CADEIA DE CONFIANÇA APLICADA A NUVEM PRIVADA

ROSSANO PABLO PINTO[1]
[1]*Fatec Americana – CST Segurança da Informação*
rossano.pinto@fatec.sp.gov.br

Chain of trust applied to private cloud

Eixo Tecnológico: *Informação e Comunicação*

Resumo

A nuvem privada da FATEC Americana é composta por nós computacionais e switches de software que permitem a criação de diferentes topologias, sendo gerenciados por um controlador SDN. Para assegurar que os nós computacionais estejam íntegros e protegidos contra-ataques sejam eles alterações acidentais ou maliciosas de software e configurações, a adoção de um sistema de verificação de integridade à prova de adulterações pode ser altamente benéfica para a gestão do ambiente computacional. Processadores das arquiteturas x86 e ARM oferecem mecanismos para estabelecer uma raiz de confiança no processo de inicialização, permitindo a verificação da integridade do firmware. No entanto, os demais componentes da cadeia de boot não são automaticamente verificados e permanecem vulneráveis a modificações. Este projeto tem como um de seus principais objetivos o estudo e a implementação de mecanismos de verificação de integridade que cubram toda a sequência de inicialização da máquina desde a energização até o carregamento completo das aplicações com ou sem ancoragem em hardware. Isso inclui a verificação do bootloader, kernel, initrd, módulos executáveis, configurações de inicialização e integridade durante o tempo de execução. Como entregáveis, espera-se: um estudo sobre o estado da arte em mecanismos de integridade baseados em raiz de confiança para medições, a elaboração de provas de conceito envolvendo boot assinado e verificado, além da implantação de nós computacionais na nuvem privada com suporte a atestação de integridade remota e/ou local.

Palavras-chave: Nuvem privada, Cadeia de confiança, Módulo de Plataforma Confiável (TPM).

Abstract

The private cloud at FATEC Americana is composed of computing nodes and software switches that enable the creation of various topologies, all managed by an SDN controller. To ensure that the computing nodes remain intact and protected against attacks — whether accidental or malicious changes to software or configurations — the adoption of a tamper-proof integrity verification system can be highly beneficial for managing the computing infrastructure. Processors based on x86 and ARM architectures offer mechanisms to establish a root of trust during the boot process, allowing for firmware integrity verification. However, other components in the boot chain are not automatically verified and remain vulnerable to tampering. One of the main objectives of this project is to study and implement integrity verification mechanisms that cover the entire machine startup process — from power-on to the full loading of applications — with or without hardware anchoring. This includes verifying the bootloader, kernel, initrd, executable modules, boot configurations, and runtime integrity. Expected deliverables include: a study on the state of the art in integrity assurance mechanisms based on root of trust for measurements, the development of proof-of-concept implementations involving signed and verified boot, and the deployment of computing nodes in the private cloud with support for remote and/or local integrity attestation.

Key-words: *Private Cloud, Chain of Trust, Trusted Platform Module (TPM).*

1. Introdução

Uma nuvem é formada por nós computacionais responsáveis por prover serviços como processamento paralelo e armazenamento. Esses nós podem ser físicos ou virtuais, assim como os dispositivos responsáveis por sua interconexão, que podem também ser implementados via software. Em datacenters, é comum o uso de switches virtuais, especialmente quando diversos nós virtuais operam sobre um único nó físico. Essa abordagem permite aproveitar a capacidade computacional de máquinas individuais para construir redes complexas e isoladas, atendendo a

diferentes requisitos de desempenho, segurança e segmentação. Em projetos anteriores, a adição e operação de nós computacionais na nuvem eram feitas de forma livre, sem exigência de verificação de integridade. Esse cenário pode representar uma vulnerabilidade significativa: a introdução de um nó malicioso compromete todo o ambiente, expondo a infraestrutura a diversos riscos de segurança. O projeto anterior também abordou o uso de controladores SDN [1,2] e o desenvolvimento de aplicações para o ONOS. Sendo um controlador centralizado, o ONOS pode se tornar um ponto único de falha e um alvo atrativo para ataques. Diante disso, torna-se essencial garantir a integridade de todos os nós da rede — sejam eles nós de processamento ou de gerenciamento.

A garantia de integridade vai além das práticas tradicionais de segurança da informação, pois envolve diferentes camadas e aspectos do sistema. O objetivo geral deste projeto é desenvolver e implementar um sistema de verificação de integridade para os nós computacionais da nuvem privada da FATEC Americana, assegurando a confiabilidade desde a admissão de novos nós até sua operação contínua. Já os objetivos específicos são: 1- Selecionar uma ou mais plataformas de verificação de integridade compatíveis com a nuvem privada da FATEC Americana; 2 - Explorar em profundidade os recursos oferecidos pela(s) plataforma(s) escolhida(s); 3 - Oferecer uma cadeia de confiança ancorada em software e/ou hardware; 4 - Explorar mecanismos de atestação de integridade local e remota; 5 - Avaliar a possibilidade de estender ou aprimorar funcionalidades da plataforma selecionada; 6 - Aplicar conceitos de verificação de integridade a imagens binárias e configurações do sistema.

Os resultados esperados do projeto são: 1 - Provas de conceito de sistemas de verificação de integridade; 2 - Protótipo funcional operando na nuvem da FATEC Americana; 3 - Produção de materiais didáticos (apostilas, tutoriais, repositórios online); 4 - Publicação dos resultados em eventos científicos e periódicos especializados.

2. Materiais e métodos

2.1. Materiais

O projeto utiliza os computadores dos laboratórios de informática em dias e horários em que estejam disponíveis, sem que haja interferência no cotidiano de uso, e sem prejudicar as atividades acadêmicas regulares. Além disso, alguns computadores de uso pessoal do pesquisador e máquinas virtuais são utilizados para fazer provas de conceito (PoC). Apenas softwares livres são utilizados na excução dos experimentos e criação das PoCs. Para uma menor interferência na instalação local dos softwares presentes nos computadores da Fatec Americana, a plataforma de nuvem roda em cima de máquina virtuais, sendo os requisitos mínimos: VirtualBox 7.0 ou superior com emulação do chip TPM 2.0 [3,4] habilitado, firmware EFI, Linux 6.0 ou superior, 1GB de memória RAM para sistemas puramente texto (mas o ideial deveria partir de 4GB) e disco de 10GB.

2.2. Metodologia

Como esta pesquisa é classificada como tecnológica e pertence à área de Ciência da Computação, o método adotado é o experimental. Isso se justifica pela ampla disponibilidade na Fatec Americana de computadores e softwares necessários à condução dos experimentos propostos. Nesse contexto, o projeto contempla exploração de plataformas-alvo, incluindo a seleção de ferramentas existentes ou o desenvolvimento de software próprio (como soluções de atestação remota e controle de falhas de integridade) para suporte à gerência SDN de uma

nuvem privada. Dado o caráter experimental da pesquisa, os experimentos estão organizados em três grandes grupos, cada um composto por atividades específicas:

1. Seleção, instalação e avaliação de um sistema de verificação de integridade - Atividades: Instalar e configurar o sistema de verificação de integridade escolhido; identificar funcionalidades que atendam aos requisitos de integridade em ambientes de nuvem privada; executar essas funcionalidades visando a proteção dos elementos da nuvem; coletar dados operacionais relacionados à eficiência e eficácia do sistema adotado.

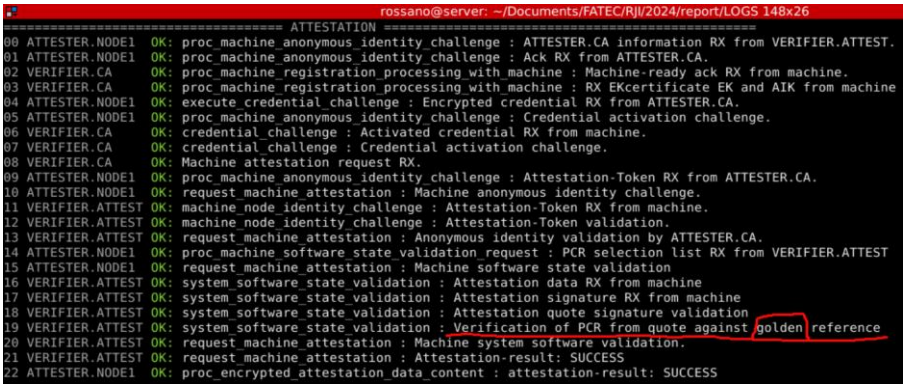
2. Exploração de mecanismos de atestação local e remota - Atividades: Compreender o funcionamento dos mecanismos de atestação local e remota; Estudar políticas e ações que possam ser definidas com base nos resultados das atestações; Testar na prática os mecanismos de atestação disponíveis; Coletar métricas de desempenho e confiabilidade dos processos de atestação.

3. Desenvolvimento ou extensão de políticas para verificação de integridade - Atividades: Identificar políticas existentes que possam ser estendidas; Desenvolver novas políticas que atendam às necessidades específicas da nuvem privada; Implementar e aplicar essas novas políticas no ambiente experimental; Avaliar a eficiência e a eficácia das políticas desenvolvidas com base em dados operacionais.

3. Resultados e Discussão

Na plataforma de nuvem desenvolvida em projetos anteriores, foram integrados ou configurados os seguintes componentes: **TPM** (Trusted Platform Module) — chip presente em placas-mãe ou emulado no VirtualBox (versão 2.0); **IMA/EVM** (Integrity Management Architecture / Extended Verification Module) [5] — arquitetura do Linux para garantir a integridade do sistema; **Attester** — serviço executado no nó cliente que envia informações de integridade para o Verifier; **Verifier** — serviço responsável por realizar a atestação remota e validar a integridade do nó cliente. A partir dessas configurações, para que um nó seja aceito na nuvem, ele deve passar por um processo de registro. Após registrado, qualquer alteração não autorizada no hardware ou software do nó será detectada, resultando em falha na atestação e exclusão automática do nó pelo gerente da nuvem, com geração de log para análise. A Figura 1 exibe o caso de atestação remota com sucesso (nó íntegro). Já a Figura 2 exibe o caso de um nó não íntegro. Outra abordagem baseada em IMA/EVM também utiliza o TPM como raiz da cadeia de confiança, mas sem a necessidade de um Attester. Nesse modelo, se for detectada violação de integridade, partes do sistema ou aplicações deixam de funcionar, tornando o nó indisponível para a nuvem.

Fig. 1 - Atestação remota resultou em SUCESSO: Nó íntegro.



Fonte: (Autor, 2024).

Fig. 2 - Atestação remota resultou em FALHA: Nó comprometido.

Fonte: (Autor, 2024).

No total, foram elaboradas 4 provas de conceito (PoC): **PoC 1** - Exploração inicial do suporte linux a TPM 2.0; **PoC 2** - Garantia de integridade online (IMA) - Atestação local; **PoC 3** - Garantia de integridade offline (EVM) - Atestação local; **PoC 4** - Atestação remota.

3.1. Análise de desempenho

Os protótipos foram testados simultaneamente em 10 máquinas. Como o ambiente de nuvem atual é baseado em máquinas virtuais (VMs), evitando alterações no sistema local, a implantação ocorre por meio da simples replicação das VMs com as configurações necessárias para garantir a integridade.Para avaliar o desempenho das PoCs, foram realizados testes focados no **tempo de boot** (Tabela 1) e na **criação de arquivos** (Tabela 2), considerando que o mecanismo appraise calcula os hashes IMA e EVM para cada arquivo acessado.

Metodologia dos testes - Foram utilizadas três configurações distintas: Kernel original sem suporte a IMA/EVM; Kernel customizado sem suporte a IMA/EVM; Kernel customizado com suporte a IMA/EVM. Para cada cenário (reinicialização e manipulação de arquivos), foram realizadas 100 medições, com cálculo das médias de tempo, a fim de reduzir o impacto de variações na carga do sistema hospedeiro. Nos testes de criação de arquivos, foram utilizados dois tamanhos: 1.2 GB e 12 MB.

Tab. 1 - Tempo médio de boot: sem e com IMA/EVM.

Tempo médio de boot em 100 amostras			
Configuração	**Tempo (s)**		
	kernel space	userspace	total
Debian 12 stock kernel 6.1.0-17-amd64	4,8045	2,8299	7,6348
Debian 12 custom kernel 6.7.1-ima-rpp	2,6356	3,0525	5,6886
Debian 12 custom kernel 6.7.1-ima-rpp (IMA/EVM)	2,9789	4,0596	7,0389

Fonte: (Autor, 2024).

Observa-se na Tabela 1 que o boot com o kernel original do Debian 12 foi o mais demorado, devido à presença de diversas opções compiladas adicionais em comparação ao kernel

customizado para a PoC. Já com os kernels customizados, a ativação do EVM resultou em um acréscimo de aproximadamente 1,3 segundos no tempo de boot. O maior impacto ocorreu na execução de código em userspace, uma vez que envolve a manipulação de um grande número de dados lidos de arquivos.

Tab. 2 - Tempo médio na criação de arquivos: sem e com IMA/EVM.

Tempo médio de criação de arquivo em 100 amostras						
Configuração	**Tempo (s)**					
	kernel space		userspace		total	
	1.2GB	**12MB**	**1.2GB**	**12MB**	**1.2GB**	**12MB**
Debian 12 stock kernel 6.1.0-17-amd64	0.1662	0.0066	4.1666	0.0826	4.3922	0.0893
Debian 12 custom kernel 6.7.1-ima-rpp	0.1643	0.0056	4.2807	0.0835	4.5431	0.0894
Debian 12 custom kernel 6.7.1-ima-rpp (IMA/EVM)	1.3258	0.0126	5.6648	0.0951	7.0668	0.1078

Fonte: (Autor, 2024).

A Tabela 2, destaque em cinza, apresenta os resultados das três configurações para arquivos de 1.2GB. Observa-se um desempenho semelhante entre as configurações sem EVM. No entanto, com EVM ativado, há um aumento de aproximadamente 55% no tempo total em comparação com o kernel 6.7.1-rpp sem EVM, passando de 4.5431s para 7.0668s. Já nas colunas sem destaque, os resultados para arquivos de 12 MB são exibidos. Nesse caso, o tempo total no sistema com EVM ativado aumentou cerca de 20% em relação ao sistema sem EVM no kernel 6.7.1-ima-rpp, variando de 0.0894s para 0.1078s. Ao comparar os tempos totais para arquivos de 1.2GB e 12MB, observa-se que arquivos maiores impõem uma carga maior em sistemas com IMA e EVM ativados. Isso ocorre porque o cálculo do hash do conteúdo do arquivo é necessário para comparação com o atributo security.ima, e o tempo dessa operação depende diretamente do tamanho do arquivo – quanto maior o arquivo, maior o tempo de processamento. Por outro lado, o tamanho do arquivo não impacta o tempo necessário para gerar o atributo security.evm, pois este é derivado apenas do hash dos metadados.

4. Considerações finais

Os objetivos estabelecidos no cronograma aprovado foram alcançados com sucesso. A incorporação de um mecanismo de verificação de integridade para os nós da nuvem aumentou significativamente a segurança do ambiente, impedindo a execução de códigos maliciosos. Esse novo requisito não funcional revelou-se essencial em cenários de uso compartilhado, como ocorre em infraestruturas de nuvem. Em especial, o aproveitamento dos laboratórios de informática da FATEC Americana como nós da nuvem reforça a importância dessa medida, uma vez que a execução de sistemas comprometidos pode impactar, ainda que indiretamente, no funcionamento dos próprios laboratórios. Assim, os resultados obtidos contribuem para a prevenção de falhas e para a melhoria da confiabilidade do ambiente computacional.

Referências

[1] GÖRANSSON, PAUL; BLACK, CHUCK. **Software Defined Networks: A Comprehensive Approach**. 2014. Morgan Kaufmann.

[2] PETERSON, LARRY L.; CASCONE, CARMELO; O'CONNOR, BRIAN; VACHUSKA, THOMAS; DAVIE, BRUCE. **Software-Defined Networks: A Systems Approach**. 2022. Disponível em https://sdn.systemsapproach.org/uses.html (https://sdn.systemsapproach.org/uses.html). Acesso em 03/06/2024.

[3] NG, RAYMOND. **Trusted Platform Module TPM Fundamental**. 2008. https://www.cs.unh.edu/~it666/reading_list/Hardware/tpm_fundamentals.pdf

[4] TRUSTED COMPUTING GROUP (TCG). **Trusted Platform Module Library Part 1: Architecture**. November 8, 2019. Disponível em https://trustedcomputinggroup.org/wp-content/uploads/TCG_TPM2_r1p59_Part1_Architecture_pub.pdf. Acesso em 25/02/2025.

[5] SAFFORD, DAVID; KASATKIN, DMITRY; ZOHAR, MIMI. **Integrity Measurement Architecture (IMA)**. Disponível em https://sourceforge.net/p/linux-ima/wiki/Home/. Acesso 25/02/2025.