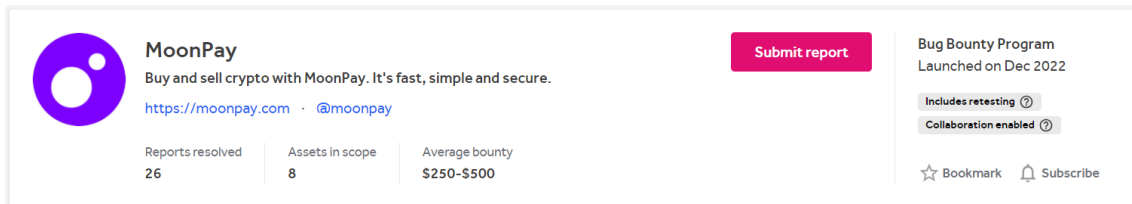


Informe de Recopilación de Información de Ciberseguridad

INTRODUCCION

El presente informe detalla el proceso de recopilación de información llevado a cabo por Sebastian Flores en el ámbito de la ciberseguridad para la compañía Moonpay. El objetivo principal de esta iniciativa fue obtener una comprensión clara y completa de los aspectos relacionados con la seguridad cibernética en el entorno empresarial de moonpay.com.

Scope “moonpay.com”



The screenshot shows the MoonPay profile on the HackerOne platform. It includes the MoonPay logo, a description of the company, and a table of statistics. To the right, there is a 'Submit report' button and details about the bug bounty program, including its launch date and features like retesting and collaboration.

Reports resolved	Assets in scope	Average bounty
26	8	\$250-\$500

Bug Bounty Program
Launched on Dec 2022

Includes retesting ⓘ
Collaboration enabled ⓘ

☆ Bookmark 🔔 Subscribe

https://hackerone.com/moonpay/policy_scopes

MoonPay es una plataforma de servicios financieros en línea que permite a los usuarios comprar criptomonedas con monedas fiduciarias actuando como un intermediario entre los usuarios y los intercambios de divisas.

Resumen Inicial del Proceso de Reconocimiento Vertical

Durante el proceso de reconocimiento vertical llevado a cabo en moonpay.com, se emplearon diversas técnicas y herramientas especializadas para recopilar información específica sin comprometer la seguridad del sistema. Estas herramientas se utilizaron para obtener una comprensión profunda y detallada de la infraestructura digital de moonpay.com sin realizar intrusiones.

FOOTPRINTING

En la fase de Footprinting, se utilizaron herramientas como "shuffIDNS" para ataques de fuerza bruta y "dnsvalidator" para validar la información obtenida, identificando servidores y registros DNS asociados con moonpay.com. Además, se analizaron datos públicos de Google Analytics para entender el tráfico del sitio web y se verificaron los certificados TLS con la herramienta "Cero". La técnica de Web Scraping con "Katana" permitió recopilar datos públicos y explorar posibles vulnerabilidades en el sitio. También se realizaron búsquedas de subdominios con "CTFR" y "Gau". Para organizar y filtrar la información, se empleó la herramienta "Unfurl". También se capturaron imágenes de subdominios web con "Gowitness"

FINGERPRINTING

En la fase de Fingerprinting, se realizaron escaneos del Sistema Autónomo de moonpay.com con "Nmap" y "Masscan". Los subdominios encontrados fueron validados con "httpx", se verificó la presencia de un firewall WAF y se realizaron pruebas de fuzzing con "FFUF". También se empleó "Nuclei" para detectar vulnerabilidades y "Wpscan" para escanear WordPress en busca de configuraciones específicas del CMS. Se llevó a cabo un análisis adicional de SSL/TLS y se revisaron las medidas de seguridad contra spoofing en los servidores de correo electrónico. Además se identificaron tecnologías específicas con "Wappalyzer" y "WHATWEB". También se realizó un análisis detallado de SSL/TLS con "textssl.sh" y se evaluaron configuraciones de seguridad del servidor de correo electrónico con "spoofcheck.py". "Subzy" se usó para identificar subdominios en desuso.

OSINT (Open Source Intelligence)

En la fase de OSINT, se utilizaron herramientas como "Maltego" para investigar posibles vulneraciones de correos electrónicos de empleados, motores de búsqueda de Google para localizar archivos PDF sensibles, "Dedigger.com" para buscar información expuesta en Google Drive y "Exiftool" para analizar metadatos de archivos. "Spiderfoot" se empleó para identificar correos electrónicos asociados al dominio objetivo, y "Aware" validó las direcciones de correo electrónico encontradas. Además, "GithubSearch" se usó para buscar archivos GIT expuestos y "FOCA" automatizó la recopilación de información de subdominios públicos, proporcionando una visión completa de la infraestructura digital de la organización.

Este proceso exhaustivo de reconocimiento vertical permitió una evaluación detallada de la seguridad digital de moonpay.com, identificando posibles vulnerabilidades y puntos de entrada que podrían ser objeto de futuros análisis y medidas de seguridad.

FOOTPRINTING

Ataques de fuerza bruta shufflDNS y dnsvalidator:

Primero, instalamos las herramientas y descargamos listas adicionales desde fuentes confiables. Utilizamos el repositorio de Daniel Miessler y otras fuentes proporcionadas por el profesor para obtener listas variadas de fuerza bruta.

COMANDOS

```
git clone https://github.com/danielmiessler/SecLists.git /home/sebastian/recopilacion
```

```
wget
```

```
https://raw.githubusercontent.com/danielmiessler/SecLists/master/Discovery/DNS/subdomains-top1million-5000.txt -O domains.txt
```

```
wget
```

```
https://raw.githubusercontent.com/blechschmidt/massdns/master/lists/resolvers.txt -O resolvers.txt
```

Validación de Servidores DNS Activos

Validamos los servidores DNS activos utilizando dnsvalidator y la lista resolvers.txt. Los resultados se guardan en un archivo de texto.

COMANDOS

```
dnsvalidator -tL
```

```
https://raw.githubusercontent.com/blechschmidt/massdns/master/lists/resolvers.txt -threads 20 -o $HOME/recopilacion/lists/resolvers.txt
```

Búsqueda de Subdominios con shufflDNS

Utilizamos shuffledns para buscar subdominios válidos utilizando las listas domains.txt y bug-bounty-program-subdomains-trickest-inventory.txt. Los resultados se almacenan en archivos subdominios.txt y subdominios2.txt.

COMANDOS

```
shuffledns -d moonpay.com -w $HOME/recopilacion/lists/domains.txt -r  
$HOME/recopilacion/lists/resolvers.txt -silent > subdominios.txt
```

```
shuffledns -d moonpay.com -w  
/home/sebastian/recopilacion/danielmiessler/SecLists/Discovery/DNS/bug-bounty-  
program-subdomains-trickest-inventory.txt -r  
/home/sebastian/recopilacion/danielmiessler/listasrecopilacion/resolvers.txt -silent >  
subdominios2.txt
```

Unión, Eliminación de Duplicados y Ordenación

Unimos las dos listas de subdominios y eliminamos las entradas duplicadas para obtener una lista depurada. Luego, ordenamos la lista para su análisis.

COMANDO

```
cat subdominios.txt subdominios2.txt | sort -u > fuerzabruta.txt
```

Este proceso de ataque de fuerza bruta y reconocimiento de subdominios ha permitido recopilar una lista completa y depurada de servidores DNS activos asociados a Moonpay.com.

Google Analytics:

A pesar de nuestros esfuerzos en el análisis de datos públicos de Google Analytics a través de la herramienta analyticsrelationships, lamentablemente no se obtuvieron resultados específicos para moonpay.com. Es posible que la configuración de privacidad del sitio web o la disponibilidad de datos públicos hayan limitado la obtención de información a través de esta fuente.

```
(root@kali)-[/home/.../recopilacion/listas/listasprueba/prueba]
# analyticsrelationships --url www.moonpay.com/es

UA-ID
DOMAINS

> Get related domains / subdomains by looking at Google Analytics IDs
> Python version
> By @JosueEncinar

[+] Analyzing url: https://www.moonpay.com/es
[-] Tagmanager URL not found

(root@kali)-[/home/.../recopilacion/listas/listasprueba/prueba]
#
```

Certificados TLS Proving:

Durante el proceso de reconocimiento vertical de Moonpay.com, se intentó verificar los certificados TLS asociados con el dominio utilizando la herramienta "Cero".

A pesar del intento de verificación, la herramienta "Cero" no arrojó resultados específicos para Moonpay.com. Esto puede deberse a diversas razones, como configuraciones de privacidad, políticas del servidor o limitaciones técnicas de la herramienta.

```
(sebastian@kali)-[~]
$ cero -d moonpay.com
moonpay.com
```

Web Scraping con Katana:

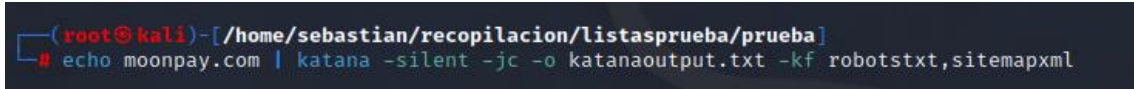
En el proceso de reconocimiento vertical de Moonpay.com, se utilizó la técnica de web scraping para recopilar datos públicos y explorar posibles vulnerabilidades o información sensible expuesta en el sitio web. Se emplearon las herramientas Katana para realizar el scraping y Unfurl para filtrar y extraer los subdominios relevantes.

Web Scraping

Se utilizó la herramienta Katana para realizar web scraping en Moonpay.com. Se pasó el dominio a Katana utilizando el comando echo para analizar. Los resultados se guardaron en un archivo llamado katanaoutput.txt.

COMANDO

```
echo moonpay.com | katana -silent -jc -o katanaoutput.txt -kf robotstxt,sitemapxml
```



```
(root@kali)-[/home/sebastian/recopilacion/listasprueba/prueba]
# echo moonpay.com | katana -silent -jc -o katanaoutput.txt -kf robotstxt,sitemapxml
```

Filtrado de Subdominios con Unfurl

Los subdominios obtenidos con Katana se filtraron y se extrajo la información relevante utilizando la herramienta Unfurl. Los resultados se guardaron en un archivo llamado katanadominios.txt.

COMANDO

```
cat katanaoutput.txt | unfurl --unique > moonpaydominios.txt
```



```
(root@kali)-[/home/sebastian/recopilacion/listasprueba]
# cat katanaoutput.txt | unfurl --unique domains > katanadominios.txt
```

El proceso de web scraping y filtrado de subdominios ha permitido recopilar una lista completa y depurada de subdominios asociados a Moonpay.com.

Búsqueda de Subdominios con CTFR:

Se empleó la aplicación CTFR (Certificate Transparency Log Framework Reconnaissance) para buscar subdominios a través de los registros de transparencia de certificados (CT). Esta herramienta proporciona una forma efectiva de descubrir subdominios asociados a un dominio específico.

Aplicación del Comando CTFR

Se utilizó el comando `ctfr -org -d moonpay.com` para buscar subdominios a través de los registros de transparencia de certificados en el dominio Moonpay.com. La opción `-org` indica la búsqueda a nivel de organización y `-d moonpay.com` especifica el dominio de interés.

COMANDO

```
ctfr -org -d moonpay.com | unfurl --unique domains > ctfrdomains.txt
```

El uso de la aplicación CTFR ha proporcionado una lista de subdominios descubiertos a través de los registros de transparencia de certificados en Moonpay.com.

Búsqueda de Subdominios con GAU:

Se utilizó la herramienta Gau para buscar subdominios del dominio específico. Gau es una herramienta de línea de comandos que permite buscar subdominios a través de motores de búsqueda y redes de entrega de contenido (CDN).

Aplicación del Comando Gau

Se empleó el comando `gau --threads 5 moonpay.com --o gauoutput.txt` para buscar subdominios de Moonpay.com utilizando Gau. La opción `--threads 5` indica el número de hilos utilizados para la búsqueda, y `--o gauoutput.txt` especifica el archivo de salida donde se guardan los resultados.

COMANDO

```
gau --threads 5 moonpay.com --o gauoutput.txt
```

Consolidación y Filtrado de Subdominios para Moonpay.com

Después de obtener datos de búsquedas usando CTFR, ataques de fuerza bruta, Gau y Katana, se implementó un proceso de consolidación y filtrado para crear una lista precisa y exclusiva de subdominios.

Los archivos, como ctfrdominios.txt, fuerzabruta.txt, gaudominios.txt y katanadominios.txt, se combinaron con:

COMANDO

```
cat ctfrdominios.txt fuerzabruta.txt gaudominios.txt katanadominios.txt | unfurl | sort |  
uniq > footprinting.txt
```

La lista final depurada se almacenó en footprinting.txt

```
(root@kali)-[/home/sebastian/recopilacion/moonpay/ficherosanteriores]  
# cat ctfrdominios.txt fuerzabruta.txt gaudominios.txt katanadominios.txt  
| unfurl | sort | uniq > footprinting.txt
```

Ahora tomamos la lista de subdominios footprinting.txt y filtramos solo los subdominios que pertenecen al dominio "moonpay.com", los convertimos a minúsculas y luego extraemos los subdominios únicos, guardando el resultado en el archivo footprinting_scope.txt.

```
cat footprinting.txt | grep -E '\.moonpay\.com$' | tr '[:upper:]' '[:lower:]' | unfurl --  
unique domains > footprinting_scope.txt
```

Tenemos nuestra lista de objetivos lista para el fingerprinting

Herramienta GoWitnесс

GoWitness es una herramienta poderosa que permite capturar imágenes de sitios web, incluyendo subdominios, proporcionando una visión visual de la infraestructura digital. En este proceso de reconocimiento vertical de Moonpay.com, GoWitness fue utilizado para capturar los subdominios obtenidos y analizar su tecnología subyacente.

Captura de Subdominios con GoWitness

Usando el archivo footprinting_scope.txt, que contenía los subdominios consolidados y depurados, se ejecutó el siguiente comando:

COMANDO

```
gowitness file -f footprinting_scope.txt > gowitnesscaptures.txt
```

Este comando capturó imágenes de los subdominios listados en footprinting.txt y guardó los resultados en gowitnesscaptures.txt. Entre estas capturas, se identificaron imágenes de la API, proporcionando una visión detallada de su funcionalidad.


```
(root@kali) - [ /home/.../recopilacion/moonpay/ficherosanteriores/screenshots ]
# ls
http-account.moonpay.com.png      http-qr.moonpay.com.png           https-memberships.moonpay.com.png
http-api.moonpay.com.png           https-access.moonpay.com.png      https-moonpay.com.png
http-auth.moonpay.com.png          https-account.moonpay.com.png     https-nft.moonpay.com.png
http-buy.moonpay.com.png           https-api.moonpay.com.png         https-page.moonpay.com.png
http-buy-staging.moonpay.com.png   https-auth.moonpay.com.png        https-qr.moonpay.com.png
http-changelog.moonpay.com.png     https-buy.moonpay.com.png         https-security.moonpay.com.png
http-clicks.moonpay.com.png        https-buy-staging.moonpay.com.png https-sell.moonpay.com.png
http-concierge.moonpay.com.png     https-changelog.moonpay.com.png   https-static.moonpay.com.png
http-consumer-api.moonpay.com.png  https-clicks.moonpay.com.png      https-status.moonpay.com.png
http-dashboard.moonpay.com.png     https-concierge.moonpay.com.png   https-support.moonpay.com.png
http-dev.moonpay.com.png           https-consumer-api.moonpay.com.png https-static.moonpay.com.png
http-docs.moonpay.com.png          https-dashboard.moonpay.com.png   http-status.moonpay.com.png
http-fp.moonpay.com.png            https-dev.moonpay.com.png        http-support.moonpay.com.png
http-go.moonpay.com.png            https-docs.moonpay.com.png       https-web3.moonpay.com.png
http-memberships.moonpay.com.png   http-security.moonpay.com.png     https-www.moonpay.com.png
http-moonpay.com.png              http-sell.moonpay.com.png        http-web3.moonpay.com.png
http-nft.moonpay.com.png           https-fp.moonpay.com.png          http-www.moonpay.com.png
http-page.moonpay.com.png          https-go.moonpay.com.png
```

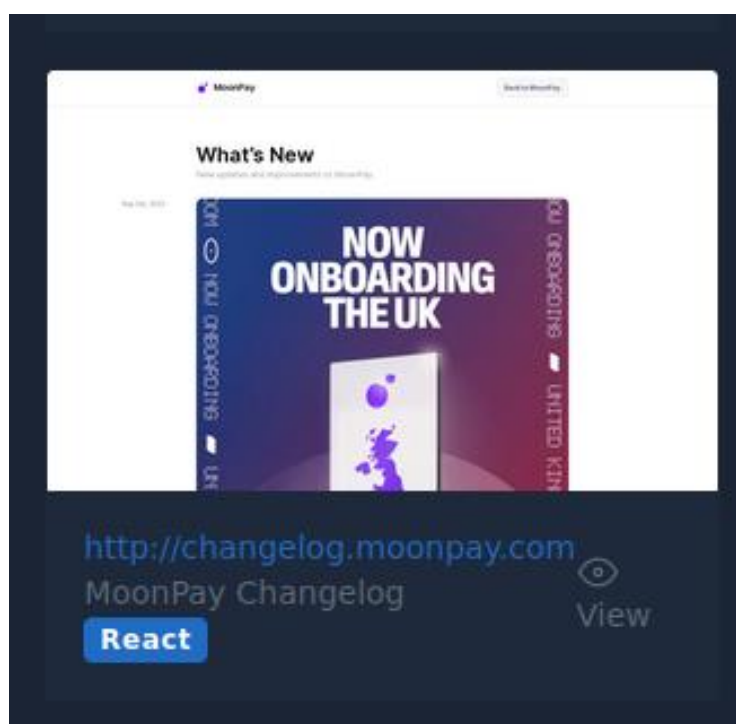
Análisis y Exploración con GoWitness

Para un análisis más profundo, se utilizó el servicio de servidor web de GoWitness.

COMANDO

```
gowitness report serve
```

Este comando habilitó una interfaz gráfica accesible a través de la URL localhost:7171 en el navegador. Aquí, se encontró información valiosa sobre la infraestructura de Moonpay.com. Por ejemplo, se descubrió que el sitio está programado con tecnología React, lo que señala la posibilidad de buscar vulnerabilidades específicas en esta tecnología.



Además, se obtuvo una variedad de información adicional para investigaciones futuras.

The screenshot displays the HTTPX tool interface for the URL <https://security.moonpay.com>. The interface is divided into several sections:

- URL Details:** Shows the URL and a 'Visit URL' button.
- TLS Information:** A table showing the TLS certificate details.
- Response Headers:** A table showing the HTTP response headers.

SUBJECT CN	ISSUER CN	SIG ALG
security.moonpay.com	GTS CA 1P5	SHA
GTS CA 1P5	GTS Root R1	SHA RSA
GTS Root R1	GlobalSign Root CA	SHA RSA

KEY	VALUE
Date	Mon, 02 Oct 2023 22:47:41 GMT
Content-Type	text/html; charset=utf-8
Link	<https://fonts.googleapis.com>; rel="preconnect"
Set-Cookie	slugid=7rxc0p9napwvxluxe; SameSite=Strict; Secure; domainid=62791daaa41825ecf79c07c0; SameSite=Strict; Secure;
Cache-Control	public, max-age=0, must-revalidate
Strict-Transport-Security	max-age=31536000; includeSubDomains
Vary	Accept-Encoding

FINGERPRINTING

Informe de Análisis de Dominios Activos con HTTPX

En este análisis, se evaluaron los dominios obtenidos mediante el proceso de footprinting para determinar cuáles de ellos están activos y servidores web. Se utilizó la herramienta especializada HTTPX, que permite el fingerprinting de direcciones HTTP y determinar su estado y configuración.

Se descargó y utilizó la herramienta HTTPX del repositorio de GitHub de Project Discovery para analizar los subdominios recopilados durante el proceso de footprinting.

Identificación de Dominios Activos:

Se empleó el siguiente comando para verificar la actividad de los dominios:

COMANDO

```
cat footprinting_scope.txt | httpx -silent -mc 200,401,403 -o httpxvivos.txt
```

```
(root@kali)-[/home/sebastian/recopilacion/moonpay/ficherosanteriores]
# cat httpxvivos.txt
https://dashboard.moonpay.com
https://page.moonpay.com
https://buy.moonpay.com
https://sell.moonpay.com
https://concierge.moonpay.com
https://account.moonpay.com
https://www.MoonPay.com
https://web3.moonpay.com
https://www.moonpay.com
https://fp.moonpay.com
https://memberships.moonpay.com
https://static.moonpay.com
https://access.moonpay.com
https://changelog.moonpay.com
https://security.moonpay.com
https://status.moonpay.com
```

Este comando realizó solicitudes HTTP a los subdominios y guardó los que respondieron con los códigos de estado 200 (OK), 401 (No Autorizado) o 403 (Prohibido) en el archivo footvivos.txt. Estos códigos indican la actividad y accesibilidad de los subdominios.

Tomamos la lista de subdominios vivos, extraemos los subdominios únicos de esa lista, manipulamos la url con unfurl para quedarnos con lo importante y guardamos los subdominios únicos en el archivo subdominios_targets.txt

COMANDO

```
cat httpxvivos.txt | unfurl --unique domains > subdominios_targets.txt
```

```
(root@kali)-[/home/.../recopilacion/entrega/fingerprinting/httpx]
# cat subdominios_targets.txt
dashboard.moonpay.com
page.moonpay.com
buy.moonpay.com
sell.moonpay.com
concierge.moonpay.com
account.moonpay.com
www.MoonPay.com
web3.moonpay.com
www.moonpay.com
fp.moonpay.com
memberships.moonpay.com
static.moonpay.com
access.moonpay.com
changelog.moonpay.com
security.moonpay.com
status.moonpay.com
```

Escaneo Adicional con HTTPX

Se realizó otro escaneo para obtener información más detallada sobre los subdominios activos. Este comando extrajo los códigos de estado, títulos de las páginas web y determinó si los dominios estaban siendo servidos a través de un CDN (Content Delivery Network):

COMANDO

```
cat footprinting_scope.txt | httpx --status-code --title --cdn > httpxstatus.txt
```

```
(root@kali)-[/home/sebastian/recopilacion/moonpay/ficherosanteriores]
# cat httpxstatus.txt
https://buy-staging.moonpay.com [302] []
https://concierge.moonpay.com [403] [Attention Required! | Cloudflare]
https://buy.moonpay.com [200] [MoonPay]
https://dashboard.moonpay.com [200] [MoonPay Dashboard]
https://page.moonpay.com [403] []
https://account.moonpay.com [200] [MoonPay Account]
https://go.moonpay.com [404] [404: This page could not be found]
https://moonpay.com [301] []
https://memberships.moonpay.com [401] [Authentication Required]
https://sell.moonpay.com [200] [MoonPay]
https://api.moonpay.com [404] [Error]
https://web3.moonpay.com [200] [secure-wallet]
https://www.moonpay.com [200] [MoonPay: Buy and sell Bitcoin, Ethereum, and
other cryptos]
https://qr.moonpay.com [301] []
https://www.MoonPay.com [200] [MoonPay: Buy and sell Bitcoin, Ethereum, and
other cryptos]
https://consumer-api.moonpay.com [404] []
https://fp.moonpay.com [200] []
https://support.moonpay.com [302] []
https://static.moonpay.com [403] []
https://access.moonpay.com [200] []
https://nft.moonpay.com [404] []
https://docs.moonpay.com [302] []
https://status.moonpay.com [200] [MoonPay Status]
https://dev.moonpay.com [302] []
https://clicks.moonpay.com [404] [404 Not Found]
```

Además, se verificó la existencia de la ruta /admin en los subdominios, proporcionando información adicional para futuros análisis

COMANDO

```
cat footprinting_scope.txt | httpx --path=admin --status-code > httpxadmin.txt
```

```
(root@kali) - /home/sebastian/recopilacion/moonpay/ficherosanteriores
# cat httpxadmin.txt
https://buy-staging.moonpay.com/admin [302]
https://dashboard.moonpay.com/admin [403]
https://buy.moonpay.com/admin [200]
https://account.moonpay.com/admin [404]
https://concierge.moonpay.com/admin [404]
https://memberships.moonpay.com/admin [401]
https://moonpay.com/admin [301]
https://go.moonpay.com/admin [404]
https://api.moonpay.com/admin [404]
https://web3.moonpay.com/admin [200]
https://sell.moonpay.com/admin [200]
https://consumer-api.moonpay.com/admin [404]
https://www.moonpay.com/admin [301]
https://qr.moonpay.com/admin [301]
https://www.MoonPay.com/admin [301]
https://fp.moonpay.com/admin [404]
https://static.moonpay.com/admin [404]
https://nft.moonpay.com/admin [404]
https://docs.moonpay.com/admin [302]
https://auth.moonpay.com/admin [404]
https://access.moonpay.com/admin [404]
https://page.moonpay.com/admin [403]
https://changelog.moonpay.com/admin [302]
https://status.moonpay.com/admin [302]
https://clicks.moonpay.com/admin [404]
https://support.moonpay.com/admin [302]
https://security.moonpay.com/admin [200]
https://dev.moonpay.com/admin [302]
```

Conclusion

Los análisis con HTTPX revelaron una lista de subdominios activos, identificando códigos de estado 200, títulos de páginas web y la presencia de la ruta /admin en algunos subdominios.

Nmap Informe de Escaneo de Puertos y Métodos HTTP

En esta fase del análisis de seguridad, se llevaron a cabo escaneos de puertos y métodos HTTP en los subdominios identificados previamente para comprender mejor la estructura de la red y las operaciones permitidas en los servidores web asociados con estos subdominios.

Escaneo inicial de los hosts para verificar su disponibilidad utilizando Nmap

COMANDO

```
nmap -sn -iL footprinting_scope.txt > nmapoutput.txt
```

Este escaneo confirmó que todos los hosts estaban activos y respondían a las solicitudes.

```
(root@kali)-[/home/.../recopilacion/entrega/fingerprinting/nmap]
# cat nmapoutput.txt
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-15 02:36 EDT
Nmap scan report for access.moonpay.com (208.127.231.159)
Host is up (0.18s latency).
Nmap scan report for account.moonpay.com (172.64.144.65)
Host is up (0.028s latency).
Other addresses for account.moonpay.com (not scanned): 104.18.43.191 2606:4700:4400::ac40:9041 2606:4700:4400::6812:2bbf
Nmap scan report for api.moonpay.com (104.18.43.191)
Host is up (0.034s latency).
Other addresses for api.moonpay.com (not scanned): 172.64.144.65 2606:4700:4400::6812:2bbf 2606:4700:4400::ac40:9041
Nmap scan report for page.moonpay.com (104.18.42.158)
Host is up (0.034s latency).
Other addresses for page.moonpay.com (not scanned): 172.64.145.98 2606:4700:4400::6812:2a9e 2606:4700:4400::ac40:9162
Nmap scan report for security.moonpay.com (151.101.218.204)
Host is up (0.034s latency).
Nmap scan report for status.moonpay.com (3.160.119.46)
Host is up (0.034s latency).
Other addresses for status.moonpay.com (not scanned): 3.160.119.11 3.160.119.119 3.160.119.128
rDNS record for 3.160.119.46: server-3-160-119-46.eze50.r.cloudfront.net
Nmap scan report for support.moonpay.com (104.16.53.111)
Host is up (0.028s latency).
Other addresses for support.moonpay.com (not scanned): 104.16.51.111
Nmap scan report for auth.moonpay.com (104.18.43.191)
Host is up (0.027s latency).
Other addresses for auth.moonpay.com (not scanned): 172.64.144.65 2606:4700
```

Extracción de las direcciones IP de los subdominios

COMANDO

cat nmapoutput.txt | grep -oE '[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+' > nmapoutput_ips.txt

```
(root@kali)-[/home/.../recopilacion/entrega/fingerprinting/nmap]
# cat nmapoutput_ips.txt
208.127.231.159
172.64.144.65
104.18.43.191
104.18.43.191
172.64.144.65
104.18.42.158
172.64.145.98
151.101.218.204
3.160.119.46
3.160.119.11
3.160.119.119
3.160.119.128
3.160.119.46
104.16.53.111
104.16.51.111
104.18.43.191
172.64.144.65
172.64.144.65
104.18.43.191
104.18.42.158
172.64.145.98
104.18.43.191
172.64.144.65
```

Se realizó un escaneo de puertos en los subdominios activos

COMANDO

`nmap -Pn -F -iL subdominios_targets.txt > nmappuertos.txt`

```
(root@kali)-[/home/.../recopilacion/entrega/fingerprinting/nmap]
# cat nmappuertos.txt
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-15 02:50 EDT
Nmap scan report for dashboard.moonpay.com (104.18.43.191)
Host is up (0.031s latency).
Other addresses for dashboard.moonpay.com (not scanned): 172.64.144.65 260
6:4700:4400::6812:2bbf 2606:4700:4400::ac40:9041
Not shown: 96 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
8080/tcp   open  http-proxy
8443/tcp   open  https-alt

Nmap scan report for page.moonpay.com (172.64.145.98)
Host is up (0.037s latency).
Other addresses for page.moonpay.com (not scanned): 104.18.42.158 2606:470
0:4400::ac40:9162 2606:4700:4400::6812:2a9e
Not shown: 96 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
8080/tcp   open  http-proxy
8443/tcp   open  https-alt

Nmap scan report for web3.moonpay.com (172.64.144.65)
Host is up (0.035s latency).
Other addresses for web3.moonpay.com (not scanned): 104.18.43.191 2606:470
0:4400::6812:2bbf 2606:4700:4400::ac40:9041
```

Este escaneo reveló que, además de los puertos estándar 80 y 443, también estaban abiertos los puertos 8080 (HTTP Proxy) y 8443 (HTTPS alternativo).

Escaneo de Métodos HTTP

Se procedió a analizar los métodos HTTP permitidos en los servidores web que operan en los puertos 80, 443, 8080 y 8443 mediante el script `http-methods` de Nmap:

COMANDO

`nmap -p80,443,8080,8443 --script http-methods -iL subdominios_targets.txt > nmapoutput_httpmethods.txt`

```
(root@kali)-[/home/.../recopilacion/entrega/fingerprinting/nmap]
# cat nmapoutput_httpmethods.txt
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-15 03:21 EDT
Nmap scan report for dashboard.moonpay.com (104.18.43.191)
Host is up (0.034s latency).
Other addresses for dashboard.moonpay.com (not scanned): 172.64.144.65 260
6:4700:4400::ac40:9041 2606:4700:4400::6812:2bbf

PORT      STATE SERVICE
80/tcp    open  http
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
443/tcp   open  https
| http-methods:
|_ Supported Methods: GET HEAD OPTIONS

Nmap scan report for page.moonpay.com (172.64.145.98)
Host is up (0.030s latency).
Other addresses for page.moonpay.com (not scanned): 104.18.42.158 2606:470
0:4400::6812:2a9e 2606:4700:4400::ac40:9162

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

Los resultados indicaron que los puertos 80 y 443 permiten los métodos GET, HEAD, POST y OPTIONS. Sin embargo, los puertos 8080 y 8443 no respondieron, lo que sugiere posibles restricciones, configuraciones de red específicas o errores en la configuración del servidor web en estos puertos.

Conclusión

El escaneo de puertos y métodos HTTP proporcionó una visión detallada de los servicios en ejecución en los subdominios identificados. Los puertos estándar 80 y 443 están activos y permiten operaciones HTTP comunes. Sin embargo, los puertos 8080 y 8443 no respondieron, lo que puede deberse a configuraciones específicas del servidor o restricciones en la red.

Escaneo de Puertos con Masscan

Se a cabo un escaneo exhaustivo de los puertos en las direcciones IP de los subdominios identificados previamente. Antes de ejecutar Masscan, se obtuvieron las direcciones IP asociadas con los subdominios y se filtraron para eliminar duplicados.

Obtención de Direcciones IP

Primero, se realizó un bucle FOR para obtener las direcciones IP de los subdominios utilizando el comando proporcionado por ChatGPT

COMANDO

```
for subdomain in $(cat subdominios_targets.txt); do dig +short $subdomain | grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}'; done > subdominiosip.txt
```



```
(root@kali)-[/home/.../recopilacion/entrega/fingerprinting/masscan]
# cat subdominios_targets_ips.txt
104.18.43.191
172.64.144.65
172.64.145.98
104.18.42.158
104.18.43.191
172.64.144.65
104.18.43.191
172.64.144.65
104.18.43.191
172.64.144.65
172.64.144.65
104.18.43.191
104.18.43.191
172.64.144.65
104.18.43.191
172.64.144.65
172.64.144.65
104.18.43.191
104.18.43.191
172.64.144.65
104.18.43.191
172.64.144.65
208.127.231.159
172.64.144.65
104.18.43.191
151.101.218.204
3.160.119.128
```

Luego, se eliminaron las direcciones IP duplicadas del archivo generado:

COMANDO

cat subdominiosip.txt | sort | uniq > subdominios_targets_ipsok.txt

```
(root@kali)-[/home/.../recopilacion/entrega/fingerprinting/masscan]
# cat subdominios_targets_ipsok.txt
104.18.42.158
104.18.43.191
151.101.218.204
172.64.144.65
172.64.145.98
208.127.231.159
3.160.119.11
3.160.119.119
3.160.119.128
3.160.119.46
```

Ejecución de Masscan

Se procedió a ejecutar Masscan en las direcciones IP obtenidas, escaneando los puertos 21, 22, 23, 25, 53, 80, 110, 111, 135, 139, 143, 443, 445, 993, 995, 1723, 3306, 3389, 5900, 8080 y 8443 con una tasa de exploración de 1000 paquetes por segundo:

COMANDO

masscan -
p21,22,23,25,53,80,110,111,135,139,143,443,445,993,995,1723,3306,3389,5900,8080,8443 --rate 1000 -iL subdominios_targets_ipsok.txt > masscan_output.txt

```
(root@kali)-[/home/./recopilacion/entrega/fingerprinting/masscan]
# cat masscan_output.txt
Discovered open port 80/tcp on 3.160.119.46

Discovered open port 80/tcp on 151.101.218.204
Discovered open port 443/tcp on 3.160.119.128
Discovered open port 443/tcp on 3.160.119.11
Discovered open port 80/tcp on 3.160.119.128
Discovered open port 8443/tcp on 104.18.42.158
Discovered open port 443/tcp on 172.64.144.65
Discovered open port 8443/tcp on 172.64.145.98
Discovered open port 443/tcp on 172.64.145.98
Discovered open port 8443/tcp on 172.64.144.65
Discovered open port 443/tcp on 104.18.42.158
Discovered open port 443/tcp on 104.18.43.191
Discovered open port 8080/tcp on 104.18.43.191
Discovered open port 8080/tcp on 104.18.42.158
```

Resultados del Escaneo:

Masscan generó un archivo llamado masscan_output.txt que contiene los resultados del escaneo de los puertos especificados en las direcciones IP de los subdominios.

Escaneo de Vulnerabilidades con Nuclei

En esta fase del análisis de seguridad, se empleó Nuclei, una herramienta especializada en escaneos de vulnerabilidades web. El objetivo fue identificar posibles fallos de seguridad en el dominio moonpay.com.

Escaneo Inicial con Nuclei

El escaneo inicial se realizó utilizando el siguiente comando

```
nuclei -u moonpay.com > nucleiescan.txt
```


Procedimiento:

Se ejecutó WAFW00F en el archivo de subdominios activos "subdominios_targets.txt" para analizar si la web está protegida por un WAF. El resultado se guardó en el archivo "wafwoof.txt" para su revisión.

Comando Utilizado:

wafw00f -i footdomainsvivos.txt > wafwoof.txt

```
(root@kali) ~/home/sebastian/recopilacion/moonpay
cat wafwoof.txt

{ WOOF! }
```



```
~ WAFW00F : v2.2.0 ~
The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://dashboard.moonpay.com
[*] The site https://dashboard.moonpay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
[*] Checking https://page.moonpay.com
[*] The site https://page.moonpay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
[*] Checking https://buy.moonpay.com
[*] The site https://buy.moonpay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
[*] Checking https://sell.moonpay.com
[*] The site https://sell.moonpay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
[*] Checking https://concierge.moonpay.com
[*] The site https://concierge.moonpay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
[*] Checking https://account.moonpay.com
[*] The site https://account.moonpay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
[*] Checking https://www.MoonPay.com
[*] The site https://www.MoonPay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
[*] Checking https://web3.moonpay.com
[*] The site https://web3.moonpay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
[*] Checking https://www.moonpay.com
[*] The site https://www.moonpay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
[*] Checking https://fp.moonpay.com
[*] The site https://fp.moonpay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
[*] Checking https://memberships.moonpay.com
[*] The site https://memberships.moonpay.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
```

Resultados del Análisis:

El análisis reveló que la web moonpay.com está protegida por el Firewall de Aplicaciones Web de Cloudflare. Cloudflare es un proveedor popular de servicios de seguridad web y optimización de rendimiento, lo que indica un nivel de protección adicional en la aplicación web.

Ataque de Fuzzing con FFUF

Se empleó FFUF, una herramienta de fuzzing que permite realizar ataques de fuerza bruta para descubrir rutas y recursos en un servidor web. El objetivo fue identificar posibles puntos de entrada y rutas específicas en el dominio moonpay.com utilizando un script específico.

Especificaciones del Ataque

Se utilizó el script "common.txt" del repositorio GIT de Daniel Miessler, que contiene rutas y recursos comunes utilizados en sitios web. El ataque se realizó con 20 hilos para evitar desencadenar alarmas de WAF y se configuró para registrar los servidores que respondieran con códigos de estado HTTP 200, 401 y 403.

Comando Utilizado:

```
ffuf -w /home/sebastian/recopilacion/danielmiessler/SecLists/Discovery/Web-Content/common.txt -t 20 -mc 200,401,403 -u https://moonpay.com/FUZZ > fuzzingatack.txt
```

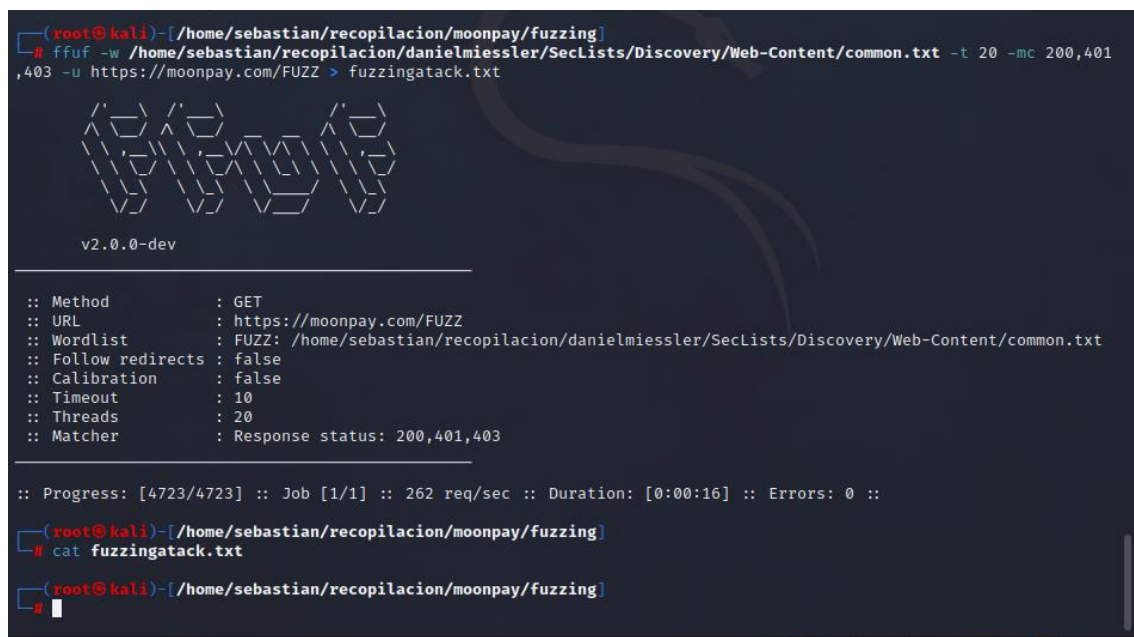
Resultados del Ataque

FFUF

Es una herramienta de Fuzzing útil para mostrar respuestas del servidor de distintas rutas y recursos especificados en un script a través de un ataque de fuerza bruta. Lo vamos a poner en 20 threads (20 hilos) para que no salten las alarmas de la WAF Cloudflare y que nos registre los servidores que respondan con código de estado HTTP 200, 401 y 403. El script que utilizaremos se encuentra en el directorio que descargamos del repositorio GIT de Daniel Miessler, será uno general dentro de la carpeta Web Content llamado common.txt

Comandamos:

```
ffuf -w /home/sebastian/recopilacion/danielmiessler/SecLists/Discovery/Web-Content/common.txt -t 20 -mc 200,401,403 -u https://moonpay.com/FUZZ > fuzzingatack.txt
```



```
(root@kali)~/home/sebastian/recopilacion/moonpay/fuzzing[
# ffuf -w /home/sebastian/recopilacion/danielmiessler/SecLists/Discovery/Web-Content/common.txt -t 20 -mc 200,401,403 -u https://moonpay.com/FUZZ > fuzzingatack.txt

v2.0.0-dev

:: Method      : GET
:: URL         : https://moonpay.com/FUZZ
:: Wordlist     : FUZZ: /home/sebastian/recopilacion/danielmiessler/SecLists/Discovery/Web-Content/common.txt
:: Follow redirects : false
:: Calibration  : false
:: Timeout     : 10
:: Threads     : 20
:: Matcher     : Response status: 200,401,403

:: Progress: [4723/4723] :: Job [1/1] :: 262 req/sec :: Duration: [0:00:16] :: Errors: 0 ::

(root@kali)~/home/sebastian/recopilacion/moonpay/fuzzing[
# cat fuzzingatack.txt

(root@kali)~/home/sebastian/recopilacion/moonpay/fuzzing[
#
```

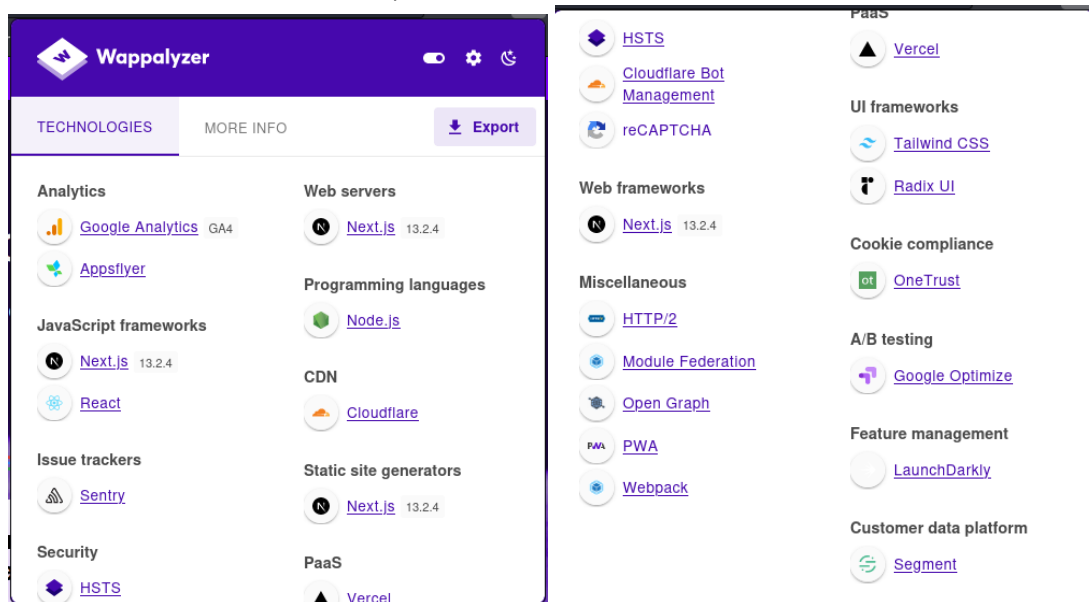
Nuestro ataque fuzzing no ha dado resultados, debemos continuar recopilando con otras técnicas.

Lamentablemente, el ataque de fuerza bruta con FFUF no arrojó resultados positivos. No se encontraron rutas o recursos adicionales en el dominio moonpay.com utilizando el script proporcionado.

Wappalyzer

Identifica tecnologías y plataformas específicas utilizadas en el sitio web, como frameworks y CMS.

Es una extensión de browsers que muestra las tecnologías sobre las que esta creada la web, esta extensión nos mostrara que tecnologías tiene moonpay.com. guardo prints para tener la informacion.



Podemos observar sus tecnologías que concuerdan con los datos de Gowitness

Análisis de Seguridad TLS/SSL con TestSSL

El análisis de seguridad TLS/SSL utilizando la herramienta TestSSL tiene como objetivo evaluar la robustez y la configuración de seguridad de los protocolos SSL/TLS en el servidor web de Moonpay.com.

Herramienta TestSSL

TestSSL es una herramienta de código abierto diseñada para realizar pruebas exhaustivas de seguridad en servidores y servicios HTTPS/TLS. Proporciona una visión detallada de la configuración de seguridad del servidor, identificando posibles vulnerabilidades y debilidades en la implementación de SSL/TLS.

Instalación de TestSSL

COMANDO

```
testss
```

Se ejecutó el análisis en el servidor Moonpay.com

COMANDO

```
testssl moonpay.com
```

Resultados del Análisis

Moonpay.com ofrece cifrados fuertes, como TLS_AES_256_GCM_SHA384 y TLS_AES_128_GCM_SHA256, que son altamente seguros y garantizan la confidencialidad de los datos transmitidos

```
Testing robust (perfect) forward secrecy, (P)FS -- omitting Null Authentication/Encryption, 3DES, RC4
```

```
PFS is offered (OK)
```

```
TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256
ECDHE-RSA-AES256-GCM-SHA384
DHE-RSA-AES256-GCM-SHA384
ECDHE-RSA-CHACHA20-POLY1305
TLS_AES_128_GCM_SHA256
ECDHE-RSA-AES128-GCM-SHA256
```

```
Elliptic curves offered:
```

```
prime256v1 secp384r1 secp521r1 X25519
```

```
DH group offered:
```

```
Unknown DH group (2048 bits)
```

En base a los resultados obtenidos, podemos concluir que Moonpay.com muestra una configuración sólida de seguridad en sus protocolos SSL/TLS.

ANALISIS DE SERVIDOR DE CORREO

En el contexto de correo electrónico, uno de los tipos de spoofing más comunes es el "spoofing de correo electrónico". En este tipo de ataque, un atacante falsifica la dirección de correo electrónico del remitente para que parezca que el correo electrónico proviene de una fuente confiable medidas de seguridad relacionadas con el correo electrónico.

Utilizamos la herramienta spoofcheck para escanear la configuración de seguridad del servidor de correo de moonpay.com. Spoofcheck es una herramienta especializada diseñada para evaluar vulnerabilidades de spoofing en las configuraciones SPF, DKIM y DMARC de un dominio.

Instalación y Ejecución:

COMANDO INSTALACION

```
git clone https://github.com/a6avind/spoofcheck.git
```

Posteriormente, se ejecutó la herramienta con el dominio moonpay.com y se guardaron los resultados en el archivo spoofcheck.txt.

```
python spoofcheck.py moonpay.com > spoofcheck.txt
```

Conclusion

El análisis reveló que la configuración SPF y DMARC del dominio moonpay.com está correctamente implementada y no permite que otros servidores envíen correos electrónicos falsificados en nombre de la empresa. Esto indica una sólida protección contra el spoofing, asegurando que los correos electrónicos de moonpay.com sean auténticos y confiables para los destinatarios

```
(root@kali)-[/home/sebastian/recopilacion/moonpay/spoofcheck]
# cat spoofcheck.txt
[*] Found SPF record:
[*] v=spf1 include:_spf.google.com include:mail.zendesk.com include:1449487
4.spf07.hubspotemail.net include:_spf.intacct.com -all
[*] SPF record contains an All item: -all
[*] Found DMARC record:
[*] v=DMARC1; p=reject; rua=mailto:5e4556e2ce7c1@ag.dmarcly.com; ruf=mailto:
:5e4556e2ce7c1@fo.dmarcly.com; sp=reject; pct=100; fo=0;
[-] DMARC policy set to reject
[*] Aggregate reports will be sent: mailto:5e4556e2ce7c1@ag.dmarcly.com
[*] Forensics reports will be sent: mailto:5e4556e2ce7c1@fo.dmarcly.com
[-] Spoofing not possible for moonpay.com
```


Subdomain Takeover

La subdomain takeover es una vulnerabilidad de seguridad que ocurre cuando un subdominio que originalmente estaba apuntando a un servicio o recurso específico ya no está en uso, pero no se ha desvinculado correctamente del proveedor de servicios en la nube o del servidor de alojamiento.

Para identificar posibles subdominios vulnerables, utilizamos la herramienta Subzy. Subzy es una herramienta de código abierto que automatiza la detección de subdominios que podrían estar en riesgo de toma de subdominios.

Primero, instalamos Subzy con el siguiente comando:
COMANDO

```
go install -v github.com/LukaSikic/subzy@latest
```

Luego, creamos un enlace simbólico para que Subzy sea accesible desde cualquier ubicación:

COMANDO

```
sudo ln -s $HOME/go/bin/subzy /usr/bin/subzy
```

Ejecutamos Subzy para analizar los subdominios
COMANDO

```
subzy run --targets subdominios_targets.txt
```

Resultado del Análisis:

```
(root@kali)-[/home/sebastian/recopilacion/moonpay]
# subzy run --targets footdomainsvivos.txt
[ * ] Loaded 16 targets
[ * ] Loaded 44 fingerprints
[ No ] HTTPS by default (--https)
[ 10 ] Concurrent requests (--concurrency)
[ No ] Check target only if SSL is valid (--verify_ssl)
[ 10 ] HTTP request timeout (in seconds) (--timeout)
[ No ] Show only potentially vulnerable subdomains (--hide_fails)
[ NOT VULNERABLE ] - dashboard.moonpay.com
[ NOT VULNERABLE ] - web3.moonpay.com
[ NOT VULNERABLE ] - buy.moonpay.com
[ NOT VULNERABLE ] - page.moonpay.com
[ NOT VULNERABLE ] - www.moonpay.com
[ NOT VULNERABLE ] - sell.moonpay.com
[ NOT VULNERABLE ] - concierge.moonpay.com
[ NOT VULNERABLE ] - www.MoonPay.com
[ HTTP ERROR ] - security.moonpay.com
[ NOT VULNERABLE ] - account.moonpay.com
[ NOT VULNERABLE ] - fp.moonpay.com
[ NOT VULNERABLE ] - status.moonpay.com
[ NOT VULNERABLE ] - memberships.moonpay.com
[ NOT VULNERABLE ] - static.moonpay.com
[ NOT VULNERABLE ] - changelog.moonpay.com
[ HTTP ERROR ] - access.moonpay.com
```

El análisis reveló que no hay subdominios vulnerables a subdomain takeover en la lista proporcionada.

WHATWEB

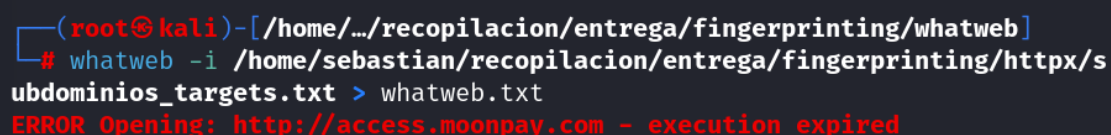
Es una herramienta de código abierto con utilidad para identificar tecnologías aplicadas en un sitio web. Comandamos:

Analiza el sitio web para identificar tecnologías y componentes específicos utilizados, proporcionando detalles sobre configuraciones y frameworks.

Se ejecutó WhatWeb en el dominio moonpay.com para identificar las tecnologías y componentes específicos utilizados en el sitio.

COMANDO

```
whatweb -i /home/sebastian/recopilacion/entrega/fingerprinting/httpx/subdominios_targets.txt  
> whatweb.txt
```



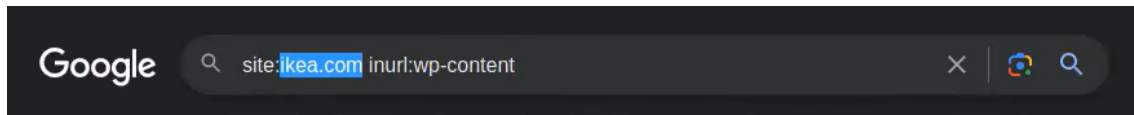
```
(root@kali)-[/home/.../recopilacion/entrega/fingerprinting/whatweb]  
# whatweb -i /home/sebastian/recopilacion/entrega/fingerprinting/httpx/s  
ubdominios_targets.txt > whatweb.txt  
ERROR Opening: http://access.moonpay.com - execution expired
```

Este error indica que WhatWeb encontró problemas al acceder a ese subdominio específico, y no pudo completar la operación de análisis debido a un tiempo de espera demasiado largo o a un problema de conectividad con ese subdominio en particular. Puede deberse a varias razones, como restricciones de red, configuraciones de firewall, problemas de DNS o el propio estado del servidor del subdominio.

Wpscan para escanear WordPress

Es una herramienta específica para escanear sitios web basados en WordPress que permite identificar posibles fallos y vulnerabilidades proporcionando información valiosa.

Antes de realizar el escaneo con WPScan, realizamos una búsqueda en Google utilizando la técnica "inurl:wp-content" para determinar si el sitio web objetivo (moonpay.com) está hecho con WordPress. No encontramos ningún resultado que indique que Moonpay.com utiliza WordPress, lo que sugiere que el sitio web no está construido con esta tecnología.



No hay evidencia de que Moonpay.com esté construido con WordPress, según los resultados de la búsqueda en Google. Esto significa que no es necesario realizar un escaneo con WPScan en este caso.

OSINT

Open Source Intelligence

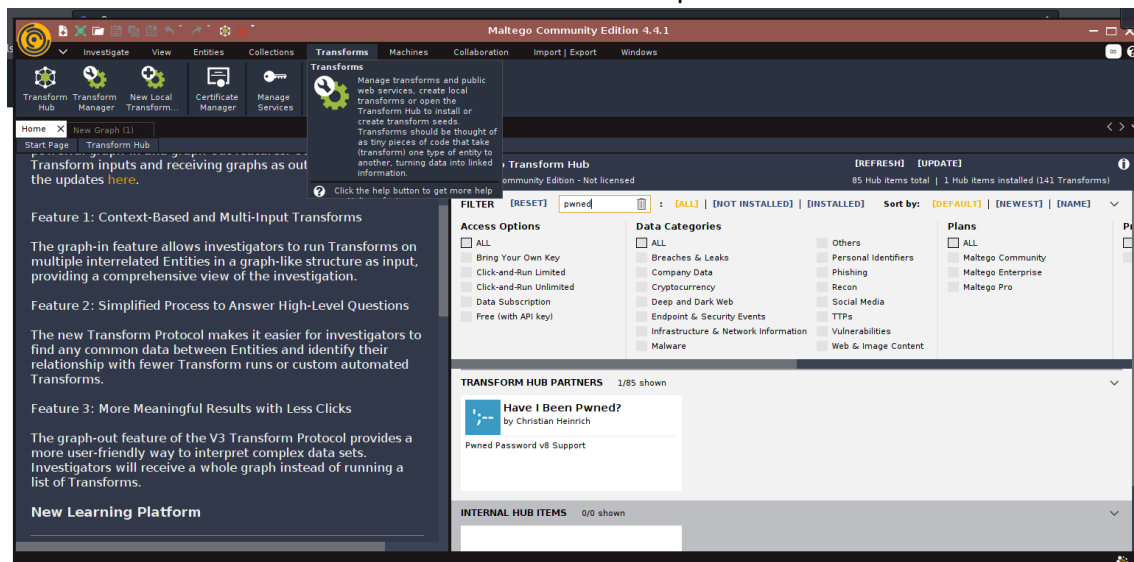
Maltego:

Utilizaremos Maltego para investigar si los correos electrónicos de empleados de moonpay.com han sido vulnerados.

Maltego es una herramienta de inteligencia y análisis de datos que se utiliza para recopilar y visualizar información de fuentes abiertas (OSINT) y llevar a cabo investigaciones de ciberseguridad

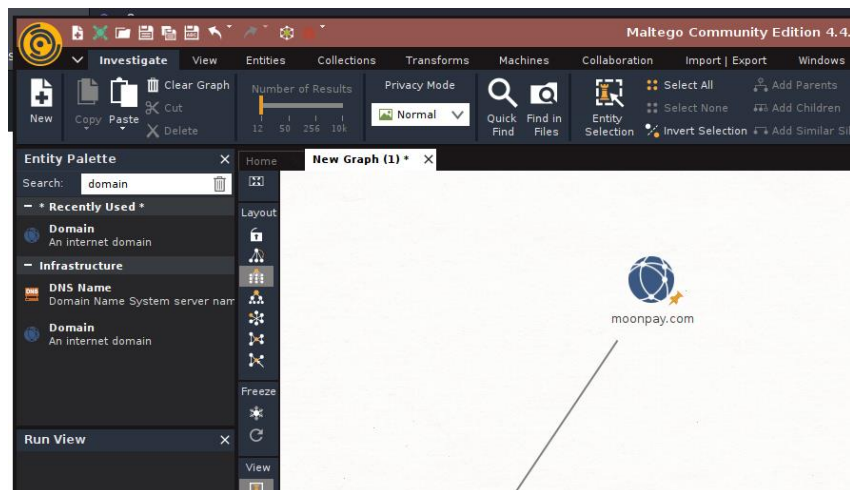
Instalación del Transformador Have I Been Pwned:

Se instaló el transformador de Have I Been Pwned en Maltego para verificar si las cuentas de correo electrónico están comprometidas.



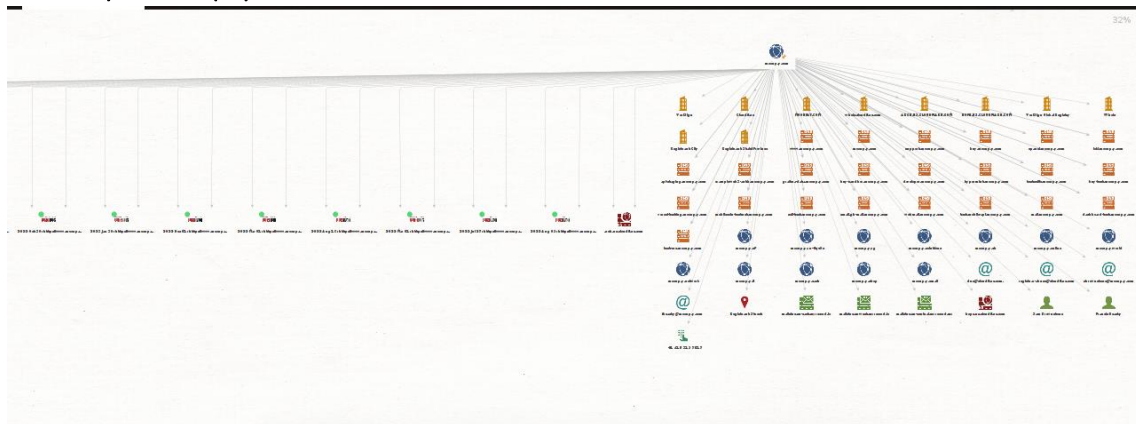
Búsqueda de Información sobre el Dominio moonpay.com:

Se agregó el dominio moonpay.com al gráfico de Maltego para recopilar información relacionada.



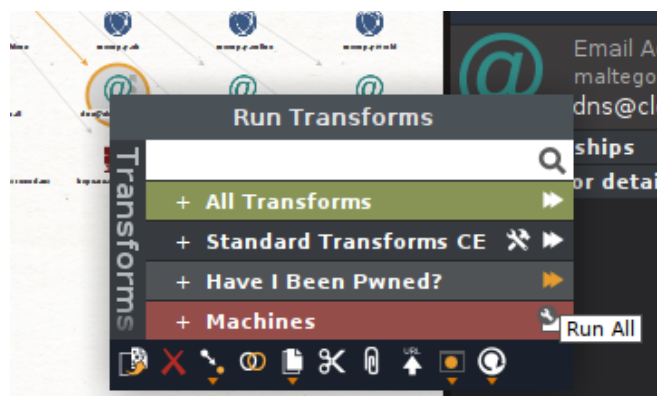
Análisis de Correos Electrónicos:

Se identificaron varios correos electrónicos asociados al dominio moonpay.com, incluyendo "registrar-abuse@cloudflare.com", "sbrownstone@moonpay.com", "dns@cloudflare.com" y "fbeasley@moonpay.com".



Verificación de Compromisos:

Se ejecutaron los transformadores de Have I Been Pwned en cada uno de los correos electrónicos para determinar si estas cuentas habían sido violadas o aparecían en listas de filtraciones.



Resultados:

registrar-abuse@cloudflare.com:

No se encontró en ninguna lista de filtraciones o violaciones.

sbrownstone@moonpay.com:

La cuenta no ha sido violada ni comprometida.

dns@cloudflare.com:

No se encontró en ninguna lista de filtraciones o violaciones.

fbeasley@moonpay.com:

La cuenta no ha sido violada ni comprometida.

```
Transform To Entities from WHOIS [IBM Watson] done (from entity "moonpay.com")
@haveibeenpwned is licensed under Creative Commons Attribution 4.0 International (from entity "moonpay.com")
Domain not breached (from entity "moonpay.com")
Transform Enrich breached domain [v3 @haveibeenpwned] returned with 0 entities (from entity "moonpay.com")
Transform Enrich breached domain [v3 @haveibeenpwned] done (from entity "moonpay.com")
Running transform Get all breaches of an e-mail address [v3 @haveibeenpwned] on 1 entities (from entity "dns@cloudflare.com.")
Running transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] on 1 entities (from entity "dns@cloudflare.com.")
@haveibeenpwned is licensed under Creative Commons Attribution 4.0 International (from entity "dns@cloudflare.com.")
Account not breached (from entity "dns@cloudflare.com.")
Transform Get all breaches of an e-mail address [v3 @haveibeenpwned] returned with 2 entities (from entity "dns@cloudflare.com.")
Transform Get all breaches of an e-mail address [v3 @haveibeenpwned] done (from entity "dns@cloudflare.com.")
@haveibeenpwned is licensed under Creative Commons Attribution 4.0 International (from entity "dns@cloudflare.com.")
Not listed in any pastes (from entity "dns@cloudflare.com.")
Transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] returned with 1 entities (from entity "dns@cloudflare.com.")
Transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] done (from entity "dns@cloudflare.com.")
Running transform Get all breaches of an e-mail address [v3 @haveibeenpwned] on 1 entities (from entity "registrar-abuse@cloudflare.com")
Running transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] on 1 entities (from entity "registrar-abuse@cloudflare.com")
@haveibeenpwned is licensed under Creative Commons Attribution 4.0 International (from entity "registrar-abuse@cloudflare.com")
16000591 breached accounts added to @haveibeenpwned for Eye4Fraud
Eye4Fraud added to @haveibeenpwned at 06 Mar 2023 04:46:58 (from entity "registrar-abuse@cloudflare.com")
Entities are "Weighted [View]" (from entity "registrar-abuse@cloudflare.com")
Transform Get all breaches of an e-mail address [v3 @haveibeenpwned] returned with 2 entities (from entity "registrar-abuse@cloudflare.com")
Transform Get all breaches of an e-mail address [v3 @haveibeenpwned] done (from entity "registrar-abuse@cloudflare.com")
@haveibeenpwned is licensed under Creative Commons Attribution 4.0 International (from entity "registrar-abuse@cloudflare.com")
Not listed in any pastes (from entity "registrar-abuse@cloudflare.com")
Transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] returned with 1 entities (from entity "registrar-abuse@cloudflare.com")
Transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] done (from entity "registrar-abuse@cloudflare.com")
Running transform Get all breaches of an e-mail address [v3 @haveibeenpwned] on 1 entities (from entity "sbrownstone@moonpay.com")
Running transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] on 1 entities (from entity "sbrownstone@moonpay.com")
@haveibeenpwned is licensed under Creative Commons Attribution 4.0 International (from entity "sbrownstone@moonpay.com")
Not listed in any pastes (from entity "sbrownstone@moonpay.com")
Transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] returned with 1 entities (from entity "sbrownstone@moonpay.com")
Transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] done (from entity "sbrownstone@moonpay.com")
@haveibeenpwned is licensed under Creative Commons Attribution 4.0 International (from entity "sbrownstone@moonpay.com")
Account not breached (from entity "sbrownstone@moonpay.com")
Transform Get all breaches of an e-mail address [v3 @haveibeenpwned] returned with 2 entities (from entity "sbrownstone@moonpay.com")
Transform Get all breaches of an e-mail address [v3 @haveibeenpwned] done (from entity "sbrownstone@moonpay.com")
Running transform Get all breaches of an e-mail address [v3 @haveibeenpwned] on 1 entities (from entity "fbeasley@moonpay.com")
Running transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] on 1 entities (from entity "fbeasley@moonpay.com")
@haveibeenpwned is licensed under Creative Commons Attribution 4.0 International (from entity "fbeasley@moonpay.com")
Not listed in any pastes (from entity "fbeasley@moonpay.com")
Transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] returned with 1 entities (from entity "fbeasley@moonpay.com")
Transform Get all pastes featuring the e-mail address [v3 @haveibeenpwned] done (from entity "fbeasley@moonpay.com")
@haveibeenpwned is licensed under Creative Commons Attribution 4.0 International (from entity "fbeasley@moonpay.com")
Account not breached (from entity "fbeasley@moonpay.com")
Transform Get all breaches of an e-mail address [v3 @haveibeenpwned] returned with 2 entities (from entity "fbeasley@moonpay.com")
Transform Get all breaches of an e-mail address [v3 @haveibeenpwned] done (from entity "fbeasley@moonpay.com")
```

Conclusiones:

Las cuentas de correo electrónico asociadas al dominio moonpay.com que fueron analizadas no muestran evidencia de compromisos o aparición en listas de filtraciones según los resultados de Have I Been Pwned en Maltego.

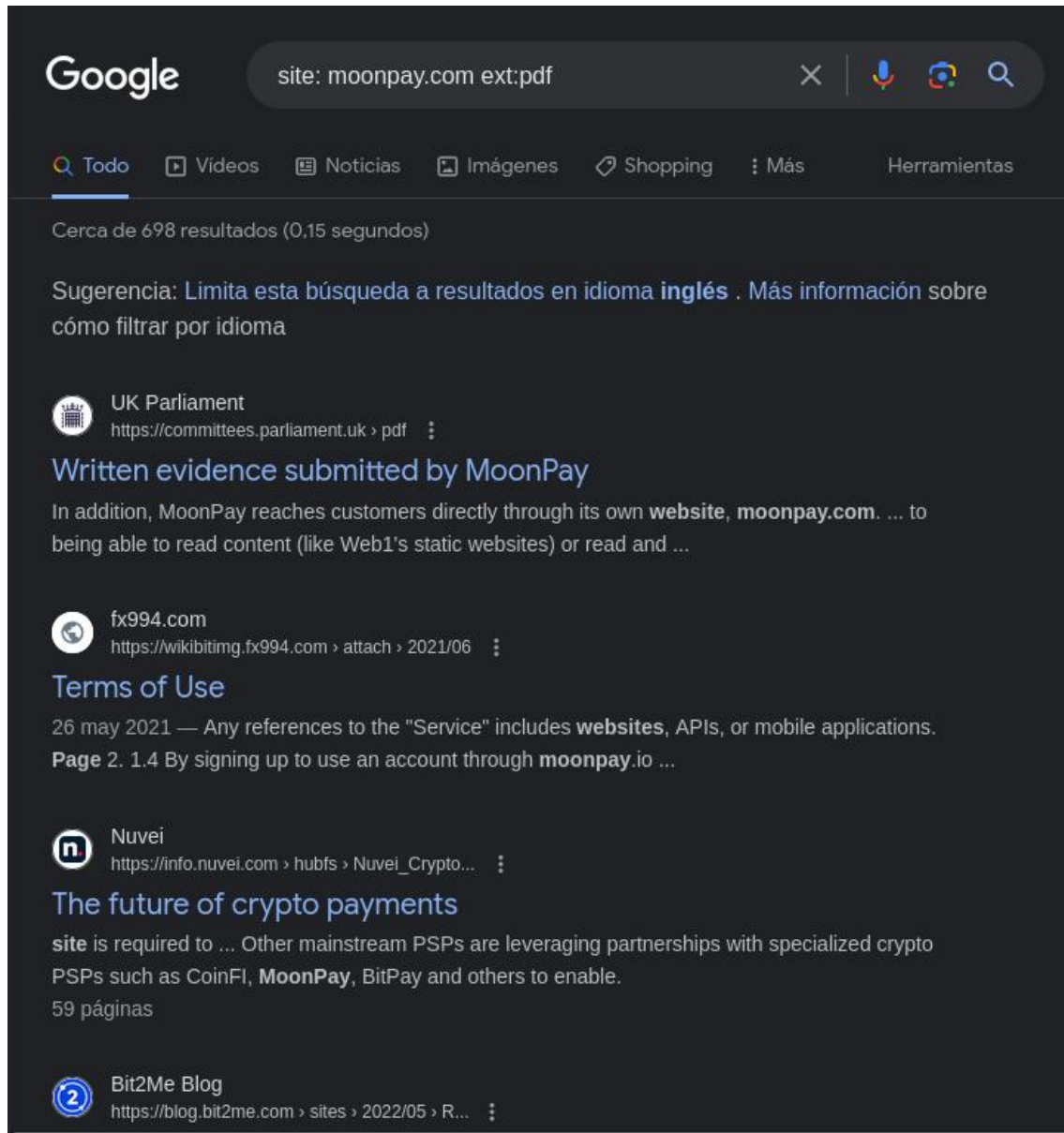
Motores de Búsqueda de Google (PDF Files):

Se realizaron búsquedas en motores de búsqueda, como Google y Bing, utilizando palabras clave específicas para encontrar posibles archivos sensibles expuestos relacionados con Moonpay y su dominio moonpay.com. El objetivo era identificar archivos PDF u otros documentos que podrían contener información sensible y que estuvieran expuestos públicamente en la web.

Búsqueda de Archivos PDF Relacionados con Moonpay:

Se realizaron búsquedas en Google y Bing utilizando palabras clave como "site:moonpay.com filetype:pdf" para encontrar archivos PDF específicos en el sitio web de Moonpay.

No se encontraron archivos PDF relacionados con Moonpay o moonpay.com en los resultados de búsqueda.



Dedigger.com Búsqueda en Google Drive:

Se utilizó la plataforma <https://www.dedigger.com/> para buscar enlaces compartidos de Google Drive que pudieran contener información delicada o sensible relacionada con Moonpay. El objetivo era identificar si había documentos o archivos expuestos públicamente en Google Drive que fueran accesibles a través de enlaces compartidos.

Se ejecutó una búsqueda utilizando palabras clave como "Moonpay", "moonpay.com", y términos relacionados para identificar posibles enlaces compartidos de Google Drive.

No se encontraron enlaces compartidos de Google Drive que condujeran a archivos sensibles o delicados.

Spiderfoot:

Se utilizó con el fin de recopilar información específica sobre correos electrónicos asociados a moonpay.com. Además obtuvimos datos e información extra que pueden tener relevancia en una evaluación de vulnerabilidades.

Iniciamos el servidor spiderfoot con el siguiente comando:

spiderfoot -l localhost:8082

```
(root@kali)-[/home/.../recopilacion/entrega/fingerprinting/whatweb]
# spiderfoot -l localhost:8082

*****
Use SpiderFoot by starting your web browser of choice and
browse to http://localhost:8082/
*****

2023-10-15 12:13:16,884 [INFO] sf : Starting web server at localhost:8082 .
..
2023-10-15 12:13:16,891 [WARNING] sf :
*****
Warning: passwd file contains no passwords. Authentication disabled.
Please consider adding authentication to protect this instance!
Refer to https://www.spiderfoot.net/documentation/#security.
*****

CherryPy Checker:
The use of 'localhost' as a socket host can cause problems on newer systems
, since 'localhost' can map to either an IPv4 or an IPv6 address. You should
d use '127.0.0.1' or '[::1]' instead.
```

Luego, se accedió a la interfaz de usuario mediante la URL

http://localhost:8082

Se ingresó el objetivo moonpay.com y se procedió a iniciar el análisis.

RESULTADOS

Encontramos emails internos de la empresa

Email Address	6	12	2023-10-15 16:03:31
Email Address - Generic	3	8	2023-10-15 16:03:31
Email Gateway (DNS MX Records)	4	4	2023-10-15 13:54:16

También números telefónicos y direcciones

Phone Number	3	19	2023-10-15 15:55:34
Physical Address	92	117	2023-10-15 15:45:24

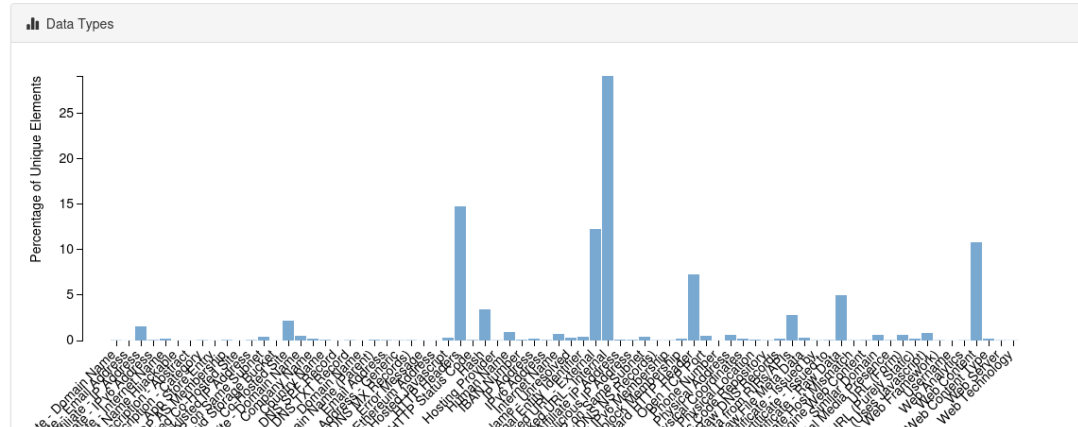
Debido a problemas con la exportación de los ficheros con la información recopilada dejo aquí pruebas y constancias de lo obtenido por medio del análisis de spiderfoot.

moonpay.com RUNNING

[Summary](#) [Correlations](#) [Browse](#) [Graph](#) [Scan Settings](#) [Log](#)

Scan Status					
Total	20805	Unique	8922	Status	RUNNING
				Errors	2454

Correlations			
High	0	Medium	0
Low	0	Info	0



[Check out our YouTube channel to see SpiderFoot HX in action.](#)

Type	Unique Data Elements	Total Data Elements	Last Data Element
Affiliate - Company Name	2	2	2023-10-15 16:42:01
Affiliate - Domain Name	3	8	2023-10-15 12:56:21
Affiliate - Email Address	33	120	2023-10-15 16:44:16
Affiliate - IP Address	152	152	2023-10-15 15:30:21
Affiliate - IPV6 Address	6	6	2023-10-15 12:56:21
Affiliate - Internet Name	65	81	2023-10-15 16:40:30
Affiliate - Internet Name Hijackable	1	2	2023-10-15 14:07:29
Affiliate - Web Content	7	12	2023-10-15 16:40:30
Affiliate Description - Abstract	5	5	2023-10-15 16:30:23
Affiliate Description - Category	39	39	2023-10-15 16:30:23
App Store Entry	1	1	2023-10-15 12:16:40
BGP AS Membership	7	80	2023-10-15 16:01:02
Blacklisted Co-Hosted Site	1	1	2023-10-15 15:02:57
Blacklisted IP Address	5	5	2023-10-15 15:18:01
Blacklisted IP on Same Subnet	66	66	2023-10-15 15:45:01
Cloud Storage Bucket	2	9	2023-10-15 16:16:58

Co-Hosted Site	207	3364	2023-10-15 16:32:46
Co-Hosted Site - Domain Name	155	181	2023-10-15 16:31:41
Company Name	33	4533	2023-10-15 16:43:59
Country Name	26	133	2023-10-15 16:36:54
DNS SPF Record	3	3	2023-10-15 16:16:51
DNS TXT Record	9	9	2023-10-15 16:16:51
Domain Name	1	107	2023-10-15 14:50:06
Domain Name (Parent)	8	8	2023-10-15 14:58:16
Email Address	6	15	2023-10-15 16:44:16
Email Address - Generic	3	12	2023-10-15 16:44:16
Email Gateway (DNS MX Records)	19	28	2023-10-15 16:26:36
Error Message	1	533	2023-10-15 16:43:59
Ethereum Address	5	22	2023-10-15 16:43:11
Externally Hosted Javascript	114	7016	2023-10-15 16:43:59
HTTP Headers	1402	1406	2023-10-15 16:43:56
HTTP Status Code	6	1411	2023-10-15 16:43:56
Hash	967	16065	2023-10-15 16:44:24
Historic Interesting File	1	1	2023-10-15 16:26:34
Hosting Provider	3	8	2023-10-15 15:03:24
Human Name	318	14735	2023-10-15 16:44:23

IBAN Number	10	81	2023-10-15 16:43:32
IP Address	18	100	2023-10-15 15:07:59
IPv6 Address	5	91	2023-10-15 15:06:56
Interesting File	1	1	2023-10-15 16:00:32
Internet Name	63	1561	2023-10-15 16:44:24
Internet Name - Unresolved	31	391	2023-10-15 16:44:24
Legal Entity Identifier	60	60	2023-10-15 15:45:23
Linked URL - External	1491	2438	2023-10-15 16:43:44
Linked URL - Internal	2853	3029	2023-10-15 16:43:54
Malicious Affiliate IP Address	13	13	2023-10-15 14:20:08
Malicious IP Address	7	7	2023-10-15 15:18:01
Malicious IP on Same Subnet	66	66	2023-10-15 15:45:01
Name Server (DNS NS Records)	32	32	2023-10-15 16:26:36
Netblock IPv6 Membership	3	11	2023-10-15 15:21:43
Netblock Membership	19	49	2023-10-15 15:25:59
Non-Standard HTTP Header	3437	7518	2023-10-15 16:44:05
Open TCP Port	44	45	2023-10-15 15:10:01
Phone Number	3	29	2023-10-15 16:31:41
Physical Address	92	117	2023-10-15 15:45:24
Physical Coordinates	28	32	2023-10-15 15:47:24

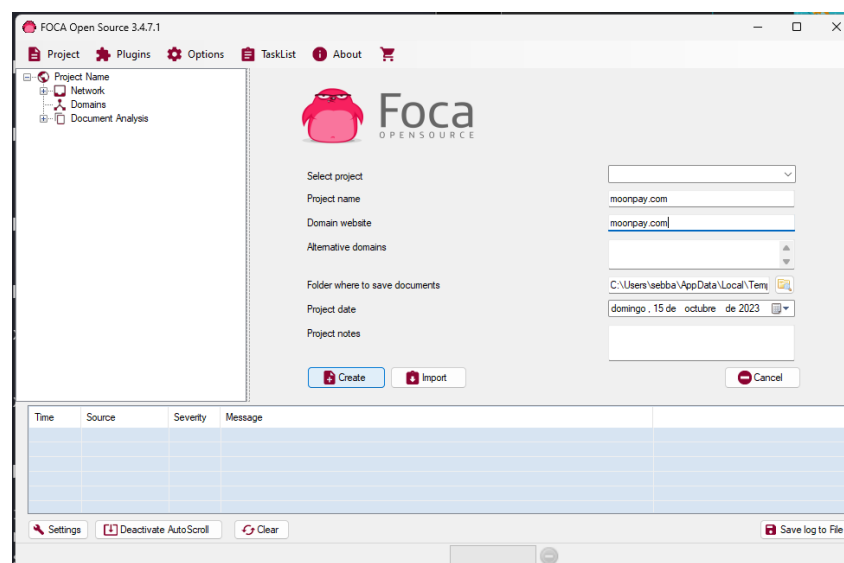
Physical Location	9	46	2023-10-15 16:01:02
Public Code Repository	10	14	2023-10-15 15:58:41
Raw DNS Records	38	38	2023-10-15 16:26:36
Raw Data from RIRs/APIs	291	308	2023-10-15 16:40:18
Raw File Meta Data	41	66	2023-10-15 16:41:17
SSL Certificate - Issued by	3	7	2023-10-15 14:54:28
SSL Certificate - Issued to	5	7	2023-10-15 14:54:28
SSL Certificate - Raw Data	466	806	2023-10-15 16:32:46
SSL Certificate Host Mismatch	3	4	2023-10-15 14:54:28
Search Engine Web Content	14	14	2023-10-15 16:30:23
Similar Domain	52	68	2023-10-15 12:27:57
<u>Social Media Presence</u>	12	12	2023-10-15 16:14:12
URL (Accepts Passwords)	1	1	2023-10-15 16:32:22
URL (Form)	818	818	2023-10-15 16:43:59
URL (Purely Static)	50	50	2023-10-15 16:41:48
URL (Uses Javascript)	933	933	2023-10-15 16:43:59
URL (Uses a Web Framework)	4	207	2023-10-15 16:43:59
Username	7	12	2023-10-15 16:14:12
Web Analytics	6	292	2023-10-15 16:43:59
Web Content	1031	1368	2023-10-15 16:43:56
Web Content Type	19	1405	2023-10-15 16:43:56
Web Server	6	1404	2023-10-15 16:44:04
Web Technology	5	580	2023-10-15 16:44:02

FOCA (Fingerprinting Organizations with Collected Archives):

Se realizó un análisis de los metadatos asociados a documentos presentes en la web de Moonpay.com utilizando la herramienta FOCA. El objetivo era obtener información valiosa de los archivos indexados por los motores de búsqueda que podrían revelar detalles sensibles o configuraciones inadvertidas.

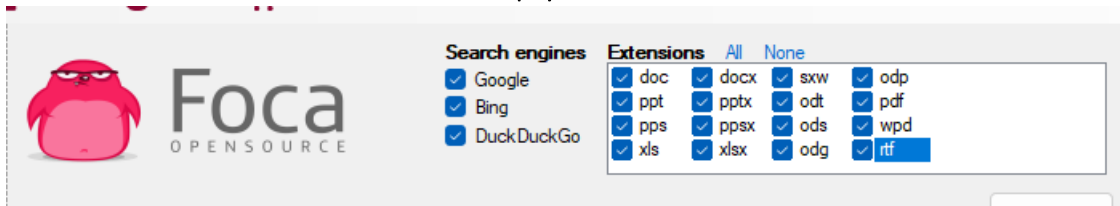
Creación del Proyecto:

Se creó un proyecto específico para el dominio Moonpay.com en FOCA. Este proyecto se configuró para buscar todos los formatos de documentos indexados por los motores de búsqueda.



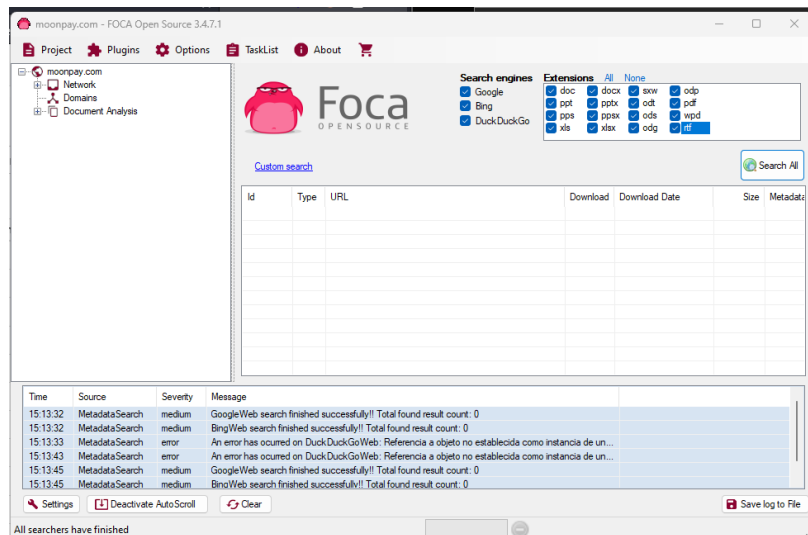
Inicio del Análisis:

Se inició el proceso de búsqueda y análisis automatizado de metadatos asociados a documentos vinculados a Moonpay.com en los buscadores.



Resultados del Análisis:

Lamentablemente, el análisis completo realizado por FOCA no encontró archivos relacionados con el dominio Moonpay.com. Esto indica que no se han indexado documentos públicos que contengan metadatos asociados a Moonpay.com en los motores de búsqueda.



Evaluación de Usuarios de GitHub con GitHub Search:

Para llevar a cabo la búsqueda de mails de usuarios de GitHub relacionados con Moonpay.com, se utilizó la herramienta GitHub Search. Primero, se clonó el repositorio y se instaló la herramienta con los siguientes comandos:

COMANDO

```
git clone https://github.com/gwen001/github-search
```

```
cd github-search
```

```
pip3 install -r requirements.txt
```

Luego, se generó un token de autenticación en GitHub y se procedió a escanear los usuarios relacionados con Moonpay utilizando el script github-users.py.

```
(root@kali)-[/home/sebastian/github-search]
# ls -l
total 544
-rwxr-xr-x 1 root root 14821 Oct 7 06:32 class.GitHubSearch.php
-rwxr-xr-x 1 root root 1821 Oct 7 06:32 class.Utils.php
-rw-r--r-- 1 root root 2261 Oct 7 06:32 dorks.txt
-rwxr-xr-x 1 root root 6332 Oct 7 06:32 git-history.py
-rwxr-xr-x 1 root root 2419 Oct 7 06:32 git-history.sh
-rwxr-xr-x 1 root root 6754 Oct 7 06:32 github-contributors.py
-rwxr-xr-x 1 root root 5463 Oct 7 06:32 github-dorks.php
-rwxr-xr-x 1 root root 5323 Oct 7 06:32 github-dorks.py
-rwxr-xr-x 1 root root 11578 Oct 7 06:32 github-employees.py
-rwxr-xr-x 1 root root 10172 Oct 7 06:32 github-endpoints.py
-rwxr-xr-x 1 root root 2614 Oct 7 06:32 github-grabrepo.php
-rwxr-xr-x 1 root root 1948 Oct 7 06:32 github-search.php
-rwxr-xr-x 1 root root 6524 Oct 7 06:32 github-secrets.py
-rwxr-xr-x 1 root root 5819 Oct 7 06:32 github-subdomains.py
drwxr-xr-x 5 root root 4096 Oct 7 06:32 github-survey
drwxr-xr-x 5 root root 4096 Oct 7 06:32 github-survey2
-rwxr-xr-x 1 root root 9924 Oct 7 06:32 github-survey2.py
-rwxr-xr-x 1 root root 5673 Oct 7 06:32 github-survey.py
-rwxr-xr-x 1 root root 5862 Oct 7 06:32 github-users.py
-rwxr-xr-x 1 root root 6351 Oct 7 06:32 git-pillage.py
drwxr-xr-x 2 root root 4096 Oct 7 06:32 goop
-rwxr-xr-x 1 root root 1031 Oct 7 06:32 gsearch-reflog.sh
-rwxr-xr-x 1 root root 655 Oct 7 06:32 gsearch.sh
-rw-r--r-- 1 root root 5980 Oct 7 06:32 HOWTO.txt
-rwxr-xr-x 1 root root 220691 Oct 7 06:32 jhaddix-tweet.png
-rw-r--r-- 1 root root 1079 Oct 7 06:32 LICENSE.md
-rwxr-xr-x 1 root root 134140 Oct 7 06:32 meeridian-tweet.png
-rw-r--r-- 1 root root 2193 Oct 7 06:32 memo.txt
drwxr-xr-x 2 root root 4096 Oct 7 06:32 modules
-rwxr-xr-x 1 root root 1639 Oct 7 06:32 README.md
-rw-r--r-- 1 root root 90 Oct 7 06:32 requirements.txt
-rw-r--r-- 1 root root 6 Oct 7 06:32 VERSION.md
```

El comando para encontrar usuarios con nuestro token siguiente:

```
python3 github-users.py -t ghp_Fhj7jAOVCOqGxeKFT9bT5k21Ylv3lO6yO4q -k moonpay >
gitsearchmoonpay.txt
```

Este comando permitió identificar posibles empleados de Moonpay en GitHub. Los resultados obtenidos se encuentran detallados en el archivo OSINT para su revisión.

```

(root@kali)-[/home/.../recopilacion/entrega/osint/github_search]
# cat gitsearchmoonpay.txt
[+] searching keyword: moonpay
[+] 16 users found, 1 pages.
[+] retrieving user list...
[+] 16 login found.
[+] retrieving profiles...
+-----+-----+-----+
+-----+-----+-----+
+-----+-----+-----+
|      Login      |      Profile      |      Name      | | | |
|      |      |      |      |      |      |
|      |      |      |      |      |      |
|      |      |      |      |      |      |
+-----+-----+-----+
+-----+-----+-----+
+-----+-----+-----+
| moonpay         | https://github.com/moonpay | MoonPay        |
| hello@moonpay.com |      |      |
|      |      |      |
+-----+-----+-----+
+-----+-----+-----+
+-----+-----+-----+
| vfaramond       | https://github.com/vfaramond | Victor Faramond |
| victor@moonpay.com | @moonpay ,moonpay |
|      |      |      |
+-----+-----+-----+
+-----+-----+-----+
+-----+-----+-----+
| jakewmiles      | https://github.com/jakewmiles | Jake Miles      |
| None           | @moonpay,moonpay |
+-----+-----+-----+

```