Harici API entegrasyonu

Wazuh Integrator modülü, Wazuh'un Slack , PagerDuty , VirusTotal , Shuffle ve Maltiverse gibi harici API'lere ve uyarı araçlarına bağlanmasını sağlar . Integrator modülünü diğer yazılımlara bağlanacak şekilde de yapılandırabilirsiniz. Bu entegrasyonlar, güvenlik yöneticilerinin orkestrasyonu geliştirmesini, yanıtları otomatikleştirmesini ve siber tehditlere karşı savunmalarını güçlendirmesini sağlar.

Yapılandırma

Bir entegrasyonu yapılandırmak için Wazuh sunucusundaki `/var/ossec/etc/ossec.conf` dosyasındaki `<ossec_config>` içindeki aşağıdaki yapılandırmayı ekleyin :

```
<integration>
  <name> </name>
  <hook_url> </hook_url> <!-- Required for Slack, Shuffle, and Maltiverse -->
  <api_key> </api_key> <!-- Required for PagerDuty, VirusTotal, and Maltiverse -->
  <alert_format>json</alert_format> <!-- Required for Slack, PagerDuty, VirusTotal, Shuffle, and Maltiverse -->

  <!-- Optional filters -->
  <rule_id> </rule_id>
  <level> </level>
  <group> </group>
  <event_location> </event_location>
  <options> </options>
</integration>
```

Nerede:

- `<name>`entegre edilecek hizmetin adını belirtir. İzin verilen değerler `slack`, `pagerduty`, `virustotal`, `shuffle`, ' dir `maltiverse`. Özel entegrasyonlar için, ad ile başlayan herhangi bir dize olmalıdır `custom-`.
- `<hook_url>`entegre edilen yazılımla iletişim için kullanılan URL'dir. Slack, Shuffle ve Maltiverse entegrasyonları için zorunludur.
- `<api_key>`PagerDuty, VirusTotal veya Maltiverse API'sinden almış olacağınız anahtardır. Bu PagerDuty, VirusTotal ve Maltiverse için zorunludur.
- `<alert_format>`uyarı dosyasını JSON biçiminde yazar. Integrator modülü, alan değerlerini almak için bu uyarı dosyasını kullanır. İzin verilen değer `json`.
- `<rule_id>`kural kimliğine göre uyarıları filtreler. İzin verilen değerler virgülle ayrılmış kural kimlikleridir.
- `<level>`0uyarıları kural düzeyine göre filtreler, böylece yalnızca belirtilen düzey veya üstündeki uyarılar gönderilir. İzin verilen değer, ile arasındaki herhangi bir uyarı düzeyidir

16.

- `<group>`uyarıları kural grubuna göre filtreler. VirusTotal entegrasyonu için yalnızca syscheck grubundan kurallar kullanılabilir. İzin verilen değerler herhangi bir kural grubu veya virgülle ayrılmış kural gruplarıdır.
- `<event_location>`uyarıları olayın nereden kaynaklandığına göre filtreler. İzin verilen değer herhangi bir sregex ifadesidir.
- `<options>`JSON nesnesinde sağlanan bilgilere göre önceki alanların üzerine yazar veya özelleştirme alanları ekler. İzin verilen değer json'dur.

> Not: Yapılandırma dosyasında herhangi bir değişiklik yaptığınızda Wazuh yöneticisini yeniden başlatın. Bu, değişikliklerin etkili olmasını sağlayacaktır.

Aşağıdaki komutla komut satırı arayüzü üzerinden Wazuh yöneticisini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

İsteğe Bağlı Filtreler

Wazuh Integrator modülü, hangi uyarıların harici platformlara gönderileceğini belirlemek için isteğe bağlı filtre alanlarını kullanır. Yalnızca filtre koşullarını karşılayan uyarılar gönderilir. Hiçbir filtre belirtilmezse, tüm uyarılar gönderilir.

Filtreler ayarlanırken aşağıdaki hususlara dikkat edilmelidir:

- Virgülle ayrılmış liste etiketini kullanarak birden fazla grup adı belirtmek mümkündür `<group>`. Uyarının grubu listedeki gruplardan herhangi biriyle eşleşirse uyarı gönderilir, aksi takdirde yok sayılır.
- Virgülle ayrılmış liste etiketini kullanarak birden fazla kural kimliği belirtmek mümkündür `<rule_id>`. Uyarı, uyarının kural kimliği listedeki herhangi bir kimlikle eşleşirse gönderilir, aksi takdirde yok sayılır.
- Daha önce açıklanan alanları birlikte belirtmek mümkündür. Uyarı, hem uyarının kural kimliği hem de grubu listelerdeki kimliklerden ve gruplardan herhangi biriyle eşleşirse gönderilir, aksi takdirde yok sayılır.

>

> Not: Yukarıda belirtilen grup ve kural tanımlayıcılarının dikkatlice kontrol edilmesi önerilir, çünkü bunların yanlış tanımlanması entegrasyona beklenen uyarıların gönderilmemesine neden olacaktır.

Slack

Slack, kuruluşlar içinde iletişimi ve ekip çalışmasını kolaylaştıran bulut tabanlı bir işbirliği platformudur. Bu entegrasyon, Slack gelen webhook'larını kullanır ve güvenlik uzmanlarının gerçek zamanlı uyarıları doğrudan belirlenmiş kanallar içinde almalarını sağlar.

Bu entegrasyonu kurmak için aşağıdaki adımları izleyin:

1. Gelen webhook'ları etkinleştirin ve Slack kanalınız için bir tane oluşturun. Bunun için gelen webhook'lar hakkındaki Slack rehberini izleyin .
2. Aşağıdaki yapılandırmayı `/var/ossec/etc/ossec.conf`Wazuh sunucusundaki dosyaya ekleyin. `<WEBHOOK_URL>`Gelen webhook'unuzla değiştirin.

```
<ossec_config>
  <integration>
    <name>slack</name>
    <hook_url><SLACK_WEBHOOK_URL></hook_url> <!-- Replace with your Slack hook URL -->
    <alert_format>json</alert_format>
  </integration>
</ossec_config>
```

> Not: Seçenekler etiketini kullanarak özelleştirme alanlarıyla bir JSON nesnesi ayarlayabilirsiniz . Kullanılabilir özelleştirme alanları hakkında bilgi için Slack API referansını ziyaret edin.

3. Değişiklikleri uygulamak için Wazuh yöneticisini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

Yapılandırma tamamlandıktan sonra seçili kanalda uyarılar gösterilmeye başlanır.

Seçili Slack kanalındaki uyarılar

Image not found or type unknown

PagerDuty

PagerDuty, BT departmanları için uygun bir SaaS olay müdahale platformudur. PagerDuty, programlara ve yükseltme politikalarına göre uyarıları doğru kişilere veya ekiplere yükselterek olay müdahale iş akışlarını yürütür. PagerDuty entegrasyonu, Wazuh uyarılarını Olay Pano'suna iletmek için PagerDuty API'sini kullanır.

Bu entegrasyonu kurmak için aşağıdaki adımları izleyin:

1. Yeni bir PagerDuty servisi oluşturarak Events API v2 entegrasyon anahtarınızı edinin .
2. Aşağıdaki yapılandırmayı `/var/ossec/etc/ossec.conf`Wazuh sunucusundaki dosyaya ekleyin. `PAGERDUTY_API_KEY`PagerDuty entegrasyon anahtarınızla değiştirin. Kural düzeyi filtresi isteğe bağlıdır ve bunu kaldırabilir veya entegrasyon için başka bir düzey değeri ayarlayabilirsiniz.

```
<ossec_config>
  <integration>
    <name>pagerduty</name>
    <api_key><PAGERDUTY_API_KEY></api_key> <!-- Replace with your PagerDuty API key -->
    <level>10</level>
    <alert_format>json</alert_format> <!-- New mandatory parameter since v4.7.0 -->
  </integration>
</ossec_config>
```

> Not: Seçenekler etiketini kullanarak özelleştirme alanlarıyla bir JSON nesnesi ayarlayabilirsiniz . Kullanılabilir özelleştirme alanları hakkında bilgi için PagerDuty API referansını ziyaret edin.

3. Değişiklikleri uygulamak için Wazuh yöneticisini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

Yapılandırma tamamlandıktan sonra Pagerduty panosunda uyarılar gösterilmeye başlar.

PagerDuty'deki uyarılar
Image not found or type unknown

VirusTotal

VirusTotal , virüsleri, solucanları, truva atlarını ve diğer kötü amaçlı içerikleri tespit etmek için dosyaları ve URL'leri analiz eden bir çevrimiçi hizmettir. Bu entegrasyon, VirusTotal veritabanını kullanarak kötü amaçlı dosyaların incelenmesine olanak tanır. Bununla ilgili daha fazla bilgiyi VirusTotal entegrasyon bölümünde bulabilirsiniz.

Bu entegrasyonu kurmak için şu adımları izleyin:

1. API anahtarınızı VirusTotal API anahtarı sayfasından alın.
2. `/var/ossec/etc/ossec.conf`Wazuh sunucusunda düzenleme yapın ve aşağıdaki gibi bir yapılandırma bloğu ekleyin. `<VIRUSTOTAL_API_KEY>`VirusTotal API anahtarınızla değiştirin.

```
<integration>
  <name>virustotal</name>
  <api_key><VIRUSTOTAL_API_KEY></api_key> <!-- Replace with your VirusTotal API key -->
  <group>syscheck</group>
  <alert_format>json</alert_format>
</integration>
```

3. Değişiklikleri uygulamak için Wazuh yöneticisini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

Shuffle

Shuffle, SOAR'ın açık kaynaklı bir yorumudur. Tak ve çalıştır uygulamalarıyla kuruluş genelinde veri aktarımı yapar. Shuffle entegrasyonu, bir webhook kullanarak Wazuh uyarılarının bir Shuffle İş Akışına iletilmesine olanak tanır .

Bu entegrasyonu kurmak için aşağıdakileri yapın:

1. Shuffle'a gidin, E-posta uygulamasını kullanarak bir İş Akışı oluşturun ve sürümü seçin.
2. E-posta yapılandırmasında **Alıcıları** ve **Konuyu** ayarlayın . `$exec`Uyarı bilgilerini eklemek için Gövde'ye koyun.
3. İş Akışına bir webhook ekleyin.
4. Webhook'u başlatın ve webhook URL'sini kopyalayın.
5. `/var/ossec/etc/ossec.conf`Wazuh sunucusunda düzenleme yapın ve aşağıdaki gibi bir yapılandırma bloğu ekleyin.
6. Shuffle webhook ID ile değiştirin `<SHUFFLE_WEBHOOK_ID>`. Kural düzeyi filtresi isteğe bağlıdır. Bunu kaldırabilir veya entegrasyon için başka bir düzey değeri ayarlayabilirsiniz.

```
<integration>
  <name>shuffle</name>
  <hook_url>https://shuffler.io/api/v1/hooks/<SHUFFLE_WEBHOOK_ID></hook_url> <!-- Replace with your Sh
  <level>3</level>
  <alert_format>json</alert_format>
</integration>
```

> Not: Seçenekler etiketini kullanarak özelleştirme alanlarıyla bir JSON nesnesi ayarlayabilirsiniz . Kullanılabilir özelleştirme alanları hakkında bilgi için Shuffle API referansını ziyaret edin.

7. Değişiklikleri uygulamak için Wazuh yöneticisini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

Yapılandırma tamamlandıktan sonra e-posta gelen kutunuzda uyarılar gösterilmeye başlar.

Shuffle'daki uyarılar

Image not found or type unknown

Maltiverse

Maltiverse, Tehlike Göstergelerini (IoC'ler) dizinlemek ve aramak için açık kaynaklı ve işbirlikçi bir platformdur. Yüzden fazla genel, özel ve topluluk tehdit istihbarat kaynağından bilgi toplar.

Bu entegrasyon, Maltiverse API aracılığıyla Wazuh uyarılarındaki IoC'leri tanımlar. Maltiverse verileriyle zenginleştirilmiş yeni uyarılar üretir. Maltiverse veri alanları, ECS standardının (Elastic Common Schema) tehdit sınıflandırmasına dayanır.

Bu entegrasyonu kurmak için aşağıdaki adımları izleyin:

1. API anahtarınızı Maltiverse sayfasından alın.
2. `/var/ossec/etc/ossec.conf`Wazuh sunucusunda düzenleme yapın ve aşağıdaki gibi bir yapılandırma bloğu ekleyin. `<MALTIVERSE_API_KEY>`Maltiverse API anahtarınızla değiştirin. Kural düzeyi filtresi isteğe bağlıdır. Bunu kaldırabilir veya entegrasyon için başka bir düzey değeri ayarlayabilirsiniz.

   ```
   <integration>
     <name>maltiverse</name>
     <hook_url>https://api.maltiverse.com</hook_url>
     <level>3</level>
     <api_key><MALTIVERSE_API_KEY></api_key> <!-- Replace with your Maltiverse API key -->
     <alert_format>json</alert_format>
   </integration>
   ```

3. Değişiklikleri uygulamak için Wazuh yöneticisini yeniden başlatın:

   Systemd

   ```
   systemctl restart wazuh-manager
   ```

   SysV Başlatma

   ```
   service wazuh-manager restart
   ```

Yapılandırma tamamlandıktan sonra, varsa zenginleştirilmiş uyarılar Wazuh Pano'sunda gösterilmeye başlar.

Wazuh panosunda zenginleştirilmiş uyarılar

Özel Entegrasyon

Wazuh Integrator modülü, Wazuh'u diğer harici yazılımlarla bağlar. Bu, Wazuh uyarı sisteminin entegrasyon betikleri aracılığıyla yazılım ürünlerinin API'leriyle entegre edilmesiyle elde edilir.

`/var/ossec/etc/ossec.conf`Aşağıda özel entegrasyon için dosyadaki bir yapılandırma bloğunun örneği verilmiştir .

```
<!--Custom external Integration -->
<integration>
  <name>custom-integration</name>
  <hook_url><WEBHOOK></hook_url>
  <level>10</level>
  <group>multiple_drops,authentication_failures</group>
  <api_key><API_KEY></api_key> <!-- Replace with your external service API key -->
  <alert_format>json</alert_format>
 <options>{"data": "Custom data"}</options> <!-- Replace with your custom JSON object -->
</integration>
```

Yer değiştirmek:

- `<WEBHOOK>`harici uygulamanın webhook URL'si ile.
- `<API_KEY>`harici uygulamanın API anahtarı ile.

Entegrasyon Betiği Oluşturma

Entegrasyon betiği oluştururken aşağıdaki talimatları izlemeniz önerilir:

1. `/var/ossec/integrations/`Yapılandırma bloğunda belirtilen adla aynı adı taşıyan betiği Wazuh sunucusundaki dizinde oluşturun .
2. Komut dosyası yürütme izinleri içermeli ve `root`grubun kullanıcısına ait olmalıdır `wazuh`. Aşağıdaki komutlar `/var/ossec/integrations/custom-script`komut dosyasına izinler ve sahiplik atar.

   ```
   chmod 750 /var/ossec/integrations/custom-script
   chown root:wazuh /var/ossec/integrations/custom-script
   ```

3. Entegrasyon betiğinin ilk satırı yorumlayıcısını belirtmelidir, aksi takdirde Wazuh betiği nasıl okuyacağını ve çalıştıracağını bilemez. Aşağıdaki örnek satır Python yorumlayıcısını

belirtir:

```
#!/usr/bin/env python
```

4. Komut dosyası aşağıdaki argümanları kontrol eder çünkü onlardan yapılandırma seçenekleri alacaktır.
   - İlk parametre uyarıyı içeren dosyanın konumunu içerir. Parametre `/logs/alerts/alerts.json` Wazuh Integrator modülünde varsayılan olarak geçirilen dosyadır:

     ```
     alert_file = open(sys.argv[1])
     ```

   - `api_key`İkinci parametre, blokta tanımlanan seçenek olan API anahtarını içerir `<integration>`:

     ```
     api_key = sys.argv[2]
     ```

   - `hook_url`Üçüncü parametre, blokta tanımlanan seçenek olan webhook URL'sini içerir `<integration>`:

     ```
     hook_url = sys.argv[3]
     ```

   Yukarıdakilerden hiçbiri belirtilmezse parametreler boş alınacaktır.
5. İlk parametrede belirtilen dosyanın içeriğini okuyun ve uyarıdan entegrasyon için ilgili alanları çıkarın. Seçenekte JSON kullanılmışsa `alert_format`, bilginin bir JSON nesnesi olarak yüklenmesi gerekir.

```
alert_level = alert_json['rule']['level']
ruleid = alert_json['rule']['id']
description = alert_json['rule']['description']
agentid = alert_json['agent']['id']
agentname = alert_json['agent']['name']
path = alert_json['syscheck']['path']
```

`/logs/alerts/alerts.json`Entegrasyon betiğinin geliştirilmesine başlamadan önce, yorumlanacak uyarıların formatını bulmak için dosyayı kontrol etmenizi öneririz .

Revision #7
Created 11 December 2024 16:30:44 by Ayşegül Sarıkaya
Updated 31 December 2024 13:26:28 by Ayşegül Sarıkaya