FIELD NOTE / SECURITY

Guide to Build XSSHunter(Latest) with Docker and Cloudflare Proxy

Petruknisme · 3 Mar, 2025 · 4 min to read

In this blog post, I'll guide you through setting up the latest version of XSSHunter using Docker, simplifying deployment. I'll also show you how to integrate Cloudflare's proxy service for enhanced security and performance, including DDoS protection and IP masking. Additionally, I'll explain how to limit access to the admin panel by whitelisting IP addresses, ensuring that only trusted users can access it.

In this article, we will not be using https://github.com/trufflesecurity/xsshunter (https://github.com/trufflesecurity/xsshunter) to build XSSHunter. This is because the repository seems to be not well-maintained, and its documentation and features are not up to date. Instead, we will be using the repository at https://github.com/rs-loves-bugs/xsshunter (https://github.com/rs-loves-bugs/xsshunter).

The repository https://github.com/rs-loves-bugs/xsshunter (https://github.com/rs-loves-bugs/xsshunter) has a more up-to-date codebase and documentation, as well as additional features that are not present in the main repository.

Here is some of the update summary:

- Single user support and multi user support
- Full Trufflehog support
- No blurred screenshots
- Local Storage support
- Slack, Discord, Telegram and custom notifications

Requirements

- A server with `git`, `docker` and `docker compose` plugin installed. You can follow these instructions (https://docs.docker.com/engine/install/) to get `docker` and `docker compose` plugin on it
- Two hostnames(for example `admin.example.com` and `xss.example.com`) you pointed to your server's ip address

Clone Repo and Initial Setup

After the requirements are completed, we will clone the repository and perform the initial setup.

> *"Keep in mind that all commands will be executed with **root privileges**, and make sure Docker is running."*

Licensed under CC BY-SA 4.0

```
# clone the repository
git clone https://github.com/rs-loves-bugs/xsshunter
# enter the directory
cd xsshunter
# creating env using template
cp env.example .env
```

Check if docker is already running. If not, try run the service.

```
root# systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor p
     Active: active (running) since Mon 2025-02-24 12:57:35 UTC; 6 days ag
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 3555596 (dockerd)
      Tasks: 65
     Memory: 358.2M
        CPU: 21min 52.877s
     CGroup: /system.slice/docker.service
             ├─3555596 /usr/bin/dockerd -H fd:// --containerd=/run/contain
```

Setting Environment File

We need to adjust the variables in the `.env` file. Important variables that need to be adjusted are listed below:

- `SESSION_SECRET_KEY` : set this to a long random string and don't forget to use "" (quote mark)
- `HOSTNAME` : (sub) domain for admin panel(ex. admin.example.com)
- `XSS_HOSTNAME` : (sub) domain for XSS payload(ex. xss.example.com)
- `PANEL_USERNAME` : admin panel username (email format: admin@ex.com)
- `PANEL_PASSWORD` : password for admin panel
- `POSTGRES_PASSWORD` : password for xsshunter database.

If you want to use Oauth login like gmail SSO instead of email and password, you can adjust this variable:

Licensed under CC BY-SA 4.0

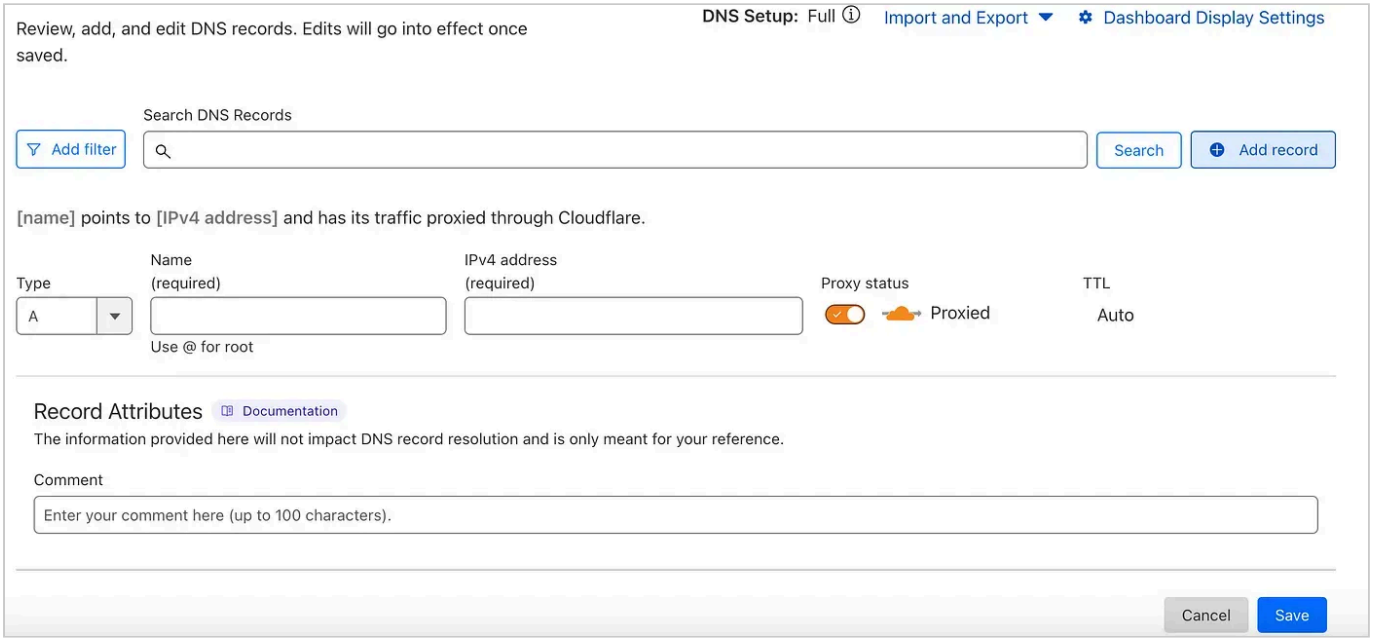
.env

```
OAUTH_LOGIN=false
CLIENT_ID=
CLIENT_SECRET=
GMAIL_ACCOUNTS=a_gmail_account@gmail.com, another_gmail_account@gmail.com
```

The rest of the variables are up to you; I will leave them blank or set to their default values.

Domain Setup

As I mentioned earlier, we need two (sub)domains for this XSSHunter setup, so we must create A records in Cloudflare. To create A record, go to DNS → Records → Add record



Fill in the name with your preferred subdomain like `admin` or `xss` (for the payload) and enter your VPS public IP address in the IPv4 field. Leave the proxy status set to the default (proxied). Click save.

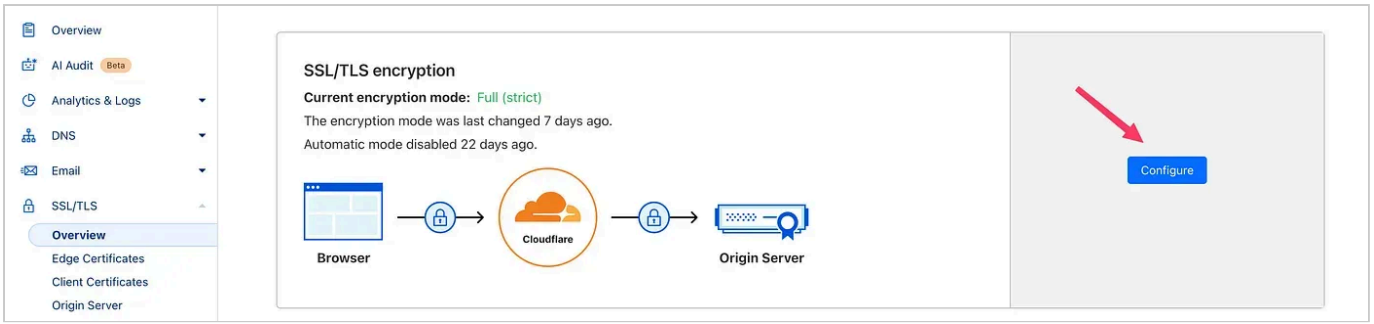
Use dns propagation tool to check if your subdomain already propagated, you can use this website https://dnschecker.org/ (https://dnschecker.org/).

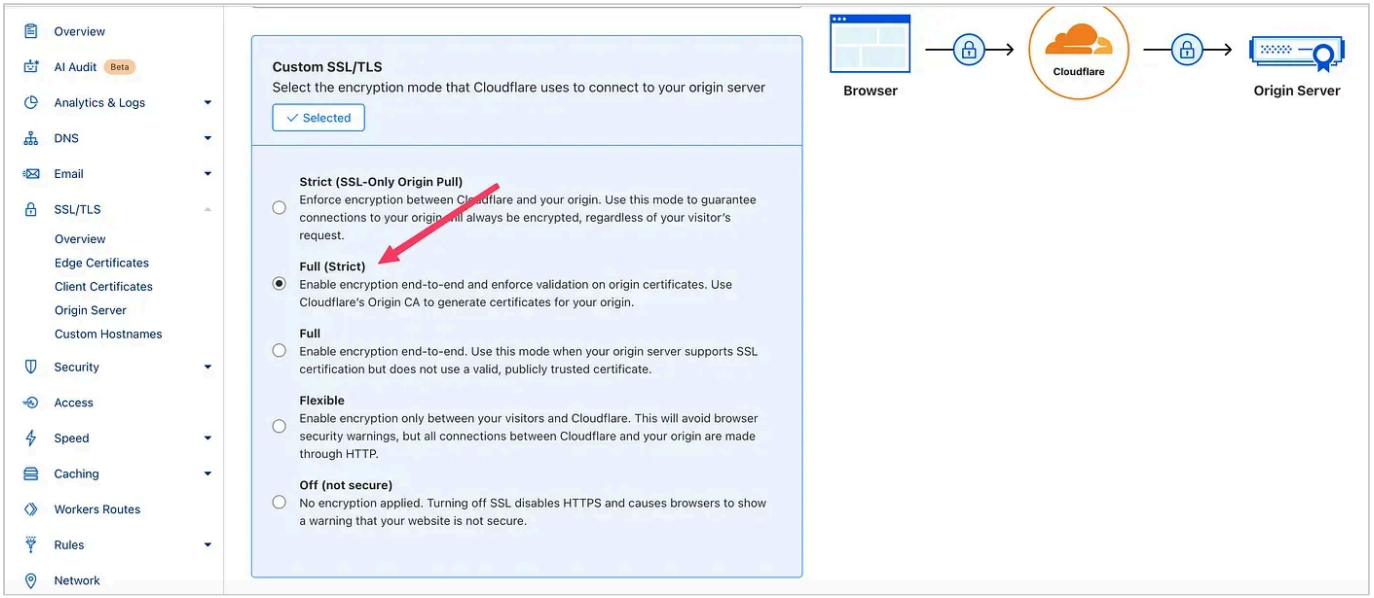
Cloudflare (TLS/SSL) Proxy Setup

Since we want to use Cloudflare's proxy to hide our IP, we also need to set the SSL/TLS encryption mode to **Full (Strict)** in Cloudflare setting. If this setting is not set to **Full (Strict)**, we will encounter too many redirects when accessing the admin panel.

Enable it in SSL/TLS → Overview → SSL/TLS Encryption → Configure

Licensed under CC BY-SA 4.0





Choose Full (Strict) for SSL/TLS Encryption

Setting up Webserver

The web server built into the repository is a dockerized Caddy. By default, the configuration allows everyone to access both the admin panel and the XSS payload. However, we need to limit access to the admin panel. According to the Caddy documentation, we can restrict access using a matcher for `remote_ip`. But when I implemented it, accessing the panel resulted in a block, whether the IP was whitelisted or not.

After some trial and error, I found that this issue was caused by using the Cloudflare proxy. As a result, Caddy cannot retrieve the real IP in `remote_ip`; instead, the real IP is located in the `X-Forwarded-For` header.

Here is an example of a Caddy log. The `remote_ip` field is filled with a Cloudflare IP instead of the real public IP of the client accessing the web.

```
{"level":"info","ts":xxxxxx.xxxxx,"logger":"http.log.access.log0","msg
":"handled request","request":{"remote_ip":"172.xxx.xxx.xxx","remote_p
ort":"23572","client_ip":"172.xxx.xxx.xxx","proto":"HTTP/2.0","method"
```

Licensed under CC BY-SA 4.0

```
:"GET","host":"admin.example.com","uri":"/app/img/icons/favicon-16x16.
png","headers":{"Sec-Fetch-Dest":["image"],"Sec-Ch-Ua":["\"Not(A:Brand
\";v=\"99\", \"Google Chrome\";v=\"133\", \"Chromium\";v=\"133\""],"Se
c-Ch-Ua-Mobile":["?0"],"Sec-Ch-Ua-Platform":["\"macOS\""],"Sec-Fetch-M
ode":["no-cors"],"Priority":["u=1, i"],"X-Forwarded-For":["REAL_IP_HER
E"],"Sec-Fetch-Site":["same-origin"],"X-Forwarded-Proto":["https"],"Cf
-Visitor":["{\"scheme\":\"https\"}"],"Cf-Ray":["xxxxxxxxx-XXX"],"Cf-Ip
country":["XX"],"Accept-Encoding":["gzip, br"],"Cdn-Loop":["cloudflare
; loops=1"],"User-Agent":["Mozilla/5.0 (Macintosh; Intel Mac OS X 10_1
5_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/53
7.36"],"Accept":["image/avif,image/webp,image/apng,image/svg+xml,image
/*,*/*;q=0.8"],"Accept-Language":["en-US,en;q=0.9,id;q=0.8"],"Cf-Conne
cting-Ip":["REAL_IP_HERE"],"Dnt":["1"],"Referer":["https://admin.examp
le.com/app/"],"Cookie":["REDACTED"]},"tls":{"resumed":false,"version":
772,"cipher_suite":4865,"proto":"h2","server_name":"admin.example.com"
}},"bytes_read":0,"user_id":"","duration":0.005094975,"size":170,"stat
us":404,"resp_headers":{"Content-Security-Policy":["default-src 'none'
"],"Content-Length":["170"],"X-Xss-Protection":["mode=block"],"Server"
:["Caddy"],"X-Content-Type-Options":["nosniff"],"Content-Type":["text/
html; charset=utf-8"],"Date":["Mon, 03 Mar 2025 10:46:53 GMT"],"Alt-Sv
c":["h3=\":443\"; ma=2592000"],"X-Frame-Options":["deny"]}}
```


From the log above, we need to modify the `Caddyfile` so that the admin panel can only be accessed by whitelisted IP addresses from `X-Forwarded-For`. Here is the modification:

Caddyfile

```
{$XSS_HOSTNAME} {
 reverse_proxy xsshunterexpress-service:8080
}
# Global Reusable Blocks
(trusted_proxies) {
    import ./configs/cloudflare-proxies
}
{$HOSTNAME} {
    log {
     output stdout
     level debug
    }
    @whitelist_ip {
    header X-Forwarded-For INSERT_PUBLIC_IP_HERE
```

Licensed under CC BY-SA 4.0

```
    }
    handle @whitelist_ip {
     reverse_proxy xsshunterexpress-service:8080
     }
    handle {
     respond 403
    }
}
```

Running the tool

After following all of the above steps, run XSSHunter using `docker compose` from the `XSSHunter` folder:

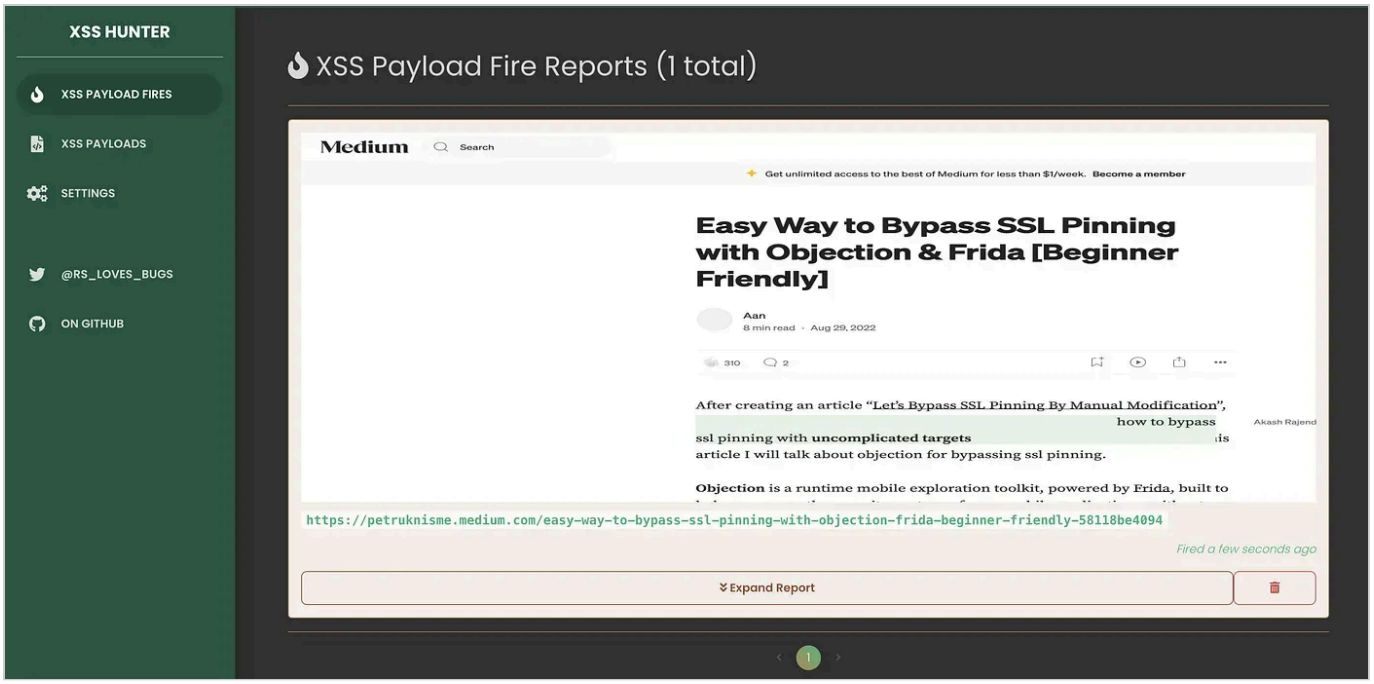
```
root@localhost # docker compose up -d
```

You can check the service using `docker ps` command.

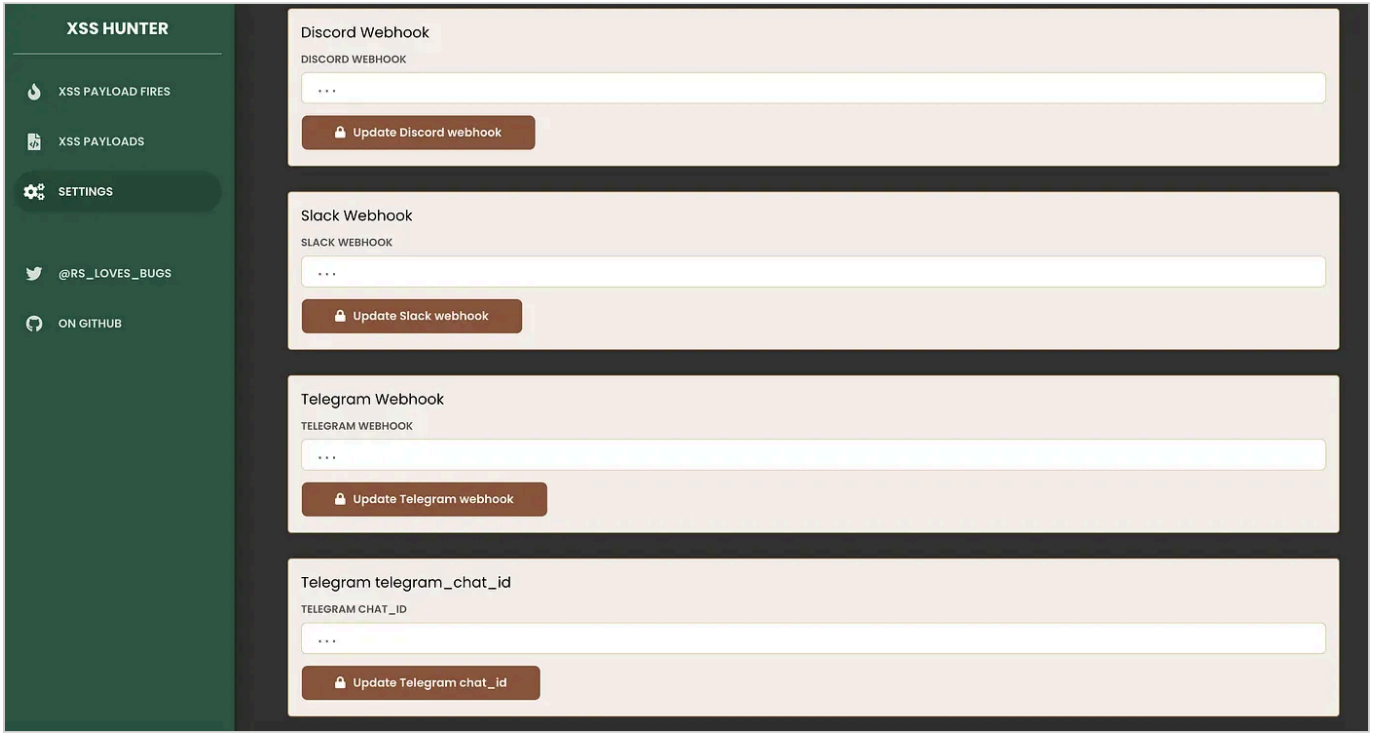
```
# docker ps
CONTAINER ID   IMAGE                                    COMMAND
77b4453bc3c5   xsshunter-xsshunterexpress-service       "/app/docker-entryp
3131e75c1926   postgres                                 "docker-entrypoint.
f362d69ddb99   caddy                                    "caddy run --config
421d0071ea8a   xsshunter-xsshunterexpress-trufflehog    "gunicorn --bind 0.
```

If everything goes well, the status will be “up,” and you will be able to access the panel from the whitelisted IPs.

Licensed under CC BY-SA 4.0



XSSHunter homepage



XSSHunter setting for notification

Done! If you think this post is helpful, feel free to share it.

Licensed under CC BY-SA 4.0

Originally published on **Medium** (https://medium.com/@petruknisme/setting-up-xsshunter-latest-with-docker-and-cloudflare-proxy-6c35db80762c).

Security # Tools # Tutorial

Licensed under CC BY-SA 4.0