# دليل إثبات - Wazuh (Windows)

## إصدار الدليل

هذا الدليل متوافق مع Wazuh 4.2.0 والإصدارات الأحدث. [انقر هنا للحصول على الإصدارات الأقدم.](https://github.com/wazuh/wazuh/wiki/Proof-of-concept-guide/b3681f865eb671a0b0d13a1b1012477fd0d932ba)

## الفهرس

* [إصدار الدليل](#version)
* [اكتشاف هجوم القوة الغاشمة - RDP](#brute_force)
* [مراقبة سلامة الملفات](#fim)
* [اكتشاف وإزالة البرامج الضارة - تكامل VirusTotal](#virustotal)
* [كشف الثغرات الأمنية](#vulnerability_detector)
* [تكامل Slack](#slack)

## مقدمة

يشرح المستند التالي كيفية إعداد بيئة Wazuh لاختبار قدرات المنتج المختلفة على نقاط النهاية التي تعمل بنظام Windows. يفترض أن المكونات التالية مثبتة بالفعل:

* Elasticsearch + Kibana + إضافة Wazuh Kibana
* مدير Wazuh + Filebeat (للتكامل مع Elasticsearch)
* عميل Wazuh (Windows)

يمكن العثور على دليل جيد حول كيفية تثبيت هذه المكونات في (Wazuh Server & Agent Installation Documentation.pdf)

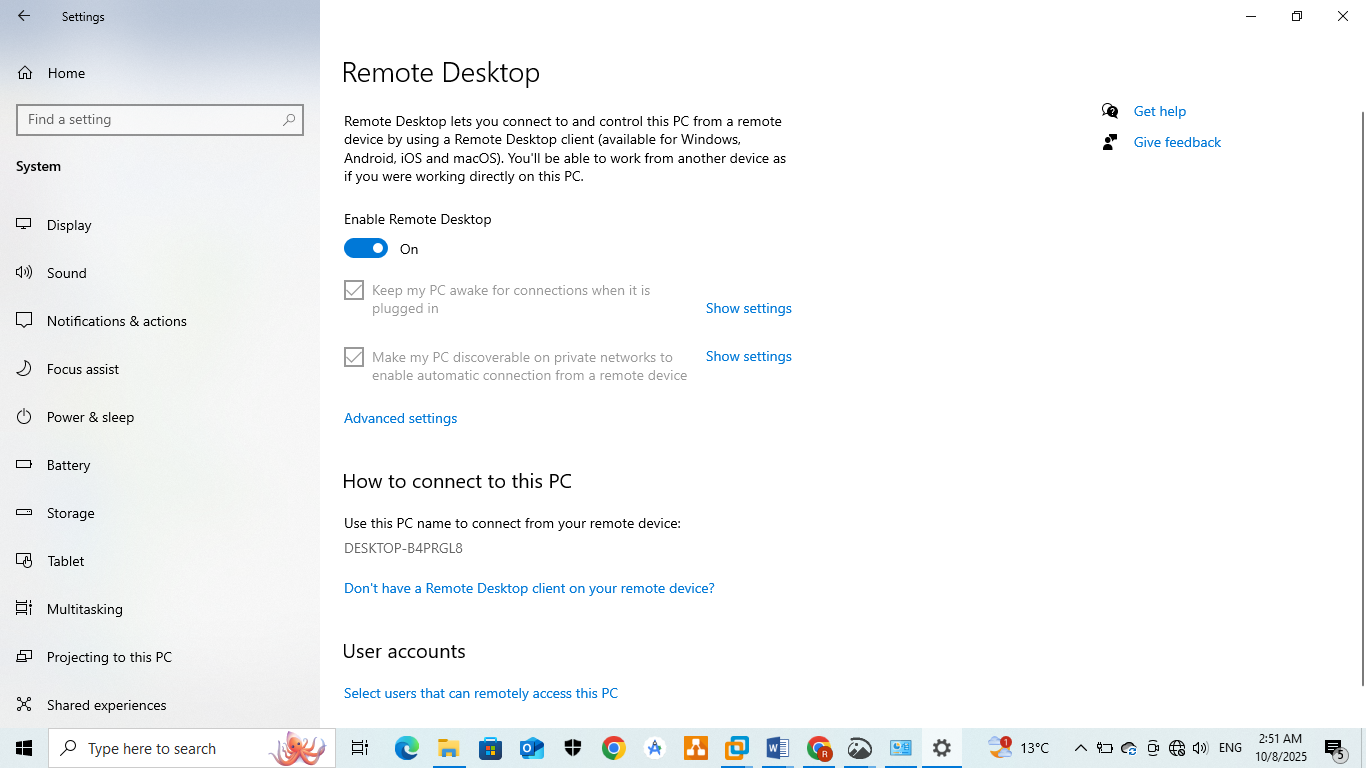
تشرح الأقسام أدناه التكوينات المطلوبة لإعداد حالات الاستخدام المختلفة على نقاط نهاية Windows.

## اكتشاف هجوم القوة الغاشمة - RDP

هجمات القوة الغاشمة على RDP (بروتوكول سطح المكتب البعيد) هي من متجهات الهجوم الشائعة على أنظمة Windows. يوفر Wazuh قواعد جاهزة قادرة على تحديد هجمات القوة الغاشمة، من خلال ربط أحداث فشل المصادقة المتعددة.

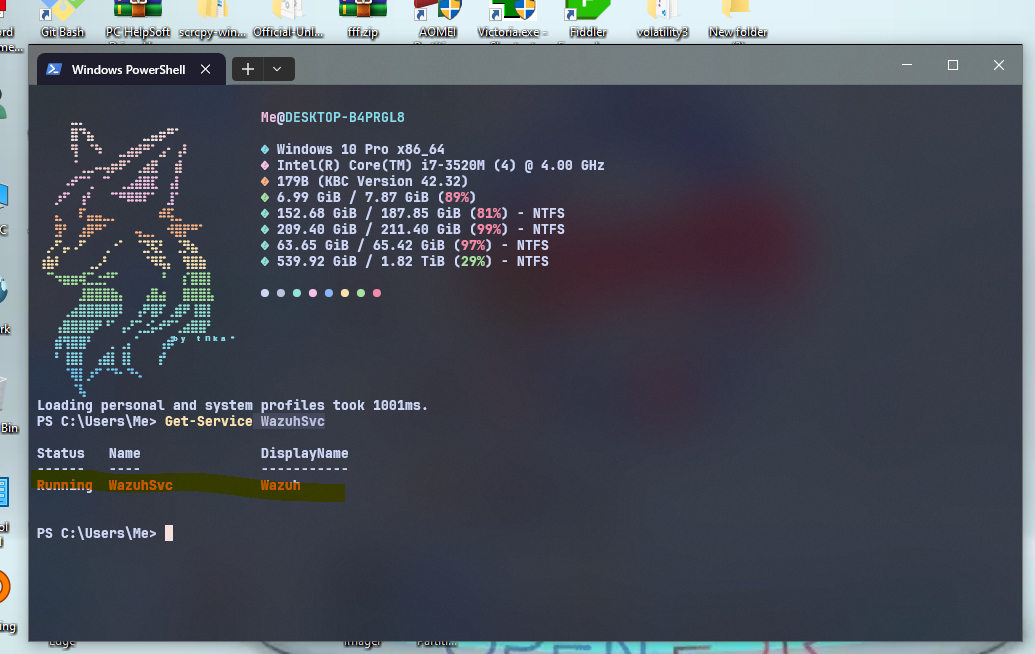
#### التكوين

* تأكد من تمكين خدمة Remote Desktop على جهاز Windows المراقب.

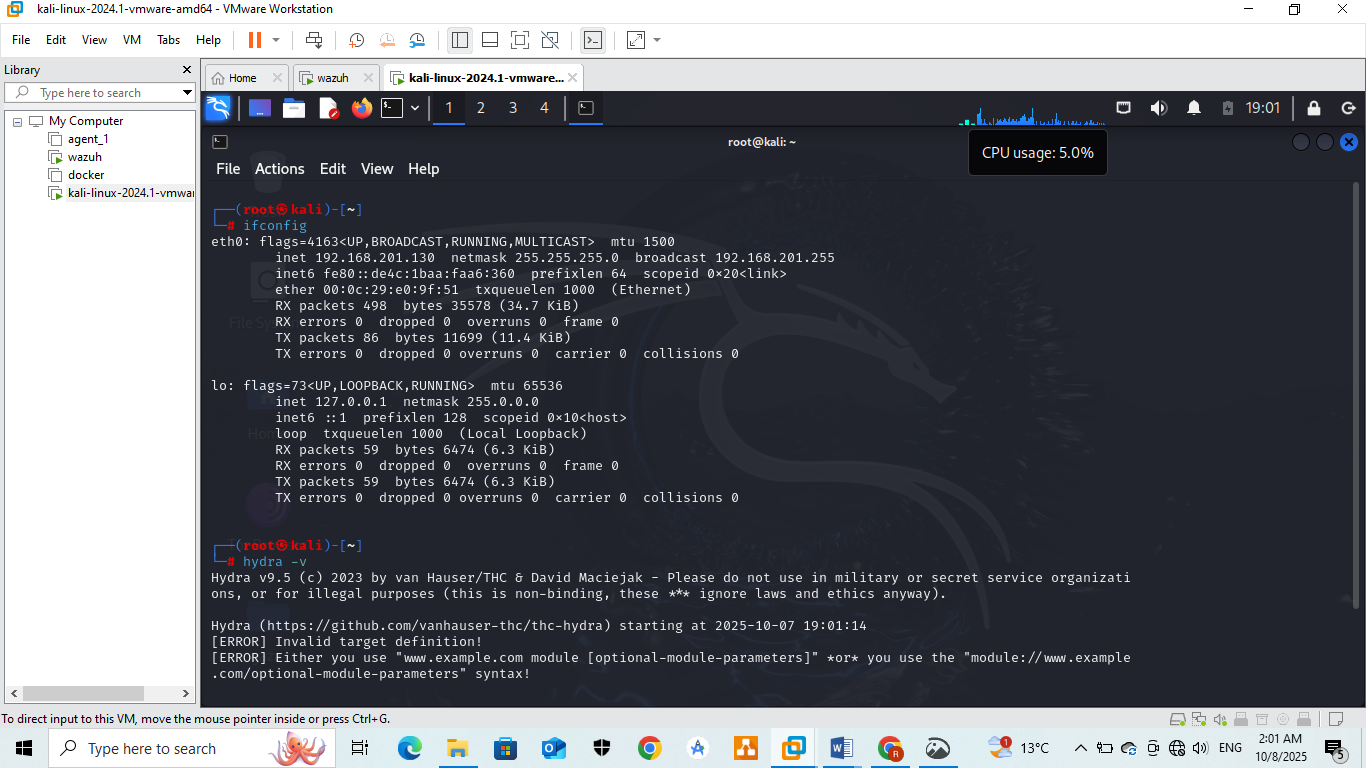


* تأكد من أن عميل Wazuh مثبت ويعمل على جهاز Windows.

Get-Service WazuhSvc



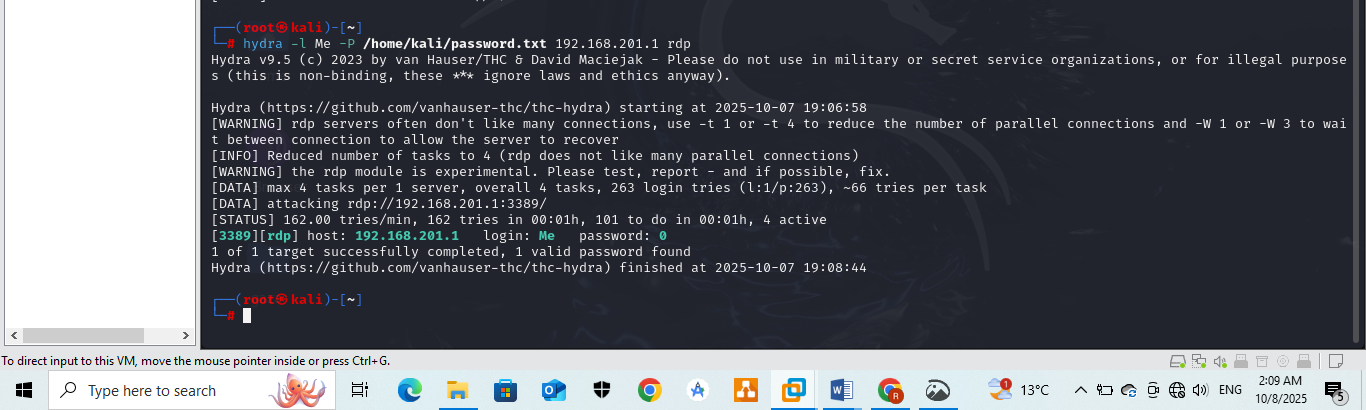
* لتنفيذ هجوم آلي من جهاز آخر، يمكنك استخدام أداة hydra:



# من جهاز Linux  
**hydra -l Target\_user -P password.txt** <windows-ip-address> **rdp**

#### خطوات توليد التنبيهات

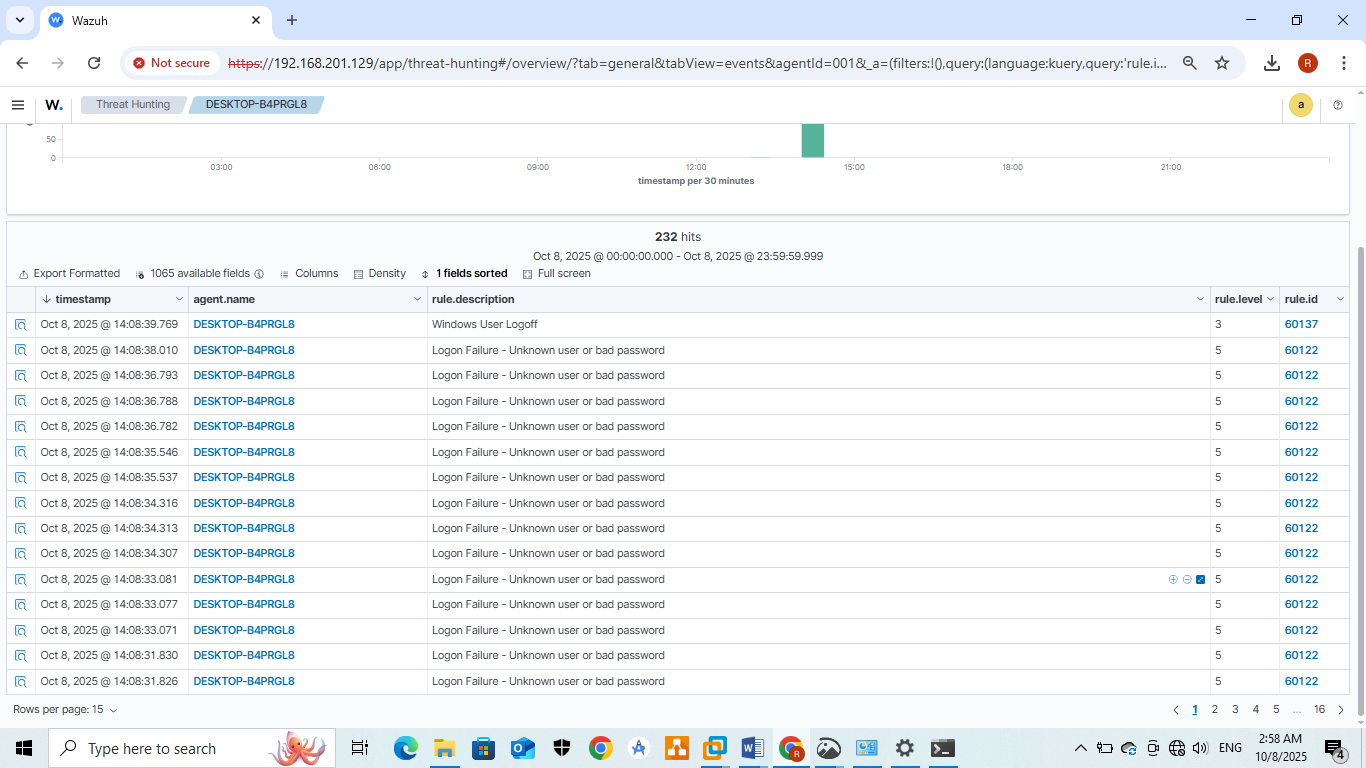
**hydra -l Me -P /home/kali/password.txt** 192.168.201.1 **rdp**

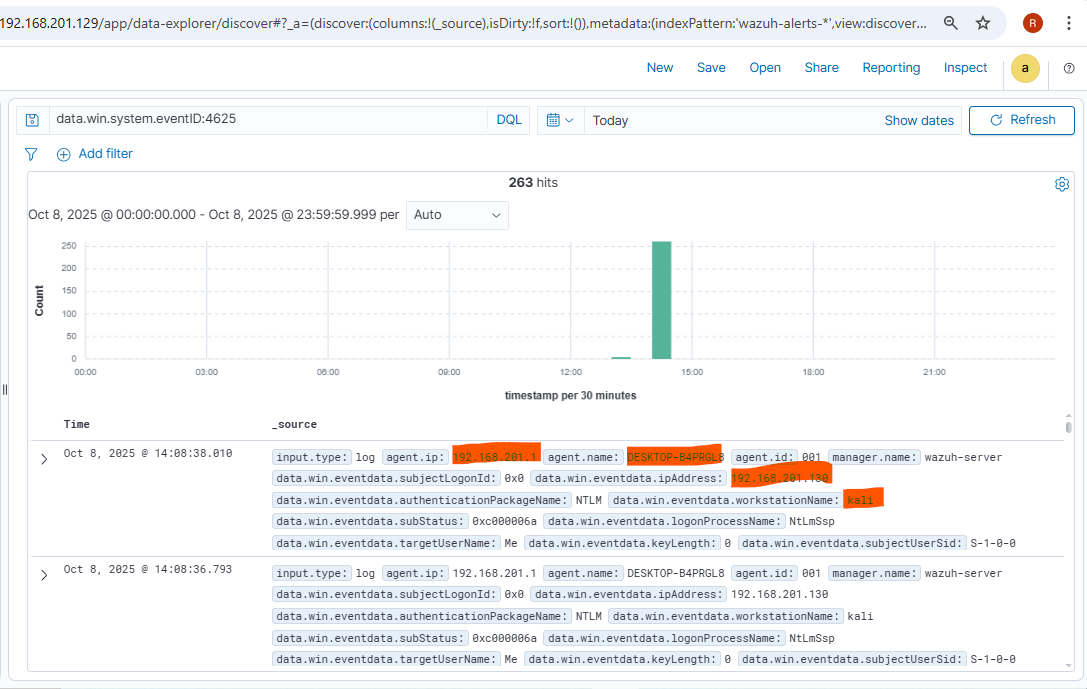


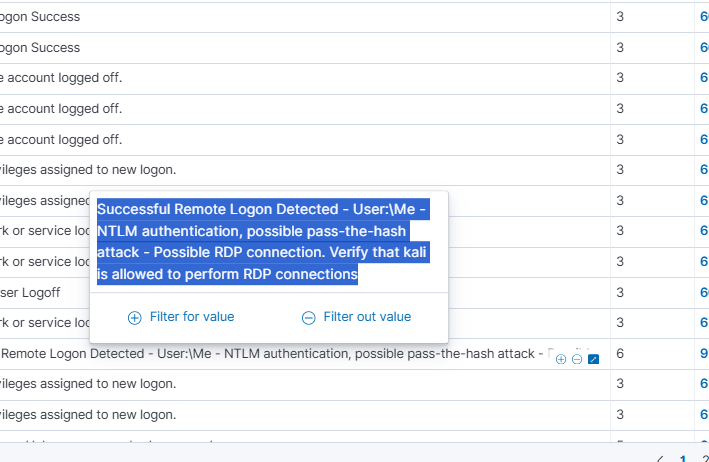
#### التنبيهات

يمكن العثور على التنبيهات ذات الصلة في Kibana باستخدام:

* rule.id:(60122 OR 60137)
* rule.description:\*brute\*force\*
* data.win.system.eventID:4625 (فشل تسجيل الدخول)



****



#### نقطة النهاية المتأثرة

* Windows

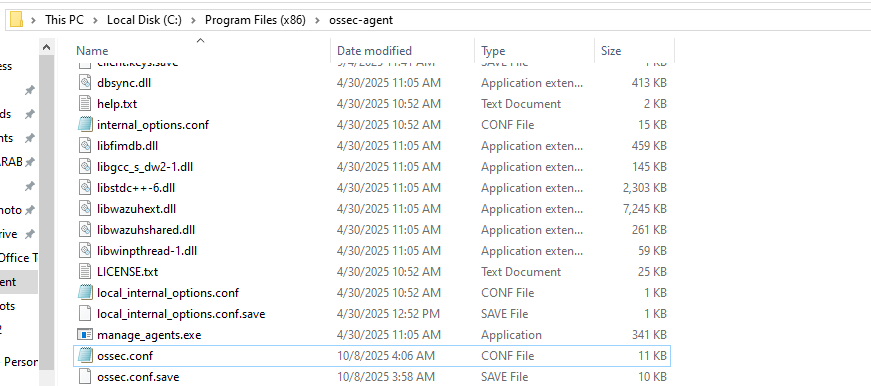
## مراقبة سلامة الملفات

مراقبة سلامة الملفات (FIM) هي ميزة أمنية حيوية تراقب التغييرات على الملفات والأدلة المهمة. يمكن لـ Wazuh اكتشاف إنشاء، تعديل، أو حذف الملفات في الوقت الفعلي.

#### التكوين

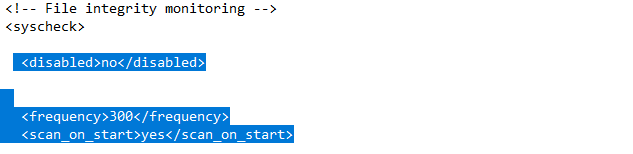
على جهاز Windows المراقب، قم بتحرير ملف تكوين عميل Wazuh الموجود في: C:\Program Files (x86)\ossec-agent\ossec.conf

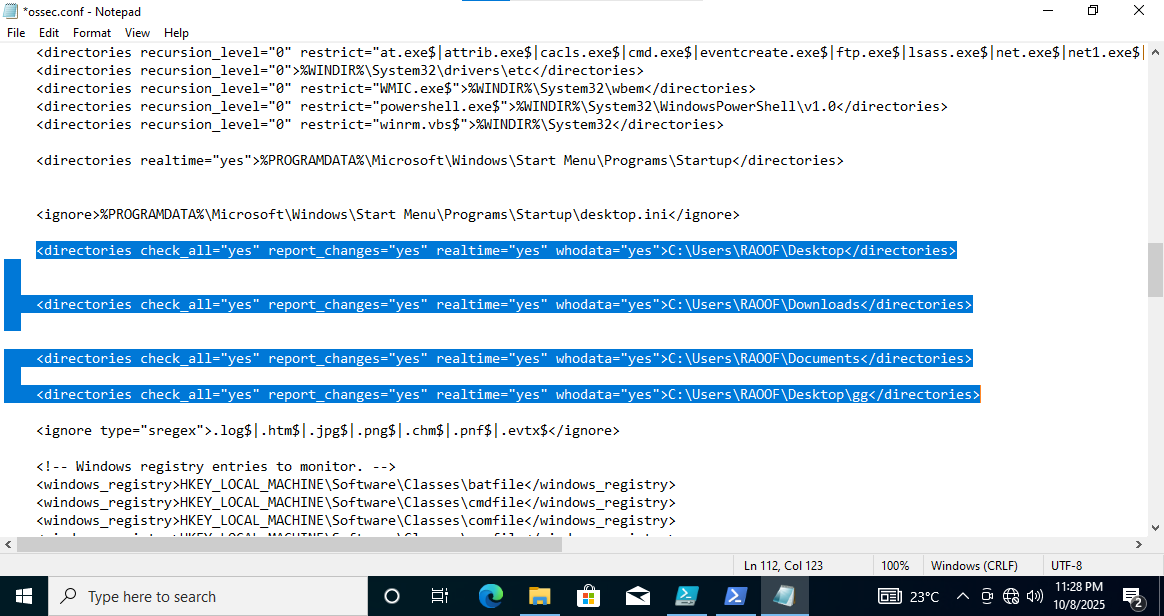
**لقطة شاشة: تظهر مسار ملف ossec.conf في Windows Explorer**



* أضف أو تأكد من وجود الأدلة التالية في قسم <syscheck>:

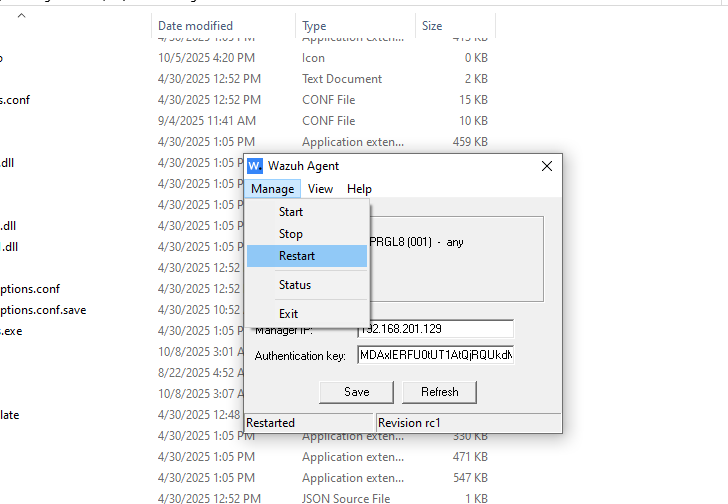
<syscheck>  
 <disabled>no</disabled>  
 <frequency>300</frequency>  
 <scan\_on\_start>yes</scan\_on\_start>  
   
 <!-- مراقبة سطح المكتب للمستخدم Administrator -->  
 <directories check\_all="yes" report\_changes="yes" realtime="yes">C:\Users\Administrator\Desktop</directories>  
   
 <!-- مراقبة مجلد التنزيلات -->  
 <directories check\_all="yes" report\_changes="yes" realtime="yes">C:\Users\Administrator\Downloads</directories>  
   
 <!-- مراقبة مجلد المستندات -->  
 <directories check\_all="yes" report\_changes="yes" realtime="yes">C:\Users\Administrator\Documents</directories>  
   
 <!-- مراقبة دليل تثبيت Wazuh -->  
 <directories check\_all="yes" report\_changes="yes" realtime="yes">C:\Program Files (x86)\ossec-agent</directories>  
   
 <!-- مراقبة ملفات Startup -->  
 <directories check\_all="yes" realtime="yes">C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup</directories>  
</syscheck>





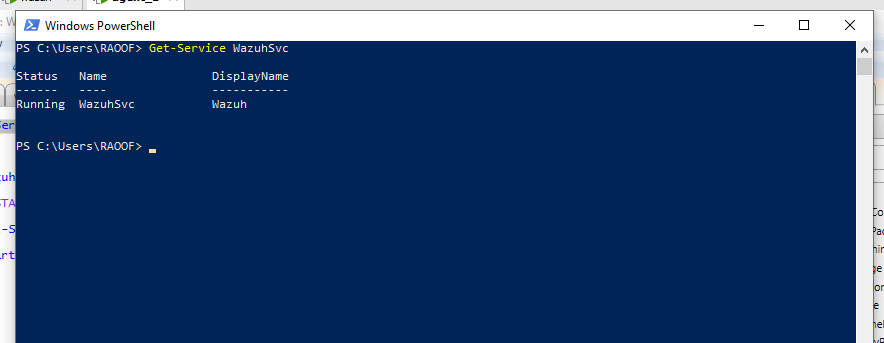
#### شرح المعاملات:

* check\_all="yes": مراقبة جميع التغييرات (الحجم، الأذونات، المالك، إلخ)
* report\_changes="yes": الإبلاغ عن محتوى التغييرات في الملف
* realtime="yes": المراقبة في الوقت الفعلي
* frequency="300": فحص دوري كل 300 ثانية (5 دقائق)
* بعد تعديل ملف التكوين، أعد تشغيل خدمة Wazuh Agent:



* تحقق من أن الخدمة أعيد تشغيلها بنجاح:

Get-Service WazuhSvc

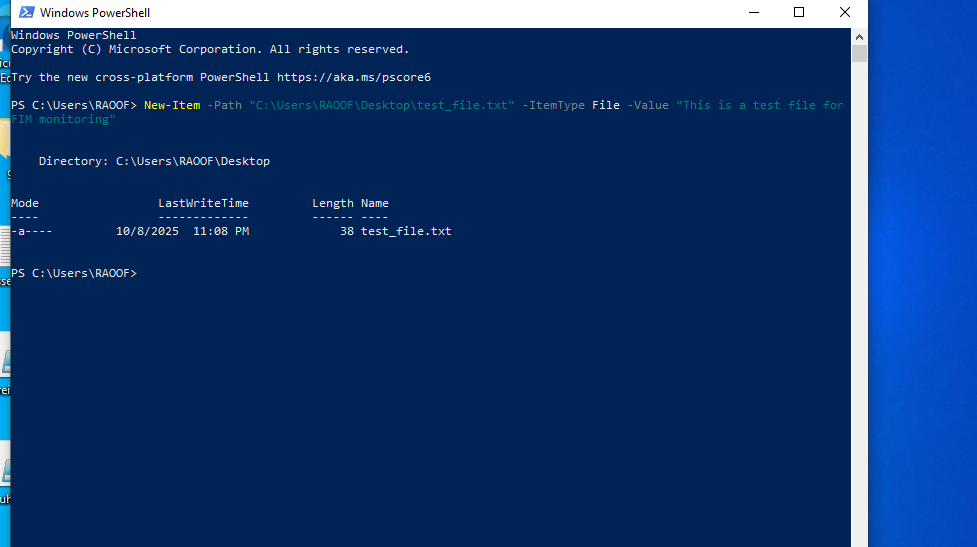


#### خطوات توليد التنبيهات

قم بإجراء تغييرات على الملفات في الأدلة المراقبة:

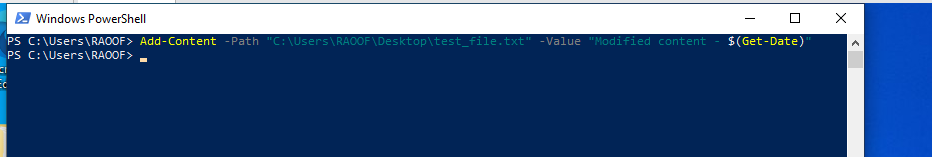
**1. إنشاء ملف جديد:**

New-Item -Path "C:\Users\<USERNAME>\Desktop\test\_file.txt" -ItemType File -Value "This is a test file for FIM monitoring"

****

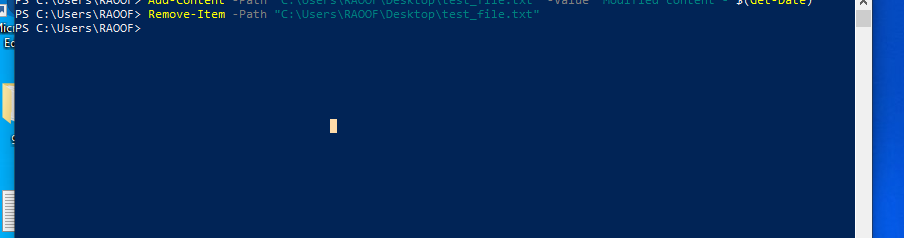
**2. تعديل ملف موجود:**

Add-Content -Path "C:\Users\RAOOF\Desktop\test\_file.txt" -Value "Modified content - $(Get-Date)"

****

**3. حذف ملف:**

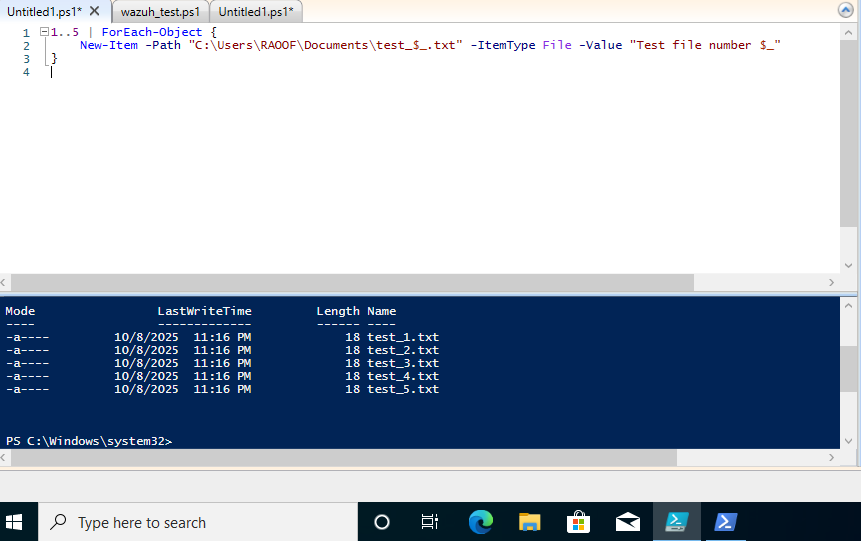
Remove-Item -Path "C:\Users\RAOOF\Desktop\test\_file.txt"

****

**4. إنشاء عدة ملفات لاختبار أشمل:**

1..5 | ForEach-Object {  
 New-Item -Path "C:\Users\RAOOF\Desktop\test\_$\_.txt" -ItemType File -Value "Test file number $\_"  
}

#### 



#### التنبيهات

يمكن العثور على التنبيهات ذات الصلة في Kibana باستخدام:

**استعلامات البحث:** -

rule.groups:"syscheck" - جميع تنبيهات FIM

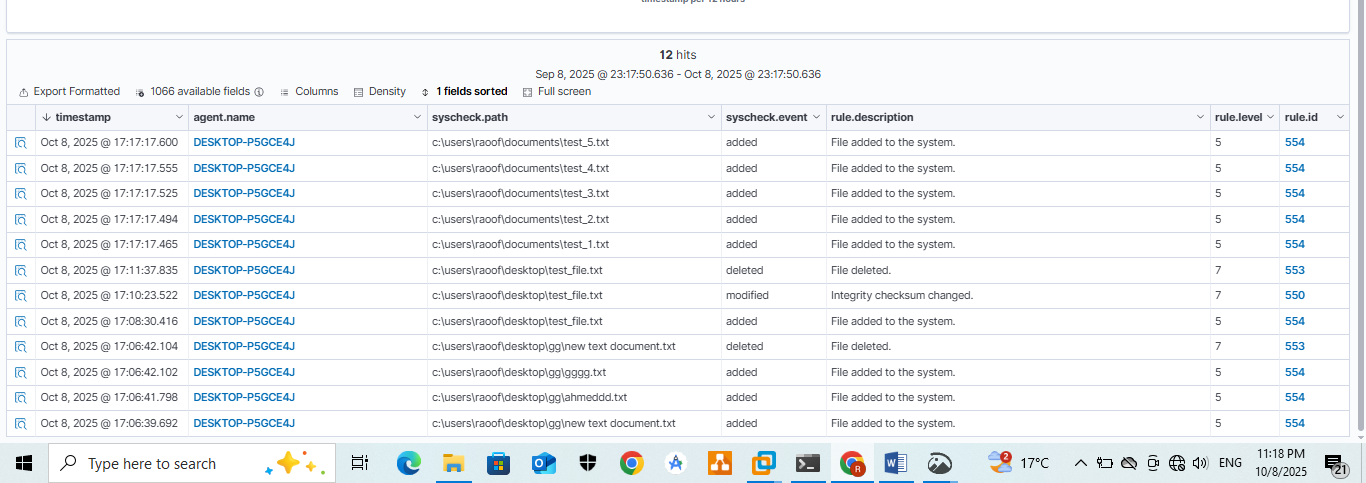
- syscheck.event:"added" - الملفات المضافة

- syscheck.event:"modified" - الملفات المعدلة

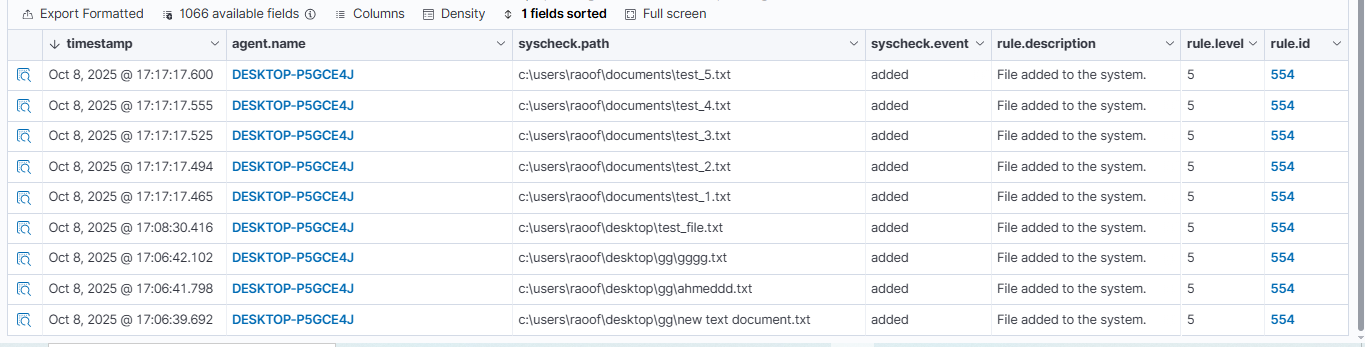
- syscheck.event:"deleted" - الملفات المحذوفة

- syscheck.path:"\*Desktop\*" - التغييرات في سطح المكتب

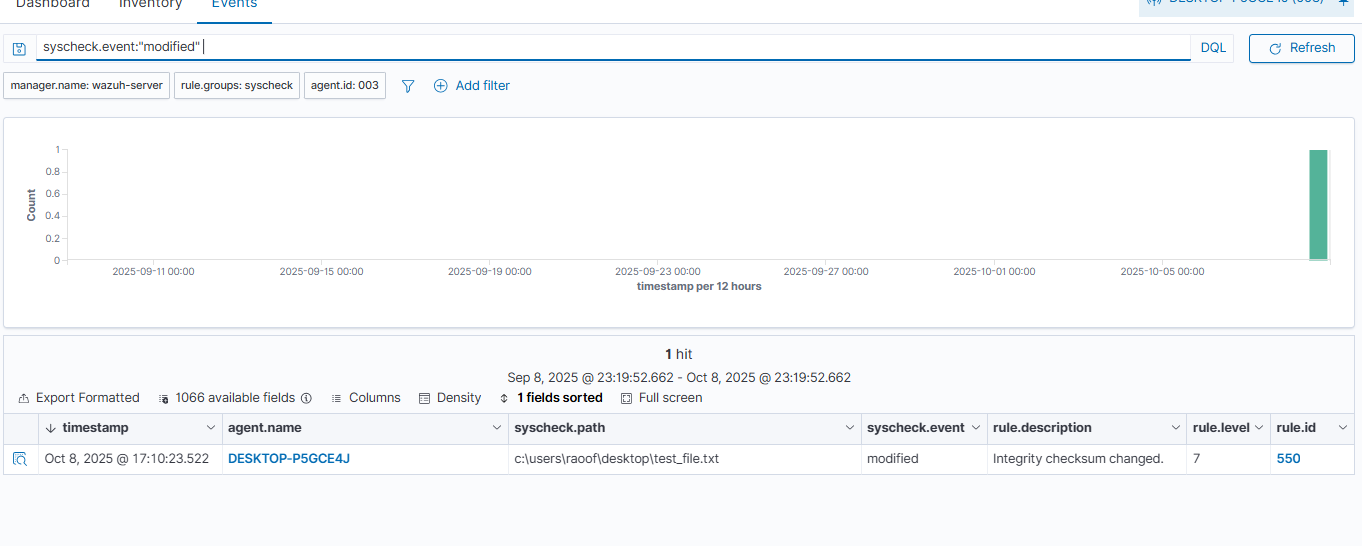
**لقطة شاشة: تظهر واجهة Kibana Dashboard مع تنبيهات FIM - عرض عام لجميع التنبيهات**

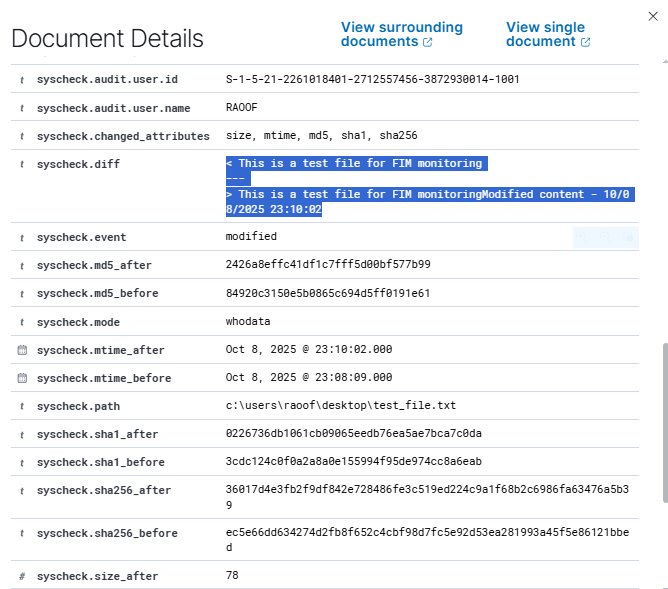


**لقطة شاشة: تظهر تفاصيل تنبيه إضافة ملف جديد (syscheck.event: added)**

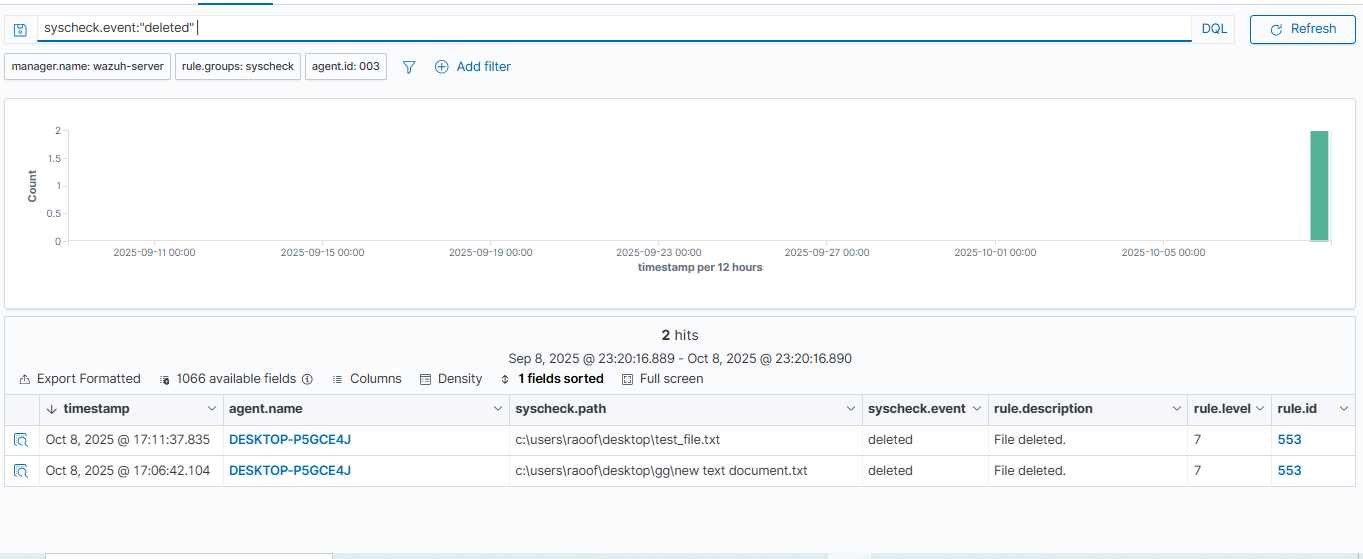


**لقطة شاشة: تظهر تنبيه تعديل ملف (syscheck.event: modified) مع التغييرات المسجلة**

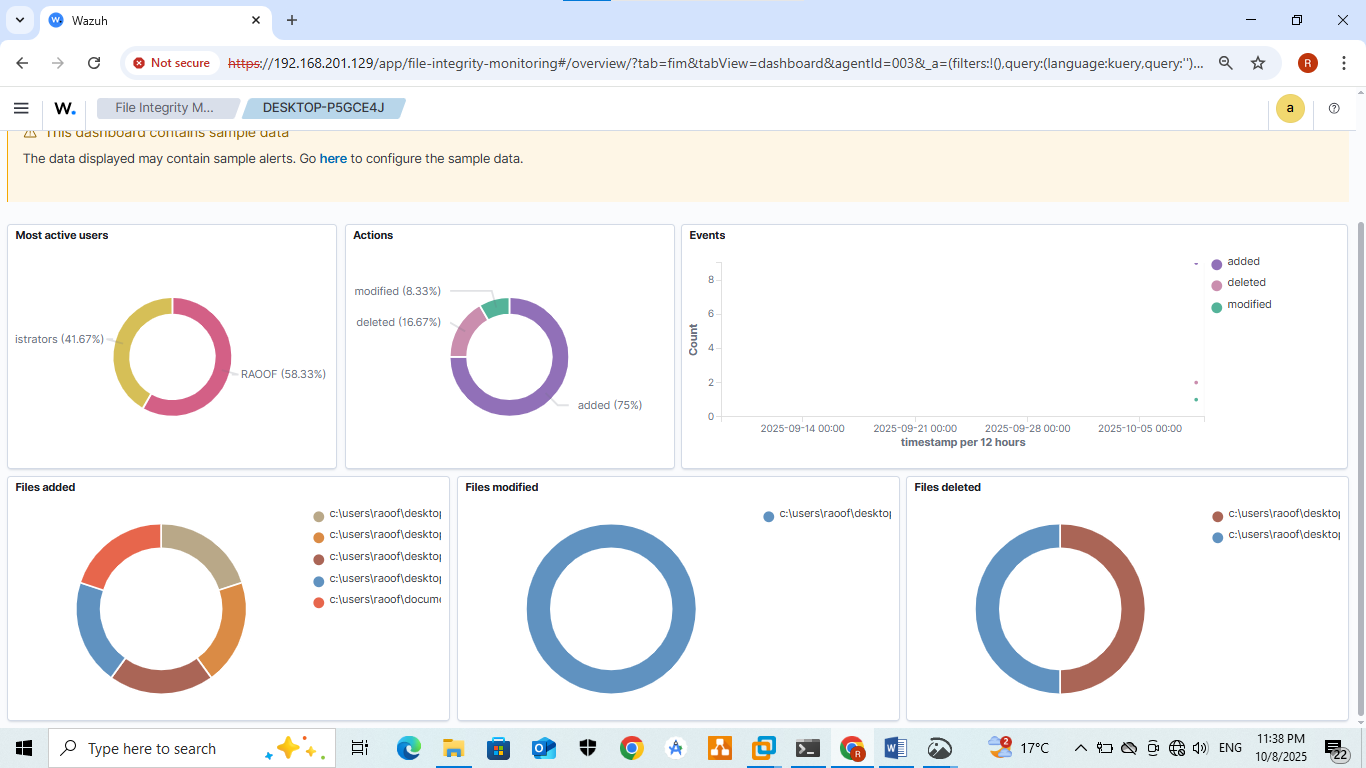
****

****

**لقطة شاشة: تظهر تنبيه حذف ملف (syscheck.event: deleted)**



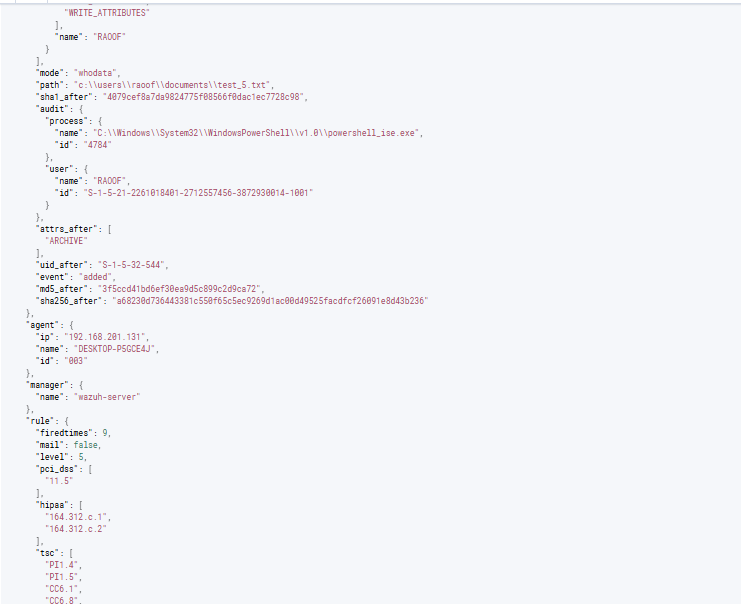
**لقطة شاشة: تظهر رسم بياني (Visualization) في Kibana يوضح توزيع أحداث FIM حسب النوع (added, modified, deleted)**



#### معلومات إضافية متوفرة في التنبيهات:

* مسار الملف الكامل
* تاريخ ووقت التغيير
* نوع الحدث (إضافة، تعديل، حذف)
* التغييرات في الصلاحيات
* التغييرات في المالك
* تغييرات حجم الملف
* Hash للملف (MD5, SHA1, SHA256)

**لقطة شاشة: تظهر JSON الكامل لأحد التنبيهات موضحًا كل الحقول المتاحة بما في ذلك hashes**



#### نقطة النهاية المتأثرة

* Windows

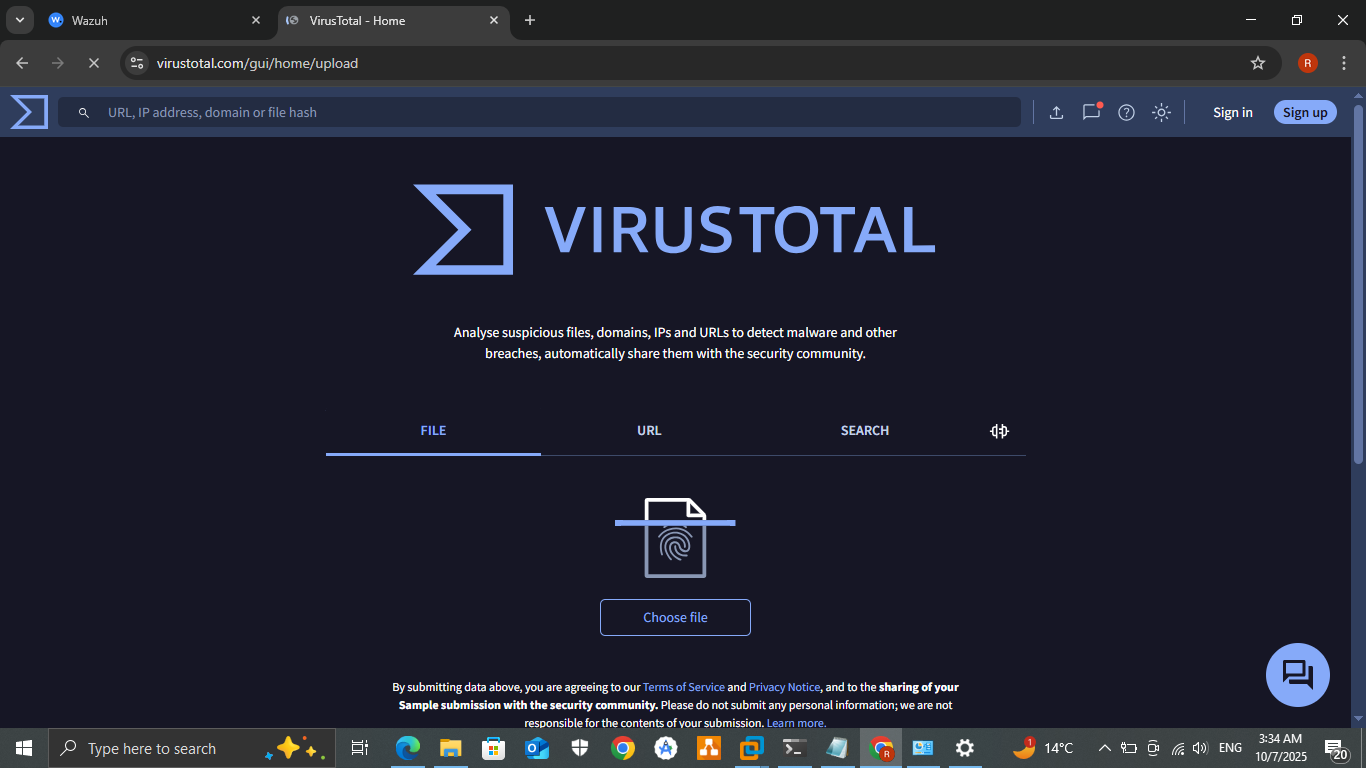
## اكتشاف وإزالة البرامج الضارة - تكامل VirusTotal

لدى Wazuh القدرة على التكامل مع VirusTotal API، وتشغيل استعلام عند اكتشاف تغيير في الملف. عندما يتم اكتشاف ملف ضار، يمكن تكوين Wazuh لإزالته تلقائيًا.

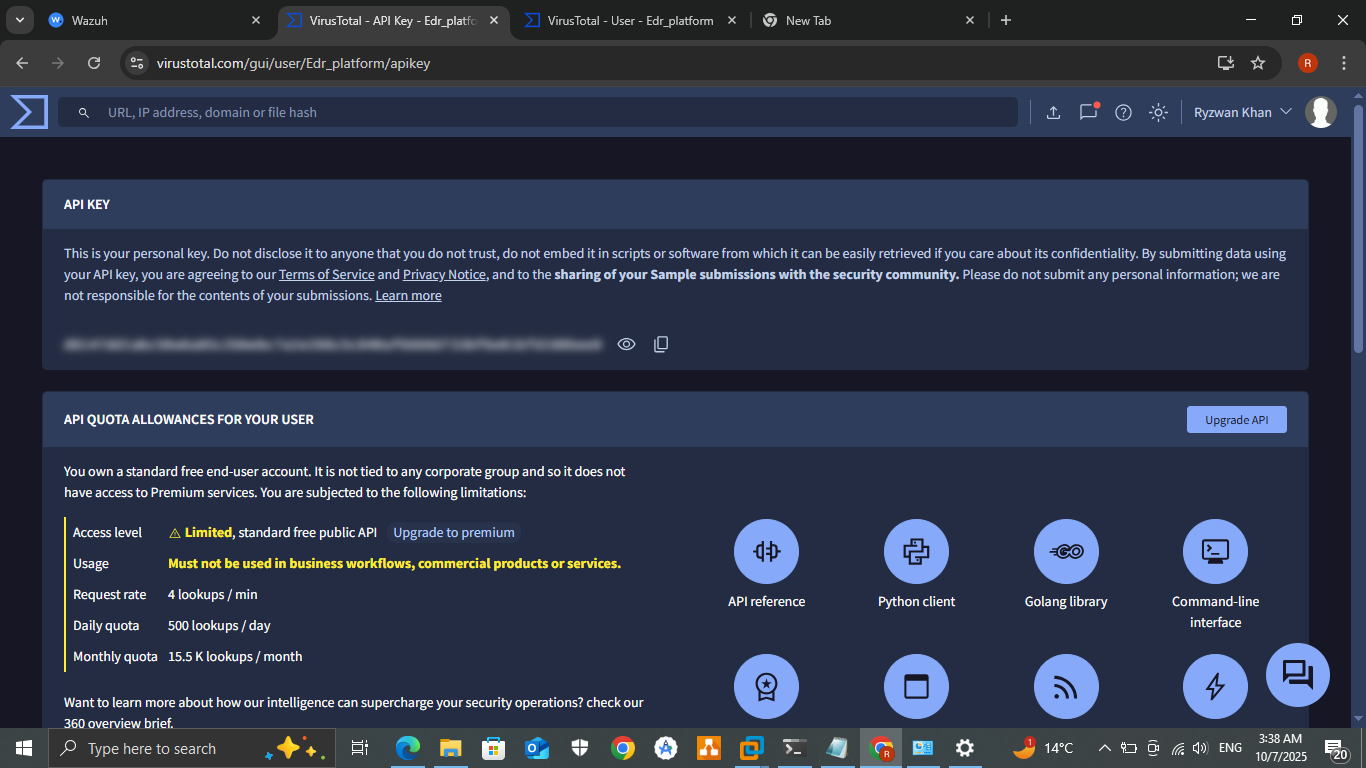
#### المتطلبات الأساسية

* مفتاح VirusTotal API (يمكن الحصول عليه مجانًا من https://www.virustotal.com)
* عميل Wazuh مثبت على Windows
* Python مثبت على مدير Wazuh

**لقطة شاشة: تظهر صفحة VirusTotal لإنشاء حساب والحصول على API Key**



**لقطة شاشة: تظهر صفحة API Key في حساب VirusTotal**



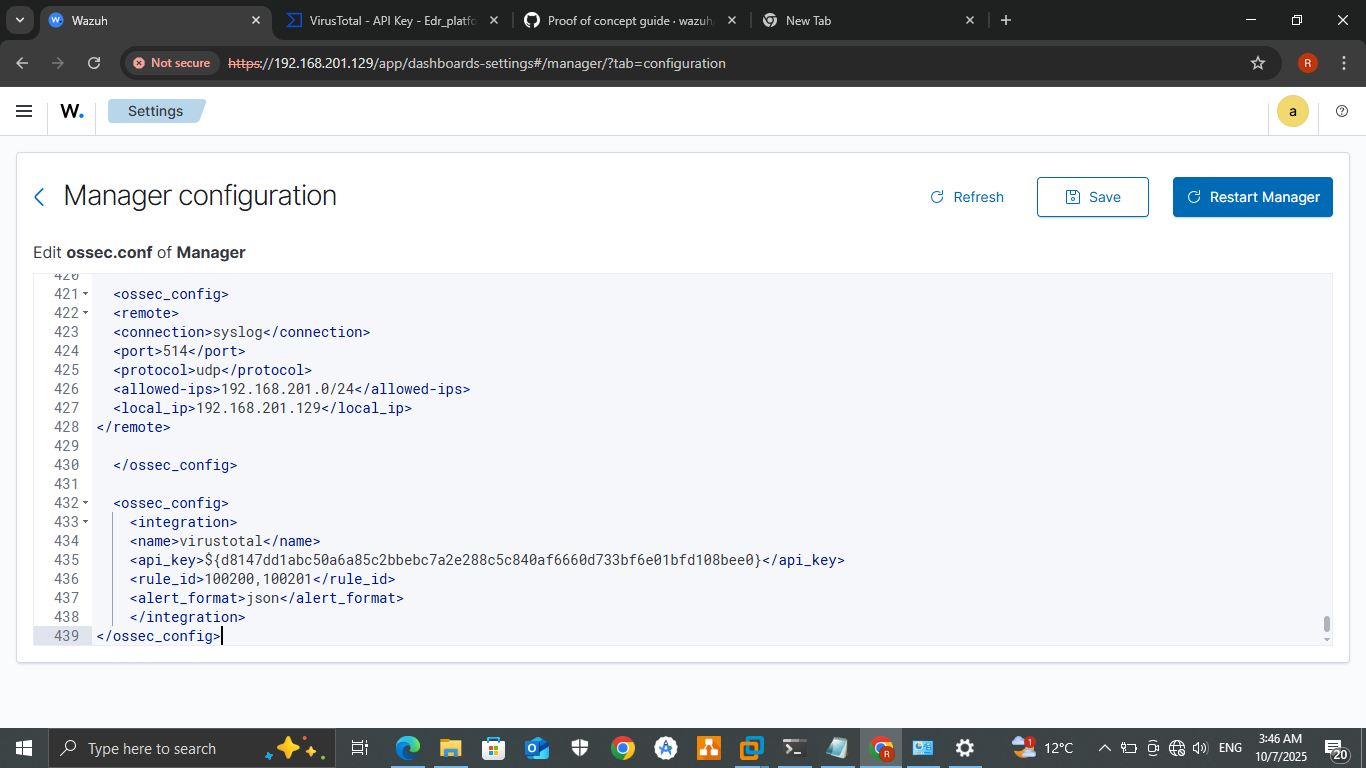
### التكوين على مدير Wazuh

#### 1. تفعيل تكامل VirusTotal

قم بتحرير ملف /var/ossec/etc/ossec.conf على مدير Wazuh:

<ossec\_config>  
 <integration>  
 <name>virustotal</name>  
 <api\_key>YOUR\_VIRUSTOTAL\_API\_KEY\_HERE</api\_key>  
 <rule\_id>100200,100201</rule\_id>  
 <alert\_format>json</alert\_format>  
 </integration>  
</ossec\_config>

**لقطة شاشة: تظهر تحرير ملف ossec.conf على مدير Wazuh مع تكوين VirusTotal integration**



#### 2. إنشاء القواعد المخصصة

أضف القواعد التالية إلى /var/ossec/etc/rules/local\_rules.xml:

<group name="syscheck,pci\_dss\_11.5,nist\_800\_53\_SI.7,">  
 <!-- قواعد لأنظمة Windows - مراقبة سطح المكتب -->  
 <rule id="100200" level="7">  
 <if\_sid>550</if\_sid>  
 <field name="file">\\Users\\RAOOF\\Desktop</field>  
 <description>File modified in RAOOF Desktop directory.</description>  
 </rule>  
   
 <rule id="100201" level="7">  
 <if\_sid>554</if\_sid>  
 <field name="file">\\Users\\RAOOF\\Desktop</field>  
 <description>File added to RAOOF Desktop directory.</description>  
 </rule>  
   
 <!-- قواعد لمجلد Downloads -->  
 <rule id="100202" level="7">  
 <if\_sid>550</if\_sid>  
 <field name="file">\\Users\\RAOOF\\Downloads</field>  
 <description>File modified in Downloads directory.</description>  
 </rule>  
   
 <rule id="100203" level="7">  
 <if\_sid>554</if\_sid>  
 <field name="file">\\Users\\RAOOF\\Downloads</field>  
 <description>File added to Downloads directory.</description>  
 </rule>  
</group>  
  
<group name="virustotal,">  
 <!-- قاعدة للملفات الضارة المكتشفة -->  
 <rule id="100300" level="12">  
 <if\_sid>87105</if\_sid>  
 <description>VirusTotal: Malicious file detected - $(data.virustotal.source.file)</description>  
 </rule>  
</group>

**لقطة شاشة: تظهر محتوى ملف local\_rules.xml مع القواعد المخصصة**

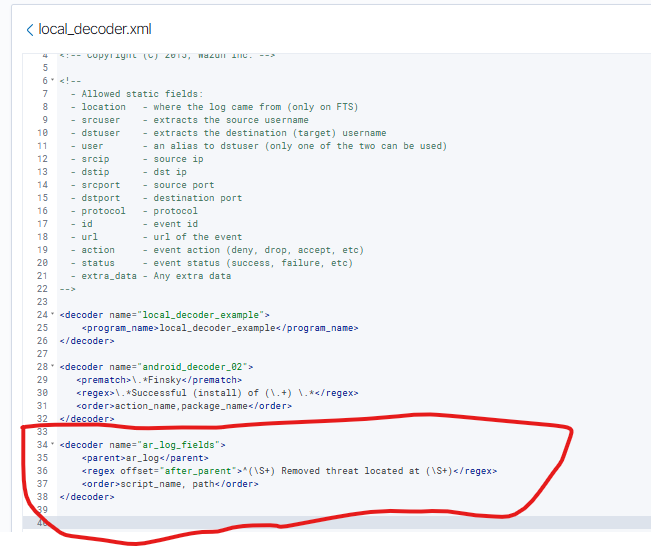


#### 3. إنشاء فك الترميز للاستجابة النشطة

قم بتحرير /var/ossec/etc/decoders/local\_decoder.xml:

<decoder name="ar\_log\_fields">  
 <parent>ar\_log</parent>  
 <regex offset="after\_parent">^(\S+) - Removed threat located at (\S+)</regex>  
 <order>script\_name, path</order>  
</decoder>

**لقطة شاشة: تظهر محتوى ملف local\_decoder.xml**

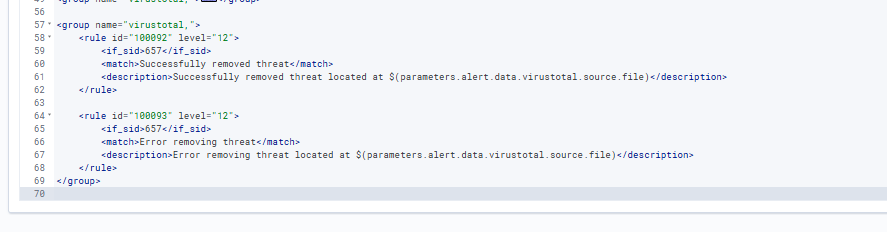


#### 4. إنشاء قواعد الإزالة التلقائية

أضف إلى /var/ossec/etc/rules/local\_rules.xml:

<group name="virustotal,">  
 <rule id="100092" level="12">  
 <if\_sid>657</if\_sid>  
 <match>Successfully removed threat</match>  
 <description>Successfully removed threat located at $(parameters.alert.data.virustotal.source.file)</description>  
 </rule>  
  
 <rule id="100093" level="12">  
 <if\_sid>657</if\_sid>  
 <match>Error removing threat</match>  
 <description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>  
 </rule>  
</group>

**لقطة شاشة: تظهر قواعد الإزالة التلقائية في local\_rules.xml**



#### 5. تكوين الاستجابة النشطة على مدير Wazuh

أضف إلى /var/ossec/etc/ossec.conf:

<ossec\_config>  
 <command>  
 <name>remove-threat-windows</name>  
 <executable>remove-threat.cmd</executable>  
 <timeout\_allowed>no</timeout\_allowed>  
 </command>  
  
 <active-response>  
 <disabled>no</disabled>  
 <command>remove-threat-windows</command>  
 <location>local</location>  
 <rules\_id>87105</rules\_id>  
 </active-response>  
</ossec\_config>

**لقطة شاشة: تظهر تكوين الاستجابة النشطة في ossec.conf على مدير Wazuh**

#### 6. إعادة تشغيل مدير Wazuh

systemctl restart wazuh-manager

**لقطة شاشة: تظهر إعادة تشغيل wazuh-manager بنجاح**

### التكوين على عميل Windows

#### 1. تكوين مراقبة سلامة الملفات

قم بتحرير C:\Program Files (x86)\ossec-agent\ossec.conf:

<syscheck>  
 <disabled>no</disabled>  
 <frequency>300</frequency>  
 <scan\_on\_start>yes</scan\_on\_start>  
   
 <!-- مراقبة سطح المكتب بالوقت الفعلي -->  
 <directories check\_all="yes" report\_changes="yes" realtime="yes">C:\Users\RAOOF\Desktop</directories>  
   
 <!-- مراقبة مجلد التنزيلات -->  
 <directories check\_all="yes" report\_changes="yes" realtime="yes">C:\Users\RAOOF\Downloads</directories>  
</syscheck>

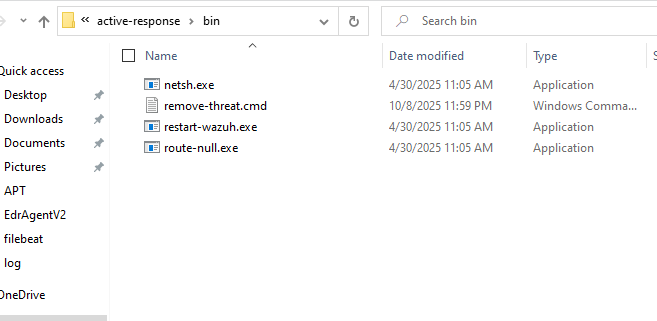
**[ضع لقطة شاشة: تظهر تكوين syscheck في ossec.conf على Windows]**

#### 2. إنشاء برنامج نصي للاستجابة النشطة

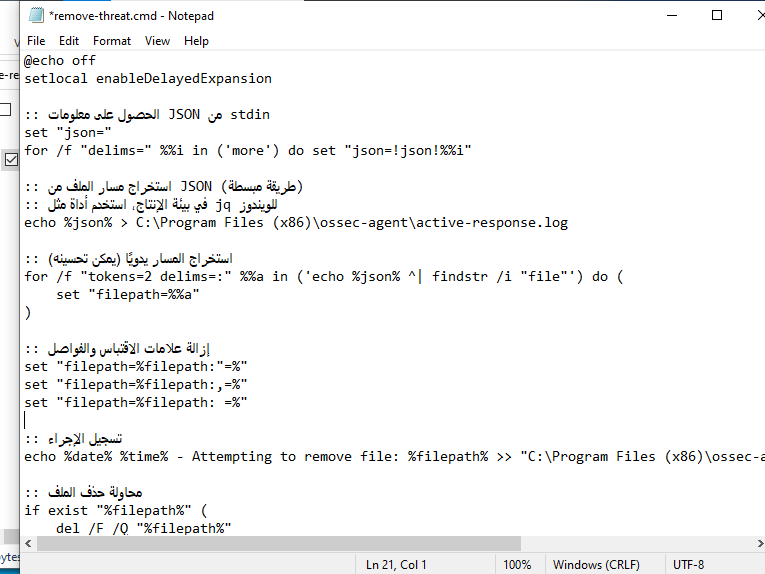
قم بإنشاء ملف remove-threat.cmd في المسار: C:\Program Files (x86)\ossec-agent\active-response\bin\remove-threat.cmd

@echo off  
setlocal enableDelayedExpansion  
  
:: الحصول على معلومات JSON من stdin  
set "json="  
for /f "delims=" %%i in ('more') do set "json=!json!%%i"  
  
:: استخراج مسار الملف من JSON (طريقة مبسطة)  
:: في بيئة الإنتاج، استخدم أداة مثل jq للويندوز  
echo %json% > C:\Program Files (x86)\ossec-agent\active-response.log  
  
:: استخراج المسار يدويًا (يمكن تحسينه)  
for /f "tokens=2 delims=:" %%a in ('echo %json% ^| findstr /i "file"') do (  
 set "filepath=%%a"  
)  
  
:: إزالة علامات الاقتباس والفواصل  
set "filepath=%filepath:"=%"  
set "filepath=%filepath:,=%"  
set "filepath=%filepath: =%"  
  
:: تسجيل الإجراء  
echo %date% %time% - Attempting to remove file: %filepath% >> "C:\Program Files (x86)\ossec-agent\active-responses.log"  
  
:: محاولة حذف الملف  
if exist "%filepath%" (  
 del /F /Q "%filepath%"  
 if !errorlevel! equ 0 (  
 echo %date% %time% - remove-threat.cmd - Successfully removed threat >> "C:\Program Files (x86)\ossec-agent\active-responses.log"  
 ) else (  
 echo %date% %time% - remove-threat.cmd - Error removing threat >> "C:\Program Files (x86)\ossec-agent\active-responses.log"  
 )  
) else (  
 echo %date% %time% - remove-threat.cmd - File not found: %filepath% >> "C:\Program Files (x86)\ossec-agent\active-responses.log"  
)  
  
exit /b 0

**لقطة شاشة: تظهر إنشاء ملف remove-threat.cmd في المسار الصحيح**



**لقطة شاشة: تظهر محتوى ملف remove-threat.cmd في Notepad**

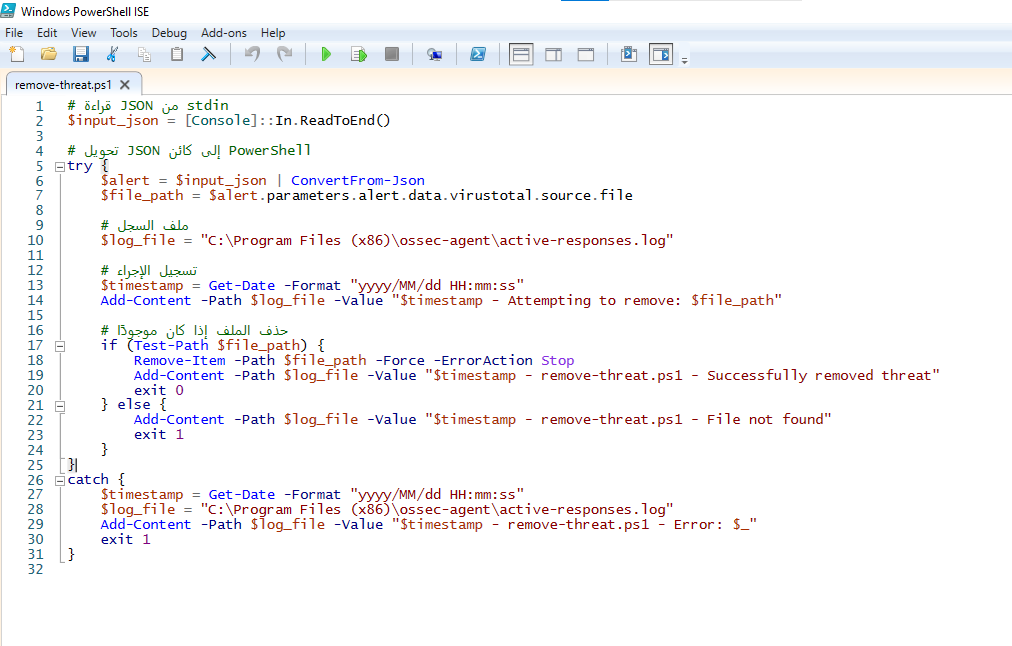


#### 3. إنشاء برنامج نصي PowerShell محسّن (اختياري)

يمكنك أيضًا إنشاء remove-threat.ps1 لحل أفضل:

# قراءة JSON من stdin  
$input\_json = [Console]::In.ReadToEnd()  
  
# تحويل JSON إلى كائن PowerShell  
try {  
 $alert = $input\_json | ConvertFrom-Json  
 $file\_path = $alert.parameters.alert.data.virustotal.source.file  
   
 # ملف السجل  
 $log\_file = "C:\Program Files (x86)\ossec-agent\active-responses.log"  
   
 # تسجيل الإجراء  
 $timestamp = Get-Date -Format "yyyy/MM/dd HH:mm:ss"  
 Add-Content -Path $log\_file -Value "$timestamp - Attempting to remove: $file\_path"  
   
 # حذف الملف إذا كان موجودًا  
 if (Test-Path $file\_path) {  
 Remove-Item -Path $file\_path -Force -ErrorAction Stop  
 Add-Content -Path $log\_file -Value "$timestamp - remove-threat.ps1 - Successfully removed threat"  
 exit 0  
 } else {  
 Add-Content -Path $log\_file -Value "$timestamp - remove-threat.ps1 - File not found"  
 exit 1  
 }  
}  
catch {  
 $timestamp = Get-Date -Format "yyyy/MM/dd HH:mm:ss"  
 $log\_file = "C:\Program Files (x86)\ossec-agent\active-responses.log"  
 Add-Content -Path $log\_file -Value "$timestamp - remove-threat.ps1 - Error: $\_"  
 exit 1  
}

**لقطة شاشة: تظهر محتوى ملف remove-threat.ps1**



#### 4. إعادة تشغيل عميل Wazuh

Restart-Service -Name wazuh

### خطوات توليد التنبيهات

#### 1. تنزيل ملف اختبار EICAR

EICAR هو ملف اختبار قياسي لبرامج مكافحة الفيروسات، وهو غير ضار لكنه يُكتشف كبرنامج ضار.

**باستخدام PowerShell:**

# الانتقال إلى سطح المكتب  
cd C:\Users\RAOOF\Desktop  
  
# تنزيل ملف EICAR  
Invoke-WebRequest -Uri "http://www.eicar.org/download/eicar.com" -OutFile "eicar.com"

**لقطة شاشة: تظهر تنفيذ أمر التنزيل في PowerShell**

**أو باستخدام المتصفح:** - افتح المتصفح واذهب إلى: http://www.eicar.org/download/eicar.com - احفظ الملف على سطح المكتب

**لقطة شاشة: تظهر صفحة تنزيل EICAR في المتصفح**

**لقطة شاشة: تظهر الملف eicar.com على سطح المكتب قبل الحذف التلقائي**

#### 2. إنشاء ملف EICAR يدويًا (بديل)

# إنشاء ملف EICAR يدويًا  
$eicar = 'X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H\*'  
Set-Content -Path "C:\Users\Administrator\Desktop\eicar\_test.txt" -Value $eicar

**لقطة شاشة: تظهر إنشاء ملف EICAR يدويًا**

#### 3. مراقبة السجلات

راقب ملف السجل للاستجابة النشطة:

Get-Content "C:\Program Files (x86)\ossec-agent\active-responses.log" -Wait

**لقطة شاشة: تظهر محتوى ملف active-responses.log في PowerShell**

### التنبيهات

يمكن العثور على التنبيهات ذات الصلة في Kibana:

**استعلامات البحث:**

1. **تنبيهات VirusTotal العامة:**

rule.groups:"virustotal"

**[ضع لقطة شاشة: تظهر جميع تنبيهات VirusTotal في Kibana Dashboard]**

1. **الملفات الضارة المكتشفة:**

data.virustotal.positives:>0

**[ضع لقطة شاشة: تظهر تنبيهات الملفات الضارة مع عدد المحركات التي اكتشفتها (positives)]**

1. **تنبيهات الإزالة الناجحة:**

rule.id:100092

**[ضع لقطة شاشة: تظهر تنبيه الإزالة الناجحة للملف الضار]**

1. **تنبيهات فشل الإزالة:**

rule.id:100093

1. **البحث عن ملف معين:**

data.virustotal.source.file:"\*eicar\*"

**[ضع لقطة شاشة: تظهر تفاصيل تنبيه VirusTotal لملف EICAR موضحًا:]** - عدد محركات مكافحة الفيروسات التي اكتشفت الملف (positives) - إجمالي عدد المحركات التي فحصت الملف (total) - مسار الملف الكامل - تاريخ ووقت الفحص

**[ضع لقطة شاشة: تظهر JSON الكامل لتنبيه VirusTotal موضحًا جميع التفاصيل التقنية]**

**[ضع لقطة شاشة: تظهر رسم بياني (Visualization) في Kibana يوضح توزيع التنبيهات حسب عدد المحركات التي اكتشفت التهديد]**

#### معلومات إضافية في التنبيهات:

* **Scan ID**: معرّف الفحص الفريد في VirusTotal
* **Positives**: عدد محركات مكافحة الفيروسات التي اكتشفت الملف كخبيث
* **Total**: إجمالي عدد المحركات التي فحصت الملف
* **Permalink**: رابط مباشر لنتائج الفحص على موقع VirusTotal
* **File hashes**: MD5, SHA1, SHA256 للملف
* **Scan date**: تاريخ ووقت الفحص

**[ضع لقطة شاشة: تظهر تفاصيل موسعة لأحد التنبيهات في Kibana مع جميع الحقول المذكورة أعلاه]**

### التحقق من نجاح الإزالة التلقائية

#### 1. التحقق من سجل الاستجابة النشطة

Get-Content "C:\Program Files (x86)\ossec-agent\active-responses.log" | Select-Object -Last 10

**[ضع لقطة شاشة: تظهر آخر 10 إدخالات في سجل الاستجابة النشطة تؤكد حذف الملف]**

#### 2. التحقق من عدم وجود الملف

Test-Path "C:\Users\Administrator\Desktop\eicar.com"  
# يجب أن يرجع False إذا تم الحذف بنجاح

**[ضع لقطة شاشة: تظهر نتيجة الأمر Test-Path تؤكد أن الملف لم يعد موجودًا]**

#### 3. عرض سطح المكتب

**[ضع لقطة شاشة: تظهر سطح المكتب بعد الحذف التلقائي - الملف الضار لم يعد موجودًا]**

### استكشاف الأخطاء وإصلاحها

#### مشكلة: لا يتم إرسال الملفات إلى VirusTotal

**الحلول:**

1. **تحقق من صحة API Key:**

# على مدير Wazuh  
grep -i "virustotal" /var/ossec/logs/ossec.log

**[ضع لقطة شاشة: تظهر سجل ossec.log للتحقق من أخطاء VirusTotal API]**

1. **تحقق من حد API:**

* المفتاح المجاني محدود بـ 4 طلبات في الدقيقة
* تأكد من عدم تجاوز الحد

**[ضع لقطة شاشة: تظهر رسالة خطأ في حال تجاوز حد API]**

#### مشكلة: الاستجابة النشطة لا تعمل

**الحلول:**

1. **تحقق من أذونات الملف:**

Get-Acl "C:\Program Files (x86)\ossec-agent\active-response\bin\remove-threat.cmd" | Format-List

**[ضع لقطة شاشة: تظهر أذونات ملف remove-threat.cmd]**

1. **تحقق من سجل العميل:**

Get-Content "C:\Program Files (x86)\ossec-agent\ossec.log" | Select-Object -Last 20

**[ضع لقطة شاشة: تظهر آخر 20 سطرًا من سجل عميل Wazuh]**

1. **اختبار البرنامج النصي يدويًا:**

# إنشاء ملف JSON اختباري  
$test\_json = @{  
 command = "add"  
 parameters = @{  
 alert = @{  
 data = @{  
 virustotal = @{  
 source = @{  
 file = "C:\Users\Administrator\Desktop\test\_file.txt"  
 }  
 }  
 }  
 }  
 }  
} | ConvertTo-Json -Depth 10  
  
# إنشاء ملف اختباري  
New-Item -Path "C:\Users\Administrator\Desktop\test\_file.txt" -ItemType File -Value "test"  
  
# تشغيل البرنامج النصي  
$test\_json | & "C:\Program Files (x86)\ossec-agent\active-response\bin\remove-threat.ps1"

**[ضع لقطة شاشة: تظهر اختبار البرنامج النصي يدويًا]**

### نقطة النهاية المتأثرة

* Windows

## كشف الثغرات الأمنية

Wazuh قادر على اكتشاف ما إذا كانت التطبيقات المثبتة تحتوي على ثغرات أمنية غير مُصلحة (CVEs) في النظام المراقب. هذه الميزة مهمة جدًا لإدارة أمان الأنظمة بشكل استباقي.

### التكوين على مدير Wazuh

#### 1. تفعيل كاشف الثغرات الأمنية

قم بتحرير /var/ossec/etc/ossec.conf على مدير Wazuh:

<ossec\_config>  
 <vulnerability-detector>  
 <enabled>yes</enabled>  
 <interval>5m</interval>  
 <ignore\_time>6h</ignore\_time>  
 <run\_on\_start>yes</run\_on\_start>  
  
 <!-- Windows OS vulnerabilities -->  
 <provider name="msu">  
 <enabled>yes</enabled>  
 <update\_interval>1h</update\_interval>  
 </provider>  
  
 <!-- National Vulnerability Database -->  
 <provider name="nvd">  
 <enabled>yes</enabled>  
 <update\_from\_year>2010</update\_from\_year>  
 <update\_interval>1h</update\_interval>  
 </provider>  
  
 </vulnerability-detector>  
</ossec\_config>

**[ضع لقطة شاشة: تظهر تكوين vulnerability-detector في ossec.conf على مدير Wazuh]**

#### شرح المعاملات:

* enabled: تفعيل الكاشف
* interval: مدة الفاصل بين عمليات الفحص (5 دقائق)
* ignore\_time: تجاهل الثغرات القديمة (6 ساعات)
* run\_on\_start: تشغيل الفحص عند بدء التشغيل
* provider name="msu": مزود ثغرات Microsoft Security Updates
* provider name="nvd": قاعدة بيانات الثغرات الوطنية

#### 2. إعادة تشغيل مدير Wazuh

systemctl restart wazuh-manager

**[ضع لقطة شاشة: تظهر إعادة تشغيل wazuh-manager بنجاح]**

#### 3. التحقق من تحديث قاعدة البيانات

# التحقق من قاعدة بيانات CVE  
ls -lh /var/ossec/queue/vulnerabilities/cve.db  
  
# مراقبة عملية التحديث في السجل  
tail -f /var/ossec/logs/ossec.log | grep -i vulnerability

**[ضع لقطة شاشة: تظهر حجم وتاريخ ملف cve.db]**

**[ضع لقطة شاشة: تظهر سجل Wazuh أثناء تحديث قاعدة بيانات الثغرات]**

### التكوين على عميل Windows

#### 1. تفعيل syscollector لجمع معلومات البرامج

قم بتحرير C:\Program Files (x86)\ossec-agent\ossec.conf:

<wodle name="syscollector">  
 <disabled>no</disabled>  
 <interval>1h</interval>  
 <scan\_on\_start>yes</scan\_on\_start>  
   
 <!-- جمع معلومات الأجهزة -->  
 <hardware>yes</hardware>  
   
 <!-- جمع معلومات نظام التشغيل -->  
 <os>yes</os>  
   
 <!-- جمع معلومات الشبكة -->  
 <network>yes</network>  
   
 <!-- جمع قائمة البرامج المثبتة - مهم لكشف الثغرات -->  
 <packages>yes</packages>  
   
 <!-- جمع معلومات التحديثات المثبتة (Hotfixes) -->  
 <hotfixes>yes</hotfixes>  
   
 <!-- جمع معلومات المنافذ المفتوحة -->  
 <ports all="no">yes</ports>  
   
 <!-- جمع معلومات العمليات الجارية -->  
 <processes>yes</processes>  
</wodle>

**[ضع لقطة شاشة: تظهر تكوين syscollector في ossec.conf على Windows]**

#### شرح المعاملات:

* packages: جمع قائمة البرامج المثبتة
* hotfixes: جمع قائمة تحديثات Windows المثبتة
* interval: تكرار الفحص (كل ساعة)
* scan\_on\_start: تشغيل الفحص عند بدء تشغيل العميل

#### 2. إعادة تشغيل عميل Wazuh

Restart-Service -Name wazuh

**[ضع لقطة شاشة: تظهر إعادة تشغيل خدمة Wazuh على Windows]**

#### 3. التحقق من عمل syscollector

# عرض آخر الأحداث في سجل Wazuh  
Get-Content "C:\Program Files (x86)\ossec-agent\ossec.log" | Select-Object -Last 30 | Select-String -Pattern "syscollector"

**[ضع لقطة شاشة: تظهر سجل العميل مع رسائل syscollector]**

### خطوات توليد التنبيهات

الثغرات الأمنية سيتم اكتشافها تلقائيًا بناءً على: 1. البرامج المثبتة على النظام 2. تحديثات Windows المفقودة 3. إصدار نظام التشغيل

#### 1. عرض البرامج المثبتة

# عرض قائمة البرامج المثبتة  
Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\\* |   
 Select-Object DisplayName, DisplayVersion, Publisher |   
 Format-Table -AutoSize

**[ضع لقطة شاشة: تظهر قائمة البرامج المثبتة في PowerShell]**

#### 2. التحقق من التحديثات المثبتة

# عرض قائمة Hotfixes المثبتة  
Get-HotFix | Sort-Object -Property InstalledOn -Descending |   
 Format-Table -AutoSize

**[ضع لقطة شاشة: تظهر قائمة Windows Updates المثبتة]**

#### 3. عرض معلومات نظام التشغيل

# معلومات نظام التشغيل  
Get-ComputerInfo | Select-Object WindowsProductName, WindowsVersion, OsHardwareAbstractionLayer

**[ضع لقطة شاشة: تظهر معلومات نظام التشغيل Windows]**

#### 4. محاكاة نظام به ثغرات (اختياري - للاختبار فقط)

لاختبار الميزة بشكل أفضل، يمكنك: - استخدام نظام Windows قديم (مثل Windows Server 2012 أو Windows 7) - تثبيت برامج قديمة معروف أن بها ثغرات - عدم تثبيت آخر التحديثات الأمنية

**ملاحظة مهمة:** هذا للاختبار فقط في بيئة معزولة!

### التنبيهات

يمكن العثور على التنبيهات ذات الصلة في Kibana:

#### استعلامات البحث:

1. **جميع تنبيهات الثغرات الأمنية:**

rule.groups:"vulnerability-detector"

**[ضع لقطة شاشة: تظهر جميع تنبيهات كاشف الثغرات في Kibana Dashboard]**

1. **الثغرات حسب مستوى الخطورة:**

data.vulnerability.severity:"High"  
data.vulnerability.severity:"Critical"  
data.vulnerability.severity:"Medium"

**[ضع لقطة شاشة: تظهر تنبيهات الثغرات مصنفة حسب مستوى الخطورة (High, Critical, Medium)]**

1. **الثغرات في برنامج معين:**

data.vulnerability.package.name:"<package\_name>"

1. **ثغرات CVE محددة:**

data.vulnerability.cve:"CVE-2021-\*"

**[ضع لقطة شاشة: تظهر البحث عن CVE معين]**

1. **الثغرات التي لها تصحيح متوفر:**

data.vulnerability.status:"Available"

### تفاصيل التنبيهات

كل تنبيه ثغرة أمنية يحتوي على المعلومات التالية:

**[ضع لقطة شاشة: تظهر تفاصيل تنبيه ثغرة أمنية كاملة في Kibana موضحًا:]**

* **CVE ID**: رقم الثغرة (مثل: CVE-2021-34527)
* **Title**: عنوان الثغرة
* **Severity**: مستوى الخطورة (Critical, High, Medium, Low)
* **Package**: اسم البرنامج أو المكون المتأثر
* **Version**: إصدار البرنامج المتأثر
* **Architecture**: معمارية النظام (x86, x64)
* **CVSS Score**: درجة CVSS (من 0 إلى 10)
* **Published Date**: تاريخ نشر الثغرة
* **Updated Date**: تاريخ آخر تحديث
* **Status**: حالة التصحيح (Pending, Available, Fixed)
* **References**: روابط لمزيد من المعلومات

**[ضع لقطة شاشة: تظهر JSON الكامل لتنبيه ثغرة أمنية]**

### Visualizations في Kibana

يمكن إنشاء لوحات معلومات مفيدة لعرض الثغرات:

#### 1. توزيع الثغرات حسب الخطورة

**[ضع لقطة شاشة: تظهر Pie Chart توضح توزيع الثغرات حسب مستوى الخطورة]**

#### 2. أهم البرامج المتأثرة

**[ضع لقطة شاشة: تظهر Bar Chart توضح أكثر البرامج التي بها ثغرات أمنية]**

#### 3. اتجاه الثغرات عبر الزمن

**[ضع لقطة شاشة: تظهر Line Chart توضح اكتشاف الثغرات عبر الوقت]**

#### 4. درجات CVSS

**[ضع لقطة شاشة: تظهر توزيع درجات CVSS للثغرات المكتشفة]**

### التوصيات والإجراءات

عند اكتشاف ثغرات أمنية:

1. **مراجعة الثغرات ذات الأولوية العالية:**
   * ركز على الثغرات ذات التصنيف Critical و High
   * راجع درجة CVSS (الثغرات بدرجة 7 أو أعلى تحتاج عناية فورية)
2. **التحقق من توفر التصحيحات:**

# البحث عن التحديثات المتوفرة  
Get-WindowsUpdate -AcceptAll -Download -Install

**[ضع لقطة شاشة: تظهر البحث عن التحديثات في Windows Update]**

1. **تحديث البرامج المتأثرة:**

# مثال: تحديث برنامج معين عبر Chocolatey  
choco upgrade <package-name> -y

1. **توثيق الإجراءات المتخذة:**
   * سجل الثغرات التي تم إصلاحها
   * وثق الثغرات المقبولة (مع التبرير)
   * تتبع خطط التصحيح للثغرات المعلقة

**[ضع لقطة شاشة: تظهر جدول Excel أو مستند يوضح تتبع الثغرات والإجراءات المتخذة]**

### إنشاء تقرير الثغرات الأمنية

يمكن استخدام Kibana Canvas أو Reporting لإنشاء تقارير دورية:

**[ضع لقطة شاشة: تظهر تقرير PDF تم إنشاؤه من Kibana يتضمن ملخص الثغرات الأمنية]**

### استكشاف الأخطاء وإصلاحها

#### مشكلة: لا يتم اكتشاف ثغرات أمنية

**الحلول:**

1. **تحقق من تحديث قاعدة البيانات:**

# على مدير Wazuh  
ls -lh /var/ossec/queue/vulnerabilities/cve.db  
# يجب أن يكون الملف موجودًا وحجمه عدة ميجابايتات

**[ضع لقطة شاشة: تظهر التحقق من وجود وحجم ملف cve.db]**

1. **تحقق من جمع معلومات syscollector:**

# على مدير Wazuh  
/var/ossec/bin/wazuh-db <agent-id> sql "select \* from sys\_programs limit 10"

**[ضع لقطة شاشة: تظهر نتيجة استعلام قاعدة البيانات للبرامج المثبتة]**

1. **تحقق من سجلات المدير:**

grep -i "vulnerability" /var/ossec/logs/ossec.log | tail -20

**[ضع لقطة شاشة: تظهر سجل ossec.log مع رسائل vulnerability detector]**

### نقطة النهاية المتأثرة

* Windows

## تكامل Slack

يمكن لـ Wazuh إرسال التنبيهات تلقائيًا إلى قنوات Slack، مما يسمح للفريق الأمني بتلقي إشعارات فورية عن الأحداث الأمنية المهمة.

### المتطلبات الأساسية

#### 1. إنشاء Slack Workspace (إذا لم يكن موجودًا)

* اذهب إلى https://slack.com/create
* أنشئ workspace جديد أو استخدم واحد موجود

**[ضع لقطة شاشة: تظهر صفحة إنشاء Slack Workspace]**

#### 2. إنشاء Incoming Webhook

1. اذهب إلى: https://api.slack.com/apps
2. انقر على “Create New App”
3. اختر “From scratch”
4. أدخل اسم التطبيق (مثل: “Wazuh Security Alerts”)
5. اختر Workspace

**[ضع لقطة شاشة: تظهر صفحة إنشاء تطبيق Slack جديد]**

1. من القائمة الجانبية، اختر “Incoming Webhooks”
2. فعّل “Activate Incoming Webhooks”
3. انقر على “Add New Webhook to Workspace”
4. اختر القناة التي تريد إرسال التنبيهات إليها
5. انسخ Webhook URL

**[ضع لقطة شاشة: تظهر صفحة Incoming Webhooks في Slack API]**

**[ضع لقطة شاشة: تظهر اختيار القناة لإرسال التنبيهات]**

**[ضع لقطة شاشة: تظهر Webhook URL (مع إخفاء الأجزاء الحساسة)]**

مثال على شكل Webhook URL:

https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXX

### التكوين على مدير Wazuh

#### 1. إضافة تكوين Slack Integration

قم بتحرير /var/ossec/etc/ossec.conf على مدير Wazuh:

<ossec\_config>  
 <integration>  
 <name>slack</name>  
 <hook\_url>https://hooks.slack.com/services/YOUR/WEBHOOK/URL</hook\_url>  
 <level>10</level>  
 <alert\_format>json</alert\_format>  
 </integration>  
</ossec\_config>

**[ضع لقطة شاشة: تظهر إضافة تكوين Slack في ossec.conf]**

#### شرح المعاملات:

* name: اسم التكامل (slack)
* hook\_url: عنوان Webhook الذي حصلت عليه من Slack
* level: مستوى التنبيهات التي سيتم إرسالها (10 = High, 12 = Critical فقط)
* alert\_format: تنسيق التنبيه (json)

#### 2. تكوين متقدم - تخصيص التنبيهات

يمكنك تخصيص التنبيهات المُرسلة:

<ossec\_config>  
 <!-- إرسال جميع التنبيهات من مستوى 10 وأعلى -->  
 <integration>  
 <name>slack</name>  
 <hook\_url>https://hooks.slack.com/services/YOUR/WEBHOOK/URL</hook\_url>  
 <level>10</level>  
 <alert\_format>json</alert\_format>  
 </integration>  
  
 <!-- إرسال تنبيهات محددة بناءً على القواعد -->  
 <integration>  
 <name>slack</name>  
 <hook\_url>https://hooks.slack.com/services/YOUR/CRITICAL/WEBHOOK/URL</hook\_url>  
 <rule\_id>100092,100093,87105</rule\_id>  
 <alert\_format>json</alert\_format>  
 </integration>  
  
 <!-- إرسال تنبيهات محددة بناءً على المجموعة -->  
 <integration>  
 <name>slack</name>  
 <hook\_url>https://hooks.slack.com/services/YOUR/MALWARE/WEBHOOK/URL</hook\_url>  
 <group>virustotal,vulnerability-detector</group>  
 <alert\_format>json</alert\_format>  
 </integration>  
</ossec\_config>

**[ضع لقطة شاشة: تظهر تكوينات Slack متعددة لقنوات مختلفة]**

#### 3. إعادة تشغيل مدير Wazuh

systemctl restart wazuh-manager

**[ضع لقطة شاشة: تظهر إعادة تشغيل wazuh-manager]**

#### 4. التحقق من تفعيل Integration

# التحقق من سجل Wazuh  
grep -i "slack\|integrator" /var/ossec/logs/ossec.log | tail -20

**[ضع لقطة شاشة: تظهر سجل ossec.log مع رسائل تأكيد تفعيل Slack integration]**

### خطوات توليد واختبار التنبيهات

#### 1. إنشاء تنبيه اختباري بسيط

على جهاز Windows المراقب، قم بإنشاء ملف في موقع مراقب:

New-Item -Path "C:\Users\Administrator\Desktop\slack\_test.txt" -ItemType File -Value "Testing Slack Integration - $(Get-Date)"

**[ضع لقطة شاشة: تظهر إنشاء ملف اختبار في PowerShell]**

#### 2. إنشاء تنبيه عالي الأولوية - VirusTotal

# تنزيل ملف EICAR  
Invoke-WebRequest -Uri "http://www.eicar.org/download/eicar.com" -OutFile "C:\Users\Administrator\Desktop\eicar\_slack\_test.com"

**[ضع لقطة شاشة: تظهر تنزيل ملف EICAR للاختبار]**

#### 3. إنشاء تنبيه هجوم القوة الغاشمة

من جهاز آخر، قم بمحاولات تسجيل دخول RDP فاشلة:

hydra -l Administrator -p wrong\_password <windows-ip> rdp

**[ضع لقطة شاشة: تظهر محاولات RDP الفاشلة]**

### عرض التنبيهات في Slack

#### 1. التنبيهات العامة

**[ضع لقطة شاشة: تظهر قناة Slack مع عدة تنبيهات من Wazuh]**

#### 2. تفاصيل تنبيه File Integrity Monitoring

**[ضع لقطة شاشة: تظهر تنبيه FIM في Slack موضحًا:]** - اسم الملف المتأثر - نوع الحدث (added, modified, deleted) - التاريخ والوقت - اسم الجهاز (Agent) - مستوى التنبيه

#### 3. تفاصيل تنبيه VirusTotal

**[ضع لقطة شاشة: تظهر تنبيه VirusTotal في Slack موضحًا:]** - اسم الملف الضار - عدد المحركات التي اكتشفته - رابط VirusTotal - حالة الإزالة

#### 4. تفاصيل تنبيه Brute Force

**[ضع لقطة شاشة: تظهر تنبيه هجوم القوة الغاشمة في Slack موضحًا:]** - نوع الهجوم - عنوان IP المصدر - عدد المحاولات - اسم المستخدم المستهدف

### تخصيص شكل التنبيهات في Slack

#### 1. إنشاء برنامج نصي مخصص

يمكن إنشاء برنامج نصي Python مخصص لتنسيق التنبيهات بشكل أفضل:

على مدير Wazuh، قم بإنشاء /var/ossec/integrations/custom-slack:

#!/usr/bin/env python3  
  
import json  
import sys  
import requests  
from datetime import datetime  
  
# قراءة JSON من stdin  
alert\_json = json.loads(sys.stdin.read())  
  
# استخراج المعلومات  
alert\_level = alert\_json.get('rule', {}).get('level', 0)  
rule\_description = alert\_json.get('rule', {}).get('description', 'N/A')  
agent\_name = alert\_json.get('agent', {}).get('name', 'N/A')  
timestamp = alert\_json.get('timestamp', 'N/A')  
  
# تحديد اللون حسب مستوى الخطورة  
if alert\_level >= 12:  
 color = "#FF0000" # أحمر - Critical  
 severity\_emoji = "🚨"  
elif alert\_level >= 10:  
 color = "#FFA500" # برتقالي - High  
 severity\_emoji = "⚠️"  
elif alert\_level >= 7:  
 color = "#FFFF00" # أصفر - Medium  
 severity\_emoji = "⚡"  
else:  
 color = "#00FF00" # أخضر - Low  
 severity\_emoji = "ℹ️"  
  
# تنسيق الرسالة  
slack\_msg = {  
 "attachments": [  
 {  
 "color": color,  
 "title": f"{severity\_emoji} Wazuh Security Alert - Level {alert\_level}",  
 "text": rule\_description,  
 "fields": [  
 {  
 "title": "Agent",  
 "value": agent\_name,  
 "short": True  
 },  
 {  
 "title": "Timestamp",  
 "value": timestamp,  
 "short": True  
 }  
 ],  
 "footer": "Wazuh SIEM",  
 "footer\_icon": "https://wazuh.com/favicon.ico",  
 "ts": int(datetime.now().timestamp())  
 }  
 ]  
}  
  
# إرسال إلى Slack  
webhook\_url = "YOUR\_SLACK\_WEBHOOK\_URL"  
response = requests.post(webhook\_url, json=slack\_msg)  
  
sys.exit(0)

**[ضع لقطة شاشة: تظهر محتوى ملف custom-slack في محرر نصوص]**

#### 2. إعطاء الأذونات اللازمة

chmod 750 /var/ossec/integrations/custom-slack  
chown root:ossec /var/ossec/integrations/custom-slack

**[ضع لقطة شاشة: تظهر تنفيذ أوامر chmod و chown]**

### إعداد قنوات متعددة

يمكن إرسال أنواع مختلفة من التنبيهات إلى قنوات Slack مختلفة:

#### 1. قناة للتنبيهات الحرجة فقط

<integration>  
 <name>slack</name>  
 <hook\_url>https://hooks.slack.com/services/YOUR/CRITICAL/WEBHOOK</hook\_url>  
 <level>12</level>  
 <alert\_format>json</alert\_format>  
</integration>

**[ضع لقطة شاشة: تظهر قناة #security-critical في Slack مع التنبيهات الحرجة فقط]**

#### 2. قناة للبرامج الضارة

<integration>  
 <name>slack</name>  
 <hook\_url>https://hooks.slack.com/services/YOUR/MALWARE/WEBHOOK</hook\_url>  
 <group>virustotal,yara,rootcheck</group>  
 <alert\_format>json</alert\_format>  
</integration>

**[ضع لقطة شاشة: تظهر قناة #malware-alerts في Slack]**

#### 3. قناة للثغرات الأمنية

<integration>  
 <name>slack</name>  
 <hook\_url>https://hooks.slack.com/services/YOUR/VULNERABILITIES/WEBHOOK</hook\_url>  
 <group>vulnerability-detector</group>  
 <alert\_format>json</alert\_format>  
</integration>

**[ضع لقطة شاشة: تظهر قناة #vulnerabilities في Slack]**

#### 4. قناة لهجمات الشبكة

<integration>  
 <name>slack</name>  
 <hook\_url>https://hooks.slack.com/services/YOUR/NETWORK/WEBHOOK</hook\_url>  
 <group>authentication\_failed,attack,</group>  
 <level>10</level>  
 <alert\_format>json</alert\_format>  
</integration>

**[ضع لقطة شاشة: تظهر قناة #network-attacks في Slack]**

### إنشاء إجراءات تفاعلية في Slack

يمكن إضافة أزرار للإجراءات السريعة:

#### مثال: زر لحظر IP

slack\_msg = {  
 "attachments": [  
 {  
 "color": color,  
 "title": f"{severity\_emoji} Brute Force Attack Detected",  
 "text": f"Multiple failed login attempts from {source\_ip}",  
 "actions": [  
 {  
 "type": "button",  
 "text": "Block IP",  
 "url": f"https://your-wazuh-api/security/block?ip={source\_ip}"  
 },  
 {  
 "type": "button",  
 "text": "View in Kibana",  
 "url": f"https://your-kibana/app/wazuh#/manager/?tab=ruleset&redirectRule={rule\_id}"  
 }  
 ]  
 }  
 ]  
}

**[ضع لقطة شاشة: تظهر تنبيه Slack مع أزرار للإجراءات التفاعلية]**

### مراقبة صحة التكامل

#### 1. فحص حالة integratord

# التحقق من أن عملية integratord تعمل  
ps aux | grep integrator

**[ضع لقطة شاشة: تظهر عملية ossec-integratord قيد التشغيل]**

#### 2. مراقبة السجلات

# مراقبة سجل integrator في الوقت الفعلي  
tail -f /var/ossec/logs/integrations.log

**[ضع لقطة شاشة: تظهر سجل integrations.log مع رسائل إرسال ناجحة إلى Slack]**

#### 3. إحصائيات الإرسال

# عد عدد التنبيهات المرسلة  
grep -c "slack" /var/ossec/logs/integrations.log

**[ضع لقطة شاشة: تظهر إحصائيات عدد التنبيهات المرسلة]**

### استكشاف الأخطاء وإصلاحها

#### مشكلة: لا يتم إرسال التنبيهات إلى Slack

**الحلول:**

1. **التحقق من صحة Webhook URL:**

# اختبار Webhook يدويًا  
curl -X POST -H 'Content-type: application/json' \  
--data '{"text":"Test message from Wazuh"}' \  
YOUR\_SLACK\_WEBHOOK\_URL

**[ضع لقطة شاشة: تظهر اختبار Webhook يدويًا من سطر الأوامر]**

**[ضع لقطة شاشة: تظهر رسالة الاختبار تصل إلى Slack بنجاح]**

1. **التحقق من مستوى التنبيهات:**

# التأكد من أن التنبيهات المولدة تفوق المستوى المحدد  
grep "level" /var/ossec/logs/alerts/alerts.json | tail -20

**[ضع لقطة شاشة: تظهر مستويات التنبيهات في alerts.json]**

1. **التحقق من أخطاء integrator:**

grep -i "error\|fail" /var/ossec/logs/integrations.log | tail -20

**[ضع لقطة شاشة: تظهر سجل الأخطاء في integrations.log]**

1. **التحقق من تكوين ossec.conf:**

# التحقق من صحة تنسيق XML  
/var/ossec/bin/verify-agent-conf

**[ضع لقطة شاشة: تظهر نتيجة التحقق من صحة التكوين]**

#### مشكلة: التنبيهات مكررة أو كثيرة جدًا

**الحلول:**

1. **زيادة مستوى التنبيهات:**

<!-- بدلاً من level 7، استخدم level 10 أو أعلى -->  
<integration>  
 <name>slack</name>  
 <hook\_url>YOUR\_WEBHOOK\_URL</hook\_url>  
 <level>12</level> <!-- Critical فقط -->  
 <alert\_format>json</alert\_format>  
</integration>

1. **استخدام قواعد محددة فقط:**

<integration>  
 <name>slack</name>  
 <hook\_url>YOUR\_WEBHOOK\_URL</hook\_url>  
 <rule\_id>100092,100093,87105,60122,60137</rule\_id>  
 <alert\_format>json</alert\_format>  
</integration>

**[ضع لقطة شاشة: تظهر التكوين المحدّث في ossec.conf]**

1. **استبعاد قواعد معينة:**

في /var/ossec/etc/rules/local\_rules.xml:

<rule id="100500" level="0">  
 <if\_sid>NOISY\_RULE\_ID</if\_sid>  
 <description>Excluding noisy rule from Slack</description>  
</rule>

#### مشكلة: التنبيهات لا تحتوي على معلومات كافية

**الحل:** استخدام برنامج نصي مخصص كما هو موضح في قسم “تخصيص شكل التنبيهات”

### أفضل الممارسات

#### 1. تنظيم القنوات

* قناة للتنبيهات الحرجة (Critical)
* قناة للبرامج الضارة والثغرات
* قناة لهجمات الشبكة
* قناة للتقارير الدورية

**[ضع لقطة شاشة: تظهر قائمة قنوات Slack المنظمة حسب نوع التنبيه]**

#### 2. إعداد التنبيهات المناسبة

<!-- مثال على تكوين متوازن -->  
<ossec\_config>  
 <!-- Critical: للتنبيهات الفورية -->  
 <integration>  
 <name>slack</name>  
 <hook\_url>CRITICAL\_WEBHOOK</hook\_url>  
 <level>12</level>  
 </integration>  
   
 <!-- High: للمراجعة خلال ساعات -->  
 <integration>  
 <name>slack</name>  
 <hook\_url>HIGH\_WEBHOOK</hook\_url>  
 <level>10</level>  
 </integration>  
   
 <!-- Malware: متخصص -->  
 <integration>  
 <name>slack</name>  
 <hook\_url>MALWARE\_WEBHOOK</hook\_url>  
 <group>virustotal,rootcheck</group>  
 </integration>  
</ossec\_config>

**[ضع لقطة شاشة: تظهر التكوين الكامل مع التعليقات التوضيحية]**

#### 3. إنشاء اتفاقية تسمية

* استخدم prefixes واضحة: [CRITICAL], [HIGH], [MALWARE]
* أضف emojis للتمييز السريع
* ضمّن اسم الجهاز المتأثر

**[ضع لقطة شاشة: تظهر تنبيهات Slack مع تسمية موحدة وواضحة]**

#### 4. إعداد Workflows في Slack

يمكن إنشاء workflows تلقائية في Slack:

* إنشاء ticket في نظام إدارة المهام
* إرسال إشعار لشخص معين
* تسجيل الحدث في جدول بيانات

**[ضع لقطة شاشة: تظهر إعداد Slack Workflow Builder]**

#### 5. المراجعة الدورية

# إنشاء تقرير أسبوعي لعدد التنبيهات  
cat /var/ossec/logs/integrations.log | \  
grep "slack" | \  
awk '{print $1}' | \  
sort | uniq -c | \  
tail -7

**[ضع لقطة شاشة: تظهر تقرير إحصائي لعدد التنبيهات المرسلة يوميًا]**

### إنشاء تقرير ملخص يومي

يمكن إنشاء برنامج نصي لإرسال تقرير ملخص يومي:

#!/bin/bash  
# /usr/local/bin/wazuh-daily-slack-report.sh  
  
WEBHOOK\_URL="YOUR\_SLACK\_WEBHOOK\_URL"  
DATE=$(date +%Y-%m-%d)  
  
# جمع الإحصائيات  
CRITICAL\_COUNT=$(grep -c '"level":12' /var/ossec/logs/alerts/alerts.json)  
HIGH\_COUNT=$(grep -c '"level":10' /var/ossec/logs/alerts/alerts.json)  
MALWARE\_COUNT=$(grep -c 'virustotal' /var/ossec/logs/alerts/alerts.json)  
  
# إنشاء الرسالة  
MESSAGE='{  
 "text": "📊 \*Wazuh Daily Security Report - '"$DATE"'\*",  
 "attachments": [  
 {  
 "color": "#36a64f",  
 "fields": [  
 {  
 "title": "Critical Alerts",  
 "value": "'"$CRITICAL\_COUNT"'",  
 "short": true  
 },  
 {  
 "title": "High Alerts",  
 "value": "'"$HIGH\_COUNT"'",  
 "short": true  
 },  
 {  
 "title": "Malware Detected",  
 "value": "'"$MALWARE\_COUNT"'",  
 "short": true  
 }  
 ]  
 }  
 ]  
}'  
  
# إرسال التقرير  
curl -X POST -H 'Content-type: application/json' --data "$MESSAGE" $WEBHOOK\_URL

**[ضع لقطة شاشة: تظهر محتوى البرنامج النصي للتقرير اليومي]**

إضافة إلى crontab للتشغيل اليومي:

# تشغيل التقرير يوميًا في الساعة 9 صباحًا  
0 9 \* \* \* /usr/local/bin/wazuh-daily-slack-report.sh

**[ضع لقطة شاشة: تظهر إضافة البرنامج النصي إلى crontab]**

**[ضع لقطة شاشة: تظهر التقرير اليومي في Slack مع الإحصائيات الملخصة]**

### التنبيهات

يمكن العثور على تأكيد التنبيهات المرسلة:

# في سجلات Wazuh  
grep "slack" /var/ossec/logs/integrations.log  
  
# في Slack  
# تحقق من القنوات المختلفة

**[ضع لقطة شاشة: تظهر سجل نجاح إرسال التنبيهات في integrations.log]**

**[ضع لقطة شاشة: تظهر مجموعة متنوعة من التنبيهات في قناة Slack واحدة]**

### نقطة النهاية المتأثرة

* مدير Wazuh (لإرسال التنبيهات)
* جميع نقاط النهاية Windows (كمصدر للتنبيهات)

## الخاتمة

هذا الدليل غطى أهم حالات الاستخدام لـ Wazuh على نقاط نهاية Windows:

1. ✅ **اكتشاف هجوم القوة الغاشمة على RDP** - حماية من محاولات الوصول غير المصرح بها
2. ✅ **مراقبة سلامة الملفات (FIM)** - تتبع التغييرات على الملفات المهمة
3. ✅ **اكتشاف وإزالة البرامج الضارة (VirusTotal)** - حماية استباقية من الملفات الخبيثة
4. ✅ **كشف الثغرات الأمنية** - إدارة التحديثات والتصحيحات الأمنية
5. ✅ **تكامل Slack** - إشعارات فورية للأحداث الأمنية المهمة

**[ضع لقطة شاشة: تظهر لوحة معلومات Kibana شاملة تجمع كل أنواع التنبيهات المذكورة أعلاه]**

### الخطوات التالية الموصى بها:

1. **إنشاء سياسات أمنية مخصصة:**
   * تحديد القواعد المناسبة لبيئتك
   * ضبط مستويات التنبيهات
   * إنشاء قواعد مخصصة لحالات خاصة
2. **أتمتة الاستجابات:**
   * تطوير استجابات نشطة إضافية
   * دمج مع أدوات الأتمتة (Ansible, PowerShell)
   * إنشاء playbooks للحوادث الشائعة
3. **التوثيق والامتثال:**
   * توثيق جميع الإجراءات الأمنية
   * إنشاء تقارير دورية للإدارة
   * التأكد من الامتثال للمعايير (PCI-DSS, HIPAA, إلخ)
4. **التدريب والتحسين المستمر:**
   * تدريب الفريق على استخدام Wazuh
   * مراجعة وتحديث القواعد بانتظام
   * البقاء على اطلاع بأحدث التهديدات

**[ضع لقطة شاشة: تظهر مخطط معماري يوضح كيفية عمل جميع المكونات معًا (Wazuh Manager, Windows Agents, Elasticsearch, Kibana, Slack)]**

### موارد إضافية:

* التوثيق الرسمي: https://documentation.wazuh.com
* المنتدى: https://groups.google.com/forum/#!forum/wazuh
* GitHub: https://github.com/wazuh/wazuh
* Slack Community: https://wazuh.com/community/join-us-on-slack/

**[ضع لقطة شاشة: تظهر صفحة الموارد الإضافية مفتوحة في المتصفح]**

## ملاحظات نهائية مهمة:

⚠️ **تحذيرات الأمان:** - لا تشارك API Keys أو Webhook URLs علنًا - استخدم HTTPS لجميع الاتصالات - قم بتغيير كلمات المرور الافتراضية - راجع الأذونات بانتظام

🔒 **أفضل الممارسات:** - اختبر جميع التكوينات في بيئة اختبار أولاً - احتفظ بنسخ احتياطية من ملفات التكوين - راقب استهلاك الموارد (CPU, Memory, Disk) - قم بتحديث Wazuh والمكونات بانتظام

📝 **التوثيق:** - وثّق جميع التغييرات على التكوين - احتفظ بسجل للحوادث والإجراءات المتخذة - أنشئ دليل تشغيل (Runbook) لفريقك

**[ضع لقطة شاشة: تظهر مستند توثيق نهائي ملخص لجميع الخطوات والتكوينات]**

تم إعداد هذا الدليل لمساعدتك في نشر وتشغيل Wazuh بكفاءة على بيئة Windows. تذكر أن الأمان السيبراني هو عملية مستمرة تتطلب المراقبة والتحديث المستمر.

**نتمنى لك تجربة آمنة وناجحة مع Wazuh! 🛡️**