Security considerations in container-based environments

Pinja Koskinen, Daniel Kouřil, Barbara Krašovec

What is a container?

- a container is just a running process
  - containers vs. full virtualization
- isolated from from other processes
  - Namespaces, cgroups
  - Isolation only on the level of kernel (which is “shared”)
- different containerization technologies available
  - Docker, LXD, Podman, Singularity, …
  - Rootless vs. requiring root-based daemon

Containers enable new services, e.g. CI

- Compromises can be devastating

SIGN IN

The Register

{* SECURITY *}

US Treasury, Dept of Commerce hacks linked to SolarWinds IT monitoring software supply-chain attack

Russia's Cozy Bear fingered, FireEye details injected backdoor and says it's worldwide

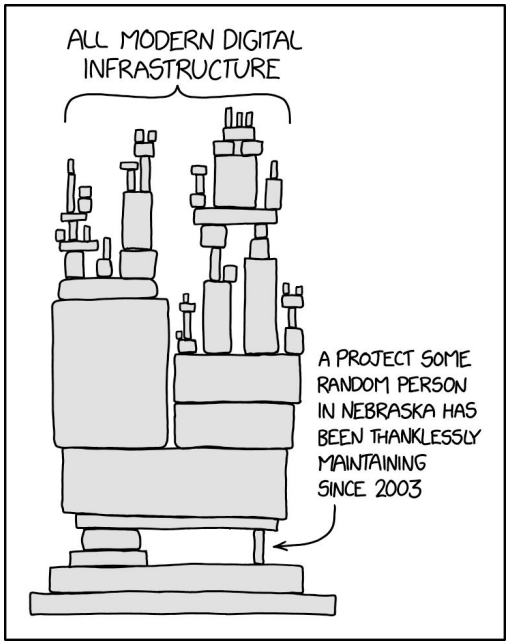
Simon Sharwood, APAC Editor

Mon 14 Dec 2020 // 02:10 UTC

79

UPDATED SolarWinds' Orion IT monitoring platform has been compromised, and speculation is swirling it was used as a base camp by state-backed hackers to infiltrate major US government organizations.

Current systems can be quite fragile



https://xkcd.com/2347/

How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript

Code pulled from NPM – which everyone was using

Chris Williams, Editor in Chief — Wed 23 Mar 2016 // 01:24 UTC

170

UPDATED Programmers were left staring at broken builds and failed installations on Tuesday after someone toppled the Jenga tower of JavaScript.

A couple of hours ago, Azer Koçulu unpublished more than 250 of his modules from NPM, which is a popular package manager used by JavaScript projects to install dependencies.

Koçulu yanked his source code because, we're told, one of the modules was called Kik and that apparently attracted the attention of lawyers representing the instant-messaging app of the same name.

According to Koçulu, Kik's briefs told him to rename the module, he refused, so the lawyers went to NPM's admins claiming brand infringement. When NPM took Kik away from the developer, he was furious and unpublished *all* of his NPM-managed modules. "This situation made me realize that NPM is someone's private land where corporate is more powerful than the people, and I do open source because Power To The People," Koçulu blogged.

Unfortunately, one of those dependencies was `left-pad`. The code is below. It pads out the lefthand-side of strings with zeroes or spaces. And thousands of projects including Node and Babel relied on it.

With left-pad removed from NPM, these applications and widely used bits of open-source infrastructure were unable to obtain the dependency, and thus fell over during development and deployment. Thousands, worldwide. Left-pad was fetched 2,486,696 times in just the last month, according to NPM. It was that popular.

https://www.theregister.com/2016/03/23/npm_left_pad_chaos/

Threat landscape

- Image management
  - images may be malicious, even official images may obsolete,
    - https://blog.aquasec.com/docker-official-images
    - https://unit42.paloaltonetworks.com/malicious-cryptojacking-images/
  - Checking authenticity and integrity
- Sensitive data embedded in images
  - Credentials, private data
- Vulnerability management
  - container/host levels - new places where obsolete package appears
  - kernel security important
- Containment of containers may be fragile
  - Escaping from containers - when the container is granted additional privileges (mount) or due to a SW vulnerability
  - Networking - access to L2, ...
- Host operation/configuration
  - Insecure configuration (or when implications aren't understood) can open severe vulnerabilities on the host (privilege escalation)
  - More complicated investigation/forensics (filesystem traces may be gone after container finishes, insufficient logs)

Security measures

- Secure configuration of the host
  - Containers share the host kernel
  - Patch management (incl. BIOS, kernel and device drivers)
    - root kernel attack: malicious container overwrites binary of runc and gains root access: see RunC vulnerabilities, also reported by SVG (runc is an interpreter for running containers)
    - Container tools: eg. CVE-2019-5736 vulnerability in runc
  - enable auditing, logging (remote)
  - turn on SELINUX or similar control mechanism
- Composing containers
  - container multi-tenancy: which users share the same host
  - use network separation of different groups of services (security groups)
- Secure containers
  - drop root privileges in containers, run services with lowest privileges possible
  - Consider enabling user-namespace

Security measures

- Secure build and maintenance of images
  - establish a secret build environment
  - best to build from scratch
  - control and automate image building using CI (incl. automated security checks)
  - don't put any sensitive data to the image - use external locations when starting
  - provide secure container registries (private/community)
  - Maintain images properly - scan images, check signatures
- Integrity and authenticity of images - Docker Content Trust (DCT)
  - Good idea, design not perfect
    - Decoupled from other PKIs
  - Not in use by common image producers
    - Docker Hub signs official images using its key => another trust anchor is necessary

Conclusions

- New technology brings new threats
  - New attack vectors (privilege escalation, escape from the contained environment)
  - Some aspects aren’t addressed in a secure way
  - Heavily based on the host kernel/OS
- Think through your threat model and assess relevant risks
- Make sure the the technology is understood
- Reach out to EGI CSIRT if you need discussion or want to contribute

Useful links

- Script to check security hardenings in Docker based environments: https://github.com/docker/docker-bench-security
  - Based on https://www.cisecurity.org/benchmark/docker/
  - CIS also provides hardening guidelines for operating systems
  - Similar script, based on CIS benchmark for K8s: https://github.com/aquasecurity/kube-bench
- Kubernetes hardening guideline: https://media.defense.gov/2021/Aug/03/2002820425/-1/-1/1/CTR_KUBERNETES%20HARDENING%20GUIDANCE.PDF
- Several container scanners: https://geekflare.com/container-security-scanners/
  - E.g. https://github.com/quay/clair
- Tool to create Docker seccomp profiles: https://github.com/antitree/syscall2seccomp