2009 Industry Report

TOP WEB 2.0 SECURITY THREATS

Secure Enterprise 2.0 Forum

TABLE OF CONTENTS

1 PREFACE

1.1 GOALS

This document outlines web application security threats unique or typical to Web 2.0. The document should serve as a guideline for assessing risk in Web 2.0 applications.

1.2 STRUCTURE

In order to focus on Web 2.0 specific threats, the document:

- Defines Web 2.0
- Lists key security aspects of Web 2.0 systems
- Discusses specific security threats that are typical to Web 2.0 applications

1.3 TARGET AUDIENCE

This document is intended for security practitioners and for individuals considering the use of consumer Web 2.0 tools and services for professional purposes. Organizations implementing social networks, widgets, RSS, instant messaging, mobile internet applications, and other Web 2.0 tools and services will benefit from reading this document.

1.4 THE SECURE ENTERPRISE FORUM

The Secure Enterprise 2.0 Forum is a group of concerned organizations and individuals who understand that the use of Web 2.0 tools and services for professional purposes is inevitable. On the contrary, they embrace the trend rather than fear it. However, these people realize that improper use of new technologies foreshadows a nightmare of information security breaches and public relations debacles.

The Forum seeks to *promote the secure use of Web 2.0 to do business*.

The Forum is comprised of top executives at Global Fortune 500 companies that are ready to address the security challenges posed by Web 2.0 technologies, such as wikis, blogs, RSS, widgets and gadgets, personalized homepages, social networks and social bookmarking, which are becoming increasingly popular in the enterprise. The Secure Enterprise 2.0 Forum promotes awareness, industry standards, best practices, and interoperability issues related to the introduction of consumer technology into the workplace.

2 Web 2.0 Defined

Wikipedia defines Web 2.0 as:

> *"The changing trends in the use of World Wide Web technology and web design that aim to enhance creativity, communications, secure information sharing, collaboration and functionality of the web[1]"*

A key element in the definition is that Web 2.0 does not focus on a specific technology, but rather on a new paradigm of using the web. Some new technologies, such as AJAX[2] are used within this paradigm, but Web 2.0 transcends the actual technology. In fact, security threats posed by Web 2.0 are first and foremost those that stem from novel uses of technology that Web 2.0 affords. Threats specific to particular technologies are only peripherally associated with Web 2.0, since technologies used in for Web 2.0 will may change over time.

While the focus of Web 2.0 threats emanate primarily from new usage patterns, several technologies are so widespread in Web 2.0 applications, that security threats associated with them are characteristically considered Web 2.0 security threats. Examples of such technologies include AJAX, widgets, and application platforms such as blogs, wikis[3] and social networks.

To fully analyze the security threat of Web 2.0 applications, it is necessary to first introduce key aspects of common Web 2.0 concepts and trends.

2.1 W1 - User Generated Content

In contrast to the static nature of Web 1.0, Web 2.0 systems rely heavily upon user generated content. In fact, Web 2.0 has been described as the "participatory Web."[4] For example blogs and photo sharing services enable consumers to add and update their own content. Other systems such as community mapping or wikis mash the information from multiple users to create a single database.

2.2 W2 - Mashups & Web Services

Open sharing of information implies the open sharing between disparate systems. To do this, Web 2.0 systems include interfaces that allow other Web 2.0 systems to communicate with them, usually using a common API based on XML and known as Web Services[5]. A "mashup" is an element that combines information from multiple systems using Web Services to provide a aggregate service.

Personal home pages are an example of a mashup that combines information from disparate sources to a single, personalized web page. Another example is the use of public mapping services to embed specific maps within web sites.

2.3 W3 - Consumer and Enterprise Worlds Convergence

By focusing on the individual rather than the organization, Web 2.0 blurs the border separating an organization from the outside world. For example, people's blogs often contain elements of both their

[1] Web 2.0 definition, Wikipedia, http://en.wikipedia.org/wiki/Web_2.0

[2] Ajax definition, Wikipedia, http://en.wikipedia.org/wiki/Ajax_(programming)

[3] Wiki definition, Wikipedia, http://en.wikipedia.org/wiki/Wiki

[4] Bart Decrem, Flock official blog, http://www.flock.com/node/4500

[5] Web Services definition, Wikipedia, http://en.wikipedia.org/wiki/Web_service

personal life and their professional life, and social networks incorporate both professional contacts and personal friends (e.g. LinkedIn and Facebook).

Services available to an organization from public sources through web services represent an opportunity for an organization to increase efficiency and streamline operations. Services no longer need to be provided strictly by enterprise software installed within the data center, but rather, they can be provided by both internal systems, augmented by external systems running as a service (i.e. SaaS). As an example, today many organizations use publicly-available Google Maps to provide a geographical overlay to their existing applications.

Another key contributor to the convergence of consumer and enterprise worlds is the proliferation of highly-portable computing platforms, such as "netbooks" and web-enabled mobile phones; two devices that people use for both work and personal purposes.

2.4 W4 - Diversity of Client Software

Mashup and syndication services dictate that information and software functions are available across many different display technologies and environments. The diversity of platforms needed to support these services extends to:

- A wide variety of hardware platforms such as computers running different operating systems and mobile phones
- Software clients running outside of the browser such as a desktop widgets
- Shared browser windows such as personal home pages.

2.5 W5 – Complexity & Asynchronous Operation

Despite the huge benefits afforded by Web 2.0; they do not come without a cost. To enable increased user interaction, integration APIs and web applications need to be more complex and they need to support an ever-increasing set of clients. The most profound technical complexity introduced by Web 2.0 applications is the asynchronous request, often referred to as AJAX (for "Asynchronous JavaScript and XML"). As the name implies, AJAX requests may be triggered automatically and not as the result of user interaction, for example in order to repaint just the portion of a screen that changed, rather than the entire screen. In contrast, traditional applications issue requests to a server when a user presses a control or a link and the request results in a display of a new page.

3 Top Web 2.0 Security Vulnerabilities

3.1 V1 - Insufficient Authentication Controls

3.1.1 Vulnerability Description

Information systems typically assume there are a small number of privileged accounts that belong to senior or experienced users. A workflow usually ensures that if changes are made by less-experienced users, they need to be authorized before it affects the system.

Since the content of a Web 2.0 application is often trusted in the hands of many users, these assumptions are no longer valid. Therefore, a risk exists that a less experienced individual might make a change that adversely affects the overall system.

3.1.2 Exploit Scenarios

V1.1 Weak Passwords

Significant contributors to online sites or resources are typically given administrative rights over the sites. Quite often, these individuals use simple-to-guess passwords or employ trivial password-reminder questions. Password reminder questions are especially dangerous, since information needed to answer them correctly is often available online, particularly for famous people (e.g. “mother’s maiden name,” “pet’s name,” etc.). Attackers can therefore login to these sites pretending they are the authentic user, and then post false information or perform unauthorized administrative actions.

(Web 2.0 root cause: W1 – User contributed content)

V1.2 Insufficient Anti-Brute Force Controls

Lack of sufficient protection from brute force attacks[6] enables attackers to guess users’ or administrators’ passwords. Even when the main login function is sufficiently protected, ancillary authentication functions such as “remember me,” password reminders, and logout are not protected from brute force attacks. The stolen password enables the attacker to masquerade as the authentic user.

(Web 2.0 root cause: W1 – User contributed content)

V1.3 Clear Text Passwords

When using AJAX, widgets, or mashups, passwords may be sent across the Internet unencrypted and stored outside the control of the host web site. In both cases, the information is easily available to anyone with a network traffic analyzer.

(Web 2.0 root causes: W2 – Mashups, W4 – Diversity of client software, W5 - Complexity)

V1.4 Single-Sign-On

In Web 2.0 personalized homepage and desktop widget environments, it is highly inconvenient to log into each widget or application separately. As a result, those environments typically incorporate a single-sign-on

[6] Brute force attacks, The Web Application Security Consortium Threat Classification, http://www.webappsec.org/projects/threat/classes/brute_force.shtml

mechanism that stores (either on the desktop or in the cloud) user credentials for accessing different resources. This exposure of passwords represents a potential leak of user credentials.

(Web 2.0 root cause: W2 – Mashups, W4 – Diversity of client software)

3.1.3 Known Incidents

- WHID 2009-2: Twitter Accounts of the Famous Hacked - a hacker used a brute force dictionary attack against a Twitter administrator account and broke into 33 user accounts including Barak Obama's and Britney Spears'.
- WHID 2007-17: Big Brother's big bother - The site of "Big Brother", a reality show in Australia issued duplicate session IDs to different users, since the session ID pool was exhausted. Naturally, the 2nd person to get the same session ID got to see all the details of the first user issued the same ID.
- A common phenomenon is for celebrities' online identities to be hacked. Some relevant stories are:
  - WHID 2008-56: Soulja Boy MySpace Hacked,
  - WHID 2009-15: Kanye West has been Hacked,
  - WHID 2009-11: Lil Kim Facebook Hacked,
  - WHID 2008-55: Hackers hijack bitchy fashion blog

3.2 V2 - Cross Site Scripting (XSS)

3.2.1 Vulnerability Description

In a stored cross site scripting (XSS) vulnerability, a malicious input sent by an attacker is stored by the system and then displayed to other users. Systems that allow users to input formatted content, such as HTML, are especially susceptible to XSS, since malicious input can be easily created, for example via scripts.

3.2.2 Web 2.0 Relevance

As the OWASP Top 10 notes[7], this type of functionality, in which many users create content viewed by others, is typical to Web 2.0 systems such as social networks, blogs or wikis, making Web 2.0 applications especially vulnerable to XSS.

3.2.3 Exploit Scenarios

V2.1 Insufficient Limits on User Input

Web 2.0 applications rely heavily upon user generated input. In order to allow the user extensive control over the content design, applications often allow HTML tags that are not safe and that can be abused via XSS.

(Web 2.0 root cause: W1 – User contributed content)

3.2.4 Known Incidents

- WHID 2008-32: Yahoo HotJobs XSS – Hackers exploiting an XSS vulnerability on Yahoo HotJobs used obfuscated JavaScript to steal session cookies of victims, who were in turn sent to an external server

[7] The Open Web Application Security Consortium (OWASP) Top 10 A1 - XSS, http://www.owasp.org/index.php/Top_10_2007-A1

in the US. The stolen cookies represented a Yahoo-wide cookie; therefore by stealing it, hackers gained control over every service accessible to the victim within Yahoo, including Yahoo! Mail.

- WHID 2008-20: XSS Worm At Justin.tv Affects 2525 Profiles - A proof of concept XSS worm crawled justin.tv, a popular life-casting platform. The worm succeeded in planting self-replicating code on 2525 accounts in less than 24 hours before the vulnerability was fixed.
- WHID 2007-69: The Orkut XSS Worm – A vulnerability in the social networking site Orkut, allowed users to inject HTML and JavaScript into their profiles. This set the stage for a persistent XSS worm that appears to have affected more than 650,000 Orkut users.
- WHID 2005-11: XSS Worm Hits MySpace - The Samy XSS worm at MySpace is now a classic, both a sophisticated attack and a well documented one; it became a case study in the web application security field.
- WHID 2007-86: Mac Blogs defaced using XSS

3.3 Cross Site Request Forgery (CSRF):

3.3.1 Vulnerability Description

In Cross Site Request Forgeries (CSRF)[8] the victim visits an innocuous-appearing, but malicious web site. While victim's browser renders the site, malicious site code generates requests to a different site to which the victim is authorized, for example through a persistent cookie. Such requests can perform operations on behalf of the victim.

3.3.2 Web 2.0 Relevance

Web 2.0 applications are potentially more vulnerable to CSRF due to the use of AJAX. In legacy applications, most user-generated requests produced a visual effect, making CSRF attacks easier to detect. Web 2.0 systems on the other hand, typically make available extensive APIs for use by other system- or client-side code without a direct visual effect. The large number of functions offered, and the lack of visual feedback, make AJAX-based Web 2.0 applications more susceptible to CSRF.

CSRF attacks are limited due to the "same origin policy"[9] which enables malicious code to issue requests, but not inspect the returned values. However different Web 2.0 environments, such as desktop widgets or personalized homepages may not enforce the same origin protection.

3.3.3 Exploit Scenarios

V3.1 Credential Sharing Between Gadgets

Cross Site Request Forgery is enabled by persistent login credentials stored on the client and sent with each request to the client automatically, such as session cookies. Mashing multiple sessions with different targets in the same desktop or a personal homepage broadens the attack surface for CSRF.

[8] Chris Shiflett, Security Corner: Cross-Site Request Forgeries, http://shiflett.org/articles/cross-site-request-forgeries

[9] Same Origin Policy definition, Wikipedia, http://en.wikipedia.org/wiki/Same_origin_policy

V3.2 CSRF USING AJAX REQUESTS

An AJAX interface enables invisible queries of a web application by the client, therefore opening a large attack surface for cross site request forgeries.

(Web 2.0 root cause: W5 - Complexity & Asynchronous Operation)

V3.3 LENGTHY SESSIONS

In Web 2.0 personalized homepages and desktop widget environments, consumers are often connected for a lengthy period of time, and therefore prefer to minimize required login procedures dictated by timeouts. As a result, within Web 2.0 environments, session expiration times are typically quite high, enlarging the risk of session base attacks such as cross site request forgery and session hijacking.

(Web 2.0 root causes: W2 – Mashups, W4 – Diversity of client software)

3.3.4 KNOWN INCIDENTS

- WHID 2009-4: Twitter Personal Info CSRF -By exploiting a CSRF bug in Twitter, site owners can get Twitter profiles of their visitors.

3.4 PHISHING

3.4.1 VULNERABILITY DESCRIPTION

In a phishing attack, a victim receives by e-mail a request to complete an online form with sensitive information, which is then sent to the attacker. The online form is placed on a fraudulent web site.

By using e-mail to point victims to a fake web site, phishing does not rely on software weaknesses. However, clear differentiation between the fake and the real web site it imitates, is a key in preventing the victim from succumbing to the fraud attempt.

3.4.2 WEB 2.0 RELEVANCE

Web 2.0 mashups, widgets, and multitude of dissimilar client software, makes it much hard for consumers to distinguish between genuine and fake sites, thusly enabling more highly-effective phishing attacks.

3.4.3 EXPLOIT SCENARIOS

V4.1 PHONY WIDGETS

Phishing is a social engineering fraud attack and therefore mitigation is based on education. Consumers are trained to recognize the symbols of a safe site such as the domain name and the SSL certificate. When one site's content is embedded in another site using Web Services or provided as a widget, these protection symbols are not present, thusly increasing the risk of a phishing attack.

(Web 2.0 root cause: W2 – Mashups, W4 – Diversity of client software)

V4.3 PHONY CONTENT USED FOR PHISHING

Web 2.0 applications (which typically allow consumers to post information), raise the risk of phishing since attackers can plant their fraudulent information on a site considered safe by the consumer.

(Web 2.0 root cause: W1 – User Contributed Content)

V4.3 XSS Exploited for Phishing

As noted above, user-generated content raises the risk of XSS. XSS is commonly used for phishing attacks, since it allows attacker-generated content to appear as if it comes from a trusted site.

(Web 2.0 root cause: W1 – User Contributed Content)

3.4.4 Known Incidents

- WHID 2006-26: Yahoo XSS used for phishing - An XSS vulnerability in Yahoo Mail is actively exploited for targeted phishing.

3.5 Information Leakage

3.5.1 Vulnerability Description

Web 2.0 applications promote user-generated content, enabled people to contribute to sites at home and at work. As the lines between work and private life blur, people may inadvertently publish information considered sensitive by their employer. Even if people are careful and do not leak sensitive information per se, the aggregation of many small data "non-sensitive" items may reveal sensitive information. For example, by analyzing information about all the employees of a small company in a social network, intelligence on the company can be gathered.

Just like humans, the Web 2.0 promotion of Web Services may cause web applications to expose too much information. Undocumented Web Services might enable access to sensitive information and certainly the access to documented Web Services, such as Web Services Description Language (WSDL)[10] files, provides valuable information to attackers.

3.5.2 Exploit Scenarios

V5.1 Sensitive Information Posted to Web 2.0 Sites

Web 2.0 applications are often provided by free public services such as social networks, blogging sites and information sharing sites. Using these sites is essential, and the same advantage cannot be gained from using alternate non-public systems. Moreover, some of those services, primarily social networks and blogging are highly user-centric. Therefore information published by individuals often mixes personal and corporate information.

Often, the blurred boundary between personal and professional experiences leads to information leakage.

(Web 2.0 root cause: W1 – User contributed content)

V5.2 Information Aggregation In Social Networks

A secondary risk associated with online posting of information is that while each piece of information can be benign, the overall impact of the information may have unintended consequences. The best example of this is the aggregation of corporate information within a social network such as LinkedIn. The number of employees, the roster of previous employees, and a list of job titles is invaluable information about the company, particularly to competitors. In addition, such a large amount of information about a company is very useful for

[10] WSDL, W3C, http://www.w3.org/TR/wsdl

launching attacks on the company, such as brute force attacks, social engineering attacks, or spearheaded phishing and malware attacks.

(Web 2.0 root cause: W1 – User contributed content)

V5.3 Mix of Private and Public Information on Public Services

Many Web 2.0 services mix personally-protected information with public information. For example, exposing a Google personal map or a Picasa personal album, requires only a single click. The result is that sensitive information such as pictures, maps, or contacts intended for internal use but hosted on external Web 2.0 sites, can easily leak to the public.

(Web 2.0 root cause: w3 – Consumer and enterprise worlds convergence)

V5.3 Easy Retrieval of Information Through Web Services

Web Services' main goal is to provide easy and automated access to information. If not designed and implemented properly, these services may provide access to more information than originally intended, leading to information leakage.

(Web 2.0 root cause: W4 – Mashups)

3.5.3 Known Incidents

- WHID 2008-26: Palin's private e-mail hacked, posted to Net - The activist group called "anonymous," best known for its jousts with the Church of Scientology, hacked into the private Yahoo e-mail account of Alaska Gov. Sarah Palin, the Republican candidate for US Vice President. The hack was done by online researching for information required to reset her password using a Yahoo password-reminder question.

3.6 Injection Flaws

3.6.1 Vulnerability Description

Web 2.0 is particularly vulnerable to new types of injection attacks, including XML injection[11], XPath injection[12], JavaScript injection and JSON injection. In addition, because they rely heavily on client side code, Web 2.0 applications more often perform some client-side input validation which an attacker can bypass.

3.6.2 Exploit Scenarios

V6.1 XML Injection

Web Services and AJAX are key technologies in Web 2.0 applications and both use XML. XML Injection is an attack in which user-supplied input is inserted into XML records without sufficient validation. The injected input then modifies the structure of the XML record by adding tags (and not just content).

[11] Attacking Web Services, Alex Stamos, http://www.owasp.org/images/d/d1/AppSec2005DC-Alex_Stamos-Attacking_Web_Services.ppt

[12] XPath Injection, The Web Application Security Consortium Threat Classification, http://www.webappsec.org/projects/threat/classes/xpath_injection.shtml

(Web 2.0 root cause: W4 – Mashups & Web Services, W5: Complexity & Asynchronous Operation)

V6.2 XPATH INJECTION

Web Services and AJAX are key technologies in Web 2.0 applications and both use XML. XML applications often use XPath as the query language to manipulate XML data in much that same way that SQL is used to manipulate relational database records. XPath injection is an attack in which specially-crafted inputs used by the application within an XPath query, alter a query to achieve the attacker's goals.

(Web 2.0 root cause: W4 – Mashups & Web Services, W5: Complexity & Asynchronous Operation)

V6.3 JSON INJECTION

JSON (JavaScript Object Notation) is a protocol often used in AJAX applications for transferring information between the client and the server. Since JSON format is valid JavaScript code, some applications process the code in order access the data. By injecting malicious JavaScript code into the JSON structure, an attacker can force execution of malicious code.

(Web 2.0 root cause: W5: Complexity & Asynchronous Operation)

3.6.3 KNOWN INCIDENTS

- WHID 2008-47: The Federal Suppliers Guide validates login credential in JavaScript - The guide is presumably limited to federal procurement agents only, but the credential checking was done on the client in JavaScript and for a single global user name and password.

3.7 INFORMATION INTEGRITY

3.7.1 VULNERABILITY DESCRIPTION

Data integrity, or correctness, is one of the key elements of data security. While the loss of integrity is often due to a malicious hack, unintentional misinformation also leads to loss of integrity. When Senator Kennedy death was announced (pre-maturely) on Wikipedia, the inaccuracy was big enough and the site used by enough people for the mistake to be discovered promptly, however more subtle changes in less-visited sites may represent a more significant issue.

Even if changes introduced are not malicious, the aggregation of individual inaccuracies, may lead to a relatively large distortion of the truth when combined into a single story.

3.7.2 EXPLOIT SCENARIOS

Note: This section discusses only information published on a web site by legitimately-authenticated users. For integrity issues related to fraudulent authentication refer to V1 - "Insufficient Authentication Controls".

V7.1 AUTHENTICATED USERS PUBLISH FRAUDULENT INFORMATION

Since Web 2.0 systems allow many users to publish information, it is difficult to ensure that all of them are trustworthy. Malicious users can take advantage of a hosting site to publish inaccurate information. An example of such an exploit would be to publish erroneous or misleading information in order to affect stock prices.

(Web 2.0 root cause: W1 – User contributed content)

V7.2 Rumor Mill

Web 2.0 systems are a very good vehicle for spreading rumors, true or not. Often, the original source of the information is not aware of the intensity and speed of the propagation of information. As such, small inaccuracies may have unintended and dire consequences.

(Web 2.0 root cause: W1 – User contributed content)

3.7.3 Known Incidents

- WHID 2009-14: My.BarackObama.com Infects Visitors With Trojan - my.barackobama.com, an open blogging service which was part of Obama's campaign web site, was exploited to point readers to malware-infecting content. There was no malicious code on the Obama site, however, HTML code that resembled a YouTube-embedded video redirected readers to an external site that carried the malware.
- WHID 2009-13: Wikipedia Biography Hacking - A wiki is a platform that allows many contributors to alter shared content. This is the Wiki philosophy; Wikipedia is the premier example of this technology. Therefore, it is not surprising that Wikipedia is a prime target to content spoofing, such as the story about the unexpected demise of two US senators during Obama's inauguration.
- WHID 2009-8: Wired.com Image Viewer Hacked to Create Phony Steve Jobs Health Story - John Abell from Wired magazine often writes about the Apple CEO's health condition. However, this report about Jobs' suffering a cardiac arrest was neither his nor was it true. The culprit was a Wired public image viewing utility that allows people to upload images. Using this facility an image was uploaded and then presented as part of the Wired web site – banner and domain included.

3.8 Insufficient Anti-automation

3.8.1 Vulnerability Description

The programmatic interfaces exposed by Web 2.0 applications enable an attacker to automate attacks. Two examples of automation discussed above are brute force attacks and CSRF. Other examples include the automated retrieval of a large amount of information and the automated opening of accounts, for example as part of a phishing attack.

A Web 2.0 application must include anti-automation mechanisms not commonly found in legacy applications, such that Captcha or request throttling.

3.8.2 Exploit Scenarios

Note: This section discusses only automation exploits that are done by legitimately authenticated users and using legitimate applications interfaces. An example of another automaton risk is detailed in V1.2 "Insufficient Anti-brute Force Controls".

V8.1 Web Spam

A common automated exploit is the automated publishing of large quantities of information, using Web 2.0 applications' acceptance of user-generated content. This phenomenon is often regarded as Web Spam. The most common use of Web Spam is to post links. Since search engines use the number of links pointing to a

web site as a measure of its popularity and therefore its ranking, such spam, referred to as comment spam, can help a site achieve top billing in a search engine result set.

(Web 2.0 root cause: W1 – User contributed content)

V8.2 Automatic Opening of User Accounts

By automatically opening user accounts, attackers can abuse Web 2.0 functionality-on-demand as part of their own application. A good example would be opening a web e-mail account in order to authenticate to a different service.

(Web 2.0 root cause: W1 – Mashup & Web Services)

V8.3 Unfair Advantage on Site

By automating tasks on the site, attackers can gain advantage over other users who do not use automated tools. Examples range from gaming sites to bidding sites to ticket purchase sites, where malicious users can buy all tickets offered and then resell them.

(Web 2.0 root cause: W1 – Mashup & Web Services)

3.8.3 Known Incidents

- WHID 2008-57: Craigslist's Battle against Spammers – Craigslist, losing the battle against automation tools, is a very good example of this serious problem.
- WHID 2008-48: TicketMaster Fighting Hackers Line Jumping – TicketMaster has an ongoing battle with hackers “line jumping” to buy event tickets and then resell them for a higher price.
- WHID 2009-3: Google Trends Falls Victim to a Stunt - An unknown attacker succeeded (more than once) to manipulate Google Trends, a Google service that lists popular search terms. In this case, a symbol presumably denoting “9/11” reached number 2 in the list of hot trends.
- WHID 2007-65: Facebook suing a porn site over automated access - Facebook filed a law suit against a user who employed a botnet to manipulate the Facebook site.

4 SUMMARY

Web 2.0 is both a set of technologies as well as a new set of consumer behaviors. The combination of these two elements has created an enormous opportunity for attackers to exploit online resources for "fun and profit." It is important to understand the implications of these new risks, particularly when considering employing Web 2.0 technologies for professional and commercial use. This document exposed some of the most significant challenges presented by Web 2.0 technologies and consumer behaviors. It is the goal of the Secure Enterprise 2.0 Forum to further expose these threats as well as to promote the secure use of Web 2.0 technologies for business so that organizations can take advantage of the huge opportunities afforded by this next generation of the Web in order to do more business.

For more information about the Forum, go to: www.secure-enterprise2.0.org.