**Discussion Question: Software Weaknesses**

While many known security breaches can be attributed to user error, poorly developed software is often at fault. For many years, the Mitre organization has kept lists of the known software weaknesses out in the world, and occasionally publishes a Top 25 Most Dangerous Software List. Take a look at the [Common Weakness Enumeration](https://cwe.mitre.org/top25/archive/2022/2022_cwe_top25.html) site.

Select one of the weaknesses (try and select one that other students have not chosen), and provide a briefing for the class.

In your brief, include the following:

* 1. ~~Date the CWE was submitted.~~
  2. A description, in your own words.
  3. How it works.
  4. Which concept(s) of security does it violate? Identification? Authentication? Authorization? Availability? How? Again, in your own words.
  5. Any mitigations recommended? If so, provide a brief description of one.

***Before you submit your thread, put your name in the subject line.***

Security breaches are sometimes due to user error, but the bigger culprit is usually poorly developed software. Software may appear to function well, but if it is not backed with secure coding, all the hard work put towards the project could be ruined, and with it may go the trust of clients and other stakeholders since it exposes them too. Every year, the Mitre organization tracks known software weaknesses and compiles them into the Top 25 Most Dangerous Software List. This discussion will focus on list #16 CWE-862: Missing Authorization. CWE-862 was submitted on 5-24-2011.

CWE-862 describes when a program foregoes authorization when a user executes an action or accesses records (Common Weakness Enumeration, 2011). This means that now an unauthorized user has access to privileged information and the ability to alter information. This data could be accessed due to a lack of restrictions and through programs not adequately secured. Further privilege could also be granted if records are altered in favor of the attacker.

This is a direct violation of the authorization security concept. Since no authorization is checked when a user is executing an action or records in CWE-862, it’s clear that proper authorization techniques have not been integrated into the system. This could also branch into the identification and authentication concepts. Someone could use another person’s computer after logging in to access a system and/or perform an action they are not supposed to.

There are some recommended mitigations for this software weakness. One is implementing least privilege is one way to combat this issue. Then, users who do not need certain software functionalities have limited access, while additional access is reserved only for privileged users. A good way to accomplish this is through roles.

**Reference**

Common Weakness Enumeration. (2011). *CWE - CWE-862: Missing Authorization (4.2)*. Cwe.mitre.org. https://cwe.mitre.org/data/definitions/862.html

**Assignment Requirements and Grading:**

* An initial post of the diagram is due by **Thursday, 11:59 p.m., CT**.
* Submit your post by clicking on the **Assignment Link** above, then **Create Thread**. You must create a thread in order to view your peers' posts.
* A minimum of three (3) responses, **to the original threads of other students**, of 100-200 words each are due by **Sunday, 11:59 p.m., CT**.
* To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf).

**(50 points)**

Hello, Nardos! You did an excellent job of touching on the common weakness enumeration you chose to focus on. Reading through the top 25 was intriguing; some were what I expected, but others I was more surprised by. Throughout the journey, improper input validation has been an important topic that has been talked about, so I was not surprised to see it on the list. Attackers are likely to try to corrupt a system by entering malicious input into a system. I like how you mentioned that the CWE website touches on users entering invalid input, which could also have bad results. Not every security breach is attributed to attackers; some people may not realize the harm they could accidentally cause when a system does not have proper input validation.

Hi, Arley! I thought you did a thorough job of explaining CWE-862: Missing Authorization. This is also the weakness I discussed in my post. Were you surprised by the placement of any of the weaknesses? I somewhat expected missing authorization to be higher on the list than #16. Was this something that you also felt? I agree that role-based access control is an excellent way to mitigate this problem. Every company I have worked for has used role-based access for their software. Although it can feel annoying when I cannot access all the information I need, it makes much more sense after learning about security practices.

Hey, Nima! Your post for this week accurately covers the questions related to your chosen common weakness enumeration. I was pretty close to covering this one but instead opted for #16, Missing Authorization, so I am glad to see that someone else chose it! I found it particularly intriguing that so many weaknesses were entered in the early 2000s and still remain there today. I also think that additional security features like tokenization and the use of digital signatures could also combat this weakness. A software service that could be good for this is Google Authenticator. Since the codes time out quickly, it makes it increasingly difficult for someone else to gain access.