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": "MDEzIE5Id0hvc3RfMiAxMC4wLjEwLjEyIDkzOTE0MmE4OTQ4YTNlMzA0ZTdiYzVmZTRhN2Q4Y2I1MjgwMWI
 },
 "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.

Revision #7
Created 31 December 2024 13:43:57 by Ayşegül Sarıkaya
Updated 31 December 2024 13:56:05 by Ayşegül Sarıkaya