SPB .NET Meetup #2

cezurity

Practice of AppSec .NET

Mikhail Shcherbakov
Product Manager at Cezurity

About me

- Product Manager at Cezurity
- One of the core developers of the source code analyzer PT Application Inspector
- Former Team Lead at Acronis, Luxoft, Boeing

Security Development

Habrahabr Example #1

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<body>
<!--
    <asp:Label runat="server">
        <%= Request["first"] + Request["second"] %>
    </asp:Label>
-->
    Preview:
    <img src="<%= ResolveUrl("~/Content/Images/" + Request["page"] + ".png")%>"/>

    <form id="form1" runat="server">
        <asp:Label ID="Label1" runat="server"></asp:Label><br/>
        <asp:RadioButtonList ID="RadioButtonList1" runat="server"/><br/>
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br/>
        <asp:Button ID="Button1" runat="server" Text="Save"/>
    </form>
</body>
</html>
```

Habrahabr Example #1

```
protected void Page_Load(object sender, EventArgs e)
{
    var text = TextBox1.Text + TextBox2.Text;
    if (text != String.Empty)
    {
        Label1.Text = "Input: " + text;
    }
    else
    {
        TextBox1.Text = Request["first"] + Request["second"];
    }

    int count;
    if (Int32.TryParse(Request["count"], out count))
    {
        for (int i = 0; i < count; i++)
        {
            var name = String.Format("base64_item{0}", i);
            var value = Request[name];
            if (value != null)
            {
                RadioButtonList1.Items.Add(
                    new ListItem(Encoding.UTF8.GetString(Convert.FromBase64String(value)), value));
            }
        }
    }
}
```

Improper Input / Output Handling

Implementation

Improper Input / Output Handling

- SQL Injection
- OS Commanding
- Cross-Site Scripting (XSS)
- XML Injection
- XPath Injection
- XQuery Injection
- LDAP Injection
- Mail Command Injection
- Null Injection
- Unrestricted File Upload
- Unrestricted File Download
- Path Traversal
- HTTP Response Splitting
- Content Spoofing
- Buffer Overflow

Cross-Site Scripting (XSS)

- Reflected
- Stored
- DOM-based

Reflected XSS

```
protected void Page_Load(object sender, EventArgs e)
{
    var text = TextBox1.Text + TextBox2.Text;
    if (text != String.Empty)
    {
        Label1.Text = "Input: " + text;
    }
    else
    {
        TextBox1.Text = Request["first"] + Request["second"];
    }

    int count;
    if (Int32.TryParse(Request["count"], out count))
    {
        for (int i = 0; i < count; i++)
        {
            var name = String.Format("base64_item{0}", i);
            var value = Request[name];
            if (value != null)
            {
                RadioButtonList1.Items.Add(
                    new ListItem(Encoding.UTF8.GetString(Convert.FromBase64String(value)), value));
            }
        }
    }
}
```

Reflected XSS

```
protected void Page_Load(object sender, EventArgs e)
{
    var text = TextBox1.Text + TextBox2.Text;
    if (text != String.Empty)
    {
        Label1.Text = "Input: " + text;
    }
    else
    {
}
```

Reflected XSS

POST http://localhost/Example
__VIEWSTATE=1WhGrdaz6wBJ67aoKvJd1oc1Nw...&
__VIEWSTATEGENERATOR=E5E1B94B&
__EVENTVALIDATION=uixzE1cGQE%2BFAGQTbTA...&
TextBox1=<&
TextBox2=img src=# onerror=alert('XSS')//&
Button1=Save

Reflected XSS

```
protected void Page_Load(object sender, EventArgs e)
{
    var text = TextBox1.Text + TextBox2.Text;
    if (text != String.Empty)
    {
        Label1.Text = "Input: " + text;
    }
    else
    {
        TextBox1.Text = Request["first"] + Request["second"];
    }

    int count;
    if (Int32.TryParse(Request["count"], out count))
    {
        for (int i = 0; i < count; i++)
        {
            var name = String.Format("base64_item{0}", i);
            var value = Request[name];
            if (value != null)
            {
                RadioButtonList1.Items.Add(
                    new ListItem(Encoding.UTF8.GetString(Convert.FromBase64String(value)), value));
            }
        }
    }
}
```

Reflected XSS

```
protected void Page_Load(object sender, EventArgs e)
{
    var text = TextBox1.Text + TextBox2.Text;
    if (text != String.Empty)
    {
        Label1.Text = "Input: " + text;
    }
    else
    {
        TextBox1.Text = Request["first"] + Request["second"];
    }
    {
        for (int i = 0; i < count; i++)
        {
            var name = String.Format("base64_item{0}", i);
            var value = Request[name];
            if (value != null)
            {
                RadioButtonList1.Items.Add(
                    new ListItem(Encoding.UTF8.GetString(Convert.FromBase64String(value)), value));
            }
        }
    }
}
```

No Vulnerability

Reflected XSS

```
protected void Page_Load(object sender, EventArgs e)
{
    var text = TextBox1.Text + TextBox2.Text;
    if (text != String.Empty)
    {
        Label1.Text = "Input: " + text;
    }
    else
    {
        TextBox1.Text = Request["first"] + Request["second"];
    }

    int count;
    if (Int32.TryParse(Request["count"], out count))
    {
        for (int i = 0; i < count; i++)
        {
            var name = String.Format("base64_item{0}", i);
            var value = Request[name];
            if (value != null)
            {
                RadioButtonList1.Items.Add(
                    new ListItem(Encoding.UTF8.GetString(Convert.FromBase64String(value)), value));
            }
        }
    }
}
```

Reflected XSS

Reflected XSS

GET
http://localhost/Example?**count**=1&**base64_item0**=PGltZyBzcm
M9IyBvbmVycm9yPWFsZXJ0KCdYU1MnKS8v

```
protected void Page_Load(object sender, EventArgs e)
{
    int count;
    if (Int32.TryParse(Request["count"], out count))
    {
        for (int i = 0; i < count; i++)
        {
            var name = String.Format("base64_item{0}", i);
            var value = Request[name];
            if (value != null)
            {
                RadioButtonList1.Items.Add(
                    new ListItem(Encoding.UTF8.GetString(Convert.FromBase64String(value)), value));
            }
        }
    }
}
```

Reflected XSS

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<body>
<!--
    <asp:Label runat="server">
        <%= Request["first"] + Request["second"] %>
    </asp:Label>
-->
    Preview:
    <img src="<%= ResolveUrl("~/Content/Images/" + Request["page"] + ".png")%>"/>

    <form id="form1" runat="server">
        <asp:Label ID="Label1" runat="server"></asp:Label><br/>
        <asp:RadioButtonList ID="RadioButtonList1" runat="server"/><br/>
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br/>
        <asp:Button ID="Button1" runat="server" Text="Save"/>
    </form>
</body>
</html>
```

Reflected XSS

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<body>
<!--
    <asp:Label runat="server">
        <%= Request["first"] + Request["second"] %>
    </asp:Label>
-->
    Preview:
    <img src="<%= ResolveUrl("~/Content/Images/" + Request["page"] + ".png")%>"/>

    <form id="form1" runat="server">
        <asp:Label ID="Label1" runat="server"></asp:Label><br/>
        <asp:RadioButtonList ID="RadioButtonList1" runat="server"/><br/>
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br/>
        <asp:Button ID="Button1" runat="server" Text="Save"/>
    </form>
</body>
</html>
```

Reflected XSS

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<body>
<!--
    <asp:Label runat="server">
        <%= Request["first"] + Request["second"] %>
    </asp:Label>
-->
        <asp:Button ID="Button1" runat="server" Text="Save"/>
    </form>
</body>
</html>
```

Reflected XSS

GET http://localhost/Example?**first**=--%3E%3C&**second**=img%20src=%27n%27%20onerror=alert%28%27XSS%27%29//

Reflected XSS

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<body>
<!--
    <asp:Label runat="server">
        <%= Request["first"] + Request["second"] %>
    </asp:Label>
-->
    Preview:
    <img src="<%= ResolveUrl("~/Content/Images/" + Request["page"] + ".png")%>"/>

    <form id="form1" runat="server">
        <asp:Label ID="Label1" runat="server"></asp:Label><br/>
        <asp:RadioButtonList ID="RadioButtonList1" runat="server"/><br/>
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br/>
        <asp:Button ID="Button1" runat="server" Text="Save"/>
    </form>
</body>
</html>
```

Reflected XSS

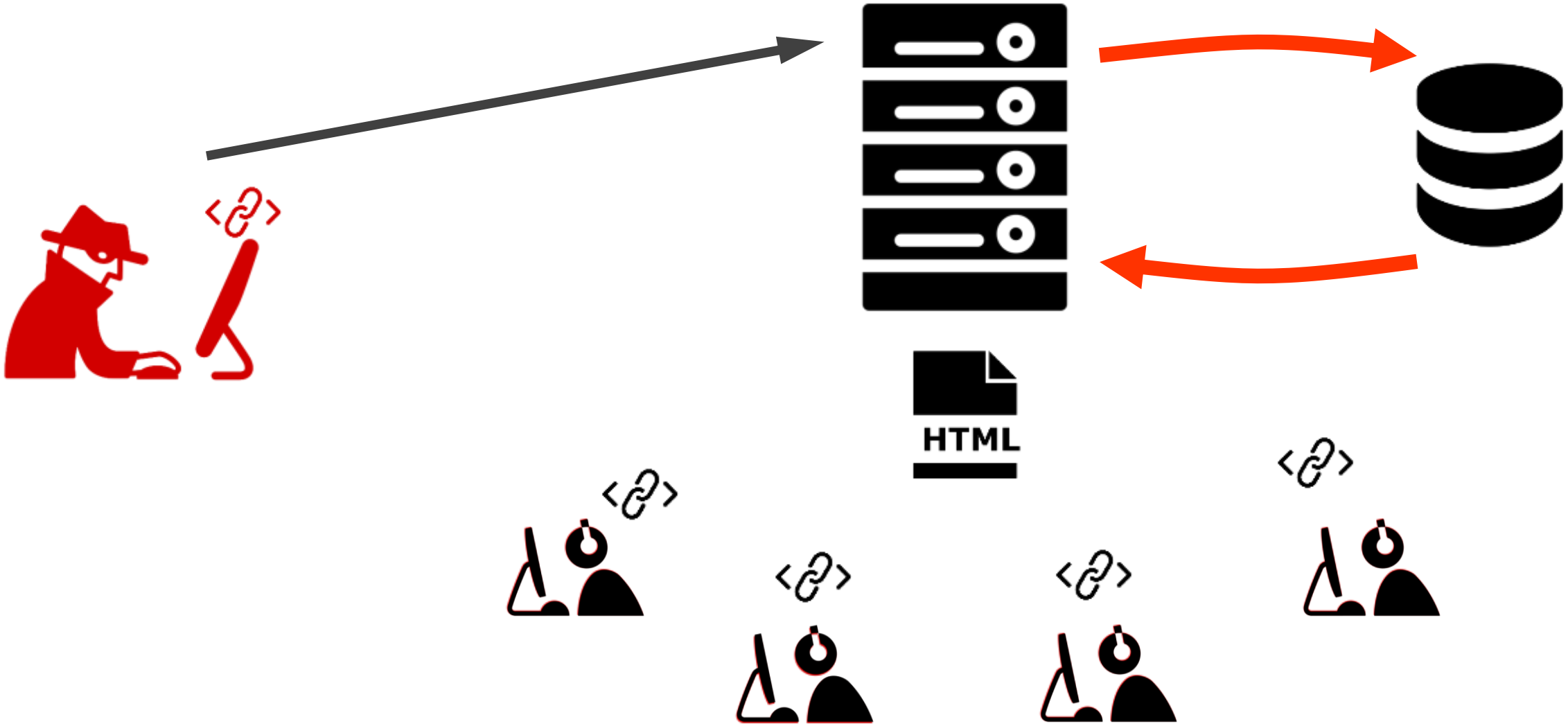
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<body>
<!--
    <asp:Label runat="server">
        <%= Request["first"] + Request["second"] %>
    </asp:Label>
-->
    Preview:
    <img src="<%= ResolveUrl("~/Content/Images/" + Request["page"] + ".png")%>"/>
```

Reflected XSS

GET
http://localhost/Example?**page**=%22%20onerror=alert%28%27XSS%27%29;//

IIS Request Validation

Stored XSS



Stored XSS

Show me the code!

DOM-based XSS

Show me the code!

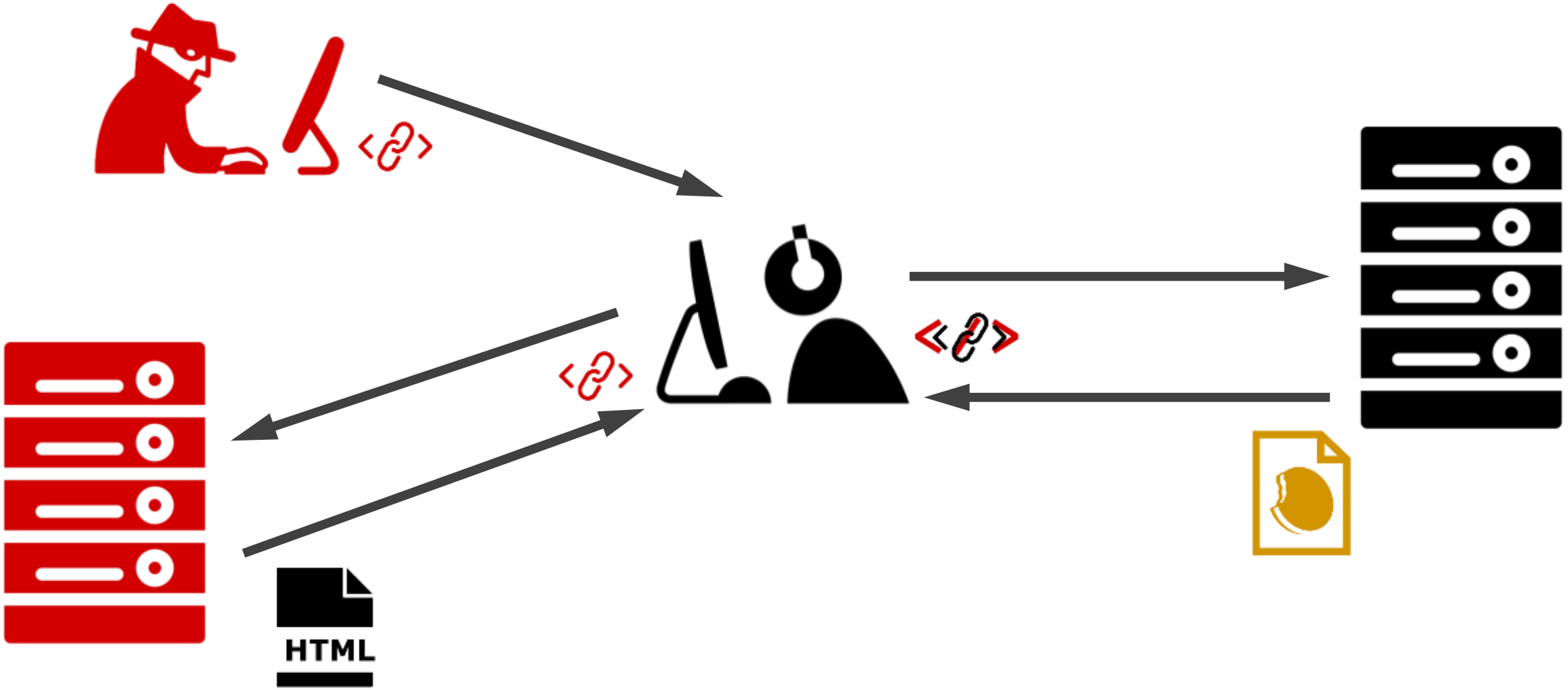
Insufficient Control Flow Management

Design / Implementation

Insufficient Control Flow Management

- Cross-Site Request Forgery (CSRF)
- Mass Assignment
- Business Logic Errors
- Abuse of Functionality

Cross-Site Request Forgery (CSRF)



CSRF

Show me the code!

CSRF

```
<script type="text/javascript">
    function sendRequest(method, path, params) {
        var form = document.createElement("form");
        form.setAttribute("method", method);
        form.setAttribute("action", path);
        form.setAttribute("target", "hidden-form");
        form.__submit = form.submit;

        for (var key in params)
        {
            var field = document.createElement("input");
            field.setAttribute("type", "hidden");
            field.setAttribute("name", key);
            field.setAttribute("value", params[key]);

            form.appendChild(field);
        }

        document.body.appendChild(form);
        var frame = document.createElement("iframe");
        frame.setAttribute("style", "display:none");
        frame.setAttribute("name", "hidden-form");
        document.body.appendChild(frame);

        form.__submit();
    }
```

CSRF Defense

- ASP.NET MVC
  - <%= Html.AntiForgeryToken() %>
  - <input name="__RequestVerificationToken" type="hidden“ …
- ASP.NET Web Forms
  - __VIEWSTATE
  - __EVENTVALIDATION

CSRF Defense

- Same Origin Policy
  - An origin is defined by the scheme, host and port
  - Documents retrieved from distinct origins are isolated

Habrahabr Example #2

```
var parameters = new List<object>();

var email = Request["email"];
if (email != null)
{
    where += String.Format(" email LIKE '{0}%'", email);
}

var field = Request["field"];
var min = Request["min"];
var max = Request["max"];
if (field != null && min != null && max != null)
{
    if (!String.IsNullOrEmpty(where))
    {
        where += " AND";
    }

    where += String.Format(" {0} >= @min AND {0} <= @max", EncodeSqlField(field));
    parameters.Add(new SqlParameter("@min", min));
    parameters.Add(new SqlParameter("@max", max));
}

var query = "SELECT * FROM CustomerLogin";
if (!String.IsNullOrEmpty(where))
{
    query += " WHERE";
    query += where;
}

var output = db.Database
    .SqlQuery<CustomerLogin>(query, parameters.ToArray())
    .ToArray();

lblOutput.Text = output.Length == 0
    ? "Not found"
    : String.Join("<br/>", output.Select(customer => customer.email + " - " + customer.customerNumber));
```

Habrahabr Example #2

```
var parameters = new List<object>();

var email = Request["email"];
if (email != null)
{
    where += String.Format(" email LIKE '{0}%'", email);
}

var field = Request["field"];
var min = Request["min"];
var max = Request["max"];
if (field != null && min != null && max != null)
{
    if (!String.IsNullOrEmpty(where))
    {
        where += " AND";
    }

    where += String.Format(" {0} >= @min AND {0} <= @max", EncodeSqlField(field));
    parameters.Add(new SqlParameter("@min", min));
    parameters.Add(new SqlParameter("@max", max));
}

var query = "SELECT * FROM CustomerLogin";
```

Habrahabr Example #2

```
var parameters = new List<object>();

var email = Request["email"];
if (email != null)
{
    where += String.Format(" email LIKE '{0}%'", email);
}
```

SQL Injection

GET http://localhost/Example?**email**='--

```
    if (!String.IsNullOrEmpty(where))
    {
        where += " AND";
    }

    where += String.Format(" {0} >= @min AND {0} <= @max", EncodeSqlField(field));
    parameters.Add(new SqlParameter("@min", min));
    parameters.Add(new SqlParameter("@max", max));
}

var query = "SELECT * FROM CustomerLogin";
```

Habrahabr Example #2

```
var min = Request["min"];
var max = Request["max"];
if (field != null && min != null && max != null)
{
    if (!String.IsNullOrEmpty(where))
    {
        where += " AND";
    }

    where += String.Format(" {0} >= @min AND {0} <= @max", EncodeSqlField(field));
    parameters.Add(new SqlParameter("@min", min));
    parameters.Add(new SqlParameter("@max", max));
}

var query = "SELECT * FROM CustomerLogin";
if (!String.IsNullOrEmpty(where))
{
    query += " WHERE";
    query += where;
}

var output = db.Database
    .SqlQuery<CustomerLogin>(query, parameters.ToArray())
```

Habrahabr Example #2

```
var min = Request["min"];
var max = Request["max"];
if (field != null && min != null && max != null)
{
    if (!String.IsNullOrEmpty(where))
    {
        where += " AND";
    }

    where += String.Format(" {0} >= @min AND {0} <= @max", EncodeSqlField(field));
    parameters.Add(new SqlPara
    parameters.Add(new SqlPara
}

var query = "SELECT * FROM Cus
if (!String.IsNullOrEmpty(wher
{
    query += " WHERE";
    query += where;
}

var output = db.Database
    .SqlQuery<CustomerLogin>(q
```

```
private string EncodeSqlField(string field)
{
    return field.Replace("'", String.Empty)
        .Replace(" ", String.Empty)
        .Replace("\\", String.Empty)
        .Replace(",", String.Empty)
        .Replace("(", String.Empty)
        .Replace(")", String.Empty);
}
```

Habrahabr Example #2

```
var min = Request["min"];
var max = Request["max"];
if (field != null && min != null && max != null)
{
    if (!String.IsNullOrEmpty(where))
    {
        where += " AND";
    }

    where += String.Format(" {0} >= @min AND {0} <= @max", EncodeSqlField(field));
    parameters.Add(new SqlParameter("@min", min));
    parameters.Add(new SqlParameter("@max", max));
}
```

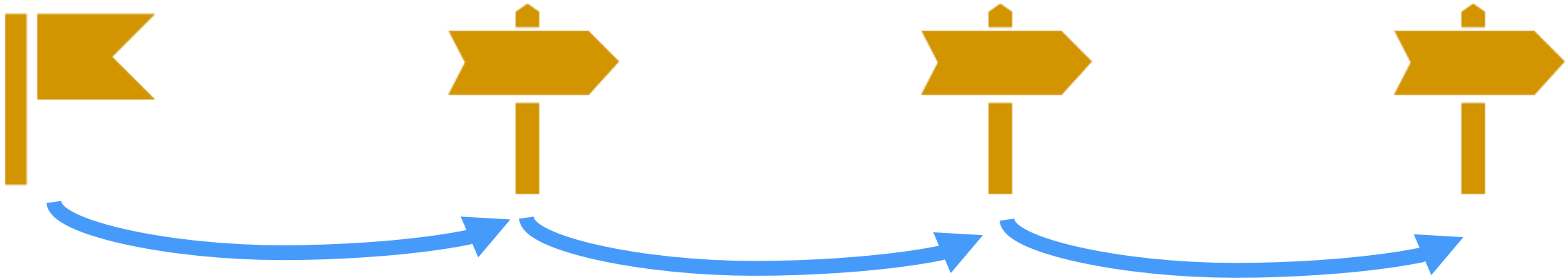
Business Logic Error

GET http://localhost/Example?**field**=password&**min**=a&**max**=b
GET http://localhost/Example?**field**=password&**min**=aD&**max**=aE

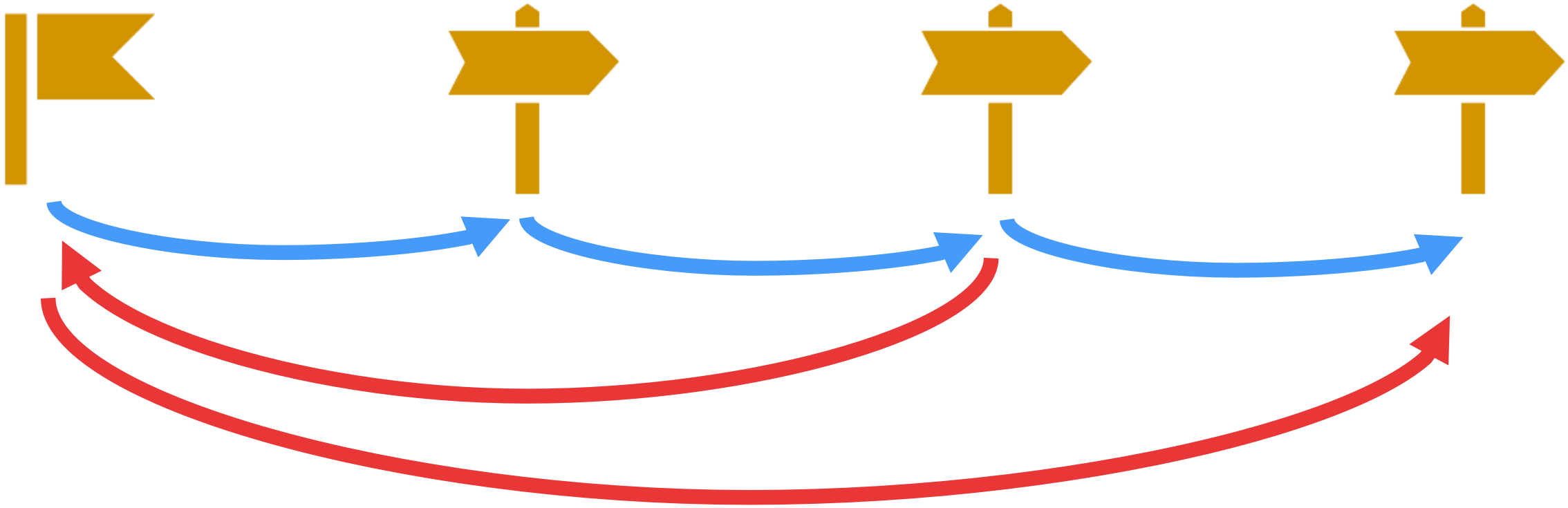
```
var output = db.Database
    .SqlQuery<CustomerLogin>(query, parameters.ToArray())
```

Business Logic Error

Business Logic Error



Business Logic Error



Business Logic Error

Show me the code!

Broken Authentication and Session Management

Design / Implementation / Deployment

Session Fixation

Show me the code!

Session Fixation Defense

- Set invalid ASP .NET session cookie when the user log in, so the user receives a new cookie

Session Fixation Defense

- ~~Set invalid ASP .NET session cookie when the user log in, so the user receives a new cookie~~
  - Issue: the order to send cookies from the browser
- Store the username in the session
- Generate Session ID on the logged user
  - NWebsec.SessionSecurity

Summary

- OWASP Top Ten Project (2010/2013) http://bit.ly/1OffewO
- Vladimir Kochetkov Blog and Workshop http://bit.ly/1DecXWI
- Troy Hunt Blog www.troyhunt.com
- OWASP Developer Guide http://bit.ly/1JcQLoh
- CWE/SANS Top 25 Most Dangerous Software Errors (2011) http://bit.ly/1bjDTOH
- OWASP Classification http://bit.ly/1GlKmGz http://bit.ly/1DE3852
- WASC Classification http://bit.ly/1d3EXYd

Thank you for your attention!

Mikhail Shcherbakov

Product Manager at Cezurity

ms@cezurity.com

linkedin.com/in/mikhailshcherbakov

github.com/yuske

@yu5k3