PowerShell and DevOps Global Summit 2019

Hands On Lab: Hunting PowerShell Badness

Instructor: Ashley McGlone (@GoateePFE), Director of Technical Account Management, Tanium

> **Read this entire page first to understand the objectives and logistics of the hands on lab.**

> *The online lab hosting for this session has been sponsored by Tanium. Tanium is committed to realtime visibility and control of all enterprise endpoints. Tanium's capabilities include automating the techniques in this lab to hunt PowerShell-based malware at speed and scale. While the online lab space is branded as Tanium, no part of this lab will use or require Tanium. This is a purely PowerShell hands on lab.*

Lab Objective

This lab will give you a brief overview of a variety of PowerShell logging techniques. Blocking and prevention are not in scope. We will primarily focus on configuring PowerShell to leave fingerprints for forensic analysis, and then explore what we can find after an attack. This lab is intended to be a quick overview, not exhaustive. An internet search will reveal that this topic is both wide and deep, requiring your own further research. This lab is an entry point on your learning journey. Topics covered here are the easiest to implement in your enterprise.

TL;DR: If you have not configured PowerShell logging in your entprise, you are mostly blind to common threat actor tactics.

Accessing the Lab Guide

The lab guide and related scripts are hosted on Github here: https://github.com/GoateePFE/PowerShellSummit2019 OR https://bit.ly/2ZpwBDe You can read them from Github or clone them to your local machine for use during the lab and future reference. If you find a typo in the lab guide, then submit a pull request. :wink:

Activating Your Lab Environment

On the handout provided by your instructor you have a unique URL to your own lab instance hosted in AWS. Proceed to the URL and log in with the credentials provided on the sheet. In the middle of the screen you will see the button `View VMs` under the column title **Virtual Clients**. Click the `View VMs` button. You will use the **RDP/SSH IP** to connect into the remote lab virtual machines.

Lab Machines

The following computer names and IP addresses will be used throughout the lab:

Computer Name	RDP/SSH IP	Internal Lab IP	Operating System	Connection

Computer Name	RDP/SSH IP	Internal Lab IP	Operating System	Connection
ts1.training.com	*See*	172.31.140.125	Windows Server 2016 Datacenter	RDP
client-01.training.com	*lab*	172.31.153.93	Windows 10 Enterprise	RDP
client-05.training.com	*console*	172.31.143.98	Kali Linux	SSH/PuTTY

Lab Credentials

The following user names and passwords will be used throughout the lab:

Windows	Linux	Password
Training\Administrator	root	FastPower$hell19
Training\Student01	user	FastTechnoMusic$

Lab Outline

This hands on lab consists of four labs. The first is the required foundation for the remaining labs. The last three are a *choose your own adventure* and can be completed in any order. There is not enough time to complete all labs in the session today. Choose what most interests you in the time available.

> **NOTE:** The techniques featured in this lab are not exhaustive. Other PowerShell investigation techniques exist and are encouraged. The intent of this lab is to introduce the most common to begin your journey.

- ***Lab 01* - Windows PowerShell Policies**

  Learn to implement policies and identify PowerShell fingerprints in the following locations:

  - Pipeline execution logging
  - Module logging
  - Script block logging
  - Transcription
  - PSReadLine command history

  Know the common logging evasion techniques and mitigations.

  Automate the investigation and synthesize an investigation strategy for PowerShell attacks in your environment.

- ***Lab 02* - PowerShell Core Policies**

  The following policies also work on Windows, MacOS, and Linux in PowerShell Core. This lab will specifically cover PowerShell Core on Linux. Learn to implement policies and identify PowerShell fingerprints in the following locations:

  - PSReadline command history
  - Script block logging
  - Transcription

- ***Lab 03*** **- Just Enough Administration (JEA) and Logging**

  JEA uses constrained remoting sessions to restrict PowerShell capabilities. Using provided scripts you will set up a JEA endpoint, assign permissions, and assign available commands. Implementing JEA is documented online, so this lab will specifically focus on how to track the activity within a JEA session.

  - Create and test a JEA experience
  - Identify activity of a JEA session through logs covered in *Lab 01*
  - Use the JEA-specific transcription output path
  - Track down remoting sessions in the WinRM log

- ***Lab 04*** **- Hunting PowerShell Badness**

  PowerShell malware toolkits are prolific today. In this lab you will apply the skills you have learned to hunt down malicious activity generated by PowerShell Empire.

  - Launch an attack from PowerShell Empire
  - Dissect the stager used to host the session
  - Investigate the logging fingerprints
  - Can you determine what happened in the attack?