**Password CTF**

**Lab Overview**

For this lab, we will simulate a Capture the Flag (CTF) style challenge to learn about ciphers, hashes, and how to crack passwords. Winning a CTF challenge usually involve some sort of puzzle or obstacle that must overcome. Players must apply security related skills, tools, and logic to solve clues to the puzzle. These clues are known as “flags”. Whoever collects all of the flags first wins the CTF.

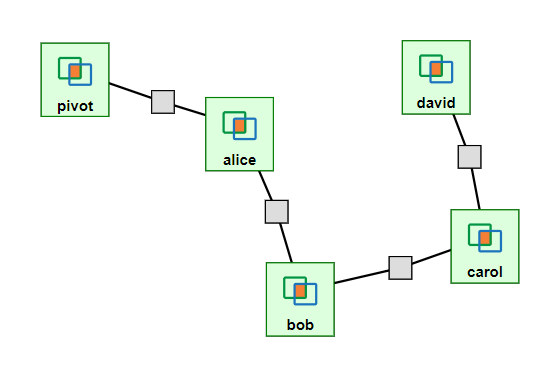
In our lab, we will play the role of a penetration tester or assumed attacker. We gain access to a user machine on some unknown network. Our objective is to gain access to every machine on the network. Each machine contains a flag which can be used to gain access to the next higher level machine. The flags we need to find are either plaintext or encrypted passwords hidden somewhere in the file system. Let's get started.

**Before you start, you will need to complete with the following setups (If you have complete 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 (VM) to each machine in the Lab (see Setup-Lab)

**RSpec File Location:**

<https://raw.githubusercontent.com/DrVoyager/EdGENI/master/Rspec-Files/PasswordCTF-Rspec.txt>

**Network Topology**

The network for this lab will consist of 5 machines connected together in a daisy-chain format. Each machine is owned by a different user. The machine named pivot will represent the machine initial access has been achieved. As an attacker, we will “pivot” from machine to machine, collecting passwords as we progress towards David’s machine.

**Useful Tools**

* [Rumkin](http://rumkin.com/tools/cipher/) Cipher Tools
* [CrackStation](https://crackstation.net/) Hash Cracker
* [JohnTheRipper](https://www.openwall.com/john/) Password Cracker

**Task 1 - Find Alice’s password**

After every machine has been configured, create a new SSH connection to the Pivot machine.

The Pivot machine represents a low privileged user’s machine accessed by means of phishing, social engineering, or unchanged default credentials.

Type **su pivot** and provide the password which happens to be 3 periods ( **…** )

The user and hostname of the terminal prompt should now change to **pivot@pivot:~$**

Look around in the files on the Pivot machine. Located somewhere on the Pivot machine’s file system is Alice's file that contains her password. (Hint: the user on the pivot is very careless so that he just put the password in a file called password.txt) **Screenshot Alice’s password and include in documentation.**

**Task 2 - Find Bob’s password**

After finding Alice’s password, we need to connect to Alice’s machine. On machine pivot, type **ssh alice@alice** and provide the password found in Task 1. The terminal prompt should be changed to **$.** Type **bash**, the terminal prompt will be changed to **alice@alice:~$**

Look around the file system on Alice’s machine. Bob’s password is hidden somewhere. When you find it, you will notice it is encrypted in caesar cipher. There are many tools on the internet that could be used to break it. Some were mentioned right before Task 1.

Solve the cipher to find Bob’s password. **Screenshot Bob’s password and include in documentation.**

**Task 3 - Find Carol’s Password**

After finding Bob’s password, we need to connect to Bob’s machine. Establish an SSH connection, on Alice’s machine, type: **ssh bob@bob** and provide the password found in task 2. Type in **bash**, the terminal prompt should change to **bob@bob:~$**

Look around the file system on Bob’s machine. There are a lot more files to look through to find Carol’s password. This can be found much faster if you use the **ls**, **find**, or **grep** commands.

**( ls -al \*** is your friend)

When you find it, you will notice Carol’s password has been encrypted into a hash. There are many tools on the internet that could be used to break Carol’s hash. Some were mentioned right before Task 1. (Hint: you can tell the type of hash based on the number of bits)

Decrypt the hash to find Carol’s password. **Screenshot Carol’s password and include in documentation.**

**Task 4 - Find David’s Password**

After finding Carol’s password, we need to connect to Carol’s machine. Establish an SSH connection, on Bob’s machine, type: **ssh carol@carol** and provide the password found in task 3. Type in **bash**, the terminal prompt should change again to **carol@carol:~$**

Look around the file system on Carol’s machine. There are many password files to look through to find David’s password. This can be found much faster if you use the **ls**, **find**, or **grep** commands. **( ls -al \*** is your friend).

When you find the correct password file. Use the cat command to display its contents:

**cat ~/path/to/file/<nameOfFIle.txt>**

**Screenshot the cat command and its output, include this in documentation.**

You will notice, like Carol, David’s password has been encrypted into a hash. There are many tools on the internet that could be used to break David’s hash. We will use JohnTheRipper to crack David’s hash to reveal David’s password.

**\*\*\*NOTE:** Cracking passwords can be a lot of work. This workload can consume many system resources and potentially take long amounts of time. To minimize problems with system resources, we will copy the hash value to our host system (ubuntu-vpark) and use our own system resources to crack David’s password.

**Task 5 - Crack David’s Password**

We now should have the hash of David’s password in the filesystem of out host machine (ubuntu-vpark). Make sure you have the ***Jumbo*** *version of JohnTheRipper* downloaded on your system. To test this, simply type john-the-ripper at the terminal of your host machine. If you do not have Jumbo version of JohnTheRipper, install it with this command:

**sudo snap install john-the-ripper**

To crack the hash, run this command:

**john-the-ripper --format=raw-sha1 ~/path/to/file/<hashfile.txt>**

for the John will start the process of cracking the hash.

**NOTE**: you need to create the hashfile.txt based on the hash you got from the last task. Search on the Internet to construct the hashfile for the correct format that matches option “raw-sha1”. You may need to avoid having special unseen characters, such as “space” and “new line” in your hashfile.txt.

This could take a variable amount time, depending on the size of the password, the randomness of characters used, and the machine resources available. For this lab, we have intentionally made David’s password less complex in hopes of minimizing this variable of time. In our attempts, this hash will take about 15-30 minutes to crack. When complete, **Screenshot the output of JohnTheRipper and include in documentation.**

Now that we have David’s password, there is one final flag to find on David’s Machine.

Establish an SSH connection from Carol’s Machine to David’s machine, Type: **ssh david@david** and provide the password found in JohnTheRipper. The terminal prompt should change again to **david@david:~$**

Look around the file system on David’s machine. There should be a file that contains the final flag. This can be found much faster if you use the **ls**, **find**, or **grep** commands. **( ls -al \*** is your friend). Use the cat command to output the contents of the final flag.

**cat ~/path/to/file/<nameOfFIle.txt>**

**Screenshot the cat command and its output, include this in documentation.**

Congratulations you have completed the Password CTF challenge!