Information and disinformation

Where to find them
and how to deal with them

Outline

Why do vulnerabilities happen?

Sources of information

CSIRT tasks

Why do vulnerabilities happen?

Laws of Nature

Customer demands

Vendor pressures

Vulnerabilities: Law of Nature

Computer networks are complex systems

- They will contain errors and inconsistencies
- Some of these will have security implications
- An expert coder creates 1 bug per 1000 lines

  Solaris 7.0 - 12 million lines

  Windows 2000 - 40 million lines
- Figures from Wietse Venema, June 1999

Vulnerabilities: Customer Demand

Buyers demand computers which are easy to use

- Who asks for security as the first priority?

Vendors default to everything on

- In case it's needed one day
- Sun tried the opposite: many "non-working" returns

Users may turn on things they need

- They will never turn off things they don't

All those bugs are exposed to the (hostile) public

Vulnerabilities: Vendor pressures

Commercial pressure to ship code

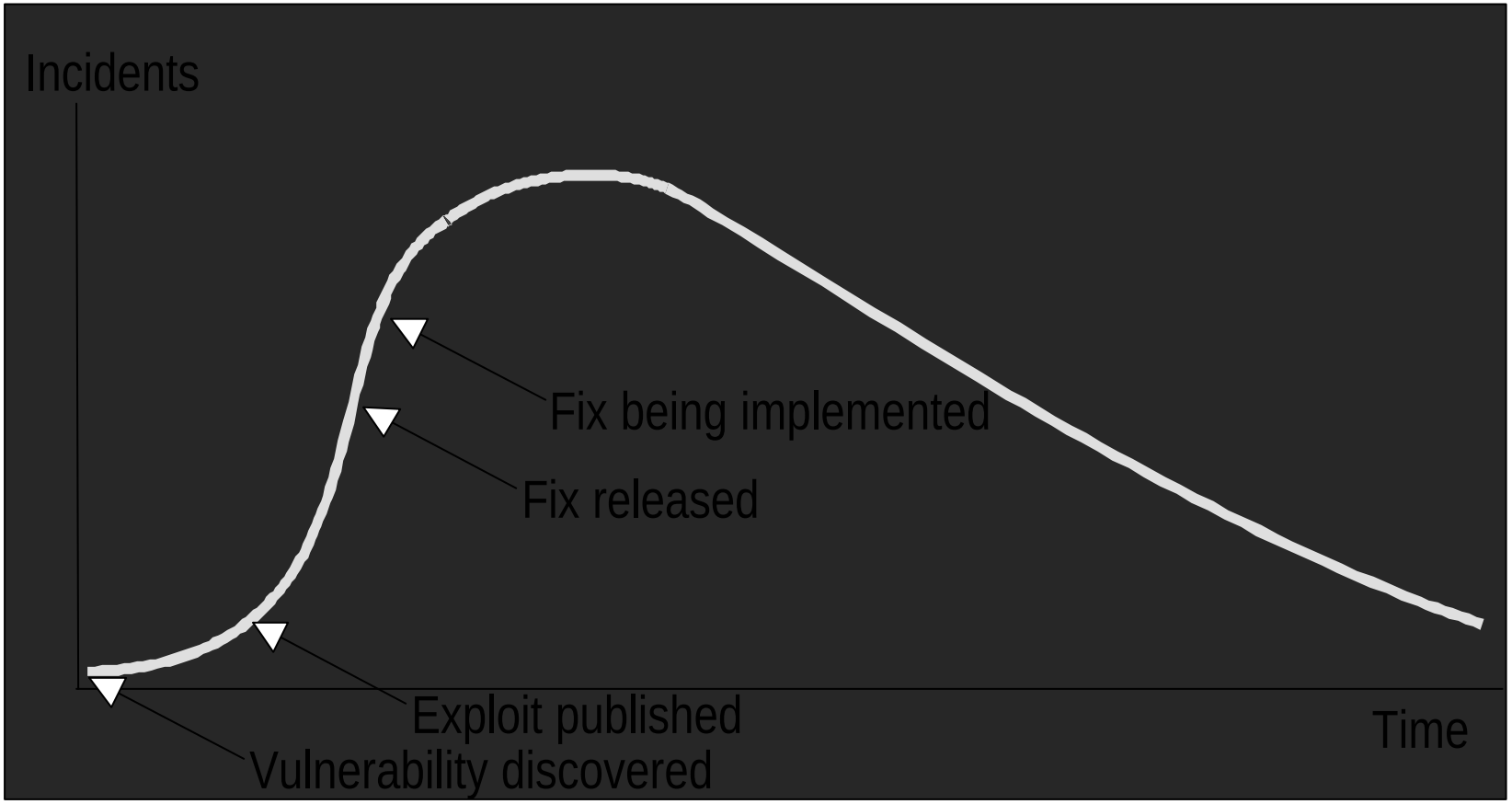
- Functional testing is often skimped
- Security testing is even harder

Reluctance to learn from others' mistakes

- Why are we still seeing buffer overflows?
- Why are web servers still run with test scripts?

Defensive coding/design is very rare

Vulnerability Curve



Sources of information

Incident reports

Full disclosure community

Hackers

Vendors

Commercial services

Other CSIRTs

The information ideal

Reliable

Timely

Complete

Suitable for our constituency

Doesn’t exist!

Information: Incidents

Advantages

- No question that there is a problem!

Disadvantages

- Unlikely to give all the necessary information
  - Intruders do their best to obscure facts!
- Often hard to interpret

Example

- Core file from snmpXdmid confirmed vulnerability
- Attrition.org

Information: Full Disclosure

Advantages

- Up to date information

Disadvantages

- Quality is very variable!
- Information seldom complete
- More problems than solutions

Examples

- Securityfocus.com, slashdot.org, …

Information: Hackers

Advantages

- Current: this is what is being used now

Disadvantages

- Need to reverse engineer tools to find problem
- Need to be very careful in handling material
- Yields incomplete information at best
- Many real experts don’t publish

Examples

- Packetstorm.securify.net, …

Information: Vendors

Advantages

- Best possible information
- Some are very comprehensive

Disadvantages

- Some are very slow
- May be competing motives

Examples

- Cisco, Sun, Microsoft, linux distributions, …

Information: Commercial services

Advantages

- High quality information
- Aim to be faster than vendors

Disadvantages

- Commercial motives
- May be restrictions on distribution

Examples

- Anti-virus vendors, ISS, …

Information: Other CSIRTs

Advantages

- Same motivation as ourselves
- Trustworthy

Disadvantages

- May be slow depending on policy and resource
- May be restrictions on distribution

Examples

- CERT-CC, AusCERT, FIRST, ...

Working with information

Not all information is created equal

Use multiple sources

- For speed, reliability and completeness

Need to verify information

- Trusted source
- Consistent with independent others
- Verify against own tests

Using information

Plan in advance

- How to use it
- Maximise benefit to constituency
- Minimse impact on others

Be a force for good, not bad

CSIRT tasks

Distribution

Interpretation

Investigation

Coordination

Tasks: Distribution

Pass information from others to own constituency

Some teams also translate into local language

CSIRT maintains mailing list/web site

Need high quality information

- E.g. from vendors or other teams

Can be long delay

- Try to publish before widespread attacks

Tasks: Interpretation

Interpret information for local constituency

- Suited to skill level, common platforms, etc.

CSIRT writes own reports or introductions

Can use multiple sources of information

- E.g. black hat, observed activity

Interpretation takes time

- Getting people to act takes even longer!

Advisory notices: content

Help readers, don't just frighten them!

Give useful information early:

- Who is vulnerable (platform, software, service)
- What is the damage (compromise, DoS, etc.)
- Assessment of threat (theoretical, …, present)
- How to fix the problem (workarounds and patches)
- Any other impact of these fixes

Advisory notices: practice

Advisories should be PGP signed if possible

- Worrying recent trend – sites rejecting signed e-mail!
- Signing may be tricky, e.g. with web pages

Advisories should have reference numbers

- Helps readers and other teams

Decide which advisories are archived, and how

Tasks: Investigation (1)

Investigate reported vulnerabilities

- Better understand the problem
- Check patches/workarounds
- Provide patches/workarounds

May be based on

- Incident artefacts
- Source code, if available
- Test systems (not on a public network!)
- Manage and document these

Tasks: Investigation (2)

Know the intended outcome

- Better advice, notification to vendor, etc.

Plan how to achieve that outcome

Be careful about release of information

- You may help the bad guys more than the good

Tasks: Co-ordination

Working with vendors to solve a problem

Requires mutual trust

- Hard to build, easy to lose

Competing demands from those involved

- Vendor – bad publicity
- Sites – need patch to prevent incidents
- Other sites – won't patch: will publicity increase risk?

Summary

Vulnerabilities are inevitable

Information sources exist

- Not always straightforward to use
- Different motivations can cause problems

Dealing with them is hard

- Technically and politically

What does your constituency need most?