Lab 7

Implement System Access Control and Intrusion Detection

Implement and test access control systems to cardholder and sensitive data by business need to know

Summary:

Exploiting authorized accounts and abusing user privileges is one of the easiest ways for criminal hackers to gain access to a system. It is also one of the most difficult types of attack to detect. Documented systems and processes should therefore be put in place to limit access rights to critical data. Access control systems should deny all access by default, and access should be granted on a need-to-know basis and according to the clearly defined job responsibilities of authorized personnel. 'Need to know' is defined in the PCI DSS as "when access rights are granted to only the least amount [sic] of data and privileges needed to perform a job".

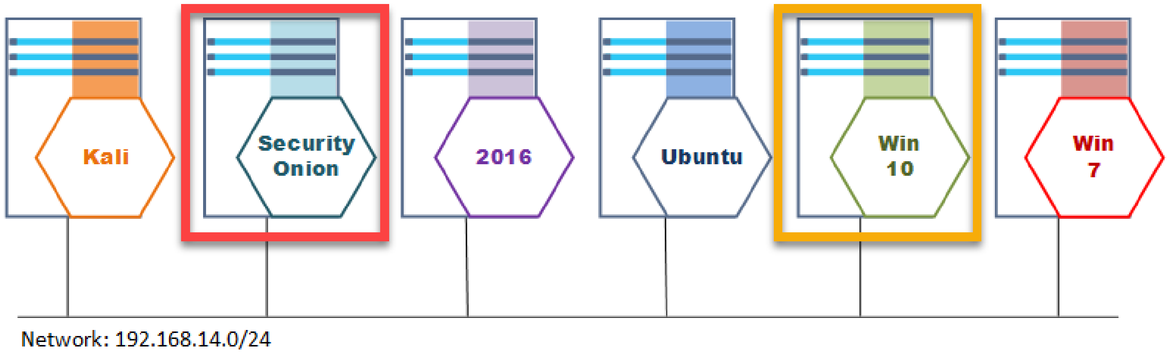
Contents:

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 standard windows updates. We can build filters to search for the installation of specific high risk updates or search for a more general “Best Practices” policy of installing updates in a timely manner. For this particular lab we are looking specifically for Windows Update compliance however with some tuning almost any log file can be imported and parsed by Kibana. This lab is a more general version of Lab 5. In Lab 5 we were looking specifically for Windows Defender Definition updates whereas we are now looking for all required updates. The same techniques we used in Lab 5 are also useful here.

PCI_DSS Mapping:

7.3 Ensure that security policies and operational procedures for restricting access to cardholder data are documented, in use, and known to all affected parties.



Windows Logs:

Common update events:

There are a few Event IDs that are related to object access. For our purposes we really only need to worry about event_id:4663.

[event_id:]

event_id	**data.EventChannel.System.Message**
4656	A handle to an object was requested (open) Can we see a fail here
4658	The handle to an object was closed (close)
4660	An object was deleted
4663	An attempt was made to access an object

In Kibana:

Key Fields for Parsing Updates:

Field	Description
event_id	The Windows Event ID
data.EventChannel.EventData.ObjectName	This will contain the name of the object you are auditing (folder name in our case)
data.EventChannel.EventData.SubjectUserName	This is the username of the account that accessed the resource.
data.EventChannel.EventData.AccessList	The access permissions that the user has to the object. (Note: not used for this lab but good info if different users have specific permissions in the policy)
data.EventChannel.System.SystemTime	The Time an Event was logged to the windows machine (Remember all times on Sec Onion are UTC) vs likely timezone setting on Windows machine. To set timezone: Management -> Advanced Settings.
data.EventChannel.System.Computer	The name of the computer the event was logged on.
agent.ip	IP address of the Wazuh agent generating the log entry.

Basic Search strings:

To find a specific event_id:

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

To search for a specific Object Name

```
>_ data.EventChannel.EventData.ObjectName
```

To remove a specific user from your search

```
>_ NOT data.EventChannel.EventData.SubjectUserName
```

Policy option we need to confirm:

Access control systems should deny all access by default, and access should be granted on a need-to-know basis and according to the clearly defined job responsibilities of authorized personnel.

To fall within this policy guideline only the “Administrator” account has job responsibilities requiring access to the directories that the critical company data is stored in.

- C:\Users\Public\Documents\CustomerInformation
- C:\Users\Public\Documents\TradeSecrets

Test to ensure the policy is being followed.

Audit HowTo:

Building filter (Type it in):

Find all successful interactions with your protected file or directory

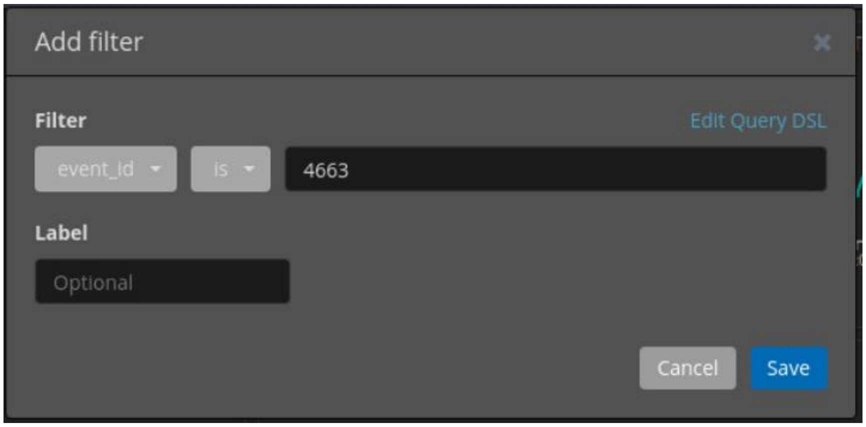
```
>_ event_id:4663 AND data.EventChannel.EventData.ObjectName=”file or directory name”
```

To remove the accounts that are supposed to have access and leave just the accounts that have accessed the file/directory that do not have permission

```
>_ event_id:4663 AND data.EventChannel.EventData.ObjectName=”file or directory name”
AND NOT data.EventChannel.EventData.SubjectUserName=”allowed username”
```

Building filter (Use the Wizard):

Step 1) Create the EventID filter:

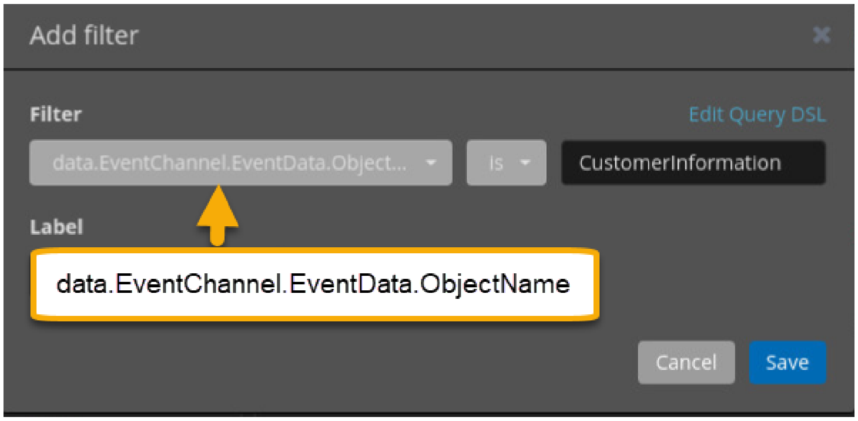


Add filter

Filter: event_id is 4663

Label: Optional

Step2) Create the ObjectName filter:



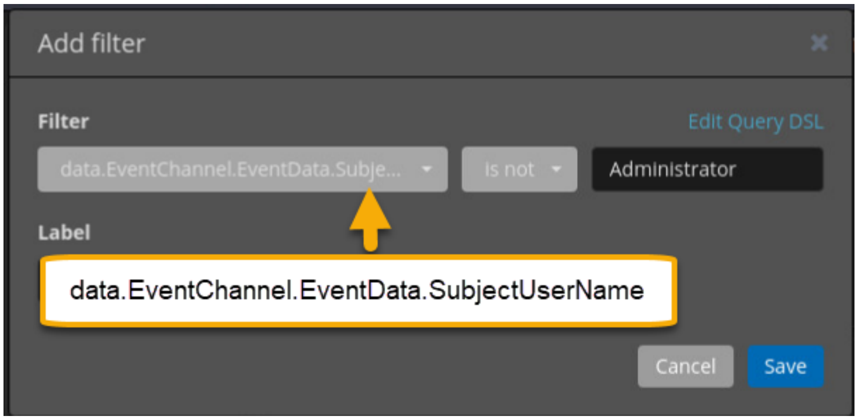
Add filter

Filter: data.EventChannel.EventData.Object... is CustomerInformation

Label

data.EventChannel.EventData.ObjectName

Step 3) Create the SubjectUsername Filter:



Add filter

Filter: data.EventChannel.EventData.Subje... is not Administrator

Label

data.EventChannel.EventData.SubjectUserName

ToDo:

1. Are any users that are not allowed in the policy able to access the directory in question? ____

2. If so what is the user account that is against policy? ____________________________

3. How would you recommend bringing the directory into compliance? ____