CSCI E-117A SPRING 2024

Lecture 11
April 15, 2024

SECURE APPLICATIONS: MANAGING THE DEPLOYMENT INFRASTRUCTURE

LECTURE 11 AGENDA

- *Happy Tax Day*
- *Users*
  - *wrap up*
- *CDF v ZTMM*
  - *Comparison, Discussion*
- *In the News*
  - *Examination of MGM ALPHV Compromise*
  - *CapitalOne Compromise*
- *Assignment IV, Capstone*
  - *Discussion, Q&A*

QUICK ANNOUNCEMENTS

Happy Tax Day (April 15) for those in USA

Canada, enjoy those extra 2 weeks

Capstone Due May 4 (Saturday!)

ASSIGNMENT 4

Mapping Mitre to ZTMM

	Application Access	Application Threat Protection	Accessible Applications
M1040 – Behavior Prevention on Endpoint	2	18	
M1038 – Execution Prevention	0	6	1
M1052 - User Account Control	5		1
M1032 – Multi-factor Authentication	29		2
M1035 - Limit Access to Resource Over Network	3	1	34
M1016 - Vulnerability Scanning	0	14	1

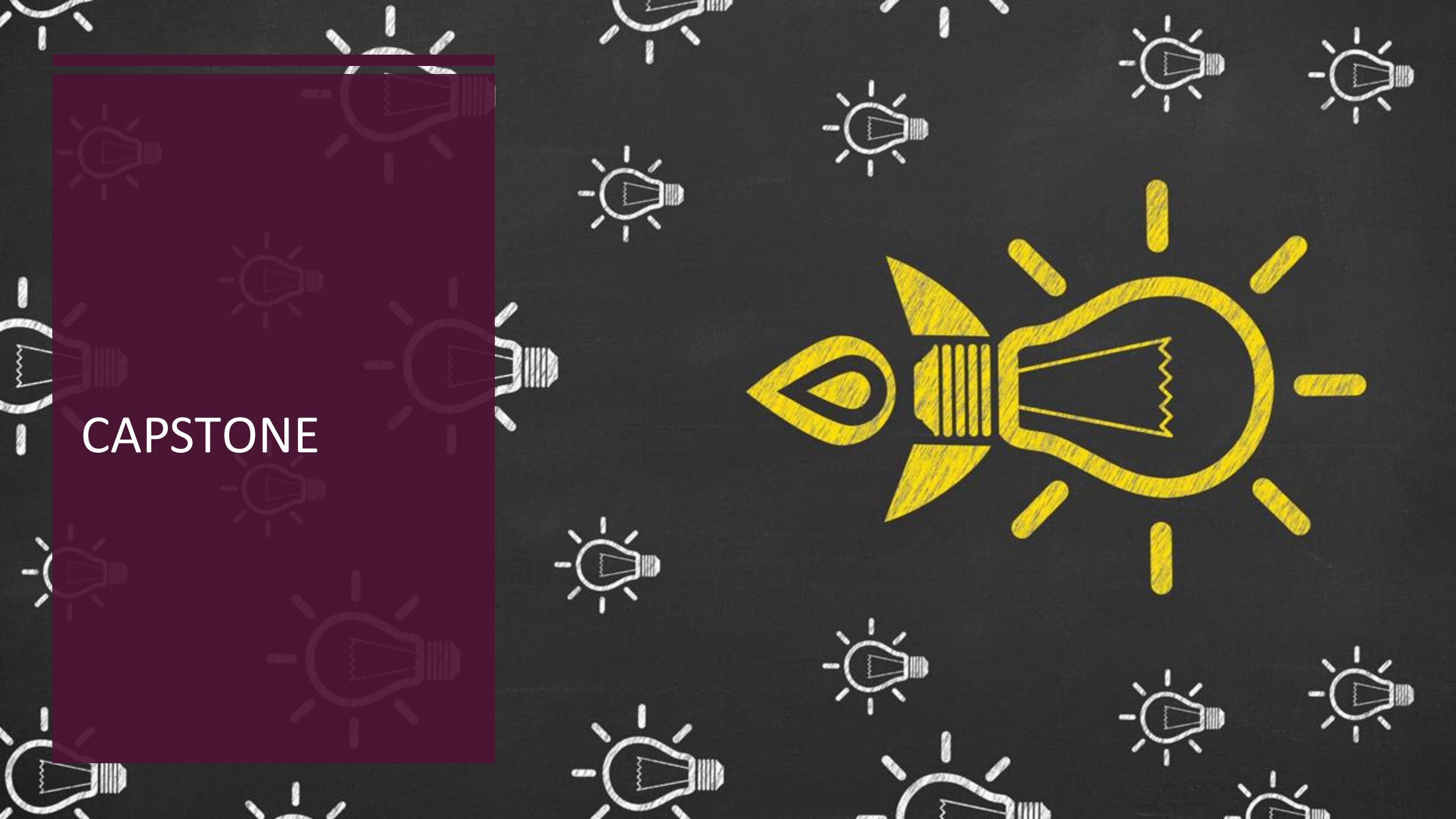
Zero Trust Application Architecture prioritization

	Priority 1	Priority 2	Priority 3
Application Access (former Access Authorization)	30	8	1
Application Threat Protections (formerly Threat Protection)	7	27	5
Accessible Applications (formerly accessibility)	2	4	33

Application controls and Generative AI

BENEFIT FROM GENAI DEFENSE	**Selected**
Application Access (former Access Authorization)	8
Application Threat Protections (formerly Threat Protection)	29
Accessible Applications (formerly accessibility)	1
BENEFIT (SUFFER) FROM GENAI ATTACK	
Application Access (former Access Authorization)	25
Application Threat Protections (formerly Threat Protection)	9
Accessible Applications (formerly accessibility)	4

CAPSTONE



CAPSTONE

Is now published

- **Its LONG in terms of the numbers of pages. That means there is a lot of reading**
- **It builds on the assignments (so if you have done Assignments 1-4 it will help you a LOT)**
- **We will dedicate time to it next class**

IDENTITY / USERS

IDENTITY / USERS

	Authentication	Identity Stores	Risk Assessments	Access Management
Traditional	Agency authenticates identity using either passwords or multi-factor authentication (MFA) with static access for entity identity.	Agency only uses self- managed, on-premises (i.e., planned, deployed, and maintained by agency) identity stores.	Agency makes limited determinations for identity risk (i.e., likelihood that an identity is compromised).	Agency authorizes permanent access with periodic review for both privileged and unprivileged accounts.
Initial	Agency authenticates identity using MFA, which may include passwords as one factor and requires validation of multiple entity attributes (e.g., locale or activity).	Agency has a combination of self-managed identity stores and hosted identity store(s) (e.g., cloud or other agency) with minimal integration between the store(s) (e.g., Single Sign- on.).	Agency determines identity risk using manual methods and static rules to support visibility.	Agency authorizes access, including for privileged access requests, that expires with automated review.
Advanced	Agency begins to authenticate all identity using phishing-resistant MFA and attributes, including initial implementation of passwordless MFA via FIDO2 or PIV	Agency begins to securely consolidate and integrate some self-managed and hosted identity stores.	Agency determines identity risk with some automated analysis and dynamic rules to inform access decisions and response activities.	Agency authorizes need- based and session-based access, including for privileged access request, that is tailored to actions and resources.
Optimal	Agency continuously validates identity with phishing-resistant MFA, not just when access is initially granted.	Agency securely integrates their identity stores across all partners and environments as appropriate.	Agency determines identity risk in real time based on continuous analysis and dynamic rules to deliver ongoing protection.	Agency uses automation to authorize just-in-time and just-enough access tailored to individual actions and individual resource needs.

(IDENTITY) AUTHENTICATION ZERO TRUST MATURITY

TRADITIONAL	INITIAL	ADVANCED	OPTIMAL
Agency authenticates identity using **either passwords or multi-factor authentication (MFA) with static access** for entity identity.	Agency authenticates identity using **MFA, which may include passwords as one factor and requires validation of multiple entity attributes (e.g., locale or activity).**	Agency begins to authenticate all identity using **phishing-resistant MFA and attributes,** including initial implementation of **password-less MFA via FIDO2 or PIV**	Agency **continuously validates identity with phishing-resistant MFA**, not just when access is initially granted.

(IDENTITY) IDENTITY STORES ZERO TRUST MATURITY

TRADITIONAL	INITIAL	ADVANCED	OPTIMAL
Agency only uses **self-managed, on-premises** (i.e., planned, deployed, and maintained by agency) **identity stores.**	Agency has a **combination of self-managed identity stores and hosted identity store(s)** (e.g., cloud or other agency) with minimal integration between the store(s) (e.g., Single Sign- on.).	Agency begins to **securely consolidate and integrate some self-managed and hosted identity stores.**	Agency securely **integrates their identity stores across all partners and environment**s as appropriate.

(IDENTITY) RISK ASSESSMENTS ZERO TRUST MATURITY

TRADITIONAL	INITIAL	ADVANCED	OPTIMAL
Agency makes **limited determinations for identity risk** (i.e., likelihood that an identity is compromised).	Agency determines **identity risk using manual methods and static rules to support visibility**.	Agency determines identity risk with some **automated analysis and dynamic rules to inform access decisions** and response activities.	Agency determines identity risk in real time based on continuous analysis and dynamic rules to deliver ongoing protection.

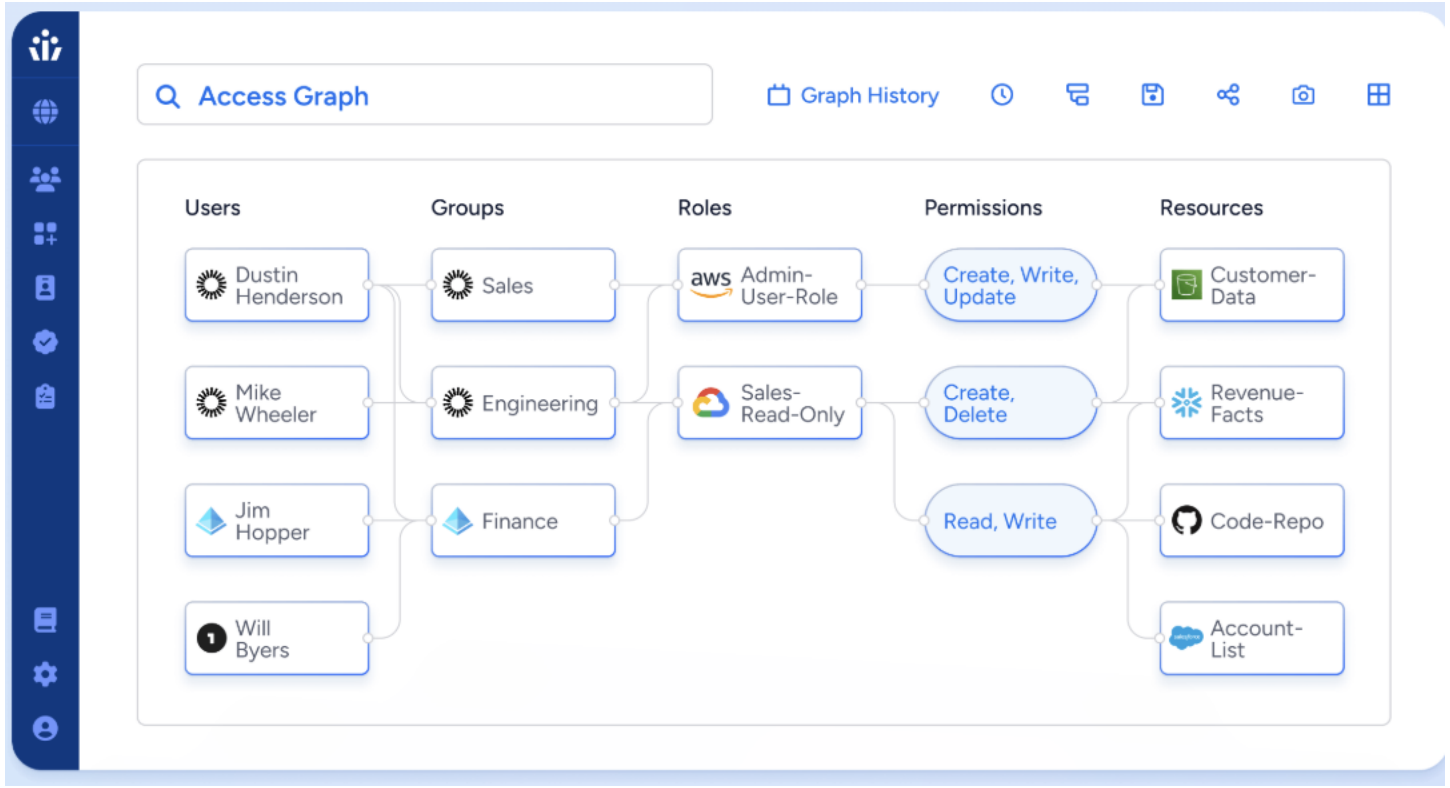
ACCESS MANAGEMENT ZERO TRUST MATURITY

TRADITIONAL	INITIAL	ADVANCED	OPTIMAL
Agency authorizes **permanent access with periodic review** for both privileged and unprivileged accounts.	Agency **authorizes access,** including for privileged access requests**, that expires with automated review**.	Agency **authorizes need-based and session-based access**, including for privileged access request, that is tailored to actions and resources.	Agency uses **automation to authorize just-in-time and just-enough access tailored to individual actions and individual resource needs**

NON HUMAN IDENTITIES

Non Human Identities

- Are widely used : estimates are anywhere from 30-50 NHI per human identity in average company
- Are not “interactive” : you can’t have interactive protocols such as MFA in place
- Are (often) tied to their authenticator & privileges : a certificate has the keys used for authentication
- Are (often) “brittle” when it comes to credential rotation (especially if rotating keys)
- Are (often) highly trusted and therefore “over permissioned”
- Are considered a juicy vector for identity compromise & lateral movement



- This is a sample NHI view from an NHI management vendor
- NOTE THE FOCUS ON SAAS ….

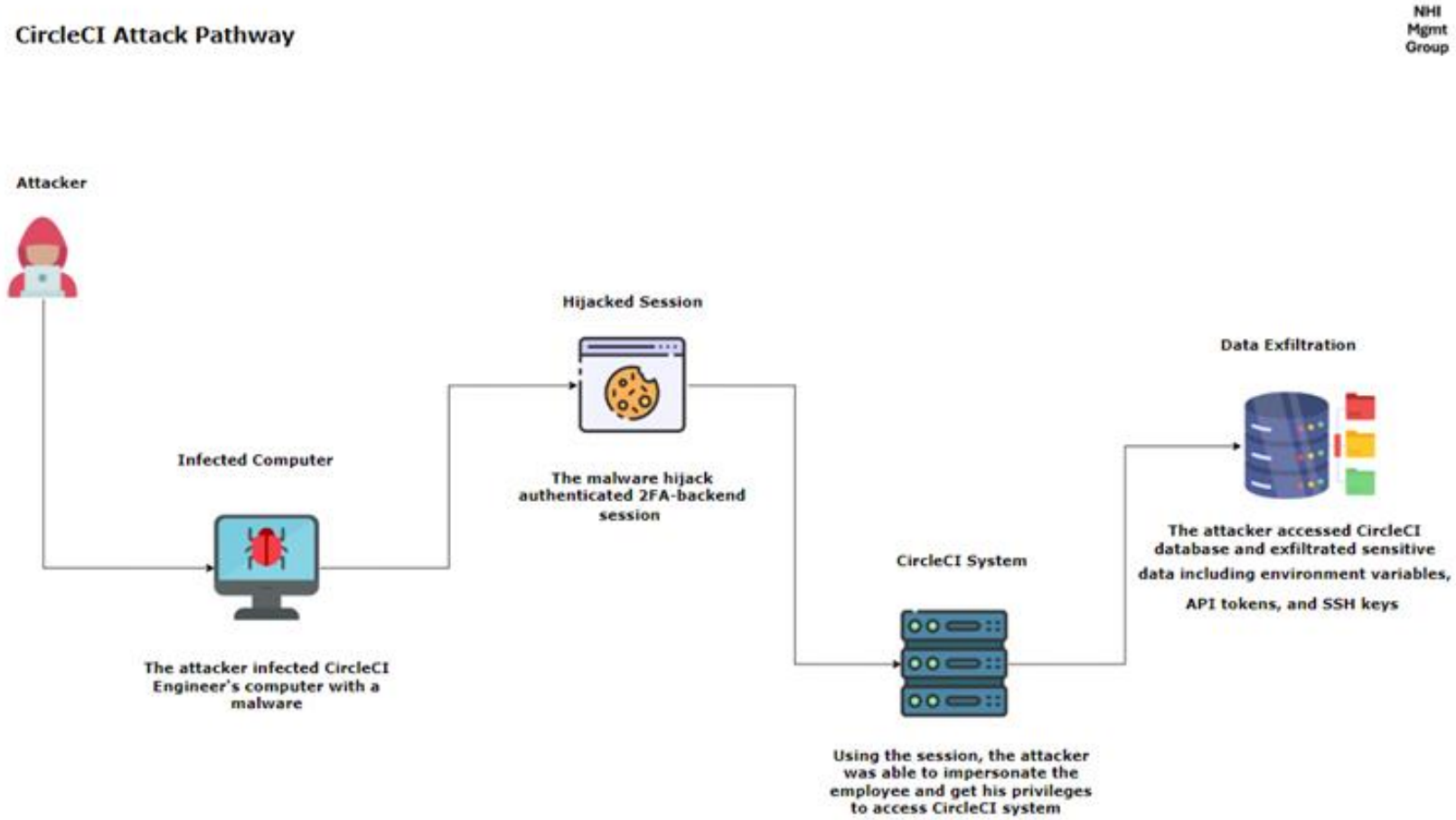
- And to be sure, (eg) AWS S3 buckets are a (sadly) very common NHI-compromised target

CLASS DISCUSSION: POLL

- *When you think of Supply Chain Security risk, what do you typically think of as the most important thing to check*
  - *How the third party (the entity in your supply chain) develops applications (its security application development environment and lifecycle, including testing)*
  - *How the third party (the entity in your supply chain) manages (including onboarding) the users that develop the applications?*
  - *How the third party (the entity in your supply chain) manages non-human identities, including rotation of credentials?*

CIRCLE CI ATTACK PATHWAY

https://nhimg.org/circleci-breach



CIRCLE CI
https://nhimg.org/circleci-breach

Exposed Customer Data:

- Secrets such as environment variables, API tokens, and SSH keys were exfiltrated.
- Integration credentials for platforms like GitHub and AWS were compromised.
- Although customer data was encrypted at rest, access to encryption keys rendered the protection ineffective.

Operational Fallout

- Customers were forced to rotate all secrets and audit their systems for unauthorized activity.
- Development pipelines relying on CircleCI were temporarily disrupted as tokens were invalidated and rotated.

Immediate Measures:

- Invalidated all customer secrets stored on the platform.
- Issued guidance for rotating API tokens, OAuth keys, and other secrets.
- Partnered with third-party platforms like GitHub and AWS to automate token rotations.

Recommendations:

- **Token Rotation -** Customers should immediately rotate all secrets stored in CircleCI, including third-party integrations like AWS, GitHub, and Kubernetes.

CIRCLE CI TOKEN ROTATIONS
https://circleci.com/docs/managing-api-tokens/#overview

CircleCI API Tokens

- To use the CircleCI API or view details about your pipelines, you will need API tokens with the appropriate permissions.
  - **Personal API tokens:** These tokens are used to interact with the CircleCI APIs and grant full read and write permissions.
  - **Project API tokens:** These tokens allow you to read/write information for specific projects. Project tokens have three scope options: *Status*, *Read Only*, and *Admin*.
- API tokens cannot be modified after they have been created. The only way to change existing tokens is to delete and recreate them

GitHub Oauth Tokens / Token Rotation

- CircleCI triggered GitHub OAuth token rotation for all clients.
- For customers working on rotating secrets and keys, you should rotate keys at the source (the system to which they provide access) and then store the new secrets on CircleCI. Simply removing them from CircleCI is not enough.

Project SSH keys:

- Go to Project Settings > SSH Keys.
- Delete the Deploy Key and add it again.
- If you were using any additional keys, then those need to be deleted and recreated.
- **Note: SSH keys will also need to be rotated from the target environment.**

Runner Tokens: using the CircleCI CLI, run the following commands:

- circleci runner token *comman information*
- Following these commands, you will need to add the created token to your launch-agent-config.yml and restart your runner service

Rotate API tokens

API Token rotation occurs when an old API token is replaced with a new token.

Because API Tokens can be shared, passed around between employees and teams, and exposed inadvertently, it is always good practice to periodically regenerate new API Tokens. Many organizations automate this process, running a script when an employee leaves the company or when a token has been considered leaked.

Rotating a personal API token

1. In the CircleCI application, go to your User settings .
2. Select Personal API Tokens .
3. Select the **X** in the **Remove** column for the token you wish to replace and confirm your deletion.
4. Select **Create New Token**.
5. In the **Token name** field, type a new name for the old token you are rotating. It the same name given to the old token.
6. Select **Add API Token**.
7. After the token appears, copy and paste it to another location. You will not be view the token again.

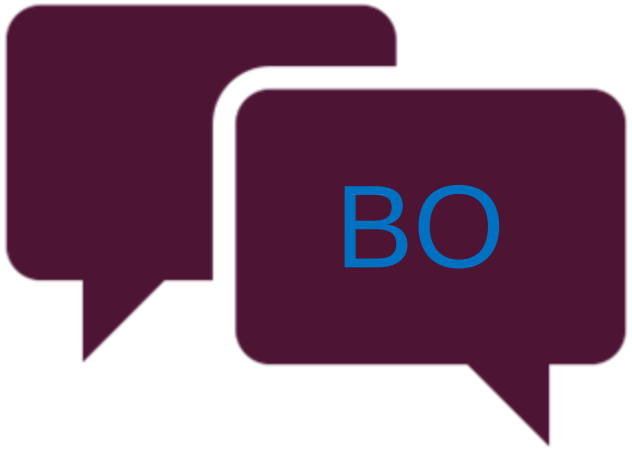
Rotating a project API token

1. In the CircleCI application, go to your project's settings. There are various ways to get there. One way is to select **Projects** in the sidebar, then the ellipsis (`...`) next to your project and select **Project Settings**.
2. Select **API Permissions**.
3. Select the **X** in the **Remove** column for the token you wish to replace. When the confirmation window appears, enter the text `DELETE` in the form and click the **Delete API Token** button.
4. Select **Create API Token**.
5. Choose the same scope used for the old token from the dropdown menu.
6. In the **Label** field, type a label for the token. It can be the same name given to the old token.
7. Select **Add API Token**.

CLASS DISCUSSION: POLL

- *Do you want to change your mind?*
- *When you think of Supply Chain Security risk, what do you typically think of as the most important thing to check*
  - *How the third party (the entity in your supply chain) develops applications (its security application development environment and lifecycle, including testing)*
  - *How the third party (the entity in your supply chain) manages (including onboarding) the users that develop the applications?*
  - *How the third party (the entity in your supply chain) manages non-human identities, including rotation of credentials?*

CLASS DISCUSSION: POLL



- *Most approaches to third party risk management (another term for supply chain security) are*
  - *A) Questionnaire/checklist based*
  - *B) Focus on secure development practices*
- *Given the discussion of Identity Proofing (last week) and Non Human Identities, is this really the right focus area?*
- *How have you seen TPRM handled / focused?*
- *If you had to drive a TPRM discipline, what would you focus on?*
- *What would you do if your critical vendors did not have a (In Your Humble Opinion) good enough Identity discipline including proofing and NHIs?*

10 min
BREAK
BACK
9:05PM ET

ZERO TRUST MATURITY MODEL – CSC117 CATEGORIES & FUNCTIONS

Identity	Devices	Networks	Applications and Workloads
Authentication	Policy Enforcement & Compliance Monitoring	Network Segmentation	Application Access
Identity Stores	Asset & Supply Chain Risk Management	Network Traffic Management	Application Threat Protections
Risk Assessments	Resource Access	Traffic Encryption	Application Resilience (*)
Access Management	Device Threat Protection	Network Resilience	Secure Application Oversight & Lifecycle (*)

XZ UTILS

https://www.invicti.com/blog/web-security/xz-utils-backdoor-supply-chain-rce-that-got-caught/

- The xz-utils project was created and maintained by Lasse Collin until a helpful and very insistent contributor going by the name of Jia Tan recently succeeded in fully taking over the project on GitHub. Among Jia's latest commits were alleged compression performance improvements to the liblzma library, published in versions 5.6.0 and 5.6.1 of xz-utils. These are the versions that included the backdoor, but the compression utility was only a stepping stone to a much bigger prize
- In some system configurations, OpenSSH depends on the liblzma library, including any running SSH
- The backdoor was reported by Red Hat as CVE-2024-3094 as "malicious code" in the package.
  - What makes it different from most software vulnerabilities is that the source code itself is clean and secure.
  - The backdoor is hidden in separate "test" files and only reassembled and inserted into the library during compilation.
- One theory is that the JiaT75 account is not an individual but an advanced threat actor group, with many pointing to APT29 (aka Cozy Bear) as a group with similarly stealthy operational patterns and sufficiently advanced tech skills. You may remember them from the SolarWinds Orion hack—also a supply chain attack, as it happens. Whatever the case, Jia (unsurprisingly) vanished into thin air when the backdoor was reported and has not been seen since.

XZ UTILS

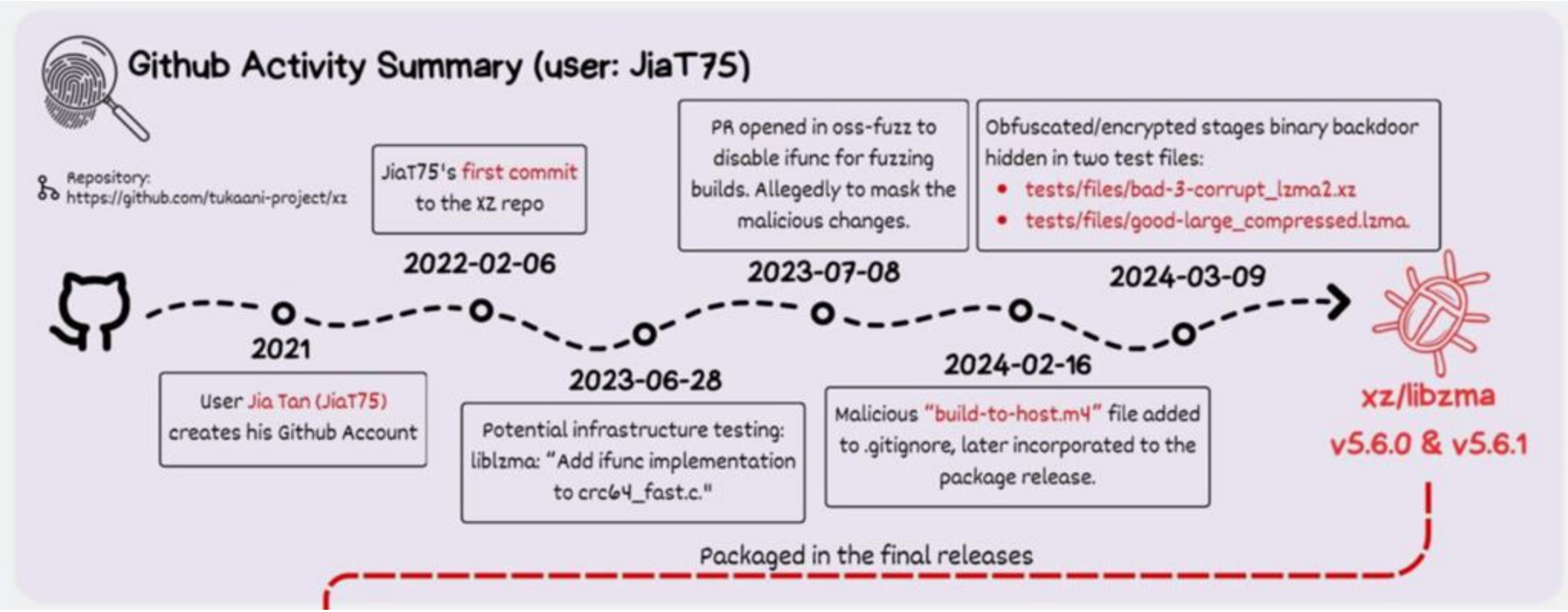
https://www.invicti.com/blog/web-security/xz-utils-backdoor-supply-chain-rce-that-got-caught/

- To avoid detection by scanners, the malware binary was, in effect, cut up into several pieces, and the gaps filled up with junk. For additional stealth, it is only included in the packaged tarball, so it’s not there if anyone examines the individual files in the repository. But if the package from an infected tarball is compiled on a system that meets specific configuration requirements, the build scripts reassemble the malicious code and attach it to the liblzma library, where it waits for a specific function call from a remote secure shell (SSH) session.
- If all the conditions are met, a malicious actor can activate the backdoor by connecting to a compromised system over SSH and sending their encrypted access key. When successful, this could allow them to bypass the entire authentication process and gain unauthenticated remote access to the system.

xz utils

https://arstechnica.com/security/2024/04/what-we-know-about-the-xz-utils-backdoor-that-almost-infected-the-world/

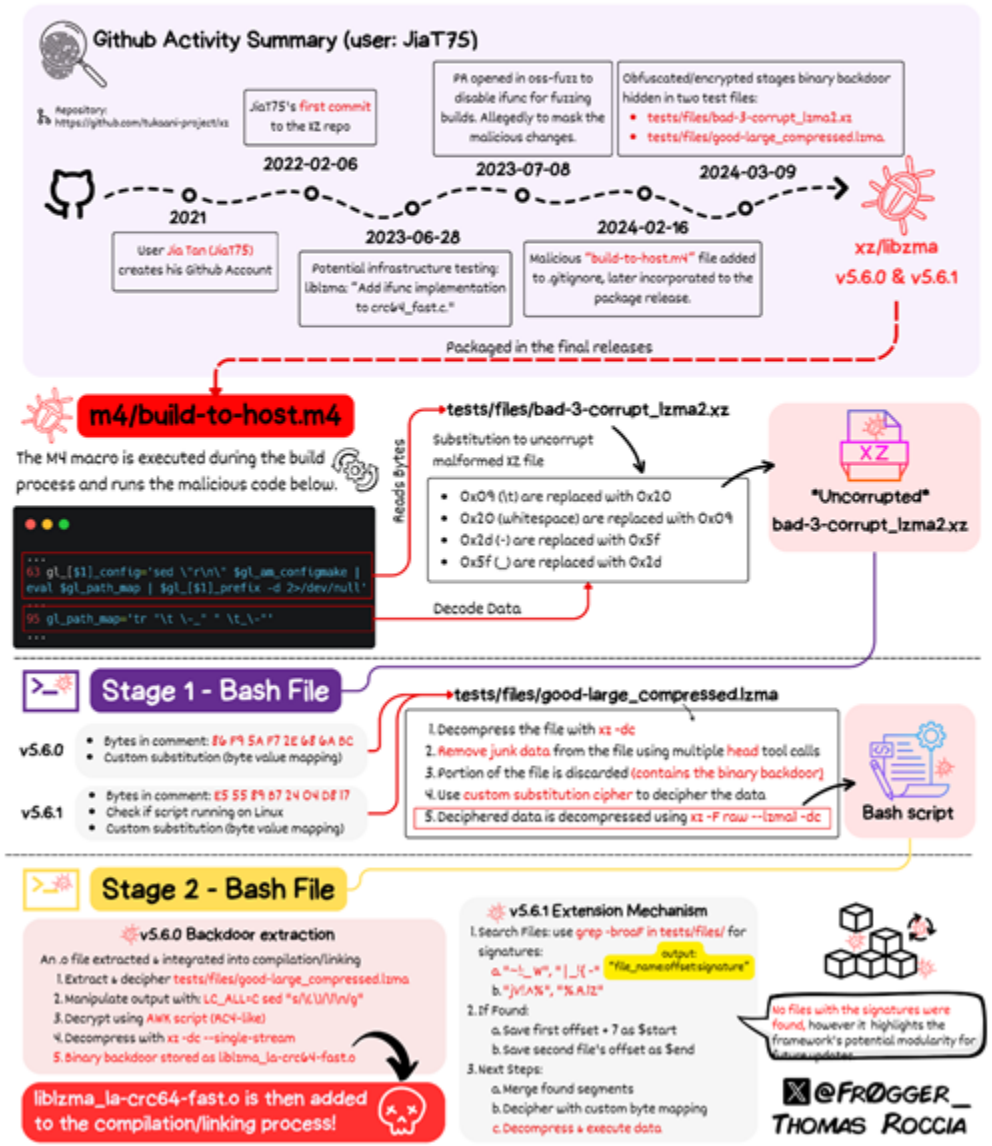
- XZ Utils provides lossless data compression on virtually all Unix-like operating systems, including Linux.
  - It's hard to overstate the complexity of the social engineering and the inner workings of the backdoor.



XZ Outbreak (CVE-2024-3094)

XZ Utils is a collection of open-source tools and libraries for the XZ compression format, that are used for high compression ratios with support for multiple compression algorithms, notably LZMA2.

On Friday 29th of March, Andres Freund (principal software engineer at Microsoft) emailed oss-security informing the community of the discovery of a backdoor in xz/liblzma version 5.6.0 and 5.6.1.



Identity	Devices	Networks	Applications and Workloads
Authentication	Policy Enforcement & Compliance Monitoring	Network Segmentation	Application Access
Identity Stores	Asset & Supply Chain Risk Management	Network Traffic Management	Application Threat Protections
Risk Assessments	Resource Access	Traffic Encryption	Application Resilience (*)
Access Management	Device Threat Protection	Network Resilience	Secure Application Oversight & Lifecycle (*)

Managing my environment (we use XZUtils)

- Open-Source Software (Device/Supply Chain)
- Client-Server Software (Device/Policy Enforcement)
- Client-Server, Server-Server Communication (Networks/Network Segmentation)
- SSH Authentication with Public-Private Key Pairs (Identity/Authentication)
- Third Party (Open Source) Developed Software (Applications/Secure Application Development)

Managing my supply chain (oversight to XZUtils and other third-party providers)

- Know who is contributing - Identity / Proofing (*)
- Monitor for unusual behaviour - Identity / Risk Assessment
- Apply secure development practices - Application / Secure Application Development

CLASS DISCUSSION: POLL

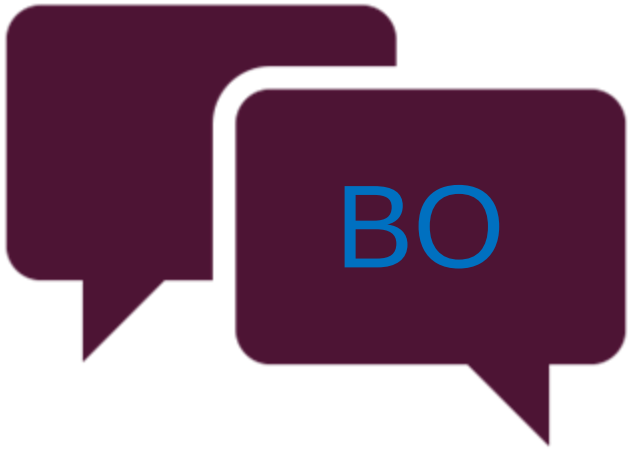
- *Given the discussion of XZ Utils, what would be your preferred ordering of importance of the ZTMM Categories & Functions:*
- For your environments and disciplines:

1. Device/Supply Chain, Device/Policy Enforcements, Identity/Authentication, Identity / Risk Assessment
2. Identity / Risk Assessment, Device/Policy Enforcements, Device/Supply Chain, Identity/Authentication

- For your third party's environments and disciplines:

1. Identity / Proofing (*), Identity / Risk Assessment, Applications/Secure Application Development
2. Applications/Secure Application Development, Identity / Proofing (*), Identity / Risk Assessment
3. Applications/Secure Application Development, Identity / Risk Assessment, Identity / Proofing (*)

CLASS DISCUSSION: POLL



- *From the XZUtils use case, it (should be) clear that no single ZTMM category or function will be sufficient in protecting against threats*
- *For example*
  - *If even JiaTan had been subjected to more rigorous identity proofing, that doesn't protect against ill-intent*
  - *Likewise, the low-and-slow approach of JiaTan built up trust and probably would have passed risk assessment monitoring*
  - *A strong discipline of compiling your own OSS might have avoided the XZUtils case*
  - *Strong application testing, including performance testing, might have highlighted anomalies leading to investigation*
  - *A strongly segmented network would limit lateral movement potential*
  - *And so on….*
- *What does this tell you about how to manage a secure environment from an operational and discipline point of view?*

Identity	Devices	Networks	Applications and Workloads
Authentication	Policy Enforcement & Compliance Monitoring	Network Segmentation	Application Access
Identity Stores	Asset & Supply Chain Risk Management	Network Traffic Management	Application Threat Protections
Risk Assessments	Resource Access	Traffic Encryption	Application Resilience (*)
Access Management	Device Threat Protection	Network Resilience	Secure Application Oversight & Lifecycle (*)

IDENTITY	TRADITIONAL	INITIAL	ADVANCED	OPTIMAL
Visibility and Analytics Capability	Agency collects user and entity activity logs, especially for privileged credentials, and performs some routine manual analysis.	Agency collects user and entity activity logs and performs routine manual analysis and some automated analysis, with limited correlation between log types.	Agency performs automated analysis across some user and entity activity log types and augments collection to address gaps in visibility.	Agency maintains comprehensive visibility and situational awareness across enterprise by performing automated analysis over user activity log types, including behavior-based analytics.
Automation and Orchestration Capability	Agency manually orchestrates (onboards, offboards, and disables) self-managed identities (users and entities), with little integration, and performs regular review.	Agency manually orchestrates privileged and external identities and automates orchestration of non-privileged users and of self-managed entities.	Agency manually orchestrates privileged user identities and automates orchestration of all identities with integration across all environments.	Agency automates orchestration of all identities with full integration across all environments based on behaviors, enrollments, and deployment needs.
Governance Capability	Agency implements identity policies (authentication, credentials, access, lifecycle, etc.) with enforcement via static technical mechanisms and manual review.	Agency defines and begins implementing identity policies for enterprise-wide enforcement with minimal automation and manual updates.	Agency implements identity policies for enterprise-wide enforcement with automation and updates policies periodically.	Agency implements and fully automates enterprise- wide identity policies for all users and entities across all systems with continuous enforcement and dynamic updates.

ANTICIPATED END OF LECTURE 11