Evolution of Information Security Design Principles

This is an extended, less-edited version of an article appearing in ***IEEE Security and Privacy*** *in December 2012. This version specifically identifies all of the textbooks I reviewed while looking at information security design principles.*

Here is the citation for the published article:

> Smith, R.E.; , "A Contemporary Look at Saltzer and Schroeder's 1975 Design Principles," *Security & Privacy, IEEE* , vol.10, no.6, pp.20-25, Nov.-Dec. 2012
> doi: 10.1109/MSP.2012.85

The information security community has a rich legacy of wisdom drawn from earlier work and from sharp observations. Not everyone is old enough or fortunate enough to have encountered this legacy first-hand by working on groundbreaking developments. Many of us receive it from colleagues or through readings and textbooks.

The Multics time-sharing system (Figure 1 - photo by Tom Van Vleck) was an early multi-user system that put significant effort into ensuring security. In 1974, Jerome Saltzer wrote an article outlining the security mechanisms in the Multics system (Saltzer, 1974). The article included a list of five "design principles" he saw reflected in his Multics experience. The following year, Saltzer and Michael Schroeder expanded the article into a tutorial titled "The Protection of Information in Computer Systems" (Saltzer and Schroeder, 1975). The first section of the paper introduced "basic principles" of information protection, including the triad of confidentiality, integrity, and availability, and a set of design principles.

Over the following decades, these principles have occasionally been put forth as guidelines for developing secure systems. Most of the principles found their way into the DOD's standard for computer security, the *Trusted Computer System Evaluation Criteria* (NCSC, 1985). The Saltzer and Schroeder design principles were also highlighted in security textbooks, like Pfleeger's *Security in Computing* (Pfleeger, 1989), the first edition of which appeared in 1989.

Different writers use the term *principle* differently. Some apply the term to a set of precisely worded statements, like Saltzer and Schroeder's 1975 list. Others apply it in general to a collection of unidentified but fundamental concepts. This paper focuses on explicit statements of principles, like the 1975 list. The principles were concise and well stated on the whole. Many have stood the test of time and are reflected in modern security practice. Others are not.

In 2008, after teaching a few semesters of introductory information security, I started writing my own textbook for the course. The book was designed to cover all topics required by selected government and community curriculum standards.

Informed by an awareness of Saltzer and Schroeder's design principles, but motivated primarily by the

curriculum requirements, the textbook, titled *Elementary Information Security*, produced its own list of basic principles (Smith, 2012). This review of design principles arises from the mismatch between the classic list and this more recent list. The review also looks at other efforts to codify general principles, both by standards bodies and by other textbook authors, including a recent textbook co-authored by Saltzer himself (Saltzer and Kaashoek, 2009).

The Saltzer and Schroeder List

Saltzer and Schroeder's 1976 paper listed eight design principles for computer security, and noted two additional principles that seemed relevant if more general.

1. Economy of mechanism – A simple design is easier to test and validate.

2. Fail-safe defaults – Figure 2 shows a physical example: outsiders can't enter a store via an emergency exit, and insiders may only use it in emergencies. In computing systems, the save default is generally “no access” so that the system must specifically grant access to resources. Most file access permissions work this way, though Windows also provides a “deny” right. Windows access control list (ACL) settings may be inherited, and the “deny” right gives the user an easy way to revoke a right granted through inheritance. However, this also illustrates why “default deny” is easier to understand and implement, since it's harder to interpret a mixture of “permit” and “deny” rights.

3. Complete mediation – Access rights are completely validated every time an access occurs. Systems should rely as little as possible on access decisions retrieved from a cache. Again, file permissions tend to reflect this model: the operating system checks the user requesting access against the file's ACL. The technique is less evident when applied to email, which must pass through separately applied packet filters, virus filters, and spam detectors.

4. Open design – Baran (1964) argued persuasively in an unclassified RAND report that secure systems, including cryptographic systems, should have unclassified designs. This reflects recommendations by Kerckhoffs (1883) as well as Shannon's maxim: “The enemy knows the system” (Shannon, 1948). Even the NSA, which resisted open crypto designs for decades, now uses the Advanced Encryption Standard to encrypt classified information.

5. Separation of privilege – A protection mechanism is more flexible if it requires two separate keys to unlock it, allowing for two-person control and similar techniques to prevent unilateral action by a subverted individual. The classic examples include dual keys for safety deposit boxes and the two-person control applied to nuclear weapons and Top Secret crypto materials. Figure 3 (courtesy of the Titan Missile Museum) shows how two separate padlocks were used to secure the launch codes for a Titan nuclear missile.Separation of privilege – A protection mechanism is more flexible if it requires two separate keys to unlock it, allowing for two-person control and similar techniques to prevent unilateral action by a subverted individual. The classic examples include dual keys for safety deposit boxes and the two-person control applied to nuclear weapons and Top Secret crypto materials.

6. Least privilege – Every program and user should operate while invoking as few privileges as possible. This is the rationale behind Unix “sudo” and Windows User Account Control, both of which allow a user to apply administrative rights temporarily to perform a privileged task.

7. Least common mechanism – Users should not share system mechanisms except when absolutely necessary, because shared mechanisms may provide unintended communication paths or means of interference.

8. Psychological acceptability – This principle essentially requires the policy interface to reflect the user's mental model of protection, and notes that users won't specify protections correctly if the specification style doesn't make sense to them.

There were also two principles that Saltzer and Schroeder noted as being familiar in physical security but applying "imperfectly" to computer systems:

1. Work factor – Stronger security measures pose more work for the attacker. The authors acknowledged that such a measure could estimate trial-and-error attacks on randomly chosen passwords. However, they questioned its relevance since there often existed "indirect strategies" to penetrate a computer by exploiting flaws. "Tiger teams" in the early 1970s had systematically found flaws in software systems that allowed successful penetration, and there was not yet enough experience to apply work factor estimates effectively.

2. Compromise recording – The system should keep records of attacks even if the attacks aren't necessarily blocked. The authors were skeptical about this, since the system ought to be able to prevent penetrations in the first place. If the system couldn't prevent a penetration or other attack, then it was possible that the compromise recording itself may be modified or destroyed.

Today, of course, most analysts and developers embrace these final two design principles. The argument underlying complex password selection reflect a work factor calculation, as do the recommendations on choosing cryptographic keys. Compromise recording has become an essential feature of every secure system in the form of event logging and auditing.

Security Principles Today

Today, security principles arise in several contexts. Numerous bloggers and other on-line information sources produce lists of principles. Many are variants of Saltzer and Schroeder, including the list provided in the Open Web Application Security Project's wiki (OWASP, 2012). Principles also arise in information security textbooks, more often in the abstract sense than in the concrete. Following recommendations in the report *Computers at Risk* (NRC, 1991), several standards organizations also took up the challenge of identifying a standard set of security principles.

Most textbook authors avoid making lists of principles. This is clear from a review of twelve textbooks published over the past ten years. This is even true of textbooks that include the word "Principles" in the title. Almost every textbook recognizes *the principle of least privilege* and usually labels it with that phrase. Other design principles, like separation of privilege, may be described with a different adjective. For example, some sources characterize separation of privilege as a control, not a principle.

Pfleeger and Pfleeger (2003) presents its own set of four security principles. They are, briefly, easiest penetration, weakest link, adequate protection, and effectiveness. These principles apply to a broader level of security thinking than Saltzer and Schroeder design principles. However, the text also reviews

Saltzer and Schroeder's principles in detail in Section 5.4.

The remaining few textbooks that specifically discuss design principles generally focus on the 1975 list. The textbook by Smith and Marchesini (2008) discuss the design principles in Chapter 3. The two textbooks by Bishop (2003, 2005) also review the design principles in Chapters 13 and 12, respectively.

"Generally Accepted Principles"

Following *Computers at Risk*, standards organizations were motivated to publish lists of principles. The OECD published a list of eight guidelines in 1992 that established the tone for a set of higher-level security principles:

Accountability, Awareness, Ethics, Multidisciplinary, Proportionality,
Integration, Timeliness, Reassessment, and Democracy.

In its 1995 handbook, "An Introduction to Computer Security," NIST presented the OECD list and also introduced a list of "elements" of computer security (NIST, 1995). Following the OECD's lead, this list presented very high level guidance, addressing the management level instead of the design or technical level. For example, the second and third elements are stated as follows:

"Computer Security is an Integral Element of Sound Management"

"Computer Security Should Be Cost-Effective"

The following year, NIST published its own list of "Generally Accepted Principles and Practices for Securing Information Technology Systems" (Swanson and Guttman, 1996). The overriding principles drew heavily from the elements listed in the 1995 document. The second and third elements listed above also appeared as the second and third "Generally Accepted Principles."

The OECD list also prompted the creation of an international organization that published "Generally Accepted System Security Principles" (GASSP) in various revisions between 1996 and 1999 (I2SF, 1999). This was intended to provide high-level guidance for developing more specific lists of principles, similar to those used in the accounting industry. The effort failed to prosper.

Following the 1999 publication, the sponsoring organization apparently ran out of funding. In 2003, the Information System Security Association tried to restart the GASSP process and published the "Generally Accepted Information Security Principles" (ISSA, 2004), a cosmetic revision of the 1999 document. This effort also failed to prosper.

In 2001, a team at NIST tried to produce a more specific and technical list of security principles. This became "Engineering Principles for Information Technology Security" (Stoneburner, et al, 2004). The team developed a set of thirty-three separate principles. While several clearly reflect Saltzer and Schroeder, many are design rules that have arisen from subsequent developments, notably in networking. For example:

- Principle 16: Implement layered security (Ensure no single point of vulnerability).
- Principle 20: Isolate public access systems from mission critical resources.
- Principle 30: Implement security through a combination of measures distributed physically and

logically.

- Principle 33: Use unique identities to ensure accountability.

While these new principles captured newer issues and concerns than the 1975 list, they also captured assumptions regarding system development and operation. For example, Principle 20 assumes that the public will never have access to "mission critical resources." However, many companies rely heavily on Internet sales for revenue. They must clearly ignore this principle in order to conduct those sales.

Training and Curriculum Standards

When we examine curriculum standards, notably those used by the US government to certify academic programs in information security, we find more ambiguity. All six of the curriculum standards refer to principles in an abstract sense. None actually provide a specific list of principles, although a few refer to the now-abandoned GASSP. A few of Schroeder and Saltzer's design principles appear piecemeal as concepts and mechanisms, notably least privilege, separation of privilege (called "segregation of duties" in NSTISSC, 1994), and compromise recording (auditing).

The Information Assurance and Security IT 2008 curriculum recommendations (ACM and IEEE, 2008) identify design principles as an important topic, and provide a single example: "defense in depth." This is a restatement of NIST's Principle 16.

Saltzer and Kaashoek

Co-authors Saltzer and Kaashoek published the textbook *Principles of Computer Design* in 2009 (Saltzer and Kaashoek, 2009). The book lists sixteen general design principles and several specific principles, including six security-specific principles. Here is a list of principles that were essentially inherited from the 1975 paper:

- General principle: Open design
- Security principle: Complete mediation
- Security principle: Fail-safe defaults
- Security principle: Least privilege
- Security principle: Economy of mechanism
- Security principle: Minimize common mechanism

Here are new – or newly stated – principles compared to those described in 1975:

- Security principle: Minimize secrets – a thoughtful addition to the list that could be prone to misunderstanding. Secrets should be few and changeable, but they should also maximize entropy, and thus increase the attacker's work factor. The simple principle is also true by itself, since each secret increases a system's administrative burden: a late 1990s fighter jet project required dozens of separately-managed crypto keys to comply with data separation requirements that had been added piecemeal.
- General principle: Adopt sweeping simplifications – a restatement that acknowledges how hopelessly complex modern systems have become. In the 1970s, a Unix operating system could support a dozen separate users with a megabyte of RAM; a single user on a modern desktop easily

consumes a gigabyte of RAM, much of it containing software programs.
- General principle: Principle of least astonishment – a concise and much clearer restatement of the "psychological acceptability" principle described in 1975.
- General principle: Design for iteration – an important first step towards incorporating continuous improvement as a design principle.

Neither of the uncertain principles listed in 1975 made it into this revised list. Despite this, event logging and auditing is a fundamental element of modern computer security practice. Likewise, work factor calculations continue to play a role in the design of information security systems. Pfleeger and Pfleeger highlighted "weakest link" and "easiest penetration" principles that reflect the work factor concept. However, there are subtle trade-offs in work factor calculations that may makes it a poor candidate for stating as a concise and easy-to-apply principle.

Elementary Information Security

The textbook *Elementary Information Security* presents a set of eight basic information security principles, While many directly reflect principles from Saltzer and Schroeder, they also reflect more recent terminology and concepts. The notion of "basic principles" stated as brief phrases seems like a natural choice for introducing students to a new field of study.

The textbook's contents were primarily influenced by two curriculum standards. The first was the "National Training Standard for Information System Security Professionals," (NSTISSC, 1994). While this document clearly showed its age, it remains the ruling standard for general security training under the US government's Information Assurance Courseware Evaluation (IACE) Program (NSA, 2012). In February, 2012, the IACE program certified the textbook as covering all topics required by the 1994 training standard. The second curriculum standard is the "Information Technology 2008 Curriculum Guidelines" (ACM and IEEE Computer Society, 2008). The textbook covers all topics and core learning outcomes recommended in the Information Assurance and Security section of the Guidelines.

To fulfill their instructional role, each principle needed to meet certain requirements. Each needed to form a memorable phrase related to its meaning, with preference given to existing, familiar phrases. Each had to reflect the current state of the practice, and not simply a "nice to have" property. Each had to be important enough to appear repeatedly as new materials were covered. Each principle was introduced when it played a significant role in a new topic, and no sooner. Students were not required to learn and remember a set of principles that they didn't yet understand or need.

This yielded the following eight principles:

1. Continuous Improvement - continuously assess how well we achieve our objectives and make changes to improve our results. Modern standards for information security management systems, like ISO 27001, are based on continuous improvement cycles. Such a process also implicitly incorporates compromise recording from 1975 and "design for iteration" from 2009. Introduced in Chapter 1, along with a basic six-step security process to use for textbook examples and exercises.

2. Least Privilege - provide people or other entities with the minimum number of privileges necessary to allow them to perform their role in the system. This literally repeats one of the 1975 principles. Introduced in Chapter 1.

3. Defense in Depth - build a system with independent layers of security so that an attacker must defeat multiple independent security measures for the attack to succeed. This echoes “least common mechanism” but seeks to address a separate problem. Defense in depth is also a well-known alternative for stating NIST's Principle 16. Introduced in Chapter 1.

4. Open Design - building a security mechanism whose design does not need to be secret. This also repeats a 1975 principle. Introduced in Chapter 2.

5. Chain of Control - ensure that either trustworthy software is being executed, or that the software's behavior is restricted to enforce the intended security policy. This is an analogy to the “chain of custody” concept in which evidence must always be held by a trustworthy party or be physically secured. A malware infection succeeds if it can redirect the CPU to execute its code with enough privileges to embed itself in the computer and spread. Introduced in Chapter 2.

6. Deny by Default – grant no accesses except those specifically established in security rules. This is a more-specific variant of Saltzer and Schroeder's “fail safe defaults” that focuses on access control. The original statement is less specific, so it applies in safety and control problems. Introduced in Chapter 3.

7. Transitive Trust - If A trusts B, and B trusts C, then A also trusts C. In a sense this is an inverted statement of “least common mechanism,” but it states the problem in a simpler way for introductory students. Moreover, this is already a widely-used term in computer security. Introduced in Chapter 4.

8. Separation of Duty – decompose a critical task into separate elements performed by separate individuals or entities. This reflects the most common phrasing in the security community. Some writers phrase it as “segregation of duty” or “separation of privilege.” Introduced in Chapter 8.

The textbook's list focused on memorable phrases that were widely accepted in the computer security community. Principles introduced in earlier chapters always resurface in examples in later chapters. In retrospect, the list is missing at least one pithy and well-known maxim: “Trust, but verify.” The book discusses the maxim in Chapter 13, but does not tag it as a basic principle.

Omitted Principles

For better or worse, three of the 1975 principles do not play a central role in modern information security practice. These are simplicity, complete mediation, and psychological acceptability. We examine each below.

There is no real market for simplicity in modern computing. Private companies release product improvements to entice new buyers. The sales bring in revenues to keep the company operating. The company remains financially successful as long as the cycle continues. Each improvement, however, increases the underlying system's complexity. Much of the free software community is caught in a similar cycle of continuous enhancement and release. Saltzer and Kaashoek (2009) call for “sweeping simplifications” instead of overall simplicity, reflecting this change.

Complete mediation likewise reflects a sensible but obsolete view of security decision making. Network access control is spread across several platforms, no one of which makes the whole decision. A packet

filter may grant or deny access to packets, but it can't detect a virus-infected email at the packet level. Instead it forwards email to a series of servers that apply virus and spam checks before releasing the email to the destination mailbox. Even then, the end user might apply a digital signature check to perform a final verification of the email's contents.

Psychological acceptability, or the “principle of least astonishment” is an excellent goal, but it is honored more in the breach than in the observance. The current generation of “graphical” file access control interfaces provide no more than rudimentary control over low-level access flags. It takes a sophisticated understanding of the permissions already in place to understand how a change in access settings might really affect a particular user's access.

Conclusion

Version:1.0 StartHTML:0000000167 EndHTML:0000001597 StartFragment:0000000502 EndFragment:0000001581

Only a handful of Saltzer and Schroeder's original 1975 design principles have stood the test of time. Nonetheless, this represents a memorable success. Kerckhoffs, a 19th century French cryptographic expert, published a list of principles for hand-operated cipher systems, some of which we still apply to cryptosystems today. But most experts only recognize a single principle as “Kerckhoffs's Principle,” and that is his view on Open Systems: a cryptosystem should not rely on secrecy, since it may be stolen by the enemy. In addition to the Open System principle, both the principle of least privilege and of separation of privilege appeared on the 1975 list and are still widely recognized by security experts.

Perhaps lists of principles belong primarily in the classroom and not in the workplace. The short phrases are easy to remember, but they may promote a simplistic view of technical problems. Students need simplicity to help them build an understanding of a more complex reality.

References

ACM and IEEE Computer Society, 2008, *Information Technology 2008 Curriculum Guideline*, http://www.acm.org/education/curricula/IT2008%20Curriculum.pdf, (retrieved March 1, 2012).

Bishop, 2003. *Computer Security: Art and Science*, Boston: Addison-Wesley.

Bishop, 2005. *Introduction to Computer Security*, Boston: Addison-Wesley.

I2SF, 1999. “Generally Accepted System Security Principles” International Information Security Foundation.

ISSA, 2004. “Generally Accepted Information Security Principles,” Information System Security Association.

Kerckhoffs, Auguste, 1883. “La cryptographie, militaire,” *Journal des sciences militaires* IX.

NCSC, 1985. *Trusted Computer System Evaluation Criteria*, Ft. Meade, MD: National Computer Security Center.

NIST, 1995, "An Introduction to Computer Security," NIST SP 800-12, Gaithersburg, MD: National Institute of Standards and Technology.

NSA, 2012. "IA Courseware Evaluation Program – NSA/CSS," web page, National Security Agency. http://www.nsa.gov/ia/academic_outreach/iace_program/index.shtml (retrieved Feb 29, 2012).

NRC, 1991. *Computers at Risk: Safe Computing in the Information Age*, Washington: National Academy Press. http://www.nap.edu/openbook.php?record_id=1581 (retrieved Feb 29, 2012).

NSTISSC, 1994. "National training standard for information security (INFOSEC) professionals," NSTISSI 4011, Ft. Meade, MD: National Security Telecommunications and Information Systems Security Committee.

OWASP, 2012, "Category: Principle - OWASP," web page, Open Web Application Security Project, https://www.owasp.org/index.php/Category:Principle (retrieved Feb 29, 2012).

Pfleeger, Charles, 1997. *Security in Computing* 2nd ed., Wiley.

Pfleeger, Charles, and Shari Pfleeger, 2003. *Security in Computing* 3rd ed. ,Wiley.

Saltzer, Jerome, 1974. "Protection and the control of information sharing in Multics," *CACM* 17(7), July, 1974.

Saltzer, Jerome, and Kaashoek, 2009. *Principles of Computer Design,* Wiley.

Saltzer, Jerome, and Schroeder, 1975. "The protection of information in computer systems," *Proc IEEE* 63(9), September, 1975.

Shannon, 1949. "Communication Theory of Secrecy Systems," *Bell System Technical Journal 28(4).*

Smith, Sean, and Marchesini, 2008, *The Craft of System Security*,

Smith, Richard, 2012. *Elementary Information Security*, Burlington, MA: Jones and Bartlett.

Stoneburner, Gary, Clark Hayden, and Alexis Feringa, 2004. "Engineering Principles for Information Technology Security," SP 800-27 A, Gaithersburg, MD: National Institute of Standards and Technology.

Swanson, Marianne, and Barbara Guttman, 1996. "Generally Accepted Principles and Practices for Securing Information Technology Systems," SP 800-14, Gaithersburg, MD: National Institute of Standards and Technology.

Textbooks Reviewed but not Cited

Forouzan, 2008. *Cryptography and Network Security*, McGraw-Hill.

Gollmann, 2006. *Computer Security* 2nd ed., Wiley.

Newman, 2010. *Computer Security: Protecting Web Resources*, Jones and Bartlett.

Stallings, 2003, *Network Security Essentials*, Prentice-Hall.

Stallings, 2006. *Cryptography and Network Security*, Prentice-Hall.

Stallings and Brown, 2008. *Computer Security: Principles and Practice*, Prentice-Hall.

Stamp, 2006. *Computer Security: Principles and Practice*, Wiley.

Whitman and Mattord, 2005. *Principles of Information Security* 2nd ed., Thomson.