Gartner

Research

Publication Date: 21 September 2005 **ID Number: G00130869**

Organizations Should Implement Web Application Security Scanning

Amrit T. Williams, Neil MacDonald

Web applications are prone to vulnerabilities and exposures because they are modified often and seldom undergo thorough development and testing best practices. Organizations should regularly scan Web applications for vulnerabilities and push for early testing.

WHAT YOU NEED TO KNOW

- Security and audit professionals — Web application scanning must be included as a required part of your vulnerability management program. Because new techniques for attacking Web applications will appear over time and application updates are inevitable, regular scanning will be necessary. Your organization may contain hundreds of Web applications. Start with those that are accessible externally and considered critical to the business. Finally, press internally to gain adoption of these tools by the development organization (or, if outsourcing development, require proof of the use of these tools contractually).
- Application development managers — Application security is one part of application quality. The longer-term solution is to adopt the use of these tools earlier in the development life cycle, before their deployment into production. Start during the QA process before moving the Web application back into the hands of developers. For developers of Web applications, consider mandatory training on common Web application vulnerabilities and how to avoid them.

STRATEGIC PLANNING ASSUMPTION(S)

Through 2010, enterprises that scan their Web applications after any modification will experience a 70 percent reduction in security incidents in those applications (0.8 probability).

By 2008, at least 40 percent of enterprises will have adopted Web scanning tools as a standard part of their application development process (0.7 probability).

ANALYSIS

Vulnerability management is the process of improving an organization's security posture through the continuous and proactive identification of weaknesses or lapses in security. Organizations need to include Web application security scanning as part of their vulnerability management program (see "Improve IT Security With Vulnerability Management"). We estimate that 90 percent of externally accessible applications today are front-ended by a Web server, and two-thirds of those production Web servers have exploitable vulnerabilities that enable attacks against the underlying servers and services. Because Web applications are intended to be externally accessed using a browser, attacks easily bypass perimeter security solutions, which typically pass ports 80 and 443 traffic without restriction.

Web Application Vulnerability Assessment Scanning

Vulnerability assessment, in the context of Web applications, is the process of using technology to assess the state of Web applications to determine if vulnerabilities, exposures and poor coding practices are present. As with traditional network-based vulnerability assessment scanning, the applications are scanned for vulnerabilities that can lead to exploits. Unlike vulnerability assessment scanning, Web application scanning does not look for specific, known published vulnerabilities for which standard patches exist. Rather, the entire Web application must be scanned for unknown vulnerabilities based on general knowledge of what types of vulnerabilities are likely to lead to exploits (for example, cross-site scripting, command injection and so on — see Note 1). Through 2010, organizations that test their Web applications for vulnerabilities after every change will experience a 70 percent reduction in security incidents in those applications (0.8 probability).

Some vendors are focusing assessments specifically on Web application development and coding errors, scanning the applications to determine potential coding or configuration errors that could result in an exploit. Some of the vendors sell products to enable you to perform your own scanning; others sell as a service only, while some vendors provide both. These vendors include Cenzic, SPI Dynamics, Watchfire and WhiteHat Security. These vendors scan the application from a "black box" perspective; in other words, they do not perform source-code analysis. They analyze the application by scanning all of the exposed surface area (for example, input fields) and by using automated routines to try to discover an underlying vulnerability (see Note 2).

Shielding Web Applications

Vulnerabilities identified in Web applications generally require a resolution to be developed, as opposed to simply applying a vendor patch. Because of the time it can take from identification of the issue to deployment of the updated application, it is important to protect the application while it is in the vulnerable state. Utilize network and endpoint security tools (including Web application firewalls) to shield the application before deployment of the remediated program (see "Application Delivery and Web Application Firewalls Are Ready to Converge").

- Recommendation: The security job doesn't end once a vulnerability has been identified. Security groups should deploy security technologies that can provide shielding of vulnerable applications before elimination of the root cause.

Integration and Automation Challenges

While many organizations perform Web application scanning of their production Web applications, the earlier in the cycle the scanning is performed, the more efficient and timely the resolution can be. Making Web application vulnerability testing a requirement during the preproduction quality assurance (QA) and audit phase has been effective for Type A organizations. Integrated Web application vulnerability testing into the software development process can be more effective, but at least 40 percent of enterprises will be able to do so before year-end 2008.

One key issue is that security and audit groups are not equipped to address the underlying vulnerabilities. Because most Web applications are developed in-house, typically internal development and testing resources will be used to fix the vulnerable application. To ensure that resources are not wasted, Web application scanning tools must not produce false-positives, even if there is a potential that legitimate vulnerabilities are not found. Web application scanning tools must provide the proper remediation information and as much automation, accountability and historical data needed for the application change management process. Look to vendors that can integrate with your existing development processes and technologies in addition to IT workflow technologies.

Recommendations

- Use process to overcome technology gaps
- Clearly define roles and responsibilities

Development and QA groups

- Integrate or export assessment data into deployed defect tracking systems to ensure there is change control accountability around changes to the application
- Longer term, integrate Web application scanning products with source control systems used for Web application development to ensure there is a record of the modification and the details of the modification

IT security and support groups

- Integrate Web application scanning output with organizational help desk and workflow tools

Key Issues

How will enterprises manage IT configurations to eliminate vulnerabilities and implement security policies?

Note 1

Common Coding Errors

Some examples of coding or configuration errors that can result in a security incident:

- Buffer overflows — A poorly coded application may attempt to store more data in a buffer than is allocated, which can result in an attack, whereby the malicious data is overflowed into another buffer to perform malicious code instructions.
- Cross-site scripting — This is a form of attack in which data entered into one area is inserted into another trusted area, which can result in an attack using the trusted credentials.
- Denial of service — This is an attack that results in the inability of the device or application to provide services.
- Error handling — Poor error handling can result in some of the attacks mentioned (cross-site scripting and denial of service); furthermore, poor error handling can result in information disclosure, which the attacker can use to determine characteristics of the application to assist in a more-advanced attack.
- Poor or nonexistent session IDs — When session IDs are not properly used, an attacker can compromise a Web session and perform multiple attacks (by assuming the credentials of another), thereby bypassing authentication mechanisms
- Command injection — This is the process of exploiting Web pages through the use of specialized characters or Structured Query Language (SQL) queries in customer-supplied data fields.
- Weak authentication — Taking advantage of weak authentication or unencrypted data for the purposes of accessing, compromising and manipulating data is a significant problem that can easily be avoided with a proper best practices approach to Web application development.
- Unprotected parameter passing — This is the use of uniform resource locator (URL) stuffing and hidden Hypertext Markup Language (HTML) tags as a means to pass parameters to the browser, under the assumption that the browser will not modify them before returning HTML to the server.

Note 2

Black-Box Testing of Applications

Black-box testing of Web applications is sometimes referred to as "dynamic" analysis, because the application is scanned as it would be deployed and made accessible by users. The goal is to

find vulnerabilities that are exposed to the outside world over port 80 (HTTP) and port 443 (Secure Sockets Layer, or SSL). There may be vulnerabilities in the underlying code that are not found using this technique, because the vulnerabilities are exploitable to the outside world using the exposed Web interface.

In contrast, some vendors perform source-code analysis of underlying applications specifically to find all sources of vulnerabilities within an application (sometimes referred to as "static" analysis). These products, services and vendors (for example, Coverity, Fortify Software, LogicLibrary, Ounce Labs and Secure Software) are outside the scope of this Research Note.

We consider the use of these two kinds of tools to be complementary, and organizations looking to improve their overall security profile should make use of both.

REGIONAL HEADQUARTERS

Corporate Headquarters
56 Top Gallant Road
Stamford, CT 06902-7700
U.S.A.
+1 203 964 0096

European Headquarters
Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Asia/Pacific Headquarters
Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600

Japan Headquarters
Gartner Japan Ltd.
Aobadai Hills, 6F
7-7, Aobadai, 4-chome
Meguro-ku, Tokyo 153-0042
JAPAN
+81 3 3481 3670

Latin America Headquarters
Gartner do Brazil
Av. das Nações Unidas, 12551
9° andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509

Gartner