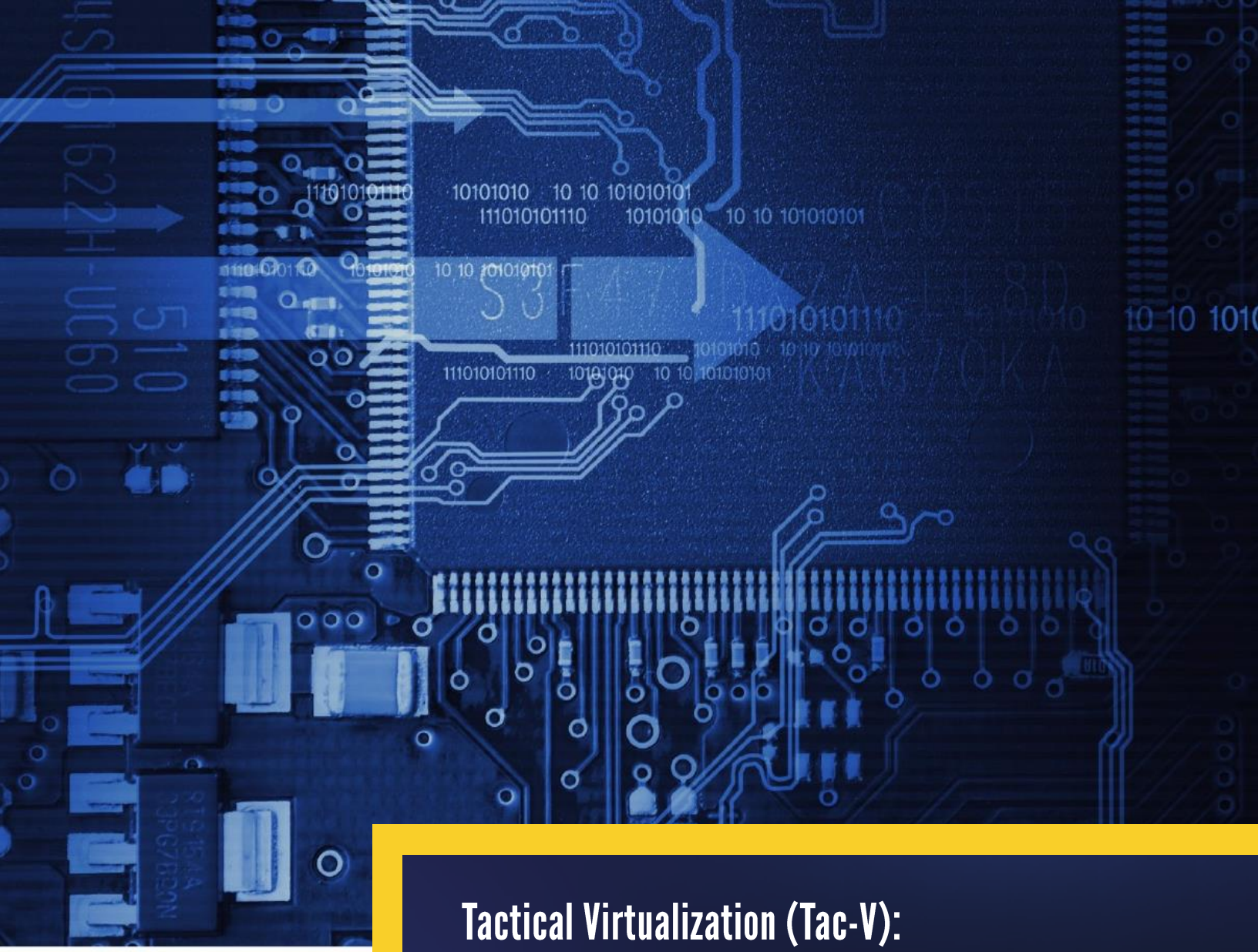

Tactical Virtualization (Tac-V): A New Design Paradigm for Combat Systems

STAR LAB SOFTWARE

Introduction

Cyber-attacks against high-value combat management and weapons control systems are a growing concern across the Department of Defense. Approaches to securing and defending safety-critical or mission-critical systems like the AEGIS Weapon System (AWS) and Ship Self-Defense System (SSDS) vary from basic network firewalls to persistent threat monitoring. The past few years has seen a significant up-tick in interest and research into cyber resilience and compromise-tolerance techniques. These techniques seek to increase the survivability of mission critical services in the face of a sophisticated, persistent adversary. Given the significant overlap with existing fault-tolerant principles, many aspects of this problem can be solved through redundancy and diversity. Despite this, a flexible and extensible cyber-resiliency solution – one which can be integrated into both existing and future combat systems with minimal re-work – is still necessary.

One approach to implementing compromise-tolerant systems is to leverage virtualization. Virtualization technologies have matured tremendously over the past decade, with significantly better hardware support and exceptional performance. This maturation has enabled their use in real-time embedded, multi-computer, and server-class processing domains. Additionally, many virtualization technologies, such as Xen, KVM, and OpenVZ are available, i.e., the solutions are open source and they receive significant support from communities comprised of individual developers, government entities, and private sector companies. Finally, virtualization technologies have already been successfully used to implement numerous secure systems, enabling breakthroughs in areas

like classified data separation, safety-critical component isolation, attack surface minimization, and moving target defense.

An opportunity exists to exploit virtualization technology on tactical hardware to achieve considerable security and maintenance benefits with little to no performance impact. This approach, referred to as Tactical Virtualization (Tac-V), can be used to create secure, resilient, and easily-upgradable combat platforms. Tac-V separates platform functionality and services into isolated execution domains. This allows underlying hardware and potentially vulnerable services to be segregated, thereby limiting the scope of a potential compromise. Additionally, these domains can be reduced to include only the minimally required functionality and interfaces – thereby reducing each domain's attack surface and providing the ability to perform quicker restart and migration. In addition to limiting and containing an attacker, this also enables an architecture in which individual services can be easily reconstituted and upgraded with limited impact to other components which rely on those services. As such, a software-defined architecture emerges. To achieve these benefits, a lightweight hypervisor serves as a trusted supervisory controller and separation kernel within the processor – configuring and controlling hardware resources and software execution in order to ensure and maintain the integrity of system operations.

Tac-V Benefits

Isolated Service Domains

Tac-V calls for dividing functionality into lightweight service domains with well-defined boundaries. This micro-disaggregation approach provides the foundation for cyber-attack isolation (risk mitigation) and resiliency by preventing errant or malicious code in one domain from being able to read/write memory, manipulate resources, or otherwise affect operations in another domain. Furthermore, memory protections and mandatory access controls ensures that the hypervisor itself remains isolated from the execution domains as well as malicious peripheral hardware.

Virtualization Without Performance Impacts

The barrier most often cited for adopting virtualization is performance. Modern processors, however, come standard with ample computing resources and hardware virtualization extensions enabling a hypervisor to control the allocation and subscription of system resources without requiring hardware emulation. This hardware-enabled virtualization can be used to ensure processing determinism and mitigate interruption (of the application virtual machines (VMs)) or interference by the hypervisor's own scheduler or other VMs. Modern hypervisors enable VMs to execute directly on bare-metal, without any of the performance impacts of emulation, hypervisor scheduling, and processor over-utilization. A Tac-V architecture goes beyond cloud / enterprise virtualization use cases – which primarily focus on oversubscription and hardware consolidation. Tac-V instead enables hardware consolidation and cyber resiliency of tactical platforms, while ensuring the requisite deterministic performance.

Combat System Cyber Resiliency

Using a Tac-V architecture minimizes the risk that cyber-attack response actions will negatively impact other subsystems. This unlocks the potential to investigate more aggressive and entirely new response actions that enable combat systems to continue providing required capabilities in the face of cyber-attacks.

Tac-V Security Characteristics

Security Through Separation and Isolation

The cornerstone feature of a Tac-V architecture is the ability to decompose software and hardware components into separate and isolated processing domains. Facilitating this is typically a trusted supervisory controller within the processor, like a hypervisor or separation kernel, that configures and controls hardware resources and software execution in order to ensure and maintain the integrity of system operations. This separation ensures software mission loads execute within private enclaves, even though they are running on the same physical processor boards. This helps to address safety and security concerns and improves overall mission assurance – especially given the reality of software vulnerabilities and cyber-attacks.

Functional Service Domains

Further decomposing the system such that critical services are segregated and isolated into service domains is also important in a Tac-V architecture. Traditional operating systems have limited protections between processes and application/system dependencies. Similarly, the operating system

kernel is not separate from the individual device services, thereby increasing the attack surface and enabling a single kernel exploit to compromise the availability of the entire system. A Tac-V architecture enables a new paradigm in which components (services and functions) are isolated via strong, hypervisor-enforced boundaries – ensuring interfaces between components are explicitly controlled. Separating the components improves the overall resiliency of the system as one component can no longer directly or indirectly affect another component. Additionally, partitioning the system into discrete components reduces the collective attack surface, increases overall system security (reduces and/or minimizes privilege escalation, resource starvation, denial of service, etc.), and lays the ground work for future fault tolerant application approaches.

Security Service Domains

In addition to functional domains, a Tac-V architecture will include security service domains. Several options are available, such as VPN domains, cryptographic service domains, security monitoring domains, and encrypted storage domains. Cryptographic services domains, for example, are often used to provide NSA/NIST certified cryptographic operations to the rest of the platform. Cryptographic services also allow system implementers to choose the most appropriate cryptographic implementation for their deployments without having to make changes to other domains. This flexibility enables developers to cope with changing NSA/NIST cryptographic requirements without having to constantly modify core applications or other system components.

Attack Surface Minimization

Tac-V prioritizes the use of build systems to remove unneeded features and components from the hypervisor and service domains. Consequently, the deployed system has a significantly reduced software attack space with hardware-enforced security controls. Additionally, the hypervisor can optionally perform runtime integrity monitoring of core service and application VMs. Finally, Tac-V requires the use of security mechanisms to deprivilege domains and restrict them to just provide mission-essential functionality, and nothing more. This makes it even more difficult for an attacker to compromise one domain in order to access another domain.

Cyber Resiliency

Traditional cyber defenses are often limited to only detecting compromises or patching a systems susceptibility to already-known attacks; however, such technologies are unlikely to stop a determined adversary. Resiliency approaches, on the other hand, aim to ensure that mission critical services and data are immutable – even in the face of an ongoing targeted attack. This is accomplished by not only anticipating adversary maneuvers, but also by withstanding successful compromises through cyber fault tolerant techniques, recovering critical assets through rapid reconstitution, and evolving and adapting to the threat through anomaly detection and moving target defense.

A Tac-V architecture leverages existing fault-tolerant protocols and recovery techniques to ensure functional and security service domains overcome / respond to / and recover from suspected compromise.

Resource Allocation and Quality of Service

A Tac-V architecture also facilitates the allocation of system resources to ensure performance. It provides pre-defined allocation and non-oversubscription of hardware resources to ensure processing determinism at the VM boundary. This ensures critical applications within each execution domain are able to complete operations without interruption or interference by the hypervisor's scheduler or other VMs. This architecture also enables any real-time operations and thru-data paths to continue as normal, unimpeded by the hypervisor and the rest of the system. These techniques have the added benefit of ensuring that compromised service domains cannot be used to execute a denial of service (DoS) attack by monopolizing processing resources.

Software Upgradability

To meet evolving mission and security requirements, a Tac-V architecture also includes an interface to facilitate efficient and secure upgrades to the hypervisor, service domains, and cryptographic support functions. Software updates are required to use approved public-key cryptography, as well as on-platform cryptographic services and replay prevention techniques to ensure that new software loads are trusted and current before performing the upgrade. These techniques prevent attackers from downgrading the system to software versions with known vulnerabilities or subverting the trustworthiness of the overall platform. The verification of updates (and rapid recovery) provides an additional failback mechanism to prevent botched system upgrades from disconnecting the platform.

Star Lab's Tac-V Solution

Star Lab's *Crucible* embedded virtualization product allows programs to implement a Tac-V architecture within combat systems. Crucible consists of Trueboot (trusted boot component designed for commodity Intel processors), Titanium Linux Security Suite (operating system hardening, mandatory access control, technology protection, data-at-rest protections), and the Crucible::RT hypervisor (secure tactical virtualization built from Xen). The Crucible Security Suite is a technical readiness level (TRL) 9 solution that provides proven security, from encryption-at-rest, through secure boot, and during software execution. The core of the Crucible Security Suite, Crucible::RT, is derived from the Xen Project hypervisor. The Xen hypervisor has hundreds of billions of hours of operational use across cloud, enterprise and high-performance computing (HPC) environments.

Crucible leverages Intel and ARM hardware features to partition, isolate, and secure a COTS platform. All of the Crucible components utilize existing, well-supported API interfaces within the kernel, hypervisor, and boot platform. The use of established, well-supported APIs decreases the cost and minimizes the effort to support older/newer kernel versions, enabling broader adoption by customers. Further, Crucible is designed for easy integration into existing program production processes, with feature-rich tooling that makes using it intuitive and straightforward. Finally, Crucible is maintained as a commercial product independent and external to program funding, with major releases every 6-months. New features and functionality are continuously being developed at no additional cost to programs.

Crucible and Titanium support most Intel / ARM architectures. Additionally, Crucible / Titanium supports all Linux distributions currently in use across DOD and its contractors, including real-time variants of Linux. Support for non-Linux VMs such as VxWorks or Windows within Crucible::RT is also supported on a per-program basis.

Crucible, which includes Titanium is certifiable as a MILS separation kernel, and is additionally capable of being certified as a CSFC solution. Star Lab is currently pursuing (and is under contract with an accredited evaluation laboratory) for the evaluation of Crucible / Titanium against a total of 6-distinct NIAP protection profiles and expects to have full government approval in Q4 of 2019. These 6 protection profiles will enable Crucible to be used for: 1) MILS Separation kernel / Hypervisor; 2) CSFC Data-at-rest (file-based encryption for Linux); and 3) CSFC Data-At-Rest (software full disk encryption for Linux). In addition to the NIAP profiles, Star Lab also maintains a comprehensive IA / Cybersecurity mapping of Crucible / Titanium against the NIST 800-53 risk management framework (RMF) requirements and enables programs to address 96% of those requirements.

Mapping Crucible to Tac-V

Crucible is the only commercially-available virtualization solution specifically designed to aid defense programs in deploying a Tac-V architecture. It addresses all the priorities mentioned above: separation and isolation; service domains; attack surface minimization; cyber resiliency; resource allocation and quality of service; and upgradability.

- Separation and Isolation – Crucible::RT operates as a separation kernel / hypervisor to isolate execution domains from each other, rogue peripheral hardware, and the hypervisor itself. This provides the foundation for cyber-attack isolation by preventing errant or malicious code in one domain from being able to read/write memory, manipulate resources, or otherwise affect operations in another domain. Furthermore, the memory protections and mandatory access controls configured by Crucible ensure that the hypervisor itself remains isolated from the execution domains as well as malicious peripheral hardware. The isolation and separation also provide mitigations for a variety of side-channel attacks on the platform.

- Service Domains – Crucible includes network and storage domains by default. Also, Crucible makes it easy to isolate underlying hardware and potentially vulnerable drivers, thereby limiting the scope of a potential exploit. Finally, Crucible includes a standard means to facilitate inter-domain communication. This functionality also includes methods for defining and enforcing Mandatory Access Control (MAC) policy to control inter-domain communication.

- Attack Surface Minimization – Crucible implements a number of best practices required for high-assurance systems, including comprehensive auditing of system activities, mandatory access control policies, and secure-by-default configuration options. The Crucible build system makes extensive use of KCONFIG in order to remove unneeded features and components from the baseline Xen hypervisor. This enables Crucible to be deployed with a significantly reduced attack space and removes the vast majority of non-essential

features and capabilities (such as qemu, memory ballooning, and transcendent memory) which have been the source of software vulnerabilities in the past. Additionally, Star Lab is performing a line-by-line traceability exercise of Xen to ensure all of the code maps to low-level requirements, and to provide additional assurances that Crucible::RT is not being deployed with any unnecessary code. This activity further inhibits the introduction of software vulnerabilities into the system.

- Cyber Resiliency – Crucible includes functionality to respond to a suspected compromise via fail-over. In cases when fail-over is not possible, Crucible uses snapshot-based reconstitution of compromised services using mechanisms for creating and maintaining trusted snapshots for speedy recovery of failed services.

  It is also possible, using Crucible, to enable the rapid deployment of other cyber resiliency response actions, such as installing communications filters to block out specific attackers. More advance resiliency solutions might involve the automated diversification and replication of mission-critical applications, with real-time state comparison / vote exchange between the replicas running in parallel. Such an approach would allow for the detection and masking out of inter-replica divergences due to successful cyber-attacks. Most importantly, enabling response actions within Crucible minimizes the risk response actions will affect other platform services.

- Resource Allocation and Quality of Service – Crucible's Foundry Tools enable pre-defined allocation, provisioning, and non-

oversubscription of hardware resources to ensure processing determinism at the VM boundary. This ensures critical applications within each execution domain are able to complete operations without interruption or interference by the hypervisor's scheduler. Crucible:RT enables real-time operations to continue as normal, unimpeded by the hypervisor.

- **Upgradability –** Crucible assumes the role of hardware compatibility and platform BSP support. This enables domains (VMs) to be upgraded without needing to worry about hardware and driver compatibility. Crucible enables VMs to execute on physical card, while remaining agnostic to the underlying architectural concerns. Additionally, Crucible provides an easy upgrade path for programs not currently utilizing virtualization, and enables a single VM to be created which utilizes all platform resources.

In addition to satisfying Tac-V considerations, Crucible also includes capabilities for:

- **Secure Boot –** Crucible's TrueBoot functionality uses a trusted instantiation process to ensure that it will only decrypt and execute sensitive application software within authorized and verified mission computing environments. On non-authorized, instrumented or modified hardware, the software remains fully-protected against exposure and reverse-engineering attacks.

- **Technology Protection –** The Crucible Security Suite is uniquely designed to shield sensitive software technologies from unauthorized access, theft, or reverse engineering. These

protections are in place at rest, during boot, and throughout system operation. In addition to TrueBoot, Crucible provides runtime memory protection, mandatory access controls, and anti-debug capabilities to protect sensitive applications and data at runtime.