Trust Your Computer Less

Emmett Witchel

Alan M. Dunn, Owen S. Hofmann, Michael Z. Lee, Suman Jana, Sangman Kim, Mark Silberstein, Yuanzhong Xu, Vitaly Shmatikov

University of Texas at Austin

Eternal Sunshine of the Spotless Machine: Protecting Privacy with Ephemeral Channels

OSDI 2012

Wanted: Application Privacy

- Goal: Run programs without leaving traces

VoIP conversation with lawyer

Biomedical researcher accessing data

Website access

- Current state: Private browsing
  - Popular feature in web browsers
  - Ideal: When private browsing session terminates, all traces erased

A Privacy Problem

- Private browsing unachieved
  - Evidence of site visits leaks into OS [Aggrawal, 2010]
- Problem: No system support
  - Applications interact with user and world
  - Data leaks into OS, system services
  - Applications cannot remove traces they leave

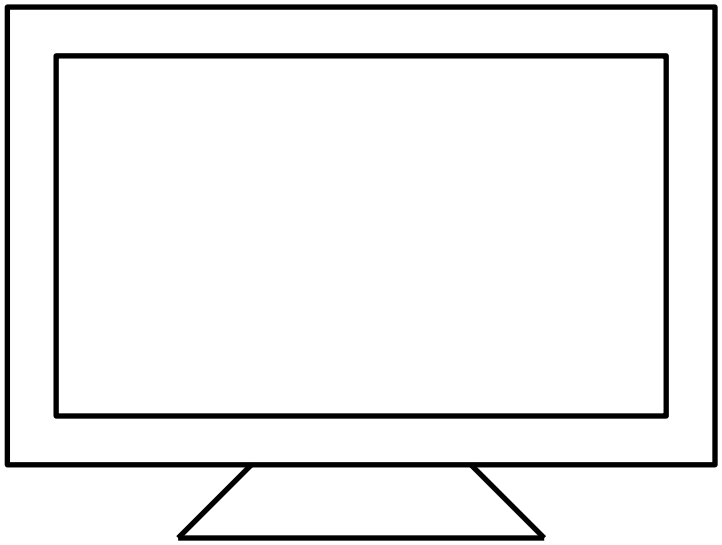
Example: Browsing a Website

Network

Audio

What traces still remain on the computer?

Leaks From Browsing



Network

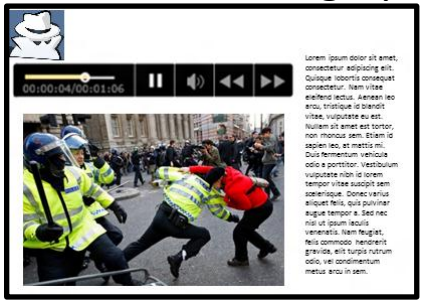
Memory contents:

Complete packets, like:

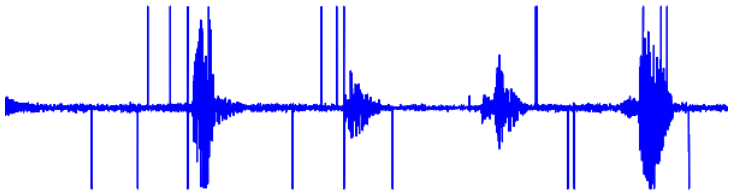
```
HTTP/1.1 200 OK
Date: Mon, 17 Sep 2012 …
Server: Apache/2.2.14 …
…
```

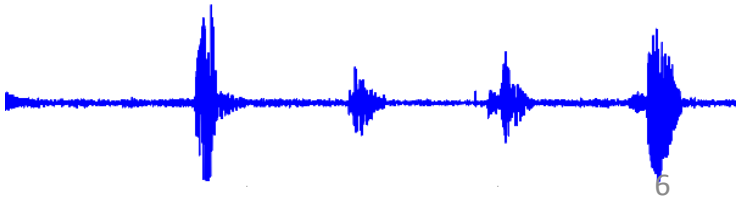
Audio

X server caches, graphics drivers



PulseAudio server





Secure Deallocation Is Not Enough

- **Secure deallocation**: Zero memory when freed
  - Research implementation [Chow, 2005]
  - PaX: Security patch for Linux kernel
- Sensitive data remains allocated
  - X caches, PulseAudio buffers not freed

Resisting a Strong Adversary

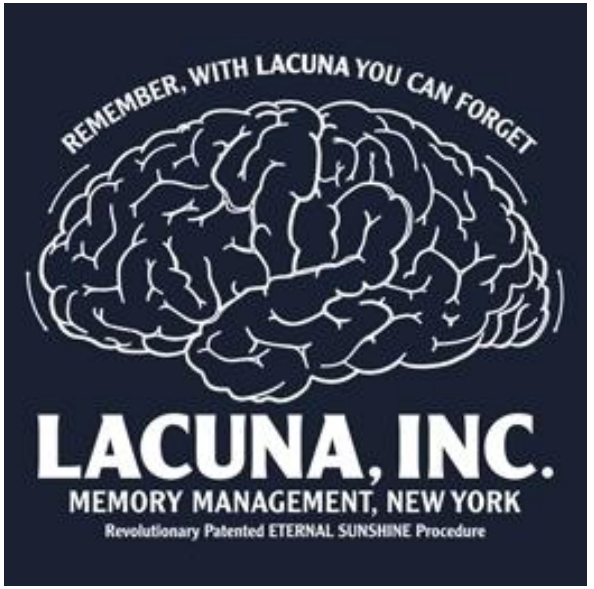
- Goal: Provide **forensic deniability** – no evidence left for non-concurrent attacker
- Once program terminated, protection maintained under extreme circumstances

Root-level compromise
(after program terminates)

Computer physically seized

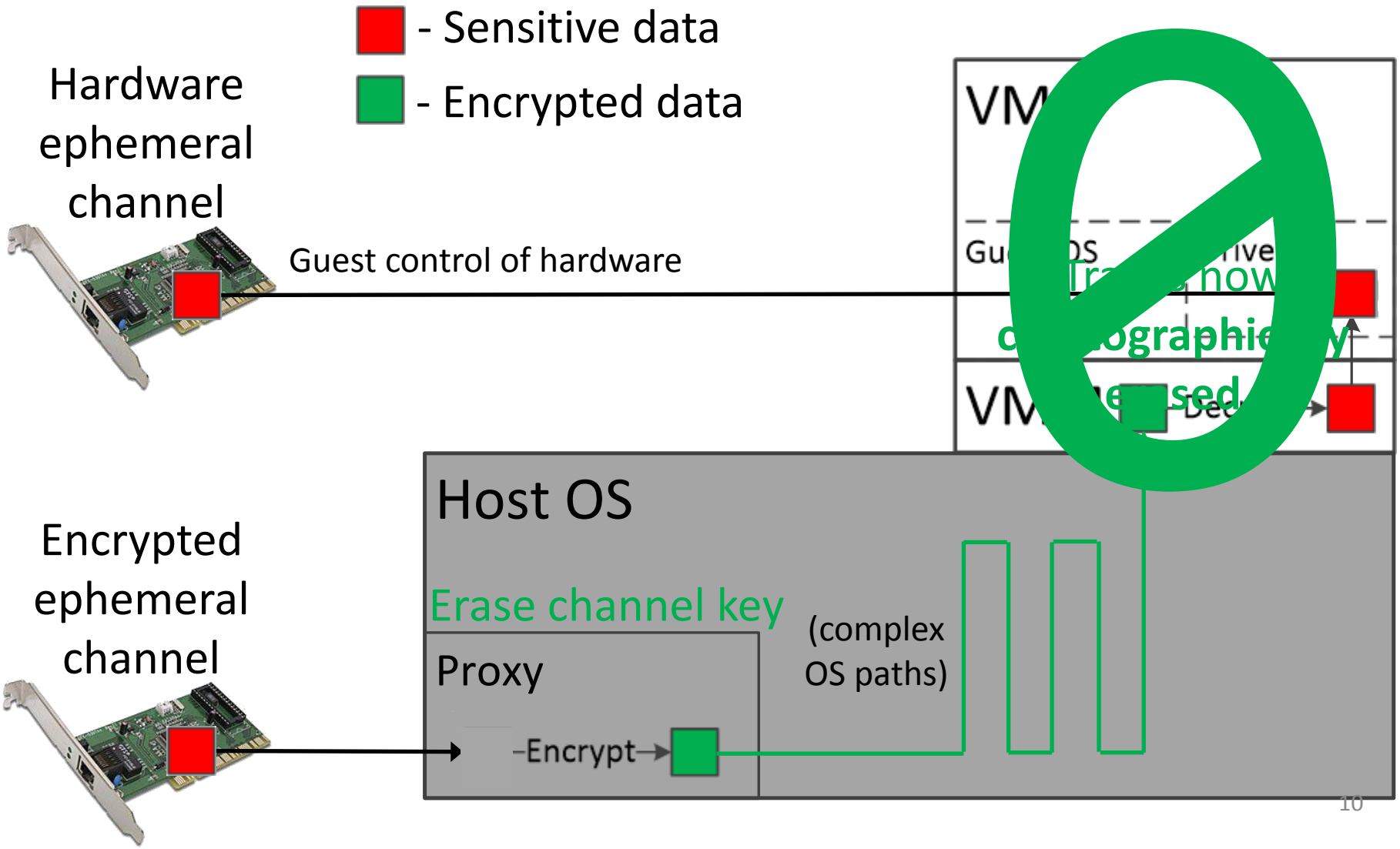
Lacuna

- System to accomplish our privacy and usability goals
- Host OS (Linux), VMM (QEMU-KVM) modified
  - Application runs in VM
- Applications unmodified



la·cu·na [luh-kyoo-nuh]
1. a gap or missing part, as in a manuscript, series, or logical argument...

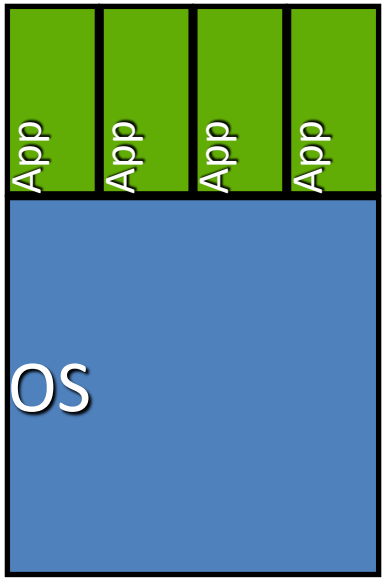
Ephemeral Channels



Don’t Trust Your OS

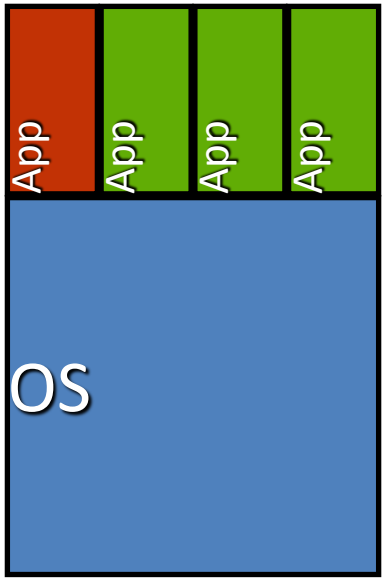
Don’t trust the OS

- The OS is a shared vulnerability
  - OS compromise infects all
- The OS is a vulnerable vulnerability
  - Syscall interface a complex attack surface
    - ioctl()
- Root often has OS-level privilege



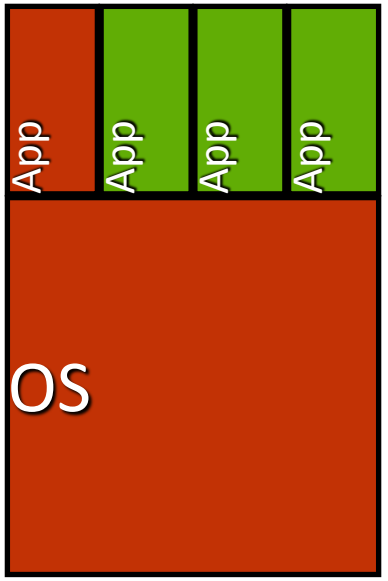
Don’t trust the OS

- The OS is a shared vulnerability
  - OS compromise infects all
- The OS is a vulnerable vulnerability
  - Syscall interface a complex attack surface
    - ioctl()
- Root often has OS-level privilege



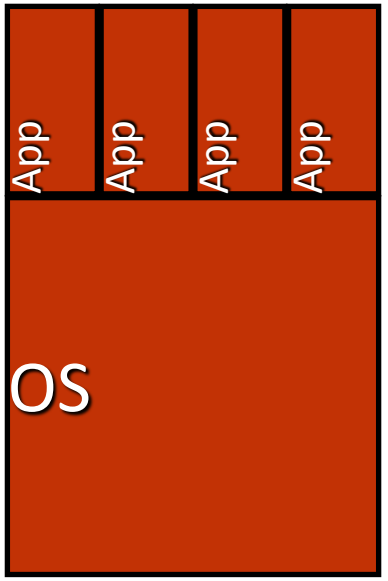
Don’t trust the OS

- The OS is a shared vulnerability
  - OS compromise infects all
- The OS is a vulnerable vulnerability
  - Syscall interface a complex attack surface
    - ioctl()
- Root often has OS-level privilege



Don’t trust the OS

- The OS is a shared vulnerability
  - OS compromise infects all
- The OS is a vulnerable vulnerability
  - Syscall interface a complex attack surface
    - ioctl()
- Root often has OS-level privilege



Untrusted OS, Trusted App & VMM

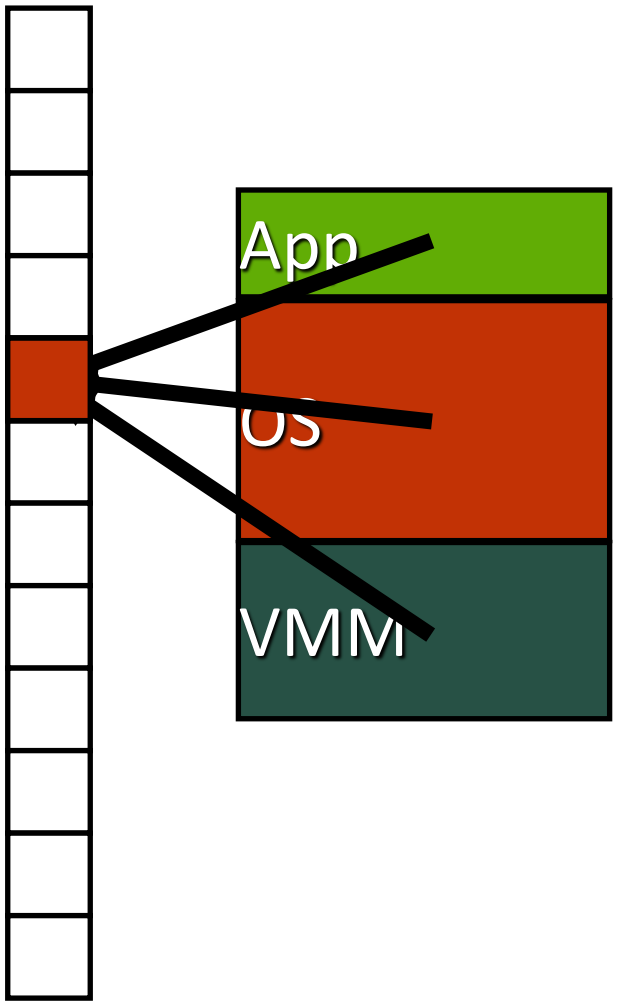
- Home user
  - Runs a small, secure hypervisor
    - E.g., SecVisor, TrustVisor, xmhf
  - Wants to visit sketchy websites
    - Picture of Nodar Kumaritashvili's luge crash
  - Wants to do online banking
- OS-level malware
  - Does not compromise privacy or integrity of banking application
  - Can deny service

What is InkTag?

- Hypervisor modifications
  - Keep them small and simple
  - Uses modern virtualization hardware
    - VMM in charge of page tables
- libc modifications
  - E.g., manages data for system calls
- Potential application changes
- OS changes
  - But I thought you said the OS was untrusted?
- Similar to: Overshadow, SP^3, Cloudvisor

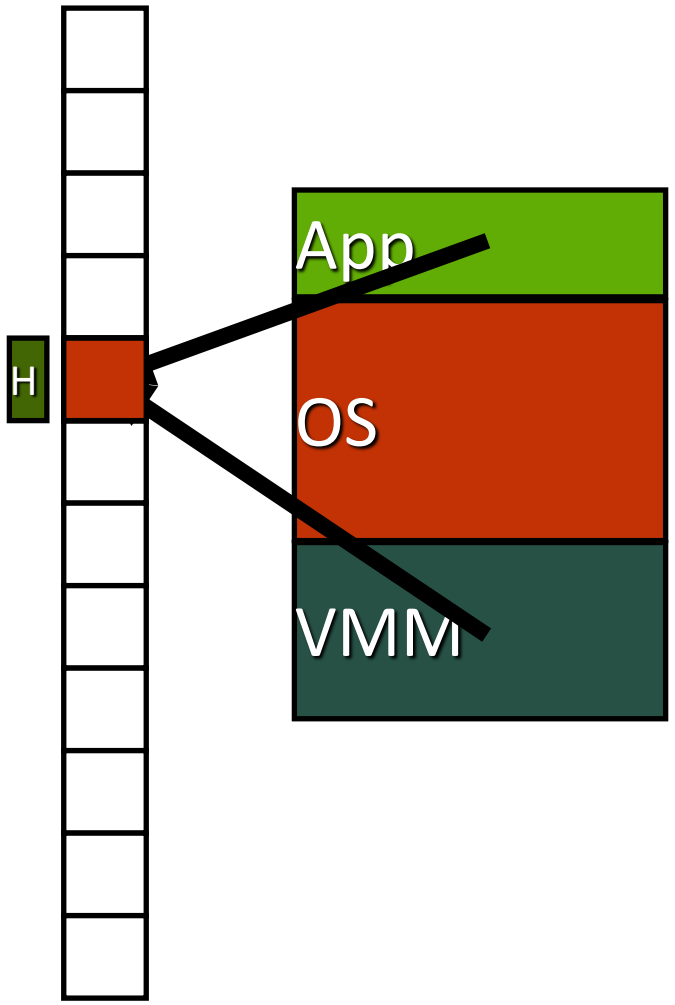
Isolation in Overshadow

- Isolate control flow, register contents
  - Secure context switch
- Isolate memory
  - OS expects to manage memory
  - Show cleartext to application
  - Show ciphertext to OS
  - Hash for integrity



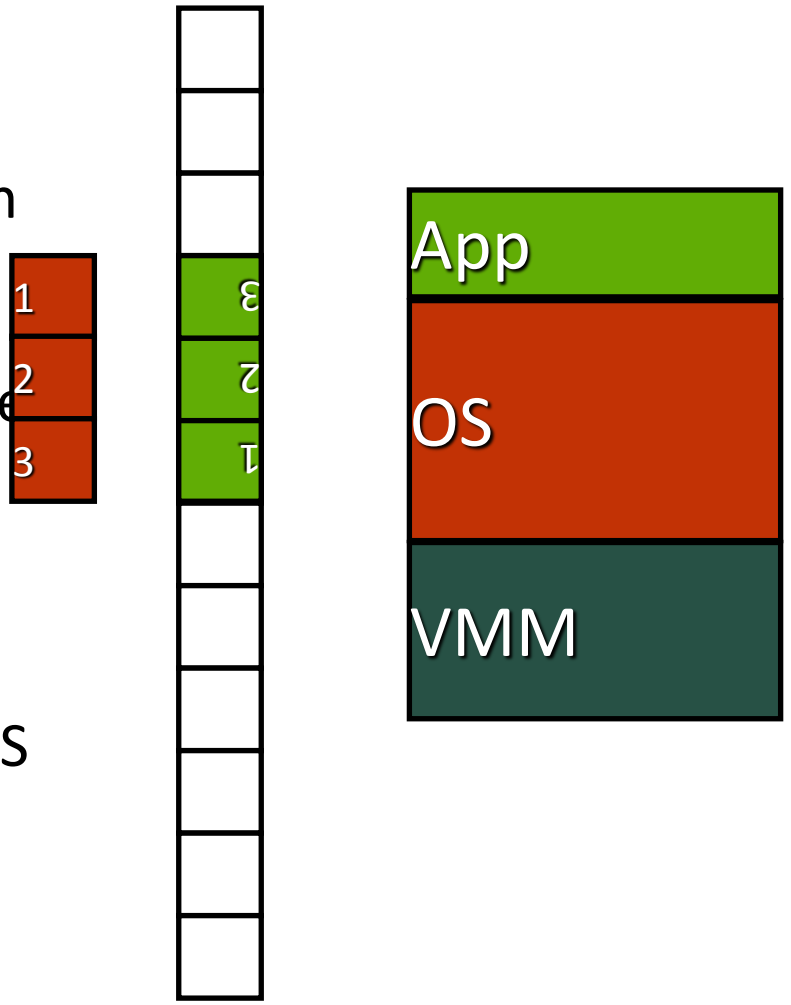
Isolation in Overshadow

- Isolate control flow, register contents
  - Secure context switch
- Isolate memory
  - OS expects to manage memory
  - Show cleartext to application
  - Show ciphertext to OS
  - Hash for integrity



Isolation in Overshadow

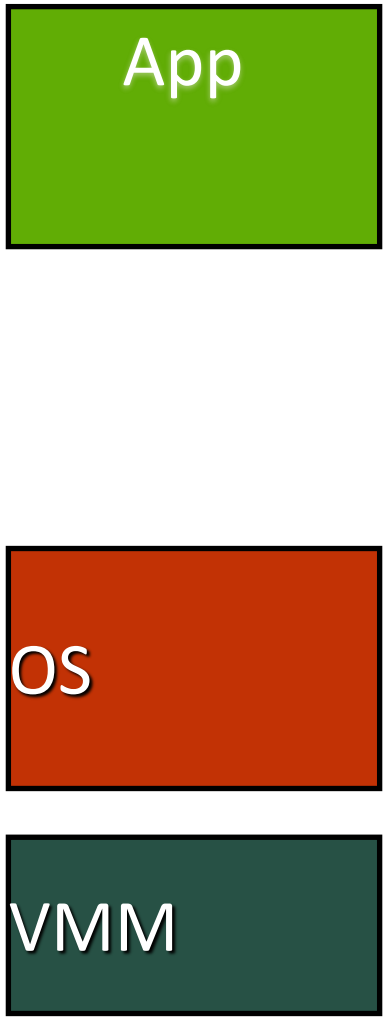
- Isolate control flow, register contents
  - Secure context switch
- Isolate memory
  - OS expects to manage memory
  - Show cleartext to application
  - Show ciphertext to OS
  - Hash for integrity



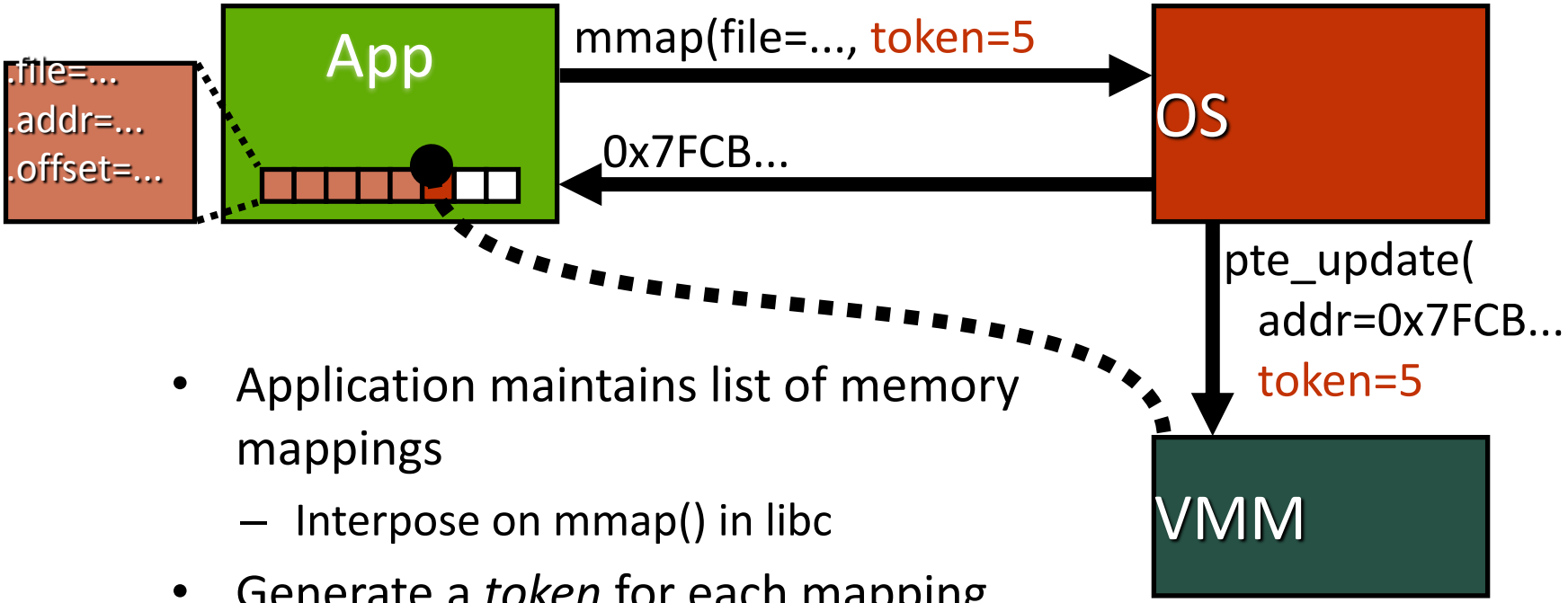
- Ensure OS constructs the correct address space

The (untrusted) OS can help

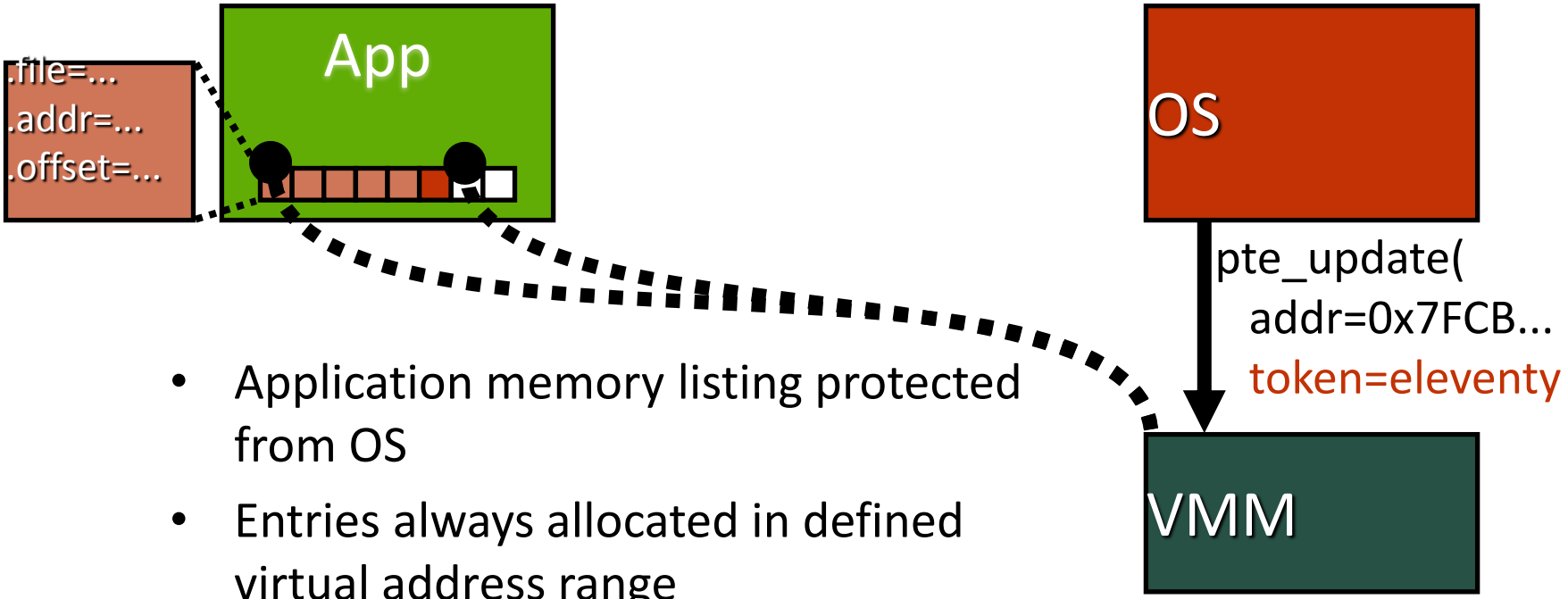
- *Paraverification*: an untrusted OS participates in its own verification
  - Take inspiration from paravirtualization
  - Extensive use of existing paravirtual interface



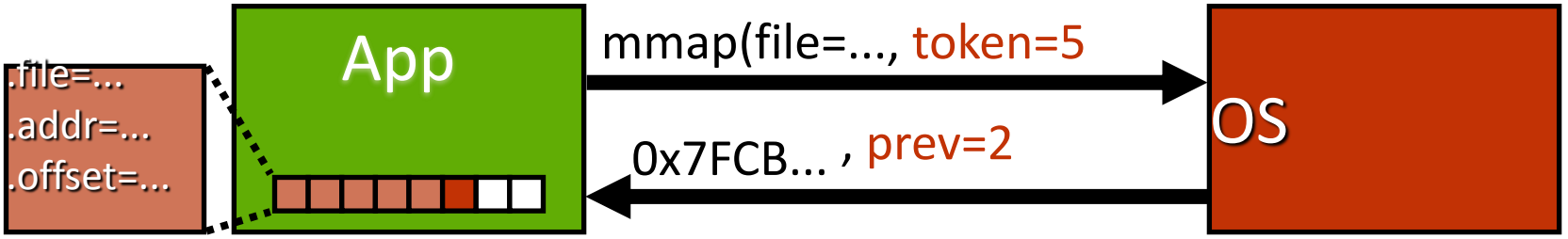
- Untrusted OS notifies VMM on page table updates
  - Regular structure
  - In update order



- Application maintains list of memory mappings
  - Interpose on mmap() in libc
- Generate a *token* for each mapping
  - Unforgeable identifier describing requested mapping
  - Index trusted array (e.g., file descriptor)



- Application memory listing protected from OS
- Entries always allocated in defined virtual address range
- Invalid entries marked



- OS returns tokens to application to assist validation
  - Application maintains linked list of mappings
  - OS specifies previous entry
  - Application checks for overlap, updates list

VMM