

**Blue Team Level 1 Certification
(Standard)****Introduction to BT1** Welcome to Blue Team Level 1!

4 Topics

 Lab and Forum Access**SECURITY FUNDAMENTALS DOMAIN** Introduction to Security Fundamentals

1 Topic

 Soft Skills

7 Topics

 Security Controls

5 Topics 1 Quiz

 Networking 101

6 Topics 1 Quiz

 Management Principles

4 Topics 1 Quiz

PHISHING ANALYSIS DOMAIN PA1) Introduction to Emails and Phishing

7 Topics 1 Quiz

 PA2) Types of Phishing Emails

10 Topics 2 Quizzes

 PA3) Tactics and Techniques Used

12 Topics 2 Quizzes

 PA4) Investigating a Phishing Email

8 Topics 2 Quizzes

 PA5) Analysing URLs, Attachments, and Artifacts

8 Topics 1 Quiz

 PA6) Taking Defensive Actions

12 Topics 1 Quiz

 PA7) Report Writing

7 Topics 1 Quiz

 PA8) Phishing Response Challenge

3 Topics 1 Quiz

THREAT INTELLIGENCE DOMAIN TI1) Introduction to Threat Intelligence

7 Topics

 TI2) Threat Actors & APTs

6 Topics 2 Quizzes

 TI3) Operational Threat Intelligence

7 Topics 1 Quiz

 TI4) Tactical Threat Intelligence

7 Topics 1 Quiz

 TI5) Strategic Threat Intelligence

5 Topics 1 Quiz

 TI6) Malware and Global Campaigns

6 Topics 1 Quiz

DIGITAL FORENSICS DOMAIN DF1) Introduction to Digital Forensics

5 Topics

 DF2) Forensics Fundamentals

10 Topics 5 Quizzes

CMD and PowerShell For Incident Response

Blue Team Level 1 Certification (Standard) > IR3) Detection and Analysis Phase > CMD and Pow...

IN PROGRESS



This lesson will introduce you to some useful CMD and PowerShell commands to assist with incident response.

Some of the actions we can take include:

- List networking information (to gather information such as IP address, MAC, and more)
- Viewing open and listening ports (to detect backdoors and beaconing)
- Viewing running processes and their related executable files (to detect malware or backdoors)
- List all users and admins on the local system (to identify unusual accounts)
- List programs that launch at system boot (to detect malicious files)
- List services and detailed information (to identify malicious services)
- And more!



Why do we use CMD?

Although using it requires the memorization of many different commands, it can allow us to complete tasks faster than interacting with the Windows graphical user interface, and also allows us to automate tasks using batch scripts. This can help in incident response scenarios, as we can query the system for information about almost anything, from users to running processes!

CMD For Incident Response

Below we're going to cover a number of commands that may be useful for security investigations and incident response. We will also include examples so you can see what the input could look like, and how to interpret the information that is printed to the terminal. These commands should be ran as an administrator to function correctly.

```
ipconfig /all
```

Description: This command will get network configuration information from the local system, including assigned IP address and the device's MAC address.

Example: In this example we can see that we have the host name "MSEDGEWIN10", a MAC address of "00-0C-29-AA-02-FA", and IPv4 address of "192.168.125.129". We can also see that the DNS server being used for name resolution is currently "102.168.125.2".

```
Microsoft Windows [Version 10.0.17763.1397]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\IEUser>ipconfig /all

Windows IP Configuration

    Host Name . . . . . : MSEDGEWIN10
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Hybrid
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No
    DNS Suffix Search List . . . . . : localdomain

Ethernet adapter Ethernet0:

        Connection-specific DNS Suffix . . . . . : localdomain
        Description . . . . . : Intel(R) PRO/1000 MT Network Connection
```

- DF3) Digital Evidence Collection
 - 8 Topics | 1 Quiz
- DF4) Windows Investigations
 - 3 Topics | 3 Quizzes
- DF5) Linux Investigations
 - 4 Topics | 2 Quizzes
- DF6) Volatility
 - 3 Topics | 1 Quiz
- DF7) Autopsy
 - 4 Topics | 1 Quiz

SECURITY INFORMATION AND EVENT MANAGEMENT DOMAIN

- SI1) Introduction to SIEM
7 Topics | 1 Quiz
- SI2) Logging
6 Topics | 2 Quizzes
- SI3) Aggregation
2 Topics | 1 Quiz
- SI4) Correlation
6 Topics | 1 Quiz
- SI5) Using Splunk
5 Topics | 2 Quizzes

INCIDENT RESPONSE DOMAIN

- IR1) Introduction to Incident Response
 - 8 Topics | 1 Quiz
- IR2) Preparation Phase
 - 10 Topics | 2 Quizzes
- IR3) Detection and Analysis Phase
 - 7 Topics | 4 Quizzes
 - Section Introduction, Detection & Analysis
 - Common Events & Incidents
 - Using Baselines & Behavior Profiles
 - Introduction to Wireshark (GUI)
 - Introduction to Wireshark (Analysis)
 - ☒ Lab) Network Traffic Analysis
 - YARA Rules For Detection
 - ☒ Legacy Activity Threat Hunting With YARA
- IR4) Containment, Eradication, and Recovery Phase
 - 5 Topics | 1 Quiz
- IR5) Lessons Learned and Reporting
 - 7 Topics
- IR6) MITRE ATT&CK
 - 10 Topics | 2 Quizzes

BTL1 EXAM

- Exam Preparation
- Using RDP and SSH
- How to Start Your Exam

```
Physical Address . . . . . : 00-0C-29-AA-02-FA
DHCP Enabled . . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::9c5:29ff:fe6af:b128%0(Preferred)
    IPv4 Address . . . . . : 192.168.125.249(PREFERRED)
        Subnet Mask . . . . . : 255.255.255.0
        Lease Obtained . . . . . : Tuesday, October 13, 2020 7:02:10 AM
        Lease Expires . . . . . : Tuesday, October 13, 2020 7:32:10 AM
        Default Gateway . . . . . : 192.168.125.2
        DHCP Server . . . . . : 192.168.125.254
        DHCPv6 IAID . . . . . : 67111977
        DHCPv6 Client DUID . . . . . : 00-01-00-01-26-68-81-B9-00-0C-29-AA-02-FA
        DNS Servers . . . . . : 192.168.125.2
        Primary WINS Server . . . . . : 192.168.125.2
NetBIOS over Tcpip . . . . . : Enabled
```

1

Description: This command will check running processes and programs and print a list to the terminal.

Example: In the below screenshot we are presented with a list of running processes, their process identifiers (PIDs), and the memory usage in the final column. This can be helpful to identify processes that are running in the background and how much system resources they are consuming. This can be a good way to identify malware such as crypto miners which will work silently, but consume a lot of system memory to work.

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	8 K
System	4	Services	0	144 K
Registry	88	Services	0	54,492 K
sms.exe	284	Services	0	968 K
csrss.exe	384	Services	0	4,604 K
wininit.exe	488	Services	0	5,908 K
csrss.exe	496	Console	1	5,574 K
abinet.exe	504	Console	1	9,528 K
services.exe	624	Services	0	8,820 K
lsass.exe	640	Services	0	15,164 K
svchost.exe	744	Services	0	3,612 K
fontdrvhost.exe	768	Services	0	3,360 K
fontdrvhost.exe	776	Console	1	5,474 K
puclient.exe	784	Services	0	28,196 K
svchost.exe	876	Services	0	17,952 K
svchost.exe	924	Services	0	7,676 K
dum.exe	992	Console	1	155,384 K
svchost.exe	688	Services	0	9,700 K
svchost.exe	704	Services	0	11,656 K
svchost.exe	680	Services	0	5,852 K
svchost.exe	1168	Services	0	17,428 K
svchost.exe	1236	Services	0	10,856 K
WUDFHost.exe	1252	Services	0	6,876 K
svchost.exe	1264	Services	0	12,688 K
svchost.exe	1372	Services	0	10,856 K
svchost.exe	1296	Services	0	5,488 K
svchost.exe	1348	Services	0	9,400 K
svchost.exe	1464	Services	0	7,440 K
Memory Compression	1488	Services	0	93,352 K
svchost.exe	1516	Services	0	8,040 K
svchost.exe	1536	Services	0	14,044 K
svchost.exe	1612	Services	0	6,948 K
svchost.exe	1636	Services	0	8,192 K
svchost.exe	1692	Services	0	11,568 K
svchost.exe	1792	Services	0	8,244 K
svchost.exe	1816	Services	0	9,904 K
svchost.exe	1896	Services	0	9,308 K
svchost.exe	1952	Services	0	8,076 K
svchost.exe	1964	Services	0	5,736 K
svchost.exe	1972	Services	0	8,448 K
svchost.exe	1048	Services	0	11,592 K
svchost.exe	1756	Services	0	9,404 K
spoolv.exe	2166	Services	0	11,776 K
svchost.exe	2236	Services	0	17,708 K
svchost.exe	2276	Services	0	7,080 K
svchost.exe	2284	Services	0	7,732 K
svchost.exe	2496	Services	0	13,488 K

```
wmic process get description, executablepath
```

Description: This command will display running processes and the associated binary file that was executed to create the process.

Example: As the description states we are able to view running processes and the executable file that initiated them. Look just below halfway down the list and you'll find Discord.exe on the left-hand side (process name). To the right on the same row we can see that Discord was launched from C:\Users\IEUser\AppData\Local\Discord\app-0.0.307\Discord.exe – we now have the full file path! We can use this command to identify unusual process names and identify where the executable file is so we can analyze it. Process that are running out of unusual locations such as /tmp/ and /Downloads/ are definitely worth investigating further!

```
C:\ Command Prompt
!lmao.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
ShellExperienceHost.exe
explorer.exe
C:\Windows\Explorer.EXE
C:\Windows\System32\svchost.exe
C:\Windows\SystemApps\ShellExperienceHost_cobinrgwv\ShellExperienceHost.exe
```

net user

Description: This command will print a list of all system users to the terminal.

Example: Using this cmd command we have printed all local system users, regardless of usergroup, to the terminal, in this case we can see:

- Administrator
 - DefaultAccount
 - Guest
 - Jeff S
 - MarkA
 - sshd
 - SteveE
 - WDAGUtilityAccount

```
C:\ Select Command Prompt  
C:\Users\IEUser>net user  
  
User accounts for \\MSEdgeWIN10  
  
-----  
Administrator           DefaultAccount       Guest  
Jeff S                 MarkA               sshd  
SteveE                 WDAGUtilityAccount  
The command completed successfully.  
  
C:\Users\IEUser>
```

```
net localgroup administrators
```

Description: This command will list all users that are in the administrators user group.

Example: In the below screenshot we can see all administrator accounts on this system. In this case it is the following:

- Administrator
 - Jeff S
 - MarkA

```
C:\Users\IEUser>net localgroup Administrators  
Alias name      Administrators  
Comment         Administrators have complete and unrestricted access to the computer/do  
main  
  
Members  
  
-----  
Administrator  
Jeff S  
MarkA  
SteveE  
The command completed successfully.
```

```
C:\Users\IEUser>
```

We can replace "administrators" with any local group that we want to enumerate. To see a list of all groups, use the command `net localgroup`. If you want to search for users in a group that includes spaces, you'll need to run your commands like this: `net localgroup "Remote Desktop Users"`.

```
sc query | more
```

Description: This command will list all services and detailed information about each one.

```
netstat -ab
```

Description: This command will list open ports on a system, which could show the presence of a backdoor.



Why do we use PowerShell?

PowerShell is amazing, and chances are you'll use it a lot while working in the security industry. We can automate complex tasks, use it for offensive security purposes, or during security investigations to get more information about a user or system. For incident response we can use it similarly to CMD, but we can often retrieve much more information.

PowerShell For Incident Response

```
Get-NetIPConfiguration and Get-NetIPAddress
```

Description: Similar to ifconfig in CMD, we can use the two above commands to get network-related information from the system.

Example:

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

PS C:\Users\BTLOTest> Get-NetIPConfiguration

Description          : Gto4 Configuration
State               : Started
AutoSharing         : Default
RelayName           : Gto4.ipv6.microsoft.com.
RelayState          : Default
ResolutionIntervalSeconds : 1000

PS C:\Users\BTLOTest> Get-NetIPAddress

IPv4Address        : fe80::ed2d:47bf:9516:9248%17
InterfaceIndex     : 1
InterfaceAlias     : Loopback Pseudo-Interface 1
AddressFamily      : IPv4
Type               : Unicast
PrefixLength       : 64
PrefixOrigin       : WellKnown
SuffixOrigin       : Link
AddressState       : Preferred
ValidLifetime      : Infinite {{TimeSpan}:.MaxValue}
PreferredLifetime : Infinite {{TimeSpan}:iMaxValue}
SkipNsSource       : False
PolicyStore         : ActiveStore

IPv6Address        : ::1
InterfaceIndex     : 1
InterfaceAlias     : Loopback Pseudo-Interface 1
AddressFamily      : IPv6
Type               : Unicast
PrefixLength       : 128
PrefixOrigin       : WellKnown
SuffixOrigin       : Link
AddressState       : Preferred
ValidLifetime      : Infinite {{TimeSpan}:.MaxValue}
PreferredLifetime : Infinite {{TimeSpan}:iMaxValue}
SkipNsSource       : False
PolicyStore         : ActiveStore
```

Get-LocalUser

Description: Using the above command we can list all local users on the system.

Example:

Name	Enabled	Description
Administrator	True	Built-in account for administering the computer/domain
True	True	
False	False	A user account managed by the system.
True	True	Built-in account for guest access to the computer/domain
True	True	
True	True	
WDAUtilityAccount	False	A user account managed and used by the system for Windows Defender Application Guard scen...

Get-LocalUser -Name BTLO | select *

Description: We can provide a specific user to the command to only get information about them. Piping (|) the results to a "select" with a wildcard (*) will give us all of the properties for the command, providing us with valuable information about the account. This can be extremely useful for us as incident responders, especially when we find local accounts that do not expire or have passwords that don't expire.

Example:

Windows PowerShell	
PS C:\Users\BTLOTest> Get-LocalUser -Name BTLO select *	
AccountExpires	:
Description	:
Enabled	: True
FullName	: BTLOPlayer
PasswordChangeableDate	
PasswordDoesntExpire	
UserMayChangePassword	: True
PasswordRequired	: True
PasswordLastSet	: 3/1/2021 10:56:48 PM
LastLogon	: 12/9/2021 10:52:28 PM
Name	: BTLO
SID	: S-1-5-21-4001622725-2027095096-2530479061-1008
PrincipalSource	: Local
ObjectClass	: User

Get-Service | Where Status -eq "Running" | Out-GridView

Description: The above command let's us quickly identify running services on the system. By piping (|) the command to Out-GridView, we are telling PowerShell to show us the results in a nice windows, which is much easier to work with than outputting the results to the PowerShell window.

Example:

Windows PowerShell		
PS C:\Users\BTLOTest> Get-Service where Status -eq "Running" Out-GridView		
PS C:\Users\BTLOTest>		
Status	Name	DisplayName
Running	gencrc	Group Policy Client
Running	ghiservic	IP Helper
Running	KeyIso	CNG Key Isolation
Running	LanmanServer	Server
Running	LanmanWorkstation	Workstation
Running	LicensingManager	Windows License Manager Service
Running	lmhosts	TCP/IP NetBIOS Helper
Running	LSM	Local Security Manager
Running	Modem	Windows Defender Firewall
Running	MSTIC	Distributed Transaction Coordinator
Running	NtService	Network Connection Broker
Running	netprof	Network List Service
Running	nginstart	nginstart
Running	NlaSvc	Network Location Awareness
Running	ni	Network Store Interface Service
Running	PaSvc	Program Compatibility Assistant Service
Running	PlugPlay	Plug and Play

Get-Process | Format-Table View priority

Description: Another great command is the ability to group running processes by their priority value. Using the above command we can see the process name, the process ID (PID), and other information, where different priority ratings are grouped into tables.

Example:

ProcessName	Id	HandleCount	WorkingSet64
	--	-----	-----

amazon-ssm-agent	2448	148	13619200
ApplicationFrameHost	5688	264	25161728
cmd	3336	79	4214784
cmd	5896	75	4218880
conhost	2968	152	12800000
conhost	3508	120	10940416
conhost	3744	142	12541952
PriorityClass: Normal			
ProcessName	Id	HandleCount	WorkingSet64
---	--	-----	-----
conhost	4052	259	20770816
conhost	5100	151	12779520
conhost	6672	251	20918272
csrss	396	541	5677956
csrss	472	324	5713920
csrss	6544	315	10792960
csrss	7532	296	10838016
ctfmon	2088	365	15462400
ctfmon	5492	387	15527936
ctfmon	7712	369	15912960

```
Get-Process -Id 'idhere' | Select *
```

Description: We can collect specific information from a service by including the name in the command (-Name 'namewhere') or the Id, as shown above and below. Piping to Select * provides us with all the properties.

Example:

```
PS C:\Users\BTLOTest> Get-Process -Name 'cmd'
Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
---- -- -- -- -- -- -
79 5 2664 4116 3336 0 cmd
75 5 2368 4120 5896 2 cmd

PS C:\Users\BTLOTest> Get-Process -Id '5896' | Select *

Name : cmd
Id : 5896
PriorityClass :
FileVersion :
HandleCount : 75
WorkingSet : 4218880
PagedMemorySize : 2424832
PrivateMemorySize : 2424832
VirtualMemorySize : 58249216
TotalProcessorTime :
SI : 2
Handles : 75
VM : 2203376472064
WS : 4218880
PM : 2424832
NPM : 5064
Path :
Company :
CPU :
ProductVersion :
Description :
Product :
```

```
Get-ScheduledTask
```

Description: Similar to Services, Scheduled Tasks are often abused and utilized a common persistence technique. With the above command we can list tasks that are set to run after certain conditions are met.

Example:

TaskPath	TaskName	State
\Microsoft\Windows\Application Experience\VerifierifierInstall	VerifierVerifierInstall	Ready
\Microsoft\Windows\Application Experience\VerifierifierDaily	VerifierVerifierDaily	Ready
\Microsoft\Windows\Application Experience\VerifierVerifierCertStoreCheck	VerifierVerifierCertStoreCheck	Disabled
\Microsoft\Windows\AutoLock\AutoLock	AutoLock	Ready
\Microsoft\Windows\BitLocker\BitLocker Encrypt All Drives	BitLocker BitLocker Encrypt All Drives	Ready
\Microsoft\Windows\BitLocker\BitLocker MDM policy Refresh	BitLocker BitLocker MDM policy Refresh	Ready
\Microsoft\Windows\Bluetooth\UninstallDeviceTask	UninstallDeviceTask	Disabled
\Microsoft\Windows\BrokeredInterfaceInitializationTask	BrokeredInterfaceInitializationTask	Ready

```
Get-ScheduledTask -TaskName 'PutANameHere' | Select *
```

Description: We can dig deeper by specifying the task we're interested in, and retrieving all properties for it.

Quizzes

Lab) CMD and PowerShell

[Previous Topic](#)

[Back to Lesson](#)

[Privacy & Cookies Policy](#)

