|  |
| --- |
|  |
| **Nmap NSE를 활용한 CVE 취약점 진단** |
|  |

|  |  |  |
| --- | --- | --- |
|  | 팀명: | 모의해킹26기X팀 |
|  | **이름:** | **주대원** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| File Name | Nmap NSE를 활용한 CVE 취약점 진단 |
| 원안작성자 | 주대원 |
| 수정작업자 | 주대원 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정 날짜 | 대표 수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2020.02.24 | 주대원 | 0.1 | 원안작성 | 보고서 초안 작성 |
| 2020.02.25 | 주대원 | 0.2 | Nmap | Nmap, Nmap NSE 개요 작성 |
| 2020.02.27 | 주대원 | 0.3 | 패킷 분석 | 패킷 분석 작성 |
| 2020.02.28 | 주대원 | 0.4 | 소스코드 분석 | 소스코드 분석 작성 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

표 ‑ 문서 정보 / 수정 내역

목 차

[1 개요 6](#_Toc33755599)

[1.1 프로젝트 주제 6](#_Toc33755600)

[1.2 프로젝트 추진 배경 및 목표 6](#_Toc33755601)

[1.3 프로젝트 요약 6](#_Toc33755602)

[2 Nmap 7](#_Toc33755603)

[2.1 Nmap 개요 7](#_Toc33755604)

[2.2 Nmap 활용 7](#_Toc33755605)

[3 Nmap NSE 10](#_Toc33755606)

[3.1 Nmap NSE 개요 10](#_Toc33755607)

[3.2 Nmap NSE 카테고리 10](#_Toc33755608)

[3.3 Nmap-vulners.nse 분석 11](#_Toc33755609)

[3.3.1 개요 11](#_Toc33755610)

[3.3.2 CVE 취약점 진단 12](#_Toc33755611)

[3.3.3 패킷 분석 14](#_Toc33755612)

[3.3.4 소스코드 분석 16](#_Toc33755613)

[4 참고 문헌 21](#_Toc33755614)

[4.1 단행본 21](#_Toc33755615)

[4.2 참조 홈페이지 21](#_Toc33755616)

표 목차

[표 1‑1 문서 정보 / 수정 내역 2](#_Toc33755617)

[표 1‑1 프로젝트 주제 6](#_Toc33755618)

[표 1‑2 프로젝트 추진 배경 및 목표 6](#_Toc33755619)

[표 1‑3 프로젝트 요약 6](#_Toc33755620)

[표 2‑1 nmap 명령어 7](#_Toc33755621)

[표 2‑2 nmap 옵션 8](#_Toc33755622)

[표 3‑1 NSE 카테고리 11](#_Toc33755623)

[표 3‑2 테스트 환경 12](#_Toc33755624)

[표 3‑3 nmap-vulners 명령어 12](#_Toc33755625)

[표 3‑4 주석 16](#_Toc33755626)

[표 3‑5 카테고리 17](#_Toc33755627)

[표 3‑6 모듈 17](#_Toc33755628)

[표 3‑7 get\_results 함수 18](#_Toc33755629)

[표 3‑8 /api/v3/burp/software 결과 중 일부 19](#_Toc33755630)

[표 3‑9 make\_links 함수 20](#_Toc33755631)

[표 4‑1 단행본 21](#_Toc33755632)

[표 4‑2 참조 홈페이지 21](#_Toc33755633)

그림 목차

[그림 2‑1 nmap 7](#_Toc33755634)

[그림 2‑2 -sT 옵션 8](#_Toc33755635)

[그림 2‑3 -p <포트번호> 옵션 8](#_Toc33755636)

[그림 2‑4 -sV 옵션 9](#_Toc33755637)

[그림 2‑5 -T1 옵션 9](#_Toc33755638)

[그림 2‑6 -T5 옵션 9](#_Toc33755639)

[그림 3‑1 /usr/share/nmap/scripts 10](#_Toc33755640)

[그림 3‑2 nmap-vulners 설치 11](#_Toc33755641)

[그림 3‑3 스크립트 업데이트 11](#_Toc33755642)

[그림 3‑4 취약점 진단 12](#_Toc33755643)

[그림 3‑5 취약점 목록 13](#_Toc33755644)

[그림 3‑6 확장자 변경 13](#_Toc33755645)

[그림 3‑7 result.html 13](#_Toc33755646)

[그림 3‑8 TCP half open 스캔 14](#_Toc33755647)

[그림 3‑9 대기 14](#_Toc33755648)

[그림 3‑10 배너 정보 획득 14](#_Toc33755649)

[그림 3‑11 bind version 9.4.2 14](#_Toc33755650)

[그림 3‑12 SSL 연결 14](#_Toc33755651)

[그림 3‑13 질의 문 15](#_Toc33755652)

[그림 3‑14 질의 결과 중 일부 15](#_Toc33755653)

# 개요

## 프로젝트 주제

|  |
| --- |
| 1. Nmap NSE를 활용한 CVE 취약점 진단 |

표 ‑ 프로젝트 주제

## 프로젝트 추진 배경 및 목표

|  |
| --- |
| 1. Nmap NSE를 활용해 CVE 취약점 진단 및 소스코드 분석 |

표 ‑ 프로젝트 추진 배경 및 목표

## 프로젝트 요약

|  |
| --- |
| 1. Nmap NSE를 활용해 CVE 취약점 진단 및 소스코드 분석 |

표 ‑ 프로젝트 요약

# Nmap

## Nmap 개요



그림 ‑ nmap

엔맵(Nmap)은 1997년 고든 라이온이 제작한 포트 스캔과 취약점 진단을 위한 도구이다. 사용자는 엔맵을 사용해 시스템에서 실행 중인 장치를 식별하고 열린 포트를 찾을 수 있다.

## Nmap 활용

|  |
| --- |
| **명령어** |
| nmap <옵션> <대상 IP> |

표 ‑ nmap 명령어

엔맵을 사용하기 위한 명령어는 표 2‑1과 같다.

|  |  |
| --- | --- |
| **옵션** | **설명** |
| -sT | TCP 포트 스캔 |
| -sS | TCP Half open 스캔 |
| -sP | Ping 스캔 |
| -sU | UDP 포트 스캔 |
| -sN | TCP Null 스캔 |
| -sF | TCP Fin 스캔 |
| -sX | TCP Xmas 스캔 |
| -sV | TCP Half open 스캔을 이용하여 대상 호스트에 어떤 버전의 프로그램이 실행 중인지 확인 |

|  |  |
| --- | --- |
| -O | 대상 호스트의 OS 판별 |
| -o | 스캔 결과를 텍스트 파일로 저장 |
| -oX | 스캔 결과를 XML 파일로 저장 |
| -p <포트번호> | 특정 포트를 지정하여 스캔 |
| -T <숫자> | 스캔 하는데 걸리는 시간 조절하며 숫자가 높을수록 시간이 적게  걸림 (기본값 3) |
| -A | 버전 정보와 스크립트를 활용한 정보들을 출력 |
| -h | 도움말 |

표 ‑ nmap 옵션

엔맵의 옵션은 표 2‑2와 같다.

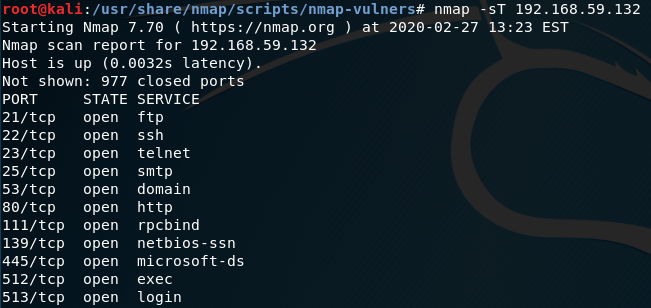


그림 ‑ -sT 옵션

TCP 포트 스캔 결과는 그림 2‑2와 같으며 열린 포트의 경우 해당 포트에 대한 정보가 출력되지만 닫힌 포트의 경우 출력되지 않는다.

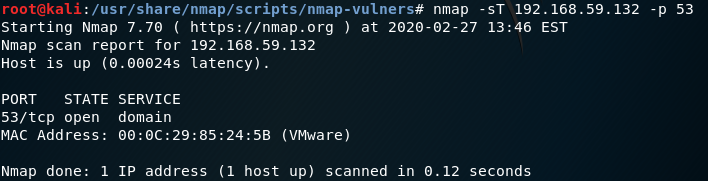


그림 ‑ -p <포트번호> 옵션

특정 포트에 대한 스캔 결과는 그림 2‑3과 같으며 해당 포트의 번호, 상태, 서비스 정보가 출력된다.

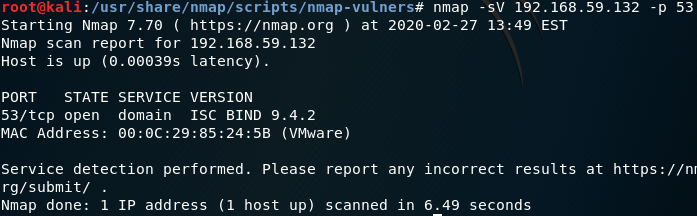


그림 ‑ -sV 옵션

대상 호스트에 대한 세부 정보 스캔 결과는 그림 2‑4와 같다. 그림 2‑2와 그림 2‑3과 달리 버전 정보가 추가되었다.

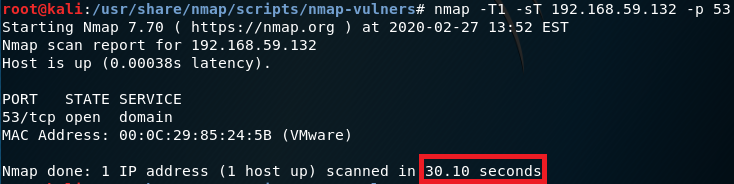


그림 ‑ -T1 옵션

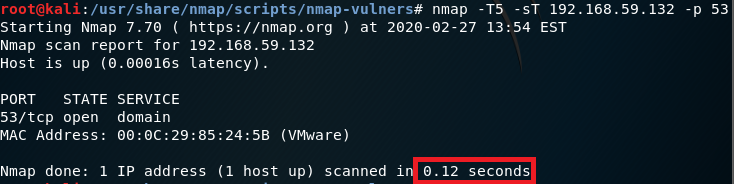


그림 ‑ -T5 옵션

-T <숫자> 옵션은 스캔 하는데 걸리는 시간을 조절한다. 그림 2‑5와 그림 2‑6을 비교하면 숫자 크기에 따른 시간 차를 확인할 수 있다.

# Nmap NSE

## Nmap NSE 개요

엔맵(Nmap)에는 기본으로 제공하는 표준 스크립트가 존재한다. 기본 기능을 이용해 다양한 정보를 수집할 수 있지만, 대상 서비스의 숫자가 늘어날수록 많은 시간이 소요되어 효율적인 분석을 하는데 어려움이 있다. 이러한 엔맵의 한계를 보완하기 위해 사용자는 NSE(Nmap Scripting Engine) 스크립트를 만들 수 있다. NSE는 Lua 프로그래밍 언어로 개발되며 무차별 대입 공격, 취약점 및 백도어 탐지 등의 기능을 수행할 수 있다.

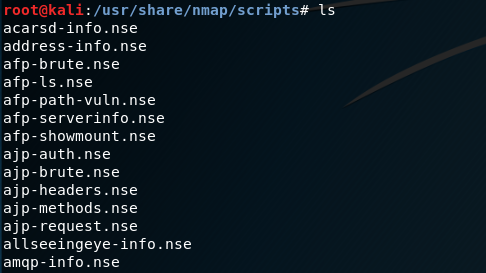


그림 ‑ /usr/share/nmap/scripts

NSE는 .nse 확장자로 저장되며 /usr/share/nmap/scripts 경로에 존재한다.

## Nmap NSE 카테고리

|  |  |
| --- | --- |
| **목록** | **설명** |
| auth | 인증 자격 증명 또는 우회 처리 |
| broadcast | 로컬 네트워크에서 브로드캐스팅 결과로 나오지 않은 호스트를 탐색 |
| brute | 무차별 대입 공격을 통해 원격 서버의 인증 자격 증명 |
| default | 기본 스크립트 |
| discovery | 공용 레지스트리, SNMP, 디렉터리 서비스 정보 등 네트워크 관련  정보 획득 |
| dos | 취약점 테스트를 위한 서비스 거부 공격 수행 |

|  |  |
| --- | --- |
| exploit | 취약점을 이용하여 공격 코드 실행 |
| external | 외부 서비스를 이용하여 결과 값을 얻음 |
| fuzzer | 패킷에 예기치 못하거나 무작위 필드를 담아 서버에 전송 |
| intrusive | 대상 시스템을 손상시키거나 중요 자원 사용 |
| malware | 대상 시스템에 백도어 설치나 악성코드 감염 여부 테스트 |
| safe | 시스템에 영향을 최소화하면서 정보 획득 |
| version | 버전 정보를 획득하며 -sV를 요청한 경우에만 실행 |
| vuln | 알려진 취약점에 대한 진단을 수행하며 찾은 결과를 보고 |

표 ‑ NSE 카테고리

NSE 스크립트는 수행하는 작업에 따라 표 3‑1처럼 분류되어 있다.

.

## Nmap-vulners.nse 분석

### 개요

nmap-vulners는 vulners.com API를 사용하여 취약점을 탐지하는 NSE 스크립트이다.

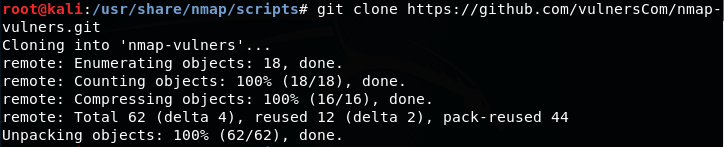


그림 ‑ nmap-vulners 설치

설치를 위해서 git clone https://github.com/vulnersCom/nmap-vulners.git 명령어를 입력한다.

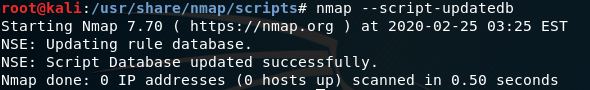


그림 ‑ 스크립트 업데이트

정상적으로 설치가 된 후 nmap --script-updatedb 명령어를 입력하여 스크립트를 업데이트한다.

### CVE 취약점 진단

|  |  |
| --- | --- |
| **구분** | **설명** |
| 가상 머신 | VMware Workstation 15 Pro |
| 가상 OS | Kali-Linux 2019.1 (공격자) > 192.168.59.130 |
| Linux metasploitable 2.6 (공격대상) > 192.168.59.132 |

표 ‑ 테스트 환경

CVE 취약점 진단을 위한 테스트 환경은 표 3‑2와 같다.

|  |
| --- |
| **명령어** |
| nmap -sV --script=nmap-vulners <대상 IP> |

표 ‑ nmap-vulners 명령어

nmap-vulners를 사용하기 위한 명령어는 표 3‑3과 같다.

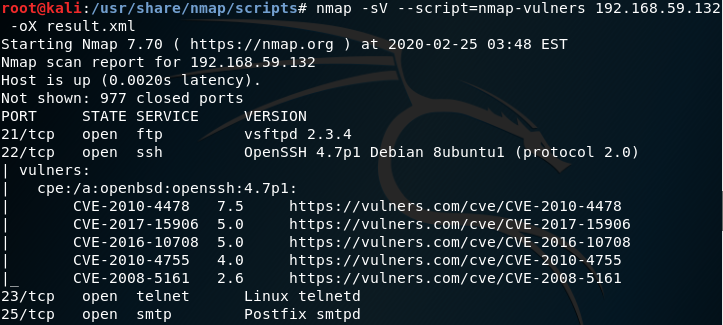


그림 ‑ 취약점 진단

nmap –sV --script=nmap-vulners 192.168.59.132 –oX result.xml 명령어로 공격 대상에 대한 취약점 진단을 수행하며 그 결과를 xml 파일로 저장했다.

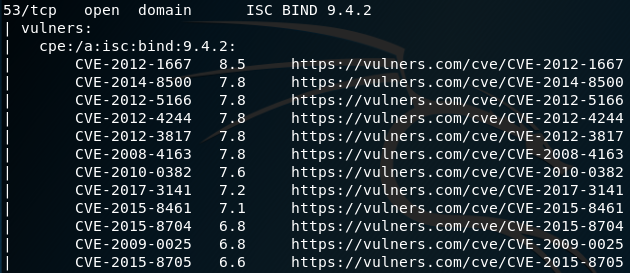


그림 ‑ 취약점 목록

취약점이 있는 포트에 대해서는 그림 3‑5와 같이 CVE 번호, CVSS 점수, 취약점 정보를 볼 수 있는 주소가 출력된다.



그림 ‑ 확장자 변경

그림 3‑4에서 저장한 xml 파일을 html로 변경하고 파이어폭스를 통해 연다.

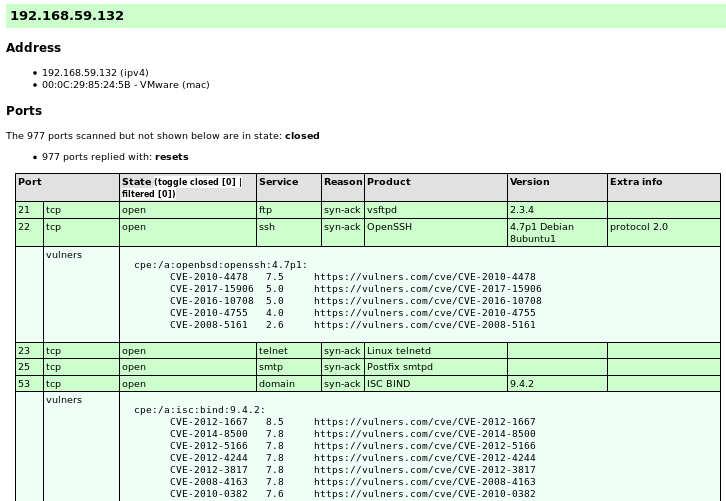


그림 ‑ result.html

html을 통해 취약점 진단 결과를 볼 수 있다.

### 패킷 분석



그림 ‑ TCP half open 스캔

nmap-vulners 스크립트를 실행하면 TCP half open 스캔을 통해 포트가 열려있는지 확인한다.



그림 ‑ 대기

포트가 열려있을 경우 다시 TCP 연결을 시도하고 6초 동안 아무 것도 보내지 않는다.



그림 ‑ 배너 정보 획득

상대방 호스트에서 배너 정보와 함께 응답이 온다.

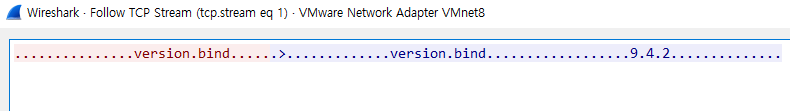


그림 ‑ bind version 9.4.2

bind version은 그림 3‑11과 같다.

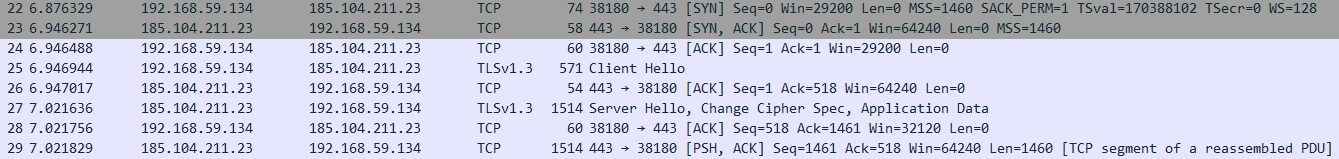


그림 ‑ SSL 연결

SSL(Secure Socket Layer) 연결을 통해 vulners.com API에 질의한다.

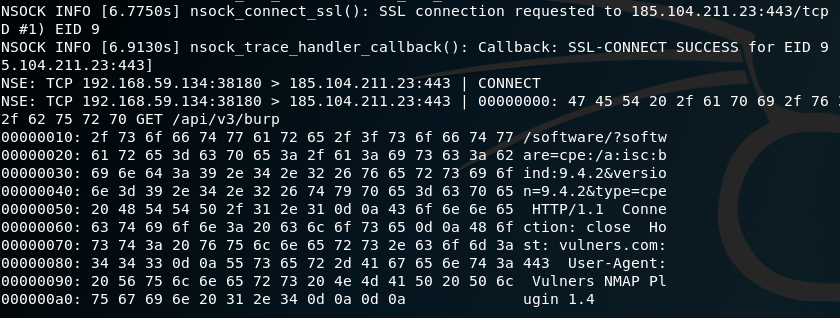


그림 ‑ 질의 문

vulners.com에 질의한 내용은 그림 3‑13과 같다.



그림 ‑ 질의 결과 중 일부

질의 문에 대한 결과 값은 그림 3‑14의 형태로 온다.

### 소스코드 분석

|  |
| --- |
| description = [[  For each available CPE the script prints out known vulns (links to the correspondent info) and correspondent CVSS scores.  Its work is pretty simple:  \* work only when some software version is identified for an open port  \* take all the known CPEs for that software (from the standard nmap -sV output)  \* make a request to a remote server (vulners.com API) to learn whether any known vulns exist for that CPE  \* if no info is found this way, try to get it using the software name alone  \* print the obtained info out  NB:  Since the size of the DB with all the vulns is more than 250GB there is no way to use a local db.  So we do make requests to a remote service. Still all the requests contain just two fields - the  software name and its version (or CPE), so one can still have the desired privacy.  ]]  ---  -- @usage  -- nmap -sV --script vulners [--script-args mincvss=<arg\_val>] <target>  --  -- @args vulners.mincvss Limit CVEs shown to those with this CVSS score or greater.  --  -- @output  --  -- 53/tcp open domain ISC BIND DNS  -- | vulners:  -- | ISC BIND DNS:  -- | CVE-2012-1667 8.5 https://vulners.com/cve/CVE-2012-1667  -- | CVE-2002-0651 7.5 https://vulners.com/cve/CVE-2002-0651  -- | CVE-2002-0029 7.5 https://vulners.com/cve/CVE-2002-0029  -- | CVE-2015-5986 7.1 https://vulners.com/cve/CVE-2015-5986 |

표 ‑ 주석

nmap-vulners.nse에 대한 사용법과 출력 결과는 주석을 통해 설명하고 있다.

|  |
| --- |
| categories = {"vuln", "safe", "external", "default"} |

표 ‑ 카테고리

NSE 카테고리는 표 3‑5와 같이 분류되어있다.

|  |
| --- |
| local http = require "http"  local json = require "json"  local string = require "string"  local table = require "table"  local nmap = require "nmap"  local stdnse = require "stdnse" |

표 ‑ 모듈

nmap-vulners 스크립트를 사용하기 위한 모듈을 불러온다.

|  |
| --- |
| function get\_results(what, vers, type)  local api\_endpoint = "https://vulners.com/api/v3/burp/software/"  local vulns  local option={  header={  ['User-Agent'] = string.format('Vulners NMAP Plugin %s', api\_version)  },  any\_af = true,  }  local response = http.get\_url(('%s?software=%s&version=%s&type=%s'):format(api\_endpoint, what, vers, type), option)  local status = response.status  if status == nil then  return  elseif status ~= 200 then  return  end  status, vulns = json.parse(response.body)  if status == true then  if vulns.result == "OK" then  return make\_links(vulns)  end  end  end |

표 ‑ get\_results 함수

get\_results 함수는 vulners.com의 API를 이용하여 cpe 이름, 버전, 타입을 질의한 결과 값을 make\_links 함수에 반환한다.

|  |
| --- |
| {  "result": "OK",  "data": {  "search": [  {  "\_index": "es6\_bulletins\_bulletin",  "\_type": "bulletin",  "\_id": "CVE-2008-4163",  "\_score": 21.187466,  "\_source": {  "lastseen": "2019-05-29T18:09:28",  "bulletinFamily": "NVD",  "description": "Unspecified vulnerability in ISC BIND 9.3.5-P2-W1, 9.4.2-P2-W1, and 9.5.0-P2-W1 on Windows allows remote attackers to cause a denial of service (UDP client handler termination) via unknown vectors.",  "modified": "2017-08-08T01:32:00",  "id": "CVE-2008-4163",  "href": "https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-4163",  "published": "2008-09-22T18:52:00",  "title": "CVE-2008-4163",  "type": "cve",  "cvss": {  "score": 7.8,  "vector": "AV:N/AC:L/Au:N/C:N/I:N/A:C"  }  }  } |

표 ‑ /api/v3/burp/software 결과 중 일부

cpe 정보를 질의한 결과 값은 표 3‑8과 같다.

|  |
| --- |
| function make\_links(vulns)  local output = {}  if not vulns or not vulns.data or not vulns.data.search then  return  end  for \_, vuln in ipairs(vulns.data.search) do  local v = {  id = vuln.\_source.id,  type = vuln.\_source.type,  -- Mark the exploits out  is\_exploit = vuln.\_source.bulletinFamily:lower() == "exploit",  -- Sometimes it might happen, so check the score availability  cvss = tonumber(vuln.\_source.cvss.score),  }  -- NOTE[gmedian]: exploits seem to have cvss == 0, so print them anyway  if v.is\_exploit or (v.cvss and mincvss <= v.cvss) then  setmetatable(v, cve\_meta)  output[#output+1] = v  end  end  if #output > 0 then  -- Sort the acquired vulns by the CVSS score  table.sort(output, function(a, b)  return a.cvss > b.cvss or (a.cvss == b.cvss and a.id > b.id)  end)  return output  end |

표 ‑ make\_links 함수

get\_results 함수에 의해 호출된 make\_links 함수는 API를 통해 얻은 값 중 cve 아이디, 타입 및 cvss 점수를 저장하며 cvss 점수는 내림차순으로 정렬한다.

# 참고 문헌

## 단행본

|  |  |  |
| --- | --- | --- |
| 도서명 | 저자 | 출판사 |
| Nmap NSE를 활용한 슬로우로리스(Slowloris)  Dos 공격 진단 이해 | 임이진, 조정원 | 브이메이커스㈜ |

표 ‑ 단행본

## 참조 홈페이지

|  |
| --- |
| 참조 홈페이지 |
| https://4lugin.tistory.com/136  (nmap 옵션 정보)  https://securitytrails.com/blog/banner-grabbing  (배너 그랩에 대한 정보)  https://github.com/vulnersCom/nmap-vulners/blob/master/vulners.nse  (nmap-vulners 깃허브 홈페이지)  https://dbza.tistory.com/8  (nmap NSE 정보)  https://www.hackingarticles.in/understanding-nmap-packet-trace/  (nmap 패킷 추적 정보)  https://medium.com/@iphelix/nmap-scanning-tips-and-tricks-5b4a3d2151b3  (TCP Null probe 정보) |

표 ‑ 참조 홈페이지