aesec™
The power of verifiable protection™

TECHNICAL NOTE

Operating System Security
Advantages of GEMSOS™ Security Kernel
Over Other Kernels
March 29, 2017

Introduction

This note describes the advantages of the GEMSOS security kernel over other operating system security approaches available to industry today, including other so-called secure kernels, microkernels, and secure operating systems. In a nutshell, GEMSOS has three major advantages, as confirmed by the NSA's National Computer Security Center [1]:

1. Mitigation of subversion from malicious software;
2. Enforcement of mandatory access control (MAC) policy, which enables components to be composed into secure networked systems; and
3. Verification under *TCSEC* "Class A1: Verified Protection," the most rigorous highest-level computer security criteria that has ever been published and proved to work.

In a 2016 *CACM* Viewpoint column [2], Aesec's Roger Schell calls these factors the "Cyber Defense Triad". These three requirements of the Cyber Defense Triad are not unique to GEMSOS, but apply in general to all operating systems that seek to provide security in the real world. Protection against subversion from malicious software is intractable unless a system implements correct enforcement for every reference to information and enforces a MAC policy to protect itself. A set of components, each with an operating system and the applications running on it, can't feasibly be composed into a secure networked system unless at least one such component enforces a system-wide MAC policy. And, to be verifiable, the protections of such components and the network system must be engineered so that an objective third party can verify these protections under rigorous criteria. This note and its references describe how GEMSOS delivers these requirements and explains why all other secure operating system approaches today fail.

GEMSOS is a security kernel, which is the only approach yet that has succeeded in delivering the Cyber Defense Triad including mitigation of subversion and verifiability. As Aesec's Mark Heckman and Roger Schell put it in a 2016 paper in *Information* [3] (Section 2.1), "[t]he proven scientific principle of the 'Reference Monitor', whose implementation is called a 'security kernel', enables engineering a verifiably secure OS." Schell's *CACM* Viewpoint column illustrates this by noting [2] (p. 22, col. 2) that, "[A]t least a half dozen security kernel-based

operating systems have been produced that ran for years (even decades) in the face of nation-state adversaries without a single reported security patch."

Of these security kernels, only GEMSOS is commercially marketed today. And there is one major advantage GEMSOS provides, even over the rest of those security kernel-based operating systems: GEMSOS is designed to be a general-purpose security kernel. That means it is engineered to be reusable for a wide variety of applications – it is application agnostic, in the same sense that untrustworthy general purpose operating systems like Linux are. In contrast the other security kernels that have been implemented are designed for specific applications and those security kernels themselves are likely to require extensive modification for new applications. For this reason, Intel engineers have informally referred to the GEMSOS security kernel in the past as "smart hardware" or "security aware hardware."

To facilitate reuse, as described in Aesec's *GEMSOS Security Kernel RTOS* product overview [4] (p. 2, col. 2), "GEMSOS is available under a proven OEM business model." Heckman shows the effectiveness of this general-purpose approach when he reports [3] (Section 4.2), "[t]his is demonstrated by OEM deployments of highly secure systems and products, ranging from enterprise 'cloud technology' to general purpose data base management systems (DBMS) to secure authenticated Internet communications, by applying commercially available security kernel technology." In fact, most of the systems and products he reports were based on this same reusable GEMSOS security kernel interface.

Hardware Foundations for Secure Operating Systems

50 years ago (!), ARPA sponsored brilliant scientists in the Multics project to address the very difficult problem of how to make computers secure. They addressed the manifest vulnerability to malicious software subversion inherent in the emerging DOS and UNIX architectures still with us today. These experts, from General Electric, Honeywell, Bell Labs and MIT, together concluded that to be practical you needed hardware capabilities specifically designed to support security. They specifically identified that the underlying processor architecture needed to include "segmentation" [5] and "protection rings" [6]. You cannot provide this support efficiently with software. The science hasn't changed, and security hasn't gotten easier. In fact, all formally evaluated highly secure systems and products that have been fielded have had such hardware support. The bottom line is simple. It has little to do with whether your target operating system is GEMSOS or some other kernel. GEMSOS, and any other practical secure system with controlled sharing of information, will leverage hardware support for protection rings and segmentation.

NSA's published formal *TCSEC* Class A1 evaluation [1] of GEMSOS included the following description:

> The software and the hardware together provide the security support in the system. The 80286, i386, and i486 processors provide hardware support for segmentation of memory. These processors also provide an architecture with four hierarchical privilege levels that provide isolation of the kernel and support domain separation. The software takes full advantage of these hardware security and protection features.

The importance of this hardware support has long been recognized by those giving careful scholarly attention to the challenges of high assurance security. This is illustrated by the following excerpts from Bill Caelli's 2002 paper in *Colloquium for Information Systems Security Education* [7]:

> Current Intel-based computer architecture, at least from the iAPX-286 CPU onwards, owes its security structure in large part to the earlier MULTICS program. This developed from the 1960s to late 1970s to create a secure, time-shared computing environment.
>
> The 286 took the unusual step of incorporating the concepts of "protection rings" and memory segmentation much along the lines of the MULTICS effort of the previous 15 years and the 4 "levels" of protection in the DEC VAX design. This fact as clearly acknowledged by two Intel designers in their 1986 book "The 80286 Architecture" by Morse and Albert, in the following words:
>
> > "The 286 protection mechanism was inspired by protection in the MULTICS operating system." (Pg. 190).
>
> Use of this ring structure was suggested in the appropriate Intel reference manuals and the basic structure has not changed to the present, even under the Intel Pentium 4 processor group. It was placed in good use in the "GEMSOS", high-trust operating system developed by Roger Schell in the mid-1980s. It became clearly useful in the development of a high trust kernel structure upon which further trusted functionality could be based.

Aesec has some history with these major hardware security innovations. Roger Schell consulted with Intel's Bob Child on these features in the original iAPX-286 microprocessor and, fortunately, these features persist in the x.86 instruction set today. Intel provides the processors for the huge majority of real modern hardware, and in fact most of these include the IA32 instruction set. That is why Schell's *CACM* Viewpoint column [2] noted, "… many commodity processors (e.g., those that implement the Intel IA32 architecture) still include the hardware segmentation and protection rings essential to efficient security kernels." So, Intel has two unassailable advantages for a commercialization strategy and an adoption model for reusable trusted devices: (1) Intel has the only widely used CPU architecture in the world with the hardware support for "segmentation" and "protection rings" essential to an efficient secure OS, and (2) Intel domestically manufactures chips with that architecture. And GEMSOS is the only general-purpose operating system security solution in the world that fully leverages these advanced security features today.

Secure Operating Systems Currently Marketed

To avoid needless confusion and erroneous conclusions in comparisons with other products, it is important to use precise terminology, especially for the term "security kernel" mentioned above. As Schell's *CACM* Viewpoint column notes [2] (p. 20, col. 2), this concept was proposed more than 40 years ago as, "a compact security 'kernel' of the operating system and supporting

hardware – such that an antagonist could provide the remainder of the system without compromising the protection provided." More than a decade later, this was presented as a term of art in the *TCSEC* [8] glossary as:

> "Security Kernel" - The hardware, firmware, and software elements of a Trusted Computing Base that implement the reference monitor concept. *It must mediate all accesses, be protected from modification, and be verifiable as correct.* [Emphasis supplied.]

This is much more rigorous than the generic term "kernel," which has been used by computer science for operating systems for decades, e.g., the *Wikipedia* article [9] states the common engineering understanding of an OS kernel as, "The kernel is a computer program that is the core of a computer's operating system, with complete control over everything in the system."

This distinction between security kernel and other kinds of kernels is crucial. Although a security kernel is an operating system kernel, other kernels do not implement the reference monitor concept as defined in *TCSEC* and computer science generally, are not verifiable as correct, and therefore fall far short of the cyber security properties of a security kernel. Unfortunately, in more recent years some cyber security solutions currently marketed have misled the unwary by abusing the term "security kernel."

To elaborate on the advantages of the GEMSOS security kernel, what follows identifies some important definitions, properties and distinctions among the following four types of kernels:

- Security Kernels
- OS Kernels
- Separation Kernels or Partition Kernels
- Microkernels

Security Kernels

The security properties of security kernels have been widely discussed in the literature, including those properties described in Schell's *CACM* Viewpoint column and Heckman's *Information* paper. We will not attempt a repetition here. However, we will highlight a few characteristics particularly important to current and future cyber security. In particular, the *CACM* Viewpoint column emphasizes that for any system to be secure it must apply the Cyber Defense Triad mentioned in the Introduction above; that is: "… the scientific or engineering rigor needed for a trustworthy system to defend the security of networked computers in three dimensions at the same time: mandatory access control (MAC) policy, protection against subversion, and verifiability."

Computer security assessments [2] (p.20, col. 1) have long "recognized subversion as the most serious threat to security." This is even truer today [2] (p. 21, col. 2) where "a witted adversary has numerous opportunities to subvert or sabotage a computer's protection software itself to introduce insidious new flaws. This is an example of 'malware', a preferred attack for many of the most serious breaches." It cannot be overemphasized that - as a practical matter - to be

secure, a system must significantly mitigate subversion. And today's reliance on encryption does nothing to change that because encryption is just an application running on the operating system. Caelli [7] emphasizes that, "[c]ontrary to accepted ideas, then, the use of cryptography actually enhances the need to reconsider security functionality and evaluation at the operating system and hardware levels." Caelli [7] also gives a lively history of Intel hardware support for high assurance security kernels, and why Microsoft failed to follow their lead on this.

The needed rigor is defined by the *TCSEC*, and "only Class A1 systems substantially deal with the problems of subversion." [2] (p. 22, col. 3). Heckman emphasizes this with a summary list of countermeasures when he points out [3] (Section 2.3), "Class A1 includes comprehensive requirements unique to this class that specifically focus on the problem of subversion of the security kernel itself. These requirements include the following development processes and artifacts." As stated in the introduction, NSA previously evaluated the GEMSOS security kernel at *TCSEC* "Class A1: Verified Protection" [1] [4].

In GEMSOS, the MAC policy is used to protect the system security kernel itself from subversion, and further protects resources (memory, storage, devices, etc.) from tampering, including alteration (integrity) or disclosure (confidentiality). The validity of the MAC policy and its correct enforcement for every reference to information is verifiable, due to the collection of engineering, manufacturing, support and delivery processes and practices used throughout the product life cycle.

The GEMSOS Trusted Distribution Interfaces leverage its high assurance MAC enforcement. These interfaces are part of the Class A1 evaluation, and use a "crypto seal" to cryptographically bind software and updates from the corporation's high integrity distribution system with a label for that high integrity source. Sealed software distributions arrive at the destination via open wireless and other untrusted (even malicious) transmission methods. The Trusted Distribution Interfaces running on the Intel-based GEMSOS processor destination platforms validate the data and label of each distribution before releasing it to the target systems. Altered and other low integrity distributions and updates cannot enter the destination because they do not have an intact crypto seal. The Trusted Distribution Interfaces leverage verifiable protection of the GEMSOS security kernel's high assurance label integrity and distributed key management mechanisms. GEMSOS provides functions and features that facilitate integrity checks on distributions of software, and will protect against malicious software attacks, too. The result is a system product that verifiably, with high assurance, protects itself and the information entrusted to it, even in the face of determined, persistent adversaries, throughout the life cycle - from design to decommissioning.

Furthermore, the system nature of the *TCSEC* Class A1 requirements is noteworthy. Heckman's *Information* paper reminds [3] (Section 2.3) "that the raison d'être for the *TCSEC* is to provide a method to evaluate the security of an entire system, not just a component". The reference monitor concept provides the way to evaluate systems created through composition of well-specified components. Heckman points out [3] (Section 2.6), "No general, closed-form, engineering-free solution to this problem has been found, and one may not be possible. Today, only two security composition methods have been proven to work: 'TCB subsets' and 'partitioned TCB'." Although it may seem tangential to composition, it is interesting to note that

the having “protection rings” in hardware is essential for most cases that apply the powerful TCB subsets method. Both methods rely on the services of a system-wide reference monitor to enforce protection and constrain the behavior of each component. Because the reference monitor is verifiably correct and secure, the various components need not all be secure themselves - they'll be simply unable violate the behavior constraints imposed upon them by the system reference monitor.

Three distinct classes of “kernel” products that are not “security kernels” are marketed today for cyber security. The following considers them in turn.

Operating System Kernels

Operating systems, such as Linux, Windows, IBM’s MVS, etc., are commonly described as “secure”. These large, monolithic operating systems represent commercial general purpose operating systems, and their size, complexity, and overall design prevent them from providing any meaningful protection against subversion and tampering by malware attacks. They claim to represent the "state of the art" or “best practice”, and their claims of security rest on a “paradigm [that] has for decades been known as ‘penetrate and patch’. . . But science tells us that trying to make a system secure in this way is effectively non-computable. Even after fixing known flaws, uncountable flaws remain.” [2] (p.21, col. 1). Observe that the booming business in anti-virus products is obvious and relevant empirical evidence of their failure to protect even themselves.

A few of these operating systems have been enhanced for MAC. An example is the Red Hat Enterprise Linux supporting SELinux that was previously evaluated as satisfying the Common Criteria Labeled Security Protection Profile (LSPP). [10] But none of these LSPP evaluations address subversion.

Separation (Partition) Kernels

Two examples of separation or partition kernels are the MILS architecture products from Green Hills Software (Integrity) and Wind River (VxWorks). (MILS stands for Multiple Independent Levels of Security – not to be confused with MLS, which stands for Multi-Level Security.) These products attempt to conform to the requirements set forth in the NSA published *Protection Profile for Separation Kernels* (SKPP) [11], which states:

> Unlike those traditional security kernels which perform all trusted functions for a secure operating system, a separation kernel’s primary security function is to partition (viz. separate) the subjects and resources of a system into security policy-equivalence classes, and to enforce the rules for authorized information flows between and within partitions. [11] (Section 1.2 “Overview”)
>
> For example, in one class of system security architecture, software programs enforce application-level (vs. kernel-level) security policies, within the constraints of the separation kernel’s policy. Examples of hosted software programs include multilevel secure reference monitors. [11] (Section 2.1 “Product Type”)

So, these products expressly exclude from their definition any system-wide MAC or other security policy reference monitor. Subversion is not systematically addressed. Their purpose is to provide isolation between security domains. They require that trustworthy middleware and applications be designed using formal methods to assure that they "define support for a coherent application-level security policy in the separation kernel's configuration data, as well as to ensure that the configuration data itself is coherent and self-consistent."

Of critical importance is the fact that, without a system-wide reference monitor, these products have no generally useful method of composing useful systems out of their various pieces. The only proven composition techniques of TCB partitions and TCB subsets used for a reference monitor are not applicable, and corresponding alternate techniques for use without it are currently not available.

The Aesec tag line reads “The power of verifiable protection.” We mean the power of verifiable protection is that GEMSOS enforces a system wide MAC and can protect an entire system, and this is regardless of the fact that the applications running on GEMSOS can be manifestly insecure. Heckman noted in the *Information* paper that SeaView [3] (Section 3.1) and our NFS demonstration [3] (Section 3.5.3) both used untrusted components for application and middleware components, but both could be treated in design as maintaining a Class A1 rating for the overall system, due to the Class A1 rating of the system-wide reference monitor GEMSOS provides.

This is completely different from the MILS approach adopted for separation kernels as expressly stated in the NSA-published literature above. With a separation kernel, each application must be independently evaluated and certified to be secure with high assurance. Each combination of applications running on the separation kernel must also be individually evaluated and certified, again with high assurance, that the combination of applications, separation kernel, and their respective configurations are secure. Simply put, the absence of a system-wide reference monitor in the separation kernel design definition turns each deployment and each application configuration into another formal analysis and design exercise for the middleware and application developer.

The consequence is that every new application configuration using a separation kernel must be individually responsible for its own formal definition, analysis, configuration and proof of security. It is also worth noting that formal methods are well outside the realm of usual and customary commercial software development skills and practices. It's exceedingly unlikely that they'll become dominant factors in product engineering development teams - ever.

The SKPP has been withdrawn by the U.S. Government as a validated protection profile. Green Hills successfully completed the evaluation of their Integrity product against the SKPP, but no other separation kernel product has done so to our knowledge.

One questionable marketing practice related to security kernels is worth noting in the context of MILS. As mentioned above, one of the MILS-designed separation kernel products (not listed as having been certified against the SKPP) is Wind River’s VxWorks MILS [12]. Wind River’s product description quotes a senior NSA security professional as saying about security kernels,

"[t]wo decades ago, a similar MLS system development would have taken 10 or more years, with monolithic secure operating system evaluation at $50 million to $100 million." This is consistent with Schell's statement in his *CACM* Viewpoint column [2] (p. 23, col. 1) that, "[i]t can be expected to take 10 – 15 years and tens of millions of dollars to build and evaluate a high-assurance security kernel." But what makes this Wind River document misleading is that it fails to acknowledge the important follow up point that very same *CACM* Viewpoint statement goes on to make [2]: "[h]owever, once completed, a general-purpose security kernel is highly reusable for delivering a new secure system in a couple of years. It is economical to use the same kernel in architectures for a wide variety of systems." As noted earlier, GEMSOS is just such a general-purpose security kernel.

Microkernels

One microkernel reportedly used by automakers today is the QNX Neutrino RTOS from BlackBerry. The QNX home page claims that QNX provides "multi-level, policy-driven security model incorporating best-in-class security technologies from BlackBerry…" QNX product literature [13] states that,

> As a true microkernel OS, it provides inherent protection and isolation for safety-critical software components – regardless of whether the system comprises of only safety-related components or a mix of safety and non-safety components. For example, with proper separation and isolation, the malfunctioning of the RPM gauge component on a digital instrument cluster cannot impact the master warning light component, even when both are running on the same hardware.

This is the right thing to say and it makes for good slideware. But it shows the importance of the third component of the Cyber Defense Triad: verifiability by an objective third party. Because in this case, it appears the CIA does not regard the manufacturer claims as credible. The recent *Wikileak*s dump of CIA hacks reportedly discloses [14], "…the CIA citing "vehicle systems" and a car operating system from QNX, owned by Blackberry Ltd, as "potential mission areas" for the CIA's "Embedded Devices Branch" to consider." Not only the CIA, but all state intelligence agencies, as well as terrorist and criminal organizations will view microkernels that are not independently verified at Class A1 as vulnerable to subversion with malicious software.

Another microkernel is the Security Enhanced L4 microkernel owned by General Dynamics C4 Systems and developed in conjunction with the Australian National Information and Communications Technology Research Center (NICTA) [15]. SEL4 is part of the L4 family of microkernels. Developers have used formal methods to demonstrate functional correctness against its specification.

SEL4 is claimed [15] (Section 9) to be "the first comprehensive verification of an entire general purpose OS kernel". This claim is negligent because GEMSOS, and its formal analysis and verification in compliance with the *TCSEC* Class A1, predates SEL4 by some 30 years. In fact, SEL4 marketing material has little, if any, recognition of the reference monitor concept at all. It does mention separation kernels, and cites the Green Hills separation kernel discussed above as related work. But lack of a reference monitor means that, like the separation kernels discussed

above, each SEL4 applications must be independently evaluated and certified to be secure with high assurance. Similarly, the composition techniques of TCB partitions and TCB subsets used for a reference monitor are not applicable, and no corresponding techniques are currently available. This is reflected when the SEL4 materials discuss [15] (Section 6) "building trustworthy systems on top of seL4, and the additional properties and analyses that are required from the kernel to do so." Just as with MILS separation kernels discussed above, this is work a large defense contractor or system integrator would love to do for the government for each new product, but it is not feasible in commercial industry.

Furthermore, subversion is not systematically addressed in the SEL4 literature. We find no evidence SEL4 has been evaluated by any qualified and objective third party against any published high assurance system criteria, such as those systematically codified in the *TCSEC* Class A1, or even carefully designed to meet such requirements.

Conclusion

GEMSOS provides a reusable system-wide reference monitor providing high assurance MAC security policy enforcement. MAC enforcement protects the reference monitor from tampering by malware, as well as the integrity and confidentiality of information (data and application code) entrusted to its care. Our engineering practices, processes, and procedures, together with innovative use of MAC enforcement by GEMSOS itself, all of which was documented by NSA in our Class A1 evaluations [1], effectively mitigate against the risk of subversion.

The result is a reusable, trusted device architecture that can be delivered using the whole range of Intel architecture processors for use in embedded, appliance, portable, mobile, server or massively parallel configurations. These trusted devices deliver economic benefit to developers by:

- using the same reusable, high assurance system-wide reference monitor for each device
- amortizing the cost of high assurance productization across a multitude of products and applications
- providing a consistent, scalable API interface across all devices
- allowing application developers to focus on product features and functionality rather than formal security

References

1. National Computer Security Center, *Final Evaluation Report, Gemini Computers, Incorporated, Gemini Trusted Network Processor, Version 1.01*, *NCSC-FER-94/008*, 28 June 1995, http://webapp1.dlib.indiana.edu/virtual_disk_library/index.cgi/1347159/FID1806/library/fers/ncsc-fer-94-008.pdf (accessed March 24, 2017).
2. Schell, Roger R., "Cyber defense triad for where security matters, *Communications of the ACM* 59, 11 (October 2016), 20-23. DOI: http://dx.doi.org/10.1145/3000606. This is available for the noncommercial use of readers of this Aesec technical note at: http://www.aesec.com/CACM-Schell-Cyber-Defense-Triad-Nov2016.html (accessed March 24, 2017).
3. Heckman, Mark R. and Roger R. Schell, "Using Proven Reference Monitor Patterns for Security Evaluation. Information," *Information*, 2016 Apr 26;7(2):23; http://dx.doi.org/10.3390/info7020023 (accessed March 24, 2017).
4. Aesec Corporation, *GEMSOS Security Kernel RTOS*, March 2, 2010, http://www.aesec.com/iot/Aesec-MLS-RTOS-100306a.pdf (accessed March 24, 2017).
5. Bensoussan, André, Charles T. Clingen and Robert C. Daley, "The Multics virtual memory: concepts and design." *Communications of the ACM* 15, no. 5 (1972): 308-318. http://www.multicians.org/fjcc2.html (accessed March 23, 2017).
6. Schroeder, Michael D., and Jerome H. Saltzer, "A hardware architecture for implementing protection rings." *Communications of the ACM* 15, no. 3 (1972): 157-170. http://www.multicians.org/protection.html (accessed March 23, 2017).
7. Caelli, William J., "Relearning 'Trusted Systems' in an Age of NIIP: Lessons from the Past for the Future," *Colloquium for Information Systems Security Education*, 2002 http://cisse.info/resources/archives/category/25-papers?download=241:cael-2002 (accessed March 24, 2017).
8. Department of Defense *Trusted Computer System Evaluation Criteria*, *DoD 5200.28-STD*, United States National Computer Security Center, December 1985 http://ftp.mirrorservice.org/sites/ftp.wiretapped.net/pub/security/info/reference/ncsc-publications/rainbow-books/5200.28-STD.pdf (accessed March 24, 2017).
9. *Wikipedia*, "Kernel (operating system)" https://en.wikipedia.org/wiki/Kernel_(operating_system) (accessed March 24, 2017).
10. atsec Information Security Corporation, *Common criteria evaluation and validation scheme validation report, IBM, Red Hat Linux version 5. Technical Report CCEVS-VR-07-0037*, NIST, NSA, 7 June 2007 https://www.niap-ccevs.org/st/st_vid10125-vr.pdf (accessed March 24 2017).
11. Information Assurance Directorate, *U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness Version 1.03*, 29 June 2007 https://www.commoncriteriaportal.org/files/ppfiles/pp_skpp_hr_v1.03.pdf (accessed March 24, 2017).
12. Wind River Systems, Inc., *Wind River High-Assurance Solutions for Aerospace & Defense*, February 2008. https://www.windriver.com/products/product-overviews/PO_MILS_Solution_Feb2008.pdf (accessed March 24, 2017).
13. QNX BlackBerry Subsidiary, *QNX OS for Automotive Safety*, http://www.qnx.com/products/certified_os/automotive-safety.html (accessed March 24, 2017)
14. *Reuters* Technology News "CIA 'mission' on cars shows concern about next-generation vehicles," Mar 8, 2017 http://www.reuters.com/article/us-cia-wikileaks-autos-idUSKBN16G09Z (accessed March 24, 2017)
15. Klein, Gerwin, June Andronick, Kevin Elphinstone, Toby Murray, Thomas Sewell, Rafal Kolanski, and Gernot Heiser. "Comprehensive formal verification of an OS microkernel." *ACM Transactions on Computer Systems (TOCS)* 32, no. 1 (2014): 2. http://www.nicta.com.au/pub?doc=7371 (accessed March 24, 2017).