



Guide

How to use NeuVector with the MITRE ATT&CK Framework

There are many attack vectors for cloud-native Kubernetes and container deployments,

some new and some traditional. To help organizations learn about these and protect against them, MITRE has published a knowledge base of techniques and tactics in a new matrix focused on containers.

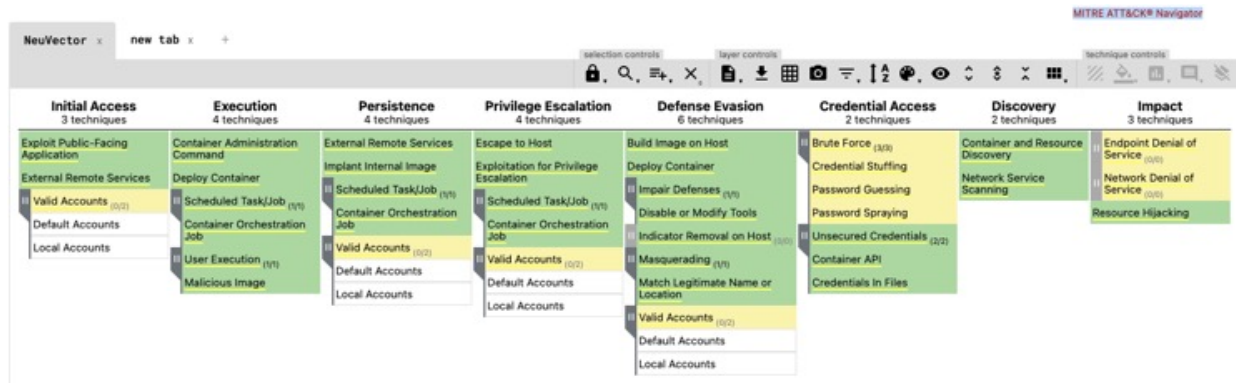
From the mitre.org website

MITRE ATT&CK® is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.

The newly published [ATT&CK for Containers](#) provides [container-specific attack vectors](#) across the lifecycle of containers, primarily focused on the run-time, or production environment (as opposed to the build CI/CD pipeline) and the host, orchestrator and container attack surface.

The attacks listed can be protected against by the NeuVector Container Security platform, as summarized in the picture below. For each attack, MITRE lists techniques, examples, and mitigation strategies. We have mapped the NeuVector mitigations to these attacks.

The green highlights are areas where NeuVector Security Controls directly address the attack; yellow is where there is partial coverage, and white is primarily where the attack vector is not in scope for a container security platform and should be mitigated by or-
chestrator, host or other controls.



The MITRE ATT&CK® Navigator is an interactive tool which can be found online [here](#) in order to examine each attack technique. To load the NeuVector protections into the MITRE ATT&CK Navigator, [click here](#). In the interactive version, mouseover an attack to see its reference and the NeuVector protections. Right click on any attack to see the full description, example attacks, and mitigations strategies.

Many of the techniques listed can be mitigated with proper configuration and RBACs for systems, orchestrators and other critical tools. In addition, image scanning to detect vulnerabilities and compliance violations which can be tied to admission control policies will help prevent known attacks.

But the most critical attacks on containers and their infrastructure will come from zero-day attacks where hackers use techniques to exploit unknown vulnerabilities, or gain access and expand the attack using techniques which can't be detected by traditional security tools.

Critical Run-Time Defense for Containers

Here are several run-time attacks which can be used to start, expand, or execute an attack.

- Deploy Container**
 This is the first gate to prevent vulnerable or exploitable deployments. NeuVector checks for file permissions (setuid/setgid) which are too broad, then in production monitors processes and file access to block unauthorized activity.

- **Exploit Public Facing Applications**
NeuVector enforces Layer7 network segmentation to prevent unauthorized connections ingress and egress into and from public facing containers. In addition, network attacks embedded within allowed protocols are detected. Common OWASP attacks such as SQL injection, XSS, XRF, malicious file uploads and others can be detected and blocked by NeuVector.
- **Privilege Escalations**
NeuVector detects privilege escalations in hosts and containers. This is critical to prevent container breakouts to the host.
- **Network Service Scanning**
NeuVector prevents service scanning by employing a zero-trust security model for network connections. Only allowed connections are possible as enforced by source application (pods), destination application (pods), and the Layer7 application protocol allowed. This built-in segmentation prevents the source of scanning as well as the target from being scanned.
- **Denial of Service / Resource Hijacking**
NeuVector includes built-in network DDoS attack prevention on container workloads. Attacks such as resource hijacking will undoubtedly create new network connections, process and file activity as malware such as crypto-mining software attempts to start.

Although the matrix is a good container-focused summary, the techniques in **ATT&CK for Containers** do not include some of the most critical run-time attacks that are possible on containers. For a look at these, we have to look at the other attack matrices for Linux and Networks.

Going Beyond ATT&CK for Containers to Deploy Defense in Depth

While reviewing container-focused attacks can provide a good starting point for securing container deployments, it does not represent all attack techniques that can target Kubernetes or container deployments. Attackers may use a combination of new and traditional network attacks, host (Linux) attacks or application attacks to accomplish their goals. Many of these are presented in other MITRE ATT&ACK matrices such as in the Linux, **Cloud**, and **Network** matrices, and there is a high degree of overlap between these matrices.

It is important to address as many of these as possible to provide the broadest detection of events in a typical “kill-chain” of an attack. Remember that attacks will use a series of actions to fully execute an attack, and there will be a variety of targets where suspicious activity can be detected. These usually include network, host (OS), orchestrator, application (container process and file activity), and data storage actions, with the network being the most critical for cloud attacks. The following sections provide additional techniques which must be considered when preparing a defensive and mitigation strategy for attacks on containers.

ATT&CK for LINUX

The following additional categories from the ATT&CK for LINUX matrix should be included in container defense strategies in order to protect against run-time attacks.

| Lateral Movement 6 techniques | Collection 12 techniques | Command and Control 16 techniques | Exfiltration 8 techniques |
|--|--|---------------------------------------|--|
| Exploitation of Remote Services | Archive Collected Data (3/3) | Application Layer Protocol (4/4) | Automated Exfiltration (0/0) |
| Internal Spearphishing | Archive via Custom Method | DNS | Data Transfer Size Limits |
| Lateral Tool Transfer | Archive via Library | File Transfer Protocols | Exfiltration Over Alternative Protocol (3/3) |
| Remote Service Session Hijacking (1/1) | Archive via Utility | Mail Protocols | Exfiltration Over Asymmetric Encrypted Non-C2 Protocol |
| SSH Hijacking | Audio Capture | Communication Through Removable Media | Exfiltration Over Symmetric Encrypted Non-C2 Protocol |
| Remote Services (2/2) | Automated Collection | Data Encoding (2/2) | Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol |
| SSH | Clipboard Data | Non-Standard Encoding | Exfiltration Over C2 Channel |
| VNC | Data from Information Repositories (0/0) | Standard Encoding | Exfiltration Over Other Network Medium (0/1) |
| Software Deployment Tools | Data from Local System | Data Obfuscation (3/3) | Exfiltration Over Bluetooth |
| | Data from Network Shared Drive | Junk Data | Exfiltration Over Physical Medium (1/1) |
| | Data from | | |

As can be seen above, NeuVector provides broad coverage for these critical run-time attacks on Linux systems. For a full interactive view of the NeuVector coverage, use [this link](#) to load into the navigator tool.

Lateral Movement

There are a number of techniques attackers can use to expand the attack and probe for the next step in the “kill-chain.” This category is related to the previous column ‘Discovery,’ and techniques often can be combined to probe for open ports (e.g. SSH -22) then exploit through the Remote Services technique.

- NeuVector effectively prevents lateral movement by employing a zero-trust security model which defines the allowed network, process and file activity for container workloads. Attempts at lateral movement can be prevented by network segmentation and allowed process rules. Even if a remote service into a container must be allowed, such as SSH, specific commands can be allow-listed and others (e.g. apt-get, ls...) blocked by default). In addition, the next step in a “kill-chain” would typically initiate a new network connection, process, or file access, all of which can be detected and blocked by NeuVector.

Collection

Many collection techniques access data sources or use man-in-the-middle (MiTM) techniques to collect sensitive information.

- NeuVector is able to monitor all file access to sensitive directories and files in containers and can block unauthorized access. If specific files or directories should only be accessed by allowed-application(s) these can be protected by NeuVector. In addition MiTM attacks are detected by NeuVector segmentation, deep packet inspection, and process monitoring. The recent [Kubernetes MiTM vulnerability](#) can compromise application containers, and can be [detected and blocked by NeuVector](#).

Command and Control

Connecting to a command and control server is one typical step in the ‘kill-chain’ and can be one of the most damaging if allowed. Once the connection is established, typically to a server outside the container cluster or network (ie., internet), additional attack tools, malware, and vulnerable software can be downloaded and installed. In addition, this connection can be used to steal sensitive information through the outbound connection.

- With a strong, unique Layer7 container firewall, NeuVector enforces ingress and **egress policies** to prevent external connections. Through a zero-trust network segmentation model, unauthorized connections will be blocked. There are many techniques in this category, and NeuVector protections against the most damaging ones to a container environment are highlighted below.
- Application layer protocol. Attacks using approved application layer protocols can be detected by NeuVector using strong segmentation rules between containers, supplemented by deep packet inspection to detect attacks within an approved connection such as SQL injection, DNS tunneling etc.
- Encrypted channel. Use of an encrypted channel is monitored by NeuVector and use of encryption allowed only where needed. In the case of service mesh pod-to-pod encryption, NeuVector is able to enforce application layer segmentation and deep packet inspection rules by inspecting traffic before the encryption occurs with the service mesh sidecar proxy.
- Protocol tunneling. Tunneling, including DNS, ICMP or other protocols, can be used to steal sensitive data or disguise communications. These are detected by NeuVector's built-in detections and can be supplemented by DLP rules to inspect the network payload for other tunneling violations.
- Proxy. Legitimate use of proxies must be declared in NeuVector's zero-trust security rules, and any attempt to funnel traffic through unauthorized proxies will be detected by segmentation rules.

Exfiltration

Stealing sensitive data or other assets from a container deployment will involve the use of the network to connect out. Various techniques, as listed in the matrix can be used to accomplish this attack step.

- NeuVector protects against these techniques by use of a zero-trust network segmentation model that blocks unauthorized connections. Strong **egress controls** with added DLP inspection of network payloads can prevent successful exfiltration. In addition, if attempts to "tunnel" data occur using alternative protocols such as DNS or ICMP, NeuVector employs deep packet inspection and process monitoring to detect these attempt.

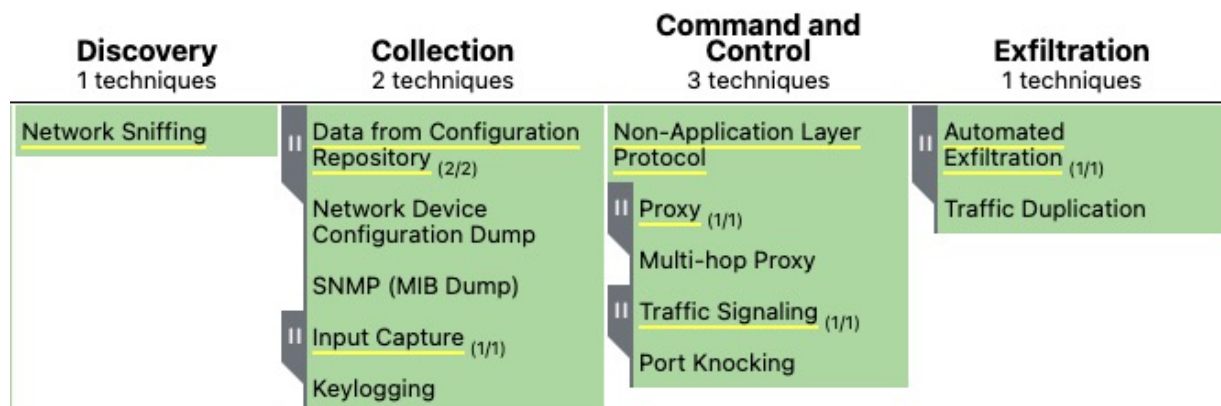
There are many additional techniques listed in this matrix which should be covered by standard linux security tools (SELinux etc), and public cloud vendors already provide hardened linux systems as their defaults.

ATT&CK for NETWORK

NeuVector provides broad, unique network protections for container-based environments. Following is NeuVector's coverage for ATT&CK for NETWORK:



The following four categories are especially important in planning defensive strategies for container environments.



Discovery

Network Sniffing either through passive monitoring or using Man In Middle (MiTM) attacks can enable attackers to capture valuable information to be used in the next step of the kill-chain.

- NeuVector ensures that only authorized workloads can be deployed through use of admission controls, and MiTM attacks such as the Kubernetes vulnerability are detected and prevented.

Collection

Data from Configuration Repositories can be stolen to understand sensitive configuration parameters.

- NeuVector employs a variety of network protections recommended including network segmentation, network filtering, and threat detection. In addition, CIS benchmarks and other compliance checks such as secrets detection help ensure that configuration information is not exposed.

Command and Control

There are several techniques which can be used to establish command and control, including use of a Non-Application Protocol, Proxy, or Traffic Signaling.

- NeuVector provides multi-protocol network security (including ICMP) and deep packet inspection to ensure all traffic is protected by segmentation rules and threat detections. Traffic signaling is detected through a combination of network segmentation (application protocols and ports) and process monitoring to ensure unauthorized processes such as reverse shells are blocked.

Exfiltration

In addition to the techniques discussed above, Automated Exfiltration can be difficult to detect. The use of network packet filtering and process/file monitoring is recommended.

- NeuVector inspects all network traffic and enforces segmentation rules to ensure that alternative channels are not used to exfiltrate data. Both process and file access monitoring will detect suspicious activity and block these attempts.

The NeuVector coverage for network attacks can be used interactively by importing [this link](#) into the navigator tool.

Expanding to the Enterprise

We've covered the most critical and damaging attacks to a container and Kubernetes environment. The full combination of ATT&CK matrices can be found in the comprehensive [ATT&CK for Enterprise](#) matrix, and this should be reviewed to make sure other channels or infrastructure that is part of the container environment are protected.

Employing a Zero-Trust Security Model to Prevent Container & Kubernetes Attacks

Enterprises are moving from a signature-based, deny-list approach to security to a zero-trust security model. In a zero-trust model, allowed behavior is declared, and everything else is deemed untrusted or suspicious. The move to a cloud, container and microservices based application model enables such a model to be possible.

NeuVector provides comprehensive security for container and Kubernetes deployments by automating security in the entire application lifecycle. The techniques used by attackers, as summarized in the Mitre ATT&CK matrices, can be detected and prevented before doing damage.

Summary of Attack Mitigations by NeuVector

- Zero-trust security model.
- Network segmentation (Layer7).
- Network threat prevention including OWASP.
- Process, File and Network security.
- Host protections.
- DLP deep packet inspection.
- Admission Controls.
- Egress Controls for Exfiltration prevention.
- Vulnerability and Compliance Image Scanning.

Contact NeuVector today for a [Free Security Assessment](#) and [Demo](#) of the NeuVector Container Security and Compliance Platform.

NeuVector, the leader in full lifecycle container security, empowers global organizations to comprehensively secure their container infrastructures without compromising business velocity. For security, DevOps, and infrastructure teams, the NeuVector continuous container security and compliance platform simplifies data protection from pipeline to production, enforces compliance, and provides unparalleled visibility and automated controls to combat known and unknown threats. To learn more about NeuVector, visit [NeuVector.com](https://neuvector.com).

Next Steps

[Want to learn more?](#)

NeuVector helps organizations manage all security issues related to container and Kubernetes deployments. To learn more visit neuvector.com.

SUSE Software Solutions
Germany GmbH
Frankenstraße 146
90461 Nürnberg
Germany
www.suse.com

For more information, contact SUSE at:
+1 800 796 3700 (U.S./Canada)
+49 (0)911-740 53-0 (Worldwide)

Innovate Everywhere

© 2023 SUSE LLC. All Rights Reserved. SUSE and the SUSE logo are registered trademarks of SUSE LLC in the United States and other countries. All third-party trademarks are the property of their respective owners.