Adversary Infrastructure Tracking with Mihari

Manabu Niseki (@ninoseki)

Recon Village @DEFCON 29

Who am I?

- @ninoseki on GitHub and Twitter
  - https://github.com/ninoseki
  - https://twitter.com/ninoseki
- CSIRT / Trust & Safety engineer and analyst
- Interested in OSINT, CTI and making things

Agenda

- Fundamentals
- How to automate tracking with Mihari

Fundamentals

Good Friends With Bad Habits

- Adversaries are (sometimes) good friends with bad habits
  - Reusing source codes / components
  - Reusing infrastructures
    - IP address
    - SSL certificate
    - SSH host key
    - Whois registrant
    - etc.
- Reusing something increases a possibility of tracking

Threat Hunting Loop

Static/Dynamic Analysis

- C2 addresses
- C2 communication protocols
- Characteristic features

Adversary Infrastructure Tracking

- New C2 addresses
- New landing pages
- New samples

YARA Hunting

- New samples
- New variants

Threat Hunting Loop

Static/Dynamic Analysis

- C2 addresses
- C2 communication protocols
- Characteristic features

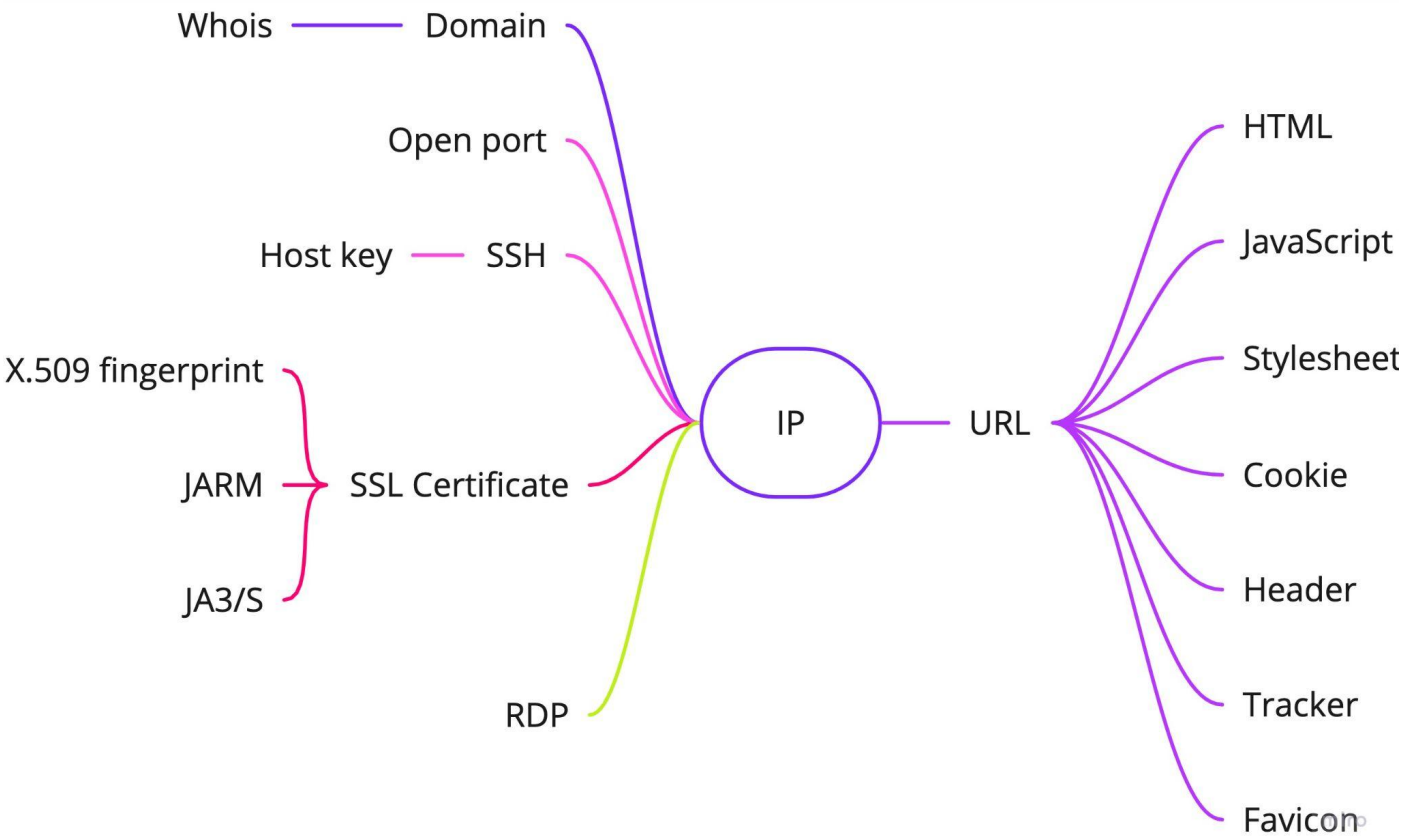
Adversary Infrastructure Tracking

- New C2 addresses
- New landing pages
- New samples

YARA Hunting

- New samples
- New variants

Fingerprints on the Internet



Active Tracking (Scanning)

Name	Description / URL
Nmap	A utility for network discovery and security auditing https://nmap.org/
Masscan	An Internet-scale port scanner https://github.com/robertdavidgraham/masscan
ZMap	A fast single packet network scanner https://github.com/zmap/zmap
httpx	A fast and multi-purpose HTTP toolkit https://github.com/projectdiscovery/httpx
JARM	An active Transport Layer Security (TLS) server fingerprinting tool. https://github.com/salesforce/jarm

Passive Tracking

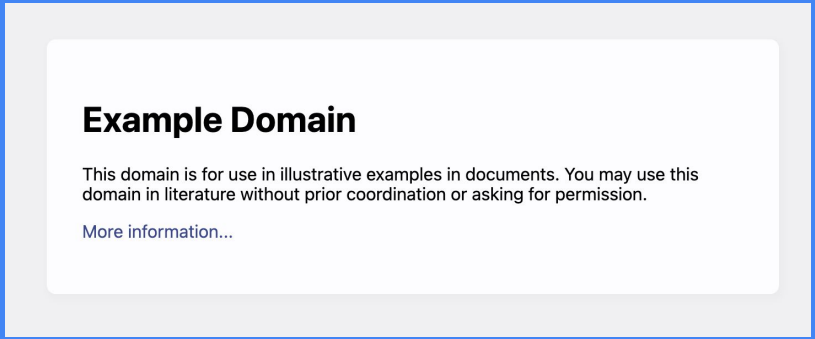
The Internet-wide scanning services	Passive DNS services	CT logs
Shodan	VirusTotal	crt.sh
Censys	PassiveTotal (RiskIQ)	Censys
BinaryEdge	SecurityTrails	
Onyphe	DomainTools	
Spyse	etc.	
ZoomEye		

Active Tracking vs. Passive Tracking

- Active tracking:
  - 👍 Can find active targets
  - 👎 Consumes a large number of computing resources including network bandwidth
- Passive tracking:
  - 👍 No needed to have your own infrastructure for tracking (but you should pay a fee to use)
  - 👎 May find inactive targets

An example of fingerprints

https://example.com



Hunting example.com

Target	Type	Value
HTML	MMH3	-2087618365
	MD5	84238dfc8092e5d9c0dac8ef93371a07
	SHA1	4a3ce8ee11e091dd7923f4d8c6e5b5e41ec7c047
	SHA256	ea8fac7c65fb589b0d53560f5251f74f9e9b243478dcb6b3ea79b5e36449c8d9
X.509	Serial	209251325845834064044156245034338833 37
	SHA256	200dcafa767c8450ece644879c062a0cdf52240fe05bb7eb284611c3aef3ec2e
DNS	A	93.184.216.34
	AAAA	2606:2800:220:1:248:1893:25c8:1946

Hunting example.com

Service	Target	Query
Shodan	HTML	http.html_hash:-2087618365 (Note: MMH3)
	X.509	ssl.cert.serial:209251325845834064044156245034338833 37
Censys	HTML	services.http.response.body_hash:"sha1:4a3ce8ee11e091dd7923f4d8c6e5b5e41ec7c047"
	X.509	services.certificate: 200dcafa767c8450ece644879c062a0cdf52240fe05bb7eb284611c3aef3ec2e
VT	DNS	https://www.virustotal.com/gui/ip-address/93.184.216.34
ST		https://securitytrails.com/list/ip/93.184.216.34
PT		https://community.riskiq.com/search/93.184.216.34

Tips

- mmhdan:
  - An app to calculate basic fingerprints of a website
    - HTML hash, favicon hash, X.509 fingerprint
    - Whois
    - DNS records
    - Tracker IDs
  - Repo: https://github.com/ninoseki/mmhdan
  - Demo: https://mmhdan.herokuapp.com/

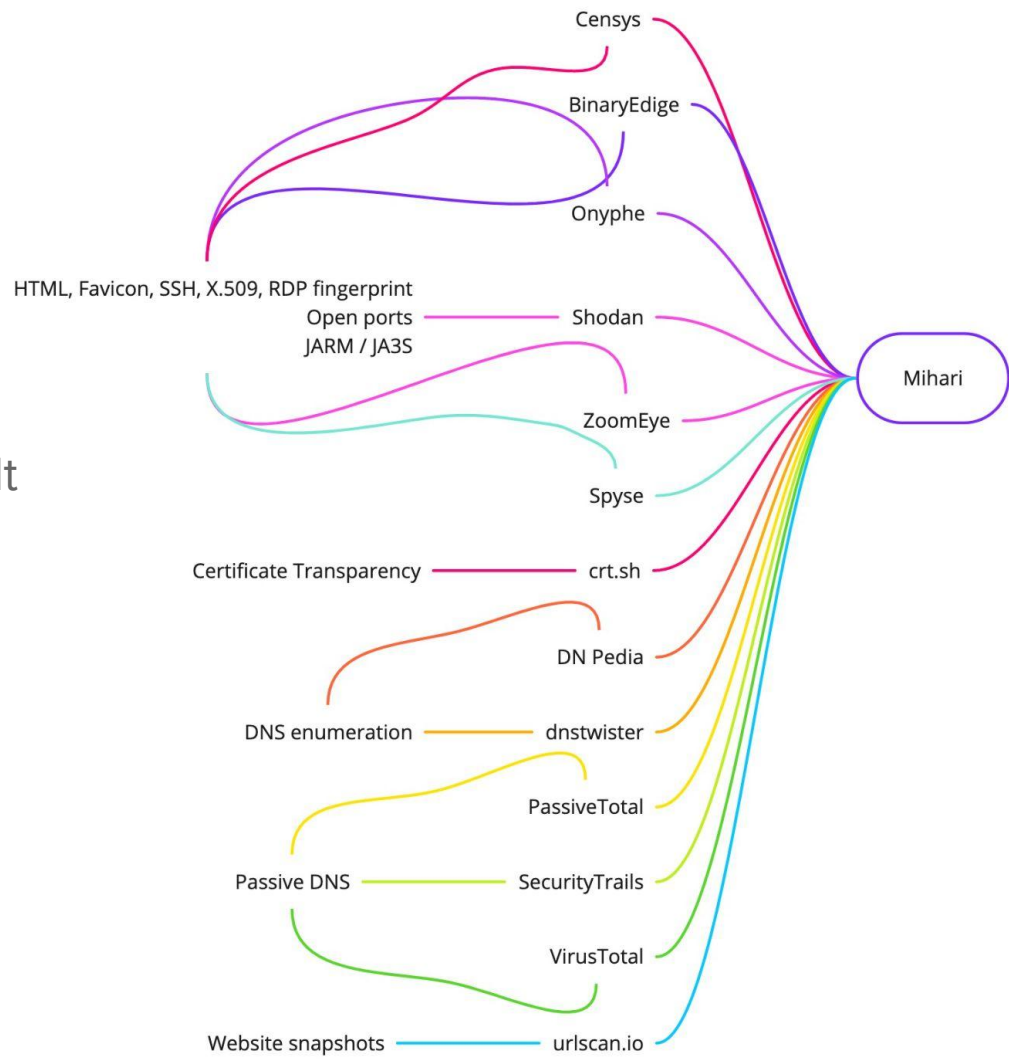
How to automate tracking with Mihari

What is Mihari

- Mihari:
  - https://github.com/ninoseki/mihari
  - A framework for continuous OSINT based threat hunting
  - Written in Ruby & packaged as a Ruby gem
    - $ gem install mihari
  - Note:
    - Mihari(見張) means “lookout” or “guard” in Japanese

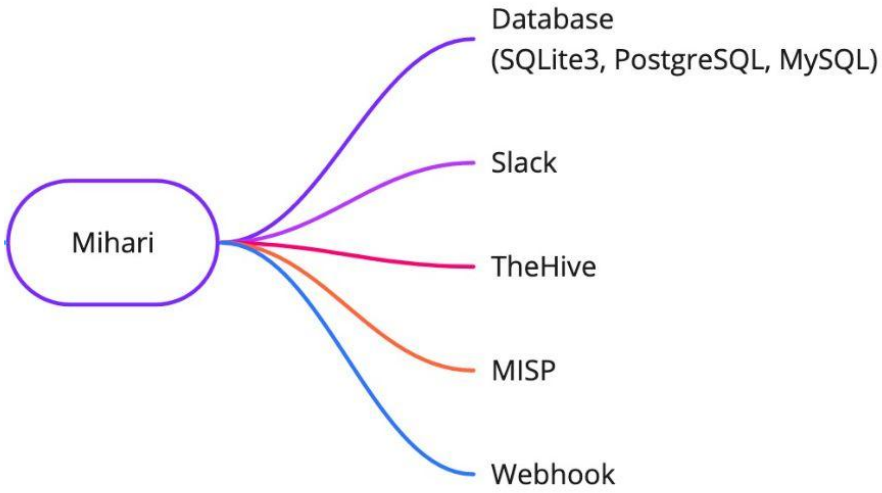
Mihari Overview

- Input:
  - Support 15+ services by default
    - Shodan
    - Censys
    - VirusTotal
    - PassiveTotal
    - etc.
  - Can integrate custom sources



Mihari Overview

- Output:
  - Database
    - SQLite3, PostgreSQL or MySQL
  - Slack
  - TheHive
  - MISP
  - Webhook

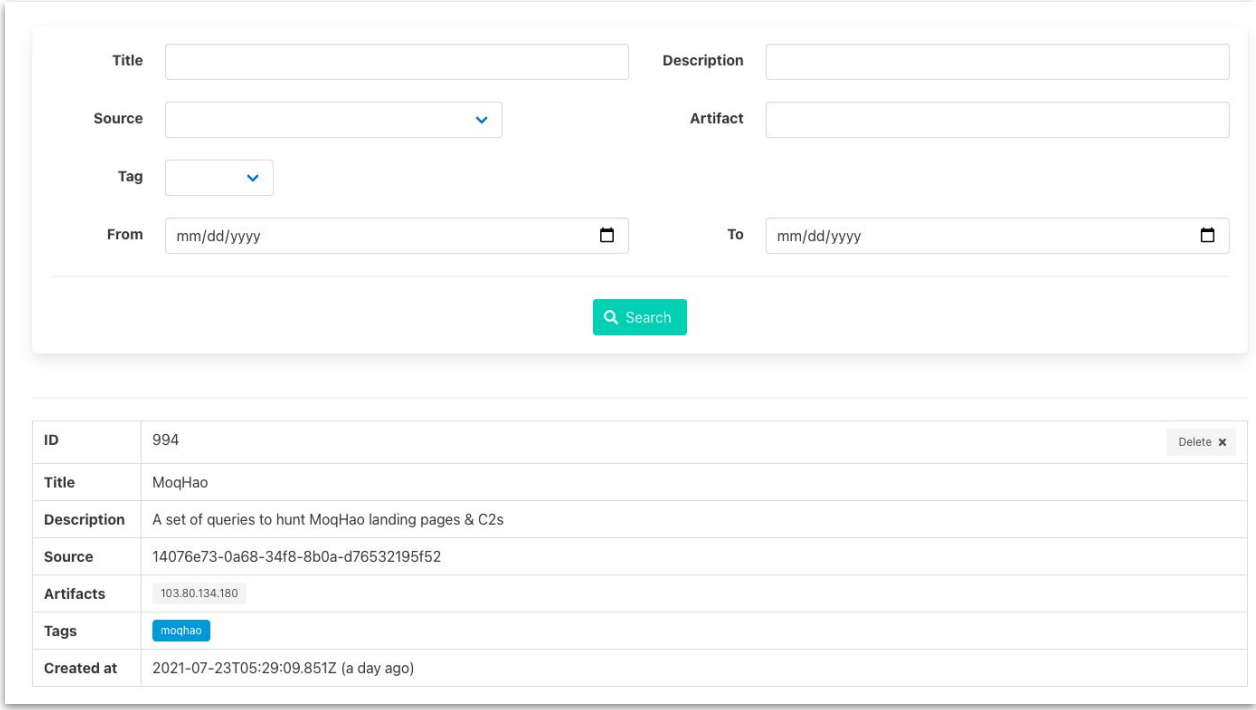


Mihari Usage

- Run an analyzer:
  - $ mihari analyze [**analyzer**] [**query**]
    - $ mihari analyze shodan ip:1.1.1.1
    - $ mihari analyze censys ip:1.1.1.1
    - $ mihari analyze virustotal 1.1.1.1
    - $ mihari analyze securitytrails 1.1.1.1
  - Matched artifacts (IP addresses, domains, URLs, email addresses, hashes) will be stored in the database

Mihari Usage

- Built-in web app:
  - $ mihari web



Mihari Rule

- Mihari has a DSL to combine a set of queries as a rule
- It is inspired by @3c7's infrastructure-tracking-schema
  - https://github.com/3c7/infrastructure-tracking-schema
- A rule is written in a YAML file

Mihari Rule's Schema

```yaml
title: ... # String (required)
description: ... # String (required)

id: ... # String (optional)
author: .. # String (optional)
created_on: ... # Date (optional)
updated_on: ... # Date (optional)

tags: [] # Array<String> (Optional, defaults to [])
allowed_data_types:
  # Array<String> (Optional, defaults to ["hash", "ip", "domain", "url", "mail"])
  - hash
  - ip
  - domain
  - url
  - mail

queries: # Array<Query> (required)
  - analyzer: ... # String (required)
    query: ... # String (required)
```

Mihari Rule's Schema

- Title:
  - A title of the rule
- Description:
  - A description of the rule
- Quereis:
  - Analyzer:
    - A name of a service to use
    - (e.g. shodan)
  - Query:
    - A query to use in a service
    - (e.g. ip:1.1.1.1)

```yaml
title: ...
description: ...

queries:
  - analyzer: ...
    query: ...
```

Mihari Rule for example.com

```yaml
title: example.com HTML
description: A rule to find hosts serve an HTML same as example.com
queries:
  - analyzer: binaryedge
    query: web.body.sha256:ea8fac7c65fb589b0d53560f5251f74f9e9b243478dcb6b3ea79b5e36449c8d9
  - analyzer: censys
    query: services.http.response.body_hash:"sha1:4a3ce8ee11e091dd7923f4d8c6e5b5e41ec7c047"
  - analyzer: shodan
    query: http.html_hash:-2087618365
  - analyzer: urlscan
    query: hash:ea8fac7c65fb589b0d53560f5251f74f9e9b243478dcb6b3ea79b5e36449c8d9
allowed_data_types:
  - ip
```

Mihari Rule for example.com

```yaml
title: example.com HTML
description: A rule to find hosts serve an HTML same as example.com
queries:
  - analyzer: binaryedge
    query: web.body.sha256:ea8fac7c65fb589b0d53560f5251f74f9e9b243478dcb6b3ea79b5e36449c8d9
  - analyzer: censys
    query: services.http.response.body_hash:"sha1:4a3ce8ee11e091dd7923f4d8c6e5b5e41ec7c047"
  - analyzer: shodan
    query: http.html_hash:-2087618365
  - analyzer: urlscan
    query: hash:ea8fac7c65fb589b0d53560f5251f74f9e9b243478dcb6b3ea79b5e36449c8d9
allowed_data_types:
  - ip
```

MMH3

SHA1

SHA256

Tracking DangerousPassword

a.k.a CryptoCore, LeeryTurtle, CryptoMimic

(Souce: https://threatbook.cn/ppt/The%20Nightmare%20of%20Global%20Cryptocurrency%20Companies%20-%20Demystifying%20the%20%E2%80%9CDangerousPassword%E2%80%9D%20of%20the%20APT%20Organization.pdf)

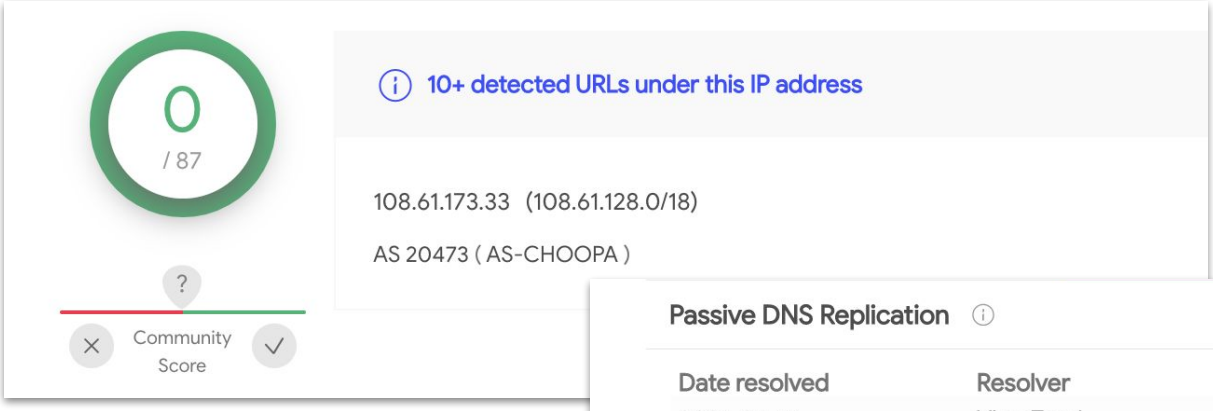
Tracking DangerousPassword

- DangerousPassowrd (a.k.a CryptoCore, LeeryTurtle, CryptoMimic):
  - An APT group that targets cryptocurrency exchanges around the world
  - Sometimes it reuses the same infrastructure for a certain period

Swiftness and responsiveness – the group's infrastructure is continuously and rapidly changing. While in some cases we have seen the same infrastructures being constantly reused, perhaps against multiple victims, the group is generally quick to register and employ new domains and links. In some cases, the freshly created bit.ly link is used immediately, on the same

(Source: https://www.clearskysec.com/wp-content/uploads/2020/06/CryptoCore_Group.pdf)

Tracking DangerousPassword



Passive DNS Replication

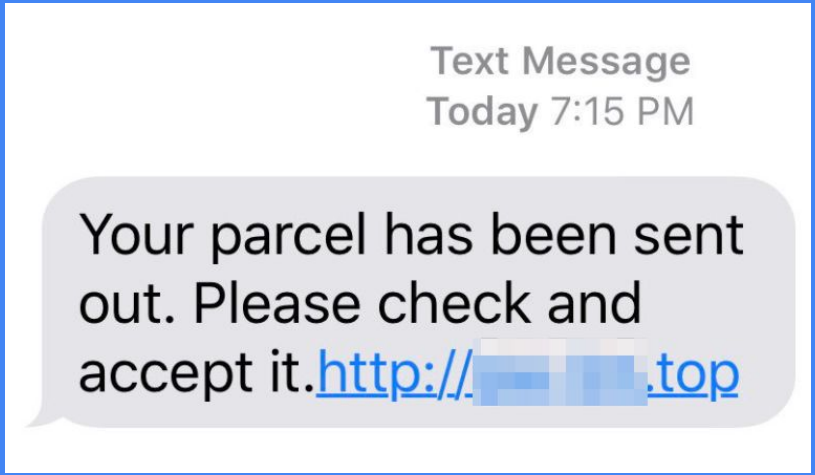
Date resolved	Resolver	Domain
2021-07-15	VirusTotal	dcview.coresharedoc.club
2021-07-14	VirusTotal	sharedlnk.statementshare.com
2021-07-12	VirusTotal	share.bloomcloud.org
2021-07-10	VirusTotal	product.onlinedoc.dev
2021-07-08	VirusTotal	share.cloud-share.org
2021-07-08	VirusTotal	drive.cloud-share.org
2021-07-02	VirusTotal	share.devprocloud.com
2021-06-28	VirusTotal	signverydn.sharebusiness.xyz
2021-06-18	VirusTotal	sharemanage.elwoodasset.xyz

Tracking DangerousPassword

```yaml
title: DangerousPassword
description: A set of queries to hunt DangerousPassword landing pages through PassiveDNS
tags:
  - dangerouspassword
queries:
  - analyzer: virustotal
    query: 149.248.8.85
  - analyzer: virustotal
    query: 108.61.173.33
  - analyzer: virustotal
    query: 152.89.247.194
  - analyzer: virustotal
    query: ...
```

Tracking MoqHao

a.k.a XLoader



Tracking MoqHao

- MoqHao (a.k.a XLoader):
  - An Android malware targets South Korea, Taiwan, Japan, Germany, etc.

(Source: https://twitter.com/LukasStefanko/status/1248296767860375555)

Tracking MoqHao

```yaml
title: MoqHao
description: A set of queries to hunt MoqHao landing pages & C2s
tags:
  - moqhao
queries:
  - analyzer: shodan
    query: ssl.cert.serial:1165[redacted]
  - analyzer: shodan
    query: http.html_hash:-1931[redacted]
  - analyzer: censys
    query:
services.certificate:76de6[redacted]
  - analyzer: censys
    query: services.http.response.html_title:[redacted] AND (services.port:10080 OR
services.port:10081) AND (service.port:5985 AND service.port:80)
```

Tracking MoqHao

```
title: MoqHao
description: A set of queries to hunt MoqHao landing pages & C2s
tags:
  - moqhao
queries:
  - analyzer: shodan
    query: ssl.cert.serial:1165[redacted]
  - analyzer: shodan
    query: http.html_hash:-1931[redacted]
  - analyzer: censys
    query:
services.certificate:76de6[redacted]
  - analyzer: censys
    query: services.http.response.html_title:[redacted] AND (services.port:10080 OR
services.port:10081) AND (service.port:5985 AND service.port:80)
```

Self signed SSL certificate

Ping tracker

Tracking MoqHao

- MoqHao uses Pinterest as C2

```
private final q r = new q(this);
private final List<c> s = d.h.k.d(new c("jp.co.smbc.direct", "https://www.pinterest.com/emeraldquinn4090/",
```

- Pinterest accounts:
  - abigailn674, catogreggex11, emeraldquinn4090, felicitynewman8858, gh6855786, husaincrisp, ingalcliffth, etc.

Tracking MoqHao

- It is possible to create a custom analyzer to ingest a custom source

```ruby
require "mihari"

module Mihari
  module Analyzers
    class Example < Base
      def title
        "example"
      end

      def description
        "example"
      end

      def artifacts
        ["9.9.9.9", "example.com"]
      end

      def tags
        ["example"]
      end
    end
  end
end
```

Tracking MoqHao

```ruby
class Pinterest < Mihari::Analyzers::Base
  def title
    "MoqHao phishing on Pinterest"
  end

  def description
    title
  end

  def artifacts
    urls.filter_map do |url|
      get_phishing_url_from_about url
    end.map do |url|
      [URI(url).host, url]
    end.flatten.uniq
  end

  def tags
    ["phishing", "moqhao"]
  end

  private

  def get_phishing_url_from_about(url)
```

ID	986
Title	MoqHao phishing on Pinterest
Description	MoqHao phishing on Pinterest
Source	Pinterest
Artifacts	http://cqsczahojc.duckdns.org cqsczahojc.duckdns.org
Tags	phishing moqhao
Created at	2021-06-07T09:44:43.300Z (2 months ago)

Tips

- It is also possible to make an input via REST API:
  - $ mihari web
  - $ http localhost:9292/api/analyzer ...

title required	string
description required	string
source required	string
artifacts required	string
tags	Array of strings
ignoreOldArtifacts	boolean Default: `false`
ignoreThreshold	integer Default: `0`

Payload

Content type
application/json

Copy Expand all Collapse all


```
{
    "title": "string",
    "description": "string",
    "source": "string",
    "artifacts": "string",
  - "tags": [
        "string"
    ],
    "ignoreOldArtifacts": false,
    "ignoreThreshold": 0
}
```

Conclusions

Conclusions

- Adversary infrastructure tracking brings new insights and findings
  - ⚠️ You should combine it with static/dynamic analysis and YARA hunting to get the whole picture
- Mihari is a tool to make the tracking easy
  - Mihari provides a unified way to interact with various services
  - Mihari pings you when there are new findings
  - 💡 You can get better coverage by combining a set of queries in a rule

Questions?

Appendix: Related Articles

- Alphathreat Soup: Burning Actors with Data (RiskIQ)
  - https://hitcon.org/2018/CMT/slide-files/d1_s2_r1.pdf
- Advanced Persistent Infrastructure Tracking (Censys)
  - https://censys.io/blog/advanced-persistent-infrastructure-tracking/
- Investigating Infrastructure Links with Passive DNS and Whois Data (Amnesty International)
  - https://citizenevidence.org/2020/06/26/investigating-infrastructure-links-with-passive-dns-and-whois-data/
- Cert Safari: Leveraging TLS Certificates to Hunt Evil (Prevailion)
  - https://www.prevailion.com/cert-safari-leveraging-tls-certificates-to-hunt-evil/