此文章已于 0:59:20 2015/12/25 发布到 Net | Linux | 安全

★Kali信息收集~3.子域名系列

★3.1Netcraft ：子域名查询

重要 官网：**http://searchdns.netcraft.com/**

**输入要查询的域名，即可得知子域名**

计算机生成了可选文字:
Results for cnblogs.com 
Found 23 sites 
Site 
www.cnblogs.com 
q.cnblogs.com 
news.cnblogs.com 
passport.cnblogs.com 
i.cnblogs.com 
kb.cnblogs.com 
home.cnblogs.com 
msg.cnblogs.com 
files.cnblogs.com 
zzk.cnblogs.com 
images.cnblogs.com 
cnblogs.com 
group.cnblogs.com 
space.cnblogs.com 
pic002.cnblogs.com 
imageso.cnblogs.com 
ing.cnblogs.com 
wz.cnblogs.com 
job.cnblogs.com 
images2015.cnblogs.com 
Site Report 
Search: 
search tips 
linux 
linux 
linux 
linux 
linux 
linux 
linux 
linux 
windows server 2008 
windows server 2008 
windows server 2008 
unknown 
linux 
linux 
unknown 
unknown 
linux 
linux 
linux 
unknown 
example: site contains .netcraft.com 
First seen 
march 2004 
november 2011 
november 2008 
june 200g 
may 2014 
may 200g 
october 200g 
december 2014 
january 2008 
april 2009 
march 2012 
july 2013 
august 2011 
january 2008 
august 2010 
july 2015 
febuary 2011 
november 2008 
october 200g 
november 2015 
Netblock 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
ntt america, inc. 
ntt america, inc. 
aliyun computing co., ltd 
aliyun computing co., ltd 
aliyun computing co., ltd 
ntt america, inc. 

3.2Fierce ：子域名查询

* 1. 概述：

fierce 是使用多种技术来扫描目标主机IP地址和主机名的一个DNS服务器枚举工具。运用递归的方式来工作。它的工作原理是先通过查询本地DNS服务器来查找目标DNS服务器，然后使用目标DNS服务器来查找子域名。fierce的主要特点就是可以用来定位独立IP空间对应域名和主机名。

* 1. 参数：

root@Kali:/home/dnt# **fierce -h**

fierce.pl (C) Copywrite 2006,2007 - By RSnake at <http://ha.ckers.org/fierce/>

Usage: perl fierce.pl [-dns example.com] [OPTIONS]

Overview:

Fierce is a semi-lightweight scanner that helps locate non-contiguous

IP space and hostnames against specified domains. It's really meant

as a pre-cursor to nmap, unicornscan, nessus, nikto, etc, since all

of those require that you already know what IP space you are looking

for. This does not perform exploitation and does not scan the whole

internet indiscriminately. It is meant specifically to locate likely

targets both inside and outside a corporate network. Because it uses

DNS primarily you will often find mis-configured networks that leak

internal address space. That's especially useful in targeted malware.

Options:

-connect        Attempt to make http connections to any non RFC1918

(public) addresses. This will output the return headers but

be warned, this could take a long time against a company with

many targets, depending on network/machine lag. I wouldn't

recommend doing this unless it's a small company or you have a

lot of free time on your hands (could take hours-days).

Inside the file specified the text "Host:\n" will be replaced

by the host specified. Usage:

perl fierce.pl -dns example.com -connect headers.txt

-delay                The number of seconds to wait between lookups.

-dns                The domain you would like scanned.

-dnsfile         Use DNS servers provided by a file (one per line) for

reverse lookups (brute force).

-dnsserver        Use a particular DNS server for reverse lookups

(probably should be the DNS server of the target). Fierce

uses your DNS server for the initial SOA query and then uses

the target's DNS server for all additional queries by default.

-file                A file you would like to output to be logged to.

-fulloutput        When combined with -connect this will output everything

the webserver sends back, not just the HTTP headers.

-help                This screen.

-nopattern        Don't use a search pattern when looking for nearby

hosts. Instead dump everything. This is really noisy but

is useful for finding other domains that spammers might be

using. It will also give you lots of false positives,

especially on large domains.

-range                Scan an internal IP range (must be combined with

-dnsserver). Note, that this does not support a pattern

and will simply output anything it finds. Usage:

perl fierce.pl -range 111.222.333.0-255 -dnsserver ns1.example.co

-search                Search list. When fierce attempts to traverse up and

down ipspace it may encounter other servers within other

domains that may belong to the same company. If you supply a

comma delimited list to fierce it will report anything found.

This is especially useful if the corporate servers are named

different from the public facing website. Usage:

perl fierce.pl -dns examplecompany.com -search corpcompany,blahcompany

Note that using search could also greatly expand the number of

hosts found, as it will continue to traverse once it locates

servers that you specified in your search list. The more the

better.

-suppress        Suppress all TTY output (when combined with -file).

-tcptimeout        Specify a different timeout (default 10 seconds). You

may want to increase this if the DNS server you are querying

is slow or has a lot of network lag.

**-threads Specify how many threads to use while scanning (default**

**is single threaded).**

-traverse        Specify a number of IPs above and below whatever IP you

have found to look for nearby IPs. Default is 5 above and

below. Traverse will not move into other C blocks.

-version        Output the version number.

-wide                Scan the entire class C after finding any matching

hostnames in that class C. This generates a lot more traffic

but can uncover a lot more information.

-wordlist        Use a seperate wordlist (one word per line). Usage:

perl fierce.pl -dns examplecompany.com -wordlist dictionary.txt

* 1. 实例：***threads 是线程数，可以自己指定***

重要 root@Kali:/home/dnt# **fierce -dns** cnblogs.com **-threads** 100

Trying zone transfer first...

Unsuccessful in zone transfer (it was worth a shot)

Okay, trying the good old fashioned way... brute force

Checking for wildcard DNS...

\*\* Found 99901599299.cnblogs.com at 42.121.252.58.

\*\* High probability of wildcard DNS.

Now performing 2280 test(s)...

**120.26.70.206        files.cnblogs.com**

**42.121.129.234        images.cnblogs.com**

**223.6.251.45        group.cnblogs.com**

**223.6.251.45        home.cnblogs.com**

**42.121.129.234        download.cnblogs.com**

**221.181.200.235        cdn.cnblogs.com**

**42.121.131.85        mail.cnblogs.com**

**42.121.129.234        downloads.cnblogs.com**

**223.6.251.45        news.cnblogs.com**

**223.6.251.45        ad.cnblogs.com**

**42.121.254.229        p.cnblogs.com**

。。。。。。。。