Motivation to Software Security

Segurança em Software

Pedro Adão, Ana Matos

TÉCNICO LISBOA

Computer Security in Real World

After thirty years of work on computer security, why are almost all the systems in service today extremely vulnerable to attack? The main reason is that security is expensive to set up and a nuisance to run, so people judge from experience how little of it they can get away with.

Butler Lampson

B. W. Lampson, "Computer security in the real world," in *Computer*, vol. 37, no. 6, pp. 37-46, June 2004.

Computer Security in Real World

Since there's been little damage, people decide that they don't need much security. In addition, setting it up is so complicated that it's hardly ever done right. While we await a catastrophe, simpler setup is the most important step toward better security.

Butler Lampson

B. W. Lampson, "Computer security in the real world," in *Computer*, vol. 37, no. 6, pp. 37-46, June 2004.

The Real Security Game

Notable among them are the subject/object access matrix model, access control lists, multilevel security using information flow, public key cryptography, and cryptographic protocols. In spite of these successes, it seems fair to say that in an absolute sense, the security of the hundreds of millions of deployed computer systems is terrible: a ***determined and competent attacker*** *could* ***destroy*** *most of the information on almost any of these systems, or* ***steal*** *it from any system that is connected to a network. Even worse, the attacker could do this to* ***millions of systems at once****.*

Butler Lampson

B. W. Lampson, "Computer security in the real world," in *Computer*, vol. 37, no. 6, pp. 37-46, June 2004.

The problem is Software

"*Behind every computer security problem and malicious attack lies a common enemy -- bad software.*"

J. Viega & G. McGraw
Building Secure Software, 2002

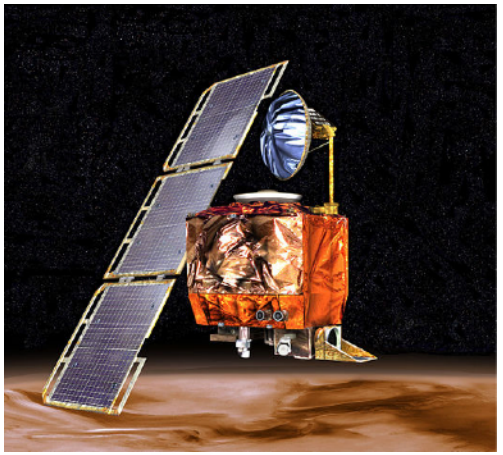
The problem is Software

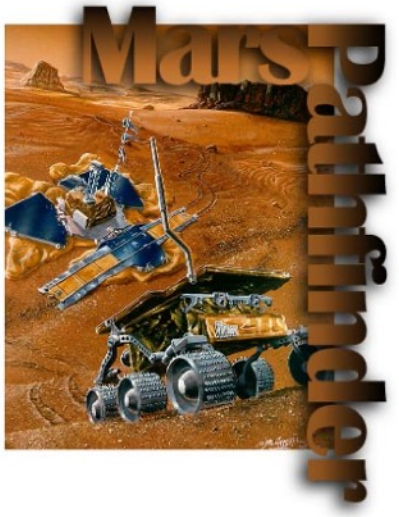
- “We wouldn’t have to spend so much (…) on network security if we didn’t have such bad software security.
- Think about the most recent security vulnerability about which you’ve read.
- Maybe it’s a killer packet that allows an attacker to crash some server (…)
- Maybe it’s one of the gazillions of buffer overflows that allow an attacker to take control of a computer (…)
- Maybe it’s an encryption vulnerability that allows an attacker to read an encrypted message (…)
- These are all software issues.”

J. Viega & G. McGraw
Building Secure Software, 2002

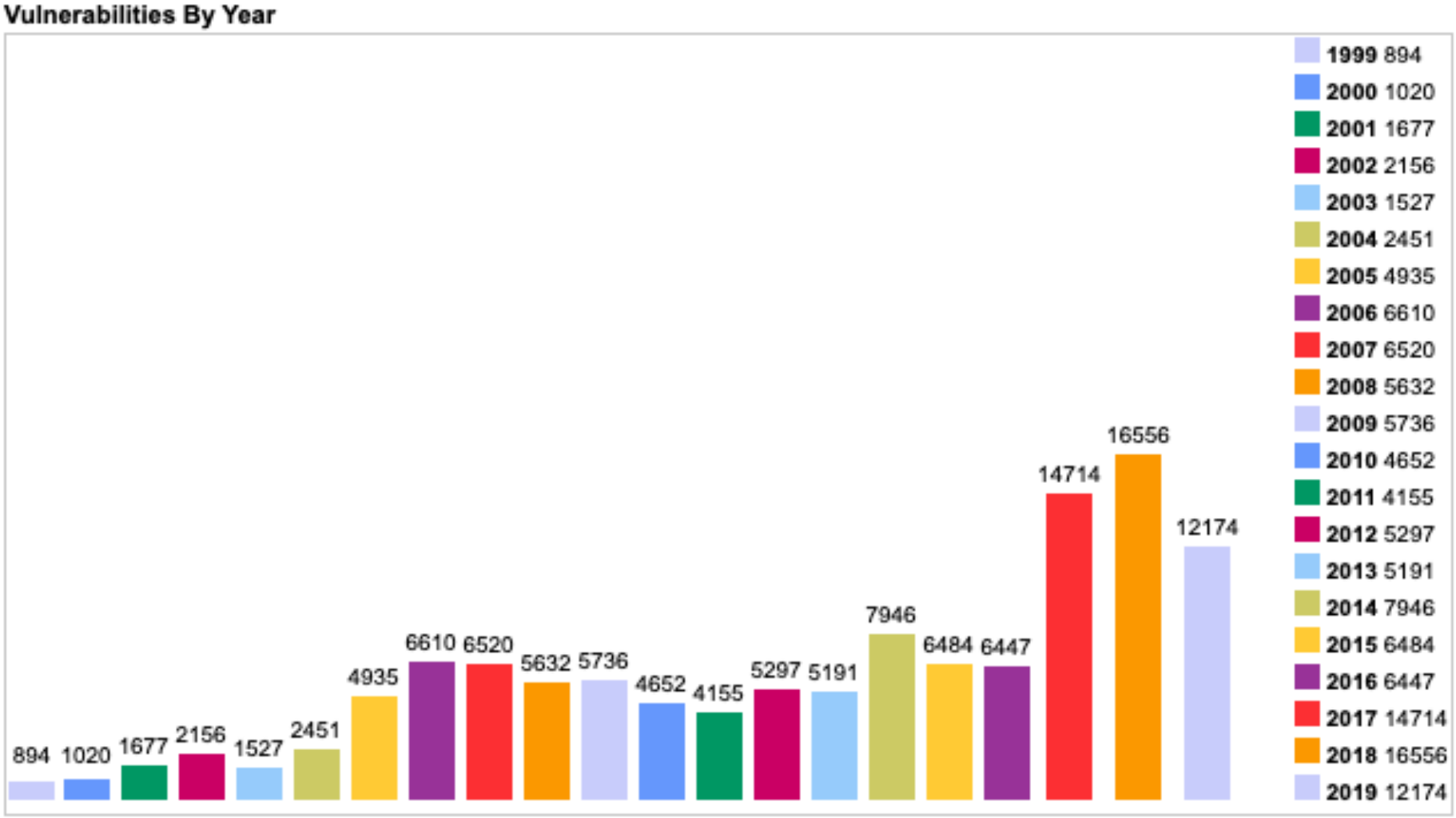
Bad software is everywhere

- NASA Mars Climate Orbiter
  - Crashed due to a units conversion bug ($165 million)
- NASA Mars Pathfinder
  - Stopped for several hours due to a priority-inversion bug ($265 million)
- Risks Forum: http://www.risks.org/
- http://www.seguranca-informatica.net





Reported Vulnerabilities - Evolution



https://www.cvedetails.com/

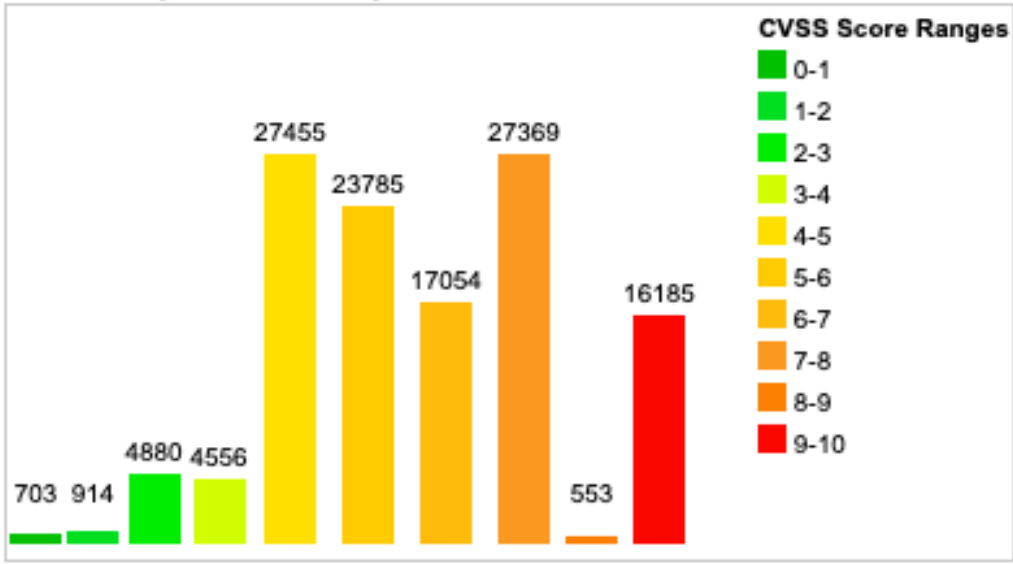
Reported Vulnerabilities - Severity

Distribution of all vulnerabilities by CVSS Scores

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	703	0.60
1-2	914	0.70
2-3	4880	4.00
3-4	4556	3.70
4-5	27455	22.20
5-6	23785	19.30
6-7	17054	13.80
7-8	27369	22.20
8-9	553	0.40
9-10	16185	13.10
Total	123454	

Weighted Average CVSS Score: **6.6**

Vulnerability Distribution By CVSS Scores



https://www.cvedetails.com/

Vulnerabilities and more vuln…

- US-CERT Technical Cyber Security Alerts
  - http://www.us-cert.gov/cas/techalerts/
-

US-CERT
UNITED STATES COMPUTER EMERGENCY READINESS TEAM

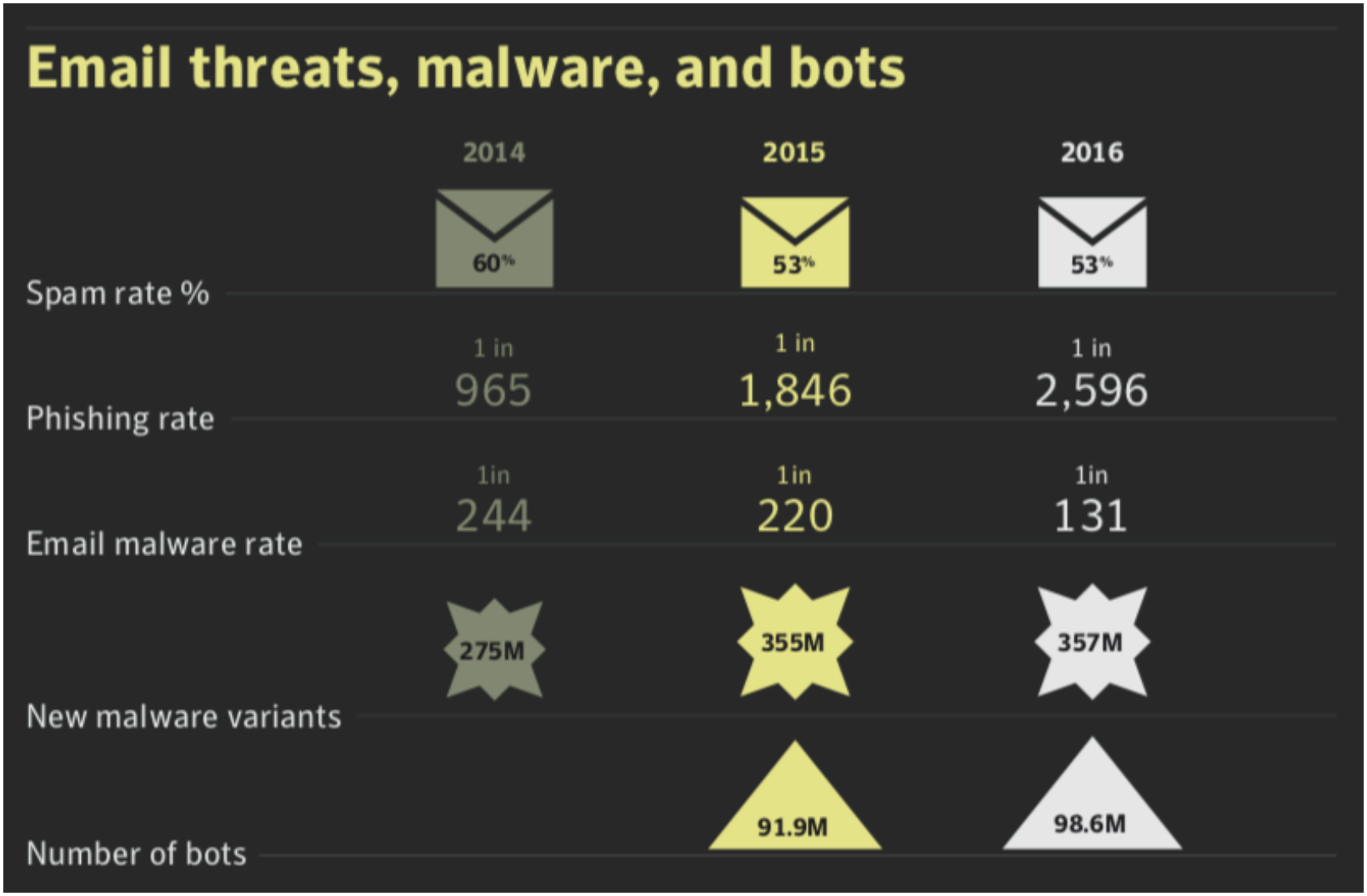
2018 | 2017 | 2016 | 2015 | 2014 | 2013 | 2012 | 2011 | 2010 | 2009 | 2008

- TA18-201A : Emotet Malware
- TA18-149A : HIDDEN COBRA – Joanap Backdoor Trojan and Brambul Server Message Block Worm
- TA18-145A : Cyber Actors Target Home and Office Routers and Networked Devices Worldwide
- TA18-141A : Side-Channel Vulnerability Variants 3a and 4
- TA18-106A : Russian State-Sponsored Cyber Actors Targeting Network Infrastructure Devices
- TA18-086A : Brute Force Attacks Conducted by Cyber Actors
- TA18-074A : Russian Government Cyber Activity Targeting Energy and Other Critical Infrastructure Sectors
- TA18-004A : Meltdown and Spectre Side-Channel Vulnerability Guidance
- TA17-318B : HIDDEN COBRA – North Korean Trojan: Volgmer
- TA17-318A : HIDDEN COBRA – North Korean Remote Administration Tool: FALLCHILL
- TA17-293A : Advanced Persistent Threat Activity Targeting Energy and Other Critical Infrastructure Sectors
- TA17-181A : Petya Ransomware
- TA17-164A : HIDDEN COBRA – North Korea's DDoS Botnet Infrastructure
- TA17-163A : CrashOverride Malware
- TA17-156A : Reducing the Risk of SNMP Abuse
- TA17-132A : Indicators Associated With WannaCry Ransomware
- TA17-117A : Intrusions Affecting Multiple Victims Across Multiple Sectors
- TA17-075A : HTTPS Interception Weakens TLS Security

Vulnerabilities and more vuln…

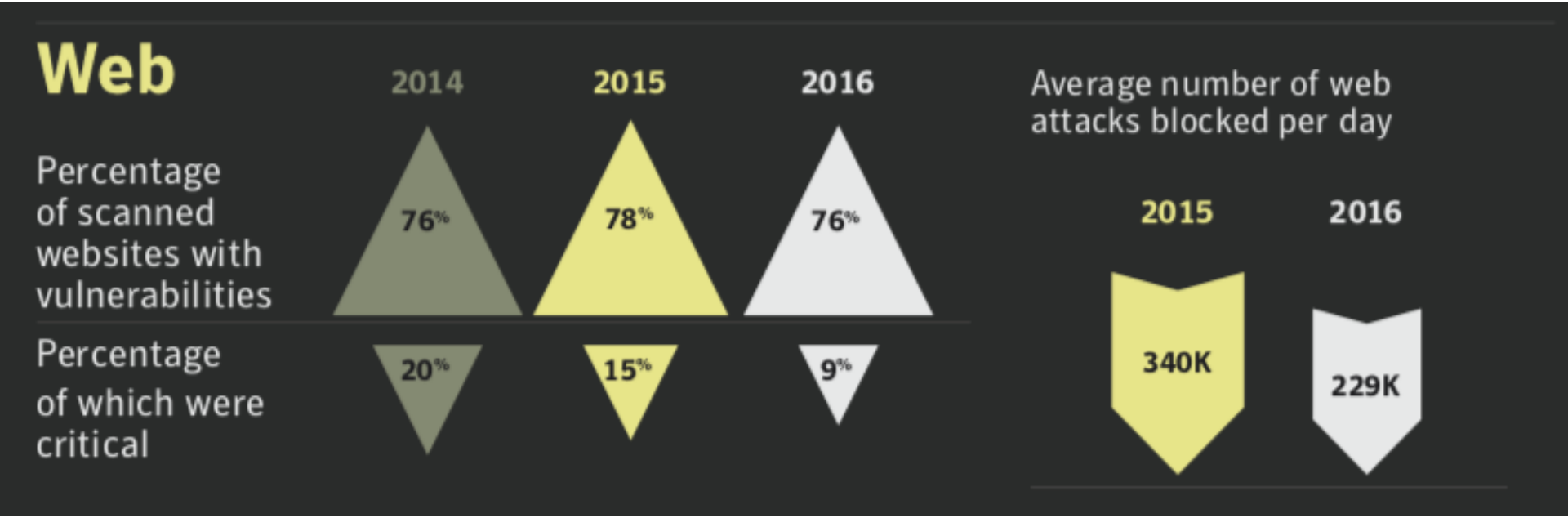
- Alert TA15-195A: Adobe Flash & Microsoft Windows Vulnerabilities
  - *Adobe Flash use-after-free and memory corruption vulnerabilities (CVE-2015-5119, CVE-2015-5122, CVE-2015-5123)*
    - Adobe Flash Player contains critical vulnerabilities within the ActionScript 3 ByteArray, opaqueBackground and BitmapData classes. Exploitation of these vulnerabilities could allow a remote attacker to execute arbitrary code on a vulnerable system.
  - *Microsoft Windows Adobe Type Manager privilege escalation vulnerability (CVE-2015-2387)*
    - The Adobe Type Manager module contains a memory corruption vulnerability, which can allow an attacker to obtain system privileges on an affected Windows system. The Adobe Type Manager is a Microsoft Windows component present in every version since NT 4.0. The primary impact of exploiting this vulnerability is local privilege escalation.

Security in Numbers



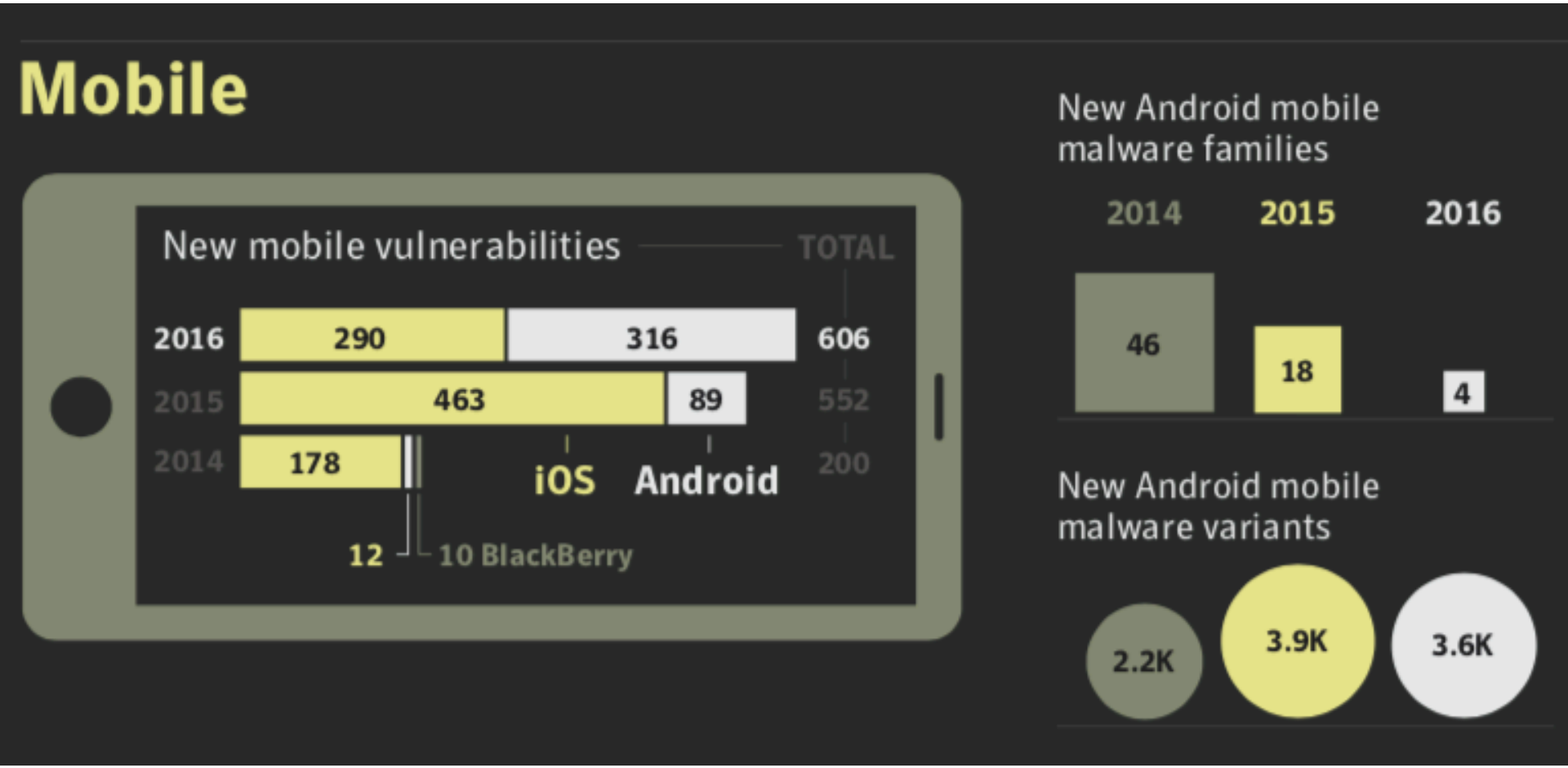
Source: *Internet Security Threat Report Symantec 2017*

Security in Numbers



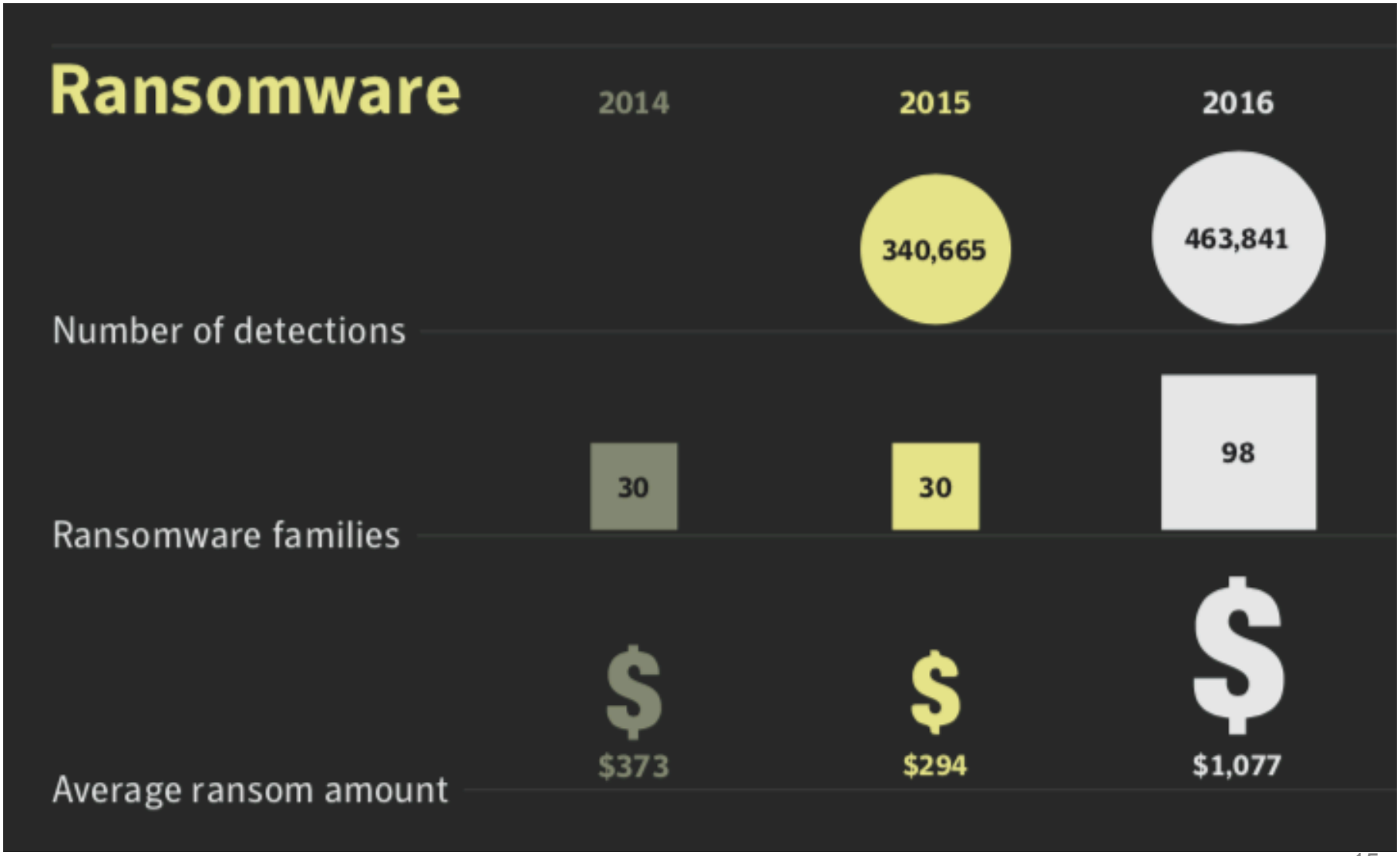
Source: *Internet Security Threat Report Symantec 2017*

Security in Numbers



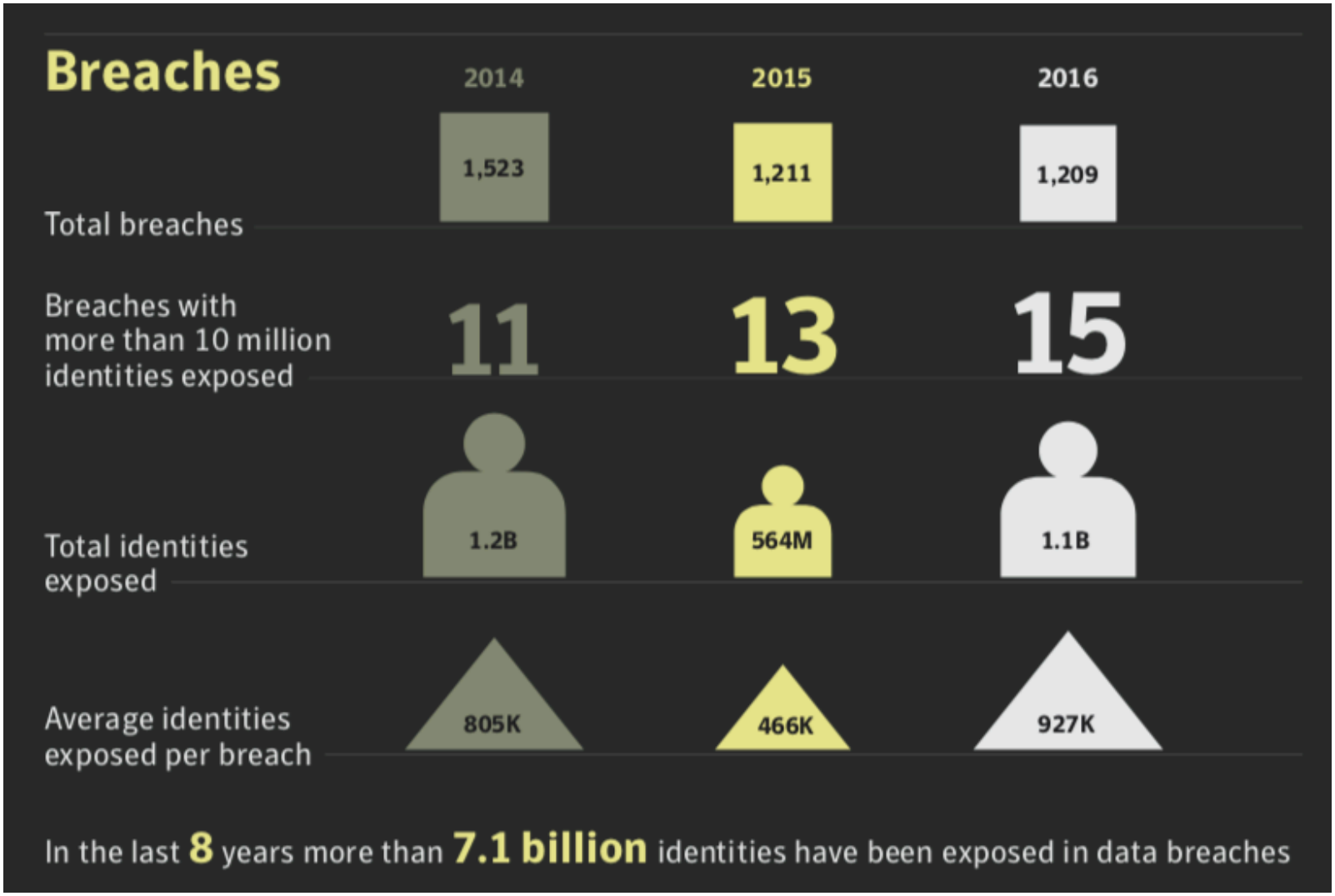
Source: *Internet Security Threat Report Symantec 2017*

Security in Numbers



Source: *Internet Security Threat Report Symantec 2017*

Security in Numbers



Source: *Internet Security Threat Report Symantec 2017*

Source: *Internet Security Threat Report Symantec 2017*

Example: energy companies (2017)

Symantec Official Blog

Dragonfly: Western energy sector targeted by sophisticated attack group

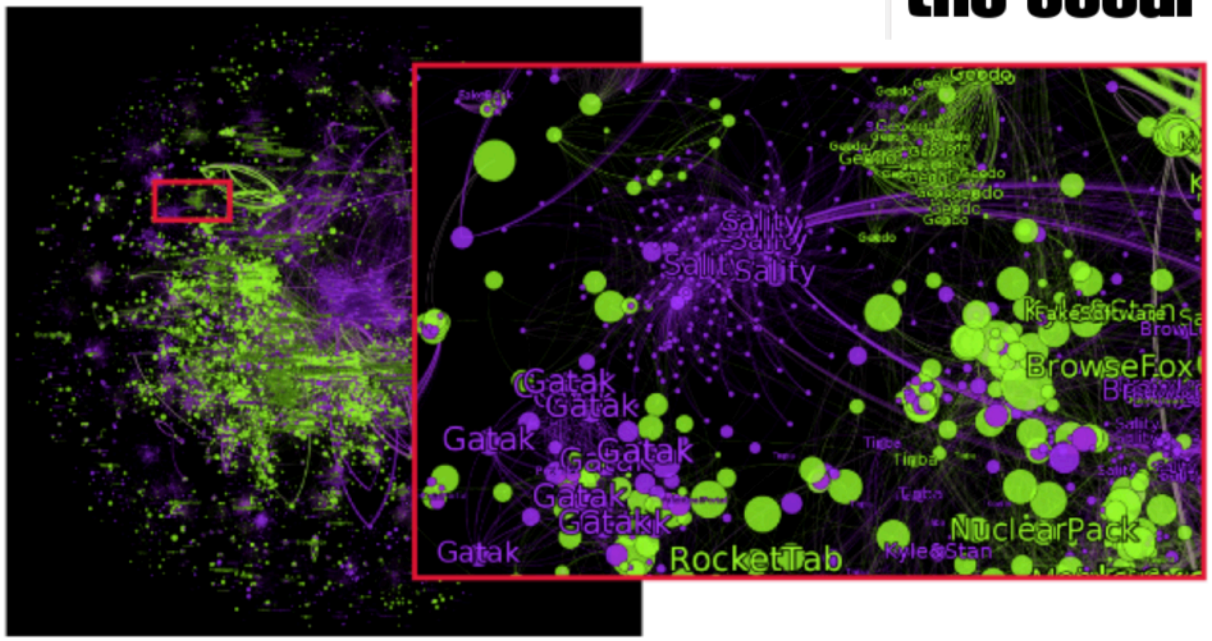
Resurgence in energy sector attacks, with the potential for sabotage, linked to re-emergence of Dragonfly cyber espionage group

Example: financial firms (2015)

Study: Financial Firms Hit Hard By Targeted Attacks

POSTED BY: PAUL JUNE 25, 2015 12:16 0 COMMENTS

the security ledger



A graphic shows threats targeting financial industry organizations vs. non financial services focused attacks in purple. (Image courtesy of Websense/Raytheon.)

In-brief: A new report from the firm Websense finds that financial services firms are being hit hard by cyber attacks, including targeted attacks aimed at luring employees into installing malicious software on corporate networks.

Example: healthcare systems (2015)

COMPUTERWORLD Popular Now:

Home > Vertical IT > Healthcare IT

NEWS

More than 80% of healthcare IT leaders say their systems have been compromised

Credit: Thinkstock

Only half of IT managers feel they are adequately prepared to prevent future attacks

MORE LIKE THIS

Healthcare IT makes improved customer service an urgent priority

Top security tools in the fight against cybercrime

The security data and survey directory

on IDG Answers

How is data stored and accessed in the cloud?

Example: Ashley Madison (2015)

Infidelity site Ashley Madison hacked as attackers demand total shutdown

Site's hackers claim 37m personal records have been stolen from notorious dating site, with Cougar Life and Established Men also compromised

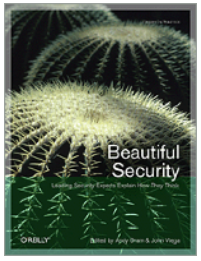
The Ashley Madison website. Photograph: Screengrab

theguardian

The problem is Software

- “the current state of security in commercial software is rather distasteful,
- marked by embarrassing public reports of vulnerabilities and actual attacks,
- scrambling among developers to fix and release patches,
- and continual exhortations to customers to perform rudimentary checks and maintenance.”

Jim Routh, Forcing Firms to Focus: Is Secure Software in Your Future, in Beautiful Security, O'Reilly, 2010



Industry's fault?

- "Software buyers are literally crash test dummies
- for an industry that is remarkably insulated against liability, accountability, and responsibility
- for any harm, damages or loss that should occur because of manufacturing defects or weaknesses
- that allow cyber attackers to break into and hijack our computer systems."
  - David Rice. "Geekonomics: The Real Cost of Insecure Software", Addison-Wesley, 2007


Exploit
Wednesday

Universities' fault?

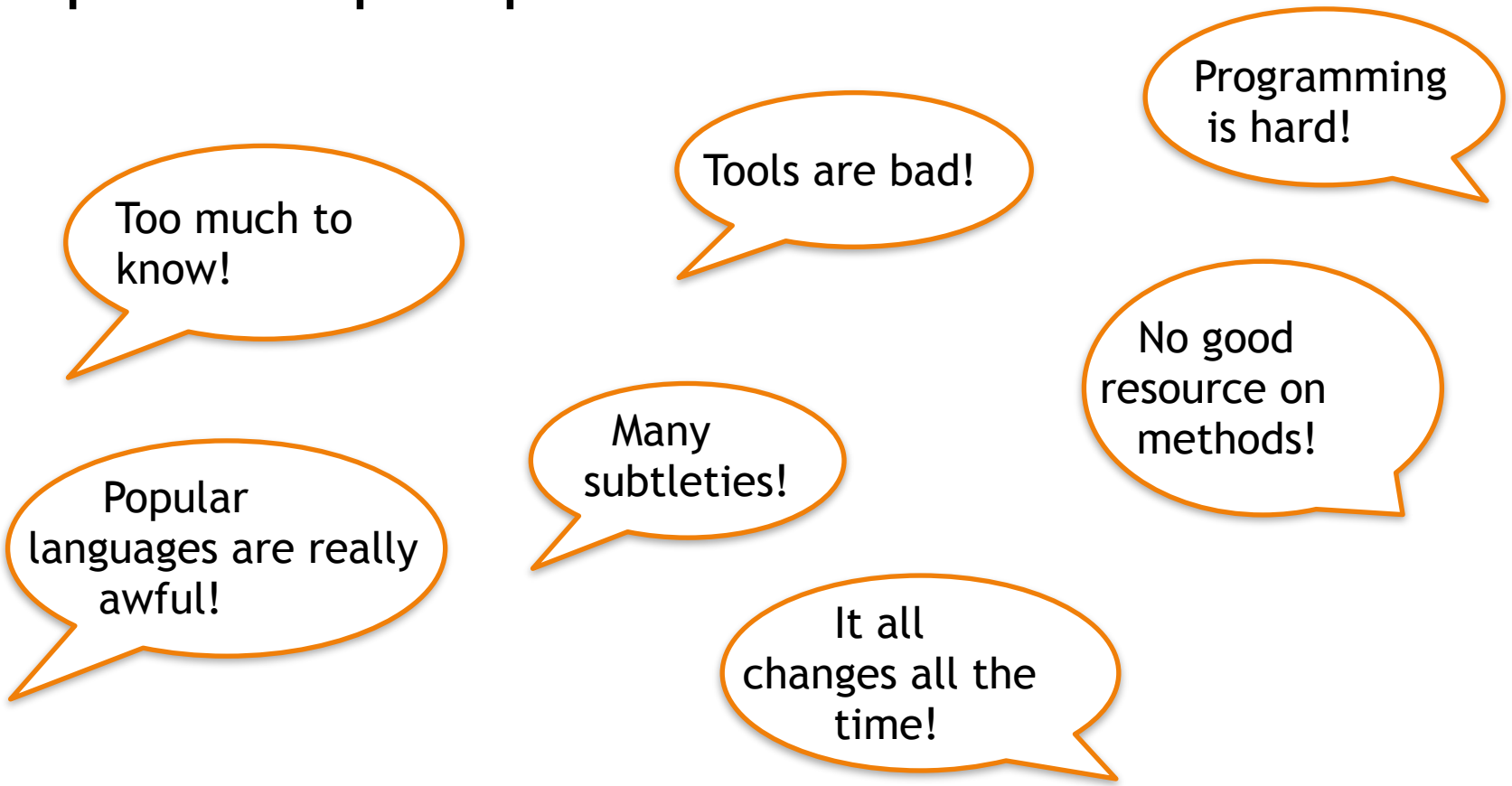
- "We at Oracle have (...) determined that most developers we hire have not been adequately trained in basic secure coding principles (…)
- We have therefore had to develop and roll out our own in-house security training program at significant time and expense. (...)
- In the future, Oracle plans to give hiring preference to students who have received such training and can demonstrate competence in software security principles."
  - Mary Ann Davidson, Oracle's Chief Security Officer

Security: a large industry

- “Global Cyber Security spending 2011: $60 billion”
  - PWC, Cyber Security M&A review, Nov. 2011
- “With a cumulative market valued at $65.5 billion (2013 – 2018), the U.S. Federal Cybersecurity market will grow steadily – at about 6.2% CAGR over the next six years.”
  - U.S. Federal Cybersecurity Market Forecast 2013-2018, April 2013
- RSA Conference 2012 S. Francisco
  - 562 speakers, 346 sessions, 350+ companies in the exhibition, several thousand attendees

Why is Software Security hard?

Superficial perspective:



Why is Software Security hard?

A more insightful perspective:

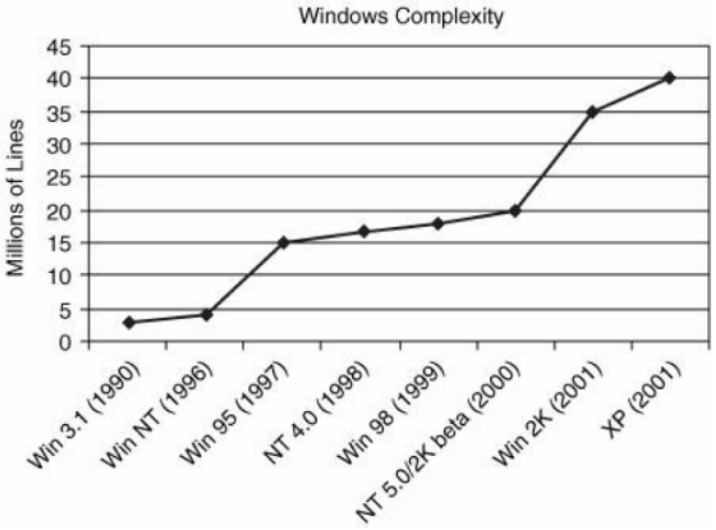
3 trends in modern computing systems makes them more susceptible to security problems, and makes it harder to secure them:

The trinity of trouble

- Complexity
- Extensibility
- Connectivity

Complexity

- Attacks exploit *bugs* called vulnerabilities
  - estimated 5-50 bugs per Klines of code (~5 if rigorous quality assurance)
  - examples (number of lines of code):
    - Solaris7 500K
    - Windows 95 <5M
    - Windows XP 45M
    - Windows Server 2003 50M
    - Linux kernel 2.6.0 5.2M
    - Linux kernel 3.6 15.9M



- Trend of growth in the size and complexity of software systems
  - The code base can grow (in executable space) even when the source code base appears to be small

Extensibility

- Current SW is inherently extensible:
  - Updates, extensions that increment on the functionality of systems
  - Device drivers, plug-ins, extensions, modules, Apps,....
  - Virtual machines and mobile code (JavaScript, Java, .NET, Flash,…)
  - Combination of several components and forms of code execution (web apps)
- Problems:
  - What is “the software”? How do you ensure its security?
  - Mobile malicious code: Worms, Virus,…

Connectivity

- Internet *(PCs, smartphones, tablets,…)*
  - Small failures can propagate widely
  - Allows automated attacks
    - Major worms 99-04: Melissa, ILOVEYOU, Code Red, Sircam, SQL Hammer, Blaster, Sobig, Mydoom, Sasser, Witty
    - DDoS attacks (400 Gbps record, Feb. 2014)
  - Economic risk
    - SWIFT net connects 10000+ financial institutions and moves zillions of dollars daily; targeted attacks at banks
  - Distance and feeling of safety

Connectivity: Internet of Things

- Not only cable and WiFi but also GSM/3G/4G/5G, Bluetooth, NFC,...
- http://www.darkreading.com/risk/cartoon-what-your-toaster-now-needs-/d/d-id/1269204

The 4th trouble: motivation

- Theft and extorsion — unrightfully obtaining financial property
  - homebanking, credit cards, blackmailing…
- Espionage — obtaining intelligence for strategic advantage between nation, corporations
- War and Terrorism — causing damage by exploiting tensions within another society, or as a weapon, or with the intent of pressuring towards a political agenda
  - Estonia, Georgia, Ucrania…

(Near) future of SW

(Near) future of SW

- More components

(Near) future of SW

- More components
- More frameworks, more combination of binary and executed code

(Near) future of SW

- More components
- More frameworks, more combination of binary and executed code
- More wireless

(Near) future of SW

- More components
- More frameworks, more combination of binary and executed code
- More wireless
- More mobile devices and embedded “things”

(Near) future of SW

- More components
- More frameworks, more combination of binary and executed code
- More wireless
- More mobile devices and embedded “things”
- More distribution

(Near) future of SW

- More components
- More frameworks, more combination of binary and executed code
- More wireless
- More mobile devices and embedded “things”
- More distribution
- More mobile code

(Near) future of SW

- More components
- More frameworks, more combination of binary and executed code
- More wireless
- More mobile devices and embedded “things”
- More distribution
- More mobile code
- Subscription services

(Near) future of SW

- More components
- More frameworks, more combination of binary and executed code
- More wireless
- More mobile devices and embedded “things”
- More distribution
- More mobile code
- Subscription services
- *More complexity, extensibility, connectivity*

Questions raised in this course

- What forms of software security vulnerabilities are there and how can they be exploited?
- How can we develop secure software?
- What are the fundamental mechanics behind the vulnerabilities?
- How can we design techniques and tools to prevent or fix them?

Objectives

Objectives

- To gain a deep insight of the **mechanics and underlying causes** of software security problems in modern computing systems

Objectives

- To gain a deep insight of the **mechanics and underlying causes** of software security problems in modern computing systems
- To become acquainted with **guidelines, techniques and tools** that can help prevent or detect software security problems

Objectives

- To gain a deep insight of the **mechanics and underlying causes** of software security problems in modern computing systems
- To become acquainted with **guidelines, techniques and tools** that can help prevent or detect software security problems
- To understand the fundamental principles involved in the **construction of secure by design mechanisms** for tackling these problems

Goals of this course

Goals of this course

- To Understand Software Vulnerabilities

Goals of this course

- To Understand Software Vulnerabilities
  - *To know the problem*

Goals of this course

- To Understand Software Vulnerabilities
  - *To know the problem*
- To Develop Secure Software

Goals of this course

- To Understand Software Vulnerabilities
  - *To know the problem*
- To Develop Secure Software
  - *To apply solutions to our problem*

Goals of this course

- To Understand Software Vulnerabilities
  - *To know the problem*
- To Develop Secure Software
  - *To apply solutions to our problem*
- To Design correct Enforcement Mechanisms

Goals of this course

- To Understand Software Vulnerabilities
  - *To know the problem*
- To Develop Secure Software
  - *To apply solutions to our problem*
- To Design correct Enforcement Mechanisms
  - *To create new solutions for our problem*

Structure of the course

Structure of the course

Intro

Panorama

Computer security principles

Structure of the course

Intro	**Part I Understanding Software Vulnerabilities**
Panorama	Conventional applications
Computer security principles	Web applications and databases
	Client-side security

Structure of the course

Intro	Part I Understanding Software Vulnerabilities	Part II Development of Secure Software
Panorama	Conventional applications	Software auditing
Computer security principles	Web applications and databases	Validation and encoding
	Client-side security	

Structure of the course

Intro	Part I Understanding Software Vulnerabilities	Part II Development of Secure Software	Part III Design of enforcement mechanisms
Panorama	Conventional applications	Software auditing	Security policies and properties
Computer security principles	Web applications and databases	Validation and encoding	Static security analysis
	Client-side security		Dynamic security analysis

Structure of the course

Intro	Part I Understanding Software Vulnerabilities	Part II Development of Secure Software	Part III Design of enforcement mechanisms	Part IV A Case Study: Java Security
Panorama	Conventional applications	Software auditing	Security policies and properties	Java security environment
Computer security principles	Web applications and databases	Validation and encoding	Static security analysis	Secure programming in Java
	Client-side security		Dynamic security analysis	

Structure of the course

Intro	Part I Understanding Software Vulnerabilities	Part II Development of Secure Software	Part III Design of enforcement mechanisms	Part IV A Case Study: Java Security
Panorama	Conventional applications	Software auditing	Security policies and properties	Java security environment
Computer security principles	Web applications and databases	Validation and encoding	Static security analysis	Secure programming in Java
	Client-side security		Dynamic security analysis	

Know the problem

Structure of the course

Intro	Part I Understanding Software Vulnerabilities	Part II Development of Secure Software	Part III Design of enforcement mechanisms	Part IV A Case Study: Java Security
Panorama	Conventional applications	Software auditing	Security policies and properties	Java security environment
Computer security principles	Web applications and databases	Validation and encoding	Static security analysis	Secure programming in Java
	Client-side security		Dynamic security analysis	

Know the problem

Know how to apply solutions

Structure of the course

Intro	Part I Understanding Software Vulnerabilities	Part II Development of Secure Software	Part III Design of enforcement mechanisms	Part IV A Case Study: Java Security
Panorama	Conventional applications	Software auditing	Security policies and properties	Java security environment
Computer security principles	Web applications and databases	Validation and encoding	Static security analysis	Secure programming in Java
	Client-side security		Dynamic security analysis	

Know the problem

Know how to apply solutions

Know how to create solutions