


TryHackMe penetration test report, Startup room

General Info and Goals

The purpose of the test is to find vulnerabilities that allow access to the machine on the given IP address, and to obtain the highest permissions in the tested environment by finding and using vulnerabilities that enable. Based on the results of the report, corrective actions are suggested.

Tools used

- nmap
- LinPeas:  <https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS>
- python3
- netcat
- ettercap
- wireshark
- revers-shell: <https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>

Process

- a) The first step was to scan the ports on the machine with the given IP using the nmap program.
- b) Login to FTP port 21 as an anonymous user.
- c) Checking possible paths with gobuster on port 80 HTTP using the most common passwords in the "common.txt" dictionary list
- d) Inserting your own file enabling enumeration and obtaining information about the machine on FTP port 21 named linPEAS.
- e) Inserting the file as an executable script in bash or python3, as in the previous step d), which allows you to create the so-called Reverse shell And access to the machine by listening with netcat.
- f) Searching for useful information in files to which we have access as a lower-level user. The "pcapng file" in the main "incidents" folder with data about traffic and connections to the machine turned out to be useful for this test. Then sending the file via netcat and its subsequent analysis using wireshark or ettercap.
- g) Logging in with the previously found login credentials on SSH port 22, then finding a way to gain higher privileges.
- h) Modification of the etc/print.sh script as the lennie user, overwriting it with your own script to run the so-called Reverse shell with the highest privileges.

Results

- a) Found open ports 21 FTP with the possibility of logging in as an anonymous user, port 22 SSH and 80 HTTP with the names of services and software.

- b) Possible login and access to files. It is also possible to add and modify your own files via FTP.
- c) Found a path named /files with access to files via FTP and the ability to execute them through a browser and HTTP port 80.
- d) Using the LinPEAS script dropped on FTP port 21 and executing it through the browser and path /files after 80 HTTP gave us some useful information like the potential user "lennie".
- e) Remote access to the machine with the ability to run a stable shell and the ability to execute your own commands, but without the highest permissions.
- f) By analyzing the file with the monitoring data of the tested machine, we could obtain information about the potential user of "lennie" and the password that was used. Thanks to this, we have the ability to log in to SSH port 22.
- g) Finding a script executable by root that cannot be overwritten by other users, but with the possibility of running it by another script with the possibility of editing it and running it by the user who is logged in at the location etc/print.sh.
- h) Uzyskanie najwyższych uprawnień ze stabilną powłoką i pełnym dostępem.

Summary and suggested corrective actions

- a) Hiding the names of services and software, if it is possible to disable SSH port 22 and FTP port 21 or disable anonymous login on FTP port 21 and hide service information on HTTP port 80.
- b) If the existence of the FTP service is necessary, it must be updated and it is necessary to disable the possibility of editing and inserting own files by an anonymous user.
- c) Changing the path address to a less obvious one.
- d) Disabling the ability to run and modify files on port 21 FTP and 80 HTTP.
- e) Earlier sub-points in this section.
- f) Change of permissions and greater protection against unauthorized access to the file monitoring traffic on the tested machine, i.e. "pcapng"
- g) Disabling the ability to execute files associated with root permissions. In this case etc/print.sh.
- h) As above in point g).