Excerpts of

Web Application Security

focusing on

Data Validation

adapted for

F.I.S.T. 2004, Frankfurt

by fs

Purpose of this course:

1. Relate to WA’s and get a basic understanding of them
2. Understand attacks on WA’s relating to unvalidated input
3. Learn to implement effective defense accordingly
4. Discuss the subject, Q&A

Introduction

What Are Web Applications?

A WA is a client/server software application that interacts with users or other systems using HTTP.

For a user, the client would most likely be a web browser like Internet Explorer; for another software application this would be an HTTP user agent that acts as an automated browser.

Examples of WA‘s

- eCommerce sites
- Communities, Portals
- Web-based chat sites
- Sites serving personalized content

What are Web Services?

- A WS is a collection of functions that are packaged as a single entity and published to the Internet for use *by other programs*.

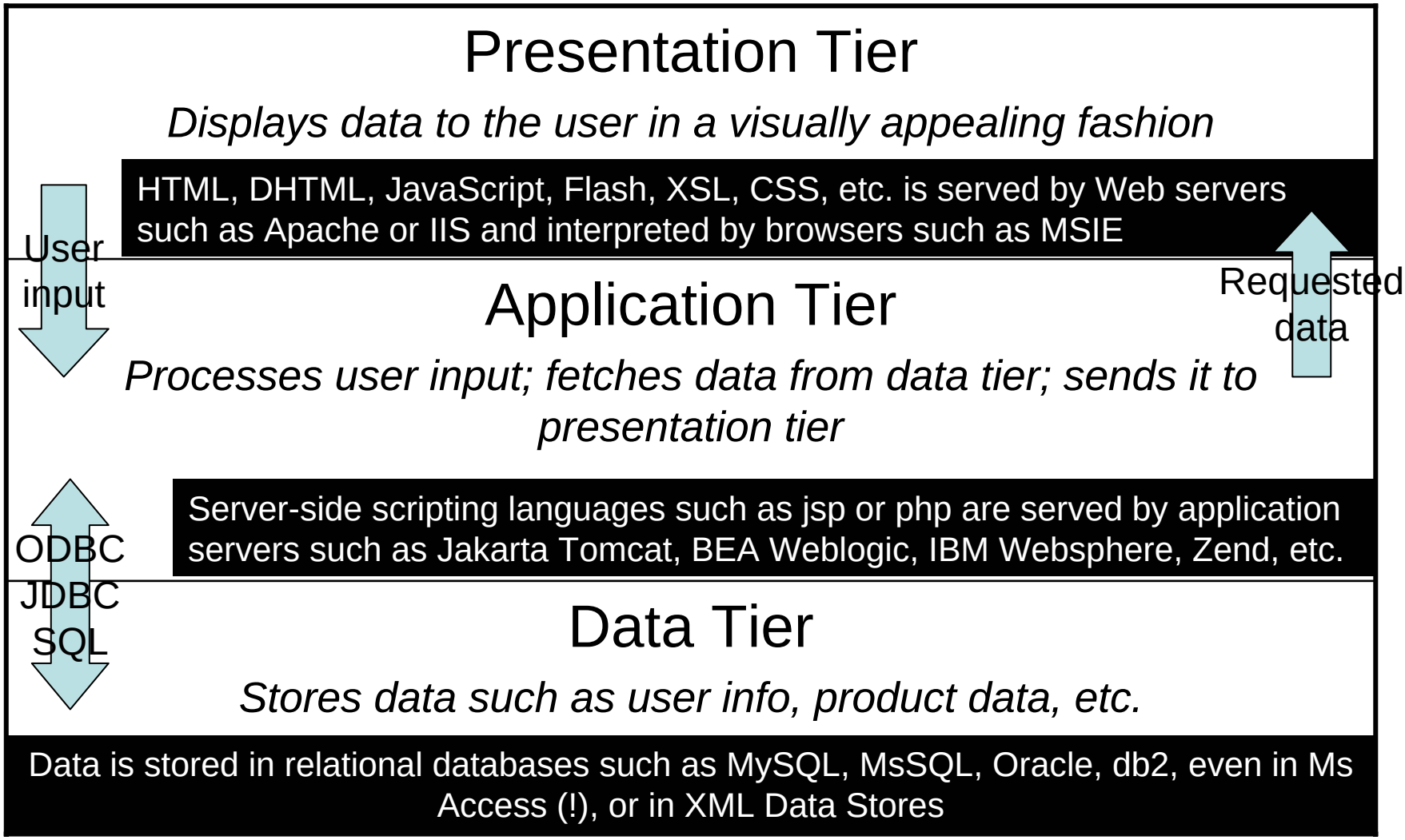
- WS publish their availability using *WSDL* (Web Services Description Language) and *UDDI* (Universal Description, Discovery, and Integration)

- WS are based on XML (extensible Markup Language) and SOAP (Simple Object Access Protocol)

Examples of WS’

- Microsoft Passport
- Project Liberty

Structure & technologies of WA‘s



Top 10 Vulnerabilities in WA‘s*

1. **Unvalidated Input**
2. Broken Access Control
3. Broken Authentication and Session Management
4. XSS
5. Buffer Overflows
6. Injection Flaws
7. Improper Error Handling
8. Insecure Storage
9. DoS
10. Insecure Configuration Management

Source: owasp.org, Jan. 2004

Relevance of WA security

- With the rising popularity of WA‘s, more and more sensitive data migrates to the Web.
- Web Services face the same security issues
- Security professionals - with their traditional focus on network and OS security - tend to neglect WA security,
- yet attacks buried in http requests go past firewalls, filtering, platform hardening, and IDS.
- SSL does not stop many of these attacks either (it only encrypts them)
- Security: a general “hot topic” since 9/11

Objective

From an Attacker's Perspective

- Identify and Exploit Vulnerabilities
- Penetrate corporate applications and compromise the system they're running on
- Use the compromised system for staged attacks

Objective

From Security Professional's Perspective

- Identify all vulnerabilities affecting a WA
- Study the methodologies employed by attackers
- Implement countermeasures to these attacks
- Strengthen default configurations
- Prepare for the unexpected

Agenda...

- Introduction
- Top Ten Web Application Security Vulnerabilities
- ➡ 1) Unvalidated Input
  - **URL Manipulation**
  - Hidden Form Fields Manipulation
  - Cookie Manipulation

URL Manipulation

Overview

- What is a QueryString?
- URL Manipulation
- Safeguards

What is a QueryString?

- Information Appended after URL using ? Mark.

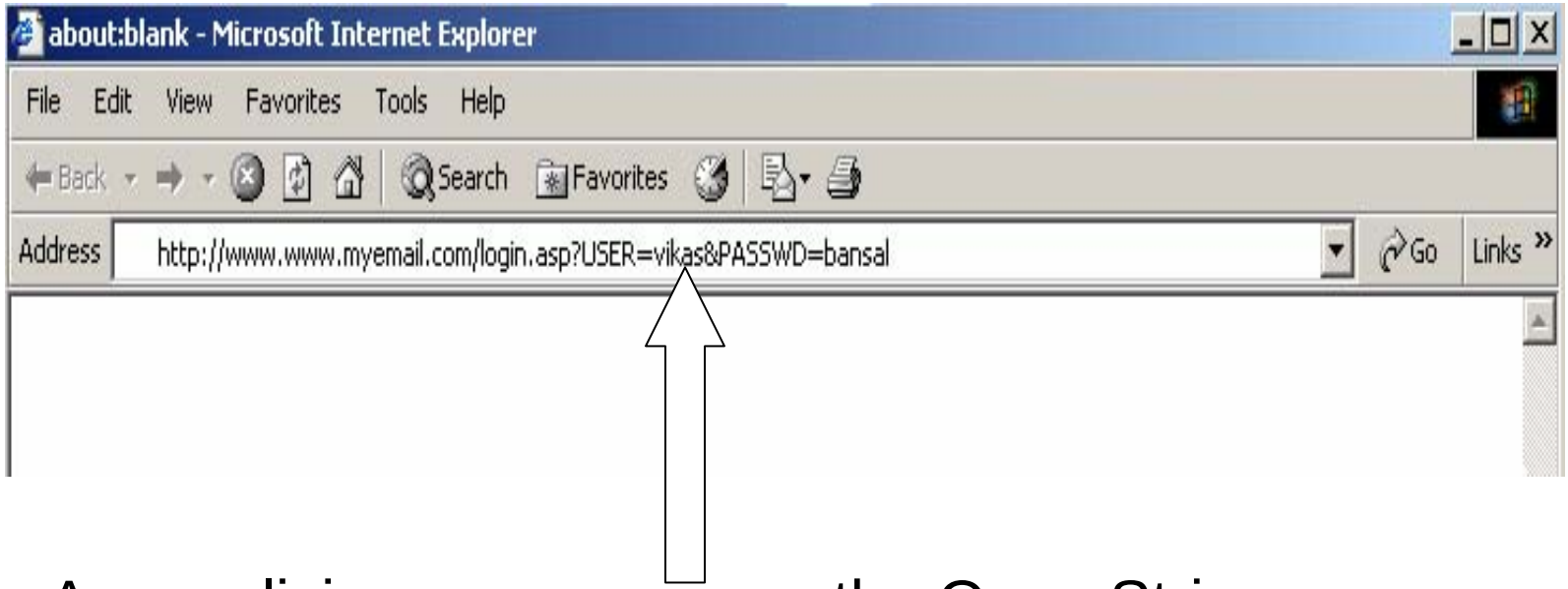
- QueryStrings are used for passing the information across the pages.

- 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

QueryString...



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

URL Manipulation

Using URL manipulation, a malicious user

- could get unauthorised access to user accounts
- could get master access of the database
- could manipulate the database contents
- even could delete the database tables

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

URL Manipulation...

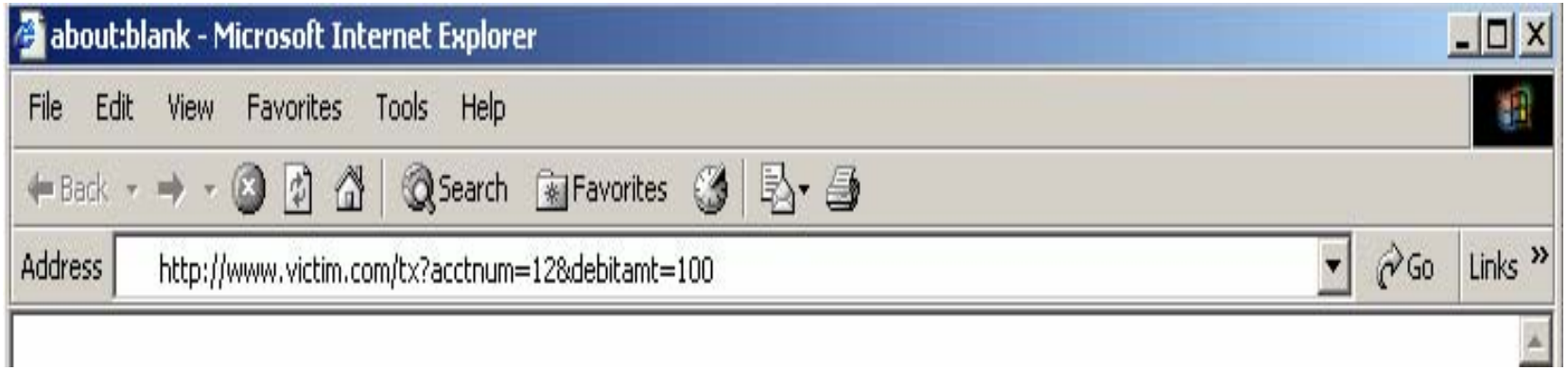
- Example: Changing SQL values
  - UPDATE usertable SET pwd='$INPUT[pwd]' WHERE uid='$INPUT[uid]';
  - Normal input: http://www.victim.com/cpwd?opwd=y&pwd=x&uid=testuser
  - Malicious input: http://www.victim.com/cpwd?opwd=y&pwd=x&uid=testuser'+or+uid+like'%25admin%25';

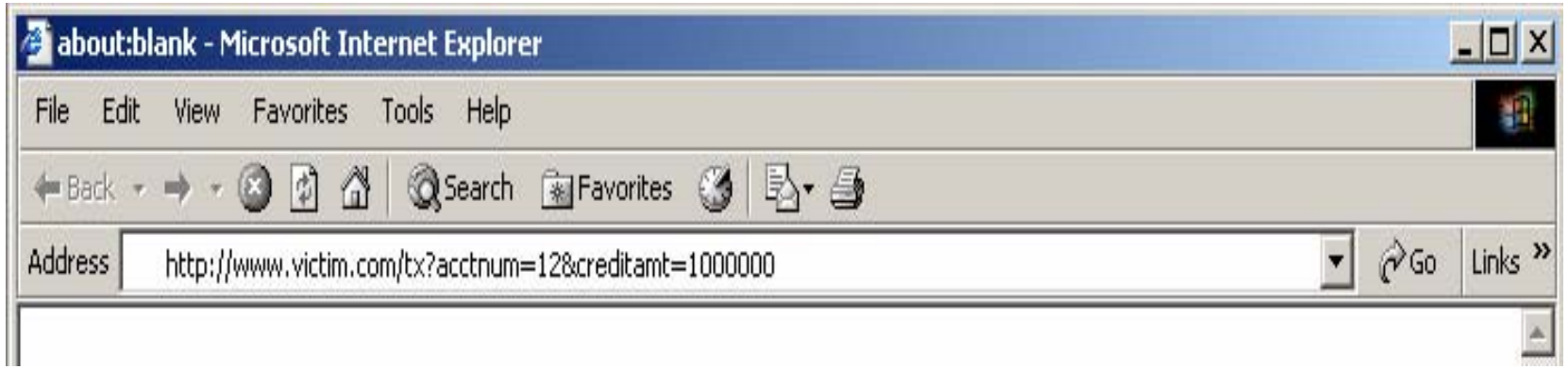
    In URL encoding %25 = %
  - Result: changed Administrator password

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

URL Manipulation...





URL Manipulation…(Safeguards)

- Use Hidden Form Fields to transfer the sensitive Information (also involves security issues)
- Use Cookies to manage session related information
- Use Secure Session Tokens along with the methods above described

Agenda

- Introduction
- Top Ten Web Application Security Vulnerabilities
  - Unvalidated Parameters
    - URL Manipulation
    - **Hidden Form Fields Manipulation**
    - Cookie Manipulation

Hidden Form Fields

Overview

- What are Hidden Form Fields?
- How can they be defined?
- Use of Hidden Form Fields
- Hidden Form Field Manipulation
- Safeguards

What are Hidden Form Fields?

- A special type of form field defined as “hidden”
- Have a name/value pair
- The value can be changed by Proxy apps like Achilles or Websleuth or simply by saving the page and editing it
- Used to:
  - transmit persistent data between two or more pages
  - Ex. E-mail support WA that attaches userAgent info to a support request.

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.

Use of Hidden Form Fields

Advantages & Disadvantages

- Advantages:
  - Requires little knowledge and is implemented quickly
- Disadvantages:
  - Not kept across sessions, so useless for maintaining persistent information about a user
  - Are easily manipulated on the client-side

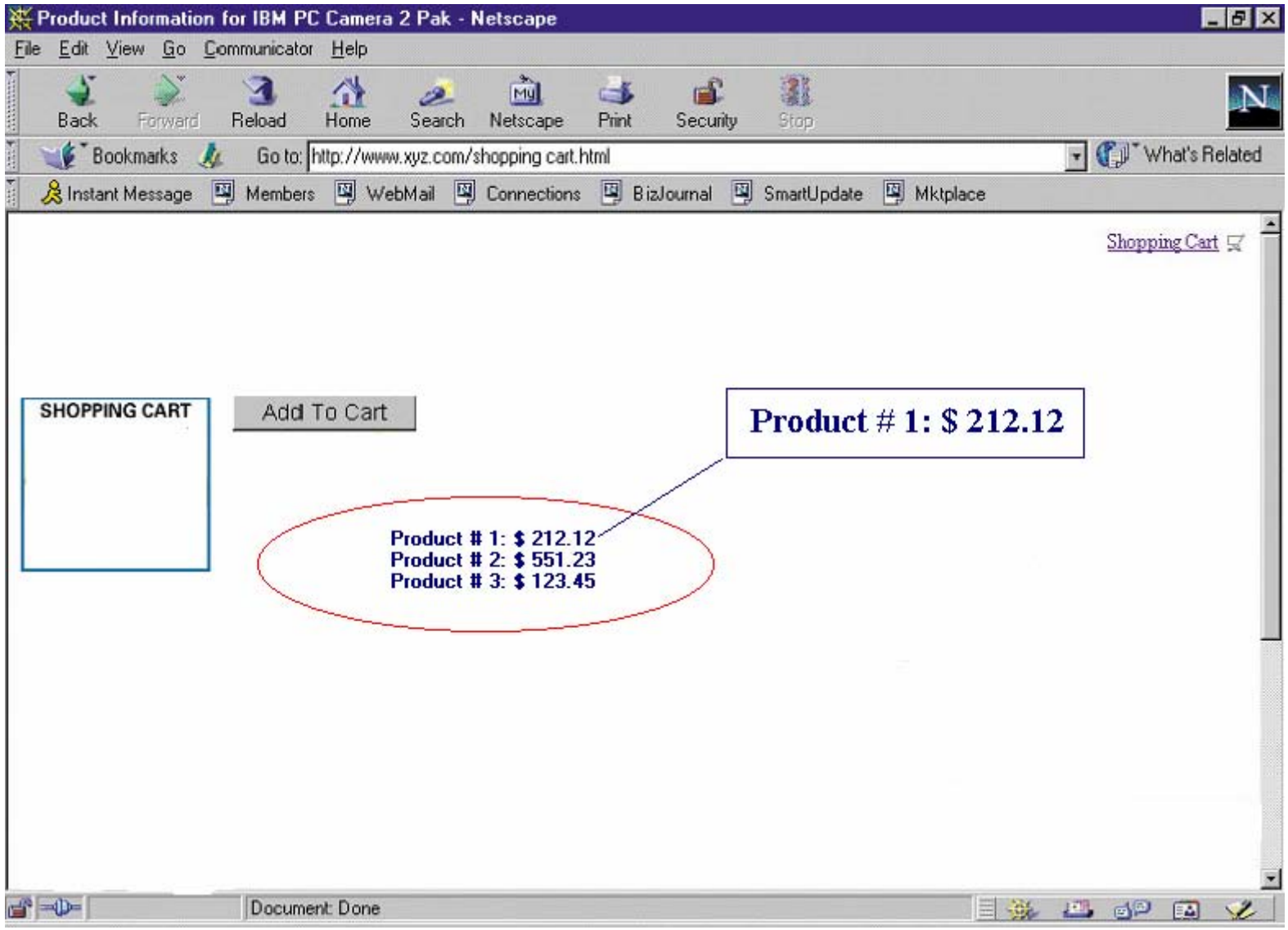
Hidden Form Field Manipulation...

- Find vulnerable form with hidden input
- Save HTML file to disk, modify hidden input
- Execute HTML file from disk, submit form
- Check the results, i.e. for ‘interesting’ error messages
- Tools like Achilles or WebSleuth can facilitate the task

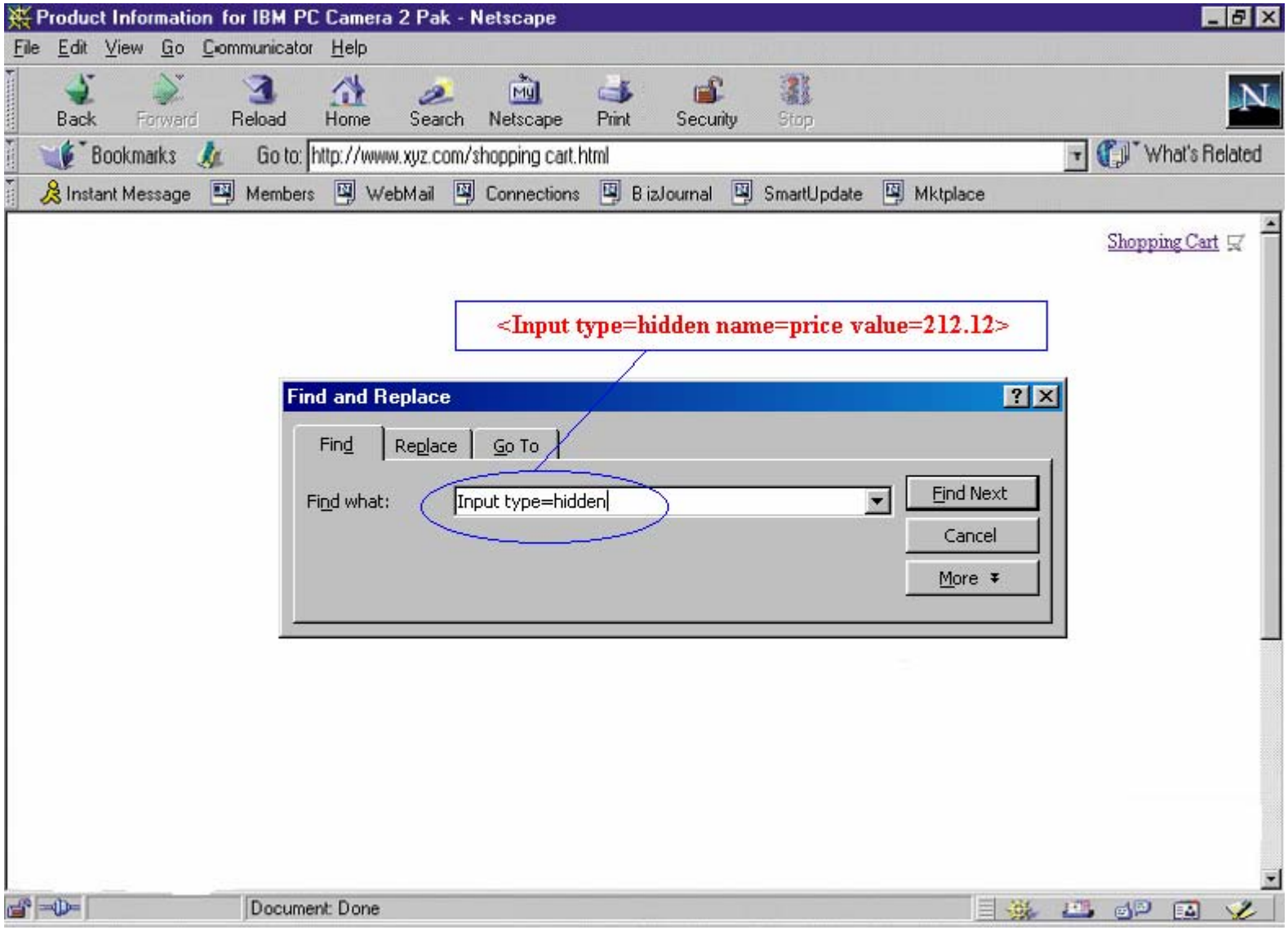
Hidden Form Field Manipulation...

- 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 form field manipulation

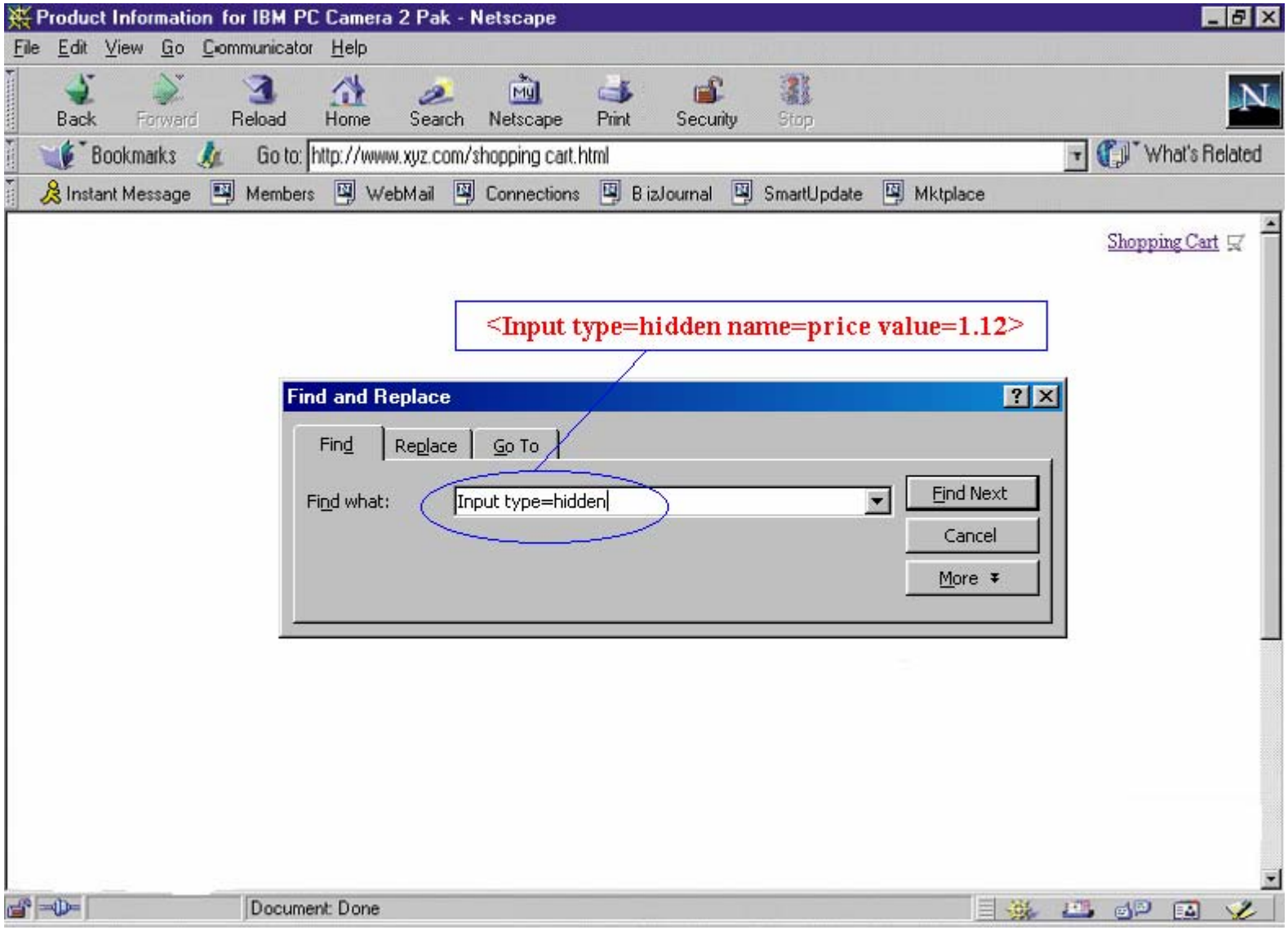
Hidden Form Field Manipulation...



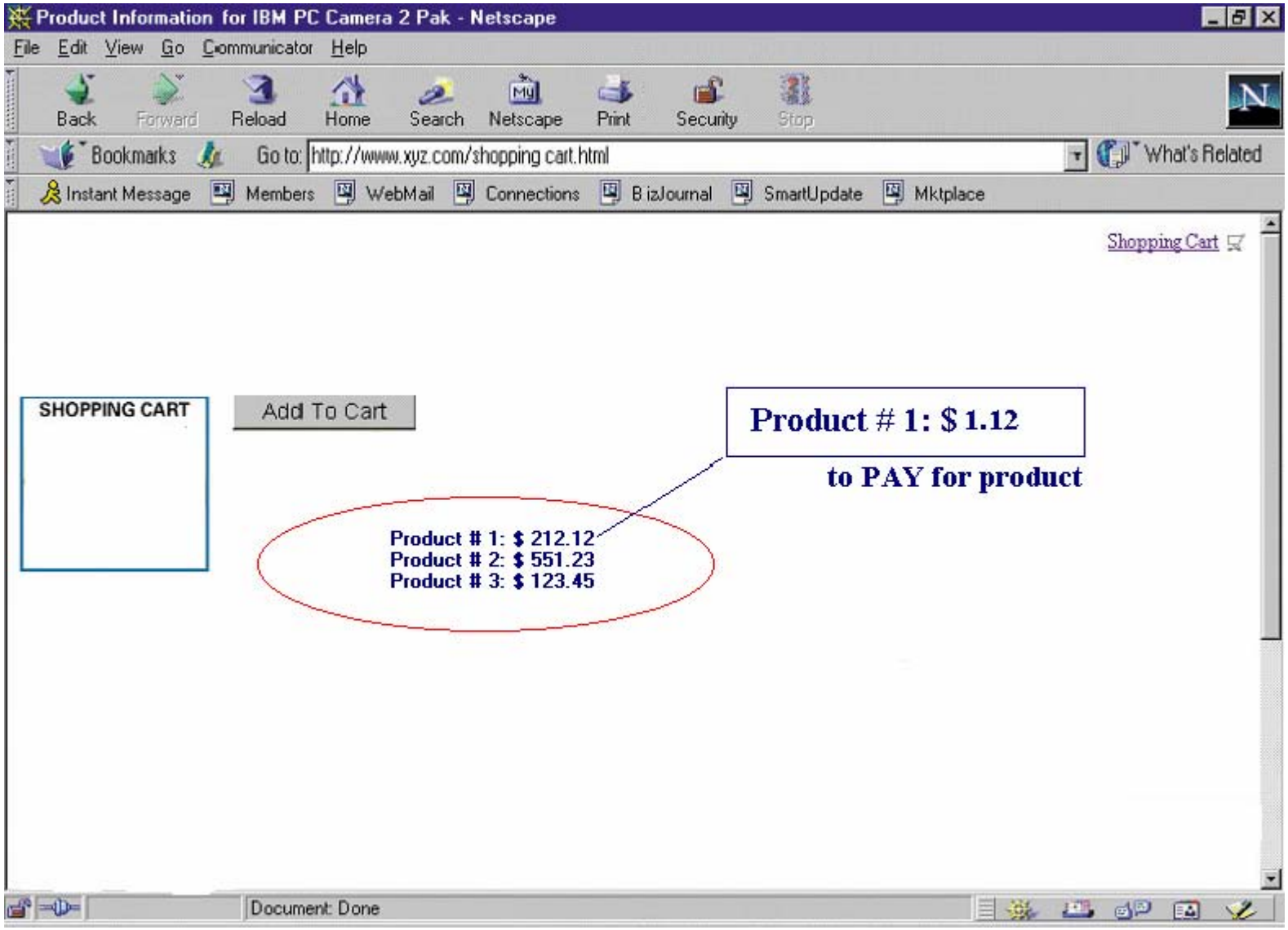
Hidden Form Fields Manipulation...



Hidden Form Fields Manipulation...



Hidden Form Fields Manipulation...



Safeguards

- Never trust hidden Input values
- Never allow unsanitized (unchecked) input to be processed at the SERVER directly.
- In this case, just validate the price with the price stored in the database or in an XML data store.
- Evaluate whether it makes sense to store a given info in an HFF at all

Agenda

- Introduction
- Top Ten Web Application Security Vulnerabilities

  1) Unvalidated Input
     - URL Manipulation
     - Hidden Form Fields Manipulation
     - **Cookie Manipulation**

- Conclusion

Cookie Manipulation

Overview

- What is a Cookie?
- Basic Elements of a Cookie
- Uses of Cookies
- A typical Cookie Algorithm
- Cookie Manipulation
- Cookies - Privacy Concerns

What is a Cookie?

- A Cookie is a 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 only by the browser which wrote it
- Developed for user convenience to allow customisation of sites without need for repeating preferences

Basic Elements of a Cookie

Domain	flag	path	secure	expiry	name	value

- Cookies have 7 key attributes: Domain, flag, path, secure, expiration, Name, Value.
- A Cookie cannot exceed 4 KB in size
- Cookies are of two types
  - Persistent: residing on the client's hard drive for a specifiable period of time
  - Non-Persistent: specific to the session and deleted as soon as it ends (when all instances of the browser are closed)

Uses of Cookies

For legitimate users:

- Store any information explicitly provided to a site so it does not have to be typed again
- Only needs to authenticate once

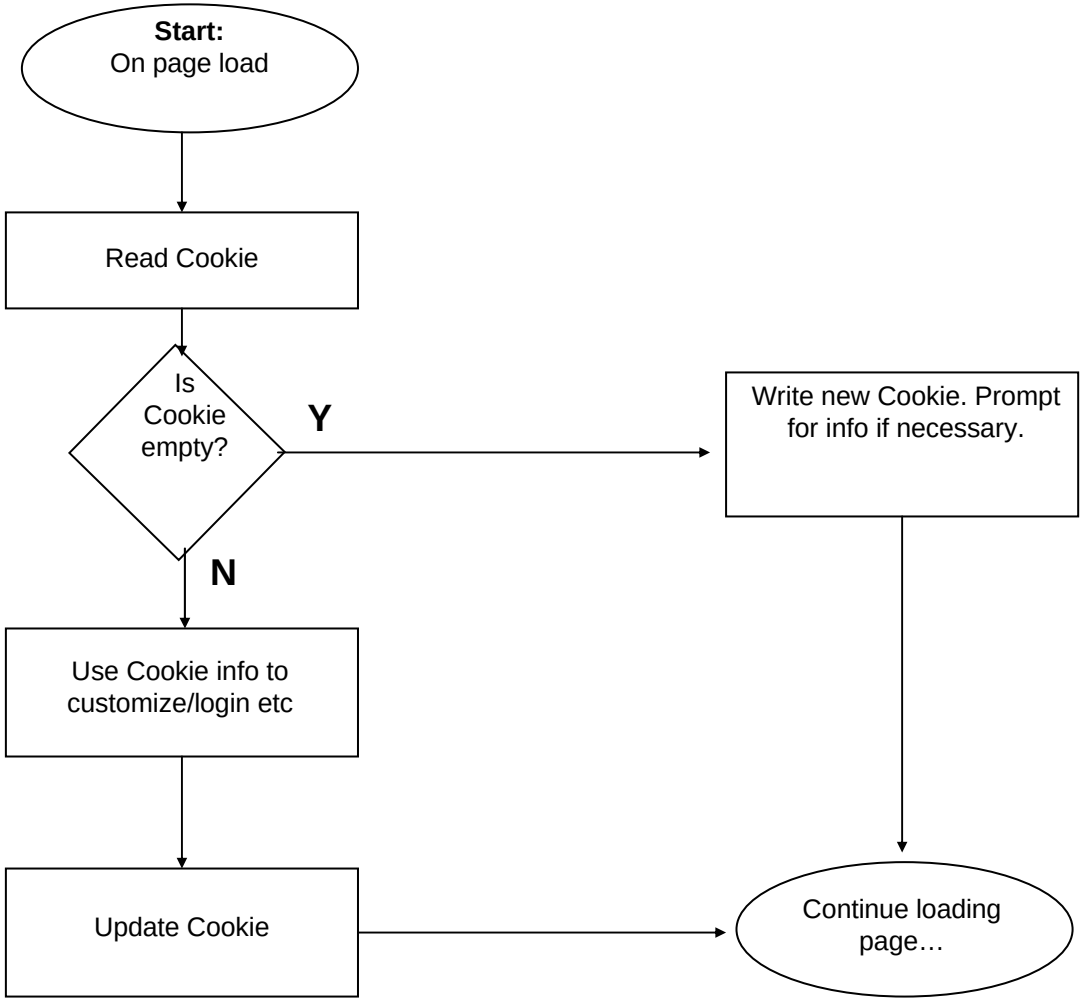
For webmasters

- Get more detailed user statistics
- Possibility to “recognize” users (1on1 Marketing)

For a cookie-stealing attacker

- Use any information typically included in cookies, i.e. IP Address, Operating System, Browser type, Session ID, etc. for malicious purposes

A typical Cookie algorithm

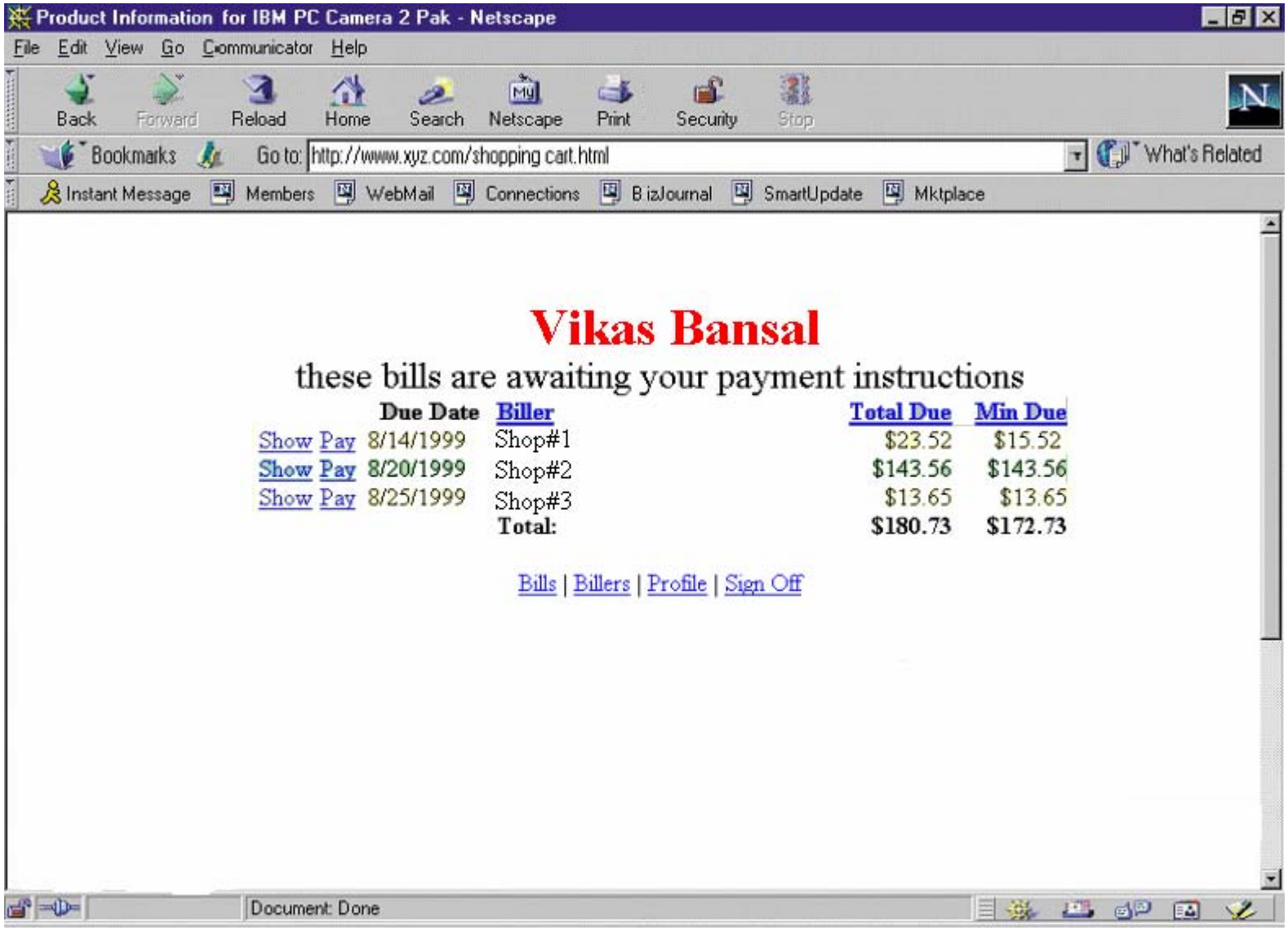


Cookie Manipulation

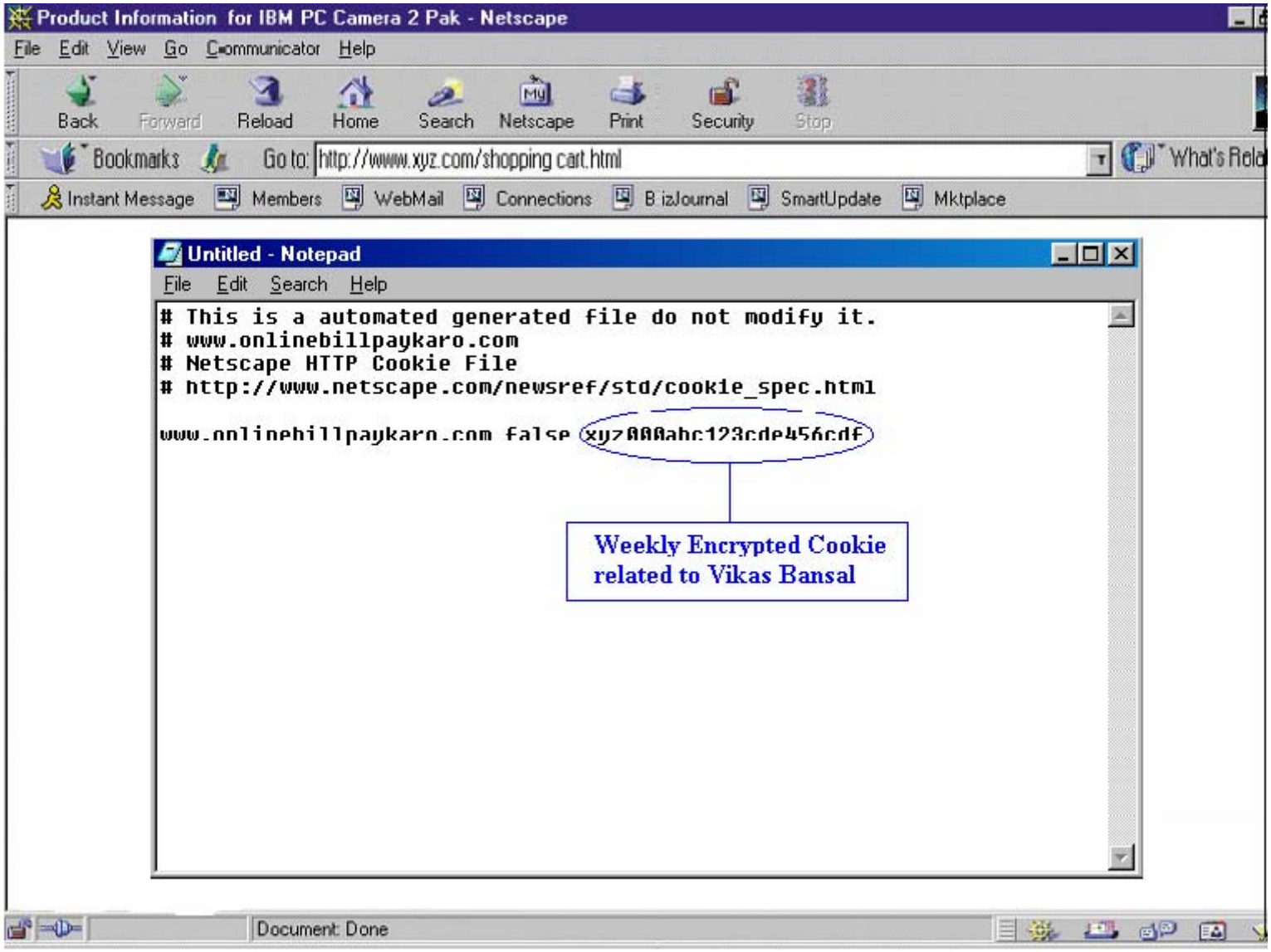
- If Cookies are not securely encoded, a hacker can interpret its values and modify them
- Example:

  “Poisoning” the cookie (Userid and timestamp)
- Risks: Bypassing authentication, gain access to accounts and information of other users.

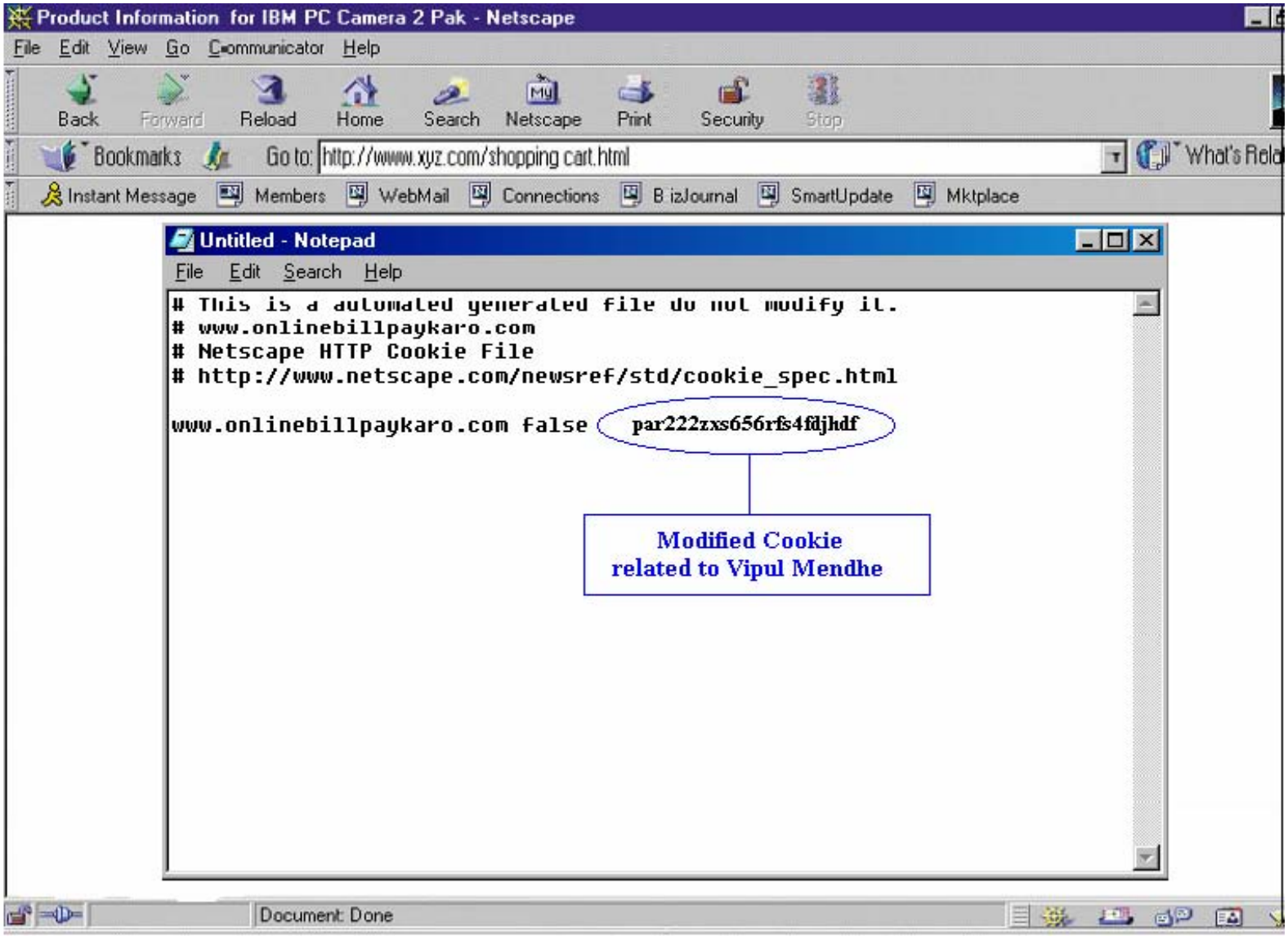
Example


Product Information for IBM PC Camera 2 Pak - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://www.xyz.com/shopping cart.html What's Related
Instant Message Members WebMail Connections BizJournal SmartUpdate Mktplace
Vikas Bansal
these bills are awaiting your payment instructions
Due Date Biller Total Due Min Due
Show Pay 8/14/1999 Shop#1 $23.52 $15.52
Show Pay 8/20/1999 Shop#2 $143.56 $143.56
Show Pay 8/25/1999 Shop#3 $13.65 $13.65
Total: $180.73 $172.73
Bills | Billers | Profile | Sign Off
Document: Done

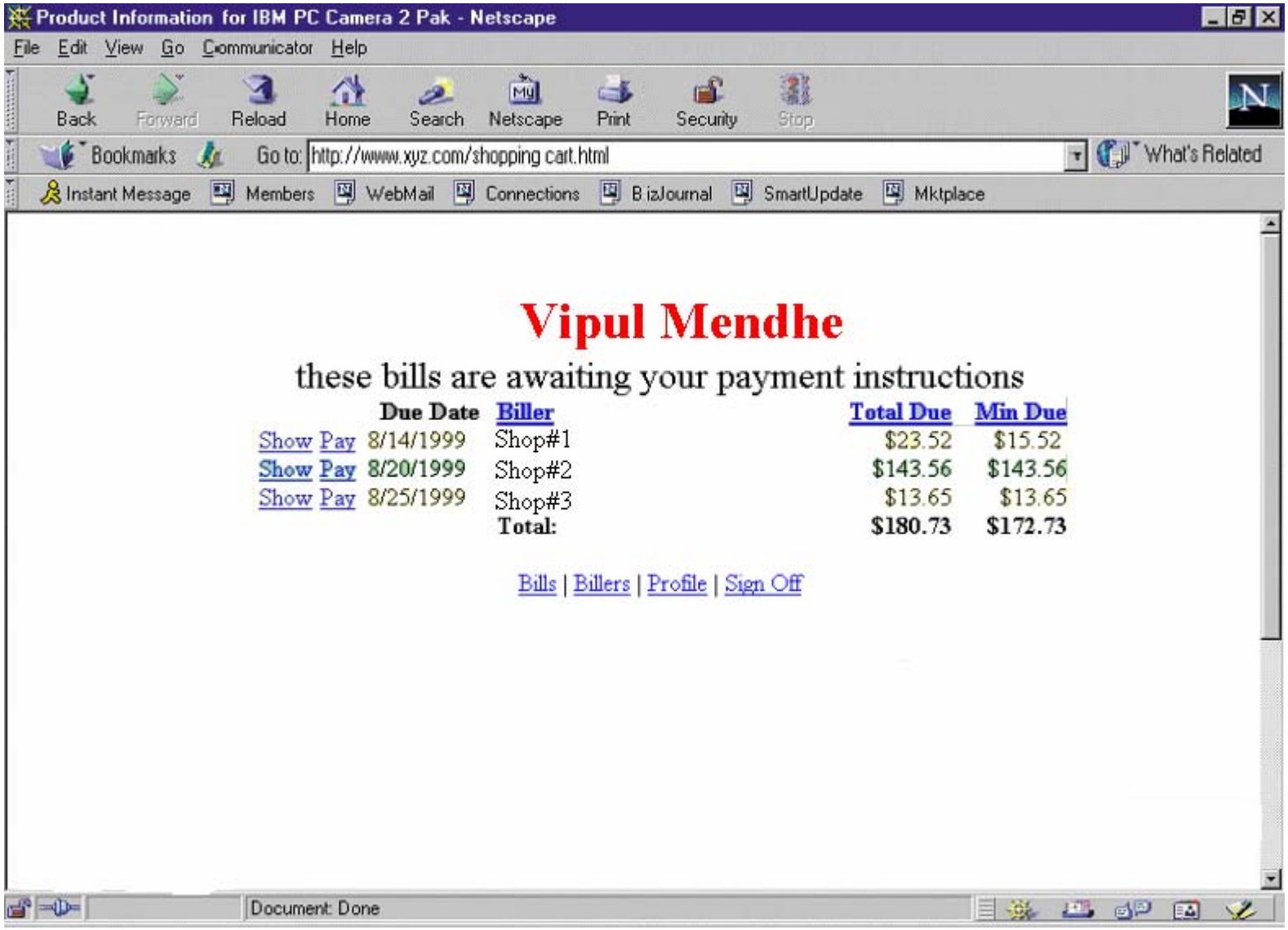
Example


Product Information for IBM PC Camera 2 Pak - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://www.xyz.com/shopping cart.html
Instant Message Members WebMail Connections BizJournal SmartUpdate Mktplace
Untitled - Notepad
File Edit Search Help
This is a automated generated file do not modify it.
www.onlinebillpaykaro.com
Netscape HTTP Cookie File
http://www.netscape.com/newsref/std/cookie_spec.html
www.onlinebillpaykaro.com false xyz000abc123cde456cdf
Weekly Encrypted Cookie related to Vikas Bansal
Document: Done

Example


Product Information for IBM PC Camera 2 Pak - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://www.xyz.com/shopping cart.html
Instant Message Members WebMail Connections B izJournal SmartUpdate Mktplace
Untitled - Notepad
File Edit Search Help
This is a automated generated file do not modify it.
www.onlinebillpaykaro.com
Netscape HTTP Cookie File
http://www.netscape.com/newsref/std/cookie_spec.html
www.onlinebillpaykaro.com false par222zxs656rfs4fdjhdf
Modified Cookie related to Vipul Mendhe
Document: Done

Example


Product Information for IBM PC Camera 2 Pak - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://www.xyz.com/shopping cart.html What's Related
Instant Message Members WebMail Connections BizJournal SmartUpdate Mktplace
Vipul Mendhe
these bills are awaiting your payment instructions
Due Date Biller Total Due Min Due
Show Pay 8/14/1999 Shop#1 $23.52 $15.52
Show Pay 8/20/1999 Shop#2 $143.56 $143.56
Show Pay 8/25/1999 Shop#3 $13.65 $13.65
Total: $180.73 $172.73
Bills | Billers | Profile | Sign Off
Document: Done

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 browsing habits, which could be sold to third parties for profit

- Non-coded name-value pairs can result 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

Agenda

- Introduction
- Top Ten Web Application Security Vulnerabilities

  1) Unvalidated Parameters
     - URL Manipulation
     - Hidden Form Fields Manipulation
     - Cookie Manipulation

- **<u>Conclusive Thoughts</u>**

Conclusive Thoughts

The ideal validation strategy?

- Accept Only Known Valid Data or
- Reject Known Bad Data or
- Sanitize Bad Data

?

What to validate?

- Data Type or
- Syntax or
- Length

?

Where to validate?

- On the client-side
- On the server-side

?

Thank You