Computer Systems Security

CSE 466
Fall 2018
Yan Shoshitaishvili

http://pwn.college
http://groups.google.com/group/cse-466
https://goo.gl/hVQJBM

What is Computer Systems Security?

What is Computer Systems Security?


SECURITY
protection
surveillance
electronic
bodyguard
private
web
close
CONTROL
SOFTWARE
padlock
danger
detection
password
SIGN
DATA
NETWORK
TECHNOLOGY
CONNECTION
video
display
key
CAMERA
GUARD
PROPERTY
digital
SAFEGUARD
INFORMATION
SAFETY
communication
MONITORING
IDENTITY
CONCEPT
SECRET
spy
internet
equipment
ACCESSPRIVACY
firewall
crime
safe
lock

What is Computer Systems Security?

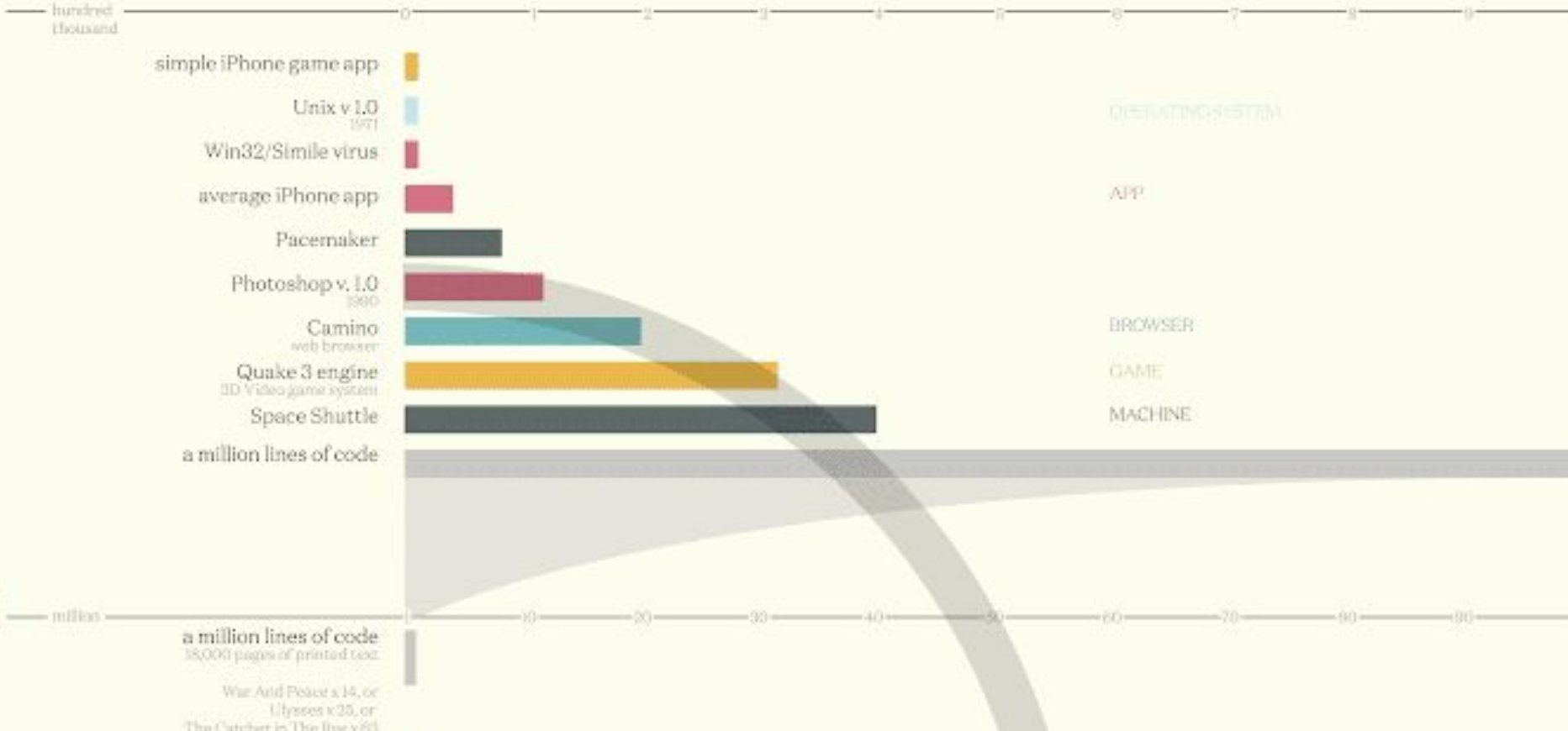

Gourmet
Coffee

A chain is only as strong as its weakest link.

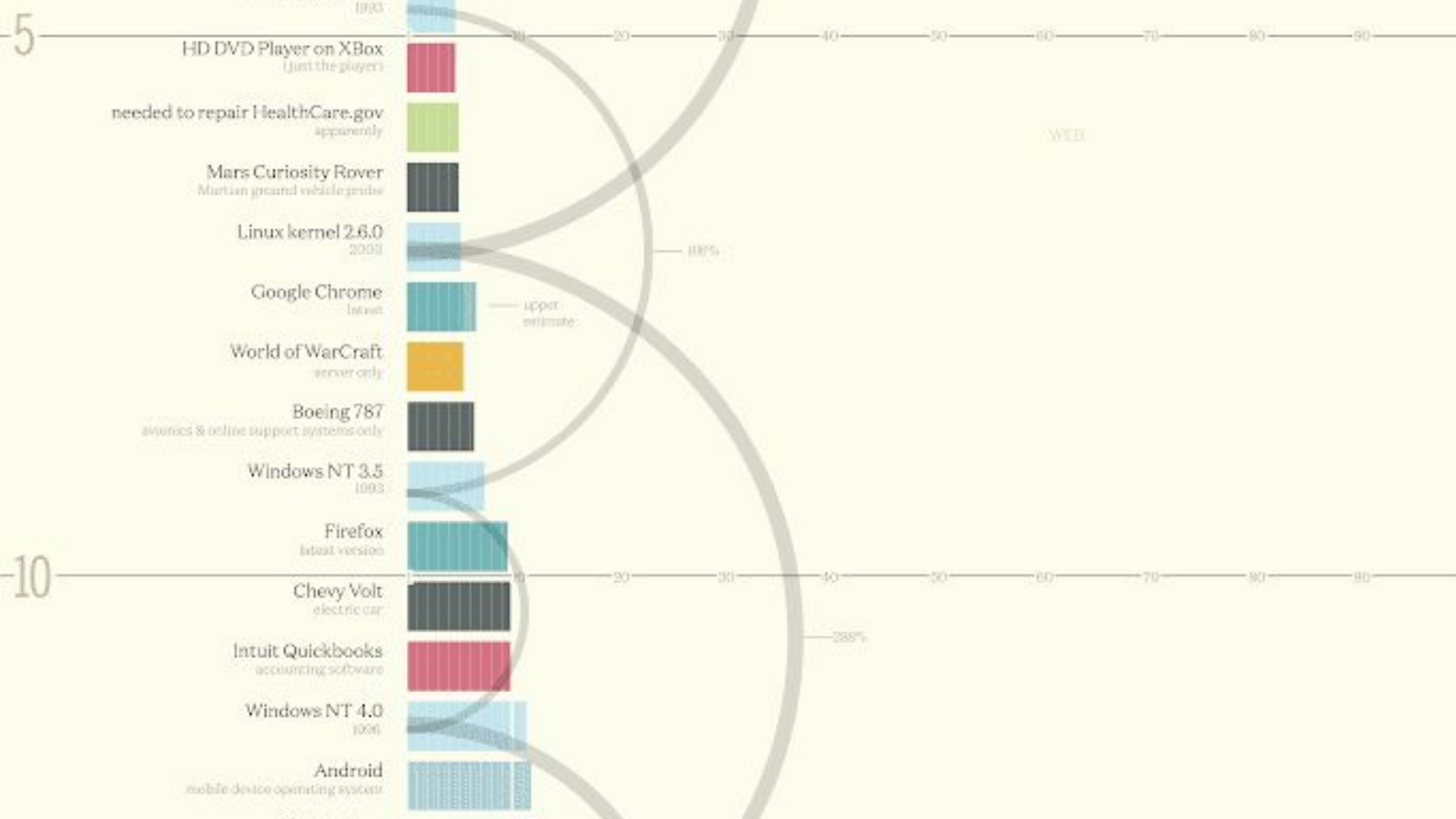
What is Computer Systems Security?

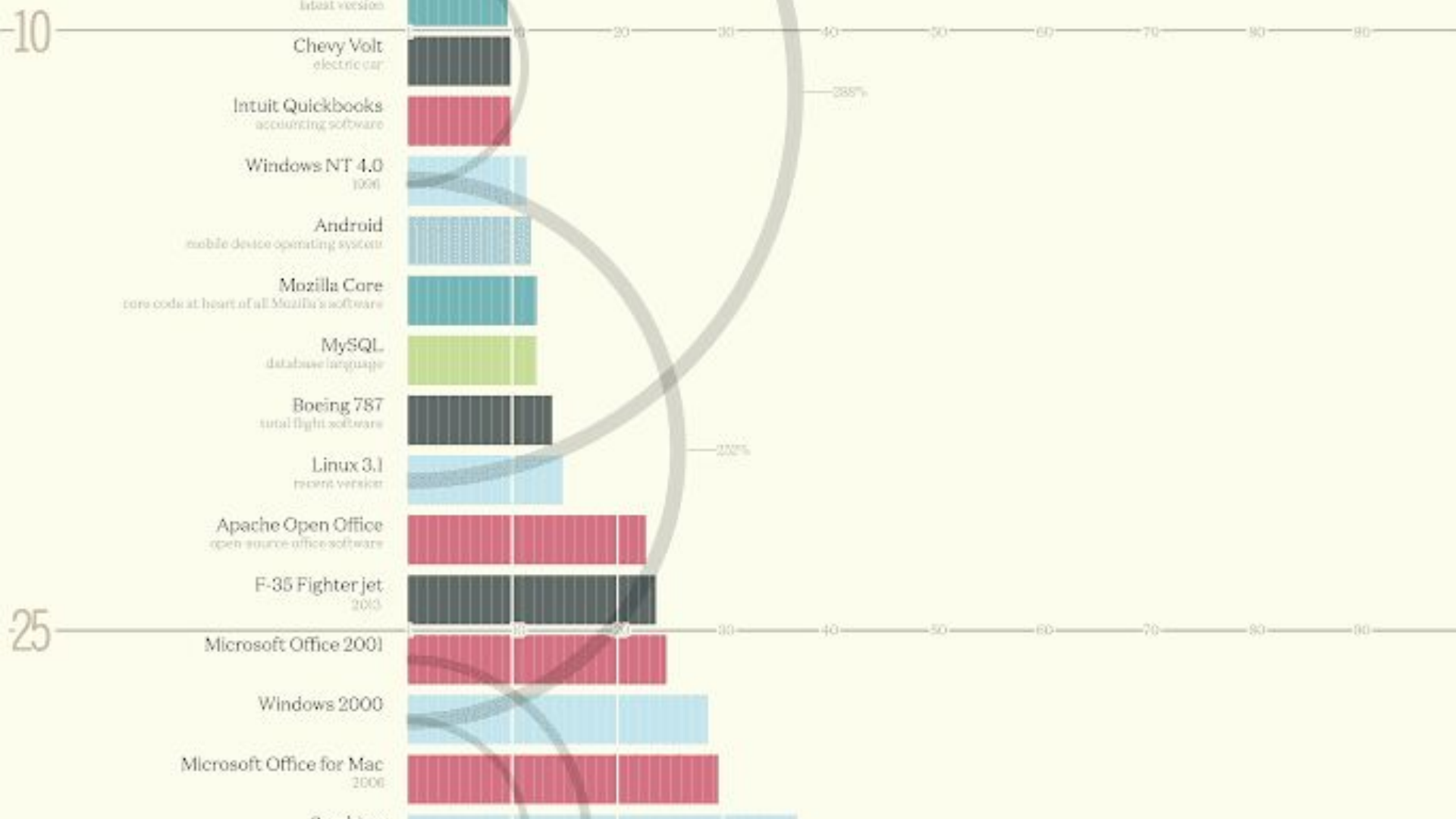
Codebases

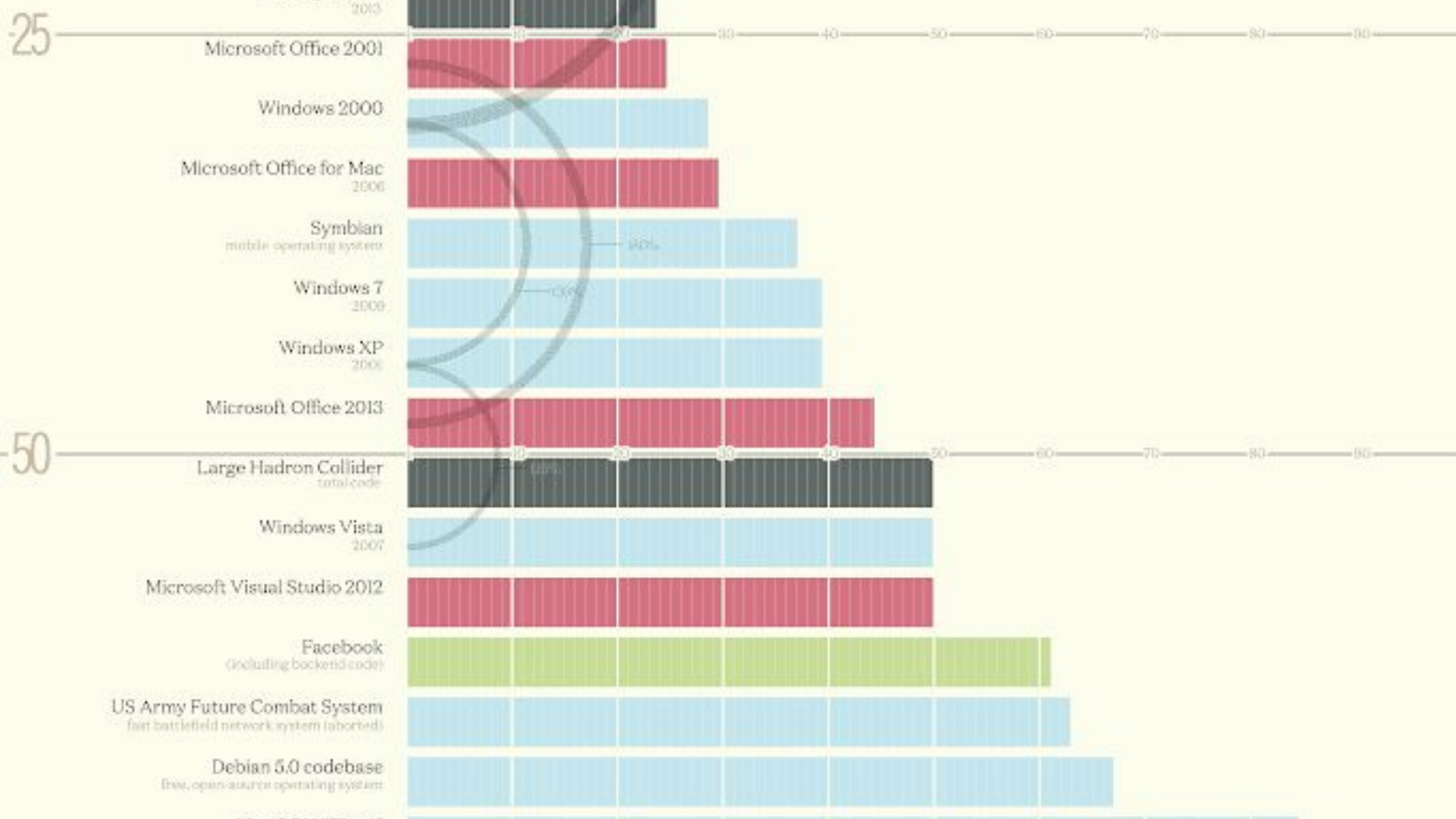
Millions of lines of code

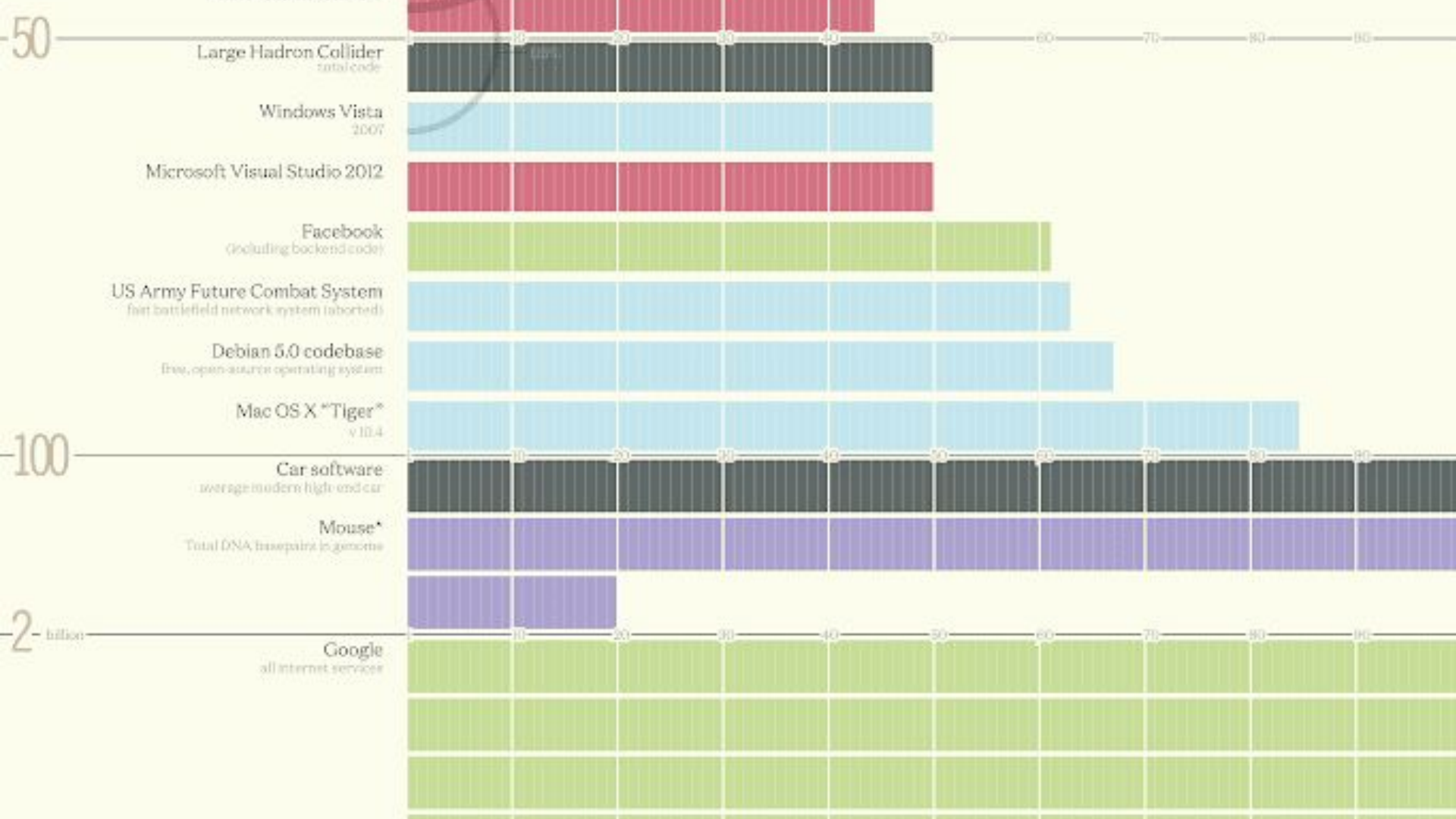


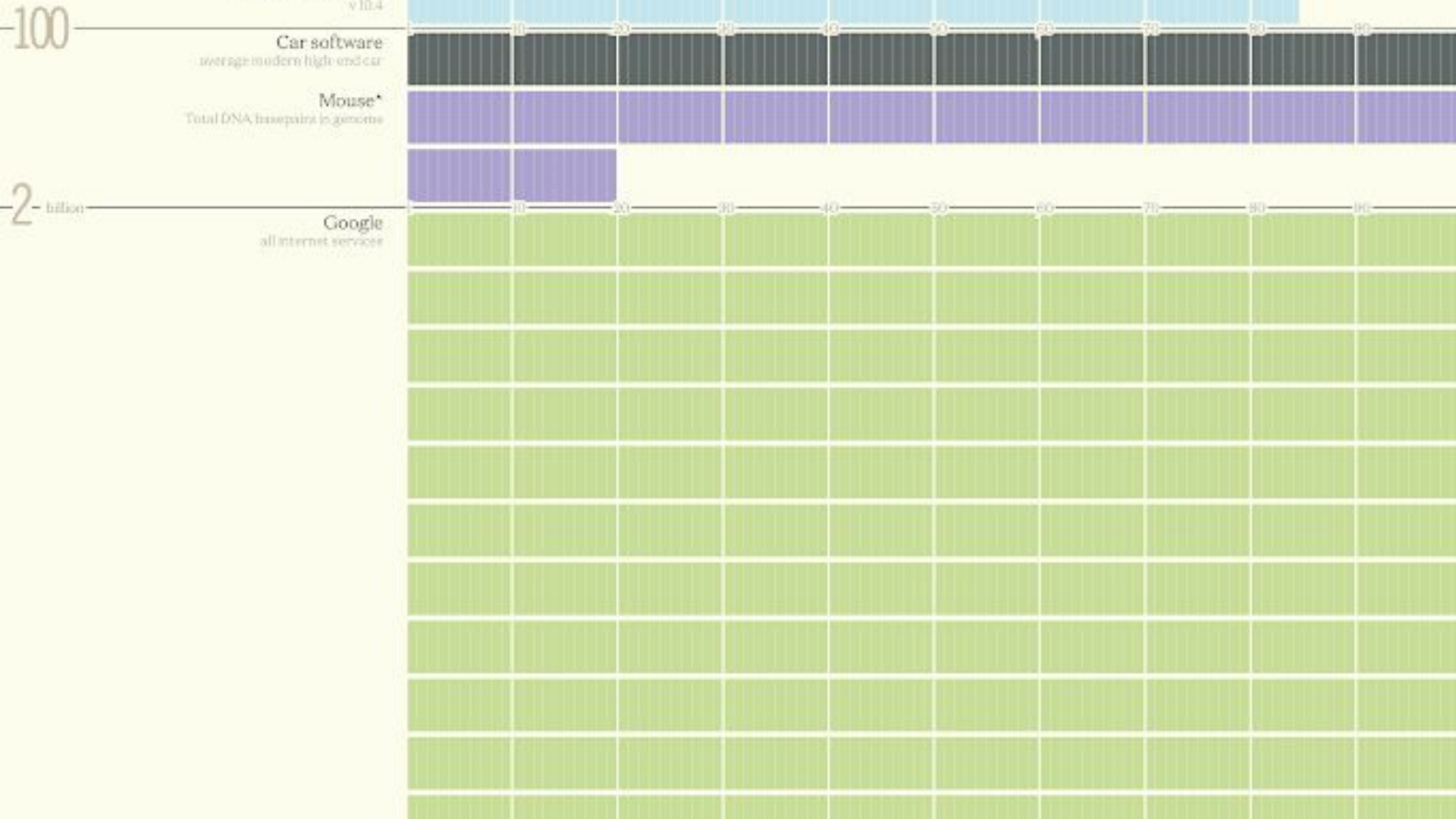

1
million
10
20
30
40
50
60
70
80
90
a million lines of code
38,000 pages of printed text
War And Peace x 14, or
Ulysses x 25, or
The Catcher in The Rye x 65
CryEngine 2
3D video game system
Bacteria
Syphilis (Treponema pallidum)
Age of Empires online
CESM Climate Model
National Center for Atmospheric Research
F-22 Raptor fighter jet
Linux Kernel 2.2.0
core code
Jurassic Park codebase
source: Dennis Nedry
Hubble Space Telescope
Unreal engine 3
3D video game system
Windows 3.1
1992
Large Hadron Collider
(root software)
ORGANISM
17.50%

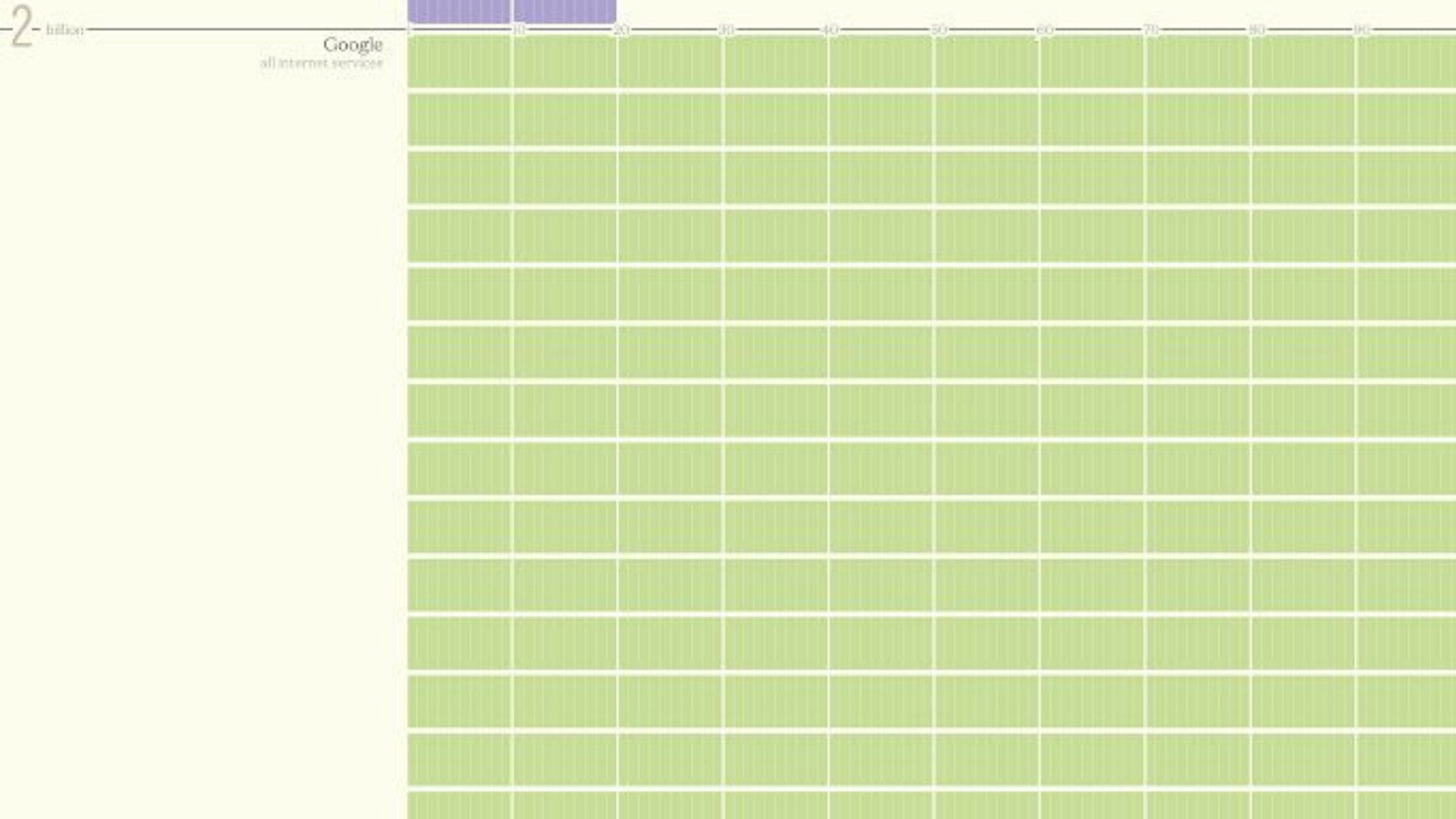

5
HD DVD Player on XBox
(just the player)
needed to repair HealthCare.gov
apparently
Mars Curiosity Rover
Martian ground vehicle probe
Linux kernel 2.6.0
2003
Google Chrome
latest
upper estimate
World of WarCraft
server only
Boeing 787
avionics & online support systems only
Windows NT 3.5
1993
Firefox
latest version
10
Chevy Volt
electric car
Intuit Quickbooks
accounting software
Windows NT 4.0
1996
Android
mobile device operating system
0
10
20
30
40
50
60
70
80
90
100%
285%

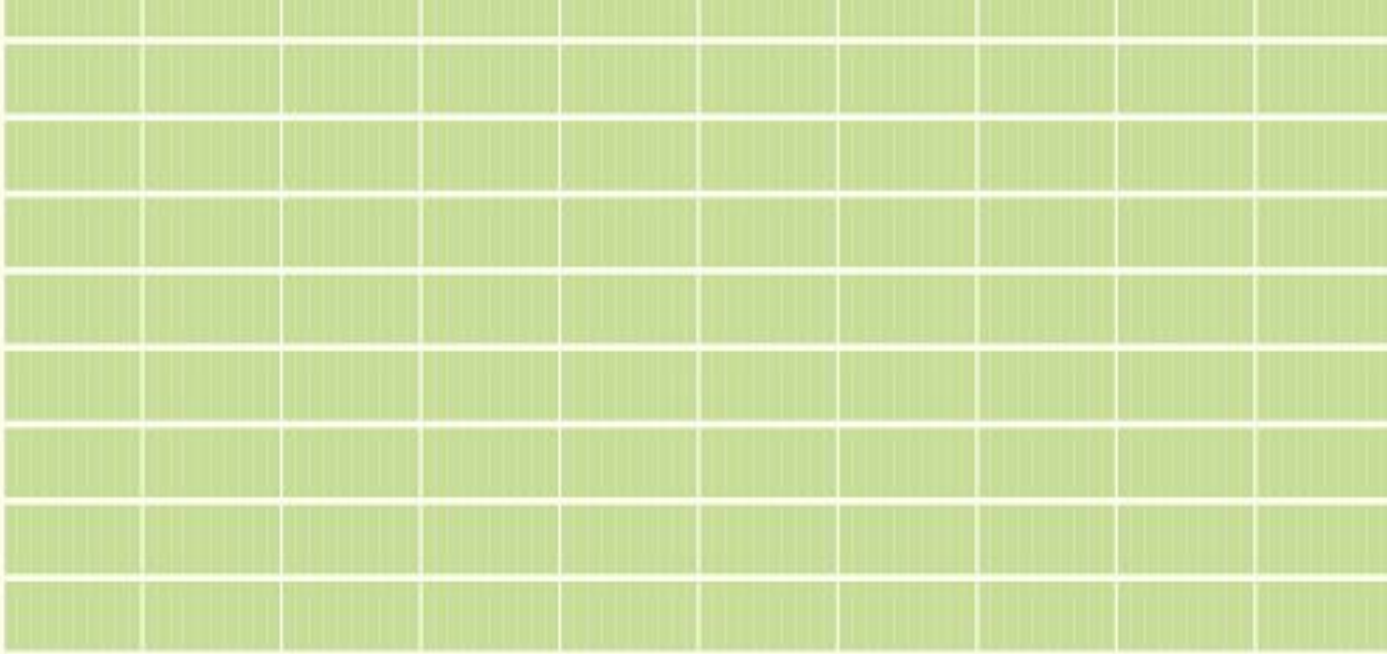

10
Chevy Volt
electric car
Intuit Quickbooks
accounting software
Windows NT 4.0
1996
Android
mobile device operating system
Mozilla Core
core code at heart of all Mozilla's software
MySQL
database language
Boeing 787
total flight software
Linux 3.1
recent version
Apache Open Office
open source office software
F-35 Fighter jet
2013
25
Microsoft Office 2001
Windows 2000
Microsoft Office for Mac
2006
0
20
30
40
50
60
70
80
90
100
288%
222%


25
Microsoft Office 2001
Windows 2000
Microsoft Office for Mac
2006
Symbian
mobile operating system
Windows 7
2009
Windows XP
2001
Microsoft Office 2013
50
Large Hadron Collider
total code
Windows Vista
2007
Microsoft Visual Studio 2012
Facebook
(including backend code)
US Army Future Combat System
fast battlefield network system (aborted)
Debian 5.0 codebase
free, open-source operating system


50
Large Hadron Collider
total code
Windows Vista
2007
Microsoft Visual Studio 2012
Facebook
(including backend code)
US Army Future Combat System
fast battlefield network system (aborted)
Debian 5.0 codebase
free, open-source operating system
Mac OS X "Tiger"
v 10.4
100
Car software
average modern high-end car
Mouse*
Total DNA basepairs in genome
2 billion
Google
all internet services
0
10
20
30
40
50
60
70
80
90


v 10.4
100
Car software
average modern high-end car
Mouse*
Total DNA basepairs in genome
2 billion
Google
all internet services


2 billion
Google
all internet services
1
10
20
30
40
50
60
70
80
90



*Human Genome = 3,300 billion "lines" of code

concept & design: David McCandless

informationisbeautiful.net

research: Pearl Doughty-White, Miriam Quick

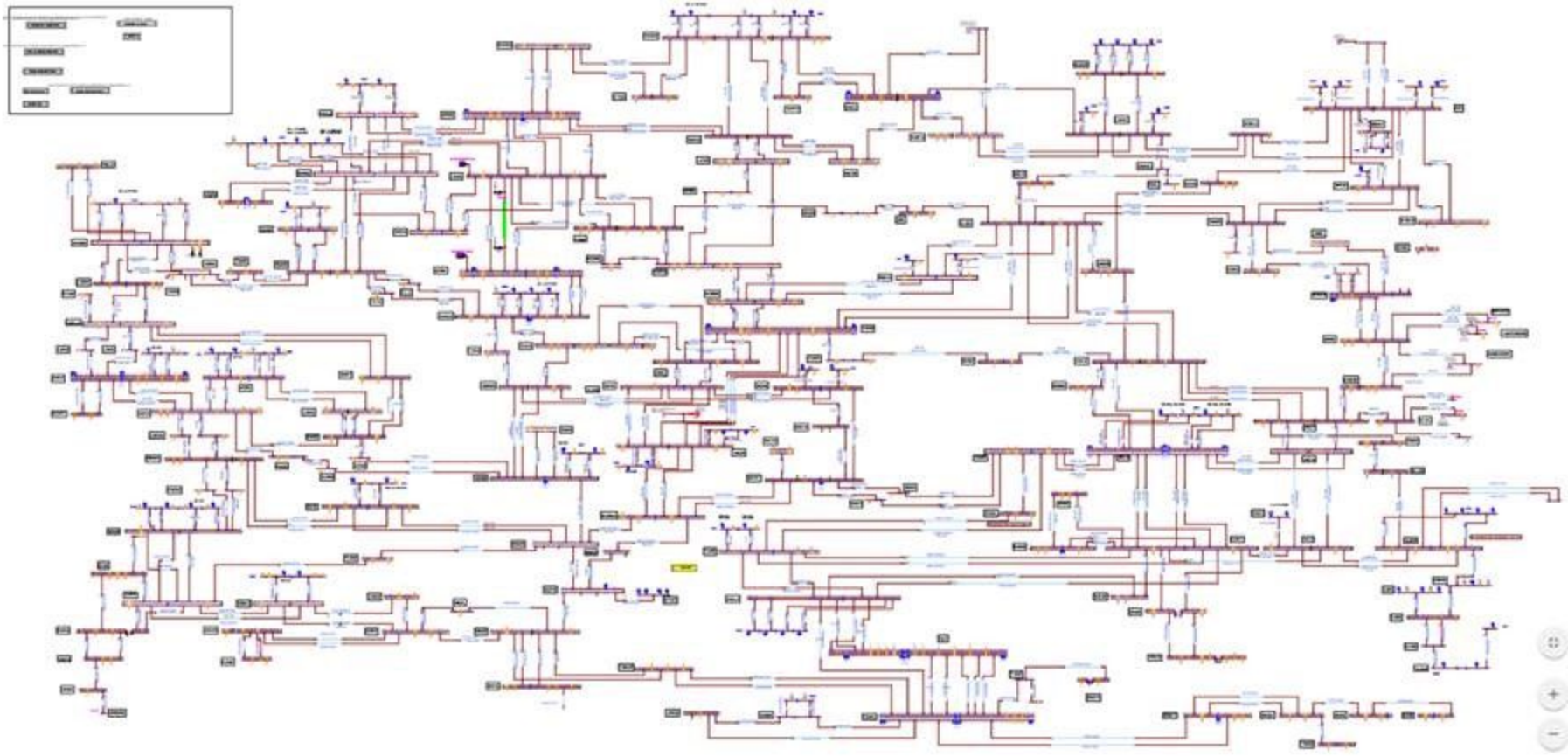
this graphic is a part of

knowledge is beautiful bit.ly/KIB_Books

sources NASA, Guard, Ohloh, Wired & press reports
note some guess work, rumours & estimates
data bit.ly/KIB_linescode

v 0.9 // Sep 24th 2015

Modern computer systems are insanely complex.



What is Computer Systems Security?

]HackingTeam[

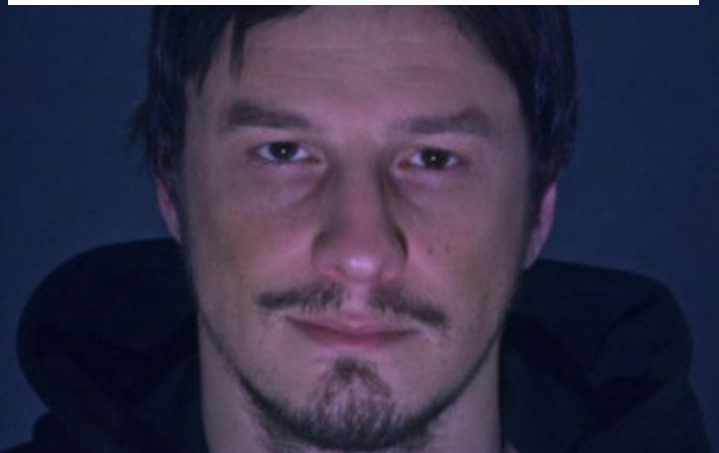
Rely on us.

]HackingTeam[

]HT[**Hacked Team**
@hackingteam

Since we have nothing to hide, we're publishing all our e-mails, files, and source code
mega.co.nz/#!Xx1IhChT!rbB…
infotomb.com/eyyxo.torrent

1:26am · 6 Jul 2015 · Twitter Web Client



Hacking the Hackers

Step 1: external reconnaissance

- Publicly available information
- LinkedIn
- Scanning tools
- Forensics on published content

http://pastebin.com/raw/0SNSvyjJ

Hacking the Hackers

Step 2: establish a bridgehead

- ~~Social engineering.~~
- ~~Buying access from previous exploiter.~~
- Exploitation of edge components.
  - Hacking Team had very little internet-facing infrastructure:
    - website
    - mail server
    - several routers
    - VPN appliance
    - spam filtering appliance
  - Phineas Fisher **found** and exploited a zero-day vulnerability in one of the dedicated devices.
  - Bridgehead established!

http://pastebin.com/raw/0SNSvyjJ

Hacking the Hackers

Step 3: internal reconnaissance

- Passive network listening.
- Slow active network scanning.
- This yielded an unsecured server with all the audio recording of the physical security system.

http://pastebin.com/raw/0SNSvyjJ

Hacking the Hackers

Step 4: expanding sphere of influence

- Fisher's scan found an accidentally-exposed backup system.
- From this, they retrieved the backup of the mail server.
- From this, they extracted the the password for a local admin account.
- This password was still valid on the currently-running mail server.
- This enabled two things:
  - disclosure of all Hacking Team email
  - recovery of passwords for other users on the mail server.
- One of these other users was a Windows Domain administrator...

http://pastebin.com/raw/0SNSvyjJ

Hacking the Hackers

Step 5: compromising the development network

- So far, Fisher was on the "production" network, but the source code was on the "development" network.
- Fisher used the stolen administrator passwords to migrate to the admins' workstations.
- Eventually, an admin decrypted a TrueCrypt volume containing a file full of system passwords, including one to the monitoring server.
- Fisher used this password, plus a vulnerability, to break into the monitoring server, which had access to both the production and development networks.
- A developers reused his (stolen) password between the production and development networks, allowing Fisher to to pull down all the source code.

http://pastebin.com/raw/0SNSvyjJ

Hacking the Hackers

Step 6: gloating

- Fisher reset the Twitter password of @hackingteam by using his control over their email, changed the name to "Hacked Team", and tweeted away.

http://pastebin.com/raw/0SNSvyjJ

Hacking Team Fallout

Hacking Team developed remote infiltration and surveillance products using undisclosed 0day vulnerabilities to target phones and PCs.

Fisher's leak revealed Hacking Team's business relationships with repressive regimes to spy on activists.

Huge embarrassment for the company, increased regulation by the Italian government, enormous loss of business.

http://pastebin.com/raw/0SNSvyjJ

What went wrong?

Many weak links!

- Vulnerable network-facing infrastructure (but what could they have done?)
- Physical security system without digital security.
- Failure to isolate backup storage system.
- Continued use of stale password on mail server (and probably others).
- Every-day user accounts that had Domain Administrator privileges.
- Known vulnerability in monitoring system.
- Link between production and development systems (probably necessary).
- Password reuse.
- Lack of two-factor authentication.

What would you have done differently, for your company?

Discussion: ethics of Hacking Team Hack?

Hacking Groundrules

RULE NUMBER ONE: Don't do anything illegal!

What does this mean in a hacking context?

- NEVER EVER EVER EVER EVER hack into a system that you do not have permission for.
- Never attempt to find vulnerabilities in a system that you do not own or have permission to audit.

How do you practice? (more on this later)

- Run a server for yourself to hack.
- Stick to software and services with *bug bounty programs*.
- Become an academic and take advantage of some (limited) exceptions.
- Become part of the competitive hacking community.

Bug Bounty Programs

- A number of web sites have started to offer Bug Bounty programs.
- They will give you money and/or fame in exchange for reporting security vulnerabilities to them.
- **Make sure that they also give you permission** (depending on the bug bounty program terms, you might have to explicitly register), and make sure you understand the rules and what is in scope.
    - violating these rules can disqualify you from the program and subject you to legal consequences

Google, Facebook, AT&T, Coinbase, Etsy, Github, Heroku, Microsoft, Paypal, https://bugcrowd.com/list-of-bug-bounty-programs

Google has rewards up to **$31,337**.

"I found a vuln! What's next?"

DO NOT EXPLOIT IT IN THE WILD.

What are your options?

1. Tell the world (**full disclosure**)
2. Tell the company/group responsible for the software (**responsible disclosure**)
3. Demonstrate it in a 0-day contest, such as pwn2own.
    a. Prizes up to $100,000 and much fame.
4. Sell the information to the grey or black market for $$$.
    a. Market price of a remote iPhone 0day: approximately $1,500,000.
    b. Market price of a *limited* Chrome 0day: approximately $200,000.

While this is a personal decision, #1 is becoming extremely rare.

100% Legal Hacker Professions

Penetration Tester

- Breaking into companies on their request.
- If Hacking Team had hired someone like Phineas Fisher to find their issues and report it to *them*, they'd have avoided quite a lot of pain. (But would it have found *all* the issues?)

In-house Security Engineer

- Working for a company to ensure security of their own systems.
- Example: Chrome Security team, Windows Security team, etc.

Industry Security Researcher

- Working for a company to (legally) identify security vulnerabilities in *other* software.
- Example: Google Project Zero, defense contractors

100% Legal Hacker Professions

Academic!

- Do research on new analysis, exploitation, mitigation techniques.
- Additional protections.

There is a HUGE supply issue of talented hackers.
This is reflected in job security and salaries.
Become a (legal and ethical) hacker!

Mostly Legal/Ethical Hacker Professions...

Hacking Team!

But how do I get good enough?

Capture the Flag: Live Cybersecurity Excercises

CTF is a style of *hacking contest*.

Fundamentally:

- Practice programs demonstrate discrete security issues.
- Hackers (or teams of hackers) compete to hack and/or protect them.

CTFs are everywhere:

- Every weekend.
- Online or in person (all-expense paid events in exotic locations)
- Prize money!

Many online resources:

- CTF directory, writeups, etc: https://ctftime.org
- List of practice "wargames": https://github.com/zardus/wargame-nexus

The Olympics of Hacking: DEF CON CTF

Takes place every year in Las Vegas.

Currently hosted by the Order of the Overflow (courtesy of yours truly!).

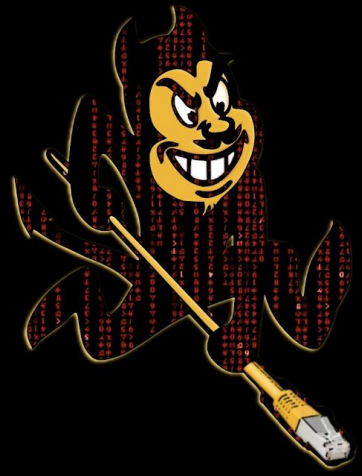
24 teams, with almost 1000 hackers attending!
- had to qualify for the competition through qualifier events

52 hours of non-stop hacking mayhem.

One winner (DEFKOR00T).


KING OF THE HILL
RPISEC


DEFKOROOT



PWNDEVILS

Who? The pwndevils, ASU's resident hacking club!

When? Tuesday 4:30pm and Thursday 4pm, and CTFs on weekends.

Where? BYENG 420

Web? pwndevils.com

But how do I get good enough - CLASS EDITION

Welcome to CSE 466!

This class...

- will teach you the building blocks of assessing Computer System Security.
- will be offense-focused.
- will take inspiration from CTF for homework assignments.
- will be INSANELY HARD.
- might be IMPOSSIBLE.

DROP THIS CLASS NOW.

Resources:

- The course website, with the syllabus, homeworks, etc, is http://pwn.college
- The course mailing list is cse466@googlegroups.com or http://groups.google.com/group/cse-466

Prerequisite Knowledge

You should know, or should be ready to learn on your own:

- Linux (on a very deep level, not just messing around on the commandline!).
- A scripting language (strong recommendation of Python).
- C (not just enough to write it, but enough to understand what the heck it's doing under the hood).
- x86 assembly (and be ready to learn many other architectures...)

Most importantly, you should be able to learn new concepts **VERY FAST** and with a high degree of independence and self-motivation.

Topic Roadmap

We will spend about a week on each of:

- Linux operating system fundamentals, program misuse and privilege escalation.
- Sandboxes and sandbox failures.
- Serialization vulnerabilities.
- Program reverse engineering.
- Traditional memory corruption (buffer overflows).
- Binary code injection.
- Advanced exploitation scenarios.
- Modern symmetric encryption security.
- Modern asymmetric encryption security.
- Content injection beyond binary code.
- Security of machine learning and AI.

These are *very diverse* topics, and you will need to learn them well enough to *hack them*.

This will largely be a class about learning on the fly.

Topic Plan

For each topic, we will have:

- Lecture introducing the topic.
- An example of a real-world systems compromise related to the topic.
- Usually about 50 to 100 challenges to hack, up to infinity.


WTF

Topic Plan

For each topic, we will have:

- Lecture introducing the topic.
- An example of a real-world systems compromise related to the topic.
- **Usually about 50 to 100 challenges to hack, up to infinity.**

This will happen *every week*.

Weekly Assignments

There will be one assignment (assigned at 8pm on Wednesday), and you will have until noon next wednesday to finish it.

For each assignment, you will need to solve some amount of a potentially infinite number of challenges.

Each challenge will give you some amount of points (this week, 1 point per chal). Up to 70 points, each point maps directly to your final grade (if you get 70 points on each assignment, you will get a C).

Past that, you will be graded on a curve, with the students with the most points getting 110% on the assignment and the student with the least (but over 70) getting 71%. 70 points or lower get X%, where X is the number of points.

Corner cases are handled (less than 40 students getting >70, etc).

Weekly Assignments

There is NO late submission. Every Wednesday at noon, the assignment goes offline.

Any availability issues with the assignments will be compensated by bonus points, not deadline extensions.

There will be NO make-up assignments.

Instead, because you can get up to 110% on each assignment, you can make up for poor performance on an assignment by rocking other assignments.

Two of the assignments will get demarcated as a "midterm" and a "final". They count the same as any other assignment but may be harder.

Your final grade is the average of all of your assignment percentages.

Bug Bounties

This class has a bug bounty program!

If you find a security issue in our infrastructure, and report it, we will give you between 5% and 50% percentage point bonus on the assignment, depending on our judgement of the severity of the flaw.

CAREFUL: if you report spurious non-issues as issues, we will give you **up to -15% for wasting time**.

The stick: if you instead *abuse* the issue, you will have violated ASU policy and will face the consequences.

General rule of thumb: anything inside the homework Docker containers, VMs, etc is fair game (all issues are intentional). The homework server itself is not.

THIS CLASS WILL BE INSANELY HARD!!!!!!!!!!!!

... but if you survive, you will be performing, at least on the small scale, at the level of a "proficient" hacker.

Prepare Your Kit

I **HIGHLY** recommend you switch to Linux on your laptop for this course.

At the **VERY** least, you will need a fully-functional Linux VM.

Bring your laptop to class every week. We will have some challenges for some homework assignments that will *have* to be completed in class (not today).

CSE 466 Week 1

Linux Trickery

You should already know everything I am about to review. If you do not, **drop this course for your own sake**.

What is Linux?

You need to already know this.

Important Linux Concepts You SHOULD ALREADY KNOW

Filesystem layout.

Users, permissions, and capabilities.

OS-userspace interaction (system calls).

Process lifecycle (setuid, suid bits, SECCOMP, etc).

"Everything is a file (descriptor)."

How to use "man".

Homework 1

What happens if just one program has a privilege exploitation vulnerability? We'll model this with the SUID file permission bit.

The homework:

- ssh to hw1@cse466.pwn.college (the password will be posted on the class mailing list)
- enter your hacker alias (make it cool, but appropriate!) and ASU ID.
- Choose a single binary to make SUID.
- Use that single binary to read the "/flag" file (which is only readable by root).
- Log out ("exit" command).
- Provide the flag when asked.
- Scoring is done automatically.
- 1 point per unique binary.