

Overview

Vulnerabilities in software and hardware components on enterprise assets like laptops, desktops and routers are one of the most common modes exploited in breaches. Often, they are the launch pad of an attack into an enterprise network.

The Balbix platform provides our customers with the most comprehensive analysis of their networks and inventory. We are continuously honing our knowledge base by adding information about vulnerabilities that could exist on assets. We compile reliable, accurate and timely information about such vulnerabilities and exposures from a diverse set of sources.

This internship program gives interns the opportunity to work on parsers that collate and feed vulnerability information for specific products identified by Balbix, and contribute to one of the core components of the platform.

The Common Vulnerabilities and Exposures (CVE) framework provides a way to identify and standardize various attributes of vulnerabilities affecting systems, including the type of vulnerability, the conditions required to be exposed to it, the affected products and available remediations. The affected products are specified using a format called Common Platform Enumeration (CPE) which is an industry standard way of identifying product attributes. It is critical to standardize and sanitize parsed data to maintain its usability in our product. To this end, the parsers are required to produce output in a prescribed format that will be validated against the requirements.

Assignment

Given below are a couple of vendor websites with vulnerability details. These sites contain important information about vulnerabilities with details of affected software products and versions.

Write a program to parse the given web sites and extract all key vulnerability metrics. Specifically, information should be parsed and formatted into JSON content as defined by the structure below.

JSON structure

The output JSON record should have these key metrics about the vulnerabilities published.

- Publish date of advisory (`published_date`)
- Current timestamp (`timestamp`)
- Product family name or suite name (`name`)
- Description of advisory or patch details (`description`)
- Web URL (`url`)
- CVE ID (unique identifier for the vulnerability (`id`))
- List of products and versions affected (CPE), specific versions or range, see examples) (`cpes`)
 - CPE list (`cpe_list`)
 - Vendor name (`vendor`)
 - Product name (`product`)
 - Product category (`category`) (always 'a' - application)
 - Product version from (`versionStartIncluding`) - needed only for version range
 - Product version to (`versionEndIncluding`)

Sample 1:

Website: <https://helpx.adobe.com/security/products/magento/apsb20-02.html>

Note: Output shows only CVE-2020-3715. Your actual output should include all CVEs in this page.

Output:

```
{
  "source": "adobe",
  "type": "vendor",
  "cves": [
    {
      "timestamp": "2020-03-12T18:06Z",
```

```

"published_date":"2020-01-28T00:00Z",
"id":"CVE-2020-3715",
"url":"https://helpx.adobe.com/security/products/magento/apsb20-02.html",
"name": "magento",
"description": "stored cross-site scripting",
"cpes":{
  "cpe_list":[
    {
      "vendor":"magento",
      "product":"magento",
      "category": "a",
      "versionEndIncluding":"2.3.3"
    },
    {
      "vendor":"magento",
      "product":"magento",
      "category": "a",
      "versionEndIncluding":"2.3.3"
    },
    {
      "vendor":"magento",
      "product":"magento",
      "category": "a",
      "versionEndIncluding":"2.2.10"
    },
    {
      "vendor":"magento",
      "product":"magento",
      "category": "a",
      "versionEndIncluding":"2.2.10"
    },
    {
      "vendor":"magento",
      "product":"magento",
      "category": "a",
      "versionEndIncluding":"1.14.4.3"
    },
    {
      "vendor":"magento",
      "product":"magento",
      "category": "a",
      "versionEndIncluding":"1.9.4.3"
    }
  ]
}
]
}
}

```

Sample 2:

Website: <https://helpx.adobe.com/security/products/experience-manager/apsb20-01.html>

Note: sample output shows CVE-2019-16466 and CVE-2019-16468. Your actual output should include all CVEs in this page.

Output:

```

{
  "source": "adobe",
  "type": "vendor",
  "cves":[
    {
      "timestamp":"2020-03-12T18:06Z",
      "published_date":"2020-01-14T00:00Z",
      "id":"CVE-2019-16466",
      "url":"https://helpx.adobe.com/security/products/experience-manager/apsb20-01.html",

```

```

"name": "Experience Manager",
"description": "cross-site script inclusion",
"cpes":{
  "cpe_list": [
    {
      "vendor": "adobe",
      "product": "experience_manager",
      "category": "a",
      "versionStartIncluding": "6.1",
      "versionEndIncluding": "6.5"
    }
  ]
},
{
  "timestamp": "2020-03-12T18:06Z",
  "published_date": "2020-01-14T00:00Z",
  "id": "CVE-2019-16468",
  "url": "https://helpx.adobe.com/security/products/experience-manager/apsb20-01.html",
  "name": "APSB20-01 Security update available for Adobe Experience Manager",
  "description": "user interface injection",
  "cpes":{
    "cpe_list": [
      {
        "vendor": "adobe",
        "product": "experience_manager",
        "category": "a",
        "versionStartIncluding": "6.3",
        "versionEndIncluding": "6.5"
      }
    ]
  }
}
]
}

```

Instructions and requirements

- Your code should take a website URL as input and output a file with JSON output as given in the samples
- Code should be written in Python3, please make sure to follow python coding standards
 - <https://docs.python-guide.org/writing/style/>
- Code, scripts and documentation (if any) should be packaged as .zip or .tar archive
- Please list all dependent libraries in a file (requirements.txt). the code should preferably be structured to run in a python3 virtualenv
- Add any instructions in a README that will help run the code on any machine
- Log enough information to help troubleshoot errors
- Make sure to test your code well before submitting
- Your code will be subjected to a quality check that includes manual review and testing.