Initial Dynamic Security Scan of Hackazon using Rips

Yingwei Liu

Liberty University

Abstract

The goal of this phase is to perform the security scan of Hackazon web application by Rips security scan software. This section requires to familiarize the scanning and enumeration via Rips. Also, provide detailed report with all vulnerable code blocks that found in Hackazon web application’s source code. Then, show discussion of three examples of XSS.

Keywords: Rips, Hackazon, XSS.

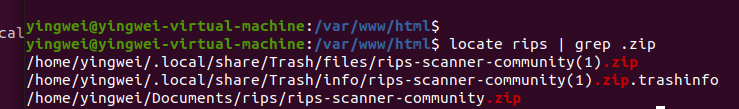
Initial Dynamic Security Scan of Hackazon using Rips

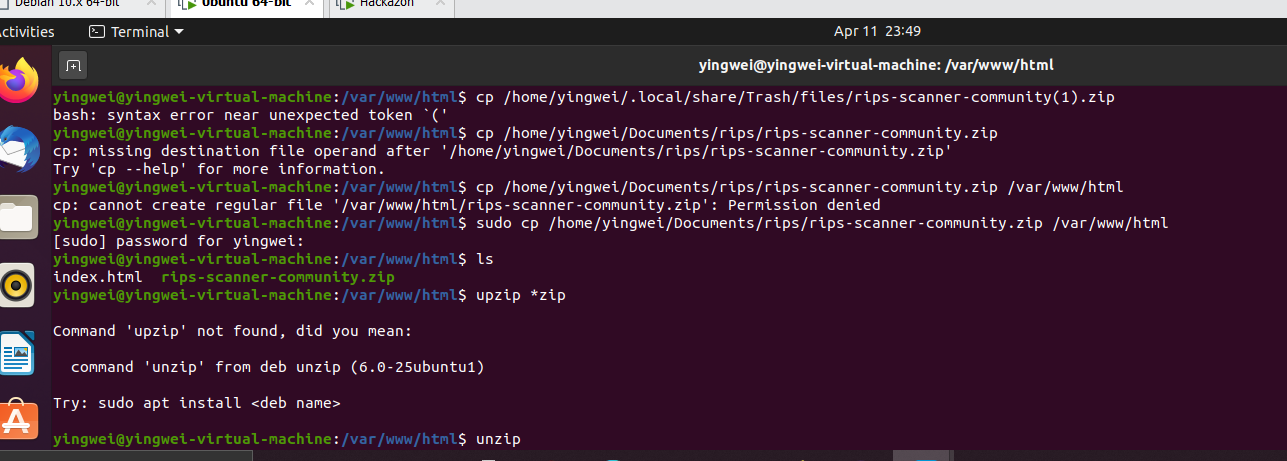
The phase two’s context is to show that I have successfully download the rips software. Then, apply the skills that we have and detect the vulnerabilities of the Hackazon web application. First, I need to provide the evidence of the configuration of rips. Second, I will present a detailed vulnerability report of the Hackazon Web Services. Third, it will demonstrate a discussion of XSS vulnerabilities.

# Evidence of download of Rips

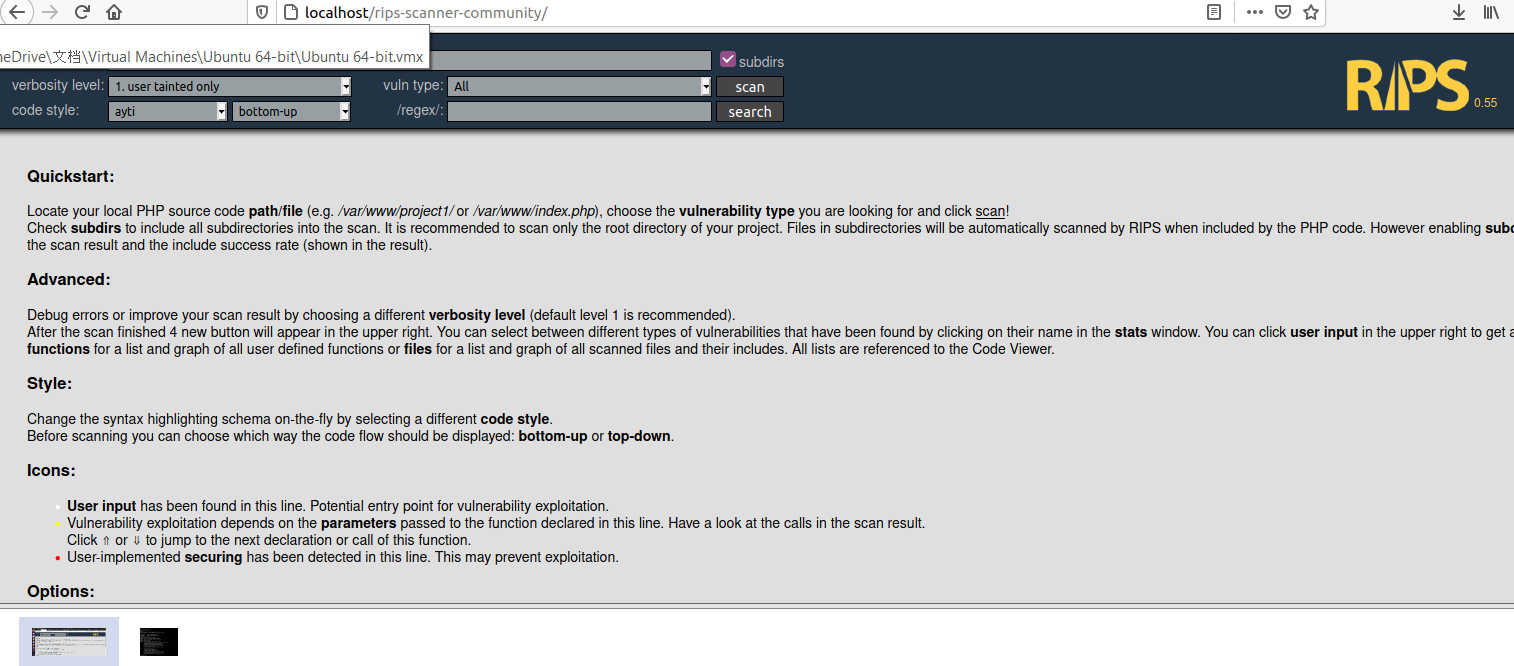
Find the root of apache2.



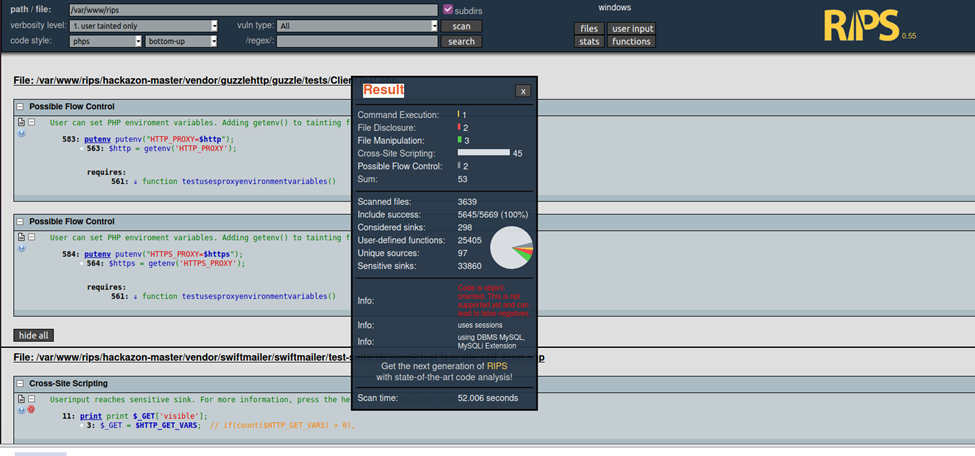




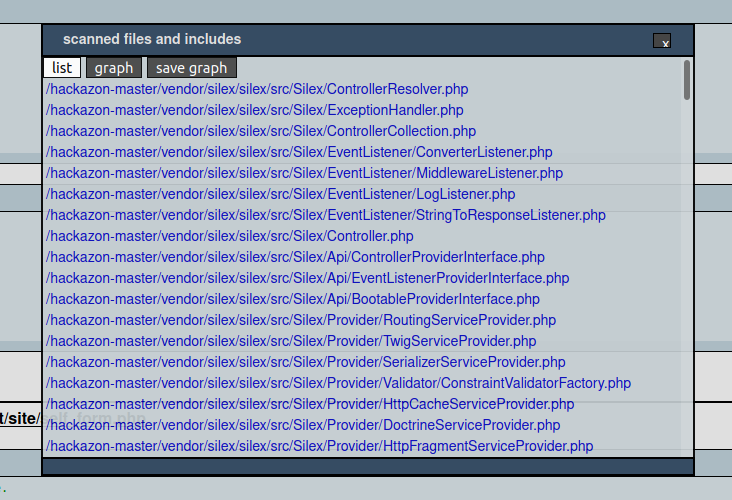
Access the RIPS

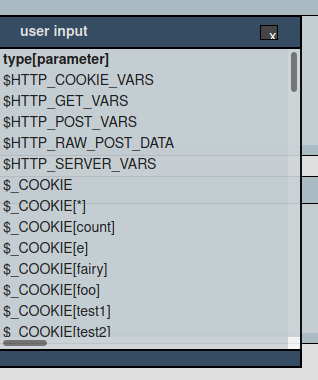


## Report1



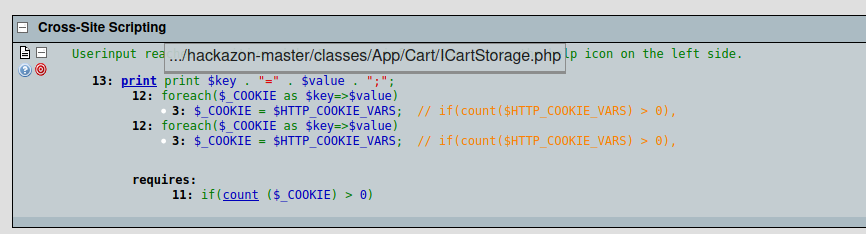
**Scanning report of the Hackazon Websites.**

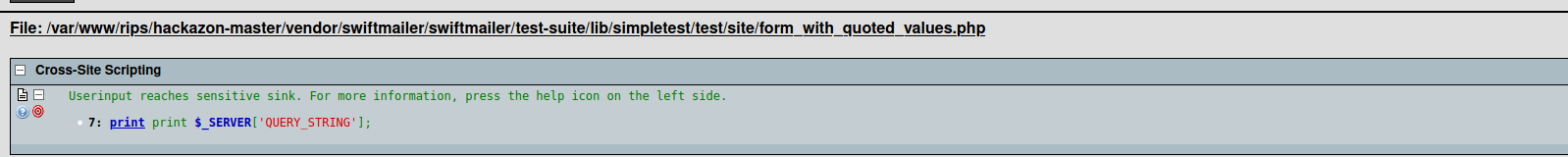


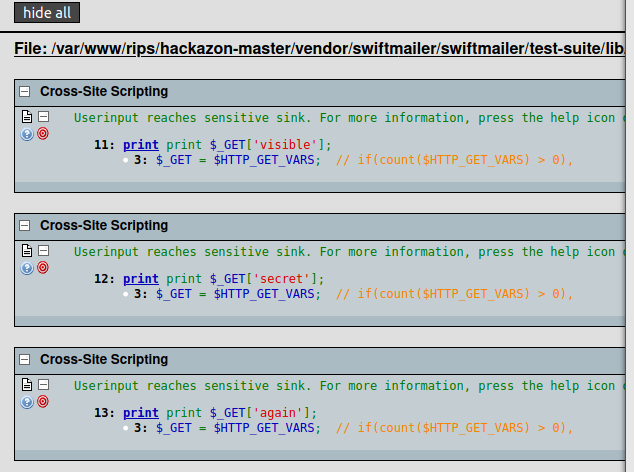


**User Input List**

**Cross-site Scripting**







Cross-site scripting is one of the severely security vulnerability that discovered in the Hackazon web application. This attack allows the attacker to inject client-side scripts into web pages. Especially, the Hackazon Web application is an e-commerce website, it will have huge amount of customer access to the websites every day. If the Hackazon do not fix these vulnerabilities, it might result huge damages to customer’s privacy. Just like Chauhan mentioned, “XSS attacks mostly occur for financial gains, a notable one is the past attack against e-commerce giant eBay. The hackers injected a malicious Javascript code into several listings for cheap iphones, which in turn redirected users to a fake login page created to compromise user credentials.” These three cross-site vulnerabilities attacks both relate the with user input. If the user inject the malicious code and it will change the source code and leads the other users to different server. According to the OWASP, “The malicious content sent to the web browser often takes the form of a segment of JavaScript, but may also include HTML, Flash, or any other type of code that the browser may execute. The variety of attacks based on XSS is almost limitless, but they commonly include transmitting private data, like cookies or other session information, to the attacker, redirecting the victim to web content controlled by the attacker, or performing other malicious operations on the user’s machine under the guise of the vulnerable site.”

References

Chauhan, B. (2020). Cross-site Scripting(XSS) Attack: All you Need to Know. Retrived from https://www.getastra.com/blog/knowledge-base/cross-site-scripting-xss-attack/

Kirsten, S. *Cross Site Scripting (XSS)* Retrieved from https://owasp.org/www-community/attacks/xss/#.