16-08-2022

Deliverable D8.6
Vulnerability Assessment as a Service Pilot Project

Deliverable D8.6

Contractual Date:	31-12-2021
Actual Date:	16-08-2022
Grant Agreement No.:	856726
Work Package	WP8
Task Item:	T3
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	NORDUnet/SUNET
Document ID:	GN4-3-22-52A8D7
Authors:	D. Heed (SUNET), M. Tauson (SUNET), D. Kulinski (PSNC), T. Apell (LRZ/DFN), P. Berus (PSNC), A. Moens (GÉANT Association)

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 856726 (GN4-3).

Abstract

This deliverable provides an introduction to common vulnerability assessment and management methodologies and tools and reports on the results of a pilot project conducted by WP8 T3.2 to evaluate different vulnerability scanning tools in production or test environments in various NRENs and institutions.

Table of Contents

Table of Figures

Table of Tables

Executive Summary

The objective of Work Package 8 Task 3.2 is to investigate feasible and scalable solutions for security vulnerability identification and the tools best suited for this purpose, building on the experience of the NRENs, and to support the NRENs in implementing vulnerability scanning facilities and services in an efficient way.

Currently there is no structured use of vulnerability research/scanning software within the GÉANT community. To document and understand the vulnerability assessment requirements of GÉANT´s constituents and partners, WP8 T3.2 engaged with NREN Chief Information Security Officers (CISOs) and Security Coordinators and conducted a pilot project to test different vulnerability scanning tools in production or test environments in various NRENs and institutions.

The pilot project tested both open-source and commercially available scanning tools to evaluate their performance and detection efficiency. The tools tested were OpenVAS / GVM, OpenVAS / GÉANT API+GUI (a generic, scalable open-source-based solution [OpenVAS_GÉANT] developed by the Task), Greenbone appliance, Detectify, Outscan, and Nessus Professional. Overviews of the results and the pilot installations are provided in Section 4. The commercial tools tested in production were those pre-purchased by NRENs or GÉANT. Holm Security tools should have been included but due to a lack of time and resources these will need to be evaluated at a later stage.

The tests were performed in a distributed manner at volunteering institutions and NRENs. Laboratory speed tests for inventory scans where performed at PSNC and scanned systems where primarily located at LRZ (Germany), SUNET (Sweden) and GÉANT (Netherlands and England).

The findings of the pilot project have shown that different tools have different strengths, with scan results and speed varying greatly for large environments. The market today is fairly mature and most tools detect most vulnerabilities but do not check for every known vulnerability so there often exists a need to run tools in parallel. Organisations therefore should not rely on the results of a single scanner, and verifications should be carried out in collaboration with IT operations teams.

The results of the pilot project have also helped to clarify the requirements for a commercial vulnerability scanner (Appendix H) that have been fed into an open GÉANT tender procedure [RFP_VMS] prepared in collaboration with the GÉANT Procurement team and which is currently underway.

1 Introduction

Having a vulnerability management programme in place enables an organisation to discover and handle vulnerabilities that could be exploited by potential attackers. Vulnerability assessment (VA) is the process of identifying, classifying and prioritising weaknesses and flaws in computer networks, applications and services. It is a key element of vulnerability management, which in turn is a key part of IT security. Vulnerability assessment typically uses automated tools such as vulnerability scanners to systematically identify issues within an organisation's IT infrastructure that represent potential security threats or risk exposures.

While a variety of scanning tools are available to identify vulnerabilities, the processing of the results can be a burdensome process due to the need to distinguish between major and minor vulnerabilities and false positives. In addition, particular challenges for research and education organisations include the huge number of IP addresses and active assets to be scanned. The benefits of vulnerability assessment as part of vulnerability management include:

- Proactive and consistent identification of threats and weaknesses in IT infrastructure.
- Timely remediation to address issues.
- Protection of sensitive systems and information against data breaches and other attacks.
- Helping to maintain business continuity and reputation.
- Reduced need for incident response.
- Compliance with cybersecurity laws and regulations.

The objective of GN4-3 Work Package 8 Security, Task 3.2 Products and Services: Vulnerability Assessment as a Service, is to provide its constituencies (i.e. GÉANT, as the operator of the backbone network, and the National Research and Education Networks (NRENs), as operators of national networks that connect to the GÉANT backbone) with the capability to assess their networks, services and applications for exposure and vulnerabilities, and to offer a suitable solution to enable them to carry out their own internal scanning.

As part of this work WP8 T3.2 has conducted a pilot project to compare different commercial and open-source vulnerability scanning tools in a production, which is the focus of this deliverable. The document first provides an introduction to common vulnerability methodologies and tools (Section 2) and describes the work being carried out by WP8 Task 3.2 towards a Vulnerability Assessment as a Service concept for the GÉANT and NREN community (Section 3). It then and presents an overview of the results of the pilot project and the tested installations, including the solution developed by Task 3.2, which implements a new API and reporting functionality in the OpenVAS platform (Section 4). Key points are drawn together in the Conclusions section (Section 5). The detailed findings and comparisons from the pilot project tools evaluation are presented in Appendices A to G.

2 Vulnerability Assessment and Management

2.1 Identifying and Verifying Vulnerabilities

Having a vulnerability management programme in place enables an organisation to discover and handle vulnerabilities that could be exploited by potential attackers. An effective vulnerability management programme centres around three areas: identifying, verifying (i.e. establishing which are "real" and which are false positives) and managing (by prioritising and remediating or mitigating) vulnerabilities. The first two of these, identifying and verifying, form part of vulnerability assessment, which is usually performed by an automated vulnerability scanning tool. (Some scanning tools offer remediation, too.) Certainly, given the great number of IP addresses and active assets in most NREN environments, manual scanning is not feasible. While penetration testing is out of scope for this document, this is also a critical step in building a complete vulnerability management capability.

As part of the scan process, the vulnerability scanner makes an inventory of working/active network elements (see Appendix A for a description of the available [Nmap] scan techniques). Making an inventory of the network for the first time, when the network is effectively "unknown" to the scanner, involves taking multiple paths or choices and it is not always safe to assume that the inventory list produced by the scanner is complete; to ensure that it is, an organisation should choose to either import an existing, validated inventory list or scan the network for online systems and open ports. Doing a full scan and comparing it with a list of expected results is recommended.

However, as port scan time is roughly proportional to the number of ports scanned, port selection may be required to achieve a reasonable balance between speed and effectiveness. Tests have shown that to achieve 99% coverage of open ports, it is sufficient to scan 15,094 ports, which will speed up a scan by approximately 400% compared to scanning all 65,535 ports (the actual number of ports in each protocol) [Nmap_PSD&S].

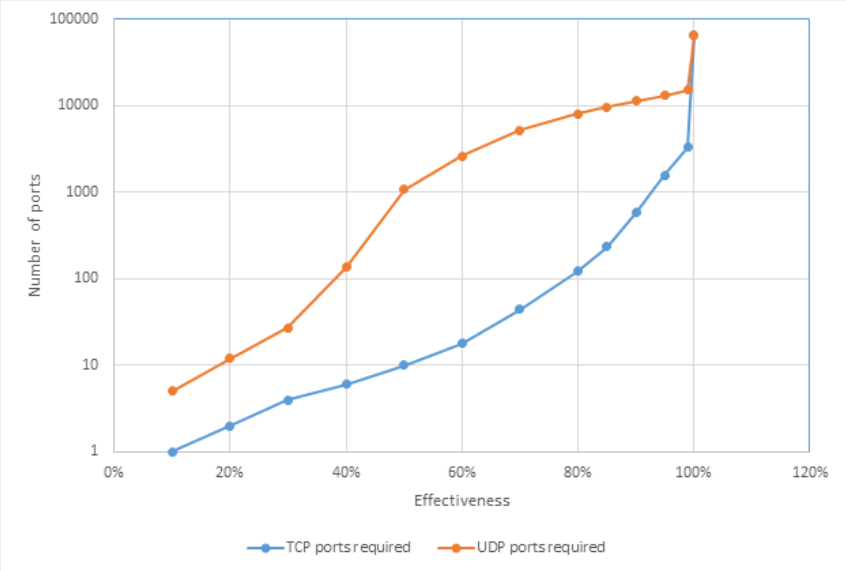

Figure 2.1: Graph of detection efficiency percentage/number of ports

The graph in Figure 2.1 shows the relationship between detection efficiency and the number of ports selected for analysis. It shows that scanning 3,328 TCP and 15,094 UDP ports is optimal, without missing too much information, which makes it possible to significantly reduce the analysis time while maintaining a high level of efficiency when scanning large environments [Nmap_PSD&S].

Findings and comparisons from the pilot project tools evaluation are presented in the appendices at the end of this document, including comparisons between the scan times of four scanners (Appendix G). This information is also maintained on the GN4-3 wiki [Wiki_VSE].

2.2 Known Vulnerabilities Data and Tracking

Most known security vulnerabilities are tracked in the Common Vulnerabilities and Exposures (CVE) database at [CVE].

The mission of the CVE programme is to identify, define and catalogue publicly disclosed cybersecurity vulnerabilities. Each vulnerability is assigned an ID, with an associated CVE record of descriptive data, and published. At the time of writing the total number of published vulnerabilities is over 170,000. The number of vulnerabilities published each year since 1999 is shown in Figure 2.2

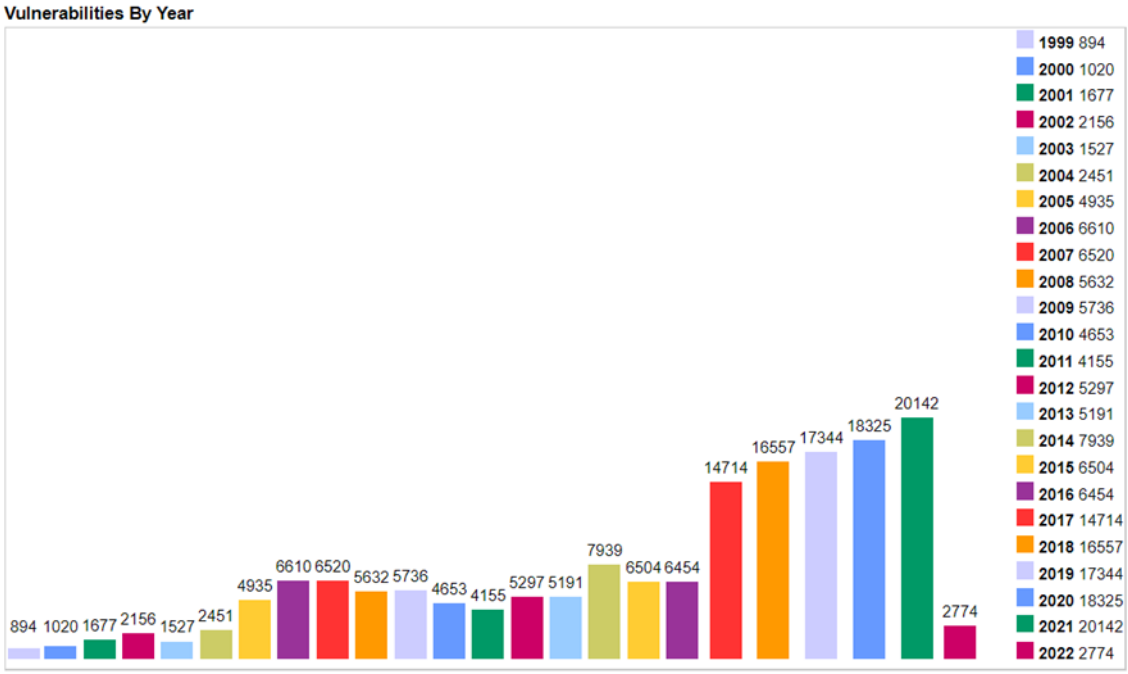

Figure 2.2: Number of vulnerabilities published in the CVE database by year

Changes to the published vulnerabilities can be tracked in the CVE vulnerabilities list [CVE Details].

2.3 Unpublished Vulnerabilities

Unpublished vulnerabilities are those that have not yet been assigned a CVE ID and are not listed in the CVE database. Zero-day (0day) attacks on such vulnerabilities are highly valuable for cyber criminals and state actors. The term 0day refers to the number of days the vendor has known of the vulnerability but not yet provided a patch. These attacks require two elements to occur; First is the vulnerability itself, along with an actual exploit or codebase to leverage it.

Remote code executions (RCEs) are a certain subset of vulnerabilities that allow code to be run remotely over a LAN or the internet. These exploits often leverage software design flaws such as buffer overflows or dynamically loading code to be interpreted.

Some researchers and companies buy and sell unpublished vulnerabilities, going against the common interest of the industry, and markets exist where exploits are made available, such as ExploitHub [ExploitHub] or Zerodium [Zerodium].

2.4 An Alternative Approach to Handling Vulnerabilities

Having in place a solid process for installing and approving the configuration of servers and for version tracking, either through a Configuration Management Database (CMDB) or through identifying assets

in use by passive network monitoring, could be an alternative or supplementary approach to keeping track of any vulnerability management intervention needed.

Such a process should include collecting information relating to:

- Software versions.
- Hardware and firmware for equipment.

In addition, following vendor and open-source information allows organisations to keep track of vulnerability information. Some examples of websites that provide security updates include:

- https://www.exploit-db.com/
- https://msrc.microsoft.com/update-guide/vulnerability/
- https://vulmon.com/searchpage?q=&sortby=byactivity
- https://www.sans.org/newsletters/at-risk/

It should be noted that basing vulnerability management on version tracking does not have the ability to detect misconfiguration or non-best practice implementations. These can be spotted by audits and gap-analysis comparisons against a good baseline. Audits and gap analysis can also help with finding and addressing root causes and so reduce the number of findings in the vulnerability scanner reports. For example, they may reveal that patch management procedures are not being followed or that developers are not following secure coding best practice and that training is required.

One of the best sources of documentation for developers to help them avoid the worst errors is the Open Web Application Security Project (OWASP) Top 10 Web Application Security Risks. This is continually updated and the latest best practices are published on [OWASP_Top10]. The 2021 OWASP Top 10 are shown in Figure 2.3 below.

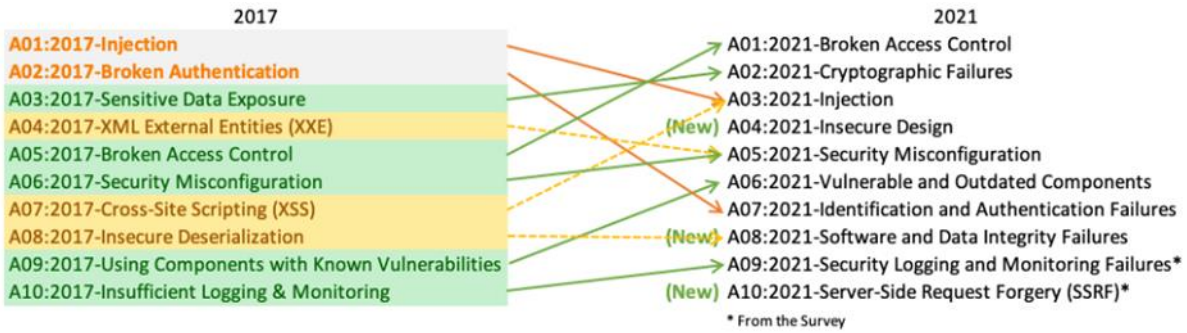

Figure 2.3: OWASP Top10 Web Application Security Risks, updated 2021

3 Vulnerability Assessment as a Service

Currently there is no structured use of vulnerability research/scanning software within the GÉANT community. Task 3.2 therefore set out to understand the needs of NRENs with regard to performing automated vulnerability assessment.

The project engaged with interested parties primarily over e-mail and also presented the topic at the 62nd TF-CSIRT meeting in January 2021. One of the key interested groups identified was the Information Security Management Special Interest Group (SIG-ISM), which offers NREN Chief Information Security Officers (CISOs) and Security Coordinators a forum in which to share experiences. The background and aims of the project were presented at the October 2020 SIG-ISM meeting following which the needs and willingness of the NRENs to invest in VA were assessed through a poll.

Other communications and stakeholder engagement activities undertaken to raise awareness of vulnerability assessment include a SOCTools Workshop organised in collaboration with Task 3.1 in Amsterdam in December 2019, attended in person or remotely by over 50 participants from more than 40 organisations, and the publication of an article on proactive security monitoring in *CONNECT* magazine [D. Heed-Interview].

From this engagement it emerged that the NRENs have differing requirements and levels of maturity in this area. Some NRENs are mature and have incorporated VA into their security infrastructure, while others do not yet have the processes and tools in place to protect themselves and their constituents. Moreover, where VA services are in place, they are often local, benefiting only a small section of the overall interconnected GÉANT network.

The team also ran a pilot project, the results of which are reported in this document, to evaluate the features of various open source and commercial vulnerability scanner offerings, testing their performance and detection efficiency. The project assessed both the tools and services currently deployed by the NRENs participating in Task 3.2 (LRZ/DFN, PSNC, SUNET), and those of other vendors whose offerings met the requirements. The results of these evaluations and comparisons are presented in Section 4 and in the Appendices.

Based on the findings of both the engagement activities and the pilot project, the decision was made to continue to support delivering both a good value proposition to the NRENs for acquiring commercial vulnerability scanning software or services and an alternative open-source solution.

The Task compiled a list of requirements for a commercial scanner (Appendix H) and worked in collaboration with GÉANT procurement to prepare a tender, which is currently underway, to purchase the right to use scanning services either as cloud services or hosted on virtual machines on premises. The ultimate objective of the procurement is to build a foundation for all connected NRENs enabling them to make use of vulnerability scanner software, systems or- services that are best suited to their

situation, while providing the GÉANT community with better and more reliable insights on how secure or vulnerable its networks are via regular reporting mechanisms.

Discussions are ongoing as to what the best model would be for delivering these kinds of propositions, whether GÉANT as a reseller, ready-to-use contracts, cloud-based licence model, negotiated discounts (to avoid the additional vendors' charges based on the number of IP addresses, which in the case of R&E organisations is always high) or other models. Enlisting the commitment of the NRENs is essential in this undertaking, as without it certain options cannot be considered and the only remaining solution in this scenario would likely be to negotiate discounts for one or more offerings. To date, very productive and constructive conversations have taken place with NRENs about opportunities and ways to collaborate.

At the same time, recognising that keeping a scanning infrastructure up to date can be a daunting task for small NRENs, the team has developed a generic, scalable open-source-based solution using existing tools, with additional features such as federated login and import of vulnerability feeds. The solution takes the form of a container (a specific software package) with all the capabilities needed to either carry out manual scanning or for integration with Security Operations Centre (SOC) tools.

To use both open-source components and professional scanning capabilities, the team decided to base its solution on the open-source online vulnerability assessment scanner OpenVAS [OpenVAS] from Greenbone [Greenbone]. The task has written its own programs to improve the functionality of the existing tools and their integration with OpenVAS, leveraging the considerable knowledge in this area of the team and the community. This solution has been included in the pilot project evaluations and comparisons.

In addition, Task 3.2 has signed a Memorandum of Understanding (MoU) with vulnerability management specialists Holm Security [Holm_Security], whose scanner is based on OpenVAS. The agreement will give the GÉANT community access to Holm's vulnerability assessment feed for the OpenVAS scanner, enabling the vulnerability checks to be kept valid and up to date. It will also give access to Holm's knowledge and expertise, including the dissemination of information on how to optimise processes and how to write testing-plugins to verify vulnerabilities.

Collaboration with companies such as Holm Security, security researchers, GÉANT's NREN partners and other interested parties in the area of vulnerability assessment is essential to raise the level and extent of protection against cyber criminals and to prevent research data and identity theft.

The team has also looked into some other aspects of deploying Vulnerability Assessment as a Service, in particular the legal implications of scanning a network for vulnerabilities. Task 3.2 investigated Swedish, German and Polish law in the scenario where a vulnerability scan disrupts services or gains access to "sensitive data/systems", concluding that in all three countries it is unlawful to scan without consent.

By sharing experiences and best practices, NRENs can be supported in implementing vulnerability scanning facilities and services in an efficient way. Going forward the Task will continue in its work to provide both tools and knowledge to the NREN community, including the development of open-source components such as the easy-to-access API and reporting functionality implemented in the OpenVAS platform, and facilitating their integration with other tools. Currently, T3.2 is working on providing a scanning service capable of supporting federated logons for all institutions.

4 Pilot Installations and Summary of Experiences

One of the challenges for NRENs is the huge amount of IP-numbers and active assets connected to their networks. This requires an inventory of assets to be completed fairly quickly and accurately. To do this, splitting the scanning engine from the inventory engine seems to be a better approach than running the inventory and scanning sequentially for each resource or service.

WP8 T3.2 conducted a pilot project to test different vulnerability scanning tools in production or test environments in various NRENs and institutions. The project tested both open-source and commercially available scanning tools to evaluate their features, testing their performance and detection efficiency. The tests were performed in a distributed manner at volunteering institutions and NRENs.

Laboratory speed tests for inventory scans where performed at PSNC and scanned systems where primarily located at LRZ(Germany), SUNET(Sweden) and GÉANT(Netherlands and England). Findings for specific hosts are not included in this document as these were reported and handed over to the local CERT/CSIRT functions. Joint testing was also carried out with other organisations but these results are excluded from the comparisons as these were mostly single scans with one scanner.

Running a comparison of different tools also helped clarify the requirements (Appendix H) towards an open GÉANT procurement for a commercial vulnerability scanner [RFP_VMS], which is currently underway.

The following vulnerability scanners were evaluated as part of the project:

- OpenVAS / GVM
- OpenVAS / GÉANT API+GUI
- Greenbone appliance
- Detectify
- Outscan
- Nessus Professional

Each scanner was evaluated, and a grade given (out of ten), for the following aspects:

- Installation process.
- Management.
- Graphical user interface (GUI).
- Reports and results.

- Logon security.
- Performance.
- Licence and cost.

It should be noted that all tests apart from the speed tests were carried out in various production environments and results can vary depending on which services were present in the environment at the time the scan was conducted, as well as due to different tools being limited by licensing for specific environments. The commercial tools tested in production were those pre-purchased by NRENs or GÉANT. Holm Security tools should have been included but due to a lack of time and resources these will need to be evaluated at a later stage.

The findings of the pilot project have shown that different tools have different strengths, with scan results and speed varying greatly for large environments. The market today is fairly mature and most tools detect most vulnerabilities but do not check for every known vulnerability so there often exists a need to run tools in parallel.

For example, it can be observed that the open source-based solution with OpenVAS does not discover all CVEs in web applications and when compared with commercial scanners also does not update the feed in a timely manner for detecting new CVEs.

At the same time, all commercial scanners can detect tens of thousands of CVEs, but none of them produce complete results and often they do not scan the individual CVEs but rather check the status of a system. This means quantitative comparisons are not useful to assess whether one scanner always performs better than another.

Therefore, NRENs should not rely on a single scanner, and those that have a commercial scanner should still compare results with those of the open source alternative solution.

Verifications should also be carried out in collaboration with IT operations teams. For the purposes of building prioritised reports and selecting the problems to address first, the exposure and criticality of services must be considered in the decision-making process.

A summary comparison of the tested tools is shown in Table 4.1 below and a short review of each tool is presented in the sections that follow. The results from the pilot project tools evaluation are presented in the appendices at the end of this document, including comparisons between the scan times of four scanners (Appendix G). The information on these and other tools is continuously updated on the Vulnerability Assessment as a Service wiki [Wiki_VAaaS].

	Nessus Profess.	Outscan	Green-bone	OpenVAS/ GÉANT API+GUI	OpenVas/ GVM	Detectify
Installation process	✓	✓	✗	✗	✗	✓
Management	✓	✓	✓	✗	✓	✓
GUI	✓	✓	✓	✗	✓	✓
Reports and results	✓	✓	✓	✓	✓	✓
Logon security	✗	✓	✗	✓	✗	✓
Performance	✓	✓	✗	✗	✗	✓
Licence and cost	✗	✓	✓	✓	✓	✗

Table 4.1: Comparison of vulnerability scanners tested capability

4.1 OpenVAS Scanner with Greenbone GVM

Overview

Greenbone is the maintainer of OpenVAS and the community feed. They have a manager web portal to control scans and reports. Greenbone Vulnerability Management (GVM) [Greenbone_GVM] is the free offering which is used by many NRENs.

Installation process

This requires many steps to complete it manually but Kali Linux can be used, which has a more streamlined process.

(grade 3)

Management

Most basic functionality exists. (grade 6)

GUI

Fairly similar to the commercial Greenbone. Screenshots available on the project wiki. (grade 5)

Reports and results

XML, DOCX and CSV are supported. (grade 7)

Logon security

Username and password. (grade 3)

Performance

Fairly bad, multiple examples online and from the project show large scans taking days or stopping responding. Number of max hosts to scan in one scan: 4095. (grade 3)

Licence and cost

Open source (grade 10)

4.2 OpenVAS Scanner with GÉANT API+GUI

Overview

As part of its work, Task 3.2 has developed a generic, scalable open-source-based solution based on the codebase of OpenVAS and the Greenbone project [OpenVAS_GÉANT]. The team's aim is to include support for eduGAIN authentication and integration of Holm Security's feed for OpenVAS. In addition, an API was created to automate queries from SOCTools, the interoperable set of Security Operations Centre (SOC) tools created by Task 3.1.

Installation process

The installation process is documented in the GÉANT Project GitLab [GÉANT_GitLab_VA].

(grade 5)

Management

Basic functionality exists to create users, teams, scans and reports.

(grade 4)

GUI

This is fairly basic just to demonstrate capabilities of automating through the API.

(grade 4)

Reports and results

PDF, XML, DOCX and CSV are supported

(grade 7)

Logon security

Username and password. With SAML integration this improves the ease of administration and access to the system. This will be added as functionality during 2022.

(potential grade 10)

Performance

Based on OpenVAS (grade 3)

Licence and cost

Open-Source (grade 10)

4.3 Greenbone appliance scanner

Overview

Task 3.2 purchased one of the smaller appliances from Greenbone Networks GmbH [Greenbone_Scanner]. The team's aim is to use this commercial repackaging of open source software together with Greenbone Networks GmbH´s commercial feed as a second reference when conducting system scans. The system is a rack based system accessible for the GÉANT community upon request.

Installation process

Installation is fairly straightforward as it comes in a physical appliance. Networking needs to be set up through the physical interface to create the first admin account. User accounts and their rights are created and set up from the web service. (grade 8)

Management

This involves automatically updating vulnerability checks triggering generation of reports. Updates are made by manually updating the server instance.

(grade 8)

GUI

The GUI is a web-based static generated page that is a bit dated but efficient.

(grade 8)

Reports and results

Presenting results data requires setting a quality of detection (QoD) scoring, with 30% being lower than standard but fairly accurate on most findings. Delta reports can be generated and exported in a variety of formats including XML, CSV, PDF and HTML.

(grade 9)

Logon security

Username and password. No integrated SSO/SAML or SMS-verification.

(grade 3)

Performance

Limited to performing scans of 2000 IP-numbers per day but in reality may scan more or less than that depending on the scanning profile. One /24 with a fullscan took over one day so parallelism should also be considered.

(grade 5)

Licence and cost

Appliance investment is a one-off cost. Task 3.2's installation is limited to 2000 IPs per day with a support contract valid until 2025..

(grade 7)

4.4 Detectify

Overview

Detectify [Detectify] is primarily a web scanner that has a crawler that can identify applications and run tests towards them. Detectify has a large crowdsource community that share new tests and methodologies.

Installation process

This is a cloud service. In order to authenticate as a legitimate owner of a domain, a token file should be placed in the DNS. Assets can then be scanned and team members added via email.

(grade 8)

Management

Detectify is fairly easy to manage as there are not that many options or changes that can be made towards the scanning itself. They provide an API for most integrations as well as Slack, Jira and Splunk integration.

(grade 8)

GUI

The GUI is web based and is fairly quick and responsive.

(grade 7)

Reports and results

Results are published as profiles under their main root asset. The date and time of the last scan are shown and data exported in most common formats such as PDF, JSON and XML as well as using integrations to Trello and Jira for exports of data.

(grade 9)

Logon security

Google integrated or SSO. The team used multifactor for the Google account with a Yubikey as an extra factor.

(grade 9)

Performance

Scanning is slow but comprehensive with regard to web application security. However, it is only performed on a small number of hosts. Within the test NREN, 12 scanning profiles for different universities and 2 main scanning profiles, all with 4 assigned web applications to monitor, were set up.

(grade 6)

Licence and cost

These are on a per-asset basis including for a separate add-on for asset-monitoring scans, making it expensive for large coverage. This is a niche tool for prioritised web services that often change.

(grade 4)

4.5 Outscan

Overview

Outscan is a hybrid-approach scanner infrastructure with capabilities for performing an inventory of services both from an external scanning point of view and using internal scanners for access behind NAT or firewall instances [Outscan].

Installation process

Enrollment is through a starting admin account via email where a mobile phone number can also be added as extra authentication via SMS. Since this is a web application, all updates are taken care of by the service provider.

(grade 9)

Management

Management is through a web application within the browser. It is an enterprise product, which makes setup highly customisable in terms of reporting and asset management.

(grade 8)

GUI

The GUI is fairly fast but with thousands of results sorting can take time. Work on improving the user experience is ongoing.

(grade 9)

Reports and results

The reports are available in most formats such as XLS, PDF and XML. PGP encryption and .zip (with or without password) is also offered.

(grade 8)

Logon security

Logon security is currently Username+Password+SMStoken. Following a dialogue with the service provider, the ability to integrate with SAML support has been added. At the time of writing, the collaborating NREN has not implemented the SAML-Proxy needed for multiple organisations to log on independently.

(grade 9)

Performance

The hybrid approach provides good scalability but the central cloud node has had performance issues when multiple scans for multiple organisations are shared such as Log4J which triggered many massive scans.

(grade 8)

Licence and cost

The licence cost depends on the number of tracked live IP assets. During the procurement, the local NREN was offered a very large discount for 100,000+ active IP assets.

(grade 8)

4.6 Nessus Professional

Overview

Nessus in an on-premises installation and an all-purpose scanner [Nessus_Prof].

Installation process

The installation process is fast and easy. The Nessus application is updated via the web interface: a reminder appears as a notification and the administrator can easily initiate the update.

(grade 9)

Management

Management is through a web application within the browser. It is a single user licence.

(grade 8)

GUI

The GUI is fast and easy to use.

(grade 9)

Reports and results

Reports are only available in HTML and CSV but have plenty of formatting options.

(grade 7)

Logon security

Log on is with username and password.

(grade 6)

Performance

The single on-premises installation makes the scanner work fast since it does not compete with other users.

(grade 9)

Licence and cost

The licence cost is for a single user but for an unlimited amount of scans and IP numbers in the database.

(grade 4)

5 Conclusions

The number of vulnerable applications and systems on the internet continues to increase and during 2021 alone, over 20,000 new vulnerabilities were discovered. For organisations it has therefore become critical to monitor services and infrastructure to provide an effective response to these threats.

With exploits being sold on an open market, it is critical to keep on top of known vulnerabilities. Most known security vulnerabilities are tracked in the [CVE] database. Unpublished vulnerabilities are those that have not yet been assigned a CVE ID and are not listed in the database. Zero-day (0day) attacks on such vulnerabilities are highly valuable for cyber criminals and state actors.

In addition, following vendor and open-source information allows organisations to keep track of vulnerabilities. One of the best sources of documentation for developers to help them avoid the worst errors is the OWASP Top 10 Web Application Security Risks. This is continually updated and the latest best practices are published on [OWASP_Top10].

In performing vulnerability assessments, one of the challenges the NRENs face is the huge amount of IP-numbers and active assets connected to their networks. This requires an inventory of assets to be completed fairly quickly and accurately. To do this, splitting the scanning engine from the inventory engine seems to be a better approach than running the inventory and scanning sequentially for each resource or service.

The pilot project run by WP8 T3.2 tested both open-source and commercially available scanning tools to evaluate their features, testing their performance and detection efficiency. The tests were performed in a distributed manner in production and test environments at volunteering institutions and NRENs. The commercial tools tested in production were those pre-purchased by NRENs or GÉANT.

The most critical aspects of scanning systems for NRENs lie in performance and detection quality. Based on the evaluation carried out by WP8 T3.2, the following process is recommended order to optimise results:

- Tools such as Nmap can help perform initial discovery and groupings of an organisation's relevant assets. This is especially useful if licence limitations exists on conducting large sweeps of unknown networks.
- Compare or create a list of IT assets (CMDB) to scope the "live" machines that should be focused on to be included in the real scanner.
- Perform tests of different types of scanners and compare the results (detections of vulnerabilities and speed of scan to verify that it completed without stalling).
- Re-evaluate whether detections are helping IT operations (and the CISO/Security function) to identify assets that need to be upgraded or have changes made to them.

- If needed, rescan the assets to verify that improvements have been made and compliance requirements are met.

Based on both the findings of the pilot project and previous engagement with NREN CISOs and Security Coordinators, WP8 T3.2 considers that the best direction for its future work is to continue to support both the delivery of a good value proposition to the NRENs for acquiring commercial vulnerability scanning software or services and the development of an alternative open-source solution.

The Task has compiled a list of requirements for a commercial vulnerability scanner (Appendix H) that have been fed into an open GÉANT tender procedure [RFP_VMS] prepared in collaboration with the GÉANT Procurement team.

As most scanners tested, with the exception of Detectify, are focused holistically on open ports and services without specifically deep diving into web services and their exposed APIs, these functionalities are currently out of scope for the project and may require setting up a specific subtask to address and compare them. Some requirements compelling vendors to improve and be transparent on this subject have nevertheless been included in the above procurement.

Task 3.2 has also signed a Memorandum of Understanding (MoU) with vulnerability management specialists Holm Security [Holm_Security] that will give the GÉANT community access to Holm's vulnerability assessment feed for the OpenVAS scanner, enabling vulnerability checks to be kept valid and up to date, as well as to Holm's knowledge and expertise.

Going forward in the GN4-3 project, the Task will continue to work towards integrating its open-source API and reporting functionality components implemented in the OpenVAS platform with other tools, as well as with a commercial offering tailored to the needs of the NRENs in terms of scale and pricing.

Currently, T3.2 is working on providing a scanning service capable of supporting federated logons for all institutions.

Appendix A Different Scan Techniques with Nmap

A.1 Scan Types

Types of Nmap scans:

- TCP SYN scan

  A method of scanning ports using the TCP protocol, which involves sending the initial TCP SYN packet and waiting for a response in the form of a TCP SYN/ACK packet, but where the last TCP ACK packet is not sent back, which eventually opens the connection.

- TCP connect scan

  A standard scanning method that creates connections to each specified port on a selected host. This is slower than the TCP SYN method, but does not require elevated privileges.

- UDP scan (-sU)

  A scan method that is used for testing ports using the UDP protocol. Due to the specifics of the protocol, port scanning may take a long time.

A.2 Host Discovery

To be able to scan ports, it is also necessary to indicate in advance whether a given host is available on the network.

- No ping (-Pn)

  If the –Pn option is chosen, *nmap* assumes that each of the scanned hosts is available. This method is very time-consuming due to the need to scan every host on the network even if unavailable.

- Port list (TCP/UDP) (-PS/-PU)

  These methods detect host activity by checking if at least one port from the list is open. In ICMP traffic filtering, this allows hosts to be detected if the presence of specific services is suspected.

A.3 Scan Multiple Hosts

To scan multiple hosts on the network, a specific range of IP addresses has to be provided. IP addresses can be presented using the following notations:

- Using CIDR notation

  The use of CIDR notation allows the entire selected subnet to be scanned, e.g. 192.168.0.0/24 will scan all addresses from 192.168.0.1-192.168.0.254.

- Using Address range

  It is also possible to provide addresses in the form of a range. Many forms are allowed:

  ```
  168.0.36-128
  ```

  ```
  10.0-254.2
  ```

- Using a comma-separated address list

  If there is a need to analyse specific addresses, they should be separated with a comma:

  ```
  168.0.1,2,3,4
  ```

  ```
  10,1,2,3.0.0.1
  ```

- Using domain address

  Finally, it's also possible to provide hosts as domain names:

  ```
  com
  ```

- Excluding hosts with --exclude parameter

  Additionally, excluding IP addresses with an extra argument --exclude is also possible. This option is useful when one is aware of the availability of some addresses, because on the one hand it is not necessary to verify their availability (saving resources and reducing the duration of the operation), while on the other it means key services are not exposed to overload.

All of these techniques may be combined with each other, thereby giving flexibility where it is necessary to scan an unusual range. Appropriate address range adjustment will reduce the time that would be wasted on scanning unnecessary hosts. Sample Query:

```
nmap scanme.nmap.org 8.8-10.8.0,1,2,3 --exclude 8.8.8.8
```

If an organisation has a large number of addresses that will be constantly scanned, the host list can be provided in a file by using the additional parameter –iL. The host list file must have a list of addresses separated from each other with a space, tab or enter character, and each entry should be in a form consistent with the above principles.

```
nmap -iL <<input_file>>
```

A.4 Service Detection – Determine Services Running on the Ports

Specifying the service is done by default. However, it is recommended to use the additional parameter *-sV*, which can estimate the version of the running service.

When scanning ports, *nmap* detects services running on them with some accuracy. However, by default, services based on the port being used are indicated using some default rules, e.g. the HTTP port is indicated on port 80. If an exemplary service runs on a non-standard port, *nmap* by default can have problems identifying it. Using the *-sV* parameter enables detection of when this result is not a false-positive. In the case of the actual occurrence of a given service, the software used will be specified (in most cases), while in the case of another (unidentified) service, the software will not be recognised.

A.5 OS Detection – Determine Server Version

Nmap can additionally scan the system to determine which operating system is running. *Nmap* software allows an additional scan to be run using the *–O* switch.

The use of this switch causes the use of numerous methods that are able to determine with certain probability which operating system is used based on specific parameters in the returned response. The following is the answer for the Debian virtual machine version "*Linux debian 4.19.0-6-amd64 # 1 SMP Debian 4.19.67-2 + deb10u2 (11/11/2019) x86_64 GNU / Linux*":

```
MAC Address: 08:00:27:2B:18:51 (Cadmus Computer Systems)

Aggressive OS guesses: Linux 2.6.32 - 3.9 (96%), Netgear DG834G WAP or
Western Digital WD TV media player (95%), Linux 2.6.32 (95%), Linux 3.1
(94%), Linux 3.2 (94%), AXIS 210A or 211 Network Camera (Linux 2.6)
(94%), Linux 3.3 (94%), Linux 2.6.32 - 2.6.35 (94%), Linux 2.6.32 - 3.2
(94%), Linux 3.0 - 3.9 (93%)


No exact OS matches for host (test conditions non-ideal).


Network Distance: 1 hop
```

In the case presented, it can be seen that the operating system probably used was found but is not fully compatible with the system actually used. In addition, it is worth paying attention to the discrepancy in the case of the MAC address, which begins with a characteristic string for the VirtualBox software but was not recognised.

A.6 NSE Scripts – List of Useful NSE Scripts with Example Usages

An additional source of information may be the use of Nmap Scripting Engine (NSE) scripts, which significantly extend the functionalities offered by *nmap* software. The user is also able to create their own scripts in Lua. In the official *nmap* documentation a guide is provided on how to create NSE scripts https://nmap.org/book/nse-tutorial.html. Example NSE scripts are also available in the *github* repository (https://github.com/nmap/nmap/tree/master/scripts). In order to facilitate this work, several useful scripts for the majority of users have been identified:

A.6.1 http-google-malware

This script enables checking of whether the host is on the blacklist provided by the Google organisation. In order to use a given plugin, it is necessary to obtain a dedicated API key.

```
nmap --script http-google-malware --script-args http-google-
malware.api=<API_KEY> <target>
```

A.6.2 dns-brute

For domain name hosts it is possible to perform a brute-force attack in order to extract all of the domains within the DNS zone. The result for the domain google.com is given below as an example.

```
nmap -p80 --script dns-brute google.com
```

Result:

```
Starting Nmap 6.40 ( http://nmap.org ) at 2020-03-31 08:31 EDT
Nmap scan report for google.com (216.58.215.110)
Host is up (0.0042s latency).
rDNS record for 216.58.215.110: waw02s17-in-f14.1e100.net
PORT   STATE SERVICE
80/tcp open  http

Host script results:
| dns-brute:
|   DNS Brute-force hostnames
|     corp.google.com - 173.194.222.129
|     www.google.com - 172.217.20.164
|     whois.google.com - 216.239.34.22
```

```
|     corp.google.com - 2a00:1450:4010:c0b:0:0:0:81
|     www.google.com - 2a00:1450:401b:802:0:0:0:2004
|     whois.google.com - 2001:4860:4802:34:0:0:0:16
|     mail.google.com - 172.217.20.5
|     ldap.google.com - 216.239.32.58
|     mail.google.com - 2a00:1450:400d:805:0:0:0:2005
|     ldap.google.com - 2001:4860:4802:32:0:0:0:3a
|     blog.google.com - 172.217.16.41
```

A.6.3 http-enum

In the case of web applications, it is possible to use the script *http-enum*, which allows the listing of popular pages that will provide additional details of the http service.

```
nmap -p80 --script http-enum google.com
```

Result:

```
Starting Nmap 6.40 ( http://nmap.org ) at 2020-03-31 08:53 EDT
Nmap scan report for google.com (216.58.215.110)
Host is up (0.0041s latency).
rDNS record for 216.58.215.110: waw02s17-in-f14.1e100.net
PORT   STATE SERVICE
80/tcp open  http
| http-enum:
|_  /partners/: Potentially interesting folder
```

A.6.4 Firewalk

This script allows firewall rules to be set using the *firewalk* technique. It involves sending TCP and UDP packets through every hop. Each time there is a packet with a TTL value greater than one, the packet will be sent onwards. If the organisation is filtering traffic the response will be dropped. But with the TTL exceeded it will be an ICMP message indicating that it was a firewall active and that the active port could be there and active. A firewall could either silently drop or actively reset a connection that is unwanted.

Host script results:

```
| firewalk:
| HOP  HOST       PROTOCOL  BLOCKED PORTS
```

```
|_1   10.0.2.2  tcp     1,3-4,6-7,9,13,17,19-20
```

A.7 Output Formats

It is possible to specify the format of nmap output that can be processed, for the purposes of using the data for other tools. However, despite the existence of different formats, there is no officially available functionality for converting output to JSON. The format is selected by entering the appropriate flag.

A.7.1 Normal Output

Writing to the output in standard format can be obtained by using -oN flag:

```
-oN <filename>
```

The output result is also saved to a file.

```
# Nmap 7.80 scan initiated Wed Apr 15 07:21:29 2020 as: nmap -oN
C:\\Users\\dawid\\output.normal 8.8.8.8

Nmap scan report for dns.google (8.8.8.8)

Host is up (0.017s latency).

Not shown: 998 filtered ports

PORT  STATE SERVICE

53/tcp  open  domain

443/tcp open  https


# Nmap done at Wed Apr 15 07:21:45 2020 -- 1 IP address (1 host up)
scanned in 16.59 seconds
```

A.1.1 XML Output

The output record in XML format can be obtained by using -oX flag:

```
-oX <filename>
```

The output result is also saved to a file in XML format.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE nmaprun>

<?xml-stylesheet href="file:///C:/Program Files (x86)/Nmap/nmap.xsl"
type="text/xsl"?>
```

```
<!-- Nmap 7.80 scan initiated Wed Apr 15 07:22:48 2020 as: nmap -oX
C:\\Users\\dawid\\output.xml 8.8.8.8 -->

<nmaprun scanner="nmap" args="nmap -oX C:\\Users\\dawid\\output.xml
8.8.8.8" start="1586928168" startstr="Wed Apr 15 07:22:48 2020"
version="7.80" xmloutputversion="1.04">

<scaninfo type="syn" protocol="tcp" numservices="1000" services="1,3-
4,6-7,9,13,17,19-26,30,32-33,37,42-43,49,53,70,79-85,88-90,99-
100,106,109-
111,113,119,125,135,139,[…],62078,63331,64623,64680,65000,65129,65389"/
>

<verbose level="0"/>

<debugging level="0"/>

<host starttime="1586928169" endtime="1586928184"><status state="up"
reason="echo-reply" reason_ttl="52"/>

<address addr="8.8.8.8" addrtype="ipv4"/>

<hostnames>

<hostname name="dns.google" type="PTR"/>

</hostnames>

<ports><extraports state="filtered" count="998">

<extrareasons reason="no-responses" count="998"/>

</extraports>

<port protocol="tcp" portid="53"><state state="open" reason="syn-ack"
reason_ttl="118"/><service name="domain" method="table"
conf="3"/></port>

<port protocol="tcp" portid="443"><state state="open" reason="syn-ack"
reason_ttl="119"/><service name="https" method="table"
conf="3"/></port>

</ports>

<times srtt="16171" rttvar="6464" to="100000"/>

</host>

<runstats><finished time="1586928184" timestr="Wed Apr 15 07:23:04
2020" elapsed="16.80" summary="Nmap done at Wed Apr 15 07:23:04 2020; 1
IP address (1 host up) scanned in 16.80 seconds" exit="success"/><hosts
up="1" down="0" total="1"/>

</runstats>

</nmaprun>
```

A.7.2 Grepable Output

To search a large amount of data faster, the output can be saved in a format that allows to search for hosts using the grep filter command. In this case the –oG flag can be used:

```
-oG <filename>
```

The output result is also saved to a file.

```
# Nmap 7.80 scan initiated Wed Apr 15 07:25:02 2020 as: nmap -oG
C:\\Users\\dawid\\output.grep 8.8.8.8

Host: 8.8.8.8 (dns.google)   Status: Up

Host: 8.8.8.8 (dns.google)   Ports: 53/open/tcp//domain///,
443/open/tcp//https/// Ignored State: filtered (998)

# Nmap done at Wed Apr 15 07:25:18 2020 -- 1 IP address (1 host up)
scanned in 16.57 seconds
```

Additionally, it is noted that in case of passing the '-' character as a parameter for the file name, the scan result can be transferred to the standard program output.

A.8 Useful Commands

To sum up, to enable beginners to more easier apply the information listed above in using the Nmap tool, the following commands are some of the most useful:

- Scan host without checking its availability with version detection

  ```
  nmap -Pn -sV 192.168.0.10
  ```

- Scan the network at 10.0.0.0/24 analysing all TCP and UDP ports

  ```
  nmap -sU -sT -p- 10.0.0.0/24
  ```

- Scan all addresses from the list.txt file paying attention to the 3328 most popular TCP ports

  ```
  nmap -iL list.txt --top-ports 3328
  ```

- Scan ports 1-1024 for the UDP protocol for testpage.com

  ```
  nmap -sU -p1-1024 testpage.com
  ```

- Scan port 21 TCP and port 53 UDP for address 10.0.0.1

  ```
  nmap -sU -sT -p T:21,U:53 10.0.0.1
  ```

- Scan at increased speed

```
nmap -t4 10.0.0.1
```

- Scan discreetly

```
nmap -t1 10.0.0.1
```

A.9 Nmap Summary

The *nmap* tool is very extensive and can be used by novice users, e.g. administrators, who want to verify the configuration on controlled servers. It can also be used by security testers for network reconnaissance, as well as more advanced activities through the use of NSE scripts. These scripts are available from the Nmap repository and can also be created for specific needs.

Appendix B Port Scanners Comparison

Port scanners are one of the most basic tools that are used in the initial stage performing vulnerability assessments as the first information to be gathered are lists of open ports that are used to determine which working services are on the server. Tests were executed on the three most popular port scanners: Nmap, MASSCAN and Unicornscan. To provide a unified environment for the tests, a virtual environment containing the following three popular services was set up:

- HTTP server (apache2) on port 80
- SSH server on changed port 2137
- Database server (Mongodb) on port 27017

B.1 Nmap

Nmap is a free and open source utility for network discovery and security auditing [Nmap]. It is under GPL Licence. It is often used by administrators for network inventory.

The listing below shows an execution of the Nmap scanner and the result from scanner itself:

```
nmap 192.168.0.2 -p 1-65535

Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-23 05:19 EDT

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is
disabled. Try using --system-dns or specify valid servers with --dns-
servers


Nmap scan report for 192.168.0.2

Host is up (0.000076s latency).

Not shown: 65532 closed ports

PORT        STATE SERVICE

80/tcp      open  http

2137/tcp  open  connect

27017/tcp open  mongod

MAC Address: 08:00:27:2B:18:51 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 3.61 seconds
```

An additional option is to extend running service detection by using the –A parameter. This performs all tests included in Nmap:

- OS detection,
- Version detection
- Script scanning
- traceroute

```
nmap 192.168.0.2 -p1-65535 -A

Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-26 04:18 EDT

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is
disabled. Try using --system-dns or specify valid servers with --dns-
servers

Nmap scan report for 192.168.0.2

Host is up (0.00031s latency).

Not shown: 65532 closed ports

PORT        STATE SERVICE VERSION

80/tcp      open  http  Apache httpd 2.4.38 ((Debian))

|_http-server-header: Apache/2.4.38 (Debian)

|_http-title: Apache2 Debian Default Page: It works

2137/tcp  open  ssh    OpenSSH 7.9p1 Debian 10 (protocol 2.0)

| ssh-hostkey:

|   2048 88:15:51:aa:5d:00:66:c3:7c:06:d3:f5:f4:70:2b:1b (RSA)

|   256 71:57:f6:77:de:47:9f:4d:be:c3:35:20:5e:96:3b:6f (ECDSA)

|_  256 0f:58:2f:28:5e:1c:65:d9:f1:3d:2a:ec:63:ec:ac:40 (ED25519)

27017/tcp open  mongodb MongoDB 4.2.0

|_mongodb-databases: ERROR: Script execution failed (use -d to debug)

|_mongodb-info: ERROR: Script execution failed (use -d to debug)

MAC Address: 08:00:27:2B:18:51 (Oracle VirtualBox virtual NIC)

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux kernel:3 cpe:/o:linux:linux kernel:4

OS details: Linux 3.2 - 4.9

Network Distance: 1 hop

Service Info: OS: Linux; CPE: cpe:/o:linux:linux kernel
```

```
TRACEROUTE
HOP RTT     ADDRESS
1   0.31 ms 192.168.0.2

OS and Service detection performed. Please report any incorrect
results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.89 seconds
```

The scan with the "-A" flag takes much longer time. Result was 11.89s but the results are much more precise. They determine used service based on their responses. If a service is available on a port other than standard, this kind of scan can detect its type. For example, on the test machine the standard ssh service port was changed from 22 to 2137 and the standard scan detected it as "connect".

Another useful option is the –Pn switch which assumes that the scanned port is online. This allows to perform a scan with filtered network traffic. Every port is scanned without host discovery phase. The scanner starts to communicate with ports and waits for a response.

B.2 MASSCAN

MASSCAN is an internet-scale port scanner. In the official repository [MASSCAN] it is considered the fastest scan utility, able to perform a scan of the entire internet in 6 minutes. It is distributed using a GPL licence.

```
masscan -p1-65535 192.168.0.2 --rate 1000000000 --interface eth0 --
router-ip 192.168.0.2
Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2019-09-23 10:37:53
GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 80/tcp on 192.168.0.2
Discovered open port 27017/tcp on 192.168.0.2
Discovered open port 2137/tcp on 192.168.0.2
```

The presented software checks opened ports but do not show working services. Research can be extended with the –banner option to determine what working software is present on the detected port. However, in the tested environment this option did not detect any specific working software.

B.3 Unicornscan

Unicornscan is an information-gathering utility created for security research and testing communities [Unicornscan]. It was designed to be a scalable, accurate and efficient engine. Unicornscan is distributed under GPL licence.

```
unicornscan 192.168.0.2 -p1-65535
TCP open                 http[   80]       from 192.168.0.2  ttl 64
TCP open          connect[ 2137]    from 192.168.0.2  ttl 64
TCP open          unknown[27017]    from 192.168.0.2  ttl 63
```

Despite its popularity, the tool is no longer under development. Its last commit was on 28 August 2012, so any bugs must be fixed by the user. Unicornscan is not recommended for use in security assessment products.

B.4 Time Comparison

To determine the fastest scanner, scan time duration criteria was used. The test was performed 12 times scanning the same host. The average time was calculated based on the formula:

$$\frac{\sum_{i=1}^{n} T(i) - \mathrm{T}_{max} - T_{min}}{n}$$

Nmap	MASSCAN	Unicornscan
5.117s	8.056s	235.006s

The main goal of the scan was to identify open ports on a test machine. This was the smallest functionality available for each of the selected tools. From the comparison run, it was observed that the fastest port scanner is Nmap.

B.5 Functionality Comparison

A full comparison of the available tools must also include their functionality. Those functionalities that may be needed to conduct security assessments were selected for this purpose:

Functionality	Nmap	MASSCAN	Unicornscan
Port scanning	YES	YES	YES
Host discovery	YES	YES	YES
Version detection	YES	NO	NO
Stealth Mode	YES	YES	NO
Consider active scan	YES	YES	NO
Supported	YES	YES	NO

Appendix C Test Case – Holm Security vs. Greenbone Appliance

Task 3.2 scanned an NREN's /24 public network comprising only Virtual Machines using both Holm Security and a Greenbone appliance. The two scans were run almost simultaneously. The resulting, reports are compared below. Greenbone was configured with the Full and Fast Scan Profile and the Standard List of Ports. Holm Security was configured with the Network scan profile – Standard.

Host Up Detection Holm uses ICMP and the TCP Ports 21, 22, 23, 25, 53, 80, 110, 111, 135, 139, 143, 443, 445, 993, 995, 1723, 3306, 3389, 5900 and 8080.

Host Up Detection in Greenbone is using Nmap with ping and TCP ports 80,137,587,3128 and 8081.

Ports Scanned from Holm can be found here: https://support.holmsecurity.com/hc/en-us/articles/212609249

Ports Scanned from Greenbone: Using the 4481 ports defined in OpenVAS default.

A /24 with 254 possible hosts was scanned but at present only 193 hosts are active in this subnet. This does not mean that all hosts are reachable. Some have blocked ICMP responses and some might not react on the standard ports tested by the scanners but offer other services. Both scans were conducted from the internet since the Greenbone appliance was hosted by SUNET and the Holm Scanner runs on the Google Cloud.

C.1 Overview of the Results

An overview of the results is provided in the table below.

Product	Dura-tion	Found Hosts	Reported Vulnera-bilities (with Log / Info)	Reported Vulnera-bilities (Filtered)	Low	Med-ium	High	Crit-ical
Holm	8h 46m	65	8603	432	59	251	110	12

Product	Dura -tion	Found Hosts	Reported Vulnera-bilities (with Log / Info)	Reported Vulnera-bilities (Filtered)	Low	Med -ium	High	Crit-ical
Greenbon e	nearl y 32h	171	10893	802	179	546	77	n/a

Note: Greenbone does not have a Critical category

It is an unexpected result that Holm has found far fewer active hosts than Greenbone. Despite this, the number of vulnerabilities detected by Greenbone is only double that detected by Holm, whereas Greenbone found almost three times as many hosts as Holm. Even when using local tests, the team found that Nessus and Greenbone do not always detect an active host, even using very exhaustive settings. In some cases active host detection had to be disabled in Greenbone and the scanner instructed that it should expect the host to be up and continue to test every port (which severely slows down scanning due to waiting on long timeouts). In this case the scan time with Greenbone was also extremely long, despite the appliance being mostly idle. It may be that it could have been configured more aggressively to counteract this.

C.2 In-depth Comparison

The results of some hosts that were scanned by both appliances are compared below. The host with the most critical results is examined first.

Product	Critical	High	Medium	Low	Total
Holm	4	46	41	2	93
Greenbone	n/a	29	58	3	90

Note: Log / Info were omitted in the Report

While it looks like Holm has found many more High and Critical Vulnerabilities, the total amount found is nearly the same between the two products. The differences are purely the result of the different classifications of Critical / High / Medium and Low. Based on the CVSS v2 Score:

Vendor	Critical	High	Medium	Low	Info
Holm	10 - 8.1	8.0 - 5.1	5.0 - 2.1	2.0 - 0.1	0.0
Greenbone	n/a	10 - 7.0	6.9 - 4.0	3.9 - 0	n/a

Severity Classes

According to the Greenbone Manual: https://docs.greenbone.net/GSM-Manual/gos-4/en/gui_introduction.html#my-settings

Compared to Holm: https://support.holmsecurity.com/hc/en-us/articles/214187065-What-do-the-different-values-and-information-for-vulnerabilities-in-Vulnerability-Tests-mean-

#	Vulnerability (reported by Holm)	Severity	Detected by Greenbone?
1	NVT: PHP End Of Life Detection (Windows)	Critical (CVSS v2: 10)	YES
2	NVT: PHP Denial of Service And Unspecified Vulnerabilities - 01 - Jul16 (Windows)	10	YES
3	NVT: phpMyAdmin End of Life Detection (Windows)	Critical (CVSS v2: 10)	YES
4	NVT: PHP Multiple Vulnerabilities - Dec18 (Windows)	Critical (CVSS v2: 8.5)	YES
5	NVT: PHP Denial of Service Vulnerability Jul17 (Windows)	High (CVSS v2: 7.8)	YES
6	NVT: PHP 'libgd' Denial of Service Vulnerabitliy (Windows)	7.5	YES
7	NVT: PHP Multiple Vulnerabilities - 02 - Sep16 (Windows)	7.5	YES
8	NVT: PHP Remote Code Execution Vulnerability-01 Jun17 (Windows)	7.5	YES

#	Vulnerability (reported by Holm)	Severity	Detected by Greenbone?
9	NVT: PHP Multiple Vulnerabilities - Feb19 (Windows)	7.5	YES
10	NVT: PHP Multiple Vulnerabilities - 03 - Sep16 (Windows)	7.5	YES
11	NVT: PHP Multiple Denial of Service Vulnerabilities - 02 - Jan17 (Windows)	7.5	YES
12	NVT: PHP Multiple Vulnerabilities - 01 - Aug16 (Windows)	7.5	YES
13	NVT: PHP Stack Buffer Overflow Vulnerability Mar18 (Windows)	7.5	YES
14	NVT: PHP Multiple Vulnerabilities - 01 - Jul16 (Windows)	7.5	YES
15	NVT: PHP 'CVE-2019-13224' Use-After-Free Vulnerability (Windows)	7.5	YES
16	PHP 'CVE-2019-11043' FPM Remote Code Execution Vulnerabitliy (Version Check)	High (CVSS v2: 7.5)	NO
17	PHP 'PHP-FPM' Denial of Service Vulnerability (Windows)	High (CVSS v2: 6.8)	YES
18	NVT: Apache HTTP Server Multiple Vulnerabilities June17 (Windows)	7.5	YES
19	Apache HTTP Server Multiple Vulnerabilities Apr18 (Windows)	High (CVSS v2: 6.8)	YES
20	PHP Denial of Service And Unspecified Vulnerabilities - 02 - Jul16 (Windows)	High (CVSS v2: 6.8)	YES

#	Vulnerability (reported by Holm)	Severity	Detected by Greenbone?
21	PHP Heap Use-After-Free Vulnerability - Sep19 (Windows)	High (CVSS v2: 6.8)	YES
22	PHP Multiple Vulnerabilities - Sep19 (Windows)	High (CVSS v2: 6.8)	YES
23	PHP Multiple Vulnerabilities May18 (Windows)	High (CVSS v2: 6.8)	YES
24	NVT: PHP Multiple Vulnerabilities - 03 - Jul16 (Windows)	7.5	YES
25	NVT: phpMyAdmin 4.5.0 <= 4.8.4 SQL Injection Vulnerabilitiy - PMASA-2019-2 (Windows)	7.5	YES
26	NVT: phpMyAdmin < 4.9.2 Multiple Vulnerabilities - PMASA-2019-5 (Windows)	7.5	YES
27	phpMyAdmin < 4.9.4, 5.x < 5.0.1 SQL Injection Vulnerability - PMASA-2020-1 (Windows)	High (CVSS v2: 6.5)	YES
28	NVT: PHP Multiple Vulnerabilities - 01 - Apr16 (Windows)	7.5	YES
29	NVT: PHP Multiple Vulnerabilities - 04 - Aug16 (Windows)	7.5	YES
30	NVT: phpMyAdmin < 4.8.6 SQL Injection Vulnerability - PMASA-2019-3 (Windows)	7.5	YES
31	NVT: PHP Multiple Vulnerabilities - 02 - Aug16 (Windows)	7.5	YES
32	NVT: PHP 'var_unserializer' Denial of Service Vulnerability (Windows)	7.5	YES

#	Vulnerability (reported by Holm)	Severity	Detected by Greenbone?
33	NVT: PHP Multiple Vulnerabilities - 05 - Jul16 (Windows)	7.5	YES
34	NVT: PHP Multiple Vulnerabilities - 03 - Aug16 (Windows)	7.5	YES
35	Apache HTTP Server 'mod_auth_digest' Multiple Vulnerabilities (Windows)	High (CVSS v2: 6.4)	YES
36	PHP 'phar_parse_pharfile' Function Denial of Service Vulnerability - (Windows)	High (CVSS v2: 6.4)	YES
37	PHP < 7.2.26 Multiple Vulnerabilities - Dec19 (Windows)	High (CVSS v2: 6.4)	YES
38	PHP < 7.2.27, 7.3.x < 7.3.14, 7.4.x < 7.4.2 Multiple Vulnerabilities - Jan20 (Windows)	High 6.4	YES
39	PHP Denial of Service Vulnerability - 02 - Aug16 (Windows)	High 6.4	YES
40	PHP Out of Bounds Read Memory Corruption Vulnerability - 01 - Mar16	High 6.4	YES
41	Apache HTTP Server < 2.4.39 mod_auth_digest Access Control Bypass Vulnerability (Windows)	High 6	YES
42	phpMyAdmin < 4.9.5, 5.x < 5.0.2 Multiple SQL Injection Vulnerabilities - PMASA-2020-2, PMSA-2020-3, PMSA-2020-4 (Windows)	High 6	YES
43	Apache HTTP Server 2.4.0 < 2.4.42 Multiple Vulnerabilities (Windows)	High 5.8	YES (TWICE)
44	Apache HTTP Server Multiple Vulernabilities (Windows)	High 5.8	YES

#	Vulnerability (reported by Holm)	Severity	Detected by Greenbone?
45	HTTP Debugging Methods (TRACE/TRACK) Enabled	High 5.8	YES
46	PHP < 7.2.29 Multiple Vulnerabilities - Mar20 (Windows)	High 5.8	YES
47	phpMyAdmin <= 4.9.0.1 CSRF Vulnerability (Windows)	High 5.8	YES
48	phpMyAdmin Multiple Vulnerabilities -01 May16 (Windows)	High 5.8	YES
49	Apache HTTP Server Man-in-the-Middle attack Vulnerability - July16 (Windows)	High 5.1	YES
50	PHP Man-in-the-Middle Attack Vulnerability - Jul16 (Windows)	High 5.1	YES
51	Apache HTTP Server 'mod_auth_digest' DoS Vulnerability (Windows)	Medium 5	YES
52	Apache HTTP Server 'mod_http2' Denial of Service Vulnerability (Windows)	Medium 5	YES
53	Apache HTTP Server 'Whitespace Defects' Multiple Vulnerabilities	Medium 5	NO
54	Apache HTTP Server < 2.4.38 HTTP/2 DoS Vulnerability (Windows)	Medium 5	YES
55	Apache HTTP Server < 2.4.38 mod_session_cookie Vulnerability (Windows)	Medium 5	YES
56	Apache HTTP Server < 2.4.39 URL Normalization Vulnerabitlity (Windows)	Medium 5	YES

#	Vulnerability (reported by Holm)	Severity	Detected by Greenbone?
57	Apache HTTP Server Denial of Service Vulnerability -02 Apr18 (Windows)	Medium 5	YES
58	Apache HTTP Server OPTIONS Memory Leak Vulnerability (Optionsbleed)	Medium 5	NO
59	Enabled Directory Listing Detection	Medium 5	NO
60	PHP 'CVE-2018-19935' - 'imap_mail' Denial of Service Vulnerability (Windows)	Medium 5	YES
61	PHP 'stream_get_meta_data' Priviledge Escalation Vulnerability (Windows)	Medium 5	YES
62	PHP 'timelib_meridian' Heap Based Buffer Overflow Vulnerability (Windows)	Medium 5	YES
63	PHP 'URL checks' Security Bypass Vulnerability Jul17 (Windows)	Medium 5	YES
64	PHP 'WDDX Deserialization' Denial of Service Vulnerability - (Windows)	Medium 5	YES
65	PHP < 7.2.28 Multiple Vulnerabilities - Feb20 (Windows)	Medium 5	YES
66	PHP < 7.2.30, 7.3 < 7.3.17, 7.4 < 7.4.5 DoS Vulnerability - Apr20 (Windows)	Medium 5	YES
67	PHP < 7.2.31, 7.3 < 7.3.18, 7.4 < 7.4.6 Multiple DoS Vulnerabilities - May20 (Windows)	Medium 5	YES
68	PHP Multiple Denial of Service Vulnerabilities - 01 - Jan17 (Windows)	Medium 5	YES

#	Vulnerability (reported by Holm)	Severity	Detected by Greenbone?
69	PHP Multiple Head Buffer Overflow and Information Disclosure Vulnerabilities (Windows)	Medium 5	YES
70	PHP Multiple Vulnerabilities - Jul17 (Windows)	Medium 5	YES
71	phpMyAdmin Information Disclosure Vulnerability	Medium 5	YES
72	phpMyAdmin Multiple Vulnerabilties -01 Feb16	Medium 5	YES
73	phpMyAdmin Multiple Vulnerabilities -02 Feb16	Medium 5	YES
74	Unprotected Web App Installers (HTTP)	Medium 5	YES
75	Cleartext Transmission of Sensitive Information via HTTP	Medium 4.8	YES
76	Apache HTTP Server Denial of Service Vulnerabilty - Jul16	Medium 4.3	NO
77	Apache HTTP Server Denial of Service Vulnerability Apr18 (Windows)	Medium 4.3	YES
78	jQuery 1.0.3 < 3.5.0 XSS Vulnerability	Medium 4.3	NO
79	jQuery 1.2 < 3.5.0 XSS Vulnerability	Medium 4.3	NO
80	jQuery < 1.9.0 XSS Vulnerability	Medium 4.3	YES
81	jQuery < 1.9.0 XSS Vulnerability (different ID)	Medium 4.3	NO
82	jQuery < 3.0.0 XSS Vulnerability	Medium 4.3	NO
83	jQuery < 3.4.0 Object Extension Vulnerability	Medium 4.3	NO

#	Vulnerability (reported by Holm)	Severity	Detected by Greenbone?
84	PHP 'PHAR' Error Page Reflected XSS And DoS Vulnerabilities (Windows)	Medium 4.3	YES
85	phpMyAdmin 4.0 <= 4.8.4 Arbitrary File Read Vulnerability - PMASA-2019-1 (Windows)	Medium 4.3	YES
86	phpMyAdmin 4.x < 4.8.4 Multiple Vulnerabilities - PMASA-2018-6, PMASA-2018-8 (Windows)	Medium 4.3	YES
87	phpMyAdmin < 4.9.0 CSRF Vulnerability - PMASA-2019-4 (Windows)	Medium 4.3	YES
88	phpMyAdmin <= 4.8.2 XSS Vulnerability - PMASA-2018-5 (Windows)	Medium 4.3	YES
89	phpMyAdmin Cross-Site Scripting Vulnerability (PMASA-2018-3)-Windows	Medium 4.3	YES
90	phpMyAdmin Multiple XSS Vulnerabilities -02 May16 (Windows)	Medium 4.3	YES
91	phpMyAdmin Multiple XSS Vulnerabilities -01 May16 (Windows)	Medium 3.5	YES
92	PHP Security Bypass Vulnerability May18 (Windows)	Low 1.9	YES
93	TCP Timestamps	Low 1.9	YES (SEVERITY 2.6)

#	Detected by Greenbone	Severity	Detected by Holm?
1	NVT: OpenSSH Denial of Service And User Enumeration Vulnerabilities (Windows)	High (CVSS: 7.8)	NO

#	Detected by Greenbone	Severity	Detected by Holm?
2	NVT: OpenSSH X11 Forwarding Security Bypass Vulnerability (Windows)	High (CVSS: 7.5)	NO
3	NVT: OpenSSH Multiple Vulnerabilities Jan17 (Windows)	High (CVSS: 7.5)	NO
4	OpenSSH User Enumeration Vulnerability-Aug18 (Windows)	Medium 5.0	NO
5	OpenSSH 'auth2-gss.c' User Enumeration Vulnerability (Windows)	Medium 5.0	NO
6	OpenSSH 'sftp-server' Security Bypass Vulnerability (Windows)	Medium 5.0	NO

It is unusual that Holm did not report on any OpenSSH vulnerabilities at all for the same host. The Holm report had at least two of the same OpenSSH Vulnerabilities in the same report but for a another host (Medium 5.0, 'aut2-gss.c' User Enumeration Vulnerability (Linux) and User Enumeration Vulnerability-Aug18 (Linux)). This means the checks (at least for Linux?) are enabled and the ports are being scanned.

Appendix D Test Case – Holm Security vs. OpenVAS

D.1 Overview of the Results

A single machine was scanned using Holm (Network Scan Profile - Standard, Default Port List) and OpenVAS and a comparison run (Fast and Full Scan, ALL IANA TCP assigned Ports). While the scan time was much lower in OpenVAS, the vulnerabilities it detected are also only 1/4 of those discovered by Holm. Therefore it was decided to repeat the scan with the Full Fast Ultimate Scan Profile (not yet the Full Deep or Full Deep and Ultimate) but aside from increasing the scan time to 12 minutes, the results did not change.

Vendor	Scan time	Detected vulnerabilities					
		Highest Severity	Critical	High	Medium	Low	Total
Holm	25 min	10	5 (3)	41 (30)	62 (45)	1	173 (including info) / 109 (without) / 79 (removed duplicate entries)
OpenVAS (Full and Fast)	8 min	7.5	n/a	6 (4)	20 (18)	1	27 (23)

Since the Webserver runs on HTTP and HTTPS (Port 80 and 443), almost all Vulnerabilities concerning HTTP(s) are reported twice (once for each port). Obviously this nearly doubles the amount of reported vulnerabilities. Removing the duplicate entries resulted in a total number of 79 vulnerabilities

reported by Holm (not including the Info events) and 23 by OpenVAS. The values in brackets () show the number of unique vulnerabilities in each category.

D.2 In-Depth Comparison

#	Vulnerability (reported by Holm)	CVSS	OpenVAS detected?
1	PHP End Of Life Detection (Linux)	10	
2	phpMyAdmin End of Life Detection (Linux)	10	
3	ProFTPD < 1.3.7 Multiple Vulnerabilities	9	
4	LimeSurvey < 3.16.1 Relative Path Vulnerability	7.5	YES
5	LimeSurvey < 3.17.14 Multiple Vulnerabilities	7.5	YES
6	Moodle 2.x / 3.x Remote Code Execution Vulnerability - Mar'17 (Linux)	7.5	
7	Moodle < 3.5.7, 3.6.x < 3.6.5, 3.7.x < 3.7.1 Multiple Vulnerabilities	7.5	
8	Moodle CMS <= 3.1.12, 3.2.x, 3.3.x <= 3.3.6, 3.4.x <= 3.4.3, 3.5.0 Multiple Vulnerabilities (Linux)	7.5	
9	Moodle CMS <= 3.1.15 SSRF Vulnerability	7.5	
10	phpinfo() output accessible	7.5	YES
11	phpMyAdmin < 4.8.6 SQL Injection Vulnerability - PMASA-2019-3 (Linux)	7.5	

#	Vulnerability (reported by Holm)	CVSS	OpenVAS detected?
12	phpMyAdmin < 4.9.2 Multiple Vulnerabilities - PMASA-2019-5 (Linux)	7.5	
13	ProFTPD <= 1.3.6 'mod_copy' Vulnerability	7.5	
14	WordPress WP Google Maps Plugin < 7.11.18 SQL Injection Vulnerability	7.5	YES
15	LimeSurvey CSRF Vulnerability	6.8	YES
16	Moodle CMS < 3.6, 3.5.x < 3.5.3, 3.4.x < 3.4.6, High 3.3.x < 3.3.9 and < 3.1.15 CSRF Vulnerability (Linux)	6.8	
17	PHP 'PHP-FPM' Denial of Service Vulnerability (Linux)	6.8	
18	PHP Heap Use-After-Free Vulnerability - Sep19 (Linux)	6.8	
19	PHP Multiple Vulnerabilities - Sep19 (Linux)	6.8	
20	phpMyAdmin Multiple Vulnerabilities -01 June15	High 6.8	
21	LimeSurvey <= 3.14.3 Multiple Vulnerabilities	High 6.5	YES
22	Moodle 2.x / 3.x Multiple Vulnerabilities - May'17 (Linux)	High 6.5	
23	Moodle 3.x Multiple Vulnerabilities - May'18 (Linux)	High 6.5	

#	Vulnerability (reported by Holm)	CVSS	OpenVAS detected?
24	Moodle < 3.5.12, 3.6.x < 3.6.10, 3.7.x < 3.7.6, 3.8.x < 3.8.3 RCE Vulnerability	High 6.5	
25	Moodle CMS 3.5.x < 3.5.2, 3.4.x < 3.4.5, 3.2.x < 3.3.8 and < 3.1.14 RCE Vulnerability (Linux)	High 6.5	
26	phpMyAdmin < 4.9.4, 5.x < 5.0.1 SQL Injection Vulnerability - PMASA-2020-1 (Linux)	High 6.5	
27	LimeSurvey File Disclosure Vulnerability	High 6.4	YES
28	Mahara <18.10.0 Mishandled User Requests Vulnerability	High 6.4	YES
29	Moodle < 3.5.9, 3.6.x < 3.6.7, 3.7.x < 3.7.3 Multiple Vulnerabilities	6.4	
30	phpMyAdmin < 4.9.5, 5.x < 5.0.2 Multiple SQL High Injection Vulnerabilities - PMASA-2020-2, PMSA- 2020-3, PMSA-2020-4 (Linux)	6	
31	Moodle <= 3.1.17, 3.4.x <= 3.4.8, 3.5.x <= 3.5.5, 3.6.x <= 3.6.3 Multiple Vulnerabilities	5.8	
32	Moodle CMS <= 3.1.16, 3.4.x <= 3.4.7, 3.5.x <= High 3.5.4 and 3.6.x <= 3.6.2 Link Injection Vulnerability	5.8	
33	phpMyAdmin <= 4.9.0.1 CSRF Vulnerability (Linux)	5.8	

#	Vulnerability (reported by Holm)	CVSS	OpenVAS detected?
34	Enabled Directory Listing Detection	Medium 5	
35	LimeSurvey < 3.17.10 Multiple Vulnerabilities	5	YES
36	LimeSurvey < 4.1.12 Multiple Vulnerabilities	5	YES
37	Moodle 3.x Spam Vulnerability - Mar'18 (Linux)	5	
38	Moodle <= 3.3.6, 3.4.* <= 3.4.3, 3.5.0 Information Disclosure Vulnerability (Linux)	5	
39	Moodle CMS 3.6.x < 3.6.2, 3.5.x < 3.5.4, 3.4.x < 3.4.7 and < 3.1.15 Multiple Vulnerabilities	5	
40	PHP 'CVE-2017-7189' Improper Input Validation Vulnerability (Linux)	5	
41	phpMyAdmin 'libraries/select_lang.lib.php' Information-Disclosure Vulnerability March15	5	
42	phpMyAdmin Denial-of-Service Vulnerability -01 Dec14	5	
43	ProFTPD < 1.3.6 Multiple Vulnerabilities	5	
44	ProFTPD < 1.3.6b and 1.3.7rc < 1.3.7rc2 Unauthenticated Denial of Service Vulnerability	5	
45	ProFTPD < 1.3.6c CRL Vulnerability	5	

#	Vulnerability (reported by Holm)	CVSS	OpenVAS detected?
46	WordPress BuddyPress Plugin < 5.1.2 Information Disclosure Vulnerability	5	YES
47	Cleartext Transmission of Sensitive Information via HTTP	4.8	YES
48	FTP Unencrypted Cleartext Login	4.8	YES
49	jQuery 1.0.3 < 3.5.0 XSS Vulnerability	4.3	
50	jQuery 1.2 < 3.5.0 XSS Vulnerability	4.3	
51	jQuery < 1.9.0 XSS Vulnerability	4.3	YES
52	jQuery < 3.0.0 XSS Vulnerability	4.3	
53	jQuery < 3.4.0 Object Extensions Vulnerability	4.3	
54	LimeSurvey < 2.72.4 XSS Vulnerability	4.3	YES
55	LimeSurvey < 3.15.6 XSS Vulnerability	4.3	YES
56	LimeSurvey < 4.3.9 XSS Vulnerability	4.3	YES
57	LimeSurvey <= 3.14.7 Multiple Vulnerabilities	4.3	YES
58	LimeSurvey <= 3.17.7 Cross-Site Scripting (XSS) Vulnerability	4.3	YES
59	LimeSurvey <= 3.19.1 Cross-Site Scripting (XSS) Vulnerability	4.3	YES

#	Vulnerability (reported by Holm)	CVSS	OpenVAS detected?
60	Moodle 3.x Multiple Vulnerabilities - Sep'17 (Linux)	4.3	
61	Moodle 3.x Multiple XSS Vulnerabilities - Mar'17 (Linux)	4.3	
62	Moodle CMS 3.5.x < 3.5.2, 3.4.x < 3.4.5, and < 3.3.8 XSS Vulnerability (Linux)	4.3	
63	phpMyAdmin 4.0 <= 4.8.4 Arbitrary File Read Vulnerability - PMASA-2019-1 (Linux)	4.3	
64	phpMyAdmin 4.x < 4.8.4 Multiple Vulnerabilities - PMASA-2018-6, PMASA-2018-8 (Linux)	4.3	
65	phpMyAdmin < 4.9.0 CSRF Vulnerability - PMASA- 2019-4 (Linux)	4.3	
66	phpMyAdmin <= 4.8.2 XSS Vulnerability - PMASA- 2018-5 (Linux)	4.3	
67	phpMyAdmin Cross-Site Scripting Vulnerability (PMASA-2018-3)-Linux	4.3	
68	SSL/TLS: Report Weak Cipher Suites	4.3	YES
69	Mahara 17.10 < 17.10.8, 18.04 < 18.04.4, 18.10 Multiple Vulnerabilities	4	YES
70	Moodle 3.x Information Disclosure Vulnerability - Nov'17 (Linux)	4	

#	Vulnerability (reported by Holm)	CVSS	OpenVAS detected?
71	Moodle 3.x Multiple Vulnerabilities - Jul'17 (Linux)	4	
72	Moodle 3.x Privilege Escalation Vulnerability - Jan'18 (Linux)	4	
73	Moodle 3.x Server Side Request Forgery Vulnerability - Jan'18 (Linux)	4	
74	Moodle < 3.7.2 Information Disclosure Vulnerability	4	
75	PHP < 7.2.33, 7.3 < 7.3.21, 7.4 < 7.4.9 DoS Vulnerability - August20 (Linux)	4	
76	SSL/TLS: Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerability	4	
77	Moodle 3.x XSS Vulnerability - Jan'18 (Linux)	3.5	
78	ProFTPD 'AllowChrootSymlinks' Local Security Bypass Vulnerability	2.1	
79	TCP timestamps	Low 1.9	YES

Appendix E Scanner Comparison for Web App and Standard Scans

The table below shows a simplified comparison of how different services and implementations are scanned specifically for a web application scan. The ability to verify certain components varies, as do the results. For example, Nessus skips the results if a service could be backported whereas Outscan notes that it could be backported.

Web App Scan	Nessus	Outscan	Detectify
Performance web scan	✓	✓	✗
Found vulnerabilities	✓	✓	✓
Services	✓	✓	✓
Backported	✗	✓	✗
Updated script database	✓	✓	✓
Scan script for critical vulnerabilities	✓	✓	✓

Table E.1: Scanner comparison for web app scan

The table below shows a simplified comparison of how different services and implementations are represented in a standard scan. The performance of Greenbone was slower than the other tools for larger networks. The Greenbone appliance is limited to smaller networks but had a larger enterprise version been used this speed would have probably improved.

Standard Scan	Nessus	Outscan	Greenbone
Performance standard scan	✓	✓	✗

Standard Scan	Nessus	Outscan	Greenbone
Found vulnerabilities	✓	✓	✓
Open ports	✓	✓	✓
Services	✓	✓	✓
Backported	✗	✓	✗
Updated script database	✓	✓	✓
Scan script for critical vulnerabilities	✓	✓	✓

Table E.1: Scanner comparison for standard scan

Appendix F Test Case – Results of Vulnerable System Scan

Found vulnerability	Nessus	Greenbone	Outscan
X11 Server CVE-1999-0526	Critical	High	High risk
PHP Unsupported Version Detection	Critical	High	High risk
FTP Server Detection	None	High	High risk
Squid Proxy Detection	None	Low	High risk

Appendix G Speed Comparison between Greenbone, Outscan, Nessus and Detectify

G.1 Web App Scan – 1 site – standard settings

Scanner	Scan time
Detectify	2h 29min
Nessus	23 min
Outscan	29 min
Greenbone	Does not have a special web-app-mode

G.2 Standard Scan – 1 IP – standard settings

Scanner	Scan time
Detectify	Does not have a standard scan-mode
Nessus	22 min
Outscan	44 min
Greenbone	46 min

Scanner	Found CVEs
Nessus	CVE-2019-9511, CVE-2019-9513, CVE-2019-9516, CVE-2019-20372, CVE-2021-23017, CVE-2021-44224, CVE-2021-44790
Outscan	CVE-2011-3389 - (Backported software Nginx, Linux kernel, Apache HTTP server)
Greenbone	CVE-2011-3389 CVE-2015-0204

Scanner	Version of nginx (severity)
Nessus	nginx 1.14.2 (high)
Outscan	nginx 1.14.2 (backported, so don't classify it as a vuln)
Greenbone	nginx 1.14.2
Detectify	nginx 1.14.2

Scanner	Version of apache (severity)
Nessus	Apache 2.4.51 Multiple Vulnerabilities (Critical)
Outscan	Apache 2.4.51 Multiple Vulnerabilities (backported, so don't evaluate it as a vuln)
Greenbone	Apache 2.4.51 (no warning)
Detectify	Apache 2.4.51 (no warning)

Scanner	Latest update in database 22-03-10
Nessus	CVE-2022-21990 Family: Windows: Microsoft Bulletins Published: 22-03-08 Updated: 22-03-09

	Risk Factor: High
Outscan	CVE-2022-21990 Check created 22-03-08 CVSS score 10.0
Greenbone	CVE-2022-26505 Published 22-03-06

Scanner	CVE-2022-21969 Microsoft Exchange Server Remote Code Execution Vulnerability - MITRE
Nessus	Critical 220111 (Updated 220112)
Outscan	CVSS 7,7 220112
Greenbone	High 220111

Appendix H Commercial Scanner Requirements (GÉANT Open Tender Procedure)

MR 1. The implemented system is able to complete an inventory of vulnerable ports and services regardless of whether the system responds to ping or not. The System is able to use different detection modes such as SYN-scan and other flag options. (Similar to nmap -Pn, or just imply that all specified targets are interpreted as online)

MR 2. The system is able to fingerprint operating systems and versions of all installed software when they are exposed either as version numbering or behaviour. This data results both in reducing scan times by eliminating unnecessary checks and building an asset database/information pool of hosts with specific versions and operating systems. The system is able to accurately determine versions and operating systems and store that information linked to the asset.

MR 3. The system is able to verify known vulnerabilities or versions with updates from the CVE database continuously with network based tests in cases where this is possible based on published data (POC code or detailed technical description). The inventory is storing versions detected and has the ability to inform of new CVEs that match inventory.

MR 4. It is possible to verify known vulnerabilities (with a high attack value and current exploitability) within 48 hours for CVSS scores above 7 for a majority (80+%) of current vulnerabilities available on the internet primarily from database at https://cve.mitre.org/ or publicly disclosed from vendors' or researchers' own information channels.

MR 5. The system warns of other security-related information, for example gives warnings on certificates that expire soon. The system is also able to verify security impacting misconfigurations or bad practice default settings. This can also be done via an agent installation.

MR 6. The system can send reports in encrypted format either reachable via the web (for example via personal login via HTTPS) or sending through mail that only the recipient can open (for example via PGP/GPG or an out-of-band password to a zip-file).

MR 7. Reports are structured in different formats, as a minimum HTML, CSV and PDF. CSVs must be separated and field specific to enable automatic import in other tools.

MR 8. Reports must be able to be sent after the scan has been completed and retrieved manually afterwards.

MR 9. The data is protected and encrypted at rest.

MR 10. If the system has an interface, the web interface must use TLS.

MR 11. The API should be protected by authkey or similar.

MR 12. If license is counted on IP/assets and not scan capacity over time: Then IPv4/IPv6 assets must be able to combine as one host in reporting based on DNS-names. If this is not achievable a license model must support all connected hosts on an NREN's or institution's network.

MR 13. The service must let each user scan at least 16 systems (detected as live) in parallel. Initiated scans shall complete and not timeout.

MR 14. The system must be able to import asset lists and domain names via csv or line import.

MR 15. This importlist must be able to parse: Hostnames and correlate these to existing defined IP-addresses as name/IP for host. The asset list and database structure must be able to support at least 512k IP/organisation (these can be split into different scan profiles/jobs).

MR 16. The system must be able to support automated processes for both starting scan jobs and getting status and reports. This includes access to an API to request scans or queries of results. Exporting and starting of tickets either through mail or defined integrations should be successfully done with at least two support systems such as Jira or RT.

MR 17. The system must be able to send reports at a detailed technical level for each individual vulnerability or centric on a particular host.

MR 18. Summarised compilation where it shows the status of results with recommended solutions for issues. MR 19. Filtering on different levels such as Informational/Low/High/Critical.

MR 20. Filter searches by groups must be made for both specific host-sets and specific vulnerabilities.

MR 21. There must be a way to include local scanner nodes within institutions and have control over them from a central instance. Alternatively, functionality to use a tunnel/VPN/relay for equal functionality.

MR 22. The system must be accessible by web without need to install other components with common browsers such as Chrome, Safari and Firefox.

MR 23. The system must support eduGAIN SAML based authentication with support for multiple organisations' Identity providers. More information about this implementation can be found at https://wiki.geant.org/display/eduGAIN. The awarded supplier will integrate and demonstrate this functionality in a pilot installation at Swedish NREN SUNET or at GÉANT.

MR 24. A local scanner that scans IP-ranges that are appearing at other networks; The database (or interface sorting) must be able to provide information about which network and scanner provided the result.

MR 25. The system should be able to perform authenticated tests over at least SMB (user/pass) and SSH (pub/private keys) for verification of local settings and compliance.

MR 26. Stored data should be protected by the system.

References

[CVE]	https://www.cve.org/
[CVE_Details]	https://www.cvedetails.com/vulnerability-list/
[Detectify]	https://detectify.com/
[D. Heed-Interview]	https://connect.geant.org/2021/01/13/proactive-security-monitoring-can-help-reduce-vulnerability
[ExploitHub]	https://exploithub.com/
[GÉANT_GitLab_VA]	https://gitlab.geant.org/vulnerabilityassessment/vulnerabilityassessment [federated login required]
[Greenbone]	https://www.greenbone.net/en/vulnerability-management/
[Greenbone_GVM]	https://github.com/greenbone/gvmd
[Greenbone_Scanner]	https://www.greenbone.net/en/hardware-appliances/
[Holm_Security]	https://www.holmsecurity.com/
[MASSCAN]	https://github.com/robertdavidgraham/masscan
[Nessus_Prof]	https://www.tenable.com/products/nessus/nessus-professional
[Nmap]	https://nmap.org/
[Nmap_PSD&S]	https://nmap.org/book/performance-port-selection.html
[OpenVAS]	https://www.openvas.org/
[OpenVAS_GÉANT]	https://gitlab.geant.org/vulnerabilityassessment [federated login required]
[Outscan]	https://outpost24.com/products/vulnerability-assessment
[OWASP_Top10]	https://owasp.org/www-project-top-ten/
[RCE]	https://encyclopedia.kaspersky.com/glossary/remote-code-execution-rce/
[RFP_VMS]	https://portal.negometrix.com/index.cfm?action=sourcing:__requestform.design_invitation&document=182896D4-D648-64C6-BF1E6AC6B7148D3C&element_id=1190#
[Unicornscan]	https://www.kali.org/tools/unicornscan/
[Wiki_VAaaS]	https://wiki.geant.org/display/gn43wp8/3.2+Vulnerability+Assessment+as+a+Service [federated login required]
[Wiki_VSE]	https://wiki.geant.org/pages/viewpage.action?spaceKey=gn43wp8&title=Vulnerability+Scanner+Evaluation [federated login required]
[Wikipedia_DNS]	https://en.wikipedia.org/wiki/Domain_Name_System
[Wikipedia_GPL]	https://en.wikipedia.org/wiki/GNU_General_Public_License
[Wikipedia_IANA]	https://en.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority
[Wikipedia_ICMP]	https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
[Wikipedia_TTL]	https://en.wikipedia.org/wiki/Time_to_live
[Wikipedia_YubiKey]	https://en.wikipedia.org/wiki/YubiKey
[Zerodium]	https://zerodium.com/

Glossary

0day	A previously unknown vulnerability is initially called a Zero day or 0day vulnerability
API	Application Programming Interface
CIDR	Classless Inter-Domain Routing (a method for allocating IP addresses and for IP routing)
CMDB	Configuration Management Database
CSV	Comma-separated values file format
CVE	Common Vulnerabilities and Exposures is a list of publicly disclosed computer security flaws
CVSS	Common Vulnerability Scoring System provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity
DNS	Domain Name System is the hierarchical and decentralised naming system used to identify computers reachable through the Internet or other Internet Protocol (IP) networks [Wikipedia-DNS]
DOCX	Office Open XML document (file extension used in Microsoft Office)
github	Code hosting platform for version control and collaboration
GitLab	Open source end-to-end software development platform with built-in version control, issue tracking, code review, CI/CD, and more
GPL License	GNU General Public License (GNU GPL or simply GPL) is a series of widely used free software licences that guarantee end users the four freedoms to run, study, share, and modify the software [Wikipedia_GPL]
GVM	Greenbone Vulnerability Management is a network security scanner with associated tools such as a graphical user front-end
GUI	Graphical User Interface
HTML	HyperText Markup Language is the standard markup language for documents designed to be displayed in a web browser
HTTP	Hypertext Transfer Protocol is an application-layer protocol for transmitting hypermedia documents, such as HTML
HTTPS	Secure version of HTTP protocol
IANA	Internet Assigned Numbers Authority, a standards organisation that oversees global IP address allocation, autonomous system number allocation, root zone management in the Domain Name System (DNS), media types, and other Internet Protocol-related symbols and Internet numbers [Wikipedia_IANA]
ICMP	Internet Control Message Protocol is a supporting protocol in the Internet protocol suite. It is used by network devices, including routers, to send error messages and operational information indicating success or failure when communicating with another IP address [Wikipedia_ICMP]
IP	Internet Protocol
Jira	A suite of agile work management solutions that powers collaboration across all teams

JSON	JavaScript Object Notation is an open data interchange format that is both human- and machine-readable
LAN	Local Area Network
Log4J	Apache log4j is one of the most widely used Java logging libraries
MAC address	Unique physical address assigned to each network adapter in a computer, or mobile device
MASSCAN	Fast and Scalable IP Port Scanner
MoU	Memorandum of Understanding
Nmap	Network Mapper, a free and open source utility for network discovery and security auditing
NREN	National Research and Education Network
NSE	Nmap Scripting Engine
Open source	The term that refers to something people can modify and share because its design is publicly accessible
OpenVAS	Open Vulnerability Assessment Scanner
OS	Operating System
OWASP	Open Web Application Security Project, used for web application development
PDF	Portable Document Format (Adobe Acrobat)
PGP	Pretty Good Privacy, an encryption program
QoD	Quality of Detection
R&E	Research and Education
RCE	Remote Code Execution, one of the most dangerous types of computer vulnerabilities. It allows an attacker to remotely run malicious code within the target system on the local network or over the Internet. Physical access to the device is not required [RCE].
SAML	Security Assertion Markup Language. Enables multiple web applications to be accessed using one set of login credentials
SIG	Special Interest Group
SIG-ISM	Special Interest Group on Information Security Management
Slack	Messaging program designed specifically for the workplace
SMS	Short Message Service
SOC	Security Operations Centre
Splunk	Software for monitoring and searching through big data
SSH	Secure Shell, cryptographic network protocol for operating network services securely over an unsecured network
SSO	Single Sign-On, an authentication method that enables users to securely authenticate with multiple applications and websites by using just one set of credentials
TCP	Transmission Control Protocol
TCP ACK	ACK is the acknowledgment flag which is used to acknowledge the successful receipt of a packet
TCP SYN	SYN is a TCP packet sent to another computer requesting that a connection be established between them
TF	Task Force
TF-CSIRT	Task Force – Computer Security Incident Response Teams
TTL	Time To Live or hop limit is a mechanism that limits the lifespan or lifetime of data in a computer or network [Wikipedia_TTL]
UDP	User Datagram Protocol
Unicornscan	Free and open-source Automated Penetration Testing tool available on GitHub

VA	Vulnerability Assessment
VM	Virtual Machine, virtualisation/emulation of a computer system
VirtualBox	Oracle VM VirtualBox is cross-platform virtualisation software which allows users to extend their existing computer to run multiple operating systems
WP	Work Package
WP8	Work Package 8 Security
WP8 Task 3	WP8 Task 3 Products and Services
WP8 Task 3.1	WP8 Task 3 Products and Services: Security Operations Centre
WP8 Task 3.2	WP8 Task 3 Products and Services: Vulnerability Assessment as a Service
XLS	Microsoft Office Excel file extension
XML	eXtensible Markup Language
YubiKey	Hardware authentication device manufactured by Yubico to protect access to computers, networks, and online services that supports one-time passwords (OTP), public-key cryptography, and authentication, and the Universal 2nd Factor (U2F) and Fast Identity Online 2 (FIDO2) protocols [Wikipedia_YubiKey]