



# IoTcube 2.0 Hatbom

# 사용자 메뉴얼

V0.0.3

2025-09-25



## [도구별 지원언어]

※ 도구 별 지원 언어(2025.9.8 현재) → 도구별 지원 언어는 점진적으로 확대 적용예정임

SBOM	OSS Dependency Graph	Vulnerability	Static Analysis
C/C++, java, python, go, php	C/C++	C/C++, java, python	C/C++

①SBOM Step      ② Vulnerability Step      ③VEX Step

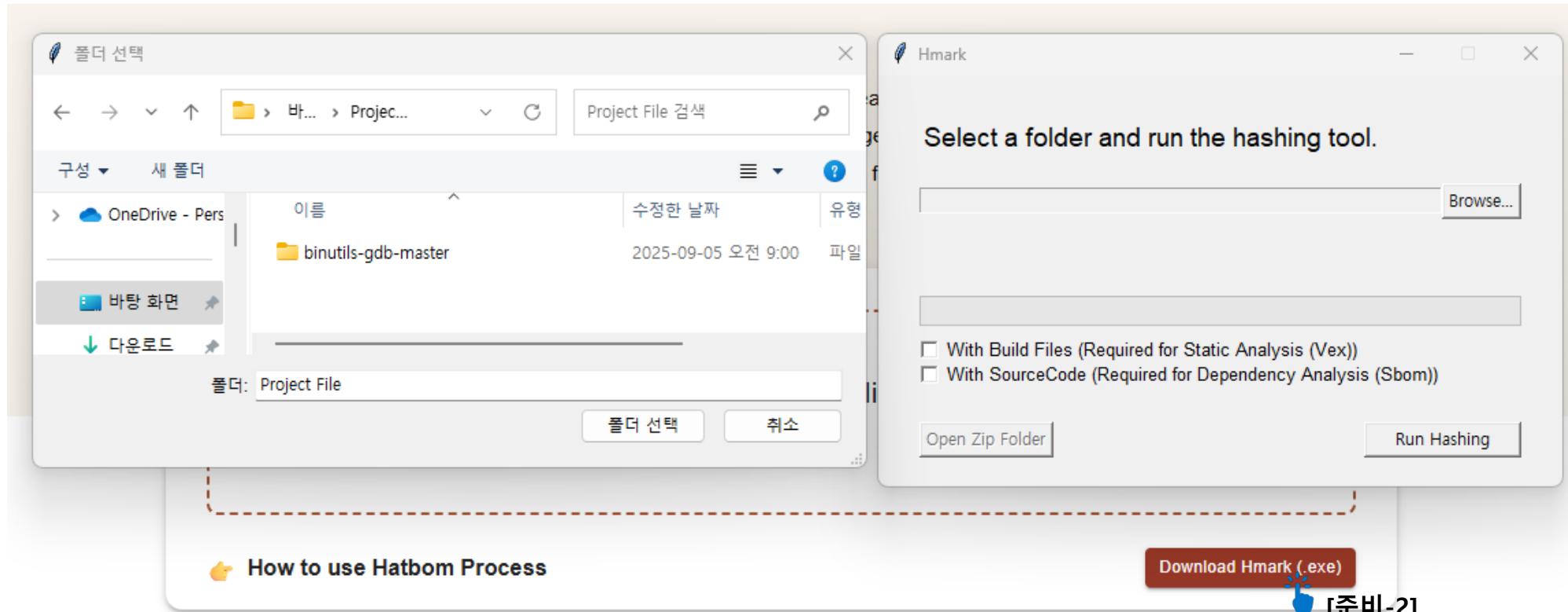
## [준비]프로젝트 zip 파일 Drag&Drop

[준비-1] 프로젝트 코드가 있는 프로젝트 파일을 .zip 파일로 압축해서 Drag&Drop

The screenshot shows the IoTcube Hatbom web application. At the top, there's a navigation bar with links: HatDB, Statistics, Contact us, IoTCube 1.0, and CSSA. Below the navigation, there's a large input field with a dashed border, containing a blue hand icon and the text "[준비-1] Drag & drop .zip files, or click to select". Below this field, it says "Accepted: .zip only". Above this input field, a file selection dialog is open, showing a list of files. One file, "binutils-gdb-master.zip", is selected and highlighted. The dialog has buttons for "열기(O)" (Open) and "취소" (Cancel). In the bottom right corner of the main page, there are two buttons: "How to use Hatbom Process" and "Download Hmark (.exe)".

# [준비] 프로젝트 zip 파일 Drag&Drop

[준비-2] Download Hmark로 Hashing 진행 후 나온 .zip 파일을 Drag&Drop

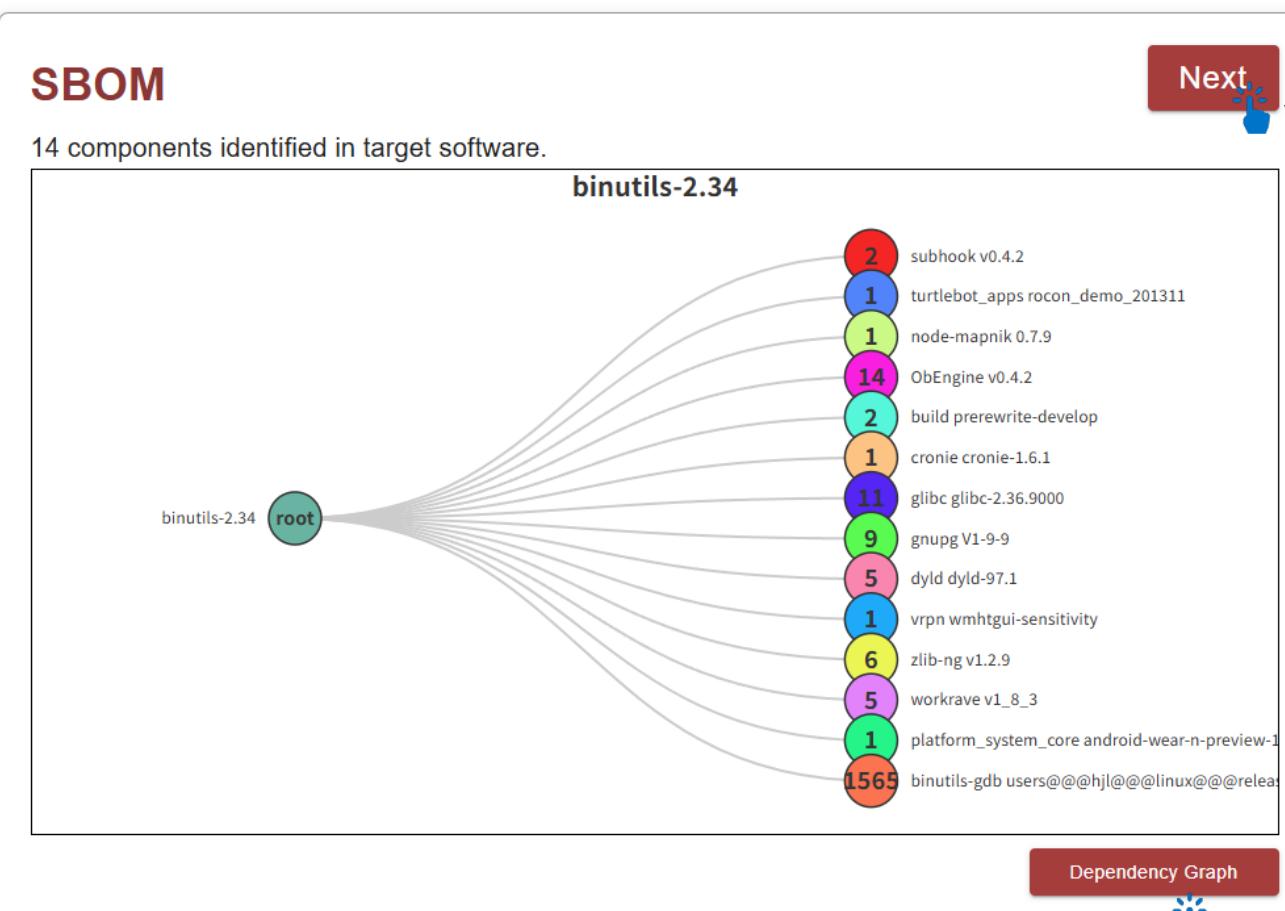


- Hmark는 로컬에서 Hashing 하는 프로그램으로 소스코드를 플랫폼에 제공하지 않아도 되는 장점이 있습니다(**코드 프라이버시**)
- With Build Files를 체크하시면 정적분석에 필요한 바이너리 파일이 포함됩니다.(C/C++ build 완료된 프로젝트 전용)
- With SourceCode를 체크하시면 종속성 그래프 생성에 필요한 **소스코드도 포함**이 됩니다.(C/C++ 전용)



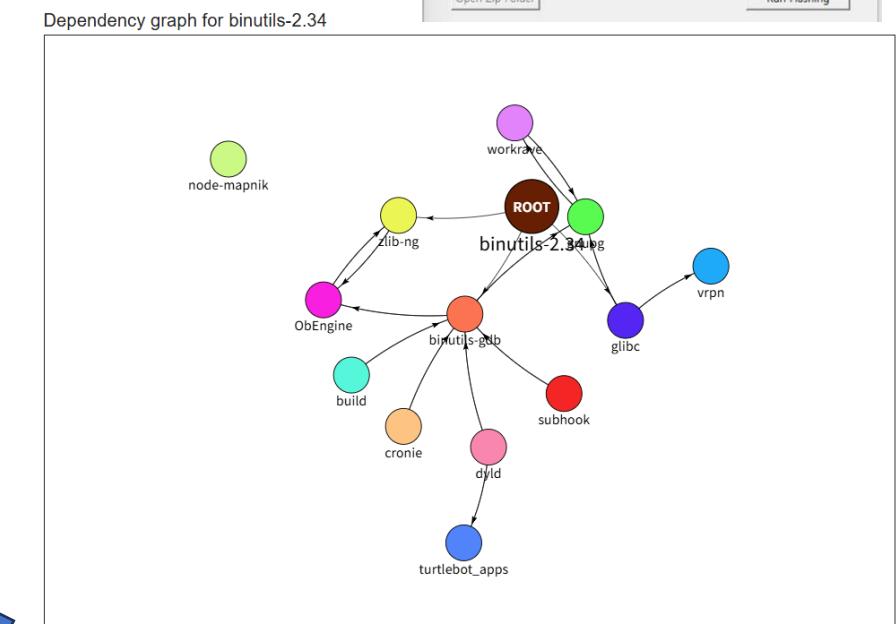
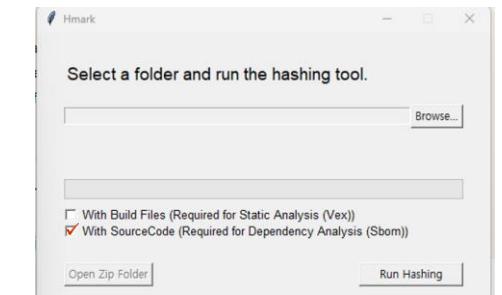
# 1. SBOM Step

1 SBOM      2 Vulnerability      3 VEX      4 Result



SBOM Step은 OSS탐지와 그에 따른 SBOM 문서 자동생성 Step입니다.  
하단의 다운로드 버튼을 통해서 SBOM 문서를 받을 수 있습니다.

- C/C++의 경우 소스코드 제공시 OSS의 Dependency도 확인 할 수 있습니다.



Dependency Graph



# 1. SBOM Step

**Result Details**

File Name	binutils-2.34
Files	1624
Dependencies	14
Input Format	ZIP File
Output Format	CycloneDX format SBOM

 **SBOM Download**

```
binutils-2.34_SBOM.json x
C:\Users\kl204\Desktop\{}\binutils-2.34_SBOM.json ...
1 {
2   "sbom": {
3     "bomFormat": "CycloneDX",
4     "specVersion": "1.4",
5     "serialNumber": "urn:uuid:9879a4d7-0a08-4025-e376-c617d97f9629",
6     "version": 1,
7     "metadata": {
8       "timestamp": "2025-09-05T05:42:55.346628+00:00",
9       "authors": [
10         {
11           "name": "IoTcube - https://iotcube.net"
12         }
13       ],
14       "component": {
15         "group": "",
16         "name": "binutils-2.34",
17         "version": "",
18         "type": "application",
19         "bom-ref": "pkg:generic/binutils-2.34",
20         "purl": "pkg:generic/binutils-2.34"
21       }
22     },
23     "dependencies": [
24       {
25         "ref": "binutils-2.34",
26         "dependsOn": [
27           "subhook v0.4.2",
28         ]
29       }
30     ]
31   }
32 }
```



## 2. Vulnerability Step

1 SBOM ————— 2 Vulnerability ————— 3 VEX ————— 4 Result

**Vulnerability**

The number of functions to be analyzed exceeds 10,000. If you require a more extensive analysis, please contact us or reach out to the CSSA office.

Detected 45 vulnerable code clones (2 kinds of CVE) in your package.

#Detected vulnerable code clones  
**45**

#Detected unique CVEs  
**2**

Rank of Top 3 Vulnerable Files

Rank	Name	Count
1	binutils/arlex.c	9
2	binutils/deflex.c	9
3	binutils/syslex.c	9

Rank of Top 3 CVE

Rank	Name	Count
1	<b>CVE-2019-16866</b>	25
2	<b>CVE-2019-18934</b>	20

Next 

Vulnerability Step은 취약점을 탐지하는 Step입니다.  
제공한 프로젝트에 어떤 취약점이 있는지 확인 할 수 있습니다.

- CVSS(Common Vulnerability Scoring System)는 취약점의 점수입니다.  
등급 구간: None(0), Low(0.1–3.9), Medium(4.0–6.9), High(7.0–8.9), Critical(9.0–10.0) 으로 산출됩니다.
- VEX문서 생성을 위해서는 탐지된 cve 들을 체크해주어야 합니다.

### VEX Step

VUDDY Vulnerable Files

id	File Path	CVE	CVSS ▲	KEV ⓘ	<input type="checkbox"/>
1	gas/bfin-lex.c	CVE-2019-18934	Medium	None	<input checked="" type="checkbox"/>
2	binutils/deflex.c	CVE-2019-18934	Medium	None	<input checked="" type="checkbox"/>
3	binutils/deflex.c	CVE-2019-18934	Medium	None	<input checked="" type="checkbox"/>
4	binutils/arlex.c	CVE-2019-18934	Medium	None	<input type="checkbox"/>
5	binutils/deflex.c	CVE-2019-16866	Medium	None	<input type="checkbox"/>
6	gas/bfin-lex.c	CVE-2019-18934	Medium	None	<input type="checkbox"/>
7	gas/itbl-lex.c	CVE-2019-16866	Medium	None	<input type="checkbox"/>
8	binutils/arlex.c	CVE-2019-16866	Medium	None	<input type="checkbox"/>
9	binutils/arlex.c	CVE-2019-16866	Medium	None	<input type="checkbox"/>
10	binutils/deflex.c	CVE-2019-16866	Medium	None	<input type="checkbox"/>



### 3. VEX Step

1 SBOM      2 Vulnerability      3 VEX      Result

**VEX**  
Here is your VEX report based on the selected packages and identified vulnerabilities.

**VEX Document Preview**

```
1 {  
2   "@context": "https://openvex.dev/ns/v0.2.0",  
3   "@id": "https://openvex.dev/docs/example/vex-ae55cf45-c144-4334-90da-c4808fc3cb5e",  
4   "author": "Hatbox",  
5   "role": "Document Creator",  
6   "timestamp": "2025-09-05T06:25:28.174Z",  
7   "version": 1,  
8   "statements": [  
9     {  
10       "vulnerability": {  
11         "name": "CVE-2019-16866"  
12       },  
13       "products": [  
14         {  
15           "@id": "binutils-2.34@v0.1null - yyensure_buffer_stack"  
16         }  
17       ],  
18       "status": "affected",  
19       "justification": "-",  
20       "impact_statement": "",  
21       "statusNodes": "The static analysis tool determined this is reachable."  
22     },  
23     {  
24       "vulnerability": {  
25         "name": "CVE-2019-16866"  
26       }  
27     }  
28   }  
29 }
```

**Result Step**

**Next**

**Detected CVE List**

Index	CVE	Products	Status	Actions
1	CVE-2019-16866	binutils-2.34@v0.1null - yyensure_buffer_stack	affected	
2	CVE-2019-16866	binutils-2.34@v0.1null - yyensure_buffer_stack	affected	
3	CVE-2019-16866	binutils-2.34@v0.1null - yyensure_buffer_stack	affected	
4	CVE-2019-16866	binutils-2.34@v0.1null - yyensure_buffer_stack	affected	
5	CVE-2019-16866	binutils-2.34@v0.1null - yyensure_buffer_stack	affected	

**+ New CVE Document** **CVE Download All** **Recover CVE List**

Rows per page: 5 ▾ 1–5 of 190 < >

**VEX file download**

**OpenVEX** Recommended Structured VEX Format

**download VEX Document**

**VDR** Vulnerability Data Repository

**CSAF** Security Advisory Framework

**CycloneDX** SBOM Standard Format

**190**  
Total vulnerabilities (vulnerable code clones)

55 Affected    90 Not Affected    - Fixed    45 Under Investigation

VEX Step은 취약점의 유효성을 문서화시키는 Step입니다. 수정할 수 있는 곳이 있어 각 취약점의 디테일한 내용들을 추가 할 수 있습니다.

- C/C++의 프로젝트의 경우 build가 된 프로젝트이면 정적 분석이 자동으로 실행됩니다.



## 4. Result Step

전체적인 step 요약을 확인 할 수 있습니다.

1 SBOM      2 Vulnerability      3 VEX      4 Result

### Result

This dashboard provides a comprehensive summary of your project's SBOM, vulnerabilities, and VEX results. Easily track your project's security status and identify key areas for improvement. All the results are presented in an organized, user-friendly format for quick analysis and decision-making.

**File Name:** binutils-2.34  
**Tool Vendor:** CSSA Korea Univ.  
**Product Vendor:** Test Product Vendor  
**Author:** Test Author

#### SBOM Summary

14	OSS Component
1624	Files
14	Dependencies

#### hidx File

Input Format

[Back to Start](#)

#### Vulnerability Summary

45	Total Vulnerable files	2	Total CVEs						
0	Critical	0	High	45	Medium	0	Low	0	None

#### VEX Summary

190	Total CVEs						
55	Affected	90	Not Affected	45	Under Investigation	0	Fixed