AVS Developer Security Best Practices

AVS Developer Security Best Practices

- Containers should be able to run with least privilege. Least privilege is AVS-dependent. AVS team should outline these privileges as part of the operator onboarding docs. In the case these privileges are not specified, it's recommended the operators ask the AVS team directly.
- Emit runtime (logs) including security events
- Use Minimal Base Images* Use [ko Go containers]
-
    - or similar to build distro-less minimal images. This reduces the attack surface significantly!
- Release updated images with security patches (for base OS etc).
- Do not store key material on the container (refer to key management docs).
- Your default user id should start with AVS-NAME-randomness to ensure there are no conflicts with the host.
- Ensure ECDSA keys utilized by AVS are solely for updates, such as modifying IP and port details within a smart contract. These keys should not hold funds. A role-based approach in smart contract design can address this issue effectively.
- AVS team should [sign their images]
- for any releases, including upgrades* If they publish to Docker, Docker will show the verified badge next to the image.
-
    - Tag new releases via updated images.
- Establish communication channels (Discord, TG) with operators. This ensures coordinating upgrades occurs with minimal friction.
- Operators should be in control of upgrades to their AVS software. Avoid software upgrade patterns where an agent checks for updated software and automatically upgrades the software.
- Release Notes should explain new features including breaking changes / new hardware requirements etc.

Suggested Key Management for AVSs

For key management, refer to the new location for the docs on [Key Security Considerations for Developers] . [Previous Key Security Considerations for Developers] [Next Multisig Governance]