**🧠 1. HTTP Request Smuggler Nedir?**

**HTTP Request Smuggler**, Burp Suite için geliştirilmiş, HTTP Request Smuggling zafiyetlerini **otomatik olarak tespit eden** bir güvenlik testi eklentisidir.

📌 Özellikleri:

* CL.TE, TE.CL, TE.TE gibi farklı parsing uyumsuzluklarını test eder
* HTTP başlık manipülasyonunu otomatik yapar
* Burp üzerinden entegre çalışır, ek terminal gerekmez
* Detaylı zafiyet raporu verir

**🎯 2. HTTP Request Smuggling Nedir?**

HTTP Request Smuggling, **ön uç sunucu (proxy, load balancer)** ile **arka uç sunucu** arasındaki HTTP istek yorumlama farklarını istismar ederek, isteklerin “gizlice” gönderilmesini sağlayan bir saldırı tekniğidir.

En yaygın türleri:

* **CL.TE** → Content-Length + Transfer-Encoding uyumsuzluğu
* **TE.CL** → Transfer-Encoding + Content-Length uyumsuzluğu
* **TE.TE** → Çift Transfer-Encoding manipülasyonu

**⚙️ 3. Kurulum (Burp Suite)**

**✅ 1. BApp Store üzerinden yükleme:**

* **Burp Suite → Extender → BApp Store → HTTP Request Smuggler → Install**

**✅ 2. Manuel yükleme (GitHub):**

ekran görüntüsü, yazı tipi, grafik, metin içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.Burp Extender üzerinden **Add → JAR** seçeneği ile yüklenir.

**🚀 4. Kullanım Adımları**

**✅ 1. Hedef isteği seçme:**

* HTTP history’den isteğe sağ tık → **Extensions → HTTP Request Smuggler → Smuggle Probe**

**✅ 2. Otomatik test:**

* Eklenti farklı HTTP başlık kombinasyonlarını dener.
* Zafiyet varsa konsolda veya Burp Results sekmesinde görünür.

**✅ 3. Manuel test:**

* **Smuggle Attack** seçeneği ile kendi payload’ınızı ekleyebilirsiniz.

**📋 5. Örnek Çıktı**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.Burp arayüzünde ayrıca **severity** (kritiklik seviyesi) ve **etki** raporu gösterilir.

**🧠 6. Kullanım Senaryoları**

| **Senaryo** | **Açıklama** |
| --- | --- |
| 🔍 Web güvenlik testi | HRS zafiyetlerinin hızlı tespiti |
| 🚀 WAF/Proxy bypass | Filtre atlatma senaryoları |
| 🧠 Red Team keşfi | Backend’e doğrudan erişim sağlama |
| 🔍 Cache poisoning | Yanıltıcı veri cache’e yükleme |

**🎯 7. Öğrenim Kazanımları**

* HTTP başlıklarının nasıl çalıştığını ve parsing farklarını öğrenme
* CL.TE, TE.CL gibi zafiyet tiplerini tanıma
* Burp eklentisi ile otomatik HRS tespiti yapma
* Web güvenliği testlerinde proxy/CDN yapılarını analiz etme