Energysec: Generating SBoMs Utilizing Structured Threat Information Expression JSON bundles

October 2021

Manuel Vazquez Jr.
DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.
Energysec: Generating SBoMs Utilizing Structured Threat Information Expression JSON bundles

Manuel Vazquez Jr.

October 2021

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517
Generating SBoMs Utilizing Structured Threat Information Expression JSON bundles
Idaho National Laboratory – Manuel Vazquez
Agenda

- Introduction
- STIX Overview
- Project Overview
- Software Bill of Materials
- Questions

This Photo by Unknown Author is licensed under CC BY
Who am I?

Introduction

- Software Developer/Researcher with Idaho National Laboratory for about 1 year.
- BS Information Systems Technology Cybersecurity / Computer Science: California State University, San Bernardino
- I enjoy ‘breaking’ things.
STIX

• What is STIX?
  − A format used to share and serialize using XML (STIX v1)/ JSON (STIX v2)
  − STIX uses its own syntax
  − Has committee that manages specification
  − Designed to improve capabilities in:
    • Automated threat exchange
    • Automated detection and response
    • Collaborative threat analysis

https://oasis-open.github.io/cti-documentation
Firmware Command and Control (FC2)

- Funding from Department of Energy Offices
- Energy Efficiency and Renewable Energy (EERE) – Solar Energy Technology Office (SETO) Guohui Yuan
- Cybersecurity, Energy Security and Emergency Response (CESER) Akhlesh Kaushiva (AK)
- EERE - Building Technology Office (BTO) Erika Gupta

Industry Partners:
Mapping components into STIX

- Some of the components that we were able to map:
  - Hardware
  - The Flow of Data

- We leverage scripts to translate software component and get results
Software Bill of Materials

What is this phenomenon?

• List of components in a piece of software.

• Software Bill of Materials aims to fix this by providing information to the end user.

• There are Different formats:
  − CycloneDX
  − Software Package Data Exchange (SPDX)
  − Software Identification (SWID) Tags

• This can be used to identify vulnerabilities or license issues.

• In the Cyber world, information is key. Here at Idaho National Laboratory, we are seeking ways to solve this issue to help identify potential issues ahead of time.
INL SBOM Generation

Scan

Generate STIX Bundle

Generate Validated SBOM

Generate CSV

STIX

NMAP

IDAH0 NATIONAL LABORATORY
IX: Mapping our Applications
Generating SBOM from STIX Bundles

• Utilizing Python we’re able to extract application from bundles to generate SBOM

• There are different formats, this example can search for software within your environment and generate a CycloneDX format Software Bill of Material.

```json
"bomFormat": "CycloneDX",
"specVersion": "1.3",
"serialNumber": "urn:uuid:3e671687-395b-41f5-a30f-a58921a69b79",
"version": 1,
"components": [
    {
        "type": "operating-system",
        "name": "Linux 2.6.9 - 2.6.33",
        "version": "Linux 2.6.9 - 2.6.33",
        "cpe": "cpe:/o:linux:linux_kernel:2.6"
    },
    {
        "type": "application",
        "name": "vsftpd",
        "version": "2.3.4",
        "cpe": "cpe:/a:vsftpd:vsftpd:2.3.4"
    },
    {
        "type": "application",
        "name": "OpenSSH",
        "version": "4.7p1 Debian Bubuntui",
        "cpe": "cpe:/o:linux:linux_kernel"
    },
    {
        "type": "application",
        "name": "Linux telnetd",
        "version": "",
        "cpe": "cpe:/o:linux:linux_kernel"
    },
    {
        "type": "application",
        "name": "Postfix smttd",
        "version": "",
        "cpe": "cpe:/a:postfix:postfix"
    },
    {
        "type": "application",
        "name": "ISC BIND",
        "version": "9.4.2",
        "cpe": "cpe:/a:isc:bind:9.4.2"
    }
],
```
<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>version</th>
<th>cpe</th>
</tr>
</thead>
<tbody>
<tr>
<td>operating-system</td>
<td>Linux 2.6.9 - 2.6.33</td>
<td>Linux 2.6.9 - 2.6.33</td>
<td>cpe:o:linux:linux_kernel:2.6</td>
</tr>
<tr>
<td>application</td>
<td>vsftpd</td>
<td>2.3.4</td>
<td>cpe:a:vsftpd:vsftpd:2.3.4</td>
</tr>
<tr>
<td>application</td>
<td>OpenSSH</td>
<td>4.7p1 Debian 8ubuntu1</td>
<td></td>
</tr>
<tr>
<td>application</td>
<td>Linux telnetd</td>
<td></td>
<td>cpe:o:linux:linux_kernel</td>
</tr>
<tr>
<td>application</td>
<td>Postfix smtpd</td>
<td></td>
<td>cpe:a:postfix:postfix</td>
</tr>
<tr>
<td>application</td>
<td>ISC BIND</td>
<td>9.4.2</td>
<td>cpe:a:isc:bind:9.4.2</td>
</tr>
<tr>
<td>application</td>
<td>Apache httpd</td>
<td>2.2.8</td>
<td>cpe:a:httpd:http_server:2.2.8</td>
</tr>
<tr>
<td>application</td>
<td>netkit-rsh rexecd</td>
<td></td>
<td>cpe:o:linux:linux_kernel</td>
</tr>
<tr>
<td>application</td>
<td>OpenBSD or Solaris rlogin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>application</td>
<td>GNU Classpath grmiregistry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>application</td>
<td>Metasploitable root shell</td>
<td></td>
<td></td>
</tr>
<tr>
<td>application</td>
<td>ProFTPD</td>
<td>1.3.1</td>
<td>cpe:a:proftpd:proftpd:1.3.1</td>
</tr>
<tr>
<td>application</td>
<td>MySQL</td>
<td>5.0.51a-3ubuntu5</td>
<td>cpe:a:mysql:mysql:5.0.51a-3ubuntu5</td>
</tr>
<tr>
<td>application</td>
<td>distccd</td>
<td>v1</td>
<td></td>
</tr>
<tr>
<td>application</td>
<td>PostgreSQL DB</td>
<td>8.3.0 - 8.3.7</td>
<td>cpe:a:postgresql:postgresql:8.3</td>
</tr>
<tr>
<td>application</td>
<td>VNC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>application</td>
<td>UnrealIRCD</td>
<td></td>
<td>cpe:a:unrealircd:unrealircd</td>
</tr>
<tr>
<td>application</td>
<td>Ruby DRb RMI</td>
<td></td>
<td>cpe:a:ruby-lang:ruby:1.8</td>
</tr>
</tbody>
</table>
# Potential Vulnerabilities

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
<th>Version</th>
<th>Update</th>
<th>Edition</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpe:/a:samba:samba:</td>
<td>View CVEs</td>
<td>cpe:/a:samba:samba:</td>
<td>View CVEs</td>
<td>cpe:/a:samba:samba:</td>
<td>View CVEs</td>
</tr>
<tr>
<td>samba</td>
<td>samba</td>
<td>samba</td>
<td>samba</td>
<td>samba</td>
<td>samba</td>
</tr>
<tr>
<td>View CVEs</td>
<td>View CVEs</td>
<td>View CVEs</td>
<td>View CVEs</td>
<td>View CVEs</td>
<td>View CVEs</td>
</tr>
</tbody>
</table>

**CVE-2021-39246**

For Browser through 10.5.6 and 11.x through 11.64 allows a correlation attack that can compromise the privacy of visits to v2 onion addresses. Exact timestamps of these onion-service visits are logged locally, and an attacker might be able to compare them to timestamp data collected by the destination server (or collected by a rogue site within the Tor network).

Published: September 24, 2021; 3:15:07 PM -0400

**CVE-2021-38877**

IBM Jazz for Service Management 1.1.3.10 is vulnerable to stored cross-site scripting. This vulnerability allows users to embed arbitrary JavaScript code in the Web UI thus altering the intended functionality potentially leading to credentials disclosure within a trusted session. IBM X-Force ID: 208405.

Published: September 23, 2021; 2:15:11 PM -0400

**CVE-2021-28905**

IBM Jazz for Service Management 1.1.3.10 and IBM Tivoli Netcool/OMNibus_GUI is vulnerable to cross-site scripting. This vulnerability allows users to embed arbitrary JavaScript code in the Web UI thus altering the intended functionality potentially leading to credentials disclosure within a trusted session. IBM X-Force ID: 207966.

Published: September 23, 2021; 2:15:11 PM -0400

**CVE-2021-28904**

IBM Jazz for Service Management 1.1.3.10 and IBM Tivoli Netcool/OMNibus_GUI displays user credentials in plain clear text which can be read by a local user. IBM X-Force ID: 207960.

Published: September 23, 2021; 2:15:11 PM -0400
Demo (Video)

• Video will show:
  − Generation of IX bundle
  − Generation of validated Software Bill of Material from IX Bundle
  − Use of Created web app to load the Software Bill of Materials
  − How this can be used to leverage potential vulnerabilities
Video Demo

```python
import nmap
from stix2 import Malware, Software, Process, Bundle, Relationship, Infrastructure, IPV4Address, MACAddress
import json
import pprint

def scan(target):
    global report_list
    nm = nmap.PortScanner()
    try:
        mmm.scan(target, arguments= '-sS -sV -sC -O')
        ports = nm[target]['tcp'].keys()
        report_list = []
        for port in ports:
            report = {'hardware': {},
                      'software': {}}
            address = nm[target]['addresses'][0]['ipv4']
            hardware = nm[target]['addresses'][0]['mac']
            os = nm[target]['osmatch'][0]['name']
            oscpu = nm[target]['osmatch'][0]['osclass'][0]['os'][0]
            state = nm[target]['tcp'][port]['state']
            product = nm[target]['tcp'][port]['product']
            service = nm[target]['tcp'][port]['name']
            version = nm[target]['tcp'][port]['version']
            cpe = nm[target]['tcp'][port]['cpe']
            report['hardware']['ip'] = address
            report['hardware']['os'] = os
            report['hardware']['oscpu'] = oscpu
            report['software']['port'] = port
            report['software']['state'] = state
            report['software']['product'] = product
            report['software']['version'] = version
            report['software']['cpe'] = cpe
            if state == 'open':
                report_list.append(report)
    except Exception as e:
        print(e)
    return report_list

def bundle(report):
    global Target
    Target = Infrastructure(name='Device')
```
Next Steps: Binary Visualization from SBOM

- Automating vulnerabilities
Project Supporting our Efforts
Questions
Sources

- https://cyclonedx.org/
- https://oasis-open.github.io/cti-documentation
- https://spdx.dev/