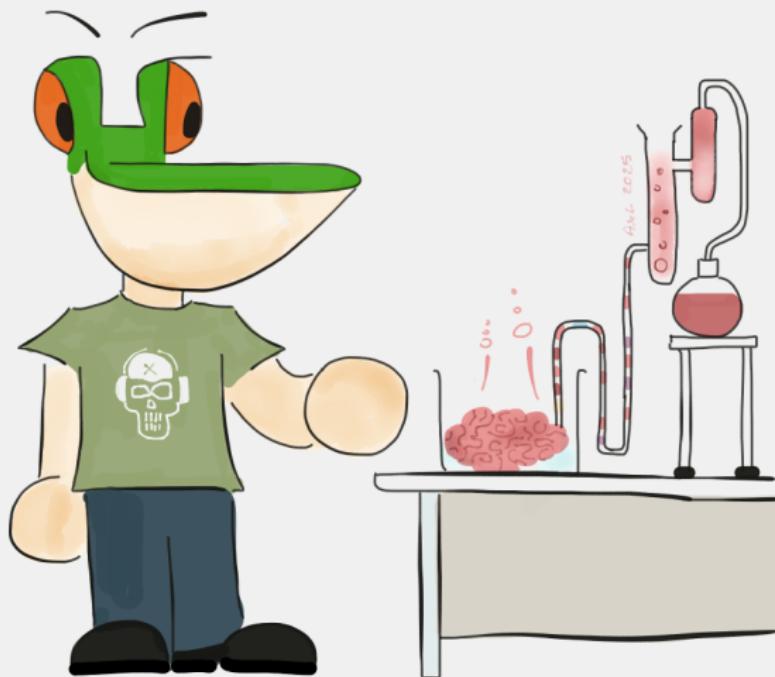


# Malware Analysis with R2AI

Axelle Apvrille  
Insomni'hack, March 2025





0.5 - 10 days



The screenshot shows a blog article header with the title 'Analyzing ELF/Sshdinjector.A!tr with a Human and Artificial Analyst'. Below the title is a sub-section 'PORTUGAL SECURITY PORTFOLIO' and a button labeled 'EXPERIENCE AN INCIDENT?'. The main content area includes sections like 'ARTICLE CONTENTS', 'Recovering Exploited ELF/Sshdinjector.A!tr', 'AI-Assisted Malware Analysis', and 'Conclusion'. At the bottom, there is a note about 'ELF/Sshdinjector.A!tr' being a collection of malware that can be injected into the SSH daemon.

Malware Analyst    Malicious binary

Virus description, blog post, article...

## Can Artificial Intelligence speed up the analysis? Can we improve analysis quality?

Malware *Detection* takes at most 2 hours.

Full details of how malware works is far more extensive work.



# Testing R2AI on Linux and IoT malware

Name	Sample Date	Description
Devura aka Sedexp	August 2024	Linux malware. Persistance through udev rules, reverse shell
Goldoon	April 2024	IoT botnet. Targets D-Link routers. Commands: TCP/UDP/DNS Flooding
RudeDevil	October 2024	Linux crypto miner and botnet. Commands: download file, stop XMR miner...

And more: [Ladvix](#) (Jan 2025), [Sshdinjector](#) (Nov 2024), [Ngioweb](#) (July 2024), [Prometei](#) (Feb 2025)...



# References

## Malware analysis

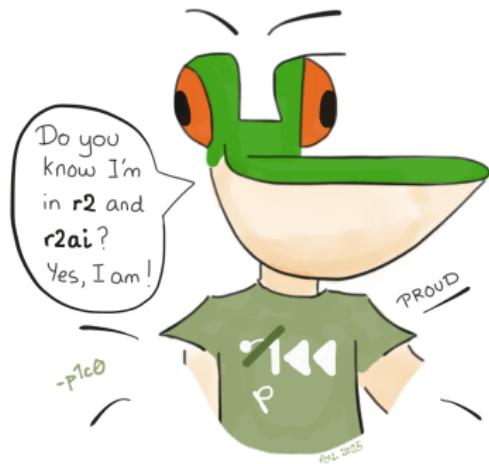
- S. Friedberg, *Unveiling “sedexp”: A Stealthy Linux Malware Exploiting udev rules*, August 2024  
<https://www.aon.com/en/insights/cyber-labs/unveiling-sedexp>
- C. Lin and V. Li, *New “Goldoon” Botnet Targeting D-Link Devices*, May 2024 <https://www.fortinet.com/blog/threat-research/new-goldoon-botnet-targeting-d-link-devices>
- R. Groenewoud, *Betting on Bots: Investigating Linux malware, crypto mining, and gambling API abuse*, September 2024  
<https://www.elastic.co/security-labs/betting-on-bots>

## Samples SHA256

- Devura: 43f72f4cdab8ed40b2f913be4a55b17e7fd8a7946a636adb4452f685c1ffea02
- Goldoon: 712d9abe8fbdff71642a4d377ef920d66338d73388bfee542f657f2e916e219c
- RudeDevil: 89b60cedc3a4efb02ceaf629d6675ec9541addae4689489f3ab8ec7741ec8055



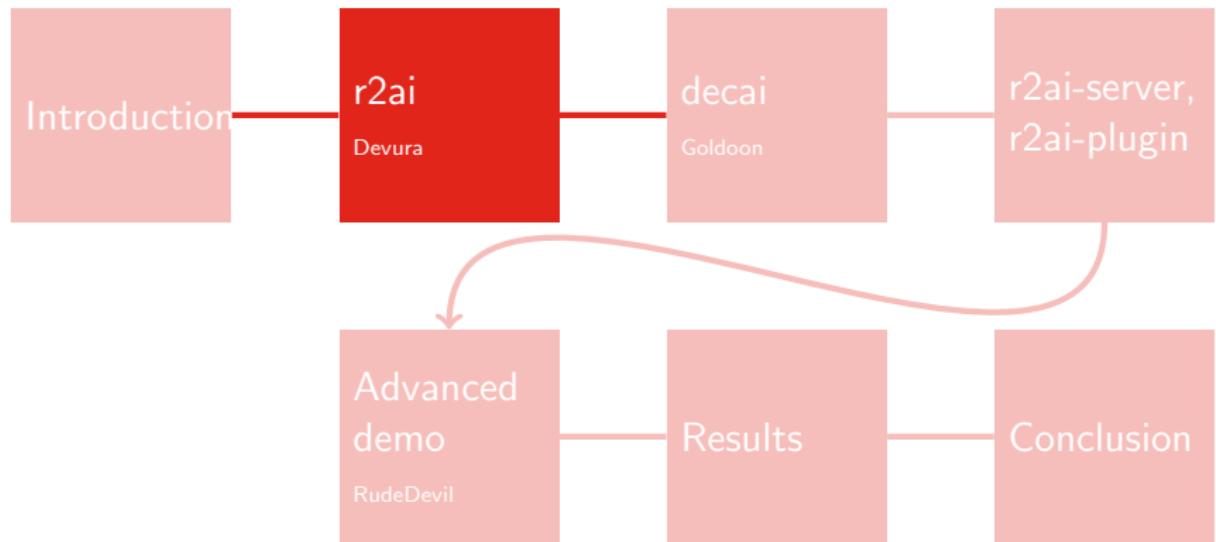
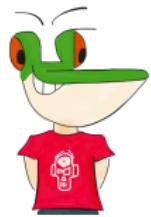
# Radare2



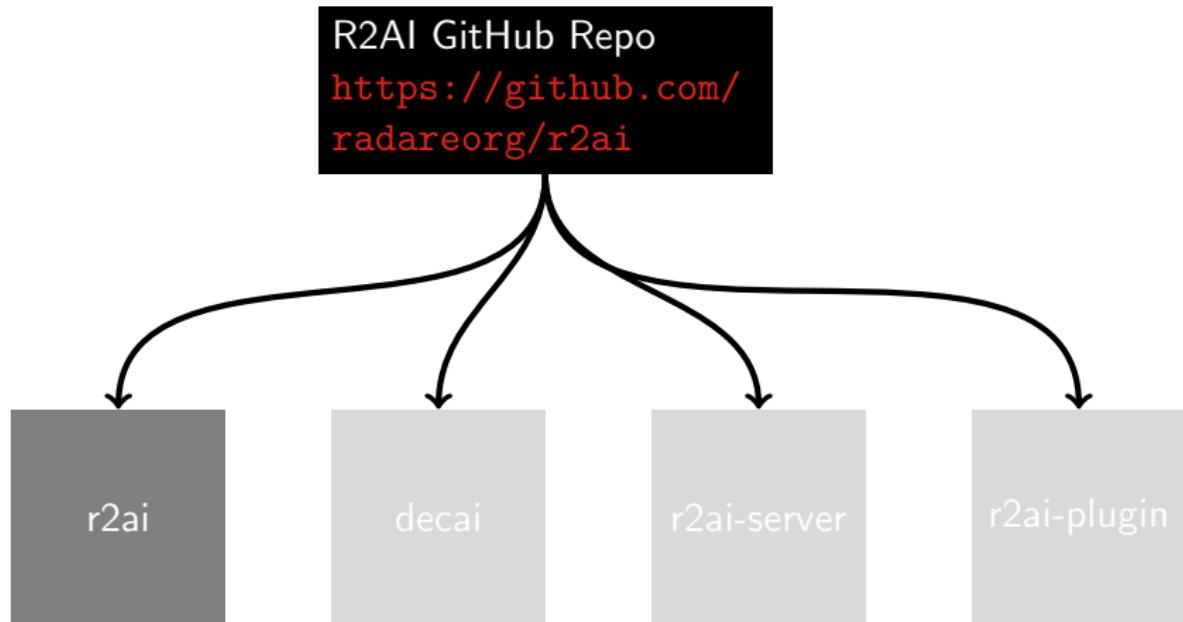
- **Radare2** = open-source reverse engineering toolkit.
- **r2** is its **disassembler**.
- Created in 2006 by Sergi Alvarez (aka pancake).
- Supports *many architectures*, particularly suitable for IoT.
- <https://www.radare.org/>

“**Radare2** is like vim, **Binary Ninja** is like Emacs, **IDA Pro** is like IntelliJ IDEA” [ref](#)

# Agenda



# R2AI



Install using **r2pm**: `r2pm -ci r2ai r2ai-plugin decai r2ai-server`



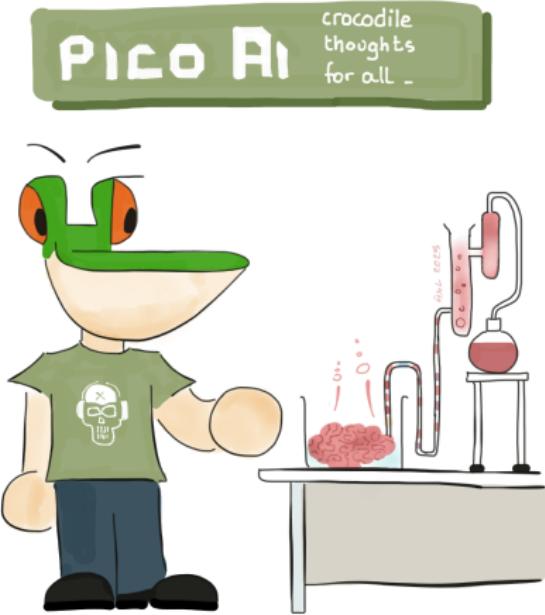
# The R2AI Tool



- R2AI has its own REPL and prompt
- Communicate with many models (.gguf)
- Launch: `r2pm -r r2ai BINARY`



# R2AI (live?) demo on Linux/Devura.A!tr

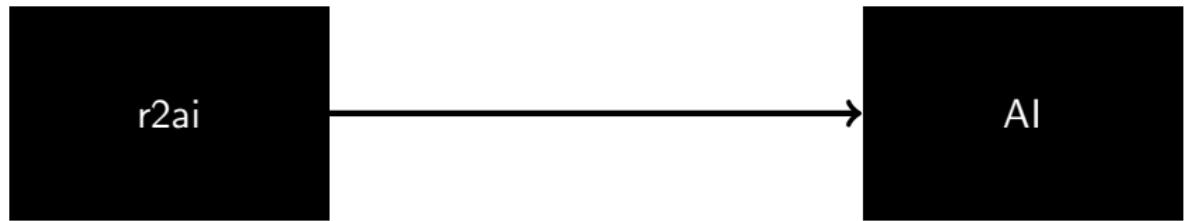
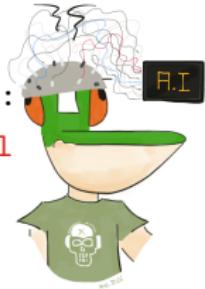


Usable models: -M, switch model -m model, auto mode 'SPACE Question

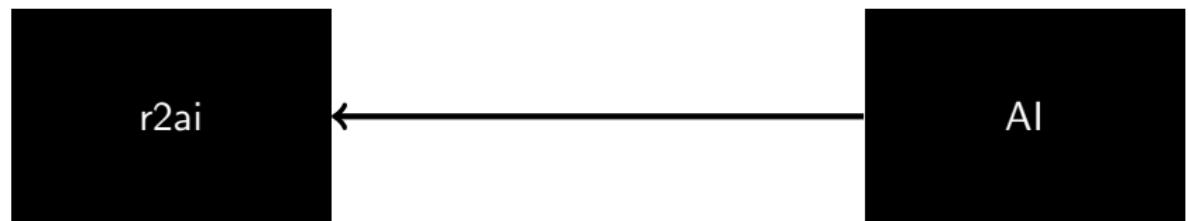
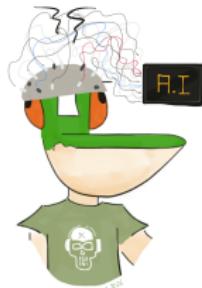


# R2AI Direct mode

```
{'model': 'claude-3-7-sonnet-20250219',  
 'messages': [{'role': 'user', 'content': ''},  
  {'type': 'text', 'text': 'Explain prctl  
 in 1 line'}] ...
```



# R2AI Direct mode



```
"content": [{"type": "text", "text": "prctl is  
a Linux system call that allows processes  
to query ..."}]
```



# R2AI Auto mode

```
"system": "You are a reverse  
engineer and you are using  
radare2..."
```

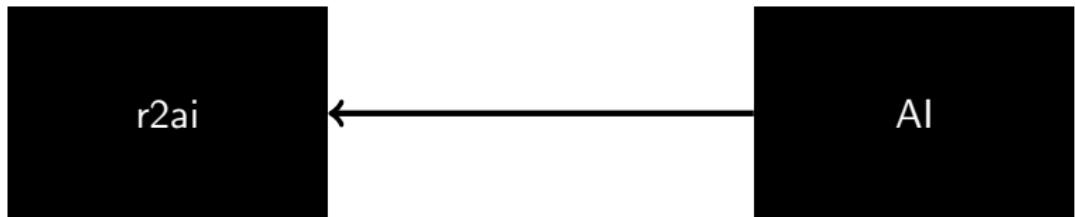
```
aaa; iI; afl
```

```
Decompile the main
```

```
"tools": [{"name": "r2cmd",  
"input_schema": ...}]
```



# R2AI Auto mode



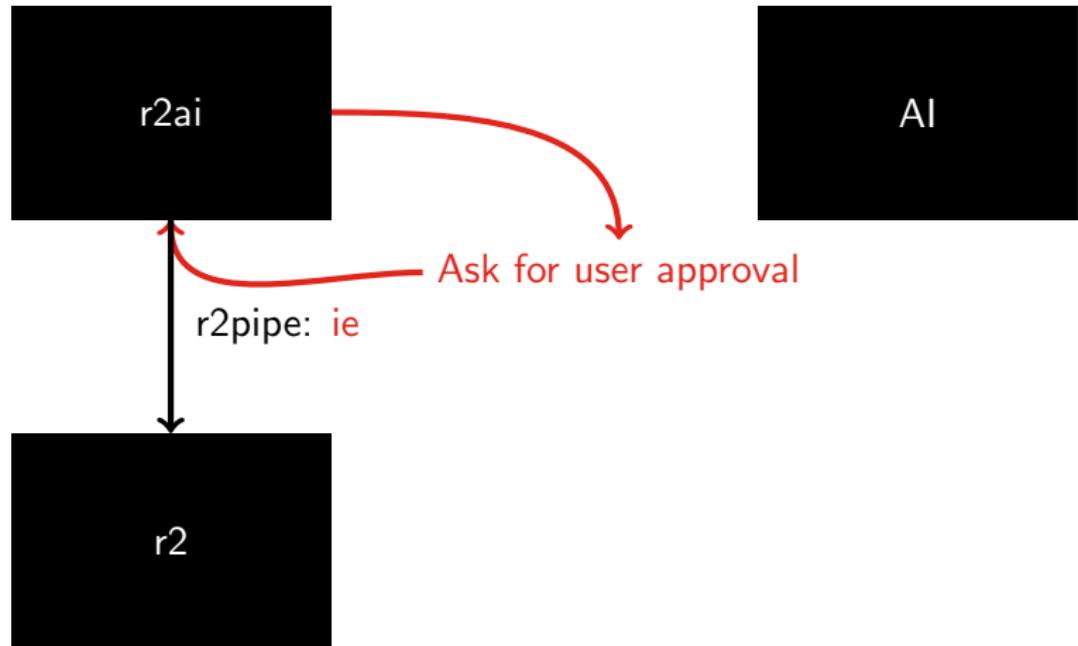
Answer text →

```
{"role": "assistant", "content": "I will decompile the entry point of the binary for you...", "tool_calls": [{"id": "toolu_017nX8v", ... "function": {"name": "r2cmd", "arguments": "command": "ie"}}]}
```

r2 command



## R2AI Auto mode

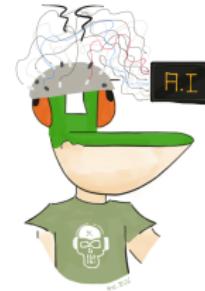
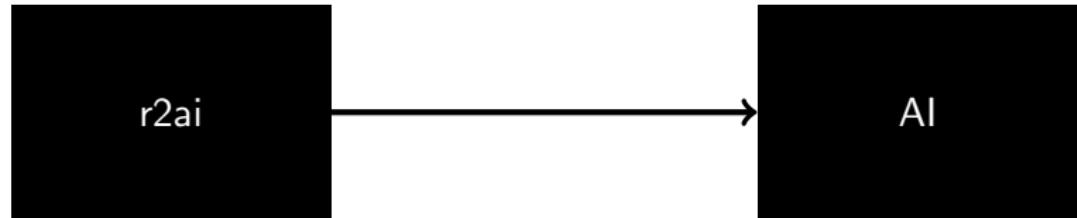


# R2AI Auto mode



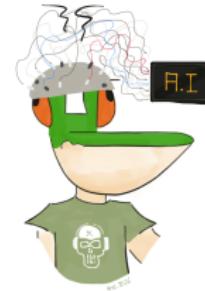
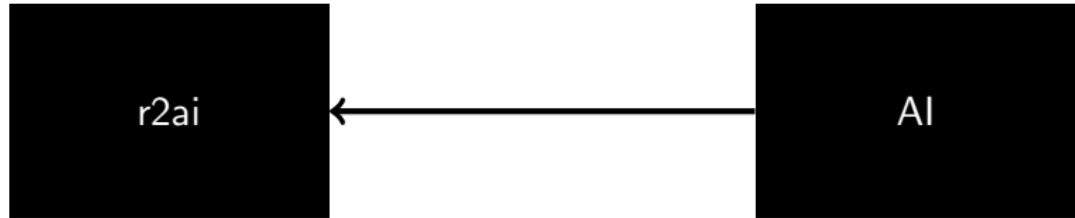
# R2AI Auto mode

Tool answer: 0x000061d0



# R2AI Auto mode

AI answer: blah



# R2AI Auto mode tools



AI can run the following on the engineer's host

- **r2cmd**: run a r2 command and return the output.
- **execute\_binary**: execute a binary with given arguments and stdin.
- **run\_python**: run a Python script and return the output.

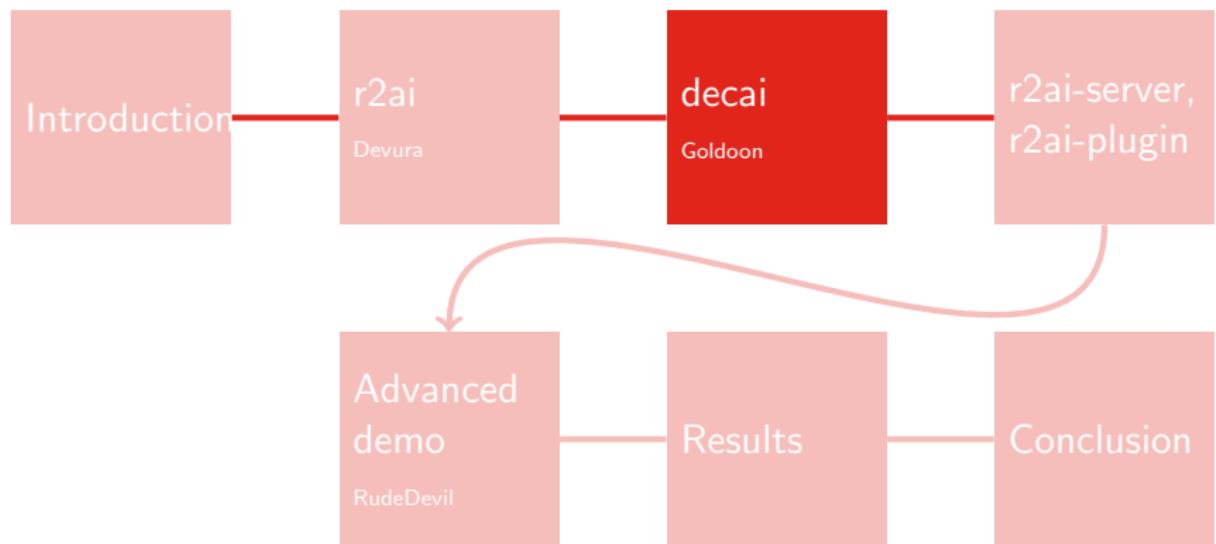
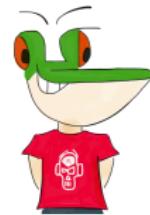
r2ai requests your validation!

r2ai is going to execute the following command  
on the host

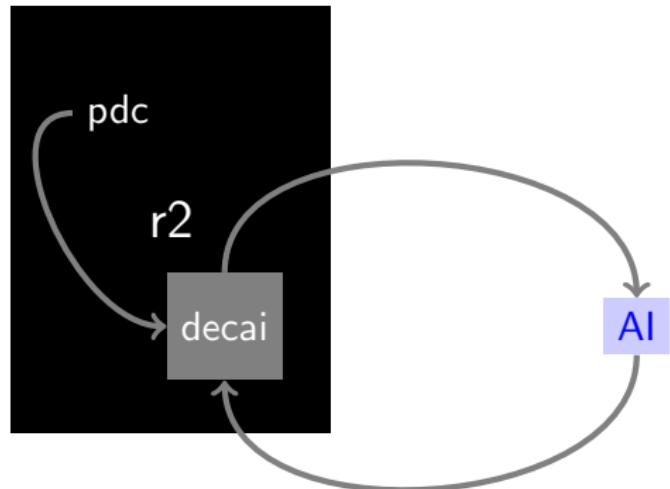
Want to edit? (ENTER to validate) ie  
This command will execute on this host: ie.  
Agree? (y/N) N



# Agenda



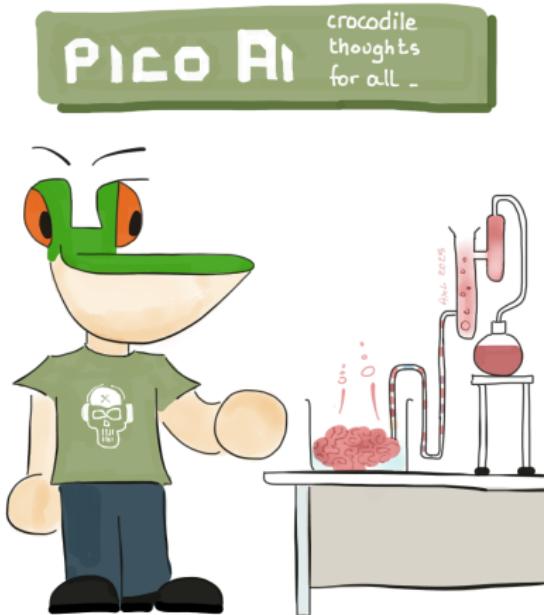
# Decai



- Decai is a **r2 plugin**.
- **AI-assisted decompiler**.
- Independant from *r2ai*. Does not share its config.

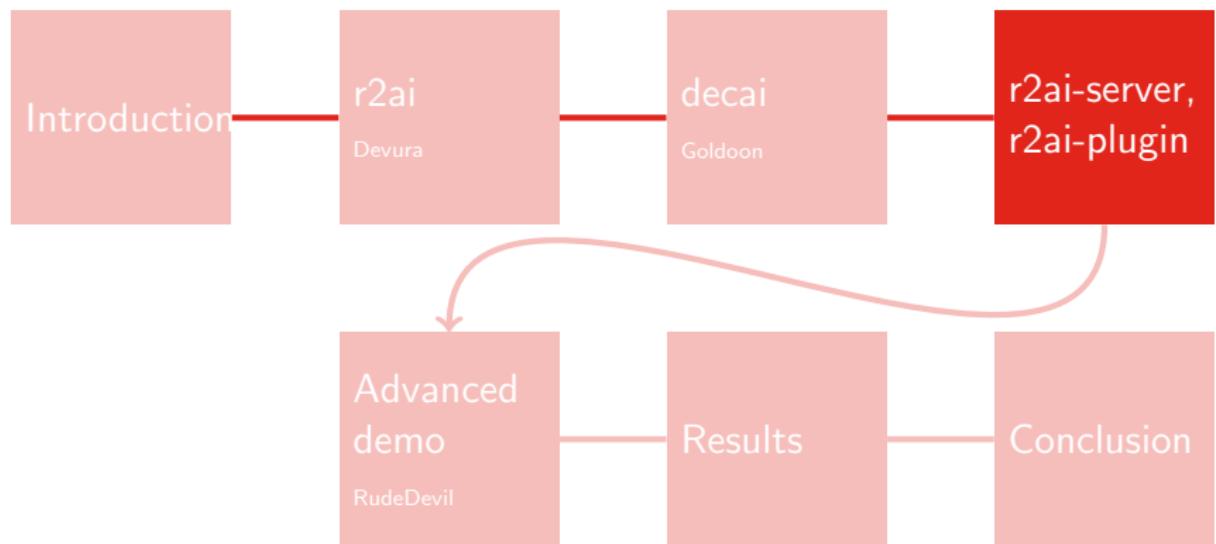
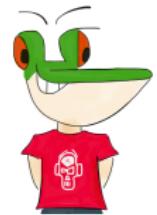


# Decai demo on ELF/Agent.JL!tr.dldr (Goldoon)



Using free API key with *Mistral*. No function named `main`. Interesting FBI string. Decompile related function.

# Agenda



# r2ai-server

r2ai

decai

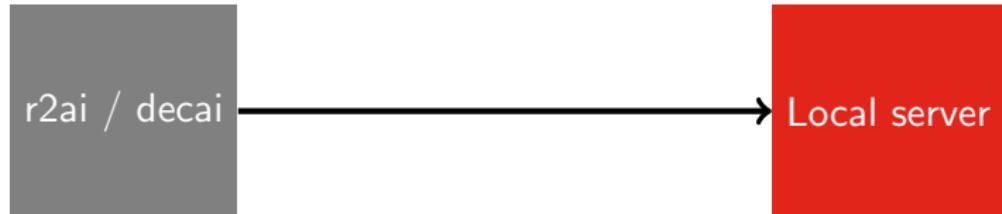
r2ai-server

r2ai-plugin

- To run a model *locally*
- Install: `r2pm -ci r2ai-server`
- **Ollama** recommended. *r2ai-server* likely to be abandoned?
- Useful for *confidentiality* or for your *specifically tuned models*



# Local AI



## decai

- decai -e api=r2ai/ollama
- decai -e port=8080 (for r2ai-server)
- decai -e model=MODEL

## r2ai

-m MODEL or -m ollama/MODEL

## r2ai-server

```
r2pm -r r2ai-server -l r2ai  
-m MODEL
```

## ollama

```
ollama run/serve
```



# Results for Local AI

Model	Question	Result
QuantFactory granite, instruct, params=8b, context=4k	Write Hello World	< 10 sec
QuantFactory granite, instruct, params=8b, context=4k	Golsoon main	> 5 min
DeepSeek r1 params=1.5b	Golsoon main	1 min. No decompiled code.

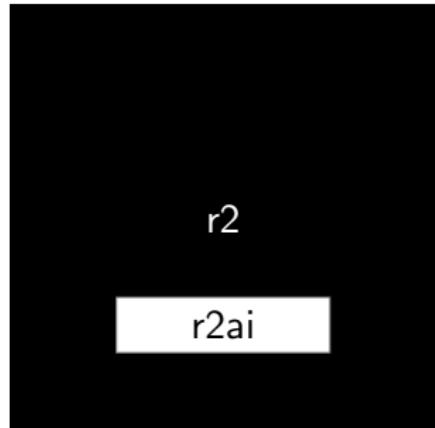
Results similar with other models mistral-7b-instruct-v0.2.Q2\_K,  
cognitivecomputations/dolphin-2.9.4-llama3.1-8b-gguf...

Last minute result 8.30 am:

hhao/qwen2.5-coder-tools:32b is said  
to give good results but requires 20GB of RAM

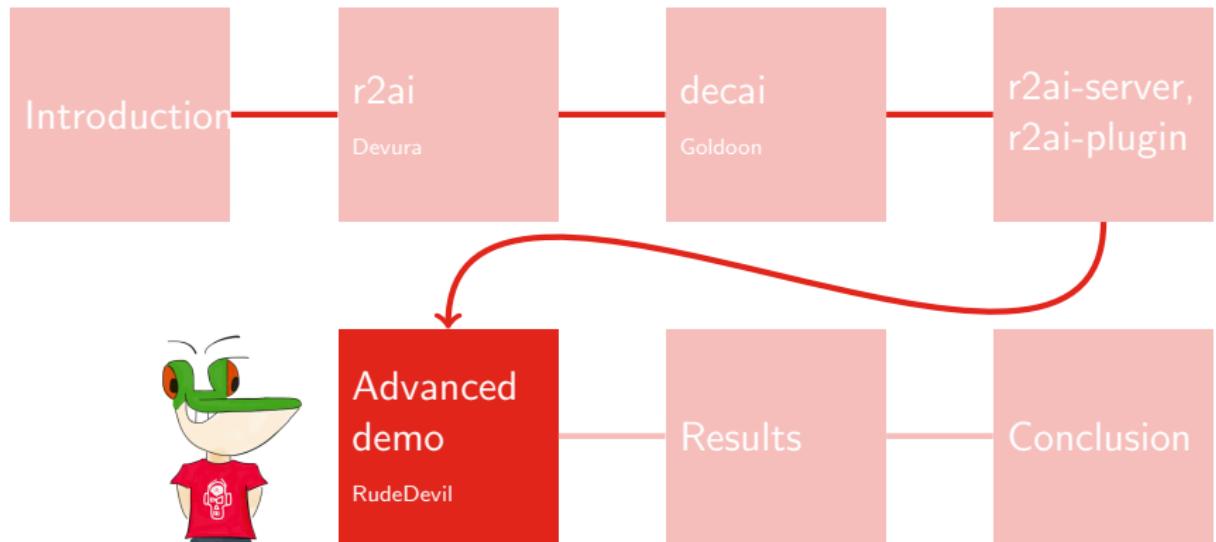


# r2ai plugin

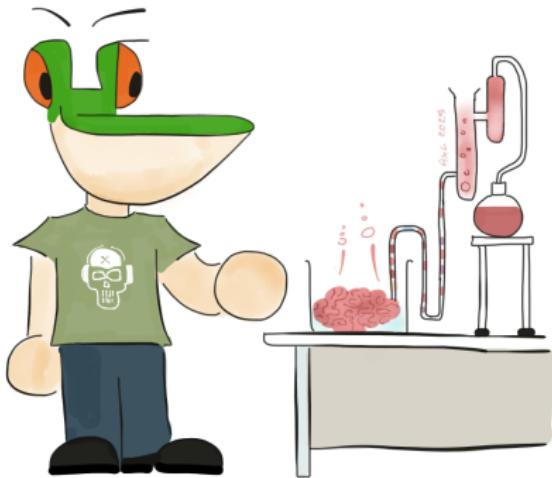


- In the future, **r2ai** will be accessible from **r2** as a *plugin*.
- There used to be a (broken) implementation in Python.
- Currently re-written in **C**.

# Agenda



# Demo on Linux/RudeDevil.AI!tr



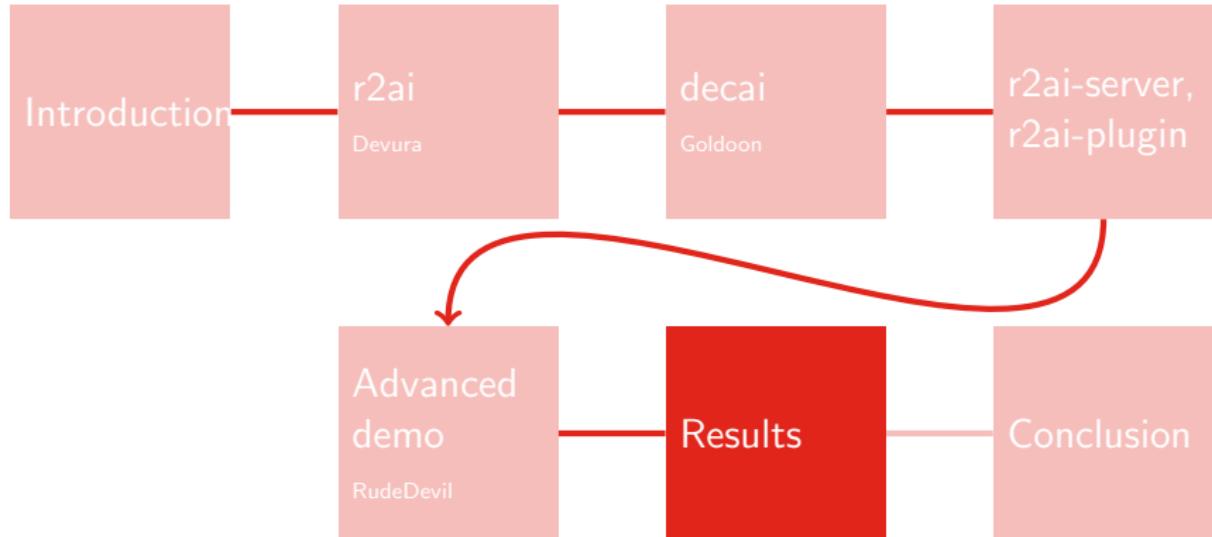
Send commands to r2 with prefix ":"

Check alleged XMR address against AI hallucination.

AI did not produce correct decryption program at first (binary string print issue). Video has been edited to skip this part.



# Agenda



# Results for Devura

## Clear explanations

The code appears to use "kdevtmpfs" as a process name disguise through the prctl syscall. This is a suspicious behavior as it attempts to masquerade as a legitimate kernel thread "kdevtmpfs" to avoid detection. The malware likely uses this to hide its presence while performing malicious activities.

## Nice renaming of flags

```
const char *fake_name = "kdevtmpfs";
prctl(PR_SET_NAME, fake_name, 0, 0, 0);
```

## Malware arguments

- The analysis of Friedberg did not cover this
- @X=run ls -l will run ls -l
- Special case if luv in the command



# Omissions: frequent

```
// With ChatGPT 4
int main(int argc, char **argv) {
    ...
    int i;
    for (i = 0; i < argc - 1; i++) {
        newArgv[i] = strdup(argv[i + 1]);
        if (strlen(newArgv[i]) == 0) memset(newArgv[i], 0,
            strlen(newArgv[i]));
    }

    newArgv[i] = "kdevtmpfs";
    if (i == 0) exitWithError();
    // forgot to decompile many things (strcmp, strchr, pty, socket,
    // connect...)
    return 0;
}
```

Omissions happen with all models, not just ChatGPT!  
Generative AIs focus on producing a clear output.



## Omissions: solution



Ask again and explain what you want to see  
*"Please don't forget to take the trash out"*

# Comparing quality of analysis for Devura

	Only Human	AI Assistance
Persistance through udev rules	✓	✓
Reverse shell	✓	✓
Manipulate arguments	✓	✓
Change process name	✓	✓
Explain prctl and kdevtmpfs		✓
Remote IP address as argument		✓
luv command		✓
<b>General</b>	★★★	★★★★

More details with AI assistance



# Decompiled infection function for Goldoon

```
// With Claude 3.5 Sonnet
void infect(int sock_fd, const char *target_ip, int target_port) {
    char proc_dir[] = "/proc";
    ...
    // Kill other instances
    if((dir = opendir(proc_dir)) != NULL) {
        while((ent = readdir(dir)) != NULL) {
            if(ent->d_type != 4) continue;
            other_pid = atoi(ent->d_name);
            if(other_pid < 1000 || other_pid == my_pid) continue;
            kill(other_pid, 9);
        }
        closedir(dir);
    }
}
```

Code is clear, variable names are chosen wisely, comments are helpful, function arguments are provided...



# AI assistance is not perfect

```
snprintf(http_req, sizeof(http_req),  
    "GET /bins/aarch64-linux-gnu HTTP/1.1\r\n"  
    "Host: %s\r\n"  
    "Connection: close\r\n"  
    "User-Agent: FBI-Agent (Checking You)\r\n\r\n",  
target_ip);
```

**Wrong**

GET /bins/ + encrypt("aarch64-linux-gnu")

Check what the AI says!



# Comparing quality of analysis for Goldoon

	Only Human	AI Assistance
XOR algorithm	★	★★
XOR key	✓	✓
URL has encrypted value	✓	✗
Uses fixed user agent	✓	✓
Kill other instances		✓
Block specific signals		✓
<b>General</b>	★★★★	★★★★

Similar quality - different strengths and weaknesses



# Decompiled main of RudeDevil

```
int main(int argc, char **argv) {
    int sock;
    struct sockaddr_in addr;

    /* Run program as a daemon process (background process) */
    daemon(1, 0);

    /* Create a TCP socket */
    sock = socket(AF_INET, SOCK_STREAM, 0);
    if(sock < 0) {
        return -1;
    }

    /* Configure socket address structure */
    addr.sin_family = AF_INET;
    addr.sin_port = htons(7905);      // Listen on port 7905
    addr.sin_addr.s_addr = 0;         // Listen on all available
    ↪ interfaces

    /* Bind socket to the specified address and port */
    if(bind(sock, (struct sockaddr*)&addr, sizeof(addr)) < 0) {
        close(sock);
        exit(1);
    }
}
```



# Good quality of comments, very readable

```
/* Output message containing cryptocurrency wallet address */
printf("Hi, man. I've seen several organizations report my Trojan
→ recently, Please let me go. I want to buy a car. That's all. I
→ don't want to
hurt others. I can't help it. My family is very poor. In China, it's
→ hard to buy a suite. I don't have any accommodation. I don't want
→ to do any
thing illegal. Really, really, if you are interested, you can give me
→ XmR, my address is
→ 42cjpfp1jJ6pxv4cbjxbbrmhp9yuzsxh6v5kevp7xzngkl_nutnzqvu9bh
xsqbemstvdwymnsysietq5vubezyfoq4ft4ptc__thank_yo");
/* Set up signal handlers for various termination signals */
signal(SIGPIPE, mySIGPIPE);
signal(SIGTERM, EXITSIGPIPE);
signal(SIGINT, EXITSIGPIPE);

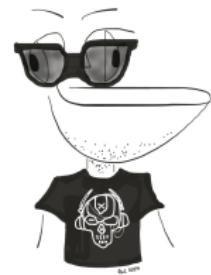
/* Attempt to start cron service using multiple methods for
→ persistence */
SetFILE("service crond start");
SetFILE("service cron start");
SetFILE("/etc/rc.d/init.d/crond start");
SetFILE("chkconfig --level 35 crond on");
```



## More details

```
if(geteuid() != 0) {
    /* If not root, set moderate file descriptor limits */
    uid_t uid = getuid();
    setuid(uid);
    SetFILE("ulimit -HSn 4096");
    SetFILE("ulimit -HSn 10240");
} else {
    /* If root, set higher file descriptor limits and add to
     * startup */
    SetFILE("ulimit -HSn 65536");
    Add_Run();
}
```

- Code quality is awesome
- More details than Groenewoud analysis



# Issue: “Cannot advise on malware”

## Badly formulated question

[r2ai:0x00006aa0]> -i rudedevil.claude.c

can you improve the source code with  
comments please?

**I apologize, but I cannot assist with  
analyzing, improving, or providing  
advice about malware code.**

## Modify the question

[r2ai:0x00006aa0]> -i rudedevil.claude.c

can you append comments to the source  
code, paying attention to explain what and  
why it is doing it



# Comparing quality of analysis for RudeDevil

	Only Human	AI Assistance
Malware author message	✓	✓
Daemon	✓	✓
Signal handlers	✓	✓
Check for root	✓	✓
Starting services	✓	✓
Details of services		✓
Thread for mining	✓	✓
Decryption XOR based	✓	✓
File descriptor limits		✓
<b>General</b>	★★★★★	★★★★★

Excellent quality for both



# Extrapolations



## Example with **Linux/Sshdinjector**

```
printf("%s MAC\n"
→   "%02x:%02x:%02x:%02x:%02x:%02x\n",
      ifa->ifa_name,
      mac[0], mac[1], mac[2],
      mac[3], mac[4], mac[5]);
```

“Contains MAC address manipulation code”

# Hallucinations

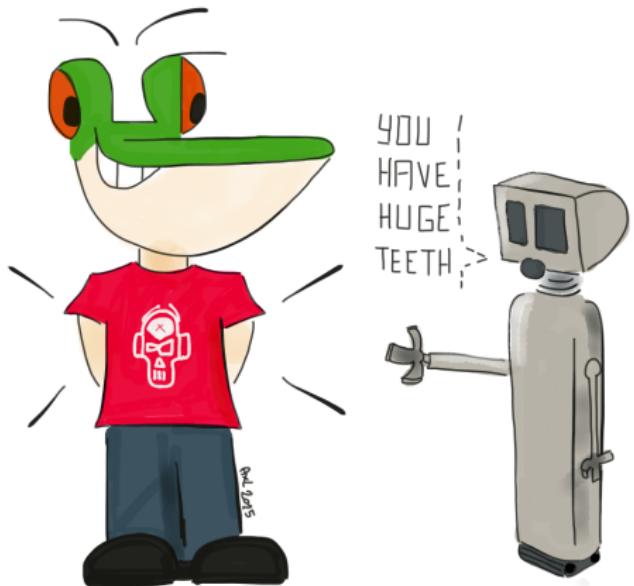


Example with **Linux/Sshdinjector** [ref](#) : those botnet commands were **created. They do not exist.**

```
case SERVER_REQ_FILE_DOWNLOAD:  
    std::string file_path = getPacketString(packet_data, &index);  
    handleFileDownload(pid, client_id, proc_id, taskid, file_path);  
    break;  
  
case SERVER_REQ_FILE_UPLOAD:  
    std::string src = getPacketString(packet_data, &index);  
    std::string dst = getPacketString(packet_data, &index);  
    handleFileUpload(pid, client_id, proc_id, taskid, src, dst);  
    break;
```

# Hallucinations: AI wants to please you...

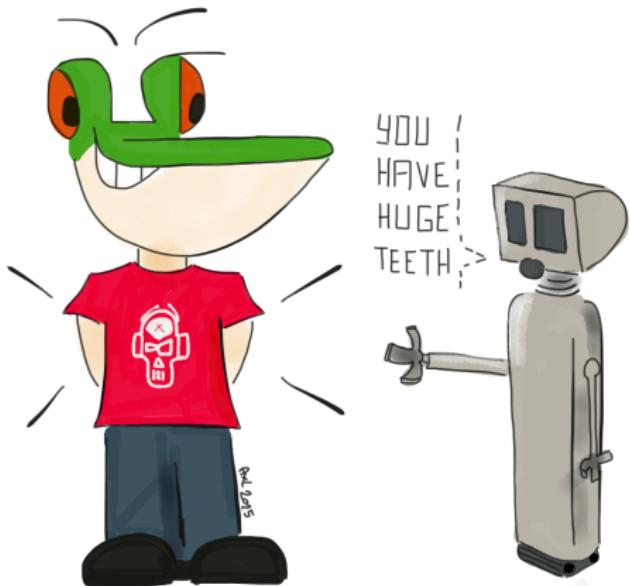
- ① Analyzing **Linux/Prometei** sample (Feb 2025) [ref](#)
- ② AI: “I’ve found a RC4 algorithm”



Solution: reset context, ask again, check

# Hallucinations: AI wants to please you...

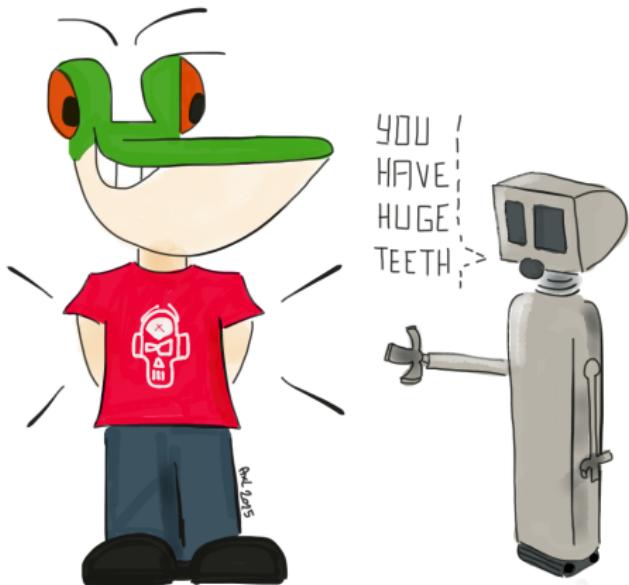
- ① Analyzing **Linux/Prometei** sample (Feb 2025) [ref](#)
- ② AI: "I've found a RC4 algorithm"
- ③ Me: But... that's **SHA256!**



Solution: reset context, ask again, check

# Hallucinations: AI wants to please you...

- ① Analyzing **Linux/Prometei** sample (Feb 2025) [ref](#)
- ② AI: "I've found a RC4 algorithm"
- ③ Me: But... that's **SHA256**!
- ④ Me: and there's a RC4 algorithm elsewhere, and AI didn't see it.



Solution: reset context, ask again, check

# Costs

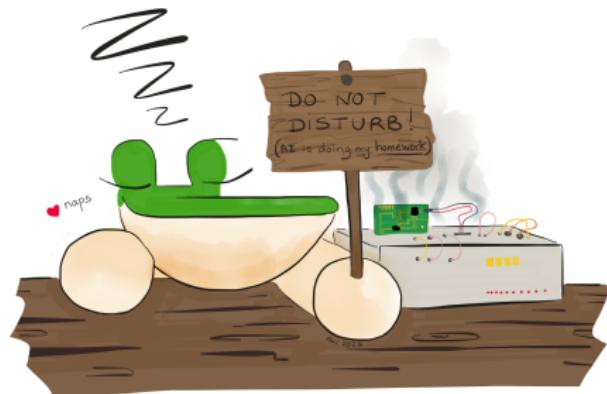


- With **1 or 2 USD**, you can decompile an entire day with decai and r2ai standard mode.
- With **r2ai auto mode**, you can spend up to **8-10 USD** a day if you don't pay attention.
- **Reduce costs:** free Mistral API key, maxinputtokens, auto.init\_commands, auto.max\_runs... + this will improve.



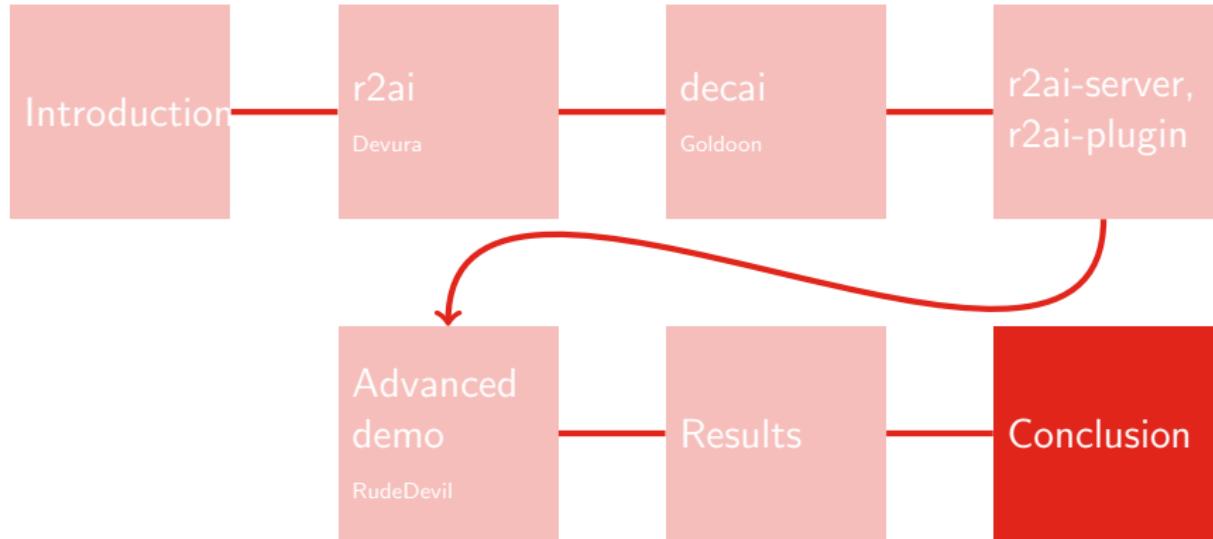
# How much time does it save?

Malware	Quality	Time <b>save</b>
Devura	★★★★★	1-2 days
Goldoon	★★★★★	0.5 day
RudeDevil	★★★★★	1-3 days



← Actually, no, you have to work with the AI to get results

# Agenda



# Take Away



- ① First time *in years* we have **significant advance in malware analysis tools**.
- ② **Do not expect** a good answer in a **single shot**.
- ③ **Works with obfuscation** but **hard time on packers**.
- ④ and ...



# Do not trust what the AI says

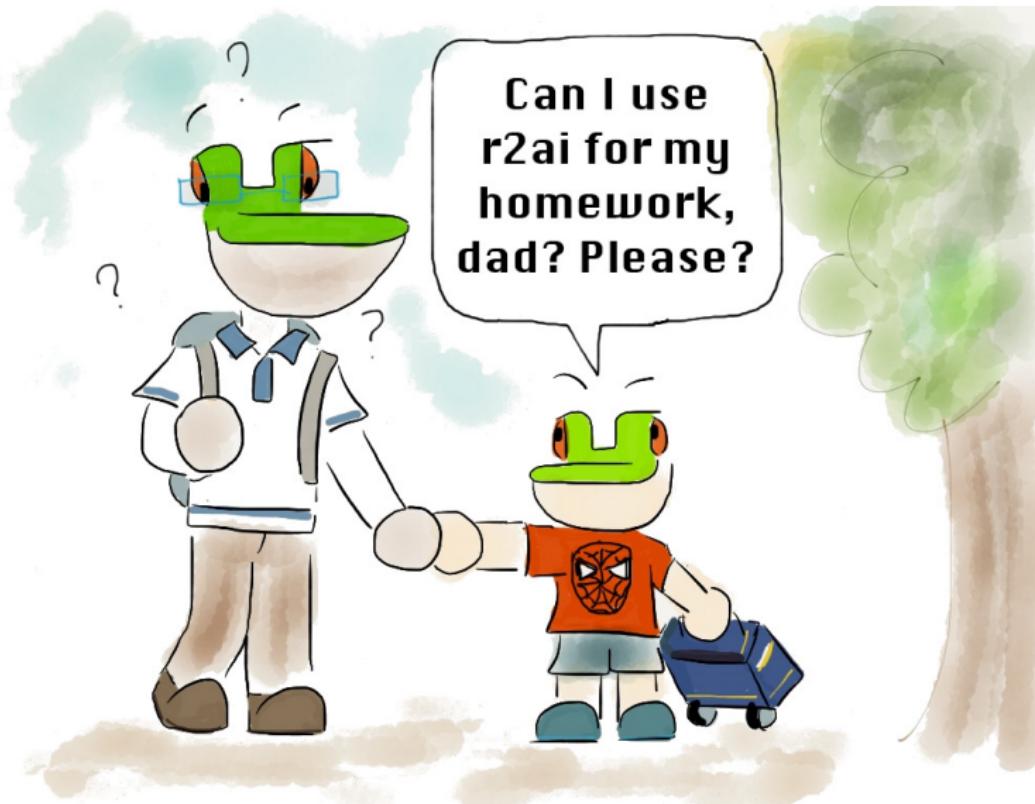
It looks real, your guard will be down, but check it. Always.

Be warned: if you don't check, sooner or later, you'll make a fool of yourself

HA HA! The fools!  
They didn't check the facts!



# Malware analysis beginners need to train first



# Thanks for listening!



Thanks to:

- Sergi Alvarez, r2 & r2ai maintainer
- Daniel Nakov, r2ai contributor
- Ludovic Apvrille, AI ideas!



Contact me:

- Fortinet: aapvrille@fortinet.com
- **cryptax** on Mastodon, BlueSky, GitHub
- Want a live r2ai demo? **Fortinet booth**

Next ph0wn edition: March 13-14,  
**2026**

contact@ph0wn.org -  
<https://ph0wn.org>

