

Lab 10

Log Management and Compliance

Regularly record, track, monitor and test all access logs to network and workstations

Summary:

The use of logging mechanisms is critical in preventing, detecting and minimizing the impact of data compromise. If system usage is not logged, potential breaches cannot be identified. Secure, controlled audit trails must therefore be implemented that link all access to system components with individual users and log their actions. This includes access to cardholder data, actions taken by individuals with root or administrative privileges, access to audit trails, invalid logical access attempts, use of and changes to identification and authentication mechanisms, the initializing, stopping or pausing of audit logs, and the creation and deletion of system-level objects. An audit trail history should be retained for at least a year, with a minimum of three months' logs immediately available for analysis. Logs and security events should be regularly reviewed to identify anomalous or suspicious activity.

Contents:

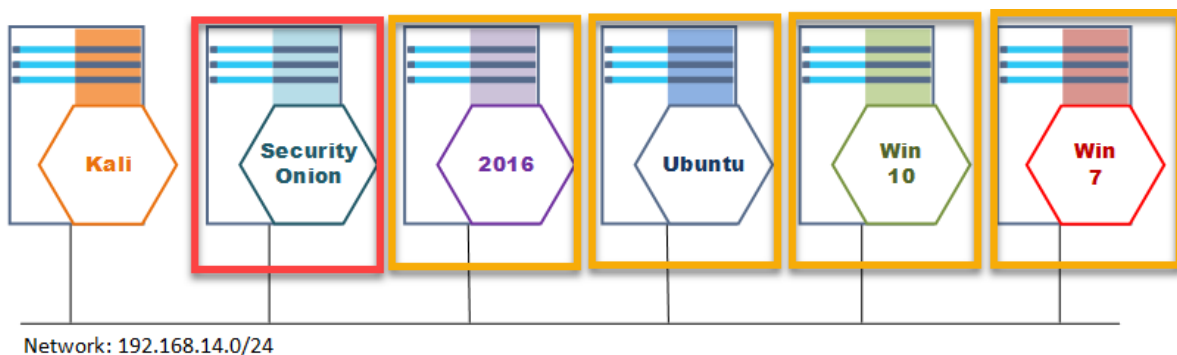
Summary:	1
Contents:	2
Lab Setup:	3
PCI_DSS Mapping:	3
Windows logon session process:	4
Windows Logs:	5
Common logon session events:	5
In Kibana:	6
Key Fields for Parsing Authentication Events:	6
Basic Search strings:	8
Policy option we need to confirm:	9
Audit HowTo:	10
Building filter (Type it in):	10
Building filter (Use the Wizard):	11
ToDo:	13
References:	13

Lab Setup:

Security onion is collecting log data from all of the systems on the network. In this exercise we will be filtering these logs to confirm compliance with user authentication and privileged access. We can build filters to search for privileged user logon times and locations to ensure actual user permissions reflect written policy.

PCI_DSS Mapping:

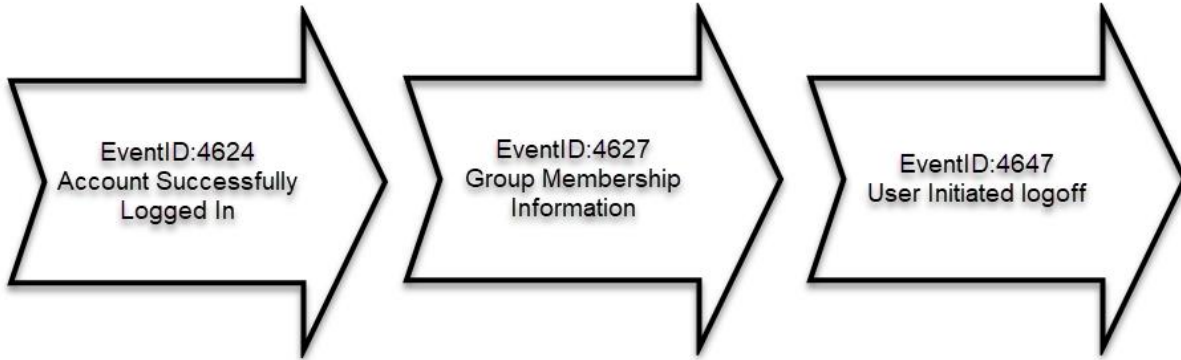
- 10.2.1 All individual user accesses to cardholder data.
- 10.2.2 All actions taken by any individual with root or administrative privileges.
- 10.2.3 Access to all audit trails.
- 10.2.4 Invalid logical access attempts.
- 10.2.5 Use of and changes to identification and authentication mechanisms—including but not limited to creation of new accounts and elevation of privileges—and all changes, additions, or deletions to accounts with root or administrative privileges.



Windows logon session process:

We need to understand the order that the windows EventLog uses to represent the logon process for an interactive user account. From that point we can figure out:

- Logon time
- Logout time
- Login locally or remotely
- Was this a standard user or admin



Questions we can answer from viewing the logs of the logon process:

Information		
Use this field to tie together all of the logon session events <code>data.EventChannel.EventData.TargetLogonId</code>		
Who Logged In <code>data.EventChannel.EventData.TargetUserName</code>	All groups User Is In <code>data.EventChannel.EventData.GroupMembership</code>	
What Did They Log Into <code>data.EventChannel.EventData.WorkstationName</code>		
When Did they Login <code>data.EventChannel.System.SystemTime</code>		When Did they Logout <code>data.EventChannel.System.SystemTime</code>
Where Did They Login From <code>data.EventChannel.EventData.IpAddress</code>		
How Did They Logon <code>data.EventChannel.EventData.LogonType</code>		
Is Account Local Or Domain <code>data.EventChannel.EventData.TargetDomainName</code>		
Is the User an Admin <code>data.EventChannel.EventData.ElevatedToken</code>		

Windows Logs:

There are many fields contained in the Windows Event logs. This information can be used to confirm that a logon policy is being properly enforced or later to correlate a users logon session with other activity on a system.

Common logon session events:

event_id	data.EventChannel.System.Message
4624	<u>An account was successfully logged on</u> (successful attempt to logon to the local computer regardless of logon type, location of the user or type of account) Note: Use "TargetLogonId" to tie this to the rest of the session
4625	<u>This identifies the user that attempted to logon and failed.</u>
4627	<u>Group membership information</u> (One or more of these events are logged whenever a user logs on or a logon session begins for any other reason) Note: Use "TargetLogonId" to tie this to the rest of the session
4634	<u>An account was logged off</u>
4647	<u>User initiated logoff</u> Used during interactive logons when the user logs out Note: Use "TargetLogonId" to tie this to the rest of the session
4672	<u>Special privileges assigned to new logon</u> Whenever an account assigned any "administrator equivalent" user rights logs on. (This will be close to 4624 when an admin logs in).

In Kibana:

Key Fields for Parsing Authentication Events:

Field	Description
Metadata about the Event	
data.EventChannel.System.ProviderName	The Windows Event Viewer log source [Microsoft-Windows-Security-Auditing]
event_id	The Windows Event ID
data.EventChannel.System.SystemTime	The Time an Event was logged <u>to the windows machine</u> (Remember all times on Sec Onion are UTC) vs likely timezone setting on Windows machine.
New Logon Info: The user who just logged on	
data.EventChannel.EventData.TargetUserName	User Account Name. Note: Can be a logon account or a system account.
data.EventChannel.EventData.TargetDomainName	Domain name of the account in either the DNS name (can be upper or lowercase) or pre-Win2k NETBIOS domain name. <u>Local User Account</u> = Name of the Computer <u>Domain User Account</u> = Name of the Domain <u>Security principal accounts</u> <ul style="list-style-type: none">● SYSTEM = "NT AUTHORITY"● LOCAL SERVICE = "NT AUTHORITY"● NETWORK SERVICE = "NT AUTHORITY"● ANONYMOUS LOGON = "NT AUTHORITY" <u>Virtual Accounts</u> = "NT Service"
data.EventChannel.EventData.TargetUserSid	SID of the account
data.EventChannel.EventData.LogonType	<ul style="list-style-type: none">● 2 = Interactive (logon at keyboard and screen of system)● 3 = Network (Mapped Drive/failed RDP)● 10 = Remote Interactive (Terminal Services, Remote Desktop or Remote Assistance) <p>There are more login types but for the scope of this lab these will do.</p>
data.EventChannel.EventData.ElevatedToken	Elevated Token: <ul style="list-style-type: none">● Yes = %%1842

	<ul style="list-style-type: none"> • No = %%1843 <p>It will be Yes if the user is a member of the Administrators group</p> <p>With interactive logons, when you are an admin and you have User Account Control (UAC) enabled. Then when you logon you actually get 2 logon sessions. One without the Administrators SID and related privileges in your security token and another session with all that authority. Everything you do happens under the unprivileged logon session until you attempt to run something requiring admin authority. After you approve the UAC dialog box, Windows runs that one operation under the other logon session. So <u>in the log you will see 2 of these events, one where this field is Yes and other No.</u> The 2 logon sessions are connected by the Linked Logon ID described below. Note: These are 2 different logon sessions so the TargetLogonId” <u>will not</u> associate the 2 events.</p>
data.EventChannel.EventData.TargetLogonId	A semi-unique (unique between reboots) number that identifies the logon session just initiated. Any events logged subsequently during this logon session will report the same Logon ID through to the logoff event 4647 or 4634.
Network Information: If the user logged in remotely	
data.EventChannel.EventData.IpAddress	<p>The IP address of the computer where the user is physically sitting (unless this logon was initiated by a server application acting on behalf of the user).</p> <p>If logon is initiated on local machine</p> <ul style="list-style-type: none"> • IP = 127.0.0.1 • IP = actual local IP address.

Basic Search strings:

To find a specific event_id:

```
>_ event_id:<id number>
```

To find all Events for a given user:

```
>_ data.EventChannel.EventData.TargetUserName:"Administrator"
```

We can also chain multiple queries together by using "AND"

To find all successful local logon attempts:

```
>_ data.EventChannel.EventData.LogonType:2 AND event_id:4624
```

To find all failed local logon attempts:

```
>_ data.EventChannel.EventData.LogonType:2 AND event_id:4625
```

To find all successful Remote Desktop logon attempts:

```
>_ data.EventChannel.EventData.LogonType:10 AND event_id:4624
```

To find all failed Remote Desktop logon attempts (Note:Type 3) :

```
>_ data.EventChannel.EventData.LogonType:3 AND event_id:4624
```

To filter out the "SYSTEM" account from your results:

```
>_ event_id:4624 AND NOT data.EventChannel.EventData.TargetUserName:"SYSTEM"
```

To find all successful logon attempts with Elevated Privileges:
(Note:this will show ALL accounts including "SYSTEM" type accounts)

```
>_ data.EventChannel.EventData.ElevatedToken:%%1842
```


Policy option we need to confirm:

Although we can extend this exercise, our goal here is to confirm compliance with actions taken by individuals with root or administrative privileges, access to audit trails, invalid logical access attempts. We want to make sure that there are no accounts logging in with privileges not defined in our policy. As such we will be checking for

- Successful logins
- Failed logins
- Logins with admin privileges

Test to ensure the policy is being followed.

1. Only the Administrators account should be able to logon locally to the Win2016 server.
2. Too many failed logon attempts should be looked at as potential attack.
3. Only the Administrators account should have administrative privileges

Audit HowTo:

Building filter (Type it in):

So to find an audit trail for any users who logged in locally with elevated (admin) privileges.

To find all successful local logon attempts with Elevated (Admin) Privileges:

```
>_ data.EventChannel.EventData.LogonType:2 AND  
data.EventChannel.EventData.ElevatedToken:%%1842
```

Once we have found the logon success event we are looking for we can find the rest of the logon/logoff process using the "TargetLogonId". Once you set the TargetLogonId you will need to remove all of the other filters so you can see all events associated with the logon session.

Note: Sometimes you can not find the logoff event if the system was shutdown while the user was logged in (often the case with educational VM's but not so much on a real production network).

Note: Your ID will vary.

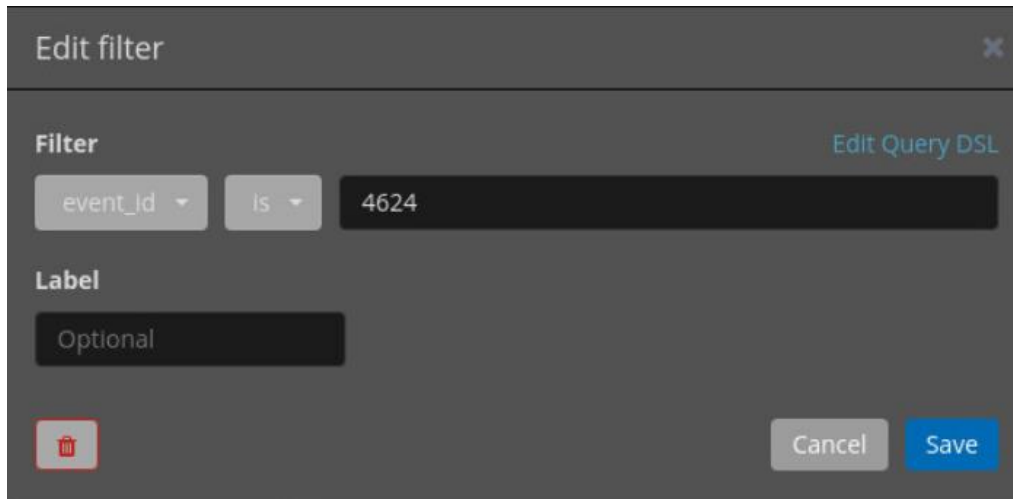
```
>_ data.EventChannel.EventData.TargetLogonId:0x34d1c3
```

This will get us:

- event_id:4624
 - data.EventChannel.System.SystemTime = Logon Time
- event_id:4647
 - data.EventChannel.System.SystemTime = Logout Time

Building filter (Use the Wizard):

Step 1) Create the EventID filter: This will find all successful logon attempts.



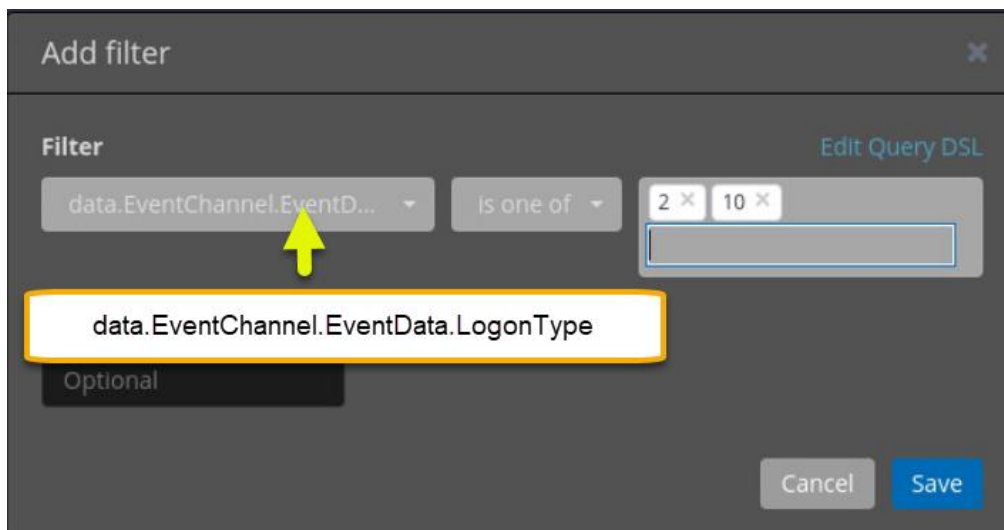
The screenshot shows the 'Edit filter' dialog box. The filter is configured as follows:

- Filter: event_Id is 4624
- Label: Optional

Buttons: Cancel, Save

Step 2) Look for LogonType

- 2 = Local logon
- 10 = Remote Desktop



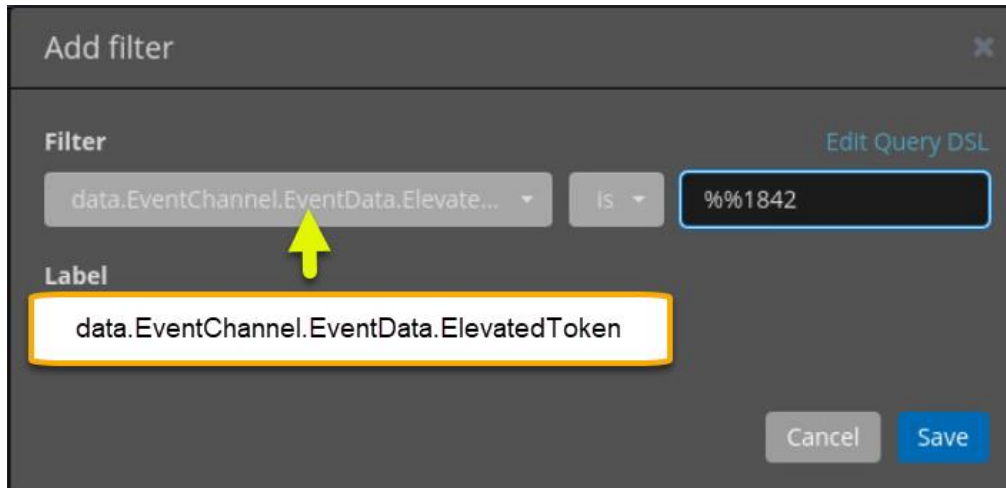
The screenshot shows the 'Add filter' dialog box. The filter is configured as follows:

- Filter: data.EventChannel.EventD... is one of 2, 10
- Label: Optional

Buttons: Cancel, Save

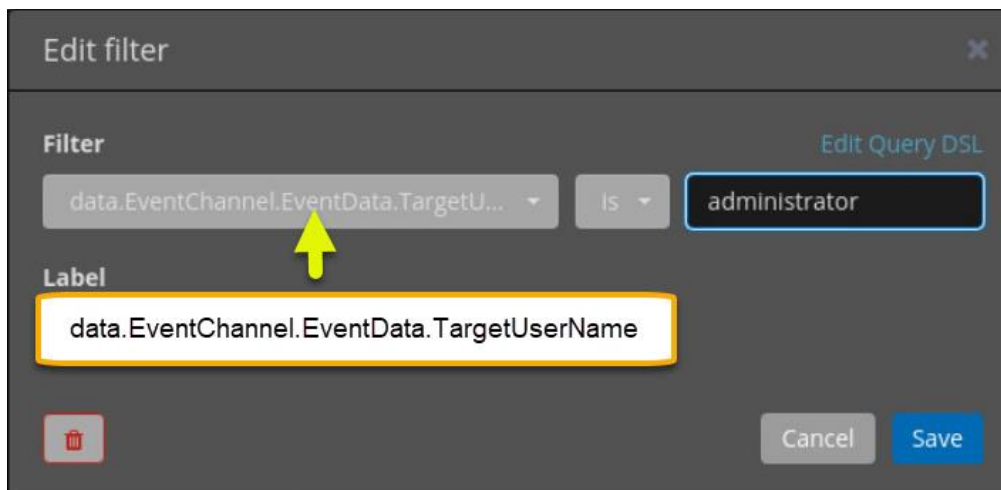
Step 3) Look for any accounts that were given admin privileges.

- %%1842 = Member of the administrators group
- %%1843 = Not a member of the administrators group



The screenshot shows a dialog box titled "Add filter" with a close button (X) in the top right corner. Below the title bar, there is a section labeled "Filter" with a link "Edit Query DSL" to its right. The "Filter" section contains a dropdown menu with the text "data.EventChannel.EventData.Elevate...", a "is" dropdown, and a text input field containing "%%1842". Below this is a section labeled "Label" with a text input field containing "data.EventChannel.EventData.ElevatedToken". At the bottom right, there are "Cancel" and "Save" buttons. A yellow arrow points to the "Filter" dropdown menu.

Step 4) Search for specific account name:



The screenshot shows a dialog box titled "Edit filter" with a close button (X) in the top right corner. Below the title bar, there is a section labeled "Filter" with a link "Edit Query DSL" to its right. The "Filter" section contains a dropdown menu with the text "data.EventChannel.EventData.TargetU...", a "is" dropdown, and a text input field containing "administrator". Below this is a section labeled "Label" with a text input field containing "data.EventChannel.EventData.TargetUserName". At the bottom left, there is a trash icon. At the bottom right, there are "Cancel" and "Save" buttons. A yellow arrow points to the "Filter" dropdown menu.

ToDo:

1. Find how many times the "Administrator" account successfully log on.

2. Find how many times the "Administrator" account unsuccessfully log on.
3. From looking at the records, what is the only machine that has ever had the "Administrator" account log into it? _____
4. Have any machines been logged into using "Remote Desktop": _____
5. Which ones: _____
6. What user/users logged into it: _____
7. How long was the user logged in: _____
Note: This one can be hard or impossible depending on how the user exited the RDP session

References:

<https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventID=4624>