An Overview of ISA/IEC 62443-4-1 and Its Supply Chain Requirements

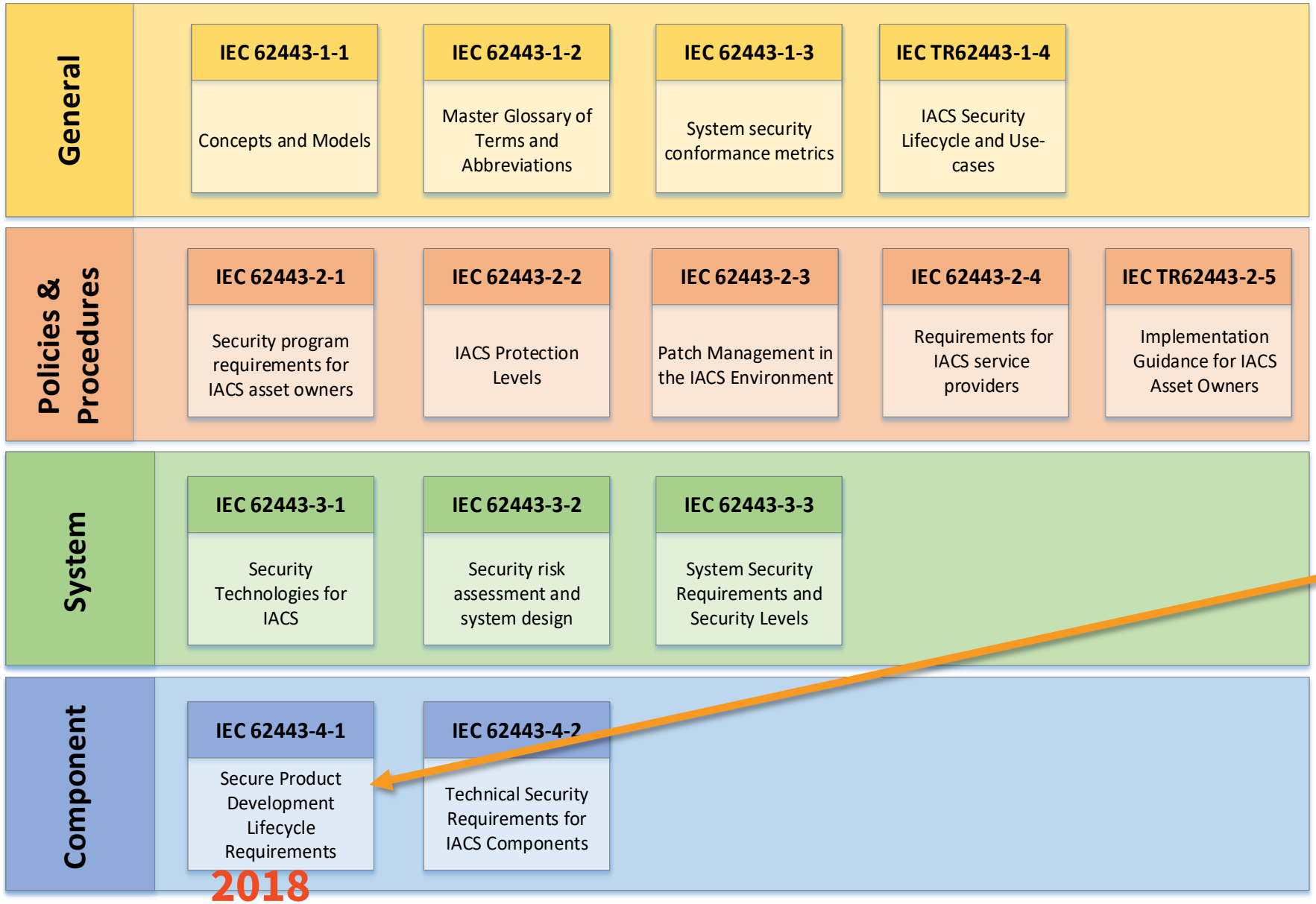
Greg Houser, CISA, CISSP, MS TC
Senior Cybersecurity Engineer

The Webinar will begin at 11AM EST

Overview

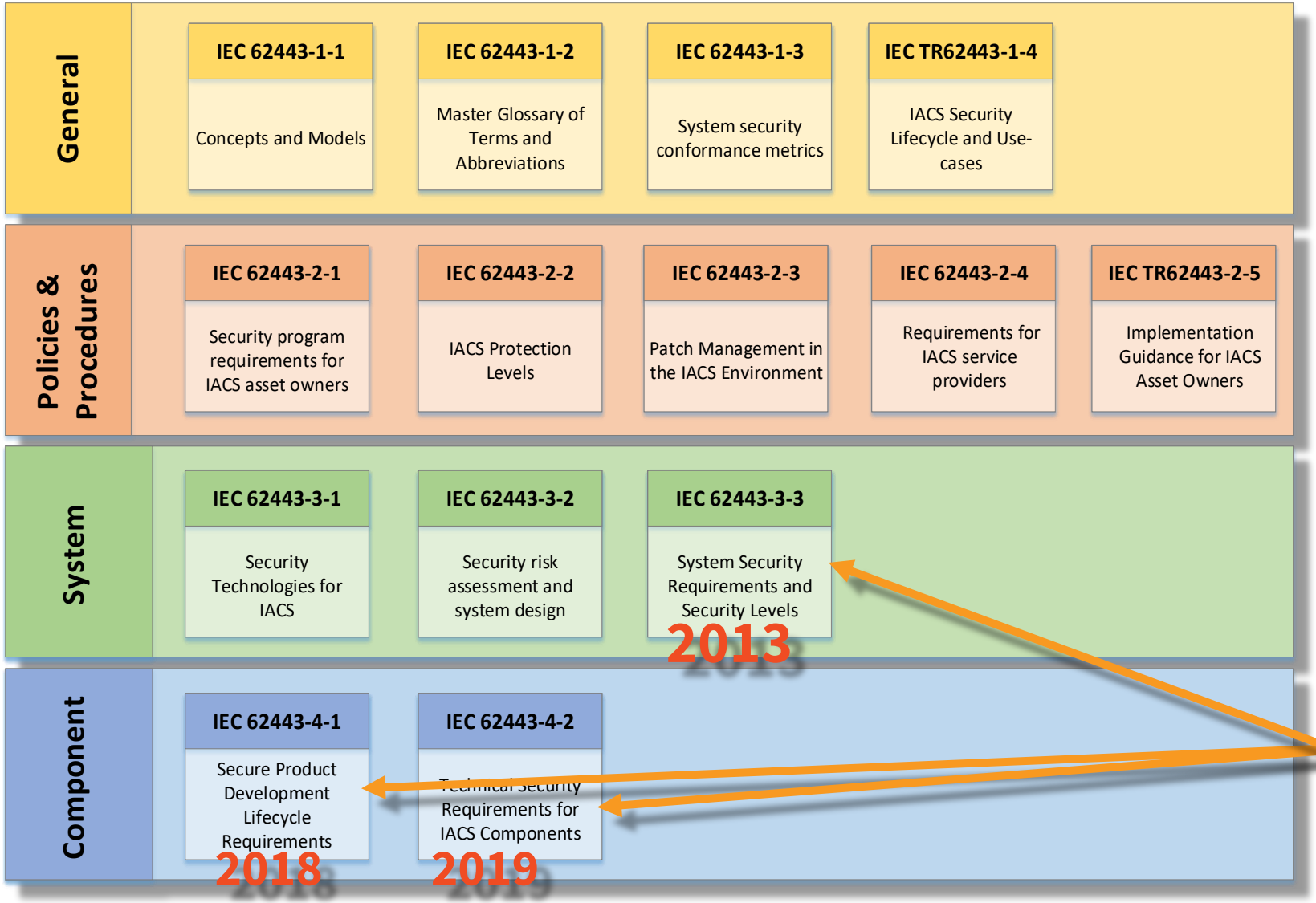
1. IEC 62443 4-1 Overview
2. Software Bill of Material (SBOM)
3. Importance of the SBOM
4. Standard formats
5. Conclusion

IEC 62443 Structure



Focus of this presentation

SDL in the IEC 62443 family of standards



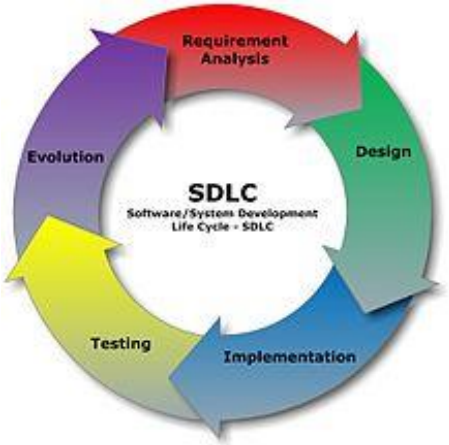
IEC 62443-4-1 Applies to:

- IACS Components (IEC 62443-4-2)
- IACS Systems (IEC 62443-3-3)
- **General software product, hardware product or system development**

IEC 62443-4-1 SDL practices

- Defines process (practices) to be used when developing products securely throughout the entire development lifecycle
  - *Practice 1 - Security Management*
  - *Practice 2 - Specification of Security Requirements*
  - *Practice 3 - Secure by Design*
  - *Practice 4 - Secure Implementation*
  - *Practice 5 - Security Verification and Validation Testing*
  - *Practice 6 - Management of security related issues*
  - *Practice 7 - Security Update Management*
  - *Practice 8 - Security Guidelines*
- ***Follows industry-best SDL practices***
- ***Not market specific – applicable to secure development of any product or system***

SDL (Secure Development Lifecycle) Definition

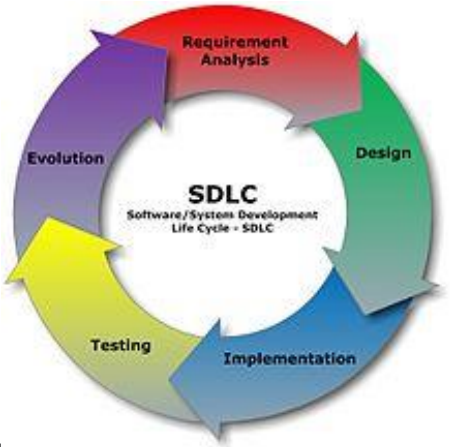


- SDL is interchangeable with:
  - SDLC (Secure Development Life Cycle)
  - Security Development Lifecycle
  - Secure Development Lifecycle
- SDL introduces security considerations throughout all phases of the development process, helping developers build highly secure products and systems, address security compliance requirements, and reducing development and sustaining costs.
- *A key focus of SDL is **building security in** up front.*
- SDL was initially focused on software but applies to hardware products and systems such as IACS (Industrial Automation And Control Systems), DCS (Distributed Control System) and electrical utility systems

SDL Maturity Levels

The IEC 62443-4-1 standard defines the SDL process maturity levels below.

- ML 1: Initial – *Organization has no SDL process or an adhoc SDL process.*
- ML 2: Managed – *SDL process is defined and organization is prepared to follow it.*
- ML 3: Defined - *Organization is consistently following their SDL process.*
- ML 4/5: Improving - *Organization is consistently following and improving their SDL process.*



Significant Cyber Incidents 2009-2017

Dallas Emergency Sirens Hack

Saudi Aramco Targeted Attack

Shamoon Virus

Finland Smart Building

German Steel Mill Attack

SF Muni Ransomware

Texas Power Company Hack

Ukraine Power Grid Attack

"NotPetya" Wiper

Unidentified Nuclear Power Plant

"WannaCry" Ransomware

2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018

Houston Water Distribution Systems Hack

2nd Ukrainian Power Grid Attack

Alpine Ski Resort Ransomware

Ransomware Attacks on Hospitals

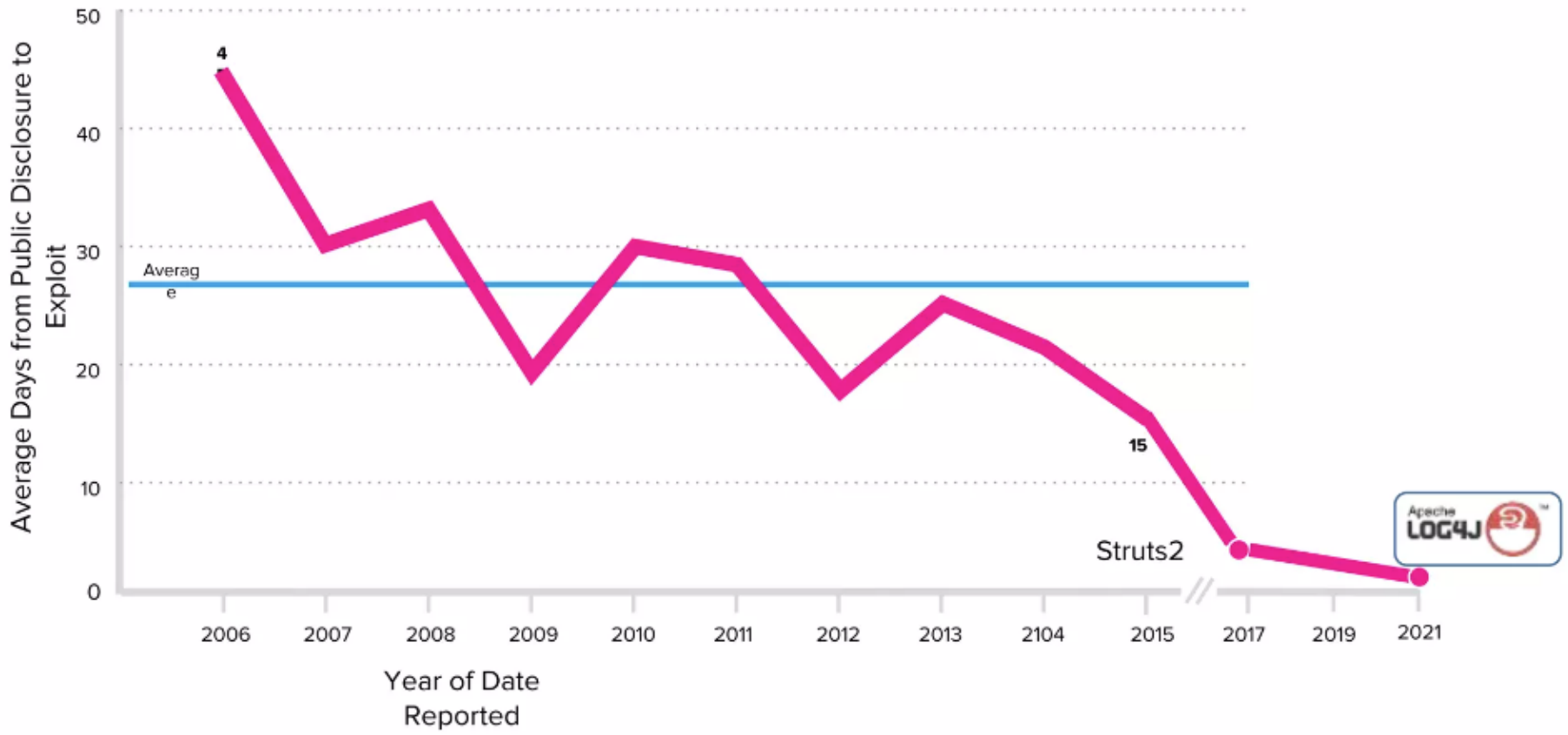
Triton hits Petro Rabigh Refinery

Unidentified Water Facility

Zero Day Progression

The Zero Day Window is Closing

Source: Adapted from IBM X-Force / Analysis by Gartner Research (September 2016)

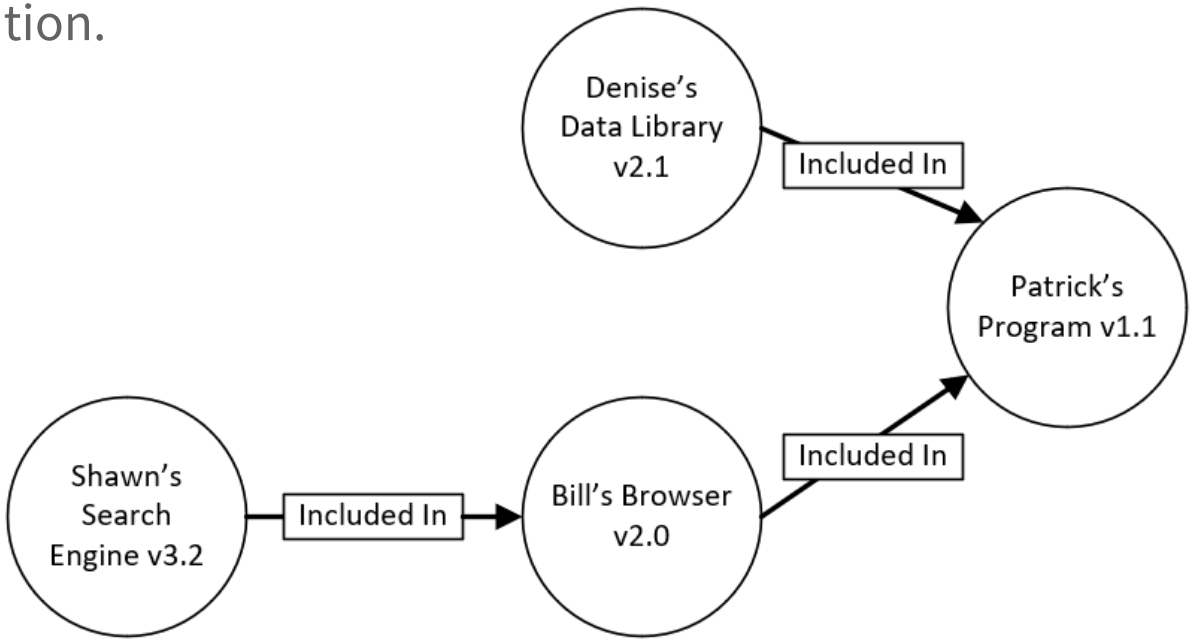


All of this leads to SBOMs

INGREDIENTS: WATER (75%), **SUGARS (12%)** (GLUCOSE (48%), FRUCTOSE (40%), SUCROSE (2%), MALTOSE (<1%)), STARCH (5%), **FIBRE E460 (3%), AMINO ACIDS** (GLUTAMIC ACID (19%), ASPARTIC ACID (16%), HISTIDINE (11%), LEUCINE (7%), LYSINE (5%), PHENYLALANINE (4%), ARGININE (4%), VALINE (4%), ALANINE (4%), SERINE (4%), GLYCINE (3%), THREONINE (3%), ISOLEUCINE (3%), PROLINE (3%), TRYPTOPHAN (1%), CYSTINE (1%), TYROSINE (1%), METHIONINE (1%)), **FATTY ACIDS (1%)** (PALMITIC ACID (30%), OMEGA-6 FATTY ACID: LINOLEIC ACID (14%), OMEGA-3 FATTY ACID: LINOLENIC ACID (8%), OLEIC ACID (7%), PALMITOLEIC ACID (3%), STEARIC ACID (2%), LAURIC ACID (1%), MYRISTIC ACID (1%), CAPRIC ACID (<1%)), ASH (<1%), PHYTOSTEROLS, E515, OXALIC ACID, E300, E306 (TOCOPHEROL), PHYLLOQUINONE, THIAMIN, **COLOURS** (YELLOW-ORANGE E101 (RIBOFLAVIN), YELLOW-BROWN E160a), **FLAVOURS** (3-METHYLBUT-1-YL ETHANOATE, 2-METHYLBUTYL ETHANOATE, 2-METHYLPROPAN-1-OL, 3-METHYLBUTYL-1-OL, 2-HYDROXY-3-METHYLETHYL BUTANOATE, 3-METHYLBUTANAL, ETHYL HEXANOATE, ETHYL BUTANOATE,, PENTYL ACETATE), 1510, NATURAL RIPENING AGENT (ETHENE GAS).

Software Bill of Materials (SBOM)

- Set of baseline information about a software application.
- SBOM information should include:
  - Name of the supplier.
  - Name of the component.
  - Name of the author.
  - Unique identifier.
  - Version and rev of the component.
  - Functional relationships of components.
  - Date and time when the SBOM was last created/updated
- Provides greater visibility into the software supply chain.
- Incentivizes the adoption of secure software development practices.
- Allows for earlier identification and mitigation of potentially vulnerable systems.
- Enhances software supply chain security.



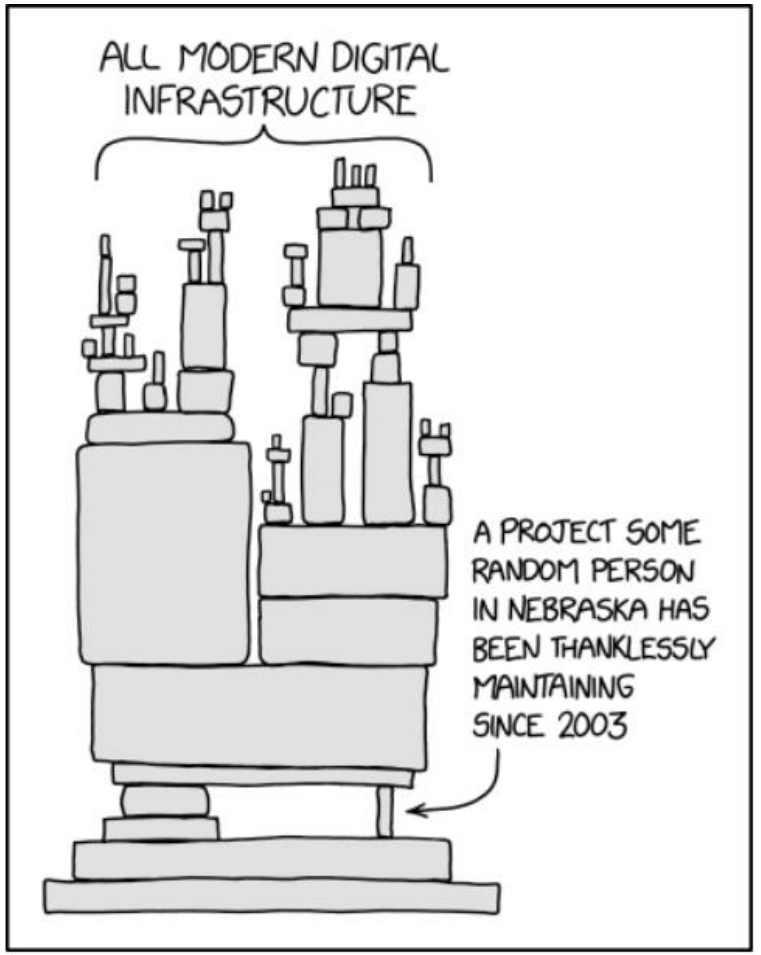
Source: *Managing Cybersecurity in the Process Industries: A Risk-Based Approach*, AIChE CCPS, 2021

Why do I need it?

- Software is going to contain libraries and modules
- The software (or its components) could be open source (FOSS) or proprietary (COTS/GOTS), or a combination of both.
- Could be widely available or access-restricted
- As an added bonus, that software is running on device that is probably connected to a network
- You need to trust your SOFTWARE to trust the DEVICE/SYSTEM it's running on!

So why haven't we been doing this?

- It can be hard to do
  - Time consuming
  - Many layers
- SW components often use multiple names, creating confusion
  - Apache, anyone?
- No one was asking for it
  - At least, not until relatively recently



Credit: XKCD

SBOM Requirements (?)

- IEC 62443 does not explicitly require an SBOM or recommend any particular SBOM format
  - SPDX? SPDX-lite? SWID? CycloneDX?, etc.?
- IEC 62443-4-1 Recommends that there be an inventory of components from third party suppliers
- It also requires a process to identify and manage the security risks of all externally provided components (SM-9, SM-10)
  - Having an SBOM makes it much easier to meet this requirement

Extends past 4-1

- 4-1 works to underpin other standards
- The requirement to report a current list of installed components is also in:
  - 3-3 (SR 7.8)
  - 4-2 CR (7.8)
- *[…]shall provide the capability to report the current list of installed components and their associated properties.*

IEC 62443 2-4 (SP.06.02) goes a little further

- The service provider shall have the capability to create and maintain an inventory register, including version numbers and serial numbers, of all devices **and their software components** in the Automation Solution for which the service provider is responsible.

Another good reason

- Commonly required as part of any Bill of Materials (BOM)
  - Part of negotiated terms
  - Compliance with regulatory requirements
  - Legal Obligations
  - Identification of software and software component dependencies
    - Inventory/asset management
    - Change Management
    - Vulnerability Management
    - Identification of supply chain risks

Even More Good Reasons

- Export
  - Export Control List
- Security

Required for

- Export
  - Export Control List
- **Security**
  - FDA, NERC, DoD, ENISA,etc.
  - IEC 62443 3-3 (SL 2), 4-2 (SL 2)
  - IEC 62443 2.4 (BR)

We're at SL 1, so it doesn't affect me

- ISCI recently put out a document making a case that all IEC 62443 4-2 certifications for IACS components be raised to SL 2 at a minimum (*ISCI Case for 62443 Security Level 2*).
- makes an inventory list of all components (including software) a de facto mandatory requirement

Should I be paying attention?

What does it do for me?

- Monitor for vulnerabilities
  - Identify known vulnerabilities
- Prepare for sunsetting and End of Life (EoL)
- Better manage code base
- Minimize bloat in design
- Ease in execution for whitelist/blacklist practices
- Better understand and comply with license and regulatory requirements
- Better understanding of the provenance of what is under the hood

How do we do this?

- IEC 62443 doesn’t really give us a method
  - But we have a few preexisting standards we can use
- Software Package Data Exchange (SPDX)
- Software Identification (SWID)
- CycloneDX

Software Package Data Exchange (SPDX)

- Designed for licensing, but can be used for SBOM very easily
- Extensible
- Open Source
- Machine readable
- Developed by the Linux Foundation

```
SPDXVersion: SPDX-2.2
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: spdx-sbom-generator
DocumentNamespace: http://spdx.org/spdxpackages/spdx-sbom-generator--57918521-3212-4369-a8ed-3d681ec1d7a1
Creator: Tool: spdx-sbom-generator-XXXXX
Created: 2021-05-23 11:25:29.1672276 -0400 -04 m=+0.538283001

##### Package representing the Go distribution

PackageName: go
SPDXID: SPDXRef-Package-go
PackageVersion: v0.46.3
PackageSupplier: NOASSERTION
PackageDownloadLocation: pkg:golang/cloud.google.com/go@v0.46.3
FilesAnalyzed: false
PackageChecksum: TEST: SHA-1 224ffa55932c22cef869e85aa33e2ada43f0fb8d
PackageHomePage: pkg:golang/cloud.google.com/go@v0.46.3
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
PackageLicenseComments: NOASSERTION
PackageComment: NOASSERTION

Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-Package-go

##### Package representing the Bigquery Distribution

PackageName: bigquery
SPDXID: SPDXRef-Package-bigquery
PackageVersion: v1.0.1
PackageSupplier: NOASSERTION
PackageDownloadLocation: pkg:golang/cloud.google.com/go/bigquery@v1.0.1
FilesAnalyzed: false
PackageChecksum: TEST: SHA-1 8168e852b675afc9a63b502feeefac90944a5a2a
PackageHomePage: pkg:golang/cloud.google.com/go/bigquery@v1.0.1
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
PackageLicenseComments: NOASSERTION
PackageComment: NOASSERTION

Relationship: SPDXRef-Package-go CONTAINS SPDXRef-Package-bigquery
```

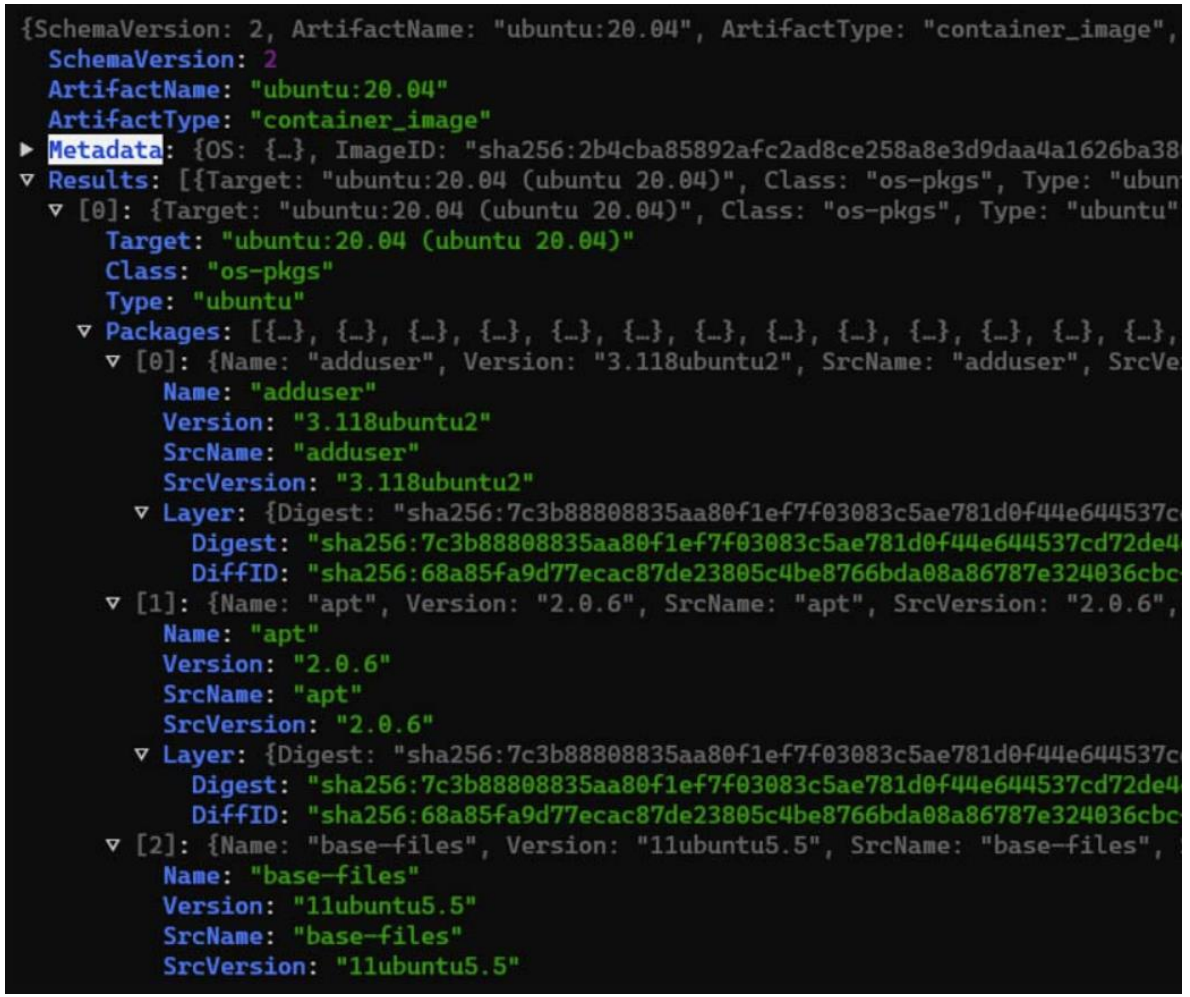
Software Identification (SWID)

- Designed for identifying software on disc
- More of a software identifier than an SBOM format
- Cantrack software inventory by storing specific information about the software release.
- ISO Standard (ISO/IEC 19770-2:2015)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<SoftwareIdentity name="NetLicensing" supplemental="true" tagId="e_1" version="2.1.0"
versionScheme="unknown" xmlns="http://standards.iso.org/iso/19770/-2/2014-
DIS/schema.xsd" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">

  <Entity name="Labs64" role="softwareCreator softwareLicensor tagCreator"/>

  <Link href="swid:other-swid-tag" rel="supplemental"/>

  <Meta description="This is what it's about" entitlementDataRequired="true" revision="3"/>

  <Evidence date="2020-04-24Z" xml:lang="123-a">

    <File name="File.xml" size="10" version="3"/>

  </Evidence>

  <Payload>

    <Directory key="true" location="/folder" root="/data"/>

  </Payload>

</SoftwareIdentity>
```

CycloneDX

- Specifically designed to allow users to create SBOMs
- Lightweight, efficient
- Primary use case is vulnerability identification
- Open source
- Format focuses on ease of adoption and automation of SBOM generation
- Developed by OWASP



In Closing

- IEC 62443 4-1 doesn't directly require it, but if we're to have an inventory of components from third party suppliers an SBOM makes it easier
- An SBOM helps us to identify and manage the security risks of all externally provided software components
- An SBOM makes it easier for when we take 4-1 and also wish to go through other IEC 62443 certification processes (2-4, 3-3, 4-)
- Improves our overall security stance, and makes tasks such as change management easier

Questions?

Email: ghouser@exida.com
Website: www.exida.com
Reference material: www.exida.com/Books
Whitepapers: http://www.exida.com/Resources/Whitepapers

An Overview of ISA/IEC 62443-4-1 and Its Supply Chain Requirements

Main Offices

- USA +1 215 453 1720
- Germany +49 89 4900 0547
- Canada +1 403 475 1943
- Mexico +52 55 5611 9858
- Singapore +65 6222 5160
- South Africa +27 31 267 1564
- Japan +81 (0)50-5539-9507
- United Kingdom +44 19 266 76125
- India +91 9930250104

Regional Offices

- Netherlands +31 318 414 505
- New Zealand +64 3 472 7707

Backup Slides

How SDL Maturity levels are linked to Security Capability Levels

An analogy for linking SDL Maturity Levels to Product Security Capability levels …

Target security capability level = 3

Component vendor A SDL Maturity Level = **0**

- *Vendor lacked a mature SDL and didn't implement Security Capabilities correctly*

Component vendor B SDL Maturity Level = **4**

- *Vendor follows a robust and improving SDL and implements all Security Capabilities correctly*

Putting it all together: Maturity Levels, Security Capability Levels and Security For Customers

The image below shows an analogy for linking SDL Maturity Levels to Product Security Capability levels

Appropriate Target Security Capability Level
- *Appropriate SL for the risk and cost*

+

Robust component vendor SDL process
- *Robust, repeatable and improving vendor SDL process*

+

Achieved Product Security Capability Level
- Security capabilities match target

+

End customer leveraging product security capabilities
- Customer understands and leverages security capabilities

=

Reduced risk of cybersecurity incidents for customers