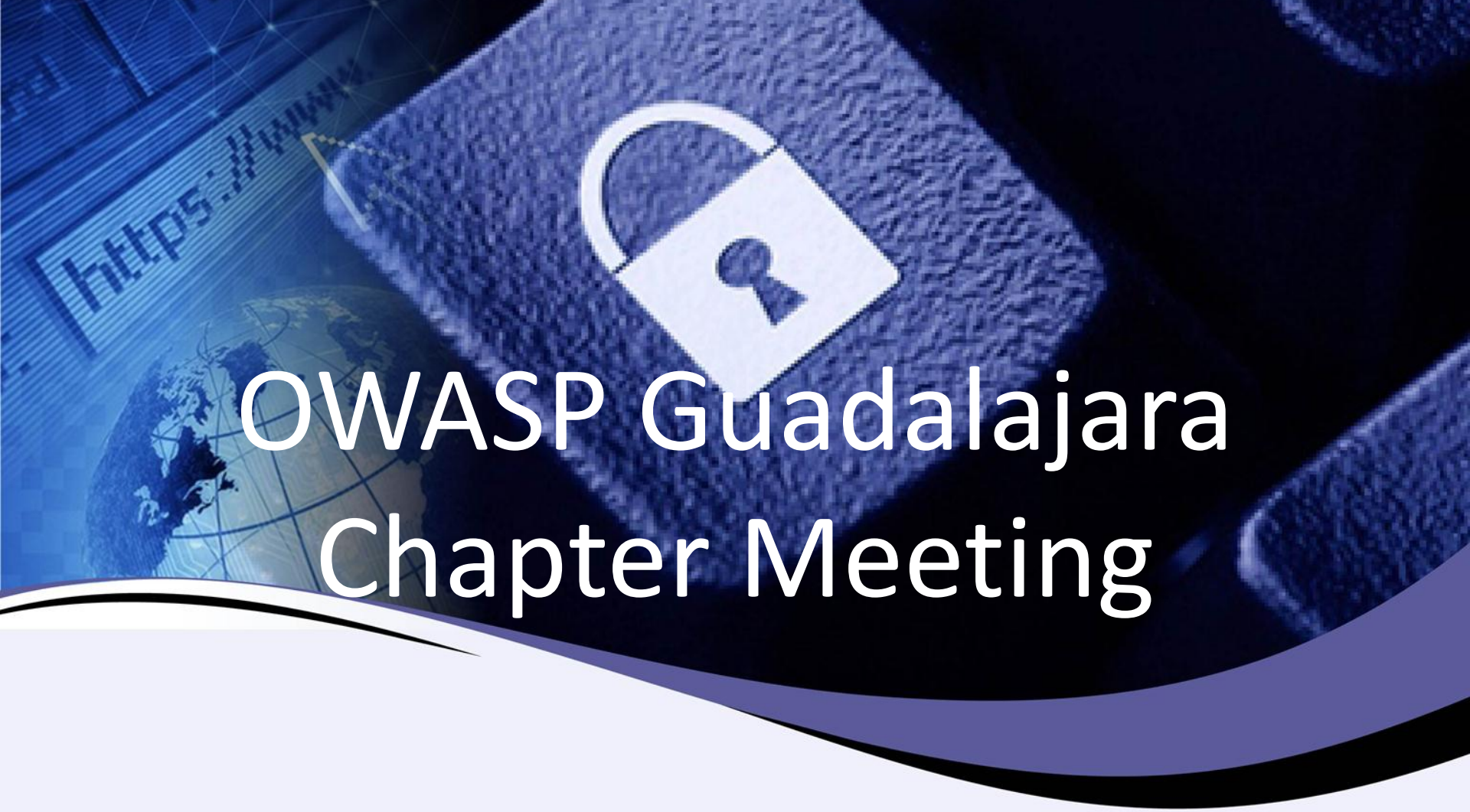


OWASP Guadalajara Chapter Meeting

OWASP

The Open Web Application Security Project

Introduction

OWASP
The Open Web Application Security Project

- Sponsor this evening:
  - Intel
- Call for additional sponsors
  - Chapter meeting places & catering
  - Support for local projects
- OWASP cannot recommend the use of products, services, or recommend specific companies

Program for this evening

OWASP
The Open Web Application Security Project

- 18h30 - 18h45:

  Manuel Lopez Arredondo, Chapter Leader
  OWASP Overview and Update

- 18h45 - 19h10:

  Somen Das, Application Security Analyst for Tata Consultancy Services Ltd.
  Cross Site Request Forgery (CSRF) - The confused deputy problem

- 19h10 - 19h20: Break

- 19h20 - 20h00:

  Eduardo Cerna, Chapter Leader

  Developing Secure Source Code (First Part)

Program for this evening

OWASP
The Open Web Application Security Project

- 18h30 - 18h45:

  Manuel Lopez Arredondo, Chapter Leader
  OWASP Overview and Update

- 18h45 - 19h10:

  Somen Das, Application Security Analyst for Tata Consultancy Services Ltd.
  Cross Site Request Forgery (CSRF) - The confused deputy problem

- 19h10 - 19h20: Break

- 19h20 - 20h00:

  Eduardo Cerna, Chapter Leader
  Developing Secure Source Code (First Part)

Mission

OWASP
The Open Web Application Security Project

The Open Web Application Security Project (OWASP) is a 501c3 not-for-profit worldwide charitable organization focused on improving the security of application software. Our mission is to make application security visible, so that people and organizations can make informed decisions about true application security risks.

Everyone is free to participate in OWASP and all of our materials are available under a free and open software license.

Our Success

OWASP
The Open Web Application Security Project

- OWASP Tools and Documentation:
  - 15,000 downloads per month
  - 30,000 unique visitors per month
  - 2 million website hits per month
- OWASP chapters are blossoming worldwide:
  - 1,500+ OWASP Members in active chapters worldwide
  - 20,000+ participants
- OWASP AppSec Conferences:
  - Chicago, New York, London, Beijing, Brazil …(México??)
- Distributed Content Portal
  - 100+ authors for tools, projects, and chapters
- Global Citations (IEEE, NIST, PCI, NSA, CSA, DISA, DHS, FFIEC, CIS and more)
- Professional Association of Information Security Peers

OWASP Guadalajara

OWASP
The Open Web Application Security Project

- Quarterly Meetings
- Local Mailing List
- Presentations & Groups
- Open forum for discussion
- Meet fellow InfoSec professionals
- Create (Web)AppSec awareness in Guadalajara
- Local projects?

OWASP
The Open Web Application Security Project

- Free & open to everyone
- No vendor pitches or $ales presentations
- Respect for different opinions
- 1 CISSP CPE for each hour of OWASP chapter meeting

Hackfests and Docsfests

OWASP Training Days by GoToMeeting and instructor-lead
(https://www.owasp.org/index.php/OWASP_Training)

OWASP
The Open Web Application Security Project

https://lists.owasp.org/mailman/listinfo/owasp-guadalajara

Keep up to date!

OWASP Updates

OWASP
The Open Web Application Security Project

1. The most recent edition of the OWASP Newsletter is now posted! Be sure to catch up on events, articles, and information from the OWASP Foundation. https://www.owasp.org/images/d/d2/51516_OWASP_Newsletter-May2012_proof05.pdf

2. Registrations are open for AppSec Research in Greece and AppSec USA in Austin, TX. Be sure to visit the conference page for details on how to reserve your seat at the event. https://www.owasp.org/index.php/Category:OWASP_AppSec_Conference

3. OWASP Austin is hosting a Global Webinar!

What: OWASP Austin: Measuring Exposure: The Root Shell Index with HD Moore: Determining the realistic scope of a particular advisory or vulnerability using large scale reconnaissance with analytics.

Who: HD Moore - CSO of Rapid7 and Chief Architect of Metasploit

When: Tuesday, June 26th, from 11:45 AM -1:00 PM CST

Space is limited.
Reserve your Webinar Seat Now at:
https://www3.gotomeeting.com/register/221160910

OWASP Updates

OWASP
The Open Web Application Security Project

4. @RevistaSG #SGCE

OWASP Capítulo Guadalajara presente mañana. Software Gurú

http://sg.com.mx/sgce/2012/sessions/owasp-top-10-web-application-vulnerabilities

https://www.owasp.org/index.php/Guadalajara

5. OWASP Guadalajara to be registered as part of the “Intel Involved” program. Those Intel employees interested of being part of the program and participating on this project, please send an email to Brenda Fernandez del Castillo (brendax.fernandez.castillo@intel.com) or Manuel Lopez Arredondo (manuel.lopez.arredondo@intel.com)

Program for this evening

OWASP
The Open Web Application Security Project

- 18h30 - 18h45:

  Manuel Lopez Arredondo, Chapter Leader
  OWASP Overview and Update

- 18h45 - 19h10:

  Somen Das, Application Security Analyst for Tata Consultancy Services Ltd.
  Cross Site Request Forgery (CSRF) - The confused deputy problem

- 19h10 - 19h20: Break

- 19h20 - 20h00:

  Eduardo Cerna, Chapter Leader
  Developing Secure Source Code (First Part)

About Somen Das

OWASP
The Open Web Application Security Project

Application Security Analyst for Tata Consultancy Services Ltd. Specialized in Static & Dynamic application vulnerability assessment techniques, main focus is spreading awareness on secure application development and related guidelines across industry verticals. Local Chapter Leader - OWASP Bhubaneswar (India).

Cross Site Request Forgery (CSRF)

The confused deputy problem

OWASP

The Open Web Application Security Project

About Me

OWASP
The Open Web Application Security Project

- Somen Das – Application Security Analyst with TATA CONSULTANCY SERVICES for past $2^{1/2}$ years
- I lead the OWASP local chapter initiative at Bhubaneswar, India
- My roles & responsibilities include the following:
  - Lead a team of security engineers in geographically distributed TCS offices
  - Create threat models that result in more secure application design
  - Design and develop security testing scenarios
  - Analyze and present results of testing to team members, managers and customers
  - Write detailed problem reports, test plans documents and mitigation recommendations as needed
  - Develop tools to aid penetration test automation and effectiveness
  - Review code for common security vulnerabilities
  - Conducting training programs across industry verticals
  - Managing the OWASP Chapter in Bhubaneswar

TATA
TATA CONSULTANCY SERVICES

Agenda

OWASP
The Open Web Application Security Project

- Overview CSRF
- Sample example
- How to detect CSRF vulnerabilities
- Protecting your website against CSRF

Overview CSRF

OWASP
The Open Web Application Security Project

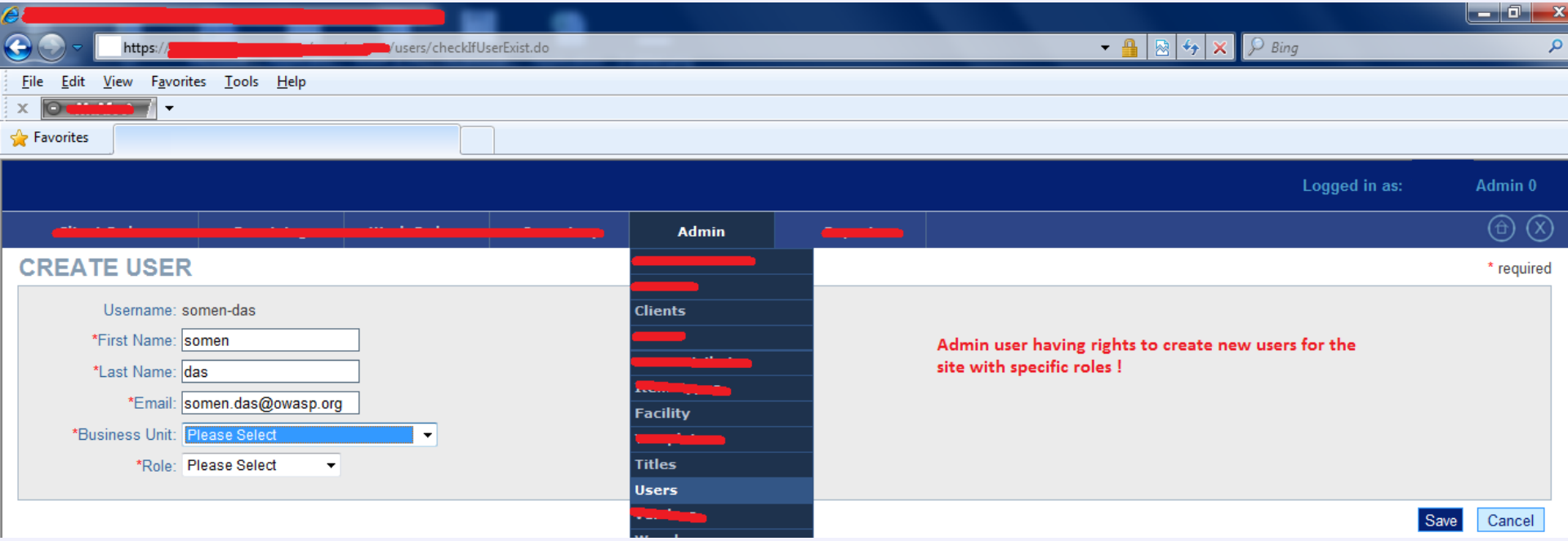
- CSRF takes advantage of the trust that a web-site has in a user's browser
- Generally works by embedding a link or script in a malicious page that accesses a web-site to which the user has already been authenticated to
- XSS != CSRF
- Preventing XSS do not put a stop on CSRF
- Requirement to exploit CSRF = The user must be logged into the website
- Web applications verify that a given browser is performing the said request by checking the cookies
- So every action a website allows a customer/user to do can be abused unless the action is CSRF protected

Sample Example

OWASP
The Open Web Application Security Project

DISCLAIMER

1. Hacking attempts on websites are illegal & are considered as cyber crime
2. All tricks & tips shown here is for educative purpose only
3. Use the techniques learnt to think as an adversary & better protect your web applications



Sample Example..

OWASP
The Open Web Application Security Project

https:/[redacted]/users/checkIfUserExist.do - Original Source

File Edit Format

```
<form name="userForm" id="userForm" action="[redacted]/users/createUser.do" method="post">
<input type="hidden" name="org.apache.struts.taglib.html.TOKEN" value="aa55dcb164fcb89d6487360ecd8615eb">
<input type="hidden" id="{actionForm.editUser}" name="{actionForm.editUser}" value="false">
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr valign="top">
    <td width="10"> </td>
                  <td class="Body1">
          <table width="100%" border="0" cellpadding="0" cellspacing="0">
    <tr>
                          <td align="left" valign="top" class="Title1">



                                Create

                          User</td>
                          <td align="right"> <span class="Body1"><span class="Required">*</span> required</span></td>
    </tr>

  </table>

    </td>
    <td width="10"> </td>
  </tr>

</table>
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr valign="top">
    <td width="10"> </td>
                  <td class="Body1">
        <table cellpadding="6" cellspacing="0" border="0">


              <tr>
                  <td  class="Success"><span></span></td>
              </tr>

          </table>
          <table width="100%" border="0" cellpadding="3" cellspacing="3" class="DataTable2">
          <tr><td  align="right" valign="top">
          <table width="100%" border="0" cellpadding="3" cellspacing="3" >
          <tr>
            <td width="120" align="right" class="FormLabel">Username:</td>
                <td class="DataText">somen-das</td>
            <input type="hidden" id="{actionForm.userId}" name="{actionForm.userId}" value="somen-das">
            <input type="hidden" id="{actionForm.encodedUserId}" name="{actionForm.encodedUserId}" value="somen-das">
```

Sample Example…

OWASP
The Open Web Application Security Project

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr valign="top">
    <td width="10"> </td>
    <td class="Body1">
        <table width="100%" border="0" cellpadding="6" cellspacing="0">
            <tr>
                <td width="25%">  </td><td width="50%" nowrap align="center" class="FormLabel"><!-- insert paging here --></td>
                <td width="25%" align="right"> 


                        <input type="button" class="Button" value="Save" onClick="Javascript:fnSubmitUser('N');">


                 
                <input type="button" class="ButtonInline" value="Cancel" onClick="Javascript:fnGoTo('cancelFromCreateEditUser.do','userForm');"></td></tr>
        </table>
    </td>
    <td width="10"> </td>
  </tr>
</table>
</form></html>
```

Sample Example….

OWASP
The Open Web Application Security Project

From: Das, Somen
To: Das, Somen
Cc:
Subject: CSRF Demo
Sent: Thu 6/14/2012 5:21 PM

Create the below URL & Send it to Admin User & hope Admin user clicks on this when they are logged into the site…

https://[redacted]/users/checkIfUserExist.do?{actionForm.userId}=SOMEN-DAS&{actionForm.firstName}=ETHICAL-HACKER&{actionForm.lastName}=TCOE&{actionForm.email}=somen.das@owasp.org&{actionForm.userRoleId}=2001&{actionForm.businessUnitId}=5&{actionForm.lineofBusiness}=3<script>fnSubmitUser('Y');</script>

Thanks & stay secure,
Somen Das

https://[redacted]/users/checkIfUserExist.do?%7bactionForm.userId%7d=SOMEN-DAS&%7bactionForm.firstName%7d=ETHICAL-H

File Edit View Favorites Tools Help

McAfee

Favorites

Logged in as: Admin 0

Admin

CREATE USER

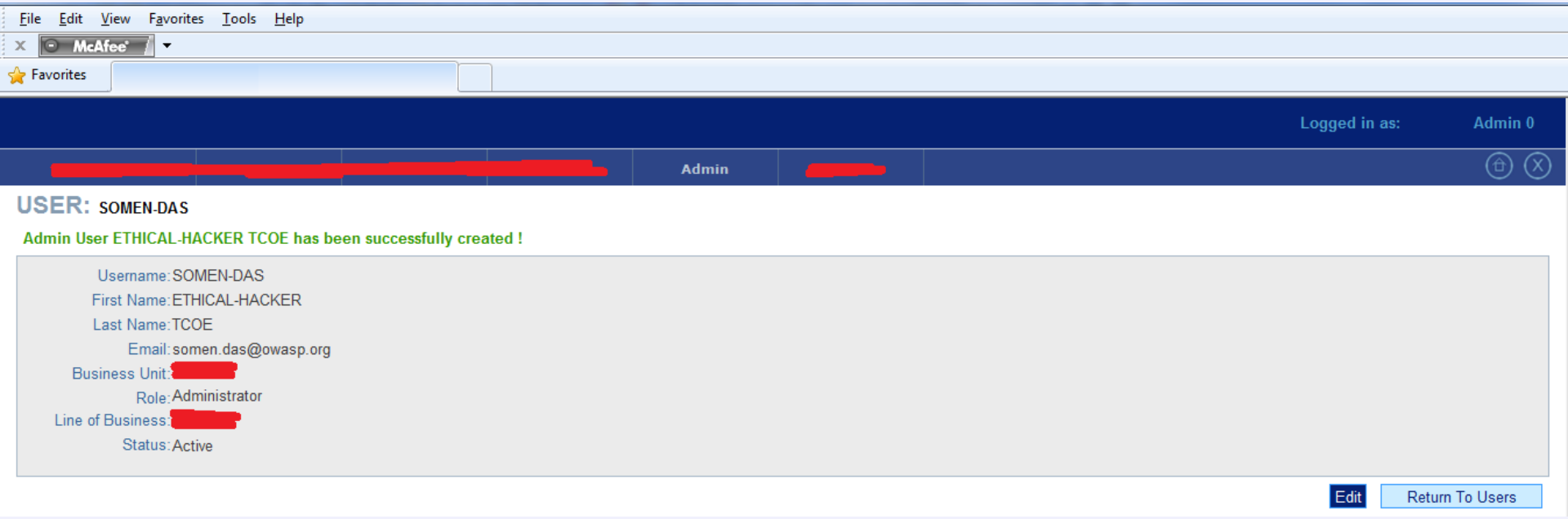
* required

Username: SOMEN-DAS
*First Name: ETHICAL-HACKER
*Last Name: TCOE
*Email: somen.das@owasp.org
*Business Unit: [redacted]
*Role: Administrator
*Line of Business: [redacted]

Save Cancel

Sample Example.....

OWASP
The Open Web Application Security Project



How to detect CSRF vulnerabilities

OWASP
The Open Web Application Security Project

1. Does the application use token to make each request unique ?
2. If answer to step1 is yes go to step 10
3. If answer to step 1 is no go to step 4
4. Use a CSRF redirector tool to test the website for csrf issues
5. Are you using third party software ?
6. If answer to step 5 is yes go to step 7 if no go to step 10
7. Look for csrf vulnerabilities for the software in Google
8. Use the following piece of code to create a html page having some background code to log the user out from the website in which he/she is logged in.
9. If user is successfully logged out in step 8 then the site is vulnerable to CSRF else go to step 10
10. Move on. The site is protected against CSRF attack

```
<html>
        <head>welcome to The example.com </head>
        <img src="http://XXXXX/TBPImages/_logo.jpg" height="100" width="100"
        onmouseover="window.location.href='http://XXXXX/LogOut.jsp';"/>
</html>
```

Protecting your website against CSRF

OWASP
The Open Web Application Security Project

- To make a Web application safe against CSRF, introduce a secret (hash, token) and put it into every link and every form, at least into every form that is sensitive
- The secret is generated after the user has logged in. It is stored in the server-side session. When receiving an http request that has been tagged with this information, compare the sent secret with the one stored in the session. If it is missing in the request or the two are not identical, stop processing the request and invalidate the session
- Use OWASP CSRF Guard to create unique token verification pattern
- Choose from two types of tokens:
  - Per Session Tokens
  - Per Request Tokens
- In case of Per Session Token’s, developers need to generate this token once for the current session where as in case of Per Request Token's the parameter name & value for the token changes for each request making it more secured
- For .NET based applications if view state is used in maintaining state then add the session id to every view state which will make each view state unique

Questions ?

Apendix

OWASP
The Open Web Application Security Project

- OWASP CSRF Cheat Sheet
- Testing for CSRF
- SESSION RIDING : A Widespread Vulnerability in Today's Web Applications
- CSRF Tester : OWASP_CSRFTester_Project

Program for this evening

OWASP
The Open Web Application Security Project

- 18h30 - 18h45:

  Manuel Lopez Arredondo, Chapter Leader
  OWASP Overview and Update

- 18h45 - 19h10:

  Somen Das, Application Security Analyst for Tata Consultancy Services Ltd.
  Cross Site Request Forgery (CSRF) - The confused deputy problem

- 19h10 - 19h20: Break

- 19h20 - 20h00:

  Eduardo Cerna, Chapter Leader

  Developing Secure Source Code (First Part)

Program for this evening

OWASP
The Open Web Application Security Project

- 18h30 - 18h45:

  Manuel Lopez Arredondo, Chapter Leader
  OWASP Overview and Update

- 18h45 - 19h10:

  Somen Das, Application Security Analyst for Tata Consultancy Services Ltd.
  Cross Site Request Forgery (CSRF) - The confused deputy problem

- 19h10 - 19h20: Break

- 19h20 - 20h00:

  Eduardo Cerna, Chapter Leader
  Developing Secure Source Code (First Part)

About Eduardo

OWASP
The Open Web Application Security Project

Information Security Engineer at Bank of America. Eduardo has over 15 years of experience in IT Management, Network Security and Operations. Core knowledge and skill areas include: Application Security, Vulnerability scanning, Intrusion Detection and Penetration Testing. (Black-Box, Grey-Box, White-Box).

OWASP
The Open Web Application Security Project

The Need For Secure Code

Part I

sabado 25, febrero 2012

b:Secure

BUSCAR

Inicio Internet Opinión Reportes Seguridad

8 February, 2012 | Carlos Fernández de Lara

Hackers filtran datos de 3,000 empleados de Telcel

Email | ¿Desea imprimir? Regístrese ahora

Follow @bsecuremagazine 3,194 followers

"Iluminati de México, somos Anonymous S
conocemos sus empresas y sabemos para
las palabras iniciales que preceden a la inf

El mensaje continua con una amenaza: "P
aquellos que atentan contra las garantías

SECTOR 404 HACKED BY:
@SestorLeaks_404
My name is PHANTOM

#OpIlimunati Sabemos quienes mueven este Pais, iremos poco a poco por cada uno de ustedes y sacaremos mucho mas datos ¿que es esto? no es nada, la SEP es NUESTRA! salu2 Mortales.

Host IP: 168.255.254.29 http://promep.sep.gob.mx/ca1/firmadoiesMEJORA.php?

What is Application Security?

OWASP
The Open Web Application Security Project

- The threats that are going to be discussed on this course are beyond our control and will always exist. The overall RISK of compromise is the product of these threats multiplied the likelihood of occurrence or existence of vulnerabilities.

OVERALL RISK = THREATS x VULNERABILITIES

- The goal of Application Security is to identify and eliminate (to the maximum extent possible) the vulnerabilities that exist within our Applications, Systems and Source Code. Risk can never be eliminated entirely.

Insecure Software Costs

OWASP
The Open Web Application Security Project

76% of Software and Applications Tested Have Serious Design and Implementation Flaws
- Foundstone Survey*

$60B in Cost to USA Economy from Poor Software Quality
–US Dept of Commerce

$3B Cost of Insecure Software to the Financial Services Industry
– NIST Survey 2002

100x 100 Times More Expensive to Fix Security Bug at Production Than Design
– IBM Systems Sciences Institute

¿Why Security?

OWASP
The Open Web Application Security Project

- Protect Assets (Proteger los Activos)
- Tangibles Assets (Proteger los Activos Materiales)
- Intangibles Assets (brand, the name , reputation) (Proteger marca, nombre, reputación)
- Company Assets (Proteger Activos de la Compañía)
- Clients Assets (Proteger Activos de los Clientes)

Security is the strategy, not the goal

¿Why Attackers Target Software?

OWASP
The Open Web Application Security Project

Objective	Property Compromised	Life Cycle Phase	Type of Software Susceptible
Modify/subvert the functionality of the software (often to more easily accomplish another compromise)	integrity	development or deployment	all systems
Prevent authorized users from accessing the software, or preventing the software from operating according to its specified performance requirements (known as "denial of service")	availability	deployment	all systems
Read data controlled, stored, or protected by the software that he/she is not authorized to read	confidentiality	deployment	information systems
Access to data, functions, or resources he/she is not authorized to access, or perform functions he/she is not authorized to perform	access control	deployment	all systems
Obtain privileges above those authorized to him/her (to more easily accomplish another compromise) (known as "escalation of privilege")	authorization	deployment	all systems

Security Process

OWASP
The Open Web Application Security Project

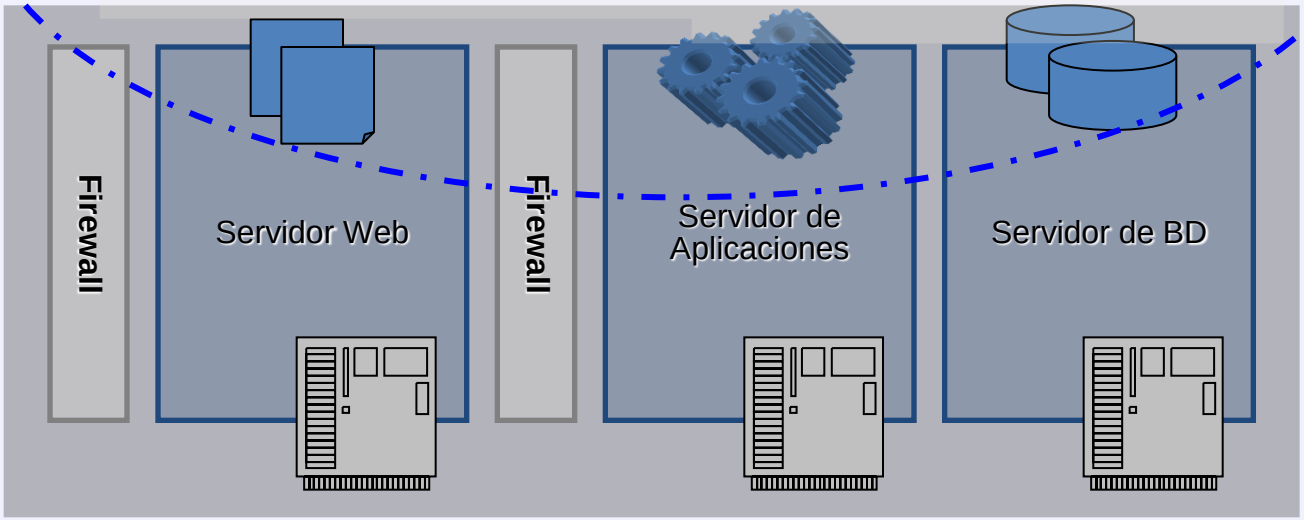
Aplicación

Validación de INput
Autenticación
Autorización
Gestión de Configuración
Datos Sensibles

Gestión de Sesiones
Criptografía
Manipulación de Parámetros
Gestión de Excepciones
Auditoría y Logging

Red

Router
Firewall
Switch



Servidor

Parches
Servicios
Protocolos

Cuentas
Ficheros y Directorios
Shares

Puertos
Registro
Auditoría y Logging

Security vs Vulnerability

OWASP
The Open Web Application Security Project

- Security- mitigating risk at a cost.
  (Seguridad – Mitigar riesgo a un costo)
- Vulnerability – exploitable fault
  (Vulnerabilidad – Avería Explotable)

This is for real! Who can Help me out?

- **OWASP**
- **WASC**
- **Security**
- **Other Security Consortiums**

OWASP Top 10 Vulnerabilities

OWASP
The Open Web Application Security Project

1.-Cross Site Scripting (XSS)
2.-Injection Flaws
3.-Malicious File Execution
4.-Insecure Direct Object Reference
5.-Cross Site Request Forgery (CSRF)
6.-Information Leakage and Improper Error Handling
7.-Broken Authentication and Session Management
8.-Insecure Cryptographic Storage
9.-Insecure Communications
10.-Failure to Restrict URL Access

OWASP
The Open Web Application Security Project

2.Injection Flaws (SQL Injection)

- SQL is a powerful language which allows us to work with Relational Database Management Systems (RDBMS), offering a great reliability, robustness, and flexibility that in turn could become a big security hole.

- What Is SQL Injection?

The ability to inject SQL commands into the database engine through an existing application.

OWASP
The Open Web Application Security Project

2.Injection Flaws (SQL Injection) What is?

```
SqlConnection conn= new SqlConnection(
         “server=localhost;Database=Northwind” +
         “user id=sa;password=pass*word;");

string sqlString="SELECT * FROM Orders WHERE “
         + “CustomerID='" + idCliente + "'";

SqlCommand cmd = new SqlCommand(sqlString, conn);
conn.Open();
SqlDataReader reader = cmd.ExecuteReader();

// ...
```

2.Injection Flaws (SQL Injection)
How works?

OWASP
The Open Web Application Security Project

Datos validos: PRUEBA

```
SELECT * FROM
Orders WHERE
CustomerID='PRUEBA'
```

Los atacantes hacen una inyección SQL

PRUEBA';exec xp_cmdshell 'format C:

```
SELECT * FROM
Orders WHERE CustomerID=
'PRUEBA';exec xp_cmdshell 'format C:'
```

2.Injection Flaws (SQL Injection)

OWASP
The Open Web Application Security Project

- There are some characters, which have special meaning for RDBMS since they're part of the RDBMS SQL language itself. The following lists some of those special characters :

– ' or "	character String Indicators
– -- or #	single-line comment
– /*…*/	multiple-line comment
– +	addition, concatenate
– ||	(double pipe) concatenate
– %	wildcard attribute indicator

2.Injection Flaws (SQL Injection)

OWASP
The Open Web Application Security Project

Let's think of a SQL command which expects a string variable to be include in the WHERE clause and such command is hard coded in some 4G language.

"SELECT First_Name FROM workers
WHERE employee_id = '" + variable +"'";

What if our variable's value is: ' or ''=' ?

Once it gets parsed it ends as follows:

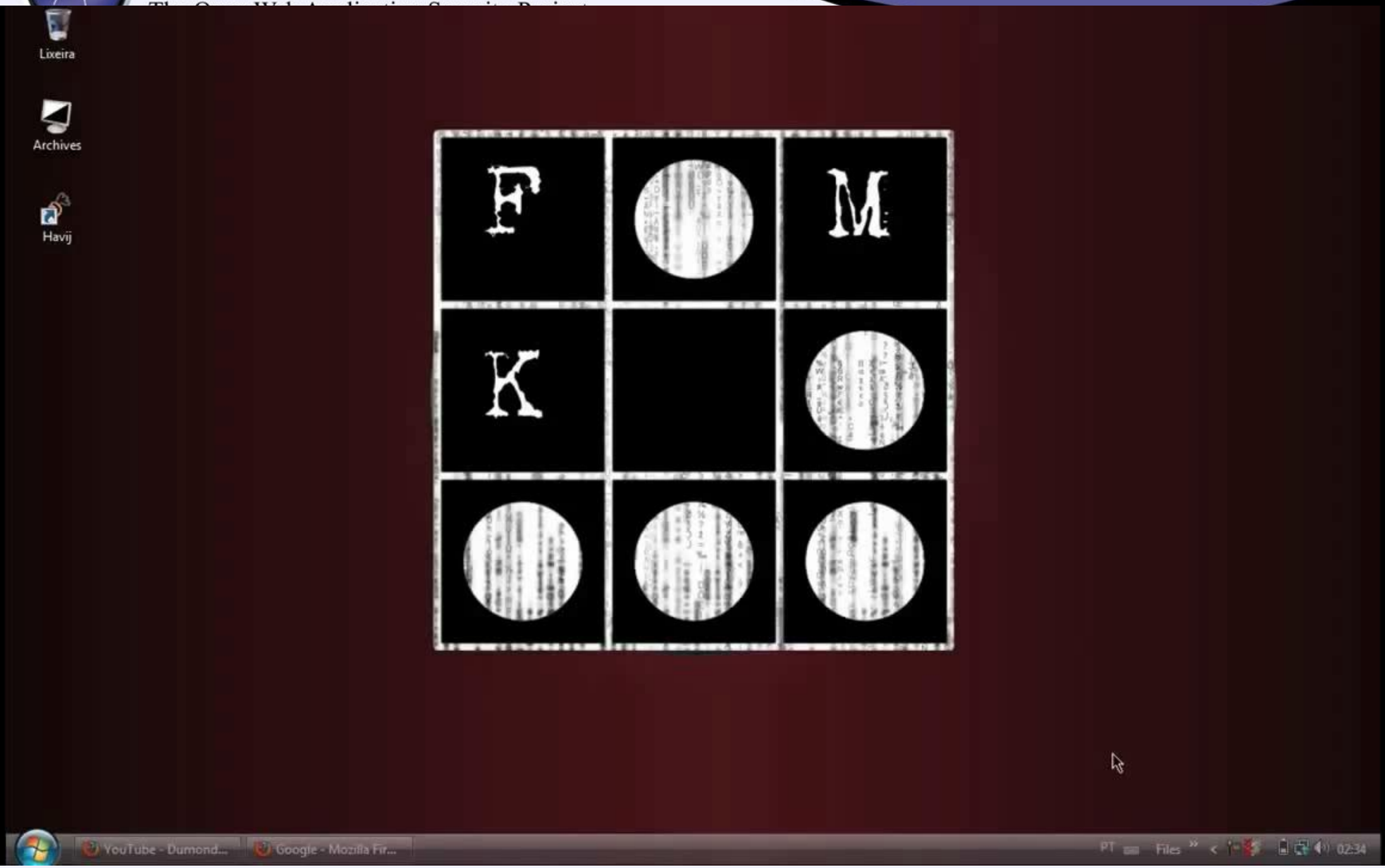
SELECT First_Name FROM workers
WHERE employee_id = '' or '' ='';

Now our condition is true for all cases. Therefore the actual SQL sentence is:

SELECT First_Name FROM workers;

OWASP
The Open Web Application Security Project

2. SQL Injection - Example


OWASP
Lixeira
Archives
Havij
F
M
K
YouTube - Dumond...
Google - Mozilla Fir...
PT
Files
02:34

OWASP
The Open Web Application Security Project

2. SQL Injection - Practice

2.Protection (SQL Injection)

OWASP
The Open Web Application Security Project

Input validation.
Positive Validation="accept known good"
Please make sure to validate the following:
-Allowed special character set.(A-Z, a-z, 0-9, @)
-code filters for ‘ / < >
-Whether Null or empty value is allowed
-whether the parameter is required or not.
-type and format of the field
-Minimum and Maximum field length
-Use of patterns (regular expressions) if numeric field, we recommend to please work with ranges.
Reject invalid input rather than attempting to sanitize potentially hostile data.
Do not forget that error messages might also include invalid data

- **Use strongly typed parameterized query APIs with placeholder substitution markers, even when calling stored procedures**
- **Enforce least privilege when connecting to databases and other backend systems**
- **Avoid detailed error messages that are useful to an attacker**
- **Do not use dynamic query interfaces (such as mysql_query() or similar)**
- **Watch out for canonicalization errors**

- **http://www.owasp.org/index.php/Top_10_2007-A2**

2.Injection Flaws (SQL Injection)

OWASP
The Open Web Application Security Project

Let's think of a SQL command which expects a string variable to be include in the WHERE clause and such command is hard coded in some 4G language.

"SELECT First_Name FROM workers
WHERE employee_id = '" + variable +"'";

What if our variable's value is: ' or ''=' ?

Once it gets parsed it ends as follows:

SELECT First_Name FROM workers
WHERE employee_id = '' or '' ='';

Now our condition is true for all cases. Therefore the actual SQL sentence is:

SELECT First_Name FROM workers;

Other References

OWASP
The Open Web Application Security Project

- CERT
  - www.cert.org
- Security focus
  - www.securityfocus.org
- NIST
  - www.nist.gov
- NSA
  - www.nsa.gov
- OWASP
  - www.owasp.org
- MS Security
  - microsoft.com/security
- Java
  - java.sun.com/security
- IT Security.com
  - www.itsecurity.com

OWASP
The Open Web Application Security Project

Questions?