US Executive Order calls for encrypting DNS

EO14144 January 17, 2025

Vicky Risk, vicky@isc.org

Outline

- History – of the prior work
- What's in this EO
  - 9 topics, best practices ++
  - Supply chain, securing existing systems, protecting communications
- Implementation Timeline
- Speculation (no point in asking me questions!)

A brief history

EO 14028 – summary from GSA.gov

- Requires service providers share incident and threat information
- Moves the Federal government to secure cloud services, zero-trust architecture, and mandates deployment of MFA and encryption
- Establishes baseline security standards for development of software sold to the government, including requiring developers to maintain greater visibility into their software and making security data publicly available.
- Creates a standardized playbook for cyber incident response
- Improves the ability to detect malicious cyber activity on Federal networks by enabling a governmentwide endpoint detection and response system and improved information sharing within the Federal government.
- Creates cybersecurity event log requirements
- Requires amendments to the FAR to align with requirements in the EO.

https://www.gsa.gov/technology/it-contract-vehicles-and-purchasing-programs/information-technology-category/it-security/executive-order-14028

NIST SSDF 1.1 - *Recommendations for Mitigating the Risk of Software Vulnerabilities*

Organizations should ensure that their people, processes, and technology are prepared to perform secure software development.

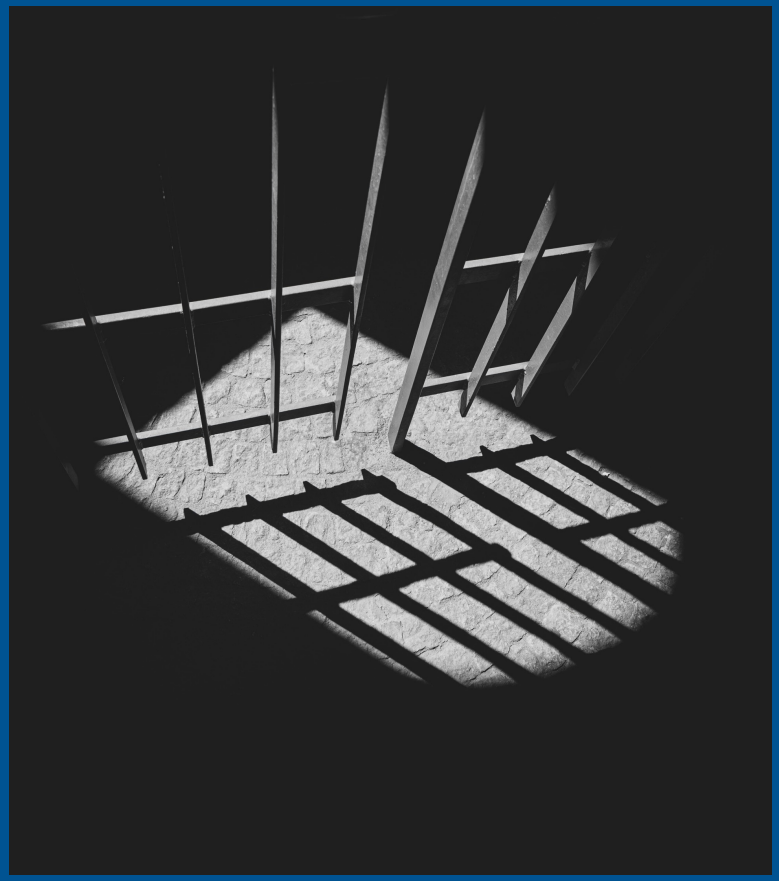
Organizations should protect all components of their software from tampering and unauthorized access.

Organizations should produce well-secured software with minimal security vulnerabilities in its releases.

Organizations should identify residual vulnerabilities in their software releases and respond appropriately to address those vulnerabilities and prevent similar ones from occurring in the future

Practices	Tasks	Notional Implementation Examples	References
Configure the Compilation, Interpreter, and Build Processes to Improve Executable Security (PW.6): Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities before testing occurs.	**PW.6.1**: Use compiler, interpreter, and build tools that offer features to improve executable security.	**Example 1**: Use up-to-date versions of compiler, interpreter, and build tools. **Example 2**: Follow change management processes when deploying or updating compiler, interpreter, and build tools, and audit all unexpected changes to tools. **Example 3**: Regularly validate the authenticity and integrity of compiler, interpreter, and build tools. See PO.3.	**BSAFSS**: DE.2-1 **BSIMM**: SE2.4 **CNCFSSCP**: Securing Build Pipelines—Verification, Automation **EO14028**: 4e(iv), 4e(ix) **IEC62443**: SI-2 **MSSDL**: 8 **SCAGILE**: Operational Security Task 3 **SCFPSSD**: Use Current Compiler and Toolchain Versions and Secure Compiler Options **SCSIC**: Vendor Software Development Integrity Controls **SP80053**: SA-15 **SP800161**: SA-15
	PW.6.2: Determine which compiler, interpreter, and build tool features should be used and how each should be configured, then implement and use the approved configurations.	**Example 1**: Enable compiler features that produce warnings for poorly secured code during the compilation process. **Example 2**: Implement the "clean build" concept, where all compiler warnings are treated as errors and eliminated except those determined to be false positives or irrelevant. **Example 3**: Perform all builds in a dedicated, highly controlled build environment. **Example 4**: Enable compiler features that randomize or obfuscate execution characteristics, such as memory location usage, that would otherwise be predictable and thus potentially exploitable. **Example 5**: Test to ensure that the features are working as expected and are not inadvertently causing any operational issues or other problems. **Example 6**: Continuously verify that the approved configurations are being used. **Example 7**: Make the approved tool configurations available as configuration-as-code so developers can readily use them.	**BSAFSS**: DE.2-3, DE.2-4, DE.2-5 **BSIMM**: SE2.4, SE3.2 **CNCFSSCP**: Securing Build Pipelines—Verification, Automation **EO14028**: 4e(iv), 4e(ix) **IEC62443**: SI-2 **IR8397**: 2.5 **MSSDL**: 8 **OWASPASVS**: 14.1, 14.2.1 **OWASPMASVS**: 7.2 **PCISSLC**: 3.2 **SCAGILE**: Operational Security Task 8 **SCFPSSD**: Use Current Compiler and Toolchain Versions and Secure Compiler Options **SCSIC**: Vendor Software Development Integrity Controls **SP80053**: SA-15, SR-9 **SP800161**: SA-15, SR-9 **SP800181**: K0039, K0070
Review and/or Analyze Human-Readable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements (PW.7): Help identify vulnerabilities so that they can be corrected before the software is released to prevent exploitation. Using automated methods lowers the effort and resources needed to detect vulnerabilities. Human-readable code includes source code, scripts, and any other form of code that an organization deems human-readable.	**PW.7.1**: Determine whether code *review* (a person looks directly at the code to find issues) and/or code *analysis* (tools are used to find issues in code, either in a fully automated way or in conjunction with a person) should be used, as defined by the organization.	**Example 1**: Follow the organization's policies or guidelines for when code review should be performed and how it should be conducted. This may include third-party code and reusable code modules written in-house. **Example 2**: Follow the organization's policies or guidelines for when code analysis should be performed and how it should be conducted. **Example 3**: Choose code review and/or analysis methods based on the stage of the software.	**BSIMM**: CR1.5 **EO14028**: 4e(iv), 4e(ix) **IEC62443**: SM-5, SI-1, SVV-1 **NISTLABEL**: 2.2.2.2 **SCSIC**: Peer Reviews and Security Testing **SP80053**: SA-11 **SP800161**: SA-11 **SP800181**: SP-DEV-002; K0013, K0039, K0070, K0153, K0165; S0174
	PW.7.2: Perform the code review and/or code analysis based on the organization's secure coding standards, and record and triage all discovered issues and recommended remediations in the development team's workflow or issue tracking system.	**Example 1**: Perform peer review of code, and review any existing code review, analysis, or testing results as part of the peer review. **Example 2**: Use expert reviewers to check code for backdoors and other malicious content. **Example 3**: Use peer reviewing tools that facilitate the peer review process, and document all discussions and other feedback. **Example 4**: Use a static analysis tool to automatically check code for vulnerabilities and compliance with the organization's secure coding standards with a human reviewing the issues reported by the tool and remediating them as necessary. **Example 5**: Use review checklists to verify that the code complies with the requirements. **Example 6**: Use automated tools to identify and remediate documented and verified unsafe software practices on a continuous basis as human-readable code is checked into the code repository.	**BSAFSS**: TV.2, PD.1-4 **BSIMM**: CR1.2, CR1.4, CR1.6, CR2.6, CR2.7, CR3.4, CR3.5 **EO14028**: 4e(iv), 4e(v), 4e(ix) **IDASOAR**: 3, 4, 5, 14, 15, 48 **IEC62443**: SI-1, SVV-1, SVV-2 **IR8397**: 2.3, 2.4 **ISO27034**: 7.3.6 **MSSDL**: 9, 10 **NISTLABEL**: 2.2.2.2 **OWASPASVS**: 1.1.7, 10 **OWASPMASVS**: 7.5 **OWASPSAMM**: IR1-B, IR2-A, IR2-B, IR3-A **PCISSLC**: 3.2, 4.1 **SCAGILE**: Operational Security Tasks 4, 7; Tasks Requiring the Help of Security Experts 10

Secure Software Development Attestation



- March 11, 2024, CISA published the SSDA form and repository for the forms
- Forms must be signed by CEO or designate
- **Criminal penalties** for false or misleading statements
- Repository is behind a Federal login (so outsiders can't see what is there)
- Mentions artifacts must be available, but without any details

Secure Software Development Attestation

1. Sw is developed and built in a secure environment
2. “good-faith effort to maintain trusted source code supply chains”
3. Automated testing, check for vulnerabilities, vulnerability handling process

Secure by Design Pledge

- Multi-factor authentication
- No default passwords
- Reduce classes of vulnerabilities
- Increase uptake of security patches
- Publish vulnerability disclosure process
- Transparent CVE reporting
- Facilitate intrusion detection (e.g. logs)

Photo by Darius Bashar on Unsplash

EO 14144 – Jan 17, 2025

Strengthening and Promoting Innovation in the Nation's Cybersecurity

The sections of the document are

1.Policy.

2. Operationalizing Transparency and Security in Third-Party Software Supply Chains.

3. Improving the Cybersecurity of Federal Systems.

4. Securing Federal Communications.

5. Solutions to Combat Cybercrime and Fraud.

6. Promoting Security with and in Artificial Intelligence.

7. Aligning Policy to Practice.

8. National Security Systems and Debilitating Impact Systems.

9. Additional Steps to Combat Significant Malicious Cyber-Enabled Activities

2. Software Supply Chain

1. Require SSD attestations, with artifacts (proof) and a list of Federal customers
2. Implement … SSDF
3. Deploy patches securely
4. Supply chain risk management
5. Open source standards, recommendations (contributing…)

3. Cybersecurity of Federal Systems

1. Improve identity and access management
   - Pilot e.g. phishing-resistant authentication options
2. Collect and share threat information
3. Requirements for cloud service providers
4. Space systems

Collecting and sharing and USING threat information

- Lengthy section
- Many considerations, including collaboration, protecting sensitive data, more timely sharing of information
- One of the sections with the longest timeframe to implement

Cyber security in space

4. Securing Federal Communications

Implement … strong identity, authentication and encryption

Best Practices ++

BGP Security

ROA

Ensure your IP addresses and AS numbers are appropriately registered

Publish ROAs with ARIN

Require ISPs to publish ROAs and do route validation filtering

… updated guidance on route leak mitigation, and source-address validation

Require Encrypted DNS

”…any product that acts as a DNS resolver (whether client or server) for the Federal Government <should> **support encrypted DNS”**

Does not specify DOH or DoT or DoQ

Enable Encrypted DNS

"Within 180 days of the date of this order, FCEB agencies shall enable encrypted DNS protocols wherever their existing clients and servers support those protocols"

Again, no specifics

Encrypt email

Photo by Daria Nepriakhina 🇺🇦 on Unsplash

- Encrypt email messages end to end
- Enforce encrypted and authenticated transport for mail client- server communications
- Require *at least* encrypted transport for voice and video conferencing and move towards E2E encryption

Cryptography threats

- Prepare for PQC
- Protect key material (HSMs)

Photo by Jason Dent on Unsplash

There is more

1.Policy.

2. Operationalizing Transparency and Security in Third-Party Software Supply Chains.

3. Improving the Cybersecurity of Federal Systems.

*4. Securing Federal Communic**ations.***

5. Solutions to Combat Cybercrime and Fraud. (digital ids - mobile drivers' licenses)

6. Promoting Security with and in Artificial Intelligence. (using AI to detect vulnerabilities and automate cyber defense)

7. Aligning Policy to Practice. (implementation: reviewing, updating minimum cybersecurity requirements)

8. National Security Systems and Debilitating Impact Systems. (exemptions for NSS, separate process for critical systems, including space systems)

9. Additional Steps to Combat Significant Malicious Cyber-Enabled Activities (freezing assets of cyber criminal organizations)

Implementation timeline (partial)

April 2025	May 2025	July 2025	October 2025	January 2027	January 2030
•Update SSDF •Minimum cybersecurity standards for contractors •Begin preparing for PQC •Best practices such as HSMs •FedRAMP cloud security incentives •IP addresses registered with RIRs	•Standards for open source patching, contributions •Enforce encrypted transport for email •Publish BGP ROAs	•Enable encrypted DNS where supported •Civil space FAR requirements •Require, asap adoption of TLS 1.3	•Update FAR to require support for encrypted DNS •Practical implementation guidance for digital IDs •Est program for AI in cyber defense •Cloud service access token security •FAR requires ROAS from ISPs •RFC on Cyber security best practices	•require the Cyber Trust Mark for consumer IOT devices purchased by the US Govt	•Deadline to complete transition to TLS 1.3 (!)

Summary

Supply Chain	Use FAR (federal purchasing system) requirements to encourage software supply chain transparency
Federal Systems	Implement recognized best practices for improved cybersecurity.
Communications	Secure Federal Communications, including routing, and encrypt DNS, email and voice/video.
Defend	Enhance cyber security & identity systems across Federal agencies, organize to investigate threat data

References

- EO 14144 (https://www.federalregister.gov/documents/2025/01/17/2025-01470/strengthening-and-promoting-innovation-in-the-nations-cybersecurity#h-1)
- EO 14028 (https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity)
- CISA Secure Software Development Attestation (https://www.cisa.gov/secure-software-attestation-form)
- CISA Secure by Design Pledge (https://www.cisa.gov/securebydesign/pledge)
- CISA Zero Trust Maturity Model (https://www.cisa.gov/zero-trust-maturity-model)
- NIST Special Publication 800-218 (*Secure Software Development Framework* (SSDF) https://csrc.nist.gov/pubs/sp/800/218/final
- NIST Special Publication 800-53 (*Security and Privacy Controls for Information Systems and Organizations*) https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final