Snatchr - CTF Documentation

Montag, 5. August 2024 10:12

Passwords look like this: <pwgen 20 1>

Bash Color Code

```
PS1='\[\033[1;32m\]\u\[\033[1;35m\]@\[\033[0;32m\]\h:\[\033[1;36m\]\w\[\033[1;35m\]\$\[\033[0m\] '
```

Hint Color: "\033[1;35m<text>\033[0m"

MOTD:

"

"

Import the image: docker load -i <file>.tar
To start the game: docker run -td -p <desired_port>:22 --name <name?> <snatchr_base_image>
SSH: ssh <user>@localhost -p <desired_port>

For maintenance: user: root / pw: eiphePhainauBoot3reesh8Zoopeequ8ahdaeroS

Levels:

1. **SSH**

   Get the password by just logging into the user and read the MOD

   MOTD:

   "\n\033[1;35mAwesome\033[0m, you managed to connect to the starting point of the CTF. From here on your mission will be to snatch as many passwords as possible. On every level, you will find a password that gives you access to the next lev>A similar message will pop up on every level, telling you what your task will be. Additionally, \033[1;35myou can type the command "task" to print this message again\033[0m.
   \033[1;35mIf you are stuck, use the command "hint" to get some help\033[0m.

   By the way, here is your password for the next level: \033[1;35mahD6aeloop4cila7nuGo\033[0m

   \033[1;35mHave fun!\033[0m\n"

   "Useful commands: \033[1;35mssh, task, hint\033[0m\n"

   Hint:
   "\nRead the MOTD. If you don't see one, use \033[1;35m\"task\"\033[0m\n"

2. **LS / CAT**
   Use "ls" to list a file that has the password as its name
   Use "cat" to print the content of the files (only one has the password)

   MOTD:
   "\nFor most of the levels, there will be \033[1;35museful commands\033[0m provided, that will help you tackle the task. At any point, you can \033[1;35mget information about what the command does \033[0m by appending "--help" (\033[1;35m<command> --help\033[0m) or prepending "man" (\033[1;35mman <command>\033[0m) for a detailed description.

   Now try to list the files in your current folder and find the password in one of the files.\n"

   "Useful commands: \033[1;35mman, ls, cat, task, hint\033[0m\n"

   Hint:
   "\nUse \033[1;35m\"ls\"\033[0m to print the files inside a folder and \033[1;35m\"cat\"\033[0m to view the content of a file\n"

3. **Subfolder**
   Go through 2 - 3 subfolders with either "cd" or "ls" to find the file

   MOTD:
   "\nThere are now multiple folders that could contain a password file. Go through them and find it.\n"

   "Useful commands: \033[1;35mcd, ls, cat, task, hint\033[0m\n"

   Hint:
   "\nEither use \033[1;35m\"cd\"\033[0m to change into a directory or use \033[1;35m\"ls\" like this: \"ls <subfolder>\"\033[0m\n"

4. **Hidden Files**
   Use "ls -a" to print hidden files

   MOTD:
   "\nHmm... No files to be seen. Maybe something is \033[1;35mhidden\033[0m here...\n"

   "Useful commands: \033[1;35mls, cat, task, hint\033[0m\n"

   Hint:
   "\n\033[1;35m\"ls\"\033[0m has an option to display more than just the visible files. Find this option with \033[1;35m\"ls --help\"\033[0m.\n"

5. **Find a file**
   Use "find" to find a file in a subfolder forest of many irrelevant files

MOTD:
"\nNow there are way too many… Find the file named \033[1;35m\"password\"\033[0m
If only there was a tool that could help us \033[1;35mfind\033[0m it…\n"

"Useful commands: \033[1;35mls, cat, find, task, hint\033[0m\n"

Hint:
"\n\033[1;35m\"find\"\033[0m can help you. Try it with this syntax \033[1;35m\"find ./ -name \"password\"\"\033[0m\n"

6. **Find with user : group attributes**

MOTD:
"\nEvery file has specific attributes. To find the next file you unfortunately can't go by name. Here you have to find it by the \033[1;35mfiles user and group attribute\033[0m, specifically for \033[1;35muser: \"snatchr_14\" and group: \"snatchr_7\"\033[0m.\n"

"Useful commands: \033[1;35mls, cat, find, task, hint\033[0m\n"

Hint:
"\n\033[1;35m\"find ./ -user \"<user>\" -group \"<group>\"\"\033[0m\n"

7. **Find with size property and error output redirection**

MOTD:
"\nNow try it with another attribute: the \033[1;35mfile size\033[0m. Look for a file that is \033[1;35mexactly 21 bytes\033[0m in size.
You may notice lots of strange output when running the find command. There is a way to \033[1;35mget rid of the error messages (error/output redirection)\033[0m.\n"

"Useful commands: \033[1;35mls, cat, find, task, hint\033[0m\n"

Hint:
"\nYou want to \033[1;35mredirect the error output\033[0m to the \"trash\". Add this at the end: \033[1;35m\"2>/dev/null\"\033[0m\n"

8. **GREP**

Use "grep" to find the line where the password is written

MOTD:
"\nSomwhere in the \"passwords\" file must be the right password. I can only remember the \033[1;35mfirst few letters\033[0m. It was something like \033[1;35m\"aiPhoo\"\033[0m …\n"

"Useful commands: \033[1;35mls, cat, grep, task, hint\033[0m\n"

Hint:
"\nWith grep you can filter output like this: \033[1;35m\"<output> | grep <filter>\"\033[0m.\n"

9. **UNIQ / SORT**

Use "sort" and "uniq" to find the entry that is the only one that is unique

MOTD:
"\nIn all these files only \033[1;35mone string occurs more than one time\033[0m. Find it, it will be the

password for the next level.\n"

"Useful commands: \033[1;35mls, grep, uniq, sort, task, hint\033[0m\n"

Hint:
"\nWith cat you can \033[1;35m print multiple files\033[0m and redirect the output. Try to look into the \033[1;35museful commands\033[0m to organize the output.\n"

10. BASE64

Password is easily found, but encrypted. Use "base64 -d" to get the password in plain

MOTD:
"\nHmm… This password doesn't work. Maybe \033[1;35mit is encrypted\033[0m in a way?\n"

"Useful commands: \033[1;35mcat, base64, task, hint\033[0m\n"

Hint:
"\nThe password is \033[1;35mbase64 encoded\033[0m\n"

11. ROT13 / CAESAR //give the tr string

Use "tr" and a pipe to redirect output from "cat" to translate a string using a predefined alphabet '[a-zA-Z]' '[n-za-mN-ZA-M]'

MOTD:
"\nEncrypted! AGAIN! Anyway… This one looks like \033[1;35mROT13\033[0m\n"

"Useful commands: \033[1;35mcat, tr, task, hint\033[0m\n"

Hint:
"\nROT13 uses an \033[1;35msubstitution cypher\033[0m. That means the alphabet is shifted and used as the cipher. \033[1;35m\033[1;35mWith tr you can try to substitute it\033[0m\033[0m.\nUse this as the option for tr: \"\033[1;35m'A-Za-z' 'N-ZA-Mn-za-m'\033[0m\""

12. SHA1

Use "john" to brute-force the content behind the hash

MOTD:
"\nThis password looks like it has been \033[1;35mhashed\033[0m. This won't be as easy as decrypting base64, you need to \033[1;35mcrack it\033[0m. For this you can use \033[1;35mjohn\033[0m the ripper. I have provided a password list, that you can use to brute-force the hash\n"

"Useful commands: \033[1;35mcat, john, task, hint\033[0m\n"

Hint:
"\nFirst you have to crack it by \033[1;35m\"john --wordlist=<wordlist> hash.txt\"\033[0m and then \033[1;35m\"john --show hash.txt\"\033[0m\n"

13. SSH - Keyfile / SCP

Log into the next user with a ssh keyfile

MOTD:
"\nTo reach the next level, you have to \033[1;35muse the provided keyfile\033[0m to log into the next user. This time, use ssh from the user you are currently in and use port 22 for connection.

\033[1;35mNo password is needed here\033[0m. Tho if you manage to reach level 13, you can still print the password with \033[1;35m\"cat /etc/sntchr_pw/snatchr_13\"\033[0m\n"

"Useful commands: \033[1;35mssh, task, hint\033[0m\n"

Hint:
"\nssh has an option for using keyfiles instead of passwords for authentication. \033[1;35mCheck the manpages\033[0m\n"

14. STRINGS

Use "strings" to get human-readable information from a executable

MOTD:
"\nThe next password is somewhere \033[1;35minside the compiled binary \"run\"\033[0m. Using cat will yield no real information. There is another command you can use to find \033[1;35mstrings \033[0m inside of binaries.\n"

"Useful commands: \033[1;35mls, strings, task, hint\033[0m\n"

Hint:
"\nUse the \033[1;35m\"strings\"\033[0m command\n"

15. NETCAT *

Use "nc" to listen to a port and receive a password

MOTD:
"\nOn this machine there is an \033[1;35mopen connection on localhost port 3000\033[0m. Connect to it using \033[1;35mnetcat\033[0m and receive the password.\n"

"Useful commands: \033[1;35mnc, task, hint\033[0m\n"

Hint:
"\nUse netcat as follows:\033[1;35m \"nc <address/localhost> <port>\"\033[0m\n"

16. TAR / GZIP

Decompress an archive using "tar" and "gzip" (and file)

MOTD:
"\nEver heard of matryoshka dolls? The following \033[1;35marchive file is compressed multiple times \033[0m. Use\033[1;35m \"tar\" and \"gzip\"\033[0m to unpack the innermost file to gain the password for the next level. In some cases simply using the commands won't work. Maybe there is \033[1;35msomething wrong with the extension\033[0m..?\n"

"Useful commands: \033[1;35mtar, gzip, mv, file, ls, task, hint\033[0m\n"

Hint:
"\nYou can identify filetypes with the command \033[1;35m\"file <file>\"\033[0m. Add the appropriate extension for gzip and tar to work properly.\n"

17. SUID

Execute commands as another user

MOTD:

"\nThere is an option you can give files and that is the \033[1;35mSUID-bit\033[0m. It allows every user to \033[1;35mopen/execute the file with the permission of the file owner\033[0m. In this case snatchr_17 left a script, that can open files. Think about it, \033[1;35mwhich file will give you the password\033[0m, now that you are basically snatchr_17.\n"

"Useful commands: \033[1;35mls, task, hint\033[0m\n"

Hint:
"\nThe file you are looking for is \033[1;35m\"/etc/snatchr_pw/snatchr_17\"\033[0m\n"

18. **CRON** *

Find out what a cronjob does, and find the corresponding script, that gives you the password

MOTD:
"\nA cronjob runs with a set time interval. You can get a list of the system-wide cronjobs by listing the directory \033[1;35m\"/etc/cron.d\"\033[0m. \033[1;35mWhat file does the snatcher_18's conjob run \033[0m. Find it, it will provide you with the password\n"

"Useful commands: \033[1;35mls, cat, task, hint\033[0m\n"

Hint:
"\nRead the file by typing \033[1;35m\"cat /etc/cron.d/snatchr_18.cron\"\033[0m\n"

19. **CURL** *

Get the content of a website with "curl"

MOTD:
"\nYou can visit websites from within your terminal with the command \033[1;35m\"curl\"\033[0m, but you will only get the raw html file output. Visit the website \033[1;35m\"mysite.com\"\033[0m and look through it, maybe you will be able to find the password.\n"

"Useful commands: \033[1;35mcurl, task, hint\033[0m\n"

Hint:
"\nAs mentioned on the website, \033[1;35myou'll find the password in the response header\033[0m. Curl has options to display these headers.\n"

20. **Binary Exploitation //difficult by choice**

The binary is vulnerable to format string attacks

MOTD:
"\n\033[1;35mYou basically made it! Congratulations!!\033[0m\nIf you still haven't had enough, this optional last level will probably give you the rest. The task was deliberately made \033[1;35mmuch more difficult\033[0m, without the expectation that everyone could solve it.\nAnyways, the only thing you will get are the keywords \033[1;35m\"Format String Vulnerability\"\033[0m.\nGood Luck!\n"

"Useful commands: \033[1;35mtask, hint\033[0m\n"

Hint:
"\n\033[1;35mp p p p p p p p p p p p p p p p p p p p p p p\033[0m\n"

21. **CONGRATULATIONS**

MOTD:

```
"\n\033[1;35mCONGRATULATIONS!!!\033[0m You did it!!!\nI hope you liked this small CTF. If you want
to continue with this type of challenges, I can strongly recommend \033[1;35m\"PicoCTF\"\033[0m or
\033[1;35m\"OverTheWire - CTF\"\033[0m\n"

"Here is your prize:\n"

"\n
     _________________
    |@@@@|     |####|
    |@@@@|     |####|
    |@@@@|     |####|
    \@@@@|     |####/
     \@@@|     |###/
      \`@@|_____|##'
          (O)
        .-'''''-.
      .'  * * *  \`.
     :  *       *  :
    : ~  W E L L  ~ :
    : ~  D O N E  ~ :
     :  *       *  :
      \`.  * * *  .'
        \`-.....-'

\n"
```

* level is globally accessible (additional permission management)

List of commands:

- **ssh**
  Ssh (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine.
- **exit**
  Closes the current shell or ssh connection.
- **task**
  Prints the task of the current level.
- **hint**
  Gives a hint for the current level.
- **man**
  Provides an in-depth description of a command and its usage.
- **ls**
  List information about the FILEs (the current directory by default).
- **cat**
  Prints the content of a file to the terminal output.
- **cd**
  With this command you can change directories.

- **find**
  Searches the provided directory (and child directories) for matching parameters (size, owner, name, etc.).
- **grep**
  Searches for a pattern in a file or from the terminal output.
- **uniq**
  Filters adjacent matching lines from files or the terminal input. (Filters out multiple occurrences)
- **sort**
  Sort terminal output/files (alphabetically, numerically, etc.).
- **base64**
  Encode/decode strings/files with base64.
- **tr**
  Translate a string from terminal input/files with a given string. (E.g. use a custom alphabet)
- **john**
  Crack hashes using a brute force-attack.
- **strings**
  Search for printable strings inside a binary.
- **nc**
  Network tool to establish connections on a lower layer.
- **tar**
  Create archive from files/folders or unpack archives to receive files.
- **gzip**
  Compress/uncompress files/folders.
- **curl**
  Network tool for transferring data from/to servers using URLs.