Information Security Updates

Security Risks for Web Application

Issue 10

Related Article

Dorset HealthCare University NHS Foundation Trust website targeted by hackers

Hackers brought down a website for a health trust serving 700,000 NHS patients in Dorset, UK. The attack, which disabled the site over the weekend, blocked access to the Dorset HealthCare University NHS Foundation Trust website.

Hackers posted a message on the site which read: "Don't mess with me. Site totally hacked" and a cartoon image of a penguin emerging from a shattered computer screen carrying a gun.

(http://www.bournemouthecho.co.uk/news/8867662.Dorset_HealthCare_University_NHS_Foundation_Trust_website_targeted_by_hackers/)

Recent Incident

UK Uncut hacks into Vodafone website

Anti-cuts campaigners from UK Uncut have hacked into the website of phone giant Vodafone and posted blogs claiming the company has avoided millions of pounds in tax.

Activists took over the blogs on the World of Difference website, the company's corporate and social responsibility initiative, demanding the company "pays its tax". Twenty minutes after activists hacked that section of Vodafone's website, it appeared to have been taken down.

(http://www.guardian.co.uk/uk/2011/mar/10/uk-uncut-hacks-vodafone-website)

I. Background

Industry Story

University of Sydney Web Defacement Uncovers Data Breach from 2007

The University of Sydney is working to respond to a complicated situation discovered after a hacker defaced the university's main website and emailed the defacement to all students in January 2011.

While investigating how the front page of the university's website was defaced, detailed records on former and current students were discovered to be publicly available. The records, part of invoices generated for students using the Higher Education Contribution, contain student names, addresses, email addresses, enrolled courses and course costs.

University of Sydney Vice-Chancellor Michael Spence confirmed, in a letter to students, that the university had been made aware of the data breach back in 2007 and the problem had been corrected. However, according to Spence, a software update at some point inadvertently removed the fix and exposed the student information once more. As a result of the breach, New South Wales acting Privacy Commissioner John McAteer has launched an investigation into the University of Sydney incident to determine if the university had violated the NSW Privacy and Personal Information Act of 1998.

See this article:
http://www.adamdodge.com/esi/archive/2011/01

Web Applications in Universities

Web applications are applications that can be accessed over a network such as the Internet or an intranet in a browser-controlled environment. They are usually developed based on the client-server architecture by using a combination of server-side script (e.g. ASP, PHP, etc) and client-side script (e.g. HTML, JavaScript, etc). A web application can be as simple as a message board on a website, or as complex as a word processor or a spreadsheet, such as Google Docs and Microsoft Office Web Apps.

Given the mobility, ease of access and cross-platform nature, web applications are now extensively used within the universities. Typical examples include web-based campus e-mail system, online student information portal, online facility booking system and interactive teaching websites.

Reference:
http://webtrends.about.com/od/webapplications/a/web_application.htm

II. Management

Key Security Risks of Web Applications

The introduction of web applications also raises new concerns on information security. Important or sensitive information can be stored within the web applications, such as student personal data, copyright teaching material, and university confidential information. Since the web applications are usually designed to be accessed by large numbers of users, they require a high level of system availability as well as information protection controls. The following describes a few common vulnerabilities of web applications which might cause universities to be exposed to hackers' attack.

1. **Insufficient Validation Checks**

   Without proper "validation and escaping" mechanism, web applications would accept untrusted data, which could cause injection flaws when deliberate instructions are sent to the database as part of a SQL query. The attacker's hostile data can trick the affected systems into executing unintended commands or accessing unauthorised data.

   In addition, Cross Site Scripting ("XSS") may also occur, which allows attackers to execute scripts in users' web browsers that can hijack user sessions, deface web sites, redirect users to phishing or malware sites, or be forwarded to access unauthorised pages.

2. **Broken Authentication and Session Management**

   Web application function s related to authentication and session management may not be sufficiently implemented, which allow attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities. In a recent incident, the AT&T network was found to have session management vulnerabilities, which resulted in iPad user information exploited by the hacker.

3. **Failure to Restrict Web Page Access**

   Privileged web pages containing confidential information or powerful configuration access should be protected by web applications through checking the user identifies before processing the web page requests. Lack of comprehensive authentication verification or mis-configuration may allow attackers to access sensitive data or privileged web application functions. For example, direct copy and paste the URL of the configuration page of a web application in the web browser may allow a hacker to access the administrative function.

4. **Exposed Network Traffic Information**

   Information exchanged between web application servers and end user web browsers may not be protected using strong authentication and encryption techniques. Weak encryption, weak algorithm, out-dated authentication method or even data transmission in plain text can adversely affect the confidentiality and integrity of sensitive network traffic for web applications.

Reference:
http://www.owasp.org/index.php/
http://www.owasp.org/index.php/Testing_for_SQL_Injection_(OWASP-DV-005)
http://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
http://www.owasp.org/index.php/Testing_for_authentication

Recent Incident

Anonymous speaks: the inside story of the HBGary hack

HBGary Federal CEO Aaron Barr was preparing to name and shame the hackers, "Anonymous", for co-ordinating the group's actions, including the denial-of-service attacks that hit MasterCard, Visa, and other perceived enemies of WikiLeaks late last year.

Anonymous response was swift and humiliating. HBGary's servers were broken into, its e-mails pillaged and published to the world, its data destroyed, and its website defaced. As an added bonus, a second site owned and operated by Greg Hoglund, owner of HBGary, was taken offline and the user registration database published.

(http://arstechnica.com/tech-policy/news/2011/02/anonymous-speaks-the-inside-story-of-the-hbgary-hack.ars/)

Related Article

Lessons Learned Thanks to HBGary and Anonymous

IT administrators need to understand that there are no silver bullets, and that there is never a point where you are "done" securing the network and data. You must implement a layered defence of reasonable security controls, then diligently monitor for threats and suspicious activities 24/7. You should make sure you do your due diligence before doing business with a security consultant or hiring a security firm. Do your best to make sure you are doing business with someone with the skills necessary to get the job done, and the moral compass to not cross the line.

(http://www.pcworld.com/businesscenter/article/220209/lessons_learned_thanks_to_hbgary_and_anonymous.html)

Related Article

California Polytechnic State University IT Security Standard: Web Application Development

Departments that develop, maintain, and support web applications must incorporate procedures to ensure these applications are appropriately managed and documented throughout their life-cycle. These procedures include:
• Formal documentation and approval of a web application throughout its life-cycle from initial proposal through deployment to a production environment
• Formal change management and approval processes that include separation of duties and/or management oversight
• Formal documentation for testing procedures, including:
  • Testing for security vulnerabilities
  • Formal user acceptance
• Use of a version control system

(http://security.calpoly.edu/docs/standards/webapp-development.pdf)

Related Article

OWASP TOP 10 Security Risk for 2010

Many organisations have relied exclusively on an occasional scan or penetration test to gain assurance for their internal and external web applications. This approach is expensive and does not provide much in the way of coverage. Like other types of security, web application security requires a risk management program that provides visibility across the entire portfolio and strategic controls to improve security.

OWASP has released an updated report capturing the top ten risks associated with the use of web applications in an enterprise.

(http://www.owasp.org/index.php/OWASPTop10-2010-PressRelease)

II. Management (Cont'd)

Web Applications Development

When developing web applications, universities' software development lifecycle procedure should be consistently followed. Management should pay close attention to a number of security considerations and determine the required security controls during the design stage of a web application development. Key security considerations include (but not limited to):

- Authentication Requirements;
- Privacy and Integrity Requirements;
- Input Data Validation;
- Exception Handling and Reporting; and
- Audit Trail Logging.

At the later testing stage, management should ascertain that sufficient tests are performed to verify the functionalities of the designed security controls prior to the release of the web application to the users.

Security Testing of Web Application

In addition to integrating of security measures during the development phase, conducting security testing is another critical process that helps to identify vulnerabilities of web applications and protect the information contained therein. To ensure that security testing can accomplish its objective, management should perform the following tasks:

1. **Risk Assessment –** Management should define the scope of the web application testing by identifying high risk web applications, key risk areas of certain web application, and the relevant database with confidential or sensitive information.

2. **Owner Identification and Scheduling –** Management should identify the owner of the web application, and assign adequate time and resources to perform security testing prior to the launch of the web application.

3. **Contingency Planning and Impact Analysis –** Since security testing can involve penetration testing which may have adverse impact on the web applications or other devices and data on the network, appropriate contingency planning as well as impact analysis should be performed prior to the performing of security testing.

Common Security Testing Types

1. **SQL Injection Testing**

   SQL Injection is one of the major security threats of web applications. Specific testing should be conducted to detect and prevent the possibility of SQL Injection. The tester should list all input fields whose values could be used in crafting a SQL query and then test them separately, with the objective of interfering with the query and to generate an error.

Reference:
http://www.owasp.org/index.php/Testing_for_SQL_Injection_(OWASP-DV-005)
http://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
http://www.owasp.org/index.php/Testing_for_authentication

Related Article

University Data Breaches Underscore Need for Employee Security Training

Three universities recently reported security breaches that compromised student and faculty private data. While unrelated, these incidents underscore the importance of educating employees about the security implications of accidentally misplacing data.

Although the data was supposed to be uploaded to a secure server accessible only to university personnel as part of the accreditation process, it ended up on an insecure server, exposing it to the Google spiders that indexes the Web. The IT team is currently working with Google to remove all leaked lists from the search engines indexes, the university said.

(http://www.eweek.com/c/a/Security/University-Data-Breaches-Underscore-Need-for-Employee-Security-Training-506070/)

Related Article

Improving Web Application Security: Threats and Countermeasures

The Development Considerations Checklist describes a list of model considerations during the development of web applications. An example is the input validation checklist, which includes:
Input to Web methods is constrained and validated for type, length, format, and range.
Input data sanitisation is only performed in addition to constraining input data.
XML input data is validated based on an agreed schema.

(http://msdn.microsoft.com/en-us/library/aa302349)

II. Management (Cont'd)

2. Cross Site Scripting ("XSS") Testing

XSS flaws can be difficult to identify and remove from a web application. One way to test for XSS flaws is to verify whether a web application will respond to requests containing simple scripts with an HTTP response that could be executed by a user's web browser.

Nessus, Nikto, and some other available tools can also help to scan web applications for these flaws.

3. Authentication Testing

In computer security, authentication is the process of attempting to verify the digital identity of the sender. Breaking the authentication mechanism of web applications is always one of the most popular means that a hacker will choose.

There are various approaches to test web applications depending on the authentication mechanisms. The generalised approach is to understand how the authentication process works and use that information to explore the possible means hackers may use to circumvent the authentication mechanism. The tester may also refer to the existing attacking techniques to construct test cases for detecting the corresponding security flaws.

Conclusion

While web applications offer great convenience and flexibility to universities, they also expose universities' information systems and resources to more security vulnerabilities. Hackers may obtain sensitive information or even forge the identities of authorised users for malicious purpose. Management should consider security as a fundamental element when developing web applications and conduct adequate security testing to detect any security flaws.

Reference:
http://www.pcworld.com/businesscenter/article/221504/8_security_tips_from_the_hbgary_hack.html