

Web Application Security

1. What general security precautions should I take for my web servers running on UNIX or Microsoft Windows systems?

There are a number of precautions you should take. For example, all unused services, command shells and programming language interpreters or compilers should be removed. Web servers should be configured correctly and file permissions should be granted on a need-to-know basis to authorised parties only. System and web logs should also be regularly checked for suspicious activity. In addition, the number of web user accounts that can login to web servers should be properly managed (e.g. ensure that all users select good passwords). User authentication on the web server should be protected by at least SSL/TLS to ensure that passwords cannot be eavesdropped by attackers. Two-factor authentication should also be considered if the system involves sensitive or confidential information.

The following can be observed for enhancing the security of web servers:

- Configure your web server securely according to the vendor's security guidelines
- Run web server processes with appropriate privilege accounts. Avoid running web server processes using full privileged accounts (e.g. 'root', 'SYSTEM', 'Administrator')
- Apply the latest security patches to your web server software
- Configure access rights so that server software cannot modify files being served to users. In other words, the web server software should have read-only access rights to those files
- Install host-based intrusion detection system (HIDS) on web servers storing or processing sensitive information to monitor suspicious activities or unauthorised creation / deletion / modification of files. Alerts and reports from the HIDS should be actively reviewed to identify security attacks at the earliest possible opportunity
- Configure your web server software to prevent any leak of information such as web server software version, internal IP address, directory structure, etc.
- Disable or remove unnecessary modules from your web server software
- Identify application files on the web sever and protect them with access controls

- When using SSL, backup the private key for server certification and protect it from unauthorised access

2. What are the most common web application vulnerabilities, and what are the common safeguards for end-users?

The following are most common vulnerabilities found in web applications:

- Cross Site Scripting (XSS)
- Injection Flaws
- Malicious File Execution
- Insecure Direct Object Reference
- Cross Site Request Forgery (CSRF)
- Information Leakage and Improper Error Handling
- Broken Authentication and Session Management
- Insecure Cryptographic Storage
- Insecure Communications
- Failure to Restrict URL Access

The following are security tips for end-users:

- Don't login to critical web applications from a public computer
- Don't cache your username and password in your workstation
- Remember to logoff at the end of a session
- Use different sets of logins and passwords for different web applications and services
- Regularly change your passwords used in critical web applications if a one-time password is not supported
- Report abnormal behaviour to the service provider immediately
- Ensure that the operating system and system components like Internet Explorer (browser) are fully patched and up-to-date
- Install a personal firewall as well as anti-virus software with the latest virus signatures
- Don't download software or plug-ins from unknown sources

3. Are there any security tips for securing a web application?

Various security controls should be considered throughout the entire development lifecycle of the project:

- Collect together the application security requirements
- Adopt standards or benchmarks according to best practices
- Define secure coding standards to eliminate attacks like SQL injections, and cross-site scripting
- Sanitise application responses to capture all output, return codes and error codes
- Do not trust HTTP referrer headers, client browser parameters, cookies, form fields or hidden parameters unless they are verified using strong cryptographic techniques
- Keep sensitive session values on the server to prevent client-side modification
- Encrypt pages containing sensitive information and prevent caching
- Implement session management
- Implement proper end-user account and access right management
- Restrict access to back end databases, and running SQL and OS commands
- With application system calls, do not make calls to actual file names and directory paths. Use mapping as a filtering layer
- Build a centralised module for application auditing and reporting
- Use the most appropriate authentication methods to identify and authenticate incoming user / system requests
- Create and perform threat modelling
- Design and implement a web application security architecture
- Perform security risk assessment during the development stages to identify the security controls required
- Enforce secure code standards execution
- Perform security tests, such as stress tests, system tests, regression tests, unit tests etc.
- Perform a thorough code review
- Conduct a full security audit before a production launch and after any major changes to the system
- Review application logs regularly
- Implement version control and a separate environment for application development
- Install a web application firewall

4. If web application development is outsourced, is there any checklist I can

use to verify and accept the product?

The following are some examples of areas that might be examined in an assessment of web application security:

Identification and Authentication

- How are users and processes authenticated?
- Is the authentication process implemented in accordance with specifications and in compliance with the security policy of the organisation?
- If the authentication is based on passwords, how are the user passwords being handled and stored?
- Is the password handling mechanism in compliance with the security policy of the organisation?
- Are there any hard-coded passwords or keys embedded in the program source?
- Is the application required to authenticate each and every session?

Data Protection

- Is the data protection mechanism implemented in accordance with the security policy of the organisation?
- Is all data protected adequately at rest?
- Is all data protected adequately in transit?
- If encryption is used, how is the encryption handled?
- Does encryption handling comply with the overall security policy of the organisation?

Logging

- Is the audit trail logging mechanism implemented in accordance with specifications?
- Are the application audit records vulnerable to unauthorised deletion, modification or disclosure?

Error Handling

- How are error messages handled?
- Is there any chance of an information leak that could be utilised in a subsequent attack?
- Would an application failure result in the system entering an insecure state?

Operation

- Are segregation of duties and least privilege principles enforced?
- Have all built-in user IDs, testing user IDs, and IDs with default passwords been

removed from the operating system, web servers and application itself before final production launch?

- Are the system administration procedures, change management procedures, disaster recovery procedures, and backup procedures fully and clearly defined?

It must be emphasised that this checklist is not exhaustive. Depending on the security requirements and specific nature of the target web application, additional test cases or checking criteria should be included according to specific needs.

In addition, when any information system is outsourced to third party service provider, proper security management processes must be in place to protect data as well as to mitigate the security risks associated with outsourced IT projects/services.

5. What are common authentication methods?

There are three basic authentication factors (i.e. "something you know", "something you have", and "something you are") commonly referred to in an authentication system. As a way of tackling the increasing threat of identity theft, two-factor authentication for conducting high-risk e-transactions should be implemented. There are five common authentication methods; namely passwords and PINs based authentication, SMS based authentication, symmetric-key authentication, public-key authentication and biometric authentication. Details of each method is available at the e-Authentication website <
<http://www.e-authentication.gov.hk/en/professional/methods.htm>>.

6. How can I determine an appropriate level of assurance associated with various electronic transactions and their security requirements?

A suggested process flow for business owners wishing to implement a secure e-Authentication system is available at the e-Authentication website <
<http://www.e-authentication.gov.hk/en/business/do.htm>>. You can find more information here on determining the assurance levels and corresponding security requirements.

7. What are the common security risks if I decide to adopt server virtualisation, and what are the security measures to mitigate those risks?

Virtualisation technology allows one or more guest operating systems to run on top of another host operating system. Each guest operating system runs in an emulated environment which is self-contained, isolated and indistinguishable from a real machine. Without adequate protection, virtualisation may increase the security risks faced by an organization.

An example of the common security threat posed by deployment of virtualisation is that security isolation between different systems may be weakened due to virtualisation. After virtualisation, isolation between different information systems may rely solely on correct configuration of the internal virtual network. Incorrect configuration could result in security compromises. Software-based network firewalls able to reside in a dedicated virtual machine (VM) may help mitigate this risk. An alternative mitigation is implementation of a hardware firewall between VMs. In order to do so, all traffic between the VMs will be governed by the hardware firewall. However this approach may have a significant impact on network performance.

Securing a virtual machine involves many of the same best practices needed for securing any operating system. This includes implementing good patch management practices and endpoint security measures, such as anti-virus measures and firewall implementation on both host and guest operating system.

8. How do intruders attack end-users via a web attack?

Key examples of major web attacks that target end-users or their computers are described below:

The 'Italian job' Web attack

In June 2007, more than 10,000 websites, including many Italian government websites, were compromised. Infected websites had a short piece of HTML "iFrame" code inserted that would redirect visitors to another website, where a malicious JavaScript would install a keylogger and a Trojan downloader program on their PCs to test and see if they could be compromised further.

The MySpace Phish / Drive-by attack

Also in June 2007, several hundred MySpace profiles were discovered injected with links to phishing sites. Users of MySpace ran the risk of being infected when they visited any MySpace profile page containing malicious JavaScript that would silently redirect them to a malicious site that would attempt to exploit a vulnerability in Internet Explorer. A commonly known proxy network bot, “flux bot”, would be installed in an attempt to hide the phishing sites behind constantly changing proxy servers.

Cross-Site Scripting (“XSS”) Worms

In October 2005, an XSS vulnerability in MySpace was exploited by the author of the Samy worm who was able to upload his infected XSS code to his personal profile page on MySpace. When other authenticated MySpace users viewed Samy’s profile, the worm forced their web browsers to add Samy as a friend, and alter their profiles with a copy of the malicious code. The Samy worm continued to spread exponentially when a user viewed Samy’s or any other infected users’ profiles. More than one million MySpace user profiles were infected this way.

Other attacks

Phishing can be termed a social engineering attack whereby criminals attempt to lure unsuspecting web surfers into logging into a fraudulent website that looks like a real website, such as eBay, or the website of an online bank. Internet search engines can also help web attacks. In December 2004, the web worm Santy.A exploited a vulnerability in the bulletin board software phpBB. Instead of randomly guessing a target IP address, the worm used the Google search engine to help find new vulnerable targets in order to launch defacement attacks via the vulnerability in phpBB.