Security Gaps in Social Media Websites for Children Open Door to Attackers Aiming To Prey On Children

Scott White

August 6, 2010

Synopsis

This is a case study of an information security analysis of many leading social media websites designed for children. This paper presents the findings of Web application security reviews of multiple sites, and discusses where the sites are proficient and, more importantly, gaps identified which could allow an attacker to prey on children.

Table of Contents

Background

Without a doubt, social media websites have been one of the most popular things that have evolved on the Internet in the last 10 years. With the birth and success of sites such as Facebook, Twitter, and MySpace, it is no wonder that other spin-off sites have popped up following in their footsteps. Within this subset of social media sites, those designed for children have become a large success. With the popularity of social media sites for children, as with everything, the more popular something is, the more attractive a target it is for hackers and people with malicious intent. Not surprisingly, children's sites, not being as mainstream as other popular sites, have not attracted the scrutiny from the information security community as sites like Facebook and Twitter have.

Types of Sites

The research for this paper was reduced in scope from all social media sites to those geared toward children. The target sites were those which appealed to minors, or those less than 18 years of age. The types of sites varied widely; however, they had the same fundamental functionality features in place. All sites had profile customization, a profile image or avatar of some sort, a means of communicating with others whether it is internal "email," chat rooms, or bulletin board style systems, as well as some sort of entertainment factor. The entertainment factors varied the most, with some examples being:

- Educational games
- Games purely for entertainment
- Maintaining some sort of "thing," ranging from pets to forms of shelter (house, room, apartment), to property (farm, garden, village/neighborhood)
- Games and/or activities to earn some sort of "credits" toward awards, a higher social status, or physical prizes
- Role playing games
- Blogging
- Etc.

With each type of site, there are different characteristics of the user base(s); however, they all have one thing in common: age.

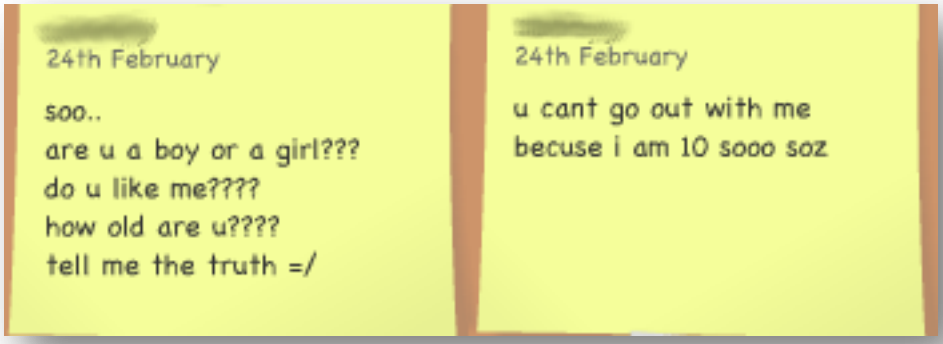
End Users

The Children's Online Privacy Protection Act of 1998 requires Website operators to follow certain rules and regulations for users 13 years of age and under pertaining to collection of personal information, and parental or legal guardian consent to use their site(s). The requirements for social media sites vary greatly from a minimum of 16 years of age to no age requirement with parental consent. Additionally, it is very easy for a child to spoof parental consent. There is no foolproof way to 100% guarantee the identity of an end user or parent/guardian.

From the research conducted, the age of users of sites designed for children varies greatly. Some user profiles blatantly posted that they were not of the required age (most often 13 years old) and put their real ages, some as young as 6, in their "profiles."

Some young users become infatuated with these online "lives," spending large amounts of time online and even having virtual girlfriends and boyfriends whom they've met through such sites.

Due to the curious nature of children under 18 years of age, they are ideal targets for an attacker because they will click on unfamiliar buttons, images, links, etc. to find out more about them. This inquisitive nature is exactly what is exploited in many phishing and drive-by malware attacks. A basic rule of thumb is: give children something "shiny" and they will play with it.



Attacker Motives

Attackers targeting children's social media sites were seen to have few motives. These include pedophilia as well as general computer compromises to aid in bot net activities, SPAM generation, hosting of phishing sites, malware hosting, pedophilia, and platforms to launch further attacks on other targets. Evidence of SPAM was present on some user profiles; however, pedophilia was the main motive observed in the Web sites surveyed. It is no surprise that pedophilia was the most commonly witnessed attacker motive, as other motives would be more easily achieved on other Web sites.

"Pedophilia was the most commonly witnessed attacker motive"

At the end of the day, most all attacks have the same goal, money (excluding defacing, hacktivism, etc.). With social media sites, trust between "friends" usually is exploited. A phishing scheme may be used to obtain a user's login ID and password, and then the compromised credentials are used to access that user's account, most of the time sending SPAM or links to malicious sites to "friends." The "friends" are more likely to fall victim to such attacks as they have an elevated level of trust with the person having the compromised account versus a random stranger.

Vulnerability Landscape

There were numerous security vulnerabilities identified while conducting the research:

- **Cross Site Scripting**

  Without a doubt, cross site scripting was the most prevalent vulnerability identified during the research. Some sites had so many instances of it that the tally was in triple digits. This is not surprising as cross site scripting is number two on the OWASP Top Ten's list of Web application security risks for 2010.

- **Plaintext protocols in use**

  The vast majority of the sites tested failed to utilize application level (OSI Model context) security. SSL/TLS were not used in most cases, leaving usernames, passwords, and other information being sent in plain text. Sniffing wireless traffic on an unsecured network at a favorite wireless hotspot or from a vehicle parked outside a home is extremely easy to do.

"The vast majority of the sites tested failed to properly utilize HTTPS."

- **SQL Injection**

  Several instances of SQL injection were identified, with one even having a database error message being indexed by Google's crawlers. Clearly, database data is free for the taking by attackers. It is unclear as to how much information could be gleaned from various sites as intrusive testing was not permitted on all sites.

- **Insecure SSL/TLS implementations**

  Transport layer security, when used, was rarely configured in the most secure way possible. SSL version 2 was in use, weak (< 128 bit) and sometimes even NULL ciphers were supported, and renegotiation was rarely done in a secure manner.

- **Weak password policies**

  The majority of password policies were lacking with regard to industry best practices. Many passwords did not have minimum length or complexity requirements. This may be intentional, as children are most likely not going to remember complex passwords.

- **File metadata leaks sensitive information**

  The majority of files hosted on the sites tested leaked sensitive information via metadata. The files observed ranged from PDFs, Microsoft Office documents, as well as Open Office documents. The types of information gained from examining such files include but are not limited to first and last names, usernames, internal domain names, absolute file paths, versions of software in use, etc.

- **Insecure HTTP methods**

  Many sites supported the HTTP TRACE/TRACK method(s) which, in turn, facilitate cross site tracing. Additionally, some sites supported WebDAV while having dangerous methods enabled including LOCK, DELETE, and PUT.

- **Unvalidated redirects and forwards**

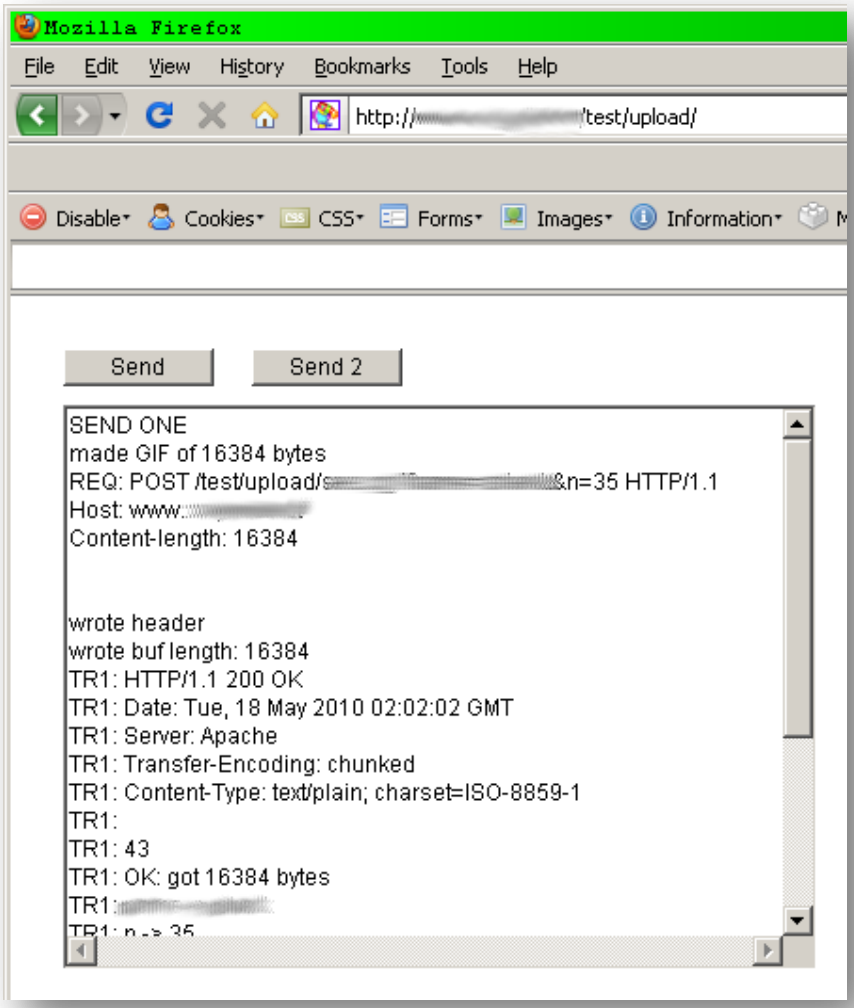
  In addition to cross site scripting, unvalidated redirects and forwards is on the OWASP Top Ten list (number ten) for 2010. Many instances were identified.

- **Source code information disclosure**

Instances of source code information disclosure were identified. These included portions of code in "broken" pages as well as raw source code found via search engines.

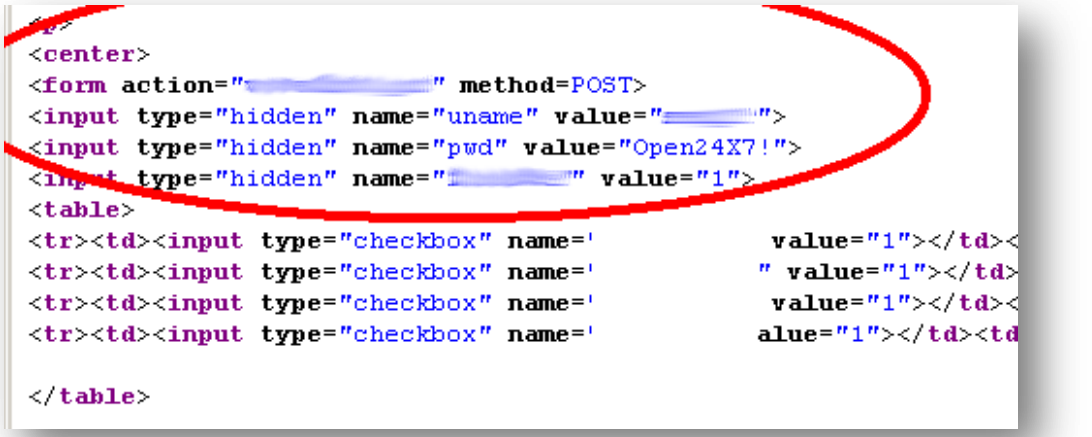
- **Test/debug/backup/default files present in production environment**

  As with any "less than tidy" environment, many test files and pages, debug options, backup files, and default files such as Web server manuals were found. Some of these included file upload functionality, MRTG pages, zip files, as well as staging/preproduction/test environment disclosures.



- **Plain text username/password storage**

  Several sites were identified that presented the user with a plain text version of their username and password. One can assume this is used for "forgot password" password functionality in most cases.



- **Custom session handling and encryption algorithms**

  It was noted that custom session handling and encryption algorithms were being used on some sites. Number three on the OWASP Top Ten, broken authentication and session management is one of the top Web application security risks for 2010. Custom session handling was an issue on several sites, utilizing what appeared to be homegrown session tracking. Sessions did not appear to timeout within several hours on one site. It also was observed that some "encryption" functions were implemented via JavaScript, some of which were custom and others of which used deprecated algorithms such as MD5 for password hashing.

- **Cross site request forgery**

  Cross site request forgery is in the number five spot on the OWASP Top Ten, and also was present on a number of sites. Some sites used "actionable URLs" for much of their operability. An example of an "actionable URL" pertaining to online banking could be a URL that does an action, such as transferring money, like the following: http://www.bank.com/transfer.aspx?frmAcct=3549856&toAcct=4345754&&amount=3434.43. Just by visiting the link, a specific action will be performed on the requester's behalf.

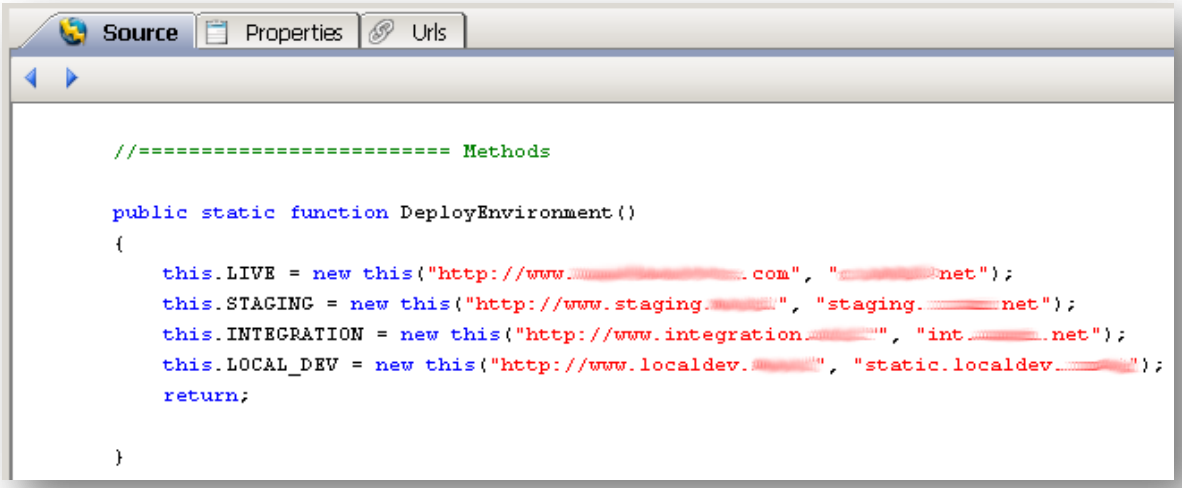
- **Insecure direct object reference**

  One site failed to properly protect internal implementation objects and allowed SecureState to "purchase" others users' virtual possessions by changing a unique ID, while also allowing the price to be specified. A simple worm could have been written to obtain everyone's possessions, for free. Virtual online theft could be performed by simply changing two variables.

- **Sensitive information hard coded in Shockwave Flash files**

  Shockwave Flash (SWF) files are very prevalent on social media sites for children as SWF is a popular platform for game development and content delivery via the Internet. To some people's surprise, SWF files can be decompiled, exposing the underlying Action Script which contains logic behind the files' activity.

Within the source code obtained from such decompilation, much sensitive information was found. A sampling of the sensitive information gleaned from these files includes internal absolute file paths, internal domain names, URLs to test/staging/preproduction environments, usernames, passwords, IP addresses, host names, etc.



- **Wildcard Cross domain policies (* entries)**

  Many sites had wildcards in their cross domain policy files allowing for abuse.

- **Public username lookups**

  Some sites provided a way to search and find other users. With this provided functionality, user enumeration was possible where the username of the individual also was used as the unique identifier to login to the site.

- **No apparent server hardening/minimum security baselines**

  Many sites appeared to lack basic server hardening and minimum security baselines. This was evident due to default settings being in place on many servers.

- **Sensitive HTML/JavaScript comments present**

  Developer comments in HTML and JavaScript would be very helpful to an attacker in some cases. Everything from usernames to email addresses and other information was found.

- **Administrative functionality accessible**

  On some sites, pages with administrative functionality were available to anyone that knew the URL. Security by obscurity was either being relied upon, or there was an authentication/authorization malfunction. Failure to restrict URL access is number eight on the OWASP Top Ten for 2010.

- **Default Error handling including full stack traces and errors indexed by Google**

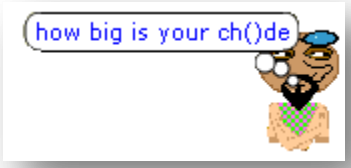
  Several sites were more than happy to give up full error details to search engines. "Google Hacking" was quite fruitful in some cases, revealing full error messages with stack traces on some instances.

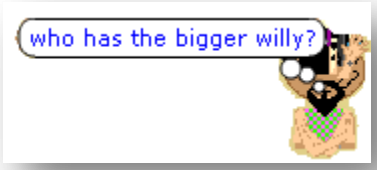
- **Employing Google Docs as a means for private data storage**

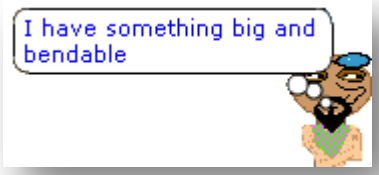
  Google Docs was identified as being used for private data storage for at least one site. Just because SWF files cannot have their traffic intercepted with a local HTTP proxy does not mean they are immune to having their traffic intercepted. Whether or not the developer had this in mind at design time, at least one instance of what appeared to be private data being stored in a public space was found.

- **Ineffective simple filtering in use**

  Many sites employed language filters for chat rooms, internal site "email" messaging, profile content, and other user-input originated content. Some "pedophiles" (as categorized by SecureState's observations of their activity) bypassed such filters with trivial modifications to their content. Examples of this were intentional misspellings, substitution of similar characters, and other tactics commonly used in SPAM email. Some notable examples of filter bypasses included using "peeness" instead of "penis", "c0ck" and "c()ck" instead of "cock", "pussi" instead of "pussy", and alternative slang terms in a sexual context such as "ch()de" and "willy" intended to have the meaning of "penis."

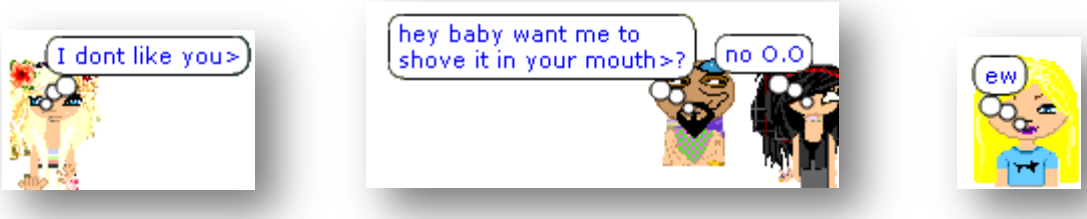






"Users seemed to rarely employ the protection mechanisms that were available to them"

It was observed that the children (using chat functionality) on the sites were well aware of the intentions of such users, and knew the exact meanings of the filter bypasses. Some even gave out telephone numbers, spelling out the numbers to avoid detection. An example of this might be using "two one six, 927, eighty-two hundred" instead of just typing out 216-927-8200. The users seemed to rarely employ the protection mechanisms that were available to them (such as reporting a user for inappropriate activity/content) on the sites. Many tolerated repeated inappropriate comments and sexual innuendos without doing more than verbally expressing disapproval of what was being said.



- **No account lock out**

  On the vast majority of sites, there was no apparent lock out to user accounts after a predetermined number of bad password login attempts. This means that someone could be making repeated attempts to guess your password after learning your username without being blocked or prevented from doing so.

It is very clear that basic Web application security best practices were not being followed wholly on any of the sites examined. Obviously, some were better than others, but no one site emerged as a clear leader of employing information security best practices. In fact, an estimated 95% or more of sites failed to employ basic Web application security measures.

"An estimated 95% or more of sites failed to employ basic Web application security measures."

Exploitation

Of the sites that SecureState was given permission to intrusively assess, many opportunities for abuse and exploitation were identified. Among these, some were purely technical, and some would be most successful combined with social engineering attacks. By far, given the vulnerability landscape of social media sites for children, client-side attacks would be the most prevalent and successful. This means that targeting the children using these sites would be more fruitful than targeting the sites themselves.

Client-side attacks would be the best option for attackers, with the abundance of cross site scripting within the sites. Combined with the data gained from file metadata available, unvalidated redirects and forwards, knowledge of internal information such as file share paths, vulnerable software versions, usernames, and other data, such attacks are estimated to be very successful if performed. In the event that a user's password is brute forced, the success rate of client-side attacks by a user's "friends" is elevated considerably.

"Targeting the children using these sites would be more fruitful than targeting the sites themselves."

On one site, SecureState was able to link multiple technical vulnerabilities with social engineering to perform proof of concept account hijacking. This was done in the following steps:

1. Discover "Mass Emailer" administrative tool via Web directory brute force attack
2. Utilize "Mass Emailer" to send test emails to SecureState account discovering cross site scripting via the subject line on the "inbox" page
3. Use "Mass Emailer" to change sender's ID so email appears from "Admin". Insert customized HTML image tag with CSS into subject field. The image tag included a source of http://<SecureState controlled public IP> + JavaScript document.cookie. When the victim goes to their "inbox," their browser attempts to automatically

load an external image from SecureState's IP address, with a path of the cookie value. The Web server logs contain the (failed) image request with the victim's cookie values, including a custom session ID.

4. Login to affected site, supplying stolen cookie value to hop into user's session and hijack account
5. Once logged in as the victim, any number of attacks could further an attacker's access

This was a simple proof of concept scenario provided for demonstrative purposes for the site operators. Other more sophisticated attacks would prove to be more fruitful to an actual attacker such as client-side browser exploits launched via JavaScript.

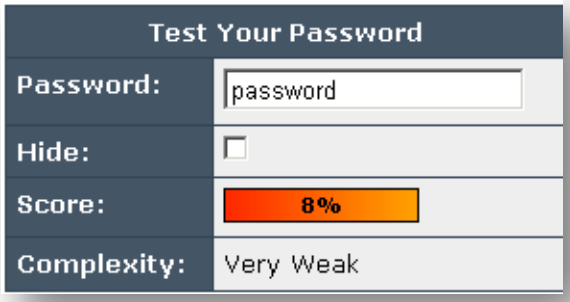
Defenses/Prevention

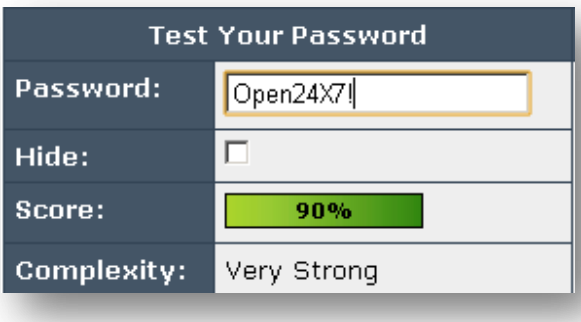
Some sites attempted to educate users about the dangers of online social media sites including what to do in such cases as a stranger or unknown person acting in an inappropriate manner, requests for personal information are made, etc. Sometimes a tutorial or test must be passed in order to enable "dangerous" features of sites such as chat rooms. While these are a good step toward prevention of online social exploitation, technical vulnerabilities still were numerous and widely available.

Most sites allowed for user enumeration in some fashion. Unique identifiers used to login to sites should be different from public "screen names" so that a login ID is not easily known. For sites that utilize only a username and password, one half of the authentication mechanism is already compromised due to public look-up of user data.

Site operators concerned about the age of the site's users possibly could utilize parent/legal guardian consent via credit card authorization during registration on their site. This could be done to enable potentially "dangerous" features of the site as well, since such a process might deter potential new users from coming to the site.

Very few of the sites surveyed provided instructions for creating a strong password. If a simple "password complexity meter" were to be displayed, such as seen here, this visual representation of the user's password likely would pique the interest of young users causing them to give more attention to password creation.





To appeal to the nature of younger users, site operators should attempt to make security features "fun." An example of this may be the implementation of a personalized "security image" and/or "phrase." Some online banking sites utilize such features. A multi-step login process usually is encountered where a user ID is provided, and subsequently an image or phrase that was chosen by the user upon account creation then is displayed. If the image or phrase displayed does not match the one the user had chosen, it is a clue that something is incorrect and that continuing to provide a password

should not be done. Perhaps providing a username, then requiring the child to choose their "correct" image out of a line up could be step 2 of the login process before entering a password. This would offer the child a sense of customization and provide visual stimulation, while simultaneously accomplishing a security goal.

"Forgot Password" functionality should not divulge whether or not a valid account was specified. Instead, a generic message can be specified informing the user that the new password or password reset instructions were emailed to the email address used when the account was created. Passwords never should be emailed plaintext, but reset. Since changing (let alone remembering) passwords may be difficult for a young user, a proposed visual password again could be used. Perhaps instead of a text password, a series of image clicks or answered questions would be more in order to keep it secure, yet fun for the user.

Lessons Learned

An information security assessment of social media sites for children identified many security vulnerabilities that could allow an attacker to target children. Pedophilia was the top motive of attackers targeting children from the observations made during this research.

Among the security vulnerabilities identified:

- The vast majority of the sites failed to utilize secure channels for communication (HTTPS)
- The majority of password policies did not reflect industry best practices.
- Sites' ineffective chat room language filters allowed pedophiles to bypass them with trivial message content modifications.

The entry requirements for social media sites vary greatly, and are easy for children to circumvent. Children's curious nature makes them vulnerable targets for an attacker because of their willingness to click on unfamiliar buttons, links, etc.

Suggestions for making social media sites safer for children include the use of parent/legal guardian consent via credit card authorization, and the use of "fun" security features so children will be encouraged to create secure passwords. Further research is suggested around sociological impacts of security requirements on children as end users to such social media sites.

References and Related Links:

http://www.socialmediamom.com/2008/02/february-roundu.html

http://mostpopularwebsites.net/

http://www.alexa.com/topsites

http://lynnepope.net/twitter-xss-attacks

http://blog.ivanristic.com/2009/11/not-just-csrf-ssl-authentication-gap-used-for-credentials-theft.html

http://www.securegoose.org/2009/11/tls-renegotiation-vulnerability-cve.html

http://www.ftc.gov/ogc/coppa1.htm

http://www.cgisecurity.com/whitehat-mirror/WH-WhitePaper_XST_ebook.pdf

http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

http://erlend.oftedal.no/blog/?blogid=107

http://www.passwordmeter.com/