Improving Security of Open Source Software in Operational Technology and Industrial Control Systems

OVERVIEW

The Cybersecurity and Infrastructure Security Agency (CISA), Federal Bureau of Investigation (FBI), National Security Agency (NSA), and U.S. Department of the Treasury are releasing this fact sheet for senior leadership and operations personnel at operational technology (OT) vendors and critical infrastructure facilities. This fact sheet will assist with better management of risk from OSS use in OT products and increase resilience using available resources. While several resources and recommendations within this fact sheet are best suited for execution by the vendor or the critical infrastructure owner, collaboration across parties will result in less friction for operator workflows and promote a safer, more reliable system and provision of National Critical Functions.

This fact sheet aims to:

- Promote the understanding of open source software (OSS) and its implementation in OT and industrial control systems (ICS) environments (hereinafter referred to as "OT").
- Highlight best practices and considerations for the secure use of OSS in OT.

CISA's OSS Initiative

In 2023, CISA's Joint Cyber Defense Collaborative (JCDC) initiated a collaborative planning effort to support the awareness, security, and cyber resiliency of OSS in critical infrastructure OT. This effort is one of the priority initiatives within the JCDC 2023 Planning Agenda, which consists of contributions from JCDC participants, including industry partners and representatives from OSS foundations. Consistent with JCDC's approach to bringing together public and private partners in development of joint cyber defense plans, this fact sheet benefitted from input by industry contributors, including Accenture, Claroty, Dragos, Fortinet, Google, Honeywell, Microsoft, Nozomi Networks, NumFOCUS, OpenSSF / Linux Foundation, Rockwell Automation, Rust Foundation, Schneider Electric, Schweitzer Engineering Laboratories, Siemens, and Xylem. Organizations can reference the Securing Open Source Software in Operational Technology web page for an overview of the OSS planning initiative, goals, and additional deliverables.

CISA recognizes the benefits of open source software in enabling software developers to work at an accelerated pace and fostering significant innovation and collaboration. With these benefits in mind, this planning effort complements the CISA Open Source Software Security Roadmap, which defines how CISA will work to enable the secure use and development of open source software, both within and outside of the federal government.

SAFETY AS A PRIORITY

In OT, both cybersecurity and safety concerns are heightened due to the potentially far-reaching impacts of incidents and associated life safety implications, specifically to connected infrastructure. Widely accepted cyber hygiene practices, such

as updating software in IT systems when a patch for a vulnerability is available, may be challenging when an underlying OSS library needs to be updated. Updating software in OT may be additionally challenging because of the potential adverse effects on other (dependent) software and potential operational risks. Implementing "secure-by-design" and "-default" approaches can help decrease these cybersecurity and safety risks in OT.

Open Source Software in Operational Technology

Open source software[1]—often referred to as OSS—is software with an open license for anyone to view, use, study, or modify, and is distributed with its source code. Source code is the human-readable formal language that software developers use to specify the actions a computer will take. OSS serves as an example of open collaboration among many parties and is available for use, modification, and distribution. Among other benefits, OSS allows organizations with similar software needs to share progress, reduce overhead, and scale innovation. In contrast to OSS, **closed source (proprietary) software** refers to software that is developed, tested, and managed close hold. Proprietary software will also include valid, authenticated licenses for users using the software, which often come with restrictions. It is common for closed source software and security tools to make use of OSS. Much of the infrastructure and tools used to develop, build, and install a closed source software project contain OSS as well.

OT[2] is defined as the hardware, software, and firmware components of a system used to detect or cause changes in physical processes through the direct control and monitoring of industrial equipment, assets, processes, and events. While often interchangeable with OT, **ICS**[3] are a subset of OT where networks are comprised of information systems that control industrial processes, such as manufacturing, product handling, production, and distribution. The diverse way OSS can be integrated into OT products can make it difficult to know whether certain software modules, and their associated vulnerabilities, are present and/or exploitable. Additional challenges include an overall minimized opportunity to patch and increased aversion to new variables added into production environments because of the often stringent uptime requirements for OT environments.

Software Security Challenges

Security is a critical step at every phase of lifecycle management for both OSS and OT software on all systems. Some considerations are especially relevant or challenging due to features existing at the intersection of managing OSS and OT software. Examples of security concerns that OSS and OT share with all software systems include:

- **Dependency vulnerabilities.** Software often relies on other libraries and components. If these dependencies have vulnerabilities, they can introduce (cyber) risks into the software.
- **Lack of commercial support.** Many software packages come with limited or no service agreements to actively monitor components and manage vulnerabilities, or the expected useful lifetime of products is longer than the software service agreement. Being outside of a support window makes patching vulnerabilities extremely challenging. Open source software licenses typically provide the software without any warranty and disclaim all responsibility; commercial organizations often provide support for a fee.
- **Inadequate documentation.** Insufficiently documented software can be difficult for users to understand and use securely.

THE IT/OT CONVERGENCE

OT may theoretically be considered **mutable infrastructure**—capable of being updated following deployment—but in practice is often treated as **immutable infrastructure** and rarely maintained or upgraded. Some hardware or controllers may be immutable, but the software and data on them, including OSS, is mutable. Ideally governed by good change management, software can be entirely replaced, changed, or upgraded at any time. **However, in practice this is not always feasible due to the change management policies and safety regulations surrounding the introduction of new, updated code to a product.**

OT components are often connected to IT networks. Consequently, malicious actors can pivot from IT to OT networks to affect or interrupt system, device, and process operations. This highlights the need for securing systems at the IT level to reduce the likelihood of a threat actor pivoting to OT systems connected to IT infrastructure. As identified in the 2015

Ukrainian power grid compromise,[4] threat actors sent phishing emails targeting IT networks before pivoting to the ICS environment and deploying BlackEnergy malware.

The security of systems in both IT and OT environments should improve in tandem. Many aspects of Security-by-Design and -Default apply to both IT and OT. Some considerations are specific to either OSS, OT, or the conjunction of both. Conversely, some considerations are not specific to OSS but have OSS-specific implementations, such as Sigstore. Sigstore serves as an OSS-specific implementation of the broader security control of code signing; it enables developers to validate that the software in use is exactly what it claims to be by using cryptographic digital signatures and transparency log technologies. A desired security control with OT-specific considerations is multifactor authentication (MFA). It is considered an important best practice in both IT and OT environments; however, MFA implementation (such as using long, complex passwords paired with a second or multiple sources of validation) may be prohibitive in high-intensity safety scenarios. With OT devices, MFA should use technology compatible with OT operational modalities.

Supply Chain Risks

As a result of the connections between OT and IT networks, OT systems are too often exposed to cyber threat actors targeting control systems and the critical infrastructure they operate. To counter these threats, the cybersecurity community recommends that defenders and operators keep all OT and IT systems up to date with patches and security updates to address known exploited vulnerabilities. However, patching and security updates create opportunities for threat actors to affect the OT supply chain—malicious threat actors can compromise the supply chain by embedding malware in a patch or by compromising the website that hosts the patch, such as replacing the patch itself with malware (known as a 'watering hole'). This is particularly problematic since most OT operators inherently trust the legitimacy of these sites. For example, Havex malware used legitimate system update installers to deploy and execute the malware along with the normal software update process. The compromised OT platform was left fully functional, but with a malicious backdoor installed.[5]

Supply Chain Risk Management

The software supply chain is a complex issue for systems and poses specific risks when accounting for the intersection of OT and OSS, necessitating a thoughtful strategy for risk management. A reliable software supply chain for an OT system with OSS components provides assurance the system will behave as intended at the time of acquisition and that all OSS components have been appropriately vetted prior to use. This is also true for software supply chain information in general. The phrases "as intended," "at time of acquisition," and "prior to use" reference business or organizational expectations that will be specific to each individual enterprise and require definition of a particular component's use in the operational environment—components defined as either a commercial product or open source software. Two examples of supply chain risk management aspects that are relevant to OSS in OT are transparency and verifiability.

Transparency includes:

- What assets it owns and operates, e.g., asset management transparency.
- What software each software asset is composed of—a Software Bill of Materials (SBOM) can assist with this.
- The supplier's process by which, for example, an OT device's firmware will be updated.
- The software the assets are running is the software that the developer wrote, and that the developer who wrote the software is the intended developer.

Verifiability—or the ability to confirm the authenticity of information and data related to systems—includes:

- The identity of users and their access restrictions.
- Data integrity—the accuracy and validity of data throughout its lifecycle.
- Software is functioning as specified.
- Overall system security.

Each variation of verifiability contains independent means of achieving it, often with an overlapping and interrelated set of controls. In this sense, OT and OSS are similar to other software. By ensuring these components can be verified, confidence in a system's defense and its ability to mitigate malicious cyber activity is heightened. For additional resources

on assessing IT supply chain risk management, see CISA's ICT Supply Chain Risk Management Task Force.

RECOMMENDATIONS

This fact sheet provides recommendations for improving security of OSS in OT/ICS, starting at the senior leadership level of an organization. Best practice resources are also provided as considerations when addressing cybersecurity concerns pertaining to OSS in OT devices and ICS environments. CISA, FBI, NSA, and U.S. Department of the Treasury encourage organizations to review the National Institute of Standards and Technology (NIST) Guide to ICS Security for further guidance. The OT/ICS industry is encouraged to apply the below tools and best practices to address general problems surrounding the use of OSS, as well as to actively participate in instances where there are unique needs for these solutions.

Vendor Support of OSS Development and Maintenance

Open source software is often developed and maintained by volunteers. Providing support to individuals and groups that develop and maintain key open source projects helps elevate the security baseline and provides more assurance in the integrity of key libraries. Every organization using OSS should support the OSS ecosystem, including by:

- **Learning about and considering participation in OSS and grant programs.** The following resources can help support the development and maintenance of critical OSS projects that are used in OT/ICS systems. These can include security audits, efforts to fix identified problems, and/or improve processes of OSS used in OT.
    - **DigitalOcean Hacktoberfest:** An annual, worldwide event held during the month of October that encourages open source developers to contribute to repositories.
    - **Open Source Security Foundation (OpenSSF) Alpha-Omega Program:** Program that partners with OSS project maintainers to systematically find new, as-yet-undiscovered vulnerabilities in open source code (and get them fixed) to improve global software supply chain security.
    - **Free and Open Source Software (FOSS) Contributor Fund:** Framework for selecting open source projects that a company supports financially. This initiative is designed to encourage open source participation and help companies take an active role in sustaining the projects they depend on.
    - **NumFOCUS Small Development Grants Program:** Program that helps fund projects' usability, community growth, and speeding up the time to major releases.
- **Partnering with existing OSS Foundations and pursuing collaborative efforts** to leverage the ecosystem knowledge for more effective, direct funding and support to key projects critical to OT/ICS security.
- **Supporting the adoption of security tools and best practices in the software development lifecycle.** Integrating security at the early stage of the software development lifecycle is critical to producing software that is Secure-by-Design. Organizations contributing to OSS should invest development time and resources towards the adoption of critical security tools as part of a project's development lifecycle.
    - Google's Open Source Security Upstream Team effort is one example advocating for the adoption of these principles.
    - The GitHub Action for OpenSSF Scorecard serves as a check that a project is using current best practices to test for security vulnerabilities before production releases, as well as export provenance metadata to support end-to-end trust in the project's supply chain.
    - See additional recommendations in the OpenSSF Best Practices Working Group's Concise Guide for Developing More Secure Software.

Manage Vulnerabilities

Vulnerability management is important for all software, though OSS and OT have unique characteristics that require further consideration. Vulnerability management includes[6] processes for organizations to communicate and accomplish vulnerability discovery, analysis, and handling, as well as report intake, coordination, disclosure, and response. In each of these phases, using common vulnerability identifiers, including the production and consumption of vulnerability information in existing formats, can reduce confusion and simplify vulnerability management. Existing formats include Common Vulnerabilities and Exposures (CVE), Common Security Advisory Framework (CSAF), and Open Source

Vulnerability (OSV). This section highlights resources for vulnerable device detection, response, and vulnerability coordination.

Risk Exposure Reduction

CISA offers a range of services at no cost, including scanning and testing to help organizations reduce exposure to threats via mitigating attack vectors. CISA Cyber Hygiene services can help provide additional review of organizations' internet-accessible assets. Cyber Hygiene can detect vulnerabilities in internet-connected software, including in OSS and OT systems. Email vulnerability@cisa.dhs.gov with the subject line, "Requesting Cyber Hygiene Services" to get started.

Vulnerability Coordination

Advancing the security and resilience of ICS is one of CISA's top priorities. As part of CISA's mission to help critical infrastructure partners manage ICS security risk, CISA is committed to equipping the community with practices that address ICS risk and operational resilience. For example, CISA helps ensure ICS vendors can assign CVE IDs by assisting organizations to become a root CVE Numbering Authority (CNA). If there is no identifier for coordinating a new ICS vulnerability, CISA will assign one as the root CNA for ICS. Additional vulnerability coordination guidance and supporting resources include:

- **Organizations developing software, including OSS, should establish a Coordinated Vulnerability Disclosure (CVD) program.** The Software Engineering Institute's (SEI) CERT Guide to CVD provides an introduction to the key concepts, principles, and roles necessary to establish a successful process.
- **Individuals and organizations who discover vulnerabilties should report to the relevant developer.** For example, vulnerability finders might report to product owners, vendors, or project maintainers. In cases where contact with the developer cannot be made, the bug finder may report via CISA.
- **Organizations participating in CVD should identify key OSS used to assist in improving CVD programs where needed.** See OpenSSF's Guide to Implementing a Coordinated Vulnerability Disclosure Process for Open Source Projects, which is intended to help open source project maintainers create and maintain a coordinated vulnerability response process.
- **Contribute effort to support and encourage vulnerability research.** Improve the security of OSS projects by discovering, reporting, and helping to remediate vulnerabilities. Consider the following incentives:
  - Google's Open Source Software Vulnerability Rewards Program
  - HackerOne's Internet Bug Bounty and Community Edition
- **Utilize Stakeholder-Specific Vulnerability Categorization (SSVC) methodologies to inform response activities.** CISA and SEI have partnered to develop the SSVC system, which presents a systematic, decision tree-based approach to analyze and prioritize vulnerability response activities based on exploitation status, impacts to safety, and prevalence.
  - CISA's SSVC Guide
  - SEI's Prioritizing Vulnerability Response: A Stakeholder-Specific Vulnerability Categorization

Patch Management

Patch management is a process at the intersection of vulnerability management and change management. Patching is just one option for vulnerability remediation,[7] which occurs when a vulnerability is eliminated or removed. Mitigation, on the other hand, occurs when the impact of a vulnerability decreases without reducing or eliminating the vulnerability. Patching is a complex decision when considering and working with OT. Other forms of **remediation** (upgrading or removing the system) or **mitigation** (increasing network controls) can reduce the functionality of the affected device and alter alignment to organizational risk tolerances and priorities.

In some industries[8],[9],[10] patches may require regulatory approval for certain devices. For example, in some instances patches can move OT systems out of a state that has been previously certified and/or approved under certain regulatory or compliance frameworks. North American Electric Reliability Corporation Critical Infrastructure Protection (NERC CIP) is a set of standards that requires covered entities to weigh a variety of risk factors when making individual patching determinations, including the reliability of the patched system.[11] Amidst possible overlap with regulatory concerns,

restarting an OT system to apply a patch may have large business or operational costs. In these situations, mitigations should be applied immediately after the vulnerability is identified until a remediation, such as a patch, is approved.

While the OSS ecosystem is diverse, vulnerabilities are collectively aggregated in the National Vulnerability Database (NVD) and driven by organizations that serve as CVE numbering authorities. The procedure of CVE assignment and tracking is common practice in U.S. government policies and standards. Some open source organizations also use the OpenSSF OSV Schema as an aggregation tool for vulnerabilities in language ecosystems, while others have not yet adopted a community vulnerability naming and tracking benchmark. To support community application of security-relevant patches, all projects and organizations should use a community-recognized vulnerability naming method. These methods are further supported by structured formats for security alerts such as the OASIS CSAF.

Vendors and consumers are further encouraged to:

- **Promote the unique understanding of patch deployment processes for OT/ICS environments.** Communication for OT/ICS-specific patch implementation should include:
    - Safety and security of the customers as a core business requirement, not just a technical feature.
    - The assessment of what mitigations are immediately needed.
    - How decisions are made for balancing the risk due to a cybersecurity vulnerability, as well as risk due to changing the OT environment.
    - The turnaround time, for example, for how long a patch should take to deploy and what the confidence level is for correct implementation.
    - How to streamline software development processes with ICS vendors, that is, when vendors ship patch updates and consumers apply periodically without the added complexity of scheduling maintenance windows.
    - How software is being tested for compability issues before a patch is deployed.
- **Maintain a comprehensive updated asset inventory to best identify software and hardware products, as well as open source components in both IT and OT environments.** Identify vulnerabilities that need to be patched based on the asset inventory and automated correlation with vulnerability databases such as the NVD.
    - **For OT/ICS systems, SBOMs can provide an inventory of what is in use**, making it easier to determine whether a device is affected by a vulnerability due to use of an out-of-date OSS dependency. Organizations are encouraged to hold vendors and suppliers accountable for maintaining provenance data, for example, by requesting SBOMs prior to purchasing products. A SBOM can also help identify OSS projects that are widely used by or otherwise critical to ICS. Organizations are encouraged to request or require SBOMs from upstream suppliers at the time of procurement, as well as for products that are already owned.
    - Vulnerability Exploitability eXchange (VEX) **provides additional information on whether a product is impacted by a specific vulnerability** and, if affected, whether there are actions recommended to remediate. VEX is machine-readable, which enables automation and supports integration into broader tooling and processes; organizations can integrate component data from SBOMs with vulnerability status information from VEXes to provide an up-to-date view of the status of vulnerabilities.
- **Establish "emergency patching procedures"** that expedite patches for critical vulnerabilities without sacrificing safety and working outside of accepted standards.

See the following additional resources for best practice guidance that addresses the management of vulnerabilities:

- Global Cybersecurity Alliance (GCA): Manage Vulnerabilities in ICS Open Source Software
- Forum of Incident Response and Security Teams (FIRST): Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure

Improve Authentication and Authorization Policies

While not unique to OSS, effective implementation of **authentication**[12] and **authorization**[13] represent powerful protective controls that can be deployed in any networked computing environment and can significantly enhance the security and consumption of commercial products and OSS projects. Authentication—verifying identity—and authoritization—ensuring

appropriate access permissions—work in tandem to prevent unauthorized and malicious changes to IT and OT infrastructure.

However, these controls can be difficult to correctly implement and are especially important in OT environments, which are less likely to maintain defense-in-depth security controls once a threat actor breaches the network boundary and obtains initial access to the environment. Considering OT devices are often deployed in production environments for longer periods of time and are less likely to receive updates compared to enterprise devices, they may also be less likely to support the latest cryptographic technologies that facilitate highly secure authentication. Furthermore, although many OT communication protocols utilized by these devices have extensions or revisions that support modern authentication and authorization schemes, the actual uptake of "more secure" protocols is mixed. The shortage of experienced OT professionals required to maintain and administer a granular authentication and authorization program in OT environments can also prove especially challenging.

Within ICS, authentication and authoritization practices can improve by:

- Using accounts that uniquely and verifiably identify individual users. For example, OT products that leverage service accounts should use role-based access control (RBAC) or a similar approach.
- Avoiding use of hard-coded credentials, default passwords, and weak configurations.
- Implementing MFA (when applicable).
- Using centralized user management solutions (e.g., Lightweight Directory Access Protocol [LDAP], Active Directory [AD]), which can streamline account management and improve traceability. This should be weighted against availability requirements.

Combining Secure-by-Default practices with least privilege—or users only having access to what they need to perform their responsibilities—is an important consideration for addressing the authorization process. Increasing the resilience against exploitation via end-user compromise reduces the prevalence of successful incidents impacting OT.

Establish Common Framework

Improving the awareness and adoption of key cybersecurity best practices and infrastructure as they relate to both IT and OT environments can establish a common framework for using OSS. CISA has developed a performance-based checklist of key organizational cybersecurity goals, which are applicable to mixed IT/ICS network environments. Developed from NIST's Cybersecurity Framework (CSF), the CISA Cross-Sector Cybersecurity Performance Goals (CPGs) describe network segmentation, vulnerability patching, and software assurance goals organizations should strive to meet, irrespective of OSS involvement in a given system. Additionally, the following recommendations should be considered to ensure vendors provide components that meet industry standards for security compatibility with existing OSS tools, and a culture is established that addresses safety and cybersecurity concerns for critical systems:

- **Develop and support an Open Source Program Office (OSPO).** Organizations that heavily interact with or utilize OSS should consider a dedicated office for coordinating these tasks. An OSPO serves as the center of competency for an organization's open source operations and structure and is responsible for defining and implementing strategies and policies to guide these efforts.[14]
- **Support safe and secure open source consumption practices.** The following tools can assist:
  - The OpenSSF Scorecard serves as an automated tool to assess risks that dependencies introduce.
  - Supply-chain Levels for Software Artifacts (SLSA)'s framework serves as an actionable checklist to improve software security, assess upstream dependencies, and evaluate the trustworthiness of the artifacts consumed.
  - The Secure Supply Chain Consumption Framework (S2C2F) provides a guideline for any organization that is directly utilizing open source components (e.g., open source that is not a component within a commercial product) to do so in a secure manner.
  - MITRE's Hipcheck can be used within the secure consumption process to assess the risk of an OSS component before use.
- **Build a targeted list of OT/ICS-specific requirements.** A collection of industry partners have created a generic security checklist[15] that constitutes what makes a product minimally and viably secure. This checklist is not

specific to OT/ICS systems; hence, a more targeted list of security requirements specific to vendors supplying these systems is needed. This targeted list should include tools that are aligned with the common themes of transparency and verifiability.

- **Support the adoption of software signing techniques.** Software signing ensures the integrity of updates, network communications, and software distribution across environments. In conjunction, **using access transparency logs and identity-based signing** can provide auditable and tamper-resistant logging, allowing OT/ICS systems to verify the authenticity of software updates and patches.
- **Support the adoption of provenance generation.** Provenance for OT/ICS software can provide knowledge about where software came from and how it was built—in a verifiable manner. Provenance may assist ICS systems in tracking the source of software components, as well as verify they were created in accordance with established organizational policies.
- **Maintain a software asset inventory** to support the identification of what packages, software, firmware, and security services (e.g., incident and vulnerability management) exist in your environment.

RESOURCES

- CISA: JCDC 2023 Planning Agenda
- CISA: JCDC Planning - Securing Open Source Software in Operational Technology
- CISA: Open Source Software Security Roadmap
- CISA: Security-by-Design and -Default
- Sigstore
- CISA: ICT Supply Chain Risk Management Task Force
- NIST: Guide to ICS Security
- DigitalOcean: Hacktoberfest
- OpenSSF: Alpha-Omega Program
- FOSS Contributor Fund
- NumFOCUS: Small Development Grants Program
- Google: Open Source Security Upstream Team
- OpenSSF: GitHub Action for Scorecard
- OpenSSF Best Practices Working Group: Concise Guide for Developing More Secure Software
- CISA: Cyber Hygiene
- CISA: ICS
- CVE: Numbering Authorities
- CVE: Partner Details - CISA
- SEI: CERT Guide to CVD
- Report to CISA
- OpenSSF: Guide to Implementing a Coordinated Vulnerability Disclosure Process for Open Source Projects
- Google: Open Source Software Vulnerability Rewards Program
- HackerOne: The Internet Bug Bounty
- HackerOne: Community Edition
- CISA: SSVC Guide
- SEI: Prioritizing Vulnerability Response - A Stakeholder-Specific Vulnerability Categorization
- NIST: National Vulnerability Database
- OpenSSF: OSV Schema
- OASIS: CSAF
- CISA: SBOM
- CISA: Minimum Requirements for VEX
- GCA: Manage Vulnerabilities in ICS Open Source Software

- FIRST: Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure
- NIST: Cybersecurity Framework
- CISA: CPGs
- OpenSSF: Scorecard
- OpenSSF: SLSA
- OpenSSF: Secure Supply Chain Consumption Framework
- MITRE: Hipcheck

REFERENCES

[1] Open Source Initiative: The Open Source Definition
[2] NIST Glossary: Operational Technology
[3] NIST Glossary: Industrial Control System
[4] Trend Micro: BlackEnergy
[5] CISA ICS Advisory: ICS Focused Malware
[6] FIRST CSIRT Services Framework: Vulnerability Management
[7] DoD: Instruction 8531.01 Vulnerability Management
[8] FDA: Deciding When to Submit a 510(k) for a Software Change to an Existing Device
[9] TÜV SÜD: UN Regulation 156 - Automotive Software
[10] Department of Commerce: Securing the Information and Communications Technology and Services Supply Chain; Connected Software Applications
[11] NERC CIP: Cyber Security - System Security Management
[12] NIST Glossary: Authentication
[13] NIST Glossary: Authorization
[14] TODO Group: OSPO Definition and Guide
[15] Security Checklist - Minimum Viable Secure Product

DISCLAIMER

VERSION HISTORY

October 10, 2023: Initial version.