UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
Scuola Politecnica e delle Scienze di Base
Corso di Laurea Magistrale in Ingegneria Informatica

tesi di laurea magistrale

Laccolith: A Novel Approach for Anti-Detection in Adversary Emulation

Anno Accademico 2021/2022

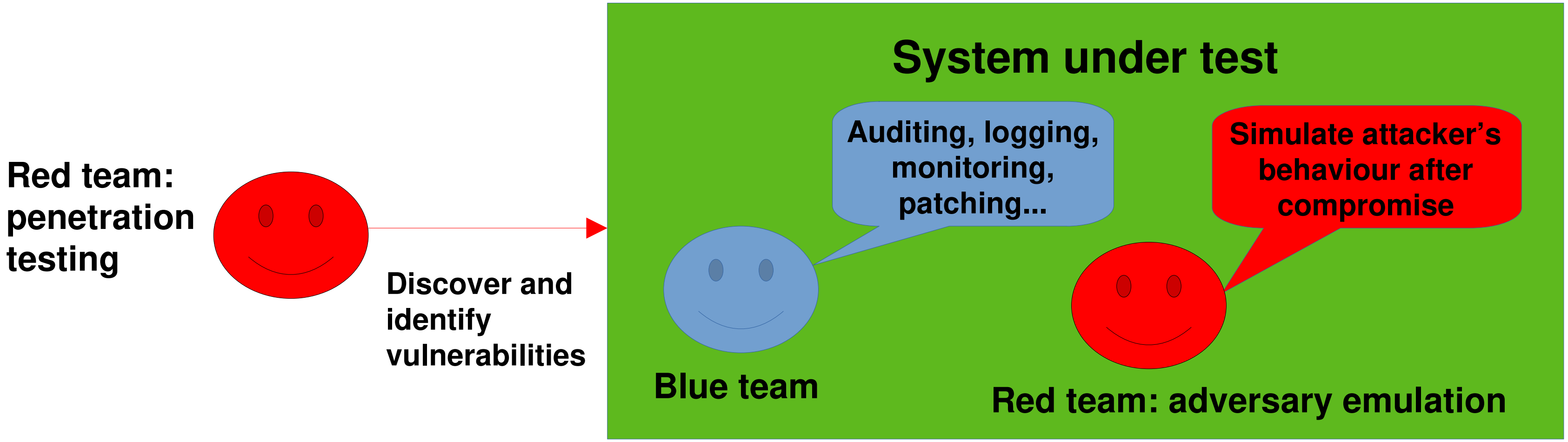
relatore
Prof. Roberto Natella

correlatore
Ing. Vittorio Orbinato

candidato
Marco Carlo Feliciano
Matr. M63001136

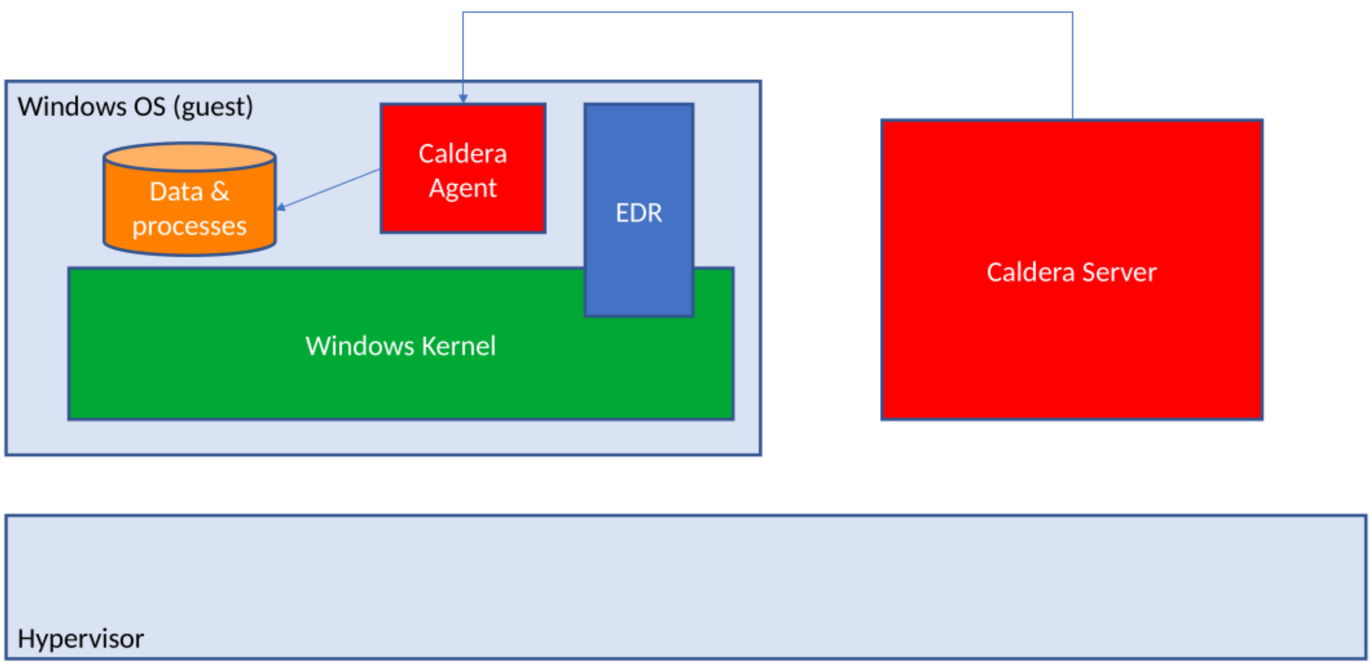
Adversary emulation

It is a type of red (or purple) team engagement that uses real-world threat intelligence to impersonate the actions and behaviours that your red team (or bad actors) would use in practice [1].



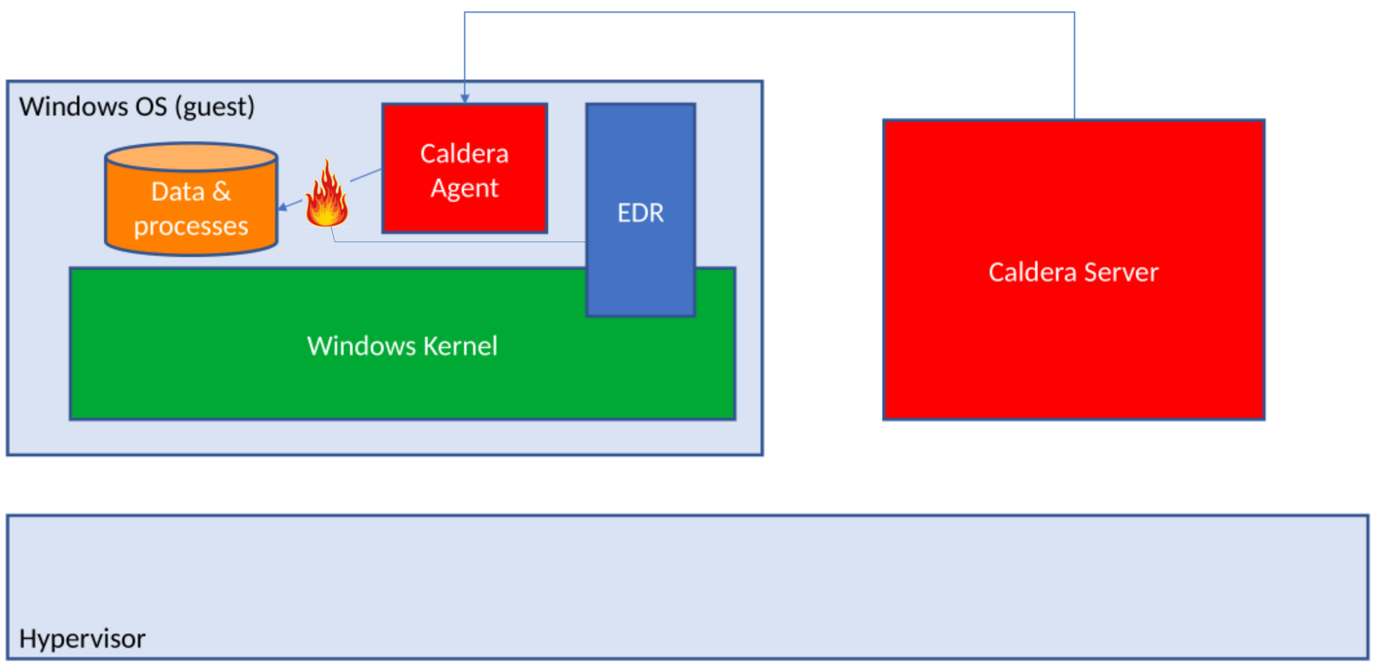
Adversary emulation: example scenario (1/2)

MITRE's Caldera is one of the most popular adversary emulation frameworks [2].



Adversary emulation: example scenario (2/2)

Caldera, as other popular frameworks, lacks defense evasion capabilities.

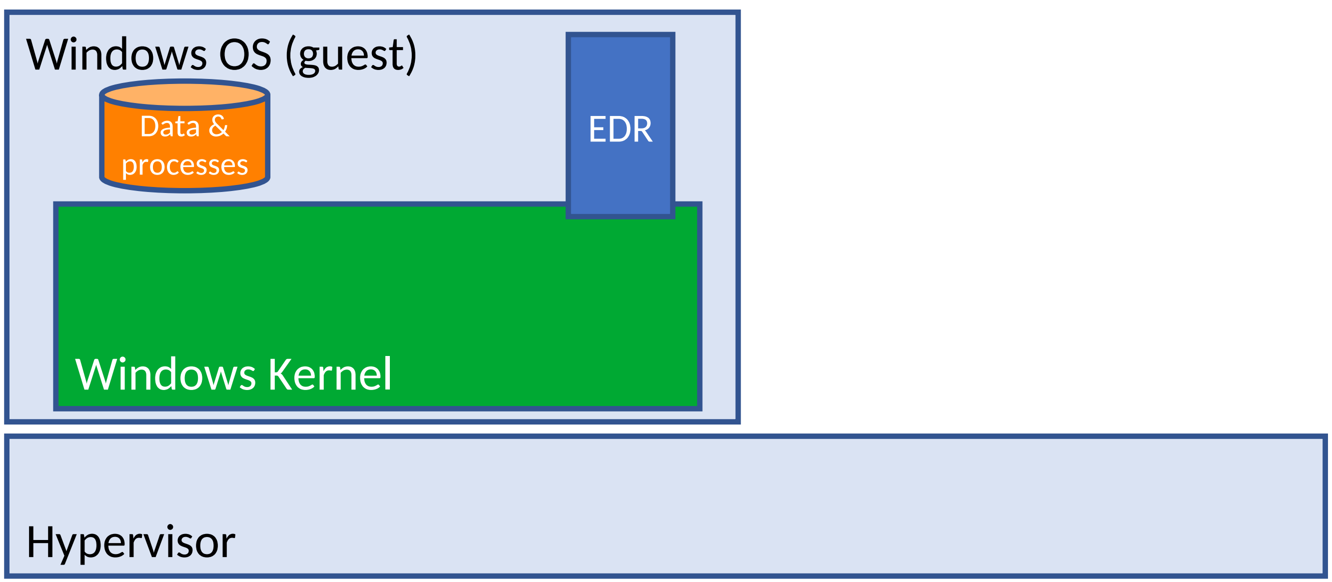


Contribute: addressing the problem

- **Experiments** to show the lack of anti-detection capabilities of popular tools, with different AV/EDR configurations;
- **Design** of a novel approach for anti-detection, by considering injection of the agent from the hypervisor;
- **Implementation** of a prototype (“Laccolith”) for the novel approach, for Linux hypervisor and Windows 10 guest;
- **Experiments** to show that the prototype for the novel approach is able to perform its actions undetected, with different AV/EDR configurations.

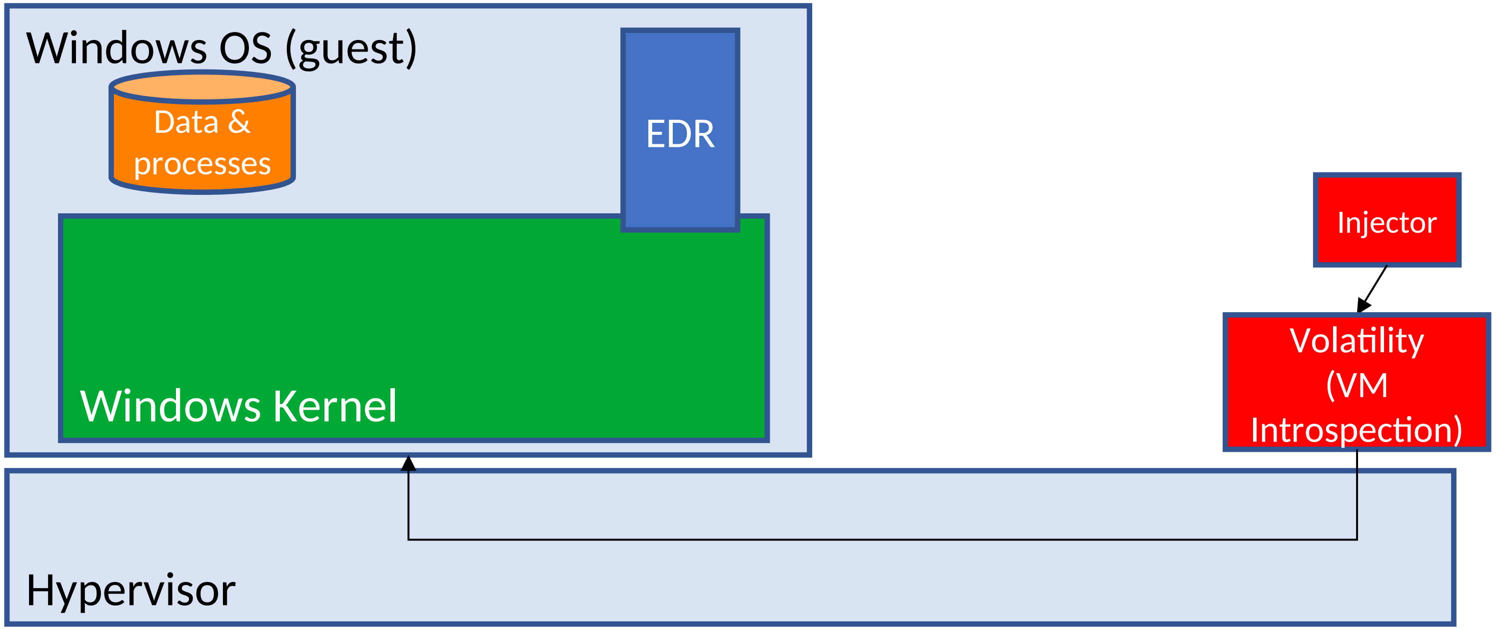
Novel approach (1/4)

1) Hypothesis: injection from hypervisor (it's a real threat [3]);
2) Use the hypervisor's privilege to gain arbitrary code execution in a virtual domain;
3) Use the code execution "primitive" to inject an *agent* capable of high-level actions, after connecting to the C2 server.



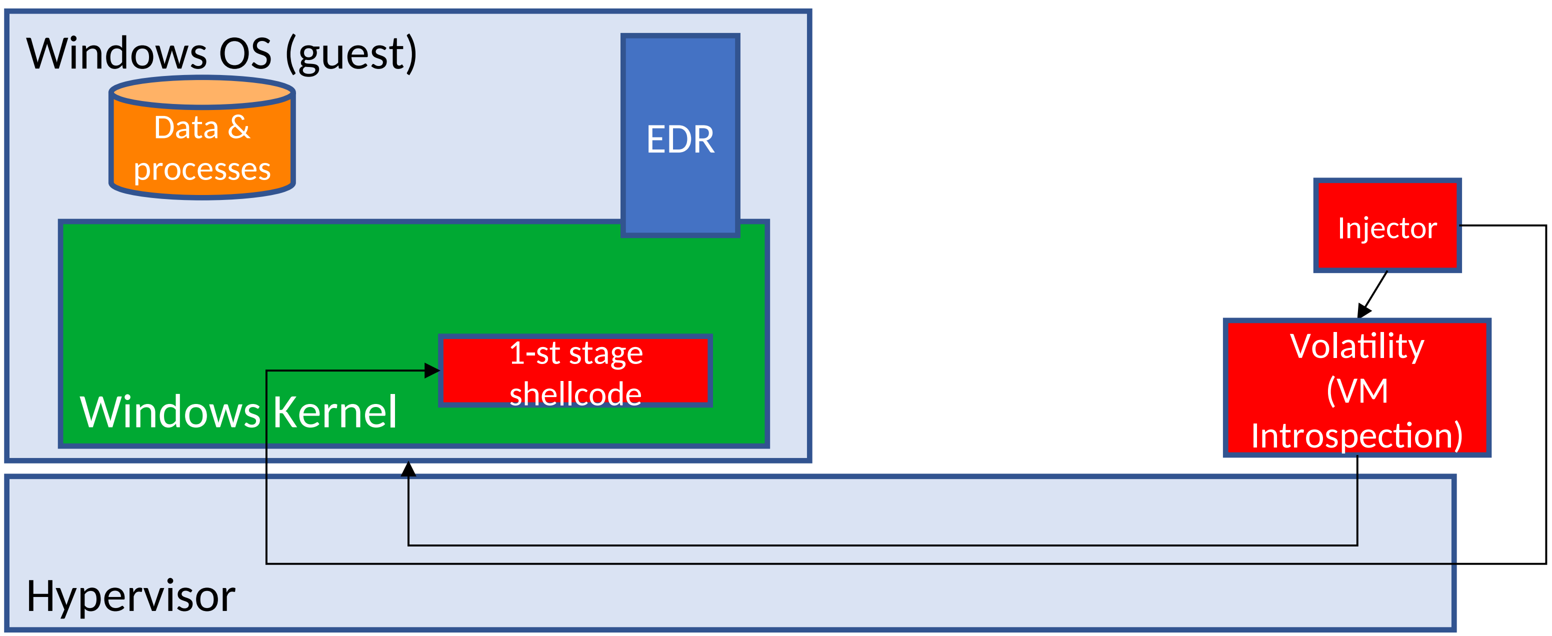
Novel approach (2/4)

1) **Hypothesis: injection from hypervisor (it's a real threat [3]);**
Arbitrary read/write access to VM's RAM, VM Introspection techniques.



Novel approach (3/4)

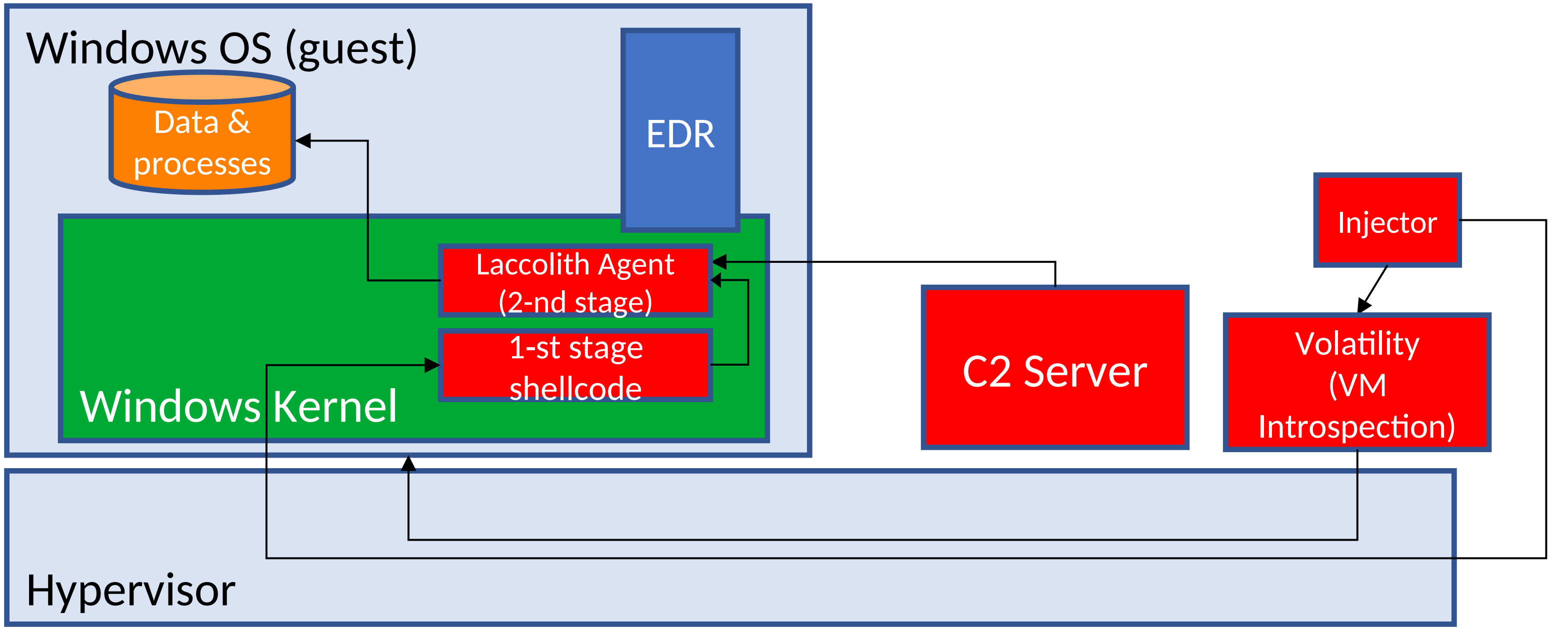
2) Use the hypervisor's privilege to gain arbitrary code execution in a virtual domain;
Overwriting the code of a system call, "new" code with strict requirements.



Novel approach (4/4)

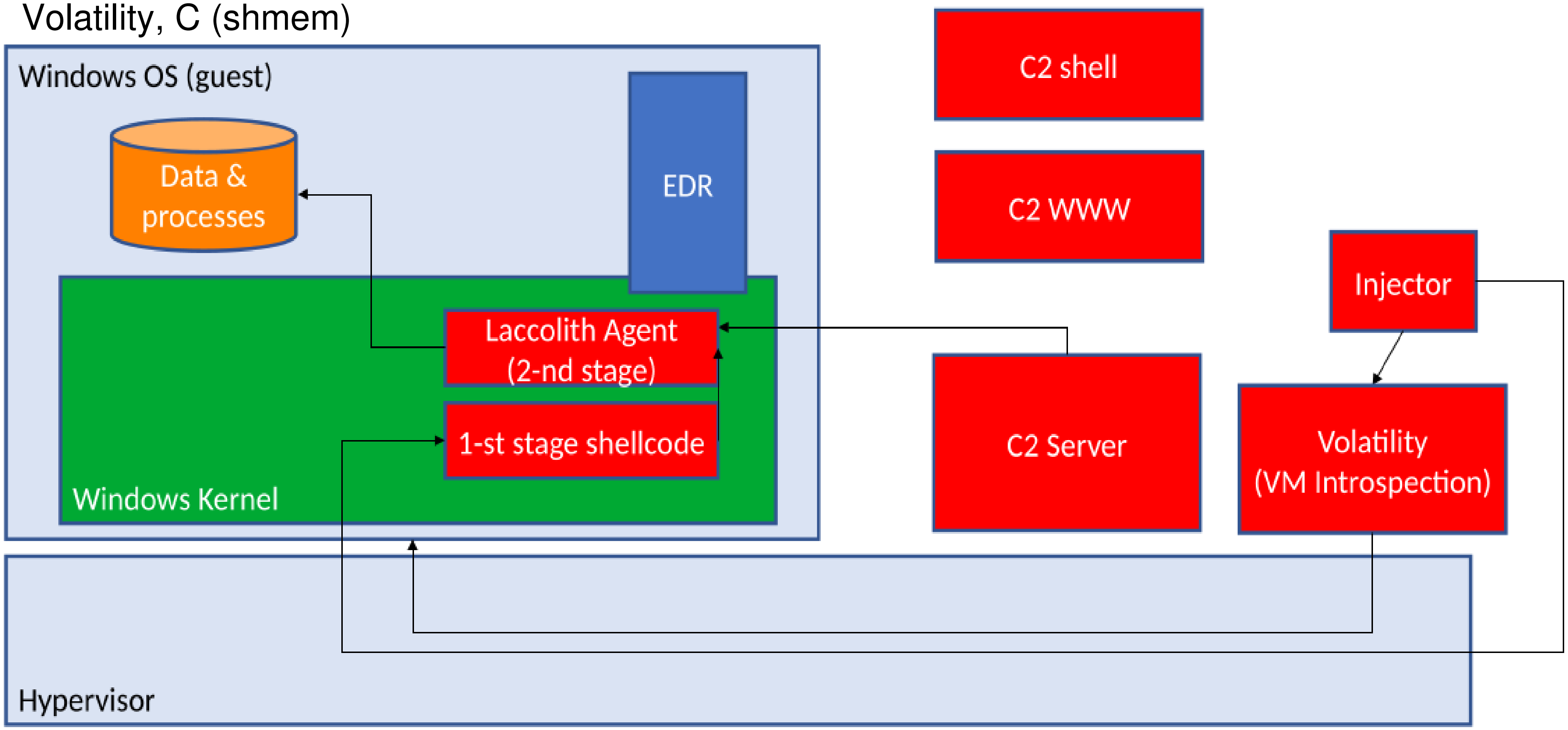
3) Use the code execution "primitive" to inject an *agent* capable of high-level actions, after connecting to the C2 server.

The agent is a kernel shellcode developed in a modular way with Rust [4].



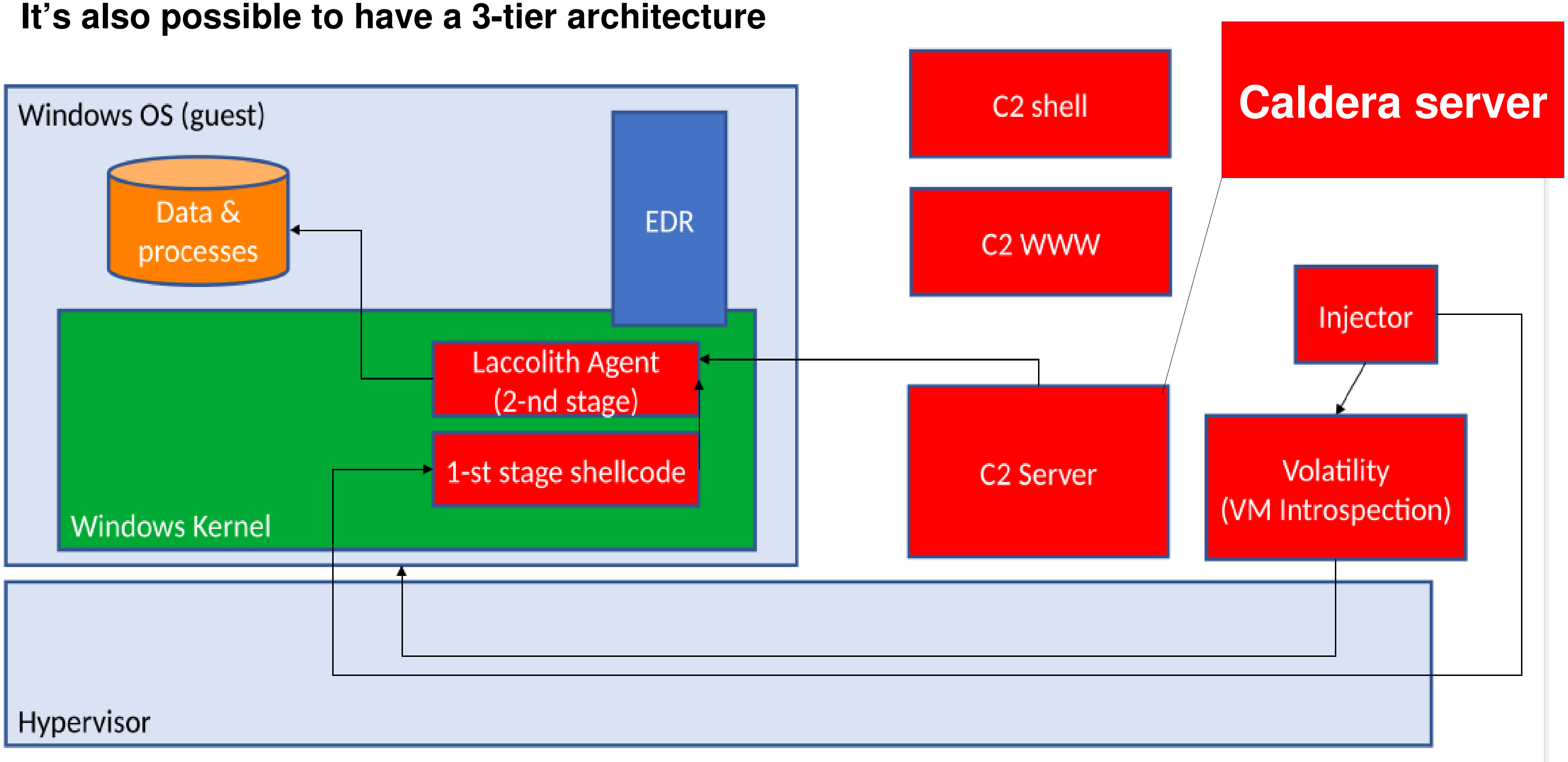
Architecture of the implemented prototype (1/2)

Technologies used: Python, Rust, Docker, Volatility, C (shmem)



Architecture of the implemented prototype (2/2)

It's also possible to have a 3-tier architecture



Experiments (1/7)

Objectives

- **Compare anti-detection capabilities** of Caldera's agent with those of the proposed approach
- **Study the reliability** of the proposed approach

Antivirus configurations

- **Windows Defender**
- **Avast**
- **AVG**
- **Kaspersky (Security Cloud)**
- **Avira**

Metrics

- **Adversary profile execution progress**
- **Empirical probability of success**

Adversary profiles

- **Basic:** Discovery, Collection, Exfiltration
- **Advanced:** Credential Access, Privilege escalation, Persistence, Impact

Experiments (2/7)

Adversary profile execution progress for Caldera's profiles

Caldera Profile	Windows Defender	Avast	AVG	Kaspersky	Avira
Discovery	9 / 9	9 / 9	9 / 9	9 / 9	9 / 9
Hunter	14 / 14	14 / 14	14 / 14	14 / 14	14 / 14
Check	6 / 6	6 / 6	6 / 6	6 / 6	6 / 6
Collection	2 / 2	2 / 2	2 / 2	2 / 2	2 / 2
Enumerator	5 / 5	5 / 5	5 / 5	5 / 5	5 / 5
Nosy Neighbor	7 / 7	7 / 7	7 / 7	7 / 7	7 / 7
Signed Binary Proxy Execution	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3
Super Spy	11 / 11	11 / 11	11 / 11	11 / 11	11 / 11
Undercover	**1 / 2**	**1 / 2**	**1 / 2**	**1 / 2**	2 / 2
Stowaway	**1 / 2**	**1 / 2**	**1 / 2**	**1 / 2**	2 / 2
Worm	**1 / 9**	**1 / 9**	**1 / 9**	**1 / 9**	9 / 9
You Shall (Not) Bypass	**2 / 4**	**2 / 4**	**2 / 4**	**1 / 4**	**1 / 4**
Ransomware	5 / 5	5 / 5	5 / 5	5 / 5	5 / 5

The first 8 profiles are **basic**, the last 5 are **advanced**.

Profiles with **Credential Access** and/or **Privilege Escalation** tactics get detected.

Furthermore, the agent needs *pre-injection*.

Experiments (3/7)

Detected abilities for Caldera's profiles

Caldera Profile	Detected Ability
Undercover	Install PowerShell Core 6
Stowaway	Inject Sandcat into Process
Worm	Run PowerKatz
You Shall (Not) Bypass	Wow64log DLL Hijack
You Shall (Not) Bypass	Bypass UAC Medium

Experiments (4/7)

Caldera's agent (Sandcat) needed *pre-injection* because the deployment of the agent itself got detected.

Other experiments were about trying to integrate Caldera with **anti-detection tools**, like **Inceptor** [5].

Problems:

- Reliability is still low, even with the most hardened set of the available anti-detection techniques;
- UAC bypass is still needed.

Experiments (5/7)

Profiles implemented using Laccolith's agent

Profile	Description	Tactics	Referenced APTs
Thief	Exfiltrate files from local user desktop	Discovery, Collection, Exfiltration	APT1, OilRig, APT3
Op-2	Upload a powershell script in a system folder and install a scheduled task that executes that script at boot, get system version and dump memory of LSASS process	Persistence, Credential access	Remsec (Strider), Ke3chang
Ransomware	Discover and exfiltrates sensitive files, encrypt them and leave a message	Discovery, Collection, Exfiltration, Impact	APT3, Bad Rabbit (multiple APTs)

Experiments (6/7)

Adversary profile execution progress for Laccolith's profiles

Profile	Windows Defender	Avast	AVG	Kaspersky	Avira
Thief	3 / 3	3 / 3	3 / 3	**0 / 3 ***	3 / 3
Op-2	4 / 4	4 / 4	4 / 4	**0 / 4 ***	4 / 4
Ransomware	5 / 5	5 / 5	5 / 5	**0 / 5 ***	5 / 5

* The profiles are not actually detected by Kaspersky; it's the injection procedure itself that in this configuration makes the system crash before making possible to perform useful actions.

Experiments (7/7)

Reliability of the injection method: 20 independent experiments for each configuration, each one with an *echo* and a *close*, counting how many times the injection AND the echo AND the graceful termination were successful.

Windows Defender	Avast	AVG	Kaspersky	Avira	Overall without Kaspersky (percentage)
15 / 20	16 / 20	17 / 20	0 / 20	16 / 20	64 / 80 (80%)

With Kaspersky, the probability of success has proven to be very low.
The other configurations can be considered belonging to the same "class", so their experiments can be aggregated. If *N* is the number of repetitions, the margin of error can be approximated [6] as 1/sqrt(N), that is ~0.112.
Therefore, probability of success is in the range [68.8%, 91.2%].

Conclusions

- Evidenced the **lack of defense evasion capabilities** of one of the most popular frameworks for adversary emulation.
- Issue addressed by **designing and implementing a novel approach.**
- **Experiments** showed the **anti-detection capabilities** and the **increased reliability** of the novel approach [7].

Future developments (unsorted):

- Increase the reliability of the injection method;
- Engineer the whole framework & manage portability;
- Implement kernel payloads and injection chain for other operating systems;
- Configurable defense evasion capabilities for simulations, by implementing a command to execute user-mode shellcodes;
- Test the attack in presence of technologies like *Intel TDX*.

References

[1]: Plextrac, https://plextrac.com/what-is-adversary-emulation-adversary-simulation/, 10/10/2022
[2]: Mitre, https://caldera.mitre.org/, 10/10/2022
[3]: Google Project Zero's Blog, https://googleprojectzero.blogspot.com/2017/04/pandavirtualization-exploiting-xen.html, 10/10/2022
[4]: zerosum0x0's Blog, https://zerosum0x0.blogspot.com/2020/08/sassykitdi-kernel-mode-tcp-sockets.html, 10/10/2022
[5]: Inceptor's GitHub repository, https://github.com/klezVirus/inceptor, 10/10/2022
[6]: Brian Caffo, Statistical inference for data science, Leanpub, 2016, 57-60
[7]: Project's GitHub repository, https://github.com/Shotokhan/adversary-emulation, 10/10/2022