□□ Hacker Tools: How to set up XSSHunter

BY ANNA HAMMOND · AUGUST 18, 2021 · LAST UPDATED ON MARCH 6, 2025

Cross-site scripting or XSS vulnerabilities are incredibly common and not to be underestimated. Oftentimes, they can even occur in the dark, in places where you can't see the result. In this week's instance of Hacker Tools, we're going to look at XSSHunter, a tool to help you find blind XSS vulnerabilities by @IAmMandatory.

> **"Tool tip: use @XssHunter by @IAmMandatory and score bounties while you're asleep. #HackWithIntigriti pic.twitter.com/oVGwDXrzBK**
>
> **— Intigriti (@intigriti) June 6, 2019"**

□□ What is XSSHunter?

Let's hear it from the XSSHunter website:

> **"XSS Hunter allows you to find all kinds of cross-site scripting vulnerabilities, including the often-missed blind XSS. The service works by hosting specialized XSS probes which, upon firing, scan the page and send information about the vulnerable page to the XSS Hunter service.**
>
> **https://xsshunter.com/features"**

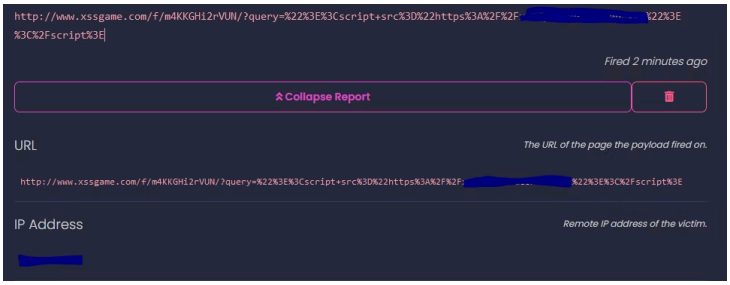
We're going to dissect that quote. There are an incredible amount of web services out there that are vulnerable to cross-site scripting attacks. If you're unaware of what XSS is, check out our Hackademy!

These XSS vulnerabilities aren't all the same. Some are reflective, meaning your input is reflected directly onto the page; Others are stored in a database and then shown at a different time; You have DOM-based XSS; But you also have blind XSS. This last type is very often overlooked because it is hard to see the results of your actions, hence being called blind XSS.

Imagine a contact form where you can input a message. This message will be shown to somebody working for the company. If this is vulnerable to XSS, then the support personel might get an alert window pop up on their screen, but you will have no knowledge of it. That's the main problem XSSHunter is trying to solve.

When the XSS payload gets triggered, it will request a probe from the XSSHunter service. This will then cause the page to be scanned and information to be sent back to you via email. This way you stay up to date on when and where you XSS fired.

Enough theory, let's set up our own instance!



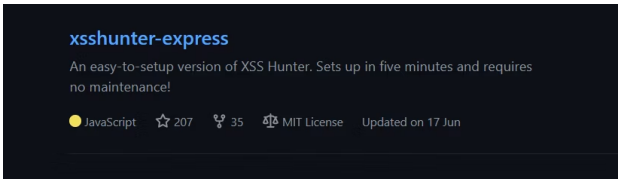
A hit to our XSSHunter service!

□□ Setting up XSSHunter Express

Whilst you can use the [XSSHunter website] to remotely use this amazing service, it might be beneficial for you to set up your own instance.

Why? Well, hosting things yourself, you have more granular control over the configuration. Additionally, this makes sure that only you can view your results. The notifications you receive via mail don't end up in your spam folder and you won't miss out on some sweet executions of your payload if the XSSHunter website were to go down or get blocked.

Reasons enough to host your instance and others would agree with you, which is why [XSSHunter Express] was created. Let's talk about setting it up!



XSSHunter Express on Github

Setup: DNS

One of the things you're going to need is a place to host your XSSHunter instance. This server then also needs to be accessible through the internet via a domain name. I bought a new domain for this purpose and added an A-record to the DNS to point the subdomain `xss` to my server's IP address. The image below shows how that looks.

DNS A record for

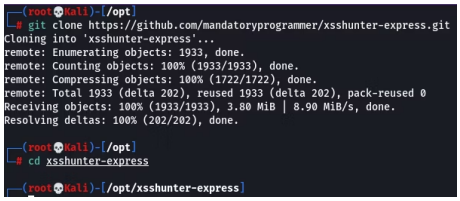
```
xss
```

subdomain

Setup: Configuration

Ready to continue? Let's clone [the repository from Github](#) and go into the newly-created directory.

```
git clone https://github.com/mandatoryprogrammer/xsshunter-express.git
cd xsshunter-express
```



Cloning the repository

There's only one file we need to edit to configure our instance: **docker-compose.yml** . Open it in your favorite text editor and let's get to work!

There are only a couple of values you need to change here and they have all been marked in the screenshot below.

- Version: For some reason, the repository holds the value **3.9** by default, but **docker-compose** does not support that version. Therefore, change this value to **3.3** .
- Hostname: This field should contain the hostname to be used for your service. Note that this should be the same one we configured in the DNS earlier.
- SSL contact email: XSSHunter will use [LetsEncrypt](#) to generate an SSL certificate to use (and will keep it up to date for us). Enter an email address to be used for this SSL certificate.
- The mail settings: XSSHunter will send notifications when one of your payloads was executed. I created a custom mail address for these and set all of the values to that. This way, I have a mailbox dedicated to just XSS vulnerabilities!

```
version: "3.3"
services:
  # XSS Hunter Express service
  xsshunterexpress:
    build: .
    environment:
      # [REQUIRED] The hostname/domain pointed to
      # the IP of the server running this service.
      # SSL will automatically be set up and
      #  renewed with LetsEncrypt.
      - HOSTNAME=your.host.name
      # [REQUIRED] Email for SSL
      - SSL_CONTACT_EMAIL=YourEmail@gmail.com
      # Maximum XSS callback payload size
      # This includes the webpage screenshot, DOM HTML,
      # page text, and other metadata. Note that if the
      # payload is above this limit, you won't be notified
      # of the XSS firing.
      - MAX_PAYLOAD_UPLOAD_SIZE_MB=50
      # Whether or not to enable the web control panel
      # Set to "false" or remove to disable the web UI.
      # Useful for minimizing attack surface.
      - CONTROL_PANEL_ENABLED=true
      # Whether or not to enable email notifications via
      # SMTP for XSS payload fires.
      - SMTP_EMAIL_NOTIFICATIONS_ENABLED=true
      - SMTP_HOST=smtp.gmail.com
      - SMTP_PORT=465
      - SMTP_USE_TLS=true
      - SMTP_USERNAME=YourEmail@gmail.com
      - SMTP_PASSWORD=YourEmailPassword
      - SMTP_FROM_EMAIL=YourEmail@gmail.com
      - SMTP_RECEIVER_EMAIL=YourEmail@gmail.com
```

docker-compose.yml file

⚙️ Setup: Running

Configuration. Check! Let's start this bad boy up!

This works in two simple steps. One; startup the database. Two; startup XSSHunter.

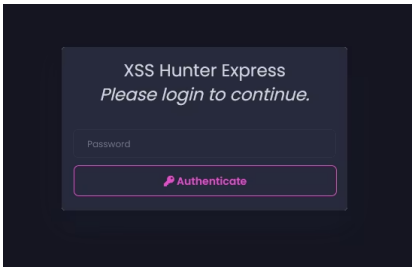
```
docker-compose up -d postgresdb
docker-compose up xsshunterexpress
```

Note that when running this for the first time, it's going to take a while to finish. That's because it needs to pull a couple of Docker images, along with creating and building a new container. Give it some time, grab a coffee or do some hunting on intigriti in the meanwhile!

After it has finished, you will see an attention message as shown in the image below. Take note of the password being generated (Write it down, save it in your password manager). And no, I didn't forget to blur out the password in this image. By the time you can read this, the instance is long gone! (Because I knew some of you were going to try it).

Attention message XSSHunter

Is that it? Yes, that's it. Let's go to **https://your.domain.name/admin** and log in!

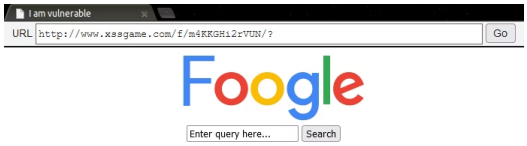


Login screen

Now it's time to have some fun. In the next paragraph, we're going to be using XSSHunter to perform a very simple attack!

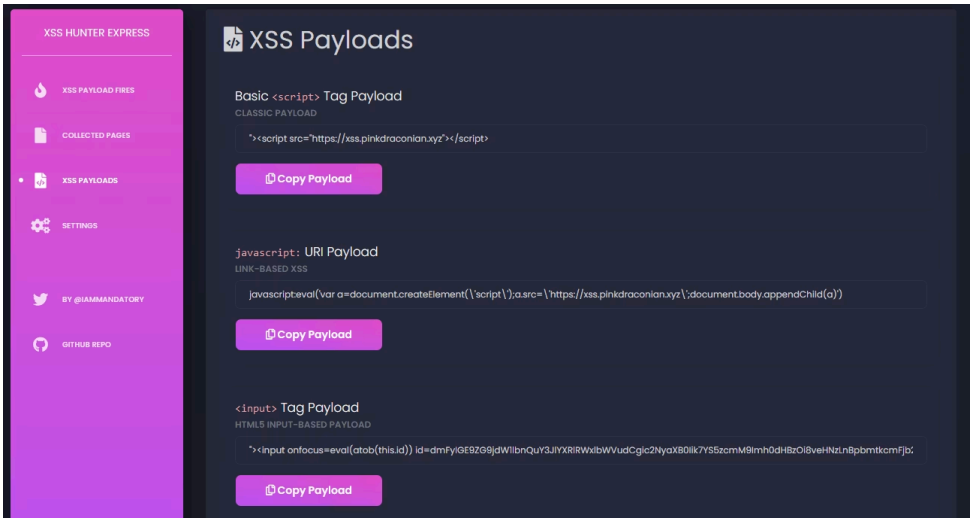
□□ Practical example

Time to see if everything works. Let's do this using [a simple XSS lab](). On this page, we have a search engine that's vulnerable to XSS.



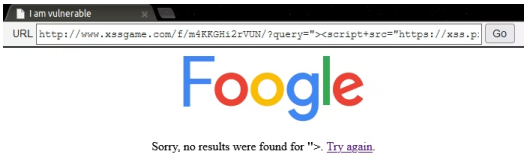
Vulnerable XSS lab

Let's exploit this lab! We need a payload and XSSHunter can help us with that. This list is great if you quickly need to get up and running, however, nothing is stopping you from getting creative and creating your own for more nuanced situations. In this case, we'll be fine just copying the basic payload.



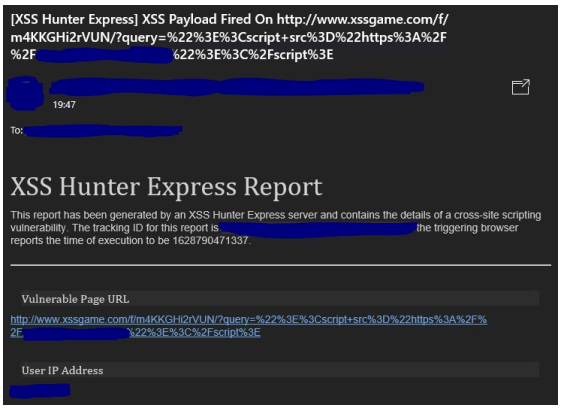
XSS payloads page

Paste your payload into the lab and search.

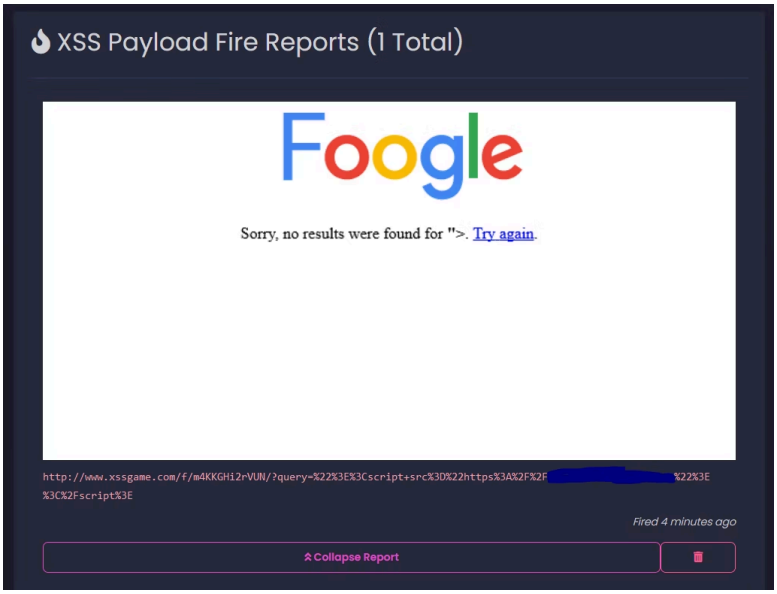


After exploitation

You will now notice that you have received an email telling you your XSS payload triggered! On your admin panel, you will also be able to see the details of the successful XSS exploitation!



Notification email



XSS payload fire report

Conclusion

XSSHunter is a powerful service that will help you find some crazy blind XSS.

If you would like to recommend a tool for us to cover next week, then be sure to let us know down below. Also be sure to check out all the previous Hacker Tools articles, such as the last one on Ciphey.

Did you know that there is a video accompanying this article? Check out the playlist!