# GitHub Denetim Günlüklerinin İzlenmesi

Kurumsal yöneticiler, kuruluşları içindeki üyeler tarafından gerçekleştirilen eylemleri incelemek için proaktif olarak GitHub denetim günlüklerini kullanır. Eylemi gerçekleştiren kullanıcı, eylemin niteliği ve yürütme zaman damgası gibi ayrıntıları içerir. GitHub için Wazuh modülü, API'si aracılığıyla GitHub'dan denetim günlüklerinin toplanmasını sağlar. Wazuh, denetim günlüklerini toplamak için GitHub API uç noktasına bir HTTP GET isteği başlatır . Daha fazla ayrıntı için GitHub REST API `/orgs/{org}/audit-log` belgelerine başvurabilirsiniz .

# GitHub Denetim Günlüklerinin İzlenmesine İlişkin Gereksinimler

Wazuh ile denetim günlüklerine erişebilmek için GitHub'da aşağıdaki gereksinimlere sahip olmanız gerekir.

- **GitHub organizasyonu** : Yalnızca GitHub organizasyonlarına ait denetim günlüklerini görüntüleyebilirsiniz.
- **GitHub Enterprise Cloud aboneliği** : Yalnızca GitHub Enterprise Cloud aboneliği olan kuruluşlar GitHub denetim günlüğü REST API'sini kullanabilir.

## GitHub'da Kişisel Erişim Belirteci Oluşturma

Gerekli kişisel erişim belirtecini oluşturmak için GitHub'da aşağıdaki adımları izleyin:

1. Kuruluş sahibine ait bir hesapla GitHub'a giriş yapın.
2. Yeni bir kişisel erişim belirteci oluşturmak için https://github.com/settings/tokens/new adresine gidin .
3. Kişisel erişim belirteci için açıklayıcı bir not ekleyin ve bir son kullanma tarihi seçin.

GitHub yeni kişisel erişim belirteci



4. Aşağı kaydırın, **audit_log** öğesini seçin ve **Jeton oluştur öğesine** tıklayın.

GitHub token üret

5. Yeni oluşturulan kişisel erişim belirtecini kopyalayın.



GitHub kopyası oluşturulan belirteç

# Wazuh'u GitHub Günlüklerini Çekecek Şekilde Yapılandırın

Wazuh'un GitHub denetim günlüklerini izlemesine, toplamasına ve analiz etmesine izin vermek için aşağıdaki adımları gerçekleştirin. Wazuh modülünü GitHub için Wazuh sunucusunda veya Wazuh aracısında yapılandırabilirsiniz.

1. Aşağıdaki yapılandırmayı `/var/ossec/etc/ossec.conf` Wazuh sunucusundaki dosyaya ekleyin.

```
<ossec_config>
  <github>
    <enabled>yes</enabled>
    <interval>1m</interval>
    <time_delay>1m</time_delay>
    <curl_max_size>1M</curl_max_size>
    <only_future_events>yes</only_future_events>
    <api_auth>
      <org_name><ORG_NAME></org_name>
      <api_token><API_TOKEN></api_token>
    </api_auth>
    <api_parameters>
      <event_type>all</event_type>
    </api_parameters>
  </github>
</ossec_config>
```

   Nerede:

   - `<enabled>`: GitHub için Wazuh modülünü etkinleştirir. İzin verilen değerler `yes` ve 'dir `no`.
   - `<interval>`: GitHub için Wazuh modülünün her yürütülmesi arasındaki zaman aralığını tanımlar. Varsayılan değer 'dir `10m` ve izin verilen değer, s (saniye), m (dakika), h (saat) ve d (gün) gibi bir zaman birimini belirten bir sonek karakteri içeren herhangi bir pozitif sayıdır.
   - `<time_delay>`: Taramanın geçerli zamana göre gecikme süresini belirtir. Varsayılan değer 'dir `30s` ve izin verilen değer, s (saniye), m (dakika), h (saat) ve d (gün) gibi bir zaman birimini belirten bir sonek karakteri içeren herhangi bir pozitif sayıdır.
   - `<curl_max_size>`: GitHub API yanıtı için izin verilen maksimum boyutu belirtir. Varsayılan değer 'dir `1M` ve izin verilen değer, b/B (bayt), k/K (kilobayt), m/M (megabayt) ve g/G (gigabayt) gibi bir boyut birimini belirten bir sonek karakteri içeren herhangi bir pozitif sayıdır.
   - `<only_future_events>`: Evet olarak ayarlandığında, GitHub için Wazuh modülü yalnızca Wazuh yöneticisini başlattıktan sonra oluşturulan olayları toplar. olarak

ayarlandığında `no`, Wazuh yöneticisini başlatmadan önce oluşturulan önceki olayları toplar. Varsayılan değer 'dir `yes`ve izin verilen değerler `yes`ve ' dir `no`.

- `<api_auth>`: Bu blok, GitHub REST API ile kimlik doğrulama için kimlik bilgilerini yapılandırır. Aşağıdaki etiketler `<org_name>`ve `<api_token>`, içindeki yapılandırma etiketleridir `<api_auth>`.
  - `<org_name>`: GitHub organizasyonunuzun adı. İzin verilen değer herhangi bir dizedir.
  - `<api_token>`: GitHub API ile kimlik doğrulaması yapmak için kişisel erişim belirteci. İzin verilen değer herhangi bir dizedir.
- `<api_parameters>`: Bu blok GitHub REST API'sindeki dahili seçenekleri yapılandırır. İçindeki bir alt yapılandırma `<api_parameters>`bloğu `<event_type>`.
  - `<event_type>`: Wazuh'un toplaması gereken olay türlerini belirtir. Kullanılabilir olay türleri web ve git olaylarıdır. Bu yapılandırma bloğu için varsayılan değer `all`, hem web hem de git olaylarını toplamaktır. İzin verilen değerler `all`, `web`, ve 'dir `git`.

Yapılandırma seçenekleri hakkında daha fazla bilgi edinmek için GitHub referansı için Wazuh modülüne bakın.

2. Değişiklikleri uygulamak için Wazuh yöneticisini veya aracı hizmetini yeniden başlatın:

   - **Wazuh yöneticisi**

     ```
     systemctl restart wazuh-manager
     ```

   - **Wazuh temsilcisi**

     ```
     systemctl restart wazuh-agent
     ```

# Birden Fazla GitHub Organizasyonunu İzleyin

Wazuh ile birden fazla GitHub kuruluşunu, kuruluş kimlik bilgilerini ayrı bölümlerde belirterek izleyebilirsiniz `<api_auth>`. Örneğin, aşağıdaki yapılandırma `organization1`ve adlı iki kuruluşu izler `organization2`.

```
<github>
  <enabled>yes</enabled>
  <interval>1m</interval>
  <time_delay>1m</time_delay>
  <curl_max_size>1M</curl_max_size>
  <only_future_events>no</only_future_events>

  <api_auth>
    <org_name>organization1</org_name>
    <api_token><API_TOKEN></api_token>
  </api_auth>
```

```
  <api_auth>
    <org_name>organization2</org_name>
    <api_token><API_TOKEN></api_token>
  </api_auth>

  <api_parameters>
    <event_type>git</event_type>
  </api_parameters>
</github>
```

Nerede:

- `<API_TOKEN>`kapsam dahilinde oluşturulan GitHub kişisel erişim belirtecidir `admin:org`.

# Kullanım Durumu

## Gereksinimler

- GitHub aktivite izleme dokümanlarına göre Wazuh'u GitHub ile entegre edin .
- Curl yüklü bir Ubuntu 23.10 uç noktası.

## GitHub

`admin:org`, `repo`, ve kapsamları içinde bir GitHub kişisel erişim belirteci oluşturun `delete_repo`. Bu belirteci, Wazuh'ta uyarıları tetikleyen kuruluşta eylemler gerçekleştirerek kullanım durumlarını test etmek için GitHub REST API'siyle kullanacağız. Belirteci oluşturmak için aşağıdaki adımları izleyin:

1. https://github.com/settings/tokens/new adresine gidin , token için bir not ekleyin, istediğiniz son kullanma tarihini seçin ve ardından `repo`ve `admin:org`kapsamlarını seçin.

GitHub yeni kişisel erişim belirteci
Image not found or type unknown

2. Sayfanın en altına gidin, ardından `delete_repo`kapsamı seçin ve **Jeton oluştur** düğmesine tıklayın.

GitHub Jeton oluştur
Image not found or type unknown

3. Yeni oluşturulan kişisel erişim belirtecini kopyalayın.

Oluşturulan belirteci kopyala
Image not found or type unknown

## Ubuntu Uç Noktası

### Örgüt Üyelerinin Manipülasyonlarını Tespit Edin

### Bir Üyeyi Davet Et

Kuruluşunuza bir üye davet etmek için aşağıdaki adımları izleyin.

1. Ubuntu uç noktasında aşağıdaki komutu çalıştırın:

   ```
   # curl -L \
     -X POST \
     -H "Accept: application/vnd.github+json" \
     -H "Authorization: Bearer <API_TOKEN>" \
     -H "X-GitHub-Api-Version: 2022-11-28" \
     https://api.github.com/orgs/<ORG_NAME>/invitations \
     -d '{"email":"<USER_EMAIL>","role":"direct_member"'
   ```

   Nerede:
   - `<API_TOKEN>`kapsam dahilinde oluşturulan GitHub kişisel erişim belirtecidir `admin:org`.
   - `<ORG_NAME>`kuruluşunuzun adıdır.
   - `<USER_EMAIL>`davet etmek istediğiniz kullanıcının e-posta adresidir.
2. Davet edilen üyenin posta kutusuna gidin ve daveti kabul edin.

### Bir Üyeyi Yönetici Olarak Terfi Ettir

Kuruluşunuzdaki bir üyeyi yönetici rolüne yükseltmek için aşağıdaki komutu çalıştırın:

```
# curl \
  -u <ADMIN_USERNAME>:<API_TOKEN> \
  -X PUT \
  -H "Accept: application/vnd.github.v3+json" https://api.github.com/orgs/<ORG_NAME>/memberships/<MEMBER_
   -d '{"role":"admin"}'
```

Nerede:

- `<ADMIN_USERNAME>`geçerli bir yöneticinin kullanıcı adıdır. Örneğin, kuruluşun sahibinin kullanıcı adı.
- `<API_TOKEN>` kapsam dahilinde oluşturulan GitHub kişisel erişim belirtecidir `admin:org`.
- `<ORG_NAME>`kuruluşunuzun adıdır.
- `<MEMBER_USERNAME>`terfi ettirmek istediğiniz kullanıcının kullanıcı adıdır.

### Yeni Bir Ekip Oluştur

Kuruluşunuzda yeni bir ekip oluşturmak için aşağıdaki komutu çalıştırın:

```
# curl -X POST \
    -H "Authorization: Bearer <API_TOKEN>" \
    -d '{"name": "<NEW_TEAM_NAME>"}' \
    "https://api.github.com/orgs/<ORG_NAME>/teams"
```

Nerede:

- `<NEW_TEAM_NAME>`yeni takımın adı.
- `<API_TOKEN>`kapsam dahilinde oluşturulan GitHub kişisel erişim belirtecidir `admin:org`.
- `<ORG_NAME>`kuruluşunuzun adıdır.

Aşağıdaki görsel, izlenen GitHub organizasyonunda yukarıdaki eylemleri gerçekleştirdikten sonra Wazuh panosunda oluşan uyarıları göstermektedir.



GitHub üyeleri izleme uyarı panosu

## Bir Depoda Yapılan Değişiklikleri Algıla

### Yeni Bir Depo Oluştur

Yeni bir depo oluşturmak için aşağıdaki komutu çalıştırın:

```
# curl -L \
  -X POST \
  -H "Accept: application/vnd.github+json" \
  -H "Authorization: Bearer <API_TOKEN>" \
  -H "X-GitHub-Api-Version: 2022-11-28" \
  https://api.github.com/orgs/<ORG_NAME>/repos \
  -d '{"name":"<NEW_REPO_NAME>"}'
```

Nerede:

- `<API_TOKEN>`kapsam dahilinde oluşturulan GitHub kişisel erişim belirtecidir `repo`.
- `<ORG_NAME>`kuruluşunuzun adıdır.
- `<NEW_REPO_NAME>`oluşturmak istediğiniz deponun adıdır.

### Deponuza Bir Takım Ekleyin

Takımları listelemek ve deponuza bir takım eklemek için aşağıdaki komutları çalıştırın:

1. Organizasyonunuzdaki takım kimliklerini listeleyin.
   ```
   # curl -H "Authorization: Bearer <API_TOKEN>" "https://api.github.com/orgs/<ORG_NAME>/teams"
   ```
2. Deponuza eklemek istediğiniz takımın ID'sini girin.
   ```
   # curl -X PUT \
       -H "Authorization: Bearer <API_TOKEN>" \
       -d '{"permission": "push"}' \
       "https://api.github.com/teams/<TEAM_ID>/repos/<ORG_NAME>/<REPO_NAME>"
   ```

   Nerede:
   - `<TEAM_ID>`Takım ID'sidir.
   - `<API_TOKEN>`kapsam dahilinde oluşturulan GitHub kişisel erişim belirtecidir `admin:org`.
   - `<ORG_NAME>`kuruluşunuzun adıdır.
   - `<REPO_NAME>`Bir takımı eklemek istediğiniz deponun adıdır.

**Ayrıcalıkları Yönet**

Ekibinizdeki üyelere depoda yönetici ayrıcalıkları vermek için aşağıdaki komutu çalıştırın:

```
# curl \
-u <ADMIN_USERNAME>:<API_TOKEN> \
-X PUT \
-H "Accept: application/vnd.github.v3+json" https://api.github.com/orgs/<ORG_NAME>/teams/<TEAM_NAME>/repo
 -d '{"permission":"admin"}'
```

Nerede:

- `<ADMIN_USERNAME>`bir kullanıcıyı yönetici olarak yükseltme iznine sahip kullanıcının kullanıcı adıdır
- `<API_TOKEN>`kapsam dahilinde oluşturulan GitHub kişisel erişim belirtecidir `admin:org`.
- `<ORG_NAME>`kuruluşunuzun adıdır.
- `<REPO_NAME>`Ekibinin erişimini yönetmek istediğiniz deponun adıdır.
- `<TEAM_NAME>`deponuzdaki belirli takımın adıdır.

**Depoyu Sil**

Kuruluşunuzdaki bir deponun silinmesi için aşağıdaki komutu çalıştırın:

```
# curl -L \
  -X DELETE \
  -H "Accept: application/vnd.github+json" \
  -H "Authorization: Bearer <API_TOKEN>" \
  -H "X-GitHub-Api-Version: 2022-11-28" \
  https://api.github.com/repos/<ORG_NAME>/<REPO_NAME>
```

Nerede:

- `<API_TOKEN>`kapsam dahilinde oluşturulan GitHub kişisel erişim belirtecidir `delete_repo`.
- `<ORG_NAME>`kuruluşunuzun adıdır.
- `<REPO_NAME>`kuruluşunuzdan silmek istediğiniz deponun adıdır.

**Takımı Sil**

Oluşturduğunuz takımı silmek için aşağıdaki komutu çalıştırın:

```
# curl -L \
  -X DELETE \
  -H "Accept: application/vnd.github+json" \
  -H "Authorization: Bearer <API_TOKEN>" \
  https://api.github.com/orgs/<ORG_NAME>/teams/<TEAM_NAME>
```

Nerede:

- `<API_TOKEN>`kapsam dahilinde oluşturulan GitHub kişisel erişim belirtecidir `admin:org`.
- `<ORG_NAME>`kuruluşunuzun adıdır.
- `<TEAM_NAME>`deponuzdaki belirli takımın adıdır.

Aşağıdaki görsel, izlenen GitHub organizasyonunda yukarıdaki eylemleri gerçekleştirdikten sonra Wazuh panosunda oluşan uyarıları göstermektedir.




GitHub deposu izleme uyarıları panosu

Revision #2
Created 30 December 2024 21:42:24 by Ayşegül Sarıkaya
Updated 31 December 2024 18:14:18 by Ayşegül Sarıkaya