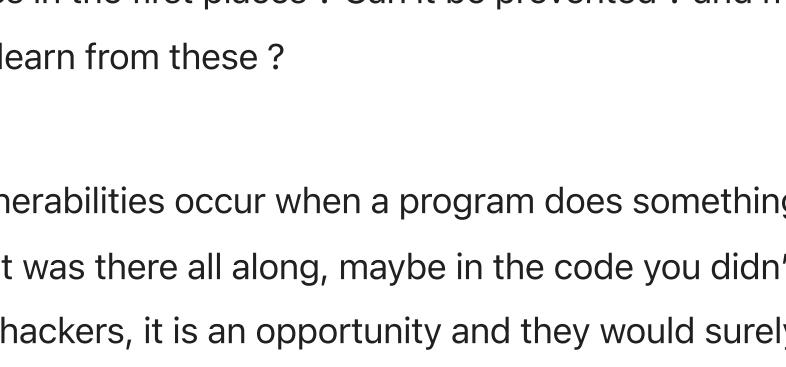
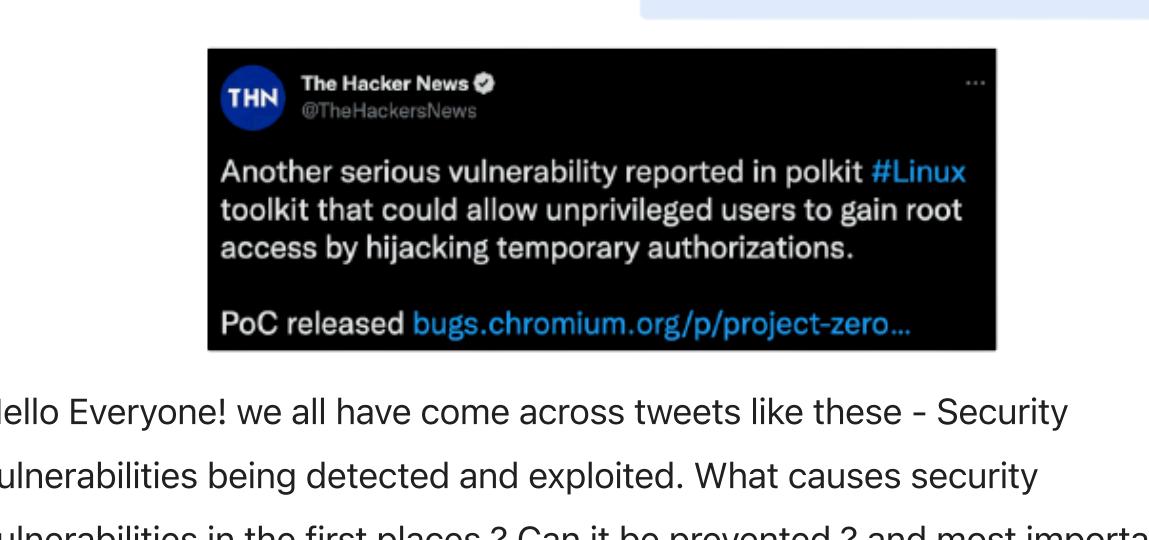


Author: Vedang Joshi
Published: October 26, 2022 · 5 min read

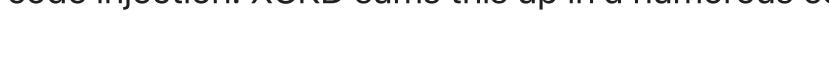
Florian Roth

What people seem to miss:
The #Log4Shell vulnerability isn't just a RCE attack.



of it. Here, I would discuss some best practices that avoid common kinds of security vulnerabilities.

the bad actors. In the case of python, the use of eval statements is highly discouraged. These types of attacks are commonly addressed as injection attacks. An attack could inject SQL statement, HTML, or Javascript which if executed on the server side gives the attackers full control. [Log4j](#) was an example of code injection. XCKD sums this up in a humorous comic.

A comic strip from XKCD by Randall Munroe. It consists of four panels. The first panel shows a character with a speech bubble: "HI, THIS IS". The second panel shows another character with a speech bubble: "OH, DEAR - DID HE". The third panel shows a third character with a speech bubble: "DID YOU REALLY". The fourth panel shows a fourth character with a speech bubble: "WELL, WE'VE LOST THIS". The comic illustrates how complex and convoluted code can become when dealing with user input.

A simple black and white line drawing of a stick figure with short hair. A curved line above the head indicates a speech bubble, but no text is present.

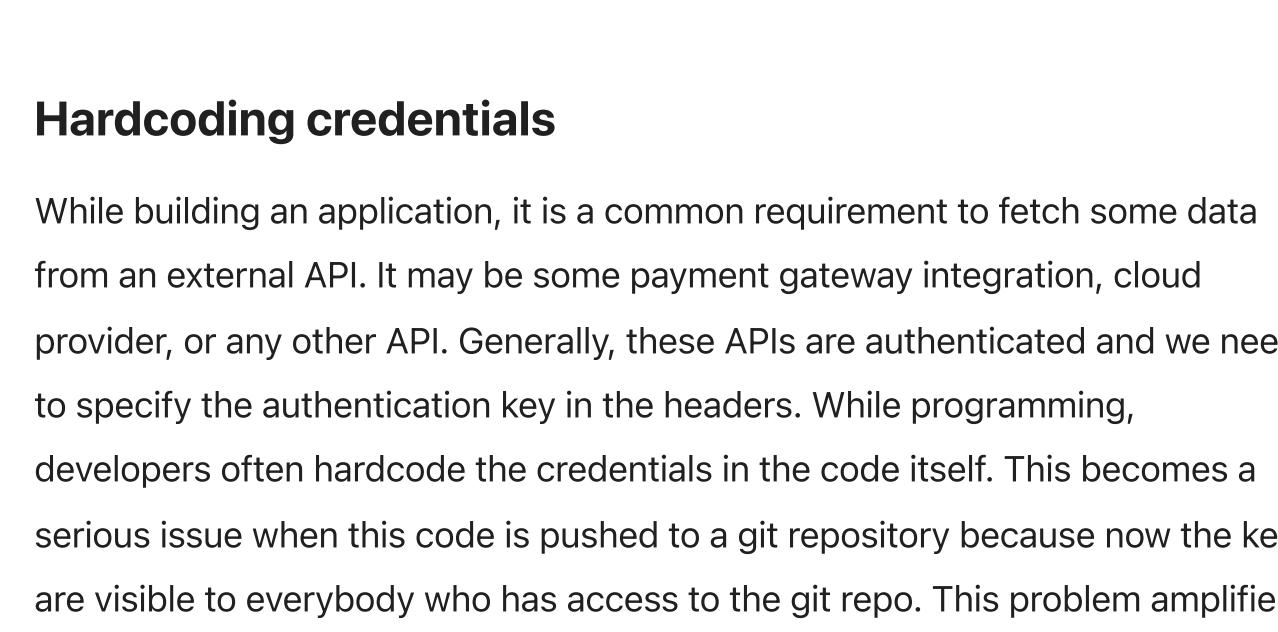
ANSWER

Developers need to make sure that there is no room for user input to get executed. Here user input can be input from the end user or even from another developer.

SQL Injection can be avoided by using an ORM instead of writing raw queries. Mostly all popular ORMs come with built in sanitization features.

Major Web frameworks like Django and Flask also come with their sanitization functions. This prevents potentially malicious HTML inputs.

Other than this, Static application security testing (SAST) tools can also detect



Not only this practice is a blunder in security but it hinders the developer's productivity too. Once the code becomes a part of a container image and is pushed to production, it becomes difficult to track and replace the credential.

It is quite easy to avoid this

1. Passing the credentials as env variables solves this problem. Developers can create a .env file if the number of variables is large. This .env file should never be checked into version control.

2. Implementing secret scanning
credentials are hardcoded here

can be added as a CI stage as well.

Parsing

Developers often need to deal with XML and YAML files. The first step in using any of these files is to parse them. There are a few common attacks through these parsers.

- XML parsing

Python XML standard library warns the users that using the etree, DOM, xmlrpclib is not secure. An attacker can create a DOS-style attack. This attack references a piece of data in the same file and doing it over and over increases the RAM usage exponentially. Thus crashing the server.

An attacker can also take advantage of an XML parser trying to fetch and

The us

Using the `yaml.load` method can be used to make system calls, thus leaving you wide open to attack.

The use of `yaml.safe_load` is recommended as a fix.

- # Updating dependencies

PyPI for your project, v

installing.