Taking Your DevOps Tools To The Dark Side

Stepping Up Your DevOps Security Using OpenZiti

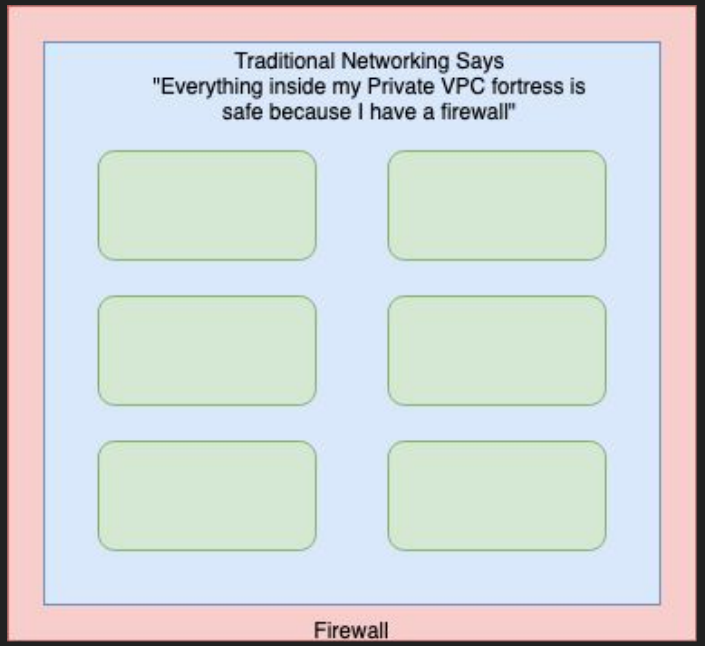
Mike Guthrie - NetFoundry

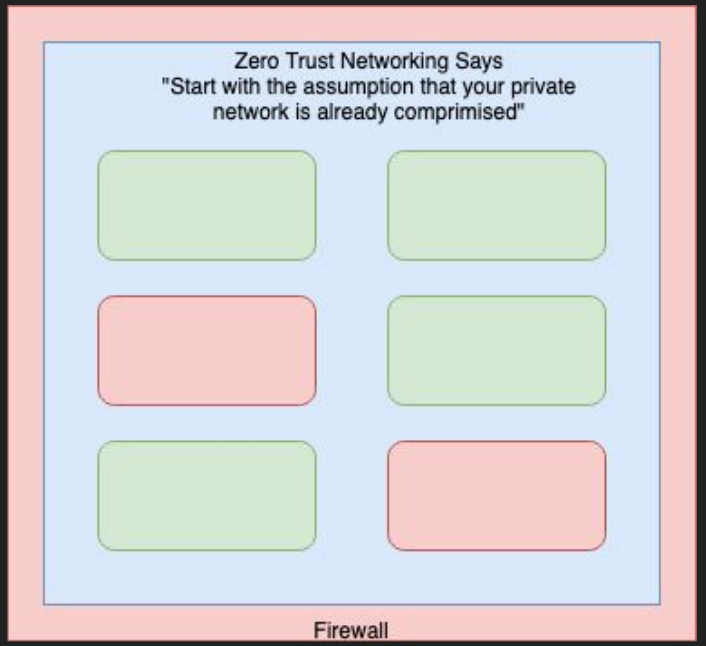
Background - Who Am I?

- Software Development / DevOps / SRE for 13 years
- Started in Software Development and Infrastructure Monitoring
- Managed CI/CD for over 100 developers
- Built a Site Reliability Department to support a NOC from the ground up
- Currently leading the DevOps (RAV) Team at NetFoundry
- Used a lot of DevOps tools, highly opinionated about which ones I like

About Zero Trust Networking…

Lots of definitions out there





The DevOps Problem

Every tool and system we use is an absolute goldmine for an attacker

- CI/CD - An automated code injection system that builds and executes code in every critical place in the environment
- Monitoring - A data mining platform that creates inventory, collects data, and can remotely execute code on everything important within your ecosystem
- ETL - a collection of loosely hacked together script and jobs to data mine every business critical data source within your ecosystem
- Data Warehouse - The one-stop shop for all of your critical business data, conveniently collected and lumped together into one place
- Config Management - A convenient place to take down or invade the entire infrastructure from one place
- Developer Access Management - Giving all the devs access to all the things in prod so that they can respond to outages instead of you. What could possibly go wrong?

The DevOps Problem

How do we deal with security?

- We pass our audits by applying the magical word “scope”
- We harden the front door of our APIs and public because that’s where we expect an attack
  - Passes compliance checks
  - Survives the audit
  - Survives the pen tester that w brought it

- Real World Gut Check
  - Would you turn an auditor loose on your CI/CD setup?
  - How many of your tools have open ports that are exposed to the internet?
  - How many of your tools have open ports exposed to the rest of your private network space?
  - Are you comfortable with how your developers and ops teams have access to systems…can you answer “what are they accessing” and “when” ?

A Paradigm Shift Towards Zero Trust - “Make It Dark!”

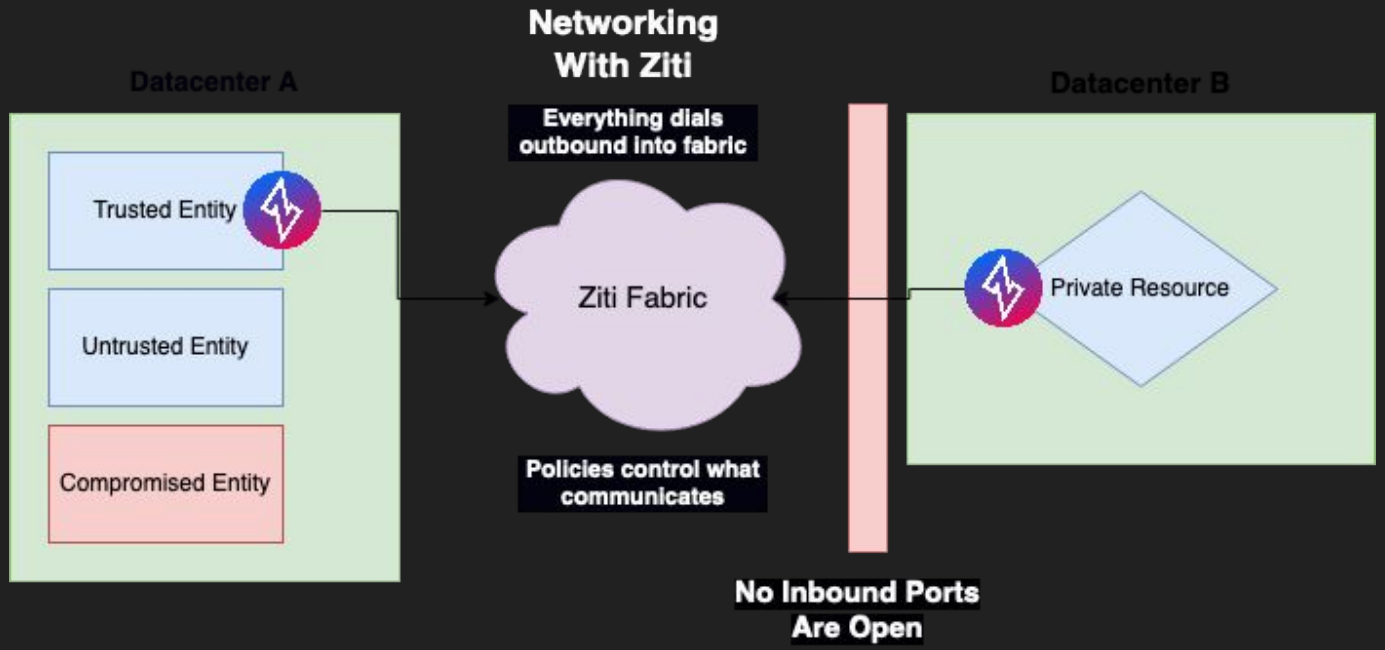
Context: OpenZiti - see https://openziti.github.io

- Stop leaving ports open to the entire Internet/VPC/Subnet/Datacenter
- Stop using VPNs that expose the entire private network
- Stop Peering VPCs and Datacenters
- “Making it dark” means NO INGRESS

KEY Concept

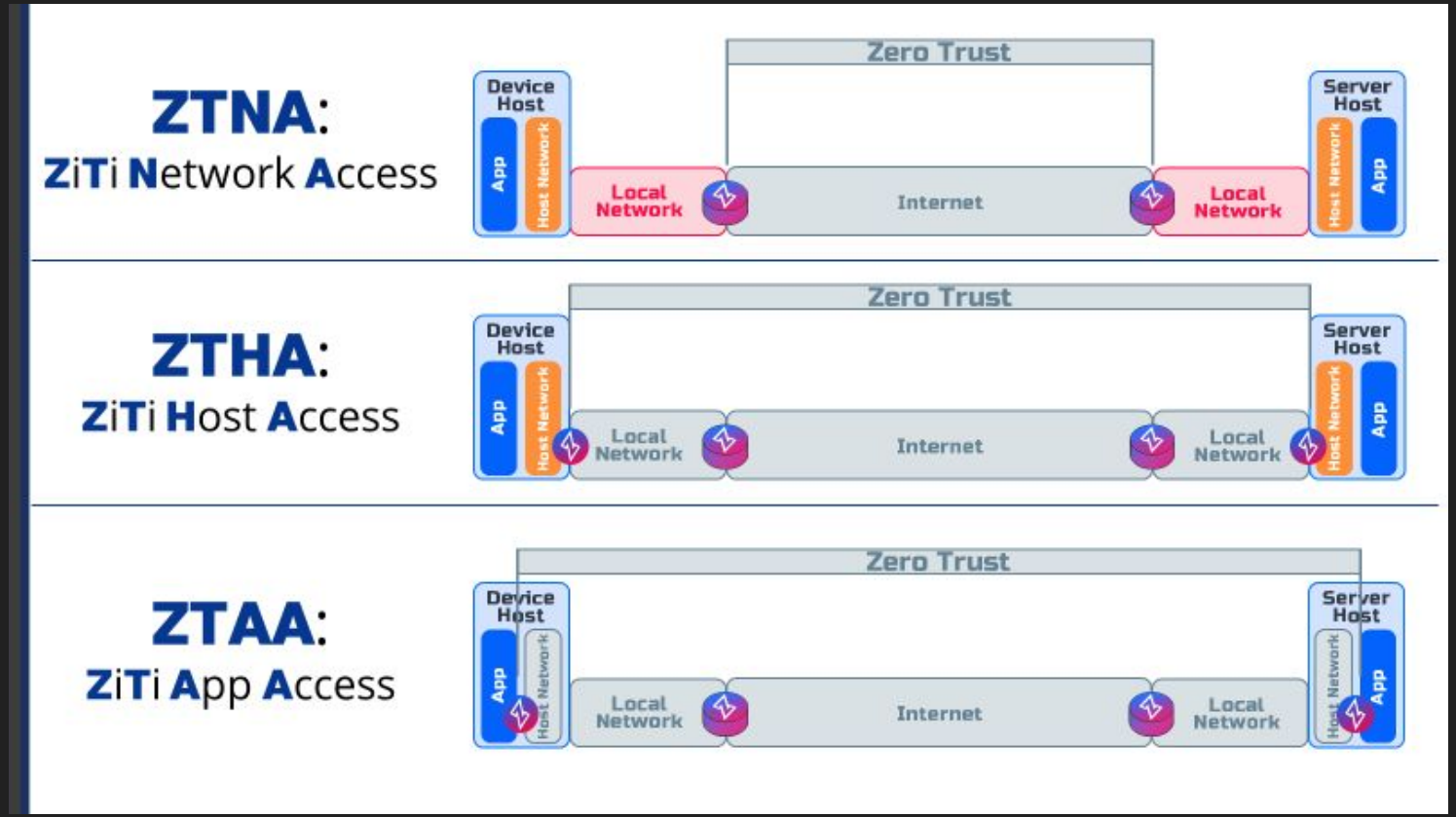
- There are no more IPs and Ports - only **Services** and **Identities**
- Access is managed through policies or “AppWans”

So How Do Things Actually Communicate?



Service Policy (AppWan) Example:
Identities Tagged with #datawarehouse
can talk to
Services tagged with #datawarehouse

Just How Dark? - 3 Typical Models



Just How Dark?

- Good
  - Install tunnelers in two data centers, no ingress and no peering between DCs
  - Allow ingress only from the tunneler inside the DC

Just How Dark?

- Better
  - Install tunnelers on the nodes on in a sidecar container
  - No ingress to the nodes
  - All services terminate on localhost addresses

Just How Dark?

- Best
  - Full application-embedded using the Ziti SDK
  - True zero trust and end-to-end encryption
  - No ingress *anywhere*

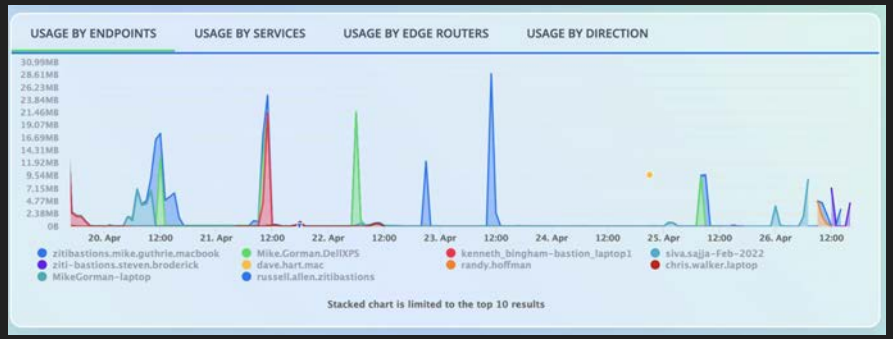
Internal Use Cases at NetFoundry

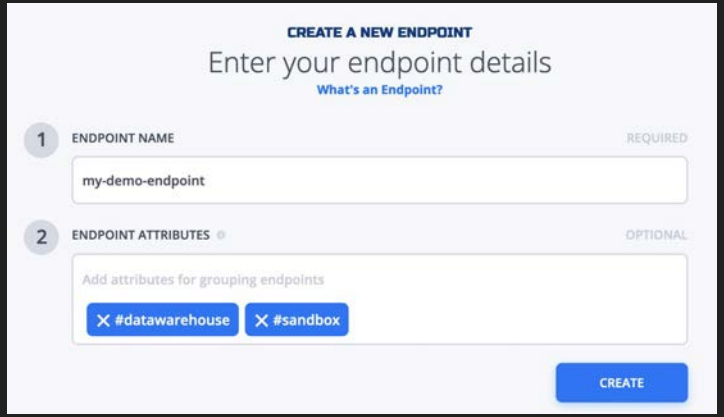
NetFoundry = Cloud Hosted OpenZiti as a SaaS Offering

- Data Warehouse
  - ETL runners all using Ziti to fetch data throughout ecosystem
  - End user access all managed through Ziti
- CI/CD System
  - No more ingress - only accessible through Ziti
- SSH Access
  - “Dark Bastions” - A Bastion with no open ports
- Mattermost
  - Slack replacement - company chat only accessible with Ziti
- Grafana
  - Was spun up “dark” on day 1 with minimal effort
- Developer Access
  - Running Ziti in sidecar containers to grant support access

Reactions: Moving to Zero Trust with OpenZiti

- Fully expected massive amounts of breakage and troubleshooting because we were “locking things down”
- Migration was substantially easier than expected because we could pre-validate everyone’s setup before “going dark”
- Access management is centralized and simple
- Access grants are granular and allow least privilege at the network level (SUPER helpful for managing developer and support access)
- By the 3rd migration, we were migrating users without even telling them because the change was so transparent
- No longer “punching holes in firewalls”





Where To Start?

- OpenZiti On Github
  - https://openziti.github.io
- OpenZiti Blog
  - https://openziti.io
- Cloud Ziti Teams Plan (FREE up to 10 endpoints):
  - https://nfconsole.io/signup

Questions?

mike.guthrie@netfoundry.io
https://www.linkedin.com/in/mike-guthrie/