Environmental Aware Vulnerability Scoring

Andreas Eitel
Fraunhofer Institute for Experimental Software Engineering IESE, Kaiserslautern, Germany

Keywords: CVSS, Environmental Metrics, IT-security, Network Security.

Abstract: When assessing the CVSS value of a vulnerability, the Environmental Metrics are often ignored. There are several reasons for this. However, this score is essential for the prioritization of vulnerabilities. The author proposes an approach that should generate the environmental score systematically and highly automated. For this purpose, various information about the systems and the network is needed, which should be managed in a model. An algorithm uses the linked information to automatically determine the Environmental Metrics. Experts without a security background should thus be able to determine this score in the same way as experts. The results should also be repeatable and independent of the evaluator.

1 RESEARCH PROBLEM

Efficient information security management is increasingly difficult for engineers. Nowadays, companies operate many servers and host many digital services as well. In addition, they operate many appliances, such as firewall or storage systems. The number and complexity of these systems is constantly increasing. Therefore, it is increasingly difficult for engineers to keep track of and assess their security vulnerabilities.

Engineers use many tools to regularly scan servers and digital services in order to find and assess security vulnerabilities. (Eschelbeck, 2005) describes such processes as common good practice.

Vulnerability scanners add criticality scores to their findings. Usually, the scanners use the CVSS (Common Vulnerability Scoring System) framework to calculate a criticality score.

The CVSS framework uses up to three metrics to calculate the risk score. The CVSS risk score can consist of a base score, a temporal score and an environmental score. The former two metrics, the Base Metrics and the Temporal Metrics, are applicable for every environment; therefore, they are typically available in vulnerability databases.

Another problem with Environmental Metrics scoring is its lack of objectivity. When determining the Environmental Metrics according to the CVSS user guide (Hanford and Heitman, 2015), depending on the information available, the security engineer has some leeway in decision making, because many parameters, such as the configuration settings or the network topology of a system, can be taken into account and assessed differently by different experts.

However, the third metric (Environmental Metrics) is not available in vulnerability databases and in most tool results. Because the Environmental Metrics score depends on the actual network environment of the vulnerability, it is not applicable in general.

Consequently, engineers tend to ignore the Environmental Metrics when determining the CVSS risk score (Frühwirth and Männistö, 2009; Allodi et al., 2017). The determination based on CVSS is a manual task requiring engineers with broad security knowledge, which renders the results less repeatable and quite time-consuming (example in Section 2). Besides, it is a complex task, especially for large networks (Holm and Afridi, 2015; Allodi et al., 2017), even without considering technical configuration details (Allodi et al., 2017), as it does not scale well with the network size.

However, it is a mistake to ignore the Environmental Metrics as this score can have a strong (positive) influence on the resulting criticality (Frühwirth and Männistö, 2009; Gallon, 2010). Ignoring this score may result in CVSS criticality scores that are not adequately reflecting the real situation (Allodi and Massacci, 2013; Allodi et al., 2017) and may lead to lower security and compliance levels (Verizon, 2015).

Inadequately scored vulnerabilities can lead to inefficient security management. In the worst case, engineers are patching vulnerabilities immediately that do not require urgent response. Wrong prioritization does not only waste effort; eventually, it may also

Eitel, A.
Environmental Aware Vulnerability Scoring.
DOI: 10.5220/0009839104780485
In *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS 2020)*, pages 478-485
ISBN: 978-989-758-426-8

cause higher costs due to an unplanned and expensive downtime for an important service outside the maintenance window (Allodi and Massacci, 2013).

In order to avoid these problems, the author of this article suggests two essential steps. First, companies need to collect as much relevant information for the process of determining the Environmental Metrics as possible in advance by including various existing information sources, such as the companies configuration management database (CMDB), risk management reports and system configurations. Second, an algorithm must be devised to determine the Environmental Metrics based on both the vulnerability information and the information collected beforehand. Introducing more automation speeds up the assessment process and supports the engineer by reducing the complexity of this task in large networks. In addition, the results will be more repeatable, and depending on the degree of automation, the process is facilitated for security maintenance engineers lacking a strong security background.

Thus, the author addresses the following overall research question: *How can the CVSS Environmental Metrics be determined in an efficient, repeatable and highly automated way in large networks by engineers that may not necessarily have a strong security background?*

2 EXAMPLE

In this example, the author of this article wants to give a short summary on which information is necessary to determine the Environmental Metrics in CVSS v3. The process of determining the environmental score starts after a vulnerability scanner identifies a vulnerability. A security engineer then identifies the asset and looks up its requirements regarding confidentiality, integrity and availability (also known as CIA). At this point, it is possible to access a database or other sources that contain this information (e.g., due to a previous risk assessment). If this information is not available, the security engineer has to determine it by, for example, consulting the responsible stakeholder. Besides the CIA requirements, the security engineer has to determine the Modified Base Metrics (see also (FIRST, 2015)). These metrics include the modified values of Attack Vector, Attack Complexity, Privileges Required, User Interaction, Scope and CIA impact. In case of the Attack Vector he needs to check whether the vulnerability is exploitable (as it may be defined in the Base Metrics) or whether a security specific component is in place that influences the Attack Vector and thus allows another estimation. As an information source, the engineer uses a network plan, documentation about the structure (e.g., vlans) and the configuration of security specific components. For the determination of the Attack Complexity, the engineer has to decide how complex the exploitation of a vulnerability would be. Hence he has to consider, for example, the system configuration. Determining the Privileges Required metric determines what kind of privileges an attacker must possess prior to exploit the vulnerability. In order to determine this value, the engineer needs to know the system configuration as well as the network structure (e.g., the user must already be authenticated before he can reach the system at all). For the next value, the User Interaction metric, the engineer must decide whether a user needs to take action in order to exploit the vulnerability. The Scope metric defines whether a successful exploitation would also affect another scope (e.g., breaking out of a sandbox). In the last step, the engineer has to determine the impact of a successful exploitation on the three CIA properties.Apparently, that this process does not scale well with large networks and with many vulnerabilities.

3 OUTLINE OF OBJECTIVES

The author of this article derives three objectives from the overall research question. First, it is necessary to devise on an environmental information model for organizing all the information about the system environment and components involved. This information model is mapped to a corresponding data model in order to store the information in a database. Second, repeatable manual and highly automated processes (as not everything may be automated) must be defined to elicit necessary information about the environment and fill the data model. The third and last objective aims at developing an algorithm that uses the available information to determine the Environmental Metrics, as well as the overall CVSS score. Version 3.0 of the CVSS framework will be the basis of this research.

3.1 Environmental Information and Data Model

When security engineers determine Environmental Metrics, they need to use multiple information sources. According to the author's experience, these sources are often based on documents or pictures (such as network infrastructure plans, table of available vlans, etc.). In practice, all solutions have in

common that the information is distributed across several places and organized within a software such as a wiki or in shared folders. In addition, the relevant pieces of information are not linked. Thus, a security engineer has the time consuming task to find and read the necessary documents (and others have to keep the documentation up to date). It is up to the security engineers doing the analysis to understand all the environmental information related to the systems. This situation is not practical for large networks. Current solutions lack a suitable linking of relevant information and a common interface, that provides uniform, machine-readable access in order to support the security engineer.

An information model facilitates understanding and an easier access to the information. Therefore, the information model must be able to reflect the relationship between systems and services. This includes information about the type of a relationship, such as "service A uses service B" or "service C protects service A by limiting access to local machines". Moreover, the model must be able to represent details about the configuration of a system, such as the offered services or hardening measures. Besides, the model must be interoperable as it needs to include information originating from sources such as a risk assessment process. Relevant results of such a process are, for example, the applicable security policy, the criticality of the service for a company or the importance of confidentiality, integrity and availability for the various system assets. Based on the information model aggregation and reasoning about environmental information is possible, which facilitates a faster determination of the environmental metrics.

The essential research question (RQ1) is: *How must an information model look like in order to provide reusable, machine-readable, interoperable information that enables reasoning about the environment of a vulnerability?*

3.2 Information Collection

It is important to have up-to-date and relevant information about the environment when calculating the Environmental Metrics. Especially in large networks, it is a time-consuming task to gather all relevant information. Many solutions are available that assist the user in gathering and storing all sorts of information related to an asset, but that output their results in their own, proprietary format which is not necessarily machine-readable. Initiatives such as CCE (Common Configuration Enumeration) assign a unique, common identifier to a security-related configuration issue (Waltermire et al., 2016). This is an important approach, especially when results are required to be machine-readable. Unfortunately, CCE currently focuses only on software-based configurations and does not consider any hardware or physical configuration. In addition, some of the required information may not be available digitally (yet) and needs a manual elicitation. Another aspect to focus on is the update interval, because not all information changes in the same interval and need an update every time one determines the Environmental Metric. Thus, it is possible to save significant time and effort, when an information item is only updated or determined when necessary.

In summary, it is a big challenge to merge all information and to organize it in an efficient data model that allows a targeted, fast and machine-readable access. This is necessary to facilitate a process with a higher degree of automation.

The essential research question (RQ2) is: *How can the relevant information sources be determined, utilized and merged in order to manage their information within an information model?*

3.3 Evaluation Algorithm

Besides the information needed for evaluation, the evaluation algorithm itself is one of the most important parts of the process. Existing algorithms (see State of the Art in Section 4) focus on special configurations and consider only a small number of variables. The CVSS user guide (Hanford and Heitman, 2015), provides guidance on how to determine the Environmental Metrics. However, it is far from trivial to map this advice to a deterministic algorithm, as it is not exactly clear which parameters to consider, which gives rise to considerable discretion in decision making. An automated evaluation algorithm should create objective, repeatable results and it should be clear which parameters it considers. After all, the algorithm should deliver a result that allows even engineers who lack security proficiency to decide about the real criticality of the vulnerability without requiring additional subjective ratings. Furthermore, the algorithm operates on the basis of the data model and its semantics implied by the underlying information model. As it runs in an automated way, it can consider many parameters like existing protection measures, the configuration of the system as well as other vulnerabilities on the same or peer systems in the network. In order to automate this algorithm, several assumptions have to be made, which need to be researched.

The essential research question (RQ3) is: *How must an evaluation algorithm for the determination of Environmental Metrics look like to identify and incorporate the relevant parameters and their weighting to*

create repeatable results?

4 STATE OF THE ART

Determining Environmental Metrics in a highly automated way involves the three objectives (RQ1-3) of collecting all necessary information, structuring and linking the data via an information model as well as providing an evaluation algorithm. This section surveys the related work covering these problem domains in order to identify the gap in the current research. Table 1 summarizes the insights gained from the related work.

(Eschelbeck, 2005) states that security vulnerabilities are discovered on a daily basis and that companies are advised to develop best practices to cope with these challenges. He suggests important best practices based on his findings after he analyzed statistical vulnerability information over a three-year period. The identified best practices, such as classify, prioritize, integrate, measure and audit, are important requirements for all parts of this research. The described vulnerability management process should be kept in mind when devising on the evaluation algorithm (cf. Subsection 3.3). One of the key aspects is prioritization which – when relying only on NVD (National Vulnerability Database) data (CVSS without environmental metrics) – is rather limited, as the severity of vulnerabilities can considerably vary among different organizational contexts (Frühwirth and Männistö, 2009; Gallon, 2010). However, adding this missing information requires substantial effort and some companies are not able to spend the money or lock the required resources. (Frühwirth and Männistö, 2009) therefore developed a method for practitioners that allows to simulate the improvement potential of environmental metrics in order to convince managers to evaluate these metrics. The authors only use distribution models from the literature and publicly available data in the NVD. (Gallon, 2010) simulated the environmental factors as well, but focused on the aspect of how the individual factors affect the overall environmental score. Although their work is based on an earlier specification of CVSS, parts of it may contribute to the evaluation algorithm because their simulation algorithm could be used as a replacement in case some information is not available at the time of evaluation.

(Rui et al., 2009) present a hierarchical asset vulnerability assessment model that considers the information collected by a vulnerability scanning tool as an environmental factor. They present an optimization of the original CVSS method that is more accurate than the original method. In comparison to the research idea presented here, their work is based on an older version of the CVSS framework. Besides, they consider only a smaller number of attributes, such as "family" (operating system, protocol), "category" (attacktype) and other vulnerabilities in the same service, but not, for example, the network topology and existing protection mechanisms. Nevertheless, their work will be considered when working on the objectives.

(Hahn, 2010) uses the CVSS framework to develop a risk scoring mechanism in a Smart Grid and introduces a CVSS-host scoring. His work is also based on an older version of the CVSS framework and of limited use for this research due to the changes in the current CVSS version. Nevertheless, his idea and implementation of CVSS-based host scores may be useful when designing the evaluation algorithm.

(Gallon and Bascou, 2011) adapt the attack graphs definition to use attack graphs in combination with the CVSS framework. They calculate a host and network damage score in order to assess the impact of attacks on a host of the target network. The CVSS part of their work is based on an older version of the CVSS framework and does not consider environmental metrics. On the one hand, the research proposed here may improve their results by providing CVSS values that consider the environmental factors. On the other hand, their way of calculating the network damage score may be helpful in the development of an evaluation algorithm.

(Allodi and Massacci, 2013) claim that patching all vulnerabilities with a "high" CVSS score is not really useful. They show that CVSS lacks a real measure of likelihood of exploitation. In their controlled experiment, they used datasets from the NVD, Exploit-DB, Symantec/Kaspersky Thread Reports and Exploit Kits to determine when it makes sense to patch a vulnerability. Their results support the assumption of this research proposal that a prioritization is very important and that it is a valuable source to keep in mind when collecting information and designing an evaluation algorithm.

(Khosravi-Farmad et al., 2014) use Bayesian attack graphs in order to model the interactions between vulnerabilities. They are referring to an older version of the CVSS framework and consider only the environmental factors confidentiality, integrity and availability. This research proposal however aims to include more environmental factors. Although it is important to consider the interconnections of vulnerabilities, including them is not necessary for the early versions of an evaluation algorithm, but it should be considered in later versions.

Table 1: Summary of the related work.

Reference	CVSS version	Considers environment	Contributes to		
			RQ1	RQ2	RQ3
(Eschelbeck, 2005)	n/a	n/a	✗	✗	✓
(Frühwirth and Männistö, 2009)	2.0	✓	✗	✗	✓
(Gallon, 2010)	2.0	✓	✗	✗	✓
(Rui et al., 2009)	2.0	✓	✓	✓	✗
(Hahn, 2010)	2.0	✓	✗	✗	✓
(Gallon and Bascou, 2011)	2.0	✗	✗	✗	✓
(Allodi and Massacci, 2013)	n/a	✗	✗	✓	✓
(Khosravi-Farmad et al., 2014)	2.0	✓	✗	✗	✓
(Hanford and Heitman, 2015)	3.0	✓	✓	✓	✓
(Wang et al., 2016)	2.0	✓	✓	✓	✗
(Allodi et al., 2017)	3.0	✓	✓	✗	✓
(Rapid7, 2018)	3.0	✓	✗	✓	✓

(Wang et al., 2016) analyzed existing vulnerability evaluation methods and because of missing environmental information, created their own method with several environmental factors to better characterize the environment. They propose a 7-tuple containing authentication information, SQL information, memory information, URL information, cron jobs, traffic information and host information. Finally, they define an impact matrix and a mapping matrix to obtain the overall evaluation result. In comparison to the goals of this research proposal, they do not integrate the results into the CVSS value. Besides, the environmental factors used in their approach seem to ignore information such as the network configuration and available security policies. Nevertheless, the work should be considered when working on the objectives.

(Allodi et al., 2017) conducted an experiment, based on the current CVSS framework version, to determine how difficult it is for a human assessor to change the vulnerability score due to changes in security requirements of networks and systems. The 29 M.Sc. students that participated in the experiment have not considered the system configurations. The results (among others things) revealed that the scoring guidelines are difficult to apply and that the approach does not scale well with the size of the network. This supports the assumptions described in the research problem. Their insights and their approach will be helpful when designing the information model and an evaluation algorithm.

(Rapid7, 2018) InsightVM estimates the criticality not only by using the CVSS framework, they are using a proprietary equation to additionally calculate the "real risk". The equation ignores the CVSS Environmental Metrics, but takes the CVSS Base and Temporal Metrics into account. In addition, it incorporates exposure, malware kits, exploit rank and time. InsightVM ranks the vulnerabilities based on these results. Thereby, the tool introduces an additional risk number in the range of 1–1000. More details about this risk estimation approach is provided in (Rapid7, 2018). The insights gained may guide the work on all objectives.

Finally, the CVSS user guide (Hanford and Heitman, 2015) gives several hints on how to score a vulnerability, which will be taken into account when working on all objectives.

The literature survey revealed several gaps: Some authors completely ignore environmental factors or just simulate them. In the approach proposed here, environmental factors and their determination play a crucial role. By considering, capturing and structuring various types of information, an algorithm shall be enabled to calculate the environmental metrics automatically and fill this gap. Apart from environmental factors, many works use CVSS 2.0 to determine environmental metrics. This research proposal uses CVSS 3.0 as a basis for scoring. Because the score is defined differently in versions 2.0 and 3.0, works based on version 2.0 are only partially applicable to this proposal. Another gap is that in some works, environmental metrics are determined manually, or they are only simulated. Thus, these works offer no information model or structured way of managing environmental information. In contrast, the research proposed here aims to determine such information in a (highly) automated way with an information model as its basis. In works that consider environmental metrics, only a limited number and often only very specific factors are considered. With the proposed information model, the number of factors and types of information taken into consideration is not limited.

5 METHODOLOGY

The author aims at an iterative approach to cope with the research questions. He can integrate lessons learned and insights from previous iterations as well as emerging research results. In doing so, the author can expect new research results and can improve the entire approach in every application. An abstract view of the overall approach is presented next. Figure 1 summarizes the methodology graphically.

5.1 Environmental Information and Data Model

The author plans the following steps for elaborating the information and data model:

1. Existing work, such as the preliminary results of the state of the art, will be analyzed to define the requirements and necessary information for the information model.
2. As a prerequisite to develop the information model, the process of determining the Environmental Metrics will be scrutinized to gain a better understanding of the relevant environmental factors and how they can be effectively and efficiently elicited. The individual steps in the process must be analyzed by conducting several evaluations in different networks. This will help identifying generic and reoccurring questions, relationships and necessary information. Based on these results and the requirements elicited in Step 1, the information model is derived.
3. To manage the data, an object-oriented data model will be developed, based on the information model from Step 2. An information set could be, for example, an asset object (host), a service (http) or a vulnerability. Before a connection can reach this host, it must pass a firewall that applies several filtering rules to the connection. The model should be capable of storing this information together with its relationship (offers a service; has vulnerability; is protected by; etc.). For example, these relationships can be modeled as object relations and details, such as the concrete rules for a host or vulnerability information, can be modeled as properties of a class. Which relationships are possible, what kind of information needs to be stored as well as possible constraints will be researched.
4. The capability and suitability of the model will be tested by conducting several manual evaluations in different networks.
5. The author plans to conduct a case study to test whether all necessary information can be formalized, managed and manipulated. The study should determine if quick and efficient automatic evaluation is possible and if this evaluation is able to derive new issues. This will be done together with the information collection approach of Section 5.2.
6. Finally, the reusability of the model in related application domains will be assessed. For example, with all the information available, an algorithm may be devised that suggests improvements in the network structure or to mitigate weaknesses.

5.2 Information Collection

Good decision making requires a significant amount of information for reasoning. The author plans the following steps to collect the information:

1. The information needs will be determined by surveying the state of the art and by carrying out manual evaluations to gain better insights into the relevant factors of vulnerability scoring. This is implicitly done and already a part of the upper Section 5.1, Step 1 and 2.
2. For every piece of information used in step 1, the author will look for tools that can derive the information in an automated way. He also needs to consider how this information can be represented within the information model. Some of the information may not be retrievable in an automated way and has to be elicited manually. In this case, the author will explore methods to facilitate the elicitation.
3. After Step 2, an import tool will be provided that parses the results of all the other tools, integrates the information and creates a data model. As also information from manual elicitations needs to be incorporated, the import tool must provide well defined interfaces. Integrating the various information sources will presumably require a large number of different parser modules. While the basic work contributes to the research question, providing an exhaustive set of parsers in support of every conceivable tool result does not. Therefore, some of the information may be considered simply as “given” or “available” as long as it can be safely argued that providing an import module would be a straightforward but tedious task.
4. The resulting tool will be evaluated by using it with several networks and analyze the results each time. The evaluation can be done together with the planned case study in Step 5 of Section 5.1.

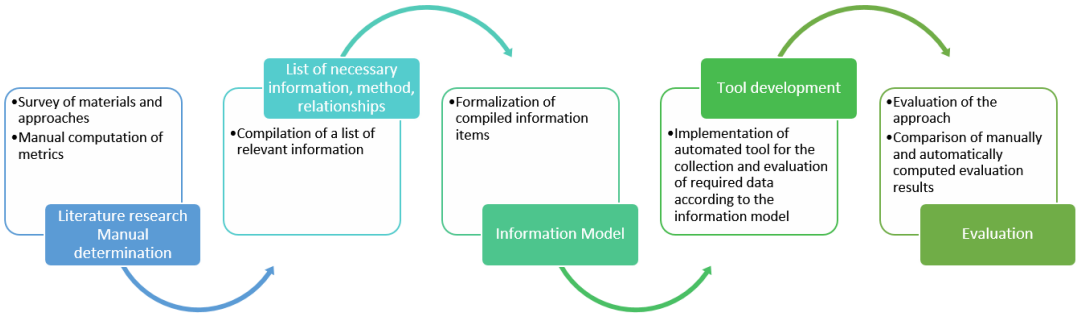

Figure 1: Graphical summary of the methodology.

5.3 Evaluation Algorithm

The third and final part of the proposed research is to develop an evaluation algorithm that takes the results of Sections 5.1 and 5.2 as input and outputs a CVSS compatible Environmental Metric as a result. So far, the following steps are planned:

1. A draft version of the evaluation algorithm will be derived from the manual evaluation procedure. While working on Steps 1 and 2 in Section 5.1, necessary parts of the algorithm can already be formulated.
2. To refine the existing draft algorithm, appropriate weightings for each combination of information and relationship type will be determined. This will require some simulations and testing.
3. As not every piece of information is always available. The author needs to define which information is obligatory and which information is optional and as a result refine the algorithm to that extend.
4. Finally, the author will test the algorithm by conducting several evaluations based on different networks and vulnerability findings. The results will be discussed with other security engineers doing the same evaluation manually for comparison. Ultimately, the algorithm shall reliably deliver similar results that are considered as valid by the other security engineers.
5. In addition, the author will check if it is possible to use, develop or extend a metric that takes more parameters into account than the CVSS framework does.

6 EXPECTED OUTCOME

The expected outcome is primarily a method with an implementation in a tool that determines the CVSS Environmental Metrics in a highly automated way (highly automated as not all information can be gathered automatically) and that uses an information model providing a machine-readable representation of the environment (e.g., network topology and configuration) and all other relevant information sources (see Figure 2). The corresponding Information Security Management Process works as follows:

- Use the tool to fill or update the information model instance with data from sources such as the network, scanner outputs or other databases.
- Regularly process the data in order to determine the Environmental Metrics.
- Calculate the new CVSS score including the Environmental Metrics part.
- Display a list of vulnerabilities ordered by their “real” criticality within the target environment.

The envisioned solution will extend the state of the art by providing an algorithm and tooling to (semi-)automatically determine the Environmental Metrics. It will provide the community with an information model that represents a network in a machine-readable way, storing also the relationships, their environmental weight and the influence of individual network nodes, components and services. The model also will allow non-experts to assess a CVSS environmental score in their local environment in a reliable and repeatable way, even in large networks. The following measurable benefits of this approach are expected:

- Due to the determination of the CVSS score with the Environmental Metric, a more realistic prioritization of a vulnerability is possible.
- The realistic prioritization allows working more efficiently and produces fewer costs as it improves maintenance planning (e.g. a fixed maintenance window can be used instead of fixing the vulnerability right away).

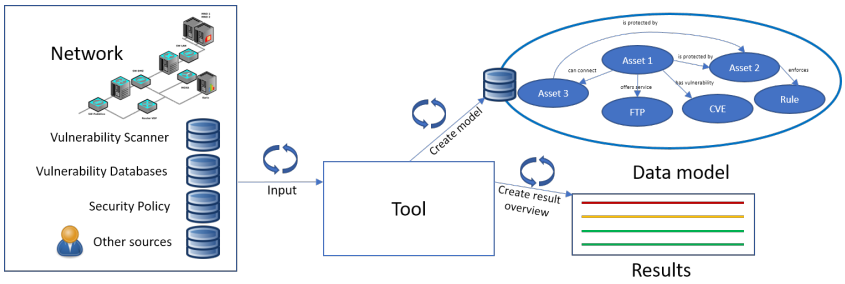

Figure 2: Schematic overview of the expected outcome.

- Even engineers with limited security proficiency can determine the Environmental Metrics in a highly automated way.
- This approach helps to determine the Environmental Metrics in a faster way, especially for larger networks.
- The Environmental Metrics are computed more objectively and repeatably, as the algorithm leaves less leeway for subjective human judgements.

7 STAGE OF THE RESEARCH

The research proposed in this article just started, and the research problem has been identified. A preliminary state of the art and state of the practice research has been conducted. The objectives, the methodology as well as the expected outcome has been described in a preliminary stage. Next, the author will discuss the current state within the security community to receive some feedback. Besides, he searches for a supervisor to start working on the problem described.

REFERENCES

Allodi, L., Biagioni, S., Crispo, B., Labunets, K., Massacci, F., and Santos, W. (2017). Estimating the assessment difficulty of CVSS environmental metrics: An experiment. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Allodi, L. and Massacci, F. (2013). How cvss is dossing your patching policy (and wasting your money). In *BlackHat USA 2013*.

Eschelbeck, G. (2005). The laws of vulnerabilities: Which security vulnerabilities really matter? *Information Security Technical Report*, 10(4):213–219.

FIRST (2015). Common Vulnerability Scoring System v3.0: Specification Document.

Frühwirth, C. and Männistö, T. (2009). Improving CVSS-based vulnerability prioritization and response with context information. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009*, pages 535–544.

Gallon, L. (2010). On the impact of environmental metrics on CVSS scores. In *Proceedings - SocialCom 2010: 2nd IEEE International Conference on Social Computing, PASSAT 2010: 2nd IEEE International Conference on Privacy, Security, Risk and Trust*.

Gallon, L. and Bascou, J. J. (2011). Using CVSS in attack graphs. In *Proceedings of the 2011 6th International Conf. on Avail., Reliability, Security, ARES 2011*.

Hahn, A. (2010). Smart grid architecture risk optimization through vulnerability scoring. In *2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply, CITRES 2010*.

Hanford, S. and Heitman, M. (2015). Common Vulnerability Scoring System v3.0: User Guide.

Holm, H. and Afridi, K. K. (2015). An expert-based investigation of the Common Vulnerability Scoring System. *Computers and Security*.

Khosravi-Farmad, M., Rezaee, R., and Bafghi, A. G. (2014). Considering temporal and environmental characteristics of vulnerabilities in network security risk assessment. In *2014 11th International ISC Conference on Information Security and Cryptology, ISCISC 2014*.

Rapid7 (2018). Quantifying Risk with InsightVM. Last Accessed: 2019-12-19. https://www.rapid7.com/globalassets/_pdfs/product-and-service-briefs/rapid7-solution-brief-quantifying-risk-insightvm.pdf.

Rui, L., Danfeng, Y., Fan, L., and Fangchun, Y. (2009). Optimization of hierarchical vulnerability assessment method. In *Proceedings of 2009 2nd IEEE International Conference on Broadband Network and Multimedia Technology, IEEE IC-BNMT2009*.

Verizon (2015). Pci compliance report. Technical Report. *Verizon Enterprise*.

Waltermire, D., Quinn, S., Booth, H., Scarfone, K., and Prisaca, D. (2016). The Technical Specification for the Security Content Automation Protocol (SCAP). *NIST Special Publication 800-126*.

Wang, S., Xia, C., Gao, J., and Jia, Q. (2016). Vulnerability evaluation based on CVSS and environmental information statistics. In *Proceedings of 2015 4th International Conf. on Computer Science and Network Tech., ICCSNT 2015*.