Description

First what Fierce is not. Fierce is not an IP scanner, it is not a DDoS tool, it is not designed to scan the whole Internet or perform any un-targeted attacks. It is meant specifically to locate likely targets both inside and outside a corporate network. Only those targets are listed (unless the -nopattern switch is used). No exploitation is performed (unless you do something intentionally malicious with the -connect switch). Fierce is a reconnaissance tool. Fierce is a PERL script that quickly scans domains (usually in just a few minutes, assuming no network lag) using several tactics.

Source: http://ha.ckers.org/fierce/ Fierce Homepage | Kali Fierce Repo

```
  Author: RSnake
  License: GPLv2
```

Utilisation

```
root@kali:~# 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
```

Exemple d'utilisation

Run a default scan against the target domain (-dns example.com):

```
root@kali:~# fierce -dns example.com
DNS Servers for example.com:
    b.iana-servers.net
    a.iana-servers.net

Trying zone transfer first...
    Testing b.iana-servers.net
        Request timed out or transfer not allowed.
    Testing a.iana-servers.net
        Request timed out or transfer not allowed.

Unsuccessful in zone transfer (it was worth a shot)
Okay, trying the good old fashioned way... brute force

Checking for wildcard DNS...
Nope. Good.
Now performing 2280 test(s)...
```