

OWASP Top 10 - Vulnerabilities and Mitigations

SoCyber

SoCyber is focused on protecting its clients. Since the first day, our primary mission is to make you feel secure and protected. Our team of experts will perform all required tests to reveal the weaknesses in your information systems, train your employees about the basic cybersecurity principles, consult you on topics related to the security of your infrastructure, and protect you from malicious attacks. Keeping your sensitive information secure is our top priority.



Who am I

- ▶ Pentester at SoCyber
- ▶ Casual CTF player
- ▶ @GitHub

These slides

<https://github.com/SoCyber/owasp10-hands-on>

The Plan

- ▶ Work on **Vulnerable App**
- ▶ Walk through OWASP Top 10 vulnerabilities until time is up



SoCyber
makes you feel secure

The Vulnerable App

- ▶ Found at [Awesome Vulnerable Web Applications](#)
- ▶ <http://hackyourselffirst.troyhunt.com/>
- ▶ Uses .Net and MSSQL

OWASP Top 10

What's OWASP?

- ▶ OWASP Top 10 Project
- ▶ Open Web Application Security Project



SoCyber
makes you feel secure

What's OWASP?

“Although the original goal of the OWASP Top 10 project was simply to raise awareness amongst developers and managers, it has become the de facto application security standard”



SoCyber
makes you feel secure

What's new in 2017?

- ▶ New:
 - ▶ XXE
 - ▶ Insecure Deserialization
 - ▶ Insufficient Logging and Monitoring
- ▶ Gone:
 - ▶ CSRF 🥲
 - ▶ Unvalidated Redirects and Forwards



SoCyber
makes you feel secure

A1 - Injection



SoCyber
makes you feel secure

Description

- ▶ User supplied data interpreted as code/query/something else
- ▶ SQLi, LDAPi, OSi, etc.



Example

<http://hackyourselffirst.troyhunt.com/CarsByCylinders?Cylinders=V12%27>
.. looks like an SQLi vulnerability

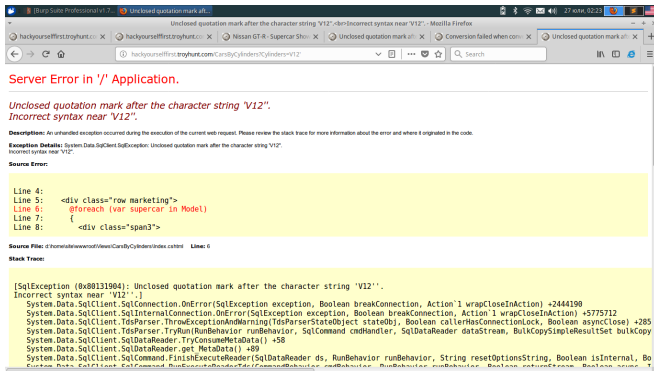


Figure 1: SQL error stack trace



SoCyber
makes you feel secure

Example

Tricky error based SQLi .. look closer



SoCyber
makes you feel secure

Example

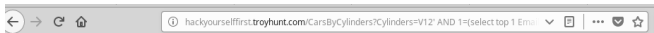
The email

```
V12' AND 1=(select top 1 Email from UserProfile where  
UserId=(select top 1 UserId from(select top 1 UserId  
from userprofile order by UserId) sq  
order by UserId DESC))--
```



SoCyber
makes you feel secure

Example



Server Error in '/' Application.

Conversion failed when converting the nvarchar value 'troyhunt@hotmail.com' to data type int.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: Conversion failed when converting the nvarchar value 'troyhunt@hotmail.com' to data type int.

Source Error:

```
Line 4:
Line 5:     <div class="row marketing">
Line 6:         @foreach (var supercar in Model)
Line 7:         {
Line 8:             <div class="span3">
```

Source File: d:\home\site\wwwroot\Views\CarsByCylinders\Index.cshtml **Line:** 6

Stack Trace:

```
[SqlException (0x80131904): Conversion failed when converting the nvarchar value 'troyhunt@hotmail.com' to data type int.]
System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 asyncCallback)
System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 asyncCallback)
System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callPostProc)
```

Figure 2: Get the email



SoCyber
makes you feel secure

Example

The password

```
V12' AND 1=(select top 1 password from UserProfile where  
UserId=(select top 1 UserId from(select top 1 UserId  
from userprofile order by UserId) sq  
order by UserId DESC))--
```



SoCyber
makes you feel secure

Example



The screenshot shows a web browser window with the address bar displaying `hackyourselffirst.troyhunt.com/CarsByCylinders?Cylinders=V12' AND 1=(select top 1 pass...`. The main content area displays a red heading "Server Error in '/' Application." followed by a red italicized message: "Conversion failed when converting the nvarchar value 'igzuwwaa' to data type int." Below this, a "Description" block states: "An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated." An "Exception Details" block shows: "System.Data.SqlClient.SqlException: Conversion failed when converting the nvarchar value 'igzuwwaa' to data type int." A "Source Error" section highlights a code snippet in a yellow box:

```
Line 4:
Line 5: <div class="row marketing">
Line 6:     @foreach (var supercar in Model)
Line 7:     {
Line 8:         <div class="span3">
```

Below the code, the "Source File" is listed as `d:\home\site\wwwroot\Views\CarsByCylinders\index.cshtml` at "Line: 6". A "Stack Trace" section follows, also in a yellow box, showing the exception details and the stack trace path:

```
[SqlException (0x80131904): Conversion failed when converting the nvarchar value 'igzuwwaa' to data
System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Acti
System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnecti
System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean c
```

Figure 3: Get the password

Mitigations

- ▶ Use a safe API
- ▶ SQL in particular: use **prepared statements**
- ▶ Make “whitelist” server side input validation
- ▶ Escape special characters according context



A2 - Broken Authentication

Description

- ▶ Allowing attackers to compromise passwords, keys, or session tokens
- ▶ Permits brute force or other automated attacks
- ▶ Does not rotate Session IDs after successful login
- ▶ Uses weak or ineffective credential recovery
- ▶ etc.



Example

A few write-ups:

- ▶ SAML Bug in Github worth 15000
- ▶ Bypassing Google Authentication on Periscope's Administration Panel



SoCyber
makes you feel secure

Mitigations

- ▶ Don't reinvent session management
- ▶ Default credentials smoke tests on deploy
- ▶ Account enumeration hardening
- ▶ Limit failed login attempts
- ▶ Implement multi-factor authentication



SoCyber
makes you feel secure

A3 - Sensitive Data Exposure

Description

- ▶ Cleartext submission of credentials
- ▶ Sensitive data stored in plain text
- ▶ Broken cryptography

Example

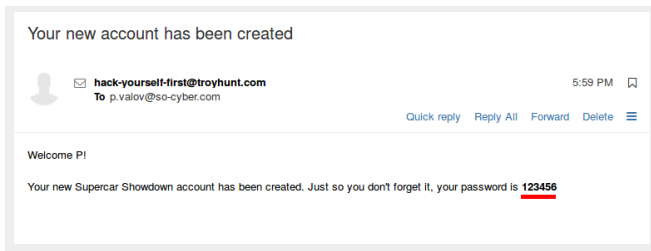


Figure 4: Plain text abuser!

Example

http:
//hackyourselffirst.troyhunt.com//api/admin/users

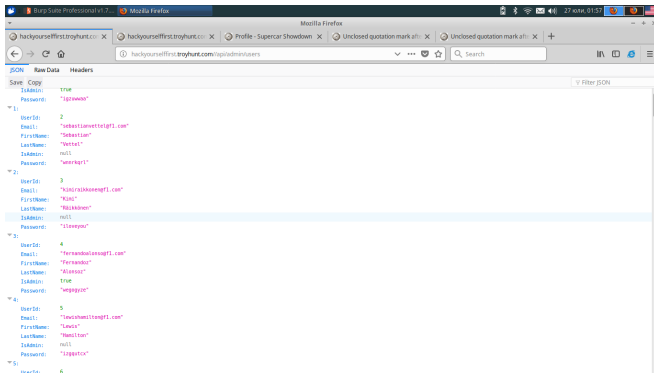


Figure 5: Exposure of other user credentials

Mitigations

- ▶ Don't store sensitive data unnecessarily
- ▶ Make sure to encrypt all sensitive data at rest
- ▶ Use standard crypto the proper way (huge topic!)



SoCyber
makes you feel secure

Mitigations

- ▶ Long but important:

*“Store passwords using strong adaptive and salted hashing functions with a work factor (delay factor) - **Argon2**, **scrypt**, **bcrypt**, or **PBKDF2**”*

- ▶ **Do not** hash passwords with md5/sha1/sha256/.. - they are not designed for that



SoCyber
makes you feel secure

A4 - XML External Entities (XXE)



SoCyber
makes you feel secure

Description

“This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser”



SoCyber
makes you feel secure

Description

- ▶ User input is supplied to XML processor
- ▶ XML processors has document type definitions (DTDs) enabled
- ▶ SOAP prior to version 1.2 is probably vulnerable



Example

- ▶ The most famous one: Facebook's XXE Bug-bounty

Mitigations

- ▶ [XXE Prevention Cheatsheet](#)
- ▶ Use less complex data formats (JSON)
- ▶ Disable XML external entity and DTD processing



A5 - Broken Access Control

Description

- ▶ Bypassing access control checks by modifying internal application state
- ▶ Elevation of privilege
- ▶ CORS misconfiguration allows unauthorized API access
- ▶ Force browsing (change ID in link or hit “hidden” resource)



Example

884	https://hackyourselffirst.tr...	POST	/Account/Login	✓	302	949	HTML
882	http://hackyourselffirst tro...	GET	/Account/Login		200	5960	HTML
880	http://hackyourselffirst tro...	GET	/		200	7819	HTML
879	http://hackyourselffirst tro...	POST	/Account/LogOff		302	515	HTML
877	http://hackyourselffirst tro...	GET	/Account/UserProfile/66		200	5186	HTML
875	http://hackyourselffirst tro...	GET	/Search?searchTerm=aada%27ad		200	4100	HTML

Request	Response
Raw	Headers

```
Server: Microsoft-IIS/10.0
Set-Cookie:
AuthCookie=D8B3AFF36DFE281DC6ADF9933A5CFA7D8A707C873C4F5E8932B6234EDEDFAEEFA90F7CB4AD590E7D7AAED0A50A8EAGC:
ACEFODAC05D3B6BA57C65268D4769D8967C4940BD427C945CFA71FBF85B8F250128F1BB38462A3; expires=Fri, 26-Jul-2019 22:
Set-Cookie: IsAdmin=false; path=/
Set-Cookie: Password=MTIaNDU2; expires=Fri, 26-Jul-2019 22:23:28 GMT; path=/
Set-Cookie: Email=p.valov@so-cyber.com; expires=Fri, 26-Jul-2019 22:23:28 GMT; path=/
X-XSS-Protection: 0
X-AspNetMvc-Version: 5.1
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Thu, 26 Jul 2018 22:23:28 GMT
Connection: close

<html><head><title>Object moved</title></head><body>
<h2>Object moved to <a href="/">here</a>.</h2>
</body></html>
```

Figure 6: Cookie manipulation and privilege escalation

Example

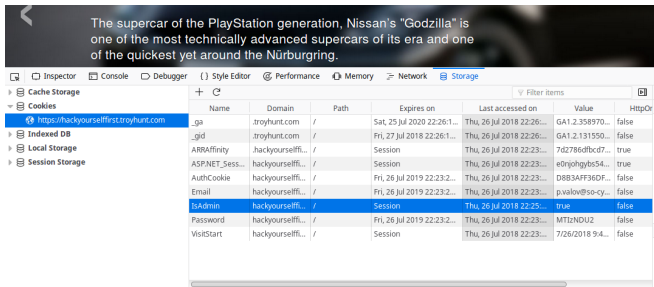


Figure 7: Cookie manipulation and privilege escalation

Example

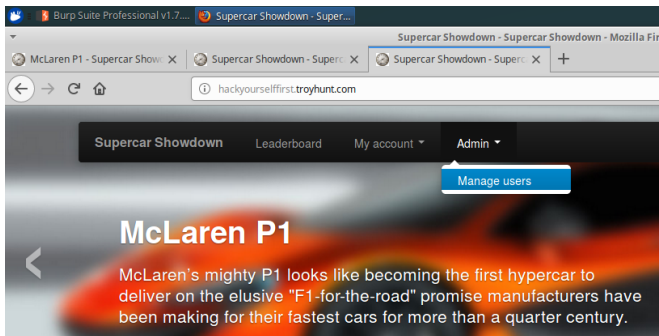


Figure 8: Cookie manipulation and privilege escalation

Mitigations

- ▶ Deny-by-default policy for non-public content
- ▶ Disable web server directory listing
- ▶ Ensure sensitive data is not in web roots
- ▶ JWT tokens should be invalidated on the server after logout



A6 - Security Misconfiguration



SoCyber
makes you feel secure

Description

- ▶ Default accounts/passwords
- ▶ DEBUG is turned on
- ▶ Lack of Security Headers
- ▶ .. much can fit here



SoCyber
makes you feel secure

Example

<http://hackyourselffirst.troyhunt.com/Make/1?orderby=supercarid%27>

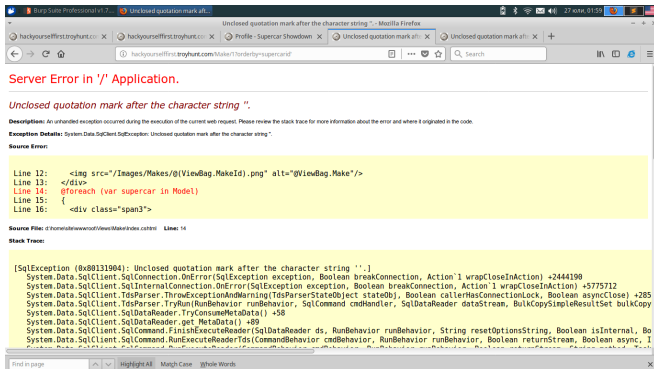


Figure 9: Stack traces enabled

Mitigations

- ▶ A minimal platform without any unnecessary features
- ▶ Good QA process



SoCyber
makes you feel secure

A7 - Cross-Site Scripting (XSS)



SoCyber
makes you feel secure

Description

- ▶ Basically *A1: Injection*, but so popular you can't ignore it
- ▶ User input is interpreted as JavaScript in the browser
- ▶ Stored, reflected and DOM-based XSSes



Example

<http://hackyourselffirst.troyhunt.com/Supercar/2>

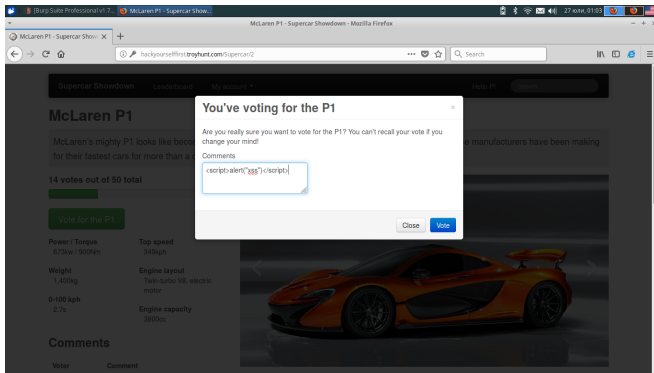


Figure 10: Stored XSS

Example

Payload:

```
<script>alert("xss")</script>
```

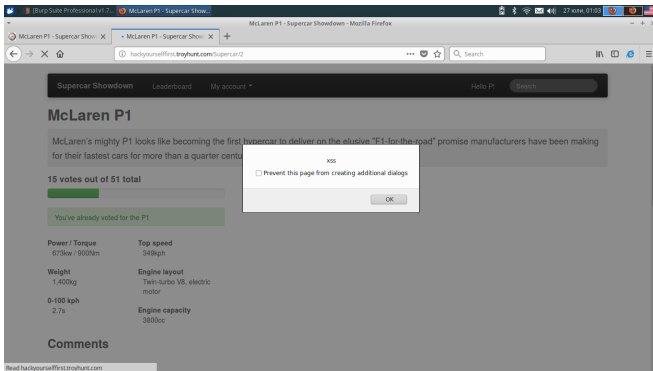


Figure 11: Stored XSS

Mitigations

- ▶ Escape special characters according context
- ▶ Use HTML entity encoding
- ▶ Watch user input when dynamically create HTML by JS



A8 - Insecure Deserialization



SoCyber
makes you feel secure

Description

Tricky to exploit, just like A4: *XXE*



SoCyber
makes you feel secure

Description

- ▶ Vulnerable deserialization of user controlled serialized objects
- ▶ Serialized objects usually are seen in:
 - ▶ HTTP cookies
 - ▶ form parameters
 - ▶ etc.
- ▶ When object tampering validation is broken



Example

A few write-ups:

- ▶ RCE at PayPal
- ▶ Instagram's Million Dollar Bug
- ▶ Exploiting Java Deserialization



SoCyber
makes you feel secure

Mitigations

- ▶ Implementing integrity checks
- ▶ Monitor deserialization



SoCyber
makes you feel secure

A9 - Using Components with Known Vulnerabilities



SoCyber
makes you feel secure

Description

When you missed security updates

Mitigations

- ▶ Continuously inventory the versions of both client-side and server-side components
- ▶ Monitor for security updates
- ▶ Patch regularly

A10 - Insufficient Logging and Monitoring



SoCyber
makes you feel secure

Description

- ▶ No log for authentication/authorization events
- ▶ No fraud detection based on activities
- ▶ etc.

Mitigations

- ▶ Log enough user context for sensitive events
- ▶ Implement fraud detection mechanisms
- ▶ Establish or adopt an incident response and recovery plan



SoCyber
makes you feel secure

Q&A



SoCyber
makes you feel secure

The End