Security: Lecture 1

It’s all about the threat model

Clémentine Maurice
L3 ENS - 2020/2021

Admin

About me

- Chargée de Recherche **CNRS** at IRISA since 2017, EMSEC group **https://www.irisa.fr/emsec/**
- Previously:
    - 2012-2015: PhD from Eurecom/Technicolor
    - 2016-2017: postdoc at TU Graz (Austria)
- Security research about:
    - **Side channels** on micro-architecture
    - Software-based **fault attacks**

Course objectives

- This is an introductory course that aims at making you “security aware”
- 6 hours of lectures is not a lot for a field that vast
- Get a feeling of the different topics in computer security
- Get a feeling of the different **research challenges**
- Not (just) technical details
- Give you the security mindset: how to **think like an attacker**?
- Security is a process: there is no magic tool to make vulnerabilities disappear

Course format

10 sessions:

- 3 lectures
- 6 project sessions
- Project presentations on **December 10**
    - Aimed at being a mini-seminar to complete the lectures

Grading

- Project report due **December 4**: 50% of your grade
    - 3 to 5 pages
    - summarize the context, problem(s) you faced and the method you used
- Project presentations on **December 10**: 50% of your grade
    - 20 minutes presentation + 10 minutes of questions
    - summarize background and the two articles given, explain the aim of your project and the steps you took
- Final report due **December 17**
    - we will ask for corrections if needed, so that we can distribute your reports to the whole class
    - you can get **up to 2 bonus points** if you made significant improvements!

Projects

1. Adversarial machine learning
2. 802.11 fingerprinting
3. Password cracking
4. Automated bug finding
5. Reverse-engineering
6. Crypto in the real world
7. Buffer overflows

Projects will be tutored by me and Guillaume Didier (TA, PhD student)

https://cmaurice.fr/teaching/ENS/

Questions?

I prefer the course to be interactive and I don’t bite

Don’t hesitate to ask questions! You’re probably not the only one with the question

Sometimes I may just be wrong (hopefully not too often)

You can also reach me by email: `clementine.maurice@irisa.fr`

Introduction

Why should you care?

Security impacts everybody’s day-to-day life

Security impacts **your** day-to-day life

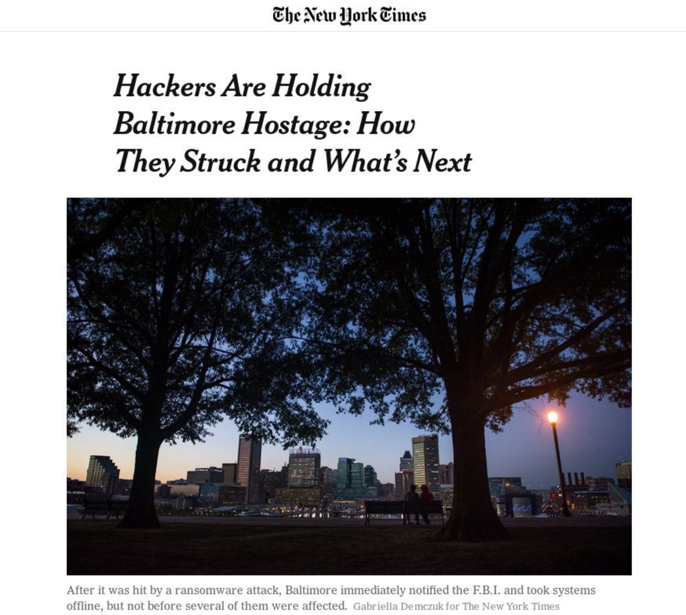
User: make safe decisions

Developer: design and build secure systems

Researcher: identify flaws and new classes of vulnerabilities, propose mitigations

Security is everywhere (in the media)

Ransomware



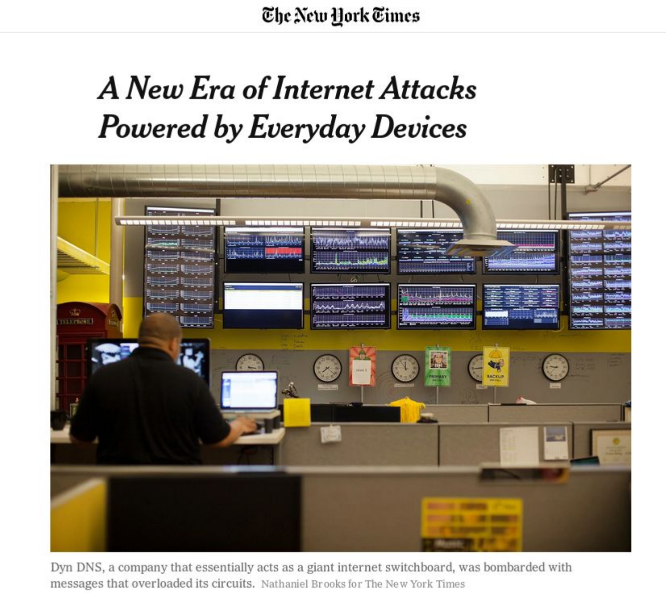
The New York Times

Hackers Are Holding Baltimore Hostage: How They Struck and What's Next

After it was hit by a ransomware attack, Baltimore immediately notified the F.B.I. and took systems offline, but not before several of them were affected. Gabriella Demczuk for The New York Times

Security is everywhere (in the media)

IoT botnet



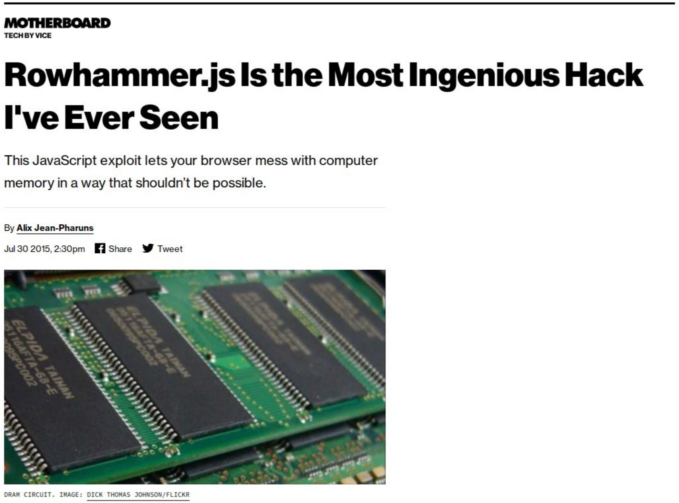
The New York Times

A New Era of Internet Attacks Powered by Everyday Devices

Dyn DNS, a company that essentially acts as a giant internet switchboard, was bombarded with messages that overloaded its circuits. Nathaniel Brooks for The New York Times

Security is everywhere (in the media)

Hardware vulnerabilities



MOTHERBOARD
TECH BY VICE

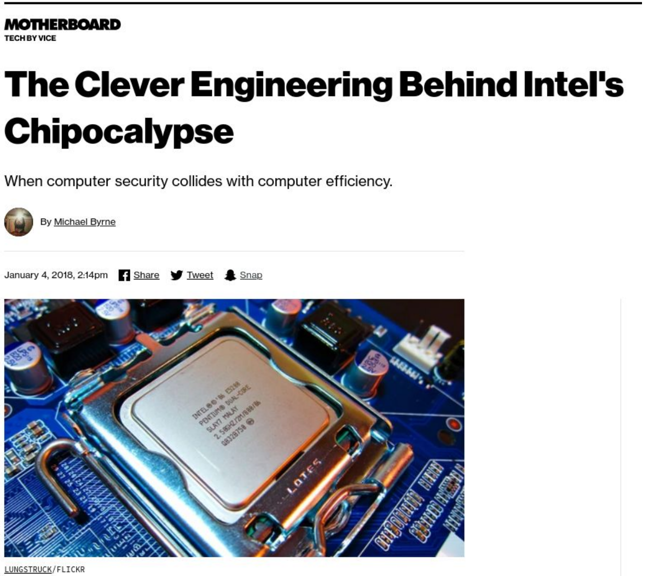
Rowhammer.js Is the Most Ingenious Hack I've Ever Seen

This JavaScript exploit lets your browser mess with computer memory in a way that shouldn't be possible.

By Alix Jean-Pharuns

Jul 30 2015, 2:30pm Share Tweet

DRAM CIRCUIT. IMAGE: DICK THOMAS JOHNSON/FLICKR



MOTHERBOARD
TECH BY VICE

The Clever Engineering Behind Intel's Chipocalypse

When computer security collides with computer efficiency.

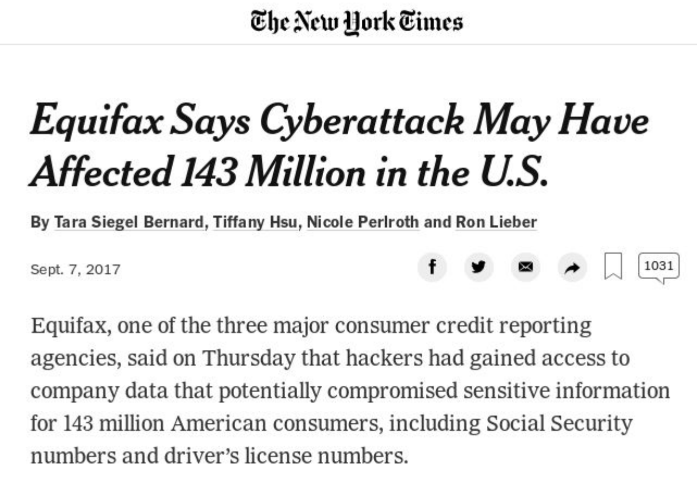
By Michael Byrne

January 4, 2018, 2:14pm Share Tweet Snap

LUNGSTRUCK/FLICKR

Security is everywhere (in the media)

Data breach



The New York Times

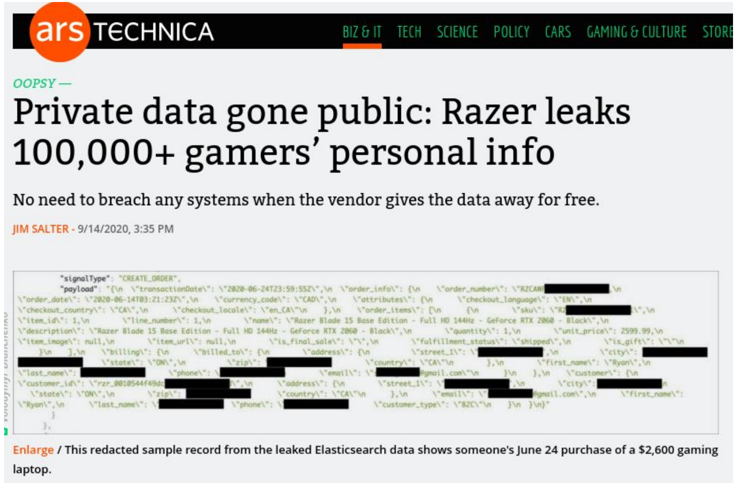
Equifax Says Cyberattack May Have Affected 143 Million in the U.S.

By Tara Siegel Bernard, Tiffany Hsu, Nicole Perlroth and Ron Lieber

Sept. 7, 2017

1031

Equifax, one of the three major consumer credit reporting agencies, said on Thursday that hackers had gained access to company data that potentially compromised sensitive information for 143 million American consumers, including Social Security numbers and driver's license numbers.



OOPSY —

Private data gone public: Razer leaks 100,000+ gamers' personal info

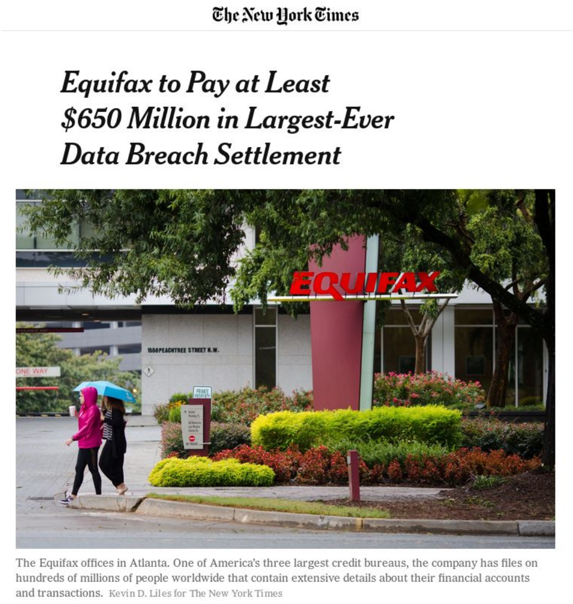
No need to breach any systems when the vendor gives the data away for free.

JIM SALTER - 9/14/2020, 3:35 PM

```
        "signalType": "CREATE_ORDER",
        "payload": "{\n  \"transactionDate\": \"2020-06-24T23:59:55Z\",\n  \"order_info\": {\n    \"order_number\": \"RZCAN[redacted],\n
\"order_date\": \"2020-06-14T03:21:23Z\",\n    \"currency_code\": \"CAD\",\n    \"attributes\": {\n      \"checkout_language\": \"EN\",\n
\"checkout_country\": \"CA\",\n      \"checkout_locale\": \"en_CA\"\n    },\n    \"order_items\": [\n      {\n        \"sku\": \"RZ[redacted]\",\n
\"item_id\": 1,\n        \"line_number\": 1,\n        \"name\": \"Razer Blade 15 Base Edition - Full HD 144Hz - GeForce RTX 2060 - Black\",\n
\"description\": \"Razer Blade 15 Base Edition - Full HD 144Hz - GeForce RTX 2060 - Black\",\n        \"quantity\": 1,\n        \"unit_price\": 2599.99,\n
\"item_image\": null,\n        \"item_url\": null,\n        \"is_final_sale\": \"\",\n        \"fulfillment_status\": \"shipped\",\n        \"is_gift\": \"\"\n
    }\n    ],\n    \"billing\": {\n      \"billed_to\": {\n        \"address\": {\n          \"street_1\": \"[redacted],\n          \"city\": [redacted]
[redacted]          \"state\": \"ON\",\n          \"zip\": [redacted]          \"country\": \"CA\"\n        },\n        \"first_name\": \"Ryan\",\n
\"last_name\": [redacted]        \"phone\": \[redacted]        \"email\": \"[redacted]@gmail.com\"\n      }\n    },\n    \"customer\": {\n
\"customer_id\": \"rzr_0010544f49dc[redacted]\",\n      \"address\": {\n        \"street_1\": \"[redacted],\n        \"city\":[redacted]n
  \"state\": \"ON\",\n        \"zip\": [redacted]        \"country\": \"CA\"\n      },\n      \"email\": \"[redacted]@gmail.com\",\n      \"first_name\":
\"Ryan\",\n      \"last_name\": \[redacted]      \"phone\": \[redacted]      \"customer_type\": \"B2C\"\n    }\n  }\n}"
      }
    },
```

Enlarge / This redacted sample record from the leaked Elasticsearch data shows someone's June 24 purchase of a $2,600 gaming laptop.

(In)security costs a lot of money



The New York Times

Equifax to Pay at Least $650 Million in Largest-Ever Data Breach Settlement

The Equifax offices in Atlanta. One of America's three largest credit bureaus, the company has files on hundreds of millions of people worldwide that contain extensive details about their financial accounts and transactions. Kevin D. Liles for The New York Times

Good and bad hackers

RFC 1392

- **Hacker**: A person who delights in having an intimate understanding of the internal workings of a system, computers and computer networks in particular. The term is often misused in a pejorative context, where "cracker" would be the correct term.
- **Cracker**: A cracker is an individual who attempts to access computer systems without authorization. These individuals are often malicious, as opposed to hackers, and have many means at their disposal for breaking into a system.

Good and bad hackers

Malicious hacking/cracking is illegal

> *« Le fait d'**accéder ou de se maintenir, frauduleusement**, dans tout ou partie d'un système de traitement automatisé de données est puni de **deux ans d'emprisonnement et de 30000 euros d'amende**. Lorsqu'il en est résulté soit la suppression ou la modification de données contenues dans le système, soit une altération du fonctionnement de ce système, la peine est de trois ans d'emprisonnement et de 45000 euros d'amende. »*

However, discussing vulnerabilities and how they are actually exploited is useful to **educate and increase awareness**

Academia and hackers

- Researchers in computer science, especially security, publish more in conferences than journals
- **Dedicated academic security conferences**: S&P, USENIX Security, CCS, NDSS
- Security tracks or workshops in most major academic conferences of **other domains**, e.g.
  - ICSE (software engineering)
  - ISCA (micro-architecture)
  - FM (formal methods)
  - WWW (web)
  - ICML (machine learning)
  - INFOCOM (network)
- Also **hacker conferences**: Chaos Communication Congress (CCC), Black Hat, DEF CON...

Definitions and principles

Security

Integrity: Data has not been altered or destroyed in an unauthorized manner

Confidentiality: Information is not made available or disclosed to unauthorized individuals, entities or processes

Availability: Data/service is accessible and usable upon demand by an authorized entity. Failure to meet this goal is called a denial of service

→ Data that is stored in a system that is unpowered and unplugged from any network has high integrity and confidentiality, but low availability…

Bug, vulnerabilities, attacks...

A human error may introduce a **bug** or **fault**

If the fault is triggered, it generates a **failure** (e.g., Windows' blue screen of death)

If the fault is security-related, it is called a **vulnerability** → not all bugs are vulnerabilities

An **attack** happens when the vulnerability is triggered, or exploited → not all vulnerabilities can be easily exploited

A **zero-day** refers to a vulnerability that has just been revealed, for which there is therefore no patch (developers had zero day to issue a patch)

Security overview

Architecture: security considerations when **designing** the application

Implementation: security considerations when **writing** the application

Operation: security considerations when the application is **in production**

Architecture and design

- Validation of requirements = building the right model
- Verification of design = building the model right
- Common problems
    - authentication and privileges
        - session replay
        - principle of least privilege
    - communication protocol design
        - sniffing, man-in-the-middle
        - session hijacking
    - denial of service

Implementation

- Classic vulnerabilities (often programming-language-specific)
- Common problems
    - buffer overflows
        - static: stack-based buffer overflows
        - dynamic: heap-based buffer overflows
    - input validation
        - URL encoding
        - document root escape
        - SQL injection
    - backdoors

Operation

- Decisions made after software is deployed
- Often not under developer’s control
- Common problems
  - denial of service (DOS)
    - network DOS
    - distributed DOS, zombies
  - administration problems
    - weak passwords
    - password cracking
    - unsafe defaults

Kerckhoffs' principle

- "A cryptosystem should be secure even if **everything about the system**, except the key, **is public knowledge**"
- Security through obscurity is considered dangerous
- Idea sometimes debated, pros and cons?

Kerckhoffs' principle

- Basic idea: given enough time, somebody will be able to figure out your "secure design", at which point it will be trivially broken
- Today's crypto standards are open: the implementation of AES and RSA is well known, and the security relies not on the fact that the attacker does not know the algorithm, but on the fact that the attacker does not know the key (and it would take them millions of years to guess it)

Principle of least privilege

- Only granting **permissions that are necessary and sufficient** for a particular task
- Applicable to processes, users, services…
- Examples?

Principle of least privilege

- Only granting **permissions that are necessary and sufficient** for a particular task
- Applicable to processes, users, services…
- Examples in military contexts (*need-to-know*), but usage throughout all modern systems
  - not all accounts have administrator rights
  - fine-grained permissions on mobile devices → a flashlight or a simple game should not ask for permission to access your contacts!
  - sandboxes like JavaScript → a website does not have arbitrary execution on your machine
  - kernel mode/user mode → vulnerabilities in one application cannot be used to exploit the rest of the machine

Don’t trust your input

- Computers are dumb: they do exactly what we ask them to do, not less, not more
- Input data is “just” a sequence of symbols or bytes, but **programmers make assumptions**, e.g., “this is a name”, “this is a file”, …
- **Unexpected input** is a large source of software vulnerabilities: buffer overflows, SQL injections, XSS
- Think like an attacker: what should I input in this system to obtain what I want?
- **Non trivial problem**

Minimize the attack surface

- Minimize the **"code" surface**
  - number of open sockets, services, services running by default, services running with high privileges
  - number of dynamic content web pages
  - number of files & directories with weak access control

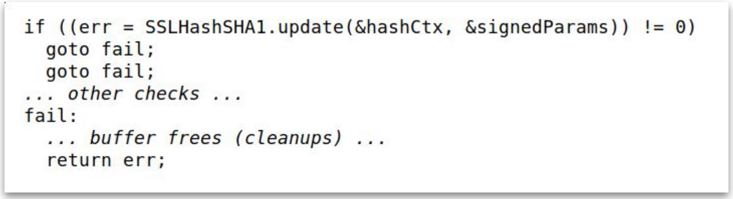
  → every additional line of code has a potential vulnerability
- Minimize the **"time" surface**
  - automatically lock screen after n minutes
  - zero-out memory containing sensitive information (e.g., decrypted information) as soon as it's no longer needed
  - other example?

Challenges in security

Why good people write bad code

Remember the Mars Climate Orbiter incident from 1999?

- **Technical** factors
  - algorithm **complexity**, multi-threaded applications, multi-user systems, **composition** (Mars Climate Orbiter), consequences of small changes hard to predict (Apple goto fail)
- **Economic** factors
  - deadlines, security is not a feature, insufficient funding, legacy software, open-source/closed-source
- **Human** factors
  - poor risk assessment, mental models: assume software is used for specific task, only check for understood errors



```
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
  goto fail;
  goto fail;
... other checks ...
fail:
  ... buffer frees (cleanups) ...
  return err;
```

Systems are increasingly complex

- In 1969, the printed code for NASA's Apollo Guidance Computer was as high as Margaret Hamilton who wrote it
- Today:
  - Google Chrome: 76 MLoC
  - Gnome: 9 MLoC
  - Xorg: 1 MLoC
  - glibc: 2 MLoC
  - Linux kernel: 17 MLoC
  - **Chrome and OS**: ~100 MLoC → 27 lines/page, 0.1mm/page ≈ **370m** → that's higher than the Eiffel Tower

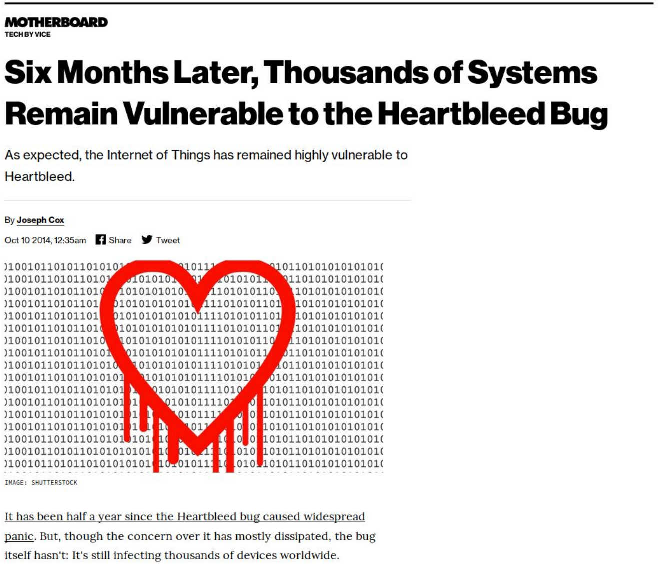
Patching is hard

A bug is found and we know how to patch it? The problem will likely remain for a long time

- incompatibility issues
- legacy systems, e.g., admin left the company
- users don't like changes in functionality

What if you need to replace the whole machine?

What if the system is a pacemaker?



MOTHERBOARD
TECH BY VICE

Six Months Later, Thousands of Systems Remain Vulnerable to the Heartbleed Bug

As expected, the Internet of Things has remained highly vulnerable to Heartbleed.

By Joseph Cox

Oct 10 2014, 12:35am Share Tweet

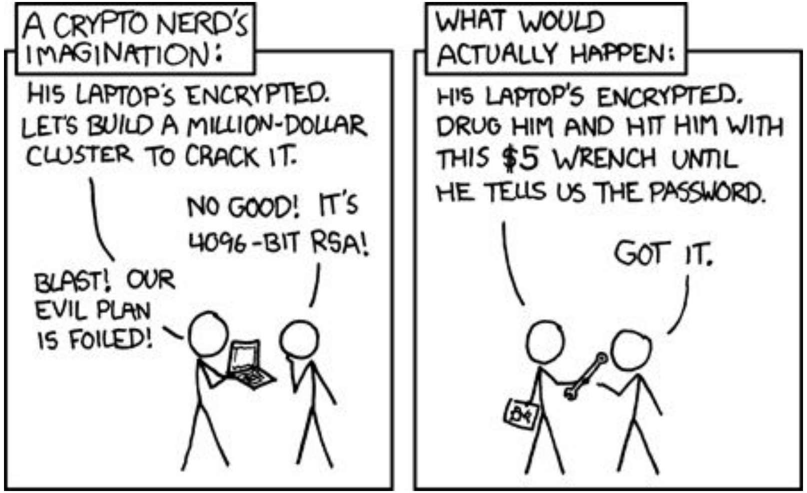
IMAGE: SHUTTERSTOCK

It has been half a year since the Heartbleed bug caused widespread panic. But, though the concern over it has mostly dissipated, the bug itself hasn't: It's still infecting thousands of devices worldwide.

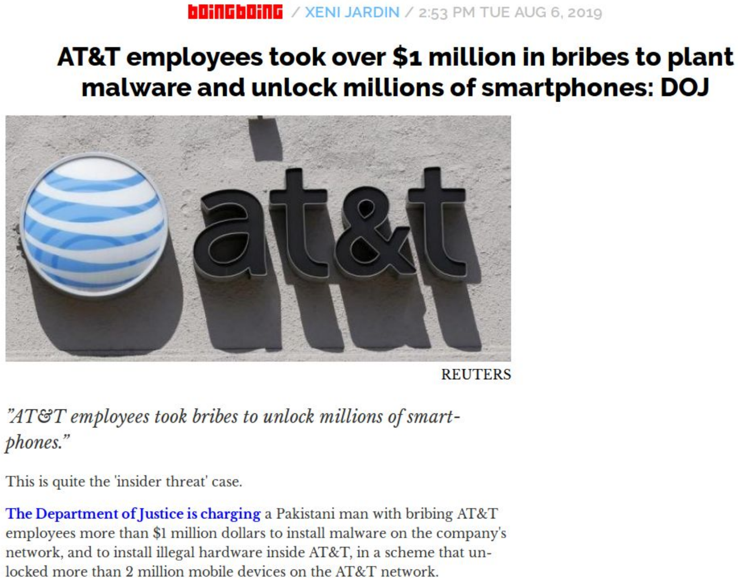
“It’s all about the threat model”

A system is only **as secure as its weakest component**

Very frequently, the user is the weakest link


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

“It’s all about the threat model”



boingboing / XENI JARDIN / 2:53 PM TUE AUG 6, 2019

AT&T employees took over $1 million in bribes to plant malware and unlock millions of smartphones: DOJ

REUTERS

”AT&T employees took bribes to unlock millions of smartphones.”

This is quite the 'insider threat' case.

The Department of Justice is charging a Pakistani man with bribing AT&T employees more than $1 million dollars to install malware on the company's network, and to install illegal hardware inside AT&T, in a scheme that unlocked more than 2 million mobile devices on the AT&T network.

“It’s all about the threat model”

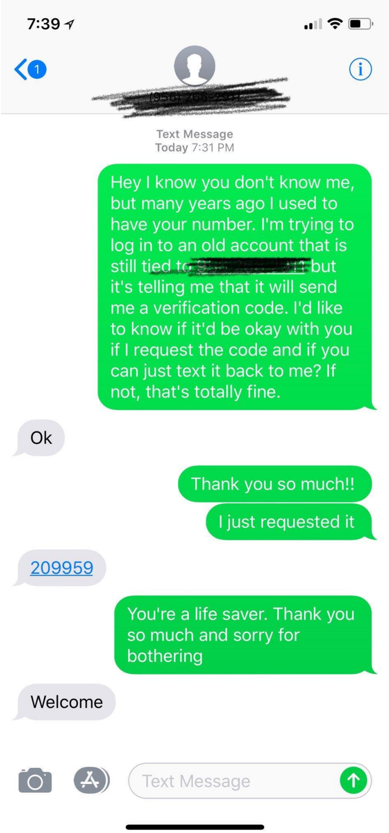
- Kevin Mitnick, famous hacker in the 80s/90s,
- First hacker “most wanted” by the FBI until his arrest in 1995
- Spent five years in prison for wire fraud, possession of unauthorized access devices, interception of wire or electronic communications, unauthorized access to a federal computer, and causing damage to a computer
- He states that he compromised computers solely by using passwords and codes gained by **social engineering**, and did not use software programs for exploiting computer or phone security.
- Now owns a security consulting firm

“It’s all about the threat model”

Several incidents using social engineering to **bypass 2-factor authentication**

- 2-factor authentication: authenticate somebody using something they know (usual password) and something they have (e.g., their phone or a specific hardware device).
- What if somebody asks to **change the phone number** on your behalf?
- What if somebody directly **asks you the code**?

http://www.businessinsider.fr/us/deray-mckesson-twitter-hacked-social-engineering-2016-6



Text Message
Today 7:31 PM

Hey I know you don't know me, but many years ago I used to have your number. I'm trying to log in to an old account that is still tied to [redacted] but it's telling me that it will send me a verification code. I'd like to know if it'd be okay with you if I request the code and if you can just text it back to me? If not, that's totally fine.

Ok

Thank you so much!!

I just requested it

209959

You're a life saver. Thank you so much and sorry for bothering

Welcome

“It’s all about the threat model”

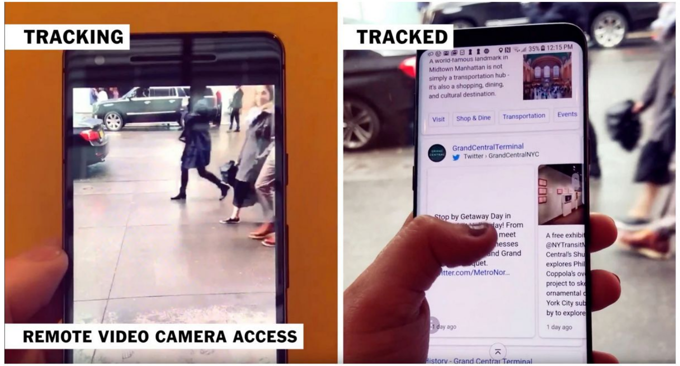
2013: **Snowden revelations on the NSA**, threat model changes. Your adversary is now highly funded and

- has **secret court orders** to sweep up phone records
- requests user data from Google, Facebook, Apple…
- forces companies to install **backdoors**
- taps fiber optic cables and **promotes weak cryptographic algorithms** to break encryption
- breaks into the links between datacenters of big companies (Yahoo, Google)

Your threat model is not my threat model

- People may have threats more immediate than the NSA
- Many apps that enable spying; sometimes target parents to keep tab on their children, others directly target people who want to spy on their partner or ex-partner
- Stalking is a top warning sign for attempted homicide in **domestic violence cases**
- Not everybody has the same threat model, especially marginalized groups

Hundreds of Apps Can Empower Stalkers to Track Their Victims



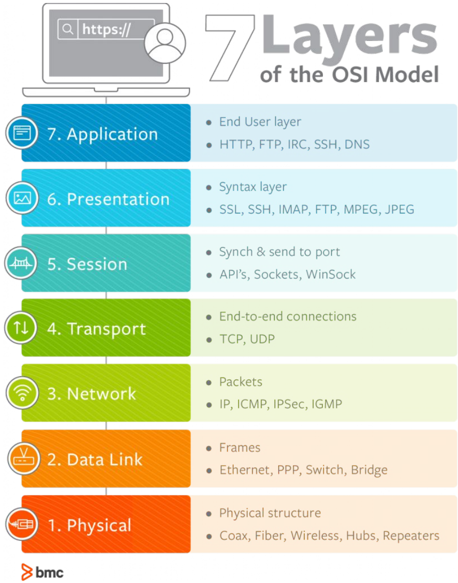
More than 200 apps and services offer would-be stalkers a variety of electronic capabilities, including basic location tracking, harvesting texts and secretly recording video. Drew Jordan/The New York Times

Usability

- A more recent domain
- Basic idea: security experts design applications, but “normal” people use them
- Easy-to-use human interface
  - easy to apply security mechanisms routinely
  - easy to apply security mechanisms correctly → **secure by default**, impossible to do things unsecurely
  - interface has to support mental model → do what is expected **intuitively**
- Example with authentication: **passwords** are a user’s nightmare
  - users tend to **reuse** them → problem in case of one breach
  - users tend to have very simple passwords (“123456”, “password”, ...) → too easy to guess
  - policies that enforce frequent changes make things worse → users make them even **easier to guess** (e.g. “password1”, then “password2”, etc.)

Abstraction layers

- Abstraction layers = separation of concerns
- **Generalization of a conceptual model**, away from any specific implementation
- The layer on top does not need to know how the layer below works, just how to interact with it
- Great for **interoperability** → you don't need to know binary code to program an app, nor how hardware component work, nor what does each of the billions of transistors in your computer



7. Application	End User layer HTTP, FTP, IRC, SSH, DNS
6. Presentation	Syntax layer SSL, SSH, IMAP, FTP, MPEG, JPEG
5. Session	Synch & send to port API's, Sockets, WinSock
4. Transport	End-to-end connections TCP, UDP
3. Network	Packets IP, ICMP, IPSec, IGMP
2. Data Link	Frames Ethernet, PPP, Switch, Bridge
1. Physical	Physical structure Coax, Fiber, Wireless, Hubs, Repeaters

bmc

Abstraction layers

> (In)Security lives and breathes in the cracks between abstraction layers.
> @halvarflake

Abstraction layers

- Original analog network of AT&T used tone dialing, with tones for internal telephone company use
- 2600Hz = tone used to signal the phone call is over
- Playing this frequency “tricks” the company switch into thinking the call is over → free long-distance calls
- John Draper discovers that the whistle toy in Cap'n Crunch cereal boxes plays this exact frequency

Abstraction layers

Multiple examples that include side-channel attacks and fault attacks, but we’ll see that in Lecture 3

How do we improve the situation?

- Strategies: **avoidance**, defense and detection
- Tools
  - detect mistakes and vulnerabilities
  - support programmer
  - formal verification
- Standards and metrics
  - hold vendors accountable
  - allow for comparison between products
- Education
  - that’s what we are trying to do here ;-)

Projects

Projects

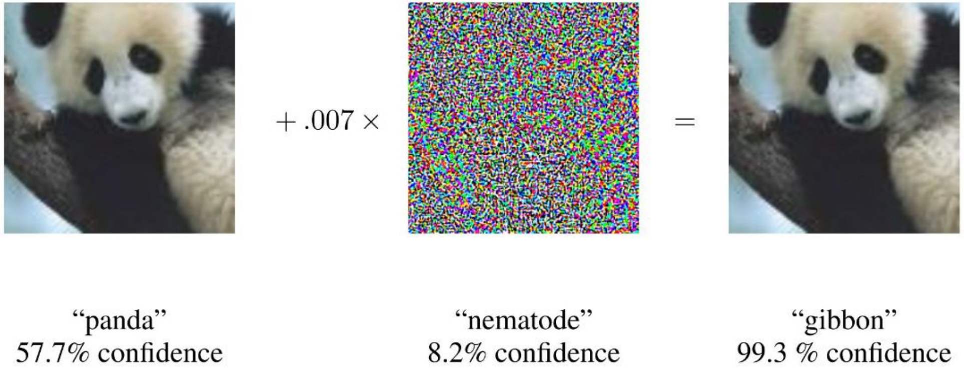
Complete list with details: https://cmaurice.fr/teaching/ENS/

May be subject to change

#1: Adversarial machine learning

Can you trick a computer into thinking a panda is a gibbon, a turtle, or a plane?

Yes.



#2 802.11 fingerprinting

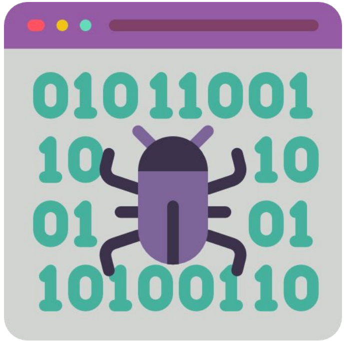
- Wi-Fi devices leave a lot of traces
- These traces can be used to track devices or people
- Observe **network packets in the wild**!

#3 Password cracking

- Passwords are the worst (but they are the best we have)
- Administrators store them unsecurely, users choose weak ones and reuse them
- All of it is a gift for **password crackers (you!)**

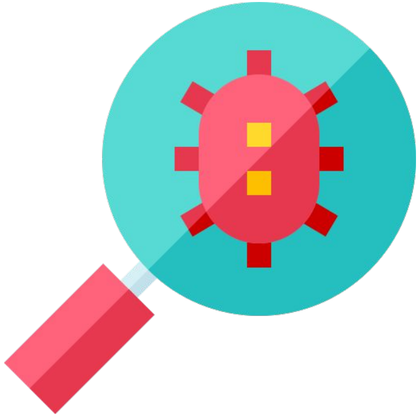
#4 Automated bug finding

- Bugs! They are everywhere and can be dangerous for security
- Discover methods to automatically find them
- Gotta catch 'em all (without too much effort)!



#5 Reverse-engineering

- You have a binary, you don’t know what it does
- Maybe it’s even malicious!
- Learn reverse-engineering techniques through a **series of challenges**

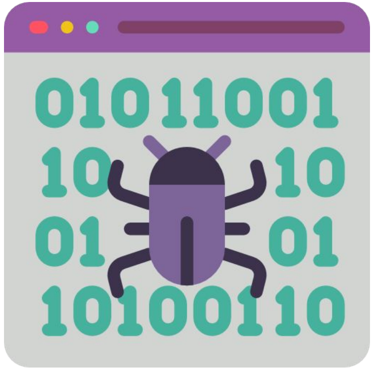


#6 Crypto in the real world

- Crypto is the best! It’s maths, so it should be secure
- But even maths can’t save you if used badly
- Break all the crypto things (**without any key**)!

#7 Buffer overflows

- Buffer overflows have been around since mid 90s but the vulnerability is still quite common today
- Learn how to **exploit all the things**!



Next lecture:

How to protect data and communications?
System and software security