DOI: 10.15587/978-617-8360-16-0.CH3

Bohdan Malitskyi, Oleksandr Cherepov, Vasyl Rizak, Mykhailo Rizak

CHAPTER 3

CYBER POLYGON AS A TOOL FOR TRAINING CYBERSECURITY PROFESSIONALS

ABSTRACT

The continuous escalation of cyber threats and the evolution of attack methods on information systems necessitate the training of highly skilled cybersecurity professionals who can effectively respond to real-world threats. There is a need for training programs that provide students not only with theoretical knowledge but also with practical experience in countering cyberattacks. Cyber polygons serve as a critical tool in preparing professionals, enabling students to develop vulnerability assessment skills and implement defense strategies in an environment that simulates real-world attack and defense scenarios.

This study is based on the cyber polygon of the Department of Solid-State Electronics and Information Security, which includes a comprehensive suite of training scenarios covering various aspects of cybersecurity. Three key scenarios are outlined in this work. The first involves web application vulnerability scanning using Qualys, allowing students to learn risk assessment and develop recommendations for enhancing security. The second scenario utilizes Metasploitable 2 as a simulation platform for practicing network attack and defense techniques. The third scenario, developed in collaboration with UnderDefense, involves tasks related to GitLab and Active Directory, where students engage in ethical hacking within a corporate infrastructure.

Through the use of the cyber polygon, students gain practical skills in vulnerability detection, risk assessment, and the application of comprehensive protection methods. They also acquire experience in managing Active Directory infrastructure, using LDAP for remote access, analyzing GitLab security, and performing attacks in realistic network environments. Team competitions and work on various scenarios enable students to master both offensive and defensive techniques, including brute forcing, remote code execution (RCE), Server Message Block (SMB), and local privilege escalation (LPE), strengthening their preparedness for careers in cybersecurity.

The training scenarios developed on the basis of the department's cyber polygon provide students with the necessary experience to work in the field of cybersecurity, deepening their understanding of risks and protection methods. The skills acquired enhance their competitiveness in the job market, equipping them to address information system security challenges in contemporary environments. The cyber polygon not only builds professional competencies but also fosters teamwork and strategic thinking, which are critically important for a successful career in cybersecurity.

KEYWORDS

Cyber polygon, cybersecurity, vulnerabilities, Active Directory, GitLab, Qualys, Metasploitable 2, ethical hacking, risk assessment, team competitions, Kerberoasting, pentesting, web application security, RCE, SMB, privilege escalation, LLMNR, educational process, practical training.

In today's world, cybersecurity occupies a central position among the strategic directions in the development of information technologies. The rising number of cyber threats and the rapid sophistication of cybercriminal tactics underscore the demand for skilled professionals. Each year, there is an increase in attacks on corporate, government, and personal systems, highlighting the need for experts who are capable not only of protecting networks and infrastructures but also of anticipating potential threats. To meet these needs, students must acquire not only theoretical knowledge but also practical skills in identifying vulnerabilities and securing systems under realistic conditions.

The primary objective of this chapter is to describe the methodology of cybersecurity training facilitated by the cyber polygon of the Department of Solid-State Electronics and Information Security. This cyber polygon provides students with the opportunity to study modern security tools while immersing them in practical attack and defense scenarios, which strengthens their knowledge and enhances their analytical and technical skills.

The chapter presents the training methods utilized within the cyber polygon, including the use of specialized tools and the organization of team competitions, which aid students in mastering ethical hacking and cyber defense skills.

The Department of Solid-State Electronics and Information Security (hereinafter, SSEIS) at Uzhhorod National University actively incorporates the cyber polygon into the educational process to ensure the practical preparation of students. This cyber polygon, developed using the latest technologies and methodologies in cybersecurity, serves as the foundation for training students through the simulation of real-world scenarios. It allows students not only to study the theoretical foundations of information security but also to apply their knowledge in practice, providing better preparation for their future careers.

The chapter is organized into several key sections, each detailing a specific aspect of the training process on the cyber polygon. The first section provides a detailed description of the cyber polygon and the principles behind its operation. Subsequent sections focus on specific training scenarios: the methodology for vulnerability scanning using the Qualys system, which enables students to gain skills in analyzing the security of web applications; the process of setting up a local network and establishing conditions for team competitions involving attack and defense exercises, demonstrated through the use of Metasploitable 2. The final section explores a new scenario developed in collaboration with UnderDefense, where students work with GitLab and Active Directory, learning contemporary methods of attack and defense within corporate systems.

3.1 OVERVIEW OF THE CYBER POLYGON

The SSEIS cyber polygon is a multifunctional environment designed to simulate real-world cyber threat scenarios and train individuals in effective response techniques. The main objectives of the cyber polygon include: developing students' professional skills and advancing the qualifications of cybersecurity specialists; conducting scientific research and testing new information protection technologies; training and retraining government employees, municipal officials, and military veterans in cybersecurity; and promoting cybersecurity awareness and the profession of cybersecurity experts.

The cyber polygon is tailored for practical exercises, covering aspects of ethical hacking, network science, and infrastructure protection. Within this environment, various scenarios of differing complexity are implemented, bringing students closer to the actual challenges faced by cybersecurity professionals. These scenarios allow participants to refine their offensive and defensive skills, fostering critical and strategic thinking, quick decision-making, and adaptability in the face of cyber incidents.

All scenarios on the cyber polygon closely resemble real-world cases and encompass a full spectrum of tasks that cybersecurity professionals may encounter. Training tasks include both offensive techniques (such as port scanning and vulnerability exploitation) and defensive measures (such as firewall configuration, blocking suspicious IP addresses, and traffic monitoring). Each scenario is designed to build skills in risk assessment and the development of corresponding defense strategies.

The SSEIS cyber polygon consists of an extensive architecture that incorporates various devices and network equipment, providing both attack and defense teams with the necessary resources to simulate cyberattacks and defensive measures. The cyber polygon's architecture is segmented between devices for the defense team and devices for the attack team, enabling the setup of symmetrical engagements in a secure environment.

The cyber polygon includes several types of devices for the blue team (defense), each serving a specific function:

1. Workstations for each defense team member: these workstations allow each defense team member to carry out tasks such as traffic analysis and attack detection and blocking. They are configured to handle complex monitoring and data processing tools that require high computational power and efficiency. The technical specifications for the workstations are as follows:

– CPU: Quad Core Processor or higher, ensuring fast processing of large data volumes and supporting multitasking;

– RAM: minimum of 8 GB, essential for the stable operation of traffic analysis and monitoring applications;

– storage: 128 GB SSD or more, providing quick data read and write speeds during attack simulations;

– graphics adapter: any type of graphics adapter to support graphical interfaces of monitoring software.

2. Honeypots: honeypots serve to create deceptive targets for the attack team. Placed within the operational network, these devices mimic active systems without actual functionality,

which can confuse attackers and lead them to spend time investigating these decoys. This approach allows the defense team to focus on real threats while using honeypots to enhance analytical thinking.

The technical specifications for honeypots include:

– CPU: Dual Core Processor or better, as the workload on these devices is minimal;

– RAM: 2–4 GB, sufficient for supporting basic system functions;

– storage: 64 GB SSD, providing adequate storage for minimal functionality.

3. Server – core of the defense system: the server is the main asset to be protected, housing the target data. As the primary target for the attack team, the defense team's role is to ensure its security and operational stability. This server typically handles high traffic volumes, making it a potential target for DDoS attacks. To withstand such attacks, the server is equipped with high-level resources. The technical specifications for the server are as follows:

– CPU: Quad Core Processor or higher, to endure heavy loads;

– RAM: minimum of 16 GB, necessary for processing intensive traffic and maintaining stability under DDoS conditions;

– storage: 256 GB SSD, providing sufficient speed and storage capacity for safeguarding critical data.

On the cyber polygon, laptops serve as endpoints for the red team (attackers), allowing team members to simulate various types of attacks and attempt to gain access to the protected resources of the defense team. The red team's tasks involve network scanning, vulnerability discovery, executing attacks, and penetrating the target server. Laptops for Attack Team Members: Each member of the attack team is provided with a personal laptop, enabling them to perform different types of attacks, such as port scanning, identifying vulnerable services, and utilizing exploits to penetrate the network. These tasks require high-performance devices to ensure smooth operation of attack and scanning tools. The technical specifications for the laptops used by the attack team are as follows:

– CPU: Quad Core Processor or higher, allowing fast data processing and efficient operation of attack software;

– RAM: minimum of 8 GB, essential for stable performance of scanning tools like nmap, Metasploit, and others;

– storage: 128 GB SSD or more, ensuring quick access to saved data and scan results;

– graphics adapter: basic graphics card, sufficient to support the interface of scanning and attack simulation tools.

Role of virtual machines in Attack Team tasks: all actions by the attack team are performed on virtual machines within VirtualBox, ensuring the security of the cyber polygon's primary infrastructure and allowing quick system resets to the initial state. Each team member uses a virtual machine running Kali Linux, a specialized distribution for security testing. Kali Linux includes a suite of attack and analysis tools, such as nmap, Metasploit, and Wireshark, enabling the red team to conduct a full attack cycle.

Tools used by the Attack Team:

– nmap: for network scanning, identifying active endpoints, open ports, and services;

– metasploit: for executing exploits and testing server and network device vulnerabilities;

– Burp Suite, Hydra, SQLmap: additional tools for web application attacks, password brute-forcing, and database vulnerability analysis.

With this setup, the attack team can simulate full-scale cyber threats, allowing participants to develop practical ethical hacking skills and understand strategies for attacking network infrastructures.

Various types of networking equipment are used to integrate all devices on the cyber polygon into a single network and control access. The cyber polygon includes firewalls, switches, and routers, enabling the creation of a multi-layered network with high levels of security and flexibility:

1. Firewall: the Cisco ASA 5506 firewall is used on the cyber polygon to protect the defense team's network from external attacks and control access to it. The firewall is a primary security tool, as it limits the red team's access to the defense network, reducing the likelihood of unauthorized intrusion.

Key specifications of the Cisco ASA 5506:

– Data Transfer Rate: 750 Mbps, providing a stable connection during high-traffic attack simulations;

– Firewall Throughput: 250 Mbps, enabling the processing of a large number of network requests;

– IDS/IPS Throughput: 125 Mbps, allowing the use of intrusion detection and prevention tools effectively;

– Ethernet Ports: 9, supporting the connection of various defense team devices and protecting the network infrastructure;

– VPN Connections: Up to 50, enabling secure access channels between subnetworks;

– VLAN Connections: Up to 30, facilitating the creation of virtual networks for improved isolation of defense team devices.

The Cisco ASA 5506 allows for the configuration of firewall rules that can restrict access to critical network components and control traffic between subnetworks. Depending on the training scenario, firewall configuration can be performed either by instructors or students, providing hands-on learning opportunities.

2. Switches: switches are used to connect the endpoints of both the attack and defense teams into a unified network. Operating at the data link layer of the OSI model, switches facilitate efficient data transfer between devices within the same team. Switches also support network segmentation, providing flexibility in configuring the network environment for both teams.

3. Routers: routers enable connectivity between the subnets of the attack and defense teams, creating a unified network that allows for interaction between the two teams. This setup is essential for executing various attack types that require direct connections between networks. The cyber polygon utilizes Cisco and D-Link routers, supporting both static and dynamic routing modes, which offer adaptability during simulations of different threat types.

To ensure the cyber polygon's security, the networks of the attack and defense teams are fully isolated from the department's real network, mitigating risks related to infiltration into the corporate or external networks. Network isolation provides participants with complete freedom to interact with vulnerable systems and model attacks, ensuring that training outcomes have no impact on the department's core infrastructure.

All practical tasks on the cyber polygon are carried out in a virtualized environment using the VirtualBox platform, ensuring infrastructure security and flexibility in scenario configuration. The use of virtual machines allows for quick creation and restoration of training environments, granting participants full autonomy within a secure and isolated setting.

The use of VirtualBox in conjunction with Kali Linux enables the creation of reproducible environments for each student or group of students. Instructors can prepare virtual machines with a pre-installed suite of software and network configurations tailored to specific training scenarios. Upon completion of each session, the system can be reverted to its initial state using snapshots, eliminating the need for prolonged setup times.

Virtualization and the specialized software Kali Linux provide students with a unique opportunity to work with real cybersecurity tools in a secure environment that closely simulates the actual working conditions of a professional.

The network topology created for the SSEIS cyber polygon is shown in **Fig. 3.1**.

Participating in the training scenarios on the cyber polygon offers students invaluable hands-on experience with real cybersecurity tools and a deep understanding of network protection principles and vulnerability assessment. Working on the cyber polygon allows students to master the full cybersecurity cycle – from attack simulation to developing defense strategies. Below, we will review the primary scenarios used on the SSEIS cyber polygon.

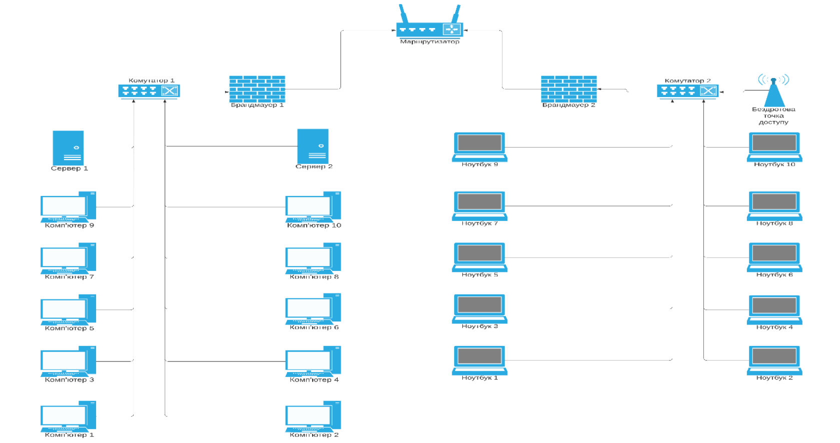


Fig. 3.1 Network topology of the SSEIS cyber polygon

3.2 METHODOLOGY FOR WEB APPLICATION VULNERABILITY SCANNING

Scanning web applications for vulnerabilities is a core responsibility for information security specialists. Modern web applications have become one of the most common entry points for attackers attempting to access confidential information, financial data, or other critical resources. With the rise of digital technologies, web applications are now integral to almost every fiel – from finance to healthcare. Ensuring their security is therefore a priority, as identifying and eliminating vulnerabilities allows specialists not only to protect information assets but also to prevent potential losses that may result from successful attacks.

The primary educational goal of the web application vulnerability scanning task is to develop students' skills in identifying and analyzing vulnerabilities that may compromise the security of information systems. This task is an integral part of the SSEIS cyber polygon training program, where students learn to use advanced tools for automated scanning and results analysis. They also gain insight into the factors that contribute to vulnerabilities, the potential consequences, and methods to mitigate risk.

The task is conducted using the Qualys platform, a leading tool for automated vulnerability scanning widely adopted in the cybersecurity industry. It provides solutions for detecting, assessing, and managing vulnerabilities across IT infrastructures, including networks, servers, workstations, and web applications. Through this platform, students become familiar with an industry-standard tool and learn to apply it under real-world conditions.

In the training process, using Qualys allows for automated security checks, enabling students to focus on analyzing identified issues. The Qualys interface allows students to easily visualize discovered vulnerabilities, assess their criticality and impact on the system, and offers detailed recommendations for remediation. These features are valuable for developing practical skills, as they enable students to understand the complete vulnerability management cycle – from identification to resolution.

This task specifically uses the Qualys Web Application Scanner (WAS), a cloud-based service for automated vulnerability scanning of web applications and APIs. Qualys WAS helps detect a wide range of vulnerabilities, including:

1. OWASP Top 10 Vulnerabilities (the most common threats to web applications), such as SQL injection, cross-site scripting (XSS), and insecure deserialization.

2. Sensitive Data Exposures (PII leaks), which could lead to violations of GDPR, HIPAA, and PCI DSS requirements.

3. Malicious Software – detection of malicious code embedded in web applications that could jeopardize user security and harm the company's reputation.

4. Insecure Configurations and Settings that could leave the web application vulnerable to attacks.

By identifying these threats, students gain a deeper understanding of the processes involved in securing web applications, develop skills in vulnerability analysis and remediation, and prepare themselves for professional roles in cybersecurity.

Before beginning the tasks in this scenario, users need to select a target web application to be tested using Qualys' automated tools. One option is the specialized website of the SSEIS department, designed to help students practice skills in both attacking and defending, making it an ideal platform for scanning. Additionally, students are encouraged to create their own web applications as scanning targets. This approach allows students to develop their own projects – ranging from a simple landing page to a more complex web application with authentication and a database. Once developed, the web application is hosted on the department's cloud servers, ensuring accessibility for subsequent scanning with the Qualys system.

To start using Qualys, each user must create an account with their institutional email. Qualys provides a free trial that allows for limited testing of applications within a set timeframe, enabling students to scan their own web applications without the need for paid services.

After account creation, users proceed to the Web Application Scanning section, where they carry out all primary steps of the task. The main stages of this process include:

1. Creating a web application record in the Qualys system.

2. Configuring scan parameters ("Option Profile").

3. Performing "Discovery" and "Vulnerability" scans to identify elements of the web application and assess existing vulnerabilities.

4. Generating a comprehensive report based on the vulnerability scan for further analysis.

5. Analyzing the report and formulating recommendations to improve the web application's security.

This approach enables students to experience the full cycle of working with a web application from a cybersecurity perspective: from developing their own product to scanning and analyzing its security using the professional tool Qualys.

Qualys provides a detailed approach to creating a web application record, allowing flexible customization of the scanning process. In the first step, users choose whether to create a new record "from scratch" or integrate elements from previously tested web applications. Within this scenario, students use the "clean" record option to familiarize themselves with all the customization features Qualys offers for creating a web application record.

Main Record Configuration Stages:

1. Asset Details: in this section, users enter the web application's name, URL, and attributes to facilitate easy identification of this record among others in the Qualys system.

2. Application Details: this section allows configuration of the web application's scan structure. Here, users can select specific URLs to scan, define the API type (or leave it unset if not applicable), which is convenient for simple web applications like landing pages. For complex applications, correctly configuring this section helps focus on critical components, optimizing scanning time.

3. Scan Settings: Scan settings can be configured later if needed, but here, users can: add an Option Profile for future scanning; select the type of scanner – external (recommended), internal (for networks), or a scanner appliance; assign a scanner to the record to prevent changes by other users; set a time limit for the scan; add robots.txt and sitemap.xml files; and configure header injections.

4. Crawl Settings: defines crawling options for testing scenarios. Integrating Selenium scripts for automated testing actions within the web application allows for personalized scanning scenarios.

5. Redundant Links: excludes unnecessary links from the scan process, which can reduce scan time and focus attention on the critical parts of the application.

6. Authentication: sets authentication parameters such as login credentials. For simple websites, this section may often be optional.

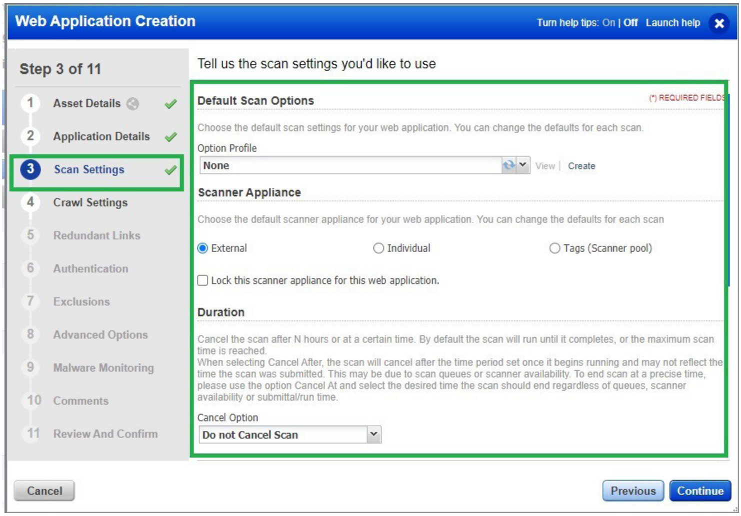
7. Exclusions: selects scan exclusions, allowing users to skip certain vulnerabilities and concentrate on critical areas.

8. Advanced options: additional parameters such as DNS Override and scan forms, which allow for more detailed scan configuration.

9. Malware monitoring: enables malware monitoring, a valuable tool for ongoing security monitoring, especially during the development stage and when testing new updates.

These settings allow students to configure the web application record in Qualys to focus the scanner on relevant aspects, avoid redundant actions, and ensure a high level of detail in the scanning process.

The next step is to create an Option Profile, which is a set of instructions defining the scanning configuration for the web application in Qualys (**Fig. 3.2**). The Option Profile includes all the necessary parameters to specify which aspects should be scanned and with what level of intensity. This setup allows the scan process to be tailored to the unique needs of the web application, improving the accuracy of vulnerability detection.



Tell us the scan settings you'd like to use

Default Scan Options (*) REQUIRED FIELDS

Choose the default scan settings for your web application. You can change the defaults for each scan.

Option Profile: None — View | Create

Scanner Appliance

Choose the default scanner appliance for your web application. You can change the defaults for each scan

● External ○ Individual ○ Tags (Scanner pool)

□ Lock this scanner appliance for this web application.

Duration

Cancel the scan after N hours or at a certain time. By default the scan will run until it completes, or the maximum scan time is reached.
When selecting Cancel After, the scan will cancel after the time period set once it begins running and may not reflect the time the scan was submitted. This may be due to scan queues or scanner availability. To end scan at a precise time, please use the option Cancel At and select the desired time the scan should end regardless of queues, scanner availability or submittal/run time.

Cancel Option: Do not Cancel Scan

⬡ **Fig. 3.2** Creating a web application record in the Qualys system

Main configuration elements of Option Profile cover the sections scan parameters and search criteria:

1. Scan Parameters – the main configuration block for setting the scan's intensity and specific characteristics (**Fig. 3.3**). It includes several sections:

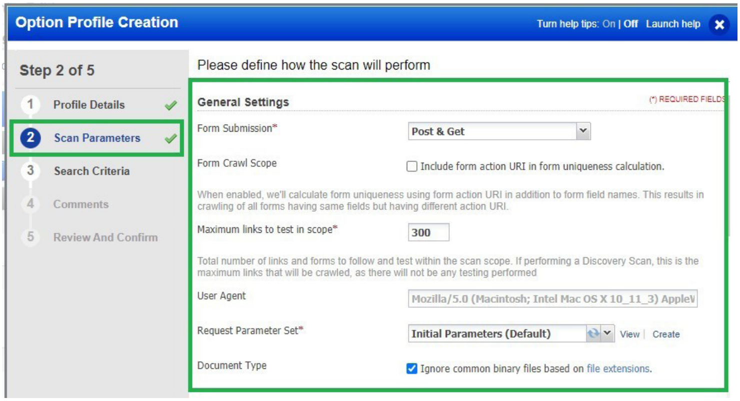
– general settings: in this section, the user selects the types of requests to be sent to the website (GET, POST, GET & POST, or None) and can set the number of unique forms to process, optimizing scan time. Additional settings include the maximum number of links to scan, agents for simulating user access, parameter templates, and the option to ignore common binary files;

– crawling options: allows the choice between full or "smart" page scanning. Full scanning covers all page components, whereas "smart" scanning focuses on core elements to optimize time, ignoring less critical parts. For this scenario, full scanning is recommended;

– behavior settings: sets a limit on the number of errors after which the scan will automatically stop. For instance, timeouts can be configured for working with slower sites, as well as handling unexpected errors;

– performance settings: enables selection of detailed or simplified scanning levels and customization of intensity – from minimal to high – based on the precision and duration requirements of the scan;

– bruteforce settings: here, the user specifies whether a bruteforce attack should be conducted on the web application.



Option Profile Creation

Turn help tips: On | Off Launch help

Step 2 of 5

1 Profile Details ✓

2 Scan Parameters ✓

3 Search Criteria

4 Comments

5 Review And Confirm

Please define how the scan will perform

General Settings (*) REQUIRED FIELDS

Form Submission* Post & Get

Form Crawl Scope □ Include form action URI in form uniqueness calculation.

When enabled, we'll calculate form uniqueness using form action URI in addition to form field names. This results in crawling of all forms having same fields but having different action URI.

Maximum links to test in scope* 300

Total number of links and forms to follow and test within the scan scope. If performing a Discovery Scan, this is the maximum links that will be crawled, as there will not be any testing performed

User Agent Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_3) AppleV

Request Parameter Set* Initial Parameters (Default) View | Create

Document Type ☑ Ignore common binary files based on file extensions.

Fig. 3.3 Creating an option profile

2. Search Criteria – a section for selecting additional criteria to enhance scan quality:

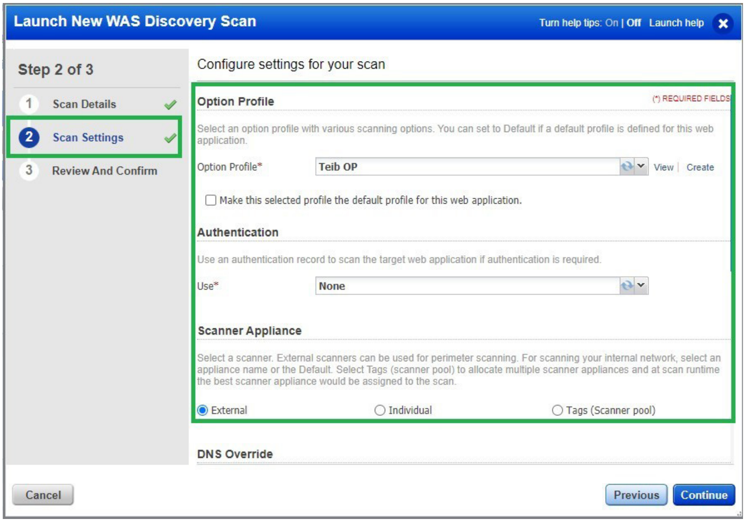
– detection scope: allows selection of the detection type and specifies if additional XSS payloads will be used to check for cross-site scripting vulnerabilities;

– sensitive content: specifies the types of sensitive information to which the scanner should pay attention, such as credit card numbers or social security data;

– keywords URL search: enables the specification of keywords for URL searches, which can narrow the scan focus and speed up the identification of critical elements.

This detailed Option Profile configuration allows students to tailor the scan to the requirements of a specific web application and focus on identifying the most critical vulnerabilities essential for ensuring information security.

The next stage is performing Discovery and Vulnerability scans based on the created Option Profile. The Discovery scan is aimed at gathering information about the web application, which can be useful for identifying vulnerabilities (**Fig. 3.4**). This preliminary scan collects data on the website's structure, configurations, and characteristics, enhancing the efficiency of the main Vulnerability scan. The Vulnerability scan, in turn, conducts a thorough examination of the web application to uncover potential security risks (**Fig. 3.5**).



Launch New WAS Discovery Scan

Configure settings for your scan

Option Profile

(*) REQUIRED FIELDS

Select an option profile with various scanning options. You can set to Default if a default profile is defined for this web application.

Option Profile* Teib OP

□ Make this selected profile the default profile for this web application.

Authentication

Use an authentication record to scan the target web application if authentication is required.

Use* None

Scanner Appliance

Select a scanner. External scanners can be used for perimeter scanning. For scanning your internal network, select an appliance name or the Default. Select Tags (scanner pool) to allocate multiple scanner appliances and at scan runtime the best scanner appliance would be assigned to the scan.

● External ○ Individual ○ Tags (Scanner pool)

DNS Override

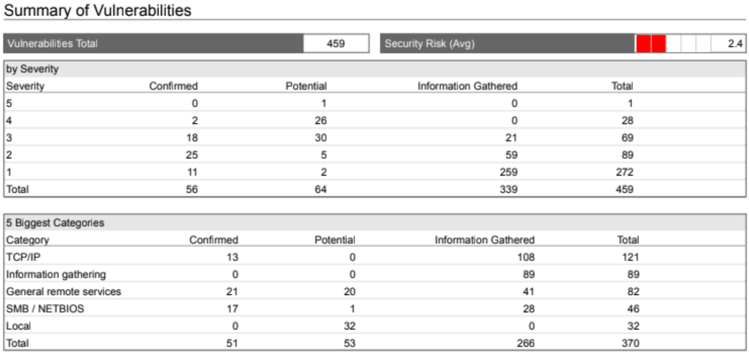
Fig. 3.4 Discovery scan settings

Qualys provides detailed scan customization options to enhance efficiency, allowing a focus on specific components of a web application or types of vulnerabilities. These settings include:

1. Scan details: this section allows users to define the main scan details, including its name, criteria for selecting web applications (by name or tags), and the specific web application to scan from the list of available applications.

2. Scan settings: this part includes all critical parameters for the upcoming scan, such as: selecting a previously created Option Profile; setting authentication parameters (if needed); choosing the type of scanner (external, internal, or a scanner appliance); activating DNS Override, setting a time limit for the scan or leaving the option "Do not stop"; and deciding whether to send email notifications upon scan completion.

After completing the Discovery scan, a full Vulnerability scan should be run with the same settings as for the Discovery. The results can be viewed in the Qualys web interface or downloaded in a convenient format.



Summary of Vulnerabilities

Vulnerabilities Total	459	Security Risk (Avg)	🟥🟥 2.4

by Severity

Severity	Confirmed	Potential	Information Gathered	Total
5	0	1	0	1
4	2	26	0	28
3	18	30	21	69
2	25	5	59	89
1	11	2	259	272
Total	56	64	339	459

5 Biggest Categories

Category	Confirmed	Potential	Information Gathered	Total
TCP/IP	13	0	108	121
Information gathering	0	0	89	89
General remote services	21	20	41	82
SMB / NETBIOS	17	1	28	46
Local	0	32	0	32
Total	51	53	266	370

Fig. 3.5 Overview of vulnerabilities found in the report

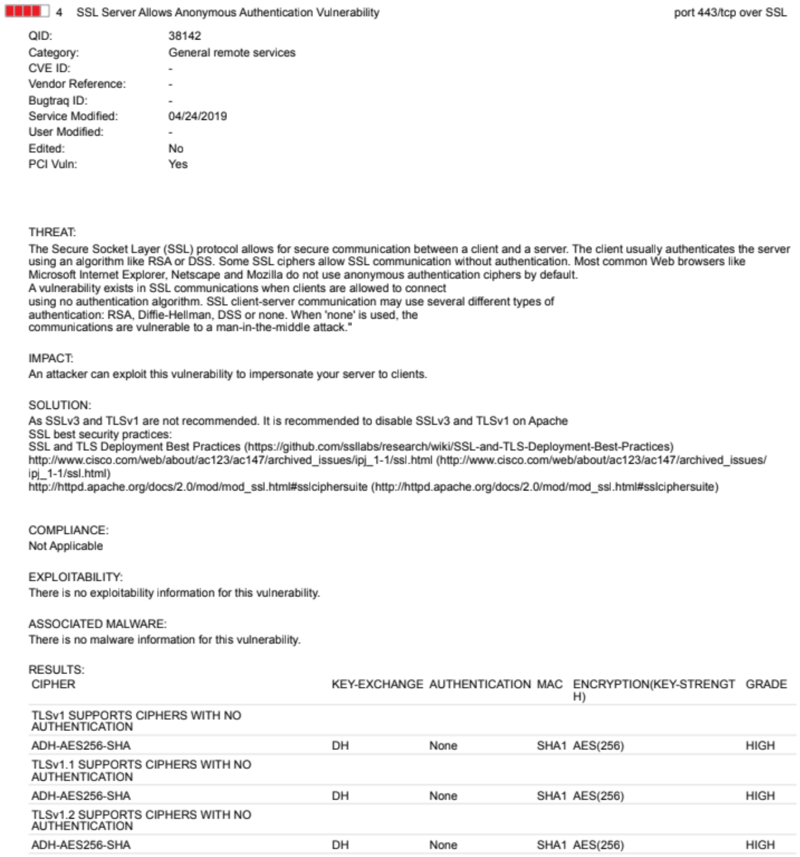
The analysis of results is a key stage in this scenario since the scanning process itself is automated. The ability to interpret a detailed report, highlight main issues, and propose solutions is an essential skill for cybersecurity professionals. This enables them to present scan results to clients in a clear format, pointing out necessary fixes and explaining methods to address vulnerabilities.

The report provided by Qualys is technically comprehensive and includes the following elements (**Fig. 3.6**):

1. The total number of identified vulnerabilities.
2. A breakdown of vulnerabilities by severity.
3. Information on operating systems and active services.
4. Detailed descriptions of each vulnerability, along with remediation recommendations.

Based on this technical report, students are tasked with creating a concise summary highlighting key vulnerabilities and proposing remediation measures. This scenario not only familiarizes them with the automated scanning process but also develops their skills in prioritizing risks, linking vulnerabilities to potential business impacts, and distilling essential information.

The exercise teaches students to present core insights in a way that is accessible to clients, even those without technical expertise in cybersecurity. This approach helps clients assess the security level of their web application and identify necessary actions to enhance its protection.



▮▮▮▮▯ 4 SSL Server Allows Anonymous Authentication Vulnerability port 443/tcp over SSL

QID:	38142
Category:	General remote services
CVE ID:	-
Vendor Reference:	-
Bugtraq ID:	-
Service Modified:	04/24/2019
User Modified:	-
Edited:	No
PCI Vuln:	Yes

THREAT:

The Secure Socket Layer (SSL) protocol allows for secure communication between a client and a server. The client usually authenticates the server using an algorithm like RSA or DSS. Some SSL ciphers allow SSL communication without authentication. Most common Web browsers like Microsoft Internet Explorer, Netscape and Mozilla do not use anonymous authentication ciphers by default.
A vulnerability exists in SSL communications when clients are allowed to connect
using no authentication algorithm. SSL client-server communication may use several different types of
authentication: RSA, Diffie-Hellman, DSS or none. When 'none' is used, the
communications are vulnerable to a man-in-the-middle attack."

IMPACT:

An attacker can exploit this vulnerability to impersonate your server to clients.

SOLUTION:

As SSLv3 and TLSv1 are not recommended. It is recommended to disable SSLv3 and TLSv1 on Apache
SSL best security practices:
SSL and TLS Deployment Best Practices (https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices)
http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_1-1/ssl.html (http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_1-1/ssl.html)
http://httpd.apache.org/docs/2.0/mod/mod_ssl.html#sslciphersuite (http://httpd.apache.org/docs/2.0/mod/mod_ssl.html#sslciphersuite)

COMPLIANCE:

Not Applicable

EXPLOITABILITY:

There is no exploitability information for this vulnerability.

ASSOCIATED MALWARE:

There is no malware information for this vulnerability.

RESULTS:

CIPHER	KEY-EXCHANGE	AUTHENTICATION	MAC	ENCRYPTION(KEY-STRENGTH)	GRADE
TLSv1 SUPPORTS CIPHERS WITH NO AUTHENTICATION					
ADH-AES256-SHA	DH	None	SHA1	AES(256)	HIGH
TLSv1.1 SUPPORTS CIPHERS WITH NO AUTHENTICATION					
ADH-AES256-SHA	DH	None	SHA1	AES(256)	HIGH
TLSv1.2 SUPPORTS CIPHERS WITH NO AUTHENTICATION					
ADH-AES256-SHA	DH	None	SHA1	AES(256)	HIGH

Fig. 3.6 Example report of a specific vulnerability

3.3 RED AND BLUE TEAM COMPETITIONS ON THE CYBER POLYGON

Red and blue team competitions are a crucial component of cybersecurity training, as they simulate real cyber conflicts. In this training format, the red team acts as attackers, emulating the actions of malicious actors, while the blue team is responsible for defense, responding to attacks, and ensuring system stability. This approach allows students to not only develop both offensive and defensive skills but also gain hands-on experience managing cyber incidents, which is invaluable for their professional growth.

The educational goal of this format is for students to learn how to analyze attack tactics and methods, devise their own defense strategies, and respond to threats swiftly. This fosters strategic thinking, enabling students to anticipate potential adversary scenarios and prepare accordingly. Mastering both attack and defense skills provides students with a comprehensive understanding of cybersecurity, helping them assess system weaknesses from both perspectives.

Beyond technical skills, red and blue team competitions foster "soft" skills such as teamwork, effective communication, decision-making under pressure, and adaptability to unpredictable situations. Students learn to coordinate their actions to achieve a common objective, which is a vital aspect of working in cybersecurity.

The red and blue team competition scenario on the cyber polygon is one of the first developed and involves a direct contest between the attacking (red) and defending (blue) teams. The red team's task is to gain access to the blue team's network infrastructure using any available tools, while the blue team learns to analyze traffic, block malicious network actions, and ensure the continuous operation of its systems.

This scenario's environment encompasses all major features of the cyber polygon. The red and blue teams are stationed in separate rooms without direct contact with each other. Each team has access to a specified set of hardware and software. The red team is equipped with ten laptops and a switch connecting all devices into a local network. Their software includes tools like nmap and Metasploit. The blue team has ten computers, two servers, a switch for network connectivity, and a router that links them to the red team's network. Additionally, the defense team has access to other cyber polygon resources, such as firewalls, servers, and intrusion detection/prevention systems (IPS/IDS). Although these additional tools are not required in the basic scenario, students can configure them to increase security and complicate the task for the opposing team.

All operations within the cyber polygon for this scenario are conducted on virtual machines. This ensures isolation from the cyber polygon's core devices, preventing potential vulnerability issues and facilitating the deployment and reset of systems to their initial state after the competition ends. Virtual machines are deployed using VirtualBox, a popular virtualization software known for its user-friendliness, relative simplicity, and support for major operating systems. Kali Linux is the operating system used for the virtual machines.

Kali Linux is a Debian-based Linux distribution specialized in security testing. For ease of use, Kali Linux includes a "Top 10 Security Tools" category, featuring widely-used security tools such as aircrack-ng, burp-suite, hydra, john, maltego, metasploit, nmap, sqlmap, wireshark, and zaproxy. This setup provides quick access to essential tools for various cybersecurity tasks, allowing the red team to efficiently simulate real-world attack scenarios.

Kali Linux also provides a comprehensive suite of tools for specialized tasks, including:

1. Reverse Engineering: Tools for debugging programs or analyzing executable files.

2. Stress Testing: Tools for endurance testing of networks, web environments, and VOIP systems.

3. Hardware Hacking: Utilities for working with Android and Arduino devices.

4. Forensics: Digital forensics tools such as Volatility, Autopsy, and Guymager, used for disk imaging, memory analysis, and file examination.

The main target of the competition is a server computer, which serves as the core objective in the scenario. The red team's goal is to gain access to this server, while the blue team must protect it, prevent unauthorized access, and block any attacks. The server runs Metasploitable 2 – a Linux distribution intentionally designed to be vulnerable for testing and demonstration of common security weaknesses.

Metasploitable 2 is designed to work with security tools and to provide a practical learning environment for common vulnerabilities (**Fig. 3.7**). This virtual machine is supported on platforms such as VirtualBox, VMware, and other popular virtualization environments. By default, the network interfaces of Metasploitable 2 are configured to NAT and Host-only, restricting access to external networks. While it is technically possible to install Metasploitable 2 as a primary or auxiliary operating system, this is strongly discouraged due to the numerous intentionally embedded vulnerabilities. The optimal approach is to deploy Metasploitable 2 as a virtual machine in VirtualBox, connected to the primary server of the blue team.

All systems in this scenario are designed to function without Internet access, meaning the cyber polygon environment is fully isolated from the department's main network. This approach allows students to work in a secure setting without concerns over potential consequences from interacting with real networks. The isolation ensures both a safe and convenient work environment as well as the security of the entire infrastructure.

The attack team follows several crucial steps to gain access to the target server and analyze its vulnerabilities for potential intrusion:

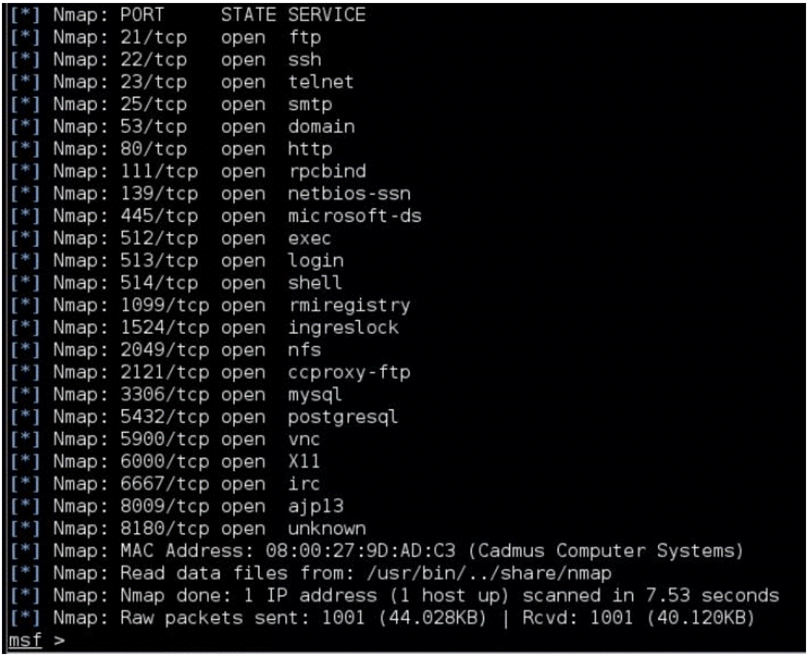
1. Scanning the Blue Team's Network: the first task for the red team is to scan the defense team's network to identify the endpoints, operating systems installed on them, open ports, and services running on each port. Armed with this information, the attack team can determine potential targets and entry points. This task is performed using the nmap utility included in Kali Linux.

Nmap (Network Mapper) is a popular and versatile network scanning tool used by cybersecurity professionals to quickly identify active devices, open ports, services, operating systems, and vulnerabilities within a network infrastructure. With Nmap Scripting Engine (NSE) capabilities, it enables not only basic scans but also specialized attacks. Using NSE, the attack team can discover vulnerabilities, exploit services, read service banners, and perform basic security audits.

Through these steps, the red team can systematically uncover weaknesses, analyze the defense team's network, and simulate potential attack paths, enhancing their understanding of penetration testing and cybersecurity tactics in a controlled, isolated environment.

2. Vulnerability identification in discovered services. After identifying the server running Metasploitable 2 and gathering information about open ports, service names, and versions, the attack team proceeds to search for vulnerabilities in one or more of the discovered services. Each team member typically focuses on a different service, with each task directed at exploring specific vulnerabilities within that service. This approach allows the team to leverage the full potential of Metasploitable 2 and gain comprehensive hands-on experience.

By dividing tasks among members, the red team can systematically investigate a wide range of services, analyzing each for known vulnerabilities that could be exploited. This targeted exploration provides a holistic view of vulnerability management and penetration testing, enhancing the team's ability to identify, assess, and exploit potential weaknesses in a real-world simulation environment.



```
[*] Nmap: PORT     STATE SERVICE
[*] Nmap: 21/tcp   open  ftp
[*] Nmap: 22/tcp   open  ssh
[*] Nmap: 23/tcp   open  telnet
[*] Nmap: 25/tcp   open  smtp
[*] Nmap: 53/tcp   open  domain
[*] Nmap: 80/tcp   open  http
[*] Nmap: 111/tcp  open  rpcbind
[*] Nmap: 139/tcp  open  netbios-ssn
[*] Nmap: 445/tcp  open  microsoft-ds
[*] Nmap: 512/tcp  open  exec
[*] Nmap: 513/tcp  open  login
[*] Nmap: 514/tcp  open  shell
[*] Nmap: 1099/tcp open  rmiregistry
[*] Nmap: 1524/tcp open  ingreslock
[*] Nmap: 2049/tcp open  nfs
[*] Nmap: 2121/tcp open  ccproxy-ftp
[*] Nmap: 3306/tcp open  mysql
[*] Nmap: 5432/tcp open  postgresql
[*] Nmap: 5900/tcp open  vnc
[*] Nmap: 6000/tcp open  X11
[*] Nmap: 6667/tcp open  irc
[*] Nmap: 8009/tcp open  ajp13
[*] Nmap: 8180/tcp open  unknown
[*] Nmap: MAC Address: 08:00:27:9D:AD:C3 (Cadmus Computer Systems)
[*] Nmap: Read data files from: /usr/bin/../share/nmap
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 7.53 seconds
[*] Nmap: Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.120KB)
msf >
```

Fig. 3.7 Example of nmap scan results for Metasploitable 2

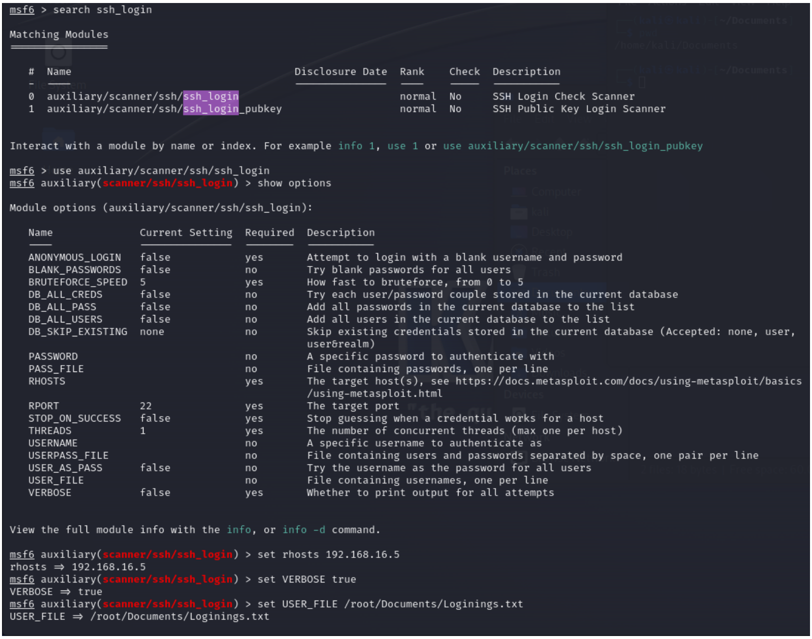
Metasploitable 2 contains an extensive list of vulnerable services across various ports. For each service, detailed information is available, helping the attack team understand which vulnerabilities can be exploited to gain access to the server. This information serves as a critical guide for targeting specific services and planning effective exploitation strategies (**Table 3.1**).

Table 3.1 Information on Entry Points in Metasploitable 2

Port No.	Service	Version	Exploite
21	FTP (vsftpd)	2.3.4	vsftpd 2.3.4 Backdoor
22	SSH	OpenSSH 4.7p1 Debian	Debian OpenSSL Predictable RNG
23	Telnet	Linux telnetd	No direct exploit
25	SMTP (Postfix)	Postfix 2.9.6	No direct exploit
53	DNS (Bind)	9.4.2	BIND TSIG Remote DoS
80	HTTP (Apache)	2.2.8	Apache Tomcat Manager Exploit
111	RPCbind	2.0	RPC DCOM Remote Overflow
139	NetBIOS/SMB	Samba 3.0.20	Samba Trans2open Overflow
445	SMB	Samba 3.0.20	Samba Trans2open Overflow
512	exec	BSD rexec	Remote Command Execution
512	login	Linux login service	No direct exploit
513	rlogin	BSD rlogin	Remote Command Execution
514	shell	BSD rsh	Remote Command Execution
1099	RMI Registry	Java RMI	Java RMI Server Insecure
1524	Ingreslock	Ingres Database	Ingreslock Backdoor
2049	NFS	Network File System	No direct exploit
2121	FTP (ProFTPD)	1.3.1	ProFTPD 1.3.1 Mod_copy Command Execution
3306	MySQL	5.0.51a	MySQL Remote Root Exploit
5432	PostgreSQL	8.3.0	PostgreSQL Pwnage
5900	VNC	VNC	VNC Authentication Bypass
6000	X11	X11 Server	Open X11 Server Exploitation
6667	IRC (UnrealIRCd)	UnrealIRCd 3.2.8.1	UnrealIRCd Backdoor Command Execution
8180	HTTP (Tomcat)	Apache Tomcat 5.5	Tomcat Manager Application Exploit

3. Exploitation and Persistence of Access. Using the gathered information on vulnerabilities, the attack team deploys appropriate exploits to gain access to Metasploitable 2 (**Fig. 3.8**). This task typically involves creating a “trace” – a file that logs details such as the port number, service name, version, the exploit used, and the timestamp of the intrusion. If access is gained through a database (such as MySQL or PostgreSQL), the task might include adding an entry to a table that has been pre-created by instructors.

At this stage, the attack team achieves its objective by securing access to the server and leaving a trace in the form of a file or database entry. This trace serves as evidence of task completion, marking the successful execution of the attack scenario.



```
msf6 > search ssh_login

Matching Modules
================

   #  Name                                    Disclosure Date  Rank    Check  Description
   -  ----                                    ---------------  ----    -----  -----------
   0  auxiliary/scanner/ssh/ssh_login                          normal  No     SSH Login Check Scanner
   1  auxiliary/scanner/ssh/ssh_login_pubkey                   normal  No     SSH Public Key Login Scanner


Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/ssh/ssh_login_pubkey

msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user,
                                                user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics
                                                /using-metasploit.html
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.16.5
rhosts ⇒ 192.168.16.5
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE ⇒ true
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /root/Documents/Loginings.txt
USER_FILE ⇒ /root/Documents/Loginings.txt
```

Fig. 3.8 Example of SSH Vulnerability Exploitation Using the Metasploit Framework

Defense Team tasks divided by knowledge level:

1. Monitoring and documenting Attack Team actions: for entry-level blue team members, the primary task is to monitor and document the actions of the attack team in detail. This approach is crucial for cybersecurity professionals, as a comprehensive cyber incident report, presented in clear language, enables management to understand the issue's nature and decide on preventive measures for the future. Effective report writing requires the ability to include all essential information about the threat source, attack methods, and potential impacts.

2. Active countermeasures against the Attack Team: for those with sufficient knowledge, the blue team members can directly counter the actions of the attack team. This includes actively monitoring the attackers' actions, analyzing server vulnerabilities, quickly patching these weaknesses,

and blocking unauthorized access to the network. In case an attack does occur, the defense team can block access to their infrastructure using the attackers' IP or MAC addresses.

The defense team uses Wireshark, a tool for in-depth packet analysis, for active network monitoring. With Wireshark, the blue team can: detect anomalies in network traffic that may indicate unauthorized access attempts (**Fig. 3.9**). Capture packets that could contain exploits or signs of port scanning. Identify the attackers' IP and MAC addresses, allowing for rapid localization of the attack source.

Tools such as Suricata and Snort offer automated threat detection by analyzing traffic in real-time, making them valuable additions for advanced network monitoring and strengthening the defense team's capabilities against complex attacks.

For vulnerability analysis, the defense team can use nmap to gather information on open ports and services, as well as to identify outdated or vulnerable components. nmap's insights help the team better understand network topology and find weaknesses in the configuration.

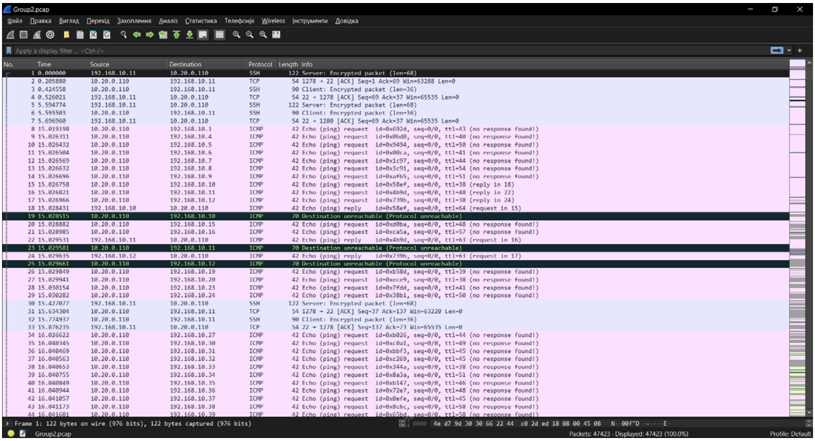
Upon detecting unauthorized access attempts, the defense team can implement several blocking methods:

1. Blocking the attackers' IP or MAC addresses at the firewall level to restrict network access.

2. Setting up traffic filtering to limit access from suspicious IP addresses or network segments used by the attack team.

3. Using Network Access Control (NAC) to restrict network access to authorized MAC addresses or according to specific access policies.

These measures help isolate the infrastructure from potential threats and maintain network control, even in case of intrusion.



```
No.   Time       Source          Destination     Protocol Length Info
    1 0.000000   192.168.10.11   10.20.0.110     SSH      122 Server: Encrypted packet (len=68)
    2 0.205880   10.20.0.110     192.168.10.11   TCP       54 1278 → 22 [ACK] Seq=1 Ack=69 Win=63288 Len=0
    3 0.424558   10.20.0.110     192.168.10.11   SSH       90 Client: Encrypted packet (len=36)
    4 0.526021   192.168.10.11   10.20.0.110     TCP       54 22 → 1278 [ACK] Seq=69 Ack=37 Win=65535 Len=0
    5 5.594774   192.168.10.11   10.20.0.110     SSH      122 Server: Encrypted packet (len=68)
    6 5.595503   10.20.0.110     192.168.10.11   SSH       90 Client: Encrypted packet (len=36)
    7 5.696960   192.168.10.11   10.20.0.110     TCP       54 22 → 1280 [ACK] Seq=69 Ack=37 Win=65535 Len=0
    8 15.019198  10.20.0.110     192.168.10.1    ICMP      42 Echo (ping) request  id=0x692d, seq=0/0, ttl=43 (no response found!)
    9 15.026311  10.20.0.110     192.168.10.4    ICMP      42 Echo (ping) request  id=0x0bd0, seq=0/0, ttl=40 (no response found!)
   10 15.026432  10.20.0.110     192.168.10.5    ICMP      42 Echo (ping) request  id=0x9494, seq=0/0, ttl=50 (no response found!)
   11 15.026504  10.20.0.110     192.168.10.6    ICMP      42 Echo (ping) request  id=0x00ca, seq=0/0, ttl=41 (no response found!)
   12 15.026569  10.20.0.110     192.168.10.7    ICMP      42 Echo (ping) request  id=0x1c97, seq=0/0, ttl=44 (no response found!)
   13 15.026632  10.20.0.110     192.168.10.8    ICMP      42 Echo (ping) request  id=0x3c91, seq=0/0, ttl=54 (no response found!)
   14 15.026696  10.20.0.110     192.168.10.9    ICMP      42 Echo (ping) request  id=0xafb5, seq=0/0, ttl=51 (no response found!)
   15 15.026758  10.20.0.110     192.168.10.10   ICMP      42 Echo (ping) request  id=0x58ef, seq=0/0, ttl=38 (reply in 18)
   16 15.026821  10.20.0.110     192.168.10.11   ICMP      42 Echo (ping) request  id=0x4b9d, seq=0/0, ttl=48 (reply in 22)
   17 15.026966  10.20.0.110     192.168.10.12   ICMP      42 Echo (ping) request  id=0x739b, seq=0/0, ttl=38 (reply in 24)
   18 15.028431  192.168.10.10   10.20.0.110     ICMP      42 Echo (ping) reply    id=0x58ef, seq=0/0, ttl=64 (request in 15)
   19 15.028515  10.20.0.110     192.168.10.10   ICMP      70 Destination unreachable (Protocol unreachable)
   20 15.028882  10.20.0.110     192.168.10.15   ICMP      42 Echo (ping) request  id=0xd0ba, seq=0/0, ttl=48 (no response found!)
   21 15.028985  10.20.0.110     192.168.10.16   ICMP      42 Echo (ping) request  id=0xca5a, seq=0/0, ttl=57 (no response found!)
   22 15.029531  192.168.10.11   10.20.0.110     ICMP      42 Echo (ping) reply    id=0x4b9d, seq=0/0, ttl=63 (request in 16)
   23 15.029581  10.20.0.110     192.168.10.11   ICMP      70 Destination unreachable (Protocol unreachable)
   24 15.029635  192.168.10.12   10.20.0.110     ICMP      42 Echo (ping) reply    id=0x739b, seq=0/0, ttl=63 (request in 17)
   25 15.029661  10.20.0.110     192.168.10.12   ICMP      70 Destination unreachable (Protocol unreachable)
   26 15.029849  10.20.0.110     192.168.10.19   ICMP      42 Echo (ping) request  id=0xb58d, seq=0/0, ttl=39 (no response found!)
   27 15.029941  10.20.0.110     192.168.10.20   ICMP      42 Echo (ping) request  id=0xece9, seq=0/0, ttl=38 (no response found!)
   28 15.030154  10.20.0.110     192.168.10.23   ICMP      42 Echo (ping) request  id=0x7fdd, seq=0/0, ttl=41 (no response found!)
   29 15.030282  10.20.0.110     192.168.10.24   ICMP      42 Echo (ping) request  id=0x38b1, seq=0/0, ttl=50 (no response found!)
   30 15.427077  192.168.10.11   10.20.0.110     SSH      122 Server: Encrypted packet (len=68)
   31 15.634304  10.20.0.110     192.168.10.11   TCP       54 1278 → 22 [ACK] Seq=37 Ack=137 Win=63220 Len=0
   32 15.774937  10.20.0.110     192.168.10.11   SSH       90 Client: Encrypted packet (len=36)
   33 15.876235  192.168.10.11   10.20.0.110     TCP       54 22 → 1278 [ACK] Seq=137 Ack=73 Win=65535 Len=0
   34 16.026622  10.20.0.110     192.168.10.27   ICMP      42 Echo (ping) request  id=0xb026, seq=0/0, ttl=44 (no response found!)
   35 16.040345  10.20.0.110     192.168.10.30   ICMP      42 Echo (ping) request  id=0xc0a1, seq=0/0, ttl=49 (no response found!)
   36 16.040469  10.20.0.110     192.168.10.31   ICMP      42 Echo (ping) request  id=0xbbf3, seq=0/0, ttl=45 (no response found!)
   37 16.040563  10.20.0.110     192.168.10.32   ICMP      42 Echo (ping) request  id=0xc269, seq=0/0, ttl=45 (no response found!)
   38 16.040653  10.20.0.110     192.168.10.33   ICMP      42 Echo (ping) request  id=0x344a, seq=0/0, ttl=38 (no response found!)
   39 16.040755  10.20.0.110     192.168.10.34   ICMP      42 Echo (ping) request  id=0x8a3a, seq=0/0, ttl=51 (no response found!)
   40 16.040849  10.20.0.110     192.168.10.35   ICMP      42 Echo (ping) request  id=0xb147, seq=0/0, ttl=46 (no response found!)
   41 16.040944  10.20.0.110     192.168.10.36   ICMP      42 Echo (ping) request  id=0x72e7, seq=0/0, ttl=48 (no response found!)
   42 16.041057  10.20.0.110     192.168.10.37   ICMP      42 Echo (ping) request  id=0x0efe, seq=0/0, ttl=45 (no response found!)
   43 16.041173  10.20.0.110     192.168.10.38   ICMP      42 Echo (ping) request  id=0x8c6c, seq=0/0, ttl=58 (no response found!)
   44 16.041681  10.20.0.110     192.168.10.39   ICMP      42 Echo (ping) request  id=0x65bd, seq=0/0, ttl=58 (no response found!)

Frame 1: 122 bytes on wire (976 bits), 122 bytes captured (976 bits)
0000  4e d7 9d 30 30 66 22 44  c0 2d ed 18 08 00 45 08   N··00f"D ·-····E·

Packets: 47423 · Displayed: 47423 (100.0%)
```

Fig. 3.9 Example of Using Wireshark for Network Traffic Analysis

Through participation in the red and blue team competition, students gain valuable hands-on experience in configuring network devices, identifying vulnerabilities, and securing network infrastructure.

Throughout the exercise, students develop skills in network scanning, traffic analysis, and threat detection, which provides them with a deeper understanding of network operations and security methods.

Red team members learn to apply scanning techniques, identify vulnerabilities in services running on open ports, and use exploits to penetrate vulnerable systems. They gain proficiency with tools like nmap and Metasploit, understand ethical hacking methods, and appreciate the importance of vulnerability remediation, helping them grasp the mindset of potential adversaries.

The blue team hones its skills in actively defending the network, identifying cyber threats, responding swiftly to incidents, and documenting cyber incidents in a clear and accessible manner. By working with tools like Wireshark, Suricata, and Snort, students practice techniques for traffic analysis and attack blocking, enabling them to operate effectively both in monitoring mode and in active defense against attackers.

This training format provides students with a comprehensive understanding of cybersecurity, fostering strategic thinking and teamwork skills essential for real-world cybersecurity roles.

3.4 PRACTICAL CYBER DEFENSE SCENARIO FOR CORPORATE INFRASTRUCTURE DEVELOPED IN COLLABORATION WITH UNDERDEFENSE

A new training scenario on the SSEIS cyber polygon, developed in partnership with UnderDefense, focuses on in-depth cybersecurity skills, particularly in detecting and exploiting vulnerabilities within complex network infrastructures. This scenario is designed for hands-on learning of ethical hacking methods and protection of critical systems, such as GitLab servers and Windows Active Directory (AD) infrastructure. Students participating in this scenario gain practical knowledge of attack and defense stages within corporate networks, aligning their experience closely with real-world cybersecurity tasks.

The primary goal of the scenario is to familiarize students with comprehensive techniques that allow them to understand the architecture of modern corporate networks, identify potential threats, and implement effective security measures. Through this scenario, students develop the following key skills:

1. Identifying external perimeter vulnerabilities: using brute-forcing techniques to locate services and exploiting vulnerabilities for remote code execution (RCE).

2. GitLab and SSH operations: exploring open repositories for confidential data (e.g., SSH keys) and accessing the server through SSH.

3. Privilege escalation on Ubuntu server: utilizing known exploits, such as pkexec (LPE), to gain elevated privileges.

4. Active directory (AD) attacks: discovering and exploiting weaknesses in SMB configurations, executing Kerberoasting, and using techniques to obtain administrative rights.

5. Traffic monitoring and interception: analyzing data, including LDAP requests, to practice information protection skills.

The scenario's development was made possible through collaboration with UnderDefense, a company that provides cybersecurity support and expertise in ethical hacking methodologies. UnderDefense contributed expert knowledge in attack scenario formulation, vulnerable system setup, educational case creation, and access to up-to-date vulnerabilities, particularly with GitLab and Active Directory. This collaboration ensures that the training tasks meet modern industry standards and cybersecurity needs.

Scenario Components: external and internal network perimeters:

1. External perimeter: the external perimeter includes an Ubuntu 20.04 server hosting GitLab, with a Pre-Auth RCE vulnerability (CVE-2021-22205) and other exploits like pwnkit for privilege escalation. In this phase, students use brute-forcing methods to locate GitLab, access an SSH key, establish a connection with the server, and subsequently elevate privileges to an administrator level.

2. Internal perimeter (AD network): the internal perimeter contains an Active Directory-based network, including a Domain Controller (DC) and two workstations. Students explore SMB signing issues, LLMNR, and perform various AD attacks, such as Kerberoasting, traffic interception, and NTDS.DIT extraction. This stage simulates real threats arising from improper corporate network configurations.

This scenario allows students to practice network infiltration across different complexity levels and develop a comprehensive approach to identifying, analyzing, and mitigating cybersecurity threats.

In this scenario, the external perimeter is represented by an Ubuntu 20.04 server with GitLab installed, serving as the primary external target for the red team. The server configuration is specifically designed to illustrate key principles of ethical hacking and assess the security of an organization's external infrastructure. The server includes several known vulnerabilities that students can leverage to complete their tasks.

GitLab serves as one of the primary services commonly used in real corporate infrastructures for code hosting, project management, and collaboration. In this scenario, GitLab is configured as a publicly accessible service on the Ubuntu server, allowing students to use subdomain enumeration and visible service analysis techniques to locate GitLab within the network. Once detected, students can attempt to exploit GitLab vulnerabilities to gain access to the server.

The server runs an outdated version of Ubuntu 20.04 LTS, containing several critical vulnerabilities that can be used for privilege escalation. One major vulnerability is pwnkit (related to the pkexec command), which enables Local Privilege Escalation (LPE). Pwnkit is a well-known vulnerability that allows a non-authenticated user to gain root access to the system by exploiting a misconfiguration in the pkexec command.

This server setup, which includes such vulnerabilities, enables students to practice exploiting weaknesses in outdated software. The primary objective of this phase is to teach students the impact of these vulnerabilities on infrastructure security and to demonstrate how a lack of regular updates can create entry points for attackers.

The attack phases on the external perimeter in this scenario are structured to allow students to complete the full cycle from service detection to system control through vulnerability exploitation. Each step is based on real-world penetration methods, allowing students to gain hands-on experience with cybersecurity tools:

1. Discovery phase using ffuf: in the initial phase, the attack team uses ffuf (Fast web Fuzzer) to brute-force subdomains and virtual hosts to locate the GitLab server, which will be the target for further penetration. With ffuf, students search for existing subdomains and hidden services that may not always be visible through standard scans. This stage helps students develop skills in active information gathering and configuring ffuf for discovering subdomain variations that may house critical services.

2. Exploitation of the Pre-Auth RCE vulnerability (CVE-2021-22205): after locating the GitLab server, the next step is exploiting the Pre-Auth RCE vulnerability (CVE-2021-22205). This vulnerability allows command execution on the server without requiring authentication. Students can use this vulnerability to gain initial access to the server, highlighting the severe consequences of not regularly updating GitLab. In this phase, students learn to use public exploits for RCE vulnerabilities and gain an understanding of the importance of proper server configuration.

After gaining access, students continue their investigation by reviewing the contents of Git repositories, which may contain confidential information. One possible way to obtain SSH access is by finding SSH keys that were accidentally left in a public repository. After locating and verifying the SSH key, students establish an SSH connection to the server using the discovered key, granting them full access to the system. This step emphasizes the importance of proper access key management and removing sensitive information from public repositories.

The final stage involves privilege escalation on the server to obtain root-level access. Students exploit the pkexec vulnerability (a flaw in pwnkit) that allows Local Privilege Escalation (LPE) on Ubuntu. Using this vulnerability, students can escalate privileges and gain complete control over the server. This stage introduces students to the risks associated with outdated software and provides insight into methods for mitigating such vulnerabilities.

By following these steps, students learn the full process of penetrating the external perimeter of a corporate network, which includes active target discovery, vulnerability exploitation for access, searching for sensitive data in repositories, and privilege escalation to establish system control.

The internal perimeter of the corporate network in this scenario is represented by an Active Directory (AD) infrastructure that includes one AD server and two workstations. This setup mimics a typical corporate network architecture, which is often a target for attacks aimed at gaining access to internal resources. The main objective for the attack team is to compromise AD and access user data, while the defense team’s goal is to prevent attacks and protect critical infrastructure components.

The internal network is divided into several components that replicate a real corporate system environment. In this part of the scenario, students work with typical AD configurations and common vulnerabilities that often remain exposed in networks due to misconfigurations:

1. Active directory (AD) server: the AD server functions as a centralized management system responsible for user authentication, access control, and permissions management within the internal network. The AD server stores user accounts, security policies, and other configurations, making it a primary target for attacks aimed at accessing critical information. Students explore attack methods on AD, such as Kerberoasting and information gathering through LDAP queries, to understand vulnerabilities characteristic of this infrastructure.

2. Two workstations: the workstations are components of the internal network and play an important role in this scenario. They are used as intermediate targets through which the attack team can reach the AD server or other resources. In a real corporate environment, workstations are often attacked due to accessible resources, SMB settings, and other services. In this scenario, students exploit vulnerabilities related to the absence of SMB signing and LLMNR (Link-Local Multicast Name Resolution) to access internal resources and gather information about other devices on the network.

SMB Signing and LLMNR Configuration. SMB Signing: The lack of SMB signing on workstations makes them vulnerable to attacks that allow adversaries to intercept or modify data transferred between devices.

In this scenario, students exploit this vulnerability to conduct a Pass-the-Hash attack, enabling them to compromise workstations and use the captured hashes to access other resources. LLMNR (Link-Local Multicast Name Resolution): LLMNR is a protocol used for resolving local names within a network, but it can become a security risk if not properly protected. Students learn to use LLMNR to capture account hashes and perform attacks related to local name resolution implementation. This protocol allows attackers to gather information for further attacks on AD or other network resources.

The AD server is the main target that the attack team aims to reach. Compromising it provides access to all user accounts and critical information about the internal network.

Workstations are used as intermediate targets for gathering information, breaking user accounts, and accessing the AD server. Students learn to leverage weak configurations to gradually penetrate the network, which helps them develop skills in privilege escalation and identifying vulnerabilities in Windows-based environments.

Attacks on the internal AD network perimeter in this scenario involve gradual network penetration through vulnerability discovery, credential compromise, and access to critical systems. This progression allows students to experience the full escalation path within a corporate environment and gain practical skills with real attack methods.

In the first stage, the attack team scans the network for vulnerabilities to identify security misconfigurations on three hosts within the internal network, specifically:

1. Lack of SMB signing, allowing attackers to intercept or alter data exchanged between hosts.

2. Active LLMNR (Link-Local Multicast Name Resolution), which can be used to intercept hostname requests and capture account hashes.

Using these vulnerabilities, students learn to gain initial access to hosts, which is an essential step for further escalation.

After identifying vulnerabilities, the attack team conducts an SMB protocol attack by exploiting the lack of signing to capture user1 account hashes. This phase teaches students Pass-the-Hash techniques, allowing them to authenticate using captured hashes without knowing the password. The obtained hashes can then be cracked with password recovery tools, providing compromised credentials that grant access to AD.

The next step involves executing a Kerberoasting attack, which allows attackers to request Kerberos tickets for privileged accounts (specifically for user2, who is a local administrator on machine1). A Kerberos ticket may contain the account's hash, which is used for system authentication. This phase helps students learn how to leverage Kerberos to access vulnerable accounts, especially in networks where local administrators are present on different hosts.

After obtaining user2's hash, students use tools to crack the Kerberos hash to retrieve the password for user2, who has administrator rights on machine1. Administrator access allows the attack team to examine the host configuration, modify security settings, and further penetrate the AD internal network. This stage demonstrates the critical importance of properly configuring and securing administrator accounts.

At this point, the attack team uses the secretsdump tool to extract sensitive data from machine1, where user2 has administrator rights. With secretsdump, students retrieve account data, including hashes for user3, which can then be used to access machine2 via RDP. This expands the attackers' control over additional network hosts, providing access to important resources.

The final phase involves using LDAPExplorer2.exe to access the LDAP server while intercepting traffic. Once connected, the attack team can query data through the LDAP protocol and set up their own LDAP listener to collect account information. After this, the attackers can log into the Domain Controller (DC) and download the NTDS.DIT file, which contains critical domain account data, including password hashes for all users.

This scenario requires various software and tools to create a realistic training environment that mirrors corporate network infrastructure. All components are configured to ensure access control, security monitoring, and environmental isolation, allowing students to safely practice cybersecurity tasks:

1. Windows Server (versions 2016, 2019, or 2022): Windows servers are essential for building an Active Directory (AD) infrastructure, widely used in corporate settings. The choice of version (2016, 2019, or 2022) depends on the training requirements and allows students to become familiar with different AD features and administrative methods, as well as configuring Domain Controllers (DC), managing Group Policies, and setting security parameters.

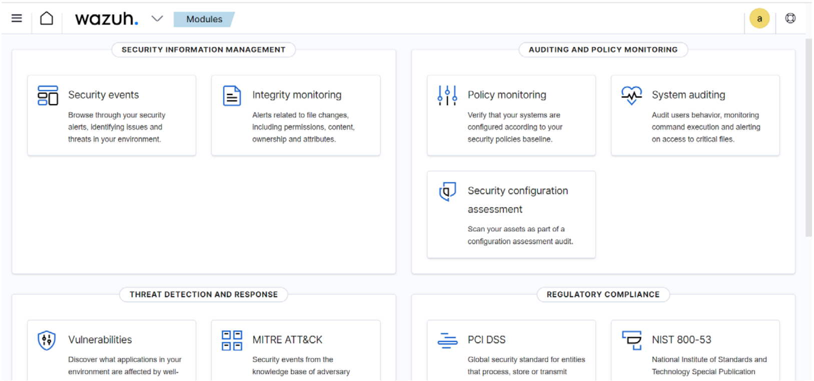
2. Windows 10 Pro: Windows 10 Pro serves as the operating system for workstations in the internal network. This OS is widely used in corporate environments and provides the necessary

functionality to model typical workstations. Windows 10 Pro supports SMB and RDP protocols, and allows for security policy configuration, making it ideal for studying vulnerabilities like SMB and LLMNR.

3. Ubuntu 20.04: an outdated version of Ubuntu 20.04 is used in the scenario, which is vulnerable to privilege escalation exploits (e.g., pwnkit). This OS is installed on the GitLab server and acts as the attack entry point. Ubuntu 20.04 allows students to explore open-source systems, analyze Linux vulnerabilities, and work with services such as SSH and GitLab.

4. LDAPExplorer2.exe: a tool for working with LDAP queries, LDAPExplorer2.exe enables students to query Active Directory, analyze AD structure, retrieve account data, and gather sensitive information. This tool helps students understand LDAP authentication mechanisms and client-server interactions.

5. Wazuh: a security monitoring and incident management system, Wazuh enables log analysis and network activity monitoring (**Fig. 3.10**). It provides detailed network event information, helping students investigate security logs, detect anomalies, and respond to incidents.



SECURITY INFORMATION MANAGEMENT

Security events
Browse through your security alerts, identifying issues and threats in your environment.

Integrity monitoring
Alerts related to file changes, including permissions, content, ownership and attributes.

AUDITING AND POLICY MONITORING

Policy monitoring
Verify that your systems are configured according to your security policies baseline.

System auditing
Audit users behavior, monitoring command execution and alerting on access to critical files.

Security configuration assessment
Scan your assets as part of a configuration assessment audit.

THREAT DETECTION AND RESPONSE

Vulnerabilities
Discover what applications in your environment are affected by well-

MITRE ATT&CK
Security events from the knowledge base of adversary

REGULATORY COMPLIANCE

PCI DSS
Global security standard for entities that process, store or transmit

NIST 800-53
National Institute of Standards and Technology Special Publication

Fig. 3.10 Example of Wazuh Interface

To support a realistic and secure learning environment, an isolated network has been created, fully separated from the university or corporate infrastructure. This network isolation ensures safety and allows students to practice complex cyber operations without risking other networks.

LLMNR and SMB: the LLMNR (Link-Local Multicast Name Resolution) and SMB protocols are configured for ease of use in training, specifically for demonstrating vulnerabilities associated with these protocols. LLMNR settings in the network allow for modeling interception attacks, while the absence of SMB signing simulates real corporate networks, which can be vulnerable to SMB attacks.

Access control: various access control settings are applied to restrict access and prevent unauthorized actions. These settings help students learn to configure security policies for external threat protection and manage user rights within the network.

These software components and network configurations provide a realistic corporate environment simulation, enabling students to practically master key aspects of cybersecurity, from monitoring and log analysis to access configuration and virtual resource management.

Skills acquired during the scenario:

1. Vulnerability detection, risk assessment, and defense method development: students learn to identify critical infrastructure vulnerabilities and assess the risks associated with these weaknesses. Developing defense methods includes practicing attack scenarios and preventive actions, providing participants with an in-depth understanding of cybersecurity. This skill is essential for security professionals who need to assess threats and propose solutions to protect systems against potential attacks.

2. Experience in AD management, GitLab protection, and LDAP for remote access: the scenario provides hands-on experience with Active Directory management, a key component of corporate networks. Students learn methods for configuring and securing GitLab, including understanding risks related to repository management. Skills in using LDAP for remote access and conducting LDAP queries give students important knowledge about AD structure and security, helping them secure data in AD environments in the future.

3. Attack and defense skills development, including brute-forcing, RCE, SMB, and LPE practice: participation in the scenario allows students to practice attack skills such as brute-forcing subdomains and virtual hosts, RCE attacks (remote code execution), and attacks on the SMB protocol, which are often left open in corporate environments. Additionally, students gain practical experience in local privilege escalation (LPE) through Linux system vulnerabilities. This comprehensive skill set enables future cybersecurity professionals to understand attack methods and protect systems from such threats.

This training scenario is as close as possible to real situations that cybersecurity professionals may encounter in their jobs. Thanks to the scenario structure, students experience the full cycle of attacks and defense – from identifying vulnerabilities to developing a comprehensive defense system. Training in environments that simulate real threats allows them to gain confidence in their technical skills and apply acquired knowledge to protect against current cyber threats.

The hands-on experience in ethical hacking and cyber defense gained through this scenario provides students with a deeper understanding of internal processes in corporate networks, helps them avoid critical configuration errors, and effectively protects information that may be targeted in attacks. As a result, they enter the job market with valuable practical skills and the readiness to take on complex cybersecurity tasks.

3.5 RESULTS OF CYBER POLYGON UTILIZATION IN EDUCATION

The cyber polygon at the Department of Solid-State Electronics and Information Security is actively used for hands-on training of students at all levels, from freshmen to master's students. Over its period of use, the range has demonstrated excellent results: students have shown high interest in new types of tasks involving real-world cyber threat modeling and protective measures. Many students provided valuable feedback, which has served as the foundation for further refinement of scenarios and expanded functionality of the cyber polygon, making the learning process more dynamic and aligned with professional challenges.

In addition to educational practice, the cyber polygon serves as a platform for preparing student teams for various competitions and professional events in cybersecurity. This environment helps students develop teamwork skills, hone technical abilities, and prepare for ethical hacking competitions, Capture the Flag (CTF) events, and other contests where quick threat response and effective collaboration are essential.

CONCLUSION

The cyber polygon at the Department of Solid-State Electronics and Information Security offers a unique training environment that allows students to practice defensive and offensive skills in conditions closely mirroring real corporate networks. Working with the cyber polygon provides students with access to various scenarios and tools, enabling them to develop a well-rounded set of practical skills essential for a modern cybersecurity professional.

The first scenario, using Qualys, focuses on developing skills in automated vulnerability scanning for web applications. Students gain experience with one of the leading tools for security assessment, which enables them to identify vulnerabilities, analyze risks, and plan mitigation strategies. Through this scenario, participants enhance their ability to systematically analyze vulnerabilities and create reports with recommendations for security improvement.

The second scenario, involving Metasploitable 2, allows students to undertake ethical hacking tasks in a controlled environment. Due to its numerous vulnerabilities, this setting provides students with the opportunity to study attack methods such as port scanning, exploit searches, and server penetration through known vulnerabilities. Practicing these techniques gives students insight into real attack vectors and effective defensive measures.

The third scenario, developed in collaboration with UnderDefense, includes work with GitLab, Active Directory, and Ubuntu. Students start with the external perimeter, conducting attacks on an Ubuntu server with GitLab, and progress to a complex internal network with Active Directory, where they encounter realistic tasks aimed at breaching AD through vulnerabilities such as the absence of SMB signing and Kerberoasting attacks. This scenario offers in-depth training in network management and protection of critical corporate infrastructure components.

Through these three scenarios, the cyber polygon provides participants with comprehensive training in cybersecurity, covering both offensive and defensive aspects. Students learn to identify vulnerabilities, develop effective measures for cyber threat protection, and work with modern cybersecurity tools. The experience gained prepares them to begin their professional careers with confidence and contributes to an overall increase in cybersecurity standards in society.

ACKNOWLEDGMENTS

The authors extend their sincere gratitude to UnderDefense for their substantial support and productive collaboration in developing training scenarios for the cyber polygon at the Department of Solid-State Electronics and Information Security, Uzhhorod National University. Thanks to their professionalism, expert knowledge, and active involvement, we were able to create innovative practical tasks that greatly enhance the quality of cybersecurity education for students. We look forward to continued collaboration, which will further contribute to the development of students' practical skills and prepare them for work in the modern cyberspace.

REFERENCES

1. Dovhan, O. D., Hulak, H. M., Hryn, A. K., Melnyk, S. V. (2012). Metodolohiia zakhystu informatsii. Kyiv: Naukovo-vydavnychyi tsentr Natsionalnoi akademii Sluzhby bezpeky Ukrainy, 184.
2. Buriachok, V. L., Toliupa, S. V., Anosov, A. O., Kozachok, V. A., Lukova-Chuiko, N. V. (2015). Systemnyi analiz ta pryiniattia rishen v informatsiinii bezpetsi. Kyiv: DUT, 345.
3. Buriachok, V. L., Toliupa, S. V., Semko, V. V., Buriachok, L. V., Skladannyi, P. M., Lukova-Chuiko, N. V. (2016). Informatsiinyi ta kiberprostory: problemy bezpeky, metody ta zasoby borotby. Kyiv: DUT – KNU, 178.
4. Hudmen, M. (2019). Zlochyny maibutnoho. Kharkiv: Fabula, 592.
5. Kisku, D. R., Gupta, P., Sing, J. K. (Eds.). (2016). Advances in Biometrics for Secure Human Authentication and Recognition. CRC, 352.
6. Lisovska, Yu. (2019). Kiberbezpeka. Ryzyky ta zakhody. Kyiv: Kondor, 272.
7. Kurban, O. V. (2016). Suchasni informatsiini viiny v merezhevomu on-lain prostori. Kyiv: VIKNU, 286.
8. Vemuri, V. R. (2019). Enhancing computer security with smart technology. CRC Press, 288.
9. Khoroshko, V. O., Kryvoruchko, O. V., Brailovskyi, M. M. et al. (2019). Zakhyst system elektronnykh komunikatsii. Kyiv: KNTEU, 164.
10. Bobalo, Yu. Ya., Dudykevych, V. B., Mykytyn, H. V. (2020). Stratehichna bezpeka systemy "obiekt – informatsiina tekhnolohiia". Lviv: Lvivska politekhnika, 260.

11. Hrebeniuk, A. M., Rybalchenko, L. V. (2020). Osnovy upravlinnia informatsiinoiu bezpekoiu. Dnipro: Dnipropetrovskyi derzhavnyi universytet vnutrishnikh sprav, 144.
12. Prysiazhniuk, M. M., Farmahei, O. I., Chekhovska, M. M. et al.; Ostroukhov, V. V. (Ed.) (2021). Informatsiina bezpeka. Kyiv: Vydavnytstvo Lira-K, 412.
13. Ostapov, S. Ye., Yevseiev, S. P., Korol, O. H. (2021). Tekhnolohii zakhystu informatsii. Lviv: Novyi Svit–2000, 678.
14. Kohut, Yu. (2021). Kiberbezpeka ta ryzyky tsyfrovoi transformatsii kompanii. Kyiv: Konsaltynhova kompaniia Sidkon, 372.
15. Korobeinikova, T. I., Zakharchenko, S. M. (2021). Tekhnolohii zakhystu lokalnykh merezh na osnovi obladnannia CISCO. Lviv: Lvivska politekhnika, 232.
16. Kohut, Yu. (2021). Kiberteroryzm. Istoriia, tsili, obiekty. Kyiv: Konsaltynhova kompaniia Sidkon, 304.
17. Panek, C. (2020). Networking fundamentals. Hoboken: John Wiley & Sons, Inc., 319.
18. Samuel, A. (2021). Network ethical hacking and penetration testing. Los Angeles, 409.
19. Diogenes, Y., Ozkaya, E. (2018). Cybersecurity – attack and defense strategies. Packt Publishing, 326.
20. Davis, R. (2020). The art of network penetration testing. Manning Publications, 310.
21. Herzog, R., O'Gorman, J., Aharoni, M. (2017). Kali Linux revealed: Mastering the penetration testing distribution. Offsec Press, 342.
22. Parasram, S., Samm, A., Boodoo, D., Johansen, G., Allen, L., Heriyato, T., Ali, S. (2018). Kali Linux – assuring security by penetration testing. Packt Publishing, 527.
23. Metasploitable 2 Exploitability Guide. Available at: https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/
24. Qualys Web Application Scanning Datasheet. Available at: https://cdn2.qualys.com/docs/mktg/was-datasheet.pdf
25. Qualys Web Application Scanning Getting Started Guide (2024). Available at: https://cdn2.qualys.com/docs/qualys-was-getting-started-guide.pdf
26. Foreshaw, J. (2018). Attacking network protocols: A hacker's guide to capture, analysis, and exploitation. San Francisco, 340.
27. Maiwald, E. (2001). Network security: A beginner's guide. The McGraw-Hill Companies, 401.
28. Troncone, P., Albing, C. (2019). Cybersecurity ops with Bash: Attack, defend, and analyze from the command line. O'Reilly Media, 288.
29. Grimes, R. A. (2017). Hacking the hacker: Learn from the experts who take down hackers. Wiley, 320. https://doi.org/10.1002/9781119396260
30. Seitz, J., Arnold, T. (2021). Black Hat Python: Python programming for hackers and pentesters. No Starch Press, 216.
31. Graham, D. (2021). Ethical hacking: A hands-on introduction to breaking In. No Starch Press, 376.
32. UnderDefense. Available at: https://underdefense.com/about-us/
33. Metasploitable 2. Available at: https://docs.rapid7.com/metasploit/metasploitable-2/