Internet security experiences

1985-2000 and beyond

Karst Koymans

Informatics Institute
University of Amsterdam

(version 16.1, 2016/09/05 09:26:35)

Friday, September 9, 2016
1 Context and background

2 General principles

3 Some real life examples

4 Principles

5 Insanity... 

6 The SNE era

7 Conclusions
Outline

1. Context and background
2. General principles
3. Some real life examples
4. Principles
5. Insanity...
6. The SNE era
7. Conclusions
Origins

- A personal view on security
- Originally presented at
  - SAFE-NL
  - June 14, 2002
- But much of it still applies
Contents

- Some stories...
- Some thoughts...
- Some ideas...
- Some warnings...
- ...out of my personal experience
Outline

1. Context and background
2. General principles
3. Some real life examples
4. Principles
5. Insanity...
6. The SNE era
7. Conclusions
Security is more than keeping (cr|h)ackers out

- Malicious (internal) actions
- Unintentional errors
- Pure stupidity
- Nuisances
  - SPAM, UCE
- ... and much more
Security is strongly related to

- Structure
- Privacy
- Identity
- Robustness
- Information
- Trust
- Usability
- Anonymity
- Laziness
- Safety
**Important frameworks**

- **AAA**
  - Who? (Authentication, Identification)
  - What? (Authorization)
  - When? (Auditing, Accounting)

- **PKI**
  - Public Key Infrastructure
  - Encryption and privacy
  - Holy grail, difficult to realise
Outline

1. Context and background
2. General principles
3. Some real life examples
4. Principles
5. Insanity...
6. The SNE era
7. Conclusions
Some real life examples

Early days example (1985)

- Netbooting on a class B broadcast network
- Client machine named “pluto” asks for bootparameters
- Talking to server machine named “plato”
- Answer came from “outer space” without sensible content
Users of all times (1985-today)

- Passwords should satisfy
  - Is at least six characters long
  - Contains non-alphanumeric character(s)
  - Is not simple to guess

- Choice made by user
  - “John” (in fact it was “Joop”)
Conclusions about users

- An easy, but probably wrong, conclusion
  - Users are stupid

- A probably better conclusion
  - Users have other priorities
Admins of all times (1988-today)

- nVIR: early Macintosh virus
- Admin comes to check for viruses...
- Admin collects viruses for a hobby...

- Before visit... virus-free
- After visit... chaos

Source: http://xkcd.com/694/
Some real life examples

Xkcd illustration...

I've got a bunch of virtual windows machines networked together, hooked up to an incoming pipe from the net. They execute email attachments, share files, and have no security patches. Between them they have practically every virus.

There are mail trojans, Warhol worms, and all sorts of exotic polymorphs. A monitoring system adds and wipes machines at random. The display shows the viruses as they move through the network, growing and struggling.

You know, normal people just have aquariums.

Good morning, Blaster. Are you and W32.Welchia getting along?

Who's a good virus? You are! Yes, you are!

Source: http://xkcd.com/350/
Conclusions about admins

- An easy, but probably wrong, conclusion
  - Admins are stupid

- A probably better conclusion
  - Admins also make mistakes
Physical security (1992)

- Separate servers from clients
- Thieves can be very brutal
- The case of the PC user…
  - …behind a Sun workstation
Outline

1. Context and background
2. General principles
3. Some real life examples
4. Principles
5. Insanity...
6. The SNE era
7. Conclusions
Definition (Postel’s Law or Robustness Principle)

Be liberal in what you accept, and conservative in what you send.

- The exact wording is from RFC 1122 (October 1989)
- It is already mentioned in other words in IEN\textsuperscript{1} 111 (August 1979)
- Can you see the problems with this principle?

\textsuperscript{1}Internet Experiment Note
Correctness principle

Definition (Correctness principle or Strictness principle)
Be strict in what you accept, and strict in what you send.
The problem with software

- Software is made by trial and error
- C supports buffer overflows
- Viruses, Worms, Trojan Horses
- Community reactions
  - CERT/CC advisories (1988)
  - BugTraq (1993)
Outline

1. Context and background
2. General principles
3. Some real life examples
4. Principles
5. Insanity...
6. The SNE era
7. Conclusions
CERT/CC insanity (1)

- CA-1988-01
  - ftput Vulnerability
    - also about sendmail and the Morris worm
    - and about passwordless alternative root accounts (uid == 0)
    - and also about bad password choices
  - ...(alarming but “innocent”)

- CA-1995-01
  - IP spoofing Attacks$^2$ and Hijacked Terminal Connections

---

$^2$BCP 38 is dated May 2000
CERT/CC insanity (2)

- CA-1995-04
  - NCSA HTTP Daemon for UNIX Vulnerability
    - Buffer overflow

- CA-1995-18
  - Widespread Attacks on Internet sites
    - NFS, NIS, RPC, Trojans, IP spoofing, ...
CERT/CC insanity (3)

- CA-1996-07
  - Weaknesses in Java Bytecode Verifier
- CA-1996-11
  - Interpreters in CGI bin Directories
- CA-1996-26
  - Denial-of-Service Attack via ping (of death)
    - Oversized ICMP echo request packet
    - No length check before reassembly of fragmented packets
CERT/CC insanity (4)

- CA-1997-08
  - Vulnerabilities in INND
    - Incomplete user input checking
- CA-1997-09
  - Vulnerabilities in IMAP and POP
    - Buffer overflow
- CA-1997-20
  - Javascript Vulnerability
    - Observing the URLs of visited documents
    - Observing data filled into HTML forms (including passwords)
    - Observing the values of cookies
CERT/CC insanity (5)

- CA-1997-28
  - IP Denial-of-Service Attacks
    - Teardrop (overlapping IP fragments)
    - Land (spoofed source == destination)

- CA-1998-01
  - Smurf IP Denial-of-Service Attacks
    - Using spoofed ICMP echo requests

- CA-1998-08
  - Buffer overflows in some POP servers
CERT/CC insanity (6)

- CA-etc-etc
  - Buffer overflows, Format string vulnerabilities
  - Trojans, Misconfigurations, ... 

- I just gave up...
A partial solution

- Minimalisation of access
  - Start with the empty set of services
  - Only add the services you really need
  - No blacklists, only whitelists

- Protect your core
  - Main servers
  - Network equipment
But the world keeps spinning...

- CA-1999-02
  - Trojan Horses
- CA-1999-04
  - Melissa Macro Virus
- CA-1999-07
  - IIS Buffer Overflow
- CA-2000-04
  - Love Letter Worm
- CA-2002-16
  - Multiple Vulnerabilities in Yahoo! Messenger
- CA-. . . - . . .
Outline

1. Context and background
2. General principles
3. Some real life examples
4. Principles
5. Insanity...
6. The SNE era
7. Conclusions
The SNE era — an arbitrary example from 2003

- CA-2003-26
  - Multiple Vulnerabilities in SSL/TLS Implementations
    - OpenSSL ASN.1 parser insecure memory deallocation
    - OpenSSL contains integer overflow handling ASN.1 tags
    - OpenSSL accepts unsolicited client certificate messages
The SNE era — an arbitrary example from 2004

- CERT advisories become part of Technical Cyber Security Alerts
  - https://www.us-cert.gov/ncas/alerts/

- Technical Cyber Security Alert TA04-293A
  - Microsoft Internet Explorer contains a buffer overflow in CSS parsing
  - Microsoft Internet Explorer Install Engine contains a buffer overflow vulnerability
The SNE era — an arbitrary example from 2005

- Technical Cyber Security Alert TA05-292A
  - Oracle Products Contain Multiple Vulnerabilities
    - Various Oracle products and components are affected by multiple vulnerabilities
    - The impacts of these vulnerabilities include unauthenticated, remote code execution, information disclosure, and denial of service
The SNE era — an arbitrary example from 2006

- Technical Cyber Security Alert TA06-256A
  - Apple QuickTime Vulnerabilities
    - Apple QuickTime movie buffer overflow vulnerability
    - Apple QuickTime fails to properly handle FLC movies
    - Apple QuickTime Player H.264 Codec contains an integer overflow
The SNE era — an arbitrary example from 2007

- Technical Cyber Security Alert TA07-355A
  - Adobe Updates for Multiple Vulnerabilities
    - Adobe Flash Player asfunction protocol may enable cross-site scripting
    - Adobe Flash Player may load arbitrary, malformed cross-domain policy files
    - Flash authoring tools create Flash files that contain cross-site scripting vulnerabilities
The SNE era — an arbitrary example from 2008

- Technical Cyber Security Alert TA08-190B
  - Multiple DNS implementations vulnerable to cache poisoning
    - Insufficient transaction ID space
    - Multiple outstanding requests
    - Fixed source port for generating queries
  
- Also known as the (Dan) Kaminsky attack
The SNE era — an arbitrary example from 2009

- Technical Cyber Security Alert TA09-088A
  - Conficker Worm Targets Microsoft Windows Systems
    - Widespread infection of the Conficker/Downadup worm
    - A remote, unauthenticated attacker could execute arbitrary code on a vulnerable system.
The SNE era — an arbitrary example from 2010

- Technical Cyber Security Alert TA10-348A
  - Microsoft Updates for Multiple Vulnerabilities
  - There are multiple vulnerabilities in
    - Microsoft Windows
    - Internet Explorer
    - Office
    - Sharepoint
    - Exchange
The SNE era — an arbitrary example from 2011

- Technical Cyber Security Alert TA11-200A
  - Security Recommendations to Prevent Cyber Intrusions
    - Almost infinite enumeration of how to eliminate bad habits
The SNE era — an arbitrary example from 2012

- Technical Cyber Security Alert TA12-024A
  - “Anonymous” DDoS Activity
    - Low Orbit Ion Cannon (LOIC) DoS-attack
  - Activism
The SNE era — an arbitrary example from 2013

- Technical Cyber Security Alert TA13-088A
  - DNS Amplification Attacks
    - Open Recursive Nameserver problem
The SNE era — an arbitrary example from 2014

- Technical Cyber Security Alert TA14-098A
  - OpenSSL 'Heartbleed' vulnerability
    - Bounds/input checking problem: private memory leakage
    - On servers, but also on clients!
The SNE era — an arbitrary example from 2015

- Technical Cyber Security Alert TA15-120A
  - Securing End-to-End Communications
    - TLS/SSL issues
    - POODLE attack
    - Also applicable to RC4 attack, FREAK, Logjam, ...
The SNE era — an arbitrary example from 2016

- Ransomware and Recent Variants TA16-091A
  - Destructive
  - Found in healthcare systems and hospitals
  - Locky
    - Spreads through spam and Office documents or attachments
  - Samas
    - Spreads through vulnerable web servers
Outline

1. Context and background
2. General principles
3. Some real life examples
4. Principles
5. Insanity...
6. The SNE era
7. Conclusions
Conclusions

Some misconceptions

- Open source is bad for security
  - No!...
  - ...proprietary software creates much bigger problems
- Security through obscurity is bad
  - Yes, but not always...
  - ...“parameter obscurity” can be good
- Performance is important
  - Only hardly ever true...
  - ...structure, modularisation and correctness proofs are much more important
Some advice

- Avoid complicated, monolithic SW
  - sendmail → postfix

- Avoid legacy
  - Start over now and then: ruu.nl → uu.nl
  - It is really time for a clean slate approach? It is!

- Centralise at the right level
  - But make sure that the central resources are at least as good and knowledgeable as decentralised ones
A new era?

- Improvements?
  - IPsec, DNSSEC
  - SSL, SSH
  - VPN
  - TTP/CA

- But also
  - NSA, Snowden
  - GCHQ
  - ???
Fighting legacy example

- IPv6
  - No addressing problems
    - But some routing challenges
  - End to end computing
    - No NATs
  - Autoconfiguration
    - Plug and play (+/-)
  - Integrated IPsec
    - Security from the start
But what happens?

- Cisco introduces IPv6 in its routers without initial IPsec support...

- Why?
  - Because there is no user demand for it...
  - ...SIGH!
Fighting legacy

- Our biggest problem
- No easy solutions
  - Not in everybody’s interest
  - Needs revolution, not evolution
  - Scientific, non-commercial effort
- Real clean slate
  - Build new system in parallel
  - Incompatible on purpose
    - without planned transition mechanisms