

[소프트웨어보안연구소(Center for Software Security and Assurance, CSSA)]

Hatbom 사용자 가이드

[Step I : SBOM, Step II : Vulnerability, Step III : VEX, Step IV : Result]

고려대학교 소프트웨어보안연구소(CSSA)
2025-8-26

1. Hatbom 개요

고려대학교 소프트웨어보안연구소(CSSA, Center for Software Security and Assurance)에서 개발한 “보안취약점 자동분석 플랫폼”이다.

1-1. IoTcube 1.0

IoTcube 1.0은 2016년 4월 19일에 공개된 소프트웨어의 ‘보안 취약점 자동분석 플랫폼’서비스의 초기버전으로

- 소프트웨어에 내재된 보안 취약점을 식별하고 분석하는 도구
- 보안 전문가가 아닌 누구라도 소스파일을 ‘드래그 앤 드롭’으로 쉽게 사용
- 이외에도 IoT 기기를 비롯한 다양한 기기에 탑재되는 소프트웨어의 보안을 관리할 수 있도록 도구를 제공

1-2. IoTcube 2.0

IoTcube 2.0은 기존 1.0 버전에서 한 단계 발전하여 다음과 같은 기능 개선을 하여 2025년 8월 26일에 Hatbom이라는 이름으로 공개되었다.

- 다양한 분석 기능을 SBOM·VEX 중심으로 개편
- 소프트웨어만 넣으면 끝나는 원스톱 사용자 친화적 프로세스
- 고려대 연구팀의 강점을 반영, 구성요소·의존성 시각화 제공
- DB 및 서버 최적화로 더 빠르고 편리한 분석 환경

2. Hatbom 주요 기능

Hatbom에는 SBOM 자동생성, 취약점(Vulnerability) 탐지 그리고 탐지된 취약점의 관리 지침으로 활용할 수 있는 VEX 문서 생성 등 3 가지 주요한 기능으로 구성되어 있다.

각 기능별로 지원가능한 언어는 다음과 같으며 점차 확대해 나갈 계획이다.

- 지원 가능한 언어

	① SBOM Step		② Vulnerability Step	③ VEX Step
	SBOM	Dependency Graph	CVE vulnerabilities	Static Analysis
지원 언어	C/C++, Java, Python, GO, PHP	C/C++	C/C++, Java, Python	C/C++

2.1 SBOM Generation Step

소프트웨어를 업로드하면 SBOM 문서를 자동 생성할 수 있다.



[그림 1] SBOM 생성

현재 버전에서는 입력으로 소스코드만 지원하며, 점차로 바이너리, 도커 이미지까지 확대할 예정이다.

- 지원 가능한 언어: C/C++, Java, Python, go, php 언어(추후 추가 예정)

※ 소프트웨어의 프라이버시를 고려하여 로컬에서 해싱(Hashing)하는 프로그램으로 소프트웨어를 해시화하여 플랫폼으로 업로드하는 기능도 제공하고 있다.(Hmark 프로그램)

[분석 순서]

- 1) <https://iotcube.net> 을 입력

- 2) 프로젝트코드가 있는 프로젝트 파일을 .zip 파일로 압축해서 Drag & Drop 한다(①, ②, ③)



3) 분석이 완료되면 다음과 같은 결과가 표시된다.

Source Code

Hatbom VUDOV VEX

This is the Source Code Upload Type SBOM/Vulnerability List/VEX Generation Page.
Our Core Engines are highly optimized and efficient in generating SBOM, Vulnerability Lists, and VEX Generation Pages from source code.

1 SBOM 2 Vulnerability 3 VEX 4 Result

SBOM

14 components identified in target software.

binutils-2.34

Dependency Graph

Result Details

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

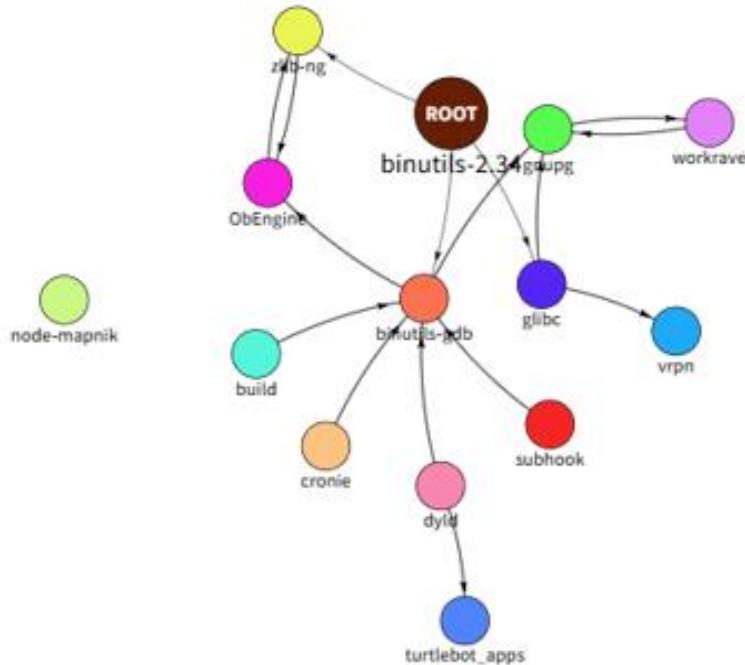
SBOM Download

Description

- 1) binutils-2.34 프로젝트 파일에는 14 개의 오픈소스 컴포넌트들이 포함되어 있다.
- 2) 14 개의 컴포넌트 노드들을 각각 클릭해 보면 재사용된 파일들의 목록을 볼 수 있다.
- 3) <Result Details>
분석에 사용된 파일과 분석 결과에 대한 각종 정보를 보여준다.
- 4) [④ Dependenc Graph]를 클릭하면 탐지된 컴포넌트들 간의 의존성을 확인할 수 있다.
- 5) [⑤ SBOM Download]를 클릭하여 생성된 SBOM 을 저장할 수 있다.

④ 탐지된 컨포넌트들간의 의존성 가시화 그래프

4



⑤ 생성된 SBOM(CycloneDX)

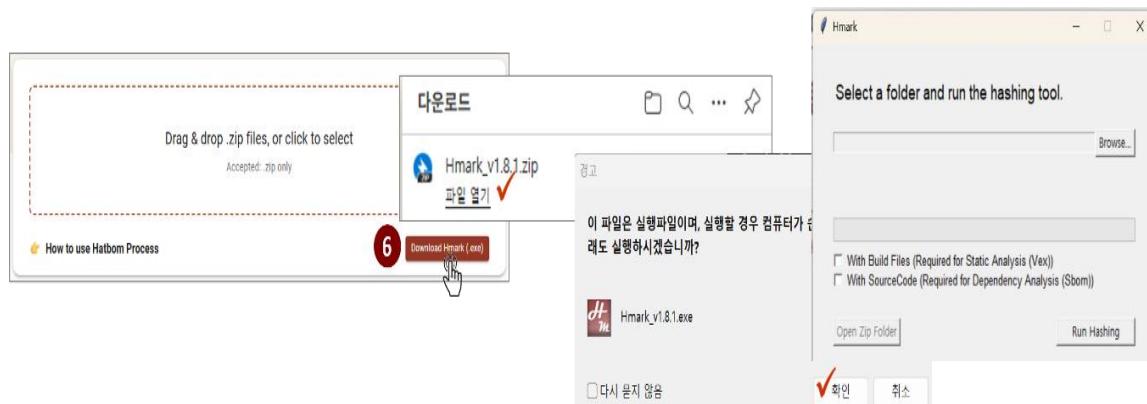
5 SBOM(CycloneDX)

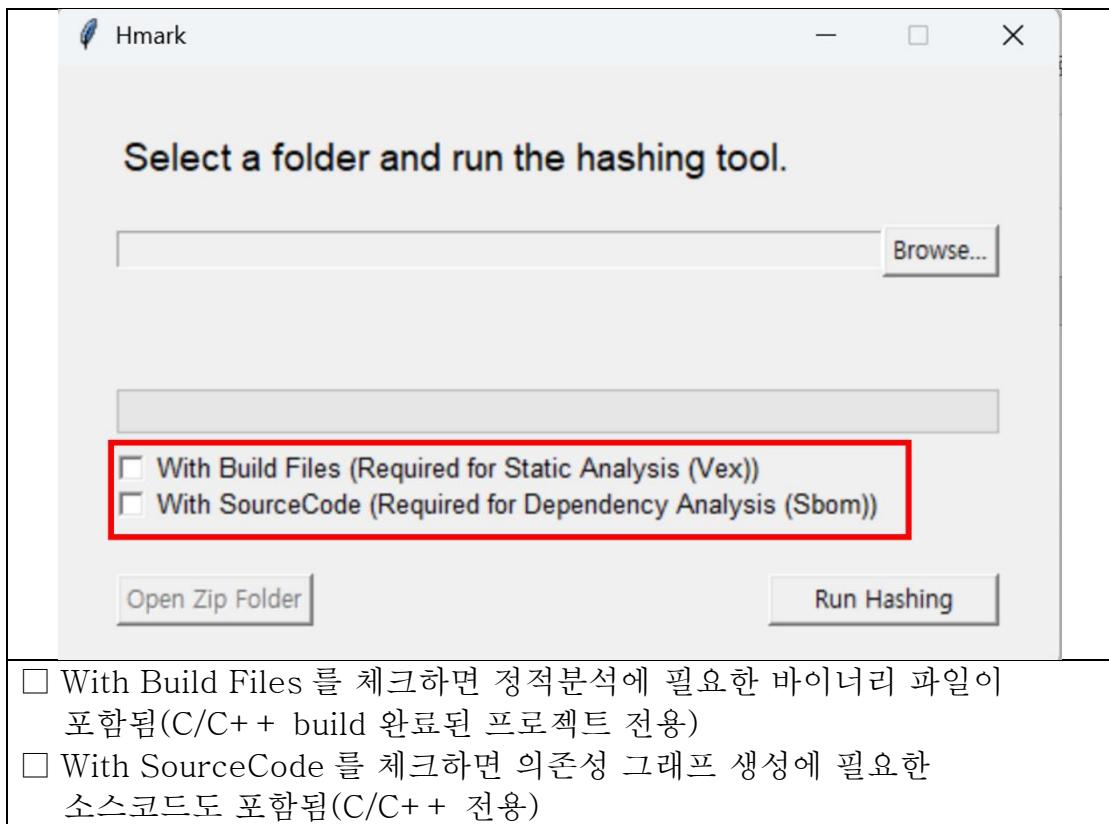
```
{  
  "sbom": {  
    "bomFormat": "CycloneDX",  
    "specVersion": "1.4",  
    "serialNumber": "urn:uuid:fca4f21b-9660-8b0b-0b47-fe0d5ea9e126",  
    "version": 1,  
    "metadata": {  
      "timestamp": "2025-09-26T05:52:42.627685+00:00",  
      "authors": [  
        {  
          "name": "IoTcube - https://iotcube.net"  
        }  
      ]  
    },  
    "component": {  
      "group": "",  
      "name": "binutils-2.34",  
      "version": "",  
      "type": "application",  
      "bom-ref": "pkg:generic/binutils-2.34",  
      "purl": "pkg:generic/binutils-2.34"  
    },  
    "dependencies": [  
      {  
        "ref": "binutils-2.34",  
        "dependsOn": [  
          "subhook v0.4.2",  
          "turtlebot_apps rocon_demo_201311",  
          "node-mapnik 0.7.9",  
          "ObEngine v0.4.2",  
          "build prerewrite-develop",  
          "cronie cronie-1.6.1",  
          "glibc glibc-2.36.9000",  
          "gnupg V1-9-9",  
          "dyld dyld-97.1",  
          "vrpn wmhtgui-sensitivity",  
          "zlib-ng v1.2.9",  
        ]  
      }  
    ]  
  }  
}
```

4) 코드 프라이버시를 위한 프로젝트 파일의 해시화

분석하고자 하는 프로젝트 파일을 그대로 업로드하지 않고 해시화하는 프로그램을 다운받아 로컬에서 해시화하여 해시화 된 파일을 업로드하는 방법

- 아래 Download Hmark 를 다운로드하여 로컬에서 Hashing 진행 한다.





- With Build Files 를 체크하면 정적분석에 필요한 바이너리 파일이 포함됨(C/C++ build 완료된 프로젝트 전용)
- With SourceCode 를 체크하면 의존성 그래프 생성에 필요한 소스코드도 포함됨(C/C++ 전용)

- 해시화하여 나온 .zip 파일을 Drag & Drop 한다.



- 이 이후 프로세스는 앞의 순서 ③부터는 동일하다.

Notice! : 소스코드 파일과 해시파일로 분석 시 차이점

Process	SBOM		Vulnerability	VEX	비고
	SBOM	Dependency Graph	Vulnerability	Static Analysis	
Source Code	○	○	○	X	VEX 문서 에디팅은 Static Analysis 와 상관없이 무조건 제공
Hmark	○	X	○	(바이너리 빌드 존재 시) ○	

2.2 Vulnerability Step

이 단계는 취약점을 탐지하는 단계이다. 제공한 프로젝트에 어떤 취약점이 있는지 확인할 수 있다. 아래 그림과 같이 ⑨ 또는 ⑩을 클릭하여 취약점 탐지를 진행한다.

Source Code

(Red) (Blue) (VEX)

This is the Source Code Upload Type SBOM/Vulnerability List/VEX Generation Page.
Our Core Engines are highly optimized and efficient in generating SBOM, Vulnerability Lists, and VEX Generation Pages from source code.

1 SBOM 2 Vulnerability 3 VEX 4 Result

SBOM

14 components identified in target software.

binutils-2.34

Dependency Graph

Result Details

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

SBOM Download

9 10

Vulnerability

Next

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	CVE-2019-16866	25
2	CVE-2019-18934	20



1



1



VUDDY Vulnerable Files

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



1 2 3 4 5



Description
<ol style="list-style-type: none">1) 총 45 개의 파일에서 취약점이 탐지됨2) 탐지된 취약점은 CVE-2019-16866 과 CVE-2019-18934 2 종류의 취약점임3) 위의 2 가지 취약점을 많이 갖고 있는 Top3 파일은 arlex.c 와 deflex.c 그리고 syslex.c 파일이다.4) CVSS(Common Vulnerability Scoring System)는 취약점의 위험도 점수로 0~10 점으로 평가<ul style="list-style-type: none">- None(0)- Low(0.1~3.9)- Medium(4.0~6.9)- High(7.0~8.9)- Critical(9.0~10.0)5) VEX 문서 생성을 위해서는 탐지된 CVE 들을 선택해 줘야 함

2.3 VEX Step

앞에서 선택한 CVE 들의 유효성을 직접 개발자들이 분석 및 검증할 수 있도록 하였다. C/C++ 의 프로젝트의 경우, build 가 된 프로젝트이면 정적분석이 자동으로 수행되어 그 결과를 함께 제공하고 있다.

Source Code

This is the Source Code Upload Type SBOM/Vulnerability List/VEX Generation Page.
Our Core Engines are highly optimized and efficient in generating SBOM, Vulnerability Lists, and VEX Generation Pages from source code.

1 SBOM 2 Vulnerability 11 VEX Result

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 CSA office.

Vulnerability

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 of Top 3 CVE		
Rank	Name	Count	Rank	Name	Count
1	binutils/aflxx.c	9	1	CVE-2019-16934	25
2	binutils/deflxx.c	9	2	CVE-2019-16934	20
3	binutils/syslxx.c	9			

< 1 > < 1 >

VUDDY Vulnerable Files

ID	File Path	CVE	CVSS	KEV	Selected
1	binutils/deflxx.c	CVE-2019-16934	Medium	None	<input checked="" type="checkbox"/>
2	binutils/aflxx.c	CVE-2019-16936	Medium	None	<input checked="" type="checkbox"/>
3	binutils/aflxx.c	CVE-2019-16934	Medium	None	<input checked="" type="checkbox"/>
4	gaa/rtbHex.c	CVE-2019-16936	Medium	None	<input checked="" type="checkbox"/>
5	binutils/aflxx.c	CVE-2019-16936	Medium	None	<input checked="" type="checkbox"/>
6	binutils/syslxx.c	CVE-2019-16934	Medium	None	<input checked="" type="checkbox"/>
7	gaa/rtbHex.c	CVE-2019-16936	Medium	None	<input checked="" type="checkbox"/>
8	binutils/syslxx.c	CVE-2019-16934	Medium	None	<input checked="" type="checkbox"/>
9	binutils/syslxx.c	CVE-2019-16936	Medium	None	<input checked="" type="checkbox"/>
10	binutils/deflxx.c	CVE-2019-16936	Medium	None	<input checked="" type="checkbox"/>

< 1 2 3 4 5 >

11 12 13 14

- ⑪ 또는 ⑫를 클릭하여 VEX Step 으로 진입하기 전에 개발자들이 취약점 단계에서 탐지된 CVE 들의 정보(⑬)를 참조하여 VEX 로 넘겨줄 CVE 들을 선택한다(⑭)
- 통상 CVSS 값들이 Low, Medium, High, Critical 중에서, Medium 또는 High 이상인 값을 갖는 CVE 들에 우선순위를 두고 분석함
- VEX 로 넘겨줄 CVE 들을 선정하였다면 VEX Step 으로 이동한다(⑪ 또는 ⑫를 클릭)

VEX

Next

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

VEX Document Preview

15

Result Lock

```
1 v {
2   "@context": "https://openvex.dev/ns/v0.2.0",
3   "@id": "https://openvex.dev/docs/example/vex-fbf3a08f-f2f2-4a07-b206-b4a177c21463",
4   "author": "Habom",
5   "role": "Document Creator",
6   "timestamp": "2025-09-28T06:42:02.479Z",
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     },
22     {
23       "vulnerability": {
24         "name": "CVE-2019-16866"
25       },
26       "products": [
27         {
28           "@id": "binutils-2.34@v0.1null - yyensure_buffer_stack"
29         }
30       ],
31       "status": "Not Affected",
32       "justification": "-",
33       "impact_statement": "-"
34     }
35   ]
36 }
```

190

Total vulnerabilities (vulnerable code clones)

55

Affected

90

Not Affected

45

Under Investigation

Detected CVE List

16

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 VDI Format

17



VDR

Vulnerability Data Repository



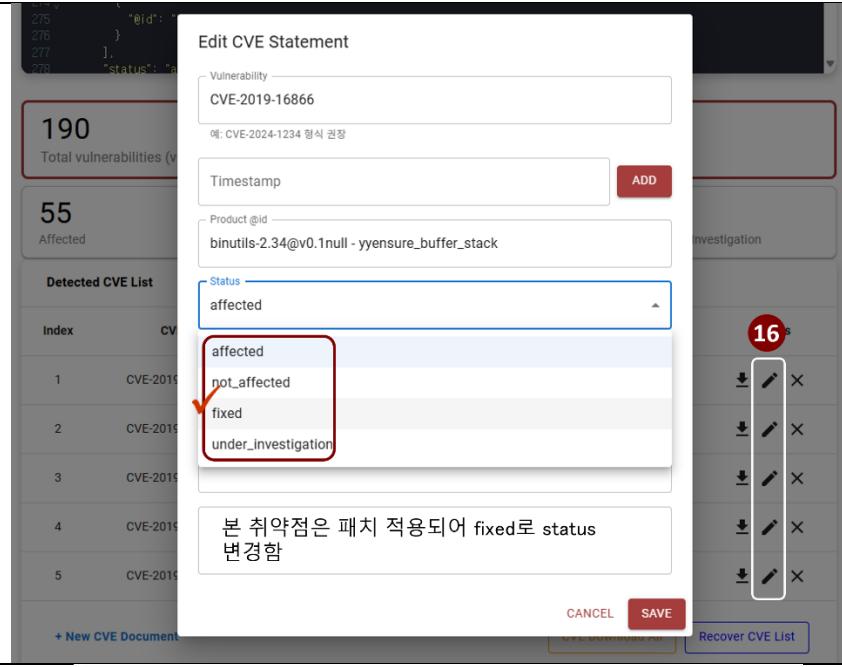
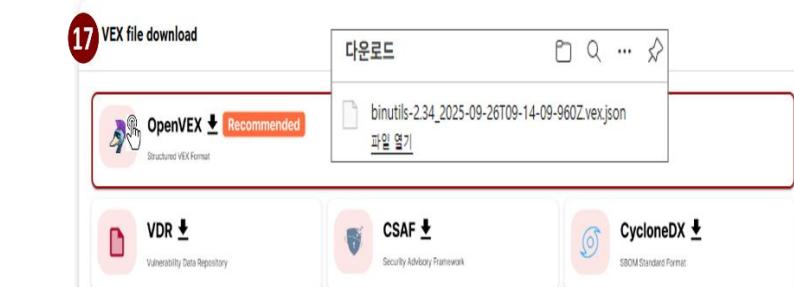
CSAF

Security Advisory Framework



CycloneDX

SБOM Standard Format

Description															
<p>15</p> <pre> 167 "vulnerability": [168 "name": "CVE-2019-16866" 169 }, 170 "products": [171 { 172 "@id": "binutils-2.34@v0.1null - yyensure_buffer_stack" 173 } 174], 175 "status": "affected", 176 "justification": "-", 177 "impact_statement": "-" 178 }, 179 "vulnerability": [180 "name": "CVE-2019-16866" 181 }, 182 "products": [183 { 184 "@id": "binutils-2.34@v0.1null - yyensure_buffer_stack" 185 } 186], 187 "status": "affected", 188 "justification": "-", 189 "impact_statement": "-" </pre>	<p>(15) 탐지된 취약점 목록에서 선택된 CVE 들에 대해 해당되는 CVE 들에 대한 Status 들이 기입되어 있다.</p> <p>개발자들이 최종적으로 각각의 선택된 취약점에 대해 분석 및 검증한 후 최종적으로 각 CVE 들에 대한 Status 값들을 결정한다.</p>														
 <p>190 Total vulnerabilities (v)</p> <p>55 Affected</p> <p>Detailed CVE List</p> <table border="1"> <thead> <tr> <th>Index</th> <th>CVE</th> </tr> <tr> <td>1</td> <td>CVE-2019-16866</td> </tr> <tr> <td>2</td> <td>CVE-2019-16866</td> </tr> <tr> <td>3</td> <td>CVE-2019-16866</td> </tr> <tr> <td>4</td> <td>CVE-2019-16866</td> </tr> <tr> <td>5</td> <td>CVE-2019-16866</td> </tr> </thead> <tbody> <tr> <td>+ New CVE Document</td> <td></td> </tr> </tbody> </table> <p>본 취약점은 패치 적용되어 fixed로 status 변경함</p>	Index	CVE	1	CVE-2019-16866	2	CVE-2019-16866	3	CVE-2019-16866	4	CVE-2019-16866	5	CVE-2019-16866	+ New CVE Document		<p>각각의 CVE 들에 대해 Status 값들을 변경할 필요가 있다고 판단되었을 때는 해당 CVE 에 대한 수정 아이콘(16)을 클릭한 후 왼쪽과 같이 4 가지 status list 중에서 선택한 후 수정할 수 있다.</p>
Index	CVE														
1	CVE-2019-16866														
2	CVE-2019-16866														
3	CVE-2019-16866														
4	CVE-2019-16866														
5	CVE-2019-16866														
+ New CVE Document															
<p>17</p> <p>VEX file download</p> 	<p>(17) 최종적으로 모든 CVE status 값들이 조정된 후에는 VEX 문서를 다운로드 할 수 있다.</p> <ul style="list-style-type: none"> - 현재는 OpenVEX 포맷을 우선 지원함 														

2.4 Result Step

지금까지 분석한 내용들을 summary 하여 보여주는 기능이다.

⑯ 또는 ⑰를 클릭하여 Result Step 으로 이동한다.

The screenshot shows a software interface for managing security vulnerabilities. At the top, there is a navigation bar with three steps: 1. SBOM, 2. Vulnerability, 3. VEX, and 18. A blue hand icon with the number 19 and the word 'Next' is positioned above the VEX step. To the right of the VEX step is a 'Result' icon with a document symbol.

VEX

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

VEX Document Preview

```
253      "Justification": "-",
254      "Impact_Statement": "-"
255      ,
```

Result Lock (checkbox)

Result

1. SBOM — 2. Vulnerability — 3. VEX — 18 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.

SBOM Summary

14	OSS Component
----	---------------

1624	Files
------	-------

14	Dependencies
----	--------------

hidx File
Input Format

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
-------------	-----------------	------------------------	---------

3. 향후 계획(Future Plan)

- Hatbom 지원 언어 추가 : Typescript, Ruby(~2025년 말)
- 바이너리 기반의 SBOM 생성 및 취약점 탐지 도구 공개
- 컨테이너 이미지 가반의 SBOM 생성 및 취약점 탐지 도구 공개

※ 문의처: cssa@korea.ac.kr