

Passive Intelligence Gathering and Analytics - It's all Just Metadata!

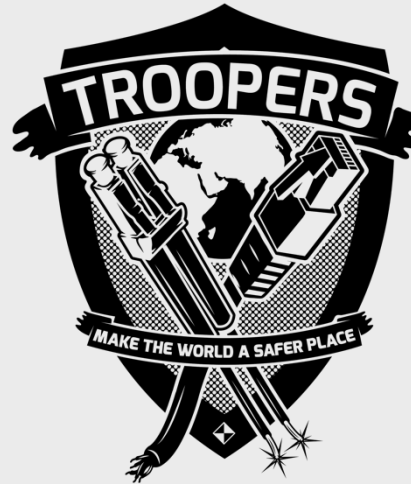
@ChrisTruncer



TROOPERS

Whoami

- Christopher Truncer ([@ChrisTruncer](#))
 - Mandiant's Red Team
 - Florida State Seminole
 - Open Source Developer
 - Veil Framework
 - Egress-Assess
 - EyeWitness, etc



What's this talk about?

- Understanding Our Threats
- Threat Intelligence
- Just-Metadata
- Intel Gathering and Analytics
- Intel/Analytics Module Development
- Tempt the Demo Gods



Understanding Our Threats

Who Are Our Threats?

- Identify - who/what are our threats?
 - What are their motivations?
 - How do they commonly operate?
 - Are they a “sophisticated threat”?
 - What do we know about them?



First Steps

- You may not know everyone who is targeting your organization
- Put yourself in a position to gather as much information as possible
 - Log, log, log
 - Actually read/use the logs
 - Parse for useful info



What Should You Get?

- What should you be looking for?
 - Anything that stands out to **you**
 - Web pages/resources only employees view (OWA, employee login)?
 - Resources on non-standard ports being discovered?



Other Data to Gather

- Company being targeted by malware
 - Perform static and/or dynamic analysis on malware
 - (Not using VirusTotal right?)
 - Let's not tip off our attackers
 - Identify callback domains/IPs
 - Determine protocol(s) used



What's Next?

- Now that you have data, what next?
 - **Don't tip off the attackers**
 - Start investigating what could be the recon phase for an attacker
- Can passive intelligence help?



Passive Intelligence Gathering

From an ethical hacking perspective, the focus is upon identifying information about the organisation under investigation, without the organisation being aware that the information has been accessed.

<http://www.technicalinfo.net/papers/PassiveInfoPart1.html>



Passive Intelligence

(Not all Inclusive)

Threat Intelligence

Threat Feeds

Norse
(RIP)

SecureWorks

etc.

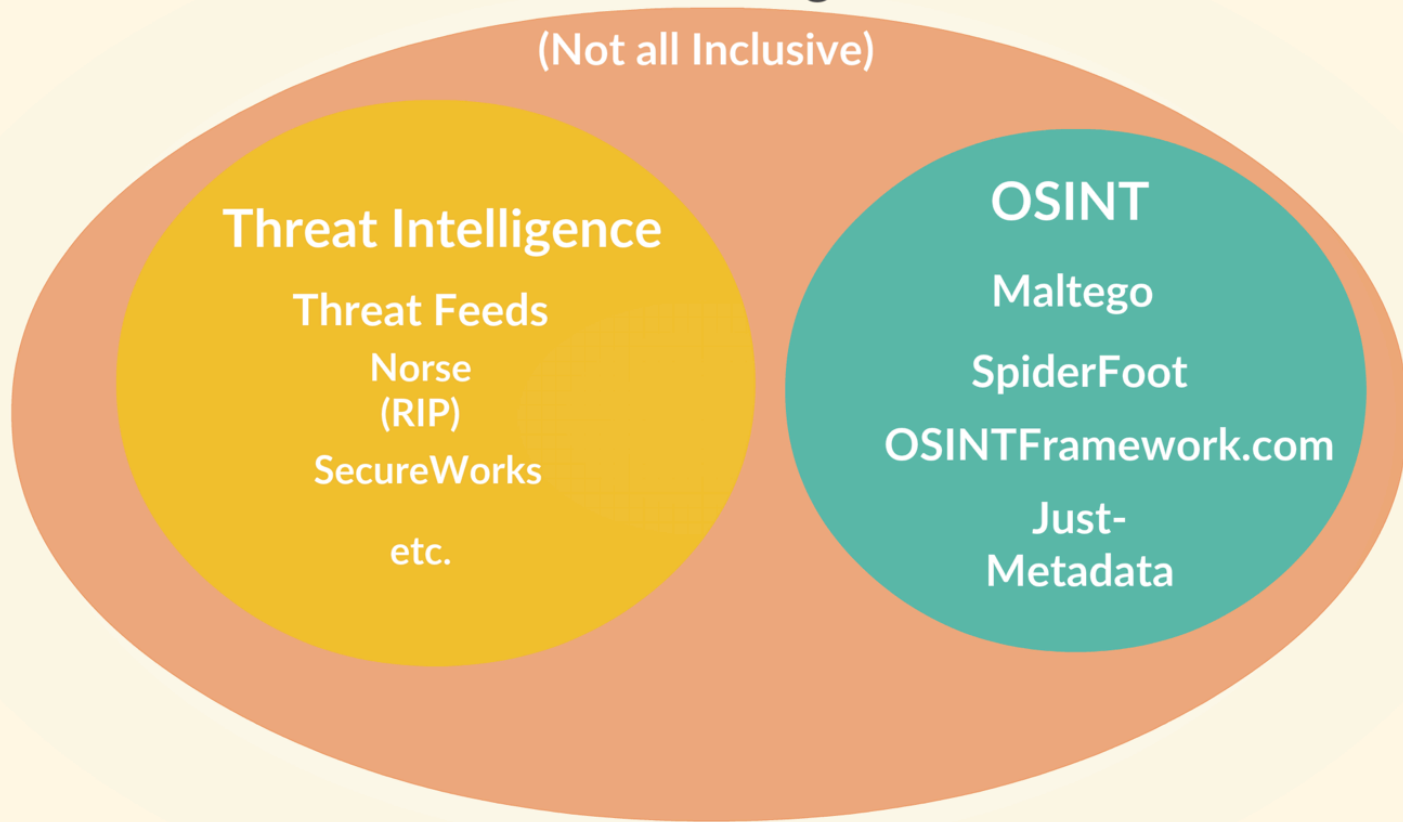
OSINT

Maltego

SpiderFoot

OSINTFramework.com

Just-
Metadata



Threat Intelligence is..

[It is] evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard.¹



Threat Intelligence is..

Amidst the commercialization and fear and the threat-intel-buying frenzy, it is easy to overlook the true meaning of threat intelligence. Threat intelligence exists to help us make decisions about how to best protect assets with limited time, money, and personnel. Knowing what is likely to affect you - how, why, what to look for, and what you can do about it - and then taking actions to mitigate those threats is what threat intelligence is all about.



What Information is Useful

- Who do attackers normally target?
 - Industries
 - Countries
- What type of data are they targeting?
 - PII? Intellectual Property?
 - Defense Data?
- Known TTPs? Motivations?



**This information would be
great to have**

Threat Intel Issues

- Feeds can be expensive
- May not contain threats relevant to your organization
- Rely on analysis of others
- Can attackers subscribe?
- Can you justify the cost through the information you receive?



Is there another option?

Open Source Intelligence Gathering

- Historically thought of by gathering freely available data manually:
 - Newspaper articles
 - Magazines
- Since changed to include electronic mediums
 - blogs, wikis, mailing lists
 - “collectors”



Open Source Intelligence Gathering

- Good alternative to traditional active information gathering techniques
- OSINT has been around for years
 - Internet & Instant communications can enhance its effectiveness
- “...the chief difficulty is in identifying relevant, reliable sources from... publicly available info...”



Open Source Intelligence Gathering

- Pros:
 - Can be very low cost, if not free
 - Don't alert target(s) of investigation
- Cons:
 - Info as good as the source
 - You aren't generating the intel



Starting Point?

- Build a list of IPs or domains
 - You want to know more about these
 - Accessing sensitive resources
 - Callback domains?
 - Something custom to your environment
- You now have a starting point



Analyzing Data

- So you have a start, what's next?
 - Don't tip off the attackers
 - You don't need to use any tool within your network
 - "Blind" intel gathering
 - Look for relationships
 - How can we do this?



Just-Metadata

```
#####  
#                               Just-Metadata                               #  
#####  
  
ip_info => Display's all info about an IP address  
save     => Saves IPs and attributes to disk for reloading in the future  
gather   => Requests information and gathers statistics on loaded IP addresses  
export    => Exports all data on all IPs to CSV  
exit      => Exits out of Just-Metadata  
import    => Import's saved state into Just Metadata  
analyze   => Run [module] on the loaded IP addresses  
help      => Displays commands and command descriptions  
load      => Loads IPs into the framework for analysis  
list      => Prints loaded [analysis] or [gather] modules  
  
[>] Please enter a command:
```

Just-Metadata Design Goals

- Fast & easy setup
- Minimal configurations
- Lightweight with minimal dependencies
- Automate the mundane processes



Just-Metadata Usage Goals

- Analysis of large datasets
 - Identify useful/interesting info
 - Enrich metadata
- Agile/Easily customizable framework
 - Utilizes a modular framework
- Ability to save/load current work state



Just-Metadata Usage

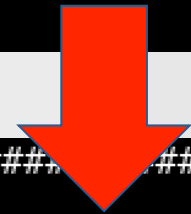
- Feed Just-Metadata a file containing IPs/Domains you are interested in
- Start the automated intel gathering process
- Identify “interesting” relationships



Loading IPs

```
#####  
#                               Just-Metadata                               #  
#####  
  
[>] Please enter a command: load ips.txt
```

```
#####  
#                               Just-Metadata                               #  
#####  
[*] Loaded 171 IPs  
  
[>] Please enter a command: █
```



Usability Point

- Data is stored in memory for quick access, but obviously is volatile
- Needed the ability to save the current state of Just-Metadata
 - All IPs loaded into the framework
 - All data gathered also in framework



Usability Point

- With a saved state, user doesn't need to gather data again
- Python Pickles to the rescue!
 - IP Objects saved
 - Need to “stress test” how many can be loaded



Saving States

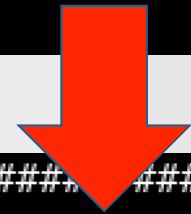
```
#####  
#                               #  
#                               Just-Metadata                               #  
#####  
  
[>] Please enter a command: save
```



```
#####  
#                               #  
#                               Just-Metadata                               #  
#####  
  
State saved to disk at metadata03132016_195247.state  
  
[>] Please enter a command: █
```

Importing States

```
#####  
#                               Just-Metadata                               #  
#####  
  
[>] Please enter a command: import metadata03072016_113830.state
```



```
#####  
#                               Just-Metadata                               #  
#####  
[*] Successfully imported metadata03072016_113830.state  
  
[>] Please enter a command:
```

Gathering Intel

Intel Gathering Reminder!

- Your intelligence is only as good as your source
 - Consider your sources when evaluating the credibility of information



Intel Gathering Goals

- Automate the intelligence gathering process
- Collect as much data as possible
- Keep data in a format that is easily parsable




Just-Metadata Intel Sources

- Network Whois (Python lib.)
- Geo-Location Information
- Shodan
 - Needs API Key
- VirusTotal
- Various Threat Feeds
 - Animus
 - Alienvault
 - etc.



```
#####  
#                               Just-Metadata                               #  
#####  
[*] Loaded 743 systems  
  
[>] Please enter a command: gather all
```

```
#####  
#                               Just-Metadata                               #  
#####  
Querying Shodan for information about 89.225.204.98  
Querying Shodan for information about 46.190.15.80  
Querying Shodan for information about 181.23.73.154  
Querying Shodan for information about 193.248.40.251  
Querying Shodan for information about 128.75.181.248  
Querying Shodan for information about 105.98.84.97  
Querying Shodan for information about 50.162.142.195  
Querying Shodan for information about 78.30.150.87  
Querying Shodan for information about 90.151.12.79
```



Analysis Modules

Analysis Modules

- At this point, we've collected a lot of data
 - But right now... it's just a bunch of data
- Extracting meaningful information from the larger dataset is what provides value



Analysis Modules

- Geo Info
- Countries
- Feed Hits
- Keys
- Port Search
- Top Ports
- VirusTotal Hits
- Whois CIDR



Geo Info

- Parses Geolocation Data - gives the top “X” results for:
 - Country
 - State
 - City
 - Zip Code
 - Timezones
 - ISP
 - GPS Coordinates



You selected "GeoInfo" module, how many i
Ex: 10

[>] Return the Top: 5

Top 5 Countries
(Country : Number of Occurances)

=====
United States : 148
Singapore : 1
United Kingdom : 1

Top 5 Cities
(City : Number of Occurances)

=====
Islandia, United States : 139
Ashburn, United States : 3
San Antonio, United States : 2
New York, United States : 2
Broadstairs, United Kingdom : 1

Top 5 Regions
(Region : Number of Occurances)

=====
New York, United States : 141
Virginia, United States : 3
Missouri, United States : 2
Texas, United States : 2
England, United Kingdom : 1

Top 5 GPS Coordinates
(GPS Coordinates : Number of Occurances)

=====
40.8064, -73.1738 : 139
39.0335, -77.4838 : 3
40.7128, -74.0059 : 2
29.4889, -98.3987 : 2
38.6312, -90.1922 : 1

Top 5 ZipCodes
(ZipCode : Number of Occurances)

=====
11749 : 139
20147 : 3
78218 : 2
63101 : 1
63017 : 1

Top 5 ISPs
(ISPs : Number of Occurances)

=====
Verizon Business : 2
AMAZON : 2
Rackspace Hosting : 2
Amazon.com : 1
Host Partners Ltd : 1

Port Modules

- Top Ports - Parses Shodan information to display the top X open ports across systems
- Port Search - Searches across all systems for a specific port



```
#####
#                               Just-Metadata                               #
#####
You selected the "Top_Ports" module, how many ports do you want returned?
Ex: 10

[>] Total: 5
*****
Top Ports : Number of Instances
*****
Port: 443 - 42 instances
Port: 80 - 35 instances
Port: 25 - 6 instances
Port: 22 - 4 instances
Port: 21 - 3 instances
```

```
#####  
#                               Just-Metadata                               #  
#####  
You selected the "Port_search" module, which port are you looking for?  
Ex: 80  
  
[>] Port: 22  
Port 22 is open on the following IPs:  
*****  
199.119.123.234  
141.202.253.191  
54.169.93.52  
141.202.253.128
```


Feed Hits

- Queries/Parses freely available threat feeds
- TOR Exit Nodes
- Animus Project
- EmergingThreats
- Alienvault
- Blocklist.de
- NoThinkMalware
- AntiSpam
- MalwareBytes
- etc...



```
#####  
#                               Just-Metadata                               #  
#####  
The following IPs are known TOR exit nodes:  
62.102.148.67  
171.25.193.131  
65.19.167.132  
178.32.53.154  
37.48.81.27  
  
No loaded IPs are in the Animus Project's attackers list!  
  
No loaded IPs are in the Emerging Threats list!  
  
The following IPs are in Alienvault's reputation list:  
171.25.193.131  
  
No loaded IPs are within Blocklist!  
  
No loaded IPs are within Dragon Research's SSH list!  
  
No loaded IPs are within Dragon Research's VNC list!  
  
No loaded IPs are within Openblock!  
  
No loaded IPs are within NoThink's Malware list!  
  
No loaded IPs are within NoThink's SSH list!  
  
No loaded IPs are within the Feodo list!  
  
No loaded IPs are on an AntiSpam list!  
  
No loaded IPs are within malc0de's list!  
  
No loaded IPs are within the MalwareBytes list!
```

Keys

- Parses Shodan information and identifies any system(s) with the same:
 - SSH Keys
 - HTTPS Certificates
 - Certificate Chains



Shared SSH Keys

AAAAB3NzaC1yc2EAAAADAQABAAQDsV7DIoM3hIVhx557Jr1LCgw9mzLCD/sD7G7X2v6q56343
3Yhon7uSk72JqYEqqHtoHzMz9DDVlxkzgXqHoGjNCdtSgPRs9zx0CyAntVz1r/rw2JnXU73iY71t
cBjRj8G11H6QU6Fc4qNEzeQiPaXM0Cx0ZWxJL8VWzE3607q8ZMJYBvIz/4wLhIV0ZZrxijJs741NA
DEKD5BhneqK+HU9nsPSb/mv/Kxq8GKYsX8UnEFZ7oE2LUWe4h+Zj+zImz+GX48fP0LINh9huJsV6
Fw5QTtIcunIW/eFk2E1hj/r3QswabECyPwliyE5i9GYjy1n0sbFIWJypD56o3R2RVB0=

SSH Key is shared across the following IPs:

208.167.254.99

108.61.13.43

208.167.254.94

AAAAB3NzaC1yc2EAAAADAQABAAQAg10rijj rKgn2VJBRw6xqjJrDkKR40T8sb1RH9toXgCP08ds
1/sj7URGI9FVm+EkKgaLQSwRPRLxh0H6p4P9FjnQwb+/vKBFQ4fobUZeVY0+6KZ+dUmGLPRFVbdg
Sgem/5DADM1slAhTpj rZXZPhhh1GB6dHJzxHnuxRCL/BRxRLrD8=

SSH Key is shared across the following IPs:

24.9.157.147

50.132.103.12

Shared HTTPS Certificates

-----BEGIN CERTIFICATE-----

MIIDaTCCA1GgAwIBAgIBATANBgkqhkiG9w0BAQsFAADB4MQswCQYDVQQGEwJUVzEQ
MA4GA1UECBMHSHNpbkNodTE0MAwGA1UEBxMFSHVlb3UxFjAUBgNVBAoTDURyYXlU
ZWsgQ29ycC4xGDAWBgNVBA5TD0RyYXlUZWsgU3VwcG9ydDEVMBMGAlUEAxMMVmln
b3IgaU91dGVyMB4XDTE0MDUxNDA2MDUwMFOxDTM5MDUxNDA2MDUwMFoweDELMAkG
A1UEBhMCMVFcxEADA0BgNVBAgTB0hzaW5DaHUxDjAMBgNVBAcTBUh1S291MRYwFAYD
VQKKEw1EcmF5VGVRlENvcnAuMRgwFgYDVQQLew9EcmF5VGVRlFN1cHBvcnQxFTAT
BgNVBAMTDGFZpZ29yIFJvdXRlcjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAJMUaTRB0d/DF4YwqS60XT9U0eZwaNsrZwprgRivBjX6e491Q/PUzVxVlGPH
5qJFIjXn0I19Vgi3dRHaicW3yxkevtQILa/yj30tFVqLsvMXQoJ22vo39EHkDo5T
uyZZ75xJuvSMCWkshVE5KT/fwY6t7lwz1yYK8oo53l9aUEGNZxuZh0HLWDq7VJqk
TvF2NqysSmHuLy1b/xMUUJWwoi0oAC4pzBX1v0VrWPxRgQ0kKQUJLzRjesjuJNcY
uRgif3vDGSsbRPSP0gai6CmSj rSrX0m/nRocjFfa60PbGbp0L0B4G+koPPV9rPko
5306L7P7sc1uQuuEu8QvbQ3MHMcCAwEAATANBgkqhkiG9w0BAQsFAA0CAQEAEG4n
m/P7y+jkA9c2flwRJCAx9eCK6l4V/VK3ZKLHsyhFd9buLM1/P+69VDpXfYo51oxZ
wqUi foanbcvTD1nLQReHCEs25hEubJGT/M5oqiDr7ubBgdgSETQjczavxwkUIe8v
j6z7Ad5H355g4bkKrBShCn0IrNRPm9i22NRyEurIv01Y5LdIXKEVw6gtdWGKdf0A
dJJyS2bUiD4u2jbl+0xBKQK0AtbsQryKrY+TeXBvjFEL1APBAzFZkC4jnu6fyG8g
HjaxcayxLxHNwjAphu4/nMNt/YCNmDrx95VikIn6W8LadtP1qfAD8YXCJ6uo1VuM
e+teFGQTJ0udsxQsTA==

-----END CERTIFICATE-----

HTTPS Certificate is shared across the following IPs:

117.7.237.121

62.172.132.130

```
91cxG7685C/b+LrTW+C05+Z5Yg4MotdqY3MxtfWoSKQ7CC2iXZDXtHwLTxFwMMS2
RJ17LJ3lXubvDGGqv+QqG+6EnriDfcFDzkSnE3ANkR/0yB0tg2DZ2HKocyQetawi
DsoXiWJYRBuriSUBAA/NxBti21G00w9RKpv0vHP8ds42pM3Z2Czqrpv1KrKQ0U11
GIo/ikGQI31bS/6kA1ibRrLDYGCD+H1Qqc7CoZDDu+8CL9IVV05EFdkKrqeKM+2x
LXY2JtwE65/3YR8V3Idv7kaWKK2hJn0KCacuBK0NvPi8BDAB
-----END CERTIFICATE-----
```

```
*****
```

Certificate Chain is shared across the following IPs:

```
141.202.253.34
141.202.248.144
141.202.248.11
141.202.253.122
141.202.253.167
141.202.248.206
208.232.182.27
141.202.246.124
206.155.98.33
54.165.49.247
199.119.123.234
141.202.253.128
141.202.246.55
141.202.248.7
141.202.248.14
141.202.246.126
141.202.246.242
141.202.248.208
```

VirusTotal

- Parses VirusTotal results and looks for the following information:
 - Shared detected samples
 - Each IP's total detected samples
 - Shared undetected samples
 - etc...



Detected Communicating URLs

http://www.miramar.com/

URL is shared across the following IPs:

141.202.253.195

http://sdc.ca.com/

URL is shared across the following IPs:

141.202.248.129

IPs and Total Detected Communicating URLs

141.202.248.129: 1 detected sample

141.202.253.195: 1 detected sample

Goal of Analysis Modules

Provide meaning to collected data





Module Development

Module Development

- Multiple “stock” intel gathering and analysis modules
 - Can't account for everything
 - I don't know what's important to your environment



Module Development

- Intel gathering and analysis modules are Python classes
- Drag and drop modules into their respective folders
 - Framework auto-detects modules
- Each IP has an IP Object
 - All data stored here



```
class IP_Information:

    def __init__(self, incoming_system):
        self.ip_address = ""
        self.domain_name = ""
        self.ip_country = ""
        self.ip_country_code = ""
        self.ip_city = ""
        self.ip_region_name = ""
        self.ip_region_code = ""
        self.ip_zipcode = ""
        self.ip_latitude = ""
        self.ip_longitude = ""
        self.ip_isp = ""
```

Intel Gathering Modules

- Two required methods
 - `__init__`
 - `cli_name`
 - `description`
 - `gather` method
 - Receives dict containing all IPs
 - Intel gathered and saved here



```
class IntelGather:

    def __init__(self):
        self.cli_name = "Whois"
        self.description = "This module gathers whois information"

    def gather(self, all_ips):

        for path, incoming_ip_obj in all_ips.iteritems():

            if incoming_ip_obj[0].ip_whois == "" and incoming_ip_obj[0].ip_ad

                try:

                    print "Gathering whois information about " + incoming_ip_
                    ip_whois = IPWhois(incoming_ip_obj[0].ip_address)
                    incoming_ip_obj[0].ip_whois = ip_whois.lookup()
                except IPDefinedError:
                    print helpers.color("[*] Error: Private IP address, skip")

        return
```



```
class IntelGather:

    def __init__(self):
        self.cli_name = "VirusTotal"
        self.description = "This module checks VirusTotal for hits on loaded"
        self.api_key = "49858c37eb67ff5a1d1f3785e7a9fc06462e097e3a3cfc8a5b2bf"
        self.api_url = 'https://www.virustotal.com/vtapi/v2/'

    def check_host(self, host):
        result = re.match("^[([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])")
        return result

# This collapse function came from @harmj0y, thanks for the help with
# it man
def collapse(self, var, tabs=0):
    result = ""

    if type(var) is dict:
        for field, value in var.iteritems():
            result += "\n" + tabs * "\t" + field + ": " + self.collapse(
                value, tabs=(tabs + 1))

    elif type(var) is list:
        for l in var:
            result += self.collapse(l, tabs=tabs) + "\n"

    else:
        result += str(var)
    return result
```

Analysis Modules

- Two required methods
 - `__init__`
 - `cli_name`
 - `description`
 - `analyze`
 - Received IP objects
 - Parsing and output done here



```

class Analytics:

    def __init__(self, cli_options):
        self.cli_name = "MyWOTDomains"
        self.description = "Parse mywot domain reputation results"
        if cli_options is None:
            self.top_number = ''
        else:
            self.top_number = int(cli_options.analyze_number)

    def analyze(self, all_ip_objects):

        if self.top_number == '':
            print "You selected \"MyWOTDomains\" module, how many domains do"
            print "Ex: 10"
            self.top_number = int(raw_input(' \n\n[>] Return the Top: ').str

        # Create dicts for site rankings
        negative_sites = {}
        questionable_sites = {}
        neutral_sites = {}
        positive_sites = {}
        trustworthiness_rating = {}
        childsafty_rating = {}

        # Looping over IP address objects
        for value in all_ip_objects.itervalues():

            # Check to make sure there is a result in mywot attribute

```

Use Cases

- Gathering information on:
 - Systems scanning you
 - Malware callback Ip
- OSINT gathering/analysis on assessment target(s)
- Many other use cases
 - Systems scanning you
 - Malware callback IPs



Future Work

- Additional intel sources
- Additional analytical modules
 - What else is useful to you?
- Time comparisons?
- Beyond IPs/Domains
 - MD5s?





- Github

- <https://github.com/ChrisTruncer/Just-Metadata>

- Chris Truncer

- [@ChrisTruncer](#)
- CTruncer@christophertruncer.com