Improved Software Vulnerability Patching Techniques Using CVSS and Game Theory

Louai Maghrabi, Eckhard Pfluegel, Luluwah Al-Fagih
School of Computer Science & Mathematics
Faculty of Science, Engineering & Computing
Kingston University
London, UK
{Louai.Maghrabi, E.Pfluegel, L.Al-Fagih}@kingston.ac.uk

Roman Graf, Giuseppe Settanni, Florian Skopik
Department of Digital Safety & Security
Austrian Institute of Technology
Vienna, Austria
{Roman.Graf, Giuseppe.Settanni, Florian.Skopik}@ait.ac.at

***Abstract*—Software vulnerability patching is a crucial part of vulnerability management and is informed by using effective vulnerability scoring techniques. The Common Vulnerability Scoring System (CVSS) provides an open framework for assessing the severity of software vulnerabilities based on metrics capturing their individual, intrinsic characteristics. In this paper, we enhance the use of CVSS for vulnerability scoring with the help of game theory by modelling an attacker-defender scenario and arguing that, under the assumption of rational behaviour of the players, an effective vulnerability patching strategy could be achieved with an optimal strategy, solving the game.**

We have implemented our strategies as new functionality in the software tool CAESAIR [1]. This research builds on our previous work [2], where we have used CVSS to inform the design of the utility functions, by performing the Nash equilibrium analysis of the game. Our findings may result in more accurate defence strategies for system administrators.

I. Introduction

Modern society increasingly depends on the use of the cyberspace: sending and receiving emails, interacting with the Internet, consuming cloud-based services, posting personal information on online social networks, and in the near future, the connection to prosumers of the Internet-of-Things (IoT). The security of this cyberspace is crucial, however cyber security breaches are occurring more and more frequently, and they pose a huge threat to organisations and individuals alike. Cyber security involves both protecting technical assets of complex IT systems, and the human users interacting with these systems.

In this paper, we focus on software assets and their protection, but we do not neglect the human aspect in security as we employ game theoretical concepts in order to model the motivation behind cyber attackers. Software assets contain vulnerabilities which expose them to threats. In order to protect these assets, organisations use vulnerability management frameworks to reduce threat impact. Typically, software assets can contain many vulnerabilities, and an organisation should manage vulnerabilities in order to deal with a potentially large number of security weaknesses. Software vulnerability patching is one of the key activities of vulnerability management, and only informed resource allocation decisions will yield effective defenses against the above threats. Hence, scoring systems such as CVSS [3], the IP360 Vulnerability Scoring System [4], and the Microsoft Exploitability Index [5] have been developed in order to rate the impact severity of specific software vulnerabilities, depending on their characteristics.

Likelihood assessment is the process of establishing the probability of an attack occurring ([6], [7]) and, together with vulnerability scoring, might be seen as a way to implement risk scoring which has been suggested to be more effective than vulnerability scoring alone [8]. Likelihood assessment appears, in general, to be a challenging and elusive task [4].

One way of conducting likelihood assessments is to assume that the interaction between potential cyber attackers and system administrators (who act as defenders) can be modelled as a non-cooperative game, where both players act rationally, seeking to optimise their expected payoffs. The game can then be analysed by making some assumptions on the players' strategies, which yield an expected optimal strategy from which system administrators can derive strategic decisions for vulnerability patching. In [9], the authors investigate the use of game theory as an alternative to CVSS-only vulnerability patching strategies. The initially studied game is a two-target security game, also known as a search game in the literature. The game theoretic analysis is used as a filter to narrow down the number of vulnerabilities that need to be considered for patching, e.g. using traditional vulnerability scoring techniques such as CVSS. In [6], the authors have adopted this methodology, focusing on the attacker's perspective inspired by game theory, using a slightly different game, but in addition they capture concurrent and simultaneous attacks and also model return of investment equations.

One aim of our research is to devise novel strategies for vulnerability patching based on sound theoretical foundations, in particular when using CVSS aided by game theory. However, another important aim is to achieve practical contributions through system implementations, thereby validating our strategies design.

The following contributions are made in this paper:

- We design novel software patching strategies, using CVSS and game theory, based on analysing a complete information, non-zero sum static security game where the utility functions of the players are informed by CVSS metrics. This extends our work in [2] by using infor-

mation derived from a mixed Nash equilibrium strategy. Compared to [6], we extend the simplified version of their model by making explicit use of CVSS metrics for the design of a more realistic and practically oriented game. In contrast to [9], we use CVSS scoring information to improve the accuracy of our game theoretic utility function, hence making our model more accurate and suitable for real world scenarios.
- We integrate our CVSS-based game theoretical vulnerability patching methodology as an additional functionality of the cyber intelligence analysis system CAESAIR (Collaborative Analysis Engine for Situational Awareness and Incident Response [1]), developed previously by co-authors of this paper (c.f. [10]). We illustrate as a case study how a particular vulnerability can be analysed, after having been extracted from the collected incident reports, and we show some of the benefits that our alternative strategy can bring.

This paper is organised as follows: in Section II, we review the idea of vulnerability scoring, in particular when using CVSS. In Section III, we use game theory concepts to aid the use of CVSS by proposing our scenario and analysing the game. Then in Section IV, we introduce CAESAIR [1], a cyber intelligence analysis system. We then evaluate our game using this tool and implement our new functions into the software in SectionV. We conclude in Section VI.

II. Vulnerability Scoring using CVSS

Vulnerability scoring is the process or method for describing the risk that a specific vulnerability presents. It is a tool making vulnerability management more effective, as it helps with assessing and responding to the risk of software vulnerability exploits, and is hence a crucial activity within vulnerability management in the broader sense. A common approach for vulnerability scoring is based on a numerical scoring function $\mu : v \rightarrow S$, that maps an individual vulnerability v into the score domain S, expressing the severity (or criticality) of v. This value could be used as a basis for priority ranking within automated vulnerability management.

A. The Common Vulnerability Scoring System (CVSS)

A few experimental, proprietary and open frameworks for vulnerability scoring exist such as the SANS vulnerability analysis scale [11], and the Microsoft Exploitability Index [5]. The Common Vulnerability Scoring System (CVSS) is a vulnerability scoring technique that is used by major security organisations for their security management, and is nowadays recognised as an open standard. It was launched by the National Infrastructure Advisory Council (NIAC) in 2004. The Forum of Incident Response and Security Teams (FIRST) currently maintains CVSS and oversaw the launch of CVSS v2.0 in 2007, and the latest version v3.0 in 2015 [3]. CVSS is an open framework that provides a quantitative analysis with standardised vulnerability scores for single vulnerability management policies. In addition, it can prioritise risks by specifying CVSS to a specific organisation or system environment using the environmental metric. NVD maintains records of all the published vulnerabilities along with their relevant CVE IDs and their corresponding CVSS v2.0 metric scores [12], and this CVE identifier is obtained from the CERT database.

There are several limitations to the CVSS for vulnerability scoring. The main problem with vulnerability scoring is that it is possible for two different vulnerabilities to have the same score, although they both have individual characteristics. There are some possible remedies, whilst this is impossible to prevent in general, a careful design of the individual scoring functions should at least take this problem into account.

B. CVSS Metrics

CVSS consists of three main metric groups as shown in Figure 1, and assesses in its most basic form the severity of software vulnerabilities through its Base Score μ_B – this metric represents the intrinsic and fundamental characteristics of v that are independent of time and specific user environments. The Base score can then be refined by defining the score for the following metrics in order to more accurately reflect the risk posed by the vulnerability to an asset:

- Temporal metrics which reflect the characteristics of the vulnerability v that change over time.
- Environmental metrics which represents the characteristics of v that are unique to the user's environment.

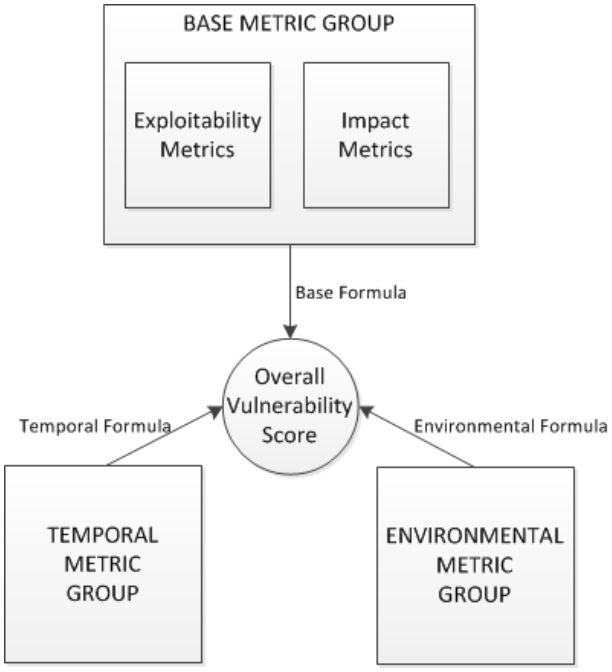

Figure 1. CVSS

Generally, the Base and Temporal metrics are specified by the organisation, while the environmental metric depends on the end-users.

In CVSS v2.0, the equation defining the Base Score is

$$\mu_{Base} = k_1 \cdot \mu_{Imp} + k_2 \cdot \mu_{Exp} - k_3$$

where $k_1 = 0.71$, $k_2 = 0.47$ and $k_3 = 1.76$ (rounded to two decimal places). The Impact subscore μ_{Imp} describes the potential impact that an exploit has on the system and is defined as

$$\mu_{Imp} = k_4 \cdot [1 - (1 - \mu_{Imp,C})(1 - \mu_{Imp,I})(1 - \mu_{Imp,A})].$$

where $k_4 = 10.41$, and $\mu_{Imp,C}$, $\mu_{Imp,I}$ and $\mu_{Imp,A}$ denote the impact on the CIA security requirements of Confidentiality, Integrity and Availability. In addition, the Exploitability sub score μ_{Exp} describes the means and ease of exploiting the vulnerability. Each of the two metrics contribute in an additive fashion to the final numerical score ranging from 0 to 10. The description of these equations is based on the presentation in [13, Section 3.2], slightly simplifying the formulae under the assumption of $\mu_{Imp} \neq 0$.

III. Game Theoretical Methodology

In this paper, our motivating scenario is the management of software assets within an organisation, and the existence of a newly found vulnerability in one of the software assets. We need to decide how critical this vulnerability might be and ultimately decide on any defense strategy. Our methodology is based on the assumption that the interaction between cyber attackers and system administrators (defenders) in charge of vulnerability management can be modelled as a non-cooperative game where both players act rationally, seeking to optimise their expected payoffs. The game can then be analysed by making some assumptions on the players' strategies, which yield an expected optimal strategy from which system administrators will be able to make a decision for vulnerability patching.

A. The Security Game

Focusing on the single software asset that contains a vulnerability, we introduce the terminology of a *single-target small security game* which can be defined in extensive normal form by the means of the following utility matrix:

Defender ↓ Attacker →	s_A^a	s_A^{-a}
s_D^d	$-\kappa$, $-A$	$-\kappa$, 0
s_D^{-d}	$-L$, $B - A$	0, 0

Figure 2. Attacker-Defender Matrix

In this table, we denote the defender's pure strategies s_D^d (defend) and s_D^{-d} (not defend) in rows, and the columns are the attacker's pure strategies: s_A^a (attack) and s_A^{-a} (not attack). We denote κ the defense cost, L the defender's loss from an attack, A the attacker's cost, and B the benefit of the attacker. This game has been studied in the past (see e.g. [6]) and is suitable for our scenario which is based on a single asset vulnerability assessment.

We now use CVSS scores to inform the utility functions of the game. The loss L in the defender's utility function is due to a threat event impact, exploiting the vulnerability and affecting the asset's CIA security requirements. This can be modelled as follows, using the CVSS CIA impact sub scores and value components of the asset:

$$L = \mu_{Imp,C} \cdot V_C + \mu_{Imp,I} \cdot V_I + \mu_{Imp,A} \cdot V_A. \quad (1)$$

Using vector notation, we introduce the CIA impact score vector

$$\bar{\mu}_{Imp} = (\mu_{Imp,C}, \mu_{Imp,I}, \mu_{Imp,A})$$

and the asset value vector

$$\bar{V} = (V_C, V_I, V_A),$$

we can write the loss as

$$L = \bar{\mu}_{Imp} \cdot \bar{V} \quad (2)$$

using the dot product. Furthermore, we assume that the attacker's benefit B equals the loss of the defender and hence

$$B = L,$$

and that his cost is proportional to the inverse of the CVSS exploitability sub score:

$$A = \alpha \cdot {\mu_E}^{-1}$$

where α is a constant that needs to be suitably defined (c.f. next section).

B. Attack Likelihood Computation

When analysing the presented security game, it can be shown that there are no pure Nash equilibrium strategies [6] and that the values

$$p^* = \frac{L - A}{L} = 1 - \frac{A}{L} \quad (3)$$

and

$$q^* = \frac{\kappa}{L} \quad (4)$$

form a mixed Nash equilibrium strategy (s_D^*, s_A^*) where p^* and q^* are the probability of defense and attack respectively. This will be the underlying method to compute our attack likelihood in this scenario, based on q^*. Equation (3) shows that we have to impose the condition

$$0 \leq 1 - \frac{\alpha}{L \cdot \mu_E} \leq 1 \Longleftrightarrow 0 \leq \frac{\alpha}{L} \leq \mu_E$$

in order to ensure that p^* can be used as a probability in the mixed Nash equilibrium strategy.

In our system, we offer the user the choice between two user *profiles*, in order to choose between the two strategies S_1 and S_2. The Game Theory (GT) Profile implements S_2 and furthermore contains a predefined range of profile settings such as *conservative* or *un-conservative*. Here, a more conservative profile assumes that the attacker is more powerful and requires less cost for his attack which would correspond to a smaller value for α, but making sure that the above inequality conditions are still met. Ultimately, a system administrator needs to choose his preferred profile, but can also switch profiles since that may help give clearer trends when trying to prioritise similar output values for different vulnerabilities.

C. Application to Vulnerability Patching

We now compare and contrast the following two vulnerabilities patching strategies:

- *High Score* (S_1). We decide to apply a patch if the overall CVSS score is relatively high, i.e. base score $\mu_B(v) > \theta$ where $0 < \theta \leq 10$ is a threshold parameter.
- *Nash Equilibrium* (S_2). We compute the p^* as a function of the individual CVSS impact sub scores and the asset value V, and the CVSS exploitability subscore $\mu_E(v)$. We adopt a mixed defense strategy where on average, we defend with a probability p^*.

Strategy S_1 is the normally proposed use of CVSS scoring functions as a severity ranking. It is clear that this strategy will only focus on the defender's vulnerability and its impact on his system, hence ignoring the attacker's motivation. It is also independent of the value of the asset that we wish to protect, and hence, less specific than Strategy S_2.

Hence, we advocate the use of strategy *Nash Equilibrium* as an alternative. Indeed, the use of the mixed Nash strategy depends on the value p^*, with values close to 1 indicating that the defense is most critical and we hence interpret p^* as a severity value, possibly scaling it to a range similar to the CVSS ranking.

IV. Cyber Intelligence Analysis Using CAESAIR

In our previous work (c.f. [10]) we introduced the concept of a cyber intelligence analysis system, called CAESAIR [1], and we demonstrated its application within a European Control System Security Incident Analysis Network (ECOSSIAN) [14]. CAESAIR provides analytical support for security experts carrying out cyber incident handling tasks on a national and international level [15], and facilitates the identification of implicit relations between available pieces of information. It provides powerful correlation capabilities, which support the tasks carried out by the analysts of a Security Operation Center (SOC) during the incident handling process. CAESAIR employs three different security information correlation techniques [16], differing in the way relevant information is extracted from the imported document. Information is acquired from a multitude of security-relevant sources. These sources include custom repository, open source intelligence (OSINT) feeds, IT-security bulletins, as well as standardized vulnerability library (CVEs). CAESAIR evaluates how the collected documents are connected to one another, and allows the analyst to select the most appropriate correlation method and to flexibly adjust relevance metrics.

Upon reception of an incident report, the system extracts its relevant features (depending on the enabled correlation method), and maps them to the document's feature vector. Based on mapping results, CAESAIR then evaluates the correlation between the analyzed document and any other document in the knowledge-base, by applying one of three developed methods. Through its GUI, CAESAIR displays the list of the derived most relevant documents, sorted according to their similarity to the selected one, and a list of identified related concepts. This allows the analysts to faster and more extensively examine significant security information, to identify meaningful relations between reported incidents, discovered vulnerabilities, targeted systems, and involved actors, allowing to achieve shorter incident response times. The advantage of this cyber threat intelligence solution, lies in the fact that CAESAIR does not only collect and aggregate incident and threat data, making it comfortably available to the analyst, it also addresses a value of protected assets and estimates exploitability effort for attackers creating additional value by extensive correlation of aggregated data with large amounts of security information collected from several relevant sources, and provides the experts with a list of related information, greatly supporting them in the decision making process while handling cyber incidents.

V. Implementation and Case Study

In this section, we illustrate how a system administrator can use the CAESAIR tool in order to retrieve detailed, real-time information about software vulnerabilities, including CVSS severity scores. We document how the opportunity to use the game theoretical strategy proposed in this paper, could improve standard vulnerability patching strategies. We will further illustrate this improvement using a specific vulnerability (the Heartbleed Bug as a case study).

Until the latest version CVSS v3.0 is supported by the National Vulnerability Database (NVD), our implementation uses the CVSS v2.0 metrics.

A. Use Case

To apply this method within CAESAIR, we consider the use case of a Cyber Incident (CI) SOC analysis team receiving multiple incident reports within a short time interval. The experts need to prioritize the incoming incidents, identifying those more critical, i.e., those affecting the most valuable asset, and handle them first. Thanks to the presented approach, the system adopts the calculated incident severity as an indicator of criticality, and therefore derives the corresponding incident priorities. The algorithm calculates the incident severity, as the probability p* described in (3), and displays it to the user, as a further property of the incident.

To calculate this probability, CAESAIR retrieves the base CVSS score and individual scores from CVSS dataset. It examines then the list of monitored assets (e.g., extracted from CPE dataset), checking if they are affected by the specific vulnerability, and defines a numeric value for each asset representing the criticality of the asset.

Provided that the cost of an attack determines the exploitability effort for an attacker, the system calculates a probability value, based on the aforementioned parameters, for each incident report and presents it on the dashboard. The expert team can hence immediately see how critical an incident is for the particularly monitored infrastructure, and react accordingly.

This functionality makes CAESAIR unique in comparison to other existing solutions. We will explain the new functionality that has been added for this research based on a

user story of an expert (Bob) who is the system administrator analysing incident reports, assisted by CAESAIR. Bob's local configuration maintains critical assets, for example in the form of a CPE list. For each asset in the list, a numerical value between 1 and 5 is stored. Bob is able to make decisions in a time-critical environment, where multiple incident reports become available in real-time and the impact of corresponding cyber attacks on the critical assets needs to be mitigated.

The events unfold as follows:

```
12:14:05 Expert Bob started his shift, analysing an
  incident report on last night's network service
  requests.
12:14:07 Bob receives another incident report,
  summarising critical availability of the company's
  database server.
12:14:10 Bob imports this new report into CAESAIR,
  using a REST API.
12:15:10 CAESAIR automatically accesses CVSS scores
  through NVD online database and displays CVSS base
  score.
12:16:00 The patching priority list seem inconclusive,
  when only looking at CVSS scores.
12:16:30 Bob decides to switch to the GT Profile.
12:17:00 CAESAIR has determined more accurate severity
  values, based on the value of a critical asset and
  a dangerously high exploitability score.
12:17:30 Bob manages to update the critical CPE
  software asset and great damage to the organisation
  can be prevented.
```

B. Heartbleed Case Study

The famous Heartbleed vulnerability CVE-2014-0160 has a CVSS v2.0 base score of 5.0, which was later revised to 7.5 in v3.0 [17] as there was some criticism in the literature to the base score in v2.0 as being too low [18]. We compute the severity score that one would obtain by using Strategy S_2. For Heartbleed we have, according to [19], $\mu_{Imp,C} = 0.66$, $\mu_{Imp,I} = \mu_{Imp,A} = 0$ and $\mu_E = 10$. Hence, we compute p^* using equation (3) for a generic confidentiality value V_c as

$$p^* = 1 - \frac{\alpha}{6.6 V_C}.$$

We now assume that our system allows asset vector component values between 1 and 5 as an estimate of the asset value (up to scaling). For an asset confidentiality value $V_C = 1$, the system needs to work with the condition $0 \leq \alpha \leq 6.6$ and for a conservative profile could set $\alpha = 2.2$, obtaining the value $p^* = 0.69$. This value would be scaled to be normalised for use with CVSS, obtaining the severity ranking 6.9. On the other hand, if the profile is un-conservative, we can see that the score diminishes. We can see that severity ranking, based on these likelihood values, is very different to that based on CVSS alone.

VI. Conclusion

In this paper, we have designed novel software vulnerability patching strategies using CVSS and game theory to decide whether to apply a specific patch or not. This improves on previous game theoretic approaches by a more rigorous use of real-world vulnerability information based on CVSS scores. In addition, our approach has been implemented as part of an existing tool. Our first experiments with this tool and its new functionality are encouraging and there is great scope for additional investigation and further results.

In our future work, we will refine the game theory model and enrich the functionality in CAESAIR. Ultimately, our research will lead to the creation of automated security assessment tools with higher and more accurate prediction of security attacks.

Acknowledgement

This work is financially supported by the King Abdullah Scholarship Program (KASP), and was partly funded by European Union FP7 project ECOSSIAN (607577).

References

[1] Austrian Institute of Technology, "CAESAIR," 2016. [Online]. Available: https://service.ait.ac.at/pydio/data/public/971d1f

[2] L. Maghrabi, E. Pfluegel, and S. F. Noorji, "Designing Utility Functions for Game-Theoretic Cloud Security Assessment : A Case for Using the Common Vulnerability Scoring System," in *International Conference On Cyber Security And Protection Of Digital Services (Cyber Security)*. London: IEEE, 2016. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7502351

[3] FIRST, "Common Vulnerability Scoring System v3.0: Specification Document," FIRST (Forum of Incident Response and Security Teams), Tech. Rep., 2015. [Online]. Available: https://www.first.org/cvss/specification-document

[4] Tripwire, "Tripwire Vulnerability Scoring System," Tripwire, Tech. Rep., 2016. [Online]. Available: http://www.tripwire.com/register/tripwire-vulnerability-scoring-system/

[5] Microsoft, "Microsoft Exploitability Index," 2014. [Online]. Available: http://technet.microsoft.com/de-at/security/cc998259.aspx

[6] L. Samarji, N. Cuppens-Boulahia, F. Cuppens, S. Papillon, W. Kanoun, and S. Dubus, "Coordination and Concurrency Aware Likelihood Assessment of Simultaneous Attacks," in *Third International Conference on Security and Privacy in Communication Networks SecureComm*, vol. 153, no. 1, 2015, pp. 524—-529. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84952309737&partnerID=tZOtx3y1

[7] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, S. Dubus, and A. Martin, "Success Likelihood of Ongoing Attacks for Intrusion Detection and Response Systems," in *International Conference on Computational Science and Engineering*, vol. 3. IEEE, 2009, pp. 83–91.

[8] T. Reguly, "Does Anybody Really Care About Vulnerability Scoring?" 2013. [Online]. Available: https://www.tripwire.com/state-of-security/risk-based-security-for-executives/risk-management/does-anybody-really-care-about-vulnerability-scoring/

[9] G. Gianini, M. Cremonini, A. Rainini, G. L. Cota, and L. G. Fossi, "A Game Theoretic approach to Vulnerability Patching," *2015 International Conference on Information and Communication Technology Research (Ictrc)*, pp. 88–91, 2015. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-23829-6_37

[10] G. Settanni, F. Skopik, Y. Shovgenya, and R. Fiedler, "A Collaborative Analysis System for Cross-Organization Cyber Incident Handling," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy*, 2016, pp. 105 – 116.

[11] T. Palmaers, "Implementing a Vulnerability Management Process," SANS, Tech. Rep., 2013.

[12] H. Booth, D. Rike, and G. Witte, "THE NATIONAL VULNERABILITY DATABASE (NVD): OVERVIEW," National Institute of Stndards & Technology (NIST), Tech. Rep. December, 2013.

[13] P. Mell, K. Scarfone, and S. Romanosky, "A Complete Guide to the Common Vulnerability Scoring System Version 2.0," FIRST (Forum of Incident Response and Security Teams), Tech. Rep., 2007. [Online]. Available: http://www.first.org/cvss/cvss-guide.pdf

[14] H. Kaufmann, R. Hutter, F. Skopik, and M. Mantere, "A structural design for a pan-European early warning system for critical infrastructures," *Elektrotechnik und Informationstechnik*, vol. 132, no. 2, pp. 117–121, 2015. [Online]. Available: http://dx.doi.org/10.1007/s00502-015-0286-5

[15] R. Fiedler, M. Carolan, D. Conroy, K. Boettinger, M. Gall, G. Brost, and C. Ponchel, "A collaborative cyber incident management system for European interconnected critical infrastructures," *Journal of Information Security and Applications*, pp. 1–19, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jisa.2016.05.005

[16] G. Settanni, Y. Shovgenya, F. Skopik, R. Graf, M. Wurzenberger, and R. Fiedler, "Correlating Cyber Incident Information to Establish Situational Awareness in Critical Infrastructures." [Online]. Available: http://www.flosko.at/ait/2016-pst.pdf

[17] FIRST, "Common Vulnerability Scoring System v3 .0 Examples," FIRST, Tech. Rep. July, 2016.

[18] M. Roytman, "Heartbleed Is Not A Big Deal?" 2014. [Online]. Available: http://blog.kennasecurity.com/2014/04/heartbleed-is-not-a-big-deal/

[19] National Vulnerability Database, "Vulnerability Summary for CVE-2014-0160," 2014. [Online]. Available: https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-0160