Hacker Highschool
SECURITY AWARENESS FOR TEENS

LESSON 10
WEB SECURITY AND PRIVACY

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM
INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES

"License for Use" Information

The following lessons and workbooks are open and publicly available under the following terms and conditions of ISECOM:

All works in the Hacker Highschool project are provided for non-commercial use with elementary school students, junior high school students, and high school students whether in a public institution, private institution, or a part of home-schooling. These materials may not be reproduced for sale in any form. The provision of any class, course, training, or camp with these materials for which a fee is charged is expressly forbidden without a license including college classes, university classes, trade-school classes, summer or computer camps, and similar. To purchase a license, visit the LICENSE section of the Hacker Highschool web page at www.hackerhighschool.org/license.

The HHS Project is a learning tool and as with any learning tool, the instruction is the influence of the instructor and not the tool. ISECOM cannot accept responsibility for how any information herein is applied or abused.

The HHS Project is an open community effort and if you find value in this project, we do ask you support us through the purchase of a license, a donation, or sponsorship.

Table of Contents

Contributors

Simon Biles

Pete Herzog, ISECOM

Bill Matthews

Hernán Marcelo Racciatti

Chris Ramirez

P. Shreekanth

Kim Truett , ISECOM

Marta Barceló, ISECOM

Dario Riquelme Zornow

10.1 Fundamentals of Web Security

What you do on the World Wide Web is your business. Or so you would think. But it's just not true. What you do on the web is about as private and anonymous as where you go when you leave the house. Again, you would think that it's your business and many, including ISECOM, would agree with you. However, consider a private investigator following you around town, writing down what you saw and who you spoke with.

The focus of this lesson is to get you learn how to protect yourself on the web and to do that, you will have to learn where the dangers are.

The World Wide Web works in a very straight-forward manner. Once connected to the Internet through you ISP, you open a browser, tell it a website, and you get that website on your screen. However, the truth is in the details. How does the web really work?

A quick trip to the World Wide Web Consortium (W3C), those fine folks who make standards for the web, will teach you all you want to know about the web. http://www.w3.org. Even the history of the web: http://www.w3.org/History.html The problem is, will definitions and standards teach you how to be safe? Apparently not. The people who want to hurt you do not necessarily follow the standards.

10.1.1 How the web really works

The steps involved in connecting to the Internet and then to the web are very detailed even if it does seem to be smooth from the user end.

So what happens for real when you just want to get to the ISECOM website? Assuming you are already connected to the internet, here are the steps that occur in order:

1. You open your browser.
2. You type in the URL (website name).
3. Website name saved in History Cache on the hard disk.
4. Your computer looks up the name of the address to your default DNS server to find the IP address.
5. Your computer connects to the server at the IP address provided at the default web port of 80 TCP if you used "HTTP://" or 443 TCP if you used "HTTPS://" at the front of the web server name (by the way, if you used HTTPS then there are other steps involved using server certificates which we will not follow in this example).
6. Your computer requests the page or directory you specified with the default often being "index.htm" if you don't specify anything. But the server decides t's default and not your browser.
7. The pages are stored in a cache on your harddisk. Even if you tell it to store the information in memory (RAM), there is a good chance it will end up somewhere on your disk either in a PAGEFILE or in a SWAPFILE.
8. The browser nearly instantaneously shows you what it has stored. Again, there is a difference between "perceived speed" and "actual speed" of your web surfing which is actually the difference between how fast something is downloaded (actual) and how fast your browser and graphics card can render the page and graphics and show them to you (perceived). Just because you didn't see it doesn't mean it didn't end up in your browser cache.

The history of the World Wide Web (just "web" from now on) started at CERN[1] in 1989. It was conceived by Tim Berners-Lee and Robert Cailliau who built a basic hypertext based system for sharing information. Over the next few years Tim Berners-Lee continued to develop the system until in 1993 CERN announced that the web was free for anyone to use, and the web as we know it now exploded onto the scene.

The Web is a client and server based concept, with clients such as Internet Explorer, Firefox, Mozilla, Opera, Netscape and others connecting to web servers such as IIS and Apache which supply them with content in the form of HTML[2] pages. Many companies, organizations and individuals have collections of pages hosted on servers delivering a large amount of information to the world at large.

So why do we care about web security then? Web servers often are the equivalent to the shop window of a company. It is a place where you advertise and exhibit information, but this is supposed to be under your control. What you don't want to do is leave the window open so that any passer by can reach in and take what they want for free, and you ideally want to make sure that if someone throws a brick, that the window doesn't shatter ! Unfortunately web servers are complex programs, and as such have a high probability of containing a number of bugs, and these are exploited by the less scrupulous members of society to get access to data that they shouldn't be seeing.

And the reverse is true as well. There are risks also associated with the client side of the equation like your browser. There are a number of vulnerabilities which have been discovered in the last year which allow for a malicious web site to compromise the security of a client machine making a connection to them.

10.1.2 Rattling the Locks

Standard HTML pages are transferred using HTTP[3], this standard TCP based protocol is plain text based and this means that we can make connections to a server easily using tools such as "telnet" or "netcat". We can use this facility to gain a great deal of information about what software is running on a specific server. For example :

```
simon@exceat:~> netcat www.computersecurityonline.com 80
HEAD / HTTP/1.0
```

HTTP/1.1 200 OK
Date: Fri, 07 Jan 2005 10:24:30 GMT
Server: Apache/1.3.27 Ben-SSL/1.48 (Unix) PHP/4.2.3
Last-Modified: Mon, 27 Sep 2004 13:17:54 GMT
ETag: "1f81d-32a-41581302"
Accept-Ranges: bytes
Content-Length: 810
Connection: close
Content-Type: text/html

By entering "HEAD / HTTP/1.0" followed by hitting the "Return" key twice, I can gain all of the information above about the HTTP Server. Each version and make of HTTP Server will return different information at this request – an IIS server will return the following :

1 *Centre Européen pour la Recherche Nucléaire* (European Centre for Nuclear Research)
2 Hyper Text Markup Language
3 Hyper Text Transfer Protocol

```
simon@exceat:~> netcat www.microsoft.com 80
HEAD / HTTP/1.0
```

HTTP/1.1 200 OK
Connection: close
Date: Fri, 07 Jan 2005 11:00:45 GMT
Server: Microsoft-IIS/6.0
P3P: CP="ALL IND DSP COR ADM CONo CUR CUSo IVAo IVDo PSA PSD TAI TELo OUR SAMo CNT COM INT NAV ONL PHY PRE PUR UNI"
X-Powered-By: ASP.NET
X-AspNet-Version: 1.1.4322
Cache-Control: public, max-age=9057
Expires: Fri, 07 Jan 2005 13:31:43 GMT
Last-Modified: Fri, 07 Jan 2005 10:45:03 GMT
Content-Type: text/html
Content-Length: 12934

You can take this further and obtain more information by using the "OPTIONS" request in the HTTP request as follows :

```
simon@exceat:~> netcat www.computersecurityonline.com 80
OPTIONS / HTTP/1.0
```

HTTP/1.1 200 OK
Date: Fri, 07 Jan 2005 10:32:38 GMT
Server: Apache/1.3.27 Ben-SSL/1.48 (Unix) PHP/4.2.3
Content-Length: 0
Allow: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK, TRACE
Connection: close

This will give you all of the allowed HTTP commands that the server will respond to.

Doing all of this by hand is rather tedious, and matching it manually against a database of know signatures and vulnerabilities is more than anyone would want to do. Fortunately for us, some very enterprising people have come up with an automated solution called "nikto".

"Nikto" is a Perl script which carries out various tests automagically ! The options are as follows:

-Cgidirs+	Scan these CGI dirs: 'none', 'all', or a value like '/cgi/'
-cookies	print cookies found
-evasion+	ids evasion technique (1-9, see below)
-findonly	find http(s) ports only, don't perform a full scan
-Format	save file (-o) Format: htm, csv or txt (assumed)
-generic	force full (generic) scan
-host+	target host
-id+	host authentication to use, format is userid:password
-mutate+	mutate checks (see below)
-nolookup	skip name lookup
-output+	write output to this file
-port+	port to use (default 80)
-root+	prepend root value to all requests, format is /directory
-ssl	force ssl mode on port
-timeout	timeout (default 10 seconds)
-useproxy	use the proxy defined in config.txt

-Version print plugin and database versions
-vhost+ virtual host (for Host header)
(+ means it requires a value)

These options cannot be abbreviated:
-debug debug mode
-dbcheck syntax check scan_database.db and user_scan_database.db
-update update databases and plugins from cirt.net
-verbose verbose mode

IDS Evasion Techniques:
1 Random URI encoding (non-UTF8)
2 Directory self-reference (/./)
3 Premature URL ending
4 Prepend long random string
5 Fake parameter
6 TAB as request spacer
7 Random case sensitivity
8 Use Windows directory separator (\)
9 Session splicing

Mutation Techniques:
1 Test all files with all root directories
2 Guess for password file names
3 Enumerate user names via Apache (/~user type requests)
4 Enumerate user names via cgiwrap (/cgi-bin/cgiwrap/~user type requests)

"Nikto" is quite comprehensive in its reporting as you can see from the following scan :

```
exceat:/# ./nikto.pl -host www.computersecurityonline.com

---------------------------------------------------------------------------
- Nikto 1.34/1.29     -     www.cirt.net
+ Target IP:       217.30.114.2
+ Target Hostname: www.computersecurityonline.com
+ Target Port:     80
+ Start Time:      Fri Jan  7 12:23:56 2005
---------------------------------------------------------------------------
- Scan is dependent on "Server" string which can be faked, use -g to override
+ Server: Apache/1.3.27 Ben-SSL/1.48 (Unix) PHP/4.2.3
- Server did not understand HTTP 1.1, switching to HTTP 1.0
+ Server does not respond with '404' for error messages (uses '400').
+     This may increase false-positives.
+ Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND,
PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK, TRACE
+ HTTP method 'PUT' method may allow clients to save files on the web server.
+ HTTP method 'CONNECT' may allow server to proxy client requests.
+ HTTP method 'DELETE' may allow clients to remove files on the web server.
+ HTTP method 'PROPFIND' may indicate DAV/WebDAV is installed. This may be used to get
directory listings if indexing is allowed but a default page exists.
+ HTTP method 'PROPPATCH' may indicate DAV/WebDAV is installed.
+ HTTP method 'TRACE' is typically only used for debugging. It should be disabled.
+ Apache/1.3.27 appears to be outdated (current is at least Apache/2.0.50). Apache 1.3.31 is
still maintained and considered secure.
+ Ben-SSL/1.48 appears to be outdated (current is at least 1.55)
+ PHP/4.2.3 appears to be outdated (current is at least 5.0.1)
+ PHP/4.2.3 - PHP below 4.3.3 may allow local attackers to safe mode and gain access to
unauthorized files. BID-8203.
+ Apache/1.3.27 - Windows and OS/2 version vulnerable to remote exploit. CAN-2003-0460
+ Apache/1.3.27 - Apache 1.3 below 1.3.29 are vulnerable to overflows in mod_rewrite and
mod_cgi. CAN-2003-0542.
+ /~root - Enumeration of users is possible by requesting ~username (responds with Forbidden
for real users, not found for non-existent users) (GET).
+ /icons/ - Directory indexing is enabled, it should only be enabled for specific directories
(if required). If indexing is not used all, the /icons directory should be removed. (GET)
+ / - TRACE option appears to allow XSS or credential theft. See
http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf for details (TRACE)
+ / - TRACK option ('TRACE' alias) appears to allow XSS or credential theft. See
http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf for details (TRACK)
+ /CVS/Entries - CVS Entries file may contain directory listing information. (GET)
```

```
+ /images/ - index of image directory available (GET)
+ /manual/ - Web server manual? tsk tsk. (GET)
+ /cgi-bin/cgiwrap - Some versions of cgiwrap allow anyone to execute commands remotely. (GET)
+ /cgi-bin/cgiwrap/~adm - cgiwrap can be used to enumerate user accounts. Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /cgi-bin/cgiwrap/~bin - cgiwrap can be used to enumerate user accounts. Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /cgi-bin/cgiwrap/~daemon - cgiwrap can be used to enumerate user accounts. Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /cgi-bin/cgiwrap/~lp - cgiwrap can be used to enumerate user accounts. Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /cgi-bin/cgiwrap/~root - cgiwrap can be used to enumerate user accounts. Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /cgi-bin/cgiwrap/~xxxxx - Based on error message, cgiwrap can likely be used to find valid
user accounts. Recompile cgiwrap with the '--with-quiet-errors' option to stop user
enumeration. (GET)
+ /cgi-bin/cgiwrap/~root - cgiwrap can be used to enumerate user accounts.  Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /css - Redirects to http://www.computer-security-online.com/css/ , This might be
interesting...
+ 2449 items checked - 15 item(s) found on remote host(s)
+ End Time:        Fri Jan  7 12:25:36 2005 (100 seconds)
-------------------------------------------------------------------------------
•  1 host(s) tested
```

Using the other options you can fine tune Nikto to do exactly what you need to achieve, including stealth, mutation and cookie detection.

10.1.3 Looking through Tinted Windows - SSL

It wasn't too long before everyone realized that HTTP in plain text wasn't much good for security. So the next variation was to apply encryption to it. This comes in the form of SSL[4], and is a reasonably secure 40 or 128 bit public key encryption method. Using a 40 bit key is a lot less secure than the 128 bit and, with specialized hardware, may well be brute force breakable within a period of minutes, where as the 128 bit key will still take longer that the age of the Universe to break by brute force. There are however more complex technical attacks using something called a known cyphertext attack – this involved calculating the encryption key by analyzing a large number of messages (> 1 million) to deduce the key. In any case, you aren't going to be rushing to try and crack 128 bit encryption – so what can we learn about SSL HTTP Servers?

Quite a lot actually. As the SSL merely encrypts the standard HTTP traffic, if we set up an SSL tunnel, we can query the server as we did in section 1.1. Creating an SSL tunnel is quite straight forward, and there is a utility called "stunnel" purely for this purpose. Enter the following into a file called stunnel.conf, (replacing ssl.enabled.host with the name of the SSL server that you want to connect to:

```
client=yes
verify=0
[psuedo-https]
accept  = 80
connect = ssl.enabled.host:443
TIMEOUTclose = 0
```

Stunnel will then map the local port 80 to the remote SSL Port 443 and will pass out plain text, so you can connect to it using any of the methods listed above :

4 Secure Sockets Layer

simon@exceat:~> netcat 127.0.0.1 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Server: Netscape-Enterprise/4.1
Date: Fri, 07 Jan 2005 10:32:38 GMT
Content-type: text/html
Last-modified: Fri, 07 Jan 2005 05:32:38 GMT
Content-length: 5437
Accept-ranges: bytes
Connection: close

10.1.4 Having someone else do it for you – Proxies

Proxies are middlemen in the HTTP transaction process. The client requests the proxy, the proxy requests the server, the server responds to the proxy and then the proxy finally passes back the request to the client, completing the transaction. Proxy servers are vulnerable to attacks in themselves, and are also capable of being a jumping off point for launching attacks onto other web servers. They can however increase security by filtering connections, both to and from servers.

10.2 Web Vulnerabilities

The simplicity of giving someone something that they ask for is made much more complex when you're in the business of selling. Web sites that sell to you, companies selling products, bloggers selling ideas and personality, or newspapers selling news, requires more than just HTML-encoded text and pictures. Dynamic web pages that help you decide what to ask for, show you alternatives, recommend other options, upsell add-ons, and only give you what you pay for require complex software. When we say goodbye to websites and hello to web applications we are in a whole new world of security problems.

10.2.1 Scripting Languages

Many scripting languages have been used to develop applications that allow businesses to bring their products or services to the web. Though this is great for the proliferation of businesses, it also creates a new avenue of attack for hackers. The majority of web application vulnerabilities come not from bugs in the chosen language but in the methods and procedures used to develop the web application as well as how the web server was configured. For example, if a form requests a zip code and the user enters "abcde", the application may fail if the developer did not properly validate incoming form data. Several languages can be used for creating web applications, including CGI's, PHP and ASP.

Common Gateway Interface (CGI): Whatis.com defines a CGI as "A standard way for a web server to pass a web user's request to an application program and to receive data back to forward to the user." CGI is part of the web's Hypertext Transfer Protocol (HTTP). Several languages can be used to facilitate the application program that receives and processes user data. The most popular CGI applications are: C, C++, Java and PERL.

PHP – Hypertext Preprocessor (PHP): PHP is an open-source server-side scripting language where the script is embedded within a web page along with its HTML. Before a page is sent to a user, the web server calls PHP to interpret and perform any operations called for in the PHP script. Whereas HTML displays static content, PHP allows the developer to build pages that present the user with dynamic, customized content based on user input. HTML pages that contain PHP scripting are usually given a file name with the suffix of ".php".

Active Server Pages (ASP): Web pages that have an .asp Active server pages (ASP), are database drive dynamically created Web page with a .ASP extension. They utilize ActiveX scripting -- usually VB Script or Jscript code. When a browser requests an ASP, the Web server generates a page with HTML code and immediately sends it back to the browser – in this way they allow web users to view real time data, but they are more vulnerable to security problems.

10.2.2 Common Web Application Problems

Web applications do not necessarily have their own special types of problems but they do have some of their own terms for problems as they appear on the web. As web application testing has grown, a specific security following has grown too and with that, a specific classification of web vulnerabilities. Common web application problems are classified below according to the OSSTMM Risk Assessment Values (http://www.isecom.org/securitymetrics.shtml), a specific way to measure security by how it affects how things work.

RAV	*What it means*	*Web Examples*
Authentication	These are the identification and authorization mechanisms used to be certain that the person or computer using the web application is the correct person to be using it.	Every time you login to a web page that has your personal data then you are authenticating. Authentication often means just giving a login and password. Sometimes it means giving an identification number or even just coming from n acceptable IP Address (white-listing).
Non-Repudiation	A record that proves that the data sent to or from the web application was really sent and where.	Although you may not see it, most web applications keep track of purchases you make from a particular IP address using a particular browser on a particular operating system as a record that it was most likely smeone on your computer who made that purchase. Without specific "authentication" they can't guarantee 100% it was you though.
Confidentiality	A way to assure that communication with the web application cannot be listened in on by another person.	The HTTPS part of interaction with a web application provides pretty good confidentiality. It does a decent job of making your web traffic with the web app from being publicly readable.

RAV	What it means	Web Examples
Privacy	A way to assure that the way you contact and communicate with the web application cannot be pre-determined by another person.	While it is very rare, it is not unimaginable that a web application that contains very private information would not even show you it is there unless you come from the right place and know the right secret combination to get the web app to be accessible. One way is to have to click a picture in 5 different places in a specific order to get to the login screen. Another manner is called port-knocking and it means that the server requires a specific sequence of interactions before it opens a port, such as the HTTP port, to the user.
Indemnification	These are ways to assure that the web application has legal protection or at the least, can be financially protected with insurance.	Some web sites clearly print on the login screen that it's for authorized personnel only. If someone steals a login and password or even brute-forces it open, the attacker, if caught, cannot say he didn't know it was private.
Integrity	This is a record of the validity of the communication with the web application to assure that what is sent and then received by the other is the same thing and if it changed, both the web pplication and the user have a record of the change.	Some web apps provide a "HASH" with files to be downloaded. This HASH is a number generated from that specifc file. When you download the file, you can check the HASH you generate from the file against the one they post. This is to assure that some attacker is not trying to trick you with a different file either replaced or through deception, such as in Cross Site Scripting.
Safety	This is how we protect the web application from it's own security devices. If security fails, we need to make sure that it does not affect the operation of the web application as a whole.	It is very possible to have an application use a daemon that can re-initialize itself or even prevent an attack from crashing any part of itself by presenting itself only virtually. You can also find scenarios where a web app uses an intrusion detection mechanism that "stops" attacks by blocking the attacker by IP address. In this case, we can't say Safety exists if the security device is configured to prevent an attacker from spoofing the web app's own resources and causing this defense to block important traffic. Instead, it is considered either a misconfiguration of the defense or in some cases a weakness of design. Don't confuse a poorly made or "accidental" defense with a designed loss control.

RAV	*What it means*	*Web Examples*
Usability	A way to prevent the user from having to make security decisions about interacting with the web application. This means that proper security is built in and the user doesn't have to choose which or what security mechanisms to turn on or off.	When a web app requires use of HTTP over SSL (HTTPS) then we can say that it is using Usability as part of security. However, if it lets you choose to interact with it less securely, for example, to send your credit card number by insecure e-mail rather than post it via a form by way of HTTPS, then it is NOT exercising Usabilty.
Continuity	This is how we keep a service based on a web application from failing to work no matter what problem or disaster occurs.	Often times a web app that receives a lot of traffic will have a reverse proxy in front of it which directs the traffic to one of many mirrored web servers. This way, if one goes down, service is not interrupted. Another example is a web application that caches its website to many different servers over the internet so when you visit one, you are nt actually going to the originating web server. If a cache goes down or gets corrupted, then the traffic will get redirected to another cache or the originating website.
Alarm	A notification, either immediate or delayed, regarding a problem with any of these mechanisms.	A basic form of alarm is the log file generated by the web server. The bad thing about an alarm is that you can choose to ignore it. This is especially true if it sounds all the time (think of the story of the boy who cried "wolf". Or in the case of a log file, it may not sound at all. Alarm is only as good as your reaction time to it.

Exercises:

1. Open up google and type in "inurl:search.asp" or "inurl:search.php". With any of the websites which come up, attempt to type in the following in the search field **<script>alert ("hello")</script>**. What happens? Try this for several sites.

2. In google, type in "inurl:login.asp" ond "inurl:login.php". With any of the websites which come up, attempt to type in special characters (@#$^&) for both the username and password. What happens? Try this for several sites.

3. Knowing the types of security mechanisms a web application may have, open your favorite, interactive website and try to identify if it has security mechanisms which conform to any of the RAV classifications.

4. Commonly discussed web vulnerabilities are Cross Site Scripting (XSS) and SQL injection. What are they and how does an attacker use them to steal data or information from a web application?

10.2.3 Guidelines for Building Secure Web Applications

While there are many opinions and most of the details to building with security in mind come from the logic of the programmer and their skill with the programming language, these basic guidelines are also derived from materials available from the OSSTMM (http://www.osstmm.org).

1. Assure security does not require user decisions.
2. Assure business justifications for all inputs and outputs in the application.
3. Quarantine and validate all inputs including app content.
4. Limit trusts (to systems and users).
5. Encrypt data.
6. Hash the components.
7. Assure all interactions occur on the server side.
8. Layer the security.
9. Invisible is best- show only the service itself.
10. Trigger it to alarm.
11. Security awareness is required for users and helpdesks.

Exercises:

1. Give examples for any three of the above guidelines.
2. Give three types of technologies that one could apply to a web application as an alarm.

10.3 HTML Basics – A brief introduction

HTML is a set of instructions that explains how information is to be presented from a web server (Apache, Internet Information Server) to a browser (Firefox, Opera). It is the heart of the World Wide Web.

HTML can do much more than just display data on a web page. It can also provide data entry forms, where data can be entered for processing by a higher level language (Perl, PHP, etc). In a business setting this is where HTML is at its most useful but in a hacker setting, this is where HTML is at its most vulnerable.

10.3.1 Reading HTML

HTML is communicated with a series of tags or markups. Each opening tag, <h1>, for instance, must have a closing tag, </h1>. This tells the browser to stop the markup described by the preceding tag. Opening and closing tags are a part of well-formed HTML.

Take, for example, the code:

```
<html>
<head><title>Hello World</title></head>
<body>
<h1>Hello World!</h1>
</body>
```

```
</html>
```

Figure 1: HTML Code

We are telling the browser this is an HTML document with the tag <html> and we have a title of 'Hello World' with the <title> tag. The <body> tag tells our browser "here is where the information you will be displaying goes." Finally, the <h1> tags tells the browser to display the information in "Heading 1" style. The tags that are preceded with a '/' are merely the closing tag, this tells the browser to stop displaying the contents described by the opening tag.

Exercise 1: Cut and paste the code in figure one and paste it into a text file called hello.html. Open that file in your browser of choice and you should see something similar to this:

Hello World!

10.3.2 Viewing HTML at its Source

All modern browsers contain a way to view the underlying HTML code that generated the web page you are looking at. In most cases, this is the "view source" option under the "view" menu in your browser.

Exercise 2: Choose View --> View Source in your browser while surfing your favorite web page.

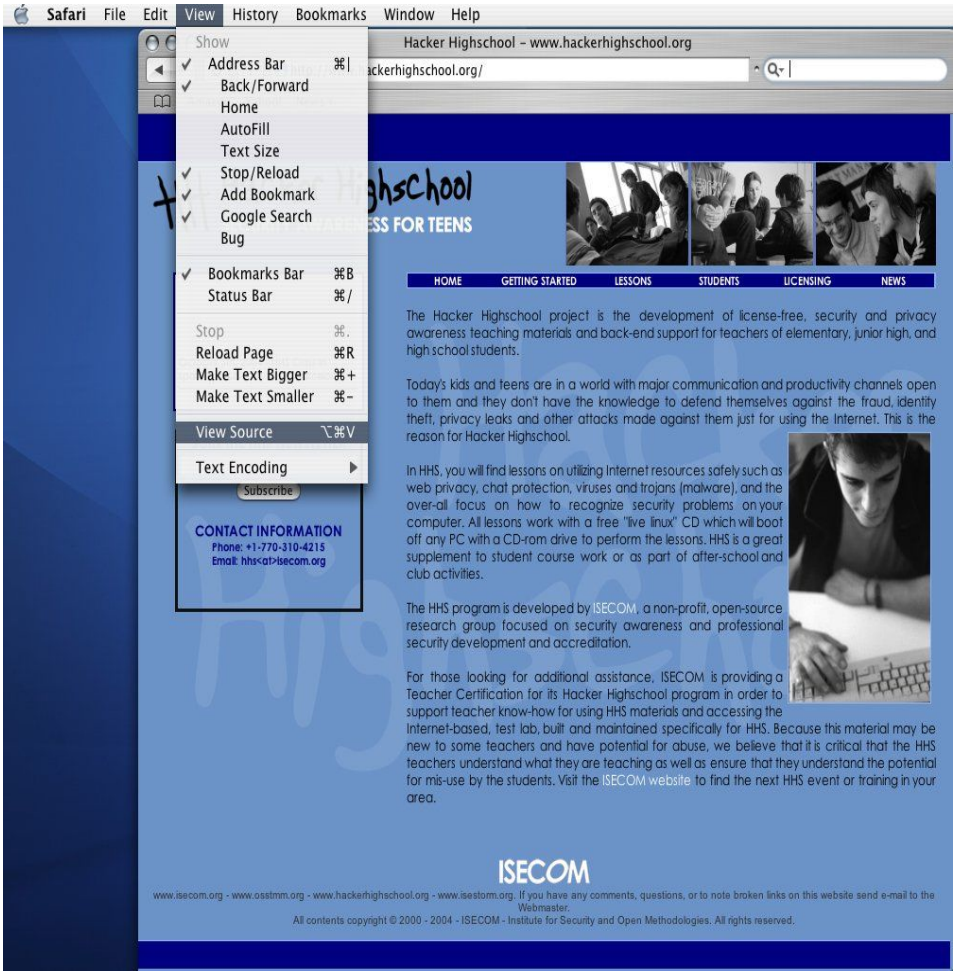

Illustration 1View Menu

The results should be something pretty similar to this:

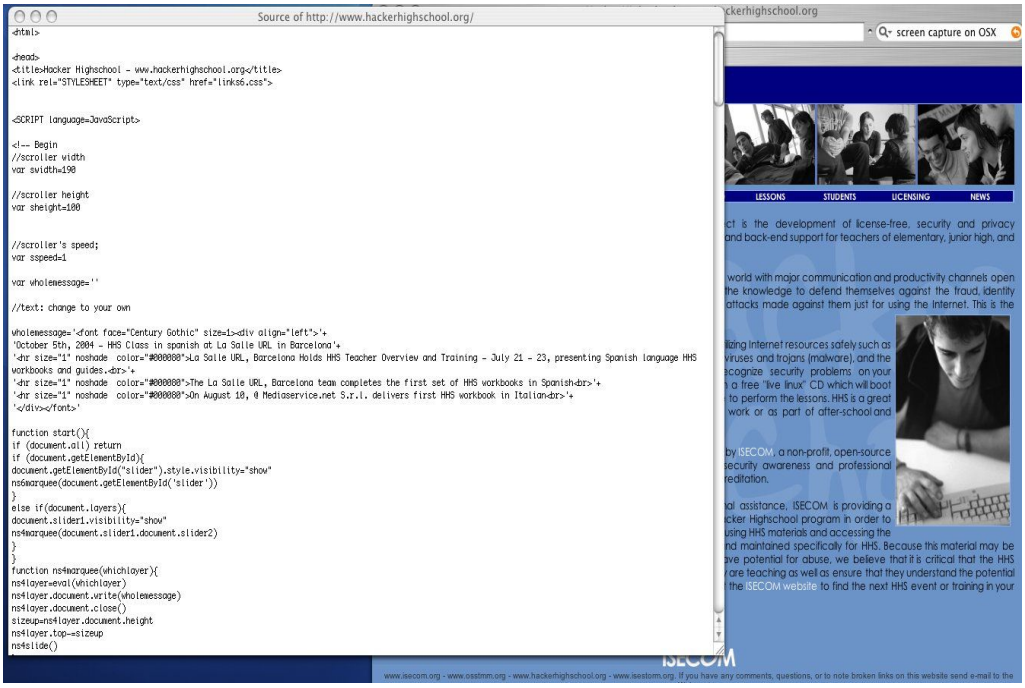

Illustration 2Source viewed in text editor

HTML code is visible to anyone with a web browser. This is why it is very important when coding web pages to not try to hide passwords or important information in the HTML source code. As you can see, its not very secret.

10.3.3 Links

Links (or hyper-links) are really the heart of HTML page building. The biggest strength of HTML is the ability to link to other documents. A link, in the context of HTML is denoted as www.yahoo.com The link will appear as www.yahoo.com on your website. This will take visitors of your site to Yahoo.

Links can be checked and followed followed by so-called link checker programs. These programs search HTML source code for the tags and then create a file or index of the found links. Spammers will often use this technique to find email addresses or contact forms they can use to spread their mass emails. Link checkers can also be used to check your website for "broken" links or links that don't go anywhere. This can happen a lot even in relatively small sites.

Exercise 1: Create a link

Create a link to www.hackerhighschool.org that displays as Hacker High School on your web page.

Bonus exercise: Use the tool

1. Find and download a link checking program
2. Run that program against www.hackerhighschool.org and document how many broken links you find.

10.3.4 Proxy methods for Web Application Manipulation

An HTTP proxy server serves as a middle man between a web server and a web client (browser). It intercepts and logs all connections between them and in some cases can manipulate that data request to test how the server will respond. This can be useful for testing applications for various cross-site scripting attacks (provide reference link here), SQL Injection attacks and any other direct request style attack. A proxy testing utility (SpikeProxy, WebProxy, etc), will assist with most of these tests for you. While some have an automation feature, you will quickly learn that it is actually a weak substitute for a real person behind the wheel of such tools.

Exercise 1: Choose your software

1. Download a proxy utility
2. Install the software according to the README file
3. Change your browser setting to point to the new proxy
   - This is usually port 8080 on localhost for these tools but read the instructions to be sure.

Once the proxy server is installed and your browser is pointed at it, surf around the site your testing. Remember, be sure to use a website that you have permission to test. Once you have surfed around, point your browser to the proxy's admin page (for SpikeProxy, it http://www.immunitysec.com/resources-freesoftware.shtml) and begin testing the site. From the admin interface you can have the tool brute force the site's authentication methods or test for cross-site scripting. (Actually, we recommend using Mozilla or Firefox and http://livehttpheaders.mozdev.org/ and http://addneditcookies.mozdev.org/ together to modify headers and cookies on the fly without the need for a seperate proxy port. Not only does it really simplify things, it's a much more powerful tool set as we teach it in ISECOM's OSSTMM Professional Security Tester class (OPST). But since you will need to know about setting up proxies for other things, like ad and spam filters, privacy filters, etc. We thought you should actually set one up for real and Spike is a good one to try.)

A proxy server can be a powerful tool in helping you determine how solid a web application is. For penetration tests or vulnerability assessments, you must have a good proxy tool in your toolbox. There are detailed tutorials available on using SpikeProxy at http://www.immunitysec.com/resources-papers.shtml.

10.4 Protecting your server

There are several steps that can be taken to protecting your server. These include ensuring that your software is always updated and patched with any security updates that are available from the manufacturer. This includes ensuring that your OS and web servers are updates as well. In addition, Firewalls and Intrusion detections systems can help protect your server, as discussed below.

10.4.1 Firewall

Firewalls originally were fireproof walls used as barriers to prevent fire from spreading, such as between apartment units within a building. The same term is used for systems (hardware and software) that seeks to prevent unauthorized access of an organization's information. Firewalls are like security guards that, based on certain rules, allow or deny access to/from traffic that enters or leaves an organization (home) system. They are important systems safe guards that seek to prevent an organization's system from being attacked by internal or external users. It is the first and most important security gate between external and internal systems.

Firewalls are generally placed between the Internet and an organization's information system. The firewall administrator configures the firewall with rules allowing or denying information packets from entering into or leaving the organization.

The rules are made using a combination of Internet Protocol (IP) address and Ports; such rules are made depending on the organization needs e.g. in a school, students are allowed in based on identity card.

The rule to the security guard in a school would be to allow all persons that carry a valid identity card and deny everyone else. However the security guard would have another rule for exiting from the school; the rule would be to allow everyone exit except small children unless accompanied by adults. A similar system is followed for firewall configuration depending on the nature of the organization, the criticality of information asset, cost of security, security policy and risk assessment.

The firewall just like a security guard cannot judge the contents of the information packet; just like the guard allows all persons with a valid identity card irrespective of nature of the persons, firewall allows entry or exit based mainly on IP address and Port numbers. Hence an entry or exit is possible by masking IP address or Port. To mitigate this risk, organizations use Intrusion Detection System, which is explained in the next section.

There are various kinds of firewall depending on the features that it has viz. packet filter (operates on IP packets), stateful firewall (operates based connection state) or application firewall (using proxy).

Example of a firewall rule could be: Block inbound TCP address 200.224.54.253 from port 135. (An imaginary example); such rule would tell a computer connected to Internet to block any traffic originating from the computer with an IP address 200.224.54.253 using Port 135.

Important activities relating to firewalls are initial configuration (creating initial rules), system maintenance (additions or change in environment), review of audit logs, acting on alarms and configuration testing.

10.4.2 Intrusion Detection System (IDS)

Imagine in a school that has proper security guards; how will the authorities detect entry of unauthorized persons? The authorities would install burglar alarm that will ring on entry of unauthorized persons. This is exactly the function of intrusion detection system in computer parlance. Firewall (security guard or fence) and IDS (burglar alarm or patrolling guard) work together; while firewall regulates entry and exits, IDS alerts/denies unauthorized access.

So how does IDS help? Just like burglar alarms, IDS alerts the authorized person (alarm rings) that an authorized packet has entered or left. Further, IDS can also instantly stop such access or user from entering or exiting the system by disabling user or access. It can also activate some other script; IDS can for example prevent or reduce impact of denial of service by blocking all access from a computer or groups of computer.

IDS can be host based or network based; host based IDS are used on individual computers while network IDS are used between computers. Host based IDS can be used to detect, alert or regulate abnormal activity on critical computers; network IDS is similarly used in respect of traffic between computers. IDS thus can also be used to detect abnormal activity.

IDS like patrolling guard regularly monitors network traffic to detect any abnormality e.g. high traffic from some computers or unusual activity on a server, e.g. user logged onto application and involved in malicious activity. IDS compare any event with historical data to detect any deviation. On detection of deviation, IDS act depending on the rule created by IDS administrator such as alerting, storing such intrusion in audit logs, stopping user from doing any activity or generating script for starting a string of activities. IDS can also detect deviation based on its database of signatures – any deviation to signature is detected and acted upon- this action is similar to anti virus software. IDS is also used for detection of any activity on critical resource or for forensic by quietly watching the suspect.

Exercises:

1. Are both firewall and Intrusion Detection System required in an organization for securing its information system? If yes why? If not, why not?
2. Think of an example of a specific use of firewall rules that is applicable to the front desk person in a school; does she need to access Internet? If not, how will the rule be enforced?
3. Can a student access the school score database that contains complete information on examination scores of all students. How will this be controlled? How will this be detected in case an external party using Internet unauthorizedly accesses it?

10.5 Secure Communications

Generally, the concept associated with security communications are the processes of computer systems that creates confidence and reduces risks. For electronic communications, three requirements are necessary to ensure security. A) Authenticity b) Integrity c) Non repudiation.

Authenticity: This concept has to do with ensuring that the source of a communication is who it claims to be. It is not difficult to falsify electronic mail, or to slightly vary the name of a web page, and thus redirect users, for example http://www.diisney.com appears to be the Disney web page, but it has 2 letters "i" and can be confusing. In this case, you are actually transferred to a gambling site and the communications are not safe.

Integrity: That a communication has Integrity means that what was sent, is exactly what arrives, and has not undergone alterations (voluntary or involuntary) in the passage.

Non repudiation: If the conditions of authenticity and Integrity are fulfilled, non-repudiation means that the emitter cannot deny the sending of the electronic communication.

For example, if a Web site grants a prize to me, and I can prove it - that is to say, if a Web site sends a discount coupon, and I verify that the Web site is authentic, and that nobody manipulated the information in the way, the site cannot deny that the coupon was sent.

The form used to assure these conditions from a Web site is called an electronic certificate.

Maintaining the conditions of security gives us tranquillity in our electronic communications, and allows to assure the principle the privacy in the cyberspace.

10.5.1 Privacy and Confidentiality

Most web sites receive some information from those who browse them - either by explicit means like forms, or more covert methods like cookies or even navigation registries. This information can be helpful and reasonable – like remembering your book preferences on Amazon.com and, therefore,in order to ensure security to the person who browses, many sites have established declarations of Privacy and Confidentiality.

Privacy refers keeping your information as yours – or limiting it to close family or your friends, or your contacts, but at the most, those who you have agreed to share the information. No one wants their information shared everywhere without control, for that reason, there are subjects declared as private, that is to say, that of restricted distribution.

On the other hand, the **confidentiality** talks about that a subject's information will stay secret, but this time from the perspective of the person receiving that information.

For example, if you desire a prize, but you do not want your information distributed, you declare that this information is private, authorize the information to a few people, and they maintain confidentiality. If for some reason, in some survey, they ask to you specifically for that prize, and you respond that if you have it, you would hope that that information stays confidential, that is to say, who receive the information keep it in reserve.

We could generalize the definition of confidentiality like "that the information received under condition of privacy, I will maintain as if it was my own private information". It is necessary to declare the conditions of the privacy of information handling, to give basic assurances of security.

Also it is recommended that you read the conditions established by the web site you visit in their privacy policy.

Exercise:

1. Review the conditions of privacy of world-wide suppliers of WebMail: Google and Hotmail and of manufacturer like General Motors motors http://www.gm.com/privacy/index.html. Are they equal? Of those, who will share the information that I give? What measures will I be able to take if they do not observe these rules?

10.5.2 Knowing if you are communicating securely

Even with conditions of Privacy and Confidentiality, somebody can still intercept the communications. In order to give conditions discussed at the beginning of this section, a layer of security has been previously discussed called SSL, which uses digital certificates to establish a safe connection (is to say that it fulfills the authenticity, integrity and non repudiation) and provides a level with encryption in communications (this is to hide information so that if somebody takes part of the information, they cannot access it, because the message is encypted so that only the sender that sends it and the receiver, with a correct certificates, is able to understand it). This layer is called Security Socket Layer, SSL, and is visible through two elements within the web browser.

The communications is considered to be safe when the web address URL changes from HTTP to https, this change even modifies the port of the communication, from 80 to 443. Also, in the lower bar of the navigator, a closed padlock appears, which indicates conditions of security in the communications.

If you put mouse on this padlock, a message will apepar detailing the number of bits that are used to provide the communications (the encryption level), which as of today, 128 bits is the recommended encryption level. This means that a number is used that can be represented in 128 bits to base the communications.

A type of called trick phishing exists (http://www.antiphishing.org/) in which a Web mimics the page to make seem from a bank (they copy the graphics, so that the clients enter their data, trusting that it is the bank, although it is not it). In order to avoid these situations, the authenticity of the site should be verified, and checked that the communications are safe (https and the closed padlock), and to the best of your knowledge, it verifies the certificate.

10.6 Methods of Verification

At this point, you have had opportunity to know the foundations the security in the Web, the main aspects related to some of the vulnerabilities found commonly in the web servers used to lodge the different sites with which we routinely interact when browsing in Internet, and the form in which different defects in the development of web applications, affect the security and/or the privacy of the users in general.

On the other hand, you have learned some of the technologies on which we rely to protect our servers and also our privacy. However, probably at this moment, you are realizing questions such as: I am safe, now that I have taken the corresponding actions? Is my system safe? The developers that have programmed some of the functionalities that I have used in my Web site, have they taked care of ensuring aspects to the security? How I can verify these aspects?

As probably you have thought, it is not enough to apply manufacturer updates or trust the good intentions of the developer, when your security or privacy is concerned. In the past, there have been several cases in which manufacturer's patches corrected one vulnerability, but causing another problem in the system, or once patched discovered a new vulnerability. Due to this and other reasons, you will have to consider, that is absolutely necessary to verify frequently the implemented systems, in order to the system "remains" safe.

Luckily, many people have developed in their own time, some "Methods of Verification", most of which are available free, so that we all may take advantage of the benefits of its use. Such they are based on the experience of hundreds of professionals, and include numerous "good practices" regarding implementing technology in safe form. Therefore, it is recommended, that you adopt these methodologies at the time of making your tasks of verification.

An example of these, the OSSTMM is discussed briefly below.

10.6.1 OSSTMM

The OSSTMM, which is an abbreviation for "Open Source Security Testing Manual Methodology" is one of the methodologies of testing security that is widely used. As described in its introduction, although certain individual tests are mentioned, these are not particularly revolutionary, the methodology altogether represents a standard of essential reference, for anyone wanting to carry out a test of security in an ordered format and with professional quality. The OSSTMM, is divided in several sections. In the same way, it is possible to identify within it, a series of specific testing modules, through which each dimension of security is tested and integrated with the tasks needed to ensure security.

This sections include: Personnel Security, Data Network Security, Telecommunications Security, Wireless Communications Security, and Physical Security, and the sections of this methodology detail security from the point of view of WHICH test to do, WHY to do it and WHEN to do it.

The OSSTMM by itself details the technical scopes and traditional operation of security, but , and this is perhaps one of the very important aspects, not the exact tests, rather it presents, what should be tested, the form in which the test results must be presented/displayed, the rules for testers to follow to assure best results, and also, incorporates the concept of security metrics with RAVs (Risk Assessment Values) to put a factual number on how much security you have. The OSSTMM is a document for professionals but it is never too early to try to understand it and learn how it works. The concepts are very thorough and it's written in an easy-to-comprehend style.

Exercises

1. Patching is a common problem today where web administrators are currently needing to patch code as new vulnerabilities are discovered. Research for a case in where a new problem occurred when installing a new security patch. Discuss about the possibilities and consequences that an administrator, who has a new patch to install, realizes that this will open a breach in its system that already was resolved. Should the patch still be installed? In relation to this subject, would it matter whether you have the source code and not?
2. Go to http://cve.mitre.org and go to search for CVEs. Enter the name of a web server (ie Apache) into the search field. When did the latest vulnerability get released? How often have vulnerabilities come out (weekly, monthly, etc.)? In reference to question number one, is patching a realistic solution to security? Why or why not? What other security measures can be used if you decide not to play the cat and mouse game of patching?
3. Download a copy of the OSSTMM and review the methodology concepts. What aspects would you emphasize from this methodology? How you think that this methodology can integrate with your verifications of security?
4. What you can find out of the RAVs?

Further Reading

http://www.osstmm.org

http://www.oreilly.com/catalog/websec2/chapter/ch08.html

http://www.w3.org/Security/Faq/

http://www.privacyalliance.org/

http://www.perl.com/pub/a/2002/02/20/css.html

http://www.oreilly.com/catalog/webprivp3p/chapter/ch01.pdf

http://www.defenselink.mil/specials/websecurity/

http://www.epic.org/

http://www.cgisecurity.com/

http://www.eff.org/privnow/

Here are some sites to check out if you want more information on creating your own web pages or HTML in general.

http://www.htmlgoodies.com/
http://www.htmlhelp.com/
http://www.w3schools.com/