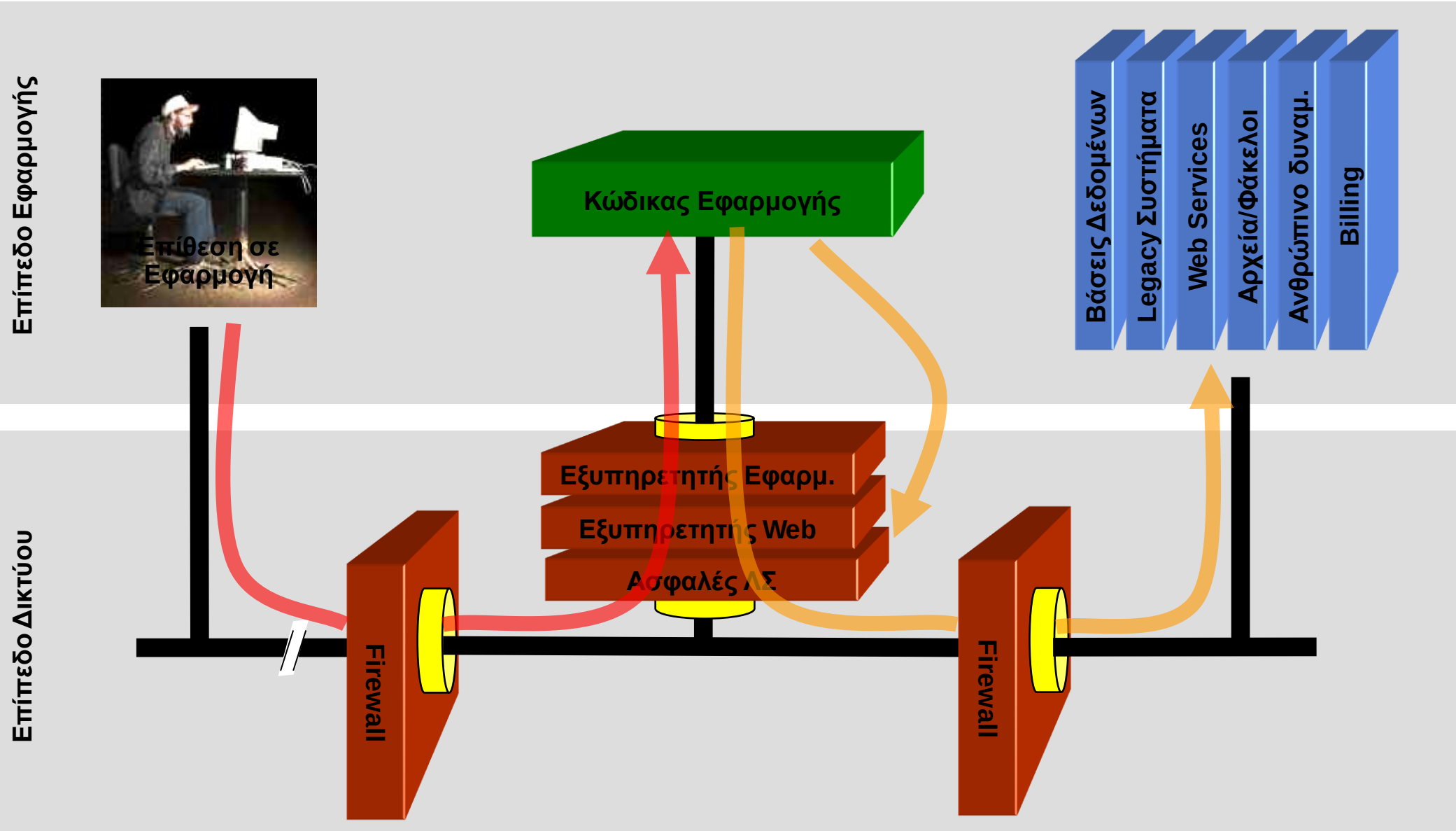
ΕΘΝΙΚΟΝ & ΚΑΠΟΔΙΣΤΡΙΑΚΟΝ ΠΑΝΕΠΙΣΤΗΜΙΟΝ ΑΘΗΝΩΝ
NATIONAL & KAPODISTRIAN UNIVERSITY OF ATHENS

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
Πρόγραμμα Μεταπτυχιακών Σπουδών (Π.Μ.Σ.)

Ευπάθειες Διαδικτυακών Εφαρμογών

Δρ. Κωνσταντίνος Παπαπαναγιώτου
conpap@di.uoa.gr

Ο κώδικας αποτελεί τμήμα της περιμέτρου



Απειλές

- Απειλές κατά την ανάπτυξη
  - Προγραμματιστής εισάγει κακόβουλο κώδικα επίτηδες
  - Προγραμματιστικά λάθη
- Απειλές κατά τη λειτουργία
  - Εκμετάλλευση γνωστών ευπαθειών που δεν έχουν διορθωθεί (ή δεν έχουν εγκατασταθεί οι αντίστοιχες ενημερώσεις)
  - «Μη αναμενόμενη» λειτουργία του προγράμματος


"6.5 Develop applications based on secure coding guidelines. Prevent common coding vulnerabilities in software development processes…"
PCI DSS
"…industry best practices for vulnerability management are for example, the *OWASP Guide*, *SANS CWE Top 25*, *CERT Secure Coding*, etc…"

OWASP

The Open Web Application Security Project

OWASP Top 10 - 2010

The Ten Most Critical Web Application Security Risks

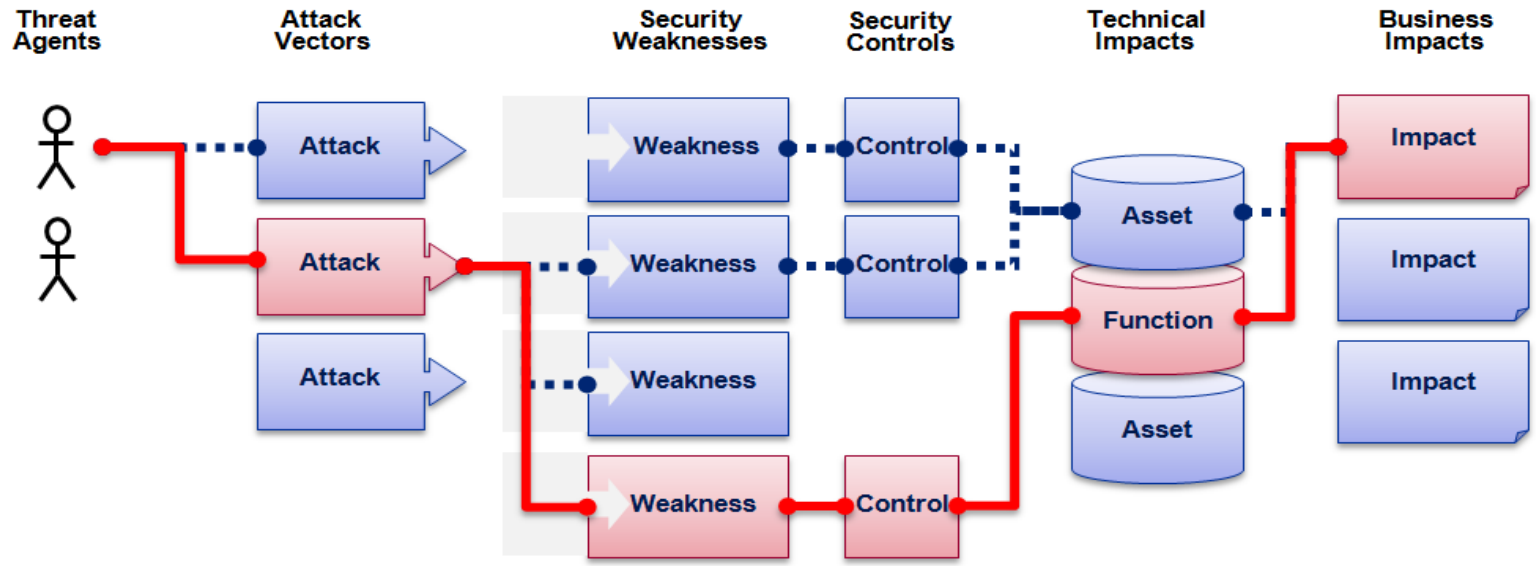
release

Σύγκριση Top 10 2007 με 2010

OWASP Top 10 – 2007 (Προηγούμενο)		OWASP Top 10 – 2010 (Νέο)
A2 – Injection Flaws	↑	A1 – Injection
A1 – Cross Site Scripting (XSS)	↓	A2 – Cross Site Scripting (XSS)
A7 – Broken Authentication and Session Management	↑	A3 – Broken Authentication and Session Management
A4 – Insecure Direct Object Reference	=	A4 – Insecure Direct Object References
A5 – Cross Site Request Forgery (CSRF)	=	A5 – Cross Site Request Forgery (CSRF)
<ήταν T10 2004 A10 – Insecure Configuration Management>	+	A6 – Security Misconfiguration (NEW)
A8 – Insecure Cryptographic Storage	↑	A7 – Insecure Cryptographic Storage
A10 – Failure to Restrict URL Access	↑	A8 – Failure to Restrict URL Access
A9 – Insecure Communications	=	A9 – Insufficient Transport Layer Protection
<δε συμπεριλαμβάνεται στο T10 2007>	+	A10 – Unvalidated Redirects and Forwards (NEW)
A3 – Malicious File Execution	-	<δε συμπεριλαμβάνεται στο T10 2010>
A6 – Information Leakage and Improper Error Handling	-	<δε συμπεριλαμβάνεται στο T10 2010>


Κίνδυνοι

OWASP Top 10 Αποτίμηση Κινδύνων

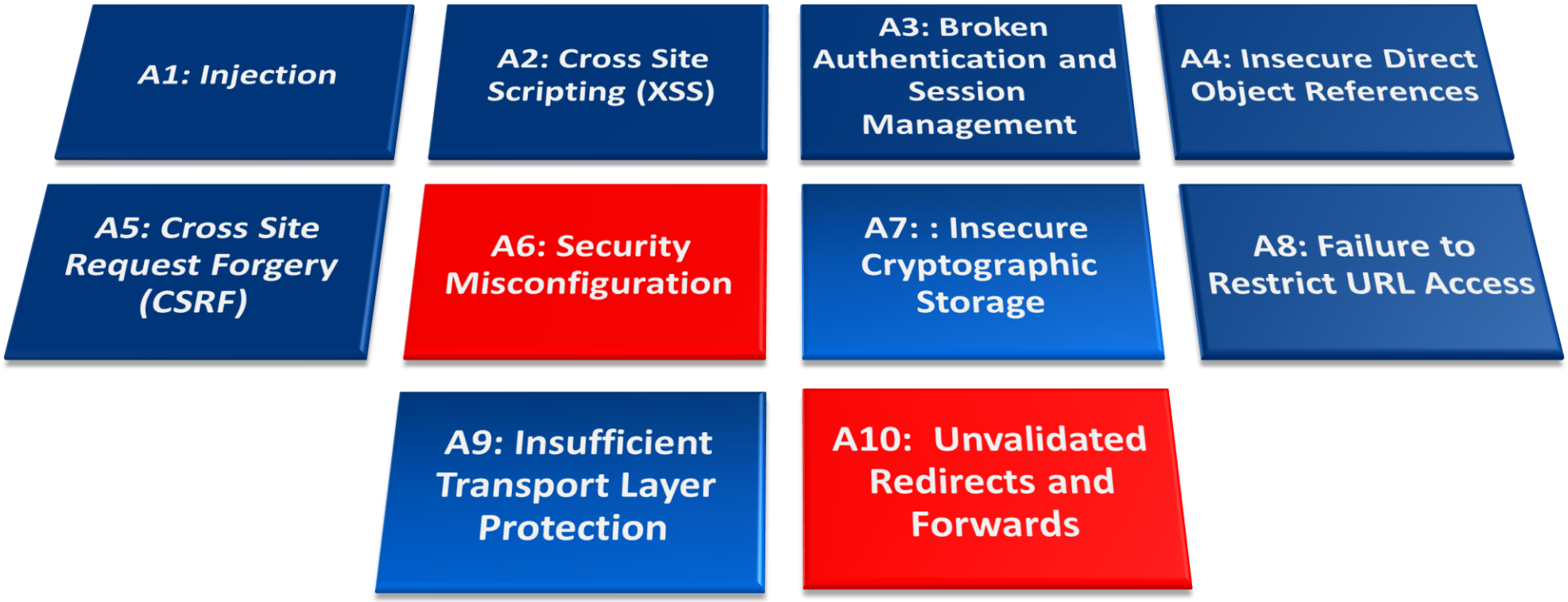


Φορέας Απειλής	Διάνυσμα Επίθεσης	Διάδοση ευπάθειας	Ευκολία Εντοπισμού	Τεχνικές Επιπτώσεις	Επιχειρησιακές Επιπτώσεις
	1 Εύκολη	Ευρέως διαδεδομένη	Εύκολος	Σοβαρές	
?	2 Μέση	Συνηθισμένη	Μέσος	Μέσες	?
	3 Δύσκολη	Ασυνήθιστη	Δύσκολος	Μικρές	
	1	2	2	1	
		1.66	*	1	

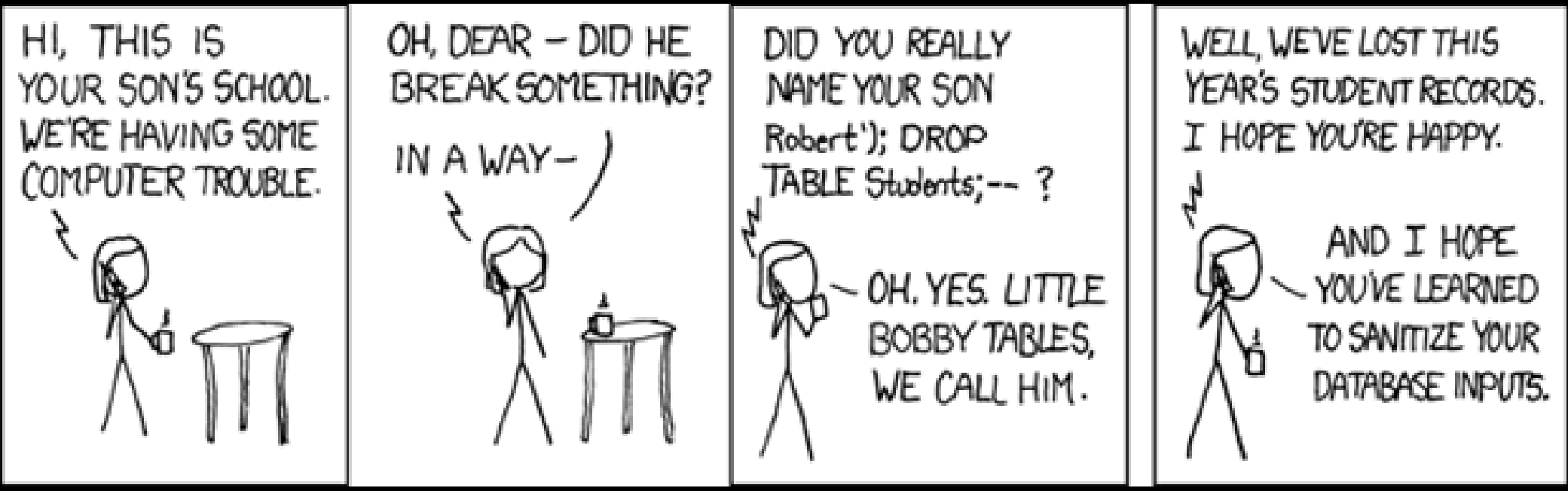
Παράδειγμα: Ένεση

1.66 σταθμισμένο ρίσκο

OWASP Top 10 2010



http://www.owasp.org/index.php/Top_10


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR - DID HE BREAK SOMETHING?
IN A WAY-
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

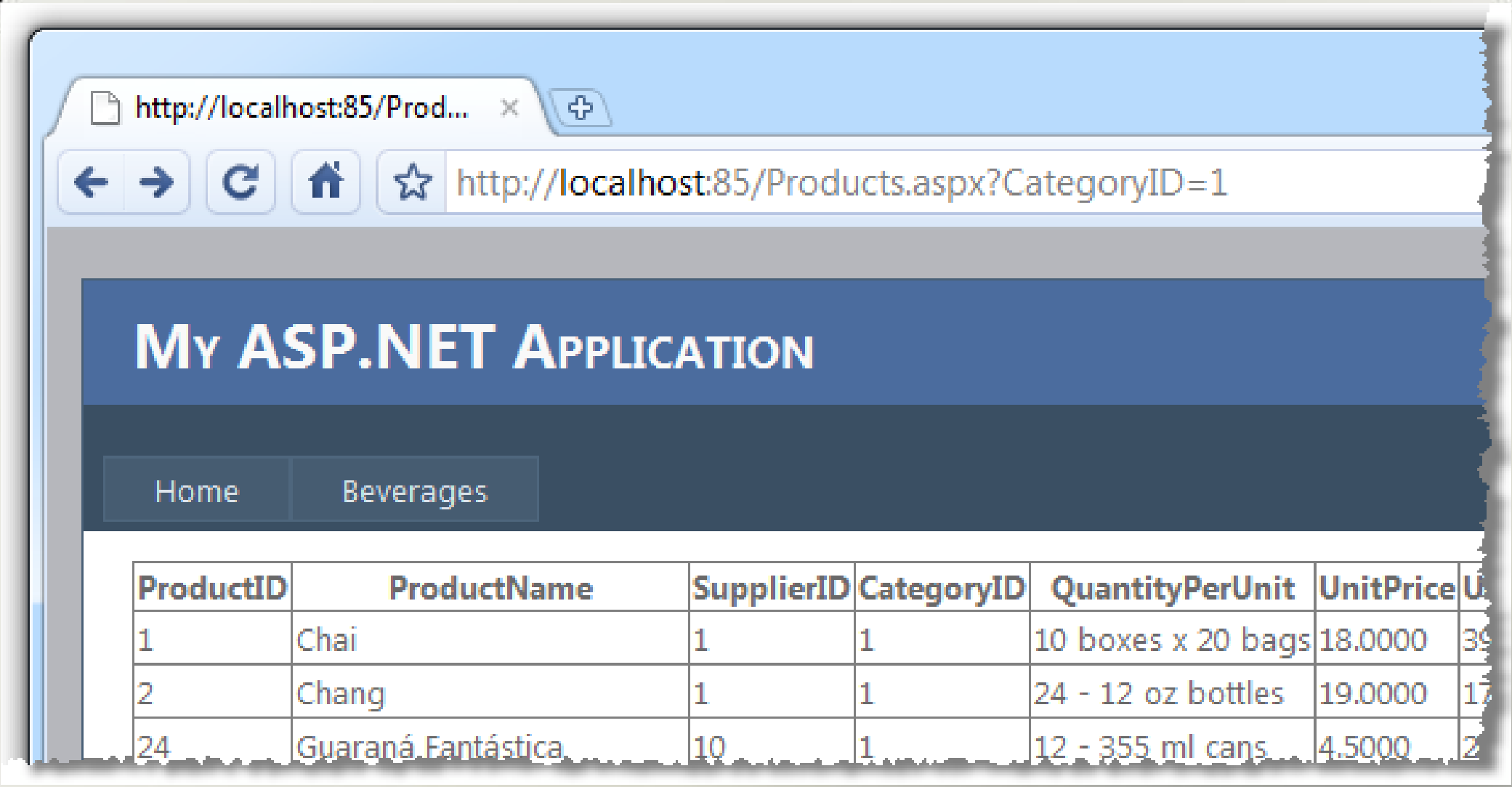
Products

Column Name	Data Type	Allow Nulls
ProductID	int	☐
ProductName	nvarchar(140)	☐
SupplierID	int	☑
CategoryID	int	☑
QuantityPerUnit	nvarchar(20)	☑
UnitPrice	money	☑
UnitsInStock	smallint	☑
UnitsOnOrder	smallint	☑
ReorderLevel	smallint	☑
Discontinued	bit	☐
		☐

Customers

Column Name	Data Type	Allow Nulls
CustomerID	nchar(5)	☐
CompanyName	nvarchar(40)	☐
ContactName	nvarchar(30)	☑
ContactTitle	nvarchar(30)	☑
Address	nvarchar(60)	☑
City	nvarchar(15)	☑
Region	nvarchar(15)	☑
PostalCode	nvarchar(10)	☑
Country	nvarchar(15)	☑
Phone	nvarchar(24)	☑
Fax	nvarchar(24)	☑
		☐

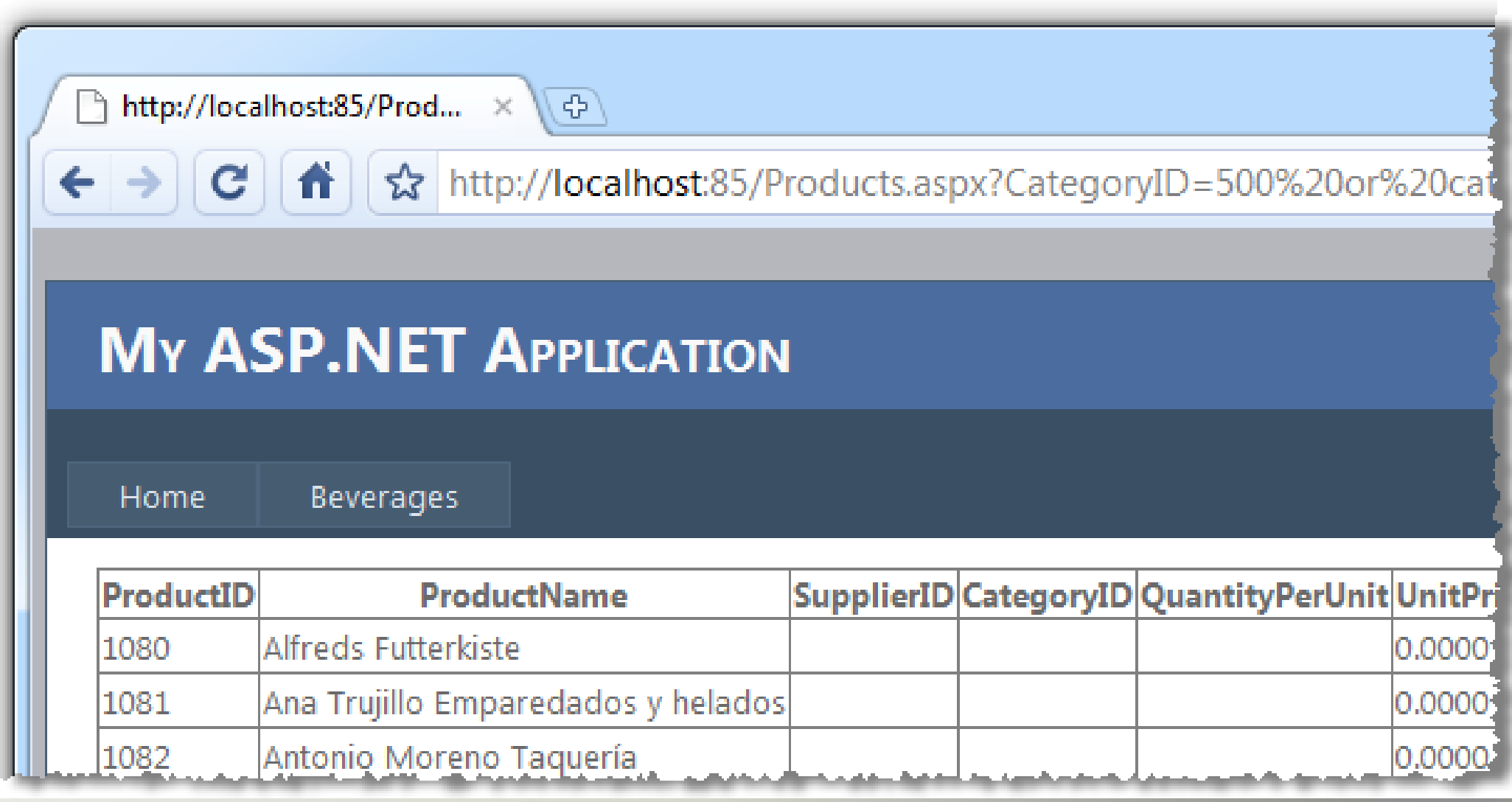
```
var catID = Request.QueryString["CategoryID"];
var sqlString = "SELECT * FROM Products WHERE CategoryID = " +
  catID;
var connString = WebConfigurationManager.ConnectionStrings

["NorthwindConnectionString"].ConnectionString;

using (var conn = new SqlConnection(connString))
{
   var command = new SqlCommand(sqlString, conn);
   command.Connection.Open();
   grdProducts.DataSource = command.ExecuteReader();
   grdProducts.DataBind();
}
```



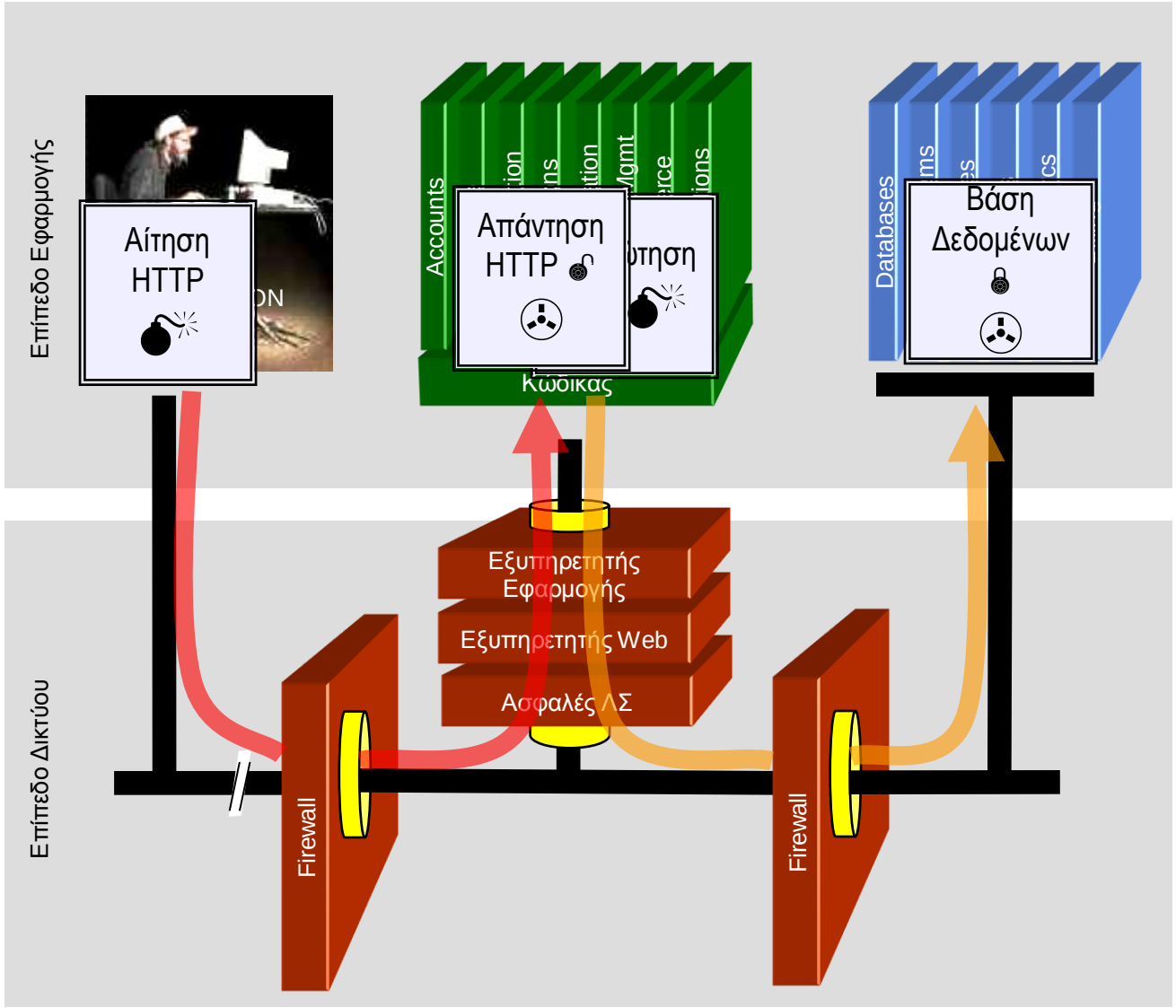

```
Products.aspx?CategoryID=1    or 1=1

SELECT * FROM Products WHERE CategoryID = 1     or 1=1
```

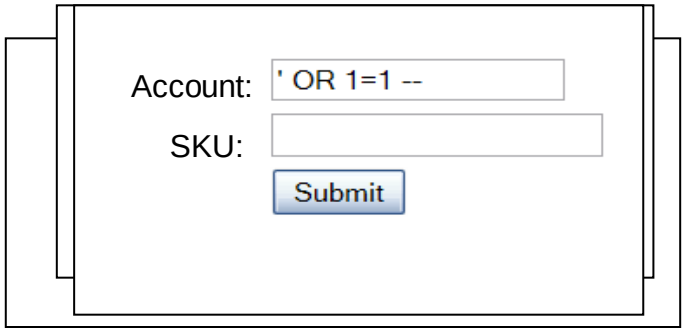
```
Products.aspx?CategoryID=1 or name=''
SELECT * FROM Products WHERE CategoryID = 1 or
   name=''


Products.aspx?CategoryID=1 or productname=''
SELECT * FROM Products WHERE CategoryID = 1 or
   productname=''


Products.aspx?CategoryID=1 or 1=(select count(*)
   from products)
SELECT * FROM Products WHERE CategoryID = 1 or
   1=(select count(*) from products)
```

```
Products.aspx?CategoryID=1;update products set
   productname = productname
SELECT * FROM Products WHERE CategoryID = 1;update
   products set productname = productname

Products.aspx?CategoryID=1;insert into
   products(productname) select companyname from
   customers
SELECT * FROM Products WHERE CategoryID = 1;insert
   into products(productname) select companyname
   from customers

Products.aspx?CategoryID= 500 or categoryid is
   null
SELECT * FROM Products WHERE CategoryID = 500 or
   categoryid is null
```


http://localhost:85/Prod...
http://localhost:85/Products.aspx?CategoryID=500%20or%20cat
My ASP.NET Application
Home
Beverages
ProductID
ProductName
SupplierID
CategoryID
QuantityPerUnit
UnitPr
1080
Alfreds Futterkiste
0.0000
1081
Ana Trujillo Emparedados y helados
0.0000
1082
Antonio Moreno Taquería
0.0000

A1. Ένεση





1. Η εφαρμογή εμφανίζει μία φόρμα.
2. Ο επιτιθέμενος αποστέλλει δεδομένα μέσω της φόρμας.
3. Η εφαρμογή προωθεί τα δεδομένα στη βάση μέσα από μία επερώτηση SQL.
4. Η βάση εκτελεί την επερώτηση και αποστέλει τα [κρυπτογραφημένα] δεδομένα πίσω στην εφαρμογή.
5. Η εφαρμογή [αποκρυπτογραφεί και] παρουσιάζει τα δεδομένα στο χρήστη.

A1 – Ένεση

Ένεση σημαίνει…

- Μία εφαρμογή «ξεγελιέται» στον να συμπεριλάβει κακόβουλες εντολές στα δεδομένα που στέλνονται στο διερμηνέα (interpreter).

Διερμηνείς…

- Συμβολοσειρές εισόδου ερμηνεύονται σαν εντολές
- SQL, OS Shell, LDAP, XPath, Hibernate, κλπ…

Η ένεση SQL εξακολουθεί να είναι πολύ διαδεδομένη

- Πολλές εφαρμογές εξακολουθούν να είναι ευάλωτες (γιατί;)
- Αρκετά εύκολη η προστασία

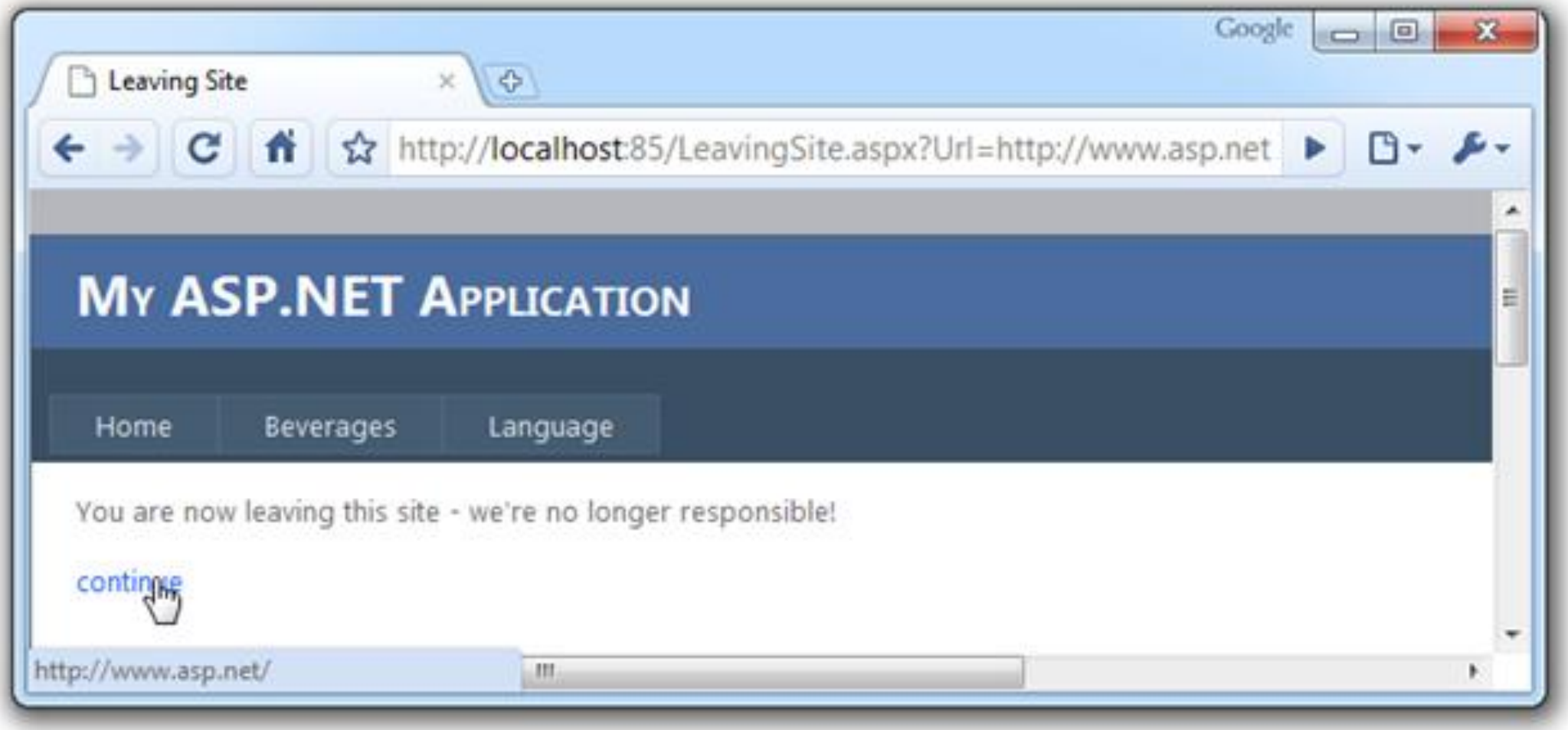
Συνήθεις Επιπτώσεις

- Συνήθως σημαντικές. Όλη η βάση μπορεί να διαβαστεί ή τροποποιηθεί
- Μπορεί να οδηγήσει στην αποκάλυψη όλης της βάσης, πρόσβαση με προνομιούχους λογαριασμούς, ακόμα και παραβίαση σε επίπεδο ΛΣ.

A1 – Αποφυγή Ευπαθειών τύπου Ένεσης

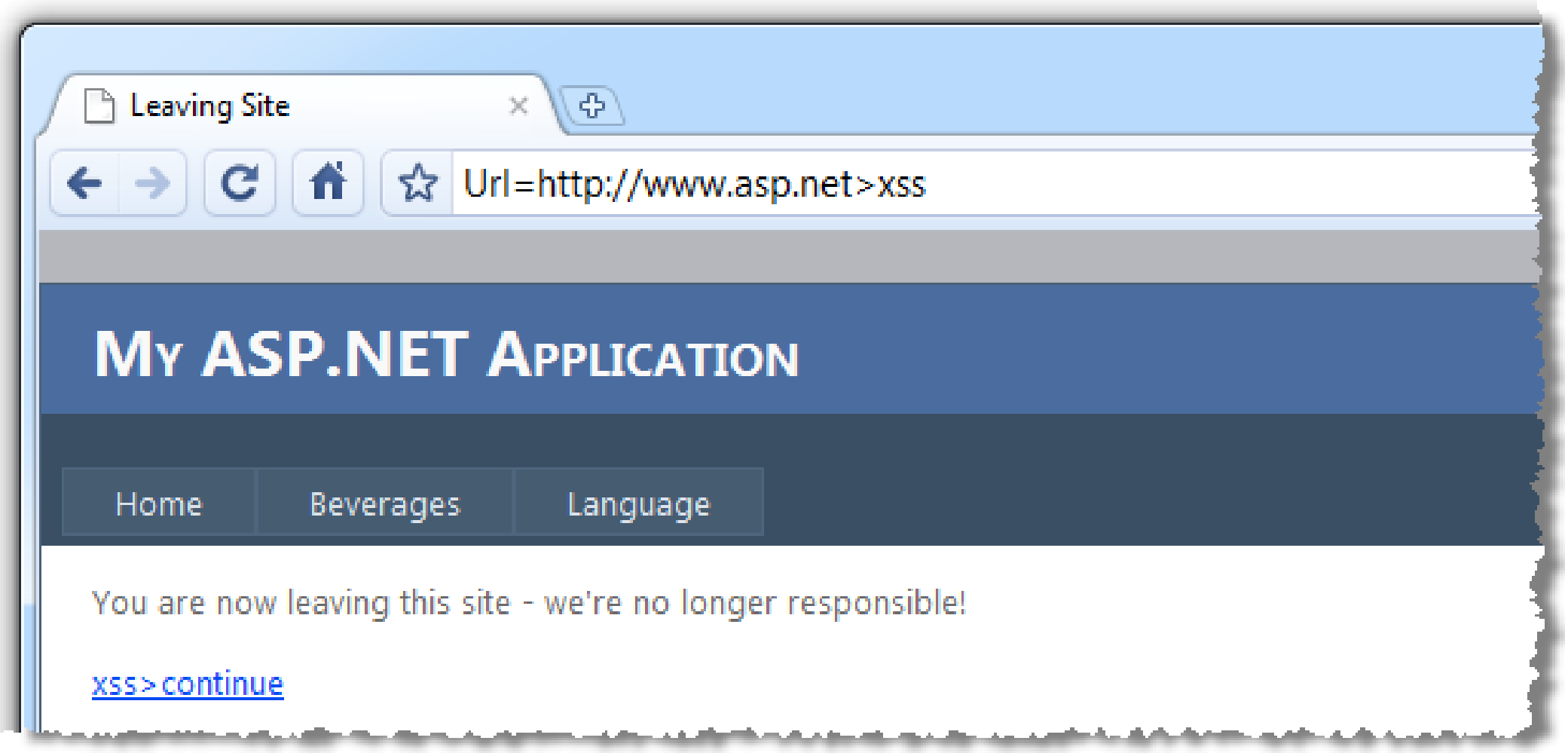
- Συστάσεις
  1. Παράκαμψη του διερμηνέα
  2. Χρήση διεπαφών που υποστηρίζουν bind variables (π.χ., prepared statements, ή stored procedures),
     - Τα bind variables δίνουν τη δυνατότητα στο διερμηνέα να διαχωρίσει τον κώδικα από τα δεδομένα
  3. Κωδικοποίηση δεδομένων εισόδου χρήστη πριν διαβιβαστούν στο διερμηνέα
  - Επαλήθευση δεδομένων εισόδου για όλα τα δεδομένα που εισάγει ο χρήστης βάσει 'white list'
  - Ελαχιστοποίηση των δικαιωμάτων στη βάση δεδομένων ώστε να μειωθούν οι επιπτώσεις από πιθανές ευπάθειες

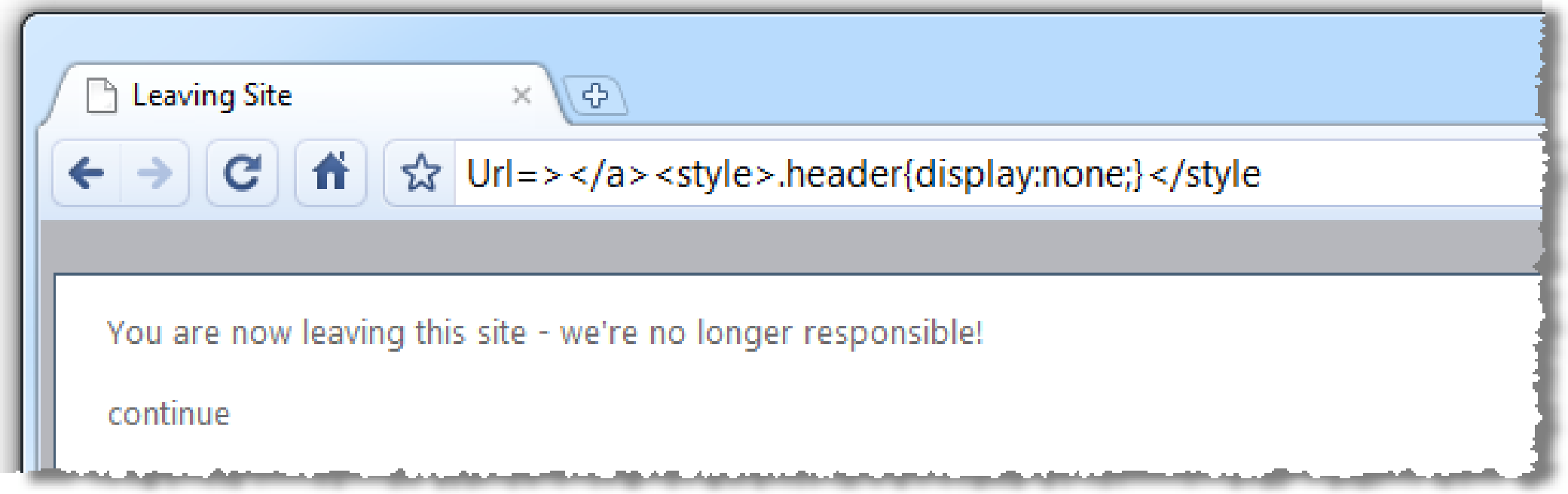
- Αναφορές
  - Για περισσότερες λεπτομέρειες μπορείτε να διαβάσετε το: http://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet

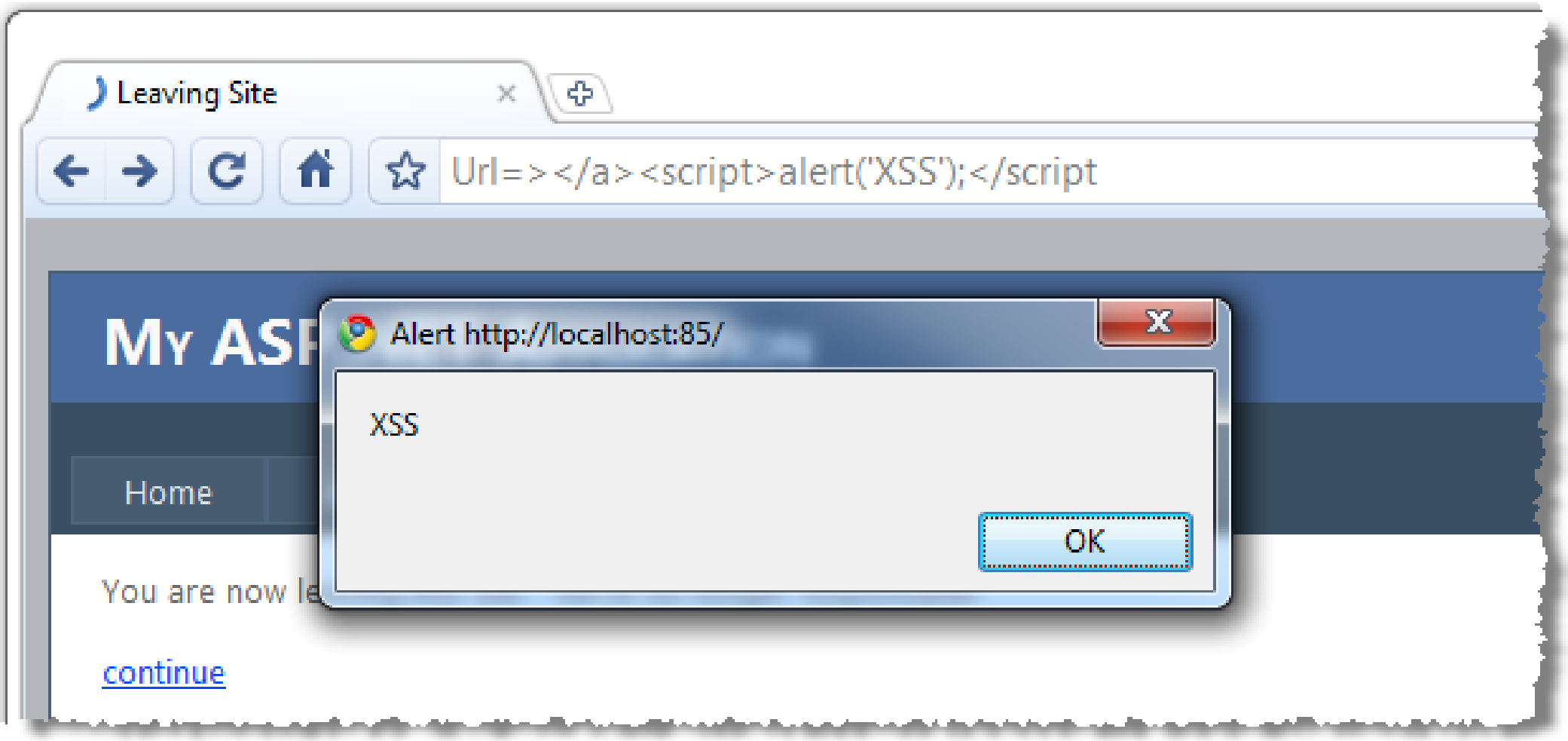



```
<p>You are now leaving this site - we're no longer responsible!</p>
<p><asp:Literal runat="server" ID="litLeavingTag" /></p>

var newUrl = Request.QueryString["Url"];
var tagString = "<a href=" + newUrl + ">continue</a>";
litLeavingTag.Text = tagString;
```

```
<p><a href=http://www.asp.net>continue</a></p>
```

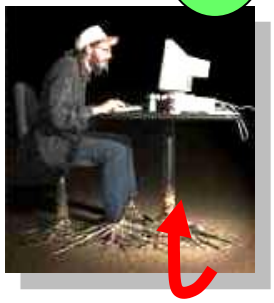

Leaving Site
Url=http://www.asp.net>xss
My ASP.NET Application
Home
Beverages
Language
You are now leaving this site - we're no longer responsible!
xss>continue


Leaving Site
Url=><style>.header{display:none;}</style
You are now leaving this site - we're no longer responsible!
continue

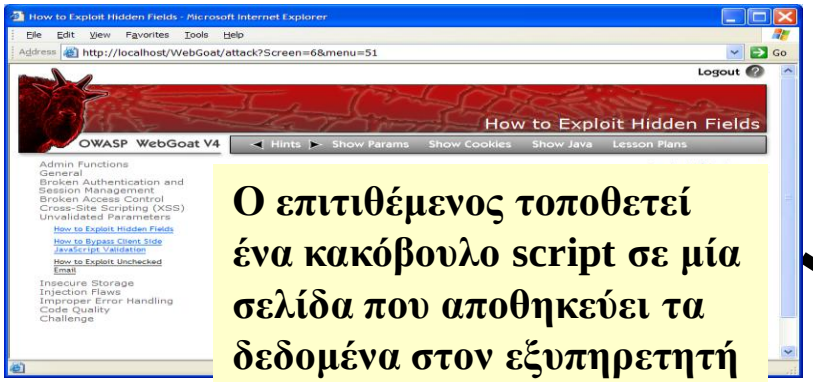

Leaving Site
Url=><script>alert('XSS');</script
My ASP
Alert http://localhost:85/
XSS
OK
Home
You are now le
continue

A2. Cross-Site Scripting

1

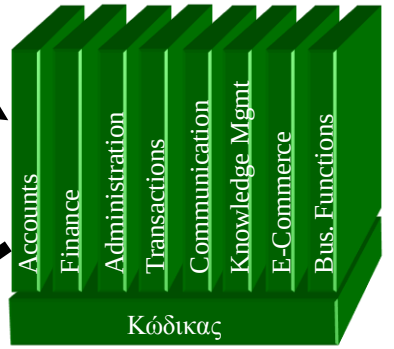


Ο επιτιθέμενος τοποθετεί την παγίδα



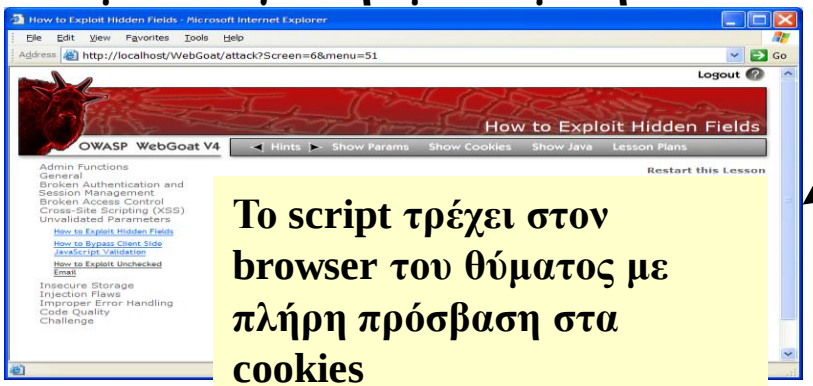
Ο επιτιθέμενος τοποθετεί ένα κακόβουλο script σε μία σελίδα που αποθηκεύει τα δεδομένα στον εξυπηρετητή

Εφαρμογή με ευπάθεια τύπου «stored XSS»



2

Το θύμα ανοίγει τη «μολυσμένη» σελίδα



Το script τρέχει στον browser του θύματος με πλήρη πρόσβαση στα cookies

3 **Το script σιωπηλά στέλνει στον επιτιθέμενο το cookie συνόδου του θύματος**

A2 – Cross-Site Scripting (XSS)

Πραγματοποιείται κάθε φορά που…

- Μη επεξεργασμένα δεδομένα από έναν επιτιθέμενο αποστέλλονται στον φυλλομετρητή ενός χρήστη.

Μη επεξεργασμένα δεδομένα…

- Αποθηκεύονται σε βάση δεδομένων
- «Αντικατοπτρίζονται» από δεδομένα εισόδου (π.χ. από φόρμες, κρυφά πεδία, URLs, κλπ.)
- Αποστέλλονται απευθείας σε έναν JavaScript client

Όλες οι web εφαρμογές έχουν αυτό το «πρόβλημα»

- Δοκιμάστε αυτό στον browser – javascript:alert (document.cookie)

Συνήθεις Επιπτώσεις

- Υποκλοπή της συνόδου του χρήστη, ευαίσθητων δεδομένων, αλλαγή περιεχομένου σελίδων, ανακατεύθυνση χρήστη σε site με phishing ή malware.
- Πιο σημαντικό: Εγκατάσταση XSS proxy που επιτρέπει τον επιτιθέμενο να παρατηρεί και να κατευθύνει όλη τη συμπεριφορά και κίνηση του χρήστη σε ευάλωτες σελίδες ανακατευθύνοντας τον αλλού.

A2 – Αποφυγή Ευπαθειών τύπου XSS

- Συστάσεις
  - Εξάλειψη ευπάθειας
    - Μη χρήση δεδομένων εισόδου από χρήστες σε άλλες σελίδες
  - Προστασία από την ευπάθεια
    - Σύσταση: Κωδικοποίηση των δεδομένων εξόδου που προέρχονται από δεδομένα εισόδου από χρήστες. (π.χ. χρησιμοποιώντας το OWASP ESAPI)

      http://www.owasp.org/index.php/ESAPI
    - Επαλήθευση βάσει 'white list' δεδομένων εισόδου για όλα τα δεδομένα που εισάγει ο χρήστης και θα συμπεριληφθούν σε κάποια σελίδα
    - Για τη διασφάλιση μεγάλων κομματιών κώδικα HTML που προέρχονται από χρήστες, μπορεί να χρησιμοποιηθεί η βιβλιοθήκη OWASP AntiSamy

      http://www.owasp.org/index.php/AntiSamy

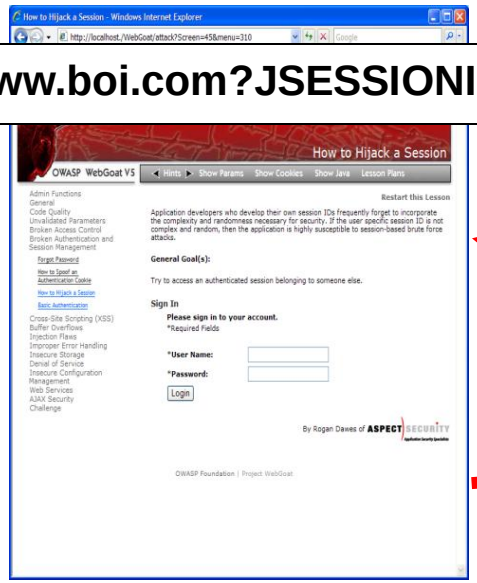
(AntiSamy)

- Αναφορές
  - Περισσότερες πληροφορίες για κωδικοποίηση δεδομένων εξόδου στο: http://www.owasp.org/index.php/XSS_(Cross Site Scripting) Prevention Cheat Sheet

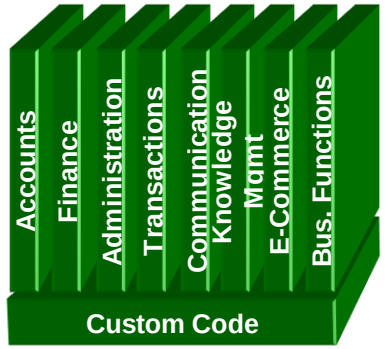
A3. Διαχείριση δεδομένων αυθεντικοποίησης και συνόδου

1

Είσοδος χρήστη



www.boi.com?JSESSIONID=9FA1DB9EA...



2

Το site χρησιμοποιεί URL rewriting

3

Ο χρήστης πηγαίνει στο http://www.hacker.com μέσω μιας φόρμας

Ο Hacker ελέγχει το referrer logs στο www.hacker.com και βρίσκει το JSESSIONID του χρήστη

4

5

Ο Hacker χρησιμοποιεί το JSESSIONID και αποκτά πρόσβαση στο λογαριασμό του χρήστη

A3 – Διαχείριση δεδομένων αυθεντικοποίησης και συνόδου

Το HTTP είναι “stateless”

- Δεδομένα αυθεντικοποίησης μεταφέρονται σε κάθε αίτηση
- Απαραίτητη η χρήση SSL οπουδήποτε χρειάζεται αυθεντικοποίηση

Ευπάθειες διαχείρισης συνόδου

- Χρησιμοποιείται το SESSION ID για την παρακολούθηση της κατάστασης (state)
  - Πολλές φορές ισοδυναμεί με την παραχώρηση δεδομένων αυθεντικοποίησης στον επιτιθέμενο
- Το SESSION ID είναι συνήθως εκτεθειμένο στο δίκτυο, το φυλλομετρητή, σε logs, κλπ.

Προσοχή στις κεκρόπορτες...

- Αλλαγή συνθηματικού, υπενθύμιση συνθηματικού, «μυστικές ερωτήσεις», αποσύνδεση (logout), κλπ.

Συνήθεις Επιπτώσεις

- Υποκλοπή λογαριασμών ή συνόδων χρηστών

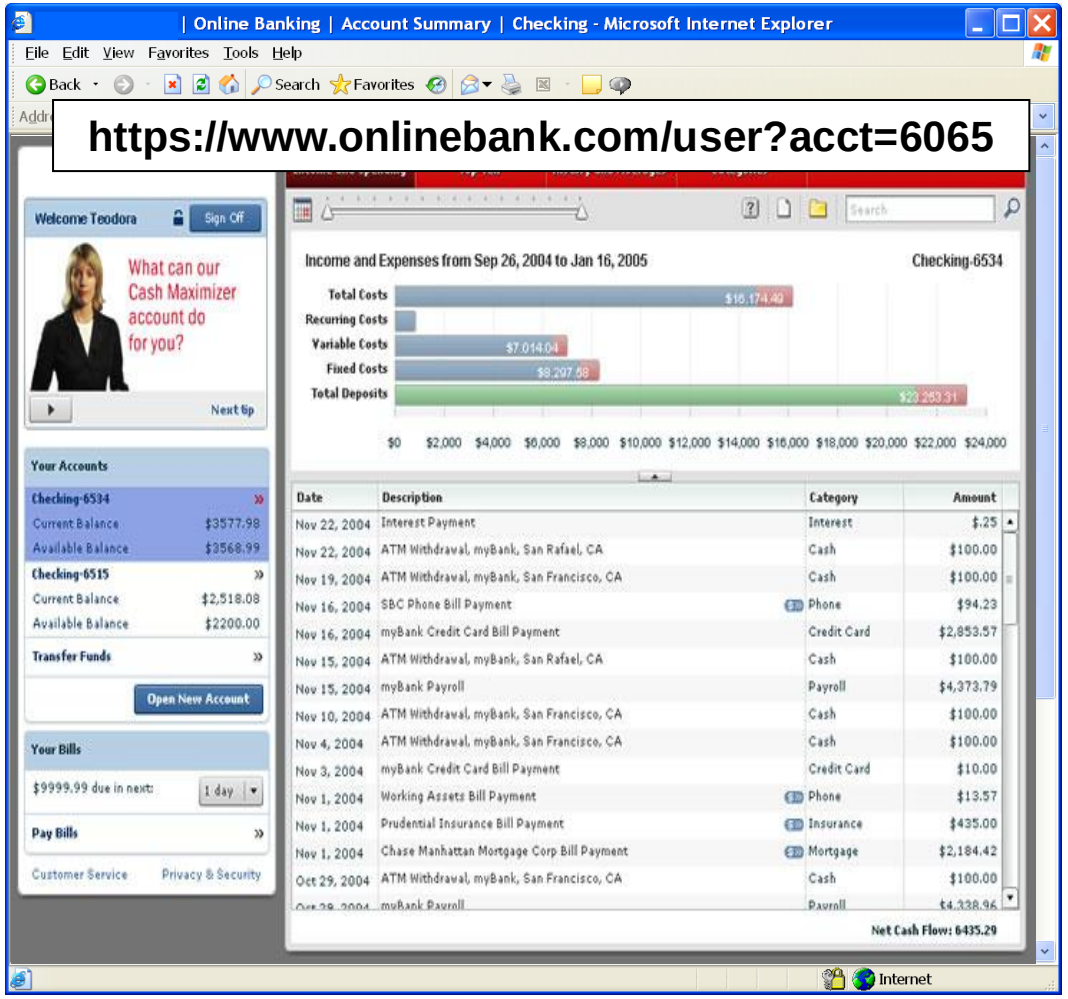
A3 – Ορθή διαχείριση δεδομένων αυθεντικοποίησης και συνόδου

- Έλεγχος αρχιτεκτονικής
  - Η αυθεντικοποίηση πρέπει να βασίζεται σε απλά, κεντρικοποιημένα πρότυπα.
  - Χρήση του κλασικού αναγνωριστικού εισόδου όπως παρέχεται από τον container
  - Βεβαιωθείτε ότι το SSL προστατεύει τα δεδομένα αυθεντικοποίησης και το αναγνωριστικό συνόδου (session id) διαρκώς

A3 – Ορθή διαχείριση δεδομένων αυθεντικοποίησης και συνόδου

- Έλεγχος υλοποίησης
  - Αδυναμία χρήσης μεθόδων αυτοματοποιημένης ανάλυσης
  - Έλεγχος πιστοποιητικού SSL
  - Επισκόπηση όλων των διαδικασιών και μεθόδων που σχετίζονται με την αυθεντικοποίηση
  - Βεβαιωθείτε ότι με την «αποσύνδεση» του χρήστη καταστρέφεται και η σύνοδος
  - Έλεγχος της υλοποίησης με το OWASP WebScarab
- Περισσότερες πληροφορίες στο:
  - http://www.owasp.org/index.php/Authentication_Cheat_Sheet

A4. Επισφαλείς αναφορές σε αντικείμενα



- Ο επιτιθέμενος παρατηρεί ότι η παράμετρος acct έχει τιμή 6065

  ?acct=6065

- Την αντικαθιστά με έναν κοντινό αριθμό

  ?acct=6066

- Βλέπει πληροφορίες για το λογαριασμό ενός άλλου χρήστη

A4 – Επισφαλείς Αναφορές σε Αντικείμενα

Πώς προστατεύετε την πρόσβαση στα δεδομένα σας;

- Αποτελεί μέρος της διαδικασίας επιβολής ορθών πρακτικών αυθεντικοποίησης και εξουσιοδότησης, όπως και το A8 – Αδυναμία περιορισμού πρόσβασης σε URL

Ένα συνηθισμένο λάθος …

- Εμφανίζοντας μόνο τα «εξουσιοδοτημένα» αντικείμενα για τον τρέχοντα χρήστη ή
- Αποκρύπτοντας τις αναφορές σε αντικείμενα σε κρυφά πεδία
- … χωρίς να επιβάλλονται οι περιορισμοί αυτό σε επίπεδο εξυπηρετητή.
- Αυτό ονομάζεται έλεγχος πρόσβασης σε επίπεδο παρουσίασης (presentation layer access control) και δεν έχει αποτέλεσμα στην πράξη.
- Ο επιτιθέμενος απλά τροποποιεί τιμές παραμέτρων

Συνήθεις Επιπτώσεις

- Μη εξουσιοδοτημένη πρόσβαση σε αρχεία ή δεδομένα

A4 – Ορθές και Ασφαλείς Αναφορές σε Αντικείμενα

- Αποφυγή απευθείας αναφορών σε αντικείμενα (direct object reference)
  - Αντικατάστασή τους με αντιστοίχηση σε προσωρινές τιμές (π.χ. 1, 2, 3)
  - Το OWASP ESAPI παρέχει υποστήριξη για αριθμητικές και τυχαίες αντιστοιχήσεις
    - IntegerAccessReferenceMap & RandomAccessReferenceMap

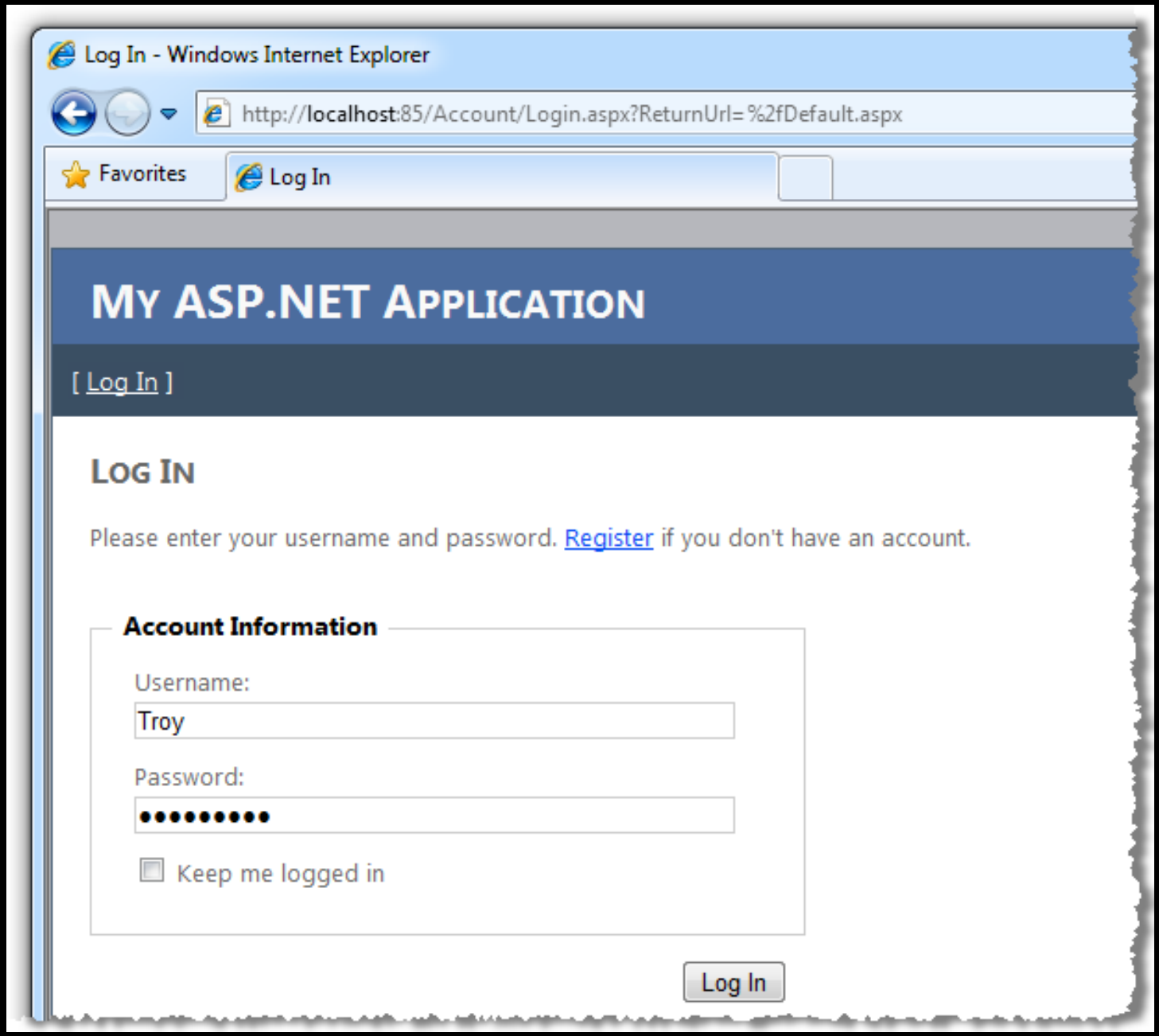
http://app?file=Report123.xls
http://app?file=1

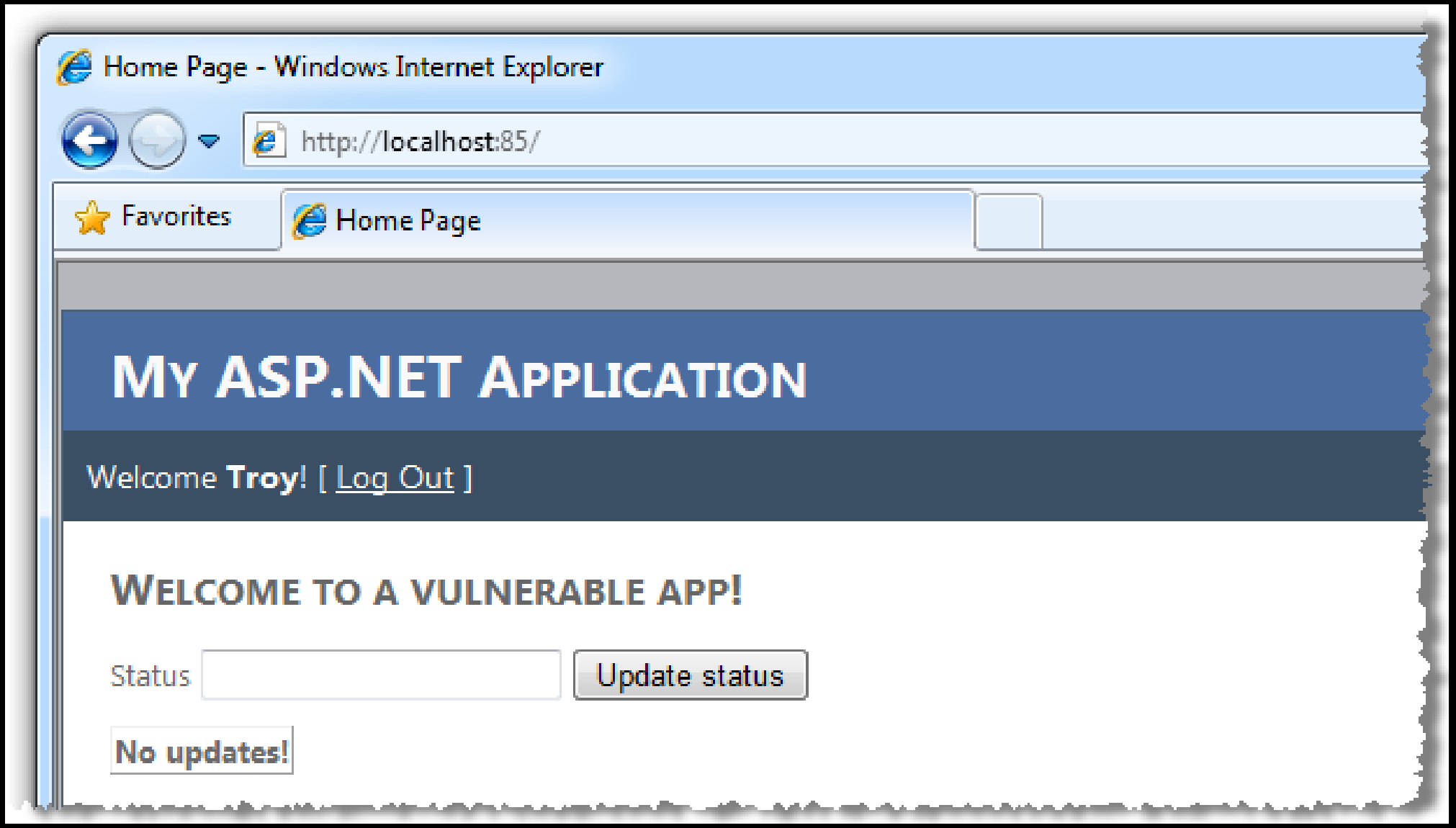
http://app?id=9182374
http://app?id=7d3J93

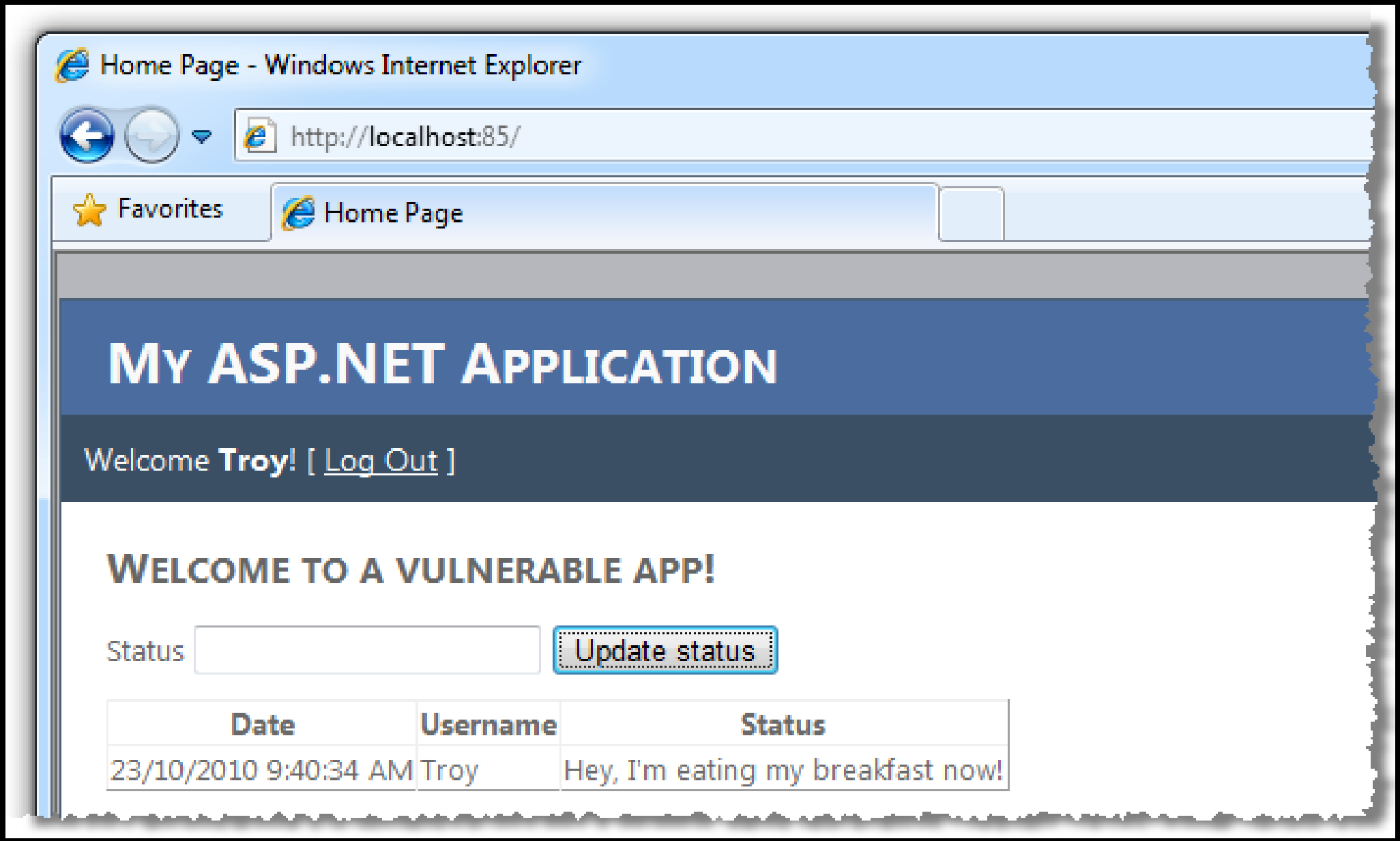
Report123.xls

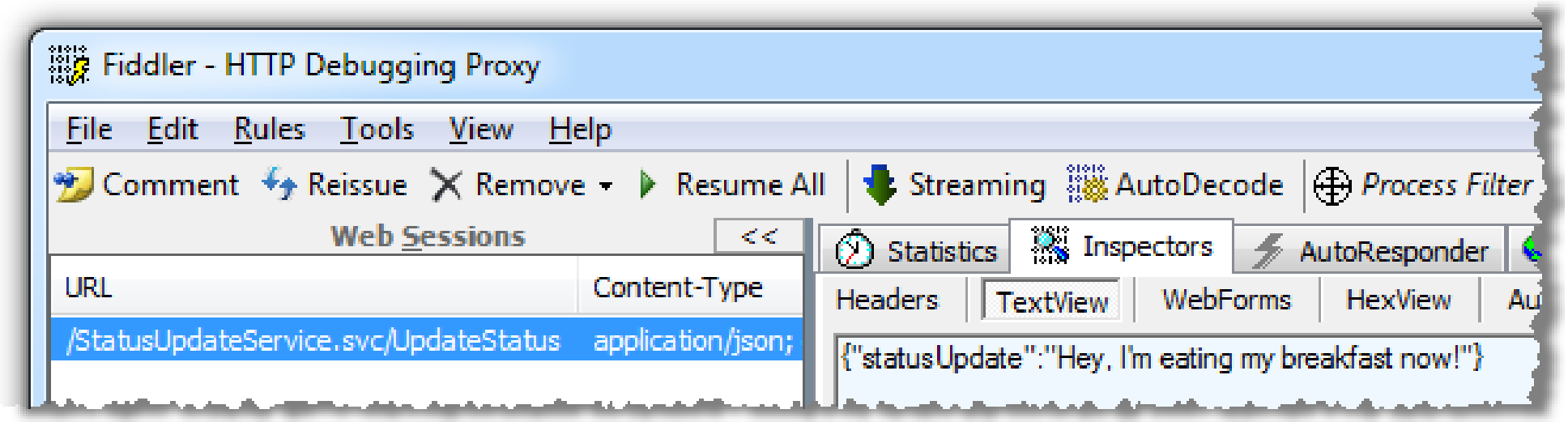
Acct:9182374

- Έλεγχος των απευθείας αναφορών
  - Έλεγχος ότι η τιμή της παραμέτρου είναι σωστά δομημένη
  - Έλεγχος ότι ο χρήστης είναι εξουσιοδοτημένος να προσπελάσει το αντικείμενο
  - Έλεγχος ότι έχει τα δικαιώματα προσπέλασης τα οποία ζητά να χρησιμοποιήσει (π.χ., ανάγνωση, εγγραφή, διαγραφή)

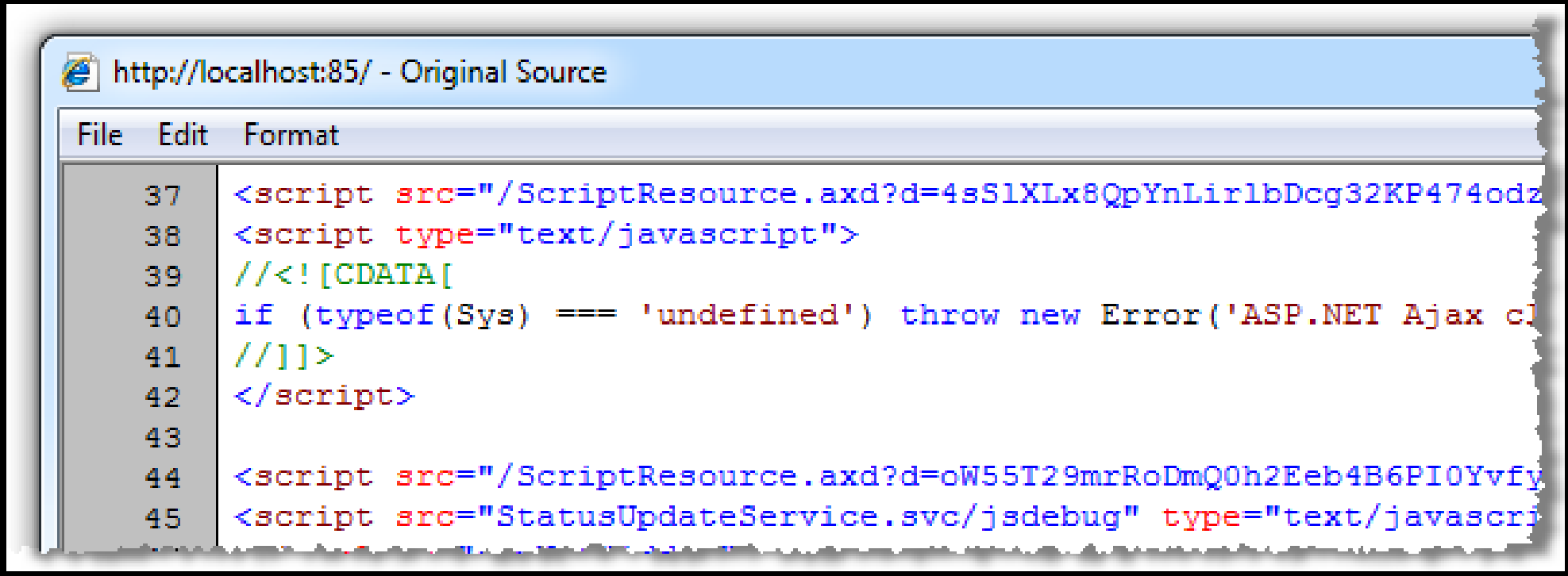

Log In - Windows Internet Explorer
http://localhost:85/Account/Login.aspx?ReturnUrl=%2fDefault.aspx
Favorites
Log In
MY ASP.NET APPLICATION
[Log In]
LOG IN
Please enter your username and password. Register if you don't have an account.
Account Information
Username:
Troy
Password:
Keep me logged in
Log In


Home Page - Windows Internet Explorer
http://localhost:85/
Favorites
Home Page
My ASP.NET Application
Welcome Troy! [Log Out]
Welcome to a vulnerable app!
Status
Update status
No updates!

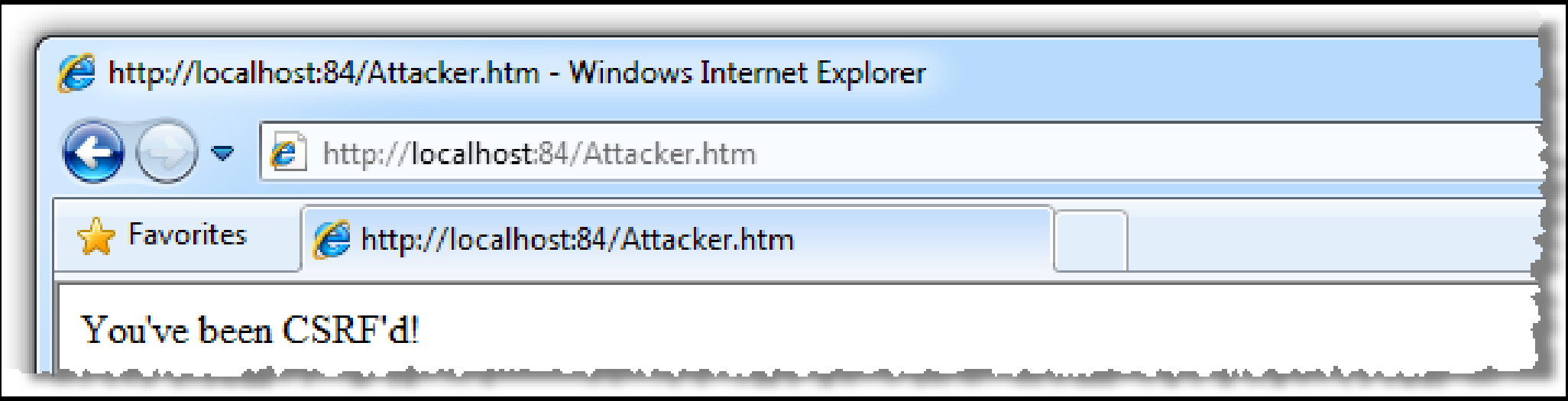

Home Page - Windows Internet Explorer
http://localhost:85/
Favorites
Home Page
MY ASP.NET APPLICATION
Welcome Troy! [Log Out]
WELCOME TO A VULNERABLE APP!
Status
Update status
Date
Username
Status
23/10/2010 9:40:34 AM
Troy
Hey, I'm eating my breakfast now!

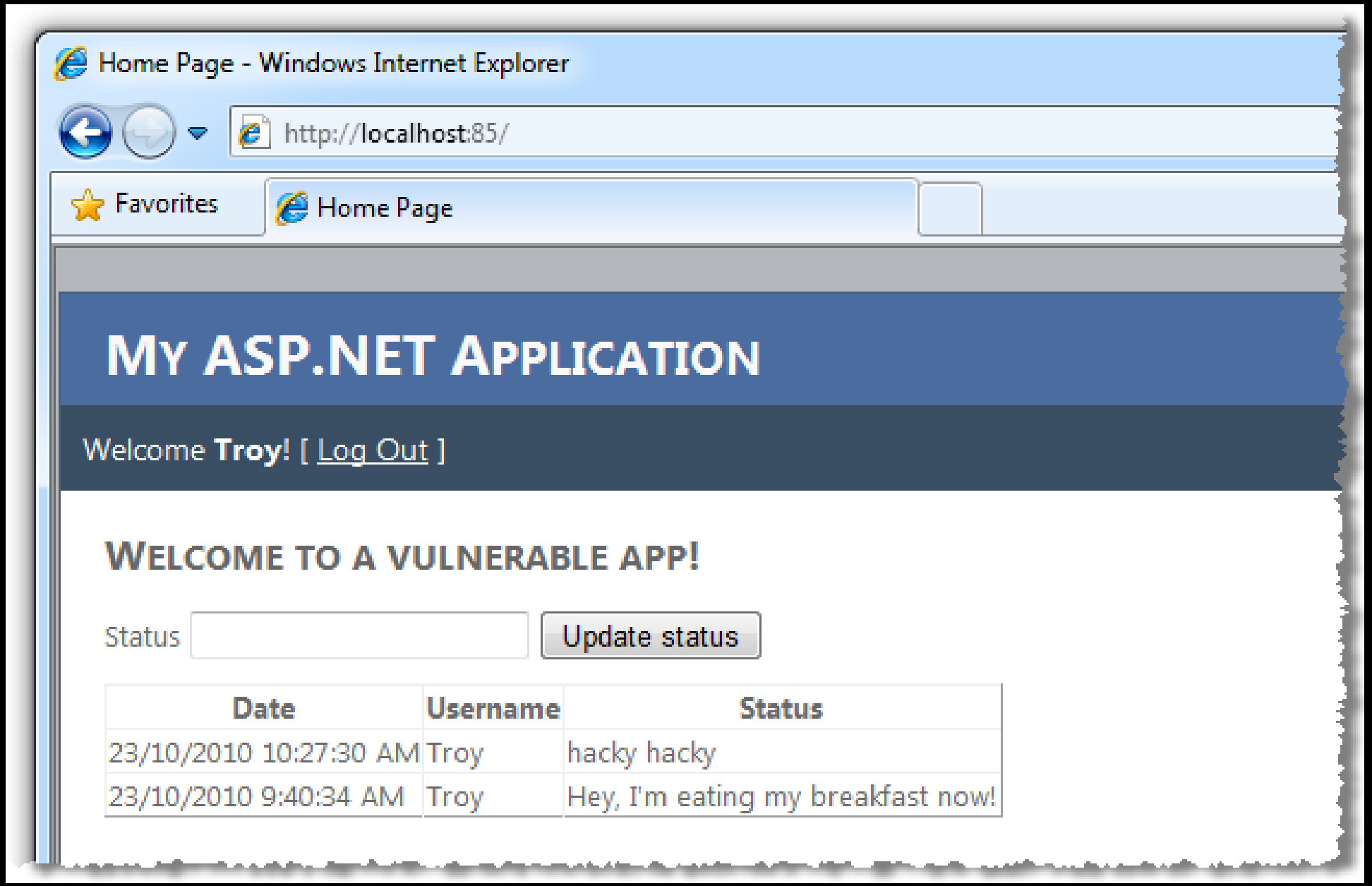


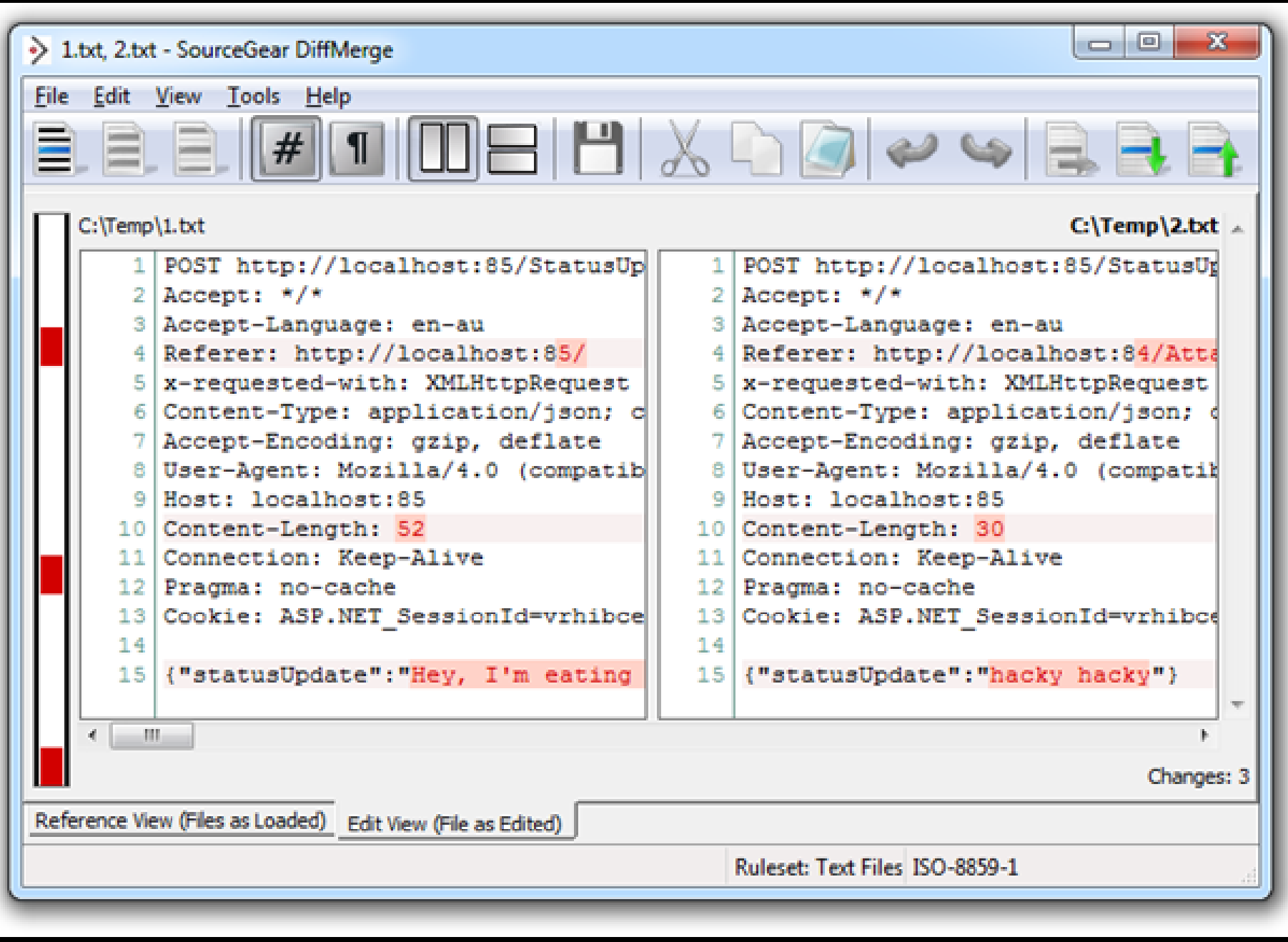

```
<input type="button" value="Update status" onclick="return UpdateStatus()" />
<script language="javascript" type="text/javascript">
// <![CDATA[
    function UpdateStatus() {
        var service = new Web.StatusUpdateService();
        var statusUpdate = document.getElementById('txtStatusUpdate').value;
        service.UpdateStatus(statusUpdate, onSuccess, null, null);
    }
    function onSuccess(result) {
         var statusUpdate = document.getElementById('txtStatusUpdate').value = "";
         __doPostBack('MainContent_updStatusUpdates', '');
    }
// ]]>
</script>
```


http://localhost:85/ - Original Source
File Edit Format
37 <script src="/ScriptResource.axd?d=4sS1XLx8QpYnLirlbDcg32KP474odz
38 <script type="text/javascript">
39 //<![CDATA[
40 if (typeof(Sys) === 'undefined') throw new Error('ASP.NET Ajax cl
41 //]]>
42 </script>
43
44 <script src="/ScriptResource.axd?d=oW55T29mrRoDmQ0h2Eeb4B6PI0Yvfy
45 <script src="StatusUpdateService.svc/jsdebug" type="text/javascri

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
  <script src="http://localhost:85/ScriptResource.axd?d=4sSlXLx8QpYnLirlbD...
  <script src="http://localhost:85/ScriptResource.axd?d=oW55T29mrRoDmQ0h2E...
  <script src="http://localhost:85/StatusUpdateService.svc/jsdebug" type="...

  <script language="javascript" type="text/javascript">
   // <![CDATA[
      var service = new Web.StatusUpdateService();
      var statusUpdate = "hacky hacky";
      service.UpdateStatus(statusUpdate, null, null, null);
   // ]]>
   </script>
</head>

<body> You've been CSRF'd! </body>

</html>
```


http://localhost:84/Attacker.htm - Windows Internet Explorer
http://localhost:84/Attacker.htm
Favorites
http://localhost:84/Attacker.htm
You've been CSRF'd!


Home Page - Windows Internet Explorer
http://localhost:85/
Favorites
Home Page
My ASP.NET Application
Welcome Troy! [Log Out]
Welcome to a vulnerable app!
Status
Update status
Date
Username
Status
23/10/2010 10:27:30 AM
Troy
hacky hacky
23/10/2010 9:40:34 AM
Troy
Hey, I'm eating my breakfast now!


1.txt, 2.txt - SourceGear DiffMerge
File Edit View Tools Help
C:\Temp\1.txt
C:\Temp\2.txt
1 POST http://localhost:85/StatusUp
2 Accept: */*
3 Accept-Language: en-au
4 Referer: http://localhost:85/
5 x-requested-with: XMLHttpRequest
6 Content-Type: application/json; c
7 Accept-Encoding: gzip, deflate
8 User-Agent: Mozilla/4.0 (compatib
9 Host: localhost:85
10 Content-Length: 52
11 Connection: Keep-Alive
12 Pragma: no-cache
13 Cookie: ASP.NET_SessionId=vrhibce
14
15 {"statusUpdate":"Hey, I'm eating
1 POST http://localhost:85/StatusUp
2 Accept: */*
3 Accept-Language: en-au
4 Referer: http://localhost:84/Atte
5 x-requested-with: XMLHttpRequest
6 Content-Type: application/json; c
7 Accept-Encoding: gzip, deflate
8 User-Agent: Mozilla/4.0 (compatib
9 Host: localhost:85
10 Content-Length: 30
11 Connection: Keep-Alive
12 Pragma: no-cache
13 Cookie: ASP.NET_SessionId=vrhibce
14
15 {"statusUpdate":"hacky hacky"}
Changes: 3
Reference View (Files as Loaded)
Edit View (File as Edited)
Ruleset: Text Files
ISO-8859-1

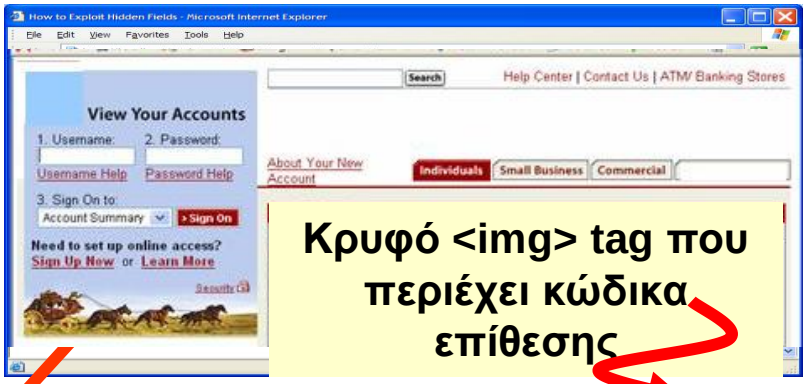
Πώς προκύπτει η ευπάθεια CSRF

- Το πρόβλημα
  - Οι φυλλομετρητές συμπεριλαμβάνουν αυτόματα δεδομένα αυθεντικοποίησης σε κάθε αίτηση.
  - Ακόμα και όταν οι αιτήσεις προκύπτουν μέσα από μια φόρμα, script, εικόνα ή άλλο site.
- Όλα τα sites που βασίζονται αποκλειστικά σε αυτόματη αυθεντικοποίηση είναι ευάλωτα!
  - (τα περισσότερα είναι έτσι…)
- Μηχανισμοί Αυτόματης Αυθεντικοποίησης
  - Cookie συνόδου
  - Basic authentication header
  - Διεύθυνση IP
  - Πιστοποιητικά SSL του πελάτη
  - Αυθεντικοποίηση βάση Windows domain

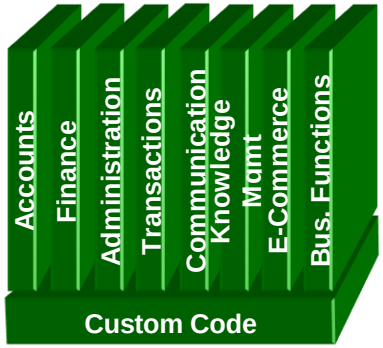
A5. Cross-Site Request Forgery (CSRF)

1

Ο επιτιθέμενος τοποθετεί την παγίδα σε ένα site που ελέγχει (ή στέλνει ένα e-mail)

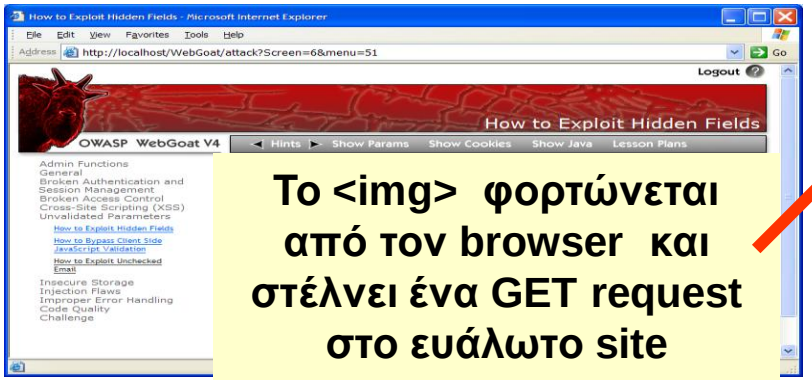


Εφαρμογή ευάλωτη σε CSRF



2

Όταν ο χρήστης μπει στο ευάλωτο site βλέπει και το site (ή το e-mail) του επιτιθέμενου



3

Το ευάλωτο site βλέπει μία κανονική αίτηση από το χρήστη και την εκτελεί

A5 – Cross Site Request Forgery (CSRF)

Cross Site Request Forgery

- Μία επίθεση κατά την οποία ο φυλλομετρητής του θύματος εξαπατείται στο να εκτελέσει μια εντολή σε ευπαθή εφαρμογή
- Προκαλείται από το γεγονός ότι οι φυλλομετρητές εμπεριέχουν αυτόματα δεδομένα αυθεντικοποίησης (session ID, διεύθυνση IP, διαπιστευτήρια Windows domain, …) σε κάθε αίτηση.

Φανταστείτε…

- Έναν hacker να μπορεί να κουνήσει το ποντίκι σας και να σας εξαναγκάσει να επιλέξετε συνδέσμους στο e-banking που χρησιμοποιείτε…
- Τι θα μπορούσε να σας εξαναγκάσει να κάνετε;

Συνήθεις Επιπτώσεις

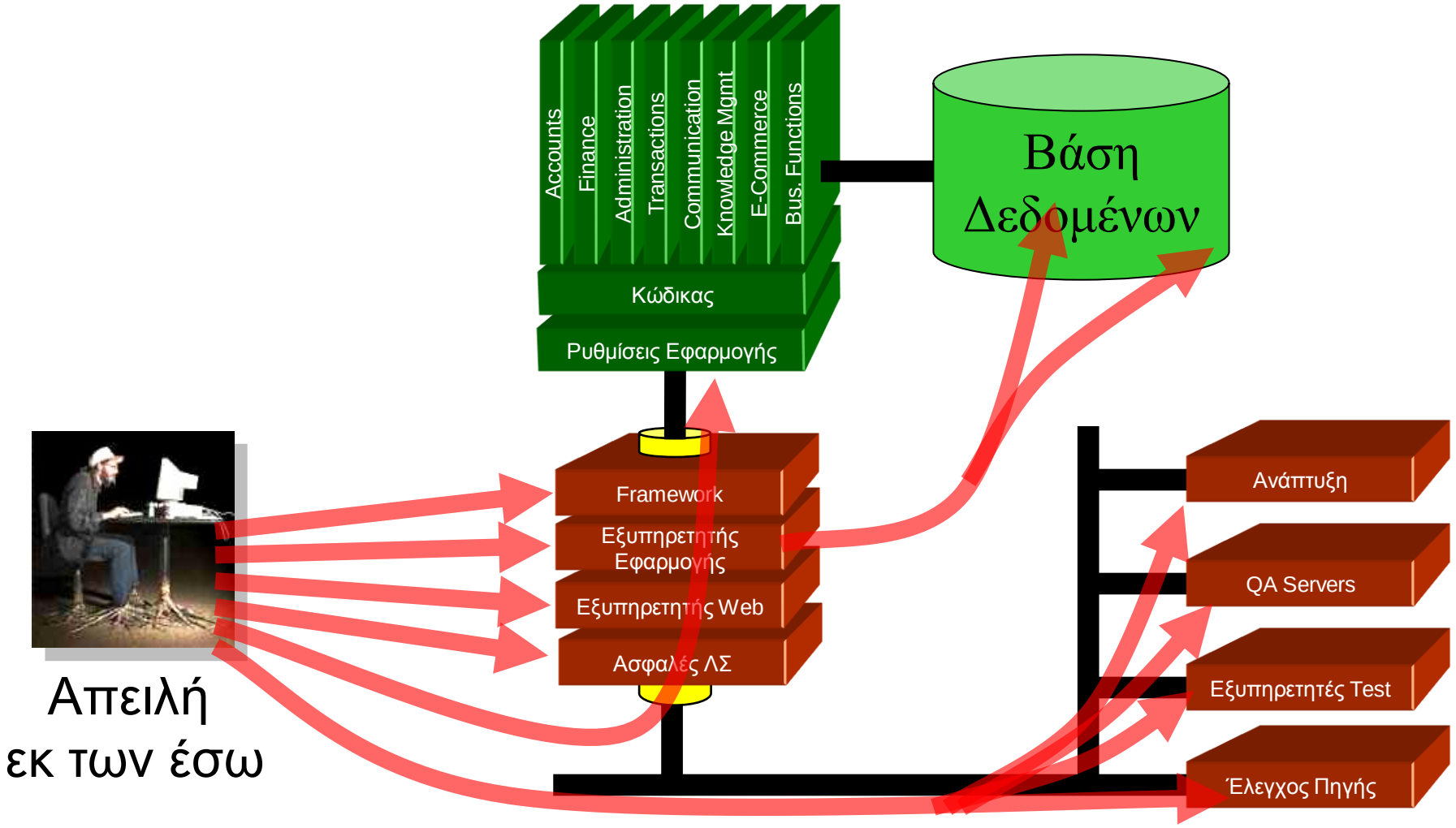
- Διενέργεια συναλλαγών (μεταφορά χρημάτων, αποσύνδεση χρήστη, κλείσιμο λογαριασμού)
- Πρόσβαση σε ευαίσθητα δεδομένα
- Αλλαγή στοιχείων λογαριασμού

A5 – Αποφυγή Ευπαθειών τύπου CSRF

- Προσθήκη ενός μυστικού token σε ΟΛΕΣ τις ευαίσθητες αιτήσεις, το οποίο δεν υποβάλλεται αυτόματα
  - Με τον τρόπο αυτό ο επιτιθέμενος δεν μπορεί να παραχαράξει την αίτηση
    - (εκτός αν υπάρχει και ευπάθεια τύπου XSS)
  - Τα tokens θα πρέπει να είναι τυχαία ή να βασίζονται σε ισχυρή κρυπτογραφία
- Επιλογές υλοποίησης
  - Αποθήκευση ενός μοναδικού token στη σύνοδο και προσθήκη του σε όλες τις φόρμες και τους συνδέσμους
    - **Κρυφό πεδίο:** `<input name="token" value="687965fdfaew87agrde" type="hidden"/>`
    - **Μοναδική χρήση URL:** `/accounts/687965fdfaew87agrde`
    - **Token σε φόρμα:** `/accounts?auth=687965fdfaew87agrde`
  - Προσοχή στην επικεφαλίδα αναφοράς (referrer header)
    - Προτείνεται να χρησιμοποιούνται κρυφά πεδία
  - Μοναδικό token για κάθε λειτουργία
    - Π.χ. σύνοψη του ονόματος της συνάρτησης, του αναγνωριστικού συνόδου μαζί με ένα μυστικό
  - Χρήση δευτερεύουσας αυθεντικοποίησης για ευαίσθητες λειτουργίες

Για περισσότερες πληροφορίες: www.owasp.org/index.php/CSRF_Prevention_Cheat_Sheet

Α6. Επισφαλείς Ρυθμίσεις Ασφάλειας



A6 – Επισφαλείς Ρυθμίσεις Ασφάλειας

Οι εφαρμογές βασίζονται σε ασφαλή θεμέλια

- Παντού, από το ΛΣ μέχρι τον εξυπηρετητή εφαρμογών
- Μην ξεχνάτε τις βιβλιοθήκες που χρησιμοποιείτε

Είναι ο κώδικας μυστικός;

- Εντοπίστε όλα τα σημεία που αποθηκεύεται ο πηγαίος κώδικας
- Η ασφάλεια δεν πρέπει να προϋποθέτει κρυφό πηγαίο κώδικα

Διαχείριση δεδομένων αυθεντικοποίησης σε κάθε σημείο της εφαρμογής

- Όλα τα δεδομένα αυθεντικοποίησης πρέπει να αλλάζουν στην παραγωγή.

Συνήθεις Επιπτώσεις

- Εγκατάσταση κακόβουλου λογισμικού (backdoor) εκμεταλλευόμενοι την έλλειψη ενημερώσεων ΛΣ ή εξυπηρετητών.
- Ευπάθειες τύπου XSS δημιουργούνται από την έλλειψη ενημερώσεων από τα πλαίσια λειτουργίας των εφαρμογών
- Μη εξουσιοδοτημένη πρόσβαση σε κοινούς (default) λογαριασμούς, δεδομένα ή λειτουργίες εφαρμογής ή μη χρησιμοποιούμενες λειτουργίες

A6 – Ορθές Ρυθμίσεις Ασφάλειας

- Έλεγχος και διαχείριση των ρυθμίσεων του συστήματος
  - Οδηγίες για σωστή διασφάλιση του συστήματος (“hardening”)
    - Η αυτοματοποίηση βοηθά πολύ
  - Κάλυψη όλης της πλατφόρμας και της εφαρμογής
  - Εγκατάσταση ενημερώσεων ασφάλειας για ΟΛΑ τα στοιχεία
    - Βιβλιοθήκες λογισμικού, κλπ. Όχι μόνο ΛΣ και εφαρμογές εξυπηρετητή
  - Ανάλυση των πιθανών επιπτώσεων στην ασφάλεια από αλλαγές

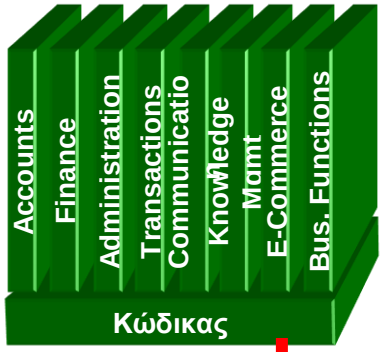
- Μπορείτε να αποτυπώσετε τις ρυθμίσεις αυτές;
  - Δημιουργία αναφορών για τις διαδικασίες ρυθμίσεων
  - Αν δεν μπορείτε να τις επαληθεύσετε, τότε δεν είναι ασφαλείς.

- Έλεγχος υλοποίησης
  - Αυτοματοποιημένος εντοπισμός προβλημάτων στις ρυθμίσεις ή έλλειψη ενημερώσεων ασφάλειας

A7. Επισφαλής Κρυπτογραφημένη Αποθήκευση

1

Ο χρήστης εισάγει αριθμό πιστωτικής κάρτας σε μία φόρμα



2

Ο διαχειριστής λαθών καταγράφει τον αριθμό της κάρτας επειδή το site της τράπεζας δεν είναι διαθέσιμο

3

Τα logs είναι διαθέσιμα σε όλη τη διεύθυνση πληροφορικής για debugging

4

Κακόβουλος υπάλληλος υποκλέπτει χιλιάδες αριθμούς καρτών

A7 – Επισφαλής Κρυπτογραφημένη Αποθήκευση

Επισφαλής αποθήκευση ευαίσθητων δεδομένων

- Αδυναμία εντοπισμού όλων των ευαίσθητων δεδομένων
- Αδυναμία εντοπισμού όλων των σημείων που αποθηκεύονται εμπιστευτικά δεδομένα
  - Βάσεις δεδομένων, αρχεία, φάκελοι, logs, backup, κλπ.
- Αδυναμία προστασίας τέτοιων δεδομένων σε όλα τα σημεία.

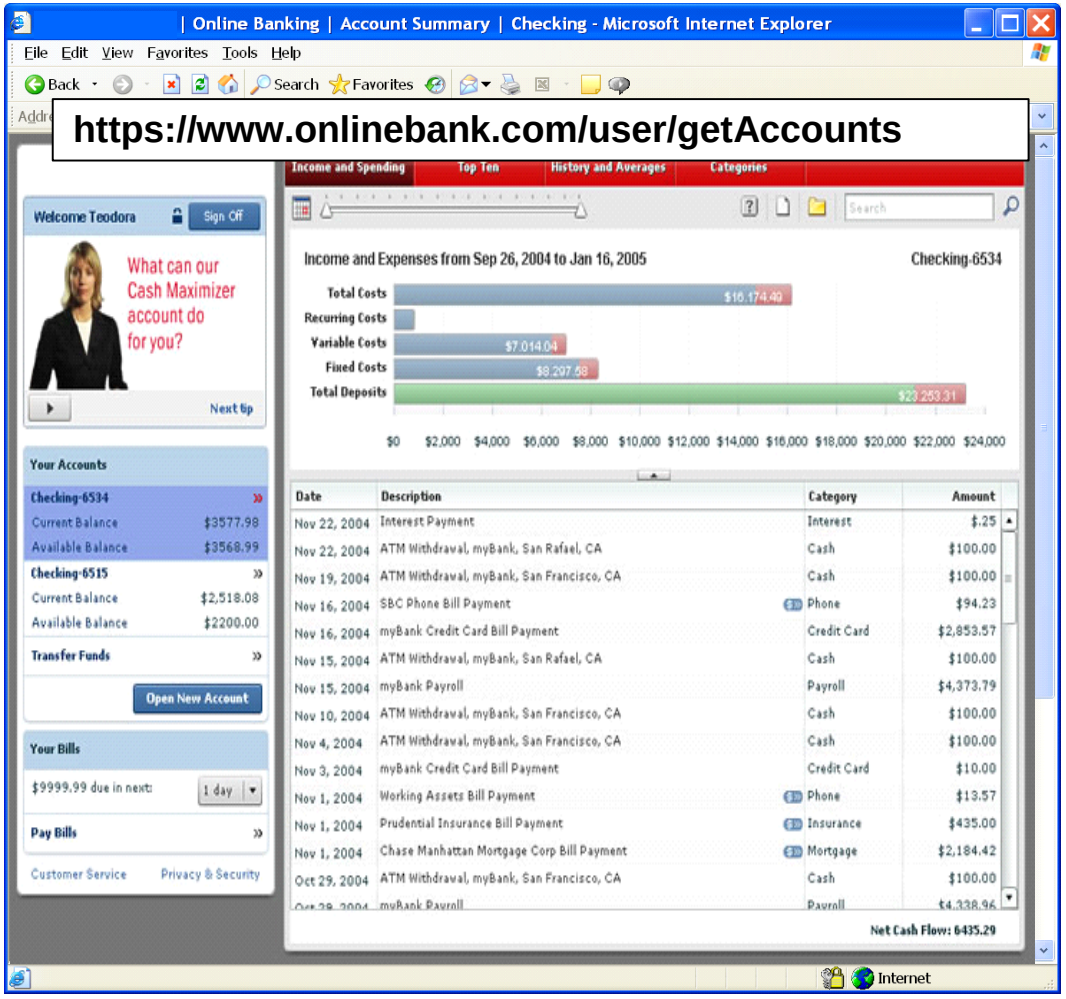
Συνήθεις Επιπτώσεις

- Οι επιτιθέμενοι έχουν πρόσβασή ή μπορούν να αλλοιώσουν εμπιστευτικές και ευαίσθητες πληροφορίες
  - Π.χ. πιστωτικές κάρτες, δεδομένα υγείας, οικονομικά στοιχεία (πελατών ή του οργανισμού), κλπ.
- Οι επιτιθέμενοι αποκτούν ευαίσθητες πληροφορίες που μπορούν στη συνέχεια να τις χρησιμοποιήσουν σε επόμενες επιθέσεις.
- Δυσφήμηση του οργανισμού, δυσαρέσκεια πελατών, απώλεια εμπιστοσύνης
- Κόστος επαναφοράς και διερεύνησης του περιστατικού: ψηφιακά πειστήρια, αλληλογραφία ενημέρωσης, επανέκδοση πιστωτικών καρτών, ασφάλεια, κλπ.
- Νομικές και οικονομικές κυρώσεις (συμμόρφωση)

A7 – Ασφαλής Κρυπτογραφημένη Αποθήκευση

- Έλεγχος αρχιτεκτονικής
  - Εντοπισμός ευαίσθητων δεδομένων
  - Εντοπισμός των σημείων που αποθηκεύονται τα δεδομένα
  - Χρήση κρυπτογράφησης για την αντιμετώπιση των απειλών
- Προστασία με κατάλληλους μηχανισμούς
  - Κρυπτογράφηση αρχείων, βάσεων δεδομένων, κλπ.
- Σωστή χρήση των μηχανισμών
  - Χρήση γνωστών, ισχυρών αλγορίθμων
  - Ασφαλής δημιουργία, διανομή και προστασία των κλειδιών
  - Συχνή αλλαγή κλειδιών
- Έλεγχος υλοποίησης
  - Χρήση ισχυρού, προτυποποιημένου αλγορίθμου που ταιριάζει για τη συγκεκριμένη χρήση
  - Όλα τα κλειδιά, πιστοποιητικά και συνθηματικά αποθηκεύονται με ασφάλεια και προστατεύονται
  - Ασφαλής διανομή κλειδιών και ύπαρξη σχεδίου αλλαγής κλειδιών
  - Ανάλυση του κώδικα κρυπτογράφησης για συνήθη λάθη

A8. Αδυναμία περιορισμού πρόσβασης σε URL



- Ο επιτιθέμενος παρατηρεί ότι στο URL αναφέρεται ο ρόλος του

  /user/getAccounts

- Τον αλλάζει σε άλλο ρόλο (φάκελο)

  /admin/getAccounts, or

  /manager/getAccounts

- Έτσι βλέπει και άλλους λογαριασμούς, υψηλότερων δικαιωμάτων

A8 – Αδυναμία περιορισμού πρόσβασης σε URL

Πώς προστατεύουμε πρόσβαση σε URLs (σελίδες);

- Αφορά την επιβολή αυθεντικοποίησης σε συνδυασμό με το A4 – Επισφαλείς Αναφορές σε Αντικείμενα

Ένα συνηθισμένο λάθος…

- Εμφάνιση μόνο εξουσιοδοτημένων συνδέσμων και επιλογών.
- Αυτό ονομάζεται έλεγχος πρόσβασης σε επίπεδο παρουσίασης.
- Ο επιτιθέμενος μπορεί απλά να αποκτήσει πρόσβαση σε «μη εξουσιοδοτημένες» σελίδες.

Συνήθεις Επιπτώσεις

- Οι επιτιθέμενοι εκτελούν διεργασίες και υπηρεσίες για τις οποίες δεν έχουν εξουσιοδότηση
- Πρόσβαση σε λογαριασμούς και δεδομένα άλλων χρηστών
- Εκτέλεση ενεργειών που απαιτούν δικαιώματα διαχειριστή

A8 – Περιορισμός πρόσβασης σε URLs

- Για κάθε URL, ένα site πρέπει να κάνει 3 πράγματα:
  - Περιορισμός πρόσβασης σε αυθεντικοποιημένους χρήστες (αν δεν είναι δημόσια)
  - Επιβολή δικαιωμάτων βάση χρήστη ή ρόλου (αν είναι ιδιωτική)
  - Καθολική απαγόρευση αιτήσεων για μη εξουσιοδοτημένους τύπους σελίδων (π.χ. αρχεία ρυθμίσεων, logs, αρχεία πηγαίου κώδικα, κλπ.)

- Έλεγχος αρχιτεκτονικής
  - Χρήση ενός απλού μοντέλου σε κάθε επίπεδο
  - Υπάρχει όντως μηχανισμός σε κάθε επίπεδο;

- Έλεγχος υλοποίησης
  - Οι αυτοματοποιημένες μέθοδοι ανάλυσης δεν είναι ακριβείς
  - Έλεγχος ότι κάθε URL στην εφαρμογή προστατεύεται είτε από:
    - Ένα εξωτερικό φίλτρο (Java EE web.xml) ή κάποιο εμπορικό προϊόν
    - Ή εσωτερικούς ελέγχους ενσωματωμένους στον κώδικα – Π.χ. μέθοδος isAuthorizedForURL() του ESAPI.
  - Έλεγχος ότι οι ρυθμίσεις του εξυπηρετητή δεν επιτρέπουν αιτήσεις για μη εξουσιοδοτημένους τύπους αρχείων.
  - Δοκιμές για μη εξουσιοδοτημένες αιτήσεις

A9. Επισφαλής Χρήση του TLS

Εξωτερικό θύμα

Κώδικας

Συστήματα Backend

Συνεργάτες

Υπάλληλοι

1

Ο επιτιθέμενος υποκλέπτει διαπιστευτήρια και δεδομένα από το δίκτυο

Εξωτερικός εισβολέας

2

Ο επιτιθέμενος υποκλέπτει διαπιστευτήρια και δεδομένα από το εσωτερικό δίκτυο

Εισβολέας εκ των έσω

A9 – Επισφαλής Χρήση του TLS

Επισφαλής αποστολή ευαίσθητων δεδομένων

- Αδυναμία εντοπισμού όλων των ευαίσθητων δεδομένων
- Αδυναμία εντοπισμού όλων των παραληπτών των ευαίσθητων δεδομένων
  - Στο web, σε βάσεις δεδομένων, σε συνεργάτες, εσωτερικά
- Αδυναμία προστασίας των δεδομένων αυτών σε κάθε σημείο

Συνήθεις Επιπτώσεις

- Οι επιτιθέμενοι έχουν πρόσβαση ή τροποποιούν εμπιστευτικές ή ευαίσθητες πληροφορίες.
  - Π.χ., πιστωτικές κάρτες, δεδομένα υγείας, οικονομικά δεδομένα, κλπ.
- Οι επιτιθέμενοι αποκτούν ευαίσθητες πληροφορίες που μπορούν στη συνέχεια να τις χρησιμοποιήσουν σε επόμενες επιθέσεις.
- Δυσφήμηση του οργανισμού, δυσαρέσκεια πελατών, απώλεια εμπιστοσύνης
- Κόστος επαναφοράς και διερεύνησης του περιστατικού
- Νομικές και οικονομικές κυρώσεις (συμμόρφωση)

A9 – Ορθή χρήση του TLS

- Προστασία με κατάλληλους μηχανισμούς
  - Χρήση TLS σε κάθε σύνδεση με ευαίσθητα δεδομένα
  - Κρυπτογράφηση μεμονωμένων μηνυμάτων πριν τη μετάδοσή τους
    - Π.χ., XML-Encryption
  - Υπογραφή μηνυμάτων πριν τη μετάδοσή τους:
    - Π.χ., XML-Signature
- Ορθή χρήση των μηχανισμών
  - Χρήση ισχυρών προτύπων και αλγορίθμων (απενεργοποίηση παλαιότερων αλγορίθμων - SSL)
  - Ορθή διαχείριση κλειδιών/πιστοποιητικών
  - Επαλήθευση πιστοποιητικών SSL πριν τη χρήση τους
  - Χρήση αποδεδειγμένα ασφαλών μηχανισμών
    - Π.χ., SSL ή XML-Encryption

- Περισσότερες πληροφορίες: http://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet

A10. Μη έγκυρες ανακατευθύνσεις

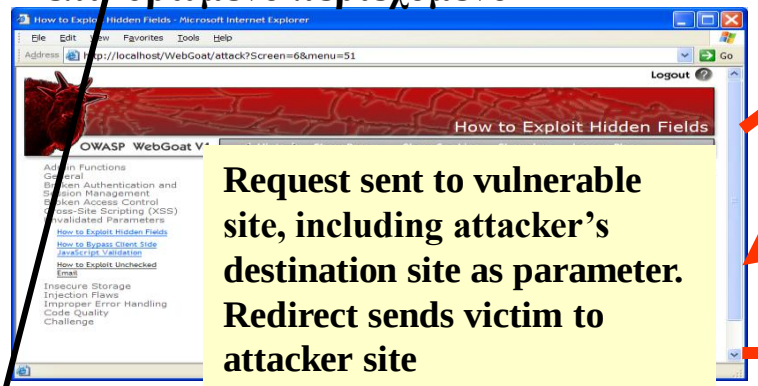
1

Η επίθεση γίνεται μέσα από σελίδα ή e-mail

From: Internal Revenue Service
Subject: Your Unclaimed Tax Refund
Our records show you have an unclaimed federal tax refund. Please click here to initiate your claim.

2

Το θύμα επιλέγει το σύνδεσμο που περιέχει μη-επικυρωμένο περιεχόμενο

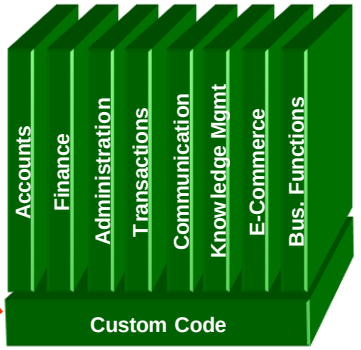


Request sent to vulnerable site, including attacker's destination site as parameter. Redirect sends victim to attacker site

http://www.irs.gov/taxrefund/claim.jsp?year=2006 & … &dest=www.evilsite.com

3

Η εφαρμογή ανακατευθύνει το χρήστη σε site του επιτιθέμενου



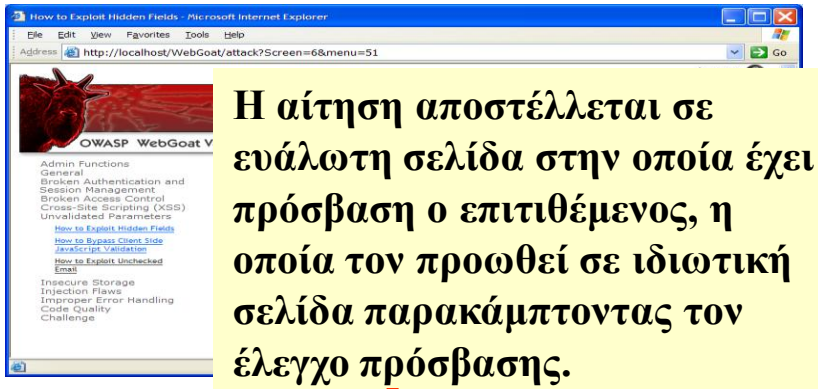
4

Στο site αυτό γίνεται επίθεση phishing ή εγκαθίσταται κακόβουλο λογισμικό στο χρήστη

A10. Μη έγκυρες προωθήσεις

1

Ο επιτιθέμενος στέλνει την επίθεση σε ευάλωτη σελίδα στην οποία έχει πρόσβαση



Η αίτηση αποστέλλεται σε ευάλωτη σελίδα στην οποία έχει πρόσβαση ο επιτιθέμενος, η οποία τον προωθεί σε ιδιωτική σελίδα παρακάμπτοντας τον έλεγχο πρόσβασης.

2

Η εφαρμογή εγκρίνει την αίτηση, και συνεχίζει την προώθηση στο ευάλωτο site

Φίλτρο

```
public void doPost( HttpServletRequest request,
HttpServletResponse response) {
    try {
        String target = request.getParameter( "dest" ) );
        ...
        request.getRequestDispatcher( target ).forward(request,
        response);
    }
    catch ( ...
```

```
public void sensitiveMethod(
HttpServletRequest request,
HttpServletResponse response) {
    try {
          // Do sensitive stuff here.
          ...
    }
    catch ( ...
```

3

Η σελίδα που κάνει την προώθηση δεν επικυρώνει την παράμετρο και στέλνει τον επιτιθέμενο σε μη εξουσιοδοτημένη σελίδα, παρακάμπτοντας τον έλεγχο πρόσβασης.

A10 – Μη έγκυρες ανακατευθύνσεις και προωθήσεις

Οι ανακατευθύνσεις είναι πολύ συνηθισμένες

- Και συχνά συμπεριλαμβάνουν παραμέτρους χρηστών στη διεύθυνση προορισμού.
- Εάν δεν επικυρωθούν σωστά μπορούν να στείλουν το θύμα σε διεύθυνση επιλογής του επιτιθέμενου.

Οι προωθήσεις (ή Transfer στο .NET) είναι εξίσου συνηθισμένες

- Στέλνουν εσωτερικά την αίτηση σε νέα σελίδα στην ίδια εφαρμογή.
- Μερικές φορές ο προορισμός καθορίζεται από παραμέτρους
- Ο επιτιθέμενος μπορεί να τις χρησιμοποιήσει για να παρακάμψει τον έλεγχο αυθεντικοποίησης και εξουσιοδότησης.

Συνήθεις Επιπτώσεις

- Ανακατεύθυνση του θύματος σε σελίδα με phishing ή malware.
- Η αίτηση του επιτιθέμενου προωθείται παρακάμπτοντας ελέγχους ασφάλειας και αποκτώντας πρόσβαση σε μη εξουσιοδοτημένες λειτουργίες ή δεδομένα.

A10 – Προστασία από μη έγκυρες ανακατευθύνσεις

- Πολλές επιλογές ☺
  1. Αποφυγή [κατά το δυνατό] προωθήσεων και ανακατευθύνσεων
  2. Εάν δε γίνεται, να μη χρησιμοποιούνται παράμετροι χρήστη για τον ορισμό της τελικής διεύθυνσης
  3. Εάν δε γίνεται, τότε
     a) Έλεγχος κάθε παραμέτρου για την εγκυρότητά της αλλά και την εξουσιοδότηση για το συγκεκριμένο χρήστη
     b) (προτιμάται) – Αντιστοίχηση στον εξυπηρετητή της επιλογής του χρήστη με την πραγματική σελίδα
- Προστασία εις βάθος: Για ανακατευθύνσεις, έλεγχος του τελικού URL μετά τον υπολογισμό του ώστε να διαπιστωθεί ότι κατευθύνει προς έγκυρο-εξουσιοδοτημένο εξωτερικό site.
- Χρήση ESAPI
  - Δείτε: SecurityWrapperResponse.sendRedirect(URL)
  - http://owasp-esapi-java.googlecode.com/svn/trunk_doc/org/owasp/esapi/filters/SecurityWrapperResponse.html#sendRedirect(java.lang.String)

A10 – Προστασία από μη έγκυρες προωθήσεις

- Πιθανές λύσεις για τις προωθήσεις
  - Ιδανικά, κλήση του ελεγκτή πρόσβασης ώστε να υπάρξει διασφάλιση ότι ο χρήστης έχει τις κατάλληλες εξουσιοδοτήσεις (ESAPI)
  - Χρήση εξωτερικών φίλτρων όπως το Siteminder, (όχι πολύ πρακτικό)
  - Διαβεβαίωση ότι οι χρήστες που έχουν πρόσβαση στην αρχική σελίδα μπορούν ΟΛΟΙ να έχουν πρόσβαση στην τελική σελίδα

PCI DSS 2.0 και Ασφάλεια Λογισμικού

- **6.5.1 Injection flaws**, particularly SQL injection. Also consider OS Command Injection, LDAP and XPath injection flaws as well as other injection flaws.
- **6.5.3 Insecure cryptographic storage** (Prevent cryptographic flaws)
- **6.5.4 Insecure communications**
- **6.5.5 Improper error handling** (Do not leak information via error messages)
- **6.5.6 All "High" vulnerabilities identified in the vulnerability identification process**
- **6.5.7 Cross-site scripting (XSS)**
- **6.5.8 Improper Access Control**, such as **insecure direct object reference, failure to restrict URL access**, and directory traversal)
- **6.5.9 Cross-site request forgery (CSRF)**

CWE/SANS Top 25 Most Dangerous Software Errors

- 3 κατηγορίες
  - Insecure Interaction Between Components
  - Risky Resource Management
  - Porous Defenses

Insecure Interaction Between Components

- [1] Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
- [2] Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
- [4] Cross-Site Request Forgery (CSRF)
- [8] Unrestricted Upload of File with Dangerous Type
- [9] Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
- [17] Information Exposure Through an Error Message
- [23] URL Redirection to Untrusted Site ('Open Redirect')
- [25] Race Condition

Risky Resource Management

- [3] Buffer Copy without Checking Size of Input ('Classic Buffer
- Overflow')
- [7] Improper Limitation of a Pathname to a Restricted Directory
- ('Path Traversal')
- [12] Buffer Access with Incorrect Length Value
- [13] Improper Check for Unusual or Exceptional Conditions
- [14] Improper Control of Filename for Include/Require Statement in
- PHP Program ('PHP File Inclusion')
- [15] Improper Validation of Array Index
- [16] Integer Overflow or Wraparound
- [18] Incorrect Calculation of Buffer Size
- [20] Download of Code Without Integrity Check
- [22] Allocation of Resources Without Limits or Throttling

Porous Defenses

- [5] Improper Access Control (Authorization)
- [6] Reliance on Untrusted Inputs in a Security Decision
- [10] Missing Encryption of Sensitive Data
- [11] Use of Hard-coded Credentials
- [19] Missing Authentication for Critical Function
- [21] Incorrect Permission Assignment for Critical Resource
- [24] Use of a Broken or Risky Cryptographic Algorithm

Web Application Firewalls (WAF)

- Συσκευή (appliance) που προστατεύει από γνωστές (ή μη) επιθέσεις
- Αντίστοιχη λειτουργία με το δικτυακό firewall αλλά σε επίπεδο εφαρμογής
- Μέθοδοι εντοπισμού
  - Υπογραφές
  - Βάση συμπεριφοράς
  - …
- Μέθοδοι λειτουργίας
  - Καταγραφή μόνο
  - Καταγραφή και αποτροπή

WAFs: Πανάκεια;

- Η ασφάλεια της εφαρμογής εξαρτάται από την ασφάλεια-ευρωστία του WAF
- Απαραίτητη η σωστή ρύθμισή του
  - Θετικώς και αρνητικώς εσφαλμένα
- Μπορεί να έχει επιπλέον ευπάθειες
- **Δημιουργεί λανθασμένη αίσθηση ασφάλειας**
  - Προωθείται σαν πανάκεια με αποτέλεσμα να μη δίνεται σημασία στην ασφαλή ανάπτυξη λογισμικού
- *Τελικά*: Απαραίτητο συμπλήρωμα του ασφαλούς κύκλου ζωής ανάπτυξης λογισμικού

Αντιμετώπιση ευπαθειών

- Ανάπτυξη Ασφαλούς Κώδικα
  - Διασφάλιση Λογισμικού και Ασφαλής Κύκλος Ζωής Ανάπτυξης Λογισμικού (OWASP OpenSAMM)
  - OWASP Guide to Building Secure Web Applications
    - http://www.owasp.org/index.php/Guide
  - OWASP Application Security Verification Standard
    - http://www.owasp.org/index.php/ASVS
  - Χρήση ασφαλών βιβλιοθηκών ανάλογα με τις εκάστοτε ανάγκες
    - Χρήση του OWASP ESAPI σαν βάση
    - http://www.owasp.org/index.php/ESAPI
- Έλεγχος των Εφαρμογών
  - Έλεγχος των εφαρμογών από ειδικούς
  - Έλεγχος των εφαρμογών εσωτερικά
    - OWASP Code Review Guide: http://www.owasp.org/index.php/Code_Review_Guide
    - OWASP Testing Guide: http://www.owasp.org/index.php/Testing_Guide

“The vulnerabilities listed at 6.5.1 through 6.5.9 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated the current best practices must be used for these requirements”

PCI DSS 2.0