**VUDDY Manual**

1. Requirement
   1. 운영체제

* 최소사양

Linux, macOS와 같은 POSIX 호환 운영체제 또는 Windows

* 주의사항

VUDDY는 POSIX 호환 운영체제에서 최고의 성능을 발휘합니다.

* 1. 의존 프로그램

- Python 2.7  
- Git  
- Diff  
- Java (POSIX 운영체제 한정)

1.2.1) Windows

Python 2.7의 최신버젼을 설치하십시오.  
<https://www.python.org/downloads/>

Git for Windows를 설치하여 git과 diff 바이너리를 얻을 수 있습니다.  
<https://git-for-windows.github.io/>

1.2.2) macOS

Python 2.7, Git과 Java를 설치하십시오.

Homebrew 패키지 매니저를 사용할 경우, 다음 명령으로 설치할 수 있습니다.

$ brew install python git  
$ brew cask install java

1.2.3) Linux

사용하고 있는 배포판의 패키지 매니저를 사용하여 Python 2.7, Git과 Java를 설치하십시오.

Ex) Debian, Ubuntu, Linux Mint  
$ sudo apt-get install python git openjdk-8-jre

* 1. config.py 설정

config.py 내에 git 저장소의 경로와 바이너리의 경로를 지정하여야 합니다.

1.3.1) gitStoragePath

git 저장소가 모여 있는 디렉터리입니다.

1.3.2) gitBinary, diffBinary, javaBinary

git, diff, java 바이너리의 경로입니다.  
명령 프롬프트 (Windows) 또는 터미널 (macOS, Linux)에서 각 바이너리를 곧바로 호출할 수 있는 경우, 별도의 변경이 필요하지 않습니다.

1.3.3) Windows 한정

- 경로 구분자로 단일 \ 대신 / 또는 \\를 사용하십시오.  
- Windows는 java 바이너리를 필요로 하지 않습니다.  
- Git이 설치된 경로가 C:\Program Files\Git일 경우, git과 diff 바이너리는 다음 경로에 설치됩니다.

gitBinary = "C:\\Program Files\\Git\\bin\\git.exe"  
diffBinary = "C:\\Program Files\\Git\\usr\\bin\\diff.exe"

* 1. CVE 데이터베이스 다운로드  
     NVDCVEcrawler 디렉터리 내의 oneclick.py를 실행하여 cvedata.pkl을 생성하십시오.  
     $ python oneclick.py

생성된 cvedata.pkl을 VUDDY 루트 리렉토리에 복사하십시오.

업데이트를 원할 경우, 추후 NVDCVEcrawler의 cveXmlUpdater.py를 실행하여 cvedata.pkl을 업데이트할 수 있습니다.

$ python cveXmlUpdater.py

1. 사용법

5가지 과정을 거쳐 한 프로젝트에 대한 hidx 파일을 생성할 수 있습니다.

이 때, 프로젝트의 소스 코드가 git으로 관리되고 있는 상태여야 합니다.

2.1) get\_cvepatch\_from\_git.py

프로젝트의 CVE 패치내역을 diff 파일로 변환합니다.

사용법  
$ python get\_cvepatch\_from\_git.py [-m] REPO

|  |  |
| --- | --- |
| REPO | Git 저장소의 이름을 지정합니다. |
| -m (선택) | 여러 저장소를 한 단위로 처리할 수 있는 Multimode를 활성화합니다.  예를 들어, Android를 구성하는 여러 Git 저장소를 “Android”라는 하나의 단위로 처리합니다.  “[gitStoragePath]/[저장소이름]” 경로에 여러 git 저장소를 구성한 뒤, “[VUDDY root dir]/repolists/list\_[저장소이름]” 경로에 저장소의 목록을 지정합니다. |

Ex) libtest란 프로젝트가 zlib과 libpng로 구성되어 있을 경우, 다음 예시와 같이 Multimode를사용할 수 있습니다.

|  |
| --- |
| [기본 경로]  VUDDY root directory = /home/ubuntu/vuddy  gitStoragePath = /home/ubuntu/gitrepos  [/home/ubuntu/gitrepos/repolists/libtest 디렉터리 구조]  libtest  ├─compress  │ └─zlib  └─image  └─libpng  [/home/ubuntu/vuddy /repolists/list\_libtest 파일]  compress/zlib  image/libpng  [터미널 명령어]  $ python get\_cvepatch\_from\_git.py -m libtest |

// 주의사항  
// Windows에서 병렬화가 지원되지 않아 POSIX에 비해 성능이 저하될 수 있습니다

2.2) get\_source\_from\_cvepatch.py

diff 파일에서 패치 전과 후의 소스코드 함수를 추출합니다.

사용법  
$ python get\_source\_from\_cvepatch.py [-m] REPO

|  |  |
| --- | --- |
| REPO | Git 저장소의 이름을 지정합니다. |
| -m (선택) | 여러 저장소를 동시에 처리할 수 있는 Multimode를 활성화합니다.  사용법은 get\_cvepatch\_from\_source.py와 동일합니다. |

// 주의사항  
// Windows에서 병렬화가 지원되지 않아 POSIX에 비해 성능이 저하될 수 있습니다

2.3) vul\_dup\_remover.py

gitStoragePath에 있는 모든 저장소의 CVE 패치 중복을 제거합니다.  
이 프로그램은 별도의 인자를 받지 않습니다.

사용법  
$ python vul\_dup\_remover.py

2.4) vul\_verifier.py

gitStoragePath에 있는 모든 저장소의 CVE 오탐을 제거합니다.  
이 프로그램은 별도의 인자를 받지 않습니다.

사용법  
$ python vul\_verifier.py

2.5) vul\_hidx\_generator.py

Git 저장소에서 hidx 해시를 생성합니다.

사용법  
$ python vul\_hidx\_generator.py -a ABSTRACTION\_LEVEL REPO

|  |  |
| --- | --- |
| -a ABSTRACTION\_LEVEL | 소스 코드의 추상화 여부를 지정합니다.  0 = 추상화를 사용하지 않음  4 = 추상화를 사용함 |
| REPO | Git 저장소의 이름을 지정합니다. |

2.6) 사용예시

/home/ubuntu/gitrepos에 openssl과 libressl 프로젝트의 git 저장소가 있을 때, config.py를 다음과 같이 수정합니다.

gitStoragePath = “/home/ubuntu/gitrepos/”

openssl과 libressl 프로젝트의 CVE 패치내역을 hidx로 변환하기 위해 다음 과정을 따릅니다.

$ python get\_cvepatch\_from\_git.py openssl   
$ python get\_source\_from\_cvepatch.py openssl

$ python get\_cvepatch\_from\_git.py libressl   
$ python get\_source\_from\_cvepatch.py libressl

$ python vul\_dup\_remover.py  
$ python vul\_verifier.py

$ python vul\_hidx\_generator.py openssl   
$ python vul\_hidx\_generator.py libressl

작업 후 [VUDDY root dir]/vul/hidx-vul 내에 각 프로젝트의 CVE 패치내역이 hidx 파일로 생성된 것을 확인할 수 있습니다.