Usecase 1- Honeytoken Access

Objective: Detect unauthorised access to a decoy file (honeytoken) to signify breach or misuse. Any access attempt is immediately suspicious as these files are not meant to be accessed

Log Source: Windows Security Event Logs

Event ID: 4663- “An Attempt was made to access an object”

Example Target Log From Event Viewer:

An attempt was made to access an object.

Subject:
- Security ID: <system name>\<username>
- Account Name: <username>
- Account Domain: <system name>
- Logon ID: 0x19AC85

Object:
- Object Server: Security
- Object Type: File
- Object Name: C:\Users\<username>\OneDrive\Desktop\Passwords\Passwords.docx
- Handle ID: 0x2194
- Resource Attributes: S:AI

Process Information:
- Process ID: 0xa68
- Process Name: C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE

Access Request Information:
- Accesses: ReadAttributes
- Access Mask: 0x80

Pre-configuration: Enabling file auditing in PowerShell via the command ‘auditpol /set /category:”Object Access” /(here I used success and failure respectively):enable’, then enabling file auditing on the passwords folder by navigating to the auditing tab via properties>security>advanced>auditing and creating a rule for auditing for any file access by anyone within the folder, also remove ID 4663 default exclusion from the Wazuh agent configuration security event log collection.

Custom Wazuh Rule:

(username is a placeholder within the filepath for the purposes of privacy)

```
<group name="windows, windows_security,json,">
    <rule id="100111" level="9">
    <if_sid>60103</if_sid>
    <field name="win.system.eventID">^4663$</field>
    <field name="win.eventdata.objectName">^C:\\\\Users\\\\username\\\\OneDrive\\\\Desktop\\\\Passwords\\\\Passwords\.docx$</field>
    <description>User attempted to access HoneyToken</description>
    </rule>
</group>
```

If_sid 60103 matches all audit success events

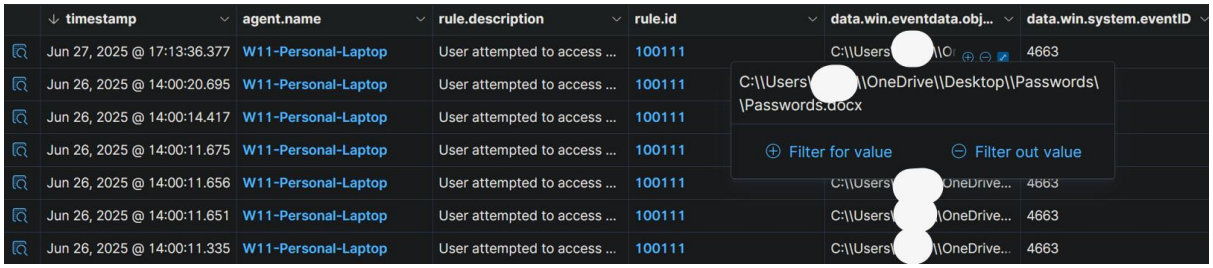
```
<rule id="60103" level="0">
  <if_sid>60001</if_sid>
  <field name="win.system.severityValue">^AUDIT_SUCCESS$|^success$</field>
  <description>Windows audit success event</description>
  <options>no_full_log</options>
</rule>
```

Field name= win.eventdata.eventID is ^4663$ filters these audit success events for successful object access events

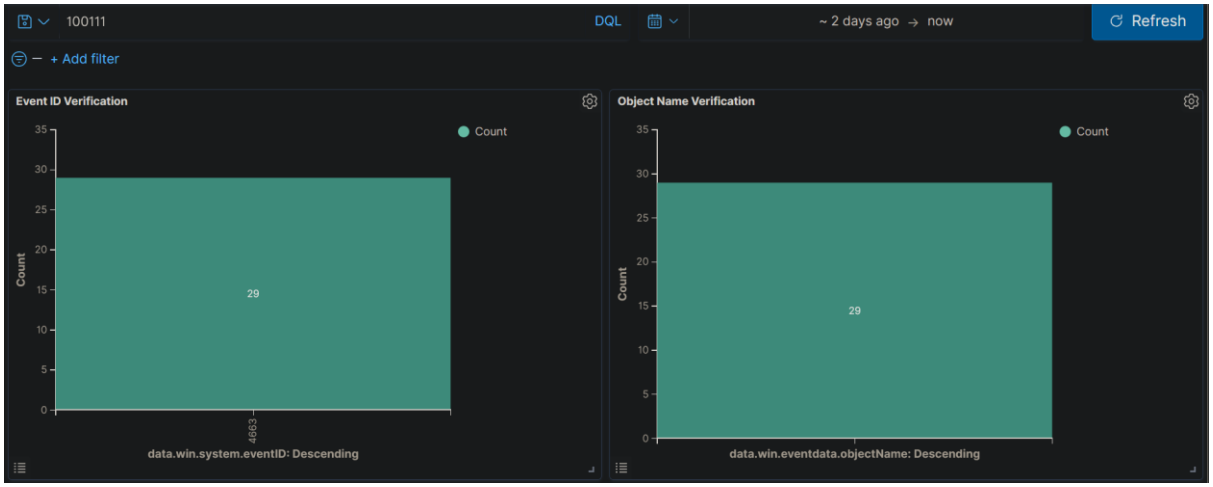
Field name= win.eventdata.eventID is ^C:\\\\Users\\\\username\\\\OneDrive\\\\Desktop\\\\Passwords\\\\Passwords\.docx$ filters for only the specified file path. Wazuh records Windows filepaths with 2 backslashes and XML considers backslash its escape character, therefore each separator requires 4 backslashes. Fullstop is the XML wildcard character, therefore a backslash is used before .docx to ensure an exact match

^ signifies the start of the match and $ signifies the end for an exact match in both fields.

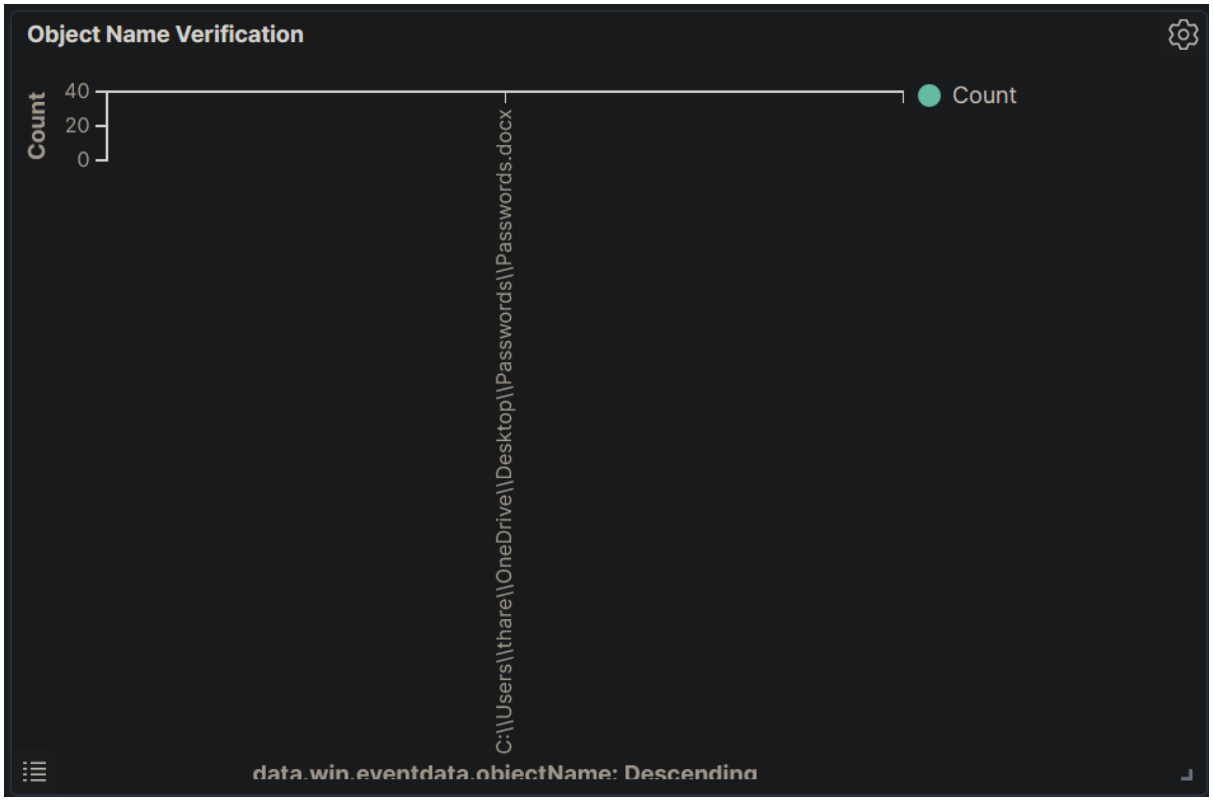
Example Alerts:



Verification of fields:



We can see there is a single term for both the Event ID and Object Name fields for the 100111 rule which is the expected behaviour, label is disabled for Object name as it makes the graph unreadable


Object Name Verification
Count
40
20
0
Count
C:\\Users\\thare\\OneDrive\\Desktop\\Passwords\\Passwords.docx
data.win.eventdata.objectName: Descending