Security incidents, weaknesses and vulnerabilities

Martin Stanek

Department of Computer Science
Comenius University
stanek@dcs.fmph.uniba.sk

Security of IT infrastructure (2017/18)

Content

Vulnerabilities

Real world

- Statistics, surveys
- Controls, regulatory and compliance frameworks

Security incidents

- Data breaches
- Other incidents

Examples of vulnerabilities – technical details

Introduction

Security incidents and failures

- various causes (or their combination): human factor, criminal activities, technical vulnerabilities etc.
- impact: “nothing” happened, loss of reputation, cost of repair/replacement of data and systems, direct financial loss, bankruptcy etc.

mostly technical failures/vulnerabilities in this lecture

- just examples . . . reality is worse (unpublished vulnerabilities, weak passwords, misconfiguration, etc.)
- National Vulnerability Database (nvd.nist.gov)
- various other sources exist
  - more sources and vulnerabilities covered, faster publication, additional detail (e.g. how to fix), . . .

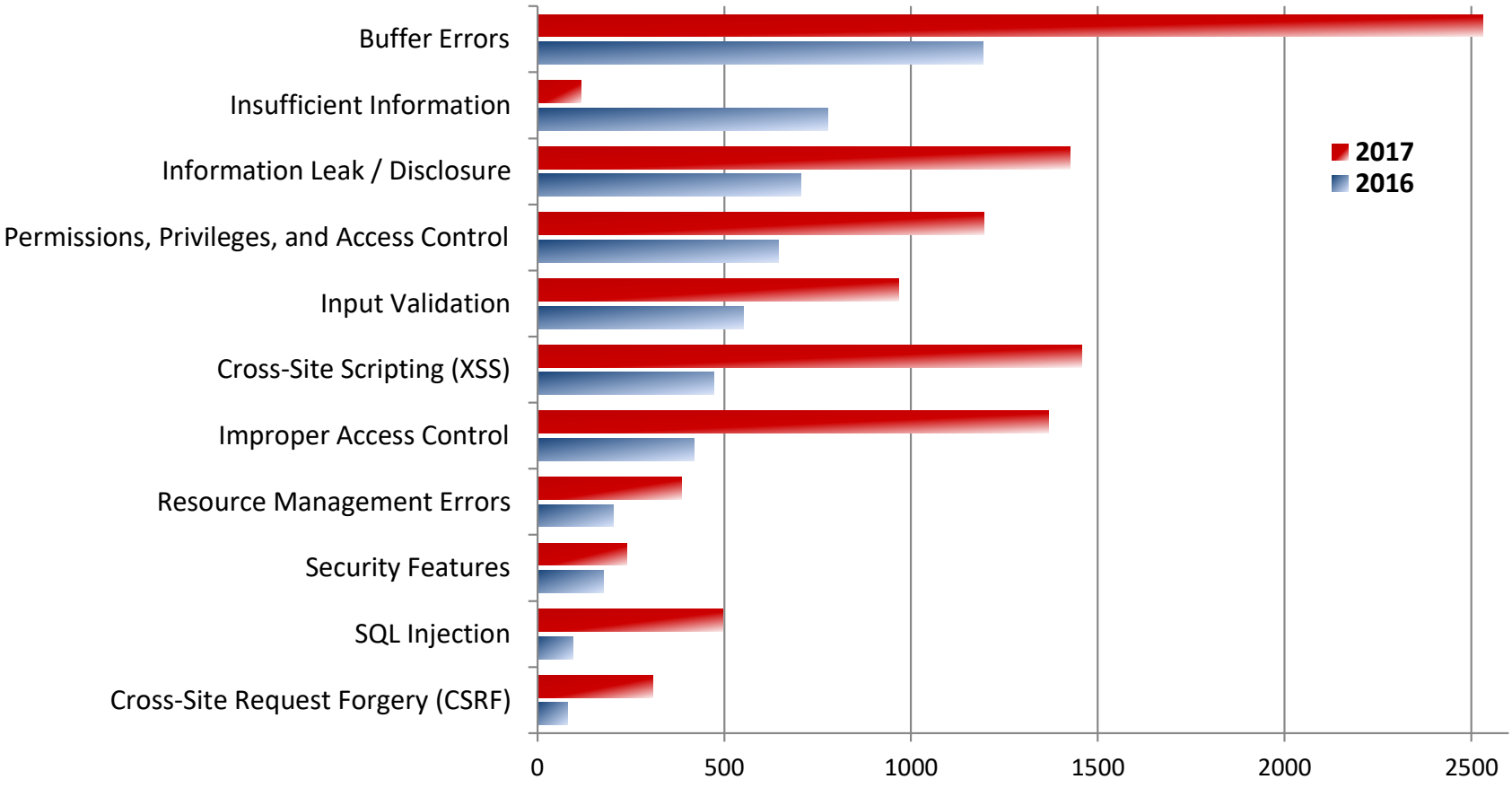
NVD

- operated by NIST
- vulnerabilities (software flaws) published:

year	2013	2014	2015	2016	2017
count	5174	7903	6453	6449	14646

- the rise – “organizational changes and increased vulnerability research”
- includes classification (categories, severity etc.)
- for more detailed analysis, see e.g.
  Skybox Security: Vulnerability and Threat Trends Report 2018 (Analysis of current vulnerabilities, exploits and threats in play)

NVD – selected vulnerabilities published in 2016 and 2017



Examples . . . (1)

Authentication Issues (CVE-2017-13872):
Apple macOS High Sierra before Security Update 2017-001 . . . It allows attackers to obtain administrator access without a password via certain interactions involving entry of the root user name.

Buffer Errors (CVE-2017-11281, CVE-2017-11282):
Adobe Flash Player has an exploitable memory corruption vulnerability in [the text handling function | the MP4 atom parser]. Successful exploitation could lead to arbitrary code execution. This affects 26.0.0.151 and earlier.

Examples . . . (2)

Cryptographic Issues (CVE-2017-[12373, 13099, 13098, 6168, . . .]):
Cisco, Citrix, F5, WolfSSL, Bouncy Castle , Radware, . . . Return Of Bleichenbacher's Oracle Threat (ROBOT)

Input Validation (CVE-2017-5638) . . . *Equifax*:
The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands via a crafted Content-Type, Content-Disposition, or Content-Length HTTP header, as exploited in the wild in March 2017 with a Content-Type header containing a #cmd= string.

Examples . . . (3)

Credentials Management (CVE-2017-3192):
D-Link DIR-130 firmware version 1.23 and DIR-330 firmware version 1.12 do not sufficiently protect administrator credentials. The tools_admin.asp page discloses the administrator password in base64 encoding in the returned web page. A remote attacker with access to this page (potentially through a authentication bypass such as CVE-2017-3191) may obtain administrator credentials for the device.

Improper Access Control (CVE-2017-11779):
The Microsoft Windows Domain Name System (DNS) DNSAPI.dll on Microsoft Windows 8.1, Windows Server 2012 R2, Windows RT 8.1, Windows 10 Gold, 1511, 1607, and 1703, and Windows Server 2016 allows a remote code execution vulnerability when it fails to properly handle DNS responses, aka "Windows DNSAPI Remote Code Execution Vulnerability". (without authentication)

Examples . . . (4)

SQL Injection (CVE-2017-16510):
WordPress before 4.8.3 is affected by an issue where $wpdb->prepare() can create unexpected and unsafe queries leading to potential SQL injection (SQLi) in plugins and themes, as demonstrated by a “double prepare” approach, a different vulnerability than CVE-2017-14723.

Security Features (CVE-2016-0019):
The Remote Desktop Protocol (RDP) service implementation in Microsoft Windows 10 Gold and 1511 allows remote attackers to bypass intended access restrictions and establish sessions for blank-password accounts via a modified RDP client, aka “Windows Remote Desktop Protocol Security Bypass Vulnerability.”

Other classifications of vulnerabilities

- MITRE:
  - Common Vulnerabilities and Exposures (cve.mitre.org)
  - Common Weaknesses Enumeration (cwe.mitre.org)
  - Common Attack Pattern Enumeration and Classification (capec.mitre.org)
- Open Web Application Security Project (OWASP, www.owasp.org)
  - primarily for web applications – vulnerabilities, attacks, risks
  - OWAPSP Top 10 (most critical web application security risks, 2017)
  - Testing Guide (v4, 2014)
- more detailed classifications, description, examples, additional information

Real world – sample issues – January 2018

- Meltdown and Spectre (various variants)
  - patches – performance and stability issues
- Strava aggregated and anonymized heat map – reveals the location of military bases
- Cisco WebVPN – remote attacker can execute arbitrary code and reload the device (CVE-2018-0101)
- Lenovo Fingerprint Manager Pro – hardcoded password, weak encryption algorithm, ...(CVE-2017-3762)
- Firefox 56-58 (CVE-2018-5124) – arbitrary code execution
- Hawaii – false ballistic missile alert

Real world – surveys, analyses, predictions

- EY's Global Information Security Survey 2017-18
- Verizon's Data Breach Investigations Report 2017
- PwC's The Global State of Information Security Survey 2018
- Skybox Security: Vulnerability and Threat Trends Report 2018
- Various Security Predictions for 2018:
  - Symantec, Kaspersky, Forcepoint, FireEye, Trend Micro, McAfee, ...

Some findings from global surveys

- EY's Global Information Security Survey 2016-17
  - approx. 1.200 respondents (CISO, CIO, etc.)
  - 75% very low to moderate maturity of vulnerability identification
  - 35% ad hoc or non-existent data protection policies
  - top two threats: phishing, malware (no change from the previous survey)
- PwC's The Global State of Information Security Survey 2018
  - approx. 9.200 respondents (executives), 122 countries
  - 44% do not have an overall information security strategy
  - 54% do not have an incident response process
  - 48% do not have an employee security awareness training program

Verizon – 2017 Data Breach Investigations Report (1)

- summary of 2016, global coverage
- Incident: A security event that compromises the integrity, confidentiality or availability of an information asset.
- Breach: An incident that results in the confirmed disclosure – not just potential exposure – of data to an unauthorized party
- datasets contributed by various security vendors
- 42.122 security incidents, 1.965 confirmed data breaches

Verizon – 2017 Data Breach Investigations Report (2)

- top 10 vulnerabilities ~ 85% of the successful exploits (2015)
- patterns:

	incidents	breaches
Web App Attacks	15.4%	29.1%
Cyber-espionage	0.8%	14.7%
Privilege Misuse	18.4%	14.1%
Miscellaneous Errors	5.9%	11.3%
POS Intrusions	0.5%	10.5%
Everything Else	2.1%	9.4%
Payment Card Skimmers	0.3%	4.5%
Physical Theft/Loss	13.5%	3.8%
Crimeware	16.4%	2.4%
Denial-of-Service	26.7%	0.3%

What to do – regulatory and compliance frameworks

- NIST SP 800-53 (Rev. 4) Recommended Security Controls for Federal Information Systems and Organizations
- NIST Framework for Improving Critical Infrastructure Cybersecurity (Cybersecurity Framework)
- ISO/IEC 27002:2013 Information technology – Security techniques – Code of practice for information security controls
- Australian Signals Directorate: Strategies to Mitigate Cyber Security Incidents
- Australian Government Information Security Manual – Executive Companion / Principles /Controls
- ISACA: COBIT 5 Framework
- CIS Critical Security Controls (ver. 6.1)
- Payment Card Industry – Data Security Standard version 3.2 (PCI DSS)

CIS Critical Security Controls (1)

https://www.cisecurity.org/controls/

1. Inventory of Authorized and Unauthorized Devices
2. Inventory of Authorized and Unauthorized Software
3. Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers
4. Continuous Vulnerability Assessment and Remediation
5. Controlled Use of Administrative Privileges
6. Maintenance, Monitoring, and Analysis of Audit Logs
7. Email and Web Browser Protections
8. Malware Defenses
9. Limitation and Control of Network Ports, Protocols, and Services
10. Data Recovery Capability

CIS Critical Security Controls (2)

https://www.cisecurity.org/controls

11. Secure Configurations for Network Devices such as Firewalls, Routers, and Switches
12. Boundary Defense
13. Data Protection
14. Controlled Access Based on the Need to Know
15. Wireless Access Control
16. Account Monitoring and Control
17. Security Skills Assessment and Appropriate Training to Fill Gaps
18. Application Software Security
19. Incident Response and Management
20. Penetration Tests and Red Team Exercises

UK: Cyber Essentials Scheme

https://www.cyberaware.gov.uk/cyberessentials/

Requirements for basic technical protection from cyber attacks

1. Boundary firewalls and internet gateways
2. Secure configuration
3. User access control
4. Malware protection
5. Patch management

Data breaches – examples

Data breaches – examples (1)

1. Equifax
   - detected: July 2017, started: May 2017
   - 143 million people affected
   - attackers used unpatched Apache Struts vulnerability (CVE-2017-5638)
   - names, SSNs, birth dates, addresses
   - in some instances, driver's license numbers, credit card numbers
2. Uber
   - October 2016, revealed: November 2017
   - leaked personal data of 50 million customers and 7 million drivers
   - names, email addresses, phone numbers
   - attack: AWS (Amazon Web Services) logon credentials accessible on GitHub
   - Uber paid the attackers $100.000 to delete data and keep quiet

Data breaches – examples (2)

3. Office of Personnel Management
   - detected: April 2015, started: March 2014
   - 21.5 million records
   - attackers with valid user credentials / contractors
   - names, SSNs, dates and places of birth, addresses, security-clearance information
   - 5.6 million sets of fingerprints
4. Anthem (managed health care company)
   - December 2014 – January 2015
   - leaked personal data of 80 million customers
   - names, dates of birth, SSN, health care ID numbers, home addresses, email addresses, employment information, income data
   - attack: some tech employees had their credentials compromised
   - detection: noticing suspicious queries

   Similar breach: Premera (11 million people)

Data breaches – examples (3)

5. Ashley-Madison
   - data breach announced in July 2015 (“Impact Team”)
   - 10GB + 19GB compressed data
   - ∼ 37 million records (customers)
   - e-mail addresses, names, credit card transactions, …
   - source code, e-mails
   - suicides, blackmailing, bcrypt + MD5
6. Friend Finder Network
   - October 2016
   - 412 million accounts (Adult Friend Finder, Cams.com, Penthouse.com, Stripshow.com …)
   - addresses, passwords, dates of last visits, browser information, IP addresses and site membership status
   - not the first time (May 2015, 4 million users)
   - plaintext and SHA-1 password (lowercase)
   - over 99% paswords cracked

Data breaches – examples (4)

7. Hacking Team
   - selling offensive intrusion and surveillance capabilities to governments, law enforcement agencies and corporations
   - data breach announced: July 2015
   - 400GB (customers, e-mails, 0-day exploits, source code, ...)
   - weak passwords, e.g. “P4ssword”, “HTPassw0rd”, “wolverine”
8. IRS (Internal Revenue Service)
   - detected: too many old tax returns (May 2015)
   - stolen credentials (probably from other data breaches, e.g. Anthem)
   - 334 thousand people
   - 15,000 falsified documents processed ...50 million USD in refunds

Data breaches – examples (5)

9. Target (USA, retail)
   - December 2013
   - 40 million credit and debit cards information
     + additional 70 million personal information
   - card information
     + names, mailing addresses, phone numbers, email addresses
   - malware installed on POS devices
   - entry using authentication credentials stolen from a heating, ventilation, and air-conditioning subcontractor
10. JPMorgan Chase (USA, banking)
    - discovered in July 2014
    - names, addresses, phone numbers and e-mail addresses of 83 million account holders
    - initial assumption: 0-day web server exploit (?)
    - reality: stolen credentials (password), 2nd factor not enabled on one server
    - 90 servers compromised when detected

Data breaches – examples (6)

9. Home Depot (USA, retail)
   - breach started in April 2014, undetected for 5 months
   - 56 million customer credit and debit card accounts
     + 53 million customer email addresses
   - malware on self-checkout registers
   - initial step: credentials stolen from a third-party vendor
10. Sony Pictures (USA, entertainment)
   - 100TB of data (?)
   - discovered in November 2014
   - personal information about employees, e-mails, salaries, copies of unreleased Sony films
   - North Korea (?)
   - the White House reacts

Other security incidents

Other security incidents (1)

- stealing money
  - Bangladesh Bank (March 2016)
  - operator's SWIFT credentials, malware
  - bank transfers from Bangladesh Bank's account in Federal Reserve Bank of New York to Philippines and Sri Lanka
  - 81 million USD (only a typo prevented 1 billion USD transfer)
  - recent example: Russian Central Bank (6 million USD, 2017)
- Ukrainian Power Grid
  - December 2015
  - BlackEnergy trojan
  - black-out (for few hours): 103 cities complete 184 cities partial
  - blocked call centers

Other security incidents (2)

- NSA
  - 2013; approx. 1.7 million files
  - Snowden (contractor)
  - gradual publication of documents and files, global surveillance programs
    - tools and methods, e.g. see Tailored Access Operations (TAO) catalog
    - identities of cooperating companies and governments
    - identities of ISPs and platforms that NSA has penetrated or attempted to penetrate
    - foreign officials and systems that NSA has targeted
- Associated Press
  - April 2013
  - AP Twitter account hacked:
    Breaking: Two Explosions in the White House and Barack Obama is Injured.
  - 136 billion USD from the S&P's 500 Index in two minutes

Other security incidents (3)

- Network Time Protocol – DoS attacks
  - NTP amplification attack (amplification factor 19)
  - single 234-byte request . . . 10 packets response (total 4 460 bytes).
  - MONLIST command (IP addresses of the last 600 machines interacting with an NTP server)
  - February 2014 . . . reported DDoS attack with 400 Gbps traffic
- DynDNS and Mirai
  - October 2016
  - DDoS attack ~ 1.2Tbps
  - primary source of the attack: Mirai botnet
  - Mirai: IoT devices – routers, DVRS and CCTV cameras (> 60 common default usernames and passwords)
  - September 2016 (KrebsOnSecurity, 620Gbps)

Other security incidents (4)

- Crypto-ransomware
  - 2017 (Kaspersky):
    - 38 new families (62 in 2016)
    - 96 thousand modifications (54 thousand in 2016)
  - top 3 families: WannaCry, Locky, Cerber
  - victims – anybody
  - Ransomware-as-a-Service
  - diversifying targets: KeRanger (OS X, 2016), Linux.Encoder.1 (Linux, 2015), MongoDB databases (2017)
  - May 2017 WannaCry
    - 4 days, 200.000 computers, 150 countries
    - EternalBlue exploit (developed by NSA, leaked by Shadow Brokers in April 2017)
  - October 2017 Bad Rabbit
    - fake Adobe Flash update
    - EternalRomance exploit (developed by NSA, leaked by Shadow Brokers)

Security failures/vulnerabilities ...

examples

The most frequent passwords

source: Splashdata, based on leaked passwords (2017 and comparison with 2016)

1.	123456	
2.	password	
3.	12345678	(+ 1)
4.	qwerty	(+ 2)
5.	12345	(− 2)
6.	123456789	(new)
7.	letmein	(new)
8.	1234567	
9.	football	(− 4)
10.	iloveyou	(new)
11.	admin	(+ 4)
12.	welcome	
13.	monkey	(new)
14.	login	(− 3)
15.	abc123	(− 2)
16.	starwars	(new)
17.	123123	(new)
18.	dragon	(+ 1)
19.	passw0rd	(− 1)
20.	master	(+ 1)
21.	hello	(new)
22.	freedom	(new)
23.	whatever	(new)
24.	qazwsx	(new)
25.	trusno1	(new)

How to verify the certificates for TLS

- 76 iOS applications from App Store vulnerable to MITM attacks (January 2017)
- not a new issue:
  - CVE-2016-6231: Kaspersky Safe Browser iOS before 1.7.0 does not verify X.509 certificates from SSL servers, which allows man-in-the-middle attackers to obtain sensitive information via a crafted certificate.
  - CVE-2016-3664: Trend Micro Mobile Security for iOS before 3.2.1188 does not verify the X.509 certificate of the mobile application login server, which allows man-in-the-middle attackers to spoof this server and obtain sensitive information via a crafted certificate.
  - many others ...
- $\sim$ 1.400 Android applications (2014):

  The ...application for Android does not verify X.509 certificates from SSL servers, which allows man-in-the-middle attackers to spoof servers and obtain sensitive information via a crafted certificate.

Randomness of cryptographic keys

- 2008 – Debian
- modification of openssl source code
  - the use of uninitialized memory
  - broken initialization of pseudorandom generator . . . initialized by PID only
  - at most 98301 unique initialization values overall (depending on particular platform)
- impact:
  - predictable keys for SSH, OpenVPN, DNSSEC, X.509 certificates, session keys in SSL/TLS, . . .
  - using library just for a single DSA signing . . . compromised private key
  - similar problem with randomness in Sony Playstation 3 (ECDSA signatures, 2010)

Later . . . in openssl 1.0 source code

```
/* DO NOT REMOVE THE FOLLOWING CALL TO MD_Update()! */
MD_Update(&m,buf,j);
/* We know that line may cause programs such as
   purify and valgrind to complain about use of
   uninitialized data.  The problem is not, it's
   with the caller.  Removing that line will make
   sure you get really bad randomness and thereby
   other problems such as very insecure keys. */
```

- Correct and secure implementation of cryptography is not easy
  - 10 vulnerabilities in openssl (NVD, published in 2010–2014) with severity High, not including the Heartbleed bug (with severity Medium)

Heartbleed

- probably the most important vulnerability in 2014
- problem in implementation of heartbeat extension (RFC6520) in OpenSSL
- the attacker can read the memory of the server

  request (as intended): send me this 6 byte payload “abcdef”
  response: “abcdef”

  request (attack): send me this 20003 byte payload “abc”
  response: “abc” + 20000 bytes of server’s memory

Timing attacks on comparisons (Google, Sun, …)

- 2009 – Keyczar (Google), Java (Sun), …
- common scenario: server compares received HMAC with calculated one
- attacker's goal: to get correct HMAC for his own message ("authentic")
- What is wrong with this code (Python)?

```
return self.Sign(msg) == sig_bytes
```

What is wrong with this code (Java)?

```
public static boolean
isEqual(byte digesta[], byte digestb[]) {
  if (digesta.length != digestb.length)
    return false;

  for (int i = 0; i < digesta.length; i++) {
    if (digesta[i] != digestb[i])
      return false;
  }
  return true;
}
```

HMAC reconstruction

How long does it take for server to answer/react to incorrect HMAC

- if the first 0, 1, 8 or 15 bytes are correct?
- HMAC reconstruction based on time-variance of responses
- 4th byte:

```
71 A0 89 00 00 . . . 00
71 A0 89 01 00 . . . 00
         . . .
71 A0 89 4A 00 . . . 00      longer time to process?
         . . .
71 A0 89 FF 00 . . . 00
```

- usually multiple measures required for a single value (noise)
- statistical evaluation of measurements

Constant-time comparison (Java)

```
public static boolean
isEqual(byte[] digesta, byte[] digestb) {
  if (digesta.length != digestb.length)
    return false;

  int result = 0;
  for (int i = 0; i < digesta.length; i++) {
    result |= digesta[i] ^ digestb[i];
  }
  return result == 0;
}
```

Adobe password encryption

- 2013, Adobe
- data breach, 38 million *active* users account information exposed
- 150 million user accounts overall
- passwords are encrypted (the key was not leaked)
  …using 3DES (block cipher with 8 B block) in ECB mode
- result:
  - equal password substring [1-8], [9-16] easily identifiable
  - guess using password hits (part of account information), e.g.
    “numbers 123456”, “c’est 123456”
    “1*6”, “sixones”
    “q w e r t y”, “6 long qwert”

WPS (WiFi Protected Setup)

- 2011
- goal: easy (and secure) method to add a device to network
- implementation:
  - 8 digit PIN code authentication (printed on a sticker)
  - theoretically 10^8 possibilities
  - practically: response to incorrect PIN leaks an information whether the first half of the PIN is wrong
    last digit is a checksum
  - $10^4 + 10^3$ possibilities
- WPS can't be turned off in some WiFi routers

Encrypted USB drives

- 2010; Kingstone, SanDisk, Verbatim
- FIPS 140-2 Level 2 certification; AES-256 encryption
- reality:
  - encryption key does not depend on user's password
  - USB key unlocks if some expected string (fixed, password- and device-independent) is received

Hash tables collisions

- 2011; Oracle, Microsoft, PHP, Apache Tomcat, ...
- analogous problem found originally in 2003; Perl, Squid
- hash table – data structure for storing (key/data) pairs
  - average complexity $O(n)$ for inserting/deleting/finding n elements
  - worst case complexity $O(n^2)$ for n elements (when keys collide)
- problem: colliding keys can be generated easily
- parameters of HTTP POST requests are parsed into hash table automatically
- DoS attack on web server:
  ~70-100kbits/s $\Rightarrow$ one i7 core busy (2011, PHP)

Hashing for hash tables

- Java 6 (java.lang.String, method public int hashCode())
  - 32-bit arithmetic (int), s_i denotes an i-th character of an $(s_{1,\ldots,n})$:

$$\sum_{i=1}^{n} 31^{n-i} \cdot s_i$$

- PHP 5 (algorithm DJBX33A, 32-bit arithmetic), s_0 is constant 5381

$$\sum_{i=0}^{n} 33^{n-i} \cdot s_i$$

- ASP.NET (algorithm DJBX33X), s_0 is constant 5381

$$\bigoplus_{i=0}^{n} 33^{n-i} \cdot s_i$$

- easy to find large multicollisions

Solutions

- limit the size of POST requests, limit CPU for single request, etc.
- better hash function
  - for example randomized hashing – the function dependent on randomly chosen parameter (when process starts)

Apple “goto” fail (2014)

```
SSLVerifySignedServerKeyExchange(...)
{
  OSStatus  err;
  ...
  if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
  if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
  if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
    goto fail;
    ...
    err = sslRawVerify(...)

  fail:
    ...
    return err;
}
```