Role-Based View

This section contains role-based introductions to the CLASP method and provides a high-level view to project managers of how they and their project team should approach security issues. This section also introduces the basic responsibilities they have. These are meant to be concise introductions that are a starting point for employees when they first need to address software security.

Table: Roles and Related Activities

The table below relates the security-related project roles to the 24 CLASP activities. See also “Activity-Assessment View” and “Activity-Implementation View.”

CLASP Activities	Related Project Roles
Institute security awareness program	• Project Manager
Monitor security metrics	• Project Manager
Specify operational environment	• Owner: Requirements Specifier • Key Contributor: Architect
Identify global security policy	• Requirements Specifier
Identify resources and trust boundaries	• Owner: Architect • Key Contributor: Requirements Specifier
Identify user roles and resource capabilities	• Owner: Architect • Key Contributor: Requirements Specifier
Document security-relevant requirements	• Owner: Requirements Specifier • Key Contributor: Architect
Detail misuse cases	• Owner: Requirements Specifier • Key Contributor: Stakeholder
Identify attack surface	• Designer
Apply security principles to design	• Designer
Research and assess security posture of technology solutions	• Owner: Designer • Key Contributor: Component Vendor
Annotate class designs with security properties	• Designer
Specify database security configuration	• Database Designer
Perform security analysis of system requirements and design (threat modeling)	• Security Auditor
Integrate security analysis into source management process	• Integrator
Implement interface contracts	• Implementer
Implement and elaborate resource policies and security technologies	• Implementer
Address reported security issues	• Owner: Designer • Fault Reporter

CLASP Activities	Related Project Roles
Perform source-level security review	• Owner: Security Auditor • Key Contributor: Implementer; Designer
Identify, implement and perform security tests	• Test Analyst
Verify security attributes of resources	• Tester
Perform code signing	• Integrator
Build operational security guide	• Owner: Integrator • Key Contributor: Designer; Architect; Implementer
Manage security issue disclosure process	• Owner: Project Manager • Key Contributor: Designer

Project Manager

Software security efforts are rarely successful without buy-in from the project manager. In most organizations, security will not be a concern to individual project members if left to their own devices. Part of the reason is because the skills required to be effective at secure development do not overlap much with traditional development skills. Another reason is because most development is feature-driven, whereas — beyond basic integration of technologies such as SSL — security rarely shows up as a feature.

The project manager generally has several key responsibilities in this space:

- First among them is promoting awareness. Usually all team members will need to have basic exposure to the application security strategy, and often several team members will need significant training, as few people have the necessary skills in their toolbox.
- Additionally, the project manager should promote awareness outside his team. The rest of the organization needs to understand the impact of application security on the business, such as schedule trade-offs and security risks that the team may not address.
- Another primary responsibility of the project manager is monitoring the health of the organization. Generally, this involves defining a set of basic business matrices and applying them on a regular basis.

Project managers are encouraged to review sections A through F of the CLASP Resources.

Requirements Specifier

The requirements specifier has these major tasks:

- He is first responsible for detailing *business requirements* that are security relevant, particularly those things that will need to be considered by an architect. In most organizations, these two roles will work closely on security concerns and will generally iterate frequently.
- After the team has identified a candidate *architecture*, the requirements specifier should look at the resources present in that architecture and determine what the *protection requirements* for those resources are. CLASP promotes a structured approach to deriving these requirements, categorizing resources into protection levels, and addressing each core security service for each protection level.
- Particularly when using a protection-level abstraction, it is possible to reuse security requirements across projects. This not only saves a tremendous amount of time for requirements specifiers; it also prompts organizations to compare the relative security of multiple projects.
- In organizations that develop use cases, a requirements specifier can also specify misuse cases, which demonstrate to the stakeholder the major security considerations that manifest themselves in the system design. For example, they may document mitigation technologies and how they impact the user, as well as risks that may still be present in a system, thereby allowing the stakeholder to develop compensating controls at an operational level.

Requirements specifiers traditionally do not have the breadth of security expertise necessary to build highly effective security requirements. For that reason, we recommend reading CLASP Resources A, B, C and D thoroughly.

Architect

In an ideal world, the architect simply figures out how — at an architectural level — necessary security technologies integrate into the overall system. This includes network security requirements, such as firewalls, VPNs etc. For this reason, the architect should explicitly document trust assumptions in each part of the system — usually by drawing trust boundaries (e.g., network traffic from outside the firewall is untrusted, but local traffic is trusted). Of course, these boundaries must be a reflection of business requirements. For instance, high-security applications should not be willing to trust any unencrypted shared network media.

Security requirements should come from the requirements specifier. To facilitate better security requirements, the architect should:

- Only need to understand the security implications of technologies well enough that he does not introduce any overt security errors.
- Enumerate all resources in use by a system — preferably to the deepest level of detail possible.
- Further supporting the building of security requirements, he should identify the roles in the system that will use each resource.
- He should identify the basic operations on each resource.
- The architect should also be prepared to help people understand how resources interact with each other through the lifetime of the system.

Designer

The primary responsibility of the designer is to keep security risks out of the application, whenever possible. This responsibility has many facets:

- First, he must figure out what technologies will satisfy security requirements and research them well enough to determine how to use those technologies properly.
- Second, if a security flaw is found in the application, it is usually up to the designer to assess the consequences and determine how to best address the problem.
- Finally, the designer needs to help support measuring the quality of application security efforts. Generally, this involves providing data that can be used as metrics or as a foundation for an application security review.

For example, the designer should explicitly document the “attack surface” of an application — which is roughly equal to the entry points to an application that may be visible to an attacker. This data can be used in a metric roughly akin to traditional software complexity metrics; it is also an excellent starting point for those who are looking to determine whether there are exploitable risks in software.

Designers have the most security-relevant work of all the traditional development roles:

- They should push back on requirements that may have unrecognized security risks.
- They need to give implementers a roadmap in order to minimize the risk of errors requiring an expensive fix.
- They also need to understand the security risks of integrating third-party software.
- In addition, they are generally the point person for responding to security risks identified in the software.

Thus, designers should maintain a high level of security awareness; we recommend reading CLASP Resources A, B, C and D thoroughly.

Implementer

Traditionally, application development is handled in an *ad-hoc* manner, and it is the implementer who must carry the bulk of the security expertise. Ultimately, this is because — in ad-hoc development — developers double as designers.

In a highly structured development environment, most implementers should be building to specification and conferring with designers when there are undocumented considerations. In such an environment, the security responsibilities of a developer are fairly minimal — primarily following coding standards and documenting the system well enough to make it easier for third parties to determine whether the software is as secure as it should be. Sometimes the documentation will be aimed at the end-users, helping to ensure that they know how to use the product securely.

For developers who perform any design tasks, we strongly recommend understanding designer activities by reading Appendices A and B and reviewing the Vulnerability database (Vulnerability View).

Test Analyst

In a structured development organization, security should not have a great impact on the overall processes used. The test organization should still be testing to requirements, implementing regression suites, and so on.

In practice, this will generally require new testing tools that are specifically geared toward security because traditional tools are not good at ferreting out security risks.

Ultimately, beyond tool training and learning about risks well enough to be able to check for them, testing groups do not need to be security experts.

Security Auditor

The basic role of a security auditor is to examine the current state of a project and try to assure the security of the current state of the project:

- When examining *requirements*, the auditor will attempt to determine whether the requirements are adequate and complete.
- When looking at a *design*, the auditor will generally attempt to determine whether there are any implications that could lead to vulnerabilities.
- In addition, when looking at an *implementation*, the auditor will generally attempt to find overt security problems, which should be mappable to deviations from a specification.

Rarely is being a project security auditor a full time job. Often, developers with a particular interest or skill in security perform auditing. Sometimes, organizations have an audit organization focused on other regulatory compliance, and these people will perform security review.

It is usually better to avoid reviewing one's own designs or one's own code since it can be difficult to see the forest for the trees.