Measure and Monitor Application Security

Sivarama Subramanian, CISM, is senior architect of technology at Cognizant Technology Solutions, where he is currently overseeing the security initiatives for retail and retail e-commerce engagements. Subramanian is a member of the ISACA Chennai, India, Chapter and can be reached at *sivaramasubramanian.kailasam@cognizant.com.*

In the increasingly digitally connected world, information is the most valuable asset. Web applications are the gateway to access information, and they are no longer confined to a simple browser interface invoked from a laptop or desktop. With smartphones and smart TVs (televisions with a Wi-Fi or network connections) and appliances, applications are available everywhere.

This thrusts a lot of responsibilities on the security community to safeguard the interests of stakeholders and the information that is exchanged via web applications. The security community[1] has published best practices, guidelines and checklists to embed security into web applications. For example, one of the best practices is to call out specifically how to handle the SQL injection or to handle the cross-site scripting in the design document itself so that when the developers implement the design, the security is inherent in the code. Securing the system development life cycle (SDLC) is no longer a separate activity.

How does one ensure the effectiveness of application security? How are the security initiatives that minimize the risks and threats from hackers measured? This article attempts to define metrics that measure the effectiveness of application security in an organization.

DEFINE THE METRICS

Metrics are the prime indicators of management initiatives in any organization. Organizations witness a slow, but steady, increase in need for information security within. In many organizations, information security has attained a fairly considerable level of maturity. Web application security, as a part of the overall information security program, plays a major role in protecting valuable information. Therefore, the best time to define a metric is at the start of the application security program. The best possible approach is to:

- Identify the metrics.
- Identify the data-collection techniques.
- Obtain agreement from key stakeholders.
- Report the metrics to key stakeholders at the agreed-upon time interval.

The identified metrics should be useful to measure the effectiveness of the security program as well as to identify the gaps for future improvement.

There are two broad categories of metrics that can be captured for application security. The first set of metrics is for incidents and vulnerabilities (**figure 1**), and the second set is for the application security program itself (**figure 2**).

Figure 1—Metrics for Incidents and Vulnerabilities

Metric	Purpose
Number of incidents reported	Represents the number of incidents reported or discovered in the measurement window and helps identify the up/down trend of incidents
Number of incidents resolved	Helps identify the up/down trend of resolutions. Downtrend can be investigated, and timely corrective action can be taken.
Number of vulnerabilities reported	Represents the number of vulnerabilities reported or discovered in the measurement window and helps identify the up/down trend of vulnerabilities
Number of vulnerabilities resolved	Helps identify the up/down trend of resolutions. Downtrend can be investigated, and timely corrective actions, such as awareness training or focused reviews, can be taken.
Total security effort	Helps identify the effort spent on all the security activities. The idea is to reduce the effort by following a secure SDLC program.
Average effort—vulnerability assessment	Helps identify where the effort is being spent for resolving the incidents and vulnerabilities. The idea is to reduce the effort by following a secure SDLC program.
Effective ratio (number of reported vulnerabilities/number of found vulnerabilities)	Measures the defect leakage of the security program. If the value of the ratio is more than one, the program is not effective.

Do you have something to say about this article?

Visit the *Journal* pages of the ISACA web site *(www.isaca.org/journal)*, find the article, and choose the Comments tab to share your thoughts.

Figure 2—Metrics for the Application Security Program

Metric	Purpose
Number of proactive scans	Indicates how many automated scans are done for the web applications in the measurement window
Number of security awareness sessions	Indicates how many training sessions are conducted to impart the current trend of security risks and also the security awareness for new employees
Design review ratio (number of design reviews/total number of designs delivered)	Indicates whether all design documents have been reviewed. For example, if there are four design documents and only three reviews held, it means that the metric value is 0.75, or 75 percent. The metric value should be 1, or 100 percent to indicate that all the designs are reviewed.
Code review ratio (number of code reviews/total number of modules)	Indicates whether all source code has been reviewed. For example, if there are four modules and only three reviews held, it means that the metric value is 0.75, or 75 percent. The metric value should be 1, or 100 percent, to indicate that all the modules are reviewed.
Number of vulnerabilities prevented from proactive activities	Quantifies the effectiveness of the security reviews
Number of articles added to the knowledge base	Helps track updates to the knowledge base

COLLECT THE DATA

The data for the metrics should be collected on an ongoing basis (e.g., weekly, per event). The data collection template needs to be defined and kept in a centralized repository. As soon as the reviews are done, the review metrics can be updated in the data-collection template.

The incident's data can be collected from the incident database. The vulnerabilities and effort can be collected from the project team. The reviews and comments should be updated in a shared repository, and data are to be reported from the repository.

REPORT THE METRICS

At the end of the month, or as per the agreed-upon cutoff date with the project team, the data can be collated and presented. A few presentation examples are illustrated in **figures 3** and **4**.

Figure 3—Total Security Effort

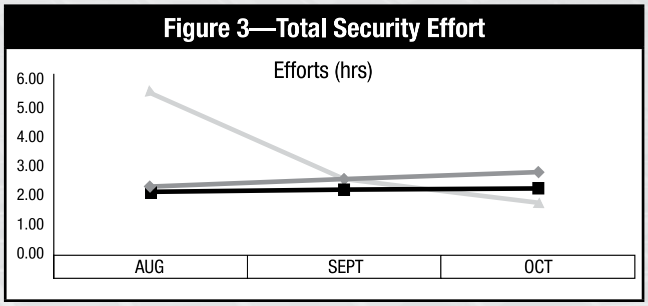
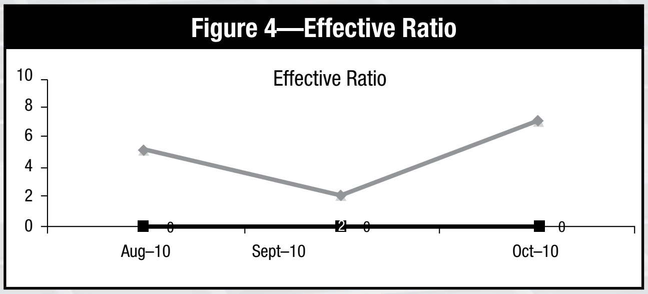


Figure 4—Effective Ratio



CASE STUDY

The following section offers a brief case study related to application security.

Problem Statement

The problem statement for the case example is: Manage the application security for the e-commerce applications of a pharmaceutical company by measuring the key metrics associated with application security.

The key metrics agreed upon for the measurement include:

- The number of vulnerabilities reported
- The number of vulnerabilities resolved
- Total security effort
- The number of security awareness sessions
- The design review ratio (number of design reviews/total number of designs delivered)

- The code review ratio (number of code reviews/total number of modules)

Implementation

After the key metrics were agreed upon by the senior management team, the kickoff meeting was scheduled with the project leadership team. The metrics were explained to the team, and a monthly measurement window was chosen. The data collection techniques were explained, and the schedule was given to the project leads.

At the end of the first measurement cycle, the data were collected and collated as shown in **figure 5**.

Figure 5—Data Collection and Collation

Metric	Units	December 2010
Number of vulnerabilities reported	Count	18
Number of vulnerabilities resolved	Count	15
Total security effort	Hours	80
Number of security awareness sessions	Count	1
Design review ratio (number of design reviews/total number of designs delivered)	Percentage	75
Code review ratio (number of code reviews/total number of modules)	Percentage	50

Business Benefit

The metrics were presented in the project-review meeting, and the project manager approved the data. **Figure 5** indicates that the secure code review process needs to be institutionalized; this would enable the project team to take corrective action and reduce further vulnerabilities. However, metrics should be collected for subsequent months to understand long-term trends.

CONCLUSION

With more stringent security controls in place for the infrastructure and networks of organizations, hackers are turning their attention to web applications. Through the vulnerabilities of web applications, the network and infrastructure can be compromised. Along with the increased adoption of cloud computing, there is more attention given to application security. The metrics discussed in this article should help organizations measure their application security postures. In this age of information, an organization needs to safeguard its information to have a competitive edge and to win the trust of key stakeholders.

ENDNOTES

[1] By “security community,” the author is referring to groups such as the Open Web Application Security Project (OWASP), the Cloud Security Alliance and social media outlets.