Securitatea Web

Ciprian Dobre
ciprian.dobre@cs.pub.ro

Web Security Programming I

Building Security in from the Start

A Simple Web Server

To illustrate what can go wrong if we do not design for security in our web applications from the start, consider a simple web server implemented in Java.

All this program does is serve documents using HTTP.

We will walkthrough the code in the following slides.

Some Preliminaries…

- (**H**yper**T**ext **T**ransfer **P**rotocol): The communications protocol used to connect to servers on the Web.
- Its primary function is to establish a connection with a Web server and transmit HTML pages to the client browser or any other files required by an HTTP application.
- Addresses of Web sites begin with an **http://** prefix.

Some Preliminaries…

- A typical HTTP request that a browser makes to a web server:

  ```
  Get / HTTP/1.0
  ```

- When the server receives this request for filename / (which means the *root* document on the web server), it attempts to load index.html. It sends back:

  ```
  HTTP/1.0 200 OK
  ```

  followed by the document contents.

SimpleWebServer: main()

```
 /* This method is called when the program is run from
   the command line. */

public static void main (String argv[]) throws Exception
   {
   /* Create a SimpleWebServer object, and run it */
   SimpleWebServer sws = new SimpleWebServer();
   sws.run();

}
```

SimpleWebServer Object

```
public class SimpleWebServer {

    /* Run the HTTP server on this TCP port. */
    private static final int PORT = 8080;

    /* The socket used to process incoming connections
       from web clients */
    private static ServerSocket dServerSocket;

    public SimpleWebServer () throws Exception {
      dServerSocket = new ServerSocket (PORT);
    }

    public void run() throws Exception {
      while (true) {
        /* wait for a connection from a client */
        Socket s = dServerSocket.accept();

        /* then process the client's request */
        processRequest(s);
    }
}
```

SimpleWebServer: processRequest 1

```
/* Reads the HTTP request from the client, and
   responds with the file the user requested or
   a HTTP error code. */

 public void processRequest(Socket s) throws Exception {

 /* used to read data from the client */
 BufferedReader br =
     new BufferedReader (new InputStreamReader (s.getInputStream()));

 /* used to write data to the client */
 OutputStreamWriter osw =
     new OutputStreamWriter (s.getOutputStream());

 /* read the HTTP request from the client */
 String request = br.readLine();

 String command = null;
 String pathname = null;
```

SimpleWebServer: processRequest 2

```
/* parse the HTTP request */
StringTokenizer st =
    new StringTokenizer (request, " ");

command = st.nextToken();
pathname = st.nextToken();

if (command.equals("GET")) {
    /* if the request is a GET
       try to respond with the file
       the user is requesting */
    serveFile (osw,pathname);
}
else {
    /* if the request is a NOT a GET,
       return an error saying this server
       does not implement the requested command */
    osw.write ("HTTP/1.0 501 Not Implemented\n\n");
}

/* close the connection to the client */
osw.close();
```

SimpleWebServer: serveFile 1

```
public void serveFile (OutputStreamWriter osw,
          String pathname) throws Exception {
  FileReader fr=null;
  int c=-1;
  StringBuffer sb = new StringBuffer();

  /* remove the initial slash at the beginning
     of the pathname in the request */
  if (pathname.charAt(0)=='/')
      pathname=pathname.substring(1);

  /* if there was no filename specified by the
     client, serve the "index.html" file */
  if (pathname.equals(""))
      pathname="index.html";
```

SimpleWebServer: serveFile 2

```
/* try to open file specified by pathname */
  try {
      fr = new FileReader (pathname);
      c = fr.read();
  }
  catch (Exception e) {
      /* if the file is not found,return the
         appropriate HTTP response code  */
      osw.write ("HTTP/1.0 404 Not Found\n\n");
      return;
  }
```

SimpleWebServer: serveFile 3

```
/* if the requested file can be
successfully opened and read, then
return an OK response code and send
the contents of the file */
osw.write ("HTTP/1.0 200 OK\n\n");
while (c != -1) {
    sb.append((char)c);
    c = fr.read();
}
osw.write (sb.toString());
```

Quiz

Can you identify any security vulnerabilities in SimpleWebServer?

What Can Go Wrong?

Denial of Service (DoS):

- An attacker makes a web server unavailable.
- Example: an online bookstore’s web server crashes and the bookstore loses revenue

DoS on SimpleWebServer?

Just send a carriage return as the first message instead of a properly formatted GET message…

DoS on SimpleWebServer?

processRequest():

```
/* read the HTTP request from the client */
  String request = br.readLine();

  String command = null;
  String pathname = null;

/* parse the HTTP request */
  StringTokenizer st =
      new StringTokenizer (request, " ");

  command = st.nextToken();
  pathname = st.nextToken();
```

DoS on SimpleWebServer?

- The web server crashes
- Service to all subsequent clients is denied until the web server is restarted

How Do We Fix This?

- *The web server should immediately disconnect from any web client that sends a malformed HTTP request to the server.*
- The programmer needs to carefully handle exceptions to deal with malformed requests.

How would you fix this code?

processRequest():

```
/* read the HTTP request from the client */
  String request = br.readLine();

  String command = null;
  String pathname = null;

/* parse the HTTP request */
  StringTokenizer st =
      new StringTokenizer (request, " ");

  command = st.nextToken();
  pathname = st.nextToken();
```

A possible solution

```
/* read the HTTP request from the client */
   String request = br.readLine();
   String command = null;
   String pathname = null;

try {
/* parse the HTTP request */
   StringTokenizer st =
       new StringTokenizer (request, " ");
   command = st.nextToken();
   pathname = st.nextToken();
} catch (Exception e) {
   osw.write (“HTTP/1.0 400 Bad Request\n\n”);
   osw.close();
   return;
}
```

Importance of “Careful” Exception Handling

- Error messages and observable behavior can tip off an attacker to vulnerabilities
- Fault Injection: Providing a program with input that it does not expect (as in the CR for SimpleWebServer) and observing behavior

Careful Exception Handling

- Two possible designs for
  `int checkPassword (String username, String password)`

- The function could fail, so what exception should the function return?

1)
```
ERROR_ACCESS_DENIED
ERROR_PASS_FILE_NOT_FOUND
ERROR_OUT_OF_MEMORY
NO_ERROR_ACCESS_ALLOWED
```

2)
```
NO_ERROR
ERROR
int getError ()
```

Be careful to not provide more information to a user than is needed.

Careful Exception Handling

```
int result = checkPassword ( … )
  if (result == ERROR_ACCESS_DENIED) {
      abort();
  }
  else {
      // Complete login
}
```

- Problem: result != ERROR_ACCESS_DENIED does not infer ERROR_ACCESS_ALLOWED
- Result could have been: ERROR_PASS_FILE_NOT_FOUND or ERROR_OUT_OF_MEMORY !

Fail-Safe

```
int result = checkPassword ( … )
  if (result == NO_ERROR) {
      // Complete login
  }
  else {
      int reason = getError();
      abort();
}
```

- Much better– less error prone!
- checkPassword failure occurs securely!

Sources

- The content of these slides was adapted from:

- "Foundations of Security: What Every Programmer Needs To Know" (ISBN 1590597842) by Neil Daswani, Christoph Kern, and Anita Kesavan.

- http://www.learnsecurity.com/ntk

Security Design Principles

Security Design Principles

- Least Privilege
- Defense in Depth
- Secure Weakest Link
- Fail-safe Stance
- Secure By Default
- Simplicity
- Usability

Principle of Least Privilege

- Just enough authority to get the job done.
- Common world example: Valet Keys
- A web server should only be given access to the set of HTML files that the web server is to serve.

SimpleWebServer and “Elevated Privileges”

- Suppose a system administrator were to run SimpleWebServer under the root account
- When clients access the web server, they can access all the files on the system!
- Maybe we can control this by not storing sensitive documents in the web server’s directory tree…

What about this?

```
GET ../../../../etc/shadow HTTP/1.0
```

Defense in Depth

- Also called redundancy / diversity
- Common world example: Banks
- Passwords:
  - Require users to choose “strong” passwords
  - Monitor web server logs for failed login attempts

Secure the Weakest Link

- Common Weak Links:
  - Unsecured Dial-In Hosts; War Dialers
  - Weak Passwords; Crack
  - People; Social Engineering Attacks
  - Buffer Overflows

Fail-Safe Stance

- Common world example: Elevators
- System failure should be expected (and planned for)
  - If firewall fails, let no traffic in
  - Deny access by default

SimpleWebServer and Fail-Safe

- serveFile()

```
/* if the requested file can be successfully opened
  and read, then return an OK response code and send
  the contents of the file */
  osw.write ("HTTP/1.0 200 OK\n\n");
  while (c != -1) {
      sb.append((char)c);
      c = fr.read();
  }
  osw.write (sb.toString());
```

An “Infinite” File

- The Linux /dev/random is a file that returns random bits (often used to generate cryptographic keys)
- *It can be used as a source of infinite data..*

- What happens when the web server receives:

```
GET //dev/random HTTP/1.0
```

How Can We Fix This?

```
/* if the requested file can be
  successfully opened and read, then
  return an OK response code and send
  the contents of the file */
  osw.write ("HTTP/1.0 200 OK\n\n");
  while (c != -1) {
      sb.append((char)c);
      c = fr.read();
  }
  osw.write (sb.toString());
```

Secure By Default

- Only enable the 20% of the products features that are used by 80% of the user population.
- “Hardening” a system:
  All unnecessary services off by default
- More features enabled ->
  more potential exploits ->
  less security!

Simplicity

- Complex software is likely to have security holes (i.e. sendmail).
- Use choke points – keep security checks localized.
- Less functionality =
  Less security exposure

Usability

- Users typically do not read documentation (Therefore: Enable security by default)
- Users can be lazy
  (Assume: They ignore security dialogs)
- **Secure by default features in software forces users and vendors to be secure.**

Security Features Do Not Imply Security

- Using one or more security algorithms/protocols will not solve all your problems!
  - Using encryption doesn’t protect against weak passwords.
  - Using SSL in SimpleWebServer doesn’t protect against DoS attacks, access to /etc/shadow, etc.

Security Features Do Not Imply Security

- Security features may be able to protect against specific threats

- But if the software has bugs, is unreliable, does not cover all possible corner cases:

  The system may not be secure despite the security features it has

“Good Enough” Security

- The fraction of time you spend designing for security in your application should be proportional to the number and types of threats that your software and business face
- But remember: Customers *expect* privacy and security

“Good Enough” Security

Design for security by incorporating “hooks” and other low-effort functionality from the beginning. This way, you can add more security as needed without having to resort to work-arounds.

And Don’t Reinvent the Wheel!

- SimpleWebServer has many security vulnerabilities…
- Building a secure, high-performance web server is a very challenging task
- Apache: www.apache.org

Source

- The content of these slides was adapted from:
- "Foundations of Security: What Every Programmer Needs To Know" (ISBN 1590597842) by Neil Daswani, Christoph Kern, and Anita Kesavan.
- http://www.learnsecurity.com/ntk

Google Hacking and Web Hacking

Happy Anniversary !

- Search Engine Hacking - First solid documentation: SimpleNomad, 1996, AltaVista textfiles.com

Web Hacking: Pick a site, find the vulnerability

Google Hacking : Pick a vulnerability, find the site.

Don’t Be A Target of Opportunity

Just the beginning ...

- Non-Public Systems
  - Intranets, access-restricted extranets, web services
- Not all internet systems crawled
  - Have to request a crawl
  - Extranets, customer portals
- Google: very limited crawl
  - Robots.txt, forms, javascript
  - Linked content only !
- Exposure has to be hard-linked
  - No tampering

The Perfect Drug

Warning ! Search engine hacking can be highly addictive

Focus on what to look for, not on the search engine.

A Few of my Favorite Things

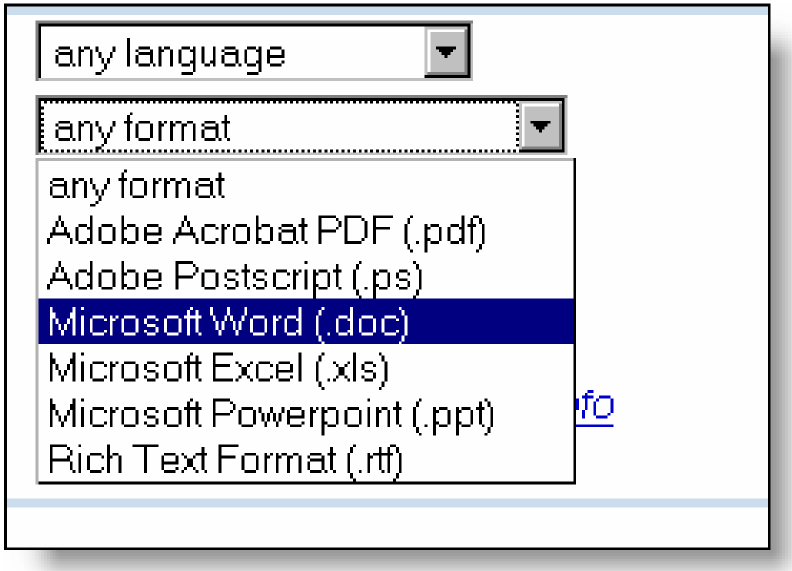
Source code galore: Need a code sample ? Grab a code sample !

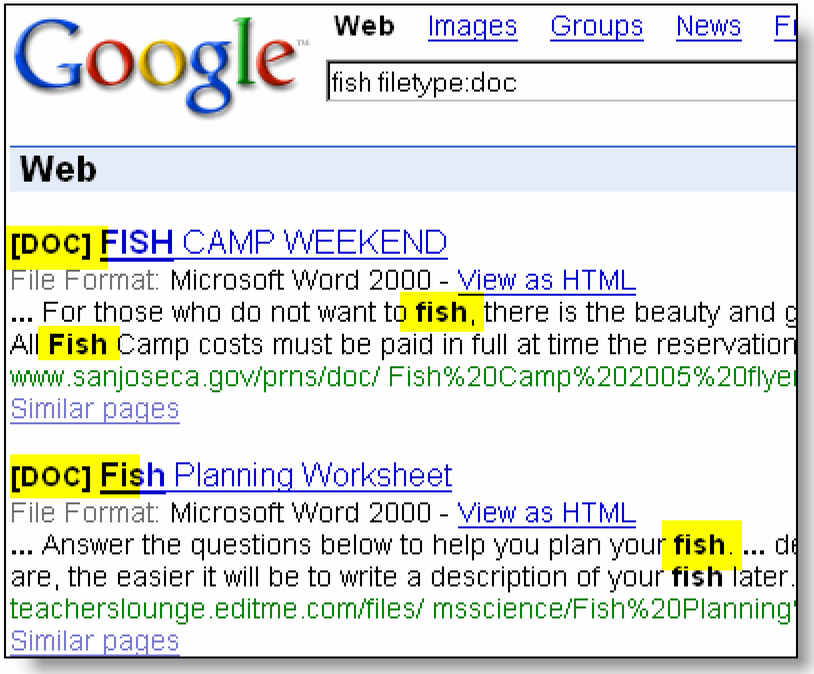
File traversals : full system read access

Command Execution : Executing shell commands through a browser, basically port 80 telnet.

File Uploads: Don’t like the content ? Make your own !

Basic Google Hacking - Using File Types





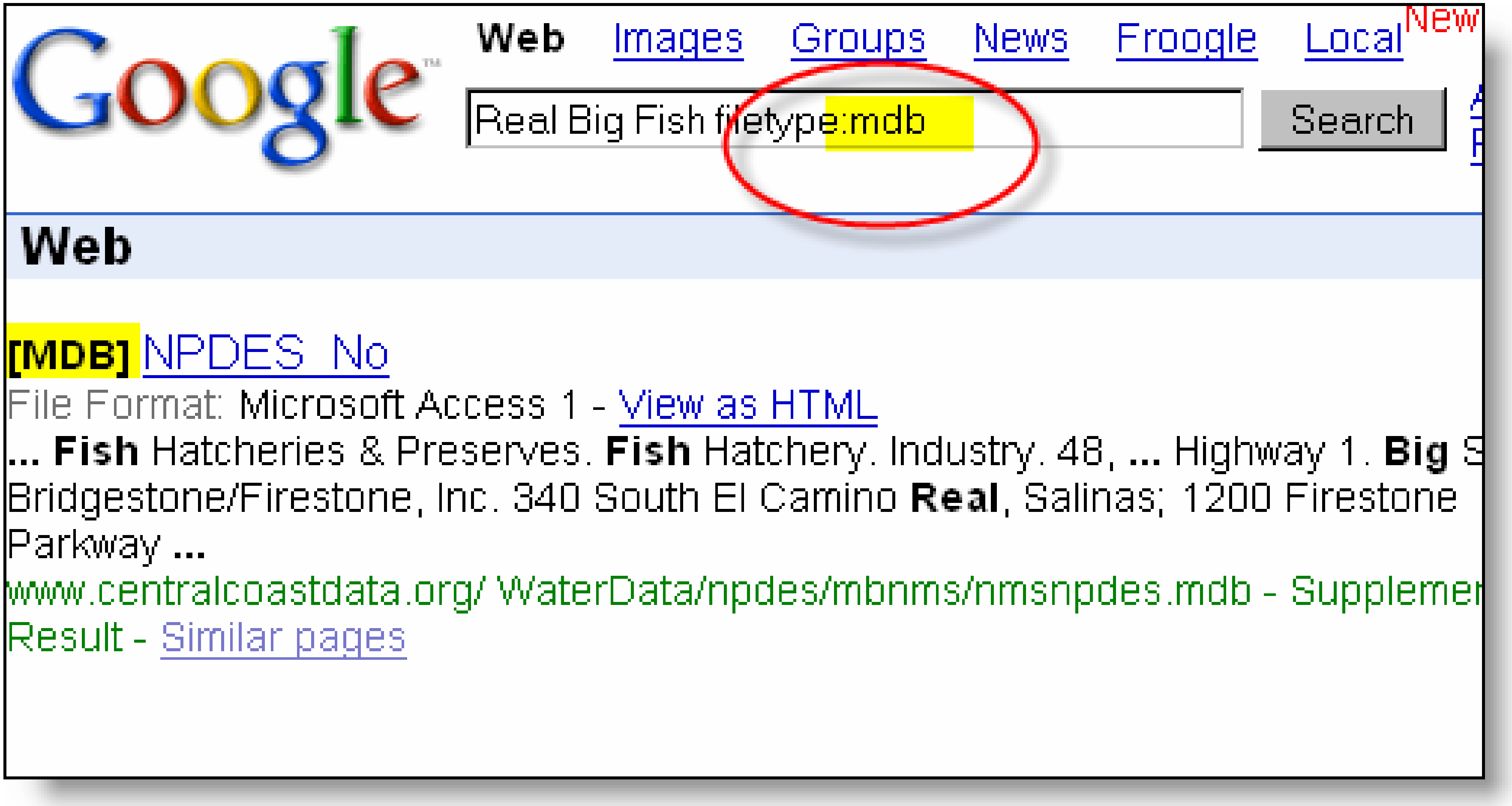
fish filetype:doc

Web

[DOC] FISH CAMP WEEKEND
File Format: Microsoft Word 2000 - View as HTML
... For those who do not want to **fish**, there is the beauty and g
All **Fish** Camp costs must be paid in full at time the reservation
www.sanjoseca.gov/prns/doc/ Fish%20Camp%202005%20flye
Similar pages

[DOC] Fish Planning Worksheet
File Format: Microsoft Word 2000 - View as HTML
... Answer the questions below to help you plan your **fish**. ... de
are, the easier it will be to write a description of your **fish** later.
teacherslounge.editme.com/files/ msscience/Fish%20Planning
Similar pages

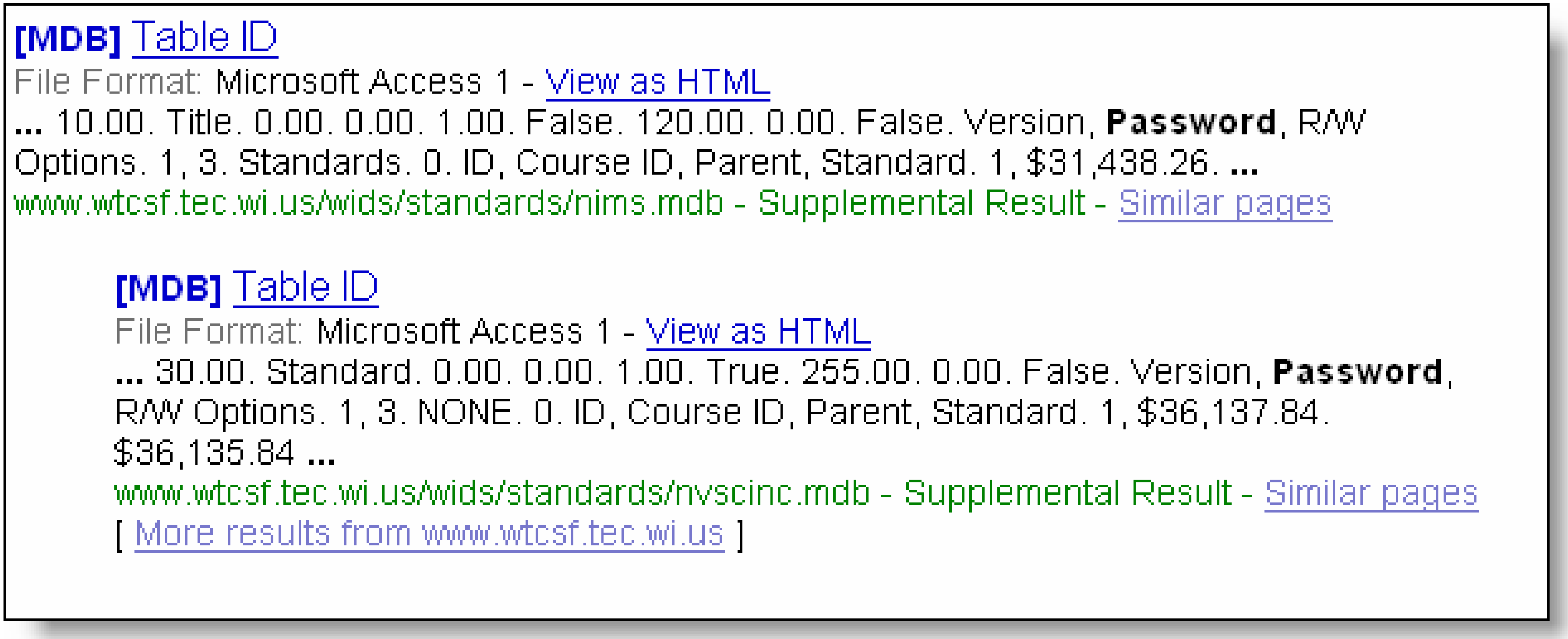
Works for many other file types



Real Big Fish filetype:mdb

[MDB] NPDES_No
File Format: Microsoft Access 1 - View as HTML
... **Fish** Hatcheries & Preserves. **Fish** Hatchery. Industry. 48, ... Highway 1. **Big** S
Bridgestone/Firestone, Inc. 340 South El Camino **Real**, Salinas; 1200 Firestone
Parkway ...
www.centralcoastdata.org/ WaterData/npdes/mbnms/nmsnpdes.mdb - Supplemen
Result - Similar pages

Curioser and Curioser



[MDB] Table ID
File Format: Microsoft Access 1 - View as HTML
... 10.00. Title. 0.00. 0.00. 1.00. False. 120.00. 0.00. False. Version, **Password**, R/W Options. 1, 3. Standards. 0. ID, Course ID, Parent, Standard. 1, $31,438.26. **...**
www.wtcsf.tec.wi.us/wids/standards/nims.mdb - Supplemental Result - Similar pages

[MDB] Table ID
File Format: Microsoft Access 1 - View as HTML
... 30.00. Standard. 0.00. 0.00. 1.00. True. 255.00. 0.00. False. Version, **Password**, R/W Options. 1, 3. NONE. 0. ID, Course ID, Parent, Standard. 1, $36,137.84. $36,135.84 **...**
www.wtcsf.tec.wi.us/wids/standards/nvscinc.mdb - Supplemental Result - Similar pages
[More results from www.wtcsf.tec.wi.us]

Googling for a Recent Exploit – Using Constraints

Cross – Site Framing

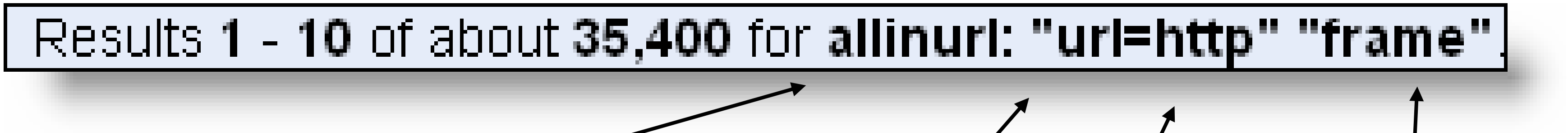
website.com/showframe.asp?src=fakesite.com/fakelogin.html

Site frames content

Content can be external

Frame source specified on client side

INURL



Results **1** - **10** of about **35,400** for **allinurl: "url=http" "frame"**

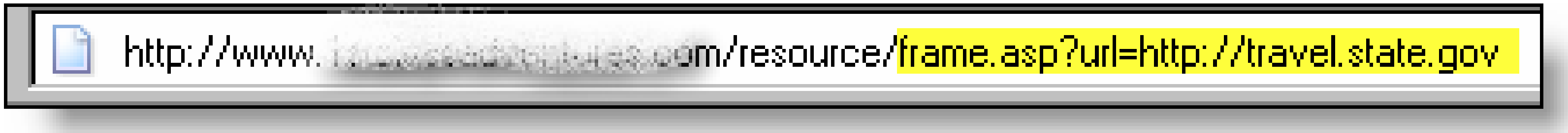
Restricts search terms to URL itself (buggy)

Want the source to be specified in the client

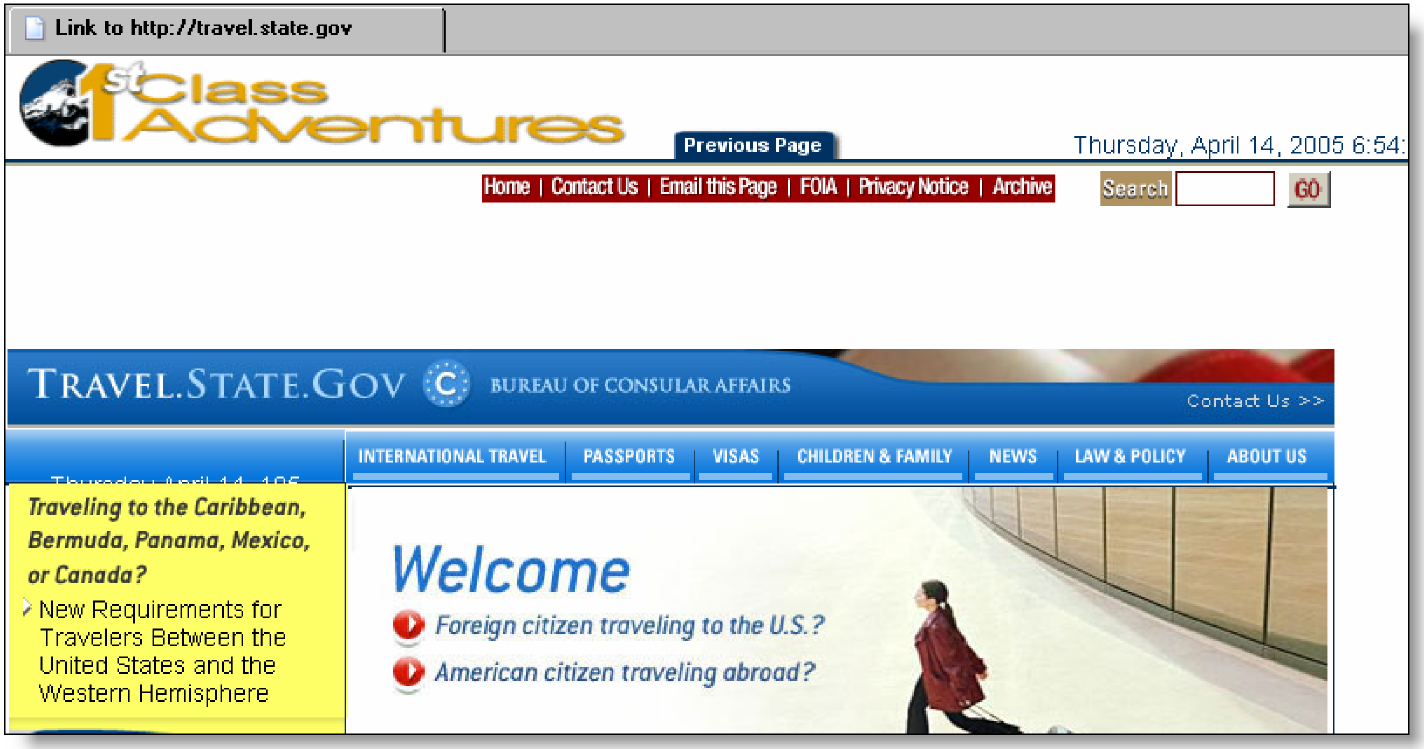
Want the source to be external; not on the same site

Further qualifier

Client-Sided Frame Source



http://www.[redacted]om/resource/frame.asp?url=http://travel.state.gov



Link to http://travel.state.gov

Thursday, April 14, 2005 6:54:

Traveling to the Caribbean, Bermuda, Panama, Mexico, or Canada?

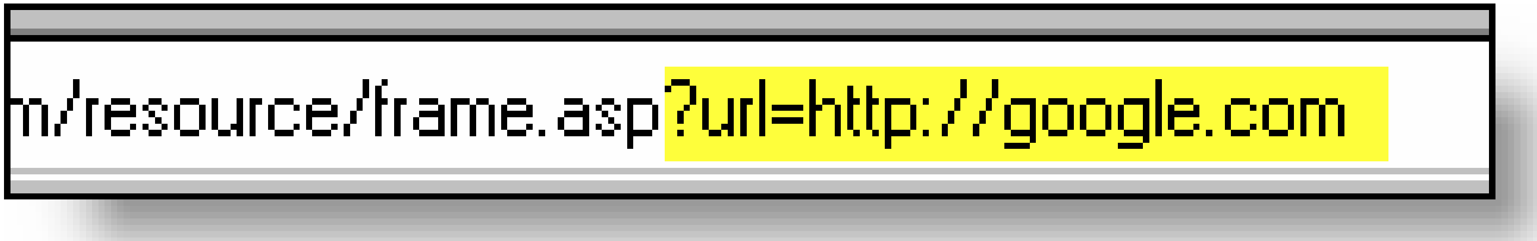
- New Requirements for Travelers Between the United States and the Western Hemisphere

Welcome

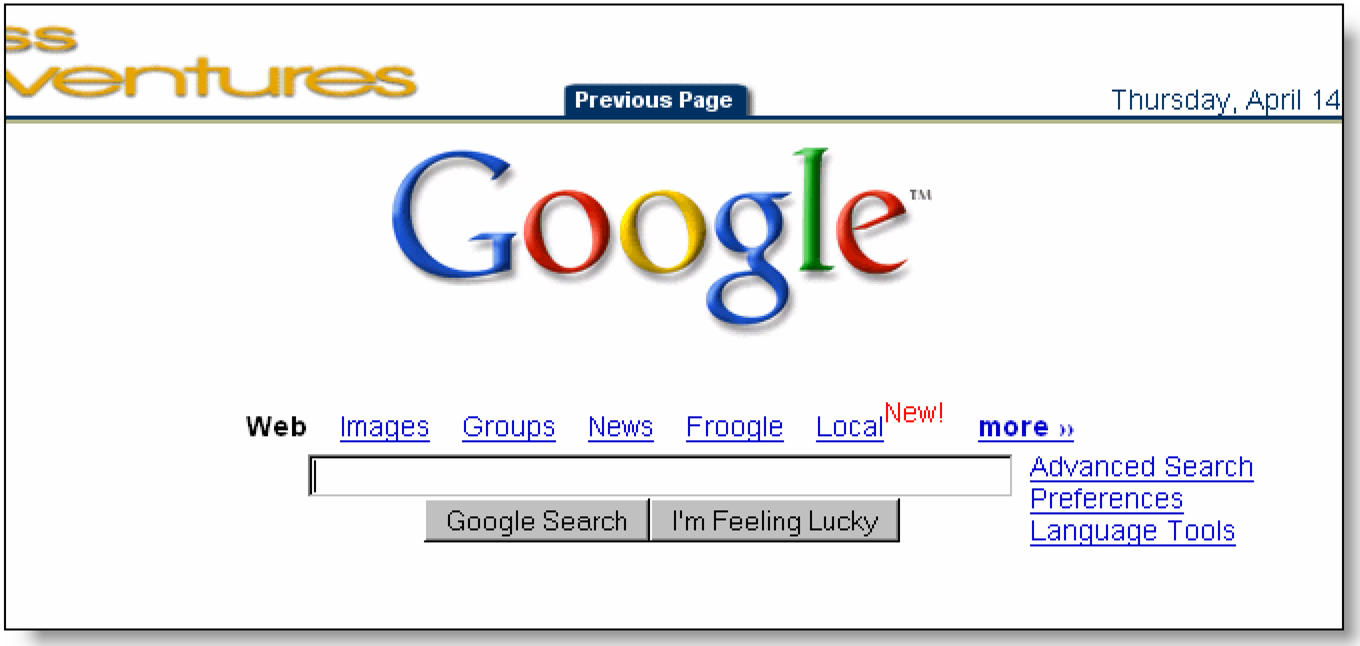
Foreign citizen traveling to the U.S.?

American citizen traveling abroad?

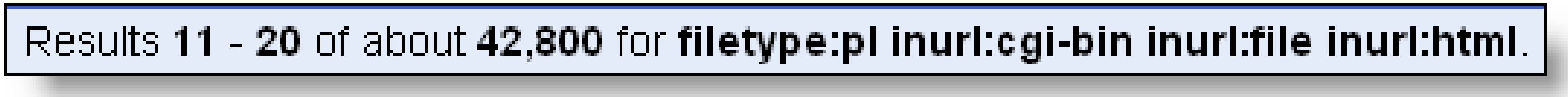
Framed.



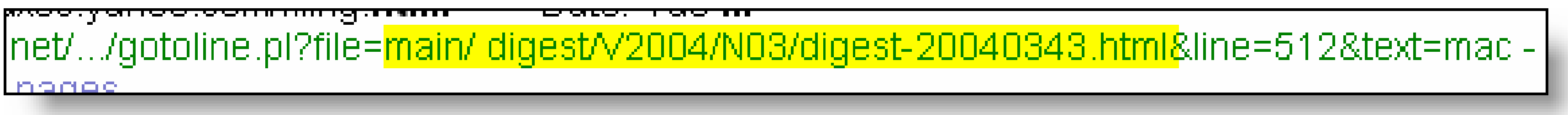
m/resource/frame.asp?url=http://google.com



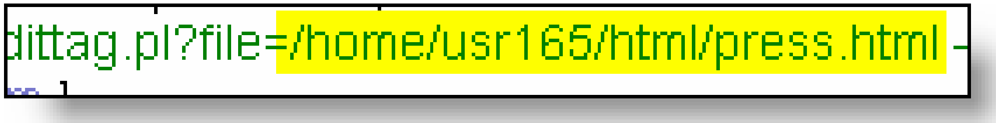
Directory Traversals !



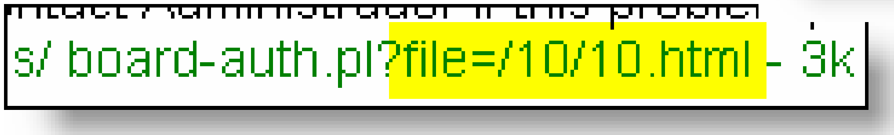
Results **11** - **20** of about **42,800** for **filetype:pl inurl:cgi-bin inurl:file inurl:html**.



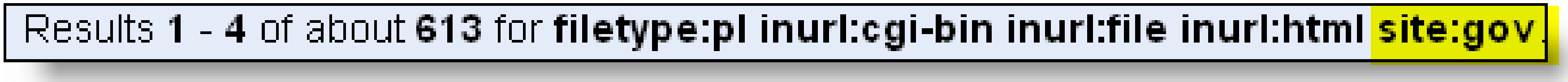
net/.../gotoline.pl?file=main/ digest/V2004/N03/digest-20040343.html&line=512&text=mac -



dittag.pl?file=/home/usr165/html/press.html -

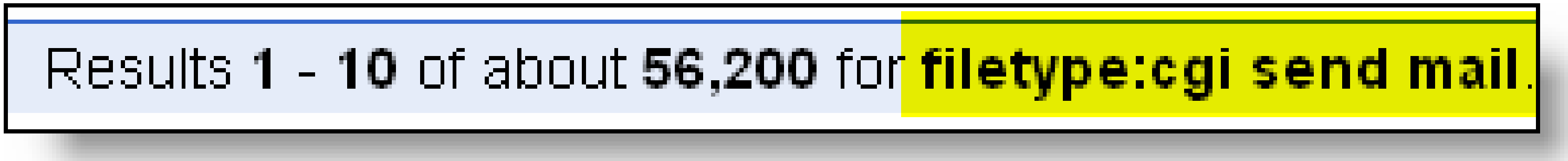


s/ board-auth.pl?file=/10/10.html - 3k

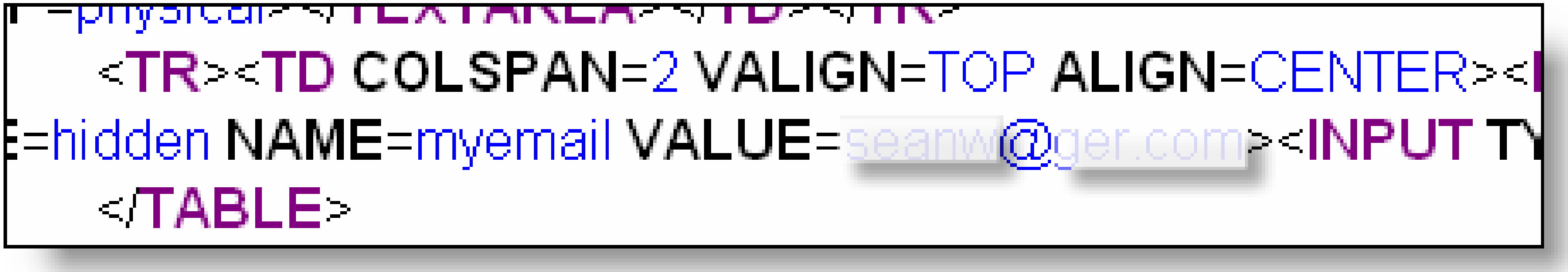


Results **1** - **4** of about **613** for **filetype:pl inurl:cgi-bin inurl:file inurl:html site:gov**.

SPAM ENGINES



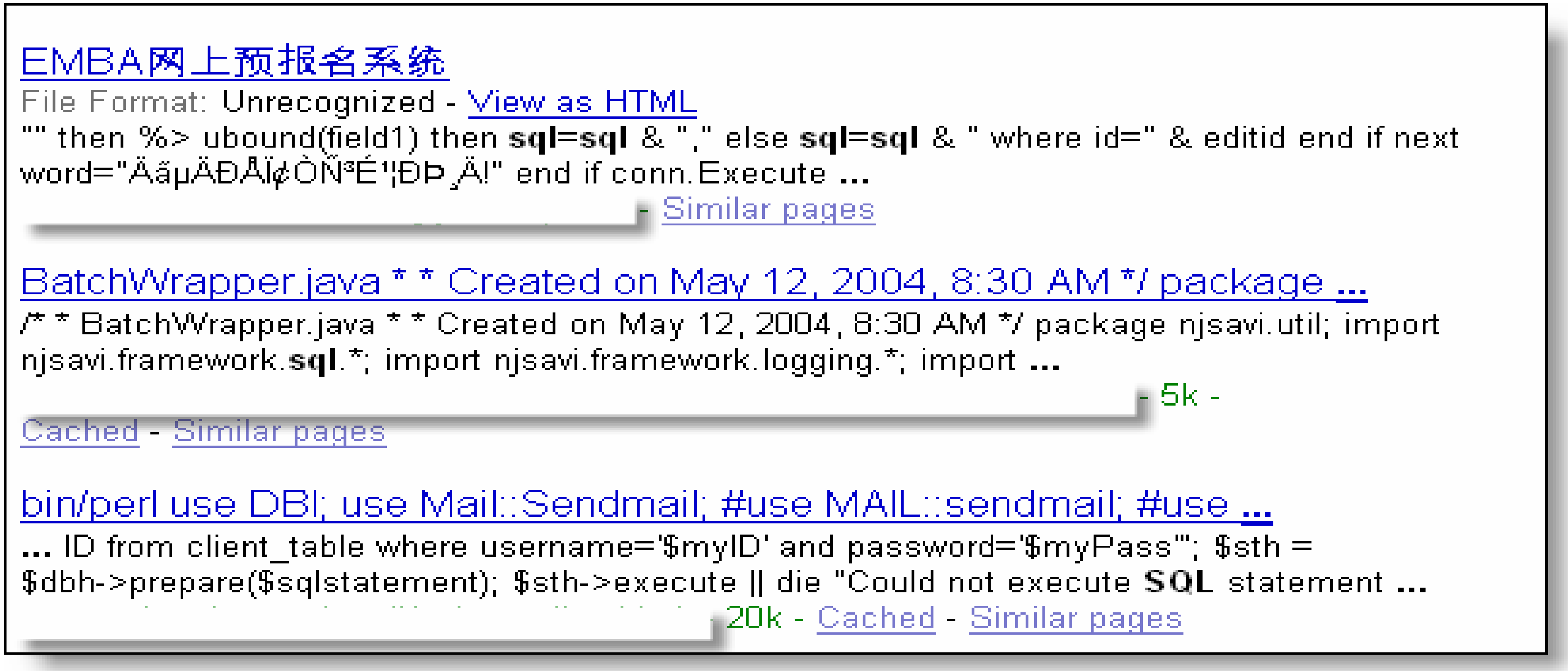
Results **1** - **10** of about **56,200** for **filetype:cgi send mail**.



```
T =physical></TEXTAREA></TD></TR>
    <TR><TD COLSPAN=2 VALIGN=TOP ALIGN=CENTER><I
E=hidden NAME=myemail VALUE=seanw@ger.com><INPUT TY
    </TABLE>
```

Source Code

Database queries.   They’re source code.
Hooray Source Code !



EMBA网上预报名系统
File Format: Unrecognized - View as HTML
"" then %> ubound(field1) then **sql=sql** & "," else **sql=sql** & " where id=" & editid end if next word="ÄãµÄÐÅÏ¢ÒÑ³É¹¦ÐÞ¸Ä!" end if conn.Execute **...**
- Similar pages

BatchWrapper.java * * Created on May 12, 2004, 8:30 AM */ package ...
/* * BatchWrapper.java * * Created on May 12, 2004, 8:30 AM */ package njsavi.util; import njsavi.framework.**sql**.*; import njsavi.framework.logging.*; import **...**
- 5k -
Cached - Similar pages

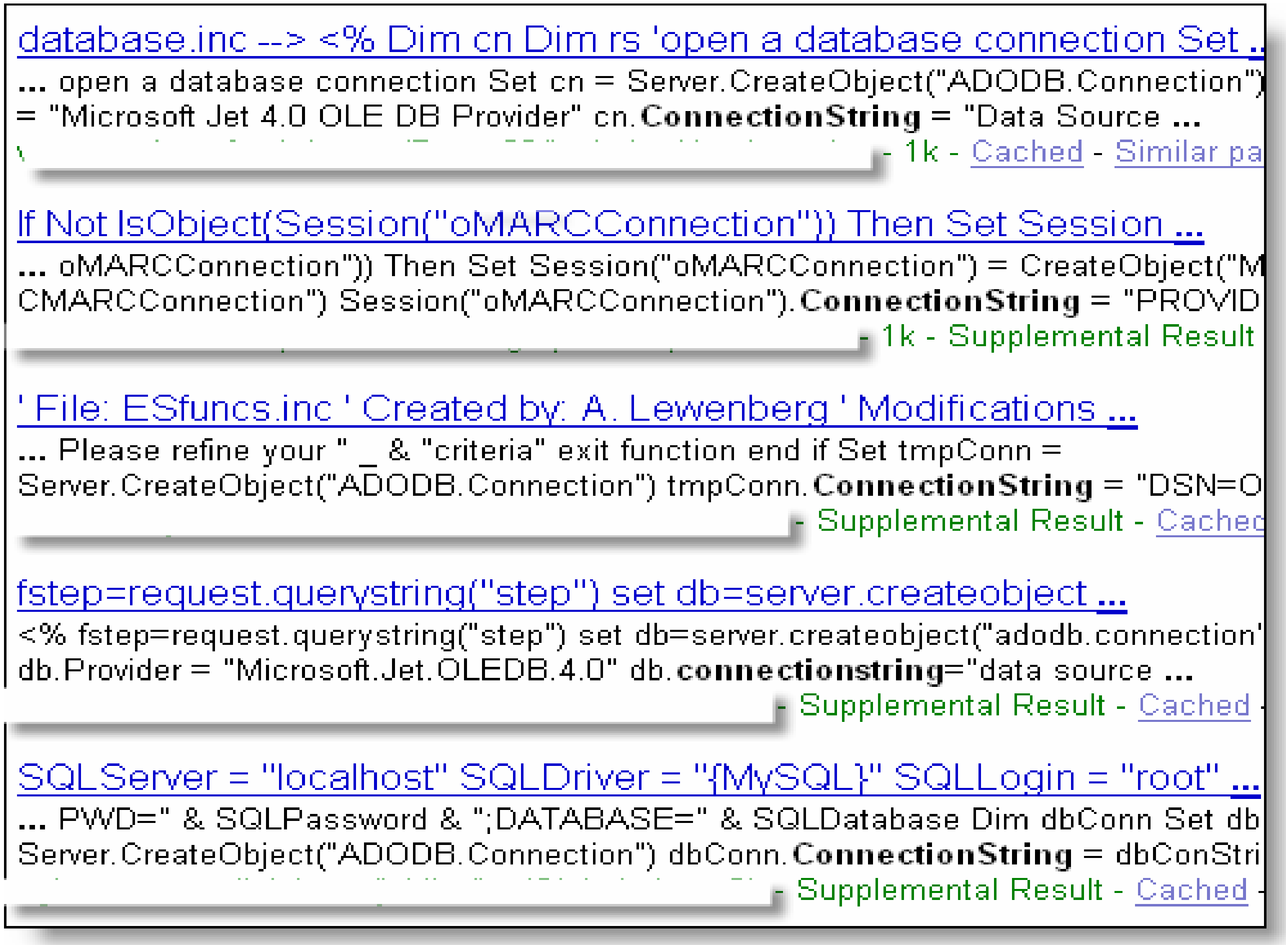
bin/perl use DBI; use Mail::Sendmail; #use MAIL::sendmail; #use ...
... ID from client_table where username='$myID' and password='$myPass'"; $sth = $dbh->prepare($sqlstatement); $sth->execute || die "Could not execute **SQL** statement **...**
20k - Cached - Similar pages

The Fun Never Stops

If you can read source code, what do source code do you read ?

Depends on what you’re interested in !

How about some database connection strings !



database.inc --> <% Dim cn Dim rs 'open a database connection Set ...
... open a database connection Set cn = Server.CreateObject("ADODB.Connection")
= "Microsoft Jet 4.0 OLE DB Provider" cn.**ConnectionString** = "Data Source ...
- 1k - Cached - Similar pa

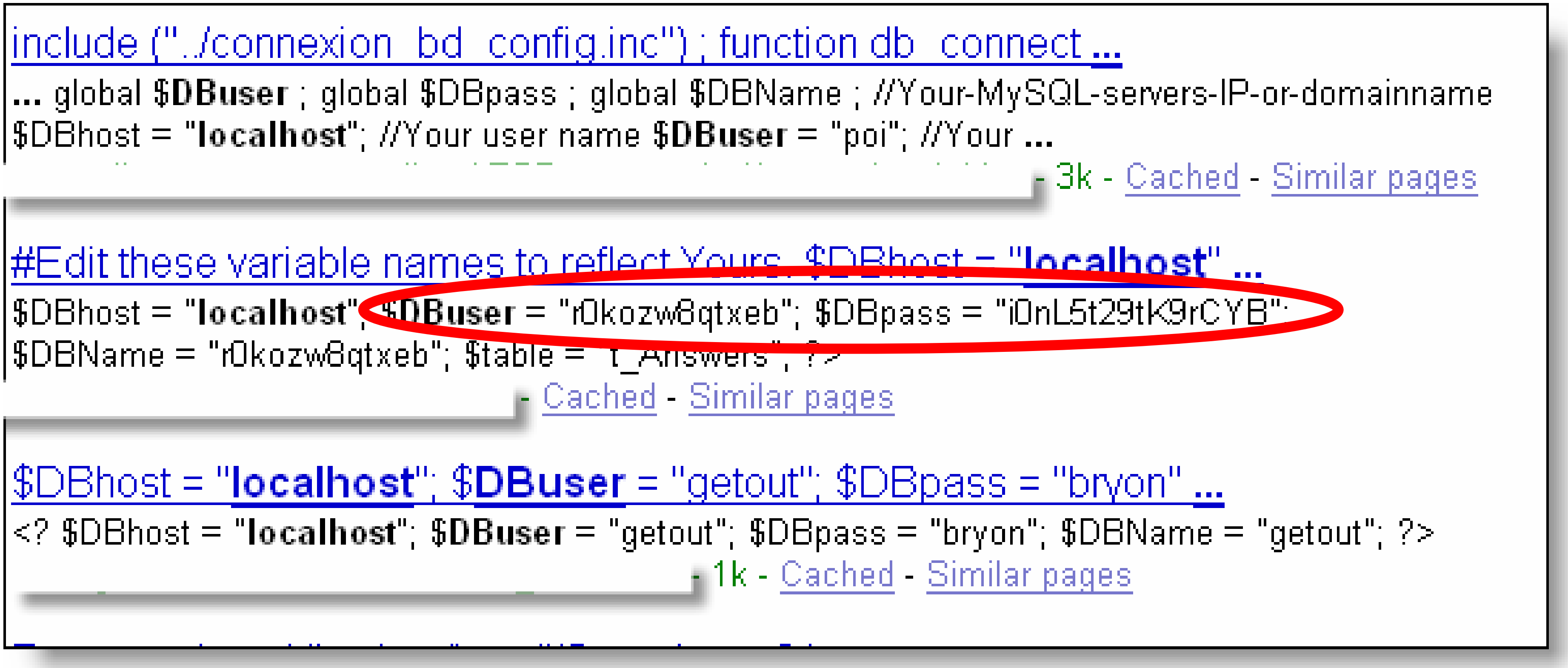
If Not IsObject(Session("oMARCConnection")) Then Set Session ...
... oMARCConnection")) Then Set Session("oMARCConnection") = CreateObject("M
CMARCConnection") Session("oMARCConnection").**ConnectionString** = "PROVID
- 1k - Supplemental Result

' File: ESfuncs.inc ' Created by: A. Lewenberg ' Modifications ...
... Please refine your " _ & "criteria" exit function end if Set tmpConn =
Server.CreateObject("ADODB.Connection") tmpConn.**ConnectionString** = "DSN=O
- Supplemental Result - Cachec

fstep=request.querystring("step") set db=server.createobject ...
<% fstep=request.querystring("step") set db=server.createobject("adodb.connection"
db.Provider = "Microsoft.Jet.OLEDB.4.0" db.**connectionstring**="data source ...
- Supplemental Result - Cached -

SQLServer = "localhost" SQLDriver = "{MySQL}" SQLLogin = "root" ...
... PWD=" & SQLPassword & ";DATABASE=" & SQLDatabase Dim dbConn Set db
Server.CreateObject("ADODB.Connection") dbConn.**ConnectionString** = dbConStri
- Supplemental Result - Cached -

The Proverbial Post-It On the Monitor



include ("../connexion_bd_config.inc") ; function db_connect ...

... global $DBuser ; global $DBpass ; global $DBName ; //Your-MySQL-servers-IP-or-domainname $DBhost = "localhost"; //Your user name $DBuser = "poi"; //Your ...

[redacted] - 3k - Cached - Similar pages

#Edit these variable names to reflect Yours. $DBhost = "localhost" ...

$DBhost = "localhost"; $DBuser = "r0kozw8qtxeb"; $DBpass = "i0nL5t29tK9rCYB"; $DBName = "r0kozw8qtxeb"; $table = "t_Answers"; ?>

[redacted] - Cached - Similar pages

$DBhost = "localhost"; $DBuser = "getout"; $DBpass = "bryon" ...

<? $DBhost = "localhost"; $DBuser = "getout"; $DBpass = "bryon"; $DBName = "getout"; ?>

[redacted] - 1k - Cached - Similar pages

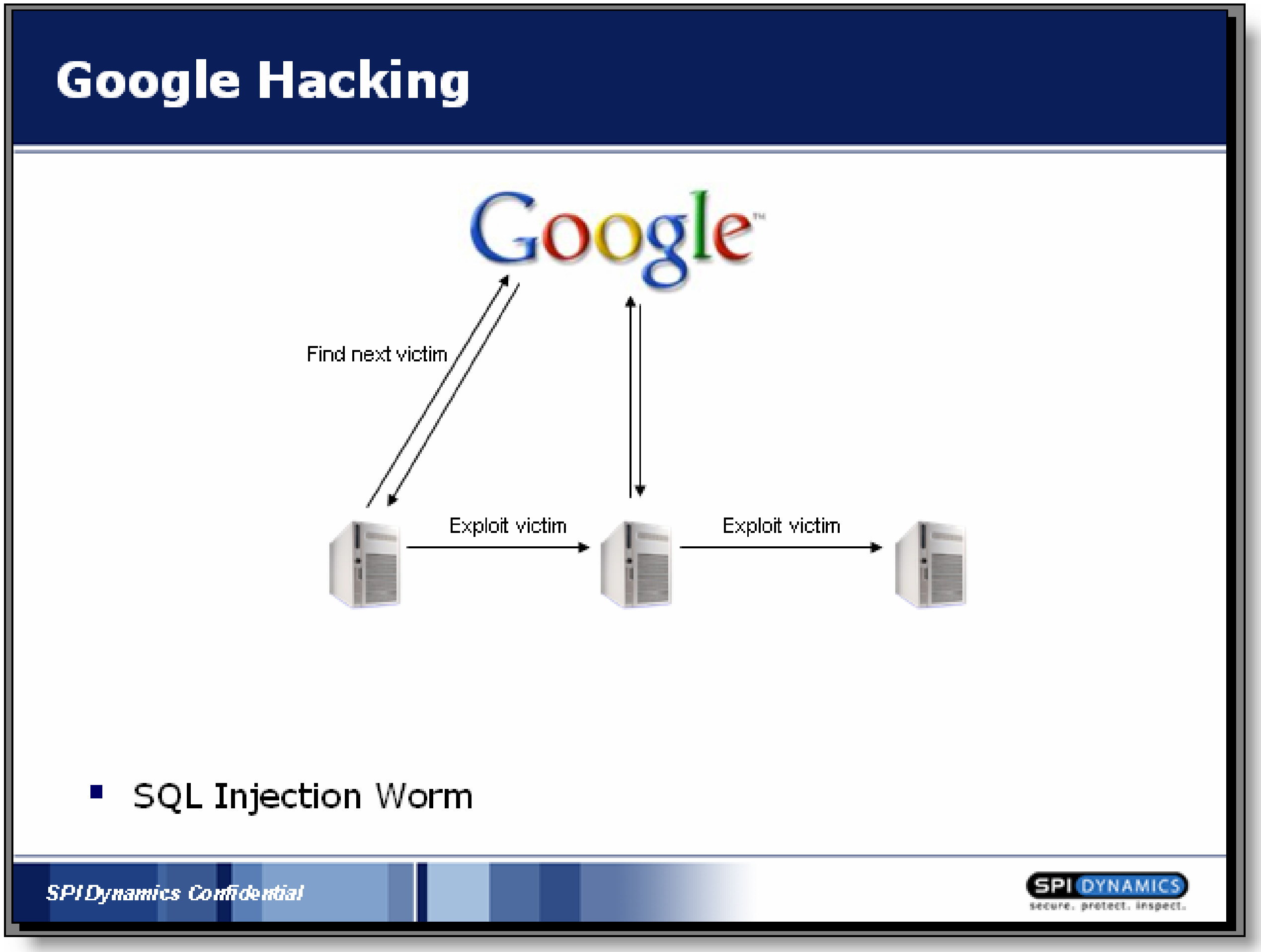
Yes, those are real live database connection strings

Yes, they contain real live usernames and passwords

No, Special Agent, I didn’t try them out.

Web App Hacking’s Cool. Google Hacking’s Cool.

Everyone Thought This Was Crazy ….



Google Hacking

Find next victim

Exploit victim

Exploit victim

- SQL Injection Worm

SPI Dynamics Confidential

SPI DYNAMICS
secure. protect. inspect.

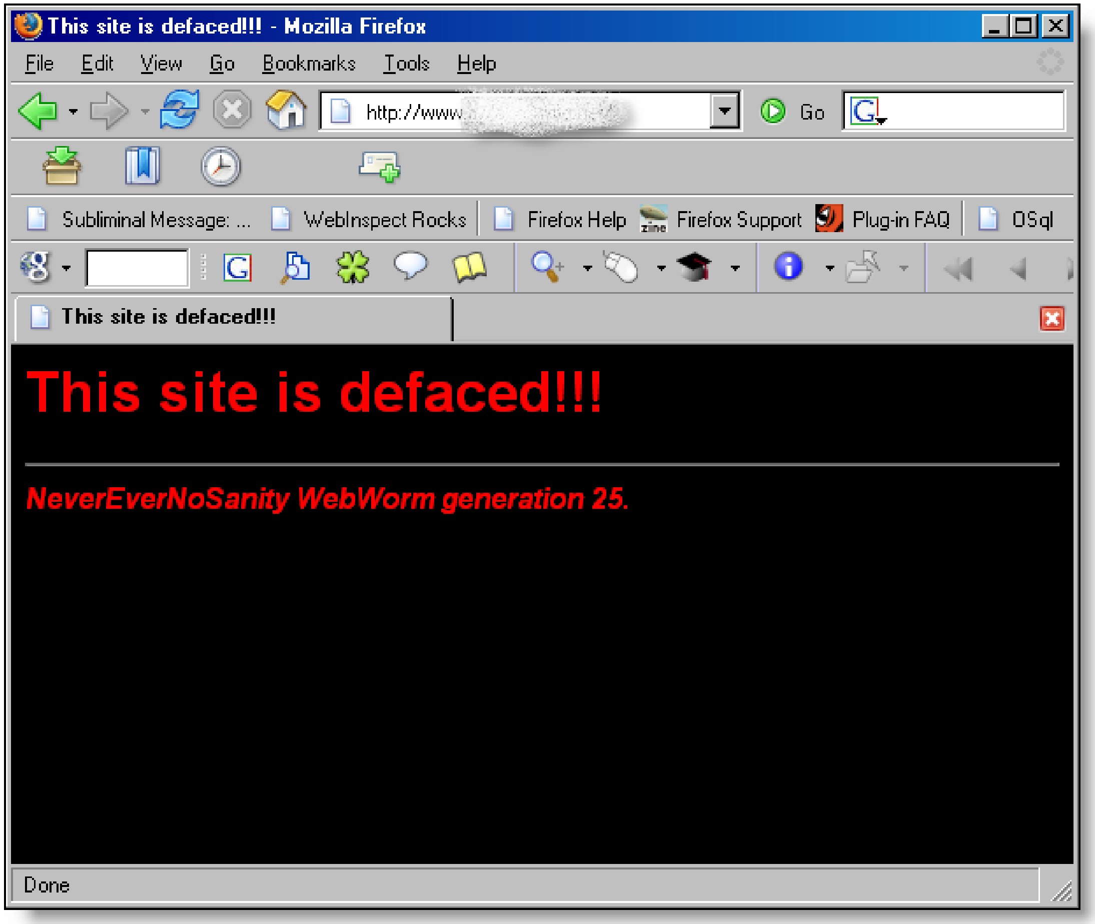
Then Santy Climbed Down the Chimney

December 20th 2004

Used a WEB APPLICATION VULNERABILITY in a common freeware PHP application

Used GOOGLE to ID new targets

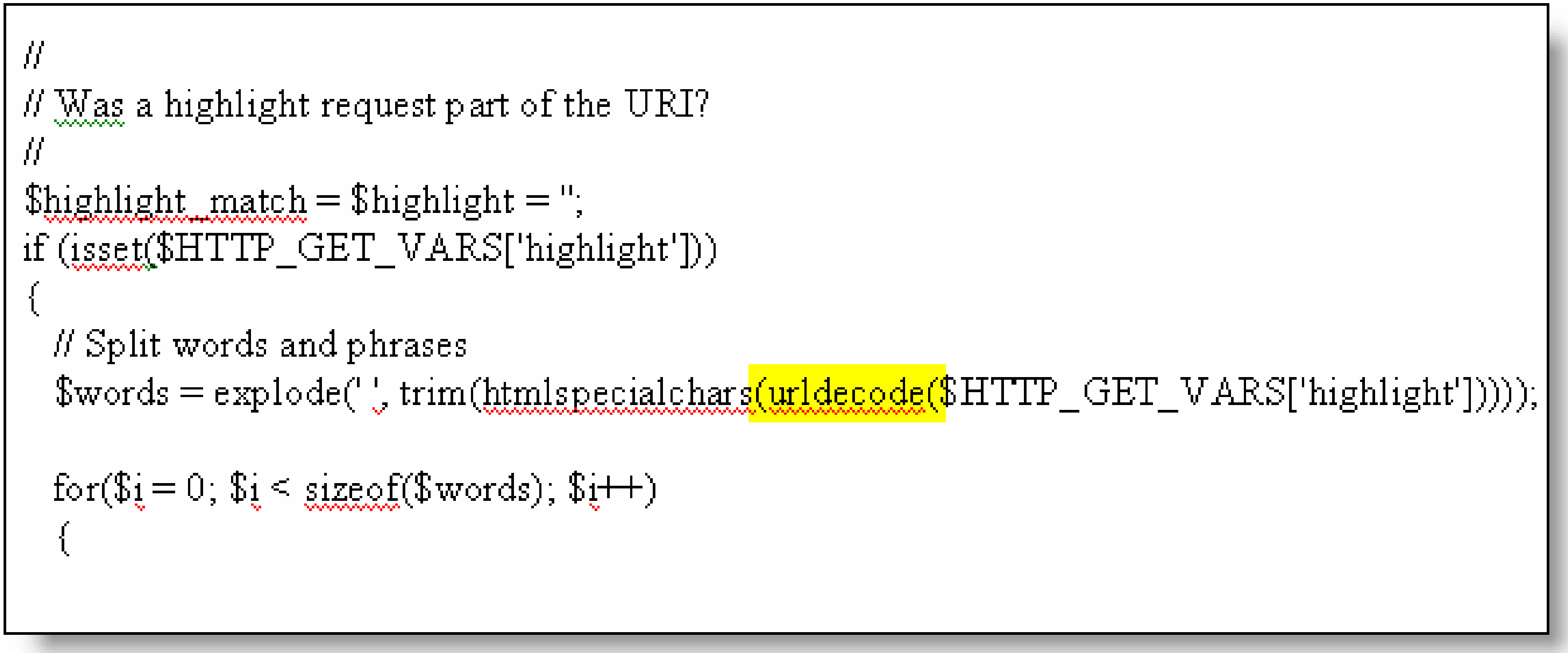
Multiple improved variants already out



This site is defaced!!!

NeverEverNoSanity WebWorm generation 25.

Code Review of the Vuln App



```
//
// Was a highlight request part of the URI?
//
$highlight_match = $highlight = '';
if (isset($HTTP_GET_VARS['highlight']))
{
   // Split words and phrases
   $words = explode(' ', trim(htmlspecialchars(urldecode($HTTP_GET_VARS['highlight']))));

   for($i = 0; $i < sizeof($words); $i++)
   {
```

URLDecode the input before removing special characters

MagicQuotes in PHP

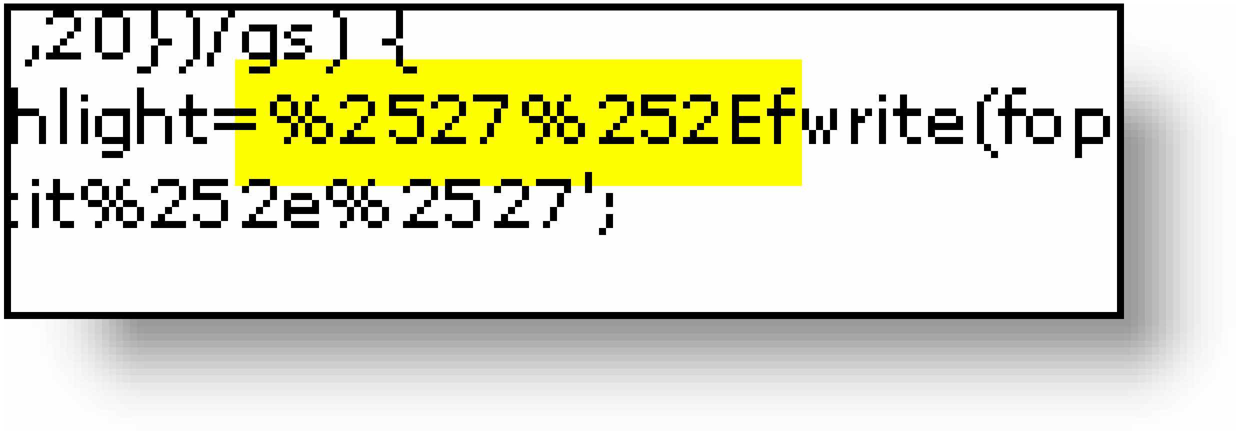
- Escapes single quotes
- Turns ‘ into \’
- Functional : prevents O’Malley and O’Brian from O’Crashing your query.
- MagicQuotes are magically functional, but not a security feature, and *were never meant to be*

Rasmus Lerdof says ...

“You always have to escape quotes before you can insert a string into a database. If you don't, you get an ugly SQL error and your application doesn't work. **After explaining this simple fact to people for the 50th time one day I finally got fed up and had PHP do the escaping on the fly**. This way the applications would work and the worst that would happen is that someone would see an extra \ on the screen when they output the data directly instead of sticking it into the database.”

Source: SitePoint.com, **Interview - PHP's Creator, Rasmus Lerdorf,** http://www.sitepoint.com/article/phps-creator-rasmus-lerdorf/3

Attack of the Worms: How it works

- 

```
,20})/gs) {
hlight=%2527%252Efwrite(fop
it%252e%2527';
```

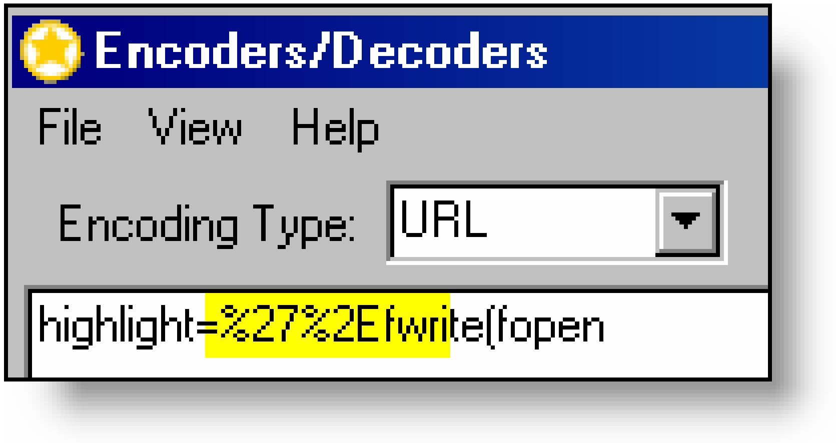
URLEncoded characters

PHP Fwrite command

PHP Fopen command

Decoding the attack

MagicQuotes recognizes plain and encoded single quotes



Encoding Type: URL

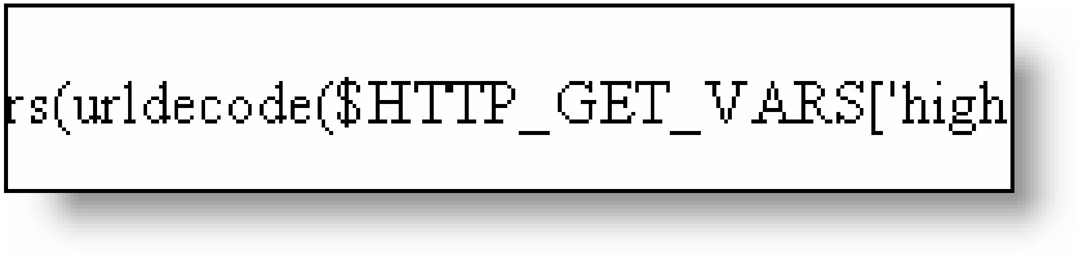
highlight=%27%2Efwrite(fopen

Decode once and compare

%27%2E is not a single quote

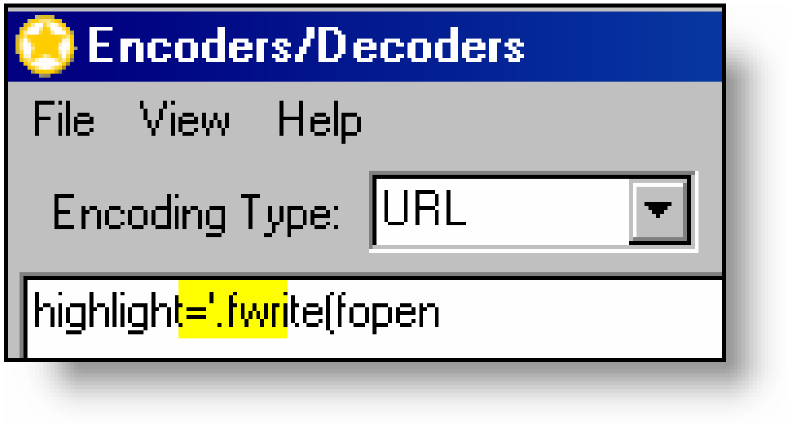
Back to the Code

Application decoded again in the code



```
rs(urldecode($HTTP_GET_VARS['high
```

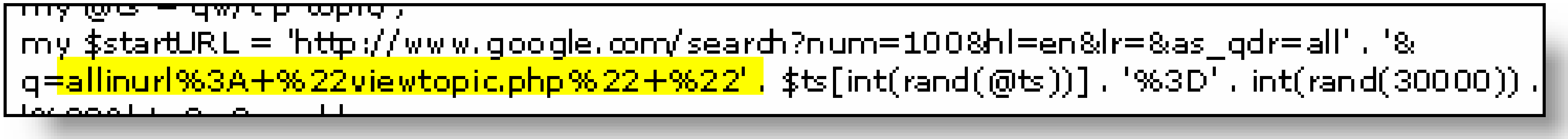
Turned the remaining %27%2E into ‘. Making the injection work.



Encoding Type: URL

highlight='.fwrite(fopen

Basic Google

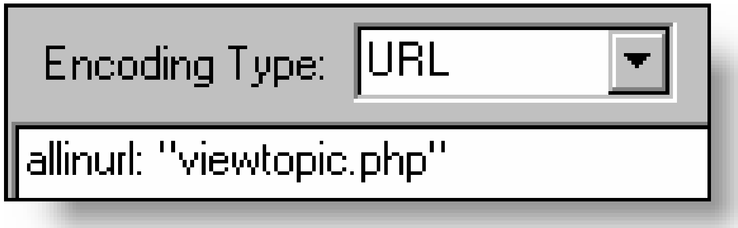


```
my $startURL = 'http://www.google.com/search?num=100&hl=en&lr=&as_qdr=all' . '&
q=allinurl%3A+%22viewtopic.php%22+%22' . $ts[int(rand(@ts))] . '%3D' . int(rand(30000)) .
```

Viewtopic.php with random numbers as a parameter (1414414=5858583)

Numbers NOT evasion – ensure different websites in each result

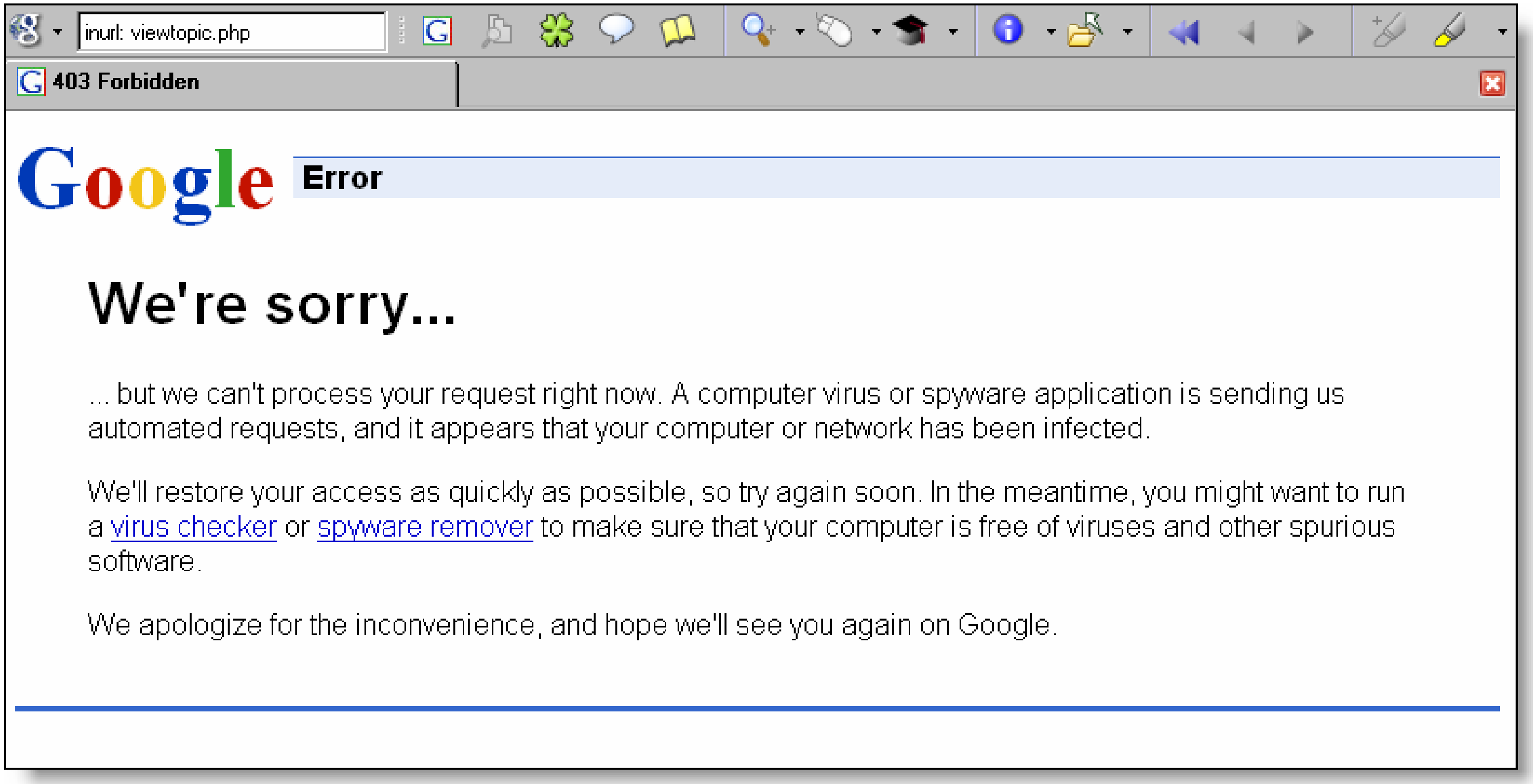
Unimaginative and easily signatured



Encoding Type: URL

allinurl: "viewtopic.php"

Google shutdown the query ...



Error

We're sorry...

... but we can't process your request right now. A computer virus or spyware application is sending us automated requests, and it appears that your computer or network has been infected.

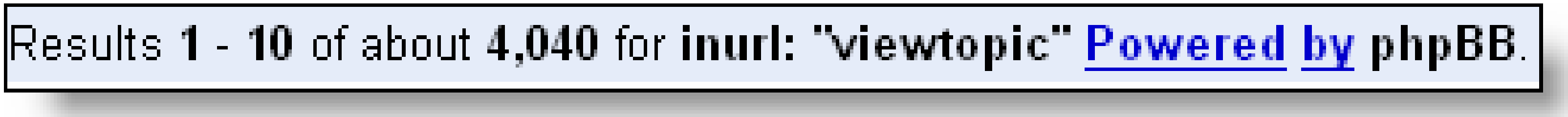
We'll restore your access as quickly as possible, so try again soon. In the meantime, you might want to run a virus checker or spyware remover to make sure that your computer is free of viruses and other spurious software.

We apologize for the inconvenience, and hope we'll see you again on Google.

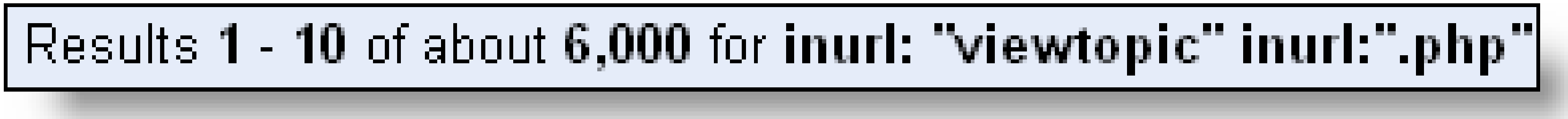
And gave me spyware advice …?

Google Evasion

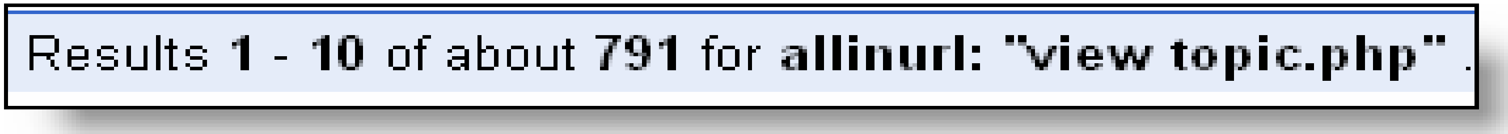
Viewtopic by itself could be anything. Add phpBB’s footer and it’s more accurate

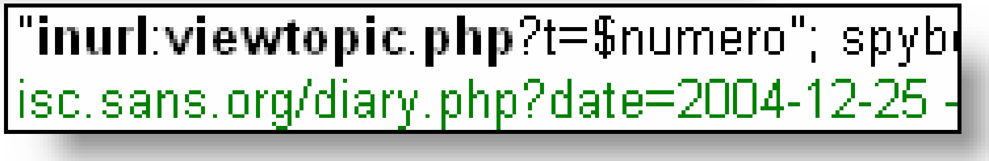

Results **1** - **10** of about **4,040** for **inurl: "viewtopic" Powered by phpBB**.

Viewtopic.php is not the same as **viewtopic** and **php**


Results **1** - **10** of about **6,000** for **inurl: "viewtopic" inurl:".php"**

Hmm …. Does Google recognize **Blank Spaces** ?

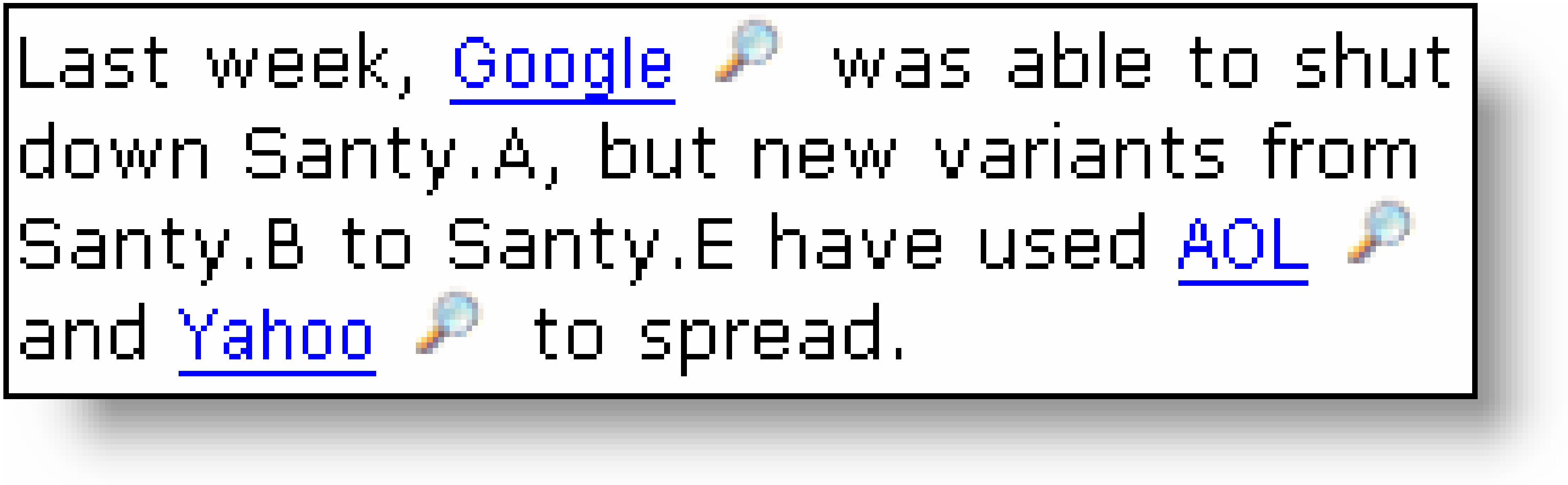

Results **1** - **10** of about **791** for **allinurl: "view topic.php"** .


"**inurl**:**viewtopic**.**php**?t=$numero"; spyb
isc.sans.org/diary.php?date=2004-12-25 -

Bonus :Spot the Google bug.

Or Just “Switch”

There’s more than one engine to search the web



Last week, Google was able to shut down Santy.A, but new variants from Santy.B to Santy.E have used AOL and Yahoo to spread.

4 Variants in JUST DAYS.

Prologue

- New Version of phpBoard released
- Remedial Action suggested to immediate users of the software was to remove the “URLDECODE”
- Prevents the second decode: ‘ remains as %27
- Still not rock solid input validation

Why Web Application Risks Occur

The Web Application Security Gap

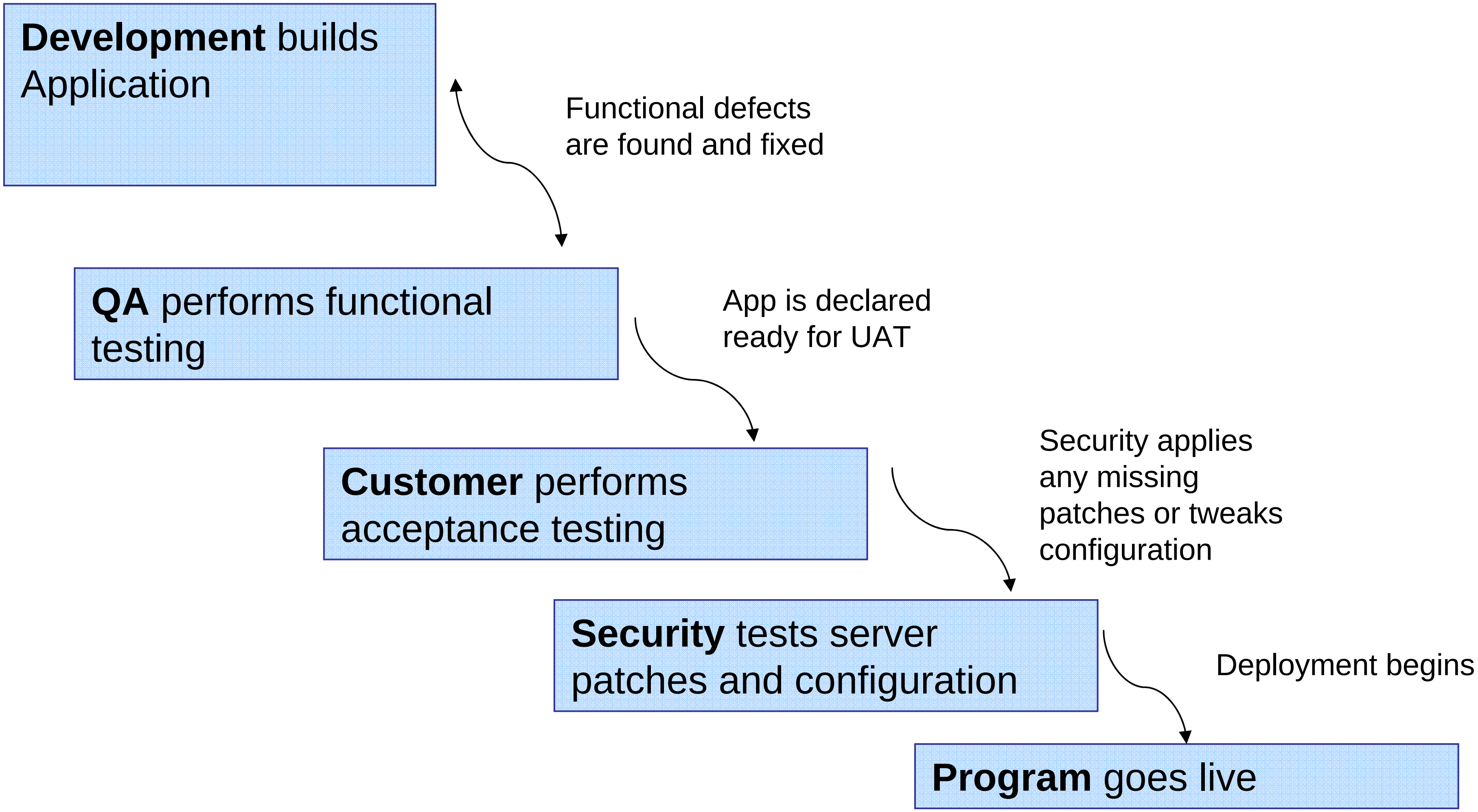
Security Professionals Don't Know The Applications

"As a Network Security Professional, I don't know how my company's web applications are supposed to work so I deploy a protective solution…but don't know if it's protecting what it's supposed to."

Application Developers and QA Professionals Don't Know Security

"As an Application Developer, I can build great features and functions while meeting deadlines, but I don't know how to build security into my web applications."

The Old Paradigm



Development builds Application

Functional defects are found and fixed

QA performs functional testing

App is declared ready for UAT

Customer performs acceptance testing

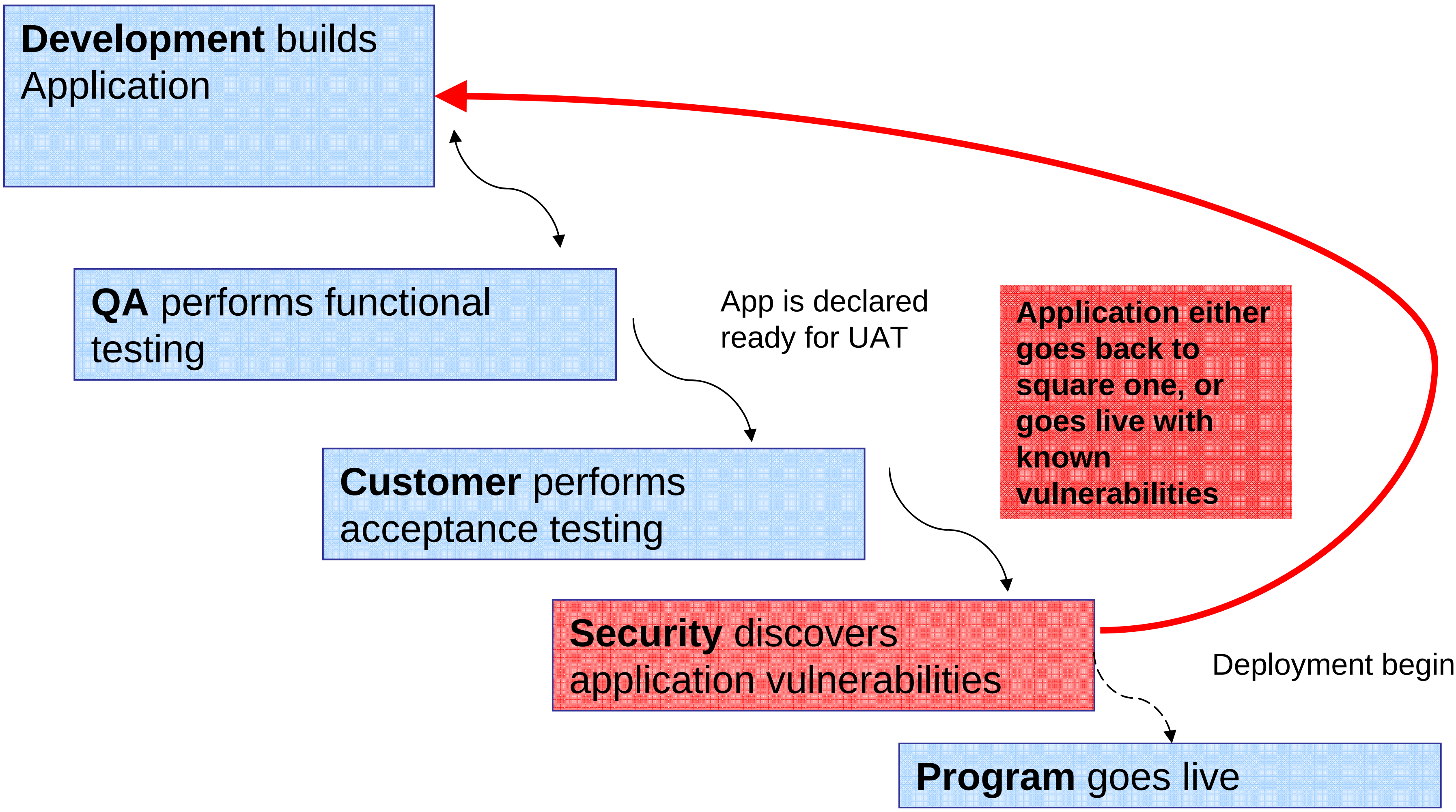
Security applies any missing patches or tweaks configuration

Security tests server patches and configuration

Deployment begins

Program goes live

Security Cannot Fix Application Issues



Development builds Application

QA performs functional testing

App is declared ready for UAT

Customer performs acceptance testing

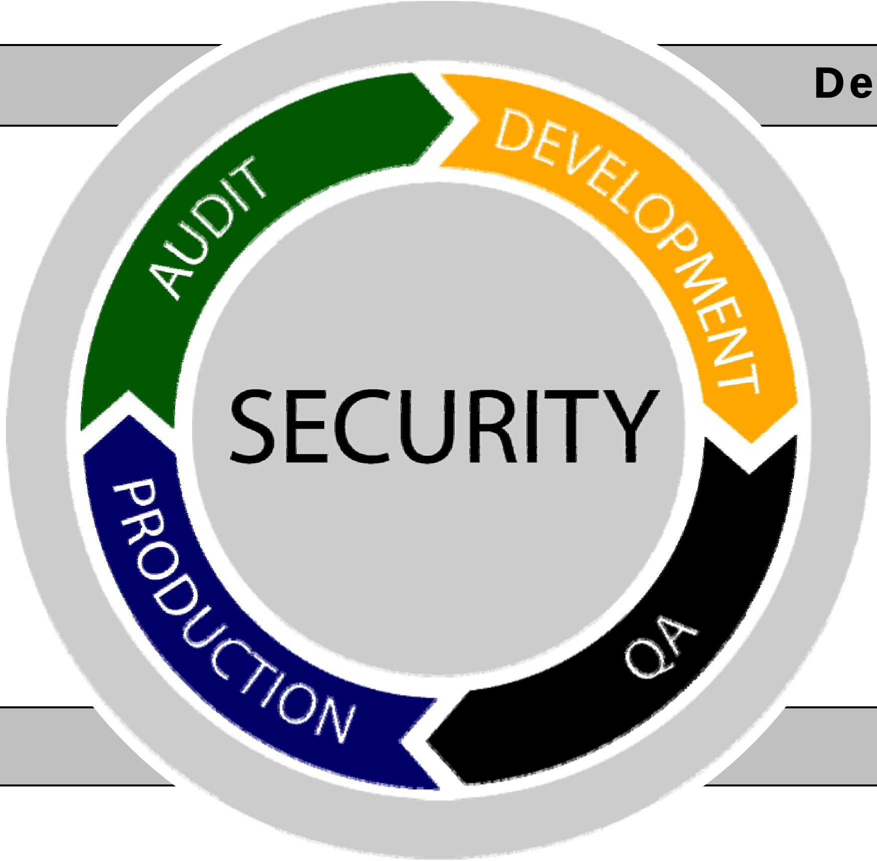
Application either goes back to square one, or goes live with known vulnerabilities

Security discovers application vulnerabilities

Deployment begins

Program goes live

Security Testing To The Application Lifecycle



Audit

Auditors, Dev, Compliance, and Business Subject Matter Experts (SME)

Development

Developers

Production

Security Operations and Auditors

QA

QA and Developers