Wazuh Server

Wazuh sunucusu, ajanlardan, harici API'lerden ve ağ cihazlarından aldığı verileri analiz eden Wazuh merkezi bileşenidir. Alınan verileri, güvenlik izleme ve yönetimi için uyarılar üretmek üzere önceden tanımlanmış bir kural kümesiyle ilişkilendirerek ve eşleştirerek analiz eder. Wazuh sunucusu iki ana bileşenden oluşur; Wazuh yöneticisi ve Filebeat . Wazuh yöneticisi veri analizi ve uyarılardan sorumludur, dizinleyici entegrasyonu ise analiz edilen verileri Wazuh dizinleyicisine iletir. Nasıl kurulacağı ve ayarlanacağı hakkında bilgi için Wazuh sunucusu kurulum belgelerine bakın.

- Alarm Yönetimi
- Olay Günlüğü Tutma
- Harici API entegrasyonu
- Indexer Entegrasyonu
- Wazuh Yöneticisi
- Sıraya Girme Mekanizmaları

Alarm Yönetimi

`/var/ossec/logs/alerts/alerts.log`Uyarılar, Wazuh aracılarından ve aracısız aygıtlardan alınan olayları işledikten sonra Wazuh yöneticisi tarafından oluşturulan bildirimlerdir. Varsayılan olarak, uyarılar ve dosyalarında saklanır `/var/ossec/logs/alerts/alerts.json`.

Varsayılan olarak, Wazuh sunucusu, oluşturulan uyarıları dizinleme için Wazuh dizinleyicisine iletmek için Filebeat'i kullanır. Ek olarak, Wazuh yöneticisini syslog sunucuları, e-posta sistemleri ve veritabanlarını içeren diğer sistemlere uyarıları iletecek şekilde yapılandırabilirsiniz.

Uyarı Eşiği

Uyarı eşiği, bir uyarının tetiklenmesi için aşılması gereken en düşük önem seviyesidir. Wazuh yöneticisi, kurallar kümesindeki eşleşen kurala göre izlenen uç noktalardan gelen her olaya bir önem seviyesi atar. Varsayılan olarak, yalnızca önem seviyesi `3`veya daha yüksek olan uyarıları tetikler.

Yapılandırma

`/var/ossec/etc/ossec.conf`Uyarı eşiği , Wazuh sunucusundaki yapılandırma dosyasında XML etiketi içerisinde yapılandırılır `<alerts>`.

Aşağıdaki kod bloğu, olaylar ve uyarıların e-posta yoluyla iletilmesi için varsayılan uyarı eşiği yapılandırmasını gösterir:

```
<ossec_config>
  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>
</ossec_config>
```

Nerede:

- `<log_alert_level>` etiket , `/var/ossec/logs/alerts/alerts.log` ve/veya `/var/ossec/logs/alerts/alerts.json` dosyada depolanan uyarıları tetiklemek için minimum önem seviyesini ayarlar . Varsayılan değer `3`'dür . İzin verilen değer, kurallar sınıflandırma kılavuzunda belirtildiği gibi `1` ila `16` arasında herhangi bir tam sayıdır
- Etiket `<email_alert_level>`, bir uyarının e-posta bildirimi oluşturması için minimum önem seviyesini ayarlar. Varsayılan değer 'dir `12`. İzin verilen değer, `1`'den 'e kadar herhangi bir

tam sayıdır `16`. Bu ayar, ayrıntılı e-posta uyarısı yapılandırmasını geçersiz kılar. Ancak, bireysel kurallar içindeki `alert_by_email` seçenek , bir e-posta uyarısını tetiklemek için hem genel hem de ayrıntılı uyarı düzeyi eşiklerini geçersiz kılabilir.

Uyarı eşiği yapılandırma hakkında ayrıntılı bilgi için uyarı başvuru kılavuzuna bakın.

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

Aşağıdaki komutla komut satırı arayüzü üzerinden Wazuh yöneticisini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

Uyarıları İletme

Wazuh yöneticisi, dizinleme ve analiz yetenekleri için uyarıları varsayılan olarak Wazuh dizinleyicisine iletir. Ayrıca, Wazuh yöneticisi, analiz ve yedekleme için uyarıları yapılandırma ve diğer sistemlere iletme yeteneği sağlar.

Syslog Çıktısını Yapılandırma

Syslog_output seçeneğini kullanarak Wazuh sunucusunu bir syslog sunucusuna uyarılar gönderecek şekilde yapılandırabilirsiniz . Uyarıları bir syslog sunucusuna iletmek, merkezi izleme ve özel raporlama için yararlı olabilir.

Yapılandırma

`/var/ossec/etc/ossec.conf`Syslog çıktısı, blok içindeki Wazuh sunucu yapılandırma dosyasında yapılandırılır . Varsayılan olarak, Wazuh yöneticisi uyarıları UDP protokolü üzerinden `<ossec_config>` port kullanarak syslog sunucularına iletir .`514`

Aşağıdaki kod bloğu, uyarıları bir syslog sunucusuna iletmek için örnek bir yapılandırmayı göstermektedir:

```
<ossec_config>
  <syslog_output>
    <level>9</level>
    <server>192.168.1.241</server>
  </syslog_output>
</ossec_config>
```

Yapılandırma seçenekleri aşağıdaki şekilde tanımlanmıştır:

- Etiket `<level>`, syslog sunucusuna iletilecek uyarıların minimum önem seviyesini ayarlar. Örnek değer, `9`Wazuh sunucusunun uyarıları yalnızca uyarı seviyesi 'den yüksekse syslog sunucusuna ilettiğini gösterir `9`. Bu seçenek tanımlanmamışsa, Wazuh sunucusu tüm uyarıları syslog sunucusuna iletir.
- Etiket `<server>`, uyarıları iletmek için syslog sunucusunun IP adresini veya ana bilgisayar adını ayarlar. `192.168.1.241`Yapılandırmadaki IP adresi bir örnek olarak kullanılır.

Değişikliklerin her yapılandırmadan sonra uygulanması için Wazuh yönetici hizmetini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

Yapılandırma dosyasında blok `<syslog_output>`içerisinde birden fazla blok tanımlayarak uyarıları birden fazla syslog sunucusuna iletebilirsiniz .`<ossec_config>`/var/ossec/etc/ossec.conf

```
<ossec_config>
  <syslog_output>
    <server>192.168.1.240</server>
  </syslog_output>

  <syslog_output>
    <level>9</level>
    <server>192.168.1.241</server>
  </syslog_output>
</ossec_config>
```

Yukarıdaki yapılandırmada,

- İlk `<syslog_output>`blok tüm uyarıları filtrelemeden IP adresine sahip syslog sunucusuna gönderir `192.168.1.240`.

- İkinci blok , yalnızca uyarı seviyesi 'den yüksekse `<syslog_output>`syslog sunucusuna uyarılar gönderir .`192.168.1.2419`

E-posta Uyarılarını Yapılandırma

Wazuh, bir Wazuh sunucusunda oluşturulduğunda e-posta sistemlerine uyarılar göndermek için bir özellik sunar. Kurallar tetiklendiğinde veya özelleştirilmiş ayarlara göre bir veya daha fazla e-posta adresine e-posta uyarıları göndermek üzere yapılandırabilirsiniz. Bu yapılandırma günlük olay raporları ve daha fazlası için size yardımcı olabilir.

Kural kimliği `553`tetiklendiğinde Wazuh tarafından gönderilen örnek bir e-posta aşağıda gösterilmektedir:

```
Wazuh Notification.
2024 Apr 29 08:58:30

Received From: wazuh-server->syscheck
Rule: 553 fired (level 7) -> "File deleted."
Portion of the log(s):

File '/var/ossec/test_dir/somefile.
txt' deleted
Mode: realtime

Attributes:
 - Size: 0
 - Permissions: rw-r--r--
 - Date: Mon Apr 29 08:46:12 2024
 - Inode: 841858
 - User: root (0)
 - Group: root (0)
 - MD5: d41d8cd98f00b204e9800998ecf8427e
 - SHA1: da39a3ee5e6b4b0d3255bfef95601890afd80709
 - SHA256: e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855


--END OF NOTIFICATION
```

Genel E-posta Seçenekleri

Wazuh'un e-posta uyarıları göndermesini yapılandırmak için `/var/ossec/etc/ossec.conf` dosyanın `<global>` bölümündeki e-posta seçeneklerini yapılandırıyoruz .

E-posta adresine uyarı göndermek için örnek bir e-posta yapılandırması `me@test.com` aşağıda gösterilmektedir:

```
<ossec_config>
  <global>
```

```
    <email_notification>yes</email_notification>
    <email_to>me@test.com</email_to>
    <smtp_server>mail.test.com</smtp_server>
    <email_from>wazuh@test.com</email_from>
  </global>
  ...
</ossec_config>
```

Yukarıdakiler yapılandırıldıktan sonra, `email_alert_level`bir e-postayı tetiklemek için seçeneğin minimum uyarı seviyesine ayarlanması gerekir. Varsayılan olarak, bu seviye olarak ayarlanır `12`.

Aşağıdaki örnek yapılandırma, e-posta uyarılarının gönderileceği minimum seviyeyi belirler `10`:

```
<ossec_config>
  <alerts>
    <email_alert_level>10</email_alert_level>
  </alerts>
  ...
</ossec_config>
```

Değişikliklerin her yapılandırmadan sonra uygulanması için Wazuh yönetici hizmetini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV İnit

```
service wazuh-manager restart
```

> Uyarı: Wazuh SMTP kimlik doğrulamasını işlemez. E-posta servisiniz bunu kullanıyorsa, bir sunucu rölesi yapılandırmanız gerekir .

Ayrıntılı E-posta Seçenekleri

Wazuh, e-posta uyarıları için ayrıntılı yapılandırma seçeneklerine izin verir. Bu ayar, dosyanın bölümünde yapılandırılan genel e-posta seçeneklerini genişletir. Ayrıntılı e-posta yapılandırmaları, dosyanın etiketi içinde tanımlanır .`<global>`/var/ossec/etc/ossec.conf`<email_alerts>`/var/ossec/etc/ossec.conf

> Uyarı: Bölümde yapılandırılan minimum önem düzeyi `<alerts>`bu ayrıntılı e-posta yapılandırmalarına uygulanır ve bunları geçersiz kılar. Örneğin, Wazuh yöneticisini kural

tetiklendiğinde bir e-posta gönderecek şekilde yapılandırırsanız `526`ancak kuralın düzeyi bölümde belirtilen minimum düzeyden düşükse `<alerts>`uyarı gönderilmez.

Seviyeye Göre E-posta Uyarısı

Bu seçenek, Wazuh yöneticisini, önem düzeyi ayarlanan değere eşit veya daha büyük olduğunda e-posta uyarıları gönderecek şekilde yapılandırır. Bu seçenek aşağıdaki şekilde yapılandırılır:

```
<email_alerts>
  <email_to>you@example.com</email_to>
  <level>4</level>
  <do_not_delay/>
</email_alerts>
```

`you@example.com`Bu yapılandırma, Wazuh yöneticisinin , seviyesi eşit veya daha büyük olan herhangi bir kural tetiklendiğinde bir e-posta göndermesine olanak tanır `4`.

> Not: Buradaki önem seviyesi `<alerts>` bölümde yapılandırılan `email_alert_level` önem seviyesinden daha düşükse , e-posta gönderilmeyecektir.

Etkinlik Lokasyonuna Göre E-posta Uyarısı

Bu `event_location`seçenek, olayın kaynaklandığı konuma göre e-posta uyarıları göndermeyi içerir. Oluşturulan uyarı, e-posta yoluyla iletilmek üzere olay konumuyla eşleşmelidir. Bu seçenek için izin verilen değerler Wazuh aracı adı, ana bilgisayar adı, IP adresi veya günlük dosyasıdır.

Bu seçenek aşağıdaki şekilde yapılandırılır:

```
<email_alerts>
  <email_to>you@example.com</email_to>
  <event_location>server1</event_location>
  <do_not_delay/>
</email_alerts>
```

`you@example.com`Bu yapılandırma, Wazuh yöneticisinin uyarıları oluşturan olayların Wazuh adlı aracıda kaynaklandığı zaman adresine bir e-posta göndermesine olanak tanır `server1`.

Kural Kimliğine Dayalı E-posta

Bu `rule_id`seçenek, kural kimliklerine dayalı uyarı e-postaları göndermek için kullanılır. Bu seçenek, yalnızca belirli tanımlanmış kurallar tetiklendiğinde e-postaların gönderilmesini sınırlar.

Bu seçenek aşağıdaki şekilde yapılandırılır:

```
<email_alerts>
  <email_to>you@example.com</email_to>
  <rule_id>515, 516</rule_id>
  <do_not_delay/>
</email_alerts>
```

Bu yapılandırma , Wazuh yöneticisinin `you@example.com`kurallar tetiklendiğinde bir e-posta göndermesine olanak tanır .`515``516`

Kural Grubuna Dayalı E-posta

Seçenek `group`, uyarıların ait olduğu bir veya daha fazla kural grubuna göre e-posta göndermek üzere yapılandırılabilir.

Bu seçenek aşağıdaki şekilde yapılandırılır:

```
<email_alerts>
  <email_to>you@example.com</email_to>
  <group>pci_dss_10.6.1,</group>
</email_alerts>
```

`you@example.com`Bu yapılandırma, Wazuh yöneticisinin, grubun parçası olan herhangi bir kural `pci_dss_10.6.1`herhangi bir Wazuh izlenen uç noktasında tetiklendiğinde bir e-posta göndermesine olanak tanır.

Birden Fazla Seçenek ve Birden Fazla E-posta

E-posta uyarıları, her biri benzersiz kriterlere sahip birden fazla e-posta adresine gönderilebilir.

Aşağıdaki örnek yapılandırma, birden fazla kritere sahip e-posta uyarılarının birden fazla e-posta adresine nasıl gönderileceğini gösterir:

```
<ossec_config>
  <email_alerts>
    <email_to>alice@test.com</email_to>
    <event_location>endpoint1|endpoint2</event_location>
  </email_alerts>

  <email_alerts>
    <email_to>is@test.com</email_to>
```

```
    <event_location>/log/secure$</event_location>
  </email_alerts>

  <email_alerts>
    <email_to>bob@test.com</email_to>
    <event_location>192.168.</event_location>
  </email_alerts>

  <email_alerts>
    <email_to>david@test.com</email_to>
    <level>12</level>
  </email_alerts>
</ossec_config>
```

Bu yapılandırma şunları gönderir:

- `alice@test.com`Herhangi bir uyarı tetiklendiğinde `endpoint1`e -posta adresinize gönderilecek `endpoint2`.
- `is@test.com`Uyarıların dosyadan gelip gelmediğine dair bir e-posta `/log/secure`.
- `bob@test.com`Uyarıların ağdaki herhangi bir uç noktadan gelip gelmediğine dair bir e-posta `192.168.0.0/24`.
- `david@test.com`Uyarıların seviyesi eşit veya daha yüksekse e-posta gönderilecektir `12`.

Bir Uyarıyı E-postayla İletmeyi Zorla

E-posta yoluyla uyarı göndermek için minimum önem seviyesi `12`varsayılan olarak. Wazuh yöneticisini yapılandırılmış minimum önem seviyesinin altında bir e-posta uyarısı göndermek üzere yapılandırabilirsiniz. Bunu yapmak için aşağıdaki kural seçeneklerinden birini kullanmanız gerekir:

- `alert_by_email`her zaman e-posta ile uyarmak.
- `no_email_alert`asla e-posta yoluyla uyarıda bulunmayın.
- `no_log`Bu uyarının kaydedilmemesi için.

Örneğin, aşağıdaki kural tanımı, `502`minimum önem düzeyi ne olarak ayarlanmış olursa olsun, kural her tetiklendiğinde bir e-posta gönderir:

```
<rule id="502" level="3">
  <if_sid>500</if_sid>
  <options>alert_by_email</options>
  <match>Ossec started</match>
  <description>Ossec server started.</description>
</rule>
```

Kimlik Doğrulamalı SMTP Sunucusu

Wazuh e-posta uyarıları, Gmail gibi kimlik doğrulaması olan SMTP sunucularını desteklemez. Ancak, bu e-postaları Postfix gibi bir sunucu rölesi aracılığıyla gönderebilirsiniz.

Postfix'i Gmail ile yapılandırmak için aşağıdaki adımları röle sunucunuzda gerçekleştirin.

1. Gerekli paketleri yüklemek için bu komutu çalıştırın. Posta sunucusu yapılandırma türü hakkında sorulursa *Yapılandırma yok'u seçin.*

   CentOS

   ```
   yum update && yum install postfix mailx cyrus-sasl cyrus-sasl-plain
   ```

   Ubuntu

   ```
   apt-get update && apt-get install postfix mailutils libsasl2-2 ca-certificates libsasl2-modules
   ```

2. Postfix'i yapılandırmak için bu satırları dosyaya ekleyin `/etc/postfix/main.cf`. Eksikse dosyayı oluşturun.

   CentOS

   ```
   relayhost = [smtp.gmail.com]:587
   smtp_sasl_auth_enable = yes
   smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
   smtp_sasl_security_options = noanonymous
   smtp_tls_CAfile = /etc/ssl/certs/ca-bundle.crt
   smtp_use_tls = yes
   ```

   Ubuntu

   ```
   relayhost = [smtp.gmail.com]:587
   smtp_sasl_auth_enable = yes
   smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
   smtp_sasl_security_options = noanonymous
   smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
   smtp_use_tls = yes
   smtpd_relay_restrictions = permit_mynetworks, permit_sasl_authenticated, defer_unauth_destination
   ```

3. Gönderenin kimlik bilgilerini dosyaya ayarlayın `/etc/postfix/sasl_passwd`ve Postfix için bir veritabanı dosyası oluşturun. `<USERNAME>`ve `<PASSWORD>`değişkenlerini sırasıyla gönderenin e-posta adresi kullanıcı adı ve parolasıyla değiştirin.

```
echo [smtp.gmail.com]:587 <USERNAME>@gmail.com:<PASSWORD> > /etc/postfix/sasl_passwd
postmap /etc/postfix/sasl_passwd
```

> Not: Şifre bir Uygulama Şifresi olmalıdır . Uygulama Şifreleri yalnızca 2 Adımlı Doğrulama özelliği açık olan hesaplarda kullanılabilir.

4. Parola DB dosyanızı yalnızca `root`kullanıcının tam okuma ve yazma erişimine sahip olması için güvenceye alın. Bunun nedeni `/etc/postfix/sasl_passwd`ve `/etc/postfix/sasl_passwd.db` dosyalarının düz metin kimlik bilgilerine sahip olmasıdır.

```
chown root:root /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db
chmod 0600 /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db
```

5. Yapılandırma değişikliklerini gerçekleştirmek için Postfix'i yeniden başlatın:

   Systemd

   ```
   systemctl restart postfix
   ```

   SysV Başlatma

   ```
   service postfix restart
   ```

6. Yapılandırmayı test etmek için aşağıdaki komutu çalıştırın:

   ```
   echo "Test mail from postfix" | mail -s "Test Postfix" -r "<CONFIGURED_EMAIL>" <RECEIVER_EMAIL>
   ```

   Yer değiştirmek:
   - `<CONFIGURED_EMAIL>`Yapılandırılmış e-posta adresinizle.
   - `<RECEIVER_EMAIL>`Alıcının e-posta adresiyle birlikte.

   Komut, alıcının e-postasına `Test Postfix` konu ve `Test mail from postfix` gövdeyi içeren bir e-posta gönderir.
   If you get the error message `fatal: tls_fprint: error computing md5 message digest` in the `/var/log/maillog` file, run the following commands to switch Postfix from the default `MD5` hashing function to `SHA-256`:

   ```
   #
   ```

`/var/log/maillog` dosyasında `fatal: tls_fprint: error computing md5 message digest` hata mesajı alırsanız , Postfix'i varsayılan `MD5` karma işlevinden `SHA-256`'ya geçirmek için aşağıdaki komutları çalıştırın :

```
postconf -e smtp_tls_fingerprint_digest=sha256
postconf -e smtpd_tls_fingerprint_digest=sha256
```

7. `<global>`Wazuh sunucusunun `/var/ossec/etc/ossec.conf`dosyasının etiketi içerisinde e-posta bildirimlerini aşağıdaki şekilde yapılandırın:

```
<global>
  <email_notification>yes</email_notification>
  <smtp_server>localhost</smtp_server>
  <email_from><USERNAME>@gmail.com</email_from>
  <email_to><RECEIVER_EMAIL></email_to>
</global>
```

   Nerede:
   - `<email_notification>`e-posta uyarılarının kullanımını değiştirir.
   - `<smtp_server>`uyarıları iletmek için kullanılacak SMTP sunucusunu tanımlar.
   - `<email_from>`yapılandırılmış gönderenin e-posta adresini belirtir. `<USERNAME>`E-posta adresinizin yapılandırılmış kullanıcı adınızla değiştirin.
   - `<email_to>`uyarıların alıcısının e-posta adresini belirtir. `<RECEIVER_EMAIL>`Alıcının e-posta adresiyle değiştirin.

8. Değişiklikleri uygulamak için Wazuh yöneticisini yeniden başlatın:

   Systemd

   ```
   systemctl restart wazuh-manager
   ```

   SysV Başlatma

   ```
   service wazuh-manager restart
   ```

Veritabanı Çıktısını Yapılandırma

Wazuh, uyarıları veritabanı sistemlerine iletmeyi destekler. Wazuh yöneticisini, oluşturulan uyarıları bir veritabanına çıktı olarak verecek şekilde yapılandırabilirsiniz. Bu yapılandırmayı elde etmek için, Wazuh yöneticisini kullanmak istediğiniz veritabanı türündeki kaynaklardan derlemelisiniz. Wazuh şu anda MySQL ve PostgreSQL veritabanlarını destekler.

> Not: Bu kılavuz, MySQL veya PostgreSQL'i zaten kurduğunuzu ve kullanıcıları ve veritabanlarını nasıl oluşturacağınızı bildiğinizi varsayar.

Ön Koşullar

Yapılandırmak istediğiniz veritabanı sistemine ait geliştirme kütüphanelerini kurmanız ve Wazuh yöneticisini gerekli veritabanı sistemini kullanacak şekilde derlemeniz gerekmektedir.

1. Veritabanı sistemi için geliştirme kütüphanelerini yükleyin:

   - **MySQL için** :

     Yum

     ```
     yum install mysql-devel
     ```

     APT

     ```
     apt-get install libmysqlclient-dev
     ```

   - **PostgreSQL için** :

     Yum

     ```
     yum install postgresql-devel
     ```

     APT

     ```
     apt-get install libpq-dev
     ```

2. Bağımlılıkları, bağımlılıkları yükleme bölümünde açıklandığı şekilde yükleyin.
3. Wazuh'un son sürümünü indirin ve çıkarın:

   ```
   curl -Ls https://github.com/wazuh/wazuh/archive/v4.9.2.tar.gz | tar zx
   ```

4. Wazuh dizinine geçmek için aşağıdaki komutları çalıştırın ve kullanılacak veritabanı türünü belirtin, `<DATABASE_TYPE>` değişkeni `mysql` veya `pgsql` ile değiştirin :

   ```
   cd wazuh-4.9.2/src
   make deps && make TARGET=server DATABASE=<DATABASE_TYPE>
   ```

   >

> Not: Sistem özelliklerinize bağlı olarak derleme işlemi biraz zaman alabilir.

5. Betiği çalıştırın `install.sh`. Wazuh kaynaklarını kullanarak kurulum sürecinde size rehberlik edecek bir sihirbaz görüntüler:

```
cd ..
./install.sh
```

6. Script size ne tür bir kurulum istediğinizi sorduğunda `manager`Wazuh yöneticisini kurmak için şunu yazın:

```
1- What kind of installation do you want (manager, agent, local, hybrid, or help)? manager
```

> Not: Kurulum sırasında kurulum yoluna karar verebilirsiniz. `install.sh` dosyasını çalıştırın ve dili seçin, kurulum modunu `manager` olarak ayarlayın, ardından kurulum yolunu ayarlayın (`Choose where to install Wazuh [/var/ossec]`/var/ossec). Varsayılan kurulum yolu /var/ossec'tir. Yaygın olarak kullanılan özel bir yol `/opt` olabilir.

> Uyarı: Varsayılandan farklı bir yol seçerseniz kritik bir kurulum dizini seçmemeye son derece dikkat edin. Dizin zaten mevcutsa, yükleyici dizini silmenizi veya Wazuh'u içine kurarak devam etmenizi isteyecektir.

7. Kurulum programı kurulumun sonunda Wazuh'u başlatmak isteyip istemediğinizi sorar. Eğer istemezseniz, aşağıdaki komutla daha sonra başlatabilirsiniz:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

Veritabanı Yapılandırması

Veritabanı sisteminize göre yeni bir veritabanı oluşturun, veritabanı kullanıcısını ayarlayın ve `src/os_dbd`kaynak kodun bulunduğu dizinde bulunan şemayı aşağıdaki komutlarla ekleyin:

- **MySQL için** :

```
mysql -u root -p
```

```
mysql> CREATE DATABASE Alerts_DB;
Query OK, 1 row affected (2.34 sec)

mysql> CREATE USER '<DATABASE_USER>'@'<DATABASE_SERVER_IP>' IDENTIFIED BY '<DATABASE_USER_
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT INSERT,SELECT,UPDATE,CREATE,DELETE,EXECUTE on Alerts_DB.* to '<DATABASE_USER>'@'
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
```

Yukarıdaki komutlarda aşağıdaki değişkenleri değiştirin:

- `<DATABASE_USER>`Veritabanı sunucusu için oluşturmak istediğiniz kullanıcıyla.
- `<DATABASE_SERVER_IP>`veritabanı sunucusunun IP adresi ile.
- `<DATABASE_USER_PASSWORD>`veritabanı sunucusuna erişmek için kullanıcı şifresi ile.

```
mysql -u root -p Alerts_DB < src/os_dbd/mysql.schema
```

- **PostgreSQL için** :

```
sudo -u postgres createuser -P <DATABASE_USER>
sudo -u postgres createdb -O <DATABASE_USER> Alerts_DB
psql -U <DATABASE_USER> -d Alerts_DB -f src/os_dbd/postgresql.schema
```

`<DATABASE_USER>`Veritabanı sunucusu için oluşturmak istediğiniz kullanıcıyla değiştirin .

> Not: Kullanıcıyı oluştururken iki kez parola girmeniz istenecektir. Wazuh yöneticisini yapılandırırken gerekli olduğundan bu parolayı not edin.

Wazuh Yöneticisi Yapılandırması

Wazuh yöneticisini veritabanı sistemine uyarılar ve diğer verileri gönderecek şekilde yapılandırmak için aşağıdaki adımları izleyin.

1. Wazuh sunucusundaki dosya `<ossec_config>`bloğunun içine aşağıdaki kod bloğunu ekleyin : `/var/ossec/etc/ossec.conf`

- **MySQL için** :

```
<database_output>
  <hostname><DATABASE_SERVER_IP></hostname>
  <username><DATABASE_USER></username>
  <password><DATABASE_USER_PASSWORD></password>
  <database>Alerts_DB</database>
  <type>mysql</type>
</database_output>
```

- **PostgreSQL için** :

```
<database_output>
  <hostname><DATABASE_SERVER_IP></hostname>
  <username><DATABASE_USER></username>
  <password><DATABASE_USER_PASSWORD></password>
  <database>Alerts_DB</database>
  <type>postgresql</type>
</database_output>
```

Nerede:

- `<hostname>`veritabanı sunucusunun IP adresini belirtir. `<DATABASE_SERVER_IP>` Veritabanı sunucusunun IP adresini değiştirin.
- `<username>`veritabanına erişecek kullanıcıyı belirtir. `<DATABASE_USER>`Yukarıda oluşturulan veritabanı kullanıcısıyla değiştirin.
- `<password>`veritabanına erişmek için kullanıcı parolasını belirtir. `<DATABASE_USER_PASSWORD>`Yukarıda oluşturulan kullanıcı parolasıyla değiştirin.
- `<database>`uyarıların depolanacağı veritabanının adını belirtir. Örneğin, `Alerts_DB` yukarıdaki yapılandırmada belirtildiği gibi.
- `<type>`veritabanının türünü belirtir (MySQL veya PostgreSQL). İzin verilen değerler `mysql`veya `pgsql`.

2. Değişiklikleri uygulamak için Wazuh yönetici hizmetini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

3. Wazuh yöneticisinin veritabanına bağlı olduğunu doğrulamak için aşağıdaki komutu çalıştırın:

```
grep wazuh-dbd /var/ossec/logs/ossec.log
```

Output

```
2024/06/24 14:49:11 wazuh-dbd: INFO: Connected to database 'Alerts_DB' at '127.0.0.1'.
```

Veritabanı artık Wazuh yöneticisinden veri almaya başlayacaktır.

Olay Günlüğü Tutma

Günlükler, Wazuh aracılarından, harici API'lerden ve ağ cihazlarından alınan ham olaylardır. Wazuh sunucusu tüm günlükleri süresiz olarak depolar. Alan optimizasyonunu en üst düzeye çıkarmak için Wazuh yöneticisi günlük dosyalarını otomatik olarak sıkıştırır.

Wazuh, iki tür günlüğü yönetir, Wazuh sunucusundan gelen dahili günlükler ve izlenen uç noktalardan gelen harici günlükler. Bu günlükler `/var/ossec/logs/`Wazuh sunucusunun dizininde süresiz olarak saklanır.

Aşağıdaki tabloda Wazuh sunucusundaki günlük dosyaları ve bunların saklanma yerleri açıklanmaktadır.

Günlük depolama dosyası	Günlük kaynağı	Tanım
`/var/ossec/logs/ossec.log`	Dahili	Wazuh sunucusu tarafından oluşturulan tüm bilgi düzeyindeki günlükleri depolar.
`/var/ossec/logs/api.log`	Dahili	Wazuh uygulamasının Wazuh sunucu API'leriyle etkileşimi sırasında oluşturulan günlükleri depolar.
`/var/ossec/logs/cluster.log`	Dahili	Wazuh kümesinin faaliyetleri tarafından oluşturulan günlükleri depolar.
`/var/ossec/logs/integrations.log`	Dahili	Üçüncü taraf uygulamalar ve sistemlerle arayüz oluştururken Wazuh entegrasyon modülü tarafından oluşturulan günlükleri depolar.
`/var/ossec/logs/active-responses.log`	Dahili	Wazuh Active Response modülü tarafından oluşturulan günlükleri depolar.
`/var/ossec/logs/firewall/firewall.log`	Dahili	Güvenlik duvarı tarafından oluşturulan günlükleri depolar.
`/var/ossec/logs/archives/archives.log`	Harici	Üçüncü taraf uygulama ve sistemlerden alınan günlükleri düz metin olarak depolar.
`/var/ossec/logs/archives/archives.json`	Harici	Üçüncü taraf uygulamalardan ve sistemlerden alınan günlükleri JSON biçiminde depolar.

Günlük Sıkıştırma ve Döndürme

Günlük dosyaları bir sistemde önemli disk alanı biriktirebilir ve tüketebilir. Bunu önlemek için Wazuh yöneticisi, günlükleri döndürme işlemi sırasında sıkıştırarak disk kullanımını verimli bir şekilde yönetmeye ve sistem performansını korumaya yardımcı olur. Wazuh yöneticisi günlük dosyalarını günlük olarak veya belirli bir eşiğe (dosya boyutu, yaş, zaman ve daha fazlası) ulaştıklarında sıkıştırır ve arşivler. Günlük döndürme işleminde Wazuh, sürekli olarak yeni olaylar

yazmak için orijinal adla yeni bir günlük dosyası oluşturur.

`/var/ossec/logs/`Günlük dosyaları günlük olarak sıkıştırılır ve MD5, SHA1 ve SHA256 karma algoritmaları kullanılarak dijital olarak imzalanır. Sıkıştırılmış günlük dosyaları, aşağıdaki biçime göre isimler taşıyan iç içe dizinler içindeki dizinde saklanır :

- Orijinal günlük dosyasının adını belirten .`log file name`
- `year`İçinde bulunulan yılın adını belirten .
- `month`Yılın o anki ayının adını belirten .

Örneğin, `/var/ossec/logs/archives/archives.log`sıkıştırılmış bir dosya dizinde saklanır . Aşağıdaki komutu çalıştırarak dizinin içeriğini görebilirsiniz:`13th APR, 2024`…`/archives/2024/Apr/`

```
ls -la /var/ossec/logs/archives/2024/Apr/
```

Output

```
total 0
drwxr-x--- 2 wazuh wazuh 62 Apr 17 08:15 .
drwxr-x--- 4 wazuh wazuh 28 Apr 12 07:30 ..
-rw-r----- 1 wazuh wazuh  0 Apr 13 00:00 ossec-archive-13.log.gz
-rw-r----- 1 wazuh wazuh  0 Apr 13 00:00 ossec-archive-13.log.sum
```

Yukarıdaki çıktıda görüldüğü gibi, sıkıştırılmış dosyanın adına ve onun sağlama toplamına sırasıyla dize ve sonek eklenir.

Yukarıdaki çıktıda görüldüğü gibi, sıkıştırılmış dosyanın adının ve sağlama toplamının başına `ossec` dizesi ve `day of the current month` son eki sırasıyla eklenir ve eklenir.

İhtiyaçlarınıza bağlı olarak, sıkıştırılmış dosyaları belirli bir süre sonra kaldırılmak üzere yapılandırabilirsiniz. Ayrıca, daha uzun süreli saklama için günlük yönetim sistemlerine, yedekleme sunucularına veya bulut tabanlı depolama aygıtlarına taşıyabilirsiniz.

Olay Günlüklerinin Arşivlenmesi

Olaylar, uygulamalar, uç noktalar ve ağ cihazları tarafından oluşturulan günlüklerdir. Wazuh sunucusu, bir kuralı tetikleyip tetiklemediklerine bakılmaksızın aldığı tüm olayları depolar. Bu olaylar, `/var/ossec/logs/archives/archives.log`ve adresinde bulunan Wazuh arşivlerinde depolanır `/var/ossec/logs/archives/archives.json`. Güvenlik ekipleri, güvenlik olaylarının geçmiş verilerini incelemek, eğilimleri analiz etmek ve tehditleri avlamak için raporlar oluşturmak amacıyla arşivlenmiş günlükleri kullanır.

Varsayılan olarak, Wazuh arşivleri devre dışıdır çünkü günlükleri Wazuh sunucusunda süresiz olarak depolar. Etkinleştirildiğinde, Wazuh yöneticisi uyumluluk ve adli amaçlar için güvenlik verilerini depolamak ve saklamak üzere arşivlenmiş dosyalar oluşturur.

Not

Wazuh arşivleri, izlenen tüm uç noktalardan toplanan günlükleri tutar, bu nedenle zamanla Wazuh sunucusunda önemli depolama kaynakları tüketir. Bu nedenle, bunları etkinleştirmeden önce disk alanı ve performans üzerindeki etkiyi göz önünde bulundurmak önemlidir.

Arşivlemeyi Etkinleştirme

Wazuh sunucunuzda arşivlemeyi etkinleştirmek için aşağıdaki adımları izleyin.

1. Wazuh yöneticisi yapılandırma dosyasını düzenleyin `/var/ossec/etc/ossec.conf`ve aşağıda vurgulanan alanların değerini şu şekilde ayarlayın `yes`:

   ```
   <ossec_config>
     <global>
       <jsonout_output>yes</jsonout_output>
       <alerts_log>yes</alerts_log>
       <logall>yes</logall>
       <logall_json>yes</logall_json>

       ...
   </ossec_config>
   ```

   Nerede:
   - `<logall>`tüm günlük iletilerinin arşivlenmesini etkinleştirir veya devre dışı bırakır. Etkinleştirildiğinde, Wazuh sunucusu günlükleri bir syslog biçiminde depolar. İzin verilen değerler `yes`ve ' dir `no`.
   - `<logall_json>`olayların günlüğe kaydedilmesini etkinleştirir veya devre dışı bırakır. Etkinleştirildiğinde, Wazuh sunucusu olayları bir JSON biçiminde depolar. İzin verilen değerler `yes`ve ' dir `no`.

   İstediğiniz biçime bağlı olarak, vurgulanan alanlardan bir veya her iki değeri de olarak ayarlayabilirsiniz `yes`. Ancak, yalnızca bu `<logall_json>yes</logall_json>`seçenek Wazuh panosundaki olayları görselleştirmek için kullanılabilecek bir dizin oluşturmanıza olanak tanır.

2. Yapılandırma değişikliklerini uygulamak için Wazuh yöneticisini yeniden başlatın:

   ```
   systemctl restart wazuh-manager
   ```

Seçtiğiniz formata bağlı olarak, dosya `archives.log`, , veya her ikisi de Wazuh sunucusundaki dizinde `archives.json`oluşturulacaktır .`/var/ossec/logs/archives/`

Wazuh varsayılan bir günlük döndürme politikası kullanır. Günlükleri günlük, aylık ve yıllık bazda döndürerek ve sıkıştırarak kullanılabilir disk alanının korunmasını sağlar.

Dashboard'daki Olayların Görselleştirilmesi

1. Filebeat yapılandırma dosyasını düzenleyin ve from `/etc/filebeat/filebeat.yml`değerini şu şekilde değiştirin :`archives: enabledfalsetrue`

```
archives:
 enabled: true
```

2. Yapılandırma değişikliklerini uygulamak için Filebeat'i yeniden başlatın:

```
systemctl restart filebeat
```

Wazuh Dashboard

1. Ana menüyü açmak için sol üst menü simgesine tıklayın. **Pano yönetimi'ni genişletin ve Pano yönetimi** > **Dizin desenleri'ne** gidin . Sonra, **Dizin deseni oluştur'a** tıklayın . Dizin deseni adı olarak kullanın `wazuh-archives-*`ve **Zaman alanı** açılır listesinde `timestamp` ayarlayın .
   Aşağıdaki GIF, endeks deseninin nasıl oluşturulacağını göstermektedir.

   wazuh-archives-* dizin deseninin oluşturulması
   Image not found or type unknown
2. Gösterge tablosundaki etkinlikleri görüntülemek için sol üst menü simgesine tıklayın ve **Keşfet'e** gidin . Dizin desenini olarak değiştirin `wazuh-archives-*`.

   Gösterge tablosunda etkinlikleri görüntüleyin
   Image not found or type unknown

Use Case: İmzalanmış İkili Proxy Yürütmeyi Algılama

`T1218.010`İmzalanmış ikili proxy yürütme, tehdit aktörlerinin kötü amaçlı kod çalıştırmak için güvenilir ikili dosyaları kullanarak uygulama beyaz listesini atlatmak için kullandıkları bir tekniktir. Bu teknik , MITRE ATT&CK çerçevesine dayalı olarak tanımlanmıştır .

Bu kullanım örneğinde, `regsvr32.exe`uygulama denetimlerini atlatmak için Windows yardımcı programı 'nın nasıl kötüye kullanılacağını gösteriyoruz. Daha sonra bu teknikle ilgili şüpheli etkinliği tespit etmek için Wazuh arşivlerindeki olayları analiz ediyoruz.

Windows 11 Yapılandırması

Windows 11 uç noktasına Sysmon ve Atomic Red Team'i (ART) yüklemek ve imzalanmış ikili proxy yürütme tekniğini taklit etmek için aşağıdaki adımları uygulayın.

Sysmon Entegrasyonu

Windows 11 uç noktasına Sysmon'ı yüklemek ve yapılandırmak için aşağıdaki adımları uygulayın.

1. Sysmon'ı Microsoft Sysinternals sayfasından indirin .
2. Sysmon yapılandırma dosyasını indirin: sysmonconfig.xml .
3. İndirilen yapılandırma dosyasıyla PowerShell'i yönetici olarak kullanarak Sysmon'u yükleyin:

```
> .\sysmon64.exe -accepteula -i .\sysmonconfig.xml
```

4. Sysmon günlüklerinin toplanacağı konumu belirtmek için `<ossec_config>`Wazuh aracı dosyasına blok içinde aşağıdaki yapılandırmayı ekleyin :`C:\Program Files (x86)\ossec-agent\ossec.conf`

```
<localfile>
  <location>Microsoft-Windows-Sysmon/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>
```

5. Değişiklikleri uygulamak için Wazuh aracısını yeniden başlatın ve aşağıdaki PowerShell komutunu yönetici olarak çalıştırın:

```
> Restart-Service -Name Wazuh
```

Atomic Red Team Kurulumu

PowerShell'i yönetici olarak kullanarak Windows 11 uç noktasına Atomic Red Team PowerShell modülünü yüklemek için aşağıdaki adımları uygulayın.

1. Varsayılan olarak, PowerShell çalışan betiklerin yürütülmesini kısıtlar. Varsayılan yürütme politikasını şu şekilde değiştirmek için aşağıdaki komutu çalıştırın `RemoteSigned`:

```
> Set-ExecutionPolicy RemoteSigned
```

2. ART yürütme çerçevesini yükleyin:

```
> IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/invoke-atomicredteam/master/install-atomicredt
> Install-AtomicRedTeam -getAtomics
```

3. Fonksiyonu kullanmak için ART modülünü içe aktarın `Invoke-AtomicTest`:

```
> Import-Module "C:\AtomicRedTeam\invoke-atomicredteam\Invoke-AtomicRedTeam.psd1" -Force
```

4. `Invoke-AtomicTest`Tekniğin ayrıntılarını göstermek için fonksiyonu kullanın `T1218.010`:

```
> Invoke-AtomicTest T1218.010 -ShowDetailsBrief
```

Output

```
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

T1218.010-1 Regsvr32 local COM scriptlet execution
T1218.010-2 Regsvr32 remote COM scriptlet execution
T1218.010-3 Regsvr32 local DLL execution
T1218.010-4 Regsvr32 Registering Non DLL
T1218.010-5 Regsvr32 Silent DLL Install Call DllRegisterServer
```

Saldırı Emülasyonu

Windows 11 uç noktasında imzalı ikili proxy yürütme tekniğini taklit edin.

1. Testi gerçekleştirmek için aşağıdaki komutu Powershell'i yönetici olarak çalıştırın `T1218.010`:

```
> Invoke-AtomicTest T1218.010
```

Output

```
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Executing test: T1218.010-1 Regsvr32 local COM scriptlet execution
Done executing test: T1218.010-1 Regsvr32 local COM scriptlet execution
Executing test: T1218.010-2 Regsvr32 remote COM scriptlet execution
Done executing test: T1218.010-2 Regsvr32 remote COM scriptlet execution
Executing test: T1218.010-3 Regsvr32 local DLL execution
Done executing test: T1218.010-3 Regsvr32 local DLL execution
Executing test: T1218.010-4 Regsvr32 Registering Non DLL
Done executing test: T1218.010-4 Regsvr32 Registering Non DLL
Executing test: T1218.010-5 Regsvr32 Silent DLL Install Call DllRegisterServer
Done executing test: T1218.010-5 Regsvr32 Silent DLL Install Call DllRegisterServer
```

Exploitin başarılı bir şekilde yürütülmesinin ardından birkaç hesap makinesi örneği açılacaktır.

Wazuh Dashboard

Wazuh arşivlerini, avlanan teknikle ilgili olayları sorgulamak ve görüntülemek için kullanın. Arşivlere danışırken bazı olayların Wazuh panosunda uyarı olarak yakalanmış olabileceğini unutmamak önemlidir. Algılama yapılmayan uyarılar ve olaylar dahil olmak üzere Wazuh arşivlerinden gelen bilgileri kullanarak özel gereksinimlerinize göre özel kurallar oluşturabilirsiniz.

1. Testin gerçekleştirildiği son beş dakika içinde meydana gelen olayları görüntülemek için bir zaman aralığı filtresi uygulayın. `agent.id`, `agent.ip`veya kullanarak belirli Windows uç noktasından günlükleri görüntülemek için filtre uygulayın `agent.name`.

   Zaman aralığı filtresi uygulanıyor
   Image not found or type unknown

   Daha önceki saldırı emülasyonuyla bir korelasyon belirlemek için inceleyebileceğiniz birden fazla isabet vardır. Örneğin, test sırasında Windows uç noktasında gözlemlenene benzer bir hesap makinesi oluşturma olayı fark edebilirsiniz.

   Hesap makinesi yumurtlama olayı
   Image not found or type unknown
2. `regsvr32`Olaylarla ilgili işlemleri kolaylaştırmak ve araştırmak için arama çubuğuna yazın `regsvr32`.

   Filtre regsvr32
   Image not found or type unknown
3. İlgili alanları görüntülemek için herhangi bir olayı genişletin.

   Etkinlikleri genişlet
   Image not found or type unknown
4. Arşivlenmiş günlüklerin JSON formatını görüntülemek için JSON sekmesine tıklayın.

   JSON sekmesi
   Image not found or type unknown

   Komutlar, hizmetler, yollar ve daha fazlası gibi etkinliklere ilişkin belirli ayrıntıları JSON günlüğünden çıkarabilir ve doğrulayabilirsiniz. Aşağıda, ilk işlem oluşturmayı ve yürütülen komutla ilgili öznitelikleri tanımlayabilirsiniz:


```
"data": {
    "win": {
      "eventdata": {
        "originalFileName": "REGSVR32.EXE",
        "image": "C:\\\\Windows\\\\SysWOW64\\\\regsvr32.exe",
        "product": "Microsoft® Windows® Operating System",
        "parentProcessGuid": "{45cd4aff-35fc-6463-6903-000000001300}",
        "description": "Microsoft(C) Register Server",
        "logonGuid": "{45cd4aff-2ce5-6463-2543-290000000000}",

       "parentCommandLine": "C:\\\\Windows\\\\system32\\\\regsvr32.exe  /s /i C:\\\\AtomicRedTeam\\\\atomics
```

```
        "processGuid": "{45cd4aff-35fc-6463-6a03-000000001300}",
        "logonId": "0x294325",
        "parentProcessId": "7652",
        "processId": "4064",
        "currentDirectory": "C:\\\\Users\\\\THECOT~1\\\\AppData\\\\Local\\\\Temp\\\\",
        "utcTime": "2023-05-16 07:51:24.512",
        "hashes": "SHA1=8E2C6B7F92A560E0E856F8533D62A1B10797828F,MD5=5F7264BD237FAEA46FB24
        "parentImage": "C:\\\\Windows\\\\System32\\\\regsvr32.exe",
        "ruleName": "technique_id=T1117,technique_name=Regsvr32",
        "company": "Microsoft Corporation",
        "commandLine": "  /s /i C:\\\\AtomicRedTeam\\\\atomics\\\\T1218.010\\\\bin\\\\AllTheThingsx86.dll",
        "integrityLevel": "High",
        "fileVersion": "10.0.22621.1 (WinBuild.160101.0800)",
        "user": "Windows11\\\\Testuser",
        "terminalSessionId": "2",
        "parentUser": "Windows11\\\\Testuser"
      },
      "system": {
        "eventID": "1",
        "keywords": "0x8000000000000000",
        "providerGuid": "{5770385f-c22a-43e0-bf4c-06f5698ffbd9}",
        "level": "4",
        "channel": "Microsoft-Windows-Sysmon/Operational",
        "opcode": "0",

        "message": "\"Process Create:\r\nRuleName: technique_id=T1117,technique_name=Regsvr32\r\nUtcTir

        "version": "5",
        "systemTime": "2023-05-16T07:51:24.5131006Z",
        "eventRecordID": "88509",
        "threadID": "3960",
        "computer": "Windows11",
        "task": "1",
        "processID": "3156",
        "severityValue": "INFORMATION",
        "providerName": "Microsoft-Windows-Sysmon"
      }
    }
  },
```


Diğer ilgili olaylar üzerinde daha fazla araştırma yaparak, regsvr32 yardımcı programı tarafından oluşturulan bir işlem enjeksiyon olayını ve yüklenen görüntüyü görebilirsiniz:


```
"data": {
    "win": {
      "eventdata": {
        "originalFileName": "mscoree.dll",
        "image": "C:\\\\Windows\\\\SysWOW64\\\\regsvr32.exe",
        "product": "Microsoft® Windows® Operating System",
        "signature": "Microsoft Windows",

        "imageLoaded": "C:\\\\Windows\\\\SysWOW64\\\\mscoree.dll",
```

```
        "description": "Microsoft .NET Runtime Execution Engine",
        "signed": "true",
        "signatureStatus": "Valid",
        "processGuid": "{45cd4aff-35fc-6463-6a03-000000001300}",
        "processId": "4064",
        "utcTime": "2023-05-16 07:51:24.774",
        "hashes": "SHA1=52A6AB3E468C4956C00707DF80C7609EEE74D9AD,MD5=BEE4D173DA78E4D3AC9E
        "ruleName": "technique_id=T1055,technique_name=Process Injection",
        "company": "Microsoft Corporation",
        "fileVersion": "10.0.22621.1 (WinBuild.160101.0800)",
        "user": "Windows11\\\\Testuser"
      },
      "system": {
        "eventID": "7",
        "keywords": "0x8000000000000000",
        "providerGuid": "{5770385f-c22a-43e0-bf4c-06f5698ffbd9}",
        "level": "4",
        "channel": "Microsoft-Windows-Sysmon/Operational",
        "opcode": "0",

        "message": "\"Image loaded:\r\nRuleName: technique_id=T1055,technique_name=Process Injection\r\n

        "version": "3",
        "systemTime": "2023-05-16T07:51:24.7768916Z",
        "eventRecordID": "88510",
        "threadID": "3960",
        "computer": "Windows11",
        "task": "7",
        "processID": "3156",
        "severityValue": "INFORMATION",
        "providerName": "Microsoft-Windows-Sysmon"
      }
    }
  },
```


5. `data.win.eventdata.ruleName:technique_id=T1218.010,technique_name=Regsvr32` Teknik kimliğini görmek için aşağıda gösterilen filtreyi uygulayın.

Filtre T1218.010 tekniği

Image not found or type unknown

6. İlgili alanları görüntülemek için olayı genişletin.

Filtre T1218.010 tekniği

Image not found or type unknown

7. Arşivlenmiş günlüklerin JSON formatını görüntülemek için JSON sekmesine tıklayın.

JSON sekmesi

Image not found or type unknown

Aşağıdaki kayıttan, olayı analiz etmeyi kolaylaştıran daha yapılandırılmış ayrıntılar çıkarabilirsiniz:

```
"data": {
    "win": {
      "eventdata": {
        "destinationPort": "443",
        "image": "C:\\\\Windows\\\\System32\\\\regsvr32.exe",
        "sourcePort": "63754",
        "initiated": "true",
        "destinationIp": "1.1.123.23",
        "protocol": "tcp",
        "processGuid": "{45cd4aff-36b5-645a-9e07-000000000e00}",
        "sourceIp": "192.168.43.16",
        "processId": "4704",
        "utcTime": "2023-05-09 21:19:25.361",

       "ruleName": "technique_id=T1218.010,technique_name=Regsvr32",

       "destinationIsIpv6": "false",
        "user": "Windows11\\\\Testuser",
        "sourceIsIpv6": "false"
      },
      "system": {
        "eventID": "3",
        "keywords": "0x8000000000000000",
        "providerGuid": "{5770385f-c22a-43e0-bf4c-06f5698ffbd9}",
        "level": "4",
        "channel": "Microsoft-Windows-Sysmon/Operational",
        "opcode": "0",

       "message": "\"Network connection detected:\r\nRuleName: technique_id=T1218.010,technique_name=

       "version": "5",
        "systemTime": "2023-05-09T12:04:07.0231156Z",
        "eventRecordID": "63350",
        "threadID": "3096",
        "computer": "Windows11",
        "task": "3",
        "processID": "3156",
        "severityValue": "INFORMATION",
        "providerName": "Microsoft-Windows-Sysmon"
      }
    }
  },
```


Algılama mantığını geliştirmek ve özel kod çözücüler ve kurallar yazmak için Wazuh arşivlerinden gelen olayları kullanabilirsiniz. Ayrıca `wazuh-logtest`, kuralları sağlanan günlüklere göre test etmek ve doğrulamak için hazır aracı da kullanabilirsiniz.

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>0`uyarıları 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

Değişiklik kılavuzunu düzenleyin

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.

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 .

Indexer Entegrasyonu

Dizinleyici entegrasyonu, verileri Wazuh yöneticisinden Wazuh dizinleyicisine veya üçüncü taraf dizinleyicilere ileten veri ileticilerini tanımlar.

Wazuh Indexer

Bu entegrasyon, Wazuh yöneticisi ile Wazuh dizinleyicisi arasında bir köprü sağlar. Verileri dizinleme için Wazuh yöneticisinden Wazuh dizinleyicisine iletir. Wazuh dizinleyici entegrasyonu iki ileticiden oluşur: Filebeat ve Wazuh dizinleyici bağlayıcısı .

Filebeat

Bu bileşen, Wazuh yöneticisi tarafından işlenen uyarıları ve arşivlenmiş olayları indeksleme ve depolama için Wazuh indeksleyicisine güvenli bir şekilde iletmek üzere tasarlanmış hafif bir veri taşıyıcısıdır. Wazuh analiz motorunun çıktısını okur ve olayları gerçek zamanlı olarak gönderir.

Yapılandırma

Aşağıdaki kod bloğu, Wazuh sunucu dosyasındaki varsayılan Filebeat yapılandırmasını gösterir `/etc/filebeat/filebeat.yml`. Bu yapılandırma dosyası, adım adım Wazuh sunucu kurulumu gerçekleştirilirken indirilir. Filebeat'i nasıl indireceğinizi, yapılandıracağınızı ve yükleyeceğinizi öğrenmek için, belgelerdeki Filebeat'i yapılandırma bölümüne bakın.

```
# Wazuh - Filebeat configuration file
output.elasticsearch.hosts:
       - 127.0.0.1:9200
#        - <elasticsearch_ip_node_2>:9200
#        - <elasticsearch_ip_node_3>:9200

output.elasticsearch:
  protocol: https
  username: ${username}
  password: ${password}
  ssl.certificate_authorities:
    - /etc/filebeat/certs/root-ca.pem
  ssl.certificate: "/etc/filebeat/certs/wazuh-server.pem"
  ssl.key: "/etc/filebeat/certs/wazuh-server-key.pem"
setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
```

```
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false

filebeat.modules:
  - module: wazuh
    alerts:
      enabled: true
    Archives:

logging.level: info
logging.to_files: true
logging.files:
  path: /var/log/filebeat
  name: filebeat
  keepfiles: 7
  permissions: 0644

logging.metrics.enabled: false

seccomp:
  default_action: allow
  syscalls:
  - action: allow
    names:
    - rseq
```

Nerede:

- `<output.elasticsearch.hosts>`bağlanılacak Wazuh dizinleyici düğümlerinin listesini belirtir. IP adreslerini veya ana bilgisayar adlarını kullanabilirsiniz. Varsayılan olarak, ana bilgisayar localhost, olarak ayarlanmıştır `127.0.0.1:9200`. Bunu uygun şekilde Wazuh dizinleyici adresinizle değiştirin. Birden fazla Wazuh dizinleyici düğümünüz varsa adresleri virgül kullanarak ayırabilirsiniz.
- `<protocol>`bağlantı için kullanılacak protokolü belirtir. Varsayılan değer 'dir `https`. İzin verilen değerler `http`ve 'dir `https`.
- `<username>`ve `<password>`Wazuh indeksleyicisine güvenli bir şekilde kimlik doğrulaması yapmak için kullanılan ortam değişkenini belirtir.
- `<ssl.certificate_authorities>`HTTPS sunucu doğrulamaları için kök sertifikalarına giden yolu belirtir. Varsayılan değer 'dir `/etc/filebeat/certs/root-ca.pem`. Olası değer herhangi bir geçerli yoldur
- `<ssl.certificate>`Filebeat SSL sertifikasına giden yolu belirtir. Varsayılan değer 'dir `/etc/filebeat/certs/wazuh-server.pem`. Olası değer herhangi bir geçerli yoldur.
- `<ssl.key>`Filebeat tarafından kullanılan SSL anahtarının yolunu belirtir. Varsayılan değer 'dir `/etc/filebeat/certs/wazuh-server-key.pem`. Olası değer herhangi bir geçerli yoldur.
- `<setup.template.json.enabled>`özel şablonların kullanımını etkinleştirir veya devre dışı bırakır. Varsayılan değer `true`.
- `<setup.template.json.path>`şablon JSON dosyasına giden dosya yolunu belirtir. Varsayılan değer 'dir `/etc/filebeat/wazuh-template.json`. Olası değer herhangi bir geçerli yoldur.
- `<setup.template.json.name>`şablonun adını tanımlar. Varsayılan değer `wazuh`.

- `<setup.ilm.overwrite>`olarak ayarlandığında `true`, yaşam döngüsü ilkesi başlangıçta üzerine yazılır. Varsayılan değer 'dir `true`.
- `<setup.ilm.enabled>`oluşturulan herhangi bir yeni endekste endeks yaşam döngüsü yönetimini etkinleştirir veya devre dışı bırakır. Varsayılan değer 'dir `false`. Olası geçerli değerler `true`ve ' dir `false`.
- `<filebeat.modules>`Filebeat'in kullanacağı modülleri belirtir.
- `<module>`kullanılacak modülü tanımlar. Varsayılan değer `wazuh`.
- `<alerts>`uyarıların Wazuh dizinleyicisine iletilmesini etkinleştirir veya devre dışı bırakır. Yapılandırma seçeneği olarak ayarlandığında `<enabled>`, `true`uyarılar Wazuh dizinleyicisine iletilir.
- `<archives>`Arşiv günlüklerinin işlenip işlenmeyeceğini ve iletileceğini belirleyen yapılandırmaları belirtir.
- `<logging.level>`günlük düzeyini tanımlar. Varsayılan değer, `info`bilgi günlüklerini temsil eder. Diğer günlük düzeyleri `debug`, `error`, ve 'dir `warning`.
- `<logging.to_files>`dosyalara günlük kaydını etkinleştirir veya devre dışı bırakır. Varsayılan değer 'dir `true`. olarak ayarlandığında `true`, filebeat tüm günlükleri bir dosyaya yazar.
- `<logging.files.path>`günlük dosyalarının saklanacağı dizini belirtir. Varsayılan günlük yolu `/var/log/filebeat`.
- `<logging.files.name>`günlüklerin depolandığı dosyanın adını belirtir. Varsayılan ad `filebeat`.
- `<logging.files.keepfiles>`saklanacak yakın zamanda döndürülen günlük dosyalarının sayısını belirtir. Varsayılan değer 'dir . İzin verilen değer ve `7`arasında bir tam sayıdır .`11024`
- `<logging.files.permissions>`günlük dosyaları için dosya izinlerini ayarlar. Varsayılan değer 'dir `0644`, bu da günlük dosyalarının sahibinin bunları okuyabileceği ve yazabileceği, diğerlerinin ise yalnızca okuyabileceği anlamına gelir.
- `<logging.metrics.enabled>`dahili ölçümlerin günlüğe kaydedilmesini etkinleştirir veya devre dışı bırakır. Varsayılan değer 'dir `true`. Olası değerler `true`ve ' dir `false`.
- `<seccomp>`filebeat işleminin yapabileceği sistem çağrılarının sayısını kısıtlayan bir secomp (güvenli bilgi işlem modu) politikası belirtir.
- `<default_action>`sistem çağrıları için varsayılan eylemi izin verecek şekilde ayarlar. Bu, syscalls listesinde açıkça belirtilmeyen herhangi bir sistem çağrısına varsayılan olarak izin verileceği anlamına gelir.
- `<syscalls>`sistem çağrısı adlarının ve karşılık gelen eylemlerin bir listesini tanımlar.
- `<action>`listelenen sistem çağrılarından herhangi biri `names`yürütüldüğünde gerçekleştirilecek eylemi belirtir. Varsayılan değer 'dir `allow`. Diğer değerler `errno`, `trace`, `trap` , `kill_thread`, `kill_process`, ve 'dir `log`.
- `<names>`sistem çağrısı adlarının bir listesini tanımlar. Listede en az bir sistem çağrısı tanımlanmalıdır. `rseq`(yeniden başlatılabilir diziler) sistem çağrısı, birden fazla iş parçacığında paylaşılan bellekte kullanıcı alanı işlemlerini hızlandırmak için kullanılır. `rseq` Sistem çağrısına bu yapılandırmada izin verilir.

Wazuh Indeksleyici Bağlayıcısı

Wazuh dizinleyici bağlayıcısı şu anda Wazuh yöneticisinden güvenlik açığı verilerini alıyor ve güvenli bir şekilde Wazuh dizinleyicisine iletiyor. Güvenlik açığı verilerini Elastic Common Schema'yı (ECS) takip eden JSON formatında alıyor ve veri tutarlılığı ve güvenilirliğini sağlamak için

durumunu Wazuh dizinleyicisiyle senkronize ediyor. Wazuh dizinleyici bağlayıcısı Wazuh yöneticisiyle birlikte gönderilir.

`/var/ossec/etc/ossec.conf`İndeksleyici bağlayıcısı için standart yapılandırma , Wazuh sunucusundaki dosyada aşağıda gösterildiği gibi belirtilmiştir :

```
<ossec_config>
 <indexer>
   <enabled>yes</enabled>
   <hosts>
    <host>https://127.0.0.1:9200</host>
   </hosts>
   <ssl>
    <certificate_authorities>
     <ca>/etc/filebeat/certs/root-ca.pem</ca>
    </certificate_authorities>
    <certificate>/etc/filebeat/certs/filebeat.pem</certificate>
    <key>/etc/filebeat/certs/filebeat-key.pem</key>
   </ssl>
 </indexer>
</ossec_config>
```

Nerede:

- `<indexer>`Wazuh indeksleyici bağlayıcısı için yapılandırma seçeneklerini belirtir.
- `<enabled>`Wazuh dizinleyici bağlayıcısını etkinleştirir veya devre dışı bırakır. Bu seçenek için izin verilen değerler ve'dir `yes`. `no`Değer, `yes`Wazuh dizinleyici bağlayıcısını etkinleştirir ve `no`devre dışı bırakır. Varsayılan değer 'dir `yes`.
- `<hosts>`bağlanılacak Wazuh dizinleyici düğümlerinin bir listesini belirtir. `host`Her düğüm bağlantısını ayarlamak için seçeneği kullanın.
- `<host>`bağlanılacak Wazuh dizinleyici düğüm URL'sini veya IP adresini belirtir. Örneğin, `http://172.16.1.11`veya `192.168.3.2:9230`. Varsayılan olarak, değer localhost ana bilgisayarına ayarlanır: `https://127.0.0.1:9200`.
- `<ssl>`SSL parametreleri için yapılandırma seçeneklerini belirtir.
- `<certificate_authorities>`doğrulama için kök sertifika dosya yollarının bir listesini belirtir. `ca`Her CA sertifika dosya yolunu ayarlamak için seçeneği kullanın.
- `<ca>`HTTPS sunucu doğrulamaları için kök CA sertifikasını belirtir. Varsayılan değer 'dir `/etc/filebeat/certs/root-ca.pem`. Olası değer herhangi bir geçerli CA sertifikasıdır.
- `<certificate>`Filebeat SSL sertifikasına giden yolu belirtir. Varsayılan değer 'dir `/etc/filebeat/certs/filebeat-key.pem`. Olası değer herhangi bir geçerli anahtardır.
- `<key>`kimlik doğrulama için kullanılan sertifika anahtarını belirtir. Varsayılan değer 'dir `/etc/filebeat/certs/filebeat-key.pem`. Olası değer herhangi bir geçerli anahtardır.

Referans kılavuzunun dizinleyici bölümünde mevcut yapılandırma seçenekleri hakkında daha fazla bilgi edinebilirsiniz .

Üçüncü Taraf Indexer

Wazuh yöneticisi uyarıları üçüncü taraf dizinleyicilere iletebilir. Wazuh yöneticilerini yalnızca günlük analizi için kullanıyorsanız ve uyarıları dizinleme ve depolama için üçüncü taraf çözümlere iletmek istiyorsanız, alternatif seçenekler mevcuttur. Wazuh, uyarıları istediğiniz çözüme aktarmak için her Wazuh yönetici düğümüne istediğiniz veri ileticisini yüklemenize olanak tanır. Şu anda Wazuh, aşağıdaki üçüncü taraf çözümler için belgeler sunmaktadır:

Çözüm	Tanım
ELK Stack	Wazuh yöneticisi uyarılarını Logstash kullanarak ELK Stack'e iletme.
OpenSearch	Wazuh yöneticisi uyarılarını Logstash kullanarak OpenSearch'e iletme.
Splunk	Wazuh yöneticisi uyarılarını Logstash kullanarak Splunk'a iletme.
	Splunk Evrensel Yönlendiriciyi kullanarak Wazuh sunucu uyarılarını Splunk'a iletme.

Bu seçenekler, Wazuh'u mevcut izleme ve analiz altyapınızla entegre etmede esneklik sağlar.

Wazuh Yöneticisi

Wazuh yöneticisi veri analizi ve uyarılardan sorumludur. Uyarıları syslog, e-postalar veya entegre harici API'ler aracılığıyla iletebilir. Wazuh'un veri analizini nasıl gerçekleştirdiği hakkında daha fazla bilgi için veri analizi belgelerine bakın.

Wazuh yöneticisi, çeşitli işlevlerden sorumlu olan çeşitli hizmetler ve bileşenlerden oluşur. Bunlara yeni Wazuh aracılarını kaydetme, güvenlik olaylarını toplama, günlükleri kod çözme, kuralları değerlendirme ve uyarı verme dahildir. Ayrıca Wazuh aracısının kimliklerini doğrulama ve Wazuh aracısı ile Wazuh sunucusu arasındaki iletişimleri şifreleme gibi diğer işlevlerden de sorumludur.

Acente Kayıt Hizmeti

Temsilci kayıt hizmeti, Wazuh temsilcilerini Wazuh yöneticisine kaydetmek için kullanılır. Kayıt hizmeti, Wazuh temsilcilerinin kaydını basitleştirir ve Wazuh yöneticisiyle güvenli bir şekilde iletişim kurmak üzere doğru bir şekilde kimlik doğrulaması yapılıp yapılandırılmalarını sağlar.

Bir uç noktaya bir Wazuh aracı yüklendiğinde ve başlatıldığında, kayıt sürecini başlatmak için otomatik olarak Wazuh yöneticisiyle iletişime geçer. Wazuh yöneticisi, Wazuh aracısıyla iletişimini şifreleyen benzersiz bir kimlik doğrulama anahtarı üretir. Kayıt süreci için parola kimlik doğrulaması, Wazuh yöneticisi kimlik doğrulaması ve Wazuh aracı kimlik doğrulaması gibi ek güvenlik önlemleri yapılandırabilirsiniz. Kayıt süreci hakkında daha fazla bilgi için Wazuh aracı kaydıyla ilgili belgelere bakın.

Yapılandırma

Aşağıdaki blok , Wazuh sunucusunun dosyasındaki `<auth>`varsayılan aracı kayıt hizmeti yapılandırmasıdır :`/var/ossec/etc/ossec.conf`

```
<auth>
  <disabled>no</disabled>
  <remote_enrollment>yes</remote_enrollment>
  <port>1515</port>
  <use_source_ip>no</use_source_ip>
  <force>
    <enabled>yes</enabled>
    <disconnected_time enabled="yes">1h</disconnected_time>
    <after_registration_time>1h</after_registration_time>
    <key_mismatch>yes</key_mismatch>
  </force>
  <purge>yes</purge>
```

```
  <use_password>no</use_password>
  <ciphers>HIGH:!ADH:!EXP:!MD5:!RC4:!3DES:!CAMELLIA:@STRENGTH</ciphers>
  <!-- <ssl_agent_ca></ssl_agent_ca> -->
  <ssl_verify_host>no</ssl_verify_host>
  <ssl_manager_cert>etc/sslmanager.cert</ssl_manager_cert>
  <ssl_manager_key>etc/sslmanager.key</ssl_manager_key>
  <ssl_auto_negotiate>no</ssl_auto_negotiate>
</auth>
```

Nerede:

- `<disabled>`Wazuh aracısının Wazuh yöneticisine kaydolma ve kimlik doğrulama işlemini etkinleştirir veya devre dışı bırakır. Varsayılan değer 'dir `no`. İzin verilen değerler `yes`ve 'dir `no`.
- `<remote_enrollment>`Wazuh yöneticisinin varsayılan olarak 1515 numaralı bağlantı noktasında TLS şifrelemesi kullanarak yeni Wazuh aracılarından gelen bağlantıları kabul etmesini sağlar. Varsayılan değer 'dir `yes`. İzin verilen değerler `yes`ve 'dir `no`.
- `<port>`Bağlantıları dinlemek için TCP bağlantı noktası numarasını belirtir. Varsayılan değer 'dir . İzin verilen değer ve `1515`arasındaki herhangi bir bağlantı noktası numarasıdır .`065535`
- `<use_source_ip>`İstemcinin kaynak IP adresinin mi yoksa "herhangi biri"nin mi kullanılacağını tanımlar. İzin verilen değerler ve'dir `yes`. `no`Değer hayır olduğunda, kayıt için kullanılan kaynak IP değişse bile Wazuh aracısı Wazuh yöneticisine bağlanabilir. Ancak değer evet olduğunda, kaynak IP adresi değişse bile Wazuh aracısı Wazuh yöneticisine bağlanamaz.
- `<force>`Wazuh aracısının etiketi içinde yeniden kaydı için yapılandırılacak seçenekleri belirtir. Yeniden kaydın başarılı olması için tüm koşulların karşılanması gerekir. Aşağıdaki seçenekler, seçeneğin ayarlarını tanımlar `force`:
  - `<enabled>`yinelenen bir ad veya IP adresi varsa bir Wazuh aracısının eklenmesinin zorlanıp zorlanmayacağını belirtir. Eğer öyleyse `enabled`, aynı ad veya IP adresine sahip eski Wazuh aracısını kaldıracaktır. Varsayılan değer 'dir `yes`. Olası değerler `yes` ve 'dir `no`.
  - `<disconnected_time>`yalnızca ayarda yapılandırılan değerden daha uzun süre bağlantısı kesilmiş olan Wazuh aracıları için bir değiştirme yapılıp yapılmayacağını belirtir. Varsayılan değer `1h`(bir saat)'tir. İzin verilen değer sıfırdan büyük veya sıfıra eşit herhangi bir sayıdır. `s`, `h`, `m`, ve gibi soneklerin `d`saniye, saat, dakika ve günü temsil etmesine izin verir. Öznitelik ayarı `enabled`varsayılan değerine sahiptir `yes`, yani değiştirme yalnızca belirtilen bağlantı kesme süresi aşıldıktan sonra gerçekleşir. Etkin özniteliğin `yes`ve olmak üzere iki olasılığı vardır `no`.
  - `<after_registration_time>`Wazuh aracısı değişiminin yalnızca Wazuh aracısı kaydının ayarda yapılandırılan değerden büyük olması durumunda gerçekleştirileceğini belirtir. Varsayılan değer 'dir `1h`. İzin verilen değer sıfırdan büyük veya ona eşit herhangi bir sayıdır. `s`, `h`, `m`, ve gibi soneklerin `d`saniye, saat, dakika ve günü temsil etmesine izin verir.
  - `<key_mismatch>`Wazuh aracısının elinde tuttuğu anahtar, yönetici tarafından kaydedilen anahtardan farklı olduğunda Wazuh aracısının değiştirilmesinin gerçekleştiğini tanımlar. Varsayılan değer 'dir `yes`. Olası değerler `yes`ve 'dir `no`.
- `<purge>`Wazuh aracıları kaldırıldığında istemci anahtarlarının silinip silinmeyeceğini belirtir. Değer olduğunda `no`, kaldırılan Wazuh aracıları kaldırılmış olarak işaretlenen istemci anahtarları dosyasında kalır. Değer olarak ayarlandığında `yes`, istemci anahtarları dosyası

temizlenir. Varsayılan değer 'dir `yes`. Olası değerler `yes`ve 'dir `no`.
- `<use_password>`paylaşımlı parola kimlik doğrulamasının kullanımını belirler. Değer olduğunda `no`, bu seçenek devre dışıdır. Değer olarak ayarlandığında `yes`, dosyadan paylaşımlı bir parola okunur `/var/ossec/etc/authd.pass`. Bu dosya mevcut değilse, rastgele bir parola oluşturulur ve `/var/ossec/logs/ossec.log`Wazuh sunucusundaki dosyada saklanır.
- `<ciphers>`SSL kullanarak ağ iletişimi için şifrelerin listesini ayarlar. Varsayılan değer `HIGH:!ADH:!EXP:!MD5:!RC4:!3DES:!CAMELLIA:@STRENGTH`.
- `<ssl_agent_ca>`istemcileri doğrulamak için kullanılan CA sertifikasına giden yolu belirtir. Wazuh kurulum dizini altındaki bağıl yol veya tam yol olarak adlandırılabilir. Olası değer herhangi bir geçerli yoldur.
- `<ssl_verify_host>`CA sertifikası belirtildiğinde kaynak ana bilgisayar doğrulamasını açar ve kapatır. İstemci kaynak IP adresi Ortak Ad alanı kullanılarak doğrulanacaktır. Varsayılan değer 'dir `no`. İzin verilen değerler `yes`ve 'dir `no`.
- `<ssl_manager_cert>`sunucu SSL sertifikasına giden yolu belirtir. Wazuh kurulum dizinindeki bağıl yol veya tam yol olarak adlandırılabilir. Varsayılan değer etc/sslmanager.cert'dir. Olası değer herhangi bir geçerli yoldur.
- `<ssl_manager_key>`sunucunun SSL anahtarına giden yolu belirtir. Wazuh kurulum dizininin altındaki bağıl yol veya tam yol olarak adlandırılabilir. Varsayılan değer etc/sslmanager.key'dir. Olası değer herhangi bir geçerli yoldur.
- `<ssl_auto_negotiate>`SSL/TLS yönteminin otomatik olarak seçilip seçilmeyeceğini değiştirir. Varsayılan olarak yalnızca TLS v1.2'ye izin verilir. olarak ayarlandığında `yes`, sistem istemciyle en güvenli ortak yöntemi müzakere eder. Yöneticinin TLS v1.2'yi desteklemediği eski sistemlerde, bu seçenek otomatik olarak etkinleştirilir. Varsayılan değer 'dir `no`. İzin verilen değerler `yes`ve 'dir `no`.

Yapılandırma dosyasında değişiklik yaptığınızda, aşağıdaki komutu kullanarak komut satırı arayüzü üzerinden Wazuh yöneticisini yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

Ajan Bağlantı Hizmeti

Aracı bağlantı hizmeti, kalıcı ve güvenli bir iletişim kanalı kurmak ve sürdürmek için Wazuh aracılarından gelen olayları dinler. Wazuh aracısı, güvenlik verilerini analiz için Wazuh yöneticisine göndermek için bu güvenli kanalı kullanır. Varsayılan olarak, hizmet `TCP`Wazuh aracısı ile Wazuh yöneticisi arasındaki iletişimi güvence altına almak için protokolü kullanır.

Yapılandırma

Aşağıdaki blok Wazuh sunucu yapılandırma dosyasındaki varsayılan bağlantı hizmeti yapılandırmasıdır `/var/ossec/etc/ossec.conf`:

```
<ossec_config>
  <remote>
    <connection>secure</connection>
    <port>1514</port>
    <protocol>tcp</protocol>
    <queue_size>131072</queue_size>
  </remote>
</ossec_config>
```

Nerede:

- `<connection>`kabul edilecek gelen bağlantının türünü belirtir. Varsayılan değer güvenlidir. İzin verilen değerler `secure`ve ' dir `syslog`.
- `<port>`olayları dinlemek için kullanılacak portu belirtir. Varsayılan port değeri `1514`güvenli bağlantı ve syslog bağlantısı içindir . İzin verilen değer ve `514`arasındaki herhangi bir port numarasıdır .`165535`
- `<protocol>`bağlantı için kullanılacak protokolü belirtir. Varsayılan değer 'dir `tcp`. İzin verilen değerler `tcp`ve 'dir `udp`.
- `<queue_size>`Uzak daemon kuyruğunun kapasitesini Wazuh aracı olaylarının sayısı olarak ayarlamanıza olanak tanır. Varsayılan değer 'dir . İzin verilen değer ile `131072`arasında bir tam sayıdır . Uzak kuyruk yalnızca Wazuh aracı olayları için kullanılabilir, syslog olayları için kullanılamaz. Bu seçenek yalnızca bağlantı güvenli olarak ayarlandığında çalışır. Bu yapılandırma ayarı hakkında daha fazla bilgi edinmek için Wazuh kuyruğu ile ilgili belgelerimize bakın .`1262144`

Değişiklikler yapıldıysa, değişiklikleri uygulamak için aşağıdaki komutla Wazuh yöneticisini komut satırı arayüzü üzerinden yeniden başlatın:

Systemd

```
systemctl restart wazuh-manager
```

SysV Başlatma

```
service wazuh-manager restart
```

Örneğin, bir Windows uç noktasındaki (IP adresi 192.168.71.125) bir Wazuh yöneticisine (IP adresi 192.168.71.203) bir Wazuh aracısının kaydı sırasında netstat kullanarak bağlantı hizmetinin çalışmasını doğrulayabilirsiniz. Ayrıca, herhangi bir Wazuh destekli uç noktada çalışan bir Wazuh aracısı, güvenlik olaylarını port üzerindeki Wazuh yöneticisine iletir . Yukarıdaki aracı bağlantı

hizmeti yapılandırma`1514` bölümünde ayrıntılı olarak açıklanan yapılandırmayı kullanır .

Wazuh yöneticisi ile Wazuh aracısı arasındaki bağlantı hizmetinin çalışmasını doğrulamak için aşağıdaki adımları gerçekleştirin:

1. Windows uç noktasında komut istemini başlatın ve uç noktadaki bağlantıları listelemek için şu komutları çalıştırın:`netstat -a`

   ```
   netstat -a
   ```

   Output

   ```
   C:\Users\Tony>netstat -a

   Active Connections

     Proto  Local Address          Foreign Address        State
      TCP    192.168.71.125:51787   a23-53-42-162:https    ESTABLISHED
     TCP    192.168.71.125:51788   a-0003:https           ESTABLISHED
     TCP    192.168.71.125:51789   a-0003:https           ESTABLISHED
     TCP    192.168.71.125:51790   a23-53-42-162:https    ESTABLISHED
     TCP    192.168.71.125:51791   192.168.71.203:1514    SYN_SENT
   ```

   `192.168.71.125`IP adresine sahip Windows uç noktasının bir TCP paketi gönderdiğini ve porttaki `SYN_SENT`IP adresine sahip Wazuh sunucusuyla bağlantı kurmayı beklediğini görebiliyoruz .`192.168.71.2031514`

2. `netstat`Wazuh sunucusunun Windows 10 uç noktasıyla ne zaman bağlantı kurduğunu görüntülemek için komutu çalıştırın.

   ```
   netstat
   ```

   Output

   ```
   Active Connections

     Proto  Local Address          Foreign Address        State
     TCP    192.168.71.125:3389    192.168.71.1:25743     ESTABLISHED
     TCP    192.168.71.125:51572   a23-64-12-19:https     CLOSE_WAIT
     TCP    192.168.71.125:51573   192.229.221.95:http    CLOSE_WAIT
     TCP    192.168.71.125:51694   192.168.71.203:1514    ESTABLISHED
     TCP    192.168.71.125:51699   192.168.20.103:ms-do   SYN_SENT
     TCP    192.168.71.125:51701   192.168.20.101:ms-do   SYN_SENT
     TCP    192.168.71.125:51703   20.231.121.79:http     SYN_SENT
     TCP    192.168.71.125:51704   192.168.20.125:ms-do   SYN_SENT
   ```

   IP adresine sahip Windows uç noktasının , port üzerindeki `192.168.71.125`IP adresine sahip Wazuh sunucusuna bağlı olduğunu görebiliyoruz .`192.168.71.2031514`

Analiz Motoru

Wazuh analiz motoru, Windows olayları, SSH günlükleri, web sunucusu günlükleri ve diğerleri gibi çeşitli günlük türlerindeki verileri analiz eder. İşlenen bilgi türünü belirlemek için kod çözücüleri ve kod çözülen olaydaki belirli kalıpları belirlemek için kuralları kullanır. Bu kurallar, bir IP adresini engelleme ve kötü amaçlı yazılımları kaldırma gibi uyarıları ve yanıt eylemlerini tetikleyebilir.

Veri Kaynakları

Wazuh, çeşitli kaynaklardan günlükler toplayarak BT altyapınızın tüm yönlerinin kapsamlı bir şekilde izlenmesine olanak tanır. Bu, Wazuh'un karmaşık tehditleri tespit etmesini, güvenlik açığı riskini azaltmasını, güvenlik politikalarına uyumu sağlamasını ve belirlenen güvenlik olaylarına hızla yanıt vermesini sağlar. Aşağıda Wazuh tarafından desteklenen bazı yaygın veri kaynakları verilmiştir:

- **İşletim sistemi günlükleri : Wazuh,** Windows , Linux ve macOS gibi çeşitli işletim sistemleri tarafından oluşturulan günlükleri toplar . Syslog, auditd, uygulama günlükleri ve diğerleri dahil olmak üzere Linux uç noktalarından çeşitli günlükler toplayabilir. Windows uç noktalarında, Wazuh varsayılan olarak Sistem, Uygulamalar ve Güvenlik olay kanallarından Windows olay günlüklerini toplar. Wazuh, macOS birleşik günlük sistemi (ULS) kullanarak macOS uç noktalarındaki günlükleri toplar. macOS ULS, tüm sistem düzeylerinde günlüklerin yönetimini ve depolanmasını merkezileştirir.
- **Syslog olayları** : Wazuh , Linux/Unix sistemleri ve Wazuh aracısı kurulumu gerektirmeyen ağ aygıtları da dahil olmak üzere çeşitli syslog özellikli aygıtlardan günlükleri toplar.
- **Aracısız izleme** : Wazuh aracısız izleme yeteneği, aracı kurulumunu desteklemeyen uç noktaları izler. Uç nokta ile Wazuh sunucusu arasında bir SSH bağlantısı gerektirir. Bu yetenek, dosyaların, dizinlerin veya yapılandırmaların izlenmesini ve uç noktada komutların çalıştırılmasını sağlar.
- **Bulut sağlayıcı günlükleri : Wazuh,** AWS , Azure , Google Cloud ve Office 365 gibi bulut hizmet sağlayıcılarından doğrudan günlükleri ve olayları toplayarak bulut altyapısını izler . Bunlara EC2 örnekleri, S3 kovaları, Azure VM'leri ve daha fazlası gibi bulut hizmetlerinden gelen günlükler dahildir.
- **Özel günlükler : Wazuh'u** VirusTotal , Windows Defender , ClamAV ve daha fazlası dahil olmak üzere çeşitli uygulamalardan ve üçüncü taraf güvenlik araçlarından günlükleri toplayacak ve ayrıştıracak şekilde yapılandırabilirsiniz .

Kod Çözme

Kod çözme, farklı veri kaynaklarından gelen günlükler gibi yapılandırılmış veya yapılandırılmamış verileri, izleme ve uyarı için kullanılabilecek anlamlı bilgiler çıkarmak için analiz etme sürecidir.

Wazuh'ta kod çözmenin temel amacı, ham verileri Wazuh yöneticisinin yorumlayabileceği ve işleyebileceği bir biçime dönüştürmektir. İki süreci içerir:

- **Ön kod çözme aşaması** : Bu aşamada, günlük analiz motoru günlük başlığından zaman damgası, ana bilgisayar adı ve program adı gibi syslog benzeri bilgileri çıkarır. Ön kod çözme aşaması günlük yapısını basitleştirir ve daha ileri analiz için hazırlar. Ön kod çözme sürecini göstermek için aşağıdaki örnek günlük girişini göz önünde bulundurun:

  ```
  Feb 14 12:19:04 192.168.1.1 sshd[25474]: Accepted password for Stephen from 192.168.1.133 port 49765 s
  ```

  Ön kod çözme aşamasını göstermek için Wazuh Logtest aracını kullanıyoruz. Wazuh sunucusunda aşağıdaki adımları gerçekleştirin:
  1. `/var/ossec/bin/wazuh-logtest`Wazuh sunucusunda komut satırından çalıştırın
  2. Yukarıdaki örnek günlüğü kopyalayıp yapıştırın ve enter'a basın.

  Ön kod çözme aşaması sonrasında elde edilen bilgiler aşağıda gösterilmektedir:

  ```
  Starting wazuh-logtest v4.8.0
  Type one log per line

  Feb 14 12:19:04 192.168.1.1 sshd[25474]: Accepted password for Stephen from 192.168.1.133 port 49765 s

  **Phase 1: Completed pre-decoding.
        full event: 'Feb 14 12:19:04 192.168.1.1 sshd[25474]: Accepted password for Stephen from 192.168.1.1
        timestamp: 'Feb 14 12:19:04'
        hostname: '192.168.1.1'
        program_name: 'sshd'
  ```

- **Kod çözme** : Bu aşamada, Wazuh analiz motoru günlükle eşleşen bir kod çözücü uygular. Kod çözücüler, günlüklerde bulunan kullanıcı adları, IP adresleri, hata kodları, URL'ler ve diğer ilgili bilgiler gibi alanları ayıklar. Aşağıdaki kod çözücüler örnek günlükle eşleşir. Bu kod çözücüler `/var/ossec/rulesets/decoders/0310-ssh_decoders.xml`Wazuh sunucusundaki dosyadadır:

  ```
  <decoder name="sshd">
    <program_name>^sshd</program_name>
  </decoder>

  <decoder name="sshd-success">
    <parent>sshd</parent>
    <prematch>^Accepted</prematch>
    <regex offset="after_prematch">^ \S+ for (\S+) from (\S+) port (\S+)</regex>
    <order>user, srcip, srcport</order>
    <fts>name, user, location</fts>
  </decoder>
  ```

  Kod çözücü `sshd`program adıyla eşleşirken `sshd`, kod çözücü örnek günlükten , , ve `ssh-success`öğelerini çıkarır .`Stephen``192.168.1.133``49765`

Kod çözme aşamasını göstermek için Wazuh Logtest aracını kullanıyoruz. Wazuh sunucusunda aşağıdaki adımları gerçekleştirin:

1. `/var/ossec/bin/wazuh-logtest`Wazuh sunucusunda from komut satırını çalıştırın .
2. Yukarıdaki örnek günlüğü kopyalayıp yapıştırın ve enter'a basın.

Kod çözme aşaması sonucunda elde edilen bilgiler aşağıda gösterilmektedir:

```
Starting wazuh-logtest v4.7.5
Type one log per line

Feb 14 12:19:04 192.168.1.1 sshd[25474]: Accepted password for Stephen from 192.168.1.133 port 49765 s

**Phase 1: Completed pre-decoding.
        full event: 'Feb 14 12:19:04 192.168.1.1 sshd[25474]: Accepted password for Stephen from 192.168.1.1
        timestamp: 'Feb 14 12:19:04'
        hostname: '192.168.1.1'
        program_name: 'sshd'

**Phase 2: Completed decoding.
        name: 'sshd'
        parent: 'sshd'
        dstuser: 'Stephen'
        srcip: '192.168.1.133'
        srcport: '49765'
```

Kural Değerlendirmesi ve Uyarı

Günlük çözüldükten sonra, Wazuh yöneticisi bunu bir kural setiyle karşılaştırır. Wazuh kural setleri XML dosyalarında tanımlanır ve farklı izleme ihtiyaçlarına uyacak şekilde özelleştirilebilir. Bu kurallar, karşılandığında uyarıları tetikleyen koşulları belirtir. `5715`Aşağıdaki kural, önceki bölümdeki örnek günlükle eşleşir. Bu kural, `/var/ossec/ruleset/rules/0095-sshd_rules.xml`Wazuh sunucusundaki dosyadadır.

```
<rule id="5715" level="3">
  <if_sid>5700</if_sid>
  <match>^Accepted|authenticated.$</match>
  <description>sshd: authentication success.</description>
  <group>authentication_success,pci_dss_10.2.5,</group>
</rule>
```

Nerede:

- `<rule id="5715" level="3">`kural kimliğini `5715`ve kural düzeyini olarak belirtir `3`. Kural kimliği kural için benzersiz bir tanımlayıcıdır, düzey ise kural eşleştiğinde olayın önem düzeyini temsil eder.
- `<if_sid>5700</if_sid>`ID'li başka bir kurala bağımlılığı belirtir `5700`. Kural yalnızca daha önce eşleşmişse değerlendirilecektir `5700`.

- `<match>^Accepted|authenticated.$</match>`ile başlayan `Accepted`veya biten herhangi bir günlük girişiyle eşleşir `authenticated.`.
- `<description>sshd: authentication success.</description>`kuralın neyi algıladığını açıklar. Bu durumda, başarılı bir SSH kimlik doğrulamasını gösterir.
- `<group>authentication_success,pci_dss_10.2.5,</group>`kuralı `authentication_success`ve `pci_dss_10.2.5` gruplarına atar.

Varsayılan olarak, Wazuh sunucusu 2'nin üzerinde bir seviyeye sahip herhangi bir kural için uyarılar üretir. Bu senaryoda, kural seviyesi 3 olduğu için günlük bir uyarıyı tetikler ve bu Wazuh panosunda görünür olacaktır.

Varsayılan olarak desteklenmeyen günlükleri analiz etmek için özel kod çözücüler ve kurallar oluşturabilirsiniz. Özel kurallar ve kod çözücülerin nasıl oluşturulacağını öğrenmek için özel kurallar ve özel kod çözücüler belgelerine bakın.

Sıraya Girme Mekanizmaları

Wazuh sunucusu, izlenen uç noktalardan olay toplanmasını kolaylaştıran bir kuyruk mekanizması içerir. Wazuh ajanlarından, syslog uç noktalarından ve ajansız cihazlardan Wazuh sunucusuna sürekli veri akışı sağlayarak olay taşmasını önler. Wazuh sunucu kuyruğu İlk Giren İlk Çıkar (FIFO) metodolojisini kullanır; bu nedenle, ilk kuyruğa alınan olay kuyruktan ilk kaldırılan ve işlenen olaydır. Dağıtılmış işleme dayalıdır ve günlük analiz görevlerinin paralel hale getirilmesine olanak tanır. Bu, günlük işleme hattının ölçeklenebilirliğini ve performansını iyileştirerek Wazuh'un büyük hacimli günlük verilerini etkili bir şekilde işlemesini sağlar.

Wazuh sunucusunda olay akışlarını yönetmek için iki yerel kuyruk bulunur:

- Wazuh aracı iletişim kuyruğu (queue_rd)
- Wazuh analiz motoru kuyruğu (queue_and)

Wazuh aracısı, olay tıkanıklığını önlemek için Wazuh aracı kuyruğunu (queue_ad) kullanır . Bu kuyruk, Wazuh aracısının Wazuh sunucusunun işleyebileceğinden daha hızlı olay göndermemesini sağlar.

Wazuh Agent İletişim Kuyruğu (queue_rd)

Kuyruk `queue_rd`, sunucu tarafı aracı iletişim hizmetinde bulunur . Wazuh aracılarından olayları alır ve olay kod çözme ve kural eşleştirme için Wazuh analiz motoruna gönderir .

Wazuh Agent İletişim Kuyruğu Nasıl Yapılandırılır

1. Wazuh sunucusundaki `/var/ossec/etc/ossec.conf` dosyasının `<queue_size>`uzak bölümünde düzenleme yaparak Wazuh aracı iletişim kuyruğunu yapılandırın:

```
<remote>
  <connection>secure</connection>
  <port>1514</port>
  <protocol>tcp,udp</protocol>
  <queue_size>131072</queue_size>
  <rids_closing_time>5m</rids_closing_time>
  <connection_overtake_time>600</connection_overtake_time>
  <agents>
    <allow_higher_versions>no</allow_higher_versions>
  </agents>
```

```
</remote>
```

Değişken `<queue_size>`, Wazuh aracı iletişim kuyruğunun kuyruk kapasitesini ayarlar. Aşağıdaki tablo `<queue_size>` değişkenin yapılandırmasını gösterir.

Varsayılan değer	İzin verilen değerler
131072	1 ile 262144 arasında herhangi bir sayı.

> Not: Wazuh aracı iletişim kuyruğu (`queue_rd`) yalnızca Wazuh aracı olayları için kullanılabilir, uzak syslog olayları için kullanılamaz. Bu seçenek yalnızca bağlantı olarak ayarlandığında çalışır `secure`.

2. Değişiklikleri uygulamak için Wazuh yönetici hizmetini yeniden başlatın.

```
systemctl restart wazuh-manager
```

Olay düşüşleri gözlemlendiğinde `/var/ossec/etc/ossec.conf` dosyasının `<remote>` bloğundaki `queue_size` değerini ve `/var/ossec/etc/internal_options.conf` dosyasındaki `worker_pool` boyutunu artırabilirsiniz.

`worker_pool`Aşağıdaki tablo Wazuh sunucusundaki boyut yapılandırmasını göstermektedir .

uzaktan.çalışan_havuzu	Tanım	Yük alımını işleyen iş parçacığı sayısı
	Varsayılan değer	4
	İzin verilen değer	1 ile 16 arasında herhangi bir tam sayı

Wazuh sunucu API'sini `wazuh-remoted` sorgulayarak veya daemon istatistiksel durum dosyasını okuyarak olay düşüşlerini izleyebilirsiniz .

Wazuh Sunucu API'sini Sorgulama

`wazuh-remoted`Aşağıdaki adımları izleyerek istatistiksel bilgileri sorgulayabilirsiniz :

1. Wazuh panosunda **Araçlar'a** ve ardından **API Konsolu'na** gidin .
2. API konsoluna aşağıdakileri ekleyin ve Wazuh sunucusu API'sine sorgu göndermek için yeşil oka tıklayın:

```
GET /manager/daemons/stats
```

3. Sorgu sonucu aşağıdaki ekran görüntüsünün sol tarafında gösterilmektedir.

Image not found or type unknown

Wazuh uzaktan istatistiklerini gösteren Wazuh daemon'larının istatistiksel sorgusu.

Sorgu, kuyruk boyutu değerini, tarafından işlenen olay sayısını `wazuh-remoted` ve atılan olay sayısını döndürür.

Aracı İletişim İstatistiksel Durum Dosyası

Bu istatistiksel dosya, `wazuh-remoted` kuyruk boyutu, atılan mesajlar, uzak bağlantı sayısı ve diğer önemli bilgiler gibi uzak daemon ile ilgili verileri sunar.

Dosyayı okumak için Wazuh sunucusunda aşağıdaki komutu çalıştırın:

```
cat /var/ossec/var/run/wazuh-remoted.state
```

Aşağıda dosyanın içeriğine dair bir örnek verilmiştir `wazuh-remoted.state`:

```
# State file for wazuh-remoted
# THIS FILE WILL BE DEPRECATED IN FUTURE VERSIONS
# Updated every 5 seconds.

# Queue size
queue_size='0'

# Total queue size
total_queue_size='131072'

# TCP sessions
tcp_sessions='1'

# Events sent to Analysisd
evt_count='126714'

# Control messages received
ctrl_msg_count='2637'

# Discarded messages
discarded_count='0'

# Total number of bytes sent
sent_bytes='4434745'

# Total number of bytes received
recv_bytes='93866086'

# Messages dequeued after the agent closes the connection
dequeued_after_close='0'
```

Wazuh Analiz Motoru Kuyruğu (queue_and)

Sıra Wazuh analiz motorunda `queue_and` bulunur ve olayların alınmasını kolaylaştırır. Wazuh analiz motoru daha sonra alınan günlükleri Wazuh sunucusundaki kurallarla eşleştirir.

Wazuh Analiz Motoru Kuyruğu Nasıl Yapılandırılır

Wazuh analiz motoru kuyruğu, `queue_and` kuyruğu kullanarak analiz için Wazuh ajanlarından günlükleri alır. Gelen tüm günlük mesajları kategorilere ayrılır ve aşağıdaki kategorilerde sıraya alınır:

- Dosya bütünlüğü izleme olayı kod çözücü kuyruğu.
- Syscollector olay kod çözücü kuyruğu.
- Kök denetimi olayı kod çözücü kuyruğu.
- Ana bilgisayar bilgisi olay kod çözücü kuyruğu.
- Olay kod çözücü kuyruğu.
- Windows olay kod çözücü kuyruğu.

Her kuyruk kategorisinin İlk Giren İlk Çıkar (FIFO) olay yönetiminden sorumlu bir dizi iş parçacığı vardır. İş parçacığı sayısı, `/var/ossec/etc/internal_options.conf` Wazuh sunucusundaki dosya aracılığıyla olay türüne göre ayrı ayrı yapılandırılabilir.

> Not: Yükseltmelerin kuyruk yapılandırmalarını geçersiz kılmamasını sağlamak için `/var/ossec/etc/local_internal_options.conf` dosyası yerine `/var/ossec/etc/internal_options.conf` dosyasını kullanın.

Aşağıdaki tabloda Wazuh analiz motoru kuyruğu (`queue_and`) için kullanılabilen yapılandırma seçenekleri gösterilmektedir.

Kuyruklar (wazuh-analysisd.state)	Ayar (local_internal_options.conf)	Varsayılan	Dakika	Maksimum
syscheck_queue_kullanımı	analizd.decode_syscheck_queue_size	16384	128	2000000
syscollector_kuyruğu_kullanımı	analizd.decode_syscollector_queue_size	16384	128	2000000
kök_kontrolu_kuyruk_kullanımı	analizd.decode_rootcheck_queue_size	16384	128	2000000
sca_queue_kullanımı	analizd.decode_sca_queue_size	16384	128	2000000

Kuyruklar (wazuh-analysisd.state)	Ayar (local_internal_options.conf)	Varsayılan	Dakika	Maksimum
hostinfo_kuyruk_kullanımı	analiz.decode_hostinfo_queue_size	16384	128	2000000
winevt_kuyruk_kullanımı	analizd.decode_winevt_kuyruk_boyutu	16384	128	2000000
dbsync_kuyruk_kullanımı	analizd.dbsync_queue_size	16384	128	2000000
yükseltme_kuyruğu_kullanımı	analizd.yükseltme_kuyruğu_boyutu	16384	128	2000000
olay_kuyruğu_kullanımı	analizd.decode_event_queue_size	16384	128	2000000
kural_eşleşen_kuyruk_kullanımı	analizd.decode_output_queue_size	16384	128	2000000
uyarılar_kuyruğu_kullanımı	analizd.uyarılar_kuyruk_boyutu	16384	128	2000000
güvenlik_kuyruğu_kullanımı	analizd.firewall_queue_size	16384	128	2000000
istatistiksel_kuyruk_kullanımı	analizd.istatistiksel_kuyruk_boyutu	16384	128	2000000
arşiv_kuyruğu_kullanımı	analizd.arşivler_kuyruğu_boyutu	16384	128	2000000
	analizd.fts_kuyruk_boyutu	16384	128	2000000
	analizd.fts_liste_boyutu	32	12	512
	analysisd.fts_min_size_for_str	14	6	128
	analizd.decoder_order_size	256	32	1024

Wazuh analiz motorunda "olay düşüşleri" gözlemlendiğinde kuyruk ayarları ayarlanmalıdır. Wazuh sunucu API'sini sorgulayarak veya daemon istatistiksel durum dosyasını okuyarak wazuh-analysisd'deki olay düşüşlerini izleyebilirsiniz .

Wazuh Sunucu API'sini Sorgulama

Wazuh analiz motorundan istatistiksel bilgileri kontrol etmek için günlük kategorisi durumu Wazuh sunucu API'si kullanılarak sorgulanabilir. Yeni istatistikler, alınan veya düşürülen olayların olay türüne göre dökümünü gösterir. Bu, yalnızca düşürmeyi gösteren kuyruk boyutlarını ayarlamak için hayati önem taşır.

Aşağıdaki adımları izleyerek Wazuh analiz motorunun istatistiksel bilgilerini sorgulayabilirsiniz:

1. Wazuh panosunda **Araçlar'a** ve ardından **API Konsolu'na** gidin .
2. Konsola aşağıdakileri ekleyin ve Wazuh sunucu API'sine sorgu göndermek için yeşil oka tıklayın:

```
GET /manager/daemons/stats
```

3. `wazuh-analysisd` Aşağıdaki ekran görüntüsünde sağ tarafta gösterilen sorgu sonucunun bulunduğu bölüme doğru aşağı kaydırın .
Wazuh analysisd istatistiklerini gösteren Wazuh daemon'larının istatistiksel sorgusu

Sorgu, kuyruk boyutu değerini, Wazuh analiz motoru tarafından işlenen olay sayısını ve atılan olay sayısını döndürür.

`/var/ossec/etc/internal_options.conf` Wazuh analiz motoru kuyruğu , Wazuh sunucusundaki dosya aracılığıyla olay türüne göre yapılandırılabilir .

> Not: Yükseltmelerin kuyruk yapılandırmalarını geçersiz kılmamasını sağlamak için `/var/ossec/etc/local_internal_options.conf` dosyası yerine `/var/ossec/etc/internal_options.conf` dosyasını kullanın.

Wazuh Analiz Motoru İstatistiksel Durum Dosyası

Wazuh analiz motoru için istatistiksel dosya şu adreste bulunur `/var/ossec/var/run/wazuh-analysisd.state`. Dosya, Wazuh sunucusundaki olay işleme sorunlarını araştırırken yararlı olabilir.

Dosyayı okumak için Wazuh sunucusunda aşağıdaki komutu çalıştırın:

```
cat /var/ossec/var/run/wazuh-analysisd.state
```

Aşağıda wazuh-remoted.state dosyasının içeriğine dair bir örnek verilmiştir:

```
# State file for wazuh-analysisd
# THIS FILE WILL BE DEPRECATED IN FUTURE VERSIONS

# Total events decoded
total_events_decoded='137726'

# Syscheck events decoded
syscheck_events_decoded='3935'

# Syscollector events decoded
syscollector_events_decoded='2590'

# Rootcheck events decoded
rootcheck_events_decoded='37'
```

```
# Security configuration assessment events decoded
sca_events_decoded='8991'

# Winevt events decoded
winevt_events_decoded='87993'

# Database synchronization messages dispatched
dbsync_messages_dispatched='26004'

# Other events decoded
other_events_decoded='8176'

# Events processed (Rule matching)
events_processed='112252'

# Events received
events_received='138283'

# Events dropped
events_dropped='0'

# Alerts written to disk
alerts_written='6707'

# Firewall alerts written to disk
firewall_written='0'

# FTS alerts written to disk
fts_written='0'

# Syscheck queue
syscheck_queue_usage='0.00'

# Syscheck queue size
syscheck_queue_size='16384'

# Syscollector queue
syscollector_queue_usage='0.00'

# Syscollector queue size
syscollector_queue_size='16384'

# Rootcheck queue
rootcheck_queue_usage='0.00'

# Rootcheck queue size
rootcheck_queue_size='16384'

# Security configuration assessment queue
sca_queue_usage='0.00'

# Security configuration assessment queue size
sca_queue_size='16384'
```

```
# Hostinfo queue
hostinfo_queue_usage='0.00'

# Hostinfo queue size
hostinfo_queue_size='16384'

# Winevt queue
winevt_queue_usage='0.00'

# Winevt queue size
winevt_queue_size='16384'

# Database synchronization message queue
dbsync_queue_usage='0.00'

# Database synchronization message queue size
dbsync_queue_size='16384'

# Upgrade module message queue
upgrade_queue_usage='0.00'

# Upgrade module message queue size
upgrade_queue_size='16384'

# Event queue
event_queue_usage='0.00'

# Event queue size
event_queue_size='16384'

# Rule matching queue
rule_matching_queue_usage='0.00'

# Rule matching queue size
rule_matching_queue_size='16384'

# Alerts log queue
alerts_queue_usage='0.00'

# Alerts log queue size
alerts_queue_size='16384'

# Firewall log queue
firewall_queue_usage='0.00'

# Firewall log queue size
firewall_queue_size='16384'

# Statistical log queue
statistical_queue_usage='0.00'

# Statistical log queue size
statistical_queue_size='16384'
```

```
# Archives log queue
archives_queue_usage='0.00'

# Archives log queue size
archives_queue_size='16384'
```

Wazuh Agent Kuyruğu (queue_ad)

Sıra `queue_ad`, aracı tarafı aracı bağlantı hizmetinde bulunur ve Wazuh aracısından Wazuh sunucusuna olay iletimini yönetir. Sıra, Wazuh sunucusuna iletmeden önce sistem olayları ve güvenlik yapılandırması değerlendirme çıktıları gibi günlükleri toplar. Ayrıca, yapılandırılabilir parametrelere göre olay iletimini kısıtlayan ve Wazuh sunucusunun işleme kapasitesini aşma riskini azaltan bir anti-flooding mekanizması içerir.

Wazuh Kuyruk Çözücü ve Kuralları

Wazuh, olay sel çıkışını analiz etmek ve Wazuh panosunda uyarılar oluşturmak için kullanıma hazır bir kod çözücü ve kurallar sağlar.

Decoder

Kod çözücü Wazuh sunucusundaki dosyada mevcuttur `/var/ossec/ruleset/decoders/0005-wazuh_decoders.xml`. Kod çözücü Wazuh sunucusundaki flood olaylarını analiz etmekten sorumludur.

```
<decoder name="agent-buffer">
  <parent>wazuh</parent>
  <prematch offset="after_parent">^Agent buffer:</prematch>
  <regex offset="after_prematch">^ '(\S+)'.</regex>
  <order>level</order>
</decoder>
```

Kurallar

Aşağıda görüldüğü gibi kurallar, `201`ile arasındaki ID'lerle tanımlanmış olup Wazuh sunucusundaki dosyada `205`mevcuttur ./`var/ossec/ruleset/rules/0016-wazuh_rules.xml`

```
<!-- Agent buffer rules -->
<rule id="201" level="0">
  <if_sid>200</if_sid>
  <match>^wazuh: Agent buffer: </match>
  <description>Agent event queue rule</description>
  <group>agent_flooding,</group>
</rule>
```

```
<rule id="202" level="7">
  <if_sid>201</if_sid>
  <field name="level">%</field>
  <description>Agent event queue is $(level) full.</description>
  <group>agent_flooding,pci_dss_10.6.1,gdpr_IV_35.7.d,</group>
</rule>

<rule id="203" level="9">
  <if_sid>201</if_sid>
  <field name="level">full</field>
  <description>Agent event queue is full. Events may be lost.</description>
  <group>agent_flooding,pci_dss_10.6.1,gdpr_IV_35.7.d,</group>
</rule>

<rule id="204" level="12">
  <if_sid>201</if_sid>
  <field name="level">flooded</field>
  <description>Agent event queue is flooded. Check the agent configuration.</description>
  <group>agent_flooding,pci_dss_10.6.1,gdpr_IV_35.7.d,</group>
</rule>

<rule id="205" level="3">
  <if_sid>201</if_sid>
  <field name="level">normal</field>
  <description>Agent event queue is back to normal load.</description>
  <group>agent_flooding,</group>
</rule>
```

Nerede:

- Kural Kimliği, `201`olay kuyruğu için temel kuraldır.
- Kural Kimliği, `202`olay kuyruğu seviyesi %90'a ulaştığında tetiklenir.
- Kural kimliği, `203`olay kuyruğu dolduğunda tetiklenir.
- `204`Olay kuyruğu dolduğunda kural kimliği tetiklenir.
- Kural Kimliği, `205`bir su baskını olayından sonra olay kuyruğu normale döndüğünde tetiklenir.