Global Information Assurance Certification Paper

The Zkey Exploit: The Capability of Malicious JavaScript Code

By David Rothermel
August 28, 2000

On August 14, a hacker who calls himself "Blue Adept" completed an exploit of the Zkey.com information storage portal. The exploit compromised the portal's security by using malicious JavaScript code to capture usernames and passwords of Zkey email users. This particular exploit is a good example of just one variety of a "cross-site" scripting attack, where the vulnerability is based on the violation of trust resulting from a malicious script or code running within the victim's site or browser.

The Zkey portal provides its approx. 300,000 users with free SSL-protected file storage space and web-based email services. The exploit allowed a malicious Zkey account user to embed malicious JavaScript code in an email that could then be sent to another Zkey users email account. When an unsuspecting Zkey user read the trojanized email, the embedded JavaScript code took complete control of the user-interface. A message box was displayed indicating the session connection had expired, and using a copied Zkey login dialog box, forced the user to re-login. Upon doing so, the code compromised the username and password of the victim and sent it to the malicious users account on another server. This particular scripting exploit could just have easily been modified to attack other Zkey portal services available to the victim user. One important note is that the exploit required the user to respond to the login prompt, and functionality was somewhat browser specific.

Through a bit of programming and testing with HTML tags, Blue Adept found a way to embed all of his malicious code inside a form's <textarea> tag. He also used a transparent gif that covered the entire message and used an onMouseOver command to trigger the embedded code. He then proceeded to create code that would control the Zkey GUI. He experimented with how much script he could use in the SSL session without getting a browser warning that the message contained insecure elements or content. He decided to use a spoofed re-login approach, so upon the user clicking any link in any frame, the script runs the re-login procedure. A login box is displayed with a message indicating "you timed out of your session, please re-login". When the user does so, their username/password data is forwarded to a Perl script that enters the information on a database on the hackers' server instead of the Zkey server. To maintain the appearance of normality, the Perl script sends the user back to the Zkey server and logging them in to continue their session. [2]

For full details on Blue Adept's code testing efforts, and to view the JavaScript source code that he used, explore the following URL's:

http://www.because-we-can.com/zkey/notes.htm
http://www.because-we-can.com/zkey/trojan.txt

To grasp the seriousness of the exploit, consider the following ramifications. "Once a malicious user knows the username/password of the victim's Zkey account, they can assume full control of the account, including the ability to:

- Download files from the victim's z-drive.
- Delete/replace files from the z-drive.
- Access/alter the victim's contact information.
- Access/alter the victim's calendar/scheduling information.
- Change the victim's username/password, locking them from their account.
- Access any shared drive z-drives from secondary accounts.
- Read/delete the victim's Zkey-email or send Zkey-email in the victim's name.
- Access email from any secondary email accounts configured for mail checking."[1]

Upon successfully testing his completed exploit, Blue Adept, apparently a "white hat" hacker did not use the exploit for his own gain, but instead immediately tried to notify the Zkey customer support and webmaster of the vulnerability and indicated that the exploit would be made public. He recommended to Zkey that they should warn their users and fix the problem. Three days later he still had not received a reply from Zkey and went public with the vulnerability. When contacted by a web-based news service regarding the exploit, the company president of Zkey acknowledged the existence of the problem and indicated that it had been resolved. He also downplayed the exploit's severity and capability.

The hacker, Blue Adept notes that "Zkey was an interesting exploit for two reasons. First, the site is SSL'd – served off a secure socket layer (medium grade encryption key RC4-40) which encrypts data in transit and also restricts the kinds of cross-site scripting techniques that can be used without alerting the user to suspicious activity. Secondly, the email service itself has filters in place to strip out malicious code from the body of email messages. But both security measures were surmountable."[2]

This exploit demonstrates that cross-site scripting attacks cannot be prevented by the use of SSL servers alone since they don't validate the legitimacy of the data being transmitted. Likewise, "malicious code that attempts to connect to a non-SSL URL may generate warning messages about the insecure connection, but the attacker can circumvent this warning simply by running an SSL-capable web server." [3] This type of attack is not vendor-specific, every web server and browser is effected.

There are two deterrents available to end-users to reduce their risks to such attacks. First, users should disable all JavaScript and active content functionality, such as ActiveX, in their browsers. In certain instances, this may disable desired functionality that the user requires. This will not eliminate the risk, and be aware that even if a browser does not support scripting, an attacker could still alter the appearance, behavior, or operation of a site's web page. Second, users should be discrete about what links they use to visit a web site, alternatively they should type the address into their browser.

Developers on the other hand should apply any relevant vendor patches, check that their pages do not contain undesired HTML tags and possibly filter data for specific characters to detect malicious code. For more specific detail on these steps refer to: http://www.cert.org/tech_tips/malicious_code_mitigation.html .

Blue Adept turned out to be a good guy, just imagine if he had not.

References:

[1] Unknown. "The Zkey Exploit And How To Protect Yourself".
URL: http://www.because-we-can.com/zkey (8/17/2000).

[2] Adept, Blue. "Blue Adept's Notes On Zkey Exploit".
URL: http://www.because-we-can.com/zkey/notes.htm (8/17/2000).

Delio, Michelle. "Scary Hole Found at Zkey".
URL: http://www.wired.com/news/technology/0,1282,38292,00.html (8/18/2000).

Unknown.
URL: http://www.because-we-can.com/zkey/trojan.txt (8/17/2000).

Unknown. "Zkey Security Hole Compromises User Accounts".
URL:
http://www.securiteam.com/securitynews/Zkey_security_hole_compromises_user_accounts.html (8/21/2000).

[3] Unknown. "CERT Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests". Last Revised: February 3, 2000.
URL: http://www.cert.org/advisories/CA-2000-02.html (8/28/2000).

Unknown. "Cross-Site Scripting Security Exposure Executive Summary". (2/2/2000).
URL: http://www.microsoft.com/technet/security/ExSumCS.asp (8/28/2000).