

### Enterprise Applications: Wide Open to Attack in 2010

*Abstract: Hidden until recently, a new class of exposures in enterprise applications is mitigated by neither network security nor current application security tools and services. This means that most banks and brokerages are sitting ducks for medium-skilled attackers. Security, compliance, privacy, risk management and internal audit functions at most companies are not rising to the challenge – in fact, most are not even aware of the challenge. Current exposure mitigation strategies are at best window dressing and at worst – grossly negligent. The true scope of these exposures is enormous and a threat not merely to corporate operations, but because the exposures are ubiquitous across all industries, they constitute a clear and present danger to national critical infrastructure.*

#### I. Introduction

Business managers are surprised ... and their customers, when they understand the risks – are even more so. As you read this whitepaper, hackers are outgunning the IT professionals at almost all of the largest and most important financial services firms in America. Application security expert Primeon, Inc. has demonstrated that most of today's enterprise web applications (as well as the back-end applications and data to which they're linked) are highly susceptible to disruption or destruction by malicious hackers. Hold your skepticism - this assertion, though bold, is demonstrably true.

While senior managers have grown more security conscious in recent years, most are completely unaware of the true extent of their risk. One result is that millions of social security numbers, credit card accounts, patient records, as well as corporate intellectual property and sensitive government information - are all currently accessible to unauthorized individuals. The larger risk, however, is that the integrity of many of the most important financial applications and data repositories – the financial foundation of critical US economic infrastructure – is much more at risk today than most leaders are aware.

**... millions of social security numbers, credit card accounts, patient records, as well as corporate intellectual property and sensitive government information - are all currently accessible to unauthorized individuals.**

Several types of corporate risk are a consequence of application exposures. These include operational risk, privacy risk, compliance/regulatory risk, as well as threats to the brand. Within the enterprise, senior managers whose attention is trained on risk management and risk reduction are the individuals most affected by application exposures. Today, however, many are promoting a strategy that embraces "good enough" risk reduction that

# Enterprise Applications Wide Open to Attack in 2010

unfortunately is not nearly good enough. When serious breaches occur, and will if only surface risk mitigation or transference measures are used, managers (and other stakeholders) learn the hard way that they have not practiced the appropriate amount of care or due diligence.

Current approaches to identifying application exposures – commonly known as ethical hacking services and application scanning tools – solve only a small part of this problem. Typically **they find fewer than 10% of the exposures** (primarily in the front-end only) that allow unauthorized system or data access. No wonder - the techniques employed by companies and government agencies to thwart attacks are no more powerful than the resources available to the hackers ... and the hackers have much more time on their hands to plot and execute their attacks.

To substantially solve this challenge, an approach is called for that employs tools and techniques that go far beyond the weapons in the hackers' arsenals. To this end, Primeon contends that identifying and closing application security exposures through a combination of static and dynamic analyses, including comprehensive source code review, let's senior managers sleep better by moving their applications from high risk to very low risk by achieving near-zero exposures in the applications themselves.

## II. Background/History

In terms of regulatory scrutiny, in 2010 it is infinitely more difficult to open a sandwich shop than to open a multimillion-dollar business on the web. The government spends significant funds and energy enforcing safe food handling and other sanitary rules to protect restaurant patrons from getting poisoned. Yet known criminals are free to put any business online and there is no power in the land that bats an eye. Clearly, almost ten years after the first browsers came into being, government regulatory oversight has yet to catch up to the Internet age.

The current condition owes itself, in large part, to the birth and rapid spread of the Internet - to organizations' rush to move their business functionality to this exciting new medium. In a mere handful of years what began as a document exchange network running static brochure-ware has evolved into a complex and massively integrated web of highly transactional, mission critical applications supporting financial services, healthcare, supply chains, government operations, etc. The benefits in terms of communications, access, and organizational/logistical cost savings have been profound. However, the dark side to all this is the fact that no one has been steering this ship.

**... the techniques employed by companies and government agencies to thwart attacks are no more powerful than the resources available to the hackers ... and the hackers have much more time on their hands to plot and execute their attacks.**

**That's right, the applications that drive the economy and the private data belonging to the majority of businesses and consumer customers are accessible by malicious hackers inside and outside the enterprise.**

# Enterprise Applications Wide Open to Attack in 2010

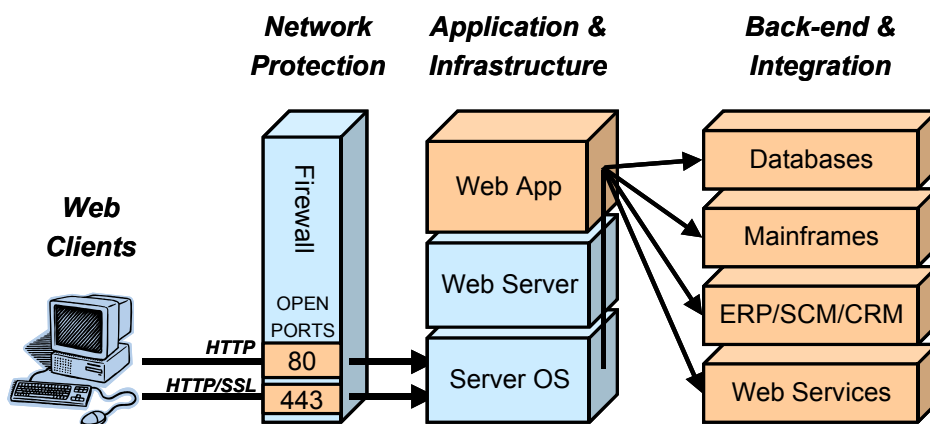
That's right, the applications that drive the economy and the private data belonging to the majority of businesses and consumer customers are accessible by malicious hackers inside and outside the enterprise. Web application attacks easily bypass traditional network security products like firewalls and intrusion detection systems, as well as hardened operating systems and servers (see Figure 1).

If this sounds incredible it's because very few application developers or security professionals are yet aware of how easily accessible the applications they create and protect are to attacks aimed directly at exposures built into web applications. Who put the exposures there? The short answer is: managers, developers, architects, testers, and administrators each own a part of the problem. But it should be noted that these people are paid to deliver functionality and meet time commitments, not build secure applications.

The basic technology problem is fairly easy to understand: web servers have a tendency to trust every request that's sent to them. Whatever the request, the web server, true to its name, delivers to the best of its ability. The server's response can range anywhere from showing a particular page of static text to showing account balances and other trusted information to downloading the contents of entire databases.

**The web server's response can range anywhere from showing a particular page of static text to showing account balances and other trusted information to downloading the contents of entire databases.**

The first products and services available to seal off these exposures (e.g., application-layer ethical hacking, application scanning tools, application firewalls, etc.) are severely underpowered and are unable to offer significant levels of protection against the arsenal



**Figure 1 - Open Ports and Attack Points**

veteran hackers have at their disposal. One of the reasons the story has yet to hit the press full force to date is that these exposures, when breached, are often difficult or impossible to detect. A second reason is that neither individuals nor organizations want to admit they have these weaknesses – so they remain substantially under-reported.

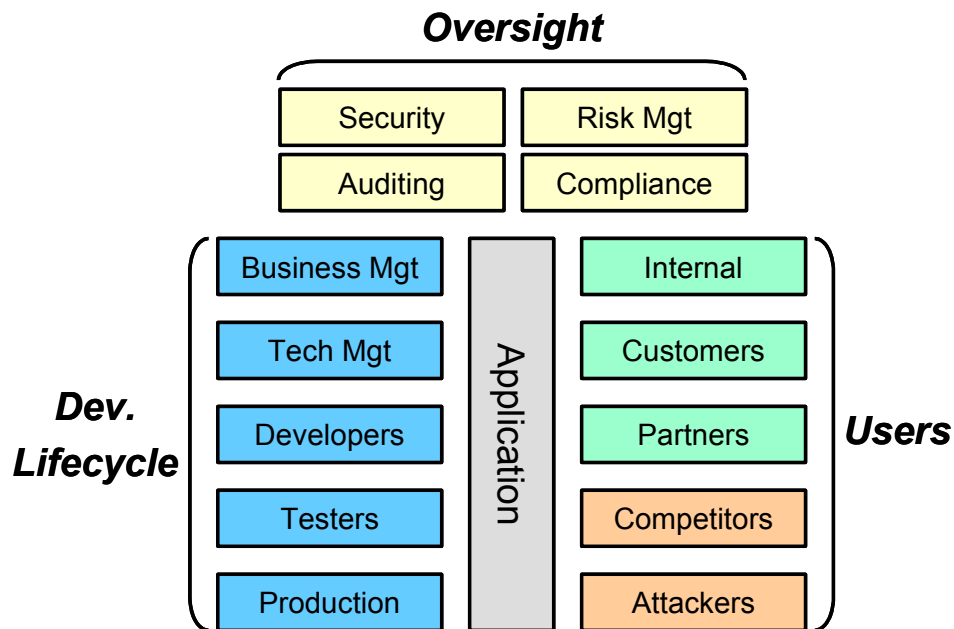
# Enterprise Applications Wide Open to Attack in 2010

## III. Application Exposure Basics

Most high profile technology problems originate in people and process issues, not with the raw technology itself. Prior to wide scale deployment of web architectures, enterprise applications operated in a world that was, in many ways, an island. Attack by malicious insiders with system access has always been an issue, and dial-in and VPN access needed to be secured, but the total number of number of people who could break into a system was very low.

With web applications, much more is required in terms of secure design, coding and testing practices to make applications that are comparably protected. Without the extra people and process controls, all that's required to begin attacking an application is a web connection

**... neither individuals nor organizations want to admit they have these weaknesses – so they remain substantially under-reported**



**Figure 2 - Application Stakeholders**

(meaning a dial-up or broadband connection) and a PC with a browser. Hence, the potential pool of hackers and attackers of any web application has grown to the hundreds of millions. And though many web applications may be useful and powerful, they are fatally flawed without significant extra protection.

There are basically two key drivers that have brought about this situation:

1. Application development teams don't have time (or training, tools or incentives) to deal with security issues;
2. Security staff focused almost exclusively on network security issues.

# Enterprise Applications Wide Open to Attack in 2010

To the first point, the application development lifecycle guides the evolution of an application from its moment of conception in the mind of a business manager, to its design and architecture determined by an application architect, to coding by developers, testing by quality assurance and user acceptability testers, and finally to deployment and maintenance in production by administrators. All of this occurs outside the realm of security because, in the minds of the application development teams, security is handled by another part of the enterprise – the IT security staff (see Figure 2). Here’s how a Director of Application Development at Fortune 100 bank puts it:

“Our engineers are goaled on business requirements and application performance. They do not have the time, training, or motivation to avoid potential security deficiencies in the code they write. This is just as true for the code they are writing today as it is for the millions of lines of code that we have in production.”

The problem is that the security staff is not taking care of application risk. Security personnel have their hands full trying to develop and promulgate sensible security policy, procedures and technological precautions throughout the enterprise and its business units. This ranges from high-end strategic issues, to the daily nitty gritty of dealing with the known vulnerabilities and patch management in operating systems, web servers, email servers, browsers, etc.

Where application exposures are concerned - without full awareness of the potential business ramifications - often no one takes responsibility. The developers consider application-layer exposures a security issue – clearly in the security staff’s domain. The security staff understands the exposures have their origin in the application development lifecycle and need to be dealt with as part of the application development process – by the developers. And both groups point to the other for the responsibility (and budget) to act on the issue.

In many cases, these organizational stovepipes have hamstrung enterprises that are otherwise attempting to deal proactively with application risk. Even companies that have had a public incident (i.e., the press published an embarrassing story about (a) how their systems leaked private customer information; or (b) how the organization saw hundreds of thousands of dollars disappear from its accounts via online fraud) remain flat-footed when it comes to actually securing their applications. Primeon has demonstrated on multiple occasions that the most identifiable “marquee” applications from some of the largest companies in the world - even those that have begun to address application risk – remain fully open to attack.

## IV. Findings from the Field

Primeon has assessed thousands of applications in the past thirteen years, and has performed risk assessments of approximately fifty large and very large applications in the past twelve months. Most of these applications are owned by financial services companies, but a few came from other sectors like telecom, healthcare, retail, etc. A summary of findings, beginning with a composite profile of a

**Primeon has demonstrated on multiple occasions that the most identifiable “marquee” applications from some of the largest companies in the world - even those that have begun to address application risk – remain fully open to attack.**

# Enterprise Applications Wide Open to Attack in 2010

“typical” application, follows.

Typical Application - The typical enterprise application assessed by Primeon is 2 years old, has a major revision once a year and many smaller revisions per year. Some of the applications go through yearly or twice yearly ethical hacking/penetration testing reviews. Some of the applications also get scanned by internal or external personnel using application-scanning tools. Each has some or all of the following characteristics:

- 300K lines of source content (population size range: 2K to 6M)
- Web server (Microsoft IIS, Apache or Netscape)
- Application server (BEA, IBM, other)
- Middleware code (C, C++, Java, other)
- Numerous scripting languages (Javascript, perl, VBscript, etc.)
- One or more databases (Oracle, SQLServer, DB2, Sybase)
- Integration with other data sources, legacy applications
- Messaging protocols

Exposure Findings – Primeon typically identifies 50 – 700 discrete exposures per application. Exposures can be classified into the following categories: (1) parameter manipulation and input validation; (2) configuration management; (3) architecture flaws; (4) business logic and compliance issues; (5) development process issues; and, (6) security management issues.

1. Parameter Manipulation and Input Validation – This means taking advantage of applications’ trust of incoming requests/queries. Includes multiple variations such as: hidden field manipulation, stealth commandeering, forceful browsing, cross-site scripting (CSS), SQL injection, Trojan Horses, failure to verify parameters, buffer overflows and cookie poisoning.
2. Configuration Management – Essentially, all the issues that make it to the press – these are things an enterprise can easily do to limit risk. Easy, but boring, and most organizations leave themselves open on many of these. Includes configuration processes, patch management, known vulnerabilities, storing critical information in configuration files, product and data center management, installation and upgrade processes.
3. Architecture Flaws – The poor choices made by architects and designers are responsible for many application risks. Includes fundamental design flaws in the areas of information protection, messaging, session token management, session timeout management, password management, backups, single sign-on mechanisms, and integration with other applications and databases.
4. Business Logic and Compliance – This means that an application can be functioning within normal parameters can be subverted by an attacker to perform functions that hurt the organization. Includes unintended presentation of sensitive/critical information to unauthorized users, violations of privacy rules, business data damage and theft, malicious code implantation, backdoors, unrecoverable system/data destruction.

## Enterprise Applications Wide Open to Attack in 2010

5. Development Processes – Essentially, all of the missed opportunities to reduce exposure risk are development process exposures. In addition to architects already mentioned, this includes developers, QA/UAT/security testers, and administrators. Includes security policies and procedures, policy enforcement, security testing processes, security and compliance assessment.
6. Security Management – Flaws here are often indicative of a lack of knowledge/awareness on the part of the security unit of the threats posed by the applications themselves. Includes incident response, business continuity and disaster plans, security knowledge base, quantitative analysis, and risk profiling.

Some Indicators of High Exposure/Risk – In general, applications most likely to have higher number of high severity tend to have some or all of the following characteristics:

- Older, usually four years old or more
- Highly transactional
- Multiple levels of privilege
- Custom access control/authentication mechanisms
- Multiple integrations into back end business systems
- Single sign-on access controls
- Developed and/or maintained offshore

Real-World Examples – The following cases are drawn from assessments of critical financial applications performed in the past six months.

### **CASE 1 – An Architecture Problem with Single Sign-On (SSO).**

An online service and support application gives business customers the ability to manage their own account and modify the services they receive; this application is integrated via SSO with approximately 100 other enterprise applications.

SSO is widely used among many financial services firms because it provides individual users the convenience of remembering and using just one login and password for a group of applications (rather than requiring a different login/password for every application). The trend today is toward more SSO access controls and more aggregation among applications. It is now common to have 10s or 100s of applications grouped together by SSO.

SSO greatly increases the threat posed by each exposure in any of the aggregated applications. And as the majority of financial services applications have one

**Mathematically speaking, the operational risk attributed to each exposure in every Single Sign-on Application (SSO) is a summation of the risks across the total number of connected applications. So a Cross-Site Scripting exposure in one application in a 10-app group carries about 10 times the risk, and about 100-times the risk in a 100-application SSO implementation.**

## Enterprise Applications Wide Open to Attack in 2010

extremely popular type of vulnerability named Cross-Site Scripting (CSS), the risk attributed to CSS exposures is substantially increased. Mathematically speaking, the operational risk attributed to each exposure in every SSO application is a summation of the exposure risk across the number of connected applications. So a CSS exposure in one application in a group of 10 carries about 10 times the risk, and about 100 times the risk in a 100-application SSO implementation.

Impact Points – CSS exposures allow attackers to create several types of damage to the target application(s), including Trojan Horses, where the attacker takes control of the application and critical/sensitive info exposure. In the case of this application, session tokens contained the passports to bypass access control on all 100 of the SSO sister applications, even ones with otherwise robust protection.

Action Review – Having closed exposures in the first few applications, the organization is preparing to assess the remaining SSO applications in this group and is reconsidering its approach to access to control for these and other enterprise applications.

**Single sign-on greatly increases the threat posed by each exposure in any of the aggregated applications.**

### **CASE 2 – A Logic Problem with Application Backdoors.**

Backdoors are put in the applications by the developers, testers, product maintenance team for the convenience for collecting information and testing, debugging, and maintaining the application.

Backdoors are common in the most financial services applications, though they are difficult to discover because they are hidden and are not accessible via references or links. The Quality Assurance (QA) and User Acceptance Testing (UAT) testers, ethical hackers, and scanning tools do not find them.

Impact Points – In this case the assessed application was an insurance system with a backdoor page used to help developers test the application's database. Using the functions on the page, attackers could choose the database tables through a pull down menu and retrieve any and all customer records.

Action Review – Following the presentation of assessment details the company has eliminated this back door and is seeking to eliminate this exposure type in other applications.

### **CASE 3 – A Parameter Manipulation Problem with Severe Cross-Site Scripting allowing creation of a Trojan Horse.**

Cross Site Scripting (CSS) is one of the most commonly found exposure types today. In fact, CSS exposures, which can allow user sessions to be hijacked by attackers, are found in the vast majority of Web applications currently in operation. More insidious than session theft is the theft of users' account names and passwords.

# Enterprise Applications Wide Open to Attack in 2010

Impact Points – In this case, the Global trade clearance and settlement application of a large bank, with customers including the largest equity and bond trading firms, allowed attackers to acquire the logins and passwords of those large customers. This was done through a flaw in the access control approach on the login page.

The reason this CSS exposure was so potentially damaging in comparison with typical CSS exposures is as follows. In order for most CSS attacks to work, the victim must be a registered user of the target system and must be currently logged in to an active session. The attacker can then take advantage of the permissions that session grants to him. Because both of these conditions have to be true, it reduces the chances for successful attacks (or in other words, makes the attacker's job more difficult).

In this case, because the CSS occurs on the user login page, it allows the creation a Trojan Horse by an attacker to catch user login information. What's at stake is not a particular session, but rather users' logins and passwords - which can be used anytime. Since the victim does not have to be logged in to an active session, the attacker's job is much easier. And since most users use the same or similar logins and passwords on all of the Web applications to which they're registered, it is highly likely that the attacker will be able to find his way in to other applications as well.

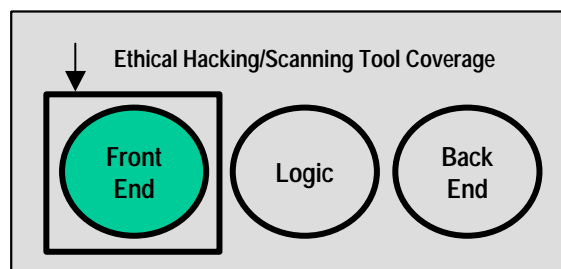
Potential impact for the bank is the loss of key customer information, including the exposure of information on its customers' customers. Unauthorized/unintended execution of trades is another risk.

Action Review – Following the presentation of assessment findings the company is working to eliminate this problem and is seeking to eliminate this exposure type in its other applications.

## V. Current Approaches in Detail

Once an organization decides it wants to act on the application exposure threat, it likely turns to one or more of the three best known approaches currently marketed as solutions: ethical hacking services, application scanning tools, and application firewalls. This section briefly describes the three approaches and the reasons they do not provide full solutions to the challenge.

At the simplest level, the explanation for how these approaches work is the same reason they do not work particularly well. These approaches begin and end at the front-end; that is, all three exist solely at the HTTP level and their view of an application is limited exclusively to HTML. While they may do a reasonable job tracking weaknesses at this level, they can go no further, and much of the complex system of application logic, data stores, legacy back-ends, messaging protocols, etc. are completely out of their reach (while well within the reach of a competent hacker).



# Enterprise Applications Wide Open to Attack in 2010

Ethical Hacking (Penetration Testing) - The looming application exposure threat to IT infrastructures demands immediate action. One of the most common first steps is often ethical hacking, also called penetration testing. Organizations are paying professional hackers and security consultants to probe their applications and report their findings. In general, hiring ethical hackers demonstrates activity to stakeholders, and there is some benefit to learning about certain exposures this way.

However, ethical hackers' capabilities are *severely limited*. In fact, in that they can only detect *symptoms* of application exposure problems because they can only reach a small percentage of an application's code and logic. Ethical hackers have little-to-no visibility into the underlying causes of often-complex application exposures. Since they only have access to dynamically generated HTML pages, typically they cannot refer back to the source code to suggest explicit corrective actions. In fact, due to the many constraints of the ethical hacking approach (e.g., time limitations, knowledge limitations, etc.), ethical hackers are often far less effective at finding exposures than are real hackers who have access to the same or better tools and much more time on their hands. Ethical hackers usually do not know nearly enough about an application to challenge the knowledge that real hackers can accumulate over a far longer period.

**So the money spent on ethical hacking services ultimately buys very little. In fact, if it creates a false sense of security by convincing senior management that their applications are now safe, it does far more harm than good.**

Because ethical hackers typically only see a small percentage of the function points, they:

- Identify fewer than 10% of the application-layer exposures in an application
- Miss the most serious vulnerabilities
- Leave applications' annual loss expectancies (ALE) nearly as high as applications that haven't been ethically hacked
- Generate findings reports that are not actionable – symptoms are identified but rarely the causes

So the money spent on ethical hacking services ultimately buys very little. In fact, if it creates a false sense of security by convincing senior management that their applications are now safe, it does far more harm than good.

Application Scanning Tools - Application scanning tools are the current “tools-based” approach marketed as the solution to the application security challenge. Scanning tools can be viewed as automated versions of ethical hacking behavior – ethical (and non-ethical) hackers often employ scanning tools to help with their work. Yet because they only deal with the parts of the application they can reach (as described in the ethical hacking summary above), scanning tools are highly limited in their ability to identify critical exposures (like ethical hacking approaches, they can expect to identify no more than 10% of the significant vulnerabilities in an application). Finally, they confer no special advantage to organizations over tools already at the disposal of malicious hackers.

# Enterprise Applications Wide Open to Attack in 2010

Ten challenges - facing vendors marketing application-scanning tools as full or significant partial solutions to application security are:

**1. Partial Code Coverage** - Scanning tools only reach some percentage of an application's code logic. This is due to scanning tools' inability to reach all possible chains of action/permutations that a particular action might have on an application.

**2. Inability to Identify Logic Flaws** - While code may be syntactically correct, it may be semantically (behaviorally) wrong. Scanning tools check for syntax, but have no capability to detect inherent logic/design flaws.

**3. Can Only Search for Known Error Patterns** - Tools are only as good as the information they contain – so they are populated with only known exposure patterns. Experienced hackers seek flaws that can be used as attack vectors and are often seek new attacks that have not been seen in the wild before.

**4. Architecture Analysis** - Poor architecture/design is often the cause of exposures, i.e., a program may function well according to spec but contain significant vulnerabilities. Tools can neither review nor critique design specifications. Admin interfaces (e.g., access to databases connected to the application) can be revealed to non-authenticated users. Test pages and administrative pages are sometimes contained in the source code, and scanning tools are oblivious to this type of risk.

**5. Security Process Analysis** - Many exposures have their roots in development and/or security policies and procedures. By definition, tools do not review development and/or security policies and procedures. DeepSource includes review of the software development lifecycle (SDLC) within the context of the organization's security policies and procedures. DeepSource is also a powerful mechanism for the enforcement of secure development practices and conformance to internal security policies.

**6. Actionable Remediation** - Scanning tools can only identify where a problem *may* lie – therefore, they cannot provide specific, targeted remediation guidance to fix exposures. As they only have access to binaries (executables), they cannot refer back to the source code to suggest corrective actions. This is particularly true for applications that generate dynamic pages.

**7. Design & Coding Practice Analysis** - Particular design choices (e.g., scripting languages, APIs, messaging protocols) have a significant impact on application security. Scanning tools have no capability to consider these issues.

**[Scanning tools] confer no special advantage to organizations over tools already at the disposal of malicious hackers.**

# Enterprise Applications Wide Open to Attack in 2010

**8. Customized Clients** - Scanning tools are limited to standard browser-based clients. Custom client interfaces to Web applications (e.g., Java applets, Active X and other extensions) are ubiquitous in several sectors (particularly financial services), and scanning tools cannot understand the customized messaging protocols that reside on top of plain HTTP requests.

**9. Work Flow Coverage** - Scanning tools only reach some of an application's business logic – many pages are not reachable unless certain sets of conditions/inputs are met. Especially for applications with dynamic and complex page structures, tools stop well short of complete coverage.

**10. Web Services and XML** - DTDs and SOAP protocols that underpin Web Services applications are often custom. Emerging Web Services arena will result in substantial application security gaps. While many Scanning tool vendors claim that they may add this capability some time in the future, today's tools simply have no capability to understand the many different variations in DTDs and SOAP protocols. Also, Web Services security standards are still evolving and in a state of flux in 2010. Already fielded Web Services applications are in substantial danger.

**[While] there's no denying the allure of tools-based solutions to enterprise IT buyers ... it's clear from the marketing language used by most of the scanning toolmakers that their claims of efficacy are grossly overstated.**

In summary, there's no denying the allure of tools-based solutions to enterprise IT buyers. However, it's clear from the marketing language used by most of the scanning toolmakers that their claims of efficacy are grossly overstated. As long as IT buyers are informed of the inherent strengths and limitations of all competing approaches, they are empowered to make the best use of tools and services-based approaches to lower their application risk.

## VI. A Full Solution is Ready

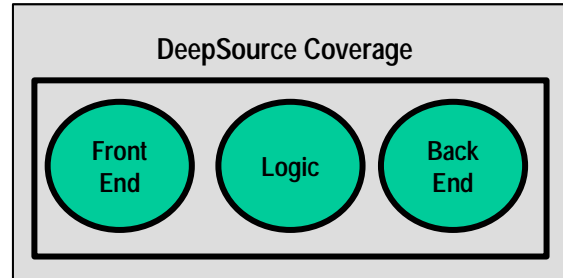
A lead application architect in one of the largest brokerages said recently:

“It would take years of training and significant cultural change in our company to achieve a high level of application security expertise and integrate it in our development cycle.”

The people closest to the problem – the architects and lead developers - it seems, don't expect to be able to solve this problem from their side any time soon. In complex applications, DeepSource assessments are - by an order of magnitude - consistently superior in revealing critical vulnerabilities when compared to either ethical hacking or application scanning approaches. Primeon's key advantages are the methodology and technology it employs.

## Enterprise Applications Wide Open to Attack in 2010

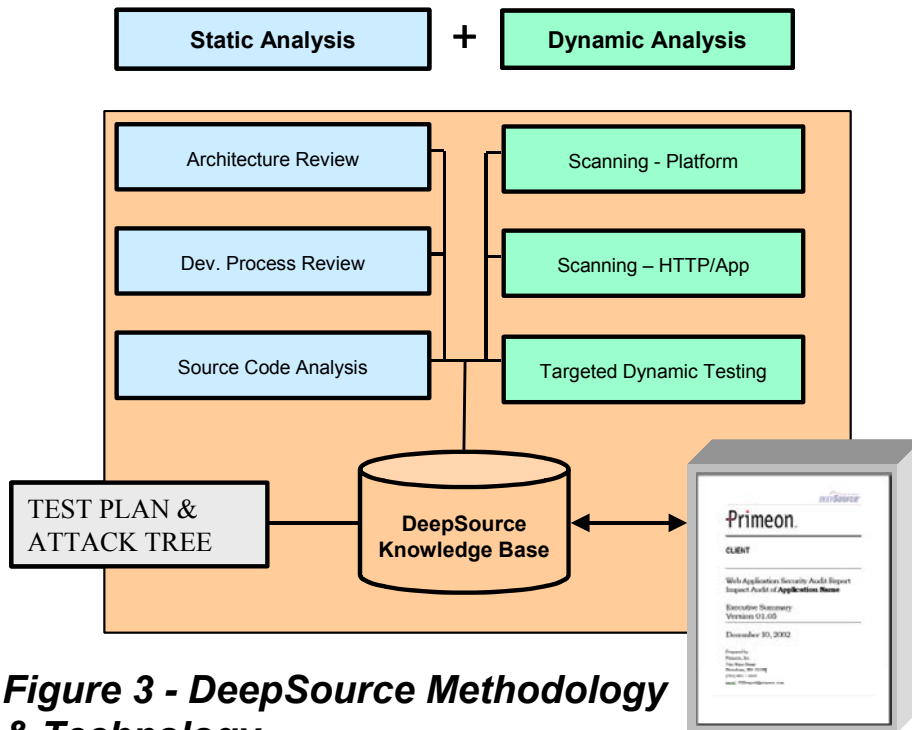
Because our experts and tools have a high-resolution view of the entire application (front end, business logic, data) in each assessment, we are able to report with confidence that our findings are virtually exhaustive. In fact, in engagement after engagement, clients tell Primeon that DeepSource finds numerous important exposures in critical applications that have already been examined by their best ethical hackers and/or latest application scanning products. This makes perfect sense to us, as those approaches have access to a very small part of the overall application and they are positioned to solve a different problem. That is, ethical hacking and application scanning are designed to demonstrate activity; DeepSource is designed to dramatically and demonstrably reduce application risk to near zero.



DeepSource Methodology - Primeon's DeepSource experts interview key development lifecycle staff members to capture a thorough understanding of their applications' architecture, development and security processes and policies that affect them. Other tasks include comprehensive static and dynamic analyses of the applications themselves, powered by application risk intelligence residing in the encyclopedic DeepSource Knowledge Base (see Figure 3). This approach allows Primeon to achieve exhaustive, near perfect exposure. Remediation guidance includes suggested fixes to code and security architecture, as well as an analysis of known vulnerabilities and process issues. Lessons learned can be shared with other application owners for use across the enterprise. Security Assessments can be performed on applications in production or as part of the testing regime for release management.

# Enterprise Applications Wide Open to Attack in 2010

DeepSource Application Risk Knowledge Base - The DeepSource Application Security Knowledge Base is the key to Primeon's success and what truly differentiates Primeon's application assessments from the competition. Each application assessment benefits from the accumulated intelligence of thousands of assessments that have preceded it, and the findings of



**Figure 3 - DeepSource Methodology & Technology**

each assessment (in client sanitized form) make the Knowledge Base more capable and more current. Subsequent assessments on the same application are made even more efficient once the DeepSource Knowledge Base captures the initial parameters of the application in the first assessment. Thanks in part to our Knowledge Base-driven solution, analysts and clients report that they have seen nothing that approaches Primeon's capabilities in the application risk audit space.

Benefits to Clients - The difference produces several critical benefits for DeepSource clients including a radically lower cost per vulnerability found. Probably the biggest advantage, however, is the fact that DeepSource is a **full solution to application risk reduction**. Hence, clients enjoy (and can demonstrate) annual loss expectancies (ALE) on DeepSourced applications approaching zero. The comprehensive source code review element also means that organizations that employ DeepSource meet the letter and the spirit of the new FFIEC guidance on risk reduction in outsourced (and non-outsourced) financial applications.

About Primeon - Primeon is a leader in enterprise-wide application security, planning, assessment, pen-testing and application QA. Primeon offers a complete solution for identifying application exposures, inefficiencies, "malicious code" and subsequent remediation measures in a simple to read actionable output report. Leveraging 13 years of application analysis experience that exceeds over 1 billion lines of source code and nearly all computing architectures & languages, Primeon is

## Enterprise Applications Wide Open to Attack in 2010

able to offer the most thorough application security solution on the market. In addition to scanning tool & pen-testing components, Primeon's DeepSource(tm) includes analysis of source code (100% code coverage), comprehensive knowledge base plus accelerated interactive testing that provides unmatched levels of exposure identification and risk mitigation with our fully independent application audits. Founded in 1995 as a premier application assessment company, Primeon's satisfied clients include many of the Fortune 500 and over 50 of the largest financial services enterprises on Wall Street who trust Primeon with their entire application portfolios.

Mike Pettiglio  
Executive Vice President  
mpettiglio@primeon.com  
Primeon, Inc.  
Tel. # 917-699-8165