

Web & Internet Security Considerations

©July 30, 2004-material compiled by Bob Carnaghi www.webpointmorpheus.com

Introduction

This document is one in a series of "Technical White Papers" that attempts to interpret and explain in non-technical language the workings of computers on the Internet. The topic of this document is Web & Internet Security Considerations. The sheer complexity and technicality of the workings of the Internet and computers in general have given new dimensions to the concept of security. In essence, the Internet is a 'wide open playing field' that was intentionally designed to be that way. Hence, security takes an increasingly difficult role. Other documents in this series are mentioned and referenced, and further reading to support the concepts introduced here may be necessary.

Security on the Internet

Initially, the Internet was designed to be open and accessible by anonymous users. The driving factors that were initially considered in the design of the system we know today as the Internet were:

Decentralization - no key, critical, centralized location.

Redundancy - the system must work even if significant parts of the network become inoperable.

Asynchronous - require direct communication not required, allowing intermediary machines to be used.

Concurrency - each system can have multiple, concurrent, communications sessions.

If this sounds military, it is. The original investor in the research that led to this system was the US Government in the ARPA (Advanced Research Projects Agency,) which later became the DARPA (Defense Advanced Research Projects Agency.) In order to fulfill the guidelines listed above, the nature of the open internet webpage request or file transfer was designed to be typically *stateless*. What this means is that to a typical web server providing web pages or files for download, one after another, there is no intrinsic way to tell that two successive requests are coming from the same person. Let's consider an example:

stories-of-interest.com provides online short stories, and the website serves about 50 html pages total. Most of the stories are about 5 pages long, so there are 10 stories posted online. At any given time, there are about 25 people reading stories. By the law of averages, since there are 10 stories, there is an average of 2.5 people reading a story at any given time. By its intrinsic nature, the web server has no way to tell which of the 2.5 people that submit a request for, say, page 2 of the story came from which specific person reading the story. Did the request come from the person who just finished page 1, or did it come from the person who just finished page 3, but back-tracked to review? There are ways to track this, but essentially, the pages are simply served without doing so.

Cookies

Cookies are the first method of approach to track *state*, and arguably, the simplest. Cookies are a small text file stored on the client computer. A Cookie is set by the web server onto the client computer, and has a tiny amount of information that tells who the web server is that set the cookie, when it was set, when it expires, etc. Other information can be contained in the Cookie as well. With a Cookie set onto the client's computer, the web server, upon page requests, can check to see if there has been a Cookie previously set. If so, the web server can now track the requests.

There are drawbacks to this system, and controversy abounds. Initially, the concept of Cookies was not well received, mostly because of privacy issues. Should unsuspecting clients have to be imposed upon in ways that they aren't aware of or have no control over? The controversy continues to this day, and is likely to continue. By basic nature, the Cookie system isn't bad. What can be bad is the use of this system for excessive control or imposition.

Oddly, the initial specification for Cookies was drafted by Netscape, and has not changed. It can still be viewed online at www.netscape.com/newsref/std/cookie_spec.html

Logins & Passwords

Some websites ask visitors to create an account with a profile. These account settings are typically controlled by what is called a 'session id', and the account information (more often than not) is stored in a database. With the account name, one can then be verified against the password that's stored in the database. If all matches, the visitor has been validated, and is permitted access to certain protected web pages.

The system used here works from what are called 'session variables', cookies, or a combination of both. This system is often used by financial institutions, online stores or other database driven websites, e-commerce websites, etc. Usually, this is any entity that requires a secure form of validation for the visitor or client. Often there is a long string of garble in the browser's address window. Part of this information is often carried from page to page as the session id.

This method is relatively secure, especially when combined with Encryption, below. However, it's not foolproof. There are several pitfalls, some of which are listed in the Hacking section, below.

Hacking

In essence and originally, the term 'hacker' was (and still often is) not about a person committing illegal, immoral, or unethical computer acts. The term 'hacker' was used to refer to a person who worked or used a computer on a daily basis. In contrast, a 'cracker' is the person who commits the illegal, immoral, or unethical actions mentioned. In keeping with the true definition of the person, this article will use the 'cracker' term for the person attempting or committing illegal, immoral, or unethical activities.

A thorough coverage of the means and methods of 'crackers' is well beyond the scope of this document. In order to simplify this presentation, the material offered will focus on those practices which target the general user of the internet. A short list of the most general 'cracker' practices follows:

- XSS - cross site scripting , or the insertion of tags or scripted code into another site's web page. This is often the result of following links that have been tampered with by inserting scripted code into the URL. Another way to XSS is to insert code into form fields before submitting the form.
- Key Logging – a practice that typically uses software that is planted on an unsuspecting person's computer. The software has the ability to log the keystrokes used. The 'cracker' then extracts account settings, passwords, or other valuable information for illegal use.
- Phishing – a practice, often through the email system, to get a person to enter their account numbers, passwords, etc. into a website or form that is disguised. Their account identity is compromised (identity theft) for illegal activity.
- Weak passwords – computer programs that can generate passwords are often used against account logins. For the length of every character that's contained in a password, the generator must attempt a match. A password of 4 characters requires relatively few character match attempts compared to a password that is 8 characters, which must also be a combination of lower-case, upper-case, and punctuation characters.
- Web server compromise – This is a multi-inclusive item on the list. The potential for problems here are many, and could easily become extremely technical. A short list of some of the general items: Domain Name (BIND) weaknesses, vulnerable CGI scripts, buffer overflows, file sharing and permission issues, mail configuration settings, ID and/or root profile compromise, etc. One of the major keys to preventing problems on the web server is the maintenance of all software to the latest upgrades and patches. The web server error logs are incremental to tell of 'cracking' attempts.

Security Certificate

A Security Certificate is a digitally signed document that is installed on the web server of the website that offers the secure connection. Clients who visit the website are then capable of validating the identity of the website by viewing its Security Certificate. When coupled with Secure Sockets Layer and Encryption methods (below,) as well as Session Variables (above,) a client is reasonably guaranteed of a secure connection to the website that they intend to interact with.

A Security Certificate can be compared to a passport. A passport is issued by a country to vouch that the holder is a citizen of that country. A Security Certificate is issued by a Signing Authority, and vouches that the entity who holds the certificate is who they say they are. In terms of Security Certificates, generally, the more one pays, the more the Signing Authority will vouch for, and the more they will investigate the identity of the entity which applied for the certificate.

Encryption

Encryption is the method used to code information into a thoroughly unintelligible form. The information is then able to be decoded by the intended recipient when received or needed. There are several encryption algorithms currently available, and to different degrees of strength. The strength is usually measured by the amount of bits, 40 bit encryption, or 128 bit encryption. The higher the measurement of bit encryption, the more secure the transaction.

SSL or Secure Sockets Layer

Web pages that are considered secure are transmitted over what's called a Secure Socket Layer. This is an additional step in the web surfing process that encrypts the web pages so that they can't be intercepted and interpreted by a 'cracker' en route from web host to client. The SSL, when activated, will often show a lock somewhere on the browser window, usually on or near the status bar. By clicking this lock, one can view the security settings for the website being viewed.

Viruses, Worms, and Trojan Horses

In the interest of simplicity, the definitions of these terms are listed below. Please consider the [webpointmorpheus](#) document Viruses, Worms, and Trojan Horses for further clarification.

- Virus: A computer program or script with the intention of doing harm, gaining control of a computer, or otherwise committing an unwanted action
- Worm: A virus capable of replicating itself. Consider the multitude of email worms which have been released.
- Trojan Horse: A virus or worm that is planted on a computer by some means in order to execute an action at a later time. Some of these are time-based – they go off at, say, 1:00 am. Others wait for certain keystrokes or a specific program is launched.

E-Commerce & Credit Card Transactions

E-Commerce and Credit Card Transactions is one of, if not the predominant, area where Web & Internet Security Considerations come full circle. The need for secure transfer of funds has provided the impetus to develop the ways and means of security listed in this article. Virtually every financial transaction conducted on the Internet is done in a secure environment. With the knowledge presented in this document, one should be able to approach these transactions with new found confidence.

Additional security provided to enterprise and government entities use the same means listed in this document. The difference is that these entities are typically securing an intranet with protected files or documents. However, the security measures, as well as system compromise, remain the same.

Conclusion

[webpointmorpheus](#) is 'Total Solution Web Design' and is committed to staying abreast of the current industry trends in Web and Internet Security Considerations. [webpointmorpheus](#) has assembled E-Commerce websites from small single-item custom scripted sites which use Paypal to larger E-Commerce solutions that offer the site owner the ability to totally administer the site themselves. All of the methods discussed in this document are familiar to [webpointmorpheus](#), and are used consistently as needed.

This article has covered (briefly and non-technically) the bare essentials of Web and Internet Security Considerations. This topic is of a crucial nature, and there are entire sections of the computing industry completely dedicated to security issues. The intent of this document has been to inform the reader in a non-technical manner of the generalities of Web and Internet Security Considerations. The ultimate goal is to edify the reader with enough information to make clear, informed decisions and actions.

SOURCES:

Hacking Linux Exposed, Brian Hatch & James Lee, McGraw Hill/Osborne, 2nd Edition
Red Hat Linux Administration, Michael Turner & Steve Shah, McGraw Hill/Osborne
PHP and MySQL Web Development, Welling & Thomson, Developer's Library, 2nd Edition

Additional

The website process, way the web works, Search Engines, and other web & internet concepts are often very confusing, especially for the typical non-technical person. [webpointmorpheus](http://webpointmorpheus.com) has assembled several documents hoping to simplify these topics. This series of documents are the result of a consistent set of questions posed by current, past, and potential webpointmorpheus clientele. The documents are listed below, and are available online at www.webpointmorpheus.com.

Documents available from [webpointmorpheus](http://webpointmorpheus.com):

Why Would I Want a Website?

What's Involved Launching a Website?

Block view of the Web Site Design Process

Block View of Typical Web Page Request on the Web

What is Web Hosting?

DNS Stands for Domain Name System

Web & Internet Security Considerations

E-commerce 101: Is it for me?

What are Search Engines?

webpointmorpheus Search Engine Services

What is PHP Server Side Scripting?

Databases and the World Wide Web

Viruses, Worms, & Trojan Horses