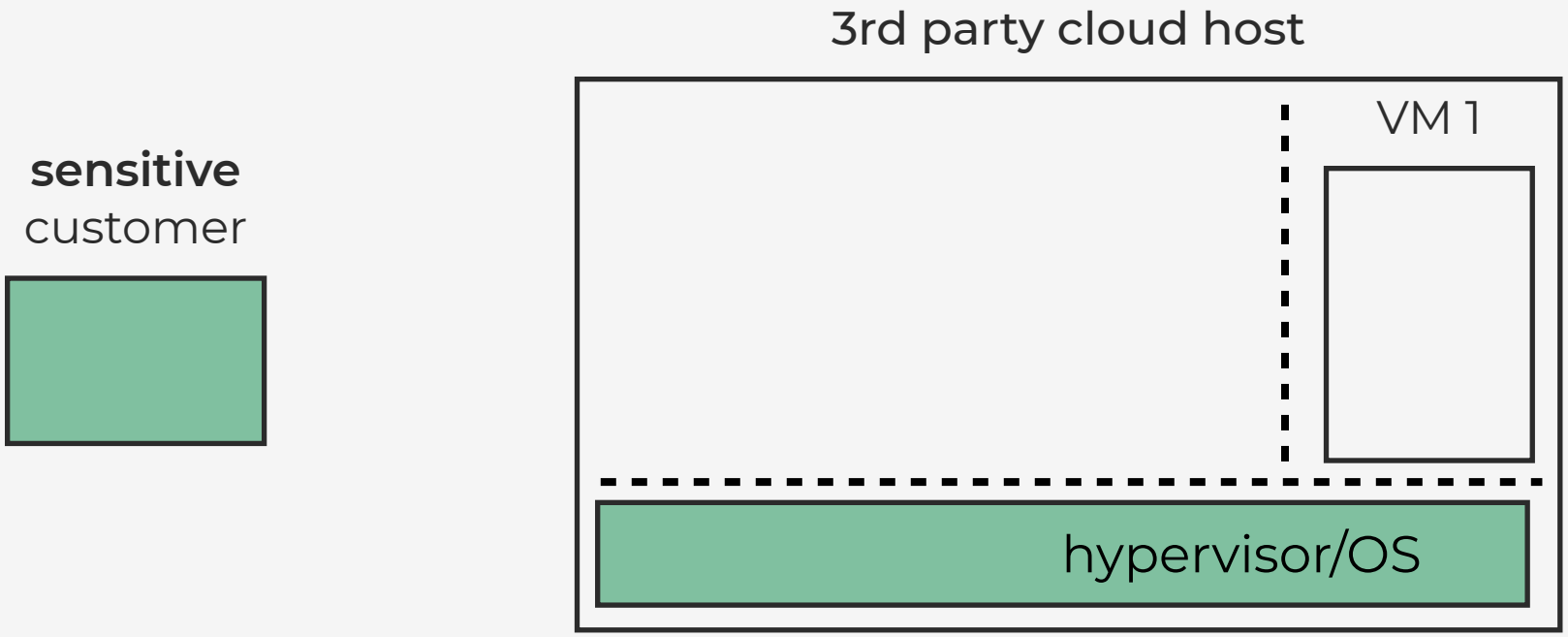
CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

00SEVen — Re-enabling Virtual Machine Forensics

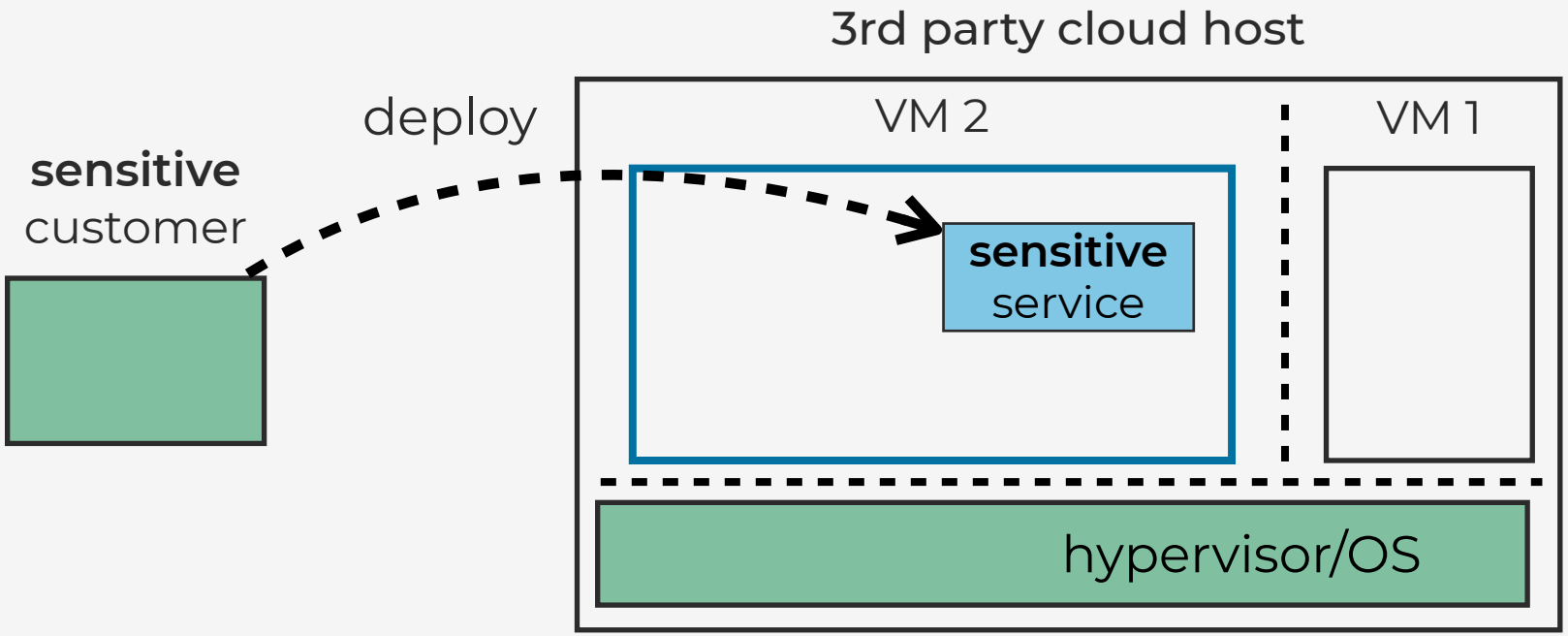
Introspecting Confidential VMs Using Privileged in-VM Agents

Fabian Schwarz and Christian Rossow | USENIX Security 2024
CISPA Helmholtz Center for Information Security

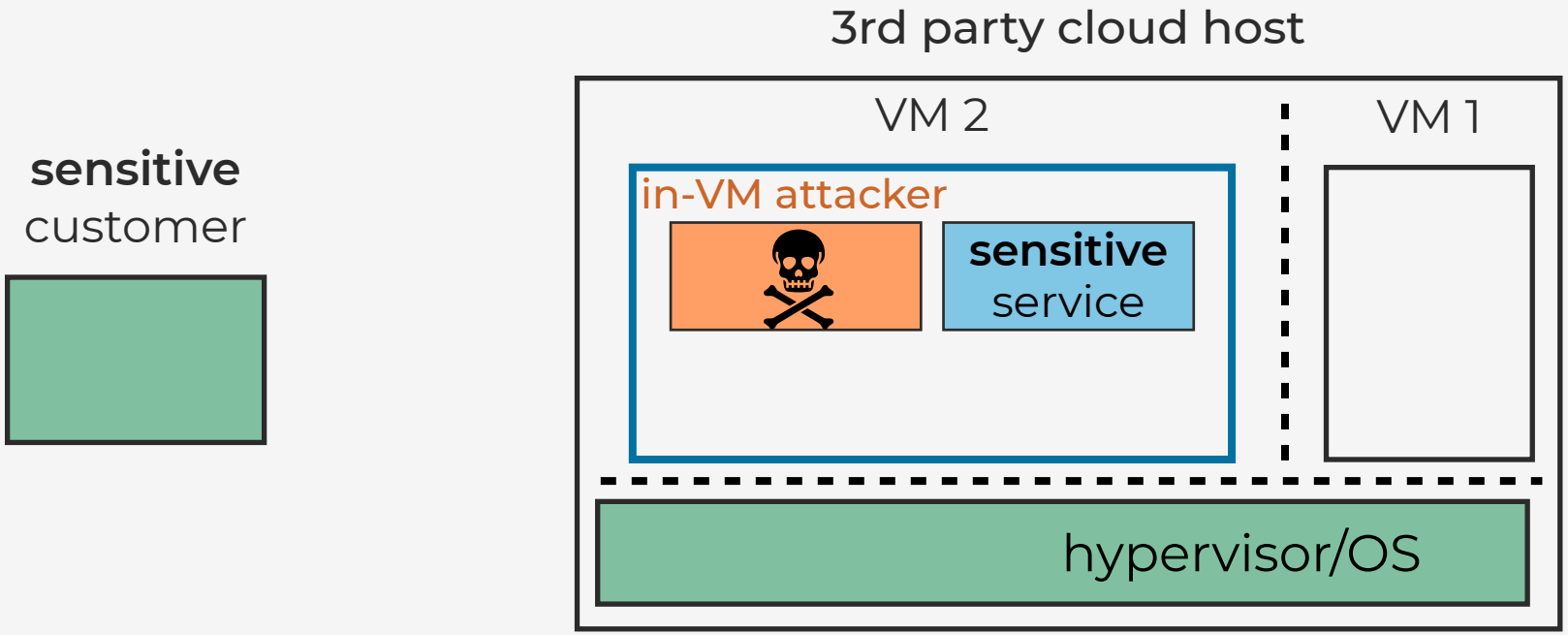
Motivation: Securely Offload and Inspect Cloud VMs



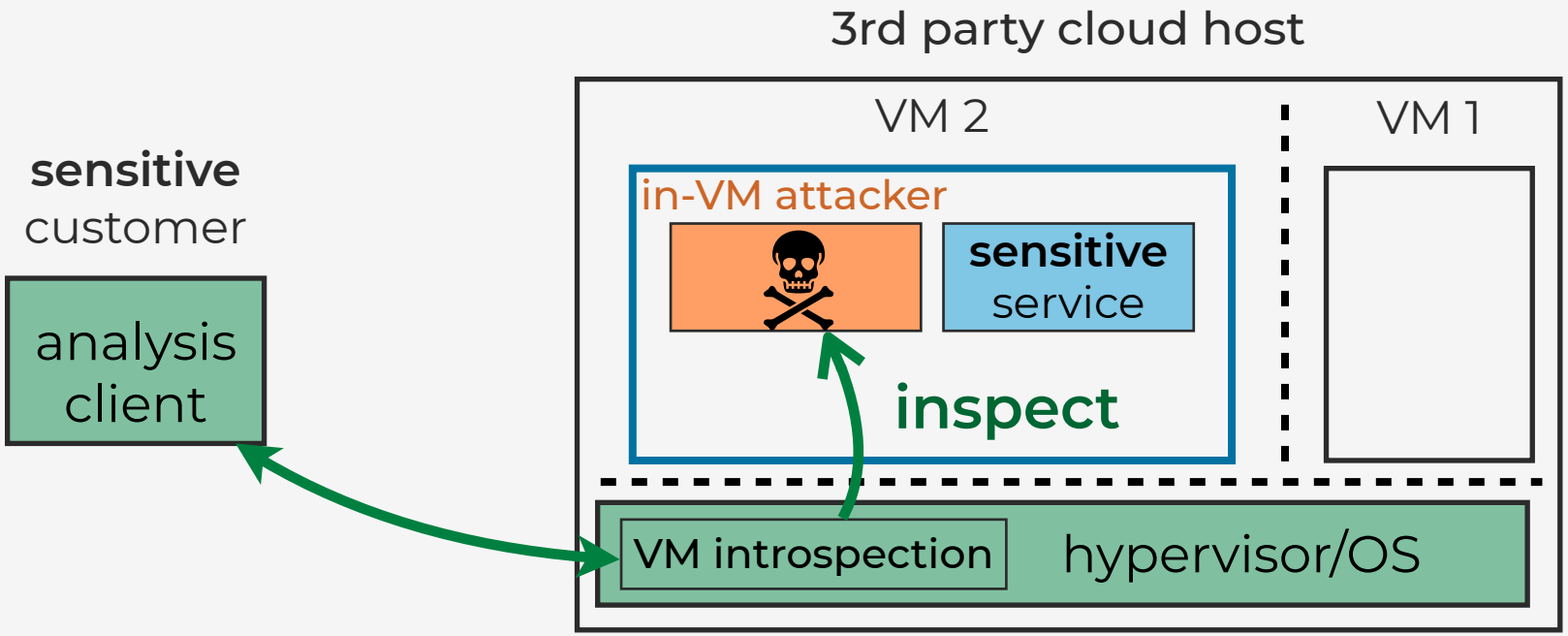
Motivation: Securely Offload and Inspect Cloud VMs



Motivation: Securely Offload and Inspect Cloud VMs

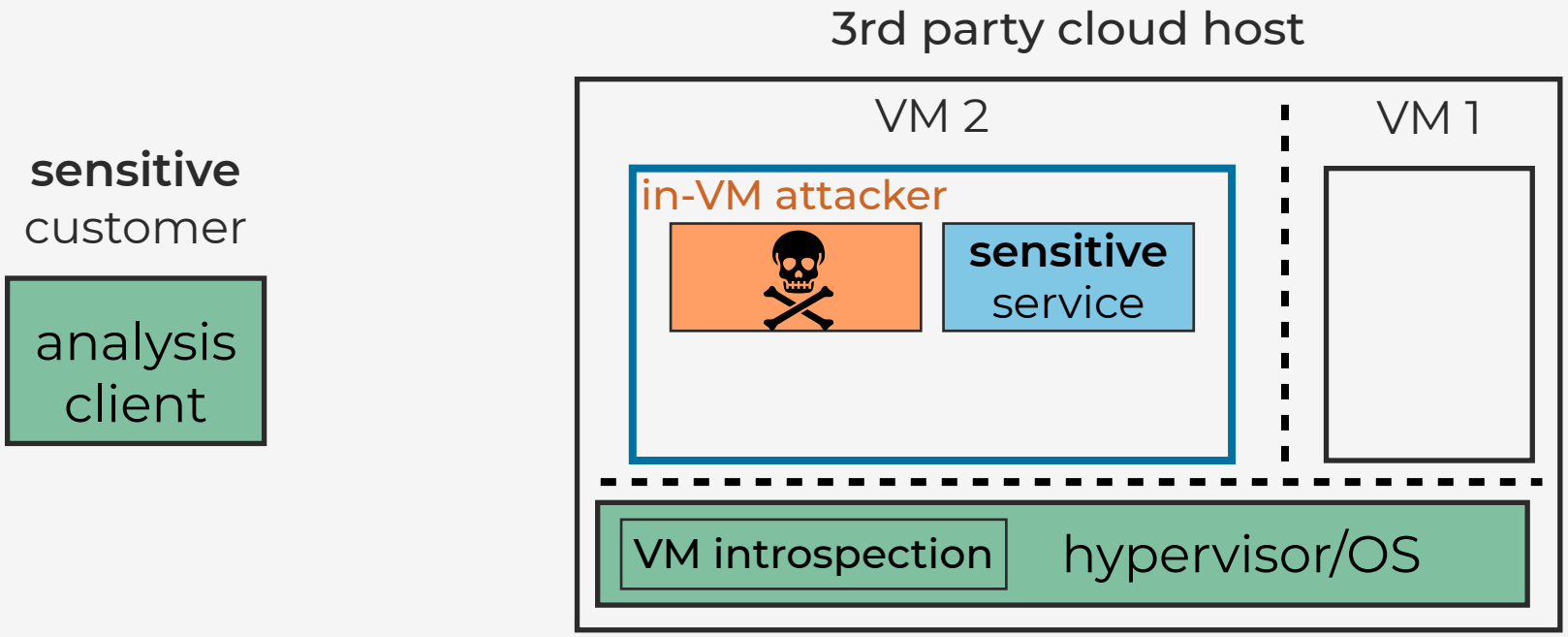


Motivation: Securely Offload and Inspect Cloud VMs

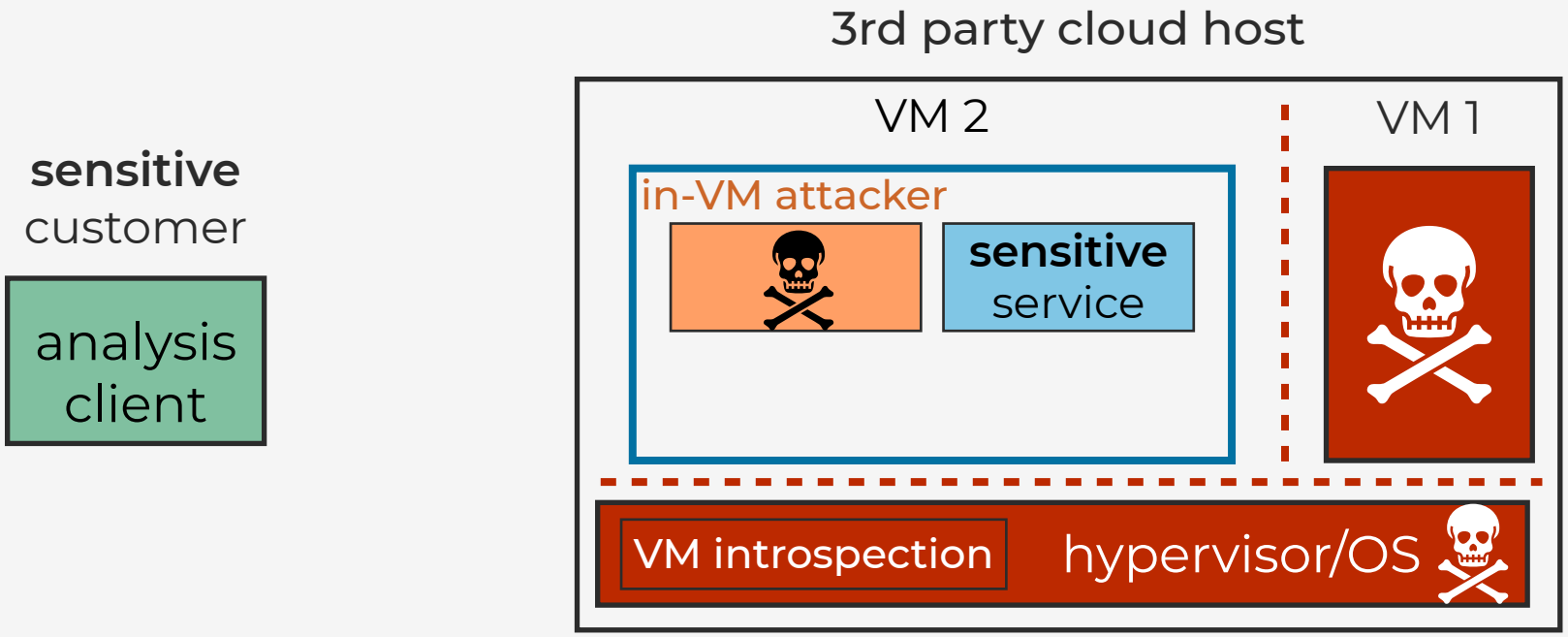


- **VM introspection (VMI)** enables secure monitoring of compromised VMs for in-VM attackers (malware, rootkits)
  - inspect memory + registers
  - pause VM on demand
  - trap VM page access

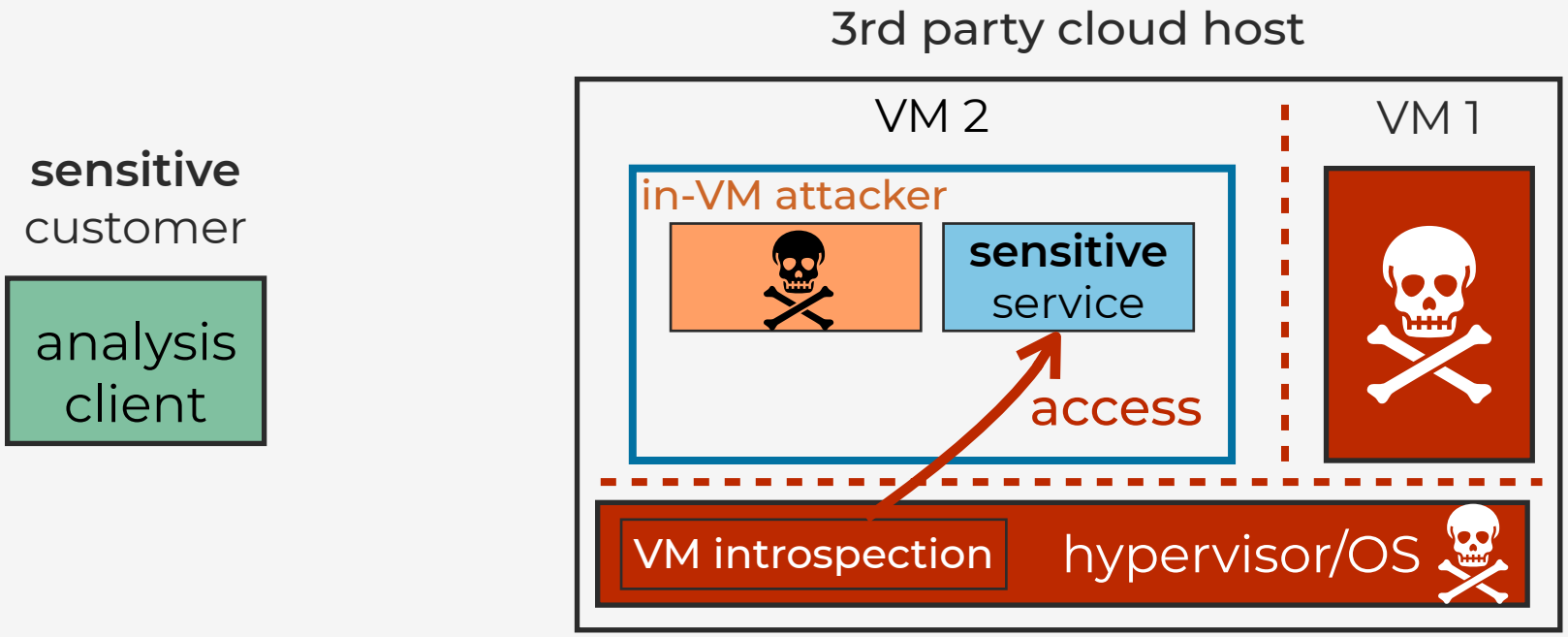
Motivation: Securely Offload and Inspect Cloud VMs



Solved Issue: VMI Exploitation to Attack Sensitive VMs



Solved Issue: VMI Exploitation to Attack Sensitive VMs

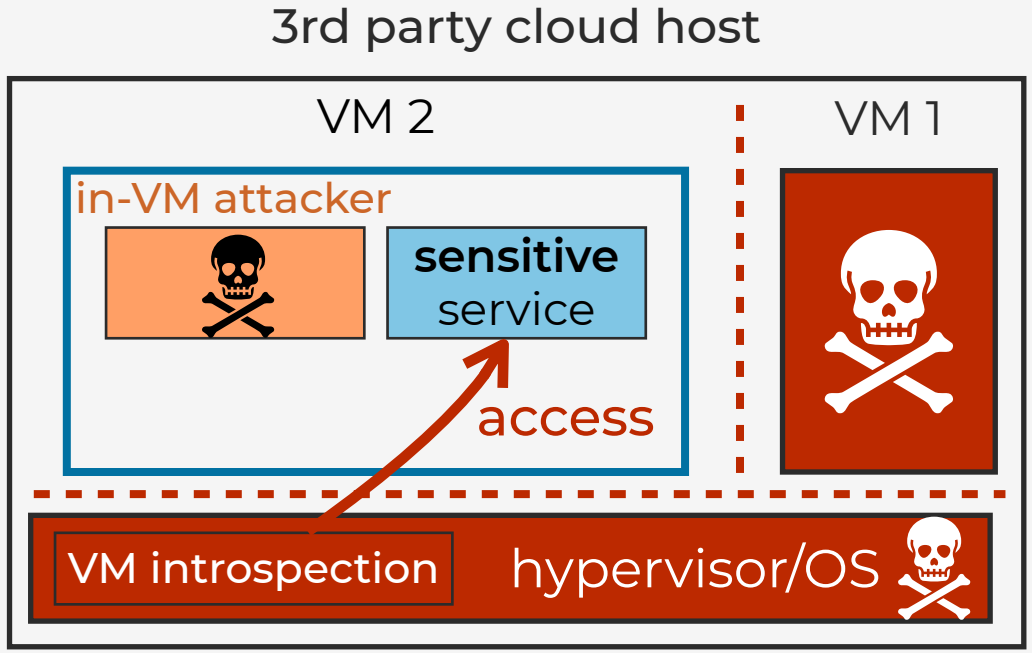


- **ISSUE:** If host is compromised, or inside attackers are present, can abuse VMI to steal or manipulate customers' sensitive services
- host can fully compromise the customer VM

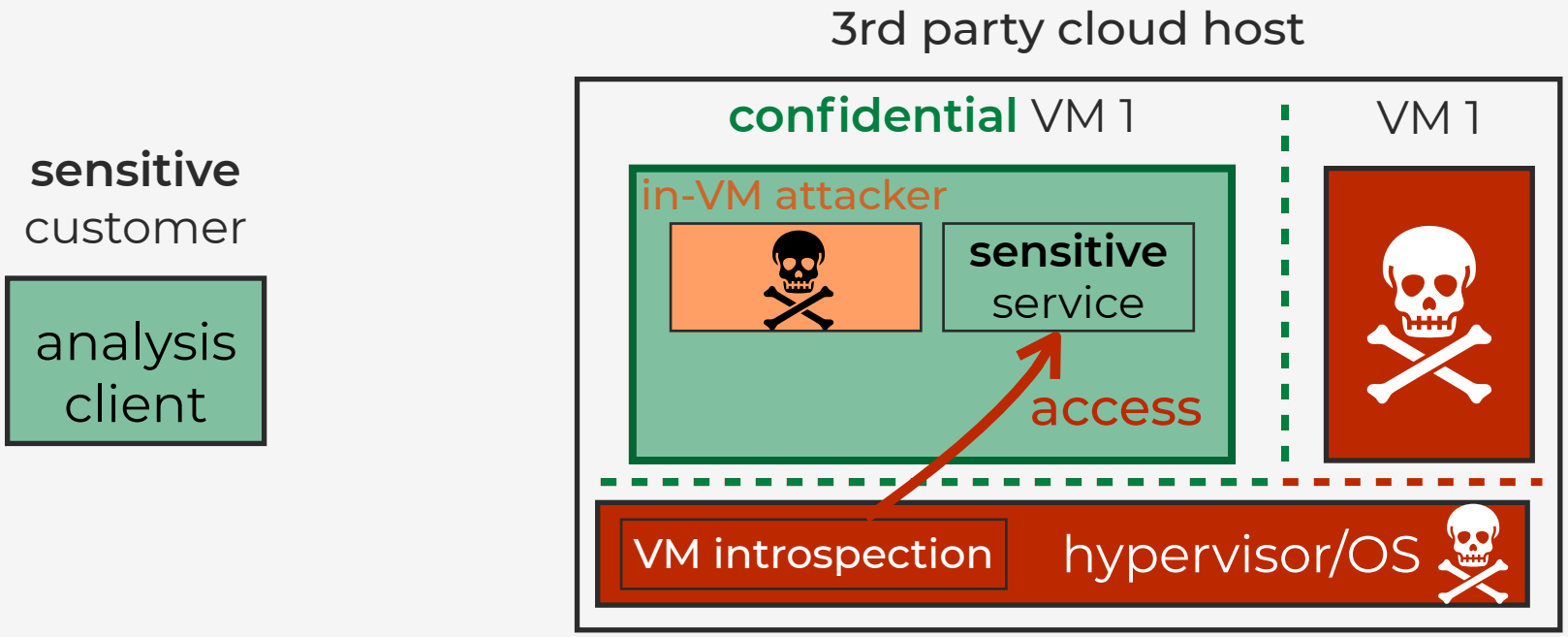
Solved Issue: VMI Exploitation to Attack Sensitive VMs

sensitive
customer

analysis client

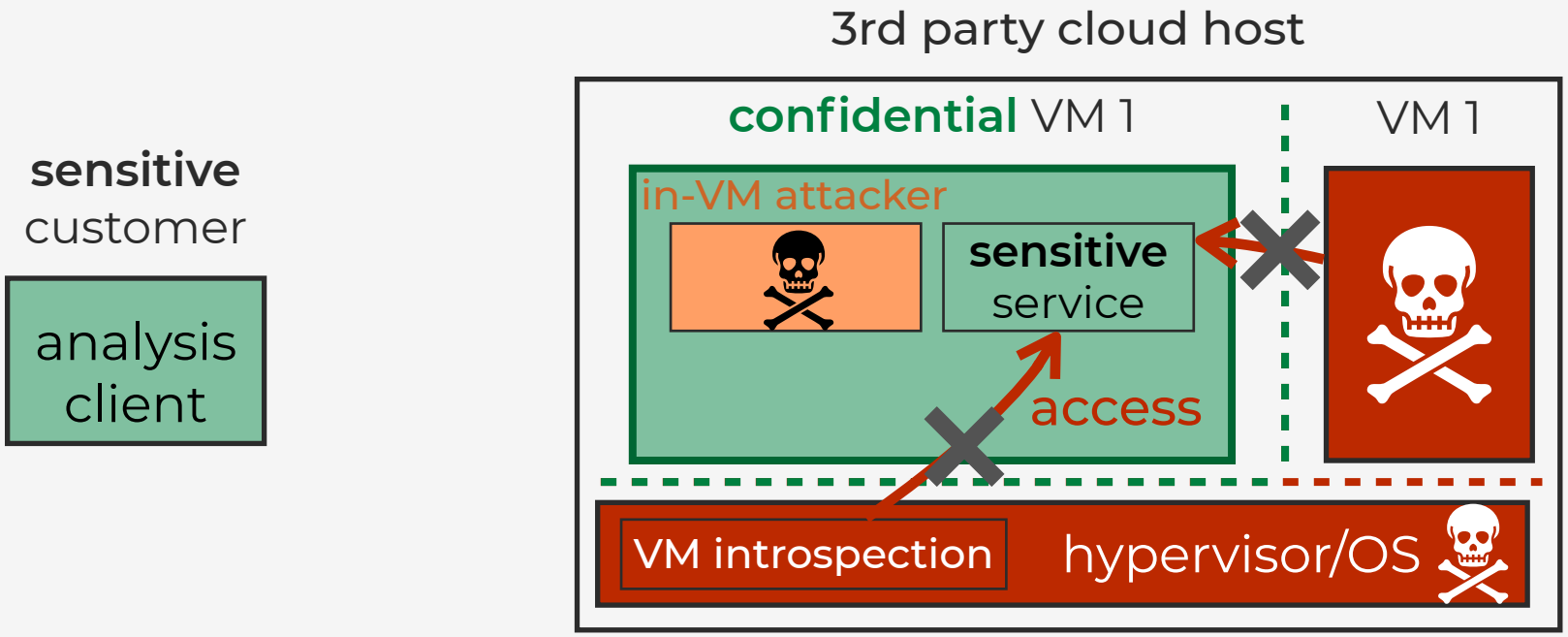


Confidential VMs and their Incompatibility with VMI



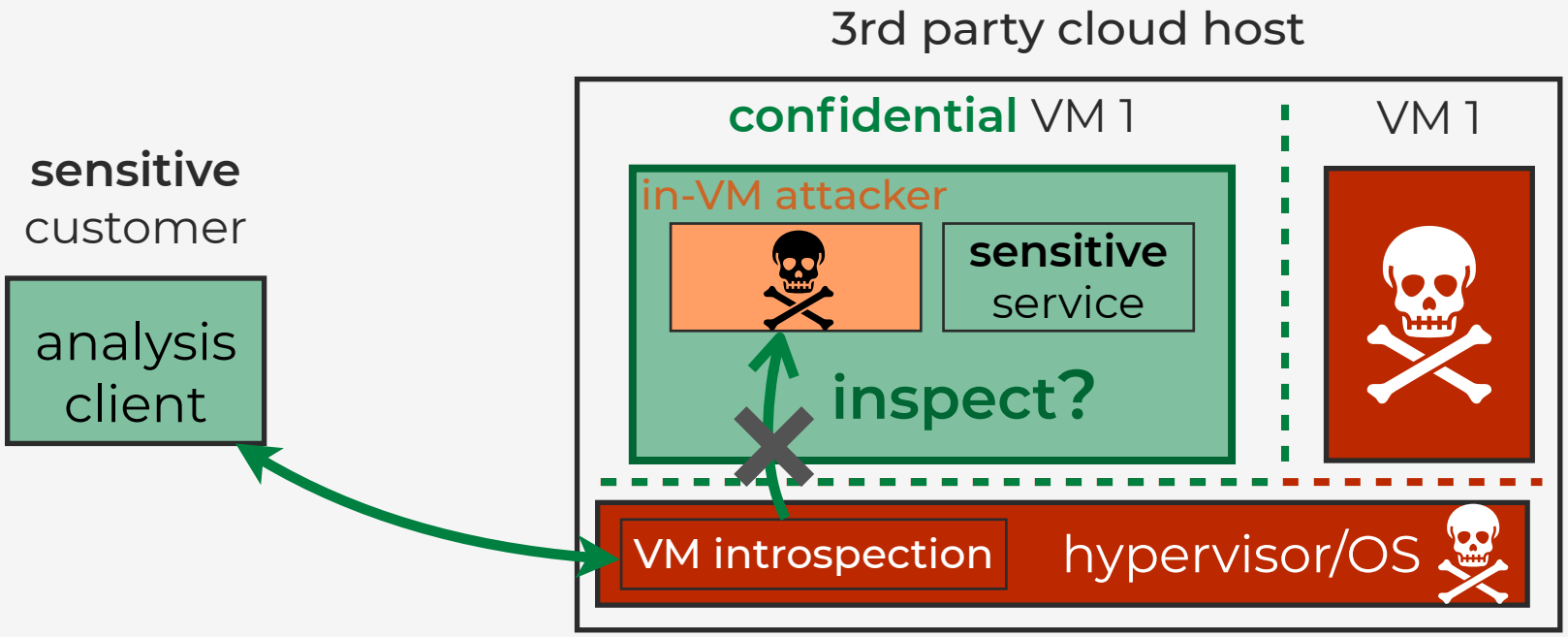
- **Confidential VMs** (cVMs, also: TEE VMs / TVMs) de-trust host and other VMs

Confidential VMs and their Incompatibility with VMI



- **Confidential VMs** (cVMs, also: TEE VMs / TVMs) de-trust host and other VMs
  - deny access by **host or other VMs** to cVMs' memory/registers
  - e.g.: **AMD SEV-SNP** *(our focus)*, Intel TDX, Arm CCA

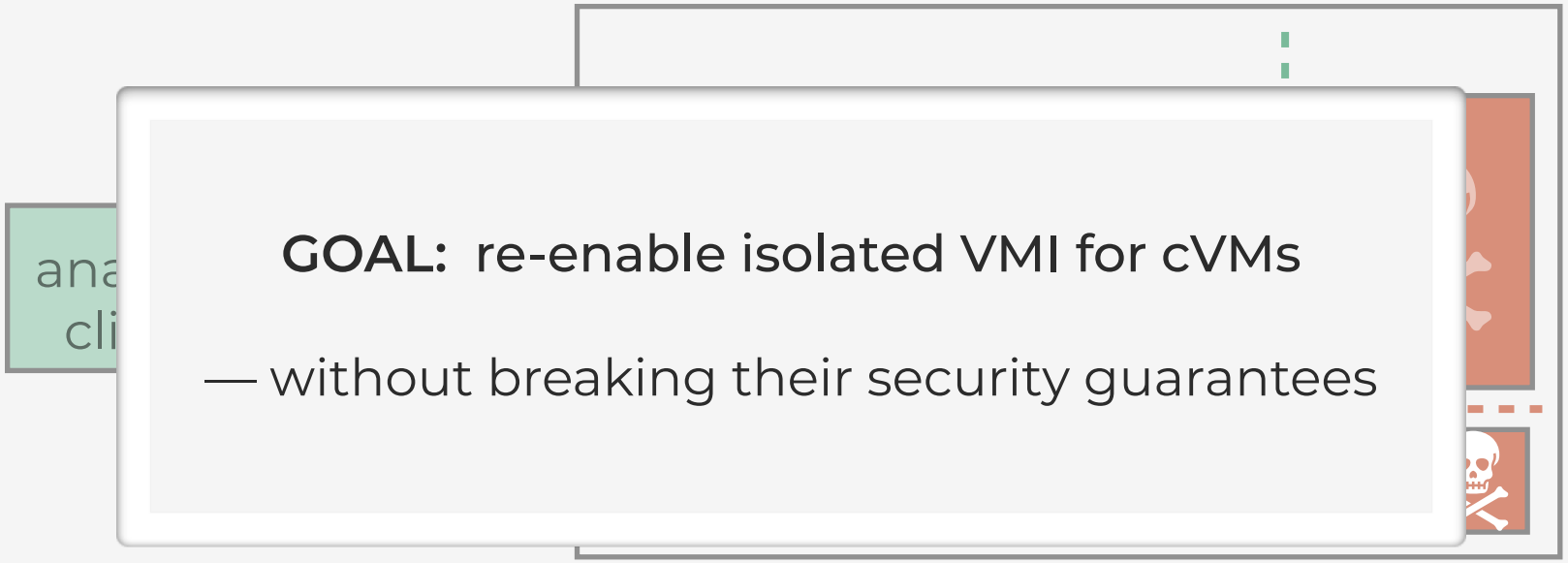
Confidential VMs and their Incompatibility with VMI



- **Confidential VMs** (cVMs, also: TEE VMs / TVMs) de-trust host and other VMs
  - deny access by **host or other VMs** to cVMs' memory/registers
  - e.g.: **AMD SEV-SNP** *(our focus)*, Intel TDX, Arm CCA

- **DOWNSIDE:** cVM's memory protection blocks VMI of attackers inside cVM

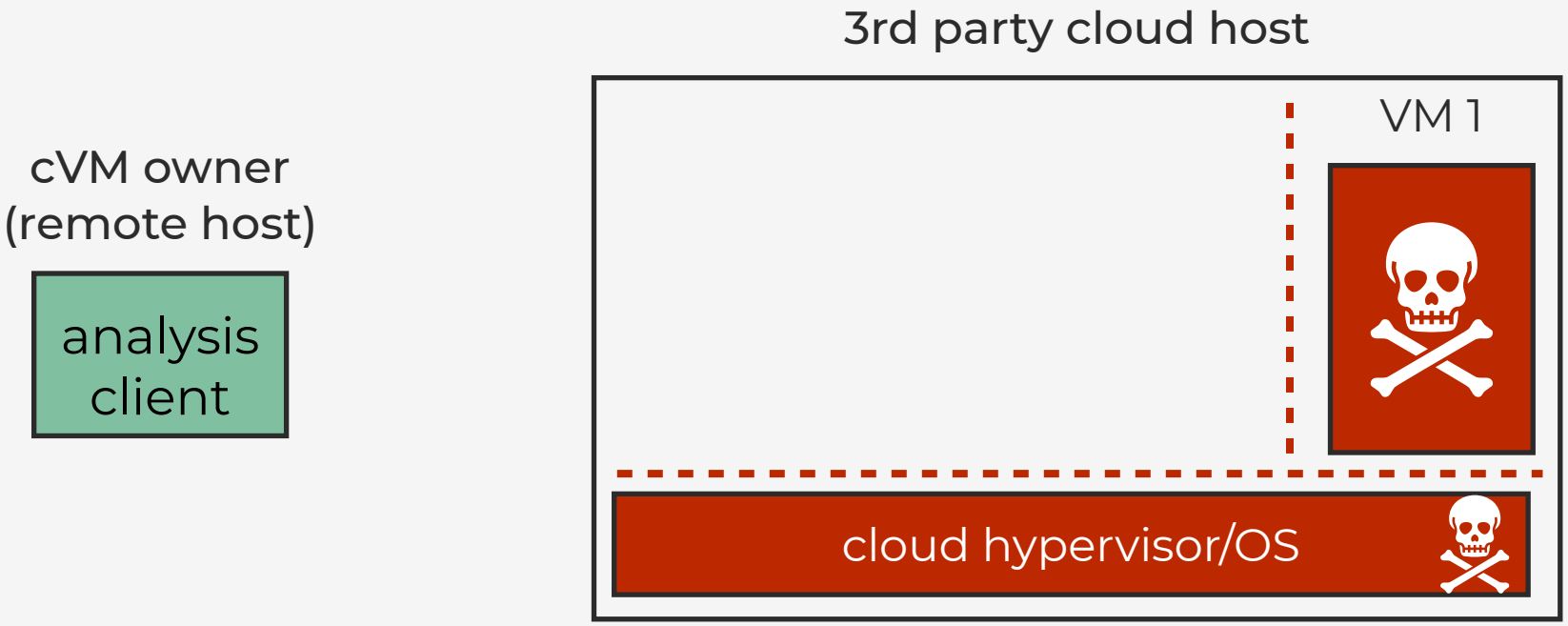
Confidential VMs and their Incompatibility with VMI



- **Confidential VMs** (cVMs, also: TEE VMs / TVMs) de-trust host and other VMs
  - deny access by **host or other VMs** to cVMs' memory/registers
  - e.g.: **AMD SEV-SNP** *(our focus)*, Intel TDX, Arm CCA

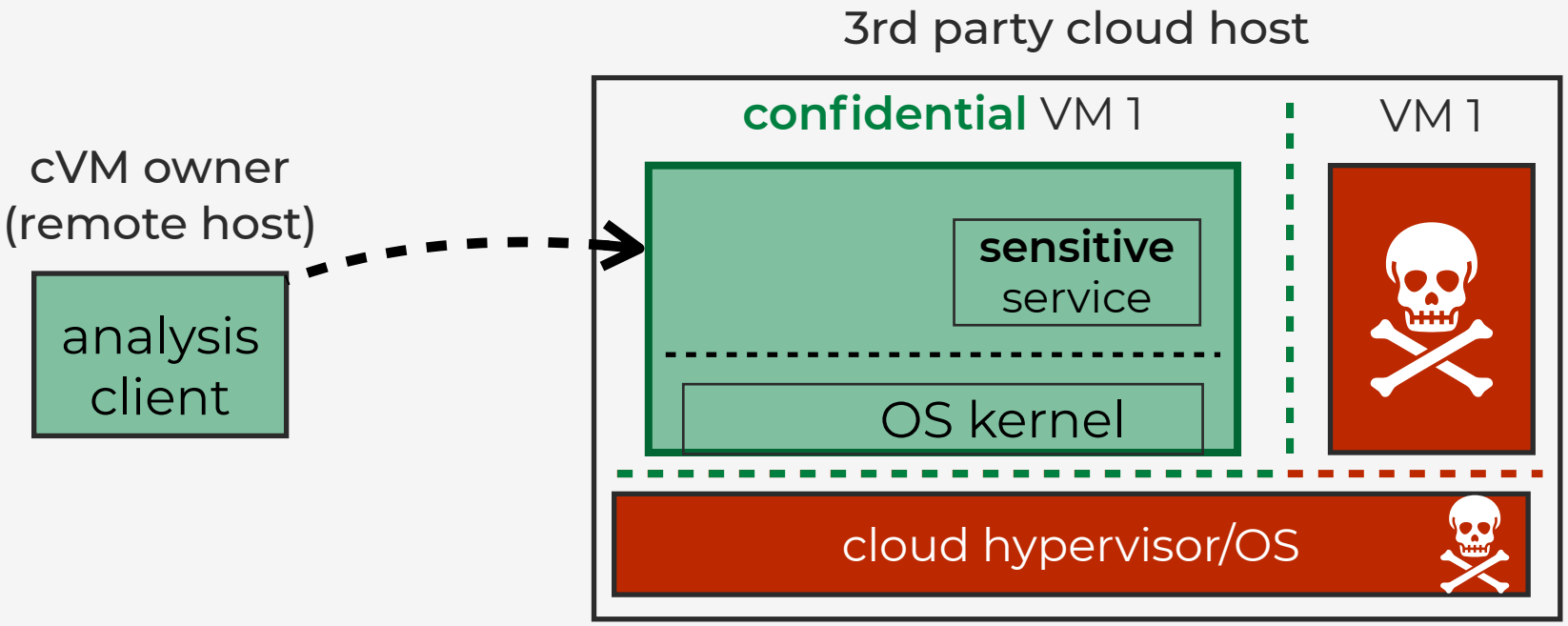
- **DOWNSIDE:** cVM's memory protection blocks VMI of attackers inside cVM

Threat Model for Introspection of cVMs



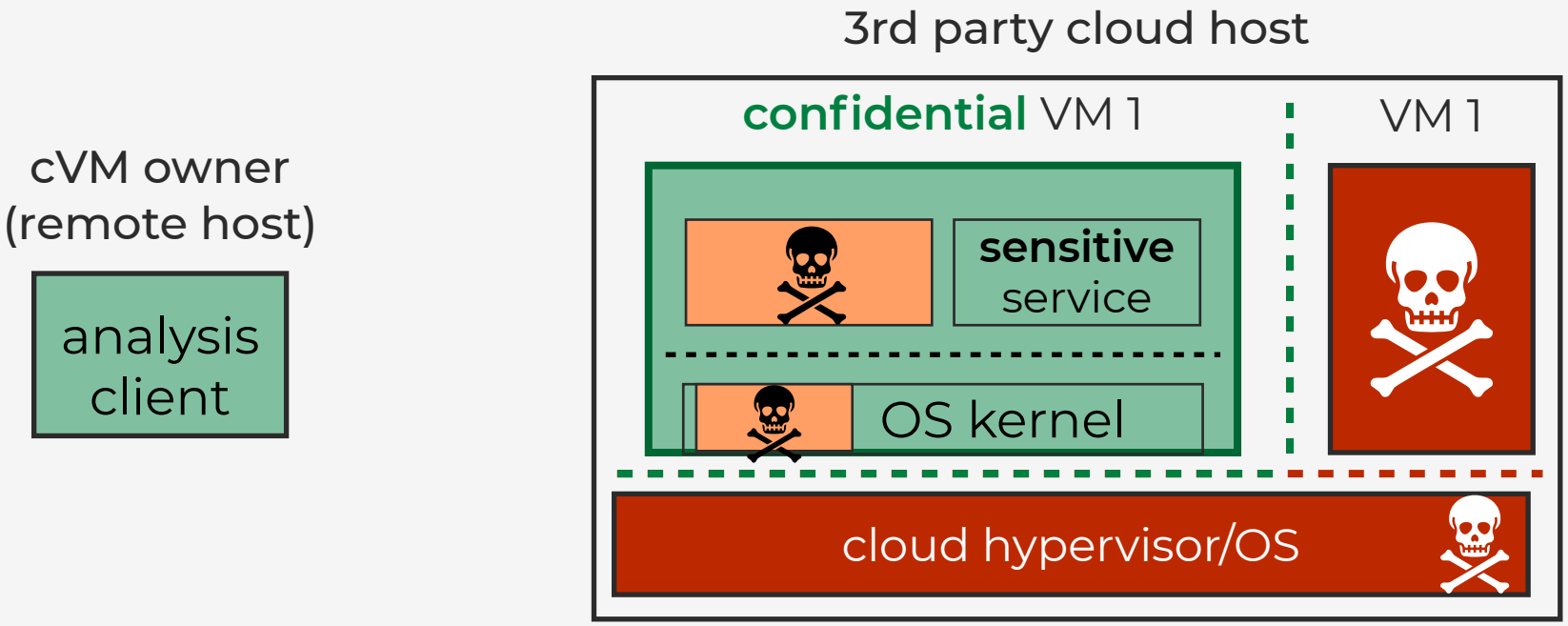
- **out-of-VM** attacker: cloud host software + other VMs

Threat Model for Introspection of cVMs



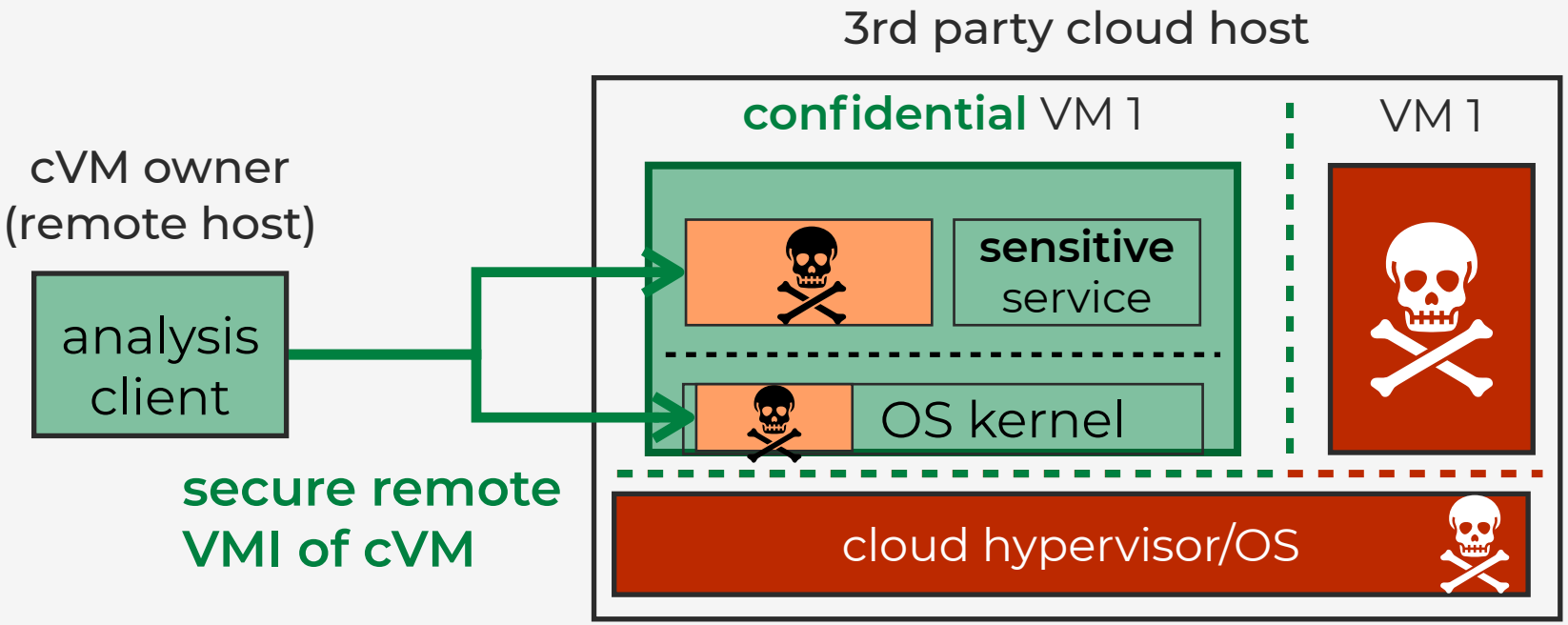
- **out-of-VM** attacker: cloud host software + other VMs
- trusted client deploys sensitive IP services in confidential VM (at cloud)

Threat Model for Introspection of cVMs



- **out-of-VM** attacker: cloud host software + other VMs
- trusted client deploys sensitive IP services in confidential VM (at cloud)
- **in-VM** attacker: malware, kernel rootkits (≠ out-of-VM attacker)

Threat Model for Introspection of cVMs

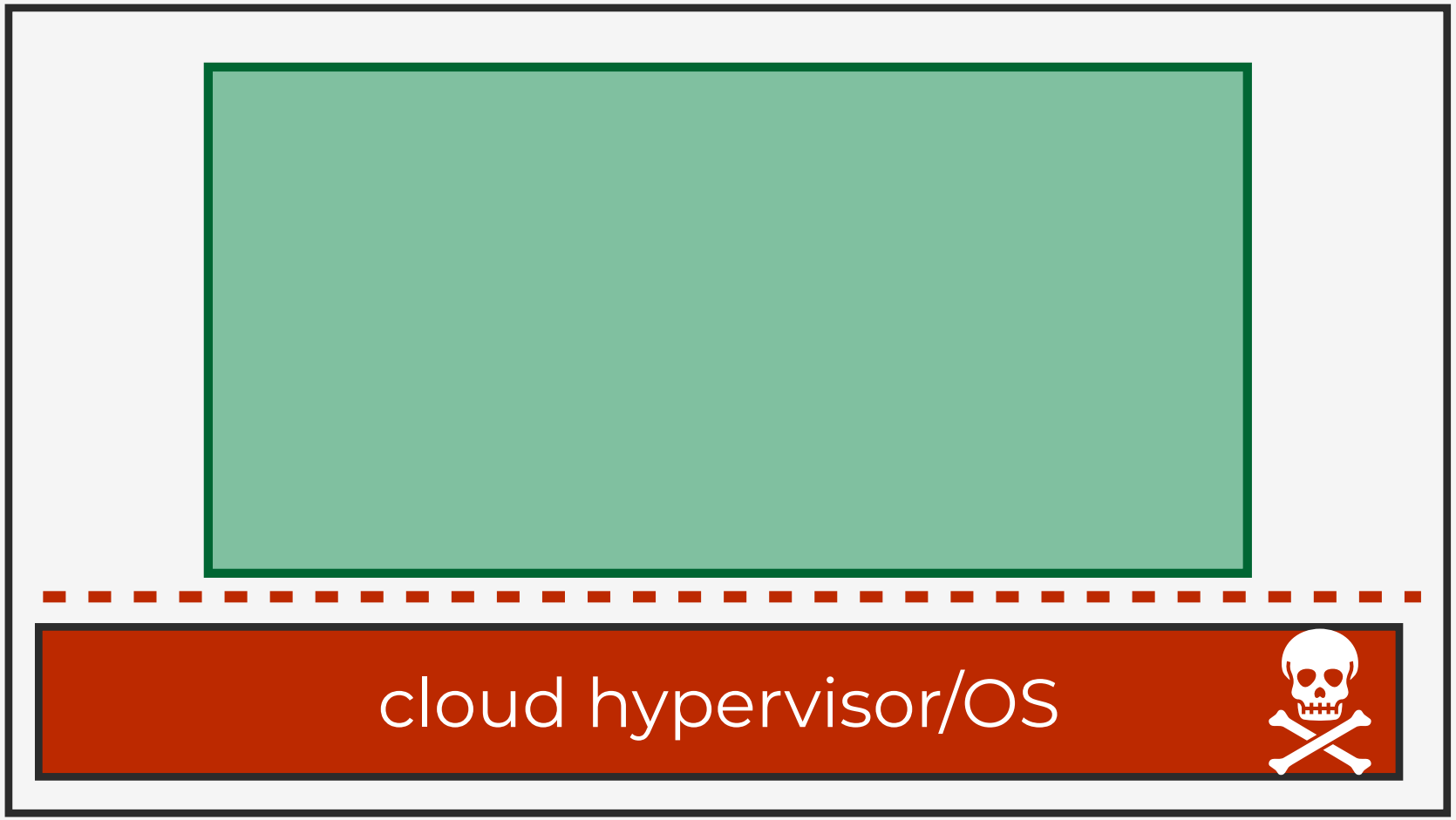


- **out-of-VM** attacker: cloud host software + other VMs
- trusted client deploys sensitive IP services in confidential VM (at cloud)
- **in-VM** attacker: malware, kernel rootkits (≠ out-of-VM attacker)
- **GOAL: trusted client wants to perform secure remote introspection (VMI) to monitor for in-VM attacks**

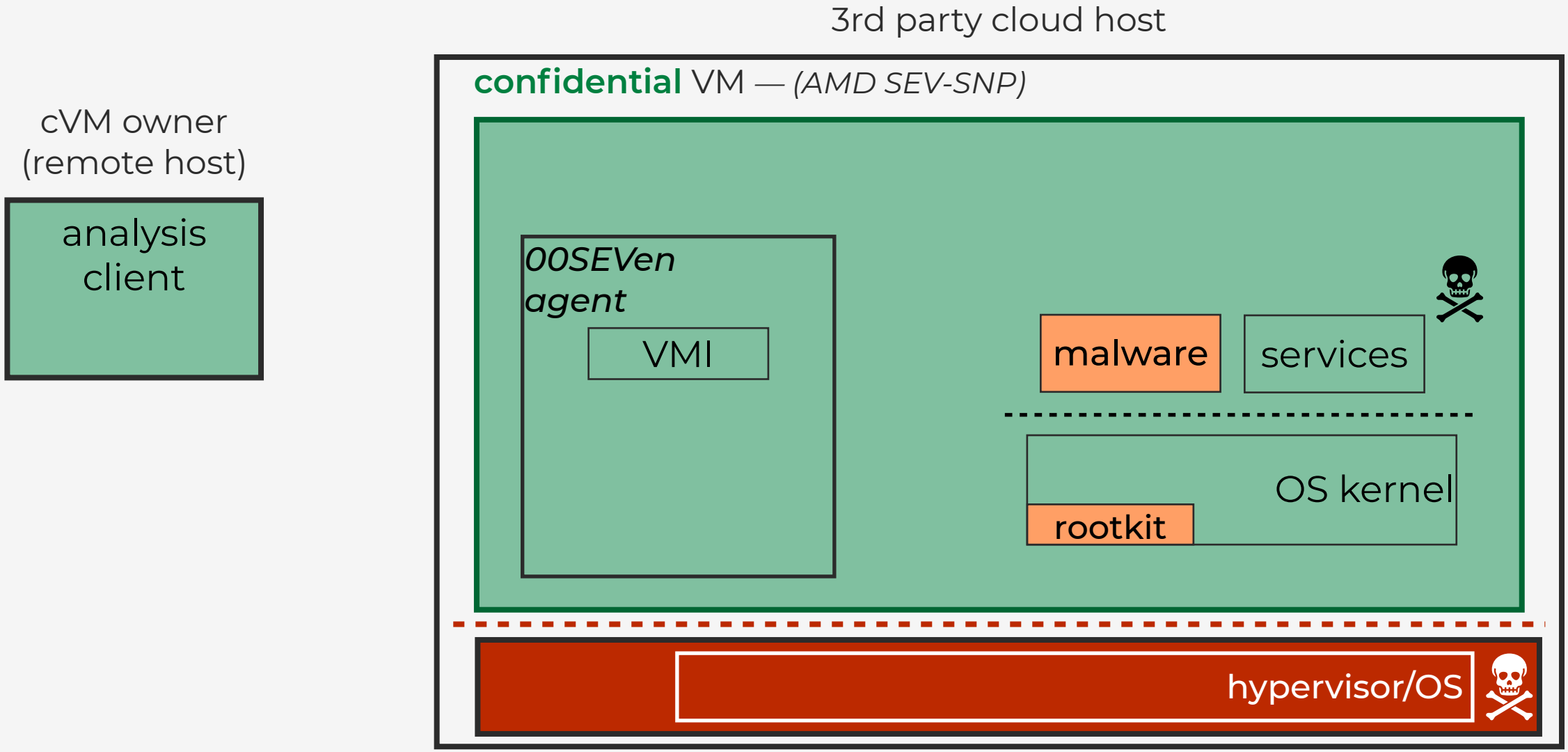
00SEVen Design Overview and Challenges

cVM owner
(remote host)

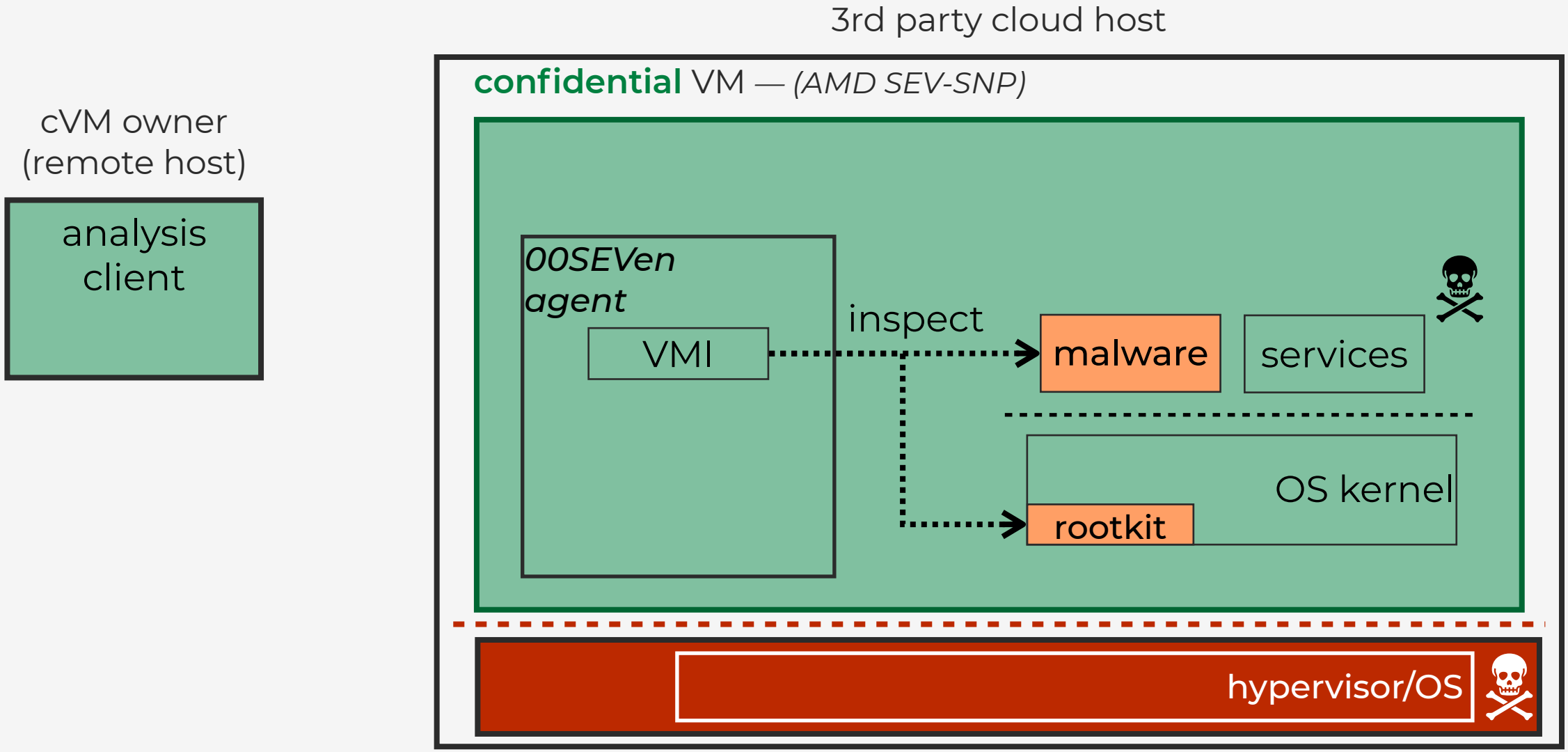
analysis client



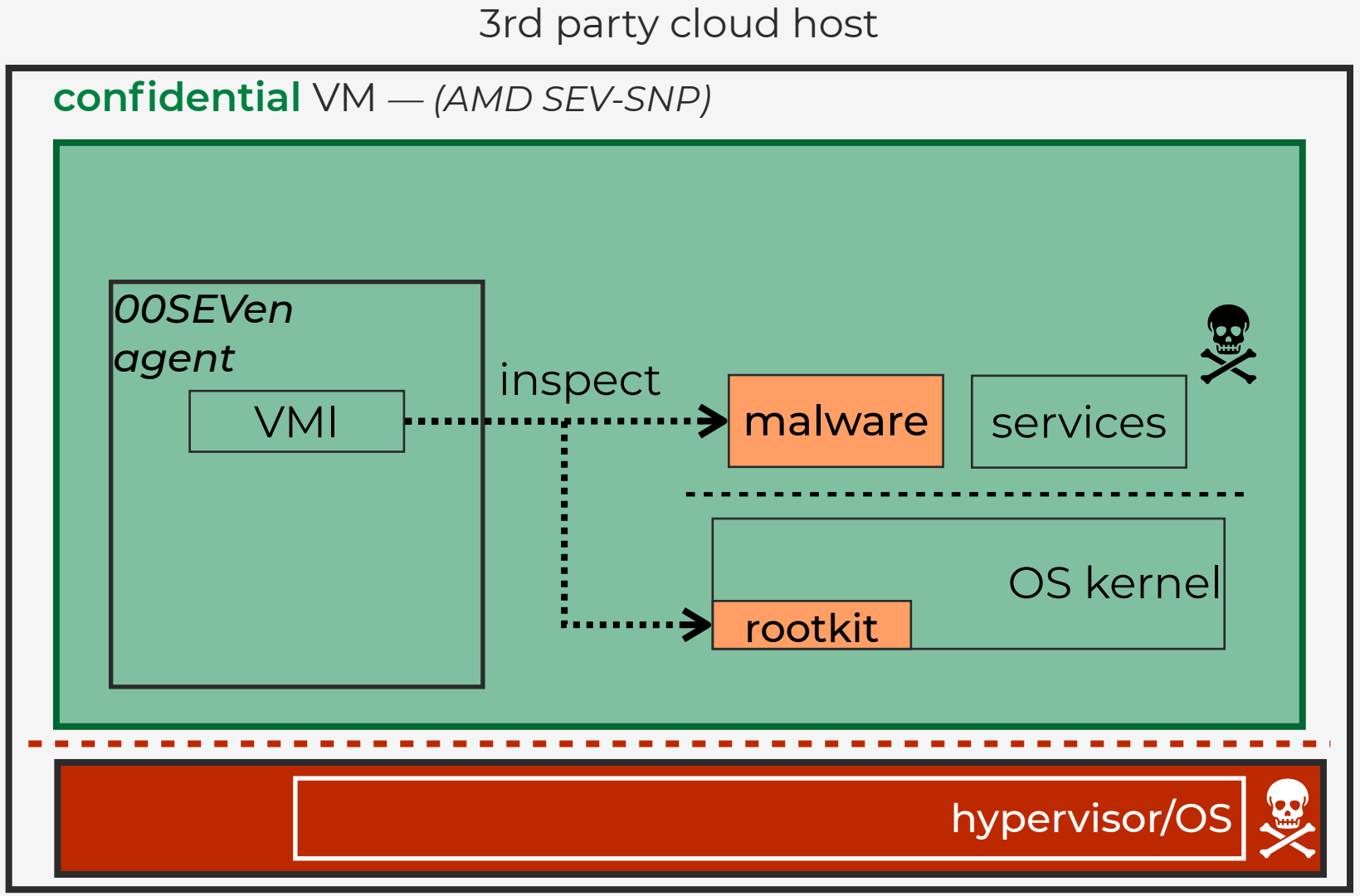
00SEVen Design Overview and Challenges



00SEVen Design Overview and Challenges



00SEVen Design Overview and Challenges



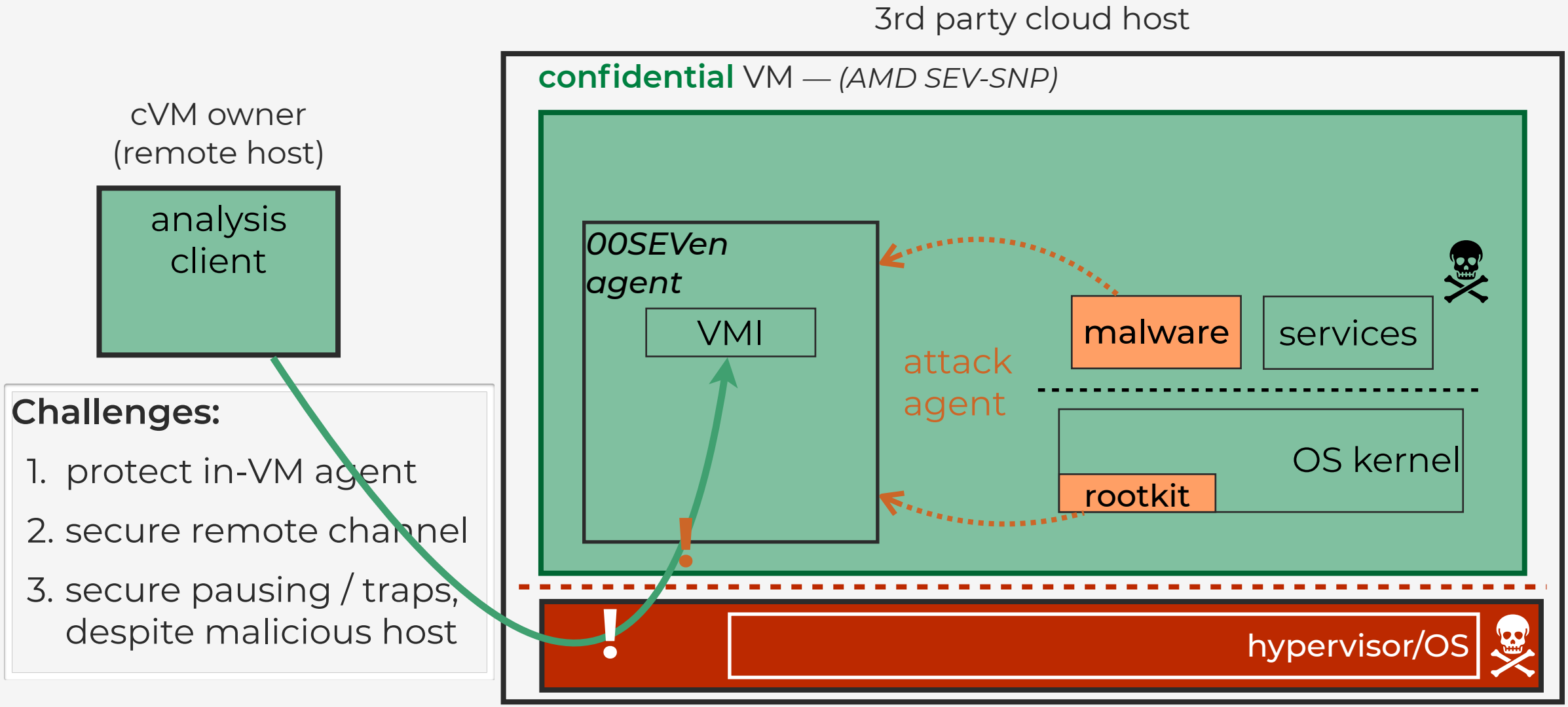
cVM owner
(remote host)

analysis client

Challenges:

1. protect in-VM agent
2. secure remote channel
3. secure pausing / traps, despite malicious host

00SEVen Design Overview and Challenges



Challenges:

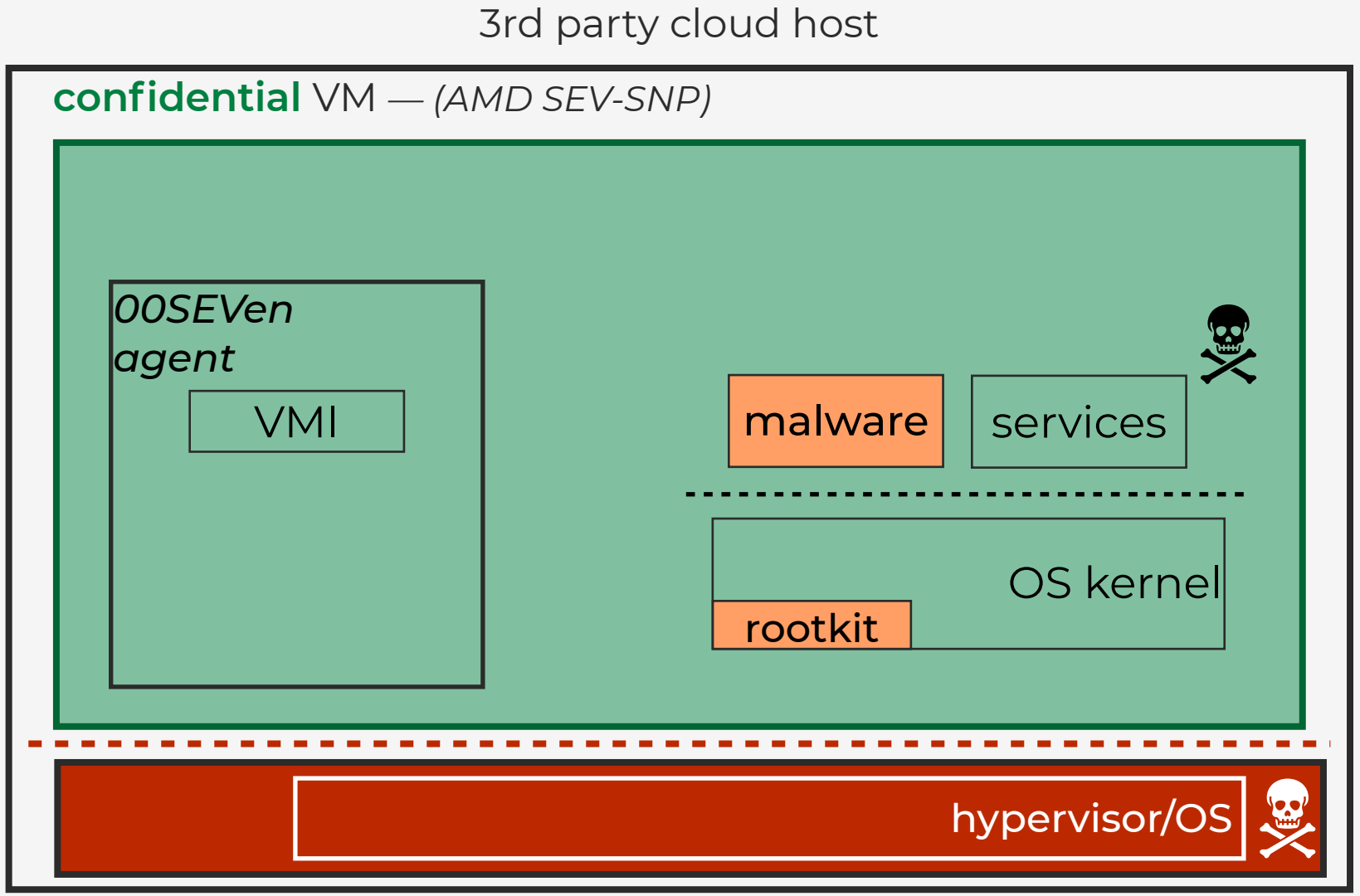
1. protect in-VM agent
2. secure remote channel
3. secure pausing / traps, despite malicious host

00SEVen Challenge #1: Isolation of in-VM Agent

cVM owner
(remote host)

analysis client

- **VMPLs:** hierarchical in-VM CPU modes, orthogonal to user/kernel
- *VMPL0 most privileged*
- **per-VMPL:** page permissions + register sets (per vCPU) saved in cVM memory

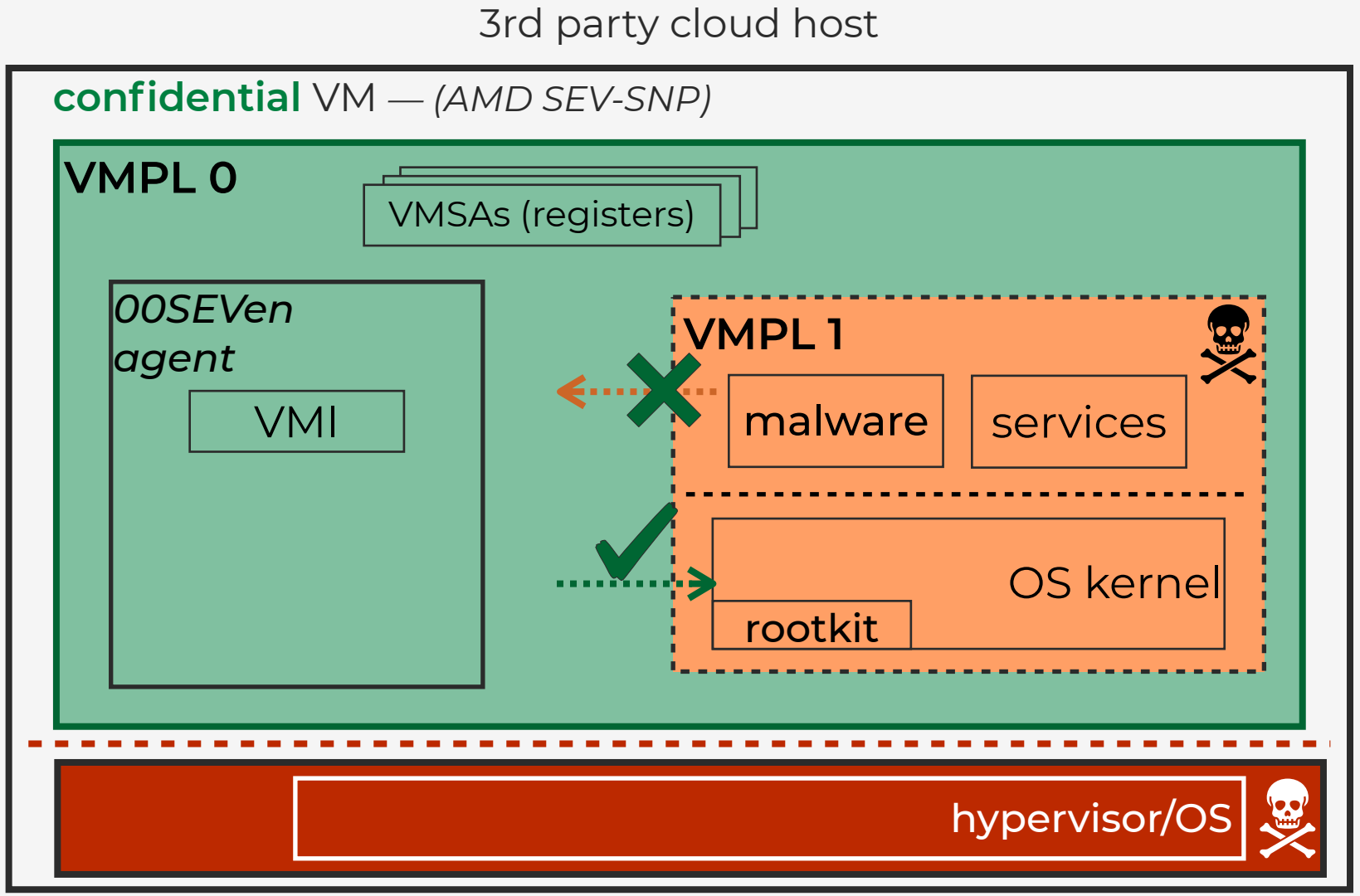


00SEVen Challenge #1: Isolation of in-VM Agent

cVM owner
(remote host)

analysis client

- **VMPLs:** hierarchical in-VM CPU modes, orthogonal to user/kernel
- *VMPL0 most privileged*
- **per-VMPL:** page permissions + register sets (per vCPU) saved in cVM memory

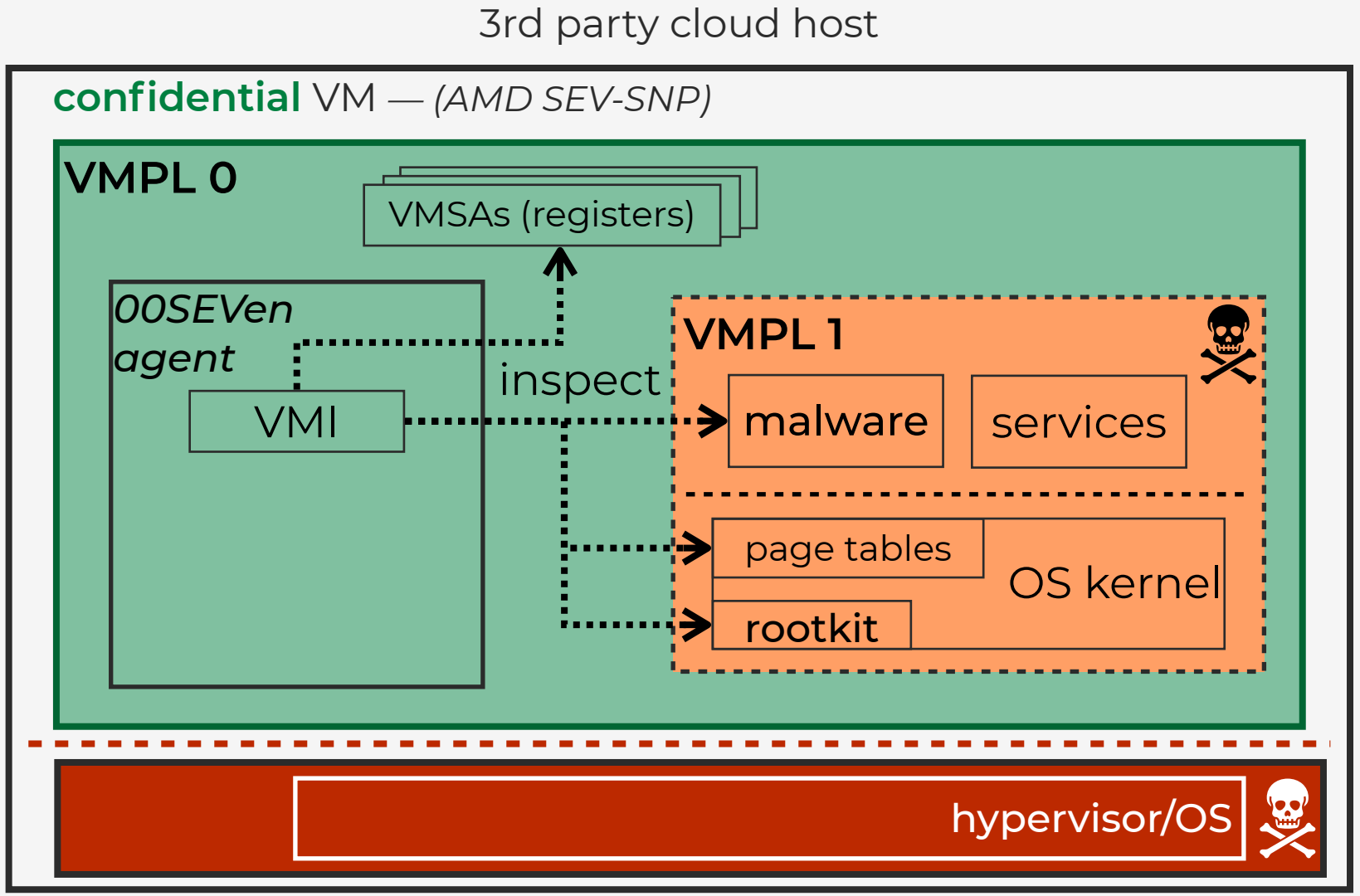


00SEVen Challenge #1: Isolation of in-VM Agent

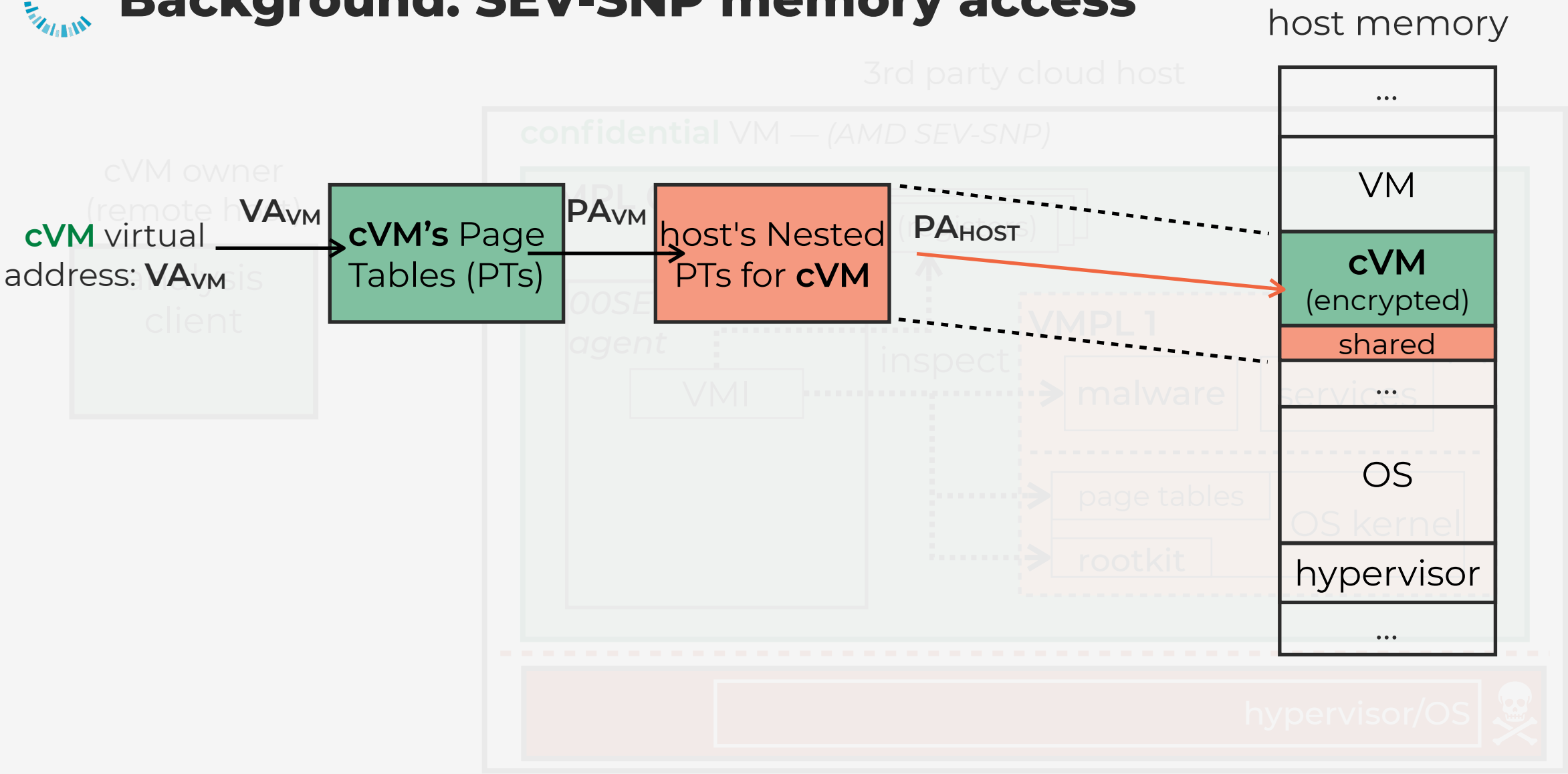
cVM owner
(remote host)

analysis client

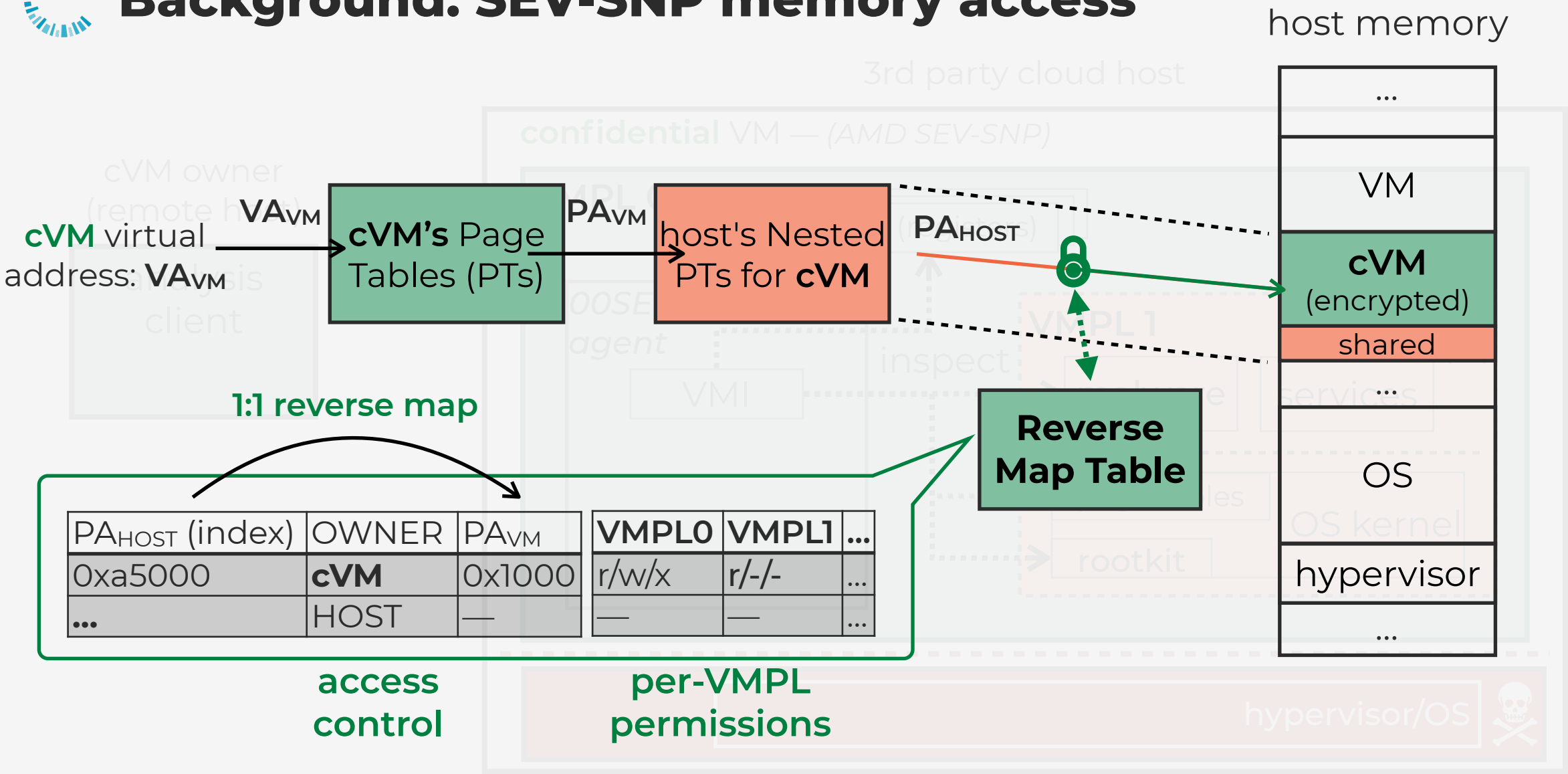
- **VMPLs:** hierarchical in-VM CPU modes, orthogonal to user/kernel
- *VMPL0 most privileged*
- **per-VMPL:** page permissions + register sets (per vCPU) saved in cVM memory



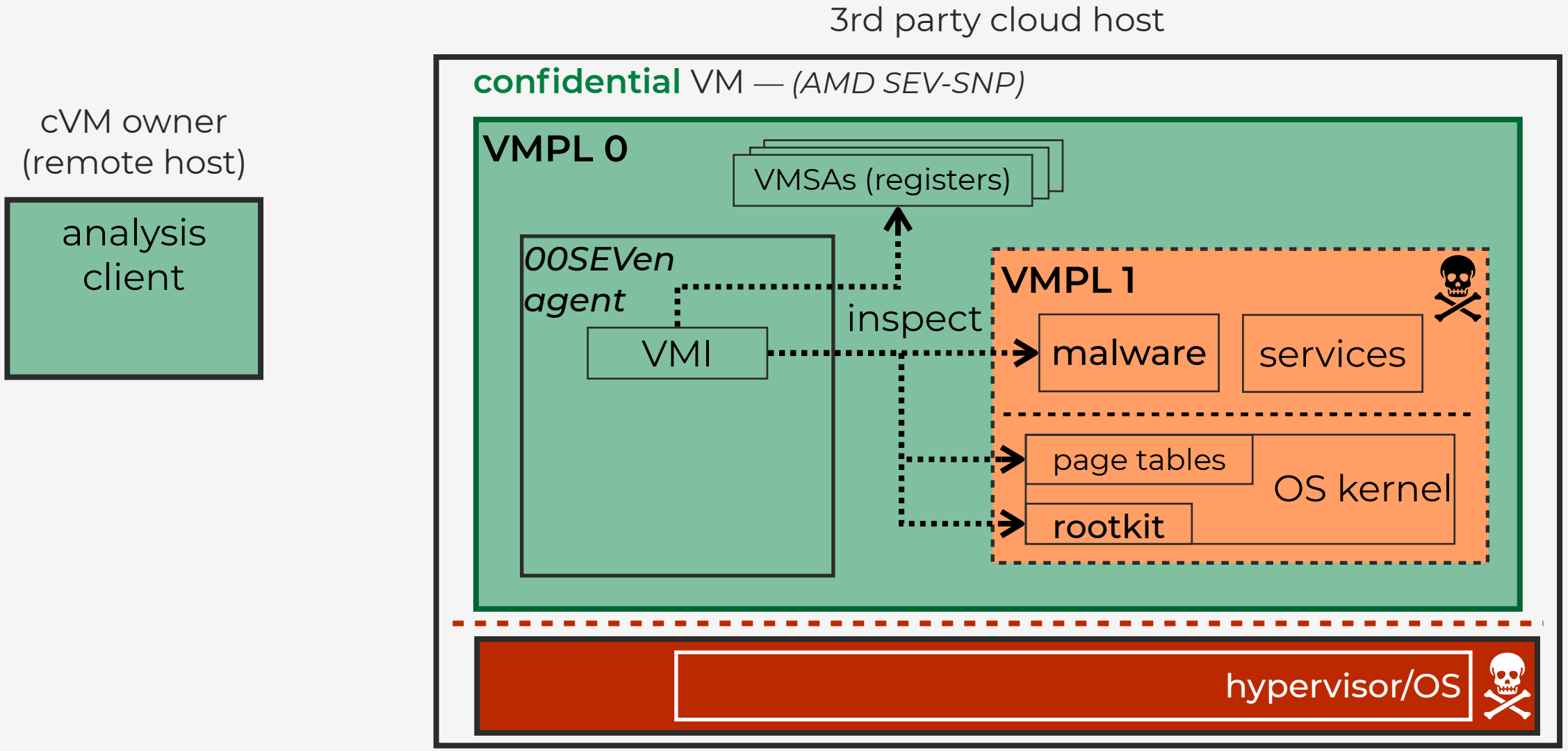
Background: SEV-SNP memory access



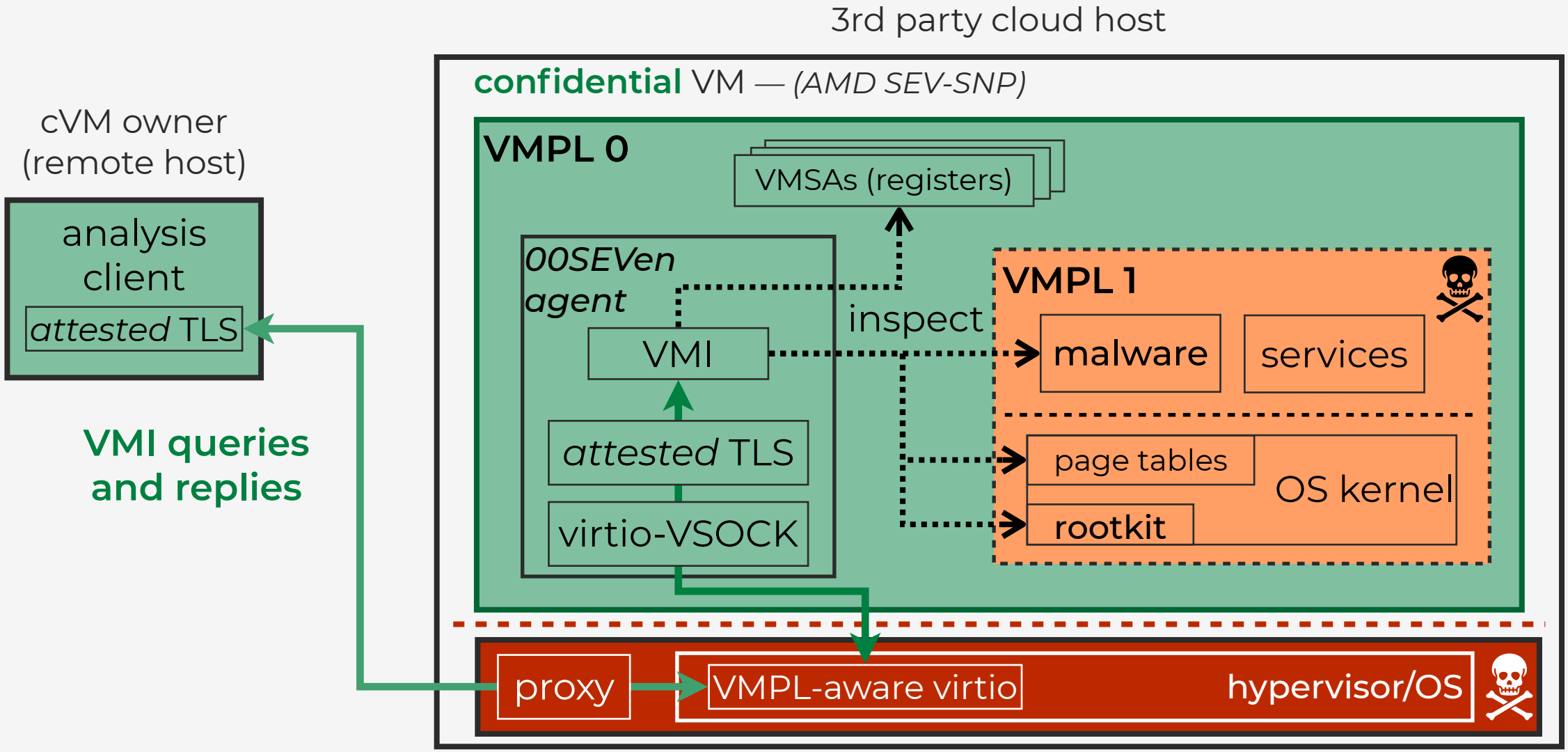
Background: SEV-SNP memory access



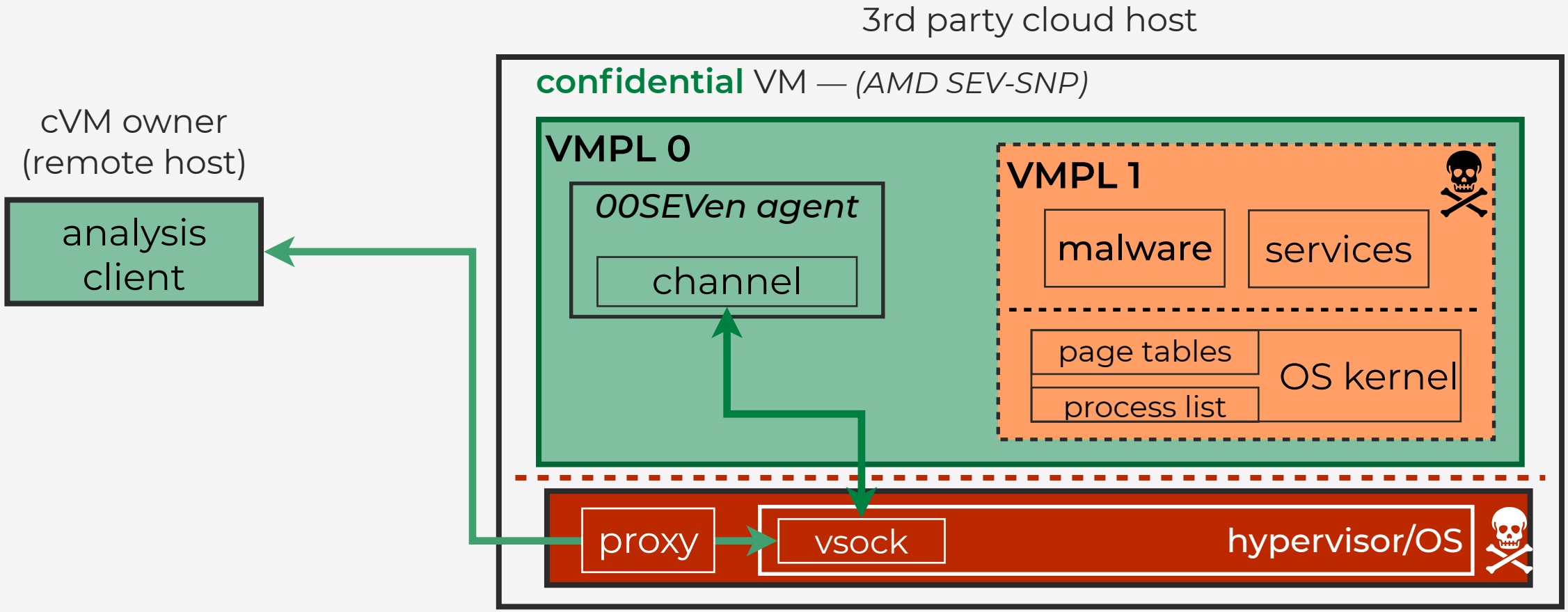
00SEVen Challenge #2: Secure Remote Channel



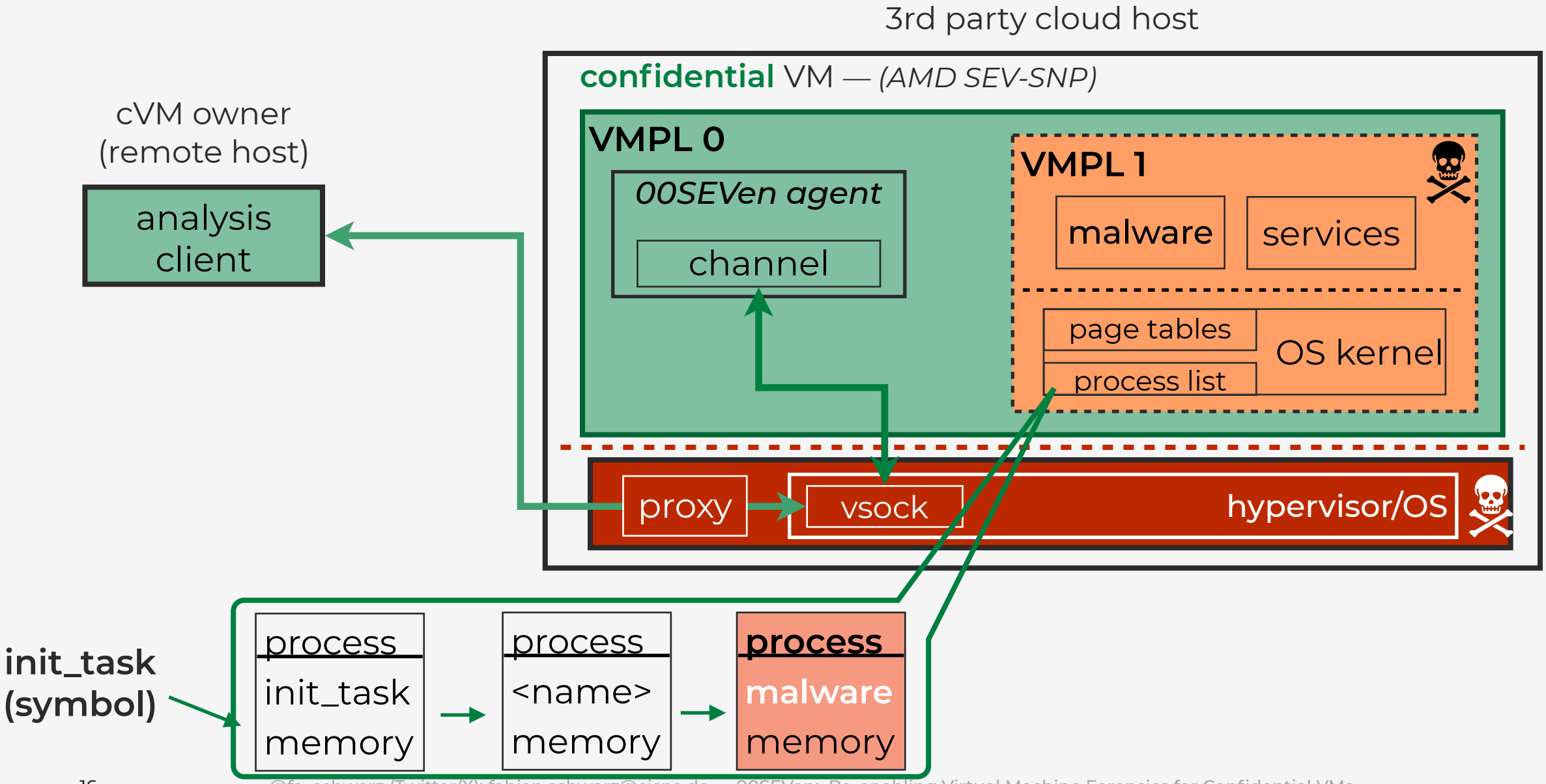
00SEVen Challenge #2: Secure Remote Channel



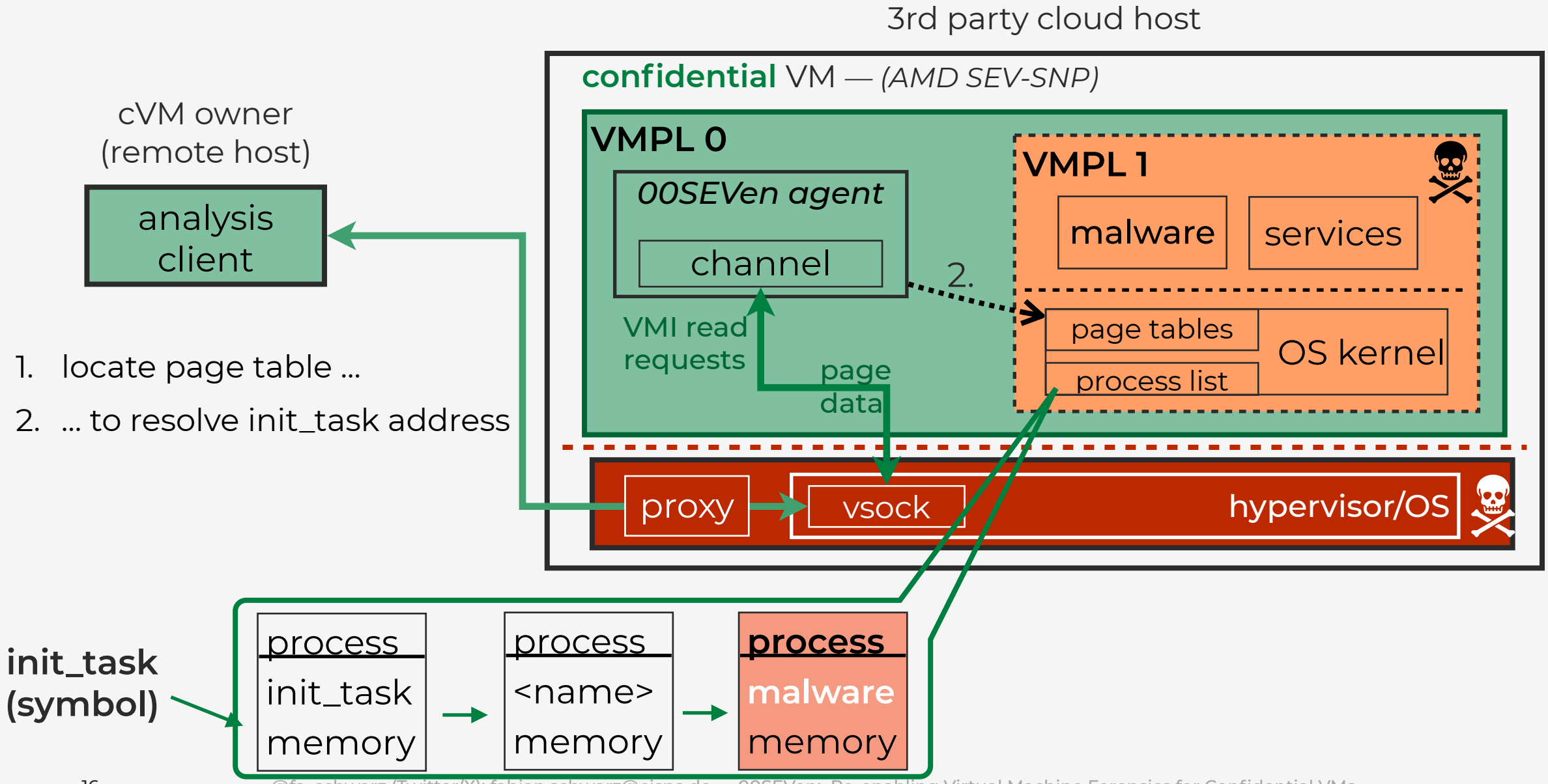
00SEVen Usage Example: Scan Process List for Malware



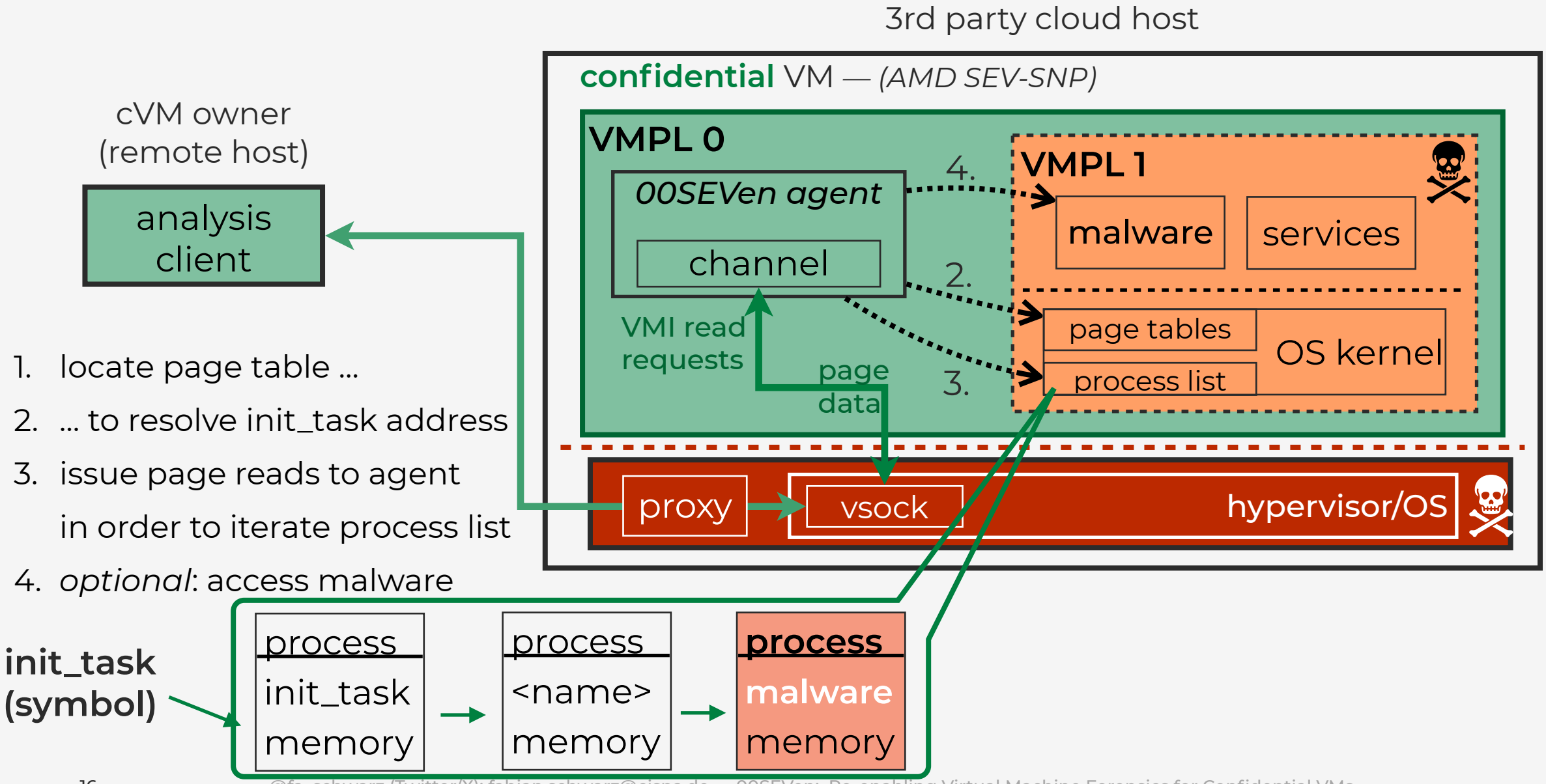
00SEVen Usage Example: Scan Process List for Malware



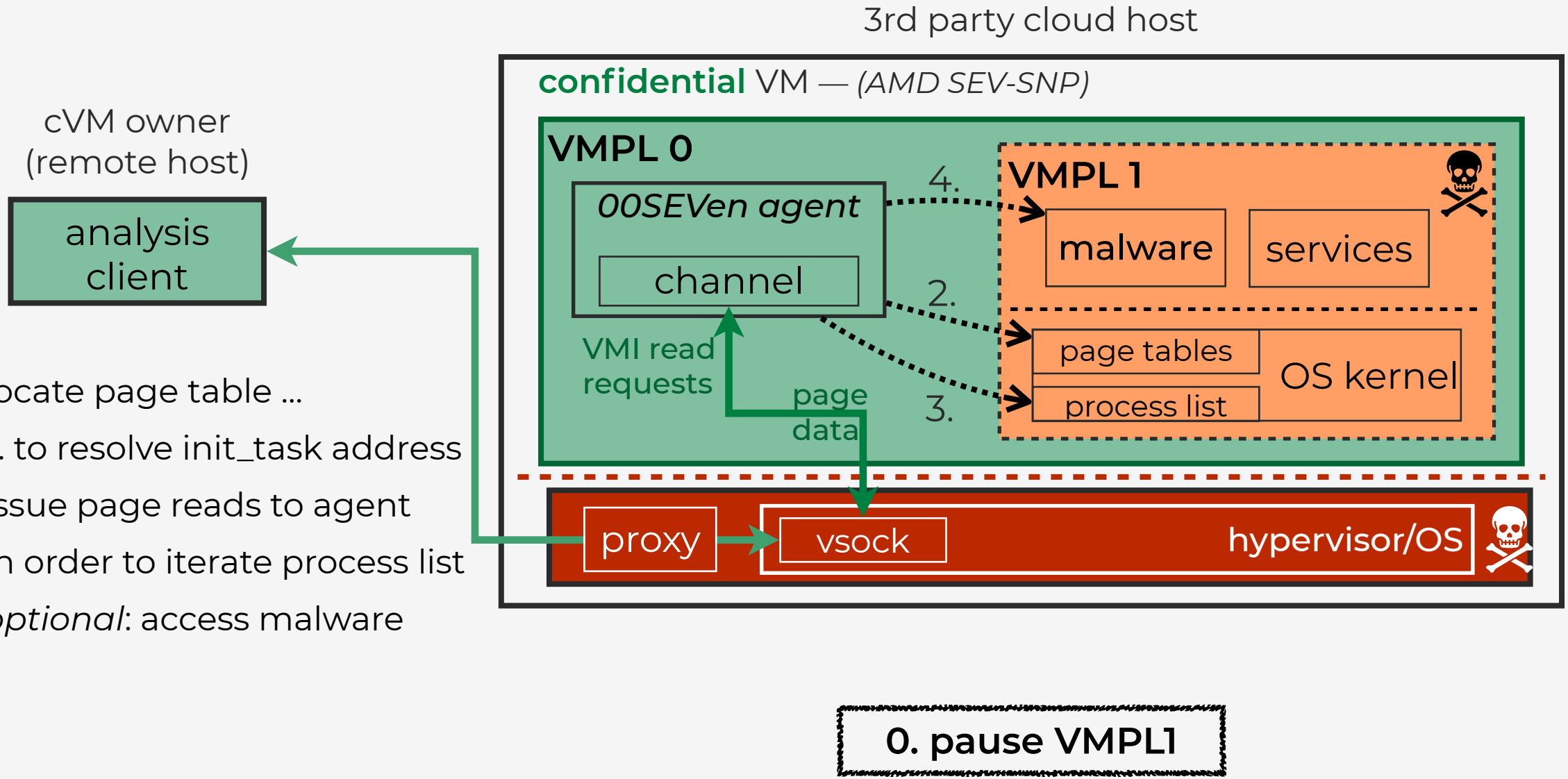
00SEVen Usage Example: Scan Process List for Malware



00SEVen Usage Example: Scan Process List for Malware

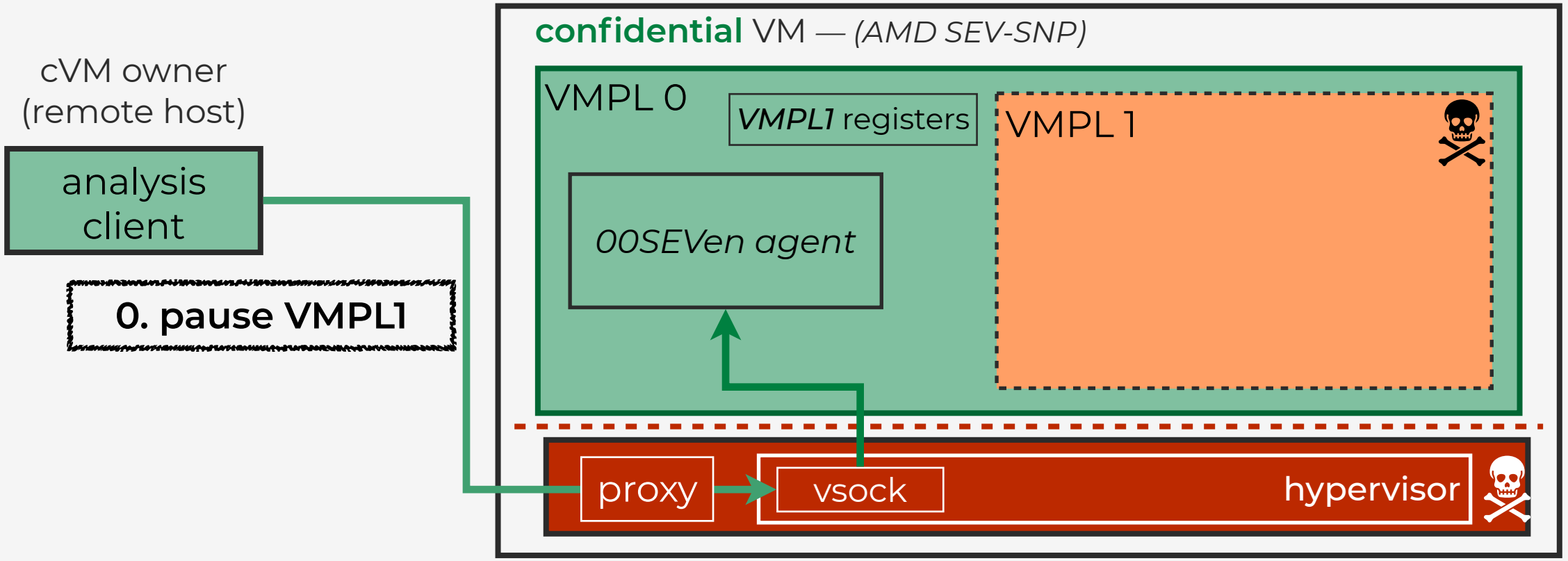


00SEVen Usage Example: Scan Process List for Malware

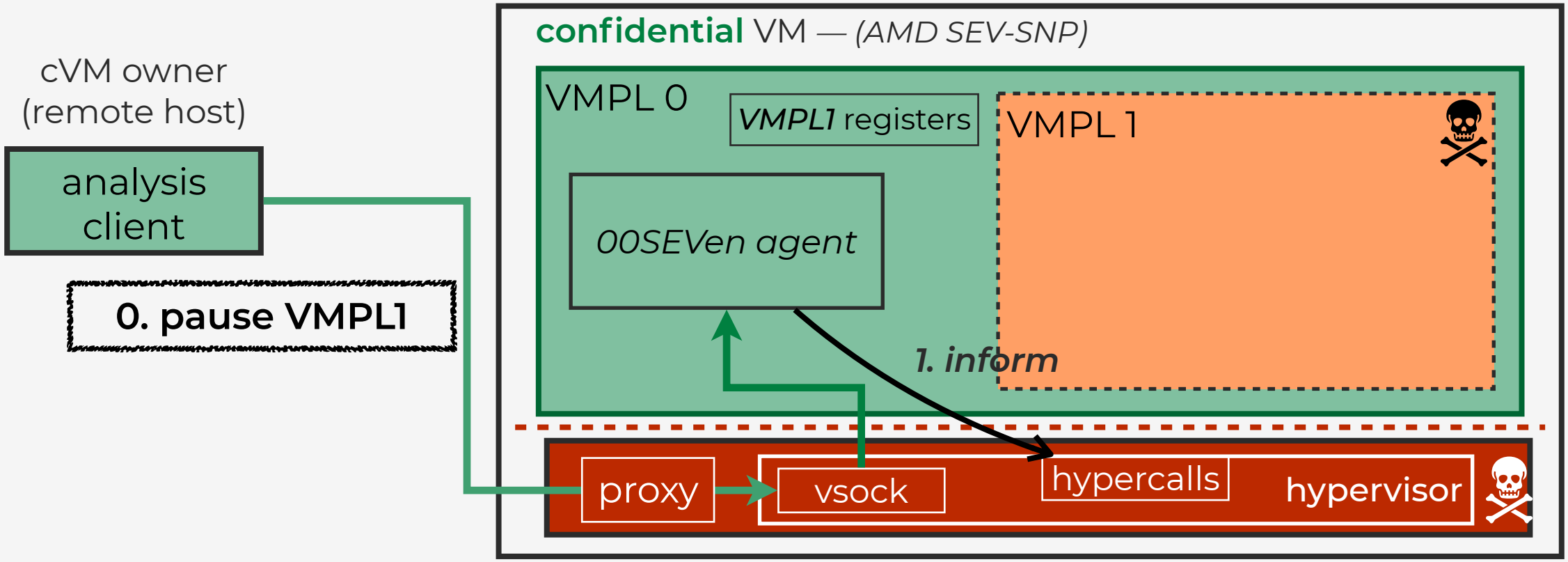


1. locate page table …
2. … to resolve init_task address
3. issue page reads to agent
   in order to iterate process list
4. *optional*: access malware

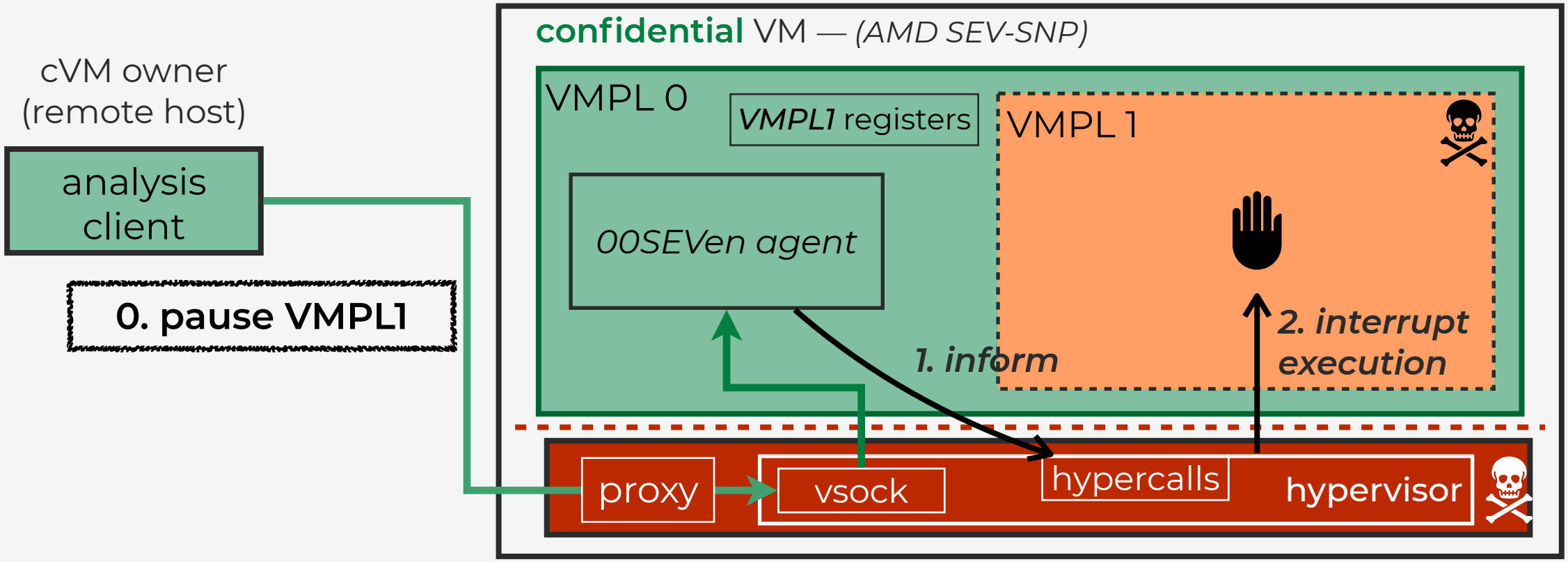
00SEVen Challenge #3: Secure Pausing of VM OS



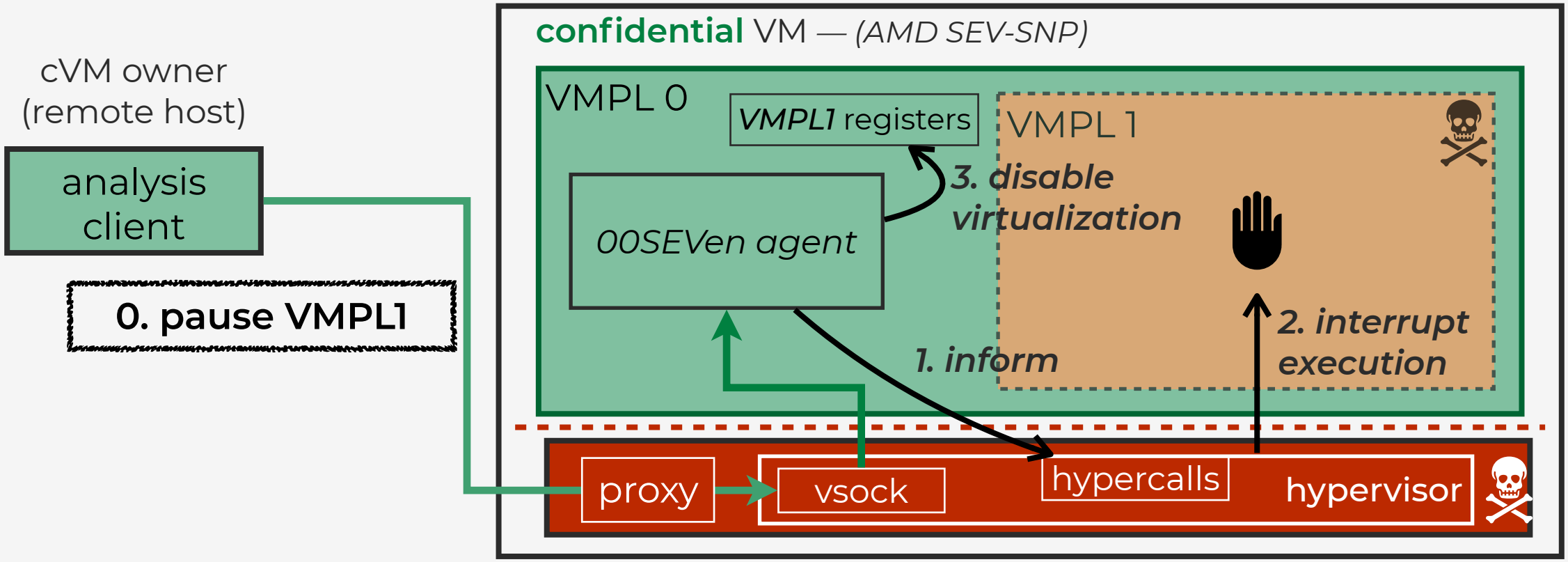
00SEVen Challenge #3: Secure Pausing of VM OS



00SEVen Challenge #3: Secure Pausing of VM OS



00SEVen Challenge #3: Secure Pausing of VM OS



- in-VM agent and hypervisor cooperate to pause VMPL1 execution
- disabling virtualization in VMPL1 registers presents resume during analysis

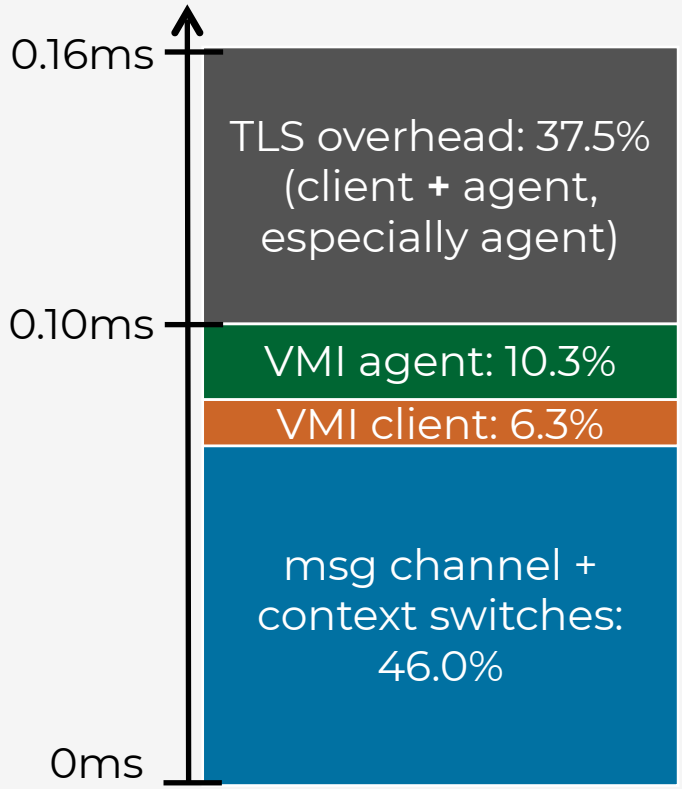
Evaluation Results of 00SEVen Prototype

- **Prototype:** AMD SEV-SNP cVMs, QEMU/KVM hypervisor, VMPL0 built on AMD's SVSM, LibVMI support on client side

- 10 VMI policies (e.g., scan process list) of RDMI (USENIX '23)
- microbenchmarks: page read, address translation
- more results in our paper (e.g., rootkit detection, traps)

- one 4kB-page read**: 0.1 ms (no TLS), 0.16 ms (TLS)**
  ++ network latency for remote inspection

- VMI policy baseline: *KVMi* on *"regular"* VMs on *same host*; vs. 00SEVen on cVMs:
  - 00SEVen client on same host: **+2 / +7 %** (no TLS / TLS)
  - 00SEVen client remote (LAN): **+20 %** (TLS)



Does 00SEVen solve the initial challenge?

GOAL: re-enable isolated VMI for cVMs — without breaking their security

- efficient remote inspection via secure in-VM agent
- VMI features:
  - memory + register access
  - VMPL1 pausing
  - event-based VMI via memory/function traps
    — *see paper for details*

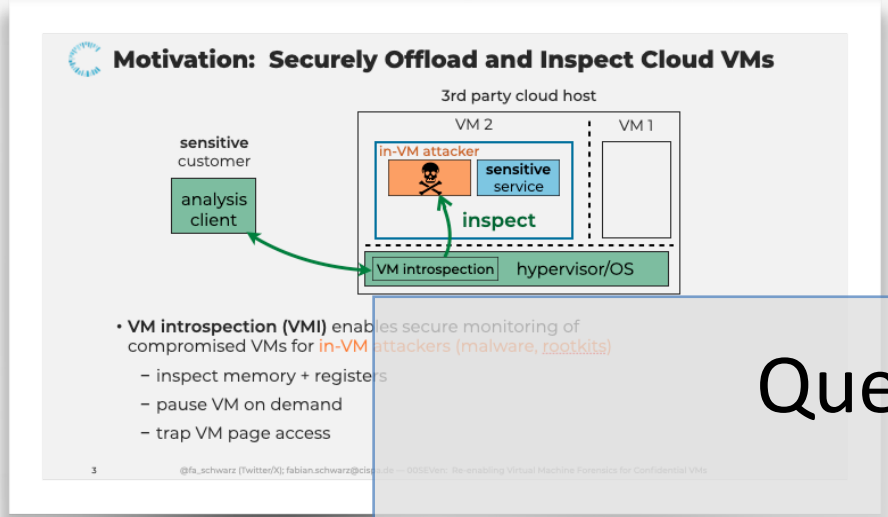
Does 00SEVen solve the initial challenge?

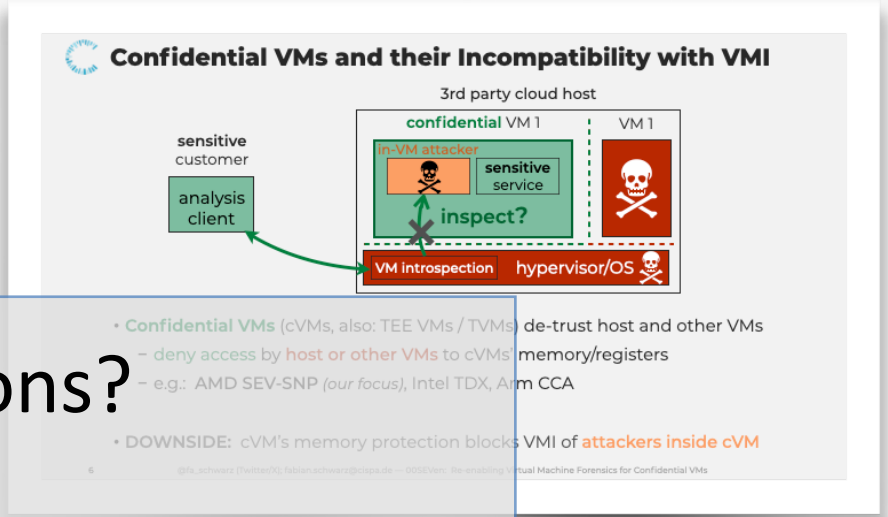
GOAL: re-enable isolated VMI for cVMs — without breaking their security

- efficient remote inspection via secure in-VM agent
- VMI features:
  - memory + register access
  - VMPL1 pausing
  - event-based VMI vi… …raps
    — *see pape…*

00SEVen combines the advantages of both: cVMs and VMI
— enabling cloud usage by sensitive customers

Summary — 00SEVen: Re-enabling VMI for cVMs





Questions?

fabian.schwarz@cispa.de

/ 𝕏 @fa_schwarz

https://github.com/sev-vmi/00seven