Online security management
White paper
December 2007

Rational® software

The dirty dozen: preventing common application-level hack attacks.

Contents

Introduction

As organizations have grown increasingly dependent on online software, the risk of malicious attacks has also become far more serious. Such attacks can bring a business to a standstill, cost a company millions of dollars in lost transactions and potentially tarnish its brand image.

Although most organizations are able to implement effective security at the network level using firewalls and encryption, many organizations inadvertently place sensitive customer and corporate information at risk by failing to protect the application layer. Consequently, by thinking like a developer and identifying shortcuts that the developer would have created, a hacker can wreak havoc on a vulnerable application and its surrounding infrastructure within a matter of hours, using nothing more than a Web browser.

Fortunately, well-governed organizations can protect their Web applications by injecting vulnerability assessments and ethical hacks into their software development and delivery processes. By using automated tools to perform these checks throughout the online application lifecycle, auditors, developers and quality assurance (QA) professionals can help foil hackers and reduce their company's exposure to potential business losses. This paper describes 12 of the most common hacker attacks and provides basic rules that you can follow to help create more hack-resistant Web applications.

Highlights

IBM Rational security experts worked with clients across several regulated industries to identify common Web application security vulnerabilities.

Common attacks include cookie poisoning, hidden field manipulation and parameter tampering.

Understanding common hacks

After working with leading enterprise clients in a wide range of regulated industries—including financial services, government and pharmaceuticals—the IBM Rational® technical team has identified and studied the 12 most common attacks.

Twelve common hacks

Type of hack	Illicit purpose	How it's done
1. Cookie poisoning	Identity theft/session hijacking	Many Web applications use cookies to save information, such as user ID or a timestamp, on the client's machine. But these cookies are not always cryptographically secure, so a hacker can modify them and fool the application into changing their values—essentially "poisoning" the cookie. The hacker can therefore gain access to other people's accounts and make fraudulent transactions such as purchases and money transfers.
2. Hidden field manipulation	E-shoplifting	Retailers often use hidden fields to save information about a customer's session, eliminating the need to maintain a complex database on the server side. Many also use such fields to store merchandise prices. On unprotected sites, hackers can view source codes, find these hidden fields and alter prices. Companies might not detect these changes and ship the hacker merchandise at an altered price—and perhaps even send a rebate.
3. Parameter tampering	Fraud	Many applications fail to confirm the correctness of common gateway interface (CGI) parameters embedded within a hyperlink, so hackers are then able to easily alter the parameters. This might enable the hacker to secure a credit card with a US$500,000 limit, skip a site login screen or gain access to other customers' orders and information.

Highlights

By exploiting common security vulnerabilities, hackers can attack enterprise Web applications using many approaches.

Type of hack	Illicit purpose	How it's done
4. Buffer overflow	Denial of service	By exploiting a flaw in a Web form, hackers can overload a server with excess information, causing it to crash and shutting down the Web site.
5. Cross-site scripting	Hijacking/identity theft	Hackers can inject malicious code into a Web site that executes as if it originated from the targeted site. This gives attackers full access to the retrieved document and may even send them the page's data.
6. Exploiting backdoor and debug options	Trespassing	Developers often embed debug options into the code to test the site before it goes live. If they forget to close these security holes, hackers can freely access sensitive information.
7. Forceful browsing	Breaking and entering	Hackers can subvert the application flow and access information and components that should be inaccessible, such as log files, administration facilities and application source code.
8. HTTP response splitting	Phishing, identity theft and e-graffiti	Hackers can poison a Web cache both at the site and on intermediate systems, which enables them to change Web pages in the cache and perform a variety of attacks against users. Plus, this tactic gives hackers an enhanced ability to conceal their activities.

Highlights

Some embedded infrastructures and protocols that support XML-based applications can introduce vulnerabilities into a site's infrastructure, protocols and content.

Type of hack	Illicit purpose	How it's done
9. Stealth/ Trojan horse	Malicious damage	Hackers can conceal dangerous commands via a Trojan horse that unleashes malicious or unauthorized code and damages the site.
10. Exploiting a third-party misconfig-uration	Malicious damage	Hackers often visit public sites that post vulner-abilities and patches. By exploiting these known misconfigurations, hackers can potentially create a new database that renders the existing database unusable by the site.
11. Exploiting known vulner-abilities	Taking control of the site	Some Web technologies have inherent weak-nesses that a persistent hacker can exploit. For example, some hackers can commandeer an entire site because they know how to access administrator passwords via Microsoft® Active Server Page (ASP) technology.
12. Exploiting XML and Web services vulner-abilities	Malicious damage	Certain embedded and external infrastructures and protocols that support XML-based appli-cations can introduce vulnerabilities into a site's infrastructure, protocols and content. Moreover, some types of attacks—including entity expansion, XPath injection, structured query language (SQL) injection in XQuery, and various denial-of-service attacks—exploit XML's flexibility and richness to inflict major damage on all of these elements.

Highlights

To protect their Web-based assets, organizations can build security into Web applications and test for vulnerabilities throughout the development and delivery lifecycle.

Building more hack-resistant applications

With so many opportunities for hackers to exploit Web technology, what can organizations do to protect their Web-based assets? First, think defensively. Instead of focusing only on how to attract users to your site, assume that some of those users will try to manipulate your applications. Help build security into your Web applications by testing for vulnerabilities throughout the development and delivery lifecycle. Use automated tools to help ensure that you're testing all your applications and detecting vulnerabilities that can slip through the cracks with manual testing. In addition, keep the following rule in mind: never trust data that comes from a user, and never make assumptions about the limits of a user's technologies.

In other words, all data from outside sources is potentially dangerous. Assume that anything a user could theoretically manipulate will be manipulated. Moreover, just because a user is supposedly employing a specific technology, do not assume that it will constrain his or her actions. For example, even if a browser does not show hidden fields in a page's HTML code, you should assume that some users will still be able to find and manipulate those fields before sending pages back to your server.

Highlights

IBM Rational AppScan is a leading suite of automated Web application security solutions that provide intelligent fix recommendations and advanced remediation capabilities.

How IBM can help

IBM Rational® AppScan® is a leading suite of automated Web application security solutions that scan and test for common Web application vulnerabilities, including the 12 identified in this paper. Unlike other solutions that inundate users with vulnerability data, IBM Rational AppScan provides intelligent fix recommendations as well as advanced remediation capabilities, such as comprehensive task lists that can help users fix vulnerabilities uncovered during the scanning process to improve your company's overall security posture.

For more information

To learn more about how IBM Rational automated lifecycle security tools can help you create security-rich Web applications and prevent common hack attacks, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/rational/offerings/testing/webapplicationsecurity

RAW14008-USEN-00