In the world today, it is hard to imagine how interconnected we are. Furthermore, the availability of information on a massive scale is absurd. Right now, if you wanted to know the diameter of Mars, I could tell you in a few seconds. The internet has grown at a substantial rate in the past decade, and it continues to grow today. It is key that information security grows at the same rate, to prevent intrusions, such as the large scale cyber attacks we hear about all too often. Public awareness and outreach is an excellent way to increase staying safe online and improving the security of information on the internet.

Back in 1990, Tim Berners Lee invented the Hypertext Markup Language, better known as HTML, the language of the web. HTML was found to be an excellent universal method of creating webpages. Five years later, Javascript was invented, and allowed users to interact with webpages. However, it was not entirely secure, and many of its downfalls still exist today.

As a result of these flaws, the Cross-Site-Scripting (XSS) attack was born. Cross-Site-Scripting Attacks can occur when a webpage loads something that it thinks it should, but really shouldn’t. For example: say I have a website with a comment thread. Whenever a user posts a comment, it will be shown to all of the other users who go to my website. When I post a comment, a piece of HTML gets placed on everyone else’s webpage, which ultimately becomes the text you see on the screen. Normally, this would be okay for most users. However, HTML has a flaw: the scripting tag. Whenever HTML sees "<script>*</script>," it will run whatever is within the two tags (the asterisk). This can be exploited by attackers. On my website, if an attacker wrote a comment with a scripting tag in it, when the comment reached another users computer, HTML would tell JavaScript to run whatever was in those tags. This type of cross-site-scripting attack is known as a Stored XSS attack. However, there is no need to be afraid of these, as most websites “sanitize” their inputs. This means that if a user types HTML or scripts into a textbox, it will be displayed exactly as is on another user's computer, and will therefore not run as JavaScript.

Another type of XSS attack is a reflective attack. Reflective XSS attacks are very common, and can also be easily prevented. As an example of how the attack works, visit Google.com or Amazon.com, or almost any website with a search box. Type in the box anything, and hit enter. Now, look at the URL of the website at the top of the page. If you look around, you will likely find the words you typed in the search box in your URL. The reflective XSS attack becomes possible because of this. When your browser sees these words in the URL, they appear on the page. For example, if you search for something on Amazon, when the results load, you will see your request in the URL and in the search box on the page. Normally, this is would be safe. However, if an attacker types a script tag (like in stored attacks) into the part of the URL that is presented on the page, when the page loads, HTML will tell JavaScript to execute the script. For example, say I created a website that has the URL example.com. When I type (ie. “test”) something into the search box on the website, the URL changes to example.com/search=test. Now, when the results page loads, the page will take “test” out of the URL and put it on different parts of the page, like on a “you searched for:” box. This becomes dangerous when attackers begin handing out URLs that have these scripts in their URL. For example, if you received an email with a URL laced with a script, when you clicked on it, the page would then execute the script within the URL. This could send private information from your computer to attackers, all due to the way the website loads. However, most websites will also sanitize their information from the URL, in order to prevent scripts from loading once they load in your browser.

Across the Internet, there are many ways in which an attacker will attempt to achieve their goals. Cross-Site-Scripting attacks represent a fairly easy way in which an attacker can wreak havoc on websites and their users. Unfortunately, it is a step that can be easily overlooked when creating a website. However, most websites safely sanitize their users inputs, therefore reducing the risk of an attack. However, these attacks should never go unnoticed or missed. If you are a user, when someone sends you a link, be sure that you trust the person who gave it to you. If you are developing websites,
be sure to sanitise inputs! On something as large and public as the internet, vulnerabilities can be found everywhere. Ask yourself, am I being safe online?

Thank you to the GenCyber and its sponsorship by the Institute for Security, Technology, and Society at Dartmouth College, as well as their Information Technology Services for providing this opportunity for outreach.

Written by Adam Frank, a Junior at Windham High School.