SURVEY OF UNITED STATES RELATED DOMAINS: SECURE NETWORK PROTOCOL ANALYSIS

DeJean Dunbar, Patrick Hill, and Yu-Ju Lin

Department of Computer Science,
Charleston Southern University, North Charleston, South Carolina

ABSTRACT

Over time, the HTTP Protocol has undergone significant evolution. HTTP was the internet's foundation for data communication. When network security threats became prevalent, HTTPS became a widely accepted technology for assisting in a domain defense. HTTPS supported two security protocols: secure socket layer (SSL) and transport layer security (TLS). Additionally, the HTTP Strict Transport Security (HSTS) protocol was included to strengthen the HTTPS protocol. Numerous cyber-attacks occurred in the United States, and many of these attacks could have been avoided simply by implementing domains with the most up-to-date HTTP security mechanisms. This study seeks to accomplish two objectives: 1. Determine the degree to which US-related domains are configured optimally for HTTP security protocol setup; 2. Create a generic scoring system for a domain's network security based on the following factors: SSL version, TLS version, and presence of HSTS to easily determine where a domain stands. We found through our analysis and scoring system incorporation that US-related domains showed a positive trend for secure network protocol setup, but there is still room for improvement. In order to safeguard unwanted cyber-attacks, current HTTPS domains need to be extensively investigated to identify if they possess lower version protocol support. Due to the infrequent occurrence of HSTS in the evaluated domains, the computer science community necessitates further HSTS education.

KEYWORDS

Network Protocols, HTTP Strict Transport Security, scoring benchmark, domain analysis, survey

1. INTRODUCTION

HTTP was a pinnacle basis in defining how information was transmitted across a network during this technical era of information technology. Since 1990, the world wide web has employed the HTTP Protocol as a stateless application-level protocol for hypermedia information systems. HTTP/0.9, the initial implementation of HTTP, was designed for raw data delivery across the Internet. As new versions of HTTP were released, no security procedures for the transport of raw data were implemented [1].Hackers can access health information, government information, and personal information. HTTPS protocol was introduced to address this significant security vulnerability.

HTTPS is a direct extension of the HTTP Protocol introduced by Netscape Communications. There were several security implementation versions of the HTTPS protocol which were SSL and TLS. SSL was first proposed in the middle of 1994 by Netscape Communications with its highest version implementation being version 3. TLS was proposed by the Internet Engineering Task Force with its highest version being version 1.3. The overall goal of HTTPS was to provide security features for HTTP such as encipherment, digital signature mechanisms, data integrity, authentication exchange mechanisms, and notarization mechanisms [2]. With these additional

security measures incorporated, another web security standard extension for HTTPS was introduced known as HTTP Strict Transport Security (HSTS).

The Internet Engineering task force (IEFT) proposed HSTS in 2012 and defined it as a security mechanism that restricts website access to only secure connections. This feature guard against bootstrap man in the middle (MITM) attacks. Additionally, HSTS provides security by converting a URI reference to a secure URI reference [3]. With these additional security benefits that can be added to HTTPS, the HSTS security mechanism helps a domain's network security strength to be even stronger.

This research is structured with related works in section two which discusses past research efforts in domain analysing and scoring benchmarks. Section three presents a preliminary finding consisting of the hardware involved, techniques for gathering the US related domains, the database used, the scanner used for domain protocol information and python parsers used. Section four addresses the research plan which includes considerations made in the research gathering, the proposed scoring system mechanics, research results and limitations. Finally, section five ends with the conclusions based off the research results.

2. RELATED WORKS

2.1. Past Domain Analysing

This segment showcases past research effort in the category of domain scanning. The work summarizes and provides an explanation of how the it contributes to the overall research in this paper.

Patrick Hill’s and Yu-Ju Lin's research establishes the analyzing of government websites in the United States due to the trustworthiness of state and local government websites being questionable. They concluded that there are instances of government websites that are not as secure as they should be against cyber-attacks [12]. However, the methodology used to collect the “.gov” domains in this work included an exhaustive search through Google for the top ten US counties, which may have left out several domains. To create a more accurate assessment, this research will use a defined list of all “.gov” domains by obtaining a complete list from the registrar DOTGOV website. Patrick’s Hill’s and Yu-Ju Lin’s research impacts this current research because it prevented a very tedious and unreliable way of gathering “.gov” domains. Another impactful way their research connects with my current research is that the inspiration of analyzing other United States based entities came from their thorough and detailed focus on “.gov” domains.

2.2. Past Scoring Benchmark

It is necessary to review prior research that has been conducted on scoring systems. The goal of this section is to give a brief overview of the related entity along with providing how our research makes a contribution.

SSL Labs is a non-profit research organization that maintains a comprehensive collection of SSL/TLS documentation, tools, and a community [11]. Their website is capable of scanning domains for SSL configurations and providing a score. The overall SSL/TLS strength score for a domain is divided into three categories: protocol support, key exchange, and cipher strength. SSL Labs' current shortcoming is that they have not included TLS 1.3 in their rating guide. Our research will contribute by incorporating a scoring system that incorporates TLS 1.3 as well as

determining whether the domain supports HSTS. This related work made an impact on my research because it gave a general concept of how to determine scoring for protocol versions which in turn steered me into my proposed scoring benchmark's focus on protocol support being the factor of scoring.

Previous research in domain scanning and scoring benchmarks paved the way for the success of this study. Both works established a foundation for domain gathering techniques and scoring benchmarks. However, some preliminary work isrequired before these concepts can be implemented and revealed in this research.

3. PRELIMINARY FINDING

This section of the study goes through the hardware used in the experiment. The origins of domains related with the United States is also highlighted. Next the database housing the domains is revealed. The domain results are then addressed. Following that, the scanning procedure used to detect the domain's SSL/TLS version is focused on . Finally, the numerous Python scripts used to help with domain analysis and database updates is described .

3.1. Hardware Used

All the tools used for this research were all run in a virtual machine using Ubuntu. The computer model is an Inspiron 16 7610 which operates on a Windows 11 Pro x64 operating system. The computer has an i7 core and 32GB RAM. It was necessary to utilize a higher core to better utilize throughput for running multiple instances of the scanning program and the python scripts

3.2. Techniques for Gathering US Based Domains

For this study, domains ending in ".us", ".gov", and ".edu" were grouped together. The number of ".us" domains gathered were 1,814,204. The number of "gov" domains gathered were 5,854. The number of ".gov" domains gathered were 7,671. The grand total of domains gathered were 1,827,729. In order to study how the domain groups were gathered, we first focus on the technique for gathering ".us" domains.

To find the source of ".us" domains, the first step was to determine what entity had access to the zone file. A".us" zone file request was sent to the registry site ABOUT and was later redirected to GODADDY[4]. A zone file contains a list of all domains that have been registered. Following the approval of the request, the zone file was provided. A new zone file is created every day with the year, month, and day due to new/existing domains being updated. The zone file we chose for parsing was from March 26,2022. Although domain names were included in the zone file, the file contained other information deemed unimportant. On aDELL laptop running Ubuntu virtually, a series of commands shown in Figure 1 was executed on an Ubuntu command prompt to extract only the domain names from the zone file. The extracted data was saved as text files.

```
$ awk '{print $1}' us.zone > domains-only.txt
$ sort- u domains-only.txt --output domains-unique.txt
$ LC_ALL=C grep '^[A-Z0-9\-]*$' domains-unique.txt > domains.txt
```

Figure 1. List of commands used to parse through ".us" zone file

Thesame technique was attempted for ".edu" domains, however the organization EDUCAUSE's cooperative agreement rules would not allow them to give us access to their zone file. To accommodate the lack of a zone file, we decided to utilize two outside sources for the gathering of ".edu" domains. The first source was from a GitHub repository which provided ".edu" domains of universities from around the world [5]. The list was filtered to only the United States. The next source of ".edu" domains came from Common Crawl, a reputable web crawling service [6]. Common Crawl provided a server for their data to be queried via Amazon's AWS service, Athena [7]. A query was run against Athena to collect domain names ending in ".edu" in the year 2021.

For gathering ".gov" domains there is an actual government site that list all the currently registered government websites in the United States [8]. The list of .gov domains gathered were stored in a csv file.

3.3. Database Used

We used the MYSQL database to maintain a consistent repository for the domain information used in this research. This database stores domain names, SSL, and TLS versions, HTTP status, HTTPS status, and HSTS status to aid in the analysis results section. MYSQL Workbench, a database application, was used to import all the domains that were gathered and stored as csv files in the previous section into the MYSQL database. On an ethical note, to note, the database is a local database housed within the virtual Ubuntu to protect the confidentiality of domain results

3.4. Scanner for SSL/TLS Identification

We invoked a well-documented tool called SSLSCAN [9] to scan the US-related domains for SSL/TLS protocol versions. This command-line tool accepts a file of domain names as input and returns in XML format the SSL/TLS protocol versions for each domain, if any. The domain names were obtained using a query against the MYSQL database and then converted to csv files to serve as the input for the SSLSCAN tool. Following that, ten instances of the SSLSCAN with the csv files were run to pipeline the scanning process. The total time to scan was thirteen days.

3.5. Python Parsers

To transfer data from SSLSCAN's XML output files to a MYSQL database, a Python application parsing the XML file output was written. An update statement within the script was executed for each domain parsed to keep the database in sync with the SSL/TLS protocol information from the XML files.

To determine whether HSTS was present in the domains collected, another Python script was written to request the domain's header information. This script took as input a csv of domain names. We queried the MYSQL database for HTTPS domain names that had successful scans with the SSLCANNER. As with the previous Python script, this one updated the MYSQL database in response to the presence of HSTS for each domain. The program execution took five days to process the domains.

By describing the methodology used for gathering domains a clear roadmap is established in this research. Additionally, describing the various tools for domain scanning and storage of results presents valuable insight into the construction of the research plan.

4. METHODS (RESEARCH PLAN)

4.1. Considerations

Before going over the results of the domains it is necessary to outline some decisions made before and during the analysis. There were originally 1,839,452 domains gathered and transferred to the MYSQL database. 11,723 of the domains did not have the correct extension stemming from the ".us" zone file provided. For example, there was a domain named "100plusus". It was ambiguous if the name should have been "100plusus.us" or "100plus.us". These 11,723 domains were removed from the database to remove this ambiguity. Another consideration made was during the SSLSCANNER application being ran on the domains. There were errors logged in the XML file for each domain that encountered an issue. These issues ranged mainly from refused connections from the domains or timeouts. The total number of usable domains after the scanning was 658,500. Due to the nature of scanning domains, we ensured that the results are only stored in a private repository in GitHub to ensure best ethical practices.

4.2. Proposed Scoring System

The proposed scoring system is described in this section and takes into account the protocol version and whether HSTS is being used. Additionally, this section presents some examples to properly illustrate the scoring system in practice with given domain configurations.

The scoring system grading is in the numerical range from 0 – 100. We consider a score of 70 to be passing while anything lower is a failure. One assumption made for this scoring system is that if a domain supports multiple HTTPS protocol versions, then the highest HTTPS protocol version is only considered for the domain's overall score.

The proposed scoring system is split into three tiers. Tier 1 includes domains that support HTTP protocol. These domains automatically receive a score of 0 due to no security being available for the protocol. Tier 2 consists of the HTTPS protocol with the SSL version variations which includes SSL 2.0 and SSL 3.0. SSL 2.0 is assigned a starting score of 5. The path for potential updates can be described as seen in Figure 2 where each transition to the next state is awarded 5 points. This pattern continues until the highest SSL protocol version with HSTS is reached which is awarded 20 points. Tier 3 has the same principle as Tier 2 using TLS version variations, but TLS protocol 1.0 starts off with 30 points. Each transition to the next state for TLS versions is awarded 10 points.

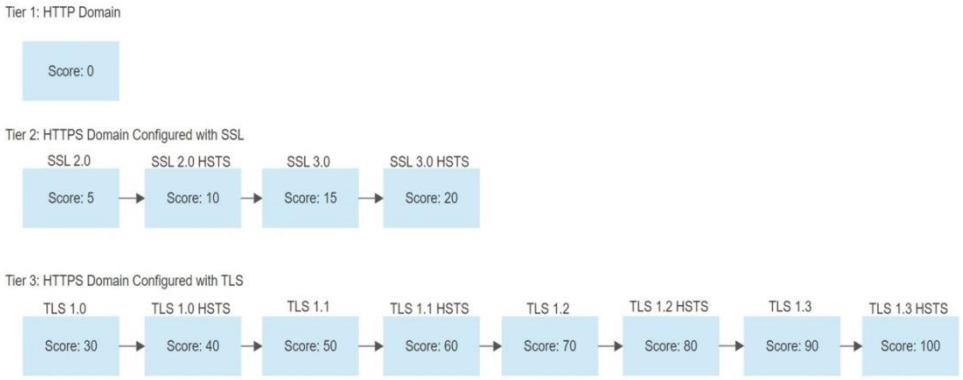

Figure 2. Three tier process for proposed scoring system

A passing score for this scoring system is when TLS 1.2 protocol is used which is awarded 70 points. The reason for this decision is due to RFC officially announcing the deprecation of protocols SSL 2.0, SSL 3.0, TLS 1.0 and TLS 1.1. The best-case scenario is that a domain contains TLS 1.3 HSTS which isa score of 100.

We use three examples to illustrate the scoring system. Example one is relatively simple with Figure 3 having the current configuration of just HTTP and because of this we give a failing score of 0. Example two in Figure 4 has SSL 3.0 with HSTS. Since we are in Tier 2 category, we start off with a base score of 5. Since we transitioned three states in order to reach SSL 3.0 with HSTS we add an additional 5 points per state which leads to a total score of 20 points. Example three shows Figure 5 with the current configuration of TLS 1.1 with HSTS. Notice that this figure shows the domain supporting earlier protocol versions. We ignore the earlier protocol versions and only consider the highest. Since we are in the Tier 3 category, we start off with a base score of 30. Since we have transitioned three states in order to reach TLS 1.1 with HSTS, we add an additional 10 points per state leading to a total score of 60 points.

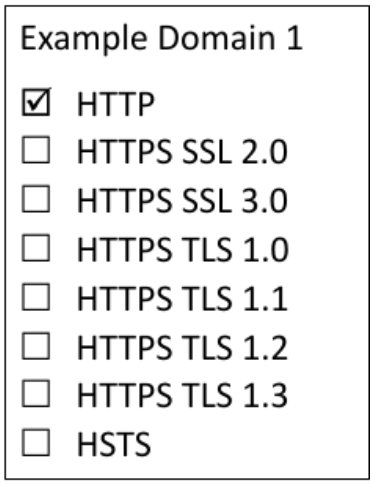

Figure 3. Scoring system example domain with HTTP configuration

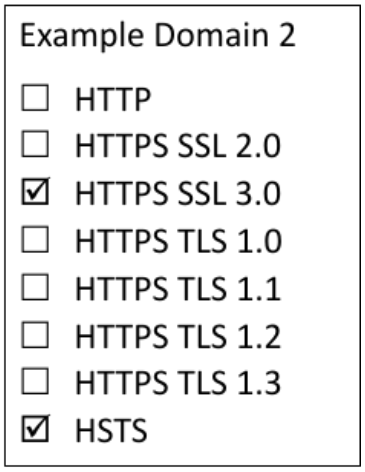

Figure 4. Scoring system example domain with HTTPS SSL 3.0 and HSTS header

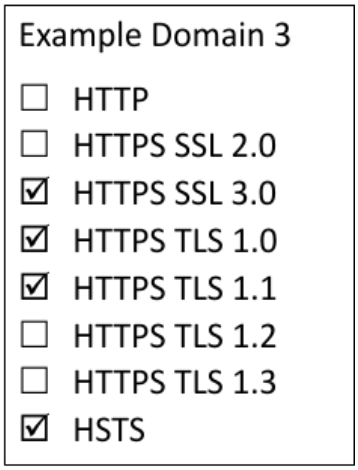

Figure 5. Scoring system example domain with HTTPS SSL 3.0, TLS 1.0, TLS 1.1, and TLS 1.2 and HSTS header.

4.3. Experiment Results/Analysis

4.3.1. Overall Domains: HTTP vs HTTPS

Of the 658,500 domains, 38% had only HTTP protocol support while 62% had only HTTPS protocol support shown in Figure 6. The noticeable percentage of US-related domains being HTTP even in modern times is alarming. A possible explanation is that the nature of the domains that contained HTTP protocol does not transmit sensitive information over the network at all which asserts that there is no need for HTTPS. This assumption is later disproven when a random sample of the analysed HTTP related websites were chosen for investigation. We found there were several instances of .edu domains that were HTTP containing login features which is not good practice.

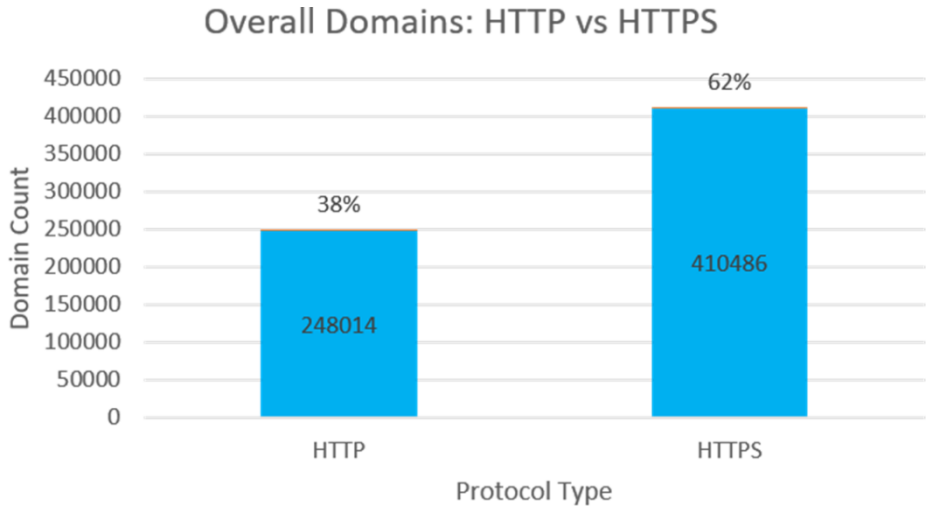

Figure 6. A bar graph visualizing the number of domains scanned that were HTTP protocol or HTTPS protocol.

4.3.2. Group Specific Domains: HTTP vs HTTPS

We compare domain groupings using the percentage metric. The domain group ".gov" and ".us" were the two domain groups that used the HTTPS protocol the most and the least, respectively. The domain group ".us" had the largest concentration of HTTP, whilst the domain group ".edu" had the lowest concentration. Since the government is involved and has specialized resources and infrastructure, initially it was believed that the ".gov" domain would have more HTTPS sites, but this was proven false as seen in Figure 7.

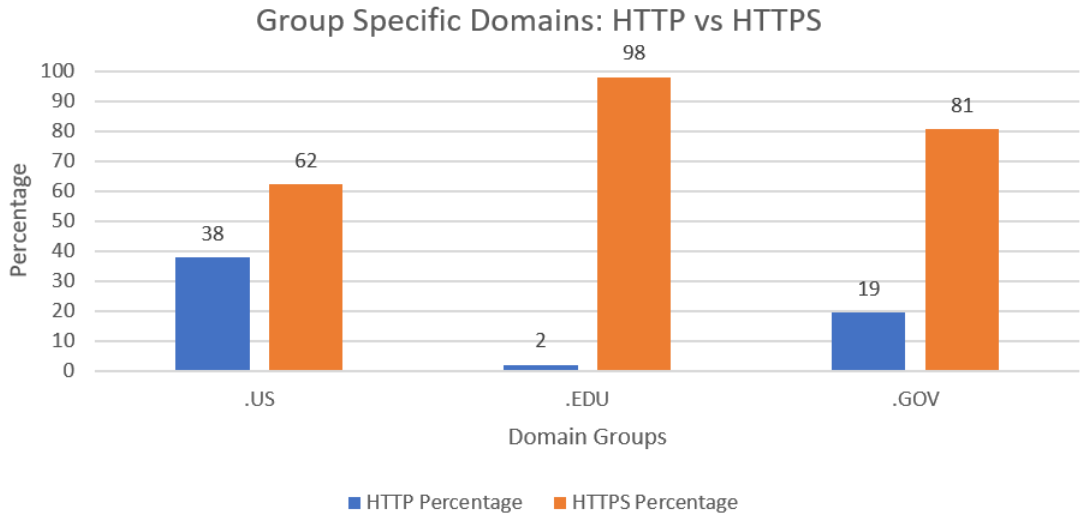

Figure 7. A bar graph visualizing the occurrence of domains by group scanned that were HTTP protocol or HTTPS protocol.

4.3.3. Overall Domains: HTTPS Protocol Type

We next further split the HTTPS into its components SSL2.0, SSL3.0, TLS 1.0, TLS 1.1, TLS 1.2, and TLS 1.3 for analysis. It is important to note that domains can have more than one version of HTTPS enabled. Of the 658500 domains analysed, under 1% of the domains were configured with SSL 2.0 and SSL 3.0; 23% of the domains contained TLS 1.0; 24% contained TLS 1.1; 62% contained TLS 1.2; 30% contained TLS 1.3. (See Figure 8)

When analysing the HTTPS protocol versions, the SSL version 2.0 served as the minimum for the number of domains. This met expectations due to it having been deprecated since 2011 by RFC 6176. Additionally, SSL 2.0 was released over two decades ago with vulnerabilities present in them that would create a high need to transition to the TLS protocol. [10]

When analysing the HTTPS protocol versions, TLS 1.2 served as the maximum for the number of domains. This trend is furthermore supported by Qualys SSL Labs. Qualys SSL Lab's past history of domain scans from January 2021 to October 2021 revealed TLS 1.2 served as the maximum for domain usage [11].

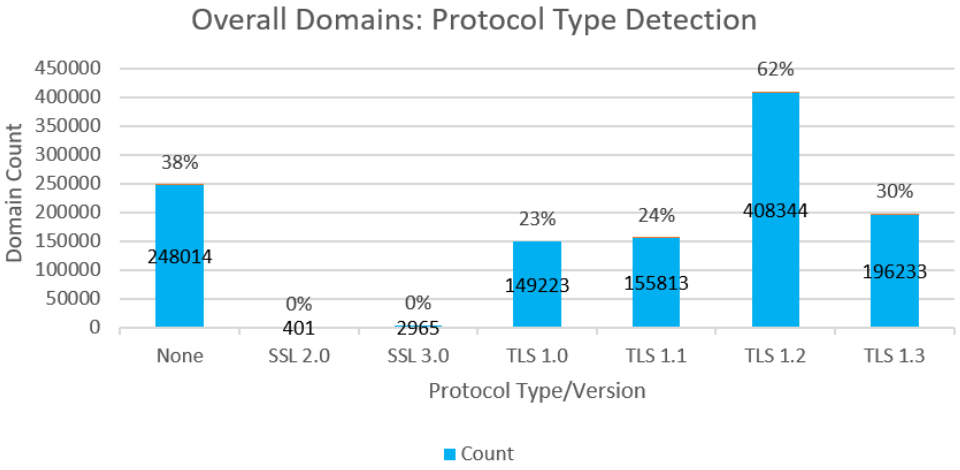

Figure 8. A bar graph visualizing HTTP domain counts along with a more broken-down analysis of HTTPS protocols with their varying versions.

4.3.4. Group Specific Domains: HTTPS Protocol Type

From a domain group standpoint, the ".edu" domain has the highest occurrence of TLS 1.3 and TLS 1.2, followed by ".gov" and ".us." It is worth noting that ".edu" not only had the highest occurrence of HTTPS domains but also the highest occurrence of the most recent protocols, as shown in Figure 9.

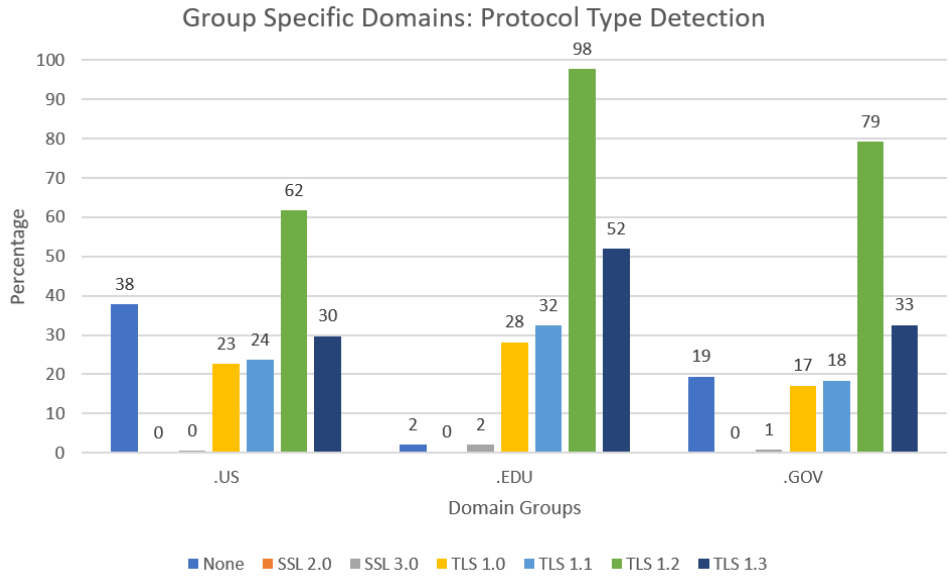

Figure 9. A bar graph visualizing HTTP domain occurrences by domain groups along with a more broken-down analysis of HTTPS protocols with their varying versions.

4.3.5. HSTS Prescence in Overall Domains

Finally, for HSTS detection of the 658,500 domains analyzed, 5% of the domains contained HSTS headers while the other 95% contained no HSTS headers. (See Figure 10). The trend of

the low amount of HSTS being detected is supported by other works conducted in the past. One research focused on government websites had a similar trend in where only 2.86% percent of those websites supported HSTS [12]. Another research paper on HSTS deployment survey conducted in 2013 revealed a similar trend. Of the 1 million websites analysed only 277 contained HSTS headers [13]. Additional research work conducted in 2018 focused on analysing the adoption of security headers in HTTP found that from the 1 million websites scanned that only 5.41% used HSTS [14]. Another research paper conducted in July 2018 focused on analysing HSTS found that from the 1 million websites scanned that only 5.35% used HSTS [15].

We believe the main culprit for why HSTS headers are low in presence is due to users not being educated or informed about HSTS. More importantly, IT professionals or computer scientists are the community of individuals who would configure the HSTS headers for domains. To explore this theory, a survey was conducted among computer science professionals and IT professionals from a Department of Energy owned facility called Savannah River Nuclear Solutions. The results (see Figure 11) found that 80% of the surveyed individuals did not know what HSTS is.

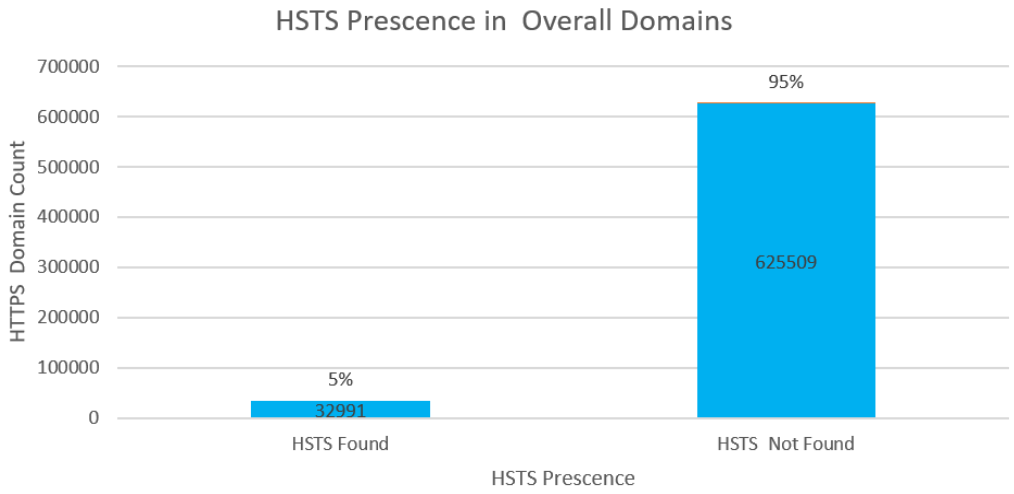

Figure 10. A bar graph visualizing the number of HSTS headers detected versus the number not detected for the scanned domains

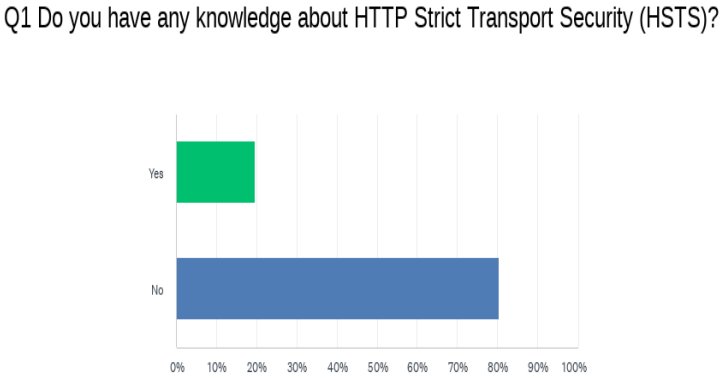

Figure 11. A bar graph from Survey Monkey visualizing the number of participant's knowledge of HSTS.

4.3.6. HSTS Prescence in Group Specific Domains

Although ".edu" domains had the highest frequency of HTTPS, TLS 1.2, and TLS 1.3, Figure 12 shows that the ".gov" domain had the highest frequency of HSTS header presence. In contrast to the other groups, ".gov" had the highest frequency of domains that did not have HSTS presence.

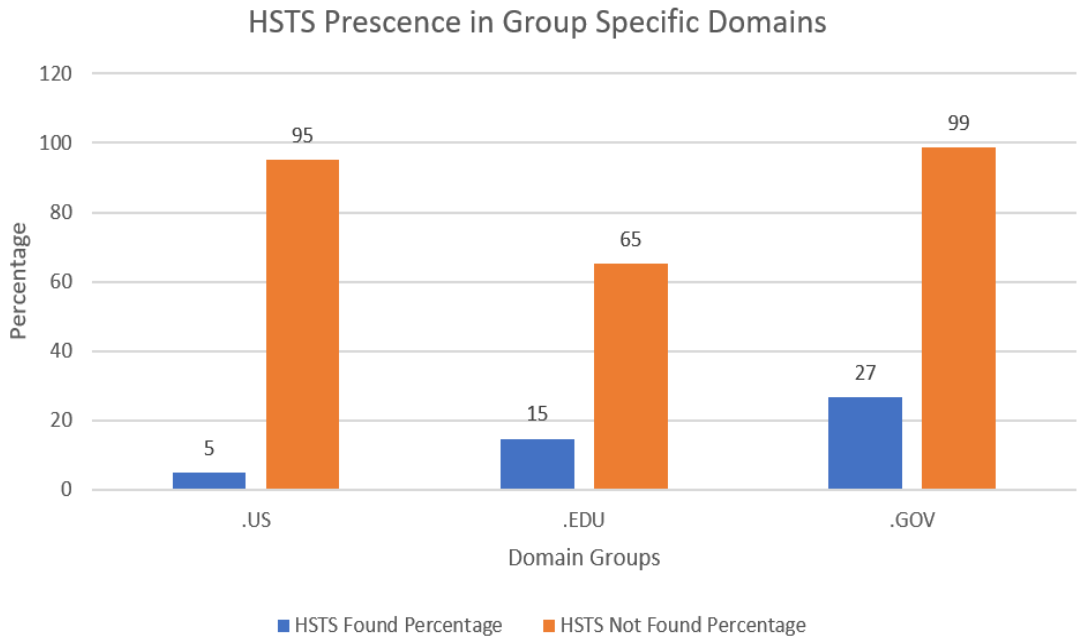

Figure 12. A bar graph visualizing the occurrence of HSTS headers detected versus the occurrence not detected for the scanned domain groups

4.3.7. Overall Domain: Scoring System Incorporation

We incorporated the proposed scoring system as part of the data analysis for all domains. The results were split into two groups. The first group included domains that scored a passing result of 70 or higher and the second group was domains that scored below a 70 which is considered a failure. Figure 13 demonstrates that based on the scoring system rules, 62% of the domains were given a passing score of 70 while 38% percent failed the scoring system. The simple benchmark scoring system can be used as a preliminary report to help establish a focus on acceptable configurations versus unacceptable configurations.

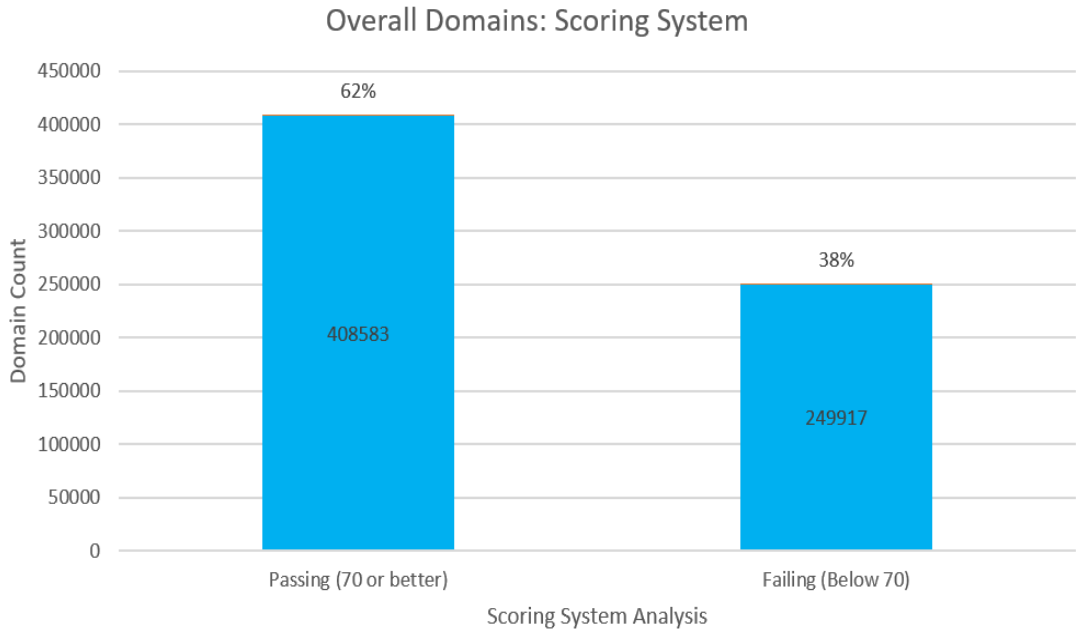

Figure 13. A bar graph visualizing the scoring system applied on the scanned domains.

4.3.8. Group Specific Domain: Scoring System Incorporation

The same scoring system was used at the domain category level for ".us,""".edu," and ".gov," as shown in Figure 14. Nearly all ".gov" domains met the proposed scoring system's passing criteria, with the highest occurrence, while ".us" had the lowest occurrence of passing scores. The ".us" domains had the highest frequency of failing results from the scoring system, while the ".edu" domains had the lowest frequency of failing results from the scoring system.

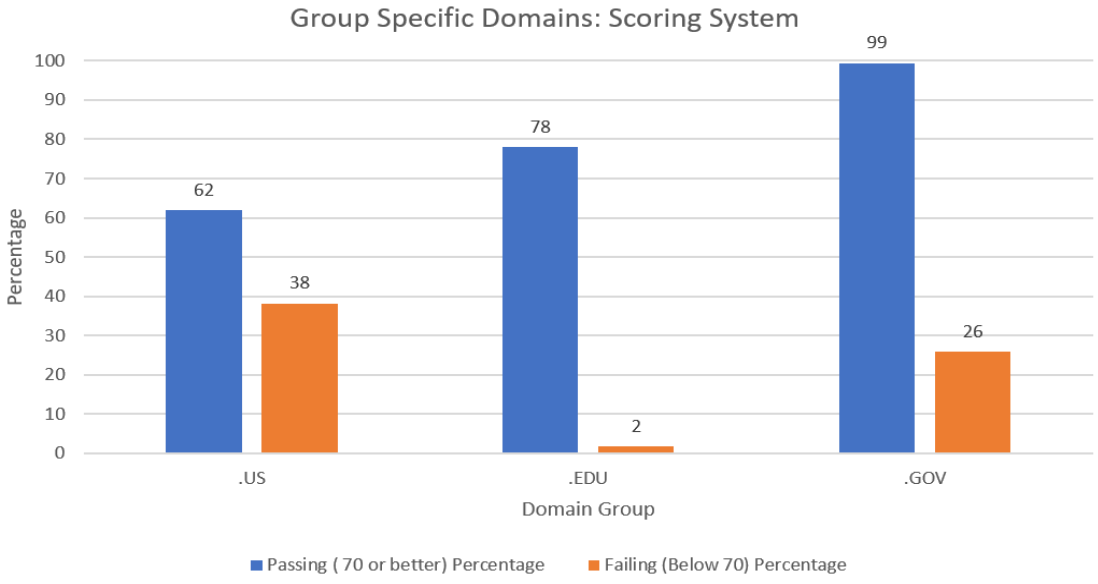

Figure 14. A bar graph visualizing the occurrences of domains with the scoring system applied by domain group.

Overall, there was a vast number of quantifiable results provided in this study. This study, however, had several limitations that were discovered during its development. When it came to the analysing the charts of data in this work, it was difficult to draw any concrete implications due to just having quantifiable information and no other information. More thorough research is required to establish the reasoning in the trends identified in the results section. Another limitation in this research was the number of domains gathered. Originally there were over a million domains analysed. Due to network connection denied error logs during the scanning of domains, many were not able to be scanned. In this research we were concerned with only domains that were successfully scanned without issues. By pursing the path of validity , 658500 domains were the final set used for result analysing. Lastly from a coverage standpoint, we lacked ".edu" domains due to not being able to obtain the zone file as explained earlier in the paper.

5. CONCLUSIONS

This study has revealed that US-related domains are not up to date with the latest protocol security and HSTS incorporation. By creating a simple scoring system with respect to RFC's most recent deprecations, a general sense of where US based domains stand numerically was easily noticeable. The scoring system rules can be applied to any domain to get a general sense of where their network protocol and HSTS presence stands. This study has also found that HSTS usage in US based domains is low and that the lack of awareness is one of the contributing factors behind it. It is important to incorporate HSTS knowledge in the workplace for any team that works with configuring network related security which would include education for all stakeholders.

It has also been revealed in the study that a high percentage of US-related domains have enabled lower versions of the HTTPS protocol which needs to be corrected to minimize downgrade related attacks. The path forward would be to notify the end user that owns the domain of this issue, however, due to ethical concerns, the domain names have been kept confidential to ensure privacy. It is therefore more effective to focus on the registrars who sell the domains pre-emptively. EDUCAUSE, GODADDY, and DOTGOV are the companies in this study that handle US-related domain extensions ".edu", ".us", and ".gov" respectively.

Future work for this research includes increasing the accuracy of gathering ".edu" domains since the zone file could not be acquired. EDUCAUSE stated that they will only give zone file if they can be positively benefited. If a future researcher partners with other major universities and sends another zone file request to EDUCAUSE, then the chances of them providing the zone file will increase the coverage of".edu" domains. An additional future work for this research is to perform another assessment of US-related domains in the next coming years and use this research as reference to show if the trend has improved or not. This study aids in the overall understanding of US-related domains and provides a compelling argument that the organizations in charge of facilitating these domains are made aware that there is susceptibility in many of the websites.

ACKNOWLEDGMENTS

The assistance provided by my professor Dr.Yu-Ju Lin was greatly appreciated. He provided me a successful path of how to organize this thesis and offered valuable input during its development. Additionally, I would like to thank Patrick Hill for allowing me to build upon his research work and utilize in this paper. I would also like to thank Raymond Wilcauskas, a former employee contracted by the Department of Energy who agreed to be part of the reviewing committee for my thesis.

REFERENCES

[1] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," www.rfc-editor.org, Jan. 1997, doi: 10.17487/RFC2068.

[2] R. Oppliger, SSL and Tls: Theory and Practice. Norwood, Ma: Artech House, 2016.

[3] J. Hodges, C. Jackson, and A. Barth, "HTTP Strict Transport Security (HSTS)," Nov. 2012, doi: 10.17487/rfc6797.

[4] "Domain Name Registry Services from GoDaddy Registry," registry.godaddy. https://registry.godaddy/ (accessed Apr. 21, 2022).

[5] "Hipo/university-domains-list," GitHub, Sep. 02, 2020. https://github.com/Hipo/university-domains-list

[6] "Common Crawl," Common Crawl. https://commoncrawl.org/

[7] "Amazon Athena - Serverless Interactive Query Service - Amazon Web Services," Amazon Web Services, Inc. https://aws.amazon.com/athena/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc

[8] "Data | .gov," home.dotgov.gov. https://home.dotgov.gov/data/ (accessed Apr. 21, 2022).

[9] rbsec, "sslscan2," GitHub, Apr. 21, 2022. https://github.com/rbsec/sslscan/ (accessed Apr. 21, 2022).

[10] N. Sullivan, "Why TLS 1.3 isn't in browsers yet," The Cloudflare Blog, Dec. 26, 2017. https://blog.cloudflare.com/why-tls-1-3-isnt-in-browsers-yet/#:~:text=The%20reductive%20answer%20to%20why%20TLS%201.3%20hasn%E2%80%99t

[11] "Qualys SSL Labs - SSL Pulse," www.ssllabs.com. https://www.ssllabs.com/ssl-pulse/

[12] P. Hill and Y.-J. Lin, "Evaluation of Trust Worthiness of State and County Government Websites." Accessed: Apr. 21, 2022. [Online]. Available: http://gator3168.temp.domains/~patriill/wp-content/uploads/2021/05/SAM21-1.pdf

[13] L. Garron, A. Dropbox, and D. Boneh, "The State of HSTS Deployment: A Survey and Common Pitfalls." Accessed: May 13, 2022. [Online]. Available: https://garron.net/crypto/hsts/hsts-2013.pdf

[14] W. J. Buchanan, S. Helme, and A. Woodward, "Analysis of the adoption of security headers in HTTP," IET Information Security, vol. 12, no. 2, pp. 118–126, Mar. 2018, doi: 10.1049/iet-ifs.2016.0621.

[15] S. De los Santos and J. Torres, "Analysing HSTS and HPKP implementation in both browsers and servers," IET Information Security, vol. 12, no. 4, pp. 275–284, Jul. 2018, doi: 10.1049/iet-ifs.2017.0030.

AUTHOR

DeJean Dunbar earned his B.S in Computer Science from Charleston Southern University in 2017. He is currently pursing his masters degree at Charleston Southern University. DeJean's interests in computer science include automation-based systems, database management systems, and networking security.