Web Application Security

Rajendra Kachhwaha
rajendra1983@gmail.com

July 22, 2015

Outline

What Do We Mean By Security?

1. Security is fundamentally about protecting assets. Assets may be a Web page or your customer database or company's reputation.
2. Security is a path, not a destination. Security is about risk management and implementing effective countermeasures.

Security relies on the following elements:

1. **Authentication: who are you?** It is the process of uniquely identifying the clients of your applications and services. These might be end users, other services, processes, or computers.
2. **Authorization: what can you do?** It is the process that governs the resources and operations that the authenticated client is permitted to access. Resources include files, databases, registry keys and configuration data.

What Do We Mean By Security?

3. **Auditing:** Effective auditing and logging is the key to non-repudiation. Non-repudiation guarantees that a user cannot deny performing an operation or initiating a transaction.
4. **Confidentiality:** Confidentiality is the process of making sure that data cannot be viewed by unauthorized users or eavesdroppers who monitors network.
5. **Integrity:** Integrity is the guarantee that data is protected from accidental or deliberate (malicious) modification.
6. **Availability:** From a security perspective, availability means that systems remain available for legitimate users. The goal for many attackers with denial of service attacks is to crash an application so that other users cannot access the application.

What is a Web Application Security.

1. Web application security is a branch of Information Security that deals specifically with security of websites, web applications and web services.
2. At a high level, Web application security draws on the principles of application security but applies them specifically to Internet and Web systems.
3. Typically web applications are developed using programming languages such as PHP, Java EE, Java, Python, Ruby, ASP.NET, C#, VB.NET or Classic ASP.
4. It is not possible to design and build a secure Web application until you know your threats. An increasingly important discipline and one that is recommended to form part of your application's design phase is threat modeling.

What is a Web Application Security.

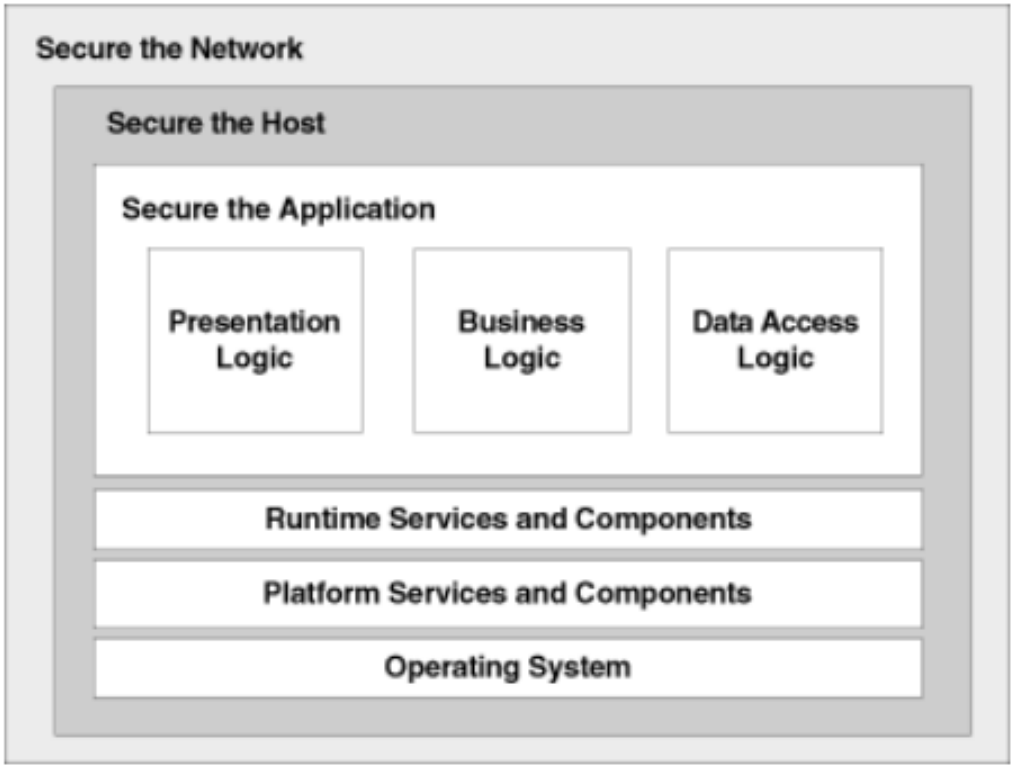
5. The purpose of threat modeling is to analyze your application's architecture and design and identify potentially vulnerable areas that may allow a user, perhaps mistakenly, or an attacker with malicious intent, to compromise your system's security.
6. After you know your threats, design with security in mind. As developers, you must follow secure coding techniques to develop secure, robust, and hack-resilient solutions.
7. The design and development of application layer software must be supported by a secure network, host, and application configuration on the servers where the application software is to be deployed.

Approach to Security

"A vulnerability in a network will allow a malicious user to exploit a host or an application."
"A vulnerability in a host will allow a malicious user to exploit a network or an application."
" A vulnerability in an application will allow a malicious user to exploit a network or a host."
By Carlos Lyons, Corporate Security, Microsoft.

Approach to Security

To build secure Web applications, security must be applied at all three layers. This approach is shown in Figure:



Securing Your Network

A secure Web application relies upon a secure network infrastructure. The network infrastructure consists of routers, firewalls, and switches. The role of the secure network is not only to protect itself from TCP/IP-based attacks, but also to implement countermeasures such as secure administrative interfaces and strong passwords.

The secure network is also responsible for ensuring the integrity of the traffic that it is forwarding. If you know at the network layer about ports, protocols, or communication that may be harmful, countermeasures must be used for those potential threats at that layer.

Securing Your Network

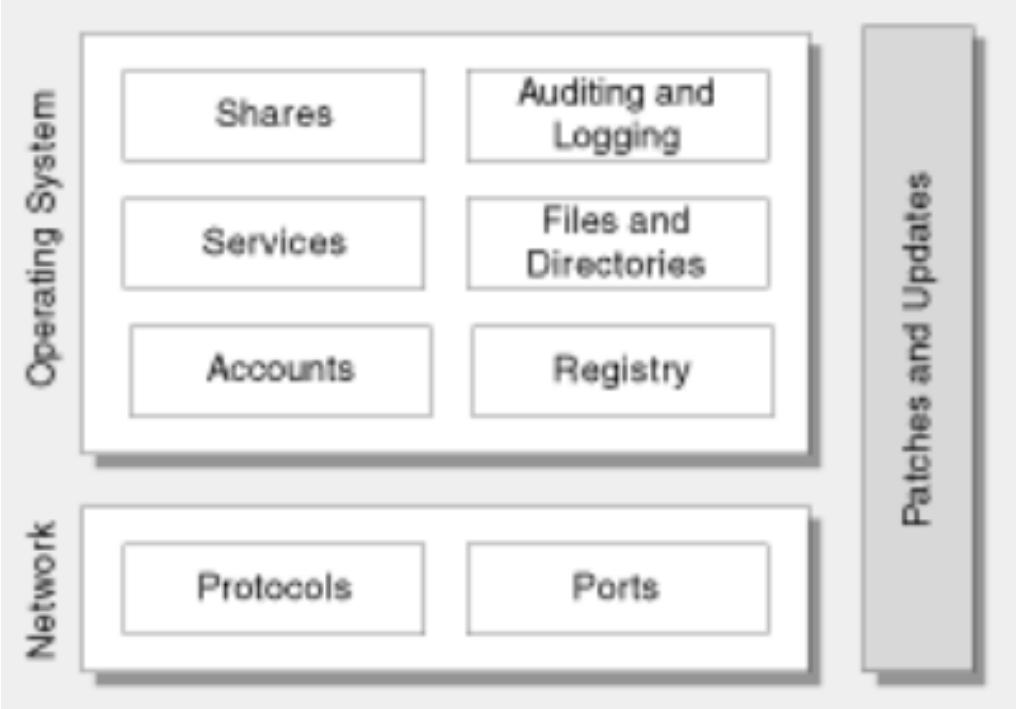
1. **Router:** Routers are your outermost network ring. They channel packets to ports and protocols that your application needs. Common TCP/IP vulnerabilities are blocked at this ring.
2. **Firewall:** The firewall blocks those protocols and ports that the application does not use. Additionally, firewalls enforce secure network traffic by providing application-specific filtering to block malicious communications.

Securing Your Host

When you secure a host, whether it is your Web server, application server, or database server, we breaks down the various secure configuration settings into separate categories. With this approach, you can focus on a specific category and review security, or apply security settings that relate to that specific category. When you install new software on your servers with this approach, you can evaluate the impact on your security settings. For example, you may address the following questions: Does the software create new accounts? Does the software add any default services? Who are the services running as? Are any new script mappings created?

Securing Your Host

Following Figure shows the various categories for securing a host:



Securing Your Host

1. **Patches and Updates:** Patching and updating your server's software is the first step toward securing the server. If you do not patch and update your server, you are providing more potential opportunities for attackers and malicious code.
2. **Services:** The service set is determined by the server role and the applications it hosts. By disabling unnecessary and unused services, you quickly and easily reduce the attack surface area.
3. **Protocols:** To reduce the attack surface area, disable any unnecessary/unused network protocols.
4. **Accounts:** The number of accounts accessible from a server should be restricted to the necessary set of service and user accounts.

Securing Your Host

5. **Files & Directories:** Files & directories should be secured with restricted permissions that allow access only to the necessary service and user accounts.
6. **Shares:** All unnecessary file shares should be removed. Secure the remaining shares with restricted permissions.
7. **Ports:** Open ports on a server must be known and audited regularly to make sure that an insecure service is not listening and available for communication.
8. **Auditing & Logging:** Auditing is a vital aid in identifying intruders or attacks in progress. Logging proves particularly useful as forensic information when determining how an intrusion or attack was performed.
9. **Registry:** Secure the registry itself by applying restricted ACLs and blocking remote registry administration.

Securing Your Application

What better way to measure the security of a application than to evaluate its potential weak points? To measure the security resilience of your application, you can evaluate the application vulnerability, to determine the security strength of an application.

1. **Input Validation:** How do you know that the input that your application receives is valid and safe? Input validation refers to how your application filters, or rejects input before additional processing.
2. **Authentication:** "Who are you?"
3. **Authorization:** "What can you do?"
4. **Sensitive Data:** It refers to how your application handles any data that must be protected either in memory, over the wire, or in persistent stores.

Securing Your Application

5. **Configuration Management:** Who does your application run as? Which databases does it connect to? How is your application administered? How are these settings secured? Configuration management refers to how your application handles these operational issues.
6. **Session Management:** Session management refers to how your application handles and protects interactions between a user and your Web application.
7. **Cryptography:** How are you keeping secrets, secret (confidentiality)? Cryptography refers to how your application enforces confidentiality and integrity.
8. **Auditing and Logging:** Who did what and when? Auditing and logging refer to how your application records security-related events.

Securing Your Application

9. **Parameter Manipulation:** Form fields, query string arguments, and cookie values are frequently used as parameters for your application. Parameter manipulation refers to both how your application safeguards tampering of these values and how your application processes input parameters.
10. **Exception Management:** When a method call in your application fails, what does your application do? How much do you reveal? Do you return friendly error information to end users? Do you pass valuable exception information back to the caller? Does your application fail gracefully?