API Kullanımı

- Başlangıç
- API Erişimi ve Yetkilendirme
- Rest Client Nedir?
- REST Client ile API İstekleri
- PyMISP ile Otomasyon

Başlangıç

MISP API, Tehdit İstihbaratı Paylaşım Platformu'nun (MISP), diğer sistemlerle entegrasyon için sağladığı güçlü bir araçtır. API, tehdit aktörlerinden zararlı yazılımlara kadar geniş bir yelpazedeki güvenlik verilerini paylaşmak için bir ortam sağlar. MISP API, otomatik veri alışverişi, analiz ve uyarı oluşturma gibi çeşitli güvenlik senaryolarında önemli rol oynar.

MISP API, uygulama geliştiricilerine, organizasyonların MISP platformunu kendi güvenlik altyapılarına entegre etmelerini sağlar. Bu entegrasyon, gerçek zamanlı tehdit bilgilerine erişim sağlayarak güvenlik operasyonlarını güçlendirir ve yanıt sürelerini iyileştirir. Ayrıca, MISP API, çeşitli güvenlik araçları ve sistemlerle etkileşim kurarak tehdit tespiti ve müdahale süreçlerini otomatikleştirmeyi kolaylaştırır. Bu sayede, kurumlar tehditlerle mücadele etmek için daha hızlı ve etkili bir şekilde hareket edebilirler.

OpenAPI Belgesi Kullanımı:

OpenAPI belgeleri, MISP API'nin kullanımını tanımlayan bir rehberdir. Bu belgeler, API endpoint'lerinin, parametrelerin ve kullanım yönergelerinin yanı sıra API'ye yapılan isteklerin nasıl yapılandırılacağı hakkında ayrıntılı bilgi sağlar. Özellikle, belgede her endpoint'in URL'si, desteklediği HTTP metotları (GET, POST, PUT, DELETE vb.), gerekli ve isteğe bağlı parametreler, yanıt formatı ve hata durumları gibi bilgiler bulunur.

OpenAPI belgeleri, MISP API'nin sağladığı işlevlerin ve servislerin ayrıntılı bir tanımını içerir. Bu belgeler, API'nin nasıl kullanılacağı hakkında kapsamlı bir rehber sağlar ve API'nin işlevselliğini tam olarak anlamak için önemlidir.

REST Client Arayüzü Kullanımı:

REST client arayüzü, kullanıcılara API'ye doğrudan erişim ve istek gönderme imkanı sağlar. Bu arayüz, API belgelerinde belirtilen parametreleri kullanarak istekler oluşturabilir ve cevapları alabilir. Kullanıcılar, API isteklerini kolayca yapılandırabilir, istenen parametreleri ekleyebilir ve istekleri göndererek API'den veri alabilirler.

REST client arayüzü genellikle bir web tabanlı uygulama veya bir masaüstü uygulama olarak sunulur. Kullanıcılar, bu arayüzü kullanarak belirli bir API endpoint'ine istek göndermek için gereken HTTP yöntemini (GET, POST, PUT, DELETE vb.) seçebilirler. Ardından, istek için gereken parametreleri ve verileri ekleyebilirler ve isteği göndererek API'den cevap alabilirler.

Bu arayüz, API'ye hızlı ve etkili bir şekilde erişmek için kullanışlı bir araçtır ve API'nin nasıl kullanılacağını anlamak için OpenAPI belgeleriyle birlikte kullanılabilir.

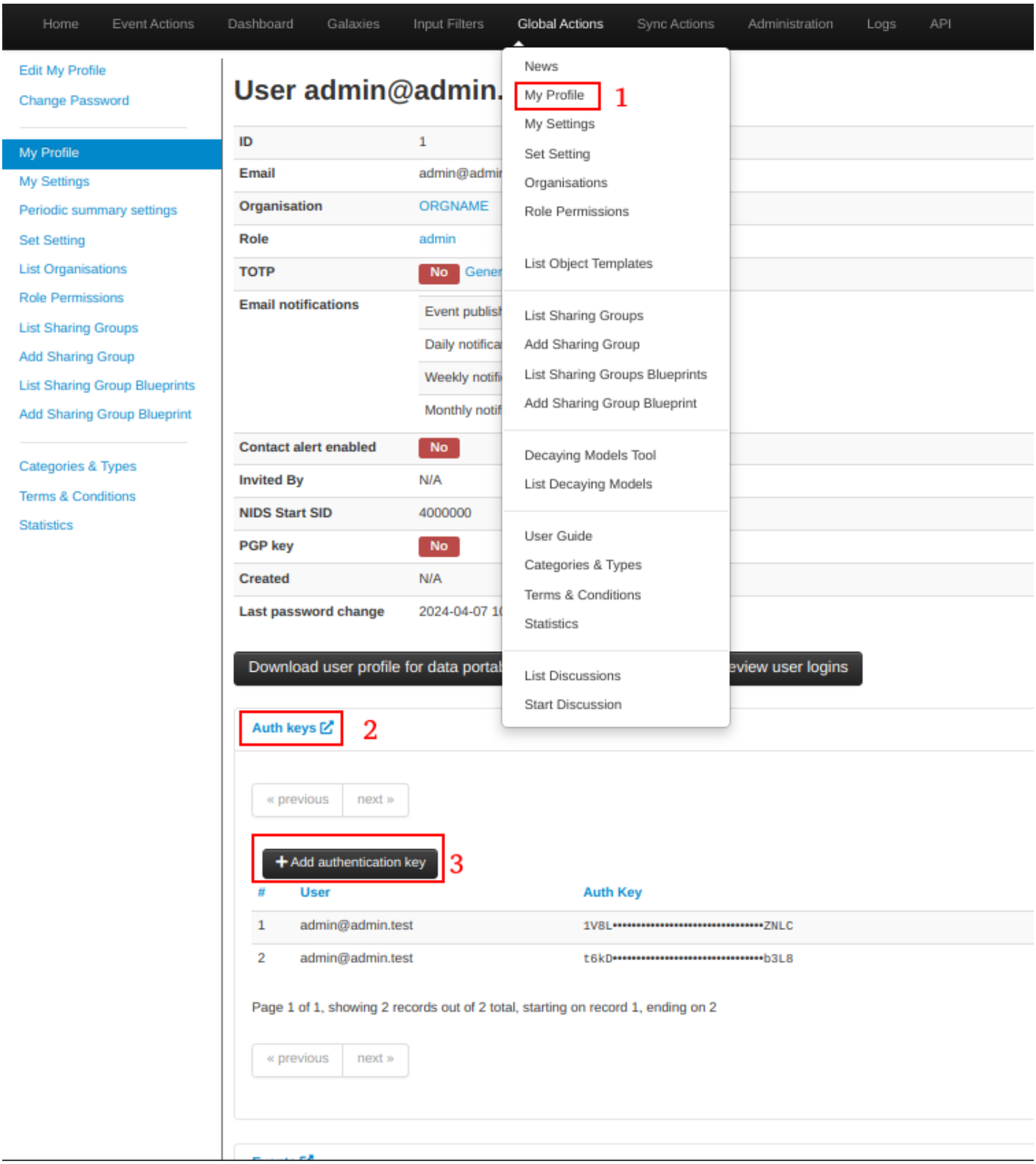
API Erişimi ve Yetkilendirme

MISP API'ye erişim, kullanıcıların güvenlik bilgilerini sağlamaları gereken yetkilendirme mekanizması üzerinden gerçekleşir. API'ye erişim için bir API anahtarı (Auth key) gereklidir ve bu anahtar, MISP kullanıcı arayüzünden ya da komut satırı aracılığı ile alınabilir.

> API anahtarı, erişimin güvenliğini sağlamak için özenle saklanmalıdır, çünkü bu anahtar tüm veri tabanına erişim sağlar.

Kullanıcı Arayüzü:

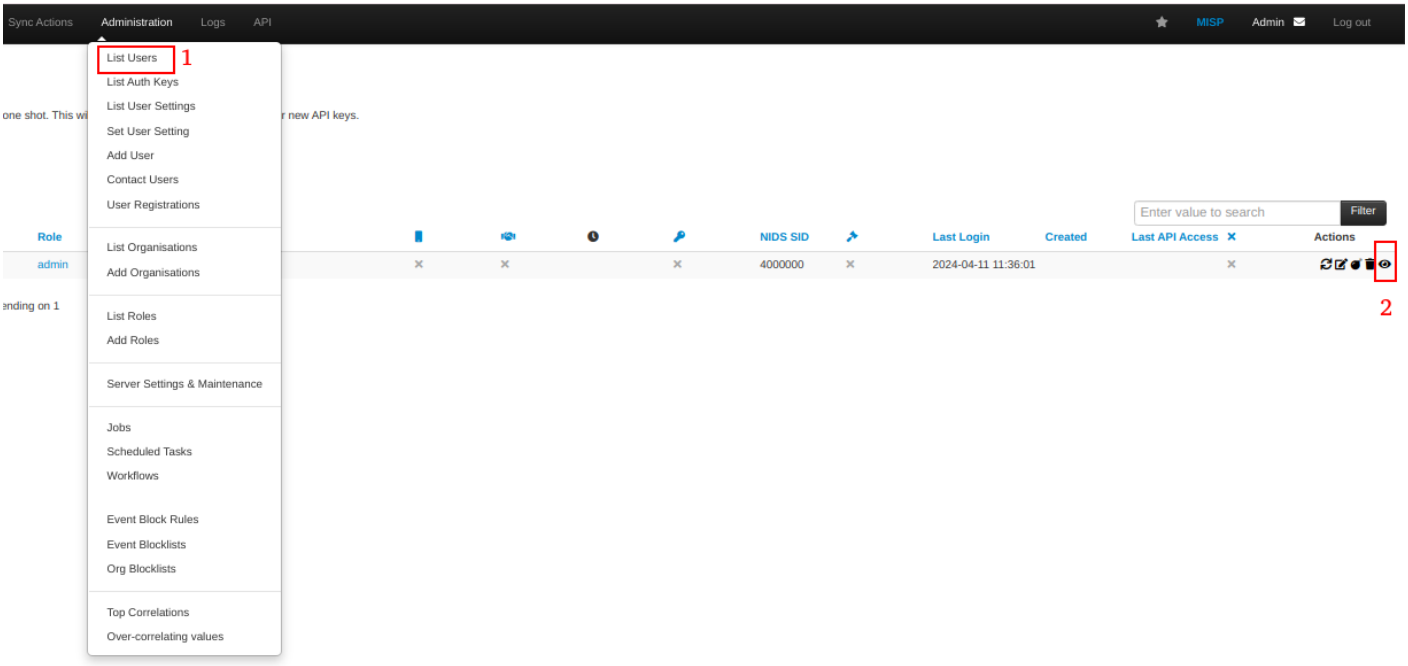
1. **Profilim -> Kimlik Doğrulama Anahtarları Bölümü:**
   - Bu adımlar, kullanıcının kendi API anahtarını oluşturmasını sağlar.
   - Kullanıcı, kendi hesabına giriş yaparak "Profilim" sekmesine gitmelidir.
   - Ardından, "Kimlik Doğrulama Anahtarları" bölümüne tıklamalı ve "Kimlik Doğrulama Anahtarı Ekle" seçeneğini seçmelidir.
   - Bu adımları takip ederek, kullanıcı kendi API anahtarını oluşturabilir ve kullanabilir.

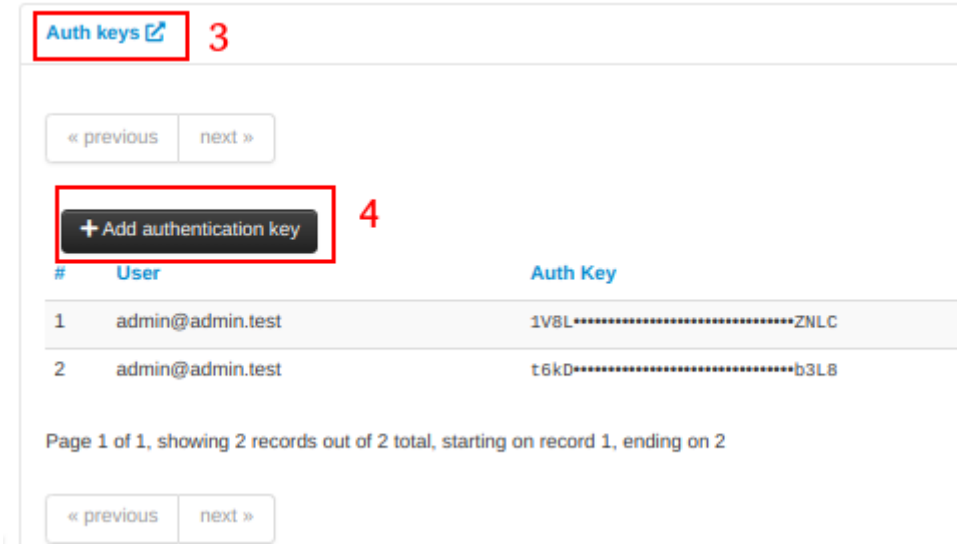


Yönetici Arayüzü:

2. **Yönetici Olarak Başka Bir Kullanıcı İçin API Anahtarı Oluşturma:**
   - Bu adımlar, yöneticinin başka bir kullanıcı adına API anahtarı oluşturmasını sağlar.
   - Yönetici, yönetici hesabına giriş yapmalı ve "Yönetim" sekmesine gitmelidir.
   - Ardından, "Kullanıcıları Listele" bölümüne tıklamalı ve istenen kullanıcının "Görünüm" sayfasına gitmelidir.

- Kullanıcının sayfasında, "Kimlik Doğrulama Anahtarları" bölümünde "Kimlik Doğrulama Anahtarı Ekle" seçeneğini seçmelidir.
- Bu adımları takip ederek, yönetici belirli bir kullanıcı adına API anahtarı oluşturabilir ve kullanıcıya iletebilir.





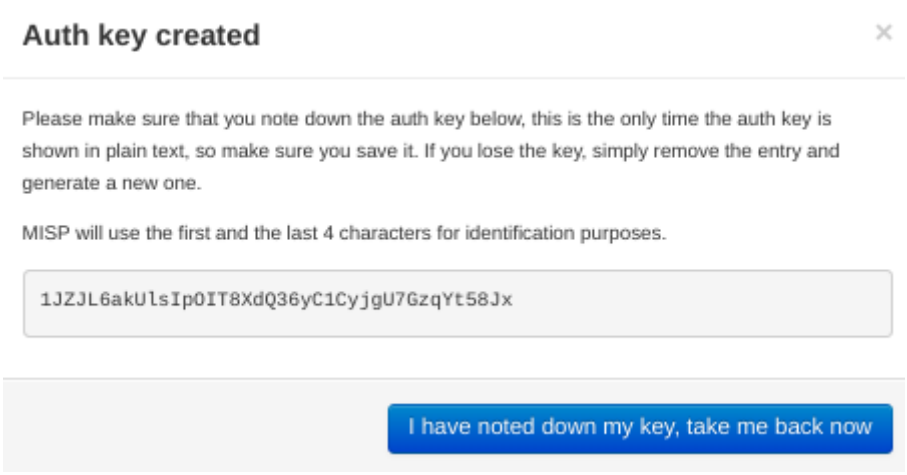
Kullanıcı Komut Satırı:

1. **Komut Satırı Kullanarak Kendi API Anahtarınızı Oluşturma:**
   - Kullanıcı, MISP'in yüklemesinin yapıldığı dizindeki CLI (Command Line Interface) aracını kullanarak API anahtarı oluşturabilir.
   - CLI aracını çalıştırmak için aşağıdaki komutu kullanabilir:
     `./app/Console/cake user change_authkey [e-posta/kullanıcı_kimliği]`
   - Bu komut, belirtilen kullanıcının API anahtarını değiştirir veya yeni bir API anahtarı oluşturur.

Yönetici Komut Satırı:

2. **Komut Satırı Kullanarak Başka Bir Kullanıcı İçin API Anahtarı Oluşturma:**
   - API Yönetici düzeyinde bir API anahtarınız olması koşuluyla, başka bir kullanıcı adına API anahtarı oluşturabilirsiniz.
   - Bu işlem için [POST]/auth_keys/add/{{user_id}} uç noktasını kullanabilirsiniz. Burada {{user_id}}, API anahtarı oluşturmak istediğiniz kullanıcının kimliğini belirtir.
   - Bu istek, API yöneticisi tarafından yetkilendirilmiş olmalı ve belirtilen kullanıcının API anahtarını oluşturmak için gerekli izinlere sahip olmalısınız.
   - Bu şekilde, başka bir kullanıcı adına API anahtarı oluşturabilir ve belirli bir kullanıcıya iletebilirsiniz.

> Kimlik doğrulama anahtarınız yalnızca bir kez görüntülenecek ve daha sonra tekrar erişilemeyecektir. Bu nedenle, anahtarı güvenli bir şekilde saklamanız önemlidir. Not almak veya güvenli bir parola yöneticisinde saklamak gibi uygun önlemler almanızı öneririz.



Bu anahtarlar, API'ye yetkilendirilmiş istekler göndermek için kullanılır ve kullanıcılara belirli bir güvenlik kimliği sağlar. API anahtarları, kullanıcıların MISP üzerinde belirli işlemleri otomatize etmelerine ve entegrasyonlar oluşturmalarına olanak tanır.

Rest Client Nedir?

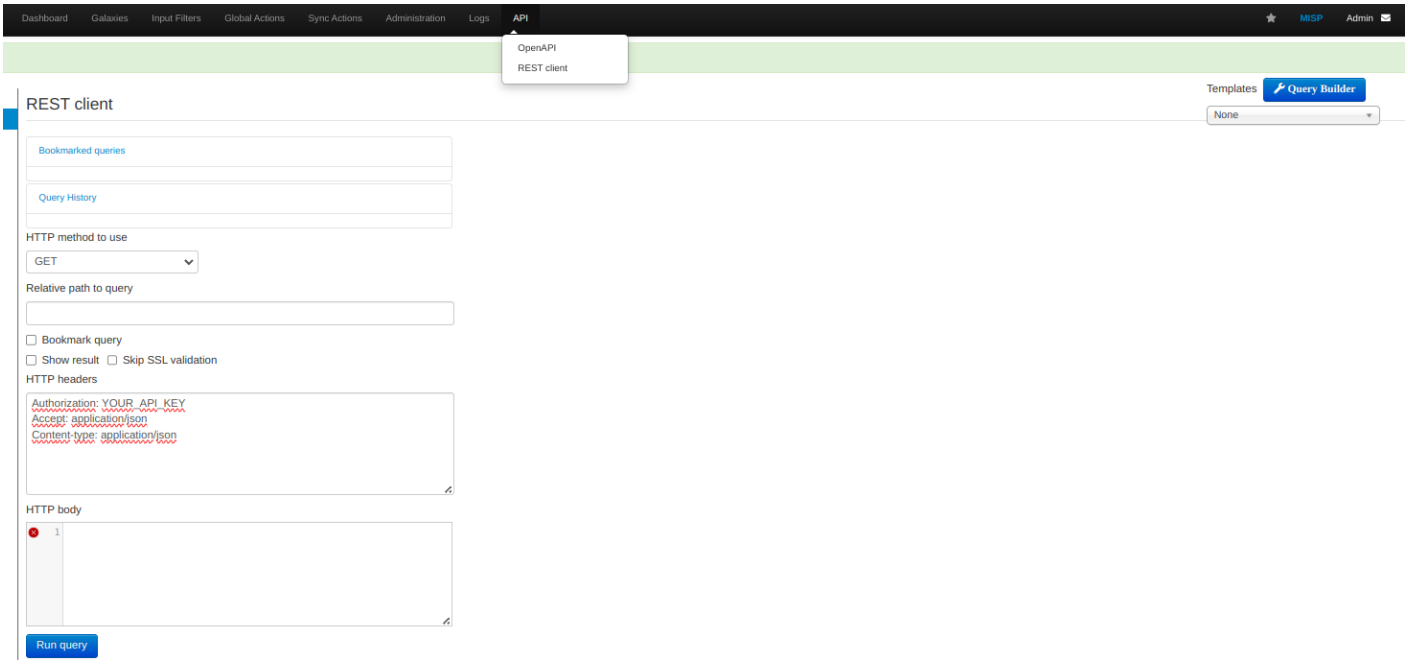
REST Client, MISP API'yi etkili bir şekilde kullanmak için tasarlanmış bir araçtır. Bu araç, MISP platformuyla etkileşim kurmak için REST (Representational State Transfer) protokolünü kullanır ve bu sayede çeşitli güvenlik senaryolarında veri alışverişi yapmak için bir arabirim sunar.

REST client, Representational State Transfer (REST) prensiplerine uygun bir şekilde çalışan bir istemcidir. Bu istemci, bir RESTful web servisiyle etkileşim kurmak için HTTP protokolünü kullanır.

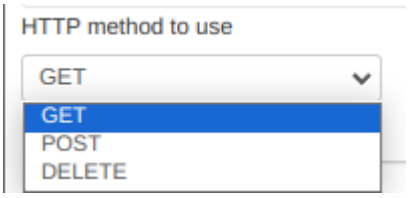
Kullanım senaryoları:

1. Veri Alma: MISP REST Client, MISP platformundan güncel tehdit bilgilerini almak için kullanılabilir. Tehdit istihbaratı paylaşımını güncel tutmak ve kuruluşun güvenlik durumunu izlemek için önemlidir.
2. Veri Gönderme: MISP REST Client, kuruluşun kendi tehdit istihbaratı verilerini MISP platformuna göndermesine olanak tanır. Kuruluşun kendi gözlemlerini diğer kuruluşlarla paylaşarak daha geniş bir tehdit görüşünü sağlamak için önemlidir.
3. Otomatik Analiz ve Uyarı: MISP REST Client, MISP platformundan alınan verileri otomatik olarak analiz ederek ve belirlenen kriterlere göre uyarılar oluşturarak güvenlik operasyonlarını otomatikleştirmek için kullanılabilir.
4. Entegrasyon: MISP REST Client, diğer güvenlik araçları ve sistemleriyle entegrasyon sağlamak için kullanılabilir. Bu, MISP platformunun güvenlik altyapısına kolayca entegre edilmesini ve çeşitli güvenlik araçlarının birlikte çalışmasını sağlar.

REST Client ile API İstekleri



- **Bookmarked Queries (Yer İmlenmiş Sorgular):**
  - Bookmarked Queries bölümü, daha önceden yapılan ve kullanıcı tarafından kaydedilmiş API sorgularını görüntülemek için kullanılır. Bu özellik, sıkça kullanılan sorguları kolayca erişilebilir bir konumda tutmak için kullanışlıdır.
- **Query History (Sorgu Geçmişi):**
  - Query History bölümü, daha önceden yapılan tüm API sorgularının geçmişini görüntülemek için kullanılır. Bu özellik, geçmişte yapılan sorguların takibini sağlar ve hata ayıklama veya tekrar kullanım için kullanıcıya referans oluşturur.
- **HTTP Method to Use (Kullanılacak HTTP Metodu):**
  - Bir API isteğinin hangi HTTP metodu kullanılarak yapılacağını belirler. Örneğin, GET, POST, PUT, DELETE gibi metotlardan biri seçilir. Bu, isteğin amacına ve API'nin desteklediği operasyonlara bağlıdır.



    - **GET**:
      - GET metodu, bir kaynağın okunması için kullanılır. Sunucudan belirtilen kaynağı almak için kullanılır. Örneğin, bir web sayfasını veya bir dosyayı almak için GET isteği yapılır.
    - **POST**:

        - POST metodu, bir kaynağa veri göndermek için kullanılır. Genellikle bir form gönderirken veya sunucuya veri kaydetmek için kullanılır. Örneğin, bir form doldurulduğunda ve gönderildiğinde, bu bilgiler POST isteğiyle sunucuya iletilir.
    - **DELETE**:
        - DELETE metodu, bir kaynağı silmek veya kaldırmak için kullanılır. Belirtilen kaynağın sunucu tarafından silinmesini istemek için kullanılır. Örneğin, bir dosyanın veya kaydın silinmesi için DELETE isteği yapılır.
- **Relative Path to Query (Sorgulanacak İlgili Yol):**
    - Bir isteğin gönderileceği URL'nin kök URL (Root URL)'ye göre konumunu belirten kısaltılmış bir yol ifadesidir. Bu, isteğin hangi endpoint'e yönlendirileceğini belirlemek için kullanılır. Göreceli yol (Relative Path), isteğin tam URL'sini oluşturmak için kök URL ile birleştirilir ve istenen endpoint veya kaynağın konumunu belirtir.
- **Bookmark Query (Sorguyu Yer İmleri Ekle):**
    - Bookmark Query özelliği, işaretlendiği takdirde oluşturulan bir sorgunun yer imlerine eklenmesini sağlar. Böylece, sıkça kullanılan veya önemli sorguların kolayca erişilebilir olmasını sağlar.
- **Show Result (Sonucu Göster):**
    - Show Result, API isteğinin sonucunun görüntülenmesini sağlar. işaretlendiği takdirde isteğin başarıyla tamamlanıp tamamlanmadığını ve alınan yanıtın içeriğini kullanıcıya gösterir.
- **Skip SSL Validation (SSL Doğrulamasını Atla):**
    - Skip SSL Validation özelliği, SSL sertifikası doğrulamasının atlanmasını sağlar. Bu, güvenli olmayan bir ortamda veya test amaçlı kullanımlarda gerekebilir, ancak genellikle tavsiye edilmez.
- **HTTP Headers (HTTP Başlıkları):**
    - HTTP headers, bir HTTP isteği veya yanıtı iletilirken kullanılan başlık alanlarıdır. Bu başlıklar, isteğin veya yanıtın ne olduğunu, nasıl işlendiğini ve ne tür veri içerdiğini belirtir. RESTful API'lerde, HTTP başlıkları önemli bilgiler içerebilir ve belirli işlevlerin gerçekleştirilmesini sağlar.

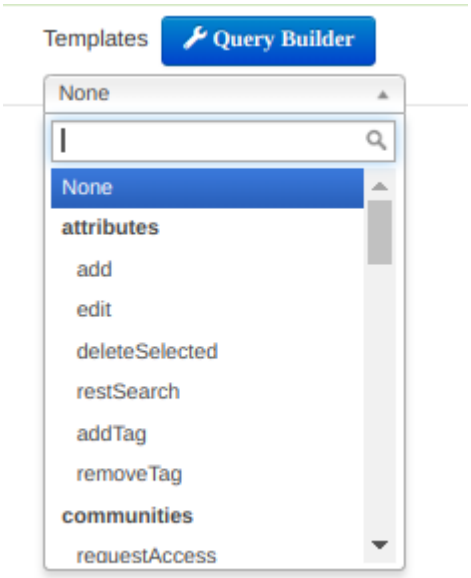


        - **"Authorization"** başlığı, bir API'ye yetkilendirme bilgilerini eklemek için kullanılır. Bu başlık, isteği gönderenin kimliğini doğrulamak ve yetkilendirilmiş erişim sağlamak için kullanılır.
        Örneğin, bir API'ye erişmek için gereken API anahtarını (API key) belirtmek için kullanılır:
        `Authorization: YOUR_API_KEY`

- **"Accept"** başlığı, istemci tarafından sunulan veri biçimini belirlemek için kullanılır. Sunucuya, istemcinin hangi medya türlerini kabul edebileceğini bildirmek için kullanılır. Bu, sunucunun yanıt olarak hangi medya türlerini gönderebileceğini belirler.
  Örneğin, istemci uygulama JSON formatında veri almak istiyorsa, aşağıdaki gibi bir başlık kullanır:
  `Accept: application/json`
- **"Content-Type"** başlığı, istemcinin sunucuya gönderdiği verinin türünü belirtmek için kullanılır. Sunucuya gönderilen verinin hangi medya türünde olduğunu belirtir. Örneğin, istemci bir POST, PUT veya DELETE isteği gönderirken veriyi JSON formatında kodladıysa, aşağıdaki gibi bir başlık kullanır:
  `Content-Type: application/json`

- **HTTP Body (HTTP Gövdesi):**
  - HTTP Body, isteğin gövdesine eklenen veri veya bilgileri sunucuya iletmektir. Bu veri, genellikle istemcinin sunucuya iletmek istediği kaynakların detaylarını veya işlem yapılacak verileri içerir.
    Örneğin, bir tehdit istihbaratı raporu oluşturmak için MISP API'ye bir POST isteği gönderirken, HTTP gövdesi bu raporun içeriğini taşır.
    Rapor JSON formatında gönderilmek istenirse, HTTP gövdesi aşağıdaki gibi bir yapıda olabilir:

```
{
 "title": "Örnek Rapor",
 "description": "Bu bir örnek rapordur.",
 "tags": ["tehdit", "analiz"],
 ...
}
```

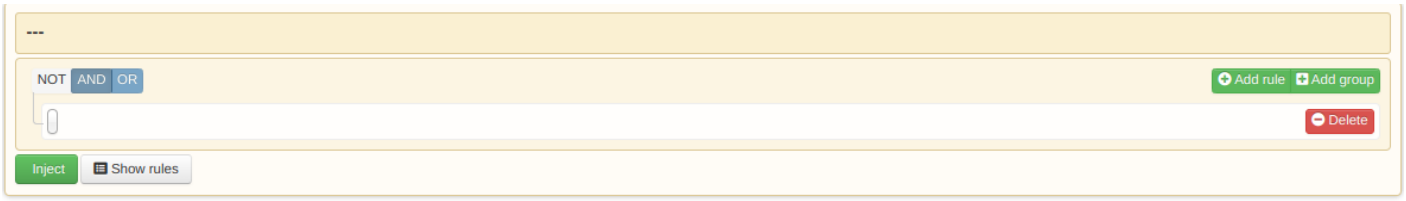


- **Templates (Şablonlar):**
  - Şablonlar, daha önceden tanımlanmış ve genellikle yaygın olarak kullanılan API isteklerinin önceden yapılandırılmış formatlarını içerir. Kullanıcılar, belirli bir şablonu seçerek, istek için gereken parametreleri doldurarak hızlıca bir API isteği

oluşturabilirler. Örneğin, "Event Search", "Attribute Creation" gibi şablonlar bulunabilir.

- **Query Builder (Sorgu Oluşturucu):**
  - Sorgu Oluşturucu, kullanıcıların bir API isteği için gerekli olan parametreleri adım adım seçmelerine olanak tanır. Kullanıcılar, belirli bir endpoint veya operasyon için gerekli olan parametreleri seçerek veya doldurarak isteklerini yapılandırabilirler. Kullanıcıların API'yi kullanırken gereken parametreleri hatırlamalarına veya manuel olarak yazmalarına gerek kalmadan kolayca istek oluşturmalarını sağlar.

> Sorgu oluşturucu butonuna tıklandığı zaman yeni bir alan açılır.



Kullanıcının daha karmaşık ve özelleştirilmiş sorgular oluşturmasına olanak tanıyan bir araçtır. Bu alanda kullanıcılar, isteklerini daha fazla filtrelemek veya belirli koşulları karşılayan verileri sorgulamak için kapsamlı sorgu kuralları oluşturabilirler.

Örneğin, bir kullanıcı belirli bir tarihten sonra oluşturulan etkinlikleri veya belirli bir tehdit seviyesine sahip olanları filtrelemek istiyorsa, query builder aracını kullanarak bu koşulları belirtebilirler. Ayrıca, bu araç sayesinde birden fazla koşulu birleştirerek daha karmaşık sorgular da oluşturulabilir.

Bu yeni alan, kullanıcılara API isteklerini daha esnek ve özelleştirilmiş bir şekilde oluşturma imkanı sunar ve istenen verilere daha doğru bir şekilde erişmelerini sağlar.

PyMISP ile Otomasyon

PyMISP - MISP'e Erişmek İçin Python Kütüphanesi

PyMISP, MISP platformlarına Python programlama dili aracılığıyla REST API'leri kullanarak erişim sağlayan bir kütüphanedir. Bu kütüphane, MISP platformları ile etkileşimi kolaylaştırır ve otomasyon için bir arayüz sunar.

PyMISP'nin Sağladığı Yetenekler:

PyMISP, MISP platformlarındaki olaylara ve verilere erişimi sağlar ve çeşitli işlemleri gerçekleştirmenizi sağlar. Bu yetenekler arasında şunlar bulunur:

- Etkinliklerin eklenmesi, alınması, güncellenmesi, yayımlanması ve silinmesi
- Etiketlerin eklenmesi veya kaldırılması
- Dosya özniteliklerinin eklenmesi: karma, kayıt defteri anahtarı, desenler, kanal, muteks
- Ağ özniteliklerinin eklenmesi: IP hedefi/kaynağı, ana bilgisayar adı, etki alanı, URL, UA, ...
- E-posta özniteliklerinin eklenmesi: kaynak, hedef, konu, ek, ...
- Örneklerin yüklenmesi/indirilmesi
- Görüntülenmelerin güncellenmesi
- Tekliflerin eklenmesi, düzenlenmesi, kabul edilmesi ve silinmesi
- Tam metin araması ve niteliklere göre arama
- STIX etkinliklerinin alınması
- İstatistiklerin dışa aktarılması ve daha fazlası (api.py dosyasına bakınız)

Kurulum:

PyMISP'yi pip kullanarak veya GitHub deposundan en son sürümü alarak yükleyebilirsiniz. Kurulum talimatlarına aşağıdaki şekillerde ulaşabilirsiniz:

- Pip ile kurulum: `pip install pymisp`
- GitHub'dan en son sürümü yükleme: `git clone https://github.com/MISP/PyMISP.git && cd PyMISP` ve `python setup.py install`

PyMISP kütüphanesini kullanabilmek için MISP örneğinizde bir Kimlik Doğrulama Anahtarı'na ihtiyacınız olacaktır.

Başlarken:

PyMISP'yi kullanmaya başlamadan önce, MISP otomasyon anahtarınızı almanız gerekmektedir. Otomasyon anahtarınızı MISP web arayüzündeki otomasyon bölümünde veya profilinizde bulabilirsiniz.

PyMISP kütüphanesini kullanarak örnekler çalıştırmak için, `git clone https://github.com/MISP/PyMISP.git` komutunu kullanarak depoyu klonlayabilir ve örnekler klasöründeki `keys.py` dosyasını düzenleyerek MISP örneğinizin URL'sini ve otomasyon anahtarınızı belirtebilirsiniz.

PyMISP Kullanımı:

PyMISP'nin kullanımını daha iyi anlamak için mevcut örneklerden birine bakalım: `add_named_attribute.py`. Bu komut dosyası, sadece türünü bildiğiniz bir özniteliği mevcut bir etkinliğe eklemenizi sağlar (kategori varsayılan olarak belirlenir).