Wazuh Server API

Wazuh sunucu API'si, bir web tarayıcısından, cURL gibi bir komut satırı aracından veya web istekleri yapabilen herhangi bir betik veya programdan Wazuh yöneticisiyle etkileşime izin veren açık kaynaklı bir RESTful API'dir. Wazuh panosu, Wazuh sunucu altyapısını uzaktan yönetmek için Wazuh sunucu API'sine güvenir. Wazuh sunucu API'sini, aracı ekleme, yöneticiyi veya aracıyı yeniden başlatma veya Dosya Bütünlüğü İzleme (FIM) hakkında ayrıntılara bakma gibi genel görevleri gerçekleştirmek için kullanabilirsiniz. Wazuh sunucusu API yeteneklerinin listesi şu şekildedir: - Wazuh acente yönetimi - Wazuh yönetici kontrolü ve genel bakışı - Küme denetimi ve genel bakış - Dosya bütünlüğü izleme denetimi ve araması - MITRE ATT&CK genel bakış - Kural seti bilgisi - Kuralların ve kod çözücülerin test edilmesi ve doğrulanması - Syscollector bilgisi - Rol Tabanlı Erişim Kontrolü (RBAC) - API yönetimi (HTTPS, yapılandırma) - Kullanıcı yönetimi - İstatistiksel bilgiler - Hata işleme - Uzaktan yapılandırmayı sorgula

- Başlarken
- Yapılandırma
- Wazuh Sunucu API'sini Güvence Altına Alma
- Rol Tabanlı Erişim Kontrolü
- Wazuh Sorgu Dili (WQL) Kullanılarak Verilerin Filtrelenmesi
- Kullanım Örnekleri

Başlarken

Bu kılavuz, Wazuh sunucu API'sini kullanmak için gereken temel bilgileri sağlar.

Wazuh Sunucu API'sini Başlatma ve Durdurma

Wazuh yöneticisini yüklediğinizde, Wazuh sunucu API'si de sürecin bir parçası olarak varsayılan olarak yüklenir. Wazuh yönetici hizmetiyle `systemctl` veya komutlarını yürüterek Wazuh sunucu API'sini yönetebilir veya izleyebilirsiniz: `service`

Systemd

```
systemctl start/status/stop/restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager start/status/stop/restart
```

Wazuh Dashboard Aracılığıyla Wazuh Sunucu API'sini Kullanma

Wazuh panosu aracılığıyla Wazuh sunucu API'siyle etkileşim kurabilirsiniz. Bunu yapmak için, yönetici ayrıcalıklarına sahip bir kullanıcıyla Wazuh panosuna giriş yapmanız gerekir. Örneğin, varsayılan `admin` kullanıcı yönetici ayrıcalıklarına sahiptir. Panodaki Wazuh sunucu API konsoluna erişmek için menü simgesine tıklayın ve **Araçlar** > **API Konsolu'na** gidin .

Pano üzerinden Wazuh sunucu API konsoluna erişin

Image not found or type unknown

API Konsolu'nda yöntemi , istek uç noktasını ve herhangi bir sorgu parametresini girin, ardından isteği yürütmek için oynat düğmesine tıklayın. Temel kavramlar hakkında daha fazla bilgi edinmek için Wazuh sunucusu API isteğini ve yanıtını anlama bölümüne bakın.

API isteğini yürüt

Image not found or type unknown

Komut Satırı Aracılığıyla Wazuh Sunucu API'sine Giriş Yapma

Güvenli erişimi sağlamak için tüm Wazuh sunucu API uç noktaları kimlik doğrulaması gerektirir. Kullanıcılar her istekte bir JSON Web Token (JWT) eklemelidir. JWT, taraflar arasında bilgileri bir JSON nesnesi olarak güvenli bir şekilde iletmek için kompakt ve kendi kendine yeten bir yöntem tanımlayan açık bir standarttır (RFC 7519). POST /security/user/authenticate kullanarak Wazuh sunucu API'sine giriş yapmak ve API uç noktalarına erişmek için gerekli bir belirteç edinmek için aşağıdaki adımları izleyin:

1. Wazuh sunucu API'sine bir kullanıcı kimlik doğrulama POST isteği göndermek ve döndürülen JWT'yi değişkende depolamak için aşağıdaki komutu çalıştırın `TOKEN`. `<WAZUH_API_USER>`ve'yi `<WAZUH_API_PASSWORD>`kimlik bilgilerinizle değiştirin.

   ```
   TOKEN=$(curl -u <WAZUH_API_USER>:<WAZUH_API_PASSWORD> -k -X POST "https://localhost:55000/security/user/authenticate?raw=true")
   ```

   > Not:
   > - `SSL`API'de (HTTPS) etkinleştirilmişse ve varsayılan kendi kendine imzalanmış sertifikaları kullanıyorsa, sunucu bağlantı doğrulamasını önlemek için parametreyi eklemeniz gerekir . cURL komutları aracılığıyla kimlik doğrulaması yaparken sorgu parametresini `-k`kullanmanızı öneririz `raw=true`, çünkü belirteci düz metin olarak döndürerek işlemeyi basitleştirir, özellikle uzun JWT'ler için yararlıdır.
   >
   > Varsayılan Wazuh sunucusu API kimlik bilgisi 'dir `wazuh:wazuh`. Ancak Wazuh dağıtımı kurulum betiği kullanılarak gerçekleştirildiyse, Wazuh API kullanıcısı 'dir ve ' komutunu çalıştırarak `wazuh`parolayı çıkarabilirsiniz .`wazuh-install-files.tar -axf wazuh-install-files.tar wazuh-install-files/wazuh-passwords.txt -O | grep -P "\'wazuh\'" -A 1`
   >
   > Eğer `wazuh` şifrenizi geri alamazsanız, kullanıcı şifresini sıfırlayabilirsiniz .

2. Jetonun oluşturulduğunu doğrulayın:

   ```
   echo $TOKEN
   ```

   Çıktı aşağıdakine benzer uzun bir dize olmalıdır:

   Output

   ```
   eyJhbGciOiJFUzUxMiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ3YXp1aCIsImF1ZCI6IldhenVoIEFQSSBSRVNUIiwibmJmIjoxM
   ```

Kimlik doğrulama başarısız olursa, çıktı bir hata mesajı görüntüler veya boş kalır. Bu gibi durumlarda, kullanıcı kimlik bilgilerinizi iki kez kontrol edin ve Wazuh sunucu API'sine ağ bağlantınız olduğundan emin olun.

3. Her şeyin beklendiği gibi çalıştığını doğrulamak için bir API isteği gönderin:

```
curl -k -X GET "https://localhost:55000/" -H "Authorization: Bearer $TOKEN"
```

Output

```
{
   "data": {
      "title": "Wazuh API REST",
      "api_version": "4.7.4",
      "revision": 40717,
      "license_name": "GPL 2.0",
      "license_url": "https://github.com/wazuh/wazuh/blob/master/LICENSE",
      "hostname": "wazuh-master",
      "timestamp": "2024-05-14T21:34:15Z"},
  "error": 0
}
```

Giriş yaptıktan sonra, aşağıdaki yapıyı kullanarak herhangi bir API uç noktasına erişebilirsiniz. `<METHOD>`İstediğiniz yöntemle ve `<ENDPOINT>`erişmek istediğiniz uç noktaya karşılık gelen dizeyle değiştirin. Bir ortam değişkeni kullanmıyorsanız, `$TOKEN`elde edilen JWT ile değiştirin.

```
curl -k -X <METHOD> "https://localhost:55000/<ENDPOINT>" -H  "Authorization: Bearer $TOKEN"
```

Scriptler Aracılığıyla Wazuh Sunucu API'sine Giriş Yapma

Bu bölüm, Wazuh sunucusuyla etkileşimleri otomatikleştirmek için temel bir adım olan komut dosyalarını kullanarak Wazuh sunucusu API'sine giriş yapma sürecini ayrıntılı olarak açıklar. Sağlanan örnekler, hem varsayılan (`false`) hem de düz metin (`true`) `raw`parametrelerini göstererek gerçek dünya uygulamalarını sergilemeyi amaçlamaktadır. `raw`Parametre, olarak ayarlandığında `true` , yanıtın düz metin veya asgari düzeyde işlenmiş bir biçimde olması gerektiği anlamına gelir. Tersine, `raw`parametre olduğunda `false`, yanıt ayrıştırma ve entegrasyonu kolaylaştırmak için daha yapılandırılmış bir JSON biçimindedir. Bu komut dosyaları, otomasyon yoluyla operasyonel verimliliğini artırmak isteyen veya özel entegrasyonlar için Wazuh sunucusu API'sine programlı olarak nasıl erişileceğini anlamak isteyen kullanıcılar için tasarlanmıştır.

Python Scriptiyle Oturum Açma

Python betiği kullanarak Wazuh sunucu API'sine kimlik doğrulaması yapabilirsiniz. Aşağıdaki betik, `wazuh_api_authenticator.py`bir JWT elde etmek için Wazuh sunucu API'siyle kimlik doğrulaması yapar. Daha sonra Wazuh aracılarının durumlarının bir özetini almak için istek başlığındaki belirteci kullanır.

```
#!/usr/bin/env python3

import json
import requests
import urllib3
from base64 import b64encode

# Disable insecure https warnings (for self-signed SSL certificates)
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

# Configuration
protocol = 'https'
host = 'localhost'
port = 55000
user = '<WAZUH_API_USER>'
password = '<WAZUH_API_PASSWORD>'
login_endpoint = 'security/user/authenticate'

login_url = f"{protocol}://{host}:{port}/{login_endpoint}"
basic_auth = f"{user}:{password}".encode()
login_headers = {'Content-Type': 'application/json',
                 'Authorization': f'Basic {b64encode(basic_auth).decode()}'}

print("\nLogin request ...\n")
response = requests.post(login_url, headers=login_headers, verify=False)
token = json.loads(response.content.decode())['data']['token']
print(token)

# New authorization header with the JWT we got
requests_headers = {'Content-Type': 'application/json',
                    'Authorization': f'Bearer {token}'}

print("\n- API calls with TOKEN environment variable ...\n")

print("Getting API information:")

response = requests.get(f"{protocol}://{host}:{port}/?pretty=true", headers=requests_headers, verify=False)
print(response.text)

print("\nGetting agents status summary:")

response = requests.get(f"{protocol}://{host}:{port}/agents/summary/status?pretty=true", headers=requests_he
print(response.text)
```

```
print("\nEnd of the script.\n")
```

`<WAZUH_API_USER>`ve `<WAZUH_API_PASSWORD>`ifadelerini doğru bilgilerle değiştirin .

Python `requests`modülünü kurun:

```
python3 -m pip install requests
```

> Not: Python modülü `urllib3`sürüm 2.0 ve üzeri yalnızca OpenSSL sürüm 1.1.1 veya üzerini destekler. Sisteminizde daha eski bir OpenSSL sürümü varsa, şunlardan birini yapmanız gerekir:
>
> - OpenSSL'i 1.1.1 veya daha üst bir sürüme yükseltin.
> - `urllib3`Mevcut OpenSSL sürümünüzle uyumlu bir sürüme geçin .

Uyumluluk sorunlarını önlemek için lütfen yazılım bağımlılıklarınızın düzgün bir şekilde hizalandığından emin olun.

Python betiğini çalıştırın `wazuh_api_authenticator.py`:

```
python3 wazuh_api_authenticator.py
```

Output

```
Login request ...
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJ3YXp1aCIsImF1ZCI6IldhenVoIEFQSSBSRVNUIiwibmJmIjoxNjAyMjM
- API calls with TOKEN environment variable ...
Getting API information:
{
  "data": {
    "title": "Wazuh API REST",
    "api_version": "4.7.4",
    "revision": 40717,
    "license_name": "GPL 2.0",
    "license_url": "https://github.com/wazuh/wazuh/blob/master/LICENSE",
    "hostname": "wazuh-master",
    "timestamp": "2024-05-14T21:34:15Z"
  },
  "error": 0
}
Getting agents status summary:
{
  "data": {
     "connection": {
        "active": 1,
        "disconnected": 0,
        "never_connected": 0,
        "pending": 0,
        "total": 1
     },
```

```
      "configuration": {
         "synced": 1,
         "not_synced": 0,
         "total": 1
      }
   },
   "error": 0
}
End of the script.
```

Bash Script Oturum Açma

Ayrıca bir Bash betiği kullanarak Wazuh sunucu API'sine kimlik doğrulaması yapabilirsiniz. Aşağıdaki betik, `wazuh_api_authenticator.sh`bir JWT elde etmek için Wazuh sunucu API'siyle kimlik doğrulaması yapar. Daha sonra Wazuh aracıları tarafından kullanılan işletim sistemlerinin bir özetini almak için istek başlığındaki belirteci kullanır.

```
#!/bin/bash

echo -e "\n- Getting token...\n"

TOKEN=$(curl -u <WAZUH_API_USER>:<WAZUH_API_PASSWORD> -k -X POST "https://localhost:55000/security/us

echo -e "\n- API calls with TOKEN environment variable ...\n"

echo -e "Getting default information:\n"

curl -k -X GET "https://localhost:55000/?pretty=true" -H  "Authorization: Bearer $TOKEN"

echo -e "\n\nGetting /agents/summary/os:\n"

curl -k -X GET "https://localhost:55000/agents/summary/os?pretty=true" -H  "Authorization: Bearer $TOKEN"

echo -e "\n\nEnd of the script.\n"
```

`<WAZUH_API_USER>`ve `<WAZUH_API_PASSWORD>`öğesini doğru kimlik bilgileriyle değiştirin

Bash betiğini çalıştırın `wazuh_api_authenticator.sh`:

```
bash wazuh_api_authenticator.sh
```

Output

```
- Getting token...
Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                             Dload  Upload   Total   Spent    Left  Speed
100  3059  100  3059    0     0  17089      0 --:--:-- --:--:-- --:--:-- 17089
- API calls with TOKEN environment variable ...
Getting default information:
{
  "data": {
```

```
    "title": "Wazuh API REST",
    "api_version": "4.7.4",
    "revision": 40717,
    "license_name": "GPL 2.0",
    "license_url": "https://github.com/wazuh/wazuh/blob/master/LICENSE",
    "hostname": "wazuh-master",
    "timestamp": "2024-05-14T21:34:15Z"
  },
  "error": 0
}
Getting /agents/summary/os:
{
  "data": {
    "affected_items": [
      "windows"
    ],
    "total_affected_items": 1,
    "total_failed_items": 0,
    "failed_items": []
  },
  "message": "Showing the operative system of all specified agents",
  "error": 0
}
End of the script.
```


Wazuh Sunucusu API İsteğini ve Yanıtını Anlama

Standart bir Wazuh sunucusu API isteği üç temel bileşenden oluşur: istek yöntemi (GET, POST, PUT veya DELETE), erişilen uç noktayı belirten API URL'si ve yetkilendirme başlığı. Bu başlık, isteği doğrulamak ve yetkilendirmek için bir JWT içermelidir. Aşağıda bir cURL isteği örneği verilmiştir:

```
curl -k -X GET "https://localhost:55000/agents/summary/os?pretty=true" -H  "Authorization: Bearer $TOKEN"
```

Her istek için cURL komutu aşağıdaki alanları içerir:

Alan	Tanım
`-X GET/POST/PUT/DELETE`	HTTP sunucusuyla iletişim kurarken kullanılacak bir istek yöntemi belirtin.
`http://<WAZUH_MANAGER_IP>:55000/<ENDPOINT>` `https://<WAZUH_MANAGER_IP>:55000/<ENDPOINT>`	Kullanılacak API URL'si. API'de SSL'nin aktif olup olmadığına bağlı olarak `http`belirtin .`https`
`-H "Authorization: Bearer <YOUR_JWT_TOKEN>"`	JWT'yi belirtmek için isteğe ek bir başlık ekleyin.
`-k`	SSL sertifika hatalarını bastırın (yalnızca varsayılan kendi kendine imzalı sertifikaları kullanıyorsanız).

Tüm yanıtlar JSON formatındadır ve çoğu şu yapıyı takip eder:

Alan	İsteğe bağlı alt alanlar	Tanım
veri	etkilenen_öğeler	İstekte başarıyla etkilenen öğelerin her birini listeleyin.
	toplam_etkilenen_öğeler	Başarıyla etkilenen öğelerin toplam sayısı.
	başarısız_öğeler	İstekteki başarısız olan her öğeyi içeren liste.
	toplam_başarısız_öğeler	Başarısız olan öğelerin toplam sayısı.
mesaj		Sonuç açıklaması.
hata		HTTP yanıtları için `200`, yanıtın tamamlanmış (`0`), başarısız (`1`) veya kısmi (`2`) olup olmadığını belirler. HTTP `4xx`veya `5xx`yanıtları için, başarısızlıkla ilgili hata kodunu belirler.

- Varsayılan olarak, veri koleksiyonları içeren yanıtlar en fazla 500 öğe döndürür. Büyük koleksiyonlar arasında yineleme yapmak için `offset`ve parametrelerini kullanabilirsiniz . Parametre 100.000 öğeye kadar izin verse de, zaman aşımı ve aşırı büyük yanıtlar gibi beklenmeyen davranışları önlemek için varsayılan 500 öğe sınırını aşmamanızı öneririz. Dikkatli kullanın.`limit``limit`
- Tüm yanıtlar bir HTTP durum kodu içerir: 2xx (başarılı), 4xx (istemci hatası), 5xx (sunucu hatası), vb.
- Tüm istekler (ve hariç) JSON yanıtını daha okunabilir bir biçime dönüştürmek için parametreyi kabul eder .`POST /security/user/authenticate``POST /security/user/authenticate/run_as``pretty`
- Wazuh sunucu API'si, seçilen günlük biçimine bağlı olarak günlükleri `api.log`veya `api.json` dosyalarında depolar. Bu günlük dosyaları Wazuh sunucusunda bulunur . Wazuh API yapılandırma dosyasında`/var/ossec/logs/` ayrıntı düzeyini değiştirebilirsiniz .
- Wazuh API günlükleri varsayılan olarak zamana göre döndürülür. Döndürme yalnızca günlüğe yeni bir giriş eklendikten sonra gerçekleşir. Örneğin, zaman tabanlı döndürme, her gece yarısı olmasa da farklı bir günde yeni bir giriş eklendiğinde tetiklenir. Döndürülmüş günlükler . `/var/ossec/logs/api/<year>/<month>/`kullanılarak depolanır ve sıkıştırılır `gzip`.
- `request_timeout`Sunucu API yapılandırma dosyasının alanında tanımlanan zaman süresinden sonra yanıt alınmazsa tüm Wazuh sunucu API istekleri iptal edilir `/var/ossec/api/configuration/api.yaml`. Bu zaman aşımını devre dışı bırakmak için parametreyi kullanabilirsiniz ; bu, özellikle PUT /agents/upgrade`wait_for_complete` gibi beklenen süreyi aşabilecek çağrılar için yararlıdır .

> Not: Maksimum API yanıt süresini ayarlamak için Wazuh sunucusundaki dosyadaki `request_timeout`değeri güncelleyin.`/var/ossec/api/configuration/api.yaml`

Hata içermeyen örnek yanıt (HTTP durum kodu 200):

Output

```
{
 "data": {
  "affected_items": [
   "master-node",
   "worker1"
  ],
  "total_affected_items": 2,
  "failed_items": [],
  "total_failed_items": 0
 },
 "message": "Restart request sent to all specified nodes",
 "error": 0
}
```


Hatalı örnek yanıt (HTTP durum kodu 200):

Output


```
{
 "data": {
  "affected_items": [],
  "total_affected_items": 0,
  "total_failed_items": 4,
  "failed_items": [
   {
    "error": {
     "code": 1707,
     "message": "Cannot send request, agent is not active",
     "remediation": "Please, check non-active agents connection and try again. Visit
     https://documentation.wazuh.com/current/user-manual/registering/index.html and
     https://documentation.wazuh.com/current/user-manual/agents/agent-connection.html
     to obtain more information on registering and connecting agents"
    },
    "id": [
     "001",
     "002",
     "009",
     "010"
    ]
   },
  ]
 },
 "message": "Restart command was not sent to any agent",
 "error": 1
}
```


Kısmi yanıt örneği (HTTP durum kodu 200):

Output


```
{
 "data": {
  "affected_items": [
```

```
    {
      "ip": "10.0.0.9",
      "id": "001",
      "name": "Carlos",
      "dateAdd": "2020-10-07T08:14:32Z",
      "node_name": "unknown",
      "registerIP": "10.0.0.9",
      "status": "never_connected"
    }
   ],
   "total_affected_items": 1,
   "total_failed_items": 1,
   "failed_items": [
    {
     "error": {
      "code": 1701,
      "message": "Agent does not exist",
      "remediation": "Please, use `GET /agents?select=id,name` to find all available agents"
     },
     "id": [
      "005"
     ]
    }
   ]
  },
  "message": "Some agents information was not returned",
  "error": 2
}
```


Yetkisiz bir isteği bildirmek için örnek yanıt (HTTP durum kodu 401):

Output


```
{
  "title": "Unauthorized",
  "detail": "The server could not verify that you are authorized to access the URL requested. You either supplied th
}
```


İzin reddedildi hatasını (HTTP durum kodu 403) bildirmek için örnek yanıt:

Output


```
{
  "title": "Permission Denied",
  "detail": "Permission denied: Resource type: *:*",
  "remediation": "Please, make sure you have permissions to execute the current request. For more information on
  "error": 4000,
  "dapi_errors": {
    "unknown-node": {
      "error": "Permission denied: Resource type: *:*"
    }
  }
}
```

Wazuh Sunucu API Kullanımına İlişkin Pratik Örnekler

Bu bölümde, cURL, Python betikleri ve PowerShell betikleri kullanarak Wazuh sunucu API'sine çeşitli istek türlerinin nasıl gönderileceğini gösteriyoruz. Bu örnekler, öngörebileceğiniz daha gelişmiş kullanım durumları için temel bilgi görevi görür.

CURL

cURL, HTTP/HTTPS istekleri ve komutları göndermek için bir komut satırı aracıdır. Birçok Linux ve macOS uç noktasına önceden yüklenmiş olarak gelir ve kullanıcıların Wazuh sunucu API'siyle doğrudan komut satırından etkileşim kurmasını sağlar. Herhangi bir uç noktayı yürütmeden önce bir JWT edinmeniz gerektiğini unutmayın. Aşağıdaki örneklerde, belirteci almak ve onu bir ortam değişkeni (`$TOKEN`) olarak kaydetmek için ham seçeneğini kullanıyoruz. JWT edinmeyle ilgili ayrıntılı talimatlar için lütfen başlarken bölümüne bakın.

GET

Aşağıdaki GET isteği, Wazuh sunucu API'si hakkında başlık, sürüm, revizyon, lisans, ana bilgisayar adı ve geçerli zaman damgası gibi temel bilgileri alır:

```
# curl -k -X GET "https://localhost:55000/" -H  "Authorization: Bearer $TOKEN"
```

Output


```
{
   "data": {
      "title": "Wazuh API",
      "api_version": "4.7.4",
      "revision": 40717,
      "license_name": "GPL 2.0",
      "license_url": "https://github.com/wazuh/wazuh/blob/master/LICENSE",
      "hostname": "wazuh-master",
      "timestamp": "2024-05-14T21:34:15Z"
   },
   "error": 0
}
```


POST

Wazuh sunucusu API'sine yapılan aşağıdaki POST isteği, istek gövdesinde kullanıcı adı `test_user`ve parola belirtilerek Wazuh sunucusunda yeni bir kullanıcı oluşturur .`Test_user1`

```
curl -k -X POST "https://localhost:55000/security/users" -H  "Authorization: Bearer $TOKEN" -H  "Content-Type: application/json" -d "{\"username\":\"test_user\",\"password\":\"Test_user1\"}"
```

Output


```
{
 "data": {
  "affected_items": [
   {
    "username": "test_user",
    "roles": []
   }
  ],
  "total_affected_items": 1,
  "total_failed_items": 0,
  "failed_items": []
 },
 "message": "User was successfully created",
 "error": 0
}
```


DELETE

Wazuh sunucusu API'sine gönderilen aşağıdaki DELETE isteği, Wazuh sunucusundaki tüm aracı gruplarını siler.

```
curl -k -X DELETE "https://localhost:55000/groups?pretty=true&groups_list=all" -H  "Authorization: Bearer $TOKEN"
```

Output


```
{
 "data": {
  "affected_items": [
   "group1",
   "group2",
   "group3"
  ],
  "total_affected_items": 3,
  "total_failed_items": 0,
  "failed_items": [],
  "affected_agents": [
   "001",
   "002",
   "003",
   "005",
   "006",
   "007",
   "008",
   "009",
   "010"
```

```
    ]
  },
  "message": "All selected groups were deleted",
  "error": 0
}
```

Python

Bağlantısı kesilen aracılar hakkında, son canlı tutma süreleri ve kimlikleri dahil olmak üzere bilgi almak için bir Python betiği kullanabilirsiniz. Bunu yapmak için, betik önce bir taşıyıcı belirteci almak için temel kimlik doğrulamasını kullanarak Wazuh sunucu API'siyle kimlik doğrulaması yapar, ardından gerekli bilgileri almak için bir GET isteği yapar.

Aşağıdaki Python betiğini şu şekilde kaydedin `get_agent_keep_alive.py`:

```
#!/usr/bin/env python3

import json
from base64 import b64encode

import requests  # To install requests, use: pip install requests
import urllib3

# Configuration
endpoint = '/agents?select=lastKeepAlive&select=id&status=disconnected'

protocol = 'https'
host = '<WAZUH_SERVER_API_IP>'
port = '<WAZUH_SERVER_API_PORT>'
user = '<WAZUH_API_USER>'
password = '<WAZUH_API_PASSWORD>'

# Disable insecure https warnings (for self-signed SSL certificates)
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

# Functions
def get_response(request_method, url, headers, verify=False, body=None):
    """Get API result"""
    if body is None:
        body = {}

    request_result = getattr(requests, request_method.lower())(url, headers=headers, verify=verify, data=body)

    if request_result.status_code == 200:
        return json.loads(request_result.content.decode())
    else:
        raise Exception(f"Error obtaining response: {request_result.json()}")

# Variables
base_url = f"{protocol}://{host}:{port}"
login_url = f"{base_url}/security/user/authenticate"
```

```
basic_auth = f"{user}:{password}".encode()
headers = {
        'Authorization': f'Basic {b64encode(basic_auth).decode()}',
        'Content-Type': 'application/json'
        }
headers['Authorization'] = f'Bearer {get_response("POST", login_url, headers)["data"]["token"]}'

# Request
response = get_response("GET", url=base_url + endpoint, headers=headers)

# WORK WITH THE RESPONSE AS YOU LIKE
print(json.dumps(response, indent=4, sort_keys=True))
```

Aşağıdaki değişkenleri değiştirin:

- `<WAZUH_SERVER_API_IP>`Wazuh sunucunuzun IP adresi ile.
- `<WAZUH_SERVER_API_PORT>`Wazuh sunucusu API port numarasıyla (varsayılan olarak port 5500).
- `<WAZUH_API_USER>`ve `<WAZUH_API_PASSWORD>`doğru belgelerle.

Python `requests`modülünü kurun:

```
python3 -m pip install requests
```

> Not: Python modülü `urllib3`sürüm 2.0 ve üzeri yalnızca OpenSSL sürüm 1.1.1 veya üzerini destekler. Sisteminizde daha eski bir OpenSSL sürümü varsa, şunlardan birini yapmanız gerekir:
> - OpenSSL'i 1.1.1 veya daha üst bir sürüme yükseltin.
> - `urllib3`Mevcut OpenSSL sürümünüzle uyumlu bir sürüme geçin.
> Uyumluluk sorunlarını önlemek için lütfen yazılım bağımlılıklarınızın düzgün bir şekilde hizalandığından emin olun.

Bağlantısı kesilen aracılar hakkında bilgi almak için Python betiğini çalıştırın:

```
python3 get_agent_keep_alive.py
```

Output


```
{
   "data": {
      "affected_items": [
         {
            "id": "009",
            "lastKeepAlive": "2020-05-23T12:39:50Z"
         },
         {
            "id": "010",
            "lastKeepAlive": "2020-05-23T12:39:50Z"
         }
```

```
        ],
        "failed_items": [],
        "total_affected_items": 2,
        "total_failed_items": 0
    },
    "message": "All selected agents information was returned",
    "error": 0
}
```


Güç Kabuğu

Bağlantısı kesilen aracıların son canlı tutma süreleri ve kimlikleri dahil olmak üzere ayrıntıları almak için bir PowerShell betiği de kullanabilirsiniz. Bunu yapmak için, betik önce bir taşıyıcı belirteci almak için temel kimlik doğrulamasını kullanarak Wazuh sunucu API'siyle kimlik doğrulaması yapar, ardından gerekli bilgileri almak için bir GET isteği yapar.

Aşağıdaki PowerShell betiğini şu şekilde kaydedin `get_agent_keep_alive.ps1`:

```
function Ignore-SelfSignedCerts {
    add-type @"
        using System.Net;
        using System.Security.Cryptography.X509Certificates;

        public class PolicyCert : ICertificatePolicy {
            public PolicyCert() {}
            public bool CheckValidationResult(
                ServicePoint sPoint, X509Certificate cert,
                WebRequest wRequest, int certProb) {
                return true;
            }
        }
"@
    [System.Net.ServicePointManager]::CertificatePolicy = new-object PolicyCert
}

# Configuration
$endpoint = "/agents?select=lastKeepAlive&select=id&status=disconnected"
$method = "get"

$protocol = "https"
$host_name = "<WAZUH_SERVER_API_IP>"
$port = "<WAZUH_SERVER_API_PORT>"
$username = "<WAZUH_API_USER>"
$password = "<WAZUH_API_PASSWORD>"

# Variables
$base_url = $protocol + "://" + $host_name + ":" + $port
$login_url = $base_url + "/security/user/authenticate"
$endpoint_url = $base_url + $endpoint
$base64AuthInfo = [Convert]::ToBase64String([Text.Encoding]::ASCII.GetBytes(("{0}:{1}" -f $username, $passwo
$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
```

```
$headers.Add("Content-Type", 'application/json')
$headers.Add("Authorization", "Basic " + $base64AuthInfo)

Ignore-SelfSignedCerts
$token_response = Invoke-RestMethod -Uri $login_url -Headers $headers
$headers["Authorization"] = "Bearer " + $token_response.data.token

# Request
try{
   $response = Invoke-RestMethod -Method $method -Uri $endpoint_url -Headers $headers
}catch{
   $response = $_.Exception.Response
}

# WORK WITH THE RESPONSE AS YOU LIKE
Write-Output $response.data
```

Aşağıdaki değişkenleri değiştirin:

- `<WAZUH_SERVER_API_IP>`Wazuh sunucunuzun IP adresi ile.
- `<WAZUH_SERVER_API_PORT>`Wazuh sunucusu API port numarasıyla (varsayılan olarak port 5500).
- `<WAZUH_API_USER>`ve `<WAZUH_API_PASSWORD>`doğru belgelerle.

Bağlantısı kesilen aracılar hakkında bilgi almak için Windows uç noktasında PowerShell betiğini çalıştırın:

```
powershell .\get_agent_keep_alive.py
```

Output

```
affected_items                                  total_affected_items total_failed_items failed_items
--------------                                  -------------------- ------------------ ------------
{@{lastKeepAlive=2020-05-23T12:39:50Z; id=009}, @{lastKeepAlive=2020-05-23T12:39:50Z; id=010}}  2                    0                  {}
```

Yapılandırma

> Not: Wazuh sunucu API'sini nasıl koruyacağınız hakkında daha fazla bilgi için lütfen Wazuh sunucu API'sini güvence altına alma bölümünü inceleyin.

Wazuh Sunucu API Yapılandırma Dosyası

`/var/ossec/api/configuration/api.yaml`Wazuh sunucusu API yapılandırması Wazuh sunucusundaki dosyada bulunur . Varsayılan olarak, tüm ayarlar yorum satırına alınır. Farklı bir yapılandırma uygulamak için, yorum satırını kaldırın ve istediğiniz satırı düzenleyin.

Yapılandırma dosyası için tüm kullanılabilir ayarlar şunlardır `/var/ossec/api/configuration/api.yaml`. Her bir ayar hakkında daha fazla bilgi için yapılandırma seçeneklerini kontrol edin:

```
host: ['0.0.0.0', '::']
port: 55000

drop_privileges: yes
experimental_features: no
max_upload_size: 10485760

intervals:
  request_timeout: 10

https:
  enabled: yes
  key: "server.key"
  cert: "server.crt"
  use_ca: False
  ca: "ca.crt"
  ssl_protocol: "auto"
  ssl_ciphers: ""

logs:
  level: "info"
  format: "plain"
  max_size:
   enabled: false

cors:
  enabled: no
  source_route: "*"
```

```
  expose_headers: "*"
  allow_headers: "*"
  allow_credentials: no

access:
  max_login_attempts: 50
  block_time: 300
  max_request_per_minute: 300

upload_configuration:
  remote_commands:
    localfile:
      allow: yes
      exceptions: []
    wodle_command:
      allow: yes
      exceptions: []
  limits:
    eps:
      allow: yes
   agents:
    allow_higher_versions:
      allow: yes
   indexer:
    allow: yes
   integrations:
    virustotal:
      public_key:
        allow: yes
        minimum_quota: 240
```

> Uyarı: Bir Wazuh sunucu kümesi çalıştırıldığında, ana düğüm yerel Wazuh sunucu API yapılandırma dosyasını otomatik olarak çalışan düğümlere göndermez. Her düğüm kendi Wazuh sunucu API yapılandırmasını korur. Bu nedenle, ana düğümdeki yapılandırma dosyasında herhangi bir değişiklik yapılırsa, tutarlılığı sağlamak için her çalışan düğümünde yapılandırmayı manuel olarak güncellemelisiniz. Her çalışanın yerel yapılandırmasında IP adresinin ve bağlantı noktasının üzerine yazılmadığından emin olun.

Yapılandırma dosyasını düzenledikten sonra Wazuh yönetici servisini kullanarak Wazuh sunucu API'sini yeniden başlattığınızdan emin olun:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

API Yapılandırma Seçenekleri

home

İzin verilen değerler	Varsayılan değer	Tanım
Geçerli IP adresleri veya ana bilgisayar adlarının listesi	['0.0.0.0', '::']	Wazuh sunucu API'sinin çalıştığı Wazuh yöneticisinin IP adresleri veya ana bilgisayar adları.

port

İzin verilen değerler	Varsayılan değer	Tanım
1 ile 65535 arasındaki herhangi bir değer	55000	Wazuh sunucu API'sinin dinleyeceği port.

use_only_authd

> 4.3.0 sürümünden itibaren kullanımdan kaldırılmıştır.

İzin verilen değerler	Varsayılan değer	Tanım
evet, doğru, hayır, yanlış	YANLIŞ	Ajanları kaydederken ve kaldırırken wazuh-authd kullanımını zorunlu kılın.

drop_privileges

İzin verilen değerler	Varsayılan değer	Tanım
evet, doğru, hayır, yanlış	doğru	Wazuh-api işlemini kullanıcı olarak çalıştırın `wazuh`.

experimental_features

İzin verilen değerler	Varsayılan değer	Tanım
evet, doğru, hayır, yanlış	YANLIŞ	Geliştirme aşamasındaki özellikleri etkinleştirin

max_upload_size

İzin verilen değerler	Varsayılan değer	Tanım
Herhangi bir pozitif tam sayı	10485760	API'nin kabul edebileceği maksimum gövde boyutunu bayt cinsinden ayarlayın (0 -> sınırsız)

intervals (aralıklar)

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım
request_timeout		10	Her API isteği için maksimum yanıt süresini (saniye cinsinden) ayarlayın

https

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım
enabled	evet, doğru, hayır, yanlış	doğru	Wazuh sunucu API'sinde SSL'yi (https) etkinleştirin veya devre dışı bırakın.
key	Herhangi bir metin dizesi	sunucu.anahtar	Özel anahtarın adı. İçinde saklanır `/var/ossec/api/configuration/ssl`.
sertifika	Herhangi bir metin dizesi	sunucu.crt	Sertifikanın adı. Şurada saklanır `/var/ossec/api/configuration/ssl`: .
use_ca	evet, doğru, hayır, yanlış	YANLIŞ	Bir Sertifika Yetkilisinden alınan sertifikanın kullanılıp kullanılmayacağı.
ca	Herhangi bir metin dizesi	yaklaşık.krt	Sertifika Yetkilisinin (CA) sertifikasının adı. İçinde saklanır `/var/ossec/api/configuration/ssl`.
ssl_protocol	TLS, TLSv1, TLSv1.1, TLSv1.2, otomatik	4.8.0 sürümündeki yenilikler. otomatik	SSL protokolüne izin vermek için. Değeri büyük/küçük harfe duyarlı değildir.
ssl_ciphers	Herhangi bir metin dizesi	Hiçbiri	SSL şifrelerine izin verilir. Değeri büyük/küçük harfe duyarlı değildir.

logs

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım

level	devre dışı, bilgi, uyarı, hata, hata ayıklama, debug2 (her seviye bir önceki seviyeyi içerir)	bilgi	Wazuh sunucusu API günlüklerinin ayrıntı düzeyini ayarlayın.
path	Herhangi bir metin dizesi.	günlükler/api.log	4.3.0 sürümünden itibaren kullanımdan kaldırılmıştır. Wazuh sunucusu API kayıtlarının kaydedileceği yol.
format		ova	4.4.0 sürümündeki yenilikler. Wazuh sunucusu API günlüklerinin biçimini ayarlayın.

max_size

> 4.6.0 sürümündeki yenilikler.

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım
enabled	evet, doğru, hayır, yanlış	YANLIŞ	Zaman tabanlı ve boyut tabanlı Wazuh API günlük döndürme arasında geçiş yapın. Bu seçeneği etkinleştirmek zaman tabanlı döndürmeyi devre dışı bırakır ve bunun yerine dosya boyutuna dayalı döndürmeyi etkinleştirir.
size	Geçerli bir birimden sonra gelen herhangi bir pozitif sayı. Kilobayt için K/k, megabayt için M/m.	1M	Boyut tabanlı günlük döndürmeyi tetiklemeyecek şekilde maksimum dosya boyutunu ayarlayın. 1 M'den düşük değerler 1 M olarak kabul edilir.

cors

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım
enabled		YANLIŞ	Wazuh sunucu API'sinde CORS kullanımını etkinleştirin veya devre dışı bırakın.
source_route	Herhangi bir metin dizesi	`*`	Kaynakların mevcut olacağı kaynaklar. Örneğin `http://client.example.org`.
expose_headers	Herhangi bir metin dizesi	`*`	Hangi başlıkların yanıtın bir parçası olarak açığa çıkarılabileceği.
allow_headers		`*`	Gerçek istek sırasında hangi HTTP başlıklarının kullanılabileceği.
allow_credentials	evet, doğru, hayır, yanlış	YANLIŞ	Tarayıcılara yanıtın ön uç JavaScript'e açılıp açılmayacağını söyleyin.

access (erişim)

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım
max_login_attempts	Herhangi bir pozitif tam sayı	50	Belirtilen saniye sayısı içerisinde yapılabilecek maksimum oturum açma girişimi sayısını ayarlayın `block_time`.
block_time		300	Oturum açma isteklerini denemek için belirlenen zaman aralığı (saniye cinsinden). Belirlenen istek sayısı (`max_login_attempts`) bu zaman sınırı içinde aşılırsa, IP adresi blok zaman aralığının sonuna kadar engellenir.
max_request_per_minute	Herhangi bir pozitif tam sayı	300	Dakikada izin verilen maksimum istek sayısı. Kimlik doğrulama istekleri hariç tüm Wazuh sunucu API uç noktaları için geçerlidir. Bu sınıra bir dakikadan kısa sürede ulaşılması, kalan süre boyunca herhangi bir kullanıcıdan gelen tüm istekleri engeller. Bir değeri `0`bu özelliği devre dışı bırakır. İstekler için, etkili değer 30'dan büyük değerler içindir.`POST /events30`

upload_configuration

> 4.4.0 sürümündeki yenilikler.

remote_commands (yerel_dosya ve wodle "komut")

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım
allow	evet, doğru, hayır, yanlış	doğru	Wazuh sunucu API'si aracılığıyla uzaktan komutlarla yapılandırmaların yüklenmesine izin verin. Bu seçeneğin ayarlanması, wodle "command" seçeneğini veya localfile etiketi içindeki seçeneği içeren dosyaların `false` yüklenmesini engeller .`ossec.conf<command>`
exceptions	komut listesi	[ ]	API aracılığıyla yüklenmesine izin verilen komutların bir listesini ayarlayın. Bu istisnalar yapılandırmadan bağımsız olarak her zaman yüklenebilir `allow`.

sınırlar

eps

> 4.4.0 sürümündeki yenilikler.

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım
allow	evet, doğru, hayır, yanlış	doğru	Wazuh sunucu API'si aracılığıyla değiştirilmiş EPS limitleriyle yapılandırmaların yüklenmesine izin verin. Bu seçeneğin ayarlanması, genel etiketin içindeki bölüm değiştiyse dosyaların `false` yüklenmesini engeller. `ossec.conf<limits><eps>`

agents

allow_higher_versions

> 4.6.0 sürümündeki yenilikler.

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım
allow	evet, doğru, hayır, yanlış	doğru	Wazuh sunucu API'si aracılığıyla daha yüksek aracı sürümlerini kabul eden yapılandırmaların yüklenmesine izin verin. Bu seçeneğin ayarlanması, auth veya remote etiketleri içindeki değere sahip bölümü içeren dosyaların `false`yüklenmesini engeller .`ossec.conf` `<allow_higher_versions>yes`

indexer

> 4.8.0 sürümündeki yenilikler.

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım
allow	evet, doğru, hayır, yanlış	doğru	Wazuh sunucu API'si aracılığıyla güncellenmiş bir dizinleyici yapılandırma bölümünün yüklenmesine izin verir . Bu seçeneğin ayarlanması `false`, yükleme sırasında dizinleyici yapılandırmasının güncellenmesini önler `ossec.conf`.

entegrasyonlar

> 4.8.0 sürümündeki yenilikler.

virüstotal (public_key)

Alt alanlar	İzin verilen değerler	Varsayılan değer	Tanım
allow	evet, doğru, hayır, yanlış	doğru	Wazuh sunucu API'si aracılığıyla genel bir API anahtarı kullanılarak güncellenmiş bir Virus Total entegrasyon yapılandırma bölümünün yüklenmesine izin verir . Bu seçeneğin ayarlanması `false`, yükleme sırasında entegrasyonların Virus Total yapılandırmasının güncellenmesini önler `ossec.conf`.
minimum_quota	Herhangi bir pozitif tam sayı	240	Virus Total genel API anahtarı için minimum kota değeri.

Wazuh Sunucu API Güvenlik Yapılandırması

`auth_token_exp_timeout`Güvenlik yapılandırmasını ve `rbac_mode`ayarlarını yalnızca Wazuh sunucu API uç noktaları aracılığıyla sorgulayabilir ve değiştirebilirsiniz : GET /security/config , PUT /security/config ve DELETE /security/config . `auth_token_exp_timeout`Bir kimlik doğrulama belirtecinin süresi dolmadan ve yenilenmesi gerekmeden önceki saniye cinsinden süreyi tanımlar. `rbac_mode`Kullanıcı rollerine ve izinlerine göre kaynaklara ve uç noktalara erişimi genel olarak izin vermek veya kısıtlamak üzere yapılandırılabilen Rol Tabanlı Erişim Kontrol sisteminin genel davranışını belirler. Daha fazla ayrıntı için Rol Tabanlı Erişim Kontrol belgelerine bakın. Yapılandırma, geçerliyse bir kümedeki her Wazuh sunucu API'sine uygulanır.

Her bir ayar hakkında daha fazla bilgi için lütfen güvenlik yapılandırma seçeneklerini kontrol edin .

```
auth_token_exp_timeout: 900
rbac_mode: white
```

> Uyarı: Güvenlik nedenleriyle, güvenlik yapılandırmasını değiştirmek tüm JWT'leri iptal eder. Değişiklikten sonra oturum açmanız ve yeni bir token edinmeniz gerekecektir.

Güvenlik Yapılandırma Seçenekleri

auth_token_exp_timeout

İzin verilen değerler	Varsayılan değer	Tanım
Herhangi bir pozitif tam sayı	900	JWT token'larının süresinin dolmasının kaç saniye süreceğini ayarlayın.

rbac_mode

İzin verilen değerler	Varsayılan değer	Tanım
siyah, beyaz	beyaz	RBAC davranışını ayarlayın. Varsayılan olarak, siyah modda her şeye izin verilirken beyaz modda her şey reddedilir. İstenen RBAC altyapısına daha uygun olan rbac_mode'u seçin. Siyah modda, sadece bazı politikalarla birkaç belirli eylem-kaynak çiftini reddetmek çok kolaydır, beyaz mod ise daha güvenlidir ve sıfırdan oluşturulmasını gerektirir.

Yapılandırma Endpoints

Wazuh sunucu API'sinin geçerli yapılandırmasını sorgulamaya izin veren birkaç uç noktası vardır.

Genel API yapılandırmasını değiştirmek için dosyayı Wazuh sunucu API yapılandırma dosyası `/var/ossec/api/configuration/api.yaml` bölümünde ayrıntılı olarak açıklandığı şekilde düzenleyin .

Yapılandırmayı Al

- GET /manager/api/config : Yerel Wazuh sunucusunun API yapılandırmasının tamamını alın.
- GET /cluster/api/config : Tüm küme düğümlerinin (veya bir listesinin) Wazuh sunucusu API yapılandırmasının tamamını alın.
- GET /security/config : Mevcut güvenlik yapılandırmasını alın.

Yapılandırmayı Değiştir

- PUT /security/config : Güvenlik yapılandırmasını değiştirin.

Yapılandırmayı Geri Yükle

- DELETE /security/config : Varsayılan güvenlik yapılandırmasını geri yükler.

SSL sertifikası

> Not: Bu işlem Wazuh sunucu API'si ilk kez çalıştırıldığında otomatik olarak gerçekleştirilir.

SSL sertifikası, Wazuh sunucu API'si ile istemcileri arasında güvenli iletişimi sağlar. Sertifika dosyaları dizinde saklanır `/var/ossec/api/configuration/ssl/`.

Wazuh sunucu API'si için yeni sertifikalar oluşturmak üzere aşağıdaki adımları izleyin:

1. Anahtar ve sertifika isteğini oluşturun (`openssl`paket gereklidir):

   ```
   cd /var/ossec/api/configuration/ssl
   openssl req -newkey rsa:2048 -new -nodes -x509 -days 365 -keyout server.key -out server.crt
   ```

   Varsayılan olarak, anahtarın parolası sunucu her çalıştırıldığında girilmelidir. Anahtar Wazuh sunucu API'si veya yukarıdaki komut tarafından üretilmişse, parolası olmazdı.
2. (İsteğe bağlı) Anahtarı bir parola ile güvenceye alın:

   ```
   ssh-keygen -p -f server.key
   ```

   Yeni şifreyi girmeniz ve onaylamanız istenecektir.

Wazuh Sunucu API'sini Güvence Altına Alma

Wazuh panosu ile Wazuh sunucu API'si arasındaki iletişim varsayılan olarak HTTPS ile şifrelenir. Wazuh sunucu API'si, kullanıcılar bunları sağlamazsa ilk çalıştırma sırasında kendi özel anahtarını ve sertifikasını oluşturur. Ek olarak, Wazuh sunucu API'si OVA kurulumuyla birlikte kurulduğunda aşağıdaki kullanıcı adı-şifre çiftini otomatik olarak oluşturur:

- `wazuh:wazuh`
- `wazuh-wui:wazuh-wui`

Wazuh dağıtımı kurulum yardımcısı betiği kullanılarak gerçekleştirildiyse, Wazuh API kullanıcı adı şudur `wazuh`ve aşağıdaki komutu çalıştırarak parolayı çıkarabilirsiniz:

```
tar -axf wazuh-install-files.tar wazuh-install-files/wazuh-passwords.txt -O | grep -P "\'wazuh\'" -A 1
```

Bu nedenle Wazuh yöneticisini kurduktan sonra Wazuh sunucu API'sinin güvenliğini sağlamak büyük önem taşımaktadır.

> Uyarı: Wazuh sunucu API'si tarafından oluşturulan sertifikanın kendi imzalı olması nedeniyle varsayılan şifreleri değiştirmenizi ve kendi sertifikanızı kullanmanızı şiddetle öneririz.

Wazuh Sunucu API'sini Güvence Altına Almak İçin Önerilen Değişiklikler

1. HTTPS Parametrelerini Değiştirin

Wazuh sunucu API'si varsayılan olarak HTTPS'yi etkinleştirmiştir. Eğer içinde kullanılabilir bir sertifika yoksa `/var/ossec/api/configuration/ssl/`, Wazuh sunucusu başlatıldığında özel anahtarı ve kendi kendine imzalanmış bir sertifikayı üretecektir. Eğer durum buysa ve API günlük biçimi olarak ayarlanmışsa `plain`, aşağıdaki satırlar görünecektir `/var/ossec/logs/api.log`:

```
INFO: HTTPS is enabled but cannot find the private key and/or certificate. Attempting to generate them.
INFO: Generated private key file in WAZUH_PATH/api/configuration/ssl/server.key.
```

```
INFO: Generated certificate file in WAZUH_PATH/api/configuration/ssl/server.crt.
```

Bu HTTPS seçeneklerini, durumlarını veya sertifika yolunu da içerecek şekilde, şu adreste bulunan Wazuh sunucu API yapılandırma dosyasını düzenleyerek değiştirebilirsiniz `/var/ossec/api/configuration/api.yaml`:

```
https:
  enabled: yes
  key: "server.key"
  cert: "server.crt"
  use_ca: False
  ca: "ca.crt"
  ssl_protocol: "auto"
  ssl_ciphers: ""
```

Değişiklikleri uygulamak için Wazuh yönetici hizmetini kullanarak Wazuh sunucu API'sini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

2. Yönetici Kullanıcıları İçin Varsayılan Parolayı Değiştirin

Yönetici kullanıcıları için varsayılan şifreyi değiştirmek için `wazuh`aşağıdaki `wazuh-wui`Wazuh sunucu API isteğini kullanabilirsiniz: PUT /security/users/{user_id} .

> Not: Kullanıcıların şifresi 8 ile 64 karakter arasında olmalıdır. En az bir büyük harf, küçük harf, rakam ve sembol içermelidir.

Aşağıda curl kullanarak şifre değiştirmenin bir örneğini gösteriyoruz :

1. Kullanıcıların kullanıcı kimlikleriyle birlikte bir listesini alın:

```
curl -k -X GET "https://localhost:55000/security/users?pretty=true" -H  "Authorization: Bearer $TOKEN"
```

2. İstenilen kullanıcının şifresini değiştirin:

```
curl -k -X PUT "https://localhost:55000/security/users/<USER_ID>" -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json" -d '{"password": "<NEW_PASSWORD>"}'
```

`<USER_ID>`Kullanıcının ID'si ve `<NEW_PASSWORD>`yeni şifre ile değiştirin.

> Uyarı: `wazuh-wui` kullanıcı parolasını değiştirmek Wazuh panosunu etkileyecektir. Yeni kimlik bilgileriyle `/usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml` yapılandırma dosyasını uygun şekilde güncellemeniz gerekecektir. Daha fazla bilgi edinmek için Wazuh gösterge tablosu yapılandırma dosyası belgesine bakın.

3. Varsayılan Host ve Portu Değiştirin

Varsayılan olarak, `host`olarak ayarlanır ve Wazuh sunucu API'sinin tüm kullanılabilir ağ arayüzlerinde gelen bağlantıları kabul etmesine olanak tanır. Erişimi kısıtlamak için, Wazuh sunucu API yapılandırmasını şurada düzenleyin :`['0.0.0.0', '::']`/var/ossec/api/configuration/api.yaml

```
host: ['0.0.0.0', '::']
```

Varsayılan portu da değiştirebilirsiniz:

```
port: 55000
```

Bu parametreleri yapılandırdıktan sonra, Systemd veya SysV init ile Wazuh yönetici servisini kullanarak Wazuh sunucu API'sini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

4. Dakika Başına Maksimum İstek Sayısını Ayarlayın

Wazuh sunucu API'sinin aşırı yüklenmesini önlemek için, API'nin dakikada işleyebileceği maksimum istek sayısını belirlemek için hız sınırlaması uygulayabilirsiniz. Bu sınır aşılırsa, API geri kalan süre boyunca herhangi bir kullanıcıdan gelen diğer istekleri reddeder.

`max_request_per_minute`Varsayılan sınır dakikada 300 istektir. Bunu, içindeki ayarı değiştirerek ayarlayın `/var/ossec/api/configuration/api.yaml`.

> Not: Hız sınırlamasını devre dışı bırakmak için değerini 0 olarak ayarlayın.

5. Maksimum Oturum Açma Girişimi Sayısını Ayarlayın

Kaba kuvvet saldırılarına karşı korunmak için, belirli bir zaman dilimi içinde aynı IP adresinden gelen oturum açma girişimlerini sınırlayabilirsiniz. Bu sınırın aşılması, IP adresini o süre boyunca engeller.

Varsayılan olarak, 300 saniyelik periyotta 50 oturum açma girişimine izin verilir. Bu sınırları ayarlamak için `max_login_attempts` ve/veya `block_time` ayarlarını `/var/ossec/api/configuration/api.yaml`'da düzenleyin.

Rol Tabanlı Erişim Kontrolü

Wazuh Sorgu Dili (WQL) Kullanılarak Verilerin Filtrelenmesi

Wazuh API'sinin sorgularını kullanarak gelişmiş filtreleme mümkündür. Sorgular `q`parametre kullanılarak belirtilir. Bir sorgunun yapısı şu şekildedir:

- **Alan adı** : Filtrelenecek alan adı. Yanlış bir alan adı kullanılırsa, bir hata oluşacaktır.
- **Operatör** : Filtreleme yapılacak operatör:
    - `=`: eşitlik.
    - `!=`: eşitlik değil.
    - `<`: daha küçük.
    - `>`: daha büyük.
    - `~`: gibi.
    - `()`: gruplama operatörleri.
- **Değer** : Filtrelenecek değer.
- **Ayırıcı** : Birden fazla "sorguyu" birleştirmek için kullanılan operatör:
    - `,`: bir . temsil eder `OR`.
    - `;`: bir . temsil eder `AND`.

> Not: Ayrılmış karakterlerin, özellikle noktalı virgüllerin (`;` → `%3B`) yüzde kodlu olması gerekir . İşlemi kolaylaştırmak için cURL içinde `--data-urlencode` kullanabilirsiniz.

Örnekler

Örneğin, 18'den yüksek sürüme sahip Ubuntu aracılarını filtrelemek için aşağıdaki sorgu kullanılır. q parametresinin değerinin şu şekilde kodlandığını unutmayın `--data-urlencode`:

```
curl -G --data-urlencode "q=os.name=ubuntu;os.version>18" -k -X GET
"https://localhost:55000/agents?limit=500&pretty=true&select=id,name,os.name,os.version,os.codename,os.m
ajor" -H  "Authorization: Bearer $TOKEN"
```

Output

```
{
  "data": {
    "affected_items": [
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.4 LTS"
        },
        "name": "wazuh-master",
        "id": "000"
      },
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.4 LTS"
        },
        "name": "wazuh-agent4",
        "id": "004"
      },
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.4 LTS"
        },
        "name": "wazuh-agent5",
        "id": "005"
      },
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.4 LTS"
        },
        "name": "wazuh-agent6",
        "id": "006"
      },
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.4 LTS"
        },
        "name": "wazuh-agent7",
        "id": "007"
```

```
      },
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.4 LTS"
        },
        "name": "wazuh-agent8",
        "id": "008"
      },
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.2 LTS"
        },
        "name": "wazuh-agent9",
        "id": "009"
      },
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.2 LTS"
        },
        "name": "wazuh-agent10",
        "id": "010"
      }
    ],
    "total_affected_items": 8,
    "total_failed_items": 0,
    "failed_items": []
  },
  "message": "All selected agents information was returned",
  "error": 0
}
```


Daha doğru bir sonuç elde etmek için aynı alan birden fazla kez kullanılabilir. Örneğin, Ubuntu 18'den daha yüksek ancak Ubuntu 18.04.4'ten daha düşük bir sürüme sahip filtreleme ajanları:

```
curl -G --data-urlencode "q=os.name=ubuntu;os.version>18;os.version<18.04.4" -k -X GET
"https://localhost:55000/agents?limit=500&pretty=true&select=id,name,os.name,os.version,os.codename,os.m
ajor" -H  "Authorization: Bearer $TOKEN"
```

Output


```
{
  "data": {
    "affected_items": [
```

```
        {
          "os": {
            "codename": "Bionic Beaver",
            "major": "18",
            "name": "Ubuntu",
            "version": "18.04.2 LTS"
          },
          "name": "wazuh-agent9",
          "id": "009"
        },
        {
          "os": {
            "codename": "Bionic Beaver",
            "major": "18",
            "name": "Ubuntu",
            "version": "18.04.2 LTS"
          },
          "name": "wazuh-agent10",
          "id": "010"
        }
      ],
      "total_affected_items": 2,
      "total_failed_items": 0,
      "failed_items": []
    },
    "message": "All selected agents information was returned",
    "error": 0
}
```


OR (,) operatörü ve LIKE AS (~) operatörünün kullanımına bir örnek, işletim sistemi adı *windows* veya *centos*içeren ajanları filtrelemek olabilir .

```
curl -G --data-urlencode "q=os.name~centos,os.name~windows" -k -X GET
"https://localhost:55000/agents?limit=500&pretty=true&select=id,name,os.name,os.version,os.codename,os.major" -H  "Authorization: Bearer $TOKEN"
```

Output


```
{
  "data": {
    "affected_items": [
      {
        "os": {
          "major": "6",
          "name": "Microsoft Windows 7 Ultimate Edition Professional Service Pack 1",
          "version": "6.1.7601"
        },
        "name": "jmv74211-PC",
        "id": "013"
      }
    ],
    "total_affected_items": 1,
```

```
    "total_failed_items": 0,
    "failed_items": []
  },
  "message": "All selected agents information was returned",
  "error": 0
}
```


Kimliği 0'dan farklı ve 4'ten küçük olan, adı alt dizeyi içeren `waz`ve ana sürümü 16 veya 18 olan Ubuntu ajanlarını elde etmek, aynı anda birden fazla operatörü içeren bir örnektir:

```
curl -G --data-urlencode "q=id!=0;id<4;name~waz;(os.major=16,os.major=18)" -k -X GET
"https://localhost:55000/agents?limit=500&pretty=true&select=id,name,os.name,os.version,os.codename,os.m
ajor" -H  "Authorization: Bearer $TOKEN"
```

Output


```
{
  "data": {
    "affected_items": [
      {
        "os": {
          "codename": "Xenial Xerus",
          "major": "16",
          "name": "Ubuntu",
          "version": "16.04.6 LTS"
        },
        "name": "wazuh-agent1",
        "id": "001"
      },
      {
        "os": {
          "codename": "Xenial Xerus",
          "major": "16",
          "name": "Ubuntu",
          "version": "16.04.6 LTS"
        },
        "name": "wazuh-agent2",
        "id": "002"
      },
      {
        "os": {
          "codename": "Xenial Xerus",
          "major": "16",
          "name": "Ubuntu",
          "version": "16.04.6 LTS"
        },
        "name": "wazuh-agent3",
        "id": "003"
      }
    ],
    "total_affected_items": 3,
    "total_failed_items": 0,
```

```
    "failed_items": []
  },
  "message": "All selected agents information was returned",
  "error": 0
}
```


Windows'ta `007` çalışan veya işletim sistemi ana sürümü 14 veya 18 olanlardan daha yüksek bir ID'ye sahip aracıları elde etmek :

```
curl -G --data-urlencode "q=id>007;(os.name~windows,(os.major=14,os.major=18))" -k -X GET
"https://localhost:55000/agents?limit=500&pretty=true&select=id,name,os.name,os.version,os.codename,os.m
ajor" -H  "Authorization: Bearer $TOKEN"
```

Output


```
{
  "data": {
    "affected_items": [
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.4 LTS"
        },
        "name": "wazuh-agent8",
        "id": "008"
      },
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.2 LTS"
        },
        "name": "wazuh-agent9",
        "id": "009"
      },
      {
        "os": {
          "codename": "Bionic Beaver",
          "major": "18",
          "name": "Ubuntu",
          "version": "18.04.2 LTS"
        },
        "name": "wazuh-agent10",
        "id": "010"
      },
      {
        "os": {
          "major": "6",
          "name": "Microsoft Windows 7 Ultimate Edition Professional Service Pack 1",
          "version": "6.1.7601"
```

```
        },
        "name": "jmv74211-PC",
        "id": "013"
      }
    ],
    "total_affected_items": 4,
    "total_failed_items": 0,
    "failed_items": []
  },
  "message": "All selected agents information was returned",
  "error": 0
}
```

Kullanım Örnekleri

Bu bölüm, Wazuh sunucu API'sinin bazı potansiyellerini göstermek için çeşitli kullanım örnekleri sunar. Tüm olası API istekleri hakkında ayrıntıları referans bölümünde bulabilirsiniz .

Kural Setini Keşfetmek

Genellikle bir uyarı ateşlendiğinde, kuralın kendisi hakkında ayrıntıları bilmek faydalıdır. Aşağıdaki istek kuralın niteliklerini sıralar `1002`:

```
curl -k -X GET "https://localhost:55000/rules?rule_ids=1002&pretty=true" -H  "Authorization: Bearer $TOKEN"#
curl -k -X GET "https://localhost:55000/rules?rule_ids=1002&pretty=true" -H  "Authorization: Bearer $TOKEN"
```

Output


```
{
  "data": {
    "affected_items": [
      {
        "filename": "0020-syslog_rules.xml",
        "relative_dirname": "ruleset/rules",
        "id": 1002,
        "level": 2,
        "status": "enabled",
        "details": {
          "match": {
            "pattern": "core_dumped|failure|error|attack| bad |illegal |denied|refused|unauthorized|fatal|failed|Segr
          }
        },
        "pci_dss": [],
        "gpg13": [
          "4.4"
        ],
        "gdpr": [],
        "hipaa": [],
        "nist_800_53": [],
        "groups": [
          "syslog",
          "errors"
        ],
        "description": "Unknown problem somewhere in the system."
      }
    ],
```

```
    "total_affected_items": 1,
    "total_failed_items": 0,
    "failed_items": []
  },
  "message": "All selected rules were returned",
  "error": 0
}
```


`web`Belirli bir ölçüte uyan hangi kuralların mevcut olduğunu bilmek de faydalı olabilir. Örneğin, PCI etiketiyle `10.6.1`ve kelimeyi içeren gruptaki tüm kuralları `failures`aşağıdaki komutla görüntüleyebilirsiniz :

```
curl -k -X GET
"https://localhost:55000/rules?pretty=true&limit=500&search=failures&group=web&pci_dss=10.6.1" -H
"Authorization: Bearer $TOKEN"
```

Output


```
{
 "data": {
  "affected_items": [
   {
    "filename": "0260-nginx_rules.xml",
    "relative_dirname": "ruleset/rules",
    "id": 31316,
    "level": 10,
    "status": "enabled",
    "details": {
     "frequency": "8",
     "timeframe": "240",
     "if_matched_sid": "31315",
     "same_source_ip": "",
     "mitre": "\n      "
    },
    "pci_dss": [
     "10.6.1",
     "10.2.4",
     "10.2.5",
     "11.4"
    ],
    "gpg13": [
     "7.1"
    ],
    "gdpr": [
     "IV_35.7.d",
     "IV_32.2"
    ],
    "hipaa": [
     "164.312.b"
    ],
    "nist_800_53": [
     "AU.6",
     "AU.14",
```

```
          "AC.7",
          "SI.4"
        ],
        "groups": [
          "authentication_failures",
          "tsc_CC7.2",
          "tsc_CC7.3",
          "tsc_CC6.1",
          "tsc_CC6.8",
          "nginx",
          "web"
        ],
        "description": "Nginx: Multiple web authentication failures."
      }
    ],
    "total_affected_items": 1,
    "total_failed_items": 0,
    "failed_items": []
  },
  "message": "All selected rules were returned",
  "error": 0
}
```


Test Kuralları ve Kod Çözücüler

Wazuh sunucu API'sini kullanarak bir wazuh-logtest oturumu başlatabilir veya özel veya varsayılan kuralları ve kod çözücüleri test etmek ve doğrulamak için mevcut bir oturumu kullanabilirsiniz. Aşağıdaki istek bir logtest oturumu oluşturur ve sağlanan günlük için eşleşen kuralları ve kod çözücüleri görüntüler. Ayrıca diğer bilgilerin yanı sıra ön kodlama aşamasını da ortaya çıkarır.

```
curl -k -X PUT "https://localhost:55000/logtest" -H  "Authorization: Bearer $TOKEN" -H  "Content-Type: application/json" -d "{\"event\":\"Jun 29 15:54:13 focal multipathd[557]: sdb: failed to get sysfs uid: No data available\",\"log_format\":\"syslog\",\"location\":\"user->/var/log/syslog\"}"
```

Output


```
{
  "error": 0,
  "data": {
    "token": "bc3ca27a",
    "messages": [
      "WARNING: (7309): 'null' is not a valid token",
      "INFO: (7202): Session initialized with token 'bc3ca27a'"
    ],
    "output": {
      "timestamp": "2020-10-15T09:40:53.630+0000",
      "rule": {
        "level": 0,
        "description": "FreeIPA messages grouped",
```

```
      "id": "82202",
      "firedtimes": 1,
      "mail": false,
      "groups": [
        "freeipa"
      ]
    },
    "agent": {
      "id": "000",
      "name": "wazuh-master"
    },
    "manager": {
      "name": "wazuh-master"
    },
    "id": "1602754853.1000774",
    "cluster": {
      "name": "wazuh",
      "node": "master-node"
    },
    "full_log": "Jun 29 15:54:13 focal multipathd[557]: sdb: failed to get sysfs uid: No data available",
    "predecoder": {
      "program_name": "multipathd",
      "timestamp": "Jun 29 15:54:13",
      "hostname": "focal"
    },
    "decoder": {
      "name": "freeipa"
    },
    "location": "user->/var/log/syslog"
  },
  "alert": false,
  "codemsg": 1
 }
}
```


Bir Wazuh Aracısının Dosya Bütünlüğü İzleme (FIM) Veritabanının Analiz Edilmesi

Wazuh FIM modülü tarafından izlenen tüm dosyalar hakkında bilgi görüntülemek için Wazuh sunucu API'sini kullanabilirsiniz. Aşağıdaki örnek, `.py` aracı kimliğine sahip izlenen bir uç noktaya yüklenen Python dosyalarıyla ilişkili tüm olayları gösterir `001`:

```
curl -k -X GET "https://localhost:55000/syscheck/001?pretty=true&search=.py" -H  "Authorization: Bearer $TOKEN"
```

Output

```
{
 "data": {
  "affected_items": [
   {
    "file": "/etc/python2.7/sitecustomize.py",
    "perm": "rw-r--r--",
    "sha1": "67b0a8ccf18bf5d2eb8c7f214b5a5d0d4a5e409d",
    "changes": 1,
    "md5": "d6b276695157bde06a56ba1b2bc53670",
    "inode": 29654607,
    "size": 155,
    "uid": "0",
    "gname": "root",
    "mtime": "2020-04-15T17:20:14Z",
    "sha256": "43d81125d92376b1a69d53a71126a041cc9a18d8080e92dea0a2ae23be138b1e",
    "date": "2020-05-25T14:28:41Z",
    "uname": "root",
    "type": "file",
    "gid": "0"
   },
   {
    "file": "/etc/python3.6/sitecustomize.py",
    "perm": "rw-r--r--",
    "sha1": "67b0a8ccf18bf5d2eb8c7f214b5a5d0d4a5e409d",
    "changes": 1,
    "md5": "d6b276695157bde06a56ba1b2bc53670",
    "inode": 29762235,
    "size": 155,
    "uid": "0",
    "gname": "root",
    "mtime": "2020-04-18T01:56:04Z",
    "sha256": "43d81125d92376b1a69d53a71126a041cc9a18d8080e92dea0a2ae23be138b1e",
    "date": "2020-05-25T14:28:41Z",
    "uname": "root",
    "type": "file",
    "gid": "0"
   }
  ],
  "total_affected_items": 2,
  "total_failed_items": 0,
  "failed_items": []
 },
 "message": "FIM findings of the agent were returned",
 "error": 0
}
```


Bir dosyayı SHA1 veya MD5 karma değerini kullanarak bulabilirsiniz. Aşağıdaki örneklerde, dosyayı hem SHA1 hem de MD5 karma değerini kullanarak alıyoruz:

```
curl -k -X GET
"https://localhost:55000/syscheck/001?pretty=true&hash=bc929cb047b79d5c16514f2c553e6b759abfb1b8" -H
```

```
"Authorization: Bearer $TOKEN"
```

Output


```
{
 "data": {
  "affected_items": [
   {
    "file": "/sbin/swapon",
    "perm": "rwxr-xr-x",
    "sha1": "bc929cb047b79d5c16514f2c553e6b759abfb1b8",
    "changes": 1,
    "md5": "085c1161d814a8863562694b3819f6a5",
    "inode": 14025822,
    "size": 47184,
    "uid": "0",
    "gname": "root",
    "mtime": "2020-01-08T18:31:23Z",
    "sha256": "f274025a1e4870301c5678568ab9519152f49d3cb907c01f7c71ff17b1a6e870",
    "date": "2020-05-25T14:29:44Z",
    "uname": "root",
    "type": "file",
    "gid": "0"
   }
  ],
  "total_affected_items": 1,
  "total_failed_items": 0,
  "failed_items": []
 },
 "message": "FIM findings of the agent were returned",
 "error": 0
}
```


```
curl -k -X GET
"https://localhost:55000/syscheck/001?pretty=true&hash=085c1161d814a8863562694b3819f6a5" -H
"Authorization: Bearer $TOKEN"
```

Output


```
{
 "data": {
  "affected_items": [
   {
    "file": "/sbin/swapon",
    "perm": "rwxr-xr-x",
    "sha1": "bc929cb047b79d5c16514f2c553e6b759abfb1b8",
    "changes": 1,
    "md5": "085c1161d814a8863562694b3819f6a5",
    "inode": 14025822,
    "size": 47184,
    "uid": "0",
    "gname": "root",
    "mtime": "2020-01-08T18:31:23Z",
```

```
      "sha256": "f274025a1e4870301c5678568ab9519152f49d3cb907c01f7c71ff17b1a6e870",
      "date": "2020-05-25T14:29:44Z",
      "uname": "root",
      "type": "file",
      "gid": "0"
     }
    ],
    "total_affected_items": 1,
    "total_failed_items": 0,
    "failed_items": []
  },
  "message": "FIM findings of the agent were returned",
  "error": 0
}
```


Yönetici Hakkında Bilgi Edinme

Wazuh sunucusu API'si aracılığıyla Wazuh yöneticisi hakkında çeşitli ayrıntıları alabilirsiniz. Bu ayrıntılar yapılandırma, durum, günlükler ve daha fazlasını içerir. Aşağıdaki örnek her Wazuh daemon'unun durumunun nasıl alınacağını gösterir:

```
curl -k -X GET "https://localhost:55000/manager/status?pretty=true" -H  "Authorization: Bearer $TOKEN"
```

Output


```
{
  "data": {
    "affected_items": [
      {
        "wazuh-agentlessd": "running",
        "wazuh-analysisd": "running",
        "wazuh-authd": "running",
        "wazuh-csyslogd": "running",
        "wazuh-dbd": "stopped",
        "wazuh-monitord": "running",
        "wazuh-execd": "running",
        "wazuh-integratord": "running",
        "wazuh-logcollector": "running",
        "wazuh-maild": "running",
        "wazuh-remoted": "running",
        "wazuh-reportd": "stopped",
        "wazuh-syscheckd": "running",
        "wazuh-clusterd": "running",
        "wazuh-modulesd": "running",
        "wazuh-db": "running",
        "wazuh-apid": "stopped"
      }
    ],
    "total_affected_items": 1,
    "total_failed_items": 0,
```

```
    "failed_items": []
  },
  "message": "Processes status were successfully read in specified node",
  "error": 0
}
```


Aşağıdaki istekle Wazuh yöneticisinin mevcut yapılandırmasını boşaltabilirsiniz (cevap, kısa olması için kısaltılmıştır):

```
curl -k -X GET "https://localhost:55000/manager/configuration?pretty=true&section=global" -H  "Authorization: Bearer $TOKEN"
```

Output


```
{
  "data": {
    "affected_items": [
      {
        "global": {
          "jsonout_output": "yes",
          "alerts_log": "yes",
          "logall": "no",
          "logall_json": "no",
          "email_notification": "yes",
          "email_to": "me@test.example",
          "smtp_server": "mail.test.example",
          "email_from": "wazuh@test.example",
          "email_maxperhour": "12",
          "email_log_source": "alerts.log",
          "white_list": [
            "127.0.0.1",
            "^localhost.localdomain$",
            "8.8.8.8",
            "8.8.4.4"
          ]
        }
      }
    ],
    "total_affected_items": 1,
    "total_failed_items": 0,
    "failed_items": []
  },
  "message": "Configuration was successfully read in specified node",
  "error": 0
}
```


Wazuh Agent Yönetimini Keşfetme

Wazuh ajanlarını yönetmek için Wazuh sunucu API'sini kullanabilirsiniz.

Aşağıdaki istek iki etkin etkeni sıralıyor:

```
curl -k -X GET
"https://localhost:55000/agents?pretty=true&offset=1&limit=2&select=status%2Cid%2Cmanager%2Cname%2C
node_name%2Cversion&status=active" -H  "Authorization: Bearer $TOKEN"
```

Output


```
{
 "data": {
  "affected_items": [
   {
    "node_name": "worker2",
    "status": "active",
    "manager": "wazuh-worker2",
    "version": "Wazuh v4.7.4",
    "id": "001",
    "name": "wazuh-agent1"
   },
   {
    "node_name": "worker2",
    "status": "active",
    "manager": "wazuh-worker2",
    "version": "Wazuh v4.7.4",
    "id": "002",
    "name": "wazuh-agent2"
   }
  ],
  "total_affected_items": 9,
  "total_failed_items": 0,
  "failed_items": []
 },
 "message": "All selected agents information was returned",
 "error": 0
}
```


API isteği göndererek ajan adını ve IP adresini kullanarak yeni bir Wazuh ajanı ekleyin:

```
curl -k -X POST "https://localhost:55000/agents?pretty=true" -H  "Authorization: Bearer $TOKEN" -H  "Content-
Type: application/json" -d "{\"name\":\"NewHost\",\"ip\":\"10.0.10.11\"}"
```

Output


```
{
 "data": {
  "id": "013",
  "key": "MDEzIE5ld0hvc3RfMiAxMC4wLjEwLjEyIDkzOTE0MmE4OTQ4YTNlMzA0ZTdiYzVmZTRhN2Q4Y2I1MjgwMWI
 },
 "error": 0
}
```

Güvenlik Olaylarını İçe Aktarın

> 4.6.0 sürümündeki yenilikler.

Güvenlik olaylarını analiz için Wazuh yöneticisine aktarmak amacıyla Wazuh sunucu API'sini kullanabilirsiniz.

Dakikada 30 istek ve istek başına 100 olay sınırı vardır. Bu sınır, uç noktaların büyük miktarda veriyi çok hızlı bir şekilde almasını önler. Bu sınırı daha da düşürmek veya özelliği devre dışı bırakmak için max_request_per_minute'i işaretleyin.

```
curl -k -X POST "https://localhost:55000/events" -H  "Authorization: Bearer $TOKEN" -H  "Content-Type: application/json" -d '{"events": ["Event value 1", "{\"someKey\": \"Event value 2\"}"]}'
```

Output

```
{
  "data": {
    "affected_items": [

    ],
    "total_affected_items": 2,
    "total_failed_items": 0,
    "failed_items": []
  },
  "message": "All events were forwarded to analisysd",
  "error": 0
}
```

Çözüm

Sonuç olarak, bu örnekler Wazuh API'nin yeteneklerini sergiliyor. Mevcut Wazuh sunucu API isteklerinin tam aralığını keşfetmek için referans belgesini inceleyin.