

Modeling and Sharing DDoS attacks with MISP

TPL:CLEAR



CIRCL

Computer Incident
Response Center
Luxembourg

Jean-Louis Huynen

Luxembourg Internet Days 16/11/2022

MISP ?

- MISP is a **threat information sharing** platform that is free & open source software
- A tool that **collects** information from partners, your analysts, your tools, feeds
- Normalises, **correlates**, **enriches** the data
- Allows teams and communities to **collaborate**
- **Feeds** automated protective tools and analyst tools with the output



Co-financed by the European Union
Connecting Europe Facility

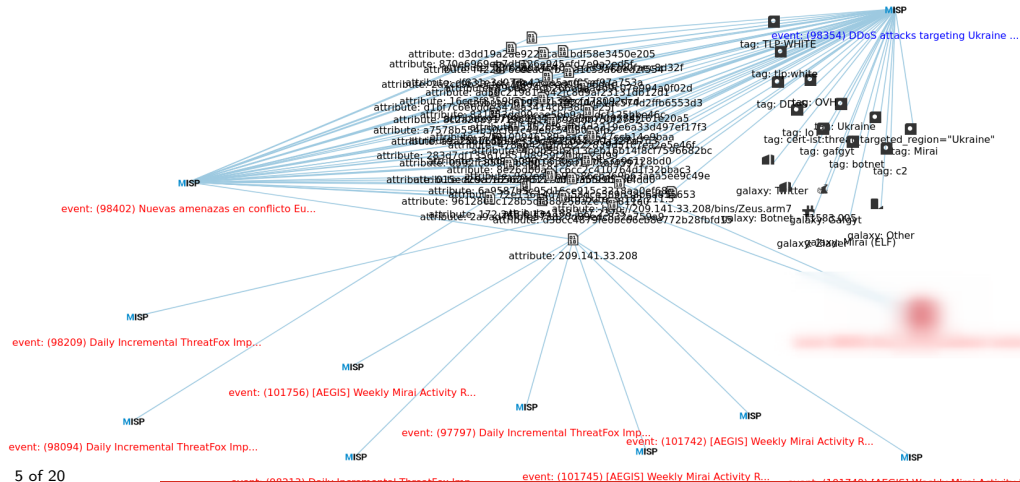
MISP ?

- There are many different types of users of an information sharing platform like MISP:
 - **Malware reversers** willing to share indicators of analysis with respective colleagues.
 - **Security analysts** searching, validating and using indicators in operational security.
 - **Intelligence analysts** gathering information about specific adversary groups.
 - **Law-enforcement** relying on indicators to support or bootstrap their DFIR cases.
 - **Risk analysis teams** willing to know about the new threats, likelihood and occurrences.
 - **Fraud analysts** willing to share financial indicators to detect financial frauds.

The matter at hand

- CIRCL runs multiple large MISP communities performing active daily threat-intelligence sharing
- yet few events are related to DDoS attacks.
- There is value in sharing DDoS information:
 - correlation between DDoS events
 - sectors affected
 - Technique Tactics and procedure
 - Threat Actors
 - tools
 - timeline
 - remediations
 - etc.
 - plus DDoS infrastructures (and botnet) stay relatively stable.

The matter at hand



The matter at hand

To make the matter worse, some entities still share Cyber Threat Intel in pdf.

How MISP can help

- Sharing DDoS Statistics
- Sharing DDoS Characteristics
- Contextualize through tags
- Sharing samples
- Sharing insights

Sharing DDoS Statistics - DDoS MISP object

- backscatter-threshold
- capture-origin
- domain-dst
- dst-port
- first-seen
- ip-dst
- ip-src
- last-seen
- protocol
- src-port
- text
- total-bps
- total-bytes-sent
- total-packets-sent
- total-pps
- type

Sharing DDoS Characteristics - Contextualize through tags

MITRE ATT&CK

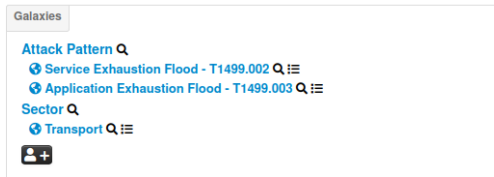
- TA0040 Impact
 - T1498 Network Denial of Service
 - T1498-001 Direct Network Flood
 - T1498-002 Reflection Amplification



Sharing DDoS Characteristics - Contextualize through tags

MITRE ATT&CK

- TA0040 Impact
 - T1499 Endpoint Denial of Service
 - T1499-001 OS Exhaustion Flood
 - T1499-002 Service Exhaustion Flood
 - T1499-003 Application Exhaustion Flood
 - T1499-004 Application or System Exploitation



Sharing DDoS Characteristics - Galaxies

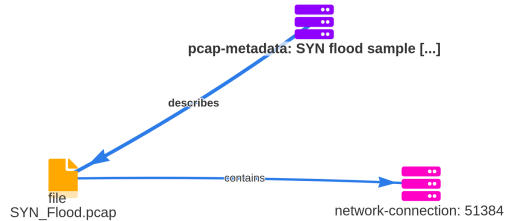
MISP also contains galaxies used to describe threat actors, tools, botnets and affected sectors that come handy for contextualisation:

- Threat Actors
 - 400 threat actors,
 - some known to perform DDoS ;)
- Well-known Tools
 - 500+
- Well-known Botnets
 - 70+

Add your own, contribute to <https://github.com/MISP/misp-galaxy>

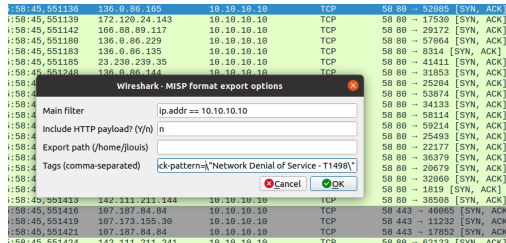
Sharing samples - network captures

- pcap-metadata:
 - capture-interface
 - capture-length
 - first-packet-seen
 - last-packet-seen
 - protocol
 - text
- file attachment
- network connection



Sharing samples - misp-wireshark

- misp-wireshark is a lua plugin for wireshark / tshark
- convert data from wireshark and convert it into MISP core format
- support DNS, TCP, and HTTP
- export HTTP payload
- more to come




Sharing samples - misp-wireshark - tshark









```
def analyse_pcap(m, path, filename):
    p = subprocess.run(["tshark",
        "-r", path,
        "-X", "lua_script:{}".format(lua_script),
        "-X", "lua_script1:include_payload=n",
        "-X", "lua_script1:filters=ip.addr == 10.10.10.10",
        "-X", "lua_script1:tags=\"tlp1,tlp2\"",
        "frame.number == 0"],
        capture_output=True, text=True)
    # useful to debug
    #print(p)
    #print(p.stdout)


    if len(p.stdout) > 0:
        event = json.loads(p.stdout)
        print(event['uuid'])
        event['info'] = filename
        misp.add_event(event)
```













Example - DDOSIA - tags





Galaxies

Threat Actor 

-  Killnet   
-  NoName057(16)   

Attack Pattern 

-  OS Exhaustion Flood - T1499.001   
-  Service Exhaustion Flood - T1499.002   
-  Network Denial of Service - T1498   

Example - DDOSIA - import

```
    "type": "",
    "value": ""
  },
  "use_random_user_agent": true,
  "timeout": 1000,
  "response": true,
  "headers": [],
  "is_deleted": false
},
{
  "id": "636a087700aff82b978bae6e",
  "ratio": "1",
  "type": "tcp",
  "method": "syn",
  "host": "lipno.sr.gov.pl",
  "address": "85.128.162.177",
  "port": 3306,
  "use_ssl": false,
  "path": "",
  "body": {
    "type": "",
    "value": ""
  },
  "use_random_user_agent": true,
  "timeout": 1000,
  "response": true,
  "headers": [],
  "is_deleted": false
},
{
  "id": "636a087700aff82b978bae6f",
  "ratio": "1",
  "type": "tcp",
  "method": "syn",
  "host": "lipno.sr.gov.pl",
  "address": "85.128.162.177",
  "port": 3306,
  "use_ssl": false,
  "path": "",
  "body": {
    "type": "",
    "value": ""
  },
  "use_random_user_agent": true,
  "timeout": 1000,
  "response": true,
  "headers": [],
  "is_deleted": false
}
```

2022-11-08		Object name: ddosia	
		References: 1 2 3	
<input type="checkbox"/>	2022-11-08	Other	capture-origin: DDOSIA target list
			text
<input type="checkbox"/>	2022-11-08	Network activity	domain-dst: lipno.sr.gov.pl
			domain
<input type="checkbox"/>	2022-11-08	Network activity	dst-port: 3306
			port
<input type="checkbox"/>	2022-11-08	Network activity	ip-dst: 85.128.162.177
			ip-dst
<input type="checkbox"/>	2022-11-08	Other	protocol: tcp
			text
<input type="checkbox"/>	2022-11-08	Other	type: flooding-attack
			text

Example - DDOSIA - import

```
body : {
  "type": "",
  "value": ""
},
"use_random_user_agent": true,
"timeout": 1000,
"response": true,
"headers": [],
"is_deleted": false
},
{
  "id": "636a087700aff82b978bae6e",
  "ratio": "1",
  "type": "tcp",
  "method": "syn",
  "host": "lipno.sr.gov.pl",
  "address": "85.128.162.177",
  "port": 3306,
  "use_ssl": false,
  "path": "",
  "body": {
    "type": "",
    "value": ""
  },
  "use_random_user_agent": true,
  "timeout": 1000,
  "response": true,
  "headers": [],
  "is_deleted": false
},
{
  "id": "636a087700aff82b978bae6f",
  "ratio": "1",
  "type": "tcp",
  "method": "syn",
  "host": "lipno.sr.gov.pl",
  "address": "85.128.162.177",
  "port": 3306,
  "use_ssl": false,
  "path": "",
  "body": {
    "type": "",
    "value": ""
  },
  "use_random_user_agent": true,
  "timeout": 1000,
  "response": true,
  "headers": [],
  "is_deleted": false
}
```

```
def init(url, key):
    return PyMISP(url, key, misp_verifycert, 'json')

if __name__ == '__main__':
    misp = init(misp_url, misp_key)

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Create event per DDoS traces')
    parser.add_argument("-f", "--file", help="Path to file")

    args = parser.parse_args()

    misp = init(misp_url, misp_key)

    if args.file is not None and os.path.exists(args.file):
        print('Working on {}'.format(args.file))
        with open(args.file, encoding = 'utf-8') as f:
            t = json.loads(f.read())
            event = misp.get_event(27, pythonify = True)

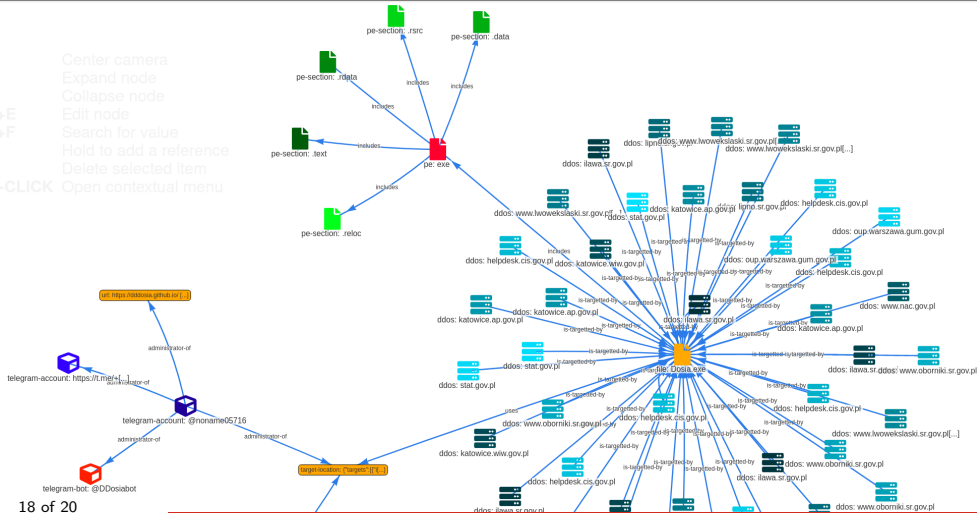
            for target in t:
                ddos = MISPObject('ddos')
                ddos.add_attribute('capture-origin', value= 'DDOSIA target list')
                ddos.add_attribute('domain-dst', value= target['host'])
                ddos.add_attribute('dst-port', value= target['port'])
                ddos.add_attribute('ip-dst', value= target['address'])
                ddos.add_attribute('protocol', value= target['type'])
                ddos.add_attribute('type', value= 'flooding-attack')
                ddos.add_attribute('text', value= target['path'])
                ddos.add_reference('233677c4-34e4-49af-acfb-4079a6240b40', 'is-targetted-by')
                event.add_object(ddos)

            res = misp.update_event(event)

    else:
        exit(0)
```

Example - DDOSIA - references

Center camera
Expand node
Collapse node
+E Edit node
+F Search for value
Hold to add a reference
Delete selected item
-CLICK Open contextual menu



Example - DDOSIA - MISP report

INSIGHTS

In [attachment Advisory-Project-DDOSIA.pdf](#), radware described the DDOSIA project launched by NoName057(16), and actor related to [threat-actor ↔ Killnet](#)

The project was launched in July 2022. It is a crowdsourced and fully automated DDoS bot project in which politically-driven hacktivists willing to download and install a bot on their computers launch a DoS attacks on several targets. DDOSIA provide a financial incentive to participants.

Background

NoName057(16) is a pro-Russian threat group known for launching defacement and DDoS attacks against Ukraine and those that directly or indirectly support Ukraine. The group formed in March of 2022 on Telegram and became a notable threat group by June. Since then, the group has gathered a following of nearly 13,000 subscribers.

Over the last few months, NoName057(16) has been operating in support of Killnet operations. Most recently, the group worked in parallel with Killnet during their campaign against civilian network infrastructure in the United States. During the operation, threat group NoName057(16) posted an invite link to a Telegram channel named [telegram-account DDosia Project](#) and also reposted the Killnet target list for U.S. airports in the same channel.

see [image manifesto.png](#)

Project DDOSIA

Mid-August, while publishing their manifesto, NoName057(16) simultaneously disclosed their 'special software' that will assist them in conducting DDoS attacks. Over the following days, the group provided more information about their 'special software' named DDOSIA and instructions on using it to contribute to the fight against Western Russophobes. Instructions, publicly available at [url https://dddosia.github.io/](https://dddosia.github.io/), explain how potential contributors can register through Telegram to receive a ZIP archive containing a Windows bot binary named 'dosia.exe' and a unique identifier file with the name 'client_id.txt.' The unique identifier allows the contributor to create a bragging alias while registration of a cryptocurrency wallet is required to receive potential financial rewards at a later phase in the project.

After the bot agent [file 7.9868541302137](#) is executed on a contributor's Windows machine, the bot registers itself with the command-and-control (C2) infrastructure of the authors. Subsequently, the C2 servers feed the bot with a list of targets [url 109.107.181.130](#), after which the malicious software begins attacking the provided targets with TLS encrypted Layer 7 and TCP-SYN denial-of-service attacks as for instance [ddos www.oborniki.sr.gov.pl](#) and [ddos www.oborniki.sr.gov.pl](#).

Credits and References

- **misp-project website:** <https://misp-project.org>
- **misp-wireshark:** <https://github.com/MISP/misp-wireshark>
- **MISP DDoS object:** https://www.misp-project.org/objects.html#_ddos
- **MISP pcap-metadata object:**
https://www.misp-project.org/objects.html#_pcap_metadata
- **PyMISP documentation:** <https://pymisp.readthedocs.io/en/latest/>
- **MISP training materials:** <https://www.circl.lu/services/misp-training-materials/>
- **PISAX:** <https://www.pisax.org>
- **RADWARE DDOSIA report** <https://www.radware.com/security/threat-advisories-and-attack-reports/project-ddosia-russias-answer-to-disbalancer/>