

Extracting Malware Information from Cybersecurity Text

Michael Maiden, Casey Hanks, Leyton Lineburg
University of Maryland, Baltimore County

OnRamp II Symposium
13-14 October 2021

This work was funded, in part, by a grant from the NSA through the On-Ramp program and by the National Science Foundation under Grant Number 2114892.

Acknowledgements

- Dr. Ahmad Ridley, NSA Moderator
- Dr. Anupam Joshi, Professor
- Dr. Tim Finin, Professor
- Priyanka Ranade, Graduate Student
- Aritran Piplai, Graduate Student

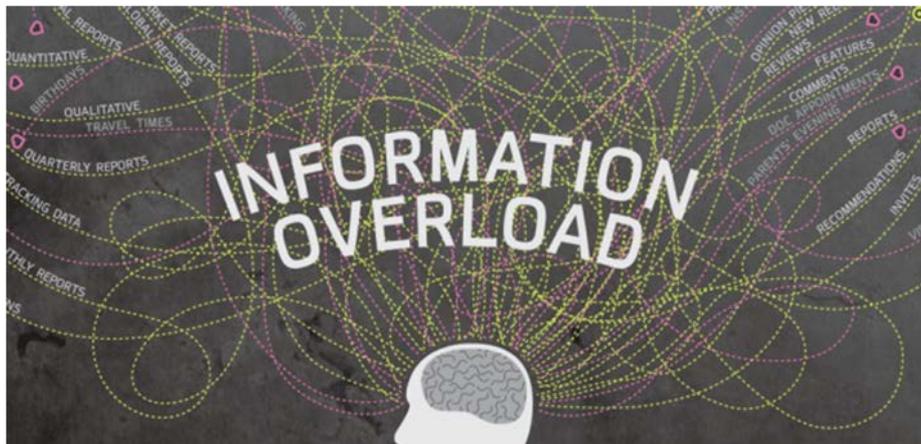


Thank you to the NSA for their support!

Research Goals and Background

Motivation

- Cyber Threat Intelligence (CTI) **aids cyber analysts** in their daily tasks of discovering and understanding evolving cybersecurity threats, exploits, and vulnerabilities
- Too much data to process!



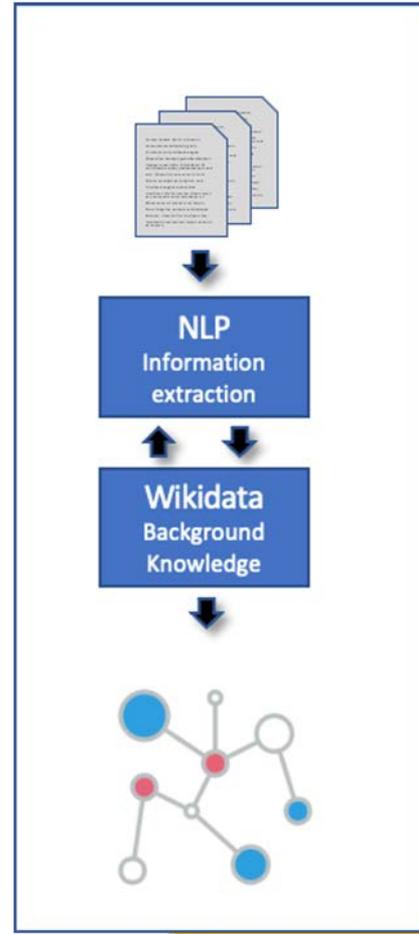
Research Question

Can we use state of the art NLP tools to process large corpora of CTI samples and provide *actionable insights* to analysts?

Overall Research Goal

Prototype a system to build and maintain a **knowledge graph** of current **cybersecurity data** extracted from public **text reports**

- Using state-of-the-art NLP systems, e.g., [spaCy](#)
- Using public knowledge graphs for background knowledge, e.g., [Wikidata](#)



Project Overviews

Use SpaCy for NLP
Information Extraction:

- Data extraction and compilation
- SpaCy capabilities and utilization
- Progress, implementation, next steps

Use Wikidata for additional data:

- Catalog Wikidata's cyber knowledge and improve it
- Use its query service to access cyber information
- Improve our entity linker to link cyber entities to Wikidata

Background: Information Extraction for Cybersecurity

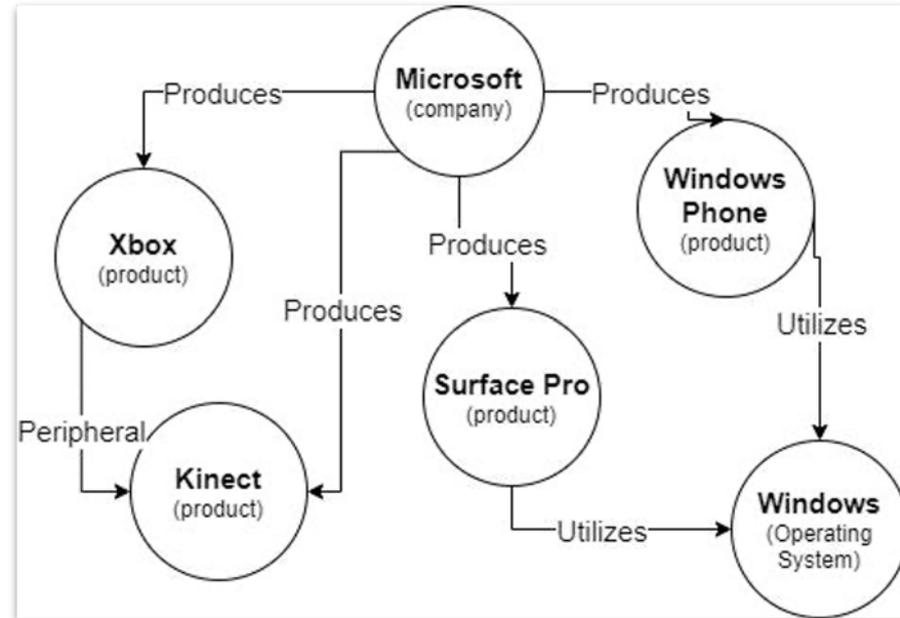
Types of Cybersecurity Text Data:

- Structured data - data that has a specific schema or set of rules for where data is stored, so it is easily searchable
- Unstructured data: Free-text, no standard content specification

Information Extraction is the process of extracting structured data from unstructured data

Background on Knowledge Graphs

- A knowledge graph encodes data as a graph with entities as nodes and relations as edges
- Often represented as a set of ordered triples
(head entity, relation, tail entity)
- Example:
(Microsoft, Produces, XBOX)
(XBOX, Peripheral, Kinect)



Background on Wikidata

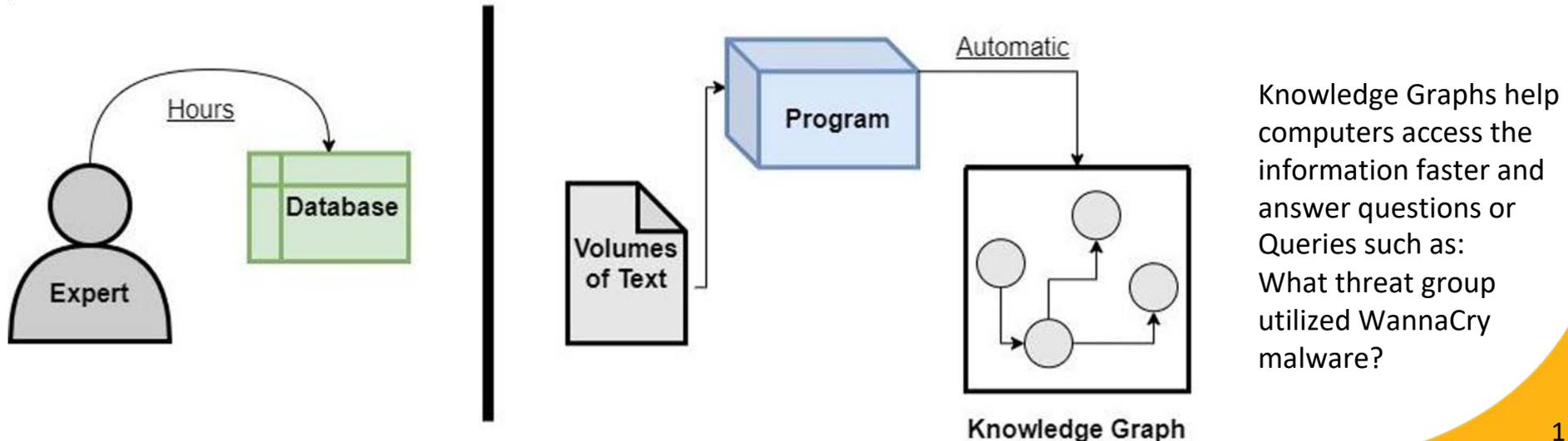
- What is Wikidata?
 - A knowledge graph derived from Wikipedias and more
 - A communal, open-source knowledge graph
- How is it accessed?
 - Has its own query service using SPARQL
- Cybersecurity knowledge
 - Has limited information available on cybersecurity
 - Each entity contains its own qualifiers

Michael Maiden, Casey Hanks

Using SpaCy to improve information extraction from cybersecurity text

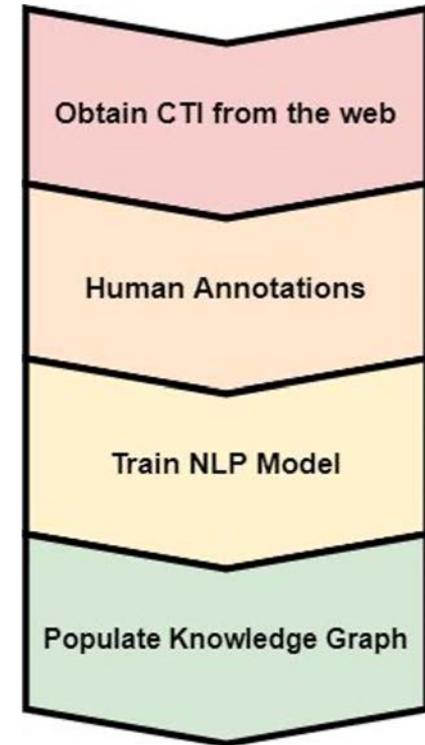
Purpose

The purpose of this project is to create an automated method of transforming text based cybersecurity data to a computer friendly format



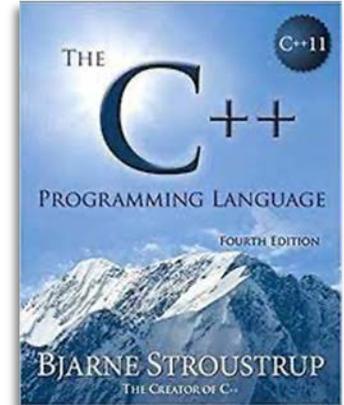
Overview

- Cyber Threat Intelligence (CTI) is often disseminated as text
- Expressing CTI information in a knowledge graph allows integration, inference, and machine understanding
- We use synthesized data, language processing and machine learning to do this



Extracting Information from Text

- Natural Language Processing (NLP) tools extract entities such as relations and events from text
- These tools are trained to recognize
 - People, Places, and Time
- Not trained for extracting cybersecurity entities
 - Malware
 - Software



all major versions from VS6 to VS2019 CARDINAL : C and C++ WORK_OF_ART run-time libraries Active Template Library ORG (ATL ORG) and Microsoft Foundation ORG Class (MFC ORG) libraries The following open-source projects as compiled with VS2015 PRODUCT , VS2017 PRODUCT , and VS2019 CARDINAL : CryptoPP

Extracting Cyber Information

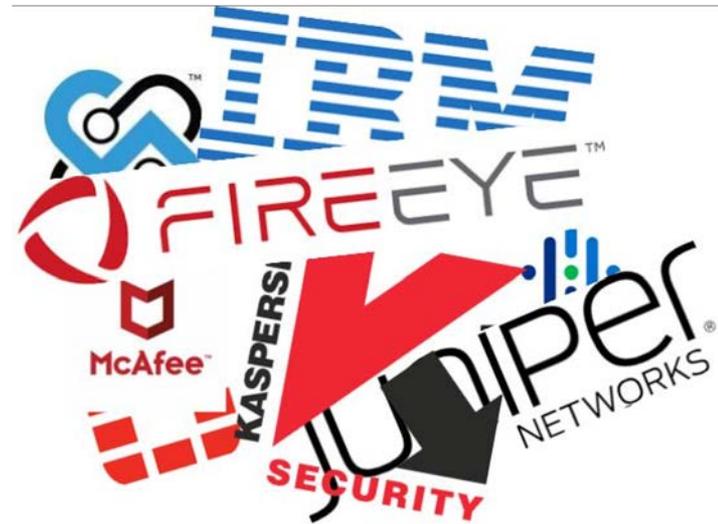
Existing NLP tools have not seen cybersecurity entities before, so must be re-trained for the cybersecurity domain by

- Collecting relevant text examples
- Getting human annotations for entities & relations
- Training machine learning tools to recognize mentions of entities, relations, and events

Methodology

Building a Cybersecurity Text Collection

- Collected open-source cybersecurity text from a variety of sources
 - 3,269 paragraphs have been collected across 8 sources
- Developing tools to automatically update corpus as cyber exploits, vulnerabilities, and news sources continue to evolve



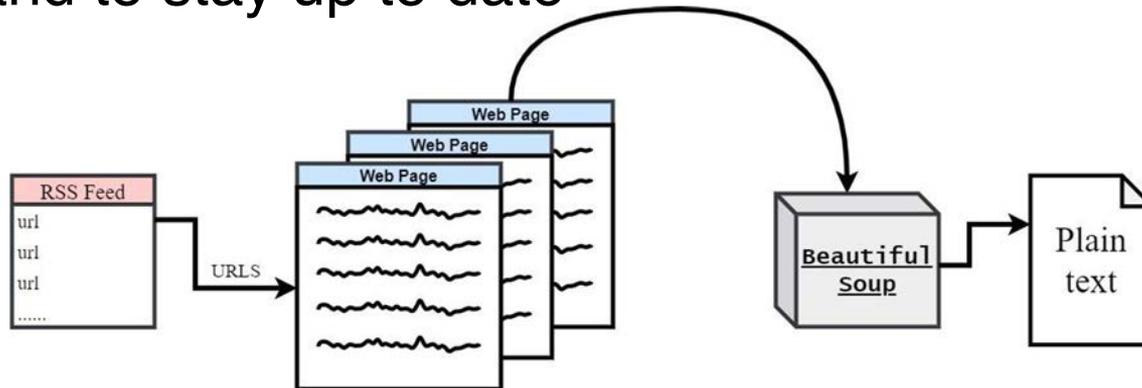
Blogs in Our Text Collection

- IBM Security
- Fortinet
- Palo Alto Networks
- Intelligence
- Kaspersky
- Juniper Networks
- McAfee
- Securonix
- Fireeye

```
"data": [  
  {  
    "http://www.fireeye.com/blog/threat-research/2021/07/capa-2-better-stronger-faster.html": {  
      "metadata": {  
        "dateAccessed": "2021-07-27",  
        "dateCreated": "2021-07-19",  
        "source": "www.fireeye.com"  
      },  
      "text": "We are excited to announce version 2.0 of our open-source tool called capa. capa automatically identifies capabilities in programs using an extensible rule set. The tool supports both malware triage and deep dive reverse engineering. If you haven't heard of capa before, or need a refresher, check out our first
```

Extracting text from web pages

- CTI information is extracted from multiple cybersecurity blogs using BeautifulSoup, a Python library for pulling data out of HTML sources
- RSS feeds of websites are used to extract URLs of new articles and to stay up to date



The UNC2529 Triple Double: A Trifecta Phishing Campaign

NICK RICHARD, DIMITER ANDONOV

MAY 04, 2021 | 25 MINS READ

[#THREAT ACTORS](#) [#UNC](#)

In December 2020, Mandiant observed a widespread, global phishing campaign targeting numerous organizations across an array of industries. Mandiant tracks this threat actor as [UNC2529](#). Based on the considerable infrastructure employed, tailored phishing lures and the professionally coded sophistication of the malware, this threat actor appears experienced and well resourced. This blog post will discuss the phishing campaign, identification of three new malware families, DOUBLEDRAG, DOUBLEDROP and DOUBLEBACK, provide a deep dive into their functionality, present an overview of the actor's modus operandi and our conclusions. A future blog post will focus on the backdoor communications and the differences between DOUBLEBACK samples to highlight the malware evolution.

UNC2529 Phishing Overview

Mandiant observed the first wave of the phishing campaign occur on Dec. 2, 2020, and a second wave between Dec. 11 and Dec. 18, 2020.

Example Article in Our Collection

 **MANDIANT**

spaCy NLP Tools

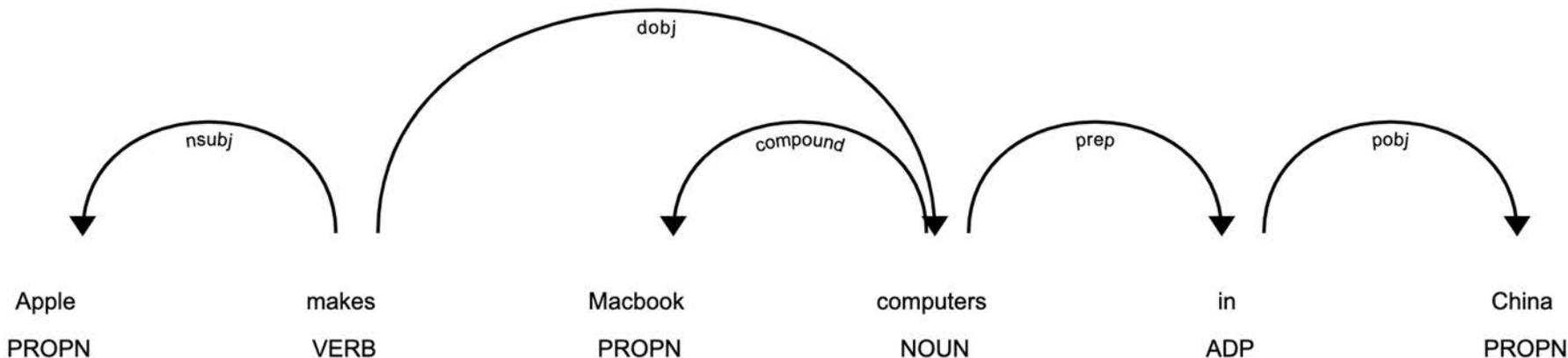
SpaCy is an state of the art, open source, NLP tool that supports

- Tokenization: segmenting text into sentences and words
- Syntactic analysis: parsing text to identify parts of speech, phrases, and sentence structure
- Text Similarity: computes similarity of text spans via embeddings
- Named Entity Recognition: Identify text spans for entities and their types, e.g., Apple is an entity of type Organization
- Training an NER model: with proper training data, SpaCy can be trained to identify new entity types and their entity labels
- Annotated text: the format of the training data

spaCy example

```
doc = nlp("Apple makes Macbook computers in China")
displacy.render(doc, style="ent")
displacy.render(doc, style="dep")
```

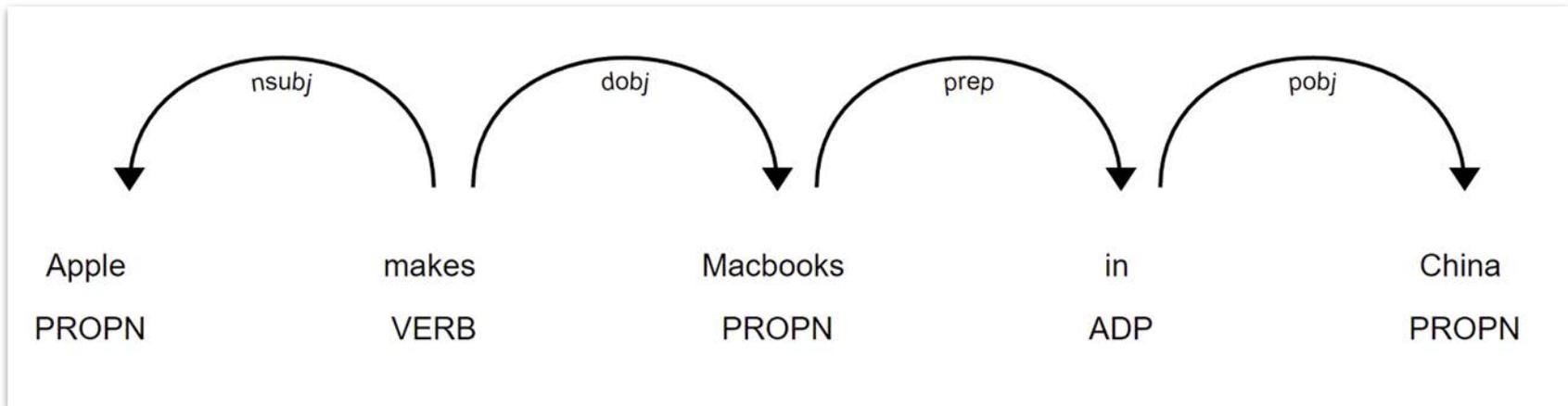
Apple **ORG** makes Macbook **PRODUCT** computers in China **GPE**



```
doc = nlp("Apple makes Macbooks in China")
ent = displacy.render(doc, style="ent")
dep = displacy.render(doc, style="dep")
```

Apple **ORG** makes Macbooks **ORG** in China **GPE**

spaCy example



- spaCy's generic model doesn't recognize Macbook as a product or computer
- Need to train on cybersecurity text and/or link items to background knowledge like Wikidata (noun chunks)

The UNC2529 Triple Double: A Trifecta Phishing Campaign

NICK RICHARD, DIMITER ANDONOV

MAY 04, 2021 | 25 MINS READ

[#THREAT ACTORS](#) [#UNC](#)

In December 2020, Mandiant observed a widespread, global phishing campaign targeting numerous organizations across an array of industries. Mandiant tracks this threat actor as [UNC2529](#). Based on the considerable infrastructure employed, tailored phishing lures and the professionally coded sophistication of the malware, this threat actor appears experienced and well resourced. This blog post will discuss the phishing campaign, identification of three new malware families, DOUBLEDRAG, DOUBLEDROP and DOUBLEBACK, provide a deep dive into their functionality, present an overview of the actor's modus operandi and our conclusions. A future blog post will focus on the backdoor communications and the differences between DOUBLEBACK samples to highlight the malware evolution.

UNC2529 Phishing Overview

Mandiant observed the first wave of the phishing campaign occur on Dec. 2, 2020, and a second wave between Dec. 11 and Dec. 18, 2020.

Example Article in Our Collection

 **MANDIANT**

How the Text is Stored

```
{
  "http://www.fireeye.com/blog/threat-research/2021/05/unc2529-triple-double-trifecta-phishing-campaign.html": {
    "metadata": {
      "dateAccessed": "2021-07-27",
      "dateCreated": "2021-05-04",
      "source": "www.
Mandiant observed the first wave of the phishing campaign occur on Dec. 2, 2020, and a second wave between Dec. 11 and Dec. 18, 2020. During the initial flurry, Mandiant observed evidence that 28 or
files-upload_string http://<fqdn>/files-upload-<string> http://<fqdn>/get_file-id-<string> http://<fqdn>/get_file-id-<string> http://<fqdn>/zip_download-<string> http://<fqdn>/zip_download-<string>
cond-stage memory-only dropper, which Mandiant tracks as DOUBLEDROP, from either hxxp://p-leh[.]com/update_java.dat or hxxp://clanvisits[.]com/mini.dat. The downloaded file is a heavily obfuscated P
<string> http://<fqdn>/dowld-<string> http://<fqdn>/download-<string> http://<fqdn>/files-<string> http://<fqdn>/id-<string> http://<fqdn>/upld-<string> Of note, the DOUBLEDROP downloader observed i
.php, hxxps://widestaticsinfo[.]com/admin4/client.php, hxxps://secureinternet20[.]com/admin5/client.php, and hxxps://adsinfocoast[.]com/admin5/client.php. Three of these domains were registered after
a-based electronics manufacturing company. Another example is a freight / transport company that received a phish with subject, "compton ca to flowery branch ga", while a firm that recruits and plac
applications for medicare supplement & part d
2nd Table 2: Sample insurance industry subject lures Interestingly, one insurance company with offices in eastern Texas received a phish with a subject
2nd Table 3: Sample pattern subject lures Industry and Regional Targeting UNC2529's phishing campaign was both global and impacted an array of industries (Industry and Regional Targeting graphics ar
er (DOUBLEDROP) (or alternatively an Excel document with an embedded macro), a dropper (DOUBLEDROP), and a backdoor (DOUBLEBACK). As described in the previous section, the initial infection vector s
system. The rest of the components are serialized in the registry database, which makes their detection somewhat harder, especially by file-based antivirus engines. Ecosystem in Details DOUBLEDROP D
f the DOUBLEDROP dropper DOUBLEDROP Dropper component Overview The dropper component is implemented as an obfuscated in-memory dropper written in PowerShell. Two payloads are embedded in the script-
ue: <default> * data: <rnd_guid_1> * value: <last_4_chars_of_rnd_guid_0> * data: <encoded_loader> * key: VersionIndependentProgID * value: <default> * data: <rnd_guid_1> * value: <first_4_chars_of_
The actual RC4 key within the buffer is given by the following calculations, shown in Figure 6 (note that the key is reversed!). <relative_offset> = buffer[32] buffer[32 + <relative_offset> + 1] = <
hell process, it creates a scheduled task with an action specified as TASK_ACTION_COM_HANDLER and the class ID - the {<rnd_guid_2>} GUID (See Figure 5). Once executed by the system, the task finds t
elevated privilege case. Bootstrap the bootstrap is implemented as an obfuscated PowerShell script, generated dynamically by the dropper. The content of the code is saved under the emulator's class
)) ) Figure 7: De-obfuscated and sanitized bootstrap code Note that the actual values for <base64_encoded_path_to_launcher>, <launcher_reg_val>, and <xor_val> are generated on the fly by the dropp
<first_4_chars_of_rnd_guid_0> value (see Figure 5) and then uses it to decrypt the payload. Once the payload is decrypted, the launcher allocates virtual memory enough to house the image in memory,
```

- No newlines
- No paragraphs
- Hardly readable

Making it readable

- For human annotation we ideally have paragraphs
- Easily digestible, can easily annotate one or two and stop
- Humans identify paragraphs by white space, what if there is none?
- Spacy is good at identifying sentences, but not so much paragraphs.

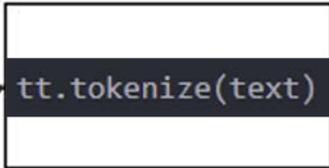
The Solution

```
{ "http://www.fireeye.com/blog
Mandiant observed the first w
files-upload_<string> http://
cond-stage memory-only droppe
<string> http://<fqdn>/dowld_
.php, hxxps://widestaticsinfo
a-based electronics manufactu
applications for medicare sup
2nd Table 3: Sample pattern s
er (DOUBLEDRAG) (or alternati
system. The rest of the compo
f the DOUBLEDROP dropper DOUB
lue: <default> * data: <end a
```

Unreadable Text

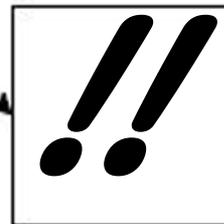


Spacy Sentencizer



```
tt.tokenize(text)
```

Separate Tool
Topical Sections

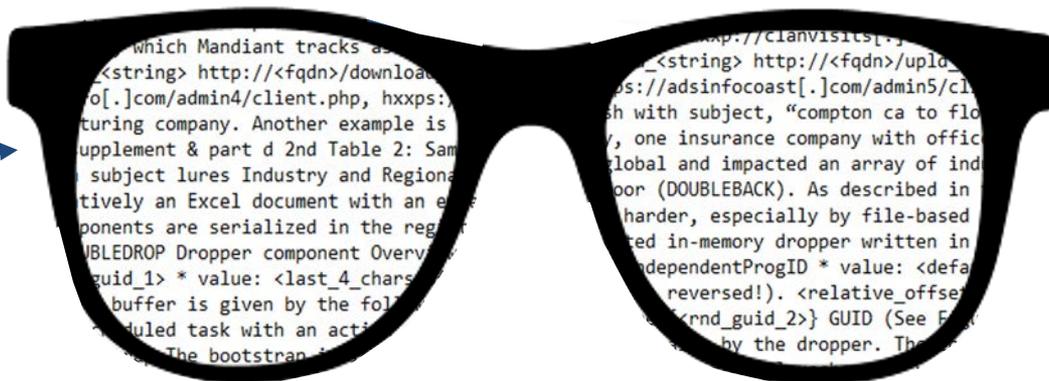


Paragraphs
(4~9 sentences)

Much better!

In December 2020, Mandiant observed a widespread, global phishing campaign targeting numerous organizations across an array of industries. Mandiant tracks this threat actor as UNC2529. Based on the considerable infrastructure employed, tailored phishing lures and the professionally coded sophistication of the malware, this threat actor appears experienced and well resourced. This blog post will discuss the phishing campaign, identification of three new malware families, DOUBLEDROP, DOUBLEDROP and DOUBLEBACK, provide a deep dive into their functionality, present an overview of the actor's modus operandi and our conclusions. A future blog post will focus on the backdoor communications and the differences between DOUBLEBACK samples to highlight the malware evolution. UNC2529 Phishing Overview Mandiant observed the first wave of the phishing campaign occur on Dec. 2, 2020, and a second wave between Dec. 11 and Dec. 18, 2020. During the initial flurry, Mandiant observed evidence that 28 organizations were sent phishing emails, though targeting was likely broader than directly observed.

None of this



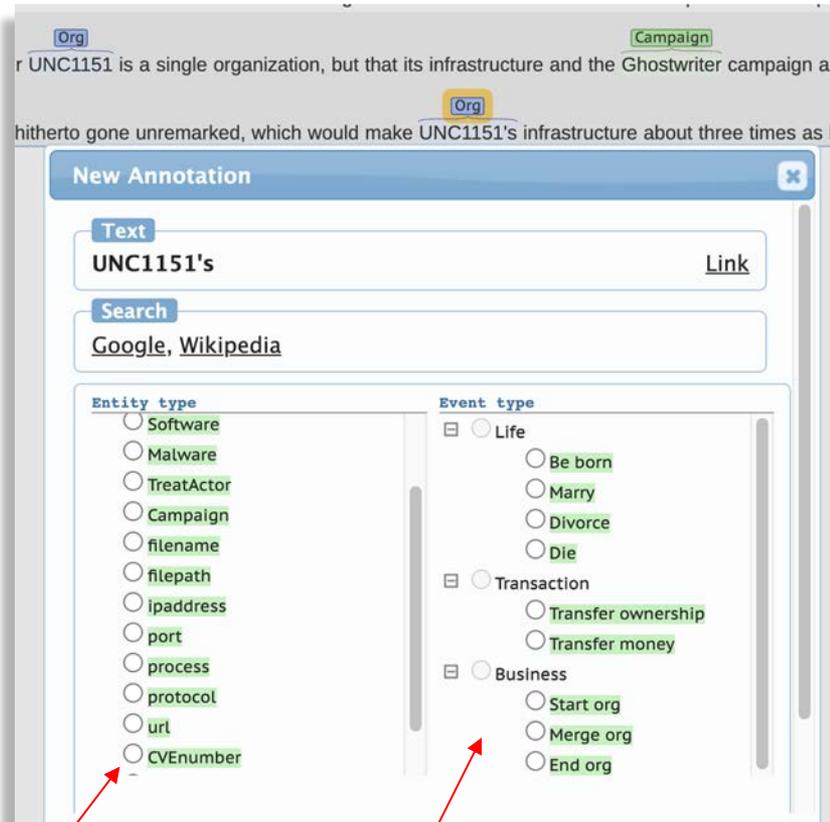
Human Annotations for Training Data

- Machines need many examples to learn
- Multiple humans act as entity recognizer and record occurrences of the desired label
 - (Ex. “MALWARE”)
- Annotation tools are often used to make the process more efficient

```
"entities": [  
  [  
    "4993",  
    "5002",  
    "MALWARE"  
  ],  
  [  
    "5196",  
    "5205",  
    "MALWARE"  
  ],  
]
```

Brat Rapid Annotation Tool

- [BRAT](#) is a web-based tool that allows users to annotate text
- Annotations identify named entities, relations, and events
- BRAT outputs annotation data in a standard format
- Used to train SpaCy models to extract new entities and relations



possible
entities

possible
relations

BRAT Example

- 1 TreatActor UNC2465 has used Advanced IP Scanner, Malware BLOODHOUND, and protocol RDP for internal reconnaissance and lateral movement activities within victim environments.
- 2 The threat actor has used Malware Mimikatz for credential harvesting to escalate privileges in the victim network.
- 3 TreatActor UNC2465 also uses the publicly available Software NGROK utility to bypass firewalls and expose remote desktop service ports, like protocol RDP and protocol WinRM, to the open internet.
- 4 Org Mandiant has observed the threat actor using PsExec and cron jobs to deploy the Malware DARKSIDE ransomware.

A team of human annotators identify entities and relations and assign cybersecurity-relevant types using our central BRAT server

Training SpaCy tools

- SpaCy tools allow us to train Cybersecurity NER models using the BRAT annotations
- The trained model will be able to recognize cybersecurity entities like mentions of malware or threat actors
- Users are able to combine their own models with SpaCy's existing models
 - We can fine-tune an existing SpACY model to fit cybersecurity data

Creating CTI Knowledge Graphs

- Nodes in the knowledge graph will be filled with cybersecurity named entities
 - Ex. Malware, Software, Threat Actors, Campaigns
- Edges represent relations between two entities
 - Threat actors **utilize** malware
- Cybersecurity professionals can use the Knowledge Graphs as additional tools for understanding threat-related information

Preliminary Results & Next Steps



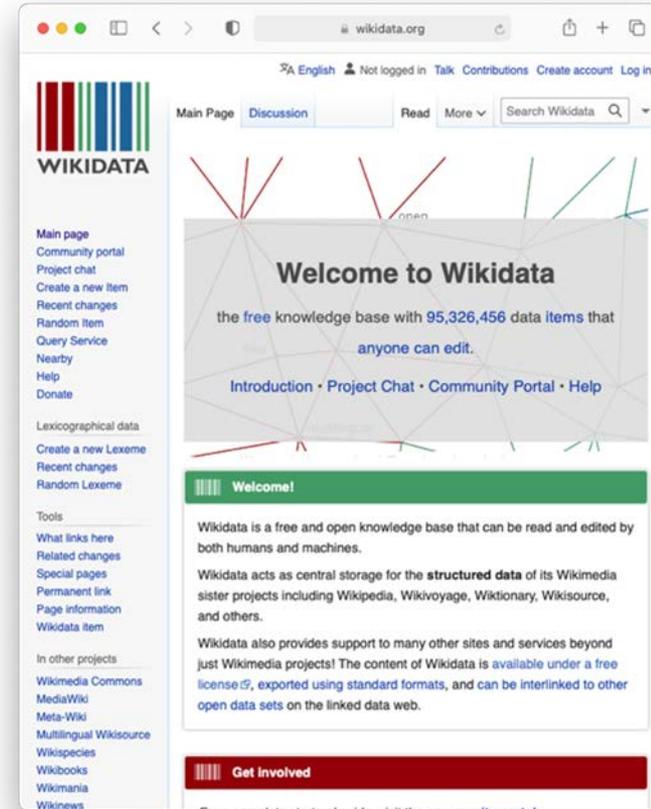
- Compiled collection of cybersecurity-related texts
- Compiling human annotation materials
- Train SpaCY Cybersecurity NER model using labeled, annotated data
- Evaluate the accuracy of the NER model
- Use named entities detected with NER model to strengthen existing cybersecurity knowledge graph

Leyton Lineburg, Mike Anoruo

Surveying and Improving Cybersecurity Knowledge in Wikidata

Wikidata Overview

- Wikidata is an open-source knowledge graph of general background knowledge
- It has both general and useful cybersecurity related knowledge
- We can use it to help train and improve our NLP information extraction systems
- And to develop plans to improve it and keep it current as a cybersecurity resource



Methodology

1. Query and retrieve cyber-related information
2. Manually search for errors and corrections
3. Update Wikidata periodically to keep information accurate and relevant
4. Link entities found in text to Wikidata entities to promote data integration

Knowledge Graphs (KGs)

- Google uses the largely private Google Knowledge Graph to better understand queries and web pages
- Wikidata is an open-source KG we can use for similar purposes
- It has a very rich schema and a powerful underlying semantic representation based on RDF and links to other semantic KGs, like [DBpedia](#)

Wikidata

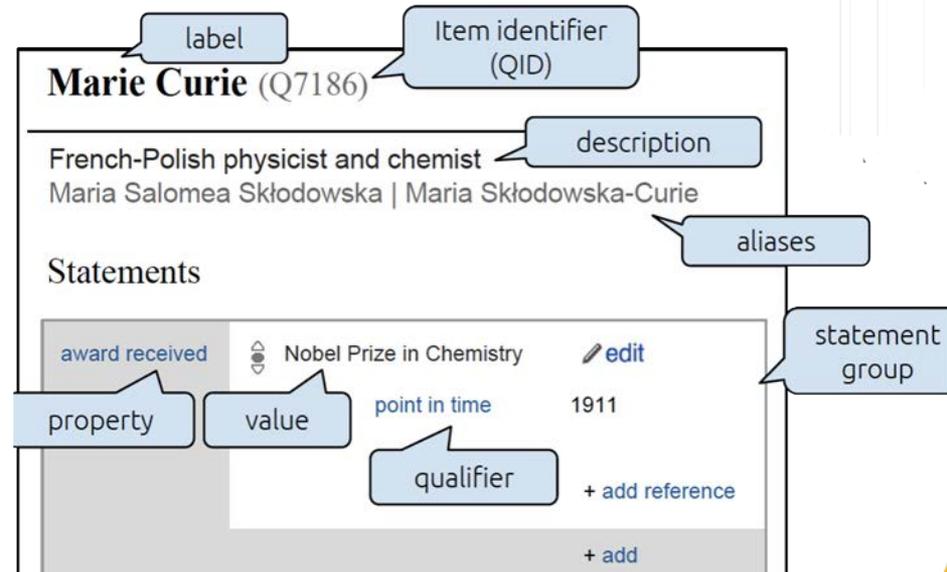
- A community knowledge graph with ~1B statements about ~100M items
- Fine-grained ontology has ~2M types and ~9K properties
- Multilingual: all text values tagged with language id
- Has both a human and query interface
- Many community tools for editing, search, visualization, update



Wikidata web interface for the UMBC entity, Q735049

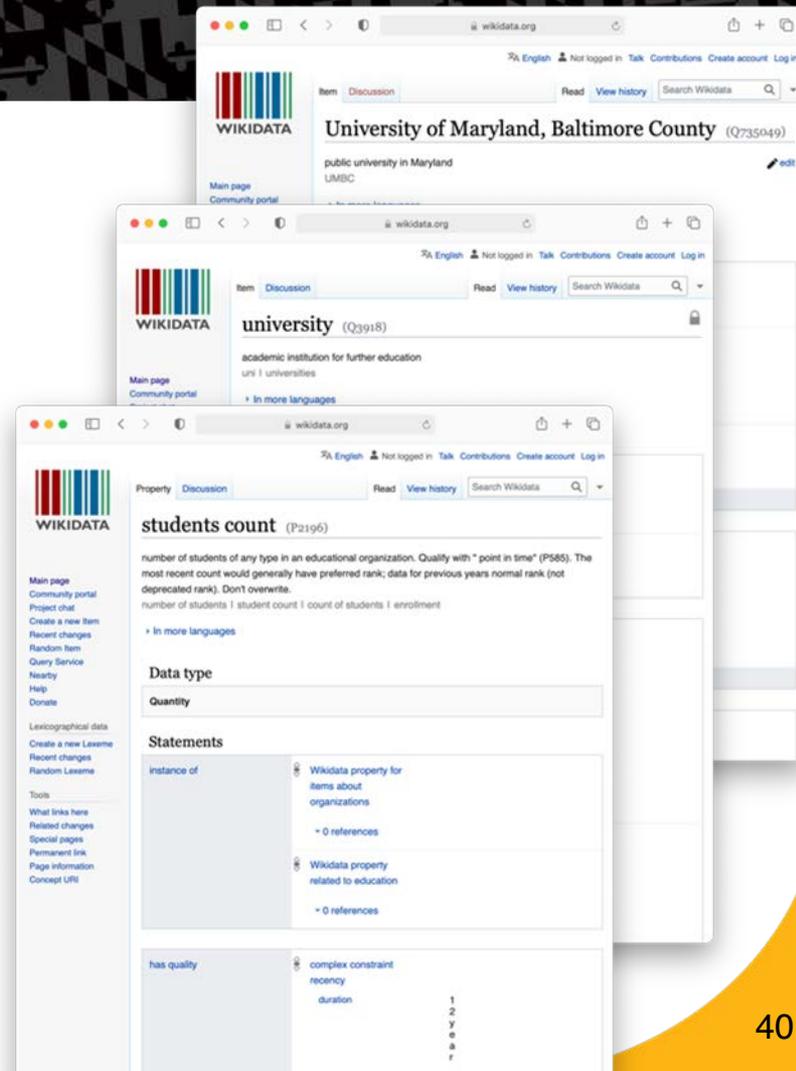
Wikidata Items

- Unique ID, e.g. Q7186
- label (canonical name) in multiple languages
- Short description and some aliases in multiple languages
- One or more types, e.g., Q5 for Human
- Collection of statements with optional qualifiers, references
- Links to entries in Wikipedia sites & other knowledge graphs



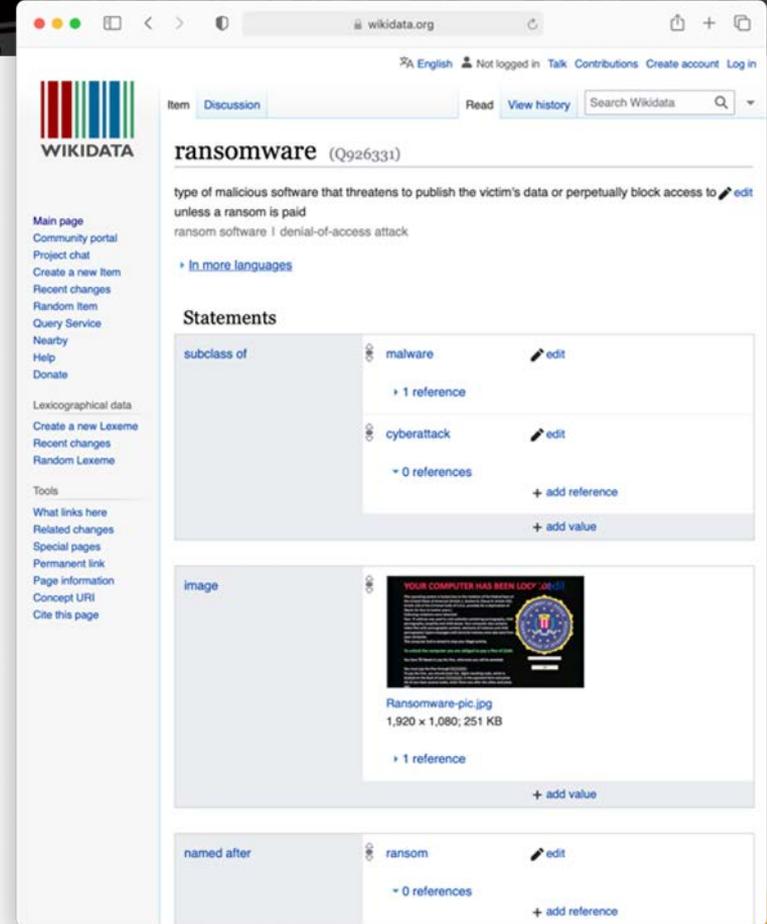
Concepts & Properties

- Wikidata items like UMBC ([Q735049](#)) refer to entities, e.g., instances of a concept
- Some Wikidata refer to a concept or type, e.g., University ([Q3918](#))
- Others refer to properties, e.g., students count ([P2196](#))
- These concepts form Wikidata's rich ontology



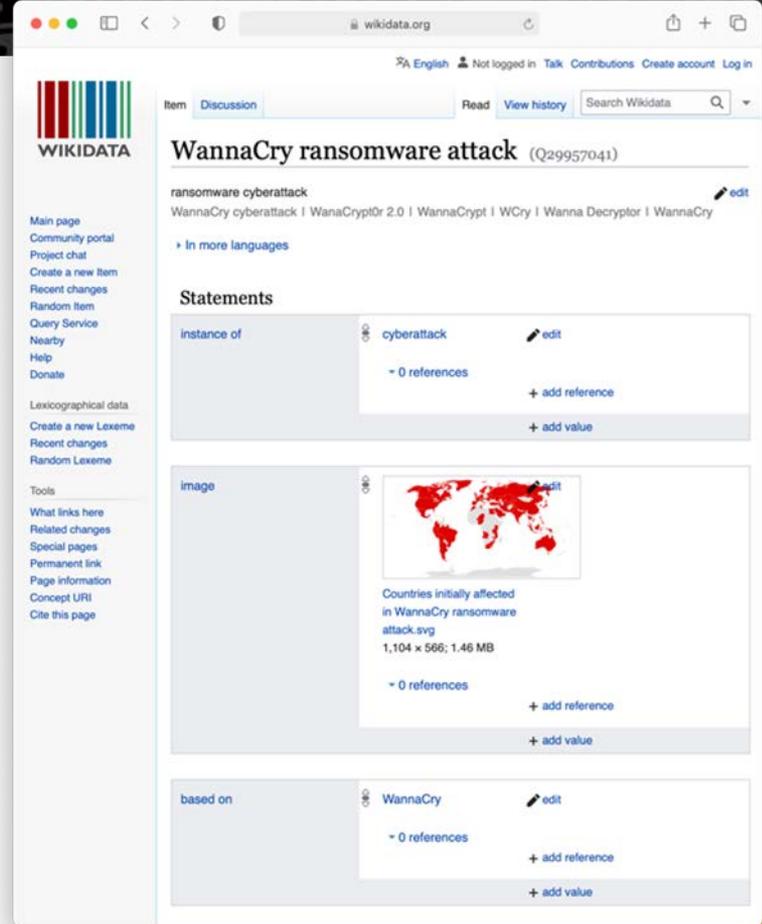
Cybersecurity Concept

- Wikidata has data on many cybersecurity related concepts
- Including, but not limited to:
 - Malware
 - Types of malware
 - Cyber attacks
 - Known threat actors
 - Operating system concepts
 - Key software products



Cybersecurity Event

- Wikidata has data on many cybersecurity events, both generic and specific
- Has information of cyber attacks, such as a description of the attack, what areas were affected, when it occurred, outcomes, etc.



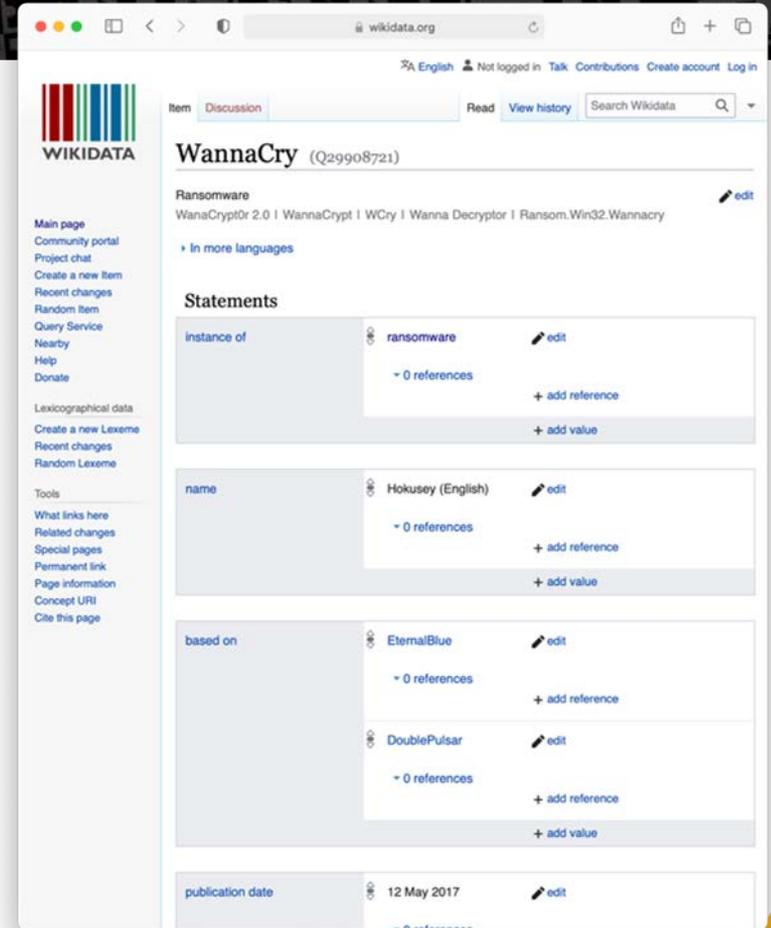
The screenshot shows the Wikidata page for the item 'WannaCry ransomware attack' (Q29957041). The page is in English and shows the following information:

- Item:** Discussion | Read | View history | Search Wikidata
- Instance of:** ransomware cyberattack (with edit and reference options)
- Image:** Countries initially affected in WannaCry ransomware attack.svg (1,104 x 566; 1.46 MB) (with edit, reference, and value options)
- Based on:** WannaCry (with edit, reference, and value options)

The left sidebar contains navigation links such as 'Main page', 'Community portal', 'Project chat', 'Create a new item', 'Recent changes', 'Random item', 'Query Service', 'Nearby', 'Help', 'Donate', 'Lexicographical data', 'Tools', 'What links here', 'Related changes', 'Special pages', 'Permanent link', 'Page information', 'Concept URI', and 'Cite this page'.

Cybersecurity Instance

- Wikidata has data on many specific cybersecurity instances
- A cybersecurity instance could be a specific instance of a virus
- It can have attributes including:
 - Name, aliases & short description
 - Discovery date
 - Operating systems affected
 - Links to other public knowledge graphs with more information

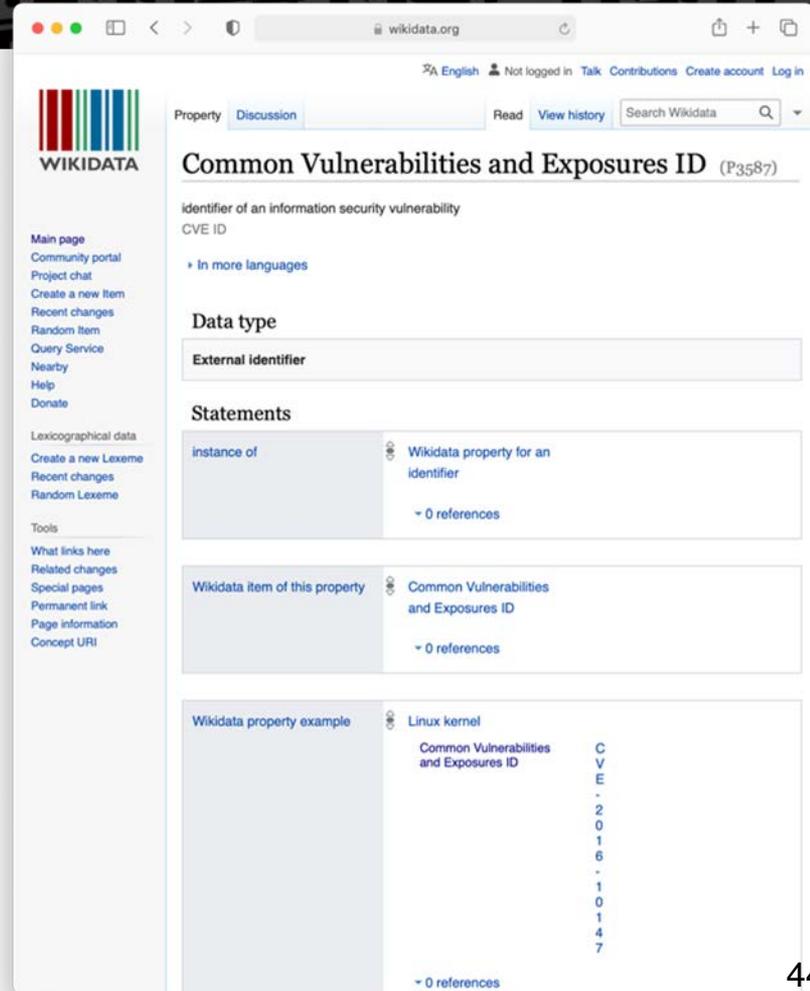


The screenshot shows the Wikidata page for the instance 'WannaCry' (Q29908721). The page is in English and shows the Wikidata logo, navigation links, and a list of statements. The statements include:

- instance of: ransomware
- name: Hokusey (English)
- based on: EternalBlue, DoublePulsar
- publication date: 12 May 2017

Cybersecurity Property

- Many properties of cybersecurity items are general and some very specific
- Operating System ([P306](#)) is a general property that's useful in describing software
- CVE ID ([P3587](#)), on the other hand, is a property that only applies to CVE items



The screenshot shows the Wikidata page for the property **Common Vulnerabilities and Exposures ID** (P3587). The page includes a navigation bar with 'Property' and 'Discussion' tabs, a search bar, and a sidebar with various Wikidata navigation options. The main content area displays the property's description as 'identifier of an information security vulnerability' and 'CVE ID'. It also shows the 'Data type' as 'External identifier' and a 'Statements' section with three entries: 'instance of' (Wikidata property for an identifier), 'Wikidata item of this property' (Common Vulnerabilities and Exposures ID), and 'Wikidata property example' (Linux kernel). The Linux kernel example shows a list of CVE IDs: CVE-2016-10147.

Wikidata's Cyber knowledge

We are exploring Wikidata's cyber-security knowledge via:

- Its web interface for people
- SPARQL queries to retrieve and analyze its cybersecurity data
- Using other tools like [wdtaxonomy](#) and graph visualization systems

```

~> wdtaxonomy Q485
computer virus (Q485) •103 ×81 ↑↑
├── boot sector virus (Q893210) •5 ×6
│   ├── Brain (Q158593) •24
│   ├── Natas (Q1037622) •6
│   ├── OneHalf (Q2025943) •6
│   ├── Ping-Pong virus (Q3905195) •5
│   ├── Alcon (Q4713339) •1
│   ├── AntiCMOS (Q4774499) •2
│   ├── Hare (Q5656691) •1
│   └── Ripper (Q10655706) •1
├── Macro virus (Q947369) •23 ×2
├── polymorphic code (Q950981) •17
├── boot sector virus (Q2174007) •1 ×1
├── AIDS (Q3600969) •8
├── A and A (Q3848426) •5
├── QH.EXE (Q3926208) •1
├── stealth virus (Q4441598) •4
├── iframe virus (Q5991258) •2
├── ??? (Q6163847) •1 ↑
├── Wiper (Q8026703) •4 ×1
├── multipartite virus (Q9427662) •1
│   └── Ghostball (Q16978760) •2
├── resident virus (Q19715216) •1
└── HyperCard virus (Q106253790) •1
    
```

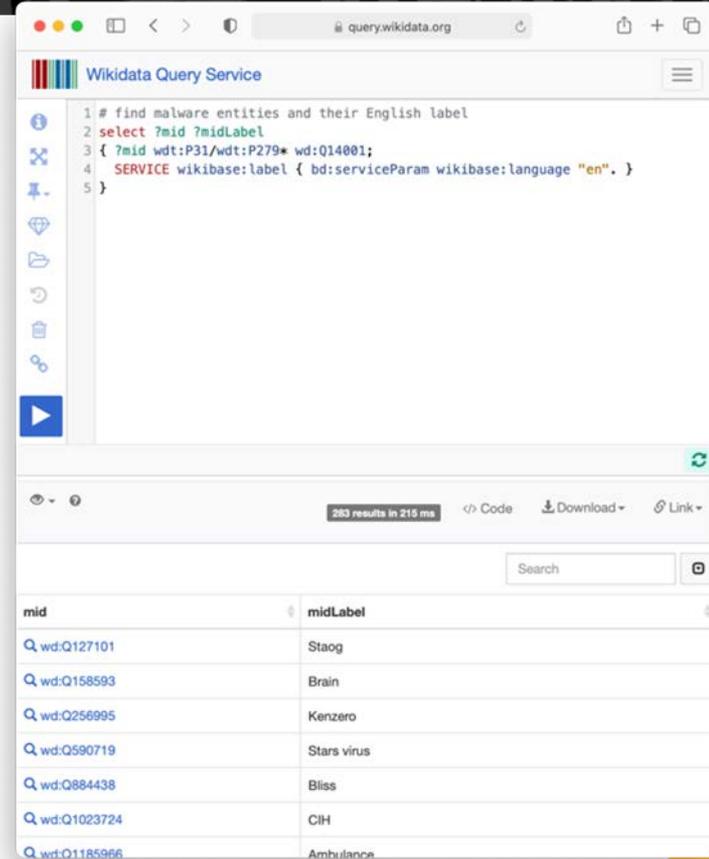
The **wdtaxonomy** command-line tool extracts data on WD concepts

Simple SPARQL query

- We query Wikidata with the SPARQL query language
- This query finds all 283 instances of malware ([Q14001](#)) and its sub-classes along with their names

```
select ?qid ?qidLabel {
  ?qid wdt:P31/wdt:P279* wd:Q14001;
  SERVICE wikibase:label
    {bd:serviceParam wikibase:language "en".}}
```

- The resulting triples can be easily ingested into a program or system



The screenshot shows the Wikidata Query Service interface. The query is:


```
1 # find malware entities and their English label
2 select ?mid ?midLabel
3 { ?mid wdt:P31/wdt:P279* wd:Q14001;
4   SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
5 }
```

 The results table shows:

| mid | midLabel |
|----------|-------------|
| Q127101 | Staog |
| Q158593 | Brain |
| Q256995 | Kenzero |
| Q590719 | Stars virus |
| Q884438 | Bliss |
| Q1023724 | CIH |
| Q1185966 | Ambulance |

Simple SPARQL query anatomy

variables whose values to return

```
select ?qid ?qidLabel {
```

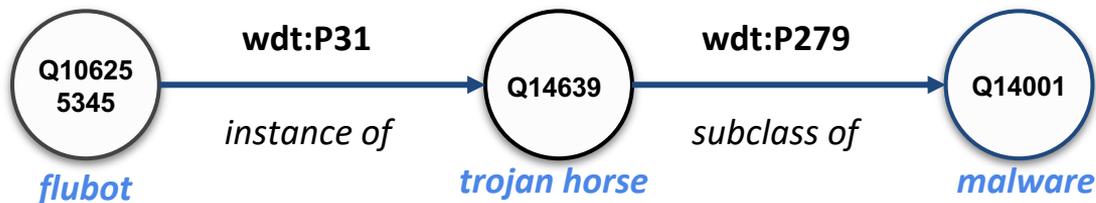
```
?qid wdt:P31/wdt:P279* wd:Q14001 ;
```

```
SERVICE wikibase:label
{bd:serviceParam wikibase:language "en"}}
```

a path from ?qid consisting of a "instance of" (P31) followed by any number of "subclass of" (P279) edges ending at "malware" (Q14001)*

Languages we want any labels to be in

One match:



More complex SPARQL query

```
select ?qid ?name ?desc ?year
  (group_concat(DISTINCT ?alias; separator="|") as ?aliases)
  (group_concat(DISTINCT ?ava;separator="|") as ?avAliases)
  (group_concat(DISTINCT ?os;separator="|") as ?opSys)
{?qid wdt:P31/wdt:P279* wd:Q14001.
  optional {?qid rdfs:label ?name filter(lang(?name)='en')}
  optional {?qid schema:description ?desc filter(lang(?desc)='en')}
  optional {?qid skos:altLabel ?alias filter(lang(?alias)='en')}
  optional {?qid wdt:P1845 ?ava.} # vendor antiVirusLabel
  optional {?qid wdt:P306/rdfs:label ?os filter(lang(?os)='en')}
  optional {?qid wdt:P575 ?date. BIND (year(?date) as ?year) }
}
GROUP BY ?qid ?name ?desc ?year
```

- query finds seven key properties of all malware instances
- Asking only for English language text values
- use of optional means data not required (not all malware has an alias)
- Queries can be entered in website or sent via an API to WD

More complex

```

select ?qid ?name ?desc ?year
  (group_concat(DISTINCT ?alias; separator="|")
  (group_concat(DISTINCT ?ava;separator="|")
  (group_concat(DISTINCT ?os;separator="|")
{?qid wdt:P31/wdt:P279* wd:Q14001.
  optional {?qid rdfs:label ?name filter(lang(?
  optional {?qid schema:description ?desc filt
  optional {?qid skos:altLabel ?alias filter(lang
  optional {?qid wdt:P1845 ?ava.} # vendor a
  optional {?qid wdt:P306/rdfs:label ?os filter
  optional {?qid wdt:P575 ?date. BIND (year(
}

```

```
GROUP BY ?qid ?name ?desc ?year
```

```

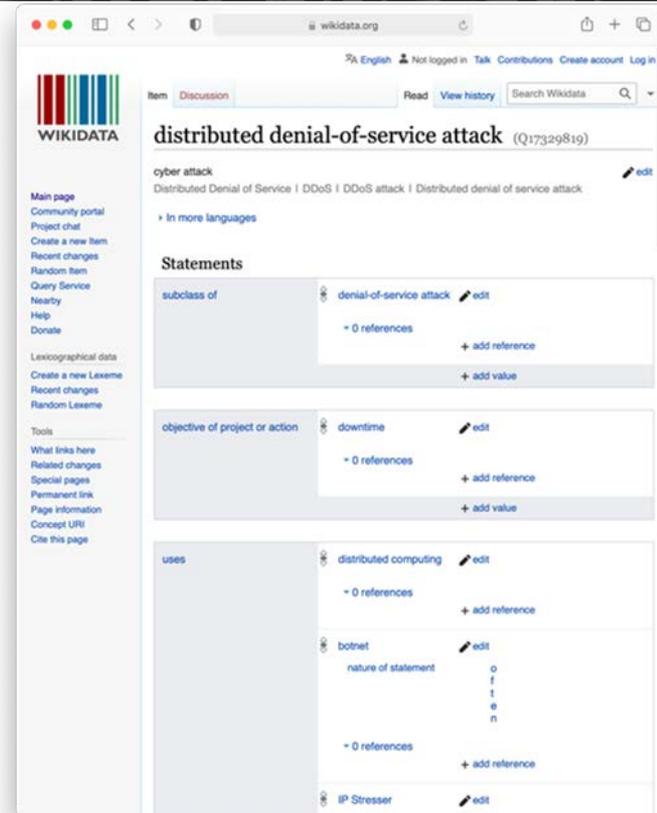
[
  {
    "qid": "http://www.wikidata.org/entity/Q106255345",
    "name": "FluBot",
    "desc": "Android malware that is propagated using
             fake SMS messages",
    "year": "2020",
    "aliases": "Cabassous",
    "avAliases":
      [
        "Trojan-Banker.AndroidOS.Flubot",
        "Trojan:Script/Wacatac.B!ml"
      ],
    "opSys": "Android"
  },
  ...
]

```

Query results available as triples, TSV, JSON, ...

Linking entities to Wikidata items

- Prototype system links text strings to Wikidata items
 - “ddos” => [Q17329819](https://www.wikidata.org/wiki/Q17329819): distributed denial-of-service attack
- Linking uses spaCy’s [word embeddings](#) and WD’s fine-grained type system
- Improves graph by merging nodes that refer to same concept, entity, or event:
 - DDoS, distributed denial of service, DDoS attack, ...



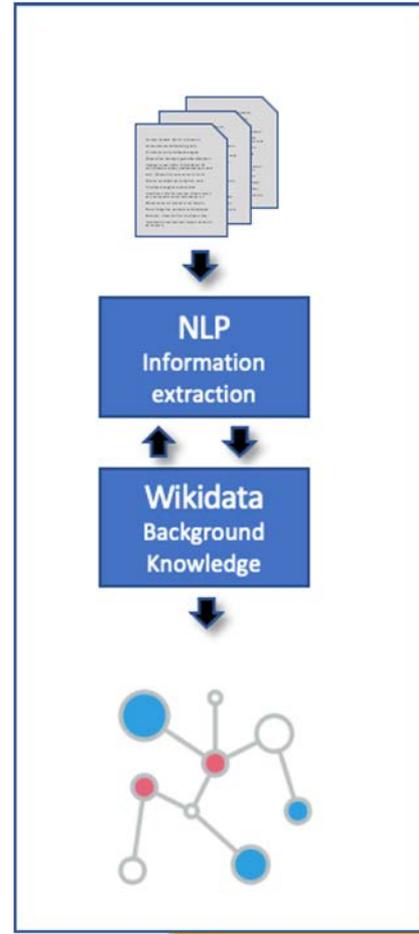
Wikidata Next Steps



- Catalog and describe what Wikidata knows about cybersecurity
- Create Wikidata bots to identify and tag likely omissions and errors
- Improve our Wikidata entity linker for cybersecurity concepts, properties, and entities
- Explore ways to (semi-) automatically update Wikidata with new cybersecurity information

Conclusion and Next Steps

- We are working toward a system maintains a **knowledge graph** of **cybersecurity data** extracted from public **text reports**
- We are using the spaCy NLP tools and the Wikidata knowledge graph
- In the future we hope to extract and integrate data from semi-structured data in [STIX](#), [MISP](#) and other formats





questions and
discussion