Introduction

The National Institute of Standards and Technology (NIST) has introduced its first publication to address the security challenges of containers. NIST SP 800-190 focuses on potential security concerns of containers and provides recommendations for addressing these concerns. This document focuses on the specific security concerns addressed by NIST and how the NeuVector Container Network Security solution addresses them.

Why NIST’s Focus on Container Security?

Container adoption is on the rise as organizations adopt microservices and convert monolithic applications to a container environment or build new applications in a cloud native fashion. Containers are ephemeral and immutable, driving three main challenges on which NIST focuses:

1. **Scale.** A typical container environment may have 100s or 1000s of containers. A container infrastructure is a dynamic environment so one minute there may be 10 active containers and the next minute there may be 1000 active containers. The container security solution must be able to scale accordingly. Also, most container communications are inter-container (east-west), so it is imperative that the container security solution embeds in the container infrastructure, protecting and providing visibility into this network traffic.

2. **Automation.** As NIST states “automation is not just important to deal with the net number of entities, but also with how frequently those entities change.” Automation is central to container management and is the basis for the continuous integration/continuous deployment (CI/CD) pipeline. The container security solution must fully integrate into the automated processes and applications (e.g., Jenkins for CI/CD orchestration, Git for image repository, and JFrog Artifactory for image scanning), for all phases from build to ship to run-time.

3. **Central policy management and enforcement.** The only way to enforce security policies in a rapidly scaling, automated infrastructure is via centralized policy expression and strict enforcement of software management policies. The container security solution must be policy aware and able to track policy violations and act accordingly: alert and enforce. As discussed below, this includes the ability to automatically quarantine containers that are violating policy.

---

Shifting Left and CI/CD

A primary goal of securing CI/CD is the concept of shifting left or shifting the responsibility for security through the entire CI/CD pipeline all the way from development to production, not just bolting it on at the end. NIST recognizes this and calls out the specific need for container security throughout the entire software development lifecycle. As NIST states, “organizations that are successful at this transition gain security benefit in being able to respond to vulnerabilities faster and with less operational burden than ever before.” However, NIST recognizes that doing so requires security teams are empowered to actively enforce security and quality throughout the full software development lifecycle. As discussed below, the NeuVector Container Network Security solution secures the entire software development lifecycle through use of intent-based intelligence that understands application intent based on meta-data and continuous behavioral analysis. Some of the benefits include:

- Automated protection that is declarative, discovering application behavior in real time
- Whitelist-based rules establishing a zero-trust model to only allow accepted container and network behavior
- Layer-7 firewall policy-based control that does not rely on underlying Linux services such as ITables, seccomp, and SELinux, or service mesh extensions such as ISTIO.
- Full life-cycle vulnerability and compliance management from build to ship to run.

NeuVector Support of NIST Countermeasure Recommendations

The following section describes how NeuVector container network security solution addresses specific countermeasures (listed in section 4.0) against the container challenges that NIST discusses in Section 3.0.

<table>
<thead>
<tr>
<th>NIST SECTION</th>
<th>NEUVeCTOR RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Image Countermeasures</td>
<td>NeuVector provides registry scanning and a Jenkins plug-in for build-time scanning. NeuVector also automatically scans running containers and hosts for both common vulnerabilities (CVE) and application specific vulnerabilities.</td>
</tr>
</tbody>
</table>

Ibid.
This approach protects images via policy-driven enforcement as they move through the continuous integration/continuous deployment (CI/CD) pipeline. The end-result is preventing vulnerable images from moving into production.

<table>
<thead>
<tr>
<th>4.1.2</th>
<th>Image configuration defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeuVector automatically enforces the CIS Docker and Kubernetes benchmarks to set best practices for building images, including validating that the base image is from a trusted source. Further, NeuVector identifies all processes running in containers, creating a baseline with allowed processes whitelisted. NeuVector can assist in the audit process by alerting on unnecessary functions. For example, validating that SSH is disabled along with other best practices. NeuVector continuously scans images during build, test, and run phases of the CI/D, identifying image configuration defects and the fix versions required to address these defects. NeuVector automatically rescans updated images, containers and hosts to verify the application of patches.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.1.3</th>
<th>Embedded malware</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeuVector uses deep packet inspection (DPI) to monitor for behavioral anomalies continuously; potentially indicative of malware. NeuVector will alert on any unauthorized communication channels or process used by malware. This virtual patching function is a feature of the NeuVector container firewall. Suspicious file system activity in containers and hosts also trigger alerts, including downloads of executables and modifications to any packages/libraries or sensitive directories.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.1.4</th>
<th>Embedded clear text secrets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded cleartext secrets is a dangerous practice that requires coding discipline. NeuVector recommends that all clients adopt key management and secrets management features and capabilities provided by NeuVector orchestration platform partners, including Kubernetes, Docker EE, Red Hat OpenShift, etc. NeuVector integrates with these partners for secrets management.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.1.5</th>
<th>Use of untrusted images</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeuVector implements trusted image management capabilities by integrating with orchestration platforms such as Kubernetes. NeuVector offers admission control that dynamically prevents untrusted images deploying into production. Operators may establish image signature verification policies based on vulnerabilities, users, namespaces, and other criteria. NeuVector can also detect when new components are added to the system and automatically rescan changed containers or scan new containers.</td>
<td></td>
</tr>
</tbody>
</table>
To support dynamic and static image and code testing, NeuVector partners with and integrates with code scanning tools such as Black Duck software and JFrog Artifactory.

### 4.2 Registry Countermeasures

#### 4.2.1 Insecure connections to registries

NeuVector helps identify non-encrypted network connections and can restrict communications to encrypted registry connections only. NeuVector can also detect encrypted connections and automatically whitelist required SSL/TLS connections.

#### 4.2.2 Stale images in registries

NeuVector focuses on vulnerabilities in image registries independent of image age, detecting all vulnerabilities in recent and older images. The admission control feature can also be used to prevent stale images from being deployed.

#### 4.2.3 Insufficient authentication and authorization restrictions

NeuVector integrates with Kubernetes and OpenShift RBACs as well as LDAP/AD. NeuVector enforces context-based authentication, so an image is never pushed to the registry before vulnerability scanning is complete.

NeuVector admission control can restrict the deployment of images from registries based on special labels and tags (and other criteria), preventing image deployment if they lack authentication/authorization settings.

NeuVector continuously discovers and monitors all containers, implementing and enforcing stateful ingress and egress network control to only established network connections. Any new connections into the network automatically trigger alerts and – if configured – automatic blocking.

### 4.3 Orchestator Countermeasures

#### 4.3.1 Unbounded administrative access

NeuVector continuously discovers and monitors all containers and orchestrator services (e.g., Kubernetes, Docker Swarm, OpenShift), implementing and enforcing stateful ingress and egress network control to only established and trusted network connections, protocols and ports. Any new connections into the network automatically trigger alerts and – if configured – automatic blocking for application workloads.

#### 4.3.2 Unauthorized access

NeuVector integrates with Kubernetes and OpenShift RBACs as well as LDAP/AD to enforce security policy access controls and prevent unauthorized access.

#### 4.3.3 Poorly separated inter-container network traffic

NeuVector provides automated container segmentation based on container communication whitelists. Connection authorization is based on Layer 7 deep packet inspection to validate the connections, with dozens of application protocols enforced and multi-protocol detection and inspection including TCP, ICMP, and UDP.

NeuVector supports policy separation and import/export between development, test and production environments. If on the same network
segments, the NeuVector Enforcer container will enforce separation of development/test from production environment containers by establishing container groups. Group designations include images, nodes, instance names, services, labels, or addresses.

Through this policy enforcement, NeuVector can isolate containers based on sensitivity level, so all applications with the same sensitivity are on the same virtual network.

| 4.3.4 | Mixing of workload sensitivity levels | NeuVector provides automated container segmentation based on container communication whitelists, allowing only authorized connections. Workloads of differing sensitivity levels will never be allowed to make unauthorized connections to other workloads based on the zero-trust whitelist policy. For example, PCI sensitive workloads (in-scope workloads) may be segmented from all other workloads regardless of network segment or host.

NeuVector automatically identifies services that are running in containers and automatically groups services based on manifests and runtime data. Operators may configure NeuVector in either Detect, Monitor, or Protect mode, based on the type of applications/services running. This approach provides fine-grained control over what services, protocols, daemons, etc. are running in each container.

| 4.3.5 | Orchestrator node trust | NeuVector automatically runs CIS Docker and Kubernetes benchmarks to detect violations of orchestrator configuration best practice. NeuVector also scans the host for vulnerabilities and conducts process monitoring to identify privilege escalations, suspicious processes, and breakouts. Operators may configure automated response rules to trigger alerts based on CVE levels or particular CVEs. These triggers can result in container quarantine. NeuVector can quickly isolate a compromised node for a Kubernetes cluster via container quarantine.

NeuVector can monitor connections between Kubernetes cluster members and automatically detect encrypted connections and automatically whitelist required SSL/TLS connections. NeuVector can block any connection not encrypted and automatically trigger an alert.

| 4.4 Container Countermeasures | NeuVector automatically scans running containers and hosts for both common vulnerabilities (CVE) and application specific vulnerabilities. High-risk vulnerabilities are flagged based on the CVSS score, triggering alerts. These triggers can result in automatic container quarantine, packet capture, and or blocking container deployment. NeuVector integrates with CI/CD tools (e.g., Jenkins and JIRA) to automatically trigger issue tickets. |
| 4.4.2 | Unbounded network access from containers | NeuVector provides automatic discovery of flows (container networking surfaces) between containers and services. NeuVector provides a Layer 7 multi-vector firewall, discovering, monitoring, and protecting via micro-segmentation of all containers, including at each Internet connection, and between DMZ and internal zones. NeuVector’s automatic flow discovery includes the monitoring and detection of encrypted and encapsulated (virtual networks) communications.

NeuVector continuously monitors for behavioral anomalies, potentially indicative of malware. NeuVector will alert on any unauthorized communication channels or process used by malware. This virtual patching function is a feature of the NeuVector container firewall.

NeuVector automatically establishes whitelist rules to enforce and restrict inbound/outbound traffic to each network segment. Operators may add custom whitelist and blacklist rules. |
| 4.4.3 | Insecure container runtime configurations | NeuVector automatically runs Docker Bench security reports and Kubernetes CIS Benchmarks to achieve configuration standards enforcement.

NeuVector provides a Layer 7 multi-vector firewall, discovering, monitoring, and protecting via micro-segmentation of all containers, including at each Internet connection, and between DMZ and internal zones. Micro-segmentation prevents any insecure configurations from being exploited and breaking out of the container. |
| 4.4.4 | App vulnerabilities | NeuVector identifies all processes running in containers, creating a baseline allowed-process whitelist. Based on manifests and runtime data, workloads may be automatically grouped into services. NeuVector can assist in the audit process by alerting on unnecessary processes.

As part of running the CIS Docker benchmark, Operators will identify all containers not running the root filesystem in read-only mode.

NeuVector tracks each container and all ingress/egress from the container.

NeuVector can automatically flag containers that are running multiple services.

NeuVector DPI tracks all container protocols and services and any new service or protocol may be indicative – and flagged – as potentially malicious activity.

This approach provides fine-grained control over what services, protocols, daemons, etc. are running in each container.

Specific events that NeuVector can track, include:
| 4.4.5 | Rogue containers | NeuVector automatically discovers every system component including rogue containers using Layer7 inspection and DPI with automatic whitelisting of the nominal container environment. All network connections from/to rogue containers are by default identified as a violation of the whitelist rules. Administrators receive real-time updates via an advanced GUI dashboard and via Syslog/webhooks.

Upon detection of a rogue container, NeuVector can also automatically quarantine the container and trigger an issue ticket via integration with CI/CD tools (e.g., SIEM, Jenkins, and Jira).

4.5 Host OS Countermeasures

| 4.5.1 | Large attack surface | NeuVector supports minimized container-specific OS, including CoreOS and Google’s Container-Optimized OS. Also, NeuVector automatically baselines all processes in running containers and presents this whitelist for review and modification. Extraneous processes may be identified, and NeuVector recommends the removal of all unnecessary libraries and packages. NeuVector can also quarantine containers which start unauthorized processes.

| 4.5.2 | Shared kernel | NeuVector provides full protection of all container workloads even when using a shared kernel. All security features continue to monitor and protect containers and the orchestrator in shared or non-shared kernel environments. Host vulnerability scanning and suspicious process monitoring also continue to in a shared kernel environment.

| 4.5.3 | Host OS component vulnerabilities | NeuVector monitors all container and hosts files systems in production to detect package/library updates, modifications of sensitive folders such as /etc., and downloads of any executables. These updates, changes, and downloads could be indicative of violating a container’s immutability and therefore trigger immediate alerts and action by the NeuVector firewall.

Along with this monitoring, NeuVector tracks all active containers and will automatically log the creation or deletion of system-level objects.

| 4.5.4 | Improper user access rights | NeuVector tracks all communications between containers and automatically logs all new communications for later event reconstruction. This communication includes any suspicious processes, including ssh sessions, commands via an interactive session, sudo commands, etc. |
### NeuVector Support of NIST Container Life Cycle Security

NIST makes it clear that continuous integration and continuous delivery are vital aspects of the container build, ship, and run lifecycle. To this end, Section 6.0 of NIST SP 800-190 discusses container technology lifecycle security considerations. It is imperative that any evaluation of container security solutions addresses both the NIST countermeasures of Section 4.0 AND the container lifecycle security considerations of Section 6.0.

NeuVector fully integrates with the CI/CD pipeline to provide security throughout the build, ship, and run phases of the software life cycle. As a host-based container firewall, NeuVector provides efficient local monitoring and protection in a fully scalable fashion that matches the dynamics of a typical container environment.

<table>
<thead>
<tr>
<th>NIST SECTION</th>
<th>NEUVVECTOR RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Initiation Phase</td>
<td>As defined by NIST, the initiation phase is the early stages of software development when stories are developed, driving product development backlogs. As NIST points out, &quot;traditional development practices, patching techniques, and system upgrade processes might not directly apply to a containerized environment.&quot; Implementing containers requires a whole new approach to instituting security policy. Organizations are now developing security policy during the initiation phase and implementing this security as code. NeuVector fully supports the automation of security policies via YAML files in Kubernetes. Through this, the security team can deploy security policies based on a security manifest. This approach is declarative (before workload deployment) and Kubernetes native.</td>
</tr>
<tr>
<td>6.2 Planning and Design Phase</td>
<td>During the planning and design phase, NIST focuses on building out forensic practices to track nominal and abnormal container behaviors. Because of the declarative nature of container image builds, NeuVector can immediately determine when a container is performing in a way inconsistent with its intended usage.</td>
</tr>
</tbody>
</table>

3 Ibid., iv.
Also, this is an appropriate time to integrate vulnerability management into the pipeline. NeuVector can scan images during the build process using tools such as its Jenkins plug-in, and scan/monitor images in registries for vulnerabilities.

NeuVector provides advanced behavioral anomaly detection as a DPI firewall. This capability immediately detects any difference between a container’s baseline operation (i.e., what the container planning says it should be doing) and its real-time activity (i.e., what it is doing during runtime).

An important point made by NIST is containers typically connect via virtualized overlay networks. These networks often use encapsulation and encryption. NeuVector automatically tracks all communications between containers: encapsulated and encrypted. NeuVector maintains an event log of all user activity and any actions performed to establish an audit trail. This functionality integrates with SIEM systems.

<table>
<thead>
<tr>
<th>6.3 Implementation Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST discusses the implementation phase regarding system prototyping and testing. In the CI/CD, this typically includes the staging and QA steps where a container environment is established to mimic the production environment. NIST calls out the point that “Implementation may require new security tools that focus specifically on containers and cloud-native apps and that have visibility into their operations: visibility that traditional tools lack.”⁴</td>
</tr>
</tbody>
</table>

Successfully securing the implementation phase requires a container firewall like NeuVector. NeuVector can be deployed in staging and QA to discover network and process behavior and automatically create the security rules for production. NeuVector can also isolate and protect containers (workloads) during the test (implementation) phase of the CI/CD. This process includes complete isolation of all ingress and egress as well as close monitoring of all container processes. Any new processes may trigger an alert as this could be indicative of a rogue container or malicious actions.

During system test, most organizations also invoke dynamic and static application testing, and NeuVector provides vulnerability scanning and partners with code scanning tools such as Black Duck software and JFrog Artifactory.

This phase is also where container deployment policies for admission control may be configured in NeuVector and Kubernetes to prevent unauthorized images from being deployed.

<table>
<thead>
<tr>
<th>6.4 Operations &amp; Maintenance Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>The operations and maintenance phase is the production environment for applications. Successfully securing the production environment requires a container security solution like NeuVector to limit inbound and outbound traffic based on Groups (Kubernetes services/deployments, DNS name, IP address/address range), protocols, and ports. NeuVector</td>
</tr>
</tbody>
</table>

---

⁴ Ibid., 34.
NeuVector can implement and enforce ingress and egress control, to and from the containerized workloads. NeuVector also runs automated tests for security compliance and prevents deployment of vulnerable images with admission control.

Specific tasks described by NIST for this phase, include:

- **Updating and distributing images** - NeuVector automatically tracks images and enforces image management by restricting images to an admission control policy for trusted images.
- **Vulnerability management** – NeuVector automatically scans container and hosts for vulnerabilities and guides fixing/removing the vulnerabilities. NeuVector enables auto-response rules to identify any un-remediated vulnerabilities found in production systems, issuing special alerts or quarantining containers.
- **Managing orchestrator and host updates** - NeuVector automatically runs CIS benchmark tests against Docker and Kubernetes to enforce configuration best practices.
- **Integration into SIEM** - NeuVector integrates with SIEM via Syslog and webhooks.

### 6.5 Disposition Phase

The declarative nature of containers and the automation of the CI/CD open the door to automated deployment and destruction of production containers. NIST discusses the disposition phase from the perspective of data retention including image destruction and revoking cryptographic keys.

By integrating into the container environment including hooks into CI/CD, NeuVector protects organizations through continuous DPI and layer 7 firewall monitoring and policy enforcement. Capabilities include monitoring when containers are removed from service and preventing retired containers from re-entering service without proper vulnerability scanning and CIS benchmark auditing.

### Conclusion

NIST is quite proactive in addressing container security, especially considering that many organizations are just at the beginning of their container and CI/CD initiatives. NIST recognizes that containers open the door to improving security posture, yet at the same time, containers significantly increase the potential attack surface and the risk of introducing new vulnerabilities into the infrastructure. Protecting against these increased risks requires focusing on the specific countermeasures to address container security challenges in the context of the greater container lifecycle perspective.

However, the NIST document places more emphasis on the pre-production (prevention) phases of container development than on the run-time container security requirements. The best preventive security will not wholly
eliminate damaging attacks from insiders and zero-day exploits. It is critical to implement strong container network security to detect and prevent the types of data breaches organizations have experienced over the last few years.

Strong container network security should include:

- A layer 7 [container firewall](#) with deep packet inspection (DPI) and multi-protocol application segmentation
- Built-in network attack detection such as DNS, DDoS, SQL injection, etc.
- Sensitive data detection in unencrypted network communication
- Auto-segmentation of workloads for PCI and non-PCI firewalling
- Monitoring of Kubernetes system containers network behavior
- Automated packet capture for debugging and forensics
- Real-time connection visualization and network attack display

As described in this document, NeuVector is uniquely positioned to address both the specific container security countermeasures and full security throughout the container lifecycle. With the only true layer 7 [container firewall](#) available today, NeuVector provides unprecedented cloud-native visibility and protection for container deployments in production.