CSCI-UA.9480
Introduction to Computer Security

Session 3.1

Understanding and Preventing Vulnerabilities

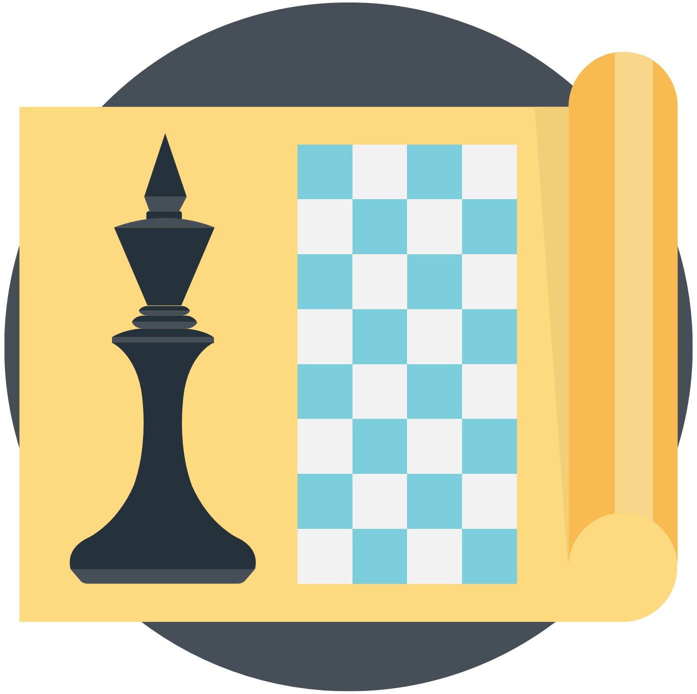
Prof. Nadim Kobeissi

What does it mean for software to be secure?

Let's consider a social network app.

- Pictures posted by a user can only be seen by that user's friends (*confidentiality*)
- A user can like any given post at most once (*integrity*)
- The service is operational more than 99.9% of the time on average (*availability*)

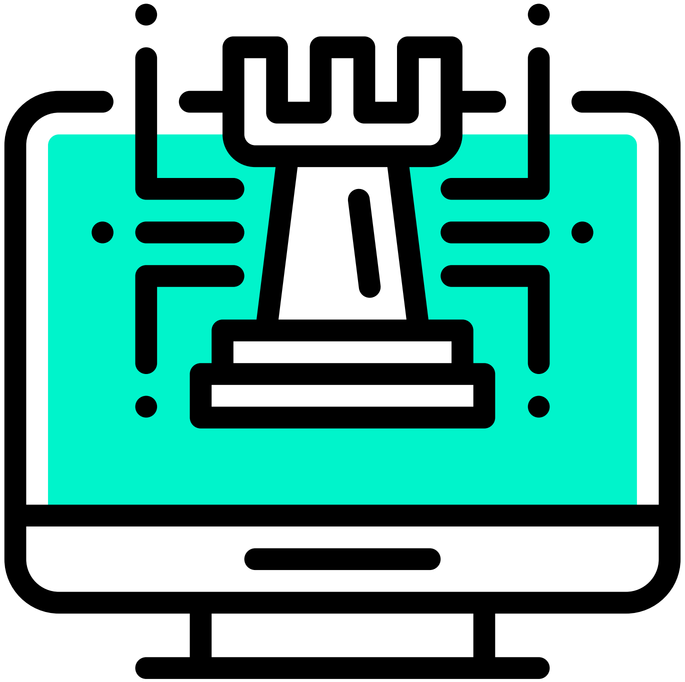
Sound familiar? Same words, but meaning is context dependent (cryptography vs. application security.)



What is a security failure?

The system can be coerced into a state in which it does not achieve its security goals.

- Can be due to a software programming error.
- May be due to a design error in the protocol specification.
- No error in the software at all, but rather user error.



Common Vulnerabilities and Exposures.

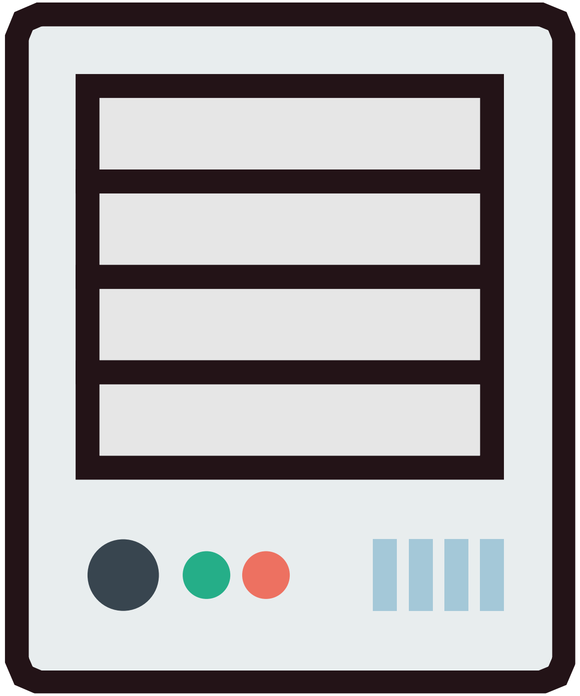
U.S. national repository of software vulnerabilities.

- Most bugs eventually obtain their own "CVE."
- Operating systems bugs, web application bugs, etc.

Diving into a CVE: CVE-2014-3205.

All Seagate BlackArmor NAS contain a hardcoded-password.

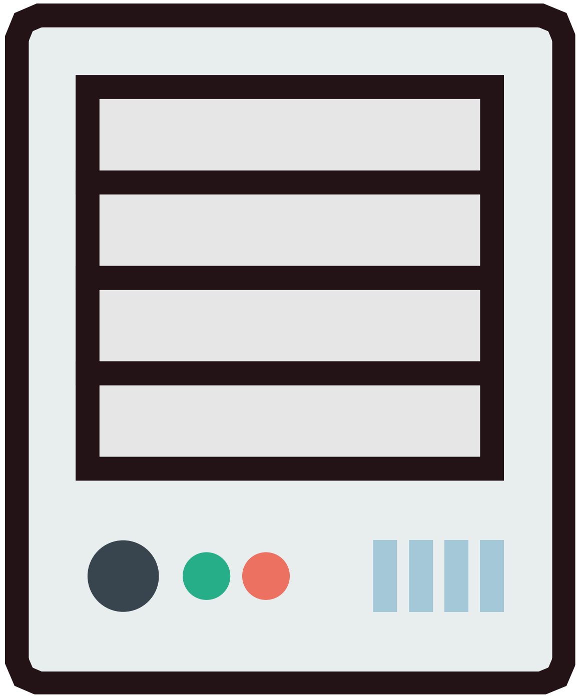
- Anyone could log in using the password "`!~@##$$%FREDESWWSED`"
- Followed by another separate CVE, CVE-2014-3206 which allowed anyone to execute arbitrary code by sending a HTTP request to a PHP file.



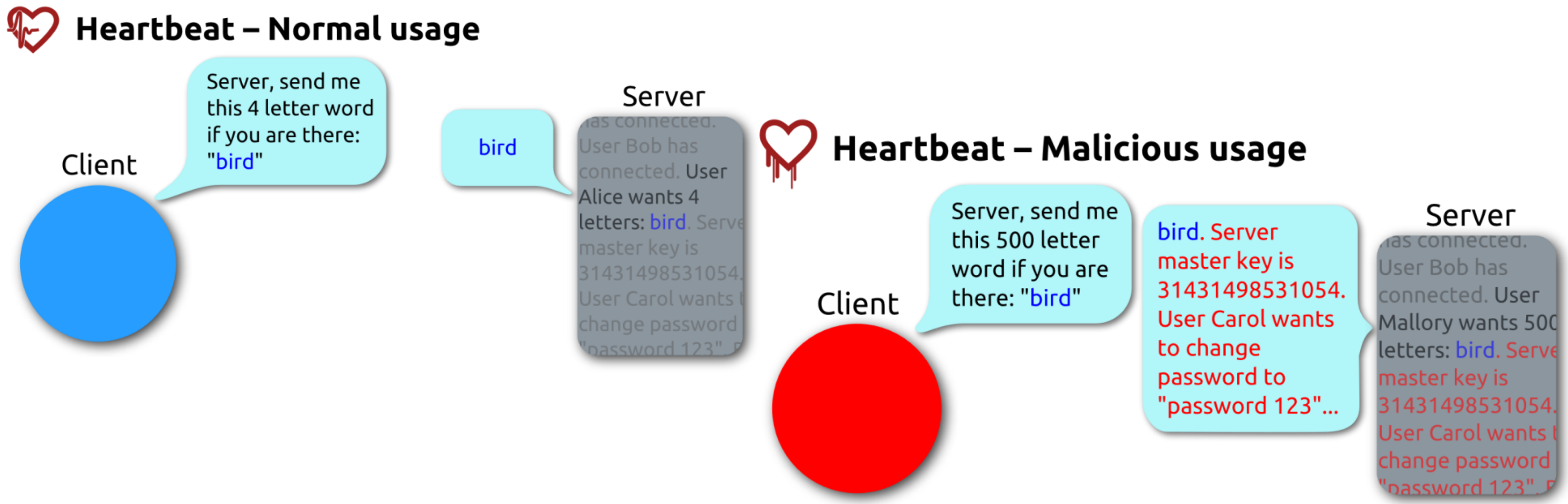
Bugs are everywhere.

So I built my own NAS.

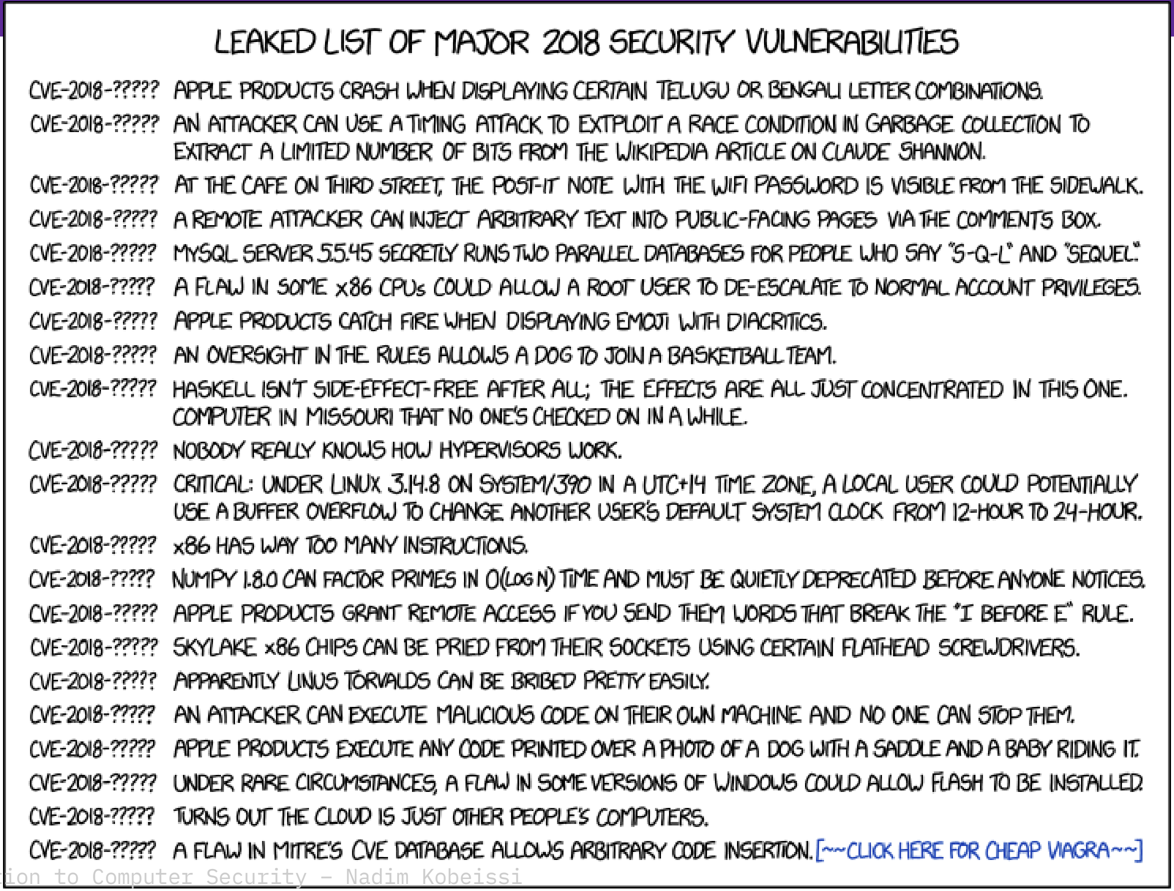
- ...which runs OpenSUSE.
- CVE-2011-3172: Log into any disabled user account in SUSE Linux.
- You can't ever avoid bugs in the long run, only *minimize your attack surface.*



Heartbleed: another notable bug.


Heartbeat – Normal usage
Client
Server, send me this 4 letter word if you are there: "bird"
bird
Server
has connected. User Bob has connected. User Alice wants 4 letters: bird. Serve master key is 31431498531054. User Carol wants t change password "password 123". P
Heartbeat – Malicious usage
Client
Server, send me this 500 letter word if you are there: "bird"
bird. Server master key is 31431498531054. User Carol wants to change password to "password 123"...
Server
has connected. User Bob has connected. User Mallory wants 500 letters: bird. Serve master key is 31431498531054. User Carol wants t change password "password 123". P

Leaked 2018 CVE list.



LEAKED LIST OF MAJOR 2018 SECURITY VULNERABILITIES

CVE-2018-?????	APPLE PRODUCTS CRASH WHEN DISPLAYING CERTAIN TELUGU OR BENGALI LETTER COMBINATIONS.
CVE-2018-?????	AN ATTACKER CAN USE A TIMING ATTACK TO EXTPLOIT A RACE CONDITION IN GARBAGE COLLECTION TO EXTRACT A LIMITED NUMBER OF BITS FROM THE WIKIPEDIA ARTICLE ON CLAUDE SHANNON.
CVE-2018-?????	AT THE CAFE ON THIRD STREET, THE POST-IT NOTE WITH THE WIFI PASSWORD IS VISIBLE FROM THE SIDEWALK.
CVE-2018-?????	A REMOTE ATTACKER CAN INJECT ARBITRARY TEXT INTO PUBLIC-FACING PAGES VIA THE COMMENTS BOX.
CVE-2018-?????	MYSQL SERVER 5.5.45 SECRETLY RUNS TWO PARALLEL DATABASES FOR PEOPLE WHO SAY "S-Q-L" AND "SEQUEL."
CVE-2018-?????	A FLAW IN SOME x86 CPUs COULD ALLOW A ROOT USER TO DE-ESCALATE TO NORMAL ACCOUNT PRIVILEGES.
CVE-2018-?????	APPLE PRODUCTS CATCH FIRE WHEN DISPLAYING EMOJI WITH DIACRITICS.
CVE-2018-?????	AN OVERSIGHT IN THE RULES ALLOWS A DOG TO JOIN A BASKETBALL TEAM.
CVE-2018-?????	HASKELL ISN'T SIDE-EFFECT-FREE AFTER ALL; THE EFFECTS ARE ALL JUST CONCENTRATED IN THIS ONE COMPUTER IN MISSOURI THAT NO ONE'S CHECKED ON IN A WHILE.
CVE-2018-?????	NOBODY REALLY KNOWS HOW HYPERVISORS WORK.
CVE-2018-?????	CRITICAL: UNDER LINUX 3.14.8 ON SYSTEM/390 IN A UTC+14 TIME ZONE, A LOCAL USER COULD POTENTIALLY USE A BUFFER OVERFLOW TO CHANGE ANOTHER USER'S DEFAULT SYSTEM CLOCK FROM 12-HOUR TO 24-HOUR.
CVE-2018-?????	x86 HAS WAY TOO MANY INSTRUCTIONS.
CVE-2018-?????	NUMPY 1.8.0 CAN FACTOR PRIMES IN O(LOG N) TIME AND MUST BE QUIETLY DEPRECATED BEFORE ANYONE NOTICES.
CVE-2018-?????	APPLE PRODUCTS GRANT REMOTE ACCESS IF YOU SEND THEM WORDS THAT BREAK THE "I BEFORE E" RULE.
CVE-2018-?????	SKYLAKE x86 CHIPS CAN BE PRIED FROM THEIR SOCKETS USING CERTAIN FLATHEAD SCREWDRIVERS.
CVE-2018-?????	APPARENTLY LINUS TORVALDS CAN BE BRIBED PRETTY EASILY.
CVE-2018-?????	AN ATTACKER CAN EXECUTE MALICIOUS CODE ON THEIR OWN MACHINE AND NO ONE CAN STOP THEM.
CVE-2018-?????	APPLE PRODUCTS EXECUTE ANY CODE PRINTED OVER A PHOTO OF A DOG WITH A SADDLE AND A BABY RIDING IT.
CVE-2018-?????	UNDER RARE CIRCUMSTANCES, A FLAW IN SOME VERSIONS OF WINDOWS COULD ALLOW FLASH TO BE INSTALLED.
CVE-2018-?????	TURNS OUT THE CLOUD IS JUST OTHER PEOPLE'S COMPUTERS.
CVE-2018-?????	A FLAW IN MITRE'S CVE DATABASE ALLOWS ARBITRARY CODE INSERTION. [~~CLICK HERE FOR CHEAP VIAGRA~~]

Categories of Vulnerabilities

3.1a

Memory Management Vulnerabilities.

- Most modern programming language have "memory management." Some expect the user to manage memory allocations manually and later de-allocate.
- *Buffer overflows*: an out-of-bounds memory index allows operations on unintended memory addresses.
- *Dangling pointers*: a program re-accesses memory that was since deallocated.

Test your knowledge!

Which of the following languages implements *garbage collection* and *memory management?*

- ☐ **A**: Go.
- ☐ **B**: C.
- ☐ **C**: C++.

Test your knowledge!

Which of the following languages implements *garbage collection* and *memory management?*

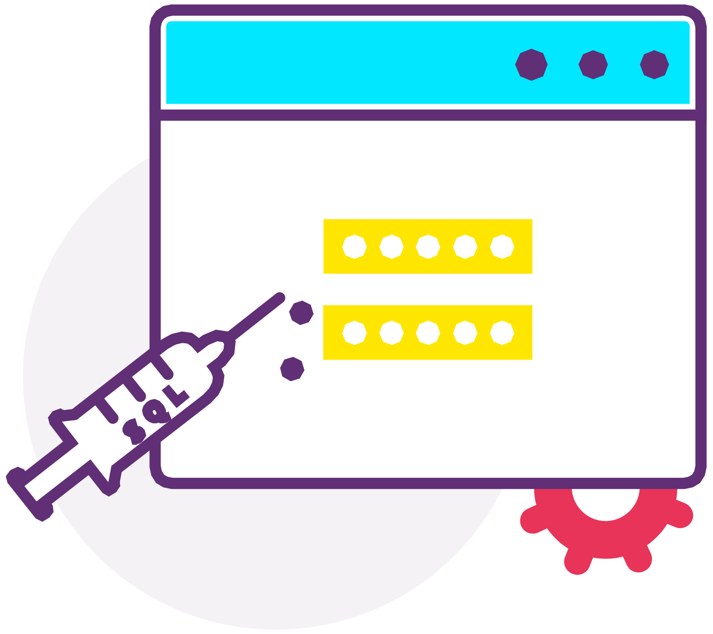
☑ **A**: Go.

☐ **B**: C.

☐ **C**: C++.

Structured Output Generation Vulnerabilities.

- Output generated by one component relies on dynamic variables, but must remain in a safe structure when processed by the receiving component.
- *SQL injections* are the most popular example.
- Can apply to command-line shells, to web scripts…



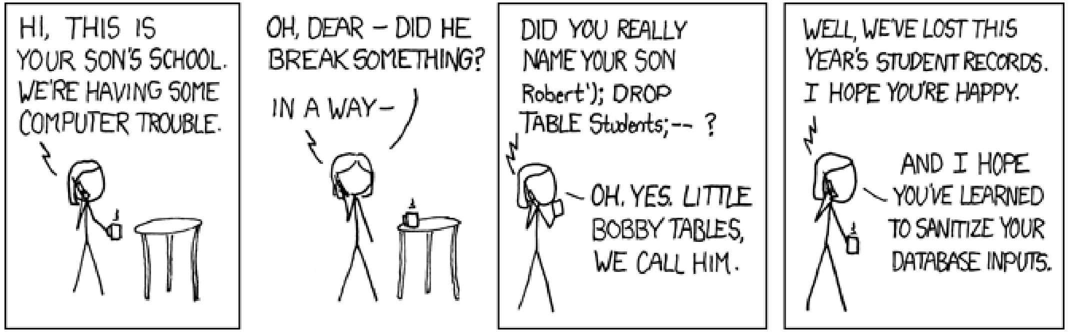
Test your knowledge!

How can this query be exploited in order to perform an SQL injection attack?

```
query = "select * from users
where name='"+ name +"'" and
pw = '"+ password +"'"
```

Test your knowledge!

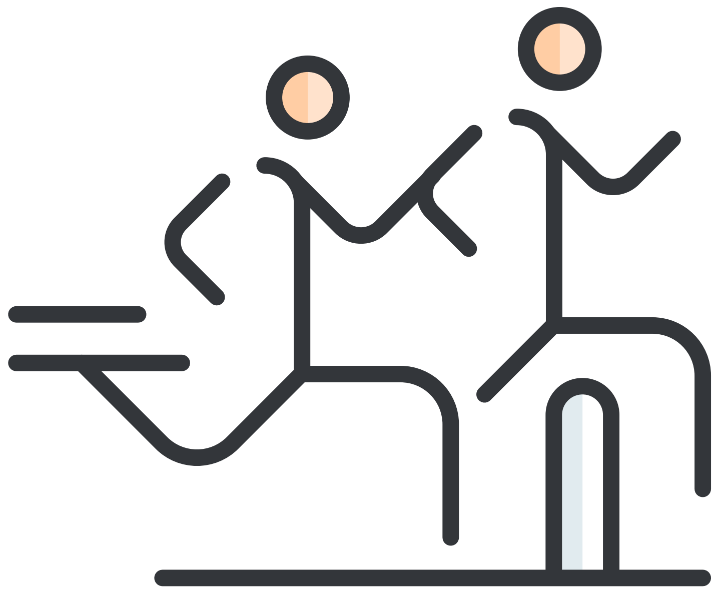
How can this query be exploited in order to perform an SQL injection attack?


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH, YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Race Condition Vulnerabilities.

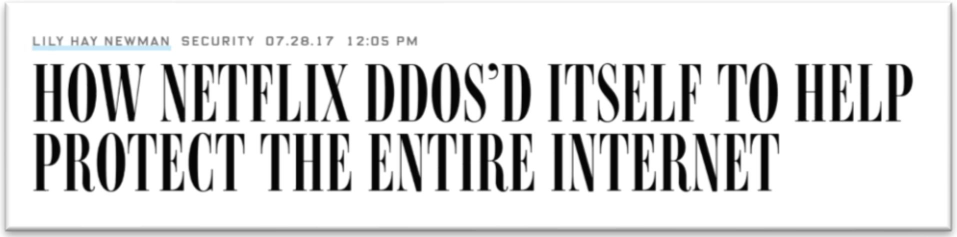
- *On a file system*: an attacker can squeeze an operation between the time permissions on a file are checked and an action is undertaken.

Seen often in programming languages focusing on concurrency (Go, or perhaps even JavaScript with Web Workers.)

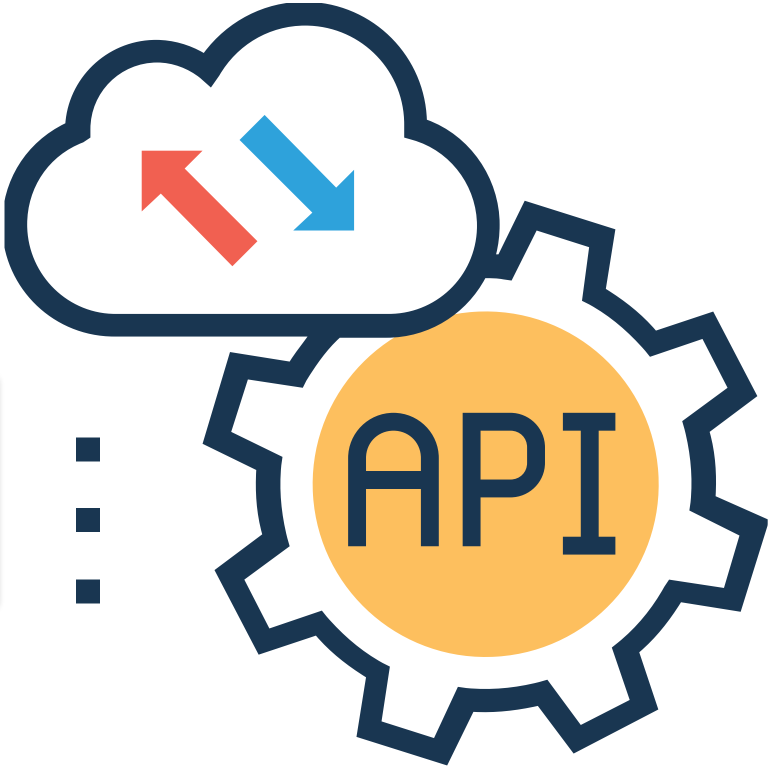


API Vulnerabilities.

- Missing access control on critical API functionality.
- Denial of service by using the API against itself.


LILY HAY NEWMAN SECURITY 07.28.17 12:05 PM

HOW NETFLIX DDOS'D ITSELF TO HELP PROTECT THE ENTIRE INTERNET



Side-channel Vulnerabilities.

We saw these when discussing cryptography.

- Power analysis can leak entire private keys.
- Timing analysis can also leak entire private keys.
- Rowhammer: maliciously crafter memory access patterns triggers reactions in high-density RAM memory cells that causes memory bits to flip.

Prevention of Vulnerabilities

3.1b

Language safety.

Perfectly well-described software means bug-free software.

- Most bugs are software not doing what we intended for it to do and computers taking us too literally.
- Garbage collection, memory management.
- Static type systems, bound checks.
- Namespace localization.

Better programming practices.

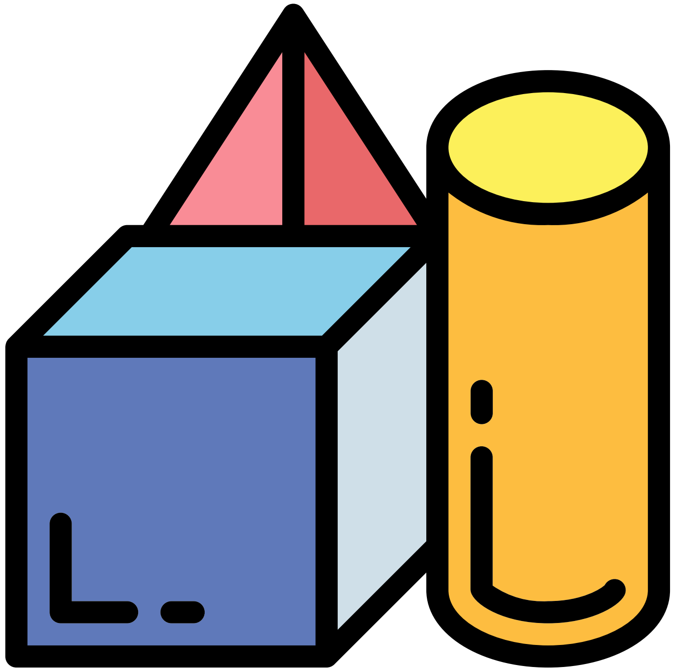
Almost completely language dependent.

- Remember to manage your pointers in C.
- Don't use `eval()` in JavaScript.
- Don't use `system()` in C.

There's an infinite number of these rules and they come largely with experience.

Typing and verifiably parsing structures.

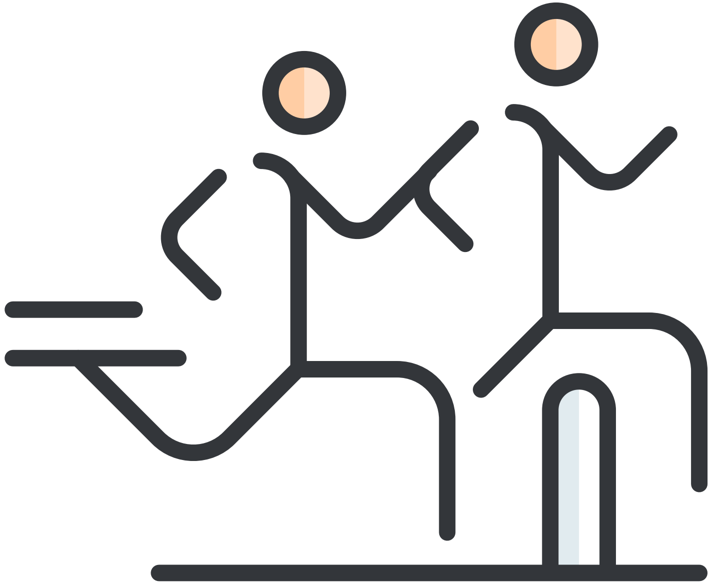
- Language Integrated Query (LINQ.)
- Regular expression types.
- Verified parsing and serializing.



Avoiding race conditions.

One relatively new method: ownership regimes.

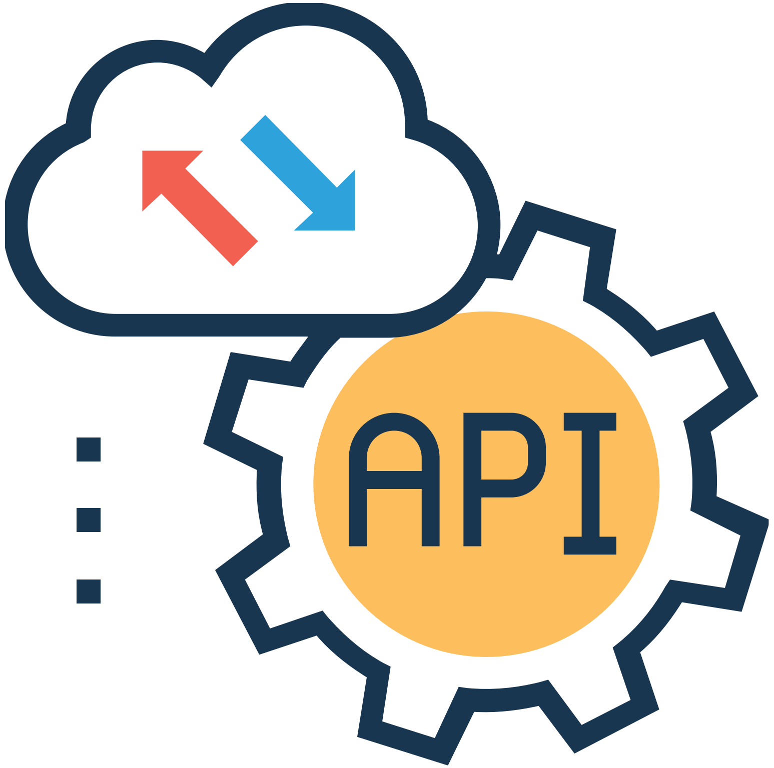
- Multiple pointers to the same resource can be created only in certain circumstances.
- Rust is the first mainstream programming language to incorporate this.



Safe API design.

It all comes down to design.

- Libsodium's entire existence is about offering a cryptography API where it's "harder to shoot yourself in the foot."
- For web APIs, compartmentalization, defensive programming play a large role.
- Implementing pre-condition and post-condition checks on APIs.



Detection of Vulnerabilities

3.1c

Static detection.

- Code analysis (automated or manual).
- Symbolic verification by building an Abstract Syntax Tree.
- Flow evaluation.

Dynamic detection and fuzzing.

- Monitoring programs and using statistic analysis.
- Black-box fuzzing: a barrage of arbitrary values over an unknown internal program structure to “see what happens.”
- White-box fuzzing: internal program structure is known, allowing optimizations to improve coverage.

Formal verification.

- For protocols, symbolic or computational verification (ProVerif, CryptoVerif, etc.) allow us to write up models that describe protocols and obtain automated proofs.
- My PhD involved translating web protocol code to formal models in ProVerif.
- F*: a new language for writing formally verified software. Dependent types, refinements, post-condition logic, etc. (ties ML to the Z3 SMT theorem prover.)

Did you know?

Microsoft Research is using F* in order to build the aptly-named Project Everest, a fully formally verified HTTPS stack.

Next time: Control Flow Hijacking

3.2