



Virtualization Enabled Integrity Services (VIS)

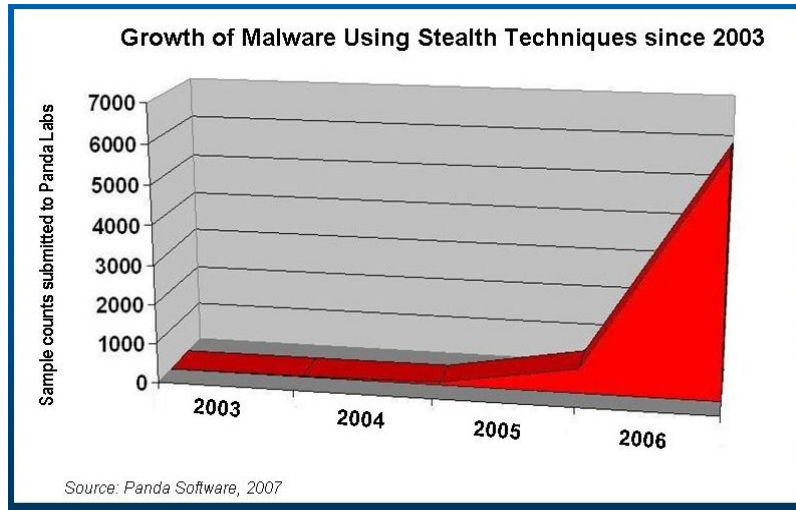
Vedvyas Shanbhogue, Ravi Sahita, Uday
Savagaonkar

(vedvyas.shanbhogue@intel.com, ravi.sahita@intel.com,
uday.savagaonkar@intel.com)

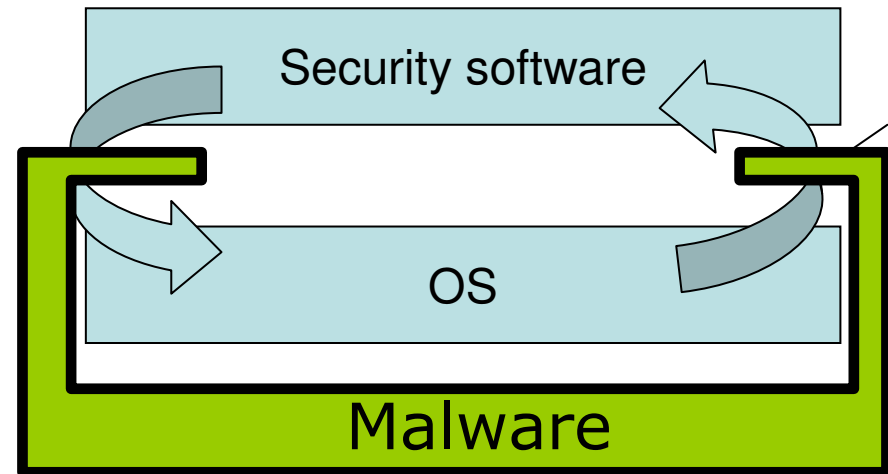


Motivation

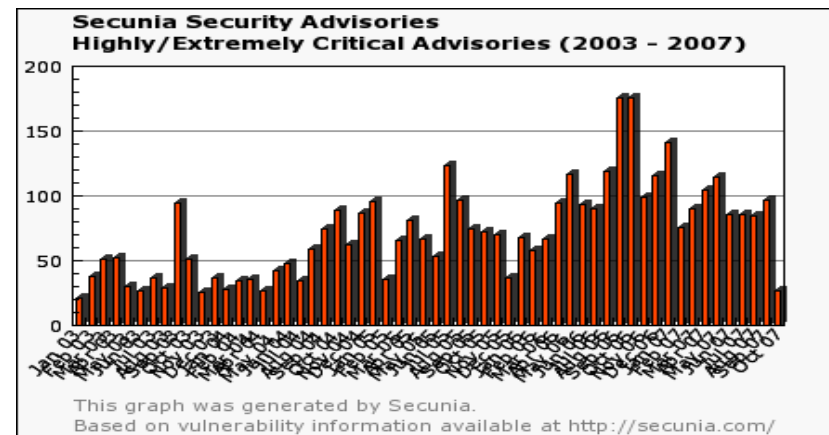
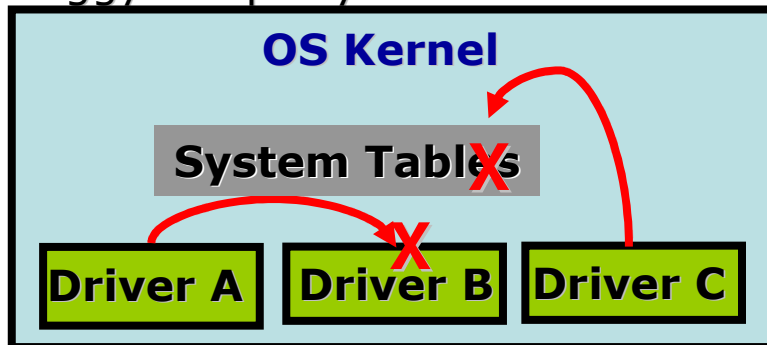
Malware and Stealth ware on the Rise



Rootkits subvert critical services

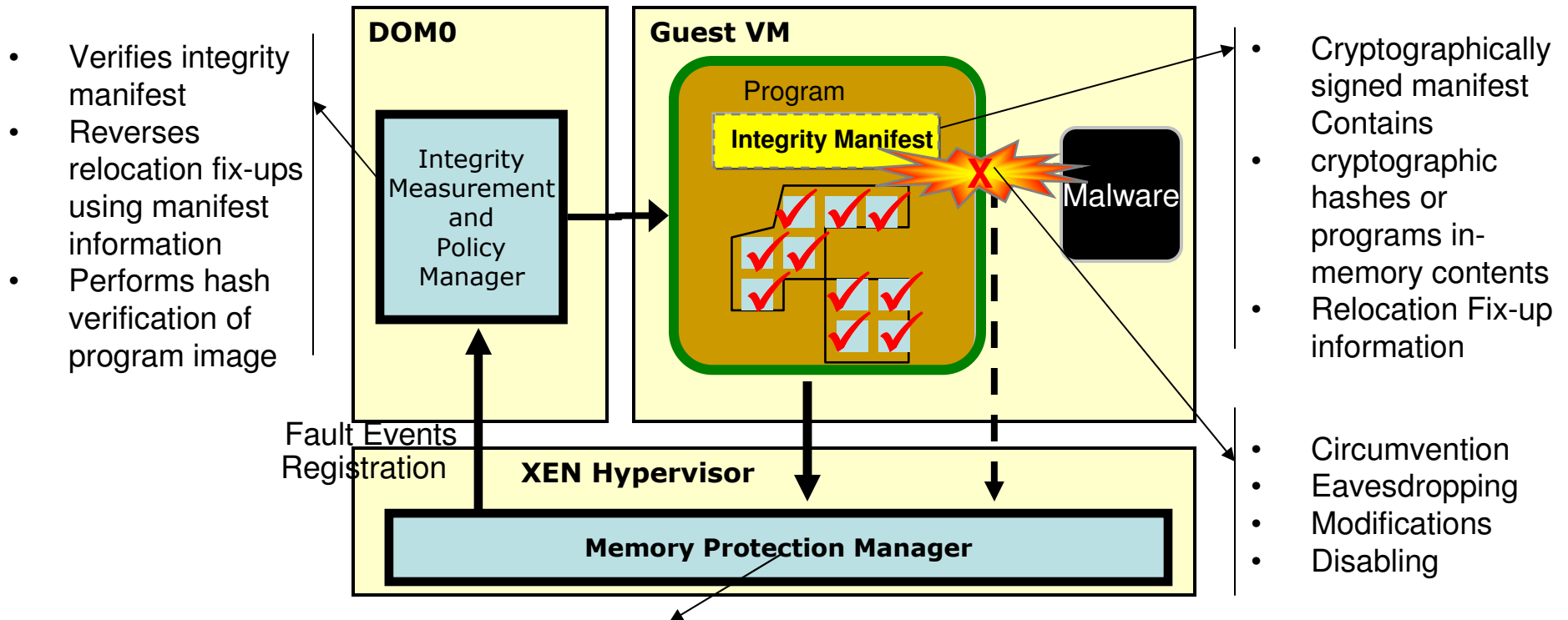


Buggy 3rd party drivers in Kernel



Shared Memory Spaces = Open Attack/Fault Surfaces

Virtualization Enabled Integrity Service (VIS) Overview



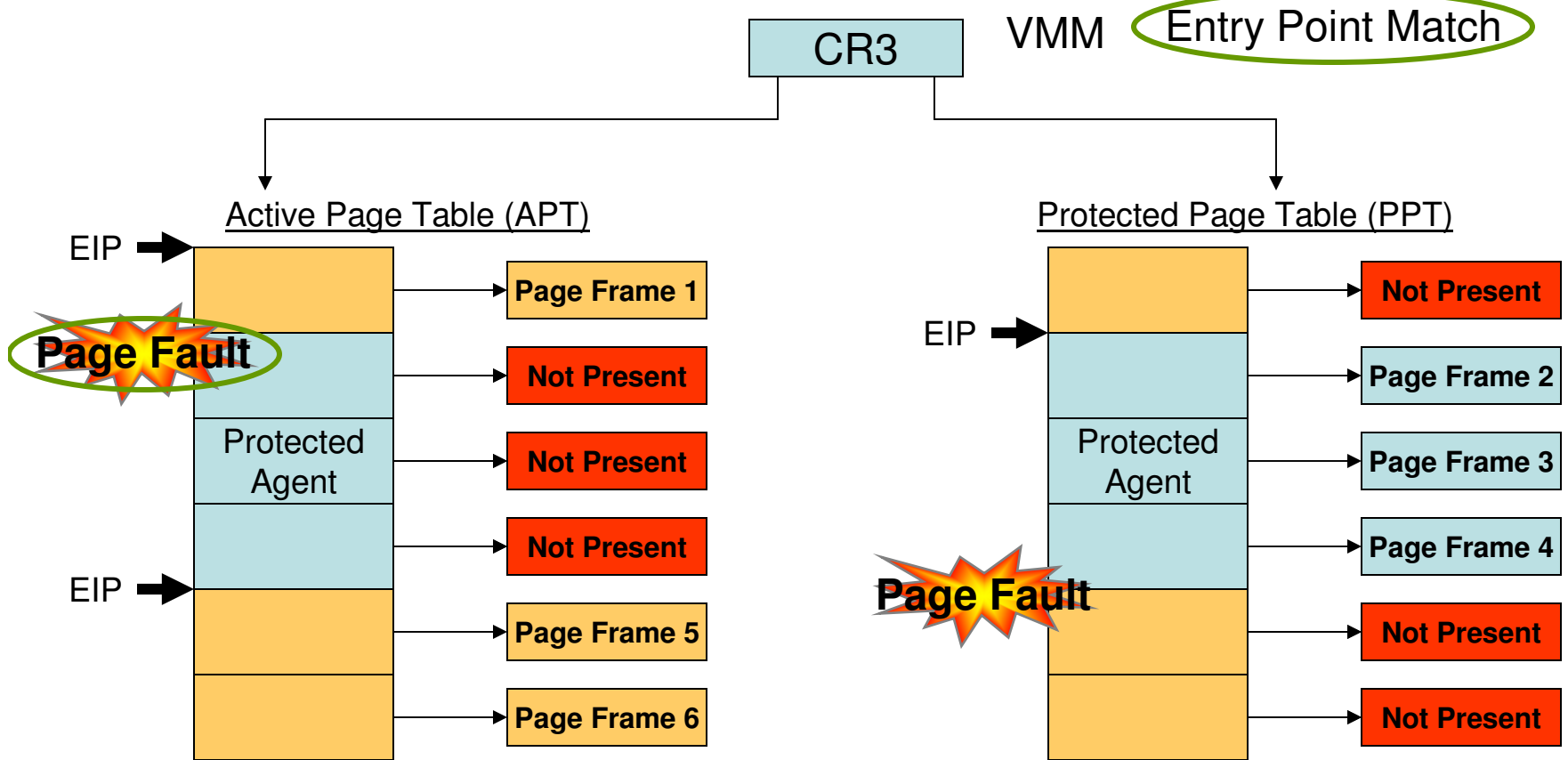
- Verifies integrity manifest
- Reverses relocation fix-ups using manifest information
- Performs hash verification of program image

- Cryptographically signed manifest
- Contains cryptographic hashes or programs in-memory contents
- Relocation Fix-up information

- Circumvention
- Eavesdropping
- Modifications
- Disabling

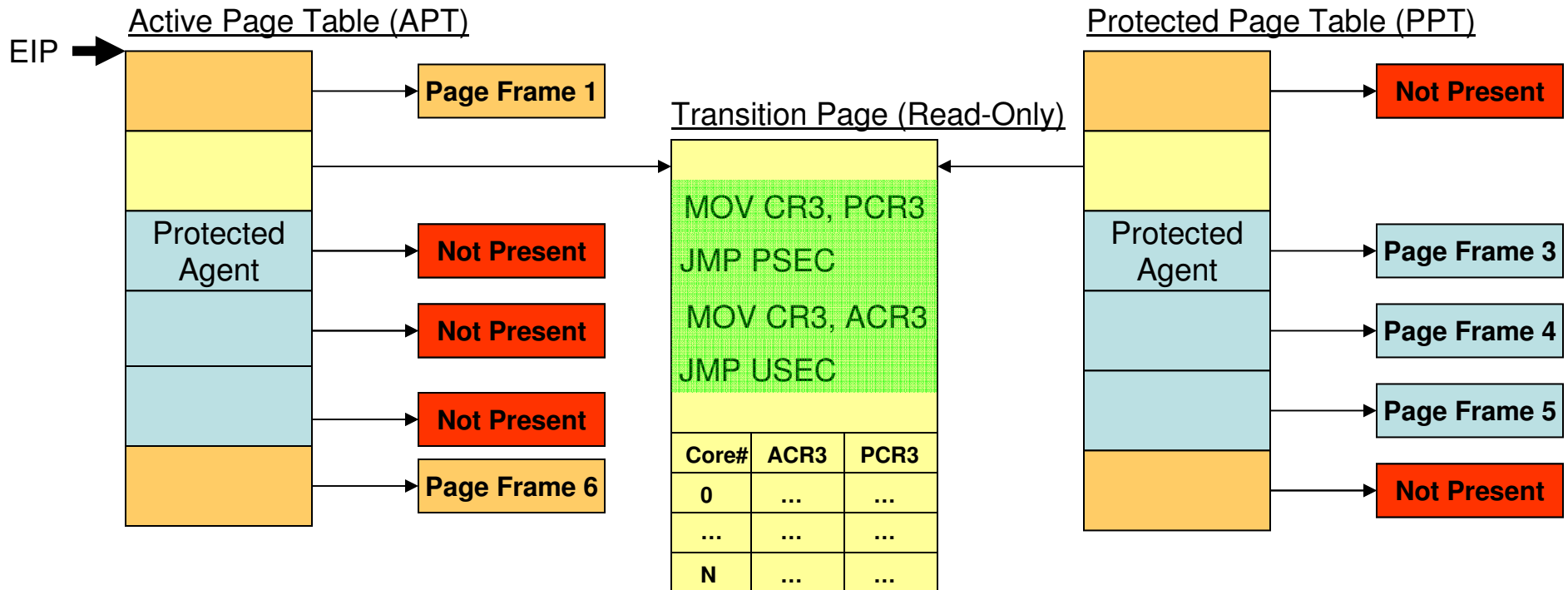
- Populates a protected page table (PPT) with measured and verified program pages corresponding to programs linear address space and updates page attributes in active page table (APT)
 - Private Code: APT (R--)/PPT(R-X)
 - Private Data : APT (---) /PPT(RW-)
 - Shared Code: APT(RWX)/PPT(RWX)
 - Shared Data : APT(RW-)/PPT(RW-)
- Enforces memory protection as guest page tables (GPT) are shadowed as APT
- Illegal accesses trap to hypervisor and handled as per policy. Notifications to policy manager
- Current implementation supports non-paged programs. Future work ongoing to support paged programs.

Memory Protection - Theory of operation



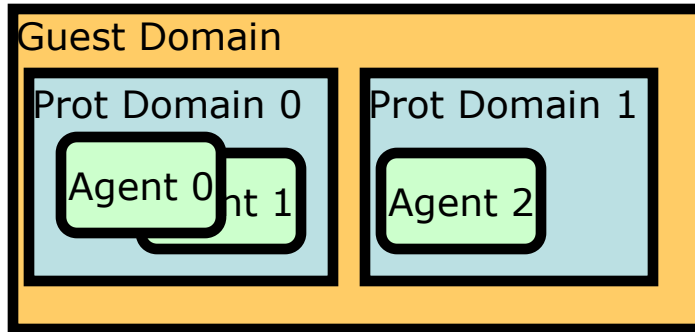
- Default scheme introduces excessive performance overheads due to page faults on each entry and exit to/from the PPT
- Optimized scheme using CR3 Target list (CR3TL)
 - List programmed with validated CR3 entries by VMM; no exit from guest on CR3 switch to values programmed in CR3TL
 - Feature on the VT enabled processors

Memory Protection – Acceleration using CR3 Target List

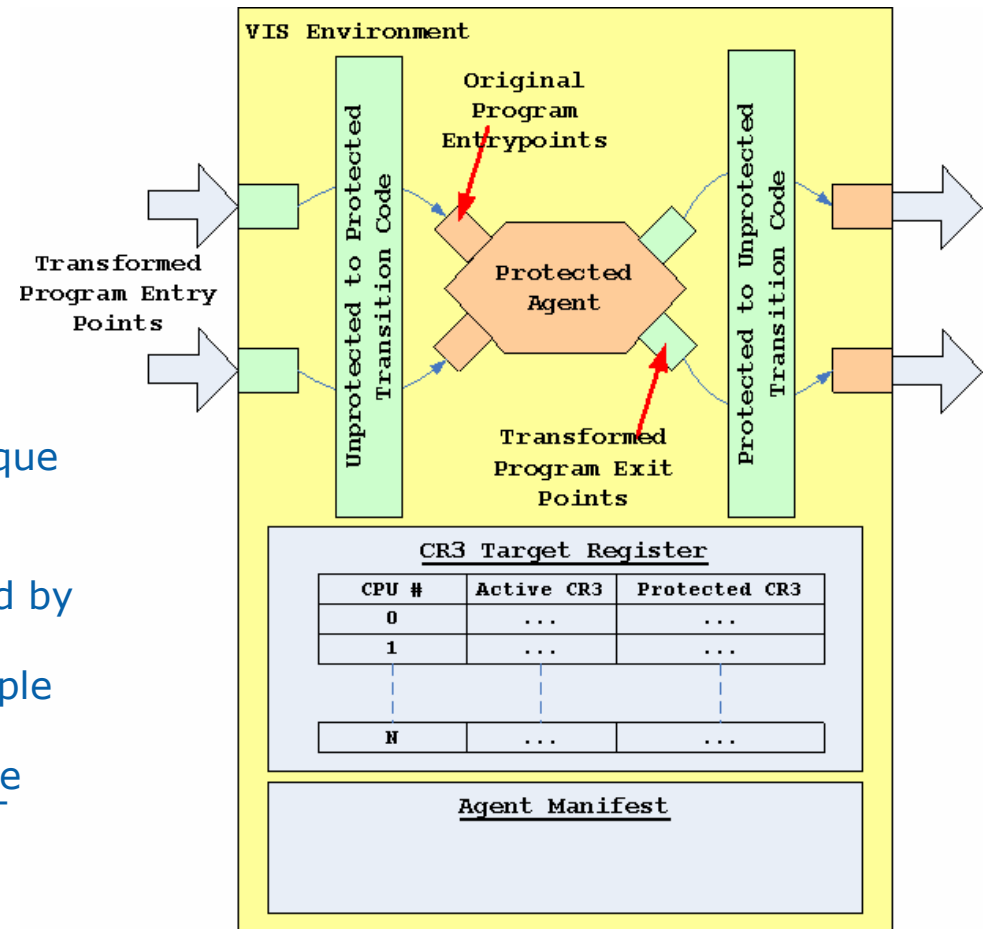


- Transition page used to enforce entry/exit from the protected agent
 - Read-only page enforced by the VMM
- On each CR3 switch, VMM programs the CR3 values for APT and PPT into a read-only table shared with protected agent
- VMM programs the CR3 values of APT and PPT into CR3 Target List

Agent Structure



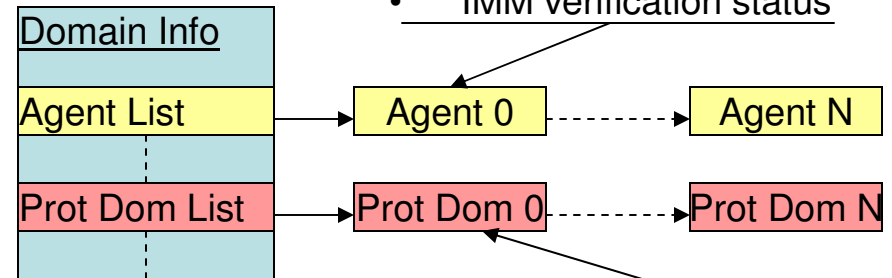
- Each protected agent identified using a unique agent UUID
- Each agent in a protection domain identified by a domain UUID
 - One protection domain can hosts multiple agents
 - Agents in a given protection domain are friends and share one PPT for each APT
- Pages belong physically to one agent
 - Pages go out of protection domain when that agent deregisters
 - Synchronization b/w friends is agents responsibility - expected behavior for co-operating agents



- Entry and Exit points into the program are controlled and enforced
 - Switch to PPT on entry
 - Switch to APT on exit

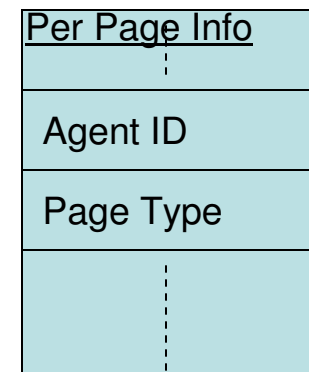
Protected Agent Registration

- **Agents register for protection using hypercall into VMM and provides below information for each agent section**
 - Agent UUID
 - Domain UUID
 - Linear address of section
 - Size of section
 - Attribute – Shared Code, Shared Data, Protected Code, Protected Data
- **VMM performs below actions on registration**
 - Update attributes of page in page database
 - Mark as protected and note the owner agent
 - Note type – Shared Code, Shared Data, Protected Code, Protected Data
 - Find all shadow mappings of page (in APT)
 - Update attributes in page table entries
 - Add agent information into registered agent list and mark as unverified
 - Notify registered IMM for verification
- **PPT for protection domains created only after IMM verification successful**

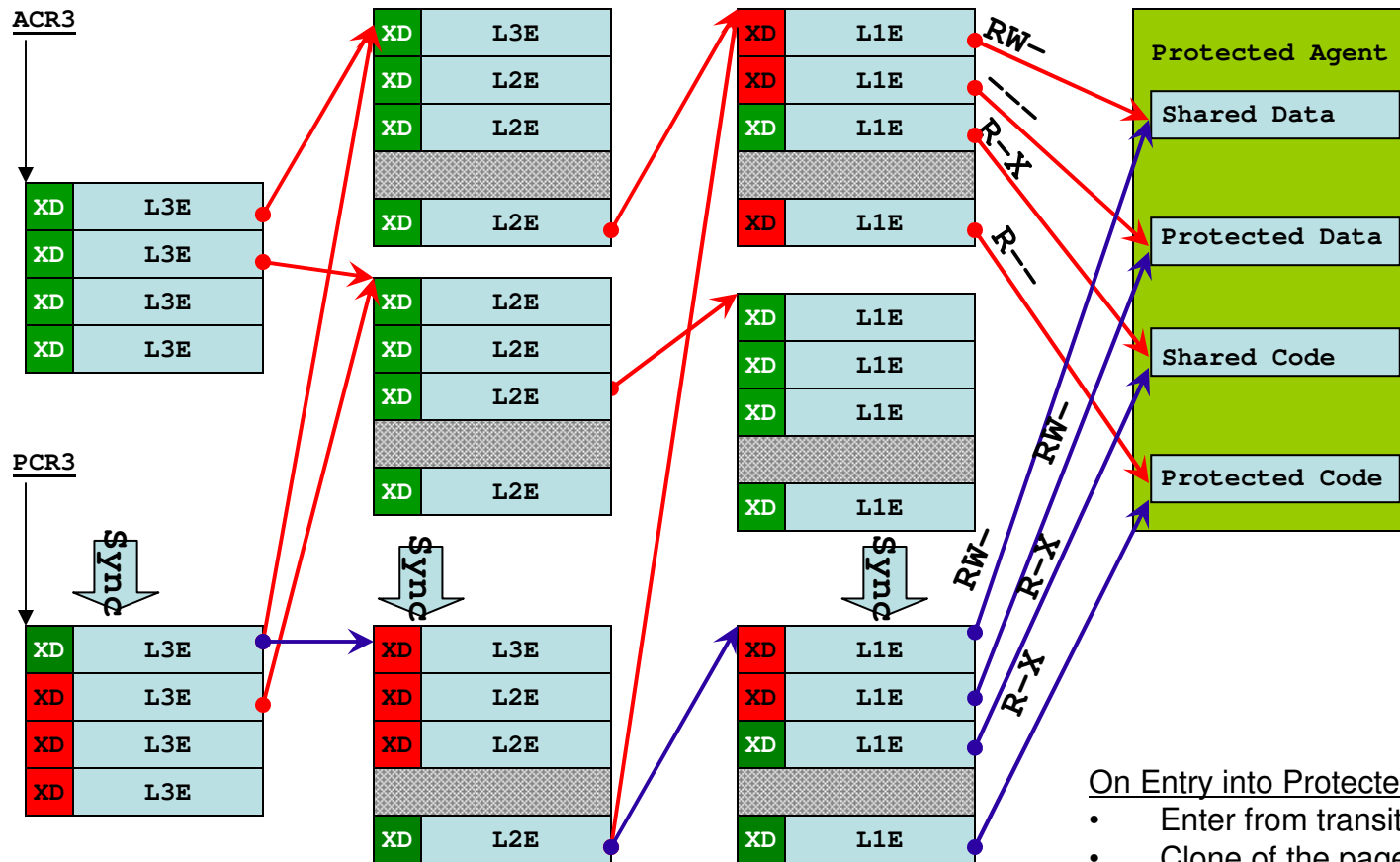


- Agent ID, Domain ID
- Protected section Info
- IMM verification status

- Domain ID
- Guest CR3, PPT CR3
- Protected shadow page database
- Synchronized to APT?
- LRU based ageing



Protected Page Table Construction



Registration

On registration the per page information for the agent pages updated and existing APT entries updated to satisfy agent protection

Synchronize APT/PPT

- Track paging events in APT and keep PPT updated
- Continually enforce protection for agent pages per registered attributes

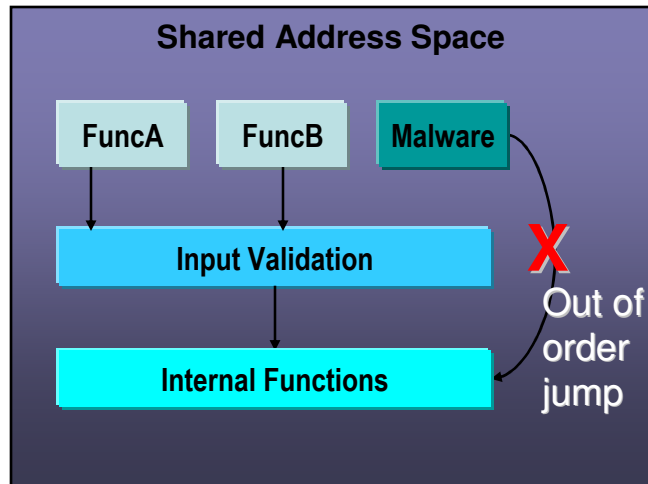
On Entry into Protected Domain

- Enter from transition page using hypercall
- Clone of the page table leading to agent pages
- Mark cloned entries as XD
- Open paths to the agent pages
 - only code pages marked as executable
- Add protected page table CR3 to CR3TL
- Add PPT to LRU list

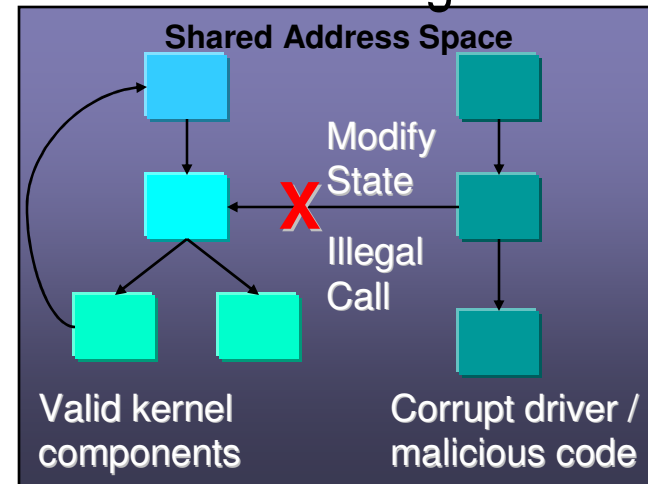


Threats Addressed

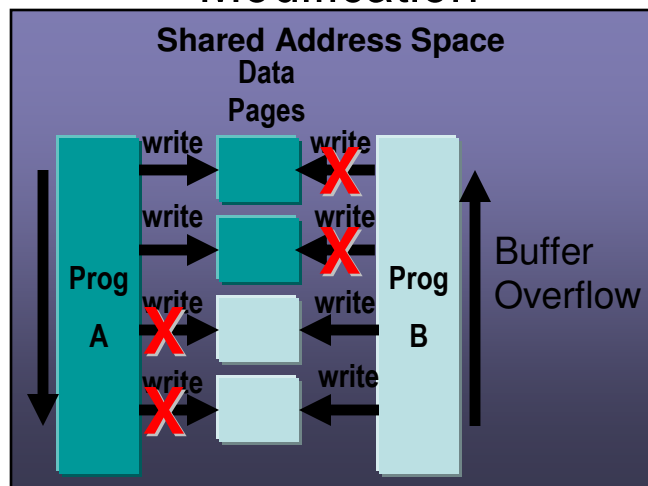
Circumvention



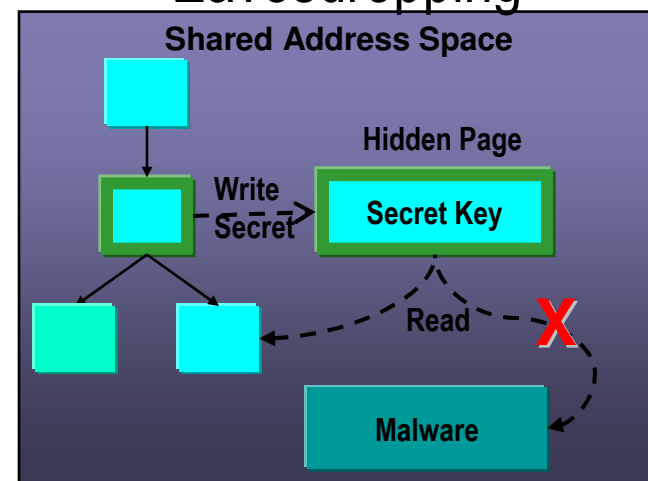
Disabling



Modification



Eavesdropping



Conclusions and Future Work

- Conclusions
 - Changes to Xen are modular and fit in well with current program structure – about 3K lines of code added
 - Penalty of CR3 switches avoided due to CR3 Target List (CR3TL) in Intel Processors
 - Transition into and out of protected page tables still incur penalties due to CR3 switching and TLB flushes
- Future Work
 - Support paged agents
 - Extend memory protection framework to use EPT

