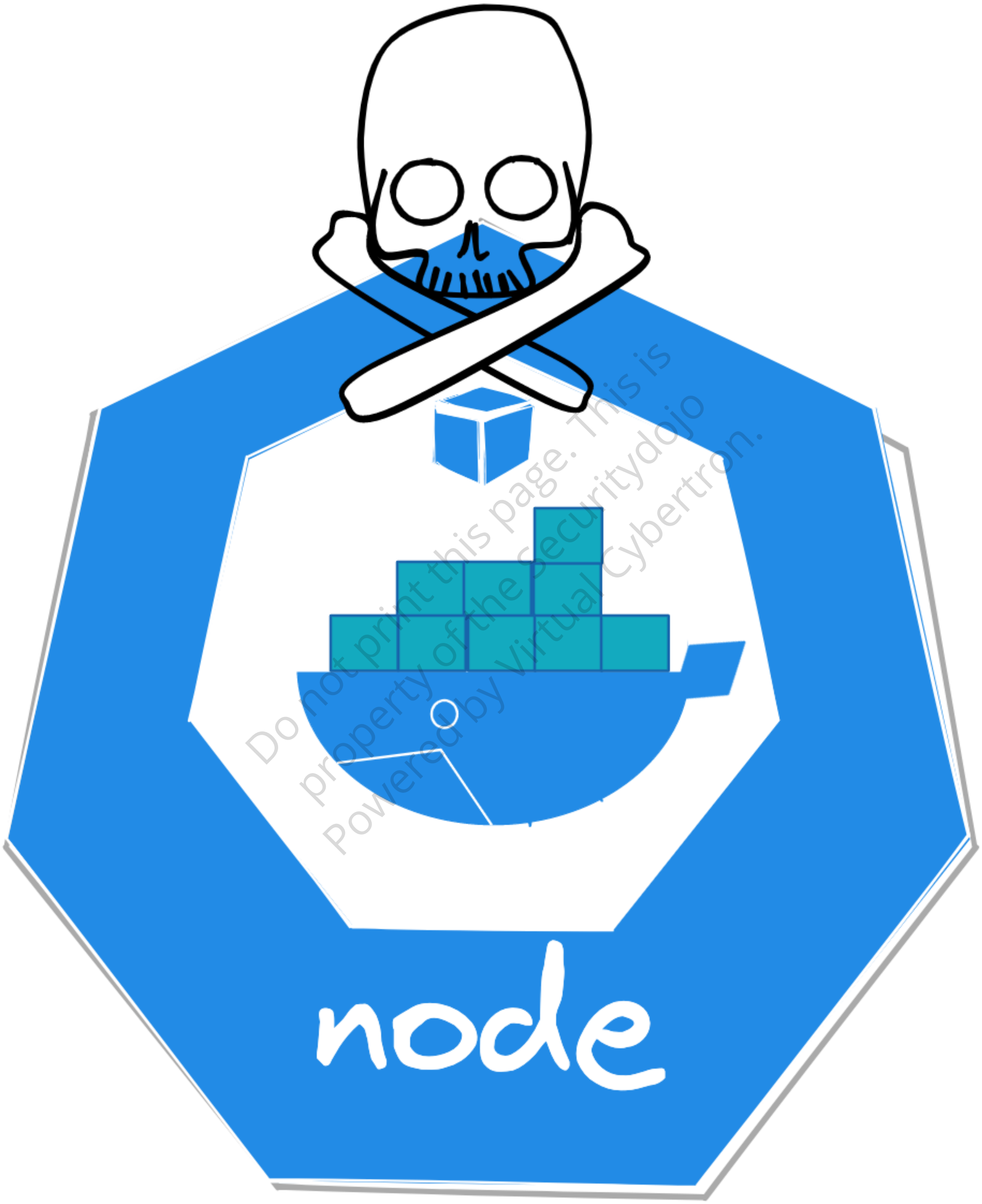
Post-exploitation: Container Breakout Techniques



Do not print this page. This is
property of the securitydojo
Powered by Virtual Cybertron.

Container Breakouts are scenario where a user, malicious or not, successfully bypasses container isolation to get access to host machine resources like the filesystem, processes, or network interfaces. The numerous setup errors and excessive rights that might cause container breakouts are covered in this section.

> It is presumed that the container's command shell is attacker controlled, this is post-exploitation.

Container Breakout Scenarios

- Host PID True: Exploiting shared host process space in containers.
- Host Network True: Gaining access to the host network from a container.
- Host IPC True: Abusing shared inter-process communication namespace in containers.
- Host Volume Mount: Escaping container isolation via host-mounted volumes.
- Privileged True: Breaking out of containers with privileged access.
- Docker Socket Mount:DIND: Exploiting Docker-in-Docker scenarios for container breakouts.
- RunC Vulnerability: CVE-2019-5736: Escaping container isolation through a known RunC vulnerability.
- Misconfigured Kube API Server: Exploiting a misconfigured Kubernetes API server.
- CVE-2021-25741: Demonstrating a container breakout using a specific CVE.
- Exploiting Process Injection: Escaping container isolation via process injection techniques.
- Unauthenticated Kubernetes Dashboard: Gaining unauthorized access to the Kubernetes dashboard.

> Note: The Container Breakout Labs featured in this course are developed by Bishop Fox. We would like to extend our gratitude and give full credit to their team for their excellent work.

Do not print this page. This is
property of the Securitydojo
Powered by Virtual Cybertron