**Nwachukwu Edumanichukwu**

**CYBR 445 - Advanced Incident Detection and Response  
Module 3 Lab – Cybersecurity APIs**

In this second lab, we will explore the application programming interfaces that are used to tie security tools together. The two types of APIs we will utilize will be REST APIs that return data to us in JSON format, and threat intelligence APIs that use XML and JSON to return threat intelligence data. Both of these APIs will be used in a future lab to build workflows that take specific actions and integrate the tools and automate research and response.

**You will be required to submit the following graded items as part of this lab:**

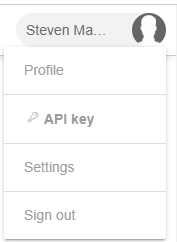
* Answer all questions listed in **BOLD**
* Provide screenshots when asked

Accessing the Lab

This lab is hosted in the university's IS Lab and requires special instructions to access it. If you are not familiar with accessing the IS Lab, please see the document in this course that walks you through accessing the Cybersecurity Desktop. You can access the Cybersecurity Desktop through the Web or using VMWare’s Horizon client. You should use the native Horizon client when possible as it provides better performance. The web client can be accessed at <http://workspace.bellevue.edu>. Make sure you log into this interface with your Bellevue student ID and password.

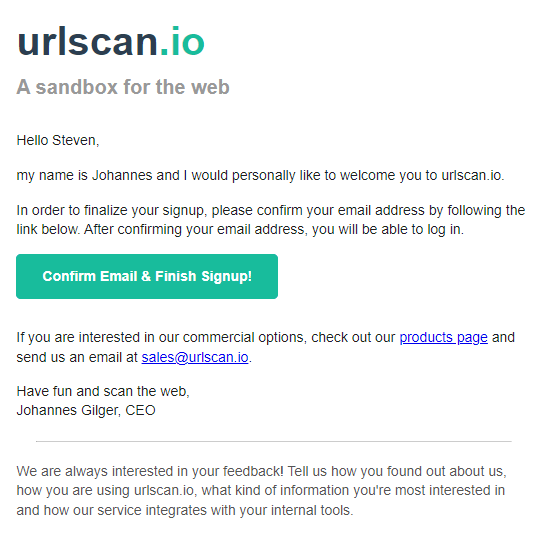
Part 1 - Virus Total, URLscan, and Open Threat Intelligence

1. In this first part, we will use an online REST & SOAP API testing tool to create API request and review API response. First, we need to create API keys for each service. Start by navigating to <https://www.virustotal.com/gui/join-us> if you don’t already have an account. Sign up for an account. You can use your school email address or a throw away email account.
2. When you have completed your registration, sign into Virus Total and click on your username. Select the API Key options to view your API key. Copy this down below for later use.

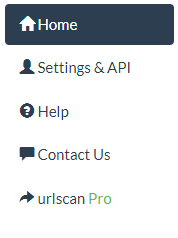


**Virus Total API Key:** c969f4a1c0f88b231b66df66b4b0ea8392b3c6662677d69c091a30b9d729d2a0

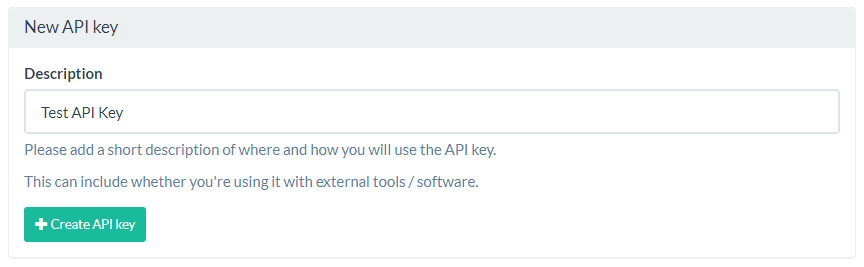
1. Navigate to <https://urlscan.io/user/signup>. Sign up for a URLScan account by putting in your information and an email address (personal, school, or throwaway). Click on the email you receive after signing up to finish the process.



1. Sign into urlscan.io and then click Settings & API on the left of the screen.



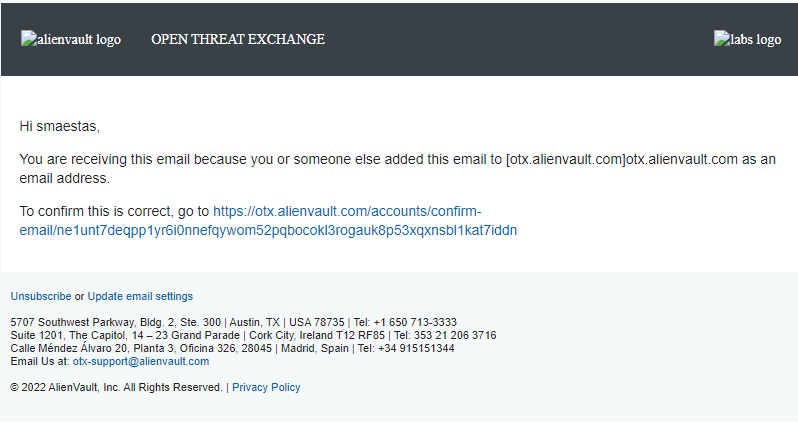
1. Click the Create API new API Key button. Give the API key the name of Test API Key on the next screen and click Create API key.



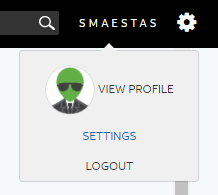
1. An obscured API key will appear at the bottom of the next page. Hover over the key to see the blurred portion. Copy the API key below.

**URLScan API Key:**  1bed63de-4405-4534-8f7e-be22d8846a47

1. Navigate to <https://otx.alienvault.com.> Sign up for an account if you don’t already have one. Note: You may have an existing account that you signed up for in CYBR 440. Use your personal, school, or a throwaway email account. As before sign up, click the email for verification and then sign into Alien Vault.



1. Once signed into Alien Vault OTX click on your username and then click on settings. Copy your OTX API key below.



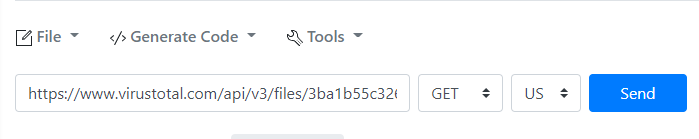
**Alien Vault OTX API Key:** 8927c6784bced246627e0f7091fce0a41fe534c0067dfeacb6782b8f41fedc08

API Documents for Virus Total can be found here: <https://developers.virustotal.com/reference/overview>

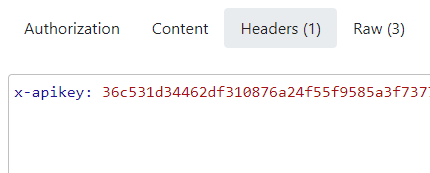
API Document for URLScan can be found here: <https://urlscan.io/docs/api/>

API Documentation for Alien Vault OTX can be found here: <https://otx.alienvault.com/api>

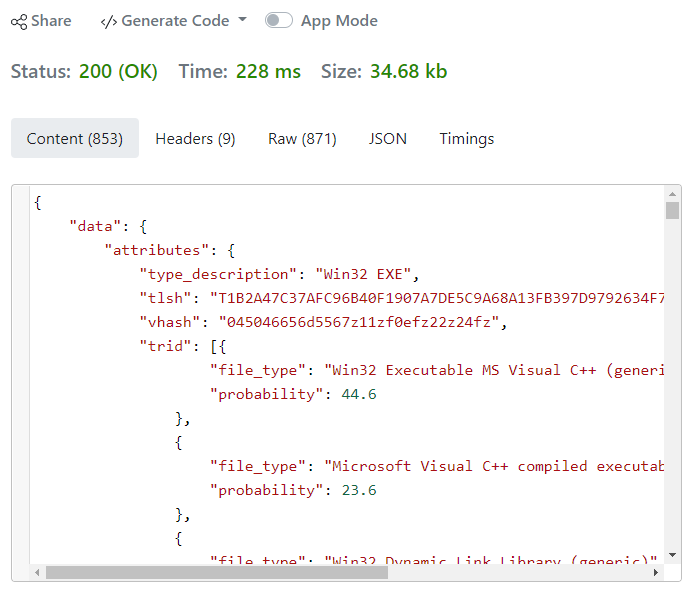
1. We will be using the only testing tool for APIs called ReqBin. Navigate to <https://reqbin.com>.
2. We will be searching for the file hash: 3ba1b55c3268529b586e154b9117d25ae6c3667a2e869747c51bd88fd2a7a581 using the API.
3. In the URL text box, type the following API URL: [https://www.virustotal.com/api/v3/files/{hash](https://www.virustotal.com/api/v3/files/%7bhash)} where hash is the hash of the file we are searching.



1. Select the headers tab under the URL textbox. In the headers text box, type: x-apikey: <your api key>, where <your api key> is your Virus Total API key.



1. Make sure the method is set to GET and language to US. Click Send. You should see a Status 200 on the right side of the page and the tab Content should have a number next to it and have JSON within the text box.



1. Change the tab to JSON and navigate to object ->data -> attributes -> last\_analysis\_results. Inspect the results for the different virus engines. Answer the following questions.

**What is the result for the FireEye antivirus engine?  
Generic.mg.0ebfd6e45dea48c7**

**What is the result for the tehtris antivirus engine?**

**Method: blacklist**

**Engine\_update: 20220526**

**What does Sophos identify the virus/file as?**

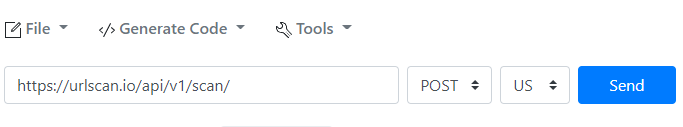
**Category: Malicious.**

**Result: ML/PE-A + Mal/Trickbt-A**

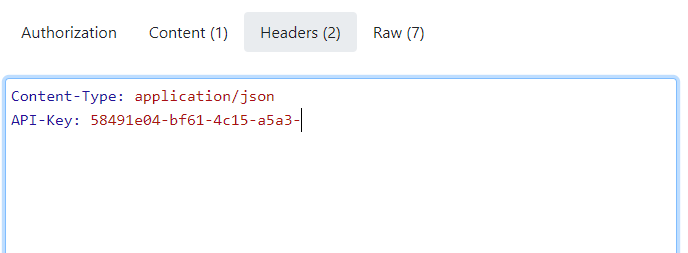
1. Pick a website of your choice to scan using URL scan. You can play with the GUI interface to get an idea of how URLScan works and what information is available.

**Website you’ve chosen to scan:** [**https://smaestas.com**](https://smaestas.com)

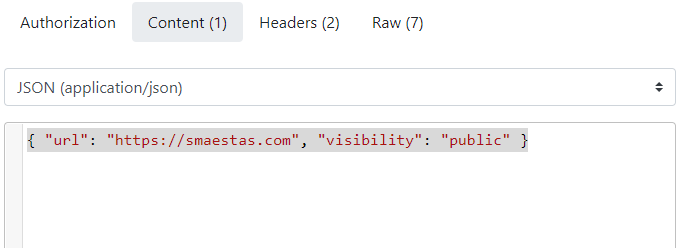
1. Change the URL to: <https://urlscan.io/api/v1/scan/> and the method to POST



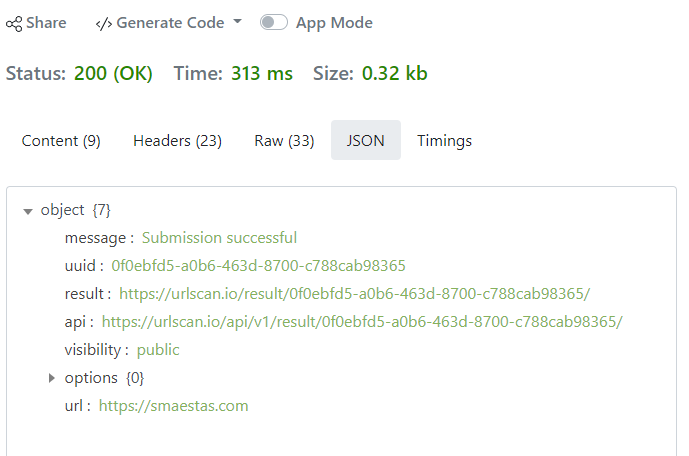
1. Replace the x-apikey: header with the following two headers:
   1. Content-Type: application/json
   2. API-Key: <your api key>
      1. Replace <your api key> with your URLScan API key



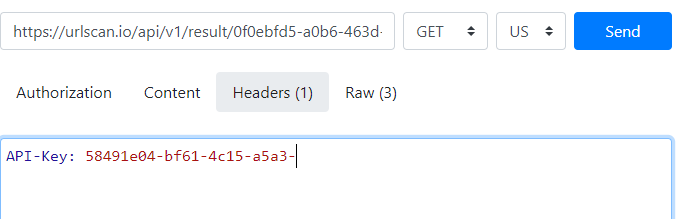
1. Enter { "url": "<your website>", "visibility": "public" } where <your website is the website you’ve chosen to scan.



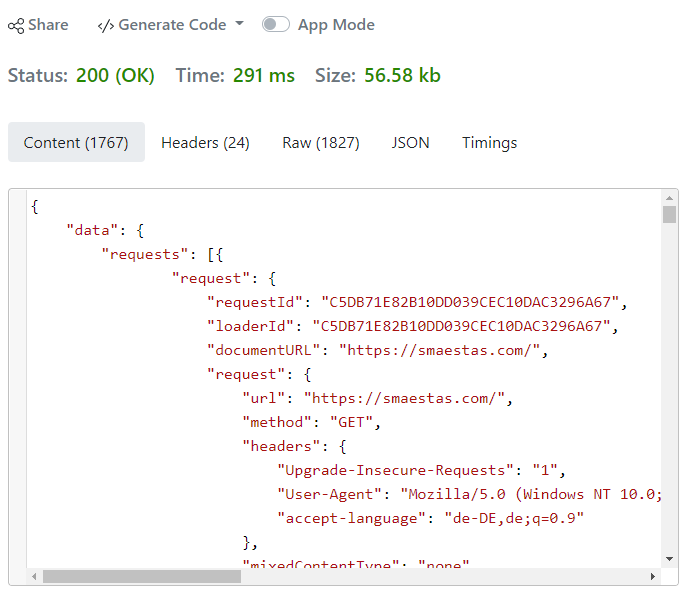
1. Click Send and wait for the 200 (OK) message. If there are any errors, correct them and try again until you get the correct results. Examine the JSON tab on the right for results.



1. Now we have only scheduled the scan of the website using the API. We will have to also get the results. Do the following.
   1. Copy the URL in the API: JSON field to the URL to query
   2. Delete the Content-Type: application/json header under the request headers on the left
   3. Delete all content under the Content type of the left
   4. Change the method from POST to GET
   5. Press Send



You should see the Status: change to 200 and results from the scan show up in JSON format in the Content tab on the right side of the page as shown below:



1. Copy and paste the first 25-30 lines of content below:

**Paste the first 25-30 lines of the returned JSON below:  
{**

**"data": {**

**"requests": [{**

**"request": {**

**"requestId": "ECB6BFE0D88DF9F32756929CB4424D71",**

**"loaderId": "ECB6BFE0D88DF9F32756929CB4424D71",**

**"documentURL": "https://smaestas.com/",**

**"request": {**

**"url": "https://smaestas.com/",**

**"method": "GET",**

**"headers": {**

**"Upgrade-Insecure-Requests": "1",**

**"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.5672.126 Safari/537.36",**

**"accept-language": "de-DE,de;q=0.9"**

**},**

**"mixedContentType": "none",**

**"initialPriority": "VeryHigh",**

**"referrerPolicy": "strict-origin-when-cross-origin",**

**"isSameSite": true**

**},**

**"timestamp": 21804723.766645,**

**"wallTime": 1685467126.957641,**

**"initiator": {**

**"type": "other"**

**},**

**"redirectHasExtraInfo": false,**

**"type": "Document",**

**"frameId": "1B426A830BEE6EB612E1D0F3F48F512C",**

**"hasUserGesture": false,**

**"primaryRequest": true**

**},**

**"response": {**

**"encodedDataLength": 664,**

**"dataLength": 1371,**

**"requestId": "ECB6BFE0D88DF9F32756929CB4424D71",**

**"type": "Document",**

**"response": {**

**"url": "https://smaestas.com/",**

**"status": 200,**

**"statusText": "",**

**"headers": {**

**"content-encoding": "br",**

**"content-type": "text/html",**

**"date": "Tue, 30 May 2023 17:18:47 GMT",**

**"etag": "W/\"6474602b-55b\"",**

**"last-modified": "Mon, 29 May 2023 08:19:55 GMT",**

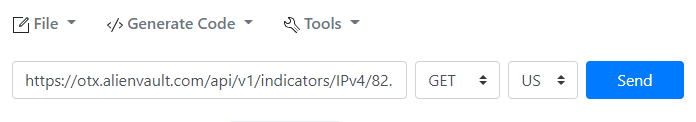
**"server": "nginx",**

**"vary": "Accept-Encoding",**

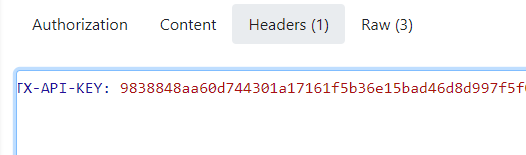
**"x-default-vhost": "1"**

**},**

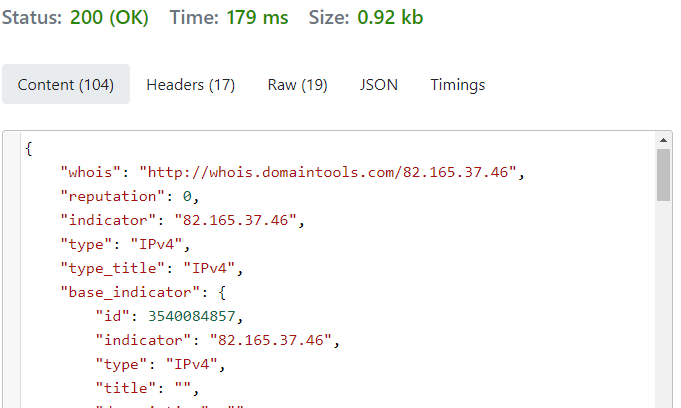
1. Now we will search Alien Vault OTX for a malicious IP address: [82.165.37.46.](https://otx.alienvault.com/indicator/ip/82.165.37.46) Enter the following URL into the URL textbox: <https://otx.alienvault.com/api/v1/indicators/IPv4/82.165.37.46/general>. Make sure the method is set to GET.



1. In the Headers tab on the left, change the text to the following:
   1. X-OTX-API-KEY: <your otx api key>
      1. Replace <your otx api key> with your OTX API key



1. Click Send. Copy the first 25-30 lines of the returned JSON below and answer the following questions.



**Paste the first 25-30 lines of return JSON here:**

**{**

**"whois": "http://whois.domaintools.com/82.165.37.46",**

**"reputation": 0,**

**"indicator": "82.165.37.46",**

**"type": "IPv4",**

**"type\_title": "IPv4",**

**"base\_indicator": {**

**"id": 3540084857,**

**"indicator": "82.165.37.46",**

**"type": "IPv4",**

**"title": "",**

**"description": "",**

**"content": "",**

**"access\_type": "public",**

**"access\_reason": ""**

**},**

**"pulse\_info": {**

**"count": 3,**

**"pulses": [{**

**"id": "606d75c11c08ff94089a9430",**

**"name": "Georgs Honeypot",**

**"description": "Honeypot",**

**"modified": "2023-05-27T14:30:02.565000",**

**"created": "2021-04-07T09:05:05.353000",**

**"tags": ["honeypot", "kfsensor", "rdp", "ssh"],**

**"references": [],**

**"public": 1,**

**"adversary": "",**

**"targeted\_countries": [],**

**"malware\_families": [],**

**"attack\_ids": [],**

**"industries": [],**

**"TLP": "white",**

**"cloned\_from": null,**

**"export\_count": 2939,**

**"upvotes\_count": 0,**

**"downvotes\_count": 0,**

**"votes\_count": 0,**

**"locked": false,**

**"pulse\_source": "api",**

**"validator\_count": 0,**

**"comment\_count": 0,**

**"follower\_count": 0,**

**"vote": 0,**

**"author": {**

**"username": "georgengelmann",**

**"id": "144450",**

**"avatar\_url": "https://otx.alienvault.com/assets/images/default-avatar.png",**

**"is\_subscribed": false,**

**"is\_following": false**

**},**

**List the different section of information available for this IP address (found in the sections array near the bottom of the JSON.**General, geo, reputation, url\_list, passive\_dns, malware, nids\_list, http\_scans.

1. On the left side of the page, find the </> Generate Code dropdown. This function allows you to turn your tested API requests into code that will do the same thing. Open the dropdown and select Python. Python code to make the OTX web request will show up on the right side of the page. Paste the code below:

import requests

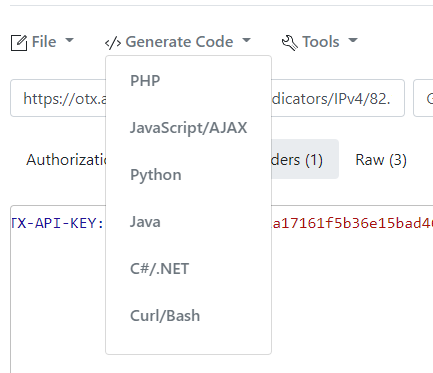
from requests.structures import CaseInsensitiveDict

url = "https://otx.alienvault.com/api/v1/indicators/IPv4/82.165.37.46/general"

headers = CaseInsensitiveDict()

headers["X-OTX-API-KEY"] = ": 8927c6784bced246627e0f7091fce0a41fe534c0067dfeacb6782b8f41fedc08"

resp = requests.get(url, headers=headers)  
print(resp.status\_code)



**API Code for OTX Python request pasted below:**

**8927c6784bced246627e0f7091fce0a41fe534c0067dfeacb6782b8f41fedc08**

Part 2 - STIX and TAXII

TAXII is protocol for exchanging threat information is a standard JSON format called STIX. We will only be using the TAXII and STIX 2.0 protocols as all free TAXII and STIX 1.0 services have been discontinued.

1. From the IS Lab virtual machine desktop, open a Firefox window. In thewindow, navigate to <https://10.98.100.11:4200>. Login in use the username analyst# and password An@lyst#!! where # is your student/analyst number.

Graphical user interface, application, Word

Description automatically generated

1. We will use a web request command line tool to fetch STIX data from MITRE using TAXII 2.0. In your terminal type the following command to perform a discovery: curl –s –X GET –H “Accept: application/vnd.oasis.taxii+json; version=2.0” <https://cti-taxii.mitre.org/taxii> | jq.

Graphical user interface, text, application

Description automatically generated

1. Notice the api\_roots array. This is a list of URLs where we can get more information. We need to make our next request to this URL. Type the command: curl –s –X GET –H “Accept: application/vnd.oasis.taxii+json; version=2.0” <https://cti-taxii.mitre.org/stix> | jq. This will provide a summary of the

Text

Description automatically generated

1. Now that we have queried our API root, we need to see what collections we have available for this API root. This is the actual library that will have our STIX threat intelligence data. All we have to do is add /collections onto the end of our URL per the TAXII 2.0 specification. Run the following command: curl –s –X GET –H “Accept: application/vnd.oasis.taxii+json; version=2.0” <https://cti-taxii.mitre.org/stix/collections> | jq.

Graphical user interface, text, application

Description automatically generated

**How many separate collections are available?**

**4**

**What are the names/titles of each?**

**Enterprise ATT&CK, PRE-ATT&CK, Mobile ATT&CK, ICS ATT&CK.**

1. Now we can request the objects stored in a collection. We will use the collection titled “Enterprise ATT&CK”. To request these objects, we have to add the id and the word objects onto the end of the URL. Use the following command to request the correct collection. Notice how, because this is STIX v2.0, that the returned data is in JSON format and not XML like STIX v1.0. Command: curl –s –X GET –H “Accept: application/vnd.oasis.stix+json; version=2.0” <https://cti-taxii.mitre.org/stix/collections/95ecc380-afe9-11e4-9b6c-751b66dd541e/objects> | jq. Remember to change the Accept header to stix from taxii at the appropriate place. There is a lot of data so you can press CTRL+C to stop curl from having to retrieve all of the data.

Graphical user interface, text, application

Description automatically generated

**Paste a screenshot of the return STIX v2.0 JSON data below.A screenshot of a computer

Description automatically generated with low confidence**