Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 45 No. 5
May 2024

Real-time Automated Internal Network Red-Teaming Using Nmap

Ramu Singamsetty[1] , Koneru Sai Chandra Has[2], Sravani Kukkapalli[3],
Veera Narayana Mannipudi[4], Dr. K.V.D Kiran[5], Dr. A V Praveen Krishna[6]
[1,2,3,4]Student-Koneru Lakshmaiah Education Foundation,
[5,6]Faculty-Koneru lakshmaiah Education Foundation

Abstract: Cyber Security is the concept of securing the devices and the information in the devices from malicious actors and attackers. Red-Teaming is a process in cyber security that resembles a real-time attack in which the attack will be done with the consent of the organization and with prior notice. With this technique, we can have a report of the existing vulnerabilities in the systems as well as the network infrastructure. Automated Red-Teaming is a form of automating the red-teaming process with automation of the tools and the concepts which would be useful for maintaining the security of the system. Real-time red-teaming is a process in which an automated script performs the red-teaming exercise in real time. The idea of this research is to automate a few red teaming tools to check the vulnerabilities in real-time and report if there is any vulnerability that needs attention. With this, the Vulnerability can be detected in real-time and the needful can be triggered to patch the vulnerability. Nmap is a very powerful and efficient tool when it comes to red teaming and reconnaissance, it has many switches and in-built scripts that help in customizing the attack as required to the target.

Keywords: Cyber Security, Network Security, Network, Information Security, Hacking, Data protection, Red Teaming, Nmap, Security Audit.

1. Introduction

The updating technologies day after day, are making human lives much easier and more comfortable and of course vulnerable to some point. With numerous new threats and vulnerabilities every minute, it has been very easy for a person to get access to people's data and also have a chance of misusing the same. To protect people's data from getting into the hands of such malicious actors, evolved the concept of cybersecurity. Cybersecurity helps secure people's data from entering into the hands of people with malicious intent.

Cybersecurity consists of two teams the Blue Team and the Red Team. Blue team is a team of cybersecurity professionals who specialize in patching vulnerabilities and maintaining the security of the system with access management, validating vulnerabilities etc. Red Team is a group of ethical hackers whose ultimate aim is to find vulnerabilities in the system and exploit the systems to gain complete access to the system. While red teamers and unethical hackers look similar in their way of approach the slight line that differs them is reporting. Once a vulnerability is found on the system, a detailed report of the vulnerability is documented, which consists of the name of the vulnerability, what particular asset of the organization is it discovered in the severity of the vulnerability and also the CVEs that are used to exploit the service/ system.

In this research, we are trying to develop, build and test a tool that automates the process of red teaming. The major advantage over automating the process of red teaming is to overcome 3 constraints.

2. Objectives

To automate the process of red teaming and to decrease the manual intervention requirement to the process. Doing so will decrease the following:

2.1 Workload:

The manual workload can be decreased by which we can achieve better resource utilization. This process of red teaming can be automated by various scans which gives insights into the network. Until this point the script handles and generates the data which can then be taken over y manual work for analyzing the data and to take necessary actions.

2.2 Time:

Time constraints can be overcome as the automated script works faster and more accurately compared to the manual work. The whole process which needs the usage of tools can be done automatically so that the only human intervention required is to analyze the data and to perform the actionable items.

2.3 Cost:

Once the resource utilization and time constraints are taken care of the cost automatically decreases.

3. Methods

3.1 Phases of Red-Teaming:

1. Planning and Preparation:

In this phase of red teaming, various aspects are defined such as scope, team, objective etc. Initially, the scope of the assets that are going to be under the red-teaming assessment is identified. Objective defines the goals and objectives of the red teaming assessment it has the devices, list of networks, data or sometimes may even have the processes to be evaluated. Scope determines the boundaries and limitations for the assessment specifying what needs to be tested and up to what level. Next, a team is selected which consists of people performing the assessment, there will be various people with diverse skills in performing the red-teaming assessment. Finally, the rules constraints and actions for which the team is granted permission are outlined. The red team will ensure the assessment will be bound to the defined rules.

2.Information-Gathering:

Information gathering is a phase in which we gather information about the organization. Initially, in this phase, we need to identify the assets which will be in the target scope and cluster them based on the criticality of the asset. As we are performing an internal network, it is important to know the architecture of the network to identify the devices, servers, network devices etc in the network to understand the data flow. Finally, it is important to check the internal security policies and procedures with which it would be easy to gain insights on how and what type of security procedures the organization is following so that the attack can be crafted specifically and customized.

3. Attack Simulation:

After the information-gathering phase, with the information gathered the exploit is generated and executed. Exploitation aims to exploit the identified vulnerabilities. These vulnerabilities can be in two ways, Technical and human-centric. Technical vulnerabilities address the vulnerabilities or bugs in the systems or applications of the organization are termed as the technical vulnerabilities where whereas the vulnerabilities of humans address the vulnerabilities within the humans using the devices like not adhering to the security practices. Once we can exploit the vulnerabilities in the system the next step will be to escalate the privileges. Privilege escalation is the process of elevating the privileges to “sudo” or “admin”. Through this, it is quantifiable to detect and mitigate the privilege escalations. Escalation of privilege can be a threat to as normal user who is assigned a set of tasks and that requires some access to the assets. If a normal user can execute tasks which he is not supposed to, then it is a threat. Once the escalated privileges are gained, Lateral movement needs to be identified. Lateral movement is the tactic to move across assets in the network to identify and infect additional assets in the network thus infecting the whole network until the ultimate target is breached.

4. Analysis and Detection Evasion:

In this phase, the organizations' ability to detect the attack or breach and respond to the attack is evaluated. This includes the capability of the SIEM team or the SOC team to identify mitigate and evaluate the threat. Detecting the attack is a crucial part as the time passes after the attack takes place the risk doubles and the probability of mitigation reduces. Anti-forensics techniques are a set of techniques which hackers employ to erase the identity of the hacker to misguide the forensic investigation and to remain anonymous. The incident response team of the organization needs to be assessed.

5. Reporting:

In this phase, the investigations, vulnerabilities found, exploits used and the TTP’s Techniques, Tactics and procedures are documented and handed over to the concerned people so that the vulnerable points can be patched and the human-

centric vulnerabilities can be overcome with proper training. With this phase, the red-teaming process will be completed.

In this research, we are trying to automate some processes of the red teaming through which the cost, time and cost can be reduced. We developed a bash script that can use multiple tools and will store the data accordingly in a folder which consists of various reports.

The script mostly uses tools that are open source and would be readily installed on Kali Linux by default. The following are the tools utilized:

1. Nmap
2. Nuclei
3. Dirb
4. Arp-scan

3.2 Usage of the listed tools:

1.Nmap

Nmap also called Network Mapper is an open source tool that is available with lots of switches and scripts built for network scanning and mapping. Its main purpose is to discover hosts and services on a computer network. It is important to gather as much information as possible in the process of red teaming as the more information we get about the victim network the more vulnerable it becomes. Nmap has multiple platforms support it can be used on Windows, MAC OS and Linux platforms.

2.Nuclei

Nuclei is an open-source automated vulnerability scanning tool which works on templates. Templates are pre-defined and consist of various vulnerabilities that the application might have. Template-based allows the users to define the vulnerabilities that they want to be checked. Nuclei employs a concurrent approach which makes the scanning more efficient by testing multiple templates at the same time. Nuclei is well known for its customizability; the scan parameters can be set precisely to meet the requirements of the user. Nuclei also supports multiple protocols through which SMTP, and DNS, can also be scanned.

3. Dirb:

Dirb is an open-source directory/file enumeration tool which has a list of possible directories or files which is known as a wordlist. The tool checks each of the directory names from the list and checks the status code of the response. If a directory is found to be open it recursively scans the open directory which helps in finding hidden directories or files. Users can provide custom wordlists which can match the requirements of the scan. Dirb offers a wide range of reporting and output formats which makes it easy to report.

4. Arp-Scan:

Arp-Scan is a device discovery tool that identifies and maps the devices of a network. It uses ARP requests to identify the devices through which it can identify devices without IP addresses. It initially sends an ARP request, to which the device responds with the Mac address of the device. Arp scan collects and analyses the responses from the devices of the network which consists of data like MAC address, manufacturer details and the network interface types.

3.3 Algorithm:

1. START
2. For existing RT-Data
3. Confirm if the user wants to remove existing data.
4. Create a new folder to store the files of this scan.
5. Check various network interfaces connected to the system and ask the user about what interface to use for the scan.
6. Gather the IP address of the device for the selected interface.
7. Take the c range as input from the user.
8. Run a Nmap scan with switches -sC, -sV and -A and write the information into a new file named nmap_scan.txt
9. Run an Nmap to gather the list of devices running with the critical ports open and save the report to critical.txt
10. Run an ARP-Scan to analyze the devices and attack surfaces.
11. To gather a list of devices running HTTP servers run a Nmap scan with switches –open and -p 80 and grep the output to save only the list of IP addresses to web_ip.txt
12. To test the web applications run nuclei scan on each specified IP address in web_ip.txt and save the report to a new directory named nuclei scans with file name ip_nuclei_report.txt
13. To test for hidden directories and files, perform a Dirb scan on the IP addresses from the web_ip.txt file and save the reports with the file name ip_web_dir.txt in a folder dirb_reports.

14. STOP

3.4 Proposed Bash Script:

```
#!/bin/bash
echo "The script started execution"
#deletion of existing folder
echo "Do you want to delete previous data?"
read -p "Do you want to continue? (yes/no): " answer
if [[ "$answer" =~ ^[Yy][Ee][Ss]$ ]]; then
        rm -r RTA_*
elif [[ "$answer" =~ ^[Nn][Oo]$ ]]; then
  echo"Please backup the files"
else
  echo "Invalid response. Please enter 'yes' or 'no'."
fi
#Creation of a new  folder
folder_name="RTA_$(date +\%d\%m\%Y_\%H\%M\%S)"
mkdir "$folder_name"
#creation of new  folder
echo "Folder '$folder_name' created."
echo "The previous folder & Data are deleted..!"
#Gathering Ip address:
echo "Select the network interface"
interfaces=($(ls /sys/class/net/))
for iface in "${interfaces[@]}"; do
        echo "$iface"
done
# Prompt the user to select a network interface
read -p "Enter the network interface you want to get the IP address from: " selected_interface
# Check if the selected interface is valid
if [[ ! " ${interfaces[@]} " =~ " ${selected_interface} " ]]; then
    echo "Invalid network interface. Exiting."
  exit 1
fi
# Get and display the IP address of the selected interface
ip_address=$(ip -4 addr show "$selected_interface" | grep -oP '(?<=inet\s)\d+(\.\d+){3}')
echo "IP Address of $selected_interface: $ip_address"
cd $folder_name
echo "Running Recon..."
#Recon
read -p "enter the CIDR range subnet: " cidr
nmap -sC -sV -A "$ip_address"/$cidr | cat > nmap_scan.txt
echo "network data stored to nmap_scan.txt"
#critical ports open
nmap -sC -sV -T4 -open -Pn -p 21,22,23,1521,3306,3389,5432,445,143 "$ip_address"/$cidr | cat > critical.txt
arp-scan --localnet --interface=$iface | cat > devices.txt
echo "attack surface analysed and stored to devices.txt"
#web service identification
echo "web servers are being analysed"
nmap --open -p 80 "$ip_address"/$cidr | grep -E -o "([0-9]{1,3}\.){3}[0-9]{1,3}" | awk '!seen[$0]++' | cat > web_ip.txt
#web security testing
# Specify the input file containing IP addresses
input_file="web_ip.txt"
# Specify the output directory
output_directory="nuclei_scans"
# Create the output directory if it doesn't exist
mkdir -p "$output_directory"
# Run Nuclei scan for each URL in the input file
while IFS= read -r url || [[ -n "$url" ]]; do
output_file="$output_directory/${url//[:\/]/_}_nuclei_report.txt"
  nuclei -u "$url" -o "$output_file"
  echo "Nuclei scan for $url completed. Report saved in '$output_file'."
done < "$input_file"
echo "All Nuclei scans completed. Reports saved in the '$output_directory' directory."
#DIR_ENUM
# Specify the input file containing IP addresses
input_file="web_ip.txt"
# Specify the output directory
output_directory="dirb_reports"
# Create the output directory if it doesn't exist
mkdir -p "$output_directory"
# Run dirb scan for each IP address in the input file
while IFS= read -r ip || [[ -n "$ip" ]]; do
output_file="$output_directory/${ip}_web_dir.txt"
  dirb "http://$ip" -o "$output_file"
  echo "Dirb scan for $ip completed. Report saved in '$output_file'."
done < "$input_file"
echo "All dirb scans completed. Reports are saved in the '$output_directory' directory.
```

3.5 Working of the bash script:

The script starts with the deletion of the existing red teaming data. The data refers to the zip folder that was created during the assessment that has been done before. The script wants to check if the data collected before can be deleted. If the input was yes the folders with the name starting with RTA would be deleted. Now the script creates a new folder with the name RTA_datetime. The folder name is based on the date and time, this makes it easy to find out the time when the red teaming assessment is done.

The IP address module gathers the IP address of the device concerning the interface. A single system might have multiple network interfaces which in case there is more than one network that is operating then the IP address changes with the network interface.

The recon module uses Nmap for running the initial information gathering. Using Nmap switches we can perform a scan on the whole network. The switches -sC, -sV, -A are used for the Nmap scan. The -sC switch runs a default set of the Nmap scripts from the Nmap Script Engine(NSE). The switch -sV finds the versions of the various services running on the server this is important for finding out the possible vulnerabilities of the service, especially the version of the application. -A switch other than that of the previous scans the -A switch also checks the traceroute and the OS detection which provides much more information about the target IP address.

Critical ports are a set of ports that use some of the services which handle some sensitive data which are not recommended to be exposed to other devices on the network or should be properly configured to be secure. There are a few commonly attacked critical ports which will be checked for being open and the same will be reported. The ports that are considered critical are as follows:

S No.	Port Number	Service
1.	21	File Transfer Protocol (FTP)
2.	22	Secure Shell (SSH)
3.	23	Telnet
4.	1521	Oracle Database Port
5.	3306	MySQL Database Port
6.	3389	Remote Desktop Protocol
7.	445	Server Message Block Protocol
8.	143	Internet Message Access Protocol
9.	5432	PostgreSQL Database Port

Table-1: Shows the various critical ports considered and services running on them.

For getting a complete understanding of a network it is important to know what devices are connected and the physical address of the devices can be used to gather information about the device details and the type of device. To gather this information in the recon scan we are using ARP-SCAN which is a tool used for mapping the devices of a network through the address resolution protocol (ARP). Through ARP scan we can get various details of the device connected such as the Manufacturer of the device, which can be useful to identify the device version, build name and supporting operating systems which gives many details about the target.

Web applications are one of the major services that would be running on the system. Web applications are vulnerable if not configured or developed properly. Identification of web servers in a network is required for which we are checking the IP addresses using Nmap where the port number is 80 and is also open. With this, we can get the list of IP addresses that are running a web server on port number 80. With the Nmap output, we grep the IP address of the devices mentioned in the Nmap output and we save them to a new file web.txt which consists of the IP addresses of the devices that have web servers hosted. Doing so will save us time by not performing the whole process of identification of web servers over the network.

Web application scanning is a crucial part of red teaming in which the web applications need to be analysed and vulnerabilities need to be identified. For this process, we are using the nuclei tool which works on a template basis. The web application scanning and finding the vulnerabilities or the CVE information is taken care of by nuclei. The output given by nuclei gives an oversight about what are the possible vulnerabilities that can be found in the web application. In web applications, the directories also have some important information about the configurations or the server information

which might be a good check. For this, we are using the tool Dirb (Directory Buster) which is a pre-built tool in Kali Linux. The tool brute forces a set of possible directories or files from the given wordlist. Time taken for the script is observed that the number of devices and a number of services that the network consists of have an impact on the time variation in the time taken for the same scan to be performed and analyzed. The execution has been checked within a network consisting of a various number of devices and the time is noted, the details are as follows:

Devices	Time for scan in minutes.
3	1
5	3
9	4
12	6
15	7
20	8
25	10
30	14
50	17

Table-2: Depicts the number of devices and time in minutes taken.

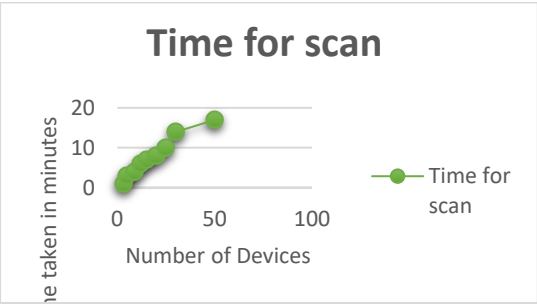

Fig-1: The graph depicts the relation between the number of devices and the time taken for the script to complete execution.

4. Results

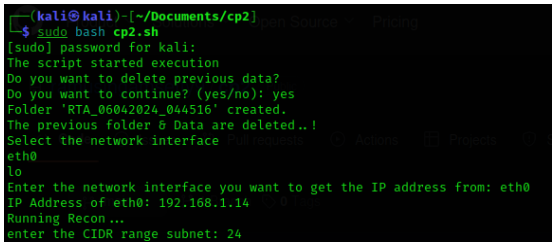

Fig-2: Fig shows the initial execution of the script where it creates the folder and gathers the IP address of the host device.

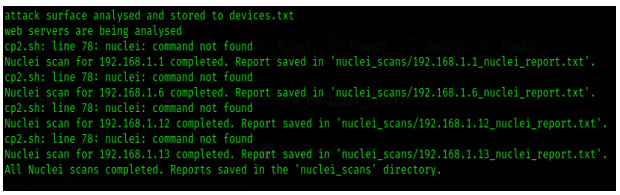

Fig-3: The script analyses the attack surface and then the web application devices are identified followed by nuclei testing and reports are generated.

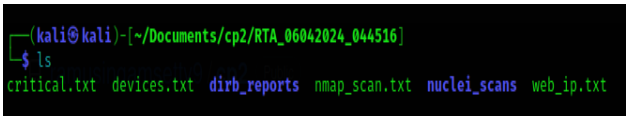

Fig-4: The output folder consisting of the reports are observed.

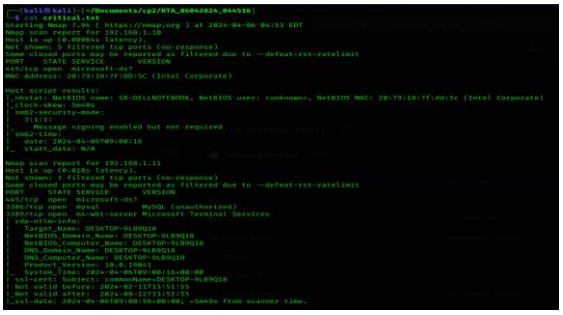

Fig-5: Critical ports that are observed with their information are stored in the critical.txt

Fig-6: The result of the Nmap scan performed all over the network is stored in the nmap_scan.txt

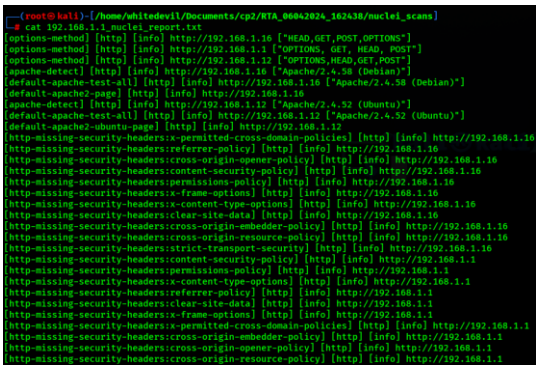

Fig-7: The fig depicts a generated Nuclei report.

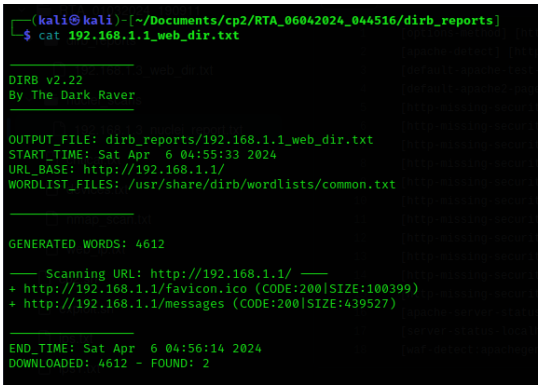

Fig-8: A Dirb report generated from the script is observed.

5. Conclusion

In this research, we developed and tested a bash script that can be useful in the process of Red-Teaming to perform the recon and vulnerability hunting automatically. The developed bash script is a collection of various tools like Nmap, Nuclei, Dirb and Arp-Scan. Using them in multiple scenarios with different switches made them perform the task as desired. The tools executed will have the report generated the generated reports are kept together in a folder that will be named with the date and time for easy identification. The output consists of various reports Nmap report, devices report, critical report, nuclei scans and drib scans. The reports are segregated and divided based on the information they contain so that the analysis of the reports is easier.

References

[1] H. T. Ray, R. Vemuri and H. R. Kantubhukta, "Toward an automated attack model for red teams," in IEEE Security & Privacy, vol. 3, no. 4, pp. 18-25, July-Aug. 2005, doi: 10.1109/MSP.2005.111.

[2] J. Rajendran, V. Jyothi and R. Karri, "Blue team red team approach to hardware trust assessment," 2011 IEEE 29th International Conference on Computer Design (ICCD), Amherst, MA, USA, 2011, pp. 285-288, doi: 10.1109/ICCD.2011.6081410.

[3] J. M. Redondo and D. Cuesta, "Towards Improving Productivity in NMap Security Audits," in Journal of Web Engineering, vol. 18, no. 7, pp. 539-577, November 2019, doi: 10.13052/jwe1540-9589.1871.

[4] Plot, Joseph & Shaffer, Alan & Singh, Gurminder. (2020). CARTT: Cyber Automated Red Team Tool. 10.24251/HICSS.2020.820.

[5] J. Mirkovic et al., "Testing a Collaborative DDoS Defense In a Red Team/Blue Team Exercise," in IEEE Transactions on Computers, vol. 57, no. 8, pp. 1098-1112, Aug. 2008, doi: 10.1109/TC.2008.42.