



Ethical Hacking and Countermeasures

Version 6

Module XVII

Web Application
Vulnerabilities



TM

Scenario

Kimberly, a web application developer works for a bank, *XBank4u*. Recently *XBank4u* introduced a new service called “Mortgage Application Service”. Kimberly was assigned the task of creating the application which supported the new service.

She finds *ShrinkWarp*, an ASP based application on the Internet. The application suited perfectly for her development. She negotiates the price with the vendor and purchases the software for the firm.

She was successful in implementing the project in time. *XBank4u* was ready to serve its customers online for the new service using the application that Kimberly had designed.

A week later *XBank4u* website was defaced!

*Was Kimberly's decision to purchase the application justified?
Is it safe to trust a third party application?*



TM

News

Posted: 2008/01/21

Web application hacking: Inside the mind of an attacker

There's a tried and true method for seeking out the maximum number of vulnerabilities possible when testing your Web applications for security flaws. No, it's not a high-end Web application vulnerability scanner but rather a free "technique" that you can improve over time. You may not learn the methods overnight, but once you do, it's virtually guaranteed to take your Web vulnerability testing to the next level. It's stepping into the mindset of a malicious attacker and delving in to see *what else* in the Web application can be exploited.

Many people refer to this approach as penetration testing, but it's actually more than that. Technically speaking, it's called ethical hacking. This term always generates a few giggles, but it's indeed a valid form of security testing. The thing is, you'll find that by looking at your Web applications from the dark side you'll uncover and exploit weaknesses that automated scanners or checklist audits wouldn't touch in a thousand years.

The malicious mindset isn't limited to the stereotyped "hacker" as we know him. Anyone can have a malicious mindset -- not just an outsider. So, think about what an authenticated and trusted insider could do. In many cases, it's not going to be fancy cross-site scripting (XSS) or SQL injection but rather basic login mechanism tampering or URL or form field manipulation. Maybe even exploiting file transfer capabilities or disabling certain security features that no one knew he had access to.

While working on a project recently, I came across an internal Web server that hosted the security management/control application for the organisation's data centre. When trying to log in to the application, it prompted me for the password. I didn't have it. This is where most security scans and checklist audits would stop. But taking things further, I thought I'd Google the Web server and application name (which were conveniently displayed on the login page) along with the words "default password". Within about 3 seconds I had the default login ID and password, and sure enough, they worked!

Source: <http://searchsecurity.techtarget.com.au/>

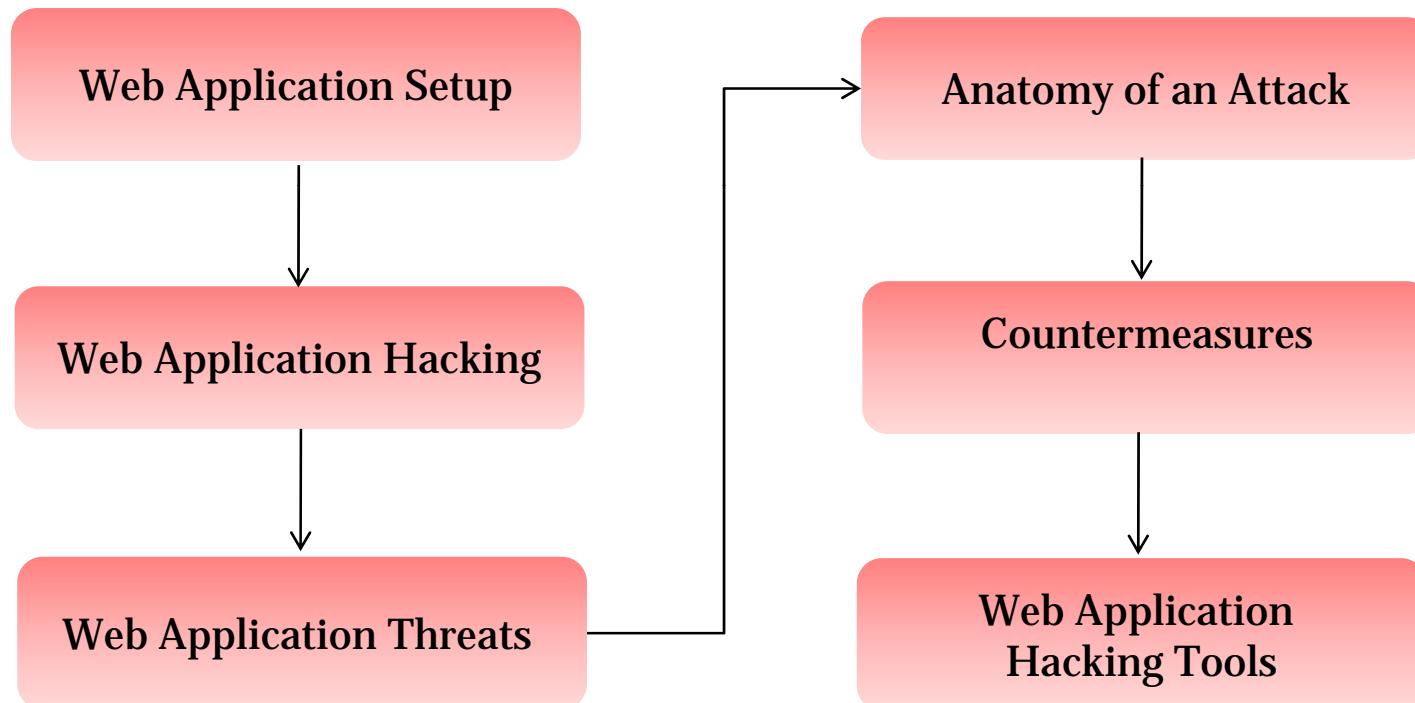


Module Objective

This module will familiarize you with :

- Web Application Setup
- Objectives of Web Application Hacking
- Anatomy of an Attack
- Web Application Threats
- Countermeasures
- Web Application Hacking Tools

Module Flow



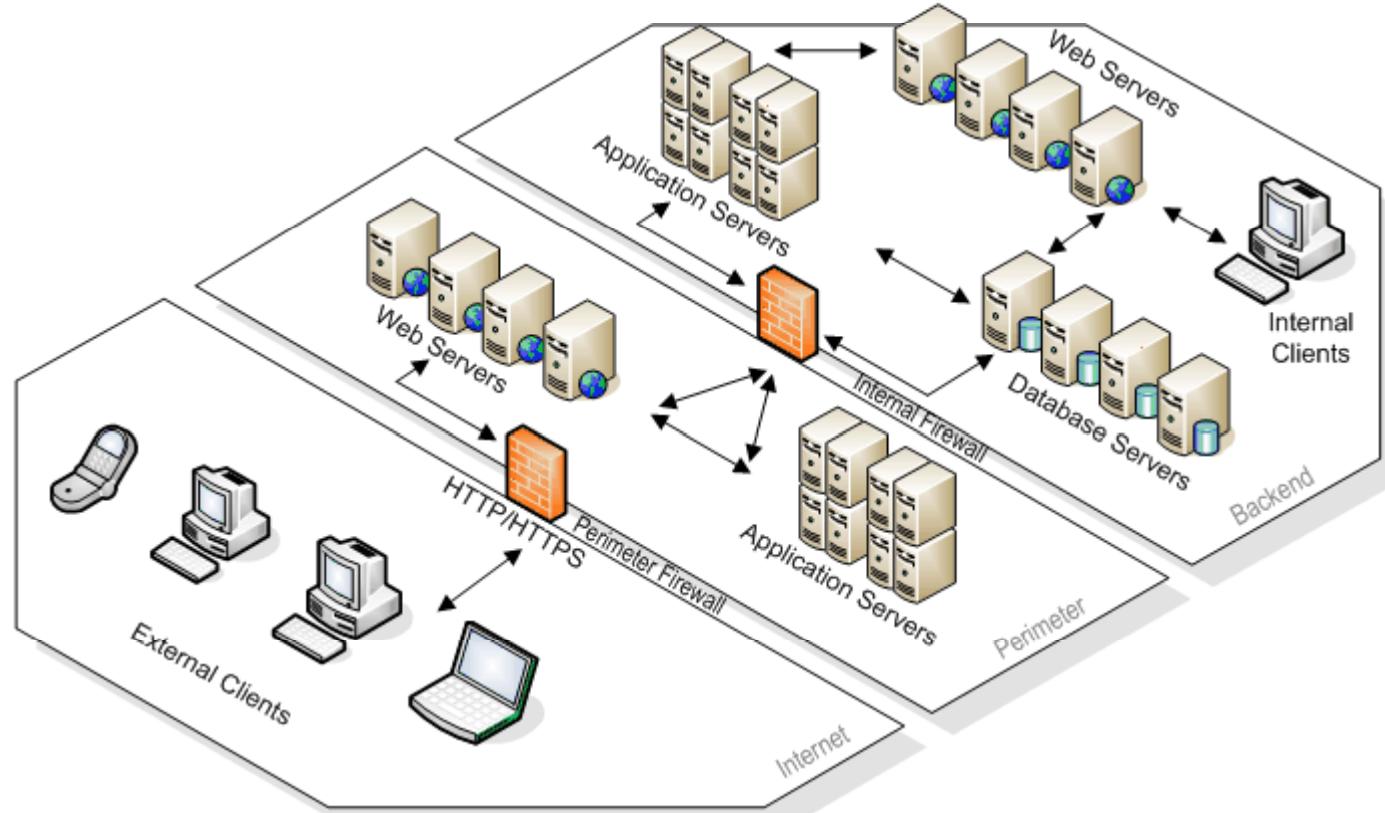
Web Application Setup

A client/server software application that interacts with users or other systems using HTTP

Modern applications are written in Java (or similar languages) and run on distributed application servers, connecting to multiple data sources through complex business logic tiers



Web Application Setup (cont'd)

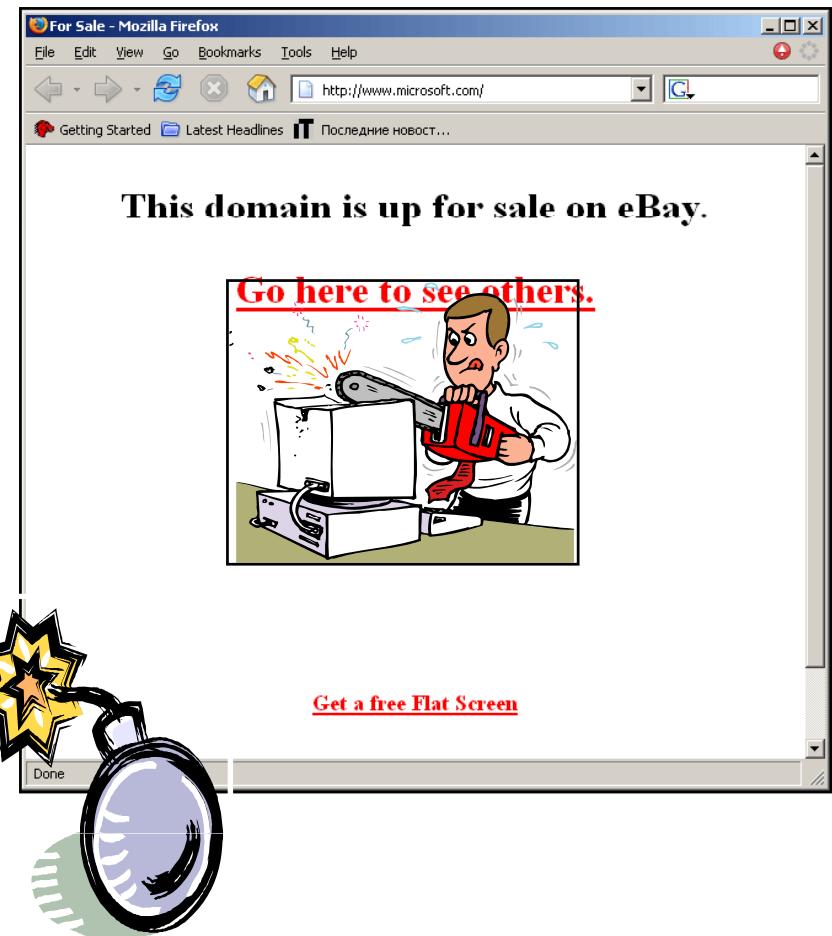




Web Application Hacking

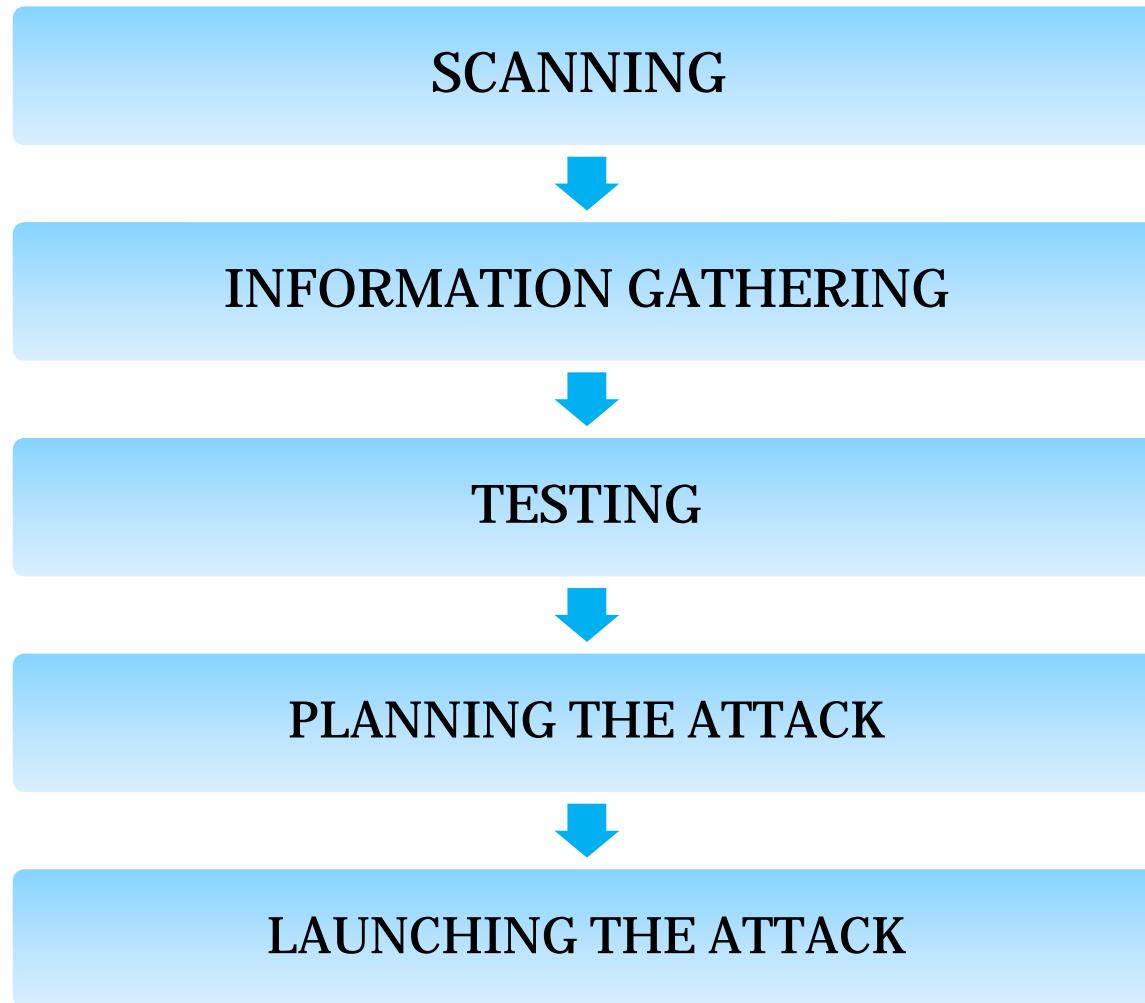
Exploitative behaviors

- Defacing websites
- Stealing credit card information
- Exploiting server-side scripting
- Exploiting buffer overflows
- Domain Name Server (DNS) attacks
- Employing malicious code
- Denial of Service
- Destruction of Data





Anatomy of an Attack





TM

Web Application Threats

Cross-site scripting

SQL injection

Command injection

Cookie/session poisoning

Parameter/form tampering

Buffer overflow

Directory traversal/forceful browsing

Cryptographic interception

Cookie snooping

Authentication hijacking

Log tampering

Error message interception attack

Obfuscation application

Platform exploits

DMZ protocol attacks

Security management exploits

Web services attacks

Zero day attack

Network access attacks

TCP fragmentation



TM

Cross-Site Scripting/XSS Flaws

Cross-site scripting occurs when an attacker uses a web application to send malicious code; generally JavaScript

Stored attacks are those where the injected code is permanently stored on the target servers in a database

Reflected attacks are those where the injected code takes another route to the victim, such as in an email message

Disclosure of the user's session cookie allows an attacker to hijack the user's session and take over the account

In cross-site scripting, end user files are disclosed, Trojan horse programs are installed, the user to some other page is redirected, and presentation of the content is modified

Web servers, application servers, and web application environments are susceptible to cross-site scripting



An Example of XSS

A hacker realizes that the XSECURITY website suffers from a cross-site scripting bug

The hacker sends you an e-mail that claims you have just won a vacation getaway and all you have to do is "click here" to claim your prize

The URL for the hypertext link is [www.xsecurity.com/default.asp?name=<script>evilScript\(\)</script>](http://www.xsecurity.com/default.asp?name=<script>evilScript()</script>)

When you click this link, the website tries to be friendly by greeting you, but instead displays, "Welcome Back!"

What happened to your name? By clicking the link in the e-mail, you have told the XSECURITY website that your name is <script>evilScript()</script>

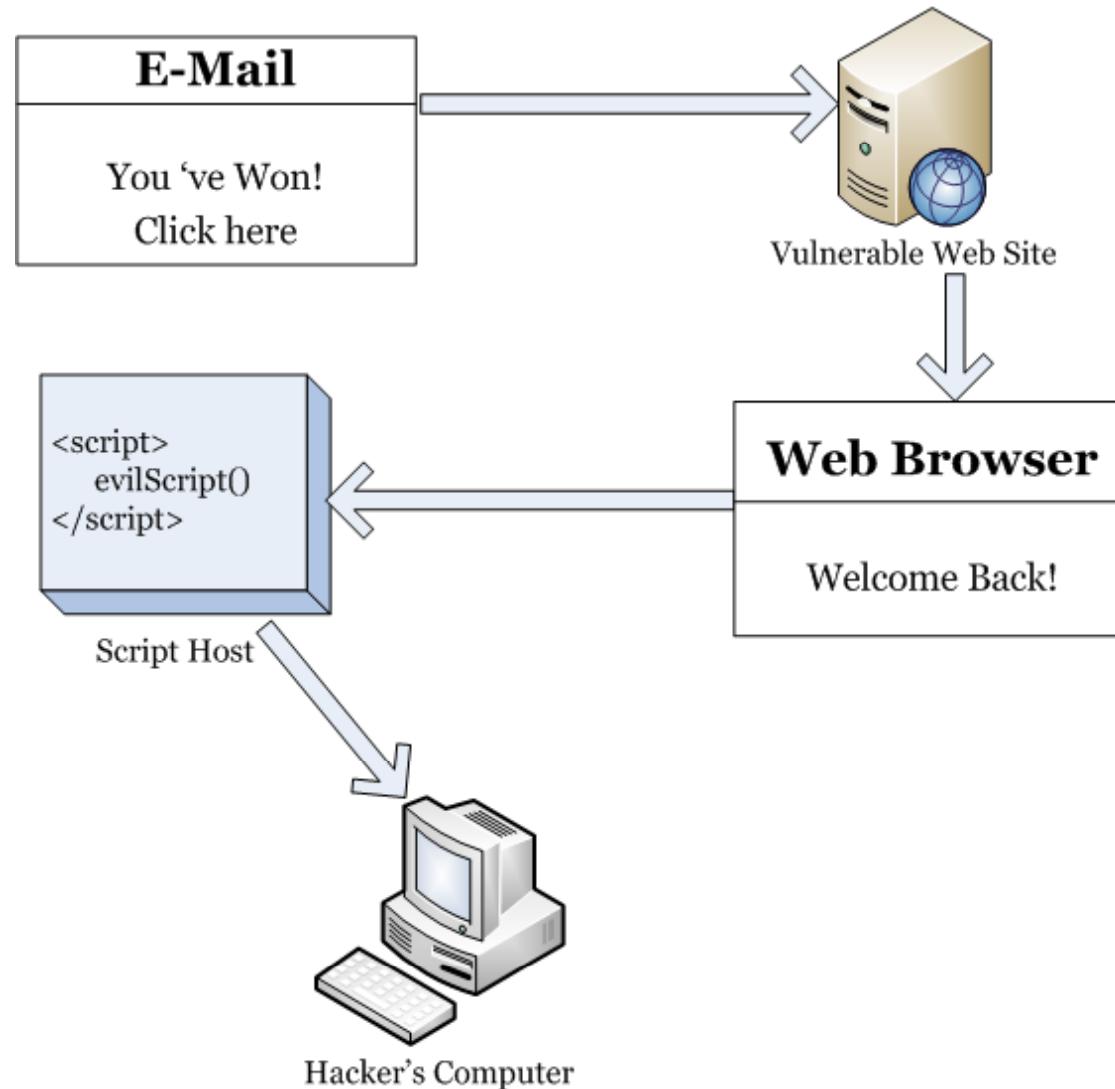
The web server generated HTML with this "name" embedded and sends it to your browser

Your browser correctly interprets this as script and runs the script

If this script instructs the browser to send a cookie containing your stock portfolio to the hacker's computer, it quickly complies

After all, the instruction came from the XSECURITY website, which owns that cookie

An Example of XSS (cont'd)



Countermeasures

Validate all headers, cookies, query strings, form fields, and hidden fields (i.e., all parameters) against a rigorous specification

Adopt a stringent security policy



Filtering script output can also defeat XSS vulnerabilities by preventing them from being transmitted to users

SQL Injection



SQL Injection uses SQL to directly manipulate database's data

An attacker can use a vulnerable web application to bypass normal security measures and obtain direct access to the valuable data

SQL Injection attacks can often be executed from the address bar, from within application fields, and through queries and searches

Countermeasure

- Check the user's input provided to database queries
- Validate and sanitize every user variable passed to the database

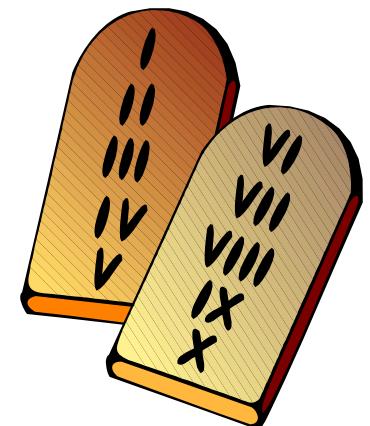


Command Injection Flaws

Command injection flaws relay the malicious code through a web application to another system

Attacks include calls to the operating system via system calls, the use of external programs via shell commands, as well as calls to the backend databases via SQL (i.e., SQL injection)

Scripts written in Perl, python, and other languages can be injected into the poorly designed web applications



Countermeasures

Use language-specific libraries that avoid problems due to shell commands

Validate the data provided to prevent any malicious content

Structure requests so that all supplied parameters are treated as data, rather than potentially executable content

J2EE environments allow the use of the Java sandbox, which can prevent the execution of system commands



Cookie/Session Poisoning

Cookies are used to maintain session state in the otherwise stateless HTTP protocol



Poisoning allows an attacker to inject the malicious content, modify the user's on-line experience, and obtain the unauthorized information



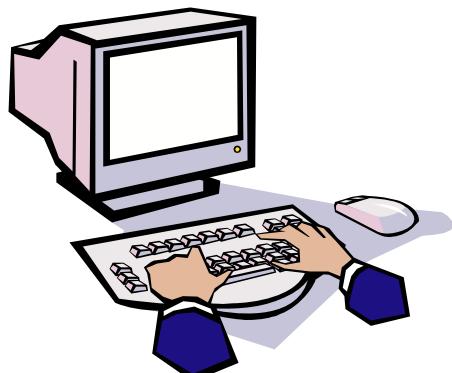
A proxy can be used for rewriting the session data, displaying the cookie data, and/or specifying a new user ID or other session identifiers in the cookie

Countermeasures



Do not store plain text or weakly encrypted password in a cookie

Implement cookie's timeout



Cookie's authentication credentials should be associated with an IP address

Make logout functions available

Parameter/Form Tampering

Parameter/Form tampering takes advantage of the hidden fields that work as the only security measure in some applications

Modifying this hidden field value will cause the web application to change according to the new data incorporated

It can cause theft of services, escalation of access, and session hijacking

Countermeasure: Field validity checking



Hidden Field at

JUGGYBOY.COM

The screenshot shows a web browser displaying the JUGGYBOY.COM website. On the left, there's a product listing for 'Training Slide Sets' with five items: 'Slide Set (1) Windows 2000 Professional', 'Slide Set (2) Windows 2000 Server', 'Slide Set (3) Windows 2000 Active Directory', 'Slide Set (4) Windows 2000 Network Infrastructure', and 'Complete Set (Includes all 4 sets - over 3000 slides)'. Each item has an 'Add to Cart' button. Below the list is a 'View Cart' button. A red arrow points from the 'Add to Cart' button of the second item to a Notepad window on the right.

The Notepad window title is 'products[1] - Notepad'. It contains the HTML source code for a PayPal payment form. Several hidden input fields are highlighted with red boxes:

```
method="post">
<input type="image"
src="https://www.paypal.com/en_US/i/btn/x-click-but22.gif"
border="0" name="submit" alt="Make payments with PayPal – it's fast, free and secure!">
<input type="hidden" name="add" value="1">
<input type="hidden" name="cmd" value="_cart">
<input type="hidden" name="business"
value="haja@juggyboy.com">
<input type="hidden" name="item_name" value="Slide Set (1)
Windows 2000 Professional">
<input type="hidden" name="amount" value="69.00">
<input type="hidden" name="no_note" value="1">
<input type="hidden" name="currency_code" value="USD">
<br>
&nbsp;</form></p>
</td>
</tr>
<tr>
<td width="59%" height="19"
style="border-style: none" bgcolor="#000000">
<font color="#FFFFFF" face="Garamond">Slide
Set (2) Windows 2000
Server</font></td>
<td width="18%" height="19" align="right"
style="border-style: none" bgcolor="#000000">
<font face="Garamond" color="#FFFFFF">USD
69.00</font></td>
<td width="23%" height="19"
style="border-style: none" bgcolor="#000000"
align="center">
<form target="paypal"
action="https://www.paypal.com/cgi-bin/webscr"
method="post">
<input type="image"
src="https://www.paypal.com/en_US/i/btn/x-click-but22.gif"
border="0" name="submit" alt="Make payments with PayPal – it's fast, free and secure!">
<input type="hidden" name="add" value="1">
<input type="hidden" name="cmd" value="_cart">
<input type="hidden" name="business"
value="haja@juggyboy.com">
<input type="hidden" name="item_name" value="Slide Set (1)
Windows 2000 Professional">
<input type="hidden" name="amount" value="69.00">
<input type="hidden" name="no_note" value="1">
<input type="hidden" name="currency_code" value="USD">
<br>
&nbsp;</form></p>
</td>
</tr>
```

Buffer Overflow

Buffer overflow is the corrupt execution stack of a web application

Buffer overflow flaws in custom web applications are less likely to be detected

Almost all known web servers, application servers, and web application environments are susceptible to attack (but not Java and J2EE environments except for overflows in the JVM itself)



```
execute error: '//bin/sh'  
This execution used shellcode that use 'Stack'.  
Its execution is very dangerous.  
Intercepting execution. This can prevent remote attack or local attack.  
  
example) Stack based Overflow. Format String attack ...  
Segmentation fault  
[root@test techNic]# ./for_xp32  
AAAAAAAAAAAAAA4$176x15$N6$47x17$N8$256x9$N10$192x11$N  
execute error: '//bin/sh'  
  
This execution used shellcode that use 'Stack'.  
Its execution is very dangerous.  
Intercepting execution. This can prevent remote attack or local attack.  
  
example) Stack based Overflow. Format String attack ...  
Segmentation fault  
[root@test techNic]#
```

Countermeasures



Validate input length in forms

Check bounds and maintain extra care when using loops to copy data

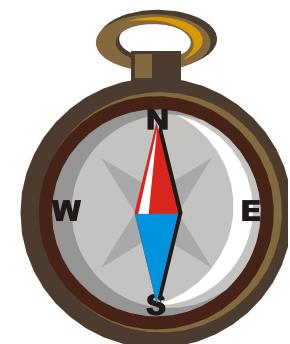
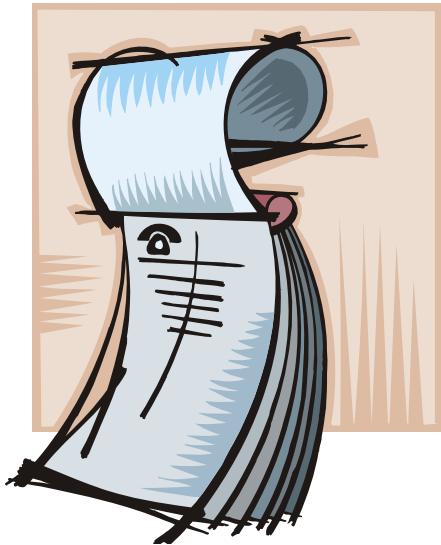
StackGuard and StackShield for Linux are tools to defend programs and systems against stack-smashing

Directory Traversal/Forceful Browsing

Directory traversal/forceful browsing attack occurs when the attacker is able to browse directories and files outside the normal application access

It exposes the directory structure of the application, and often the underlying web server and operating system

An attacker can enumerate contents, access secure or restricted pages, and gain confidential information, locate source code, and so on





TM

Countermeasures

Certified Ethical Hacker

Define access rights to the protected areas of the website

Apply checks/hot fixes that prevent the exploitation of the vulnerability such as Unicode to affect directory traversal

Web servers should be updated with security patches in a timely manner



Cryptographic Interception

Using cryptography, a confidential message can be securely sent between two parties



Encrypted traffic flows through network firewalls and IDS systems and is not inspected

If an attacker is able to take advantage of a secured channel, he/she can exploit it more efficiently than an open channel

Countermeasure

- Use of Secure Sockets Layer (SSL) and advanced private key protection



Cookie Snooping

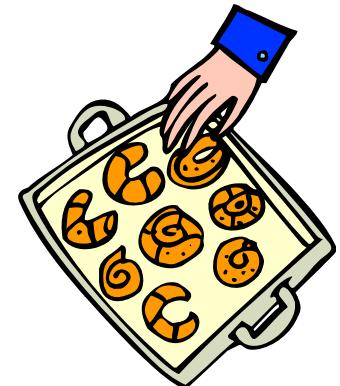
In an attempt to protect cookies, site developers often encode the cookies

Easily reversible encoding methods such as Base64 and ROT13 (rotating the letters of the alphabet 13 characters) give a false sense of the security regarding the use of cookies

Cookie snooping techniques can use a local proxy to enumerate cookies

Countermeasures:

- Use encrypted cookies
- Embed source's IP address in the cookie
- Integrate cookie's mechanism fully with SSL functionality for secured remote web application access



Authentication Hijacking

Authentication prompts a user to supply the credentials that allow access to the application

It can be accomplished through:

- Basic authentication
- Strong authentication methods

Web applications authenticate in varying methods

Enforcing a consistent authentication policy between multiple and disparate applications can prove to be a real challenge

A security lapse can lead to theft of service, session hijacking, and user impersonation

The screenshot shows a classic Microsoft .NET Passport sign-in interface. At the top, it says ".NET Passport Sign-in" and "Help". Below that is an "E-mail Address" field containing "name@hotmail.com". Next is a "Password" field with masked input. To the right of the password field is a checkbox labeled "Sign me in automatically." Below the fields is a large "Sign In" button. Underneath the "Sign In" button is another checkbox labeled "Do not remember my e-mail address for future sign-in. (Select this when using a public computer.)". At the bottom of the form, it says "Don't have a .NET Passport? [Get one now.](#)". Below that are links for "Member Services", "Terms of Use", and "Privacy Statement". A copyright notice at the bottom states "Some elements © 1999 - 2004 Microsoft® Corporation. All rights reserved." The Microsoft .NET logo is visible in the bottom right corner.



TM

Countermeasures

Use authentication methods that use secure channels wherever possible

Instant SSL can be configured easily to encrypt all traffic between the client and the application

Use cookies in a secure manner where possible



Log Tampering

Logs are kept to track the usage patterns of the application

Log tampering allows attackers to cover their tracks or alter web transaction records

Attackers strive to delete logs, modify logs, change user information, or otherwise destroy evidence of any attack

Countermeasure

- Digitally sign and stamp logs
- Separate logs for system events
- Maintain transaction log for all application events



Error Message Interception

Information in error messages is often rich with site-specific information that can be used to:

- Determine the technologies used in the web applications
- Determine whether the attack attempt was successful
- Receive hints for attack methods to try next

Countermeasure

- Website cloaking capabilities make enterprise web resources invisible to hackers



Attack Obfuscation

Attackers often work hard to mask and otherwise hide their attacks to avoid detection



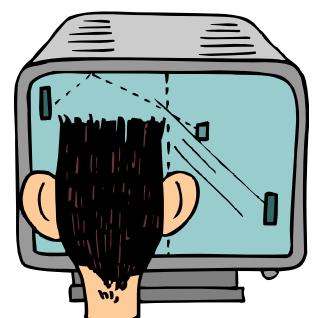
Most common method of attack obfuscation involves encoding portions of the attack with Unicode, UTF-8, or URL encoding

Multiple levels of encoding can be used to further bury the attack

It is used for theft of service, account hijacking, information disclosure, website defacement, and so on

Countermeasures:

- Thoroughly inspect all traffic
- Block or translate Unicode and UTF-8 encoding to detect attacks



Platform Exploits



Web applications are built upon application platforms, such as BEA Weblogic, ColdFusion, IBM WebSphere, Microsoft .NET, and Sun JAVA technologies

Vulnerabilities include the misconfiguration of the application, bugs, insecure internal routines, hidden processes and commands, and third-party enhancements



The exploit of application platform vulnerabilities can allow:

- Access to developer areas
- The ability to update application and site content



TM

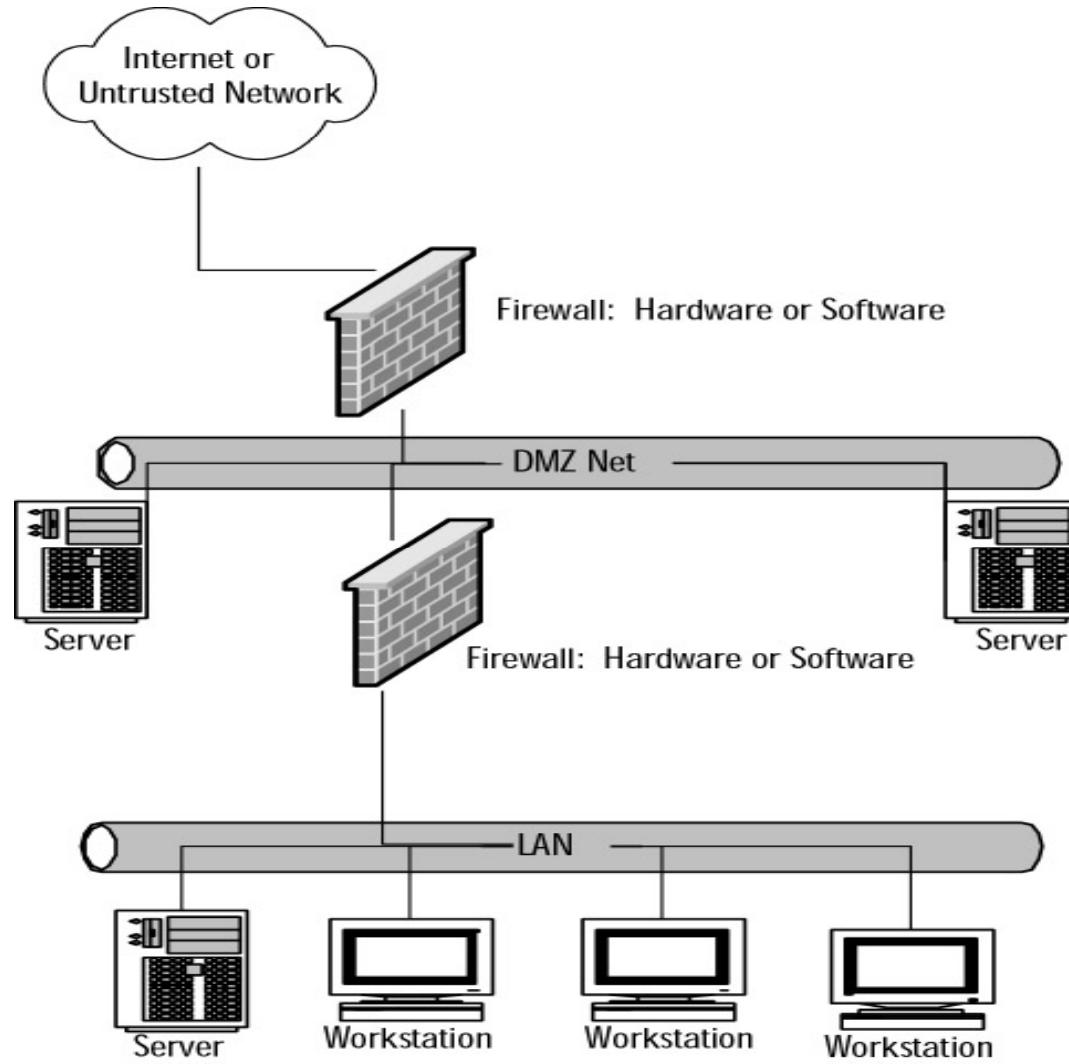
DMZ Protocol Attacks

DMZ (Demilitarized Zone) is a semi-trusted network zone that separates the untrusted Internet from the company's trusted internal network

Most companies limit the protocols allowed to flow through their DMZ

An attacker who is able to compromise a system that allows other DMZ protocols, has access to other DMZ and internal systems. This level of access can lead to:

- Compromise of the web application and data
- Defacement of websites
- Access to internal systems, including databases, backups, and source code



Countermeasures

Deploy a robust security policy

Adopt a sound auditing policy

Use signatures to detect and block well-known attacks

- Signatures must be available for all forms of attack and must be continually updated



Security Management Exploits

Security management systems are targeted to turn off security enforcement

An exploit of security management can lead to the modification of protection policies

Countermeasures

- There should be a single consolidated way to manage the security that is specific to each application
- Firewalls should be used



Web Services Attacks

Web services allow process-to-process communication between web applications



An attacker can inject a malicious script into a web service that will enable disclosure and modification of the data



Countermeasures:

- Turn off web services that are not required for regular operations
- Provision for multiple layers of protection
- Block all known attack paths without relying on signature database alone

Zero-Day Attacks

Zero-day attacks take place between the time a vulnerability is discovered by a researcher or attacker and the time that the vendor issues a corrective patch

Most zero-day attacks are only available as hand-crafted exploit code, but zero-day worms have caused rapid panic



Zero-day vulnerability is the launching point for further exploitation of the web application and environment

Countermeasures:



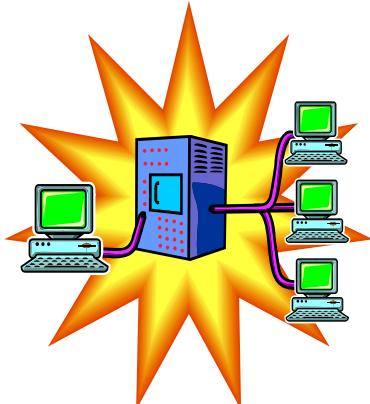
- No security solution can claim that they will totally protect against all zero-day attacks
- Enforce stringent security policies
- Deploy a firewall and enable heuristics (*heuristics*—common-sense rules drawn from experience—to solve problems) scanning

Network Access Attacks

All traffic to and from a web application traverses networks

These attacks use techniques like spoofing, bridging, ACL bypass, and stack attacks

Sniffing network traffic will allow viewing of application commands, authentication information, and application data as it traverses the network



Countermeasures

- Shut down unnecessary services thereby shutting unnecessary listening ports
- Define firewall rules to pass only legitimate traffic

TCP Fragmentation

Every message that is transferred between computers by a data network is broken down into packets

Often packets are limited to a pre-determined size for interoperability with physical networks

An attack directly against a web server would specify that the "Push" flag is set, which would force every packet into the web server's memory. In this way, an attack would be delivered piece-by-piece, without the ability to detect the attack

Countermeasure:

- Use packet filtering devices and firewall rules to thoroughly inspect the nature of the traffic directed at a web server





Hacking Tools

Instant Source

Wget

WebSleuth

BlackWidow

WindowBomb

Burp

cURL





Instant Source

Instant Source tool allows you to see and edit the HTML source code of the web pages

It can be executed from Internet Explorer where a new toolbar window displays the source code for any selected part of the page in the browser window

```
25 </head>
26 <body text="#000000"
27   bgcolor="#FFFFFF">
28   <table width="1000"
29     <tr>
30       <td width="200"
31       </td>
32       <td valign="top"
33         <div align="center"
34           </div>
35           <p class="BodyText">
36             <h1 class="HeaderText">Computer
37               Entertainment</a>
| <a href="
```

Source: <http://www.blazingtool.com>



TM

Instant Source: Screenshot

The screenshot shows a Microsoft Internet Explorer window displaying the Microsoft homepage (<http://www.microsoft.com/en/us/default.aspx>). A 'Page objects' tool window is open, listing external objects on the page, including several Microsoft URLs and a Flash movie section labeled 'None'. Below the tool window, the browser's developer tools display the raw HTML code for a specific element, which includes a reference to a Microsoft AlphaImageLoader script.

```
<A style="FILTER: progid:DXImageTransform.Microsoft.AlphaImageLoader(src='/global/a113ad1c-dacb-456b-b0bb-ee6a0ddfef3e.png')" onclick="WTID('08-00-073SUS0206');" href="http://www.microsoft.com/windows/products/windowsvista/buyorupgrade/upgradeadvisor.mspx"><IMG height=1 alt="Windows Vista Upgrade Advisor" src="http://i.microsoft.com/global/a113ad1c-dacb-456b-b0bb-ee6a0ddfef3e.png" width=1></A>
```



Wget is a command line tool for Windows and Unix that will download the contents of a website

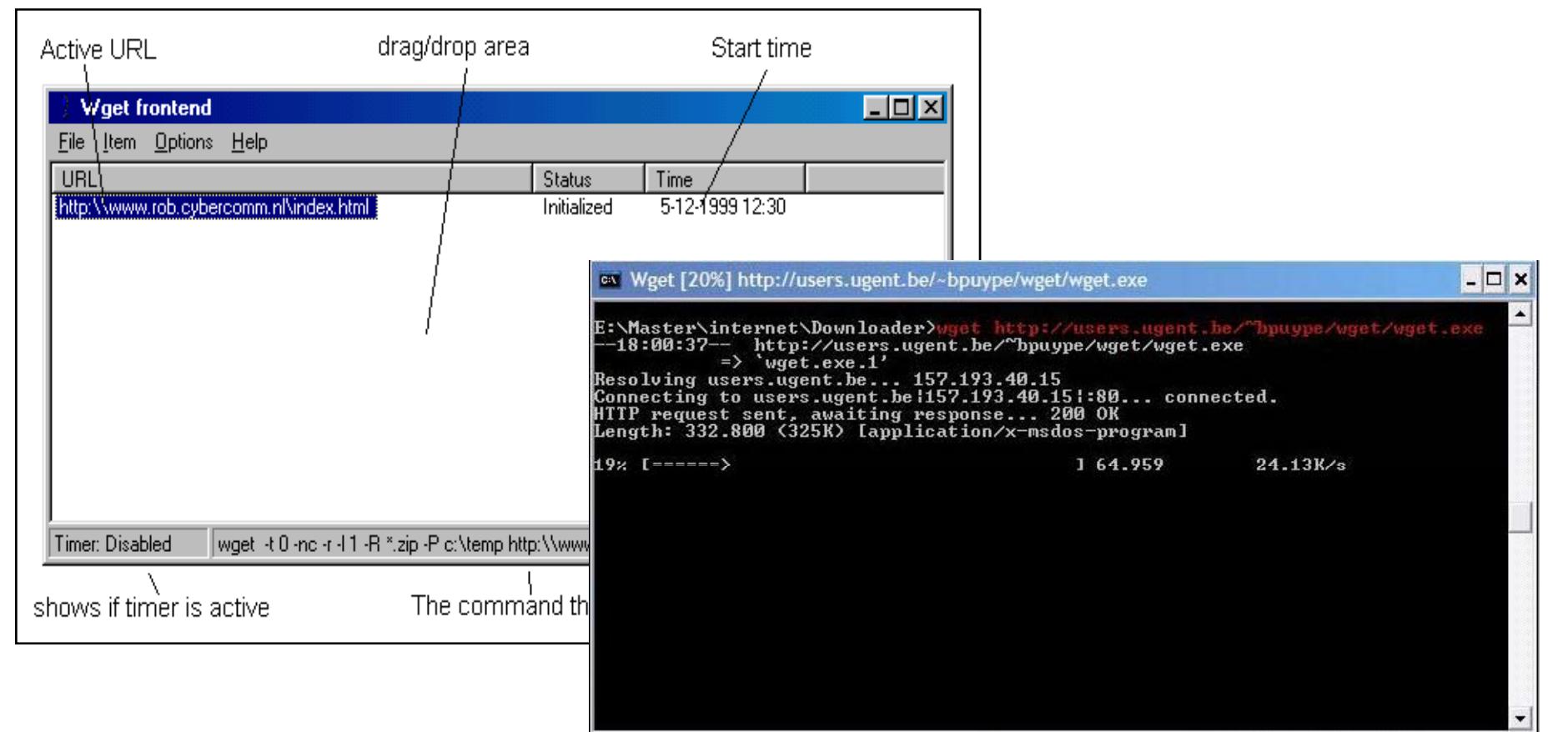
It works non-interactively in the background after the user logs off

It works particularly well with slow or unstable connections by continuing to retrieve a document until the document is fully downloaded

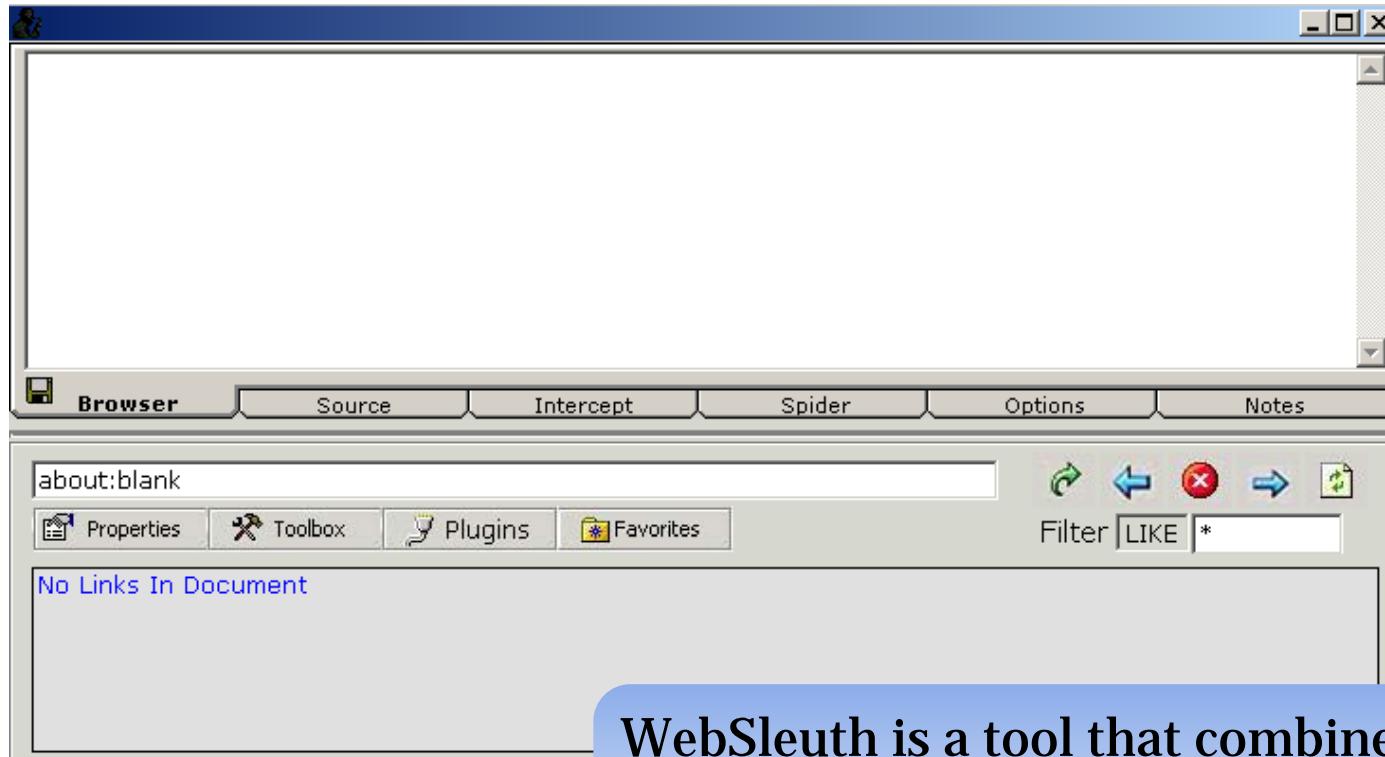
Both http and ftp retrievals can be time stamped, so Wget can see if the remote file has changed since the last retrieval and automatically retrieve the new version if it has

Source: www.gnu.org/

Wget: Screenshot



WebSleuth: Screenshot



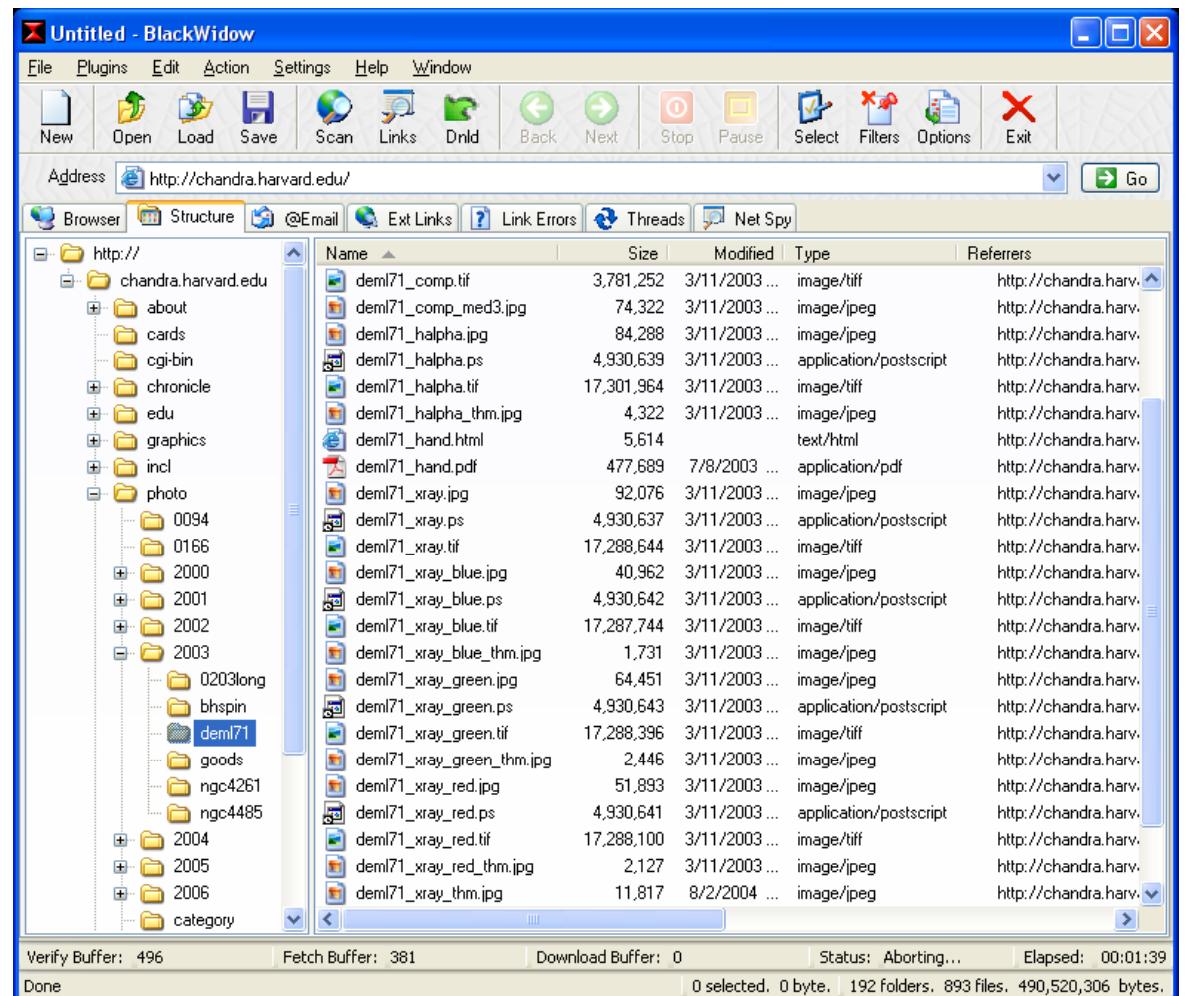
WebSleuth is a tool that combines spidering with the capability of a personal proxy such as Achilles

Picture Source: <http://sandsprite.com/sleuth/>

BlackWidow

Black widow is a website scanner, a site mapping tool, a site ripper, a site mirroring tool, and an offline browser program

It can be used to scan a site and create a complete profile of the site's structure, files, Email addresses, external links, and even link errors

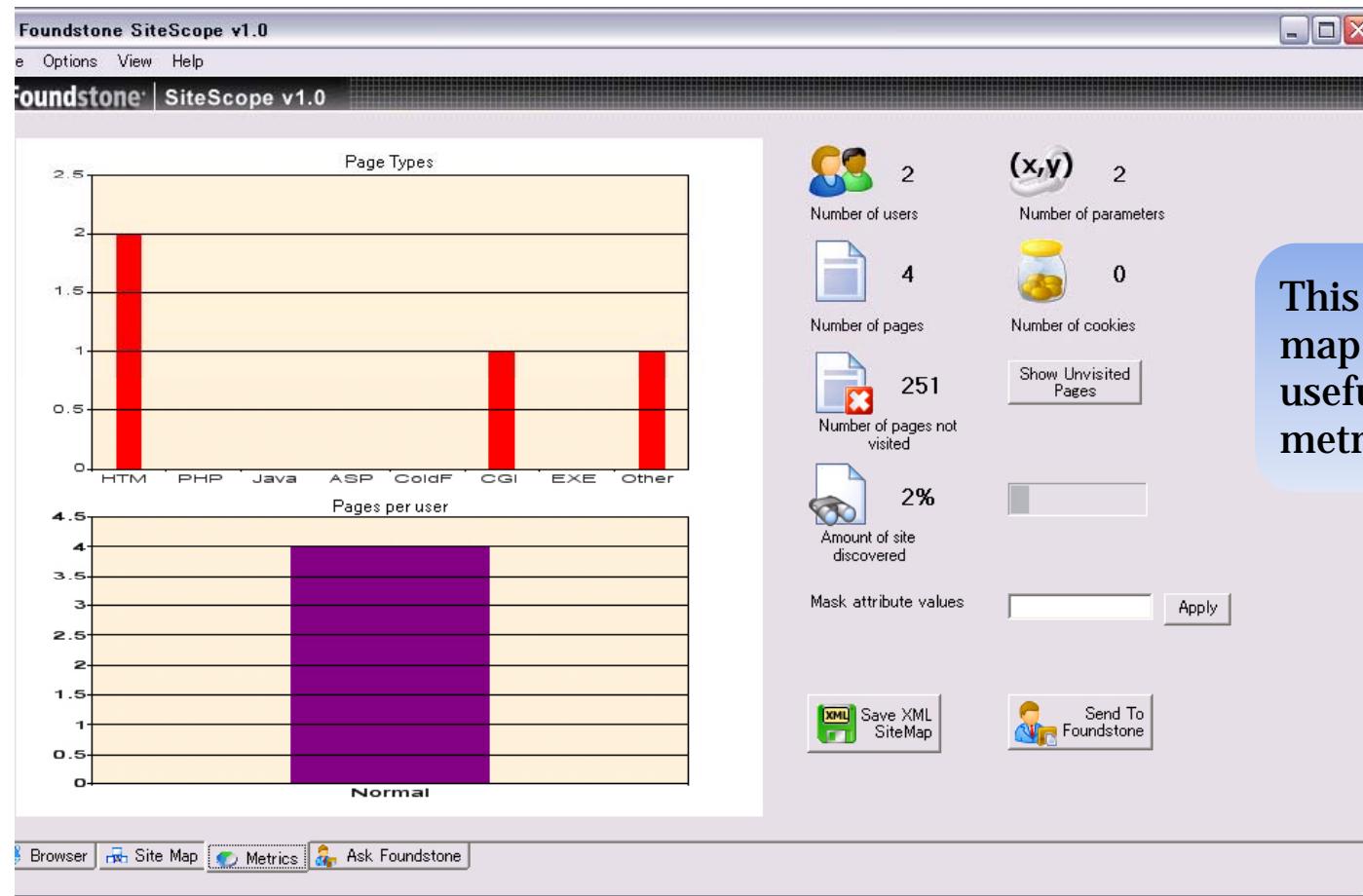


Source: <http://softbytelabs.com>



SiteScope Tool

Foundstone SiteScope is a free tool that helps website owners, developers, and managers to easily map out the navigation of a web application



This tool creates a site map and gathers useful data for basic metrics

WSDigger Tool – Web Services Testing Tool

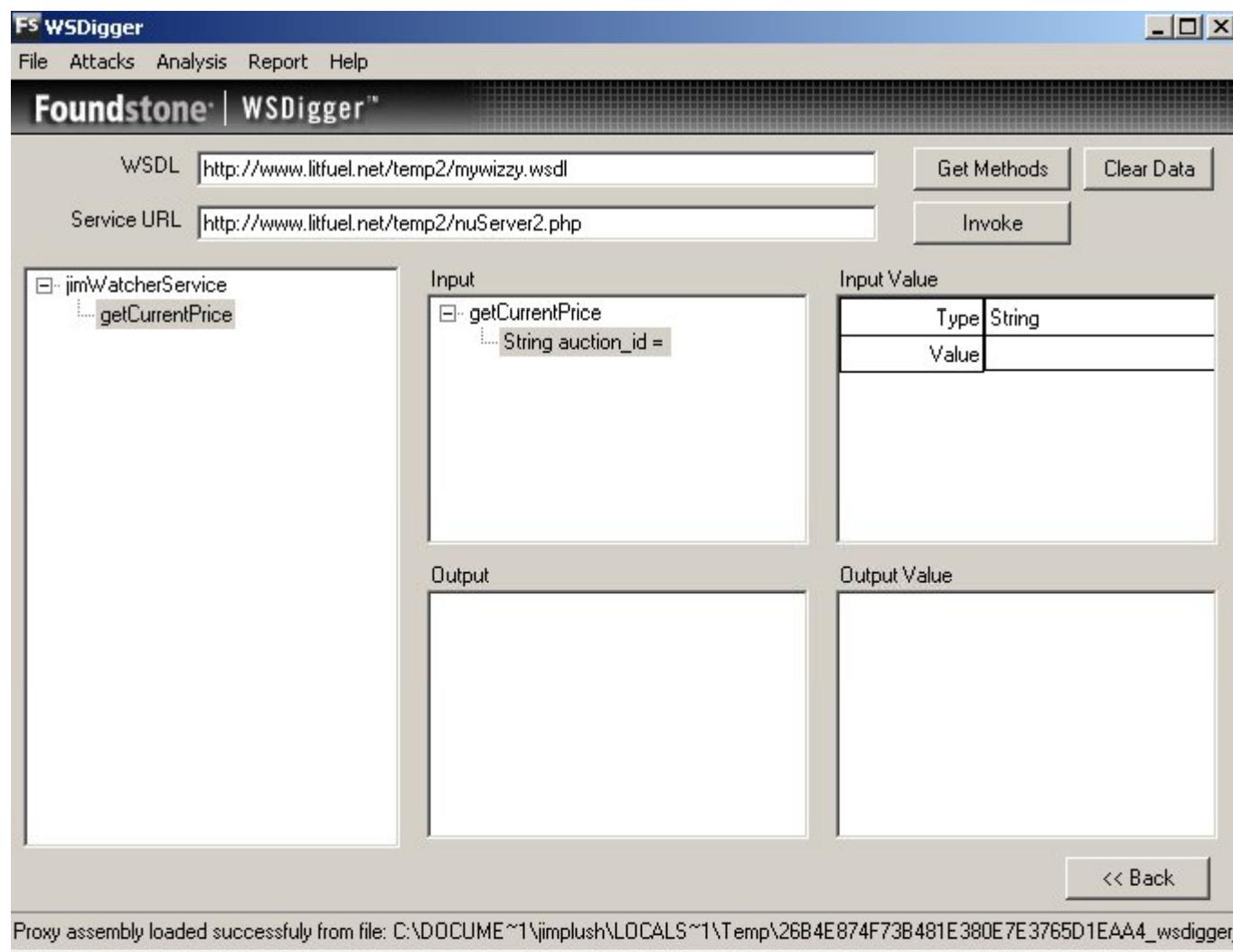
WSDigger is a free open source tool designed by Foundstone to automate black-box web services security testing

It is more than a tool; it is a web services testing framework

This framework contains sample attack plug-ins for SQL injection, cross site scripting, and XPATH injection attacks



WSDigger: Screenshot





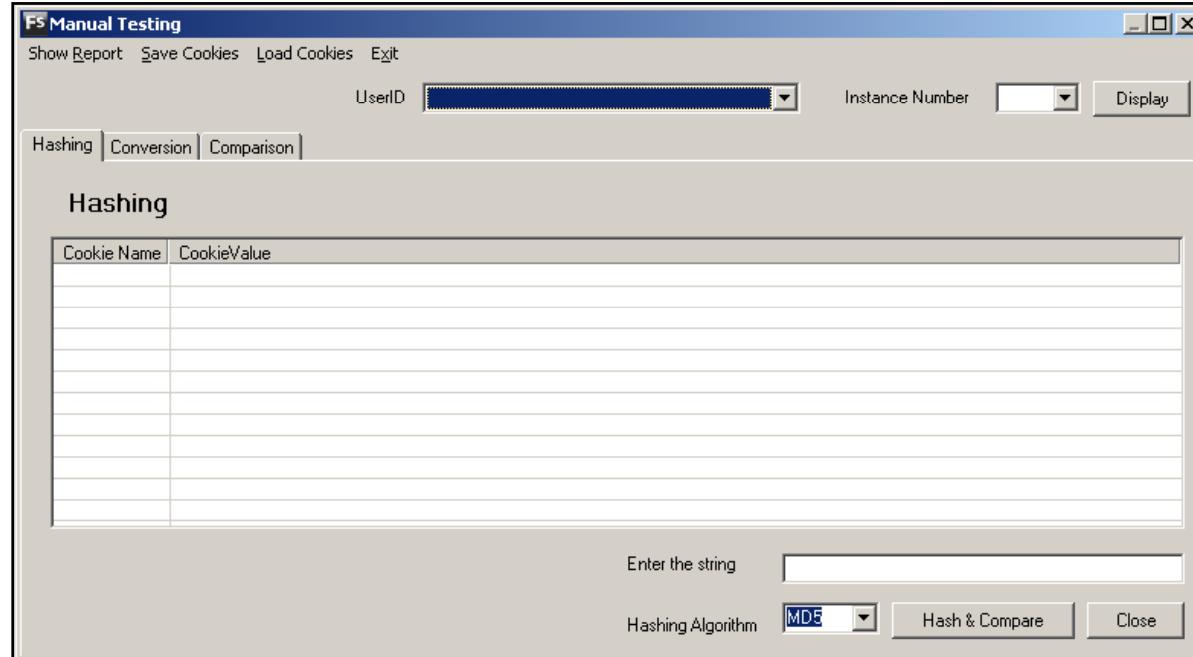
TM

CookieDigger Tool

CookieDigger helps identify weak cookie generation and insecure implementations of the session management by web applications

The tool works by collecting and analyzing cookies issued by a web application for multiple users

The tool reports on the predictability and entropy of the cookie and whether critical information, such as user name and password, are included in the cookie values



SSL Digger Tool

SSLDigger is a tool to assess the strength of SSL servers by testing the supported ciphers

Some of these ciphers are known to be insecure

The screenshot shows the Foundstone SSLDigger interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Help', and a 'Logout' button. Below it is a toolbar with icons for 'Scan', 'Report', 'Help', and 'About'. The main window has a title bar 'Foundstone | SSLDigger™'. An address bar shows 'C:\Documents and Settings\Administrator\WINDOWS\Desktop\SSLDigger.html' with a 'Go' button. A large text area displays 'Ciphers Supported' followed by a table:

Server URL	No Security	Weak Security	Strong Security	Excellent Security	Grade
https://www.eccouncil.org	0	5	3	0	B

Below this is a section titled 'Detailed Results' which is currently empty. At the bottom, there's a table titled 'SSL Certificate Details' with the following data:

Parameter	Value
Server	www.eccouncil.org
SSL Version	TLS 1.0
Key Algorithm	RSA_MD5
Key Algorithm Parameters	0500
Key Length	1024

Hacking Tool: WindowBomb

An email sent with this html code attached will create pop-up windows until the PC's memory gets exhausted

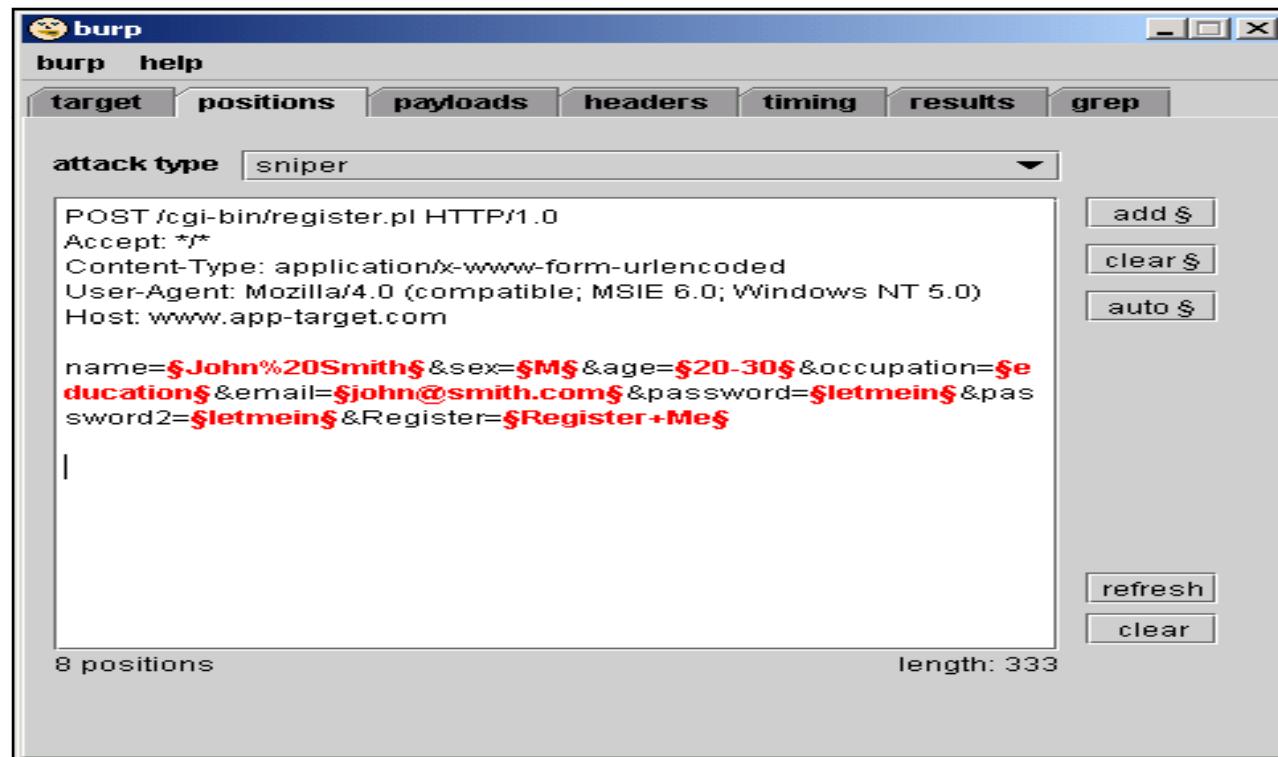
JavaScript is vulnerable to simple coding such as the example given below:

```
<HTML>
<HEAD>
<TITLE>WARNING [] INFECTING VIRUS </TITLE>
</HEAD>
<BODY ca load = "WindowBomb ()">
<SCRIPT LANGUAGE= "Java Script">
{ FUNCTION WINDOW BOMB }
{ var : counter =0 // dummy counter
    while (true)
{
    window open {"http://www.netscape.com",
"CRUSHING"
        + : counter, width=1, height=1,
resizable=90}
: counter ++
}
{/SCRIPT}
</BODY>
</HTML>
```



Burp: Positioning Payloads

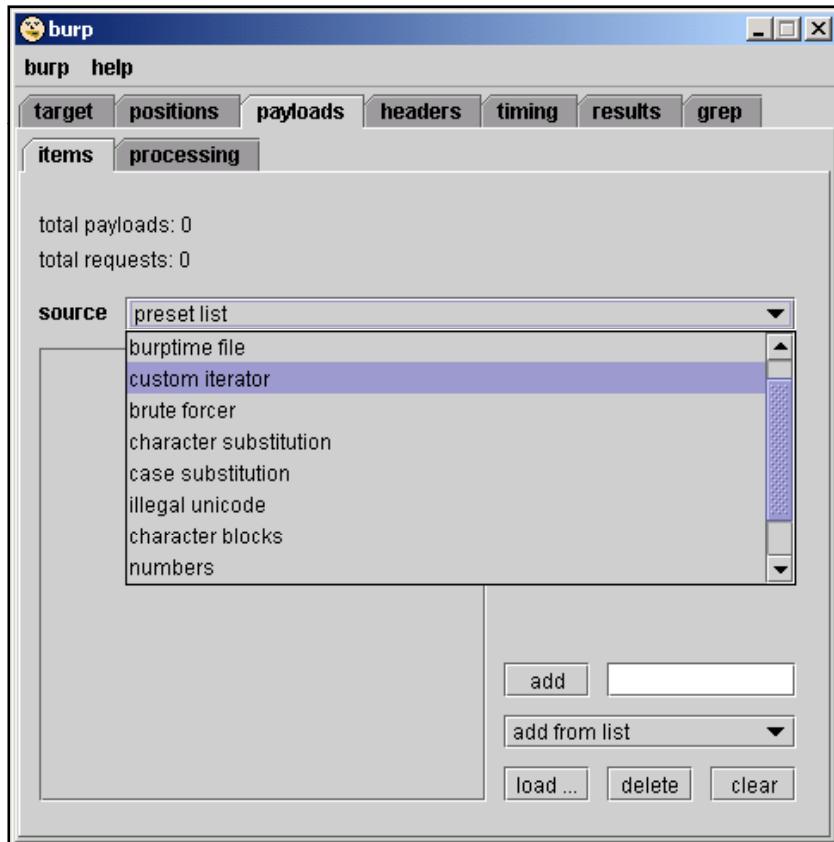
Burp is a tool for performing automated attacks against web-enabled applications



Source: <http://portswigger.net>

Burp: Configuring Payloads and Content Enumeration

Burp comes preconfigured with attack payloads and it can check for common databases on a Lotus Domino server



The screenshot shows the Burp results window. At the top, there are tabs: burp, save, view, tools, and help (which is selected). Below the tabs, there are two tabs: results (selected) and settings. The main area is a table with the following columns: request, payload, response, length, and "not authorized". The table contains 16 rows of data:

request	payload	response	length	"not authorized"
1 statrep.nsf		HTTP/1.1 200 OK	2133	<input type="checkbox"/>
2 schema.nsf		HTTP/1.1 404 Not Found	527	<input type="checkbox"/>
3 reports.nsf		HTTP/1.1 200 OK	2374	<input checked="" type="checkbox"/>
4 names.nsf		HTTP/1.1 200 OK	832	<input type="checkbox"/>
5 log.nsf		HTTP/1.1 200 OK	1681	<input type="checkbox"/>
6 events4.nsf		HTTP/1.1 200 OK	2338	<input type="checkbox"/>
7 doladmin.nsf		HTTP/1.1 200 OK	2375	<input checked="" type="checkbox"/>
8 dbdirman.nsf		HTTP/1.1 404 Not Found	527	<input type="checkbox"/>
9 certsv.nsf		HTTP/1.1 200 OK	2374	<input checked="" type="checkbox"/>
10 certlog.nsf		HTTP/1.1 200 OK	2374	<input checked="" type="checkbox"/>
11 admin4.nsf		HTTP/1.1 200 OK	1616	<input type="checkbox"/>
12 catalog.nsf		HTTP/1.1 200 OK	2123	<input type="checkbox"/>
13 domlog.nsf		HTTP/1.1 200 OK	621	<input type="checkbox"/>
14 bookmark.nsf		HTTP/1.1 200 OK	525	<input type="checkbox"/>
15 domcfg.nsf		HTTP/1.1 200 OK	516	<input type="checkbox"/>
16 connect.nsf		HTTP/1.1 404 Not Found	527	<input type="checkbox"/>

Burp: Password Guessing

Burp can be used for password guessing as well as data mining

The screenshot shows the Burp Suite interface with the "results" tab selected. The table displays a series of requests, each with a unique ID, payload, response status, error status, timeout, length, and a column for "login incorrect". Most responses are "HTTP/1.0 200 Ok" with a length of 3733, except for one at the top which is "HTTP/1.0 200 OK" with a length of 757. The "login incorrect" column contains several checked boxes.

request	payload	response	error	timeout	length	"login incorrect"
7091 favella		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7092 favella		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7093 favisms		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7094 favored		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7095 favorer		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7096 favours		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7097 favuses		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7098 fawners		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7099 fawnier		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7100 fawning		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7101 fazenda		HTTP/1.0 302 Object...	<input type="checkbox"/>	<input type="checkbox"/>	757	<input type="checkbox"/>
7102 fearers		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7103 fearful		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7104 fearing		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7105 feasing		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>
7106 feasted		HTTP/1.0 200 Ok	<input type="checkbox"/>	<input type="checkbox"/>	3733	<input checked="" type="checkbox"/>

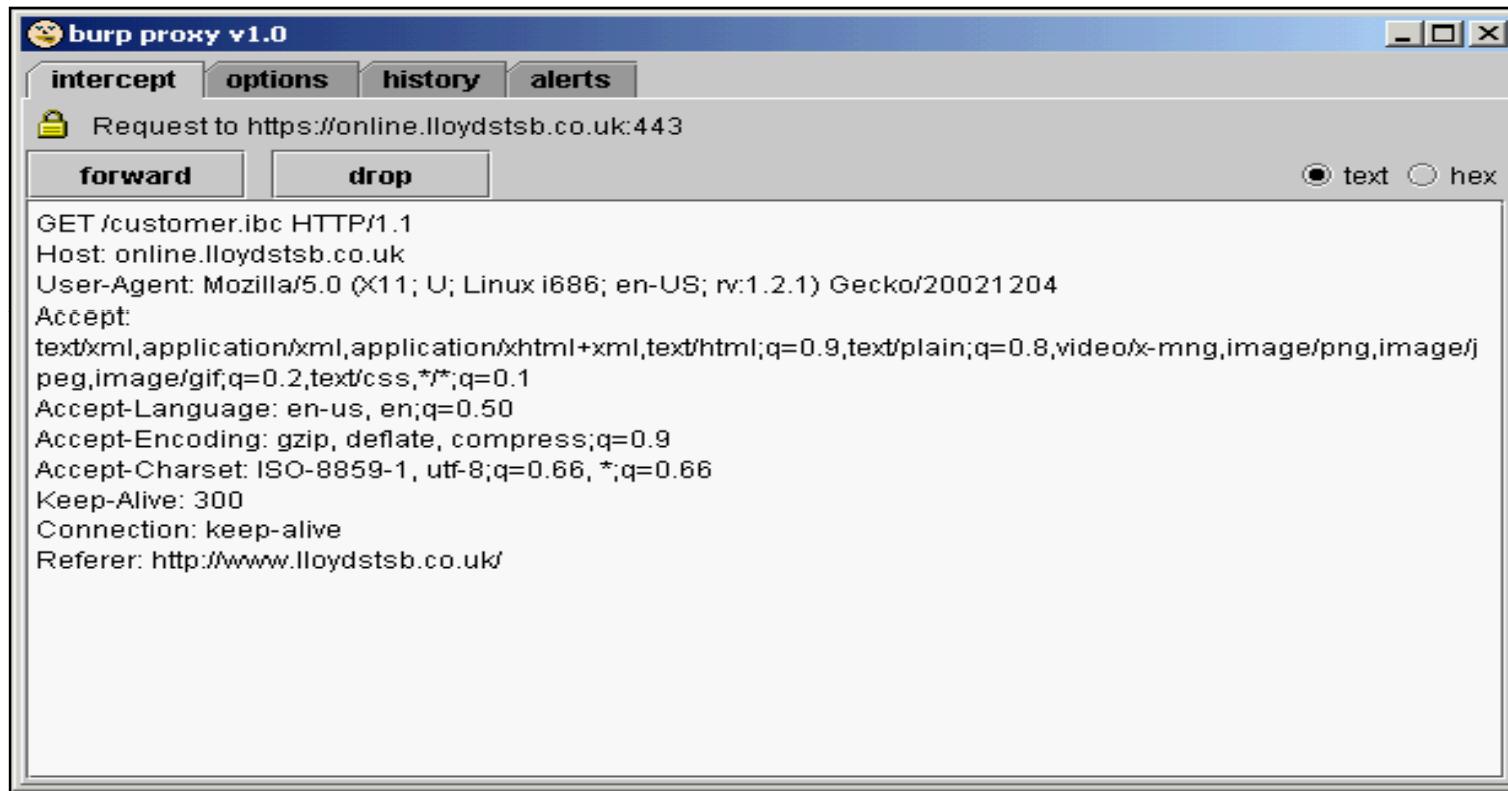
progress: 18%

The screenshot shows the Burp Suite interface with the "results" tab selected. The table displays a series of requests, each with a unique ID, payload, response status, error status, length, and two columns for "name=username" and "name=password". The "name=username" column contains names like susanlt, dthomas, rbentley, nicholasw, des, richardx, and johnnevile. The "name=password" column contains passwords like monday44, godfather, chueh219, password, gateway, richardx, infamy, firewall, and teacher. Most responses are "HTTP/1.0 200 Ok" with a length of 1610, except for one at the top which is "HTTP/1.0 404 Not found" with a length of 195.

request	payload	response	length	name="username" va...	name="password" va...
285 0384		HTTP/1.0 200 Ok	1610	susanlt	monday44
286 0385		HTTP/1.0 404 Not found	195		
287 0386		HTTP/1.0 200 Ok	1611	dthomas	godfather
288 0387		HTTP/1.0 200 Ok	1611	rbentley	chueh219
289 0388		HTTP/1.0 200 Ok	1612	nicholasw	password
290 0389		HTTP/1.0 200 Ok	1605	des	gateway
291 0390		HTTP/1.0 404 Not found	195		
292 0391		HTTP/1.0 404 Not found	195		
293 0392		HTTP/1.0 200 Ok	1611	richardx	richardx
294 0393		HTTP/1.0 404 Not found	195		
295 0394		HTTP/1.0 404 Not found	195		
296 0395		HTTP/1.0 404 Not found	195		
297 0396		HTTP/1.0 200 Ok	1614	administrator	infamy
298 0397		HTTP/1.0 200 Ok	1617	administrator2	firewall
299 0398		HTTP/1.0 200 Ok	1613	johnnevile	teacher
300 0399		HTTP/1.0 404 Not found	195		

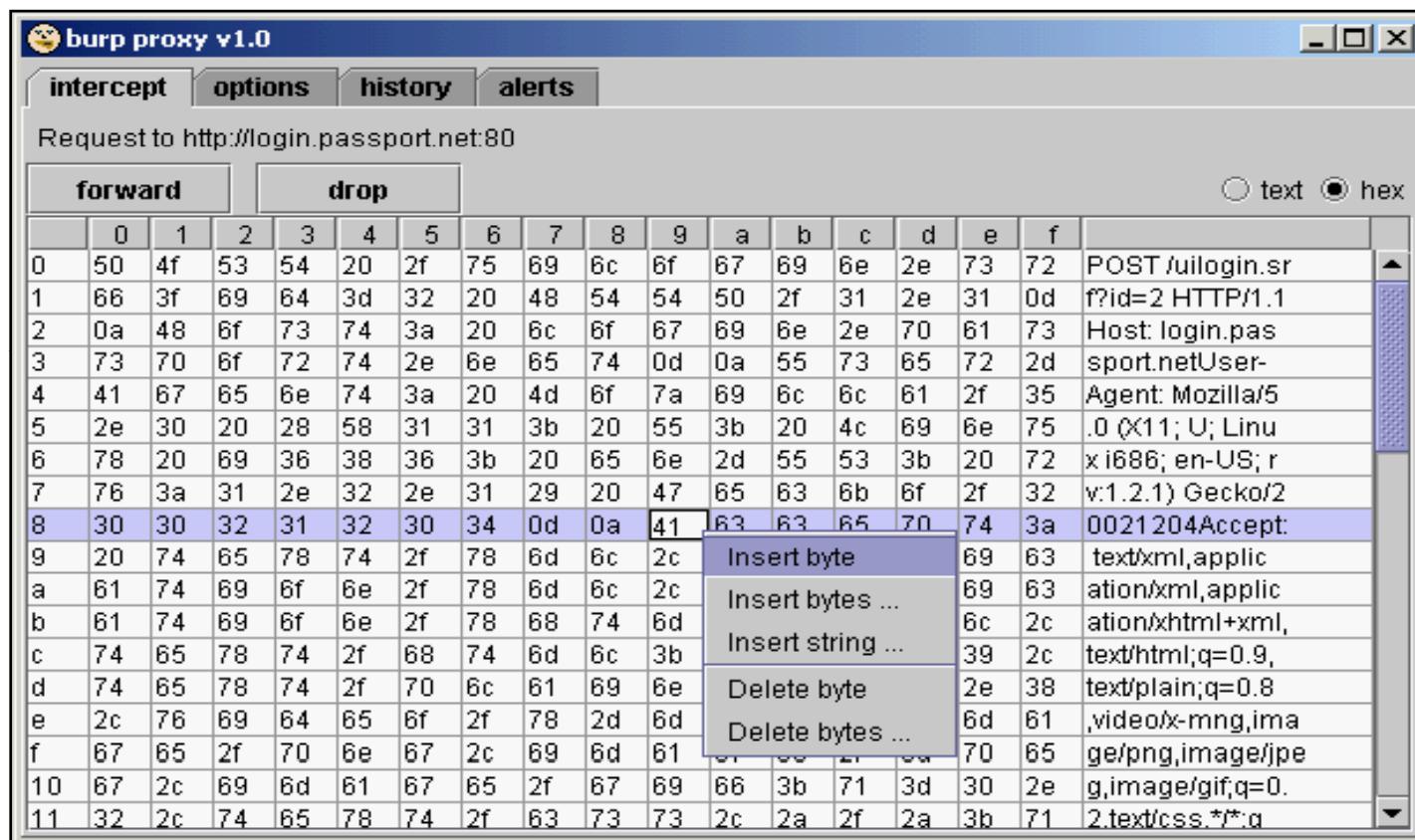
progress: 100%

Burp Proxy: Intercepting HTTP/S Traffic



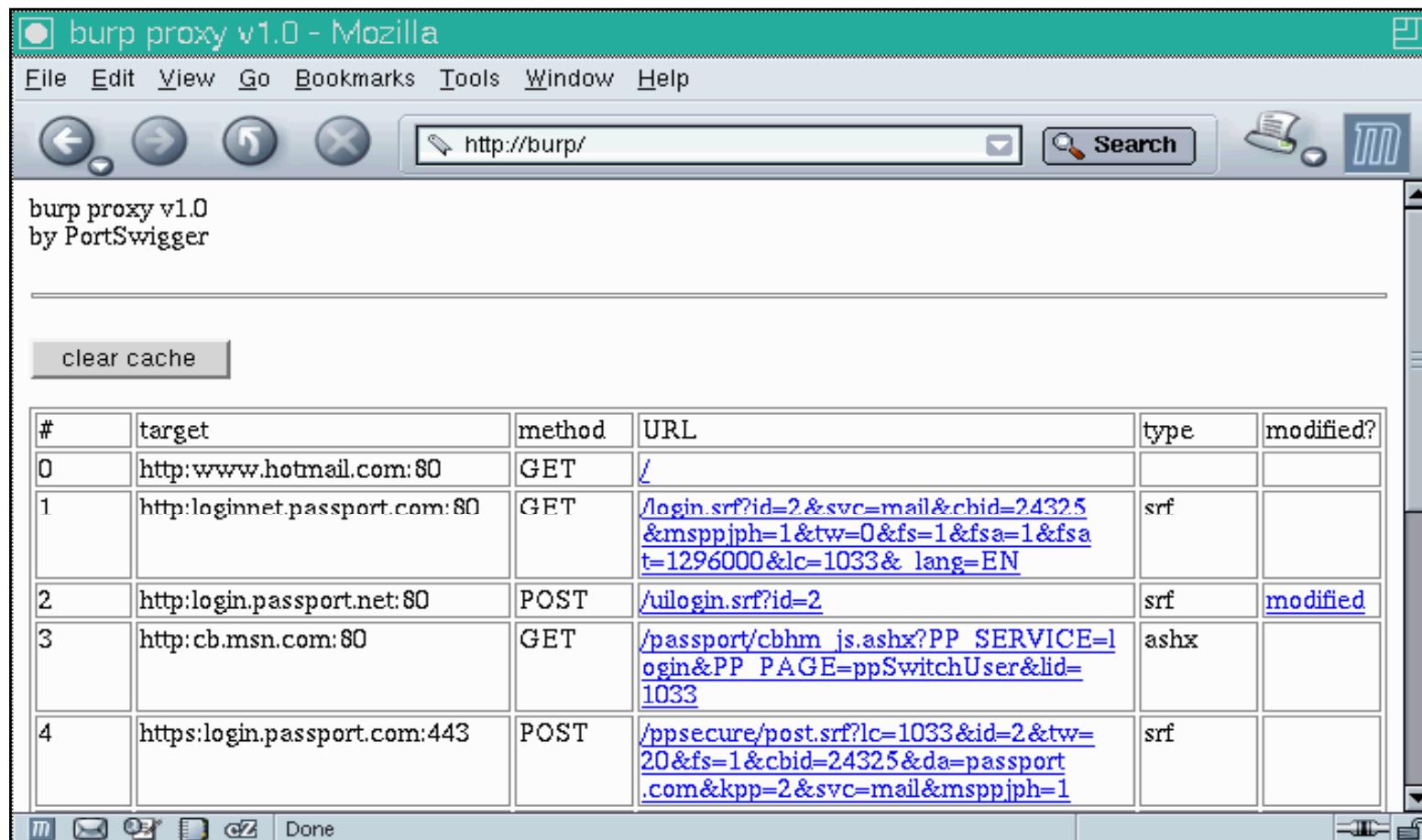
Burp proxy operates as a man-in-the-middle between the end browser and the target web server, and allows the attacker to intercept, inspect, and modify the raw traffic passing in both directions

Burp Proxy: Hex-editing of Intercepted Traffic



Burp proxy allows the attacker to modify intercepted traffic in both text and hexadecimal form; so even transfers of binary data can be manipulated

Burp Proxy: Browser Access to Request History



Burp proxy maintains a complete history of every request sent by the browser



TM

Tool: Burpsuite

Burp suite is an integrated platform for attacking web applications

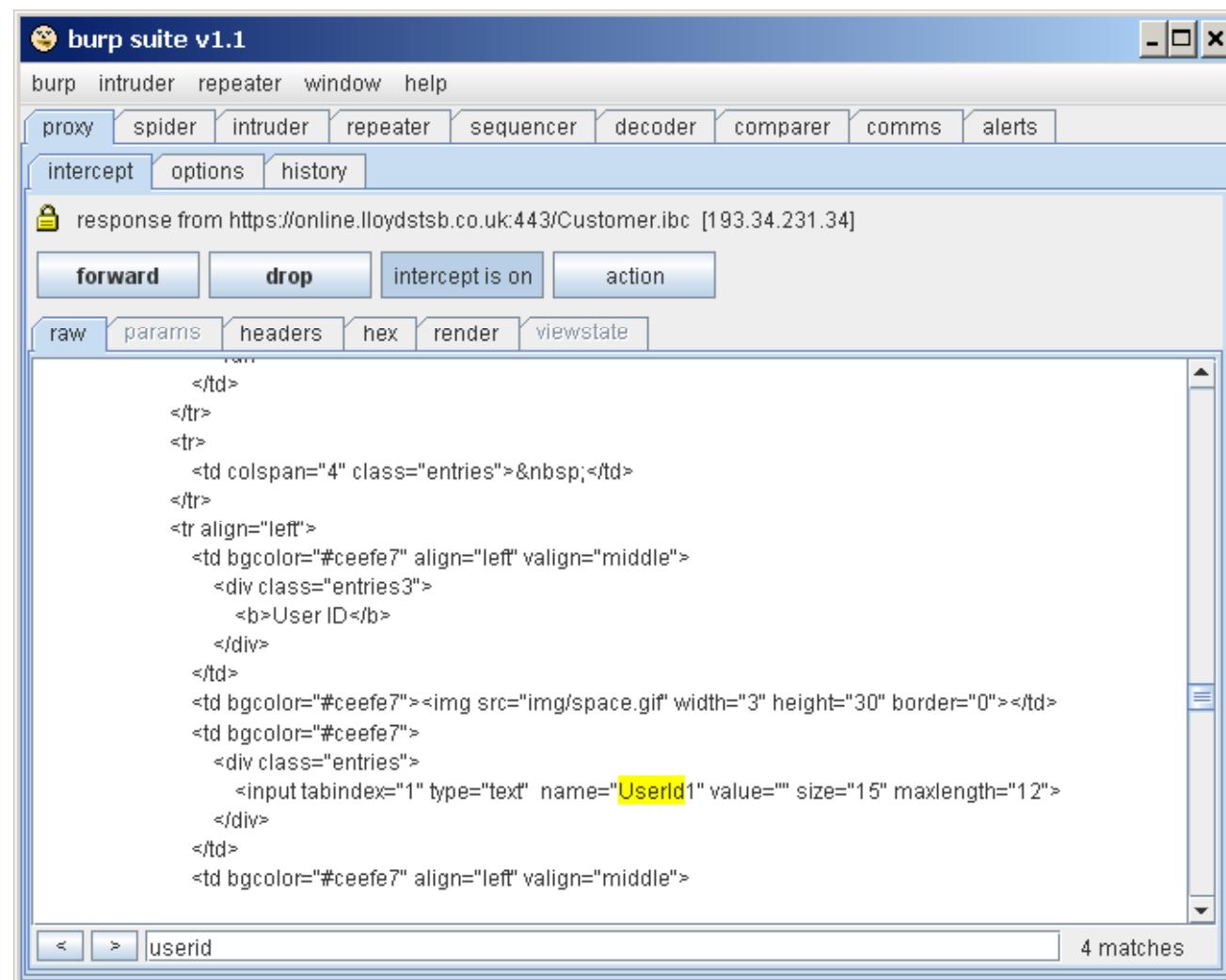
It allows an attacker to combine manual and automated techniques to enumerate, analyze, attack, and exploit web applications

The various burp tools work together effectively to share information and allow findings identified within one tool to form the basis of an attack using another

Key features include:

- Ability to passively spider an application in a non-intrusive manner
- One-click transfer of interesting requests between plug-ins, e.g. from proxy request history, or a web page form enumerated with burp spider
- Extensibility via IBurpExtender interface, which allows third-party code to extend functionality of burp suite
- Centrally configured settings for downstream proxies, web and proxy authentication, and logging
- Plug-ins can run in a single tabbed window, or be detached in individual windows
- All plug-ins and suite configuration is optionally persistent across program loads
- Runs in both Linux and Windows

Burpsuite: Screenshot 1





TM

Burpsuite: Screenshot 2

The screenshot shows the Burp Suite interface version 1.1. The main window displays a table of network requests. The columns include target, method, URL, type, SSL, IP, status, length, and cookies. Several requests from 'ecx.images-amazon.com' are listed, along with a request to 'amazon.co.uk:80'. A context menu is open over the last row, showing options like 'GET request to http://amazon.co.uk/s/ref=nb_ss_w_h_?url=search...'. The menu also includes 'send to intruder', 'send to repeater', 'send to spider', 'send to sequencer', 'send request to comparer', 'send response to comparer', 'copy URL to clipboard', and 'copy all URLs to clipboard'. The status bar at the bottom right shows '11_inst...'.

target	method	URL	type	SSL	IP	status	length	cookies
ecx.images-amazon.com...	GET	/images/l/116ofzeUJwL.jpg	jpg		216.38.160.117	200	4,108	
ecx.images-amazon.com...	GET	/images/l/21H3zFHMNzL.jpg	jpg		216.38.160.117	200	5,868	
ecx.images-amazon.com...	GET	/images/l/21N1rTEFcgl.jpg	jpg		216.38.160.117	304	650	
ecx.images-amazon.com...	GET	/images/l/21N1rTEFcgl.jpg	jpg		216.38.160.117	304	490	
ecx.images-amazon.com...	GET	/images/l/21XHFghSzeL.jpg	jpg		216.38.160.117	304	373	
ecx.images-amazon.com...	GET	/images/l/21XHFghSzeL.jpg	jpg		216.38.160.117	304	426	
amazon.co.uk:80	GET	/s/ref=nb_ss_w_h_?url=search-alias%3Daps&field-...			87.238.81.129	200	121,454	at-acb...
z-e	GET request to http://amazon.co.uk/s/ref=nb_ss_w_h_?url=search-alias%3Daps&field-keywords=web+application+hacker%27s+handbook							
z-e	send to intruder							
z-e	send to repeater							
z-e	send to spider							
z-e	send to sequencer							
ec:	send request to comparer							
ec:	send response to comparer							
ad:	copy URL to clipboard							
ad:	copy all URLs to clipboard							
ad.yieldmanager.com:80	GET	/imp?Z=160x600&s=225722&_salt=120503245&B=...			208.67.66.11	302	3,355	fl_inst...
ad.uk.doubleclick.net:80	GET	/adj/N3434.EType/B2564937.4;sz=160x600;ord=119...	817%		209.62.178.57	200	4,396	
m.uk.2mdn.net:80	GET	/879366/flashwrite_1_2.js	js		216.73.84.103	304	161	
m.uk.2mdn.net:80	GET	/888452/160x600_cablecar_nov2007_uk.swf?clickT...	swf		216.73.84.103	200	14,275	
amazon.co.uk:80	GET	/favicon.ico	ico		87.238.81.129	200	1,679	



Hacking Tool: cURL

cURL is a multi-protocol transfer library

It is a client side URL transfer library supporting FTP, FTPS, HTTP, HTTPS, GOPHER, TELNET, DICT, FILE, and LDAP

cURL supports HTTPS certificates, HTTP POST, HTTP PUT, FTP uploading, Kerberos, HTTP form-based upload, proxies, cookies, user+password authentication, file transfer resume, http proxy tunneling, and more

Proof of Concept

Source: <http://curl.haxx.se>

Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited



TM

cURL: Screenshot

```
curl 7.10 (win32) libcurl/7.10
Usage: curl [options...] <url>
Options: <H> means HTTP/HTTPS only, <F> means FTP only
-a/--append      Append to target file when uploading <F>
-A/--user-agent <string> User-Agent to send to server <H>
-b/--cookie <name:string/file> Cookie string or file to read cookies from <H>
-B/--use-ascii   Use ASCII/text transfer
-c/--cookie-jar <file> Write all cookies to this file after operation <H>
-C/--continue-at <offset> Specify absolute resume offset
-d/--data <data>  HTTP POST data <H>
    --data-ascii <data>  HTTP POST ASCII data <H>
    --data-binary <data>  HTTP POST binary data <H>
    --disable-epsv  Prevents curl from using EPSV <F>
-D/--dump-header <file> Write the headers to this file
--egd-file <file> EGD socket path for random data <SSL>
-e/--referer     Referer page <H>
-E/--cert <cert[:passwd]> Specifies your certificate file and password <HTTPS>
    --cert-type <type> Specifies certificate file type <DER/PEM/ENG> <HTTPS>
    --key <key>        Specifies private key file <HTTPS>
    --key-type <type> Specifies private key file type <DER/PEM/ENG> <HTTPS>
    --pass <pass>      Specifies passphrase for the private key <HTTPS>
    --engine <eng>     Specifies the crypto engine to use <HTTPS>
    --cacert <file>    CA certificate to verify peer against <SSL>
    --capath <directory> CA directory (made using c_rehash) to verify
                        peer against <SSL, NOT Windows>
    --ciphers <list>   What SSL ciphers to use <SSL>
    --compressed       Request a compressed response (using deflate).
    --connect-timeout <seconds> Maximum time allowed for connection
    --crlf            Convert LF to CRLF in upload. Useful for MVS (OS/390)
-f/--fail          Fail silently (no output at all) on errors <H>
-F/--form <name=content> Specify HTTP POST data <H>
-g/--globoff       Disable URL sequences and ranges using <> and []
-G/--get           Send the -d data with a HTTP GET <H>
-h/--help          This help text
-H/--header <line> Custom header to pass to server. <H>
-i/--include       Include the HTTP-header in the output <H>
-I/--head          Fetch document info only <HTTP HEAD/FTP SIZE>
-j/--junk-session-cookies Ignore session cookies read from file <H>
    --interface <interface> Specify the interface to be used
```



TM

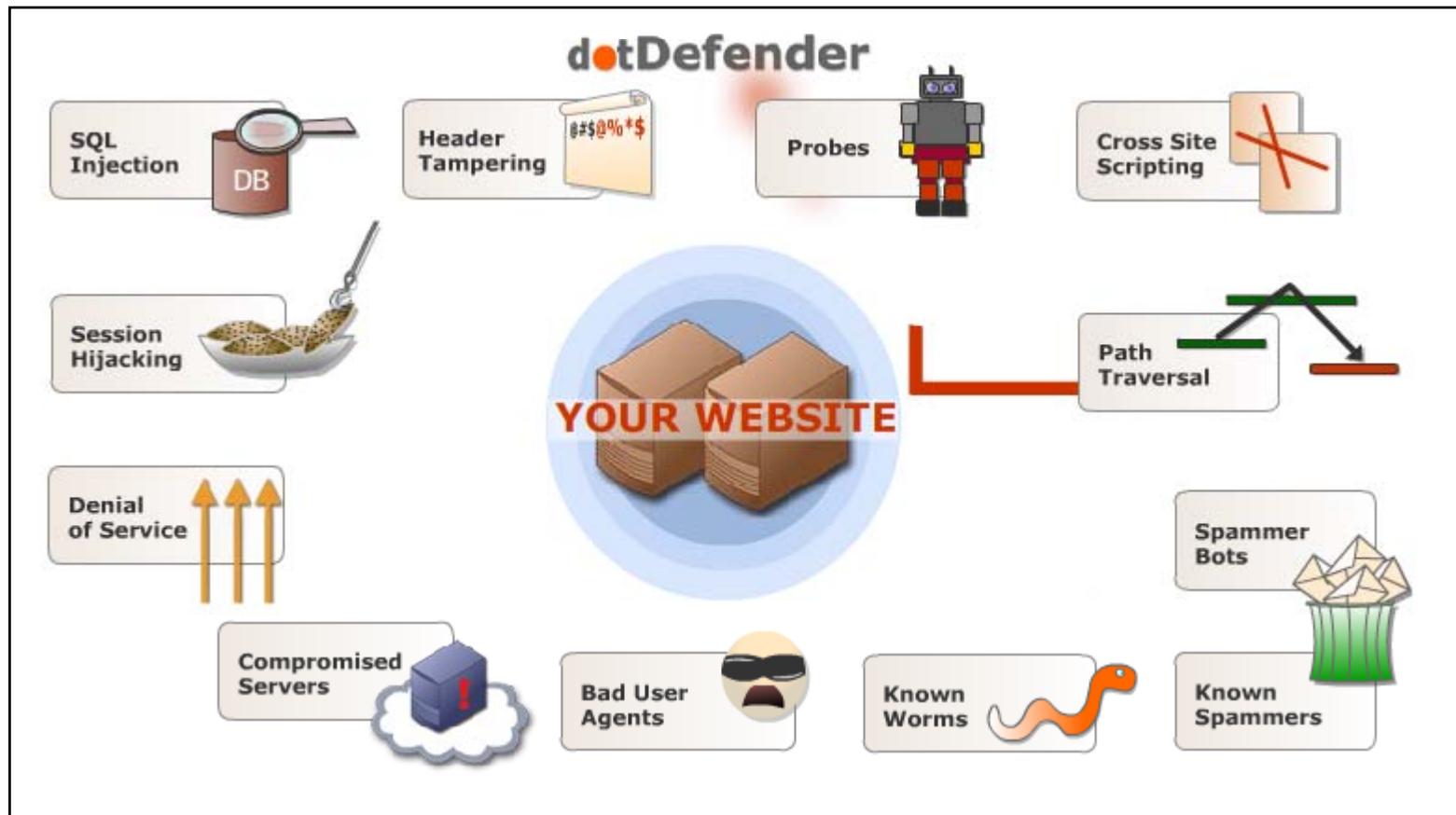
dotDefender

dotDefender is a web application attack protection tool that blocks attacks that are manifested within the HTTP request logic such as:

- **SQL Injection** - dotDefender intercepts and blocks attempts to inject SQL statements that corrupt or gain access to the corporate data
- **Proxy Takeover** - dotDefender intercepts and blocks attempts to divert traffic to an unauthorized site
- **Cross-site Scripting** - dotDefender intercepts and blocks attempts to inject malicious scripts that hijack the machines of subsequent site visitors
- **Header Tampering** - dotDefender identifies and blocks requests containing the corrupted header data
- **Path Traversal** - dotDefender blocks attempts to navigate through the host's internal file system
- **Probes** - dotDefender detects and blocks attempts to ferret the system's information
- **Known Attacks** - dotDefender recognizes and blocks attacks bearing known signatures

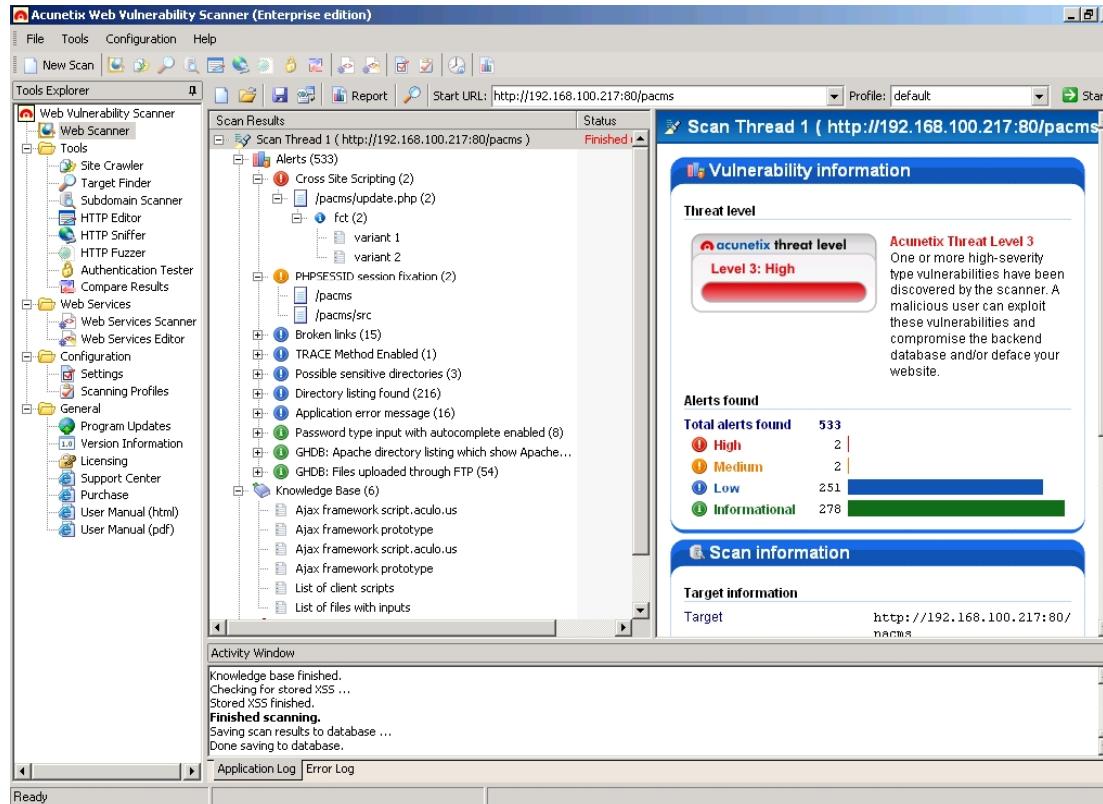
Source: <http://www.dotdefender.com>

dotDefender



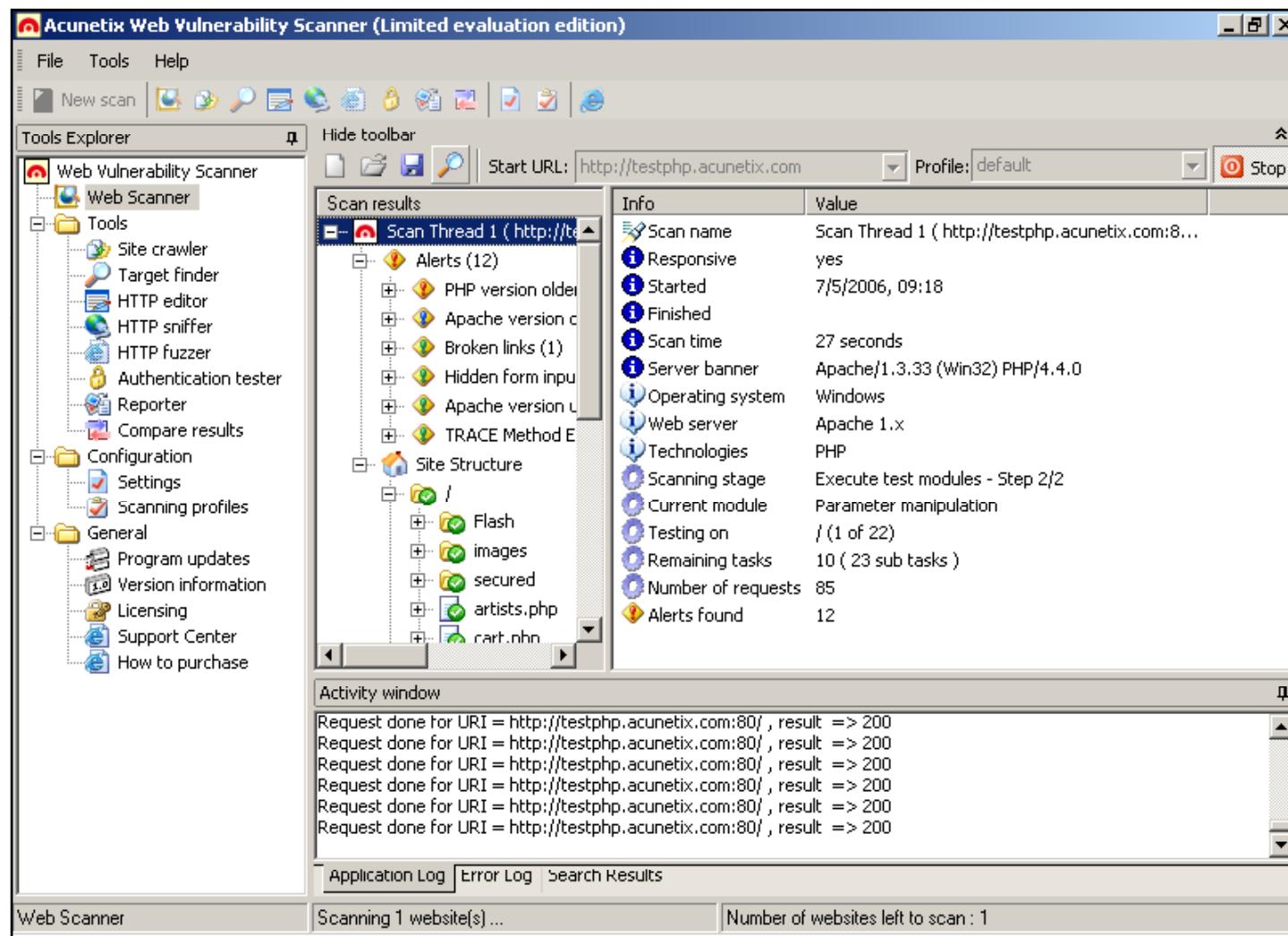
Acunetix Web Scanner

Acunetix launches all the Google hacking database queries onto the crawled content of your website, to find any sensitive data or exploitable targets before a “search engine hacker” does



Source: <http://www.acunetix.com>

Acunetix Web Scanner: Screenshot

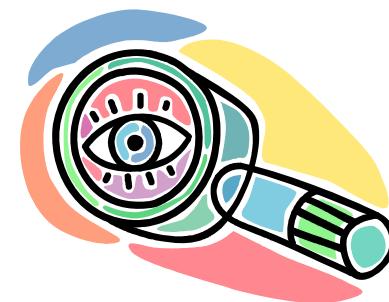


AppScan – Web Application Scanner

AppScan provides security testing throughout the application development lifecycle, which tests security assurance in the development stage

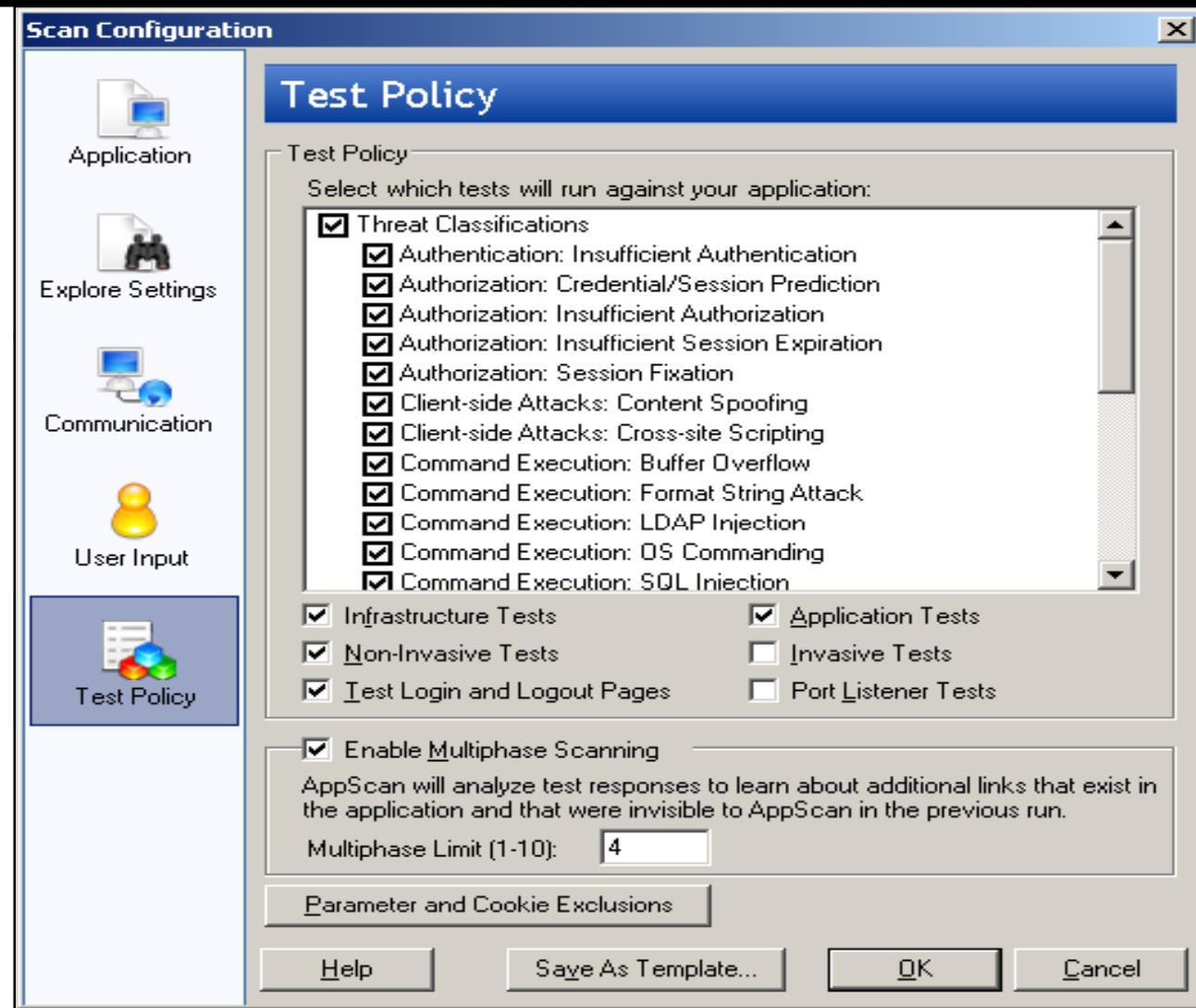
Vulnerability detects by simulating hacker attacks such as:

- Cross-Site Scripting
- HTTP Response Splitting
- Parameter Tampering
- Hidden Field Manipulation
- Backdoors/Debug Options
- Stealth Commanding
- Forceful Browsing
- Application Buffer Overflows
- Cookie Poisoning
- Third-party misconfigurations
- Known vulnerabilities
- HTTP Attacks
- SQL Injection
- Suspicious Content
- XML/SOAL Tests
- Content Spoofing
- LDAP Injection
- Session Fixation



Source: www.watchfire.com

AppScan: Screenshot





TM

AccessDiver

Certified Ethical Hacker

AccessDiver is a security tester for WEB sites. It incorporates a set of powerful features which help you find and organize failures and weaknesses from your web site. Here is a quick list of the features available:

- Contains fast security that uses up to 100 bots to do its analysis.
- Detects directory failures by comparing hundreds of known problems to your site.
- AccessDiver is fully proxy compliant and has a proxy analyzer (speed / anonymity) and a proxy hunter built-in.
- A built-in word leecher helps you increase the size of your dictionaries to expand and reinforce your analysis.
- A powerful task automizer manages your jobs transparently. You can tackle unrelated tasks while Accessdiver is working, saving you time.
- An on-the-fly word manipulator lets you increase the strength of your dictionaries easily when doing your analysis.
- A PING tester is included to tell you the efficiency of your site and the efficiency of an Internet address you would like to access.
- A DNS resolver lets you look up the host name of an IP address and reverse the process to learn an unknown host name.
- A feature called 'HTTP debugger' helps your understanding of how actual HTTP protocol works. It opens up the process so you see what really happens during a connection problem.
- A WHOIS gadget lets you retrieve owner information of a domain name (in case you would like to buy the domain or contact the actual owner).
- An update notifier automatically tells you when a new version of AccessDiver is available.
- And I welcome you to discover the other extremely features by yourself... :)
- A leeching system allows you to collect new proxies to make better proxy lists
- A proxy Hunting system allows you to get even more proxies by scanning IP regions...
- A file splitting and file merging system allows you to mixup files, or cut them in multiple parts. That's good to handle wordlists or proxylists.

Requirements

- A PC compatible machine
- Windows 9x,Me,2000,XP,2003
- 7 MB of space on your disk
- 192 MB of memory (RAM)
- Pentium 200 or higher
- Screen resolution : 800x600

Source: <http://www.accessdiver.com>



TM

AccessDiver: Screenshot

AccessDiver v4.270+

My Skill Tools Wordlist Weak History Help My other tool: HornyVision!!! Please make a donation to the author

flatus:flatus 6% 00:06:13

Test Basic-authentication Stop Server: http://www.juggyboy.com/ Test speed: 50 Bots

Dictionary History Settings Proxy Socks Search Progression

Bot	Last UserName tried	Last Password tried	Last Response received	Proxy used & (Country Code)
1	mjb1969	mjb1969	200 - OK	None
2	tor2000	tor	200 - OK	None
3	snatch	snatch	200 - OK	None
4	boy	wonder	200 - OK	None
5	tacobell	tacobell	200 - OK	None
6	bushman	bushman	200 - OK	None
7	suck	cock	200 - OK	None
8	edyenko	botong	200 - OK	None
9	forget	plaster	200 - OK	None
10	begood	begood	200 - OK	None
11	dba	dba	200 - OK	None
12	samj	samj	200 - OK	None
13	parton	dolly	200 - OK	None
14	jackpot	jackpot	200 - OK	None
15	666666	3333332000	200 - OK	None
16	kurt	123456	200 - OK	None
17	laying	holidays	200 - OK	None
18	ultimate	ultimate	200 - OK	None
19	jamestd	kelcie	200 - OK	None
20	curtis	curtis	200 - OK	None
21	monica	blewclintski	200 - OK	None
22	otiscat	otiscat	200 - OK	None
23	starfuck	starfuck	200 - OK	None
24	roxy	roxy	200 - OK	None
25	rosemary	rosemary2000	200 - OK	None
26	obi-wan	obi-wan	200 - OK	None
27	matthew	matt	200 - OK	None
28	walter123	filler	200 - OK	None

Found logins 44933 /hour

Weak logins: 0

Speed: 44933 /hour

Time spent: 00:00:27

Time left: 00:06:13

Attempts: 342

Attempt left: 4658

Tool: Falcove Web Vulnerability Scanner

Falcove is used by web-site owners to see whether their web sites are hackable or vulnerable to attacks

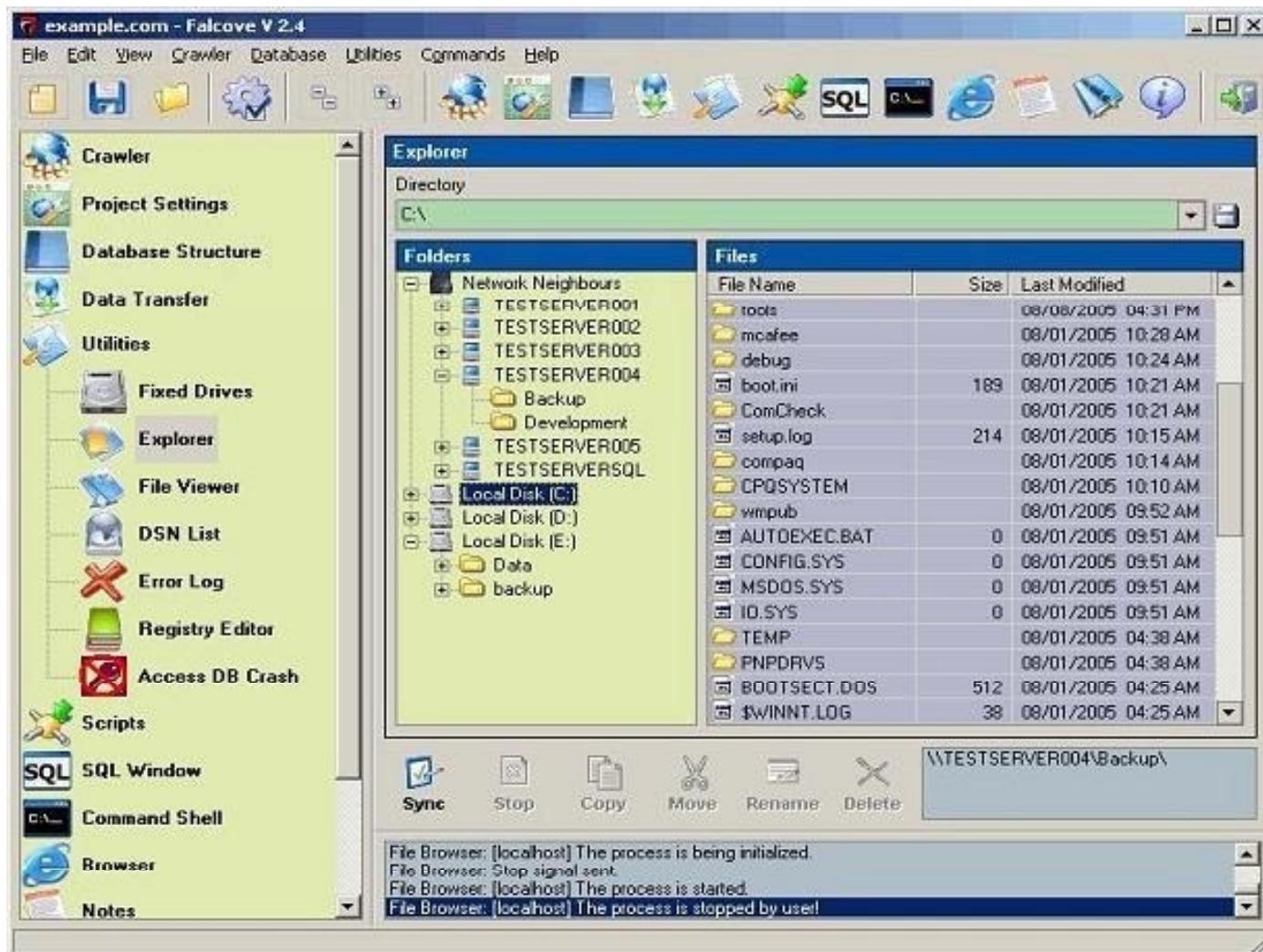
It finds vulnerabilities before hackers do and takes necessary precautions to implement the corrective actions

Features:

- Gives you an idea whether your website is secure against web attacks
- Crawler feature automatically checks for web vulnerabilities
- Audits all dynamic content including password fields, shopping carts, and other web applications
- Generates penetration reports that give you a certain idea about your websites' security level



Falcove Web Vulnerability Scanner: Screenshot



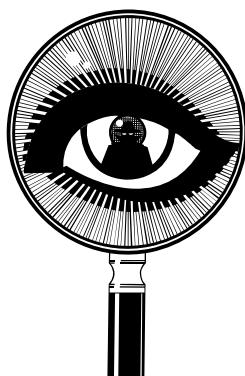
Tool: NetBrute

NetBrute scans a range of IP addresses for shared resources that have been shared via Microsoft File and Printer Sharing



It shows any SMB compatible shared resources (i.e. Samba Servers on a Unix/Linux machine)

It is used by system administrators or home users to see what types of resources are shared and to warn the computer users if any unsecured resources are displayed

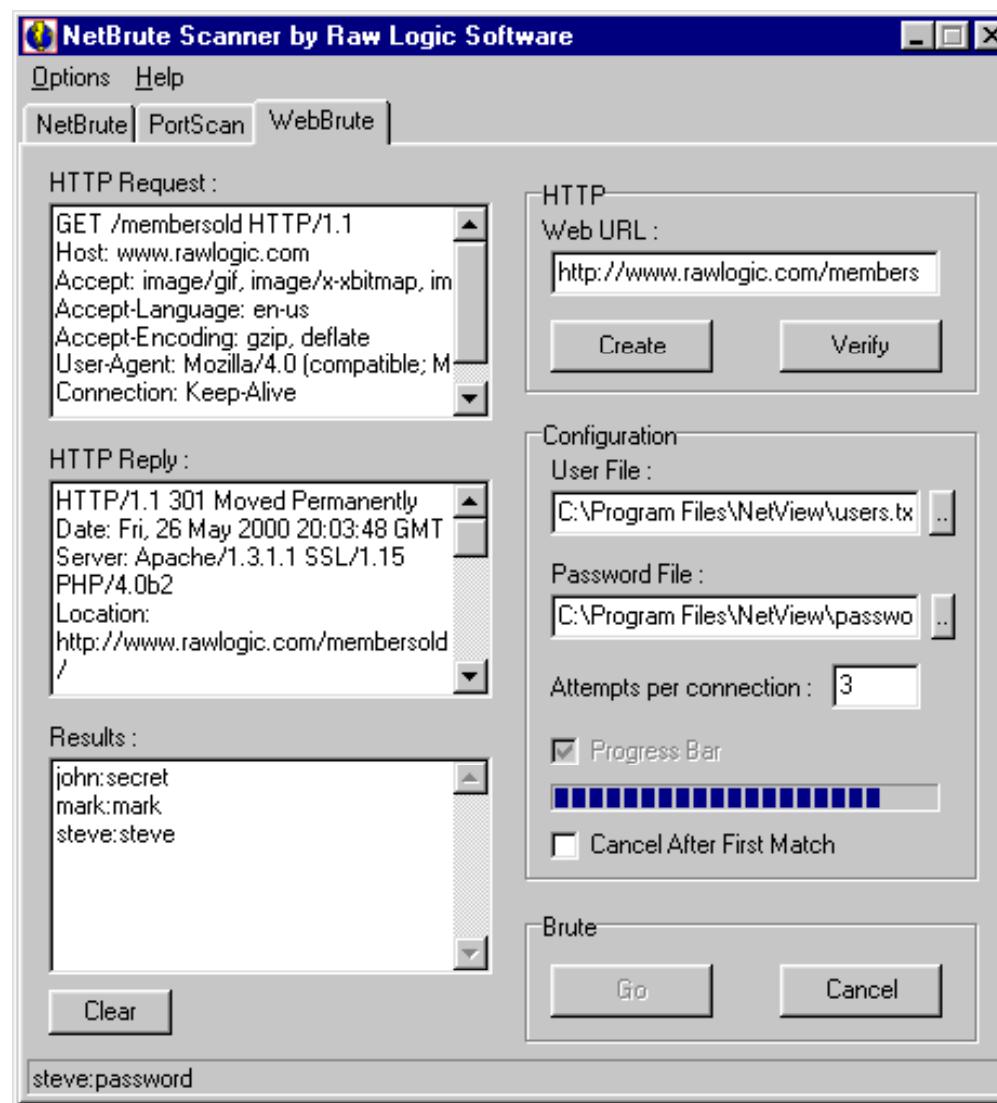


It finds all resources, whether they have passwords or not



TM

NetBrute: Screenshot



Tool: Emsa Web Monitor

Emsa web monitor is a small web monitoring program that runs on your desktop and allows the user to monitor uptime status of several websites

It works by periodically pinging the remote sites, and showing the ping time as well as a small graph that allows the user to quickly view recent monitoring history

It is rather simple but useful in monitoring a set of websites



Tool: KeepNI

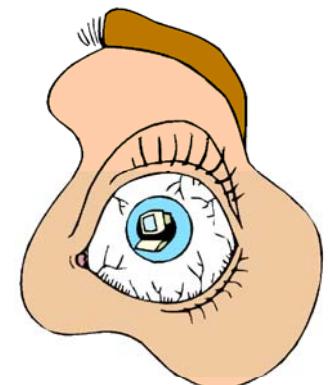
Keep an eye on your web site's functionality

It assures that your site is up and fully functional every time

Whenever a malfunction is detected, KeepNI immediately alerts you

KeepNI has an extensive logging facility to watch and alert

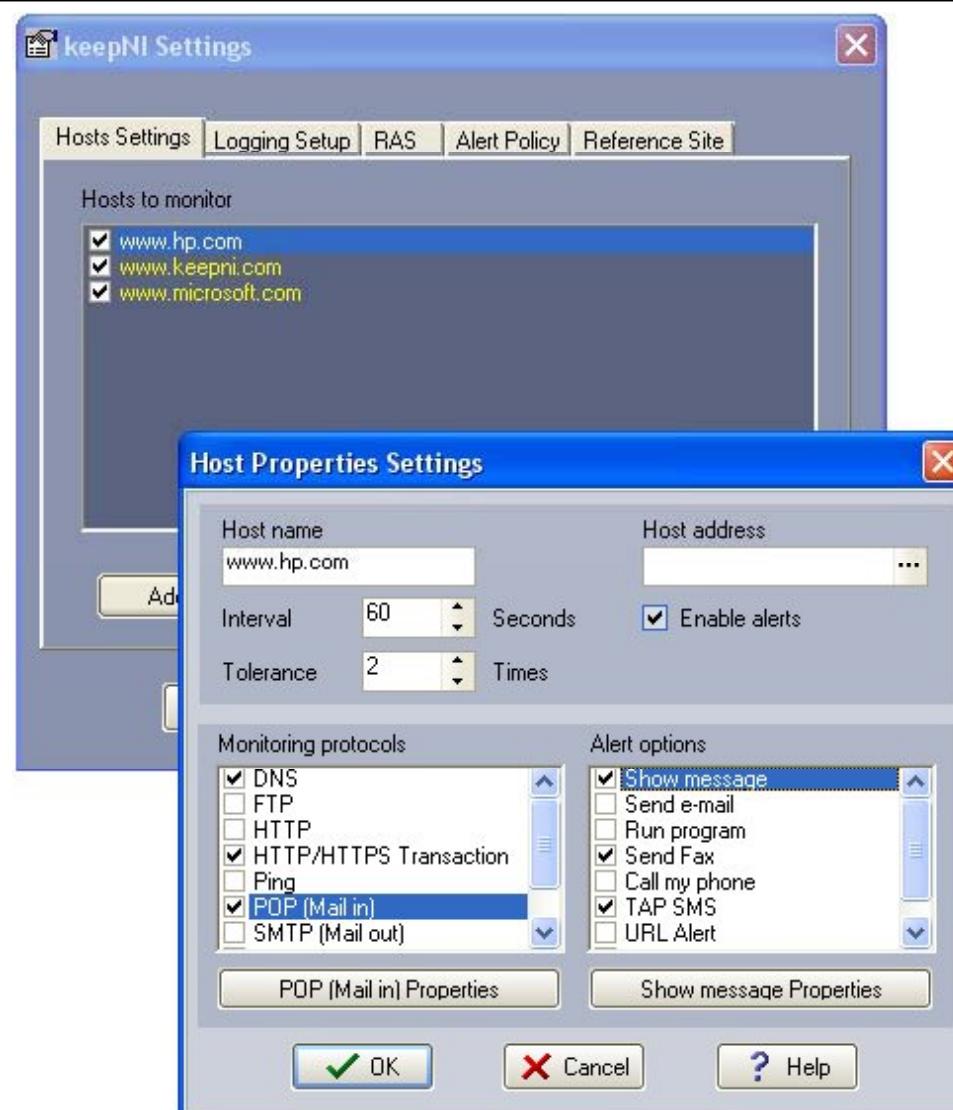
It logs and analyzes the collected data to present a full comprehensive view of your web site's performance





TM

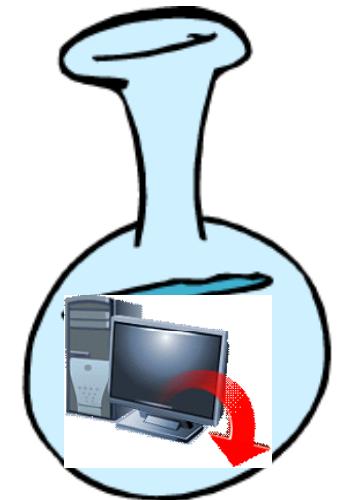
KeepNI: Screenshot



Tool: Parosproxy

Parosproxy is written in Java and useful for testing web applications and insecure sessions

Paros's proxy nature, all HTTP and HTTPS data between server and client, including cookies and form fields, can be intercepted and modified



The screenshot shows the Paros 3.2.3 - Untitled Session interface. The menu bar includes File, Edit, View, Scanner, Report, Tools, and Help. The main window has tabs for Sites, Request, Response, and Trap. The Request tab is selected, displaying the following details:

HTTP/1.1 301 Moved Permanently
Date: Mon, 15 Aug 2005 21:29:20 GMT
Server: Apache/1.3.31 (Unix) DAV/1.0.3 mod_fastcgi/2.4.2 mod_gzip/1.3.26.1a PHP/4.3.10 mod_ssl/2.8.19 OpenSSL/0.9.6c
Location: http://www.litfuel.net/plush/
Connection: close
Content-Type: text/html; charset=iso-8859-1

The Response tab shows the HTML content:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>301 Moved Permanently</TITLE>
</HEAD><BODY>
```

A "Raw View" dropdown is visible at the bottom left of the Response pane.



Parosproxy: Screenshot

A screenshot of the Parosproxy application window. The title bar reads "Paros 3.2.3 - Untitled Session". The menu bar includes File, Edit, View, Scanner, Report, Tools, and Help. The main interface has three tabs: Request, Response, and Trap. The Response tab is selected. On the left, there's a tree view under the "Sites" tab, showing several entries under "Sites": http://mitfuel.net, http://www.artlebedev.com, http://www.daskeyboard.com, http://www.litfuel.net, and http://www.thecandlestick.com. The right pane displays the HTTP response details and the raw HTML content.

HTTP/1.1 301 Moved Permanently
Date: Mon, 15 Aug 2005 21:29:20 GMT
Server: Apache/1.3.31 (Unix) DAV/1.0.3 mod_fastcgi/2.4.2 mod_gzip/1.3.26.1a PHP/4.3.10 mod_ssl/2.8.19 OpenSSL/0.9.6c
Location: http://www.litfuel.net/plush/
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>301 Moved Permanently</TITLE>
</HEAD><BODY>

Raw View ▾

Tool: WebScarab

WebScarab is a Java framework for analyzing applications that communicate using the HTTP and HTTPS protocols

It operates as an intercepting proxy, allowing operator to review and modify requests created by the browser before they are sent to the server and vice versa

WebScarab can intercept both HTTP and HTTPS communication

Operator can also review the conversations (requests and responses) that have passed through WebScarab



WebScarab: Screenshot 1

The screenshot shows the WebScarab application window. The menu bar includes File, View, Tools, and Help. The toolbar below the menu contains buttons for Summary, Message log, Proxy, Manual Request, WebServices, Spider, Extensions, SessionID Analysis, Scripted, Fragments, Fuzzer, and Compare. The main pane is titled "Summary" and displays a tree view of a conversation list. The tree shows a request to `http://www.owasp.org:80/` which resulted in a 301 Moved... response, followed by several GET requests for sub-resources like `banners/`, `images/`, `index.php/`, and `skins/`. Below the tree is a detailed table of the captured requests:

ID	Date	Method	Host	Path	Parameters	Status	Origin
5	2006/06/23...	GET	http://www.owasp.org:80	/skins/monobook/main....	?/?	200 OK	Proxy
4	2006/06/23...	GET	http://www.owasp.org:80	/skins/common/IEFixes...		200 OK	Proxy
3	2006/06/23...	GET	http://www.owasp.org:80	/skins/common/commo...		200 OK	Proxy
2	2006/06/23...	GET	http://www.owasp.org:80	/index.php/Main_Page		200 OK	Proxy
1	2006/06/23...	GET	http://www.owasp.org:80	/		301 Moved ...	Proxy

A progress bar at the bottom indicates "5.27 / 63.56".



TM

WebScarab: Screenshot 2

The screenshot shows the WebScarab application interface. At the top, it displays "WebScarab - conversation 1" and the URL "1 - GET http://www.owasp.org:80/ 301 Moved Permanently". Below this, there are two tabs: "Parsed" (selected) and "Raw".
Parsed View:
Method: GET URL: http://www.owasp.org:80/ Version: HTTP/1.0
Header:

Header	Val
Accept	image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application
Accept-Language	en-us
User-Agent	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
Host	www.owasp.org

Hex View:
Position: 0 1 2 3 4 5 6 7 8 9 A B C D E F String:

Raw View:
Version: HTTP/1.1 Status: 301 Message: Moved Permanently
Header:

Header	Val
Date	Fri, 23 Jun 2006 13:20:02 GMT



TM

WebScarab: Screenshot 3

The screenshot shows the WebScarab application window. The title bar reads "WebScarab". The menu bar includes "File", "View", "Tools", and "Help". The top navigation bar has tabs: "Summary", "Message log", "Proxy" (which is highlighted with a red box), "Manual Request", "WebServices", "Spider", "Extensions", "SessionID Analysis", "Scripted", "Fragments", "Fuzzer", and "Compare". Below the tabs, there's a toolbar with icons for "Proxy", "Listeners", "Manual Edit" (which is highlighted with a red box), "Bean Shell", and "Miscellaneous".

The main configuration area has two sections:

- Intercept requests :** **Case Sensitive Regular Expressions ?**
 - Methods:** GET, POST (selected), HEAD, PUT, DELETE, TRACE.
 - Include Paths matching :** `*`
 - Exclude paths matching :** `^.(gif|jpg|png|css|js|ico|swf)$`
- Intercept responses :**
Only MIME-Types matching : `text/*`

A progress bar at the bottom indicates "4.06 / 63.56".

Tool: Watchfire AppScan

Watchfire® AppScan® automates web application security audits to ensure the security and compliance of websites

Benefits:

- Fully outsourced web application vulnerability management
- Direct access to Watchfire security experts and industry best practices
- Best path to actionable data for web application's security management
- Dramatically reduces the learning curve and adoption time
- Shields against loss of knowledge related to turnover or reorganization





TM

Watchfire AppScan: Screenshot

The screenshot shows the IBM Rational AppScan 7.7 interface. The main window title is "Appscan 7.7 Altoro Mutual.scan - IBM Rational AppScan". The menu bar includes File, Edit, View, Scan, Tools, and Help. The toolbar contains icons for Scan Configuration, Scan Expert, Scan Log, Report, and Update.

The left sidebar has three sections: "View" (Security Issues, Remediation Tasks, Application Data), "My Application (63)" (listing files like comment.aspx, default.aspx, disclaimer.htm, feedback.aspx, search.aspx, subscribe.swf, survey_questions.aspx, bank (54) containing account.aspx, apply.aspx, customize.aspx, login.aspx, logout.aspx, main.aspx, querxpath.aspx), and "Issue Severity Gauge" (Total number of issues: 63, bar chart showing 23 High, 15 Medium, 18 Low, 7 Info).

The central pane displays a tree view of "63 Security Issues (264 variants) for 'My Application'". The issues are categorized by severity: Critical (Red), High (Orange), Medium (Yellow), and Low (Green). Examples of issues include Blind SQL Injection, Cross-Site Scripting, and SQL Injection.

The bottom right pane shows a "Variant Details" section for ID 9056. It details a change applied to the original request: setting the 'creditAccount' parameter's value to '>%22%27<img%20src%3d%22javascript:alert(34105)%22'. The "Reasoning" section explains that this makes the application vulnerable to a Cross-Site Scripting attack.

At the bottom, status bars show Visited URLs (46/46), Completed Tests (3955/3955), and 63 Security Issues.

Tool: WebWatchBot

WebWatchBot is a monitoring and analysis software for web sites and IP devices including Ping, HTTP, HTTPS, SMTP, POP3, FTP, Port, and DNS checks

It provides in-depth monitoring and alerting functionality as well as tools to analyze and visualize historical data with real-time charting and graphs

Additional features include an option to run as a Windows Service, customizable 3D charts with print support, SQL database storage, etc.



WebWatchBot: Screenshot

ExclamationSoft WebWatchBot - Enterprise - Application Mode - (*DEBUG *)

File Edit View Tools Reports Actions Help

New Properties Delete Run Now Run All Now Suspend Activate

Control Panel

Explorer

Watch Explorer

Filters:Status

- All
- Active
- Suspended
- Down
- Up

Filters:Types

- Ping
- HTTP
- HTTPS
- SMTP
- POP3
- FTP

Explorer

Scheduler

Dashboard

All

...	Trans...	Alar...	Watch...	Watch Item Name	Check Frequ...	Action	Last...	Avg ...	Time
...	No	UP	HTTP	amazon.co.uk	1 minute	*SUSP...	4063	4063	1
...	No	UP	HTTPS	bluecycle.com - HTPPS	1 minute	*SUSP...	4171	4171	1
...	No	Down	FTP	bluecycle.com FTP	10 minutes	*SUSP...	906	906	1
...	No	UP	HTTP	bluecycle.com New Fr...	1 minute	*SUSP...	2795	2795	1
...	No	UP	HTTP	bluecycle.com/	1 minute	*SUSP...	3861	3861	1
...	No	UP	HTTP	ebay	1 minute	*SUSP...	2936	2936	1
...	No	Up	Ping	Example #1_Ping	1 hour	*SUSP...	0	0	0

Log - amazon.co.uk

```
[10/28/04 19:35:09.703] Watch URL: http://www.amazon.co.uk
[10/28/04 19:35:09.703] Watch Type: HTTP
[10/28/04 19:35:09.703] Interval: 1 minute
[10/28/04 19:35:09.703] Times Checked: 1
[10/28/04 19:35:09.703] Times Failed: 0
[10/28/04 19:35:09.703] Failure Pct: 0.00%
[10/28/04 19:35:09.703] [DEBUG] m_hwndParentWnd: 4917528
[10/28/04 20:05:44.734] -----*[ End of Log: 10/28/2004
20:05:44] -----
[10/28/04 20:34:01.718] -----*[ WebWatchBot 3.0 Build 0 (DEBUG) Started
]-----
[10/28/04 20:34:01.718] Watch Name: amazon.co.uk
```

Real-Time Chart Output Log Response Time Chart

WebWatchBot is Ready

1 of 22 Selected

Ratproxy

Ratproxy is a semi-automated and largely passive web application security audit tool

It is designed specifically for an accurate and sensitive detection, and automatic annotation of potential problems

It is optimized for security-relevant design patterns based on the observation of existing, user-initiated traffic in complex web environments





How Does it Avoid False Positives?

For accurately reporting of problems and to reduce the number of false alarms, ratproxy has to consider the following points:

- What the declared and actually detected MIME type for the document is?
- How pages respond to having cookie-based authentication removed?
- Whether requests seem to contain non-trivial, sufficiently complex security tokens, or other mechanisms that may make the URL difficult to predict?
- Whether any non-trivial parts of the query are echoed back in the response, and in what context?
- Whether the interaction occurs on a boundary of a set of domains defined by runtime settings as the trusted environment subjected to the audit, and the rest of the world?



Screenshot

Ratproxy - security testing proxy - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Ratproxy audit report

Generated on: 2008/06/10 12:01
Input file: example.log

NOTE: Not all of the issues reported necessarily correspond to actual security flaws. Findings should be validated by manual testing and analysis where appropriate. When in doubt, contact the author.

Report risk and risk modifier designations:

LOW	to	HIGH
INFO		
ECHO	/	echo
PRED	/	pred
AUTH	/	auth

Issue urgency classification (composite of impact and identification accuracy)

Non-discriminatory entry for further analysis

Query parameters echoed back / not echoed in HTTP response, respectively

Request URL or query data likely is / is not predictable to third parties, respectively

Request requires / does not require cookie authentication, respectively

POST query with no XSRF protection [toggle]

Parameter-accepting POST requests that lack security tokens. Some POST requests change application state, and may be vulnerable to cross-site request forgery attacks.

- HIGH echo PRED AUTH POST** http://test.example.com:80/examples/res665 dispatcher - 200 [view trace]
Payload: reqReloadSnapshots
Response (45): {"snapshots": ["2008-03-18-21", "2008-03-18"]}
MIME type: text/html, detected: application/x-javascript, charset: UTF-8
[edit values](#)
- HIGH echo PRED AUTH POST** http://test.example.com:80/examples/res041 dispatcher - 200 [view trace]
Payload: reqReloadDirectoriesSnapshot_id
Response (39): {"dirs": [{"name": "", "numfiles": 12}]}
MIME type: text/html, detected: application/x-javascript, charset: UTF-8
[edit values](#)
- HIGH echo PRED AUTH POST** http://test.example.com:80/examples/res041 dispatcher - 200 [view trace]
Payload: reqReloadDirectoriesSnapshot_id
Response (39): {"dirs": [{"name": "", "numfiles": 12}]}
MIME type: text/html, detected: application/x-javascript, charset: UTF-8
[edit values](#)

Done Apache/2.0.55...



Mapper helps you map the files, file parameters, and values of any site you wish to test

Simply browse the site as a normal user while recording your session with Achilles (Mapper supports other proxies as well), and run Mapper on the resulting log file

It will create an Excel CSV file that allows you to study the directory and file structure of the site, the parameter names of every dynamic page encountered (such as ASP/JSP/CGI), and their values for every time you request for them

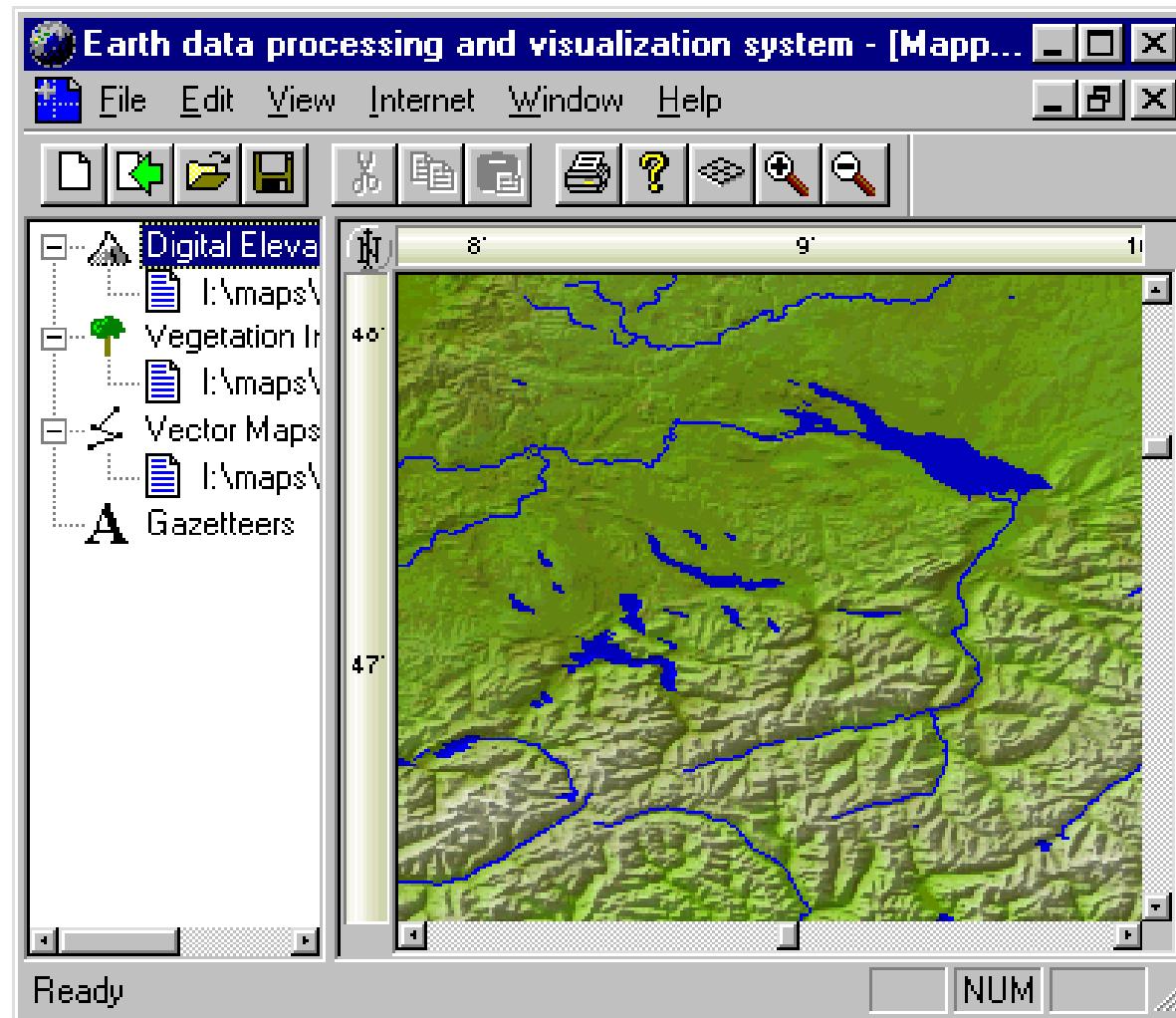
It helps you to quickly locate design errors and parameters that may be prone to SQL Injection or parameter tampering problems

Supports non-standard parameter delimiters and MVC-based web sites



TM

Mapper: Screenshot





TM

What Happened Next

Kimberly could not solve the mystery behind the hack. Jason Springfield, an Ethical hacker was called in to investigate the case.

Jason conducted a penetration test on the website of XBank4u. The test results exposed a vulnerability in the ShrinkWarp application which could lead to web page defacement.

Some other loopholes found on the website were also fixed by Jason.



TM

Summary

Web applications are client/server software applications that interact with users or other systems using HTTP

Attackers may try to deface the website, steal credit card information, inject malicious codes, exploit server side scriptings, and so on

Command injection, XSS attacks, Sql Injection, Cookie Snooping, cryptographic Interception, and Buffer Overflow are some of the threats against web applications

Organization policies must support the countermeasures against all such types of attacks



TM

© 2000 Randy Glasbergen.
www.glasbergen.com



**"Today at work, I received 650 E-mails from
feedme@homecat.com! Was that you?"**



TM

Copyright 2003 by Randy Glasbergen.
www.glasbergen.com



**"I wouldn't say my computer skills are outdated.
I prefer to think of them as 'classic'."**