# Finding Targets: an initial approach

**Author: Alexandre Borges**
**Date: 18/FEV/2014**
**Revision: 1.0**

Hi everybody, how are you ? This week some students of mine asked about how is possible to collect IP address of machines related with a potential target when initiating a penetration test. The old and first usual way is to find DNS server names running the command ***whois <domain>*** :

```
root@kali:~# whois alexandreborges.org

Domain Name:ALEXANDREBORGES.ORG
Domain ID: D169434947-LROR
(omitted output)
Name Server:NS1.WORDPRESS.COM
Name Server:NS2.WORDPRESS.COM
```

Picking both DNS servers we can try a domain transfer:

```
root@kali:~# dig @ns1.wordpress.com alexandreborges.org axfr
; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @ns1.wordpress.com
alexandreborges.org axfr
; (1 server found)
;; global options: +cmd
; Transfer failed.


root@kali:~# dig @ns2.wordpress.com alexandreborges.org axfr
; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @ns2.wordpress.com
alexandreborges.org axfr
; (1 server found)
;; global options: +cmd
; Transfer failed.
```

Sure, this case both attempts have failed because the DNS servers from Wordpress were well configured. Nonetheless, I´m sure you can find a misconfigured DNS server easily. ☺

Another good method to make a list of target hosts is to try to find references (and its IP address) in the target website . For example, the following basic script can help us this task:

```
root@kali:~# vi search_target.sh
#!/bin/bash
echo -n "Enter the domain: "
read domain
echo
echo -n "Enter the target site: "
read site
wget $site -O $domain.txt -o /dev/null
grep 'href=' $domain.txt | cut -d"/" -f3 | grep $domain | sort -
u > out.txt
```

```
for target in $(cat out.txt);
do
host $target | grep address | cut -d" " -f4 | sort -u
done
```

It follows an example:

```
root@kali:~# ./search_target.sh
Enter the domain: alexandreborges.org
Enter the target site: http://alexandreborges.org
192.0.80.250
192.0.81.250
66.155.11.238
66.155.9.238
76.74.254.120
76.74.254.123
```

This case I've tried the script against my blog and many results were related with Wordpress hosts. However, if you to execute the **search_target.sh** script entering a real website, you'll collect more valuable information from there. Finally, a brutal force method is to find any IP address from target (for example, executing **ping *website*** ), to take its network address (usually, a class C network) and to try the following script (**search_target_2.sh**):

```
root@bt:~# vi search_target_2.sh
#!/bin/sh
echo "Enter an IP network range: "
echo "eg:192.88.57"
read network
for hostip in `seq 1 254`;do
host $network.$hostip | grep "pointer" | cut -d" " -f5 >>
targethostnames.txt
done
```

Unfortunately some filter out will be necessary in the results to restrict found hostnames to the required target domain. Pay attention: we could have used a *grep* command after "**-f5**" option expliciting the domain to narrow the results, but some targets have more than one registered domain. ☺

Have a nice day.

**Alexandre Borges.**