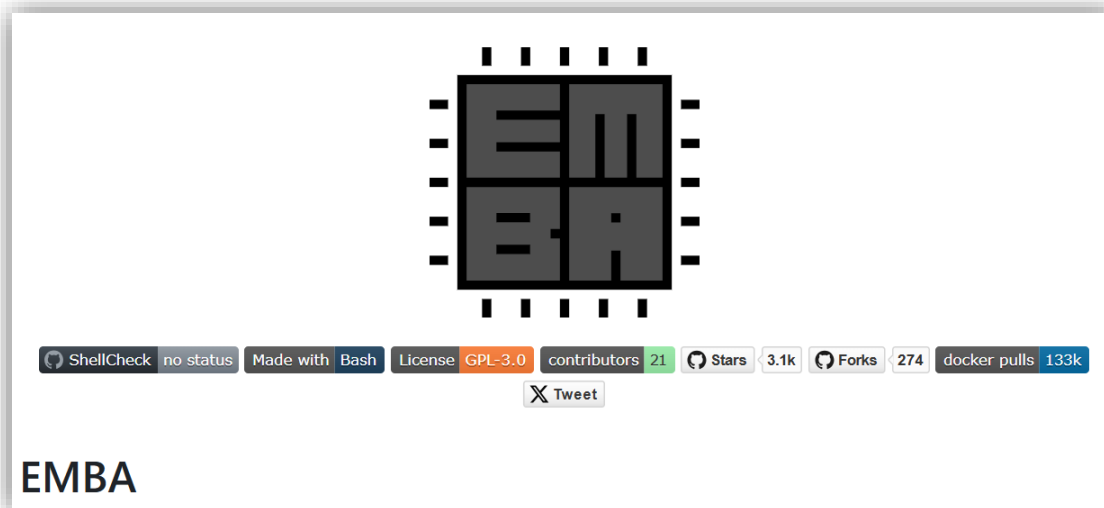


EMBAを使ってみました

平能 耀介 ルネサスエレクトロニクス株式会社

EMBA(Embedded Linux Analyzer)

- オープンソース(GPLv3)の組み込みLinux向けの脆弱性解析ツール
- エンジニアのMichael Messnerによって開発された(Siemens Energy AG)
- 過去にBlack Hatなどのカンファレンスなどで紹介されていた
- 業務でSBOMに関わっているので、最近SBOM生成機能が追加されたことを知って調査することにした





Publications, Talks and Live demos

- TROOPERS 25 - SBOMs the right way (June 2025) - [Schedule/Slides](#) [EN]
- Black Hat MEA Arsenal - Riyadh 2024 (November 2024) - [Schedule/Slides](#) [EN]
- FLOSS Weekly Episode 802: EMBA – Layers Upon Layers Of Bash (September 2024)
- Leveraging Automated Firmware Analysis with the Open-Source Firmware Analyzer [medium.com](#) [EN]
- Black Hat Arsenal - Asia 2024 (April 2024 - Virtual) [Schedule](#) [EN]
- BSides Las Vegas - USA 2023 (August 2023) [Schedule/Recording](#) [EN]
- Black Hat Arsenal - USA 2023 (August 2023) [Schedule](#) [EN]
- DEF CON 31 - ICS Village - USA 2023 (August 2023) [Schedule](#) [EN]
- Firmware Scraper - AMOS project presentation (2022/2023) - [Project page](#) [EN]
- Black Hat Arsenal - Europe 2022 (December 2022) [Schedule/Slides/Recording](#) [EN]

EMBA(Embedded Linux Analyzer)

- もともとはペネトレーションテストを行う人向けのツールだが、開発者は今後はソフトウェア開発者にも使用してほしいと考えている
- 2024年10月 v1.5からSBOMの生成機能が追加された
- 2025年3月 v1.5.2 CRAに備えて今後はSBOMツールとして開発していくことになった（これまで通りfirmware analyzer機能の開発も継続）
- SBOMの精度を高めたり、関連機能の開発することに意欲的な様子

EMBA v1.5.2 - SBOM - The next generation Latest

 m-1-k-3 released this Mar 11 · 739 commits to master since this release · v1.5.2-SBOM-... · fbe1811 

We need to talk about serious SBOM tooling! The [CRA](#) will hit us all ... quite hard and very soon. Check the dates (from [Wikipedia](#)):

Journal reference [OJ L, 2024/2847, 20.11.2024](#)

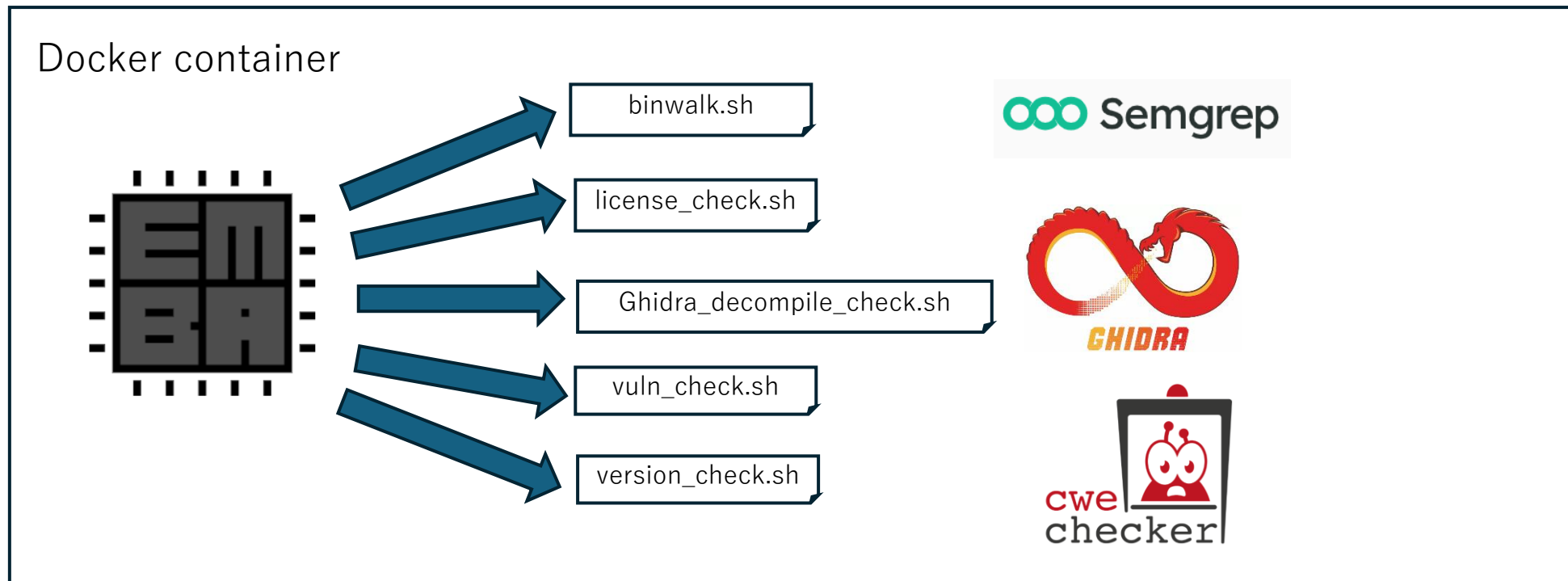
History	
Date made	23 October 2024
Entry into force	12 November 2024
Applies from	11 December 2027

Current legislation

And check the SBOM requirements [here](#):

EMBAの動作概要

- ペネトレーションテスターが手動で行っているような解析作業を集約して自動実行するもの
- docker環境に解析に必要なOSSをインストールし、それらを組み合わせて脆弱性解析を自動実行する



EMBAの動作概要

- binwalkなどでバイナリに含まれているデータを抽出（ファイルシステム、カーネルイメージ、ELFバイナリなど）
- 抽出したデータに対してbashスクリプトによる操作とOSSを利用して解析を行う
- 以下のような作業を自動実行してくれる
 - ✓ カーネルの脆弱性検出
 - ✓ 文字列検索でライブラリのバージョン/ライセンス情報特定
 - ✓ QEMU上でエミュレーションしてソフトウェアのバージョン/ライセンス情報特定
 - ✓ cwe_checkerを利用したC/C++の危険なコードパターンのチェック
 - ✓ GhidraとSemgrepを利用したソースコードレベルの脆弱性解析

インストール

- Githubの手順に従ってリポジトリをクローンし、installer.sh を実行
- 動作環境の指定は厳しく、原則Kali LinuxとUbuntu:jammy (22.04 LTS)のみで動作する
- 動作環境を満たしていても、インストール時にエラーが発生した
- Github上にインストールができないというissueも多くあり、困っている人は多そう

実行

- 解析したいファイルと設定ファイルを引数に指定してCLI上で./embaを実行
- 解析対象としてRaspberry Pi OS 64bitを使用 [Raspberry Pi OS downloads – Raspberry Pi](#)
- 解析にかかった時間は10時間28分
- cwe_checkerを利用したC/C++の危険なコードパターンのチェックに最も時間がかかり、5時間程度かかった
- 動作環境
 - ✓ OS : Ubuntu 22.04.5 LTS
 - ✓ CPU : INTEL(R) XEON(R) GOLD 5512U
 - ✓ RAM : 256GB

結果の確認

- 解析結果はhtml形式のレポートとして保存され、ブラウザから解析結果を確認できる

The screenshot displays the EMBA web interface, which provides a comprehensive security analysis report. The sidebar on the left contains navigation links for various analysis stages, including 'Binary firmware file analyzer', 'Binwalk binary firmware extractor', 'Analysis preparation', 'Binary firmware basic analyzer', 'Firmware and testing details', 'Check bootloader and system startup', 'EMBA central package SBOM environment', 'Static binary firmware versions detection', 'Check binary protection mechanisms', 'Check binaries for weak functions (radare mode)', 'Check decompiled binary source code for vulnerabilities', 'Check binaries for vulnerabilities with cwe-checker', 'Check scripts with shellcheck and semgrep', and 'Binary and Configuration'.

The main content area shows the following findings:

- Found 18 potential security issues in 224 lua scripts.**
- Verified 101 kernel vulnerabilities (kernel symbols).**
- Found the following configuration issues:**
 - Found 1 areas with weak permissions.
 - Found 1 authentication issues.
 - Found 10 password related details via STACS.
 - Found 188 kernel modules with 0 licensing issues.
 - Found 0 interesting files and 3 files that could be useful for post-exploitation.
- Identified the following binary details:**
 - Found 391 (100%) binaries without enabled stack canaries in 391 binaries.
 - Found 386 (99%) binaries without enabled RELRO in 391 binaries.
 - Found 391 (100%) binaries without enabled NX in 391 binaries.
 - Found 128 (33%) binaries without enabled PIE in 391 binaries.
 - Found 203 (52%) stripped binaries without symbols in 391 binaries.
- cwe-checker found a total of 2989 security issues in 15 tested binaries:**
 - CWE119 - Buffer Overflow - 5 times.
 - CWE125 - Out-of-bounds Read - 3 times.
 - CWE134 - Externally Controlled Format String - 173 times.
 - CWE190 - Integer Overflow or Wraparound - 38 times.
 - CWE252 - Unchecked Return Value - 248 times.
 - CWE415 - Double Free - 33 times.
 - CWE416 - Use After Free - 166 times.
 - CWE467 - Use of sizeof on a Pointer Type - 22 times.
 - CWE476 - NULL Pointer Dereference - 1172 times.
 - CWE676 - Use of Potentially Dangerous Function - 944 times.
 - CWE782 - Exposed IOCTL with Insufficient Access Control - 1 times.
 - CWE787 - Out-of-bounds Write - 170 times.
 - CWE789 - Large memory allocation - 14 times.
- Found 3741 possible vulnerabilities (via semgrep in Ghidra decompiled code) in 15 tested binaries.**
- Found 164 usages of strcpy in 391 binaries.**

The interface also displays two tables of results:

STRCPY - top 10 results:

COUNT	BINARY NAME	common	linux file:	y/n	CWE CNT	SEMGREP	RELRO	IBIN_CANA	NX state	SYMBOLS	NETWORKING
25	libc-0.9.3	common	linux file:	no	Vulns: 249	/ 1168	RELRO	No Canary	NX disabled	No Symbols	No Networking
15	libcurses.so.5	common	linux file:	no	Vulns: 191	/ 439	No RELRO	No Canary	NX disabled	No Symbols	No Networking
12	libp6tc.so.0.1	common	linux file:	yes	Vulns: NA	/ NA	No RELRO	No Canary	NX disabled	No Symbols	No Networking
12	libp4tc.so.0.1	common	linux file:	yes	Vulns: NA	/ NA	No RELRO	No Canary	NX disabled	No Symbols	No Networking
12	iinfo.so	common	linux file:	no	Vulns: 25	/ 37	No RELRO	No Canary	NX disabled	No Symbols	No Networking
8	libxtables.so.1	common	linux file:	no	Vulns: 139	/ 150	No RELRO	No Canary	NX disabled	No Symbols	No Networking
8	libintl.so.8.1	common	linux file:	no	Vulns: 120	/ 182	No RELRO	No Canary	NX disabled	No Symbols	No Networking
6	libuci.so	common	linux file:	no	Vulns: 141	/ 123	No RELRO	No Canary	NX disabled	No Symbols	No Networking
6	libntfs-3g.so.8	common	linux file:	no	Vulns: 897	/ 620	No RELRO	No Canary	NX disabled	No Symbols	No Networking
5	libubox.so	common	linux file:	no	Vulns: 52	/ 97	No RELRO	No Canary	NX disabled	No Symbols	Networking

SYSTEM - top 10 results:

COUNT	BINARY NAME	common	linux file:	y/n	CWE CNT	SEMGREP	RELRO	IBIN_CANA	NX state	SYMBOLS	NETWORKING
12	ledctl	common	linux file:	no	Vulns: 31	/ 28	No RELRO	No Canary	NX disabled	No Symbols	Networking
2	libssh.so.4.4.1	common	linux file:	no	Vulns: 434	/ 387	No RELRO	No Canary	NX disabled	No Symbols	Networking
1	libpthread-0.9	common	linux file:	no	Vulns: NA	/ NA	RELRO	No Canary	NX disabled	No Symbols	No Networking
1	liblua.so.5.1.4	common	linux file:	no	Vulns: 292	/ 101	No RELRO	No Canary	NX disabled	No Symbols	No Networking
0	xtables-multi	common	linux file:	yes	Vulns: NA	/ NA	No RELRO	No Canary	NX disabled	No Symbols	Networking un
0	xt_time	common	linux file:	yes	Vulns: NA	/ NA	No RELRO	No Canary	NX disabled	No Symbols	Networking un

カーネルの脆弱性検出

- 解析対象バイナリから抽出したカーネルのバージョン、アーキテクチャ、kconfigを利用してカーネルをdry compile
- dry compileにより、コンパイル時に使用されたソースファイルを抽出
- カーネルのシンボル情報を抽出
- 抽出した情報とCVEの情報を照らし合わせて精度の高いCVE検出を実現
- 参考： Towards Reliable and Scalable Linux Kernel CVE Attribution in Automated Static Firmware Analyses
<https://arxiv.org/pdf/2209.05217>
- 今回のRaspberry Pi OSの結果ではCVE-2019-3819が1つ検出された(HID系のバグ)

```
[+] Identified 29 unverified CVE vulnerabilities for kernel version 6.12.25
[*] Detected architecture arm64
[*] Extracted 179602 unique symbols from kernel and modules
[*] Extracted 30326 used source files during compilation
[*] Found 27 advisories with missing vulnerable path details
[*] Found 1 path details in CVE advisories but no real kernel path found in vanilla kernel source
[*] Found 1 path details in CVE advisories with real kernel path
[*] Found 0 path details in CVE advisories with real kernel path but wrong architecture
[*] 1 symbol usage verified
[*] 1 vulnerable paths verified via symbols
[*] 1 vulnerable paths verified via compiled paths

[+] Verified CVEs: 1 (exported symbols)
[+] Verified CVEs: 1 (compiled paths)
[+] Verified CVEs: 1 (one mechanism succeeded)
[+] Verified CVEs: 1 (both mechanisms overlap)
```

バージョン/ライセンス特定

- QEMUのユーザモードエミュレーションで-V, -helpオプション指定の実行でバージョンやライセンスを特定
- 用意しておいたワードリストを利用してバイナリ中の文字列検索(strings, grep)も行う

```
-h: applet not found
[*] Emulating binary ./usr/bin/busybox with parameter -help
-h: applet not found
[*] Emulating binary ./usr/bin/busybox with parameter --help
BusyBox v1.35.0 (Debian 1:1.35.0-4+b3) multi-call binary.
BusyBox is copyrighted by many authors between 1998-2015.
Licensed under GPLv2. See source distribution for detailed
copyright notices.

Usage: busybox [function [arguments]...]
or: busybox --list[-full]
or: busybox --show SCRIPT
or: busybox --install [-s] [DIR]
or: function [arguments]...
```

QEMU上で--helpコマンドを実行している

```
[+] Identified software components - via usermode emulation.

This module extracts version and license details from the results of the user-mode emulation module (s115).

[+] Version information found catman 2.11.2 in binary /usr/bin/catman (license: GPL-3.0-or-later) (emulation).
[+] Version information found Version: 1.0 in binary /usr/lib/apt/methods/rsh (license: unknown) (emulation/strict).
[+] Version information found lspd version 3.9.0 in binary /usr/bin/lspd (license: unknown) (emulation).
[+] Version information found procs-ng 4.0.2 in binary /usr/bin/free (license: unknown) (emulation).
[+] Version information found BusyBox v1.35.0 (Debian 1:1.35.0-4+b3) multi-call binary in binary /usr/bin/busybox (license: GPL-3.0-or-later) (emulation).
[+] Version information found mke2fs 1.47.0 in binary /usr/sbin/mke2fs (license: unknown) (emulation).
[+] Version information found rsync version 3.2.7 protocol version 32 in binary /usr/bin/rsync (license: unknown) (emulation).
[+] Version information found ethtool version 6.1 in binary /usr/sbin/ethtool (license: unknown) (emulation).
[+] Version information found whatis 2.11.2 in binary /usr/bin/whatis (license: unknown) (emulation).
[+] Version information found (GNU coreutils) 9.1 in binary /usr/bin/l (license: GPL-3.0-only) (emulation).
[+] Version information found (XZ Utils) 5.4.1 in binary /usr/bin/lzmainfo (license: unknown) (emulation).
[+] Version information found avahi-daemon 0.8 in binary /usr/sbin/avahi-daemon (license: LGPL-2.0-or-later) (emulation).
[+] Version information found start-stop-daemon 1.22.6 in binary /usr/sbin/start-stop-daemon (license: unknown) (emulation).
[+] Version information found bash, version 5.2.15 in binary /usr/bin/bash (license: GPL-3.0-only) (emulation).
[+] Version information found ischroot, version 5.7 in binary /usr/bin/ischroot (license: unknown) (emulation).
[+] Version information found sdpool - SDP tool v5.66 in binary /usr/bin/sdpool (license: unknown) (emulation).
[+] Version information found ZipNote 3.0 in binary /usr/bin/zipnote (license: unknown) (emulation).
[+] Version information found logrotate 3.21.0 in binary /usr/sbin/logrotate (license: unknown) (emulation).
[+] Version information found iwconfig Wireless-Tools version 30 in binary /usr/sbin/iwconfig (license: unknown) (emulation).
[+] Version information found gzip 1.12 in binary /usr/bin/gzip (license: unknown) (emulation).
[+] Version information found ZipSplit 3.0 in binary /usr/bin/zipsplit (license: bsd-style) (emulation).
[+] Version information found GNU Make 4.3 in binary /usr/bin/make (license: unknown) (emulation).
[+] Version information found (GNU cpio) 2.13 in binary /usr/bin/cpio (license: GPL-3.0-only) (emulation).
[+] Version information found tune2fs 1.47.0 in binary /usr/sbin/tune2fs (license: unknown) (emulation).
[+] Version information found dumpe2fs 1.47.0 in binary /usr/sbin/dumpe2fs (license: unknown) (emulation).
```

検出したソフトウェアバージョンのリスト

cwe-checkerによる危険なコード検出

- cwe-checkerはオープンソースの静的解析ツール
- arm,x86,MIPSなどマルチアーキテクチャ対応
- ELFバイナリに対してC/C++言語のCWEに該当する危険なコードパターンを検出する

```
[+] cwe-checker found a total of 2829 and 13 different security issues in libX11.so.6.4.0 (common linux file: no):  
CWE119 - Buffer Overflow - 4 times.  
CWE125 - Out-of-bounds Read - 20 times.  
CWE134 - Externally Controlled Format String - 23 times.  
CWE190 - Integer Overflow or Wraparound - 27 times.  
CWE252 - Unchecked Return Value - 49 times.  
CWE415 - Double Free - 26 times.  
CWE416 - Use After Free - 585 times.  
CWE467 - Use of sizeof on a Pointer Type - 21 times.  
CWE476 - NULL Pointer Dereference - 1389 times.  
CWE676 - Use of Potentially Dangerous Function - 610 times.  
CWE782 - Exposed IOCTL with Insufficient Access Control - 2 times.  
CWE787 - Out-of-bounds Write - 9 times.  
CWE789 - Large memory allocation - 64 times.
```

ライブラリに対して解析が行われた結果

GhidraとSemgrepを利用した脆弱性解析

- バイナリをGhidraでデコンパイルして疑似コードを生成
- 疑似コードに対して静的解析ツールSemgrepを適用することでバイナリのみでソースコードレベルの脆弱性解析を実現
- 参考：<https://security.humanativaspa.it/automating-binary-vulnerability-discovery-with-ghidra-and-semgrep/>



Ghidraとsemgrepを利用した脆弱性解析

- ファイルシステムに含まれているライブラリに対して脆弱性解析を行った結果

```
[+] Found 311 issues in native binary libhistory.so.8.2 (common linux file: no)
[+] Found 1088 issues in native binary libksba.so.8.14.3 (common linux file: no)
[+] Found 1547 issues in native binary aarch64-linux-gnu-objdump (common linux file: no)
[+] Found 1253 issues in native binary libreadline.so.8.2 (common linux file: no)
[+] Found 3906 issues in native binary libX11.so.6.4.0 (common linux file: no)
[+] Found 5177 issues in native binary libbfd-2.40-system.so (common linux file: no)
[+] Found 3454 issues in native binary libsystemd-core-252.so (common linux file: no)
[+] Found 1080 issues in native binary libslang.so.2.3.3 (common linux file: no)
[+] Found 2775 issues in native binary libicuuc.so.72.1 (common linux file: no)
[+] Found 9280 issues in native binary libsystemd-shared-252.so (common linux file: no)
[+] Found 28072 issues in native binary libpython3.11.so.1.0 (common linux file: no)
```

どのライブラリで脆弱性が検出されたか

```
[+] Found 57943 possible vulnerabilities (via semgrep on Ghidra decompiled code) in 15 tested binaries:
command-injection - 2 times.
double-free - 16 times.
format-string-bugs - 808 times.
incorrect-unsigned-comparison - 135 times.
incorrect-use-of-free - 54 times.
incorrect-use-of-memset - 4 times.
incorrect-use-of-strncpy-stncpy-strncpy - 2 times.
insecure-api-access-stat-lstat - 131 times.
insecure-api-sprintf-vsprintf - 81 times.
insecure-api-strcpy-stpcpy-strcat - 680 times.
integer-truncation - 27629 times.
integer-wraparound - 599 times.
interesting-api-calls - 22431 times.
off-by-one - 3 times.
pointer-subtraction - 661 times.
signed-unsigned-conversion - 4076 times.
unchecked-ret-malloc-calloc-realloc - 419 times.
unchecked-ret-setuid-setuid - 1 times.
unsafe-ret-sprintf-vsprintf - 11 times.
unterminated-string-strncpy-stncpy - 169 times.
use-after-free - 31 times.
```

脆弱性の種類と検出された回数

GhidraとSemgrepを利用した脆弱性解析

- ブラウザで脆弱性が検出された部分の疑似コード確認ができる

```
[+] Identified source function: /logs/s16_ghidra_decompile_checks/haruspex_libhistory.so.8.2/FUN_00107d7c_00107d7c.c
Semgrep rule: external.semgrep-rules-0xdea.c.raptor-integer-truncation
Issue description:
Truncation errors occur when a primitive is cast to a primitive of a smaller size and data is lost in the conversion. The value cannot be trusted and the application will be in an undefined state.
15 - iVar4 = _rl_get_char_len(param_4 + param_5, param_6)

[+] Identified source function: /logs/s16_ghidra_decompile_checks/haruspex_libhistory.so.8.2/copy_history_entry_00102ed4.c
Semgrep rule: external.semgrep-rules-0xdea.c.raptor-interesting-api-calls
Issue description:
Locate all calls to interesting and potentially insecure API functions (candidate points). The auditor can backtrace from these candidate points to find pathways allowing access from untrusted input.
15 - pcVar2 = (char *)xmalloc(sVar3 + 1)

[+] Identified source function: /logs/s16_ghidra_decompile_checks/haruspex_libhistory.so.8.2/copy_history_entry_00102ed4.c
Semgrep rule: external.semgrep-rules-0xdea.c.raptor-insecure-api-strcpy-strncpy-strcat
Issue description:
A buffer overflow condition exists when a program attempts to put more data in a buffer than it can hold, or when a program attempts to put data in a memory area outside of the boundaries of a buffer.
16 - pcVar2 = strcpy(pcVar2, (char *)param_1[1])

[+] Identified source function: /logs/s16_ghidra_decompile_checks/haruspex_libhistory.so.8.2/copy_history_entry_00102ed4.c
Semgrep rule: external.semgrep-rules-0xdea.c.raptor-interesting-api-calls
Issue description:
Locate all calls to interesting and potentially insecure API functions (candidate points). The auditor can backtrace from these candidate points to find pathways allowing access from untrusted input.
16 - pcVar2 = strcpy(pcVar2, (char *)param_1[1])
```

```
long copy_history_entry(undefined8 *param_1)
{
    long iVar1;
    char *pcVar2;
    size_t sVar3;
    undefined8 uVar4;

    if (param_1 != (undefined8 *)0x0) {
        iVar1 = alloc_history_entry(*param_1, 0);
        pcVar2 = (char *)param_1[1];
        if (pcVar2 != (char *)0x0) {
            sVar3 = strlen(pcVar2);
            pcVar2 = (char *)xmalloc(sVar3 + 1);
            pcVar2 = strcpy(pcVar2, (char *)param_1[1]);
        }
        uVar4 = param_1[2];
        *(char **)(iVar1 + 8) = pcVar2;
        *(undefined8 **)(iVar1 + 0x10) = uVar4;
        return iVar1;
    }
    return 0;
}
```


SBOMについて

- Cyclone DX SBOMを出力する
- 結果を見るとCyclone DXのroot-level elementsであるmetadata, components, dependencies, vulnerabilitiesの4つがある
- NTIAが定義するSBOM最小要素は満たしていると思われる

NTIA SBOM最小要素

Supplier Name

Component Name

Version of the Component

Other Unique Identifiers

Dependency Relationship

Author of SBOM Data

Timestamp

```
{
  "$schema": "http://cyclonedx.org/schema/bom-1.5.schema.json",
  "bomFormat": "CycloneDX",
  "specVersion": "1.5",
  "serialNumber": "urn:uuid:199a2ce7-8ebe-4d2d-a222-8cb3a5589657",
  "version": 1,
  "metadata": { ...
},
  "components": [ ...
],
  "dependencies": [
],
  "vulnerabilities": []
}
```

components

```
{
  "type": "library",
  "name": "openssl",
  "version": "3.0.16-1~deb12u1+rpt1",
  "supplier": {
    "name": "Debian OpenSSL Team <pkg-openssl-devel@alioth-lists.debian.net>"
  },
  "group": "debian_pkg_mgmt",
  "bom-ref": "b26d5802-cf47-4830-8087-7d0f4711a3bb",
  "scope": "required",
  "cpe": "cpe:2.3:a:openssl:openssl:3.0.16-1~deb12u1+rpt1:*:*:*:*:*:*:*",
  "purl": "pkg:deb/debian/openssl@3.0.16-1~deb12u1+rpt1?arch=arm64&distro=debian-12",
  "properties": [...],
  "hashes": [...],
  "description": "EMBA SBOM-group: debian pkg mgmt - name: openssl - version: 3.0.16-1~deb12u1+rpt1"
}
```

metadata

```

"metadata": {
  "timestamp": "2025-09-09T19:29:11+00:00",
  "tools": {
    "components": [
      {
        "type": "application",
        "author": "EMBA community",
        "name": "EMBA binary analysis environment",
        "version": "1.5.2-a46ccc6d86c3efb0d68edc2f6e3677948c3374ec",
        "description": "EMBA firmware analyzer - https://github.com/e-m-b-a/emba"
      }
    ]
  }
},

```

SBOMについて

- 出力フォーマットもXML,JSON,PROTOBUF,CSVと多数対応
- CycloneDXからSPDXにも変換可能
- SPDXに変換した場合はauthorやコンポーネントの依存関係(Dependency Relationship)は失われる

NTIA SBOM最小要素

Supplier Name

Component Name

Version of the
Component

Other Unique Identifiers

Dependency Relationship

Author of SBOM Data

Timestamp

```
"spdxVersion": "SPDX-2.2",  
"creationInfo": {  
  "comment": "This SPDX document has been converted from CycloneDX format.",  
  "created": "2025-09-09T19:29:11Z"  
},
```

packages

```
{  
  "SPDXID": "SPDXRef-b26d5802-cf47-4830-8087-7d0f4711a3bb",  
  "checksums": [...],  
  "copyrightText": "NOASSERTION",  
  "description": "EMBA SBOM-group: debian_pkg_mgmt - name: openssl - version: 3.0.16-1~deb12u1\u002Brpt1 - descrip",  
  "downloadLocation": "NOASSERTION",  
  "homepage": "NOASSERTION",  
  "licenseConcluded": "NOASSERTION",  
  "licenseDeclared": "NOASSERTION",  
  "licenseInfoFromFiles": [  
    "NOASSERTION"  
  ],  
  "name": "openssl",  
  "originator": "NOASSERTION",  
  "supplier": "Person: Debian OpenSSL Team \u003Cpkg-openssl-devel@alioth-lists.debian.net\u003E ()",  
  "versionInfo": "3.0.16-1~deb12u1\u002Brpt1"  
},
```


まとめと感想

- EMBAは組込みLinuxに対する脆弱性解析をメインとしたツール
- デコンパイルやエミュレーション技術、OSSを活用して解析を行っている
- SPDXの変換には課題はあるが、今後改善していく可能性はある

感想

- バイナリのみで脆弱性の検出とSBOMの生成を実現しようとしているオープンソースのツールは珍しい
- ソースコードもSBOMも提供されないソフトウェアに対して独自にSBOMを生成する手段の一つとなる
- 提供されたSBOMの妥当性を検証する手段としても利用できる
- 今回の発表はEMBAを使ってみただけで、結果の正確性については確認していません

ご清聴ありがとうございました