NIST guidelines and controls provide guidance to private, public, and non-profits for organizations to achieve across multiple controls a secure posture.

NIST requests positions for delivery NIST guidelines for the following:

1. *Criteria for designating "critical software."* Functional criteria should include, but not be limited to, level of privilege or access required to function, integration, dependencies, direct access to networking and computing resources, performance of a function critical to trust, and potential for harm if compromised. See EO Section 4(g).
2. *Initial list of secure software development lifecycle standards, best practices, and other guidelines acceptable for the development of software for purchase by the federal government.* This list of standards shall include criteria and required information for attestation of conformity by developers and suppliers. See EO Section 4(e)(i, ii, ix, and x).
3. *Guidelines outlining security measures that shall be applied to the federal government's use of critical software,* including but not limited to, least privilege, network segmentation, and proper configuration. See E.O. Section 4(I).
4. *Guidelines for software integrity chains and provenance.* See EO Sections 4(e)(ii, vi, and viii).

Problem: Adherence to NIST controls is reactive. Detection / reactive NIST control tools identify NIST controls after the fact vs building and demonstrating NIST guidelines and controls built in from start

Over roughly the past 20 years, to achieve NIST guidelines and implement controls organizations are using manual tools that provide reactive, after the fact control visibility at a point in time for NIST controls. This detection / reactive approach to demonstrate best practice, regulations and other controls such as NIST guidelines and controls, is not feasible, scalable in today's complex environment and creates a black box for security attack surfaces and NIST control management.

The detection tools using reactive - after the fact approaches are vulnerability scanning i.e., agent-based scanning, data and event correlation tools i.e., SIEM, searching that identify vulnerabilities and now ML ops that predict vulnerabilities. All these detection / reactive tools have the same common large challenge by identifying the vulnerability and organizations are to manually remediate, and this includes NIST controls. Organizations using the reactive tools have a consistent problem that continuous to grow - large unmanageable attack surfaces, the inability to effective control our environments and manage our NIST posture and risk.

Recommended solution: Require the "Secure the software supply chain" to build in NIST controls from start and demonstrate repeatable, building control outcomes vs. detection/reactive tools.

As organizations have more complex environments, are moving to the cloud and are using new technologies, “smart building” technology should be required to build in NIST controls to demonstrate controls are built in from the start vs. using reactive tools of always trying to detect NIST controls to meet guidelines and remediate.

Moving away from using reactive tools and focusing on “smart building” capabilities to deliver NIST guideline controls, eliminates the complexities, delivers consistent, repeatable NIST control postures. This new “smart building” approach delivers a preventative, fast, secure, and at scale NIST controls from the start while reducing the security attack surface in minutes and maintains control of the NIST security posture at all times.

NIST and federal organizations should mandate the preventative NIST guidelines for “Build Assurance”. This new guideline is for adherence for demonstrating, repeatable building process for NIST controls execution assurance and building audits to demonstrate that preventative NIST guidelines are achieved from the start.

NIST requirements for “smart building solution” to demonstrate the software supply chain, critical guidelines are delivered continuously from the start, controlled versus depending on reactive tools to scan, fix, predict or search after. This allows clear distinction between the preventative and detection / reactive solution.

Recommended NIST guidelines start in the overall software supply chain but are also required in the software development lifecycle, integrity chains and selecting critical software to deliver secure software development lifecycle standards, best practices as each of these guidelines are dependent on each other. An overall **“NIST Building Control Assurance” guideline** and control should all have consistent requirements for building guidelines to be enforced for a preventative approach to be delivered as the following:

- Identifying Software supply chain process to identify the process from prevent (build) vs reactive requirements across people, process and tools.
- Requiring **“NIST Building Control Assurance” guidelines and controls**
    - NIST controls to be built in from the start
    - Demonstrate and delivered controls in a repeatable, secure, building control process to achieve NIST controls real time
    - Tools required: Smart building solutions
    - Not valid tools: Detection tools such vulnerability scanning, SIEMS, or ML ops (reactive, after the fact).
- **“Built NIST Controls Output”** are to be demonstrated, controlled and visible real time.
    - Output is repeatable, all controls are built in and achieve adherence to NIST controls from the start
    - Each build output is controlled and not dependent on detection tools to identify if NIST controls are in adherence
- Use software that builds and demonstrates NIST controls are preventative such as “smart building” solutions.

- Detection NIST solutions are categorized and recognized as detection and reactive technology that can validate but do not provide NIST guideline building control assurance or Built NIST controls output audits.
- Reactive tools are vulnerability scanning, ML Operations, search, discovery tools are not valid solutions for preventative, NIST controlled approach.