Scoring Vulnerabilities After Seeing a Chained Vulnerability Demonstration

Nikki Robinson
Capitol Technology University
nerobinson@captechu.edu

Abstract— The general problem was the NIST SP 800-40r3 (Souppaya & Scarfone, 2013) or the CVSS (FIRST, 2018a) did not provide enough information to prioritize vulnerability remediation. The specific problem was CVSS severity rankings were specific to individual vulnerabilities, which limited organizations to remediate vulnerabilities based on the potential downstream impact to other systems (Franklin, Wergin, & Booth, 2014). The purpose of this quantitative study was to use a pre-test / pro-test experiment to compare how cybersecurity professionals in the USMC rate vulnerabilities before and after seeing examples of vulnerability chaining using the CVSS calculator. The research question was, what score would cybersecurity professionals in the USMC give individual vulnerabilities before and after seeing vulnerabilities used in combination to create a more severe cyberattack? The research method used a quasi-experimental method with a pre-test / post-test design to identify how vulnerabilities would be scored before and after seeing a chained vulnerability demonstration. The results of the vulnerability scores were compared between the control and treatment groups, as well as the CVSS scores provided in versions 2.0 and 3.0 for each vulnerability. Participants from the control group changed two vulnerabilities from a Medium score to a High score; CSRF (from 7.5 to 9.0) and XSS (8.3 to 9.0). The treatment group did not change any vulnerability scores in a statistically significant manner, but the researcher found this was due to the overall higher scores for each vulnerability.

Keywords—vulnerability, chaining, NIST, scores

I. INTRODUCTION

This study explored the importance of Medium vulnerabilities, and how vulnerabilities were used in combination to create an attack as detrimental as a high or critical. Since the detrimental cyberattacks against large organizations, including the Equifax breach of 2017 (Berghel, 2017) and the Office of Personnel Management (OPM) breach of 2015 (Harvey & Evans, 2016), government agencies and businesses must be hyper vigilant about remediating vulnerabilities to ensure the protection of sensitive data. With the increased threat of cybersecurity attacks (Hammond, 2016), organizations may only have time to focus on the remediation of Critical and High vulnerabilities. If an organization opts out of addressing vulnerabilities which were classified at a lower level, were they more susceptible to a cyberattack? Should organizations create and maintain patch management solutions for Low and Medium exploitable vulnerabilities?

The National Institute of Standards and Technology (NIST) provided guidance for government agencies to create patch management and configuration management documentation, but it does not sufficiently deliver guidance for remediating vulnerabilities. However, the NIST SP 800-40r3 was not the only guide used to create patching strategies. The Common Vulnerability Scoring System (CVSS) is used to rate the severity of vulnerabilities but has moved from version 2.0 to version 3.0 (FIRST, 2018a). When versions CVSS 2.0 and CVSS 3.0 were compared, vulnerability scores for some vulnerabilities changed from *Medium* to *High*, and vice versa (FIRST, 2018b). This creates difficulty in patch and risk mitigation strategies as old vulnerabilities can be re-classified at a higher level, which lead to the reason for this study.

To address the potential gaps in knowledge, this study discovered if individuals rank *Low* and *Medium* vulnerabilities differently after knowledge of a chained attack. The introduction explored the explanation of why this research was important, general and specific problem statements, purpose, significance, and nature of the study. Further into this paper the research questions, the theoretical framework, definitions, assumptions, and scope of the study will be discussed.

II BACKGROUND

The general problem is the NIST SP 800-40r3 (Souppaya & Scarfone, 2013) or the CVSS (FIRST, 2018a) does not provide enough information to prioritize vulnerability remediation. Organizations were not given enough detail for each vulnerability to create a proper patch management plan to secure their environment. Both documents left prioritization and mitigation of vulnerabilities to the organization, which could leave critical applications or legacy systems vulnerable to cyberattacks.

The specific problem is the CVSS severity rankings are specific to individual vulnerabilities, which limited organizations to remediate vulnerabilities based on the potential downstream impact to other systems (Franklin, Wergin, & Booth, 2014). *Low* or *Medium* vulnerabilities can be used together to create a more sophisticated and harmful attack (FIRST, 2018a). Organizations may not be aware of the importance of patching or remediating *Low* and *Medium* vulnerabilities. By gaining more knowledge about vulnerability chaining, and how this was used to compromise systems, organizations will be better able to prioritize vulnerability remediation and create a more secure environment.

The purpose of this quantitative study is to use a pre-test / pro-test quasi-experiment to compare how cybersecurity professionals in the USMC rate vulnerabilities before and after seeing examples of vulnerability chaining using the CVSS calculator. This study also showed how vulnerability chaining was used with what the CVSS scoring system classify as *Low* and *Medium* vulnerabilities to provide more complex cyberattack. The intention of the pre-test measurement was to discover if individuals understood the basic CVSS scores and accurately scored *Low* and *Medium* vulnerabilities. The intention of the post-test measurement was to find out if, after seeing a chained attack, the individuals scored those vulnerabilities differently. In this quasi-experiment, the researcher hoped to find out if examples of vulnerability chaining would change the overall score of an individual vulnerability.

A Research Questions

The question this research sought to answer was, what was the score that cybersecurity professionals in the USMC gave individual vulnerabilities before and after seeing vulnerabilities used in combination to create a more severe cyberattack? The hypothesis was the CVSS score that USMC cybersecurity professionals assigned to vulnerabilities after seeing chained attacks will be statistically different than a control group not exposed to the chained attacks. The null hypothesis was the CVSS score that USMC cybersecurity professionals assigned to vulnerabilities after seeing chained attacks will not be statistically different than a control group not exposed to the chained attacks. Through identifying how these individuals rank vulnerabilities, there were several qualitative research questions, which could be asked, based on the results.

Research Question: Will cybersecurity professionals in field operations of the Marine Corps rank vulnerabilities higher, the same, or lower after seeing how vulnerabilities can be chained together?

Hypothesis: The CVSS score that Marine Corps cybersecurity professionals assign to vulnerabilities after seeing chained attacks will be statistically different than a control group not exposed to the chained attacks.

(Null) Hypothesis: The CVSS score that Marine Corps cybersecurity professionals assign to vulnerabilities after seeing chained attacks will not be statistically different than a control group not exposed to the chained attacks.

I. Data Analysis and procedures

The set of procedures for this quasi-experimental quantitative study were outlined to include the selection of participants and the control group. Before the control group or treatment group were surveyed, a pilot study took place to ensure validity and reliability of the quasi-experimental research design. The participants were selected at random for the control group and the experiment using a link provided to the USMC POC to distribute. Each participant was assigned a random number by a research assistant to ensure the anonymity of the individuals. Half of the sample was chosen at random to participate in the control group based on the AB text function in SurveyMonkey. The remaining half of participants were used in the experiment.

From both the control group and the experimental group, several pieces of data were analyzed, as follows. From the control group, it was important to find out how cybersecurity professionals ranked vulnerabilities with (or without) knowledge of chained vulnerability scenarios. The demographic information gathered from the participants provided insight into if knowledge of vulnerability chaining determines the score of vulnerabilities. Another piece of data to analyze was how much experience everyone had in cybersecurity and vulnerability management. Examining the control group was important before looking into the group which receives the treatment.

Once the survey results were analyzed from the control group, the treatment group results were examined. The most important component of the study was to see if scores changed between the pre-test and post-test scores from participants. This was done using a chart to show how individuals scored vulnerabilities before and after the demonstration. After determining if vulnerabilities were scored higher, lower, or the same, the length of time to complete the survey was inspected. It was possible the length of time to complete the survey meant the participants were unsure of how to answer or did not understand the question.

Inferential statistics were used to test the hypothesis, and the sample was identified using the statistics related to demographics. An inferential statistic review consisted of looking at the independent variable related to the dependent variable and using a One Sample t Test (Kent State University, 2018). This type of test looked at the statistical difference between zero and a change score (Kent State University, 2018). This test helped to determine if there was a change in score compared to the original measurements (scores for vulnerabilities provided by CVSS) (Kent State University, 2018).

I. chosen Vulnerabilities for experiment

The third and fifth pages of the survey were the pre-test and post-test questions and included the same in format and content. There was a description above each of these pages, which included how CVSS scores vulnerabilities ranged from *None* (0.0), *Low* (0.1 – 3.9), *Medium* (4.0 – 6.9), *High* (7.0 – 8.9), or *Critical* (9.0 – 10.0). The description also requested the participants to score vulnerabilities in a decimal format and provided an example for how to correctly input responses. Nine vulnerabilities were presented including a description directly from the OWASP website. The intention of providing the description was to ensure each participant understood fully which vulnerability to score. After each description, the participant was provided with a prompt to score the vulnerability based on the given description, as well as their own experience with vulnerability management.

Each of the nine vulnerabilities was chosen based on the inconsistencies of vulnerability score between versions 2.0 and 3.0 of CVSS, as well as any iterations of the vulnerability where it was scored as a *Medium*. The nine vulnerabilities chosen included Server-Side Request Forgery (SSRF), Cross-Site Request Forgery (CSRF), Carriage Return Line Feed (CRLF) Injection, Deserialization, HTTPOnly flag, Cross-site Scripting (XSS), Remote File Inclusion (RFI), SQL Injection (SQLi), and Authentication Bypass. Depending on the software or operating system affected by these vulnerabilities, the CVSS score was dissimilar. Each of these vulnerabilities was described, CVE number and description provided, as well as a possible CVSS score.

A. *SSRF*

The first vulnerability presented was SSRF, which was the potential for an attacker to make custom requests to a web server on an internal network (Särud, 2018). One example of a SSRF vulnerability was CVE-2018-1999039 which was released in August of 2018 (NIST, 2018i). This vulnerability was classified with a base score of 4.3 (*Medium*) in CVSS 3.0 and a base score of 4.0 (*Medium*) in CVSS 2.0 (NIST, 2018i). This specific CVE is for an SSRF which existed in plugin version 2.0.1 for Jenkins Confluence Publisher (NIST, 2018i). Depending on the software affected, CVE's related to SSRF on the NIST website were classified *Medium*, *High*, or *Critical*.

B. *CSRF*

The second vulnerability presented was CSRF, which allowed an attacker to log the victim in to a system with the attacker's credentials (Särud, 2018). One example of a CSRF vulnerability was CVE-2018-13401 which was released in December of 2018 (NIST, 2018f). This vulnerability was classified with a base score of 6.1 (*Medium*) in CVSS 3.0 and a base score of 5.8 (*Medium*) in CVSS 2.0 (NIST, 2018f). This specific CVE was for a CSRF vulnerability which existed in Jira versions before 7.13.1 and allowed attackers to obtain the CSRF token through a redirect vulnerability (NIST, 2018f). Depending on the software affected, CVE's related to CSRF on the NIST website were classified *Medium*, *High*, or *Critical*.

C. *CRLF*

The third vulnerability presented was CRLF Injection, which allowed an attacker to inject a header to grant internal servers the permissions to deploy other systems via a callback (CWE, 2019a). One example of a CRLF Injection was CVE-2017-7528 which was released in August of 2018 (NIST, 2018c). This vulnerability was classified with a base score of 6.5 (*Medium*) in CVSS 3.0 and a base score of 3.3 (*Low*) in CVSS 2.0 (NIST, 2018c). This specific CVE was for Ansible Towers which had Red Hat Engine 5 and was vulnerable to CRLF attacks (NIST, 2018c). Depending on the software affected, CVE's related to CRLF on the NIST website were classified *Low*, *Medium*, or *High.*

D. *Deserialization*

The fourth vulnerability presented was Deserialization, which was when an application deserialized untrusted data without verifying the information was valid (CWE, 2019b). One example of a Deserialization vulnerability was CVE-2016-9585, which was released in March of 2018 (NIST, 2018b). This vulnerability was classified with a base score of 5.3 (*Medium*) in CVSS 3.0 and a base score of 2.6 (*Low*) in CVSS 2.0 (NIST, 2018b). This CVE was for Red Hat JBoss EAP version 5, which was vulnerable to deserialization in the JMX endpoint (NIST, 2018b). This type of vulnerability could result in a denial of service attack against the machine (NIST, 2018b). Depending on the software affected,

CVE's related to Deserialization on the NIST website were classified *Medium*, *High*, or *Critical*.

E. HTTPOnly Flag

The fifth vulnerability presented was the HTTPOnly flag, which allowed attackers to obtain sensitive information through access to cookies (NIST, 2018a). One example of an HTTPOnly flag vulnerability was CVE-2014-9635, which was released in September 2017 (NIST, 2018a). This vulnerability was classified with a base score of 5.3 (*Medium*) in CVSS 3.0 and a base score of 5.0 (*Medium*) in CVSS 2.0 (NIST, 2018a). This CVE was for any version of Jenkins before 1.586 which did not set a Set-Cookie header when run on Tomcat 7.0.41 (NIST, 2018a). Depending on the software affected, CVE's related to HTTPOnly flags on the NIST website were classified *Medium*, *High*, or *Critical*.

F. XSS

The sixth vulnerability was XSS, which was able to steal cookies using this vulnerability. An example of an XSS vulnerability is CVE-2018-17952, which was released in December 2018 (NIST, 2018h). This vulnerability was classified with a base score of 6.1 (*Medium*) in CVSS 3.0 and a base score of 4.3 (*Medium*) in CVSS 2.0 (NIST, 2018h). This CVE was for an XSS vulnerability in eDirectory software prior to version 9.1 Service Pack (SP) 2 (NIST, 2018h). Depending on the software affected, CVE's related to XSS on the NIST website were classified *Not Available*, *Medium*, or *High*. The most surprising find about this vulnerability were the amount of XSS CVE's which did not have a CVSS version 2.0 or 3.0 score, considering XSS made OWASP's Top 10 list in 2017 (OWASP, 2018a).

G. RFI

The seventh vulnerability presented was RFI, which allowed a directory to be loaded as a file into a share (Kure, 2015). An example of an RFI vulnerability was CVE-2018-11101, which was released in May of 2018 (NIST, 2018e). This vulnerability was classified with a base score of 6.1 (*Medium*) in CVSS 3.0 and a base score of 4.3 (*Medium*) in CVSS 2.0 (NIST, 2018e). This CVE mentioned the use of two vulnerabilities, including XSS and using RFI to inject HTML code as a message (NIST, 2018e). Depending on the software affected, CVE's related to RFI on the NIST website were classified *Not Available*, *Medium*, or *High*.

H. SQL Injection

The eighth vulnerability presented was SQLi, which allowed an attacker to execute SQL commands on a database to read data from tables (NIST, 2018d). An example of a SQLi vulnerability is CVE-2018-11065, which was released in August of 2018 (NIST, 2018d). This vulnerability was classified with a base score of 4.3 (*Medium*) in CVSS 3.0 and a base score of 4.0 (*Medium*) in CVSS 2.0 (NIST, 2018d). This CVE was related to a component of RSA Archer, called WorkPoint, and was only vulnerable on versions prior to 6.4.0.1 (NIST, 2018d). Depending on the software affected, CVE's related to SQLi on the NIST website were classified *Medium*, *High*, or *Critical*.

I. Authentication Bypass

The ninth vulnerability presented was an Authentication Bypass, which was accomplished by using a modified URL parameter, manipulating a form, or counterfeiting sessions from the user (OWASP, 2018b). An example of an Authentication Bypass vulnerability was CVE-2018-1650, which was released in December of 2018 (NIST, 2018g). This vulnerability was classified with a base score of 5.5 (*Medium*) in CVSS 3.0 and a base score of 2.1 (*Low*) in CVSS 2.0 (NIST, 2018g). This CVE was specific to IBM QRadar SIEM versions 7.2 and 7.3, which allowed attackers to bypass authentication the administrator had configured (NIST, 2018g). Depending on the software affected, CVE's related to SQLi on the NIST website were classified *Medium*, *High*, or *Critical*. Table 2 shows how each vulnerability was scored in CVSS version 2.0 versus CVSS version 3.0.

TABLE I.

Vulnerability	**CVSS v2.0 Versus v3.0 Scores**	
	CVSS v2.0	***CVSS v3.0***
SSRF	4.3	4.0
CSRF	6.1	5.8
CRLF	6.5	3.3
Deserialization	5.3	2.6
HTTPOnly	5.3	5.0
XSS	6.1	4.3
RFI	6.1	4.3
SQLi	4.3	4.0
Authentication Bypass	5.5	2.1

After the participants were shown the pre-test survey page, which requested vulnerability scores for each of the nine vulnerabilities, the treatment group received a page containing possible chained vulnerability attacks. These attacks were chosen based on the potential for them to exist in real-world scenarios. Each chained vulnerability example contained a reference to someone who either used these chained vulnerability attacks or proposed the potential to exploit. Figure 1 displays how each of the chained vulnerability examples worked, which included the vulnerabilities presented in the pre-test and post-test questions.

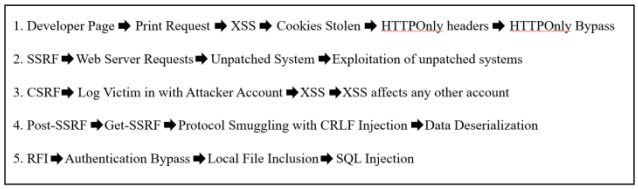

Fig 1. Vulnerability chain examples used in the experiment.

II. RESULTS

A. Pilot Results

The initial pilot participant noted some issues with understanding how and why the vulnerabilities were chosen. The individual believed the vulnerabilities were more like vulnerability categories, instead of specifically chosen vulnerabilities with a CVE ID. The individual also noted the researcher used OWASP definitions for vulnerabilities, where the individual thought STIG references would be more appropriate. The researcher politely disagreed with the individual and explained this in a follow-up e-mail.

The vulnerabilities were chosen because each one specifically fit into a vulnerability chaining example, not because they were based on any one technology or vulnerability in a system. The reason OWASP definitions were used were based on its wide usage within the cybersecurity community. OWASP was also used because several of the vulnerabilities were included in the OWASP Top Ten list (as mentioned in Chapter 3). Once the participant received this information, they agreed the definitions and vulnerabilities were correct and was satisfied with the three changes the researcher made to the survey.

The other participants did not provide additional feedback after taking the survey. As the researcher did receive helpful and descriptive information from one participant, this helped to shape the study. The pilot participants were asked to complete the survey within five business days and three of the participants complied. On the closing date the researcher sent a final note to Dr. Letteer with detailed instructions on the changes to the survey. The link to the survey was also provided to Dr. Letteer for distribution to the USMC cybersecurity groups.

B. Control Group

A total of 3 participants received the survey without the treatment, but only 2 participants scored both sets of vulnerabilities. The control group took an average of 20 minutes to complete the survey without the treatment. Table 3 contains the results of the control group pre-test and post-test mean scores. The vulnerabilities which saw a statistical difference of +/- 1 were CSRF and HTTPOnly. Authentication Bypass was the sole vulnerability which received a lower score, though it would not be considered statistically significant per the CVSS scale. The standard deviation was included to provide context between the pre-test and post-test mean scores.

Table ii.

Vulnerability	Control Group Results		
	Control Pre-Test Mean	**Control Post-Test Mean**	**Std. Deviation**
SSRF	8.7	9.2	9
CSRF	7.5	9	8.28
CRLF	8.2	8.2	0
Deserialization	9.5	9.6	9.57
HTTPOnly	7.6	8.7	8.19
XSS	8.5	9	8.75
RFI	9.6	9.7	9.7
SQLi	9	9	0
Authentication Bypass	9.2	9	9.12

C. Treatment Group

A total of 7 participants received the survey with the treatment, but only 6 completed scoring of both sets of vulnerabilities. The treatment group took an average of 27 minutes to complete the survey with the treatment. Table 4 contains the results of the treatment group pre-test and post-test mean scores. The vulnerabilities which saw a statistical difference of +/- 1 were CSRF and XSS. HTTPOnly and RFI received a lower score, though it would not be considered statistically significant per the CVSS scale. The standard deviation was included to provide context between the pre-test and post-test mean scores.

Table iii.

Vulnerability	**Treatment Group Results**		
	Treatment Pre-Test Mean	*Treatment Post-Test Mean*	*Std. Deviation*
SSRF	8.5	8.5	0
CSRF	7.4	8.6	8.06
CRLF	6.3	6.5	6.44
Deserialization	7.8	7.9	7.89
HTTPOnly	3.6	3.1	1.83
XSS	7.8	8.9	8.84
RFI	8.3	7.9	8.29
SQLi	9.1	9.2	9.16
Authentication Bypass	8.1	8.2	8.15

D. Mean – Control Group

Table 4 displays the variance between CVSS version 2.0, version 3.0, and control group pre-test and post-test scores. These scores show the wide variance not only between CVSS version 2.0 and 3.0, but also how the participants scored the vulnerabilities with only their knowledge and experience. While the pre-test and post-test scores are statistically similar, vulnerabilities show a statistically significant score for all vulnerabilities between CVSS versions and the control groups responses.

Table iv.

Vulnerability	**Control Group Results - Mean**			
	CVSS v2.0	*CVSS v3.0*	*Control Pre-Test Mean*	*Control Post-Test Mean*
SSRF	4.3	4.0	8.7	9.2
CSRF	6.1	5.8	7.5	9
CRLF	6.5	3.3	8.2	8.2
Deserialization	5.3	2.6	9.5	9.6
HTTPOnly	5.3	5.0	7.6	8.7
XSS	6.1	4.3	8.5	9
RFI	6.1	4.3	9.6	9.7
SQLi	4.3	4.0	9	9
Authentication Bypass	5.5	2.1	9.2	9

E. Mean – Treatment Group

Table 5 displays the variance between CVSS version 2.0, version 3.0, and treatment group pre-test and post-test scores. These scores show a wide variance not only between CVSS version 2.0 and 3.0, but also how the participants scored the vulnerabilities with only their knowledge and experience. While the pre-test and post-test scores are statistically similar on all but CSRF and XSS, vulnerabilities show a statistically different score for all vulnerabilities between CVSS versions and the control groups responses.

Table v.

Vulnerability	**Treatment Group Results - Mean**			
	CVSS v2.0	*CVSS v3.0*	*Trtmt Pre-Test Mean*	*Trtmt Post-Test Mean*
SSRF	4.3	4.0	8.5	8.5

Vulnerability	Treatment Group Results - Mean			
	CVSS v2.0	*CVSS v3.0*	*Trtmt Pre-Test Mean*	*Trtmt Post-Test Mean*
CSRF	6.1	5.8	7.4	8.6
CRLF	6.5	3.3	6.3	6.5
Deserializatio n	5.3	2.6	7.8	7.9
HTTPOnly	5.3	5.0	3.6	3.1
XSS	6.1	4.3	7.8	8.9
RFI	6.1	4.3	8.3	7.9
SQLi	4.3	4.0	9.1	9.2
Authentication Bypass	5.5	2.1	8.1	8.2

III. Conclusions

To address the research question and hypothesis, the interpretation of the findings will be discussed. Findings will include any relation between demographic information and vulnerability scores, relation to participants vulnerability scores and CVSS scores, as well as any difference between the control and treatment groups. Of major interest to the paper, will be the analysis of control and treatment groups, to find if vulnerability scores changed due to the chained vulnerability examples. The researcher will also discuss if the evidence collected was bound or unbound, and if the data confirms or contradicts the research question.

The research question asked if USMC personnel would score vulnerabilities differently or the same after reviewing a demonstration of chained vulnerabilities. While reviewing the differences in pre-test and post-test scores, the control group scored two of the nine vulnerabilities the same in both sets of questions, while the other vulnerability scores changed. The vulnerabilities scores which did not change were CRLF (8.2) and SQL injection (9). The average vulnerability score for the control group changed two vulnerabilities from a *Medium* score to a *High* score according to the CVSS calculator; CSRF (from 7.5 to 9.0) and XSS (8.3 to 9.0). However, one vulnerability score went down on the second round of scoring, the Authentication Bypass was changed from a 9.2 to a 9.0. Results from the control group were inconsistent, but the only statistically significant changes were shown in the CSRF and XSS vulnerabilities. This could just mean that the participants meant to

The hypothesis noted that USMC cybersecurity professionals would score the vulnerabilities in a statistically different way than the control group, which did not see the chained vulnerability demonstration. There were four vulnerabilities which saw statistically significant changes in scores between the control group and the treatment group. CRLF, Deserialization, HTTPOnly, and RFI vulnerabilities were scored at a +/-1 between the control and the treatment groups. This showed a difference in how vulnerabilities were scored after seeing a chained vulnerability attack, but the most fascinating part was the vulnerabilities were scored lower in the treatment group. The treatment group scored vulnerabilities overall lower than the control group. So, while the vulnerability scores themselves from the treatment group were not statistically different, compared to the control group it was a significant difference.

The most interesting finding of this study was the increased scores that participants gave to all vulnerabilities, except for the HTTPOnly flag. This was interesting since CVSS scored this vulnerability +/- 2 points higher than the average participant. It is possible this vulnerability is not as well known, and therefore could be not well understood in the community. The vulnerability chaining example did not increase the score of this vulnerability but decreased in the post-test treatment group. But the reasoning for this remains a mystery, as there is no indication in the data why the individuals chose to score this vulnerability lower after the demonstration. One could deduce the participants found this vulnerability less severe after seeing how it was used in a chained attack.

IV Future work

While the analysis determined that the treatment group only changed one vulnerability in a statistically significant manner, this leads to many other questions the researcher would like to ask. The first question is why these individuals did not score the other vulnerabilities higher, or whether the treatment did not affect their initial higher scores of the vulnerabilities. A qualitative study could be done with the same sample, to find if there is a correlation between the training they receive and the ability to score vulnerabilities higher. Another quantitative study could be done to show the exact CVE numbers and find if the individuals would be able to accurately score the vulnerabilities based on explicit technical detail.

Another possibility for future research is to perform this experiment with either a government or private industry organization. It could be very interesting to find out how other participants would score vulnerabilities, and if their training and education background would change their answers. The USMC clearly has some excellent documentation and training material, and it seems like their staff is well trained on vulnerability scoring. It would also be helpful to find a larger sample to conduct this experiment on. If the questions were shortened, it is possible more people would be willing to complete the survey.

It could also be interesting to choose another set of vulnerabilities and show new vulnerability chaining examples. The intention of this research was to take relatively well-known vulnerabilities to score, but it may be more interesting to choose lesser known vulnerabilities to see how the scores would change. This could lead to further research as participants may need more information on the vulnerabilities and could potentially be done using a qualitative method. It could provide more concrete evidence as to why individuals score vulnerabilities, and not strictly the scoring numbers.

References

Easttom, C. (2018, March). The Role of Weaponized Malware in Cyber Conflict and Espionage. In *ICCWS 2018 13th International Conference on Cyber Warfare and Security* (p. 191). Academic Conferences and publishing limited.

Berghel, H. (2017). Equifax and the latest round of identity theft roulette. *Computer; New York*, *50*(12), 72-76. doi: http://dx.doi.org.login.captechu.edu:2048/10.1109/MC.2017.4451227

Common Weakness Enumeration. (2019a). *CWE-93: Improper neutralization of CRLF sequences ('CRLF injection')*. Retrieved from http://cwe.mitre.org/data/definitions/93.html

Common Weakness Enumeration (2019b). *CWE-502: Deserialization of untrusted data*. Retrieved from http://cwe.mitre.org/data/definitions/502.html

FIRST (2018a). *Common vulnerability scoring system v3.0: Examples*. Retrieved from https://www.first.org/cvss/specification-document

FIRST (2018b). *Common vulnerability scoring system v3.0: Specification document*. Retrieved from https://www.first.org/cvss/examples

Franklin, J., Wergin, C., & Booth, H. (2014). National Institute of Standards and Technology (2014, April). *CVSS Implementation Guidance*. NIST Interagency Report 7946. doi: http://dx.doi.org/10.6028/NIST.IR.7496

Hammond, B. (2016). DHS official: Cyber threat data should be public good more than profit maker. *Cybersecurity Policy Report*. Retrieved from https://search-proquest-com.login.captechu.edu:2443/docview/1773928842?accountid=44888

Harvey, S., & Evans, D. (2016). *Defending against cyber espionage: The US office of personnel management hack as a case study in information assurance*. Paper presented at the 2016 National Conference on Undergraduate Research, University of North Carolina Asheville, Asheville, NC.

Kent State University (2019, February 1). *SPSS tutorials: One sample t test*. Retrieved from https://libguides.library.kent.edu/SPSS/OneSampletTest

National Institute of Standards and Technology (2018a). CVE-2014-9635 detail. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2014-9635

National Institute of Standards and Technology (2018b). CVE-2016-9585 detail. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2016-9585

National Institute of Standards and Technology (2018c). CVE-2017-7528 detail. Retrieved from https://nvd.nist.gov/vuln/detail/CVE2017-7528

National Institute of Standards and Technology (2018d). CVE-2018-11065 detail. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2018-11065

https://nvd.nist.gov/vuln/detail/CVE-2018-11101
National Institute of Standards and Technology (2018f). CVE-2018-13401 detail. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2018-13401

National Institute of Standards and Technology (2018g). CVE-2018-1650 detail. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2018-1650

National Institute of Standards and Technology (2018h). CVE-2018-17952 detail. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2018-17952

National Institute of Standards and Technology (2018i). CVE-2018-1999039 detail. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2018-1999039

Open Web Application Security Project (2018a). OWASP top 10 – 2017.Retrieved from https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

Open Web Application Security Project (2018b). Testing for bypassing authentication schema (otg-authn-004). Retrieved from https://www.owasp.org/index.php/Testing_for_Bypassing_Authentication_Schema_(OTG-AUTHN-004)

Särud, L. (2018, February 6). *Do not dismiss the small vulnerabilities!* [Web blog post]. Retrieved from https://blog.detectify.com/2018/02/06/small-vulnerabilities/

Souppaya, M., & Scarfone, K. (2013, July). National Institute of Standards and Technology. *Guide to enterprise patch management technologies*. NIST Special Publication 800-40; Revision 3. Retrieved from http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf

