

UNCLASSIFIED

DHT-1

Defensive Host Training: Host Interrogation



Instructor: Capt Jon “MAMBA” Bynum

USAF Weapons School • Nellis AFB

UNCLASSIFIED



UNCLASSIFIED

So What?

- Shake off the rust
- *Do you wanna pass TA phase???*

UNCLASSIFIED



UNCLASSIFIED

Overview

- *PowerShell*
- Metaspouse
- Sysinternals
- FRED

UNCLASSIFIED



PowerShell

- Modern command shell that includes best features of other popular shells
 - Command-line history
 - Supports Tab completion and command prediction
 - Pipeline for chaining commands
 - In-console help system
- Commonly used to build, test, and deploy solutions
- Built on the .NET Common Language Runtime (CLR)
 - All inputs and outputs are .NET objects
 - *No need to parse text output to extract information from output*
 - Built-in support for common data formats like CSV, JSON, XML



UNCLASSIFIED

PowerShell Treats Data as Objects

The diagram illustrates how PowerShell treats data as objects. On the left, three arrows labeled "Object" point from the bottom towards a table. Above the table, three downward-pointing arrows labeled "Property" point to the column headers: "Name", "DisplayName", and "Status". The table has three rows, each representing an object. The first row contains the object "spooler", its display name "Print Spooler", and its status "Running". The second row contains the object "wuauserv", its display name "Windows Update", and its status "Stopped". The third row contains the object "vss", its display name "Volume Shadow Copy", and its status "Stopped".

Name	DisplayName	Status
spooler	Print Spooler	Running
wuauserv	Windows Update	Stopped
vss	Volume Shadow Copy	Stopped

UNCLASSIFIED



3 Important Commands

- **Get-Command**
 - Used to search for installed commands
- **Get-Help**
 - Displays how-to information for commands
- **Get-Member**
 - Displays properties and methods of objects



Documentation

- **Get-History**
 - Gets a list of the commands executed
 - *Invoke-history –id #*
 - Out-File to text
- **Start-Transcript**
 - Creates a record of all or part of a PowerShell into a text file
 - *Start-Transcript –path {PATH} -append*



UNCLASSIFIED

Slice & Dice

- Where-Object
 - Selects objects from a collection based on their property values
- Select-Object
 - Select objects or object properties
- Sort-Object
 - Sorts objects by property value

UNCLASSIFIED



UNCLASSIFIED

DEMO

- Find a cmdlet to display all the services on your ops station
- Using that same cmdlet slice & dice the output to only show running services
- Sort your services by Status and select 3 properties of your choosing and then save it to a CSV

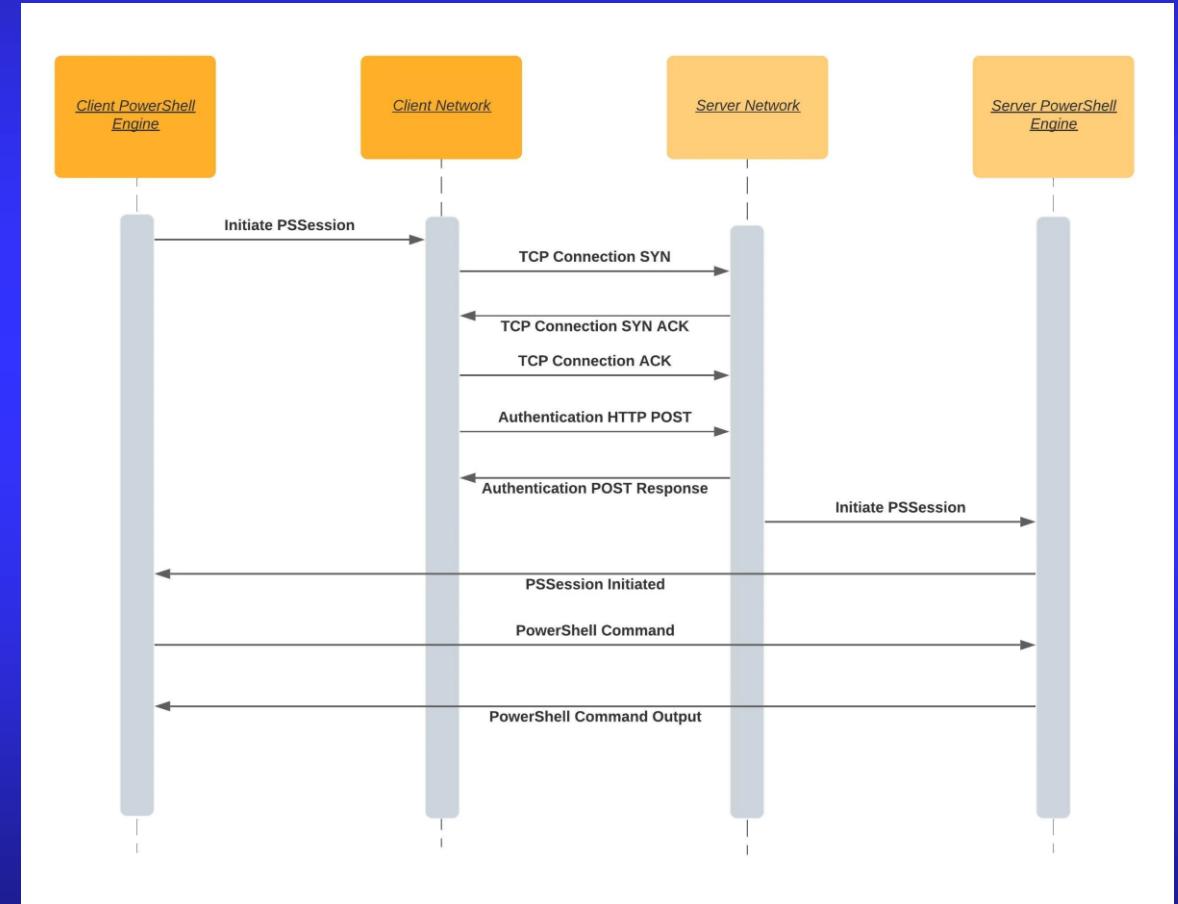


UNCLASSIFIED



PSRemoting

- Enables you to run commands on remote computers
- Client connects to destination on a WinRM listener
- Client connects over HTTP or HTTPS and authenticates
- Once complete a session is established





PSRemoting Dependencies

- Trusted Hosts
 - Set-Item WSMan:\localhost\Client\TrustedHosts –Value *
 - Basic Authentication not secure
- Creds
 - Get-Credential
 - Gets a credential object based on a user name and password
 - Stored as a Secure String
- WinRM
 - Windows Remote Management
 - Must be running and allowed through the firewall
 - *Winrm qc*



How Do We Make a Connection

- **New-PSSession**
 - Creates a persistent connection to a local or remote computer
- **Enter-PSSession**
 - Starts an interactive session with a remote computer
 - Can utilize session id's to enter a persistent connection
- **\$cred = Get-Credential**
- **\$Session = New-PSSession –ComputerName [IP] –Credential \$cred**
- **Enter-PSSession –id #**



File Traversal in PSSession

- **Copy-item**
 - cmdlet copies an item from one location to another location in the same namespace
- **Copying an item to a Remote System**
 - \$Session = New-PSSession -ComputerName [IP] -Credential \$cred
 - Copy-Item "D:\Folder004\scriptingexample.ps1" -Destination "C:\Folder004_Copy\scriptingexample_copy.ps1" -ToSession \$Session
- **Copying an item from a Remote System**
 - \$Session = New-PSSession -ComputerName [IP] -Credential \$cred
 - Copy-Item "C:\MyRemoteData\test.log" -Destination "D:\MyLocalData\" -FromSession \$Session



DEMO

- Create a session on Workstation 1 using PSRemoting
- Copy any item on your ops station to Workstation 1 and then validate using Get-ChildItem
- On the remote system get the list of running services, export it to CSV and copy from the session to your ops station
- Using GCM find a cmdlet to invoke commands on the remote computer using your session
 - Hint: help [cmdlet] –Examples
 - Use –ScriptBlock to determine the PSVersion of Workstation 1





UNCLASSIFIED

Overview

- PowerShell
- *Metasponse*
- Sysinternals
- FRED

UNCLASSIFIED



MetaWhat? MetaWho?

- Cyber-Operations mission distribution framework that enables incident responders and network defense personnel to collect, normalize, and analyze defensive and IR data
- Agentless Solution
- GUI and command-line based capability
- Pre-built collectors perform data retrieval
- Multiple built-in remediation options (Windows)
 - Delete file, service removal, driver uninstallation
 - Windows Firewall rule modification
 - Forcefully disconnect user or shutdown system
- Outputs in Table View and JSON View, Elasticsearch, or Splunk



Metasponse Dependencies

- Supported OS:
 - Windows Server 2000/2003/2008/2012
 - Windows XP, Vista, 7, 8, 8.1, 10, 11
 - Linux, Solaris (Limited Collectors)
- Will not run with FIPS compliance enabled
- Domain Admin/SSH Credentials
- Each Transport has its own dependency/requirement



Collectors Vs Transport

- Collectors
 - Plugin that executes on the remote system
 - Native PowerShell scripts, native executables, Python scripts, WMI scripts, .NET binaries, or Batch scripts
 - Reliant on Metaspouse Transports
- Transports
 - Plugin that provides communication methods to remote system
 - Configuration changes may be required
 - Certain Collectors require specific transports



UNCLASSIFIED

Common Collectors

- Accounts
- Autoruns
- Logins
- USB Drive
- SysInfo
- NetStat
- Survey

UNCLASSIFIED



UNCLASSIFIED

Rapid Analysis For Incident Response (RAIR) Collector

- Collector enumerates the memory of each process and loaded module
- Uses the process's Virtual Address Descriptor (VAD) tree enumerate allocated memory
- Performs deep inspection to detect generic indicators of:
 - Hidden and injected modules
 - Code modifications
 - Weak or modified memory region protection
 - Hidden processes
 - Function hooks
 - Dynamically allocated code
 - Packed Code
- Output only readable in JSON

UNCLASSIFIED



Help

- All pre-installed plugins have READMEs
 - Included in Metaspouse Folder
- Web UI host a Wikipage

Metasponse Job List Plugins New Job Wiki

Survey (collectors/survey)

Contents

- 1. Output
- 2. Analysis
- 3. Plugin Options

The Survey collector gathers details about processes, services, and drivers from the Remote Host. Survey uses remote Windows Management Interface calls to query system attributes.

Output

The Survey Collector will save objects to three collections, depending on the survey processes, survey drivers, and survey services options. Each record in the collections corresponds to a single instance on the Remote System. For example, each object in the `processes` collection will be a single process running on a single remote system.

Analysis

The Survey Collector will perform analysis on collected processes, services, and drivers after the Job has executed if the job.analysis option is set to `True`.

Plugin Options

- survey.drivers - Retrieve list of loaded drivers
- survey.processes - Retrieve list of running processes
- survey.services - Retrieve list of running services

Plugin Information

Id	aa.collectors.survey
Name	Survey
Path	collectors/survey
Type	<input checked="" type="radio"/> Collector <input type="radio"/> Suite Plugin
Version	1.6.0
Author	Metaspouse Team < metaspouse@airforce.com >
Collections	<ul style="list-style-type: none">driversprocessesservices
Dependencies	<ul style="list-style-type: none">transport.survey
Provides	<ul style="list-style-type: none">collect.survey
Altitude	10000



If You Can't Get A Job Create One!

- Show collectors
 - Displays all the collectors available (optional if you know which one you want)
- Use collectors/accounts
- Use transports/mswmi
- Use transports/smb
- Set job.rhost [IP]
 - Can be a subnet, list (set job.rhost < ip.txt), or a single command separated by a white space
- Set job.domain [domain name]
- Set job.user [username]
- Set job.password



If You Can't Get A Job Create One!

- Set job.log.level debug
- Job rename [name]
 - Must be unique
- Check deps
- Show options
- Schedule now
 - Jobs are set for auto pickup completion time varies per collector



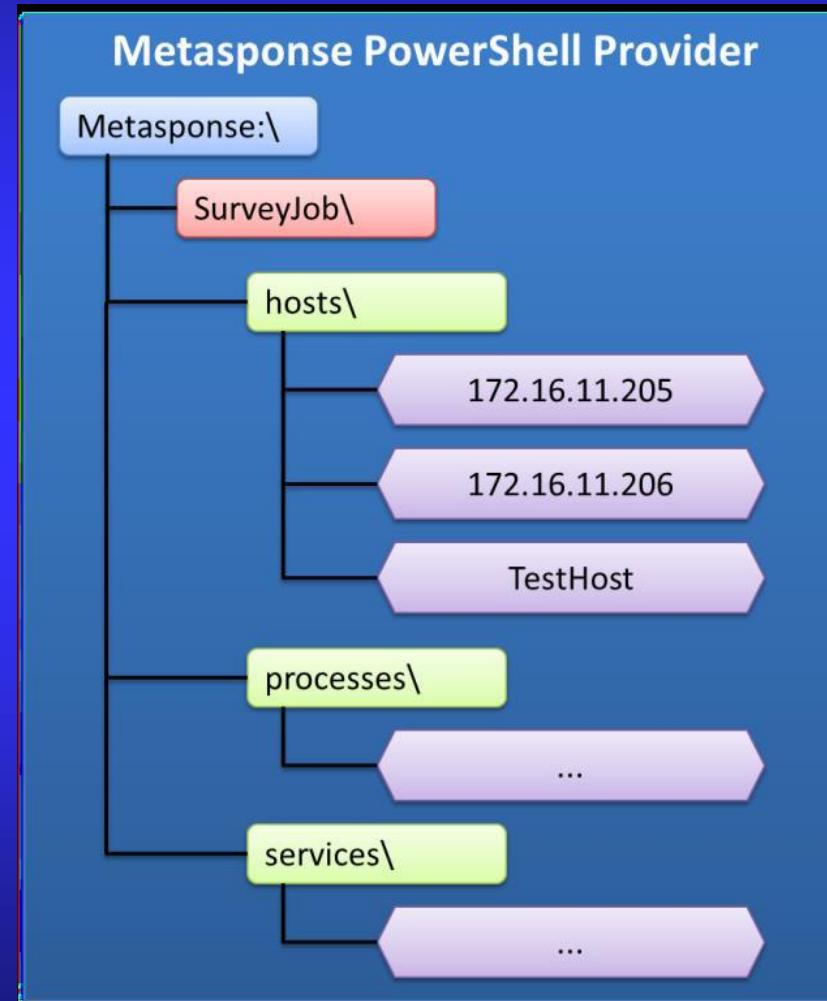
Where Dat Data At?

- Data that is collected during Job execution is enumerated, normalized, and saved to the Metaspouse Datastore.
- The Metaspouse Provider is the official method of interacting with the Metaspouse Datastore.
 - Hint: If you can use PowerShell you can use the Provider
 - Existing PowerShell cmdlets work within the Provider
 - Execution Policy must be set or the Provider will fail to launch
 - The Metaspouse Provider, which transparently wraps the Datastore collected objects and structures as PowerShell objects and provides a physical filesystem interface



UNCLASSIFIED

Metaspouse Provider File Structure



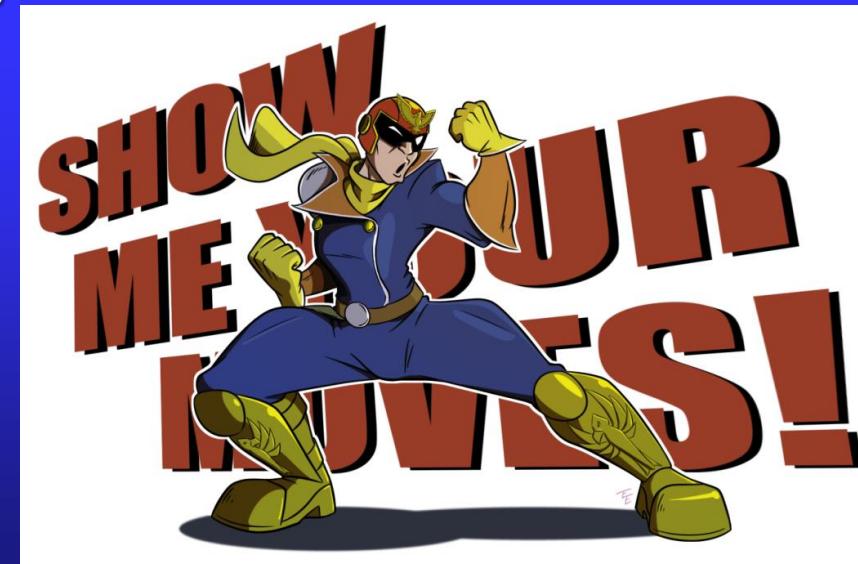
UNCLASSIFIED



UNCLASSIFIED

Demo

- Execute the Accounts Collector against Workstation 1
 - In the Provider Slice & Dice your output to show you only the local admin account
- Execute the Survey Collector against Workstation 1
 - In the Provider Slice & Dice your output to show only 5 services, PID, Path, and HostIP then export your results into a CSV



UNCLASSIFIED



UNCLASSIFIED

You Wanna Get Fancy Lets Get Fancy

- Metasploit Macros are a quick way of automating a list of steps by typing a single command
- Macros will allow the use of macro arguments and variables
- Macros are built similar to jobs
 - macro autoruns
 - use collectors/autoruns
 - use transports/mswmi
 - use transports/smb
 - job rename \$1
 - set job.rhost [IP] or \$hosts
 - End
- Var host = “host-1 192.168.0.100”
 - Optional not required

UNCLASSIFIED



UNCLASSIFIED

You Wanna Get Fancy Lets Get Fancy

- To Run your Macro
 - [Macro Name] [Macro Argument 1] ...
 - Schedule now
- Automate your execution by saving your Macros in a script file
 - Path: C:\Program Files\Metasponse\Core
 - Extension: .rc
 - In the console type:
 - Include [macro script name]
- Check to ensure your macros are loaded
 - Macro –s [macro name]
 - Gives a description
 - Macro –l
 - List all macros

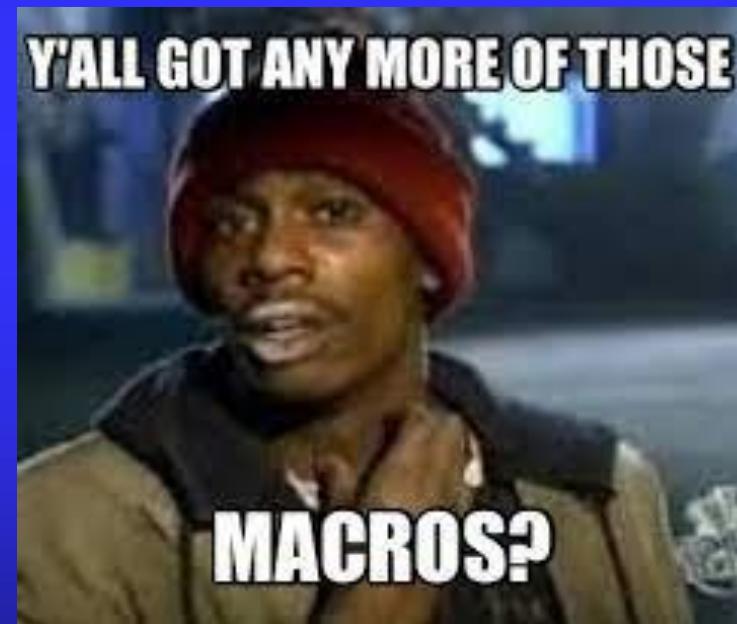
UNCLASSIFIED



UNCLASSIFIED

Demo

- Build a Macro using any collector and execute it against Workstation 1



UNCLASSIFIED



UNCLASSIFIED

Overview

- PowerShell
- Metasploit
- *Sysinternals*
- FRED

UNCLASSIFIED



SysInternals Demo

- Copy the SysInternals Suite Folder onto Workstation 1
 - Execute Strings against a file on the workstation
- Execute autoruns via the command line on Workstation 1
- Delete SysInternals Suite Folder on Workstation 1



UNCLASSIFIED

Overview

- PowerShell
- Metasploit
- Sysinternals
- *FRED*

UNCLASSIFIED



UNCLASSIFIED

FRED Demo

- Copy FRED 3.3 onto Workstation 1
- Execute FRED on Workstation 1
- Familiarize yourself with its output

UNCLASSIFIED



UNCLASSIFIED

Summary

- PowerShell
- Metasploit
- Sysinternals
- FRED

UNCLASSIFIED



UNCLASSIFIED

Questions?

- Instructor's name: Capt Jon "MAMBA" Bynum
- Instructor's address: USAF Weapons School
4269 Tyndall Avenue
Nellis AFB NV 89191-6062
- Instructor's phone: (702) 679-2215
- Instructor's e-mail: jon.bynum@us.af.mil

UNCLASSIFIED

UNCLASSIFIED

DHT-1

Defensive Host Training: Host Interrogation



Instructor: Capt Jon “MAMBA” Bynum

USAF Weapons School • Nellis AFB

UNCLASSIFIED