



# Software Immunity

**Manjula Sridhar** Founder and CTO, Aujas Networks

Software security is increasingly becoming the focus of the security industry. Research sources suggest that 75% of the new hack attempts are targeted at Software Security and 90% of the vulnerabilities are in software. This white paper presents a very high level overview of software security and various aspects that contribute to it. It dwells on the risk modeling as applicable to software security and suggests the remedies and counter measures from process, technology and process perspective.

Security industry is sometimes blamed for running on the FUD (Fear, Uncertainty and Doubt) factor. While this might be true in some instances, software security is one area where the threat is grossly underestimated. The magnitude of the problem warrants a fresh look at the way software is built and the perception of security being an external control. Jeremiah Grossman, founder of Whitehat, presents a staggering statistics to drive home the point. (See the box for data).

- As of 2008, there are 17 million programmers worldwide;
- Average of 6000 lines of code each year, amounting to 102 billion new lines of code;
- Conservative estimates state 1 defect for 10000 lines of code. This means NEW undiscovered vulnerabilities of 10200000 per year, 850000 per month and 28000 per day;
- If only 1% of these vulnerabilities are exploitable this amounts to 102000 Zero days per year.

**Figure 1: Software Security Statistics**

This becomes immensely important when you consider the fact that threat landscape has evolved from external to internal hacking and hacking itself has evolved from thrill seeking script kiddies to organized syndicates going after big bucks.

### Source of Software Security issues

Security bugs happen due to various reasons.

#### Willful trade-off

Software developers are a heckled lot. Apart from the technology issues of making a program work, there is, in many cases, an immense pressure in terms of –

- Time to market
- Performance of the systems and
- Quality issues of the product.

Security is considered as an afterthought if it gets considered at all. Processing power is always a costly resource when you are dealing with millions of transactions and security is always looked at as an additional feature and is often willfully sacrificed for better performance.

### Misconceptions

Even in cases where security is considered important as part of the overall architecture, it is designed as part of the external control, rather than designing it as an in-built feature. This approach stems from the wide-spread perception, that security is the responsibility of network engineers rather than that of software developers. Also many systems are designed keeping external threats in mind and hence ignore the more serious aspect of insider misuse.

### Lack of awareness

The last one is usually the most dangerous sort as the developers are hardly aware that functionally correct and seemingly innocuous code could be potentially utilized for serious exploitation. (See the list below of software code issues that lead to security vulnerabilities).

MTRE, SANS and OWASP have done extensive analyses on the vulnerabilities in the software industry and have published exhaustive list of root causes that lead to them. Most of these issues boil down to very simple coding errors which could have been easily avoided.

The need for awareness is at multiple levels. While making software developers aware of the coding and review guidelines, it is also important to have the overall process and management policies support the initiatives. This needs to get extended to third party software and off-the-shelf software that gets integrated into the systems.

- A1 - Cross Site Scripting (XSS)
- A2 - Injection Flaws
- A3 - Malicious File Execution
- A4 - Insecure Direct Object Reference
- A5 - Cross Site Request Forgery (CSRF)
- A6 - Information Leakage and Improper Error Handling
- A7 - Broken Authentication and Session Management
- A8 - Insecure Cryptographic Storage
- A9 - Insecure Communications
- A10 - Failure to Restrict URL Access

**Figure 2: OWASP's List of top 10 security vulnerabilities**

## Why address software security?

### Not addressing is not an option

As the case TJ Max, Hanford and scores of others have proved software security issues are increasingly utilized for serious crimes. Gartner estimates that 90% of the existing vulnerabilities are in software and 75% of all new hacking attempts are targeted at software vulnerabilities. While the financial losses itself could be staggering, it also destroys brands and could lead to potentially serious legal implications.

### Regulations

Many regulations, such as PCI, have mandated secure coding as one of the requirements to do business with PCI industry. In India, RBI requires the application security concerns to be addressed. In many cases software security is a pre-requisite for purchase and many customers are engaging third-party assessors to come up with security ratings for the off-the-shelf software as part of purchase agreements.

### Business Value Driver

Secure software development practices are a differentiator in the competitive markets and helps software enterprises to move up the value chain. This is relevant in case of the crowded outsourced software development market, as well as the product vendors addressing financial and other security sensitive industries.

## How much security is enough?

While everyone understands the needs of security, the cost and complexity involved in making the software secure is an essential parameter. ROI for security has always been an issue of consideration among business, among management and policy makers. Therefore it is essential to conduct the risk analysis of software applications.

Many Software Risk assessment models exist which assess the software risks against the threats in terms of severity, likelihood of occurrence and financial implications of the risks.

Once this analysis is done it becomes easier to assess the security controls that can be built into software for adequate protection.

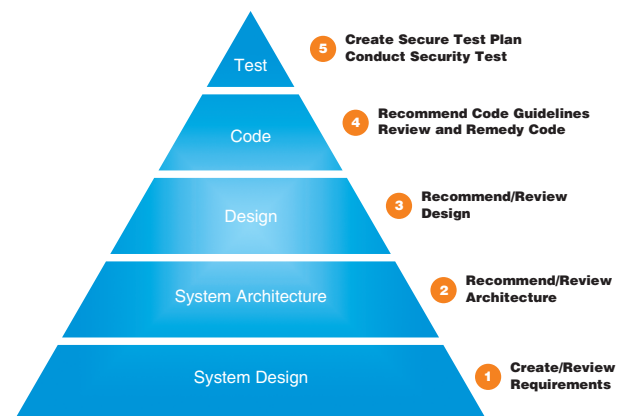
## How do we address this?

### Stakeholder Awareness

Disseminate the importance security in software community. This should be done at all layers of SDLC (Software Development Life Cycle) stake holders as well as business and policy decision makers.

### Move from SDLC to Secure SDLC

Security cannot be addressed in isolation. It needs be part of the regular development cycle. A simple technique like having a place holder for security in templates like design documents, requirement documents that will force the software designers and developers to think about various security aspects.



### Customize the best practices to your environment

While understanding the potential threats from top vulnerabilities is important, it is essential to map them to SDE of your enterprise. Vulnerabilities are usually a mix of OS, databases, languages, third party libraries coupled with custom code. Therefore it is critical to understand the coding patterns in a product and provide ready reference for implementation.

[www.aujas.com](http://www.aujas.com)

## About Author

---

Manjula Sridhar is the co-founder and CTO of Aujas Networks, a Digital Life-cycle Security Services company. She has managed large product development groups at Alcatel-Lucent and Bosch Telecom and was part of the Tiger team at Lucent which disseminated Secure product development practices. She has done extensive work in software patterns and automated code generation in which area she has filed a number of patents. She is a B.E, M.S (Network Security) and holds CISSP certification.

The author blogs on key security issues at [blog.aujas.com](http://blog.aujas.com)

## About Aujas

---

Aujas Networks is a Digital Life Cycle Security Company. One of its key focus areas is Application Security. Aujas offers customized secure SDLC services across various platforms and technology frameworks and aids in meeting industry and regulatory compliances.