

A First Look at the Usability of OpenVAS Vulnerability Scanner

M. Ugur Aksu, Enes Altuncu, Kemal Bicakci
TOBB University of Economics and Technology Ankara, Turkey
{m.aksu, ealtuncu, bicakci}@etu.edu.tr

Abstract—Vulnerability scanning is a fundamental step for assuring system security. It is also an integral component of IT system risk assessment to manage the identified vulnerabilities in a timely and prioritized way. It is critical that the tools for vulnerability scanning are usable so that cybersecurity practitioners get the most out of them. In this work, we evaluate the usability of a commonly used open source vulnerability scanning tool – OpenVAS 9.0. For this purpose, we carry out expert-based and user-based testings. Expert-based testing is carried out by employing the heuristic analysis and cognitive walkthrough approaches. User-based testing is performed by selecting 10 cybersecurity experts as participants. As a result, we identify pitfalls that lead to insecurity or false sense of security and suggest improvements to overcome them. We also discuss the effectiveness of the methodologies employed for usability testing. Lastly, a set of heuristics compiled from the existing work and adapted to our case is provided to be reused in similar studies.

Index Terms—usability, usable security, vulnerability scanner, OpenVAS

I. INTRODUCTION

Vulnerability scanning is an indispensable practice for cybersecurity specialists. These individuals employ vulnerability scanning tools to identify weaknesses in the systems and try to eliminate the discovered deficiencies so that the system security can be assured. The scan results of such tools are also used to evaluate the overall risk level of the systems in order to manage the discovered vulnerabilities in a prioritized fashion [1].

Expected to be used by power users, such as cybersecurity specialists or pentesters, vulnerability scanners are not usually designed with usability in mind [2] [3]. Yet, the usability of such tools is of high importance in order to generate correct and comprehensive scanning results and to evaluate the scan reports properly, so that the undesired attacks due to unnoticed or residual vulnerabilities in the systems can be prevented proactively.

With this motivation, in this work, we evaluate the usability of the OpenVAS 9.0 vulnerability scanner in terms of the level of security provided through correct usage by the targeted users. OpenVAS has been chosen for analysis due to its

widespread use among practitioners for being an open source tool as well as having a comprehensive library of vulnerability detection plugins.

For the analysis, usability testing is carried out through expert-based and user-based testings. Expert-based testing is practiced with a two-step process of heuristic walkthrough that combines both heuristic analysis and cognitive walkthrough in order to offset the limitations of the both methodologies and to maximize the number of findings [4]. User-based testing is exercised to identify deficiencies not discovered by usability inspections as well as to confirm the findings of the former approach.

The contribution of our work is as follows. To the best of our knowledge, our work is the first to analyze OpenVAS from a usable security perspective. Through analysis of both expert-based and user-based testings, we identify critical pitfalls that either hinder the usability of the OpenVAS or cause false sense of security that could lead to insecurity at the systems scanned. Further, we suggest improvements or corrections where possible. We also provide a comprehensive set of heuristics, compiled from the existing work and adapted to our case, to be reused in similar studies. Moreover, given the fact that usability studies conducted with cybersecurity specialists are rare, since they are typically reluctant to carry out such studies and it is usually hard to find enough of such users, our work with cybersecurity specialists is significant. Lastly, though only OpenVAS is analyzed in our work, several other vulnerability scanners have similar features and they follow almost the same progression of steps for conducting vulnerability scans. Thus, most of the findings outlined in this work can also be useful for the evaluation of other vulnerability scanners.

The rest of the paper proceeds as follows. Background information and related work is given in Section II. Section III explains the methodology exercised for usability evaluation of the OpenVAS. We present the results in Section IV and discuss the findings in Section V. Lastly, conclusions and future work are presented in Section VI.

II. BACKGROUND AND RELATED WORK

In general terms, a vulnerability is a weakness in an information system that could be exploited by a threat source. A variety of vulnerabilities may exist across an IT network such as known software bugs, missing operating system patches, vulnerable services or insecure default/customized configurations. These deficiencies can be detected by the

use of vulnerability scanners. Two groups of vulnerability scanners can be named according to the type of the system targeted for assessment. One group of vulnerability scanners such as OpenVAS, Nessus, and Nexpose aims to enumerate application-based or configuration-related deficiencies while the other group including Nikto and Acutenix focuses on discovering web application or web server vulnerabilities.

Among these, OpenVAS is an open source and powerful vulnerability assessment tool capable of both vulnerability scanning and management. Additionally, it can identify the active services, open ports and running applications across the machines. It has a scan engine updated regularly with vulnerability detection plugins called Network Vulnerability Tests (NVTs). In addition to being free, its capability to detect vulnerabilities for a wide range of operating systems and applications makes it a popular tool among the pentesters.

Even though the literature on usability of security applications is rich and expanding, only a few have assessed the usability of vulnerability scanners and to our knowledge this work is the first specifically targeting OpenVAS.

To summarize the earlier work, Wang and Yang [5] reviewed open source vulnerability scanners in search of identifying a candidate scanner for pedagogical purposes. Their work identifies OpenVAS as a potential candidate for being a free and powerful scanner though it is not considered the easiest to install and use. This work does not carry out any usability study in order to justify the opinions expressed about the usability of the OpenVAS.

Jøsang et al. [6] presented a list of usable security principles for vulnerability analysis and risk assessment. They provided examples how these can be used to define vulnerabilities for conducting risk assessments.

Yoshimoto et al. [3] investigated the usability of Nessus and compared it with a tool they have developed. They found critical usability issues in Nessus but their findings were limited in terms of both scope and content. For the evaluation, user-based testing with six subjects of general users was preferred.

In a more recent work by Bingham et al. [2], the authors analyzed the usability of Nessus using heuristic walkthrough methodology for the evaluation. Our work differs from theirs in three aspects. First, we also carry out user-based testing. Second, we conduct usability inspections with a set of more extensive heuristics focusing on usable security. Third, the system under evaluation (SUE) in our work is OpenVAS rather than Nessus, which is no longer open-source software.

Among the rich literature on usable security, the seminal work by Whitten and Tygar [7] is significant for our work as well in revealing and emphasizing the need to define a different set of principles for the usability of security applications instead of a set of generic principles. In our work, we revisit the definition of usability for security and make a usability evaluation in a similar manner but for a different application, that is OpenVAS 9.0.

III. EVALUATION METHODOLOGY

By making use of the usability definition in a security context by Whitten and Tygar [7], in our evaluation, we focus on finding the answers to the following questions relating to carrying out the tasks of vulnerability scanning and evaluating the results using the OpenVAS vulnerability scanner.

- *Will the users realize what needs to be done?*
- *Will they figure out how to do the tasks?*
- *Will there be any errors caused by user actions that lead to insecurity?*
- *Will the users be able to use the product comfortably and continue to use it in the future?*

Expanding the high level goals given above, we aim to measure the extent to which the following set of evaluation objectives (O) can be met using OpenVAS.

- O1. *Users must be able to start and complete a vulnerability scan successfully.*
- O2. *Users must be able to conduct the scan with the user desired parameters/options, understanding clearly what the options mean and what the respective consequences are.*
- O3. *Users must be informed about the default configurations or the actions taken automatically on behalf of the users.*
- O4. *Users should not make any critical errors i.e., the application must prevent users from making critical errors that may result in insecurity.*
- O5. *Users must be able to understand how complete and comprehensive the results are.*
- O6. *Users must be able to explain how trustworthy the results are. They must be able to identify any false negatives/positives and explain the possible causes for the false/missing results.*
- O7. *Users must be able to identify and interpret the severity of the identified vulnerabilities and understand why they are critical.*
- O8. *Users must be guided to decide on the appropriate actions to be taken to mitigate the risks associated with the vulnerabilities.*
- O9. *Users must be comfortable with the general usage of and willing to continue to use the OpenVAS in the future.*

In this context, we conduct **expert-based testing** i.e., usability inspections, a set of informal methods based on informed intuitions [8] [9], and **user-based testing**, an empirical approach to evaluate the usability of the OpenVAS, specifically to measure the extent to which security can be assured through usability. Employing both expert-based and user-based testings could reveal issues that might be overlooked if only one of them is used instead. Earlier work [10] [11] also suggested to combine several methods to reach more extensive results in usability studies.

Following the approach taken by Bingham et al. [2], expert-based testing is further conducted utilizing **heuristic evaluation** and **cognitive walkthrough** methodologies so that we can

benefit from the advantages of both approaches and enrich our findings.

Combining both heuristic evaluation and cognitive walkthrough as a two-pass evaluation methodology is also named as *heuristic walkthrough* by Sears [4]. A significant advantage of using heuristic walkthrough as an evaluation methodology is that the evaluators can get accustomed to the common use-case scenarios with the first pass of the cognitive walkthrough so that they can be more focused on frequently used tasks in the second pass, performed as a free-form approach of heuristic evaluation [2].

After conducting expert-based testing, we carry out user-based testing, which is the most commonly employed approach in evaluating usability [9], to further identify deficiencies not discovered by usability inspections, as well as to compare and verify the findings of the former approach.

In usability studies, it has been established that small teams could identify issues not possible by only individuals [11]. However, determining the size of the team is still disputed. Some suggest that five is enough [12] to find approximately 80% of usability problems, while Schmettow [13] suggests more than 10.

In our study, expert-based testing involves three double-expert evaluators i.e., both usability experts and have enough expertise in cybersecurity domain and vulnerability scanning. These three experts are also the authors of this work. Additionally, user-based tests are conducted with 10 cybersecurity specialists.

A. The Test-bed and Pilot Study

For the test-bed, we have created a simple virtual network with three hosts on it. The first host run the OpenVAS on top of the Ubuntu 18.04 operating system. The second host was installed with Fedora 27 operating system and the last host was an intentionally vulnerable Ubuntu Linux virtual machine designed for testing purposes (Metasploitable 2). Before the evaluation, OpenVAS was set up on the first host and had the most recent plugins.

For the test environment, participants were provided with a desktop computer with dual monitors. On the first monitor OpenVAS was displayed via a web browser while the second monitor displayed a Google search page to be used for conducting searches on the usage of OpenVAS.

Before conducting the actual tests, we first conducted pilot studies, both for the user-based and expert-based tests in order to refine the tasks, the goals and the procedures.

B. Cognitive Walkthrough

Modeled after the code walkthrough technique practiced in software engineering, as a usability inspection approach, cognitive walkthrough is based on simulating the users according to a prespecified usage scenario in order to identify probable deficiencies in the SUE [7] [9]. Usually, the results of exercising the tasks are compared with pre-defined user goals to determine the discrepancies between them [14]. Mental simulation of the users with the use cases is achieved by the

evaluators taking on the personas of the targeted users of the SUE [2].

In our evaluation, the simulated and actual personas are the same i.e., cybersecurity specialists, the targeted users of the SUE.

We ask the evaluators to consider six core tasks (T), seen in a typical progression of a vulnerability scanning process:

- T1. *Login to the OpenVAS.*
- T2. *Scan the local host with OpenVAS to discover the highest number of existing vulnerabilities.*
- T3. *Scan two hosts at the network including the local host.*
- T4. *Scan the whole network.*
- T5. *Evaluate the results and form a prioritized remediation plan.*
- T6. *Determine the necessary remediation actions for each vulnerability.*

C. Heuristic Analysis

Compared to cognitive walkthrough, heuristic analysis is more of a free form, but an in-depth evaluation technique where specialists make judgments based on established usability and security principles and rules of thumb [9]. Yet, it is capable of discovering a higher number of and usually more serious issues. It is also reported to be easier to learn [8].

Evaluators are usually domain experts and expected to explore the SUE freely and identify any issue given their past experiences and a set of guiding heuristics. It is argued that if the evaluators are also usability experts [7], then more useful results could be obtained [15].

The heuristics are usually in the form of general guiding principles, such as the *10 Usability Heuristics for User Interface Design* defined by Nielsen [16] or the *8 Golden Rules of Interface Design* described by Shneiderman [17]. However, in usable security studies, it is more appropriate to define domain-specific heuristics in addition to the general heuristics since such work requires more tailored standards to assess the level of security assured, as proposed by Whitten and Tygar [7].

Though the comprehensive and widely accepted set of general usability heuristics defined by earlier work are useful, in this work, we focus more on identifying any issues that may hinder accomplishing security tasks and reveal any issues that may cause insecurity or false sense of security. In this respect, we find the security action and security conclusion usability principles defined by Jøsang et al. [6] to fit in to our case and use a slightly tailored version of those heuristics to evaluate the SUE. Specifically, the set of security action heuristics relate to the task of conducting vulnerability scanning, while the security conclusion heuristics guide evaluating the scan results.

Heuristics of Security Action (A):

- A1. *Users must understand the security actions they are expected to conduct.*
- A2. *Users must be informed with sufficient knowledge to take the correct security actions i.e., must be prevented from*

taking erroneous or forgetful security actions that may lead to insecurity.

- A3. Users must be informed about any possible undesired consequences or side-effects of the security action taken.
- A4. Conducting the security action must be tolerable for the users mentally and physically.
- A5. Users must be comfortable with taking the security action so that they will not give up using the application in the future.

Heuristics of Security Conclusion (C):

- C1. Users must understand the security conclusions they are expected to make.
- C2. Users must be provided with sufficient information for deriving the security conclusion in a correct and exhaustive way.
- C3. Conducting the security conclusion must be tolerable for the users mentally and physically.
- C4. Users must be comfortable with taking the security conclusion so that they will not give up using the application in the future.

D. User-Based Testing

In addition to the usability inspections described above, a user-based testing with formative and summative approaches has been conducted to explore further empirically the issues that result in security deficiencies. With the formative testing, we have identified the usability issues with regards to security, as practiced in the earlier sections, and with the summative evaluation, we measured the extent to which the tasks were completed successfully.

User-based testing involved 12 cybersecurity experts (2 of them were only involved in the pilot study) who were volunteers recruited out of about 30 professionals that worked in the same cybersecurity company. As an encouragement for participation, it was announced that the study could serve as a short hands-on training opportunity for the OpenVAS. The participants were given a pre-study survey to derive the demographics data and to determine if they fit in our definition of the targeted users, given the education level and the professional experience they had. In our study, only the participants that had B.S. degree in Computer Engineering/Science with at least two years of recent professional experience in the cybersecurity domain and those that hold at least one of the security related certifications were considered to be in the target user group. The participants of our study hold at least one of the following certificates; Certified Information Systems Security Professional (CISSP), Offensive Security Certified Professional (OSCP), Certified Ethical Hacker (CEH), and ISO 27001 Auditor. Some of the representative profession titles of the participants were; cybersecurity domain expert, pentester and risk analyst, threat intelligence expert, and incident response specialist. 5 out of the 10 participants had previous experience with a vulnerability scanner other than OpenVAS (such as Nessus, Nexpose etc.), thus they were categorized as *expert users* while the other 5 users were categorized

as *novice*. All of the participants had a good understanding of vulnerability scanning while none of them had used the OpenVAS previously. Other demographics data of interest for the participants are summarized in Table I.

TABLE I
PARTICIPANT DEMOGRAPHICS

| Gender | Participants | Education | Participants |
|--------|--------------|-------------------|--------------|
| Female | 3 | Bachelor's Degree | 4 |
| Male | 7 | Graduate Degree | 6 |

| Age | Participants | Years of Experience | Participants |
|-------|--------------|---------------------|--------------|
| 26-30 | 4 | 2-5 | 5 |
| 31-35 | 3 | 6-10 | 3 |
| 36-40 | 3 | 11-15 | 2 |

In the study, participants were asked to complete the six tasks specified in the cognitive walkthrough section. The task list was deliberately kept to be short and simple. The subjects were asked to perform the tasks while they were not prompted for the sub-tasks they were expected to follow in order to test whether the SUE is informative enough to take all the security actions.

To complete the tasks, users were allowed to resort to the help page of the OpenVAS or to make online web searches. The participants were also informed that they were expected to discover the highest number of vulnerabilities that could be detected. In other terms, users were asked to make the proper configurations in order to detect as many existing vulnerabilities as possible on the network they scan.

For the evaluation, the subjects were asked to think aloud while they performed the tasks so that we could grasp further issues that were not obvious through bare sighting, as suggested by Lewis and Rieman [18]. The body motions and facial expressions of the subjects were also closely watched for and noted to check if they were indicative of or related to the issues they experience.

Participants were lastly given a post-study survey in which they were asked to evaluate the general usability of the OpenVAS and express if they would opt to use the OpenVAS as a vulnerability scanner in the future after conducting this study.

IV. RESULTS

This section describes our findings of the two different methodologies exercised; expert-based (cognitive walkthrough and heuristic analysis) and user-based testing.

A. Cognitive Walkthrough Results

Our main findings pertaining to cognitive walkthrough are as follows:

1) *Login to the OpenVAS (T1)*: The login screen of the OpenVAS is reached by typing "https://ip-address:4000" at the address bar of any browser. On this screen, users are asked for a username and a password, which are set as "admin" and "admin" by default as illustrated in Figure 1. However, users are not prompted or forced to change these default credentials and this may lead to a critical security issue if users continue

to use them as is. Any attacker, who knows the IP address of the machine on which the OpenVAS is running, can reach the application remotely and scan the network to identify exploitable vulnerabilities. Even worse, s/he can reach the vulnerability scan reports that will be useful for conducting successful attacks, violating our evaluation objective of O4.

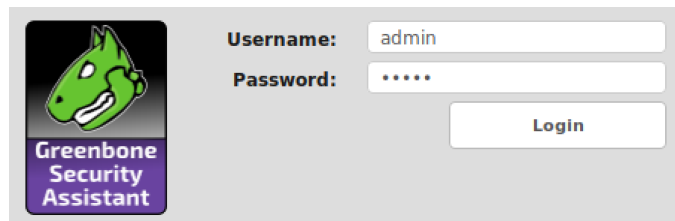


Fig. 1. Login Screen of the OpenVAS

2) *Scanning Local Host with the Task Wizard (T2)*: To scan a host using the wizard, there are two options under the *Scans* → *Tasks* menu item: *Task Wizard* and *Advanced Task Wizard*. When the *Task Wizard* option is selected, a pop-up screen, illustrated in Figure 2 appears. After the IP address field for the machine to be scanned, a short list of the tasks to be carried out behind the scene is depicted. Though this option allows for a quick start of a scan for a host, it does not inform the users about the default scan settings or allow access to customize the settings. We identify this issue to violate our evaluation objective of O3.

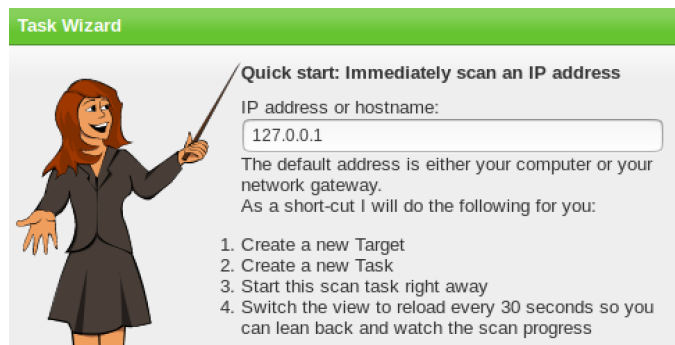


Fig. 2. Local Host Scan with the Default Parameters

3) *Scanning Local Host with Advanced Task Wizard (T2)*: Optionally, local host can be scanned using *Advanced Task Wizard* under the *Scans* → *Tasks* menu item. This option allows the user to define the scan configuration manually, as illustrated in Figure 3. For instance, the *Scan Config* options are enumerated as: *Full and fast*, *Full and fast ultimate*, *Full and very deep*, *Full and very deep ultimate*, *Host Discovery*, and *System Discovery*. However, it fails to inform the users about what the scan configuration options mean and what the advantages and disadvantages are i.e., *how comprehensive the scan will be*, *how long the scan will take to complete* or *what load of network traffic will be generated*, so that the users can make deliberate choices. Similarly, users are informed that they can conduct either authenticated or unauthenticated scans

with no clarification on what the differences or consequences are depending on the selection. In this respect, we note that our evaluation objective of O2 has not been met.

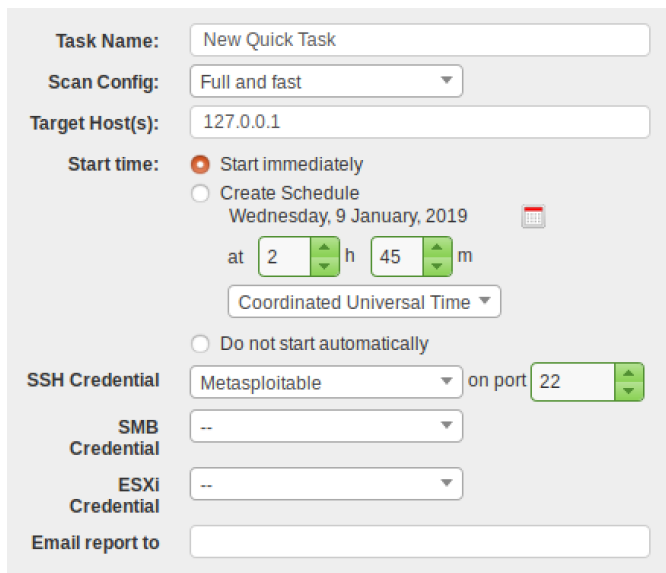


Fig. 3. Local Host Scan with the Advanced Task Wizard

4) *Scanning Multiple Hosts with the Advanced Task Wizard (T3 and T4)*: To scan multiple hosts, it is unclear how to enter multiple host IP addresses to the *Target Host(s)* text field. Moreover, if the entered text for the target hosts is not in the expected format, an error message is displayed indicating that the host specification is wrong, with no guidance on the acceptable syntax. After a Google search, we discover that multiple host IPs should be separated with comma signs and CIDR notation needs to be used to scan a whole network or a specific sub-network. This issue is illustrated in Figure 4, and it violates our evaluation objective of O1.

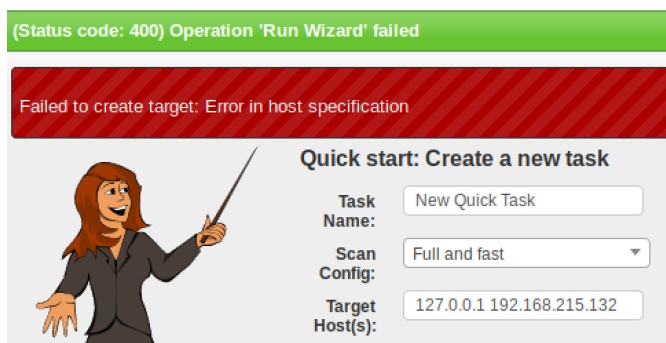


Fig. 4. Scanning Two Hosts with the Advanced Task Wizard

5) *Credentialed Scan of Multiple Hosts (T3 and T4)*: To define a credentialed scan, we input the authentication data to the *SSH Credential* field, and to our surprise, we get the “No results found” message, as illustrated in Figure 5. After some Google search, we discover that credential information must be defined prior to a scan task under the

menu item of *Configuration* → *Credentials*. The user has to quit configuring the current scan task and needs to define the required credentials first, under another menu item. This issue reveals that the task order is unknown to the user before creating a scan task. Due to this issue, users may decide to continue to scan with no credentials supplied, in violation of the evaluation objectives of O2, or they may accidentally do so, violating the O4. Finally, even if the credentials are defined beforehand, setting multiple credentials for multiple hosts is not supported since only one credential item can be selected through the GUI. Using multiple credentials is only supported through creating a custom credentials file and using it with the *omp* command through the Command Line Interface (CLI).

Fig. 5. Credentialed Scan of Two Hosts

6) *Listing the Vulnerabilities Identified (T5)*: For evaluating the results, findings of a scan can be listed by clicking on the *Date* field under the *Scans* → *Tasks* menu item, as illustrated in Figure 6, rather than clicking on the numbers under the severity categories of *High*, *Medium*, *Low*, *Log* and *False Pos*. In this view, only the results of severity level of *Low* and above are shown in the order of severity ratings, hiding the results for severity levels of *Log* and *False Pos*. by default. Searching through the options on the GUI, we discover that the severity levels for listing the results can be customized through the *Update Filter* option. However, it is very demanding for the first time users to locate this option. In this respect, we regard this issue in violation of O5, since users cannot understand easily why the full list of findings is not displayed.

7) *Interpreting the Severity of the Results (T5)*: Continuing on the task of evaluating the results, the detected vulnerabilities are categorized by their respective CVSS scores and colored appropriately to inform the user visually with regard to the severity levels of the findings, as depicted in Figure 7. Moreover, the results can be narrowed down by clicking on the pie chart or the bar chart that categorizes the vulnerabilities,

| Date | Status | Task | Severity | Scan Results | | |
|--------------------------|--------|---------------------------------------|--------------|--------------|--------|-----|
| | | | | High | Medium | Low |
| Sat Oct 27 09:47:49 2018 | Done | Localhost Full and Very Deep Ultimate | 5.0 (Medium) | 0 | 3 | 0 |
| Sat Oct 27 07:53:14 2018 | Done | Localhost Full and Very Deep Ultimate | 5.0 (Medium) | 0 | 3 | 0 |

Fig. 6. Scan Results of a Task

as well as choosing the word of interest from the word cloud, as illustrated in Figure 8. However, how the rating scores are calculated and the categorical effects of successful exploitation of vulnerabilities are unknown to the users, as also highlighted by Bingham et al. for Nessus [2]. In this case, we believe that users could more correctly prioritize the vulnerabilities if they have insight into the lower metrics of the severity calculations [19] or the final impacts of the vulnerabilities [20]. Thus, our objective of O7 seems to be met partially, since the question of what makes them critical is unanswered.

| Vulnerability | Severity | QoD |
|---|--------------|-----|
| Ubuntu Update for apache2 USN-1765-1 | 5.0 (Medium) | 97% |
| PostgreSQL Denial of Service Vulnerability (Linux) | 4.0 (Medium) | 80% |
| Ubuntu Update for perl USN-1643-1 | 7.5 (High) | 97% |
| PostgreSQL Version Detection (Linux) | 0.0 (Log) | 80% |
| Ubuntu Update for tiff vulnerability USN-1102-1 | 6.8 (Medium) | 97% |
| PostgreSQL Version Detection (Linux) | 0.0 (Log) | 80% |
| Ubuntu Update for net-snmp USN-1450-1 | 3.5 (Low) | 97% |
| Ubuntu Update for apr USN-1134-1 | 4.3 (Medium) | 97% |
| Pidgin Oscar Protocol Denial of Service Vulnerability (Linux) | 5.0 (Medium) | 80% |
| Ubuntu Update for php5 USN-1231-1 | 7.5 (High) | 97% |

Fig. 7. Scan Results View (Vulnerability Listing)

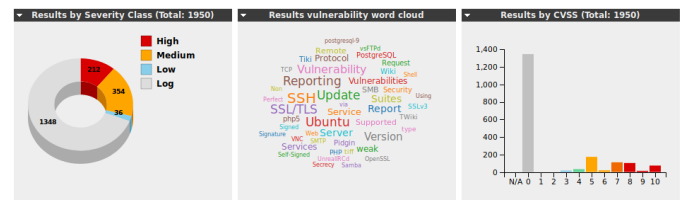


Fig. 8. Scan Results View (Pie Chart, Word Cloud, and Bar Chart)

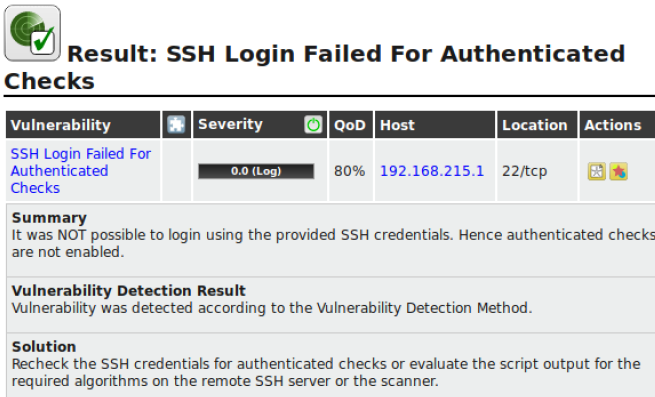
8) *Identifying the Remediation Actions (T6)*: After discovering the vulnerabilities and prioritizing them by their severity, the necessary actions to remove the vulnerabilities needs to be determined. For this purpose, when clicked on the name of the vulnerabilities, on the vulnerability details page, users are advised with a list of actions in the categories of *VendorFix*, *Mitigation* or *Workaround*. Thus, we consider that the evaluation objective of O8 seems to be met.

B. Heuristic Analysis Results



Here, we present only the findings that are different from the results of the *Cognitive Walkthrough* assessment, discussed in the previous subsection.

1) *Credentialed or Non-Credentialed Scans*: Scan configurations affect significantly both the quality of the results and the way the scans are performed. Specifically, the choice of credentialed or non-credentialed scan makes crucial difference on the results. With a credentialed scan, usually there is less traffic load on the network and the better the results are. On the other hand, a non-credentialed scan might be preferred to see the network the way an unsophisticated attacker would see. For the authentication method, SMB authentication is typically used for Windows based hosts, while SSH authentication is used for Unix hosts. Though the users are allowed to make these decisions through the GUI, they are not informed when to choose which, so that they can make informed and correct decisions for scanning networks. We notice this issue through the heuristics of A2 and A3, and observe that it violates the evaluation objective of O2.

2) *Failing to Display Critical Error Logs*: As discussed in the Cognitive Walkthrough section, scan results of *Low* and upper severity levels are listed by default, disregarding the *Log* messages. However, we notice that some log messages are of high value for the users since they inform what goes wrong during a scan. An example of such a case is when a credentialed scan cannot be conducted due to an error and a non-credentialed scan is performed instead, unbeknownst to the user, as depicted in Figure 9. Thus, failing to notify the user about such critical issues could lead the user to reach incorrect security conclusions and gives false sense of security (C2), violating the evaluation objective of O5.



Result: SSH Login Failed For Authenticated Checks

| Vulnerability | Severity | QoD | Host | Location | Actions |
|---|-----------|-----|---------------|----------|---|
| SSH Login Failed For Authenticated Checks | 0.0 (Log) | 80% | 192.168.215.1 | 22/tcp |   |

Summary
It was NOT possible to login using the provided SSH credentials. Hence authenticated checks are not enabled.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Solution
Recheck the SSH credentials for authenticated checks or evaluate the script output for the required algorithms on the remote SSH server or the scanner.

Fig. 9. SSH Login Failure Log

3) *Side-Effects on Network Performance and System Availability*: Conducting vulnerability scan tasks using a network-based scanner like OpenVAS usually generates large amount of system requests and network traffic for a considerably long duration at the medium to large-sized networks. Thus, it is critical that the scan tasks are not causing any deterioration in the network performance, such as the unavailability of some of the critical services. It is usually assumed that the pentesters are aware of this fact and they configure the scan jobs with carefully chosen parameters so that the generated network traffic overhead is not interrupting other critical tasks. However, the pentesting world is full of such stories

where they had to learn this fact with costly experiences. In this respect, it would be useful to inform the users with a notification indicating the level of network traffic overhead so that the users could go back and change the scan parameters for a more optimized scan task. This issue is identified with our heuristic of A3 and it violates our evaluation objective of O4.

4) *Up-to-Dateness of the Plugins (NVT) Database*: OpenVAS, as a vulnerability scanner relies on the plugins i.e., Network Vulnerability Tests (NVTs) to discover the potential vulnerabilities at the hosts. Since OpenVAS can only check for those vulnerabilities for which there exist corresponding NVTs, the number of detected vulnerabilities depends on how large and up-to-date the NVT database is. This database is usually updated through the CLI during the installation of the OpenVAS. However, users are unaware of the up-to-dateness of the NVT database and are not notified when the database gets outdated. Instead, users are expected to call the *greenbone-nvt-sync* command periodically through the CLI for the update process. As a result, users may reach a false security conclusion of the system scanned since all the potential vulnerabilities may not have been detected. The fact that users are neither informed for (A1) nor prevented from missing the update process (A2) and make a security conclusion with missing information (C2) violate the evaluation objectives of O4, O5 and O6.

5) *Identifying Results With False Positives and Negatives*: In the parlance of vulnerability scanning, a false negative is the failure to recognize an existing vulnerability in the SUE, whereas a false positive is the incorrect determination of vulnerability. With OpenVAS, the reliability of the produced results can be observed through the *Quality of Detection (QoD)* ratings, as shown in Figure 7. However, users are not informed about the possible false negative scenarios.

False negatives may occur, firstly due to the deficient scan configurations. Decisions such as port range to be scanned or the NVT type may affect the completeness of the results significantly. To give an example, some security checks are disabled by default to prevent causing harm to the systems. In this case, OpenVAS will rely on banners instead of actually performing the security checks with harmful effects, resulting in a less reliable report. Moreover, users are unaware of such a customization and this option can only be configured manually through the *openvasd.conf* configuration file.

Secondly and more significantly, users are not informed about how many of the known vulnerabilities (CVEs) can be potentially detected by the OpenVAS given the NVTs to detect them. To our knowledge, out of the more than 110 000 CVEs currently available through National Vulnerable Database (NVD), only about 40 000 can be detected through OpenVAS NVTs. Thus, even with an up-to-date NVT database, it is not possible to discover all the potential vulnerabilities with the OpenVAS, unbeknownst to the users. We identify this issue through the heuristics of A2 and C2 and it is in violation of the evaluation objective of O2, O3 and O6.

The findings of the expert-based analysis is summarized

at the Table II, depicting respectively which methods and identifiers are used to discover the issues, proposed solutions for mitigating the issues and the corresponding evaluation objectives violated. In Table II, C/W stands for *Cognitive Walkthrough* while H/A is short for *Heuristic Analysis*.

C. User-Based Testing Results

In this section, we first depict and explain the results of the summative evaluation and then discuss our findings of the formative evaluation in detail.

For the summative evaluation, the extent to which the tasks can be completed successfully is measured by a scaled rating of five categories. If users can complete the tasks successfully on their own without the help of the documentation or search engines, they are considered to have completed the task. If users, on the other hand, cannot complete a task with the guidance of the GUI and resort to the help documentation or conduct an online search on how to carry out the task, then it is considered to be completed with help. While carrying out the task, if a critical configuration is overlooked or cannot be defined properly before a task is started, then the task is considered to be completed partially. Lastly, if users cannot complete a task, they are marked as “Not Completed”. In this respect, the summative evaluation is summarized at Table III.

Overall, the tasks of logging in to the OpenVAS (T1), differentiating the critical vulnerabilities reported (T5), and determining the remediation actions for the vulnerabilities (T6) could easily be completed successfully by the users. However, most of the users failed to scan the hosts with the necessary configurations made. For scanning the local host (T2), only one user completed the task successfully while the rest failed to configure the scan task properly though they thought they were successful. Similarly, only 3 users have completed the tasks of scanning multiple hosts (T3) and the whole network (T4) with expected configurations. Though the tasks of T3 and T4 are more demanding, the success rate is slightly higher compared to the rate of T2. This is due to the fact that after completing T2 with the *Task Wizard* option, most of the users thought that a more capable page on the GUI is required to define multiple hosts to be scanned and they used the *Advanced Task Wizard* for this purpose.

For the tasks of T2, T3 and T4, what astounded us most is that the users thought they completed the tasks successfully. However, 7 out of the 10 participants failed to define properly configured scan tasks even though they were clearly asked to detect as many vulnerabilities as possible. They were mainly focused on starting the tasks somehow instead of all the configurations checked. Out of those 7, 3 of the users did not check out any of the configuration options and started the scan tasks immediately while the other 4 users decided not to choose any option after checking out a few of the options and finding them confusing and not informative.

It is also thought provoking to observe that the participants did not resort to help pages or the online searches to figure out what the configuration options mean, even though they

expressed they did not understand some of the critical configurations. Overall, only 2 users made online searches at points where they could not proceed with the tasks since a necessary field was missing. *This indicates that security is of second concern even for the security experts since they opt to take the easier path of leaving the default configurations as is rather than looking up online to set them correctly.*

For the formative evaluation, in terms of general usability, participants were comfortable with navigating on the menu items and detecting the desired sub-menu items. However, almost all of the participants expressed the need to enlarge the icons for the action items on the GUI since they had trouble in noticing them. For instance, users spent two minutes on average to detect the new task or task wizard icons, once they were on the correct page.

For the task of logging in to the OpenVAS (T1), the users neither identified the default credentials as a security issue nor attempted to change them at the first use, supporting our previous finding by the expert-based analysis.

For scanning the local host (T2), 9 of the users have used the *Task Wizard*, where only IP address of the host to be scanned is needed to be determined. In terms of general usability, it is almost always a good idea to have quick task wizards so that the users can go with the default options if they are not comfortable with the advanced options of the software that they use. However, in a security application like OpenVAS, users at least need to be informed about the default configurations chosen on behalf of them and they must be directed to the advanced configurations pages in order to get the most out of the application to achieve the desired security level and prevent false sense of security. This finding also supports our assertion made previously in the expert-based analysis.

For the tasks of scanning multiple hosts (T3 and T4), users experienced significant difficulty in defining multiple hosts in the only mandatory field of the *Advanced Task Wizard*, since they did not know which character to use to separate multiple hosts. All of the users expressed that they did not understand how to set multiple IPs and 3 of them guessed it successfully to be a comma sign while the rest tried either a semicolon or space at their first trial. Again, we successfully identified this previously in expert-based analysis.

To conduct a credentialed scan, 3 participants who completed the task of T3 and T4 successfully, navigated to the *Advanced Task Wizard* page first only to find that the credentials needed to be defined under another menu item, as discussed in detail previously. Thus our finding on the need to explain the task order for credentialed scan is confirmed by the user-based testing.

Regarding determining the criticality of the detected vulnerabilities, 9 of the users have chosen the *Severity* attribute to be a correct identifier though 5 of them got confused with the *QoD* attribute and expressed that they did not understand what it means or what it stands for. Even worse, 1 user could not determine the criticality of the vulnerabilities since he could not choose between the *Severity* and *QoD* as a correct identifier.

TABLE II
SUMMARY OF THE EXPERT-BASED ANALYSIS RESULTS

| Detection Method | Identifier | Issue | Proposed Solution | Violation |
|------------------|------------|--|--|------------|
| C/W | T1 | Default login credentials. | Forcing users to change the credentials at first use. | O4 |
| C/W | T2 | In the "Text Wizard", users are not informed about the default scan configurations. | Inform the users about the default configurations. | O3 |
| C/W | T2 | In the "Advanced Text Wizard", scan configuration features and their advantages/disadvantages are not clear. | Present mechanisms to explore the each configuration feature in detail. | O2 |
| H/A | A3 | Network traffic overhead caused by a scan task is not known before starting a task. | Network traffic load should be indicated for a given scan task configuration. | O4 |
| C/W | T3, T4 | Acceptable syntax for defining multiple IP addresses is not elucidated. | | O1 |
| C/W | T3, T4 | Task order for a credentialed scan cannot be determined before defining a scan task. | Inform the users about the possible task orders or allow for defining credentials at the task wizard page. | O2, O4 |
| H/A | A2, A3 | Users are not informed on when to choose credentialed or non-credentialed scans. | A comparative information on the outcomes of the both methods can be divulged to the users. | O2 |
| C/W | T5 | Not displaying the results for the "Log" category by default. | Allow for an easier way to display optionally the results of "Log" category. | O5 |
| H/A | C2 | Failing to display critical error logs. | Critical error logs must be presented to the users with alarming notifications. | O5 |
| C/W | T5 | Lower metrics for CVSS scores are not exposed to the users. | Lower metrics for severity calculations can be visualized to the users for better comprehension. | O7 |
| H/A | A1, A2, C2 | Users are not informed on the update status of the NVT database. | Users must be indicated on the update status and must be warned if the NVT database is out-of-date. | O4, O5, O6 |
| H/A | A2, C2 | Outcomes with false negatives due to not existing NVTs are not apparent to the users. | How many of the vulnerabilities at the NVD can be detected by OpenVAS potentially must be divulged. | O2, O3, O6 |

TABLE III
SUMMATIVE EVALUATION OF THE TASKS

| Tasks Status / Tasks | T-1 | T-2 | T-3 | T-4 | T-5 | T-6 |
|-----------------------------|-----|-----|-----|-----|-----|-----|
| Completed | 10 | 1 | 3 | 3 | 9 | 10 |
| Completed w/ Help | - | - | - | - | - | - |
| Partially Completed | - | 9 | 6 | 5 | - | - |
| Partially Completed w/ Help | - | - | 1 | 2 | - | - |
| Not Completed | - | - | - | - | 1 | - |

We missed to detect this issue with expert-based testing since the evaluators were familiar with the term QoD and had in-depth knowledge on how the QoD ratings are made by the OpenVAS. This finding exemplifies the usefulness of user-based testing for discovering issues not caught in expert-based testing.

Major findings for the formative analysis of user-based testing are summarized at Table IV.

D. Post-Study Survey Results

To determine if OpenVAS is comfortable enough and is preferred to be used continually in the future, with a post-study survey, we asked the participants to evaluate the general usability of the OpenVAS and if they would prefer to use it as a vulnerability scanner in the future. For the results, all the participants expressed that they would use the OpenVAS in the future since they found it useful even though it had usability issues.

E. Limitations

In interpreting the results and the findings of this study, the following limitations needs to be considered.

We have conducted the laboratory experiments with 10 cybersecurity experts since increasing the sample size is a

challenging task (It is usually hard to find enough cybersecurity specialists and they are usually reluctant to attend such studies.). Thus, our findings can be confirmed by repeating the experiments with additional participants.

We have conducted the user-based testing with participants that have not used OpenVAS before in order to eliminate issues with regards to learning effect. In this respect, another experiment with participants that have experience in using OpenVAS can be conducted to compare the results.

Previously, we have argued that only 2 out of 10 participants conducted online searches when they could not understand some of the configuration parameters for the OpenVAS. Regarding this finding, Hawthorne effect needs to be taken into consideration i.e., the participants' behaviors might have been affected due to the attention they receive from the experiments, and our results can be confirmed by additional studies.

Our post-study survey result might be prone to social desirability bias though we have used an open-ended question and asked the reasoning behind the preferences as well.

V. DISCUSSION

In this work, although we analyze only the OpenVAS, most of our findings are generic and can be applied to enhance the usability of similar products. We discuss only the most significant findings here.

Quick wizards and defaults configurations are preferable in most applications to increase the usability for the novice users. However, for vulnerability scanners, novice users may reach incomplete security conclusions using such features if the capabilities and limitations of the wizards and default configurations are unbeknownst to them. Thus it is critical to notify the users regarding the limitations of such features to prevent false sense of security.

TABLE IV
SUMMARY OF THE FORMATIVE RESULTS FOR THE USER-BASED ANALYSIS

| Issue | Proposed Solution |
|---|--|
| Default credentials proved to be a security issue since none of the participants tried to change them. | We have detected these issues previously in the expert-based analysis section and have explained the respective solutions to each of them. |
| Not informing the users about the default configurations in the “Task Wizard” leads to false sense of security. | |
| Not explaining the scan configuration items in the “Advanced Task Wizard” is troublesome for the users. | |
| Task order for conducting a credentialed scan is confusing for the users. | |
| In terms of general usability, icons for action items are too small to notice. | Enlarge the icons to be noticed to increase usability. |
| In analyzing the vulnerability results, most of the users do not understand what “QoD” stands for. | The fact that “QoD” stands for “Quality of Detection” and what it is useful for should be conveyed to the users. |
| Even for security experts performing vulnerability scanning, security is a second goal. When the users are presented with a second option of faster but less secure one, that would be the path to be taken in most of the cases. | We suggest this hypothesis to be confirmed with a controlled experiment in a future work. |

Another security concern arises when vulnerability scan configuration options are incomprehensible for the novice users and they have the option to leave them in default or blank. We suggest intuitive naming for the features with hover-on textual explanations for the details. Additionally, users can be warned about the limitations of the default configurations.

Displaying information on the freshness of the scan script database is also critical to prevent false sense of security. Informing the users with the date of the last successful update or showing a warning if the database is older than an acceptable duration are two practical approaches to prevent critical user misunderstandings.

Conveying the correct task order to the users through GUI should be another major target. In our study, all of the expert users expressed the need to define the credentials first before defining a scan task, since they had experiences with other scanners, and 3 of them quit the *Task Wizard* in the correct way to define the credentials first. On the other hand, with no knowledge on the correct task order, novice users tried to enter the credentials on the *Task Wizard*, only to fail.

In addition to these, we also identified that *even for security experts that use a security application, security is a second goal*, and found it to be thought provoking. In our case, 7 out of the 10 participants opted to use the default or empty configurations when they were tasked with using optimal scan parameters. This supports the finding of Clark et al. [21] that it should not be assumed that the path that is slower but more secure is to be taken by the users when they are also presented with a second option of faster but less secure one.

VI. CONCLUSIONS AND FUTURE WORK

IT security software are often developed with functionality in mind first, sacrificing the usability. However, IT security products require a high level of usability standard to provide the security functionality they are designed for, as the earlier work have already pointed out.

In this context, we evaluated a vulnerability scanner tool, OpenVAS 9.0, with respect to usable security. With *expert-based testing*, employing the *heuristic walkthrough* method, we discovered critical usability flaws that impede the security the product aims to provide. More significantly, we identified

issues that cause false sense of security due to usability issues such as not informing the users with the meanings or the capabilities of some of the scan task configuration options or not exposing the default configurations taken on behalf of the users.

In addition to the expert-based testing, we conducted *user-based testing* in order to discover usability flaws that are encountered in the real-life scenarios, with 10 participants of cybersecurity experts. As a result, only 3 out of the 10 users could use the OpenVAS to its full potential to detect the most amount of vulnerabilities with proper scan task configurations, while the rest completed the tasks with missing or default configurations.

Between the two approaches of expert-based and user-based testings conducted, we find the expert-based testing technique of heuristic walkthrough to be easier, more practical and more efficient, given that the evaluators are both usability and domain experts. In our case, higher number of and more critical issues were discovered through the heuristic walkthrough approach, supporting the earlier assertion made by Mack [8] and Jeffries [14] that the heuristic evaluation to be more comprehensive and efficient. However, we find it to be very useful to conduct user-based tests to complement the findings of the expert-based approach.

We also both defined a domain-specific set of evaluation objectives and compiled a set of heuristics from the earlier work and adapted them to our case in order to evaluate the OpenVAS 9.0. Both the objectives and heuristics can be utilized in evaluation of similar tools and can be enhanced further as part of a future work.

For the future work, OpenVAS online community can be followed periodically to identify further usability issues and these newly detected issues can also be analyzed with user-based testings to enhance the usability of the OpenVAS.

ACKNOWLEDGEMENT

We would like to thank our shepherd, Julie Haney and the anonymous reviewers for their valuable feedback.

This work was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK), Grant no: 118E399.

REFERENCES

- [1] M. U. Aksu, M. H. Dilek, E. I. Tatli, K. Bicakci, H. I. Dirik, M. U. Demirezen, and T. Aykir, "A Quantitative CVSS-Based Cyber Security Risk Assessment Methodology For IT Systems," in *The 51st International Carnahan Conference on Security Technology*, 2017.
- [2] M. Bingham, A. Skillen, and A. Somayaji, "Even Hackers Deserve Usability: An Expert Evaluation of Penetration Testing Tools," *9th Annual Symposium on Information Assurance (ASIA14)*, pp. 13–21, 2014.
- [3] M. Yoshimoto, T. Katoh, B. B. Bista, and T. Takata, "Development and evaluation of new user interface for security scanner with usability in human interface study," *1st International Conference on Network-Based Information Systems, NBIS 2007*, vol. 4658 LNCS, pp. 127–136, 2007.
- [4] A. Sears, "Heuristic Walkthroughs: Finding the Problems Without the Noise," *International Journal of Human-Computer Interaction*, 1997.
- [5] Y. Wang and J. Yang, "Ethical hacking and network defense: Choose your best network vulnerability scanning tool," *Proceedings - 31st IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2017*, pp. 110–113, 2017.
- [6] A. Jøsang, B. AlFayyadh, T. Grandison, M. AlZomai, and J. McNamara, "Security usability principles for vulnerability analysis and risk assessment," *Proceedings - Annual Computer Security Applications Conference, ACSAC*, pp. 269–278, 2007.
- [7] A. Whitten and J. Tyger, "Why Johnny Can't Encrypt : A Usability Evaluation of PGP 5.0 University of California Understanding the problem," *the 8th USENIX Security Symposium*, 1999.
- [8] R. L. Mack and J. Nielsen, "Usability Inspection Methods," *SIGCHI '92*, vol. 25, no. 1, pp. 28–33, 1992.
- [9] J. Nielsen, "Usability Inspection Methods," *SIGCHI '95*, pp. 413–414, 1995.
- [10] H. Desurvire, J. Kondziela, and M. Atwood, "What is gained and lost when using methods other than empirical testing," *Posters and short talks of the 1992 SIGCHI conference on Human factors in computing systems*, p. 125126, 1992.
- [11] C. Karat, R. Campbell, and T. Fiegel, "Comparison of empirical testing and walkthrough methods in user interface evaluation," *Conference on Human Factors in Computing Systems, Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 397–404, 1992.
- [12] R. A. Virzi, "Refining the test phase of usability evaluation: How many subjects is enough?" *Human Factors*, 1992.
- [13] M. Schmettow, "Sample size in usability studies," *Communications of the ACM*, vol. 55, no. 4, p. 64, 2012.
- [14] R. Jeffries, J. R. Miller, C. Wharton, and K. Uyeda, "User interface evaluation in the real world," *Proceedings of the SIGCHI conference on Human factors in computing systems Reaching through technology - CHI '91*, vol. 91, no. c, pp. 119–124, 1991.
- [15] J. Nielsen, "Finding usability problems through heuristic evaluation," *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92*, pp. 373–380, 1992.
- [16] —, "10 Usability Heuristics for User Interface Design," 1995.
- [17] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist, and N. Diakopoulos, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 6th ed. Pearson, 2016.
- [18] C. Lewis and J. Rieman, "Task-Centered User Interface Design: A Practical Introduction," *Text*, p. 190, 1993.
- [19] R. M. Savola and P. Heinonen, "A visualization and modeling tool for security metrics and measurements management," *2011 Information Security for South Africa - Proceedings of the ISSA 2011 Conference*, 2011.
- [20] M. U. Aksu, M. H. Dilek, E. . Tatli, K. Bicakci, and M. Ozbayoglu, "Automated Generation Of Attack Graphs Using NVD," in *24th ACM Conference on Computer and Communications Security*, Tempe, AZ, 2018.
- [21] J. Clark, P. C. van Oorschot, and C. Adams, "Usability of anonymous web browsing: an examination of Tor interfaces and deployability," *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pp. 41–51, 2007.