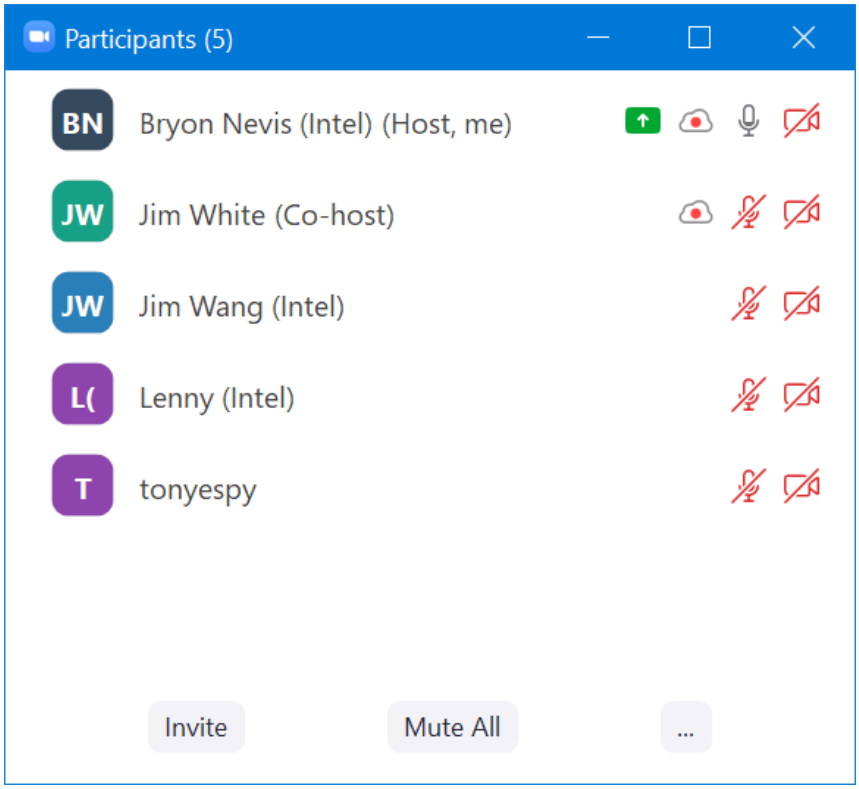
EdgeX Security WG Meeting

https://wiki.edgexfoundry.org/display/FA/Security+Working+Group

January 5, 2022

Attendees



Agenda

- Opens
    - None
- Microservice authentication
    - Problems to be solved:
        - Peer authentication of microservices—was it really core-command that asked me to do something, or was it really Kong that authenticated and forwarded the external request?
        - Inbound request authentication—authentication and authorization of remote callers
        - “Users”---e.g. do we need a “user” in the EdgeX UI?

    - https://github.com/charithmadhuranga (Slack: @nic huge) asking for support for keycloak or Hashicorp Boundary in order to support identity management at the edge. https://github.com/edgexfoundry/edgex-go/issues/3845 . Soliciting Nic to come and present at a future security WG meeting.
    - Microservice authentication ADR https://github.com/edgexfoundry/edgex-docs/pull/659. Proposal to extend delayed service start ADR based on SPIFFE/SPIRE to apply to all EdgeX microservices and simplify API gateway and allow service identity to be optionally federated in the cloud.
        - E.g. A docker-compose that runs the SPIRE server in the cloud. Default is to run it inside the edgeX instance. Would be a good idea to test remote SPIRE server.
        - This does not fully address the issue of distributed EdgeX
        - Increasing need to manage security centrally (e.g. Nic's suggestion above, SPIRE server in the cloud, etc) – broadening security to cover more than just a single EdgeX instance
        - Industry is moving to a zero-trust model, where each service must authenticate, not just authenticate at a boundary. Is this something that someone wants to do, if everything is running on the same box. May trust the box, but still need an API gateway for external calls. Any way to decouple that?
        - If in an enterprise environment with multiple instances, managing at scale vs a single box is a larger problem.
        - Can we live with mTLS based on spiffe/spire. Problem with mTLS is the API gateway – no API gateways that support spiffe/spire out of the box. Alternative is to expose every microservice on its own port, and no API gateway at all.
        - ADR is written for – how to keep a reverse proxy in place, yet still get the benefits of peer authentication with SPIFFE/SPIRE.
        - Request to uplevel the discussion about what the major decisions are. Might be a little early to raise to the broader community. Would be a great topic for F2F–"do we get rid of the API gateway?"
        - Plan: Pause new ADR. Implement current ADR. Mid-cycle - bring the big picture options. Modify ADR based on feedback. Big choices:
            - Do we need an API gateway?
            - Do we want to use TLS everywhere?
- Backlog otherwise unchanged

Standing Agenda

- Review Security Board
- Securing Consul Board (skip)

- Review CIS docker scan (will skip unless something changes) (click latest run, go to classic, view console output).
    - Last checked: Tue Nov 16 05:36:01 UTC 2021
- Review Snyk (Jenkins) (will skip unless something changes) (Imagelist)
    - Awaiting version bump: https://github.com/edgexfoundry/go-mod-core-contracts/pull/663
    - Alpine 3.14 findings - will wait for dot release since 3.14 is an LTS
- Review action items from previous week

Action Items

- 7/14: Bryon: Update security policy documentation w.r.t. when to use GitHub security advisories to notify users of issues.