

Análise Automatizada de Indicadores de Ameaça Cibernética Utilizando Técnicas de Processamento de Linguagem Natural

Alunos:

Jonas Aguiar Junior


Keli Tauana Prass Ruppenthal

Leonardo de Jesus Diz Conde



PUC Minas

De onde veio a ideia

**Rômulo Rocha** · 2º
Information Security Engineering Manager at Nubank | Seasoned...
9 m · Editado ·

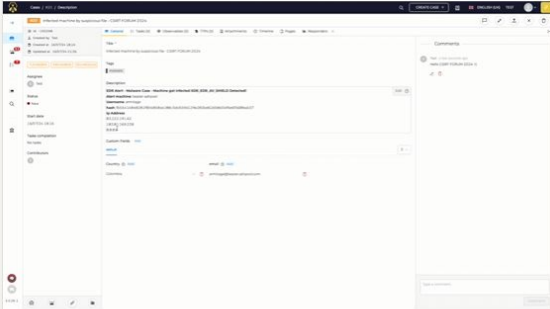
+ Seguir · ···

Hey LinkedIn, segue o video da POC apresentada na minha apresentação. (Para quem não assistiu, o vídeo é uma POC simples, com automação de enriquecimento de observáveis (IP) usando linguagem natural para adicionar contexto ao caso de um incidente.)
Foram utilizados TheHive+Tines.io (community) +AbusePDB e GPT 3.5, o custo foi de 105 cents na API da OpenAI e foram gastos 2 horas pra fazer.)

Slide Deck:
<https://lnkd.in/d76uWXjY>
- Adicionei alguns comentários nos slides para ajudar no entendimento.
- Adicionei também na referência o link para o JSON do workflow da POC para que você possa utilizar e experimentar como desejar.

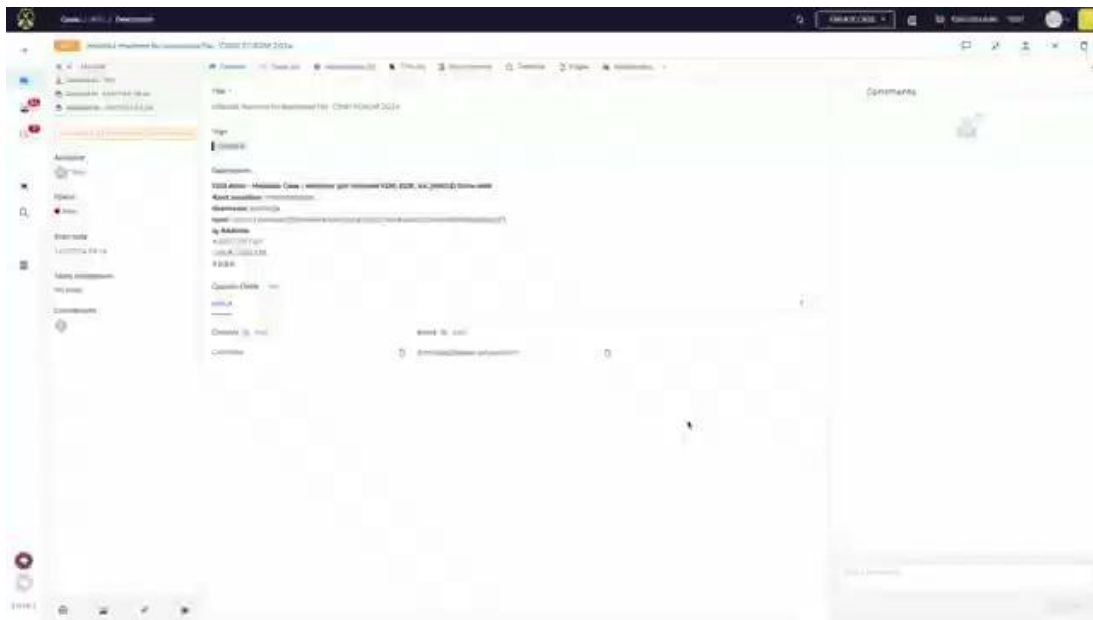
Ainda estou aqui pela conferência e podem ficar a vontade para me procurar caso tenham alguma dúvida.

Espero que gostem! 🚀
#soar #automation #csirt #casemanagement #hive #tines #torq #shuffler #workflows #soc



92

4 comentários · 3 compartilhamentos



O problema

- Crescente volume e complexidade dos incidentes de segurança cibernética;
- Trabalho do analista de segurança -> análise de grande volume de dados/alertas -> identificar ameaças de forma manual;
- **TheHive**: gestão de incidentes de segurança cibernética;
- **Cortex**: plataforma de análise que se integra ao TheHive, permite automação de ações para verificação de ameaças;
- **VirusTotal**: permite a análise de arquivos e URLs suspeitos e exporta um relatório sobre o item em questão

Objetivo

- Sistema automatizado que integre TheHive, Virus Total e NLP para analisar indicadores de ameaças e, no retorno, inserir alertas na plataforma, reduzindo o esforço manual do analista.

Etapas

- 1. Recepção de alerta no TheHive com observáveis relevantes (como um IP, domínio, hashes etc);
- 2. Para cada observável, utilizar a API do VirusTotal para obter o relatório completo sobre esse indicador;
- 3. Analisar o texto desse relatório para identificar sinais de risco com técnicas de NLP supervisionado e então classificar malicioso, suspeito ou benigno;
- 4. Gerar um comentário automatizado sobre a análise do indicador;
- 5. Inserir esse comentário automaticamente no alerta correspondente dentro do TheHive, por meio da API.

Dados utilizados

- Arquivos JSON gerados no TheHive:

```

1 {
2   "summary": {
3     "taxonomies": [
4       {
5         "level": "malicious",
6         "namespace": "VT",
7         "predicate": "GetReport",
8         "value": "2/94"
9       },
10      {
11        "level": "malicious",
12        "namespace": "VT",
13        "predicate": "GetReport",
14        "value": "200 resolution(s)"
15      }
16    ]
17  },
18  "full": {
19    "type": "ip_address",
20    "attributes": {
21      "reputation": -5,
22      "whois": "Amazon.com, Inc. AMAZO-4 (NET-44-192-0-0-1) 44.192.0.0 - 44.255.255.255\nAmazon.com, Inc. AMAZO-ZPDX (NET-44-224-0-0-1) 44.224.0.0 - 44.255.255.255\n",
23      "last_analysis_stats": {
24        "malicious": 2,
25        "suspicious": 0,
26        "undetected": 32,
27        "harmless": 60,
28        "timeout": 0
29      },
30      "network": "44.224.0.0/11",
31      "regional_internet_registry": "ARIN",
32      "continent": "NA",
33      "last_modification_date": 1747231291,
34      "last_analysis_results": {
35        "Acronis": {
36          "method": "blacklist",

```



```

1 {
2   "summary": {
3     "taxonomies": [
4       {
5         "level": "malicious",
6         "namespace": "VT",
7         "predicate": "GetReport",
8         "value": "1/94"
9       },
10      {
11        "level": "malicious",
12        "namespace": "VT",
13        "predicate": "GetReport",
14        "value": "200 resolution(s)"
15      }
16    ]
17  },
18  "full": {
19    "type": "ip_address",
20    "attributes": {
21      "whois": "NetRange: 104.16.0.0 - 104.31.255.255\nCIDR: 104.16.0.0/12\nNetName: CLOUDFLARENET\nNetHandle: NET-104-16-0-0-1\nParent: NET104 (NET-104-0-0-0)\nNetType
22      "tags": [],
23      "network": "104.20.0.0/15",
24      "last_analysis_stats": {
25        "malicious": 1,
26        "suspicious": 0,
27        "undetected": 31,
28        "harmless": 62,
29        "timeout": 0
30      },
31      "reputation": 0,
32      "last_modification_date": 1745599651,
33      "last_analysis_date": 1735388288,
34      "asn": 13335,
35      "as_owner": "CLOUDFLARENET",
36      "total_votes": {
37        "harmless": 0,

```

```

{
  "full": {
    "attributes": {
      "last_analysis_results": {
        "Acronis": {
          "method": "blacklist",
          "engine_name": "Acronis",
          "category": "harmless",
          "result": "clean"
        },
        "0xSI_f33d": {
          "method": "blacklist",
          "engine_name": "0xSI_f33d",
          "category": "undetected",
          "result": "unrated"
        },
        "Abusix": {
          "method": "blacklist",
          "engine_name": "Abusix",
          "category": "harmless",
          "result": "clean"
        },
        "ADMINUSLabs": {
          "method": "blacklist",
          "engine_name": "ADMINUSLabs",
          "category": "harmless",
          "result": "clean"
        },
        "Axur": {
          "method": "blacklist",
          "engine_name": "Axur",
          "category": "undetected",
          "result": "unrated"
        },
        "Criminal IP": {
          "method": "blacklist",
          "engine_name": "Criminal IP",
          "category": "malicious",
          "result": "malicious"
        },
        "AILabs (MONITORAPP)": {
          "method": "blacklist",
          "engine_name": "AILabs (MONITORAPP)",
          "category": "harmless",
          "result": "clean"
        }
      }
    }
  }
}

```

Bibliotecas utilizadas

```
# Imports
import json
from pathlib import Path
from typing import List, Tuple
import joblib
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import nltk
import spacy
import string
import os
from collections import Counter
```

Algumas definições:

```
# Diretórios
ASSETS_DIR = Path("./assets")
MODEL_DIR = Path("./models")
MODEL_DIR.mkdir(exist_ok=True)

# Palavras-chave
KEYWORDS_MALICIOUS = {"malicious", "malware", "trojan", "phishing", "botnet", "miner"}
KEYWORDS_SUSPICIOUS = {"suspicious", "spam", "unrated", "risk", "unknown"}
RULE_WEIGHTS = {**{k: 1 for k in KEYWORDS_SUSPICIOUS}, **{k: 2 for k in KEYWORDS_MALICIOUS}}

# Modelos
PIPELINES = {
    "bow_nb": Pipeline([
        ("vect", CountVectorizer()),
        ("clf", MultinomialNB()),
    ]),
    "tfidf_svc": Pipeline([
        ("tfidf", TfidfVectorizer()),
        ("clf", LinearSVC()),
    ]),
}
```

1) Converte texto em vetores
(Bag-of-Words) e Classificador Naive Bayes

2) Converte texto em vetores TF-IDF
e Classificador SVM Linear

Carga de modelos:

```
# NLP
nlp = spacy.load("en_core_web_sm")
nltk.download('stopwords')
stopwords = nltk.corpus.stopwords.words('english')
```

Modelo de NLP para tokenização e Lista de stopwords. Tokenização e limpeza de texto dependem disso!

Técnicas NLP utilizadas

- Pré-processamento de Texto:
 - Tokenização (dividir texto em palavras).
 - Normalização (converter para minúsculas).
 - Remoção de pontuação e stopwords.
 - Reconstrução do texto limpo.

Técnicas NLP utilizadas

```
pontuacao_lista = list(string.punctuation.strip()) + ['...', '“', '”']

# Limpeza de texto
def clean_text(text: str) -> str:
    tokens = nlp(text)
    tokens = [str(t).lower() for t in tokens if str(t) not in pontuacao_lista]
    tokens = [str(t) for t in tokens if str(t) not in stopwords]
    return " ".join(tokens)
```

Transforma texto bruto em uma representação padronizada para análise.

Técnicas NLP utilizadas

- Classificação Baseada em Regras:
 - Aplica `clean_text()` (NLP) para limpar o conteúdo.
 - Classifica com regras baseadas em keywords.
 - A limpeza do texto é essencial para o ML entender os dados.

Técnicas NLP utilizadas

```
# Normalizador universal de relatórios
```

```
✓ def normalize_report(report: dict) -> dict:
    if "full" in report and "attributes" in report["full"]:
        return report
    if "data" in report and "attributes" in report["data"]:
        return {"full": report["data"]}
    if "attributes" in report:
        return {"full": report}
    raise ValueError("Formato de relatório não reconhecido.")
```

```
# Carregamento dos relatórios
```

```
✓ def load_reports() -> Tuple[List[str], List[str]]:
    texts, labels = [], []
    for file in ASSETS_DIR.glob("*.txt"):
        try:
            data = json.loads(file.read_text())
            data = normalize_report(data)

            last_analysis = data['full'].get('attributes', {}).get('last_analysis_results', {})
            malicious_count = sum(
                1 for val in last_analysis.values()
                if val.get('category') in KEYWORDS_MALICIOUS or val.get('result') in KEYWORDS_MALICIOUS
            )

            label = (
                "malicioso" if malicious_count >= 5 else
                "suspeito" if 1 < malicious_count < 5 else
                "benigno"
            )

            raw = json.dumps(data)
            texts.append(clean_text(raw))
            labels.append(label)
        except Exception as e:
            print(f"[ERRO] ao processar {file.name}: {e}")
    return texts, labels
```

Técnicas NLP utilizadas

- Texto já pré-processado (por `clean_text()`).
- Vetorização (Bag-of-Words ou TF-IDF) → converte texto em números.
- Treina modelos (Naive Bayes / SVM) nos dados vetorizados.
- `CountVectorizer`/`TfidfVectorizer` são técnicas clássicas de NLP para representar texto. Dessa forma, os modelos aprendem padrões nos dados processados.

Técnicas NLP utilizadas

```
# Treinamento dos modelos
def train():
    X, y = load_reports()
    print(f"[INFO] Distribuição das classes: {Counter(y)}")

    if len(set(y)) < 2:
        print("[ERRO] Dados insuficientes: é necessário ao menos duas classes diferentes.")
        return {}

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42, stratify=y
    )

    models = {}
    for name, pipe in PIPELINES.items():
        try:
            pipe.fit(X_train, y_train)
            joblib.dump(pipe, MODEL_DIR / f"{name}.joblib")
            preds = pipe.predict(X_test)
            print(f"\n*** {name} ***")
            print(classification_report(y_test, preds, zero_division=0))
            models[name] = pipe
        except Exception as e:
            print(f"[ERRO] Falha ao treinar {name}: {e}")

    print("✅ Treinamento concluído.")
    return models
```

Técnicas NLP utilizadas

- Modelo de Machine Learning
 - Modelos de ML (bow_nb, tfidf_svc) fazem previsões.
 - **Regra baseada em keywords** (rule_based_predict) também vota.
 - O **ensemble** decide a classe final (votação majoritária).
 - **Gera explicação** baseada em estatísticas.

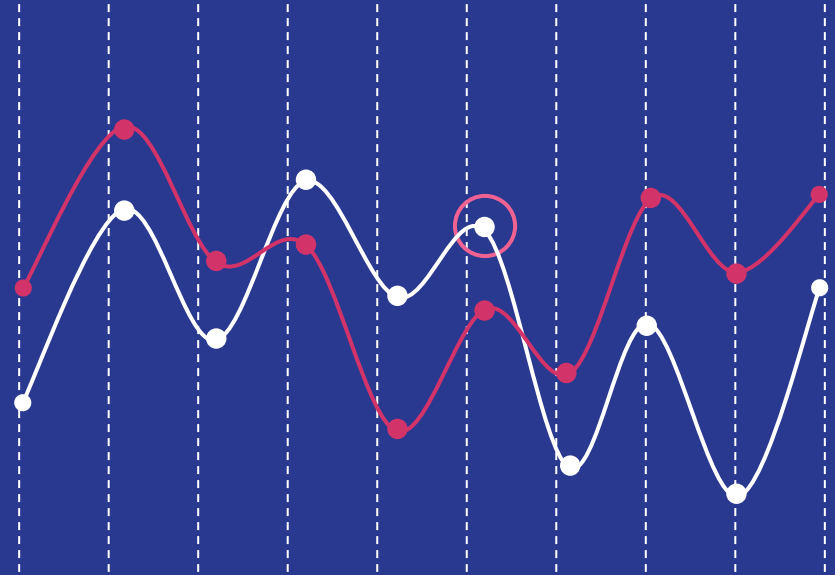
Técnicas NLP utilizadas

```
# Predição com novo modelo
def predict(path: Path, models):
    try:
        raw = json.loads(path.read_text())
        raw = normalize_report(raw)
        text = clean_text(json.dumps(raw))
        label = ensemble_predict(text, models)
        comment = generate_comment(label, raw)
        print(f"📄 {path.name} -> {label.upper()}")
        print(comment)
    except Exception as e:
        print(f"[ERRO] ao processar {path.name}: {e}")
```

Resumo

Etapa	O que faz	Técnicas de NLP usadas
1. Configuração	Define paths e keywords	-
2. Pré-processamento	Limpeza de texto (<code>clean_text</code>)	Tokenização, lowercase, remoção de stopwords
3. Carregar dados	Lê relatórios e aplica NLP	Limpeza + classificação por keywords
4. Treinar modelos	Vetoriza texto e treina ML	Bag-of-Words, TF-IDF
5. Predição	Classifica novos textos com ensemble	Combina ML + regras de NLP

Resultados



*** bow_nb ***

	precision	recall	f1-score	support
benigno	0.93	1.00	0.97	14
malicioso	0.67	1.00	0.80	2
suspeito	1.00	0.50	0.67	4
accuracy			0.90	20
macro avg	0.87	0.83	0.81	20
weighted avg	0.92	0.90	0.89	20

*** tfidf_svc ***

	precision	recall	f1-score	support
benigno	0.82	1.00	0.90	14
malicioso	0.50	0.50	0.50	2
suspeito	1.00	0.25	0.40	4
accuracy			0.80	20
macro avg	0.77	0.58	0.60	20
weighted avg	0.83	0.80	0.76	20

Indicador suspeito com 0 alertas. IP 202.131.82.244, ASN 'Cambo TechnologyISP Co.,Ltd' (KH). Monitoramento contínuo é recomendado.
 [] benigno_66-249-64-0.txt -> BENIGNO
 Indicador benigno com 63 detecções limpas. IP 66.249.64.0, ASN 'GOOGLE' (US). Sem sinais de risco atuais.
 [] benigno_13-70-0-0.txt -> BENIGNO
 Indicador benigno com 62 detecções limpas. IP 13.70.0.0, ASN 'MICROSOFT-CORP-MSN-AS-BLOCK' (HK). Sem sinais de risco atuais.
 [] malicioso_64-62-197-238.txt -> MALICIOSO
 Indicador classificado como malicioso com 11 detecções confirmadas. O IP 64.62.197.238 pertence ao ASN 'HURRICANE' (US), com reputação -2. Tags: . Recomenda do bloqueio e investigação.
 [] suspeito_103-241-67-157.txt -> SUSPEITO
 Indicador suspeito com 2 alertas. IP 103.241.67.157, ASN 'KAMATERA' (ES). Monitoramento contínuo é recomendado.
 [] benigno_13-66-0-0.txt -> BENIGNO
 Indicador benigno com 62 detecções limpas. IP 13.66.0.0, ASN 'MICROSOFT-CORP-MSN-AS-BLOCK' (US). Sem sinais de risco atuais.
 [] benigno_42-179-217-67.txt -> BENIGNO
 Indicador benigno com 0 detecções limpas. IP 42.179.217.67, ASN 'CHINA UNICOM China169 Backbone' (CN). Sem sinais de risco atuais.
 [] benigno_13-72-0-0.txt -> BENIGNO
 Indicador benigno com 0 detecções limpas. IP 13.72.0.0, ASN 'MICROSOFT-CORP-MSN-AS-BLOCK' (US). Sem sinais de risco atuais.
 [] benigno_13-76-0-0.txt -> BENIGNO
 Indicador benigno com 63 detecções limpas. IP 13.76.0.0, ASN 'MICROSOFT-CORP-MSN-AS-BLOCK' (SG). Sem sinais de risco atuais.
 [] benigno_101-32-214-229.txt -> BENIGNO
 Indicador benigno com 0 detecções limpas. IP 101.32.214.229, ASN 'Tencent Building, Kejizhongyi Avenue' (HK). Sem sinais de risco atuais.
 [] benigno_13-75-0-0.txt -> BENIGNO
 Indicador benigno com 62 detecções limpas. IP 13.75.0.0, ASN 'MICROSOFT-CORP-MSN-AS-BLOCK' (HK). Sem sinais de risco atuais.
 [] suspeito_41-141-11-43.txt -> BENIGNO
 Indicador benigno com 59 detecções limpas. IP 41.141.11.43, ASN 'MT-MPLS' (MA). Sem sinais de risco atuais.
 [] suspeito_27-112-79-160.txt -> SUSPEITO
 Indicador suspeito com 2 alertas. IP 27.112.79.160, ASN 'PT Cloud Hosting Indonesia' (ID). Monitoramento contínuo é recomendado.
 [] suspeito_103-243-242-61.txt -> SUSPEITO
 Indicador suspeito com 0 alertas. IP 103.243.242.61, ASN 'PacketFabric Japan Co., Ltd.' (JP). Monitoramento contínuo é recomendado.
 [] suspeito_201-231-83-229.txt -> BENIGNO
 Indicador benigno com 60 detecções limpas. IP 201.231.83.229, ASN 'Telecom Argentina S.A.' (AR). Sem sinais de risco atuais.
 [] benigno_13-80-0-0.txt -> BENIGNO
 Indicador benigno com 62 detecções limpas. IP 13.80.0.0, ASN 'MICROSOFT-CORP-MSN-AS-BLOCK' (NL). Sem sinais de risco atuais.
 [] benigno_13-74-0-0.txt -> BENIGNO
 Indicador benigno com 62 detecções limpas. IP 13.74.0.0, ASN 'MICROSOFT-CORP-MSN-AS-BLOCK' (IE). Sem sinais de risco atuais.
 [] benigno_13-82-0-0.txt -> BENIGNO
 Indicador benigno com 63 detecções limpas. IP 13.82.0.0, ASN 'MICROSOFT-CORP-MSN-AS-BLOCK' (US). Sem sinais de risco atuais.
 [] benigno_13-82-0-0.txt -> BENIGNO



#16 IP observado como possível atividade suspeita



Id ~122962160

Created by Leonardo

Created at 19/05/2025 09:23

SEVERITY: HIGH

TLP: AMBER

PAP: AMBER

Assignee

L Leonardo

Status

● New

General

Tasks (0)

Observables (1)

TTPs (0)

Attachments

Timeline

Pages

History

* Title

IP observado como possível atividade suspeita

Tags

Tags

Description

Foi detectada comunicação entre o ambiente interno e o IP que requer análise de reputação.

Análise PLN:

Indicador classificado como benigno com 57 mecanismos classificando como inofensivo. O IP 172.105.218.179 pertence a 'Akamai Connected Cloud' e não apresenta sinais atuais de risco.

Trabalhos Futuros

- **Expansão do Dataset:** Incorporar mais relatórios rotulados para refinar os modelos.
- **Integração em Tempo Real:** Acionar o pipeline automaticamente via webhooks do TheHive.
- **Análise Multimodal:** Combinar NLP com metadados (ex: reputação de IP) para maior precisão.
- **Feedback de Analistas:** Usar classificações manuais para ajuste contínuo (active learning).

Referências

- Jurafsky, D., & Martin, J. H. (2024). *Speech and Language Processing* (3rd ed.). Pearson.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python* (1st ed.). O'Reilly.
- ROCHA, R. SOAR Automation for CSIRT Teams. LinkedIn, 2024. Disponível em:
https://www.linkedin.com/posts/romrocha_soar-automation-csirt-activity-7224103678842986497-XfBA/. Acesso em: 28 abril. 2024.

Obrigado!



PUC Minas