

# Flan Scan

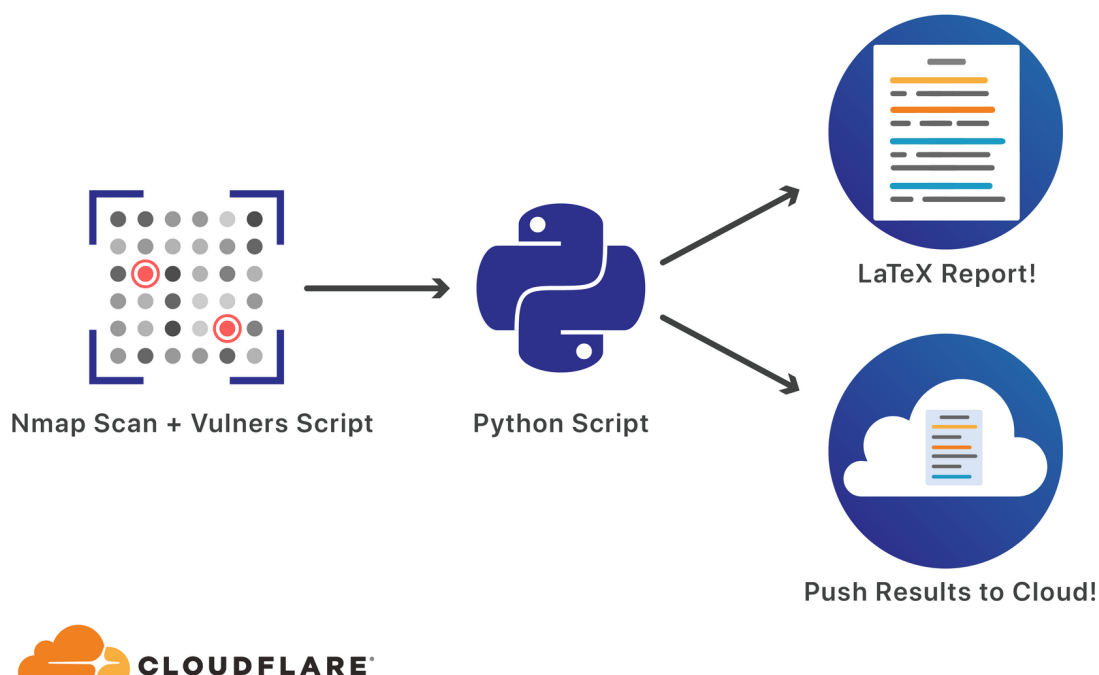
---

## ¿Qué es Flan Scan ?

Es una herramienta de escaneo de vulnerabilidades creado por cloudflare basada en nmap. La principal diferencia que existe la observamos en la instalación mientras que nmap lo instalamos por medio de paquetes o la fuente, **flan scan** se instala por medio de contenedores de dockers o por kubernetes.

**Flan Scan** permite creación de reportes de la salida normal a reportes de LaTeX y a su vez enviar estos reportes a buckets de S3 o de Google Cloud Buckets. De igual manera como parte los beneficios de esta herramientas es el motor de scripts NSE, que buscas resultados con los CVE de diversas fuentes.

La imagen a continuación vemos el funcionamiento de *Flan scan*



## Requerimientos

1. VPS les recomiendo [vultr.com](https://vultr.com)
2. Docker instalado

3. Tener los repositorios básicos de compilación en ubuntu o la distro que utilicen <https://zoomadmin.com/HowToInstall/UbuntuPackage/make>

## Instalación

1. Clonamos el repositorio

```
git clone https://github.com/cloudflare/flan.git
```

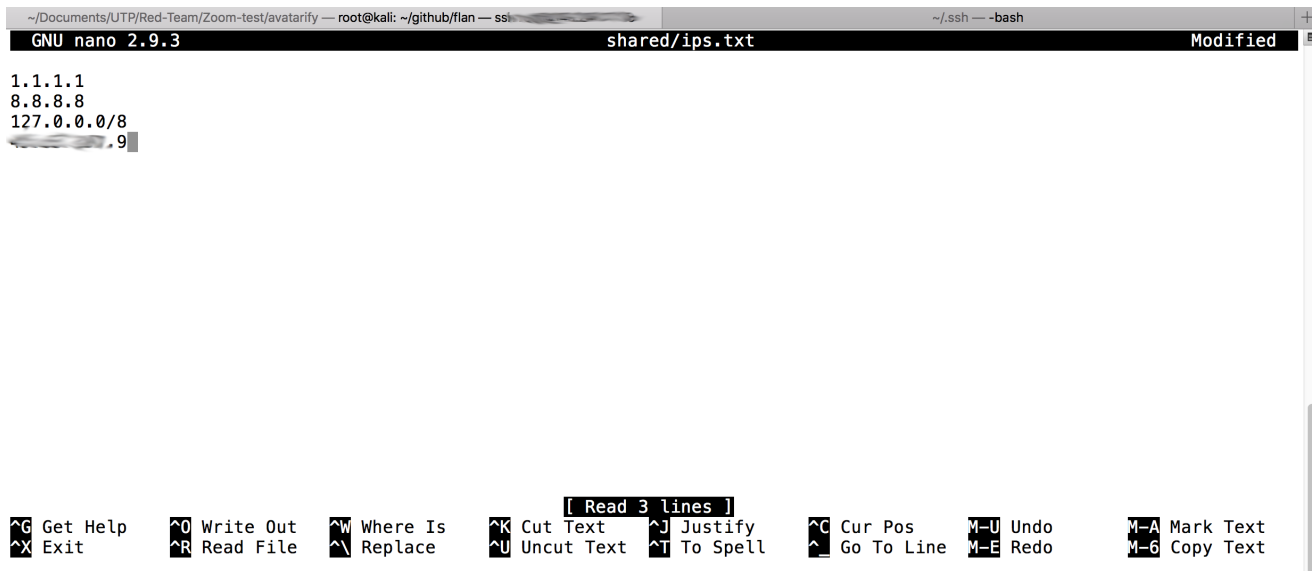
2. Accedemos al repositorio

```
cd flan
```

```
root@kali:~/github# cd flan/
root@kali:~/github/flan# ls
Dockerfile  Makefile  aws_push.py  gcp_push.py  output_report.py  run.sh
LICENSE     README.md  contrib      kubernetes_templates  requirements.txt  shared
root@kali:~/github/flan#
```

3. Modificamos el archivo **ips.txt** en la ruta **~flan/share/** y agregamos el ip del objetivo dentro

```
nano shared/ips.txt
```



```
~/Documents/UTP/Red-Team/Zoom-test/avatarify — root@kali: ~/github/flan — ssh — ~/.ssh — -bash
GNU nano 2.9.3 shared/ips.txt Modified
1.1.1.1
8.8.8.8
127.0.0.1
.9
[ Read 3 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos   M-U Undo     M-A Mark Text
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line M-E Redo     M-6 Copy Text
```

Debemos eliminar los ips que vienen por defecto **1.1.1.1 8.8.8.8 127.0.0.1**

4. Vamos a hacer la compilación del flan scan en el directorio **flan**

```
make build
```

```

root@kali:~/github/flan# make build
docker build -t flan_scan .
Sending build context to Docker daemon 216.1kB
Step 1/11 : FROM python:3.5-alpine
3.5-alpine: Pulling from library/python
cbdbe7a5bc2a: Pull complete
26ebcd19a4e3: Pull complete
79756be9c34e: Pull complete
7d0102152d61: Pull complete
390c669aade5: Pull complete
Digest: sha256:587435130cdefa4b79568d5ca18ae2b8061f71c6771a20ac879
Status: Downloaded newer image for python:3.5-alpine
----> 55fabf28273d
Step 2/11 : RUN apk add --no-cache nmap nmap-scripts git
----> Running in 6883c8ac2a53

```

5. Iniciamos el scaneo con el comando

```
make start
```

```

root@kali:~/github/flan# make start
docker run --name flan_1589082862 -v "/root/github/flan/shared:/shared:Z" flan_scan
# Nmap 7.80 scan initiated Sun May 10 03:54:23 2020 as: nmap -sV -oX /shared/xml_files/2020.05.10-03.54/45-1589082862.xml -oN -
-v1 --script=vulners/vulners.nse 158.9.vultr.com (45.158.9.9)
Nmap scan report for 158.9.vultr.com (45.158.9.9)
Host is up (0.068s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
25/tcp    filtered smtp
80/tcp    open  http      Apache httpd 2.4.29
|_http-server-header: Apache/2.4.29 (Ubuntu)
| vulners:
|   cpe:/a:apache:http_server:2.4.29:
|     CVE-2019-0211 7.2 https://vulners.com/cve/CVE-2019-0211
|     CVE-2018-1312 6.8 https://vulners.com/cve/CVE-2018-1312
|     CVE-2017-15715 6.8 https://vulners.com/cve/CVE-2017-15715
|     CVE-2019-10082 6.4 https://vulners.com/cve/CVE-2019-10082
|     CVE-2019-0217 6.0 https://vulners.com/cve/CVE-2019-0217
|     CVE-2020-1927 5.8 https://vulners.com/cve/CVE-2020-1927

```

Este comando realiza lo siguiente:

```
nmap -sV -oX /shared/xml_files -oN - -v1 \${ip} --script=vulners/vulners.nse
ip.obetivo
```

A continuación la descripción de cada parámetro y comando

- **-sV** permite la detección de la versión.
- **-oX** habilita la escritura de los resultados en formato XML en el directorio **shared/xml\_files**
- **-oN** Permite imprimir los resultados en la terminal y v1 muestra el nivel de verbose
- **--script=vulners/vulners.nse** script de NMAP para detectar los CVE en el host
- **ip.obetivo** es el ip del objetivo.

6. Podemos observar el contenedor que se levanto mientras se esta escaneando nuestro objetivo

```

root@kali:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
fed3d80f9dfc   flan_scan  "/run.sh"               38 minutes ago Up 38 minutes        flan_
1589080397
root@kali:~#

```

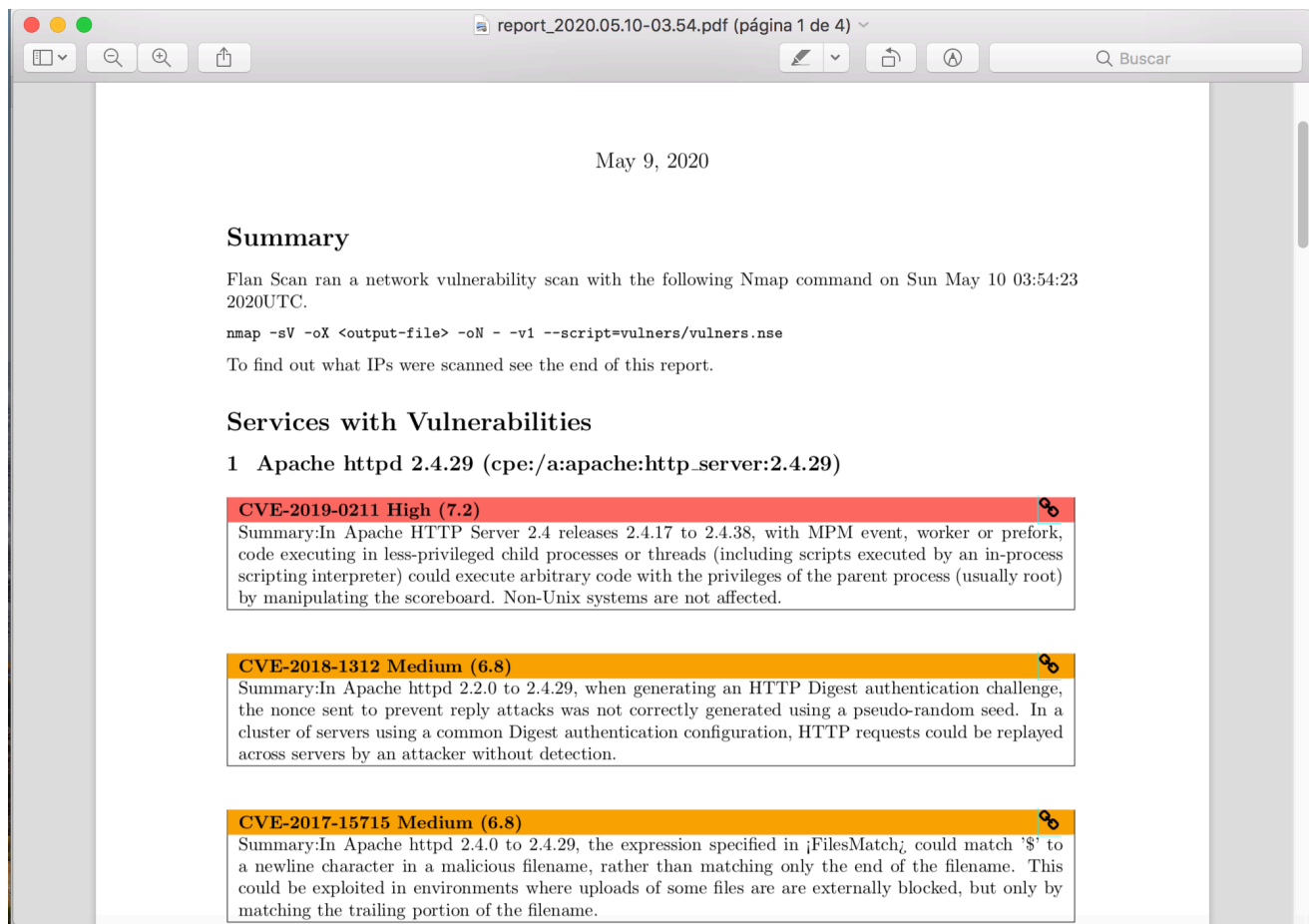
El escaneo podemos observar como va realizando el procedimiento normal de nmap

```
PORT      STATE      SERVICE    VERSION
22/tcp    open       ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; proc
25/tcp    filtered  smtp
80/tcp    open       http        Apache httpd 2.4.29
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_vulners:
|_cpe:/a:apache:http_server:2.4.29:
|_CVE-2019-0211 7.2 https://vulners.com/cve/CVE-2019-0211
|_CVE-2018-1312 6.8 https://vulners.com/cve/CVE-2018-1312
|_CVE-2017-15715 6.8 https://vulners.com/cve/CVE-2017-15715
|_CVE-2019-10082 6.4 https://vulners.com/cve/CVE-2019-10082
|_CVE-2019-0217 6.0 https://vulners.com/cve/CVE-2019-0217
|_CVE-2020-1927 5.8 https://vulners.com/cve/CVE-2020-1927
|_CVE-2019-10098 5.8 https://vulners.com/cve/CVE-2019-10098
|_CVE-2020-1934 5.0 https://vulners.com/cve/CVE-2020-1934
|_CVE-2019-10081 5.0 https://vulners.com/cve/CVE-2019-10081
|_CVE-2019-0220 5.0 https://vulners.com/cve/CVE-2019-0220
|_CVE-2019-0196 5.0 https://vulners.com/cve/CVE-2019-0196
|_CVE-2018-17199 5.0 https://vulners.com/cve/CVE-2018-17199
|_CVE-2018-1333 5.0 https://vulners.com/cve/CVE-2018-1333
|_CVE-2017-15710 5.0 https://vulners.com/cve/CVE-2017-15710
|_CVE-2019-0197 4.9 https://vulners.com/cve/CVE-2019-0197
|_CVE-2019-10092 4.3 https://vulners.com/cve/CVE-2019-10092
|_CVE-2018-11763 4.3 https://vulners.com/cve/CVE-2018-11763
|_CVE-2018-1283 3.5 https://vulners.com/cve/CVE-2018-1283
443/tcp   open       ssl/http   Apache httpd 2.4.29 ((Ubuntu))
```

La salida la encontramos en

```
root@kali:~/github/flan/shared# ls
ips.txt  reports  xml_files
root@kali:~/github/flan/shared# █
```

El directorio **reports** son los reportes en formato LaTeX



The above 18 vulnerabilities apply to these network locations:

- 45.33.30.1 Ports: ['80', '443']

## Services With No Known Vulnerabilities

### 1 smtp

- 45.33.30.1 Ports: ['25']

### 2 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (cpe:/a:openbsd:openssh:7.6p1) (cpe:/o:linux:linux\_kernel)

- 45.33.30.1 Ports: ['22']

## List of IPs Scanned

- 45.33.30.1/7.9

## Parametros Personalizados

```
docker run -v $(CURDIR)/shared:/shared flan_scan <Nmap-flags>
```

Se pueden enviar comandos adicionales de nmap con el comando arriba expuesto, dado que es un contenedor que contiene **NMAP**

## Conclusiones

- Los reportes en LaTeX son más organizados que la salida de texto y aunque se pueden procesar la salida y consumir los archivos en xml para obtener reportes más bonitos y con colores, para una prueba rápida conviene flan scan, en lugar de importar o usar algún plugin, además las herramientas de pago dan reportes basados en nmap <https://github.com/1N3/Sn1per>
- De igual manera que cualquier contenedor contiene parámetros que podemos enviar además como dentro del contenedor lo que tiene es nmap podemos enviar búsquedas más avanzadas.
- Permite enviar los reportes a S3 para ser consumidos luego por algún grupo de seguridad.
- La única desventaja está en la instalación de LaTeX y sino sabes usarlo para componer el pdf, pero no toma más de 5 minutos instalarlo y componer el pdf final.

## Referencias

- <https://blog.cloudflare.com/introducing-flan-scan/>
- <https://securitytrails.com/blog/flan-scan-vulnerability-scanner>
- <https://securitytrails.com/blog/nmap-vulnerability-scan>
- <https://www.latex-project.org/get/>
- <https://github.com/cloudflare/flan>
- <https://www.hypn.za.net/blog/2018/01/25/running-nmap-in-aws-lambda/>