Secudium Intelligence

API 정의서

Secudium Intelligence

Version 1.0

Published Date: 2017-12-14

SK infosec

1. Secudium Intelligence API 시작하기

1.1. Secudium Intelligence API

Secudium Intelligence에서 제공하는 다양한 분석 정보를 고객에게 전달 하기 위한 API의 종류 및 사용법을 정의한다.

1.2. 변경 이력

NO	버전	업데이트 날짜	변경 내역
1	Ver.0.1	2017.12.14	- 신규생성
2	Ver.0.9	2017.12.18	- 파일 행위분석 정보 API 추가 - 위협 분석 정보 API 추가 - URL/IP 체크 API GET 접속 허용 - 최신 CVE정보 API URL 변경 - 기타 내용 수정
3	Ver.1.0	2017.12.19	고객 배포용 Document 정의.
4			
5			

2. Secudium Intelligence API 주요 기능

2.1. Domain/URL Text 체크 API

- Text를 입력 받아 Domain / URL을 구분하고, 필요 시 각각에 대해 기본 DB 및 Correlation DB를 체크하고, 악성 여부 및 해당 정보를 요청 할 수 있는 API입니다.

2.2. IP Text 체크 API

- Text를 입력 받아 IP 추출 후, 기본 DB 및 Correlation DB를 체크하고, 악성 여부 및 해당 정보를 요청 할 수 있는 API입니다.

2.3. HASH Text 체크 API

- Text를 입력 받아 HASH 추출 후 기본 DB 및 Correlation DB를 체크하고 악성 여부 및 해당 정보를 요청 할 수 있는 API입니다.

2.4. File Upload & 분석 API

- 업로드 된 파일을 입력 받아 HASH를 추출하고 DB를 체크하여
  - DB 내 동일한 파일이 존재하는 경우 관련 정보를 요청
  - DB 내 동일한 파일이 존재하지 않는 경우 1차 분석(MAS, MAV) + 2차 분석(Correlation)을 수행하도록 절차 진행하여 해당 상태에 대한 value를 요청

  할 수 있는 API입니다.

2.5. URL 분석 API

- Text를 입력 받아 HASH 추출 후 DB를 체크하여
  - DB 내 정보가 있는 경우 악성 여부 및 해당 정보를 요청
  - DB 내 정보가 없는 경우 1차 분석(MAS, Web Crawling) + 2차 분석(Correlation)을 수행해야 하며, 해당 정보로 DB 업데이트 후 처리 여부에 대한 Flag를 요청

  할 수 있는 API입니다.

2.6. CVE 조회 API

- CVE Code를 입력 받아 해당 정보를 요청 할 수 있는 API입니다.

2.7. 최근 자주 사용되는 IP 정보(TOP10) API

- 최근 자주 사용되는 IP 정보 TOP 10을 요청 할 수 있는 API입니다.

2.8. 최근 자주 사용되는 HASH 정보(TOP10) API

- 최근 자주 사용되는 HASH 정보 TOP 10을 요청 할 수 있는 API입니다.

2.9. 최신 CVE 정보 (TOP10) API

- 최신 CVE 정보 TOP 10을 요청 할 수 있는 API입니다.

2.10. 보안에 관련된 최신 SNS 정보 API

- 보안에 관련된 최신 SNS 정보를 요청 할 수 있는 API입니다.

2.11. 보안에 관련된 최신 RSS 정보 API

- 보안에 관련된 최신 RSS 정보를 요청 할 수 있는 API입니다.

2.12. 최근 사용되는 악성 IP, File Hash 지리 정보 API

- 최근 사용되는 악성 IP, File Hash 지리 정보를 요청 할 수 있는 API입니다.

2.13. 최근의 높은 스코어 가진 CVE 정보 API

- 최근의 높은 스코어 가진 CVE 정보를 요청 할 수 있는 API입니다.

2.14. 파일 기본 정보 API

- 파일 기본 정보를 요청 할 수 있는 API입니다.

2.15. 파일 정적 분석 정보 API

- 파일 정적 분석 정보를 요청 할 수 있는 API입니다.

2.16. 파일 행위 분석 정보 API

- 파일 행위 분석 정보를 요청 할 수 있는 API입니다.

2.17. 위협 분석 정보 API

- 위협 분석 정보를 요청 할 수 있는 API입니다.

3. Secudium Intelligence API 세부 내용

3.1. Domain/URL Text 체크 API

- **Text를 입력 받아 Domain / URL을 구분하고, 필요 시 각각에 대해 기본 DB 및 Correlation DB를 체크하고, 악성 여부 및 해당 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

Domain/URL Text 체크 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/url/report.php
Method : GET
Parameters :
api_key = 사용자 API Key
url = 조회할 URL

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

Domain/URL Text 체크 응답 JSON

```
{
    "result_code" : 결과 코드 (Number),
    "result_msg" : 결과 메시지 (String),
    "date" : API 요청 시간 (String),
    "url" : 조회 요청한 URL (String),
    "secudiumti" : {
        "scan_date" : URL 엔진 검사 시간 (String),
        "scans" : {
            "ip" : URL Hostname에 대한 IP (String),
            "domain" : URL에 대한 도메인 (String),
            "hash" : 바이러스 파일명 hash (String),
            "type" : 바이러스(black) 및 피싱 도메인 구분 (String),
            "geo" :  IP에 대한 국가코드 (String),
            "Malicious Site" : 의심 사이트 탐지정보 {
                "detected" : 탐지 여부 (Boolean),
                "result" : 탐지 종류 (String)
            }
        }
    }
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'url': 'http://www.atoffshore.net/wp-includes/SimplePie/Content/Type'}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/url/report.php', params =params, verify=False)
print response.text
```

호출 결과 예제


```
{
    "date": "2017-12-14 11:43:44",
    "url": "http://www.atoffshore.net/wp-includes/SimplePie/Content/Type",
    "result_code": 1,
    "result_msg": "Data exists",
    "secudiumti": {
        "scan_date": "2017-10-10 09:30:05",
        "scans": {
            "ip": "205.134.249.11",
            "domain": "atoffshore.net",
            "hash": "-",
            "type": "phishing_domain",
            "geo": "US",
            "Malicious Site": {
                "result": "Phishing Domain",
```

```
                    "detected": true
                }
            }
        }
}
```

3.2. IP Text 체크 API

- **Text를 입력 받아 IP 추출 후, 기본 DB 및 Correlation DB를 체크하고, 악성 여부 및 해당 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

IP Text 체크 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/ip/report.php

Method : GET

Parameters :

api_key = 사용자 API Key

ip = 조회할 IP

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

IP Text 체크 응답 JSON

```
{
  "result_code" : 결과 코드 (Number),
  "result_msg" : 결과 메시지 (String),
  "date" : API 요청 시간 (String),
  "ip" : 조회 요청한 IP (String),
  "secudiumti" : {
      "scan_date" : URL 엔진 검사 시간 (String),
      "scans" : {
          "domain" : URL에 대한 도메인 (String),
          "url" : IP에 대한 URL (String),
          "hash" : 바이러스 파일명 hash (String),
          "type" : 바이러스(black) 및 피싱 도메인 구분 (String),
          "geo" :  IP에 대한 국가코드 (String),
          "Malicious Site" : 의심 사이트 탐지정보
            {
              "detected" : 탐지 여부 (Boolean),
              "result" : 탐지 종류 (String)
```

```
            }
        }
    }
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'ip': '205.134.249.11'}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/ip/report.php',
params =params, verify=False)
print response.text
```

호출 결과 예제


```
{
    "date": "2017-12-14 11:49:52",
    "ip": "205.134.249.11",
    "result_code": 1,
    "result_msg": "Data exists",
    "secudiumti": {
        "scan_date": "2017-10-27 19:04:03",
        "scans": {
            "domain": "mak-me.ae",
            "url": "http://mak-me.ae",
            "type": "phishing_domain",
```

```
                "geo": "US",
                "Malicious Site": {
                    "result": "Phishing Domain",
                    "detected": true
                }
            }
        }
}
```


3.3. HASH Text 체크 API

- **Text를 입력 받아 HASH 추출 후 기본 DB 및 Correlation DB를 체크하고 악성 여부 및 해당 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

HASH Text 체크 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/hash/report.php

Method : GET

Parameters :

api_key = 사용자 API Key

hash = 조회할 파일의 MD5

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

HASH Text 체크 응답 JSON

```
{
  "result_code" : 결과 코드 (Number),
  "result_msg" : 결과 메시지 (String),
  "date" : API 요청 시간 (String),
  "md5" : 조회 요청한 파일의 MD5 (String),
  "scan_date" : 파일 엔진 검사 시간 (String),
  "scans" : {
      "Malware Detection" : 악성코드 탐지정보 {
          "detected" : 탐지 여부 (Boolean),
          "result" : 탐지 결과 (String)
        }
    }
 }
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'hash': C3D859642967B38E29D24BBCAE78DAB8'}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/hash/report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제


```
{
    "date": "2017-12-14 12:58:05",
    "md5": "C3D859642967B38E29D24BBCAE78DAB8",
    "result_code": 1,
    "result_msg": "Data exists",
    "scan_date": "2017-10-10 12:45:59",
    "scans": {
        "Malware Detection": {
            "detected": true,
            "result": "Bkav : W32.FakeDocD.fam.Trojan|TotalDefense ... "
        }
    }
}
```

3.4. File Upload & 분석 API

- **업로드 된 파일을 입력 받아 HASH를 추출하고 DB를 체크하여**
  - **DB 내 동일한 파일이 존재하는 경우 관련 정보를 요청**
  - **DB 내 동일한 파일이 존재하지 않는 경우 1차 분석(MAS, MAV) + 2차 분석 (Correlation)을 수행하도록 절차 진행하여 해당 상태에 대한 value를 요청**

  할 때 사용하는 API로 HTTP POST 방식을 이용합니다.
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

File Upload & 분석 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/file/upload.php

Method : POST

Parameters :

api_key = 사용자 API Key

filename = 분석 요청할 파일명

files = 분석 요청할 파일

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

File Upload & 분석 응답 JSON

case 1.

```
{
    "result_code": 결과 코드 (Number),
    "result_msg": 결과 메시지 (String),
    "date": API 요청 시간 (String),
    "md5": "65e30dbc09f926dd1385b41a80ec6ba6",
    "file_from": 파일 업로더 (String),
    "file_added_time": 파일 업로드 타임(String)
    "status": 파일 분석 진행 상태(String)
}
```

case 2.

```
{
    "result_code" : 결과 코드 (Number),
    "result_msg" : 결과 메시지 (String),
    "date" : API 요청 시간 (String),
    "md5" : 분석 요청한 파일의 MD5 (String)
}
```

- **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
2	Upload Success Duplication	File upload요청에 대해 동일한 파일이 이미 등록되어 있음.
1	Upload Success	File 업로드 성공
0	Upload Failed	File 업로드 실패
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'filename': 'test.exe'}
files = {'file': ('test.exe', open(' test.exe 파일 절대경로', 'rb'))}
response = requests.post('https://api.skinfosec.co.kr/idb/v2/api/file/upload.php',
files=files, data=params, verify=False)
print response.text
```

호출 결과 예제

```
{
    date': '2017-12-14 13:59:58',
    md5': 'd577273ff885c3f84dadb8578bb41399',
    'result_code': 1,
    'result_msg': 'Upload Success'
}
```

3.5. URL 분석 API

■ **Text를 입력 받아 HASH 추출 후 DB를 체크하여**

- **DB 내 정보가 있는 경우 악성 여부 및 해당 정보를 요청**
- **DB 내 정보가 없는 경우 1차 분석(MAS, Web Crawling) + 2차 분석 (Correlation)을 수행해야 하며, 해당 정보로 DB 업데이트 후 처리 여부에 대한 Flag를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**

■ **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

URL 분석 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/url/analysis.php
Method : GET
Parameters :
api_key = 사용자 API Key
url = 분석 요청할 URL

■ **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
■ **JSON의 형태는 아래와 같습니다.**

URL 분석 응답 JSON

```
{
    "result_code" : 결과 코드 (Number),
    "result_msg" : 결과 메시지 (String),
    "date" : API 요청 시간 (String),
    "url" : 분석 요청한 URL (String),
    "url_hash" : 분석 요청한 URL Hash (String),
    "status" : URL 분석 진행 상태 - 미진행, 진행중, 분석완료, ERROR (String)
}
```

■ **결과 코드**

코드	메시지	설명
501	URL analysis is in progress.	URL 분석 진행 중. 분석상태는 status 필드로 제공 * status 값 : Ready(준비), Pending(미진행), Running(진행중), Completed(분석완료), Failed(분석에러)
500	Failed to save analysis request	URL 분석요청을 DB로 저장하지 못함
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Analysis request was successful,	URL 분석요청 성공. 이후 레포팅

	please request later for the report.	을 위해서는 Domain/URL Text 체크 API 호출 필요
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'url': http://www.atoffshore.net/ '}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/url/analysis.php',
params=params, verify=False)
print response.text
```

호출 결과 예제

```
{
      "date": "2017-12-14 14:09:21",
      "url": "http://www.atoffshore.net/ ",
      "url_hash": "6bd69fbf2d380f4e4bf1b0c5121267e5",
      "result_code": 1,
      "result_msg": "Analysis request was successful, please request later for the
report",
      "status": "Ready"
}
```

3.6. CVE 조회 API

■ **CVE Code를 입력 받아 해당 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**

■ **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

CVE 조회 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/cve/report.php

Method : GET

Parameters :

api_key = 사용자 API Key

cve_code = 조회할 CVE Code

■ **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**

■ **JSON의 형태는 아래와 같습니다.**

CVE 조회 응답 JSON

```
{
    "result_code" : 결과 코드 (Number),
    "result_msg" : 결과 메시지 (String),
    "date" : API 요청 시간 (String),
    "cve_code" : 조회한 CVE Code (String),
    "cve": {
        "description": CVE Code 정보 (String),
        "cvss_score": CVSS Score (String)
    },
    "pattern": {
        "event": SNORT event 정보 (String)
    },
    "exploit": {
        "description": Exploit 정보 (String)
    },
    "solution": {
        "proventia": [
            {
              "event_name": 이벤트명 (String),
              "description": 이벤트 정보 (String),
              "infosec_score": 인포섹 Real-Time Score (String)
              }
        ],
        "sniper_ips": [
            {
              "event_name": 이벤트명 (String),
              "description": 이벤트 정보 (String),
              "infosec_score": 인포섹 Real-Time Score (String)
              }
        ]
    }
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not	필수 Parameter(파라미터명) 값이

	exist	없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

■ **호출 코드 예제**

```
import requests
params = {'api_key': 'API_KEY', 'cve_code': 'CVE-2011-3402'}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/cve/report.php',
params=params, verify=False)
print response.text
```

■ **호출 결과 예제**


```
{
    "date": "2017-12-14 14:19:57",
    "cve_code": "CVE-2011-3402",
    "result_code": 1,
    "result_msg": "Data exists",
    "cve": {
        "description": " Unspecified vulnerability in the kernel-mode drivers ...",
        "cvss_score": "9.3"
    },
    "solution": {
        "proventia": [
            {
                "event_name": "HTML_Duqu_Font_Escalation",
                "description": "<P><B>Proventia Network IPS, IBM Security ... ",
                "infosec_score": "98"
            }, {
                "event_name": "JPEG_Duqu_CnC",
                "description": "<P><B>Proventia Network MFS, Proventia ... ",
                "infosec_score": "98"
            }, {
                "event_name": "CompoundFile_Duqu_Font",
                "description": "<P><B>Proventia Network MFS, Proventia ... ",
                "infosec_score": "98"
```

```
                }
            ]
        },
        "pattern": {
            "event": " alert tcp $EXTERNAL_NET $FILE_DATA_PORTS -> $HOME ... "
        }
}
```


3.7. 최근 자주 사용되는 IP 정보(TOP10) API

- **최근 자주 사용되는 IP 정보 TOP 10을 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

최근 자주 사용되는 IP 정보(TOP10) API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/top/report.php

Method : GET

Parameters :

api_key = 사용자 API Key

top_type = 데이터 타입(IP)

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

최근 자주 사용되는 IP 정보(TOP10) 응답 JSON


```
{
    "date": API 요청 시간 (String),
    "top_type":  "데이터 타입 (String)",
    "result_code": 결과 코드 (Number),
    "result_msg": 결과 메시지 (String),
    "secudiumti": {
        "top": {
            "1": top1 ip (String),
            "2": top2 ip (String),
            "3": top3 ip (String),
            "4": top4 ip (String),
            "5": top5 ip (String),
            "6": top6 ip (String),
            "7": top7 ip (String),
            "8": top8 ip (String),
            "9": top9 ip (String),
```

```
            "10": top10 ip (String),
        }
    }
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'top_type': 'IP'}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/top/report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제

```
{
    "date": "2017-12-14 14:53:37",
    "top_type": "IP",
    "result_code": 1,
    "result_msg": "Data exists",
    "secudiumti": {
        "top": {
            "1": "148.81.111.121",
            "2": "205.185.208.154",
            "3": "204.11.56.48",
            "4": "23.253.126.58",
```

```
                "5": "208.100.26.251",
                "6": "104.18.55.167",
                "7": "104.31.75.124",
                "8": "104.18.54.167",
                "9": "185.6.177.19",
                "10": "104.239.157.210"
            }
        }
}
```


3.8. 최근 자주 사용되는 HASH 정보(TOP10) API

- **최근 자주 사용되는 HASH 정보 TOP 10을 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

최근 자주 사용되는 HASH 정보(TOP10) API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/top/report.php

Method : GET

Parameters :

api_key = 사용자 API Key

top_type = 데이터 타입(HASH)

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

최근 자주 사용되는 HASH 정보(TOP10) 응답 JSON

```
{
    "date": API 요청 시간 (String),
    "top_type":  "데이터 타입 (String)",
    "result_code": 결과 코드 (Number),
    "result_msg": 결과 메시지 (String),
    "secudiumti": {
        "top": {
            "1": top1 hash (String),
            "2": top2 hash (String),
            "3": top3 hash (String),
            "4": top4 hash (String),
            "5": top5 hash (String),
            "6": top6 hash (String),
```

```
            "7": top7 hash (String),
            "8": top8 hash (String),
            "9": top9 hash (String),
            "10": top10 hash (String),
        }
    }
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'top_type': 'hash'}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/top/report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제

```
{
    "date": "2017-12-14 14:57:38",
    "top_type": "hash",
    "result_code": 1,
    "result_msg": "Data exists",
    "secudiumti": {
        "top": {
            "1": "fe06ab7369fca4bd289a3e4fb80c728d",
            "2": "c17103ae9072a06da581dec998343fc1",
```

```
                "3": "a71a2319cf8c74a89501eb80acd04fe6",
                "4": "0ee914c6f0bb93996c75941e1ad629c6",
                "5": "d82a429efd885ca0f324dd92afb6b7b8",
                "6": "b110b2290df96fba42275120676aea89",
                "7": "00a0194c20ee912257df53bfe258ee4a",
                "8": "0ccc0b4b0b610d5fd681390b055d06c8",
                "9": "3813cab188d1de6f92f8b82c2059991b",
                "10": "c5f897e9adb3848a97d1c0241ebd0bbd"
            }
        }
}
```


3.9. 최신 CVE 정보 (TOP10) API

- **최신 CVE 정보 TOP 10을 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

최신 CVE 정보(TOP10) API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/top/report.php

Method : GET

Parameters :

api_key = 사용자 API Key

top_type = 데이터 타입(CVE)

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

최신 CVE 정보(TOP10) 응답 JSON

```
{
    "date": API 요청 시간 (String),
    "top_type":  "데이터 타입 (String)",
    "result_code": 결과 코드 (Number),
    "result_msg": 결과 메시지 (String),
    "secudiumti": {
        "top": {
            "1": top1 cve code (String),
            "2": top2 cve code (String),
            "3": top3 cve code (String),
            "4": top4 cve code (String),
```

```
                "5": top5 cve code (String),
                "6": top6 cve code (String),
                "7": top7 cve code (String),
                "8": top8 cve code (String),
                "9": top9 cve code (String),
                "10": top10 cve code (String),
            }
        }
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'top_type': 'CVE'}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/top/report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제

```
{
    "date": "2017-12-14 16:17:34",
    "top_type": "CVE",
    "result_code": 1,
    "result_msg": "Data exists",
    "secudiumti": {
        "top": {
```

```
                "1": "CVE-2017-5638",
                "2": "CVE-2017-7494",
                "3": "CVE-2017-0199",
                "4": "CVE-2017-7269",
                "5": "CVE-2016-9244",
                "6": "CVE-2017-8759",
                "7": "CVE-2016-4657",
                "8": "CVE-2017-6074",
                "9": "CVE-2017-1000367",
                "10": "CVE-2017-0005"
            }
        }
}
```


3.10. 보안에 관련된 최신 SNS 정보 API

- **보안에 관련된 최신 SNS 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다. 최신 순으로 상위 5개만 응답합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

보안에 관련된 최신 SNS 정보 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/social/report.php
Method : GET
Parameters :
api_key = 사용자 API Key
type = 정보 형태(SNS)

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

보안에 관련된 최신 SNS 정보 응답 JSON

```
{
    "date": API 요청 시간 (String),
    "result_code": 결과 코드 (Number),
    "result_msg": 결과 메시지 (String),
    "type" : 정보 형태,
    "sns": {
        "category_title": [
            "SNS Title1 (String)",
```

```
            "SNS Title2 (String)",
            "SNS Title3 (String)",
            "SNS Title4 (String)",
            "SNS Title5 (String)"
        ],
        "category_link": [
            "SNS Link1 (String)",
            "SNS Link2 (String)",
            "SNS Link3 (String)",
            "SNS Link4 (String)",
            "SNS Link5 (String)"
        ]
    }
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'type': 'SNS'}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/social/report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제

```
{
    "date": "2017-12-14 16:41:02",
    "type": "SNS",
```

```
    "result_code": 1,
    "result_msg": "Data exists",
    "sns": {
        "category_title": [
            "RT @FamousDex: Thank you God for my Twitter back 💯 … ",
            "RT @charliekirk11: If the left is so worried about Russia … ",
            "【 Juuri's writer is so fineeeee ~ Cheer up. Don't say you're … ",
            "RT @_NikaEnot: *hacking noises* ₩n₩n#sidebyside #sombra … ",
            "RT @McAfee: #Petya was designed for maximum destruction … "
        ],
        "category_link": [
            "https://twitter.com/hacking/status/881876822035714058",
            "https://twitter.com/hacking/status/881876977464074240",
            "https://twitter.com/hacking/status/881876989157588992",
            "https://twitter.com/hacking/status/881877067079589888",
            "https://twitter.com/Petya/status/881876716741943296"
        ]
    }
}
```


3.11. 보안에 관련된 최신 RSS 정보 API

- **보안에 관련된 최신 RSS 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다. 최신 순으로 상위 5개만 응답합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

보안에 관련된 최신 RSS 정보 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/social/report.php
Method : GET
Parameters :
api_key = 사용자 API Key
type = 정보 형태(RSS)

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

보안에 관련된 최신 SNS 정보 응답 JSON


```
{
```

```
"date": API 요청 시간 (String),
"result_code": 결과 코드 (Number),
"result_msg": 결과 메시지 (String),
"type" : 정보 형태,
"sns": {
    "category_title": [
        "RSS Title1 (String)",
        "RSS Title2 (String)",
        "RSS Title3 (String)",
        "RSS Title4 (String)",
        "RSS Title5 (String)"
    ],
    "category_link": [
        "RSS Link1 (String)",
        "RSS Link2 (String)",
        "RSS Link3 (String)",
        "RSS Link4 (String)",
        "RSS Link5 (String)"
      ]
  }
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'type': 'RSS'}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/social/report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제


```
{
    "date": "2017-12-18 12:59:53",
    "type": "RSS",
    "result_code": 1,
    "result_msg": "Data exists",
    "sns": {
        "category_title": [
            "MS 윈도우 번들의 비밀번호 관리 프로그램서 취약점 발견",
            "'평양·주체' 해커, 가상화폐 지갑 노린 악성코드 유포",
            "구글, "뉴스 사이트 국적 밝혀야만 검색 노출"",
            "모질라, 해커 드라마 홍보하려다가 후폭풍 맞아",
            "MS 윈도우 번들의 비밀번호 관리 프로그램서 취약점 발견"
        ],
        "category_link": [

"http://www.boannews.com/media/view.asp?idx=58668&amp;kind=1&amp;sub_kind=",

"http://www.boannews.com/media/view.asp?idx=58669&amp;kind=1&amp;sub_kind=",

"http://www.boannews.com/media/view.asp?idx=58670&amp;kind=1&amp;sub_kind=",

"http://www.boannews.com/media/view.asp?idx=58671&amp;kind=1&amp;sub_kind=",

"http://www.boannews.com/media/view.asp?idx=58668&amp;kind=&amp;sub_kind="
        ]
    }
}
```


3.12. 최근 사용되는 악성 IP, File Hash 지리 정보 API

- **최근 사용되는 악성 IP, File Hash 지리 정보를 요청할 때 사용하는 API로 HTTP**

GET 방식을 이용합니다.

- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

최근 사용되는 악성 IP, File Hash 지리 정보 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/geo/report.php

Method : GET

Parameters :

api_key = 사용자 API Key

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

최근 사용되는 악성 IP, File Hash 지리 정보 응답 JSON

```
{
    "country_list": {
        국가코드(String) : [
            악성 IP (String),
                ...(중략)
            악성 IP (String)
        ],
                ...(중략)
        국가코드(String) : [
            악성 IP (String),
                ...(중략)
            악성 IP (String)
        ]
    },
    "colors_list": {
        국가코드(String) : color,
                ...(중략)
        국가코드(String) : color
    },
    "cnc_ip_cnt": cns ip count (Number),
    "file_hash_cnt": File HASH count (Number)
}
```

- **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음

10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY'}
response = requests.get('https://api.skinfosec.co.kr/idb/v2/api/geo/report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제


```
{
    "country_list": {
        "US": [
            "192.185.140.171",
            "98.124.199.39",
            "50.87.150.117",
            "209.99.64.53",
            "184.168.221.15",
            "69.64.147.39",
            "104.223.149.158",
            "98.124.251.168",
            "184.168.221.104",
            "207.148.248.145",
            "208.100.26.251",
            "206.188.192.70",
            "72.52.4.90",
            "151.101.192.133",
            "98.124.251.76",
            "192.241.210.182",
             "174.37.240.163"
        ],
        "SG": [
            "180.210.207.119",
```

```
            "107.6.122.136"
        ],
        "UA": [
            "178.20.153.95"
        ],
        "DE": [
            "5.189.187.177",
            "217.160.231.201",
            "80.237.133.235",
            "82.135.122.10"
        ],
        "IT": [
            "62.149.128.163"
        ],
        "NL": [
            "37.48.125.120"
        ],
        "VN": [
            "123.30.135.239"
        ],
        "CN": [
            "218.6.23.43",
            "118.193.175.153",
            "121.12.98.72"
        ],
        "RU": [
            "87.236.16.175",
            "195.14.105.12"
        ],
        "CA": [
            "47.74.135.203"
        ],
        "CH": [
            "5.226.149.110"
        ],
        "FR": [
            "94.247.178.77"
        ]
    },
```

```
    "colors_list": {
        "US": "#dd4b39",
        "SG": "#dd4b39",
        "UA": "#dd4b39",
        "DE": "#dd4b39",
        "IT": "#dd4b39",
        "NL": "#dd4b39",
        "VN": "#dd4b39",
        "CN": "#dd4b39",
        "RU": "#dd4b39",
        "CA": "#dd4b39",
        "CH": "#dd4b39",
        "FR": "#dd4b39"
    },
    "cnc_ip_cnt": 37,
    "file_hash_cnt": 0
}
```


3.13. 최근의 높은 스코어 가진 CVE 정보 API

- **최근의 높은 스코어 가진 CVE 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

최근의 높은 스코어를 가진 CVE 정보 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/cve/score_report.php
Method : GET
Parameters :
api_key = 사용자 API Key
top_type = 데이터 타입(CVE)

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

최근의 높은 스코어를 가진 CVE 정보 응답 JSON

```
{
  "date": API 요청 시간 (String),
  "top_type":  "데이터 타입 (String)",
```

```
"result_code": 결과 코드 (Number),
"result_msg": 결과 메시지 (String),
"secudiumti": {
      "event_cve": [
          cve code (String),
                ...(중략),
          cve code (String)
      ],
      "twiter_severity": [
           SNS severity (String),
                ...(중략),
          SNS severity (String)
      ],
      "exploit_severity": [
           exploit severity(String),
                ...(중략),
          exploit severity (String)
      ],
      "event_risk": [
          event risk(String),
                ...(중략),
        event risk(String)
          ]
    }
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'top_type': 'CVE'}
```

```
response =
requests.get('https://api.skinfosec.co.kr/idb/v2/api/cve/score_report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제


```
{
    "date": "2017-12-14 17:11:25",
    "top_type": "CVE",
    "result_code": 1,
    "result_msg": "Data exists",
    "secudiumti": {
        "event_cve": [
            "CVE-2017-0199",
            "CVE-2017-0199",
            "CVE-2017-0199",
            "CVE-2017-0199",
            "CVE-2017-11882",
            "CVE-2016-4275",
            "CVE-2017-11394",
            "CVE-2017-6622",
            "CVE-2017-3078",
            "CVE-2015-2342"
        ],
        "twiter_severity": [
            "Medium",
            "Medium",
            "Medium",
            "Medium",
            "Critic",
            "Low",
            "Low",
            "Low",
            "Low",
            "Low"
        ],
        "exploit_severity": [
            "High",
            "High",
```

```
                "High",
                "High",
                "High",
                "High",
                "High",
                "High",
                "High",
                "High"
            ],
            "event_risk": [
                "High",
                "High",
                "High",
                "High",
                "High",
                "High",
                "High",
                "High",
                "High",
                "High"
            ]
        }
}
```


3.14. 파일 기본 정보 API

- **파일 기본 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

파일 기본 정보 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/hash/summary_report.php

Method : GET

Parameters :

api_key = 사용자 API Key

hash = hash값

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

파일 기본 정보 응답 JSON

```
{
  "date": API 요청 시간 (String),
  "result_code": 결과 코드 (Number),
  "result_msg": 결과 메시지 (String),
  "secudiumti": {
       "scans": {
            "FileName": 파일이름 (String),
            "FileOrgName": 원본 파일 이름 (String),
            "FileType": 파일 형식 (String),
            "FileSize": 파일크기 (String),
            "Version": 파일 버전 (String),
            "Trid": File Description (String),
            "Sha1": 조회 요청한 파일의 Sha1 (String),
            "Sha256": 조회 요청한 파일의 Sha256 (String),
            "Md5" : 조회 요청한 파일의 MD5 (String),
            "TimeStamp": 파일 최초 생성일 (String)
          }
      }
 }
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'hash':
'c8a1ead8163198ad0f6ed786ecf1ff0ac3600b6d00c469e1febeeaea0836aefd'}
response =
requests.get('https://api.skinfosec.co.kr/idb/v2/api/hash/summary_report.php',
```

```
params=params, verify=False)
print response.text
```

호출 결과 예제


```
{
      "date": "2017-12-14 17:23:33",
      "result_code": 1,
      "result_msg": "Data exists",
      "secudiumti": {
            "scans": {
                  "FileName":
"c8a1ead8163198ad0f6ed786ecf1ff0ac3600b6d00c469e1febeeaea0836aefd",
                  "FileOrgName": "N/A",
                  "FileType": "text/html",
                  "FileSize": "N/A",
                  "Version": "N/A",
                  "Trid": "100.0% (.HTML) HyperText Markup Language (3000/1/1)",
                  "Sha1": "ea01486de984112606cfca1f8f24cb81784fbba2",
                  "Sha256":
"c8a1ead8163198ad0f6ed786ecf1ff0ac3600b6d00c469e1febeeaea0836aefd",
                  "Md5": "61b954f5040341e91a2d73a191b92d9e",
                  "TimeStamp": "N/A"
            }
      }
}
```


3.15. 파일 정적 분석 정보 API

- **파일 정적 분석 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

파일 정적 분석 정보 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/hash/static_report.php

Method : GET

Parameters :

api_key = 사용자 API Key

hash = 조회할 hash

■ **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**

■ **JSON의 형태는 아래와 같습니다.**

파일 정적 분석 정보 응답 JSON

```
{
    "date": API 요청 시간 (String),
    "result_code": 결과 코드 (Number),
    "result_msg": 결과 메시지 (String),
    "secudiumti": {
        "scan_date": 조회시간 (String),
        "pe_info": {
            "mime": PE 정보 (String),
            "type": "N/A",
             "Trid": File Description (String),
        },
        "codesign": {
            "has_sig": code sign 유/무,
            "sig_data": [
                {code sign 데이터(String)},
                        ...(중략)
                {code sign 데이터(String)}
            ],
            "insert_time": 입력시간 (String)
        }
    }
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'hash': 'bc9cf1f246e1bb9e4d8dfc7a8230efb3'}
response =
requests.get('https://api.skinfosec.co.kr/idb/v2/api/hash/static_report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제


```
{
    "date": "2017-12-14 17:30:28",
    "result_code": 1,
    "result_msg": "Data exists",
    "secudiumti": {
        "scan_date": "2012-09-18 23:54:58",
        "pe_info": {
            "mime": "application/x-dosexec",
            "type": "N/A",
            "trid": "63.7% (.EXE) Win32 EXE Yoda's Crypter (26569/9/4)₩n 15.7%
(.DLL) Win32 Dynamic Link Library (generic) (6578/25/2)₩n 10.8% (.EXE) Win32
Executable (generic) (4508/7/1)₩n  4.8% (.EXE) Generic Win/DOS Executable
(2002/3)₩n  4.8% (.EXE) DOS Executable Generic (2000/1)"
        },
        "codesign": {
            "has_sig": true,
            "sig_data": [
                {
                    "subejct": "/C=ES/ST=Barcelona/L=Barcelona/O=Softonic
International/OU=Digital ID Class 3 - Microsoft Software Validation
v2/CN=Softonic International",
                    "type": "Signer #0",
                    "issuer": "/C=US/O=VeriSign, Inc./OU=VeriSign Trust
Network/OU=Terms of use at https://www.verisign.com/rpa (c)10/CN=VeriSign
Class 3 Code Signing 2010 CA"
                },
                {
                    "subejct": "/C=ES/ST=Barcelona/L=Barcelona/O=Softonic
International/OU=Digital ID Class 3 - Microsoft Software Validation
v2/CN=Softonic International",
                    "type": "Cert #0",
```

```
                    "issuer": "/C=US/O=VeriSign, Inc./OU=VeriSign Trust
Network/OU=Terms of use at https://www.verisign.com/rpa (c)10/CN=VeriSign
Class 3 Code Signing 2010 CA"
                },
                {
                    "subejct": "/C=US/O=VeriSign, Inc./OU=VeriSign Trust
Network/OU=(c) 2006 VeriSign, Inc. - For authorized use only/CN=VeriSign Class
3 Public Primary Certification Authority - G5",
                    "type": "Cert #1",
                    "issuer": "/C=US/O=VeriSign, Inc./OU=VeriSign Trust
Network/OU=(c) 2006 VeriSign, Inc. - For authorized use only/CN=VeriSign Class
3 Public Primary Certification Authority - G5"
                },
                {
                    "subejct": "/C=US/O=VeriSign, Inc./OU=VeriSign Trust
Network/OU=Terms of use at https://www.verisign.com/rpa (c)10/CN=VeriSign
Class 3 Code Signing 2010 CA",
                    "type": "Cert #2",
                    "issuer": "/C=US/O=VeriSign, Inc./OU=VeriSign Trust
Network/OU=(c) 2006 VeriSign, Inc. - For authorized use only/CN=VeriSign Class
3 Public Primary Certification Authority - G5"
                }
            ],
            "insert_time": "2017-11-03T19:01:27.153000"
        }
    }
}
```


3.16. 파일 행위 분석 정보 API

- **Hash를 입력 받아 필요 시 각각에 대해 기본 DB 및 Correlation DB를 체크하고, 해당 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

파일 행위 분석 정보 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/hash/behavior_report.php

Method : GET

Parameters :

api_key = 사용자 API Key

hash = 조회할 hash

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

파일 행위 분석 정보 응답 JSON

```
{
   "date": API 요청 시간 (String),
   "hash": 조회 요청한 hash (String),
   "result_code": 결과 코드 (Number),
   "result_msg": 결과 메시지 (String),
   "behavior":[
            {
            "rule_name" : 탐지룰 이름 (String),
            "rule_note" : 탐지룰 설명 (String),
            "ruid_id" : 탐지룰 id값 (String),
            "*domain" : 접속 Domain (String),
            "*header" : HTTP header (String),
            "*url" : HTTP URL 정보 (String),
            "*event_class" : FFile, Registry, Process (String),
            "*operation" : 특정 이벤트의 Operation (e.g. Read, RegQueryValue, etc.) (String),
            "*pid" : 룰에 탐지된 Process ID (String),
            "*tid" : 룰에 탐지된 Thread ID (String),
            "*ppid" : 룰에 탐지된 부모 Process ID (String),
            "*mal_pid" : 최초 실행된 Process ID (String),
            "*cmdline" : command line (String),
            "*path"   : 참조하는 리소스의 경로 (String),
            "*process_name" : process 이름 (String),
            "*protocol" : 통신 프로토콜 (String),
            "*ip" : IP Address (String),
            "*port" : 통신 port (String),
            "*country" : IP의 Geo정보 (String),
            "*black_ip" : BlackIP 유무 (String),
            "*autorun_name" : autorun 등록 이름 (String),
            "*autorun_path" : autorun 등록 registry 경로 (String),
            "*detail_path" : autorun 등록 파일 (String),
            "*autorun_type" : autorun 등록 및 수정 (String),
            "*detail_hash" : autorun 등록 파일 hash (String),
            "*detail_time" : autorun 등록 시간 (String),
            "*version" : 필요한 OS 버전 정보 (String),
            "*entrypoint" : Entry Point (String),
            "*timestamp" : debug 데이터가 작성된 시간 (String),
            "*imagebase" : imagebase (String),
            "*imphash" : imphash (String),
            "*macro" : [오피스 파일 내 매크로 정보 (String), ...]
            }, ...
```

```
    ],
    "summary":[
            {이벤트명 (String) : 이벤트 설명 (String)},
            ... 중략
            {이벤트명 (String) : 이벤트 설명 (String)}
    ],
    "file_info":{
            "filehash": 파일 해시 (String),
            "filename": 파일명 (String),
            "trid": 파일 trID (String),
            "type": 파일 magic (String),
            "mav_result" : mav 결과 (String),
            "mav_result_rate":  mav 탐지율 (String),
            "*version" : 필요한 OS 버전 정보 (String),
            "*entrypoint" : Entry Point (String),
            "*timestamp" : debug 데이터가 작성된 시간 (String),
            "*imagebase" : imagebase (String)2
            "*imphash" : imphash (String),
    },
    "file_dw_url" : 해당 파일 다운로드 URI (String),
    "severity" : 악성코드 심각도 (Number),
    "memory" : [
            /* memory 분석 결과값이 없어 확인 불가*/
            "*memory_ssdt":[],
            "*memory_psscan":[],
            "*memory_mutexes":[],
            "*memory_driverscan":[],
            "*memory_ldrmodules":[],
            "*memory_symlinkscan":[],
    ],
    "yara_info":[
            {
                    "category":         yara rule category (String),
                    "tag": [ 룰 태그이름 (String), ... ],
                    "rule" : yara rule 이름 (String),
                    "match_string":[
                            {
                                    "idx" : offset (String),
                                    "key" : rule의 변수명 (String),
                                    "value": 룰에 매치된 데이터 (String)
                            }
                    ]

            }
    ],
    "pdb_info": pdb 정보 (String)
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'hash':
'3cbcd339ef449532c59e5dfa3bb32baee44d25e15d53e6a96745aa2edf584822'}
response =
requests.get('https://api.skinfosec.co.kr/idb/v2/api/hash/behavior_report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제


```
{
    "date": "2017-12-18 13:35:28",
    "hash":
"3cbcd339ef449532c59e5dfa3bb32baee44d25e15d53e6a96745aa2edf584822",
    "result_code": 1,
    "result_msg": "Data exists",
    "vm_info": {
        "os": "Windows 7",
        "exec_time": "2017-12-15 23:09:28"
    },
    "chain_of_event": {
        "nodes": [
            {
                "id": "p_2840",
```

```
                    "pid": "2840"
                },
                중략...
            ],
            "links": [
                {
                    "id": "p_2840-t_2844",
                    "from": "p_2840",
                    "to": "t_2844"
                },
                중략...
            ]
        },
        "behavior": [
            {
                "domain": "skype.sd.softonic.fr",
                "rule_name": "HTTP_Connection",
                "rule_note": "HTTP 프로토콜을 통한 통신 시도",
                "header": "GET
http://skype.sd.softonic.fr/29287/universaldownloader-prefetch HTTP/1.1₩nKeep-
Alive₩nMozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0;
SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media
Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)
SoftonicDownloader/1.41.10₩n*/*₩ngzip, deflate₩nko₩n₩n[Hypertext Transfer
Protocol, key 0],Source GeoIP: Unknown,Destination GeoIP: Unknown,GET
/29287/universaldownloader-prefetch HTTP/1.1₩₩r₩₩n,₩₩r₩₩n,HTTP request
1/2",
                "url": "http://skype.sd.softonic.fr/29287/universaldownloader-
prefetch",
                "rule_id": "3"
            },
            중략...
        ],
        "drop_info": [
            {
                "mav_result": "",
                "filehash": "e1b09f60d9294113803d3b0cbc3d136e",
                "trid": "100.0% (.DAT) Microsoft Internet Explorer cache (24000/1)",
                "filename": "index.dat",
```

```
                "mav_result_rate": "N/A",
                "type": "Internet Explorer cache file version Ver 5.2"
            },
            중략...
        ],
        "net_info": {
            "domain": [
                "46.28.209.20",
                중략...
            ],
            "http": [
                "http://skype.sd.softonic.fr/29287/universaldownloader-prefetch",
                중략...
            ],
            "ip": [
                "46.28.209.20",
                중략...
            ]
        },
        "threat_event": [
            "HTTP_Connection",
            중략...
        ]
}
```


3.17. 위협 분석 정보 API

- **Hash를 입력 받아 필요 시 각각에 대해 기본 DB 및 Correlation DB를 체크하고, 해당 정보를 요청 할 때 사용하는 API로 HTTP GET 방식을 이용합니다.**
- **API 요청은 아래와 같은 방식을 통해 이루어집니다.**

위협 분석 정보 API 요청 방식

URL : https://api.skinfosec.co.kr/idb/v2/api/hash/threat_report.php

Method : GET

Parameters :

api_key = 사용자 API Key

hash = 조회할 hash

- **API 요청이 제대로 전달된 경우에는 JSON 형태의 응답을 받을 수 있습니다.**
- **JSON의 형태는 아래와 같습니다.**

위협 분석 정보 응답 JSON

```
{
   "date": API 요청 시간 (String),
   "hash": 조회 요청한 hash (String),
   "result_code": 결과 코드 (Number),
   "result_msg": 결과 메시지 (String),
   "behavior":[
            {
            "rule_name" : 탐지룰 이름 (String),
            "rule_note" : 탐지룰 설명 (String),
            "ruid_id" : 탐지룰 id값 (String),
            "*domain" : 접속 Domain (String),
            "*header" : HTTP header (String),
            "*url" : HTTP URL 정보 (String),
            "*event_class" : FFile, Registry, Process (String),
            "*operation" : 특정 이벤트의 Operation (e.g. Read, RegQueryValue, etc.) (String),
            "*pid" : 룰에 탐지된 Process ID (String),
            "*tid" : 룰에 탐지된 Thread ID (String),
            "*ppid" : 룰에 탐지된 부모 Process ID (String),
            "*mal_pid" : 최초 실행된 Process ID (String),
            "*cmdline" : command line (String),
            "*path"  : 참조하는 리소스의 경로 (String),
            "*process_name" : process 이름 (String),
            "*protocol" : 통신 프로토콜 (String),
            "*ip" : IP Address (String),
            "*port" : 통신 port (String),
            "*country" : IP의 Geo정보 (String),
            "*black_ip" : BlackIP 유무 (String),
            "*autorun_name" : autorun 등록 이름 (String),
            "*autorun_path" : autorun 등록 registry 경로 (String),
            "*detail_path" : autorun 등록 파일 (String),
            "*autorun_type" : autorun 등록 및 수정 (String),
            "*detail_hash" : autorun 등록 파일 hash (String),
            "*detail_time" : autorun 등록 시간 (String),
            "*version" : 필요한 OS 버전 정보 (String),
            "*entrypoint" : Entry Point (String),
            "*timestamp" : debug 데이터가 작성된 시간 (String),
            "*imagebase" : imagebase (String),
            "*imphash" : imphash (String),
            "*macro" : [오피스 파일 내 매크로 정보 (String), ...]
            }, ...
   ],
   "summary":[
            {이벤트명 (String) : 이벤트 설명 (String)},
```

```
            ... 중략
            {이벤트명 (String) : 이벤트 설명 (String)}
    ],
    "file_info":{
            "filehash": 파일 해시 (String),
            "filename": 파일명 (String),
            "trid": 파일 trID (String),
            "type": 파일 magic (String),
            "mav_result" : mav 결과 (String),
            "mav_result_rate":  mav 탐지율 (String),
            "*version" : 필요한 OS 버전 정보 (String),
            "*entrypoint" : Entry Point (String),
            "*timestamp" : debug 데이터가 작성된 시간 (String),
            "*imagebase" : imagebase (String)2
            "*imphash" : imphash (String),
    },
    "file_dw_url" : 해당 파일 다운로드 URI (String),
    "severity" : 악성코드 심각도 (Number),
    "memory" : [
            /* memory 분석 결과값이 없어 확인 불가*/
            "*memory_ssdt":[],
            "*memory_psscan":[],
            "*memory_mutexes":[],
            "*memory_driverscan":[],
            "*memory_ldrmodules":[],
            "*memory_symlinkscan":[],
    ],
    "yara_info":[
            {
                    "category":          yara rule category (String),
                    "tag": [ 룰 태그이름 (String), ... ],
                    "rule" : yara rule 이름 (String),
                    "match_string":[
                            {
                                    "idx" : offset (String),
                                    "key" : rule의 변수명 (String),
                                    "value": 룰에 매치된 데이터 (String)
                            }
                    ]

            }
    ],
    "pdb_info": pdb 정보 (String)
}
```

■ **결과 코드**

코드	메시지	설명
30	Internal Server Error	API 서버 오류
21	Allow HTTPS connection only	HTTPS 접속만 허용
20	Invalid URL	URL 형식이 잘못 되었음
10	Parameter(파라미터명) does not exist	필수 Parameter(파라미터명) 값이 없음
1	Data exists	정보 검색 성공
0	Data is not exist	정보가 DB에 존재하지 않음
-11	No matching data to API Key	API Key 오류

■ **사용 예제**

호출 코드 예제

```
import requests
params = {'api_key': 'API_KEY', 'hash':
'3cbcd339ef449532c59e5dfa3bb32baee44d25e15d53e6a96745aa2edf584822'}
response =
requests.get('https://api.skinfosec.co.kr/idb/v2/api/hash/threat_report.php',
params=params, verify=False)
print response.text
```

호출 결과 예제

```
{
    "date": "2017-12-18 14:25:41",
    "hash":
"3cbcd339ef449532c59e5dfa3bb32baee44d25e15d53e6a96745aa2edf584822",
    "result_code": 1,
    "result_msg": "Data exists",
    "summary": [
        {
            "Malicious Connection": "개요₩n악성 URL로의 접근₩n₩n행위분석₩n
악의적인 URL 접근을 통해 파일을 다운로드₩n악의적인 URL 접근을 통해 주요 정
보 유출₩n₩n주요 탐지 내용₩nbinary Download₩nOutBound Traffic₩n₩n권고사항
"
        },
        중략...
    ],
    "file_info": {
```

```
            "mav_result": "avg :   Found Win32/DH{WySBDoEPgmU?}₩nclamav :
Win.Trojan.Softonic-6391316-0₩ncomodo :
Application.Win32.SoftonicDownloader.A₩n",
            "filehash": "9a2fea0bfaa8f9463f48714bf8ff1c20",
            "imagebase": "0x400000",
            "filename":
"3cbcd339ef449532c59e5dfa3bb32baee44d25e15d53e6a96745aa2edf584822",
            "mav_result_rate": "3/50",
            "version": "5.1",
            "entrypoint": "0xfd810",
            "timestamp": "2014-12-17 00:54:46",
            "trid": "63.7% (.EXE) Win32 EXE Yoda's Crypter (26569/9/4)₩n 15.7% (.DLL)
Win32 Dynamic Link Library (generic) (6578/25/2)₩n 10.8% (.EXE) Win32
Executable (generic) (4508/7/1)₩n   4.8% (.EXE) Generic Win/DOS Executable
(2002/3)₩n   4.8% (.EXE) DOS Executable Generic (2000/1)",
            "type": "PE32 executable (GUI) Intel 80386, for MS Windows",
            "imphash": "7345b83325c8c576b3202c7067e70934"
      },
      "file_dw_url": "http://secudiumti.skinfosec.co.kr/download/2017-12-
07/CTA/1512607745_71_9a2fea0bfaa8f9463f48714bf8ff1c20/3cbcd339ef449532c59e
5dfa3bb32baee44d25e15d53e6a96745aa2edf584822",
      "behavior": [
            {
                  "domain": "skype.sd.softonic.fr",
                  "rule_name": "HTTP_Connection",
                  "rule_note": "HTTP 프로토콜을 통한 통신 시도",
                  "header": "GET http://skype.sd.softonic.fr/29287/universaldownloader-
prefetch HTTP/1.1₩nKeep-Alive₩nMozilla/4.0 (compatible; MSIE 7.0; Windows NT
6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR
3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)
SoftonicDownloader/1.41.10₩n*/*₩ngzip, deflate₩nko₩n₩n[Hypertext Transfer
Protocol, key 0],Source GeoIP: Unknown,Destination GeoIP: Unknown,GET
/29287/universaldownloader-prefetch HTTP/1.1₩₩r₩₩n,₩₩r₩₩n,HTTP request
1/2",
                  "url": "http://skype.sd.softonic.fr/29287/universaldownloader-prefetch",
                  "rule_id": "3"
            },
            중략...
      ],
```

```
    "severity": 100,
    "memory": [],
    "yara_info": [
        {
            "category": "packer_compiler_signatures",
            "tags": [
                "PECheck"
            ],
            "rule": "IsPacked",
            "match_strings": []
        },
        중략...
    ],
    "pdb_info": ""
}
```

End of Document