1 Chapter 7: Handling Artefacts

1 Chapter 7: Handling Artefacts 1
- 1.1 Handling Domains (URLs) 2
  - 1.1.1 Chasing a URL 2
  - 1.1.2 Look for ‘similar’ websites 4
  - 1.1.3 Look for new sites 4
  - 1.1.4 Social media references to the site 4
- 1.2 Handling Tweets 5
  - 1.2.1 Chasing a hashtag 5
  - 1.2.2 Chasing botnets 5
- 1.3 Chasing an image 6
- 1.4 Handling Video and Audio 6
  - 1.4.1 Checking video 6
  - 1.4.2 Save an audio file from Facebook Messenger 6
- 1.5 Searching through Facebook Groups 7

Artefacts are the things we can see online - they’re what we track and use to understand what’s happening in an incident, how everything in it fits together, and what we can usefully pass on as information about it at the incident level, or usefully do to influence it. The artefacts that we see most often include:

- Tweets
- Twitter accounts
- Facebook groups
- Domains - websites
- Hashtags
- Images
- Videos
- Audio fragments (e.g. voice messages)
- yas

The next layer up includes:

- Narratives
- Botnets

The basic questions: What is this thing. How is it impacting the things we care about? Are there other teams doing something about it? What can we do about it? How much impact can we make in the things we care about, for the resources we need to expend?

1.1 Handling Domains (URLs)

1.1.1 Chasing a URL

So you've got a URL. Now what? Well, you probably want to know about the URL - who created it, when, what's it connected to etc.

Check for company

- Is anyone else tracking this url? Check reddit etc. - you might save yourself time if other groups have already tracked lists, social media etc.

All the Google Dorks! (h/t to [Name Redacted])

Using Google Dorks to Check Primary sources (from Henk van Ess's Finding patient zero)

Websites as primary sources: This is useful when your searches within specific sites or urls are coming up empty

Step 1: Look at the failing link

- Ex. https://www.sec.gov/litigation/apdocuments/3-17405-event-11.pdf
- Pull out just the domain name and Top Level Domain (Ex. sec.gov)

Step 2: Use "site:"

- Go to a generic search engine.
- Start with the query ("Dutch police") and end with "site:" followed directly with the URL (no spaces).
- Ex. "Dutch police" site:sec.gov

Step 3: Adapt the "primary source formula" to your needs

- Include specific folders (Ex. "Dutch police" site:sec.gov/public)
- Predict folders you think might be there

Following the trail of Documents

Step 1: Establish the document type

- Is it a doc | pdf | xls | txt | ps | rtf | odt | sxw | psw | ppt | pps | xml file?
- Use filetype: and the type of file with no spaces (Ex. "filetype:pdf")

Step 2: Include a phrase you'd like to search with in the document (could include a date)

- Ex. You're searching for an invitation to an event from May 13, 2014, event. (Be sure to search for both the cardinal and ordinal forms, May 13 and May 13th.)

Step 3: Who is involved?

- Do you know the creator/host and it's website?
- Ex. The organizer is "Friends of Science" and its website is friendsofscience.org.

When you combine all three steps, the query in Google will be:
"May 13th, 2014" filetype:pdf site:friendsofscience.org

Filtering social media for primary sources
YouTube
YouTube's search tool has a problem: it won't let you filter for videos that are older than one year. To solve this,

- In a Google search include the keywords and site:youtube.com
- manually enter the preferred date into a Google.com search by using the "Tools" menu on the far right
- Then select "Any time" and "Custom Range."

Process for investigating the authenticity of a website :

Web searching a domain: Since we want to find out what other sites are saying about the site while excluding what the site says about itself, we use a special search syntax that excludes pages from the target site

- Search syntax is website -site:website
- (Ex. baltimoregazette.com -site:baltimoregazette.com)
- Scan the set of results looking for sites we trust

Finding out who runs a site with WHOIS :

- Enter the domain name into the WHOIS Domain Tools
- Note who the domain was registered to
    - Unfortunately, WHOIS blockers have dramatically reduced the value of WHOIS searches, so you may only find a proxy.
- Note when the domain was registered

Use a backlink checker like ahrefs or smallseotools that allows you to see all websites that link to a particular site

Look up the url

- Builtwith:
    - look on builtwith.com - if you're lucky that will tell you when and who
    - It will also tell you which sites have the same tags as this site: this helps you find connected sites
    - Use CSC code run_builtwith.ipynb - same thing, but gives you json and a dataframe of those connected sites

Look at the URL contents

- Are there phrases you can use in a googlesearch, to find related objects? Run the search that allows repeated results, to see identical pages. About and terms pages are usually good places to look for these.
- Use CSC code googlesearch_for_terms.ipynb to search for terms/ pages.
- Are there people connected to the site? Start searching for them
- Are there companies?

Think about geography etc

- Look at the title and url of the site. Do they have elements that might be repeated? E.g. if you have xxxmichigan.com, check for the same pattern with other states' names, e.g. xxxwisconsin.com. Astroturfers try to cover an area, whether it's geographical or demographic, and if they're doing it for money, they'll usually have multiple sites.

Look for links

- Check social media - are there references to the URL, or groups / pages / accounts with the same name?
- If there are references to the URL, are there common hashtags, phrases or people in common you can use to search for more sites?

Examples:

- http://overcognition.com/2020/01/22/data-safari-rough-notes-pink-slime-network/

1.1.2 Look for 'similar' websites

Typosquatting is when you create a site whose url is *almost* the same as a real or well-known one, often using combinations of letters (e.g. 'nn' instead of 'm') or urls (e.g. .gov.us) to fool people on a casual glance.

Useful python libraries for generation typosquats include dnstwist

- Near-duplicates https://github.com/justinnbt/SnaPy
- https://github.com/topics/typosquatting

Feed of domains that were created each day: https://whoisds.com/newly-registered-domains

- Idea: we could search this feed each day for domain names matching the things of interest to us, e.g. MMS

1.1.3 Look for new sites

Github code check_new_registrations.ipynb searches for strings of interest in newly-registered domains (from https://whoisds.com/newly-registered-domains). But newly registered alone isn't really an indication of anything; domains that are newly registered and active all within 24hrs, are worth watching, as is any recently active and questionable domain. We have e.g. the Zetalytics API for searching through those.

1.1.4 Social media references to the site

Crowdtangle chrome extension will give you a list of references to a site you're looking at, on Facebook, Twitter, Instagram and Reddit https://apps.crowdtangle.com/chrome-extension

1.2 Handling Tweets

1.2.1 Chasing a hashtag

“what do we consider worthy of collecting from twitter?” - FrankC

Good question. The TL;DR is that the reason we use the code that we do (andypatel_gettwitter.py from CSC tracking repo) is because we’re looking for the objects that dominate and are related to the hashtag:

- we want to know which users are promoting it
- Which other hashtags are used heavily with it
- Which users on the hashtag are in suspicious configurations - e.g. one user linked out to lots of other people who aren’t connected to each other (that’s someone either pushing or pulling, depending on the direction of the links), or groups of users connected heavily to each other but not to anyone else on that hashtag (typical configuration for a botnet)
- we want to know which URLs are associated with the hashtag - if this is being used to make money, that money has to come from somewhere, and that’s usually either online advertising, merchandise or paid services: either way, each of those is going to have a web address associated with it, and any grifter worth their salt is going to be pushing that address heavily
- We also collect images - that gives a good idea of what the themes are, because most good disinformation merchants know that images are more often exchanged than text. That’s why you see all those posters with text on

The finding the configurations part - we use Gephi to look at the network; botnets and distributors stand out like little flowers in a Gephi network. But we could use networkx to do the same thing. There are also a set of tools in OSOME that will help you examine relationships quickly.

Raw data is useful too - it’s where we start. But really, in social engineering, it’s the relationships that count.

1.2.2 Chasing botnets

I use bot sentinel and tools like it - ones like Hamilton68 monitor accounts from nation state actors (Russia, China etc - think embassy twitter feeds, RussiaToday etc), ones like Botsentinel monitor accounts active in earlier campaigns that might or might not be bots. The most valuable thing they give you is trends: what the recent chatter online is.

Bot detection is an art now. Once upon a time, it was as easy as “there are 100 accounts posting all the time, and they’re all posting the same text”, and finding them was basically “look for the Qanon hashtags”. Now it’s more subtle. There are some rules of thumb, like being suspicious of anything tweeting more than 100 times a day, but there’s more to it, and a bunch of tools to help.

1.3 Chasing an image

There are a few things you’re going to want to do with an image:

- Extract the text from it
- See where else it exists online
- Check to see if it’s been altered / is fake

Extracting text: You can usually extract text from images using OCR (optical character recognition). There are libraries like Tesseract that can be called from Python (as e.g. pytesseract), but they have mixed results. A more reliable way to do this is to use the OCR built into search engines to pull the text from each image: yandex.com appears to be best at this (although always check because OCR still doesn’t produce perfect results) but is Russian: if that’s an issue for you, bing.com image search does this too.

Seeing where else an image is online:

- Mostly you’ll be doing this by hand for new images, but a good first check is to see if an image (e.g. a photo) has been reused from an earlier event. Reverse image search from yandex.com and bing.com works well - tineye.com will call all the big image search engines for you (and you can laugh at some of the things they return…).

Checking for alterations: Bellingcat are the masters of online image forensics, and have a good guide to this (Bellingcat guide). Look at tools like FotoForensics.

1.4 Handling Video and Audio

1.4.1 Checking video

https://twitter.com/InVID_EU

1.4.2 Save an audio file from Facebook Messenger

The workaround is:

- Using Chrome browser (but NOT on mobile)
- Access facebook via m.facebook.com
- Then click on the messenger icon
- Go to the chat that has the audio
- Right mouseclick on the (...) at the end of the message and you'll have the option to "Save Audio As"

1.5 Searching through Facebook Groups

A lot of Covid19 disinformation is happening and/or moving at some point through facebook groups. We've been tracking some of these by hands whilst working out how to automate creating watchlists of groups, pages, accounts to check for new disinformation incidents forming before they hit the mainstream press.

Some academic references on this, focussed on antivax (one of the best-known and well-studied modern conspiracy theories)

- The online competition between pro- and anti-vaccination views
    - Hidden resilience and adaptive dynamics of the global online hate ecology | Request PDF
    - https://science.sciencemag.org/content/352/6292/1459 with https://science.sciencemag.org/content/sci/suppl/2016/06/15/352.6292.1459.DC1/Johnson-SM.pdf
-