Leaky Faucet Server Build Guide v1.3.0

Operating System Requirements

This has been tested and implemented using Ubuntu 22.04.

Networking Requirements

The server requires a static public IP address since your A record will need to point here. You can technically use a dynamic address but this will require a lot of DNS record updating and you will encounter ‘outages’ as you wait for the updates to propagate.

This server is currently built for use with IPv4.

Domain Name Requirements

You need to register a domain name. Make sure you choose a service that allows you to set both A and NS record types.

In this example we will assume you have registered sampledomain.xyz, and we are going to use a subdomain for our DNS tunnel server. Ie. sub.sampledomain.xyz. Set up the following records:

1. Type = NS Name = sub Content = sampledomain.xyz
2. Type = A Name = ns1 Content = <server public ip>

It may take up to 24 hours for this to propagate across the internet. Be patient.

Package Requirements

The server is a collection of python and bash scripts. Thus, you will need to install python and few additional packages:

1. sudo apt update
2. sudo apt install python3
3. sudo apt install python3-pip
4. sudo pip3 install dnslib
5. pip3 install twilio

To send SMS messages, you will need to integrate with a service. These scripts are designed to work with Twilio. There are some build requirements required for the API calls. You will need to follow the instructions here: https://www.twilio.com/docs/sms/quickstart/python . (For now, ignore

the Twilio instruction regarding environmental variables. This will not work for this server and a better method is covered in the next section.

Environmental Requirements

There are a few mandatory configuration changes, and a few optional. Let's start with mandatory:

Twilio API Settings

1. **Twilio API Keys** - Your account ID, auth key, and phone number are not hardcoded into the SMS sender script. Instead, it looks for environmental variables. The Twilio documentation tells you to edit your bash profile but this will not work if you intend to have a cron job restart the processes periodically. Nor will it work if a watchdog script has to restart the services in case of a failure. Instead you will need to do one of the following two things:
    a. Specify your twilio SID, authkey, and phone number in the actual crontab. This way the scripts can read them as environmental variables. You will need to specify: TWILIO_ACCOUNT_SID, TWILIO_AUTH_TOKEN, and TWILIO_PHONE_SENDER.
    b. The other option is to set these environmental variables in /etc/environment. Note: This will set these variables for all users. I've tried using a .sh file /etc/profile and cron does not pick it up. Add the following lines (without <>):
        i. export TWILIO_ACCOUNT_SID=<SID>
        ii. export TWILIO_AUTH_TOKEN=<AUTHTOKEN>
        iii. export TWILIO_PHONE_SENDER=<+PHONENUMBER>

You will also need to place your phone number in /home/ubuntu/lf_phonelog/lf_phonewhitelist. The lf_sms.py script will not send you a sms if you number is not present in that file. If that file doesn't exist, just create it.

Domain Listener Setting

You will need to specify your listening domain in the lf_listener_1.0.0.py script. To do so, open it in your editor of choice modify this line:

```
m = re.search(r'\;(\S+)\.sampledomain\.xyz', str(msg), re.MULTILINE)
```

Replace `.sampledomain` and `.xyz` with your domain. If you are using a subdomain, you will need to add an additional `\.subdomain` in front of `\.sampledomain`. Here are two examples assuming jar.com and cookie.jar.com:

1. `m = re.search(r'\;(\S+)\.jar\.com', str(msg), re.MULTILINE)`
2. `m = re.search(r'\;(\S+)\.cookie\.jar\.com', str(msg), re.MULTILINE)`

Optional: Path References

I've hardcoded some path values into the scripts. For example, the scripts expect to sit in ~/leakyfaucet/. They do not need to live here if you'd rather put them in a folder or are running from a different user. You will need to adjust your soft links and paths specified in each script.

CRONTAB Requirements

Since there is nothing stopping someone from sending real confidential data to your server, you will want to keep your hands clean and wipe all received data hourly. To do this, you will need to set up a cron job to run lf_run.sh every hour.

Command: crontab -e (select and editor if you haven't already) and enter the following line into the file: 0 */1 * * * /home/ubuntu/leakyfaucet/Server//lf_run.sh

lf_run.sh will wipe the logged data and also restart your listener and filemon services.

Step-by-Step Vanilla Setup

If you don't really care to know what each command does, you can blindly type in each of these commands in order and your server will be ready. (I'll eventually build a prep script to do all of this.)

Assumptions

- You are using Ubuntu and plan to place everything in the ubuntu home folder.
- You've already configured your dns records.
- You are running all of these commands from /home/ubuntu/
- You are running this on a freshly installed server. If you are using an old system full of all kinds of config changes, your results may vary.

Blindly Follow

1. Change to home directory: `cd ~`
2. Clone git repo: `git clone https://github.com/mrwaterhouse/leakyfaucet`
3. By pass sudo passwd req for ubuntu user:
   `sudo visudo`
   Below the line that says %sudo ALL=(ALL) ALL add:
   `ubuntu ALL=(ALL) NOPASSWD: /usr/bin/python3, /usr/local/bin/lf_listener.py`
4. Update APT: `sudo apt update`
5. Install Python: `sudo apt install python3`
6. Install PIP: `sudo apt install python3-pip -y`
7. Install DNSLib: `sudo pip3 install dnslib`
8. Install Twilio: `pip3 install twilio`
9. Create sym link for listener:
   `sudo ln -s ~/leakyfaucet/Server/lf_listener.py /usr/local/bin/`
10. Create sym link for filemon:
    `sudo ln -s ~/leakyfaucet/Server/lf_filemon.py /usr/local/bin/`
11. Create sym link for sms sender:
    `sudo ln -s ~/leakyfaucet/Server/lf_sms.py /usr/local/bin/`
12. Create sym link for run script:
    `sudo ln -s ~/leakyfaucet/Server/lf_run.sh /usr/local/bin/`
13. Add your domain to listener script: `nano ~/lf_listener.py` (Edit line 41 accordingly)
14. Check if port 53 is already in use: `sudo lsof -i :53` (If something is present, you need to disable it. Likely:)
    - `sudo nano /etc/systemd/resolved.conf`
    - (Set DNSStubListener=no)
15. Restart System Resolved Service: `sudo systemctl restart systemd-resolved`
16. Create log folder: `mkdir lflog`

17. Create data folder: `mkdir lfdata`
18. Create auth folder: `mkdir lfauth`
19. Create commands folder: `mkdir lfcommands`
20. Create lf_data file: `touch ~/lfdata/lf_data.txt`
21. Create lf_exfil folder: `mkdir ~/lfdata/lf_exfil`
22. Make phone log folder: `mkdir ~/lflog/lf_phone.log`
23. Create sms whitelist file: `touch ~/lfauth/lf_phone.auth`
24. Put your phone number in whitelist:
    `echo "yourPhoneNumber" >> ~/lf_auth/lf_phone.auth`
25. Create sms log file: `touch ~/lflog/lf_smsSent.log`
26. Create unauthorized attempt log: `touch ~/lflog/lf_unauthAttempts.log`
27. Set twilio environmental variables: `sudo nano /etc/environment`
    (Add the following)

```
export TWILIO_ACCOUNT_SID=YourSID
export TWILIO_AUTH_TOKEN=YourToken
export TWILIO_PHONE_SENDER=+YourTwilioNumber
```

28. Create systemd listener service:
    `sudo nano /etc/systemd/system/lf_listener.service` (Enter the following)

```
[Unit]
Description=LF Listener Service
After=network.target
[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/
ExecStart=sudo /usr/bin/python3 /usr/local/bin/lf_listener.py
Restart=always
RestartSec=10
StandardOutput=syslog
StandardError=syslog
[Install]
WantedBy=multi-user.target
```

29. Create filemon listener service:
    `sudo nano /etc/systemd/system/lf_filemon.service` (Enter the following)

```
[Unit]
Description=LF Filemon Service
After=network.target
[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/
Environment="TWILIO_ACCOUNT_SID=YourSID"
Environment="TWILIO_AUTH_TOKEN=YourToken"
Environment="TWILIO_PHONE_SENDER=+YourTwilioNumber"
ExecStart=/usr/bin/python3 /usr/local/bin/lf_filemon.py
Restart=always
```

```
RestartSec=10
StandardOutput=syslog
StandardError=syslog
[Install]
WantedBy=multi-user.target
```

30. Make listener service executable:
    `sudo chmod +x /etc/systemd/system/lf_listener.service`
31. Make filemon service executable:
    `sudo chmod +x /etc/systemd/system/lf_filemon.service`
32. Reload systemd: `sudo systemctl daemon-reload`
33. Enable listener service: `sudo systemctl enable lf_listener`
34. Enable filemon service: `sudo systemctl enable lf_filemon`
35. Create cron job for hourly wiping: `crontab -e` (Enter the following)
    `0 */2 * * * /home/ubuntu/lf_run.sh`
36. Check status of listener: `sudo systemctl status lf_listener` (if not running type)
    - `sudo systemctl start lf_listener`
37. Check status of filemon: `sudo systemctl status lf_filemon` (if not running type)
    - `sudo systemctl start lf_filemon`
38. `sudo reboot` (You need to make sure services will start on reboot.)

Problems - Why isn't my server working?

I have started the services but still no data?
Use `sudo systemctl status lf_listener` *to check if it is running. Start it if not. If it won't launch when you start it, you likely haven't made it executable. Try doing so with* `chmod +x` *While you're at it, check the lf_filemon service, too.*

I've started my services and I see lf_listener and lf_filemon present when I run ps -aux. I'm still not getting any data in lf_data.txt.
Have you updated the lf_listener script with your domain info? It will only listen for queries to the specified domain.

My services are running but I am still not getting any sms messages.
This could be caused by server things. Check the following, in order:

1. *Are you seeing your queries in lf_data.txt. If no, ensure you have edited lf_listener with your domain info, and restart the service using:* sudo nohup python3 /home/ubuntu/lf_listener_1.0.0.py &
2. *If this issue persists, check if a file corresponding to your session id (shown on the client side) is present in* `/home/ubuntu/lf_exfil/`. *If the session file is not present, run ps -aux to ensure your lf_filemon service is running. Start it if required using:* nohup python3 /home/ubuntu/lf_filemon.py &
3. *Still no sms message? Check the captured phone number in* `/home/ubuntu/lf_exfil/<session>.txt`. *Is it formatted correctly? It must be eleven digits and start with the 1 country code. If the number is correctly formatted, check that it exists in the whitelist file:* `/home/ubuntu/lf_phonelog/lf_phonewhitelist`. *It must be in there or not sms will be sent.*
4. *All that it still isn't working? Send a test sms message by running the following command from /home/ubuntu/:* `python3 lf_sms.py 1<phonenumber> <sessionid> 0` *For example, if your phone number is 5556667777 and your sessionid is 12345, you'd run:* `python3 lf_sms.py 15556667777 12345 0` *If you don't receive an sms message, there is a misconfiguration in your Twilio API settings. Verify your SID, AUTHTOKEN, and SENDERPHONE are correct and exported in* `/etc/environment`.

I did all this, I see entries in lf_data.txt, <session>.txt, and test sms messages work. Still nothing.
Beats me. Pull a Windows and reboot? Also, remember this is testing security on the client side. Is it possible security has blocked you?