Sri Lanka Institute of Information Technology

Web Auditing

IE2062 – Web Security

Submitted by:

Student Registration Number	Student Name
IT19029146	Eranda H.P.D

Paypal.com

Table of Contents

Abstract

As the 2nd year 2nd semester Cyber Security undergraduates at SLIIT, under the module **Web Security (IE2062)**, we were asked to perform a web audit on a vulnerable web domain. Then we are supposed to create a complete audit report explaining the full process of auditing. So that the vulnerable web domain that I have selected for this assignment is **"www.paypal.com"**.

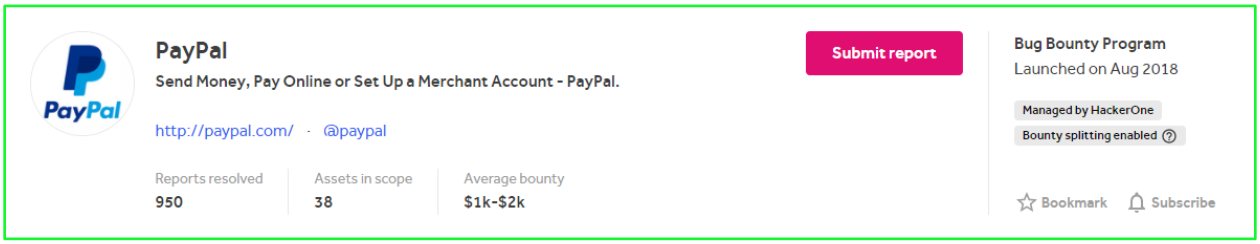
This audit report would not have been a success without the kind support and guidance of the lecturer in charge for **Web Security (IE2062)** module.

So first of all I would be very grateful to our lecturer in charge Dr. Lakmal Rupasinghe for giving me this wonderful opportunity to enhance my knowledge on Web Auditing. His immense kind and consistent support lead the pathway for me in choosing a better web domain for this assignment. He organized several webinars for us in order to educate us more about this particular assignment.

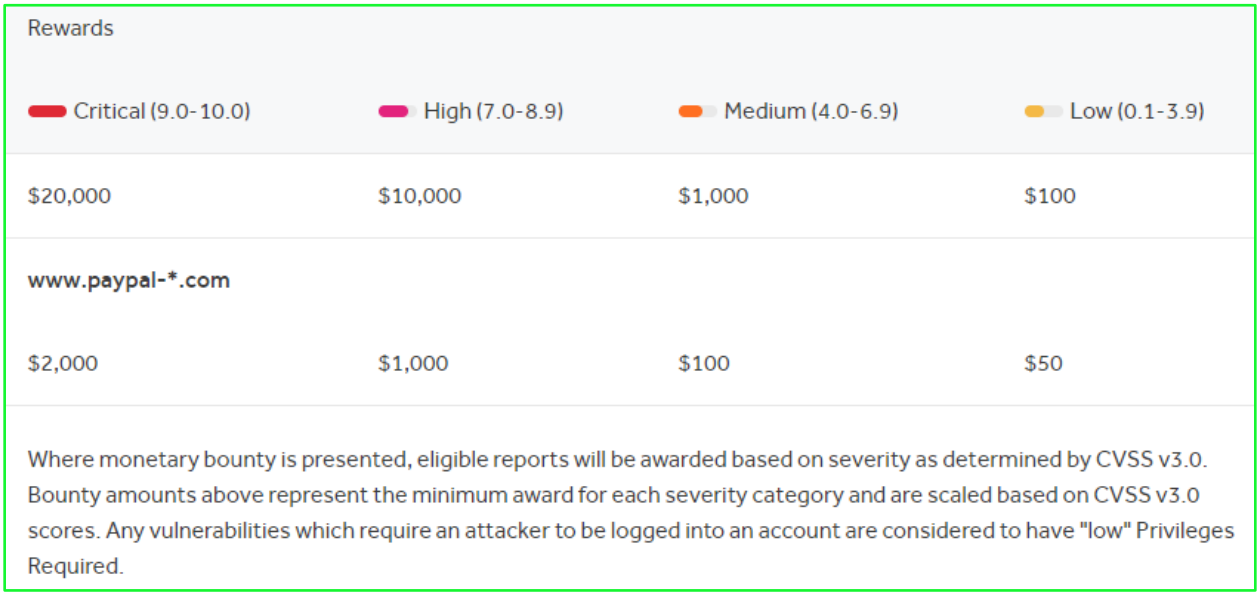
So once again I express my greatest gratitude to the Lecturer in Charge and associated lecturers and instructors for giving me suggestions and recommendations to improve this report.

Identifying a Web Domain for Auditing

So as I have mentioned previously, the web domain that I've selected for the web audit is **"www.paypal.com"**. In order to find this web domain, I surfed a lot through the **Hackerone's** bug bounty programs. Then I was able to find this particular **"PayPal"** web domain under the **"Directory"** tab in Hackerone.



Under this bug bounty program, we are able to find the following rewards offered by the "PayPal".



According to that program, the following PayPal brands are in scope.

The following PayPal brands are in scope:

- PayPal
- Venmo
- Xoom
- Braintree
- Swift Financial/ Loanbuilder

Brands and acquisitions not listed above are not in scope.

Now, let's move on to the **scope of the Web Application.**

It's very much important to read and understand the scope of the particular web application. Here you are allowed to perform vulnerability analysis only on **"In-Scope Vulnerabilities"**.

In-Scope Vulnerabilities

Accepted, in-scope vulnerabilities include, but are not limited to:

- Disclosure of sensitive or personally identifiable information
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF) for sensitive functions in a privileged context
- Server-side or remote code execution (RCE)
- Authentication or authorization flaws, including insecure direct object references and authentication bypass
- Injection vulnerabilities, including SQL and XML injection
- Directory traversal
- Significant security misconfiguration with a verifiable vulnerability
- Exposed credentials, disclosed by PayPal or its employees, that pose a valid risk to an in scope asset

The following facts are considered as **"Out-of-Scope"**. So you are not legally permitted to perform vulnerability analysis on such vulnerabilities.

Out-of-Scope Vulnerabilities

Certain vulnerabilities are considered out-of-scope for the Bug Bounty Program. Those out-of-scope vulnerabilities include, but are not limited to:

- Any physical attacks against PayPal property or data centers
- Reports that involve a secondary user account where an existing business relationship is being leveraged and the impact is limited solely to the parent account
- Username enumeration on customer facing systems (i.e. using server responses to determine whether a given account exists)
- Scanner output or scanner-generated reports, including any automated or active exploit tool
- Attacks involving payment fraud, theft, or malicious merchant accounts
- Man-in-the-Middle attacks
- Vulnerabilities involving stolen credentials or physical access to a device
- Social engineering attacks, including those targeting or impersonating internal employees by any means (e.g. customer service chat features, social media, personal domains, etc.)
- Vulnerabilities for which there are existing, documented controls (e.g. https://developer.paypal.com/docs/classic/paypal-payments-standard/integration-guide/encryptedwebpayments/)
- Open redirection, except in the following circumstances:
  - Clicking a PayPal-owned URL immediately results in a redirection
  - A redirection results in the loss of sensitive data (e.g. session tokens, PII, etc)

- Host header injections without a specific, demonstrable impact
- Denial of service (DOS) attacks using automated tools
- Self-XSS, which includes any payload entered by the victim
- Any vulnerabilities requiring significant and unlikely interaction by the victim, such as disabling browser controls
- Login/logout CSRF
- Content spoofing without embedding an external link or JavaScript
- Infrastructure vulnerabilities, including:
  - Issues related to SSL certificates
  - DNS configuration issues
  - Server configuration issues (e.g. open ports, TLS versions, etc.)
- Most vulnerabilities within our sandbox, lab, or staging environments, except Braintree.
- Vulnerabilities only affecting users of outdated or unpatched browsers and platforms
- Vulnerabilities that only affect one browser will be considered on a case-by-case basis, and may be closed as informative due to the reduced attack surface
- Information disclosure of public or non-protected information (e.g. code in a public repository, server banners, etc.), or information disclosed outside of PayPal's control (e.g. a personal, non-employee repository; a list from a previous infodump; etc.)
- Exposed credentials that are either no longer valid, or do not pose a risk to an in scope asset
- Any XSS that requires Flash. Flash is disabled by default in most modern browsers, thus greatly reducing the attack surface and associated risk.
- Any other submission determined to be low risk, based on unlikely or theoretical attack vectors, requiring significant user interaction, or resulting in minimal impact
- Vulnerabilities on third party libraries without showing specific impact to the target application (e.g. a CVE with no exploit)

In PayPal's bug bounty program, there is a special scope for Mobile Applications too.

Scope for Mobile Applications

In-Scope Vulnerabilities

In addition to in-scope items mentioned above, some additional vulnerability types will be considered in-scope for mobile applications. These include:

- Man-in-the-Middle attacks
- Attacks requiring physical access to a mobile device

Certain vulnerabilities with a working proof of concept on some of our Android mobile app(s) may qualify for an additional bounty through the Google Play Security Rewards Program. To see which apps and vulnerabilities may qualify for a bounty, please refer to the Google Play Security Rewards Program's Scope and Vulnerability Criteria.

Out-of-Scope Vulnerabilities

The following mobile vulnerabilities are out-of-scope and will not be accepted:

- Vulnerabilities requiring a rooted, jailbroken, or otherwise modified device
- Username enumeration on customer facing systems (i.e. using server responses to determine whether a given account exists)
- Vulnerabilities requiring extensive user interaction
- Exposure of non-sensitive data on the device
- Vulnerabilities on third party libraries without showing specific impact to the target application (e.g. a CVE with no exploit)

When submitting a bug report, there are several requirements that you need to fulfill.

Bug Submission Requirements

Required information

For all submissions, please include:

- Full description of the vulnerability being reported, including the exploitability and impact
- Evidence and explanation of all steps required to reproduce the submission, which may include:
- Videos
- Screenshots
- Exploit code
- Traffic logs
- Web/API requests and responses
- Email address or user ID of any test accounts
- IP address used during testing
- For RCE submissions, see below
- Failure to include any of the above items may delay or jeopardize the Bounty Payment

The following image shows brief summary about the **Ownership of Submissions, Termination and Confidentiality** of the PayPal's bug bounty program.

Ownership of Submissions

As a condition of participation in the PayPal Bug Bounty Program, you hereby grant PayPal, its subsidiaries, affiliates and customers a perpetual, irrevocable, worldwide, royalty-free, transferrable, sublicensable (through multiple tiers) and non-exclusive license to use, reproduce, adapt, modify, publish, distribute, publicly perform, create derivative work from, make, use, sell, offer for sale and import the Submission, as well as any materials submitted to PayPal in connection therewith, for any purpose. You should not send us any Submission that you do not wish to license to us.
You hereby represent and warrant that the Submission is original to you and you own all right, title and interest in and to the Submission. Further, you hereby waive all other claims of any nature, including express contract, implied-in-fact contract, or quasi-contract, arising out of any disclosure of the Submission to PayPal. In no event shall PayPal be precluded from discussing, reviewing, developing for itself, having developed, or developing for third parties, materials which are competitive with those set forth in the Submission irrespective of their similarity to the information in the Submission, so long as PayPal complies with the terms of participation stated herein.

Termination

In the event (i) you breach any of these Program Terms or the terms and conditions of the PayPal Agreements; or (ii) PayPal determines, in its sole discretion that your continued participation in the Bug Bounty Program could adversely impact PayPal (including, but not limited to, presenting any threat to PayPal's systems, security, finances and/or reputation) PayPal may immediately terminate your participation in the Bug Bounty Program and disqualify you from receiving any Bounty Payments. Please see our recommendations on the proper procedures for testing our applications.

Confidentiality

Any information you receive or collect about PayPal or any PayPal user through the Bug Bounty Program ("Confidential Information") must be kept confidential and only used in connection with the Bug Bounty Program. You may not use, disclose or distribute any such Confidential Information, including, but not limited to, any information regarding your Submission and information you obtain when researching the PayPal sites, without PayPal's prior written consent.

Following images represent the sub domains which are within the Scope.

Scopes			
In Scope			
Domain	**www.paypal-*.com** PayPal's Partner Sites (www.paypal-__.com) are mainly marketing based sites that are not part of the core PayPal customer domains (.paypal.com) and are managed by hosting vendor companies. They have variable timelines and are often decommissioned. A listing of these sites designated for deprecation will not be publically maintained due to frequent changes. When researching bugs on these sites, please keep this in mind as bug Submissions for sites on schedule for deprecation will not be honored. Submissions of bugs relating to services or domains not referenced above or for sites on schedule for deprecation are ineligible for the Bug Bounty Program and will not be eligible for a Bounty Payment.	Critical	Eligible
Domain	***.xoom.com**	Critical	Eligible
Domain	***.paypal.com**	Critical	Eligible

Domain	***.braintreegateway.com** For testing and account creation, please use *.sandbox.braintreegateway.com rather than production.	Critical	Eligible
Domain	***.paydiant.com**	Critical	Eligible
Domain	***.venmo.com**	Critical	Eligible
Domain	**paypalobjects.com**	Critical	Eligible
Domain	**paypal.me**	Critical	Eligible
Domain	**py.pl**	Critical	Eligible
Domain	***.braintreepayments.com** For testing and account creation, please use *.sand.braintreepayments.com rather than production.	Critical	Eligible
Domain	***.braintree-api.com** For testing and account creation, please use *.sandbox.braintree-api.com rather than production.	Critical	Eligible

Domain	***.braintree.tools** Please note, this is a development environment that is constantly in flux. Accordingly, vulnerabilities found on this asset will generally have lower impact and payouts.	Critical	Eligible
Domain	**prequal.swiftfinancial.com** We are aware that the root URL of this domain returns an error, the API is functioning correctly.	Critical	Eligible
Domain	**partner.swiftfinancial.com** We are aware that the root URL of this domain returns an error, the API is functioning correctly.	Critical	Eligible
Domain	**decision.swiftfinancial.com** We are aware that the root URL of this domain returns an error, the API is functioning correctly.	Critical	Eligible
Domain	**pigeon.swiftfinancial.com** We are aware that the root URL of this domain returns an error, the API is functioning correctly.	Critical	Eligible

Domain	**scrutiny.swiftfinancial.com** We are aware that the root URL of this domain returns an error, the API is functioning correctly.	Critical	Eligible
Domain	**www.swiftcapital.com**	Critical	Eligible
Domain	**www.loanbuilder.com**	Critical	Eligible
Domain	**www.swiftfinancial.com**	Critical	Eligible
Domain	**api.swiftfinancial.com**	Critical	Eligible
Domain	**my.swiftfinancial.com**	Critical	Eligible
Domain	**api.loanbuilder.com**	Critical	Eligible
Domain	**my.loanbuilder.com**	Critical	Eligible
Domain	**loanbuilder.com**	Critical	Eligible

Domain	**swiftfinancial.com**	Critical	Eligible
Domain	**swiftcapital.com**	Critical	Eligible
Android: Play Store	**com.xoom.android.app**	Critical	Eligible
Android: Play Store	**com.paypal.merchant.client**	Critical	Eligible
Android: Play Store	**com.paypal.android.p2pmobile**	Critical	Eligible
Android: Play Store	**com.venmo**	Critical	Eligible
Android: Play Store	**com.paypal.here**	Critical	Eligible
iOS: App Store	**com.paypal.herehd**	Critical	Eligible

iOS: App Store	**net.kortina.labs.Venmo**	Critical	Eligible
iOS: App Store	**com.xoom.app**	Critical	Eligible
iOS: App Store	**com.yourcompany.PPClient**	Critical	Eligible
iOS: App Store	**com.paypal.here**	Critical	Eligible
iOS: App Store	**com.paypal.merchant**	Critical	Eligible

Scanning for Sub-Domains

In order to scan sub-domains of a particular web domain, we use the tool **"Sublist3r"** in Kali Linux.

So, in here first of all you need to clone the Sublist3r repository from the GitHub.

```
root@kali:/home/dilshan# git clone https://github.com/aboul3la/Sublist3r.git
Cloning into 'Sublist3r' ...
remote: Enumerating objects: 383, done.
remote: Total 383 (delta 0), reused 0 (delta 0), pack-reused 383
Receiving objects: 100% (383/383), 1.12 MiB | 211.00 KiB/s, done.
Resolving deltas: 100% (212/212), done.
```

Then you need to perform a requirement checking while installing the Sublist3r.

```
root@kali:/home/dilshan/Sublist3r# pip3 install -r requirements.txt
Collecting argparse
  Downloading argparse-1.4.0-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: dnspython in /usr/lib/python3/dist-packages (from -r requirements.txt (line 2)) (1.16.0)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from -r requirements.txt (line 3)) (2.23.0)
Installing collected packages: argparse
Successfully installed argparse-1.4.0
```

So after the installation, you can run the tool without any issue.

```
root@kali:/home/dilshan/Sublist3r# ./sublist3r.py

                 ____        _     _ _     _   _____
                / ___| _   _| |__ | (_)___| |_|___ / _ __
                \___ \| | | | '_ \| | / __| __| |_ \| '__|
                 ___) | |_| | |_) | | \__ \ |_ ___) | |
                |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

Usage: python ./sublist3r.py [Options] use -h for help
Error: argument -d/--domain is required
root@kali:/home/dilshan/Sublist3r#
```

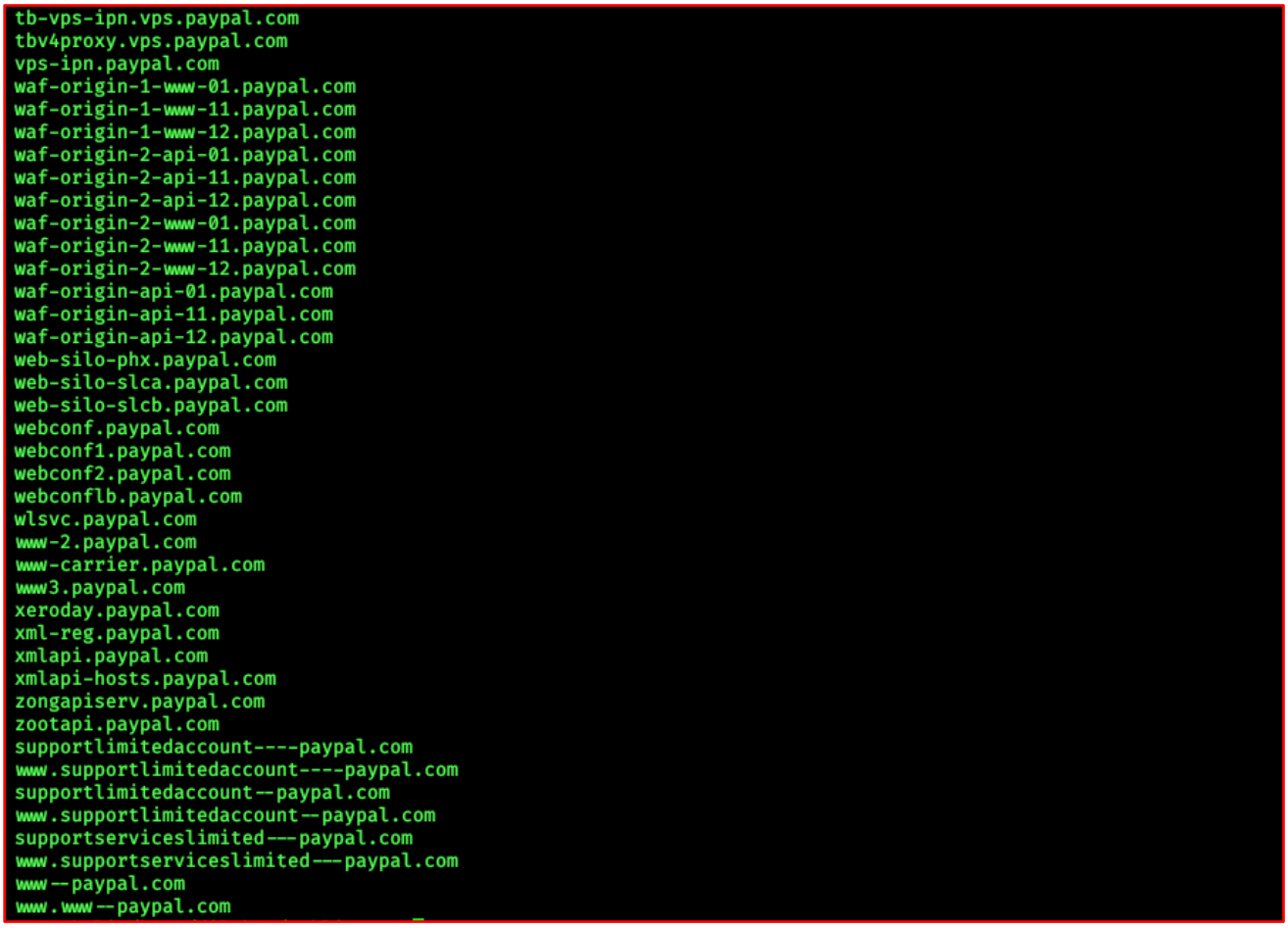
Here, you can get an idea about all the commands that can be used, by simply typing as **"./sublist3r.py --help"**.

```
root@kali:/home/dilshan/Sublist3r# ./sublist3r.py --help
usage: sublist3r.py [-h] -d DOMAIN [-b [BRUTEFORCE]] [-p PORTS] [-v [VERBOSE]]
                    [-t THREADS] [-e ENGINES] [-o OUTPUT] [-n]

OPTIONS:
  -h, --help            show this help message and exit
  -d DOMAIN, --domain DOMAIN
                        Domain name to enumerate it's subdomains
  -b [BRUTEFORCE], --bruteforce [BRUTEFORCE]
                        Enable the subbrute bruteforce module
  -p PORTS, --ports PORTS
                        Scan the found subdomains against specified tcp ports
  -v [VERBOSE], --verbose [VERBOSE]
                        Enable Verbosity and display results in realtime
  -t THREADS, --threads THREADS
                        Number of threads to use for subbrute bruteforce
  -e ENGINES, --engines ENGINES
                        Specify a comma-separated list of search engines
  -o OUTPUT, --output OUTPUT
                        Save the results to text file
  -n, --no-color        Output without color

Example: python ./sublist3r.py -d google.com
```

So, by providing the relevant web domain, you are able to find out the sub domains of that particular web domain.

```
root@kali:/home/dilshan/Sublist3r# ./sublist3r.py -d paypal.com

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for paypal.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
```

So by using this technique, I was able to find out huge number of sub-domains for **“Paypal.com”**. Following screenshot displays some of those sub-domains.



By using the **“host”** tool in Kali Linux, you are able to find out the existing domain name servers of a particular web domain. Here we are using the technique called as **“Zone Transferring”.**

```
root@kali:/home/dilshan/Sublist3r# host -t ns paypal.com
paypal.com name server ns1.p57.dynect.net.
paypal.com name server pdns100.ultradns.com.
paypal.com name server ns2.p57.dynect.net.
paypal.com name server pdns100.ultradns.net.
```

Then by using the **“DNS Lookup Utility (dig)”** tool, you are able to find out lot more information regarding the PayPal’s DNS servers.

```
root@kali:/home/dilshan/Sublist3r# dig paypal.com

; <<>> DiG 9.11.14-3-Debian <<>> paypal.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51271
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 6

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 0b0700094619f58847ca37745f782382e7de7b4e85ad15b7 (good)
;; QUESTION SECTION:
;paypal.com.                    IN      A

;; ANSWER SECTION:
paypal.com.             300     IN      A       64.4.250.37
paypal.com.             300     IN      A       64.4.250.36

;; AUTHORITY SECTION:
paypal.com.             300     IN      NS      ns1.p57.dynect.net.
paypal.com.             300     IN      NS      pdns100.ultradns.com.
paypal.com.             300     IN      NS      ns2.p57.dynect.net.
paypal.com.             300     IN      NS      pdns100.ultradns.net.

;; ADDITIONAL SECTION:
ns1.p57.dynect.net.     50157   IN      A       208.78.70.57
ns2.p57.dynect.net.     50157   IN      A       204.13.250.57
pdns100.ultradns.com.   163250  IN      A       156.154.64.100
pdns100.ultradns.net.   63885   IN      A       156.154.65.100
pdns100.ultradns.com.   163250  IN      AAAA    2001:502:f3ff::88

;; Query time: 80 msec
;; SERVER: 202.129.232.233#53(202.129.232.233)
;; WHEN: Sat Oct 03 12:31:47 +0530 2020
;; MSG SIZE  rcvd: 303
```

Then by using the tool called, **“dnsenum”,** you are able to enumerate information on a web domain and to discover non-contiguous IP Blocks.

```
root@kali:/home/dilshan/Sublist3r# dnsenum paypal.com
dnsenum VERSION:1.2.6

-----   paypal.com   -----


Host's addresses:
__________________

paypal.com.                              300      IN    A        64.4.250.37
paypal.com.                              300      IN    A        64.4.250.36


Name Servers:
______________

ns1.p57.dynect.net.                      49163    IN    A        208.78.70.57
ns2.p57.dynect.net.                      49163    IN    A        204.13.250.57
pdns100.ultradns.com.                    3230     IN    A        156.154.64.100
pdns100.ultradns.net.                    62891    IN    A        156.154.65.100


Mail (MX) Servers:
___________________

mx1.paypalcorp.com.                      3600     IN    A        173.224.165.17
mx2.paypalcorp.com.                      3600     IN    A        173.224.161.141


Trying Zone Transfers and getting Bind Versions:
_________________________________________________


Trying Zone Transfer for paypal.com on ns1.p57.dynect.net ...
AXFR record query failed: REFUSED

Trying Zone Transfer for paypal.com on ns2.p57.dynect.net ...
AXFR record query failed: REFUSED

Trying Zone Transfer for paypal.com on pdns100.ultradns.net ...
AXFR record query failed: REFUSED
```

```
Brute forcing with /usr/share/dnsenum/dns.txt:
_______________________________________________

a.paypal.com.                           300      IN    A        66.211.168.97
a.paypal.com.                           300      IN    A        173.0.85.51
a.paypal.com.                           300      IN    A        173.0.89.51
accounts.paypal.com.                    3600     IN    CNAME    reports.paypal.com.
reports.paypal.com.                     300      IN    A        173.0.88.139
admin.paypal.com.                       300      IN    A        173.0.88.10
c.paypal.com.                           185      IN    CNAME    c.glb.paypal.com.
c.glb.paypal.com.                       190      IN    CNAME    c.paypal.com-a.edgekey.net.
c.paypal.com-a.edgekey.net.             7098     IN    CNAME    e9935.x.akamaiedge.net.
e9935.x.akamaiedge.net.                 12       IN    A        23.13.127.55
mail.paypal.com.                        3600     IN    A        159.127.187.12
mobile.paypal.com.                      300      IN    CNAME    mobile.paypal.com.edgekey.net.
mobile.paypal.com.edgekey.net.          180      IN    CNAME    e1038.a1.akamaiedge.net.
e1038.a1.akamaiedge.net.                360      IN    A        104.111.160.24
mx.paypal.com.                          3600     IN    A        10.190.3.55
ns1.paypal.com.                         3600     IN    A        64.4.244.70
ns2.paypal.com.                         3600     IN    A        64.4.244.71
secure.paypal.com.                      300      IN    A        173.0.88.39
secure.paypal.com.                      300      IN    A        173.0.88.7
secure.paypal.com.                      300      IN    A        173.0.92.14
secure.paypal.com.                      300      IN    A        173.0.84.39
smtp.paypal.com.                        3600     IN    CNAME    smtp-in.paypal.com.
smtp-in.paypal.com.                     3600     IN    A        64.4.244.68
training.paypal.com.                    3600     IN    CNAME    www.training.paypal.com.
www.training.paypal.com.                3600     IN    CNAME    endymion.paypal.com.
endymion.paypal.com.                    3600     IN    A        64.4.254.252
www.paypal.com.                         2484     IN    CNAME    www.glb.paypal.com.
www.glb.paypal.com.                     125      IN    CNAME    www-fastly.glb.paypal.com.
www-fastly.glb.paypal.com.              163      IN    A        151.101.129.21
```

```
paypal.com class C netranges:
_____________________________

 64.4.244.0/24
 64.4.250.0/24
 64.4.254.0/24
 66.211.168.0/24
 151.101.129.0/24
 159.127.187.0/24
 173.0.84.0/24
 173.0.85.0/24
 173.0.88.0/24
 173.0.89.0/24
 173.0.92.0/24


Performing reverse lookup on 2816 ip addresses:
_______________________________________________

42.250.4.64.in-addr.arpa.               300      IN    PTR      stage-api.edge.paypal.com.
44.250.4.64.in-addr.arpa.               300      IN    PTR      stage-api-m.edge.paypal.com.
45.250.4.64.in-addr.arpa.               300      IN    PTR      stage-www.edge.paypal.com.
0.254.4.64.in-addr.arpa.                3600     IN    PTR      (
1.254.4.64.in-addr.arpa.                3600     IN    PTR      node-64-4-254-1.networks.paypal.com.
2.254.4.64.in-addr.arpa.                3600     IN    PTR      node-64-4-254-2.networks.paypal.com.
3.254.4.64.in-addr.arpa.                3600     IN    PTR      node-64-4-254-3.networks.paypal.com.
4.254.4.64.in-addr.arpa.                3600     IN    PTR      node-64-4-254-4.networks.paypal.com.
5.254.4.64.in-addr.arpa.                3600     IN    PTR      node-64-4-254-5.networks.paypal.com.
6.254.4.64.in-addr.arpa.                3600     IN    PTR      node-64-4-254-6.networks.paypal.com.
7.254.4.64.in-addr.arpa.                3600     IN    PTR      node-64-4-254-7.networks.paypal.com.
8.254.4.64.in-addr.arpa.                3600     IN    PTR      node-64-4-254-8.networks.paypal.com.
9.254.4.64.in-addr.arpa.                3600     IN    PTR      node-64-4-254-9.networks.paypal.com.
10.254.4.64.in-addr.arpa.               3600     IN    PTR      node-64-4-254-10.networks.paypal.com.
11.254.4.64.in-addr.arpa.               3600     IN    PTR      node-64-4-254-11.networks.paypal.com.
12.254.4.64.in-addr.arpa.               3600     IN    PTR      node-64-4-254-12.networks.paypal.com.
13.254.4.64.in-addr.arpa.               3600     IN    PTR      node-64-4-254-13.networks.paypal.com.
14.254.4.64.in-addr.arpa.               3600     IN    PTR      node-64-4-254-14.networks.paypal.com.
15.254.4.64.in-addr.arpa.               3600     IN    PTR      node-64-4-254-15.networks.paypal.com.
16.254.4.64.in-addr.arpa.               3600     IN    PTR      node-64-4-254-16.networks.paypal.com.
17.254.4.64.in-addr.arpa.               3600     IN    PTR      node-64-4-254-17.networks.paypal.com.
18.254.4.64.in-addr.arpa.               3600     IN    PTR      node-64-4-254-18.networks.paypal.com.
```

Vulnerability Discovery and Analysis

In the field of Cyber Security, there are considerable amount of tools which can be used to scan web vulnerabilities.

1) Nikto

"Nikto" is a great tool which can be used to perform vulnerability analysis.

```
root@kali:/home/dilshan# nikto --help
Unknown option: help

        -config+            Use this config file
        -Display+           Turn on/off display outputs
        -dbcheck            check database and other key files for syntax errors
        -Format+            save file (-o) format
        -Help               Extended help information
        -host+              target host/URL
        -id+                Host authentication to use, format is id:pass or id:pass:realm
        -list-plugins       List all available plugins
        -output+            Write output to this file
        -nossl              Disables using SSL
        -no404              Disables 404 checks
        -Plugins+           List of plugins to run (default: ALL)
        -port+              Port to use (default 80)
        -root+              Prepend root value to all requests, format is /directory
        -ssl                Force ssl mode on port
        -Tuning+            Scan tuning
        -timeout+           Timeout for requests (default 10 seconds)
        -update             Update databases and plugins from CIRT.net
        -Version            Print plugin and database versions
        -vhost+             Virtual host (for Host header)
                + requires a value

         Note: This is the short help output. Use -H for full help text.
```

So by providing the **target Hostname** and the **port number**, we are able to do a vulnerability analysis on the particular web domain.

```
root@kali:/home/dilshan# nikto -h paypal.com -p 80
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          64.4.250.37
+ Target Hostname:    paypal.com
+ Target Port:        80
+ Message:           Multiple IP addresses found: 64.4.250.37, 64.4.250.36
+ Start Time:        2020-10-04 10:30:35 (GMT5.5)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: https://www.paypal.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ ERROR: Error limit (20) reached for host, giving up. Last error: opening stream: can't connect (timeout): Operation now in progress
+ Scan terminated:  19 error(s) and 3 item(s) reported on remote host
+ End Time:          2020-10-04 10:42:17 (GMT5.5) (702 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

2) Nessus

“Nessus” is a very powerful web penetration testing tool which is mostly used by professional pen testers.

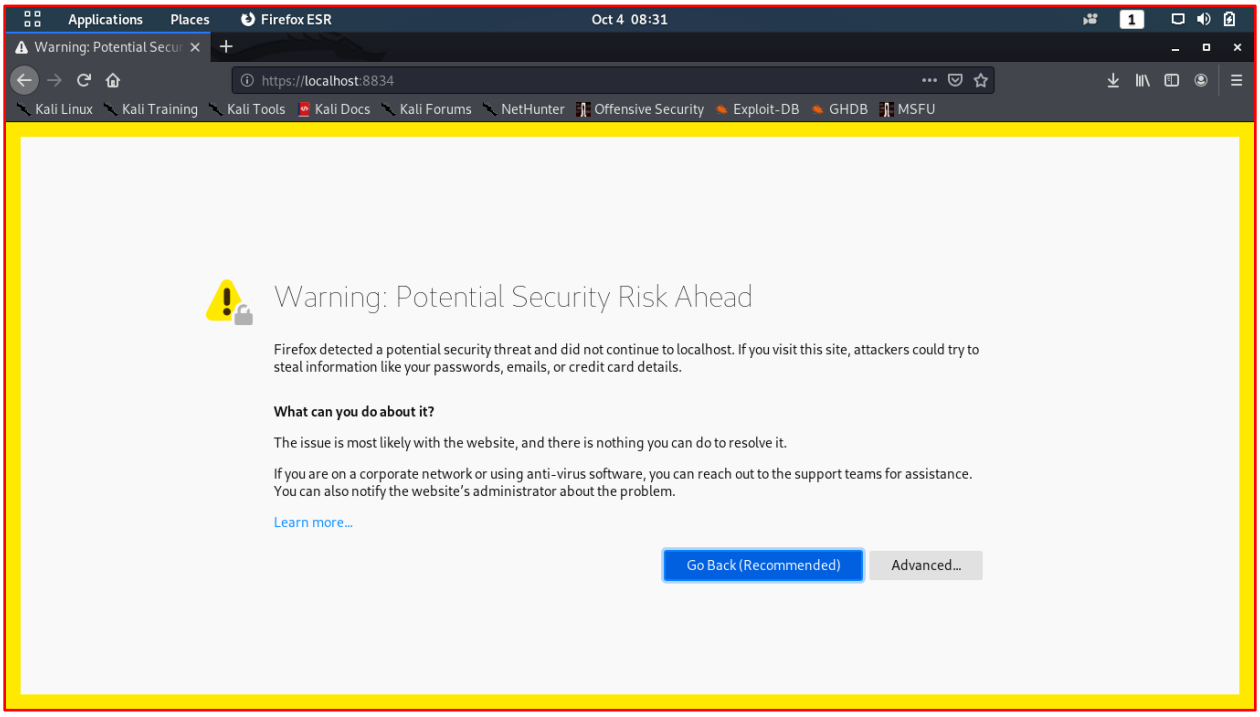
You can install Nessus by providing the following command.

```
root@kali:/home/dilshan/Downloads# dpkg -i Nessus-8.11.1-debian6_amd64.deb
Selecting previously unselected package nessus.
(Reading database ... 531756 files and directories currently installed.)
Preparing to unpack Nessus-8.11.1-debian6_amd64.deb ...
Unpacking nessus (8.11.1) ...
Setting up nessus (8.11.1) ...
Unpacking Nessus Scanner Core Components...

 - You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
 - Then go to https://kali:8834/ to configure your scanner
```

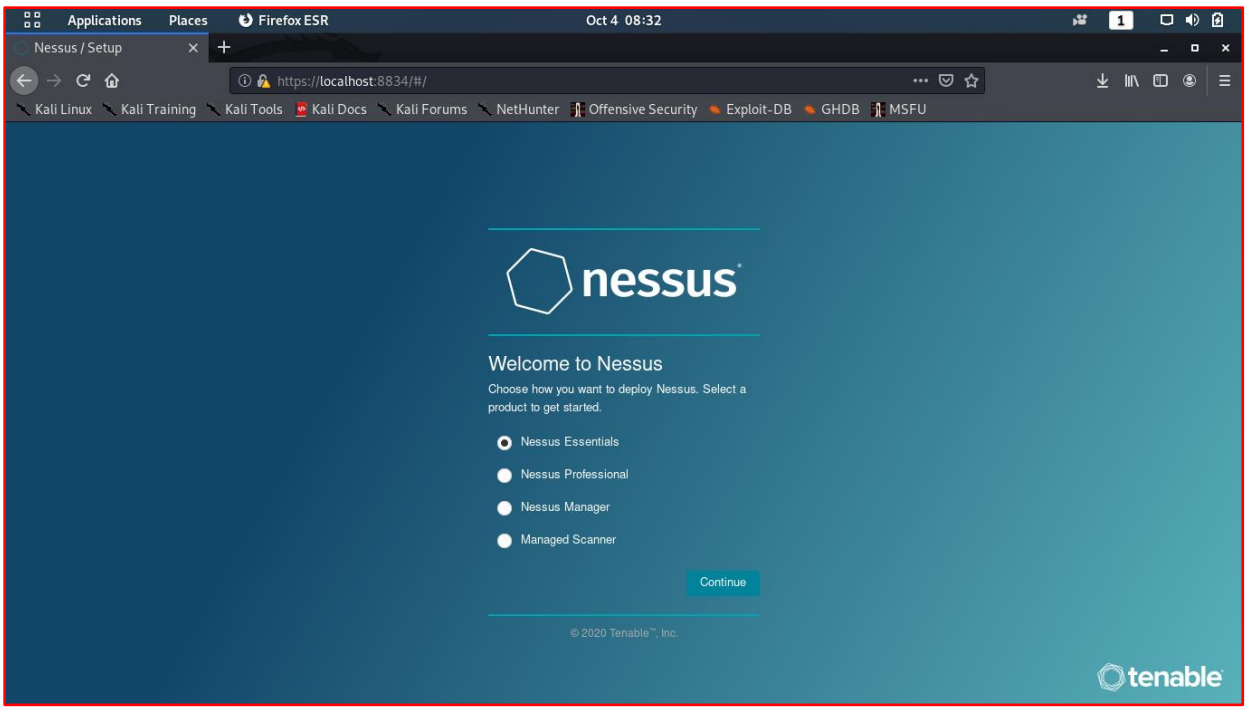
So in the terminal, you need to provide the following command in order to start the **Nessus** service.

```
root@kali:/home/dilshan# service nessusd start
```

After that in your default Web Browser, you need to provide the URL as **https://localhost:8834**.



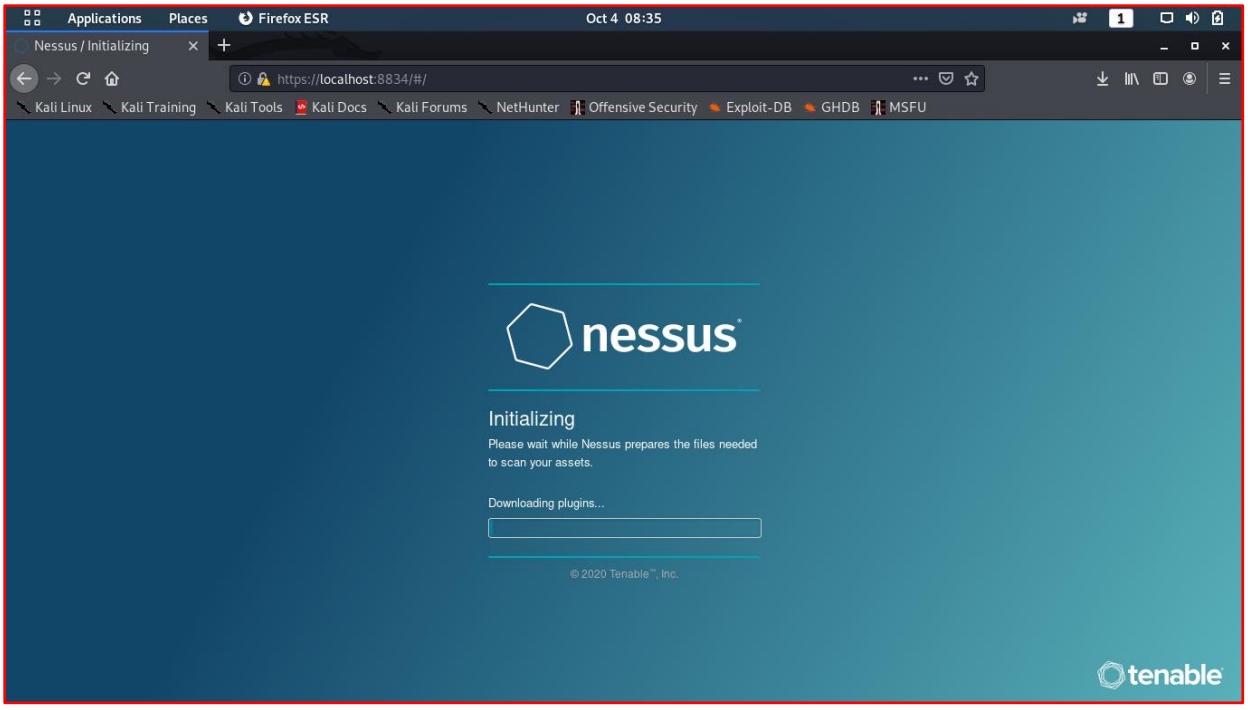
In this stage, you need to go to the **advanced** tab, and then give the permissions to open this website.



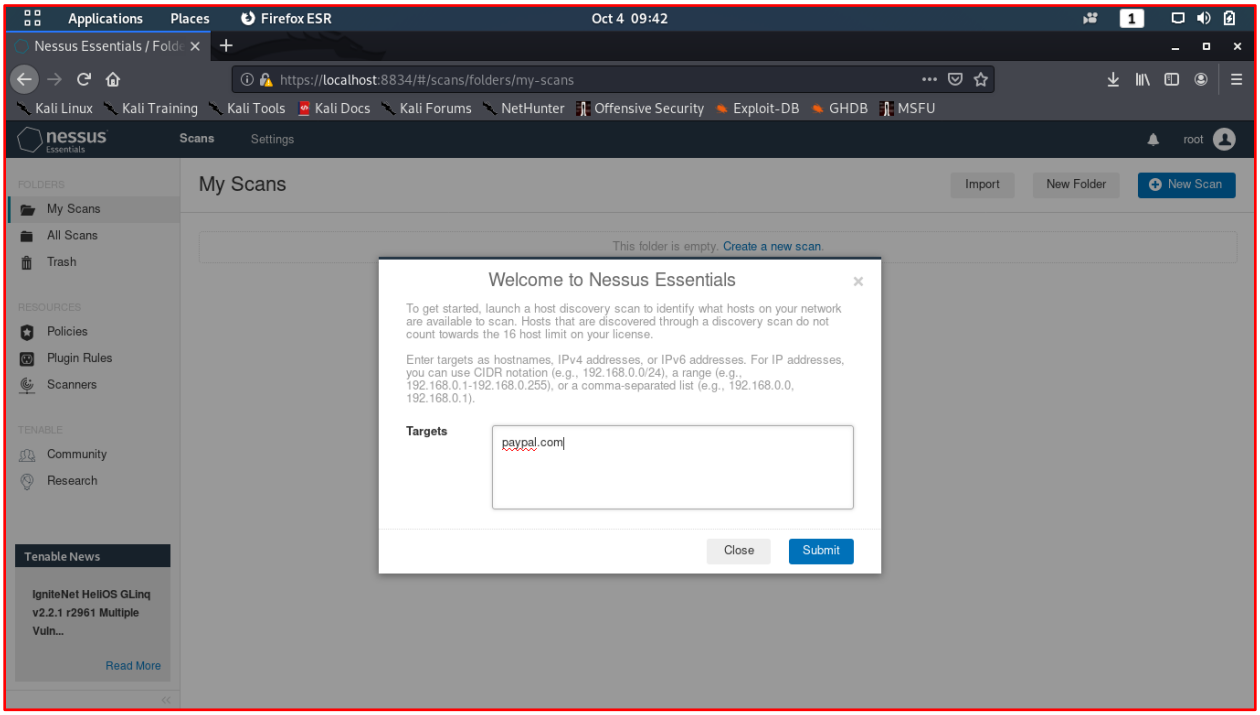
Except the **"Nessus Essentials",** all the other products are paid professional versions of Nessus.

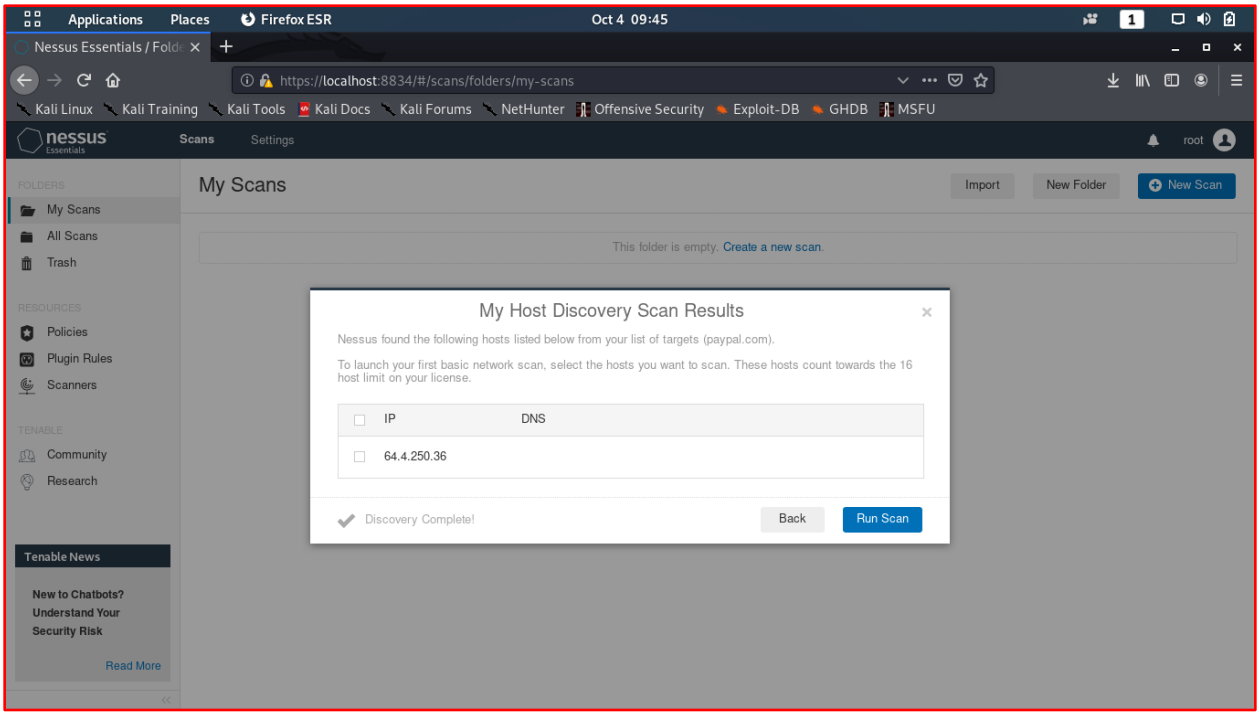
So after choosing the product, it's going to automatically update the **signatures database.**

In this process it downloads the necessary plugins and compiles them together.

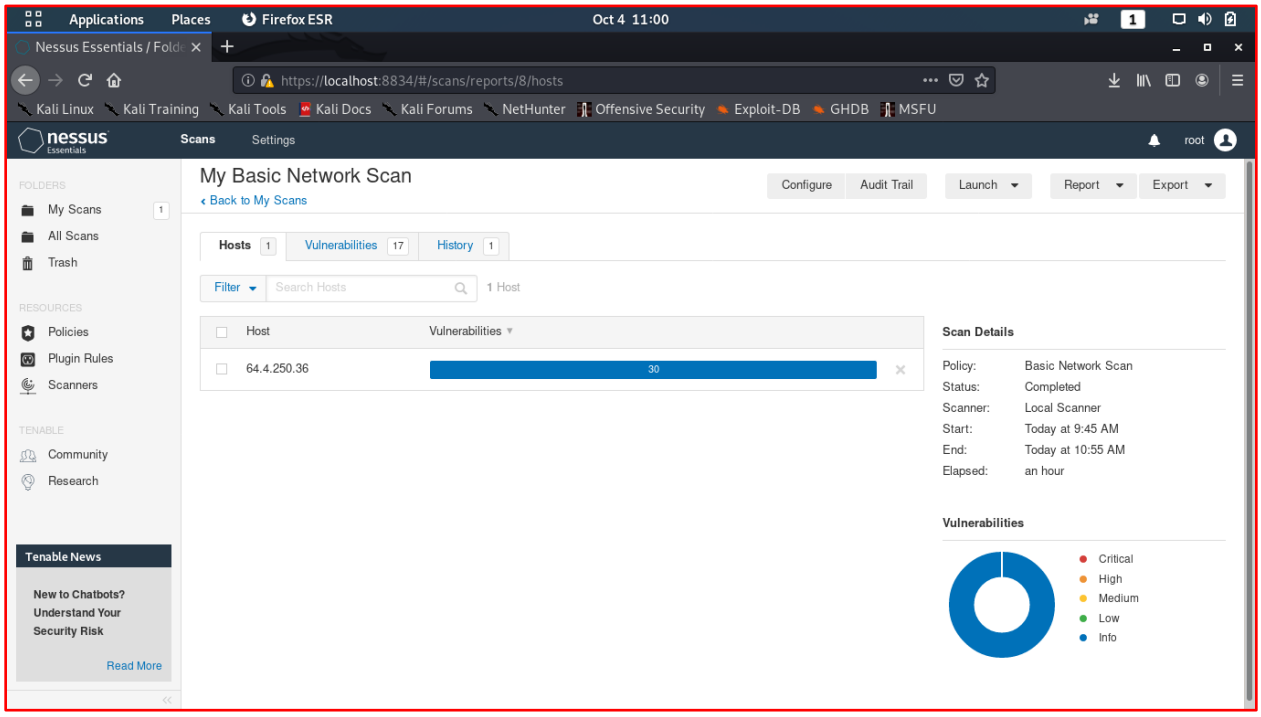


After updating the signature database, you need to provide the targets that you want to scan. So in this occasion I'm going to use **"paypal.com"** as my target.

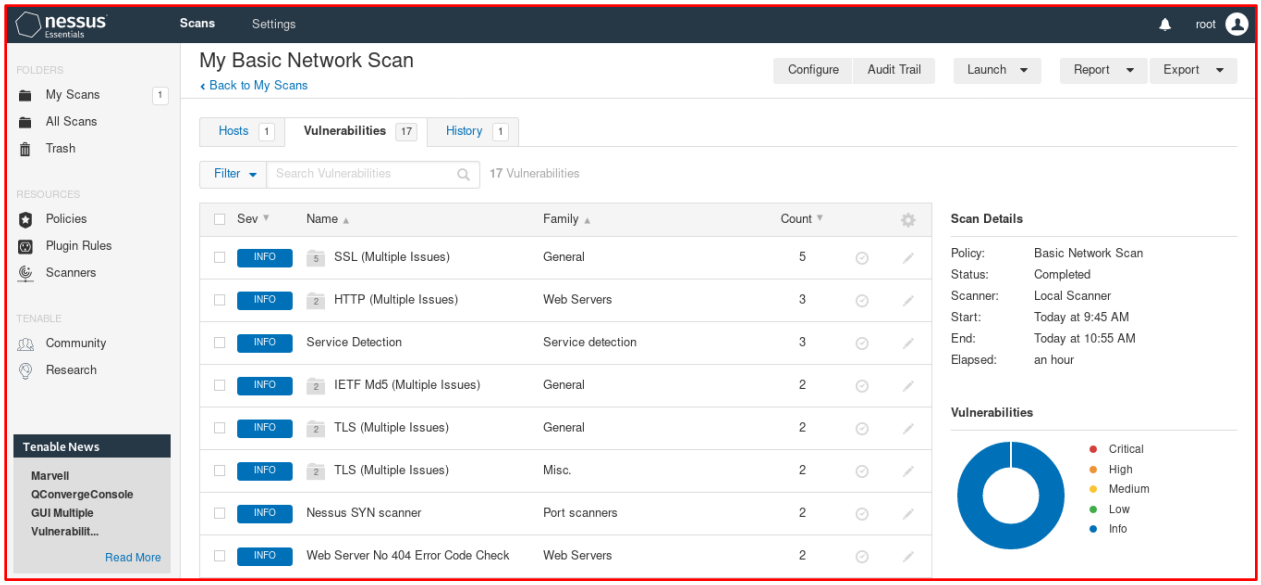


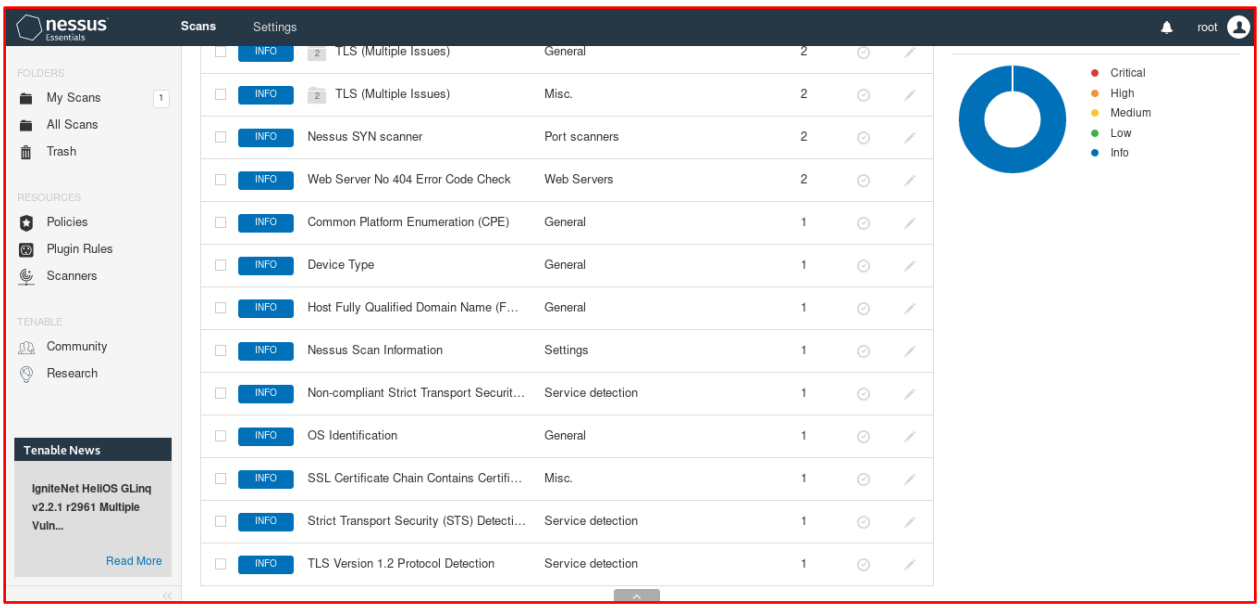


So the following images show the results of **My Basic Network Scan** in **“paypal.com”.**



So with the help of **“Nessus”** tool, we were able to find out 17 vulnerabilities in **“paypal.com”.**





NOTE :-

Although the Basic Network Scanning was done without any issues, the final report doesn't display the severity of those vulnerabilities.

Only the "Information" regarding the vulnerability, is displayed as the output in this report.

3) OpenVAS

“Open Vulnerability Assessment System (openVAS)” is also a free and open-source web penetration tool just like **“Nessus”**. But unlike in **“Nessus”**, although we use the community version of **“OpenVAS”** in this occasion, we get the opportunity to experience almost all the features in the paid version.

So before starting to work with the tool **“OpenVAS”** directly, first you need to do the **initial configuration** of the tool.

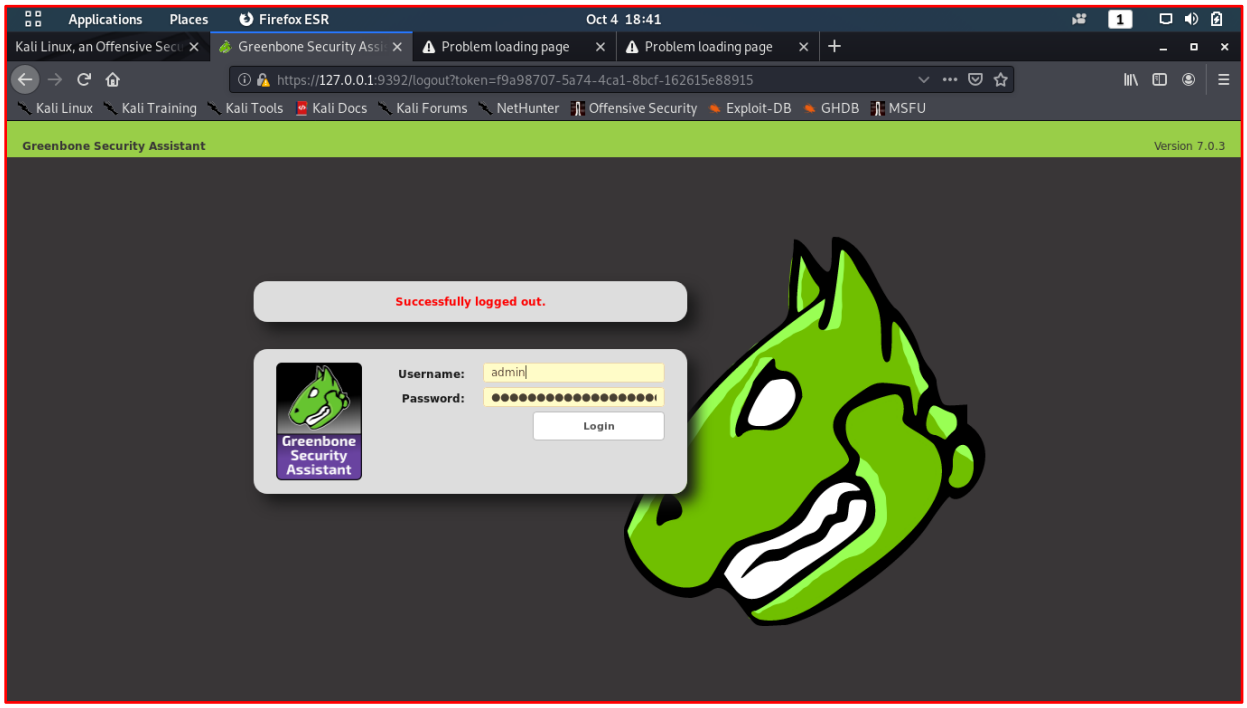
```
root@kali:/home/dilshan# openvas-setup
```

At the end of the configuration process, it generates the **username** and **password** for the OpenVAS account.

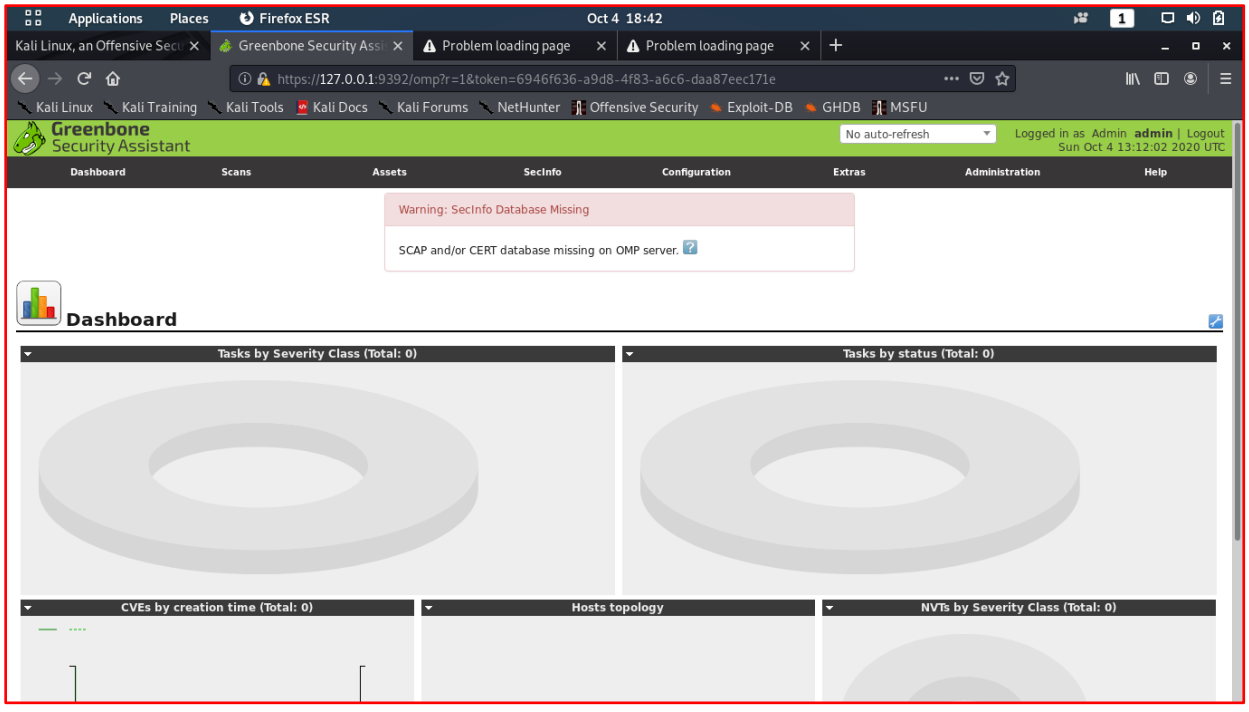
```
Oct 04 13:31:34 kali systemd[1]: Starting Open Vulnerability Assessment System Manager Daemon...
Oct 04 13:31:34 kali systemd[1]: openvas-manager.service: Can't open PID file /run/openvasmd.pid (yet?) after start: Operation not permitted
Oct 04 13:31:34 kali systemd[1]: Started Open Vulnerability Assessment System Manager Daemon.

[*] Opening Web UI (https://127.0.0.1:9392) in: 5 ... 4 ... 3 ... 2 ... 1 ...

[>] Checking for admin user
[*] Creating admin user
User created with password 'fb53d34d-6b1c-4911-be8b-46aa49893b54'.

[+] Done
```

After that in your default Web Browser, you need to provide the URL as **https://127.0.0.1:9392**.

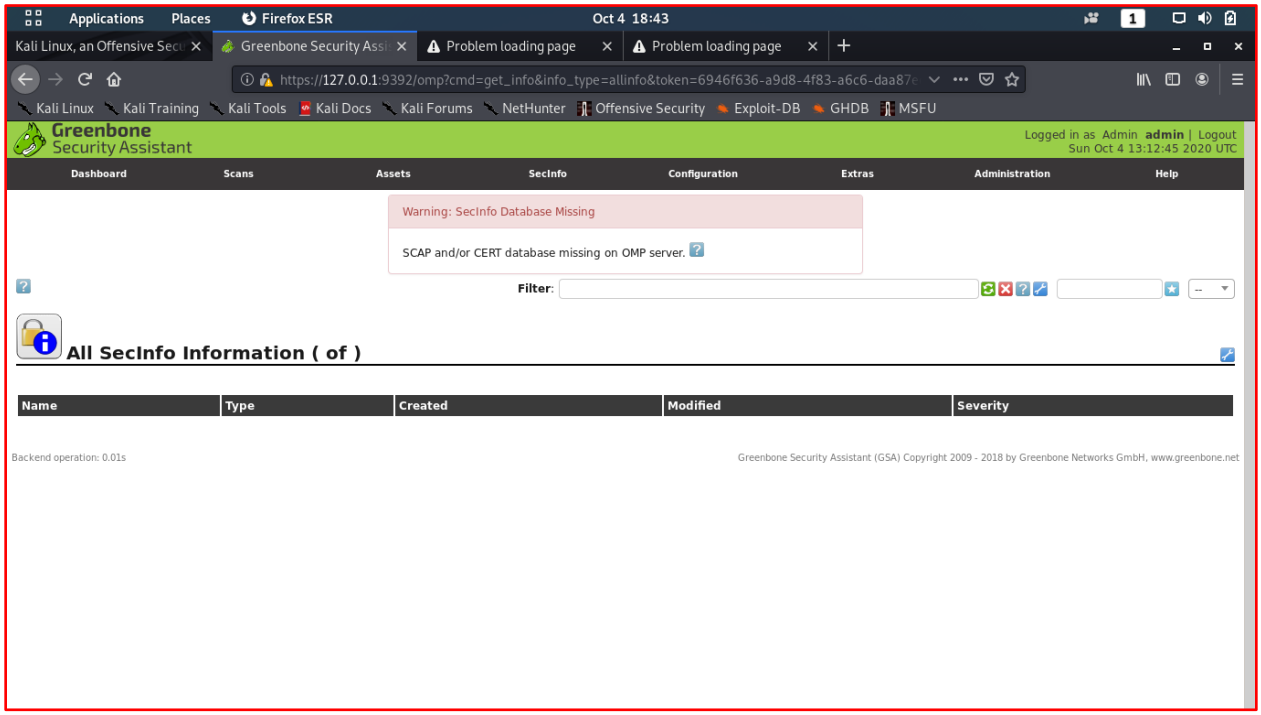
Then you need to provide the correct user login credentials that we previously retrieved.



After the successful login, you are able to see the Dashboard of **"OpenVAS"**.



But in here, there is an issue with the **“SecInfo”** Database. It always shows a pop-up error message saying that the **“SecInfo Database missing”**.

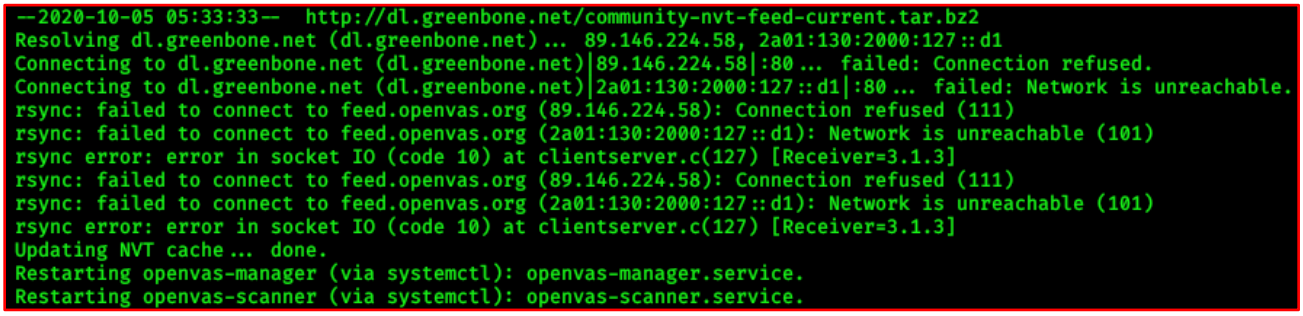


I spent so many hours watching YouTube videos, and surfing through different web-forums in order to find a proper solution for this issue. This issue was raised by many people on those web forums. But unfortunately I was not able to find a proper solution to fix the **“SecInfo Database missing”** issue.

I tried so many methods using the terminal. Not only that I also created a shell code too in order to fix this problem.

```
dilshan@kali:~/Desktop$ ls
openvas-update2020.sh  wireless
dilshan@kali:~/Desktop$ cat openvas-update2020.sh
#!/bin/bash
/usr/sbin/greenbone-nvt-sync
/usr/sbin/greenbone-certdata-sync
/usr/sbin/greenbone-scapdata-sync
/usr/sbin/openvasmd --update --verbose --progress
/etc/init.d/openvas-manager restart
/etc/init.d/openvas-scanner restart
```

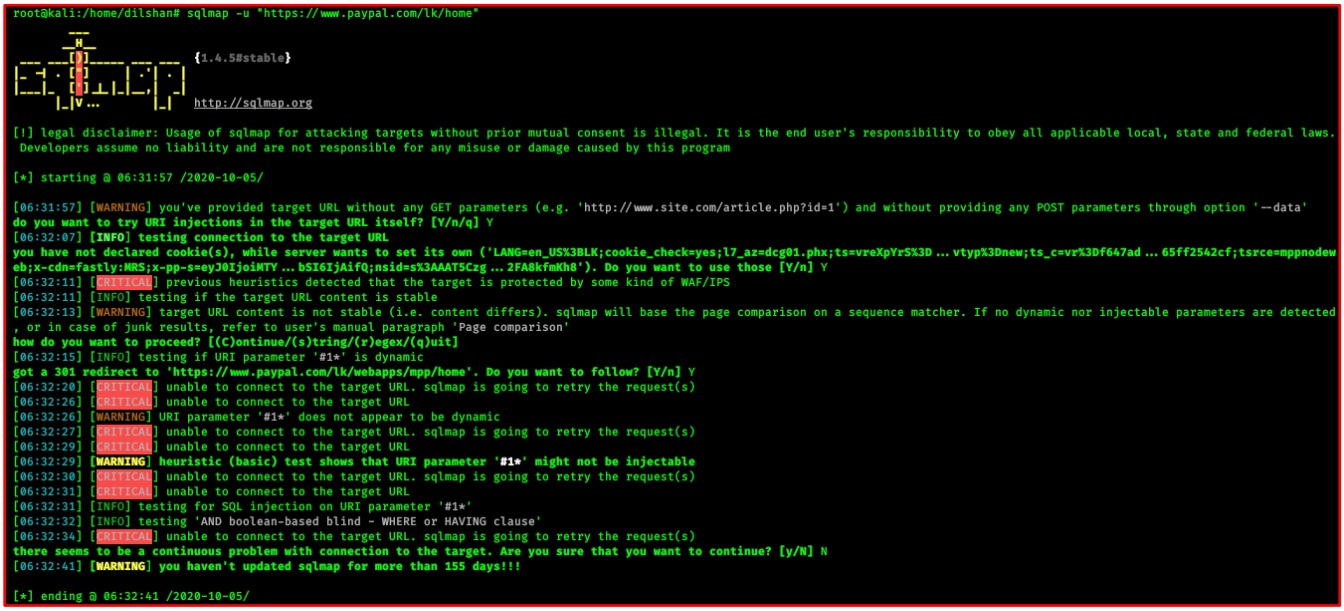
But none of those solutions were able to update that particular database. I guess that it's some kind of fault with their web servers. Because in the terminal it shows that their server's network cannot be reached.



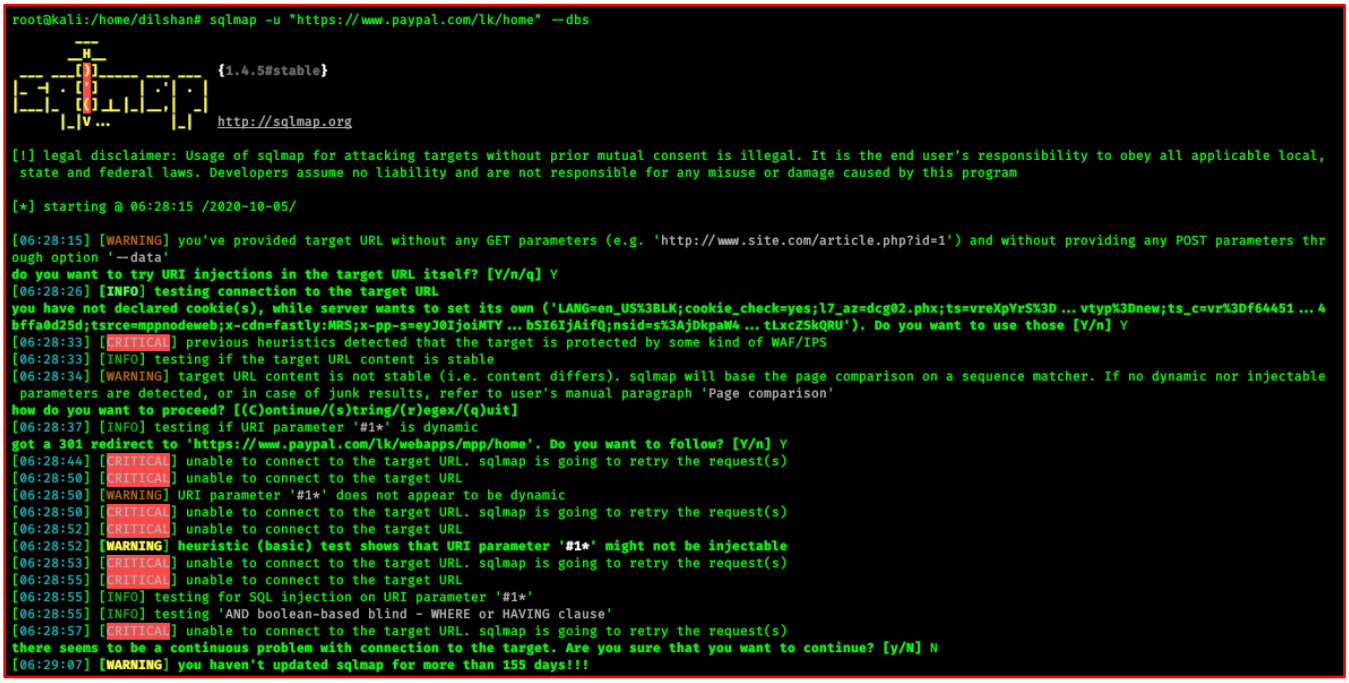
4) Sqlmap

“SQLmap” is also a great vulnerability assessment tool, which is mostly used in identifying database related vulnerabilities.

In here, we are able to perform several SQL Injection attacks on the host target as shown below.

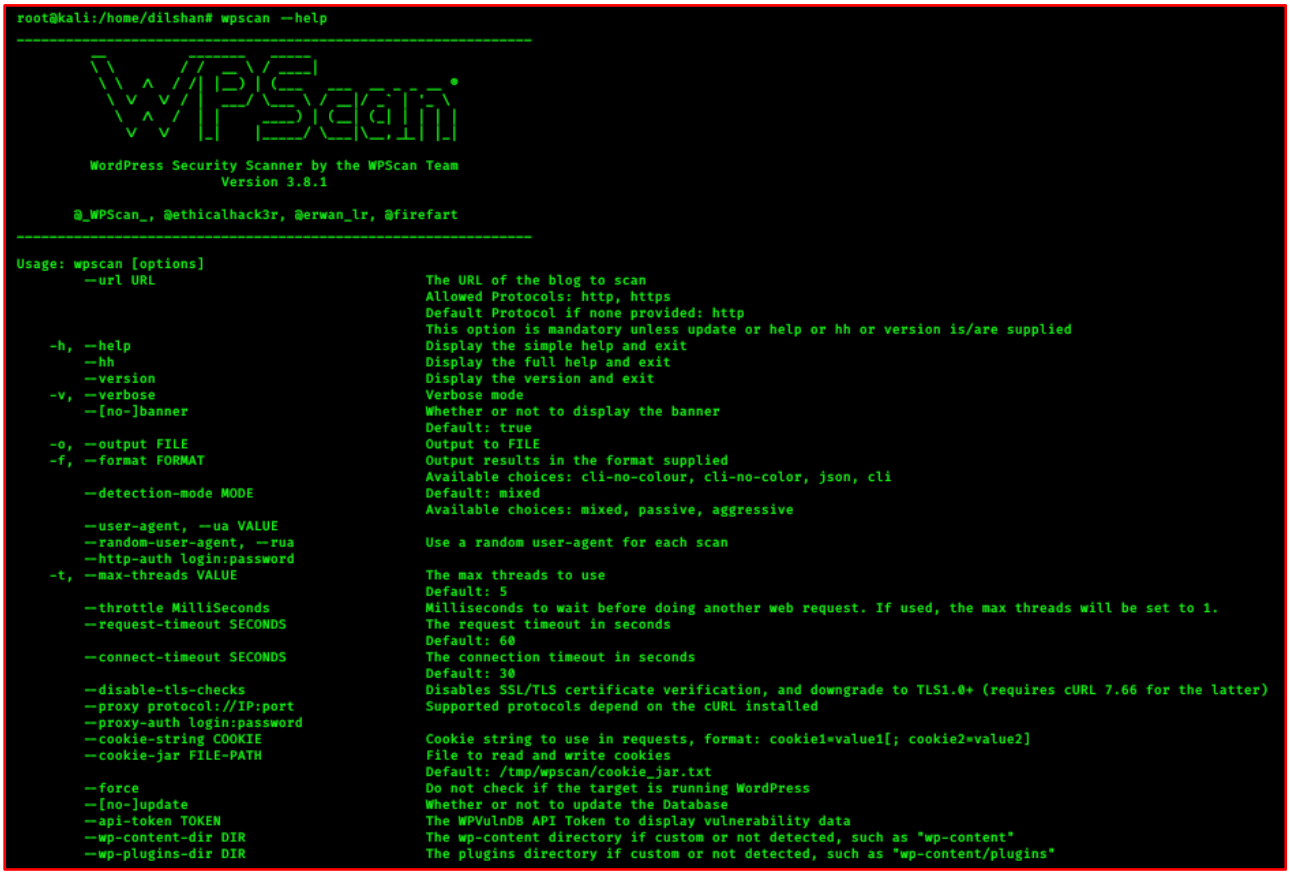


We can use the command **sqlmap –u "https://www.paypal.com" --dbs** to retrieve tables from the target web server. But in **PayPal**, their web server is secured in many different ways against SQL Injection attacks.

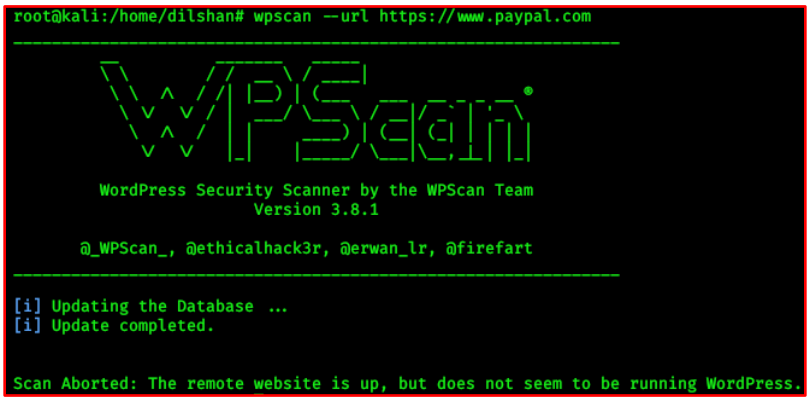


5) WPscan

“WordPress Scan (WPscan)” tool is one of the greatest web application vulnerability scanning tools out there.



But the only downside of this tool is, only the web domains which are designed using WordPress can be scanned by this tool.



6) nmap

"nmap" is usually considered as the most popular vulnerability analysis tool among the professionals.

Not only that, **nmap** is very much useful especially in scanning for vulnerabilities of a particular web application.

First of all, let's see **"what systems are alive"** using the **nmap** tool.

```
root@kali:/home/dilshan# nmap -sP paypal.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-05 07:59 +0530
Nmap scan report for paypal.com (64.4.250.37)
Host is up (0.00051s latency).
Other addresses for paypal.com (not scanned): 64.4.250.36
rDNS record for 64.4.250.37: 37.250.4.64.in-addr.arpa
Nmap done: 1 IP address (1 host up) scanned in 5.13 seconds
```

Now let's check for the open ports in our target host.

```
root@kali:/home/dilshan# nmap -sT -p 80,443 64.4.250.37
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-05 08:05 +0530
Nmap scan report for paypal.com (64.4.250.37)
Host is up (0.067s latency).

PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 1.55 seconds
```

With the help of nmap, we are able detect currently running OS of the targeted web server.

```
root@kali:/home/dilshan# nmap -O paypal.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-05 08:42 +0530
Nmap scan report for paypal.com (64.4.250.37)
Host is up (0.011s latency).
Other addresses for paypal.com (not scanned): 64.4.250.36
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|switch|phone|game console
Running (JUST GUESSING): Linux 1.0.X (87%), Cisco embedded (87%), Nokia Symbian OS (86%), Ouya embedded (85%)
OS CPE: cpe:/o:linux:linux_kernel:1.0.9 cpe:/h:cisco:catalyst_1900 cpe:/o:nokia:symbian_os
Aggressive OS guesses: Linux 1.0.9 (87%), Cisco Catalyst 1900 switch (87%), Nokia 3600i mobile phone (86%), OUYA game console (85%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.10 seconds
```

So now let's try to perform some aggressive scanning operations such as **Version Detection, Script Scanning and Trace Route** on "paypal.com".

```
root@kali:/home/dilshan# nmap -A paypal.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-05 08:46 +0530
Nmap scan report for paypal.com (64.4.250.36)
Host is up (0.0052s latency).
Other addresses for paypal.com (not scanned): 64.4.250.37
Not shown: 998 filtered ports
PORT    STATE SERVICE    VERSION
80/tcp  open  tcpwrapped
443/tcp open  tcpwrapped
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Cisco 3000 switch (IOS 10.3) (94%), Cisco Catalyst 1900 switch (93%), Nokia 3600i mobile phone (93%), Cisco ATA 188 VoIP adapter (91%), Apple T
ime Capsule NAS device (90%), Oracle Virtualbox (87%), QEMU user mode network gateway (87%), GNU Hurd 0.3 (87%), Huawei Echolife HG520-series ADSL modem (87%), TP-LIN
K TD-W8951ND wireless ADSL modem (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   3.18 ms 10.0.2.2
2   3.20 ms paypal.com (64.4.250.36)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 118.98 seconds
```

So now let's move on for scanning the host for vulnerabilities with the help of **automated scripts.**

```
root@kali:/home/dilshan# nmap --script vuln paypal.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-05 08:53 +0530
Nmap scan report for paypal.com (64.4.250.37)
Host is up (0.024s latency).
Other addresses for paypal.com (not scanned): 64.4.250.36
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
443/tcp open  https
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_sslv2-drown:

Nmap done: 1 IP address (1 host up) scanned in 239.42 seconds
```

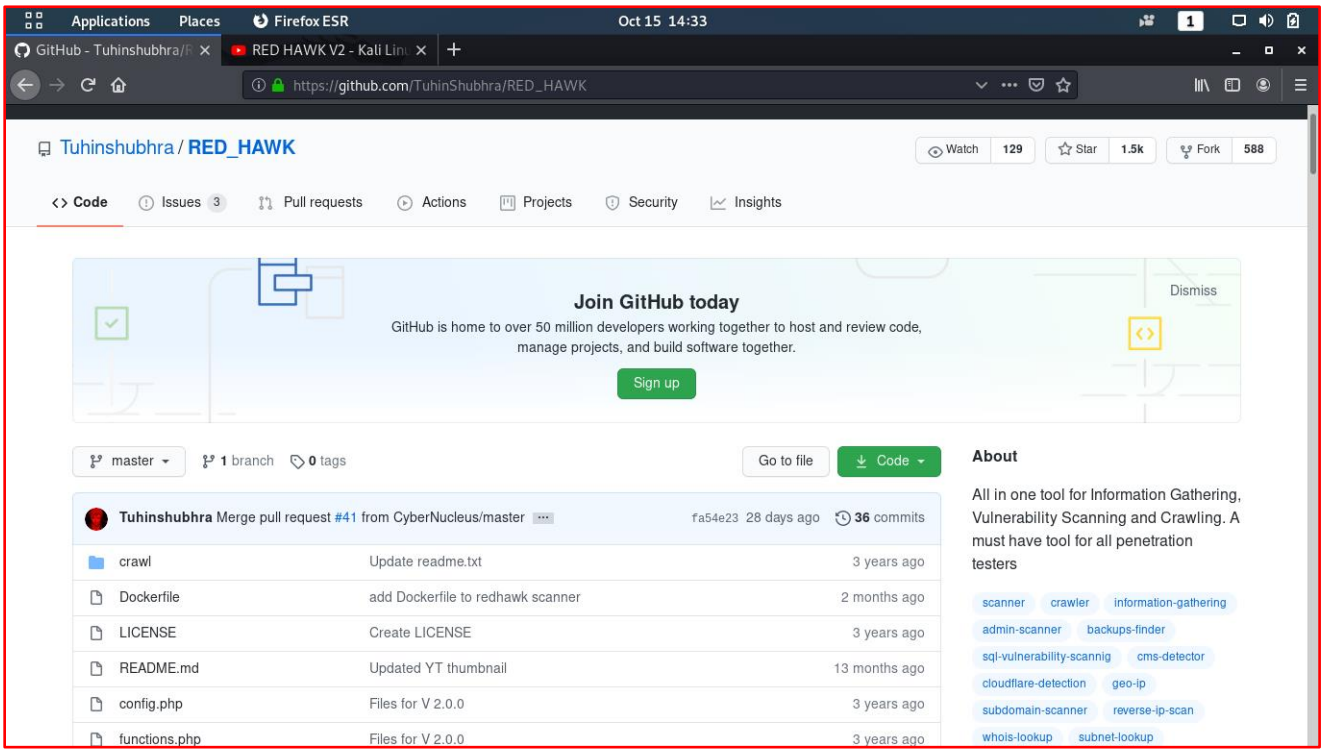
7) Red Hawk (Version 2)

“Red Hawk” can be considered as an **all in one information gathering tool** and a **vulnerability scanner** for **Kali Linux.**

This is also one of the most powerful web crawling tools out there specially in Kali Linux. This comes under the field of reconnaissance.

Now let’s see how to install it and use it.

So as the very first step you need to **clone the Red Hawk GIT Repository.**

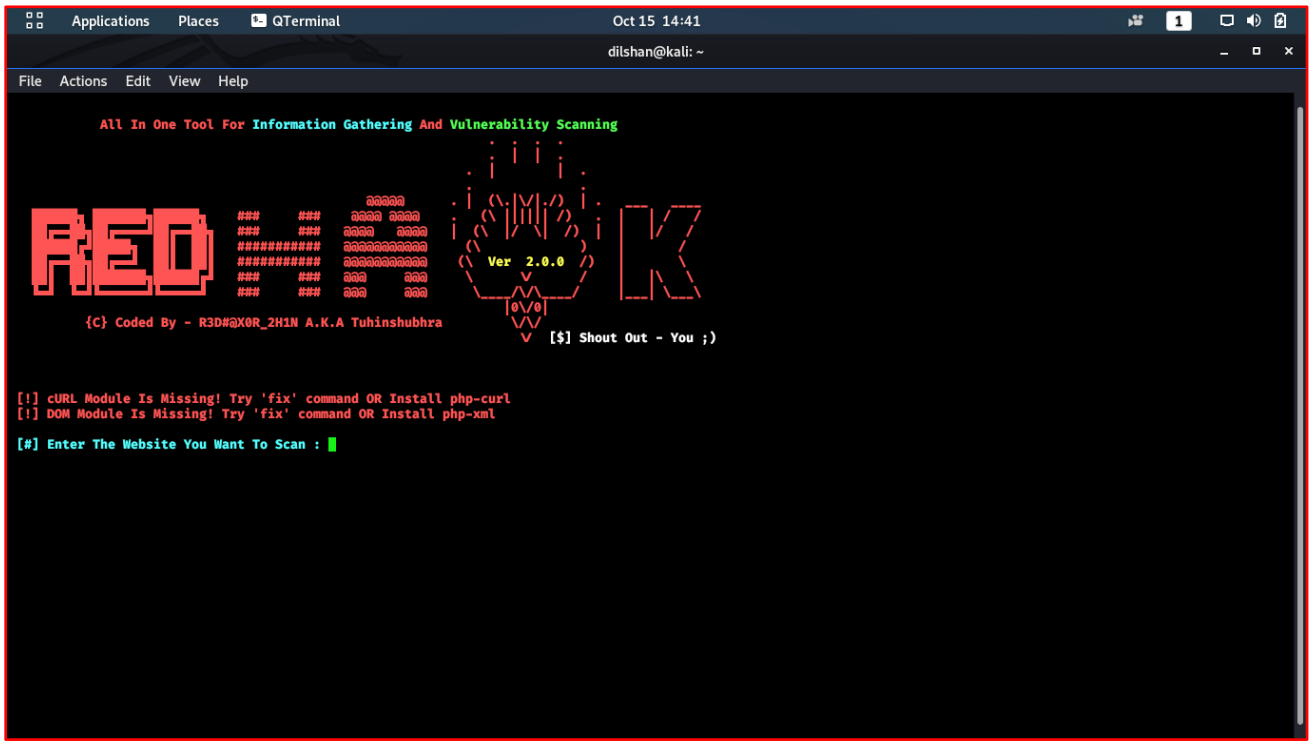



```
root@kali:/home/dilshan# git clone https://github.com/Tuhinshubhra/RED_HAWK.git
Cloning into 'RED_HAWK' ...
 remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 106 (delta 0), reused 2 (delta 0), pack-reused 102
Receiving objects: 100% (106/106), 47.02 KiB | 149.00 KiB/s, done.
Resolving deltas: 100% (43/43), done.
```

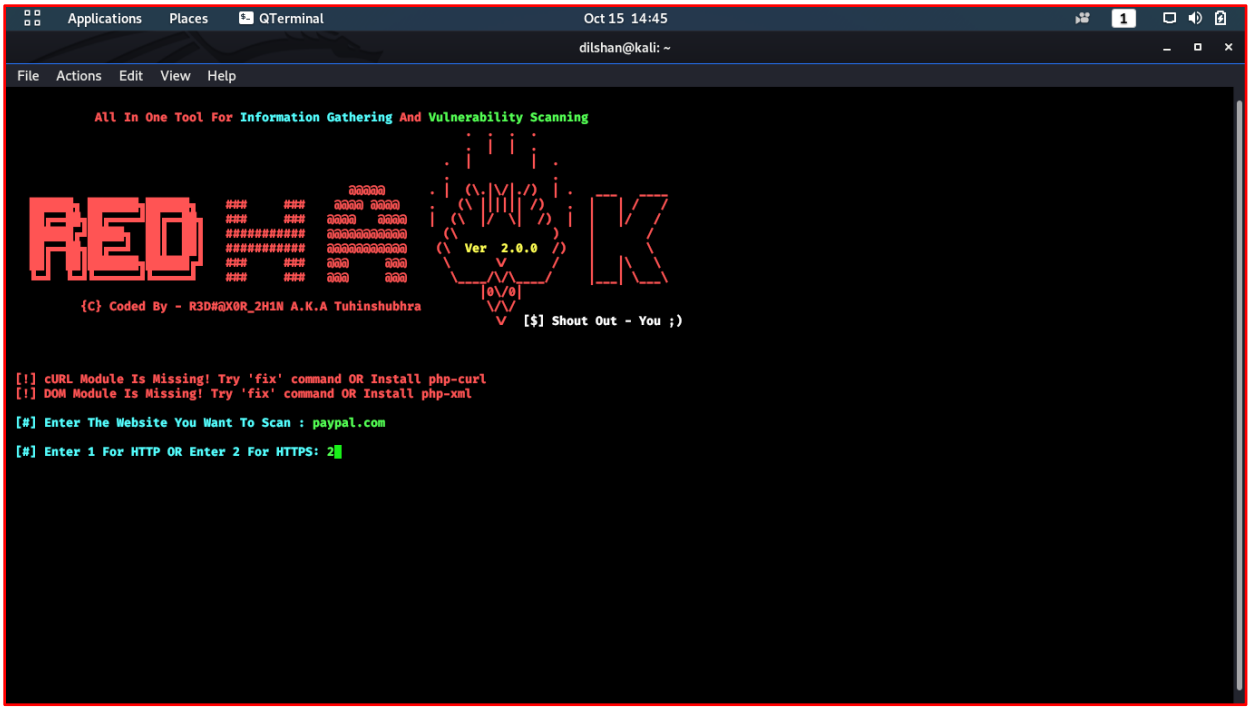
So after cloning the repository, you just simply need to launch the **php based "redhawk"** tool. There is no installation process for this particular tool.

```
root@kali:/home/dilshan# cd RED_HAWK/
root@kali:/home/dilshan/RED_HAWK# ls
config.php  crawl  Dockerfile  functions.php  LICENSE  README.md  rhawk.php  sqlerrors.ini  var.php  version.txt
root@kali:/home/dilshan/RED_HAWK# php rhawk.php
```

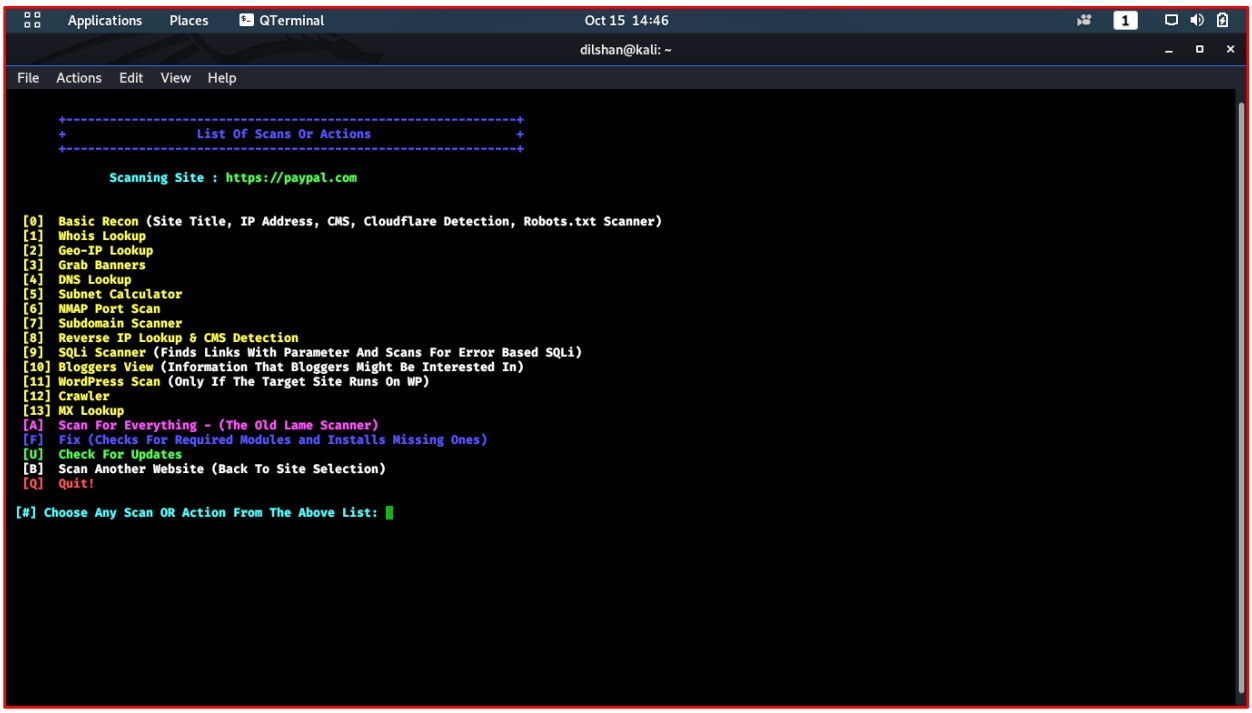
So as you can see in the following image, this is how the tool's interface looks like.



So in here you need to input your target web domain. In my case it's **"Paypal.com"**. Then you need to define whether that particular web domain is based on **HTTP** or **HTTPS**.



So now you are able to see the list of scans or actions that can be performed on that particular specified target.



i. Basic Reconnaissance

```
[#] Choose Any Scan OR Action From The Above List: 0

[+] Scanning Begins ...
[i] Scanning Site: https://paypal.com
[S] Scan Type : BASIC SCAN

[iNFO] Site Title:
[iNFO] IP address: 64.4.250.36
[iNFO] Web Server: Varnish
[iNFO] CMS: Could Not Detect
[iNFO] Cloudflare: Not Detected
```

ii. Subnet Calculator

```
[#] Choose Any Scan OR Action From The Above List: 5

[+] Scanning Begins ...
[i] Scanning Site: https://paypal.com
[S] Scan Type : SubNet Calculator


[SubNet Calc] Address       = 64.4.250.37
[SubNet Calc] Network       = 64.4.250.37 / 32
[SubNet Calc] Netmask       = 255.255.255.255
[SubNet Calc] Broadcast     = not needed on Point-to-Point links
[SubNet Calc] Wildcard Mask = 0.0.0.0
[SubNet Calc] Hosts Bits    = 0
[SubNet Calc] Max. Hosts    = 1   (2^0 - 0)
[SubNet Calc] Host Range    = { 64.4.250.37 - 64.4.250.37 }

[*] Scanning Complete. Press Enter To Continue OR CTRL + C To Stop
```

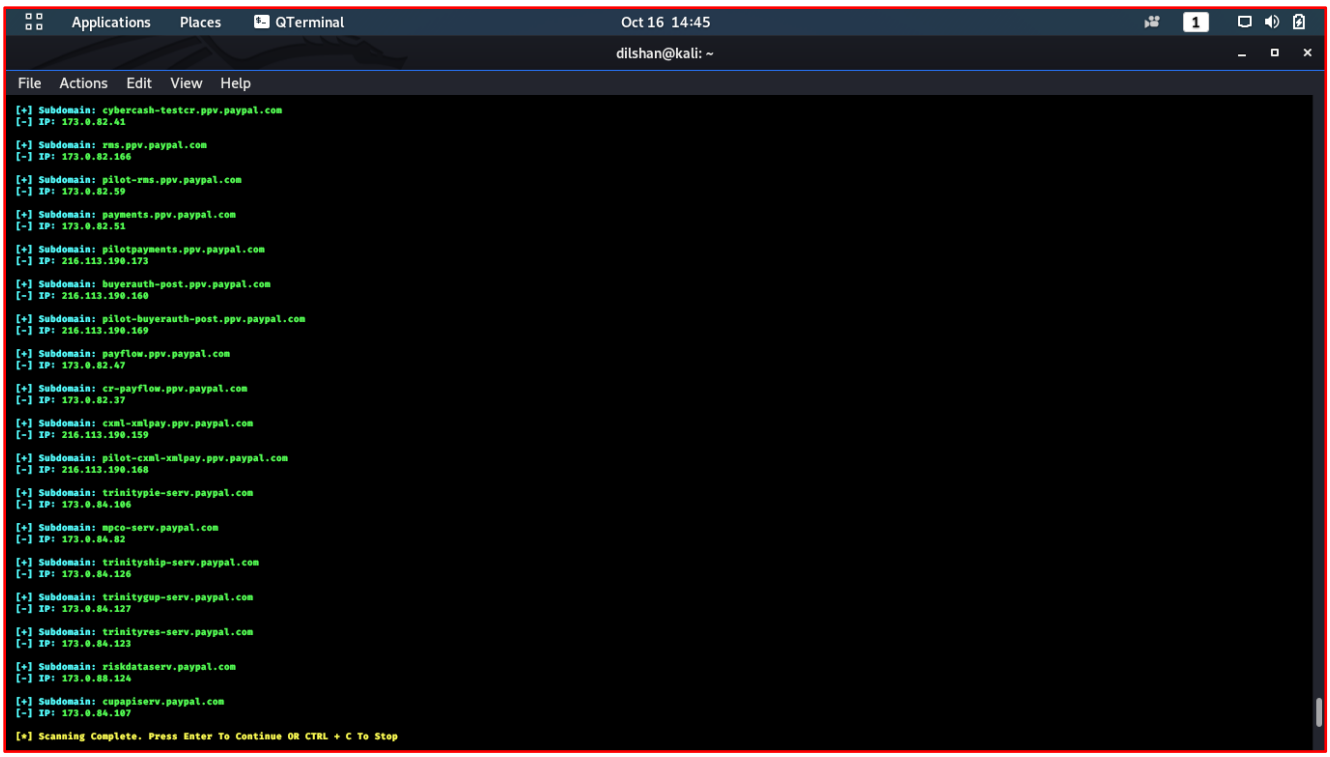
iii. NMAP Port Scan

```
[#] Choose Any Scan OR Action From The Above List: 6

[+] Scanning Begins ...
[i] Scanning Site: https://paypal.com
[S] Scan Type : Nmap Port Scan
[~] Port Scan Result:

Starting Nmap 7.70 ( https://nmap.org ) at 2020-10-16 09:12 UTC
Nmap scan report for paypal.com (64.4.250.37)
Host is up (0.034s latency).
Other addresses for paypal.com (not scanned): 64.4.250.36

PORT     STATE    SERVICE
21/tcp   filtered ftp
22/tcp   filtered ssh
23/tcp   filtered telnet
80/tcp   open     http
110/tcp  filtered pop3
143/tcp  filtered imap
443/tcp  open     https
3389/tcp filtered ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 1.41 seconds


[*] Scanning Complete. Press Enter To Continue OR CTRL + C To Stop
```

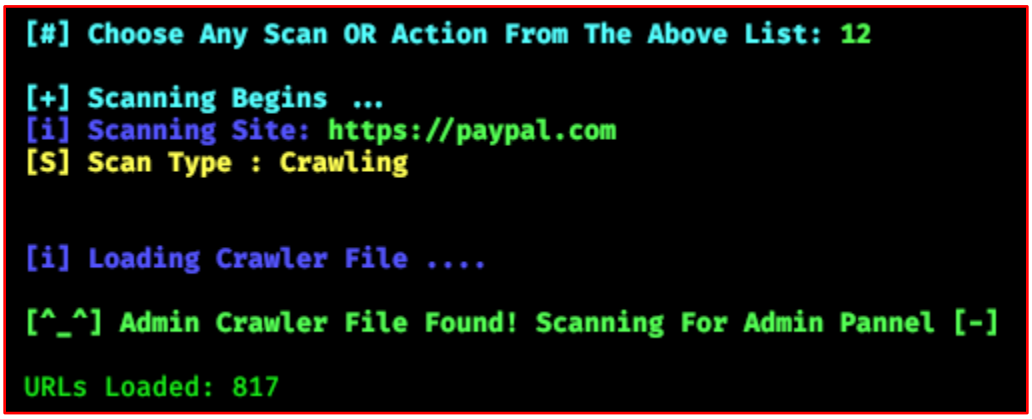
iv. SQL Vulnerability Scanner

```
[#] Choose Any Scan OR Action From The Above List: 9

[+] Scanning Begins ...
[i] Scanning Site: https://paypal.com
[S] Scan Type : SQL Vulnerability Scanner
```

v. Sub-Domain Scanner



vi. Crawling



Web Application Firewall Detection (WAF00F)

Web Application Firewall Detection is the mitigation procedure that is implemented within the web servers, in order to protect web applications from attacks.

So what this firewall basically does is it **blocks the different kinds of attacks** which are reached to the server.

So when performing an exploitation, firstly you need to **encode the data in a specific way to bypass the firewall. Otherwise it will be blocked by the firewall.** So that the requests won't be processed by the server.

If a particular web application uses this firewall, first we need to detect this firewall using the tool **"WAFW00F"**.

So as you can see above, https://www.paypal.com is behind **"Kona SiteDefender (Akamai)"** web application firewall.

So most of the attacks or manipulation of data will be blocked for certainly. So we won't get the response as we wanted.

So now let's see how to get our scan done without getting our IP Address blocked.

1) ***Use a VPN –>*** It will hide our IP Address and encrypts your connection and routes it through an intermediary server in another location
2) ***Use Tor Web browser –>*** This will encrypt our requests and encrypts our internet connection and routes it through a random sequence of servers
3) ***Reduce the throughput of requests sent per second***

So in this occasion I'm going to use the 3rd method while scanning the targeted web domain.

Professional Pen testing Tools and Web Audit Reports

1) Netsparker Professional Edition

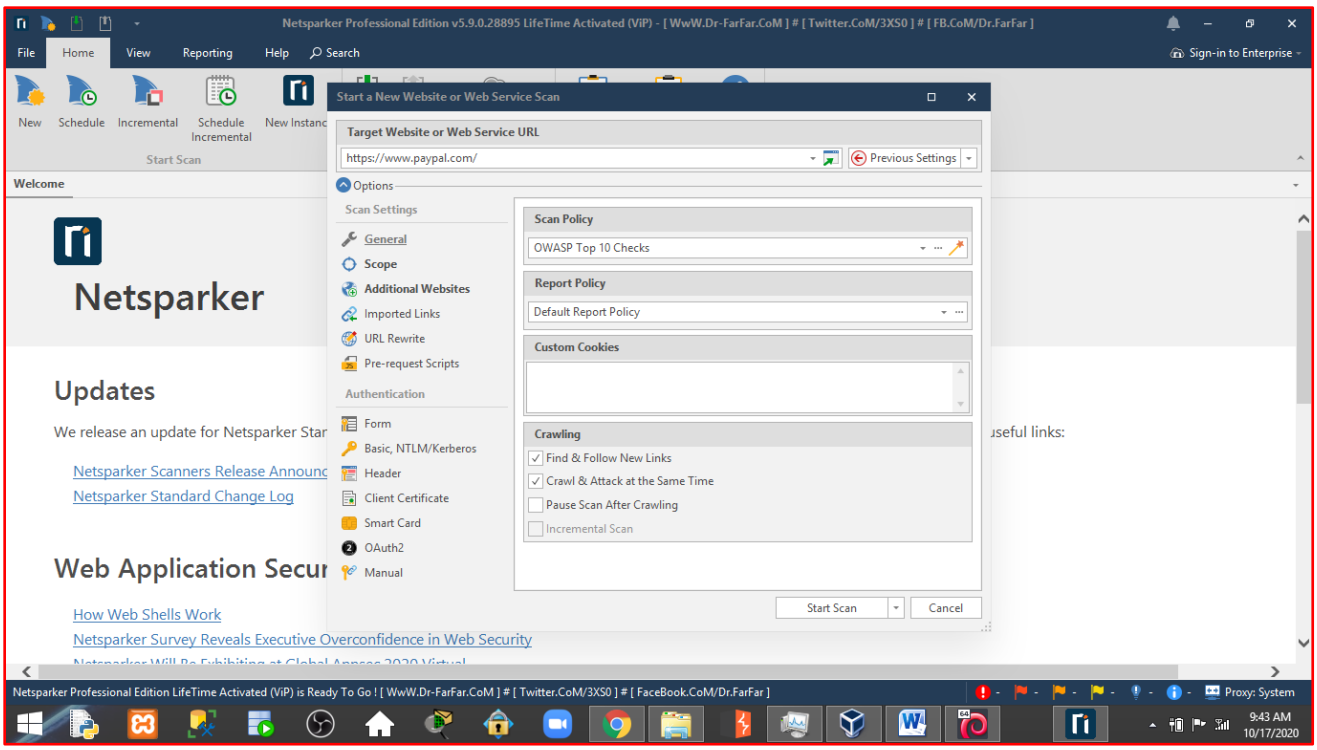
First of all, I should mention that original **"Netsparker Professional Edition"** costs more than 660$ per month. So as I'm unable to afford such expense, I downloaded and installed the **"Cracked Full Netsparker Professional Edition"** for free. To be honest, I spent several hours on studying how to install this cracked software because it wasn't an easy task at all. Fortunately although this is a cracked version, **it contains almost all the professional web pen testing features.**

"Netsparker" is a fully automated web application security scanner which allows its users to perform huge variety of tasks on a particular targeted website.

Some of those tasks are;

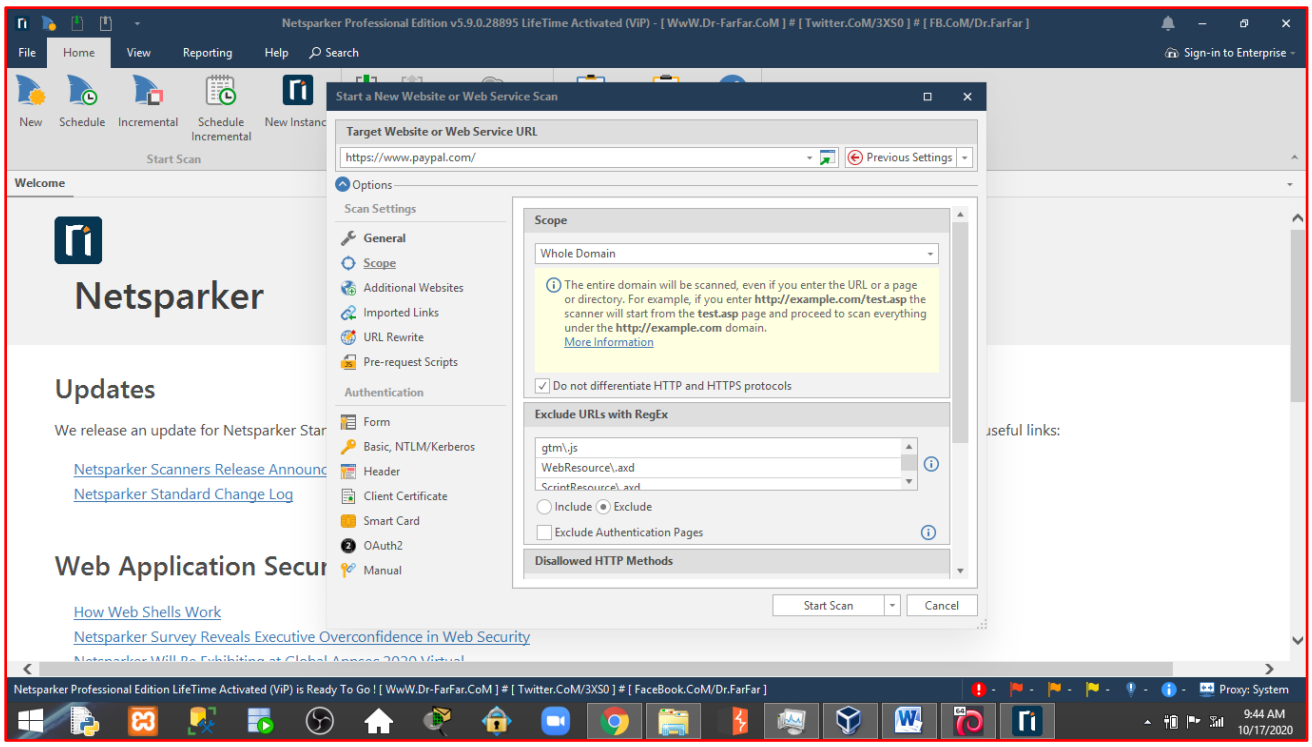
1. Scan the targeted website
2. Identify security flaws
3. Attack and exploiting the vulnerabilities
4. Advanced Crawling and Auditing mechanism
5. Generating Industry Standard Audit Reports

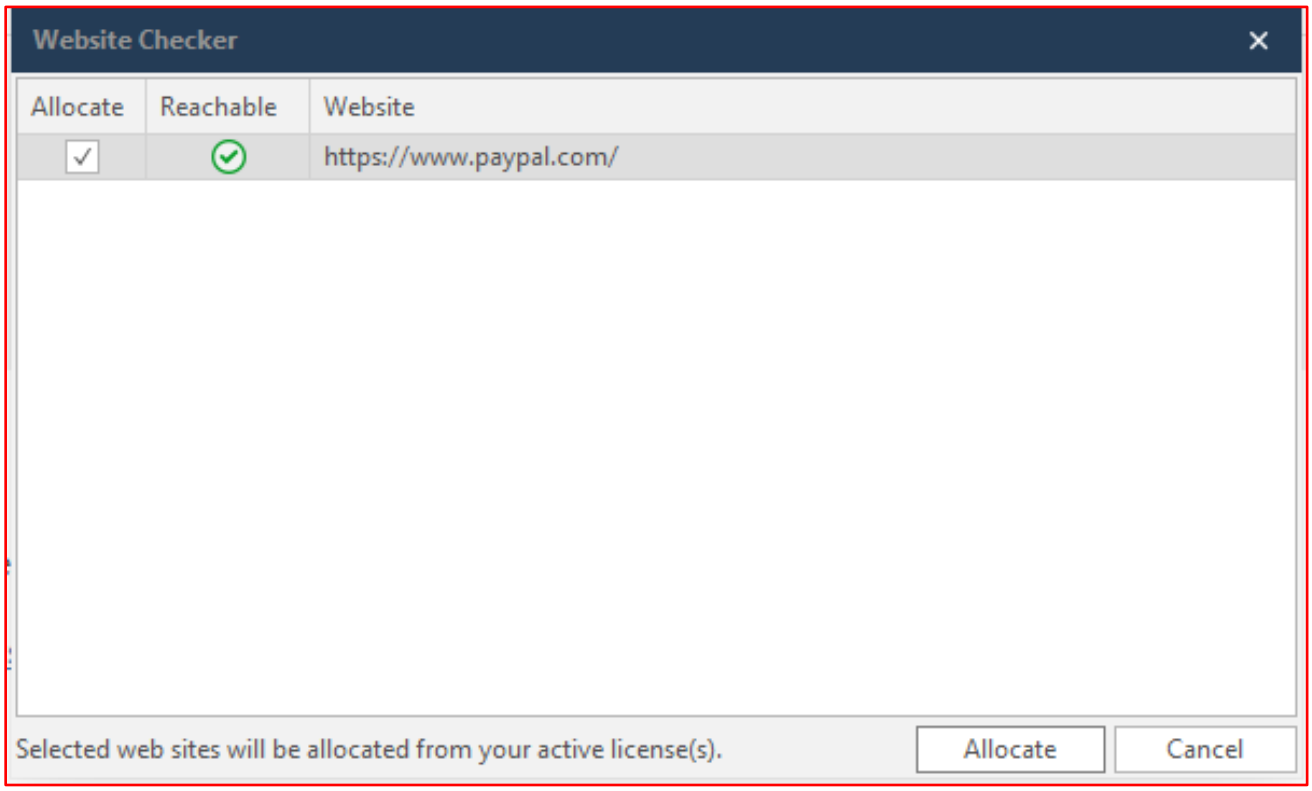
So now let's move on directly for scanning our targeted web domain.



So as I have mentioned previously, **Netsparker Professional Edition** consists with huge number of services. As you can see in the above image, after defining our **target website**, we need to define the scan policy. In this software, it has provided us with many scan policy options to be selected. But as our Lecturer in Charge Dr.Lakmal Rupasinghe mentioned, I'm going to choose the scan policy as **"OWASP Top 10 Checks".**

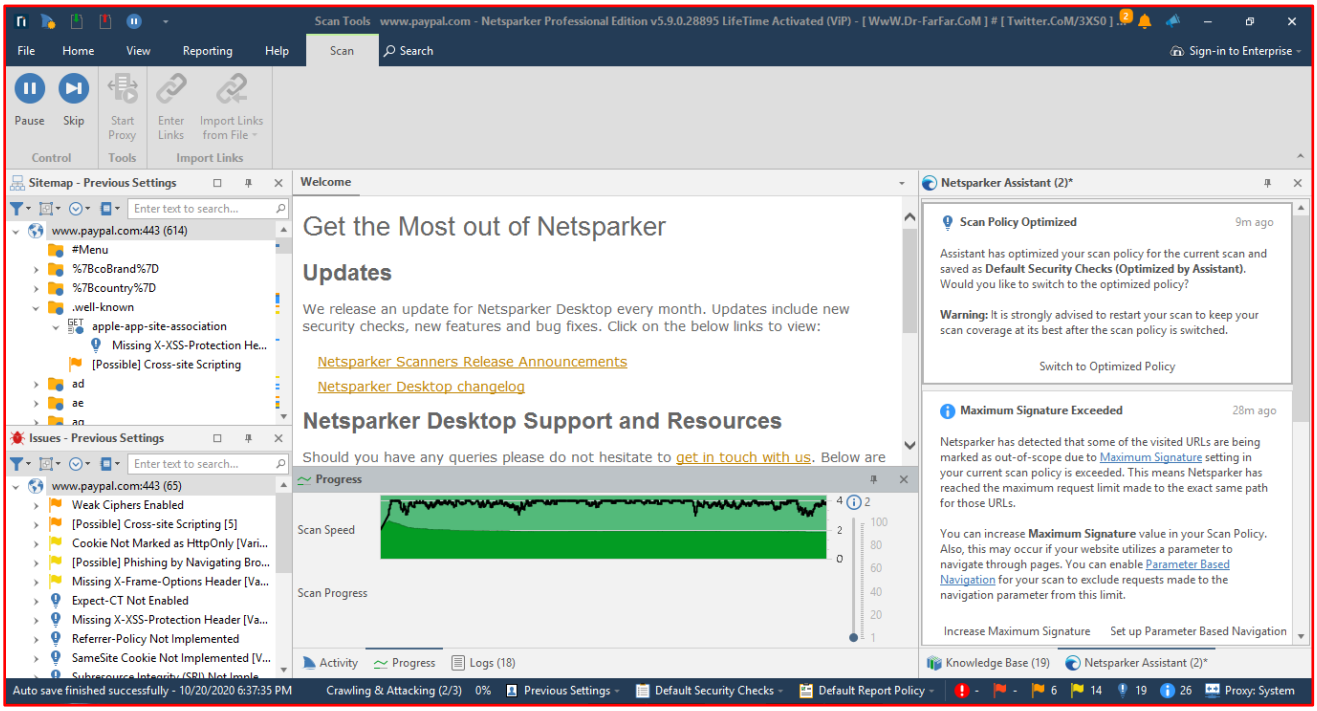
After completing this stage, we need to move on for defining the scope. As this web scanner is capable of scanning the whole web domain at once, I had decided to choose that option. Not only that, this option is surely going to be very useful specially in generating the final reports on that particular web domain.


Netsparker Professional Edition v5.9.0.28895 LifeTime Activated (ViP) - [WwW.Dr-FarFar.CoM] # [Twitter.CoM/3XS0] # [FB.CoM/Dr.FarFar]
File
Home
View
Reporting
Help
Search
Sign-in to Enterprise
New
Schedule
Incremental
Schedule Incremental
Start Scan
Welcome
Netsparker
Updates
We release an update for Netsparker Star
Netsparker Scanners Release Announc
Netsparker Standard Change Log
Web Application Secur
How Web Shells Work
Netsparker Survey Reveals Executive Overconfidence in Web Security
Start a New Website or Web Service Scan
Target Website or Web Service URL
https://www.paypal.com/
Previous Settings
Options
Scan Settings
General
Scope
Additional Websites
Imported Links
URL Rewrite
Pre-request Scripts
Authentication
Form
Basic, NTLM/Kerberos
Header
Client Certificate
Smart Card
OAuth2
Manual
Scope
Whole Domain
The entire domain will be scanned, even if you enter the URL or a page or directory. For example, if you enter http://example.com/test.asp the scanner will start from the test.asp page and proceed to scan everything under the http://example.com domain.
More Information
Do not differentiate HTTP and HTTPS protocols
Exclude URLs with RegEx
gtm\.js
WebResource\.axd
Include
Exclude
Exclude Authentication Pages
Disallowed HTTP Methods
Start Scan
Cancel
Netsparker Professional Edition LifeTime Activated (ViP) is Ready To Go ! [WwW.Dr-FarFar.CoM] # [Twitter.CoM/3XS0] # [FaceBook.CoM/Dr.FarFar]
Proxy: System
9:44 AM
10/17/2020

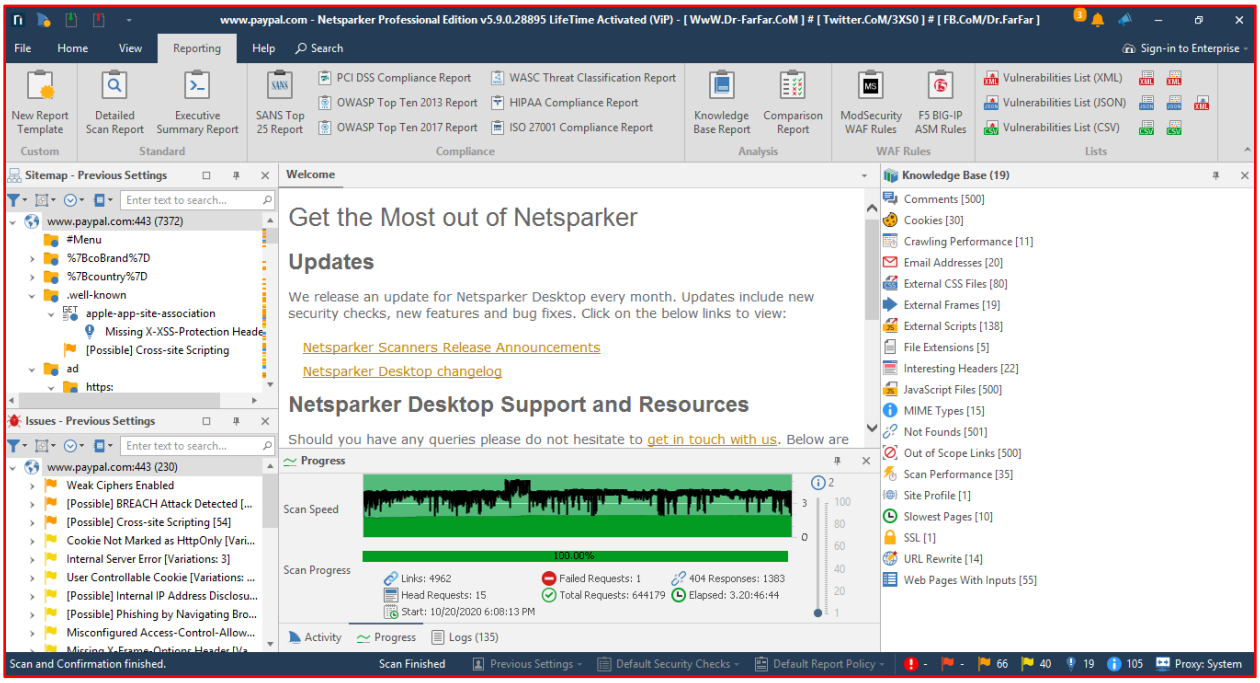

Website Checker
Allocate
Reachable
Website
https://www.paypal.com/
Selected web sites will be allocated from your active license(s).
Allocate
Cancel

Usually before scanning a particular web domain, a pen tester should get written permissions from the owners of that website. Otherwise it is considered as a criminal offense or else as a cyber-crime. But as I have mentioned in the previous section, **most of the today's web servers are protected by firewalls.** Most of the E-Commerce related websites use multiple firewalls to protect their web servers against cyber-attacks. So in my case, PayPal also uses such kind of firewalls in order to protect their web servers from the attackers. **Usually when a pen tester scans a particular web domain in a legal manner, his IP Address is whitelisted by the firewall.** But if anyone who does this without any kind of written permission, his IP Address is surely going to be blocked.

To be honest, this thing happened to me several times. **My IP Address got blacklisted several times due to the attacking process.** But after performing a small research, I was able to find a proper solution for this. I had decided to **limit the throughput of the requests towards the targeted web domain.** I have chosen to send **2 requests per a second** towards the target web domain. By applying this method, **I was able to successfully bypass PayPal's web-application firewall without getting my IP Address blocked.**



So, after about 96 long hours of scanning process, it finally came to an end.



After that, you get the opportunity to generate a report type as you wish. But here, I'm going to use **"OWASP Top 10 2017"** report structure. In here, the software provides us the chance to export the report in both HTML and PDF formats.

So I will be publishing the **"Web Vulnerability Audit Report"** as an attachment ***(as both HTML and PDF files)*** with this document in a ZIP file.

Additionally for the better understanding of the specific vulnerabilities, I have included the **following report types** as well inside the zip file.

1) **Detailed Scan Report**
2) **Executive Summary Report**
3) **SANS Top 25 Report**

2) Acunetix Premium Edition

So just like previous scenario, **"Acunetix Premium Edition"** costs more than 400$ per month. Although the **"Community Editions"** and **"Standard Editions"** of those tools are available for free, they don't provide us the necessary web crawling and auditing services that we actually need. So the only remaining solution for me was to download and install the cracked version of **"Acunetix Premium Edition".**

Like I have mentioned previously, installing cracked software is not a very easy task. I had to try several times re-installing the software in order to work properly.

"Acunetix" is an automated web application security testing tool that **can do both auditing and crawling by checking the vulnerabilities such as Cross-Site Scripting (XSS), SQL Injection etc.**

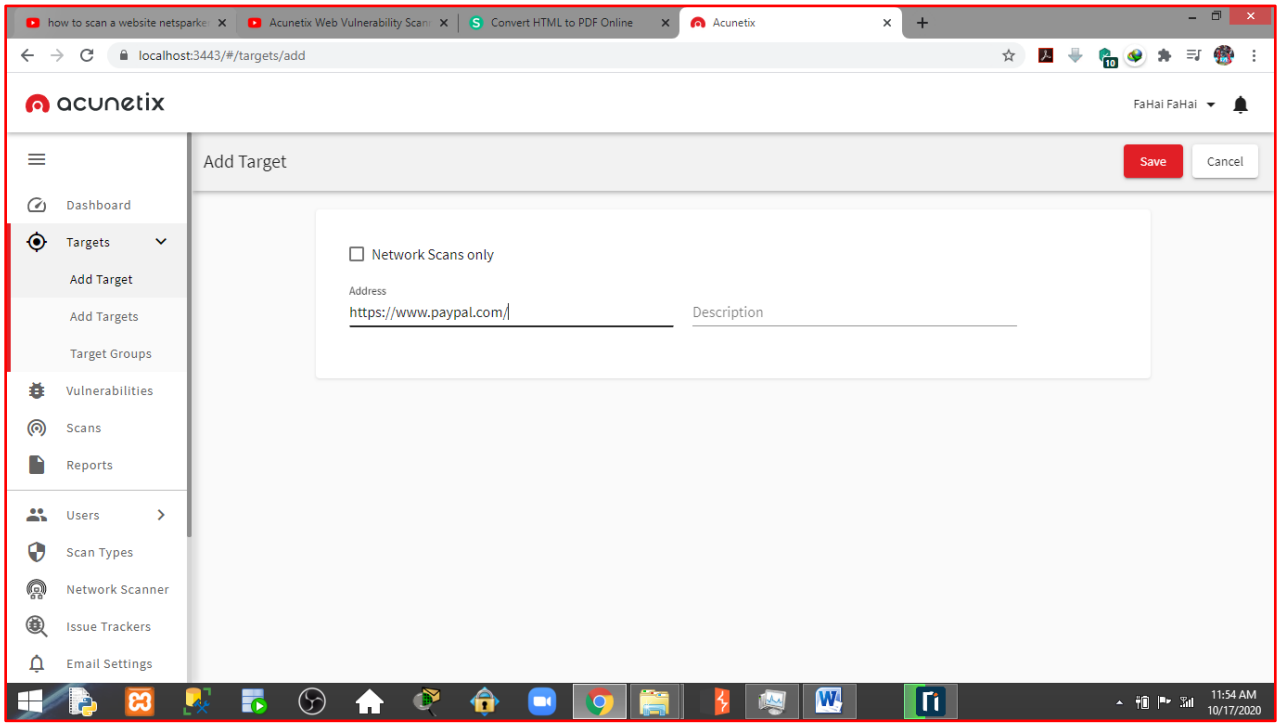
This tool uses a special and a very unique technology called **"AcuSensor Technology"** to identify more vulnerabilities than the other web application scanners.

Not only that, **"Acunetix Premium Edition"** generates only a very few false positives. So when compared to other web application scanners, "Acunetix" can be trusted to a greater extent.

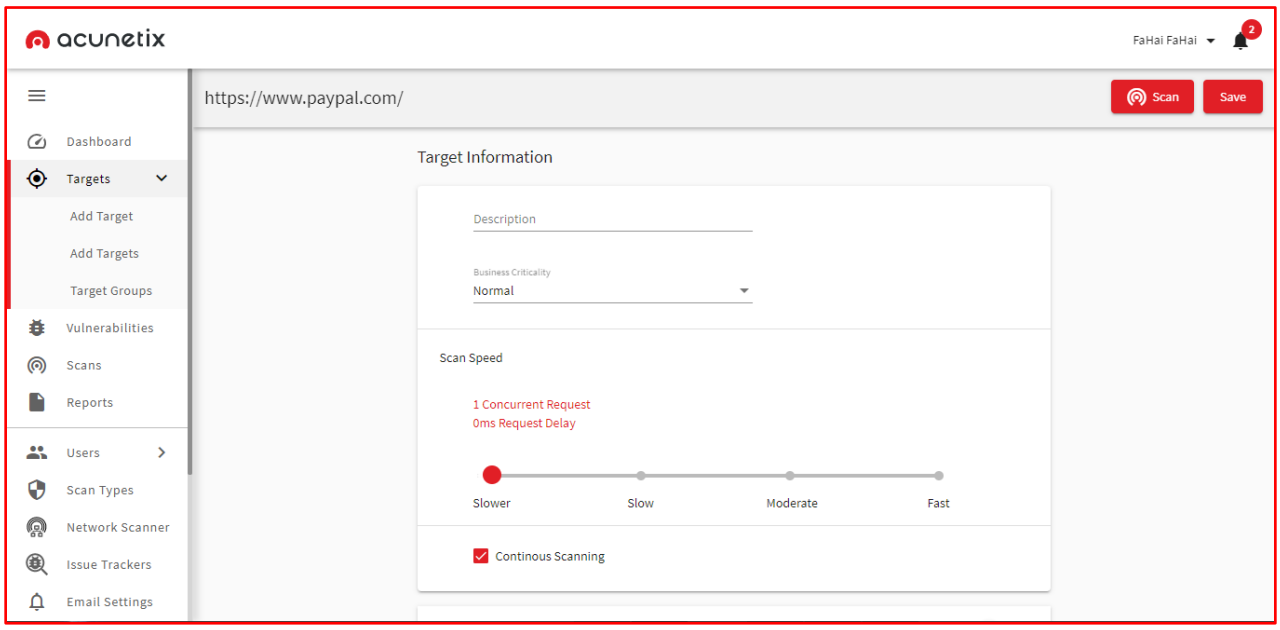
When compared to other web application scanners, Acunetix consists with more advanced and very useful features such as;

1. Higher vulnerability detection rate with 0% of false postives
2. Identifying and testing hidden inputs which are not discovered during black-box scanning (DAST)
3. Prioritizing and classifying the detected issues
4. Advanced Crawling and Auditing mechanism enables to crawl JavaScript websites and SPAs
5. Generating Industry Standard custom Audit Reports

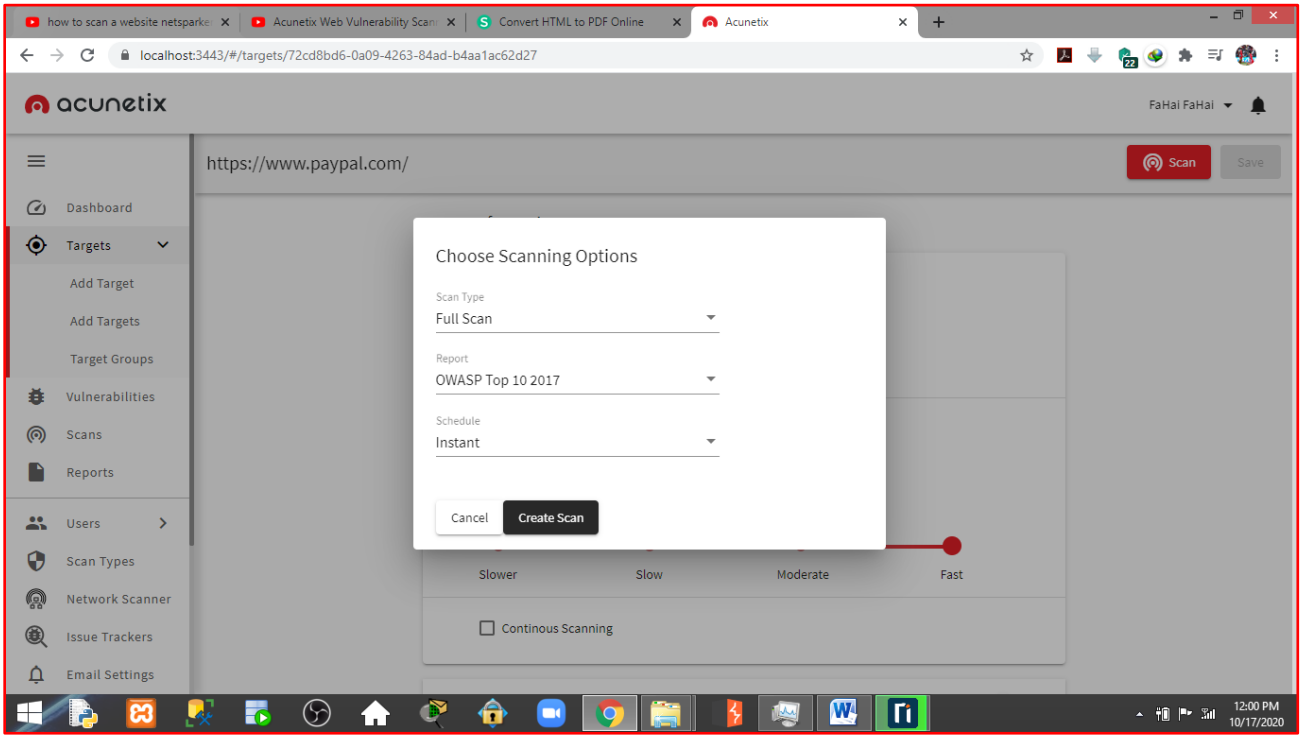
So as the very 1st step, you need to define the target domain in the provided text area.



Then you need to set the scanning speed as you wish. But in order to bypass the Web Application Firewall (WAF), I'm choosing a slower scan speed.



Then under the **“Choose Scanning Option”** tab, we need to select the report type that we need as the final output. So In here I’m going to choose the **“OWASP Top 10 2017”** standard for the report.



But unfortunately soon after I hit the **“Create Scan”** button, my laptop immediately got stuck.

Honestly I tried several times to run this scan after resetting my laptop each and every time.

But unfortunately the RAM of my laptop was unable to handle such a heavy workload. Each and every time I press the start scan, RAM usage rises up to the maximum level and then suddenly system gets stuck completely.

So I was unable to generate a **“Web Vulnerability Audit Report”** by using this powerful tool.

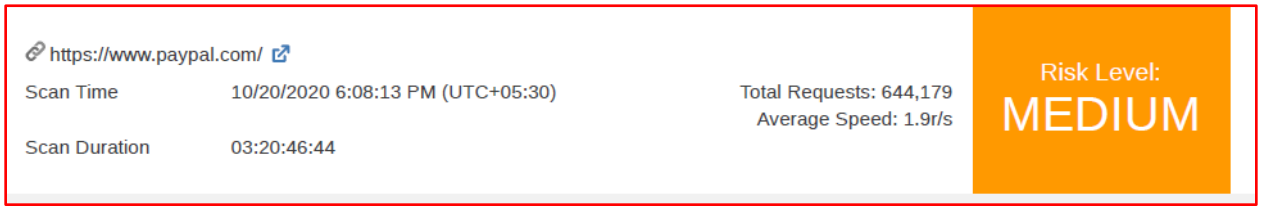
Conclusion and References

So after a long 96 hours of scanning process, it finally came to an end. Usually **"PayPal"** is considered as one of the most trusted and the safest online financial service out there. A person can just simply add his/her bank account, credit card or debit card details whenever transferring money using PayPal.

There are plenty of other popular online payment mechanisms out there such as Apple Pay, Samsung Pay, and Google Pay etc. But if we consider about PayPal, it offers more benefits to its customers when related to other payment options. Not only that, PayPal is very much famous among the world for its unique security. Currently PayPal uses more advance AI based security mechanisms in order to protect their customers' trust and money.

However as we all know, each and every software or web application in the world have its own loopholes. Developing a perfectly 100% secured software is not a possible task at all. Although one day if developers created such a product, attackers are surely going to find vulnerabilities of it and exploit it.

So to be honest, before scanning PayPal's web application, I also strongly believed that PayPal is a very much secure web application which is impenetrable for attackers. But soon after seeing the final **Web-Vulnerability Audit report,** I realized that I was completely wrong. The final summary report proved that, **https://www.Paypal.com** is a web application which acquires a **Risk Level of Medium.**



So as I have mentioned previously, with the **"Netsparker Professional Edition",** I generated different kinds of web-vulnerability reports. So according to the **standards of the particular reports**, it displayed different counts of vulnerabilities.

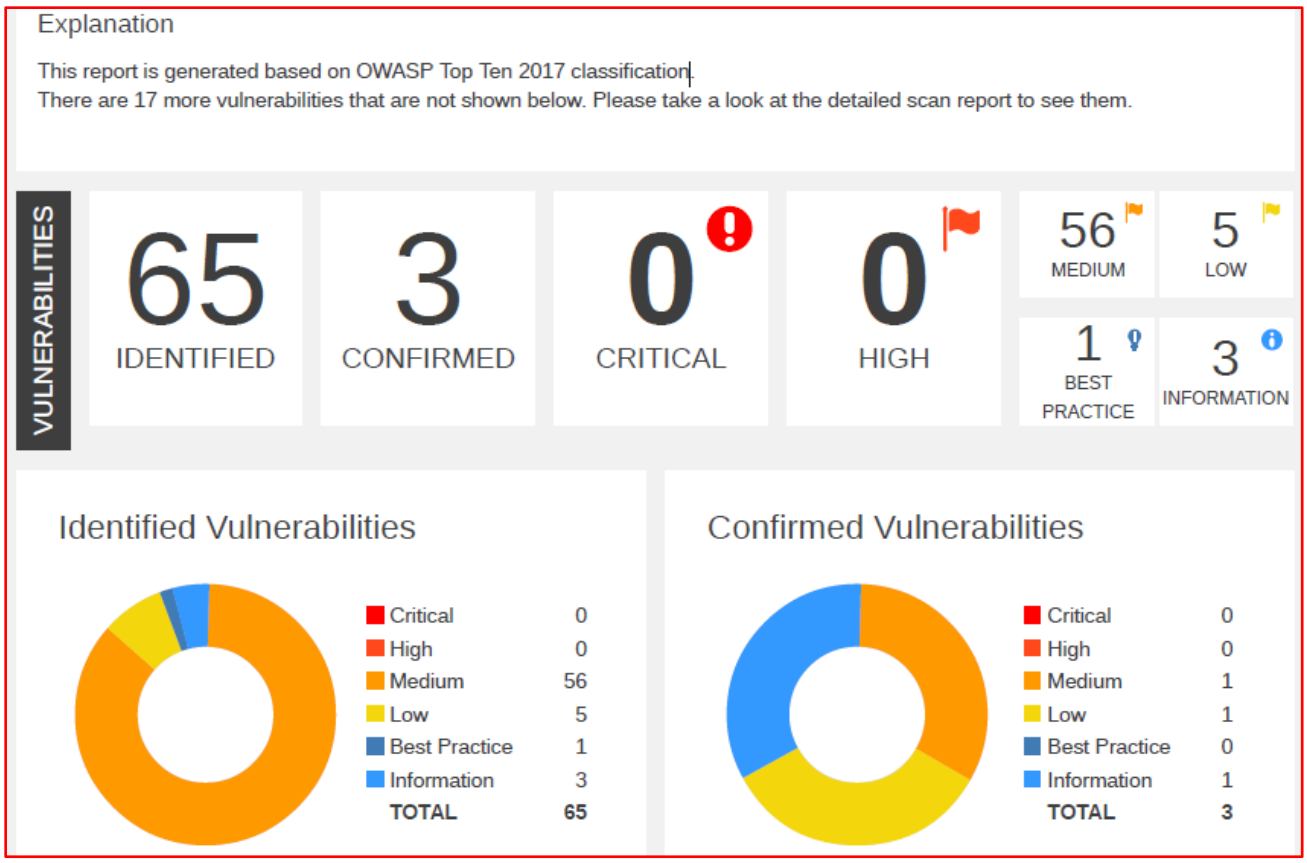
Compliance Summary

Compliance	Vulnerabilities
PCI DSS v3.2	55
OWASP 2013	65
OWASP 2017	65
WASC	71
HIPAA	56
ISO27001	81

As you can see in the above image, **OWASP Top Ten 2017 classification** does not display all the vulnerabilities that the tool was able to find out. A total number of 17 vulnerabilities are not shown in OWASP Top 10 standard report. We need to look at the **Detailed Scan Report,** in order to identify all the vulnerabilities including the hidden ones.

So let's consider about the report under the standard **"OWASP Top 10 2017**".

As the very first step, let's move on to a brief summary of my targeted web application.



So as you can see in the above image, it was able to identify total number of **65 vulnerabilities.** Among those **56** vulnerabilities are considered as **"Medium Level Risk"** vulnerabilities. There is total number of **5 "Low Level Risk"** vulnerabilities in our targeted web application. So although we like it or not **PayPal has a major security flaw in their web application.**

Netsparker web application security scanner used many advanced attacking and payload injection mechanisms in order to identify vulnerabilities. So most of the vulnerabilities were found out by using attack mechanisms such as Cross-Site Scripting (XSS) and SQL Injection.

1. [Possible] Cross-site Scripting

MEDIUM 54

Netsparker detected Possible Cross-site Scripting, which allows an attacker to execute a dynamic script (*JavaScript, VBScript*) in the context of the application.

This allows several different attack opportunities, mostly hijacking the current session of the user or changing the look of the page by changing the HTML on the fly to steal the user's credentials. This happens because the input entered by a user has been interpreted as HTML/JavaScript/VBScript by the browser. Cross-site scripting targets the users of the application instead of the server. Although this is a limitation, since it allows attackers to hijack other users' sessions, an attacker might attack an administrator to gain full control over the application.

Although Netsparker believes there is a cross-site scripting in here, it could**not confirm it**. We strongly recommend investigating the issue manually to ensure it is cross-site scripting and needs to be addressed.

Impact

There are many different attacks that can be leveraged through the use of XSS, including:

- Hijacking user's active session.
- Changing the look of the page within the victim's browser.
- Mounting a successful phishing attack.
- Intercepting data and performing man-in-the-middle attacks.

Not only that, this tool was able to find out that the web application is vulnerable to **Breach Attacks.**

2. [Possible] BREACH Attack Detected

MEDIUM 1

Netsparker detected that BREACH (Browser Reconnaissance & Exfiltration via Adaptive Compression of Hypertext) attack is possible on this website.

Due to elements that make BREACH attack possible, SSL/TLS protected traffic remains vulnerable and can be attacked to uncover information from the website.

Regardless of which version of SSL/TLS you use, attacks are still possible. Attacks do not require TLS-layer compression and they can work against any cipher suite.

Impact

Even if you use an SSL/TLS protected connection, an attacker can still view the victim's encrypted traffic and cause the victim to send HTTP requests to the vulnerable web server (by using invisible frames). Following these steps, an attacker could steal information from the website and do the following:

- Inject partial plaintext they have uncovered into a victim's requests
- Measure the size of encrypted traffic

Then Netsparker detected that **weak ciphers are enabled during secure communication (SSL).**

3. Weak Ciphers Enabled

MEDIUM 1 CONFIRMED 1

Netsparker detected that weak ciphers are enabled during secure communication (SSL).

You should allow only strong ciphers on your web server to protect secure communication with your visitors.

Impact

Attackers might decrypt SSL traffic between your server and your visitors.

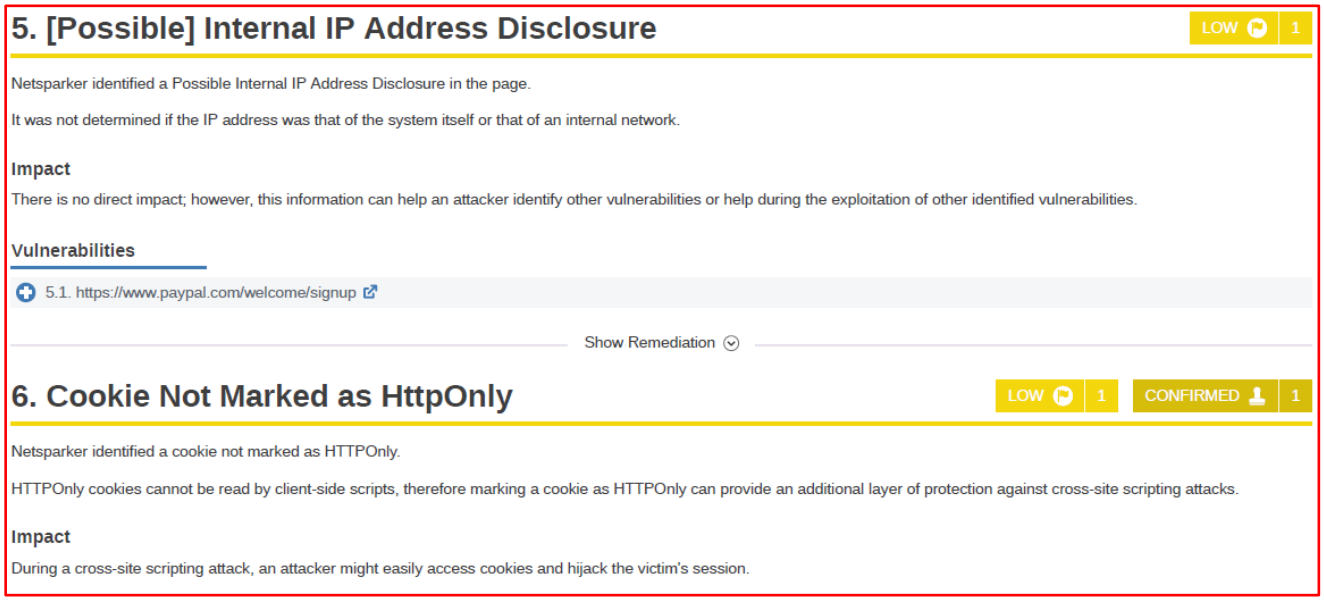
Vulnerabilities

3.1. https://www.paypal.com/ CONFIRMED

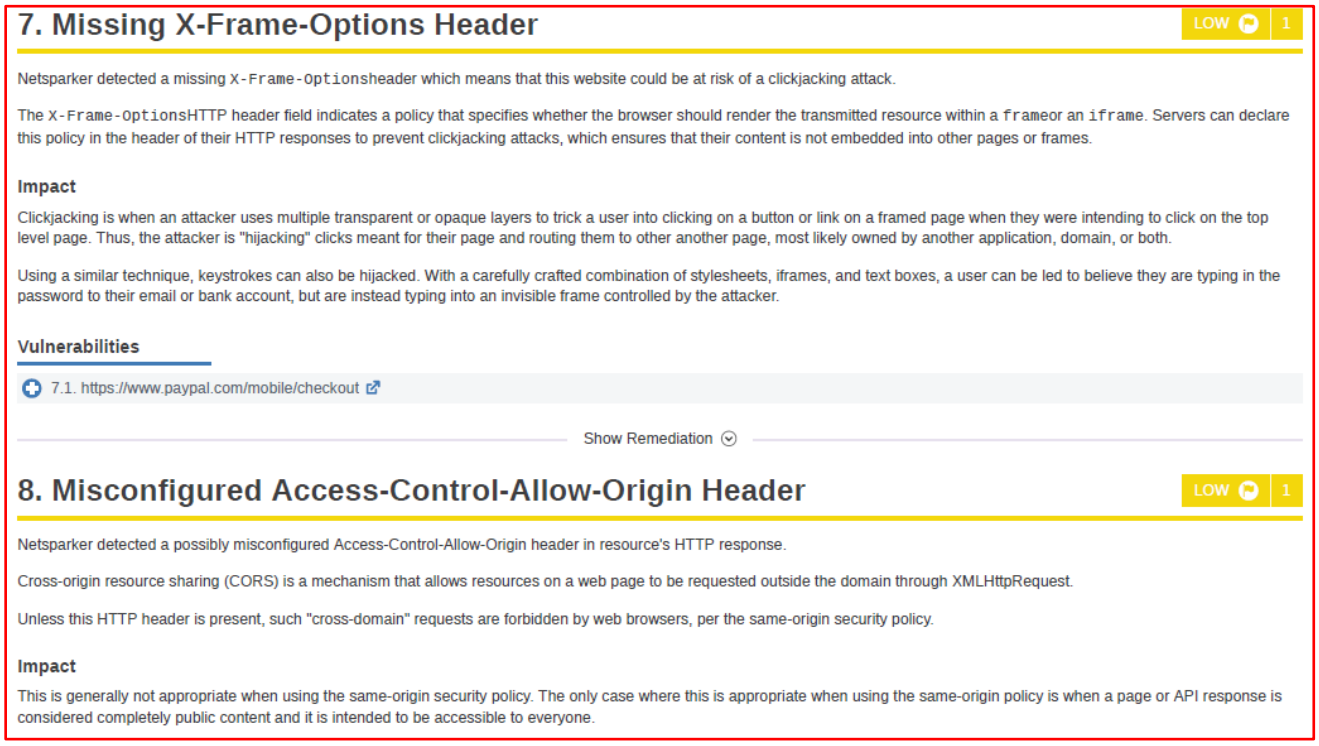
Some of the **web cookies** that are used in the PayPal's web application are identified for **not marking as "HTTPOnly".**

So that the attackers get the oppurtunity to write and read client-side malicious scripts.

Not only that, Netsparker was able to find out an **Internal IP Address Disclosure** also.



So now, let’s move our attention to other types of **Low-Level Severity Vulnerabilities.**



7. Missing X-Frame-Options Header

LOW 1

Netsparker detected a missing `X-Frame-Options`header which means that this website could be at risk of a clickjacking attack.

The `X-Frame-Options`HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a `frame`or an `iframe`. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Impact

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Vulnerabilities

7.1. https://www.paypal.com/mobile/checkout

Show Remediation

8. Misconfigured Access-Control-Allow-Origin Header

LOW 1

Netsparker detected a possibly misconfigured Access-Control-Allow-Origin header in resource's HTTP response.

Cross-origin resource sharing (CORS) is a mechanism that allows resources on a web page to be requested outside the domain through XMLHttpRequest.

Unless this HTTP header is present, such "cross-domain" requests are forbidden by web browsers, per the same-origin security policy.

Impact

This is generally not appropriate when using the same-origin security policy. The only case where this is appropriate when using the same-origin policy is when a page or API response is considered completely public content and it is intended to be accessible to everyone.

Some of those vulnerabilities have **direct and major impact** on the security of the web application.

Impacts

Severity	Impact
Critical	**An attacker could access and control logged in user or admin accounts if attack succeeds** An attacker can abuse a vulnerability on the website to attack individual users of the website. If this attack succeeds, the attacker can hijack their session. This would enable them to take any action that those users can take and to steal their information. For example, an admin might have complete access to the database and the ability to change the website.
High	**An attacker could access user information sent over the internet or public Wi-Fi or a similar environment** This might include passwords, usernames, and the content of web pages viewed.
Medium	**An attacker could use your website to trick your users into providing them with sensitive information** This could include usernames, passwords and credit card details. This could be done by redirecting users from your site to separate web pages that look like your site.
Low	**An attacker could view information about your system that helps them find or exploit vulnerabilities** This may enable them to take control of your website and access sensitive user and admin information. These issues mostly indicates the lack of the security best practice implementation.
Low	**An attacker could access information that helps them to exploit other vulnerabilities** This information gives them a better understanding of your system.

Although the **Critical Level** and **High Severity** Level impacts **cannot be found,** there is still a considerable amount of **Medium Level Severity** Vulnerabilities.

So now let's move on to some of the **Medium Level and Low Level Vulnerabilities** and the **suggested solutions** to prevent from attacks.

Vulnerability	Suggested Action
[Possible] BREACH Attack Detected	**Confirmed soon:** You should fix them soon. Once you've done this, you may want to rescan to check they're gone.
[Possible] Cross-site Scripting	**Confirmed soon:** You should fix them soon. Once you've done this, you may want to rescan to check they're gone.
Weak Ciphers Enabled	**Fix soon:** You should fix them soon. Once you've done this, you may want to rescan to check they're gone.
[Possible] Internal IP Address Disclosure	**Consider fixing after confirmed:** These vulnerabilities aren't very bad but they might help an attacker. You should think about fixing them.
[Possible] Phishing by Navigating Browser Tabs	**Consider fixing after confirmed:** These vulnerabilities aren't very bad but they might help an attacker. You should think about fixing them.
Cookie Not Marked as HttpOnly	**Consider fixing:** These vulnerabilities aren't very bad but they might help an attacker. You should think about fixing them.
Internal Server Error	**Consider fixing:** These vulnerabilities aren't very bad but they might help an attacker. You should think about fixing them.
Misconfigured Access-Control-Allow-Origin Header	**Consider fixing after confirmed:** These vulnerabilities aren't very bad but they might help an attacker. You should think about fixing them.
Missing X-Frame-Options Header	**Consider fixing:** These vulnerabilities aren't very bad but they might help an attacker. You should think about fixing them.
User Controllable Cookie	**Consider fixing:** These vulnerabilities aren't very bad but they might help an attacker. You should think about fixing them.

So I hope that, those proofs are more than enough to realize that there is major security flaw in the PayPal's web application.

I strongly believe that with the time passed by those vulnerabilities might be get fixed by the necessary authorities.

So this is the end of my **Web Audit process** on **https://www.paypal.com** web domain.

So finally I would like to be very thankful to all my lecturers, instructors and friends who supported me a lot in this valuable opportunity.

References :-

Web-Related

https://hackerone.com/paypal?type=team

https://github.com/aboul3la/Sublist3r

https://www.tenable.com/products/nessus

https://www.greenbone.net/

https://github.com/Tuhinshubhra/RED_HAWK

https://kalilinuxtutorials.com/wafw00f

https://github.com/EnableSecurity/wafw00f

https://www.dr-farfar.com/netsparker-professional-full

YouTube-Related

https://www.youtube.com/watch?v=DI8-tNq7hFc

https://www.youtube.com/watch?v=DI8-tNq7hFc

https://www.youtube.com/watch?v=pi1VmNTdzq0

https://www.youtube.com/watch?v=GH9qn_DBzCk

https://www.youtube.com/watch?v=K78YOmbuT48&t=298s

https://www.youtube.com/watch?v=x87gbgQD4eg

https://www.youtube.com/watch?v=fEANg6gyV5A

https://www.youtube.com/watch?v=koMo_fSQGlk

https://www.youtube.com/watch?v=j42I81DWsx8

https://www.youtube.com/watch?v=2YD4vygeghM

https://www.youtube.com/watch?v=QXZKgDizObQ

https://www.youtube.com/watch?v=sQ4TtFdaiRA

https://www.youtube.com/watch?v=4t4kBkMsDbQ

https://www.youtube.com/watch?v=W0KRYkZppIw

https://www.youtube.com/watch?v=Q8hAjEaAgE8&t=238s

https://www.youtube.com/watch?v=EmWXfq51pE0

https://www.youtube.com/watch?v=0tfrkiMZEiM

https://www.youtube.com/watch?v=PRKf4m_hE7U

https://www.youtube.com/watch?v=4C2xbZNf9YU&t=206s