SECURITY GUIDANCE FOR 5G CLOUD INFRASTRUCTURES

Part II:
Securely Isolate Network Resources

2021

TLP:WHITE

DISCLAIMER OF ENDORSEMENT

PURPOSE

NSA and CISA developed this document in furtherance of their respective cybersecurity missions, including their responsibilities to develop and issue cybersecurity specifications and mitigations. This information may be shared broadly to reach all appropriate stakeholders.

CONTACT

Client Requirements / Inquiries: Enduring Security Framework nsaesf@cyber.nsa.gov

Media Inquiries / Press Desk:

- NSA Media Relations, 443-634-0721, MediaRelations@nsa.gov
- CISA Media Relations, 703-235-2010, CISAMedia@cisa.dhs.gov

TABLE OF CONTENTS

BACKGROUND

The Enduring Security Framework (ESF) hosted a 5G study group comprised of government and industry experts over the course of eight weeks during the summer of 2020 to explore potential threat vectors and vulnerabilities inherent to 5G infrastructures. At the conclusion of the study, the group recommended a three-pronged approach to explore this threat space[1]:

1. Identify and assess threats posed to 5G;
2. Determine what standards and implementations can achieve a higher baseline of 5G security; and
3. Identify risks inherent to the cloud that affect 5G security.

In support of this task, the ESF established a 5G Cloud Working Panel to engage with experts across government and industry to document 5G cloud security challenges, threats, and potential mitigations, to include guidance, standards, and analytics. The result of this collaboration is a four-part series of publications that addresses the third task identified by the 5G study group: applying a threat-based approach to identify and mitigate risks in 5G networks that derives from the use of cloud technologies, and providing mitigations that can be applied to harden 5G cloud infrastructure.

SCOPE

This four-part series builds on the ESF *Potential Threat Vectors to 5G Infrastructure* white paper, released in May 2021, which focused specifically on threats, vulnerabilities, and mitigations that apply to the deployment of 5G cloud infrastructures.[2]

Although all 5G network stakeholders can benefit from this guidance, the recommendations are intended for service providers and system integrators that build and configure 5G cloud infrastructures. This includes core network equipment vendors, cloud service providers, integrators, and mobile network operators. The audience for each set of recommendations will be identified throughout the series, providing a layered approach to building hardened 5G cloud deployments.

5G CLOUD SECURITY CHALLENGE OVERVIEW

5G networks are being designed to handle the bandwidth, compute, and storage requirements that will be required for a predicted massive increase in network capacity as

[1] The ESF is a cross-sector working group that operates under the auspices of Critical Infrastructure Partnership Advisory Council (CIPAC) to address threats and risks to the security and stability of U.S. national security systems. It is comprised of experts from the U.S. government as well as representatives from the Information Technology, Communications, and the Defense Industrial Base sectors. The ESF is charged with bringing together representatives from private and public sectors to work on intelligence-driven, shared cybersecurity challenges.

[2] ESF, *Potential Threat Vectors to 5G Infrastructure*, 2021. https://www.nsa.gov/news-features/press-room/Article/2601078/nsa-odni-and-cisa-release-5g-analysis-paper

well as connected devices. For scalability, resilience, and agility, 5G networks leverage cloud infrastructures, both in the radio access network, core, and network edge. Cloud technologies underpin the implementation of virtual networking in 5G, enabling the dynamic allocation and management of networks for specific use cases, mobile network operators, or customers.

A characteristic of cloud infrastructure that presents a significant security challenge in 5G is multitenancy, the use of a shared physical infrastructure by multiple cloud infrastructure customers, e.g., mobile network operators. Multitenancy highlights the need to harden and securely configure technologies that isolate the workloads (e.g., virtualization/ containerization) for each of those customers. In addition, cloud providers and mobile network operators may share security responsibilities in a manner that requires the operators to take responsibility to secure their tenancy "in the cloud." An additional factor creating security challenges is the increasing deployment of a multi-cloud deployment model in 5G with diverse and evolving architectures and design approaches used by wireless carriers.

5G THREAT

Among the threat vectors presented in the *Potential Threat Vectors to 5G Infrastructure* analysis paper, several pertained to 5G cloud infrastructure, including *Software/ Configuration*, *Network Security*, *Network Slicing*, and *Software Defined Networking*.

5G networks, which are cloud-native, will be a lucrative target for cyber threat actors who wish to deny or degrade network resources or otherwise compromise information. To counter this threat, it is imperative that 5G cloud infrastructures be built and configured securely, with capabilities in place to detect and respond to threats, providing a hardened environment for deploying secure network functions. It is also important that 5G network functions be implemented using security best practices. This four-part series will address the former, providing guidance on hardening 5G cloud infrastructure deployments that are driven by threat information. This approach supports the May 2021 Presidential Executive Order on *Improving the Nation's Cybersecurity*, which called for secure products and services and enabling easier detection of unexpected behaviors and actions.[3]

5G CLOUD SECURITY GUIDANCE

Based on preliminary analysis and threat assessment, the Cloud Working Panel concluded that the top 5G cloud infrastructure security challenges could be divided into a four-part series that addressed different aspects of securing 5G clouds, facilitating the application of broad sets of mitigations.

[3] Executive Office of the President, *Executive Order on Improving the Nation's Cybersecurity*, 2021. https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity

- **Part I: Prevent and Detect Lateral Movement**: Detect malicious cyber actor activity in 5G clouds and prevent actors from leveraging the compromise of a single cloud resource to compromise the entire network.

- **Part II: Securely Isolate Network Resources**: Ensure that there is secure isolation among customer resources with emphasis on securing the container stack that supports the running of virtual network functions.

- **Part III: Protect Data in Transit, In-Use, and at Rest**: Ensure that network and customer data is secured during all phases of the data lifecycle (at-rest, in transit, while being processed, upon destruction).

- **Part IV: Ensure Integrity of Infrastructure**: Ensure that 5G cloud resources (e.g., container images, templates, configuration) are not modified without authorization.

Zero Trust is the concept that perimeter defenses are no longer sufficient to secure a network, and that there should always be an assumption that a threat actor has established a foothold in the network[4]. This four-part series will document best practices that strive to bring a Zero Trust mindset into 5G cloud endpoints and growing multi-cloud environments. All actions should be explicitly verified and monitored. Although the best practices documented in this series do not constitute a complete Zero Trust template for securing 5G cloud infrastructures, if the best practices are applied, a 5G cloud environment will have made significant strides toward the implementation of Zero Trust principles.

[4] NIST Special Publication 800-207. Zero Trust Architectures.
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

SECURELY ISOLATE NETWORK RESOURCES (POD SECURITY)

Pods are the isolated environments used to execute 5G network functions in a 5G container-centric or hybrid container/virtual network function design and deployment. Pods provide highly configurable, flexible workloads that can be scaled and orchestrated from a central control plane, while enforcing isolation of each workload. The scale and interoperability requirements of 5G cloud components makes securely configuring Pods a challenging but important ongoing effort. A strong Pod security posture leverages containerization technology to harden the deployed application, protects interactions between Pods, and detects malicious/anomalous activity within the cluster.

Part II of this four-part series will describe several aspects of Pod security including:

- Strengthening Pod isolation, such as limiting permissions on deployed containers;
- Cryptographically isolating critical Pods using trusted execution environments;
- Using best practices to avoid resource contention & DOS attacks;
- Implementing container image security through build processes, scanning, and enhancements to the trust environment; and
- Implementing real-time threat detection through minimizing noise, curating baseline behavior, and alerting on anomalous activity.

GUIDANCE FOR ENABLING ISOLATION OF RESOURCES

This section provides guidance and rationale of strengthening isolation of containers. The Center for Internet Security (CIS) Benchmarks for Kubernetes[5], a container orchestrator, and Docker[6], a container platform, provides additional implementation details for a secure environment with multi-tenant isolation.

Audience: Cloud Providers, Mobile Network Operators, Customers

Guidance/Mitigations:

RESTRICT CONTAINERS RUNNING IN PRIVILEGED MODE

Containers share the same kernel as the host, and containers that run in privileged mode inherit capabilities associated with the root. If an attacker were to exploit a vulnerability in the kernel to escape the isolation boundaries of the Container Runtime, they could escalate their privileges and gain access to sensitive information including Kubernetes Secrets. Additionally, the attacker could perform any application programming interface requests authorized for the kubelet, which could allow them to move laterally through the cluster.

Reject Pods with containers configured to run as privileged using technical controls and policies provided by the container orchestration platform. Seldom do containers need the types of privileges associated with root to function properly, but exceptions may occur. For example, certain applications and plug-ins such as kube-proxy must run as privileged to configure the host's network settings. In these cases, scope privileged Pods to a particular namespace (such as kube-system) and limit access to that namespace. For all other service accounts/namespaces, implement a highly restrictive policy and provide scoped exceptions where necessary.

DO NOT RUN PROCESSES IN CONTAINERS AS ROOT

Containers run as root by default. This could be problematic if an attacker is able to exploit a vulnerability in the application and gain arbitrary execution in the container. The Kubernetes PodSpec includes a set of fields that specify the user and/or group to run the application. Alternatively, the Dockerfile USER directive instructs the engine to run the container as a non-root user. Container orchestration platforms provide technical controls and policies to mandate non-root execution.

DO NOT ALLOW PRIVILEGED ESCALATION

Privileged escalation allows a process to change the security context under which it is running. Sudo is a good example of this, as are binaries with the SUID or SGID bit. Privileged escalation is a way for users to execute a file with the permissions of another user or group.

[5] CIS Benchmarks Securing Kubernetes, 2021. https://www.cisecurity.org/benchmark/kubernetes/
[6] CIS Benchmarks Securing Docker, 2021. https://www.cisecurity.org/benchmark/docker/

Container orchestration platforms provide technical controls and policies to prevent privilege escalation.

RESTRICT THE USE OF HOSTPATH

hostPath is a volume that mounts a directory from the host directly to the container. Rarely will Pods need this type of access. By default, Pods that run as root will have write access to the file system exposed by *hostPath*. This could allow an attacker to modify the kubelet settings, create symbolic links to directories or files not directly exposed by the hostPath (e.g., `/etc/shadow`), install SSH keys, read secrets mounted to the host, and take other malicious actions. Container orchestration platforms provide technical controls and policies to restrict directories used by hostPath and ensure that those directories are read only.

CRYPTOGRAPHICALLY ISOLATE CRITICAL CONTAINERS USING TEES

Today, it is near-universal practice in cloud and enterprise to protect data at rest using strong encryption, such as AES 256, in local and/or network-attached storage. However, when the same data is being processed by the central processing unit (CPU), it is held as plain text in memory and not protected by encryption. Memory contains high-value assets such as storage encryption keys, session keys, personally identifiable information (PII), and credentials. For sensitive/regulated workloads, there is a desire to protect data-in-use from the underlying privileged system stack. Therefore, it is critical that data in memory has comparable protection to data at rest in storage devices. This is the focus of confidential computing – protecting data in use on compute devices using hardware-based techniques like Trusted Execution Environments (TEE).

A TEE is an area in memory protected by the processor in a computing device. Hardware ensures confidentiality and integrity of code and data inside a TEE. The code that runs in the TEE is authorized, attested, and verified. Data inside a TEE cannot be read or modified from outside the TEE even by privilege system processes. Data is only visible while in the CPU cache during execution.

Audience: Mobile Network Operators, Customers

Guidance / Mitigations

TEEs reduce the need to trust firmware and software layered on the system processing the workload. The trusted compute base (TCB), the hardware, firmware, and software components acting as the trusted system, is very small. In most TEEs, the TCB is the CPU (hardware and microcode) and the code defined by the owner. In some cases, the code is just a specific application; in others, it might be a purpose-built micro OS and the application. The CPU includes the TEE and the Rich Execution Environment (REE), allowing for decisions on where applications and data should be processed according to protection needs. The REE executes non-sensitive data, whereas the TEE can be programmed to execute encryption functions or the processing of sensitive applications for instance.

Having a TEE can protect from the following threats:

- **Malicious/compromised admin CSP**: A bad actor at the Cloud Service Provider (CSP) cannot access the TEE memory even with physical access to the server.
- **Malicious/compromised tenant of a hypervisor:** A rogue app or compromised component of the system cannot access the TEE memory, even with a privilege escalation on the virtual machine manager (VMM).
- **Malicious/compromised network**: A rogue app or actor using a compromised network cannot access the data/IP inside the TEE.
- **Compromised firmware/BIOS:** Tampered BIOS or firmware will not be able to access the TEE memory.
- **Physical access attacks at the edge:** A bad actor targeting the edge compute nodes cannot access the TEE memory even with physical access to the system.
- **Malicious/compromised admin at the edge**: A bad actor at the edge provider cannot access the TEE memory even with physical access to the system.

5G Network Functions (NF) can leverage TEEs in one of two ways. In the per-process TEE mode, the NF can be factored into untrusted and trusted components with the latter running in containers in the TEE. An alternate model would have the entire NF running on containers in a TEE without the need to refactor.

RUNTIME SECURITY

Runtime security provides active protection to detect and prevent malicious activity during container execution. Secure computing (seccomp[7]) can prevent a containerized application from making certain system calls (syscalls) to the underlying system's kernel. While the Linux operating system has a few hundred system calls, many of them are not necessary for running containers. Restricting permitted syscalls to an allow-list decreases the application's attack surface.

An alternative to using seccomp is adding/dropping Linux capabilities. Capabilities involve various checks in kernel functions reachable by syscalls. The check can be done either at the beginning of a specific syscall, or target deeper in the kernel in areas that might be reachable through multiple different syscalls. Commercial alternatives to seccomp are also available. Several have moved beyond static profiles and have begun using machine learning to block or alert on suspicious activity.

[7] Secure computing mode is a Linux kernel feature, which restricts Operating System calls that a container has access to.

USE GOOD CONTAINER SECURITY HYGIENE TO AVOID RESOURCE CONTENTION & DOS

A Pod without limits can theoretically consume all of the resources available on a host. As additional Pods are scheduled onto a node, the node may experience CPU or memory pressure, which can cause the kubelet to terminate or evict Pods from the node. *Limits* are the maximum amount of CPU and memory resources that a container is allowed to consume. If a container exceeds its CPU limit, it will be throttled.

A resource quota specifies the total amount of resources, such as CPU or RAM, allocated to a namespace. When applied to a namespace, the quota bounds available resources for all containers deployed into that namespace. By contrast, *limit ranges* provide more granular control of the allocation of resources. *Limit ranges* are applied per Pod or per container within a namespace and can also set default limit values if none are provided.

Audience: Mobile Network Operators, Customers

Guidance/Mitigations

- Use the PodSpec to set limits to help minimize resource contention and mitigate risk arising from poorly written or compromised applications that consume an excessive number of resources.
- Setting a *resource quota* or creating a *limit range* can force the use of limits on a namespace.

IMPLEMENT REAL-TIME THREAT DETECTION AND INCIDENT RESPONSE

Given the current threat landscape, incident detection and incident response in 5G cloud infrastructure supporting multi-tenancy must be real-time. In order to achieve real-time detection and response, the attack surface should be reduced following the principals described in this document, including enabling isolation and logging for immediate response of unexpected behaviors. Anomalous behaviors can be detected and prevented in real-time if infrastructure is configured to permit allowed behaviors, with dynamic authentication and verification of people, components, and devices. 5G cloud infrastructure supporting an allow-list approach aligned to zero-trust architectural tenets will improve overall security posture and reduce resource demands, as this follows a built-in security approach.

Audience: Cloud Providers, Mobile Network Operators

Guidance/Mitigations

REAL-TIME ATTESTATION OF SETTINGS AND METRICS

At boot and runtime, attestation technology can verify configuration policy and container metrics (e.g., hash of files, time to execute a module). Attestation technology takes a snapshot of current configurations or metrics, digitally signing that evidence. The evidence is verified against a protected set of expected policies and measurements. The Trusted Platform Module

(TPM) or equivalent platform (or infrastructure) may serve as the Root of Trust or a hand-off to a TEE. Attestations can be performed against configuration settings individually or in groups. This method of attestation enables an allow-list approach centered on expected behaviors rather than a deny-list approach that leaves gaps.

The Security Content Automation Protocol (SCAP)[8] provides a full set of standards to verify and remediate configurations and may compliment attestations. Attestation is in use for some aspects of verifying container security and will expand in time, ideally becoming part of SCAP in the future. Built-in security that is verified, with alerting or remediation on exceptions, is the only way to achieve real-time detection to enable real-time incident response.

INCIDENT RESPONSE

The ability to react quickly to an incident can help minimize damage caused from a breach. Having a reliable alerting system that warns of suspicious behavior is the first step in a good incident response plan. When an incident does arise, identify and isolate the offending Pod for forensic investigation and root cause analysis. Response should minimally include:

Isolate the Pod with a network policy that denies all ingress and egress traffic to the Pod

A deny all traffic rule may help stop an attack that is already underway by severing all connections to the Pod. However, a network policy may prove ineffective if an attacker has gained access to underlying host. In this case, stateful firewall rules can isolate a compromised host from other hosts.

Cordon the worker node

By cordoning the impacted worker node, Kubernetes is informing the scheduler to avoid scheduling Pods on the affected node. This allows removal of the node for forensic study without disrupting other workloads.

Enable termination protection on impacted worker node

An attacker may attempt to erase their misdeeds by terminating an affected node. Some cloud providers offer a feature to prevent nodes from terminating. Additionally, instance scale-in protection will protect the node from a scale-in event which would otherwise cause the worker node to terminate.

Capture volatile artifacts on the worker node

Capture the operating system memory, running processes, open ports, and other ephemeral artifacts that would be lost during a system reboot. Use container engine commands to extract additional evidence from the container including: running processes, daemon level held logs,

[8] NIST Security Content Automation Protocol, 2021.
https://csrc.nist.gov/projects/security-content-automation-protocol/

open ports, and changes to files and directories since its initial launch. Pause the container for forensic capture and snapshot the node's persistent storage volumes.

Practice security game days

Organizations can use security game days to test and practice incident response plans. Divide security practitioners into two teams: red and blue. The red team focuses on probing different systems for vulnerabilities while the blue team attempts to detect and block the red team's attacks. Additionally, consider hiring an outside entity with experience in Kubernetes penetration testing.

CONCLUSION

Securely isolating network resources is only one aspect of hardening a 5G cloud infrastructure. Preventing a process that runs in a container from escaping the isolation boundaries of its container and gaining access to the underlying host is threat that must be addressed. Capabilities that enable the detection of unexpected behavior, such as dynamic verification through attestation or use of behavior profiles, need to be industry best practices.

In *Part III: Protect Data in Transit, In-Use, and at Rest*, the following topics will be addressed:

- Protection of data in transit using secure protocols with a secure root of trust.
- Protection of data at rest with approved cryptography and proper key rotation.
- Protection of data in use with Trusted Execution Environments.