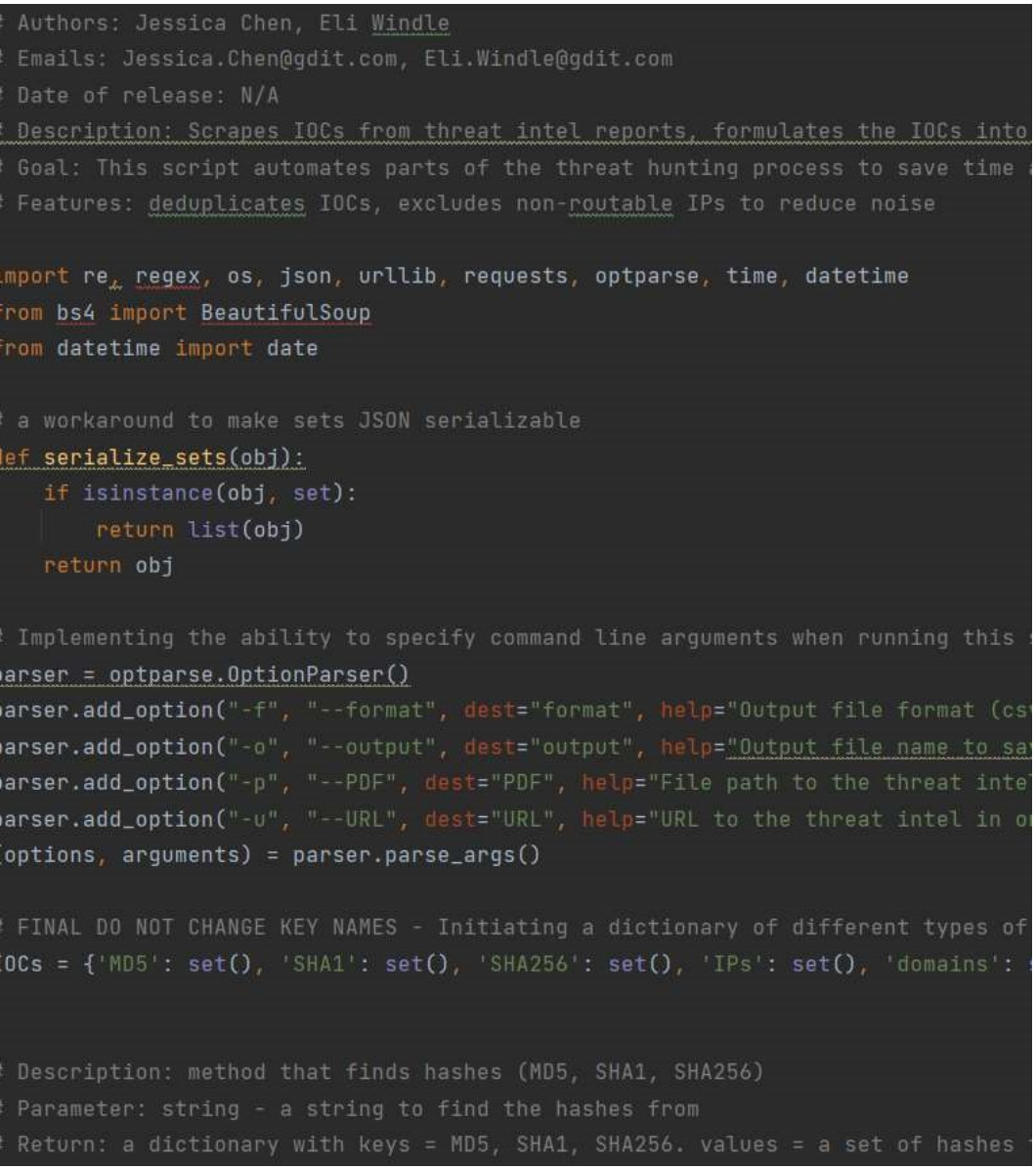
Threat Hunting Automation

Jessica Chen

GENERAL DYNAMICS
Information Technology



```
# Authors: Jessica Chen, Eli Windle
# Emails: Jessica.Chen@gdit.com, Eli.Windle@gdit.com
# Date of release: N/A
# Description: Scrapes IOCs from threat intel reports, formulates the IOCs into
# Goal: This script automates parts of the threat hunting process to save time
# Features: deduplicates IOCs, excludes non-routable IPs to reduce noise

import re, regex, os, json, urllib, requests, optparse, time, datetime
from bs4 import BeautifulSoup
from datetime import date

# a workaround to make sets JSON serializable
def serialize_sets(obj):
    if isinstance(obj, set):
        return list(obj)
    return obj

# Implementing the ability to specify command line arguments when running this
parser = optparse.OptionParser()
parser.add_option("-f", "--format", dest="format", help="Output file format (cs
parser.add_option("-o", "--output", dest="output", help="Output file name to sa
parser.add_option("-p", "--PDF", dest="PDF", help="File path to the threat inte
parser.add_option("-u", "--URL", dest="URL", help="URL to the threat intel in o
(options, arguments) = parser.parse_args()

# FINAL DO NOT CHANGE KEY NAMES - Initiating a dictionary of different types of
IOCs = {'MD5': set(), 'SHA1': set(), 'SHA256': set(), 'IPs': set(), 'domains':


# Description: method that finds hashes (MD5, SHA1, SHA256)
# Parameter: string - a string to find the hashes from
# Return: a dictionary with keys = MD5, SHA1, SHA256. values = a set of hashes
```

Project Background

- To hunt for threats, cyber analysts need to go through threat intel reports which include many Indicators of Compromise (IOC)
  - IOCs are forensic artifacts that identify malicious activity
  - Examples: IPs, file hashes, URLs
- Analysts had to manually extract these IOCs, formulate them into queries, and run the queries on company security tools
- This manual process is time-consuming, it should be automated
- Tools used: Python, Regex, Beautiful Soup, APIs

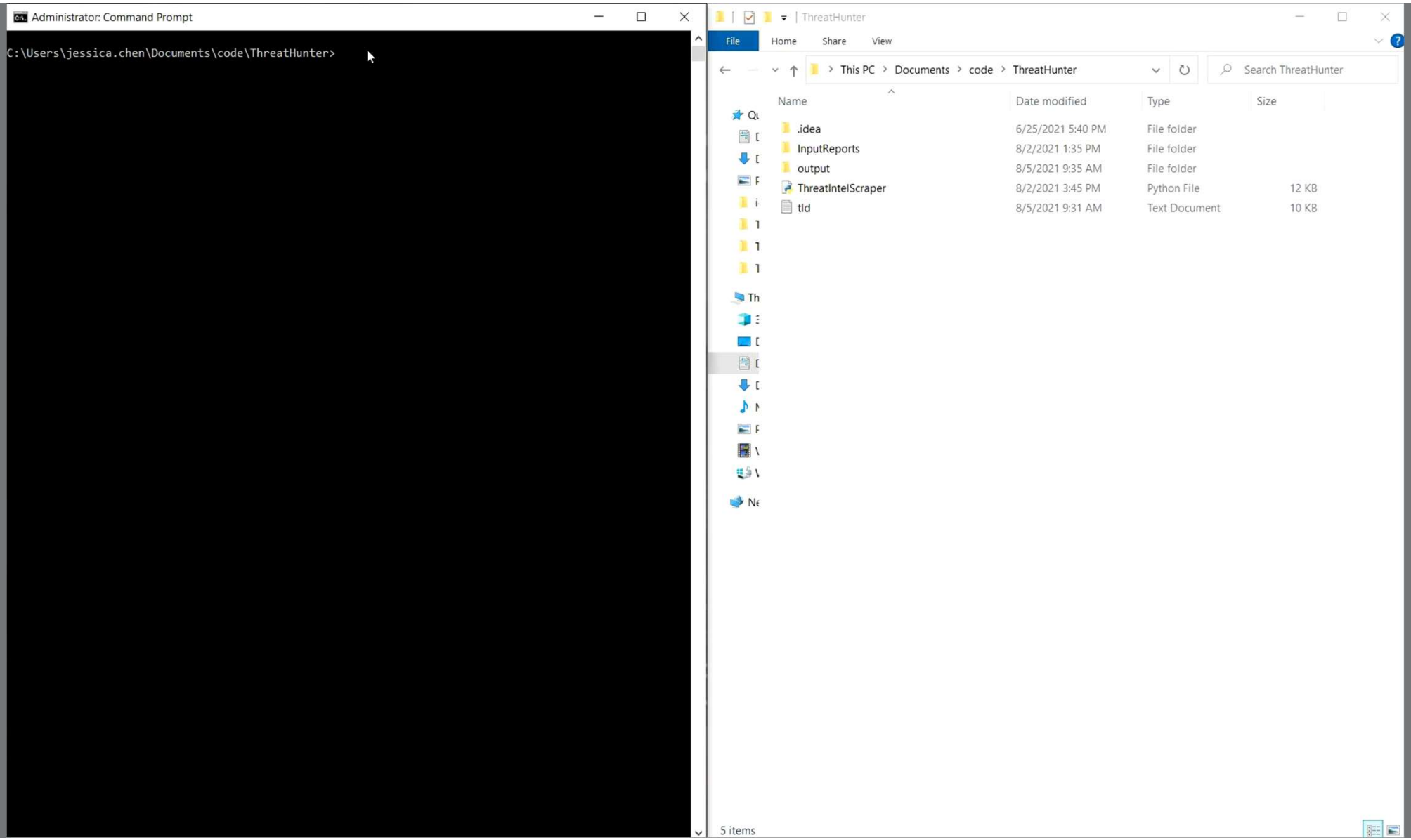
Input

- Threat intelligence report
- Accepted format: PDF and URL

Output

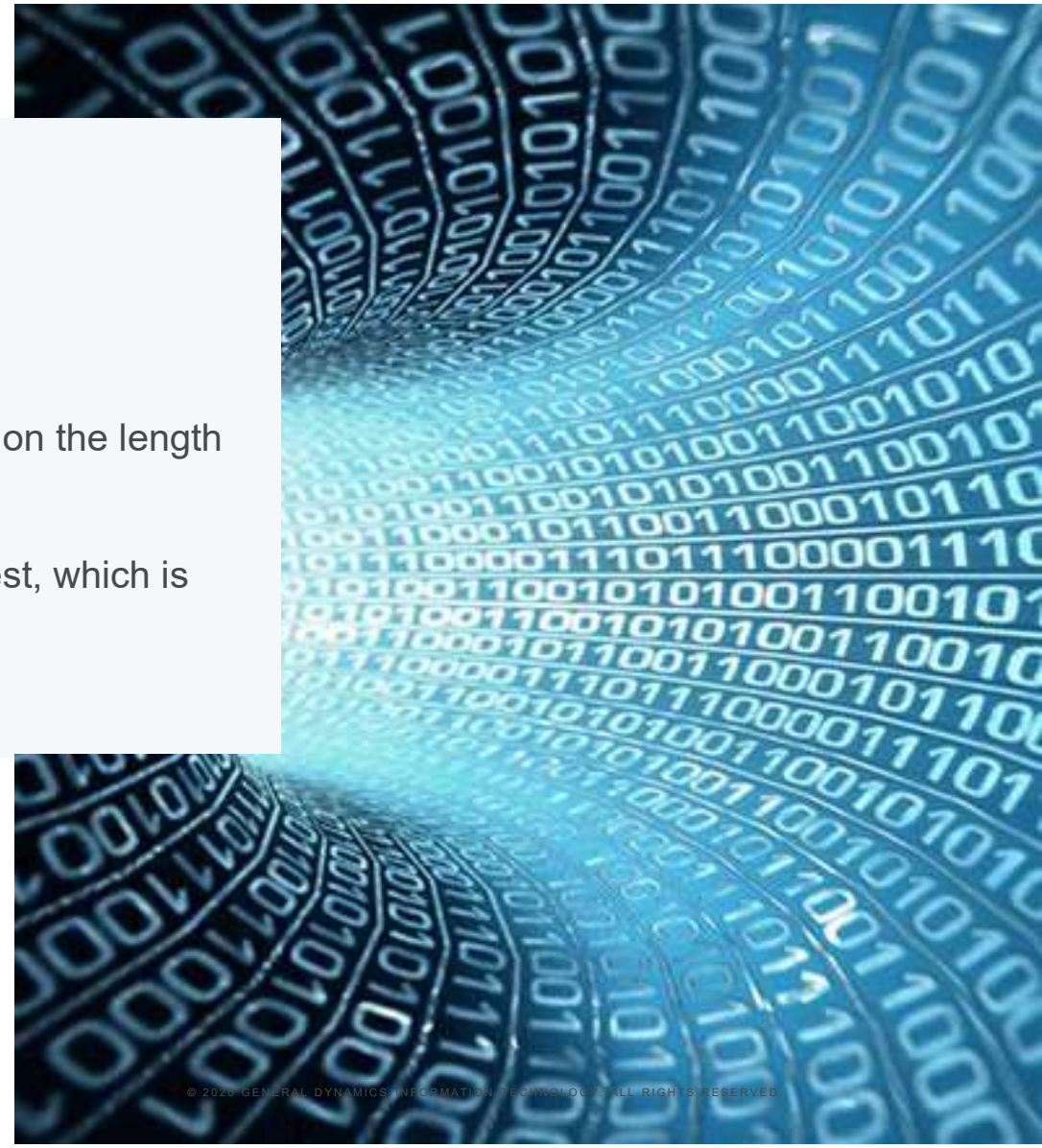
- A list of IOCs in JSON or CSV format
- A text file of the queries
- A JSON file of the detection results

GDIT

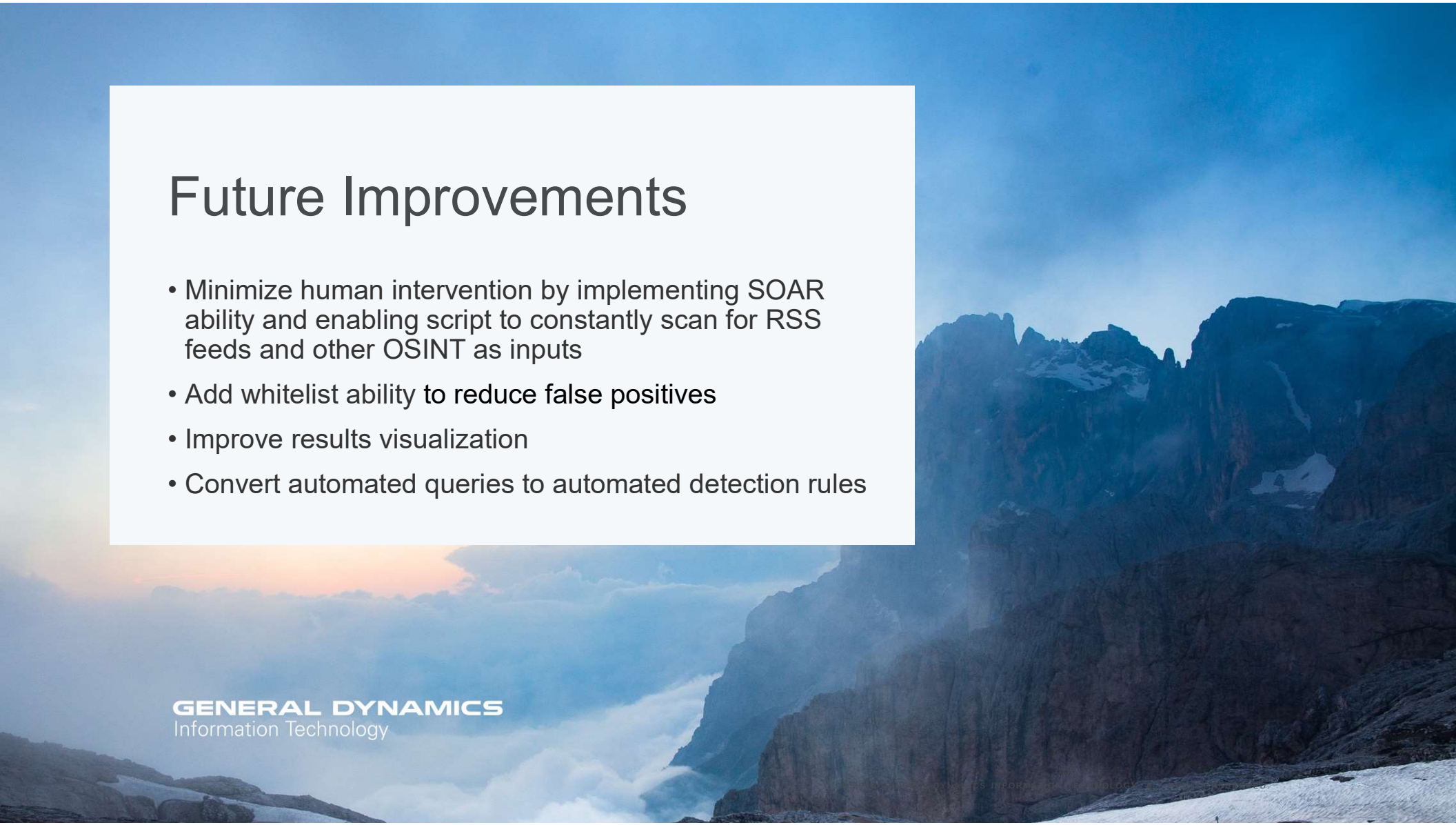

Administrator: Command Prompt
C:\Users\jessica.chen\Documents\code\ThreatHunter>
ThreatHunter
File
Home
Share
View
This PC
Documents
code
ThreatHunter
Search ThreatHunter
Name
Date modified
Type
Size
.idea
6/25/2021 5:40 PM
File folder
InputReports
8/2/2021 1:35 PM
File folder
output
8/5/2021 9:35 AM
File folder
ThreatIntelScraper
8/2/2021 3:45 PM
Python File
12 KB
tld
8/5/2021 9:31 AM
Text Document
10 KB
5 items

Impact

- Saves many hours of work time depending on the length and amount of threat intel reports ingested
- Minimize human error and security risk
- Analysts can now focus on what they do best, which is analyze



GENERAL DYNAMICS
Information Technology



Future Improvements

- Minimize human intervention by implementing SOAR ability and enabling script to constantly scan for RSS feeds and other OSINT as inputs
- Add whitelist ability to reduce false positives
- Improve results visualization
- Convert automated queries to automated detection rules

GENERAL DYNAMICS
Information Technology