Open Standards and Security

David A. Wheeler
July 12, 2006

This presentation contains the views of the author and does not indicate endorsement by IDA, the U.S. government, or the U.S. Department of Defense.

Outline

- Stories (examples)
- Definitions: Key terms open systems & open standards
- Two ways open standards support security
  - Open standards support key technical approaches for security
  - More importantly: Open standards create economic conditions necessary for creating secure components
- Notes about open standards
  - Market changes, OSS, miscellaneous,open systems
- Conclusions

Open standards are necessary for security

A few stories...

- **Magic food (independence from supplier)**
  - **Only need food 1/year, all vitamins & minerals, first 1 $1**
  - **... but you can eat ONLY it from now on (others poison), and there's ONLY ONE manufacturer. Think the prices will go up? What's social cost of crack? Dependence is a security problem!**
  - **Not attacking MS/RH/etc. Need suppliers; not dependence on 1**
  - **Two IT independence strategies: Open standards & OSS (differ!)**
- **Firehose couplings (so defenders can cooperate)**
  - **1904 Baltimore fire: cities' couplings differ, 2,500 buildings lost**
  - **Multiple "standards" NOT good; multiple *implementations***
- **Railroad gauge – Contributed to Confederacy's loss**
  - **Eliminate unnecessary costs/time, freeing up money/time**
  - **Plug&play (cars/engines with tracks) allows innovation & improvement (steam→diesel). *No one organization* does all innovation. See also audio equipment**

Firehose couplings: Massive incompatibility

Glamorgan Pipe & Foundry (Pat 1897)

Hamilton Water Works 1859

Kennedy Valve 1890s

Crane Company c. 1900

Holyoke Iron Works 1890s

Source: http://www.firehydrant.org/pictures/oldermodels.html included as fair use (Transformative: changed purpose of work from focus on hydrants; Nature of work: non-fiction, non-art display of objects; Amount: small, not heart; Market: Non-commercial use, photos already displayed without fee, subset does not reduce value of original site.)

Open systems and open standards

- Goal isn't standards per se—goal is (modular) *open systems*
  - Open System = "A system that employs modular design, uses widely supported and consensus based standards for its key interfaces, and has been subjected to successful V&V tests to ensure the openness of its key interfaces". Open systems *depend* on open standards [DoD OSJTF]
  - *Competing marketplace of replaceable components*. "Standards exist to encourage & enable multiple implementations" [Walli]
- Governments widely view open systems as critically necessary
  - Extensive network of people with know-how (talk to them!)
  - U.S. DoD (serious emphasis: DISA, DISR, OSJTF...)
    - "shall be employed, where feasible." [DoD Directive 5000.1]
  - European Commission – major policy thrust
    - "guidance needs to focus on open standards"
- Advantages: Greater interoperability & flexibility, lower costs... focus today: security

What are open standards?

Not just "open mouth". Merged Perens'/Krechmer's/EC's definition:

1. **Availability: available for all to read and implement**
2. **Maximize End-User Choice: Create a fair, competitive market for implementations; NOT lock the customer in. Multiple implementors**
3. **No Royalty: Free for all to implement, with no royalty or fee**
4. **No Discrimination: Don't favor one implementor over another (open meeting, consensus/no domination, due process)**
5. **Extension or Subset: May be extended or offered in subset form**
6. **Predatory Practices: May employ license terms that protect against subversion of the standard by embrace-and-extend tactics**
7. **One World: Same standard for the same capability, world-wide**
8. **On-going Support: Supported until user interest ceases**
9. **No or nominal cost for specification (at *least*; open access?)**

See http://www.dwheeler.com/essays/opendocument-open.html

Two ways open standards support security

1. Open standards support key technical approaches for security
2. More importantly: Open standards create economic conditions necessary for creating secure components

Saltzer & Schroeder (1975): How open standards support security

- **Open design (open review)**
  - **Availability: Available for all to read and implement**
  - **Worldwide review can eliminate key problems**
- **Psychological acceptability / Easy to use**
  - **Familiarity is key**
  - **One world, maximize end-user choice**
- ***Modularity* helps with Economy of mechanism/Simplicity, Least privilege, Complete mediation, Separation of privilege, Least common mechanism (unshared)**
  - ***Any* standard defines a boundary, creating modularity**
  - **Maximize end-user choice, no discrimination**

Modularity: Key for security, and has other benefits too

- **“[MS' OS project] was restarted in the summer of 2004... it became clear [Longhorn] would not work. Two years' worth of work was scrapped... The new work, Microsoft decided, would take a new approach... [build] more in small modules that then fit together like Lego blocks.” -- NY Times, March 27, 2006, Lohr & Markoff, “Windows is So Slow, but Why?”**
- **“Complexity kills. It sucks the life out of developers, it makes products difficult to plan, build, and test, it introduces security challenges and it causes end-user and administrator frustration.” -- Ray Ozzie, CTO Microsoft**
- **Infrastructure managers have an even greater need for modularity (because an OS is only one small piece)**
- **“Composibility” issue, but that's a problem for monoliths too**

Open standards create necessary economic conditions for security

- **Without open standards, security evaporates**
  - **If high transition costs→stuck with supplier**
  - **Supplier raises profits by increasing prices while providing fewer benefits (inc. security)**
  - **Dependency→vulnerability→insecurity ("magic food")**
- **Open Standards make security *possible***
  - **Competing suppliers (continuously competing)**
  - **Can choose based on security & switch if security inadequate**
  - **Suppliers compete on their security, so improve**
  - **Allow connecting in new ways, so can cooperate for security ("firehose couplings")**
  - **Lowers costs over time (freeing budget), enables / encourages innovation (inc. security innovation) ("railroad gauge")**
- **True for COTS, custom, & mixed**

Leaders note the value of competition

- **Microsoft (MS)**
  - **"We welcome competition in the marketplace and believe it is healthy for the industry as a whole and good for customers." Erik Ryan, Senior marketing manager, March 2006**
  - **“There's a lot of industry competition... Openness to me means that anything can be cloned... no patents... no IP that stands in the way of somebody creating something that's compatible but better. And the beauty of that is that it forces you to keep prices extremely low and listen to the customer feedback about how you can do better” Bill Gates, 1996**
- **Red Hat (RH)**
  - **“with someone pushing us, we're going to have to leapfrog back and come up with a new set of technology. It's that competition in the marketplace that's causing [us to] innovate faster”... “If Novell does better job... than Red Hat... those customers will go to Novell...” Bob Young, co-founder RH, 1999 & 2005**

Standardization sometimes causes market lead changes

- Standards sometimes led by secondary suppliers
  - Dominant vendor often resists commoditization
  - Secondary competitors willing to standardize, innovation from competition can leapfrog past
  - “It is not necessarily the dominant vendor's product that is to be standardized, but the product market space” [Walli]
- Larger vendor, dominant position, and/or (initial) technical superiority typically not enough to resist standardization
  - Sony Betamax (lost to VHS)
  - DEC VAX VMS (lost to POSIX)
  - IBM SNA & Novell IPX/SPX & MS MSN/Blackbird & ... (lost to TCP/IP)

Relationship of open standards with open source software

- Open standards do not require use of OSS
  - Neutral on OSS vs. proprietary software
  - Yet there is a relationship between open standards & OSS
- Open standards aid OSS projects
  - Makes it easy for users to adopt an OSS program, because users not locked in – eases migration & integration
  - Simplifies OSS development (developers know what to do)
  - Open standards aid proprietary projects same way
- OSS aids open standards
  - OSS implementations help create & keep open standards open (reference model demos implementability & how, clarifies spec)
  - Rapidly increases use of open standard. "Implement by downloading" makes standard widespread, & downward cost pressure reigns in price of proprietary (increasing use)
  - Practice: Successful open standards have OSS implementation

Miscellanous

- Security is a process; need continuous deployment process
- Transition costs real but only happen once (tracks, hydrants)
  - Look at multi-year ROI (competitive bidding pressures costs!)
  - How long do you plan to be in business & doing that? Governments typically last a LONG time, needs don't go away
  - Tracks & hydrants: expensive & worth it
- Be pragmatic while transitioning from legacy
  - Strategize long-term (architectures that identify key interfaces, implementation plan); plan beyond this year's budget
  - Spec open standards, test for them, roll-out incrementally, DO IT
  - Web-based systems: Spec open standards, test with validators & multiple browsers (esp. Firefox) & platforms
  - For security, concentrate on replacing insecure with secure (old Sendmail/Exchange→Postfix, IE→Firefox, Outlook→anything), inc. security tests (fuzz, injection, cleartext passwd, etc.), secure "inside" systems, reuse tests!

Open standards key enabler for the larger goal: Modular open systems

- **OSJTF identified 5 principles of (modular) open systems:**
  1. **Establish an enabling environment – supportive requirements, strategies, business practices. DoD reviews programs, lose funding if don't support it!**
  2. **Employ modular design – develop architectures based on modular design tenets (don't just "buy stuff")**
  3. **Designate key interfaces – identify interfaces impacting performance/cost/support**
  4. **Use open standards – consensus based, wide support**
  5. **Certify compliance – assure openness (test replaceability)**
- **Must create a plan to do this; focus on *goal***
- **Balance: top-down *and* bottom-up**

Source: Open Systems Joint Task Force (OSJTF)
http://www.acq.osd.mil/osjtf/

Conclusions

- **Two ways open standards support security:**
  1. **Open standards support key technical approaches for security**
  2. **More importantly: Open standards create economic conditions necessary for creating secure components and systems (in the long run)**
- **Remember the stories**
  - **Magic food (independence from supplier)**
  - **Firehose couplings (needed so defenders can coop)**
  - **Railroad gauge (eliminate unnecessary costs/time, competition via modularity yields innovation/ improvement inc. security)**

Open standards are necessary for security

Acronyms

COTS – Commercial Off-The-Shelf
CTO – Chief Technical Officer
DISA – Defense Information Systems Agency
DISR – DoD IT Standards Registry (DISR)
DoD – Department of Defense
IP – Intellectual Property (aka Intellectual Rights)
IT – Information Technology
MS – Microsoft
MSN – Microsoft Network
OS – Operating System
OSS – Open Source Software (aka FLOSS)
OSJTF – Open Systems Joint Task Force
RH – Red Hat
ROI – Return on Investment
V&V – Verification and Validation

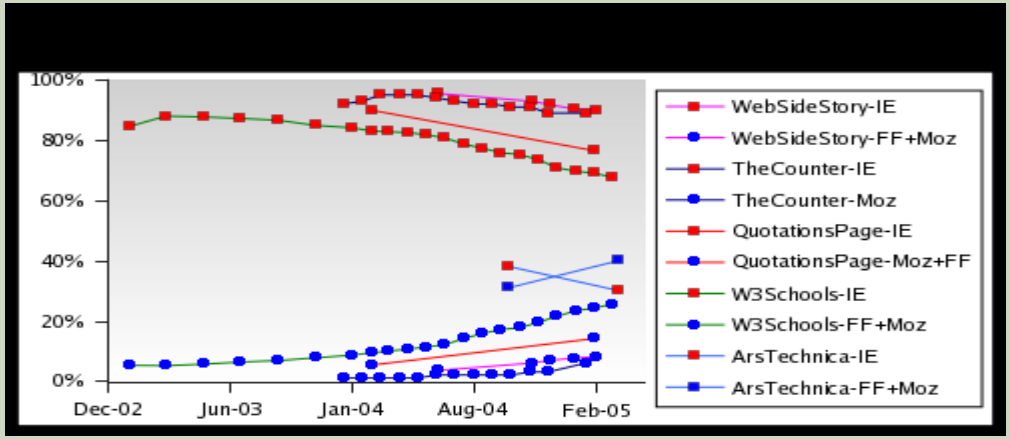
For More Information

- ANSI, “Through History with Standards” http://www.ansi.org/consumer_affairs/history_standards.aspx
- European Commission, *European Interoperability Framework.* http://europa.eu.int/idabc/en/document/3761
- Lohr & Markoff, “Windows is So Slow, but Why?”, NY Times, March 27, 2006
- Open Systems Joint Task Force (OSJTF) Web site, http://www.acq.osd.mil/osjtf/
- Perens, Bruce. “Open Standards: Principles and Practice”. http://perens.com/OpenStandards/Definition.html
- Puffert, Douglas. “Path Dependence in Spatial Networks: The Standardization of Railway Track Gauge”
- Walli, Stephen R. “Under the Hood: Open Source and Open Standards Business Models in Context” *Open Sources 2.0*. Ed. Chris diBona et al. O'Reilly. 2005.
- Wheeler, David A. *Is OpenDocument an Open Standard? Yes!* http://www.dwheeler.com/essays/opendocument-open.html

Backup

Case Study: Mozilla Firefox vs. Internet Explorer

- Leading web browser has been Internet Explorer (IE)
- IE serious security problems; Mozilla Firefox released. HTTP, HTML
  - Browser "unsafe" days in 2004: 98% IE, 15% Firefox* [Scanit]
  - IE 21x more likely to get spyware vs. Firefox [U of Wash.]
  - Faster response: Firefox 37 days, Windows 134.5 days [W. Post]
- "Stronger Security" key supplier pitch & switching rationale
- Firefox costs more (both free, but IE pre-installed), yet use grown
- IE development restarted, a stated focus is security. Competition!



Red: IE
Blue: Mozilla (inc. Firefox)

*1/2 Mac-only

OpenDocument

- What is it?
- Standardization
- Adoption
- Is it an open standard?
- Security and office implementations

Who owns your data? A vendor, or you?

What's OpenDocument (ODF)?

- **Fully open standard for exchanging office documents between arbitrary programs**
  - **Word processing (.odt), presentation (.odp), spreadsheet (.ods), graphics (.odg), ...**
  - **Full capabilities (formatting styles, charts, math formulas, templates, Ruby, etc.)**
  - **Zip-compressed XML format: Small & easily processed**
  - **Reuses standards (MathML, SVG, SMIL, XForms, etc.)**
- **Goal: Users can own & control their own information**
  - **Proprietary formats: Vendor owns your data**
  - **ODF: can use different office suites, store long-term**
  - **"we cannot have our public documents locked up in [a] proprietary format, perhaps unreadable in the future, or subject to a ... license that restricts access." [Kriss]**

OpenDocument Standardization

- OASIS OpenDocument group begins 2002-12-16
  - Chose OpenOffice.org 1 format as base; only featureful office suite with native XML (2000)
- Participants: Many implementors & users, inc.:
  - Adobe, Arbortext, Corel (WordPerfect), IBM (Lotus 1-2-3, Workplace), KDE (Koffice), Sun (StarOffice/OpenOffice.org)
  - Boeing & Intel (complex large documents), National Archives of Australia & NY Attorney General (long-term storage), Novell, Society of Biblical Literature (multilingual, long-term), Sony
- Universal intermediate data format for legacy systems (inc. MS Office, and more)
- ODF becomes OASIS standard: 2005-05-01
- ISO/IEC 26300 approved 2006-05-03

OpenDocument Adoption

- EU analysis, tells MS to join OpenDocument group 2004-04
- ODF becomes OASIS standard: 2005-05-01
- OpenOffice.org/StarOffice, Koffice: ODF as native format
- MA adopts ODF (not MS) on 2005-09 for deployment 2007-01
  - Nasty big-money political infighting fails to stop
- Microsoft creates ECMA group to create competing standard 2005-12-09 as ODF derail attempt
- National Archives of Australia selects ODF 2006-03-31
- ISO/IEC 26300 approved 2006-05-03
- Belgium adopts ODF 2006-06-23, rejecting MS format; all docs in ODF by 2008-09 (readable 2007-09)
  - Other EU countries expected to follow
- Microsoft caves 2006-07-06, announces it will create OpenDocument implementation for MS Office (caveats!)

OpenDocument an Open Standard? YES.

1. Availability: Yes, any can implement
2. Maximize End-User Choice: Yes, competing implementations; multiple implementors created it
3. No Royalty: Yes
4. No Discrimination: Yes, no favored vendor*
5. Extension or Subset: Yes
6. Predatory Practices: Yes (vacuously)
7. One World: Yes
8. On-going Support: Yes, not one vendor
9. No or nominal cost for specification: Yes, no cost

* As shown by open meetings, due process, & consensus processes, as well as evidence of technical changes created by all that affect implementors

Security is irrelevant for office software, right? :-)

- **“A hole in Microsoft Excel has been identified that could allow attackers to take control of a computer, a security group said Thursday--the third vulnerability affecting the popular spreadsheet program to surface in less than a month.” *ZDNet*, 2006-07-06, “Another security hole found in Excel”**
- **“Microsoft plans to issue patches for critical Windows and Office security problems as part of a regular update scheduled for Tuesday”, *ZDNet*, 2006-07-06, “Windows, Office to get 'critical' fixes”**

Security and Document Format Standards

- Open standard lets users choose
  - HTML is an open standard
  - IE lost significant market share as users switched to Firefox to get its better security
- Competition via open standards force improvements
  - IE languished for 5 years, endless vulnerabilities; Firefox caused IE security reviews & design changes
- Closed standard prevents choice
  - Users constantly patch Office, hoping that they'll get ahead of the attackers this time
- Text formats also show we can agree on formats
  - Nobody cares which text editor you use; “just works”
  - ASCII etc. isn't the original market leader (EBCDIC)

OpenDocument

- **Who owns your data?**
  - **Can you easily switch and interchange between different competing vendors?**
  - **If not, you have a problem. OpenDocument's purpose is to solve it**

Open Standards and Security

David A. Wheeler
July 12, 2006

This presentation contains the views of the author and does not indicate endorsement by IDA, the U.S. government, or the U.S. Department of Defense.

1

This is a briefing about open standards and security. In particular, it explains why open standards are *necessary* for security. The briefing is much better in person; I add more information than is written in the presentation (which is essentially just an outline), and I use props to help illustrate my points (making the presentation more interesting and memorable). Still, if you can't be at my presentation, reading this presentation is a start. My thanks to James Martin of the (Washington) District of Columbia Fire Department for providing me with some sample firehose couplings. These couplings help illustrate (in a very concrete way) why standards are important. People have died from lack of standards; let's help make that situation a historical footnote instead of current practice.

Outline

- **Stories (examples)**
- **Definitions: Key terms open systems & open standards**
- **Two ways open standards support security**
  - **Open standards support key technical approaches for security**
  - **More importantly: Open standards create economic conditions necessary for creating secure components**
- **Notes about open standards**
  - **Market changes, OSS, miscellaneous,open systems**
- **Conclusions**

Open standards are necessary for security

2

This is the outline of my presentation. I'll start with a few stories, which I'll use as examples, and define two key terms: open systems and open standards.

I then introduce the meat of the presentation – the two ways that open standards support security. First, open standards support key technical approaches for security. And second – and more importantly – open standards create the economic conditions necessary for creating secure components in the long term. I then have a few various notes and conclusions.

If you can remember nothing else from this presentation, remember this: Open standards are *necessary* for security.

A few stories...

- **Magic food (independence from supplier)**
  - Only need food 1/year, all vitamins & minerals, first 1 $1
  - ... but you can eat ONLY it from now on (others poison), and there's ONLY ONE manufacturer. Think the prices will go up? What's social cost of crack? Dependence is a security problem!
  - Not attacking MS/RH/etc. Need suppliers; not dependence on 1
  - Two IT independence strategies: Open standards & OSS (differ!)
- **Firehose couplings (so defenders can cooperate)**
  - 1904 Baltimore fire: cities' couplings differ, 2,500 buildings lost
  - Multiple "standards" NOT good; multiple *implementations*
- **Railroad gauge – Contributed to Confederacy's loss**
  - Eliminate unnecessary costs/time, freeing up money/time
  - Plug&play (cars/engines with tracks) allows innovation & improvement (steam→diesel). *No one organization* does all innovation. See also audio equipment

3

Here are three stories to illustrate the need for standards: "magic food" (demonstrating the need for independence from a supplier), firehose couplings, and railroad track width. More details about the latter two are in

http://www.ansi.org/consumer_affairs/history_standards.aspx?menuid=5:

"In 1904, a fire broke out in the basement of the John E. Hurst & Company Building in Baltimore. After taking hold of the entire structure, it leaped from building to building until it engulfed an 80-block area of the city. To help combat the flames, reinforcements from New York, Philadelphia and Washington, DC immediately responded—but to no avail. Their fire hoses could not connect to the fire hydrants in Baltimore because they did not fit the hydrants in Baltimore. Forced to watch helplessly as the flames spread, the fire destroyed approximately 2,500 buildings and burned for more than 30 hours." US standard is 11 threads/inch.

In the U.S. Civil War, "[The North] had woven an integrated rail system.. basically, all gauges were the same. [The South's] were of different gauges, and few lines were connected... a real reason [the South lost] the Civil War." [Ned Harrison, "'States Rights' doomed Confederate nation", Nov. 12, 2005] "the Railroad was a fast, economical and effective means of sending products cross-country. This feat was made possible by the standardization of the railroad gauge, which established the uniform distance between two rails on a track... During the Civil War the U.S. government recognized the military and economic advantages to having a standardized track gauge. The government worked with the railroads to promote use of the most common railroad gauge... 4 feet, 8 ½ inches, a track size that originated in England. This gauge was mandated for use in the Transcontinental Railroad in 1864 and by 1886 had become the U.S. Standard."

Other examples of standards include U.S. Highways and audio equipment.

Firehose couplings: Massive incompatibility

Glamorgan Pipe & Foundry (Pat 1897)

Hamilton Water Works 1859

Kennedy Valve 1890s

Crane Company c. 1900

Holyoke Iron Works 1890s

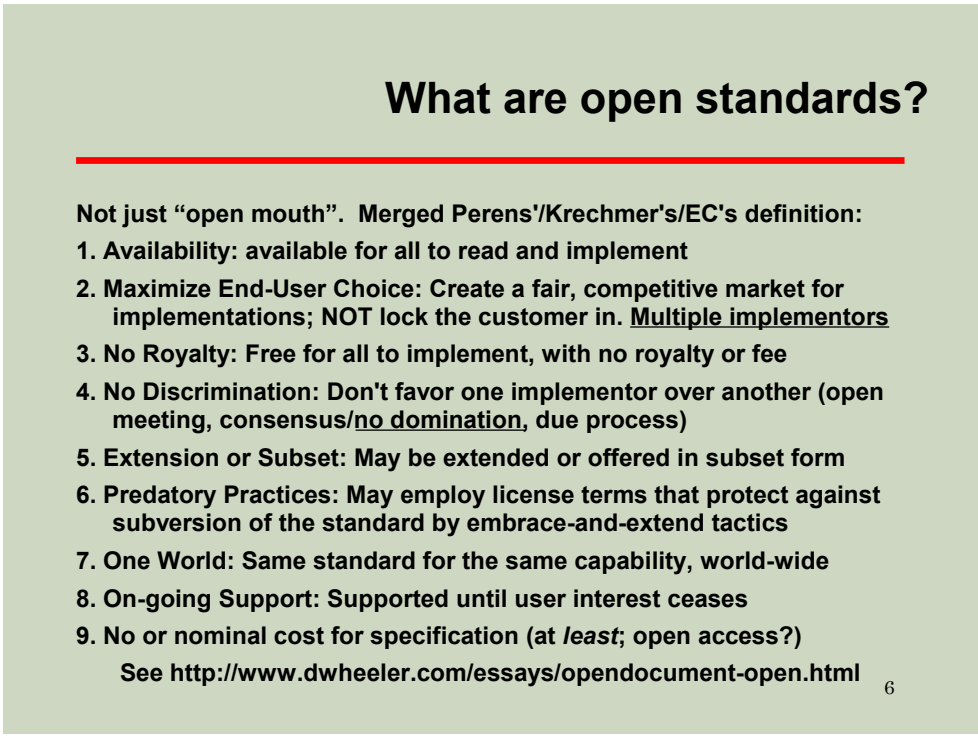
Source: http://www.firehydrant.org/pictures/oldermodels.html included as fair use (Transformative: changed purpose of work from focus on hydrants; Nature of work: non-fiction, non-art display of objects; Amount: small, not heart; Market: Non-commercial use, photos already displayed without fee, subset does not reduce value of original site.)

4

Here are some pictures of pre-1904 fire hydrants. A firehose coupling for one was not compatible with another.

Open systems and open standards

- Goal isn't standards per se—goal is (modular) *open systems*
  - Open System = "A system that employs modular design, uses widely supported and consensus based standards for its key interfaces, and has been subjected to successful V&V tests to ensure the openness of its key interfaces". Open systems *depend* on open standards [DoD OSJTF]
  - *Competing marketplace of replaceable components*. "Standards exist to encourage & enable multiple implementations" [Walli]
- Governments widely view open systems as critically necessary
  - Extensive network of people with know-how (talk to them!)
  - U.S. DoD (serious emphasis: DISA, DISR, OSJTF...)
    - "shall be employed, where feasible." [DoD Directive 5000.1]
  - European Commission – major policy thrust
    - "guidance needs to focus on open standards"
- Advantages: Greater interoperability & flexibility, lower costs... focus today: security



It's been said that "any vendor with an open mouth has an open system." Thankfully, various smart people have worked to figure out what customers *mean* when they say they want open standards. The most popular definition of the term is Bruce Perens'; other popular ones include Ken Krechmer's and the European Commission's. They are all trying to describe the same thing, so by merging their individual definitions, I think we end up with a better, more encompassing definition than any one by itself (each one notes important issues another doesn't).

See http://www.dwheeler.com/essays/opendocument-open.html for an explanation of how each of these points was derived, and an example of their application to a real standard.

In the late 1980s there was some debate about the "no royalty" requirement, but that's widely accepted now in the software arena. After all, open standards must not discriminate, and royalty-bearing standards always discriminate (they completely prevent open source software implementations, which are dominant or #2 in a vast number of markets, and they always discriminate against *any* developer outside the club of those who hold some of the patents requiring royalties).

Nowadays the question is on specification costs. At one time, it was considered acceptable to charge fees for specifications, but as third world countries increasingly use standards, more and more standards are used for an application, and the need publication fees has completely vanished, this is becoming unjustifiable. The IETF, W3C, OASIS, and others have shown that there is no need for fees to acquire standards, and in a few years' time it will probably become *required* that open standards be available for free (and possibly as open source specifications). This is another example of the basic "no discrimination" requirement creating a new derived requirement, as more people use standards.

Two ways open standards support security

1. Open standards support key technical approaches for security
2. More importantly: Open standards create economic conditions necessary for creating secure components

Saltzer & Schroeder (1975): How open standards support security

- **Open design (open review)**
  - **Availability: Available for all to read and implement**
  - **Worldwide review can eliminate key problems**
- **Psychological acceptability / Easy to use**
  - **Familiarity is key**
  - **One world, maximize end-user choice**
- ***Modularity* helps with Economy of mechanism/Simplicity, Least privilege, Complete mediation, Separation of privilege, Least common mechanism (unshared)**
  - ***Any* standard defines a boundary, creating modularity**
  - **Maximize end-user choice, no discrimination**

8

Some people say that "no one knows how to develop secure software", and that is simply nonsense. The basic principles of how to develop secure software were developed in the mid-1970s and have not been debunked since. A seminal work is Saltzer & Schroeder of 1975, which identified a set of basic principles. Of those principles, it turns out that a number of them are supported by open standards.

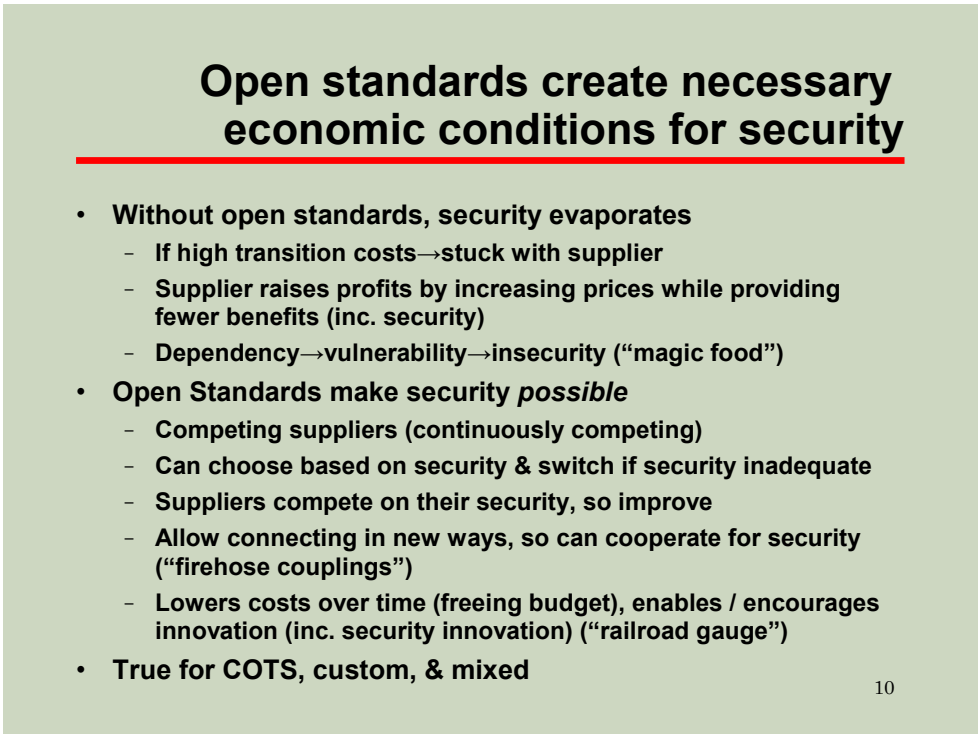
Most obviously, the principle of open design (leading to open review) is fundamentally supported in the development of open standards.

Another key principle is "psychological acceptability / ease of use". It turns out that the best predictor of ease of use is familiarity... and when components implement open standards, they quickly adopt similar concepts that lead to familiarity.

Many of their other principles can be summed up as *modularity* (at least in the main); any standard defines a boundary, and thus creates modularity.

Modularity: Key for security, and has other benefits too

- **"[MS' OS project] was restarted in the summer of 2004... it became clear [Longhorn] would not work. Two years' worth of work was scrapped... The new work, Microsoft decided, would take a new approach... [build] more in small modules that then fit together like Lego blocks." -- NY Times, March 27, 2006, Lohr & Markoff, "Windows is So Slow, but Why?"**
- **"Complexity kills. It sucks the life out of developers, it makes products difficult to plan, build, and test, it introduces security challenges and it causes end-user and administrator frustration." -- Ray Ozzie, CTO Microsoft**
- **Infrastructure managers have an even greater need for modularity (because an OS is only one small piece)**
- **"Composibility" issue, but that's a problem for monoliths too**



Although there are technical advantages for open standards, they are not the most important. The *real* key is that open standards create the economic conditions that make it *possible* to implement secure systems.

Without open standards, security evaporates. If a given product has key proprietary interfaces that only one supplier can implement (e.g., because they're secret, or the necessary expertise is in only one supplier), then as you build that product into your larger infrastructure, all of your other components may end up depending on these proprietary interfaces. The key is whether or not you can reasonably switch to another supplier, or if you are locked into that supplier because of the proprietary interfaces. If, over time, you become locked into that supplier, then that supplier will have no incentive to charge reasonable prices or provide benefits (like security). Even if things start off well, such "lock-ins" create pressures that subvert all attempts to achieve or improve security. The issue is not if there's *some* proprietary interface, or *some* open standard... the issue is lock-in to a supplier.

In contrast, open standards make security possible. They enable continuous competition between suppliers, so you can now choose the supplier who provides adequate security. Suppliers now compete on security, so they will have to improve. The key is using open standards appropriately to enable competition.

Note that this is true for commercial off-the-shelf (COTS), custom products, and mixtures. In theory, custom products with non-standard interfaces could be secure, at least in the short term, because a government can "always" keep paying to keep the custom component secure. But in practice, that isn't true. Custom suppliers, once they've locked in their customers, will have the same incentives to lower benefits and increase prices, and governments have limited funds. Even custom components must appropriately employ open standards to make security affordable.

Leaders note the value of competition

- **Microsoft (MS)**
  - **"We welcome competition in the marketplace and believe it is healthy for the industry as a whole and good for customers." Erik Ryan, Senior marketing manager, March 2006**
  - **"There's a lot of industry competition... Openness to me means that anything can be cloned... no patents... no IP that stands in the way of somebody creating something that's compatible but better. And the beauty of that is that it forces you to keep prices extremely low and listen to the customer feedback about how you can do better" Bill Gates, 1996**
- **Red Hat (RH)**
  - **"with someone pushing us, we're going to have to leapfrog back and come up with a new set of technology. It's that competition in the marketplace that's causing [us to] innovate faster"... "If Novell does better job... than Red Hat... those customers will go to Novell..." Bob Young, co-founder RH, 1999 & 2005**

11

Erik Ryan quote from http://news.com.com/Google+buys+Web+word-processing+technology/2100-1032-6048136.html

Bill Gates quote from Massachusetts Institute of Technology Distinguished Lecture Series 1996, Bill Gates Keynote Address, Wednesday May 30th, 1996, "http://www.microsoft.com/billgates/speeches/internet/mit/mit.asp" Most would disagree with parts of his definition of openness. The point of openness is not just that you can "clone" something; that would leave the "clonee" free to dictate every next change to the industry, and thus would be fundamentally discriminatory. However, he notes the strong value of competition without software patents or other barriers to compatibility, because they cause prices to be low and suppliers to listen to their customers.

Bob Young interviews: "Interview: Robert Young of Red Hat Software" by Marjorie Richardson, 1999-08-01, Linux Journal (http://www.linuxjournal.com/article/3553), "Interview: Bob Young after Red Hat", November 08, 2005, by Joe 'Zonker' Brockmeier, Newsforge, http://business.newsforge.com/article.pl?sid=05/11/01/1534231&tid=3&pagenum=3

Standardization sometimes causes market lead changes

- **Standards sometimes led by secondary suppliers**
  - **Dominant vendor often resists commoditization**
  - **Secondary competitors willing to standardize, innovation from competition can leapfrog past**
  - **"It is not necessarily the dominant vendor's product that is to be standardized, but the product market space" [Walli]**
- **Larger vendor, dominant position, and/or (initial) technical superiority typically not enough to resist standardization**
  - **Sony Betamax (lost to VHS)**
  - **DEC VAX VMS (lost to POSIX)**
  - **IBM SNA & Novell IPX/SPX & MS MSN/Blackbird & ... (lost to TCP/IP)**

12

Walli also notes that "standards develop once the marketplace reaches a point where the market leader begins to overdeliver." It sometimes happens even sooner (e.g., for videotapes).

The last point is especially interesting. IBM had its own proprietary network system (SNA), Novell had its own proprietary local area network system (IPX/SPX), and Microsoft had been pushing its own proprietary network for years, then known as Microsoft Network and project Blackbird. The Microsoft effort is often not even remembered, but up through 1995-1996 this was a major effort by Microsoft. A few references on Microsoft's work include David Brancaccio's December 7, 1995 report in "Marketplace" http://marketplace.publicradio.org/shows/1995/12/07_mpp.html, and "Microsoft Ends Plans for Tool Designed For On-Line Service as It Shifts Focus" by Don Clark, Wall Street Journal; copy available at http://www.anu.edu.au/mail-archives/link/link9602/0015.html

The reasons the Microsoft effort (in which Microsoft invested significant resources) is not often remembered can only be speculated. The name "MSN" (Microsoft Network) was later reused for another project, so many in the general public may have confused them. Also, while Microsoft had dedicated significant resources to their proprietary network, the world around was installing TCP/IP drivers and using them, and did not treat Microsoft's efforts seriously, so much so that Microsoft's efforts to control the network had no significant commercial effect. This rejection was the fundamental reason Microsoft changed direction. When Microsoft realized that everyone was moving towards the TCP/IP open standards instead of their proprietary standards, Microsoft made a major push to support TCP/IP.

Relationship of open standards with open source software

- Open standards do not require use of OSS
  - Neutral on OSS vs. proprietary software
  - Yet there is a relationship between open standards & OSS
- Open standards aid OSS projects
  - Makes it easy for users to adopt an OSS program, because users not locked in – eases migration & integration
  - Simplifies OSS development (developers know what to do)
  - Open standards aid proprietary projects same way
- OSS aids open standards
  - OSS implementations help create & keep open standards open (reference model demos implementability & how, clarifies spec)
  - Rapidly increases use of open standard. "Implement by downloading" makes standard widespread, & downward cost pressure reigns in price of proprietary (increasing use)
  - Practice: Successful open standards have OSS implementation

In practice, successful open standards today have at least one OSS implementation. In some sense this is very surprising; there is no fundamental requirement that an open standard must have an OSS implementation. For example, almost no organization that creates open standards requires this, and many open standards are based on proprietary implementations (not OSS ones). However, I have been unable to find even one example of a successful open standard without an OSS implementation. In contrast, typical examples of open standards (such as TCP/IP, HTTP, HTML, DNS, SMTP, POP3, IMAP, IPSec, SSH, SSL, and OpenDocument) all have at least one OSS implementation. Even specialty open standards and/or those developed by non-traditional standards bodies (who by definition must be non-discriminatory) have OSS implementations (such as Z39.50 for search queries or GEDCOM for exchanging genealogical data). There may be a successful open standard without an OSS implementation, but the difficulty in finding an example suggests that having an OSS implementation is the rule, not the exception.

I can only speculate why this might be true. An OSS implementation can help ensure that a standard is implementable (as a reference model), clarify how to implement the standard for other implementors, and help prove that the license is non-discriminatory. Many standards projects, even those with proprietary developers as part of the project, have explicitly used an OSS implementation as a testbed to ensure that the standard is implementable, verify its effectiveness, and/or clarify the standard. The low cost of OSS implementations may help make the standard widespread, and forcing the cost of proprietary implementations may encourage use further (since lower prices are likely to increase use). It may simply be that OSS is so widespread that any customer need is likely to have an OSS implementation, and having OSS implementations of open standards is simply a side-effect. Whatever the reason, successful open standards are generally implemented by at least one OSS implementation.

Miscellanous

- Security is a process; need continuous deployment process
- Transition costs real but only happen once (tracks, hydrants)
  - Look at multi-year ROI (competitive bidding pressures costs!)
  - How long do you plan to be in business & doing that? Governments typically last a LONG time, needs don't go away
  - Tracks & hydrants: expensive & worth it
- Be pragmatic while transitioning from legacy
  - Strategize long-term (architectures that identify key interfaces, implementation plan); plan beyond this year's budget
  - Spec open standards, test for them, roll-out incrementally, DO IT
  - Web-based systems: Spec open standards, test with validators & multiple browsers (esp. Firefox) & platforms
  - For security, concentrate on replacing insecure with secure (old Sendmail/Exchange→Postfix, IE→Firefox, Outlook→anything), inc. security tests (fuzz, injection, cleartext passwd, etc.), secure "inside" systems, reuse tests!

Open standards key enabler for the larger goal: Modular open systems

- OSJTF identified 5 principles of (modular) open systems:
  1. Establish an enabling environment – supportive requirements, strategies, business practices. DoD reviews programs, lose funding if don't support it!
  2. Employ modular design – develop architectures based on modular design tenets (don't just "buy stuff")
  3. Designate key interfaces – identify interfaces impacting performance/cost/support
  4. Use open standards – consensus based, wide support
  5. Certify compliance – assure openness (test replaceability)
- Must create a plan to do this; focus on *goal*
- Balance: top-down *and* bottom-up

Source: Open Systems Joint Task Force (OSJTF)
http://www.acq.osd.mil/osjtf/

Conclusions

- **Two ways open standards support security:**
  1. **Open standards support key technical approaches for security**
  2. **More importantly: Open standards create economic conditions necessary for creating secure components and systems (in the long run)**
- **Remember the stories**
  - **Magic food (independence from supplier)**
  - **Firehose couplings (needed so defenders can coop)**
  - **Railroad gauge (eliminate unnecessary costs/time, competition via modularity yields innovation/ improvement inc. security)**

Open standards are necessary for security

Acronyms

COTS – Commercial Off-The-Shelf
CTO – Chief Technical Officer
DISA – Defense Information Systems Agency
DISR – DoD IT Standards Registry (DISR)
DoD – Department of Defense
IP – Intellectual Property (aka Intellectual Rights)
IT – Information Technology
MS – Microsoft
MSN – Microsoft Network
OS – Operating System
OSS – Open Source Software (aka FLOSS)
OSJTF – Open Systems Joint Task Force
RH – Red Hat
ROI – Return on Investment
V&V – Verification and Validation

For More Information

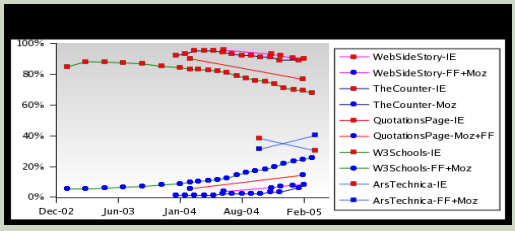
- **ANSI, "Through History with Standards" http://www.ansi.org/consumer_affairs/history_standards.aspx**
- **European Commission, *European Interoperability Framework.* http://europa.eu.int/idabc/en/document/3761**
- **Lohr & Markoff, "Windows is So Slow, but Why?", NY Times, March 27, 2006**
- **Open Systems Joint Task Force (OSJTF) Web site, http://www.acq.osd.mil/osjtf/**
- **Perens, Bruce. "Open Standards: Principles and Practice". http://perens.com/OpenStandards/Definition.html**
- **Puffert, Douglas. "Path Dependence in Spatial Networks: The Standardization of Railway Track Gauge"**
- **Walli, Stephen R. "Under the Hood: Open Source and Open Standards Business Models in Context" *Open Sources 2.0*. Ed. Chris diBona et al. O'Reilly. 2005.**
- **Wheeler, David A. *Is OpenDocument an Open Standard? Yes!* http://www.dwheeler.com/essays/opendocument-open.html**

Backup

- Click to add an outline

Case Study: Mozilla Firefox vs. Internet Explorer

- Leading web browser has been Internet Explorer (IE)
- IE serious security problems; Mozilla Firefox released. HTTP, HTML
  - Browser "unsafe" days in 2004: 98% IE, 15% Firefox* [Scanit]
  - IE 21x more likely to get spyware vs. Firefox [U of Wash.]
  - Faster response: Firefox 37 days, Windows 134.5 days [W. Post]
- "Stronger Security" key supplier pitch & switching rationale
- Firefox costs more (both free, but IE pre-installed), yet use grown
- IE development restarted, a stated focus is security. Competition!



Red: IE
Blue: Mozilla (inc. Firefox)

*1/2 Mac-only

OpenDocument

- What is it?
- Standardization
- Adoption
- Is it an open standard?
- Security and office implementations

Who owns your data? A vendor, or you?

What's OpenDocument (ODF)?

- **Fully open standard for exchanging office documents between arbitrary programs**
  - **Word processing (.odt), presentation (.odp), spreadsheet (.ods), graphics (.odg), ...**
  - **Full capabilities (formatting styles, charts, math formulas, templates, Ruby, etc.)**
  - **Zip-compressed XML format: Small & easily processed**
  - **Reuses standards (MathML, SVG, SMIL, XForms, etc.)**
- **Goal: Users can own & control their own information**
  - **Proprietary formats: Vendor owns your data**
  - **ODF: can use different office suites, store long-term**
  - **"we cannot have our public documents locked up in [a] proprietary format, perhaps unreadable in the future, or subject to a ... license that restricts access." [Kriss]**

OpenDocument Standardization

- OASIS OpenDocument group begins 2002-12-16
  - Chose OpenOffice.org 1 format as base; only featureful office suite with native XML (2000)
- Participants: Many implementors & users, inc.:
  - Adobe, Arbortext, Corel (WordPerfect), IBM (Lotus 1-2-3, Workplace), KDE (Koffice), Sun (StarOffice/OpenOffice.org)
  - Boeing & Intel (complex large documents), National Archives of Australia & NY Attorney General (long-term storage), Novell, Society of Biblical Literature (multilingual, long-term), Sony
- Universal intermediate data format for legacy systems (inc. MS Office, and more)
- ODF becomes OASIS standard: 2005-05-01
- ISO/IEC 26300 approved 2006-05-03

OpenDocument Adoption

- EU analysis, tells MS to join OpenDocument group 2004-04
- ODF becomes OASIS standard: 2005-05-01
- OpenOffice.org/StarOffice, Koffice: ODF as native format
- MA adopts ODF (not MS) on 2005-09 for deployment 2007-01
  - Nasty big-money political infighting fails to stop
- Microsoft creates ECMA group to create competing standard 2005-12-09 as ODF derail attempt
- National Archives of Australia selects ODF 2006-03-31
- ISO/IEC 26300 approved 2006-05-03
- Belgium adopts ODF 2006-06-23, rejecting MS format; all docs in ODF by 2008-09 (readable 2007-09)
  - Other EU countries expected to follow
- Microsoft caves 2006-07-06, announces it will create OpenDocument implementation for MS Office (caveats!)

OpenDocument an Open Standard? YES.

1. Availability: Yes, any can implement
2. Maximize End-User Choice: Yes, competing implementations; multiple implementors created it
3. No Royalty: Yes
4. No Discrimination: Yes, no favored vendor*
5. Extension or Subset: Yes
6. Predatory Practices: Yes (vacuously)
7. One World: Yes
8. On-going Support: Yes, not one vendor
9. No or nominal cost for specification: Yes, no cost

* As shown by open meetings, due process, & consensus processes, as well as evidence of technical changes created by all that affect implementors

Security is irrelevant for office software, right? :-)

- **“A hole in Microsoft Excel has been identified that could allow attackers to take control of a computer, a security group said Thursday--the third vulnerability affecting the popular spreadsheet program to surface in less than a month.” *ZDNet*, 2006-07-06, “Another security hole found in Excel”**
- **“Microsoft plans to issue patches for critical Windows and Office security problems as part of a regular update scheduled for Tuesday”, *ZDNet*, 2006-07-06, “Windows, Office to get 'critical' fixes”**

Security and Document Format Standards

- Open standard lets users choose
  - HTML is an open standard
  - IE lost significant market share as users switched to Firefox to get its better security
- Competition via open standards force improvements
  - IE languished for 5 years, endless vulnerabilities; Firefox caused IE security reviews & design changes
- Closed standard prevents choice
  - Users constantly patch Office, hoping that they'll get ahead of the attackers this time
- Text formats also show we can agree on formats
  - Nobody cares which text editor you use; “just works”
  - ASCII etc. isn't the original market leader (EBCDIC)

OpenDocument

- **Who owns your data?**
  - **Can you easily switch and interchange between different competing vendors?**
  - **If not, you have a problem. OpenDocument's purpose is to solve it**