

# Purple Teaming Cloud Identity: Simulation Labs for Red and Blue teams

BSidesDFW 2022

Jason Ostrom

November 5, 2022

# Hey What's up, I'm Jason

- Builder of things @SANS Institute
- Certified Instructor @SANS, SEC588 → “Cloud Penetration Testing”
- Founder, Principal @Stora Security
- Community tool author
- Family, hockey, football
- I'm from DFW and I graduated from HS here. Hi Mom!



```
(kali㉿kali)-[~]
$ voiphopper -V
VoIP Hopper 2.04
Copyright (C) 2012 Jason Ostrom <jpo@pobox.com>
Location: http://voiphopper.sourceforge.net
```



**New tool!** Cloud “edge” bug bounty  
and recon tool  
[github.com/iknowjason/edge](https://github.com/iknowjason/edge)



# Agenda for today

- Purple Teaming
- Infrastructure as Code (IaC)
- Labs Overview
- BlueCloud & PurpleCloud
- Demo

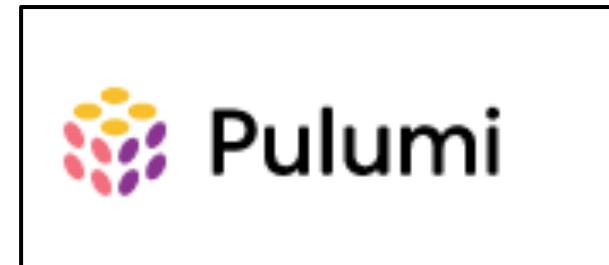


# Infrastructure as Code (IaC)



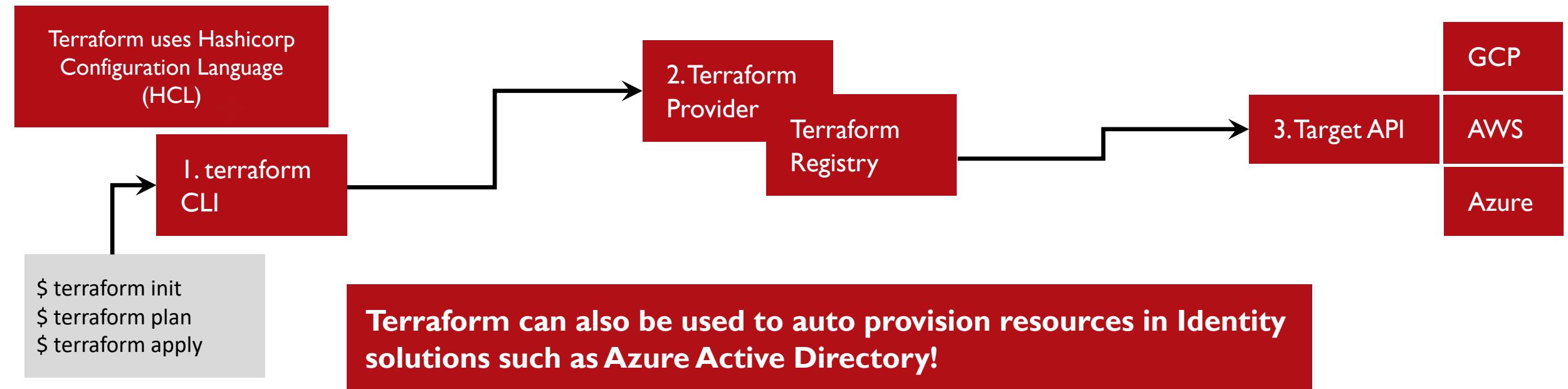
# Infrastructure as Code (IaC)

- **Infrastructure as Code** is the managing and provisioning of infrastructure automated through code instead of manually.
- Configuration is code stored in a VCS (Github, Gitlab)
- Declarative, tracking desired state
- Benefits: Speed, consistency, repeatability, lower cost
- Each cloud provider has their own IaC service
- Two popular multi-cloud platform tools: **Pulumi & Terraform**



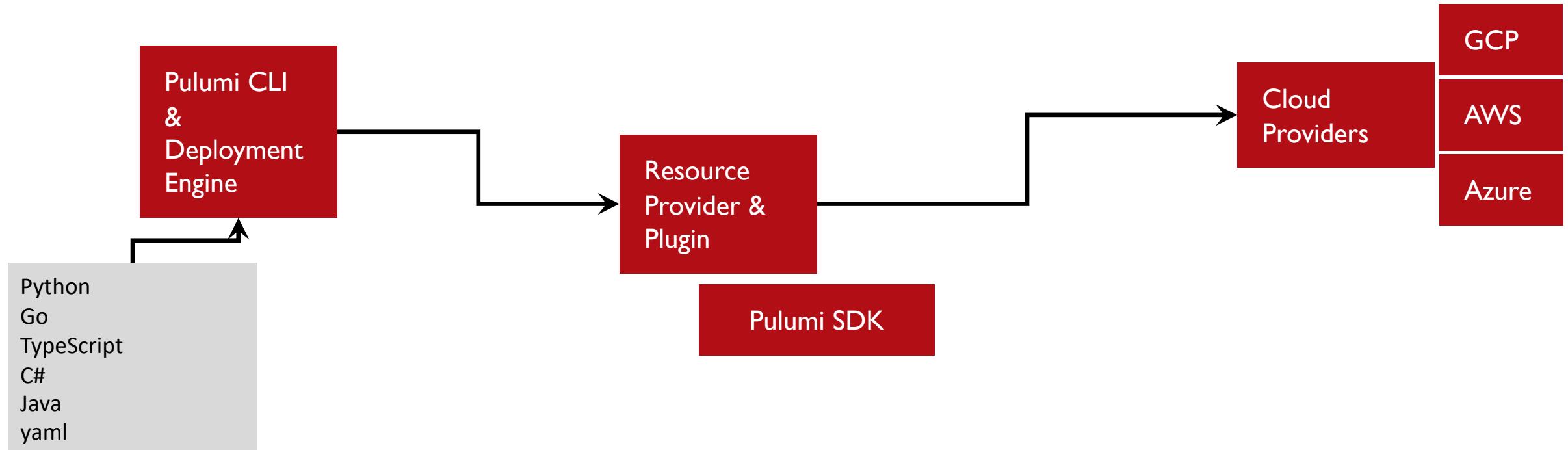
# Terraform Overview

- **Terraform** is a free, universal, and popular IaC tool that can manage infrastructure with declarative files
  - Build, change, and version infrastructure in AWS, GCP, Azure
  - Providers: plugins that talk to API for different cloud providers and can provision Infrastructure, DNS, SaaS services, Kubernetes



# Pulumi Overview

- **Pulumi** is a free and universal IaC tool that leverages existing programming languages to interact with cloud resources through the Pulumi SDK.
  - Use Python, Go, JavaScript to build, change, and version infrastructure in AWS, GCP, Azure
  - Also uses the desired state model like Terraform



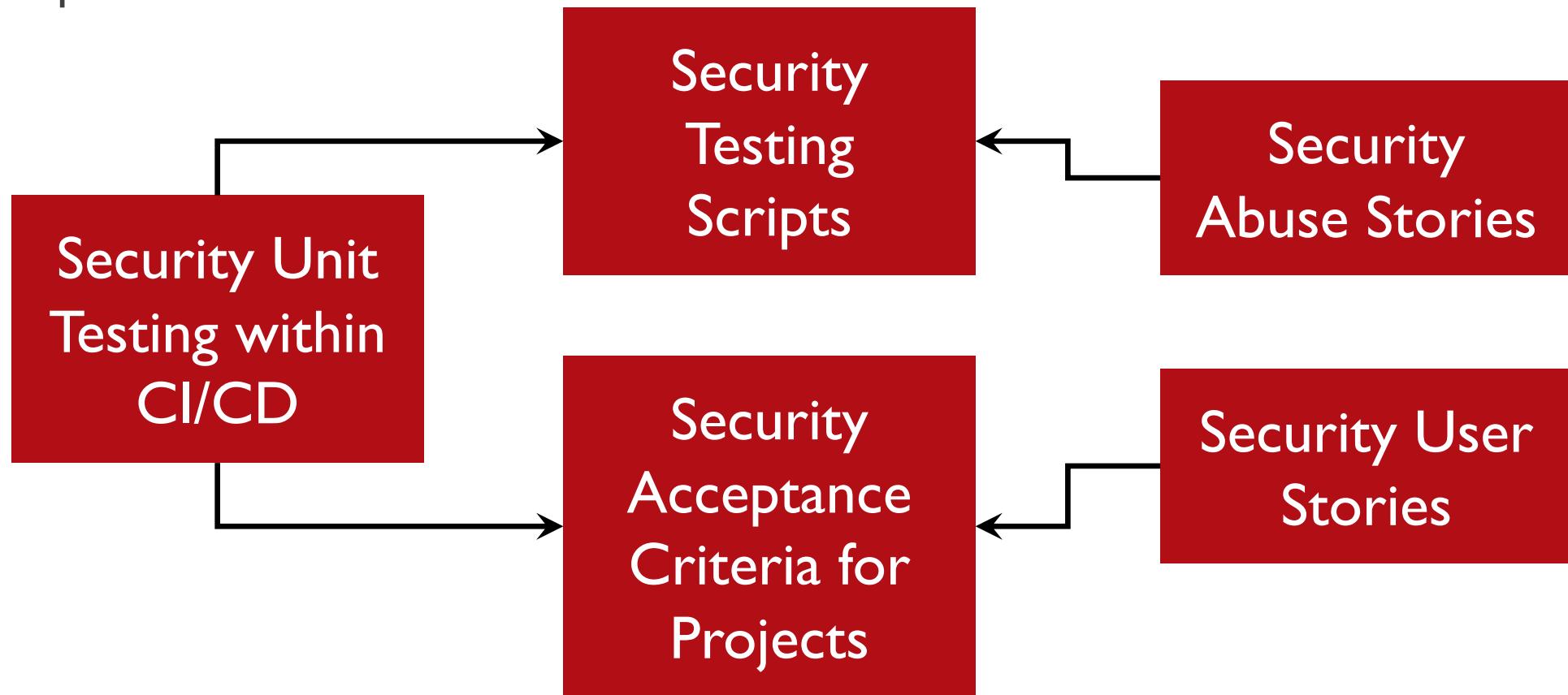
# Ansible Overview



- **Ansible** is a post-deployment Configuration Management (CM) tool
- Run YAML Ansible Playbooks using the Ansible tool for automation
- Pushes out small programs called modules
- Usually used after an OS has been deployed, for configuration of services
- Agentless: Uses SSH or WinRM to remotely access Windows or Linux systems
- Can be used to push changes across a fleet of hundreds or thousands of systems for IT Automation
- Ansible can be used along with Terraform for cloud engineering use cases

# Security as Code (SaC)

- Using code within a DevOps SDLC environment to help meet security requirements.

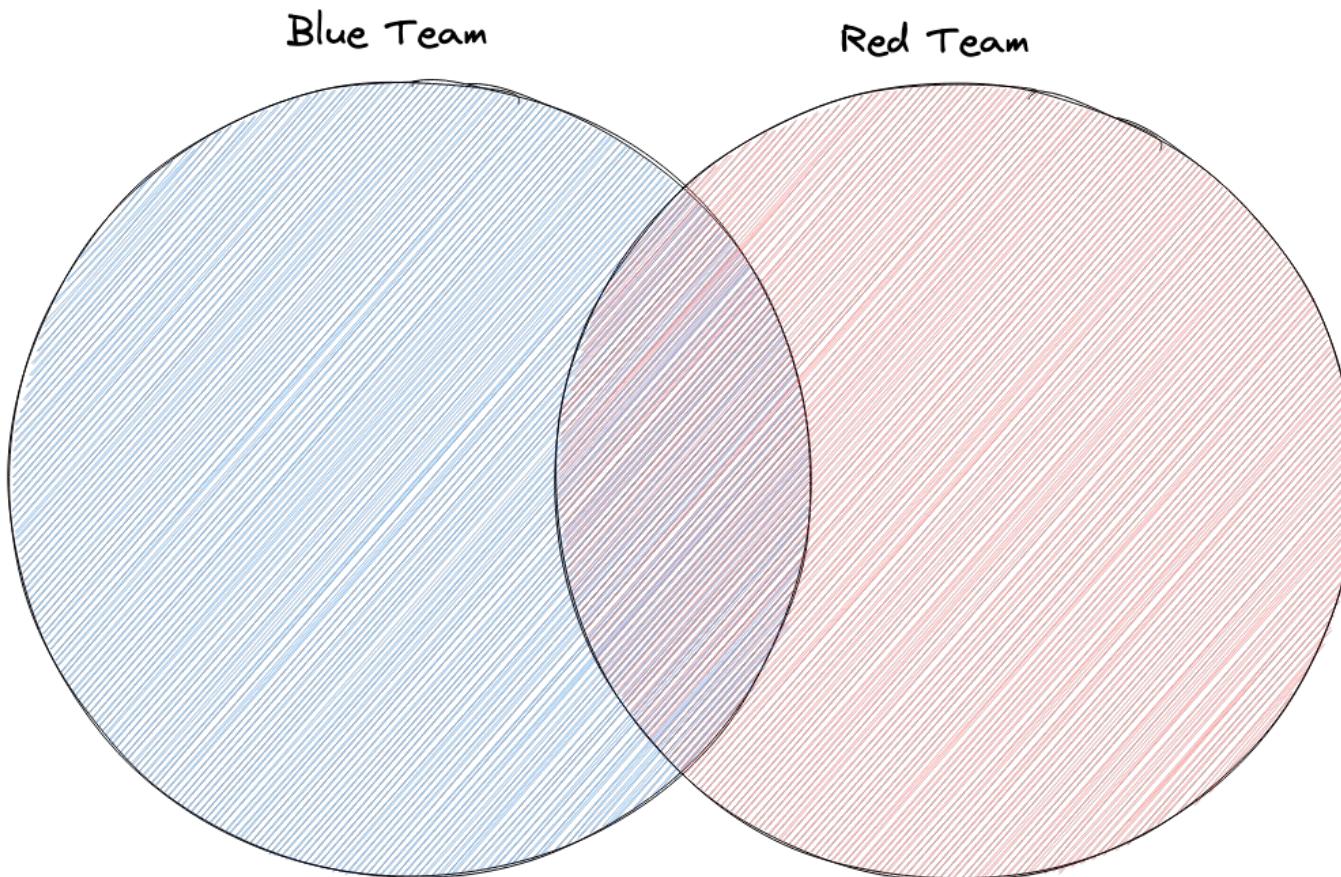


# Purple Teaming



# Purple Teaming Overview (1)

- ***Purple Teaming:*** Red and Blue teams collaborating as a team to make defense better



# Purple Teaming Overview (2)

- Adversary Emulations run, from collaboration comes improvements:
- Adding better log sources
- Log Enrichment
- Improve Detection Engineering
- Process improvements
- Training for blue team

# Purple Teaming Overview (3)

Home > Blog > Shifting from Penetration Testing to Red Team and Purple Team



Jorge Orchilles

## Shifting from Penetration Testing to Red Team and Purple Team

Penetration Testing to Red Team is mentality. Red Team is 'the practice of looking at a problem or situation from the perspective of an adversary'.

Home > Blog > Building an Internal Red Team? Go Purple First

March 17, 2022



Jorge Orchilles

## Building an Internal Red Team? Go Purple First

If you are asked to build an internal red team program today, start with a Purple Team collaboration across stakeholders early on.



Jorge Orchilles

April 11, 2022

<https://www.sans.org/blog/shifting-from-penetration-testing-to-red-team-and-purple-team/>

<https://www.sans.org/blog/building-internal-red-team-go-purple-first/>

<https://www.sans.org/blog/purple-teaming-threat-informed-detection-engineering/>

## Purple Teaming and Threat-Informed Detection Engineering

May 24, 2022

# Labs Overview



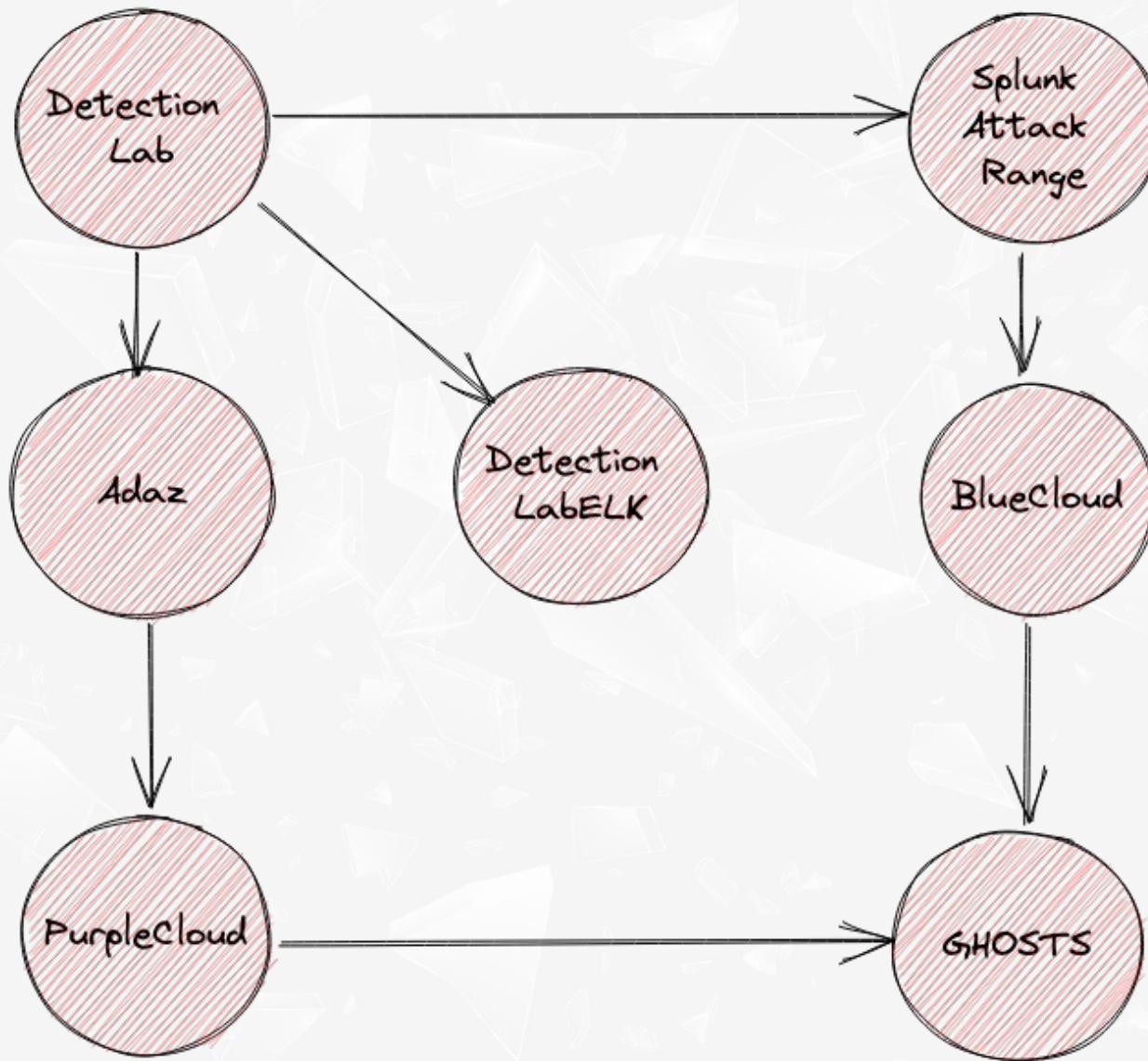
# Why Simulation Labs?

- Learning a new technology
- Detection Engineering labs for properly instrumenting logs to detect attacks
- "**Detection as Code**" as an ideal state
- Training Red and Blue teams, experimenting with a specific tool or technique against a simulated EDR endpoint or AD environment
- **Purple Teaming** exercises
- R&D security research
- Malware lab
- Fun!

Purple Teaming: To test and improve people, process, and technology. In Cyber Ranges, the focus is on training people and improving technology. The Detection Engineering process is running emulations in the Cyber Range to improve logging and technology. These changes eventually go into production.

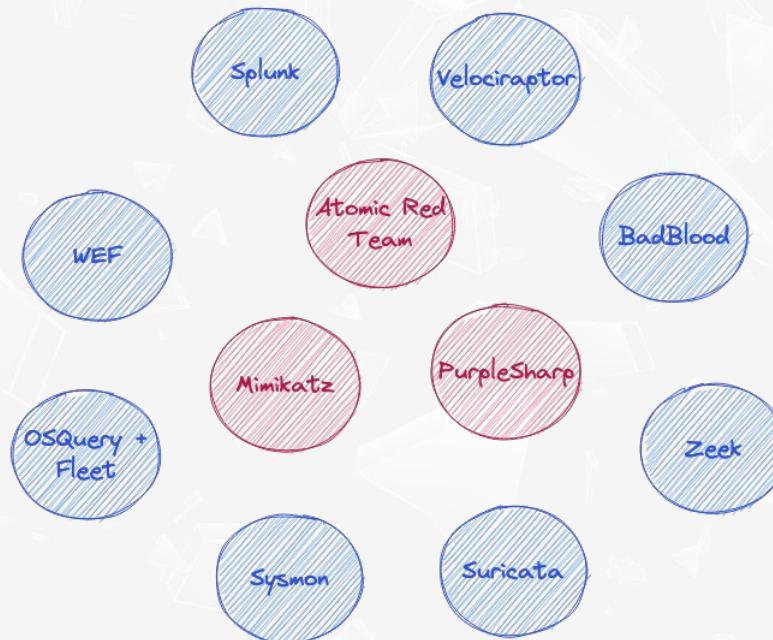


# Overview of Cyber Ranges



# DetectionLab

- Detection Engineering lab created by Chris Long automatically configured with logging best practices
  - <https://detectionlab.network>
- Great for endpoint security testing & logging research
- Packer, Vagrant, Powershell, Ansible, and Terraform scripts (AWS, Azure)



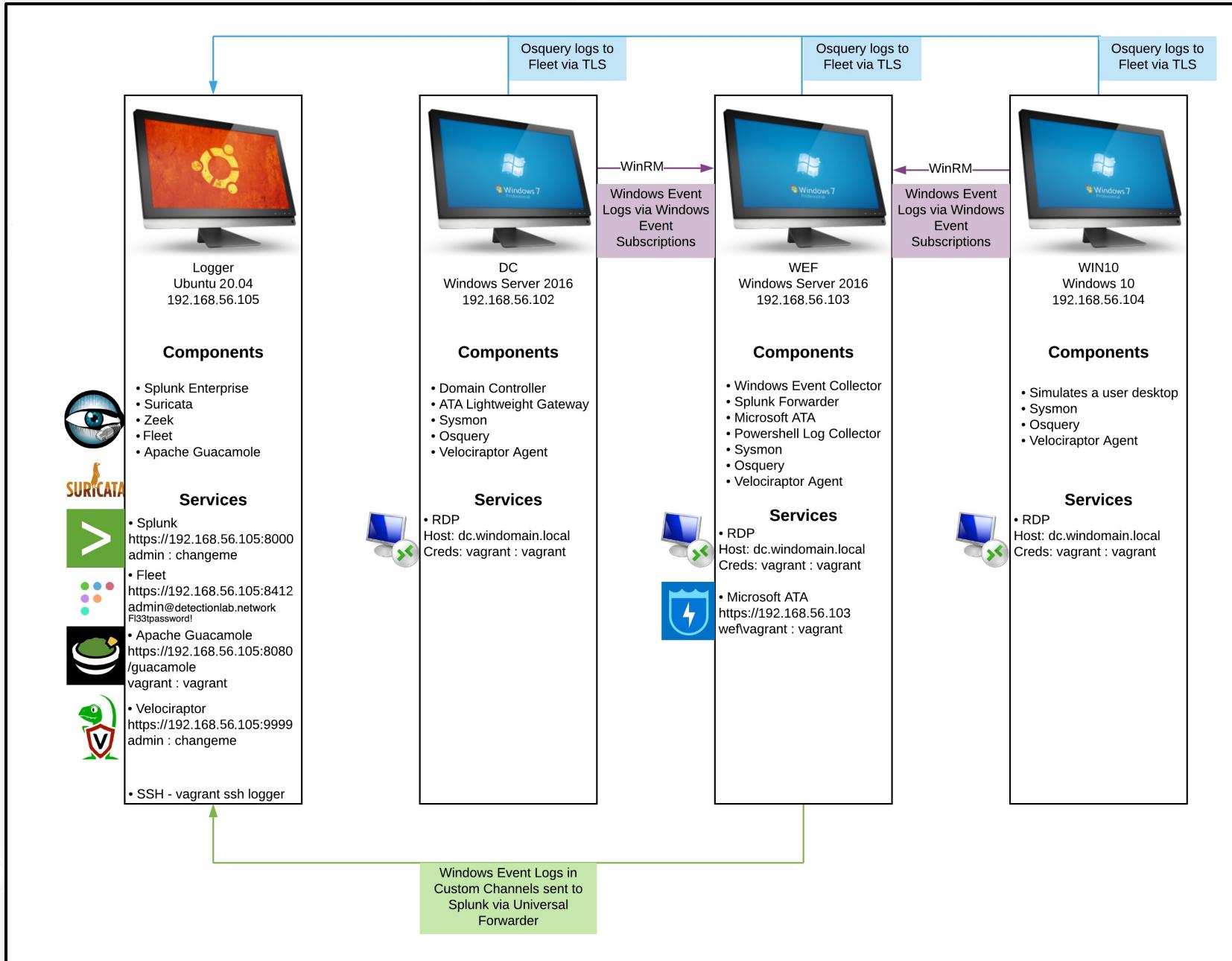
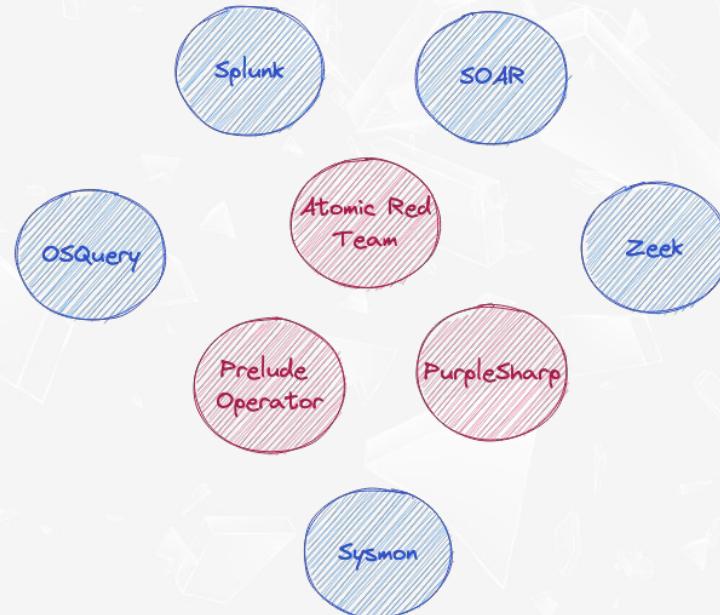


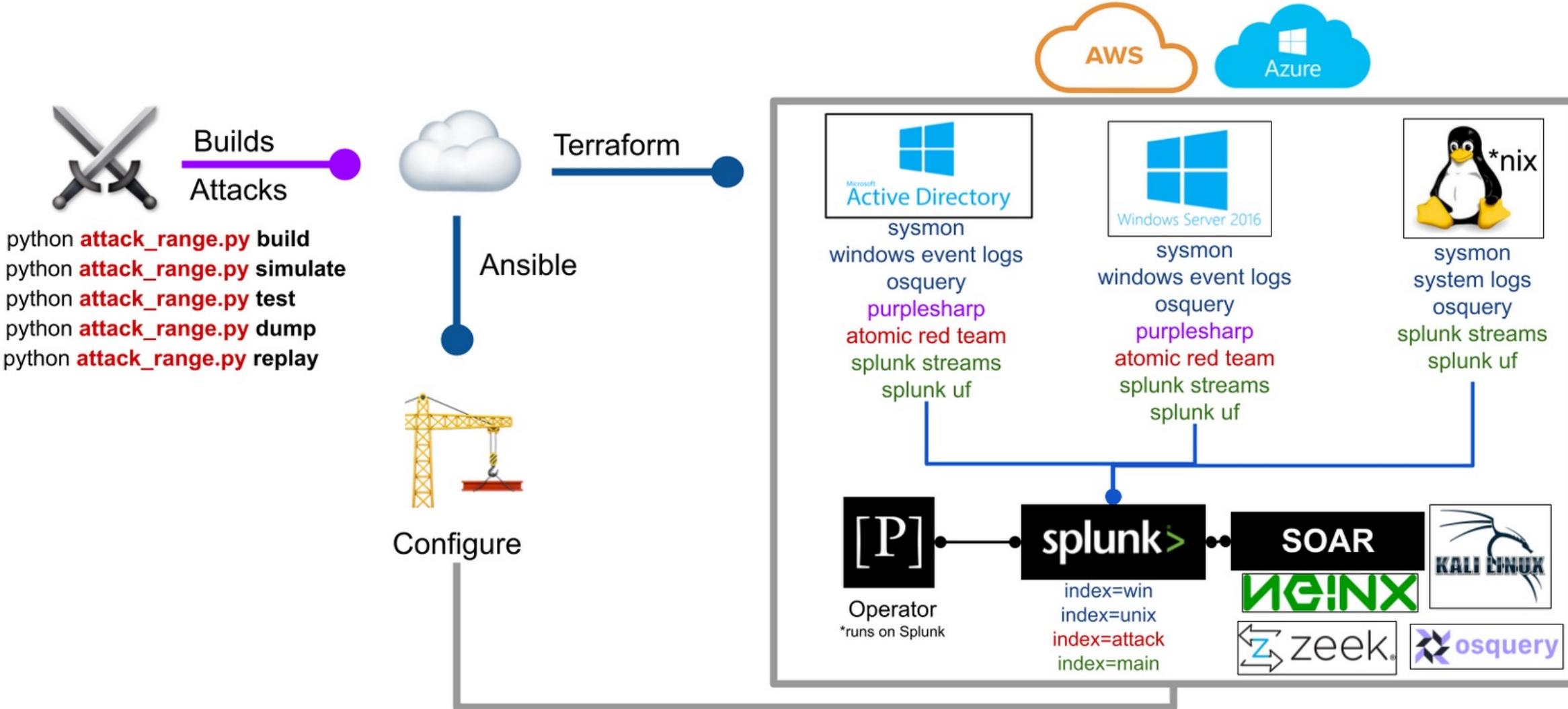
Image Source: detectionlab.network

# Splunk Attack Range



- Detection development platform featuring Splunk
- [https://github.com/splunk/attack\\_range](https://github.com/splunk/attack_range)
- Creates a small lab with python scripts for remote management of attack simulations using Atomic Red Team or PurpleSharp
  - Docker container; AWS, Azure; Interactive CLI creates a custom range configuration





# Adaz: Active Directory Hunting Lab in Azure

- Easily spin up a customizable Active Directory lab in Azure with domain-joined workstations
  - <https://github.com/christophetd/Adaz>
  - Author: Christophe Tafani-Dereeper
- Use Cases: Detection engineering, Learn or test Active Directory
- ELK Server: Logs forwarded to ELK
- Other nice features
  - A Domain configuration file in YAML that is very easy to customize
  - Ansible playbooks for pushing AD changes or OS patches after deployment
  - Windows Event Forwarding (WEF) with audit policies
  - Sysmon
  - Add multiple Windows 10 workstations with Domain Join



# Adaz Architecture

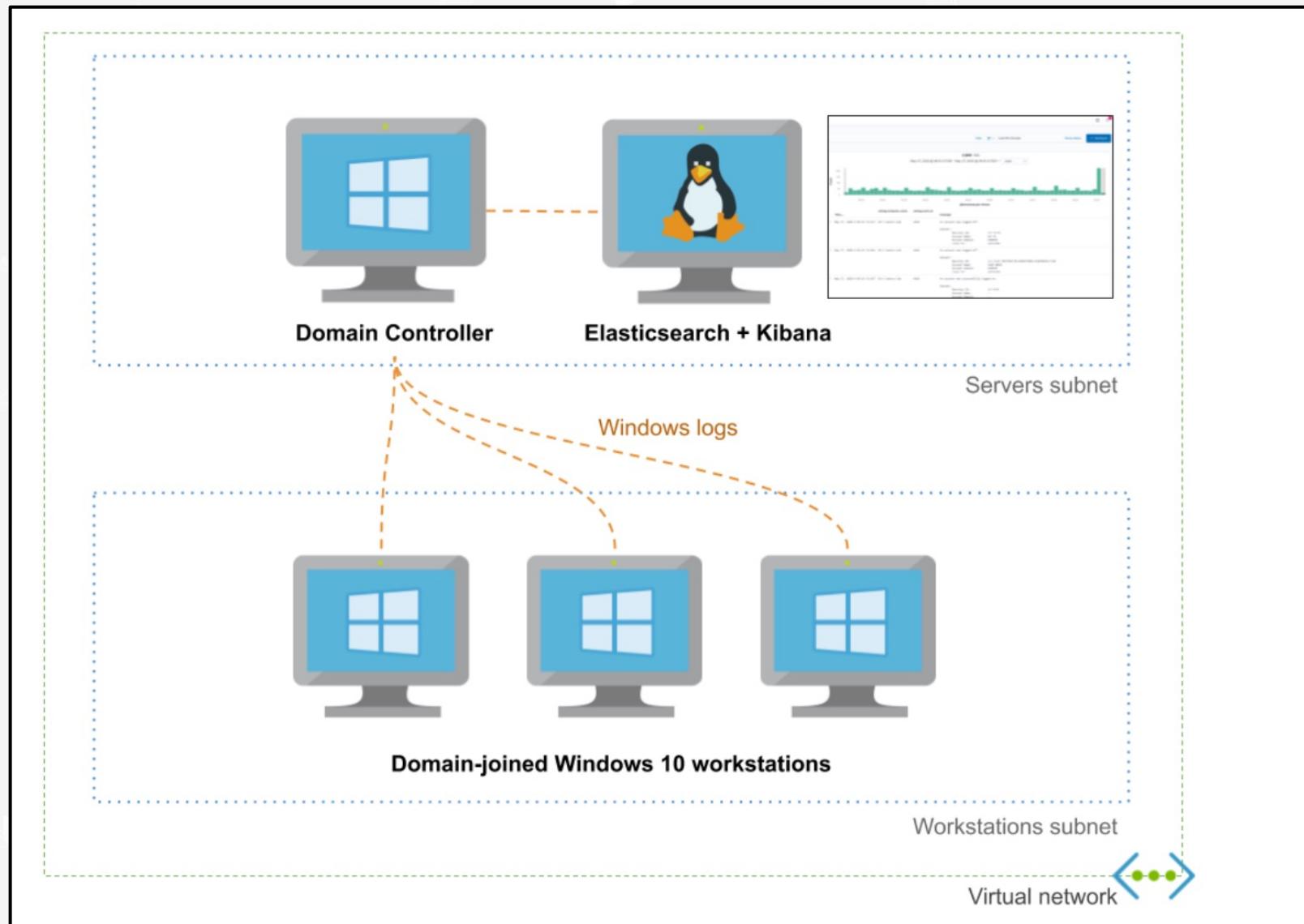


Image Source: [github.com/christophetd/Adaz](https://github.com/christophetd/Adaz)

# GHOSTS: User Simulation Framework (1)

- A user simulation framework for complex, realistic NPC orchestration
- **NPC:** Non-player characters: Realism of users and their behavior running applications on an enterprise network
- Test skills and train network defenders with real NPC players operating on the network creating static and background noise



Image Source: <https://github.com/cmu-sei/GHOSTS>

# GHOSTS: User Simulation Framework (2)

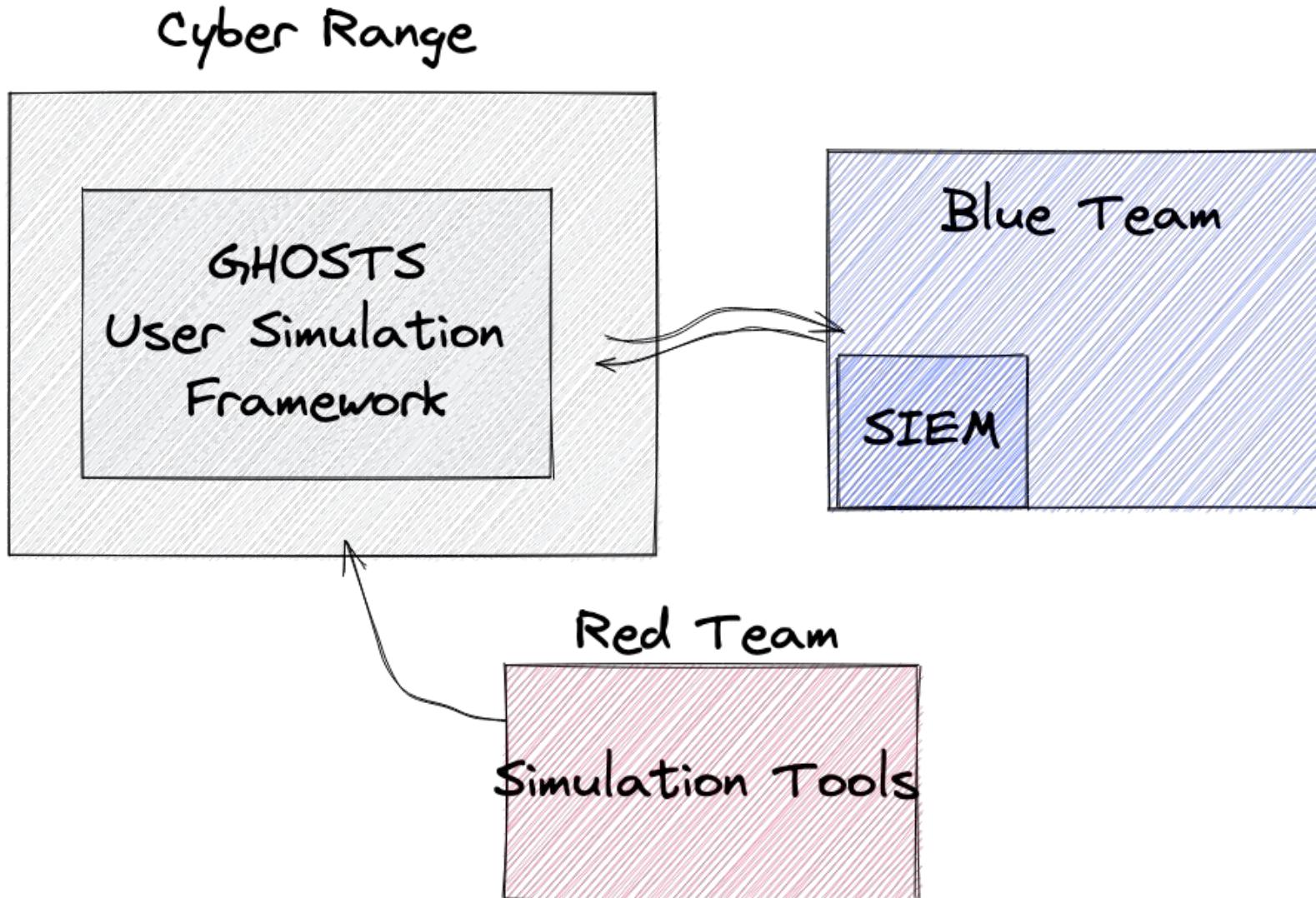
- Grafana Dashboards and API
- GHOSTS Windows client binary runs user behavior based on JSON files
- Created by Carnegie Mellon University Software Engineering Institute:  
<https://github.com/cmu-sei/GHOSTS>
- GHOSTS is a great tool for enhancing any Cyber Range with realistic, user behavior



Image Source: <https://github.com/cmu-sei/GHOSTS>



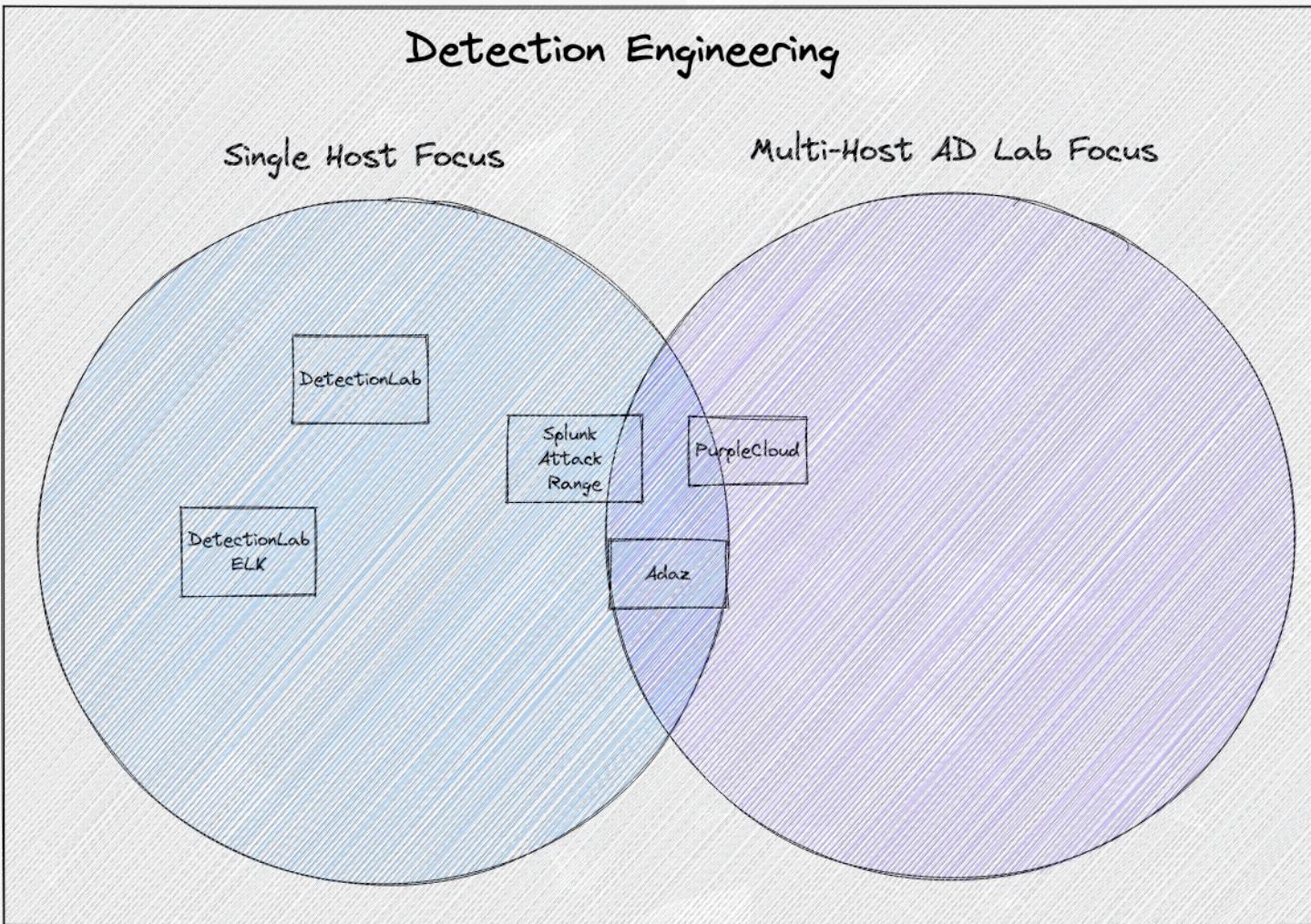
# GHOSTS Use Case with Simulation Labs



# BlueCloud + PurpleCloud



# Range Types by Focus Area



Detection Engineering transforms an idea of how to detect a specific condition or activity into a concrete description of how to detect it.  
Credit: Florian Roth

Detection Engineering is used to run emulations that improve logging and technology. The Ranges that include multi-host AD will better inform complete attack coverage, exploiting trust relationships between domain joined systems and authenticated sessions.



# Labs Evolution



Jason Ostrom

Oct 25, 2020 · 9 min read · [Listen](#)



## Building Azure Cyber Ranges for Learning and Fun

Advance your Cyber Security skills and get your Azure Security Engineer Associate certification

### Overview

Research shows there is a Cybersecurity skills shortage that is growing worse ([Oltsik, 2020](#)). Sadly, we've grown accustomed to hearing news of companies falling victim to data breaches.

Azure HELK

Azure  
Velociraptor

BlueCloud

PurpleCloud



# BlueCloud

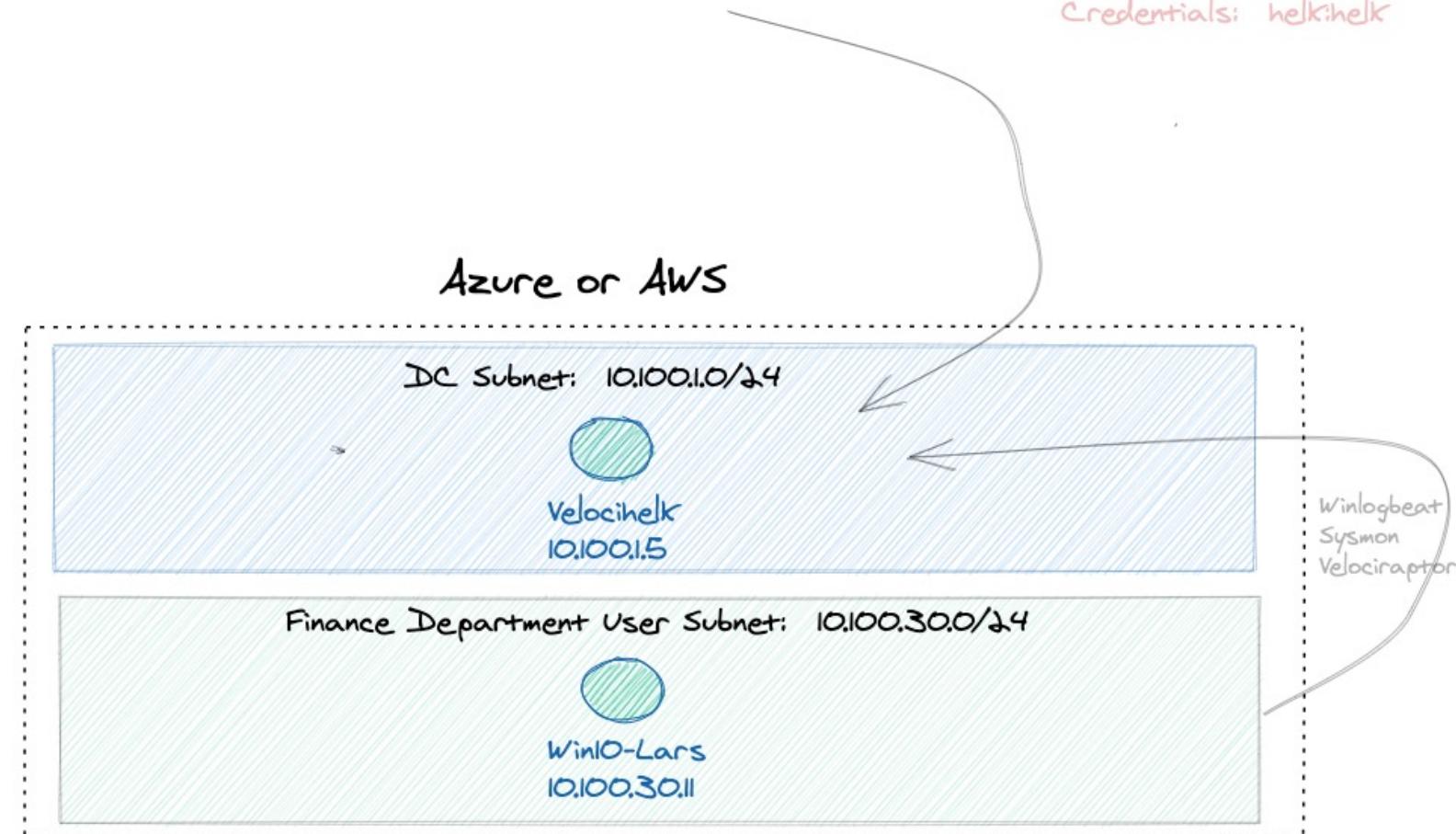
Kibana UI: <https://<VELOCIHELK>> Credentials: helkjhunting

Velociraptor Frontend: <https://<VELOCIHELK>:8889>

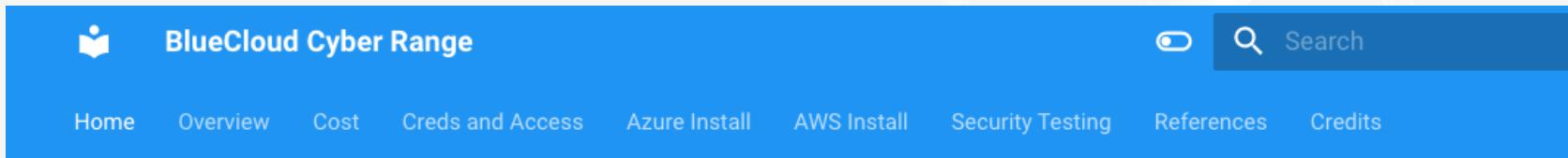
Credentials: helkjhelk



White listed Azure NSGs:  
\* TCP/3389 (RDP)  
\* TCP/5985 (WinRM)  
\* TCP/5986 (WinRM-HTTPS)  
\* TCP/22 (SSH)  
\* TCP/443 (Kibana)  
\* TCP/8080 (Apache Spark)  
\* TCP/8088 (KQL)  
\* TCP/2181 (Zookeeper)  
\* TCP/8889 (Velociraptor Frontend)  
\* TCP/8000 (Velociraptor Agent)



# BlueCloud Cyber Range



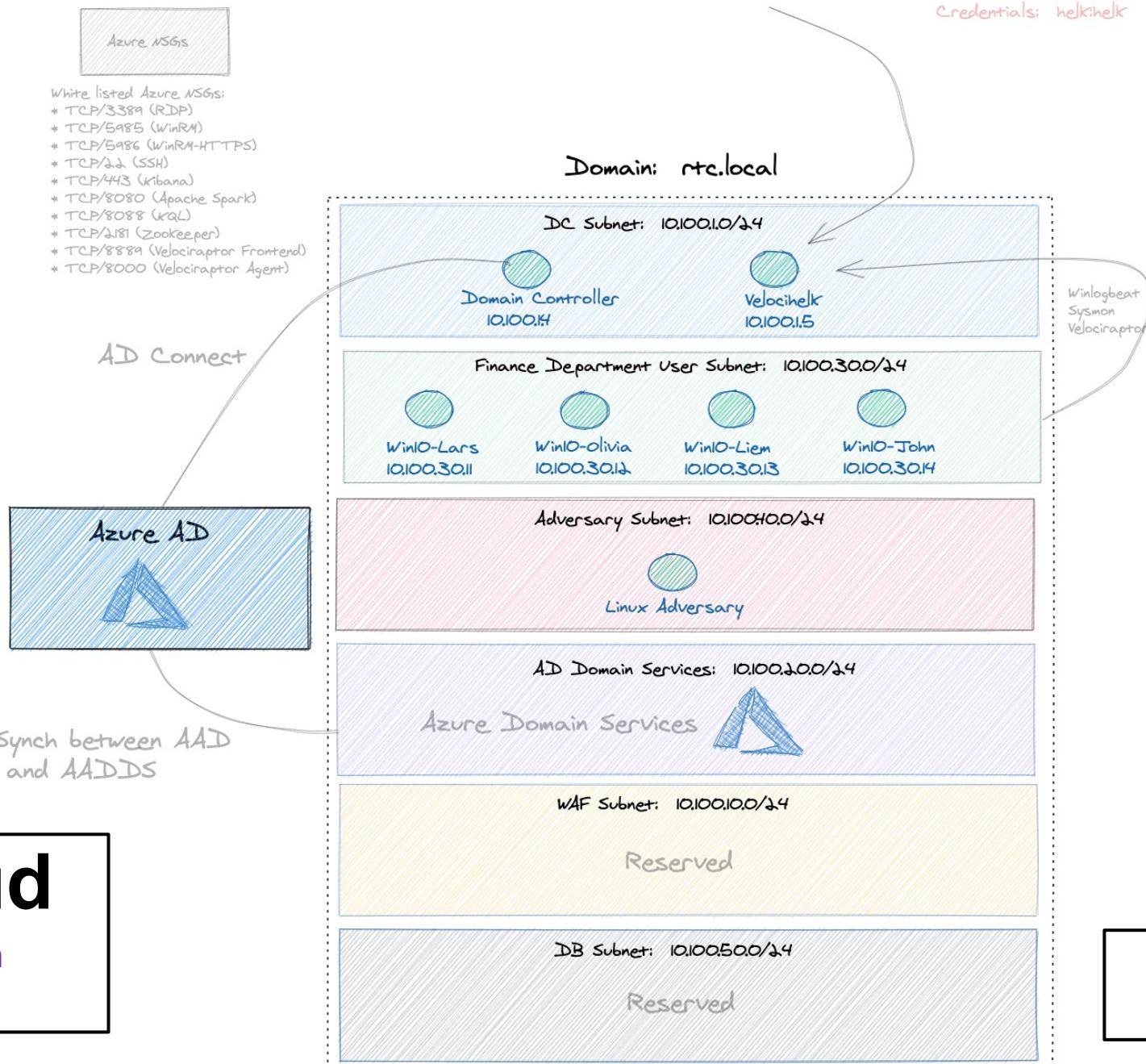
The screenshot shows a blue header bar with the title "BlueCloud Cyber Range". On the left is a logo icon. To the right are a search bar with a magnifying glass icon and a "Search" button, and a toggle switch. Below the header is a navigation menu with the following items: Home, Overview, Cost, Creds and Access, Azure Install, AWS Install, Security Testing, References, and Credits.

- Easily spin up a small Detection Engineering lab in AWS or Azure
- Logging server runs Velociraptor + HELK (*velocihelk*)
- Windows endpoint instrumented with Velociraptor agent that auto-registers
- Windows endpoint instrumented with Winlogbeat that ships Sysmon logs using Kafka transport to HELK
- Three tools on endpoint for adversary simulation
  - Atomic Red Team, Elastic Detection RTA, and APTSimulator

BlueCloud is currently single Windows host.



Kibana UI: <https://KVELOCIHELK> Credentials: helk:hunting  
Velociraptor Frontend: <https://KVELOCIHELK>:8889>  
Credentials: helk:helk



# PurpleCloud

A little tool to play with  
Azure Identity

Azure AD lab creation  
tool

# PurpleCloud Labs (1)



Hybrid + Identity Cyber Range      Overview  
[Home](#)

- ***PurpleCloud*** is an open-source Cyber Range that automates creation of simulation labs in Azure
  - **Site:** <https://www.purplecloud.network>
  - **Author:** Jason Ostrom
  - Terraform code generators that create unique ranges for different use cases



# PurpleCloud Labs (2)



A screenshot of the GitHub repository homepage for "iknowjason/PurpleCloud". The top navigation bar shows "Hybrid + Identity Cyber Range" and "Home". The main content area has a light gray background with a faint network graph watermark. It features a large title "Overview" and a "Edit" button with a pencil icon.

- ***What it is***
  - “Build your own lab” style of lab creation
  - For security researchers, Blue, Red, other security enthusiasts
  - Run attack simulations and understand defenses
  - Bug Bounty
  - Mix and match labs to create a custom, Hybrid Identity enterprise
  - User Story: used it for creating a Detection Engineering training class
- ***What it is not***
  - Guided vulnerability labs



# Generators Overview

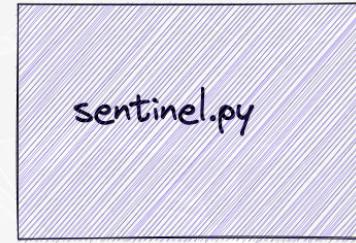
Create an AD Lab + SIEM



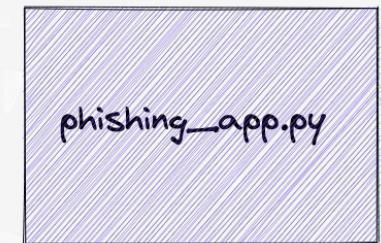
Create an Azure AD Lab



Create an Azure Sentinel Lab



Create an App Consent Phishing



Create an ADFS Lab



Azure AD Join VMs



Create a managed identity lab



Create Azure Storage lab



# Generator Usage & Workflow

- **Step 1:** Run python terraform code generator
- **Step 2:** Initialize terraform configuration
- **Step 3:** Create a terraform plan
- **Step 4:** Apply terraform plan
- **Step 5:** Terraform destroy

```
% cd generators/adfs  
% python3 adfs.py  
% terraform init  
% terraform plan --out=plan.run  
% terraform apply plan.run  
% terraform destroy
```

# Azure Active Directory



# Azure AD Lab Generator

- Creates an **Azure Active Directory** lab filled with Azure users, groups, and applications
- **Key Features**
  - Randomly generate Azure AD users using faker library for simulated users
  - Customizable list of Groups Added (Default: 11)
  - Customizable list of Azure AD Applications and Service Principals (Default: 7)
  - Users auto-assigned randomly into Groups
  - Includes a vulnerable privilege escalation scenario and attack scripts
- **Example Usage**

```
% python3 azure_ad.py --upn rtcfingroup.com --count 500 --apps 3 --groups 5
```

First of its kind Azure AD open-source tool to auto-generate Azure AD lab



# Create 1,000 Azure AD users

```
sec588@slingshot:~/tools/PurpleCloud$ python3 azure_ad.py --upn tecnica.co -c 1000
[+] Number of users desired: 1000
[+] upn suffix: tecnica.co
[+] Creating unique user list
    [-] Duplicate user Christopher Bailey ~ not adding to users list
    [-] Duplicate user Michelle Hernandez ~ not adding to users list
    [-] Duplicate user Michelle Hamilton ~ not adding to users list
    [-] Duplicate user Michael Miller ~ not adding to users list
    [-] Duplicate user Cody Martin ~ not adding to users list
    [-] Duplicate user Tammy Brown ~ not adding to users list
    [-] Duplicate user David Bell ~ not adding to users list
    [-] Duplicate user Michael Taylor ~ not adding to users list
    [-] Duplicate user William Johnson ~ not adding to users list
    [-] Duplicate user James Reed ~ not adding to users list
[+] Number of users added into list: 1000
[+] Number of duplicate users filtered out: 10
[+] Creating output files for Azure AD Users
    [+] Users csv file: azure_users.csv
    [+] Username txt file: azure_usernames.txt
    [+] Email addresses txt file: azure_emails.txt
    [+] Terraform file: users.tf
sec588@slingshot:~/tools/PurpleCloud$ █
```



# Outputs Files in Text and CSV for other tools

```
sec588@slingshot:~/tools/PurpleCloud$ wc -l azure_users.csv azure_usernames.txt azure_emails.txt
1000 azure_users.csv
1000 azure_usernames.txt
1000 azure_emails.txt
3000 total
sec588@slingshot:~/tools/PurpleCloud$
sec588@slingshot:~/tools/PurpleCloud$ more azure_users.csv
Wendy Pierce,wendypierce,wendypierce@tecniqa.co
Jennifer Hardy,jenniferhardy,jenniferhardy@tecniqa.co
Allen Camacho,allencamacho,allencamacho@tecniqa.co
Matthew Miller,matthewmiller,matthewmiller@tecniqa.co
Samantha Mejia,samanthamejia,samanthamejia@tecniqa.co
Sean Guerrero,seanguerrero,seanguerrero@tecniqa.co
Ashley Parks,ashleyparks,ashleyparks@tecniqa.co
Nancy Cole,nancycole,nancycole@tecniqa.co
Deanna Castillo,deannacastillo,deannacastillo@tecniqa.co
Aaron White,aaronwhite,aaronwhite@tecniqa.co
Valerie Sherman,valeriesherman,valeriesherman@tecniqa.co
Jackie Gibson,jackiegibson,jackiegibson@tecniqa.co
Michael Taylor,michaeltaylor,michaeltaylor@tecniqa.co
Judith Rivera,judithrivera,judithrivera@tecniqa.co
```



# Create Azure Applications and Groups: Auto-assign users into groups

```
sec588@slingshot:~/tools/PurpleCloud$ python3 azure_ad.py --upn tecniqa.co --apps 3 --groups 5
[+] No users specified ~ creating 100 users by default
[+] upn suffix: tecniqa.co
[+] Desired applications enabled: 3
[+] Desired groups enabled: 5
[+] Creating unique user list
[+] Number of users added into list: 100
[+] Number of duplicate users filtered out: 0
[+] Creating output files for Azure AD Users
  [+] Users csv file: azure_users.csv
  [+] Username txt file: azure_usernames.txt
  [+] Email addresses txt file: azure_emails.txt
  [+] Terraform file: users.tf
[+] Creating terraform file: apps.tf
[+] Creating terraform file: groups.tf
  [+] Adding all Azure users to this group: Users
sec588@slingshot:~/tools/PurpleCloud$
```

100 users are randomly placed into 5 different Azure AD Groups.

```
# Azure AD Group
resource "azureread_group" "Sales_Team" {
  display_name = "${var.upn_suffix} - Sales"
  security_enabled = true
  members = [
    azureread_user.user1.object_id,
    azureread_user.user4.object_id,
    azureread_user.user5.object_id,
    azureread_user.user9.object_id,
    azureread_user.user15.object_id,
    azureread_user.user17.object_id,
    azureread_user.user19.object_id,
    azureread_user.user20.object_id,
    azureread_user.user26.object_id,
```

# Service Principal Abuse Attack Scenario Created

```
sec588@slingshot:~/tools/PurpleCloud$ python3 azure_ad.py -c 25 --upn tecniqa.co --apps 7 -aa -ga -pra
[+] Number of users desired: 25
[+] upn suffix: tecniqa.co
[+] Desired applications enabled: 7
[+] Creating unique user list
[+] Number of users added into list: 25
[+] Number of duplicate users filtered out: 0
[+] Creating output files for Azure AD Users
    [+] Users csv file: azure_users.csv
    [+] Username txt file: azure_usernames.txt
    [+] Email addresses txt file: azure_emails.txt
    [+] Terraform file: users.tf
[+] Creating terraform file: apps.tf
    [+] Assigning the Privileged Role Administrator to MailReader_Application
    [+] Assigning the Global Administrator role to HelpDesk_Application
sec588@slingshot:~/tools/PurpleCloud$ █
```

Hat tip and credit to security researchers (Andy Robbins, Dirk-jan Mollema) for their writeups on this issue. Original articles are included in references section.



# Adversary Behaviors

- Simulate adversary techniques mapped to the Azure Threat Research Matrix: <https://microsoft.github.io/Azure-Threat-Research-Matrix/>

AZT202 - Password Spraying

An adversary may potentially gain access to AzureAD by guessing a common password for multiple users.

Resource
Azure Active Directory

Actions
N/A

Examples
Az PowerShell (Secret)      Azure CLI
Connect-AzAccount

Detections
Logs

Data Source	Application	Resource	Log Location
Azure Active Directory	Azure Portal	Windows Azure Service Management API	Sign-in Logs
Azure Active Directory	Microsoft Azure PowerShell	Windows Azure Service Management API	Sign-in Logs



# Detection Engineering (1)

- Start the process of Detection Engineering with Adversary Emulations
- Password Spray:** Attacker rotates IP addresses using Amazon API Gateway

User sign-ins (interactive)		User sign-ins (non-interactive)		Service principal sign-ins		Managed identity sign-ins		IP address	Location	Conditional Access
Date	Request ID	User	Application	Status						
10/26/2022, 7:33:56 PM	f2a7a2df-460d-41f5-8b...	Jason Ostrom	Azure Portal	Success	150.249.204.46	Chiyoda-Ku, Tokyo, JP	Not Applied			
10/26/2022, 9:10:16 AM	d7682cf4-4803-4964-9...	Rachel Mcdonald	Office 365 Management	Success	35.181.128.179	Paris, Paris, FR	Not Applied			
10/26/2022, 9:10:12 AM	1e505348-f238-43ff-8c...	Anna Ramirez	Microsoft Teams	Success	3.12.219.188	Columbus, Ohio, US	Not Applied			
10/26/2022, 9:10:08 AM	eaaa4454-d6b2-4929-a...	Tammy Stevenson	Microsoft Azure Active ...	Success	15.228.151.21	Sao Paulo, Sao Paulo, BR	Not Applied			
10/26/2022, 9:10:04 AM	2ee6a6a3-db87-426c-8f...	Jamie Wiggins		Failure	16.171.48.192	Stockholm, Stockholms ...	Not Applied			
10/26/2022, 9:09:59 AM	c45cdf44-48820-4ef8-8af...	Bethany Yates	Microsoft Authenticator...	Success	3.101.200.140	San Jose, California, US	Not Applied			
10/26/2022, 9:09:56 AM	89257053-71a8-48fd-8...	Emily Freeman		Failure	3.72.33.135	Frankfurt Am Main, Hes...	Not Applied			
10/26/2022, 9:09:51 AM	039d1f53-f4f1-422f-8b7...	Jason Stevens		Failure	13.40.207.210	London, Greater Londo...	Not Applied			
10/26/2022, 9:09:46 AM	24981e82-f0b8-436b-b...	Andrew Bowman	SkypeForBusinessAuth	Interrupted	35.93.127.125	Boardman, Oregon, US	Not Applied			
10/26/2022, 9:09:43 AM	1af28d2e-a85a-4d18-9...	Robert Vasquez		Failure	44.206.4.236	Ashburn, Virginia, US	Not Applied			
10/26/2022, 9:09:39 AM	9cd90787-1685-475a-8...	James Bennett	Office365 Shell WCSS-C...	Failure	108.128.160.235	Dublin, Dublin, IE	Not Applied			
10/26/2022, 9:09:34 AM	fef9b58e-20c9-4758-b9...	Barbara Jones		Failure	13.38.133.137	Paris, Paris, FR	Not Applied			
10/26/2022, 9:09:30 AM	5315239c-f007-45a8-be...	Crystal Baxter	Microsoft Exchange Onl...	Success	3.15.35.73	Columbus, Ohio, US	Not Applied			
10/26/2022, 9:09:26 AM	032fc0ef-ce07-43d7-90...	Amy Martinez	Microsoft Office	Success	18.229.99.159	Sao Paulo, Sao Paulo, BR	Not Applied			
10/26/2022, 9:09:21 AM	dec2c525-1e2e-4a70-a...	Joel Walker		Failure	16.171.49.70	Stockholm, Stockholms ...	Not Applied			
10/26/2022, 9:09:17 AM	12d547c0-6767-4e4b-b...	Christopher Carr	Office365APIEditor	Interrupted	13.52.201.217	San Jose, California, US	Not Applied			
10/26/2022, 9:09:14 AM	ae03351f-c742-4eff-b8...	Kristin Smith	Microsoft Azure	Success	3.66.172.9	Frankfurt Am Main, Hes...	Not Applied			
10/26/2022, 9:09:09 AM	ebcdc5cc-a9ee-49c1-a5...	Luis Thompson	ADIbizaUX	Failure	13.40.205.187	London, Greater Londo...	Not Applied			

Adversary runs a Password Spray and rotates their source IP address using Amazon API Gateway. Note the different location and user in each request.



# Detection Engineering (2)

- Password Spraying using Amazon API Gateway can be detected and blocked
  - Use Amazon's published IP prefixes to parse all API Gateway prefixes
  - Lookup IP address of password attempt against the list
  - Instrument SIEM, WAF, or Firewall to detect a threshold of password attempts and block

```
curl https://ip-ranges.amazonaws.com/ip-ranges.json | jq -r '.prefixes[] | select(.service=="API_GATEWAY") | .ip_prefix'
```

```
sec588@slingshot:~/Desktop/requests-ip-rotator$ ./cmd.sh
[+] Listing all Amazon API Gateway prefixes from ip-ranges.json
140.179.144.128/25
140.179.176.0/23
52.81.135.128/25
52.81.137.0/24
52.81.216.0/23
161.189.148.0/23
52.82.127.0/24
```



# Sentinel Detections



- Use Azure Sentinel and KQL queries for detection improvement
- Example KQL: ***union SigninLogs***

A screenshot of the Azure Sentinel KQL query interface. The top navigation bar includes "Run", "Time range : Custom", "Save", "Share", "New alert rule", "Export", "Pin to", and "Format query". The query itself is a simple "union SigninLogs" command. The results section shows a table with columns: SourceGroup, Identity, Level, Location, AlternateSignInName, and AppDisplayName. The data in the table is as follows:

SourceGroup	Identity	Level	Location	AlternateSignInName	AppDisplayName
soft.aadiam	Jason Ostrom	4	JP		Azure Portal
soft.aadiam	Andrew Mitchell	4	BR	andrewmitchell@rtcfingroup....	Office 365 Management
soft.aadiam	Marcus Garcia	4	GB	marcusgarcia@rtcfingroup.c...	O365 Suite UX
soft.aadiam	Mary Mathis	4	SE	marymathis@rtcfingroup.com	Microsoft Azure Active Direct...
soft.aadiam	Felicia Mathis	4	IE	feliciamathis@rtcfingroup.com	Office365APIEditor
soft.aadiam	Mary Prince	4	US	maryprince@rtcfingroup.com	Office365APIEditor
soft.aadiam	Larry Ortiz	4	GB	larryortiz@rtcfingroup.com	Azure Active Directory Power...

Build KQL detections and automate processes for detection engineering. Map Azure-specific TTPs using the Azure Threat Research Matrix.



# Service Principal Privilege Abuse Primitives

## AZT405 - Azure AD Application

Adversaries may abuse the assigned permissions on an Azure AD Application to escalate their privileges.

ID	Name	Description	Action	Resources
AZT405.1	Application Role	By compromising a user, user in a group, or service principal that has an application role over an application, they may be able to escalate their privileges by impersonating the associated service principal and leveraging any privileged assigned application role.		AzureAD
AZT405.2	Application API Permissions	By compromising a service principal whose application has privileged API permissions, an attacker can escalate their privileges to a higher privileged role.		AzureAD
AZT405.3	Application Registration Owner	By compromising an account who is an 'Owner' over an application that is configured with additional roles or API permissions, an attacker can escalate their privileges by adding a certificate or credentials & logging in as the service principal.		AzureAD



# Detection Engineering & Sentinel

- Example KQL

AuditLogs

```
| where OperationName == "Update application – Certificates and secrets management" and  
Category == "ApplicationManagement"
```

- Detect Service Principal Logins
- Detect changes to App Role Assignments
- Detect adding secrets to applications

Results    Chart    Add bookmark

TimeGenerated [UTC]	ResourceId	OperationName	OperationVersion	Category
> 10/30/2022, 8:13:35.333 AM	/tenants/1a82558d-66e0-48b0-b370-72df4caf1852/pr...	Add member to group	1.0	GroupManagement
> 10/30/2022, 8:13:35.399 AM	/tenants/1a82558d-66e0-48b0-b370-72df4caf1852/pr...	Add member to group	1.0	GroupManagement
> 10/30/2022, 8:14:13.769 AM	/tenants/1a82558d-66e0-48b0-b370-72df4caf1852/pr...	Add member to group	1.0	GroupManagement
> 10/30/2022, 8:46:26.790 AM	/tenants/1a82558d-66e0-48b0-b370-72df4caf1852/pr...	Update service principal	1.0	ApplicationManagement
> 10/30/2022, 8:46:26.868 AM	/tenants/1a82558d-66e0-48b0-b370-72df4caf1852/pr...	Update application – Certificates and secrets managem...	1.0	ApplicationManagement
> 10/30/2022, 8:46:26.884 AM	/tenants/1a82558d-66e0-48b0-b370-72df4caf1852/pr...	Update application	1.0	ApplicationManagement
> 10/30/2022, 8:46:41.003 AM	/tenants/1a82558d-66e0-48b0-b370-72df4caf1852/pr...	Add member to role	1.0	RoleManagement
> 10/30/2022, 8:47:41.325 AM	/tenants/1a82558d-66e0-48b0-b370-72df4caf1852/pr...	Add member to role outside of PIM (permanent)	1.0	RoleManagement



# Active Directory with SIEM & Velociraptor



# Active Directory + SIEM Lab Generator (1)

- ***ad.py*** is a Terraform generator for an Active Directory Lab created with Azure VMs and includes support for Velociraptor and a SIEM using Hunting ELK
- **Key Features**
  - Create a custom IaaS Active Directory environment with Azure VMs
  - Deploys a SIEM (Hunting ELK) and endpoints instrumented with Sysmon/Winlogbeat/Velociraptor/Atomic Red Team
  - Endpoints include PurpleSharp
  - Downloads Azure AD Connect msi installer onto DC's desktop
- **Example Usage**

```
% python3 ad.py --domain_controller --ad_domain rtcfingroup.com --admin RTCAdmin --password  
MyPassword012345 --csv users.csv --endpoints 2 --domain_join
```



# Active Directory + SIEM Lab Generator (2)

- Create a customizable, realistic, large AD environment
  - Import users from CSV, or randomly generate users
  - Random Generator creates as many users as you desire
- Automatically creates OU, AD Groups, and assigns users into OU, Groups
- Automatic Domain Join: Configurable Domain Join per VM
- Auto-logon Domain users with Domain credentials, for realistic simulations (Interactive Type 2 Logon, Mimikatz)
- Great for practicing or learning Active Directory



# Simulate an On-Premise Active Directory Lab

Create an AD Domain with 500 AD users. Create three Windows 10 Professional endpoints, joining them to the domain.

```
sec588@slingshot:~/tools/PurpleCloud$ python3 ad.py --domain_controller --ad_domain stora.io --admin StoraAdmin --ad_users 500 --endpoints 3 --domain_join --auto_logon
[+] Public IP address detected: 99.182.28.252
[+] Setting Azure NSG Whitelist to: 99.182.28.252
[+] Local Admin account name: StoraAdmin
[+] Setting AD Domain to build AD DS: stora.io
[+] Creating unique user list
  [-] Duplicate user Joshua Jones ~ not adding to users list
[+] Number of users added into list: 500
[+] Number of duplicate users filtered out: 1
[+] Number of Windows 10 Pro endpoints desired: 3
[+] Using default Resource Group Name: PurpleCloud
[+] Using default location: eastus
[+] Domain Join is set to true
[+] Auto Logon is set to true
[+] Creating the main terraform file: main.tf
[+] Creating the providers terraform file: providers.tf
[+] Creating the nsg terraform file: nsg.tf
[+] Building Windows 10 Pro
  [+] Number of systems to build: 3
    [+] Getting default configuration template for Windows 10 Pro
    [+] Base Hostname: win10
    [+] Administrator Username: StoraAdmin
    [+] Administrator Password: 9a80sgQzCK
```



# Users Placed into OU and AD Groups automatically

```
sec588@slingshot:~/tools/PurpleCloud$ wc -l ad_users.csv
506 ad_users.csv
sec588@slingshot:~/tools/PurpleCloud$ more ad_users.csv
name,upn,password,groups,oupath,domain_admin
Lars Borgerson,larsborgerson@stora.io,M1naKXXy4n,IT,OU=IT;DC=stora;DC=io,False
Olivia Odinsdottir,oliviaodinsdottir@stora.io,BHvy04tR9w,IT,OU=IT;DC=stora;DC=io,True
Liem Anderson,liemanderson@stora.io,tV6QPh9cmo,IT,OU=IT;DC=stora;DC=io,False
John Nilsson,johnnilsson@stora.io,83sopkFaNh,IT,OU=IT;DC=stora;DC=io,False
Jason Lindqvist,jasonlindqvist@stora.io,Y9EGolh8k0,IT,OU=IT;DC=stora;DC=io,True
Brittany Martin,brittanymartin@stora.io,mE6YdstoM4,Engineering,OU=Engineering;DC=stora;DC=io,False
Danielle Gill,daniellegill@stora.io,mE6YdstoM4,Engineering,OU=Engineering;DC=stora;DC=io,False
James Grant,jamesgrant@stora.io,mE6YdstoM4,IT,OU=IT;DC=stora;DC=io,False
Victor Johnson,victorjohnson@stora.io,mE6YdstoM4,IT,OU=IT;DC=stora;DC=io,False
Mary Johnson,maryjohnson@stora.io,mE6YdstoM4,Marketing,OU=Marketing;DC=stora;DC=io,False
Emily Byrd,emilybyrd@stora.io,mE6YdstoM4,Executive,OU=Executive;DC=stora;DC=io,False
Scott Jones,scottjones@stora.io,mE6YdstoM4,Executive,OU=Executive;DC=stora;DC=io,False
Sherry Oneal,sherryoneal@stora.io,mE6YdstoM4,Legal,OU=Legal;DC=stora;DC=io,False
John Contreras,johncontreras@stora.io,mE6YdstoM4,Sales,OU=Sales;DC=stora;DC=io,False
Kenneth Hickman,kennethhickman@stora.io,mE6YdstoM4,Sales,OU=Sales;DC=stora;DC=io,False
Brandi McDaniel,brandimcdaniel@stora.io,mE6YdstoM4,Executive,OU=Executive;DC=stora;DC=io,False
```



# Each Windows 10 Pro Endpoint has a custom Terraform file created (for further editing if desired)

```
[+] Building Windows 10 Pro Endpoint 1
[+] Hostname: win10-1
[+] IP address: 10.100.20.10
[+] Auto Logon Domain user
    [+] Getting the default ad user and password
    [+] Auto Logon this Win10 Pro to AD User: Alan Gill
    [+] Username: alangill
    [+] Password: mE6YdstoM4
[+] Setting Domain Controller for this endpoint to join domain: 10.100.10.4
[+] Created terraform: win10-1.tf
```

```
sec588@slingshot:~/tools/PurpleCloud$ ls -al win10-*
-rw-rw-r-- 1 sec588 sec588 5524 Jul  9 11:21 win10-1.tf
-rw-rw-r-- 1 sec588 sec588 5520 Jul  9 11:21 win10-2.tf
-rw-rw-r-- 1 sec588 sec588 5518 Jul  9 11:21 win10-3.tf
sec588@slingshot:~/tools/PurpleCloud$
```

# Virtual Machines Created

 **PurpleCloud-tze4s** ⚡ ⭐ ...

Resource group | i Directory: storasecurity

« » Create Manage view Delete resource group Refresh Export to CSV Open query Assign

Overview Activity log Access control (IAM) Tags Resource visualizer Events

^ Essentials

Subscription ([move](#)) : [Stora Pay-As-You-Go Subscription](#) Deployments : [No deployments](#)  
Subscription ID : bb9c8c9f-34c2-4442-89ff-3c67517c1b22 Location : Central US  
Tags ([edit](#)) : [Click here to add tags](#)

Resources Recommendations

Type equals **Virtual machine** Location equals **all** + Add filter

Showing 1 to 4 of 4 records.  Show hidden types ⓘ

<input type="checkbox"/> Name ↑	Type ↑↓
 dc1	Virtual machine
 win10-1-tze4s	Virtual machine
 win10-2-tze4s	Virtual machine
 win10-3-tze4s	Virtual machine

Cost Management

Cost analysis



# Active Directory Created with 3 Domain Joined

This screenshot shows the Active Directory Users and Computers interface. The left pane displays a tree view of the domain structure under 'stora.io'. The right pane lists a large number of users, each with their name, type (User), and a brief description. A red box highlights the user list, and a black arrow points from a text box below to the bottom of the user list.

Name	Type	Description
Alexander Lang	User	
Amanda White	User	
Andrew Foster	User	
Andrew Horton	User	
Andrew Howell	User	
Andrew Maldonado	User	
Angela Rodriguez	User	
April Anderson	User	
Austin Miller	User	
Becky Hughes	User	
Brenda Nunez	User	
Brian Wilson	User	
Brittany Stephenson	User	
Carmen Jackson	User	
Charles Smith	User	
Christian Smith	User	
Christine Black	User	
Christopher Hays	User	
Clinton Anderson	User	
Corey Avila	User	
Crystal Gordon	User	
David Lopez	User	
David Robinson	User	
Dean McDaniel	User	
Devon Golden	User	
Dwayne Krause	User	
Edward Molina	User	
Edward Rios	User	

500 Domain Users assigned into different OU and AD Groups.

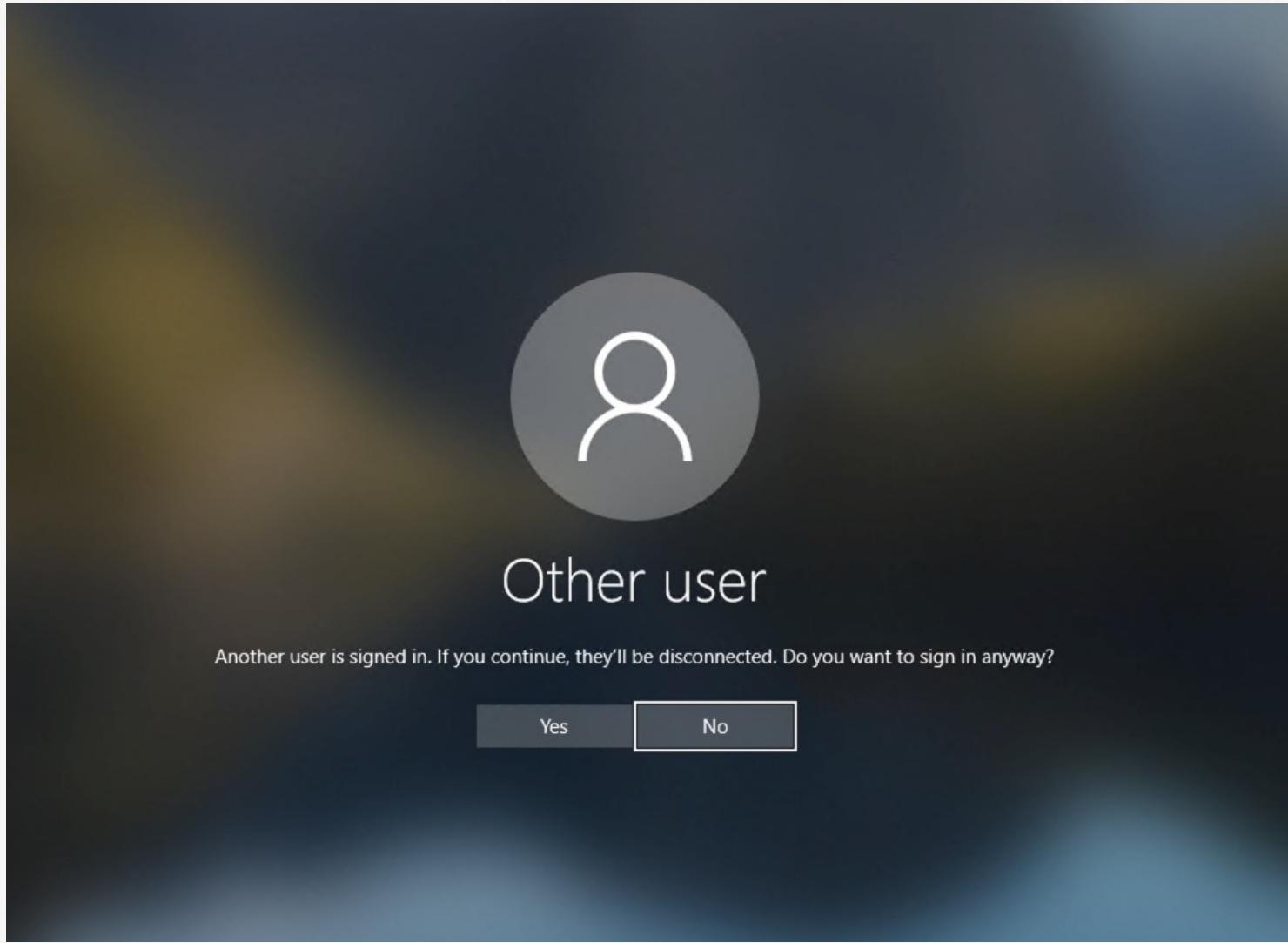
This screenshot shows the Active Directory Users and Computers interface. The left pane displays the domain structure under 'stora.io'. The right pane lists three joined computers: 'win10-1', 'win10-2', and 'win10-3', all categorized as 'Computer'. A red box highlights the computer list, and a black arrow points from a text box below to the bottom of the computer list.

Name	Type	Description
win10-1	Computer	
win10-2	Computer	
win10-3	Computer	

Three Windows 10 Pro joined to the domain based on Python script.



# Auto Logon Domain Users with AD credentials



With this feature, you can practice lateral movements across domain joined systems and extracting domain credentials from LSASS memory.



# Passwords in AD default to Strong, but customizable

```
[+] Default Local Administrator Credentials on all Windows  
[+] Username: StoraAdmin  
[+] Password: MiXz47PXnh ←  
[+] Built AD DS Domain: stora.io
```

Default behavior is to auto-generate a strong password and assign to all users, putting into CSV file.

```
Lars Borgerson,larsborgerson@stora.io,4NgLlSurEg,IT,OU=IT;DC=stora;DC=io,False  
Olivia Odinsdottir,oliviaodinsdottir@stora.io,Nzzw2AzjTy,IT,OU=IT;DC=stora;DC=io,True  
Liem Anderson,liemanderson@stora.io,dtdBP8USiT,IT,OU=IT;DC=stora;DC=io,False  
John Nilsson,johnnilsson@stora.io,EMX0BjYoN8,IT,OU=IT;DC=stora;DC=io,False  
Jason Lindqvist,jasonlindqvist@stora.io,CcoW4whUK6,IT,OU=IT;DC=stora;DC=io,True  
Samantha Simmons,samanthasimmons@stora.io,c2aiIXy4ef,Sales,OU=Sales;DC=stora;DC=io,False  
Renee Ramirez,reneeramirez@stora.io,c2aiIXy4ef,Sales,OU=Sales;DC=stora;DC=io,False
```

Specify your desired password for all users via command line parameter.

```
sec588@slinashot:~/tools/PurpleCloud$ python3 ad.py --domain_controller --ad_domain stora.io --admin StoraAdmin --password MyPassword012345 --ad_users 500 --endpoints 3 --domain_join --auto_logon
```

# Build Hunting ELK + Velociraptor with one endpoint

```
sec588@slingshot:~/tools/PurpleCloud$ python3 ad.py --helk --endpoint 1
[+] Public IP address detected: 99.182.28.252
[+] Setting Azure NSG Whitelist to: 99.182.28.252
[+] Number of Windows 10 Pro endpoints desired: 1
[+] Using default Resource Group Name: PurpleCloud
[+] Using default location: eastus
[+] HELK server enabled 10.100.30.4
[+] Installing velociraptor, winlogbeat agents on endpoints and exporting logs to HELK
[+] Creating the main terraform file: main.tf
[+] Creating the providers terraform file: providers.tf
[+] Creating the nsg terraform file: nsg.tf
[+] Building Windows 10 Pro
[+] Number of systems to build: 1
[+] Getting default configuration template for Windows 10 Pro
[+] Base Hostname: win10
[+] Administrator Username: RTCAdmin
[+] Administrator Password: 1C4qvSRIFj
[+] Join Domain: false
[+] Auto Logon Domain User: false
[+] Install Sysmon: true
[+] Install Atomic Red Team (ART): true
[+] Forwarding winlogbeat logs to 10.100.30.4
[+] Installing velociraptor and registering to 10.100.30.4
[+] Subnet Association: user_subnet
[+] Building Windows 10 Pro Endpoint 1
```



# PurpleCloud IaaS Advantage

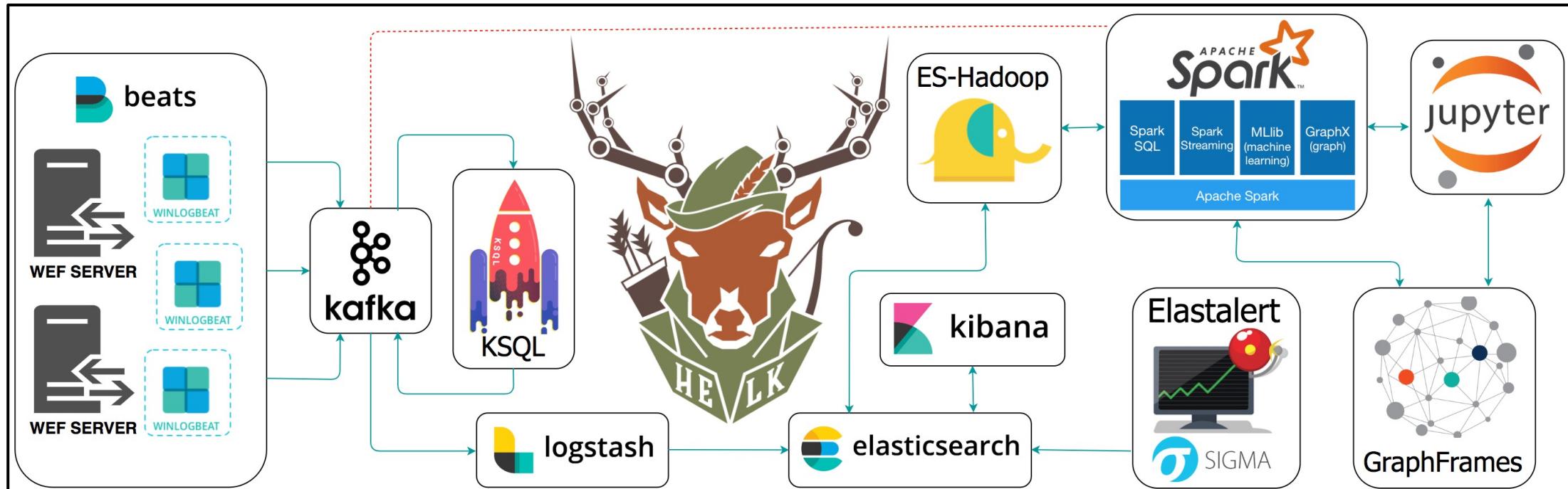
- Advantage of ***PurpleCloud***:
- Practice simulations against Domain Joined workstations with users logged in with domain user credentials
- Simulate lateral movement in AD enterprise
- Simulate extraction of domain user credentials from memory
- Windows and Sysmon logs shipped to a SIEM (HELK)
  - Sysmon v14
  - Customize your XML (uses SwiftOnSecurity)
- Study forensic artifacts with Velociraptor

PurpleCloud is a solid lab to learn  
Active Directory



# HELK: The Hunting ELK (1)

- *PurpleCloud* automatically builds HELK as the SIEM
  - <https://github.com/Cyb3rWard0g/HELK>
- Endpoints ship Sysmon logs via Winlogbeat agent
- HELK hardware option #4 is built for Jupyter Notebooks + ElastAlert



# HELK: The Hunting ELK (2)

- Endpoints configured with Sysmon Version 14
- SwiftOnSecurity XML configuration
- Files included in Github repository and can be customized
  - winlogbeat.yml
  - Sysmon.zip
  - Symonconfig-export.xml

```
user@host ad % cd files/velocihelk
user@host velocihelk %
user@host velocihelk % ls
helk.sh.tpl
sysmonconfig-export.xml
winlogbeat.yml.tpl
user@host velocihelk %
```

# Velociraptor Live Response



- *PurpleCloud* automatically install a Velociraptor Server
- Deploys Velociraptor agent on Windows systems
- <https://github.com/Velocidex/velociraptor>
- Endpoint visibility tool for digital forensics and live response
- Uses Velociraptor Query Language (VQL) to interrogate hosts and pull forensic artifacts

# Hybrid Identity



# Hybrid Identity Lab

- Advantage of *PurpleCloud*
- Creates an Azure AD lab + On-Premise AD lab
- Automates creation of a mixed-use Hybrid Identity lab
- Drops latest Azure AD Connect on DC's Desktop
  - Use Azure AD Connect agent to synchronize users from on-premise to Azure AD
  - Saves time and effort doing manual installation
- Attack simulations against Azure AD Joined Windows 10 and Hybrid Joined devices that are also joined to On-Premise AD

We are seeing a large growth of companies deploying Hybrid cloud, mixing On-Premise with Cloud systems, & synchronizing users into the Cloud.



# Microsoft Sentinel



# Microsoft Sentinel Lab Generator

- Creates a **Microsoft Sentinel** and log analytics workspace lab with endpoints
- **Key Features**
  - Endpoints install Azure Monitoring agents and send logs to Log Analytics workspace
  - Endpoints install Sysmon v14
  - Sentinel Data Connector for Office
  - Sentinel Data Connector for AAD
  - Supports full Active Directory deployment with Domain Join (same as ad.py)
- **Example Usage**

```
% python3 sentinel.py --domain_controller --ad_domain rtcfingroup.com --admin RTCAdmin --password MyPassword012345 --csv users.csv --endpoints 2 --domain_join
```



# Detection Engineering & Sentinel

- Microsoft Sentinel KQL for Windows Event Logs:
  - Windows Event Logs: *union SecurityEvent*
  - Sysmon: *union Event*
- Run *PurpleSharp*

```
PS C:\tools>
PS C:\tools> .\PurpleSharp.exe /t T1055.002,T1055.003,T1055.004
10/29/2022 04:26:27 [*] Starting T1055.002 Simulation on win10-1
10/29/2022 04:26:27 [*] Simulator running from C:\tools\PurpleSharp.exe with PID:9408 as win10-1\RTCAdmin
10/29/2022 04:26:27 [*] Process notepad.exe with PID:7624 started for the injection
10/29/2022 04:26:28 [*] Calling OpenProcess on PID:7624
10/29/2022 04:26:28 [*] Calling VirtualAllocEx on PID:7624
10/29/2022 04:26:28 [*] Calling WriteProcessMemory on PID:7624
10/29/2022 04:26:28 [*] Calling CreateRemoteThread on PID:7624
10/29/2022 04:26:28 [*] Simulation Finished
10/29/2022 04:26:28 [*] Playbook Finished
```

Source	Microsoft-Windows-Sysmon
EventLog	Microsoft-Windows-Sysmon/Operational
Computer	win10-1.rtcfingroup.com
EventCategory	1
EventLevel	4
EventLevelName	Information
UserName	NT AUTHORITY\SYSTEM
ParameterXml	<Param>-</Param><Param>2022-10-29 04:16:04.008</Param><Param>{DAC2892A-A904-635C-BF00-000000000600}</Param><Param>3904</Param>00000000</Param><Param>0x3e7</Param><Param>0</Param><Param>System</Param><Param>MD5=B073F18D23BE85799A640147AF9ABA99,SHA256=1C54DDCDC0651A1E0D8AA5F03D11D6B0521177F,IMPHASH=2B44D2206B9865383429E9C1524F1CAC</Param>
EventData	<DataItem type="System.XmlData" time="2022-10-29T04:16:04.0103235+00:00" sourceHealthServiceId="E96E82D9-734B-7A20-AE83-99DFDAF5E91F">Microsoft Corporation</Data><Data Name="OriginalFileName">cscript.exe</Data><Data Name="CommandLine">"C:\windows\system32\cscript.exe" /nologo</Data>
EventID	1

PurpleSharp is downloaded to all  
Windows 10 and can be accessed:  
**C:\tools\PurpleSharp.exe**

PurpleSharp is a C# adversary simulation tool that executes adversary techniques. It can be used to generate and study Sysmon logs that are shipped to Sentinel.

<https://github.com/mvelazc0/PurpleSharp>

# Consent Phishing



# Application Consent Phishing Lab

- Creates a multi-tenant Azure AD application that can be used for consent phishing attack and defense
- **Key Features**
  - Multi-tenant Azure AD application that can be used to practice app consent phishing
  - Customizable Graph API permissions (Default: Mail.Read, Files.Read)
  - Script parameter to specify custom application name
  - Script parameter to specify redirect\_uri
- **Example Usage**

```
% python3 phishing_app.py --name PhishingApp --redirect_uri https://www.evilcorp.io/get_token
```



# Azure Threat Research Matrix

## AZT203 - Malicious Application Consent

An adversary may lure a victim into giving their access to a malicious application registered in AzureAD.

### Resource

Azure Active Directory

### Actions

Any user can consent to an application which will impersonate that user's privileges.

### Examples

N/A

### Detections

#### Logs

Data Source	Application	Resource	Log Location
Azure Active Directory	N/A	AAD	Log Analytics



# User Consents to Application

 Microsoft  
jostrom@stora.io

## Permissions requested

**Sample App**  
**unverified**

This app may be risky. Only continue if you trust this app. [Learn more](#)

This app would like to:

- ▽ Maintain access to data you have given it access to
- ▽ Read user contacts
- ▽ Sign in and read user profile
- ▽ Read user mail
- ▽ Send mail as a user
- ▽ Have full access to all files user can access
- ▽ Read user files
- ▽ Read all files that user can access
- Consent on behalf of your organization

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at <https://myapps.microsoft.com>. [Show details](#)

Does this app look suspicious? [Report it here](#)

[Cancel](#) [Accept](#)



# Detection Engineering & Sentinel

- Azure Portal: *Audit Logs*
- Example KQL: *union AuditLogs*

Sample App | Audit logs ...

Enterprise Application

Overview

Date : Last 1 month Show dates as : Local Service : All Category : All Activity : All Add filters

Date	Service	Category	Activity	Status	Status reason	Target(s)	Initiated by (actor)
10/30/2022, 8:18:27 AM	Core Directory	ApplicationManagement	Consent to application	Success		Sample App	jostrom@stora.io

Feedback Qu

Run Time range : Custom Save Share New alert rule Export Pin to Format query

1 union AuditLogs

Results Chart Add bookmark

TimeGenerated [UTC]	ResourceId	OperationName	OperationVersion	Category	ResultSignature
> 10/29/2022, 11:17:37.880 PM	/tenants/9d9817d9-f209-4430-8f4f-cc03332848cb/pr...	Update user	1.0	UserManagement	None

Audit logs can trigger alarms on user behavior for application consent to application permissions.

# Managed Identity



# Managed Identity Lab Generator

- Creates a **Managed Identity** lab with an Azure VM Identity automatically created and assigned
- **Key Features**
  - Creates one Azure Virtual Machine with a managed identity assigned
  - Default user assigned identity with role of reader on the subscription
  - Script parameter to change role to owner or contributor
  - Randomly generates one Azure AD user with a role of Virtual Machine Contributor
  - Script parameter to add a system-assigned identity to the Virtual Machine
  - Creates storage account, containers, blobs, shares, and key vaults
- **Example Usage**

```
% python3 managed_identity.py -u rtcfingroup.com -n rtcfin -l eastus -a RTCAdmin -p MyPassword012345 -ua reader -sa
```



# Managed Identity Simulations

## AZT601.1 - Steal Managed Identity JsonWebToken: Virtual Machine IMDS Request

By utilizing access to IMDS, an attacker can request a JWT for a Managed Identity on an Azure VM if they have access to execute commands on the system.

**Resource**

Virtual Machine

**Actions**

- Microsoft.Compute/virtualMachines/write
- Microsoft.Compute/virtualMachines/extensions/\*

**Examples**

**PowerShell**   **Azure Portal**

```
powershell.exe -c $a=Invoke-RestMethod -Uri 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01' &"resource=https://management.azure.com/" -Method GET -Headers @{Metadata='true'} -UseBasicParsing;$a.access_token
```



# Acquire an Identity Access Token

```
PS C:\Users\RTCAdmin>
PS C:\Users\RTCAdmin> Invoke-WebRequest -Uri 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https%3A%2F%2Fmanagement.azure.com%2F' -Headers @{Metadata="true"} -UseBasicParsing

StatusCode      : 200
StatusDescription : OK
Content          : {"access_token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6IjJaUXBKM1VwYmpBwVhZ2FYRUp...OGxWMFRPSS...Ly...
RawContent       : HTTP/1.1 200 OK
Content-Length   : 1709
Content-Type     : application/json; charset=utf-8
Date             : Sun, 30 Oct 2022 08:20:41 GMT
Server           : IMDS/150.870.65.797

{"access_token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJSU...
```

An attacker with RDP access can acquire the Identity's access token or use az vm run command.

An attacker can sign into Azure AD using the VM's managed identity. Sign in logs can track this behavior.

```
jason@Azure:~$ spID=$(az resource list -n rtcfin --query [*].identity.principalId --out tsv)
jason@Azure:~$ echo $spID
9235040a-a847-43c9-96cf-d84427dbfe85
jason@Azure:~$ 
jason@Azure:~$ az login --identity
[
  {
    "environmentName": "AzureCloud",
```

# Detection Engineering & Sentinel

- Azure Portal: *Managed Identity sign-ins*
- Example KQL: *union AADManagedIdentitySignInLogs*

User sign-ins (interactive) User sign-ins (non-interactive) Service principal sign-ins Managed identity sign-ins

Sign-ins in the table below are grouped by application. Click on a row to see all the sign-ins for an application on that date and time.

Date	↑↓ Request ID	↑↓ Managed identity...	↑↓ Managed identity...	Status	IP address	↑↓ Resource	Resource ID	↑↓ # sign ins
> 10/30/2022, 9:00:00 /	6c2c471a-cdae-4dfc-...	b25e22f7-c654-489c-b3	rtcfin	Success	Windows Azure Servic...	797f4846-ba00-4fd7-...	1	

```
1 union AADManagedIdentitySignInLogs
2
```

Results Chart Add bookmark

<input type="checkbox"/> TimeGenerated [UTC]	OperationName	OperationVersion	Category	ResultType	ResultSignature
> 10/30/2022, 8:20:06.198 AM	Sign-in activity	1.0	ManagedIdentitySignInLogs	0	None



# Storage Lab



# Storage Lab Generator

- Creates an **Azure Storage** security lab
- **Azure Resources Created**
  - Storage account
  - Three storage containers with different permission levels (private, blob, container)
  - Two azure shares
  - Upload of fake, sensitive files to shares and containers as blobs
  - Key vault with secrets, private keys, and certificates
- **Example Usage**

```
% python3 storage.py
```



# Security Testing for Anonymous Blob Access

Name	Last modified	Public access level
<input type="checkbox"/> container1	10/30/2022, 9:21:55 AM	Blob
<input type="checkbox"/> container2	10/30/2022, 9:21:55 AM	Container
<input type="checkbox"/> container3	10/30/2022, 9:21:55 AM	Private

The **Container** access level enables indexing of files.

Name	Modified	Access tier
<input type="checkbox"/> cc.csv	10/30/2022, 9:21:56 AM	Hot (Inferred)
<input type="checkbox"/> customers.csv	10/30/2022, 9:21:56 AM	Hot (Inferred)
<input type="checkbox"/> finance.xlsx	10/30/2022, 9:21:56 AM	Hot (Inferred)
<input type="checkbox"/> hr.xlsx	10/30/2022, 9:21:56 AM	Hot (Inferred)

The simulation lab uploads sample files.



← → ⌂ https://purplestorage3zg78.blob.core.windows.net/container2?comp=list

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<EnumerationResults ContainerName="https://purplestorage3zg78.blob.core.windows.net/container2">
  <Blobs>
    <Blob>
      <Name>cc.csv</Name>
      <Url>
        https://purplestorage3zg78.blob.core.windows.net/container2/cc.csv
      </Url>
      <LastModified>Sun, 30 Oct 2022 00:21:56 GMT</LastModified>
      <Etag>0x8DABA0CC12CB3C3</Etag>
      <Size>30</Size>
      <ContentType>application/octet-stream</ContentType>
      <ContentEncoding/>
      <ContentLanguage/>
    </Blob>
    <Blob>
      <Name>customers.csv</Name>
      <Url>
        https://purplestorage3zg78.blob.core.windows.net/container2/customers.csv
      </Url>
      <LastModified>Sun, 30 Oct 2022 00:21:56 GMT</LastModified>
      <Etag>0x8DABA0CC129F506</Etag>
      <Size>10</Size>
      <ContentType>application/octet-stream</ContentType>
      <ContentEncoding/>
      <ContentLanguage/>
    </Blob>
  </Blobs>
</EnumerationResults>
```

Appending a string of “`?comp=list`” to a storage container with “container” access level will reveal all of the files in the target container.

# ADFS Federation



# ADFS Lab Generator

Credit to Roberto Rodriguez  
and his Azure Simuland tool.  
The ADFS Lab Generator was  
inspired by Simuland.

- Creates an ***Active Directory Federation Services (ADFS)*** lab.
- **Key Features**
  - Deploys an ADFS server joined to a created Domain Controller with AD Domain
  - Deploys Azure Sentinel
  - Deploys Azure monitoring agent on ADFS server and ADFS Audit log best practices for shipping logs into Sentinel
  - Supports self-signed ADFS certificate deployment (Default)
  - Supports ADFS trusted certificate import using optional script parameters
  - Implements ADFS Audit log best practices
- **Example Usage**

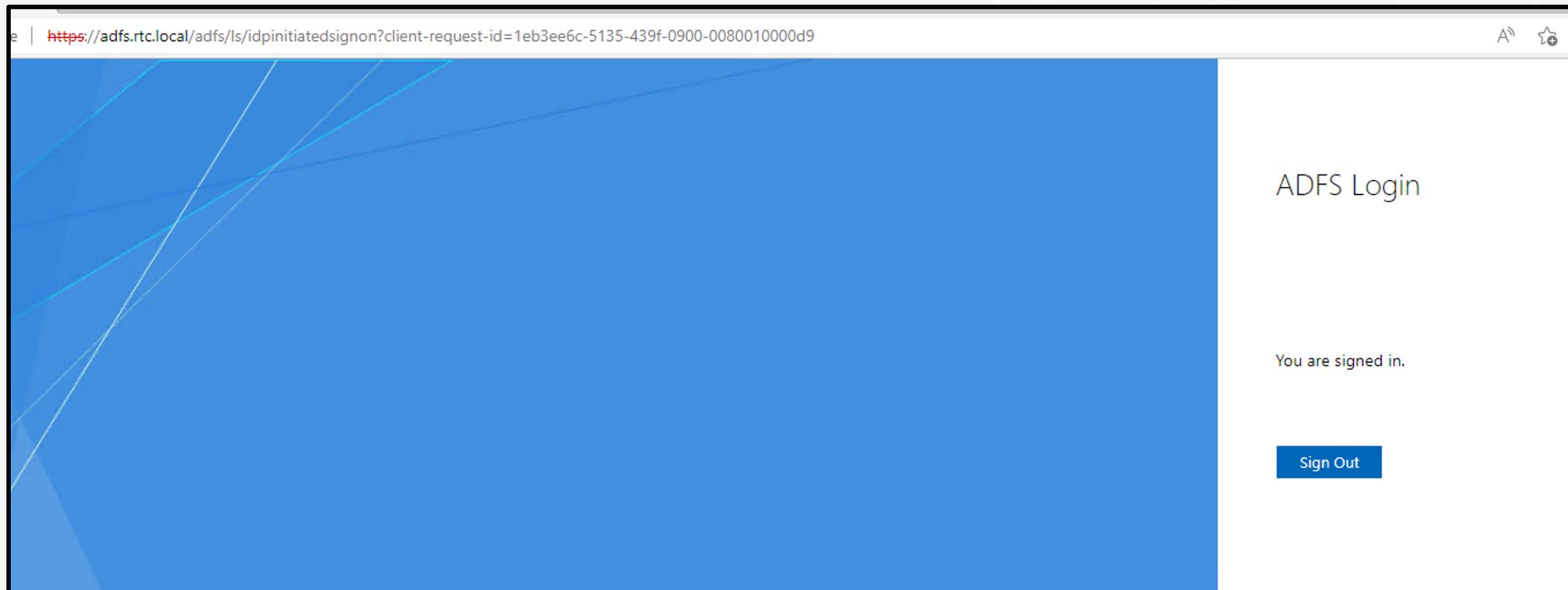
```
% python3 adfs.py --trusted_cert adfs.pfx --pxf_password password
```



# ADFS Security Auditing logs

- Generate authentication events for success and failure

Credit to Roberto Rodriguez  
for his blog on enabling ADFS  
Security Auditing.



# Detection Engineering & Sentinel

- Example KQL

```
SecurityEvent  
| where TimeGenerated >= ago(1d)  
| where EventSourceName == 'AD FS Auditing'
```

The screenshot shows the Microsoft Sentinel Query Editor interface. At the top, there are various navigation and configuration buttons: Run, Set in query, Save, Share, New alert rule, Export, Pin to, and Format query. Below these is a code editor window containing the KQL query:

```
1 SecurityEvent  
2 | where TimeGenerated >= ago(1d)  
3 | where EventSourceName == 'AD FS Auditing'  
4
```

Below the code editor is a results table. The first row contains column headers: TimeGenerated [UTC], Computer, EventSourceName, Channel, Task, and Level. The second row displays a single event with the following details:

TimeGenerated [UTC]	Computer	EventSourceName	Channel	Task	Level
10/30/2022, 3:59:40.549 ...	adfs rtc.local	AD FS Auditing	Security	3	8

Expanding the event row reveals additional properties:

- TenantId: 182faaad-d36b-42c1-8b14-2aabf04ce6d2
- TimeGenerated [UTC]: 2022-10-30T03:59:40.549115Z
- SourceSystem: OpsManager
- Computer: adfs rtc.local
- EventSourceName: AD FS Auditing
- Channel: Security
- Task: 3
- Level: 8



# Extract Token Signing Certificate Used in ADFS

- *Golden SAML*

```
PS C:\Users\RTCAdmin\Desktop>
PS C:\Users\RTCAdmin\Desktop> Install-Module -Name AADInternals -Force
PS C:\Users\RTCAdmin\Desktop> Import-Module -Name AADInternals
Loading module..
[Progress Bar]
v0.7.7 by @DrAzureAD (Nestori Syynimaa)
PS C:\Users\RTCAdmin\Desktop>
PS C:\Users\RTCAdmin\Desktop> Export-AADIntADFSCertificates
WARNING: Elevating to LOCAL SYSTEM. You MUST restart PowerShell to restore adfs\RTCAdmin rights.
WARNING: Additional signing certificate is same as the current signing certificate and will no
WARNING: Additional encryption certificate is same as the current encryption certificate and will
PS C:\Users\RTCAdmin\Desktop>
PS C:\Users\RTCAdmin\Desktop> ls

Directory: C:\Users\RTCAdmin\Desktop

Mode                LastWriteTime          Length Name
----                -----          ---- 
-a----   10/30/2022  6:18 AM           2560 ADFS_encryption.pfx
-a----   10/30/2022  6:18 AM           2544 ADFS_signing.pfx
-a----   10/30/2022  12:41 AM      150609920 AzureADConnect.msi

PS C:\Users\RTCAdmin\Desktop>
```

Credit to Dr. Nestori  
Syynimaa, his **Golden SAML**  
research, and his  
AADInternals tool.

<https://aadinternals.com>



# Instrument Sysmon Log Detections with KQL

- **KQL:** union Event

1 union Event

...

**Results** Chart | Add bookmark

TimeGenerated [UTC]	Source	EventLog	Computer	EventCategory
10/30/2022, 2:10:45.911 ...	Microsoft-Windows-Sysmon	Microsoft-Windows-Sysmo...	adfs rtc.local	13
TenantId	182faaad-d36b-42c1-8b14-2aabf04ce6d2			
SourceSystem	OpsManager			
TimeGenerated [UTC]	2022-10-30T02:10:45.9118749Z			
Source	Microsoft-Windows-Sysmon			
EventLog	Microsoft-Windows-Sysmon/Operational			
Computer	adfs rtc.local			
EventCategory	13			
EventLevel	4			
EventLevelName	Information			
UserName	NT AUTHORITY\SYSTEM			



# Azure AD Join



# Azure AD Join Lab Generator

- Creates an **Azure AD Join** security lab with Azure Virtual Machines joined to Azure Active Directory.
- **Key Features**
  - Creates optional number of Windows 10 Endpoints and automatically joins them to Azure Active Directory
  - Deploys a system assigned Identity and optional user assigned identity on all Azure VMs
  - Creates simulated Azure AD users with role of Virtual Machine Administrator Login and Virtual Machine User Login
- **Example Usage**

```
% python3 aadjoin.py --endpoints 1 -u rtcfingroup.com
```



# Detection Engineering & Sentinel

- Azure Portal: *User sign-ins (non-interactive)*
- Example KQL: ***union AADNonInteractiveUserSignInLogs***
- Connection IP addresses appear sourced from an Azure data center IP

The screenshot shows the Azure portal interface. On the left, there's a navigation bar with 'User sign-ins (interactive)', 'User sign-ins (non-interactive)' (which is underlined in blue), 'Service principal sign-ins', and 'Managed identity sign-ins'. Below this, a table displays sign-in logs. The columns are: Date, Request ID, Username, Application, Status, IP address, and Resource. Two rows are visible: one for an aggregate sign-in and one for a specific Bing sign-in from an Azure data center IP. On the right, a KQL query is shown in the 'Format query' pane:

```
1 union AADNonInteractiveUserSignInLogs
```

Below the query, the 'Results' tab is selected, showing a table with columns: TimeGenerated [UTC], OperationName, OperationVersion, Category, and ResultType. Two log entries are listed, both from 10/26/2022 at 9:26 PM, categorized as 'NonInteractiveUserSignInLogs' with ResultType 0.

Date	Request ID	Username	Application	Status	IP address	Resource
> 10/26/2022, 9:00:00	Aggregate	valeriesummers@rtcf...	Microsoft Application	Success	40.117.189.207	Microsoft Device Dir...
> 10/26/2022, 9:00:00	75c0b705-ce2d-4b19-8...	valeriesummers@rtcf...	Bing	Success	40.117.189.207	Microsoft Graph

TimeGenerated [UTC]	OperationName	OperationVersion	Category	ResultType
> 10/26/2022, 9:26:37.786 PM	Sign-in activity	1.0	NonInteractiveUserSignInLogs	0
> 10/26/2022, 9:25:35.964 PM	Sign-in activity	1.0	NonInteractiveUserSignInLogs	0

# Purple Teaming Cloud Identity



# PurpleCloud Use Cases

- *PurpleCloud* enables anyone to auto-create an Azure AD security lab for a variety of use cases:
- Create an Azure AD Lab mirroring customer tenant, to practice privilege escalation
- App Consent phishing campaigns + Social Engineering
- Create the lab with exact number of Azure AD users, to practice recon tooling, username enumeration, password spraying behavior
- Blue teams to instrument Azure sign-in logs correctly + Detection Engineering + Purple Teaming exercises
- R&D security research for new vulnerabilities or techniques

PurpleCloud can be used for Purple Teaming exercises!



# Demo



# Free SANS Workshop: Building an Azure AD Pentest lab for Red Teams

- Free guided scenario vulnerability lab with playbook
- Registration URL: <https://www.sans.org/webcasts/sans-workshop-building-azure-pentest-lab-red-teams>

The slide features a white background with a red header bar. On the left, the SANS logo is displayed vertically. The main title "Building an Azure AD Pentest lab for Red Teams" is centered in large red text. Below the title, the names "Jason Ostrom | Aaron Cure" and "Copyright 2022 SANS Institute | All Rights Reserved" are shown in smaller black text. In the bottom right corner, there is a small circular logo with a stylized "Q" and diagonal lines.

SEC588

SANS Workshop Series

SANS

Building an Azure AD  
Pentest lab for Red Teams

Jason Ostrom | Aaron Cure  
Copyright 2022 SANS Institute | All Rights Reserved

# Thank you for joining! Thank you BSidesDFW!

- **Contact Information**

- Jason Ostrom
- @securitypuck
- jostrom@stora.io



# References



# References

- <https://posts.specterops.io/azure-privilege-escalation-via-service-principal-abuse-210ae2be2a5>
- <https://dirkjanm.io/azure-ad-privilege-escalation-application-admin/>
- <https://cyb3rops.medium.com/about-detection-engineering-44d39e0755f0>
- <https://microsoft.github.io/Azure-Threat-Research-Matrix/>
- <https://m365internals.com/2021/11/30/lateral-movement-with-managed-identities-of-azure-virtual-machines/>
- <https://techcommunity.microsoft.com/t5/microsoft-sentinel-blog/enabling-ad-fs-security-auditing-and-shipping-event-logs-to/ba-p/3610464>

