Sri Lanka Institute of Information Technology

Web Security
IE 2062

Assignment – Web Audit

Final Submission

Student Number	Name
IT19102924	Jayasinghe J.G.L.A

Terms of Reference

This report, about a Web Audit focusing the associated domain is submitted in fulfillment of the requirements for module Web Security (IE2062), Department of Computer Systems Engineering, Sri Lanka Institute of Information Technology.

Acknowledgment

I wish to express my sincere gratitude to Dr. Lakmal Rupasinghe, lecturer in charge of the module IE 2062 – Web Security for his invaluable advices and guidance which was vital towards initiating this web audit.

I would also like to thank the co-lecturer Ms. Chethana Liyanapathirana], Ms. Laneesha Rugghahakotuwa, and Ms. Chathu Udagedara for their immense support and guidance on this web audit.

Table of Contents

Note

Security is now one of the major parts of web development, as both website owners and users need to absolutely guarantee that their personal information is protected. It seems almost every day there is a story in the Newsline about a newly hacked website or a web-app. Until it happens to most people, they think that they will never be victims. In any case, hackers can scope out our website to find any flaw that they can use for their malicious activities to use our website and/or server. It is important that we conduct a security audit on our website/web-app if we are not willing to fall victim to their tactics.

1. What is a Web Security Audit ?

A website security audit is the process that tests the web framework, including vulnerabilities and security foundation, extensions, templates, and other infrastructure. Usually, a comprehensive web security audit includes static & dynamic review of code, data model error checking, configuration checking, etc. Both hidden vulnerabilities on the website and security framework are included in Website Security audits and are normally followed by a penetration test. While the aim of a security audit is to analyze and locate vulnerable areas, a penetration test focuses on exploiting them. Penetration Tests are more like emulating the scenario of a hacker and a real-life attack and leveraging the vulnerabilities to determine the danger associated with each vulnerability.

Types of Web Security Audits:-

- Vulnerability Scanner
- Automated Security Audits
- Manual Security Audits
- Professional Security Audit

1.1 How should we initiate a Web Security Audit ?

A robust web security audit provides an assessment of security policies, security controls, and possible risks associated with all IT assets including websites and mobile apps. Automated

software and human expertise are used for the most reliable security audits. Information gathering and Exploitation are considered as the initiatives of a web security audit.

Listed below can be considered as some important aspects and methodologies concerning the starting and the continuation of a web security audit.

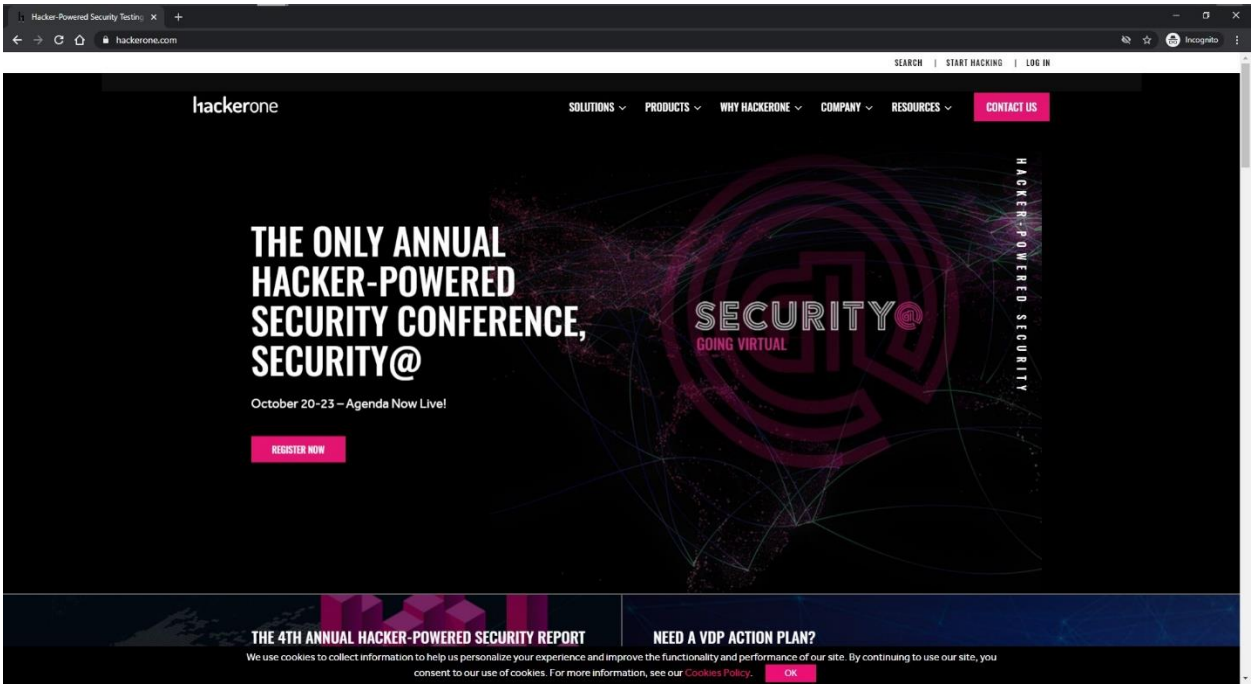
- Host Discovery
- Inspecting Services
- Site Structuring
- Website/App Analysis
- Code Assessment
- Vulnerability Research
- Attack Method Selection
- Vulnerability Confirmation
- Remediation

2. Selecting a Program

As per the guidelines for the assignment, we were given some platforms which had active bug bounty programs initiated through their platforms. Listed below are some of them.

- Hackerone
- Bug Crowd
- Cobalt
- Synack
- Facebook
- Google

After careful consideration, I opted to select “Hackerone” platform to pick a domain since I was already a user and had a quite good exposure on how that platform works.

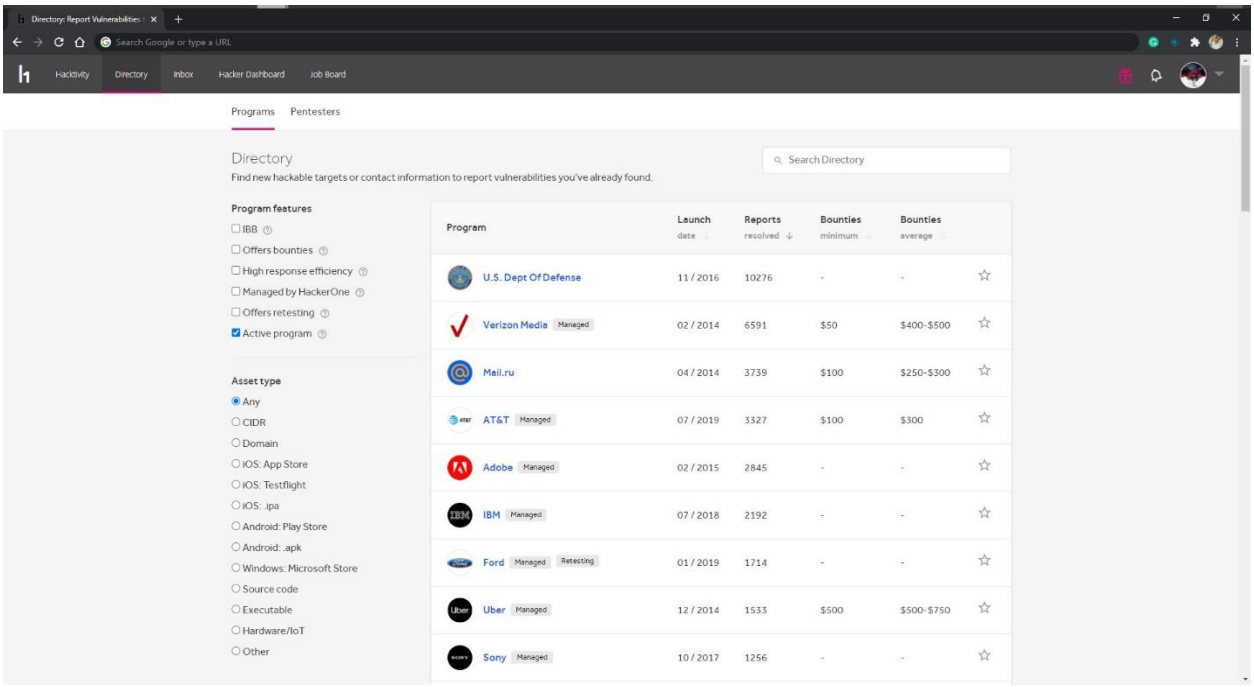


- HackerOne is a vulnerability coordination and bug bounty platform that connects businesses with penetration testers and cybersecurity researchers. It was one of the first companies, along with Synack and Bug crowd, to embrace and utilize crowd-sourced security and cybersecurity researchers as linchpins of its business model; it is the largest

cybersecurity firm of its kind. As of May 2020, Hackerone's network had paid $100 million in bounties.

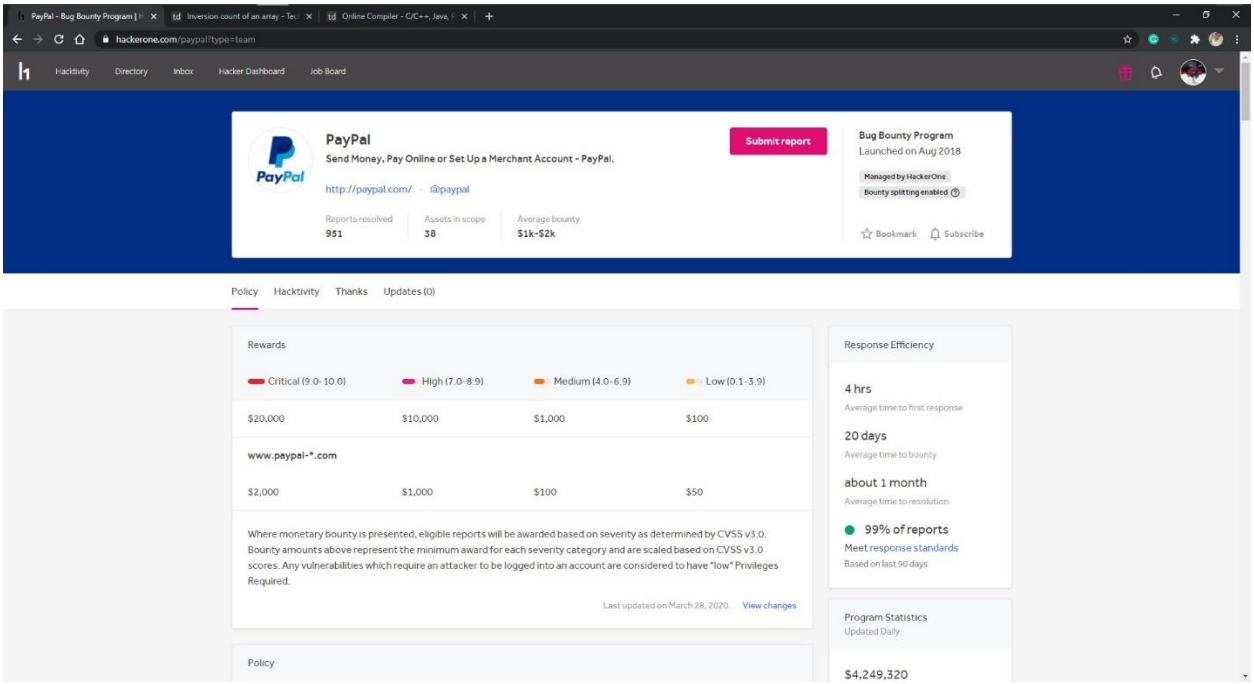
2.1 Domain Selection

When we access the directory page on www.hackerone.com, we can see the listed domains sorted in various program features and types. Each domain has its own unique specifications and policies which they allow us to do the working. Here are some of the domains visible on the top of the page according to the resolve involvement.



Here, I can sort the domains and programs by asset type and program features which will change the domains, respectively. More specifically the launch date, the number of reports resolved, and the bounty prices are also displayed in this section which enables us to sort them as our wish and go through them in the selection process.

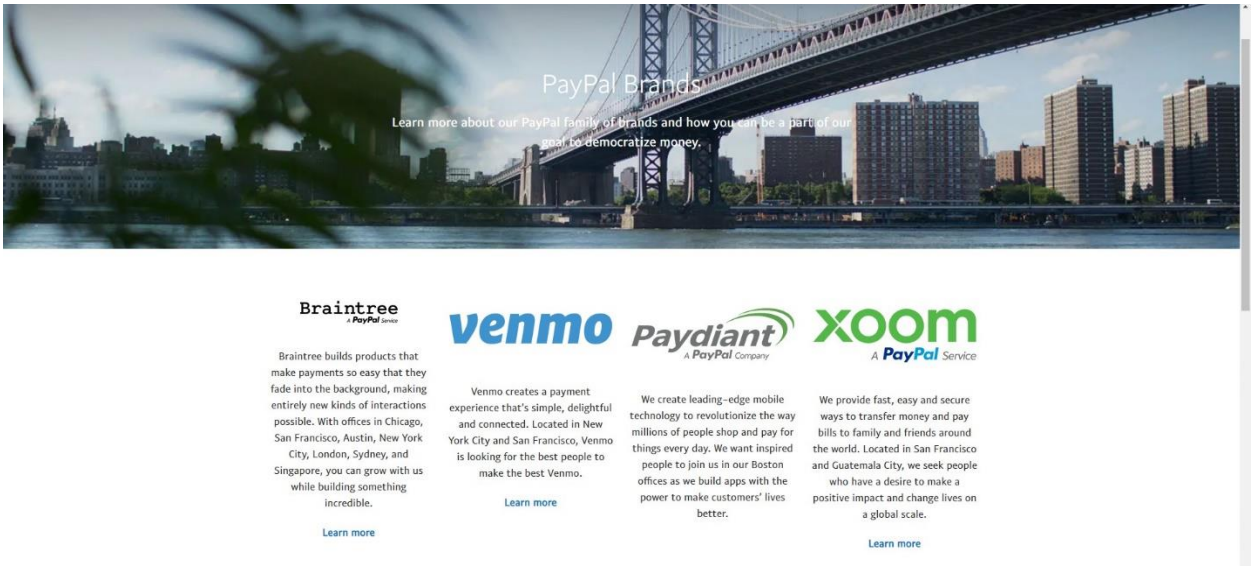
After going through and observing the respective scopes, rules of engagement, and policies my decision was to go forward with the domain "http://paypal.com/". When I access the bug bounty program offered by PayPal, it was clearly stated the requirements and guidelines which has to be followed when studying the system.



2.2 Exploring http://paypal.com/

After registering the PayPal domain, it was needed to refer the terms, policies, disclosures, bounties, and specially in-scope and out-scope vulnerabilities that they expect us to perform. The most important factors were the in-scope and out-scope vulnerabilities and the associated domains with them. The special case with PayPal is that they also have included five of their brand names to this hackerone bounty program which makes the domain broader. The listed brand names were,

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



2.3 Identifying Core Points

Before starting to go through the technical facts, it is important to identify and study the program terms, bug submission requirements, RCE Guidelines, how the bounty payments are received, and the policies related to ownership, confidentiality, and termination. They will play an important role in improving our chance of earning a bug bounty if we are to go and exploit their system as per their rules and regulations.

Considering the nature of the assignment and the rules of the bounty program, I will not be going deeply into those points deeply in this report even though a comprehensive analysis is done focusing each of those mentioned points individually.

2.4 Scope for Web Applications

PayPal has offered scopes for vulnerabilities both pertaining to Web Applications and Mobile Applications. Here with respective to this web security audit assignment, I will be focusing more about the web application vulnerabilities.

2.4.1 In-Scope Vulnerabilities

Accepted, in-scope vulnerabilities include,

- 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

2.4.2 Out-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:

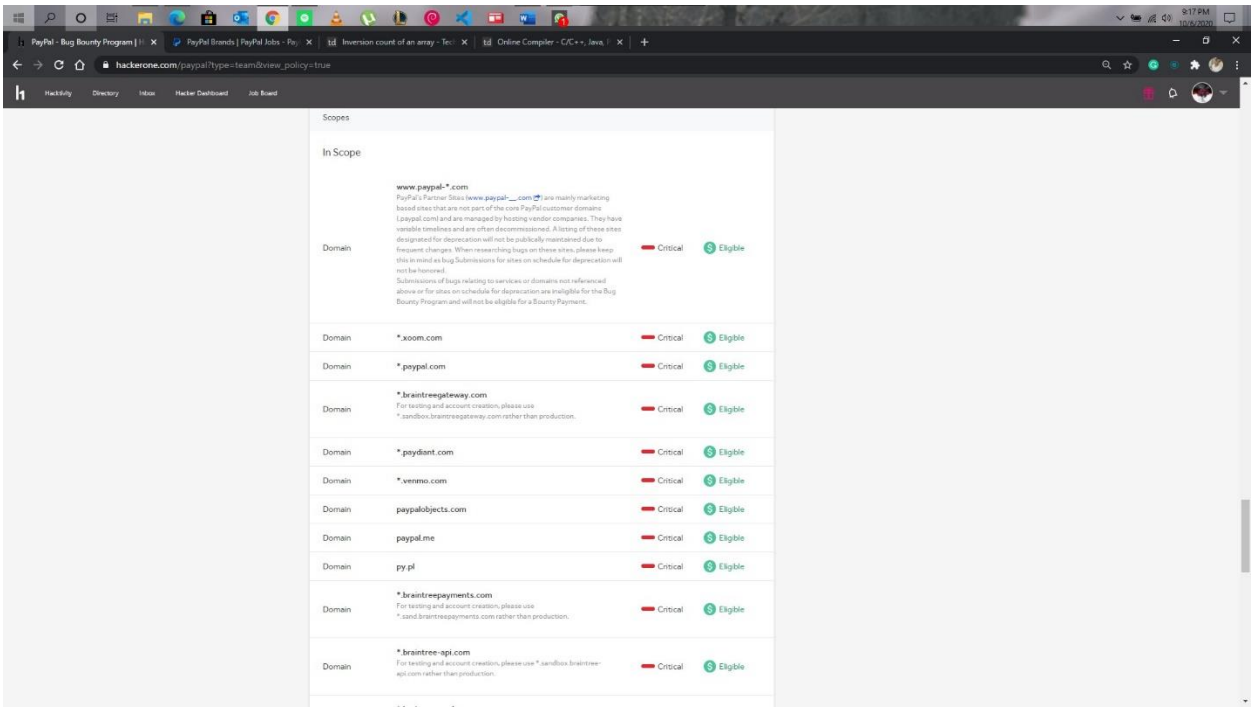
- Any physical attacks against PayPal property or data centers
- Reports that involve a secondary user account where an existing business relationship is being leveraged and the impact is limited solely to the parent account
- Username enumeration on customer facing systems (i.e. using server responses 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
- 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.)
- 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)

It is of great importance to through each of these points separately because each means a unique and a significant indication of what we are expected to and not do.

2.4.3 In-Scope Domains

Since PayPal has offered some of their brand names under this program, some of the offered domains by PayPal might not be applicable to my scope as they are acting as separate domains even though they are included in the In-Scope list of domains in this program. This will be demonstrated in the illustration video providing facts.



These main domains mentioned here contain the subdomains of the companies which are owned by PayPal under their brand name.

E.g.

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

As mentioned above, these will not be the focus in this web audit. Moreover, sub domains under PayPal will be focused depending on the number of subdomains and the content inside them.

3. Defining the Scope

It is now evident that this selected domain has a huge diversity. So, it is important that I select a proper methodology and set a scope which I can adhere to when continuing with the audit. I will be using OWASP Top 10 checklist when doing the auditing. Even though there might seem a little conflict with the out-scope vulnerabilities with PayPal's bug bounty program being in the OWASP checklist, since the focus is around the Web Security Audit as required in the assignment, I believe I am eligible to audit and document them.

3.1 What is OWASP ?

OWASP stands for the Open Web Application Security Project, an online community that produces articles, methodologies, documentation, tools, and technologies in the field of web application security.

3.2 OWASP Top 10

The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications. The following list will provide the OWASP top 10 checklist with a brief introduction describing what it means in simple terms.

1. **Injection**. Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
2. **Broken Authentication**. Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
3. **Sensitive Data Exposure**. Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes.

Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

4. **XML External Entities (XXE)**. Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
5. **Broken Access Control**. Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
6. **Security Misconfiguration**. Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information.
7. **Cross-Site Scripting XSS**. XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
8. **Insecure Deserialization**. Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
9. **Using Components with Known Vulnerabilities**. Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.
10. **Insufficient Logging & Monitoring**. Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.

4. Information Gathering

Information Gathering and getting to know the target systems is the first process in any exploitation or hacking. Reconnaissance is a set of processes and techniques (Footprinting, Scanning & Enumeration) used to covertly discover and collect information about a target system.

4.1 Initiating Reconnaissance

Information Gathering and getting to know the target systems is the first process in any exploitation or hacking. Reconnaissance is a set of processes and techniques (Footprinting, Scanning & Enumeration) used to covertly discover and collect information about a target system. During recon, the attacker tries to collect as much information about a target system as possible. Some of the common methodologies can be listed as follows.

- Gather initial info.
- Identifying the range of the network
- Identify active nodes
- Find open ports and access points
- Fingerprinting the O/S
- Mapping the network

We can also divide Recon into two parts such as,

- Active Reconnaissance
- Passive Reconnaissance

4.1.1 Active Reconnaissance

Active reconnaissance is a type of computer attack, in which an attacker communicates with the targeted device to capture vulnerability information. This can be achieved by automated or manual scanning through different tools such as ping, traceroute, netcat etc. The attacker must communicate with the object of this form of recon. This recon is quicker and precise but makes a lot of noise as well. But since attacker needs to communicate with the target to obtain information, a firewall or one of the network security appliances can catch up with the recon. (Firewalls network, intrusion detection systems, etc.)

4.1.2 Passive Reconnaissance

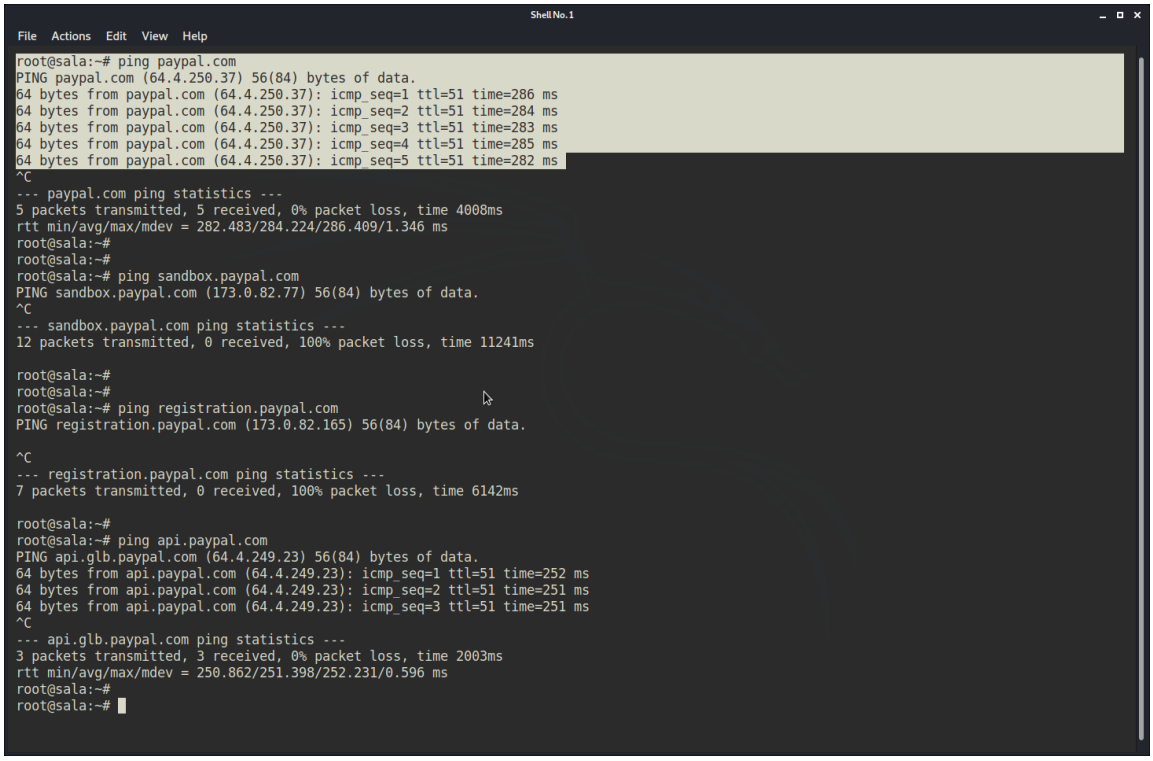
Passive reconnaissance aims to obtain knowledge about targeted machines and networks without intervening in applications actively. It collects data without alerting the victim. If the victim's host has been alerted, then protection against the attack is greatly improved.

4.2 First look at https://paypal.com

Without jumping directly to scans and recon, it seemed important to identify how PayPal responds when we look it up on the Linux terminal.

4.2.1 Ping paypal.com

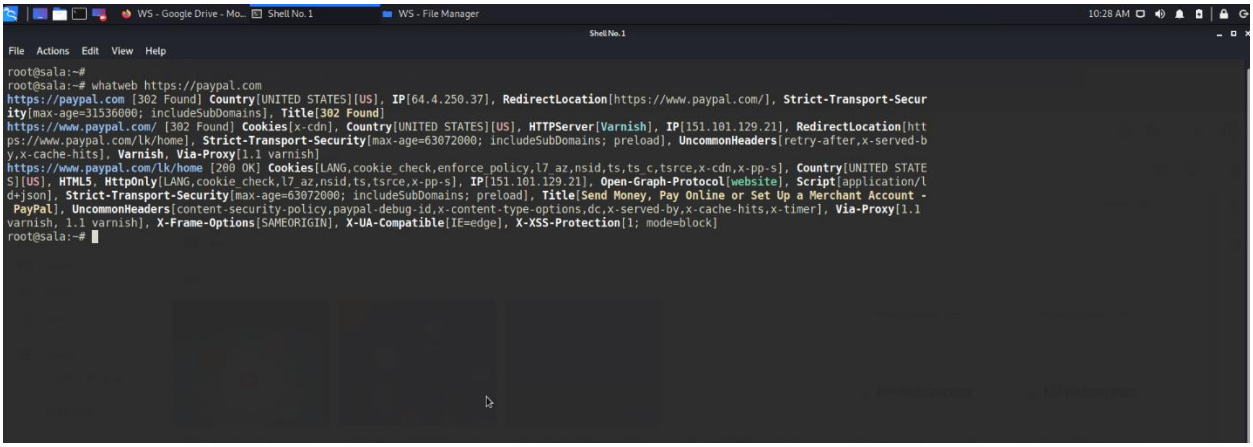
This is the output it displays when I issue the command “ping paypal.com” in the Linux terminal.



In this screenshot, it has also mentioned about some random subdomains of paypal.com and their respective IP addresses after trying to echo through their servers. It can be noted that the IP address of the paypal.com seems to be 64.4.250.37, but it also redirects to 64.4.250.36 address occasionally.

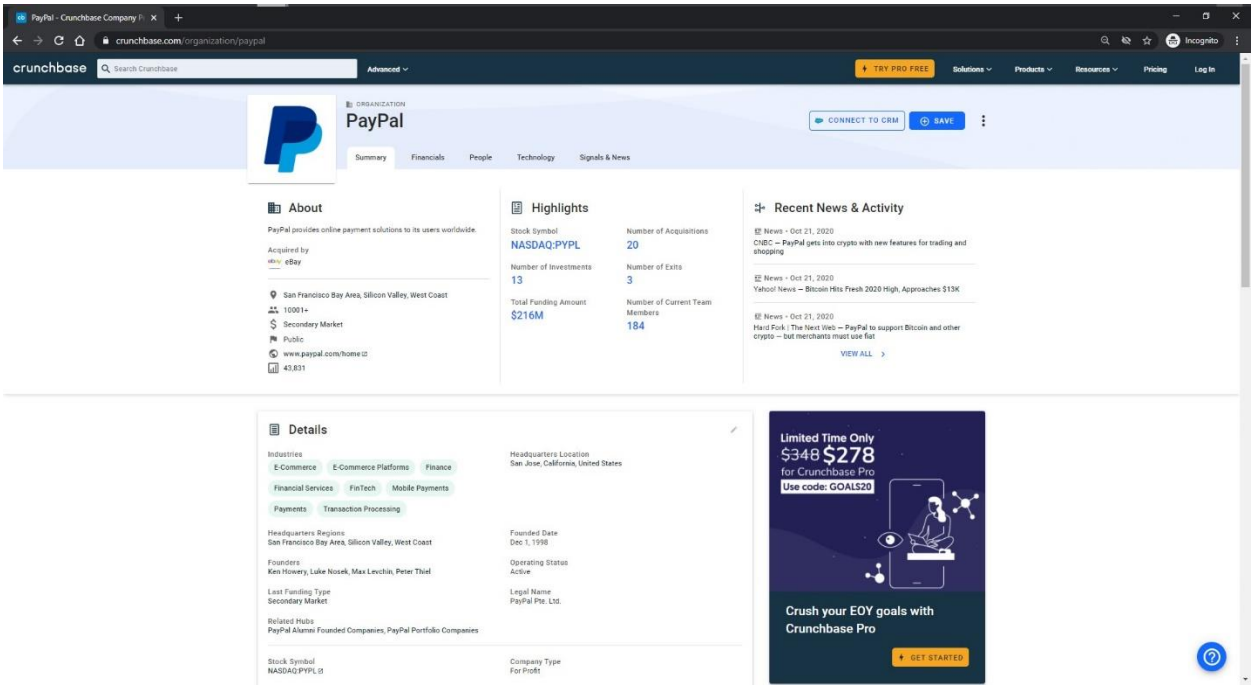
4.2.2 Identifying technologies used by PayPal

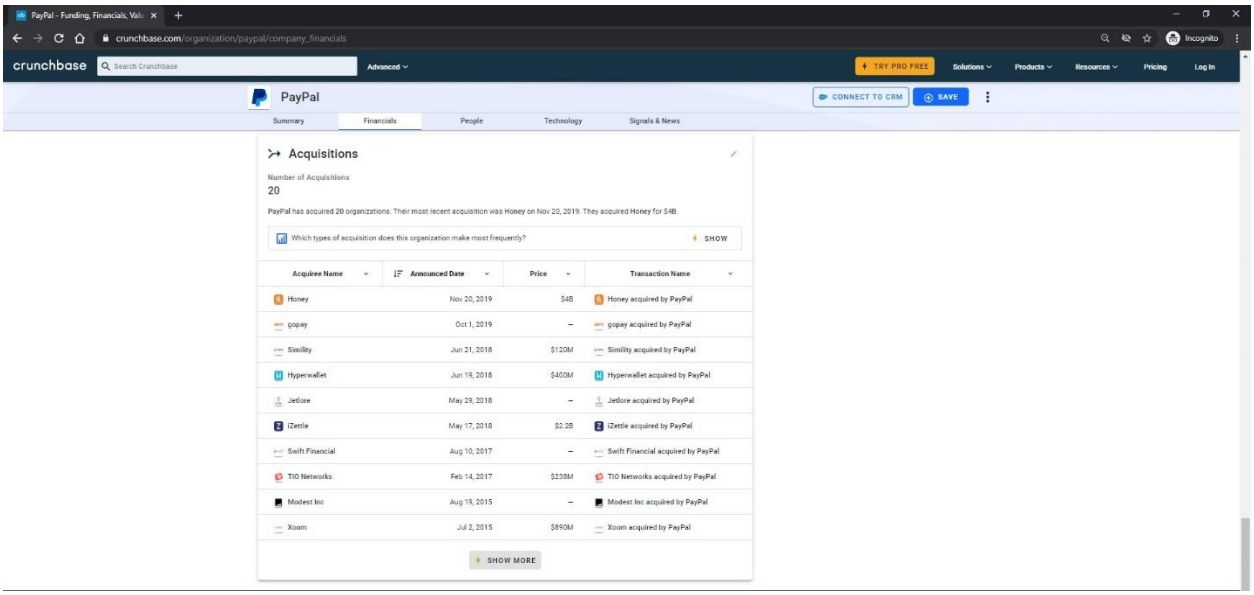
Using the tool Whatweb and passing parameters we can get the details about what the website is, and the technologies associated with it. By advanced passing, we can get version numbers, email addresses, etc. It can be noted that the Web Server associated is **Varnish**



4.3 Exploring the Acquisitions of PayPal

It is also important to get a basic idea about the company we are dealing with and what other collaborators are associated with it. With Crunchbase, many useful details can be listed and noted. It also provides the structure of the organization.





It can be noted that there are 20 acquisitions related to the PayPal organization. Since this is a paid web app, we need to get the subscription to go through all the acquisitions in the list.

More importantly, both **Xoom** and **Swift Financial** are also in this list which were also there under the brand names of PayPal offered in the bug bounty program on Hackerone.

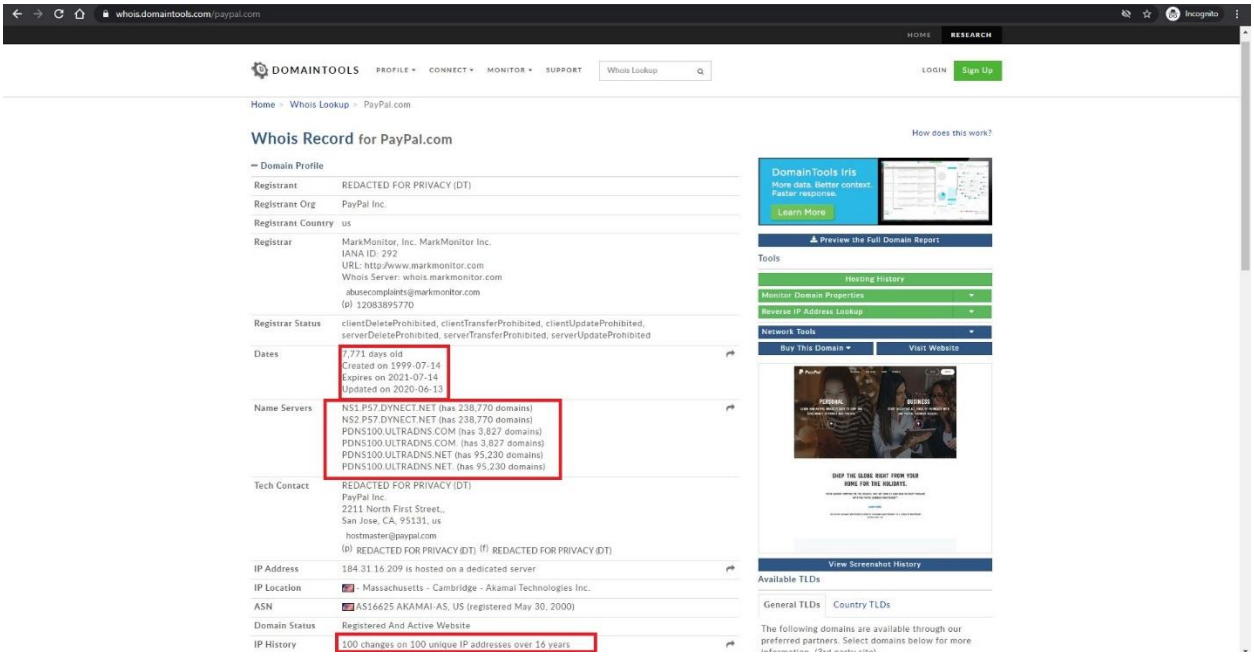
Investors, Management group, App metrics and Financial information are also retrieved from their respective servers, so that one can get details about both financial and technological summary from Crunchbase.

4.4 Domain Name Information

We can use "whois" feature in www.domaintools.com to get more information about the domain name information including its owner, its registrar, date of registration, expire, name server, owner's contact information, etc.

Important notes

- IP History
- Dates
- Registrar
- Name Servers
- ASN
- Hosting History



4.4.1 Full Whois Record (last updated 22.10.2020)

```
Domain Name: paypal.com
Registry Domain ID:
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2020-06-13T02:21:11-07:00
        2020-06-13
Creation Date: 1999-07-14T22:32:11-07:00
        1999-07-15
Registrar Registration Expiration Date: 2021-07-14T00:00:00-07:00
        2021-07-15
Registrar: MarkMonitor, Inc.
        MarkMonitor Inc.
Sponsoring Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: 12083895770
Status:
        clientDeleteProhibited
        clientTransferProhibited
        clientUpdateProhibited
        serverDeleteProhibited
        serverTransferProhibited
        serverUpdateProhibited
Registry Registrant ID:
Registrant Name: REDACTED FOR PRIVACY (DT)
Registrant Organization: PayPal Inc.
Registrant Street: 2211 North First Street,
Registrant City: San Jose
```

```
Registrant State/Province: CA
Registrant Postal Code: 95131
Registrant Country: us
Registrant Phone: 18882211161
Registrant Phone Ext:
Registrant Fax: 14025375774
Registrant Fax Ext:
Registrant Email: hostmaster@paypal.com
Registry Admin ID:
Admin Name: REDACTED FOR PRIVACY (DT)
Admin Organization: PayPal Inc.
Admin Street: 2211 North First Street,
Admin City: San Jose
Admin State/Province: CA
Admin Postal Code: 95131
Admin Country: us
Admin Phone: REDACTED FOR PRIVACY (DT)
Admin Phone Ext:
Admin Fax: REDACTED FOR PRIVACY (DT)
Admin Fax Ext:
Admin Email: hostmaster@paypal.com
Registry Tech ID:
Tech Name: REDACTED FOR PRIVACY (DT)
Tech Organization: PayPal Inc.
Tech Street: 2211 North First Street,
Tech City: San Jose
Tech State/Province: CA
Tech Postal Code: 95131
Tech Country: us
Tech Phone: REDACTED FOR PRIVACY (DT)
Tech Phone Ext:
Tech Fax: REDACTED FOR PRIVACY (DT)
Tech Fax Ext:
Tech Email: hostmaster@paypal.com
Registry Billing ID:
Billing Name:
Billing Organization:
Billing Street:
Billing City:
Billing State/Province:
Billing Postal Code:
Billing Country:
Billing Phone:
Billing Phone Ext:
Billing Fax:
Billing Fax Ext:
Billing Email:
Nameservers:
        ns1.p57.dynect.net
        ns2.p57.dynect.net
        pdns100.ultradns.com
        pdns100.ultradns.com.
        pdns100.ultradns.net
        pdns100.ultradns.net.
```

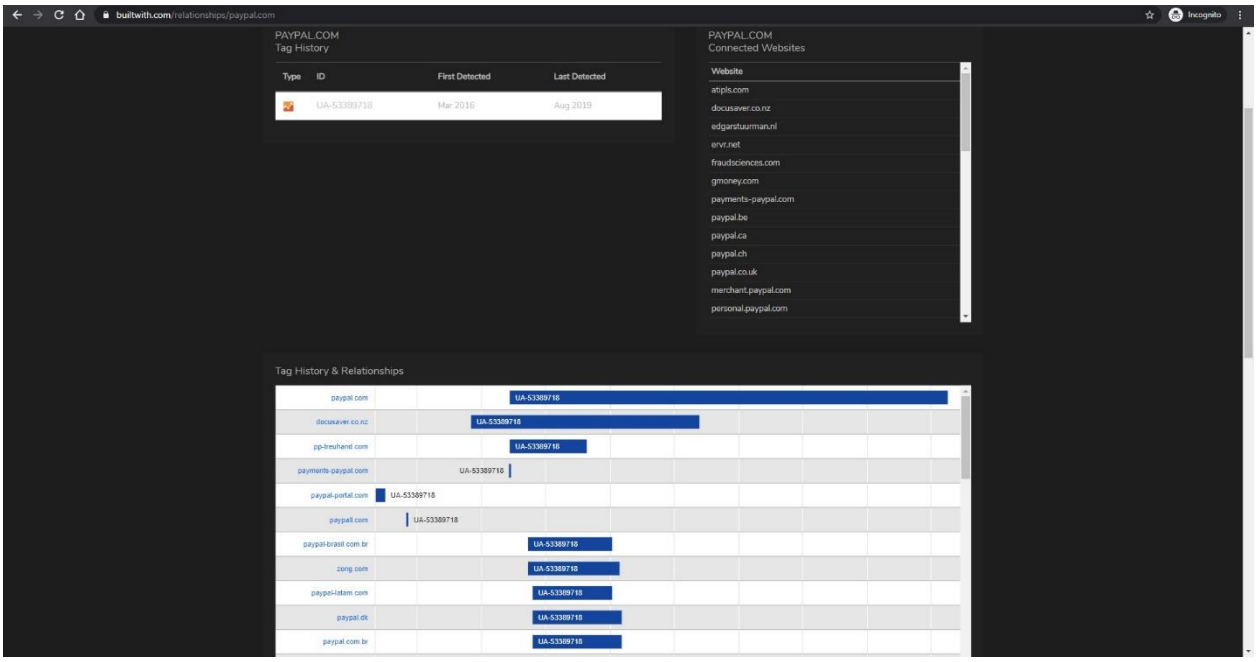
```
Registry ID: 8017040_DOMAIN_COM-VRSN
DNSSEC: signedDelegation
```

4.4.2 JSON Api for PayPal on whoxy.com

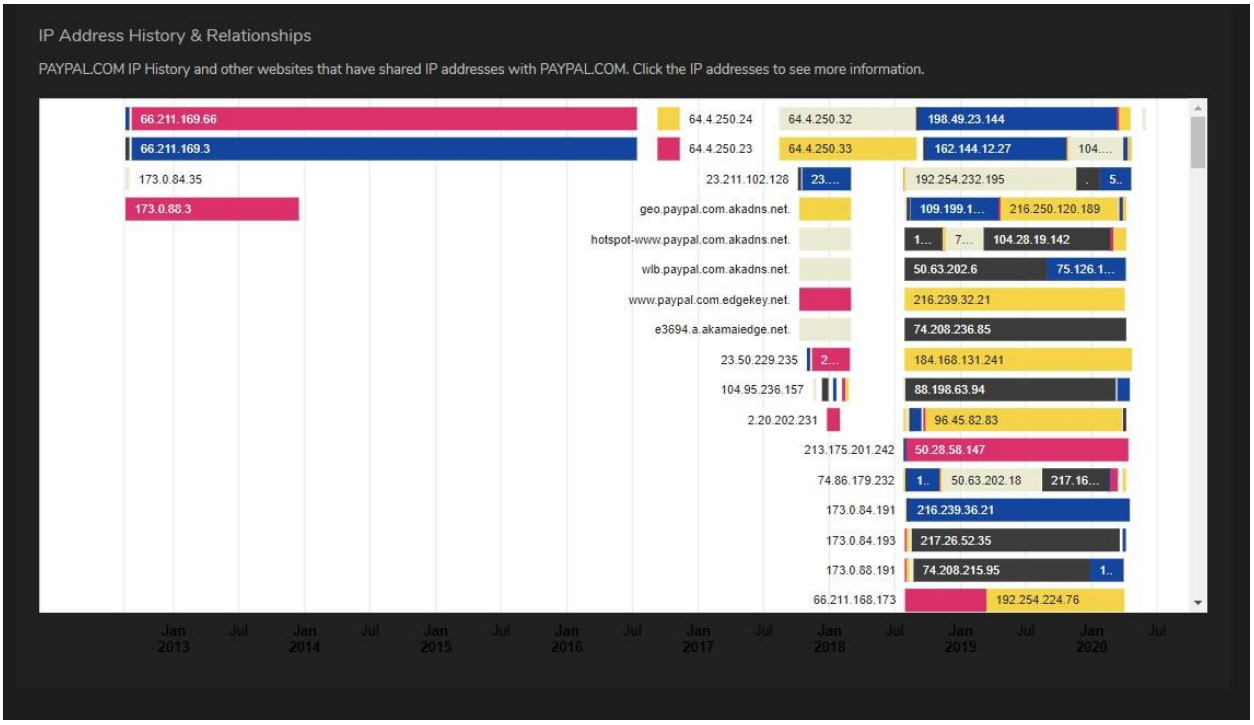
This demonstrate only a part of the PayPal API look up when accessed via whoxy.com


```
{
    "status": 1,
    "domain_name": "paypal.com",
    "query_time": "2020-10-06 12:39:07",
    "whois_server": "whois.markmonitor.com",
    "domain_registered": "yes",
    "create_date": "1999-07-15",
    "update_date": "2020-06-13",
    "expiry_date": "2021-07-14",
    "domain_registrar": {
        "iana_id": 292,
        "registrar_name": "MarkMonitor, Inc.",
        "whois_server": "whois.markmonitor.com",
        "website_url": "http://www.markmonitor.com",
        "email_address": "abusecomplaints@markmonitor.com",
        "phone_number": "+1.2083895770"
    },
    "registrant_contact": {
        "full_name": "Domain Administrator",
        "company_name": "PayPal Inc.",
        "mailing_address": "2211 North First Street,",
        "city_name": "San Jose",
        "state_name": "CA",
        "zip_code": "95131",
        "country_name": "United States",
        "country_code": "US",
        "email_address": "hostmaster@paypal.com",
        "phone_number": "+1.8882211161",
        "fax_number": "+1.4025375774"
    },
    "administrative_contact": {
        "full_name": "Domain Administrator",
        "company_name": "PayPal Inc.",
        "mailing_address": "2211 North First Street,",
        "city_name": "San Jose",
        "state_name": "CA",
        "zip_code": "95131",
        "country_name": "United States",
        "country_code": "US",
        "email_address": "hostmaster@paypal.com",
        "phone_number": "+1.8882211161",
        "fax_number": "+1.4025375774"
    },
```

4.4.3 Ad / Analytics Relationships



The below figure depicts PAYPAL.COM IP History and other websites that have shared IP addresses with PAYPAL.COM



1.1

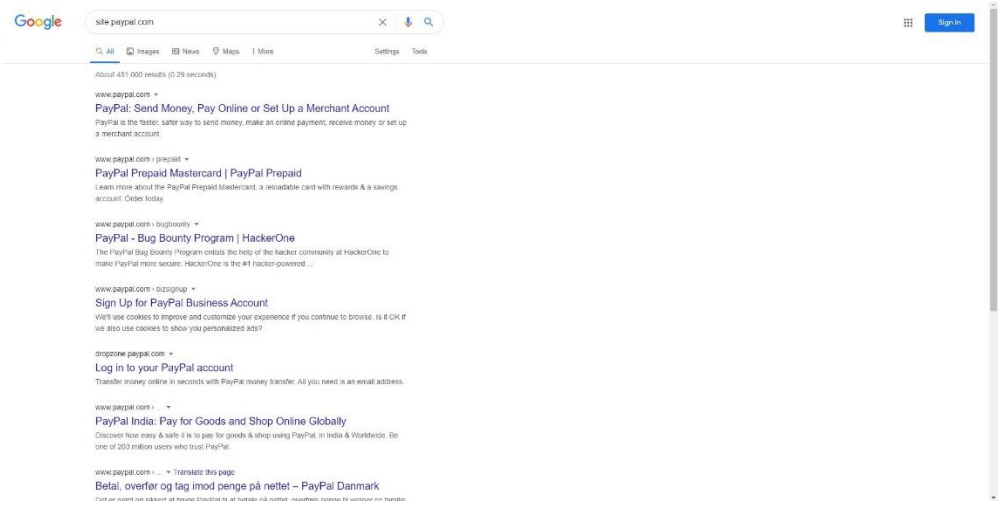
4.5 Google – Fu on PayPal

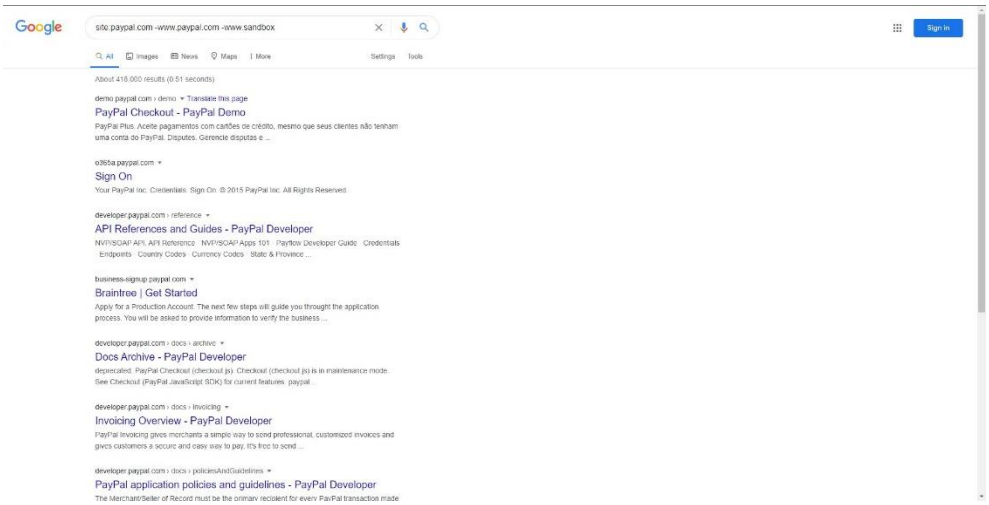
Up to now, we have only touched the administrative and financial details related to PayPal organization. In this step I will be using Google Dorks to play around https://paypal.com before starting the identification of subdomains, just to see what Google Search Engine offers.

What is Google-Fu?

Google-Fu is defined to be a skill used in the Google Search Engine to have access to a lot of accurate information(results)

4.5.1 Retrieving the results to display the sites of paypal.com

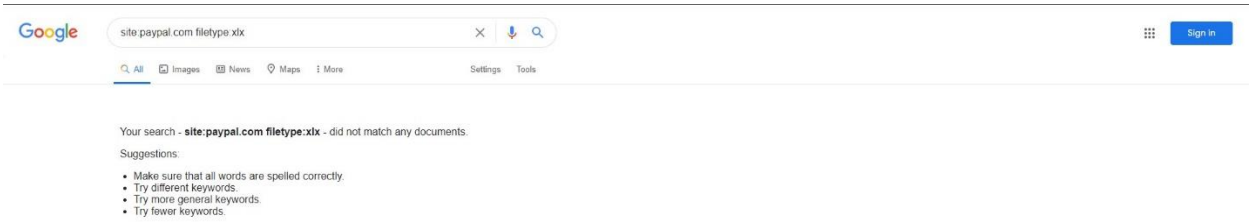




Both the figures above shows the results we get when we search for the site paypal.com and secondly searching the site omitting the popular subdomains such as www.paypal.com and www.sandbox.payal.com as they seem more popular. In that case we can go through many subdomains and URL's related to the web system.

4.5.2 Trying the hidden file formats and specific servers

I could understand that PayPal has made more secure that I was not able to find URL's revealing such search parameters. However, we can manipulate these strings as we like differently and try many more combinations to crawl deep and gather information just using Google Search Engine.

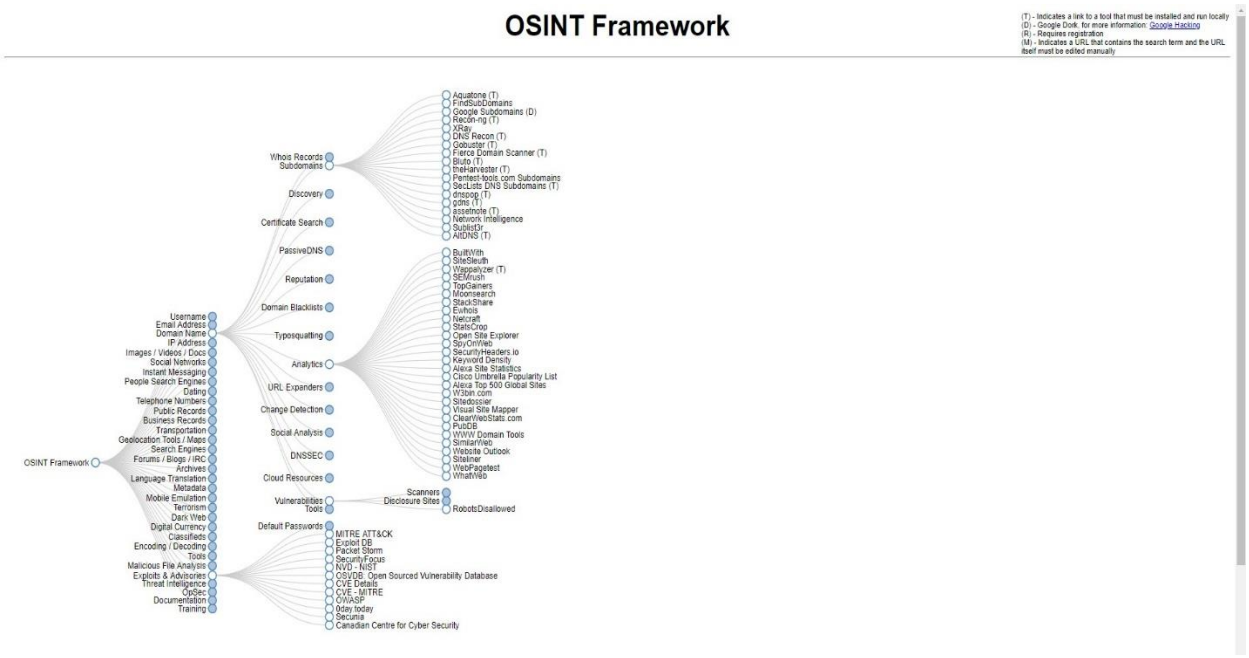


4.6 OSINT Framework

Before moving into practicing tools, it is important to understand about OSINT Framework that I most of the tools I use on the path to reconnaissance will belong to OSINT.

4.6.1 What is OSINT Framework

Open source intelligence or OSINT Framework, as its name implies, is a cybersecurity framework, a collection of OSINT tools to make your intel and data collection tasks easier. It provides a simple web-based interface that allows you to browse different OSINT tools filtered by categories.

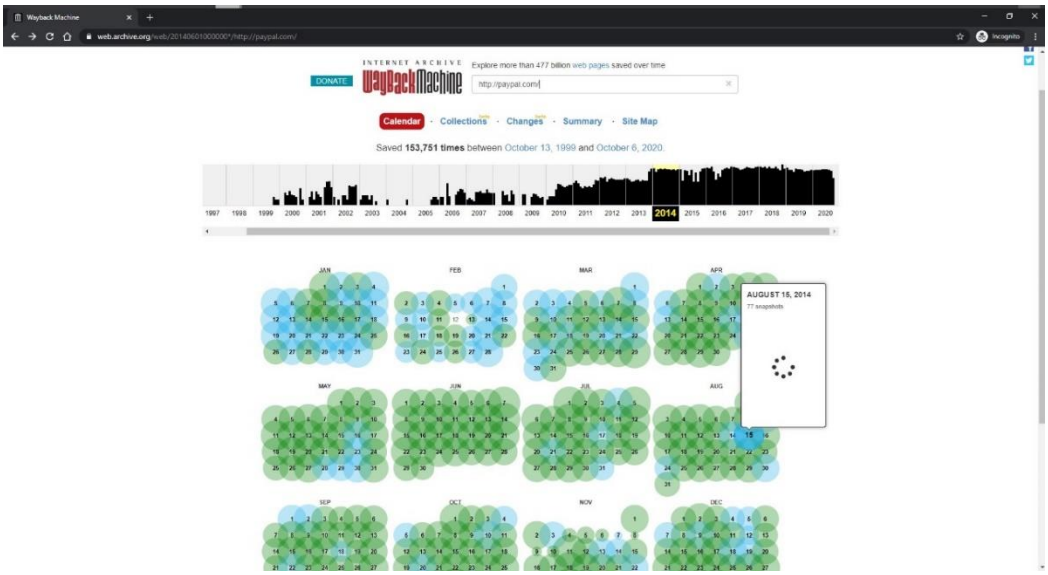


It also provides an excellent classification of all existing intel sources, making it a great resource for knowing what infosec areas you are neglecting to explore, or what will be the next suggested OSINT steps for your investigation.

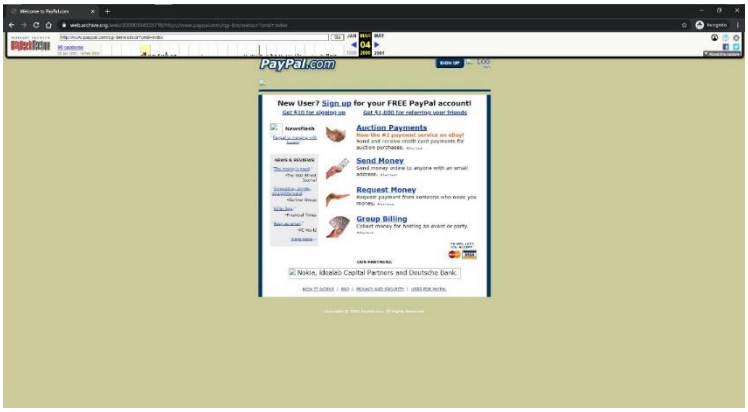
This acts as a tree in such a way it expands revealing tools that are highly operational according the way we sort them.

4.7 Going back in Time

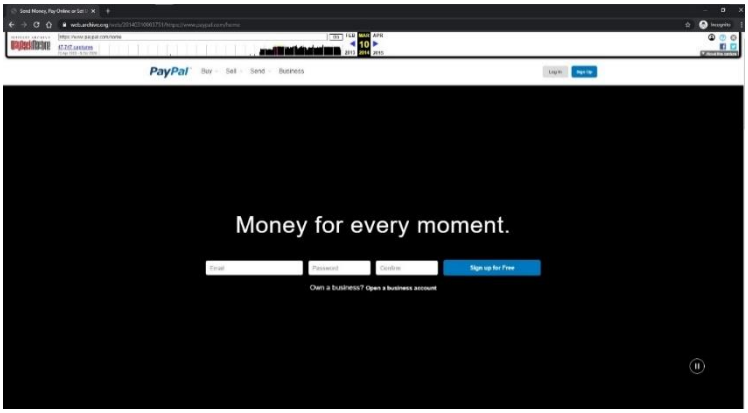
Let us see how https://paypal.com has been evolving since the day it was launched. It helps us understand how these systems have adapted to latest technologies and integrated their platforms to compete with the latest trends. For this, we can use the internet archive or the archive.org to go back in times and observer how this site looks back in time. Using “Wayback Machine” we can search for the relevant URL’s and get the results.



This shows how the PayPal website has evolved over time and we can access the available timelines and get the respective site images loaded. I will be using 2 timelines to observe how PayPal has managed to upgrade their web growth



How PayPal looked as of March 4th in 2000



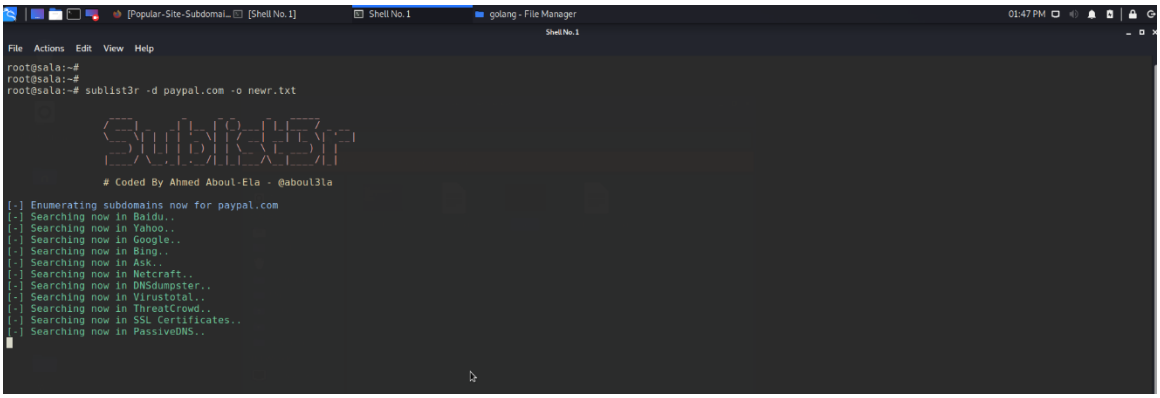
How PayPal looked as of March 10th in 2014

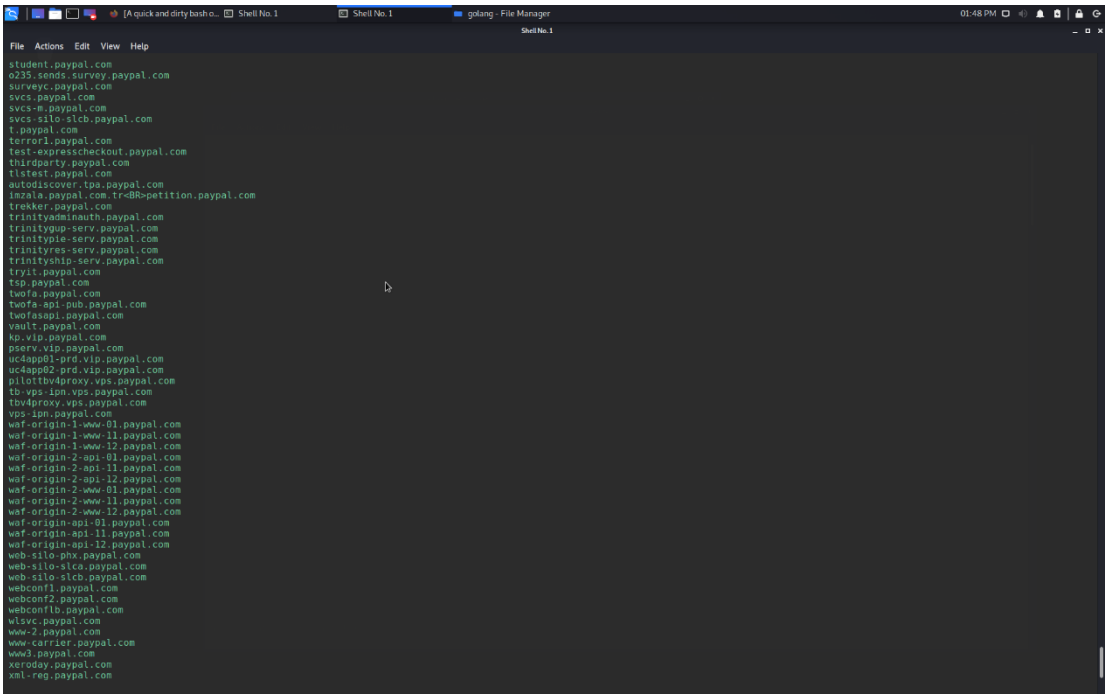
4.8 Subdomain Enumeration

PayPal consists with thousands of various subdomains related. So it is important to take not of what exactly are the sub domains associated and what sites are up and running filtering http and https protocols.

4.8.1 Using Sublist3r to retrieve the sub domains

Command – Sublist3r -d paypal.com -o newr.txt

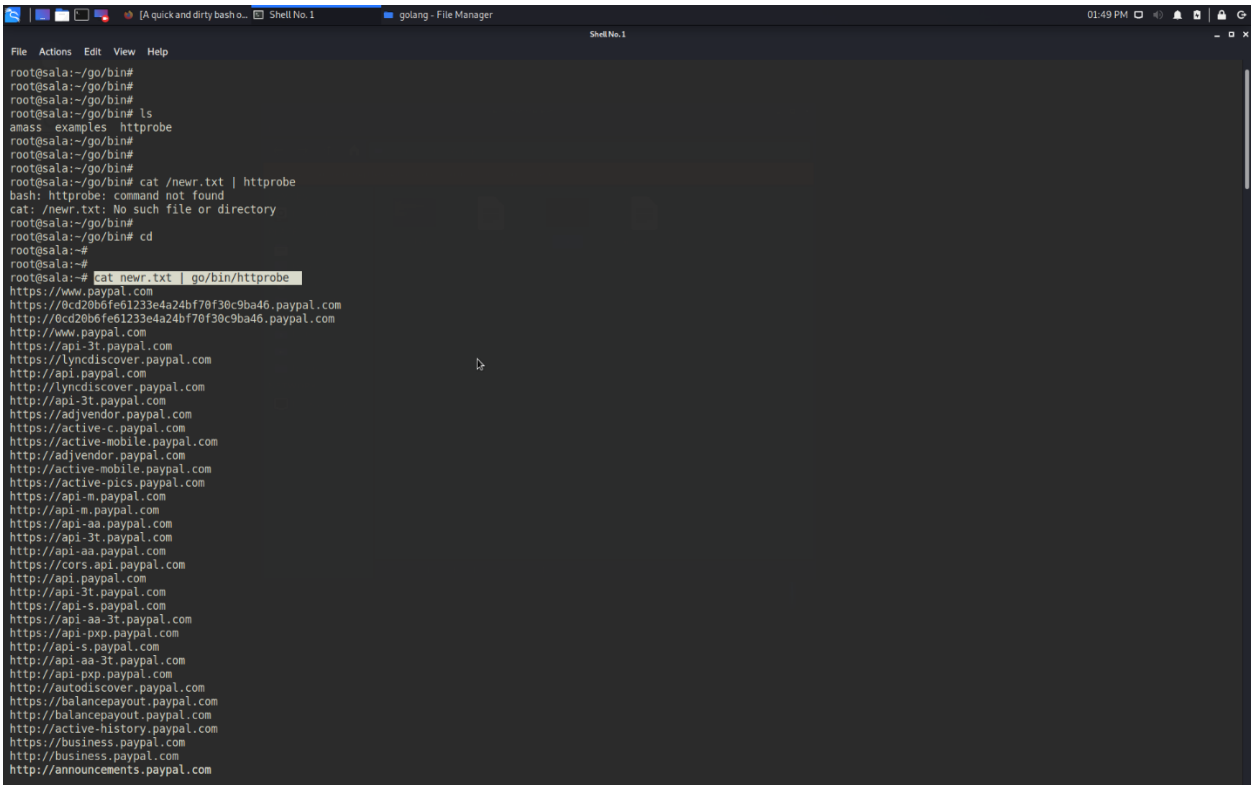




Since this command outputs thousands of sub domains referencing various services, endpoints, and API's, the results are directly sent to a txt file name "newr.txt", so that it eases the task.

4.8.2 Using httprobe to retrieve the live sub domains (http / https)

Command – cat newr.txt | go/bin/httprobe



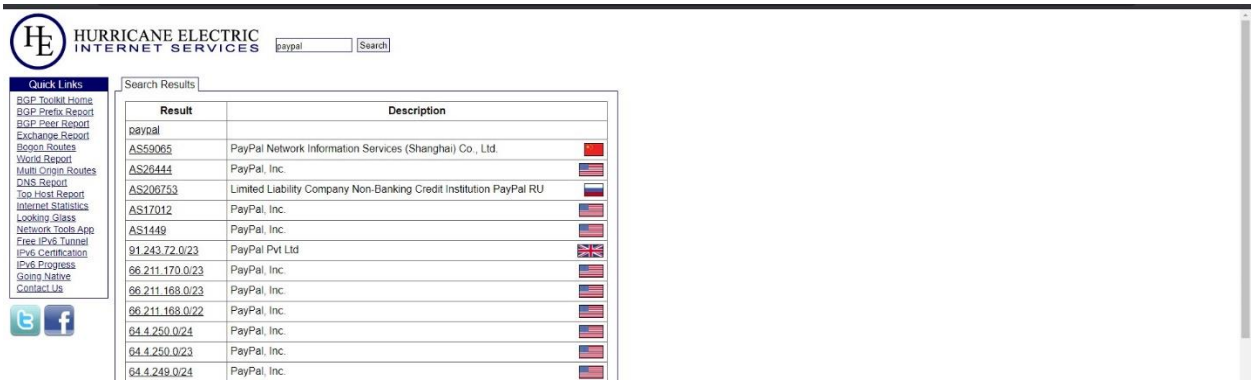
4.8.3 Identifying Important Sub domains for the Audit

Auditing all these live sub domains require so much time and processing. There fore from the list with live sub domains, here are the list of sub domains that will be focused on this audit that have been identified as most crucial sub domains in PayPal organization.

- https://active-www.paypal.com
- https://business.paypal.com
- https://business.sandbox.paypal.com
- https://api-3t.paypal.com
- https://checkout.paypal.com
- https://demo.paypal.com
- https://developer.paypal.com
- https://financing.paypal.com
- https://safetyhub.paypal.com
- https://api-aa.paypal.com
- https://api.financing.paypal.com
- http://announcements.paypal.com
- https://sandbox.paypal.com
- http://registration.paypal.com
- https://shopping.paypal.com

4.9 ASN Enumeration

Autonomous Systems are important given any network in an organization, specially a broader one. Finding ASN will help us identify netblocks of the domains.



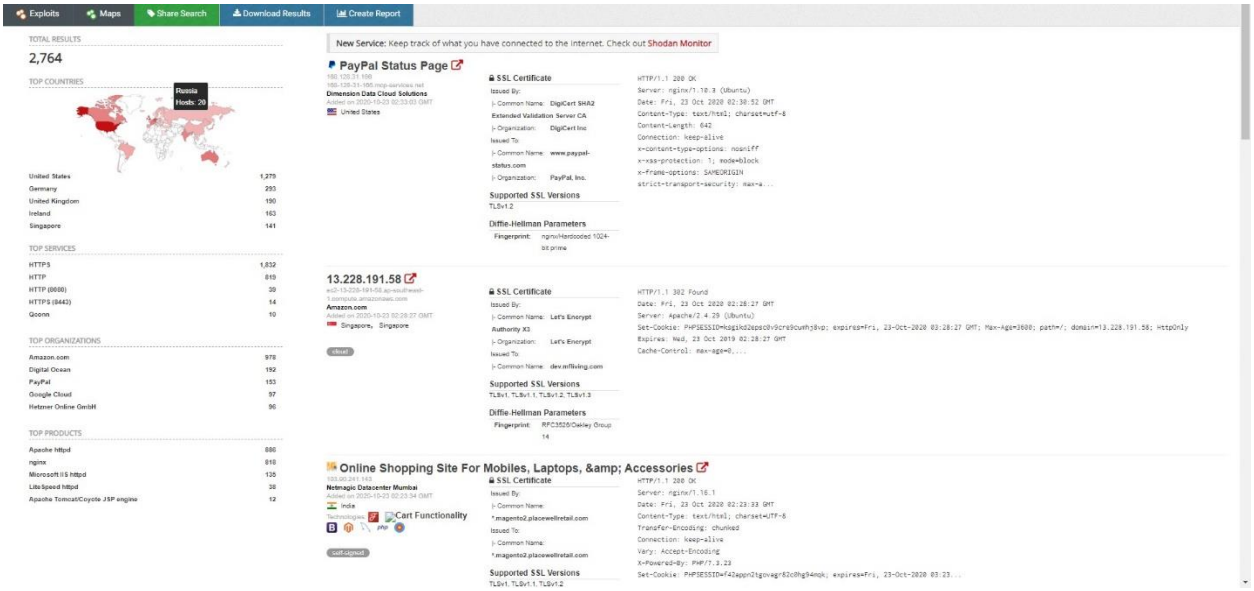
We can also conduct these Sub domain Enumeration and ASN Enumeration through several other means, yet I selected these particulars to demonstrate this. My initial choice was Amass since it had a great deal of combinations. But it seemed that it was not running efficiently in my execution environment.

4.10 Using Shodan on PayPal

Shodan is the search engine for everything on the internet. While Google and other search engines index only the web, Shodan indexes pretty much everything else — web cams, water treatment facilities, yachts, medical devices, traffic lights, wind turbines, license plate readers, smart TVs, refrigerators, anything and everything you could possibly imagine that's plugged into the internet.

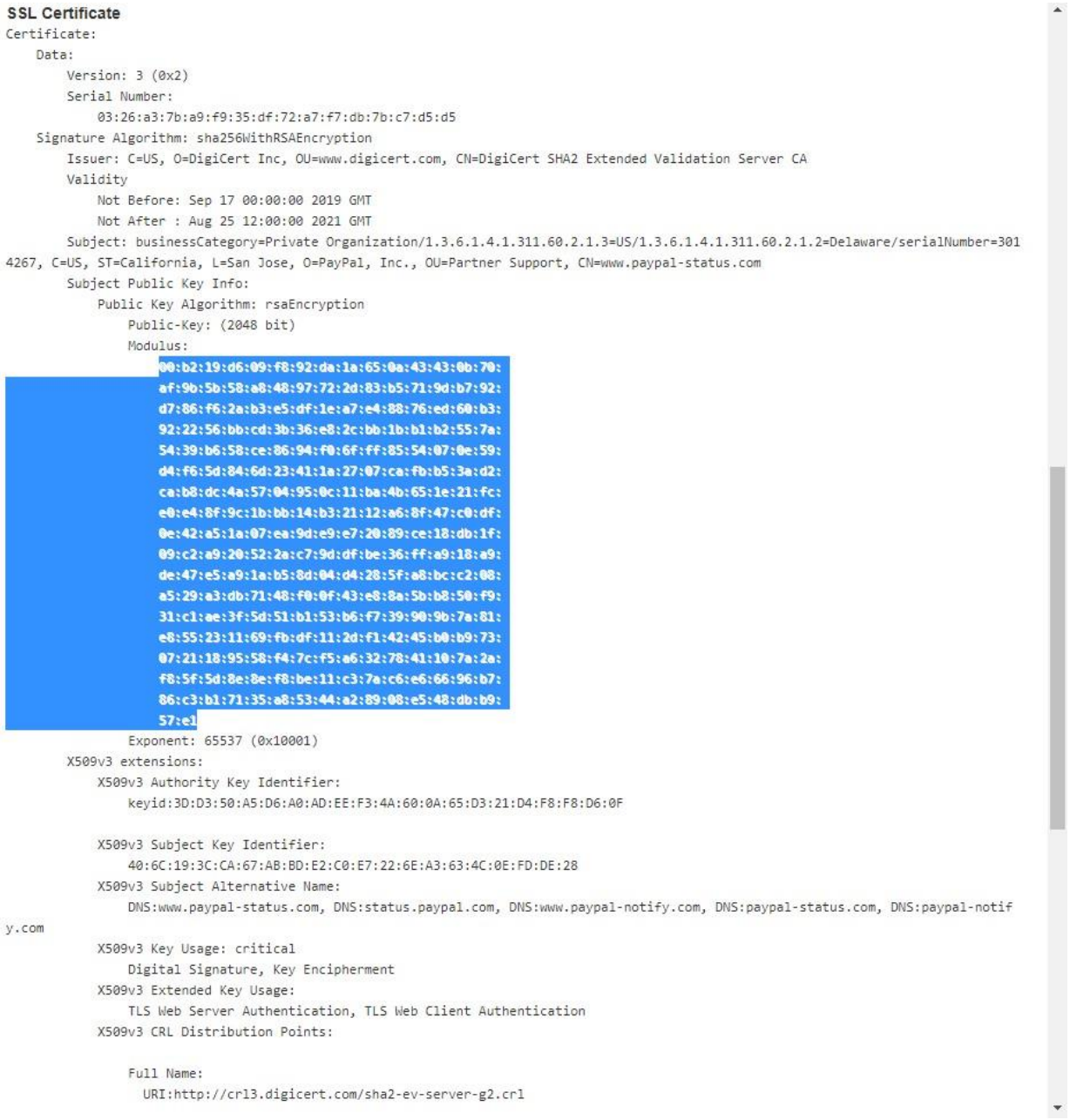
The information gained from these services is applied to many areas:

- **Network Security**: keep an eye on all devices at your company that are facing the Internet
- **Market Research**: find out which products people are using in the real-world
- **Cyber Risk**: include the online exposure of your vendors as a risk metric
- **Internet of Things**: track the growing usage of smart devices
- **Tracking Ransomware**: measure how many devices have been impacted by ransomware



To get the most out of Shodan it's important to understand the search query syntax. Along with these features it offers much more reliable ways of gathering information by sorting them with exploits, maps and we can also download our results. But the main odd is that to access all the features a premium account should be subscribed .

I was able to get the details of the SSL Certificate through navigation and exploring many IP addresses but the true essence of Shodan comes to premium accounts.



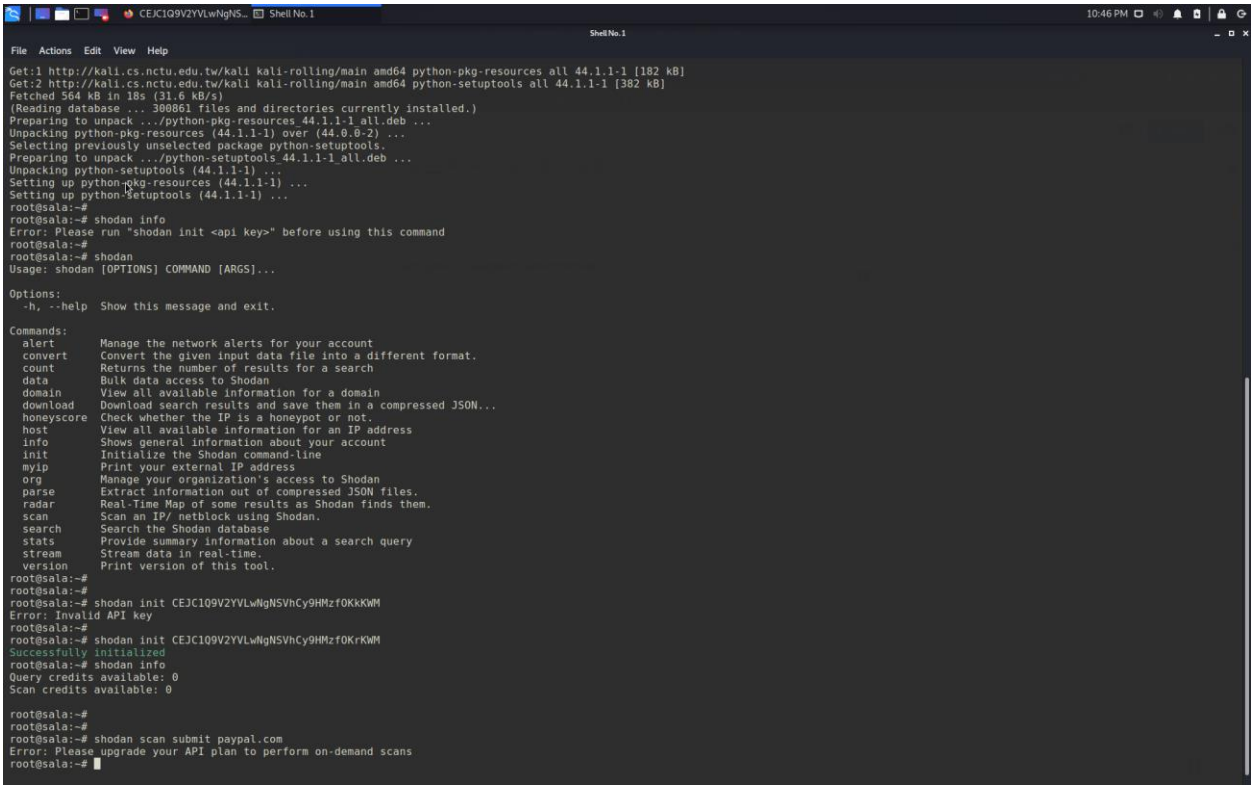
SSL Certificate

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            03:26:a3:7b:a9:f9:35:df:72:a7:f7:db:7b:c7:d5:d5
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 Extended Validation Server CA
        Validity
            Not Before: Sep 17 00:00:00 2019 GMT
            Not After : Aug 25 12:00:00 2021 GMT
        Subject: businessCategory=Private Organization/1.3.6.1.4.1.311.60.2.1.3=US/1.3.6.1.4.1.311.60.2.1.2=Delaware/serialNumber=301
4267, C=US, ST=California, L=San Jose, O=PayPal, Inc., OU=Partner Support, CN=www.paypal-status.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:b2:19:d6:09:f8:92:da:1a:65:0a:43:43:0b:70:
                    af:9b:5b:58:a8:48:97:72:2d:83:b5:71:9d:b7:92:
                    d7:86:f6:2a:b3:e5:df:1e:a7:e4:88:76:ed:60:b3:
                    92:22:56:bb:cd:3b:36:e8:2c:bb:1b:b1:b2:55:7a:
                    54:39:b6:58:ce:86:94:f0:6f:ff:85:54:07:0e:59:
                    d4:f6:5d:84:6d:23:41:1a:27:07:ca:fb:b5:3a:d2:
                    ca:b8:dc:4a:57:04:95:0c:11:ba:4b:65:1e:21:fc:
                    e0:e4:8f:9c:1b:bb:14:b3:21:12:a6:8f:47:c0:df:
                    0e:42:a5:1a:07:ea:9d:e9:e7:20:89:ce:18:db:1f:
                    09:c2:a9:20:52:2a:c7:9d:df:be:36:ff:a9:18:a9:
                    de:47:e5:a9:1a:b5:8d:04:d4:28:5f:a8:bc:c2:08:
                    a5:29:a3:db:71:48:f0:0f:43:e8:8a:5b:b8:50:f9:
                    31:c1:ae:3f:5d:51:b1:53:b6:f7:39:90:9b:7a:81:
                    e8:55:23:11:69:fb:df:11:2d:f1:42:45:b0:b9:73:
                    07:21:18:95:58:f4:7c:f5:a6:32:78:41:10:7a:2a:
                    f8:5f:5d:8e:8e:f8:be:11:c3:7a:c6:e6:66:96:b7:
                    86:c3:b1:71:35:a8:53:44:a2:89:08:e5:48:db:b9:
                    57:e1
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Authority Key Identifier:
                keyid:3D:D3:50:A5:D6:A0:AD:EE:F3:4A:60:0A:65:D3:21:D4:F8:F8:D6:0F

            X509v3 Subject Key Identifier:
                40:6C:19:3C:CA:67:AB:BD:E2:C0:E7:22:6E:A3:63:4C:0E:FD:DE:28
            X509v3 Subject Alternative Name:
                DNS:www.paypal-status.com, DNS:status.paypal.com, DNS:www.paypal-notify.com, DNS:paypal-status.com, DNS:paypal-notif
y.com
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 CRL Distribution Points:

                Full Name:
                  URI:http://crl3.digicert.com/sha2-ev-server-g2.crl
```

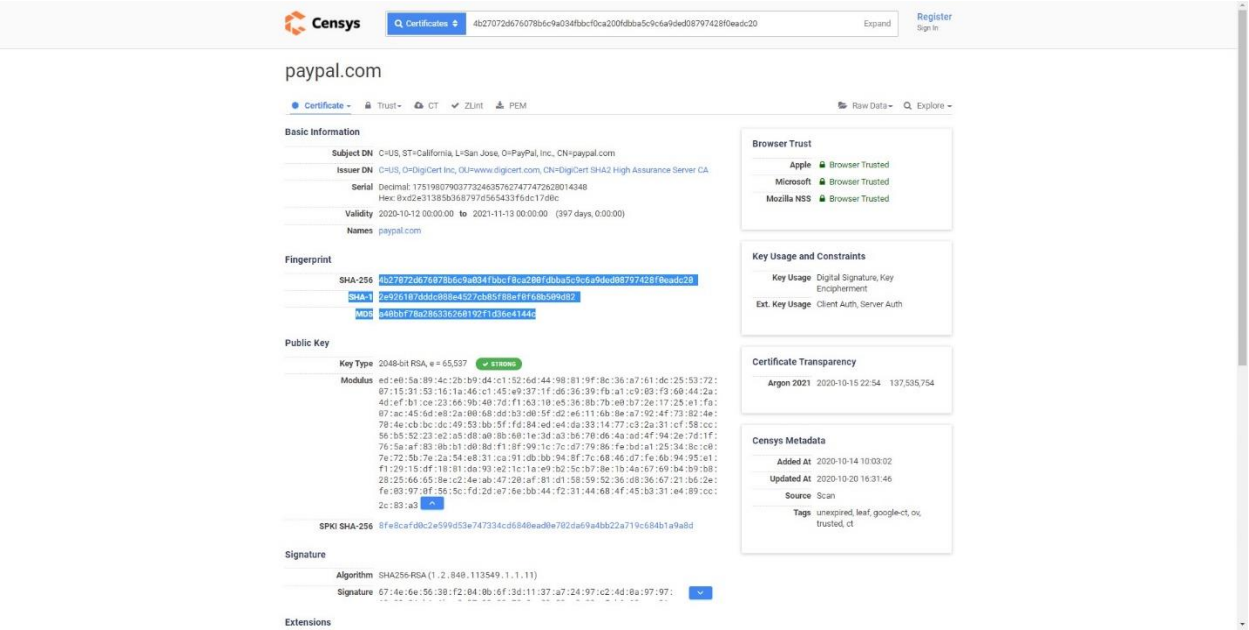
Working with the Shodan API to retrieve information is also fascinating, yet it also requires a premium subscription. The below figure shows how it works in the Linux terminal.

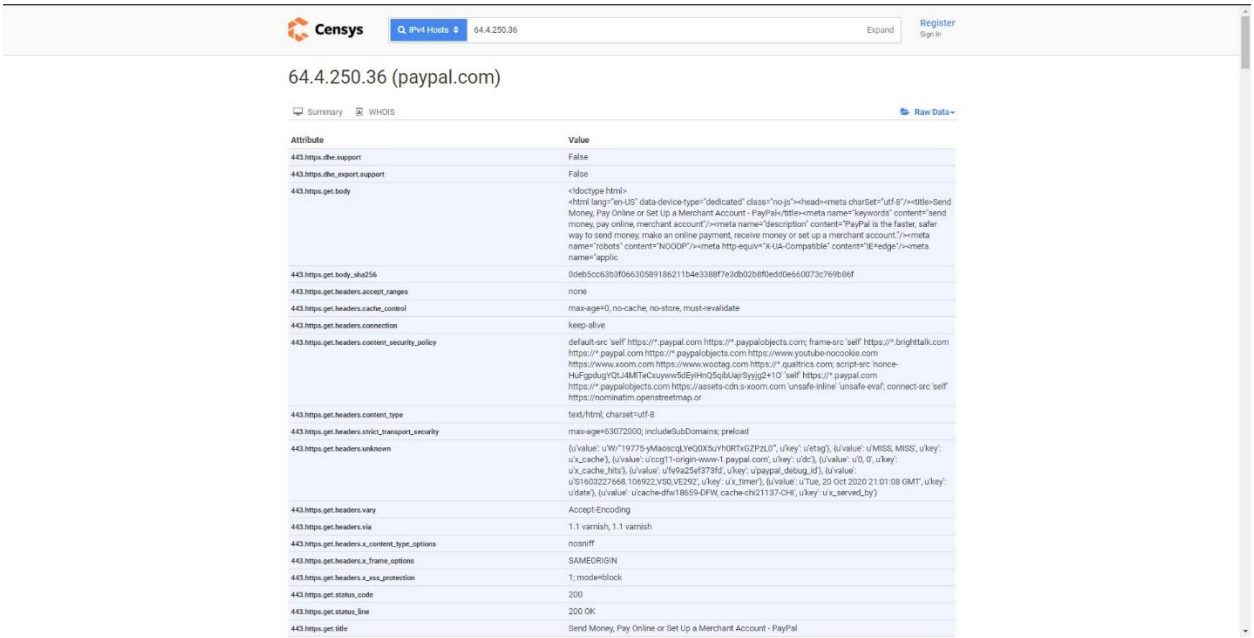


4.11 Using Censys on PayPal

Like the Shodan, censys.io is also a new Search Engine for devices exposed on the Internet, it could be used by experts to assess the security they implement. It maintains a complete database of every device exposed on the Internet. It represents a privileged instrument for the hackers that must search for a specific target and need to gather information on its configuration. At the same time, security experts could easily locate poorly protected devices exposed over the internet.

The following results were obtaitned through the censys.io requesting its services on the IP address “64.4.250.36”. Even though we were able to scratch many details regarding to the SSL and the Fingerprint from many other instances censys gives them in an orderly manner for a user to go through and undertstand clearly.





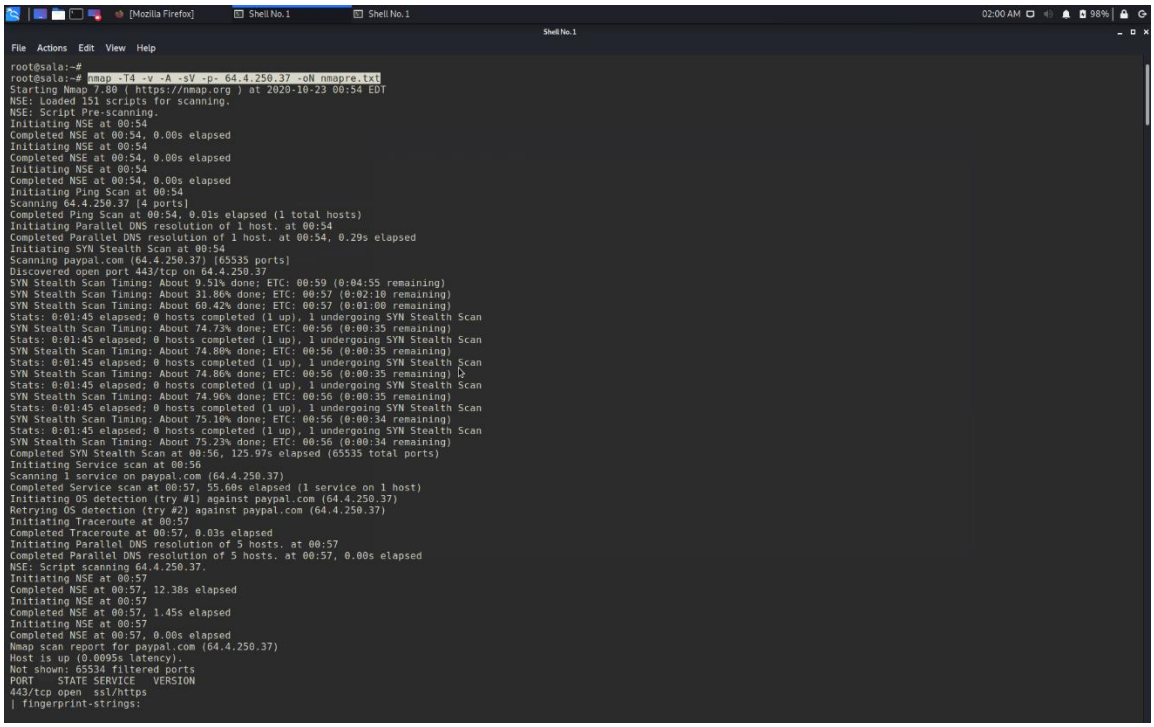
This is a full details summary of all the requests related to port 443 or the HTTPS related to the given IP adddress.

5. Scanning with Nmap

Nmap, short for Network Mapper, is a free, open-source tool for vulnerability scanning and network discovery. We can use Nmap to identify what devices are running on their systems, discovering hosts that are available and the services they offer, finding open ports and detecting security risks. Nmap can be used to monitor single hosts as well as vast networks that encompass hundreds of thousands of devices and multitudes of subnets.

5.1 Nmap on PayPal (64.4.250.37)

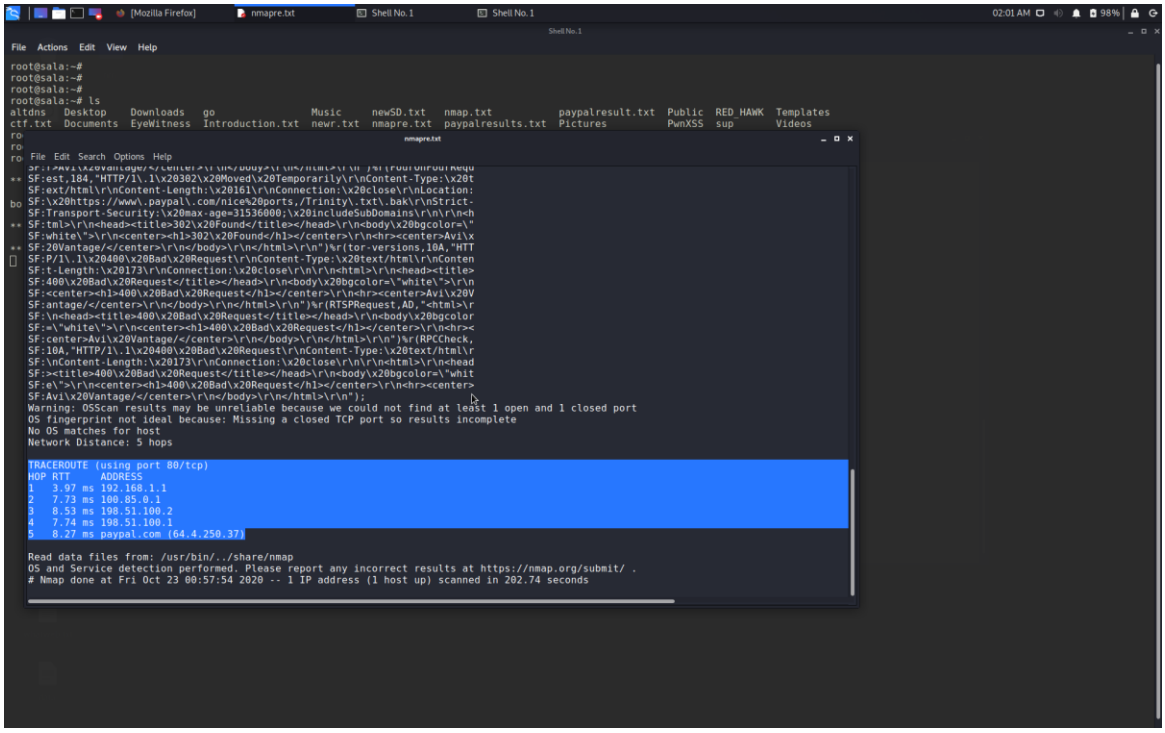
Nmap has a manual page inside the Linux kernel, so that we can study it before using the tool. It offers us numerous chances and combinations to use and get the information according to our commands.



Command – nmap -T4 -v -A -sV -p- 64.4.250.37 -oN nmapre.txt

- -T4 -> engages in speed scans pertaining to time and performance

- -sV -> Attempts to determine the version of the service running on port
- -p- -> Port scan all ports
- -A -> Enables OS detection, version detection, script scanning, and tracerout.
- -oN -> output the result to a file



Points to consider :

- Traceroute
- Open ports
- Services

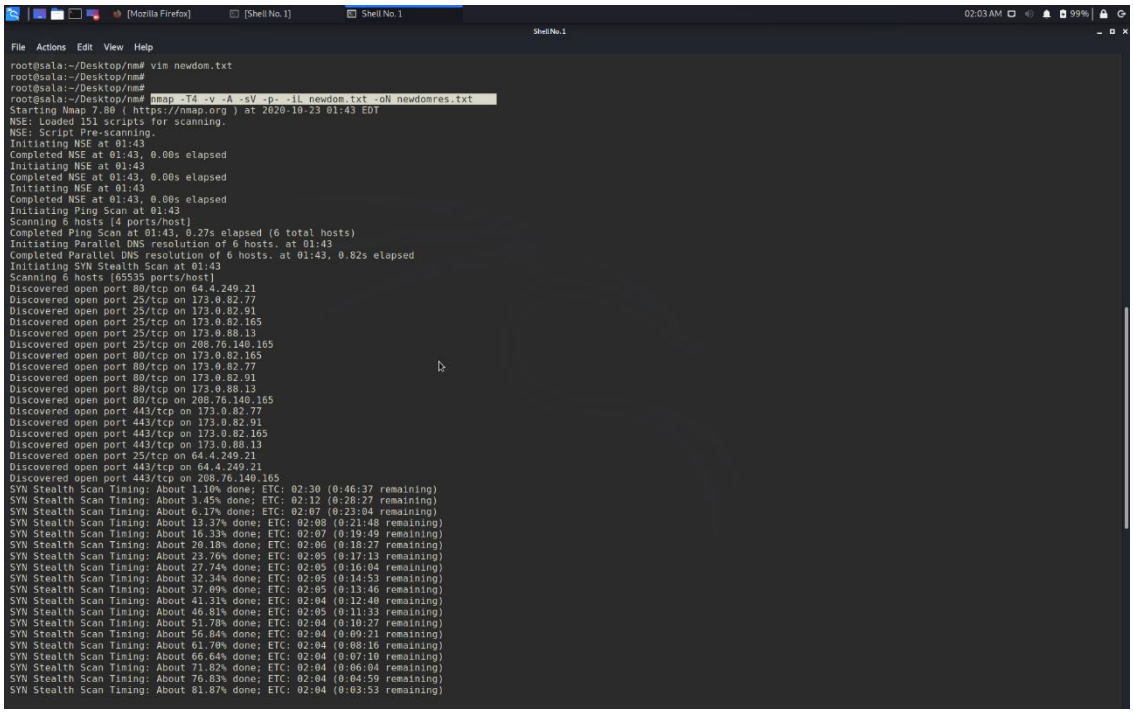
5.2 Automating the task combining more subdomains using a text file

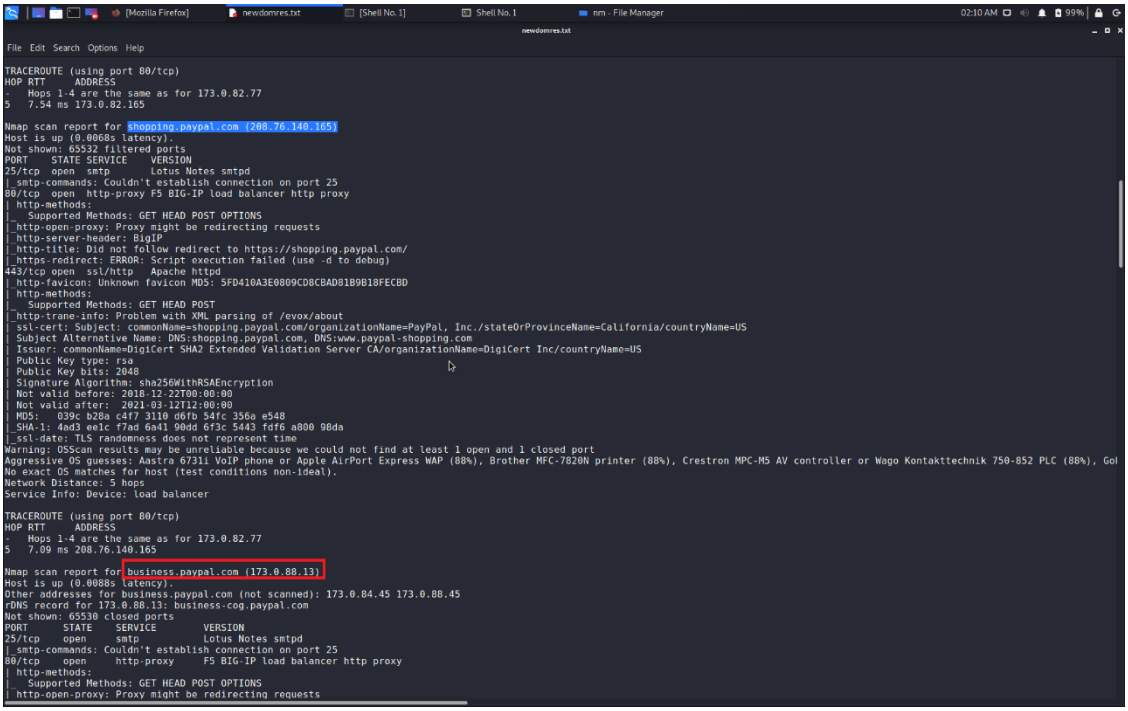
Here, a few live subdomains are stored to a file and using Nmap all those subdomains will be filtered and scanned setting parameters to display just like the first demonstration.

List of sub domains :

- sandbox.paypal.com
- registration.paypal.com
- shopping.paypal.com

- business.paypal.com
- business.sandbox.paypal.com
- api-3t.paypal.com

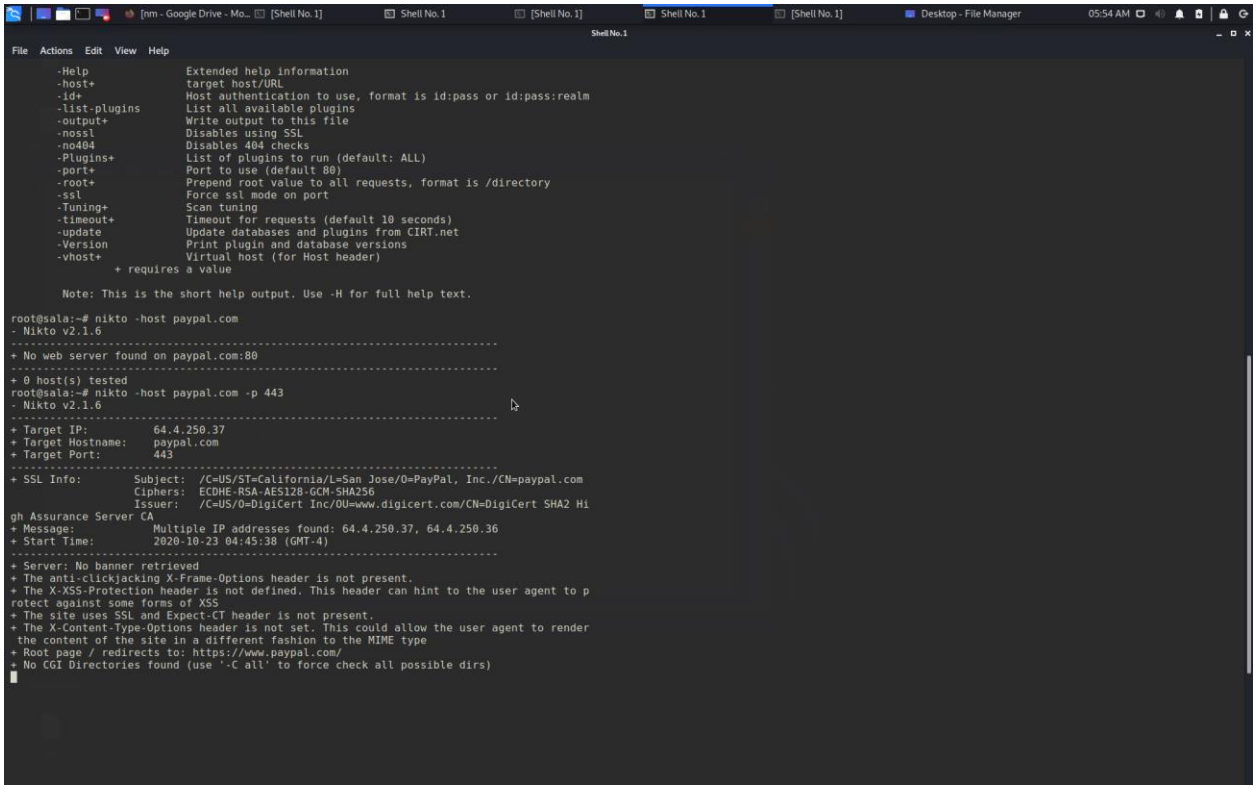




6. Nikto

Nikto is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items, including over 6700 potentially dangerous files/programs, checks for outdated versions of over 1250 servers, and version specific problems on over 270 servers. It also checks for server configuration items such as the presence of multiple index files, HTTP server options, and will attempt to identify installed web servers and software. Scan items and plugins are frequently updated and can be automatically updated.

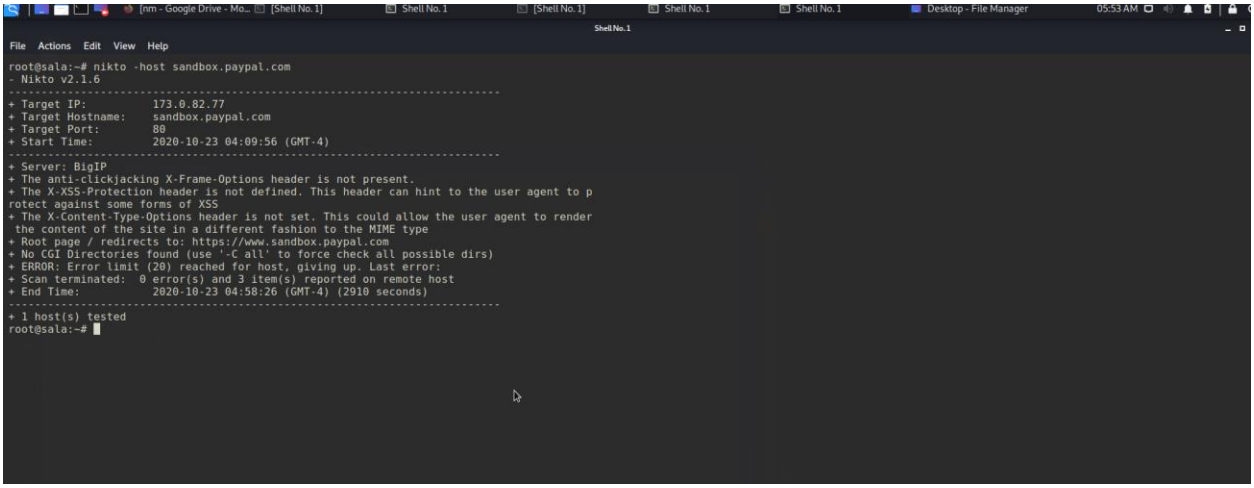
*Note – PayPal has secured most of their databases and servers with high end security solutions, so that Nikto did not work **Successfully** on most of the sub domains even though it was tried many times over a period of 2 days. Screenshots uploaded here are taken at the last attempt.

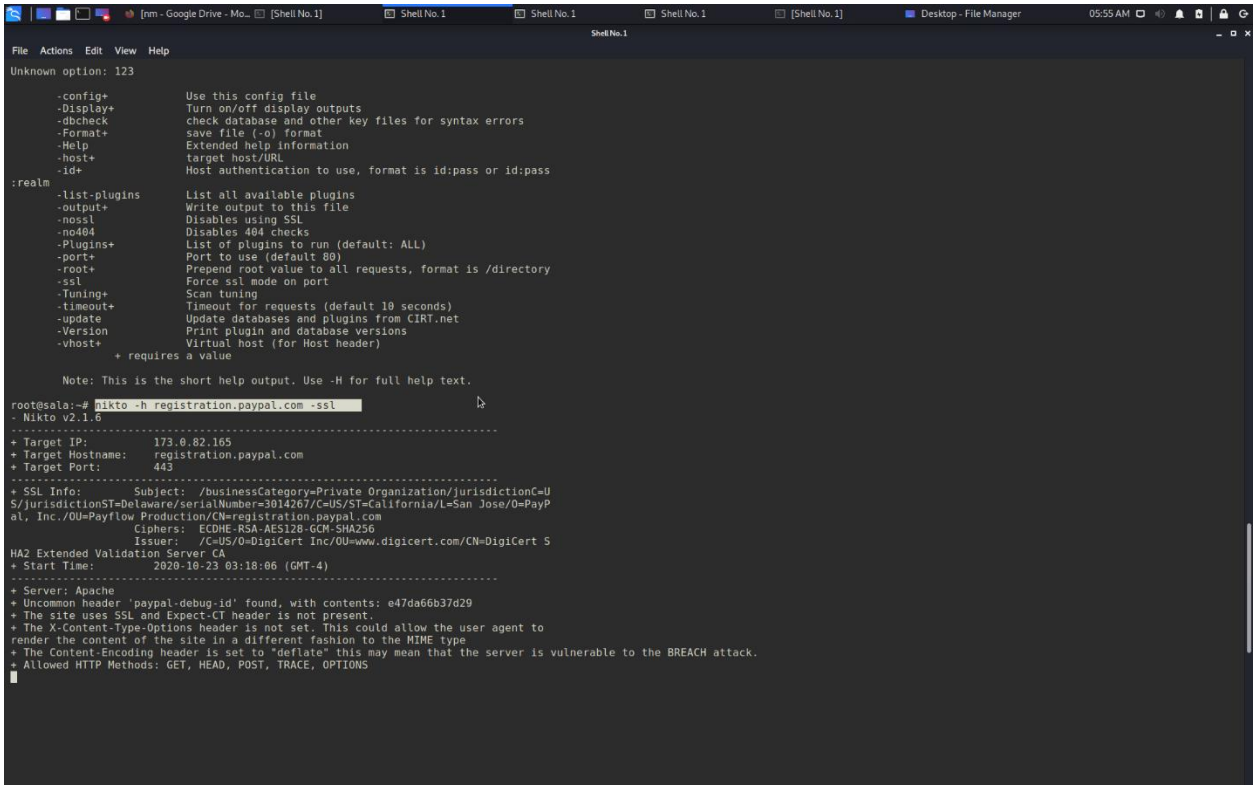


This indicates,

- XSS- protection header issue that it has not been defined.

- Indicates the SSL usage over their traffic transportation.





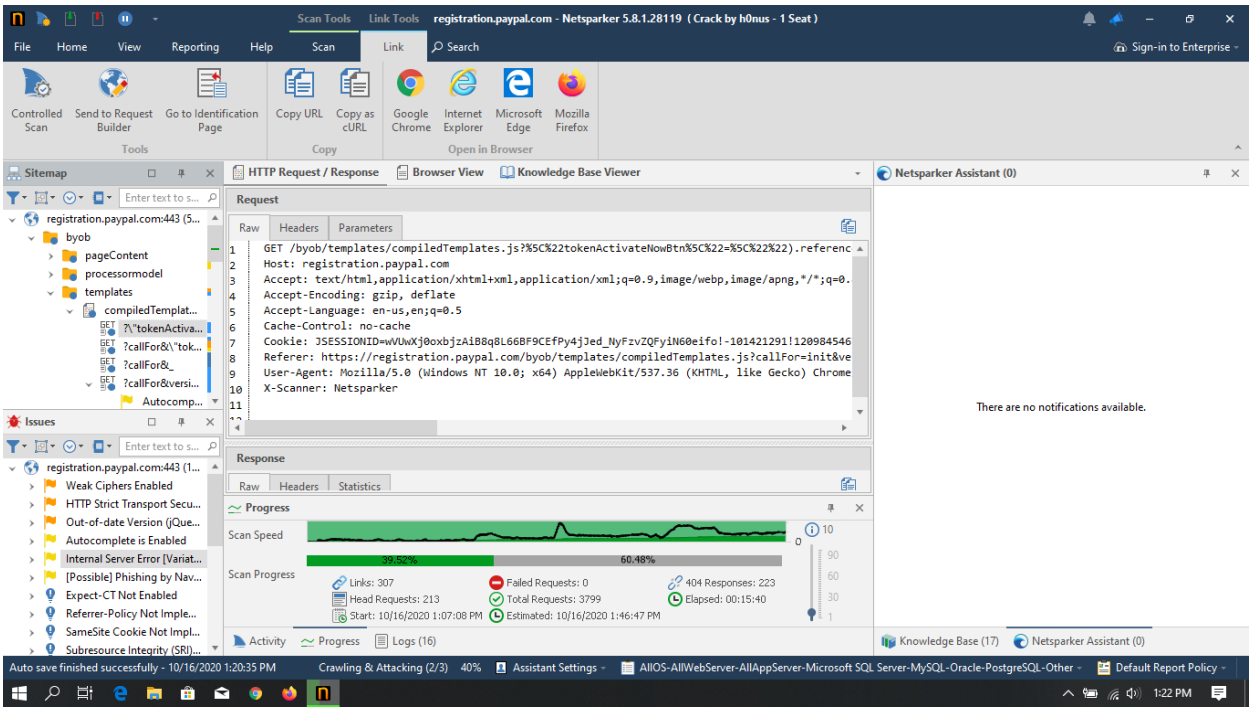
7. Scan PayPal using Netsparker

Netsparker is an automated, yet fully configurable, web application security scanner that enables you to scan websites, web applications and web services, and identify security flaws. Netsparker can scan all types of web applications, regardless of the platform or the language with which they are built.

Netsparker is the only online web application security scanner that automatically exploits identified vulnerabilities in a read-only and safe way, to confirm identified issues. It also presents proof of the vulnerability so that you do not need to waste time manually verifying it. For example, in the case of a detected SQL injection vulnerability, it will show the database name as the proof of exploit.

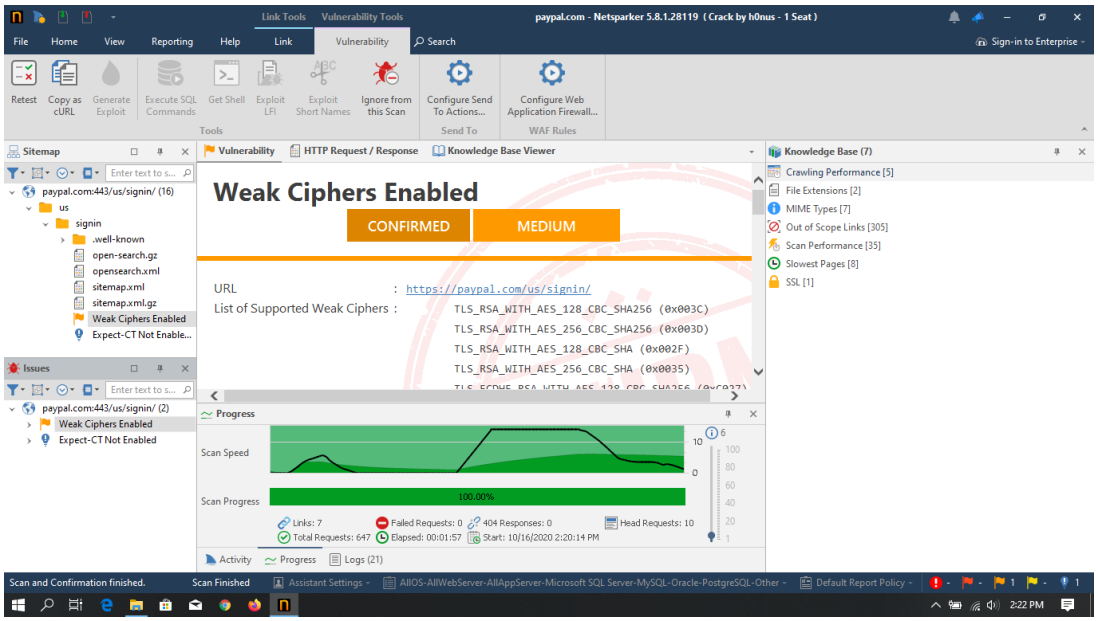
Assessments for the results will be demonstrated later.

7.1 registration.paypal.com

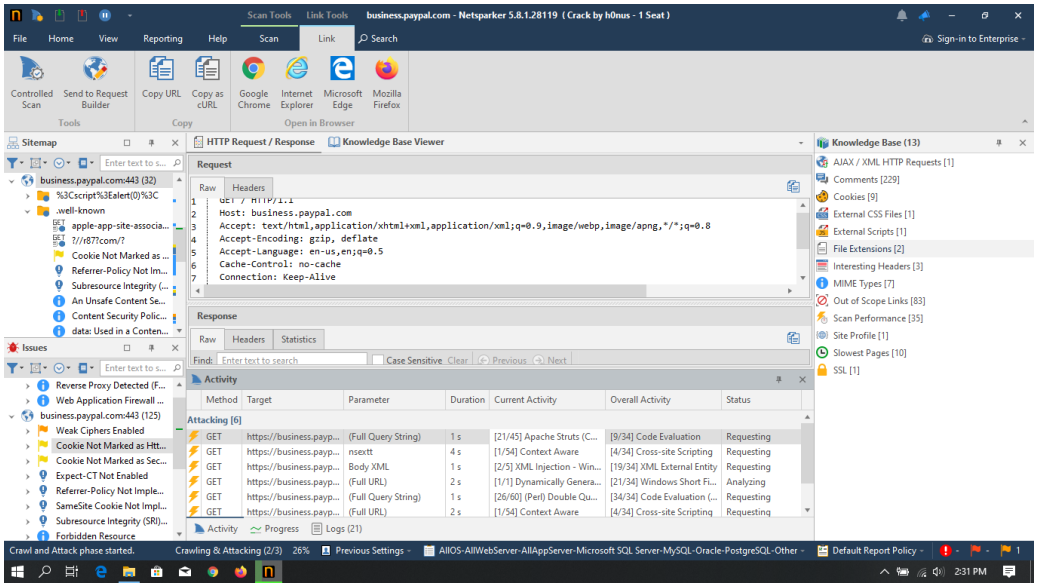


Netsparker was used separately for full domains to scan and enable vulnerability reports but most of them did not get through since the firewalls are enabled even though the form authentication feature was enabled at the time of scanning.

7.2 paypal.com



7.3 business.paypal.com

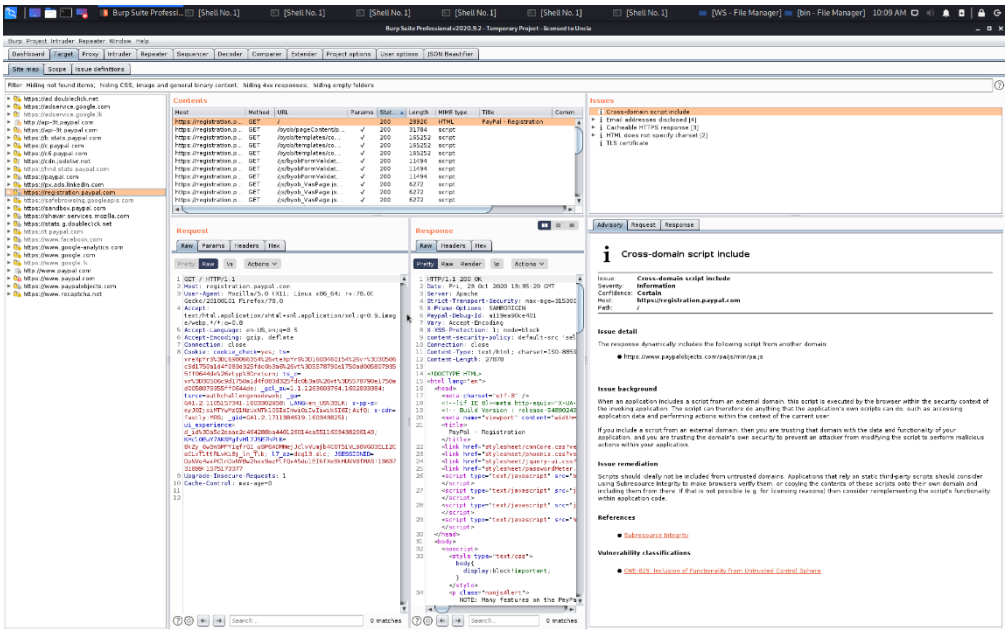


8. See through PayPal using BurpSuite

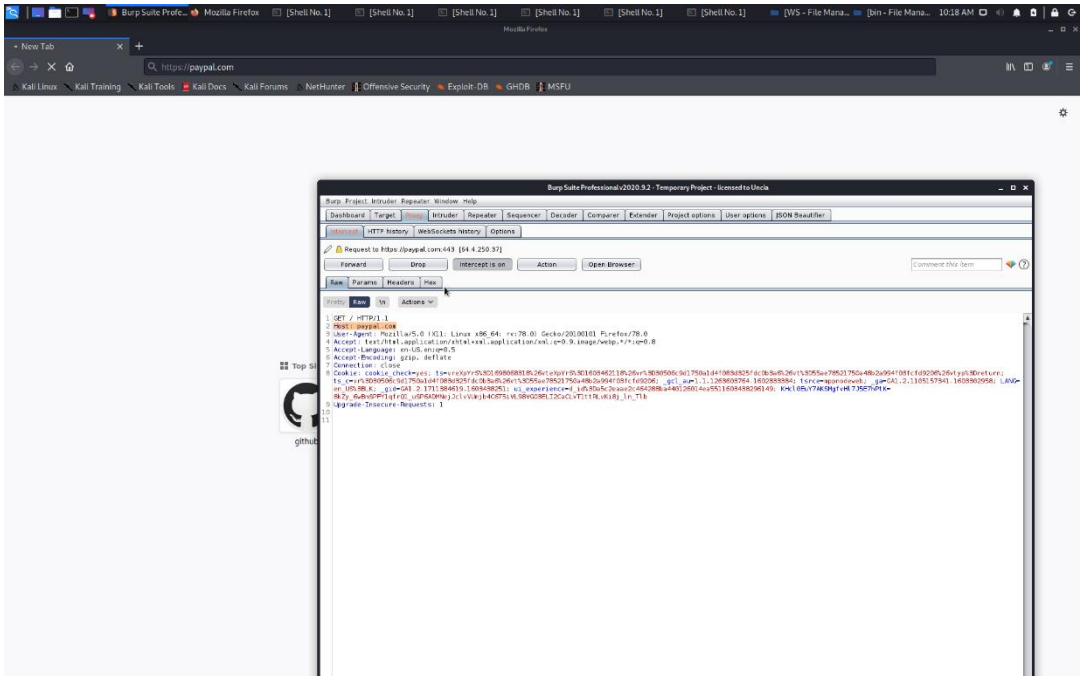
Burp Suite is a Java based Web Penetration Testing framework. It has become an industry standard suite of tools used by information security professionals. Burp Suite helps you identify vulnerabilities and verify attack vectors that are affecting web applications. Because of its popularity and breadth as well as depth of features, we have created this useful page as a collection of Burp Suite knowledge and information.

Burp Suite can be classified as an Interception Proxy. While browsing their target application, a penetration tester can configure their internet browser to route traffic through the Burp Suite proxy server. Burp Suite then acts as a (sort of) Man In The Middle by capturing and analyzing each request to and from the target web application so that they can be analyzed. Penetration testers can pause, manipulate, and replay individual HTTP requests in order to analyze potential parameters or injection points. Injection points can be specified for manual as well as automated fuzzing attacks to discover potentially unintended application behaviors, crashes, and error messages.

8.1 Accessing BurpSuite

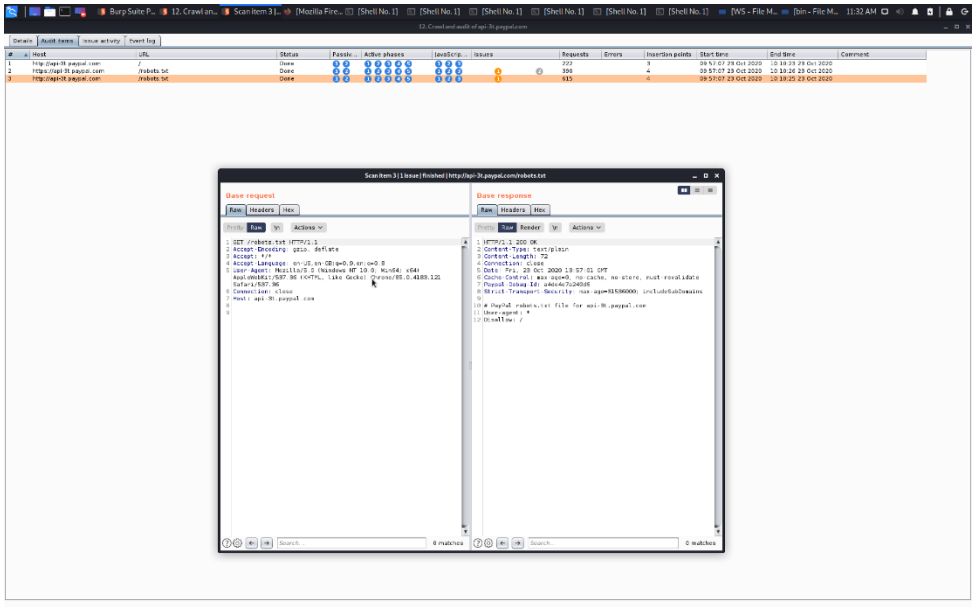


8.2 First request to PayPal intercepted by Burp

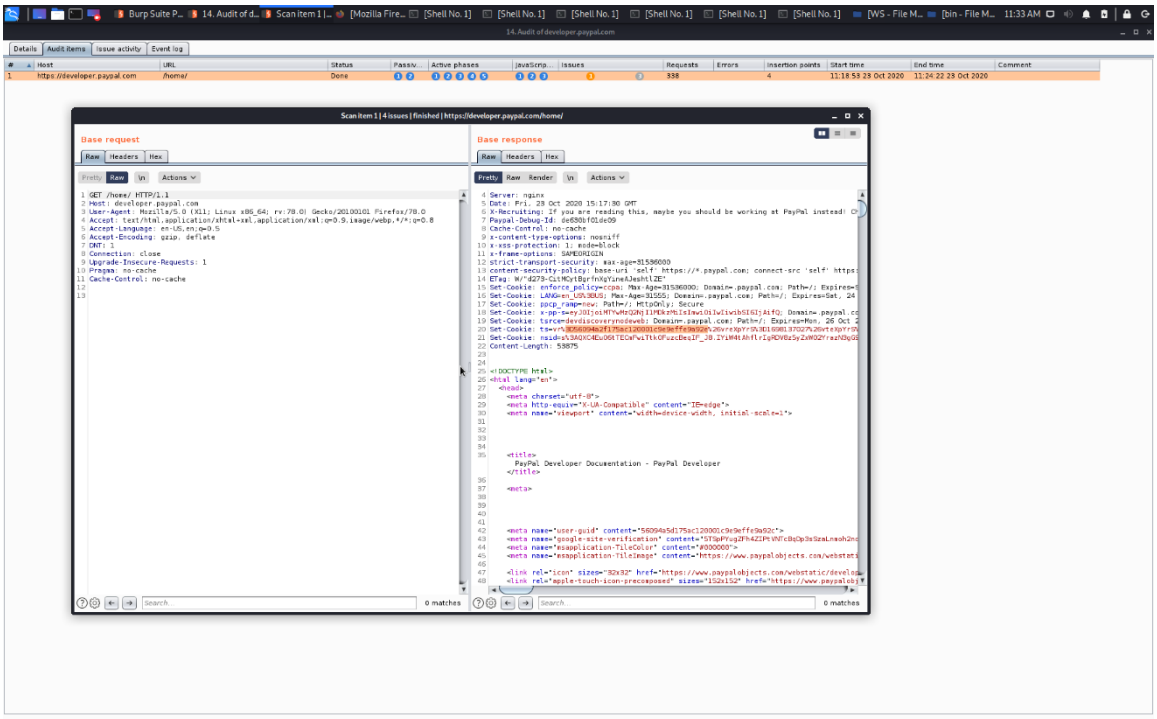


I have crawled and audited several sub domains of PayPal both using the BurpScanner and manipulating HTTP/HTTPS requests that are mentioned below as the overall results.

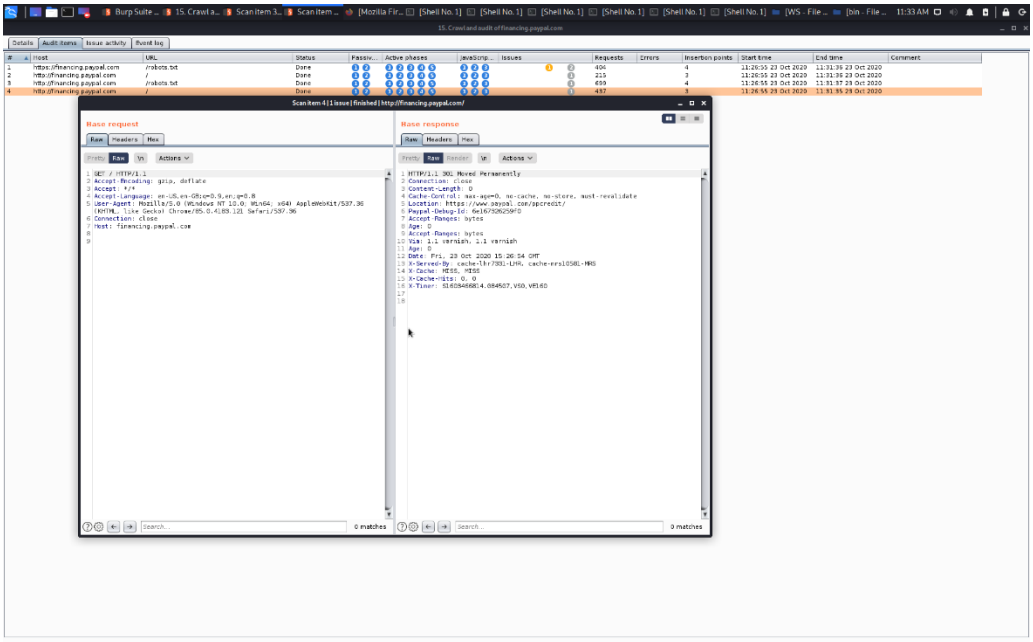
8.3 Api-3t.paypal.com



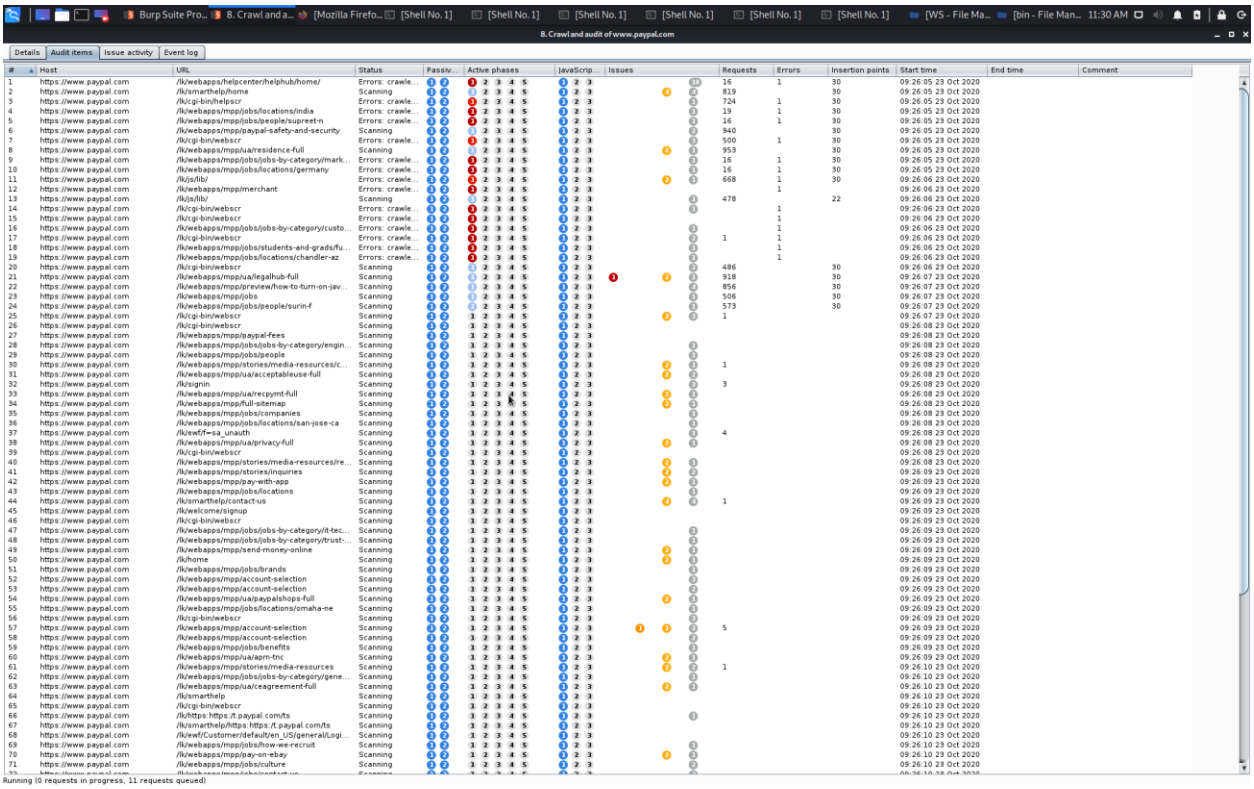
8.4 Developer.paypal.com



8.5 Financing.paypal.com



8.6 Paypal.com



Several other configurations, results and remedies are discussed in Assessment module

9. Assessment

All the identified threats, vulnerabilities and disclosures are studied, and solutions are provided and documented in this section.

9.1 Summary of https://paypal.com

9.1.1 Weak Ciphers Enabled

It was identified 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

https://paypal.com/us/signin

List of Supported Weak Ciphers

- TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
- TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003D)
- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)
  TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)
  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the httpd.conf.

2. Lighttpd:

   ssl.honor-cipher-order = "enable"

   ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"

3. For Microsoft IIS, you should make some changes to the system registry. Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.

Remedy

Configure your web server to disallow using weak ciphers.

9.1.2 LDAP Injection (High Severity)

Issue detail

The **l7_az** cookie appears to be vulnerable to LDAP injection attacks.

The payloads **88eb6x4ig6)(objectClass=*** and **flj0uobzsm)(!(objectClass=*)** were each submitted in the l7_az cookie. These two requests resulted in different responses, indicating that the input may be being incorporated into a disjunctive LDAP query in an unsafe manner.

Issue background

LDAP injection arises when user-controllable data is copied in an unsafe way into an LDAP query that is performed by the application. If an attacker can inject LDAP metacharacters into the query, then they can interfere with the query's logic. Depending on the function for which the query is used, the attacker may be able to retrieve sensitive data to which they are not authorized, or subvert the application's logic to perform some unauthorized action.

Note that automated difference-based tests for LDAP injection flaws can often be unreliable and are prone to false positive results. Scanner results should be manually reviewed to confirm whether a vulnerability is actually present.

Issue remediation

If possible, applications should avoid copying user-controllable data into LDAP queries. If this is unavoidable, then the data should be strictly validated to prevent LDAP injection attacks. In most situations, it will be appropriate to allow only short alphanumeric strings to be copied into queries, and any other input should be rejected. At a minimum, input containing any LDAP metacharacters should be rejected; characters that should be blocked include ( ) ; , * | & = and whitespace.

Vulnerability classifications

- CWE-90: Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection')
- CWE-116: Improper Encoding or Escaping of Output

Request 1

GET /lk/webapps/mpp/ua/legalhub-full HTTP/1.1
Host: www.paypal.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en-US,en-GB;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36
Connection: close
Cache-Control: max-age=0
Referer: https://www.paypal.com/lk/webapps/mpp/home
Cookie: nsid=s%3AM7jZ-Csogd32G9jxtFu5YUQfgX3a2HdP.cK9zZNqyAqMvwmHuG%2BjwNYsKz7FOL5%2Bu8Y7iulSZKGE; x-cdn=fastly:MRS; l7_az=88eb6x4ig6)(objectClass%3d*; ts_c=vr%3D5591fe541750ad0469b317a2fed0d7a6%26vt%3D5591fe541750ad0469b317a2fed0d7a5; cookie_check=yes; navcmd=_home-general; consumer_display=USER_HOMEPAGE%3d0%26USER_TARGETPAGE%3d0%26USER_FILTER_CHOICE%3d0%26BALANCE_MODULE_STATE%3d1%26GIFT_BALANCE_MODULE_STATE%3d1%26LAST_SELECTED_ALIAS_ID%3d0%26SELLING_GROUP%3d1%26PAYMENT_AND_RISK_GROUP%3d1%26SHIPPING_GROUP%3d1%26HOME_VERSION%3d1603544834%26MCE2_ELIGIBILITY%3d4294967295;

KHcl0EuY7AKSMgfvHl7J5E7hPtK=M58RMXRNe7K6KHPS2rNJutska7vY6Os0CBHOcZGtr
JllHfc61aSX9Wy6apcbf5X44V6ROurp7kpdYt1q; navlns=0.0;
cwrClyrK4LoCV1fydGbAxiNL6iG=isk5a36vLu23CeQrYmCa3avt--
r3eu6QyT1jxDsKihbtO5HMpTTi88z_7AxomkvyPX56ysmC9rozsLKzo5-
40ziYMZGOVwPx9sxvFT0nHKED7JcnerJiCjg79Gv0n7PV3uuaWHZAv0MC0WHCvjdCp2M
49VsV1RuoJJmktqggebpLNdFhWIlMRbMk0w0k0a32R22ug2wad_C6U4sk3bjyM1u9hTotd3v
yZRAUuyRddsZM-6AftayEYp2GGR8Y4Phc1AVuqnBytsfRFnMH3hwn1ST1QFm-
ntIh_iKyd24Y4y0s0kYfXN2l2Dfqev86dx28HBhlAX2otpMoav1PdvHRd5Mg97DkYDoXUuM
pCIR2B2s5UltVd-
UuGUoLmp3cE8X9tZss6a0ZRqG8vVK0NlIsL9HzpXgrUQgNAwMEnAarb7IyIg49dq7bfJHY
Vxu; LANG=en_US%3BLK; tsrce=mppnodeweb; x-pp-
s=eyJ0IjoiMTYwMzQ1ODQzNTkzNyIsImwiOiIwIiwibSI6IjAifQ;
ts=vreXpYrS%3D1698066435%26vteXpYrS%3D1603460235%26vr%3D5591fe541750ad0469
b317a2fed0d7a6%26vt%3D5591fe541750ad0469b317a2fed0d7a5%26vtyp%3Dnew

Response 1

HTTP/1.1 200 OK
Connection: close
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Content-Security-Policy: default-src 'self' https://*.paypal.com https://*.paypalobjects.com;
frame-src 'self' https://*.brighttalk.com https://*.paypal.com https://*.paypalobjects.com
https://www.youtube-nocookie.com https://www.xoom.com https://www.wootag.com
https://*.qualtrics.com; script-src 'nonce-
cizod4cbZrYdzv8RwpkjaoC+qMzcUT7NZhddK4ZRpSDPxgx7' 'self' https://*.paypal.com
https://*.paypalobjects.com https://assets-cdn.s-xoom.com 'unsafe-inline' 'unsafe-eval'; connect-
src 'self' https://nominatim.openstreetmap.org https://*.paypal.com https://*.paypalobjects.com
https://*.google-analytics.com https://*.salesforce.com https://*.force.com https://*.eloqua.com
https://nexus.ensighten.com https://api.paypal-retaillocator.com https://*.brighttalk.com
https://*.dialogtech.com https://*.qualtrics.com; style-src 'self' https://*.paypal.com 'self' https:
data:; form-action 'self' https://*.paypal.com https://*.salesforce.com https://*.eloqua.com
https://secure.opinionlab.com; base-uri 'self' https://*.paypal.com; object-src 'none'; frame-
ancestors 'self' https://*.paypal.com; block-all-mixed-content;; report-uri
https://www.paypal.com/csplog/api/log/csp
Content-Type: text/html; charset=utf-8
Etag: W/"5afd-pHler9jHhEbpJesOOMdeMyupwO8"
Paypal-Debug-Id: 9f8c5124dd432
Set-Cookie: LANG=en_US%3BLK; Max-Age=31556; Domain=.paypal.com; Path=/;
Expires=Fri, 23 Oct 2020 22:38:48 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: enforce_policy=; Domain=.paypal.com; Path=/; Expires=Thu, 01 Jan 1970 00:00:00
GMT; Secure; SameSite=None
Set-Cookie: x-pp-s=eyJ0IjoiMTYwMzQ2MTE3Mjk5MyIsImwiOiIwIiwibSI6IjAifQ;

Domain=.paypal.com; Path=/; HttpOnly; Secure; SameSite=None
Set-Cookie: l7_az=dcg12.slc; Path=/; Domain=paypal.com; Expires=Fri, 23 Oct 2020 14:22:53 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: ts=vreXpYrS%3D1698069172%26vteXpYrS%3D1603462972%26vr%3D5591fe541750ad0469b317a2fed0d7a6%26vt%3D55bbce9f1750a7805f988dcffed8b302%26vtyp%3Dreturn; Path=/; Domain=paypal.com; Expires=Mon, 23 Oct 2023 13:52:53 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: ts_c=vr%3D5591fe541750ad0469b317a2fed0d7a6%26vt%3D55bbce9f1750a7805f988dcffed8b302; Path=/; Domain=paypal.com; Expires=Mon, 23 Oct 2023 13:52:53 GMT; Secure; SameSite=None
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 1; mode=block
DC: ccg11-origin-www-1.paypal.com
Accept-Ranges: bytes
Via: 1.1 varnish
Accept-Ranges: none
Date: Fri, 23 Oct 2020 13:52:53 GMT
Via: 1.1 varnish
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
X-Served-By: cache-lhr7370-LHR, cache-mrs10563-MRS
X-Cache: MISS, MISS
X-Cache-Hits: 0, 0
X-Timer: S1603461173.811145,VS0,VE270
Vary: Accept-Encoding
Set-Cookie: x-cdn=fastly:MRS; Domain=paypal.com; Path=/; Secure
Content-Length: 23293

<!DOCTYPE html>
<html lang="en-LK" class="no-js" data-device-type="dedicated">
<head>
<title>PayPal</title>
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="robots" content="N

9.1.2 Session token in URL (Medium Severity)

Issue detail

The response contains the following links that appear to contain session tokens:

- https://www.paypal.com/auth/createchallenge/8aa9d0bede977ed1/recaptchav3.js?_sessionID=GdArxe45ajgMyRNj7WXetX6thlIO3sam

Issue background

Sensitive information within URLs may be logged in various locations, including the user's browser, the web server, and any forward or reverse proxy servers between the two endpoints. URLs may also be displayed on-screen, bookmarked or emailed around by users. They may be disclosed to third parties via the Referer header when any off-site links are followed. Placing session tokens into the URL increases the risk that they will be captured by an attacker.

Issue remediation

Applications should use an alternative mechanism for transmitting session tokens, such as HTTP cookies or hidden fields in forms that are submitted using the POST method.

Vulnerability classifications

- CWE-200: Information Exposure
- CWE-384: Session Fixation
- CWE-598: Information Exposure Through Query Strings in GET Request

Response

```
HTTP/1.1 200 OK
Connection: close
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Content-Security-Policy: default-src 'self' https://*.paypal.com https://*.paypalobjects.com; style-
```

src 'self' https://*.paypal.com https://*.paypalobjects.com 'unsafe-inline'; frame-src 'self' https://*.paypal.com https://*.paypalobjects.com https://*.cardinalcommerce.com https://*.qualtrics.com; script-src 'nonce-yUaQGVcvxdOYpQB/rEOrseKXsYBYW410pFOGjk/09jTtEHNr' 'self' https://*.paypal.com https://*.paypalobjects.com 'unsafe-inline' 'unsafe-eval'; connect-src 'self' https://*.paypal.com https://*.paypalobjects.com https://nexus.ensighten.com https://accounts.google.com https://*.qualtrics.com; img-src 'self' https: data:; object-src 'none'; font-src 'self' https://*.paypal.com https://*.paypalobjects.com; base-uri 'self' https://*.paypal.com; form-action 'self' https://*.paypal.com https://*.cardinalcommerce.com; block-all-mixed-content;; report-uri https://www.paypal.com/csplog/api/log/csp
Content-Type: text/html; charset=utf-8
Etag: W/"74fc-dPpY5aVfAdUlyMthzmyAiQUIg3E"
Paypal-Debug-Id: 4905524cf8fd7
Set-Cookie: enforce_policy=; Domain=.paypal.com; Path=/; Expires=Thu, 01 Jan 1970 00:00:00 GMT; Secure; SameSite=None
Set-Cookie: LANG=en_US%3BLK; Max-Age=31556; Domain=.paypal.com; Path=/; Expires=Fri, 23 Oct 2020 22:01:29 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: tsrce=progressivenodeweb; Max-Age=259199; Domain=.paypal.com; Path=/; Expires=Mon, 26 Oct 2020 13:15:32 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: x-pp-s=eyJ0IjoiMTYwMzQ1ODkzMzM2NCIsImwiOiIwIiwibSI6IjAifQ; Domain=.paypal.com; Path=/; HttpOnly; Secure; SameSite=None
Set-Cookie: nsid=s%3AGdArxe45ajgMyRNj7WXetX6thlIO3sam.HnmBytn7Z3a3tI7ci6q59CnMzOcw%2Bs5Mc%2BM9TJeXl9k; Path=/; HttpOnly; Secure; SameSite=None
Set-Cookie: l7_az=dcg14.slc; Path=/; Domain=paypal.com; Expires=Fri, 23 Oct 2020 13:45:33 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: ts=vreXpYrS%3D1698066933%26vteXpYrS%3D1603460733%26vr%3D559986381750a49186041b7efae31c5e%26vt%3D559986381750a49186041b7efae31c5d%26vtyp%3Dnew; Path=/; Domain=paypal.com; Expires=Mon, 23 Oct 2023 13:15:33 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: ts_c=vr%3D559986381750a49186041b7efae31c5e%26vt%3D559986381750a49186041b7efae31c5d; Path=/; Domain=paypal.com; Expires=Mon, 23 Oct 2023 13:15:33 GMT; Secure; SameSite=None
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 1; mode=block
DC: ccg11-origin-www-1.paypal.com
Accept-Ranges: bytes
Via: 1.1 varnish
Accept-Ranges: none
Date: Fri, 23 Oct 2020 13:15:33 GMT
Via: 1.1 varnish

Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
X-Served-By: cache-lhr7361-LHR, cache-mrs10571-MRS
X-Cache: MISS, MISS
X-Cache-Hits: 0, 0
X-Timer: S1603458933.103503,VS0,VE342
Vary: Accept-Encoding
Set-Cookie: x-cdn=fastly:MRS; Domain=paypal.com; Path=/; Secure
Content-Length: 30164

<!doctype html>
<html lang="en-LK" data-device-type="dedicated" class="no-js"><head><script src="https://www.paypalobjects.com/webcaptcha/ngrlCaptcha.min.js"></script><meta charSet="utf-8"/><title>Sig
...[SNIP]...
<script async defer src="/auth/createchallenge/8aa9d0bede977ed1/recaptchav3.js?_sessionID=GdArxe45ajgMyRNj7WXetX6thlIO3sam"></script>
...[SNIP]...

9.1.3 Password field with autocomplete enabled (Low Severity)

Issue detail

The page contains a form with the following action URL:

- https://www.paypal.com/lk/welcome/signup

The form contains the following password fields with autocomplete enabled:

- /paypalAccountData/password
- /paypalAccountData/confirmPassword

Issue background

Most browsers have a facility to remember user credentials that are entered into HTML forms. This function can be configured by the user and also by applications that employ user credentials.

If the function is enabled, then credentials entered by the user are stored on their local computer and retrieved by the browser on future visits to the same application.

The stored credentials can be captured by an attacker who gains control over the user's computer. Further, an attacker who finds a separate application vulnerability such as cross-site scripting may be able to exploit this to retrieve a user's browser-stored credentials.

Issue remediation

To prevent browsers from storing credentials entered into HTML forms, include the attribute **autocomplete="off"** within the FORM tag (to protect all form fields) or within the relevant INPUT tags (to protect specific individual fields).

Please note that modern web browsers may ignore this directive. In spite of this there is a chance that not disabling autocomplete may cause problems obtaining PCI compliance.

Vulnerability classifications

- CWE-200: Information Exposure

Response

HTTP/1.1 200 OK
Connection: close
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Content-Security-Policy: default-src 'self' https://*.paypal.com https://*.paypalobjects.com; style-src 'self' https://*.paypal.com https://*.paypalobjects.com 'unsafe-inline'; frame-src 'self' https://*.paypal.com https://*.paypalobjects.com https://*.cardinalcommerce.com https://*.qualtrics.com; script-src 'nonce-NlBXuytJPPmNgG1YiktUlbc0zUjuecssvocFjF/CPK9zeNzV' 'self' https://*.paypal.com https://*.paypalobjects.com 'unsafe-inline' 'unsafe-eval'; connect-src 'self' https://*.paypal.com https://*.paypalobjects.com https://nexus.ensighten.com https://accounts.google.com https://*.qualtrics.com; img-src 'self' https: data:; object-src 'none'; font-src 'self' https://*.paypal.com https://*.paypalobjects.com; base-uri 'self' https://*.paypal.com; form-action 'self' https://*.paypal.com https://*.cardinalcommerce.com; block-all-mixed-content;; report-uri https://www.paypal.com/csplog/api/log/csp

Content-Type: text/html; charset=utf-8
Etag: W/"af56-oLS/17BLDxdrMdsHrpB/YAoJTnc"
Paypal-Debug-Id: 3dddcd2acb2c9
Set-Cookie: enforce_policy=; Domain=.paypal.com; Path=/; Expires=Thu, 01 Jan 1970 00:00:00 GMT; Secure; SameSite=None
Set-Cookie: LANG=en_US%3BLK; Max-Age=31556; Domain=.paypal.com; Path=/; Expires=Fri, 23 Oct 2020 21:51:40 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: tsrce=progressivenodeweb; Max-Age=259199; Domain=.paypal.com; Path=/; Expires=Mon, 26 Oct 2020 13:05:43 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: cookie_prefs=; Path=/; Expires=Thu, 01 Jan 1970 00:00:00 GMT; Secure; SameSite=None
Set-Cookie: cookie_prefs=P%3D1%2CF%3D1%2Ctype%3Dimplicit; Max-Age=31536000; Domain=.paypal.com; Path=/; Expires=Sat, 23 Oct 2021 13:05:44 GMT; Secure; SameSite=None
Set-Cookie: x-pp-s=eyJ0IjoiMTYwMzQ1ODM0NDE4OCIsImwiOiIwIiwibSI6IjAifQ; Domain=.paypal.com; Path=/; HttpOnly; Secure; SameSite=None
Set-Cookie: nsid=s%3AqviF2PibO-anJhUXIUhI6F4_ZYf6KqqB.xdwL2eNB3ppTgklp5KnRSjijjX%2BMX2OzEhBlhgI1%2Bbc; Path=/; HttpOnly; Secure; SameSite=None
Set-Cookie: l7_az=dcg13.slc; Path=/; Domain=paypal.com; Expires=Fri, 23 Oct 2020 13:35:44 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: ts=vreXpYrS%3D1698066344%26vteXpYrS%3D1603460144%26vr%3D55909e1c1750a760c6820b6cfff09dbb%26vt%3D55909e1c1750a760c6820b6cfff09dba%26vtyp%3Dnew; Path=/; Domain=paypal.com; Expires=Mon, 23 Oct 2023 13:05:44 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: ts_c=vr%3D55909e1c1750a760c6820b6cfff09dbb%26vt%3D55909e1c1750a760c6820b6cfff09dba; Path=/; Domain=paypal.com; Expires=Mon, 23 Oct 2023 13:05:44 GMT; Secure; SameSite=None
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 1; mode=block
DC: ccg11-origin-www-1.paypal.com
Accept-Ranges: none
Via: 1.1 varnish, 1.1 varnish
Date: Fri, 23 Oct 2020 13:05:44 GMT
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
X-Served-By: cache-lhr7360-LHR, cache-mrs10560-MRS
X-Cache: MISS, MISS
X-Cache-Hits: 0, 0
X-Timer: S1603458344.926412,VS0,VE350
Vary: Accept-Encoding
Set-Cookie: x-cdn=fastly:MRS; Domain=paypal.com; Path=/; Secure

Content-Length: 45102

<!doctype html>
<html lang="en-LK" data-device-type="dedicated" class="no-js"><head><script src="https://www.paypalobjects.com/webcaptcha/ngrlCaptcha.min.js"></script><meta charSet="utf-8"/><title>Sig
...[SNIP]...
<form id="PageMainForm" class="signupAppContent" method="POST">
...[SNIP]...
<input type="password" aria-describedby="country_code_prefix paypalAccountData_password_helptext" aria-invalid="false" aria-autocomplete="none" style="padding-left:15px" value="" name="/paypalAccountData/password" id="paypalAccountData_password"/>
...[SNIP]...
<input type="password" aria-describedby="country_code_prefix paypalAccountData_confirmPassword_helptext" aria-invalid="false" aria-autocomplete="none" style="padding-left:15px" value="" name="/paypalAccountData/confirmPassword" id="paypalAccountData_confirmPassword"/>
...[SNIP]...

9.1.4 Link Manipulation (DOM- based) (Low Severity)

Issue detail

The application may be vulnerable to DOM-based link manipulation. Data is read from **location.href** and passed to **the 'action' property of a DOM element**.

Issue background

DOM-based vulnerabilities arise when a client-side script reads data from a controllable part of the DOM (for example, the URL) and processes this data in an unsafe way.

DOM-based link manipulation arises when a script writes controllable data to a navigation target within the current page, such as a clickable link or the submission URL of a form. An attacker may be able to use the vulnerability to construct a URL that, if visited by another application user, will modify the target of links within the response. An attacker may be able to leverage this to perform various attacks, including:

- Causing the user to redirect to an arbitrary external URL, to facilitate a phishing attack.
- Causing the user to submit sensitive form data to a server controlled by the attacker.
- Causing the user to perform an unintended action within the application, by changing the file or query string associated with a link.
- Bypassing browser anti-XSS defenses by injecting on-site links containing XSS exploits, since browser anti-XSS defenses typically do not operate on on-site links.

Burp Suite automatically identifies this issue using static code analysis, which may lead to false positives that are not actually exploitable. The relevant code and execution paths should be reviewed to determine whether this vulnerability is indeed present, or whether mitigations are in place that would prevent exploitation.

Issue remediation

The most effective way to avoid DOM-based link manipulation vulnerabilities is not to dynamically set the target URLs of links or forms using data that originated from any untrusted source. If the desired functionality of the application means that this behavior is unavoidable, then defenses must be implemented within the client-side code to prevent malicious data from introducing an arbitrary URL as a link target. In general, this is best achieved by using a whitelist of URLs that are permitted link targets, and strictly validating the target against this list before setting the link target.

Vulnerability classifications

- CWE-20: Improper Input Validation

Request

```
GET /digitalassets/c/website/marketing/global/kui/js/opinionLab-2.0.0.js HTTP/1.1
Host: www.paypalobjects.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
```

Accept-Encoding: gzip, deflate
Connection: close
Referer: https://www.paypal.com/lk/home

Response

HTTP/1.1 200 OK
Server: Apache
Last-Modified: Thu, 26 Jul 2018 16:45:50 GMT
Accept-Ranges: bytes
Content-Type: application/x-javascript
X-Pad: avoid browser bug
Content-Length: 42322
Cache-Control: max-age=3600
Expires: Fri, 23 Oct 2020 13:39:24 GMT
Date: Fri, 23 Oct 2020 12:39:24 GMT
Connection: close
Vary: Accept-Encoding
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

/* OnlineOpinion v5.9.3 Released: 09/21/2015. Compiled 09/30/2015 12:09:31 PM -0500
Branch: 5.9.3 efe6bf2541deb563c2a9884b2a3034c881047acf Components: Full UMD: disabled
The following code is Copyri
...[SNIP]...
tps://secure.opinionlab.com/ccc01/comment_card_json_4_0_b.asp?r='+location.href:'https://secure.opinionlab.com/ccc01/comment_card_d.asp';if(b.commentCardUrl){c.action=b.commentCardUrl;if(b.onPageCard){c.action+='?r='+location.href}}e.name='params';e.value=x(b);c.appendChild(e);d.body.appendChild(c);return c}function A(){return{width:screen.width,height:screen.height,referer:location.href,prev:document.referrer,time1:(new Date()
...[SNIP]...

9.1.5 Cookie without HttpOnly flag set (Information)

Issue detail

The following cookie was issued by the application and does not have the HttpOnly flag set:

- cookie_prefs

The cookie does not appear to contain a session token, which may reduce the risk associated with this issue. You should review the contents of the cookie to determine its function.

Issue background

If the HttpOnly attribute is set on a cookie, then the cookie's value cannot be read or set by client-side JavaScript. This measure makes certain client-side attacks, such as cross-site scripting, slightly harder to exploit by preventing them from trivially capturing the cookie's value via an injected script.

Issue remediation

There is usually no good reason not to set the HttpOnly flag on all cookies. Unless you specifically require legitimate client-side scripts within your application to read or set a cookie's value, you should set the HttpOnly flag by including this attribute within the relevant Set-cookie directive.

You should be aware that the restrictions imposed by the HttpOnly flag can potentially be circumvented in some circumstances, and that numerous other serious attacks can be delivered by client-side script injection, aside from simple cookie stealing.

References

- Configuring HttpOnly

Vulnerability classifications

- CWE-16: Configuration

```
Set-Cookie: cookie_prefs=; Path=/; Expires=Thu, 01 Jan 1970 00:00:00 GMT; Secure; SameSite=None
Set-Cookie: cookie_prefs=P%3D1%2CF%3D1%2Ctype%3Dimplicit; Max-Age=31536000; Domain=.paypal.com; Path=/; Expires=Sat, 23 Oct 2021 13:05:22 GMT; Secure; SameSite=None
Set-Cookie: x-pp-s=eyJ0IjoiMTY
```

9.2 Summary of https:/api-3t./paypal.com

9.2.1 XML Injection (Medium Severity)

Issue detail

The URL path filename appears to be vulnerable to XML injection. The payload **]]>><** was appended to the value of the URL path filename. The application's response indicated that this input may have caused an error within a server-side XML or SOAP parser, suggesting that the input has been inserted into an XML document or SOAP message without proper sanitization.

Issue background

XML or SOAP injection vulnerabilities arise when user input is inserted into a server-side XML document or SOAP message in an unsafe way. It may be possible to use XML metacharacters to modify the structure of the resulting XML. Depending on the function in which the XML is used, it may be possible to interfere with the application's logic, to perform unauthorized actions or access sensitive data.

This kind of vulnerability can be difficult to detect and exploit remotely; you should review the application's response, and the purpose that the relevant input performs within the application's functionality, to determine whether it is indeed vulnerable.

Issue remediation

The application should validate or sanitize user input before incorporating it into an XML document or SOAP message. It may be possible to block any input containing XML metacharacters such as < and >. Alternatively, these characters can be replaced with the corresponding entities: < and >.

Vulnerability classifications

- CWE-91: XML Injection (aka Blind XPath Injection)
- CWE-116: Improper Encoding or Escaping of Output

- CWE-159: Failure to Sanitize Special Element
- CWE-611: Improper Restriction of XML External Entity Reference ('XXE')
- CWE-776: Improper Restriction of Recursive Entity References in DTDs ('XML Entity Expansion')

Request

GET /robots.txt]]%3e%3e%3c HTTP/1.1
Host: api-3t.paypal.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en-US,en-GB;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36
Connection: close
Cache-Control: max-age=0

Response

HTTP/1.1 404 Not Found
Content-Type: text/xml; charset=utf-8
Content-Length: 1156
Connection: close
Date: Fri, 23 Oct 2020 14:07:44 GMT
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Paypal-Debug-Id: 86ab255b97c93
Strict-Transport-Security: max-age=31536000; includeSubDomains

<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:cc="urn:ebay:apis:CoreComponentTypes" xmlns:ed="urn:ebay:apis:EnhancedDataTypes" xmlns:wsu="http://schemas.xmlsoap.
...[SNIP]...

9.2.2 HTTP Strict Transport Security (HSTS) Errors and Warnings (Medium Severity)

Impact

The HSTS warning and error may allow attackers to bypass HSTS, effectively allowing them to read and modify your communication with the website

Error

Preload directive not present

Resolution

Submit domain for inclusion in browser's HTTP Strict Transport Security preload list

Remedy

Ideally, after fixing the errors and warnings, you should consider adding your domain to the the HSTS preload list. This will ensure that browsers automatically connect your website by using HTTPS, actively preventing users from visiting your site using HTTP. Since this list is hardcoded in users' browsers, it will enable HSTS even before they visit your page for the first time, eliminating the need for Trust On First Use (TOFU) with its associated risks and disadvantages. Unless you fix the errors and warnings your website will not meet the conditions required to enter the browser's preload list.

Browser vendors declared.

- Serve a valid certificate
- If you are listening on port 80, redirect all domains from HTTP to HTTPS on the same host, Serve all subdomains over HTPPS
- Serve an HSTS header on the base domain for HTTPS requests.

9.2.3 Robots.txt file (Information)

Issue detail

The web server contains a robots.txt file.

Issue background

The file robots.txt is used to give instructions to web robots, such as search engine crawlers, about locations within the web site that robots are allowed, or not allowed, to crawl and index.

The presence of the robots.txt does not in itself present any kind of security vulnerability. However, it is often used to identify restricted or private areas of a site's contents. The information in the file may therefore help an attacker to map out the site's contents, especially if some of the locations identified are not linked from elsewhere in the site. If the application relies on robots.txt to protect access to these areas, and does not enforce proper access control over them, then this presents a serious vulnerability.

Issue remediation

The robots.txt file is not itself a security threat, and its correct use can represent good practice for non-security reasons. You should not assume that all web robots will honor the file's instructions. Rather, assume that attackers will pay close attention to any locations identified in the file. Do not rely on robots.txt to provide any kind of protection over unauthorized access.

Vulnerability classifications

- CWE-200: Information Exposure

Request

```
GET /robots.txt HTTP/1.1
Host: api-3t.paypal.com
Connection: close
Cookie: cookie_check=yes; _gcl_au=1.1.1263603764.1602833384; _ga=GA1.2.1105157341.1603302958; _gid=GA1.2.1711384619.1603438251;
ui_experience=d_id%3Da5c2eaae2c464288ba440126014ea5511603438296149; _gat_gtag_UA_53389718_12=1;
tcs=main%3Amktg%3Apersonal%3A%3Ahome%7Cheader%7Clogin; x-cdn=fastly:MRS; sc_f=Bu1oCWmTzsbqDvRGr0D9PF-JttI4H3-cQaIcv6ryC7OTJRxVqP-
9F5p_sE3nqqVSB6BcKvNgTtHi9-IRHRKnWuL6IE8X7OxSywwFtm;
KHcl0EuY7AKSMgfvHI7J5E7hPtK=8kZy_6wBnSPFY1qfrQ1_uSP6ADMNejJclvVUmjb4C6T5iVL98VGO3ELI2CaCLvT1ttRLvKi8j_ln_Tlb; l7_az=dcg13.slc;
LANG=en_US%3BLK; tsrce=mppnodeweb; x-pp-s=eyJ0IjoiMTYwMzQ2MDI5ODk4NiIsImwiOiIwIiwibSI6IjAifQ;
ts_c=vr%3D30506c9d1750a1d4f083d325fdc0b3a6%26vt%3D55ae78521750a48b2a994f03fcfd9206;
ts=vreXpYrS%3D1698068318%26vteXpYrS%3D1603462118%26vr%3D30506c9d1750a1d4f083d325fdc0b3a6%26vt%3D55ae78521750a48b2a994f03fcfd9206
%26vtyp%3Dreturn
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/plain
Content-Length: 72
Connection: close
Date: Fri, 23 Oct 2020 13:57:09 GMT
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Paypal-Debug-Id: d2741376a2b83
Strict-Transport-Security: max-age=31536000; includeSubDomains

# PayPal robots.txt file for api-3t.paypal.com
User-agent: *
Disallow: /
```

9.2.4 Missing X-XSS Protection Header (Information)

The scan detected a missing X-XSS-Protectionheader which means that this website could be at risk of a Cross-site Scripting (XSS) attacks.

Impact

This issue is reported as additional information only. There is no direct impact arising from this issue.

Remedy

Add the X-XSS-Protection header with a value of "1; mode= block"

- X-XSS-Protection: 1; mode=block

9.3 Summary of https://developer.paypal.com

9.3.1 TLS Certificate (Medium Severity)

Issue detail

The following problem was identified with the server's TLS certificate:

- The server's certificate is not trusted.

Note: Burp relies on the Java trust store to determine whether certificates are trusted. The Java trust store does not include every root CA certificate that is included within browser trust stores. Burp might incorrectly report that a certificate is not trusted, if a valid root CA certificate is being used that is not included in the Java trust store.

The server presented the following certificates:

Server certificate

Issued to: developer.paypal.com
Issued by: DigiCert EV RSA CA G2
Valid from: Tue Jul 07 20:00:00 EDT 2020
Valid to: Wed Jul 13 08:00:00 EDT 2022

Certificate chain #1

Issued to: DigiCert EV RSA CA G2
Issued by: DigiCert Global Root G2
Valid from: Thu Jul 02 08:42:50 EDT 2020
Valid to: Tue Jul 02 08:42:50 EDT 2030

Certificate chain #2

Issued to: DigiCert Global Root G2
Issued by: VeriSign Class 3 Public Primary Certification Authority - G5
Valid from: Sun Nov 05 19:00:00 EST 2017
Valid to: Sat Nov 05 19:59:59 EDT 2022

Certificate chain #3

Issued to: VeriSign Class 3 Public Primary Certification Authority - G5
Issued by: VeriSign Class 3 Public Primary Certification Authority - G5
Valid from: Tue Nov 07 19:00:00 EST 2006
Valid to: Wed Jul 16 19:59:59 EDT 2036

Issue background

TLS (or SSL) helps to protect the confidentiality and integrity of information in transit between the browser and server, and to provide authentication of the server's identity. To serve this purpose, the server must present an TLS certificate that is valid for the server's hostname, is issued by a trusted authority and is valid for the current date. If any one of these requirements is not met, TLS connections to the server will not provide the full protection for which TLS is designed.

It should be noted that various attacks exist against TLS in general, and in the context of HTTPS web connections in particular. It may be possible for a determined and suitably-positioned attacker to compromise TLS connections without user detection even when a valid TLS certificate is used.

References

- SSL/TLS Configuration Guide

Vulnerability classifications

- CWE-295: Improper Certificate Validation
- CWE-326: Inadequate Encryption Strength
- CWE-327: Use of a Broken or Risky Cryptographic Algorithm

9.3.2 Backup file (Information)

Issue description

Publicly accessible backups and outdated copies of files can provide attackers with extra attack surface. Depending on the server configuration and file type, they may also expose source code, configuration details, and other information intended to remain secret.

Issue remediation

Review the file to identify whether it's intended to be publicly accessible, and remove it from the server's web root if it isn't. It may also be worth auditing the server contents to find other outdated files, and taking measures to prevent the problem from recurring.

References

- Review Old, Backup and Unreferenced Files for Sensitive Information

Vulnerability classifications

- CWE-530: Exposure of Backup File to an Unauthorized Control Sphere

Request 1

```
GET /home~ HTTP/1.1
Host: developer.paypal.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
Cookie: nsid=s%3AQXC4EuO6tTECmFwiTtkOFuzcBeqlF_J8.IYiW4tAhflrlgRDV8z5yZxW02YrazN3gG9r3mRu5MxY; ppcp_ramp=classic; cookie_check=yes;
_gcl_au=1.1.1263603764.1602833384; _ga=GA1.2.1105157341.1603302958; _gid=GA1.2.1711384619.1603438251;
ui_experience=d_id%3Da5c2eaae2c464288ba440126014ea5511603438296149; _gat_gtag_UA_53389718_12=1;
tcs=main%3Amktg%3Apersonal%3A%3Ahome%7Cheader%7Clogin; x-cdn=fastly:MRS; sc_f=Bu1oCWmTzsbqDvRGr0D9PF-Jttl4H3-
cQalcv6ryC7OTJRxVqP-9F5p_sE3nqqVSB6BcKvNgTtHi9-IRHRKnWuL6IE8X7OxSywwFtm;
KHcl0EuY7AKSMgfvHI7J5E7hPtK=8kZy_6wBnSPFY1qfrQ1_uSP6ADMNejJclvVUmjb4C6T5iVL98VGO3ELI2CaCLvT1ttRLvKi8j_In_Tlb;
tsrce=cspreportnodeweb; LANG=en_US%3BUS; enforce_policy=ccpa; x-pp-s=eyJ0IjoiMTYwMzQ2NjI4MDI2MSIsImwiOiIwIiwibSI6IjAifQ; I7_az=dcg12.slc;
ts=vreXpYrS%3D1698074281%26vteXpYrS%3D1603468081%26vr%3D30506c9d1750a1d4f083d325fdc0b3a6%26vt%3D5608cef5175ac120001fec30ffd306
13%26vtyp%3Dreturn; ts_c=vr%3D30506c9d1750a1d4f083d325fdc0b3a6%26vt%3D5608cef5175ac120001fec30ffd30613
```

Response 1

```
HTTP/1.1 302 Found
Content-Type: text/html; charset=utf-8
Content-Length: 58
Connection: close
Server: nginx
Date: Fri, 23 Oct 2020 15:24:04 GMT
Location: /home~/
Vary: Accept
Strict-Transport-Security: max-age=31536000; includeSubDomains

<p>Found. Redirecting to <a href="/home~/">/home~/</a></p>
```

9.4 Summary of https://business.paypal.com

9.4.1 Breach Attack Detected

I 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

Remedy

Scan reported a Possible BREACH Attack issue because the target web page meets the following conditions that facilitate it:

- Served from a server that uses HTTP-level compression (ie. gzip)
- Reflects user-input in the HTTP response bodies
- Contains sensitive information (such as a CSRF token) in HTTP response bodies

To mitigate the issue, we recommend the following solutions:

- If possible, disable HTTP level compression
- Separate sensitive information from user input

- Protect vulnerable pages with CSRF token. The SameSite Cookie attribute will mitigate this issue, because to exploit this issue an attacker forces the victim to visit a target website using invisible frames

Request

```
GET /?%2f%2fr87?com%2f%3f=%2527 HTTP/1.1
Host: www.paypal.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Connection: Keep-Alive
Cookie: cookie_check=yes; nsid=s%3Ao5YzhS6WMuwzTO-dG-u0F8weR4zjTf1o.SScaJ2%2B7zTdrkH2NbmrGJhUbfY%2B9TVY
VCD8pq66laHE; connect.sid=s%3Amjfbu-1C0qwNmZE13xHwBb2PPKmLoz3b.m85HcoGOpNztp1TDunFqp8We69x80UheRMYA2RQ%
2Bvvk; l7_az=dcg12.slc; ts=vreXpYrS%3D1697446788%26vteXpYrS%3D1602840588%26vr%3D30a2acd11750a1d5a38b590
6fdbf46d3%26vt%3D30a2acd11750a1d5a38b5906fdbf46d2%26vtyp%3Dnew; ts_c=vr%3D30a2acd11750a1d5a38b5906fdbf4
6d3%26vt%3D30a2acd11750a1d5a38b5906fdbf46d2; x-cdn=fastly:MRS; tsrce=errorsnodeweb; LANG=en_US%3BUS; co
okie_prefs=T%3D1%2CP%3D1%2CF%3D1%2Ctype%3Dexplicit_banner; enforce_policy=ccpa; x-pp-s=eyJ0IjoiMTYwMjgz
OTAyNDM0MCIsImwiOiIxIiwibSI6IjAifQ
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 1237.613 Total Bytes Received : 24594 Body Length : 21115 Is Compressed : No

```
HTTP/1.1 200 OK
X-Cache: MISS, MISS
X-Timer: S1602839025.472133,VS0,VE253
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Etag: W/"51a3-SBA8ItTy8ZbCcxvHMwDb2Zoc+Tw"
Set-Cookie: enforce_policy=ccpa; Max-Age=31536000; Domain=.paypal.com; Path=/; Expires=Sat, 16 Oct 2021
 09:03:45 GMT; Secure; SameSite=None
Set-Cookie: ui_experience=d_id%3D22e4a0ff9c3f48648951496d96c3c8891602839025591; Max-Age=63113851; Domai
n=.paypal.com; Path=/; Expires=Sun, 16 Oct 2022 20:41:16 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: LANG=en_US%3BUS; Max-Age=31556; Domain=.paypal.com; Path=/; Expires=Fri, 16 Oct 2020 17:49:
41 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: tsrce=unifiedloginnodeweb; Max-Age=259199; Domain=.paypal.com; Path=/; Expires=Mon, 19 Oct
 2020 09:03:44 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: HaC80bwXscjqZ7KM6VOxULOB534=; Path=/; Expires=Thu, 01 Jan 1970 00:00:00 GMT; HttpOnly; Secu
re; SameSite=None
Set-Cookie: cookie_prefs=; Path=/; Expires=Thu, 01 Jan 1970 00:00:00 GMT; Secure; SameSite=None
Set-Cookie: cookie_prefs=T%3D1%2CP%3D1%2CF%3D1%2Ctype%3Dexplicit_banner; Max-Age=31536000; Domain=.payp
al.com; Path=/; Expires=Sat, 16 Oct 2021 09:03:45 GMT; Secure; SameSite=None
Set-Cookie: x-pp-s=eyJ0IjoiMTYwMjgzOTAyNTYzNiIsImwiOiIxIiwibSI6IjAifQ; Domain=.paypal.com; Path=/; Http
Only; Secure; SameSite=None
Set-Cookie: l7_az=dcg12.slc; Path=/; Domain=paypal.com; Expires=Fri, 16 Oct 2020 09:33:45 GMT; HttpOnl
y; Secure; SameSite=None
Set-Cookie: ts=vreXpYrS%3D1697447025%26vteXpYrS%3D1602840825%26vr%3D30a2acd11750a1d5a38b5906fdbf46d3%26
vt%3D30a2acd11750a1d5a38b5906fdbf46d2%26vtyp%3Dnew; Path=/; Domain=paypal.com; Expires=Mon, 16 Oct 2023
 09:03:45 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: ts_c=vr%3D30a2acd11750a1d5a38b5906fdbf46d3%26vt%3D30a2acd11750a1d5a38b5906fdbf46d2; Path=/;
 Domain=paypal.com; Expires=Mon, 16 Oct 2023 09:03:45 GMT; Secure; SameSite=None
Set-Cookie: x-cdn=fastly:MRS; Domain=paypal.com; Path=/; Secure
Strict-Transpo
…
```

9.4.2 An Unsafe Content Security Policy (CSP) Directive in Use

Netsparker detected that one of following CSP directives is used:

- unsafe-eval
- unsafe-inline

By using `unsafe-eval`, you allow the use of string evaluation functions like `eval`.

By using `unsafe-inline`, you allow the execution of inline scripts, which almost defeats the purpose of CSP. When this is allowed, it's very easy to successfully exploit a Cross-site Scripting vulnerability on your website.

Impact

An attacker can bypass CSP and exploit a Cross-site Scripting vulnerability successfully.

Vulnerabilities

9.1. https://business.paypal.com/.well-known/

Unsafe Directive Used In Csp

- unsafe-inline

Certainty

Request

```
GET /.well-known/ HTTP/1.1
Host: business.paypal.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: LANG=en_US%3BUS; cookie_check=yes; enforce_policy=ccpa; l7_az=dcg12.slc; ts=vreXpYrS%3D16974467
68%26vteXpYrS%3D1602840568%26vr%3D30a2acd11750a1d5a38b5906fdbf46d3%26vt%3D30a2acd11750a1d5a38b5906fdbf4
6d2%26vtyp%3Dnew; ts_c=vr%3D30a2acd11750a1d5a38b5906fdbf46d3%26vt%3D30a2acd11750a1d5a38b5906fdbf46d2; t
srce=unifiedloginnodeweb; x-cdn=fastly:MRS; x-pp-s=eyJ0IjoiMTYwMjgzODc2ODkzNiIsImwiOiIwIiwibSI6IjAifQ
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 656.5799 Total Bytes Received : 13829 Body Length : 12004 Is Compressed : No

```
HTTP/1.1 421 Misdirected Request
Set-Cookie: enforce_policy=ccpa; Max-Age=31536000; Domain=.paypal.com; Path=/; Expires=Sat, 16 Oct 2021
 08:59:31 GMT; Secure; SameSite=None
Set-Cookie: LANG=en_US%3BUS; Max-Age=31556; Domain=.paypal.com; Path=/; Expires=Fri, 16 Oct 2020 17:45:
27 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: tsrce=errorsnodeweb; Max-Age=259199; Domain=.paypal.com; Path=/; Expires=Mon, 19 Oct 2020 0
8:59:30 GMT; HttpOnly; Secure; SameSite=None
Set-Cookie: x-pp-s=eyJ0IjoiMTYwMjgzODc3MTgzMyIsImwiOiIwIiwibSI6IjAifQ; Domain=.paypal.com; Path=/; Http
Only; Secure; SameSite=None
Set-Cookie: connect.sid=s%3AI6mvgls_tzHhfcibnbvwsWomc_f7w87X.LwKYnUissqS3qGbKDorP95goIKt4aQyM24WClHn4hF
A; Path=/; HttpOnly
Server: nginx
x-content-type-options: nosniff
Connection: keep-alive
x-xss-protection: 1; mode=block
content-security-policy: default-src 'self' https://*.paypal.com https://*.paypalobjects.com; script-sr
c 'nonce-3Z0/ZkeqH/ZdV0RhfwWSlroCMXf6PSzwWE20RMZpovLaquBl' 'self' https://*.paypal.com https://*.paypal
objects.com 'unsafe-inline'; img-src https: data:; object-src 'none'; font-src 'self' https://*.paypal.
com https://*.paypalobjects.com; form-action 'self' https://*.paypal.com; base-uri 'self' https://*.pay
pal.com; style-src 'self' 'unsafe-inline' https://*.paypal.com https://*.paypalobjects.com; block-all-m
ixed-content;; report-uri https://www.paypal.com/csplog/api/log/csp
Content-Length: 12004
X-Recruiting: If you are reading this, maybe you should be working at PayPal instead! Check out http
s://www.paypal.com/us/webapps/mpp/paypal-jobs
x-frame-options: SAMEORIGIN
HTTP_X_PP_AZ_LOCATOR: dcg13.slc
ETag: W/"2ee4-dSmVuK/Vd5SvDtyAWVWZc6i+wOU"
Paypal-Debug-Id: e7c27ba0c9792
Content-Type: te
…
licy: default-src 'self' https://*.paypal.com https://*.paypalobjects.com; script-src 'nonce-3Z0/ZkeqH/
ZdV0RhfwWSlroCMXf6PSzwWE20RMZpovLaquBl' 'self' https://*.paypal.com https://*.paypalobjects.com 'unsafe
-inline'; img-src https: data:; object-src 'none'; font-src 'self' https://*.paypal.com https://*.paypa
lobjects.com; form-action 'self' https://*.paypal.com; base-uri 'self' https://*.paypal.com; style-src
 'self' 'unsafe-inline' https://*.paypal.com https://*.paypalobjects.com; block-all-mixed-content;; rep
ort-uri https://www.paypal.com/csplog/api/log/csp
Content-Length: 12004
X-Recruiting: If you are reading this, maybe
…
```

Remedy

If possible remove `unsafe-eval`and `unsafe-inline`from your CSP directives.

9.5 Summary of https://safetyhub.paypal.com

9.5.1 TLS Cookie without secure flag set

Issue detail

The following cookies were issued by the application and do not have the secure flag set:

- lul
- luiacs

The cookies do not appear to contain session tokens, which may reduce the risk associated with this issue. You should review the contents of the cookies to determine their function.

Issue background

If the secure flag is set on a cookie, then browsers will not submit the cookie in any requests that use an unencrypted HTTP connection, thereby preventing the cookie from being trivially intercepted by an attacker monitoring network traffic. If the secure flag is not set, then the cookie will be transmitted in clear-text if the user visits any HTTP URLs within the cookie's scope. An attacker may be able to induce this event by feeding a user suitable links, either directly or via another web site. Even if the domain that issued the cookie does not host any content that is accessed over HTTP, an attacker may be able to use links of the form http://example.com:443/ to perform the same attack.

To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

Issue remediation

The secure flag should be set on all cookies that are used for transmitting sensitive data when accessing content over HTTPS. If cookies are used to transmit session tokens, then areas of the application that are accessed over HTTPS should employ their own session handling mechanism, and the session tokens used should never be transmitted over unencrypted communications.

Vulnerability classifications

- CWE-614: Sensitive Cookie in HTTPS Session Without 'Secure' Attribute

Request

```
GET /AIP/portal/home.do HTTP/1.1
Host: safetyhub.paypal.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en-US,en-GB;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: TS0154e00a=01ccd508a781d5a5a87ae92adaf722af04e90ee0cdc9326214e4c47a78a9d4399efaecfdcef1f96b31058ae5b90a955bfba28cbb25;
JSESSIONID=E0941735D2CA12F5DED65A58C13BE67E;
TS0147e78b=01ccd508a78febda6cf1b98b7b91408858fcd030f1c9326214e4c47a78a9d4399efaecfdcef84a1926eac83c2b5da48329e4ca1d5d;
```

Response

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Language: en-US
Content-Type: text/html;charset=UTF-8
Date: Fri, 23 Oct 2020 15:34:54 GMT
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: lul=ZW5fVVM=; Max-Age=63072000; Expires=Sun, 23-Oct-2022 15:34:55 GMT
Set-Cookie: lul=ZW5fVVM=; Max-Age=63072000; Expires=Sun, 23-Oct-2022 15:34:55 GMT
Set-Cookie: luiacs=533c0728-dd36-4c9a-85b9-7d29575e2101; Max-Age=63072000; Expires=Sun, 23-Oct-2022 15:34:55 GMT
Strict-Transport-Security: max-age=31536000
X-Frame-Options: DENY
Connection: close
Strict-Transport-Security: max-age=63072000
Via: 1.1 sin1-bit12
Set-Cookie:
TS0154e00a=01ccd508a7438f9b3e00867b5fbe2910e83155530ec9326214e4c47a78a9d4399efaecfdceca85df1fd2f4f0352980d515b9e5d4947c79c3116536e
082b51fe0dca2aa829c0ebccdf0bc49939204903a61ec234d32bc8b337dc8d83782bacb0da23f9ea4bc; Path=/; Secure; HTTPOnly
Content-Length: 17986
```

9.6 Summary of https://registration.paypal.com

9.6.1 Out-of-Date Version (jQuery)

Impact

Since this is an old version of the software, it may be vulnerable to attacks.

jquery Cross-Site Scripting (XSS) Vulnerability

jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.

Affected Versions

1.2.1 to 1.11.3

External References

- CVE-2019-11358

jquery Cross-Site Scripting (XSS) Vulnerability

jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.

Affected Versions

1.2.1 to 1.11.3

External References

- CVE-2015-9251

Vulnerabilities

2.1. https://registration.paypal.com/js/min/phoenix.js?version=142126110301770400000027246026405014312340717763516255 07

Method	Parameter	Value
GET	version	142126110301770400000027246026405014312340717763516255 07

Identified Version

- 1.8.2

Latest Version

- 1.12.4 (in this branch)

Branch Status

- This branch has stopped receiving updates since 6/20/2016.

Vulnerability Database

- Result is based on 04/27/2020 17:30:00 vulnerability database content.

Certainty

Request

```
GET /js/min/phoenix.js?version=142126110301770400000027246026405014312340717763516255O7 HTTP/1.1
Host: registration.paypal.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: JSESSIONID=D5IwWYWnoXlDFBNZT-sdz7M_hzPkofjfiJxuDlCl8yBFK6oW1wJ0!-101421291!1209845467
Referer: https://registration.paypal.com/
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 1718.7459 Total Bytes Received : 753514 Body Length : 752705 Is Compressed : No

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Server: Apache
X-XSS-Protection: 1; mode=block
content-security-policy: default-src 'self' https://*.paypal.com https://*.paypalobjects.com; frame-src
 'self' https://*.paypal.com https://*.paypalobjects.com; script-src 'self' https://*.paypal.com http
s://*.paypalobjects.com 'unsafe-inline' 'unsafe-eval'; connect-src 'self' https://*.paypal.com https://
*.paypalobjects.com; style-src 'self' https://*.paypal.com https://*.paypalobjects.com 'unsafe-inline';
 img-src 'self' https: data:;
X-Frame-Options: SAMEORIGIN
Last-Modified: Wed, 09 Sep 2020 14:06:50 GMT
Strict-Transport-Security: max-age=31536000; includeSubDomains
Accept-Ranges: bytes
Content-Type: text/javascript
Content-Encoding:
Date: Fri, 16 Oct 2020 07:40:11 GMT
Vary:
…
f(prop.search("height|width|left|top")==-1){value=(value==1||value===true)?"yes":"no"}if(value&&prop!
="name"){settings+=prop+"="+value+","}}var win=window.open(anchor.href,name,settings);return win};
/* jQuery v1.8.2jquery.com | jquery.org/license */
(function(a,b){function G(a){var b=F[a]={};return p.each(a.split(s),function(a,c){b[c]=!0}),b}function
 J(a,c,d){if(d===b&&a.nodeType===1){var e="data-"+c.replace(I,
…
```

Remedy

Please upgrade your installation of jQuery to the latest stable version.

Note :

- **All the screenshots and codes are more prominently explained in the demonstration video because this document was made in parallel with the workout in the video.**