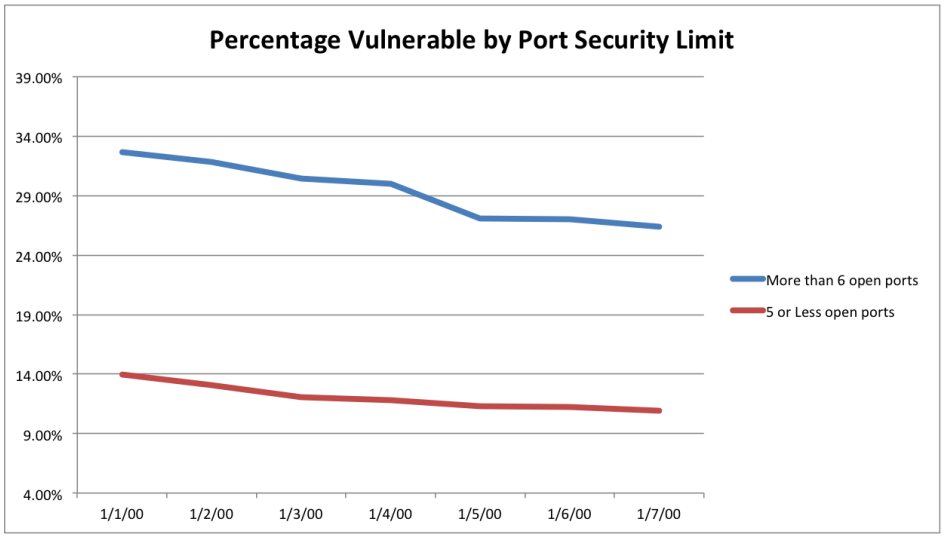


Figure 5.6: Poodle Vulnerability Percentage for Port Security Limit

### 5.1.1 Discussion

There were many different characteristics that affected the vulnerability of a site to the poodle attack. Some of these characteristics include site ranking, site category, and port statuses. This section discusses the results found from the previous section.

| Date | Total Sites Patched | Sites Patched by Disabling SSL3 | Sites Patched by Adding SCSV |
|---|---|---|---|
| 11/13/2014 | 650 | 608 | 62 |
| 11/22/2014 | 497 | 431 | 66 |
| 12/12/2014 | 689 | 633 | 56 |
| 12/28/2014 | 185 | 161 | 24 |
| 1/18/2015 | 706 | 692 | 14 |
| 2/12/2015 | 34 | 28 | 6 |
| 3/1/2015 | 254 | 235 | 19 |

Table 5.4: Overall Poodle Attach Patching Breakdown

The results of the rank category would suggest that the top 100 sites was the most vulnerable rank category in our study. This is a surprising result, as we speculated the top 100 websites to be the most secure category. An important aspect of the web is the client server model. As time progresses, both clients and servers upgrade for the sake of performance and security. As a server upgrades, older clients are rendered obsolete and can no longer connect to the server. Perhaps many of the top 100 sites chose not to patch against poodle in order to maintain compatibility with these older clients.

The sites categorized by Alexa categories produced expected results. Sites categorized by shopping or business were the least vulnerable to the poodle attack. Since shopping and business sites often deal with financial information, security should be very important. Sites that likely dealt with non-sensitive information, such as arts or society, had the highest vulnerability to the poodle attack.

There was one unusual characteristic from the site categorizations. Sites that were not categorized by Alexa had a much higher vulnerability rate than sites that were categorized. Alexa states that it "uses crawling, archiving, categorizing, and data mining techniques to build the Related Links" which is used to ultimately determine a site category [3]. Given that Alexa uses related links to build category lists, sites without a category may not have enough related links. Extending this further, perhaps sites that are unconnected, or sites that haven't linked to or been linked from sites are more vulnerable to sites that are connected.

There were 3 prominent trends within the port category. Figure 5.3 demonstrates the most common trend. While servers can use ports for any purpose, the ports following this trend are commonly used for the following purposes: ftp, ssh, dns, and email protocols. Servers with these ports open follow the average patching rate for all sites.

The port states for the previous mentioned ports exhibit some interesting statistics. It is expected that sites having open ports will be more vulnerable than sites with ports closed. A site having open ports suggests that the server provides additional purposes other than a web server. A server with multiple purposes allows more attack vectors for security threats. Perhaps sites that have a greater number of open ports are less concerned with security, and that explains the reason sites with open ports have a higher vulnerability than a site with a closed port.

Another observation can be made between the filtered and closed port statuses. NMap defines a closed port as being accessible without an accepting application, while a filtered port is a port status that cannot be determined due to firewall interference [8]. Sites that implement a firewall are expected to be more secure than those without firewalls. Our research shows that sites using a firewall, or ports found having the filtered designation, had a lower vulnerability percentage than sites that had closed ports.

For some ports, the vulnerability percentage for the open port was lower than the closed or filtered port. The ports following this trend often had a number of other ports open, although the ports were varied for each server. This could suggest that there are secure applications which use a port that run on many servers. For example, Microsoft SQL Server, which could be considered a widely used and secure application, uses port 1433 which exhibits a lower vulnerability percentage for open ports rather than closed. The last prominent port trend displays a high initial vulnerability percentage for open ports which encounters a dramatic drop off in vulnerability at some time during the study. This trend suggests an application which runs on many of the top 100,000 sites that is initially highly vulnerable to the poodle attack. The drop off in vulnerability percentage indicates that a

high percentage of web servers running a certain application updated all at once, adding protection against the poodle attack for many sites. It is possible that the ports where open ports had a lower poodle vulnerability percentage than the closed ports followed this trend, but the study did not start tracking the poodle attack early enough.

The port security limit demonstrates the most largest disparity between 2 categories. A server which allows a large number of open ports can be considered loosely managed. In other words, when a server administrator chooses to expose many ports on a server, that administrator is not adequately securing the server. Sites which limit the amount of exposed ports follow better security practices and thus are less vulnerable to the poodle attack.

One limitation in our results is that we were unable to track the poodle attack until 18 days after disclosure. It is possible that many servers were already patched by the time our study had began scanning. Even so, there were many trends observed over the course of the study.

## 5.2 Poodle TLS

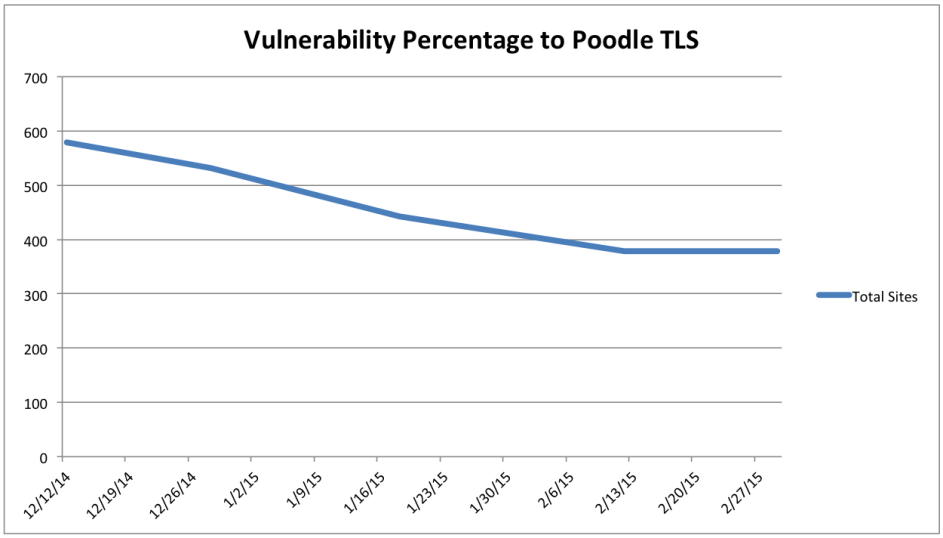


Figure 5.7: Poodle TLS Vulnerable Sites per Date

The Poodle TLS attack was a much narrower vulnerability than the initial poodle attack. We were able to start our scans only 4 days after the disclosure of the bug. Our initial scan

showed a poodle tls vulnerability percentage of less than 1 percent for all sites. Figure 5.7 shows the total number of vulnerable sites over the entire study. Since there was a very limited amount of sites that were vulnerable to the Poodle TLS attack, no correlations or categorizations were found in the study.

### 5.2.1 Discussion

As stated in the background, the Poodle TLS vulnerability was not a protocol vulnerability, but rather a vulnerability in the implementation of the protocol. Months before the disclosure, affected vendors were notified of the vulnerability who likely released necessary patches before the poodle tls announcement. The small vulnerability count in this study may show the effectiveness of predisclosure notifications. Unfortunately, it does not allow us to draw many other findings.

## 5.3 FREAK Attack

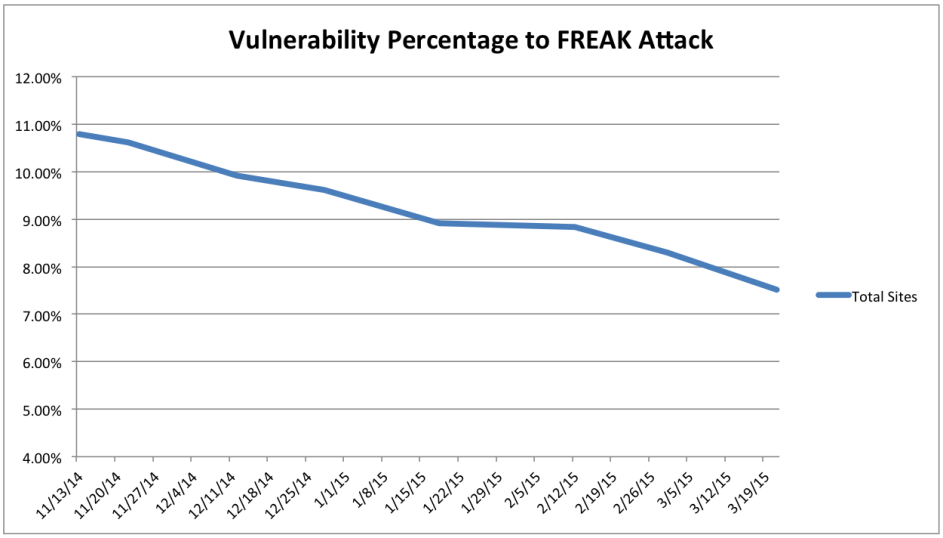


Figure 5.8: FREAK Attack Overall Vulnerability Percentage

This study was able to follow the vulnerability statistics for the FREAK attack since November 11th, 2014. Given that the FREAK Attack was disclosed on March 3rd, 2015, this gives a lot of information of the pre disclosure behavior that was missing from the previous

2 vulnerabilities. The initial results showed that 10.79% of all the sites were vulnerable to the FREAK attack on our first scan. Figure 5.8 shows vulnerability percentage over the course of the study. The patching rate to the FREAK attack was approximately 600 sites per month even before the disclosure date.

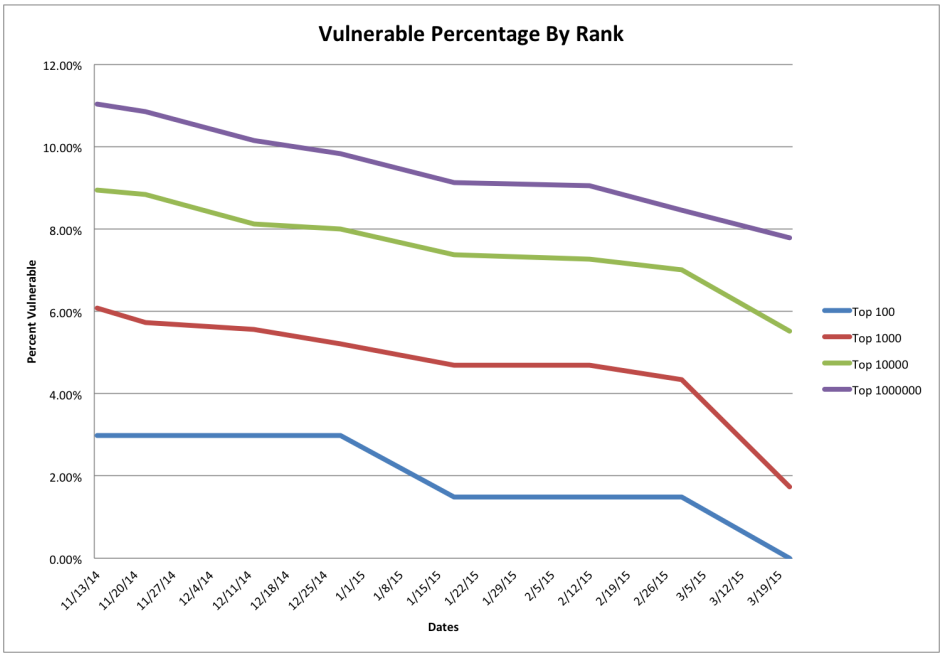


Figure 5.9: FREAK Attack Vulnerability Percentage by Site Rank

Figure 5.9 shows the vulnerability rates to the FREAK attack over the course of the study. The results show the expected result, with higher ranked websites having lower vulnerability rates. The figure shows a steady rate of patching over the course of the study, with a dramatic increase of patching after disclosure for all rank categories except the top 100,000. In fact, the top 100 sites decreased to have 0 vulnerable websites after disclosure.

The vulnerability rates to the FREAK attack for Alexa categorized sites are shown by Figure 5.10. Sites categorized as a shopping site had the lowest vulnerability percentage, having a vulnerability percentage of 4.33% at the end of the study. Correspondingly, sites categorized as either world, reference, or arts had the highest vulnerability percentage, with vulnerability percentages of respectively 10.10%, 9.43%, and 9.43%. The patching

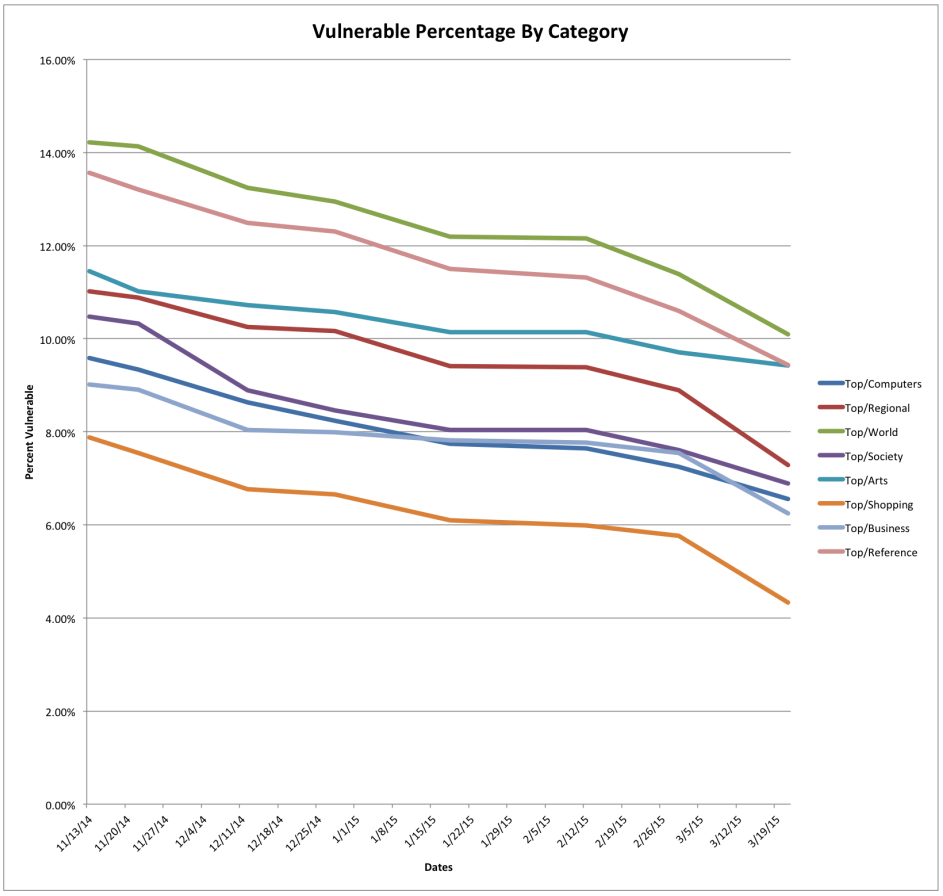


Figure 5.10: FREAK Attack Vulnerability Percentage by Alexa Category

rates match the overall patching rates, with only a few categories increasing patching after disclosure. Those categories were shopping, business, and regional.

There are only 2 trends that can be identified when looking at the FREAK attack vulnerability by port. The first trend is demonstrated by Figure 5.11, where the open port has a higher vulnerability percentage than the closed port. Additionally, the ports that were filtered had a lower vulnerability percentage than both the open and closed port statuses. The ports that followed this trend were the same ones that followed the trend for the POODLE attack.

The second trend can be seen by Figure 5.12. Sites with port 8080 open have a lower vulnerability percentage to the FREAK attack than sites with port 8080 filtered or closed. On our March 31st scan, sites with port 8080 open were 3.47% vulnerable compared with

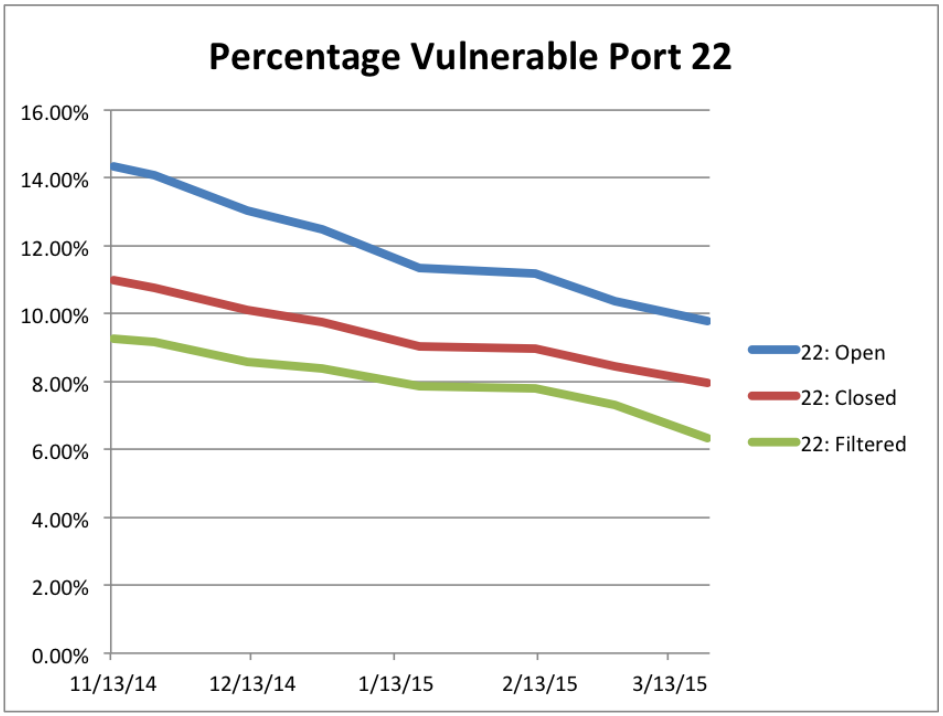


Figure 5.11: FREAK Attack Vulnerability Percentage on Port 22

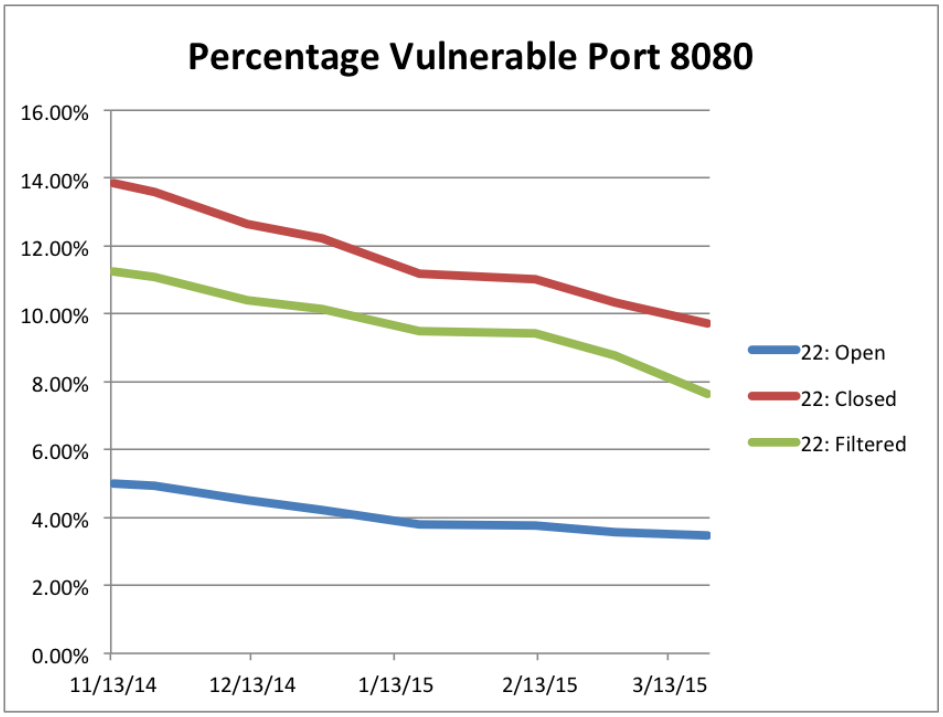


Figure 5.12: FREAK Attack Vulnerability Percentage on Port 8080

9.72% for a closed status and 7.65% for a filtered status. The ports following this trend were the same ports with this characteristic from the POODLE attack.

There was no drop off trend identified with the FREAK attack that was found with the POODLE attack. All ports that had a large drop off in vulnerability percentage with the POODLE attack followed the previous trend mentioned with the FREAK attack.

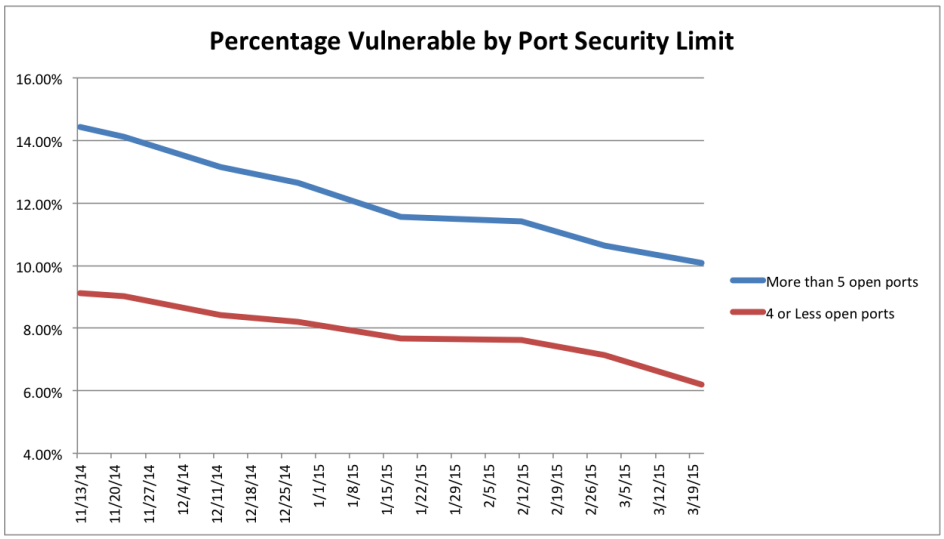


Figure 5.13: FREAK Attack Vulnerability Percentage for Port Security Limit

The same port security concept from the POODLE attack was applied to the FREAK attack. A port security limit of 4 was found to minimize the vulnerability percentage to the FREAK attack. Sites found with 5 or more open ports were found to have a higher vulnerability percentage. Figure 5.13 shows the vulnerability percentages of the FREAK attack for sites matching the port security limit compared to sites that are over the limit over the course of the study. The figure shows a large gap in vulnerability percentage between sites that adhere to the port security against sites that do not. While this disparity is present, there was no difference in patching rates found.

### 5.3.1 Discussion

The results show that sites have been removing export cipher suites at a steady rate during the entire study. This could be caused by 2 different reasons. Sites would patch

early if there was a pre disclosure notification detailing the vulnerability. On the other hand, export cipher suites have been known to be less secure. In order to increase security, sites could have preemptively removed export cipher suites to avoid a future vulnerability.

The rank categories suggest the expected results. That the most visited sites have the lowest vulnerability percentage, while sites with less traffic are more susceptible to the FREAK attack. While there was an steady rate of removing the export cipher suites for all categories, the top 10,000 sites increased the patching rate after the disclosure. This demonstrates a greater security response in the top 10,000 sites.

Sites categorized by Alexa had different levels of vulnerability. Shopping sites, which deal with financial information, had the lowest vulnerability levels. On the other hand, sites in the world, reference, and arts categories had the highest vulnerability levels. The vulnerability patching levels remained consistent for all categories until the disclosure date. The categories which increased their response after the disclosure indicates categories more concerned with security. These categories were shopping, business, and regional. Unlike the POODLE attack, non-categorized sites had the same vulnerability percentage as the overall vulnerability levels.

Overall, sites that had closed or filtered ports had a lower vulnerability percentage to the FREAK attack as reaffirmed by the port security limit. This would suggest that sites having open ports are less likely to properly secure their servers. Also, sites that had a port filtered always had a lower vulnerability rate than its closed counterpart. Since a filtered port would suggest a site using a firewall, sites using firewalls had less vulnerabilities than sites that did not employ a firewall.

There were also exceptions to the normal trend. There were a number of ports that had a better vulnerability percentage when open rather than closed or filtered. Using information from the POODLE attack, this would suggest a widely used application employed on secure servers.

It is still difficult to decipher the exact reason as to why the FREAK attack has been steadily patched starting months before its disclosure. It would be useful to have information dating even earlier than the study started. Given that few site categories increased their patching rates and that overall patching rates remained the same, the results seem to show that sites have known about the insecurities of export cipher suites and have been steadily removing them.

### 5.4 Survey

We received fewer responses than expected to the survey. To elicit more responses, more surveys were sent out directly to people known to manage web servers. The cumulative result from the survey sent using a random sampling of the top websites and the survey sent directly to people are shown in this section.

The first 5 questions ask general information about server configuration. All the respondents used self-hosted servers. Most of the servers use a Linux Operating System, while several others used Windows. When asked to classify data, there were equal amounts of data sensitive, data insensitive, and business confidential with most sites having multiple data classifications.

Most of the respondents responded as never hearing of the poodle attack. This is a surprisingly high amount given the scope of the vulnerability. When asked "how do you stay updated about new vulnerabilities," the respondents that have not heard about poodle mostly stated mailing lists. The respondents that have heard about poodle named a variety of websites as their main source.

When asked about the patching characteristics, many of the respondents were found to patch at most once a month. Less than half of the respondents patched at least once a month. When asked for reasons to avoid patching, several reasons were mentioned. The top reasons for skipping patches included server downtime, compatibility issues, and low risk

patches. When asked about the patching process, most respondents used vendor updating services, and only a few respondents updated on an individual patch basis.

Respondents were then asked more detailed questions about their server configuration. Most respondents stated that they monitored their server for suspicious activity with a large number having automated detection as well. While most of the respondents stated they remotely managed their server, only half used VPN.

### 5.4.1 Discussion

While the survey had low participation, there was some intriguing information received. First, the amount of people who have not heard about the poodle attack was much higher than expected. This could be caused by the source of vulnerability information. People who had not heard of the poodle attack commonly used mail lists to stay up to date with vulnerabilities, while people who had heard of the poodle attack used websites as their main source of information. This would suggest that websites are a better source of information. Some of the websites mentioned were Google News, Hacker News, Twitter, and Kreps on Security.

Another surprising response was the patching frequency. While half of the respondents patched at least once a month, several respondents had patching frequencies of greater than 6 months. This leaves a large window of vulnerabilities for those servers. The main reason people avoided patching was compatibility. Respondents were concerned with system compatibility and 3rd party application compatibility. This could suggest that compatibility is a greater priority than security for server administrators.

Even though most server administrators listed compatibility as a main reason for avoiding patches, a very small number of respondents mentioned having a test server. Perhaps if more server administrators had a test server, then patching frequency could increase due to alleviated concerns about compatibility. Another common concern was server downtime due to patching. Respondents mentioned prolonging any updates that required any downtime

at all. If patches could be designed to avoid any server downtime, patching frequency could improve.

Most of the respondents had a method to remotely manage a server. Given the convenience and sometimes requirement of remote access, this result was expected. However; only half of the these respondents utilize a VPN. This indicates that the web servers had extra ports open to the Internet and could suggest a less secure server compared to servers which utilize a VPN.

### 5.4.2 Comprehensive Discussion

In this section, we discuss the relations between the 3 previous vulnerabilities. In particular, we identify common traits and characteristics that can protect or harm the security of servers to SSL/TLS vulnerabilities. Lastly, we connect the results of our survey concerning these vulnerabilities.

Only 25% of the servers vulnerable to the FREAK attack were vulnerable to the original POODLE attack. This highlights how difficult it can be for server administrators to correctly configure a server for security. While a site may be secure against one vulnerability, it may be vulnerable to another one.

The biggest indicator of server vulnerability to both attacks was the port security limit. Having 5 or more open ports increased the vulnerability percentage for both the POODLE and FREAK attacks.

Since the Poodle TLS vulnerability is an implementation issue, this study does not consider any server configuration able to prevent the vulnerability at the time of attack. While server administrators are freed of blame, they rely completely on the software vendors to provide a fix in adequate time.

The survey revealed a serious lack of web vulnerability dissemination. Over half of the respondents had not even heard of the Poodle attack. This shortcoming could explain why patching rates are not higher. Additionally, server administrators are under serious pressure

to maintain compatibility and server up time. When faced with a vulnerability, these needs can take higher precedence.

Chapter 6

Conclusions and Future Work

### 6.0.3 Conclusions

This research had two goals. The first goal was to identify how the web reacts and responds to known vulnerabilities. The research presented in Chapter 5 uncovers the different ways the web reacts, with the majority reacting slowly to vulnerabilities. Fulfilling our second goal, the research finds differing characteristics between secure and insecure servers.

Our research indicates that while server administrators are constantly improving server security, web servers are facing a steady attack of evolving web vulnerabilities. The SSL/TLS protocol was found vulnerable in three different ways within the span of four months. Additionally our research indicates that server administrators are under pressure to maintain compatibility and server up time in ways that could conflict with measures to secure a server.

### 6.0.4 Future Work

Since our research provided promising results from four months of study, a longer study would be useful to explore more vulnerabilities as well as identifying a longer time period to track single vulnerabilities. Additionally our research could be expanded outside of SSL/TLS protocols to provide a broader picture of Internet vulnerabilities.

Bibliography


[1] Paul Adamczyk, Munawar Hafiz, and Ralph E. Johnson. *Non-compliant and Proud: A Case Study of HTTP Compliance*. UIUCDCS-R-2008-2935. URL: http://hdl.handle.net/2142/11424.

[2] Roy Fielding et al. *Hypertext Transfer Protocol – HTTP/1.1*. June 1999. URL: https://tools.ietf.org/html/rfc2616.

[3] *Alexa Web Information Service FAQs*. URL: http://aws.amazon.com/awis/faqs/#URL_5.

[4] *AWS — Alexa Top Sites - Up-to-date lists of the top sites on the web*. URL: http://aws.amazon.com/alexa-top-sites/.

[5] Benjamin Beurdouche et al. *SMACK: State Machine AttaCKs*. Mar. 2015. URL: https://www.smacktls.com/.

[6] Matt Bishop. *Introduction to Computer Security*. 1st ed. Addison-Wesley Professional, Nov. 2004. ISBN: 0321247442.

[7] Scott Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. Mar. 1997. URL: https://www.ietf.org/rfc/rfc2119.txt.

[8] *Chapter 15. Nmap Reference Guide*. URL: http://nmap.org/book/man-port-scanning-basics.html.

[9] T. Dierks and C. Allen. *The TLS Protocol*. URL: https://www.ietf.org/rfc/rfc2246.txt.

[10] Zakir Durumeric et al. "The Matter of Heartbleed". In: *Proceedings of the 2014 Conference on Internet Measurement Conference*. IMC '14. Vancouver, BC, Canada: ACM, 2014, pp. 475–488. ISBN: 978-1-4503-3213-2. DOI: 10.1145/2663716.2663755. URL: http://doi.acm.org/10.1145/2663716.2663755.

[11] *IBM Security Services 2014 Cyber Security Intelligence Index*. URL: http://www-935.ibm.com/services/us/en/it-services/security-services/2014-cyber-security-intelligence-index-infographic/.

[12] *Implications of the IBM Global Study on the Economic Impact of IT Risk*. URL: http://www-935.ibm.com/services/us/gbs/bus/html/risk_study.html.

[13] SSL Labs. URL: https://www.trustworthyinternet.org/ssl-pulse/.

[14] Adam Langley. *POODLE attacks on SSLv3*. Oct. 2014. URL: https://www.imperialviolet.org/2014/10/14/poodle.html.

[15] Adam Langley. *The POODLE bites again*. Dec. 2014. URL: https://www.imperialviolet.org/2014/12/08/poodleagain.html.

[16] Bodo Mller, Thai Duong, and Krzysztof Kotowicz. *This POODLE Bites: Exploiting The SSL 3.0 Fallback*. URL: https://www.openssl.org/~bodo/ssl-poodle.pdf.

[17] *OpenSSL: About, General*. URL: https://www.openssl.org/about/.

[18] Ivan Ristic. Dec. 2014. URL: https://www.ssllabs.com/downloads/SSL_TLS_Deployment_Best_Practices.pdf.

[19] Ivan Ristic. *Poodle Bites TLS*. Dec. 2014. URL: https://community.qualys.com/blogs/securitylabs/2014/12/08/poodle-bites-tls.

[20] *SSL/TLS in Detail*. URL: https://technet.microsoft.com/en-us/library/cc785811\%28v=ws.10\%29.aspx.

[21] *The Secure Sockets Layer and Transport Layer Security*. URL: http://www.ibm.com/developerworks/library/ws-ssl-security/.

[22] *The World in 2009: ICT Facts and Figures*. URL: http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2009.pdf.

[23] *The World in 2014: ICT Facts and Figures*. URL: http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf.

Appendices

Appendix A

Initial Scanning Code

```
echo "##BEGIN DOMAIN TESTS $1"
echo "#SSLv3 TEST $1"
./openssl s_client -connect $1:443 -state -ssl3 < au-ssl-command
echo "#SSLv3+SVSC $1"
./openssl s_client -connect $1:443 -state -fallback_scsv -ssl3 < au-ssl-command
echo "#SSLv2 $1"
./openssl s_client -connect $1:443 -state -ssl2 < au-ssl-command
echo "#TLS1 $1"
./openssl s_client -connect $1:443 -state -tls1 < au-ssl-command
echo "#TLS1.1 $1"
./openssl s_client -connect $1:443 -state -tls1_1 < au-ssl-command
echo "#TLS1.2 $1"
./openssl s_client -connect $1:443 -state -tls1_2 < au-ssl-command
echo "#cipherscan"
./cipherscan $1
echo "##END DOMAIN TESTS $1"
```

Appendix B

Final Scanning Code

```
echo "##BEGIN DOMAIN TESTS $1"
echo "#SSLv3 TEST $1"
./openssl s_client -connect $1:443 -state -ssl3 < au-ssl-command
echo "#SSLv3+SVSC $1"
./openssl s_client -connect $1:443 -state -fallback_scsv -ssl3 < au-ssl-command
echo "#SSLv2 $1"
./openssl s_client -connect $1:443 -state -ssl2 < au-ssl-command
echo "#TLS1 $1"
./openssl s_client -connect $1:443 -state -tls1 < au-ssl-command
echo "#TLS1.1 $1"
./openssl s_client -connect $1:443 -state -tls1_1 < au-ssl-command
echo "#TLS1.2 $1"
./openssl s_client -connect $1:443 -state -tls1_2 < au-ssl-command
echo "#cipherscan"
./cipherscan $1
echo "#tlsprobe"
python2.7 tlsReport.py $1
echo "#portScan"
./nmap-scan.sh $1
echo "##END DOMAIN TESTS $1"
```