

ELK: Threat Hunting and Intelligence

Date: 05.07.2025

Project: ELK: Threat Hunting and Intelligence

By Thomas Lium

Introduction.....	2
Project Objectives.....	3
Lab machines overview	3
Disclaimer.....	3
Threat hunting and event correlation	3
1. Initial Access.....	3
Hunting Brute Force via SSH	4
Hunting Web based Remote Code Execution	6
Hunting Phishing Links and Attachments.....	10
2. Execution.....	13
Hunting suspicious usage of CLI tools	13
3. Command and control.....	16
Hunting Command and Control over DNS.....	17
Hunting Command and Control over Encrypted HTTP	19
Threat intelligence enrichment (IP/domain/hash analysis).....	22
Domain/IP threat intelligence	23
File hash threat intelligence.....	31
References and Tools	36

Introduction

This project involves conducting a multi-stage threat hunting and intelligence investigation within controlled SIEM lab environment using Elastic Stack (ELK). The primary objective is to demonstrate how a SIEM platform can be used to **proactively hunt threats**, enrich findings with **Open Source intelligence tools**, and map attacker behavior to the **MITRE ATT&CK framework**.

The **Elastic Stack (ELK)** is a set of tools used to collect, store, search, and visualize log and event data. It includes **Elasticsearch** for fast search, **Logstash** for data processing, and **Kibana** for visualization. It's commonly used for log analysis, monitoring, and security (SIEM) purposes.

By combining Elastic ELK's search capabilities with threat intelligence platforms such as VirusTotal, this project shows how defenders can turn raw data into actionable insights during an attack scenario.

Project Objectives

The lab setup consists of multiple virtual machines endpoints configured with logging agents that simulate log generation and forwarding under a cyber attack. An attacker machine is used to carry out realistic offensive actions in stages from the **MITRE ATTACK** framework such as **Initial access**, **Execution** and **Command and Control**. Elastic Stack (ELK) can be used as the central SIEM platform to ingest logs from these systems and analyze the attacker tactics.

Objectives to be explored in this project:

- Threat hunting and event correlation
- Mapping observed behaviors to MITRE ATT&CK techniques
- Threat intelligence enrichment (IP/domain/hash analysis)

Lab machines overview

Host	Operating System	Purpose
JUMPHOST	Ubuntu 20.04	Serves as the Bastion server for managing access to the internal network from an external network.
WEB01	Ubuntu 20.04	The external-facing web application
WKSTN-1	Windows 10	One of the workstations used by the employees.
WKSTN-2	Windows 10	One of the workstations used by the employees.

Disclaimer

This project is for educational purposes only. All scenarios and activities were conducted within the controlled virtual environment hosted on my local network. No real systems, networks, or users were involved or harmed.

Threat hunting and event correlation

1. Initial Access

The Initial Access Tactic (TA0001) of the MITRE ATTACK framework represents the adversaries' techniques and strategies to breach an organisation. This stage of an attack cycle focuses on delivering the payload to the target system or network. The primary objective during this phase is to gain a foothold in the network, which can be achieved through a variety of means.” (TryHackMe, 2025). Includes some of the following tactics:

- Social Engineering techniques such as phishing.
- Exploiting vulnerabilities through public-facing servers.
- Brute forcing credentials through exposed authentication endpoints.
- Installing software with hidden malicious code.

With an understanding of the Initial Access tactic and how adversaries might attempt to gain a foothold in an organisation's network or system, our next focus is hunting these initial access attempts. Our goal is to identify signs of the various methods outlined above. Hence, we will use the following scenarios to explore hunting for Initial access tactics:

- Brute-forcing attempts via SSH.
- Exploitation of a web application vulnerability.
- Phishing via links and attachments.

Hunting Brute Force via SSH

Starting with this scenario, we will use the filebeat-* index of ELK SIEM and hunt for brute-forcing attempts via SSH on our jumphost server on July 3, 2023.

Host	Operating System	Purpose
JUMPHOST	Ubuntu 20.04	Serves as the Bastion server for managing access to the internal network from an external network.

Brute-forcing attacks are focused on authentication events, which generate several failed attempts before successfully retrieving a valid credential. We will hunt for activity for failed logins:

host.name: jumphost AND event.category: authentication AND system.auth.ssh.event: Failed

1. Set the timestamp to July 3.

2. Set the index to filebeat.

3. Set the Table Index (filebeat), Rows (source. ip and user. name), and Metrics (count).

4. Use the KQL query to list all failed SSH auth events on the Jumphost server:

"host.name: jumphost AND event.category: authentication AND system.auth.ssh.event: Failed"

Top values of source.ip	Top values of user.name	Count of records
167.71.198.43	dev	1,133
218.92.0.115	root	746
190.123.34.126	root	12
190.123.34.126	admin	1
190.123.34.126	centos	1
190.123.34.126	Other	16
222.75.0.116	1	1
222.75.0.116	adempire	1
222.75.0.116	ansadmin	1
222.75.0.116	Other	19
14.39.23.47	1111	1
Other	root	2
Other	dmdba	1

- Set the timestamp to July 3.
 - Set the index to filebeat.
 - Set the Table Index (filebeat), Rows (source. ip and user. name), and Metrics (count).
 - Use the KQL query to list all failed SSH auth events on the Jumphost server:
- "host.name: jumphost AND event.category: authentication AND system.auth.ssh.event: Failed"

Top values of source.ip	Top values of user.name	Count of records
167.71.198.43	dev	1,133
218.92.0.115	root	746
190.123.34.126	root	12
190.123.34.126	admin	1

Upon checking the results above, it can be observed that the table provided the count of failed login attempts on specific users, including the source of the attack. These two IP addresses and accounts are highly notable since they generated over 500 failed authentication events each within the given timeframe.

Now that we have gathered significant information about brute-force attempts, let's find a successful authentication. By doing this, we can verify if the attacks yielded successful results; in this case, the attacker accessed the Jumphost server successfully via SSH. To do this, we can replace the search query with the following:

host.name: jumphost AND event.category: authentication AND system.auth.ssh.event: Accepted AND source.ip: (167.71.198.43 OR 218.92.0.115)

This query focuses on the top 2 IP addresses where the SSH authentication event was **Accepted** using a valid credential.

Visualize Library

Create

host.name: jumphost AND event.category: authentication AND system.auth.ssh.event: Accepted AND source.ip: (167.71.198.43 OR 218.92.0.115)

KQL

Jul 3, 2023 @

+ Add filter

filebeat-*

Table

Search field names

Filter by type 0

Records

Available fields 59

@timestamp

Top values of source.ip	Top values of user.name	Top values of system.auth.ssh.event	Count of records
167.71.198.43	dev	Accepted	1

1 event found of successful brute force login from source IP “167.71.198.43”. Now that we have confirmed that the attacker from 167.71.198.43 accessed the Jumphost server using the “dev” account, we have successfully hunted an intrusion attempt on this server.

Following a threat hunter's mindset, the next step of this investigation is to identify the commands execution issued by the dev user after authenticating via SSH. But before we look into executions, let’s explore hunting for threats.

Hunting Web based Remote Code Execution

In the following scenario, we will use the packetbeat-* index and hunt for suspicious actors attacking our web application (web01) on July 3, 2023.

WEB01	Ubuntu 20.04	The external-facing web application of the emulated organisation.
-------	--------------	---

- Set the timestamp to July 3.
- Set the index to packetbeat.
- Set the Table Index (packetbeat), Rows (source.ip and http.response.status_code), and Metrics (count).
- Use the search query to list all ingress network connections to the web server: “host.name: web01 AND network.protocol: http AND destination.port: 80”

Visualize Library Create

Inspect Do

host.name: web01 AND network.protocol: http AND destination.port: 80

KQL

Jul 3, 2023 @ 00:00:00.000 → Jul 3, 2023 @

+ Add filter

packetbeat-*

Search field names

Filter by type 0

Records

Available fields 77

@timestamp

agent.ephemeral_id

Table

Top values of source.ip	Top values of http.response.status_code	Count of records
167.71.198.43	404	6,742
167.71.198.43	200	39
167.71.198.43	400	3

Table

packetbeat-*

Rows

Top values of source.ip

Top values of http.respo

Add or drag-and-dro

Top values of source.ip	Top values of http.response.status_code	Count of records
167.71.198.43	404	6,742
167.71.198.43	200	39
167.71.198.43	400	3

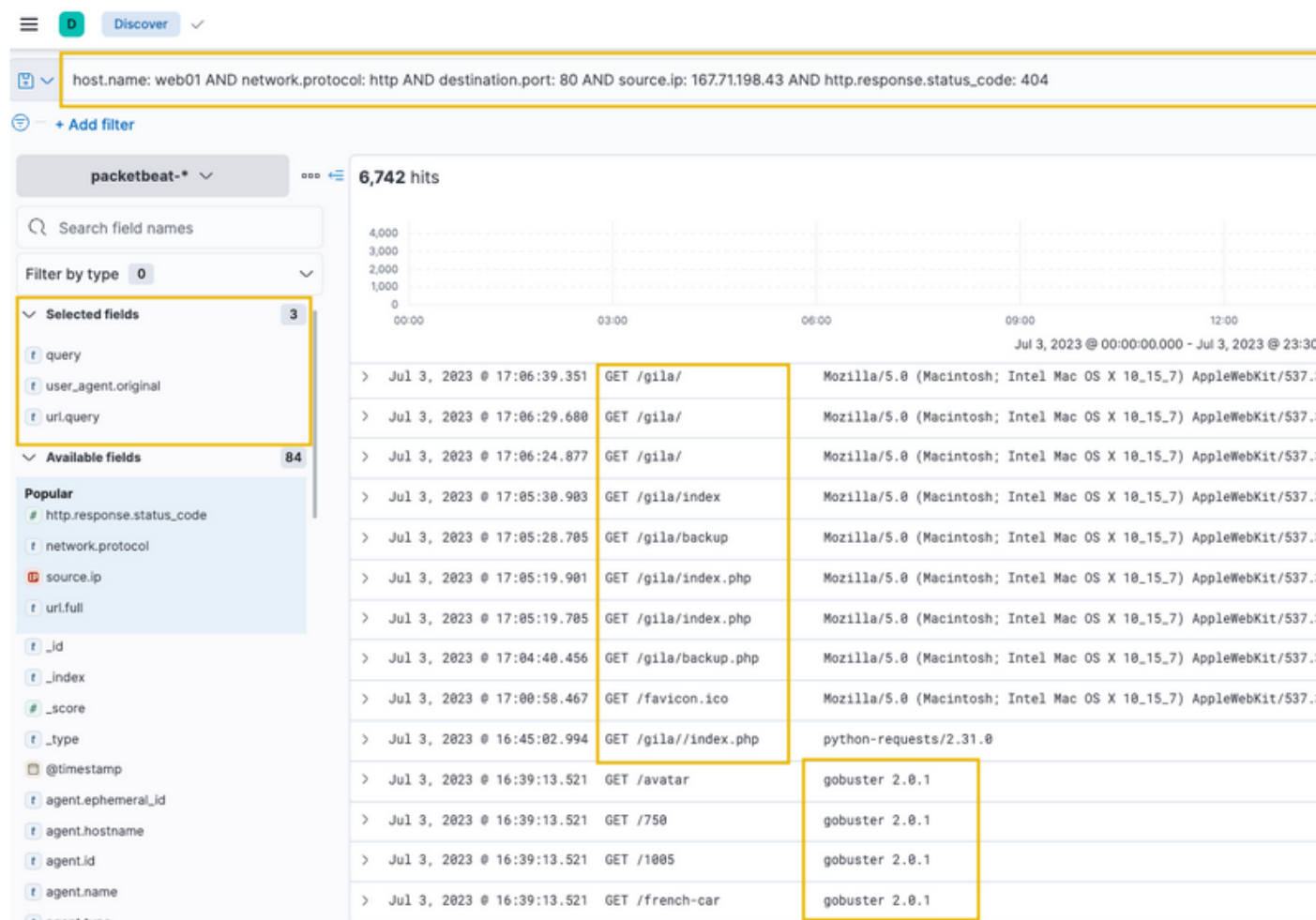
Upon checking the results above, it can be observed that the query provided a high count of status code 404, indicating a directory enumeration attempt by 167.71.198.43 since the attack produces many "404 Page Not Found" results due to its behaviour of guessing directories.

To better understand the attack, we can continue the investigation using the Discover tab with a query focused on status code 404 and the attacker's IP address. Let's use the following KQL query in the Discover tab:

```
host.name: web01 AND network.protocol: http AND destination.port: 80 AND source.ip: 167.71.198.43 AND http.response.status_code: 404
```

In addition, select the following fields and add them as a column:

- query
- user_agent.original
- url.query



Based on the results, it can be seen that the attacker used **Gobuster** (inferred via the User Agent) to enumerate the directories in the web application and eventually focused on the `/gila` directory, which may indicate that the attacker is attempting to exploit the said application.

To continue, let's replace the search query with **status codes 200, 301, and 302** to focus on valid endpoints accessed by the attacker.

host.name: web01 AND network.protocol: http AND destination.port: 80 AND source.ip: 167.71.198.43 AND http.response.status_code: (200 OR 301 OR 302)

41 hits



Time	query	user_agent.original	url.query
> Jul 3, 2023 @ 15:53:07.129	GET /	curl/7.68.0	-
> Jul 3, 2023 @ 15:53:56.007	GET /	gobuster 2.0.1	-
> Jul 3, 2023 @ 15:53:56.099	GET /test	gobuster 2.0.1	-
> Jul 3, 2023 @ 16:39:13.072	GET /	gobuster 2.0.1	-
> Jul 3, 2023 @ 16:39:13.188	GET /test	gobuster 2.0.1	-
> Jul 3, 2023 @ 16:39:13.519	GET /gila	gobuster 2.0.1	-
> Jul 3, 2023 @ 16:41:16.928	GET /gila/	curl/7.68.0	-
> Jul 3, 2023 @ 16:45:02.974	GET /gila/	<?php system(\$_GET['x']); include 'src/core/bootstrap.php'; ?>	c=admin
> Jul 3, 2023 @ 16:48:14.091	GET /gila/	curl/7.68.0	x=whoami
> Jul 3, 2023 @ 16:59:21.157	GET /gila/	curl/7.68.0	x=id
> Jul 3, 2023 @ 17:00:00.964	GET /gila/	curl/7.68.0	x=id
> Jul 3, 2023 @ 17:00:18.037	GET /gila/	curl/7.68.0	x=hostname
> Jul 3, 2023 @ 17:00:24.557	GET /gila/	curl/7.68.0	x=ifconfig

<?php system(\$_GET['x']); include 'src/core/bootstrap.php'; ?>	c=admin
curl/7.68.0	x=whoami
curl/7.68.0	x=id
curl/7.68.0	x=id
curl/7.68.0	x=hostname
curl/7.68.0	x=ifconfig

Based on the results:

- After discovering the **/gila** endpoint, the attacker focused on accessing it.
- The attacker then used a suspicious PHP code on the User-Agent field. The code uses x as a GET parameter to execute server bash commands.
- Lastly, the attacker used the x parameter to execute host commands.

With these findings, the attacker successfully compromised the web server, exploiting a Remote Code Execution vulnerability in our Gila web application. Following a threat hunter's mindset, the next step of this investigation is to identify the impact of the commands executed by the attacker via Remote Code Execution.

Hunting Phishing Links and Attachments

Phishing emails containing malicious links or attachments of malicious payloads are either downloaded or opened directly from the email client before being executed. This section will hunt for the following behaviours that satisfy this idea on the workstation machines “WKSTN-*”:

- Files downloaded using a web browser.
- Files opened from an email client (Outlook email service)

WKSTN-1	Windows 10	One of the workstations used by the employees.
WKSTN-2	Windows 10	One of the workstations used by the employees.

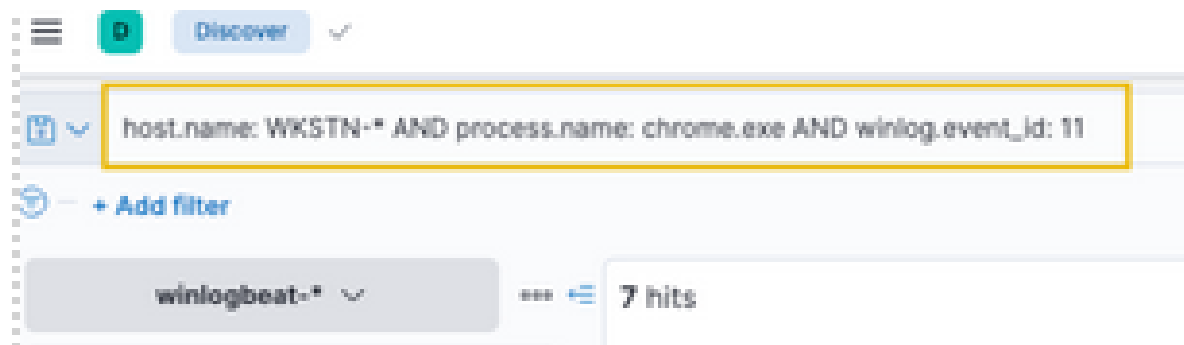
Files Downloaded using Google Chrome browser

First focus on phishing links downloaded using a web browser. By using the following KQL query, we will hunt file creations (Sysmon Event ID 11) generated by chrome.exe:

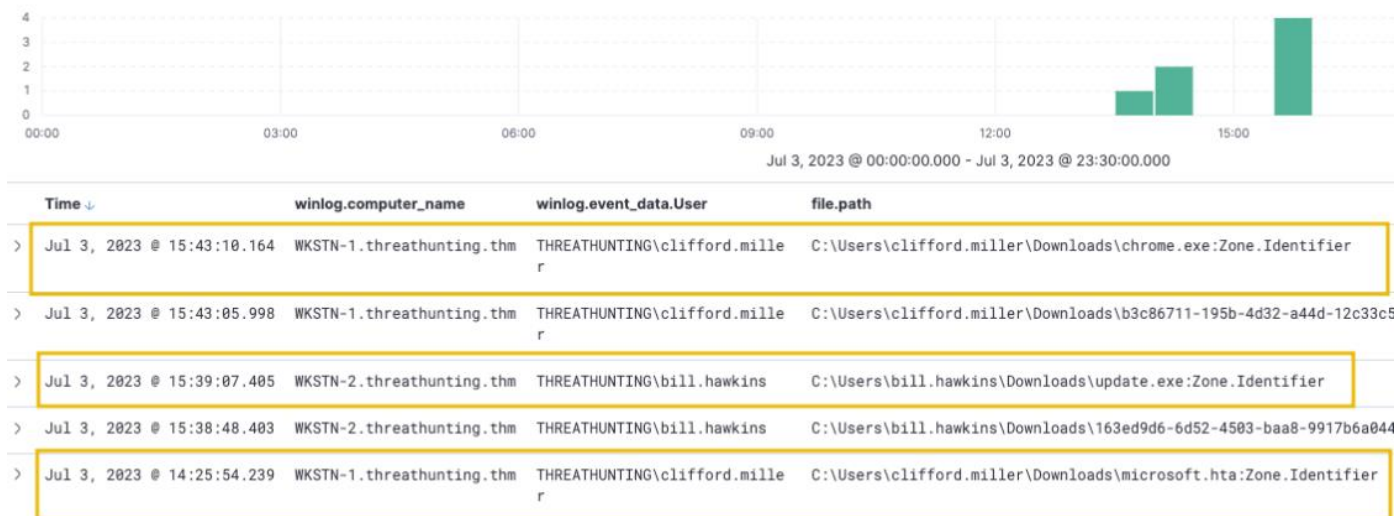
```
host.name: WKSTN-* AND process.name: chrome.exe AND winlog.event_id: 11
```

In addition, ensure that the following fields are added as columns in ELK search page:

- winlog.computer_name
- winlog.event_data.User
- file.path



7 hits



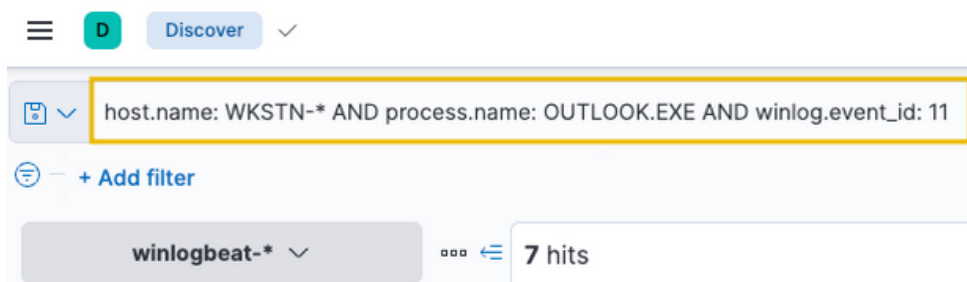
Based on the results, we can see that the following users on their respective workstations have downloaded unusual files.

User	Workstation	Files Downloaded
THREATHUNTING\clifford.miller	WKSTN-1	1. C:\Users\clifford.miller\Downloads\chrome.exe 2. C:\Users\clifford.miller\Downloads\microsoft.hta
THREATHUNTING\bill.hawkins	WKSTN-2	C:\Users\bill.hawkins\Downloads\update.exe

Files Opened using Outlook Email service

For an alternative way of hunting malware payloads delivered via phishing emails, we will hunt phishing attachments opened using an Outlook client. Using the same setup of the Discovery tab, use the following KQL query to track files created by the Outlook client:

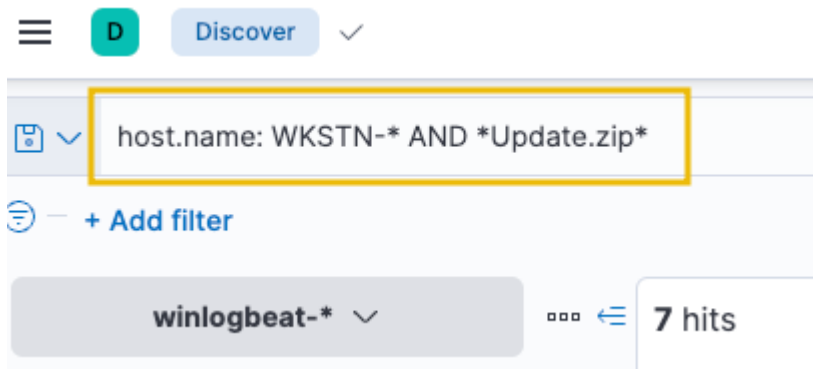
host.name: WKSTN-* AND process.name: OUTLOOK.EXE AND winlog.event_id: 11



winlog.computer_name	winlog.event_data.User	file.path
WKSTN-2.threathunting.thm	THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\SW26QUN1\Update (002).zip
WKSTN-2.threathunting.thm	THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\SW26QUN1\Update (002).zip
WKSTN-2.threathunting.thm	THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\SW26QUN1\Update (002).zip
WKSTN-2.threathunting.thm	THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\SW26QUN1\Update.zip:Zone
WKSTN-2.threathunting.thm	THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\SW26QUN1
WKSTN-2.threathunting.thm	THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\SW26QUN1\Update.zip
WKSTN-2.threathunting.thm	THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\SW26QUN1\Update.zip

“Update.zip” was downloaded through Outlook on WKSTN2 under user “bill.hawkins”

To confirm the zip file's contents, we can use the following KQL query to find events connected to it:
host.name: WKSTN-* AND *Update.zip*



winlog.event_data.User	file.path
THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Temp\Temp1_Update.zip\update.lnk
THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Temp\Temp1_Update.zip\update.lnk
THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Temp\Temp1_Update.zip\update.lnk
THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Temp\Temp1_Update.zip\update.lnk
THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\SW26QUN1\Update.zip:Zone.Identifier
THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\SW26QUN1\Update.zip
THREATHUNTING\bill.hawkins	C:\Users\bill.hawkins\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\SW26QUN1\Update.zip

Update.zip folder contains “update.lnk” file. A shortcut file (.lnk) archived to zip is a typical malware attachment threat actors use.

The next steps would be to do **threat intelligence** on the downloaded files. This would be done by checking the upcoming events for the bill.hawkins user, obtaining the file hash and use open source threat intelligence on the hash value, and also by running the file in a sandbox environment to analyze file behaviour. But more on this in final chapter.

2. Execution

“The [Execution Tactic \(TA0002\)](#) of MITRE ATTACK refers to adversaries' techniques to execute or run their malicious code in conjunction with the initial access techniques or ways of delivering the attack. This stage in the cyber-attack lifecycle is crucial as it enables the attackers to successfully run their commands remotely and continue with the series of attacks to establish further access.” (TryHackMe, 2025)

Example techniques used by adversaries are the following:

- Execution through command-line tools like PowerShell, cmd, bash
- Execution through built-in system tools
- Execution through scripting/programming tools, such as Python or PHP.

Hunting suspicious usage of CLI tools

With this scenario, we will use the winlogbeat-* index and hunt for executions of built-in Windows command-line tools, such as PowerShell and Command Prompt, from employee workstations on July 3, 2023.

WKSTN-1	Windows 10	One of the workstations used by the employees.
WKSTN-2	Windows 10	One of the workstations used by the employees.

Threat actors commonly abuse CLI in the Execution process to perform malicious commands and control the compromised host. Given this, we will hunt for behaviours that show numerous usage of command-line tools, accompanied by unusual command executions and network connections.

host.name: WKSTN-* AND winlog.event_id: 1 AND process.name: (cmd.exe OR powershell.exe)

In addition, ensure that the following fields are added as columns to aid us in our investigation:

- winlog.computer_name
 - user.name
 - process.parent.command_line
-

- process.command_line

D Discover

host.name: WKSTN-* AND winlog.event_id: 1 AND process.name: (cmd.exe OR powershell.exe)

+ Add filter

winlogbeat-*

104 hits

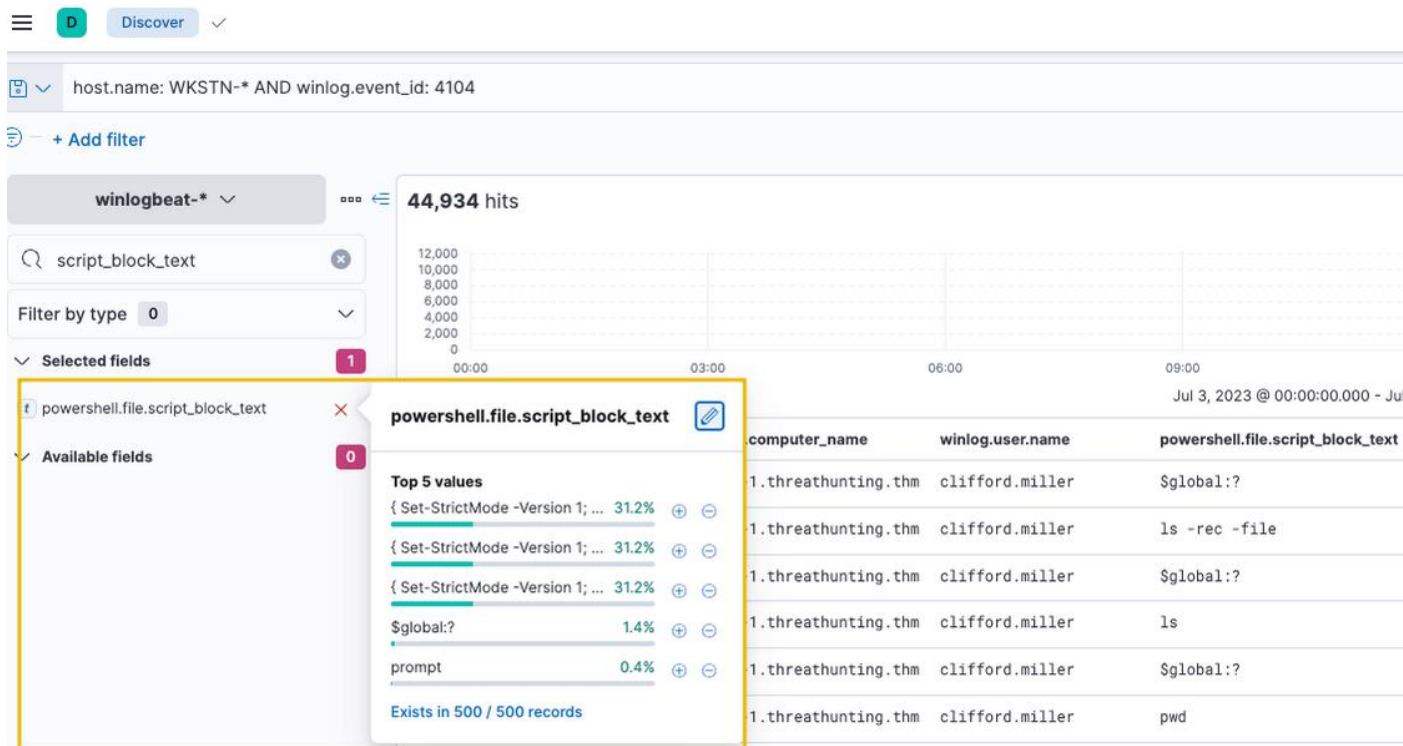
Time	winlog.computer_name	user.name	process.parent.command_line	process.command_line
Jul 3, 2023 @ 17:50:48.830	WKSTN-1.threathunting.thm	clifford.miller	cmd /c "powershell ls -rec -file"	powershell ls -rec -file
Jul 3, 2023 @ 17:50:48.615	WKSTN-1.threathunting.thm	clifford.miller	C:\Windows\Temp\installer.exe	cmd /c "powershell ls -rec -file"
Jul 3, 2023 @ 17:50:44.261	WKSTN-1.threathunting.thm	clifford.miller	cmd /c "powershell ls"	powershell ls
Jul 3, 2023 @ 17:50:44.126	WKSTN-1.threathunting.thm	clifford.miller	C:\Windows\Temp\installer.exe	cmd /c "powershell ls"
Jul 3, 2023 @ 17:50:36.453	WKSTN-1.threathunting.thm	clifford.miller	cmd /c "powershell pwd"	powershell pwd
Jul 3, 2023 @ 17:50:36.321	WKSTN-1.threathunting.thm	clifford.miller	C:\Windows\Temp\installer.exe	cmd /c "powershell pwd"
Jul 3, 2023 @ 17:50:29.866	WKSTN-1.threathunting.thm	clifford.miller	C:\Windows\Temp\installer.exe	cmd /c pwd
Jul 3, 2023 @ 17:50:24.240	WKSTN-1.threathunting.thm	clifford.miller	C:\Windows\Temp\installer.exe	cmd /c "dir C:\Users"
Jul 3, 2023 @ 17:50:20.425	WKSTN-1.threathunting.thm	clifford.miller	C:\Windows\Temp\installer.exe	cmd /c whoami
Jul 3, 2023 @ 17:50:17.969	WKSTN-1.threathunting.thm	clifford.miller	C:\Windows\Temp\installer.exe	cmd /c hostname
Jul 3, 2023 @ 17:49:28.243	WKSTN-1.threathunting.thm	clifford.miller	C:\Windows\Explorer.EXE	"C:\Windows\system32\cmd.exe"
Jul 3, 2023 @ 17:49:24.446	WKSTN-1.threathunting.thm	clifford.miller	C:\Windows\Explorer.EXE	"C:\Windows\System32\cmd.exe" /q /c del /q Drive\StandaloneUpdater\OneDriveSetup.exe"
Jul 3, 2023 @ 17:49:24.328	WKSTN-1.threathunting.thm	clifford.miller	C:\Windows\Explorer.EXE	"C:\Windows\System32\cmd.exe" /q /c del /q "

Out of the 104 hits, it can be observed that numerous commands are used that seem unusual. One example is the execution of cmd.exe by C:\Windows\Temp\installer.exe, as shown in its parent-child process relationship. It is more remarkable that the parent process binary is located from C:\Windows\Temp, a typical folder threat actors use to write malicious payloads.

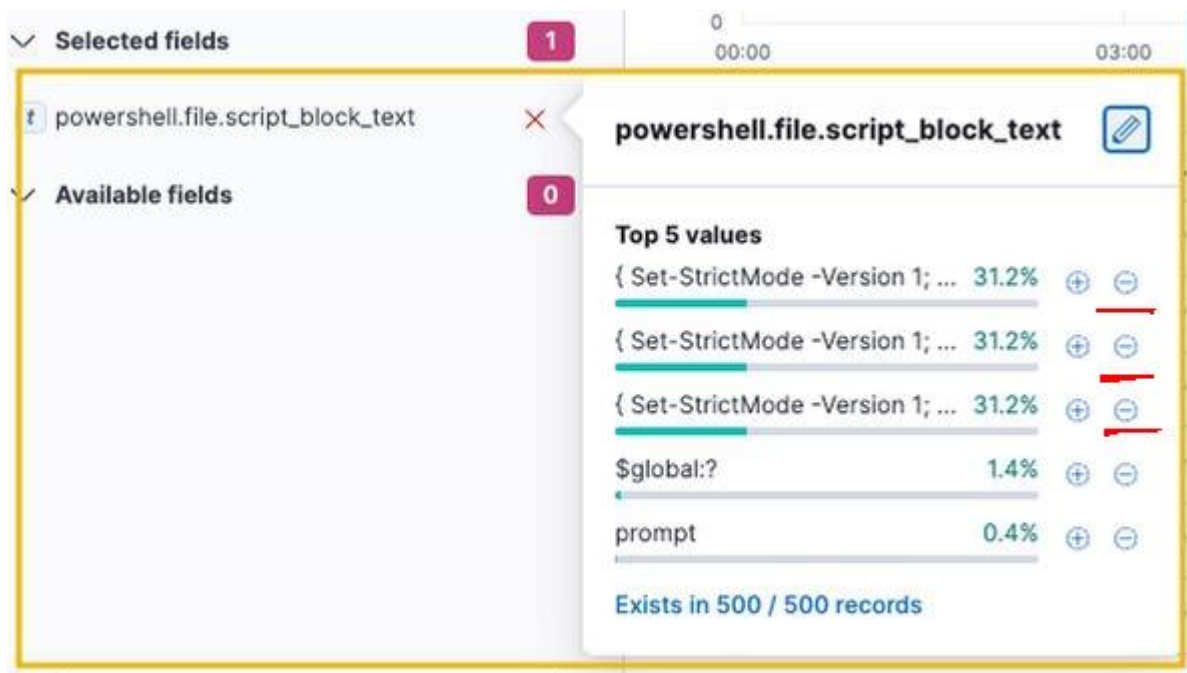
Event ID 4104 in Windows Event Logs typically refers to PowerShell script block logging events.

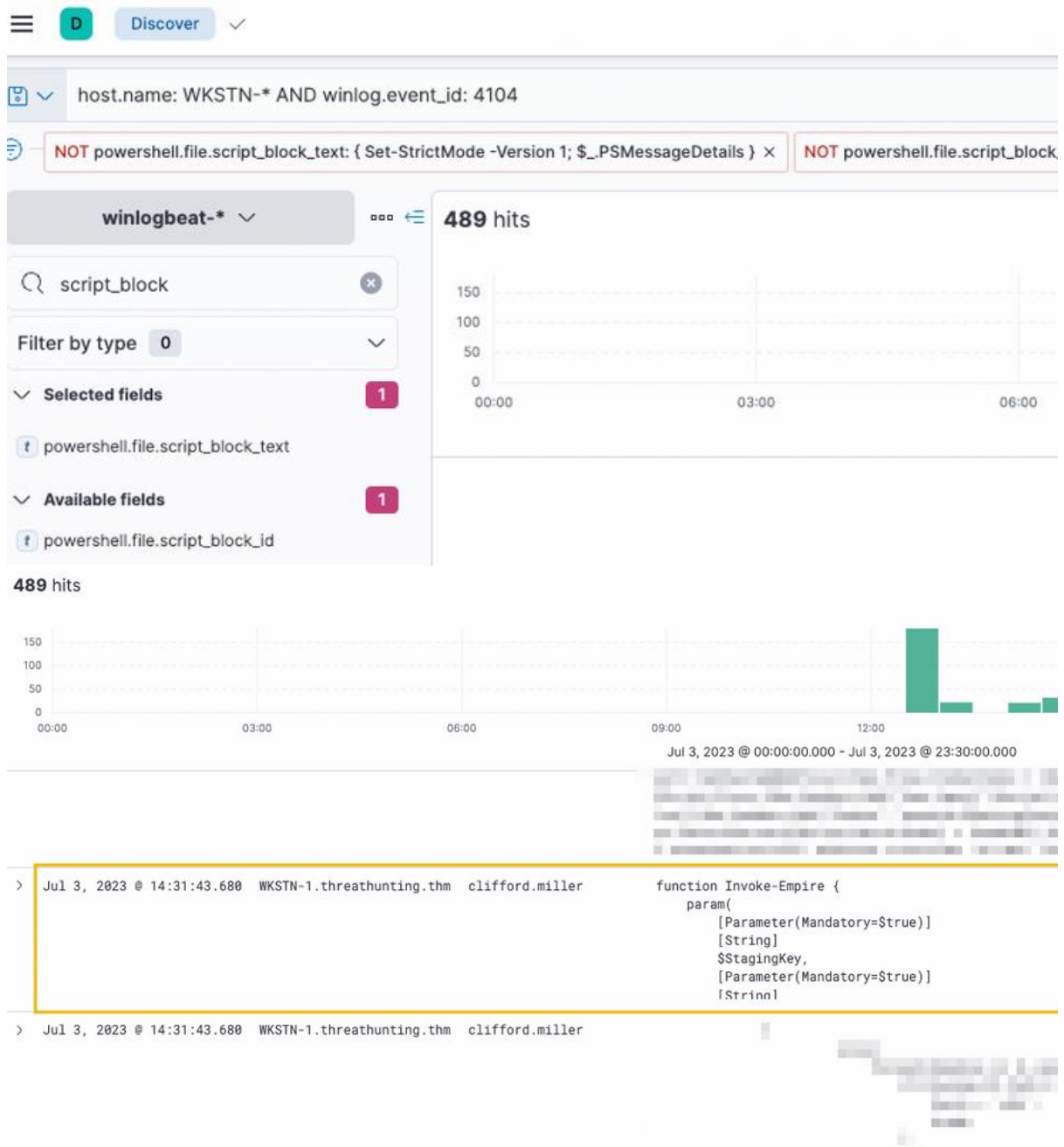
To add on PowerShell analysis, an alternative way to hunt unusual PowerShell execution is through the events generated by PowerShell's Script Block Logging. We can use the following KQL syntax to list all events generated by it:

```
host.name: WKSTN-* AND winlog.event_id: 4104
```



44,934 events. We can reduce this by removing the noise generated by the events. In this case, remove the "Set-StrictMode" events in "Selected fields" by clicking the minus button in the image below:





After applying the filters, you will see that the events have been reduced to 489 hits, which makes hunting suspicious events easier. By scrolling through the executed PowerShell scripts, it can be observed that **Invoke-Empire** (signature of Empire C2 agent) was used in WKSTN-1.

3. Command and control

"The [Command and Control Tactic \(TA0011\)](#) of MITRE ATTACK involves the methods by which an adversary communicates with the compromised systems within a target network. This is the stage at which an attacker usually directs or continuously issues remote commands to the compromised system

to fulfil the attacker's objectives, such as further internal network compromise.” (TryHackMe, 2025). Communication can occur via various channels, such as:

- Standard network protocols, such as DNS, ICMP, HTTP/s.
- Known cloud-based services.
- Encrypted custom HTTP/s server.

Hunting Command and Control over DNS

The hunt for Command and Control involves uncovering communication channels amidst regular network traffic. Adversaries use standard protocols to blend in with typical network traffic or use cloud storage services as unconventional command channels to avoid raising suspicion.

In the following sections, we will delve into strategies and techniques for hunting Command and Control activities, interpreting network events, and recognising anomalies through DNS.

Starting with this scenario, we will use the packetbeat-* index in ELK and hunt for potential C2 over DNS on July 3, 2023. In addition, we will use the winlogbeat-* index to correlate the DNS queries to identify the malicious process generating it.

C2 over DNS, or more accurately Command and Control over DNS, is a technique used by adversaries where DNS protocols are utilised to establish a Command and Control channel. In this technique, adversaries can disguise their C2 communications as typical DNS queries and responses, bypassing network security measures. Given this, we will hunt for unusual DNS query patterns based on the following:

- High count of unique subdomains
- Unusual DNS requests

To start hunting, use the Visualize Library again and create a visualisation table using Lens. Ensure that the table is configured with the following:

- Set the Table Index (packetbeat), Rows (dns.question.registered_domain and host.name), and Metrics (Unique Count of dns.question.subdomain).
- Filter the query to list all DNS queries and exclude all reverse DNS lookup requests

(.arpa domains are used for network housekeeping (like reverse lookups). In threat hunting, they are typically excluded to reduce false positives and focus on domains more likely to indicate malicious activity)

network.protocol: dns AND NOT dns.question.name: *.arpa

Visualize Library Create

network.protocol: dns AND NOT dns.question.name: *arpa

+ Add filter

packetbeat-*

Search field names

Filter by type 0

Records

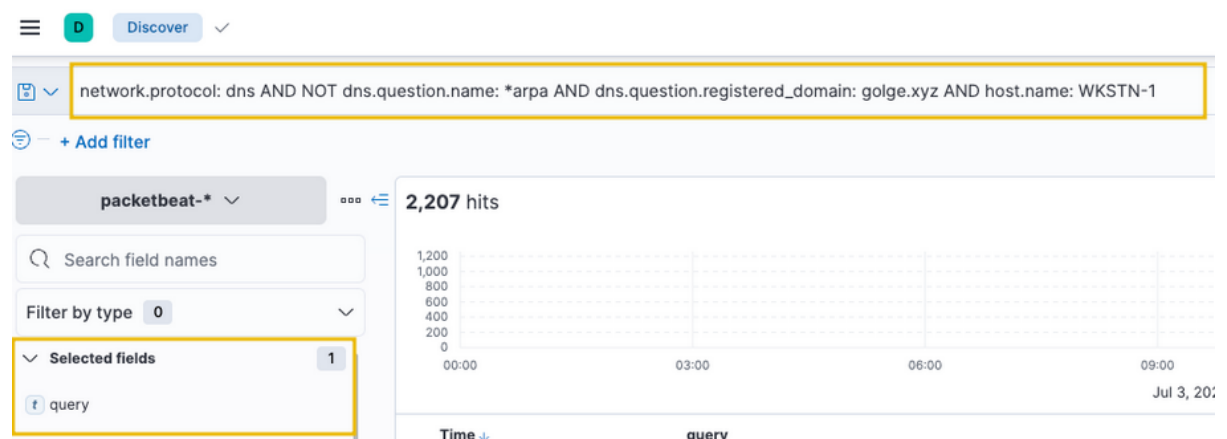
Available fields 97

Top values of dns.question.registered_domain	Top values of host.name	Unique count of dns.question.subdomain
golge.xyz	WKSTN-1	2,191
golge.xyz	DC01	1
golge.xyz	WKSTN-2	1
amazonaws.com	WKSTN-1	541

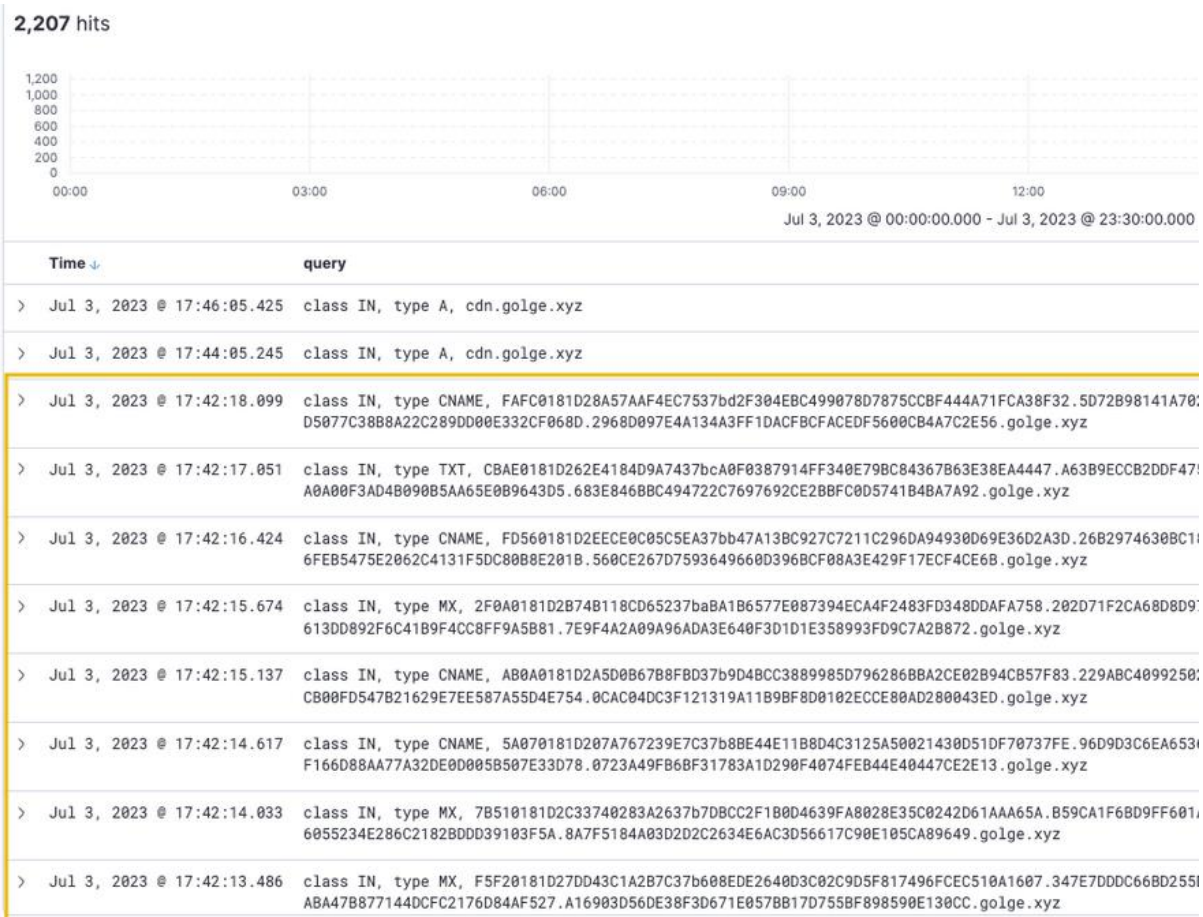
Upon checking the results above, it can be observed that an unusual domain (golge[.]xyz) queried 2191 unique subdomains, which may indicate a potential C2 over DNS activity coming from WKSTN-1.

Add in the suspicious domain and hostname to a search query in Discover tab in ELK:

network.protocol: dns AND NOT dns.question.name: *arpa AND dns.question.registered_domain: golge.xyz AND host.name: WKSTN-1



(Add the query field as a column to see its values.)



Based on the results, the workstation seems to be continuously querying on *golge.xyz, using different query types and using hexadecimal subdomains. Random unique subdomains like:

FAFC0181D2BA57AAF4EC7537BD2F304EBC49907B87B75CCBF444A71FCA38F32.5D72898141A782739E351A31C98767541BAB2ACC6716B88A6F4D296AC0F9190.golge.xyz

These subdomains are being used to **encode data in hex** and send it out via DNS queries.

These findings strongly suggest the conclusion that WKSTN-1 is compromised and likely using **DNS tunneling** method to communicate with an external C2 server under the attacker's control.

Hunting Command and Control over Encrypted HTTP

C2 over Encrypted HTTP traffic is a typical command and control type. The main notable thing about this technique is that attackers use their own C2 domain, including traffic encryption over HTTP. Given this, we will hunt for unusual HTTP traffic based on the following:

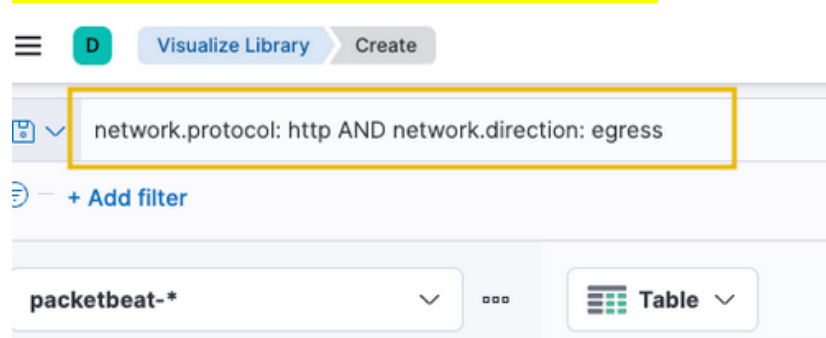
- High count of HTTP traffic to distinctive domains
- High outbound HTTP bandwidth to unique domains

To start hunting, use the Visualize Library again in ELK and create a visualisation table using Lens.

The table is configured with the following:

- Set the Table Index (packetbeat), Rows (host.name, destination.domain, http.request.method), and Metrics (count).
- Set search query to list all outbound HTTP requests using “egress” direction

network.protocol: http AND network.direction: egress



Top values of host.name	Top values of destination.dom	Top values of http.request.me	Count of records
WKSTN-2	cdn.golge.xyz	get	16,078
WKSTN-1	cdn.golge.xyz	get	6,442
WKSTN-1	download.windowsupdate.com	get	159
DC01	download.windowsupdate.com	get	159
WKSTN-2	download.windowsupdate.com	get	114
WKSTN-2	edgedl.me.gvt1.com	get	55
WKSTN-1	edgedl.me.gvt1.com	get	52

Based on the results, it is highly notable that HTTP connections to cdn.golge.xyz from both workstations are numerous. This may indicate that a continuous C2 connection has been running for an extended time. We can modify the Lens table and focus the query to cdn.golge.xyz

host.name: WKSTN-* AND network.protocol: http AND network.direction: egress AND destination.domain: cdn.golge.xyz

In addition, we can modify the rows and focus only on host.name and query fields:

Visualize Library Create

host.name: WKSTN-* AND network.protocol: http AND network.direction: egress AND destination.domain: cdn.golge.xyz KQL Jul 3, 2023 @

+ Add filter

packetbeat-* Table

Search field names

Filter by type 0

Records

Available fields 80

@timestamp

agent.ephemeral_id

Top values of host.name	Top values of query	Count of records
WKSTN-2	GET /admin/get.php	5,475
WKSTN-2	GET /login/process.php	5,335
WKSTN-2	GET /news.php	5,268
WKSTN-1	GET /login/process.php	2,168
WKSTN-1	GET /news.php	2,167
WKSTN-1	GET /admin/get.php	2,107

Based on the results, it can be observed that the volume of requests is GET requests to 3 .php endpoints.

Using the following search query in Discover page in ELK provided us with some insights regarding the associated process: `host.name: WKSTN-* AND *cdn.golge.xyz*`

Discover

host.name: WKSTN-* AND *cdn.golge.xyz*

+ Add filter

winlogbeat-* 10 hits

Search field names

Filter by type 0

Selected fields 3

- host.name
- process.name
- winlog.event_data.User

Time

> Jul 3

Select the Selected fields for a better view:

10 hits



Jul 3, 2023 @ 00:00:00.000 - Jul 3, 2023 @ 23:30:00.000			
Time ↑	host.name	process.name	winlog.event_data.User
> Jul 3, 2023 @ 14:23:47.655	WKSTN-2.threathunting.thm	powershell.exe	THREATHUNTING\bill.hawkins
> Jul 3, 2023 @ 14:26:05.342	WKSTN-1.threathunting.thm	powershell.exe	THREATHUNTING\clifford.miller
> Jul 3, 2023 @ 14:26:07.719	WKSTN-1.threathunting.thm	-	-
> Jul 3, 2023 @ 14:31:39.410	WKSTN-1.threathunting.thm	svchost.exe	NT AUTHORITY\NETWORK SERVICE
> Jul 3, 2023 @ 14:31:39.414	WKSTN-1.threathunting.thm	powershell.exe	THREATHUNTING\clifford.miller
> Jul 3, 2023 @ 14:31:41.308	WKSTN-1.threathunting.thm	-	-
> Jul 3, 2023 @ 14:44:13.974	WKSTN-2.threathunting.thm	powershell.exe	THREATHUNTING\bill.hawkins
> Jul 3, 2023 @ 15:29:23.577	WKSTN-2.threathunting.thm	powershell.exe	THREATHUNTING\bill.hawkins

Based on the results, it looks like the C2 connection to cdn.golge.xyz was established using a malicious PowerShell command from powershell.exe on both workstations "WKSTN-1" and "WKSTN-2".

Threat intelligence enrichment (IP/domain/hash analysis)

Threat intelligence is a critical part of what SOC (Security Operations Center) analysts do to stay ahead of cyber threats. Threat intelligence (TI) is **information about threats**, including:

- Malicious actors
- Threat Origin
- Tools, techniques, and procedures (TTPs)
- Indicators of compromise (IOCs)
- Vulnerabilities and exploits

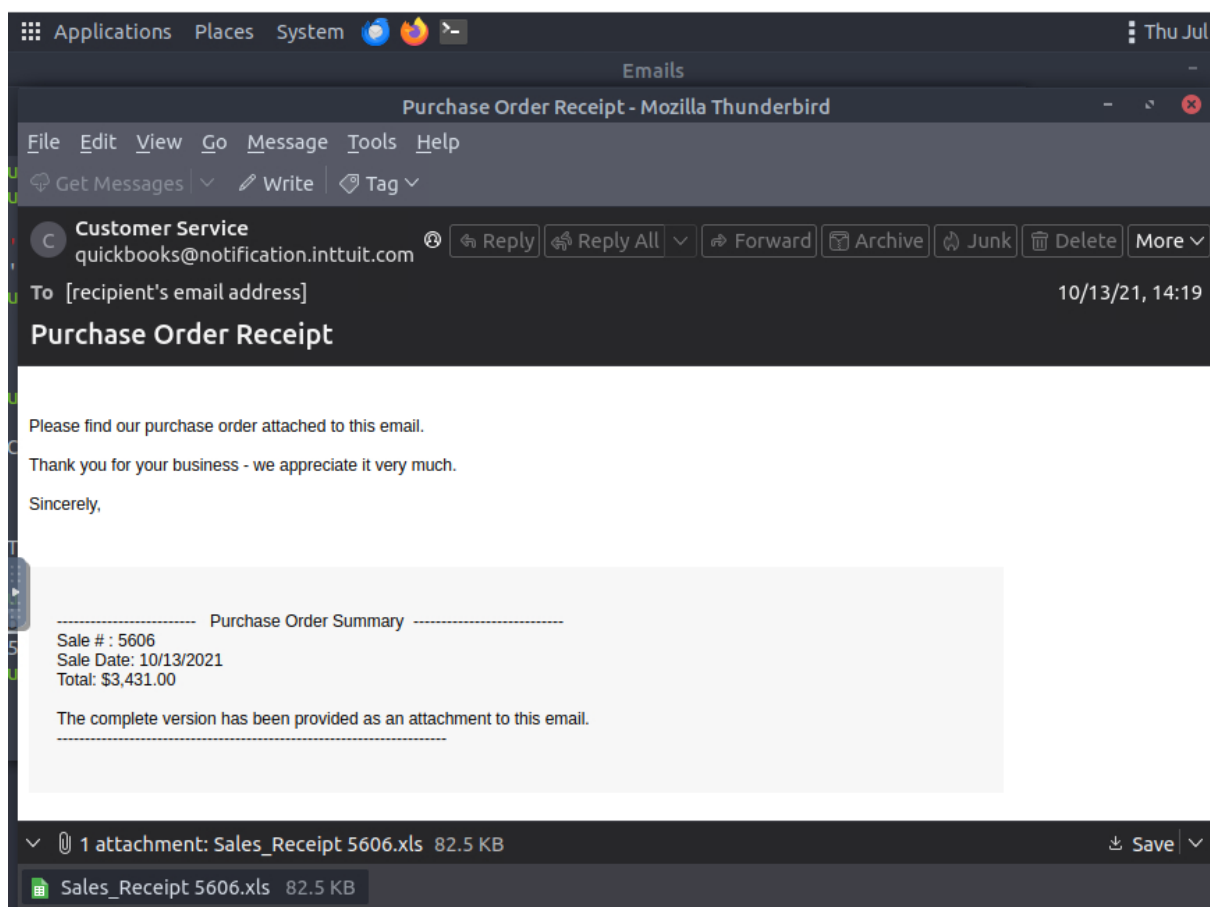
SOC analysts use TI to detect, prevent, and respond to security incidents. Analysts gather threat data from multiple sources:

- **Internal sources:**
 1. SIEM alerts (e.g., Elastic, Splunk)
 2. Firewall logs
 3. Endpoint logs
 4. Network logs
- **External sources:**
 - Correlate IOCs (IP addresses, domains, hashes) with known threats
 - Open-source intel sources (OSINT) such as:

1. VirusTotal
2. Cisco Talos
3. Urlscans.io
4. MITRE ATT&CK framework for mapping Tactics
5. Any.run (file behaviour analysis)

Domain/IP threat intelligence

Scenario: You are a SOC Analyst. Several suspicious emails have been forwarded to you from other coworkers. You must obtain details from each email to triage the incidents reported.



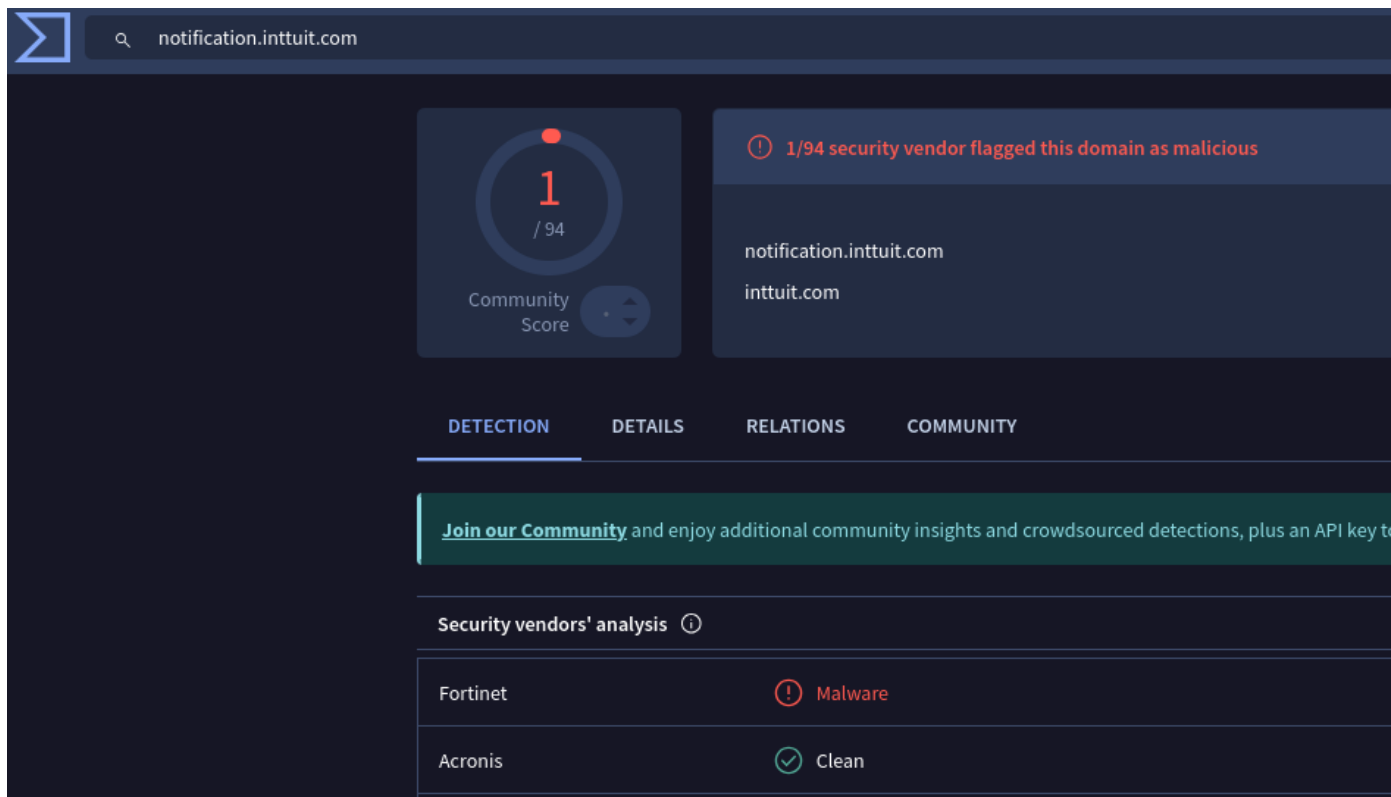
From: quickbooks@notification.intuit.com

To: (Not listed)

The email is a purchase order receipt dated 10/13/21, 14:19, containing a purchase order summary with Sale Date: 10/13/2021, Sale #: 5606, Total: \$24,431.00. It mentions that a complete version has been provided as an attachment to the email.

1 attached file: Sales_Receipt_5606.xls

Use VirusTotal tool to search up the senders domain "notification.intuit.com":



The screenshot shows the VirusTotal web interface for the domain `notification.intuit.com`. The top section displays a "Community Score" of 1/94, indicating a low score. A warning message states: "1/94 security vendor flagged this domain as malicious". Below this, the domain names `notification.intuit.com` and `intuit.com` are listed. The "DETECTION" tab is active, showing a table of security vendors' analysis. The table has two rows: Fortinet, which is flagged as "Malware", and Acronis, which is flagged as "Clean".

Security vendors' analysis ⓘ	
Fortinet	⚠ Malware
Acronis	✅ Clean

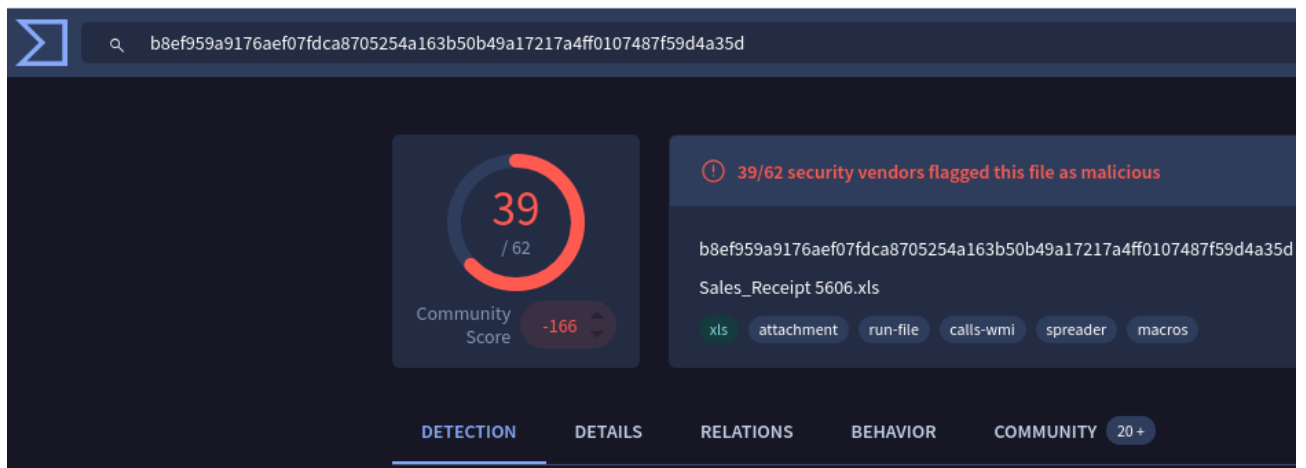
1 security vendor flagged the domain as malicious, Fortinet claim malicious content. Already a bad sign, get more information of the senders intent by analyzing the attached file "Sales_Receipt_5606.xls".

Get the file hash in SHA256 by using the "sha256sum" command in bash followed by the listed file:

```
ubuntu@tryhackme:~/Desktop$ sha256sum Sales_Receipt\ 5606.xls
b8ef959a9176aef07fdca8705254a163b50b49a17217a4ff0107487f59d4a35d Sales_Receipt
5606.xls
ubuntu@tryhackme:~/Desktop$
```

b8ef959a9176aef07fdca8705254a163b50b4a1721a4ff0107487f59d4a35d

Copy the file hash and paste in VirusTotal for analysis:



39 security vendors flagged the file hash as malicious. Highlighted keywords include: Attachment, run-file, spreader, macros.

Click on “Behaviour” section on VirusTotal:

Activity Summary Download Artifacts

Detections 3 MALWARE	Mitre Signatures 4 LOW 11 INFO	IDS Rules NOT FOUND	Sigma Rules NOT FOUND	Dropped Files 1 XML 26 OTHER 1 LNK
--------------------------------	--	-------------------------------	---------------------------------	--

Behavior Tags ⓘ

calls-wmi detect-debug-environment

Dynamic Analysis Sandbox Detections ⓘ

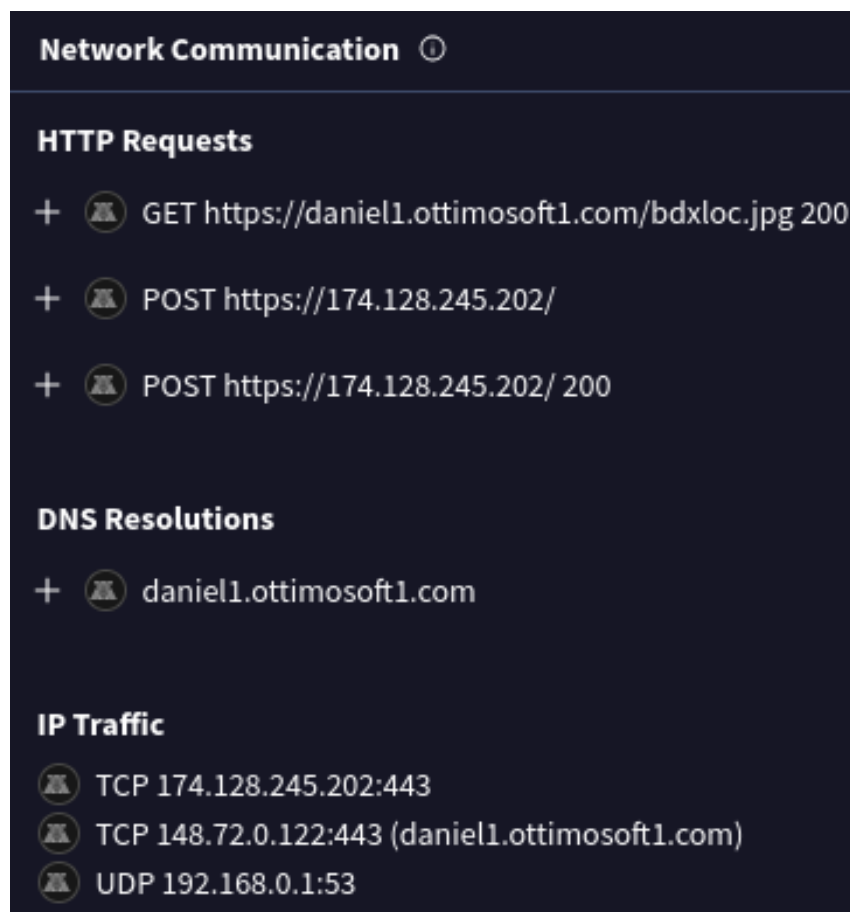
- The sandbox **VMRay** flags this file as: MALWARE
- The sandbox **DOCGuard** flags this file as: MALWARE
- The sandbox **BitDam ATP** flags this file as: MALWARE

MITRE ATT&CK Tactics and Techniques

- + Execution TA0002
- + Privilege Escalation TA0004
- + Defense Evasion TA0005
- + Discovery TA0007
- + Command and Control TA0011

File behaviour has been flagged as “malware”, includes MITRE ATTACK Tactics such as Execution, PE, Defense Evasion, Discovery and C2. This means this software is designed to execute code, escalate privileges, evade defense restrictions and contact command and control server. The malware creates up to 28 files from its execution.

Scrolling down to Network Communication section:



Network section states that the malware makes a HTTP web request to:

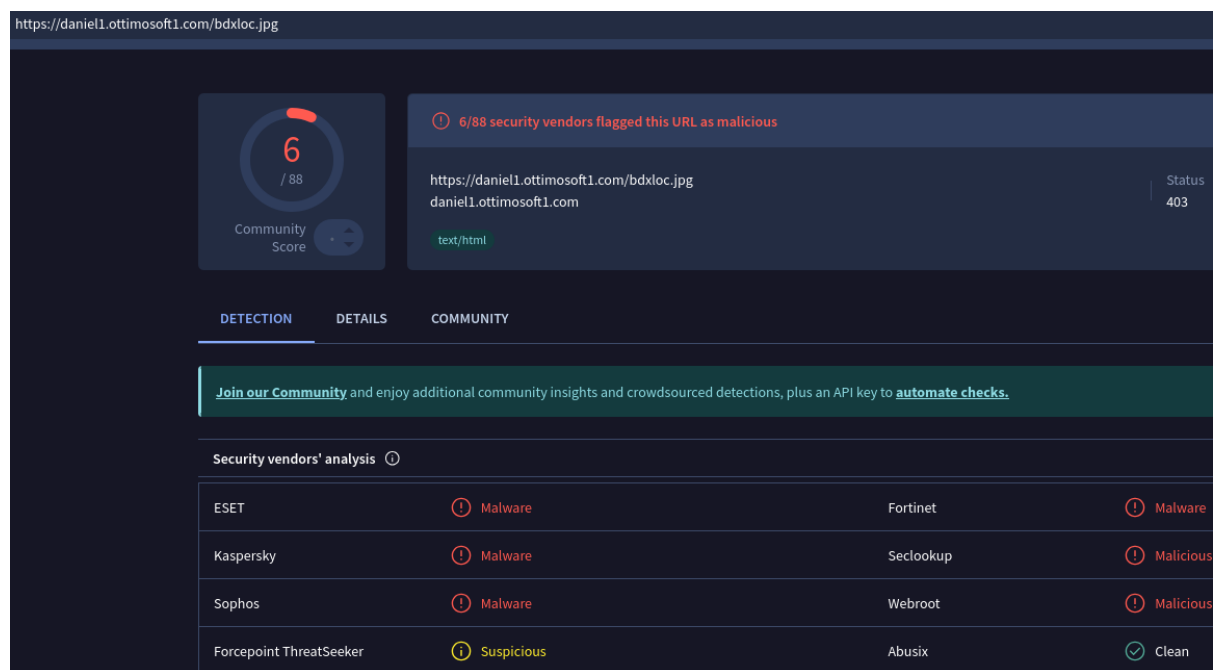
- GET <https://daniel1.ottimosoft1.com/bdxloc.jpg>
- POST <https://174.128.245.202/>
- POST <https://174.128.245.202/>

TCP traffic to IP 174.128.245.202:443 and domain “daniel1.ottimosoft1.com”

Next, do threat intelligence on the domain and IP address.

Threat intel on “daniel1.ottimosoft1.com”

Look up the listed URL <https://daniel1.ottimosoft1.com/bdxloc.jpg> on VirusTotal:



6 security vendors flagged the URL as Malicious.

Use URLscan.io to scan the listed domain. URLscan will look up registered information on the domain, useful info and a brief screenshot of the webpage:

daniel1.ottimosoft1.com

148.72.0.122 Public Scan

URL: <https://daniel1.ottimosoft1.com/>

Submission: On October 13 via automatic, source urlhaus (October 13th 2021, 2:07:57 pm UTC) — Scanned from

Summary HTTP 12 Redirects Behaviour Indicators Similar DOM Content API Verdicts

Summary

This website contacted **2 IPs** in **1 countries** across **2 domains** to perform **12 HTTP transactions**. The main IP is **148.72.0.122**, located in **Ashburn, United States** and belongs to **AS-26496-GO-DADDY-COM-LLC, US**. The main domain is **daniel1.ottimosoft1.com**.
 TLS certificate: Issued by **cPanel, Inc. Certification Authority** on October 4th 2021. Valid for: 3 months.

[daniel1.ottimosoft1.com](#) scanned **2 times** on urlscan.io

Show Scans 2

urlscan.io Verdict: No classification

Live information

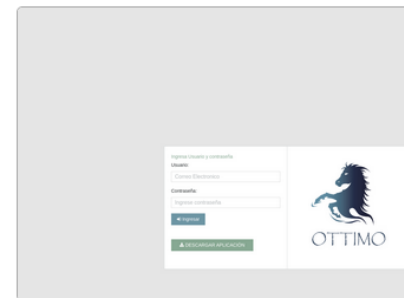
Google Safe Browsing: No classification for [daniel1.ottimosoft1.com](#)

Domain & IP information

IP/ASNs	IP Detail	Domains	Domain Tree	Links	Certs	Frames
	IP Address	AS Autonomous System				
11	148.72.0.122	26496 (AS-26496-GO-DADDY-COM-LLC)				
1	142.250.184.202	15169 (GOOGLE)				

Screenshot

Live screenshot



Page Title

Iniciar Sesión | Ottimo 2.0


Detected technologies

- Bootstrap** (Web Frameworks)
- Font Awesome** (Font Scripts)
- jQuery** (JavaScript Libraries)
- iQuerv UI** (JavaScript Libraries)

The IP address is located in Ashburn, US and is owned by a user under “Go Daddy”. The domain web page itself seems to be a login page to “Ottimo 2.0”. I tried to look up if theres a known C2 framework with the same name, but couldnt find anything. Could be a C2 login page.

I noticed this domain is a subdomain under “ottimosoft1.com”, look up the main domain itself on URLscan.io:

Look up the domain on VirusTotal:

ottimosoft1.com

2

/ 94

Community Score

2/94 security vendors flagged this domain as malicious

ottimosoft1.com

Registrar
GoDaddy.com, LLC

Cre
6 y

Malicious (alphaMountain.ai)

Malware Sites

top-1M

DETECTION

DETAILS

RELATIONS

COMMUNITY

Security vendors' analysis

alphaMountain.ai

Malicious

Fortinet

Malware

2 security vendors flagged the domain as malicious.

Subdomains (349)

leinerduran.ottimosoft1.com	0 / 94			
ottimosoft1.com	2 / 94	192.99.84.41	51.79.35.164	198.50.155.236
tiorico.ottimosoft1.com	0 / 94	51.79.35.164	148.72.0.122	
victordaniel.ottimosoft1.com	0 / 94	51.79.35.164	148.72.0.122	
luzstellavalencia.ottimosoft1.com	0 / 94	51.79.35.164		
erikatarapoto.ottimosoft1.com	0 / 94	148.72.0.122		
breineracosta.ottimosoft1.com	0 / 94	148.72.0.122		
diegovalencia.ottimosoft1.com	0 / 94	51.79.35.164	148.72.0.122	

Under the “Details” section of VirusTotal, scrolling down to Subdomains.

Over 349 subdomains were found under “ottomosoft1.com”, indicating the domain might be used for C2 communications.

Threat intel on IP “174.128.245.202”

Scan the listed IP address “[174.128.245.202](#)” on VirusTotal:

5
/ 94

Community Score

5/94 security vendors flagged this IP address as malicious

174.128.245.202 (174.128.224.0/19)
AS 46844 (SHARKTECH)

DETECTION

DETAILS

RELATIONS

COMMUNITY 2

5 security vendors flagged the IP address as Malicious.

5
/ 94

Community Score

5/94 security vendors flagged this IP address as malicious

174.128.245.202 (174.128.224.0/19)
AS 46844 (SHARKTECH)

DETECTION

DETAILS

RELATIONS

COMMUNITY 2

Comments (2)

goodbear

1 year ago

C2 Dridex - 174.128.245.202:443
By my bot @TrackerC2Bot

parthmaniar

3 years ago

This IP carried out Apache Log4j RCE attempt(s) (also known as CVE-2021-44228 or Log4Shell). For more info

2 comments were made from others users about this IP:

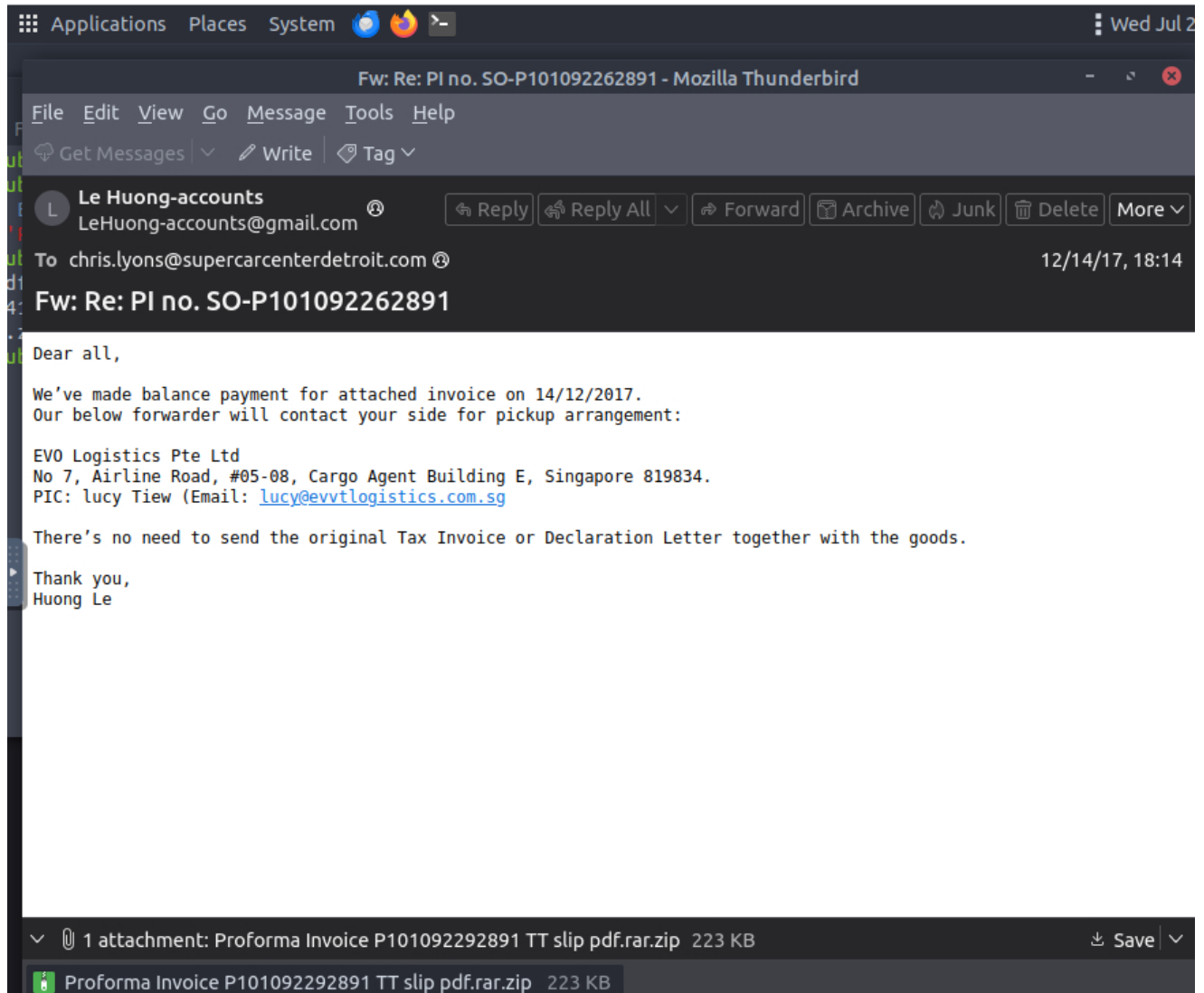
“C2 Dridex – 174.128.245.202:443”

“This IP carried out Apache Log4j RCE attempts...”

At this point it is safe to assume that this IP is likely a C2 server based on all findings.

File hash threat intelligence

Scenario: You are a SOC Analyst. Several suspicious emails have been forwarded to you from other coworkers. You must obtain details from each email to triage the incidents reported.



From: "LeHuong-accounts@gmail.com"

To: chris.lyons@supercarcenterdetroit.com

The email is a forwarded message confirming that payment for an attached invoice was made on **14/12/2017**. It provides contact details for the logistics company **EVO Logistics Pte Ltd** in Singapore, including the person in charge, **Lucy Tiew**, and her email address.

1 attached .zip file in mail for the supposed invoice.

Now get the file hash of the .zip file attached in the mail by using bash command sha256sum:

```
ubuntu@tryhackme:~/Desktop$ sha256sum Proforma\ Invoice\ P101092292891\ TT\ slip\ pdf.rar.zip
435bfc4c3a3c887fd39c058e8c11863d5dd1f05e0c7a86e232c93d0e979fdb28 Proforma Invoice P101092292891 TT
lip pdf.rar.zip
```

435bfc4c3a3c887fd39c058e8c11863d5dd1f05e0c7a86e232c93d0e979fdb28

53 / 67
Community Score -3

53/67 security vendors flagged this file as malicious

435bfc4c3a3c887fd39c058e8c11863d5dd1f05e0c7a86e232c93d0e979fdb28
Proforma Invoice P101092292891 TT slip pdf.rar.zip
Size: 223.10 KB
Last Analysis Date: 11 hours ago

zip attachment spreader long-sleeps malware direct-cpu-clock-access runtime-modules detect-debug-environment checks-user-input contains-pe

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 6

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.noon/fareitvb Threat categories trojan Family labels noon fareitvb hploki

Security vendors' analysis

Acronis (Static ML)	Suspicious	AhnLab-V3	Win-Trojan/VBKrypt.RP02.X1828
Alibaba	Trojan:Package/phishing.8	AliCloud	Trojan:Win/Injector.UUU
Antiy-AVL	HackTool[VirTool]/Win32.VBInject	Arcabit	Trojan.Generic.D232CB01
Avast	Win32:Evo-gen [Trj]	AVG	Win32:Evo-gen [Trj]

Virus total flags the file as Malicious as 53/67 security vendors states the file is either a Trojan, suspicious or malicious.

Bundle Info

Warnings

Contains one or more Windows executables.

Contents Metadata

Contained Files	1
Uncompressed Size	460.00 KB
Earliest Content Modification	2017-12-14 21:58:38
Latest Content Modification	2017-12-14 21:58:38

Contained Files By Type

PORTABLE EXECUTABLE	1
---------------------	---

Contained Files By Extension

EXE	1
-----	---

Bundled Files (1) ⓘ			
Scanned	Detections	File type	Name
<div> 2025-07-23 </div>	64 / 72	Win32 EXE	Proforma Invoice P101092292891 TT slip pdf.rar.exe

Looking into the "Details" section of VirusTotal hash page, it states the .zip file contains a Windows Executable file and the compressed file contains only 1 file in total.

It seems the supposed invoice file is masked as a ".pdf" in the title, despite being an .exe file.

DETECTION
DETAILS
RELATIONS
BEHAVIOR
COMMUNITY 6

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

☒ Display grouped sandbox reports

☒ CAPE Sandbox

⬆ 1

⚙ 6

🛡 1

🔍 3

🔗 2

📈 6

☒ VirusTotal Jujubox

⬆ 0

⚙ 0

🛡 0

🔍 0

🔗 0

📈 0

☒ Zenbox

⬆ 3

⚙ 7

🛡 0

🔍 1

🔗 3

📈 2

Activity Summary

Download Artifacts
Full Reports
Help

⬆ 3 Detections

2 MALWARE

1 TROJAN

1 EVADER

⚙ Mitre Signatures

2 HIGH

15 LOW

23 INFO

🛡 IDS Rules

1 HIGH

🔍 Sigma Rules

1 CRITICAL

1 HIGH

2 MEDIUM

🔗 Dropped Files

1 PE_EXE

1 OTHER

1 JAVASCRIPT

1 TEXT

📈 Network comms

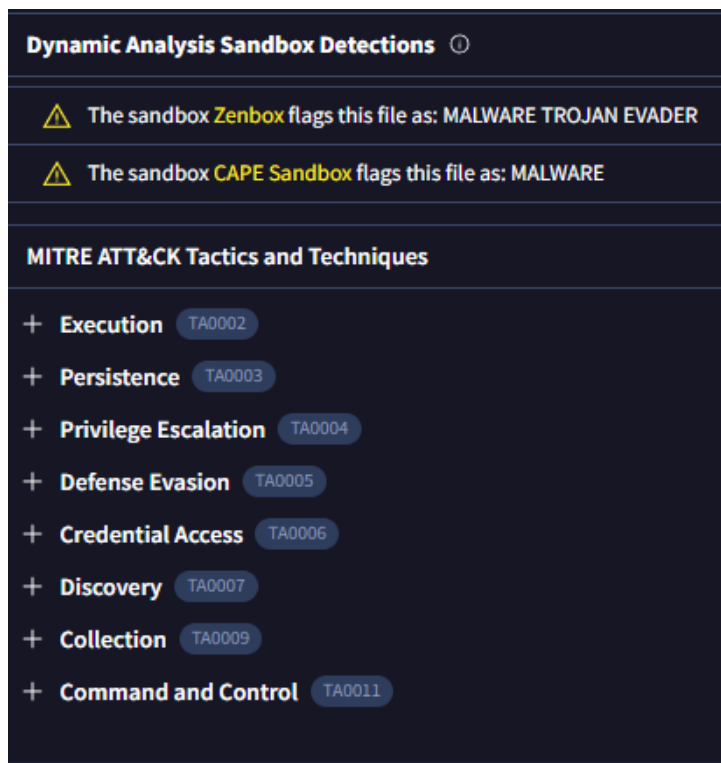
1 HTTP

6 DNS

1 IP

In the "Behaviour" section of Virustotal, it states sandbox services found various malicious artifacts to the file, including:


- Malware:** Indicates the presence of malicious software detected.
- Trojan:** Suggests a type of malware that disguises itself as legitimate software.
- Evasion:** Indicates attempts to avoid detection by security systems.
- Mitre Signatures (2 High, 15 Low, 23 Info):** Refers to identified tactics, techniques, and procedures from the MITRE ATT&CK framework.
- IDS Rules (1 High):** Intrusion Detection System rules triggered, with 1 high-severity alert.
- Sigma Rules (1 Critical, 1 High, 2 Medium)**
- Dropped Files (1 PE EXE, 1 JavaScript, 1 Text):** Files dropped during execution, including 1 Portable Executable (EXE), 1 JavaScript, and 1 text file.
- Network Communications (1 HTTP, 6 DNS, 1 IP):** Network activity detected, including 1 HTTP request, 6 DNS queries, and 1 IP communication.



File behaviour has been flagged as "MALWARE TROJAN EVADER", includes MITRE ATTACK Tactics such as Execution, Persistence, PE, Defense Evasion, Credential Access, Discovery and C2.

This means this software is designed to execute code, escalate privileges, establish persistence, evade defense restrictions, steal credentials, contact command and control server, as well as enumerate system information and internal network.

Next, use Any.run file upload analysis tool to scan the file for see a report on file behaviour:



ANALYZE MALWARE

> Huge database of s
> Unlimited submissi

ANY.RUN

INTERACTIVE MALWARE ANALYSIS

General

Behavior

MalConf

Static

General Info

File name:

Proforma Invoice P101092292891 TT slip pdf.rar.zip

Full analysis:

<https://app.any.run/tasks/9571fc47-9df3-46fa-bad2-9aedcb55d0d8>

Verdict:

Malicious activity

Threats:

Formbook Stealer Trojan

FormBook is a data stealer that is being distributed as a MaaS. FormBook differs threat actors to use FormBook virus.

Analysis date:

October 25, 2022 at 04:18:51


OS:

Windows 7 Professional Service Pack 1 (build: 7601, 32 bit)

Tags:

formbook trojan stealer

Indicators:




Any.run has flagged the file as “Malicious Activity” and states the file is a Trojan Infostealer malware and is part of “Formbook” data stealer.

“Formbook is a data stealer that is being distributed as a MaaS. Formbook differs from other MaaS products for its simplicity that allows unexperieced threat actors to use Formbook virus.”

Analysis date: October 25, 2022 at 04:18:51

OS: Windows 7 Professional Service Pack 1 (build: 7601, 32 bit)

Tags: formbook trojan stealer

Indicators: 

MIME: application/zip


File info: Zip archive data, at least v2.0 to extract

MD5: 4132A73C448CD2B5813DC2D34868ABA9

SHA1: 52C4CC2B87BCF41C6ACB800F9803BF3F26918614

SHA256: 435BFC4C3A3C887FD39C058E8C11863D5DD1F05E0C7A86E232C93D0E979FDB28

SSDEEP: 3072:3yQa5l5vqdNpiPa/fU7KYcjwxtLArEQuGV0B/xs6e77rFBDzxEx0vpfiTxdpbbo:3cl5EiPa/s7Kn8ILGtO3FB6z0vZil3bs

 ANY.RUN is an interactive service which provides full access to the guest system. Information in this report could be distorted by user ac
ANY.RUN does not guarantee maliciousness or safety of the content.

We can see that the file hash is identical to the hash we found from the file.

Behavior activities

MALICIOUS	SUSPICIOUS
<p>Application was dropped or rewritten from another process</p> <ul style="list-style-type: none">• Proforma Invoice P101092292891 TT slip pdf.rar.exe (PID: 3772)• Proforma Invoice P101092292891 TT slip pdf.rar.exe (PID: 1100)• Proforma Invoice P101092292891 TT slip pdf.rar.exe (PID: 3992)• Proforma Invoice P101092292891 TT slip pdf.rar.exe (PID: 2892) <p>FORMBOOK detected by memory dumps</p> <ul style="list-style-type: none">• wuauclt.exe (PID: 3916) <p>Changes the autorun value in the registry</p> <ul style="list-style-type: none">• wuauclt.exe (PID: 3916) <p><u>Formbook is detected</u></p> <ul style="list-style-type: none">• Firefox.exe (PID: 2580)• wuauclt.exe (PID: 3916)• Explorer.EXE (PID: 1488) <p><u>FORMBOOK was detected</u></p> <ul style="list-style-type: none">• Explorer.EXE (PID: 1488) <p><u>Connects to the CnC server</u></p> <ul style="list-style-type: none">• Explorer.EXE (PID: 1488)	<p>Application launched itself</p> <ul style="list-style-type: none">• Proforma Invoice P101092292891 TT slip pdf.rar.exe (PID: 1100)• Proforma Invoice P101092292891 TT slip pdf.rar.exe (PID: 3992) <p><u>Starts CMD.EXE for commands execution</u></p> <ul style="list-style-type: none">• wuauclt.exe (PID: 3916) <p>Loads DLL from Mozilla Firefox</p> <ul style="list-style-type: none">• wuauclt.exe (PID: 3916)

File behaviour analysis states that “Formbook” infostealer was detected, file starte cmd.exe for command executions and connect to Command and Control server.

File is highly malicious.

References and Tools

TryHackMe, (2025), “Threat hunting: Foothold”: tryhackme.com

Baker, Kurt, (2025), “Introduction to Cyber Threat Hunting”: crowdstrike.com

MITRE ATTACK, (2025): attack.mitre.org

Elastic, (2025), “The Elastic guide to threat hunting”: elastic.com

Elastic Stack SIEM (ELK): elastic.com

VirusTotal: virustotal.com

Cisco Talos Intelligence: talosintelligence.com

Urlscan.io: urlscan.io

AnyRun: any.run
