The 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
18-21 September 2019, Metz, France

A User Study of the Visualization-Assisted Evaluation and Management of Network Security Detection Events and Policies

Volker Ahlers[1] , Bastian Hellmann[1] , Gabi Dreo Rodosek[2]
[1] University of Applied Sciences and Arts Hannover,
Ricklinger Stadtweg 120, 30459 Hannover, Germany, volker.ahlers@hs-hannover.de
[2] Bundeswehr University München,
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany, gabi.dreo@unibw.de

***Abstract* — Intrusion detection systems and other network security components detect security-relevant events based on policies consisting of rules. If an event turns out as a false alarm, the corresponding policy has to be adjusted in order to reduce the number of false positives. Modified policies, however, need to be tested before going into productive use. We present a visual analysis tool for the evaluation of security events and related policies which integrates data from different sources using the IF-MAP specification and provides a "what-if" simulation for testing modified policies on past network dynamics. In this paper, we will describe the design and outcome of a user study that will help us to evaluate our visual analysis tool.**

Keywords* — *Network security, User interfaces, Visualization, Information visualization

I. Introduction

Today's computer networks are highly dynamic, with mobile devices being temporarily attached, users logging on and off, and applications and services being updated at frequent, but non-regular intervals. Together with growing network sizes this proposes a challenge to the monitoring of network activity and the detection of security threats or attacks.

Network security detection systems rely on data from many different sources: firewalls, network access control (NAC) components, vulnerability scanners, intrusion detection systems (IDS), etc. Usually several different detection systems are used in parallel, as no common data model for these different sensors and actors exists. Furthermore the amount of data produced and security events triggered by detection systems as well as the network dynamics make it difficult for network administrators to decide whether an event is critical, of minor significance, or even a false positive.

Several visualization approaches have been proposed to support the analyst in monitoring and inspecting the security relevant data, e. g., visualizing relations between hosts, users, and applications [1], [2], visualizing the information from log files of different security components like Snort [3], or visualizing attack graphs in order to specifically analyze intrusion detection events [4]. Some approaches provide dashboard-like visualization systems to identify possible interesting spots of data [5], [6].

Our approach tries to integrate these different approaches to visualize specific parts of the data into one integrated data model and visualization, that allows to use arbitrary data from extensible sources to then perform both monitoring and in-depth analysis.

Security detection systems like IDS rely on rules and policies describing potential vulnerability or attack patterns, which have to be configured for the specific network in order to balance the detection rate against the false positive rate. Different approaches exist to automate or assist the configuration with varying amount of manual or automatic means, e. g., [7]–[10].

In our approach we try to integrate the process of adjusting the configuration of detection systems based on a previous analysis and to simulate and afterwards evaluate the outcome of the changes. This simulation can be done with historical data, i.e. real events that happened in the network, in order to re-evaluate them with an adapted configuration of the detection systems.

II. Visual Analysis and "What-If" Simulation

In our previous work we presented a visual network analysis framework which allows the integration of data from different data sources as well as the visualization of historic data and network changes between two time instances [11] (and references therein). The software architecture and the data model rely on the Interface for Metadata Access Points (IF-MAP) specification [12]. IF-MAP defines a graph-based data model including physical and logical network components via identifiers and metadata. A MAP server collects data from different sources and allows clients to subscribe to the information. In Fig. 1 the basic architecture of the framework is

This work was financially supported by the German Federal Ministry of Education and Research (BMBF), projects VisITMeta (grant no. 17PNT032) and SIMU (grant no. 16KIS0045).

DOI of the original publication: https://doi.org/10.1109/IDAACS.2019.8924439

shown. The capabilities of the framework can be extended by adding additional IF-MAP clients, that can act as sensors and actors. These clients can either attach existing software and hardware to the IF-MAP environment or integrate completely new methods of analysis to the framework.

The framework includes the possibility to visualize the network state together with a graphical representation of the policy which triggered a security event. An example is shown in Fig. 2. Within the graphical representation, the connection between the security event, the responsible policy elements and the processed input for the detection are explicitly shown, minimizing additional manual aggregation of these different sources of information. In a further step our framework allows to modify the rules of the detection policies and simulate the policy evaluation with the past network dynamics [11], thus providing a "what-if" functionality that allows the testing of policy changes, e. g., in order to reduce the number of false positive events.

The policies and their rules as well as detected incidents are therefore also integrated as IF-MAP data. This allows to map the incident itself to both the data that was used for the detection process as well as the actual parts of the policy that were used. This allows for the user to investigate the incidents cause and plausibility, as all further information attached to the actual input can be taken into account easily and directly. The visual analysis tool presents means to manipulate the rules directly. The taken changes are then implemented via standard IF-MAP data exchange mechanisms. The "what-if" functionality works by adding a dry-run processing mode to the used detection components, where all relevant input to a previously detected incident is evaluated with the changed policy, concluding in a evaluation result that can then be compared to the original result.

The software framework, including the IF-MAP server irond, the two correlation engines irondetect and irongpm, the visualization component VisITMeta, as well as a set of IF-MAP clients that connect existing software—e.g. OpenVAS, nmap—, is available as a suite of open-source projects on Github [13].

III. Design of the User Study

We designed a user study to evaluate our solution approach and the prototypical implementation. The group of test subjects includes persons with various backgrounds: undergraduate and graduate students who have attended lectures in computer networks (some also in IT security), Ph.D. students and scientific personnel from different research projects, and network administrators.

The study consists of three different scenarios along with a list of tasks that have to be performed by using several of the software components depicted in the framework, with the visual support component as the main tool for interaction. The test subjects are asked to

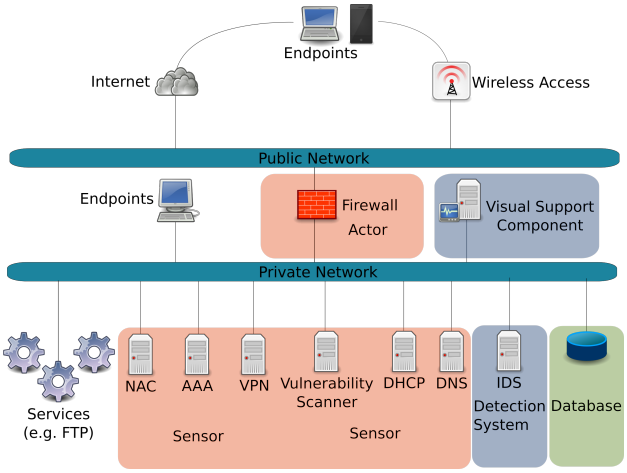

Figure 1. The framework architecture including sensors and actors (red), analysis components (blue), and databases (green).

answer the tasks on paper, including a questionnaire based on a German translation of the System Usability Scale by John Brooke et al. [14], with adaptations in respect to the prototypical nature of the software itself and the primary focus on the general handling with the data and the tasks at hand for the first scenario. For an additional qualitative evaluation, a questionnaire at the end of the study document allows the participants to express their personal opinions on the software and the benefits or drawbacks of the proposed methods for completing the tasks.

The first scenario introduces the test subjects to a small network depicting a simplified enterprise network consisting of infrastructure components, some business related services (e.g. a database or a web-server) as well as some endpoints used by users within the network. The tasks of the first scenario asks the subjects to answer different questions on the state of the network and its components by aggregating the information presented via the visualization component VisITMeta and the underlying data model. In comparison, the test subjects are then asked to perform similar tasks and answer similar questions, but this time without the visual representation via an integrated data model. The test subjects will have to use and aggregate the required information by using a set of log files, configuration files and reports from the same components that were used before. The sequence of these two variations—first working with VisITMeta and then based on log files and vice versa—will be changed randomly for every participant. This aims at minimizing the learn effect between the two variants; a participant completely inexperienced with network analysis could possibly perform better at the second variant, as the participant may have learned about the specific keywords and connections during working on the first variant.

In the second scenario, the subjects have to perform

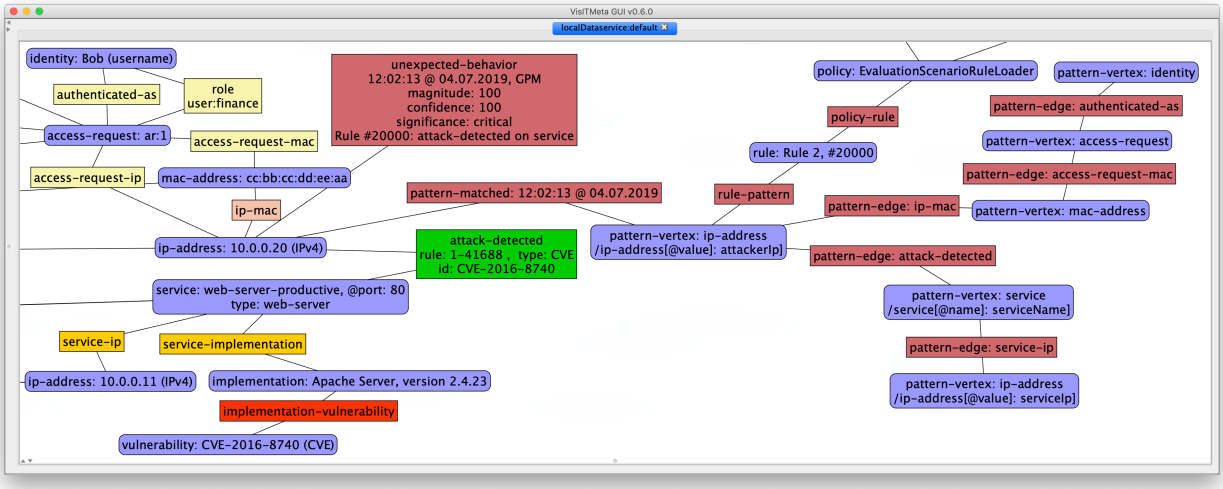

Figure 2. Screenshot of VisITMeta with an example graph showing the network state (right) together with the policy that triggered a security event (left), where the actual detection result is represented via the "pattern-matched" node (middle); some UI elements were hidden for this screenshot.

an analysis of a situation where a given policy of a detection component with multiple rules has detected several security events in order to determine, which rule did work correctly in regard to what it should have detected. This includes the analysis of the connections between the events, their corresponding rules from the policy, and the input data that was used. The scenario therefore depicts a situation where two different attackers try to gain access to a service within the network, which a specific vulnerability was identified for by the infrastructure components. One attacker uses an attack method—identified by a running Snort component—that utilizes the proper common vulnerability and exposure (CVE) id related to the vulnerability, while a second attacker uses a different attack method with a CVE id that does not match any present vulnerabilities. The policy of the detection system in use consists of two similar rules, one of which is defined specifically to detect attacks to existent vulnerabilities—i.e. CVE id of the attack matching the CVE id of an identified vulnerability—, whereas the second rule does not check if the used vulnerability is even existing. The test subjects will then have to retrace the detection process and the involved rules to determine the usefulness of both rules.

The third scenario will ask the test subjects to adapt a given configuration based on finding an error within a rule together with the what-if simulation with a previously false registered event and to evaluate their changes based on the simulation results. The test subjects will have to identify the correct policy element that has to be changed, perform that change and trigger the "what-if" simulation afterwards, comparing the result of before and after the changes to the policy.

The three scenarios are meant to build upon one another, introducing the test subjects to the idea of the integrated data model as well as the visualization component as their main interaction point, and both expand the scope of what can and has to be done within the tasks and simultaneously maintaining the same mental image for the representation of the network and its processes.

All tasks will be done with simulated data in respect to data from sensors and therefore be identical for every test subject. The data will mimic actual sensors and actors of the framework that adapt different real world systems like OpenVAS, Snort or FreeRADIUS. To simulate the data, i.e. the actual IF-MAP graph data, a simulation environment that allows to mimic the real-life IF-MAP clients and their output is used, while running a real MAP server, the detection components and the visualization component.

The study concludes with a set of specific questions in regard to the participants opinion about working on the tasks before, like working with the VisITMeta software in contrast to using log files and configuration data, or if other parts of the concept—consistent visualization between scenarios or integration of rule data—were helpful.

The participants also have the possibility to answer with free text in regard to their experienced difficulties while working on the tasks of the study, functionality they might have missed for answering the tasks and the possibility to express general comments on the user study.

A small questionnaire finally asks about their personal background—highest education, current employment or education—as well as their previous experience with the analysis of log output, software and hardware for network security tasks in general, and whether they had prior

knowledge on the IF-MAP specification or the VisITMeta software.

IV. Execution of the Study

The user study was successfully conducted during April and May 2019 with in total 12 participants. Regarding highest education, 6 of them had finished their bachelor's degree in Applied Computer Science, 4 had finished their master's degree in Applied Computer Science, and 2 did have the qualification for university admission.

In terms of their current employment or education, 6 participants were studying for their master's degree in Applied Computer Science while 1 was studying for their bachelor's degree. 2 participants were working as administrators for a university IT support team; one of them was also at the same time one of the master's students. 2 participants were currently working as research assistants, with one of them actively working on their doctoral thesis. 1 participant was working at a company as an IT security consultant.

According to the questionnaire on their prior knowledge and experience, the group of participants was moderately experienced in the analysis of log output—based on an average score of 2.3 in the corresponding question, with values ranging from 0 meaning no experience up to 4 for very experienced—as well as the usage of software and hardware for network security—also based on a average value of 2.3. Only 2 participants stated that they had no previous experience in log analysis (1 participant) or network security (1 participant). In regard to already being experienced with either the IF-MAP specification or the VisITMeta software, 5 already had knowledge of IF-MAP—average value of 0.6—and 3 of VisITMeta—average value of 0.6.

As mentioned in the previous section, the study was conducted in 2 variants, where the sequence of the first two tasks was changed, distributed randomly and evenly between all participants—6 used variant A of the study while the other 6 used variant B.

During the study, the participants were able to ask the leader of the study on any problems they might have with the tasks or software. On average, the duration of a single study took about 1 hour and 45 minutes, while the fastest and slowest participants took 1 hour and 26 minutes and 2 hours and 10 minutes, respectively. The study was designed to consist of rather complex scenarios, including only a short introduction into the data provided for the tasks and therefore relied on the participants to investigate and become familiar with the software, the data and thus the network represented by the data, which resulted in a high average duration.

V. Results of the Study

The system usability score was determined for both variants of the first scenario. In average, the first scenario with the basis for completing the tasks being logging output and configuration files was a score of 40.6, with a minimum of 5.0 and a maximum of 90.0, the median being 26.3.

For the second variant, i.e. working on similar tasks but with the usage of VisITMeta and an IF-MAP database, the average SUS score was 69.8, with the minimum being 20.0 and the maximum score at 95.0. The median SUS score was 73.8.

Comparing the two scores, the second variant with VisITMeta was rated better that the first variant. The highest values for the first variant were mainly given by participants with medium to high previous knowledge and experience with logging output analysis and network security in general; most of the participants with low or no previous knowledge gave a higher SUS score for the second variant.

A. Errors and Results of the Actual Tasks

The results of the actual tasks of the user study as noted by the participants allow to determine whether they could answer them correctly.

Given the answers to the first scenario while using logging output and configuration files, they show that the participants could answer questions on what devices and services were present in the network mostly correctly. When answers were incomplete, it seems that the files provided were not checked thoroughly; one participant actually skipped half of the tasks due to the amount of text to examine.

The second variant of the first scenario, i.e. using VisITMeta to solve similar tasks, was also mostly answered correctly. When answers lacked information, it seems due to either a slightly imprecise wording of the questions or not identifying the IF-MAP node having the questioned information.

In the second scenario, the participants had to analyze a given rule system in combination with some detection events. The main errors in regard to the given answers were that of the three distinct detection events oftentimes only two were noted, which could be due to a possible overlapping or occlusion in the graphical representation. Two participants were not able to identify the attacker and/or the target of the events. A task to identify the rule of a given set of two rules which would likely produce more false positive results was answered correctly in 10 of 12 cases, with the rest being unanswered.

The third scenario involved identifying a specific node in the data, edit it and perform a what-if evaluation with subsequent comparison of the results. This was done mostly correctly, were in only one case no answer was given and in several cases the actual node was not given, but the one directly one layer higher in the rule hierarchy.

In general, of the 14 tasks in total per user study—excluding one additional task where the participants only had to follow some instruction and thus no answer or feedback had to be given—and among the 12 participants—

making a total 168 tasks in all studies combined—, 5 were unanswered (3,0%), another 5 were incorrectly (3,0%), 51 partially correctly (30,4%) and 107 (about 63,7%) correctly answered.

B. Answers Given in the Questionnaire

The last part of the user study results are given by the answers of the participants to the enclosing questionnaire in the study document.

The first part consists of 4 questions, where the participants had to state their agreement to a statement on a range from from 0—meaning no agreement—up to 4—meaning full agreement.

The first of these rating questions asked whether the tasks of the first scenario could better be worked on by using the VisITMeta solution in contrast to working with logging output and configuration files. The average rating was 3.3, with minimum at 0 and maximum at 4. As the median rating was also 4, this indicates that most participants do find working with VisITMeta—at least in regard to the first scenario of network monitoring—to be easier.

The second questions asked whether the consistent visualization between the different scenarios when using the VisITMeta software and the IF-MAP database was helpful. Similar to the previous questions, the average rating was 3.3, with a minimum of 2 and a maximum of 4. With a median rating of 3, this also indicates that the participants rather agree with the statement that consistent visualization was helpful.

The third question asked the participants whether they were aware of the fact, that the information presented by the VisITMeta software was gathered by different components in the first place. The average rating was 2.6, with a minimum of 1 and maximum of 4, while the median rating being 3. This represents that the participants are slightly aware of the nature of the information; it was not necessary to be aware of that for answering the questions, so this may be a hint that the interconnection of data reduced the need to know the origin of that data. As long as the connecting edges support logical names, the bridge between actual distinct information sources blurs.

In the fourth question the participants had to state their agreement to the statement that the integration of the rules of a detection component into the general database as well as the interconnection between them was helpful. Here the average rating was 2.8, the minimum and maximum were 1 and 4 respectively, and the median rating was 3. On average, the participants did agree with the integration of the rule system being helpful. Perhaps the agreement could have been higher if the participants had been given more time to get to know the rule structures and the connection to the data their corresponding detection systems used.

C. Participant Feedback

The last questions regarding the content of the user study were giving the participants the possibility to express themselves via three free text questions. One of them aimed at the problems that had occurred while gathering all information that were necessary to answer the tasks. The most problems were caused by the layout of the nodes by and in VisITMeta, as associated information could be positioned apart from each other and would have to be identified by following all outgoing edges. The navigation from one node to another via the connecting edges was also mentioned as a problem, especially with overlapping edges of other nodes. Part of the layout problems was the possible occlusion of some nodes by others.

Problems rooted in the nature of the data were mainly expressed in regard to the rule systems of the used detection systems irongpm and irondetect. Also, some participants said that their missing experience with VisITMeta and IF-MAP graphs, i.e. the concrete data types, specific nodes and structure of the data was a cause of problems or delays when answering the questions.

The second question aimed at giving direct feedback on missing functionality in VisITMeta to successfully or at least better work on the scenarios of this study. Similar to the answers to the previous question, the wish for a better layout of the information—including less overlapping, occlusion and positioning of associated nodes over too much space—was mentioned.

Additionally, a functionality to group multiple nodes and be able to collapse and expand them as needed as well as a functionality to directly filter the data were asked for.

Another repeatedly mentioned missing functionality regarded the emphasis and highlighting of information, especially the outgoing edges of a selected node to be able to find the neighbors more easily.

One feature idea was a combination of both the log output or configuration data and the VisITMeta approach; selecting information in either one could be used to select all occurrences in the other representation, e.g. selecting an IP address in the DHCP servers configuration or lease file would select the corresponding IF-MAP node for further analysis.

Some quality-of-life features like adding a shortcut to directly enter the search field or an undo-functionality were also described, as well as the availability of some sort of help functions and a tutorial.

The last free text questions allowed to state general comments on the study; one participant explicitly noted that previous knowledge in networks and network security was required to answer the questions of the study

In addition to the feedback of the participants gathered by their answers and notes in the study document, oral comments from during and after the study were also noted. These also included wishes for additional or differ-

ent functionality of the VisITMeta software, such as different behavior when dragging and dropping nodes in the visualization, or the question whether their exists another representation of the rule system of the irongpm detection system. Also, during observation of the participants while using the VisITMeta software, some additional bugs and unwanted behavior could be noted. A wide range of users without experience with the software using it the first time, this user study also qualified as a kind of beta test of the software itself.

VI. Assessment and Discussion of the Results

The outcome of the user study in general shows that our approach for the visual analysis of network security is suitable and usable even by people new to the software.

Among other things, this is indicated by the difference in the system usability scores of both used variants. In average, a participant rated the VisITMeta variant at about 29.2 points better than the non-VisITMeta variant, in spite of the problems and missing features of the prototypical implementation.

Although almost all participants used the VisITMeta software for the first time and were also new to both the IF-MAP database and the different rule systems of components like irongpm or irondetect, they mostly executed the tasks of the study successfully and correctly.

All rating questions from the questionnaire—excluding the third—showed that at least some participants did not agree with the helpfulness of our approach respectively were rather neutral on the helpfulness. This was not mainly stated by the participants with high experience in network security and analysis, but found at different experience levels.

The identified problems within the software VisITMeta are mainly located in the areas of node positioning, information highlighting and filtering. These are in general considered in our approach and sometimes already noted as possible enhancements to the software, as it being a prototype and thus not including all conceptual parts in detail.

VII. Future Work

Based on the results of the user study the user interface and the visualization approaches will be further enhanced by implementing missing features and working on the problems identified by the user study.

Afterwards it would be possible to conduct a second user study, partially designed to check whether these problems could be minimized. This second user study could also include a more detailed introduction and preparation of the participants in regard to the software and the utilized data types.

One open task is the scalability of the system for large networks. In terms of the visualization, different techniques to reduce the amount of data shown to the user could be investigated. Such techniques could include a level of detail-mechanism, where the amount of nodes shown on screen depends on the zoom scale; groups of associated nodes would be collapsed into meta-nodes at a specific zoom distance.

Besides that further analysis components are developed or adopted to the IF-MAP specification, thus enhancing the environment of the prototypical implementation, e. g., using machine learning methods for the creation of detection rules.

Acknowledgment

The fruitful collaboration with C. Kleiner, F. Heine, M. Reichenbach, L. Renners, and T. Oelsner is gratefully acknowledged. Furthermore, we would like to thank the participants of the user study for their time and efforts.

References

[1] M. Dumas, J. Robert, and M. J. Mcguffin, "Alertwheel: radial bipartite graph visualization applied to intrusion detection system alerts," *IEEE Network*, vol. 26, no. 6, pp. 12–18, November 2012.

[2] Q. Liao, A. Striegel, and N. Chawla, "Visualizing graph dynamics and similarity for enterprise network security and management," in *Proc. VizSec*. New York, NY, USA: ACM, 2010, pp. 34–45.

[3] Y. Zhao, F. Zhou, X. Fan, X. Liang, and Y. Liu, "Idsradar: a real-time visualization framework for ids alerts," *Science China Information Sciences*, vol. 56, no. 8, pp. 1–12, Aug 2013. [Online]. Available: https://doi.org/10.1007/s11432-013-4891-9

[4] M. Angelini, N. Prigent, and G. Santucci, "Percival: proactive and reactive attack and response assessment for cyber incidents using visual analytics," in *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*. IEEE, 2015, pp. 1–8.

[5] S. McKenna, D. Staheli, C. Fulcher, and M. Meyer, "Bubblenet: A cyber security dashboard for visualizing patterns," in *Computer Graphics Forum*, vol. 35, no. 3. Wiley Online Library, 2016, pp. 281–290.

[6] J. R. Goodall, E. D. Ragan, C. A. Steed, J. W. Reed, G. D. Richardson, K. M. T. Huffer, R. A. Bridges, and J. A. Laska, "Situ: Identifying and explaining suspicious behavior in networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 204–214, Jan 2019.

[7] Z. Yu, J. J. P. Tsai, and T. Weigert, "An automatically tuning intrusion detection system," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 373–384, April 2007.

[8] M. Rehák, E. Staab, V. Fusenig, M. Pěchouček, M. Grill, J. Stiborek, K. Bartoš, and T. Engel, "Runtime monitoring and dynamic reconfiguration for intrusion detection systems," in *Proc. RAID*. Berlin, Germany: Springer, 2009, pp. 61–80.

[9] D. Ippoliti and X. Zhou, "A self-tuning self-optimizing approach for automated network anomaly detection systems," in *Proceedings of the 9th International Conference on Autonomic Computing*, ser. ICAC '12. New York, NY, USA: ACM, 2012, pp. 85–90. [Online]. Available: http://doi.acm.org/10.1145/2371536.2371551

[10] M. Kumar and M. Hanumanthappa, "Self tuning ids for changing environment," in *2014 International Conference on Computational Intelligence and Communication Networks*, Nov 2014, pp. 1083–1087.

[11] B. Hellmann, V. Ahlers, and G. Dreo Rodosek, "Integrating visual analysis of network security and management of detection system configurations," in *Proc. IDAACS*. Piscataway, NJ, USA: IEEE, 2017, pp. 1020–1025.

[12] Trusted Network Communications Working Group, "TNC IF-MAP binding for SOAP, version 2.2, revision 10," Trusted Computing Group, March 2014.

[13] Trust@HsH Group, "Iron/VisITMeta project suite on Github," https://github.com/trustathsh/.

[14] J. Brooke *et al.*, "SUS – a quick and dirty usability scale," *Usability Evaluation in Industry*, vol. 189, no. 194, pp. 4–7, 1996.