**OWASP SKANDA – SSRF Exploitation Framework v0.1**

Is your server protected against port scanning? The general answer will be “Yes, I have a firewall which restricts access to internal servers from outside.”

What if I tell you I can still scan the ports on your server and your firewall wouldn’t know about it!

People usually think that it is not possible to do a complete port scan on the web server and other servers behind the firewall. This article will make you think otherwise.

If the web application running on a server has SSRF (Server Side Request Forgery) vulnerability then it is possible to do port scans on the devices behind the firewall. Once you find a SSRF vulnerable server, SKANDA (<https://www.owasp.org/index.php/OWASP_Skanda_SSRF_Exploitation_Framework>) can do an automated scan for you and provide you the status of the ports present on that vulnerable server.

**So what are we going to talk about in this article:**

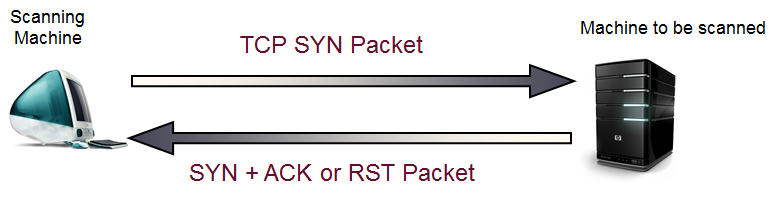
In this article, the agenda is mainly, Cross Site Port Attack. Cross Site Port Attack is a type of SSRF vulnerability (@ONsec\_lab, [http://lab.onsec.ru](http://lab.onsec.ru/)).Using this attack, Riyaz Walikar (@riyazwalikar, <http://www.riyazwalikar.com/>) was able to do a port scan on the internal servers present in facebook’s intranet. Similarly, he was able to exploit this vulnerability on Google, Apigee, StatMyWeb, Mozilla.org, Face.com, Pinterest, Yahoo, Adobe Omniture and several others. **All together he was able to earn a whopping $5k as bounties using this one type of vulnerability**. YES, it can fetch you that much money!!

SSRF can help an attacker do port scan on intranet and external Internet facing servers, fingerprint internal network aware services, perform banner grabbing, identify web application frameworks, exploit vulnerable programs, run code on reachable machines, exploit web application vulnerabilities listening on internal networks, read local files using the file protocol and much more.

**First of all let’s see how NMAP’s works:**

NMAP (<http://nmap.org>) uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. In short, it uses TCP/IP protocol to do a port scan and the packets are sent from your machine which is running the NMAP scan.

So firewalls designed to protect from port scan are keeping a check on the ports and it decides which port is supposed to respond to any packet coming from a machine outside of intranet.

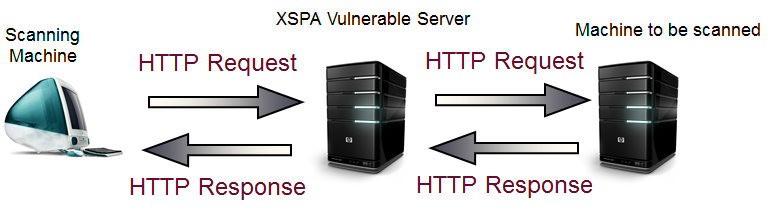


**So is it possible to bypass these firewalls? –> Cross Site Port Attack**

Cross Site Port Attack (XSPA) is a kind of SSRF vulnerability. An application is vulnerable to Cross Site Port Attacks, if the application processes user supplied URLs and does not verify/sanitize the backend response received from remote servers before sending it back to the client.

Port specific payloads are crafted by the attacker and sent to the server. By analyzing the errors or the time delays, in different responses for different ports, the attacker can figure out the status of the ports present on the server. And while exploiting SSRF, the attacker’s machine is not directly interacting with the target server, the vulnerable server is doing all the dirty work for the attacker.

If a server has an application where proper sanitization of the responses is not done and is vulnerable to SSRF, the attacker can insert port specific payloads and scan a target machine using the vulnerable server. Worse, instead of scanning some other target machine the payloads can be crafted which will be directed to the same vulnerable server itself. In this case, the http packets are sent from the server to the same server and the application sends the response to the attacker. By analyzing the responses (response error/time delay), the port status of the vulnerable server can be determined.



**How is SKANDA’s port scan different from normal scanning :**

Other scanners use TCP protocols to scan a particular server whereas SKANDA uses HTTP requests to scan the ports.

With normal port scanners, the attacker’s machine is scanning the ports of a server whereas SKANDA makes the vulnerable server scan its ports and provides you the port status.

**SKANDA – some prior knowledge**

As a pen tester, my goal is to secure my server and check whether the web applications running on my server are vulnerable to SSRF or not. So the payloads in SKANDA are designed such that they attack the server itself, on which the vulnerable web application is running.

In SKANDA, the ports are divided into three states:

* **Closed:** where the port is closed.
* **Open(Error Based):** the port status is determined based on the error message received when connecting to the port.
* **Open (Blind XSPA):** The port status is determined based on the response time.

**SKANDA –** how to use

SKANDA is built as a module for IronWASP and is bundled along with it. To use SKANDA you must first start IronWASP, configure your browser to use IronWASP as the proxy and then browse to the target site. This way IronWASP will collect all the site information in its proxy logs.

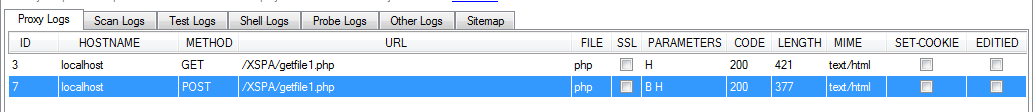


Figure 1 – Setting up IronWASP as proxy

Open up the target site which has a SSRF vulnerable server. Browse through the vulnerable web application flow, how it should be ideally used by a general user. Ex: I have a php test bed (running on apache) which has the functionality to fetch an image from the entered url and saves it locally.



Figure 2 – Recording the flow in IronWASP proxy logs

Once you have completed the flow go to the proxy logs and select that request and start SKANDA.

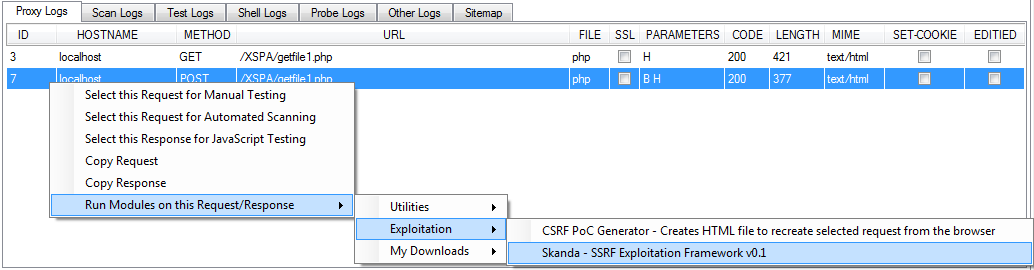


Figure 3 – Starting SKANDA from the logs

As the module is run, two windows will open up. On CLI and other, a GUI interface which will require some details from you before it starts scanning. (Do no close the any window if you want to carry on with the scan)

Below window will appear. SKANDA uses this particular http request as the base request to get the port status. If there is a special case then you may change the request, otherwise click “Next Step ->“.

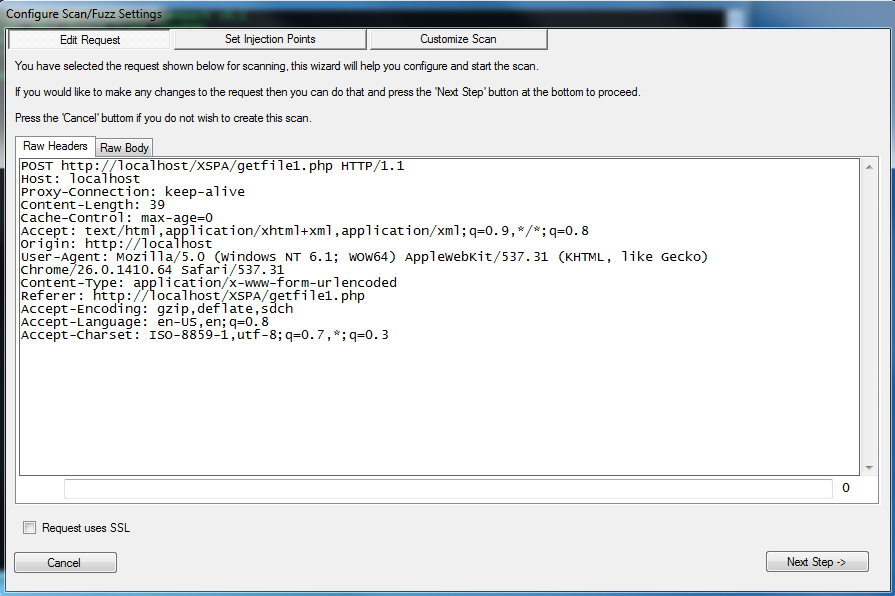


Figure 4 – Making change to the base request

Now the next window which opens up, asks you to locate the suspicious parameter which is SSRF vulnerable. In the case of test bed used by me, the vulnerable parameter is in request body named as ‘url’ (select multiple parameters, if more than one parameter is SSRF vulnerable). Click Next after you are done.

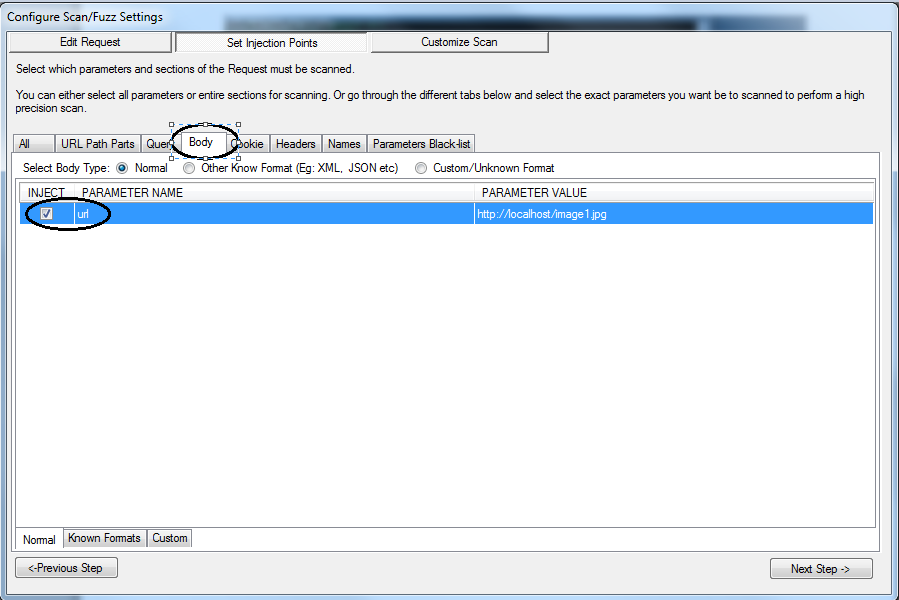


Figure 5 – Select the vulnerable SSRF parameter

Since we are sending http requests, depending on the web application the request may require an active session.

So if you have created any session plugins which you want to be used while scanning, select it when the below screen appears. And click Done.

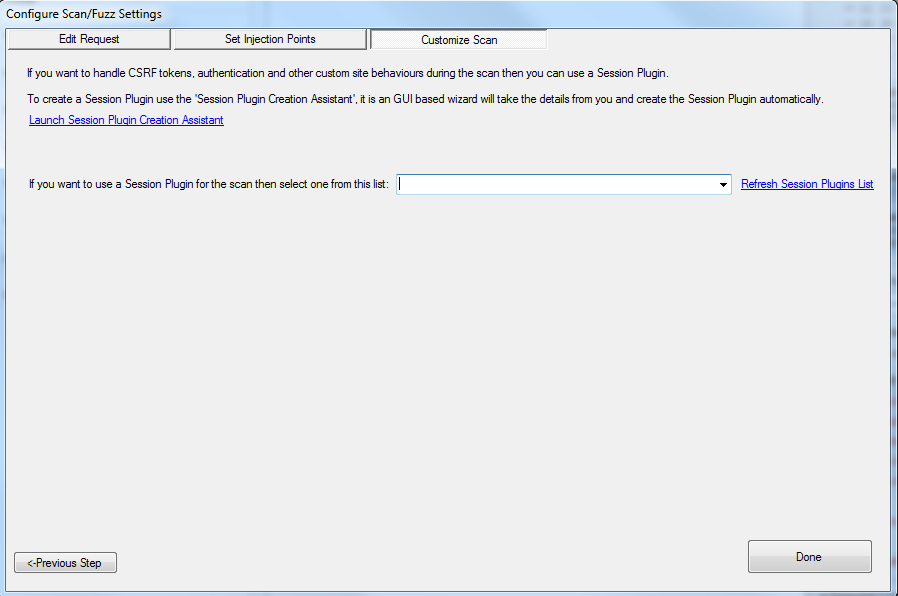


Figure 6 – Session plugin selection

Now the GUI will close and the CLI will come into action.

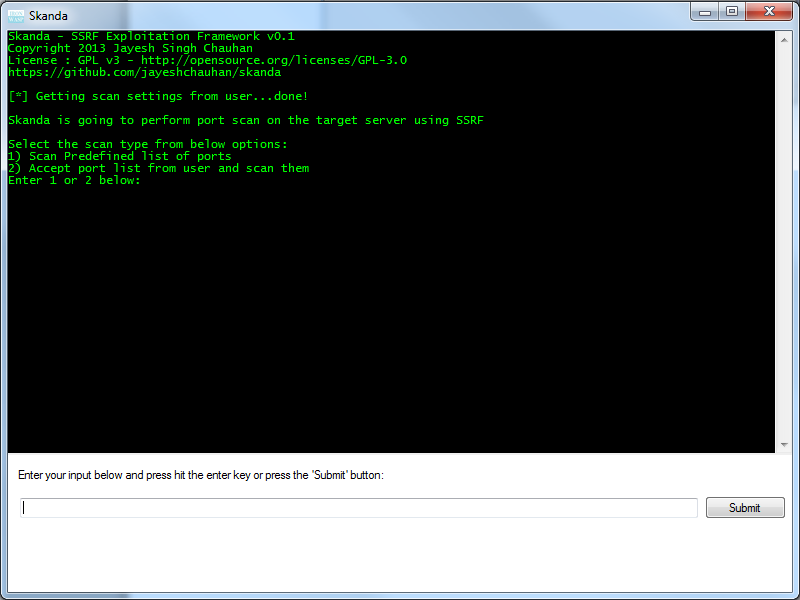


Figure 7 – Command Line Interface

To start with the port scan, enter ‘1’ and submit. We will come to the second option(‘2’) in a while.

Now SKANDA will do the following before starting to scan the ports:

* **Initial Diagnostics:** The moment you start the scan, SKANDA sends the base request a few number of times so that it checks how the network is responding and creates the best delay time for you.

Once SKANDA is done diagnosing the network, it will start port specific scan:

* Among the parameters selected in the base request (GUI selection), SKANDA attacks the server parameter by parameter.
* For every parameter it sends payloads targeted to all the ports.
* It does a detailed analysis of how the responses (error/time delay) received are different from each other, for different payloads.
* Accordingly it makes an informed decision whether the port is Closed, Open or Open(Blind XSPA).
* Initially SKANDA scans for the pre-defined list of important ports which are more probable to be used by the servers, to increase the chances of discovering an open port quickly.

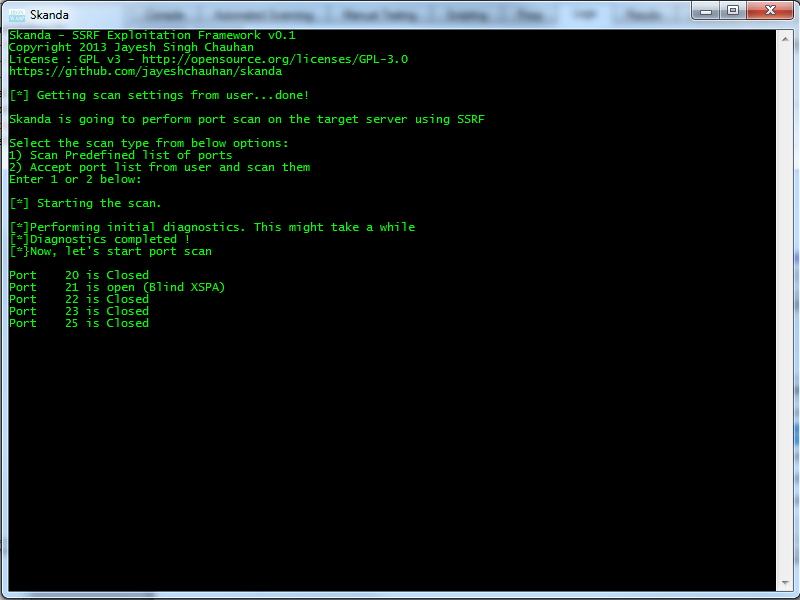


Figure 8 : Scanning important ports after diagnostics

Once done with the list of important ports. SKANDA tells you below details (as shown in Figure 9):

* If there is an open port found (Error Based XSPA/Blind XSPA).
* It gives you the list of ports which are scanned till now.
* Time taken to scan those ports.
* Now after important ports are scanned, it calculates the time which will be taken if all the ports are scanned, i.e., 1-65535.
* If the user wish to stop, he can press ‘n’ and the scan will stop.
* If the user enters any other key and hit enter, it carries forward the scan for all ports from 1-65535 (Figure 10).

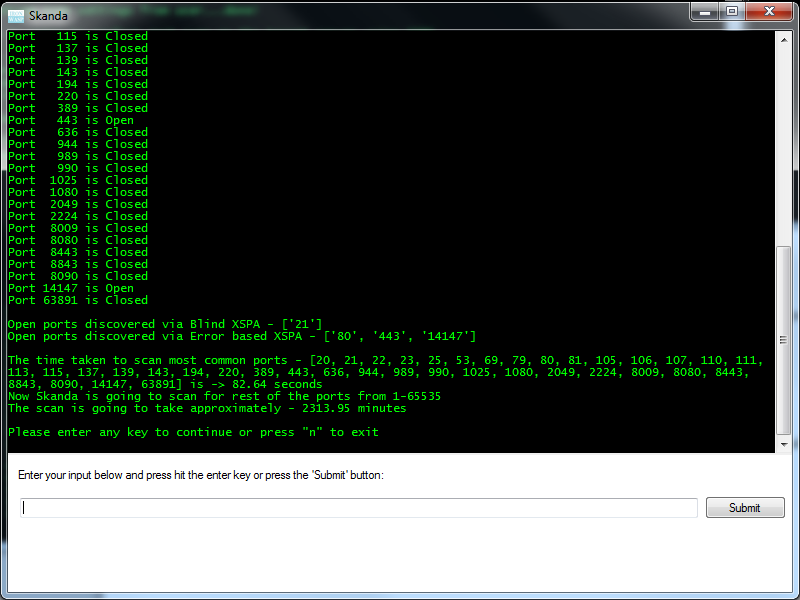


Figure 9 – After scanning the pre-defined list of important ports

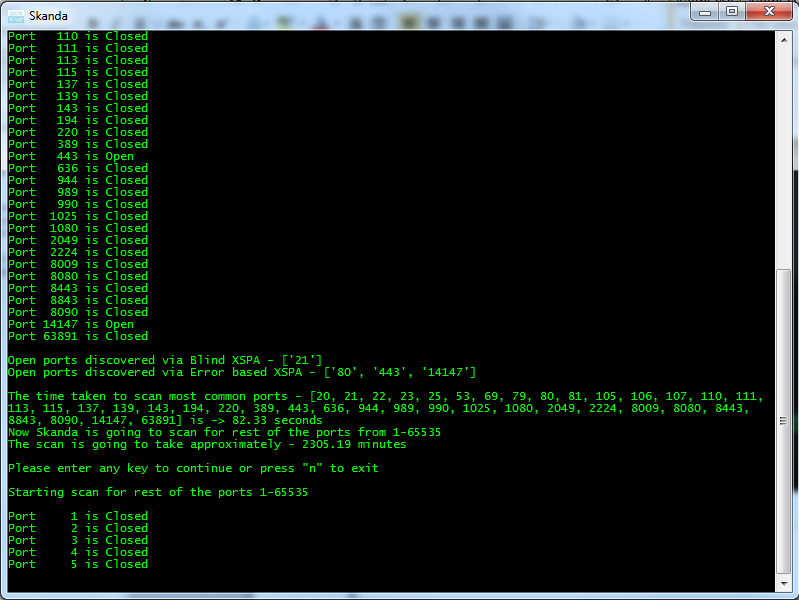


Figure 10 – Scanning started from port 1-65535

**Customized Scan (Remember the option ’2’ !):**

Now there are times where you don’t want to run the scan for all the ports, you want specific port to be scanned. SKANDA is customizable, you can enter the range of ports you want to scan and you are good to go.

Start SKANDA from the scratch (complete the steps from Figure 1 to Figure 6) and when you come to the window in Figure 10, select option ‘2’.

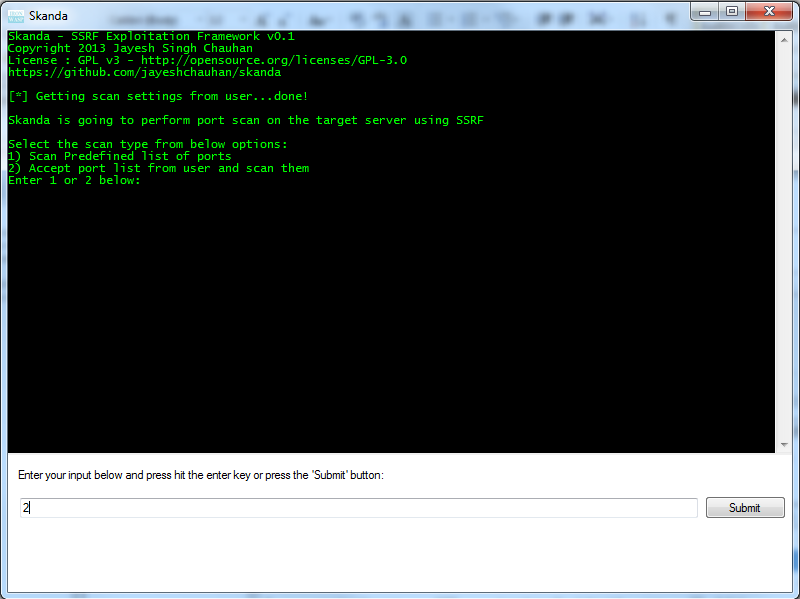


Figure 11 – Port Specific Scan

Now you will asked to enter the port range you want SKANDA to scan.

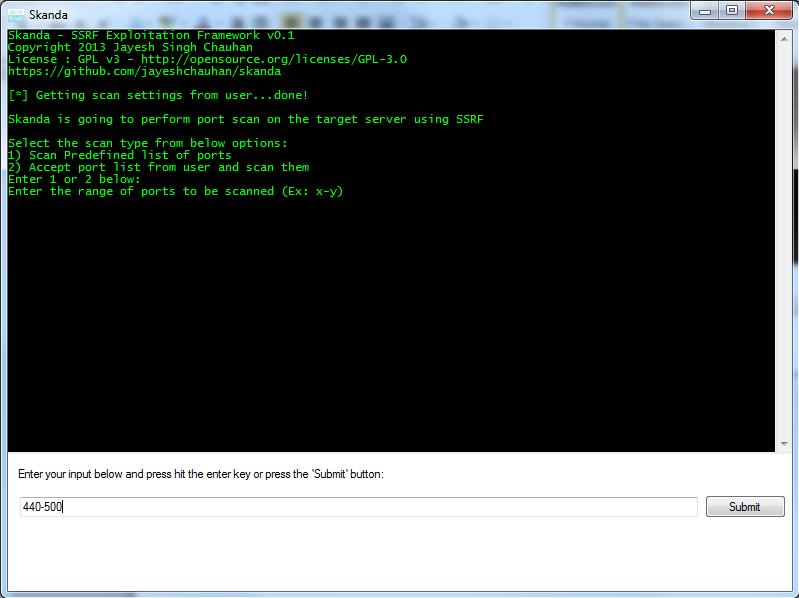


Figure 12 – Providing inputs for port specific scan

And SKANDA will attack those specific ports only.

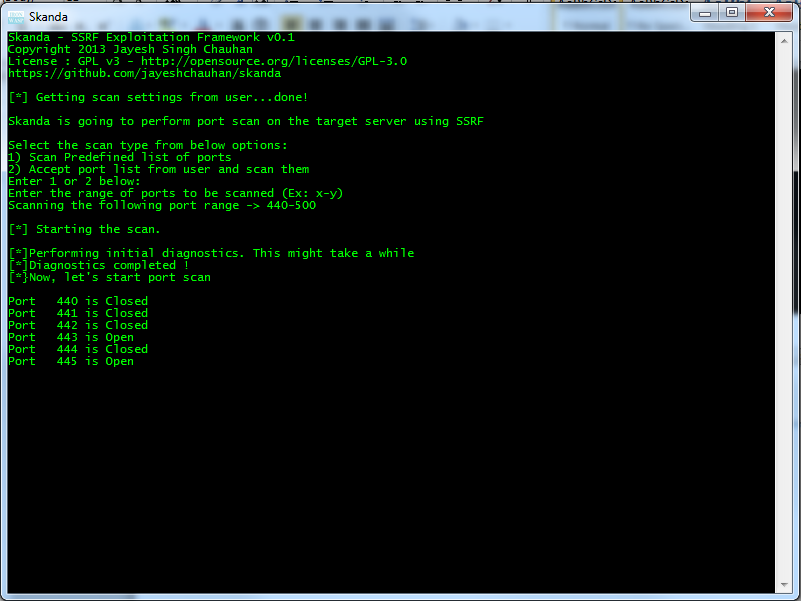


Figure 13 – Port specific scan in process

So, if you think your firewall is protecting you, think again!

If you encounter a suspicious parameter in a request, make sure you run SKANDA on it.

The current version of SKANDA (SSRF Exploitation Framework v0.1) can do a port scan. The future versions of SKANDA will be able to discover hosts behind the firewall, services running on those hosts and exploiting them accordingly.

Thank you all for reading.

Happy Scanning!!

Cheers!!