Análise de Malware com Automação de coleta Open-Source

Bruno Odon
Caique Barqueta

TLP:CLEAR

Caique Barqueta

Caique Barqueta

Especialista em Cyber Threat Intelligence (ISH)

DFIR

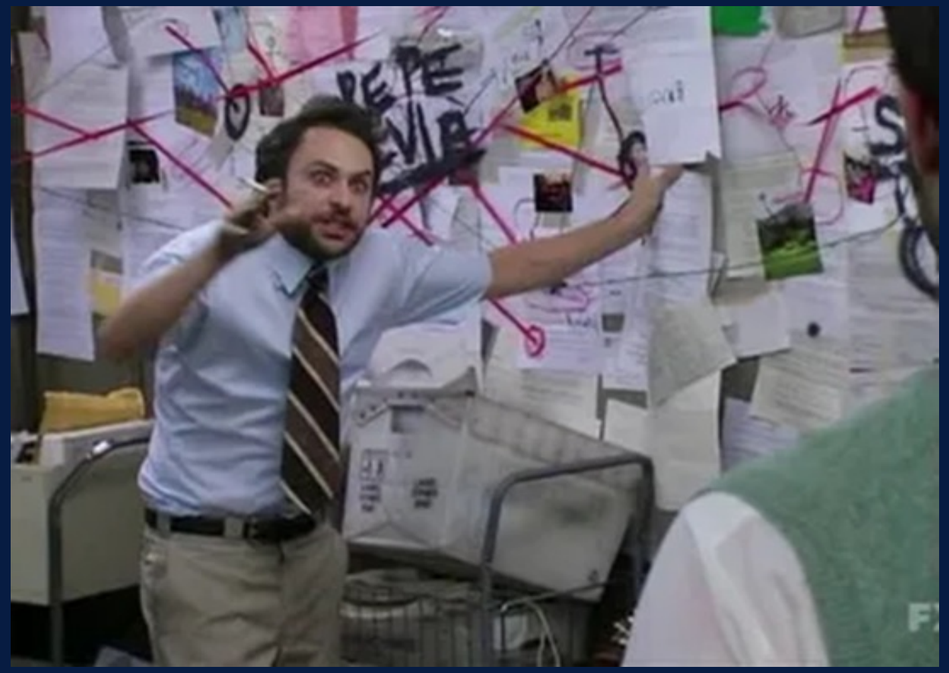
Pesquisador de novos grupos de Ransomware

Professor (DFIR, Malware e CTI)

Pós-graduado em Comp. Forense e Defesa Cibernética

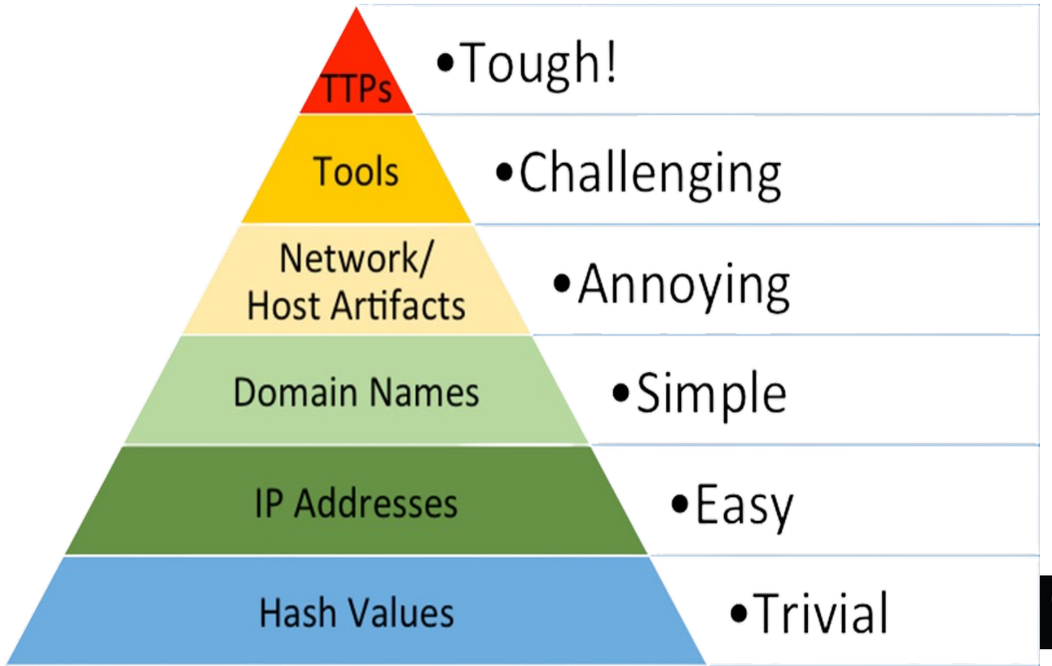
Algumas certificações

TLP:CLEAR



Cyber Threat Intelligence

Consiste na **Coleta, Processamento e Análise** de informações para entender possíveis **motivos, alvos e comportamento** de um ator de ameaça.



TLP:CLEAR

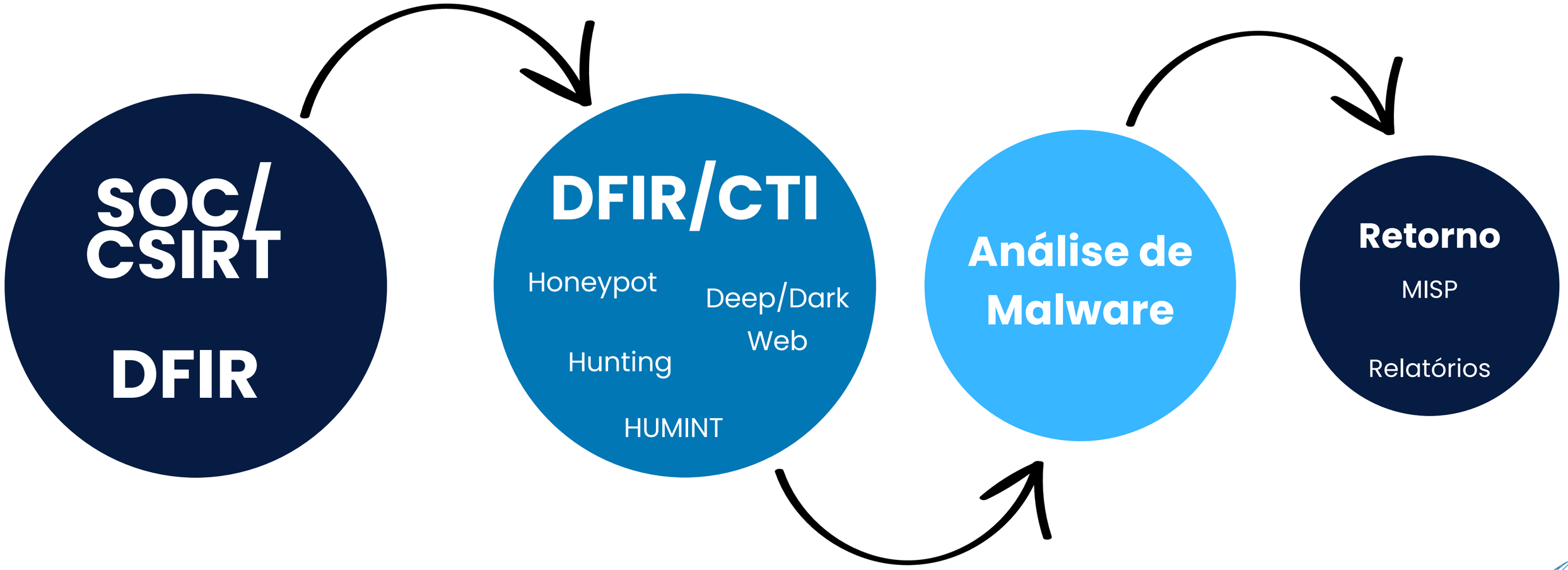
Análise de Malware

Estudar e processar determinada **funcionalidade**, **origem** e **impacto** que um malware pode ocasionar.

Para isso é realizada a análise, onde é possível determinar o tipo de ameaça e a criticidade.

Exemplos: *RATs, Ransomwares, Trojans...*

TLP:CLEAR



Integração com outras áreas

TLP:CLEAR

Exemplo de Caso

Incidente de Segurança envolvendo os operadores de Ransomware LockBit

TLP:CLEAR

2019

Início das Operações como:

Ransomware ABCD

Nome de ator de ameaça:

BITWISE Spider

All your important files are encrypted!
There is only one way to get your files back:
1. Contact with us
2. Send us 1 any encrypted your file and your personal key
3. We will decrypt 1 file for test(maximum file size - 1 MB), its guarantee what we can decrypt your files
4. Pay
5. We send for you decryptor software

We accept Bitcoin

Attention!
Do not rename encrypted files.
Do not try to decrypt using third party software, it may cause permanent data loss.
Decryption of your files with the help of third parties may cause increased price(they add their fee to our)

Contact information: goodmen@countermail.com

Be sure to duplicate your message on the e-mail: goodmen@cock.li

Nota de resgate Ransomware abcd

TLP:CLEAR

2020

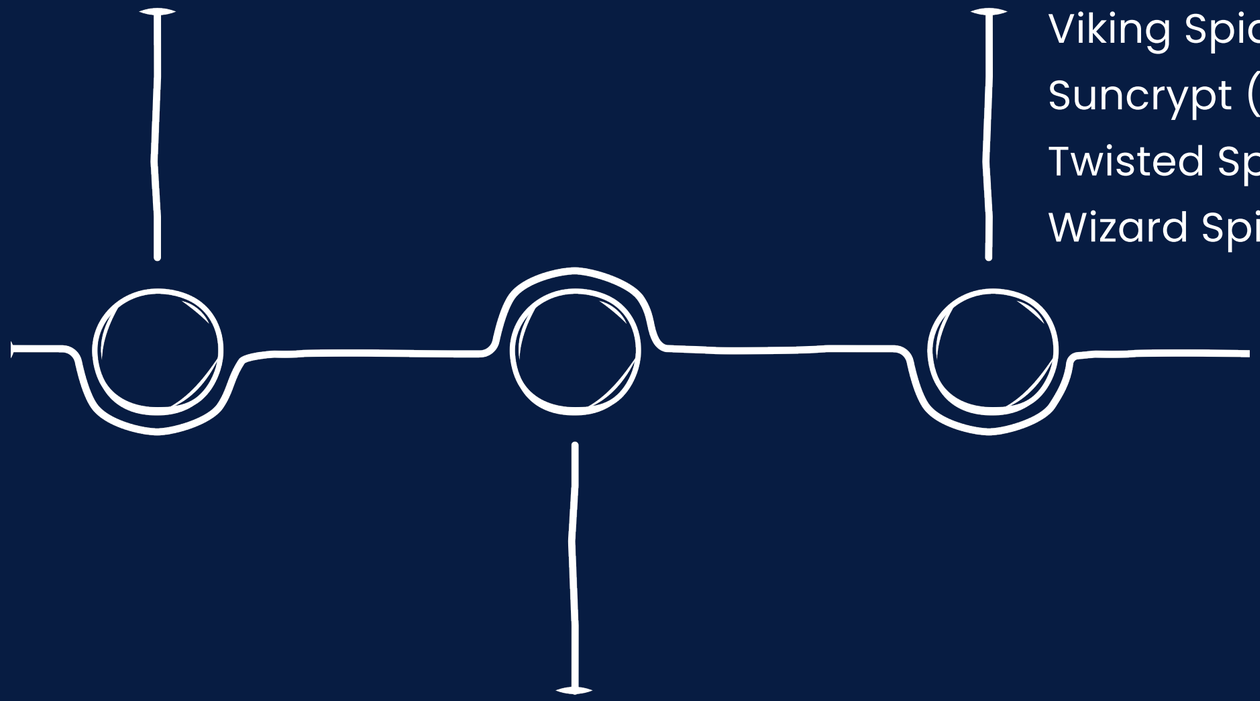
Nova variante:

LockBit e programa RaaS

Criação do

Cartel de Ransomware

Viking Spider (Ragner Locker)
Suncrypt (Suncrypt Ransomware)
Twisted Spider (Maze e Egregor)
Wizard Spider (Ryuk, Conti e Egregor)



Concurso:

"Summer Paper Contest"

US$ 10k

TLP:CLEAR

2021

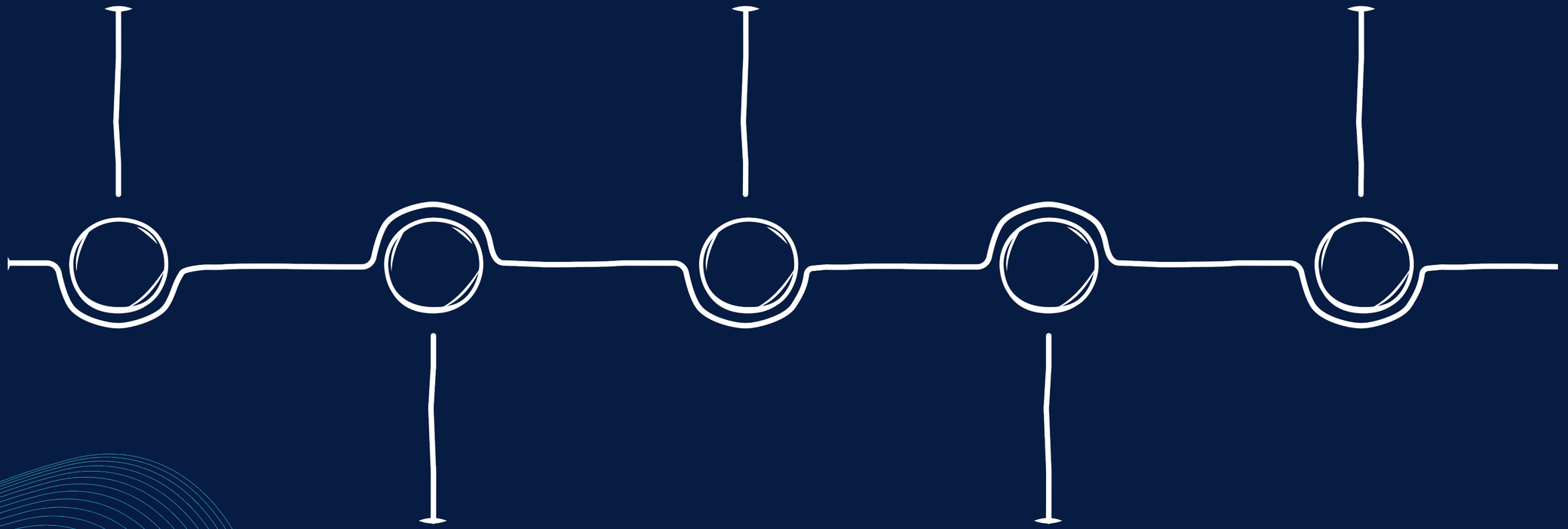
Nova variante:
LockBit2.0 ou LockBit Red

Lançamento do:
StealBit

Nova variante:
LockBit Linux-ESXi

Contratação do dev da DarkSide (FIN7, Alphv)
Surgimento de novas variantes com códigos do BlackMatter

Brigas com outros operadores de Ransomwares (REvil e Hive)



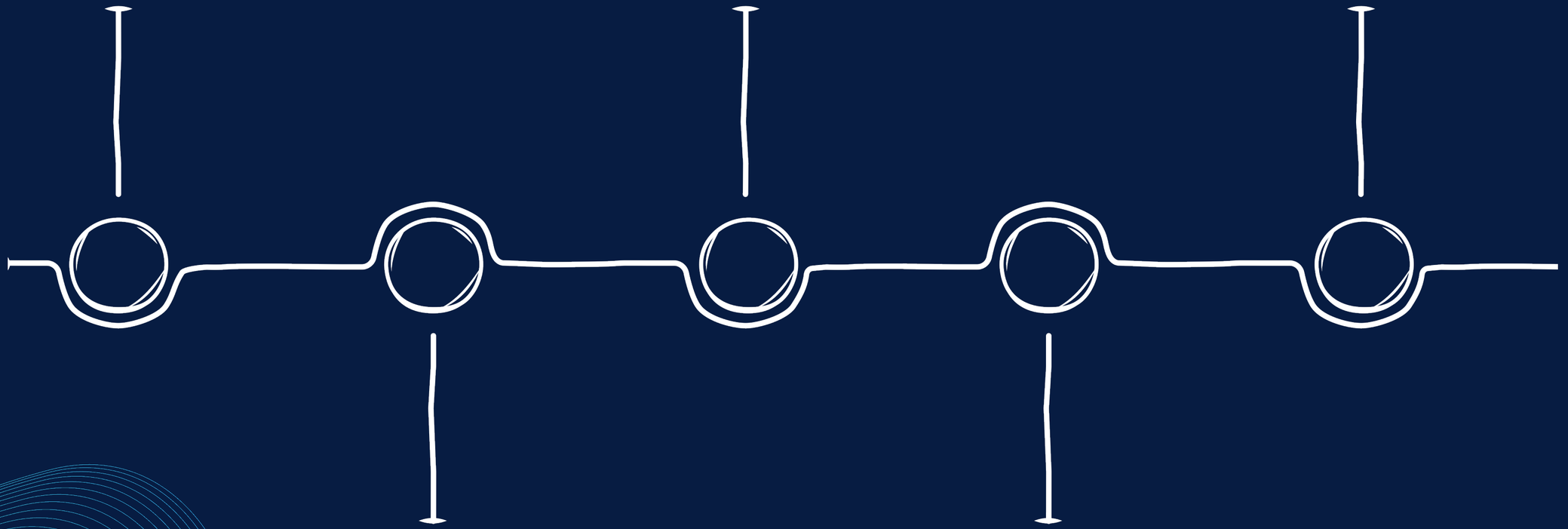
TLP:CLEAR

2022

Sem envolvimento na guerra (UK x RU)

Programa de **Bug Bounty** e Concurso de **Tatuagem**

Vazamento de um builder do LockBit3.0 pelo dev. (Fórum CSIRT)



Nova variante: LockBit3.0 ou LockBit Black

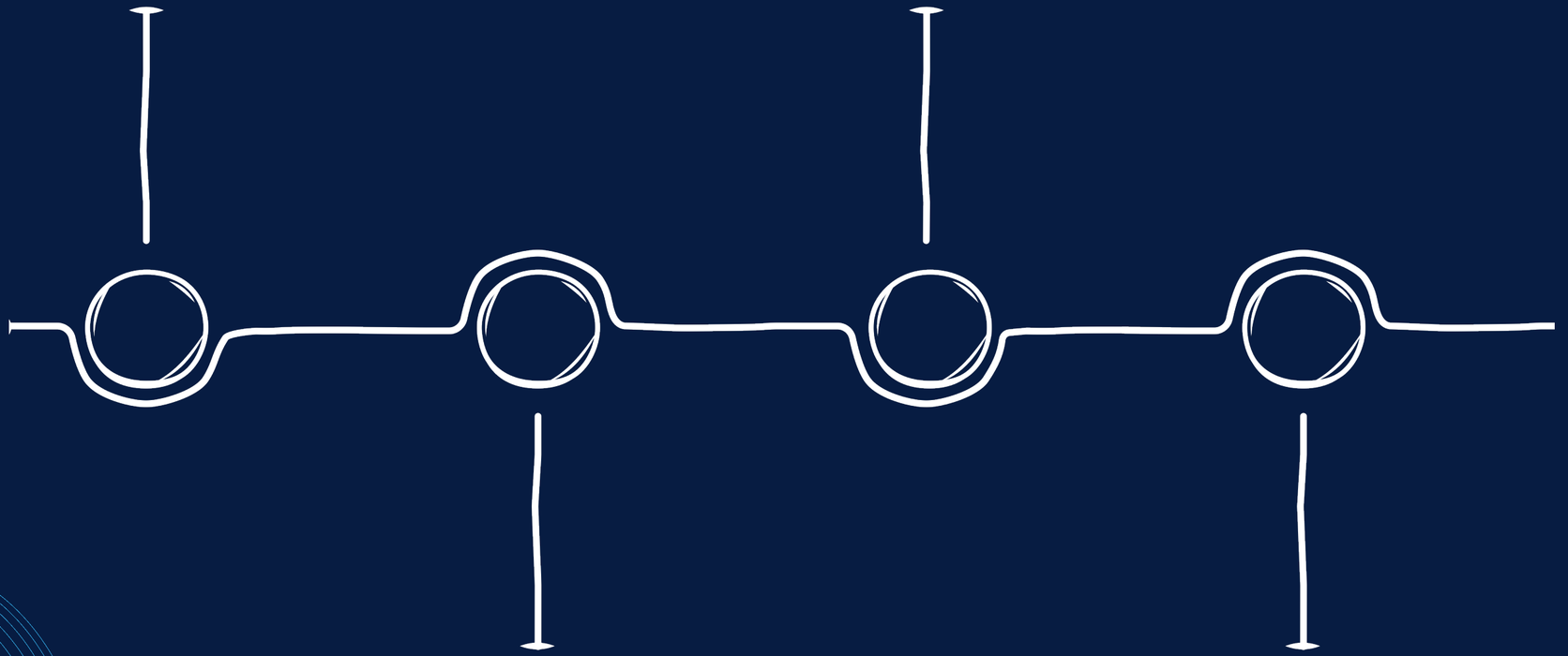
Sofreu ataque **DDoS** de uma empresa de Segurança

TLP:CLEAR

2023

Nova variante:
LockBit Green
(foco em cloud)

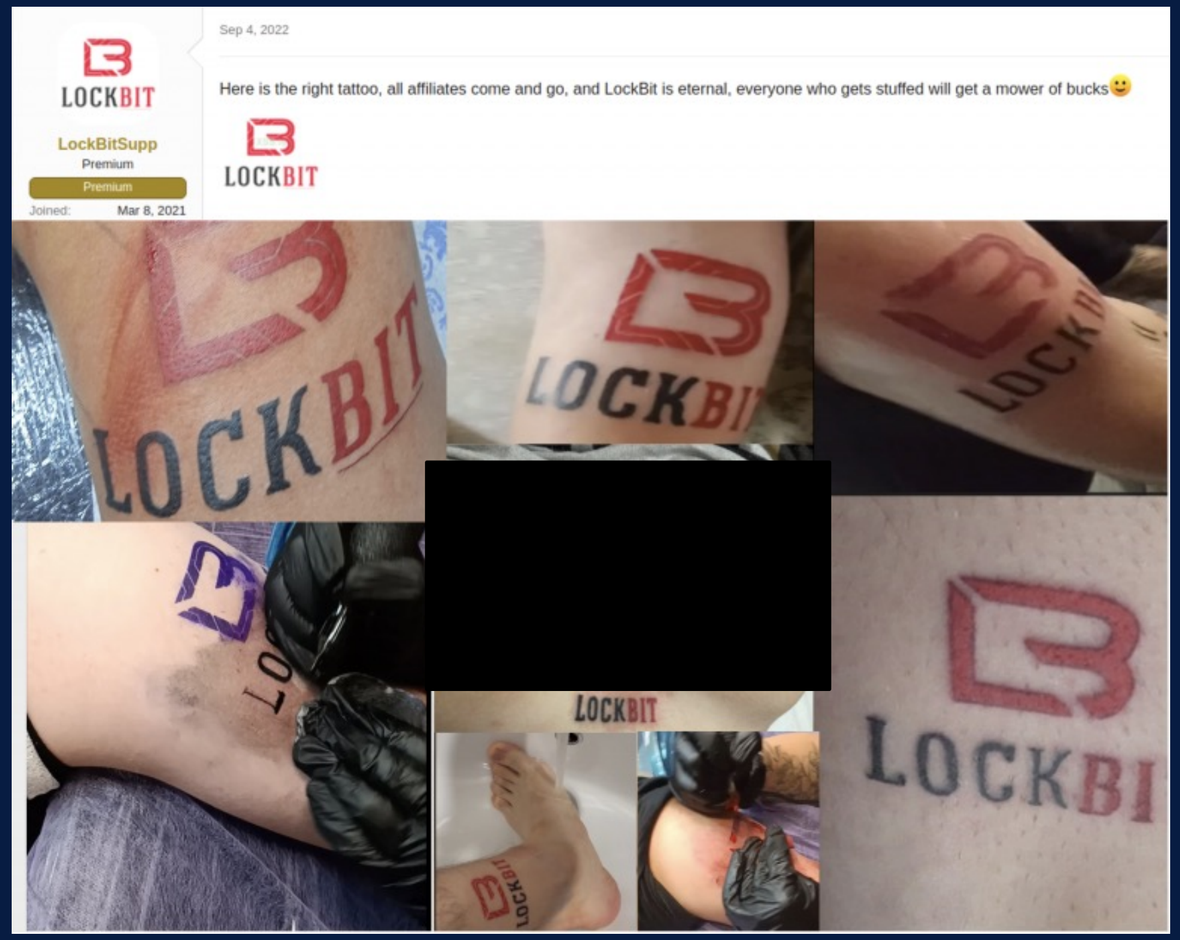
Nova variante:
LockBit para macOS



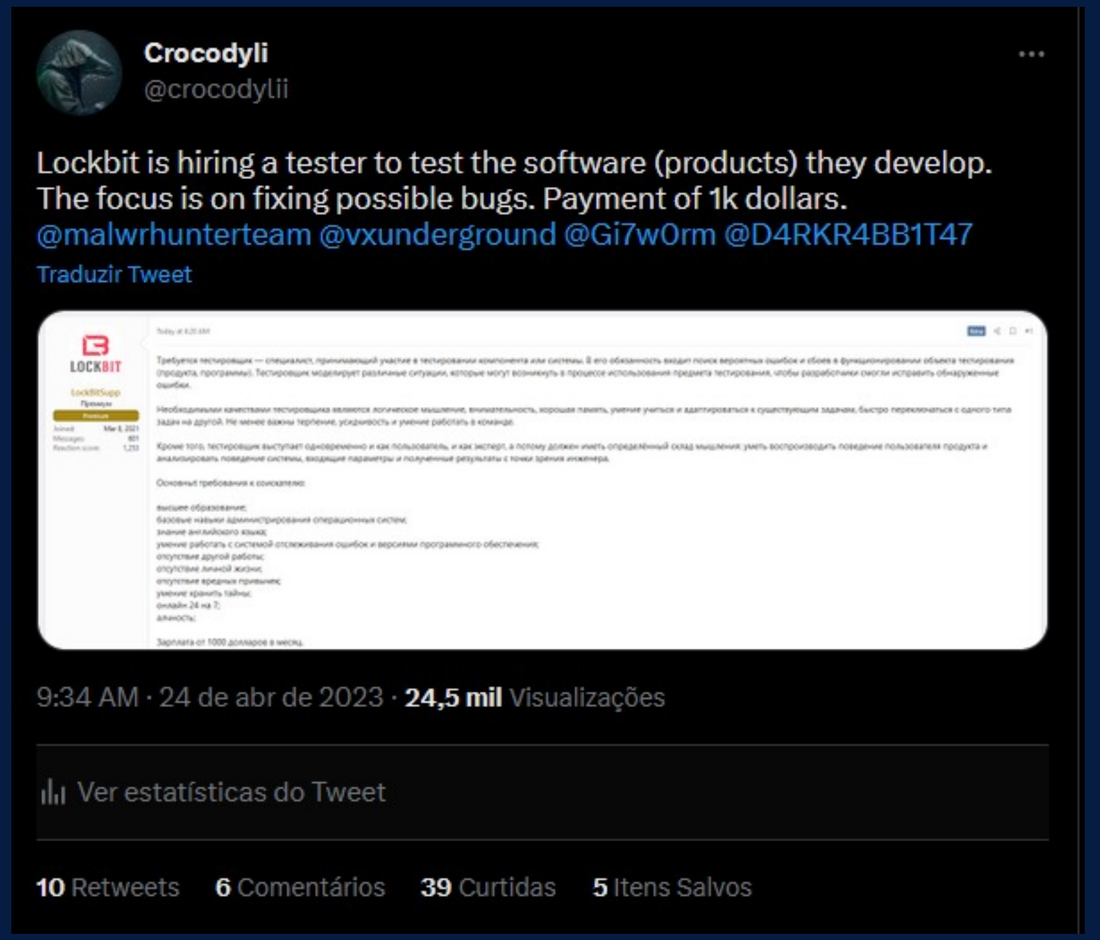
Leaked Data
Site de vazamentos de dados

Recrutamento de "tester" para os "produtos" do LockBit

TLP:CLEAR



Exemplos de pessoas que tatuaram



Não possuir maus hábitos

TLP:CLEAR

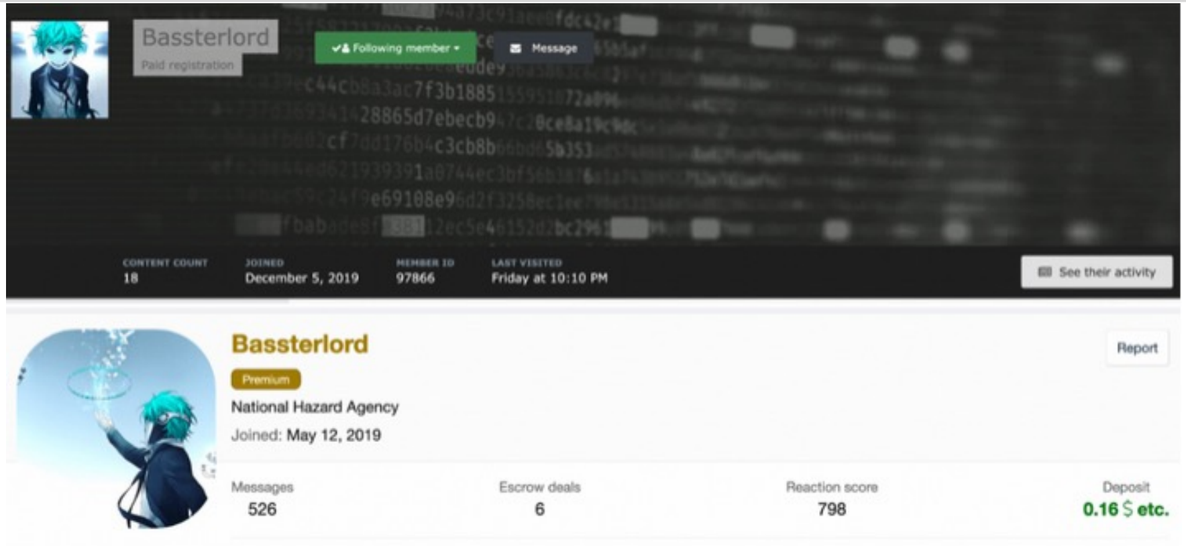
Afiliados

Basterlord

- National Hazard Agency
- REvil, RansomEXX, Avadon e LockBit

Wazawaka:

- Babuk, Hive e LockBit



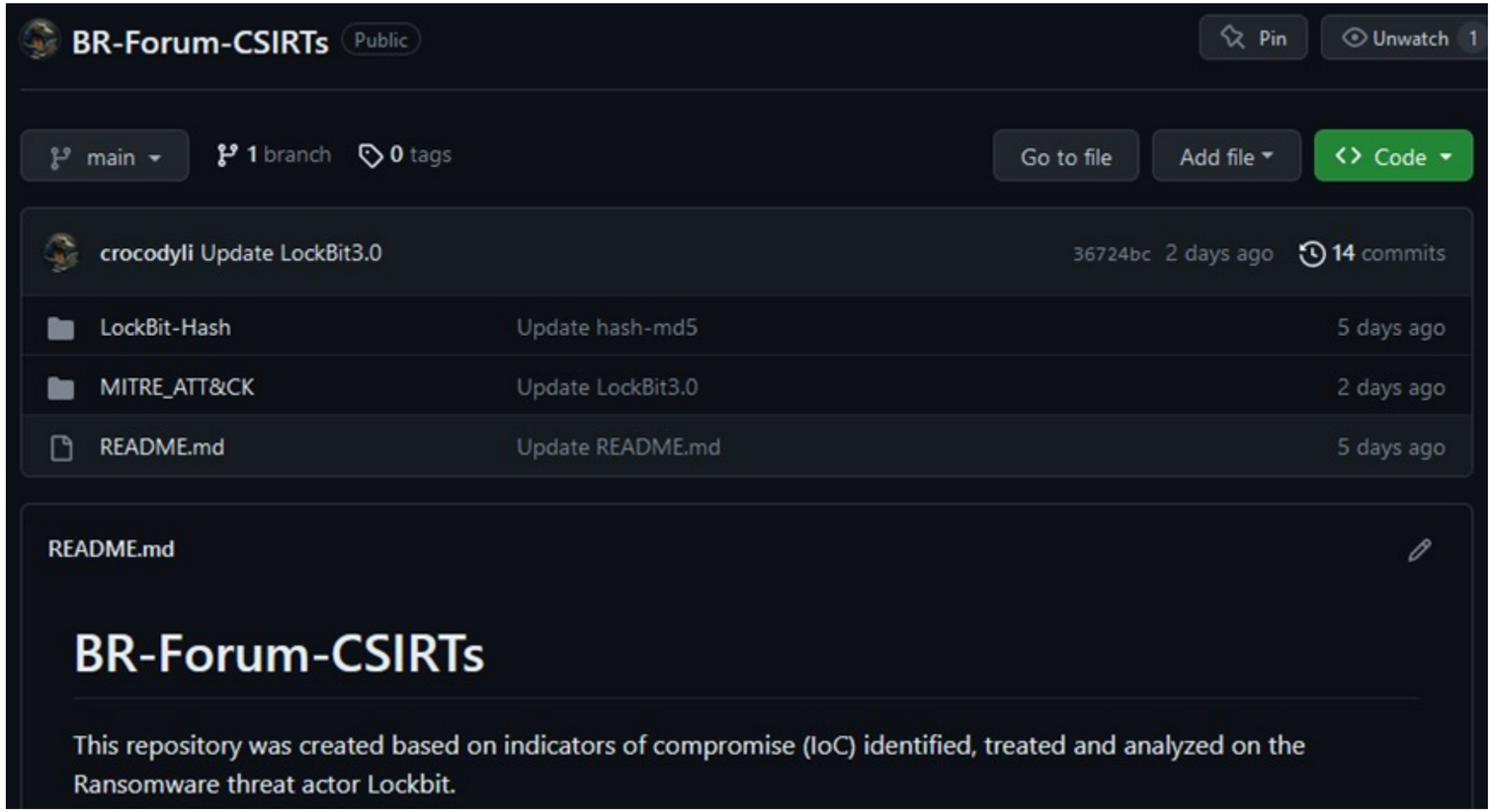
TLP:CLEAR

Coletadas as informações, o que fazer?

TLP:CLEAR

Dados coletados e tratados

Exemplo de repositório que pode ser utilizado para armazenamento de informações e compartilhamento.



TLP:CLEAR

Dados coletados e tratados

Exemplo de dados tratados:

- TTP's - MITRE ATT&CK
- Ferramentas utilizadas

Initial Access		
Technique Title	**ID**	**Use**
Valid Accounts	T1078	LockBit 3.0 actors obtain and abuse credentials of existing accounts as a means of gaining initial access.
Exploit External Remote Services	T1133	LockBit 3.0 actors exploit RDP to gain access to victim networks.
Drive-by Compromise	T1189	LockBit 3.0 actors gain access to a system through a user visiting a website over the normal course of browsing.
Exploit Public-Facing Application	T1190	LockBit 3.0 actors exploit vulnerabilities in internet-facing systems to gain access to victims' systems.
Phishing	T1566	LockBit 3.0 actors use phishing and spearphishing to gain access to victims' networks.
Execution		
Technique Title	**ID**	**Use**
Execution	TA0002	LockBit 3.0 launches commands during its execution.
Command and Scripting Interpreter: Windows Command Shell	T1059.003	LockBit affiliates use batch scripts to execute malicious commands.
System Services: Service Execution	T1569.002	LockBit3.0 uses PsExec to execute commands or payloads.
Software Deployment Tools	T1072	LockBit 3.0 uses Chocolatey, a command- line package manager for Windows.

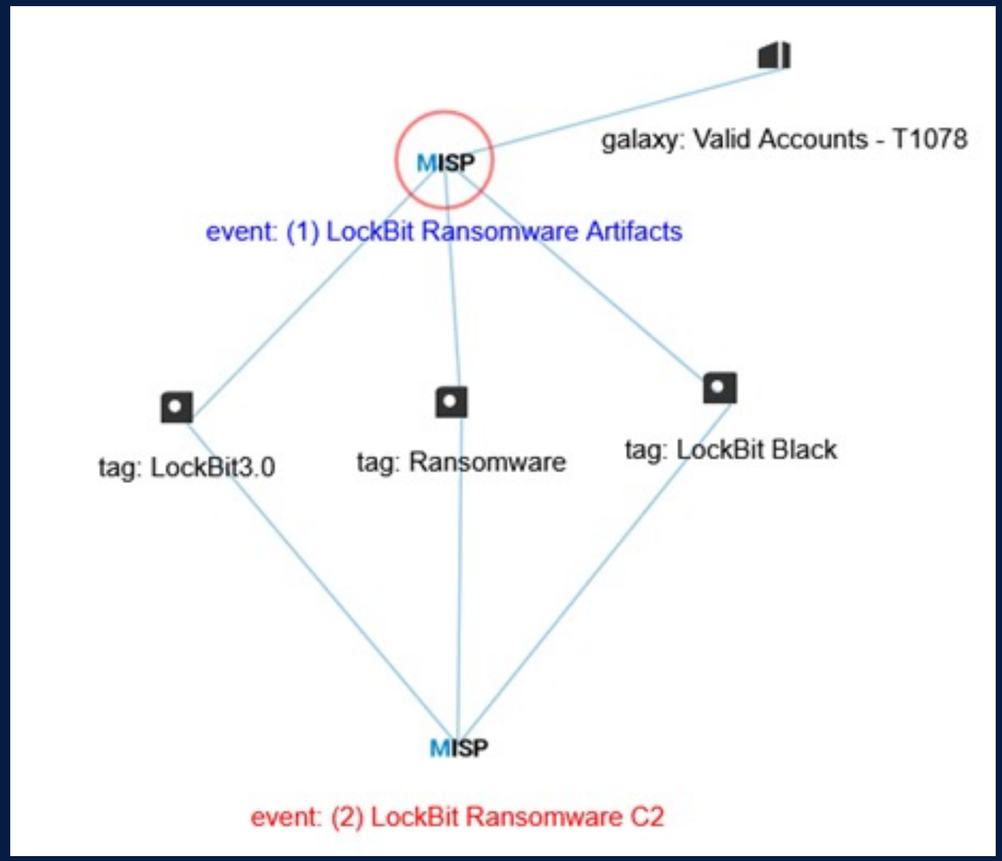
Tool	Intended Use	Repurposed Use by LockBit Affiliates	MITRE ATT&CK ID
7-zip	Compresses files into an archive.	Compresses data to avoid detection before exfiltration.	T1562 Impair Defenses
AdFind	Searches Active Directory (AD) and gathers information.	Gathers AD information used to exploit a victim's network, escalate privileges, and facilitate lateral movement.	S0552

TLP:CLEAR

Input no MISP

LockBit Ransomware C2

Event ID	2
UUID	4126722d-9d67-49bf-adb6-5903e1539a4d
Creator org	CTI Research
Owner org	CTI Research
Creator user	caique.barqueta@cti.research
Protected Event (experimental)	Event is in unprotected mode.
Tags	LockBit Black x LockBit3.0 x Ransomware x
Date	2023-05-27
Threat Level	High
Analysis	Completed
Distribution	All communities
Published	No
#Attributes	7 (0 Objects)
First recorded change	2023-05-27 00:45:46
Last change	2023-05-27 00:45:46
Modification map	
Sightings	0 (0) - restricted to own organisation only.



TLP:CLEAR

Input no MISP

mitre-pre-attack | **mitre-attack** | mitre-mobile-attack

0 — 1 — Show all

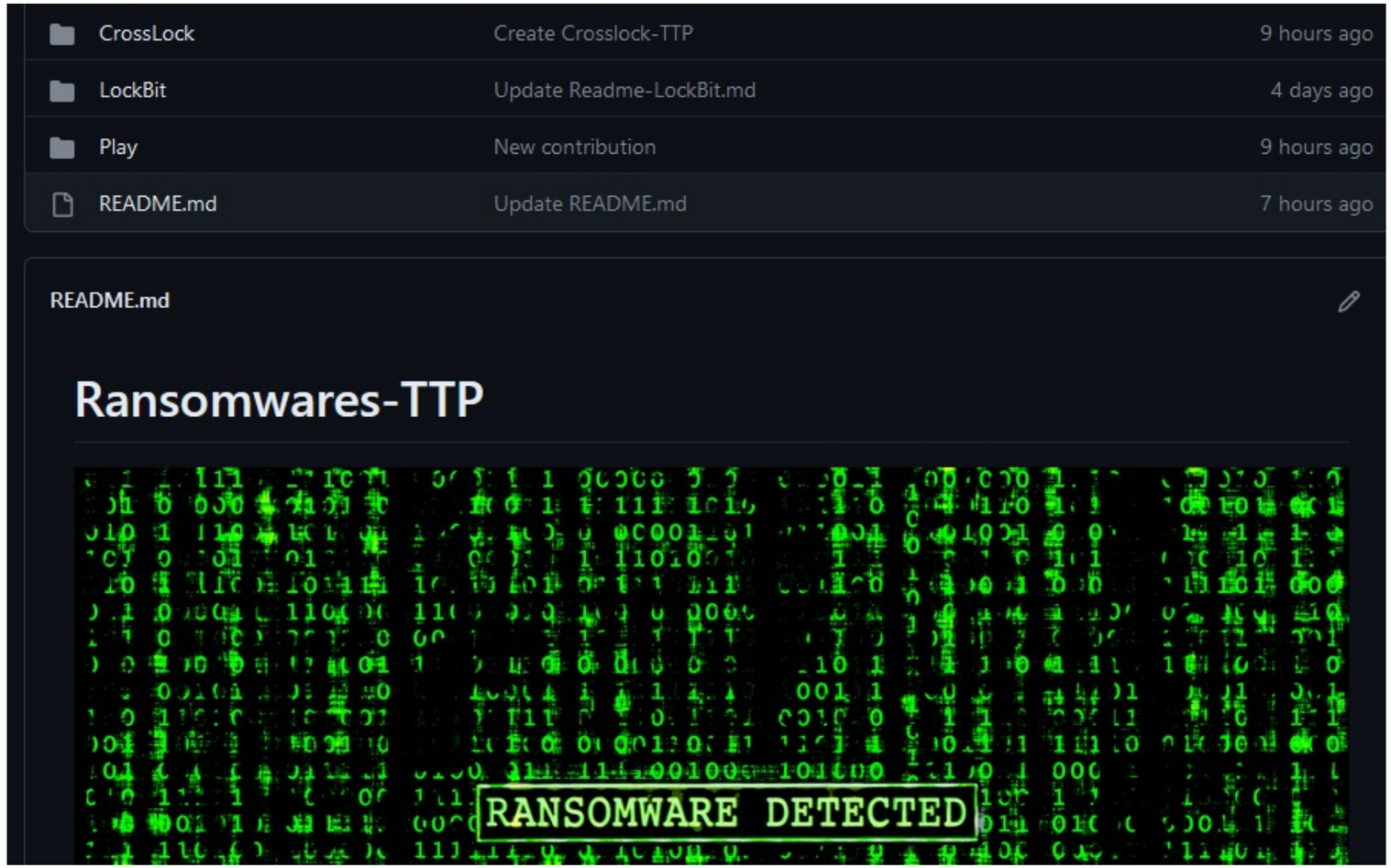
Reconnaissance	Resource development	Initial access	Execution	Persistence	Privilege escalation	Defense evasion	Credential access	Discovery	Lateral movement	Collection	Command and control	Exfiltration	Impact
Active Scanning	Acquire Infrastructure	Drive-by Compromise	Exploitation for Client Execution	Boot or Logon Autostart Execution	Boot or Logon Autostart Execution	Execution Guardrails	LSASS Memory	Network Service Discovery	Remote Desktop Protocol	ARP Cache Poisoning	Application Layer Protocol	Exfiltration Over Web Service	Data Destruction
Business Relationships	Botnet	Exploit Public-Facing Application	Software Deployment Tools	External Remote Services	Valid Accounts	Indicator Removal	OS Credential Dumping	System Information Discovery	Software Deployment Tools	Adversary-in-the-Middle	Protocol Tunneling	Exfiltration to Cloud Storage	Data Encrypted for Impact
CDNs	Botnet	External Remote Services	AppleScript	Valid Accounts	Abuse Elevation Control Mechanism	Obfuscated Files or Information	/etc/passwd and /etc/shadow	System Language Discovery	Application Access Token	Archive Collected Data	Asymmetric Cryptography	Automated Exfiltration	Defacement
Client Configurations	Cloud Accounts	Phishing	AppleScript	Accessibility Features	Access Token Manipulation	Valid Accounts	ARP Cache Poisoning	System Location Discovery	Application Access Token	Archive via Custom Method	Bidirectional Communication	Data Compressed	Inhibit System Recovery
Code Repositories	Cloud Accounts	Valid Accounts	At (Linux)	Accessibility Features	Accessibility Features	Abuse Elevation Control Mechanism	AS-REP Roasting	Account Discovery	Application Deployment Software	Archive via Library	Commonly Used Port	Data Encrypted	Service Stop
Credentials	Code Signing Certificates	Cloud Accounts	At	Account Manipulation	Accessibility Features	Access Token Manipulation	Adversary-in-the-Middle	Application Window Discovery	Component Object Model and Distributed COM	Archive via Utility	Communication Through Removable Media	Data Transfer Size Limits	Account Access Removal
DNS	Code Signing Certificates	Compromise Hardware Supply Chain	CMSTP	Active Setup	Active Setup	Application Access Token	Bash History	Browser Bookmark Discovery	Distributed Component Object Model	Audio Capture	Custom Command and Control Protocol	Exfiltration Over Alternative Protocol	Application Exhaustion Flood
DNS/Passive DNS	Compromise	Compromise	Command and	Add-ins	AppCert DLLs	Application Access	Bash History	Cloud Account	Exploitation of	Automated	Custom	Exfiltration Over	Application or

TLP:CLEAR

Repositório

Repositório criado para compartilhar TTPs públicas de atores de ameaças.

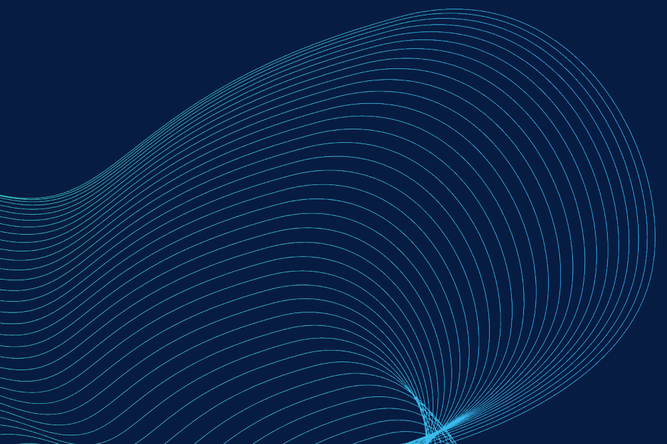
Será alimentado conforme o tempo e também aceita contribuições.



TLP:CLEAR

Feeds Open-Source

Poderá ser realizada o ***Hunting*** em feeds open-sources para localizar dados e informações do ator de ameaça, neste caso "LockBit"



MALWARE bazaar by ABUSE.ch

malwares.com

MALSHARE

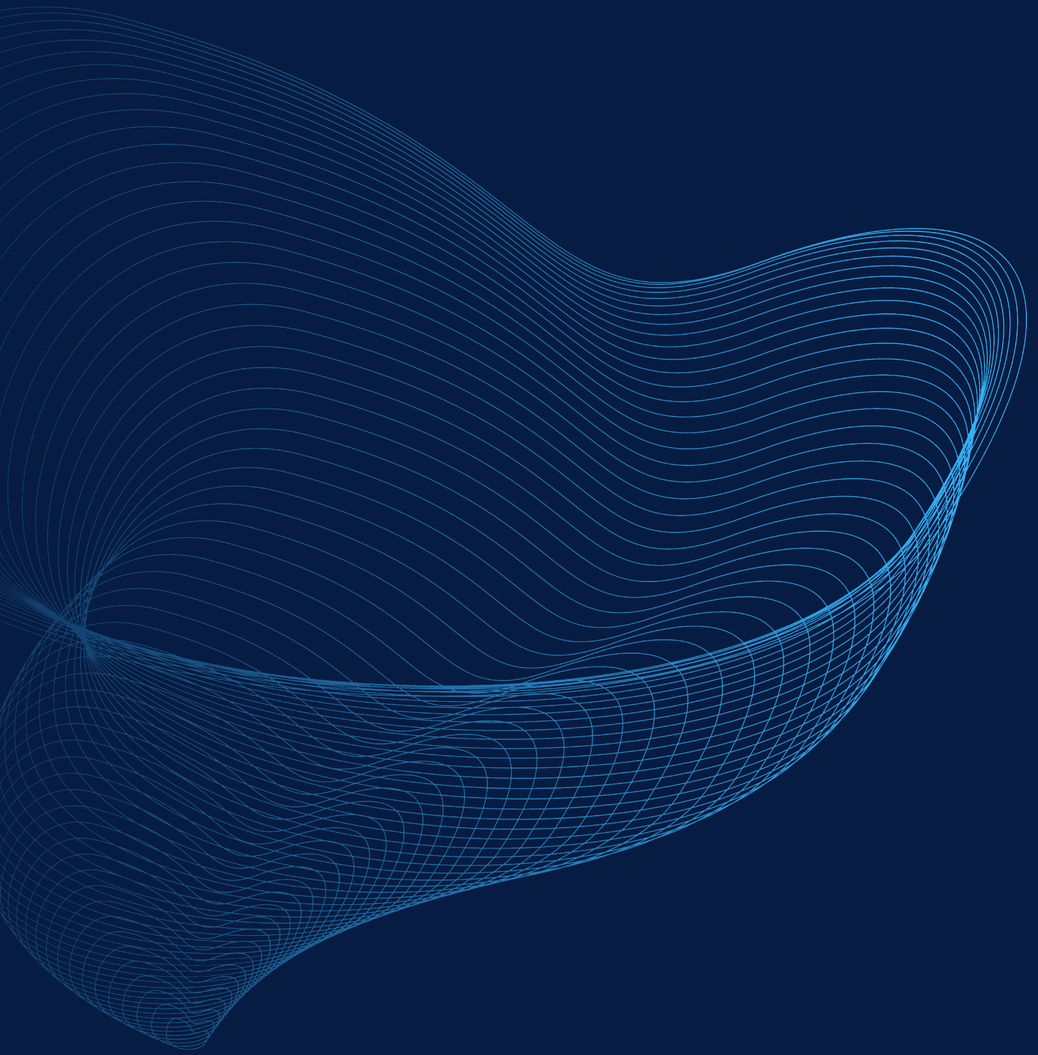
KOODOUS

INQUEST LABS

phishunt.io

VIRUSTOTAL

TLP:CLEAR



Automação de Coleta em Fontes *Abertas*

TLP:CLEAR

Bruno Odon

Bruno Odon

Especialista em Cyber Threat Intelligence (ISH)

Desenvolvedor (Backend)

Entusiasta Linux, Elastic Stack e MISP

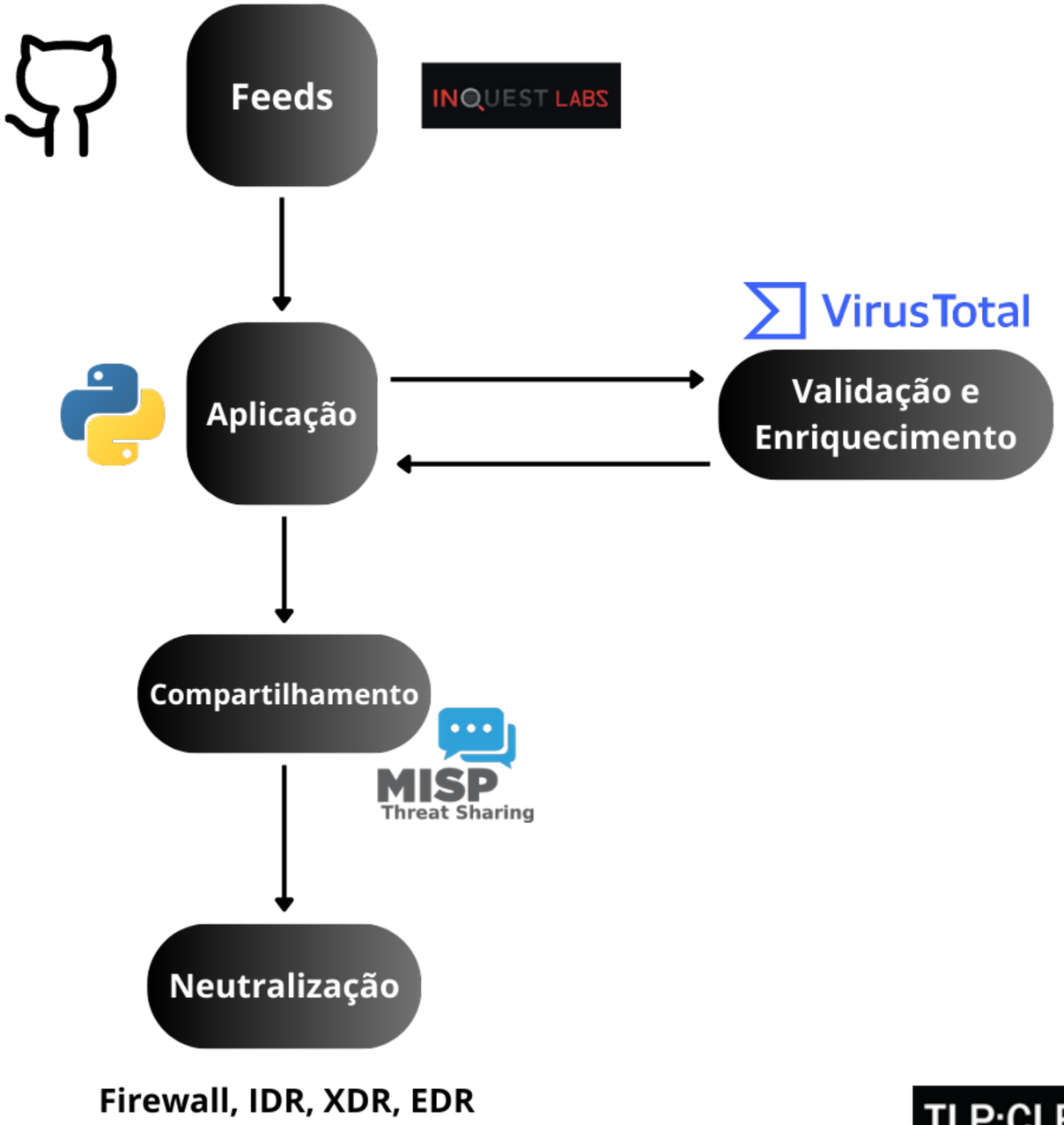
Pós-graduando em Ciência de Dados & Analytics (PUC-Rio)

CEH Hall of Fame 2023

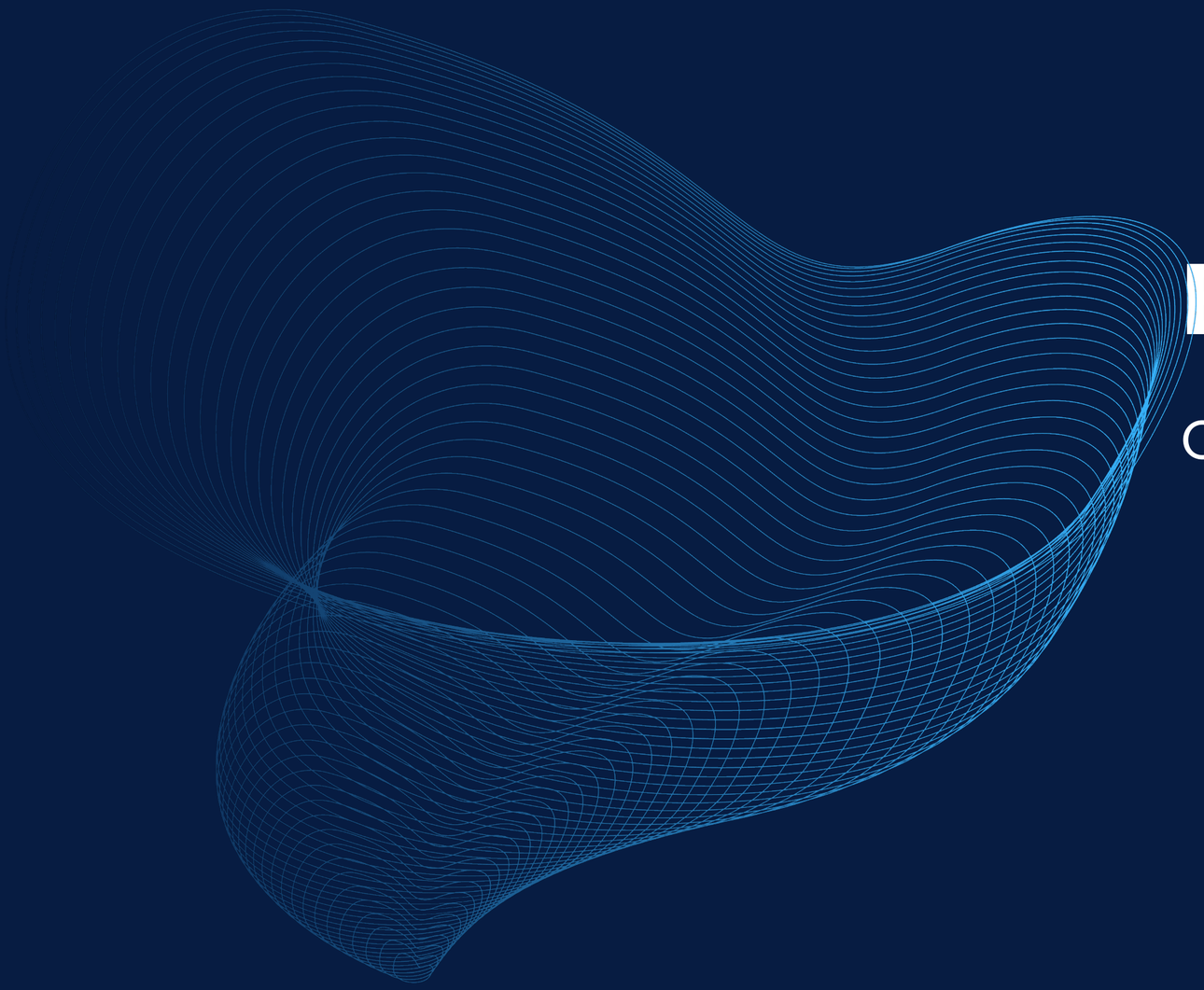
TLP:CLEAR

Automação de Threat Hunting ....por quê?

- ✓ Ganhar tempo na coleta das ameaças
- ✓ Enriquecer com outras fontes
- ✓ Integrar com plataformas de validação, evitando falso-positivo
- ✓ Integrar com plataformas de defesa
- ✓ Tornar recorrente todo o processo



TLP:CLEAR



Exemplo de Caso

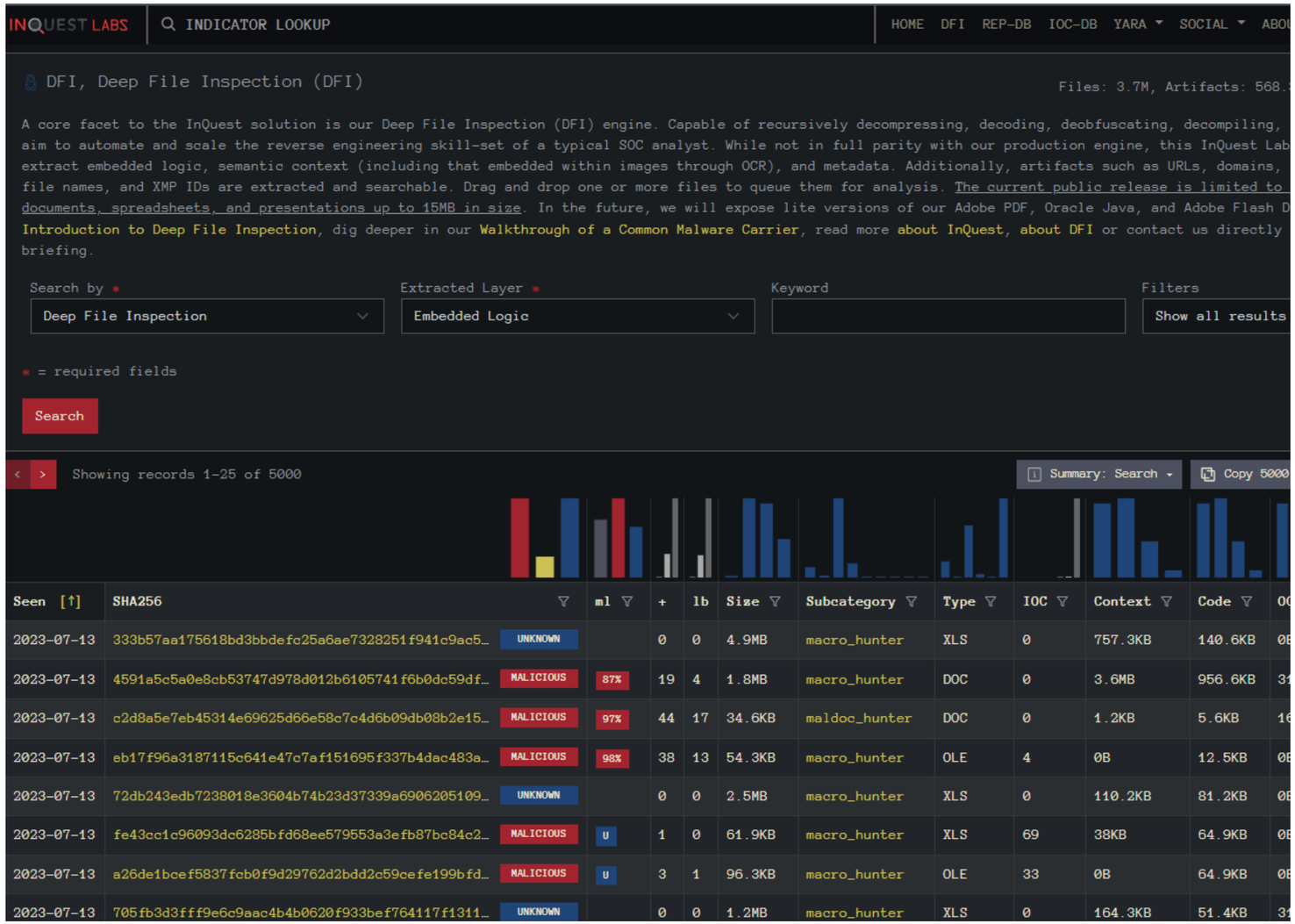
Coleta de Hashes do projeto Inquest

INQUEST LABS

TLP:CLEAR

Inquest

Disponibiliza, diariamente, hashes de arquivos que foram analisados pelos pesquisadores do projeto. Eles recebem os rótulos **‘SUSPICIOUS’**, **‘MALICIOUS’** ou **‘UNKNOWN’** (quando não é possível determinar se é relacionado a algum tipo de ameaça ou não).



TLP:CLEAR

Inquest

É disponibilizada também, gratuitamente, uma API REST para consulta desses IoC e retorno em formato JSON, o que facilita muito a integração com outras plataformas.


```
{
  'analysis_completed': True,
  'classification': 'MALICIOUS',
  'collections': [
  ],
  'downloadable': True,
  'file_type': 'DOC',
  'first_seen': '2023-07-12T22:42:58',
  'image': False,
  'inquest_alerts': [3],
  'inquest_ml_score': 0,
  'last_inquest_featext': '2023-07-12T22:43:32',
  'len_code': 19149,
  'len_context': 20,
  'len_metadata': 111,
  'len_ocr': 0,
  'mime_type': 'application/vnd.openxmlformats-
officedocument.wordprocessingml.document',
  'num_iocs': 54,
  'sha256': '92aa6a836849f4d774ef8367eb73faf003b7abb189122505e18554497fae3f2a',
  'size': 48201,
  'subcategory': 'macro_hunter',
  'subcategory_url': 'https://github.com/InQuest/yara-
rules/blob/master/labs.inquest.net/macro_hunter.rule',
  'tags': [
  ],
  'vt_positives': 42,
  'vt_weight': 14.100000381469727
}
```


TLP:CLEAR

Inquest

Script de integração (linguagem Python)

- ✓ Importa as bibliotecas necessárias, dentre elas, PyMISP e virus-total-apis;
- ✓ Define variáveis importantes, como as URLs do MISP, Inquest e VT e as API-Keys correspondentes;
- ✓ Define os parâmetros para a criação do evento no MISP.

```
import requests
import json
import hashlib
import sys
from virus_total_apis import PublicApi as VirusTotalPublicApi
from pymisp import MISPEvent, MISPObject, PyMISP, ExpandedPyMISP, MISPSharingGroup
import requests
import sys
import time
import datetime
import json
import os
vt_api_key = '[redacted]'
today=str(datetime.date.today())
url_hash = "https://labs.inquest.net/api/dfi/list"
headers_hash = {'Authorization': 'Basic [redacted]'}
r_hash = requests.get(url_hash, headers_hash)
r_json = json.loads(r_hash.text)
misp_url = "https://[redacted]"
key = '[redacted]'
misp_verifycert = True
misp = ExpandedPyMISP(misp_url, key, misp_verifycert)
event = MISPEvent()
event.info = "Análise de Malware - Coleta de Fontes Abertas - "+today+""
event.analysis = "2"
event.published = True
event.distribution = "3"
#event.sharing_group_id = "3"
event.threat_level_id = "1"
event.add_tag('tlp:clear')
event.add_tag('Malware')
```

TLP:CLEAR

Inquest

Script de integração (linguagem Python)

- ✓ Percorre o documento JSON de cada resultado do Inquest;
- ✓ Caso o *hash* seja classificado como ‘Malicious’, ele será verificado pela API do Virus Total;
- ✓ Cada atributo será inserido na lista de atributos do evento do MISP;
- ✓ O evento será criado.

```
try:
  for i in r_json['data']:
    if i['classification'] in 'MALICIOUS':
      filehash = str(i['sha256'])
      vt_positives =  int(i['vt_positives'])
      vt = VirusTotalPublicApi(vt_api_key)
      response = vt.get_file_report(filehash)
      json_doc = json.dumps(response, sort_keys=False, indent=1)
      print(json_doc)
      time.sleep(15)
      try:
        for x in json.loads(json_doc)['results']['scans']:
          if x == 'Microsoft' or x == 'TrendMicro' or x == 'Kaspersky':
            threat_name = json.loads(json_doc)['results']['scans'][''+x+'']['result']
            if threat_name != 'None':
              print(threat_name)
              event.add_attribute('sha256', str(i['sha256']), disable_correlation=True, to_ids=False, comment='Filetype:'
              event.add_attribute_tag(""+str(x)+":"+str(threat_name)+"", str(i['sha256']))
      except:
        print('Não tem resultado do VT')
except:
  print('Não tem resultado do inquest')
event = misp.add_event(event)
```

TLP:CLEAR

Inquest

Ao lado, segue o documento JSON do **Virus Total**, que vem com as análises sobre cada artefato.

Escolhemos 3 plataformas para validação dos hashes: **Microsoft Defender, Karspersky e Tend Micro**.


```
"Microsoft": {
 "detected": true,
 "version": "1.1.23060.1005",
 "result": "TrojanDownloader:O97M/Emotet.ARJ!MTB",
 "update": "20230723"
},
"Cynet": {
 "detected": true,
 "version": "4.0.0.27",
 "result": "Malicious (score: 99)",
 "update": "20230723"
},
"AhnLab-V3": {
 "detected": true,
 "version": "3.23.3.10396",
 "result": "Downloader/DOC.Emotet.S1294",
 "update": "20230723"
},
"Acronis": {
 "detected": true,
 "version": "1.2.0.114",
 "result": "suspicious",
 "update": "20230219"
},
"VBA32": {
 "detected": false,
 "version": "5.0.0",
 "result": null,
 "update": "20230721"
```


TLP:CLEAR

Aplicação

Por fim, o evento é compartilhado com todas as comunidades via MISP.

Análise de Malware - Coleta de Fontes Abertas - 2023-07-14

Event ID	42
UUID	553f3fbe-aaa7-472b-b1d2-d89d47f02a59
Creator org	CTI Research
Owner org	CTI Research
Creator user	api@cti.research
Protected Event (experimental)	Event is in unprotected mode.
Tags	tlp:clear x Malware x
Date	2023-07-14
Threat Level	High
Analysis	Completed
Distribution	All communities
Published	Yes 2023-07-14 05:04:51
#Attributes	4 (0 Objects)
First recorded change	2023-07-14 05:04:51
Last change	2023-07-14 05:04:51
Modification map	
Sightings	0 (0) - restricted to own organisation only.

sha256	0865692d9bf207c3f14942b54c831997f393133ebb2c7c4ef22da36fe68b622e	Kaspersky:HEUR:Trojan-Downloader.Script.Generic x
sha256	a9ab46ba9e83434fe4c5e976e59af78e7ac233ddf11aa64899b124d2b3c165bd	Kaspersky:HEUR:Trojan-Downloader.Script.Generic x
sha256	51642c833aff521d8227c93431b66c57e8bbfa0ee80518e3be6c5acf0c116c11	Kaspersky:HEUR:Trojan.Script.Agent.gen x TrendMicro:W2KM_POWLOAD.SME x Microsoft:Trojan:X97M/LionWolf.A x

TLP:CLEAR

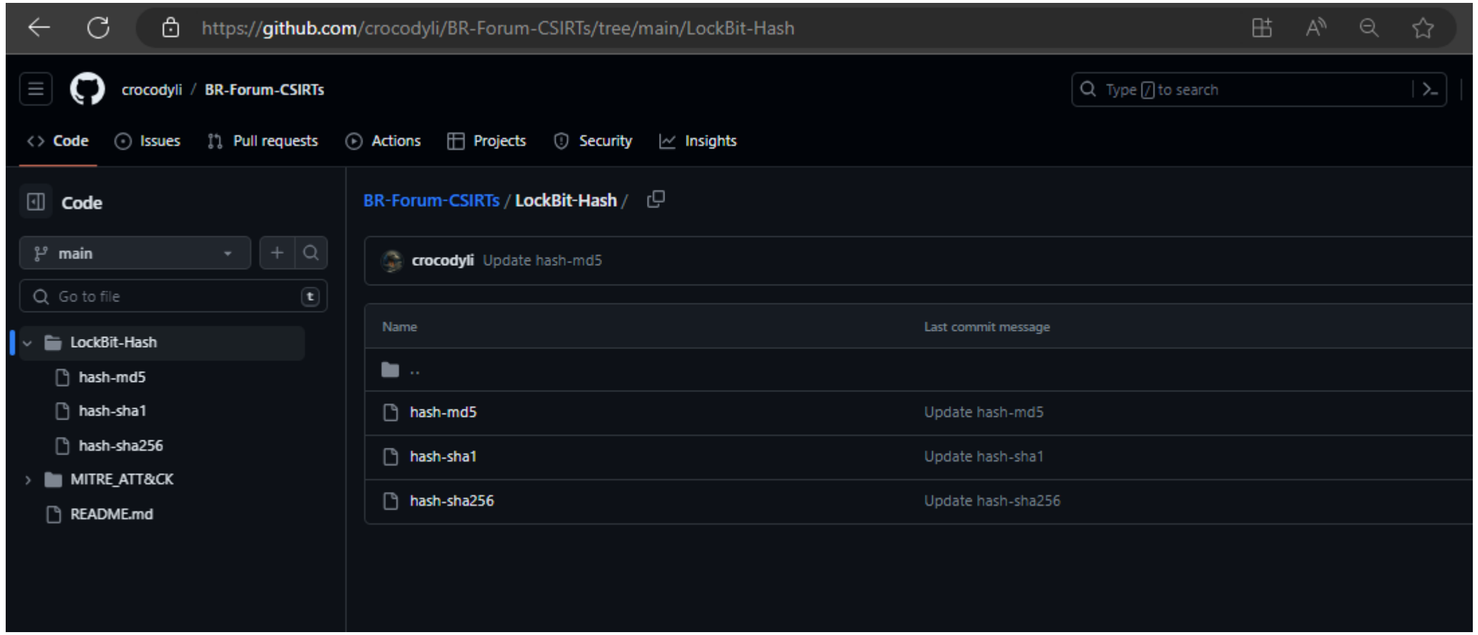
Exemplo de Caso

Coleta de Hashes da pesquisa do Caique Barqueta

TLP:CLEAR

GitHub

Neste caso, a coleta é feita de repositório público, onde o Caique inseriu os artefatos de suas análises sobre o Ransomware LockBit 3.0.



TLP:CLEAR

GitHub

Script de integração (linguagem Python)

- ✓ Define as variáveis de URLs e APIKeys;
- ✓ Define os parâmetros do evento que será inserido via PyMISP;
- ✓ Define os datasets dos hashes compartilhados pelo Caíque.

```
vt_api_key = '<VT_APIKEY>'
today=str(datetime.date.today())
misp_url = "<MISP_URL>"
key = '<MISP_KEY>'
misp_verifycert = False
misp = ExpandedPyMISP(misp_url, key, misp_verifycert)
event = MISPEvent()
event.info = "Lockbit 3.0 - File Hashes - "+today+""
event.analysis = "2"
event.published = True
event.distribution = "3"
#event.sharing_group_id = "3"
event.threat_level_id = "1"
event.add_tag('tlp:clear')
event.add_tag('Malware')
event.add_tag('Lockbit3.0')
#endereços dos documentos com hashes maliciosos
url_md5 = 'https://raw.githubusercontent.com/crocodyli/BR-Forum-CSIRTs/main/LockBit-Hash/hash-md5'
url_sha1 = 'https://raw.githubusercontent.com/crocodyli/BR-Forum-CSIRTs/main/LockBit-Hash/hash-sha1'
url_sha256 = 'https://raw.githubusercontent.com/crocodyli/BR-Forum-CSIRTs/main/LockBit-Hash/hash-sha256'
#lendo o conteúdo dos arquivos com a lib 'pandas'
names=['hash']
r_md5 = pd.read_csv(url_md5, names=names)
r_sha1 = pd.read_csv(url_sha1, names=names)
r_sha256 = pd.read_csv(url_sha256, names=names)
```

TLP:CLEAR

GitHub

Script de integração (linguagem Python)

- ✓ Usa a API do Virus Total para revalidar e analisar cada tipo de hash do dataset;
- ✓ Caso o resultado tenha sido gerado por uma das plataformas que escolhemos, o hash será publicado no evento do MISP.

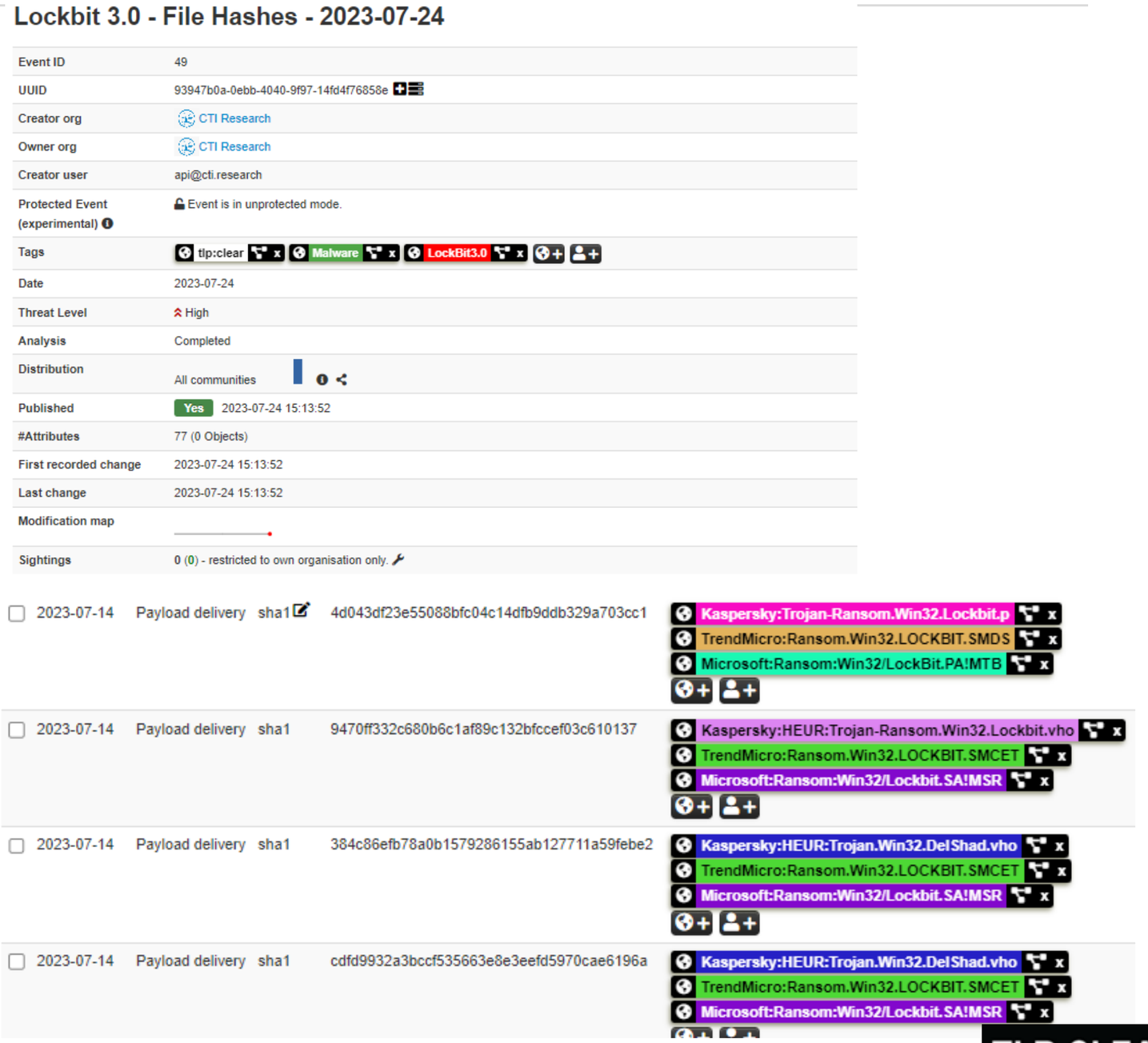
```
for a in r_sha256['hash']:
  try:
    vt = VirusTotalPublicApi(vt_api_key)
    response_sha256 = vt.get_file_report(a)
    json_doc_sha256 = json.dumps(response_sha256, sort_keys=False, indent=1)
    try:
    #inserindo os MD5
      for x in json.loads(json_doc_sha256)['results']['scans']:
        if x == 'Microsoft' or x == 'TrendMicro' or x == 'Kaspersky':
          threat_name = json.loads(json_doc_sha256)['results']['scans'][''+x+'']['result']
          if 'None' not in threat_name:
            print(threat_name)
            event.add_attribute('sha256', str(a), disable_correlation=True, to_ids=False)
            event.add_attribute_tag(""+str(x)+":"+str(threat_name)+"", str(a))
            time.sleep(15)
    except:
      print('Não tem resultado do VT')

  except:
    print('NONE')
event = misp.add_event(event)
```

TLP:CLEAR

GitHub

Por fim, os IoC são compartilhados com todas as comunidades via MISP.

É importante dar o máximo de detalhes possíveis sobre a ameaça, bem como classifica-la de forma correta.



TLP:CLEAR

Links de referências

- ✓ Repositório do Bruno Odon para Automação
- ✓ Repositório de TTPs e Tools – Fórum CSIRTs
- ✓ Repositório de TTPs de Ransomwares – Caique Barqueta
- ✓ MISP – Documentação
- ✓ PyMISP – Documentação
- ✓ VírusTotal API v3
- ✓ InQuest Labs API v.1.0.2

TLP:CLEAR

Dúvidas?

Bruno

Caique

TLP:CLEAR