

**Cybersecurity Capture the Flag - Verify**

**Lab Overview & Description**

People keep trying to trick my players with imitation flags. I want to make sure they get the real thing! I'm going to provide the SHA-256 hash and a decrypt script to help you know that my flags are legitimate.

Additional details will be available after launching your challenge instance.

**Hints**

* Checksums let you tell if a file is complete and from the original distributor. If the hash doesn't match, it's a different file.
* You can create a SHA checksum of a file with sha256sum <file> or all files in a directory with sha256sum <directory>/\*
* Remember you can pipe the output of one command to another with |

**Lab Instructions**

**Step 1.**

1. Open the Linux terminal and SSH into the Pico host using the following command:

ssh -p 51095 ctf-player@rhea.picoctf.net

2. Enter the password f3b61b38 when prompted.

1. Type yes to accept the server's fingerprint.
2. Use the command ls to list the contents of the current directory.

you should be displayed with something that looks like this:   
A screenshot of a computer

AI-generated content may be incorrect.

**Step 2.**

Use the command cat checksum.txt to display the checksum contained inside the checksum.txt file.  
A screenshot of a computer

AI-generated content may be incorrect.

**Step 3.**

To create the SHA Checksum of a file you have a few different options:  
  
Option 1: To create the checksum of a single file:  
1. Use command cd files to go into the files directory.  
2. Use the ls command to list all the files contained in that directory.  
3. Use the command sha256sum <file> to display the checksum for that specific file.  
  
example: sha256sum 02kLdPvr  
A screenshot of a computer

AI-generated content may be incorrect.  
  
Option 2: This will create SHA Checksums for all files in a directory:  
1. Use command sha256sum >directory>/\* to create and display the checksum for all files contained within the directory.

example: sha256sum files/\*  
A screen shot of a computer screen

AI-generated content may be incorrect.

**Step 4.**

Now you want to decrypt the checksum to see if it matches the provided Checksum from the challenge.  
1. Use command ./decrypt.sh files/<file> to verify the checksum.  
  
example: ./decrypt.sh files/02kLdPvr  
A screenshot of a computer

AI-generated content may be incorrect.  
in this case we got the response “bad magic number – Failed to decrypt” because the flag was a fake so this is not the correct file.  
  
Now you could do that for each file individually, which would take some time.  
To stream line the process you can use pipe the sha256sum command into the grep command.  
  
To do this use command sha256sum files/\* | grep "<checksum>"  
  
For this challenge you would replace <checksum> with the checksum provided in the challenge description.  
  
example: sha256sum files/\* | grep "fba9f49bf22aa7188a155768ab0dfdc1f9b86c47976cd0f7c9003af2e20598f7"  
  
This command will run the sha256sum command on each file in the 'files' directory to generate a checksum. It will then pipe those results into the grep command, which will compare the checksums against the one entered in the quotation marks and print the file that matches.  
A computer screen with white text

AI-generated content may be incorrect.

**Step 5.**

The final step is to decrypt the correct file from Step 4 using ./decrypt.sh files/<file>  
In the case of this challenge we would use command:

./decrypt.sh files/87590c24  
The result that prints is the CTF flag:  
picoCTF{trust\_but\_verify\_87590c24}  
**A computer screen with white text

AI-generated content may be incorrect.**

Copy the CTF flag and paste it into the Pico CTF challenge page and click submit and your challenge is completed.

Bonus - Scripting:  
You can use a script to automate the entire process for you:  
  
This is a script that can be pasted into the terminal for this challenge:  
  
# Define the expected checksum and target directory

expected\_checksum="fba9f49bf22aa7188a155768ab0dfdc1f9b86c47976cd0f7c9003af2e20598f7"

directory="files"

# Loop through all files in the "files" directory

for file in $directory/\*; do

# Calculate the SHA-256 checksum of the file

checksum=$(sha256sum "$file" | awk '{print $1}')

# Check if the checksum matches the expected checksum

if [ "$checksum" == "$expected\_checksum" ]; then

echo "Checksum matches for $file. Decrypting..."

./decrypt.sh "$file"

else

echo "Checksum does not match for $file. Skipping decryption."

fi

done  
  
This script will verify the checksum of each file against the expected checksum provided in the challenge description. If the checksum does not match, the script will skip decryption until it finds a match. Once a match is found, it will decrypt the file and print the PicoCTF Flag.  
A screenshot of a computer

AI-generated content may be incorrect.  
  
 **\*Verify\* End**