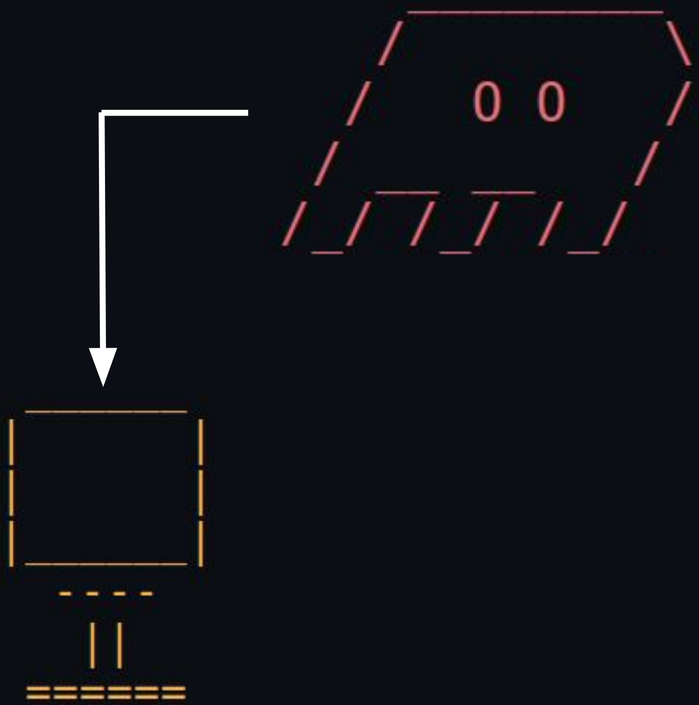


Capstone project: Semargl

Semargl C2 Framework

Versatile Command-and-Control for Red Team Operations

Team members:

- Artur Lukianov
- Alexander Efremov
- Vadim Yarullin
- Andrew Boronin
- Alexander Tomashov
- Nika Chekhonina
- Viktoria Patrina

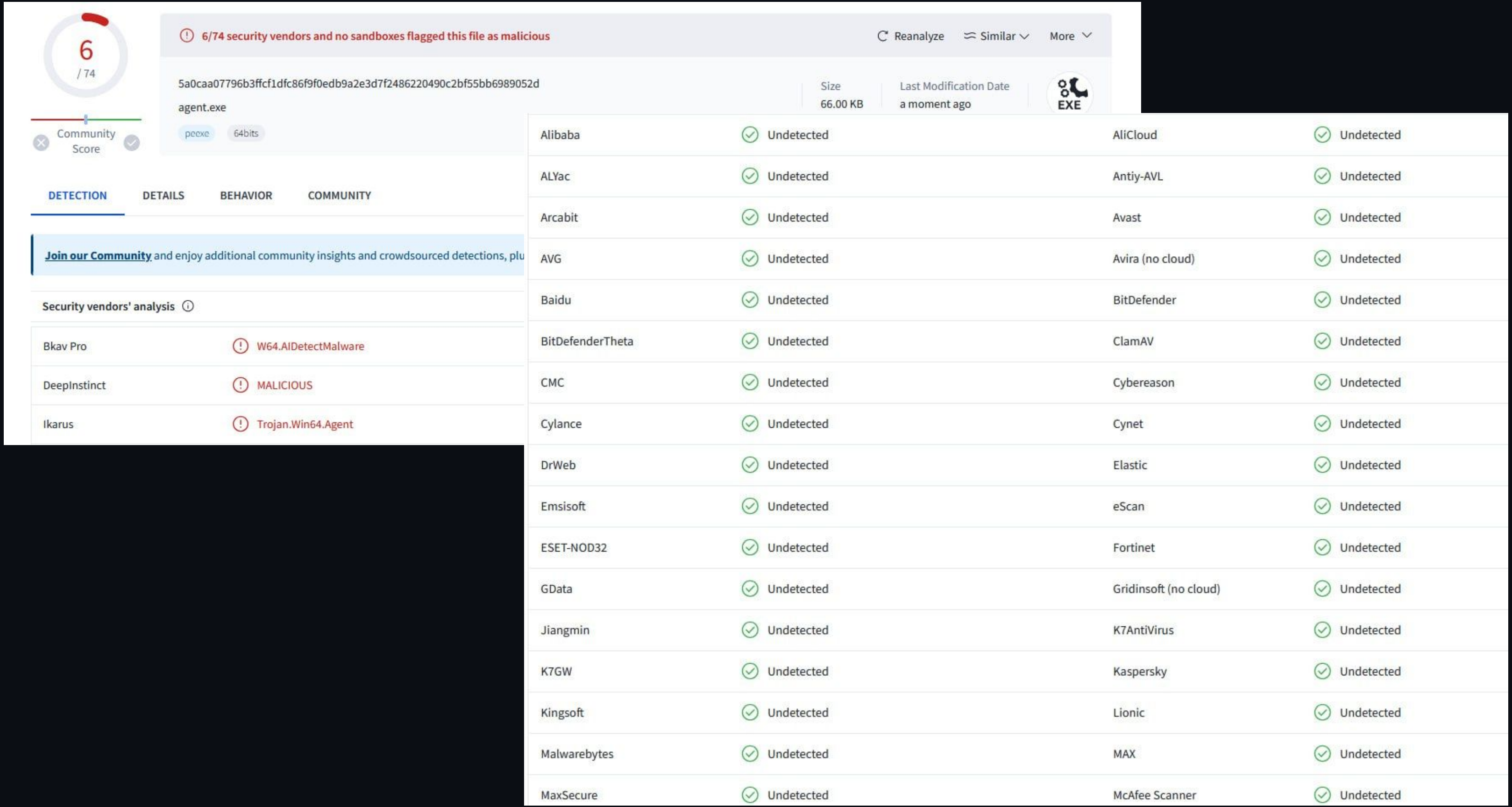
The problem

The cybersecurity industry faces significant challenges in conducting full adversary emulations. Current tools provide a closed and untrusted environment and lack the techniques used by cyber-criminals, making it difficult to effectively test Security Operation Centers.

Our solution

Introducing Semargl C2 Framework - solution for red team operations. It supports modules for planning, execution, and reporting.

Quick Demo



6
/ 74

Community Score

6/74 security vendors and no sandboxes flagged this file as malicious

5a0caa07796b3ffcf1dfc86f9f0edb9a2e3d7f2486220490c2bf55bb6989052d

agent.exe

peexe 64bits

Size
66.00 KB

Last Modification Date
a moment ago

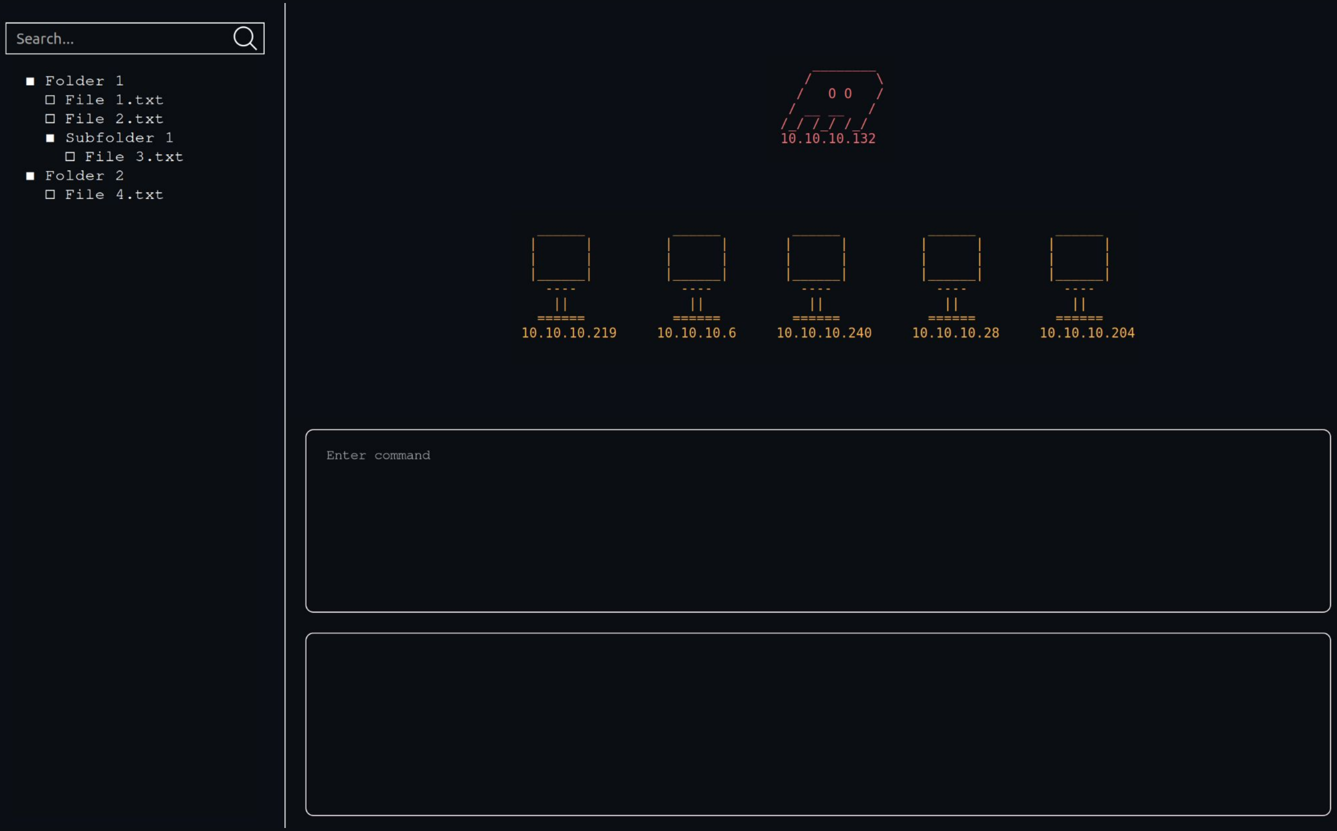
Join our Community and enjoy additional community insights and crowdsourced detections, plu

Security vendors' analysis

Bkav Pro	W64.AIDetectMalware
DeepInstinct	MALICIOUS
Ikarus	Trojan.Win64.Agent

Alibaba	✓ Undetected	AliCloud	✓ Undetected
ALYac	✓ Undetected	Antiy-AVL	✓ Undetected
Arcabit	✓ Undetected	Avast	✓ Undetected
AVG	✓ Undetected	Avira (no cloud)	✓ Undetected
Baidu	✓ Undetected	BitDefender	✓ Undetected
BitDefenderTheta	✓ Undetected	ClamAV	✓ Undetected
CMC	✓ Undetected	Cybereason	✓ Undetected
Cylance	✓ Undetected	Cynet	✓ Undetected
DrWeb	✓ Undetected	Elastic	✓ Undetected
Emsisoft	✓ Undetected	eScan	✓ Undetected
ESET-NOD32	✓ Undetected	Fortinet	✓ Undetected
GData	✓ Undetected	Gridinsoft (no cloud)	✓ Undetected
Jiangmin	✓ Undetected	K7AntiVirus	✓ Undetected
K7GW	✓ Undetected	Kaspersky	✓ Undetected
Kingsoft	✓ Undetected	Lionic	✓ Undetected
Malwarebytes	✓ Undetected	MAX	✓ Undetected
MaxSecure	✓ Undetected	McAfee Scanner	✓ Undetected

GUI



- ■ Folder 1
  - □ File 1.txt
  - □ File 2.txt
  - ■ Subfolder 1
    - □ File 3.txt
- ■ Folder 2
  - □ File 4.txt

Development Process

Our journey began with identifying the need for a robust C2 framework. We faced several challenges, including ensuring communication, integrating multi-language agents, and achieving scalability. We overcame these with solutions like secure gRPC, agent modules and dynamic generation.

Team Contributions

- Artur Lukianov (Lead): Server and Client development, gRPC design
- Alexander Efremov: Agent development in C with WinAPI
- Vadim Yarullin: Agent development in Rust/Powershell with WinAPI
- Andrew Boronin: Docker, Ansible, CI/CD
- Alexander Tomashov, Nika Chekhonina, Viktoria Patrina: UI design and implementation

Technical Stack

- Our technical stack includes Golang and gRPC for both backend and client components, ensuring high efficiency and scalability.
- The agents are developed in C and Rust with WinAPI for low-level access and stealth.
- We utilize Docker and Kubernetes for deployment and scalability, and Ansible for configuration management.
- We use Golang with Wails and View framework for GUI.

Architecture

The Semargl C2 architecture consists of three main components: the Server (manages command dispatch, data aggregation, orchestration), the Client (interfaces with the server to receive commands and send back data), and the Agent (executes commands on target machines, collects data, and ensures covert operations)

Key Features and Benefits

- Integration with MITRE ATT&CK framework
- Utilizes gRPC for efficient and reliable network communication.
- Supports agents written in Rust, C, and PowerShell for diverse environment compatibility.

User Impact

Semargl C2 provides significant benefits for Red Teamers, Penetration Testers, and Businesses. It offers real-world use cases such as external and internal adversary emulation, penetration testing, and cybersecurity training, making engagements more transparent and robust.

Feedback and Validation

We gathered feedback through beta testing, user surveys, and controlled testing environments. Key insights include high user satisfaction, and the need for real-time monitoring features. This feedback has been instrumental in shaping the development process.

Future Plans

Our future roadmap includes developing a graphical interface, expanding documentation, implementing real-time monitoring, adding modules, and continuously improving the framework based on user feedback. Our long-term vision is to create a user-centric, scalable, and flexible C2 framework that aligns with the evolving needs of cybersecurity professionals.

Conclusion

In conclusion, the Semargl C2 framework addresses the critical need for a robust and versatile C2 solution in red team operations. We have developed a transparent, secure, and scalable framework that benefits both cybersecurity professionals and businesses. We invite your questions and feedback to further enhance our project.