Reverse Engineering Vulnerability Assessment

A report on the current practices to manage risks that stem from unauthorized access to source code via reverse engineering.

Publication Date: January 15, 2007

Abstract

This report presents an analysis of the responses to the Application Risk Assessment Survey distributed in November, 2006 to over 20,000 corporations. This benchmark offers organizations a means to estimate their exposure to risks that stem from uncontrolled access to source code via reverse engineering and to evaluate the controls and tools that are currently in place to mitigate those risks.

1 The survey questions are included in Appendix A
Executive Summary

This paper provides the first comprehensive assessment of the risks that corporations are facing that stem from the reverse engineering of managed code, e.g. .NET or Java applications, as well as an evaluation of the current state of the practices and controls in use today.

Risks

Easy access to source code via reverse engineering tools for Java and .NET dramatically increases the risk of:

- **Intellectual Property (IP) theft**: source code can be extracted from .NET and Java binaries in seconds enabling competitors to immediately capitalize on your R&D investments and competitive practices.
- **Source code vulnerability exploitation**: simple scripts can locate, reverse engineer and probe for vulnerabilities automating the vulnerability search process and generating thousands of exploitation opportunities per day.²
- **Privacy violations and information loss**: once information systems are compromised through vulnerability exploitation or de-compilation, modification and recompilation, all of the information that these systems manage are at risk for disclosure or corruption.
- **Revenue loss**: software piracy or the subversion of applications that restrict access to other products and services, e.g. financial, Internet, et al result in hundreds of millions of dollars of revenue loss annually.

Disconnect between development community and business stakeholders

In a recent survey³ of Java and .NET developers, 75% of the respondents indicated that their management would modify their company’s IT controls (and in some cases suspend .NET and Java development until those controls were implemented) if they understood the risks of distributing and relying upon applications distributed “in the clear.”

Dozens of industries are at risk

Roughly two thirds of those organizations who are at the greatest risk do not have a consistent set of controls and/or technologies to mitigate these risks. The following analysis examines the trends and gaps on an industry-by-industry basis.

Recommendations

- Formally recognize these risks and assess their materiality.
- Where appropriate, consistent IT policies must be established and communicated.
- Effective processes and technologies must be selected and made available for effective risk mitigation.
- Monitor emerging technologies and practices to ensure that current IT controls remain aligned with the evolving patterns of risk.

The following analysis offers organizations a means to estimate their exposure to risks that stem from uncontrolled access to source code via reverse engineering and to evaluate the controls and tools that are currently in place to mitigate those risks.

---

² For more information on this threat and additional detective controls from PreEmptive Solutions, email solutions@preemptive.com

Introduction

Java and .NET platform adoption is accelerating as organizations recognize the advantages that managed code can offer. As .NET and Java applications become increasingly ubiquitous, security, compliance and risk professionals need to consider to what extent a greater reliance on managed code may impact their existing assumptions about security, compliance and operational performance. One well-understood side-effect inherent in these two environments is the relative ease in which source code can be extracted (reverse engineered).

Reverse engineering is now a common practice
There are over 500,000 copies of source code extraction tools currently in use worldwide. Ready access to these tools have made reverse engineering a common practice for support, education and debugging activities. Reverse engineering is no longer the exclusive domain of the black hat.

The risks associated with reverse engineering are well-understood but poorly managed
In a recent survey that included over 300 corporations, the primary risks identified from reverse engineering were Intellectual Property (IP) theft (37%), application vulnerability exposure (26%) and revenue loss (24%). Yet, 75% of those responding indicated that if their senior management understood the risks associated with reverse engineering, they would modify current IT controls to ensure that the risks were managed effectively and consistently. In other words, there is a knowledge gap within 75% of those companies surveyed. Do your technologists know something about your risk that you don’t?

Obfuscation is a widely accepted compensating control
Obfuscation is a component of the development process that modifies .NET and Java binaries to make source code extraction materially more difficult for humans and machines without adversely impacting application functionality, performance or development productivity. Obfuscation is widely adopted across industries and geographies and is routinely recommend as a compensating control by IT auditors.

Disclosure: PreEmptive Solutions is an obfuscation technology supplier
PreEmptive Solutions licenses an enterprise obfuscation solution. With over 3,000 corporate clients, inclusion in Microsoft’s Visual Studio development platform and tens of thousands of active users, PreEmptive Solutions has been witness to a dramatic shift in development practices and a heightened emphasis on application security and compliance.

Our purpose in offering the following analysis is to provide a reliable resource to assist organizations who want to appropriately and cost effectively manage a category of risk that, while widely appreciated amongst technologists, is still under-recognized and poorly managed at the enterprise level. For more information on enterprise obfuscation, including technical and process requirements, download http://www.preemptive.com/documentation/EnterpriseObfuscation.pdf.
Survey background

The information reflected in this research includes verified responses from over 700 companies. Every organization included in these results has at least 20 dedicated developers. Industry data is included only if there are at least 5 complete company responses within that industry. All responses have been verified. These responses were collected from a larger community that included:

- 10,000 individuals from North American corporations with over $100,000,000 USD in annual revenue.
- 15,000 individuals who have registered with PreEmptive Solutions indicating an interest in either Java or .NET development.
- 3,000 PreEmptive Solutions corporate client contacts in over 50 countries.

Content

The survey examines three aspects of managing the risks that stem from the reverse engineering of .NET and/or Java applications.

I: Exposure and materiality of risk

Exposure and materiality of risk can be influenced by:

- The size, complexity and distribution of application development within and across organizations.
- The role or use of applications within critical or material operations or business functions.
- The extent of regulatory obligations and oversight.
- Application access to sensitive information.
- The revenue or value associated with the applications’ source code.

II: Availability of preventative measures to reduce exposure

The availability of preventative measures such as obfuscation reduces the likelihood of applications being reverse engineered (but not the materiality of the risk). In addition, good development practices that reduce the likelihood of vulnerabilities being introduced further reduces overall risk by, in some cases, reducing the potential for damage (but not IP theft or revenue loss).

III: Practice of processes and controls to ensure that protective measures are consistently, effectively and appropriately applied

Obfuscation and other means of reducing the risks associated with uncontrolled access to source code act as compensating controls. However, if controls are not consistently applied, organizations will either not achieve optimal levels of risk mitigation or they will incur unnecessary expense as controls are over used. All of this can be influenced by:

- Overall sophistication and implementation of IT controls.
- Incorporation of application vulnerability management strategies at all phases of the application development lifecycle.
- Specific controls and documentation addressing reverse engineering risk.

How to apply these results

The following sections assess specific risks and mitigating practices across comparable organizations. Substantial differences between your organization and

---

4 Throughout this report, the use of the term “application” refers to applications developed with managed code, e.g. .NET or Java.
comparables may indicate material differences in how technology is being deployed, risk is being managed or IT practices are being governed. Understanding these differences can lead to improved performance and more effective risk management. If your organization appears to be an outlier, it is possible that you are at greater risk for:

- Becoming a target of attackers and facing increased risk as a result or;
- Appearing to create an unnecessarily permissive environment that may increase liability in the eyes of auditors, regulators or the courts.

Survey results

When determining risk ratings, a “baseline risk level” is determined. The base risk is the total risk without any consideration for mitigating controls or practices. The score is calculated by assigning a weight to the various factors reported in the Exposure and materiality of risk section of the survey.

This score is then reduced first by the availability of measures to mitigate these risks within an organization and then by the effectiveness of the processes and controls in place to ensure that those measures are applied in a consistent and effective manner.

Industry trends

The results showed that reverse engineering of applications can pose a material risk to organizations within virtually every industry segment. However, there are differences in the average baseline risk level and the degree of mitigation (best practices) across industries. These differences in risk and risk management provide important insight into the relative role of managed applications at this point in time across industries and the maturity of their respective risk management processes.

Baseline Risk across industry segments

The following factors drive baseline risk:

- **Role or use of applications within critical business functions.** The extent to which applications play a critical role in material operations or offer competitive advantage will increase the materiality of risk.

- **Application access to sensitive information.** The materiality of risk increases proportionately with the potential that a compromised application might lead to information loss, information corruption or privacy violations.

- **Degree of regulatory obligations and oversight.** The materiality of risk increases proportionately with the likelihood that a compromised application might lead to regulatory violations or material incidents due to operational disruption, information loss or privacy violations resulting in fines, damage to reputation, etc.

- **Revenue or value associated with the applications’ source code.** The materiality of risk increases when IP theft, software piracy or the theft of other products and services can result from access to an application’s source code.

- **Size, complexity and distribution of application development.** The likelihood that a malicious party will have access to your compiled application, that vulnerabilities may be introduced and/or that artifacts within applications may hold unrecognized potential for vulnerability exploitation is increased in proportion to the complexity, scale and distribution of your (or your suppliers’) development activities.
The 15 industries with the highest baseline risk ratings are shown in Figure 1.

**Observations**

- Industries where managed code has become an integral part of larger product and service offerings are at a higher risk than independent software vendors (ISVs). The seven industries with the highest baseline risk\(^5\) all sustain massive application development efforts and those applications are components within broader business offerings. The central role that these industries play in our daily lives, the capital they represent and the high degree of oversight and regulation make IP theft and application vulnerability exploitation particularly damaging.
- Of the top fifteen, four industries have particularly limited risk mitigation practices\(^6\). While the research does not give any explanation as to why particular industries behave as they do, it may be worth noting that these four industries are particularly reliant upon work-for-hire and subcontracting business models where cost control and delivery schedules are likely to be given far more weight than ensuring that there are effective and sustainable risk management programs for the enterprise consumer. This would be particularly true when these considerations were not included in the original service level or acceptance agreements.
- As managed code has become increasingly commonplace and the advantages of these virtual environments become more widely accepted, the risks associated with uncontrolled access to the source code of these applications will become more

---

\(^5\) The top seven are computer hardware, banking, electronics, diversified services, aerospace and defense, telecommunications and transportation.

\(^6\) Automotive, Consulting, Media and Aerospace and defense
widespread across industries. As such, CIOs, CCOs, CSOs and all of the other stakeholders in business performance, security and compliance will need to consider the likelihood and potential materiality of these risks in light of their corporate appetite for risk.

Risk mitigation and industry “net risk”

Another useful way of looking at the survey results is to contrast the degree of risk mitigation as a percentage of each industry’s baseline risk score. In other words, regardless of the total baseline score, which industries do the best job (are most efficient) at reducing their risk? Figure 2 identifies which industries on average have the most effective combination of preventative measures and controls to manage the risks associated with uncontrolled access to application source code.

**Observations**

- Technology companies share similar development practices even though their baseline risk scores are quite different. Computer hardware (#1), electronics (#3), computer software (#8) and Internet (#19) have each adopted very similar processes and controls within their development processes. This is not particularly surprising in that the development teams within each of these industries probably consider themselves to be application developers first and are not conditioned to measure the different levels of risk they each face due to the context of how their applications are being deployed. It may be the case that one or more of these industries is over (or under) investing in managing these risks.
- A number of industries that did particularly well at mitigating risk were not in the top 15 (education, energy, Internet and manufacturing). Again, the survey data does
not directly address why organizations have adopted their current controls, but it may be worth noting that the top four industries in mitigating these risks above are each subject to numerous and diverse regulations and have an ingrained culture of governance and transparency. Given this tendency to manage risk wherever it is found, it may be the case that their level of risk mitigation is more closely aligned with their true appetite for risk indicating that those industries with equal or greater baseline risk score may be under-investing.

- The bottom six industry scores are somewhat misleading in that the average percentage of risk mitigation does not reflect the high degree of variance across each responding company. In other words, individual companies were more likely to have either very good risk mitigation scores or very poor risk mitigation scores and the relatively low average is more a reflection of inconsistent practices within these industries than mediocre practices applied consistently. This is, in our view, a sign that best practices are being adopted with laggards still doing little or nothing while more forward looking organizations have already begun adopting controls that are likely to become common practice over the next few years.

Risk mitigation maturity
The increasing ubiquity of managed code coupled with heightened security awareness and a growing adoption of a risk-based approach to compliance and IT governance has resulted in a sea change regarding the awareness of risks that stem from reverse engineering and the role of obfuscation as a bona fide compensating preventative control to mitigate these risks. However, changes in policy, control and development practices naturally lag as different industries, organizations and individuals respond to these changes in their own timeframes. Figure 3 illustrates the relative risk mitigation across all industries.

Figure 3: Risk mitigation maturity

Unmanaged risk: 27% of the respondents have no reported controls in place to mitigate reverse engineering risks. These organizations have no policy nor were the respondents aware of any technologies or processes that were in place should individual development groups elect to manage these risks at a local level. Given the trends we are seeing, we expect this group to be forced to establish some sort of policy – even if it is a declaration that reverse engineering risks are immaterial and should not be treated specially.

Locally managed risk: 23% of respondents were reducing their baseline score by no more than 25% and another 12% were reducing their score by as much as 50%. These two groups collectively (35%) did have obfuscation technology available to their developers and were, to some degree, incorporating obfuscation as well as other
application development practices that tracked IP within code and worked to minimize application vulnerabilities. However, these organizations relied upon individual development groups or the direct intervention of auditors to drive the use of these tools and practices. These organizations did not have corporate policies to ensure that, where the business called for compensating controls to prevent reverse engineering, that these controls were actually being applied effectively or consistently. Over time, one would expect corporate recommendations to emerge while the requirement to comply with these recommendations will vary from company to company.

**Managed risk**: 14% of respondents were reducing their baseline score by as much as 75% and the final 24% were reducing their score by as much as 95%. This final group (36% collectively) has mature risk mitigation processes and tools. This does NOT mean that all managed code is being obfuscated. What these numbers do imply is that roughly one third of the respondents have a corporate policy on managing reverse engineering risk that is aligned with their specific baseline risk and their appetite for risk, a set of tools and practices to manage these risks when appropriate and the training and auditing programs to ensure that these risks are in fact being managed as prescribed.

**Obfuscation as a preventative control**

At its core, obfuscation is defined as a collection of transformations that are applied to compiled applications that make reverse engineering materially more difficult for people and machines but do not alter the behavior of the obfuscated application. However, the current climate of heightened emphasis on application security, compliance and development best practices has rendered this definition inadequate. A complete definition has three dimensions; technological, as development process and as an IT control.

**Figure 4: The three dimensions of enterprise obfuscation**

**Technology**

Obfuscation transformations fall into a number of categories including:

- **Renaming**: altering the names of methods, variables, etc. to make source code more difficult to understand. Strong renaming algorithms use overloading to reuse names, forcing every line to be analyzed.

- **Control flow obfuscation**: logic and flow are re-expressed making translation into valid C# (or any other language) impossible. Sophisticated approaches provide different levels to strike the right balance between obfuscation and performance.

- **String encryption**: strings such as login prompts, SQL queries, etc. are encrypted and decryption function calls are injected into the instruction stack before the string is needed.

- **Other**: there are numerous other techniques including metadata stripping, application watermarking, etc. that raise the bar for reverse engineering above what is required to reverse engineer native code (such as C or Cobol).
Process
The process of obfuscation has emerged as the pivotal element in driving the viability and the value of obfuscation. Without a well-defined and integrated obfuscation process, the complexities and risks introduced by obfuscation may ultimately outweigh the perceived benefits that it promises. Obfuscation can complicate debugging, patch generation and management, distributed development practices and the reuse of libraries, components and web services. However, tight integration with development platforms (such as Visual Studio), the inclusion of tools and utilities that can unwind and/or reuse obfuscation transformations and lastly the integration with operations management platforms can mitigate these potentially costly side-effects.

Figure 5: Obfuscation development process

Figure 5 demonstrates a modern approach to enterprise obfuscation. The developer best practice is to have developers indicate where obfuscation transformations may or may not be appropriate. Reflection may confuse renaming transforms, high performance algorithms may call for lite control flow settings, etc. Obfuscation is applied after the build and should also include additional services such as compaction (stripping of unused code to reduce size) and linking (combining multiple DLLs into one to simplify distribution). Further, all transformations should be captured (and previous transformations reused as appropriate) to support the many development scenarios outlined above. Again, all of this should be embedded within the development platform to ensure automation, transparency and quality.

IT control
IT controls are documented obfuscation processes that address the specific risks that are being managed, the processes and training programs that ensure obfuscation is being applied appropriately (not over- or under-used) and consistently (using approved technologies and practices).

Recommendations
Managed code introduces risks that are widely understood and are being managed by a broad cross-section of organization across virtually every industry. The approaches to manage these risks are maturing into well-defined components within the larger, risk based security and governance frameworks and organizations that rely upon applications developed with managed code should consider the potential impact that reverse engineering may have on their operations and respond appropriately.

Every organization must take a position
Each organization must reach a consensus on the answers to the following questions.
• Does your organization develop or rely upon applications developed in either Java or .NET?
• If yes, what is the value and/or potential exposure that can stem from uncontrolled access to the source code of these applications?
• To what extent are your organization’s current policies and controls inline with your peers?
• Should your organization establish or refine policies to better manage these risks in alignment with your overall approach to risk management?
• If the answer to the previous question is in the affirmative, are the processes and technologies in place to ensure that policies are well-understood and applied consistently?

NOTE: If your organization is interested in an assessment and recommendation that would benchmark your specific exposures and practices against the data set described herein, please email solutions@preemptive.com.

Recognize that obfuscation is a technology, a process and an IT control
Obfuscation is the only widely implemented control to prevent reverse engineering of managed code. In order to ensure that obfuscation is applied effectively and consistently, corporations must establish preferred obfuscation suppliers, define the obfuscation process within the overall development environment and ensure that there is a consistent use of obfuscation across all development projects. For a more detailed discussion on enterprise obfuscation, download Enterprise Obfuscation at http://www.preemptive.com/documentation/EnterpriseObfuscation.pdf.

Continue to monitor emerging risks
Hackers, competitors and software pirates are working overtime to find and exploit vulnerabilities in your code, your infrastructure and your business operations. With the arrival of source code indexers, increasingly powerful reverse engineering tools, whitebox testing tools, etc it is likely that new threats will emerge from the uncontrolled distribution of managed code “in the clear.” Not all managed code should be obfuscated, but the policy as to when and how obfuscation as a control is one that should be established and revisited for adequacy on a regular basis just like any other control.

Reasonable responses to well-understood risks coupled with continuous improvement are the hallmarks of effective governance, risk and compliance management programs. In today’s climate, managing the risks that stem from uncontrolled access to managed source code is prudent and appropriate.
Appendix A: Survey Questionnaire

1. The size, industry, organizational structure and role of the development team are all indicators that can help in determining the volume, relevance and complexity of the applications being developed, as well as their potential for increasing risk. Check all that apply to your company's business.

- Develop software for internal use
- Develop software for sale
- Develop software for external use
- Develop web-based applications
- Develop in .NET
- Develop in Java

2. Indicate your company's primary industry.

- Aerospace and defense
- Automotive
- Banking
- Chemicals
- Computer hardware
- Computer software
- Conglomerates
- Consumer durables
- Consumer non-durables
- Diversified services
- Drugs
- Electronics
- Energy
- Financial services
- Food and beverages
- Health services
- Insurance
- Internet
- Leisure
- Manufacturing
- Materials and construction
- Metals and mining
- Real estate
- Retail
- Specialty retail
- Telecommunications
- Transportation
- Utilities
- Other (please specify)

3. How many developers does your organization engage

(full-time, consulting and outsource all apply)?
1-5 Full time equivalents (FTEs)
6-20 FTEs
20-100 FTEs
101-1000 FTEs
1000+ FTEs

4. Check all that apply. Development, QA, support and maintenance distributed across...
   - Workgroups
   - Physical locations
   - National boundaries
   - Multiple companies/entities

5. Development Practices
The likelihood that vulnerabilities may exist within applications that may ultimately be found and exploited through reverse engineering is directly proportionate to the degree of process and control throughout the development lifecycle. Check all that apply.

My organization has controls that are consistently applied to detect current (and to prevent future) vulnerability insertions into the following development phases.
   - Requirements, business processes flow and algorithm specifications
   - The interrelations of modules and (Web) services, logic and data flow
   - Language instructions, implementation of logic and data flow
   - Assembly of executables and UI
   - Patch management, administration and configuration management

5. Development Practices
The likelihood that vulnerabilities may exist within applications that may ultimately be found and exploited through reverse engineering is directly proportionate to the degree of process and control throughout the development lifecycle. Check all that apply.

My organization has controls that are consistently applied to detect current (and to prevent future) vulnerability insertions into the following development phases.
   - Requirements, business processes flow and algorithm specifications
   - The interrelations of modules and (Web) services, logic and data flow
   - Language instructions, implementation of logic and data flow
   - Assembly of executables and UI
   - Patch management, administration and configuration management

6. Development Practices
The likelihood that an application may be reverse engineered and its contents harvested must be considered alongside the materiality or severity of the potential threat should reverse engineering occur.

Please answer the following true/false as they pertain to your company’s applications.

- a. Applications contain Intellectual Property (IP) or trade secrets?
- b. Applications are deployed outside of your firewall?
- c. Applications are licensed or rented to third parties?
- d. Applications play a material role in the delivery of other products or services sold?

6e. Please check all that apply. In my organization, applications play a role in ongoing...
   - Financial operations
   - Manufacturing operations
   - Customer facing services or support
   - HR operations
   - Other (please specify)

7. Information Management: Information and information systems represent two sides of the same privacy, compliance and security coin. If your information systems are vulnerable, the information that is managed is equally at risk. Application security is a cornerstone of information security.

Please answer the following true/false as they pertain to your company’s applications.

- a. Applications access information that is considered confidential
- b. Applications access information that is governed by privacy policies

7c. Check all that apply. Applications access information whose integrity is mandated by...
   - Financial regulations
   - Safety regulations
8. IT Controls IT controls to reduce and manage risks that stem from uncontrolled access to source code should be a component of a broader IT control framework. Many of these controls, such as obfuscation, are widely accepted as a best practice.

8a. Does your organization comply/align its IT operations with a formal IT control framework?
- Yes - it is derived from a computing standard, e.g. COBIT, ITIL,...
- Yes, it is proprietary
- No, development teams establish their own policy locally

9. If your organization obfuscates - please answer the following.

<table>
<thead>
<tr>
<th></th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Obfuscation is defined as part of a formal IT process.</td>
<td>![False]</td>
<td>![True]</td>
</tr>
<tr>
<td>b. Obfuscation is documented as part of the development process, e.g., describing developer, QA, build, support and release management responsibilities and requirements.</td>
<td>![True]</td>
<td>![False]</td>
</tr>
<tr>
<td>c. Obfuscation technology is integrated into our SDLC (Software Development Lifecycle) platform, e.g., Visual Studio, etc..</td>
<td>![True]</td>
<td>![False]</td>
</tr>
<tr>
<td>d. We have a corporate standard obfuscation supplier.</td>
<td>![True]</td>
<td>![False]</td>
</tr>
</tbody>
</table>