Threat Hunting Project

In this lab, you will learn about performing threat hunting and tracking down symptoms related to IoC (Indicators of Compromise).

As a cybersecurity analyst, you are working to discover weaknesses and vulnerabilities that your organization, Structureality Inc., needs to mitigate throughout its internal network. In this lab, you will first use firewall logging to discover questionable network traffic. Next, you will use netstat to discover an IoC observable related to traffic abuse against a secure website. Next, you will perform focused threat hunting activities related to a few scenarios. Finally, you will investigate strange DNS activity to discover yet another IoC.

Understand your environment

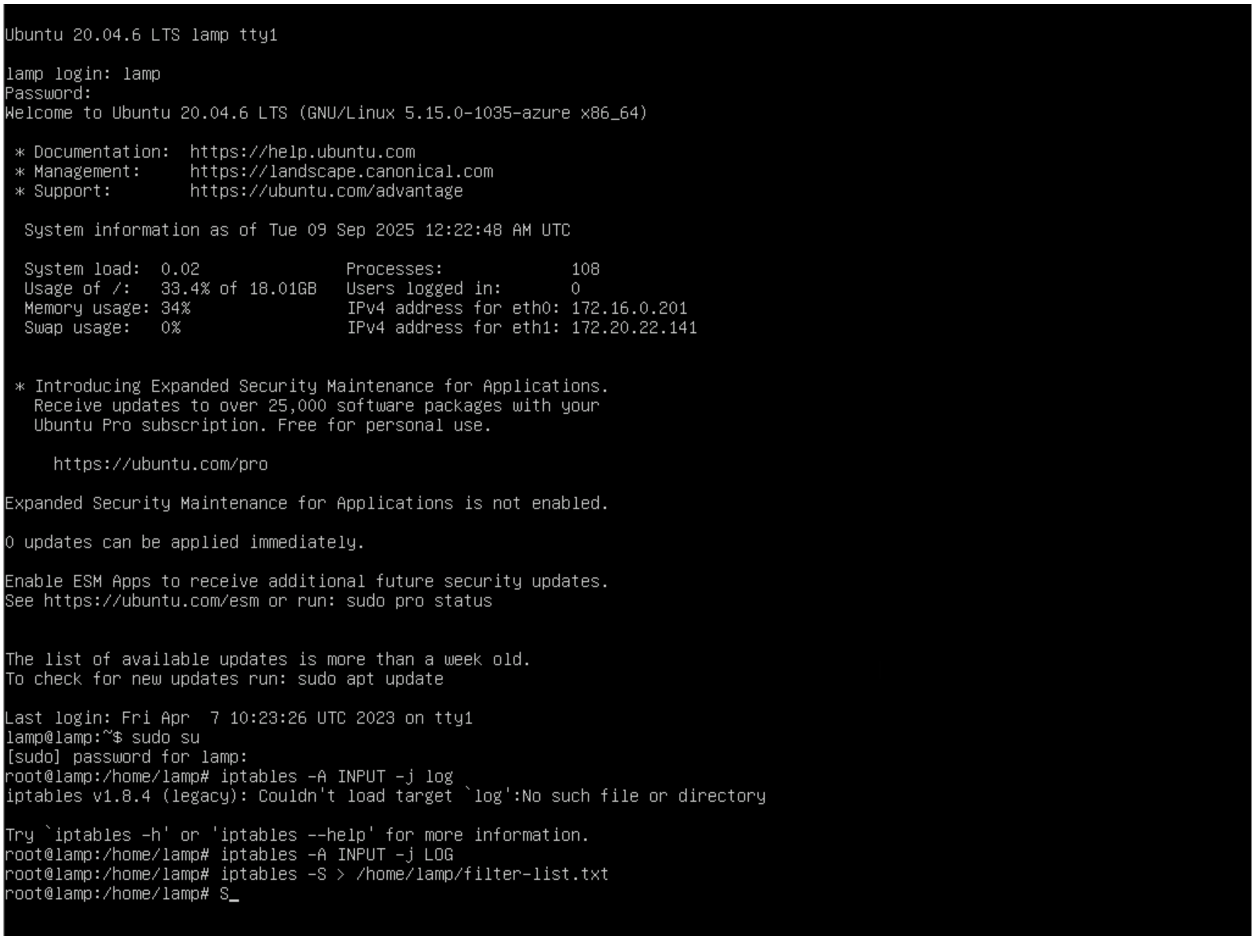
You will be working from several virtual machines in this lab:

- **LAMP** hosting Ubuntu server
- **KALI** hosting a pen-testing build of Debian Linux
- **DC10** hosting Windows Server 2019, serving as the domain controller, and hosting a secure website
- **MS10** hosting Windows Server 2016
- **PC10** hosting Windows Server 2019, which is serving as a client in this lab environment

Threat hunting network events

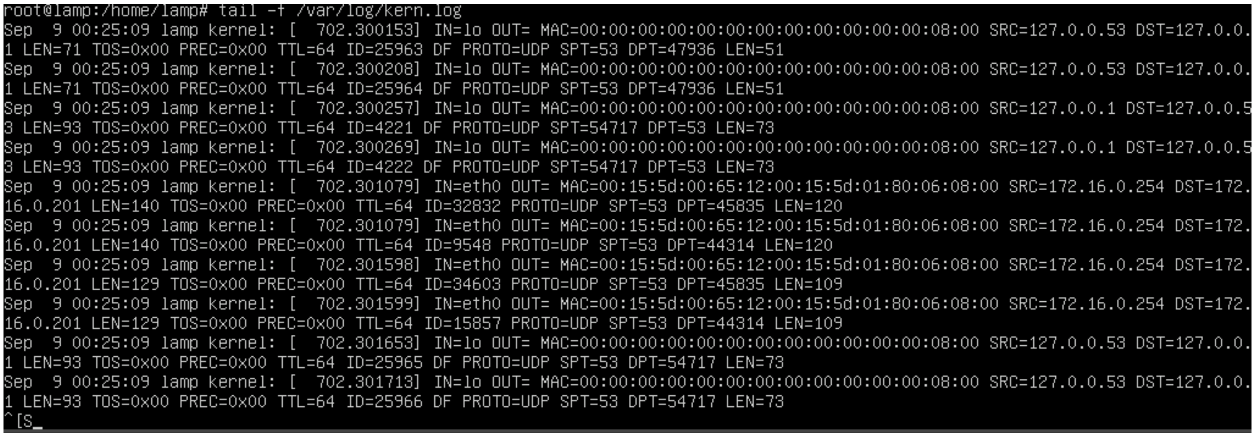
This exercise is an example of a common threat hunting process. Generally, to perform threat hunting, you are aware of an IoC (indicator of compromise), and you then use the details of that IoC to look for symptoms and occurrences within your own environment. The goal is to determine whether you have already been compromised or harmed by an exploit or attack concept that you only just learned about. This exercise has you perform network communication log analysis to find an IoC.

Sign into LAMP and elevate root privileges.



```
Ubuntu 20.04.6 LTS lamp tty1

lamp login: lamp
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1035-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue 09 Sep 2025 12:22:48 AM UTC

  System load:  0.02               Processes:             108
  Usage of /:   33.4% of 18.01GB   Users logged in:       0
  Memory usage: 34%                IPv4 address for eth0: 172.16.0.201
  Swap usage:   0%                 IPv4 address for eth1: 172.20.22.141


 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Apr  7 10:23:26 UTC 2023 on tty1
lamp@lamp:~$ sudo su
[sudo] password for lamp:
root@lamp:/home/lamp# iptables -A INPUT -j log
iptables v1.8.4 (legacy): Couldn't load target `log':No such file or directory

Try `iptables -h' or 'iptables --help' for more information.
root@lamp:/home/lamp# iptables -A INPUT -j LOG
root@lamp:/home/lamp# iptables -S > /home/lamp/filter-list.txt
root@lamp:/home/lamp# S_
```

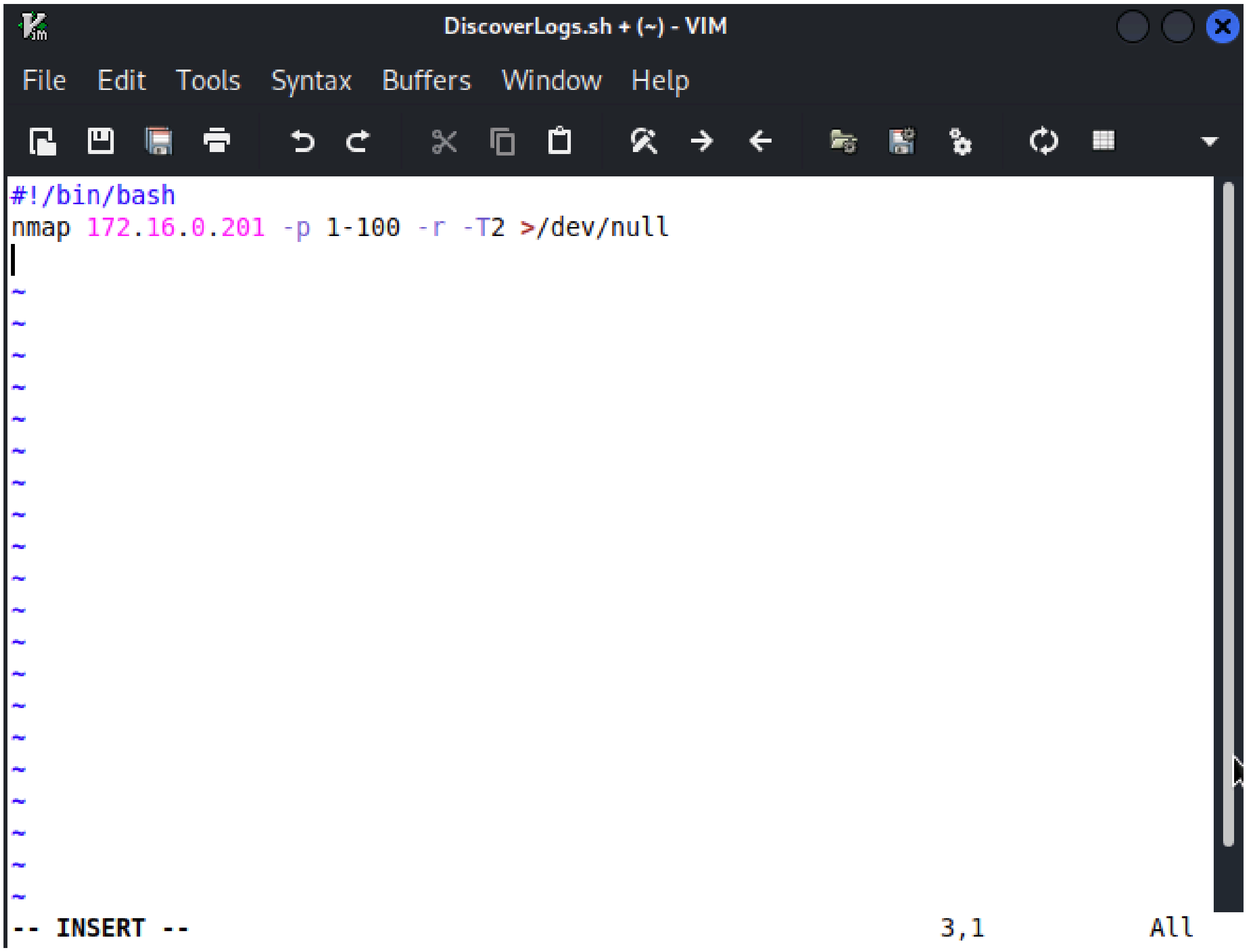
This command creates a file containing the current filters of iptables. This is necessary because the verification script below is unable to read the filters directly due to permission limitations.



```
root@lamp:/home/lamp# tail -f /var/log/kern.log
Sep  9 00:25:09 lamp kernel: [  702.300153] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.53 DST=127.0.0.
1 LEN=71 TOS=0x00 PREC=0x00 TTL=64 ID=25963 DF PROTO=UDP SPT=53 DPT=47936 LEN=51
Sep  9 00:25:09 lamp kernel: [  702.300208] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.53 DST=127.0.0.
1 LEN=71 TOS=0x00 PREC=0x00 TTL=64 ID=25964 DF PROTO=UDP SPT=53 DPT=47936 LEN=51
Sep  9 00:25:09 lamp kernel: [  702.300257] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.5
3 LEN=93 TOS=0x00 PREC=0x00 TTL=64 ID=4221 DF PROTO=UDP SPT=54717 DPT=53 LEN=73
Sep  9 00:25:09 lamp kernel: [  702.300269] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.5
3 LEN=93 TOS=0x00 PREC=0x00 TTL=64 ID=4222 DF PROTO=UDP SPT=54717 DPT=53 LEN=73
Sep  9 00:25:09 lamp kernel: [  702.301079] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:80:06:08:00 SRC=172.16.0.254 DST=172.
16.0.201 LEN=140 TOS=0x00 PREC=0x00 TTL=64 ID=32832 PROTO=UDP SPT=53 DPT=45835 LEN=120
Sep  9 00:25:09 lamp kernel: [  702.301079] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:80:06:08:00 SRC=172.16.0.254 DST=172.
16.0.201 LEN=140 TOS=0x00 PREC=0x00 TTL=64 ID=9548 PROTO=UDP SPT=53 DPT=44314 LEN=120
Sep  9 00:25:09 lamp kernel: [  702.301598] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:80:06:08:00 SRC=172.16.0.254 DST=172.
16.0.201 LEN=129 TOS=0x00 PREC=0x00 TTL=64 ID=34603 PROTO=UDP SPT=53 DPT=45835 LEN=109
Sep  9 00:25:09 lamp kernel: [  702.301599] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:80:06:08:00 SRC=172.16.0.254 DST=172.
16.0.201 LEN=129 TOS=0x00 PREC=0x00 TTL=64 ID=15857 PROTO=UDP SPT=53 DPT=44314 LEN=109
Sep  9 00:25:09 lamp kernel: [  702.301653] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.53 DST=127.0.0.
1 LEN=93 TOS=0x00 PREC=0x00 TTL=64 ID=25965 DF PROTO=UDP SPT=53 DPT=54717 LEN=73
Sep  9 00:25:09 lamp kernel: [  702.301713] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.53 DST=127.0.0.
1 LEN=93 TOS=0x00 PREC=0x00 TTL=64 ID=25966 DF PROTO=UDP SPT=53 DPT=54717 LEN=73
^[S_
```

This command displays the last ten (10) entries in the log file. The `-f` parameter will auto-update the result as new entries are added to the log file. Leave that running.

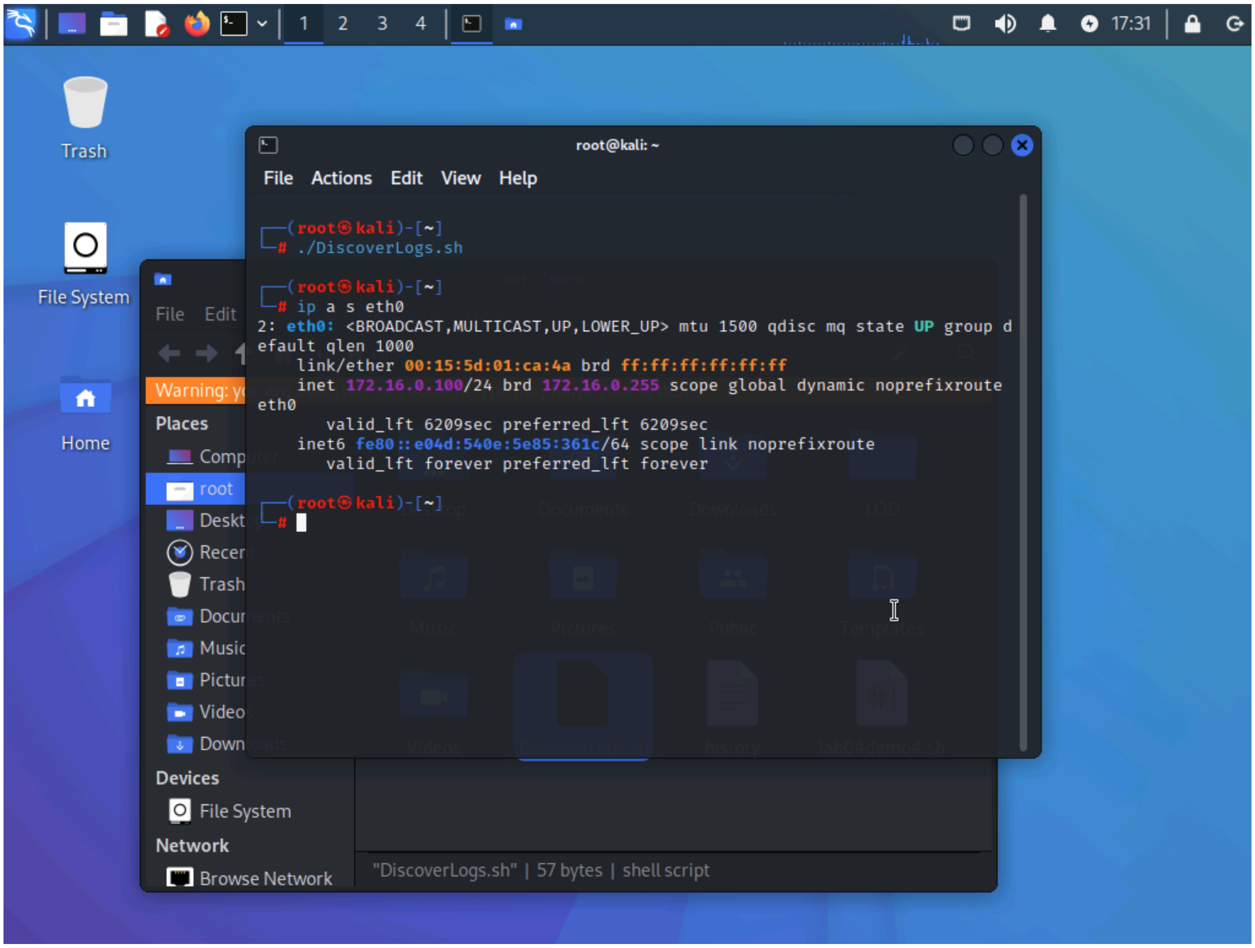
Go to Kali and write the following script to perform an operation for you to discover in the log on LAMP.



DiscoverLogs.sh + (~) - VIM
File Edit Tools Syntax Buffers Window Help

```
#!/bin/bash
nmap 172.16.0.201 -p 1-100 -r -T2 >/dev/null

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                             3,1           All
```

Go ahead and run it in the terminal.

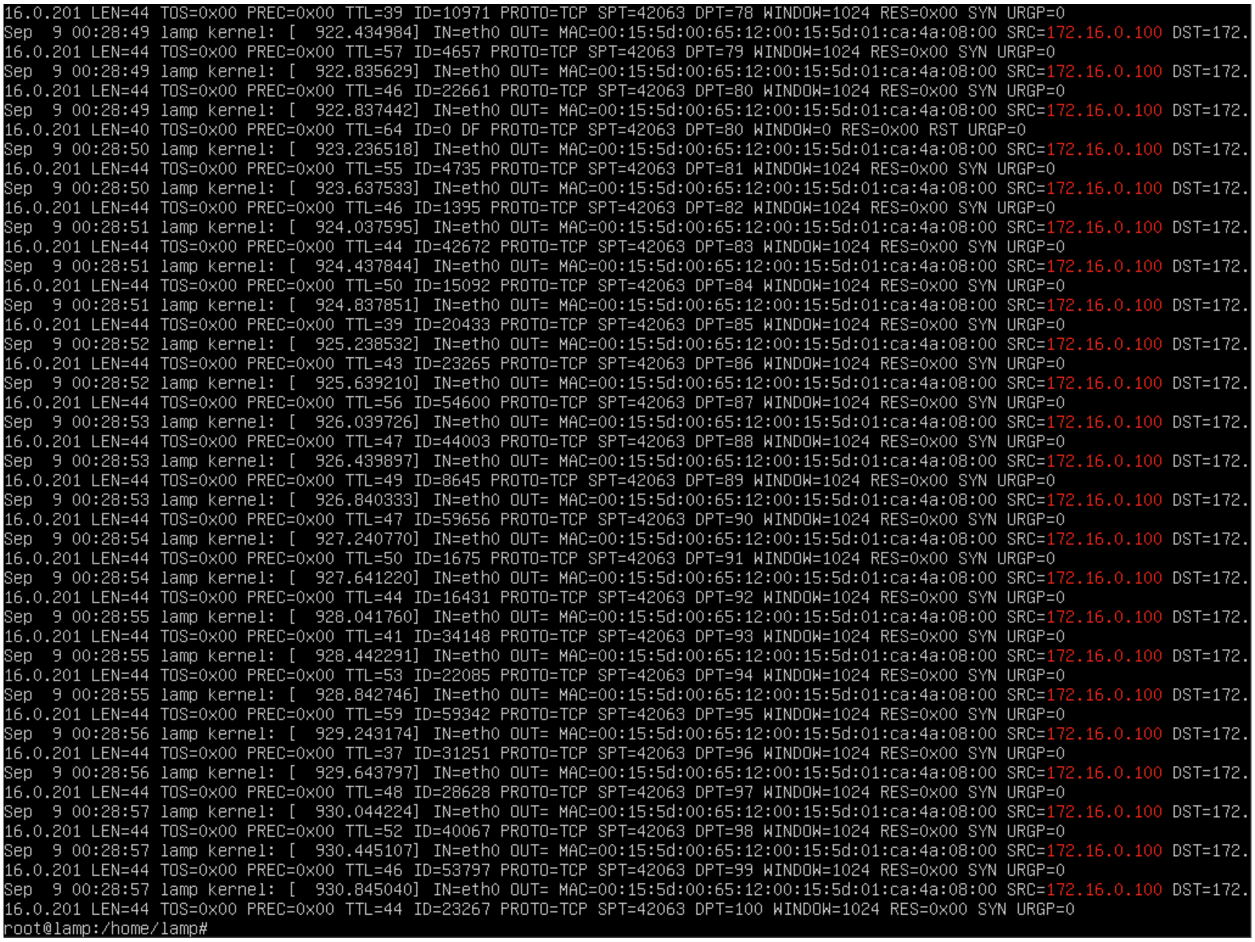


```
┌──(root㉿kali)-[~]
└─# ./DiscoverLogs.sh

┌──(root㉿kali)-[~]
└─# ip a s eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group d
efault qlen 1000
    link/ether 00:15:5d:01:ca:4a brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.100/24 brd 172.16.0.255 scope global dynamic noprefixroute
eth0
       valid_lft 6209sec preferred_lft 6209sec
    inet6 fe80::e04d:540e:5e85:361c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(root㉿kali)-[~]
└─# 
```

We can also show the IP address of the Kali machine that we will use in LAMP. Go back to LAMP. You suspect that the system using the IPv4 address of 172.16.0.100 is performing unwanted network communications with the lamp system. To investigate this, enter the following:

```
grep 172.16.0.100 /var/log/kern.log
```

This command performs a grep search of the `/var/log/kern.log` file and displays the entries that match the key term of 172.16.0.100.



```
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=39 ID=10971 PROTO=TCP SPT=42063 DPT=78 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:49 lamp kernel: [  922.434984] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=57 ID=4657 PROTO=TCP SPT=42063 DPT=79 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:49 lamp kernel: [  922.835629] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=46 ID=22661 PROTO=TCP SPT=42063 DPT=80 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:49 lamp kernel: [  922.837442] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=40 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=TCP SPT=42063 DPT=80 WINDOW=0 RES=0x00 RST URGP=0
Sep  9 00:28:50 lamp kernel: [  923.236518] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=55 ID=4735 PROTO=TCP SPT=42063 DPT=81 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:50 lamp kernel: [  923.637533] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=46 ID=1395 PROTO=TCP SPT=42063 DPT=82 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:51 lamp kernel: [  924.037595] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=44 ID=42672 PROTO=TCP SPT=42063 DPT=83 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:51 lamp kernel: [  924.437844] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=50 ID=15092 PROTO=TCP SPT=42063 DPT=84 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:51 lamp kernel: [  924.837851] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=39 ID=20433 PROTO=TCP SPT=42063 DPT=85 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:52 lamp kernel: [  925.238532] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=43 ID=23265 PROTO=TCP SPT=42063 DPT=86 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:52 lamp kernel: [  925.639210] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=56 ID=54600 PROTO=TCP SPT=42063 DPT=87 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:53 lamp kernel: [  926.039726] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=47 ID=44003 PROTO=TCP SPT=42063 DPT=88 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:53 lamp kernel: [  926.439897] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=49 ID=8645 PROTO=TCP SPT=42063 DPT=89 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:53 lamp kernel: [  926.840333] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=47 ID=59656 PROTO=TCP SPT=42063 DPT=90 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:54 lamp kernel: [  927.240770] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=50 ID=1675 PROTO=TCP SPT=42063 DPT=91 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:54 lamp kernel: [  927.641220] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=44 ID=16431 PROTO=TCP SPT=42063 DPT=92 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:55 lamp kernel: [  928.041760] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=41 ID=34148 PROTO=TCP SPT=42063 DPT=93 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:55 lamp kernel: [  928.442291] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=53 ID=22085 PROTO=TCP SPT=42063 DPT=94 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:55 lamp kernel: [  928.842746] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=59 ID=59342 PROTO=TCP SPT=42063 DPT=95 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:56 lamp kernel: [  929.243174] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=37 ID=31251 PROTO=TCP SPT=42063 DPT=96 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:56 lamp kernel: [  929.643797] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=48 ID=28628 PROTO=TCP SPT=42063 DPT=97 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:57 lamp kernel: [  930.044224] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=52 ID=40067 PROTO=TCP SPT=42063 DPT=98 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:57 lamp kernel: [  930.445107] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=46 ID=53797 PROTO=TCP SPT=42063 DPT=99 WINDOW=1024 RES=0x00 SYN URGP=0
Sep  9 00:28:57 lamp kernel: [  930.845040] IN=eth0 OUT= MAC=00:15:5d:00:65:12:00:15:5d:01:ca:4a:08:00 SRC=172.16.0.100 DST=172.
16.0.201 LEN=44 TOS=0x00 PREC=0x00 TTL=44 ID=23267 PROTO=TCP SPT=42063 DPT=100 WINDOW=1024 RES=0x00 SYN URGP=0
root@lamp:/home/lamp#
```

After inspecting the logs, the attacking machine is evidently performing a port scan. This is an indicator of compromise. This exercise had you discover the pattern of port scanning. This IoC is when there are numerous connection attempts from the same source IP address but to different port numbers. In this example, the port numbers were scanned in numerical order from ports 1 to 100.

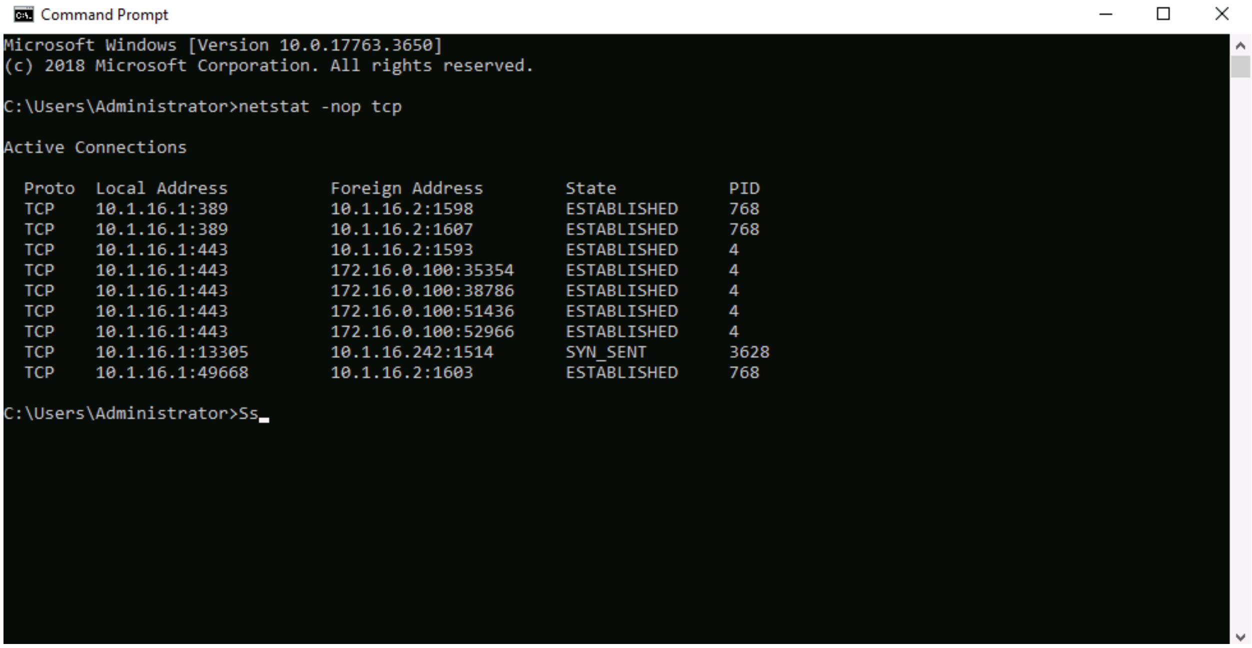
Track down abnormal connections

In this exercise, you need to be aware that the DC10 and MS10 systems are located in the internal LAN, and the Kali system is located in the screened subnet. Your threat intelligence service has just provided you with a new IoC. You will perform threat hunting in this exercise to determine if your environment is being affected by this new threat.

IoC: observable: Secure websites are being targeted in a resource exhaustion attack. The attack can originate from any system with the ability to access the targeted website. The offending process will not present as a standard web client.

You have been notified by system management tools that the Web service on DC10 is being subjected to some sort of repeated connection activity which is consuming a significant portion of system resources.

Access DC10 as an administrator and begin the investigation by running the following command:

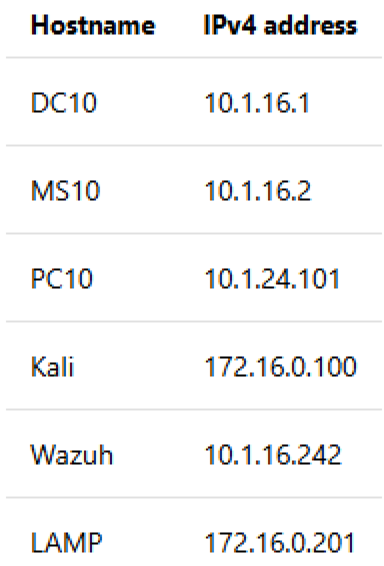


```
Microsoft Windows [Version 10.0.17763.3650]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>netstat -nop tcp

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    10.1.16.1:389          10.1.16.2:1598         ESTABLISHED     768
  TCP    10.1.16.1:389          10.1.16.2:1607         ESTABLISHED     768
  TCP    10.1.16.1:443          10.1.16.2:1593         ESTABLISHED     4
  TCP    10.1.16.1:443          172.16.0.100:35354     ESTABLISHED     4
  TCP    10.1.16.1:443          172.16.0.100:38786     ESTABLISHED     4
  TCP    10.1.16.1:443          172.16.0.100:51436     ESTABLISHED     4
  TCP    10.1.16.1:443          172.16.0.100:52966     ESTABLISHED     4
  TCP    10.1.16.1:13305        10.1.16.242:1514       SYN_SENT        3628
  TCP    10.1.16.1:49668        10.1.16.2:1603         ESTABLISHED     768

C:\Users\Administrator>Ss
```

The netstat command is used to view current network connections and their status.

- The `-n` parameter forces numbers only to be displayed instead of hostnames, FQDNs, and protocol acronyms (such as TCP or HTTP).
- The `-o` parameter displays the associated process ID (PID).
- The `-p tcp` parameter limits the display to the selected protocol, in this instance: TCP.

You consult your network configuration documentation to determine the following details:



Hostname	IPv4 address
DC10	10.1.16.1
MS10	10.1.16.2
PC10	10.1.24.101
Kali	172.16.0.100
Wazuh	10.1.16.242
LAMP	172.16.0.201

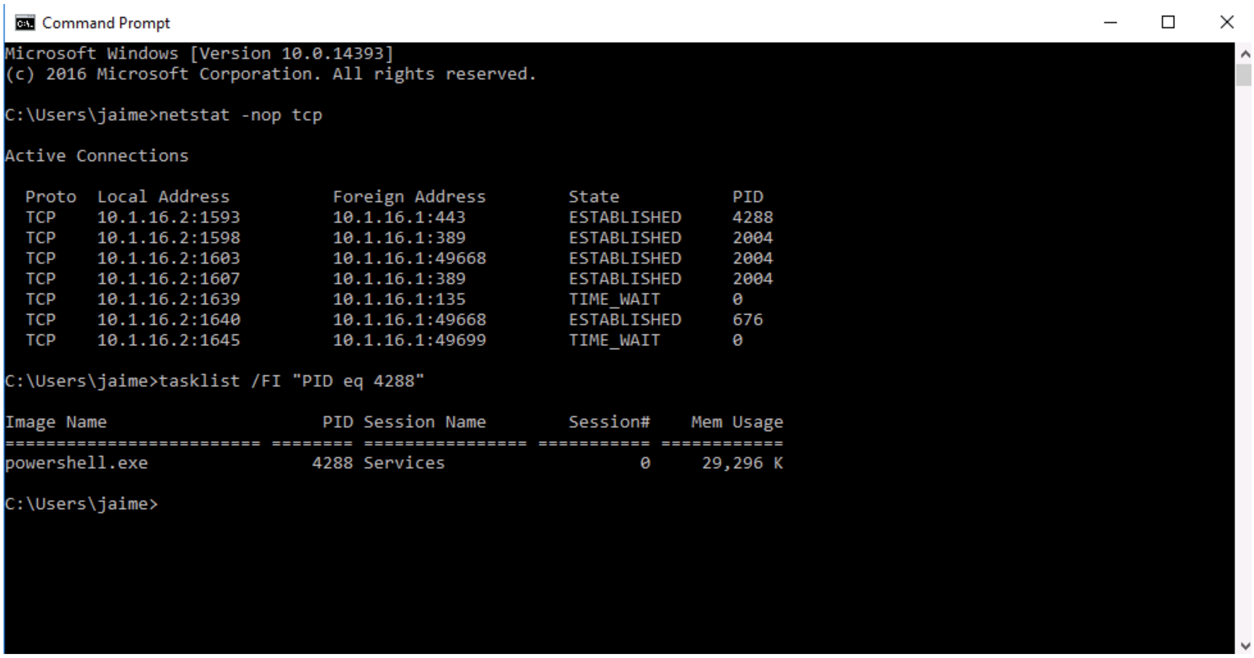
Based on the netstat output and the network configuration documentation, you elect to investigate MS10 next.

The PID of secure services connected is 4. So, do

Enter `tasklist /FI "PID eq 4"` to see only the process associated with the PID of concern.

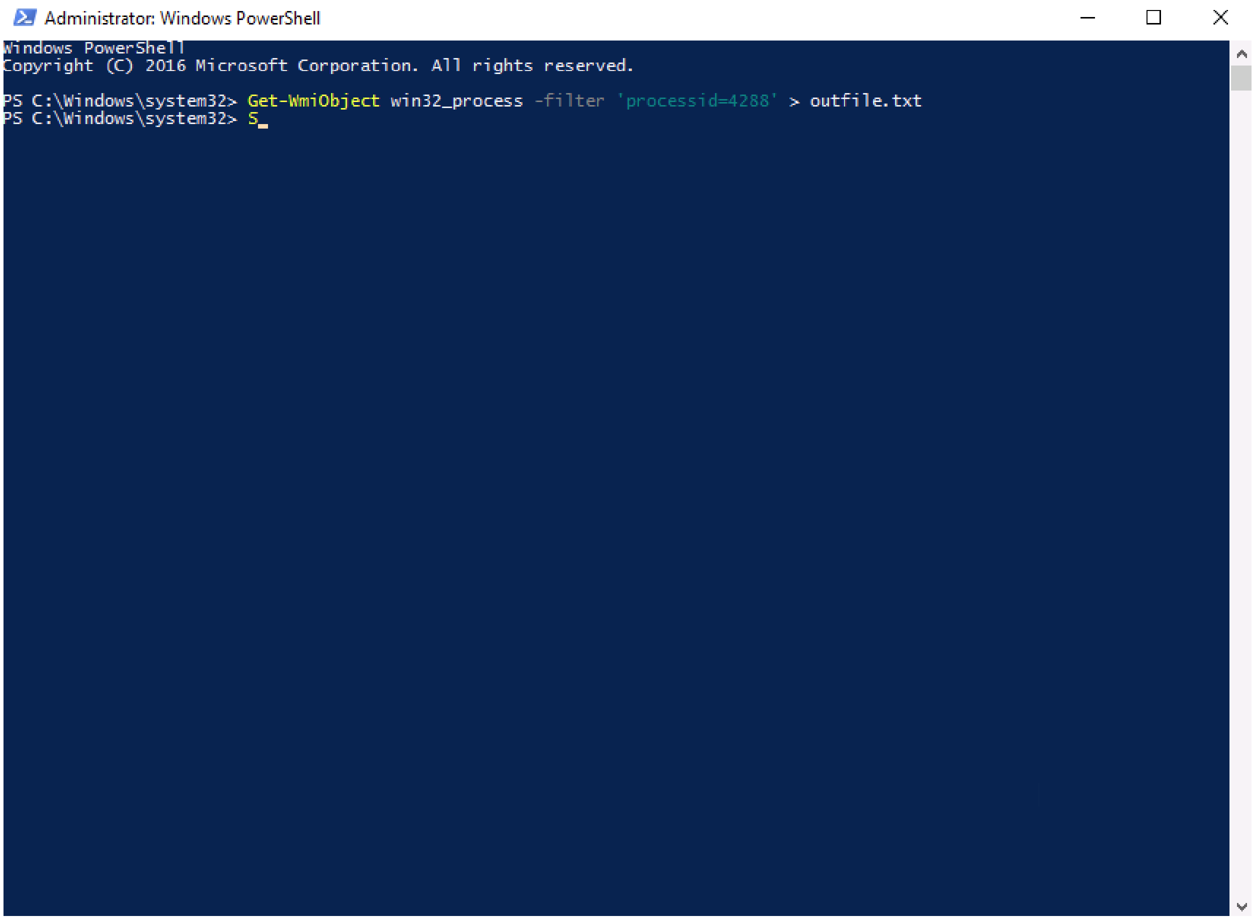
The name of the process associated with this PID is System. Given this information, I decided to investigate MS10 instead.

We perform a similar process to view current session and see the process associated with PID of the secure connection.



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\jaime>netstat -nop tcp

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    10.1.16.2:1593         10.1.16.1:443          ESTABLISHED     4288
  TCP    10.1.16.2:1598         10.1.16.1:389          ESTABLISHED     2004
  TCP    10.1.16.2:1603         10.1.16.1:49668        ESTABLISHED     2004
  TCP    10.1.16.2:1607         10.1.16.1:389          ESTABLISHED     2004
  TCP    10.1.16.2:1639         10.1.16.1:135          TIME_WAIT       0
  TCP    10.1.16.2:1640         10.1.16.1:49668        ESTABLISHED     676
  TCP    10.1.16.2:1645         10.1.16.1:49699        TIME_WAIT       0

C:\Users\jaime>tasklist /FI "PID eq 4288"

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
powershell.exe                4288 Services                   0     29,296 K

C:\Users\jaime>
```

Open powershell and run a command to export to a file the details about the process of concern.

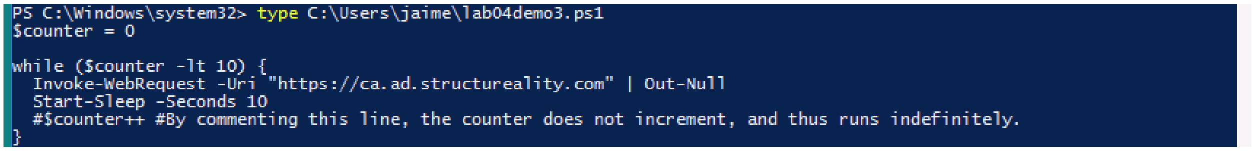


```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Get-WmiObject win32_process -filter 'processid=4288' > outfile.txt
PS C:\Windows\system32> S_
```

Enter Select-String -Path .\outfile.txt -Pattern 'CommandLine' to view the CommandLine element of the PowerShell process to see the name of the script is running.

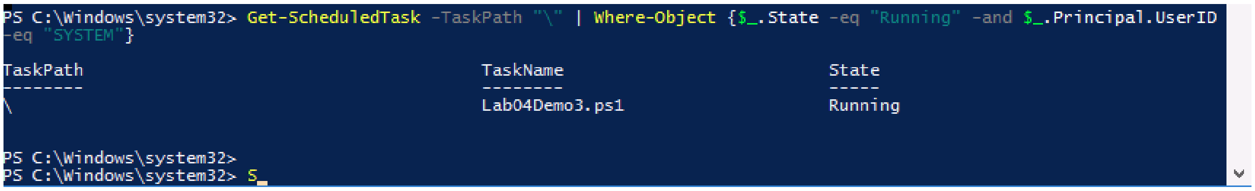
The script that is executing in the PowerShell process with PID: 4288 is `C:\Users\jaime\lab04demo3.ps1`

`View the contents of the script:`



```
PS C:\Windows\system32> type C:\Users\jaime\lab04demo3.ps1
$counter = 0

while ($counter -lt 10) {
  Invoke-WebRequest -Uri "https://ca.ad.structureality.com" | Out-Null
  Start-Sleep -Seconds 10
  #$counter++ #By commenting this line, the counter does not increment, and thus runs indefinitely.
}
```

You suspect this is a regularly scheduled task and check:



```
PS C:\Windows\system32> Get-ScheduledTask -TaskPath "\" | Where-Object {$_.State -eq "Running" -and $_.Principal.UserID
-eq "SYSTEM"}

TaskPath                                   TaskName                        State
--------                                   --------                        -----
\                                          Lab04Demo3.ps1                  Running


PS C:\Windows\system32>
PS C:\Windows\system32> S_
```

Now that you have eliminated MS10 as a cause of IoC-related connections, you will shift your attention over to Kali.

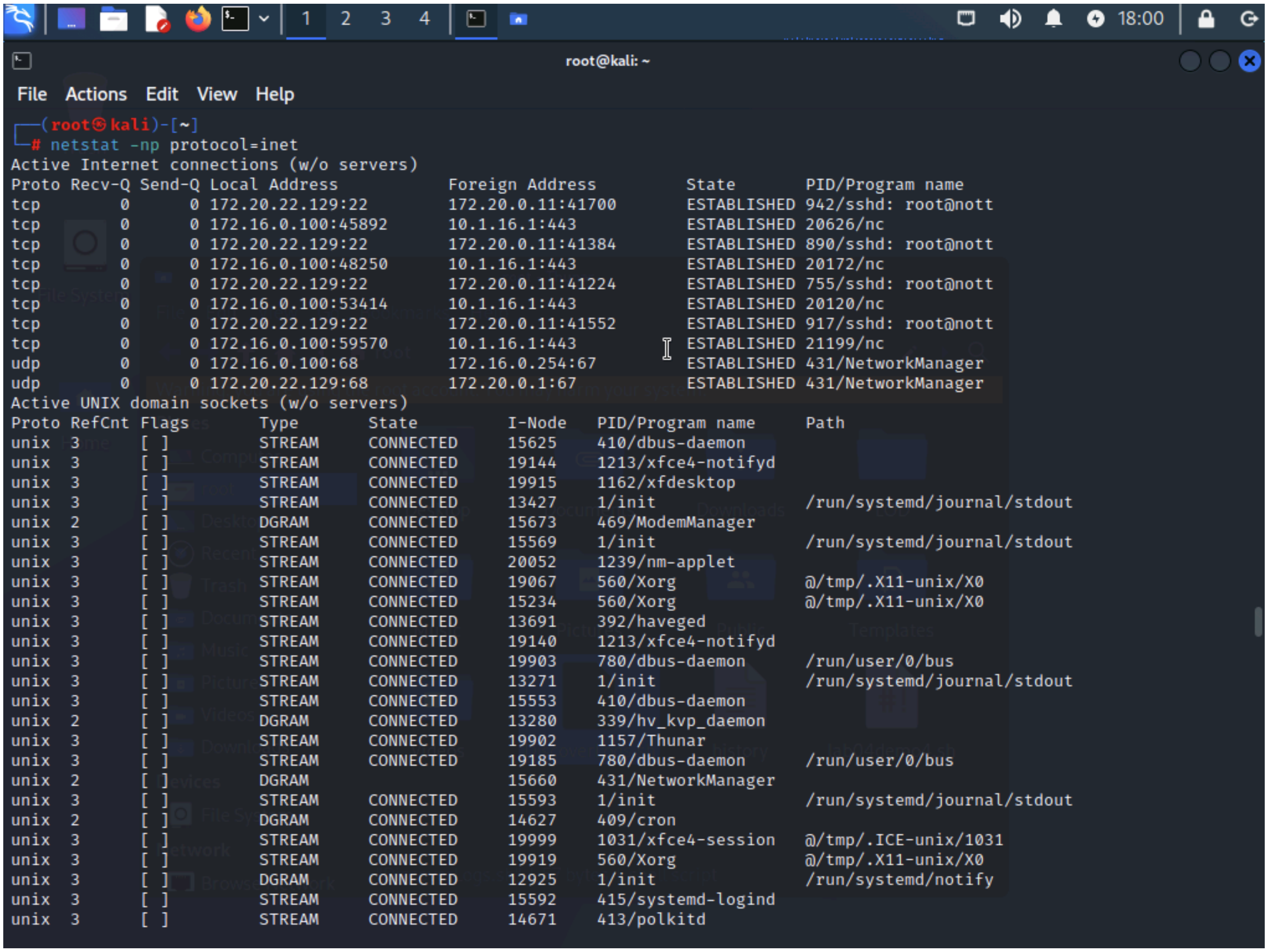
In Kali,

Enter **netstat -np --protocol=inet** to view the active processes on Kali related to IPv4 connections.

The netstat command on Linux is similar to but not exactly the same as the command on Windows. To view the full syntax, enter *netstat -h*. The parameters used here are:

- The `-n` parameter forces numbers only to be displayed instead of hostnames, FQDNs, and protocol acronyms (such as TCP or HTTP).
- The `-p` parameter displays the PID and process/program name
- The `--protocol=inet` parameter limits the display to only IPv4 protocols.
-

Notice how there are potentially several secure web sessions from Kali to DC10.



```
┌──(root㉿kali)-[~]
└─# netstat -np protocol=inet
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 172.20.22.129:22        172.20.0.11:41700       ESTABLISHED 942/sshd: root@nott
tcp        0      0 172.16.0.100:45892      10.1.16.1:443           ESTABLISHED 20626/nc
tcp        0      0 172.20.22.129:22        172.20.0.11:41384       ESTABLISHED 890/sshd: root@nott
tcp        0      0 172.16.0.100:48250      10.1.16.1:443           ESTABLISHED 20172/nc
tcp        0      0 172.20.22.129:22        172.20.0.11:41224       ESTABLISHED 755/sshd: root@nott
tcp        0      0 172.16.0.100:53414      10.1.16.1:443           ESTABLISHED 20120/nc
tcp        0      0 172.20.22.129:22        172.20.0.11:41552       ESTABLISHED 917/sshd: root@nott
tcp        0      0 172.16.0.100:59570      10.1.16.1:443           ESTABLISHED 21199/nc
udp        0      0 172.16.0.100:68         172.16.0.254:67         ESTABLISHED 431/NetworkManager
udp        0      0 172.20.22.129:68        172.20.0.1:67           ESTABLISHED 431/NetworkManager
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   PID/Program name     Path
unix  3      [ ]         STREAM     CONNECTED     15625    410/dbus-daemon
unix  3      [ ]         STREAM     CONNECTED     19144    1213/xfce4-notifyd
unix  3      [ ]         STREAM     CONNECTED     19915    1162/xfdesktop
unix  3      [ ]         STREAM     CONNECTED     13427    1/init               /run/systemd/journal/stdout
unix  2      [ ]         DGRAM      CONNECTED     15673    469/ModemManager
unix  3      [ ]         STREAM     CONNECTED     15569    1/init               /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     20052    1239/nm-applet
unix  3      [ ]         STREAM     CONNECTED     19067    560/Xorg             @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     15234    560/Xorg             @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     13691    392/haveged
unix  3      [ ]         STREAM     CONNECTED     19140    1213/xfce4-notifyd
unix  3      [ ]         STREAM     CONNECTED     19903    780/dbus-daemon      /run/user/0/bus
unix  3      [ ]         STREAM     CONNECTED     13271    1/init               /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     15553    410/dbus-daemon
unix  2      [ ]         DGRAM      CONNECTED     13280    339/hv_kvp_daemon
unix  3      [ ]         STREAM     CONNECTED     19902    1157/Thunar
unix  3      [ ]         STREAM     CONNECTED     19185    780/dbus-daemon      /run/user/0/bus
unix  2      [ ]         DGRAM                    15660    431/NetworkManager
unix  3      [ ]         STREAM     CONNECTED     15593    1/init               /run/systemd/journal/stdout
unix  2      [ ]         DGRAM      CONNECTED     14627    409/cron
unix  3      [ ]         STREAM     CONNECTED     19999    1031/xfce4-session   @/tmp/.ICE-unix/1031
unix  3      [ ]         STREAM     CONNECTED     19919    560/Xorg             @/tmp/.X11-unix/X0
unix  3      [ ]         DGRAM      CONNECTED     12925    1/init               /run/systemd/notify
unix  3      [ ]         STREAM     CONNECTED     15592    415/systemd-logind
unix  3      [ ]         STREAM     CONNECTED     14671    413/polkitd
```

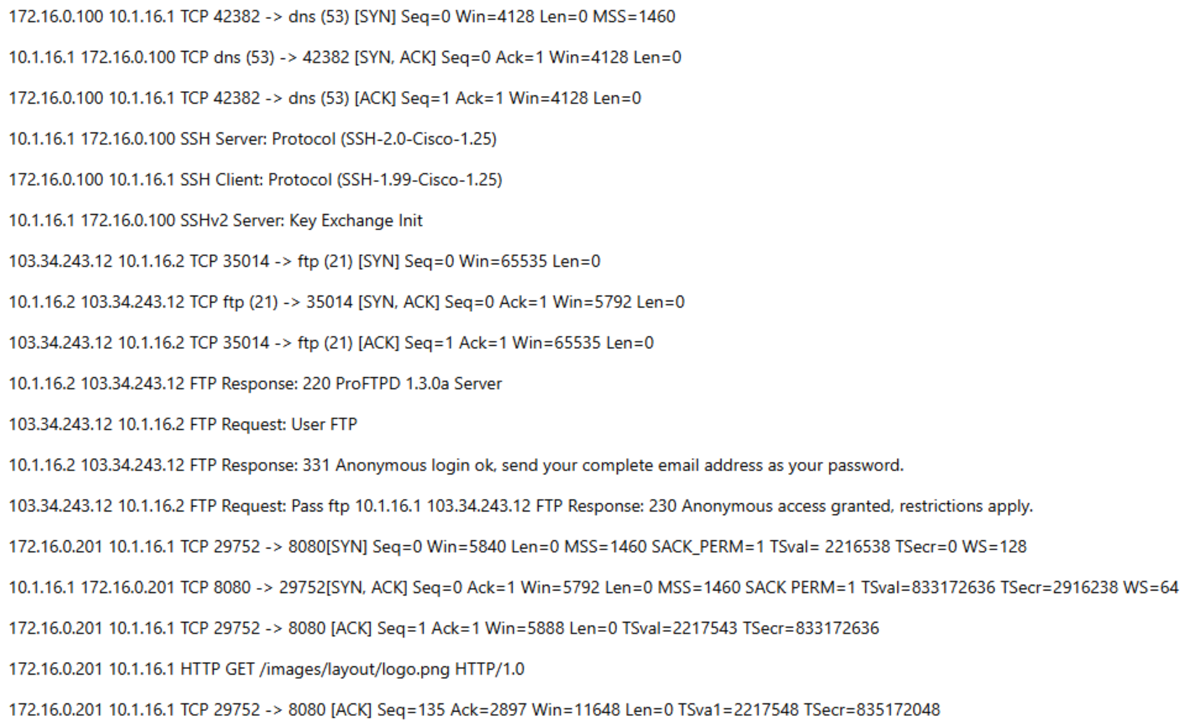
Having identified the system from where the suspicious secure web connections are originating, you need to compare your findings to the elements of the IoC: observable.

In this exercise, you have used an IoC to perform threat hunting. You traced the unwanted activity from a secure website host (i.e., DC10) to the origins of the abuse. You were able to eliminate MS10 as a suspected host of malware. Then you confirmed that Kali was the host of the abusive connections.

Threat Hunting challenges

There are innumerable ways for adversaries to cause problems within a network or on a single system. In this exercise, you are presented with various examples of logs or system information that represent a problem or an IoC.

Problem 1:



```
172.16.0.100 10.1.16.1 TCP 42382 -> dns (53) [SYN] Seq=0 Win=4128 Len=0 MSS=1460

10.1.16.1 172.16.0.100 TCP dns (53) -> 42382 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0

172.16.0.100 10.1.16.1 TCP 42382 -> dns (53) [ACK] Seq=1 Ack=1 Win=4128 Len=0

10.1.16.1 172.16.0.100 SSH Server: Protocol (SSH-2.0-Cisco-1.25)

172.16.0.100 10.1.16.1 SSH Client: Protocol (SSH-1.99-Cisco-1.25)

10.1.16.1 172.16.0.100 SSHv2 Server: Key Exchange Init

103.34.243.12 10.1.16.2 TCP 35014 -> ftp (21) [SYN] Seq=0 Win=65535 Len=0

10.1.16.2 103.34.243.12 TCP ftp (21) -> 35014 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0

103.34.243.12 10.1.16.2 TCP 35014 -> ftp (21) [ACK] Seq=1 Ack=1 Win=65535 Len=0

10.1.16.2 103.34.243.12 FTP Response: 220 ProFTPD 1.3.0a Server

103.34.243.12 10.1.16.2 FTP Request: User FTP

10.1.16.2 103.34.243.12 FTP Response: 331 Anonymous login ok, send your complete email address as your password.

103.34.243.12 10.1.16.2 FTP Request: Pass ftp 10.1.16.1 103.34.243.12 FTP Response: 230 Anonymous access granted, restrictions apply.

172.16.0.201 10.1.16.1 TCP 29752 -> 8080[SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval= 2216538 TSecr=0 WS=128

10.1.16.1 172.16.0.201 TCP 8080 -> 29752[SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK PERM=1 TSval=833172636 TSecr=2916238 WS=64

172.16.0.201 10.1.16.1 TCP 29752 -> 8080 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=2217543 TSecr=833172636

172.16.0.201 10.1.16.1 HTTP GET /images/layout/logo.png HTTP/1.0

172.16.0.201 10.1.16.1 TCP 29752 -> 8080 [ACK] Seq=135 Ack=2897 Win=11648 Len=0 TSva1=2217548 TSecr=835172048
```

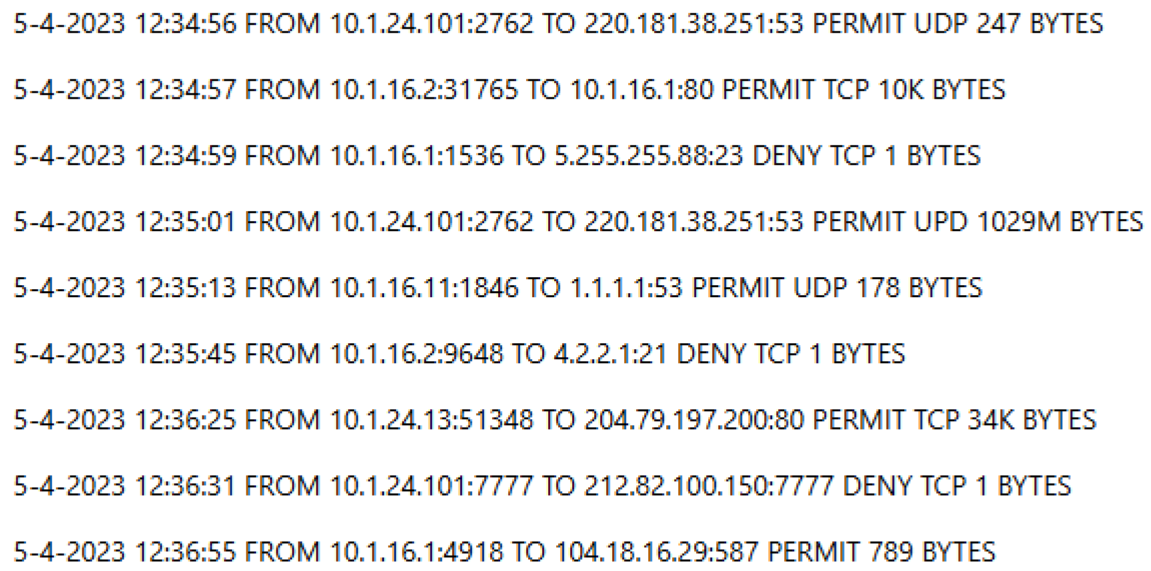
What is the most concerning issue you discovered in this packet capture?

An anonymous connection was made to an FTP server.

10.1.16.2 is likely a victim of malware infection.

Problem 2 (Firewall Log):

You have been tasked with investigating the exfiltration of a significant amount of sensitive company data. You are reviewing a portion of the firewall log around the time the breach occurred. Your goal is to identify IoCs.

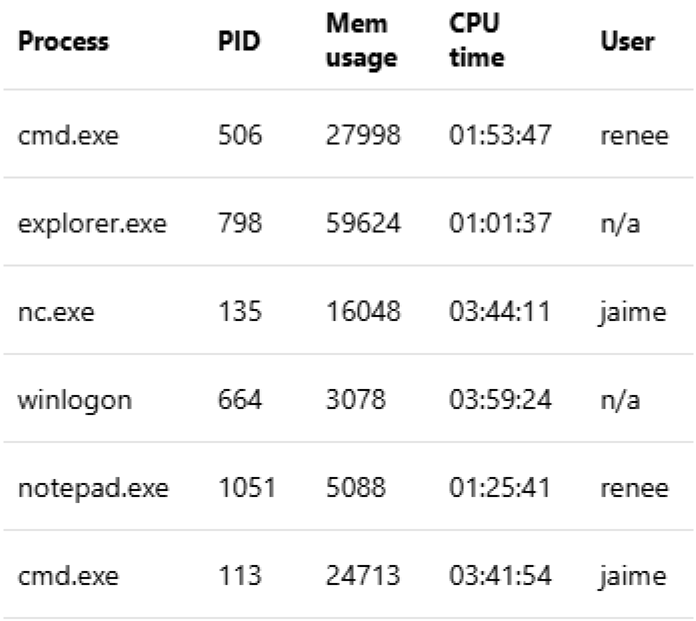


```
5-4-2023 12:34:56 FROM 10.1.24.101:2762 TO 220.181.38.251:53 PERMIT UDP 247 BYTES
5-4-2023 12:34:57 FROM 10.1.16.2:31765 TO 10.1.16.1:80 PERMIT TCP 10K BYTES
5-4-2023 12:34:59 FROM 10.1.16.1:1536 TO 5.255.255.88:23 DENY TCP 1 BYTES
5-4-2023 12:35:01 FROM 10.1.24.101:2762 TO 220.181.38.251:53 PERMIT UPD 1029M BYTES
5-4-2023 12:35:13 FROM 10.1.16.11:1846 TO 1.1.1.1:53 PERMIT UDP 178 BYTES
5-4-2023 12:35:45 FROM 10.1.16.2:9648 TO 4.2.2.1:21 DENY TCP 1 BYTES
5-4-2023 12:36:25 FROM 10.1.24.13:51348 TO 204.79.197.200:80 PERMIT TCP 34K BYTES
5-4-2023 12:36:31 FROM 10.1.24.101:7777 TO 212.82.100.150:7777 DENY TCP 1 BYTES
5-4-2023 12:36:55 FROM 10.1.16.1:4918 TO 104.18.16.29:587 PERMIT 789 BYTES
```

10.1.24.101 permits 1029M Bytes of data, a large amount of data over the DNS port, a clear IoC.

Problem 3:

Your ISP has reported to your organization that they suspect one of your internal systems is functioning as a command and control (C&C) server for a botnet. You have been tasked with evaluating internal systems and identifying any IoCs related to this issue. You pull an active process report for a client system. Here is a portion of that report:



Process	PID	Mem usage	CPU time	User
cmd.exe	506	27998	01:53:47	renee
explorer.exe	798	59624	01:01:37	n/a
nc.exe	135	16048	03:44:11	jaime
winlogon	664	3078	03:59:24	n/a
notepad.exe	1051	5088	01:25:41	renee
cmd.exe	113	24713	03:41:54	jaime

The primary indicator that this client system is running a botnet is the presence of nc.exe. This is the Netcat utility executable on Windows. It can be used to establish remote network connections. Netcat can be used as both a client initiating connections and a server receiving connections.

Another aspect of this process report which supports nc.exe as the problem process, is that it has been executing for almost as long as the system has been running. The winlogon process will start at boot. The report shows that a Command Prompt was launched less than 20 mins after booting, and then the Netcat process was launched a few minutes later.

You might also notice that the report shows that the user Jamie provided the user context for the Netcat process. This does not necessarily mean Jaime was running the C&C on purpose. It is possible Jaime was fooled by a social engineering attack that tricked them into executing something that launched Netcat in the background.

Since the process report shows active processes for both Jaime and Renee. This indicates that Jaime was using the client system first, then instead of logging out, a switch user function was employed for Renee to log in.

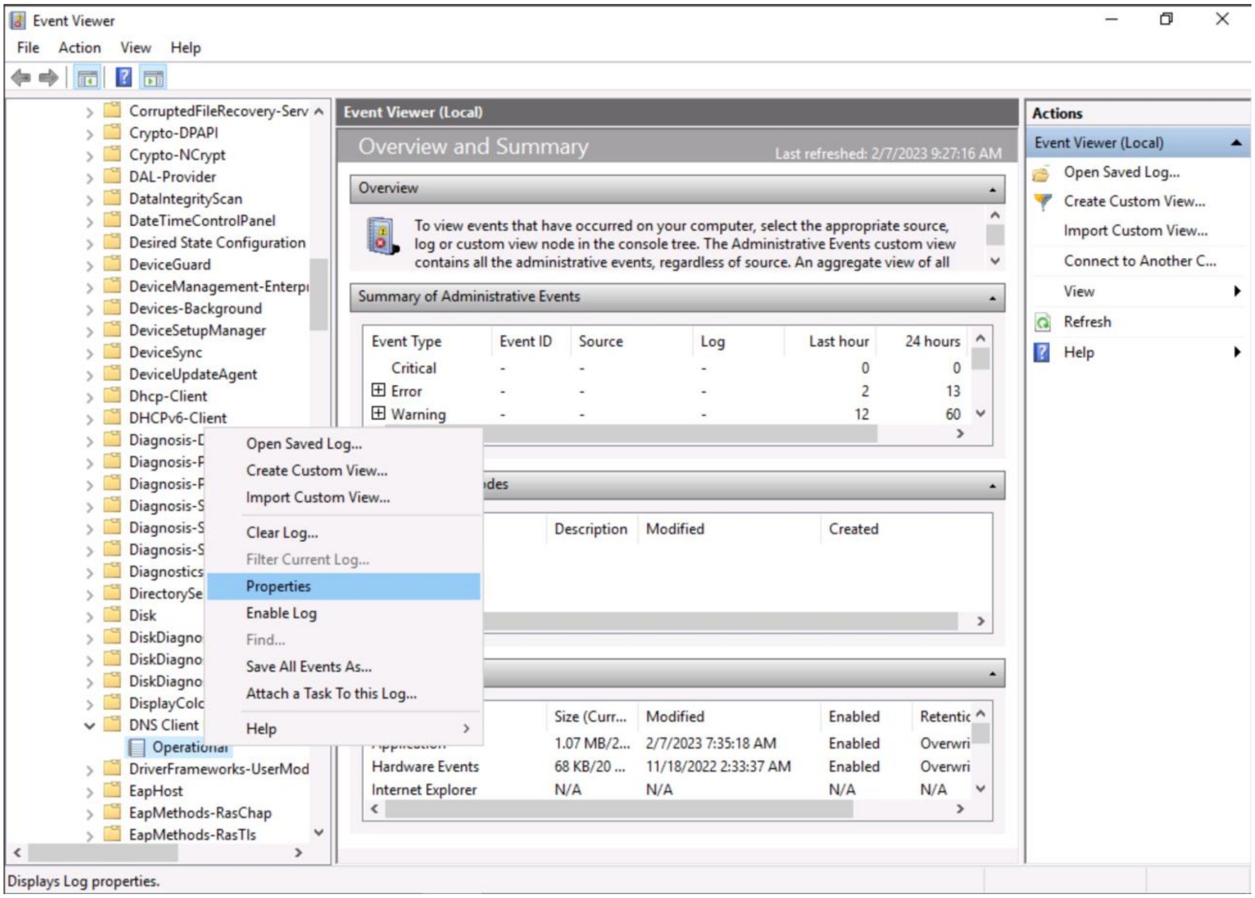
Investigate Strange DNS activity

The security team at your ISP has informed you that there is suspicious activity taking place across the Internet connection. The communications are initiated by a system in the Structureality private network. You elect to start your investigation of the issue on the PC10 client system. This is one of the clients that has had issues in the past due to the user's poor security hygiene. You decide to enable DNS logging to see if it can detect IoCs related to suspicious activity.

1. Select **Type here to search** from the taskbar, type event, then select **Event Viewer** from the results.
2. Maximize the Event Viewer window.
3. In the left pane, select the arrow beside **Applications and Service logs** to expand its contents.
4. Select the arrow beside **Microsoft**, then select the arrow beside **Windows** to expand its contents.
5. Scroll down to locate, then select the arrow beside **DNS Client Events** to expand its contents.

You may need to click-hold-drag-release the pane divisions to resize them. You may need or want to readjust the panes throughout this exercise.

6. Right-click **Operational**, then select **Properties**.



To view events that have occurred on your computer, select the appropriate source, log or custom view node in the console tree. The Administrative Events custom view contains all the administrative events, regardless of source. An aggregate view of all

Event Type	Event ID	Source	Log	Last hour	24 hours
Critical	-	-	-	0	0
⊞ Error	-	-	-	2	13
⊞ Warning	-	-	-	12	60

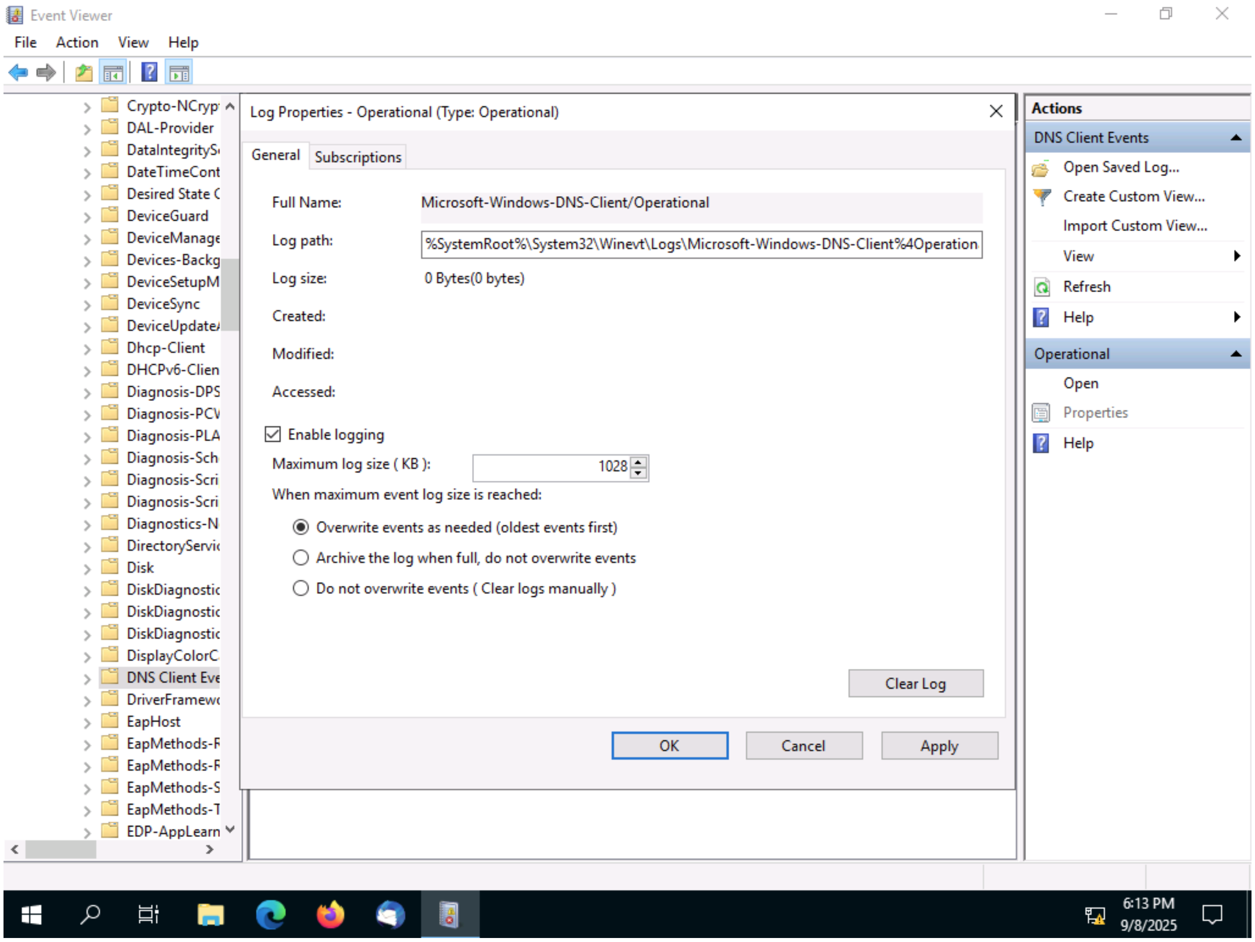
Description	Modified	Created

	Size (Curr...	Modified	Enabled	Retentic
Application	1.07 MB/2...	2/7/2023 7:35:18 AM	Enabled	Overwri
Hardware Events	68 KB/20 ...	11/18/2022 2:33:37 AM	Enabled	Overwri
Internet Explorer	N/A	N/A	N/A	N/A

Displays Log properties.

7. Select to mark the checkbox **Enable logging**, then select **OK**.

In a real-world investigation, you would allow the log to collect entries for a period of time, then begin reviewing the recorded events to look for suspicious activities. In this exercise, you will be initiating a script that will perform the activities that will be labeled as "suspicious" as you perform threat hunting.



Log Properties - Operational (Type: Operational)

Full Name: Microsoft-Windows-DNS-Client/Operational

Log path: %SystemRoot%\System32\Winevt\Logs\Microsoft-Windows-DNS-Client%4Operation

Log size: 0 Bytes(0 bytes)

Created:

Modified:

Accessed:

☑ Enable logging

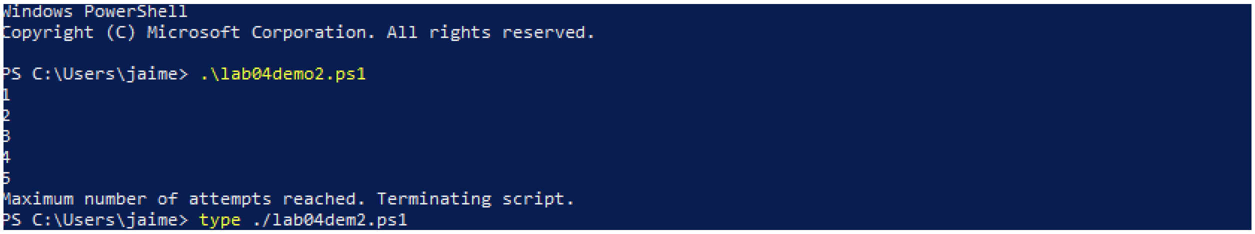
Maximum log size (KB): 1028

When maximum event log size is reached:

● Overwrite events as needed (oldest events first)

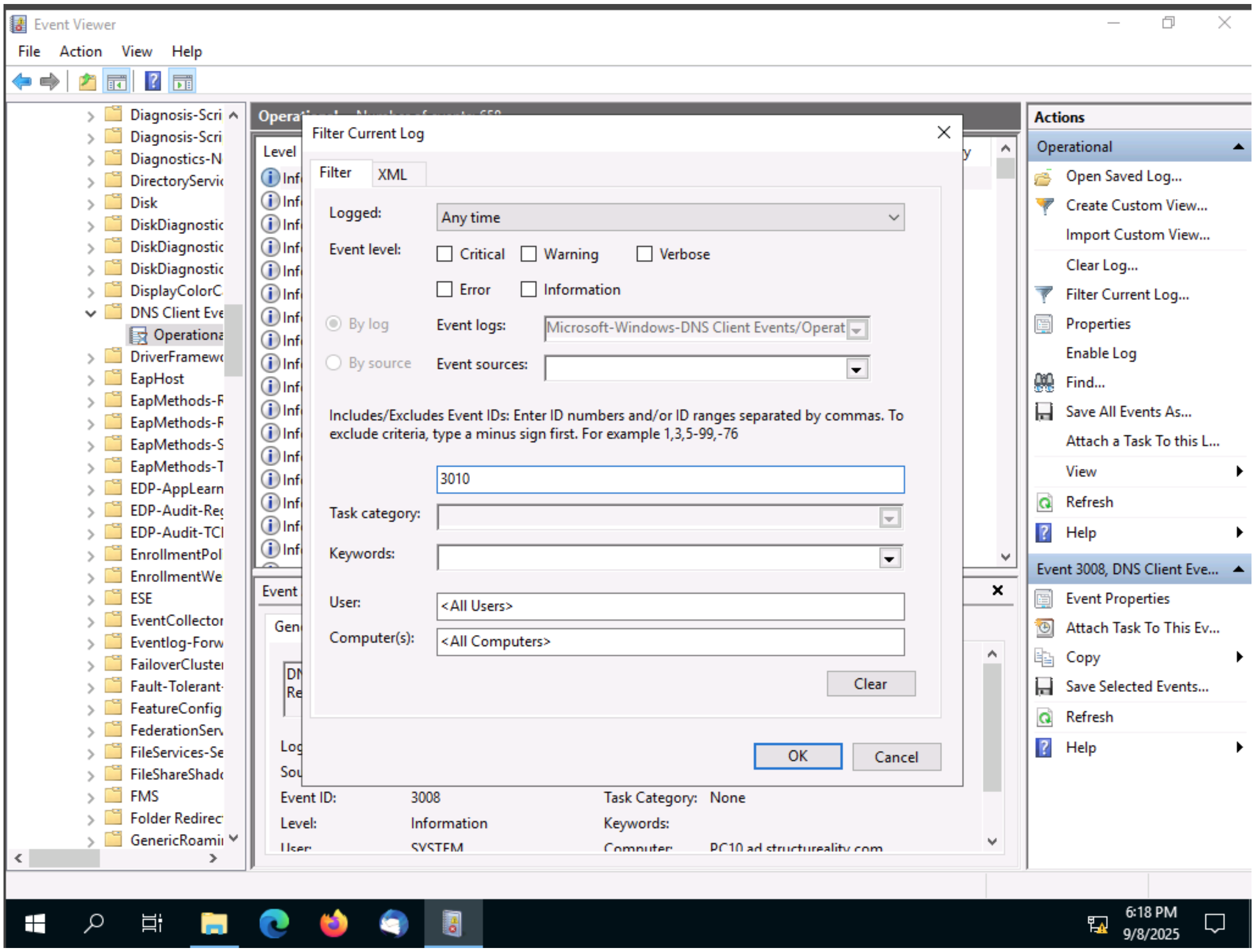
○ Archive the log when full, do not overwrite events

○ Do not overwrite events (Clear logs manually)



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\jaime> .\lab04demo2.ps1
1
2
3
4
5
Maximum number of attempts reached. Terminating script.
PS C:\Users\jaime> type ./lab04dem2.ps1
```

Go back into Event Viewer and filter the log. In this log, the Microsoft assigned Event ID of 3010 is for the initial DNS query.



Logged: Any time

Event level: □ Critical □ Warning □ Verbose □ Error □ Information

Event logs: Microsoft-Windows-DNS Client Events/Operat

Event sources:

Includes/Excludes Event IDs: Enter ID numbers and/or ID ranges separated by commas. To exclude criteria, type a minus sign first. For example 1,3,5-99,-76

3010

Task category:

Keywords:

User: <All Users>

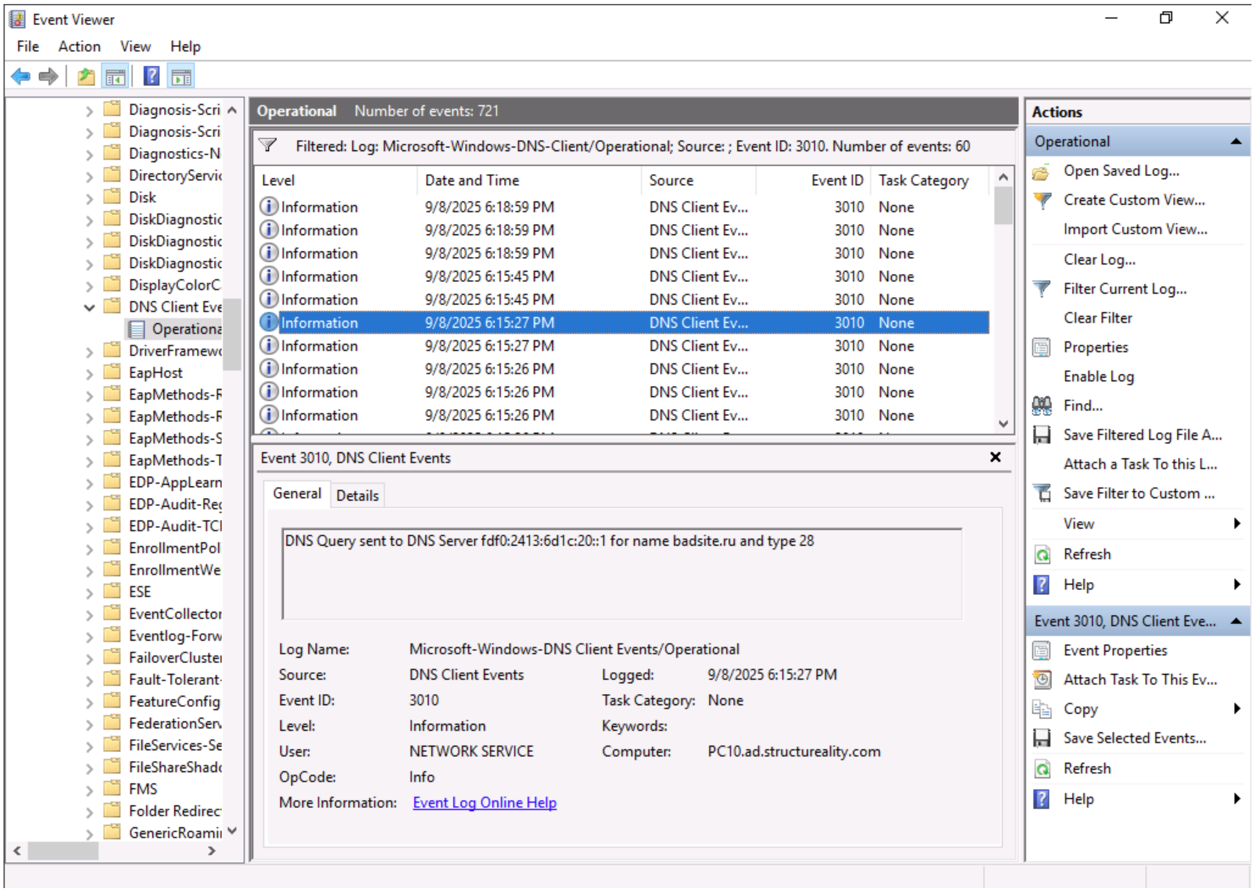
Computer(s): <All Computers>

Event ID: 3008 Task Category: None

Level: Information Keywords:

User: SYSTEM Computer: PC10.ad.structureality.com

Now, we can find the FQDN of the DNS threat.



Operational Number of events: 721

Filtered: Log: Microsoft-Windows-DNS-Client/Operational; Source: ; Event ID: 3010. Number of events: 60

Level	Date and Time	Source	Event ID	Task Category
Information	9/8/2025 6:18:59 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:18:59 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:18:59 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:45 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:45 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:27 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:27 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:26 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:26 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:26 PM	DNS Client Ev...	3010	None

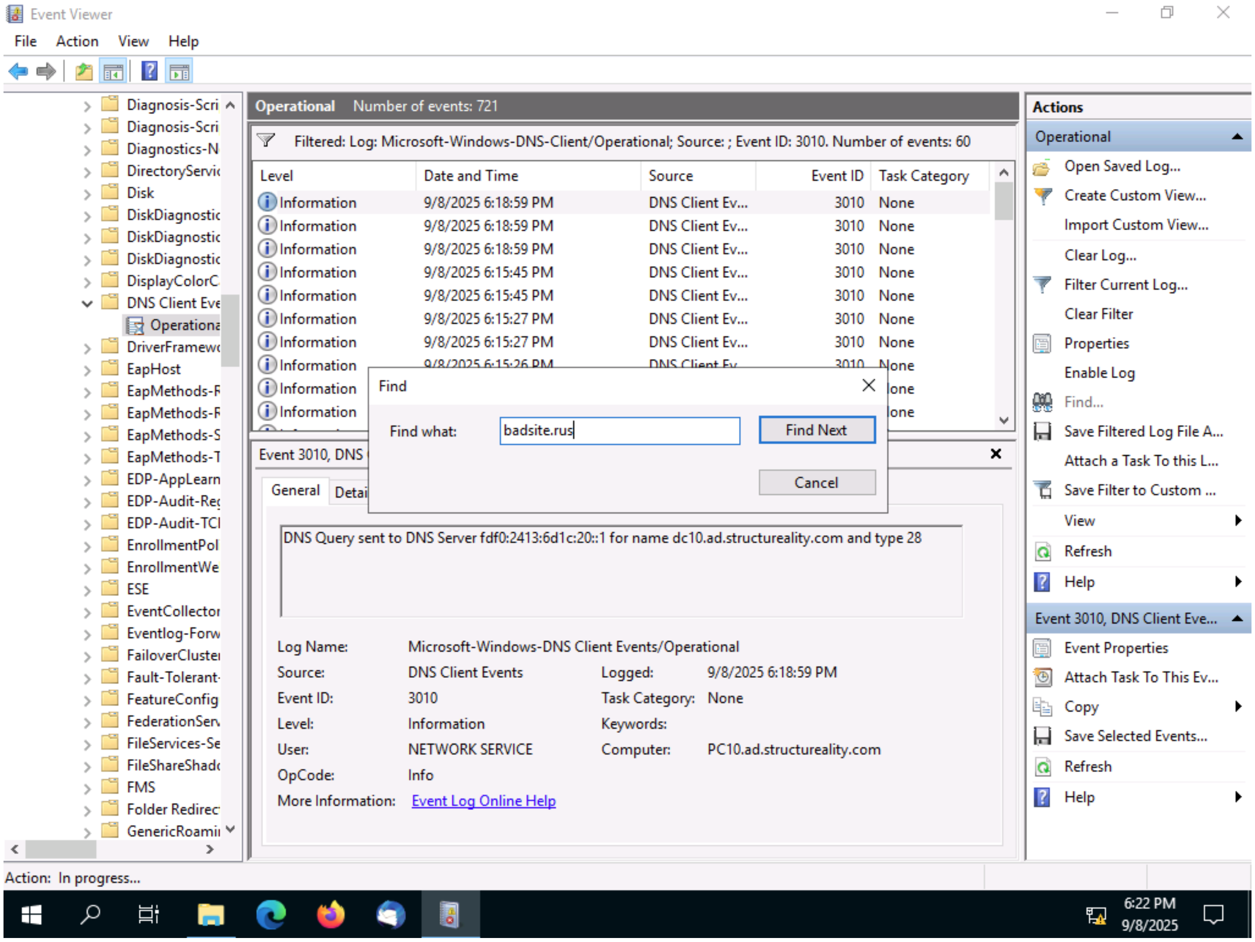
Event 3010, DNS Client Events

General Details

DNS Query sent to DNS Server fdf0:2413:6d1c:20::1 for name badsite.ru and type 28

Log Name: Microsoft-Windows-DNS Client Events/Operational
Source: DNS Client Events Logged: 9/8/2025 6:15:27 PM
Event ID: 3010 Task Category: None
Level: Information Keywords:
User: NETWORK SERVICE Computer: PC10.ad.structureality.com
OpCode: Info
More Information: Event Log Online Help

Find the next badsite.ru and keep going.



Operational Number of events: 721

Filtered: Log: Microsoft-Windows-DNS-Client/Operational; Source: ; Event ID: 3010. Number of events: 60

Level	Date and Time	Source	Event ID	Task Category
Information	9/8/2025 6:18:59 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:18:59 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:18:59 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:45 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:45 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:27 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:27 PM	DNS Client Ev...	3010	None
Information	9/8/2025 6:15:26 PM	DNS Client Ev...	3010	None
Information				lone
Information				lone

Find what: badsite.rus

Event 3010, DNS

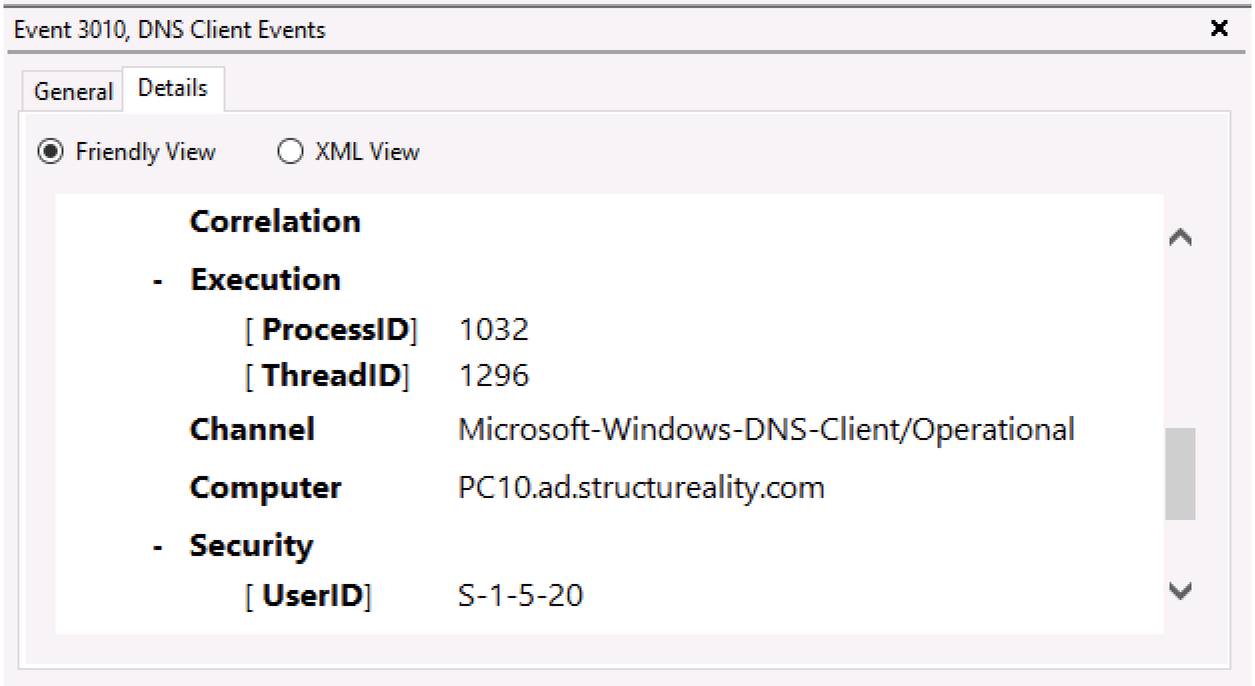
DNS Query sent to DNS Server fdf0:2413:6d1c:20::1 for name dc10.ad.structureality.com and type 28

Log Name:	Microsoft-Windows-DNS Client Events/Operational		
Source:	DNS Client Events	Logged:	9/8/2025 6:18:59 PM
Event ID:	3010	Task Category:	None
Level:	Information	Keywords:	
User:	NETWORK SERVICE	Computer:	PC10.ad.structureality.com
OpCode:	Info		
More Information:	Event Log Online Help		

Action: In progress...

We can use this to find the interval of time between each query to the malicious site. It ends up being 5 seconds between intervals. This repeated attempt to resolve a FQDN on a regular interval by unknown software is known as beaconing.

You can find the PID of the query in the details tab.



● Friendly View ○ XML View

```
    Correlation
  - Execution
       [ ProcessID]   1032
       [ ThreadID]    1296
    Channel           Microsoft-Windows-DNS-Client/Operational
    Computer          PC10.ad.structureality.com
  - Security
       [ UserID]      S-1-5-20
```

This is the PID of the process that initiated the query. If the offending process was continually running (unlike the lab04demo2.ps1 script, which only runs for about 10 seconds), then this PID could be used in a *tasklist* query to discover the process name (as performed in a previous exercise in this lab). Once you identify the offending process, then you can consider your next actions. Options for further action include determining how the offending process came to be on the system and what can be done to mitigate the issue (i.e., terminate its execution and remove it from the system).

When reviewing the output of any security tool, vulnerability scanner, or investigation, it is important to keep several issues in mind. First, you need to validate or verify vulnerabilities before initiating mitigations. False positive items do not need to be resolved, only true positive issues. Second, you need to prioritize the verified issues. This provides guidance to the security team as to the order and urgency to address the reported problems. Finally, you should make recommendations on responses or remedies when known. As a cybersecurity analyst, you may often know how to resolve specific problems you discover. Passing this information along to the security team can make their response to vulnerabilities more efficient.