|  |
| --- |
| **DoD ZT capabilities**: User, Device, Application & Workload, Data, Network & Env, Automation & Orchestration, Visibility & Analytics; **ZT maturity journey**: Traditional, Initial, Advanced, Optimal; Z is all about protecting data (digital assets); mapping data flows is critical; **Data flow mapping maturity**: Preparation (identify data locations) – Basic (map physical & logical data flows; mapping at this level) – Intermediate (complete list of apps & identified critical data flows, some automation implemented) – Advanced (complete inventory of data flows, Automation monitors controls, mitigates issues); Identity is the foundation of ZT; **ICAM**: Identity, Credential, Access management (set of tools, policies, systems to enable right individual to access right resource, at right time, for right reason, in support of fed business obj); 3 core ICAM capabilities: IdP (**Identity Provider**, system that directly authenticates entities based on credentials & issues assertions w/ attributes derived from creds), MUR (**Master User Record**, data repository of all entitlements that are granted), AAP (**Automated Account Provisioning**, Identity governance services like user entitlement mgmt., business role auditing & enforcement, account creation/deprovisioning); **Adaptive authentication**: Adjust level of authn based on context factors & real-time risk assessment; key factors: Context analysis, Risk assessment, MFA integration, Continuous monitoring; **Dynamic authorization**: real-time decision making, context-aware access control, policy-based control, attribute-based access control, integration with IAM; Context: Directory attributes, MFA attributes, Device attributes, Metadata; **ZT orchestration**: Trust identity? Trust device? Trust context? Value of resource? Risk of location? Centralized identity policy may not be feasible or desirable (like one ring to rule all); **Business risks**: SPOF, single breach gives away everything, sub-optimized (no one gets everything they want), vendor lock, standards lock, user resistance; **Federation is the solution**: Flexibility (more external partners to play), supports M&A, less resource intensive, easier for buy-in, less vendor lock; **Role-Based Access Control (RBAC) is insufficient**: RBAC employs static, simple logic, relies of apps to make access decisions, doesn’t take real-time context into consideration; Hence, we must choose: Fine Grained authz w/ attribute based access control: look at as many attributes as possible, use attributes to determine risk of each access attempt, adjust access permissions as needed; **ABAC levels**: Coarse grained (app-based); Medium grained (URL based); Fine grained (conditional / context based); ABAC is better than RBAC: increase security & flexibility, deliver better customer experience, regulatory compliance, increase agility; Cloud ≠ SaaS; **Can ABAC help with CICD**: yes! Tools for architecture planning: Analytical tools (current state assessment, risk assessment, threat modeling, technical validation), Informational tools, Modeling and design tools; **Effectiveness**: Do security measures in use provide risk reduction value? **Maturity**: Are processes that support existing measures reliable and resilient? **Efficiency**: Are security measures in place cost-effective relative to their value? |
| **AI based phishing on the rise**: Spear Phishing, Voice phishing (vishing); Cyber Kill Chain: Reconnaissance (harvest emails, info), Weaponization (Couple exploit w/ backdoor), Delivery (Send weaponized bundle to victim), Exploitation (Execute code to exploit vuln), Installation (malware on asset), C2 (hands-on-keyboard, remote manipulation of victim), Action on Objectives (accomplish original goals); **Governance**: **Sarbanes-Oxley**: Ensure reliability of financial reporting; IT general controls published part of **SOX** compliance; **HIPPA**: protects privacy of protected health info (PHI) – health insurance portability & accountability act; **FERPA**: Family educational rights & privacy act (student data); **FISMA**: Fed Info Security Mgmt Act (NIST security controls); **NIST docs**: SP 800-37: Risk mgmt. framework; SP 800-53: Sec controls for Fed IT systems; SP 800-82: Sec controls for ICS; SP 800-171: Sec controls for nonfederal systems processing controlled unclassified info; **PCI-DSS**: 2006 founded by Amex, Discover, JCB International, MasterCard, Visa; merchants & financial institutions understand & impl stands; Vendors for creating secure payment solns; **GDPR**: 25/May/2018; protection of privacy data in system design, impl, ops; Exemption for journalistic, academic, artistic, or literary expression; **Right to be forgotten**: given citizens right to have their data being corrected or deleted from internet; Companies to respond to data requests within 30 days; **Security Program**: Infra protection, App Sec, Sec Ops, Governance, Project/Program mgmt., Buss Partners; **Value of adopting standard**: Structure, baseline (to measure), help identify gaps, innovation (expand from baseline), roadmap (improvement/maturation); **CMMI**: 1 (Initial, ad hoc, little or no process); 2(Managed, repeatable, process absent); 3(Defined, work done, documented process, standardized); 4(Quantitatively managed, repeatable, std process, performance metrics are collected); 5(Optimized, virtuous cycle of improvements); CMMC Model 1.0: 1(Basic, 17 practices, third-party assessment), 2(Intermediate, 72 practices, 2 maturity process, no assessment), 3(Good, 130 practices, 3 process, 3rd party assessment), 4(Proactive, 156 practices, 4 process, no assessment), 5(Advanced, 171 practices, 5 process, 3rd part assessment); CMMC Model 2.0: 1(Foundational, 17 practices, annual self-assess), 2(Advanced, 110 practices by NIST SP 800-171, Triennial 3rd party assessment for critical national sec info; self-assess annually), 3(Expert, 110+ practices by NIST SP 800-172, Triennial gov-led assessment); |
| Risk **ALE, ARO, SLE**; ALE = ARO x SLE; **Risk assessment utility theory**: data-driven approach can reduce risk of vias on decision making; structed approach to solve cyber risk estimation; assign values to outcomes of decision, when randomness is involved; **Decision Trees**: Consequences are at ends of branches; sum of cost/earnings of path; chose only one alternative after each decision node (all alternatives should be defined explicitly), outcome of a chance must be complete; decision trees represent all possible future scenarios; **Expected Value** (EV): Probability x impact; **Threat modeling**: activities for identifying threats, defining countermeasures to prevent/mitigate effects of threat to system; threat sources + attack vectors + targets; **Approach**: Assess scope – Identify what can do wrong – Identify countermeasures – Assess your work; **Steps**: Decompose application – Determine & rank threats (DREAD) – Determine countermeasures & mitigation – **Attack tree** (systematic approach to model what can go wrong via what vuln), STRIDE – Technical validation (logs, scan results, metrics) – Informational tools (**effectiveness measures** [vuln scan reports, s/w mgmt. agent logs, OS tools, system logs, FIM, patch mgmt. / config tools], **economic measurements** [procurement, impact on infra, impl, sustainment / licensing, staffing impacts, buss process impacts], risk measurements); **NIST hierarchy of Risks (triangle)**: Tier-1 (Org), Tier-2 (mission / buss processes), Tier-3 (Info systems); Triangle base: Tactical risk, Triangle right side: Inter-tier & intra-tier comms, Feedback loops for improvements; Triangle left side: Traceability & Transparency of risk based decisions, Org wide risk awareness; **Risk = probability and impact / Damage**; low, medium, high, critical, heatmap; **Discrete vs Continuous Risk rating**: Discrete (easy to understand, compare, triage, provide discrete risk score, good for multiple options; no granularity to triage/prioritize, difficult to quantify values, get hung up on %, figures lie); Continuous: (realistic approach, get stakeholder buy-in, use in executive reports, dashboards, Intuitive, re-assess & eliminate outdated risks, too much interpretation, emotional arguments/ties to soln, subjective nature, difficult to compare); Story telling is critical; build a storyboard; **Architecture modeling and design tools**: Archi, Microsoft TMT, SysML, Basic data flow diagrams, Cameo; Operational view point with UML diagram; Data flow diagram; Freemind, Mindmup; **Collaboration tools**: Evernote, Teams, Slack, Google docs, Jira, kanban boards; |
| **Security Controls**: Doesn’t document HOW you must do something, doesn’t address WHY an org should do a thing: Controls are security capabilities that provide idealized security-relevant strategy (**playbook**); **Enterprise blueprints**: capture architecture vision (identify elements), document vision (create design); TOGAF, DoDAF, NAF; **Architecture principles**: **Key stakeholders** (beneficiaries, decision makers); **Key resources** (personnel, budget, how to engage, how to get access to other teams, SMEs); **ASARP**: Perf/availability – Security (confidentiality/Integrity) – TCO ($, people resources, training); **key principles**: what approach to guide our design, what we hope to accomplish (TOGAF); NAME – STATEMENT – RATIONALE – IMPLICATIONS; Set Scope; Desired (target) state; Assets: Threats (SDL): Define sec reqs – create app diagram – identify threats – mitigate threats – validate threats are mitigated; **4-diemnsional chess**: (Assets \* threats = Risk) \* f(time); Threat actors change TTPs, technology changes, press releases, value of data changes over time, Documentation; Create a program; Roadmap (charter or goal statement, scope, principles, goals & outcomes, transition mileposts, proj mgmt. artifacts, impacts); **Architecture definition**: Roadmap + Architecture models, gap analysis, transition architecture; **Architecture model**: Buss model, data model, app model, technology; **Reference architecture:** DoD ZTA; **Application Architecture Blueprints**: App design considerations, Life cycle models, waterfall projects, agile projects, devsecops projects, app sec design; **Security architect role**: ensure s/w is robust, resilient to attack, protects any app data that is stored or processed; Influence software deisgn & impl; secure s/w design, coding standards; Influence env in which s/w be used: Containers/VM/IaC, Microsegmentation, Data centric security; **Waterfall**: Req gathering (buss req, user req, func req, transition or temporal reqs, Non-func son or system req), design, implementation, verification, Maintenance, User acceptance testing, Functional testing, Performance testing; **Agile**: set of principles; Adaptive proj mgmt. technique that involes breaking project into phases & emphasize collaboration & improvement; **Scrum**: Agile project mgmt. framework that helps teams structure, manage work thru set of values, principles, practicesl Encourages teams to learn thru experience, self-organize, reflect on wins & losses to improve; **User story**: team divides work to be done into func increments that follow format as “As a .. I want .. So that .. “; **Epic**: Large user story that cannot be delivered as defined within single iteration or split into smaller user stories; **Requirements**: User stories each “*shall statements*” for lunch; **Minimum Marketable Feature (MMF)**: small, self-contained feature quickly developed, delivers significant value to user; **Minimum Viable Product (MVP)**: Version of new product for team to collect max validated learning about customers with least effort; **Product backlog**: list of new features, changes to existing features, bug fixes, infra changes, other activities that team may deliver to achieve specific outcome; **Sprint**: short, time-boxed period when scrum team works to complete set task; **Sprint planning**: collaboration event where team answers: What work can be done in this sprint and how chosen work gets done?”; **Daily scrum**: team checks in about work progress; Surface any blockers / challenges that impact ability to deliver; **Sprint review**: team demonstrates what they completed during spring review; showcase work to stakeholders before production; **Sprint retrospective**: identify areas of improvement for next sprint to be ready for next cycle; **Kanban**: Framework for real-time communication of capacity, full transparency of work; visually represented work items, allows team members to see at any time; “visual signal”; **Conception phase**: Identifying buss opportunities: drivers behind needed to develop app in first place (explore market niche & enduser pain points, itemization of buss needs); **Establishing context and assumptions**: determine scope of effort, context to operate, assumptions that impact how it is developed (build or buy decision), identify hidden drivers, data points that influence work; **Evaluating viability**: Work is technically, operationally, financially viable; |
| **Risks of using AI**: Bias, Cyber threats (phishing emails), data privacy, environment impact, IP infringement, Lack of explainability, transparency, misinformation, manipulation, lack of context, fast, confident but wrong answers, privacy concerns (no explicit federal law), operational concerns (model collapse), Data poisoning (bias introduced, low accuracy, precision, recall, system failure or exploitation); **DevOps**: Methodology in s/w dev & IT industry that integrates / automates work of soft dev and IT ops as means for improving & shortening dev life cycle; **DevSecOps**: integrating security testing at every stage of s/w dev process; SAST, DAST; AppSec design, App threat modeling, Identify App flow / integration points, create DFD, governance, intelligence (attack models, security features & design, stds & requirements); Implementation strategy: 5 Ws and 1 H; **Future proofing**: establish virtuous cycle (**PLAN-DO-CHECK-ACT**) of continuous improvement, monitor for env (buss, tech, req) & external changes (threat changes, industry shifts, competitive landscape/market pressure); |
| Effectiveness: Do the security measures in use provide risk reduction value?  Maturity: Are the processes that support existing measures reliable and resilient?  Efficiency: Are the security measures in place cost-effective relative to their value?  ISACA: The ISACA Risk IT Framework proposes a scenario-driven approach to risk identification. Under this model, risk scenarios are created either through a "top-down" approach looking in-depth at business goals to find the most germane sources of disruptions to those goals, or a "bottom-up" approach whereby generic scenarios are used as a starting point for the creation of a more customized view based on the organization under analysis.  ISO: ISO/IEC 31010:2009 outlines three possible methods for risk analysis: evidence-based (that is, checklists, historical methods), "systematic processes" (details unspecified), and inductive reasoning.  The Factor Analysis of Information Risk (FAIR) model, for example, introduces concepts such as threat capability, control strength, and primary and secondary loss factors, as well as introduces new ways to measure and model risk.  Threat sources: What are the threats and how will they behave?  Attack vectors: How will an attack occur or a threat scenario play out?  Targets: What are the most compelling targets?  The Open Group Architecture Framework (TOGAF) calls the Architecture Vision and what the Sherwood Applied Business Security Architecture (SABSA) calls the Conceptual Security Architecture.  The TOGAF Architecture Development Method (ADM)  Architecture definition process  • Address stakeholder security concerns  • Define secure functionality for the system level  • Develop views, viewpoints, and models  • Set security domains, boundaries, and interfaces  • Identify security-relevant system elements  • Align the architecture with security requirements  • Ensure any enabling services for security are available  • Trace architecture to stakeholder security requirements  • Identify security-related metrics  Architecture definition  • Scope  • Goals, objectives, and constraints  • Architecture principles  • Baseline architecture  • Architecture models (for each state to be modeled)  • Rationale and justification for the architectural approach  • Mapping to the architecture repository  • Gap analysis  • Impact assessment  • Transition architecture |