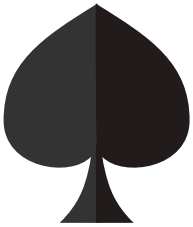
Project: Vulnerbility

Penetration Testing module



Ace Tools

Presented by

Eithan Saragosti

Table of Contents

Introduction:

Ace Tools Vulner: A Network Vulnerability Scanner Script

This document introduces the Bash script "Ace Tools Vulner," a network vulnerability scanner designed to identify potential weaknesses in your network infrastructure.

Key Features:

- Scan Options: Choose between a Basic Scan (focusing on service identification and version detection) or a Full Scan (including vulnerability script scanning with Searchsploit integration).
- Weak Credential Checks: Attempts to identify weak passwords using a provided password list for common services like SSH, FTP, RDP, and Telnet.
- Vulnerability Mapping: Utilizes Searchsploit to gather information about discovered vulnerabilities (Full Scan only).
- Result Management: Allows users to search within the generated reports and save the results as a compressed ZIP file.
- User-Friendly Interface: Provides clear instructions and informative messages throughout the scanning process.

Benefits:

By employing Ace Tools Vulner, you can proactively identify potential security risks in your network. The script helps to:

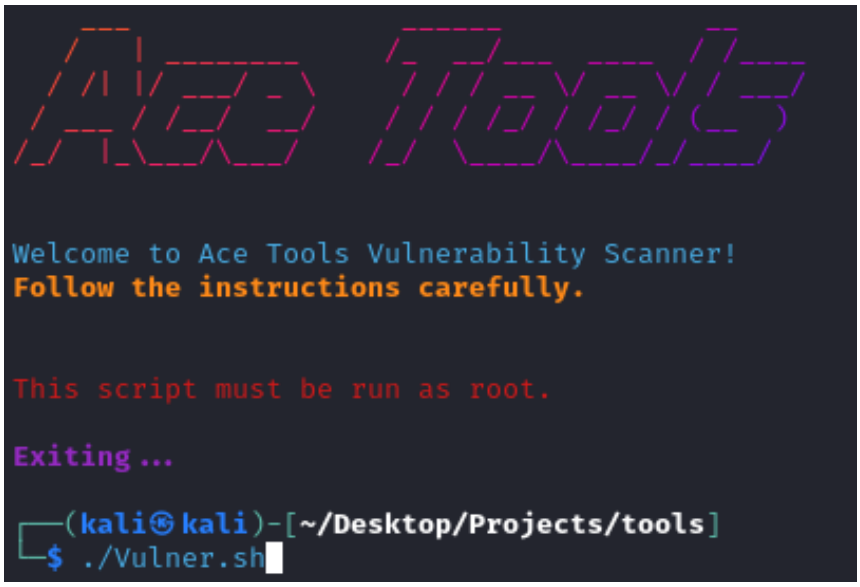
- Reduce Attack Surface: Identify potential entry points that attackers could exploit with weak credentials or known vulnerabilities.
- Improve Security Posture: Gain insights into your network's security vulnerabilities to prioritize remediation efforts.
- Automate Vulnerability Scanning: Streamline the process of vulnerability assessment, saving time and resources.

Disclaimer:

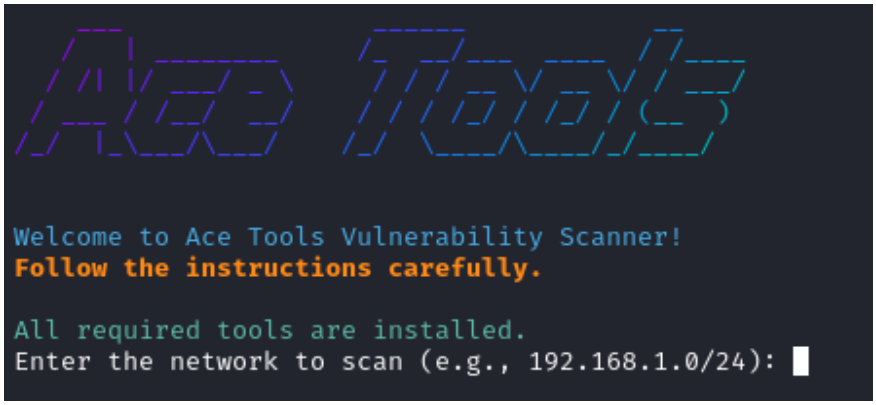
It's important to note that Ace Tools Vulner is for educational purposes only. Running vulnerability scans against systems you don't own or have permission to scan is illegal. It's crucial to obtain proper authorization before conducting any network scans.

Demo of the Project:

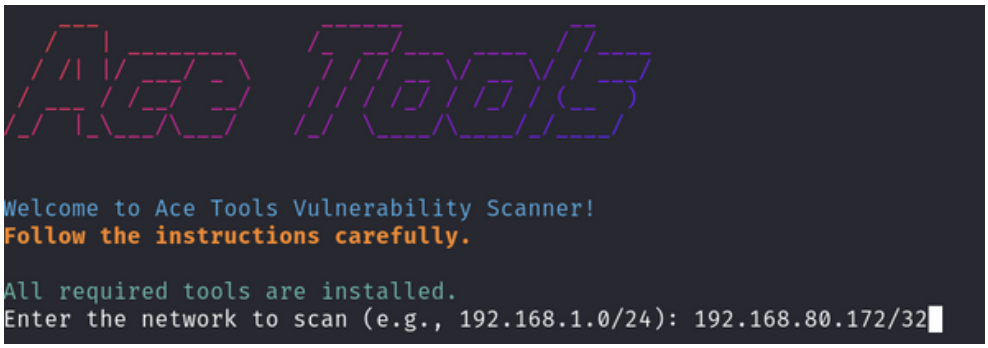
1. Running the script as a normal user will output a message saying you must have root privileges and exit.



2. Running the script as root will make sure every tool needed is installed and will output the first instruction which is to enter the network you wish to scan.

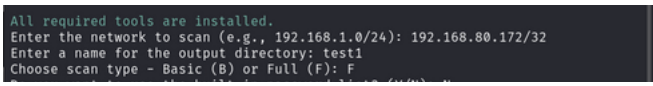


3. In this demo i will be investigating a single machine that has a few vulnerabilities so the input will be as such:

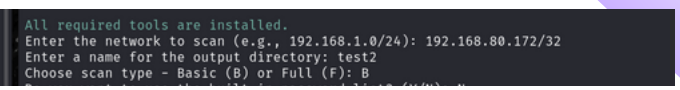


4. After inputting the network we wish to scan we are asked to name the directory for the ouput and the type of scan we wish to execute.

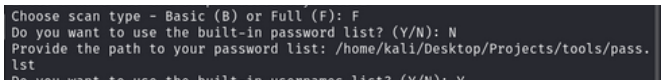
Basic scan

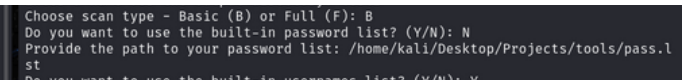


Full scan

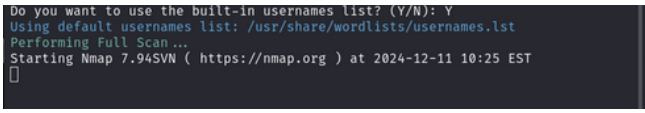


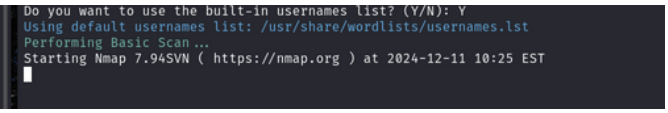
5. After that we are asked to choose between the default password list (rockyou.txt) or to provide one of our own, For the sake of the demo I chose one of my own:



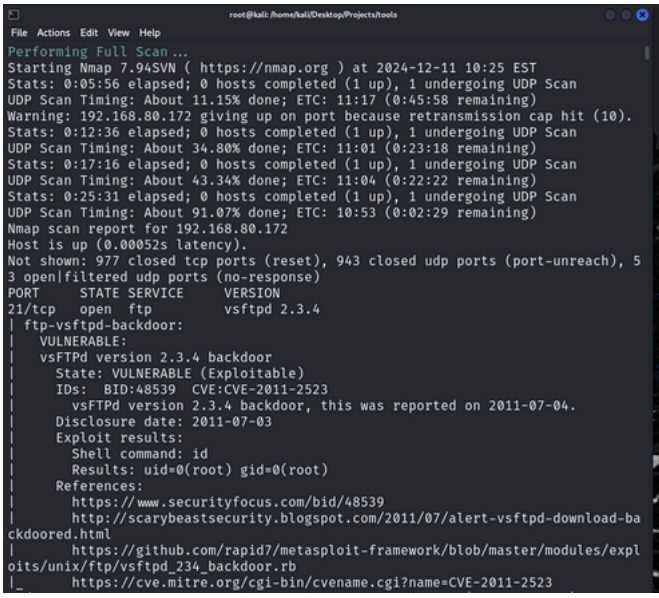


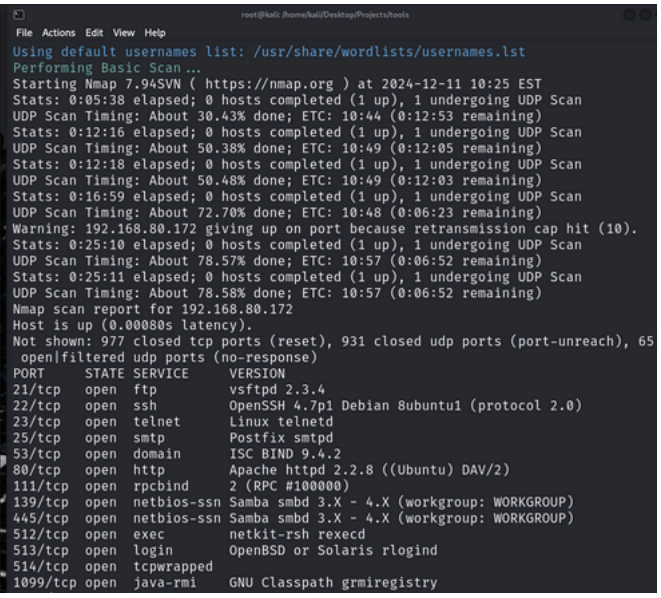
6. After that we are asked about a username list i chose to use the Default one (usernames.lst), now the script runs and preforms the scan.





7. You can see the diffrences between the basic and full scan:

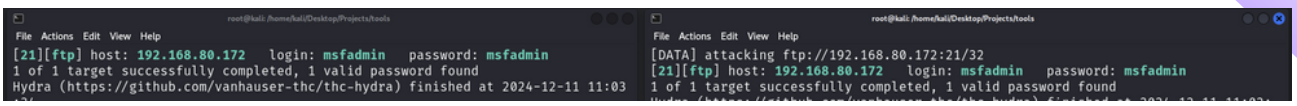




8. you can see the both scans managed to find a weak password

Basic scan

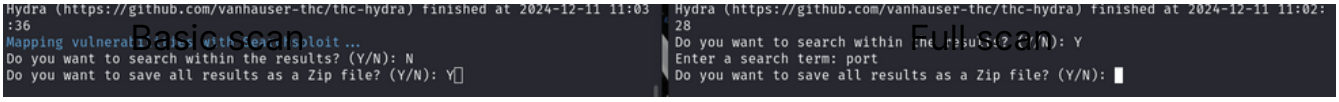
Full scan



9. Now we are asked if we want to search within the results the prompt opens the results with the “less” option for the user to search ports for example:

Basic scan

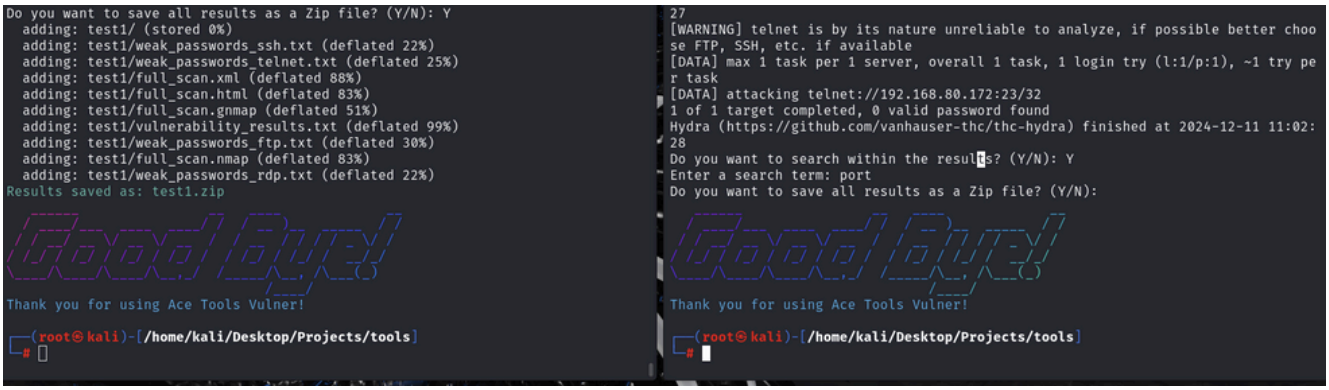
Full scan



10.Lastly we are asked if we wish to save all the results in a single zip file, a “Goodbye” message appears and exits the script.

Basic scan

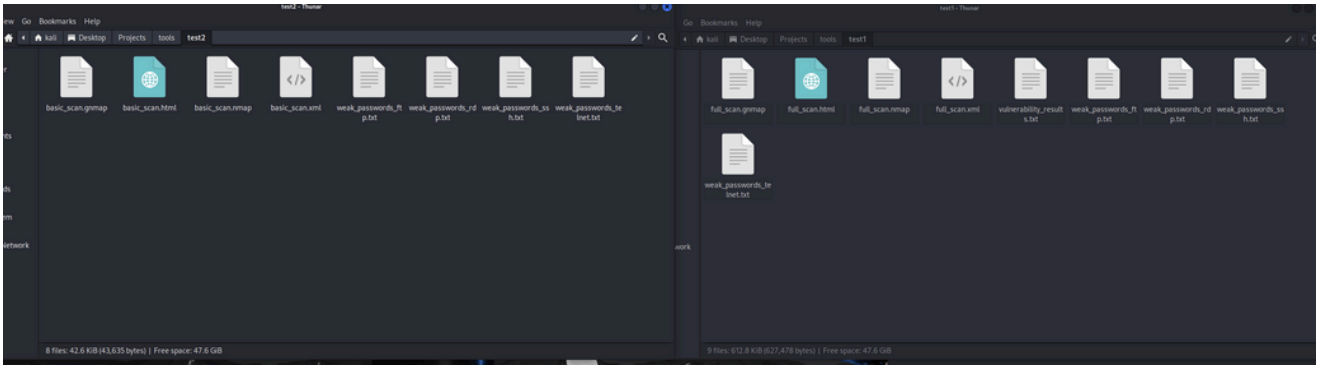
Full scan



11. the tool’s results are saved as such:

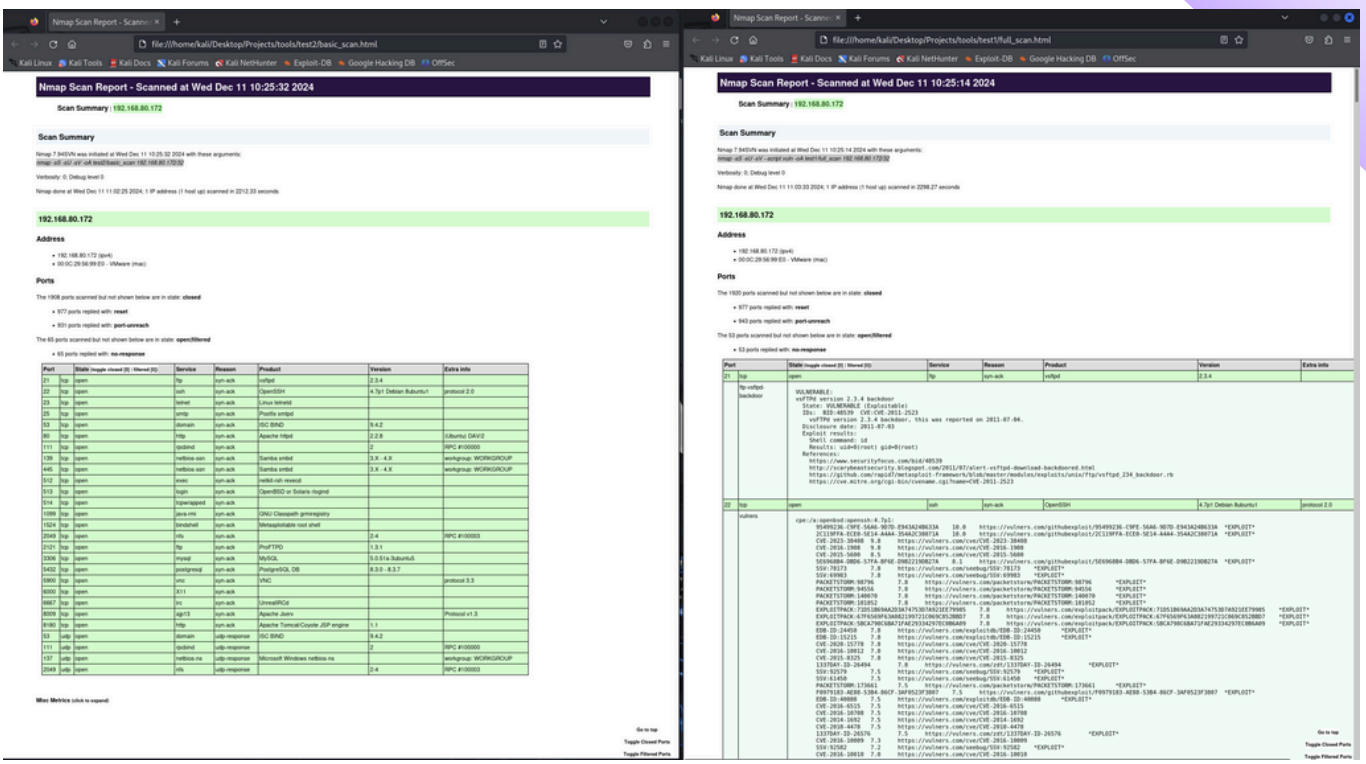
Basic scan

Full scan



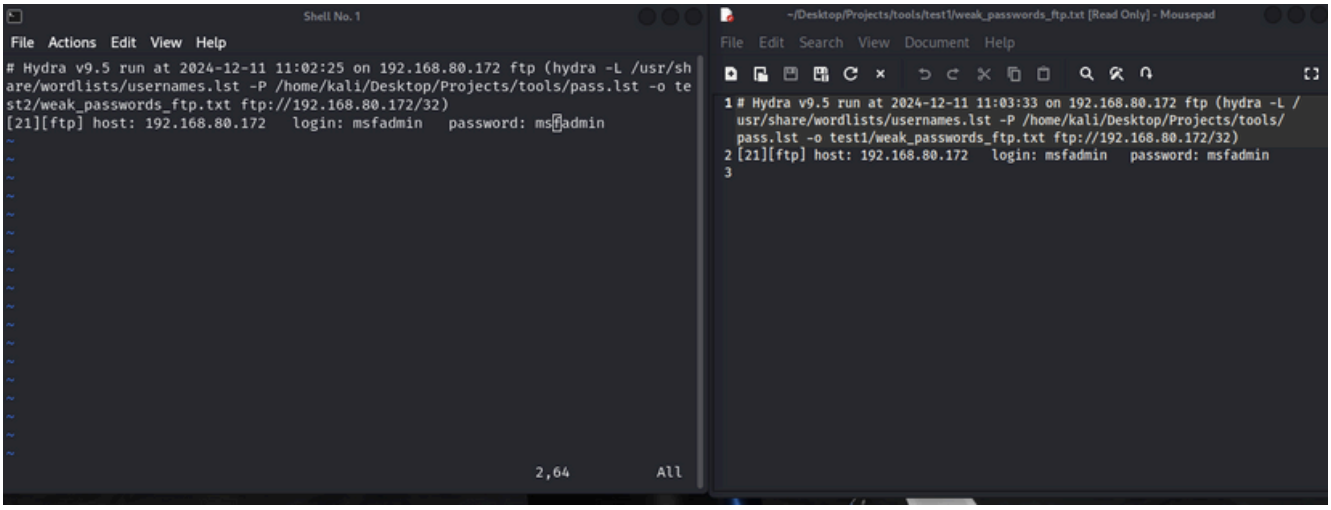
12. Example of the Nmap results as html files for a user friendly view:

Basic scan Full scan



13. Example of a weak password found during the execution of the script:

Basic scan Full scan



COLORS AND FORMATTING:

To enhance the user experience, the script employs ANSI escape codes for color-coding its output. This makes the script's messages more visually distinct and user-friendly:

- Green: Indicates success or positive feedback, such as successful installation of tools or completion of tasks.
- Cyan: Highlights informational messages, like step-by-step instructions and prompts for user input.
- Red: Flags errors or critical issues that require immediate attention, such as missing files or permission errors.
- Yellow: Warns the user about missing tools, potential missteps, or areas that require caution.
- Magenta: Used for neutral or exit-related messages, providing a polished and professional script termination.
- No Color: Resets the terminal output to default formatting to avoid visual clutter.

These color codes ensure that critical information is easily recognizable, improving usability and reducing user error.

Tools & Technologies Used:

NMAP:

- **Purpose:** Nmap specializes in network discovery and security auditing. It identifies hosts, services, and vulnerabilities within a network.
- **Usage in the Script:** The script uses Nmap to perform both Basic and Full Scans. Basic Scans identify live hosts and services, while Full Scans utilize the Nmap Scripting Engine (NSE) to detect specific vulnerabilities.
- **How it Works:** Nmap sends crafted packets to target hosts and analyzes responses to determine open ports, running services, and potential vulnerabilities. NSE scripts extend its functionality by probing for specific security flaws and producing detailed assessments.

HYDRA:

- **Purpose:** Hydra is a versatile credential brute-forcing tool, capable of testing login credentials for various network services.
- **Usage in the Script:** he script leverages Hydra to test username-password combinations on protocols like SSH, FTP, RDP, and Telnet, identifying weak or default credentials.
- **How it Works:** Hydra iterates through combinations from user-provided wordlists, attempting authentication on target services. Successful login attempts are logged, highlighting insecure accounts.

SEARCHSPLOIT:

- **Purpose:** Searchsploit provides access to an offline exploit database, mapping known vulnerabilities (CVEs) to available exploits.
- **Usage in the Script:** During Full Scans, the script uses Searchsploit to find exploits corresponding to vulnerabilities identified by Nmap.
- **How it Works:** Searchsploit searches its local repository of exploits for matches with identified CVEs, providing detailed information about each exploit for further action.

XSLTPROC:

- **Purpose:** Xsltproc transforms XML data into readable formats using XSLT stylesheets, making raw scan results more accessible.
- **Usage in the Script:** The script employs Xsltproc to convert Nmap's XML output into user-friendly HTML reports.
- **How it Works:** Xsltproc processes XML files with an appropriate stylesheet, creating structured and visually appealing HTML documents for review and sharing.

FIGLET AND LOLCAT:

- **Purpose:** Figlet generates ASCII art text, and Lolcat adds colorful gradients, creating visually appealing text displays.
- **Usage in the Script:** The script uses Figlet and Lolcat to display an eye-catching welcome banner, enhancing the user interface.
- **How it Works:** Figlet creates text art in various fonts, while Lolcat pipes this output through a coloring function to produce a gradient effect.

ZIP:

- **Purpose:** Zip is a file compression utility used to archive and compress files for efficient storage and transfer.
- **Usage in the Script:** The script uses Zip to bundle scan results into a single compressed file, simplifying data management and sharing.
- **How it Works:** Zip aggregates specified files into a single archive, reducing file size while preserving directory structure and file integrity.

Summary of Tools and Their Roles in the Script

TOOL	PURPOSE	ROLE IN THE SCRIPT
Nmap	Network discovery and security auditing	Performs both Basic and Full Scans, identifying hosts, services, and potential vulnerabilities.
Hydra	Credential brute-forcing	Tests username-password combinations on various network services, identifying weak credentials.
Searchsploit	Exploit database	Maps identified vulnerabilities (CVEs) to available exploits (Full Scan only).
Xsltproc	XML transformation	Converts Nmap's XML output into user-friendly HTML reports.
Figlet & Lolcat	ASCII art and colorization	Enhances script aesthetics with banners and colorful output.
Zip	File compression	Compresses scan results into a single ZIP file for easy storage and transfer.

Helper Functions:

show_banner():

- Displays a visually appealing banner using figlet and lolcat to welcome the user. It sets a professional tone and informs users about the tool's purpose while creating a memorable first impression.

check_root():

- Before running the script, this function checks if the user has root (administrator) privileges. Certain tasks, like installing tools or accessing system files, require elevated permissions. If the script is run without root access, it outputs an error message and stops execution to prevent potential failures.

check_file():

- Ensures the script is executed with root privileges to prevent permission-related issues. It checks the user's effective ID (EUID) and exits gracefully with a clear error message if root access is not granted.

install_tools():

- Verifies the availability of essential tools and installs missing ones. This function automates dependency resolution, ensuring a seamless execution environment while displaying progress and success messages for each tool.

validate_network():

- Validates user input for network addresses, ensuring correct IP or CIDR format. It prevents invalid scans by enforcing proper input standards and provides error messages to guide users in correcting their input.

check_weak_passwords():

- Conducts brute force password testing using Hydra on various protocols. This function tests default or user-provided credentials on services like SSH, FTP, RDP, and Telnet, outputting results to dedicated text files for easy review.

map_vulnerabilities():

- Extracts CVEs from scan results and cross-references them with Searchsploit. It compiles vulnerability details into a comprehensive report and provides actionable insights for addressing identified vulnerabilities.

search_results():

- Provides an interactive option for users to search within generated reports. This function enhances usability by enabling keyword-based filtering, saving time when reviewing large volumes of scan data.

save_results():

- Compresses all output files into a ZIP archive for portability and storage. It simplifies data management by bundling results into a single file and ensures that all relevant files are easily accessible.

Main Script Logic:

The following steps outline its execution flow and highlight its professional approach to vulnerability scanning:

1. Initialization:

The script begins with an initialization phase that sets the foundation for subsequent operations. Key tasks include:

- **Displaying the Banner:** The show_banner function creates a professional introduction using ASCII art and colorful formatting. This visually engaging step sets a welcoming tone and provides immediate recognition of the tool's purpose.

- **Checking Root Privileges:** The check_root function ensures that the script is executed with appropriate permissions. Without root access, many scanning tools cannot function effectively. If root access is missing, the script gracefully exits, providing an informative error message.

- **Installing Necessary Tools:** The install_tools function checks for the availability of required utilities (e.g., Nmap, Hydra, Searchsploit). If any tools are missing, they are automatically installed. This ensures that users do not face disruptions later due to missing dependencies.

2. User Input:

The script gathers critical input from the user to customize the scanning process:

- **Network Address Prompt:** Users are prompted to enter the target network address in CIDR notation (e.g., 192.168.1.0/24). The validate_network function ensures that the input adheres to valid formatting, preventing errors during scanning.

- **Output Directory Creation:** The script requests a name for the output directory. This directory serves as a centralized location for storing all generated files, such as scan results and reports. If the directory does not exist, it is created automatically.

3. Scan Type Selection

The script offers two distinct scanning modes to accommodate different levels of analysis:

Basic Scan:

- Purpose: Focuses on identifying live hosts and open ports, along with basic service detection.
- Execution: Utilizes Nmap's standard options to perform a swift and efficient scan.
- Output: Generates an XML report detailing the identified services and their corresponding port states.

Full Scan:

- Purpose: Provides a comprehensive vulnerability assessment, including detailed service analysis and exploit mapping.
- Execution: Leverages Nmap's scripting engine to identify vulnerabilities and integrates Searchsploit for exploit discovery.
- Output: Produces both XML and HTML reports, along with a vulnerability mapping file containing CVE-to-exploit correlations.

The user selects the desired scan type via an intuitive prompt. This modular approach allows for flexibility depending on the user's objectives.

4. Weak Password Testing

The check_weak_passwords function identifies insecure credentials on critical network services. It achieves this by:

- **Brute Force Testing:** Using Hydra, the script tests username-password combinations against services such as SSH, FTP, RDP, and Telnet.
- **Custom Wordlist Support:** Users can supply their own wordlists for more targeted password testing.
- **Output:** Results are saved in individual text files for each protocol, providing clear documentation of any weak credentials discovered.

This step is pivotal for highlighting potential entry points that attackers might exploit.

5. Vulnerability Mapping

For Full Scans, the script performs a detailed vulnerability mapping process:

- **Extracting CVEs:** The map_vulnerabilities function parses Nmap's output to extract identified CVEs.
- **Searching for Exploits:** Searchsploit cross-references these CVEs against its local exploit database, compiling a list of relevant exploits.
- **Generating Reports:** The function outputs a consolidated report that includes CVE details, exploit availability, and actionable insights for remediation.

This feature transforms raw scan data into practical intelligence, enabling users to prioritize their security efforts.

6. Result Management

The script includes robust mechanisms for organizing and managing scan results:

- **Interactive Search:** The search_results function allows users to search within generated reports using keywords. This is particularly useful for quickly locating specific vulnerabilities or hosts.
- **Report Conversion:** Nmap's XML output is transformed into HTML using Xsltproc, making the data more accessible and visually appealing.
- **Data Archiving:** The save_results function compresses all output files into a single ZIP archive. This simplifies data sharing and ensures that all results are preserved in an organized manner.

7. Script Termination

The script concludes with a clean and professional termination:

- **Completion Message:** A summary of the completed tasks is displayed, including the location of the saved results.
- **Exit Handling:** The script ensures that all temporary files are cleaned up, leaving the system in its original state.

This final step underscores the script's attention to detail and user-centric design.

Differences Between Basic and Full Scan:

The primary distinction between the Basic and Full Scans lies in their depth and scope:

Basic Scan:

- **Scope:** Focuses on service identification and open port detection.
- **Speed:** Faster due to the limited scope of analysis.
- **Use Case:** Ideal for quick assessments or preliminary reconnaissance.

Full Scan:

- **Scope:** Includes detailed vulnerability detection and exploit mapping.
- **Speed:** Slower due to the comprehensive nature of the analysis.
- **Use Case:** Suitable for in-depth security evaluations and identifying actionable vulnerabilities.

By offering these two options, the script caters to both time-sensitive tasks and comprehensive security assessments.

File Structure:

The script employs a logical and organized file structure to manage its outputs efficiently. This structure ensures clarity and accessibility, even for large-scale scans:

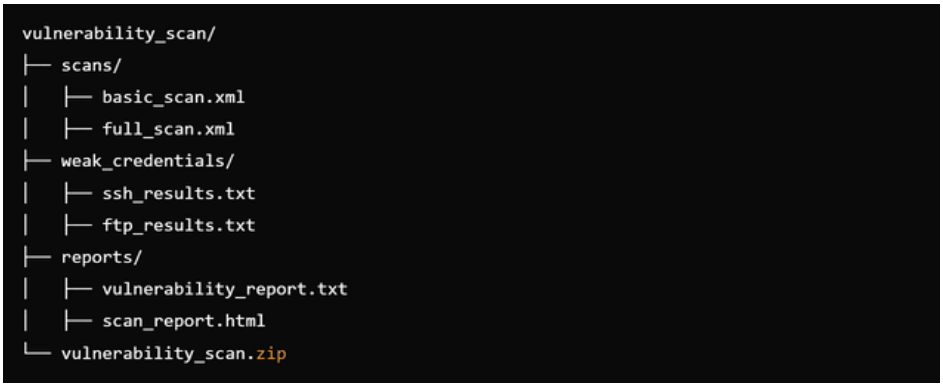
1. Base Directory:
The user specifies a name for the base directory, which serves as the root location for all generated files.

2. Subdirectories:

- Scans: Contains the Nmap results in XML format (basic_scan.xml, full_scan.xml).
- Weak Credentials: Stores results of the Hydra tests, with separate files for each protocol (ssh_results.txt, ftp_results.txt, etc.).
- Reports: Includes the vulnerability mapping report (vulnerability_report.txt) and the HTML-converted Nmap scan (scan_report.html).

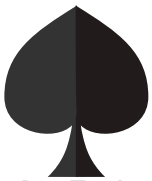
3. ZIP Archive:
The final output is compressed into a single archive named <output_directory>.zip, which bundles all generated files for convenient storage and sharing.

Example file structure:



Credits:

ChatGPT – https://chatgpt.com
Reddit - r/Kalilinux. - https://www.reddit.com/r/Kalilinux/
Github - https://github.com/
Nmap- https://nmap.org/



Questions?
Contact Me.

https://www.linkedin.com/in/eithan-saragosti/
https://github.com/Eithan200/AceTools
Eithan200@gmail.com