

WEB 2.0 AND SECURITY

February 2008

© The Government of the Hong Kong Special Administrative Region

The contents of this document remain the property of, and may not be reproduced in whole or in part without the express permission of the Government of the HKSAR.

Disclaimer: Whilst the Government endeavours to ensure the accuracy of the information in this paper, no express or implied warranty is given by the Government as to the accuracy of the information. The Government of HKSAR accepts no liability for any error or omission arising from or related to the use of the information.

TABLE OF CONTENTS

Summary	2
I. What is Web 2.0?.....	3
II. Possible Security Threats and Concerns with Web 2.0.....	4
Threats posed by A High Participatory Culture	4
Security Threats related to AJAX	5
Security Threats related to Web Feeds.....	6
Data Privacy and Intellectual Property Rights	6
Impact on Internet Resources.....	7
III. Precautionary Measures	8
Protection for IT Practitioners.....	8
Protection for End-Users.....	11
IV. Conclusion	12

SUMMARY

Web 2.0 has become a catch-all phrase to describe websites that are more than just plain, static information dissemination. In the Web 2.0 world, the web serves as a platform for people to create and share their own content online in form of blogs (or weblogs), videos or photos. RSS is also embraced by many websites to give a summary of news headlines of interests. By making this platform as user-friendly and accessible as possible, people are encouraged to visit often, and to post and view content. Popular social networking sites, such as MySpace, or video sharing sites, such as YouTube, are prime examples employing Web 2.0 technologies.

However, there are two sides to every coin. While Web 2.0 technologies offer many advantages in terms of enriching the Internet and improving the user experience, they are also bringing a number of security concerns and attack vectors into existence. In this paper, we discuss the possible threats introduced by Web 2.0 technologies and suggest corresponding counter-measures.

I. WHAT IS WEB 2.0?

“Web 2.0” does not have a precise definition. To many people, the phrase refers to special web application technologies and websites, such as weblogs and wikis, which use the Internet in a collaborative way to provide services to users. Web 2.0 relies in large part on the user-as-publisher model of interaction and allows for user-created content to be developed and implemented by large groups of individuals. These technologies are increasingly being used by companies for better staff collaboration and communication. O’Reilly outlined seven principles that can help distinguish the core features of Web 2.0 applications¹. A number of Web 2.0 services and sites have appeared in the recent years. Some popular services are YouTube (<http://www.youtube.com>), Facebook (<http://www.facebook.com>), MySpace (<http://www.myspace.com/>), etc.

¹ <http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

II. POSSIBLE SECURITY THREATS AND CONCERNS WITH WEB 2.0

A coin always has two sides. While Web 2.0 technologies offer many advantages in terms of enriching the Internet and improving the user experience, they are also bringing a number of security concerns and attack vectors into existence. Because one characteristic for a Web 2.0 application is to encourage greater user involvement, the exposure of the individual user or client to security threats and vulnerabilities increases. The following are common threats in the Web 2.0 arena.

THREATS POSED BY A HIGH PARTICIPATORY CULTURE

Web 2.0 enables individuals to create and host content on a variety of collaborative platforms such as blogs and wikis. However, depending on how these interactions are managed, the possibility exists for malicious exploitation on those platforms, becoming distribution points for links to fraudulent websites, malicious code, and other security threats such as spyware. Attackers will often take advantage of the implied trust between a community of individual developers and the sites hosting content to compromise individual users and/or websites. For example, a fake video file containing the Zlob trojan was embedded in a video on the YouTube website, and affected a number of users². Mechanisms to assure the authenticity and trust-worthiness of publishers will become a major consideration as the content updating process becomes more decentralised.

² http://www.theregister.co.uk/2007/06/20/youtube_security/

Another possible threat is leakage of corporate information. When a staff member writes and shares information or opinion through a blog, it becomes harder for the organisation to control what information has been published and officially released. Sensitive corporate and customer information may be leaked. In addition, attackers might be able to harvest information about the organisation and launch a social engineering attack on the organisation.

SECURITY THREATS RELATED TO AJAX

In order to provide a rich user experience, many Web 2.0 sites have employed lightweight user interface code such as asynchronous JavaScript and XML (AJAX). In the traditional client-server models, the majority of requests are handled and processed on the server side. AJAX allows a higher proportion of requests to be processed on the client side. This may give malicious users more opportunity to modify any application code running on a client computer when probing and testing an application for vulnerabilities.

As AJAX can be used in conjunction with a large number of web services, enabling connectivity between them, this could present additional attack vectors into which malicious users could inject hostile content. As an example, AJAX could serve to amplify the potential of cross-site scripting (XSS) attacks, which seek to inject code into legitimate websites in order to mislead users and steal their information. Not only would this allow an attacker to steal confidential information, it could also allow an attacker to insert malicious code onto the host through malicious scripts.

One security vendor has categorised a new class of vulnerability as *JavaScript Hijacking*. This class of vulnerability specifically affects Web 2.0 AJAX-style web applications³. Through this vulnerability, an unauthorised party can read confidential data contained in JavaScript messages. An application may be vulnerable if JavaScript is used as a data transfer format, and in particular, when sensitive or confidential information is being handled.

SECURITY THREATS RELATED TO WEB FEEDS

Under the Web 2.0 characteristic of decentralised and distributed content, web information is distributed to other sites via lightweight syndication protocols, such as RSS and Atom. These web feeds allow both users and websites to obtain content headlines and body text without visiting the site in question. There is no standard mechanism to authenticate the publishers of feed entries. As such, malicious attackers can make use of these web feeds to inject literal JavaScript into the RSS feeds to generate attacks on the client browser. All the attacker needs to do is to insert a literal script injection into standard RSS or Atom elements, such as the Title, Link or Description XML tags for RSS. When an end-user visits this particular website and loads the page with the RSS feed, the malicious script will be executed.

DATA PRIVACY AND INTELLECTUAL PROPERTY RIGHTS

A further point should be noted with regard to user concerns about privacy and rights to protect their own data. In many of the early Web 2.0 applications, copyright was only loosely enforced. For example, Amazon lay claim to all and any reviews submitted to the

³ <http://www.securityfocus.com/news/11456>

site, but in the absence of enforcement, people may repost the same review elsewhere. However, as organisations begin to realise that control over data may be their chief competitive advantage, there will be greater attempts to control access or distribution of data.

IMPACT ON INTERNET RESOURCES

The wide adoption of AJAX may also impact the network. The use of AJAX technology can result in frequent (non-user-triggered) or even constant data exchanges between a client and a server, and any excessive delay or data loss during these data transfers may have effects that are visible to the users. While it is possible that AJAX (and AJAX developers) will evolve to suit the network (for example, handling the delay or loss in the background), users may also demand more consistent and reliable network performance than today's Internet can deliver. Currently, any request for consistent network performance is usually met by IP QoS mechanisms, but implementing QoS on the public Internet will be considerably more challenging than on a private intranet. Certainly other solutions may surface, making this an interesting area to watch in the future.

III. PRECAUTIONARY MEASURES

The security concerns discussed in the previous section need to be addressed with necessary precautionary measures. Below we list a small selection.

PROTECTION FOR IT PRACTITIONERS

Security through Design

According to Gartner, poor application development and a lack of oversight when integrating security best practices and tools into the System Development Life Cycle (SDLC) are two of the biggest security issues facing Web 2.0 developers. In the rush to ride the tide of these new services, Web 2.0 applications may not receive the same level of security auditing as traditional client-based applications and services.

Like other applications, security considerations should be taken into account at all phases of the SDLC. In particular, implementation of the proper authentication controls, input validation, error handling controls, and so on, is essential to avert threats that may result in unauthorised intrusion. To further ensure adequate protection being implemented, security risk assessments should be conducted before launching any new applications or program releases.

Security through Controls

As mentioned, Web 2.0 applications are highly client centric. This approach may pose significant threats to a system if adequate controls are not in place. To build an interactive and secure Web 2.0 application, a secure architecture with appropriate controls is an essential component. Some of the building blocks of this architecture include:

1. A solid session management scheme to ensure that authentication and authorisation is performed inside a trusted part of architecture.
2. Data validation is performed in both directions on the server-side at various layers to limit or prevent injection and other forms of attacks.
3. All calls to backend services are performed by trusted server-side business logic.

Security through Openness

Given that open-source software or APIs are exposed to open scrutiny, they are usually developed with security in mind. Hence, they generally have increased security built in. Instead of following a proprietary approach, proven security protocols and industry standards should be used. If open source software or APIs are used, the software should be tracked to ensure that all licences are valid for use, and published vulnerabilities from these open source software solutions should be addressed in time.

Corporate Governance on Web 2.0

Although current Web 2.0 services are mostly public services, outside the organisation, management still needs to be aware of the risks that may impact corporate members who have access to these services. Policies should be established to protect sensitive corporate and/or customer information, and ensure this will not be disclosed in open websites such as blogs. Regular awareness training should also be conducted to educate staff about the

company's IT Security Policy and strengthen security awareness around the risks associated with these new technologies.

To avoid the risks associated with web feeds, only data feeds from reputable sources should be trusted. For application developers who provide web feeds, preventive measures such as white-listing only those necessary HTML tags should be deployed. This can reduce the possibility of XSS attacks on web feeds.

The following are additional best practices that IT practitioners should consider:

1. Although wikis can lead to broader and more rapidly evolving coverage of topics, they are vulnerable to misinformation and anonymous authors could make malicious or unauthorised changes to information being published. In cases where a wiki-type application is to be deployed, editorial controls should be imposed to restrict updates to only legitimate and authorised areas. Proper authentication and access control should also be imposed to better ensure the integrity of content.
2. When a blog is used to communicate an organisation's vision, or for other promotional purposes, care must be taken to avoid possible leakage of sensitive or proprietary information. Monitoring and filtering of all Blog content should be implemented. Acceptable use policies for blogs should also be distributed to all users.
3. Like other applications, Web 2.0 programs should undergo vigorous vulnerability testing to identify loopholes and uncover any weaknesses, including command injection, cross-site scripting and buffer overflow vulnerabilities. All problem areas should then be fixed and security threats mitigated before the application is released into the production environment. In addition, periodic security assessments should be conducted on a regular basis.

PROTECTION FOR END-USERS

For end-users, relevant security regulations and policies should be observed, and web feeds from suspicious sources should not be trusted. In balancing functionality with security, one may consider limiting the use of JavaScript in the browser to protect from malicious script⁴ attacks. In addition, the latest security patches recommended by the anti-virus product vendors should be applied.

When writing and publishing blogs, care must be taken to protect one's own personal data, as well as sensitive or even confidential information about other persons or organisations. For example, personal information such as email addresses, mobile phone numbers, or even personal photos should not be disclosed without good reason. The Office of the Privacy Commissioner for Personal Data has also advised that young people publishing on the web should consider the risks involved when releasing personal information in the public domain, so as to protect themselves from possible abuse or illegal activities⁵. Young people should also be reminded to respect others people's privacy before disclosing personal data of others over the Internet.

⁴ http://www.cert.org/tech_tips/malicious_code_FAQ.html

⁵ http://www.pcpd.org.hk/english/infocentre/press_20070829.html

IV. CONCLUSION

Web 2.0 brings new developments to the web and the Internet. However, new security risks also need to be taken into account. In particular, attackers may shift their focus from the server side to the client side, which is usually considered the weakest link in the security chain.

Many lessons in security implementation learnt from the earlier Web 1.0 era can be applied to Web 2.0. Although the increased capabilities of Web 2.0 may bring increased risks, many basic security principles routinely put in place for application development in Web 1.0 apply here as well.

The fundamental tenets of application security should not be ignored and overlooked. Security should be built into Web 2.0 applications from the earliest stages of development. Security processes, controls and management oversight should be in place before applications are deployed. Periodic and ongoing security risk assessments should be conducted to identify and fix vulnerabilities. Management, application developers and end-users all need to work together to tackle these challenges in the new era of Web 2.0.