Protected Python: It’s time we had “the talk” about cyber security

JAMES MERTZ

About Me

Software Quality Assurance Engineer at NASA's Jet Propulsion Laboratory

- Europa Clipper Project
- Cybersecurity Assurance Project

Author for Real Python

Teacher at Pasadena City College

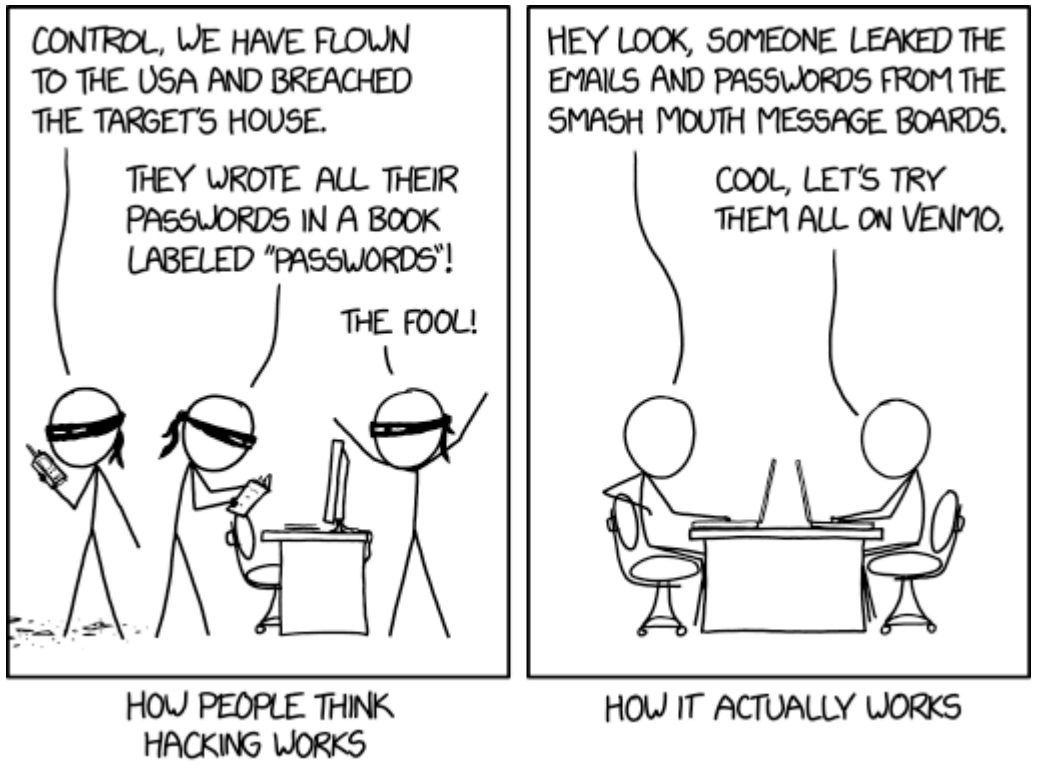
- Intro to Programming (Python)

Working & Teaching Python ~10 years

JPL

Jet Propulsion Laboratory
California Institute of Technology

What Is Cyber Security?

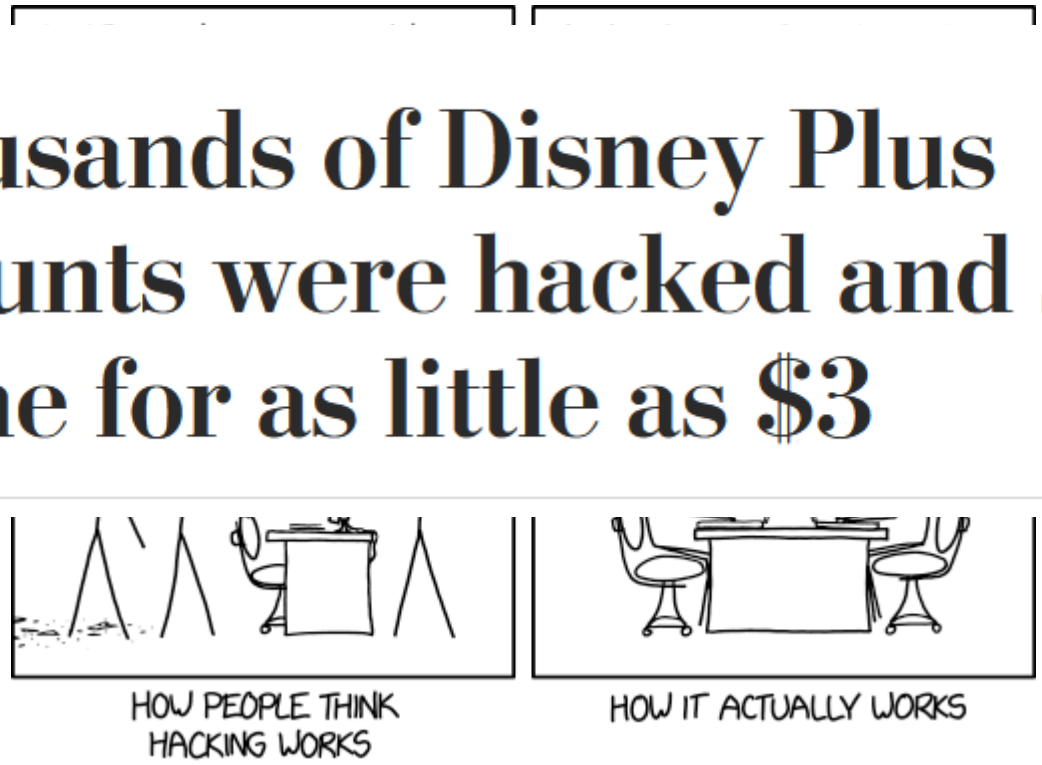


https://xkcd.com/2176/

What Is Cyber Security?

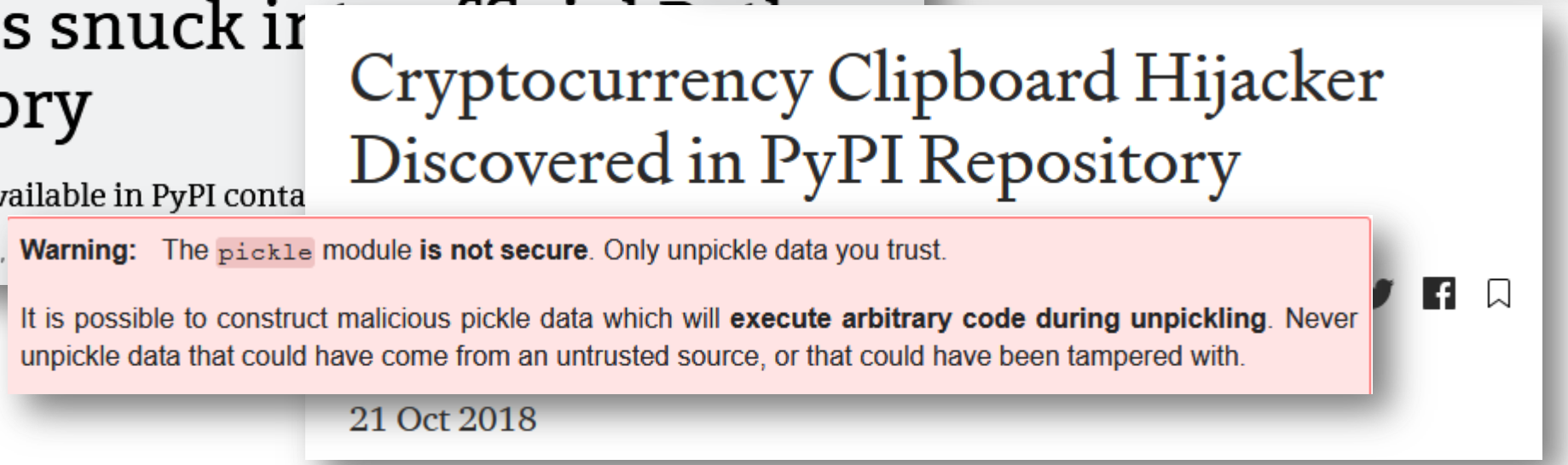
Business

Thousands of Disney Plus accounts were hacked and sold online for as little as $3



https://www.washingtonpost.com/business/2019/11/19/thousands-disney-accounts-were-hacked-sold-online-little/

Why Should Python Developers and Users Care?



Cybersecurity is…

Complicated

Messy

Faked

Awkward

Time Consuming

Cybersecurity is…

Nobody wants to talk about it…

But

Everybody Says They're "Doing It"

The first step to having good cybersecurity…

Honest & Open Communication

Formal Definition of Cybersecurity

"Measures and controls that ensure confidentiality, integrity, and availability of information system assets including hardware, software, firmware, and information being processed, stored, and communicated." (Stallings 2)

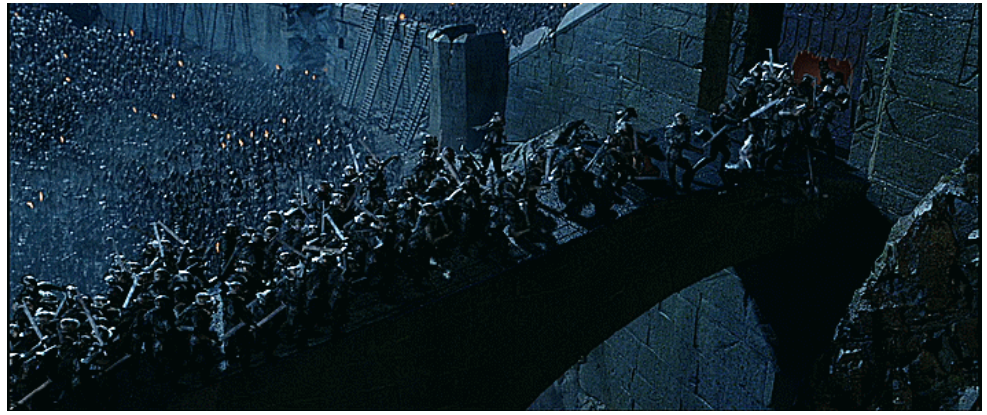
CIA Triad:

- Confidentiality
- Integrity
- Availability

Defense Strategies

Tower Defense (Old Strategy)

- Protect everything that is assaulting you from the outside
- Not paying attention to the stuff on the inside



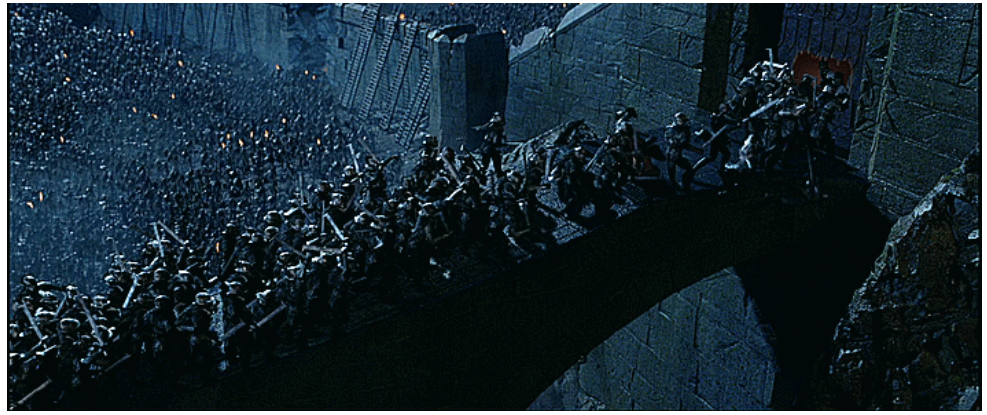
Defense Strategies

Tower Defense (Old Strategy)

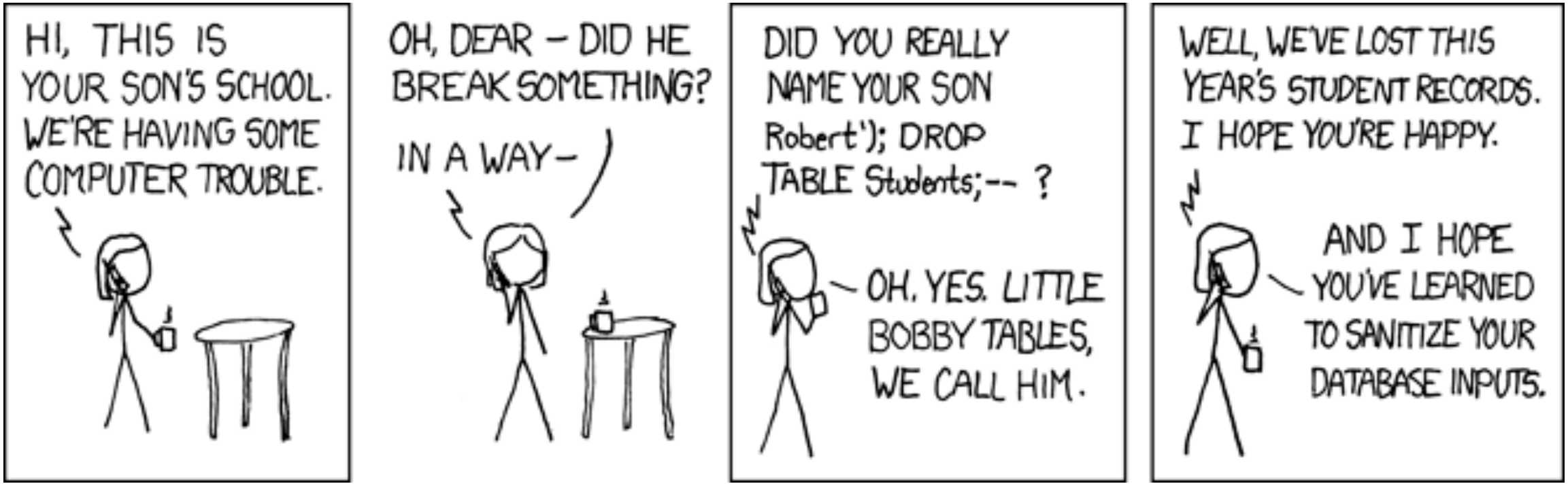
- Protect everything that is assaulting you from the outside
- Not paying attention to the stuff on the inside

Defense in Depth

- Tower Defense with a mixture of Clue



Common Attack Vectors and How To Protect



https://www.xkcd.com/327/

Little Bobby Tables

Web Applications

django

Untrusted Data

- If the data ever leaves your hands, it's "Untrusted"
- SQL Injection
- File/Data uploads
- Messaging

Cross-Site Scripting (XSS)

- Injection of malicious code into the users browser

Cross-Site Request Forgery (CSRF)

- "Tricking" the user into unauthorized access of your site

Sanitize your data

Use crypto safe libraries (Don't create your own!)

Use built-in tools and recommendations for web frameworks

- https://docs.djangoproject.com/en/3.0/topics/security/
- https://flask.palletsprojects.com/en/1.1.x/security/?highlight=security

Use SSL whenever possible

Developer Environments

Untrusted Data

- Loading
- Executing

Malicious 3rd Party Packages (Typo-squatting)

Avoid using **`pickle`** or **`yaml`** for de-serializing data

- Use json instead
- Use **`defusedxml`** for xml parsing to avoid XML bombs

Avoid using **`exec`** or **`subprocess.call( shell=True)`** for execution of code/outside process calls

Always check the package name you're about to install

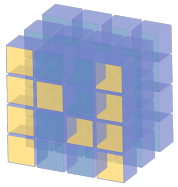
- When in doubt, look at the source!

Consider using Anaconda and their official package repo for package management

PyPi is looking at securing packages better as well:

- https://github.com/pypa/warehouse/milestone/16
- "Bringing Two Factor Authentication to PyPI" by William Woodruff

Code Base and Repositories



NumPy

GitHub

Data spillage

Outdated package dependencies

Improper 3rd Party Package Usage

- Did you know that `numpy` uses `pickle` in certain cases?

Using C extensions

Don't place keys, passwords, tokens, etc, into your repository

- Don't fall into the trap of *"I'll just .gitignore this file"*
- Use environment variables instead

Keep your `requirements.txt` (and dev env) up to date with 3rd party package updates

Read the documentation for security warnings

Pay special attention to C extensions and its exploits

- Buffer/stack overflows
- Use Static Analysis on your custom C extensions

Security Tools/Resources

[Bandit]
- Security Scanner with ability to create custom tests

[PyUp]
- Scans `requirements.txt` and makes sure packages are up to date

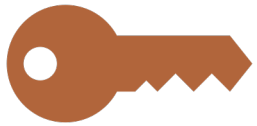
[Python Security PyCharm add-in]
- Security checks

[OWASP Top 10]
- A list of the top 10 security risks for applications

[Mitre's Common Vulnerability and Exposures]
- A collection of the most common vulnerabilities

[SANS Institute]
- Technical Training and Papers

Final Thoughts

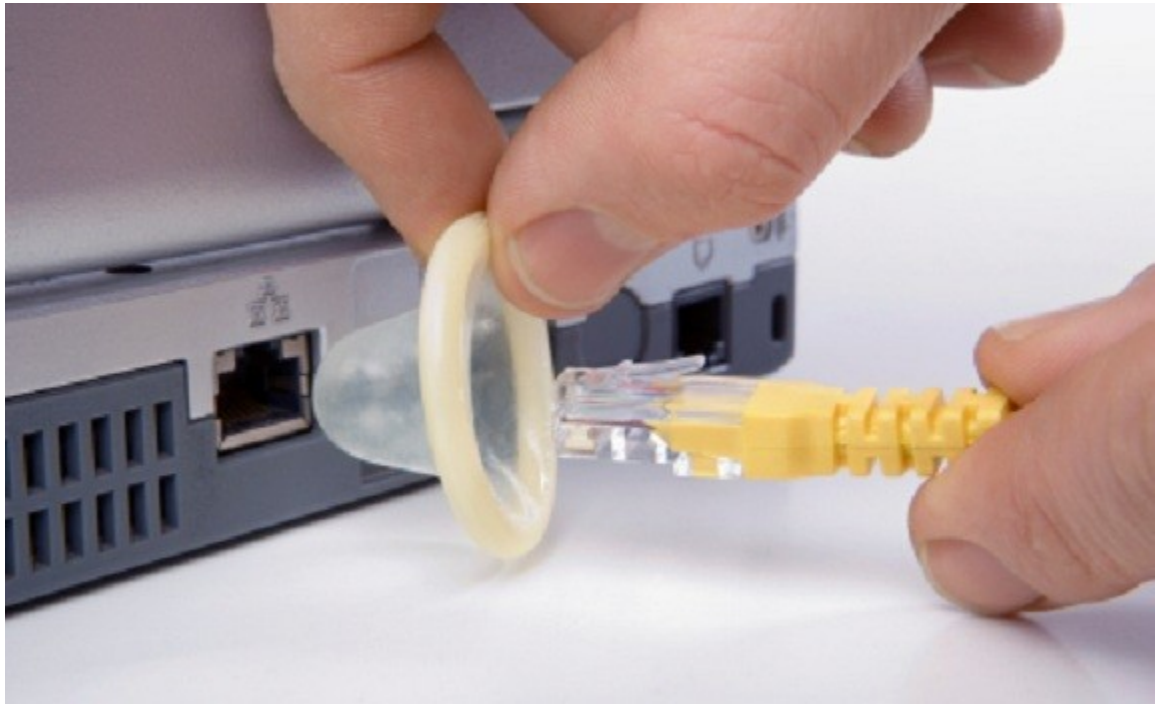


Awareness is key

Seek out help

And…

Always Use Protection



References

Stallings, William, Lawrie Brown. Computer Security, 4th Ed

Wikipedia. Information Security.

Digging for Security Bugs in Python by Travis McPeak

10 common security gotchas in Python and how to avoid them by Anthony Shaw

Advanced Security Topics, PyCon 2012 by Paul McMullin

Python Security Best Practices Cheat Sheet by synk