Agentic Malware Analysis: From Assistance to Automated Investigation

Tim Blazytko

@mr_phrazer
synthesis.to
tim@blazytko.to

About Tim

- PhD in binary program analysis & reverse engineering
- **independent** — consulting, training & tooling
- focus: (de)obfuscation, malware analysis & automation

Setting the Scene

Foundations

Agentic reverse engineering

Agentic malware analysis

Foundations of Agentic RE

From Prompting to Agents

Prompting

- prompting: paste code, ask questions, copy results back
- no tool access, no iteration, no memory
- agents can read files, run commands, modify code

A Shift in Reverse Engineering

manual

disassembler

+ analyst

A Shift in Reverse Engineering

manual

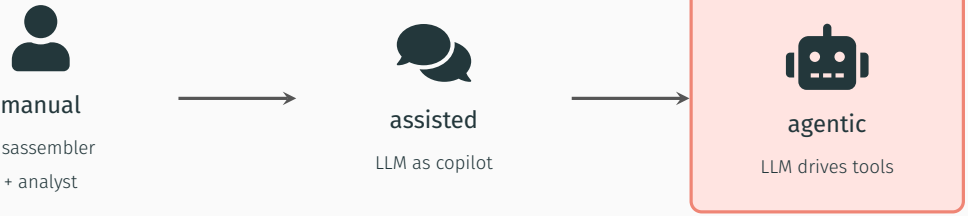
disassembler

+ analyst

assisted

LLM as copilot

A Shift in Reverse Engineering



A Shift in Reverse Engineering

manu

disassem

+ analyst

LLM as copilot

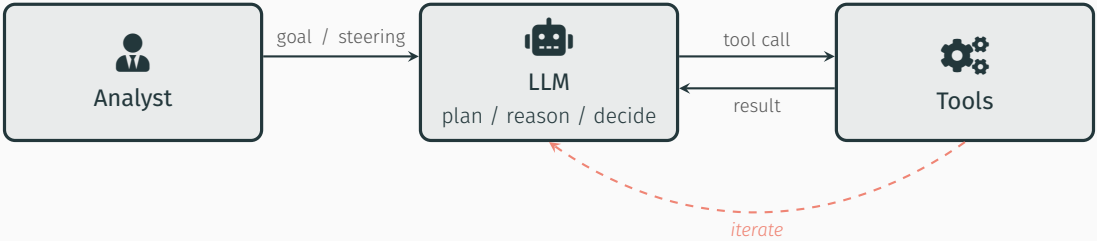
gentic

LLM drives tools

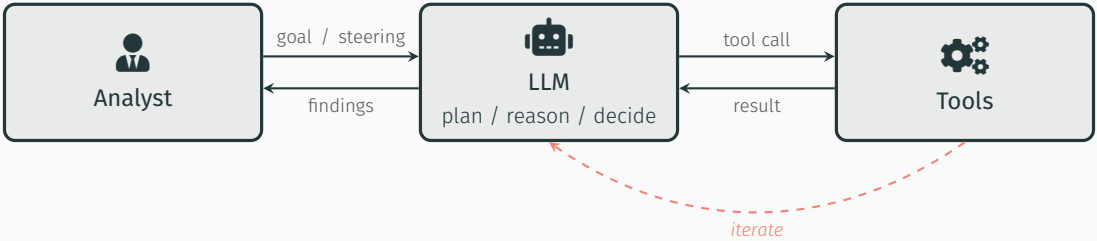
agents now drive the analysis

The Agent Loop

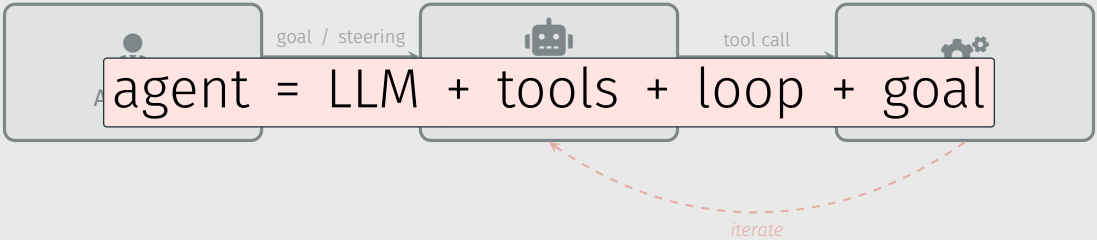
The Agent Loop



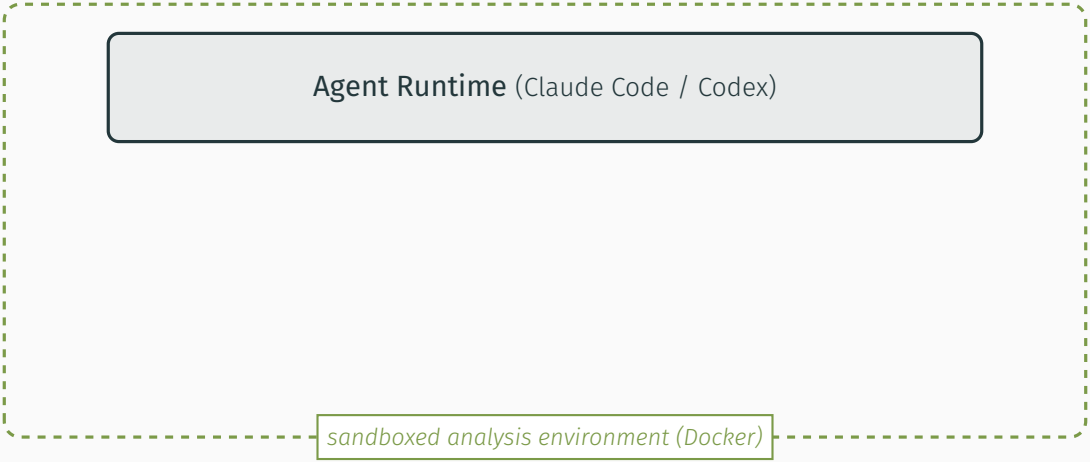
The Agent Loop



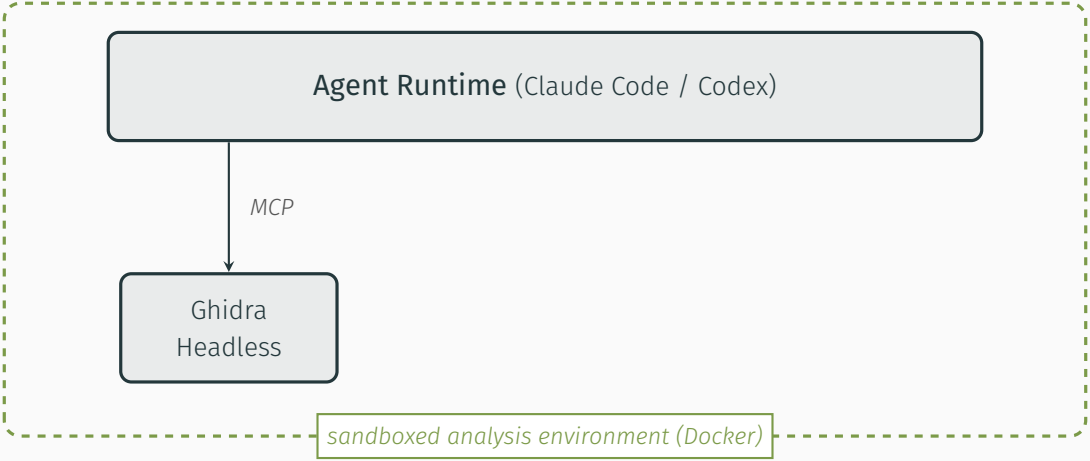
The Agent Loop



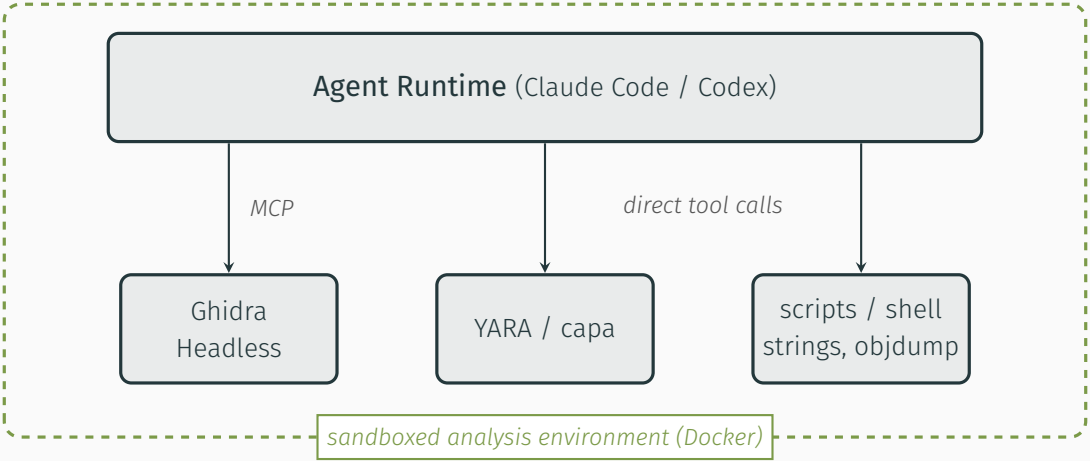
Agentic Analysis Environment



Agentic Analysis Environment



Agentic Analysis Environment



Beyond Standard Tools

Why Dedicated Tools?

- agents *can* script everything from scratch
- but scripting wastes context and hurts reliability
- dedicated tools: less context, better results

Why Dedicated Tools?

- agents *can* script everything from scratch
- but scripting wastes context and hurts reliability

give agents the right tools, not just a shell

- dedicated tools: less context, better results

MCP

Model Context Protocol

standardized protocol
for tool access

MCP & Skills

MCP

Model Context Protocol

standardized protocol
for tool access

Skills

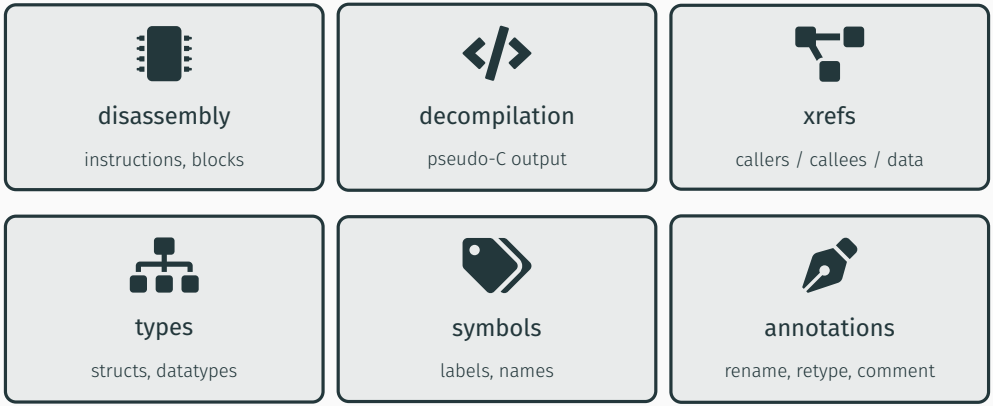
reusable agent workflows

recipes, templates,
procedural know-how

MCP Changes the Interface Layer

- standardized protocol for agent-to-tool calls
- agents discover, invoke, and process tool results
- from assistant chat to autonomous tool execution

Ghidra Headless MCP

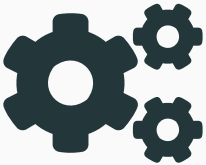


https://github.com/mrphrazer/ghidra-headless-mcp

Skills: Reusable Workflows

- task-specific guidance that tells agents **how** to approach a problem
- include workflows, scripts, templates, output expectations
- limit exploration by providing structured guidance
- composable: combine skills for complex multi-stage tasks

Agentic Malware Analysis



Task Automation

The Repetitive Side of Malware Analysis


string & config decryption


API hashing & imports


unpacking & multi-stage


IOC & artifact extraction


triage & fingerprinting


yara / capa rule support

The Repetitive Side of Malware Analysis

string & config decryption

API hashing & imports

u focused subtasks for the agent ion

triage & fingerprinting

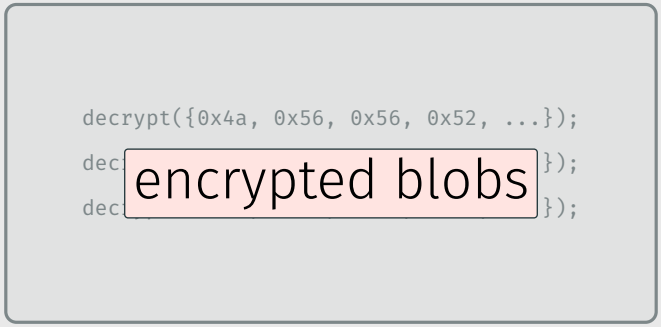
yara / capa rule support

String Decryption

Agentic String Decryption

```
decrypt({0x4a, 0x56, 0x56, 0x52, ...});
decrypt({0x61, 0x57, 0x50, 0x50, ...});
decrypt({0x52, 0x4d, 0x55, 0x47, ...});
```

Agentic String Decryption



Agentic String Decryption

```
def decrypt(buf):
  return bytes(
    b ^ 0x22
    for b in buf
  )
```

Agentic String Decryption

```
def decrypt(buf):
  return bytes(
```

agent writes a script

```
)
```

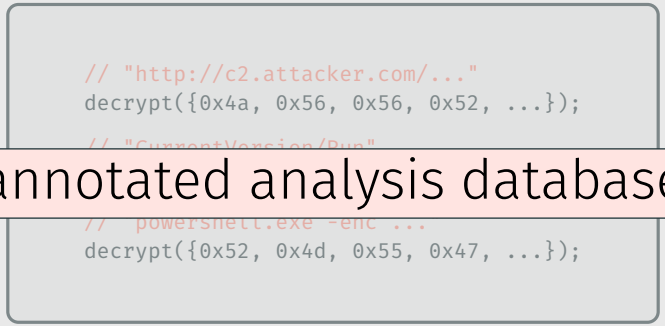
Agentic String Decryption

```
// "http://c2.attacker.com/..."
decrypt({0x4a, 0x56, 0x56, 0x52, ...});

// "CurrentVersion/Run"
decrypt({0x61, 0x57, 0x50, 0x50, ...});

// "powershell.exe -enc ..."
decrypt({0x52, 0x4d, 0x55, 0x47, ...});
```

Agentic String Decryption



API Hashing

Agentic API Hashing

```
if hash(api) == 0x6F0A52E1
if hash(api) == 0xB45C8907
if hash(api) == 0x29E3D104
```

Agentic API Hashing

```
if hash(api) == 0x6F0A52E1
if hash(api) == 0xB45C8907
if hash(api) == 0x29E3D104
```

hash checks in malware

Agentic API Hashing

```
if hash(api) == 0x6F0A52E1
if hash(api) == 0xB45C8907
if hash(api) == 0x29E3D104
```

```
NtCreateFile
NtAllocateVirtualMemory
NtProtectVirtualMemory
NtClose, ...
```

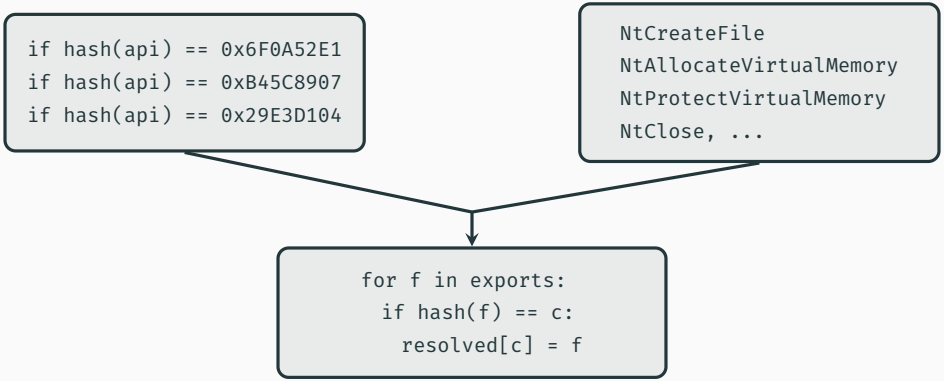
Agentic API Hashing

```
if hash(api) == 0x6F0A52E1
if hash(api) == 0xB45C8907
if hash(api) == 0x29E3D104
```

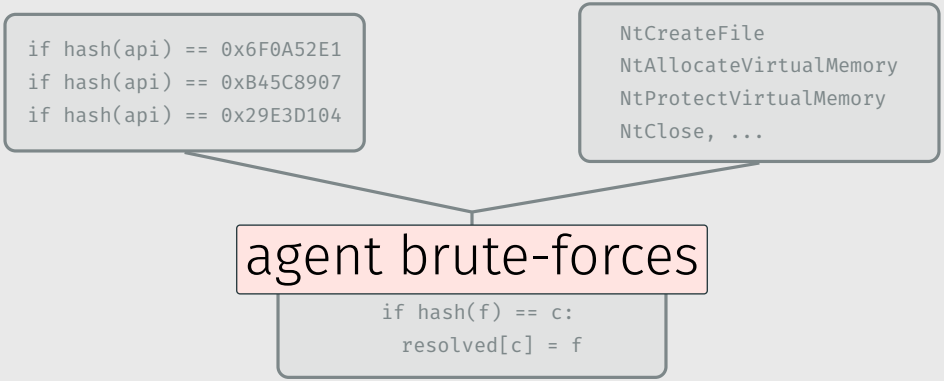
```
NtCreateFile
NtAllocateVirtualMemory
NtProtectVirtualMemory
NtClose, ...
```

list of ntdll exports

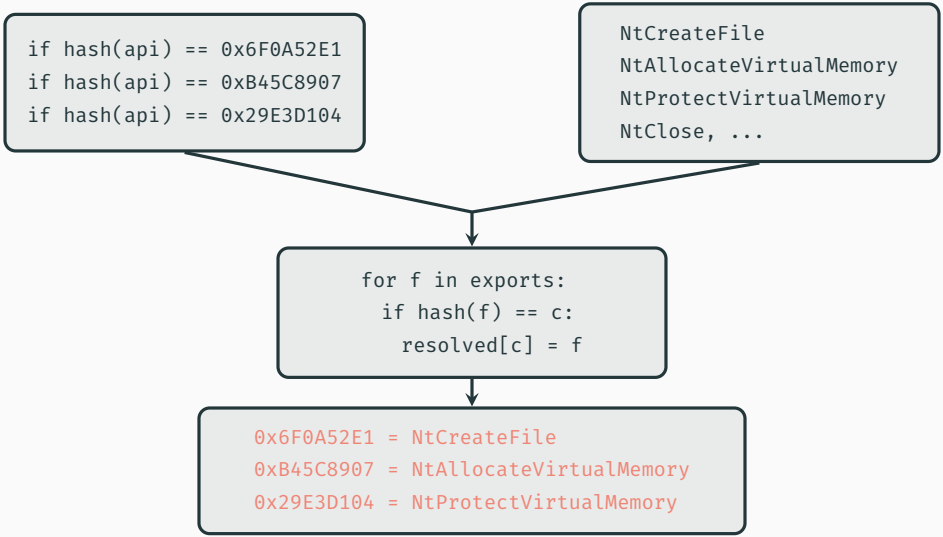
Agentic API Hashing



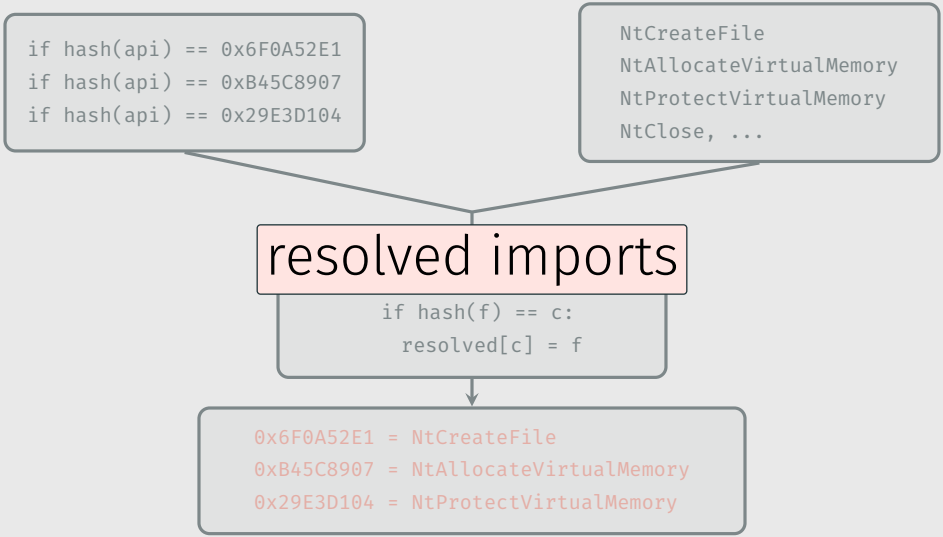
Agentic API Hashing



Agentic API Hashing

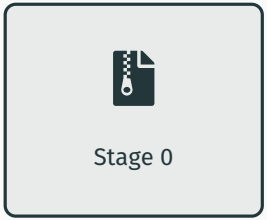


Agentic API Hashing



Multi-Stage Unpacking

Agentic Multi-Stage Unpacking



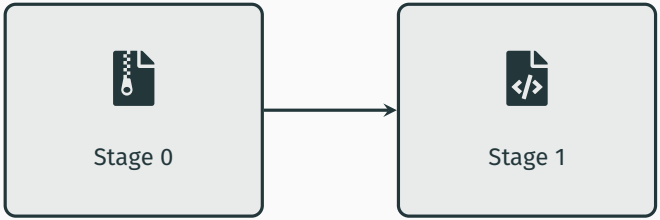
https://www.amnesty.org/en/latest/research/2020/09/german-made-finspy-spyware-found-in-egypt-and-mac-and-linux-versions-revealed/

Agentic Multi-Stage Unpacking

encrypted loader / dropper

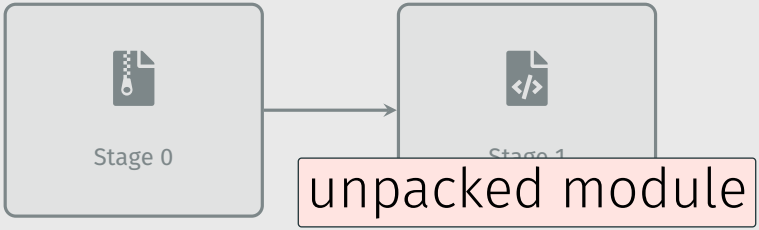
https://www.amnesty.org/en/latest/research/2020/09/german-made-finspy-spyware-found-in-egypt-and-mac-and-linux-versions-revealed/

Agentic Multi-Stage Unpacking



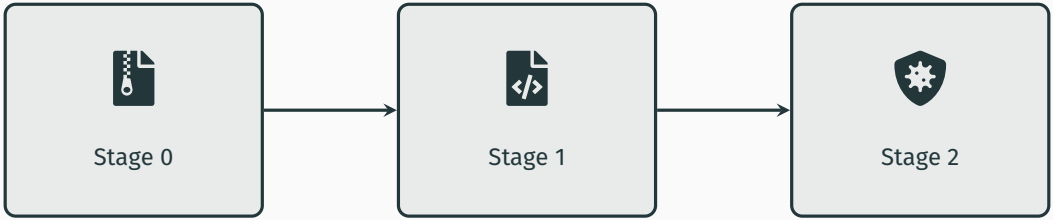
https://www.amnesty.org/en/latest/research/2020/09/german-made-finspy-spyware-found-in-egypt-and-mac-and-linux-versions-revealed/

Agentic Multi-Stage Unpacking



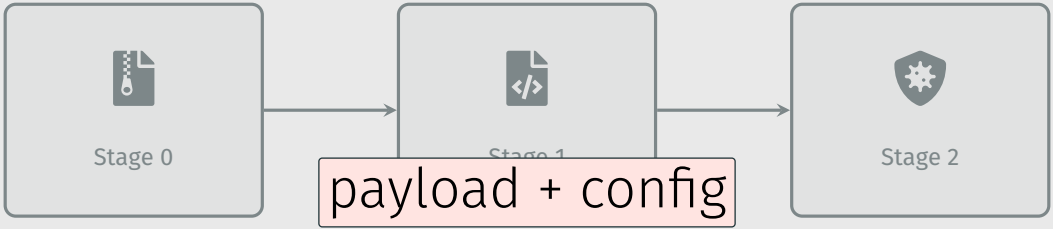
https://www.amnesty.org/en/latest/research/2020/09/german-made-finspy-spyware-found-in-egypt-and-mac-and-linux-versions-revealed/

Agentic Multi-Stage Unpacking



https://www.amnesty.org/en/latest/research/2020/09/german-made-finspy-spyware-found-in-egypt-and-mac-and-linux-versions-revealed/

Agentic Multi-Stage Unpacking



https://www.amnesty.org/en/latest/research/2020/09/german-made-finspy-spyware-found-in-egypt-and-mac-and-linux-versions-revealed/

Recap: The Repetitive Side of Malware Analysis


string & config decryption


API hashing & imports


unpacking & multi-stage


IOC & artifact extraction


triage & fingerprinting


yara / capa rule support

Recap: The Repetitive Side of Malware Analysis

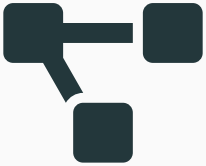

string & config decryption


API hashing & imports


many tasks can be given to agents


triage & fingerprinting

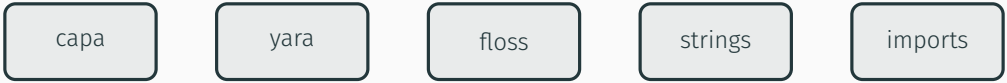

yara / capa rule support



Agentic Pipelines

Automated Initial Triage

Triage Pipeline



Triage Pipeline

capa

yara

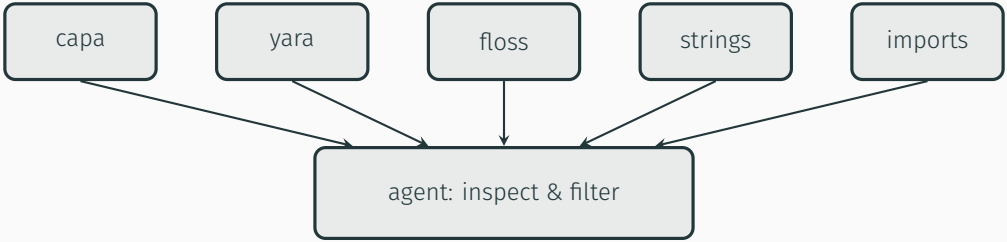
floss

strings

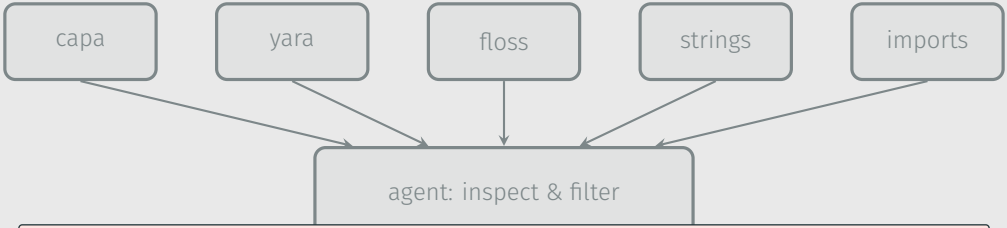
imports

standard tools surface points of interest

Triage Pipeline

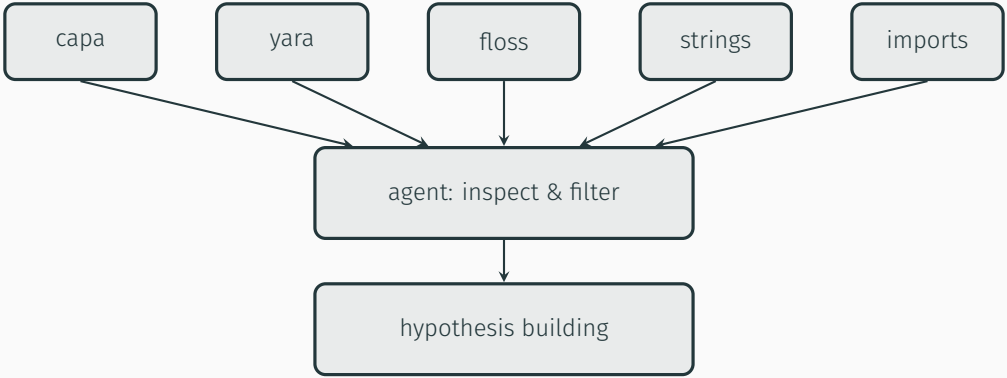


Triage Pipeline

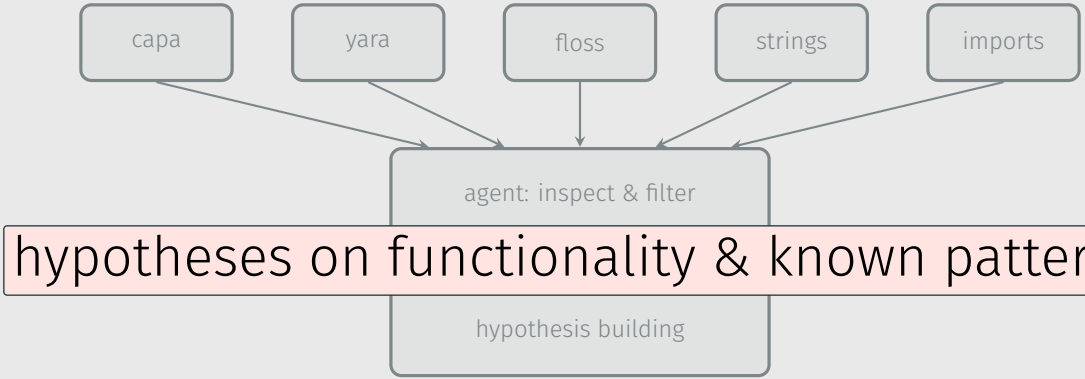


revealing strings, suspicious imports, ...

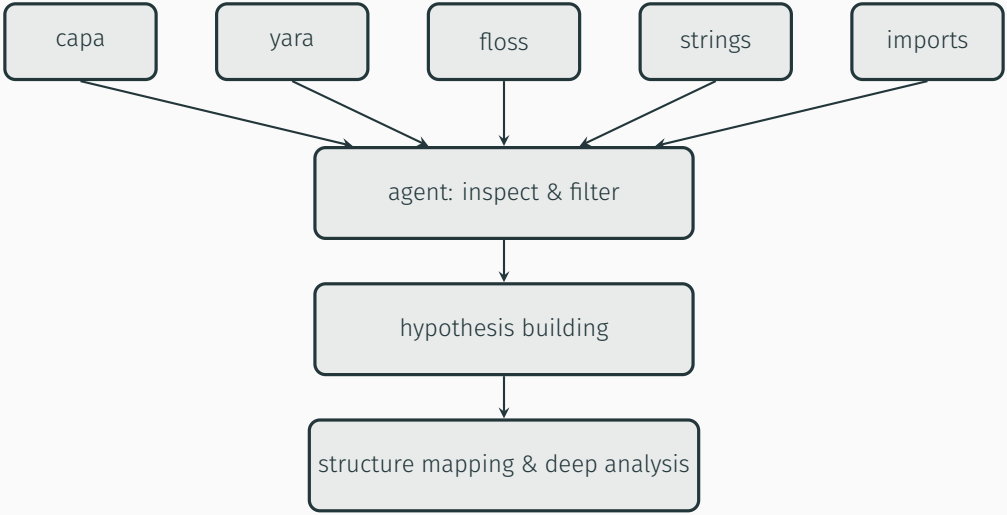
Triage Pipeline



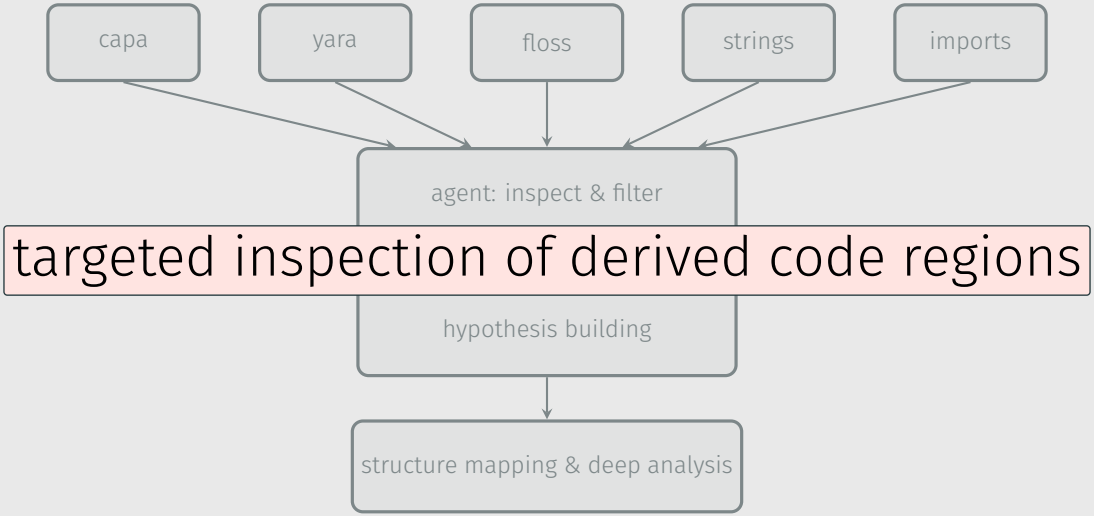
Triage Pipeline



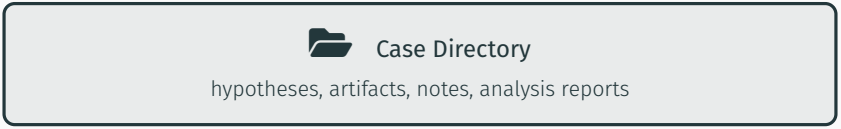
Triage Pipeline

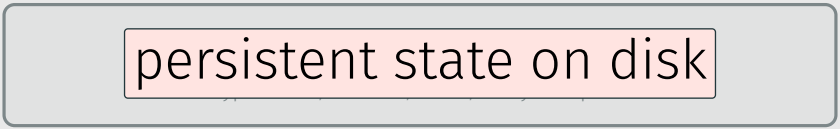


Triage Pipeline

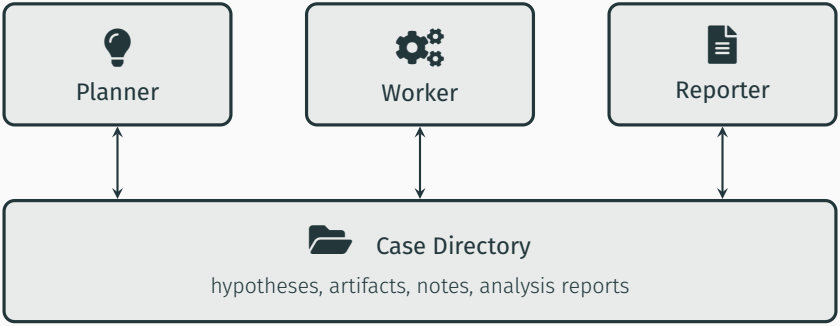


Pipeline Architecture

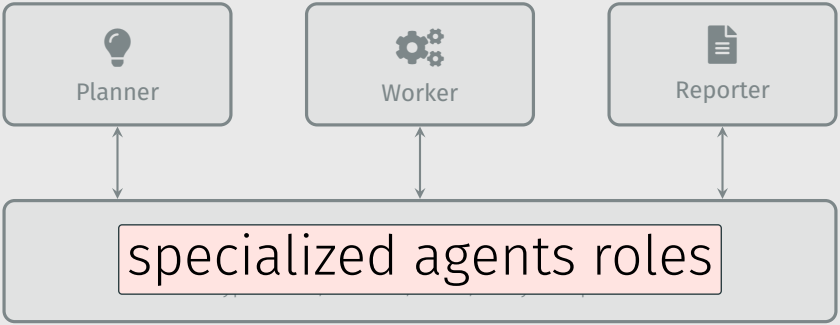



persistent state on disk

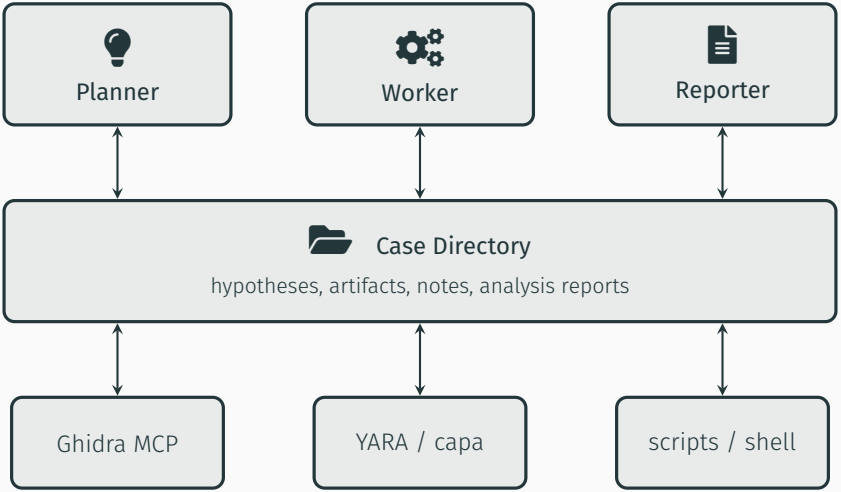
Pipeline Architecture



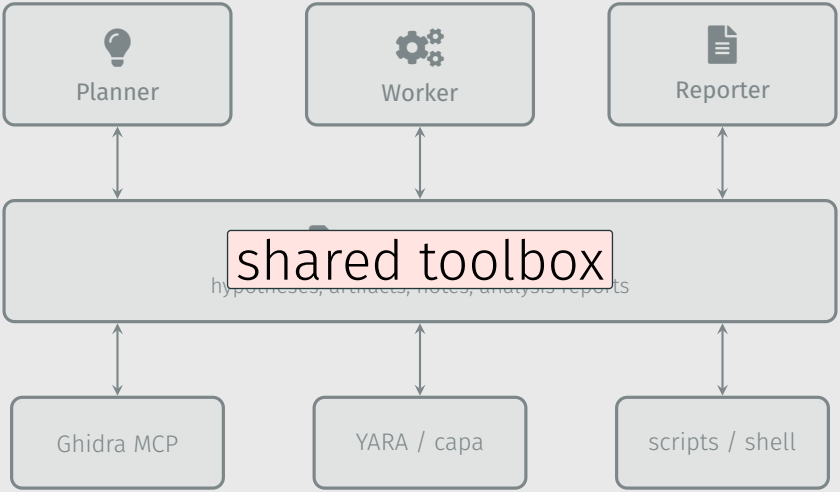
Pipeline Architecture



Pipeline Architecture



Pipeline Architecture



Handover to the Analyst

Case Directory

ranked hypotheses with confidence

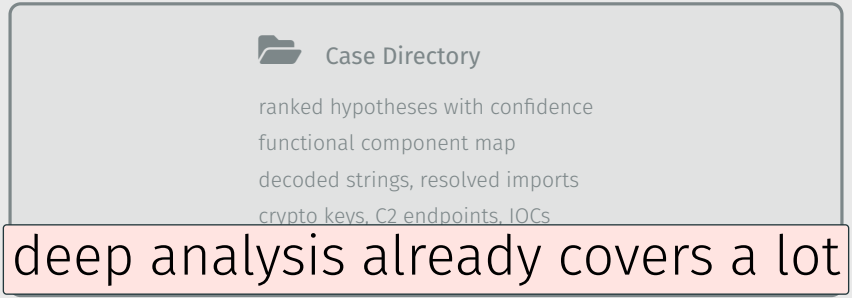
functional component map

decoded strings, resolved imports

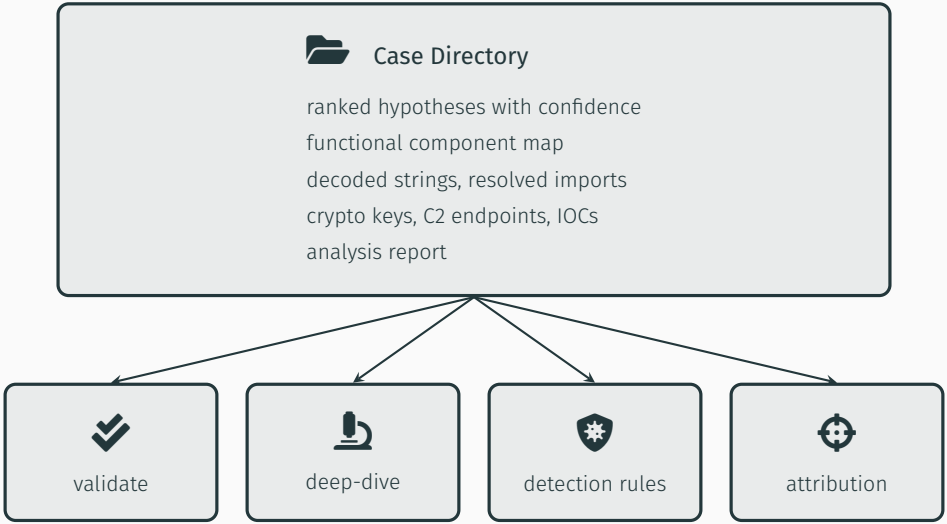
crypto keys, C2 endpoints, IOCs

analysis report

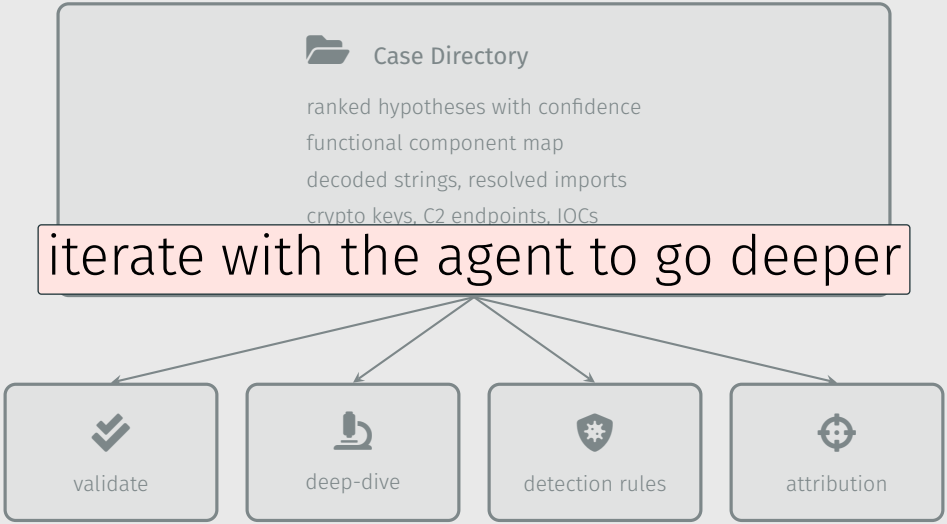
Handover to the Analyst



Handover to the Analyst



Handover to the Analyst



Example: Bundestrojaner (mfc42ul.dll)

```
H1: Full-Featured Surveillance RAT —VALIDATED
- Discovered complete command dispatch vtable at 0x1003d7d0 with 10 remotely-callable commands
- Mapped orchestrator vtable at 0x1003d7f8 with C2 data loop, fingerprinting

H6: C2 with AES-128 Encryption —VALIDATED
- Protocol vtable at 0x1003d330: encrypt+send, recv+decrypt, raw I/O via WinSock send()/recv()
- Encrypt: sub_10012130, Decrypt: sub_10013260, Key expansion: sub_10014410
- Wire format: C3PO-r2d2-POE →4-byte length →AES-128-ECB encrypted payload
```

https://synthesis.to/2026/03/18/agentic_malware_analysis.html

Local LLMs

When Cloud Isn't an Option

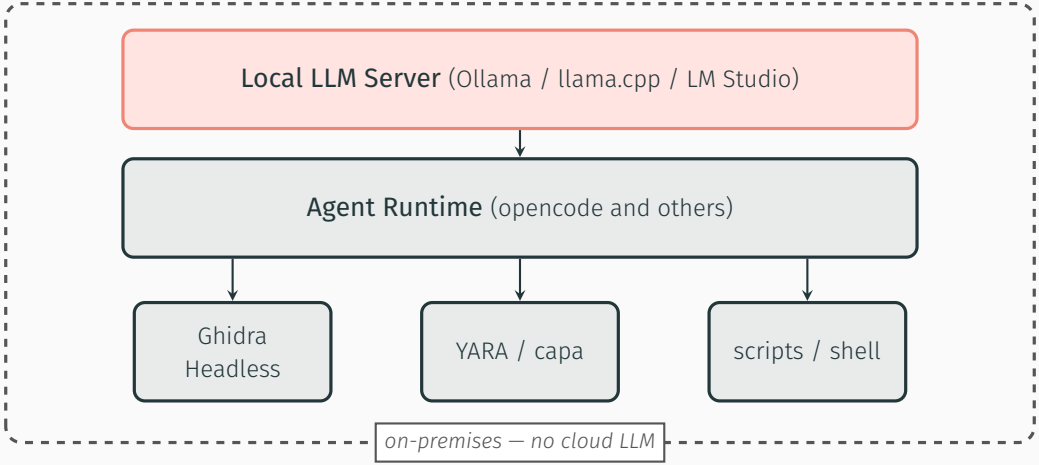
privacy & data protection

digital sovereignty

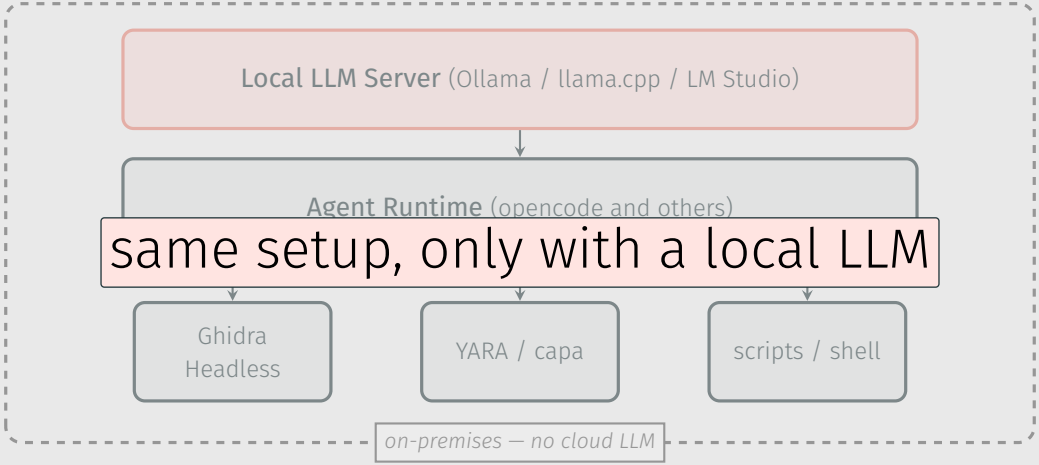
air-gapped networks

chain of custody

Agentic Analysis Environment with Local LLMs



Agentic Analysis Environment with Local LLMs



Local LLMs — Pros & Cons

Pros

- runs offline
- privacy-sensitive
- fully self-hosted

Cons

- slower per query
- below frontier capability
- needs GPU / workstation

Open-Weight Models

Qwen3.6	GLM-5.2	Kimi K3	DeepSeek-V4
27B / 35B-A3B	753B	2.8T	Flash / Pro

workstation-friendly: Qwen3.6 with quantization

on-prem cluster: GLM-5.2, Kimi K3, DeepSeek-V4

Qwen3.6	GLM-5.2	Kimi K3	DeepSeek-V4
27B / 35B-A3B	753B	2.8T	Flash / Pro

open weights ≠ workstation-sized

workstation-friendly: Qwen3.6 with quantization

on-prem cluster: GLM-5.2, Kimi K3, DeepSeek-V4

Wrap-Up

Current Limitations

- non-deterministic reruns
- context-window limits
- analyst still in the loop
- output needs validation

Takeaways

1. agents = LLM + tools + loop + goal, driven via MCP and skills
2. they reliably handle focused RE and malware tasks
3. pipelines turn task automation into deeper, resumable investigation
4. analyst expertise shifts towards steering and validation

agents change the economics of reverse engineering

Trainings & Contact

Training topics

- Reverse Engineering & Binary Program Analysis
- Software Deobfuscation Techniques
- Android & Firmware Reverse Engineering
- Agentic Reverse Engineering & Malware Analysis

on-site & remote — tailored to your team

Tim Blazytko | `https://synthesis.to` | tim@blazytko.to

`synthesis.to`

References & Resources

- **Binary Cartography** — webinar series on RE, malware analysis & protection
  `https://github.com/mrphrazer/binary-cartography`

- **Building a Pipeline for Agentic Malware Analysis** — blog post
  `https://synthesis.to/2026/03/18/agentic_malware_analysis.html`

- **Companion repository** — pipeline, skills, Docker setup
  `https://github.com/mrphrazer/agentic-malware-analysis`

- **Ghidra Headless MCP** | **Binary Ninja Headless MCP**
  `https://github.com/mrphrazer/ghidra-headless-mcp`
  `https://github.com/mrphrazer/binary-ninja-headless-mcp`