AnonShield

Pseudonimização segura e escalável de dados de incidentes e vulnerabilidades em CSIRTs

`host: 100.111.20.23`  `[IP_ADDRESS_a4f2b1c3]`

Cristhian Kapelinski

Orientador: Diego Kreutz

Ciência da Computação, UNIPAMPA, Campus Alegrete

unipampa Universidade Federal do Pampa

O que é um CSIRT?

A equipe que cuida da segurança digital. Como os bombeiros, mas para ataques digitais.

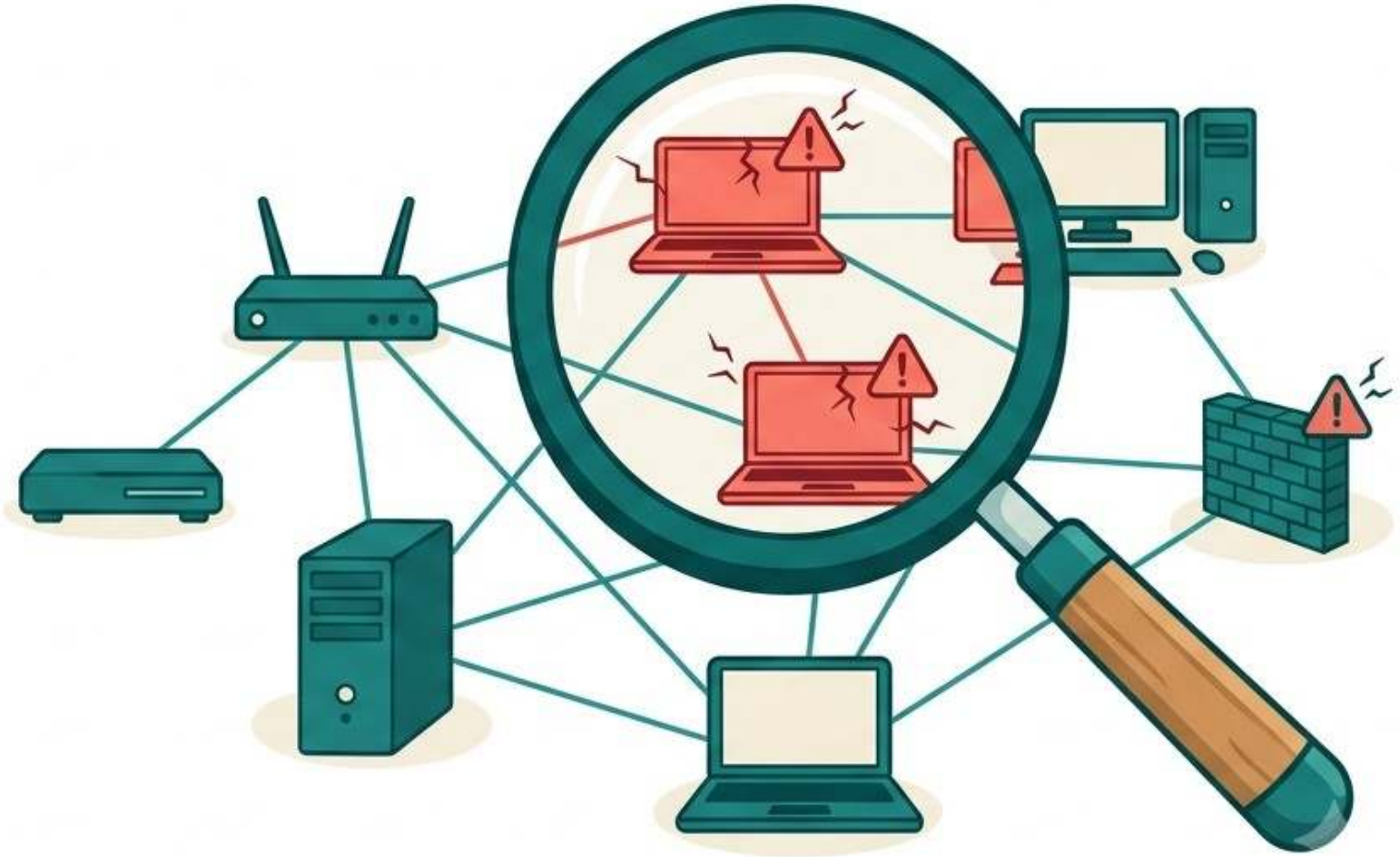
- **Recebe**
  alertas de ataques e falhas
- **Investiga**
  o que aconteceu e o impacto
- **Responde**
  contém e corrige o problema

Vulnerabilidades de rede

- **Varreduras**
  de vulnerabilidades

- **De rede, não de código**
  host, portas e serviços

- **Raio-X da rede**
  revela portas e fraquezas

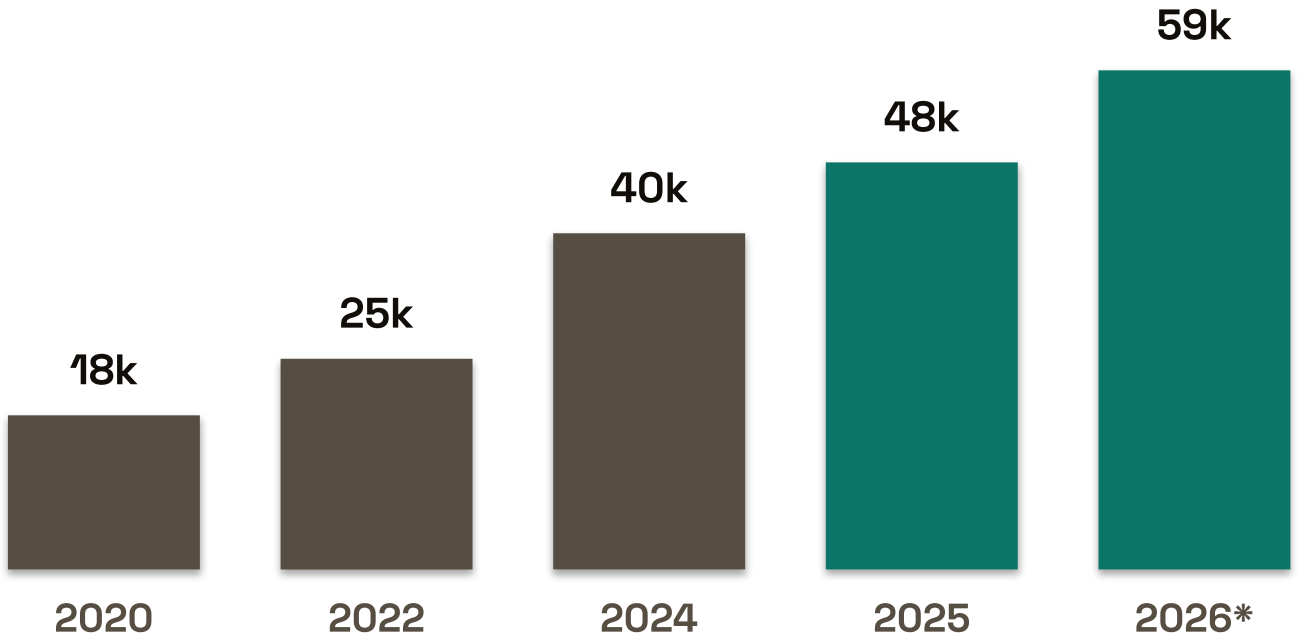


Registros de incidentes

- **Tickets de incidente**
  ataques já detectados
- **Vêm dos CSIRTs**
  cais RNP cert.br
- **Expõem a vítima**
  e-mails, nomes, IPs

A escala do problema: vulnerabilidades

O número de vulnerabilidades cresce muito mais rápido que a capacidade de analisá-las.



CVEs publicados por ano (*previsão)

NIST

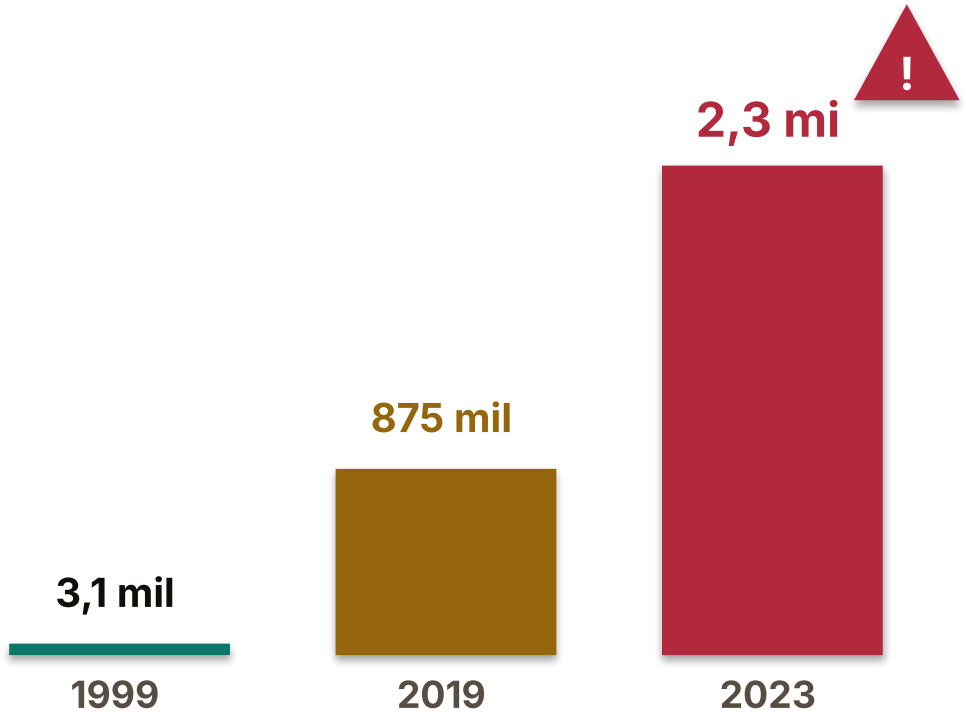
Crise de enriquecimento da NVD/NIST (2026)

A NIST só analisa os CVEs prioritários. O resto fica sem classificação, e sobra para o CSIRT local.

(NIST, 2026)

A escala do problema: incidentes

As notificações de incidentes saltaram de milhares para milhões em duas décadas.



Notificações de incidentes ao CERT.br, por ano (CERT.br/NIC.br)

A capacidade não acompanha

Déficit global de ~4,8 milhões de profissionais de segurança. Cada registro tratado por uma CSIRT carrega dados pessoais a proteger.

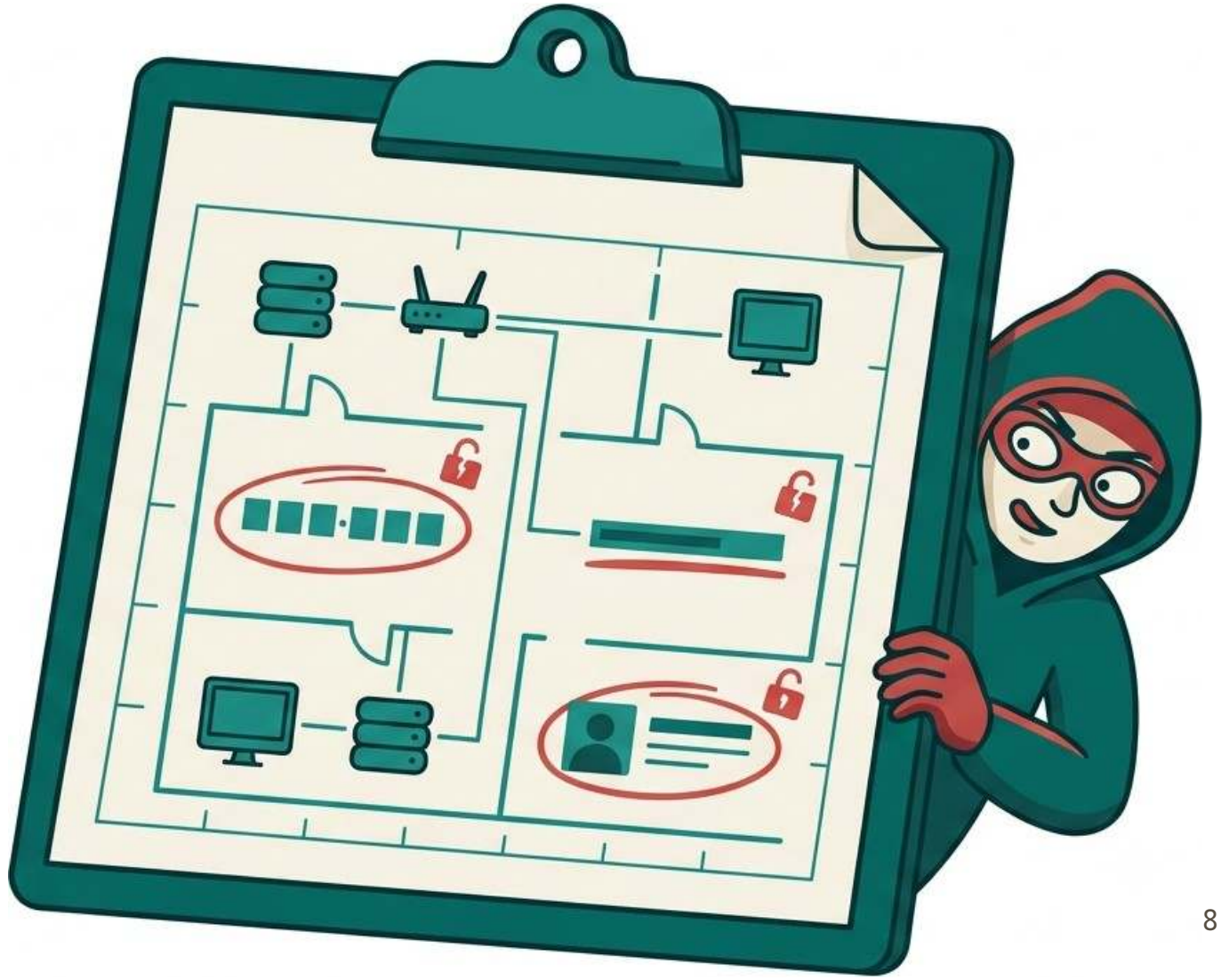
(ISC2 Workforce Study, 2024)

Automação com IA

LLMs que priorizam, classificam e sugerem respostas.

Sensibilidade dos dados

- **Identificadores de rede**
  IP, hostname, certificado
- **Dados pessoais**
  nomes e e-mails das vítimas
- **Perigoso se vazar**
  expõe a rede e fere a LGPD



Como preparar esses dados para usar a IA com segurança, em escala, sem perder a utilidade?

A resposta deste trabalho: pseudonimização.

O que é uma entidade?

- **Um tipo reconhecido**

  um dado pessoal ou um artefato técnico

- **Exemplos de entidade**

  IP, hostname, nome, hash, certificado

```
host: "100.111.20.23"   →   "[IP_ADDRESS_a4f2b1c3]"
```

Exemplo: antes e depois (Tenable)

Original

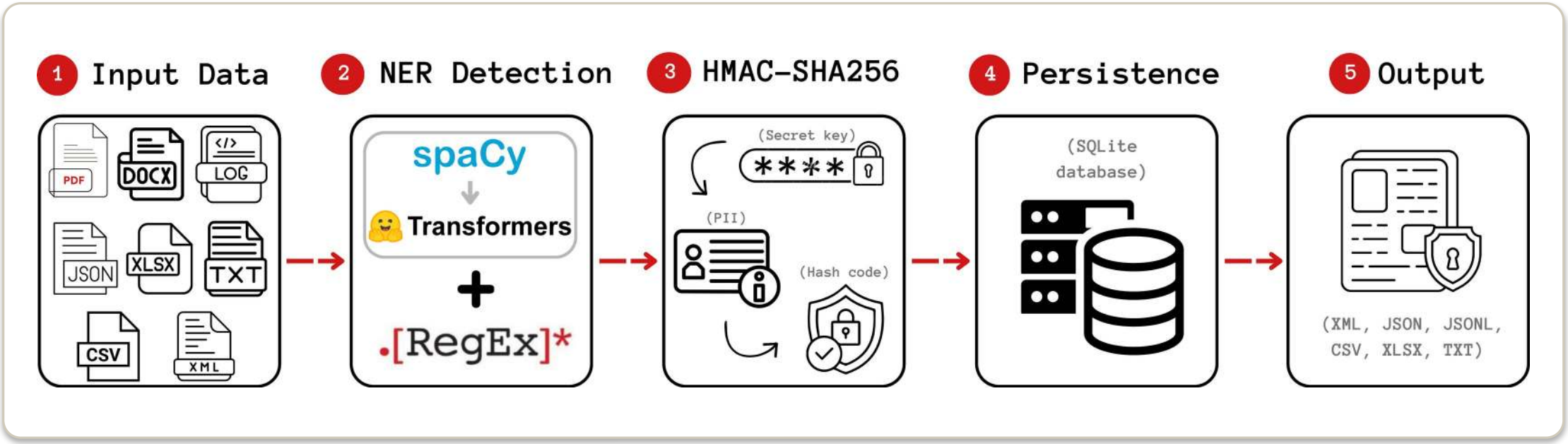
```
{
        "host-ip": "100.111.20.23",
       "host-fqdn": "dns01.corp.br",
     "mac-address": "00:1A:2B:3C:4D",
         "severity": "High",
    "cvss3_base_score": 7.4,
             "port": 443
}
```

Anonimizado

```
{
    "host-ip": "[IP_ADDRESS_48624b5c]",
     "host-fqdn": "[HOSTNAME_7e9d2a]",
   "mac-address": "[MAC_ADDRESS_c8b1]",
         "severity": "High",
    "cvss3_base_score": 7.4,
             "port": 443
}
```

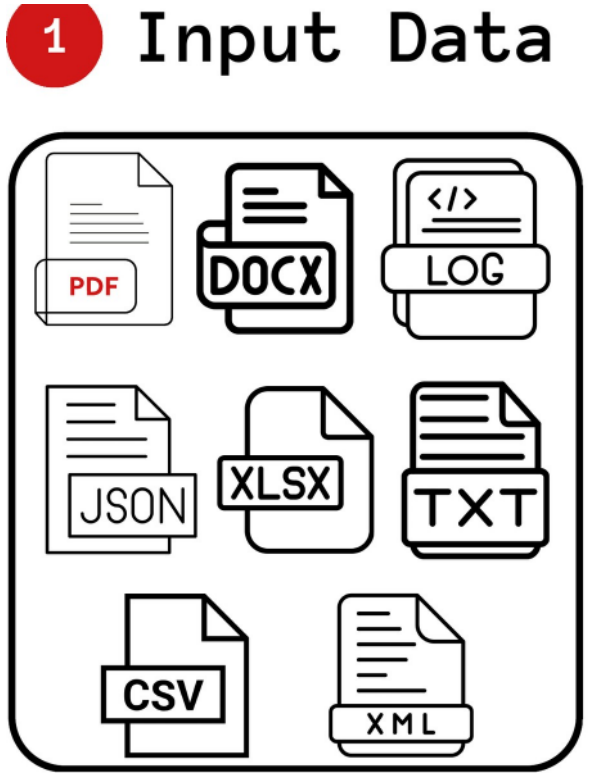
Vermelho: sensível · Teal: pseudonimizado · Preto: preservado · campos do Tenable (exemplos)

O pipeline de pseudonimização

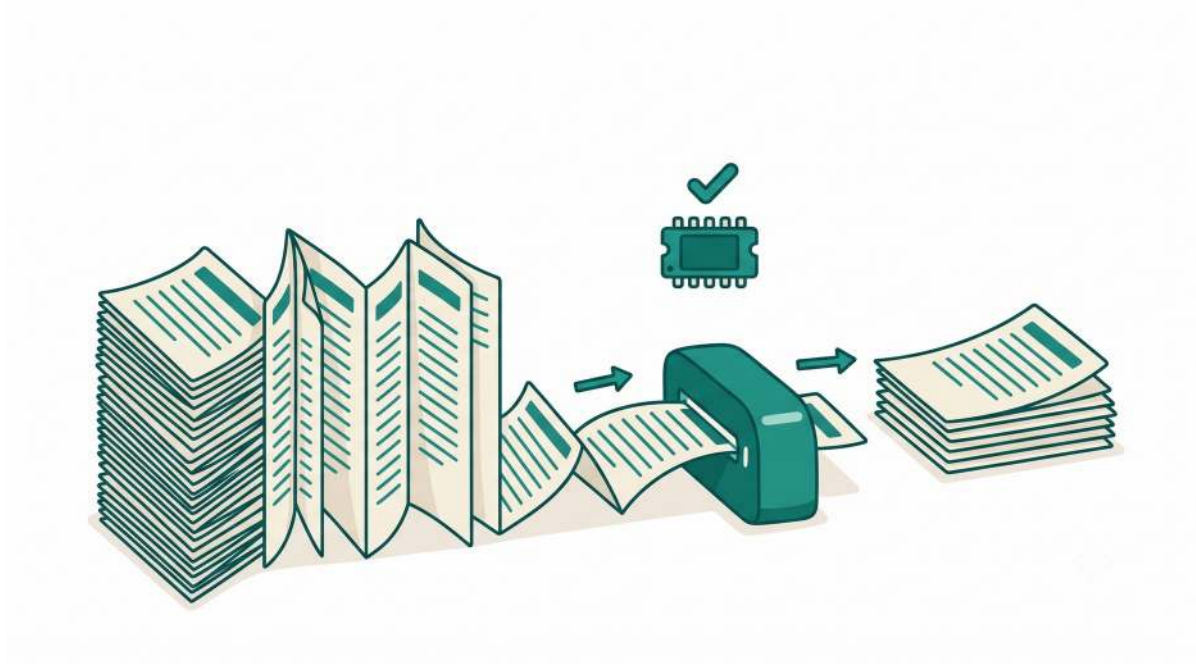


Estágio 1: Entrada

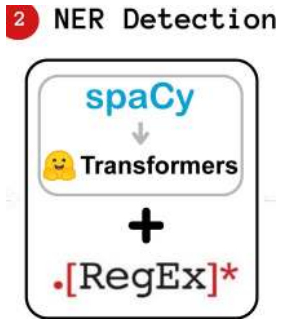
O sistema lê vários formatos. Arquivos grandes são lidos em fluxo.



Leitura em fluxo (streaming)



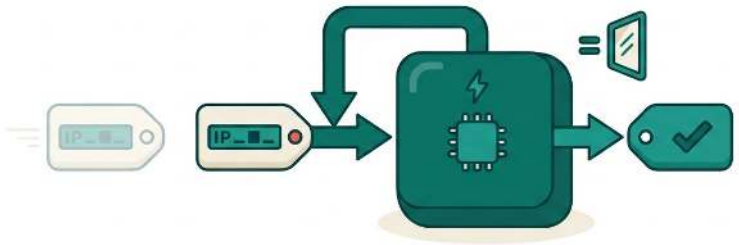
Estágio 2: Detecção (NER)



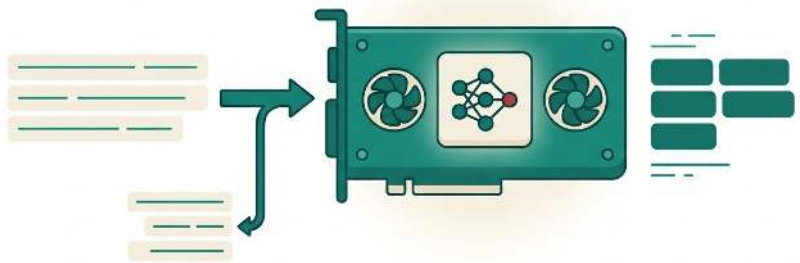

```
Failed login from 100.111.20.23  user=cristhian.kapelinski
```

o NER e o RegEx reconhecem o IP e o nome como entidades

Cache (vem antes)



GPU no modelo NER

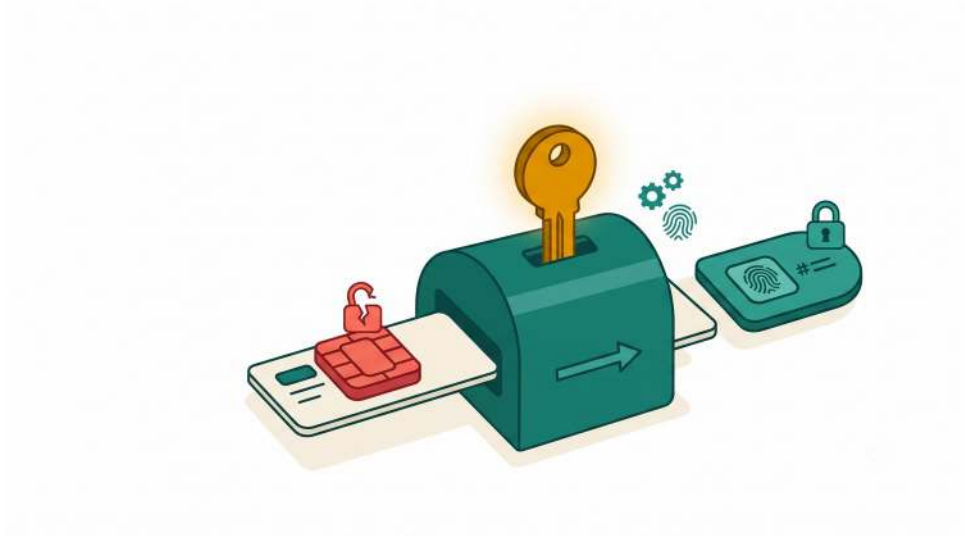


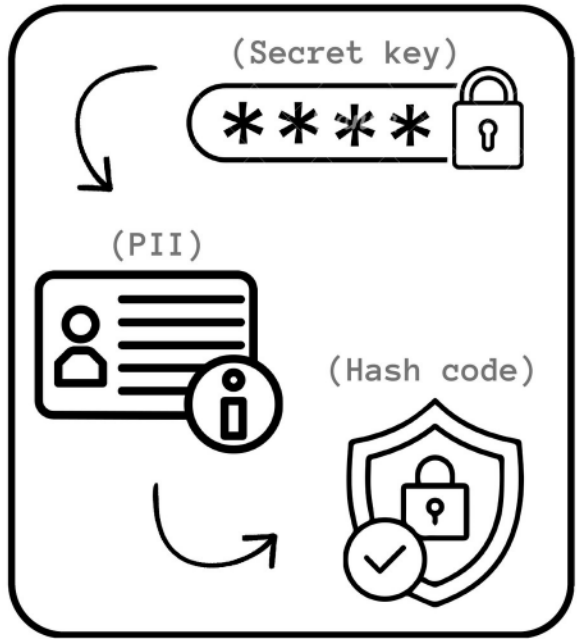
🔑 Estágio 3: HMAC-SHA256

Cada entidade vira um pseudônimo gerado com uma chave secreta.

`100.111.20.23` ➡ `[IP_ADDRESS_48624b5c]`

o prefixo guarda o **tipo da entidade**; o resto é o **hash gerado com a chave secreta**.





Estágios 4 e 5: Persistência e Saída

Persistência

banco SQLite (tabela `entities`): guarda o hash completo

Tipo	Original	Pseudônimo	Hash completo (64c)
IP_ADDRESS	100.111.20.23	[IP_ADDRESS_48624b5c]	48624b5c9a3f2e1d…
HOSTNAME	srv-web-01	[HOSTNAME_e5c53b15]	e5c53b15a1d2f3e4…

Saída (ex.: XML)

estrutura mantida; sensíveis (teal) viram pseudônimo, o resto (preto) é preservado

```
<result>
  <host>[IP_ADDRESS_5b02f379]</host>
  <hostname>[HOSTNAME_e5c53b15]</hostname>
  <port>[PORT_2edd4ed6]</port>  <cvss>10.0</cvss>  <severity>Critical</severity>
</result>
```

A linha de pesquisa AnonLFI

AnonLFI 1.0

primeiro pipeline de outra equipe (Bandel et al.)

→

AnonLFI 2.0

arquitetura extensível (WRSeg)

→

AnonShield

escala (SBRC, Salão Fer.)

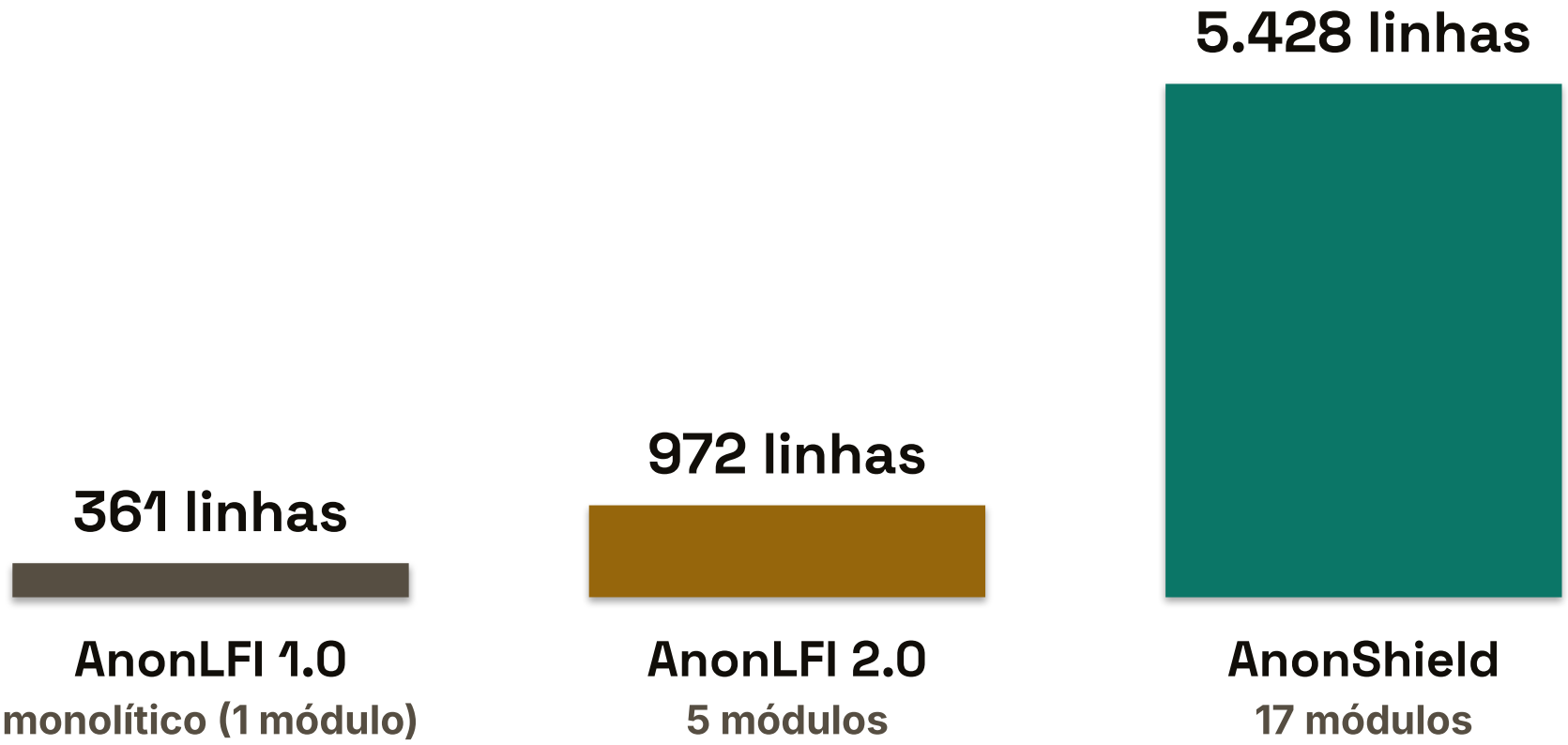
SBSeg 2025

WRSeg 2025

SBRC 2026

Evolução da linha AnonLFI

Linhas de código e número de módulos por versão.



Motivação: a escala operacional

- **70.951 vulnerabilidades**
  do CAIS/RNP
- **Antes: +92 horas**
  inviável em escala
- **Daí o AnonShield**
  feito para escalar

Conjuntos de dados

Base	Origem	Formatos	Arquivos	Tamanho	Finalidade
D1	OpenVAS (laboratório VulnLab)	CSV, TXT, PDF, XML	130	9 a 34 MB	Acurácia e desempenho
D1C	Conversão do D1 (outros formatos)	XLSX, DOCX, PDF-img, JSON	130	até 1,5 GB	Acurácia e desempenho

Laboratório de serviços vulneráveis

VulnLab: 130 containers Docker vulneráveis, em 11 categorias de serviços.

Aplicações web vulneráveis · 21

OWASP · DVWA

Linguagens e runtimes · 12

Python · Node.js · PHP

Monitoramento e logging · 6

Grafana · Kibana · Prometheus

Rede e infraestrutura · 19

OpenSSH · OpenLDAP · Memcached

CMS e plataformas web · 11

WordPress · Drupal · GitLab

Mensageria e streaming · 5

RabbitMQ · Kafka · ActiveMQ

Bancos de dados · 17

MySQL · PostgreSQL · MongoDB

DevOps e CI/CD · 9

Jenkins · SonarQube · Nexus

CVEs específicos · 5

Log4Shell

Servidores web e de aplicação · 17

Apache · Nginx · Tomcat

Sistemas operacionais · 8

Ubuntu · Debian · CentOS

Varredura com OpenVAS

Cada container é varrido com OpenVAS/GVM (Greenbone); os relatórios formam o D1.

- **OpenVAS/GVM (Greenbone)**
  motor de varredura de vulnerabilidades
- **Cobertura de portas**
  todas as portas IANA atribuídas (TCP e UDP)
- **Perfil Full and fast**
  ~30 a 60 min por host
- **130 alvos**
  cada container do VulnLab é varrido

~14 h de varredura total **520** relatórios (130 alvos × 4 formatos: PDF, XML, CSV, TXT)

Metodologia

- **Amostra: 67 de 6.472 (D1)**

  90% confiança, 3 especialistas

- **13 tipos de entidades**

  avaliados manualmente

- **Métricas**

  Precisão, Recall (crítica), F1

Acurácia

Versão	Precisão	Recall	F1
AnonLFI 1.0	32,4%	18,8%	23,8%
AnonLFI 2.0	80,9%	40,7%	54,2%
AnonShield (standalone)	**91,7%**	**96,1%**	**93,8%**

Comparação na estratégia standalone (detecção própria do AnonShield: NER + RegEx). Modelo NER: SecureModernBERT.

Conjuntos de dados

Base	Origem	Formatos	Arquivos	Tamanho	Finalidade
D2	Tenable (CAIS/RNP, privado)	CSV, JSON	1	550 MB	Desempenho
D3	Sintético (gêmeo público do D2)	CSV, JSON	1	445 MB	Desempenho

Desempenho em escala operacional

738×

speedup sobre o AnonLFI 2.0

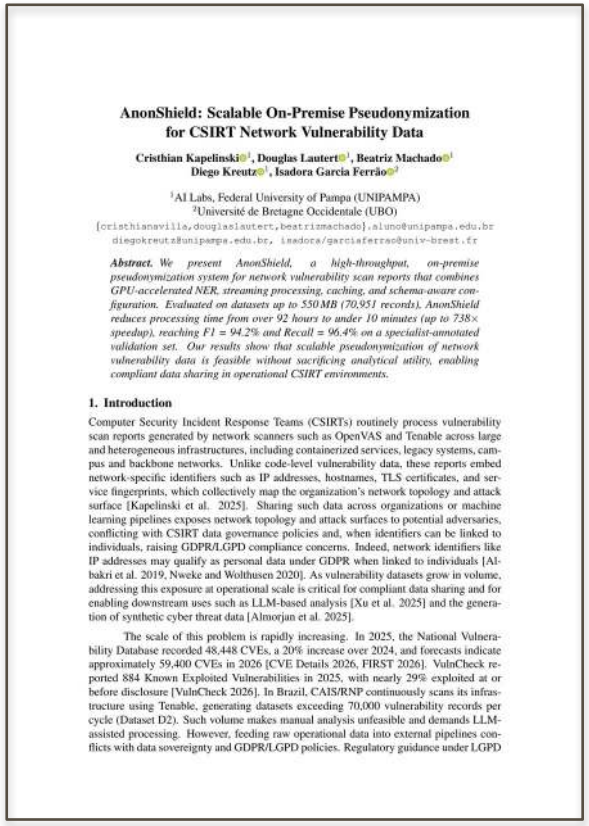
ANTES (AnonLFI 2.0, estimado)

~92 horas

DEPOIS (AnonShield)

7,5 minutos

Reconhecimento



AnonShield: Scalable On-Premise Pseudonymization for CSIRT Network Vulnerability Data

Cristhian Kapelinski[1], Douglas Lautert[1], Beatriz Machado[1]
Diego Kreutz[1], Isadora Garcia Ferrão[2]

[1]AI Labs, Federal University of Pampa (UNIPAMPA)
[2]Université de Bretagne Occidentale (UBO)
{cristhianavilla,douglaslautert,beatrizmachado}.aluno@unipampa.edu.br
diegokreutz@unipampa.edu.br, isadora/garciaferrao@univ-brest.fr

Abstract. *We present AnonShield, a high-throughput, on-premise pseudonymization system for network vulnerability scan reports that combines GPU-accelerated NER, streaming processing, caching, and schema-aware configuration. Evaluated on datasets up to 550 MB (70,951 records), AnonShield reduces processing time from over 92 hours to under 10 minutes (up to 738× speedup), reaching F1 = 94.2% and Recall = 96.4% on a specialist-annotated validation set. Our results show that scalable pseudonymization of network vulnerability data is feasible without sacrificing analytical utility, enabling compliant data sharing in operational CSIRT environments.*

1. Introduction

Computer Security Incident Response Teams (CSIRTs) routinely process vulnerability scan reports generated by network scanners such as OpenVAS and Tenable across large and heterogeneous infrastructures, including containerized services, legacy systems, campus and backbone networks. Unlike code-level vulnerability data, these reports embed network-specific identifiers such as IP addresses, hostnames, TLS certificates, and service fingerprints, which collectively map the organization's network topology and attack surface [Kapelinski et al. 2025]. Sharing such data across organizations or machine learning pipelines exposes network topology and attack surfaces to potential adversaries, conflicting with CSIRT data governance policies and, when identifiers can be linked to individuals, raising GDPR/LGPD compliance concerns. Indeed, network identifiers like IP addresses may qualify as personal data under GDPR when linked to individuals [Albakri et al. 2019, Nweke and Wolthusen 2020]. As vulnerability datasets grow in volume, addressing this exposure at operational scale is critical for compliant data sharing and for enabling downstream uses such as LLM-based analysis [Xu et al. 2025] and the generation of synthetic cyber threat data [Almorjan et al. 2025].

The scale of this problem is rapidly increasing. In 2025, the National Vulnerability Database recorded 48,448 CVEs, a 20% increase over 2024, and forecasts indicate approximately 59,400 CVEs in 2026 [CVE Details 2026, FIRST 2026]. VulnCheck reported 884 Known Exploited Vulnerabilities in 2025, with nearly 29% exploited at or before disclosure [VulnCheck 2026]. In Brazil, CAIS/RNP continuously scans its infrastructure using Tenable, generating datasets exceeding 70,000 vulnerability records per cycle (Dataset D2). Such volume makes manual analysis unfeasible and demands LLM-assisted processing. However, feeding raw operational data into external pipelines conflicts with data sovereignty and GDPR/LGPD policies. Regulatory guidance under LGPD

Resultados da avaliação de artefatos (CTA)

SBRC2026
PRAIA DO FORTE - BAHIA

Melhor Artefato do SBRC 2026

Selecionado entre os 65 artefatos avaliados no evento.

Selos de artefato concedidos

Utilidade analítica dos dados

Os dados pseudonimizados já sustentam pesquisa publicada:

Severo et al. (2025)

Classificação automatizada de incidentes em SOCs com LLMs e engenharia de prompt.

Melhor resultado: **92,27% de acurácia** (Gemini 2 + Progressive-Hint Prompting)

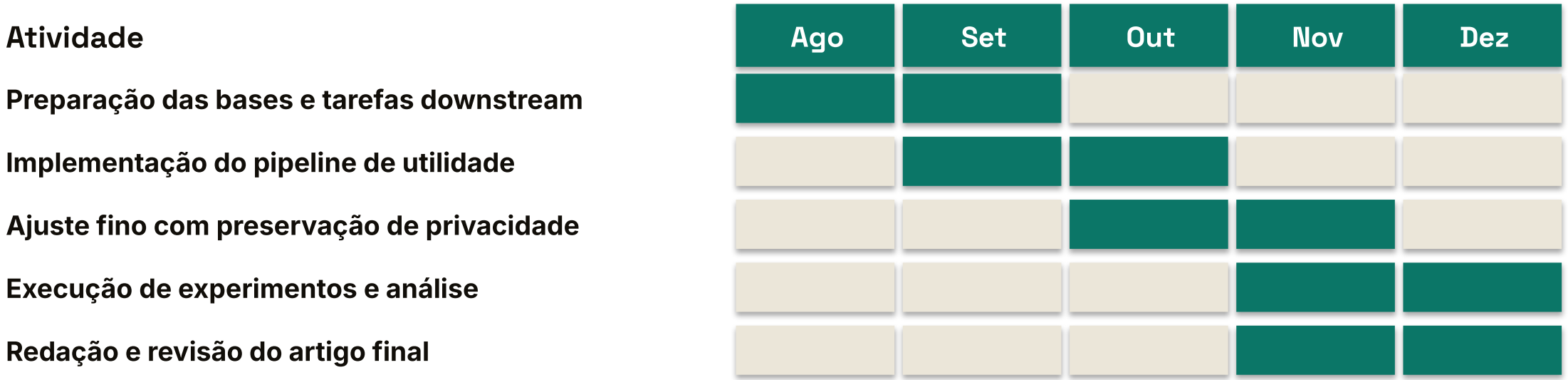
Almeida et al. (2025)

Modelos de linguagem on-premise vs. comerciais para classificar incidentes em CSIRTs.

On-premise **~60%** vs. **>90%** dos comerciais

Trabalhos futuros (TCC II)

Avaliação comparativa entre dados pseudonimizados e originais, medindo o equilíbrio entre privacidade e utilidade na classificação de vulnerabilidades e incidentes com modelos de linguagem.



Artigos e materiais

SBSeg 2025

WRSeg 2025

SBRC 2026

anonshield.org

código e artefatos

Obrigado.

Apoio: RNP CNPq

Cristhian Kapelinski · Orientador: Diego Kreutz