Alexander Smetana

Lab02 - CIS311

03/03/2023

CAPTCHA

CAPTCHA is an acronym that stands for, "**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part." A CAPTCHA is defined as a "program that protects websites against bots by generating and grading tests that humans can pass but current computer programs cannot" (Captcha.net). In other words, it is a web form that asks users to enter text from a distorted image of randomly generated text. A CAPTCHAs main goal is to solve the problem of bots.

The process of a CAPTCHA is very simple in concept, however very difficult to execute. The main goal being, make an authentication process that humans can solve while keeping bots out. For humans to be able to pass a CAPTCHA test, factors of culture and age need to be considered for the test to be passable by any human. It needs to be a test that a computer wouldn't be able to pass, however the computer still needs to be able to grade answers too. The idea in theory is that robots cannot decipher a distorted image and read the text, with a human is able to do so. Humans can read from many different types of angles while computers were very bad at this.

The CATPCHA process at one time, was secure and very protected. However, over time computers have evolved and gotten much smarter, so better solutions have arrived. In the early 2000's CAPTCHAs worked very well. Unfortunately, as time progressed computers have gotten much smarter. No longer was distorted text being able to stop computers. In 2005, this led to the invention of reCAPTCHA. It used two words, one generated randomly, one from an article, to verify a human. In 2014, ReCAPTCHA V2 was introduced, which asked humans to identify images instead of text. Finally, RECAPTCHA V3 which would verify based on behavior.

That leads to the question, are there better solutions to solve this issue? Currently, it appears that RECAPTCHA V3 is the best solution that someone can get. Now, websites are basically tracking you at all moments in order to tell whether someone is a bot or not. Computers have evolved since the start of CAPTCHA, allowing for them to break CAPTCHAS V1 and V2 very consistently. Although not all methods of prevention are full proof, CAPTCHAS has evolved since its origin.

Tweet Deck Worm

The following code below is the Tweet Deck Worm, a XXS (cross-site scripting) attack that was discovered in 2014. It was started by a 19-year-old computer programmer named Firo in Austria. He was a script kiddie that managed to get himself into more trouble than he expected. His worm managed to spread to 82,138 Twitter users, with the BBC retweeting the tweet to 10.1 million accounts (Bill Napier).

XXS is a flaw in a website that allows for the injection of client-side script code by unauthorized users. The vulnerability in the Tweet Deck allowed for the worm to spread fast and far. Anyone who viewed the tweet allowed it to spread and get automatically retweeted. The tweet worked by taking advantage of the Tweet deck bug while executing the JavaScript code inside of the browser of Tweet deck users (Naked Security).

<script class="pigeon">$('.pigeon').parents().eq(1).find('a').eq(1).click();$('[data-action=retweet]').click();alert('LMAO GOTTEM')</script>

Breaking down the code (Acunetix Blog):

1. <script class=”xss”>
   - Opens an inline script tag. **xss** is the script tag.

2. $(‘.xss’).parents().eq(1).find(‘a’).eq(1).click()
   - **$(‘.xss’)** – Indicates JQuery to select the html tag named xxs defined earlier
   - **$(‘.xss’).parents().eq(1)** - selects the second parent of the tag. parents(). Return the set of parents and eq(1) means to select element one within that set. In other words, it will. Select the tweet container.
   - **$(‘.xss’).parents().eq(1).find(‘a’).eq(1).click();** - Searches all links from the container and selects the second link. This is the retweet link.

3. $(‘[data-action=retweet]’).click();
   - This code retweets the link and clicks on the confirmation link.

4. alert(‘XSS in Tweetdeck’)
   - Creates an alert tweeting ‘XSS Tweet deck” Warning that there is a vulnerability.

Fortunately, the Tweet Deck Worm was patched before it could spread even further, but some steps could’ve been taken to prevent the worm from occurring. Only users who had the

installed Tweet Deck Web app for google chrome were primarily affected. The vulnerability renders the tweet as code in the in the browser, allowing for XXS to occur. In order to prevent this attack, Twitter immediately took down Tweet Deck and patched out the bug. To prevent bugs like this in the future, some best practices include filtering your inputs with a whitelist of allowed characters, encoded data on output, use appropriate response headers, and content security policy to prevent XXS vulnerabilities (Port Swigger).

Citations

billatnapier, Written by. “Tweetdeck Hack: How a 19 Year-Old Playing with Hearts Broke the Internet.” *Billatnapier*, 12 June 2014, https://billatnapier.wordpress.com/2014/06/11/tweetdeck-hack-the-lesson-of-poor-software-design-and-testing/.

Calin, Bogdan. “The Tweetdeck Worm: How It Worked.” *Acunetix*, 4 Oct. 2017, https://www.acunetix.com/blog/articles/tweetdeck-worm-worked/.

“CAPTCHA: Telling Humans and Computers Apart Automatically.” *The Official CAPTCHA Site*, http://www.captcha.net/.

Dan Goodin - Jun 11, 2014 8:21 pm UTC, and ColinABQArs Praefectuset Subscriptor jump to post. “Powerful Worm on Twitter Unleashes Torrent of out-of-Control Tweets.” *Ars Technica*, 11 June 2014, https://arstechnica.com/information-technology/2014/06/powerful-worm-on-twitter-unleashes-torrent-of-out-of-control-tweets/.

“What Is Cross-Site Scripting (XSS) and How to Prevent It?: Web Security Academy.” *What Is Cross-Site Scripting (XSS) and How to Prevent It? | Web Security Academy*, https://portswigger.net/web-security/cross-site-scripting.

Wisniewski, Chester, et al. “Twitter Jumps to Block XSS Worm in Tweetdeck.” *Naked Security*, 11 June 2014, https://nakedsecurity.sophos.com/2014/06/11/twitter-jumps-to-block-xss-worm-in-tweetdeck/.

YouTube, YouTube, 14 May 2021, https://www.youtube.com/watch?v=lUTvB1O8eEg. Accessed 5 Mar. 2023.