

Security on Internet Time

1

The Problem

- Security is **very very** hard to accomplish ...
- But everything is being increasingly computerized (and, more importantly, networked!)
- *Roll-over-play-dead* is not an option
...We have to keep trying because the alternative is worse

2

The Environment: 1

- Hundreds of millions of dollars injected into Internet market start a firestorm
 - Firestorm further fed by wave of IPOs in 1995-1996
 - IPO model/public companies under quarterly inspection: **must ship product**
- So much capital in silicon valley **has** to fundamentally change the 'net

3

The Environment: 2

- Product lifecycles have been shortened to ~3 months (quarterly)
 - Compression of releases totally de-emphasizes the notion of “patch”
 - Run the latest and greatest and hope the bugs are fixed
 - Run the latest and greatest and get the newest bugs

4

The Environment: 3

In: Shovelware

Out: Testing

In: Features

Out: Design

In: Cross-licensing

Out: Standards

In: Running the beta

Out: Code that works

Total:

In: Talkin'bout security

Out: Security

5

Sources of Problems:

- Non-technical
 - Market forces
 - Regulation
- Technical
 - People bandwidth
 - Layering of Mistakes
 - Mistakes

6

**Non-technical problems are more
deadly than technical ones...**

7

Market Forces: Customers

- Secure **BlahBlahBlah** makes people comfortable
 - Just add cryptography and “**Thing**” becomes “**Secure Thing**”
 - Ignore the details of what’s going on at the edges of the transaction
 - Ignore the question of whether the data is **valid**
 - Trade press aids and abets this attitude

8

Market Forces: Customers

- Case study: SSL / S-HTTP
- Add crypto to the Web
 - Never mind frequent huge CGI holes
 - Never mind frequent huge host security holes on web servers
- Web server software available at CompUSA: “Secure Web Server!” (supports SSL)

9

Market Forces: Time-to-Market

- The software industry is largely driven by market share
 - Market share and mind share are driven by who gets out there first
 - Whatever gets out there first is not likely to be **good** - just **first**
 - More to the point it is almost certainly going to have security left out
 - But if it sells, who cares?

10

Market Forces: Time-to-Market

- Case study: Netscape
 - Browser has had a large number of security flaws
 - Still very popular
 - If Netscape had waited to ship their browser until it had fewer bugs would they be Netscape today?
 - More simply: Do you ship buggy code and drive a Ferrari or take the time to get it right?

11

Market Forces: Standards

- The key to security is leverage provided by robust implementations we can trust
 - This entails ***standardization***
- Current market pressure is away from standards in favor of market share and mind share
 - IETF has no clout anymore
 - Standards now set by trade rags & Wall St.

12

Market Forces: Standards

- Case Study: IPSEC key exchange
 - First there was Photuris (which worked fine)
 - Then Sun tried to ram through SKIP (which worked fine but was Sun's idea)
 - Then ISAKMP comes along (which is kind of a mix of both)
 - Upshot: It's been about 4 years and still no viable standard has emerged

13

Market Forces: Standards

- What's going on?
 - Standards bodies are representational
 - To join, you need to be:
 - 1) Breathing (or at least warm)
 - 2) Able to pay dues/airfare to get there
 - Note that technical knowledge not needed
 - 1990: Vendors first start packing standards bodies with lobbyists (Sun tries to get IEEE to brand SPARC a standard)

14

Market Forces: Compatibility

- Vendor-sponsored incompatibility is the latest trend
 - Enforce your market lock by advancing a competing non-interoperable incompatible standard
 - Vendors bolster positions and viability of their standards using trade rags & Wall St.
 - Eventually we're stuck with 2 solutions or a protracted useless war (2 1/2-assed solutions != 1 whole solution)

15

Market Forces: Compatibility

- Case Studies:
 - Motif / Openlook (winner: Windows)
 - SSL / S-http (winner: SSL)
 - PGP / PEM (winner: PGP)
 - SKiP / ISAKMP (winner: ?)
 - SEPP / SETT / Cybercash (winner: ?)
 - Verisign / Entrust / etc.. (winner: ?)
 - Java / ActiveX (winner: ?)

16

Market Forces: Compatibility

- Loser: the customer
- Divide-and-conquer versus Grow-the-market-and-prosper has done more to delay the uptake of E-commerce than any other single factor
- It drives up costs and many just decide to wait until the dust settles (like they did for UNIX, ATM, X.500, and OSI)

17

Market Forces: Marketing

- Windows NT is **Secure** -- Byte Magazine says so!
- It took 25 years worth of UNIX security bugs to create a market perception that it is insecure
- It took 1 year of Microsoft marketing clout to create a market perception that NT is secure (but the reality is emerging)

18

Market Forces: Marketing

- Case Study: Lotus notes being sold as a “firewall” by one consultant
 - No need for it to actually be secure:
 - Make the promise
 - Grab their money
 - Promise fixes in future releases
 - Since you have their money, they’ll wait
 - Unless security re-emerges as a dirty word we’ll see it widely abused (“secure UPS!”)

19

Summary

- The market is not ripe for security
- Oddly, customers spent \$200million on security products in 1996
- Inefficiency breeds profits: in the land of the blind, the one-eyed man is king
 - In the security market, deliberately blinding your customers and competitors makes you a prince

20

Regulation: Crypto Export

- Cryptography is regulated as a munition
- Security is one of those fortunate technologies where technology and national defense interests intersect
 - Government has adopted a deliberate strategy to cool the market for any products containing cryptography
 - Net effect: security is undermined

21

Regulation: Crypto Export

- Case Study: 40-bit encryption in browsers
 - Crypto regulation limits exportable browsers to 40-bit key lengths
 - “Ok, go do electronic commerce using cryptography that the average house cat can crack”

22

Regulation: Patents

- Patent office is hopelessly naïve in keeping up with technology
- Patents granted contradict or overlap huge areas of technology
- Nowadays a patent is used as a defensive or offensive weapon (“shield patents” versus “hunting license”)
 - Small companies can’t afford to play

23

Regulation: Patents

- Case Study: A vendor is granted a patent on the idea of a meta-programmable packet switching and security inspection technology
 - Arguably, this is what routers have been doing for a long time
- Who wins?
 - Lawyers (co-incidentally the same clowns that wrote the rules!)

24

Summary

- The government didn't build the Internet (despite what Al Gore thinks)
- Internet technology ramp-up is faster than government comprehension/absorbtion rate!
 - This means “they” will never fully understand what's going on
 - This has real implications for security

25

....Ok, now let's look at some of the *technical* issues we face!!

26

Technical/People Bandwidth: Scope

- Security is an absolute game
- You **must** get all the details right: one hole is all it takes
 - People simply are not trained to think in terms of **whole** problems
 - People don't have time (brain bandwidth) enough to fix everything!
- The problem is too big: ignore it?

27

Technical/People Bandwidth: Scope

- Case Study: Network security
 - The guys who make the wire assume security is a protocol problem
 - The guys who designed the protocol assume security is an O/S problem
 - The guys who design the O/S assume it's an application problem
 - The guys who write the application rely on the IP address and clear transmissions

28

Technical/People Bandwidth: Ignorance

- Any idiot with a compiler can write the next killer app
 - Maybe (s)he has heard of the concept of network security
 - Most likely not
- Teach them to do it right, or fix it after it's broken?
 - Either is too expensive and impractical

29

Technical/People Bandwidth: Ignorance

- Case Study: HTTP
 - There are people who know how to design application protocols
 - HTTP wasn't designed by any of them - and it shows
- So let's adopt it as the basis for the future of E-commerce!***

30

Technical/People Bandwidth: Ignorance

- Get it right the first time
- or
- Get it wrong and then fix it

31

Technical/People Bandwidth: Testing

- “Internet Time” has killed the concept of software testing
 - Evolutionally speaking having high quality code is not a successful strategy!
 - Therefore having secure code is not a successful strategy!
- Many organizations rely on “beta test” code that isn’t even alpha test quality

32

Technical/People Bandwidth: Testing

- Case Study: Java
 - Research hack flung into the market in a flurry of hype
 - Nearly 2 years later it still randomly crashes wide varieties of browsers and has many problems with security
 - **But** - if Sun hadn't tossed Java over the fence we might be using something **worse!**
(like ActiveX)

33

Summary

- Implementing security in developing systems is a full-time job
- Security is “product friction” except in a very small market
- Formal approaches (certification, audit, orange book, etc.) would stifle innovation and destroy US domination of world software scene

34

Technical/Layering of Mistakes: No Security Model

- It's almost always impossible to retrofit a good security model onto something that was designed without one
- Everything layered above a system with no security model will be insecure
- Constant demand for features can stretch a model 'till it breaks

35

Technical/Layering of Mistakes: No Security Model

- Case Study: ActiveX

36

Technical/Mistakes: Bad defaults

- Majority of applications do not choose defaults that promote security
- Frequently there is a lack of feedback when an unsafe option is taken
 - In some cases it warns you but lets you specify “don’t pester me again”

37



38

Technical/Mistakes: Bad defaults

- Case Study: Windows Apps
 - Most Windows NT apps coded to cross-operate on Windows 95
 - Since Windows 95 has no security model guess what gets left out of all the NT apps?

39

Technical/Mistakes: Granularity of Control

- Software models don't give user enough feedback about what they propose to do to or on behalf of the user
 - Do it and suffer the consequences
 - or
 - Don't run it and never find out

40

Granularity of Control

Click **one**:

Click Here
and ***something***
will happen

Never mind:
I don't
trust you

41

Granularity of Control (cont)

- Case Study: The Web
 - Integrated point-and-click everything

42

Technical/Mistakes: Remote Management

- Everything is becoming networked
- Secure remote management doesn't exist
 - There are non-interoperable one-offs for specific products
- SNMP
 - Left security out
 - SNMP V2 also (couldn't agree on security parts of standard)

43

Technical/Mistakes: Remote Management

- Case Study: A certain firewall that shall remain nameless
 - System engineers tell customer to enable TELNET to firewall
 - ...then ***log in over the Internet*** to fix a configuration problem

44

Technical/Mistakes: Most Privilege

- Opposite of “Least Privilege”
- It takes more skill to write a program that runs with a minimum amount of privilege than to write one that runs as “root”
- Next generation of s/w engineers (the spawn of W95) grew up in an environment with no priv model at all!

45

Technical/Mistakes: Most Privilege

- Case Study: a vendor that remains nameless had Xterm setuid root so it could write `/etc/utmp`
 - It could also save its configuration information (as root) on top of any file in the system including `/etc/passwd`

46

Summary V1.0

- We're doomed

47

Summary: version 2.0

- We have job security

48

Summary: version 3.0

- Software industry is still in its infancy
- We haven't yet realized that code is potentially life-valuable and life-risking
- Safety technology usually comes to an industry after years of unbroken death and disaster
 - Cars introduced **1890's**, seatbelts **1970's**...

49