# Experimentation on operational cyber security in CRATE

**Teodor Sommestad**
Olaus Magnus väg 42
581 11 Linköping
SWEDEN

Teodor.Sommestad@foi.se

## *ABSTRACT*

*Many theories and tools in the cyber security domain are difficult to test with data from operational environments. First, the sensitivity of data makes asset owners wary of letting researchers collect the data the need from their systems. Second, since the ground truth is partly unknown in operational environments, data collected in "the wild" cannot be easily used to test situational awareness aspects. For example, tests of theories related to system vulnerability and intrusion detection are difficult to test. The difficulty of performing empirical tests with operational data and the lack of experimental tools hampers research in the cyber security domain. Empirical tests of vulnerability assessments methods are rare and intrusion detection accuracy is typically tested using the datasets produced by DARPA in the late 1990's, despite that this dataset is outdated and was known to have serious validity issues already in the early 2000's.*

*The advent of cyber ranges is a potential solution to this problem. In cyber ranges, modern virtualization technology can be used to simulate cyber environments under controlled conditions. As cyber ranges make it possible to control the ground truth, they allow experimentation and observation of variables and processes related to cyber security. For instance, by simulating the usage patterns and of an operational environment and injecting attacks under controlled conditions, it is straightforward to test if a visualization solution increases the situational awareness of decision makers. This paper describes how CRATE, the cyber range of the Swedish Defence Research Agency (FOI), has been used and will be used to test hypotheses and tools related to security assessments and situational awareness in the cyber security domain. Examples of previous experimental setups are provided and challenges related to research using cyber ranges are discussed.*

## 1.0 INTRODUCTION

The increasing importance of cyber security, in both military and civil contexts, increases the need for knowledge about various issues in the cyber security domain. One important issue cyber situational awareness. However, research within cyber situational awareness is still immature. A recent review of cyber security research on situational awareness by Franke and Brynielsson [1] found that much of the research that has been published on situational awareness is entirely conceptual – more than half of the situational awareness papers lack a non-trivial empirical contribution, even with a very inclusive definition of a non-trivial contribution. Franke and Brynielsson [1] concluded that there seems to be a potential for more research with an empirical basis, for example, by using cyber security exercises as a source of empirical data. An idea previously presented in [2].

This aim of this paper is to further strengthen the arguments for using cyber ranges to support empirical studies on cyber situational awareness. Focus is placed on two subtopics of cyber situational awareness: *system vulnerability assessments* and *intrusion detection*. Both system vulnerability assessments and intrusion detection have received considerable attention by the scientific community in recent decades. However, as for cyber situational awareness in general, very little empirical support is available for the theories and models that have been proposed. This paper argues that it is possible to perform valuable

empirical tests of theories related to both vulnerability assessment theories and intrusion detection using modern cyber ranges. The evidence for this is primarily empirical: examples of successful empirical research carried out in CRATE, the cyber range developed and operated by the Swedish Defence Research Agency (FOI).

The remaining of this paper is structured as follows. Section 2 offers a brief overview of the current state in research related to vulnerability assessments and intrusion detection. Section 3 introduces CRATE. Section 4 presents a number of empirical studies carried out using CRATE. Section 5 discusses future plans for situational awareness experiments in CRATE and some general challenges related to experimentation in cyber ranges.

## 2.0 PREDICTING VULNERABILITY AND INTRUSION DETECTION

Cyber situational awareness can be seen as a subset or specialization of the model of the situational awareness model of Endsley [1]. Endsley's model [3] states that there are three levels of situational awareness: 1) perception of elements in current situation, 2) comprehension of current situation, and 3) projection of future status. There are good reasons to think that an interpretation or adaptation of this theory is necessary for it to fit for cyber security. For example, because threat agents may purposely reduce a humans situational awareness, which is as unlikely scenario in the domains that Endsley's model originally was developed for. Unfortunately, an established interpretation or adaptation of this theory to the cyber security domain is yet to be presented. Two issues in cyber security domain that relates to cyber situational awareness are vulnerability assessments and intrusion detection. Both these have to do with perceiving elements in the current situation, comprehending what these elements mean and making predictions that support cyber security related decisions. The more established theories related to these issues are briefly presented below.

### 2.1 Vulnerability assessments

To assess and rank a system's vulnerabilities requires that potential vulnerabilities are identified. One of the more established ideas within information/computer/network/cyber security is that security is about maintaining or protecting a systems confidentiality, integrity and availability. Consequently, to identify vulnerabilities of a cyber-environment is essentially a matter of how adversaries can compromise these attributes. Adversaries often exploit errors or flaws introduced in the system to compromise cyber security. Because of this, established security models for secure design such as those by Bell-Lapadula [4] and Biba [5] are of limited help to identify vulnerabilities. These models as supposed to help a designer construct a secure system, and not about making predictions of flaws in systems' design, implementation or operation. In practice, the identification of such errors vulnerabilities is done through analysis of the software code (e.g. using static analysis tools and reverse engineering endeavours) or analysis of operational computer networks (e.g. through network scans and penetration testing). And, in practice, there are often many vulnerabilities that pose a potential threat to cyber security and limited resources to manage them. Thus, the identified vulnerabilities need to be prioritized or ranked somehow, e.g. by predicting how severe they are. This can be done on different levels of abstraction.

On a low level of abstraction, the arguably most well-known theory for prioritizing and predicting the severity of individual vulnerabilities is the Common Vulnerability Scoring System (CVSS), developed within the Forum for Incident Response and Security Teams (FIRST). The latest version of this theory [6] posit that a the severity of a vulnerability can be predicted by the attack vector, the attack's complexity, the privileges it requires, if it requires user interaction to be exploited, if the exploits makes it possible to impact other "scopes" (e.g. other machines) than the vulnerable one, and if the exploitation leads to loss of confidentiality, integrity and/or availability. In addition to describing these relationships, the CVSS comes with complex system of equations and constants has been developed to make prediction of the severity level

on a continuous scale 0 to 10. No documented rationale has been provided for these equations. A test comparing the ratings of the previous version of CVSS to the subjective assessments of experts suggests that there is some agreement between experts' perceptions of severity and the predictions of CVSS [7]. However, to this date no published empirical test using data from real systems has assessed the validity of these equations (nor the equations of the previous versions of CVSS).

On a higher level of abstraction, addressing whole networks of computers, attack graphs is an often cited theory/model. Attack graphs use information about system privileges and access control policies in a computer network to predict how attackers can use one or more vulnerabilities together in order to penetrate a computer network [8]. While attack graphs has received considerable attention by scholars (with over 500 articles indexed by the database Scopus mentioning it the title or abstract), the use of attack graphs to predict security is far from an established practice. Furthermore, the only published test on the prediction accuracy (further described in section 4) suggests that the practical utility of today's solutions can be questioned [9]. CySeMoL [10][11], a more complex theory which use attack graphs as well as other quality-related information about the cyber environment to predict vulnerabilities, has only been validated against expert's perception of security.

General notions of cyber security on an even higher level of abstraction is also mentioned in the literature. For example, it is often stated that no system is stronger than its weakest link and that the number of layers of security controls indicate the level of security. However, these notions have not been formalized as theories or operationalized into something testable, even though some steps has been taken towards this (e.g. by [12] and [13]). Thus, while theories has been proposed for predicting cyber security (or vulnerability), there is no theory that has been empirically validated in the sense that it has been shown to predict success or failure to attacks. It should be noted that this validation problem is not new or unknown. When Verendel [14] reviewed the available methods for quantifying security the lack of empirical support for the proposed methods lead to the conclusion that quantified security is a weak hypothesis, and that "for most cases, it is unknown if the methods are valid or not in representing operational security."

As it largely unknown how security should be assessed it is also unknown how vulnerabilities of a network is best visualized or presented to an analyst. Nevertheless, there are several ideas on how this should be done. For example, Chu et al. [15] present a tool that visualize attack graphs.

## 2.2 Intrusion detection

Similar to vulnerability assessments of computer systems, intrusion detection in computer systems is an issue that has been extensively researched. Literary thousands of papers have been published in scholarly journals and workshop or conference proceedings. There is no widely accepted theory on how to detect intrusions or how to classify events as intrusions or not, but most researchers agree that there are two basic approaches for intrusion detection: one based on modelling benign events and one based on modelling malicious events.

Models of benign events has attracted considerable attention among researchers and in scholarly research, often with the rationale that models of benign behaviours are required to detect novel attacks, e.g. zero-day exploits. The idea on models of benign events for intrusion detection was first clearly expressed by Denning in 1987 [16]. Denning essentially proposed that metrics and statistics for measuring deviation from normal behaviour (e.g. in terms of user sessions) should be used to detect intrusions. While this type of model is more frequently addressed in scholarly research, systems used in practice typically use signatures of potentially malicious actions [17]. Thus, most operational intrusion detection systems do not maintain a model of how normal behaviour is, but rather maintain a model of how intrusions look like. It is hard to distinguish any established theories within the research on any of these models. For example, there is no overall agreement concerning the features of what models of normal behaviour should cover to enable detection of threatening anomalies.

It may seem straightforward to identify the attacks that a model of malicious events will detect, simply by looking at the signatures the model looks for. However, in practice, there are similarities between attacks that means that a signature may also raise alarms on attacks it was not written for [18]. It is also difficult to predict which attacks models of benign behaviour detects, e.g. because the coverage is dependent on characteristics of the monitored system is and how the intrusion detection system has been trained. However, for both these types of models, the number of false positives is the major issue, and not the portion of attacks they can detect. Because most cyber environments produce significantly larger number of benign events than malicious events even a low probability that a benign event will be classified as malicious by the intrusion detection system will mean that false positives will outnumber the number of true positives [19].

It is likely that a contributing factor to the issue of false alarms in modern solutions is the lack of data available for testing intrusion detection systems. When intrusion detection systems are tested, this is typically done using the DARPA-dataset [20] (or a derivate of it). This dataset was synthetically produced in the late 1990's, with the type IT-systems used then (e.g. operating systems such as Windows 95), and its validity has been questioned for a number of reasons [21] [22]. Thus, while a large number of alternative solutions are available for intrusion detection and many ideas have been presented, it is hard to know which solution or scheme that is best for different systems and environments. And this uncertainty extends to other theoretical issues relating to intrusion detection, such as how to best model attacks in models of malicious events, which type of threat intelligence that would support the detection, or which features that models of benign events should focus on.

Finally, in practice, operators are known to have an important role in intrusion detection [23][24]. The high portion of false alarms is perhaps the most important reason for this [25]. The operators analyse the alarms and correlates it with other information in order to filter out actual threats, prioritize these and carry out responses. A few experiments has been conducted on intrusion detection operators and their performance: [26] tested the support of visual and textual tools for intrusion detection, [27] tested an operators ability to filter alarms (further described in section 5), [28] tested how the duration of alerts on the screen related to detection, [29] tested how knowledge about the own network helps the operator, and [30] tested how different types of feedback on operators' classifications influence future performance. But, of course, much is still unknown concerning the work of intrusion detection operators and their situational awareness even though a number of experiments has been performed.

# 3.0 SYNTHETIC DATA GENERATION IN CRATE

There exists a large number of cyber ranges and cyber security simulation tools in the world. According to [31], the US Air Force alone had 78 simulators for cyber security issues in 2013, and more than 100 such simulators were used in the US. There are several attempts to describe state-of-the-art in cyber security related simulations (e.g. [32]) and the most significant cyber ranges and simulators available today (e.g. [33]). There are also overviews of the tools that can support cyber ranges and cyber security simulations (e.g. [34]). This section provides a brief overview of CRATE, a cyber range built and maintained by the Swedish Defence Research Agency (FOI). Section 3.1 describes the types of synthetic environments it can be instrumented with and section 3.2 describes how events are generated.

## 3.1 Synthetic environments

Physically, CRATE consist of approximately 750 severs, a number of switches and various auxiliary equipment (e.g. for remote access). During an experiment these servers are instrumented with between five and twenty VirtualBox machines of various types. Instrumentation is straightforward for any type of operating system and application software that VirtualBox can handle, but most applications of CRATE use a mix of Linux and Windows machines. In addition to the typical enterprise software, simplified versions of

social network services, industrial control systems and search engines are available.

Instrumentation is done through a web based graphical user interface, illustrated in Figure 1. This interface allows the experiment designer to manipulate parameters controlling operating system, application software, firewall configurations, network topology, users, windows domains, and more. After the design stage, as set of scripts will deploy machines that matches the design. While the aim is to have the whole instrumentation process fully automated, the experimenter often need to perform some configurations on machines in order to make them fit for the experiment at hand. For example, an experimenter may want to change a web server's configuration to make it vulnerable to some attack or plant files in network folders with user credentials. This can be done by creating a new virtual machine template with these configurations or by interacting with the machine after deployment.

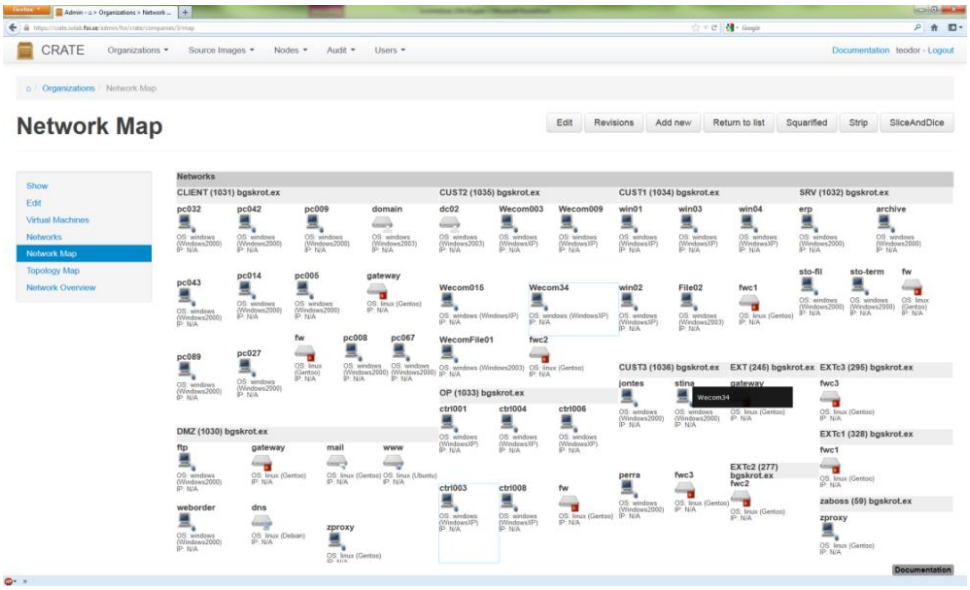


**Figure 1: Screenshot of the web-based configuration tool of CRATE (CrateWeb), illustrating the virtual machines and network topology of a computer network.**

## 3.2 Synthetic event generation

During an experiment, there is typically a need to produce events representing both malicious usage or attacks or normal benign system usage.

For malicious usage, previous experiments in CRATE have relied on manually produced attacks from a set of cyber security experts. This fits well with the notion that "cyber-security is science in the presence of adversaries" [35] and ensures that are relevant and performed by an intelligent adversary. However, involvement of competent cyber security experts is costly and the manual execution of attacks does not always ensure the level of control which is desirable during experimentation. For instance, logs produced by these attackers typically have a time resolution of minutes while intrusion detection alarms have a resolution of seconds or less. Because of this, work has recently started on a tool called SVED (Security Vulnerabilities, Exploitation and Detection), which automates of the use of publicly available penetration testing tools on systems instantiated in CRATE [36]. The tool will allow experimenters and cyber security experts to identify attacks matching the environment in CRATE, plan attacks of any complexity in the form of attack graphs, automatically execute these attacks according to the plan, and record the response from systems and sensors in CRATE. This will reduce experiment costs, offer higher reliability and offer better traceability. Figure 2 shows a screenshot of SVED.

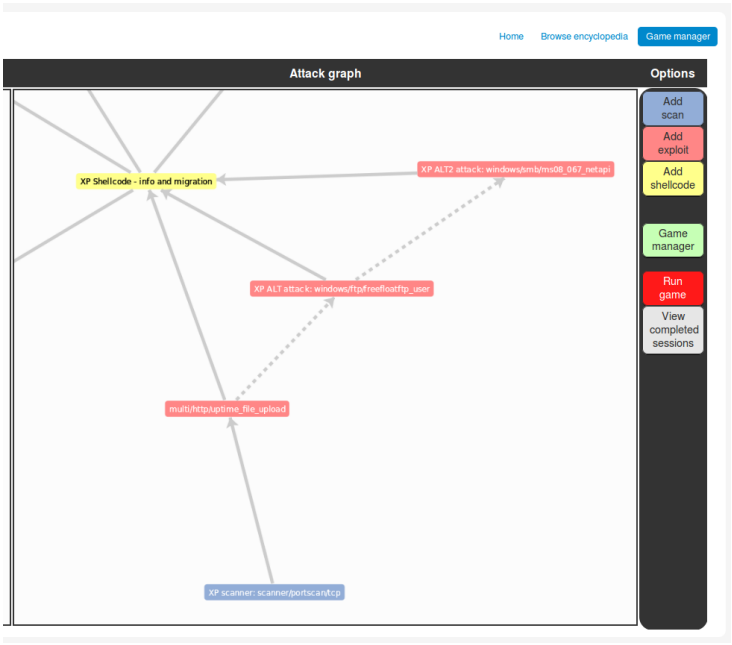


**Figure 2: Screenshot of SVED, a tool for planning and automating attacks in CRATE.**

Normal, benign, user activity can be produced manually by interacting with the desktop or command prompt of machines. But normal user activity can also be generated from scripts. In Linux machines bash commands are used to perform actions at predefined occasions; in Windows machines the automation tool AutoIT [37] is used to perform actions via components of the graphical user interface. Tools are currently available to collect historical user activity from operational systems and use this as a basis for the scripts that automate users. As for generation attacks, further development is planned. In particular, there are plans to implement the user agents using image recognition and displays exported to the virtual machine host rather than AutoIT. This will reduce the scripts' footprint in the experimental environment and make them appear as real users to those in CRATE's cyber environment.

# 4.0 VULNERABILITY ASESSESSMENT AND INTRUSION DETECTION EXPERIMENTS IN CRATE

This section offers some examples of studies related to situational awareness that have been facilitated by CRATE. It should be noted that there are other studies, performed using other experimental platforms, that are similar to these. For instance, the DARPA Information Assurance and Operational Partners in Experimentation Programs performed similar experiments around year 2000 [38]. This section is limited to a subset of research performed using CRATE and presents research on the accuracy of vulnerability scanners (section 4.1), host vulnerability metrics (section 4.2) attack graph assessments (section 4.3), intrusion detection operators' filtering capability (section 4.4), and filtering IDS alerts with situational data (section 4.5).

## 4.1 The accuracy of vulnerability scanners

Many cyber security management tools and solutions assumes or relies upon the availability of information concerning publicly known vulnerabilities in the own system. In many cases (e.g. as with attack graph tools [39]), it is assumed that this information is provided by vulnerability scanners that probe the systems in the network to identify the software and the vulnerabilities that systems have. Thus, the accuracy of vulnerability

scanners is important for cyber situational awareness.

In 2010, domain experts instrumented CRATE with a network representing a small industrial control system for an exercise. This network was used to tests seven vulnerability scanners [40] [41]. As the network was non-operational and located in a cyber range it was possible to repeatedly run scans on it; as it was well-documented and available further information gathering it was possible identify the ground truth concerning existing vulnerabilities; as it had been created by domain experts it could be assumed that the software and vulnerabilities in it was representable for the domain.

The tests showed, among other things, that the scanners performed better on Window machines than Linux machines [40], that they performed significantly better when they are provided user credentials for the machines [40], that about half of the publicly known vulnerabilities in a computer network of this sort was detected by a scanner [40], and that about two thirds of the vulnerabilities would be remediated if their recommendations were followed [41].

## 4.2 Host vulnerability metrics

In some cases it makes sense to abstract individual software vulnerabilities and assess the vulnerability and status of hosts (an operating system and its hosted software). For example, it is common to focus on hosts rather than specific software products in cyber security visualization [42]. However, hosts typically have multiple products with vulnerabilities of various sorts and severity, which does not fit with visualization of hosts. A way to handle this is to score the aggregate vulnerability information to a host level metric that represents the overall vulnerability of different hosts.

Several proposals have been presented in the literature for how such aggregation should be done. The estimates produced by these proposals were tested using data collected from the Baltic Cyber Shield, an exercise performed in CRATE during 2010 [43]. The time it took for the attackers in the exercise to compromise hosts and detailed vulnerability data on these hosts were used to test six hypotheses concerning vulnerability rating methods. The results indicated that the more information a method used as input to the aggregated value, the more accurate it was. However, the correlation between the rating of the best method and the time-to-compromise in the exercise was less than 0.3, suggesting that they offer a weak support for gaining situational awareness.

## 4.3 Attack graph assessments

An often cited method for rating host vulnerability, which was not addressed in the study described in section 4.2, is attack graphs. Attack graphs fuse vulnerability information with information on their users, the users' privileges in the network, and network access control restrictions (e.g. firewall rules). On a network level, attack graphs predict the steps attackers can take in the network by exploiting vulnerabilities to elevate their privileges. On a host level they provide information about the privileges that attackers can obtain on hosts given starting points. Thus, they can be seen as a more network vulnerability assessment method, or a more complex method of assessing host vulnerability (i.e., as the methods in section 4.2).

The accuracy of a commonly cited attack graph tool was tested in CRATE using experimental data collected in 2012 [9]. During this experiment, the Computer Emergency Response Team of the Swedish Armed Forces and cyber security researchers at the Swedish Defence Research Agency (FOI) performed a large number of attacks to penetrate 199 machines distributed over a number of computer networks. Their attempts were logged and subsequently compared to the predictions made by the attack graph tool. The predictions by the attack graph tool matched poorly to the performance and choices of the attackers. In fact, attackers were more successful at compromising hosts the tool marked as unreachable than they were at compromising hosts the tool marked as reachable (29% vs. 8%). In addition, the attack graph tool produced a

very complex output, with almost 500 000 attack paths of 60 attacker steps[1] or less. Analysis of the predictions suggests that this inaccuracy is due to both the inaccuracy of the vulnerability scanner used to provide input to the attack graph tool and the assumptions made concerning privileges by the attack graph tool.

### 4.4 Intrusion detection operators' filtering capability

As noted above, most operational systems for detecting attacks depend heavily on the operator who receives the alerts. The operator's performance is important for the intrusion detection capability, and a number of human factors can be expected to be of relevance [44]. One of the operator's primary tasks is to review the alerts produced by intrusion detection sensors to separate the alerts that come of real threats from the alerts that come of normal benign activity. The ability of an operator in doing so was tested in CRATE during an experiment conducted during 2011.

In the experiment [27], the operator monitored a system he was familiar with using a sensor he had manually tuned. The system was attacked by a team from the Computer Emergency Response Team of the Swedish Armed Forces and the operator received alerts in real time. Based on these alerts, and other information he could collect by interacting with the monitored systems, the operator's had the task of writing down the attacks he believed took place. This log and the output of the tuned intrusion detection sensor was subsequently compared to the log of the attackers. This comparison showed that the operator did a good job at filtering out the actual attacks. The intrusion detection system sensor raised alerts for 69% of the attacks, but only 11% of the alerts could be traced to actions taken by the attackers. The intrusion detection operator raised an alert for 58% of the attacks, and 57% of these alerts could be traced to actions taken by the attackers. Interviews with the operator suggested that the most useful information in this task was, other the alerts produced by the intrusion detection system, his knowledge about security in general, the network that was attacked, and his knowledge about the attackers in the experiment.

### 4.5 Filtering IDS alerts with situational data

Intrusion detection system operators typically fuse alert information with other information about the monitored network, something that is exemplified by the results described in section 4.4. Automating this data fusion task would certainly be desirable. To investigate how this should be done, a test was carried out using the data from 2012 which is described in section 4.3. This test examined whether accuracy could be improved if alerts were correlated with information provided by a vulnerability scanner [17]. In other words, if filters based on computer system information (e.g. vulnerabilities, vulnerabilities properties and open ports) could help reduce false alarms, but at the same time maintain most of the true alerts. Unfortunately, none of the 18 tested filters, nor any of combination of them, were effective. The most effective filter managed to nearly reduce the number of false alarms by half, but at the cost of also reducing the number true alerts by half. One possible interpretation of this is that better vulnerability information is needed than what today's vulnerability scanners have to offer; another possible interpretation is that more advanced data fusion is required; a third possible interpretation is that this task is difficult to automate and should be left to a human operator; and a fourth possible interpretation is that the experiment must have been unrealistic and lack ecological validity.

## 5. DISCUSSION AND FUTURE WORK

In the cyber security domain, a number of conceptual analyses has been performed of cyber situational awareness, e.g. [45]. However, despite the large interest for the topic, no comprehensive and well-formulated theories on situational awareness have been developed yet. Such a theory could, for example, adapt the theory of Endsley [3] and explain which properties of a system an intrusion detection system or intrusion

[1] An example step is running the exploit MS08-067 (CVE-2008-4250) against the SMB service on a Windows XP system.

detection operator will need to comprehend in order to accurately predict which assets that are compromised or will be compromised in the immediate future. Good ideas already exists on both what needs to be perceived, comprehended, and projected in cyber security management. For example, interviews and observations of intrusion detection system operators and administrators indicate that it is important to keep track of the vulnerabilities in the network, the normal behaviour of machines in the network and the current behaviour of machines [23][24][27]. It is reasonable to assume that this information will be useful for making decisions of relevance to cyber security. These decisions can, for example, be the of the types that cyber security operations centers make concerning interventions for ongoing attacks: blocking activity, deactivating user accounts, and informing some other party [46]. Or, it can be the types of decisions mentioned in cyber security incident handling guidelines. For example, NIST's incident handling guidelines [47] stresses the need to prioritise incidents and explains that decisions related to containment strategies needs to be made.

Thus, while both established practice and reasonable ideas exists on cyber security situational awareness, a comprehensive theory or model over how various variables fit together is missing. One possible reason for why comprehensive theories on cyber situational awareness is missing is that cyber security researchers, historically, have had a hard time obtaining relevant data. Because, without data to test theories, theory generation is of little value to science. Cyber ranges like CRATE makes it possible to perform controlled experiments on a wide range of issues related to cyber security, including issues related to situational awareness. Thus, they may help cyber security researchers to develop useful and comprehensive theories on cyber situational awareness that, after rigorous testing, can become widely accepted. CRATE will be further developed to facilitate realistic training and tests of cyber situational awareness, in particular by improving the generation of background data and automating the execution of attack tools.

## 5.0 REFERENCES

[1] U. Franke and J. Brynielsson, "Cyber situational awareness – A systematic review of the literature," *Comput. Secur.*, vol. 46, pp. 18–31, Oct. 2014.

[2] T. Sommestad and J. Hallberg, "Cyber security exercises and competitions as a platform for cyber security experiments," in *NordSec*, 2012.

[3] M. R. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems," *Hum. Factors*, vol. 37, no. 1, pp. 32–64, 1995.

[4] D. E. Bell, "Looking Back at the Bell-La Padula Model," in *21st Annual Computer Security Applications Conference (ACSAC'05)*, 2005, pp. 337–351.

[5] K. J. Biba, "Integrity considerations for secure computer systems," Storming Media, 1977.

[6] FIRST, "Common Vulnerability Scoring System v3.0: Specification Document," https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf, 2015.

[7] H. Holm and K. K. Afridi, "An expert-based investigation of the Common Vulnerability Scoring System," *Comput. Secur.*, vol. 53, pp. 18–30, Sep. 2015.

[8] T. Heberlein, M. Bishop, E. Ceesay, M. Danforth, C. Senthilkumar, and T. Stallard, "A Taxonomy for Comparing Attack-Graph Approaches," *netsq.com*, 2004. [Online]. Available: http://netsq.com/Documents/AttackGraphPaper.pdf. [Accessed: 28-Jun-2010].

[9] T. Sommestad and F. Sandström, "An empirical test of the accuracy of an attack graph analysis tool," *Inf. Comput. Secur.*, vol. 23, no. 5, pp. 516–531, Nov. 2015.

[10] T. Sommestad, M. Ekstedt, and H. Holm, "The Cyber Security Modeling Language: A Tool for Assessing the Vulnerability of Enterprise System Architectures," *IEEE Syst. J.*, no. 99, pp. 1–1, 2013.

[11] T. Sommestad, "A framework and theory for cyber security assessments," Royal Institute of Technology (KTH), 2012.

[12] R. Nunes-Vaz, S. Lord, and J. Ciuk, "A More Rigorous Framework for Security-in-Depth," *J. Appl. Secur. Res.*, vol. 6, no. January 2015, pp. 372–393, 2011.

[13] M. Coole, J. Corkill, and A. Woodward, "Defence in Depth , Protection in Depth and Security in Depth : a Comparative Analysis Towards a Common Usage Language," pp. 21–23, 2007.
[14] V. Verendel, "Quantified security is a weak hypothesis," in *Proceedings of the 2009 workshop on New security paradigms workshop - NSPW '09*, 2009, pp. 37–49.
[15] M. Chu, K. Ingols, R. Lippmann, S. Webster, and S. Boyer, "Visualizing attack graphs, reachability, and trust relationships with NAVIGATOR," in *Proceedings of the Seventh International Symposium on Visualization for Cyber Security*, 2010, pp. 22–33.
[16] D. E. Denning, "An Intrusion-Detection Model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.
[17] T. Sommestad and U. Franke, "A test of intrusion alert filtering based on network information," *Secur. Commun. Networks*, vol. 8, no. 3, pp. 2291–2301, Sep. 2015.
[18] H. Holm, "Signature Based Intrusion Detection for Zero-Day Attacks: (Not) A Closed Chapter?," in *Hawaii International Conference on System Sciences*, 2014.
[19] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 3, pp. 186–205, Aug. 2000.
[20] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA on-line intrusion detection evaluation," *Comput. Networks*, vol. 34, 2000.
[21] J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, Nov. 2000.
[22] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, no. Cisda, pp. 1–6.
[23] R. Werlinger, K. Muldner, K. Hawkey, and K. Beznosov, "Preparation, detection, and analysis: the diagnostic work of IT security incident response," *Inf. Manag. Comput. Secur.*, vol. 18, no. 1, pp. 26–42, 2010.
[24] J. R. Goodall, W. G. Lutters, and A. Komlodi, "I know my network: collaboration and expertise in intrusion detection," in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 2004, pp. 342–345.
[25] R. Werlinger, K. Hawkey, K. Muldner, P. Jaferian, and K. Beznosov, "The challenges of using an intrusion detection system: is it worth the effort?," in *SOUPS '08 Proceedings of the 4th symposium on Usable privacy and security*, 2008, no. 1, pp. 107–118.
[26] J. R. Goodall, "An Evaluation of Visual and Textual Network Analysis Tools," *Inf. Vis.*, vol. 10, no. 2, pp. 145–157, Apr. 2011.
[27] T. Sommestad and A. Hunstad, "Intrusion detection and the role of the system administrator," *Inf. Manag. Comput. Secur.*, vol. 21, no. 1, pp. 30–40, 2013.
[28] B. D. Sawyer, V. S. Finomore, G. J. Funke, V. F. Mancuso, M. E. Funke, G. Matthews, and J. S. Warm, "Cyber Vigilance: Effects of Signal Probability and Event Rate," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 58, no. 1, pp. 1771–1775, Sep. 2014.
[29] N. Ben-Asher and C. Gonzalez, "Effects of cyber security knowledge on attack detection," *Comput. Human Behav.*, vol. 48, pp. 51–61, Jul. 2015.
[30] N. Ben-Asher and C. Gonzalez, "Training for the unknown : The role of feedback and similarity in detecting zero- day attacks," in *6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences*, 2015, no. August.
[31] S. D. Harwell and C. M. Gore, "Synthetic Cyber Environments for Training and Exercising Cyberspace Operations," *M&S J.*, 2013.
[32] S. Leblanc and A. Partington, "An overview of cyber attack and computer network operations simulation," in *Proceedings of the 2011 Military Modeling & Simulation Symposium*, 2011, pp. 92–100.
[33] J. Davis and S. Magrath, "A Survey of Cyber Ranges and Testbeds (DSTO-GD-0771)," Edinburgh South Australia, Australia, 2013.
[34] C. Siaterlis and M. Masera, "A survey of software tools for the creation of networked testbeds,"

*Int. J. Adv. Secur.*, vol. 3, no. 1, pp. 1–12, 2010.

[35] JASON, "Science of Cyber-Security," McLean, Virginia, 2010.

[36] H. Holm and T. Sommestad, "SVED: Scanning, Vulnerabilities, Exploits and Detection," in *MILCOM 2016*, 2016.

[37] Jonathan Bennett AutoIt Consulting Ltd, "AutoIt Script Editor," 2015. [Online]. Available: http://www.autoitscript.com/site/autoit/. [Accessed: 01-Jul-2016].

[38] D. Levin, "Lessons learned in using live red teams in IA experiments," in *Proceedings DARPA Information Survivability Conference and Exposition*, 2003, vol. 1, pp. 110–119.

[39] O. M. Sheyner, "Scenario graphs and attack graphs," Carnegie Mellon University, 2004.

[40] H. Holm, T. Sommestad, J. Almroth, and M. Persson, "A quantitative evaluation of vulnerability scanning," *Inf. Manag. Comput. Secur.*, vol. 19, no. 4, pp. 231–247, 2011.

[41] H. Holm, "Performance of automated network vulnerability scanning at remediating security issues," *Comput. Secur.*, vol. 31, no. 2, pp. 164–175, Mar. 2012.

[42] H. Shiravi, A. Shiravi, and A. a. Ghorbani, "A survey of visualization systems for network security," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 8, pp. 1313–1329, 2012.

[43] H. Holm, M. Ekstedt, and D. Andersson, "Empirical Analysis of System-Level Vulnerability Metrics through Actual Attacks," *IEEE Trans. Dependable Secur. Comput.*, vol. 9, no. 6, pp. 825–837, Nov. 2012.

[44] P. Lif and T. Sommestad, "Human factors related to the performance of intrusion detection operators," in *Human Aspects of Information Security, Privacy, and Trust*, 2015.

[45] G. P. Tadda and J. S. Salerno, "Overview of Cyber Situation Awareness," in *Advances in Information Security*, vol. 46, 2010, pp. 15–35.

[46] C. Zimmerman, *Ten Strategies of a World-Class Cybersecurity Operations Center*. Bedford, MA: The MITRE Corporation, 2014.

[47] P. Cichonski and K. Scarfone, "Computer Security Incident Handling Guide Recommendations of the National Institute of Standards and Technology (SP 800-61)," Gaithersburg, MD, USA, 2012.