МНОГОУРОВНЕВЫЙ ПОДХОД К ОЦЕНКЕ БЕЗОПАСНОСТИ ПРОГРАММНЫХ СРЕДСТВ

Грегори Рибер
Кеннет Малмквист
Алексей Щербаков

Статья посвящена вопросам планирования процесса оценки безопасности программных систем с учетом особенностей присущих информационной среде конкретной организации. Делается анализ процесса оценки безопасности в контексте различных бизнес–факторов, таких как требуемый уровень безопасности, бюджетные ограничения, окупаемость инвестиций и т.д. Дается общая характеристика типов уязвимостей программного обеспечения и описываются основные методы тестирования безопасности программных систем: динамический и статический анализ программного кода. Рассмотрены достоинства и недостатки автоматического и ручного режимов статического анализа. Предлагается пример подхода к планированию процесса оценки безопасности программных систем путем создания набора уровней оценки безопасности, где каждый уровень представляет собой комбинацию нескольких методов анализа программного кода. Статья будет интересна руководителям и специалистам в области безопасности информационных систем осуществляющим разработку, планирование и управление процессом обеспечения безопасности программных средств предприятий и организаций.

***Ключевые слова**: безопаснось программных средств, уязвимость, управление рисками, оценка безопасности, статический анализ, динамический анализ*

MAPPING THE APPLICATION SECURITY TERRAIN

Gregory Reber
Kenneth Malmquist
Alexey Shcherbakov

Enterprise application security requirements, vulnerability types, discovery methodologies, and various application assessment strategies are considered. A multilevel approach to application security assessments is described.

***Keywords**: application security, vulnerability, risk management, security assessment, static analysis, dynamic analysis*

Introduction

Network perimeter security has become more and more effective as products and services have matured. Internet applications are now the target of choice for criminals to obtain restricted information and unwarranted access to companies' protected assets. The number and type of protection measures for these applications is growing. The selection of an appropriate application security risk management solution should take into account the business's diverse requirements and factors. There is no single solution that will fit every company's needs.

Those responsible for the security of their environments need to understand what risks are present in their applications, as each vulnerability has an associated criticality that is based on various factors. Armed with this knowledge, an appropriate risk management strategy can be developed with prioritized action to reduce these threats.

Modern economies are characterized by increasing variety of enterprises of different size, structure, and specialization. Every organization is different to certain extent: it has different business needs, different information systems, and different security requirements. It is quite logical to propose a set of service levels in the realm of application security assessments from which the management could choose the most cost-effective type or level of service that would match the organization's business needs and security requirements.

So what is the required level of application security assessment?

As enterprise application security requirements are considered, it is useful to put them in the same context as various other software attributes that we usually deal with:

- Functionality
- Usability
- Performance
- Reliability
- Security

However, we can't deal with the characteristics of our applications in isolation; we consider them in the context of business requirements and real world business factors including feasibility, funding, return on investment, and opportunity cost.

While better is always desirable, we can't evaluate what is better without understanding the status quo. We need to answer the "better than what?" question. This requires sufficient analysis/assessment to identify a comparative baseline.

For example: A company's web-facing newsletter sign-up page is found to have a Cross-Site Scripting vulnerability. Addressing this risk may require $20,000 in development costs. Is this the best use of funds for this company?

In theoretical terms we want absolute safety. In practical terms we want a "reasonable or better" level of security. The definition of "reasonable" is only meaningful within the context of a specific application and business. The definition may be based upon government (e.g. DoD levels of classification [1]), industry group requirements (PCI DSS [2]), and business domain.

The very act of measuring security, performance, or reliability has an associated variable cost based upon the precision and thoroughness of the analysis, the skills of the analysts, etc.

An application security assessment process is the method of identifying application security vulnerabilities so that the business can make informed risk management decisions that include the evaluation of the financial and opportunity costs associated with mitigating the identified security risks. The thoroughness, depth, and cost of an application security assessment process should reasonably vary with business requirements.

Now that we familiarized ourselves with a high level overview of the application security space let's discuss the different types of security vulnerabilities and discovery methodologies.

What types of security risks should be considered?

A useful starting reference point is the vulnerability taxonomy maintained by OWASP, the Open Web Application Security Project. There, one can find hundreds of articles defining common application security flaws. OWASP also maintains a Top 10 list [3] of the most critical web application vulnerabilities. While the OWASP Top 10 list is a very useful document to increase security awareness, like most lists of this sort, it is neither intended to be comprehensive nor a sufficient definition of application security. AsTech maintains a more wide-ranging catalog of vulnerability classes which we have developed over the past 15 plus years.

There are a number of approaches to assessing application security involving varying combinations of automated and manual analysis from an external (black box) and internal (white box) perspective.

External Web Application Scanning

Dynamic application scanning involves interacting with a running application (essentially using and attacking the application) as a black box to identify points of vulnerability. While the best of breed commercial automated scanning tools can produce some valuable results, they still can't approach the quality and breadth of results that can be identified by a highly skilled ethical hacker.

The strength of application scanning is that because the application is actually attacked, the resulting proof of vulnerability is usually quite concrete and compelling. For example, the results of a successful SQL injection attack might include data or metadata accessed without authorization. If you can see another user's account data or display the structure of the database, it is hard to argue with the existence of the vulnerability.

The weakness of application scanning is that it identifies only a limited range of vulnerabilities and often requires a highly skilled practitioner. Since the application user interface is the attack vector, the approach is ill-suited to examining business component, back-end,

or external service vulnerabilities. For example, if sensitive data such as social security numbers are not being encrypted, or third-party services operate without proper protection, or critical security events such as failed logins are not being adequately logged, these vulnerabilities are likely to go undetected.

Automated Static Analysis

Static analysis involves the review of the application code for vulnerabilities. For most tools, this usually refers to the source code but less frequently refers to the binary code. This would be considered a 'white box' assessment, as nothing is hidden from the analyst. The application code is a much larger and richer analysis target than the user interface addressed by external, or 'black box' application scanning, and therefore a broader range of vulnerabilities can be identified.

The best of breed static analysis tools utilize sophisticated compiler technologies such as data flow analysis, control flow analysis, and pattern recognition to identify security vulnerabilities. The results of automated analysis generally include a high degree of false positives, requiring a highly skilled security engineer to analyze the results with the source code in hand to distinguish between the truly and the falsely reported vulnerabilities.

Each type of application security analysis tool has its strengths and weaknesses. Thorough understanding of these strengths and weaknesses is crucial for implementing a successful application security program using the right tools.

What are the strengths and weaknesses of Static Analysis?

Static analyzers are best at identifying vulnerabilities that can be represented as identifiable patterns. Examples of these risks include:

- A missing entry in an XML configuration file
- The use of a dangerous function, including non-validated user input data in a web page
- Output (Cross-Site Scripting vulnerability)
- Including non-validated input data in the construction of a database query (SQL Injection vulnerability)

Automated Static Analysis

Most static analysis tools can also identify a range of poor programming practices such as the use of uninitialized variables or the lack of error handling.

The main stren gth of automated static analysis is that the analyzers reliably identify candidate issues (which could turn out to be false positives) and can do so in the face of highly complex application structure and control flow that might daunt most humans. For the software expense and the skilled labor required, the results can be quite cost effective.

However, the main limitation of these automated tools is that currently they can only find approximately 50%-80% of the types of security vulnerabilities that should be evaluated in a security assessment to provide a comprehensive view of risks present in an application. With the current state of the technology, automated analyzers are generally not capable of testing algorithms, security policy adherence, and issues that may be derived from the application domain. Examples of these areas include:

- Authentication
- Authorization
- Disclosure of confidential data
- Audit logging
- Cross-Site Request Forgery (CSRF)
- Identifying application 'back-doors'

Manual Static Analysis

Manual static analysis involves a review of the application architecture and source code by highly skilled software security engineers. The resulting analysis is comprehensive and is, overall, the most reliable of the approaches. Thus it has been the method of choice where application security is of paramount concern, such as most financial services organizations.

The strength of manual analysis is the level of depth and thoroughness of the assessment. The full range of security vulnerabilities can most readily be identified with high reliability. Specific attributes of the application domain (credit card numbers, account numbers, classified data, etc.) can be taken into account.

The main drawback of manual analysis is that engineers with the necessary skills and experience – both extensive enterprise application development experience coupled with deep security knowledge – are scarce and in high demand. The time required and the level of effort involved makes this approach more costly than other options.

Vendor Claims

Predictably, vendors of specific technologies or services tend to tout the strengths of their specific approaches and diminish the value of the alternatives. The vendors of automated static analysis tools promote their cost effectiveness and minimize the importance of potentially material coverage gaps, which as we have shown may be significant. Providers of purely manual assessment services tout comprehensive coverage and minimize the impact of cost and schedule.

Of course, there is no 'one size fits all' approach to application security. A sound risk management strategy will make the most appropriate use of any available technology or process.

A multilevel approach

There are many different types of applications in use today, encompassing myriad functionalities and business purposes. Therefore, there can be no 'one size fits all' approach to risk management when contemplating application security. An internally utilized client-server application that tracks office equipment purchases will not have the same security requirements as those of a publicly accessible banking application.

Previously, we described the relative effectiveness of various assessment methodologies at discovering risks.

Now, let's think about the ways to use that knowledge to efficiently identify and plan security assessments for various types of applications taking into account the depth of analysis coverage, the costs, and the residual risk. Among several possible ways to approach this task we chose as an example a method that we will call a multilevel approach to application security assessments. The main idea behind the method is creating a set of levels of security assessment based on the range of "white box" risk discovery options. Each level can be described as a combination of an automated static analysis of the source code and a manual code review. For an added level of verification any of these levels can be further enhanced by adding an external or "black box" vulnerability assessment in the form of a manual or automated dynamic analysis.

1. Comprehensive Assessment – Automated analysis with complete manual analysis

To obtain the most comprehensive results and ensure the lowest residual risk, this level employs automated source code static analysis tools to identify a preliminary set of vulnerabilities and a full manual analysis of the source code for types of vulnerabilities not reliably found through automated tools. This level is most appropriate for commercial applications that have the highest security requirements such as applications involving a high volume or high value financial transactions.

2. Perimeter Assessment – Automated analysis with attack surface manual analysis

To provide breadth of analysis while lowering cost, this level employs automated source code static analysis tools to identify a preliminary set of vulnerabilities. The preceding phase is followed by a manual analysis focused on those areas of the source code that represent the greatest risk for types of vulnerabilities not reliably found through automated tools. Representative areas of focus include the code representing the attack perimeter of the application such as user interfaces and use of external services as well as authentication, authorization, and data protection. Since the manual review is somewhat limited, there is some amount of residual risk with this approach.

3. Perimeter Audit – Automated analysis with attack surface manual audit

To further reduce cost but still provide some breadth of analysis, this level employs automated source code static analysis tools to identify a preliminary set of vulnerabilities and a manual audit of the source code focused on those areas of the source that represent the risk for types of vulnerabilities not reliably found through automated tools. The auditing process samples a portion of the code which is taken to contain representative examples within the range of vulnerabilities present in the application. Since the manual review is even more limited, there is a greater level of residual risk.

4. Automated Assessment – Automated static source analysis audit

To minimize expense while obtaining some reliable level of security assessment, automated static analysis of the application is performed and validated. This provides a reasonable assessment for some of the most frequent critical vulnerabilities such as SQL Injection and Cross-Site Scripting. However it leaves other key areas not addressed by automated analysis unassessed. The level of residual risk is therefore higher still compared to other approaches and thus may not be appropriate for an application that is business critical.

	Level of Risk Identification	Relative Cost	Resulting Residual Risk
Comprehensive Assessment	Highest	Higher	Lowest
Perimeter Assessment	Higher	Moderate	Low
Perimeter Audit	High	Low	Moderate
Automated Assessment	Moderate	Lower	Significant

Conclusions

Every day, more threats and exploits against Internet applications are being discovered. Many applications contain vulnerabilities that haven't been discovered by those responsible for securing these systems, rendering it impossible to implement effective risk management strategies. There are more than a few options available to identify these vulnerabilities, but the decision of which to use in a given business environment can be complicated, since every option has its pros and cons. The multilevel approach presented above can help any organization to identify the right scope of application security assessment within the available budget and ensure the application security needs are met.

AsTech Consulting has been performing application security assessments for top-tier clients since 2001. Our assessment processes combine the skills of some of the industry's best security engineers with the best of class automated analysis tools. We continually modify our processes to take into account the improvements in automated tools, the changes in threats, and industry standards and best practices. Our goal is to deliver the most effective application security process possible, based on each client's unique risk appetite and business objectives.

References

1. DoD Information Security Program: Overview, Classification, and Declassification, DoD Manual 5200.01, Volume 1, p.34. US Department of Defense. 2012.
2. Payment Card Industry (PCI) Data Security Standard, v3.0. PCI Security Standards Council. 2013.
3. OWASP Top 10 – 2013. The Open Web Application Security Project. 2013.