



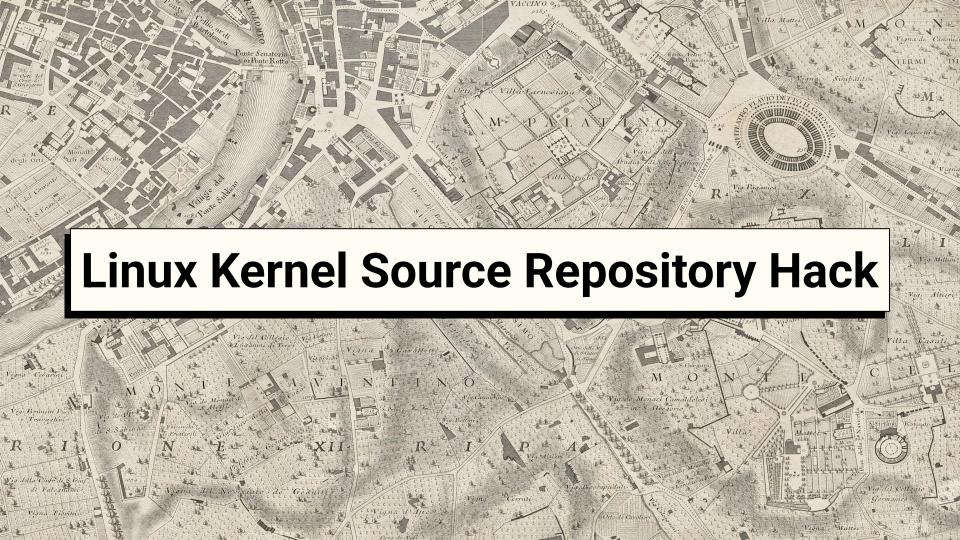


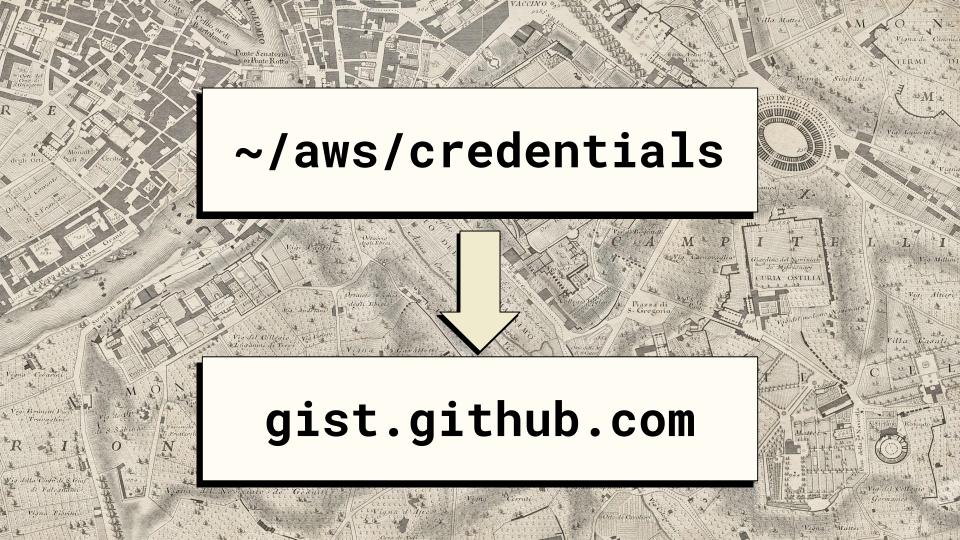
Supply Chain Attack or Not





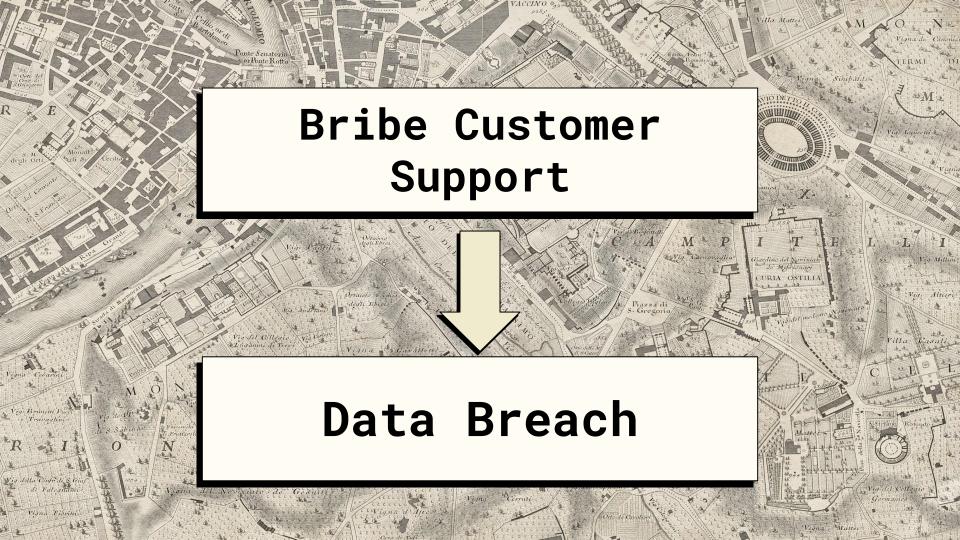


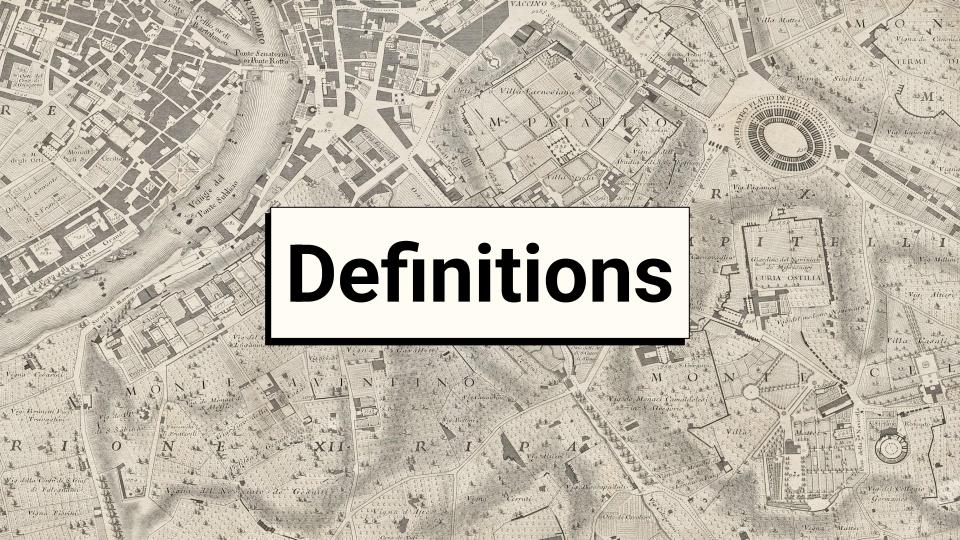












Software Supply Chain Attack

Insertion of nefarious code into trusted software before delivery.

Russ Cox. 2025.

Fifty Years of Open Source Software Supply Chain Security

Software Supply Chain Vulnerability

An exploitable weakness in trusted software caused by a third-party, component of that software.

Russ Cox. 2025.

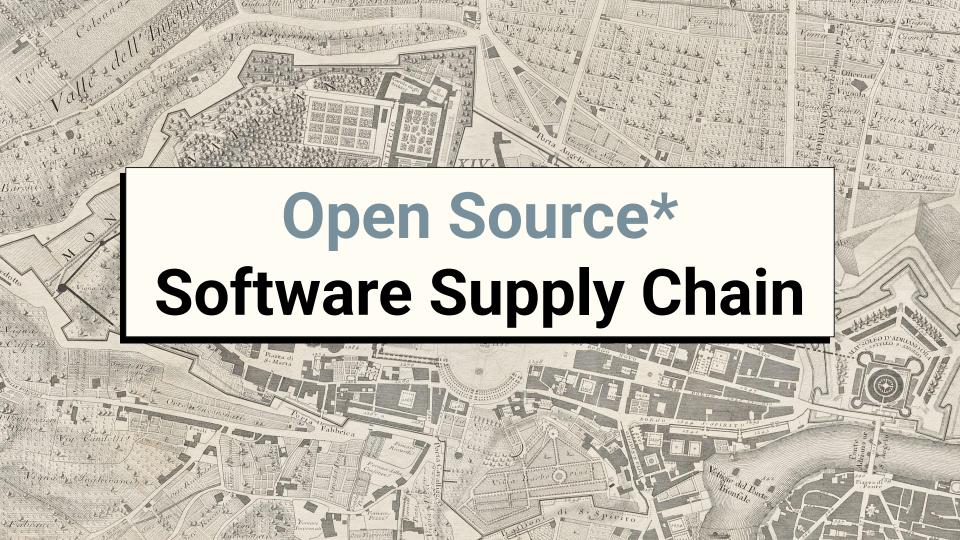
Fifty Years of Open Source Software Supply Chain Security

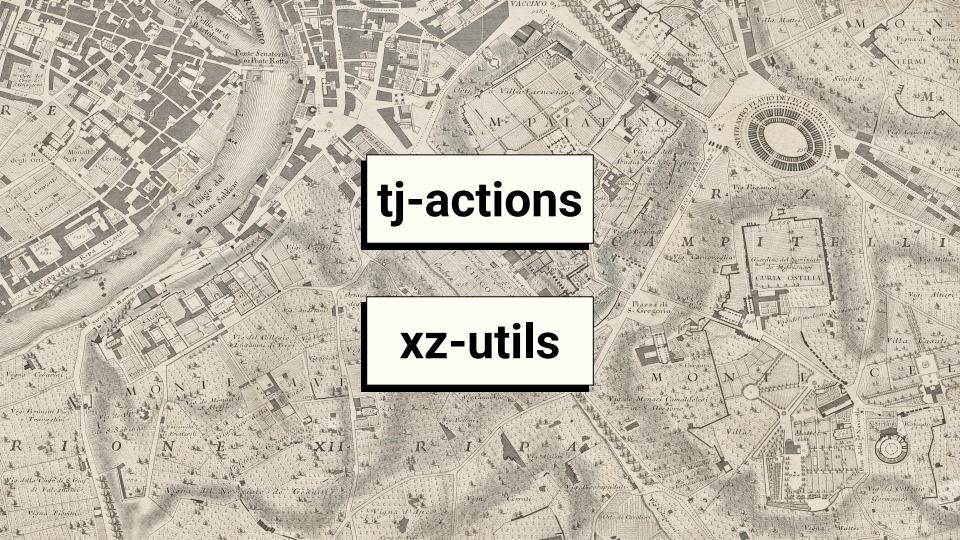
Software Supply Chain Security

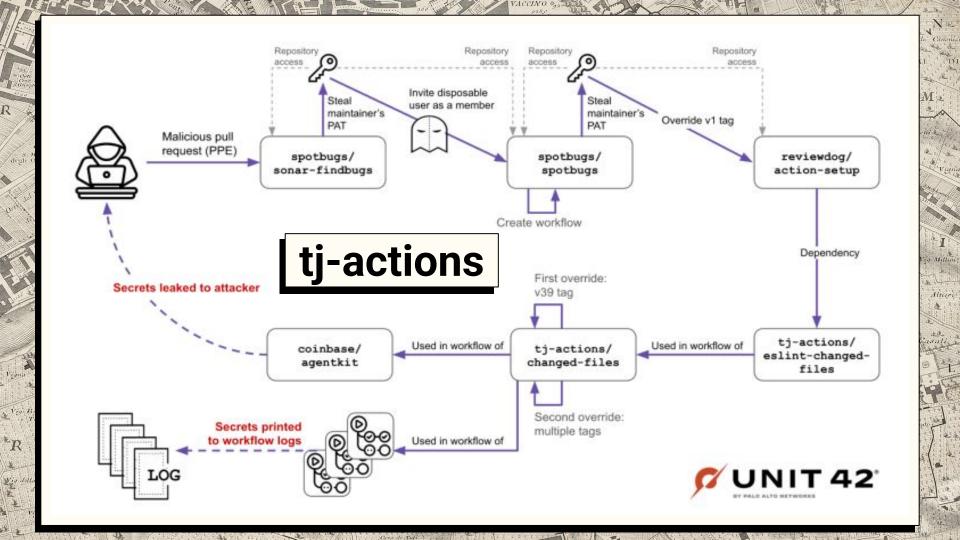
The engineering of defenses against software supply chain attacks and vulnerabilities.

Russ Cox. 2025.

Fifty Years of Open Source Software Supply Chain Security

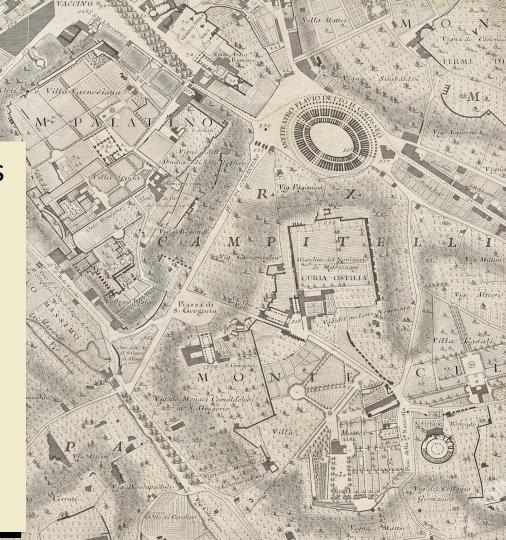








- Immutable GitHub Actions
- Transparency Logs
- Version Pinning
- Tag Protection
- Malicious Fork/Branch Scans
- Vulnerable CI Scans

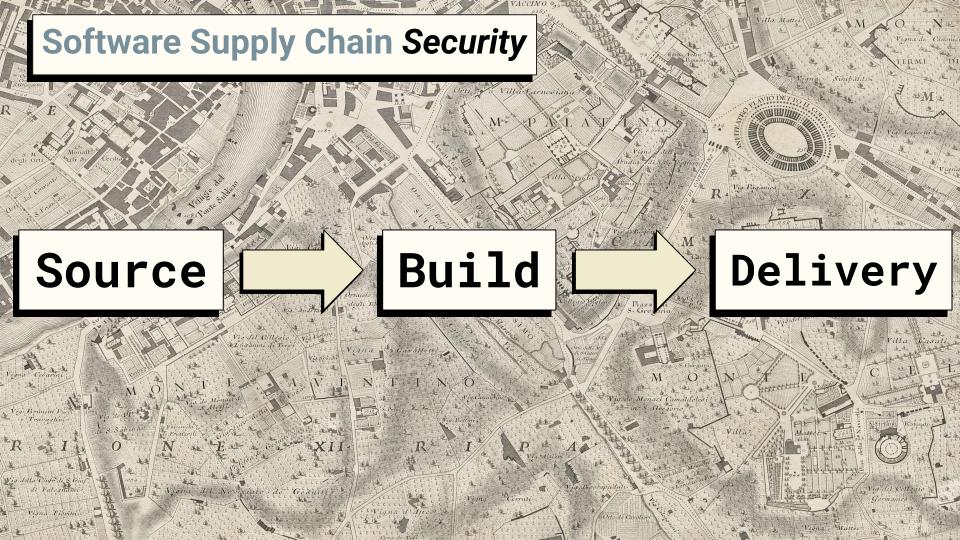


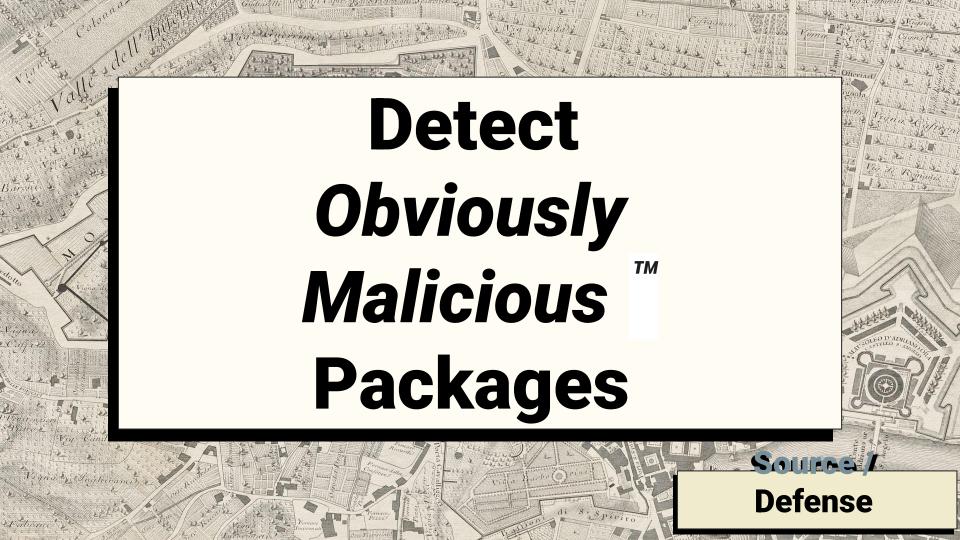
tj-actions

xz-utils

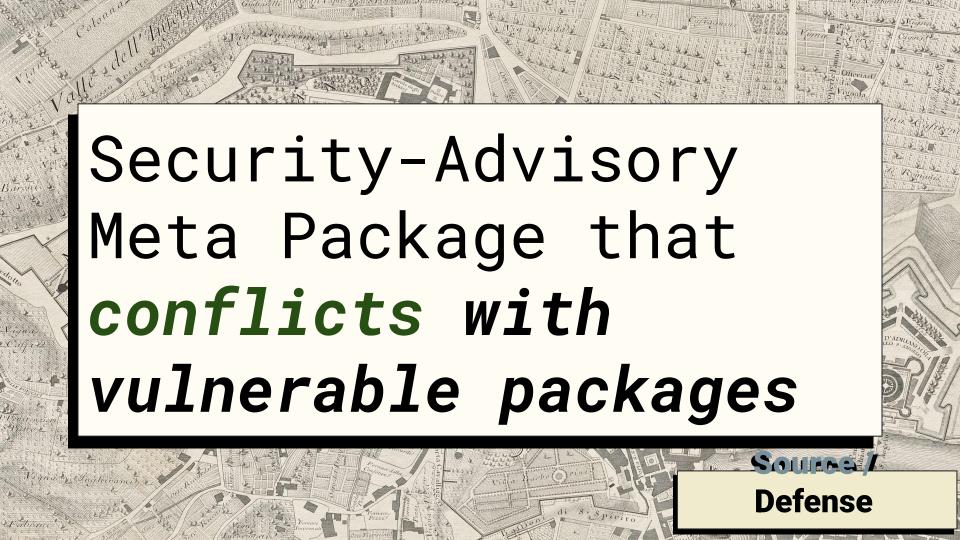
- Immutable GitHub Actions
- Transparency Logs
- Version Pinning
- Tag Protection
- Malicious Fork/Branch Scans
- Vulnerable CI Scans

- ozz fuzz
- Minimal Dependency
- Dynamic Loading
- Source/Release diffs
- Security Audits



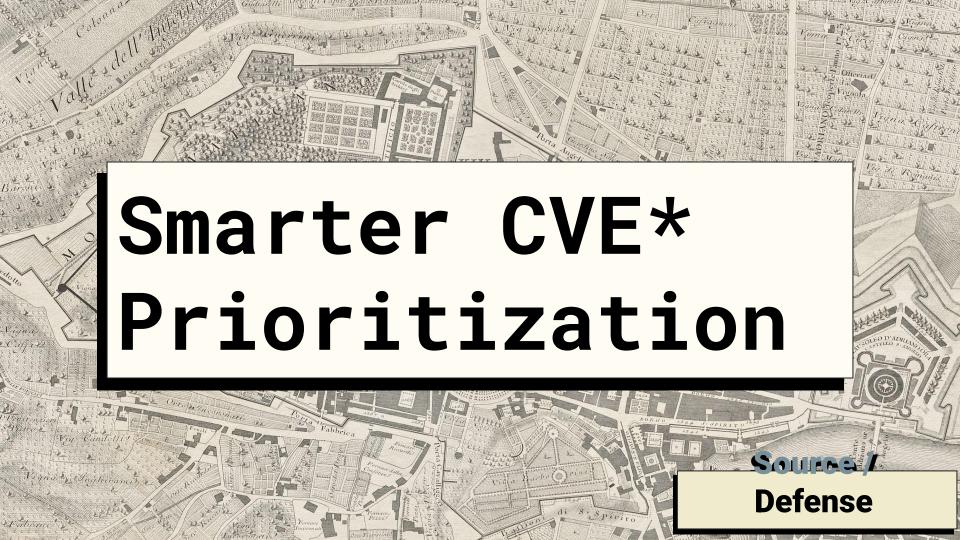


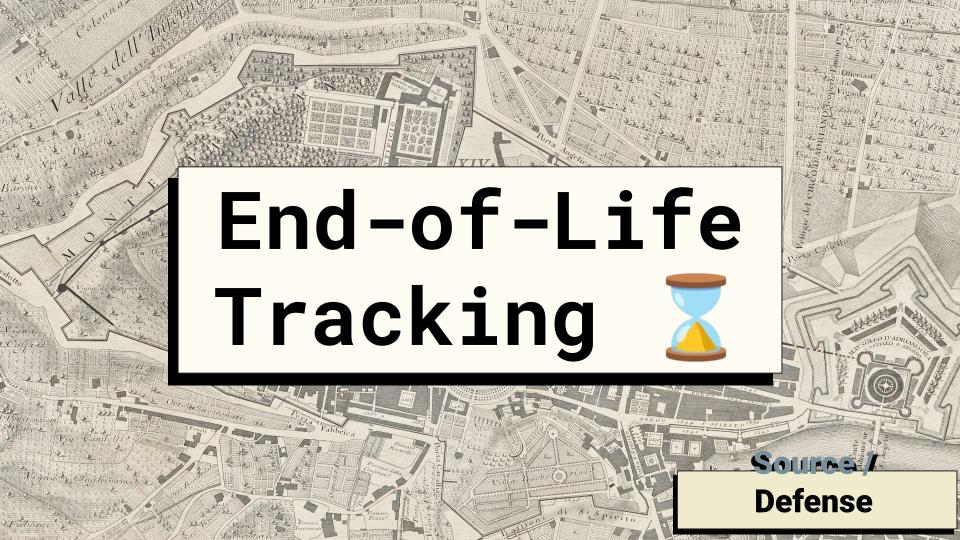




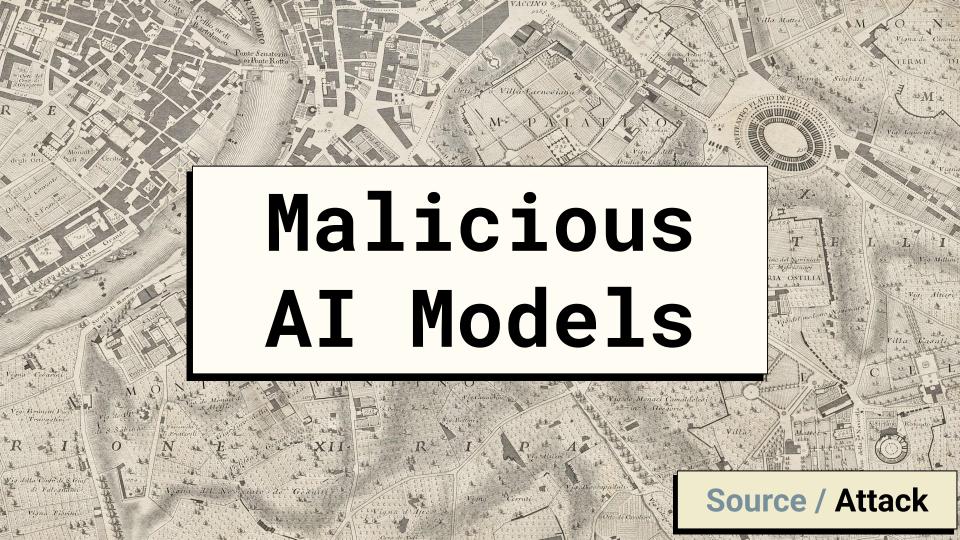












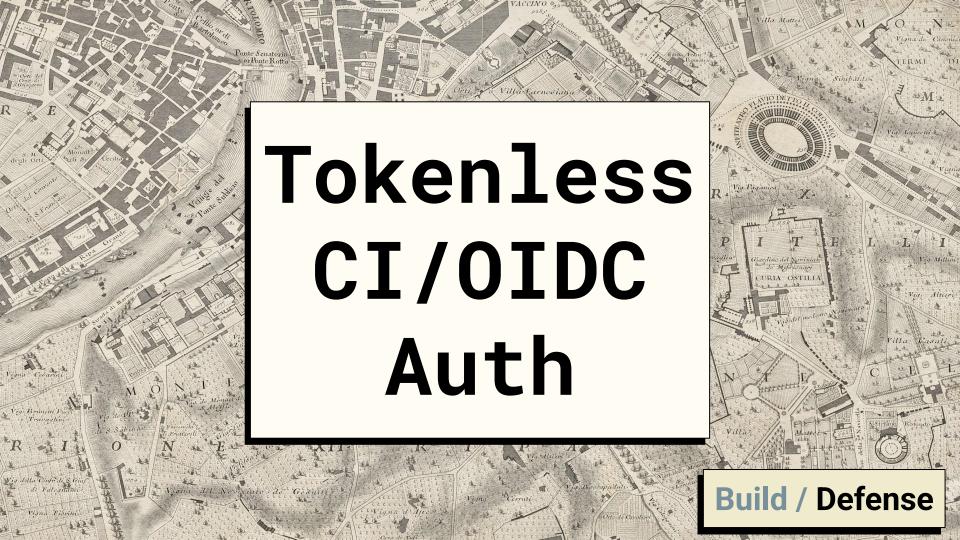


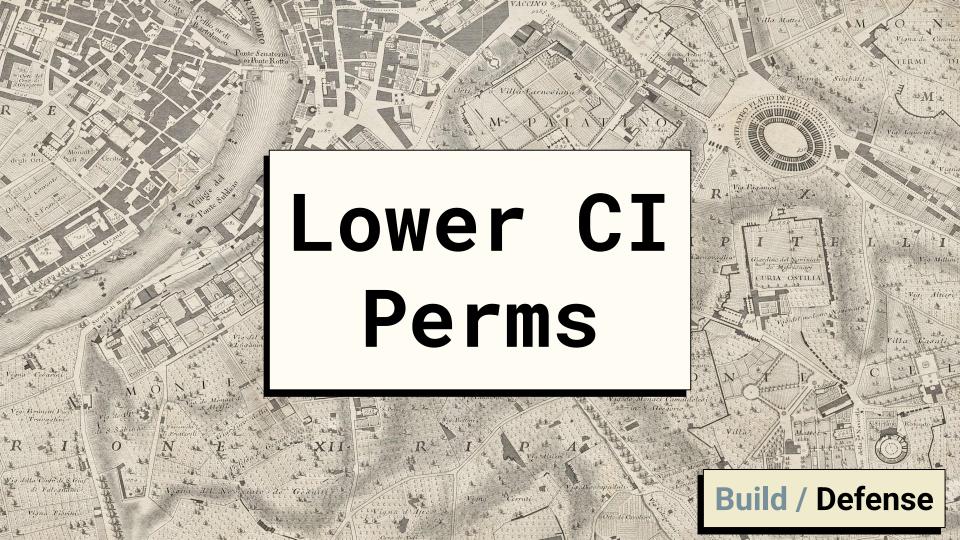




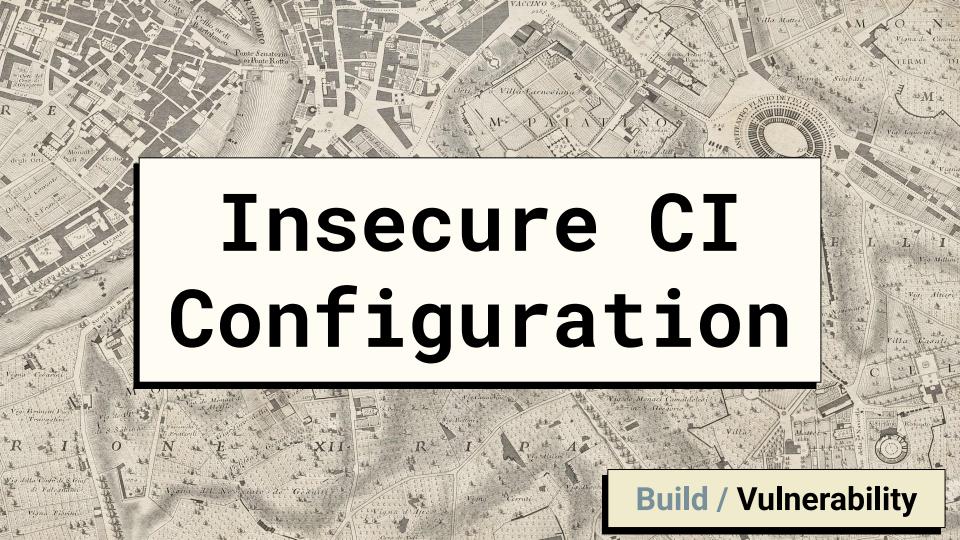




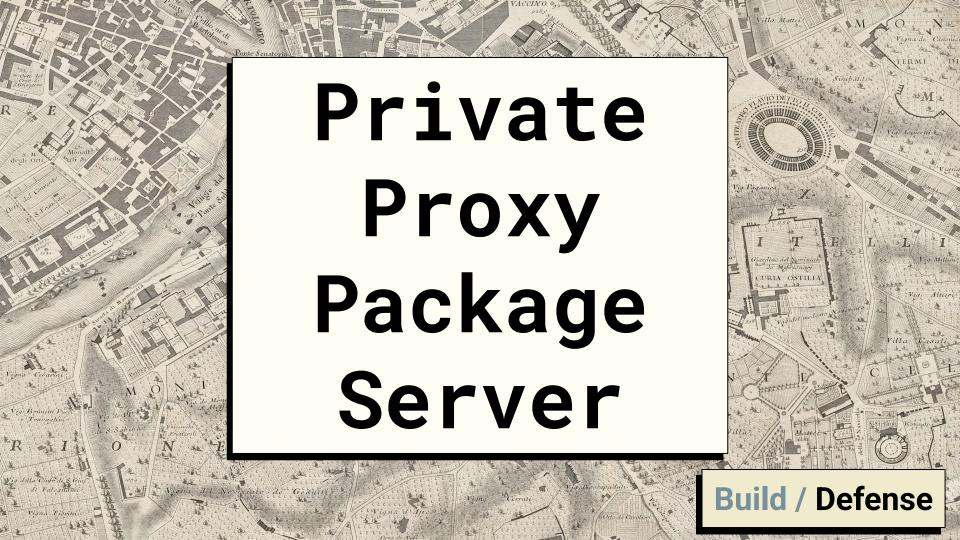












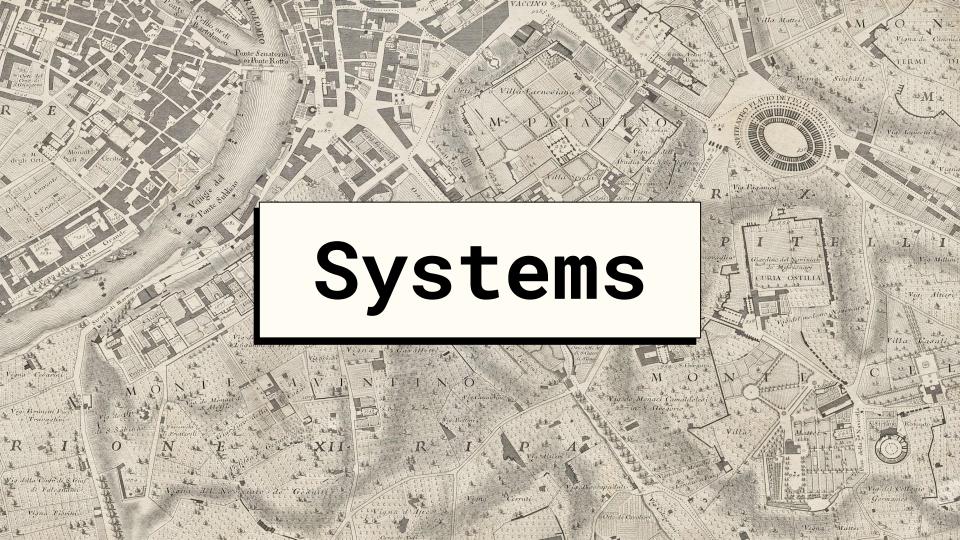




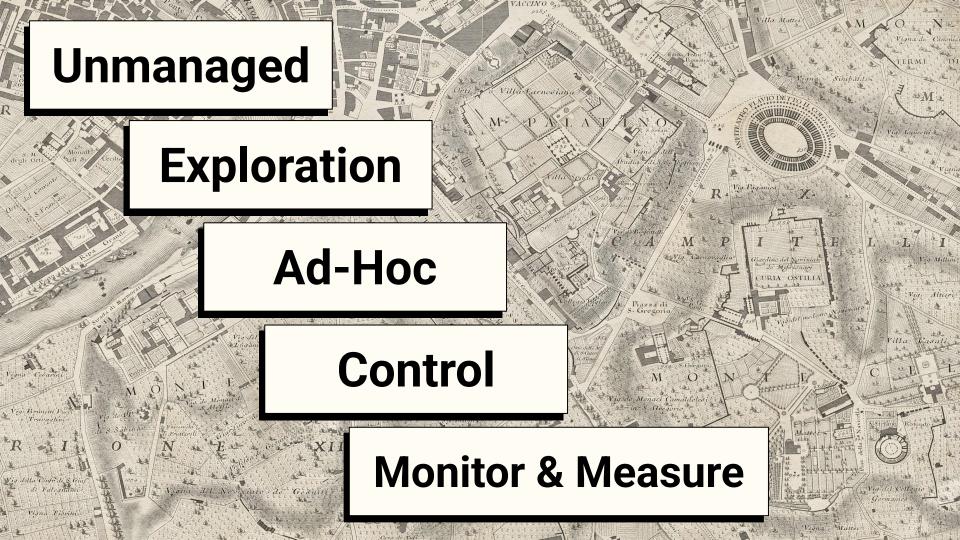












Policy Governance Compliance

Consistency / Build & Release

Inventory /
Supplier Hygiene /
Transparency

Resilience / Remediation

Policy Governance
Compliance

Consistency /
Build & Release

Inventory /
Supplier Hygiene /
Transparency

Resilience / Remediation





#1

Vulnerability and patch management



#2

Insufficient visibility of software dependencies or software supply chain



#3

Trustworthiness of software source*



#4

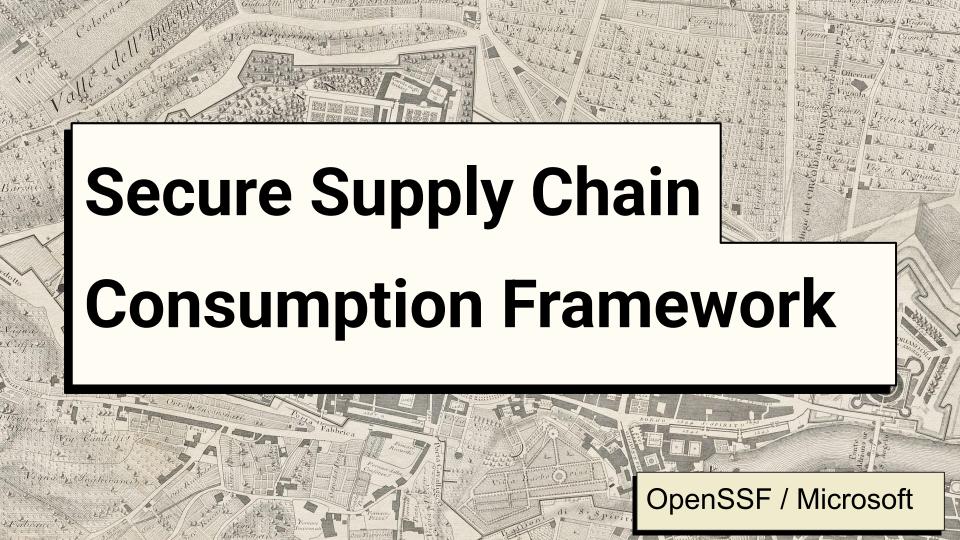
Short upstream security maintenance/ support periods



#5

Lack of in-house skills and experience

IDC Survey, Q4 2024 by Canonical/Google







R

Vig 1

 $^{*}R$

Minimum OSS Governance Program

- Use package managers [ING-1]
- Local copy of artifact [ING-2]
- Scan with known vulns [SCA-1]
- Scan for software licenses [SCA-2]
- Inventory OSS [INV-1]
- Manual OSS updates [UPD-1]



Secure Consumption and Improved MTTR

- Scan for end of life [SCA-3]
- Have an incident response plan [INV-2]
- Auto OSS updates [UPD-2]
- Alerts on vulns at PR time [UPD-3]
- Audit that consumption is through approved ingestion method [AUD-2]
- Validate integrity of OSS [AUD-3]
- Secure package source file configuration [ENF-1]



Malware Defense and Zero-Day Detection

- Deny list capability [ING-3]
- Clone OSS source [ING-4]
- Scan for malware [SCA-4]
- Proactive security reviews [SCA-5]
- Enforce OSS provenance [AUD-1]
- Enforce consumption from curated feed [ENF-2]



M

Advanced Threat Defense

- Validate the SBOMs of OSS consumed [AUD-4]
- Rebuild OSS on trusted infrastructure [REB-1]
- Digitally sign rebuilt OSS [REB-2]
- Generate SBOM for rebuilt OSS [REB-3]
- Digitally sign protected SBOMs [REB-4]
- Implement fixes [FIX-1]

s2c2f





