**Finding Vulnerable Targets**

Finding Vulnerable Targets is our First Phase

For that, we will use Search Engines Like Shodan, Zoom Eye etc.,

Then we will either test them using automated tools or manually.

1. **Methodology**

First, we will be targeting an organization by using Shodan dorks, to find out the Log4J Vulnerability,

For us used the Dork:

org: “Organization Name”, product: java

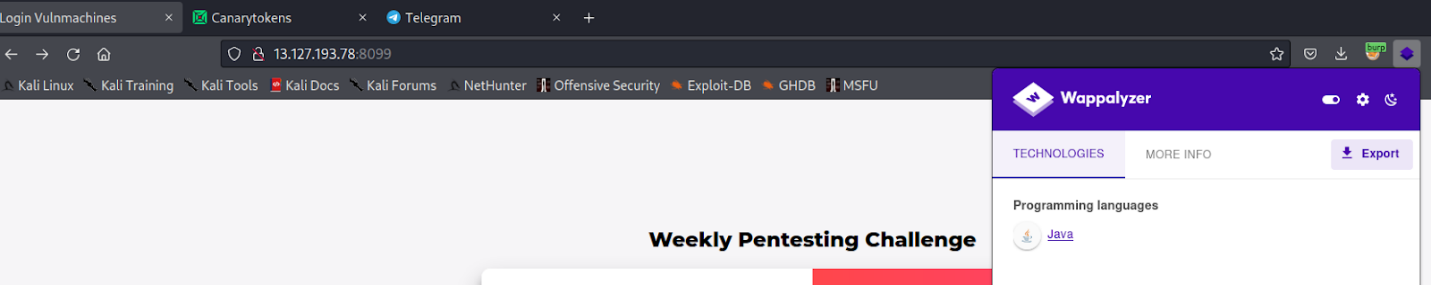
It will return us the List of Hostnames, and the services that are running on the Java Platform. We don’t know precisely whether these targets are vulnerable or not for that, we will check for the targets, if they are live, if the targets are live then, we will use some Log4J vulnerable scanners to detect the Vulnerability (CVE-2021-44228)

1. **Tools Used**

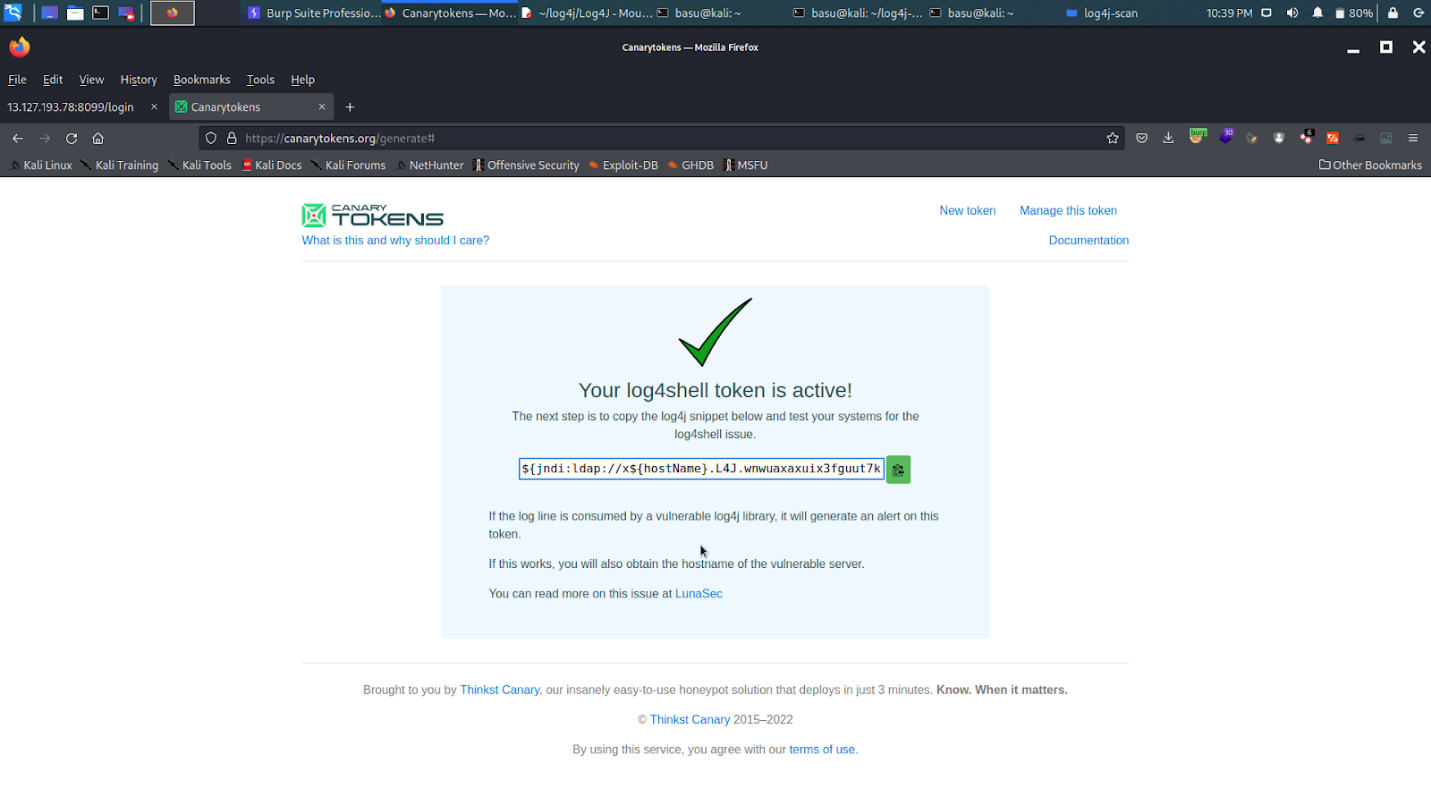
* Firefox Web Browser
* Wappalyzer Extension
* Canary Tokens
* Burp Suite

1. **Steps to Procedure**

* For the demonstration purpose, we used the lab **Log4J** offered by Vuln machines (https://www.vulnmachines.com/)
* We have observed through the Wappalyzer extension that the Target Website runs on Java.

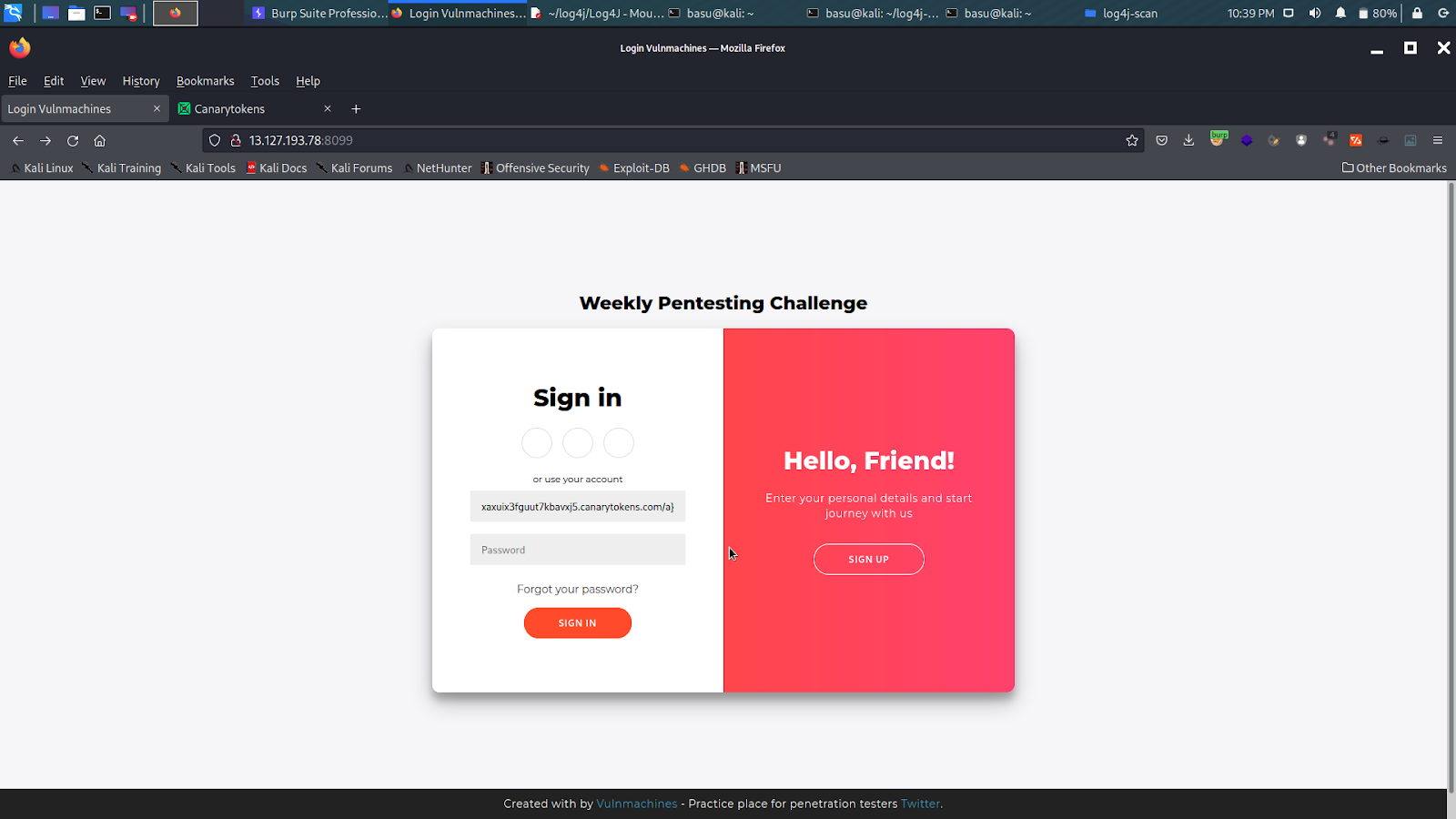


* Then We used the web service Canary Token to test for the log4Shell vulnerability.
* Canary tokens is a website, that notifies us if the target has been triggered by our payload.

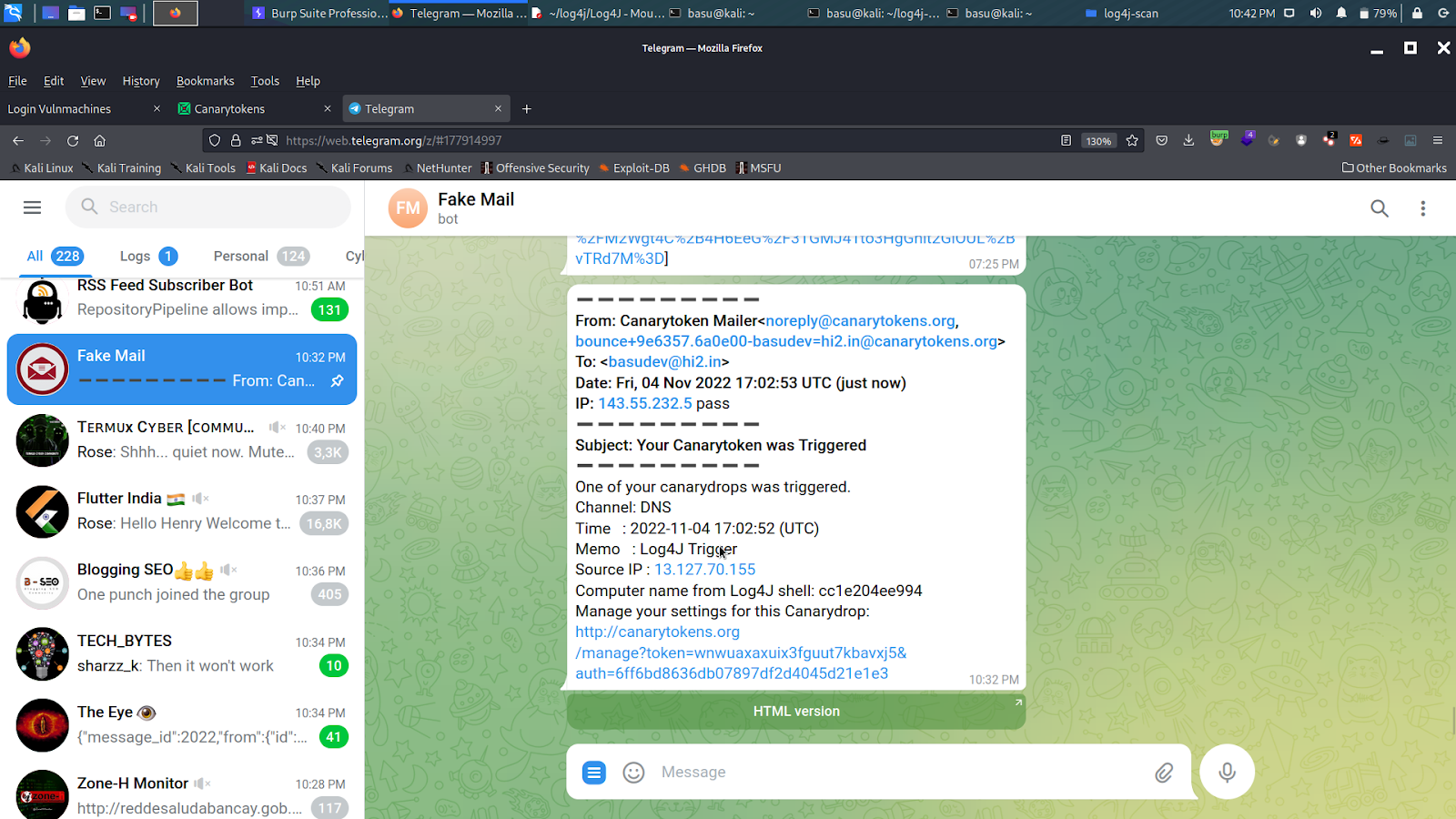


[https://canarytokens.org/generate#](https://canarytokens.org/generate)

* After Generating a Well-Crafted payload from Canary Tokens, we have tried to put the payload everywhere on the website, as it can trigger at any moment and on any page.

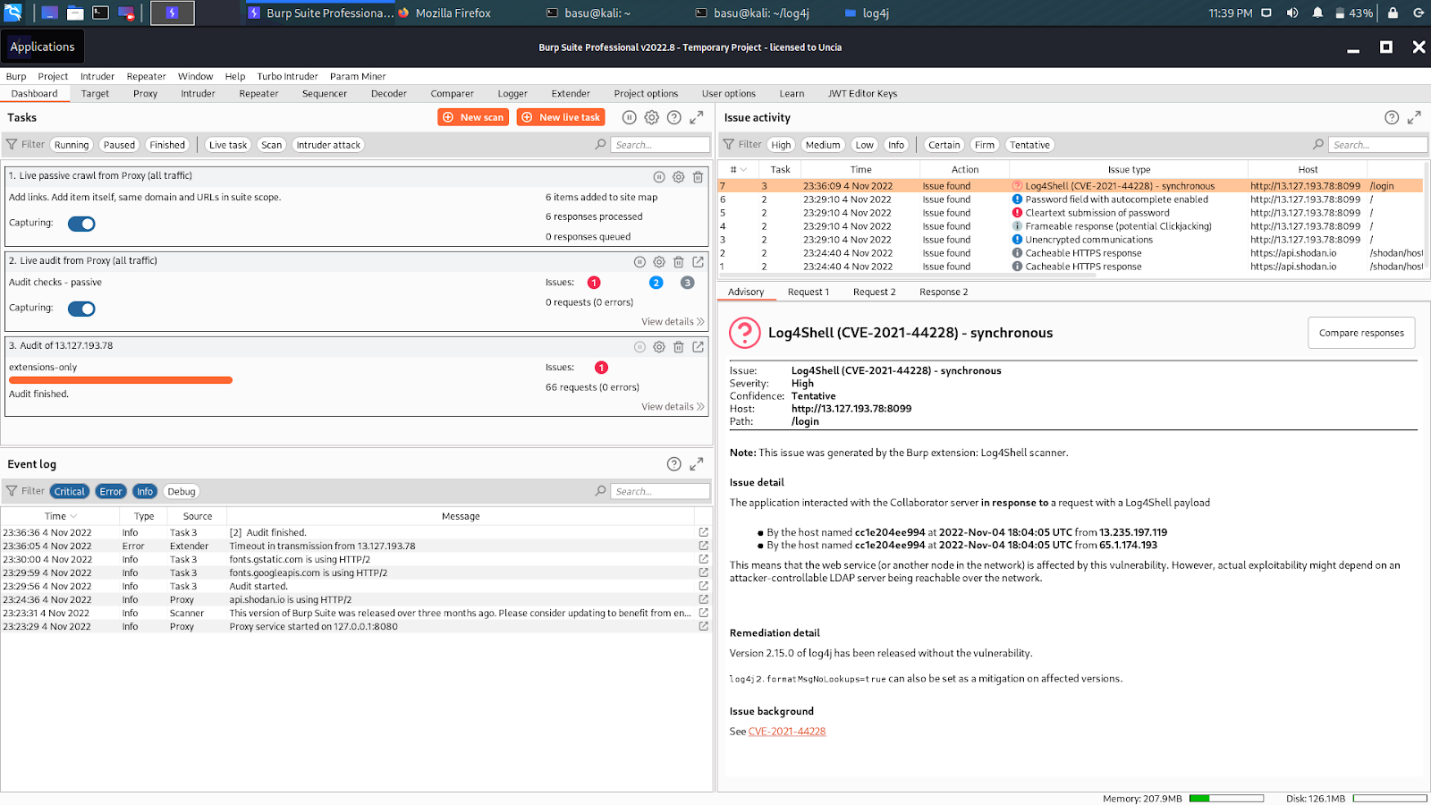


* We used the token in the Login fields; after a few minutes, the payload worked, and I got a notification to my email. (For testing purposes, I used Fake Mail a telegram bot used to defeat spam)



**Detecting Log4Shell using Burp Suite**

* For detecting the Log4shell, we can use the Burp suite,
* We have to install the log4J scanner burp extension, and capture the traffic of the target and send it the scanner that’s it, it will automatically scan for the Log4J vulnerability and notify us



1. **Conclusion**

Since Log4J is a dangerous vulnerability, it is a Remote code execution possibility by nature. Even if any firewall protects the target, there are many publicly known exploits to bypass it.

As of today, it is super hard to find log4j vulnerability, and even detection is super hard. Still, hackers always find a new way to exploit it