Chapter 5
Radically Simplifying Cyber Security

Dan Kruger and John N. Carbone

Introduction

Cyber security professionals are publicly acknowledging [1, 2], that the traditional approaches to securing information fail because the threat environment has become impossibly complex. This chapter on a fundamentally new approach to securing information that can radically simplify cyber security.

The definition of the problem is the problem. Cyber security has become unmanageably complex because the definitions of security do not match the operational environment—and they haven't for a long time. Historical approaches to securing information were more applicable in the mainframe era where systems were more self contained, independent, and didn't touch many external systems as they do in the web era today. Hence, the following security definitions were in use then are still in force today:

Perimeters: Information is protected when you control access to it by establishing a perimeter. Defining the perimeter was simple—control physical access to the building and the terminals. As an example, years ago an information systems director of a major oil company became nearly apoplectic when personal computers were introduced to his company. He said, "Once you let them in, we'll never have control of our data again." He was right.
Processing Capacity: For information to be readily useful it needs to be unprotected—stored "in the clear." Protecting information takes significant processor capacity and makes it more time consuming and difficult to process.
Training: Keep users from leaking information by training them to understand and follow the organization's security policies.

D. Kruger (✉)
Absio Corporation, 8321 S. Sangre De Cristo Rd, Ste 302, Littleton CO 80127-6426, USA
e-mail: dan.kruger@absio.com

J. N. Carbone
2730 Woods Lane, Garland TX 75044, USA
e-mail: jcarbone@raytheon.com

S. C. Suh et al. (eds.), *Applied Cyber-Physical Systems*,
DOI: 10.1007/978-1-4614-7336-7_5,

Therefore, this paper addresses the challenge of fundamentally enhancing and changing existing security paradigms by minimizing system ambiguities and significantly simplifying the cyber security of information content.

Perimeter Ambiguity

Historical assumptions regarding perimeters are no longer valid today; starting with the notion that one can simply define a perimeter around the complex system interconnects that currently exist. It's worthwhile to look at just how hard it is to merely define the perimeter—much less defend it. Begin with a mainframe and terminals in a building you control. Then add PCs, each with their own processors, local storage, I/O ports and local applications; each PC has a perimeter of its own. Then add portable computers designed to go outside the physical perimeter. Subsequently adding a local area network, adds physical perimeter elements you must now defend which can be prone to tapping and possible eavesdropping. Lastly, adding wide area networks now expands your perimeter elements to include somebody else's wires and switches.

These definitions describe distributed computing in the mid 1980s—the last time anybody was able to do a marginally credible job of defining a perimeter—and this is where the notions of cyber security have been mired in seemingly outdated principles for today's complex system environments. It is well known in the cyber security field to discuss protecting networks as if that were synonymous with protecting information. Network protection, even if perfect, is a partial security solution at best.

Next we make the move to the mid 1990s and add the commercialized Internet. Then add ubiquitous email and the World Wide Web, both enabling the unconstrained distribution of information by unauthenticated users. Next add high-density portable drives and make some of them smaller than a postage stamp. Then add radios in multiple flavors—cellular, Wi-Fi and Bluetooth, with more on the way. Include applications, too—peer-to-peer networking, streaming media and social networking. Can any organization actually define its perimeter? If it were possible, would you be able to defend it? Let's look at the complexity of defending a small workgroup.

Every digital device, operating system (OS), application, transmission path, file and human being has multiple attack vectors (paths to the target) and attack surfaces (that which can be attacked). When devices are booted, users log in, launch applications, open emails, browse websites, manipulate files or transmit information, those actions open vectors and expose surfaces, creating attack opportunities.

Let's do some rough math: 10 users $\times$ 3 devices each $\times$ 20 applications each $\times$ 5 attack vectors $\times$ 100 interactions daily. That's 3,000 perimeter elements to defend and 300,000 threat opportunities a day (Argue the math if you want—it's actually worse than the example.). If that perimeter could be precisely defined, who

would have the time, money and expertise to close all of the holes and keep them closed?

In addition to sheer numbers, threats are unpredictable and dynamically complex. Addressing one can produce unintended consequences in others, and nifty new technologies compound the problem. It's only a matter of time before we hear something like this: "A major breach was traced back to an exploit that used the smart refrigerator interface in the CIO's home network to infect his tablet, which then invaded his company's network."

Even if you could define all of the perimeters, they are too complex to defend. This is not a call to abandon the perimeter; the better the perimeter defense, the more sophisticated the attacker needs to be. But it is clear that the current definition of the perimeter is only the outer layer of defense [3–5]. Therefore, defense in depth is required.

Processing Capacity

If information protection is properly engineered, there is more than enough processor capacity to protect information everywhere it's at rest or in motion—if you include the processor on edge devices. It's critical to include edge device capacity in any approach to cyber security because increasingly sophisticated edge devices:

- Are where most information lives
- Have I/O ports for exporting information
- Are often portable and easily stolen or captured
- Are what users are using and will continue to use

That brings us to the cloud. The drive toward clouds and the drive for more powerful mobile devices are, in many ways, at odds with each other. Cloud computing assumes that processing is done in the cloud and that edge devices host what are essentially visually pleasing dumb terminal emulations that always have a good connection. We have recreated the mainframe model in an attempt to define the perimeter. The conceptual security advantages of clouds are obvious: They have a definable perimeter. But the perimeter itself raises a fundamental security problem [6].

If users can take the product of cloud computing and freely distribute it, then cloud security stops at the perimeter of the cloud (more precisely, at the datacenter's interface to the Internet). Information security solutions must comprehend information security from cloud to cloud, cloud to edge device, and edge device to edge device. Information security must persist.

Disable-your-hardware solutions for securing cloud connections have not and are not going to be fully implemented. People are not going to disable their devices' I/O ports. Users will store and share information on their edge devices regardless of security policies.

People and Training

The third assumption was always more hope than reality. Do we expect users to follow security policies that make their jobs more difficult than they already are—especially when they know the policies make little difference? Do we expect that good training will stop all social engineering and/or thwart suborned or malicious insiders? Stop outsiders who masquerade as insiders? Do we expect that sufficient training will ensure that every user in the distribution chain will make the correct security decisions about a piece of information [7]? If the perimeter has been breached and the network has been invaded. We cannot stop it. What can we do? We must, "Change the game".

Methods

Enhanced Security Via Separation of Knowledge and Context

Knowledge and contextual understanding of it has been debated for years. Brillouin [8] defined knowledge succinctly as resulting from a certain amount of thinking and distinct from information which had no value, was the "result of choice", and was the raw material consisting of a mere collection of data. Additionally, Brillouin concluded that a hundred random sentences from a newspaper, or a line of Shakespeare, or even a theorem of Einstein have exactly the same information value. Therefore, information content has "no value" until it has been thought about and thus turned into knowledge. Following this train of thought, knowledge is created through the amount of context, which can be recombinantly assimilated over time until a threshold of relationship understanding is achieved [9]. Gruber [10] states that collective intelligence emerges if data collected from all people is aggregated and "recombined" to create new knowledge. To form an understanding of the relationship between different knowledge and contexts when assimilating knowledge, the associated relationships can be written symbolically as knowledge Ki and the associated context relationship Rj where, Ki(Rj) represents a recombination of knowledge and context as shown in Equation 1 below.

For preventing cyber attackers this is a key understanding since the amount of context received is a function of how much access can be achieved. The amount of access obviously increases the context and the possibility of damage. The more an attacker knows the more he can do. Thus, to remain secure we need to separate the knowledge and information content and hence negate an attackers ability to gain context. Hence, the next sections discuss the simplified mechanisms to enhance cyber security by obfuscating the knowledge and context itself.

$$\sum_{i,j} K_i(\ R_j)$$

Equation 1, Recombinant Knowledge Assimilation

Information that Protects Itself

Cyber security is radically simplified if we move primary information security into the information itself. With today's processors, storage density and the right engineering, it is possible to make every piece of information a "hard target" that protects itself and is still easy to use. The focus of cyber security can shift from the utterly impossible (defending the undefendable perimeter and persuading the unpersuadable user) to the merely difficult (moving information out of the clear). Hence, if information can protect itself, the assumptions behind cyber security are very different (Table 5.1).

Establishing Persistent Distribution Control at the Object Level

The mechanism to obfuscate knowledge by securing contextual information content is performed by the creation of an individually protected object called a SEGCO—a Secure Extensible Global Content Object. As shown in Fig. 5.1, the application architecture supporting the creation, storage, transportation and authentication of SEGCOs, along with the formalized application programming interface (API), thus making it available to developers, is called the Persistent Distribution Control System (PDCS). PDCS enables developers to build persistent distribution control into their applications, which enables the encapsulation of information within SEGCOs as any new information is created, enables storage and transmission of only SEGCOs and hence, creates a secure environment where information at rest and in motion is dynamically secured. If an attacker actually

Table 5.1 Changing cyber security assumptions

Current assumptions	New assumptions
Information is protected when you control access to it by establishing a perimeter	Information is protected when it is bound to persistent distribution controls and is never in the clear except when actually being used
Information needs to be stored in the clear to be easy to use	Information does not need to be stored in the clear to be easy to use
You keep users from leaking information by establishing rules and hoping they follow them	You keep users from leaking information by establishing rules that their applications enforce

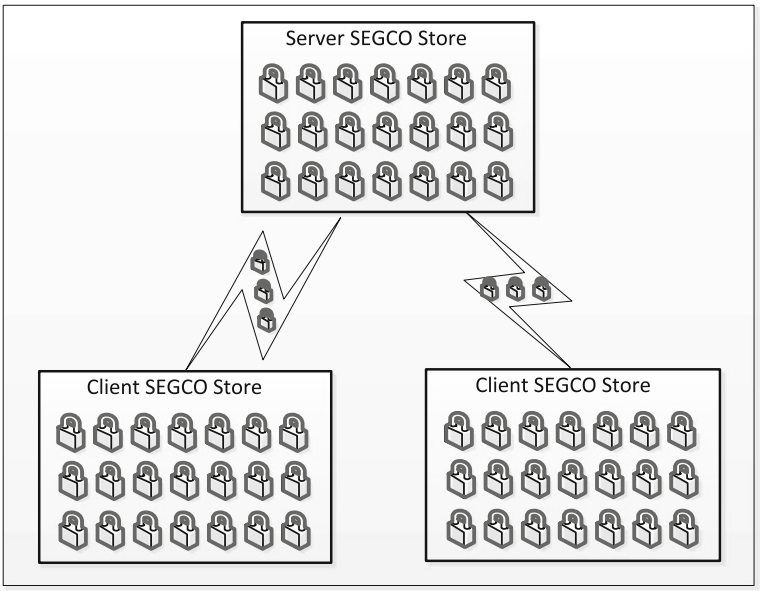

Fig. 5.1 Secure extensible global content objects

succeeds in breaching a client device, server or signal, the only content they achieve access to is a mountain of individually secured SEGCOs.

Key SEGCO/PDCS attributes include the following:

- SEGCOs are individually encrypted, each with a distinct key. That radically reduces the number of useful attack vectors and offers the most resistant of all attack surfaces. Attacks designed to copy and illicitly export data fail to deliver information attackers can use.
- PDCS stores SEGCOs uniformly. SEGCOs are indistinguishable from each other. This makes getting the right information object analogous to finding a particular grain of sand on a beach of identical grains of sand. The more grains of sand, the harder the problem for attackers.
- SEGCOs are transmitted in an encrypted tunnel. A breached tunnel yields only SEGCO's.
- PDCS and SEGCOs make it possible to build tools and applications that work across almost any kind of hardware and OS. Information that is secured in a SEGCO from the moment it is created will be safe in motion and at rest wherever it exists.
- SEGCOs and PDCS make it possible to build mechanisms to authenticate users, devices and applications prior to decrypting the information (chain-of-trust fingerprinting).

Therefore, the consistency of SEGCO metadata and the audit function of PDCS make it possible to implement auditing systems that monitor the movement and

use of information in near real time. Those systems may be able to flag the misuse of information objects fast enough to keep the attack from succeeding and the comprehensiveness and immutability of the audit trail can establish a legally admissible chain of custody to aid in prosecution.

Finally, SEGCOs and PDCS can therefore provide an array of capabilities that application developers could use to create new solutions, including:

- Fine-grained control of secondary distribution such as forward, export, copy/paste, or print.
- Secure documents that redact themselves as they move through distribution.
- Cross-domain secure collaboration without inter-domain access.
- Copyright protection that does not unduly restrict the user's access to content.
- Commercial authentication without risk of exposing personally identifiable information.

SEGCO-PDCS Requirements and Architecture

Fundamentally, a SEGCO must: be platform and content-agnostic (support any data type) as shown in Fig. 5.2, be distinctly encrypted (provide a different key), remain encrypted in motion and at rest, contain distribution control information inside the encrypted envelope, contain audit information inside the encrypted envelope, not indicate the type of content it contains and be randomly and/or nonsensically named.

Subsequently, PDCS must be implemented using client–server architecture to disperse the encryption/decryption processor load across millions of edge processors and to provide a common multiplatform security API as shown in Fig. 5.3. This point is critical since developers rarely have the knowledge required to build information security into their applications [11, 12], and it would likely do more

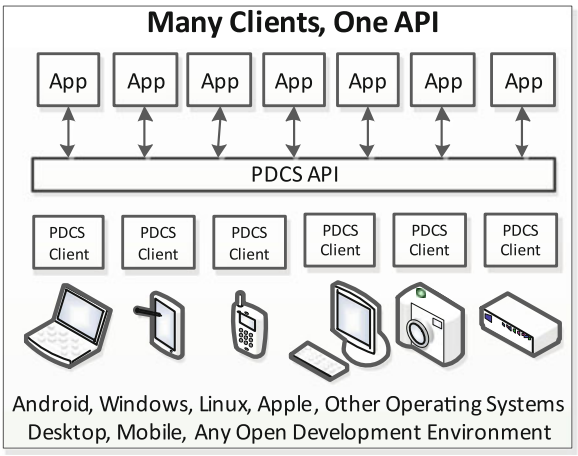

Fig. 5.2 SEGCO-PDCS architecture

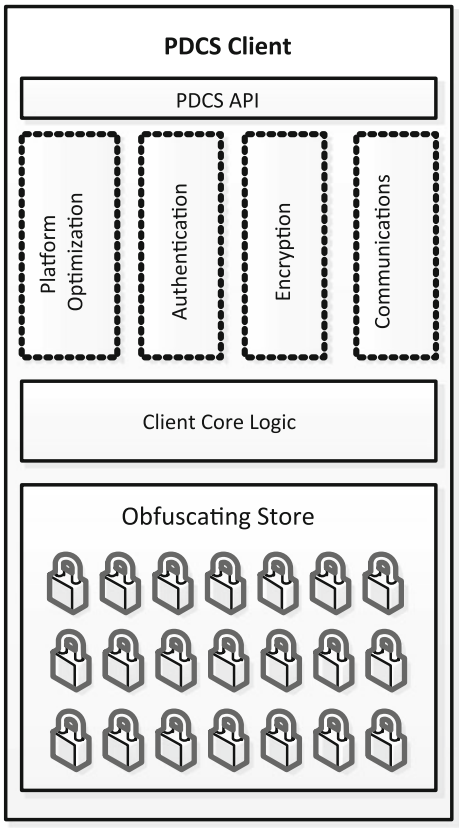

Fig. 5.3 PDCS architecture

harm than good if their security solutions were all different. Methods and developer tools for securing information objects must be easy to use, standardized, broadly applicable and inexpensive. They should also support disconnected use, enable secure storage on edge devices and servers, and enable hierarchical server architecture in order to place servers into the resource-loaded environment where they are needed. From a communications or data-in-motion perspective the architecture should enable secure transport and interim storage for applications that require large amounts of data in the clear (e.g. cloud to cloud, cloud to edge) and enable intelligent prioritization of bandwidth not just for communications optimization but also for cyber security. Empty space always leaves room for infiltration.

PDCS clients must be OS-optimized, authentication method agnostic, encryption method agnostic and communications method agnostic (support any Internet protocol transmission path) so that clients can run efficiently on any platform. The PDCS client architecture provides:

- An API with common calls across platforms.
- Platform-specific performance optimization.
- Platform-specific module for authentication.

- Platform-specific module for encryption.
- Platform-specific module for communications.
- Common core logic across platforms.
- A SEGCO store that makes SEGCOs uniform.

PDCS content servers should be as naïve and lightweight as possible—and they can be since client devices are performing application processing and "heavy lifting" such as encrypt/decrypt. The key functions of PDCS servers will be to route, store and forward SEGCO's, while limiting access to authenticated users and devices (no support for system level anonymity) and to provide a mesh network set of management functions because of their distributed nature and finally, to provide the necessary resilience and redundancy components necessary for seamless disaster recovery functionality.

Results

Outcomes of SEGCOs and Persistent Distribution Control

What current hard problems would be simplified if applications stored and moved information in SEGCOs managed through a PDCS application architecture? These are some we have identified:

- Insider threat: If Bradley copies 200,000 SEGCOs to a Lady Gaga CD and gives them to Julian, Julian will have a useless CD—and nothing to publish.
- Malware-based storage attacks: If malware is able to convey only SEGCOs to outsiders, the malware is of no value to the malware writer or purveyor.
- Signal intercepts (man-in-the-middle attack): If a signal is intercepted and the signal contains only a stream of SEGCOs, the intercept is of no value to the eavesdropper.
- Device capture: If a device containing thousands or more SEGCOs is captured, the device has no value to the thief.
- Personal identity protection: Applications can be built that enable the complete separation of personally identifiable information from the information objects that represent the person.
- Cross-domain information sharing without cross-domain access: Applications can be built that support software-based secure intermediate logical networks.
- Copyright preservation without undue restrictions on usability: Applications can be built that ensure copyright holders can track and get paid for their content while allowing buyers to access their content on whatever device they are using at the moment.
- Intelligent traffic prioritization: Applications can be built that support client-based prioritization of traffic—a critical need anywhere bandwidth is scarce and urgency is high.

Remaining Vulnerabilities and Next Steps

It should be noted that persistent distribution control does not eliminate information security problems. However, PDCS shrinks the problem considerably by undermining the value of storage and eavesdropping exploits, reducing the requirement for users to follow security policies and increasing the risk, time, and effort of attempting exploits.

Hence, full information security requires a comprehensive combination of tactics for closing the holes in our security of our hardware and software supply chains and "at the bottom of the stack" [13]. A number of solutions are needed, such as separation kernels, secure operating systems and associated access to them, encrypted and dynamically allocated memory, effective micro virtual machines with discrete levels of focused capability, robust chain-of-trust solutions based upon pedigree creation, modification and deletion. Last but not least a comprehensive roadmap is needed for achieving a full spectrum of capabilities security nexus and can be summarized in these suggested next steps: reduce the attack surface, validate trust and verify your user base, continuously evaluate your enemy's methodologies, dynamically discover and vanquish the intruders, measure, automate, and audit comprehensively, vary your methods, processes, and locations to make yourself a moving target, and most importantly grow the vitality of your force through proper preparation, education and focused training.

Conclusions

This chapter addressed the challenge of fundamentally enhancing and changing existing security paradigms by minimizing system ambiguities and significantly simplifying the cyber security of information content. Thus, by shrinking the size and scope of the threat universe, which cyber professionals describe as reducing the attack surface, the PDCS described herein showed how cyber security professionals and the industry can focus on the small number of truly sophisticated attacks and attackers while simultaneously reducing the time needed to address nuisance attacks. The SEGCO mechanisms and PDCS architecture described showed how cyber attackers will be forced to have more expertise, expend more capital and take on more risk since attacking systems secured with these formalized methods will become much more expensive and have a much smaller chance of success. Therefore, information that can protect itself ruins the economics of hacking.

References

1. D. Barrett, “U.S. Outgunned in Hacker War”, Wall Street Journal, March 28, 2012.
2. ThreatPost, Kapersky Labs: Experts Tell Senate: Government Networks Owned, Resistance Is Futile, March 21, 2012
3. Axel Buecker, Per Andreas and Scott Paisley, “Understanding IT Perimeter Security,” IBM Corporation, 2008.
4. Marcia Savage, “Perimeter Defenses Deemed Ineffective Against Modern Security Threats,” Information Security, June 30, 2010.
5. Simson Garfinkel, “The Deperimeter Problem,” CSO Online, November 1, 2005.
6. Anh Nguyen, “Infosec: Cloud Computing ‘Explodes’ the Security Perimeter,” CSO Online, April 25, 2011.
7. “Top Cause of Data Breaches? Negligent Insiders,” Help Net Security, Ponemon Institute, March 22, 2012.
8. L. Brillouin, Science and information theory: Dover, 2004.
9. Crowder, J. A., Carbone, J. N., “The Great Migration: Information to Knowledge using Cognition-Based Frameworks.” Springer Science, New York (2011).
10. T. Gruber, “Collective knowledge systems: Where the social web meets the semantic web,” Web Semantics: Science, Services and Agents on the World Wide Web, vol. 6, pp. 4–13, 2008.
11. Adam Cummings and Ron Bendes “Information Security in Application Development Projects,” cmu95752, Carnegie Mellon University, March 7, 2012
12. “Risk Across the Phases of Application Security,” Help Net Security, Ponemon Institute, March 21, 2012.
13. A. Metke, “Security technology for smart grid networks” IEEE Transactions on Smart Grid, 2010.