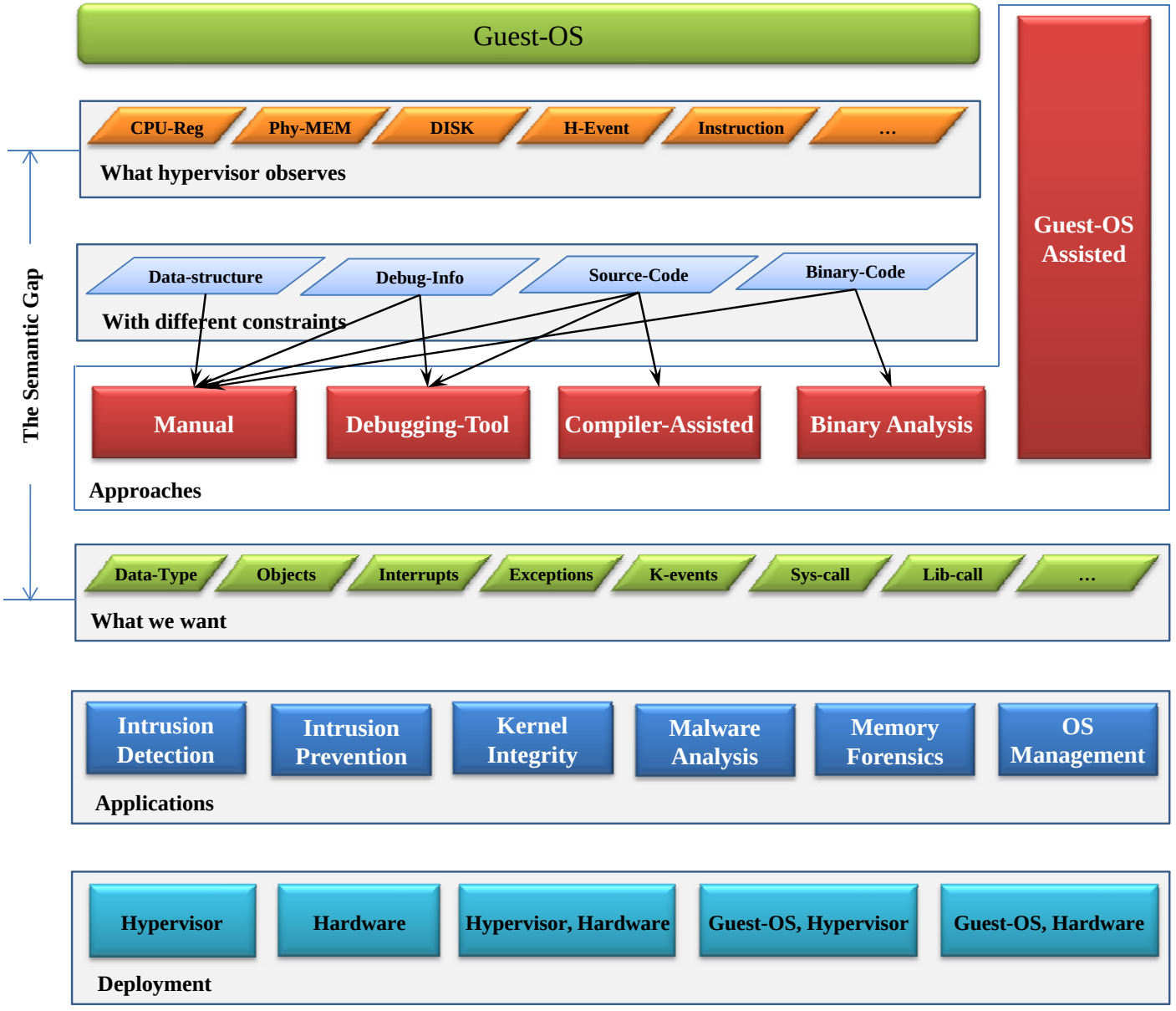
All You Ever Wanted to Know About Virtual Machine Introspection: Applications

Zhiqiang Lin

Department of Computer Sciences
The University of Texas at Dallas

August 24th, 2015

The Road Map


Guest-OS
CPU-Reg
Phy-MEM
DISK
H-Event
Instruction
…
What hypervisor observes
Data-structure
Debug-Info
Source-Code
Binary-Code
With different constraints
The Semantic Gap
Manual
Debugging-Tool
Compiler-Assisted
Binary Analysis
Approaches
Guest-OS Assisted
Data-Type
Objects
Interrupts
Exceptions
K-events
Sys-call
Lib-call
…
What we want
Intrusion Detection
Intrusion Prevention
Kernel Integrity
Malware Analysis
Memory Forensics
OS Management
Applications
Hypervisor
Hardware
Hypervisor, Hardware
Guest-OS, Hypervisor
Guest-OS, Hardware
Deployment

Outline

Outline

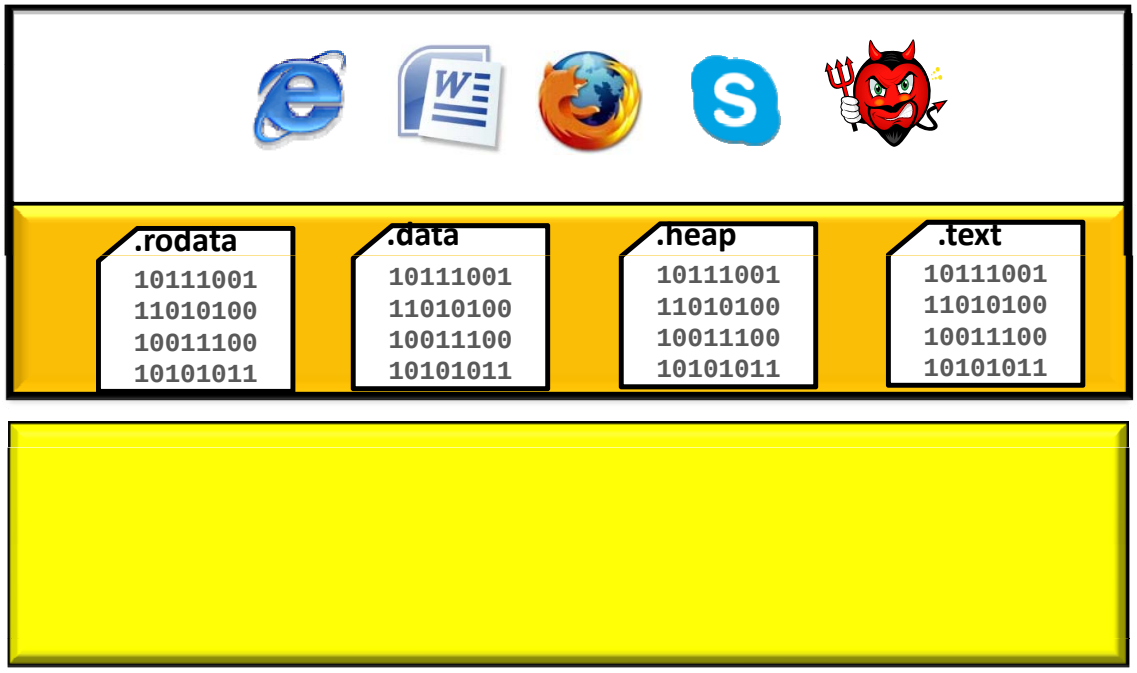
Security Applications

Security

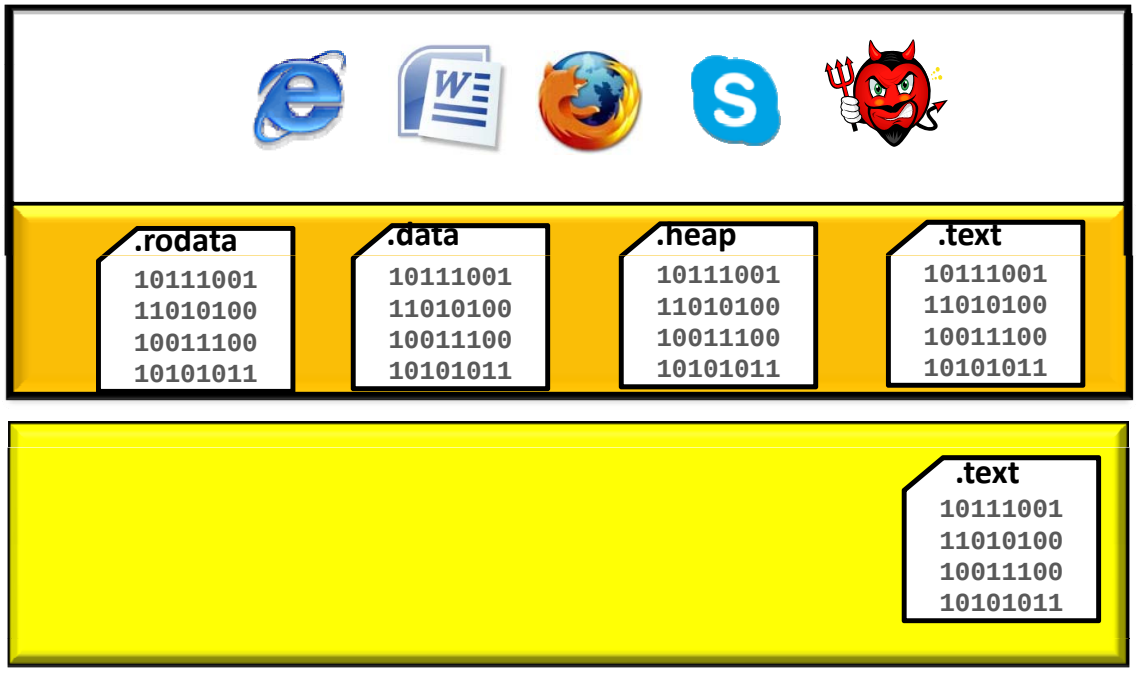
1. Detection
   - Kernel level
   - User level
   - Code modification detection
   - Data modification detection
   - Forensic analysis (including malware analysis)
2. Prevention
   - Kernel level
   - User level
   - Code protection
   - Data protection
3. Recovery

Outline

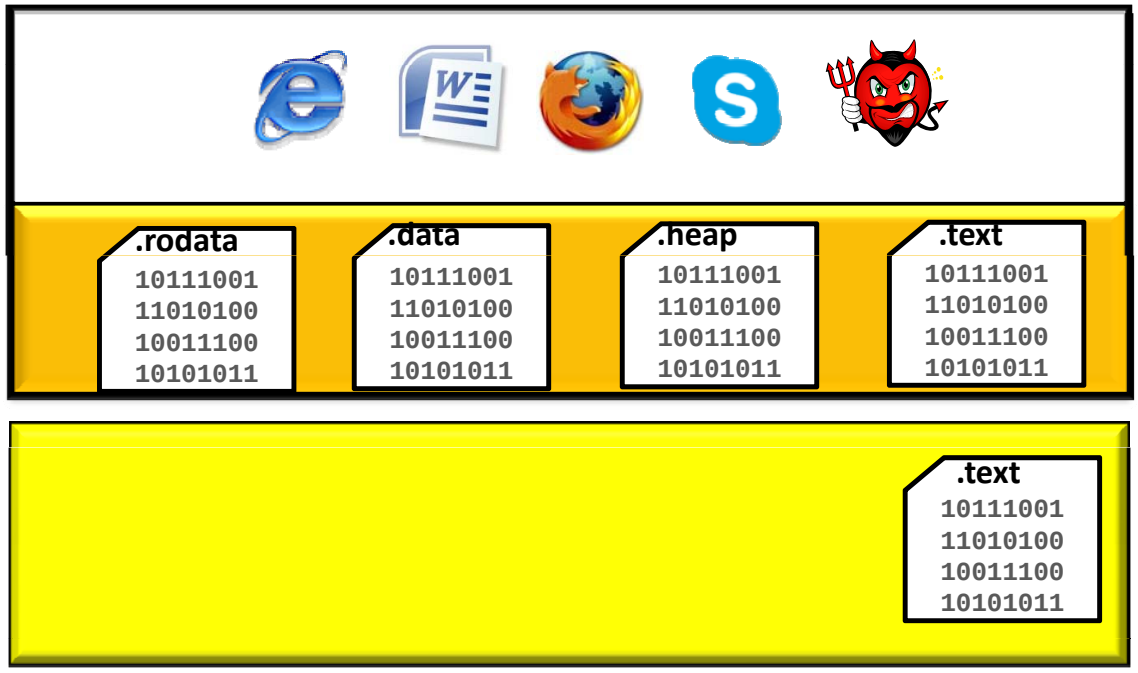
Detection: Kernel Code and Read-only Data



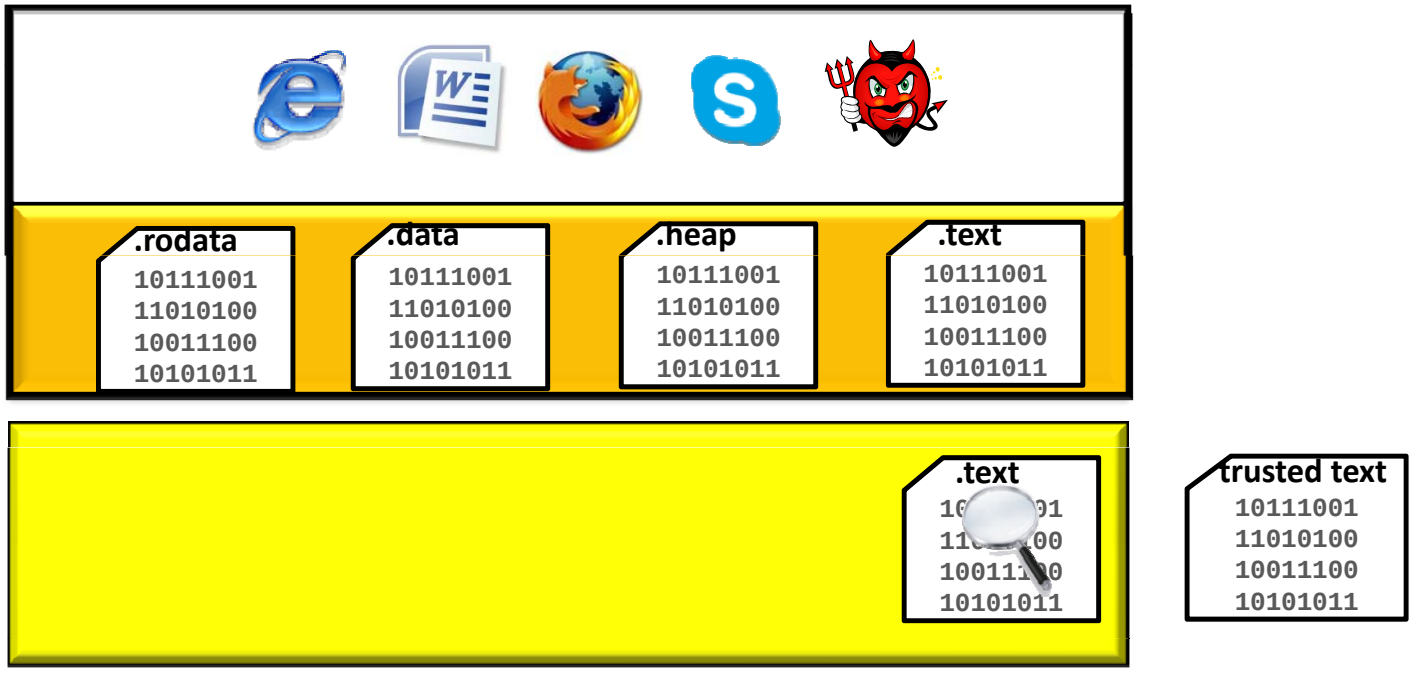
Detection: Kernel Code and Read-only Data



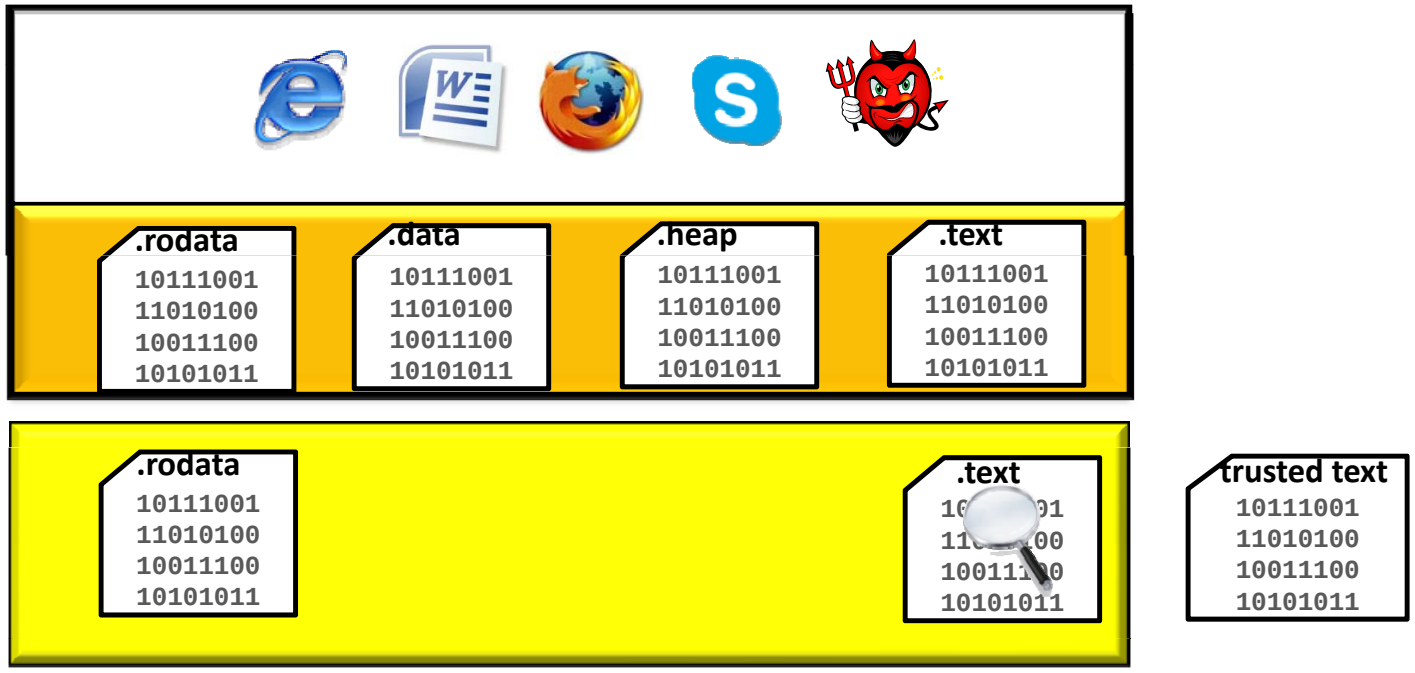
Detection: Kernel Code and Read-only Data



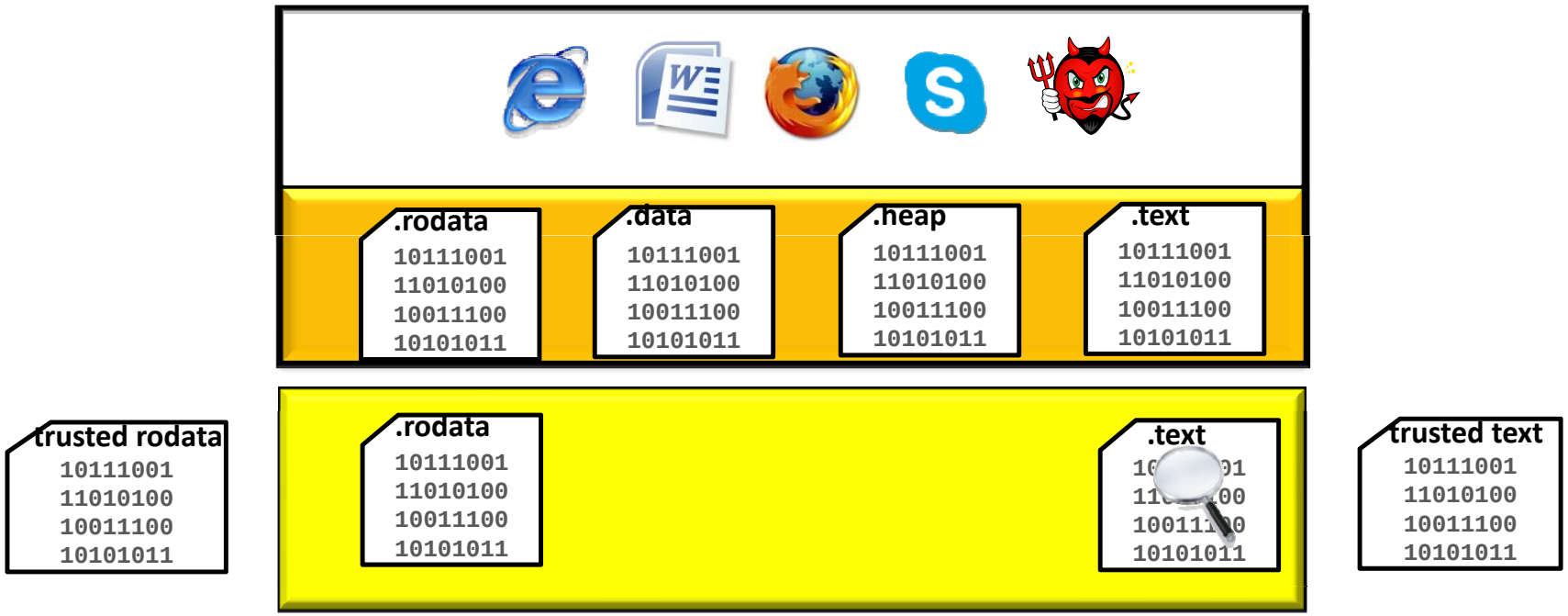
Detection: Kernel Code and Read-only Data



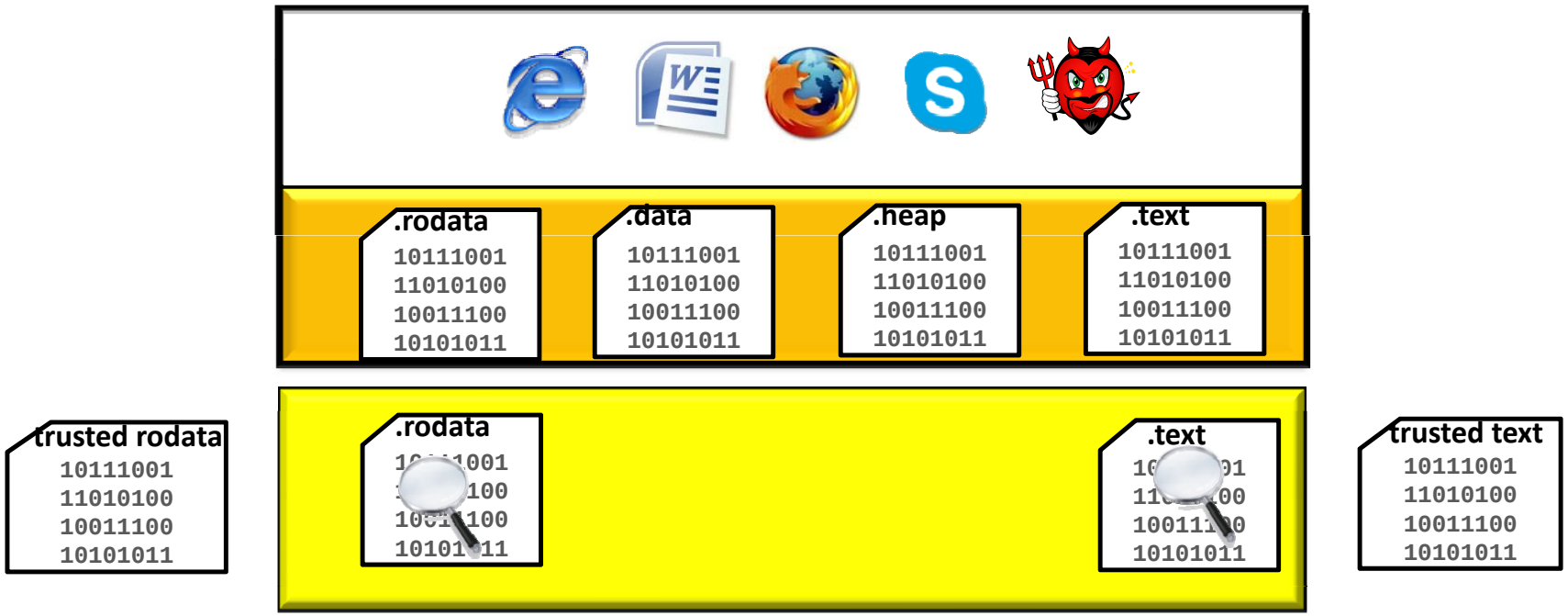
Detection: Kernel Code and Read-only Data



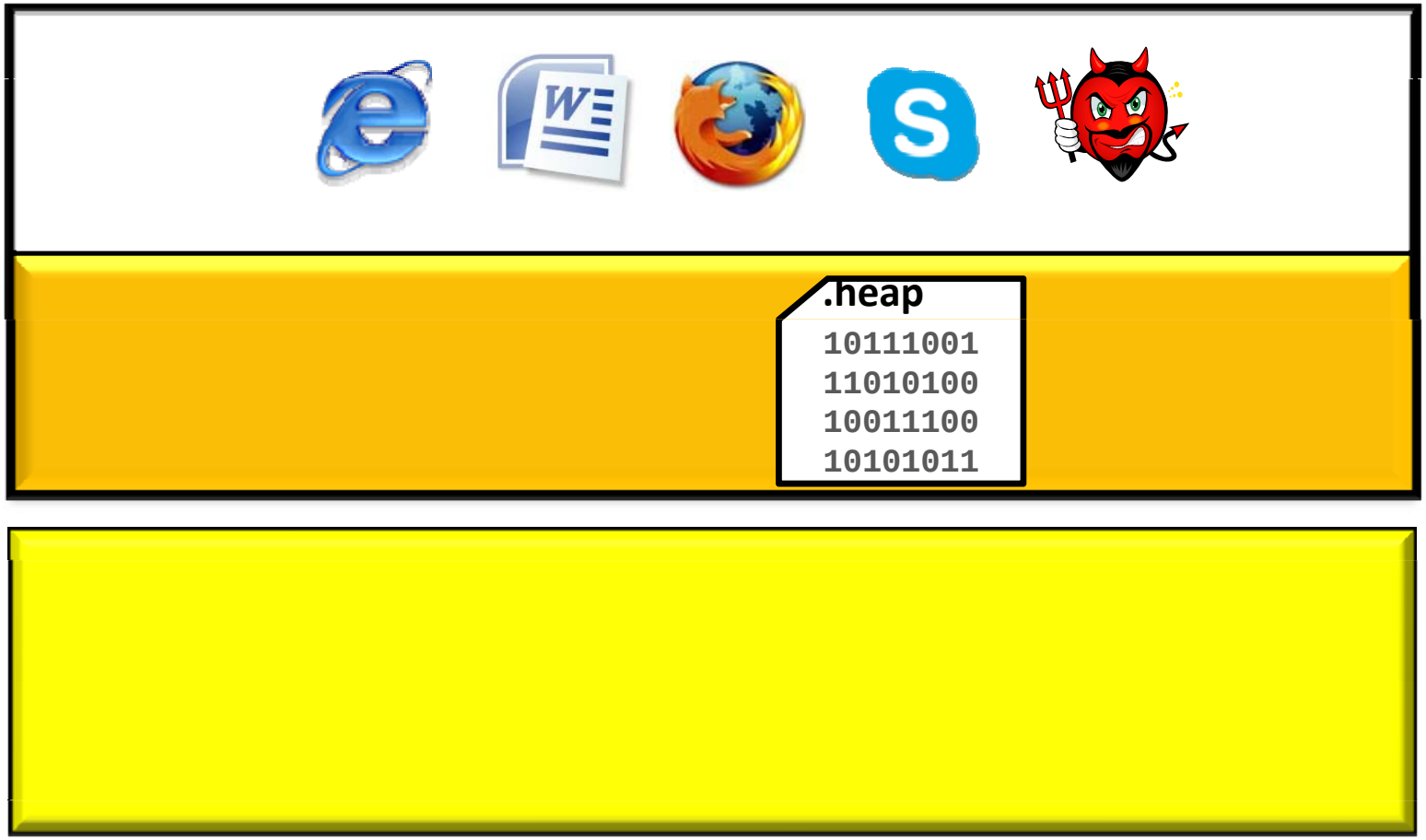
Detection: Kernel Code and Read-only Data



Detection: Kernel Code and Read-only Data

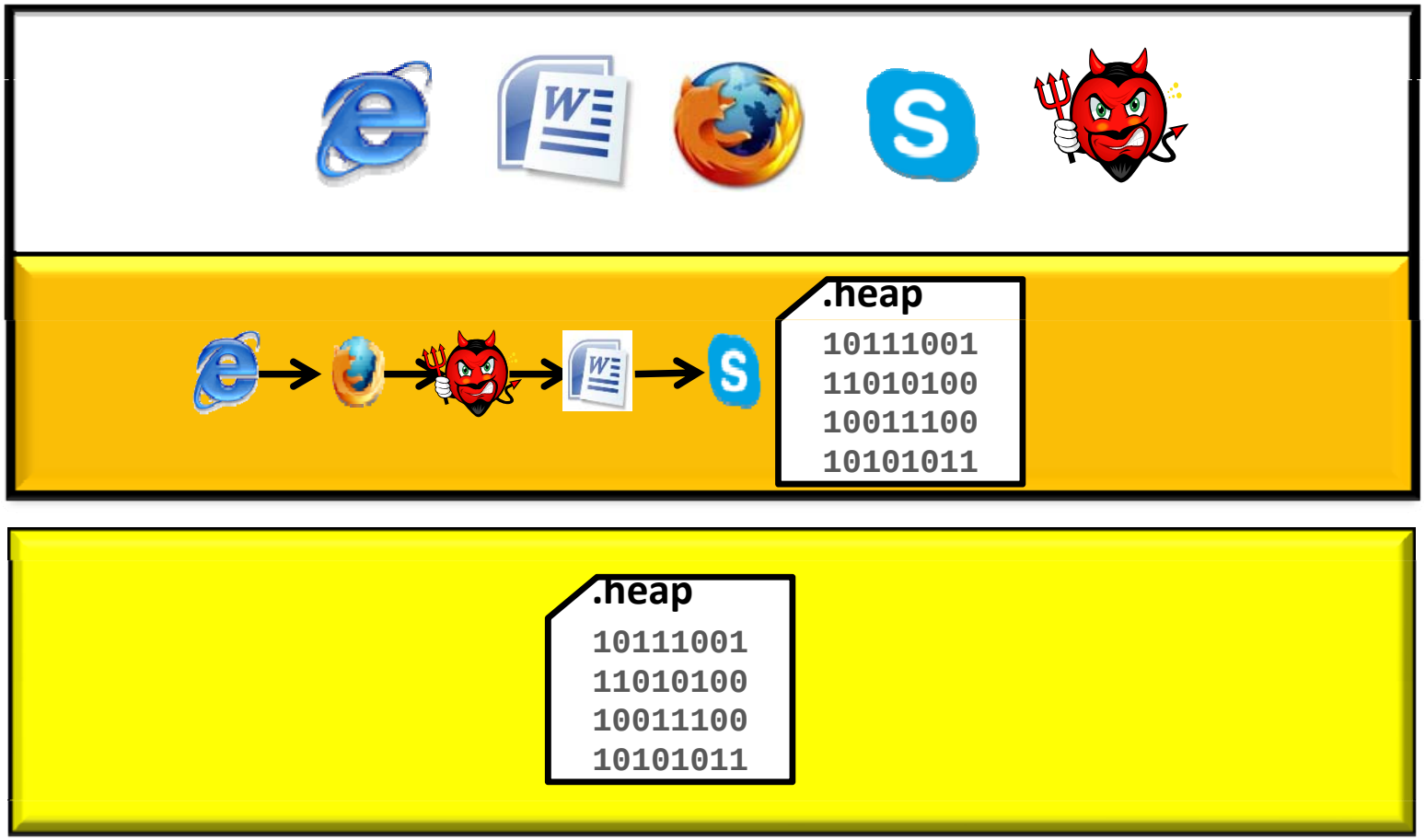


On Kernel Heap Data: Data Structure Traversal

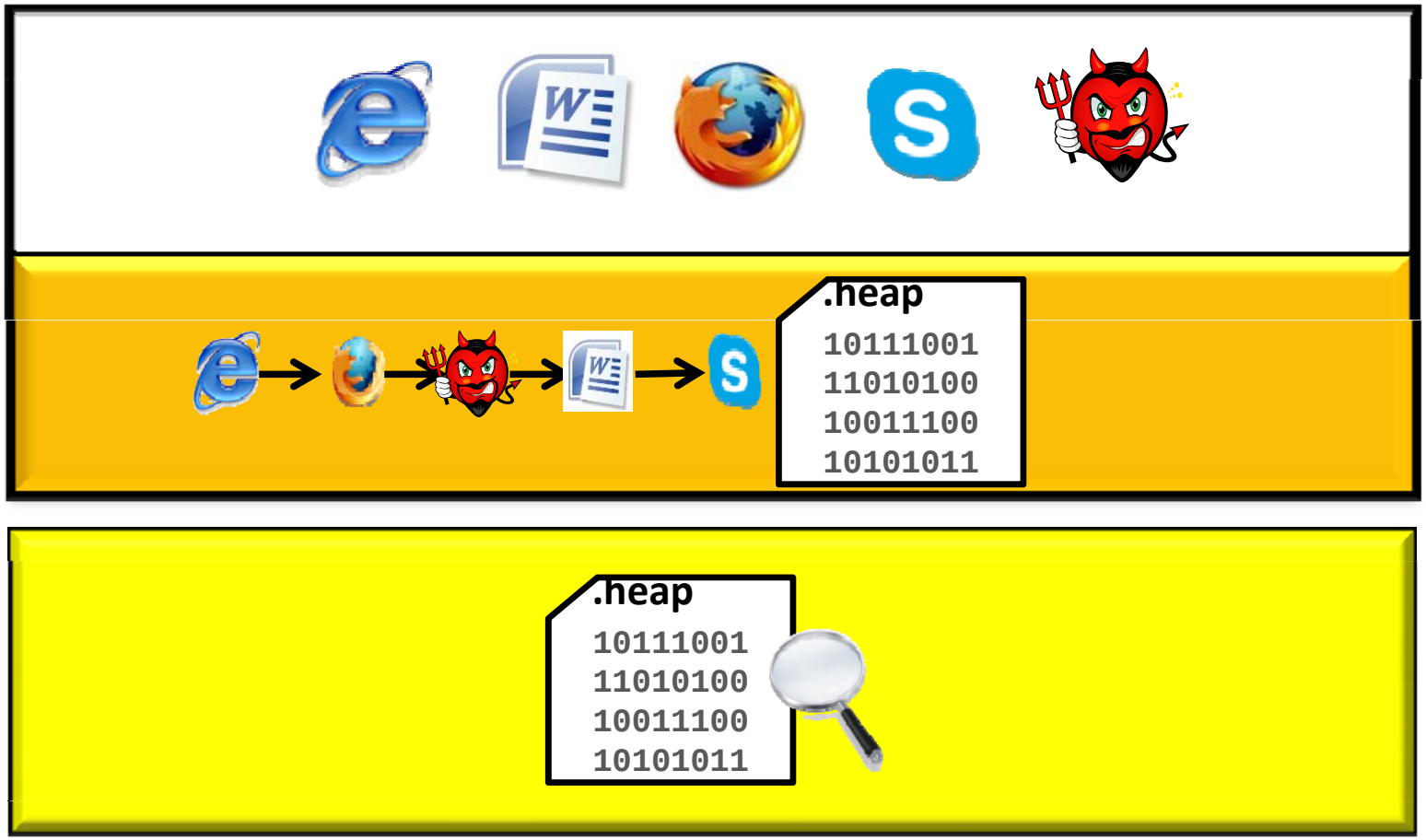


On Kernel Heap Data: Data Structure Traversal

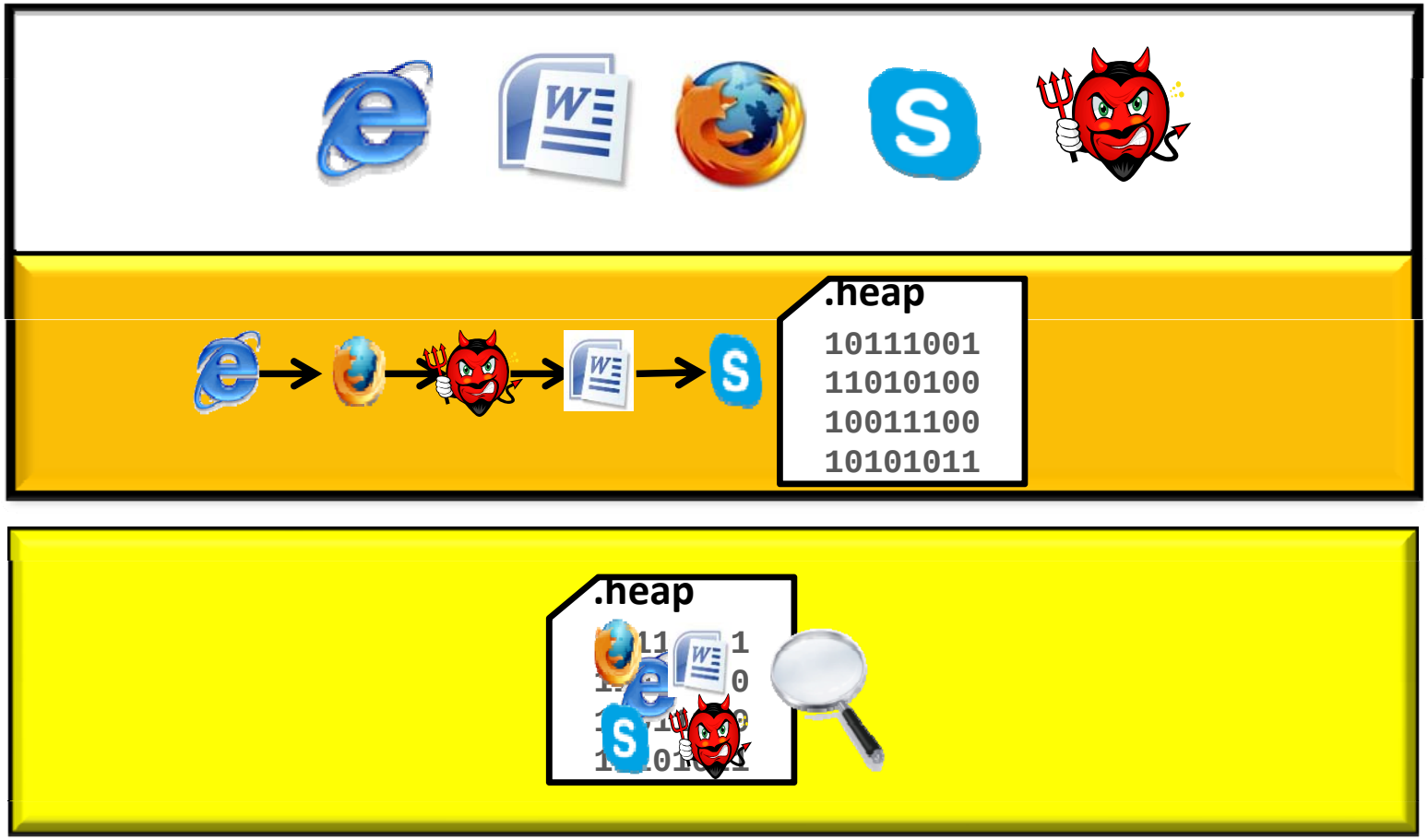
On Kernel Heap Data: Data Structure Traversal



On Kernel Heap Data: Data Structure Traversal

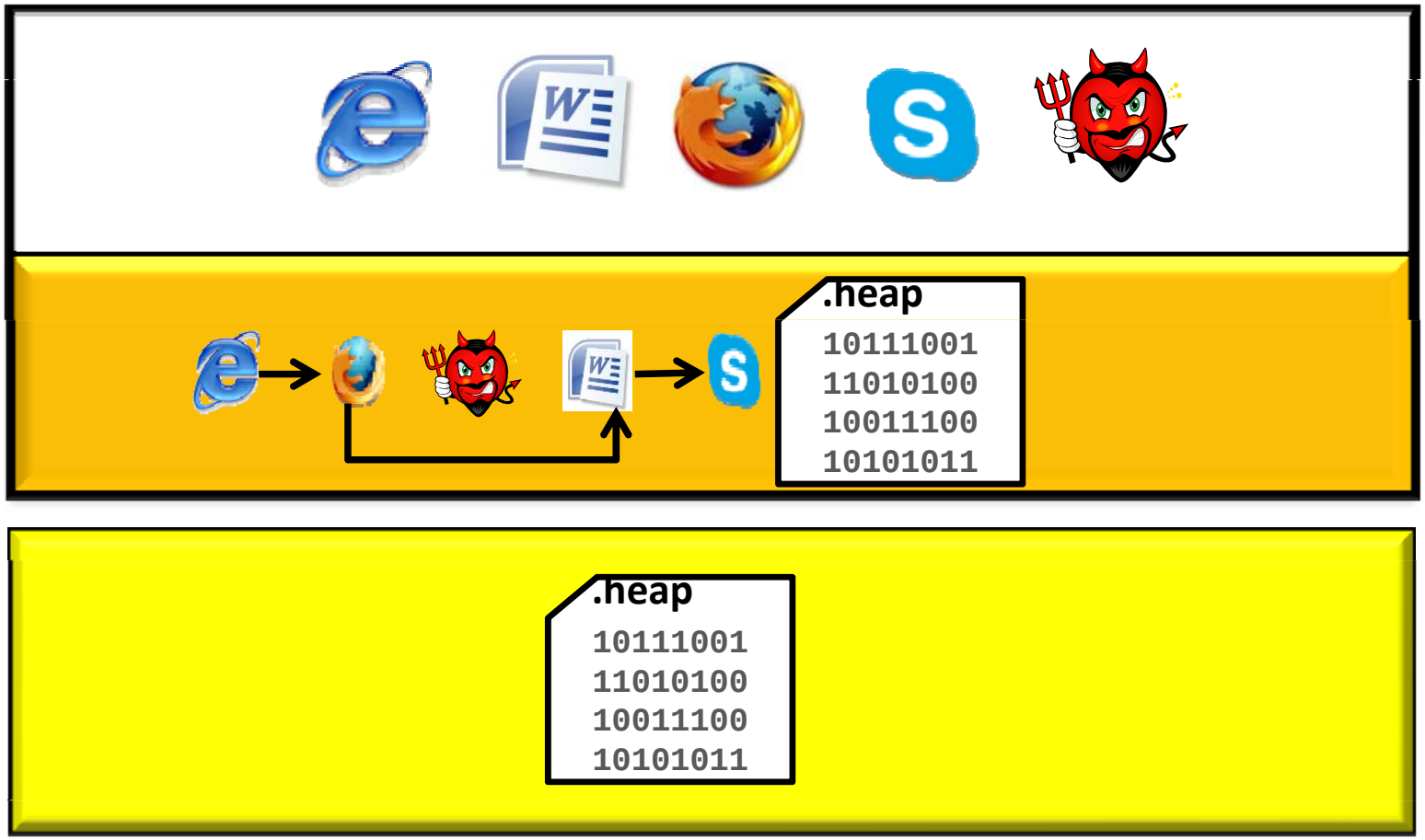


On Kernel Heap Data: Data Structure Traversal

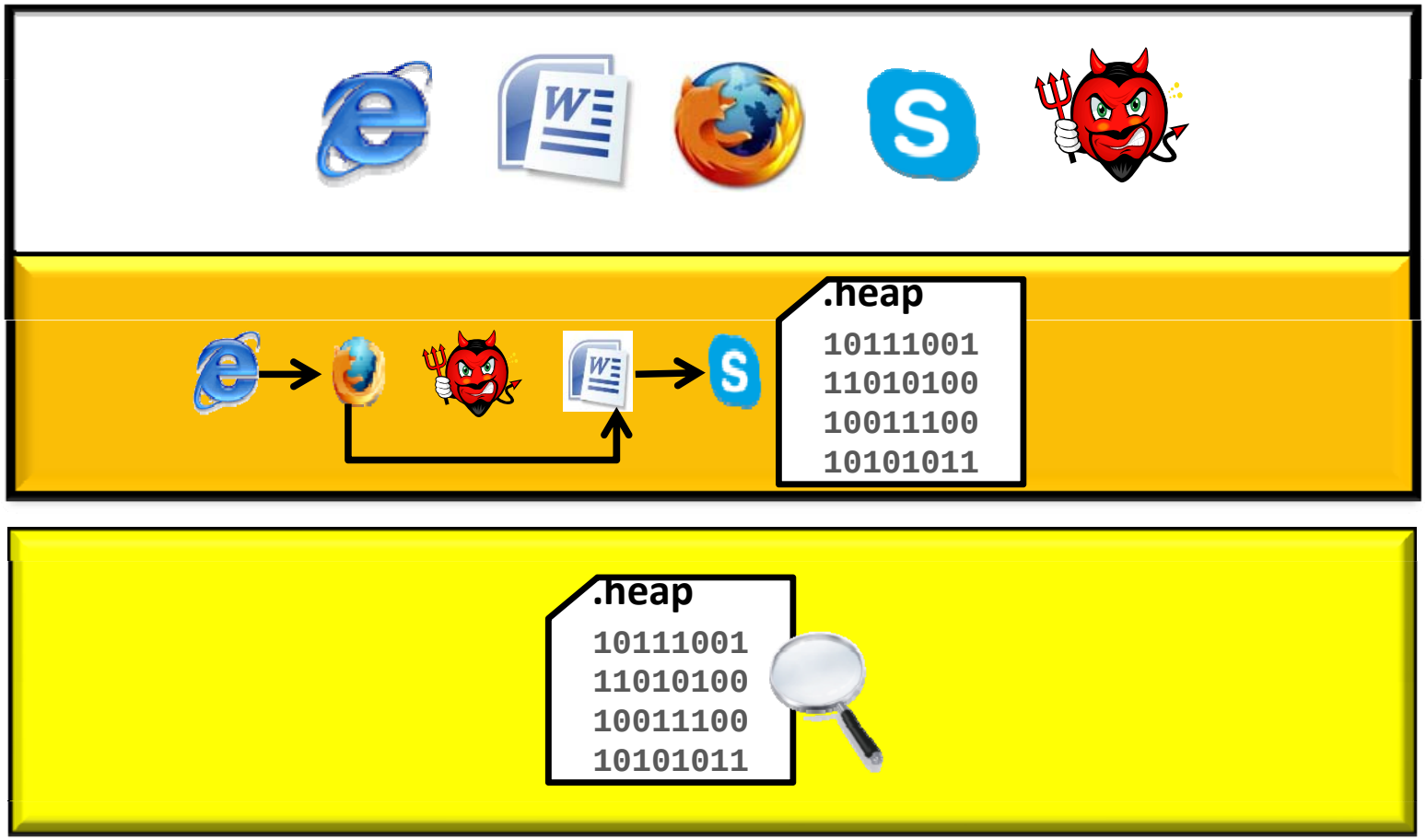


On Kernel Heap Data: Data Instance Scanning

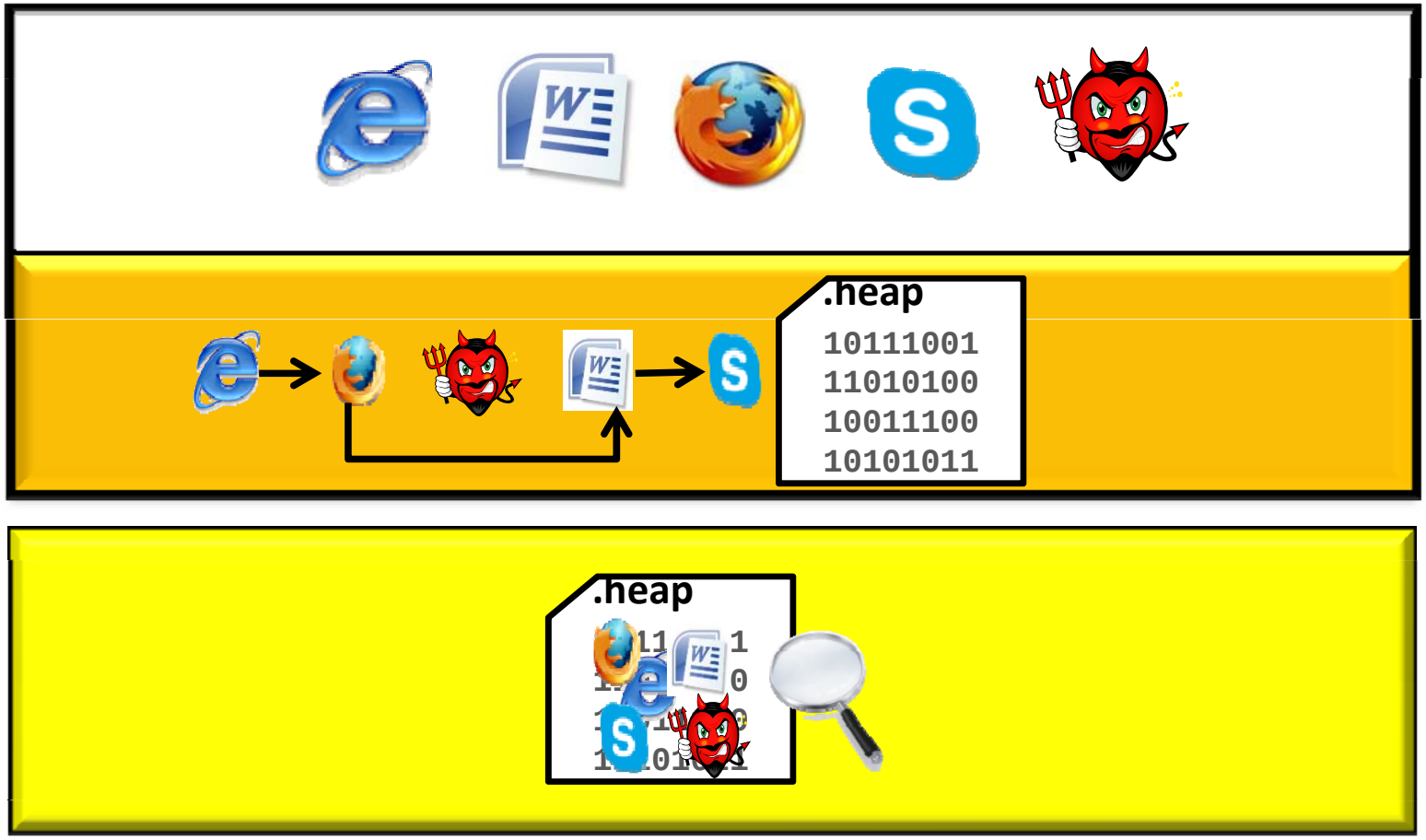
On Kernel Heap Data: Data Instance Scanning



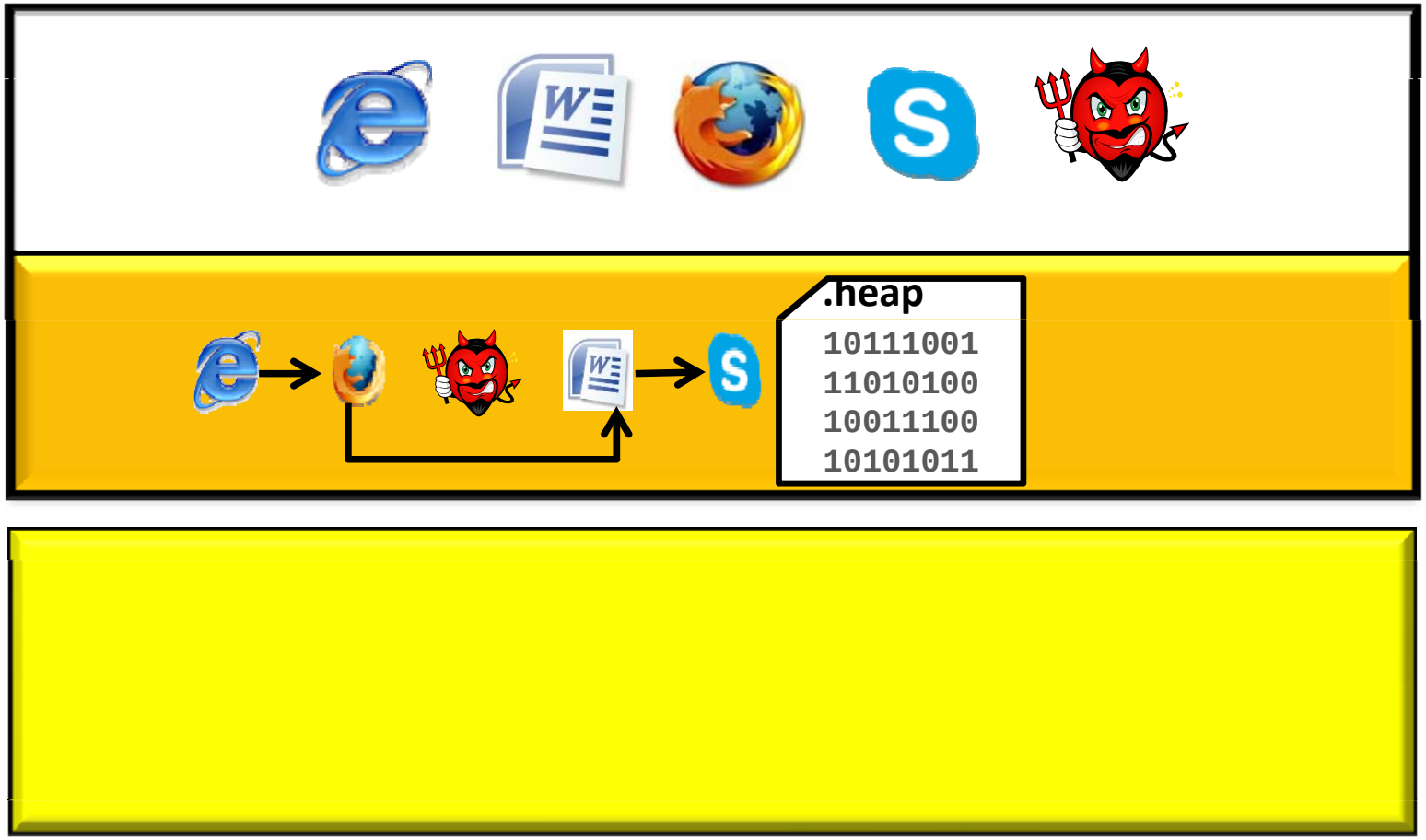
On Kernel Heap Data: Data Instance Scanning



On Kernel Heap Data: Data Instance Scanning

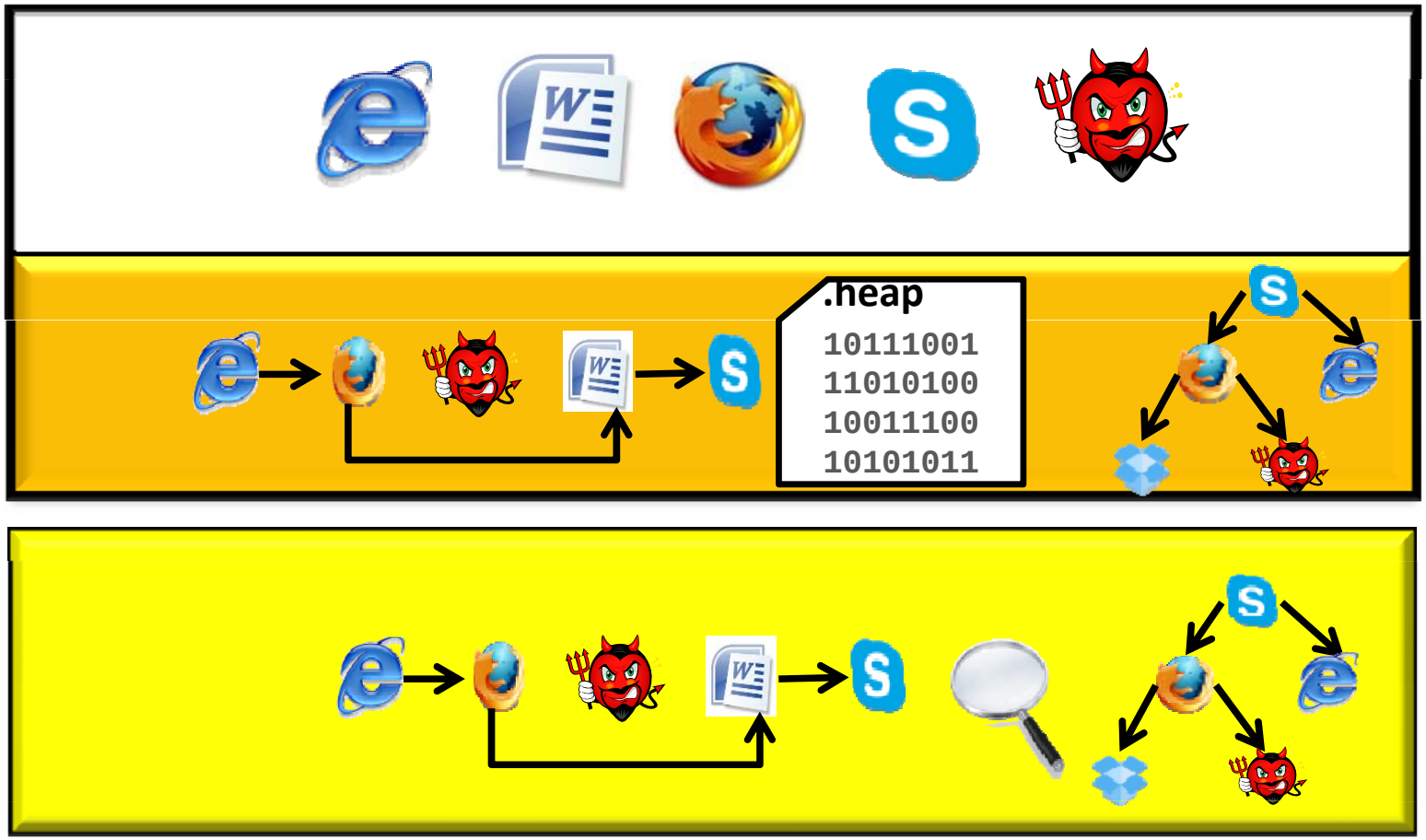


On Kernel Heap Data: Using Invariant

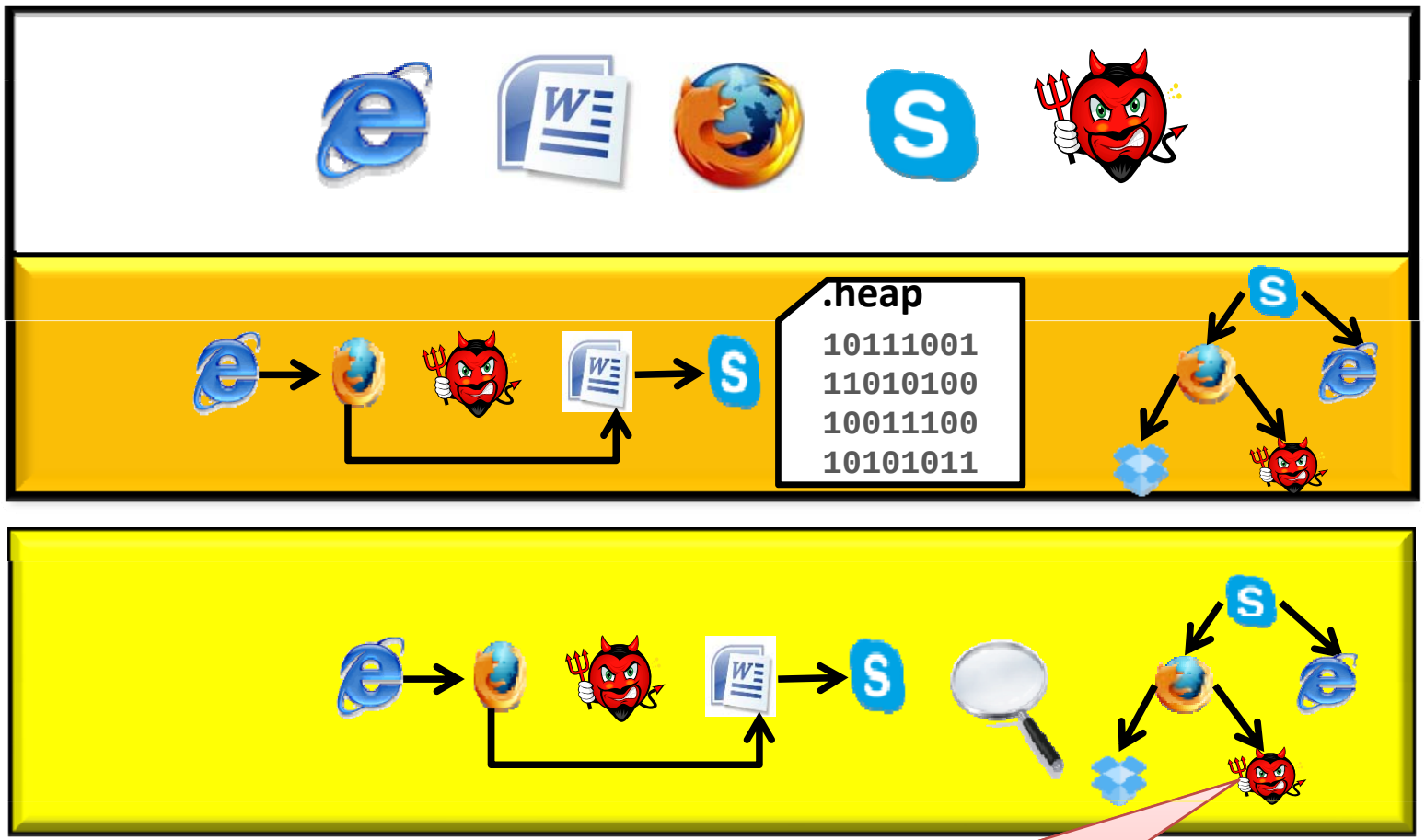


On Kernel Heap Data: Using Invariant

On Kernel Heap Data: Using Invariant

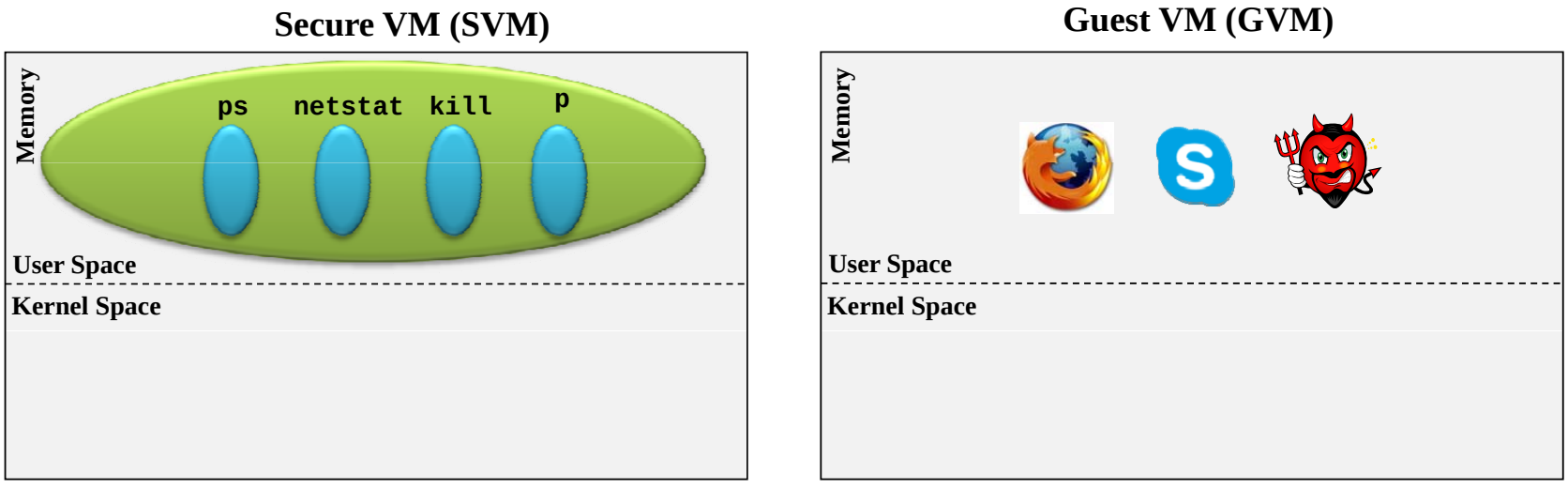


On Kernel Heap Data: Using Invariant

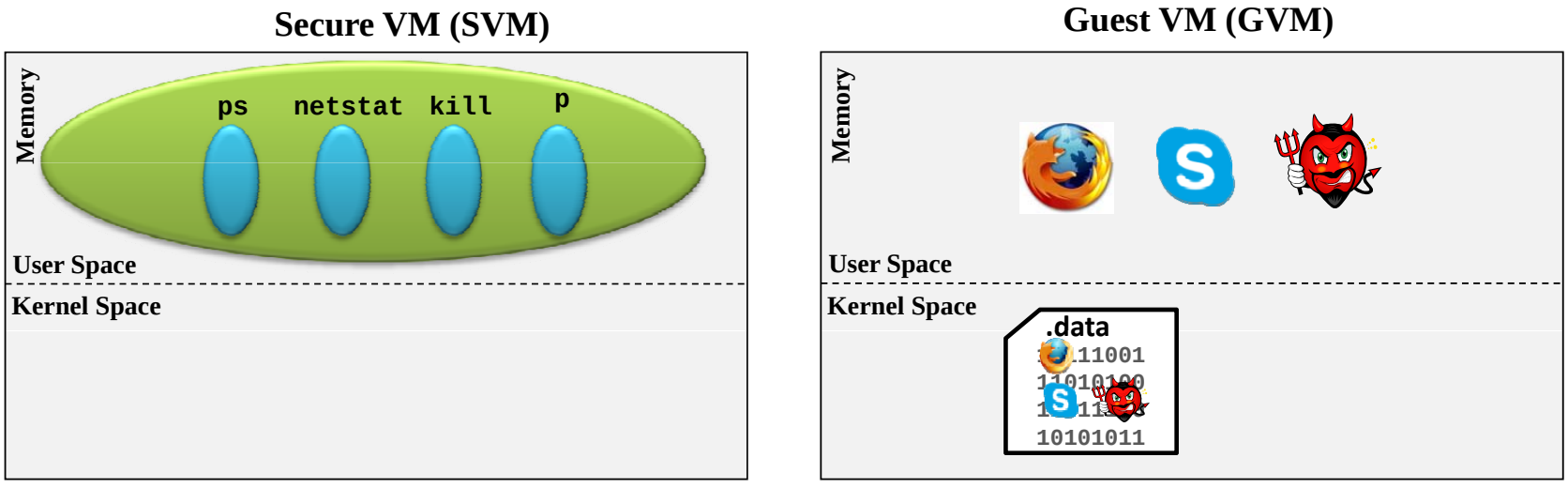


Invariant: Set of processes in Scheduler tree = Process list

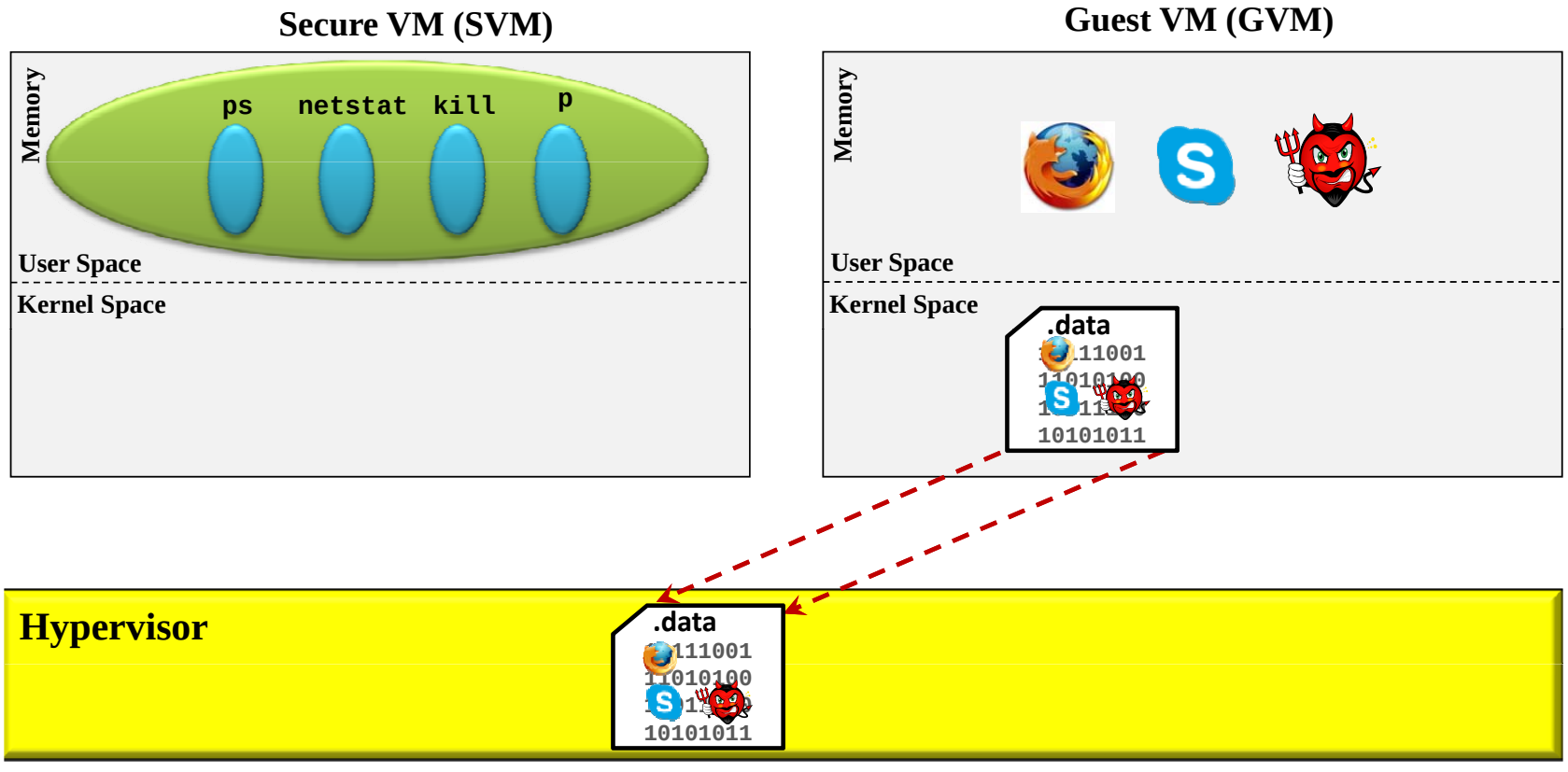
Using Data Out Grafting/Redirection [CCS'11,SP'12]



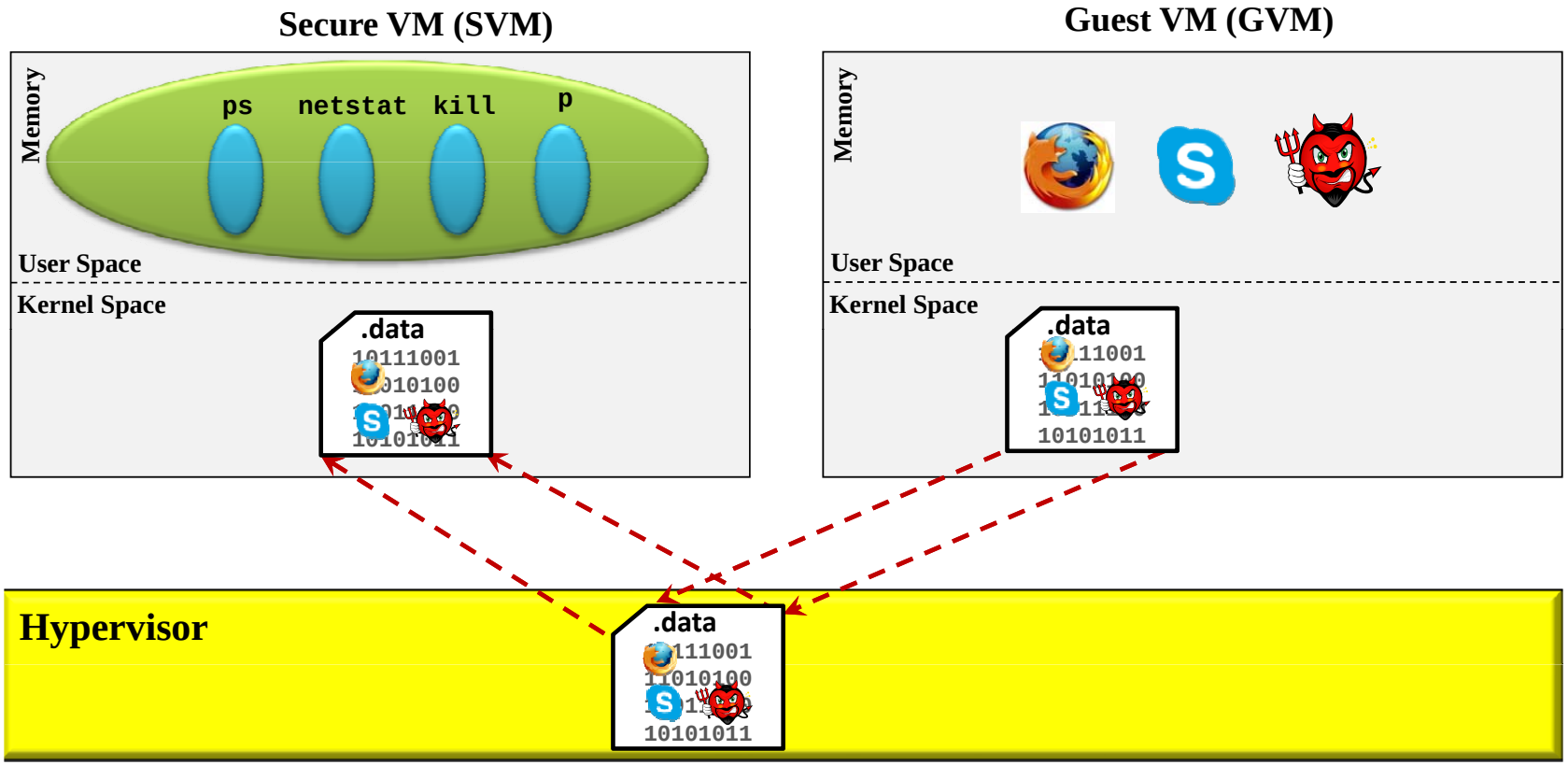
Using Data Out Grafting/Redirection [CCS'11,SP'12]



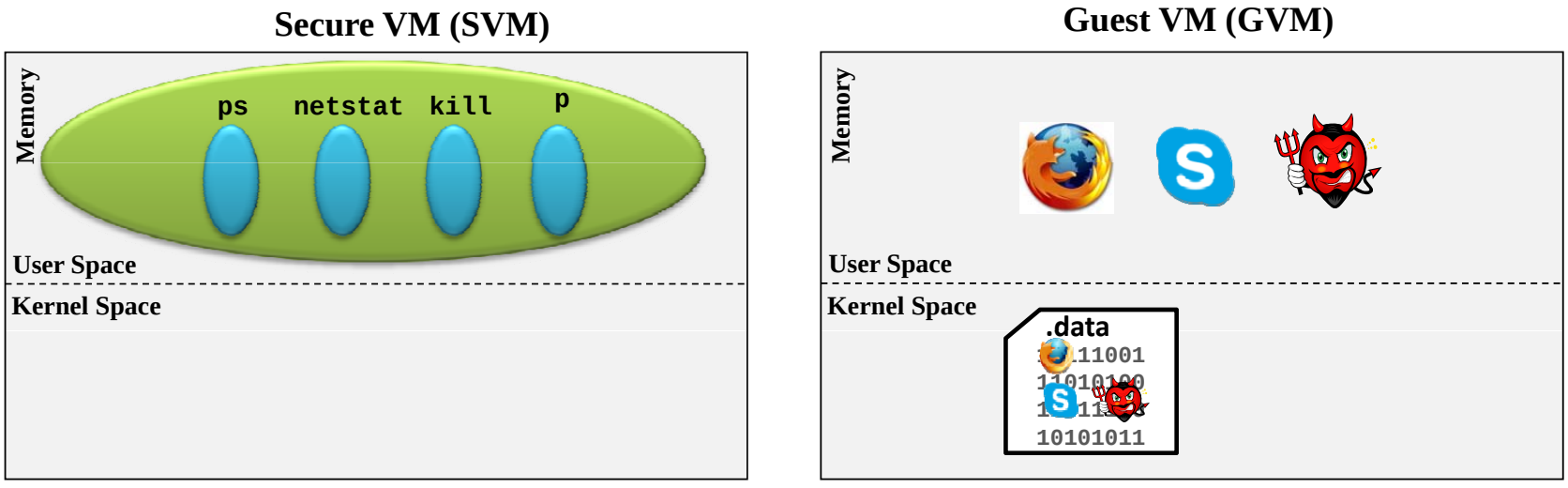
Using Data Out Grafting/Redirection [CCS'11,SP'12]



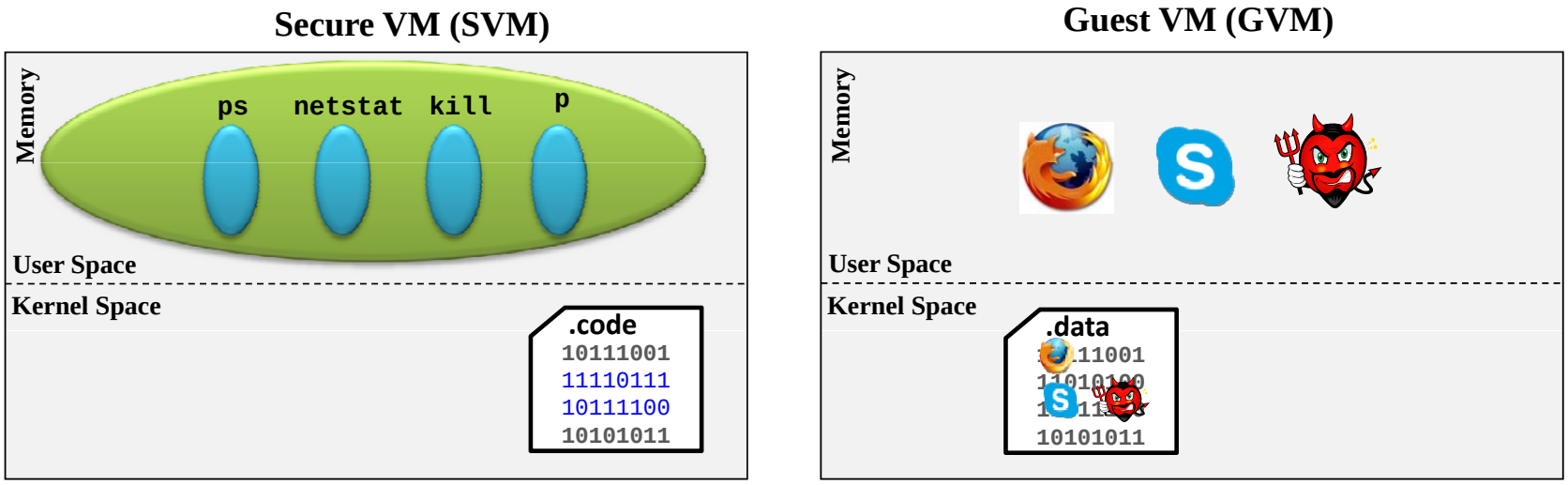
Using Data Out Grafting/Redirection [CCS'11,SP'12]



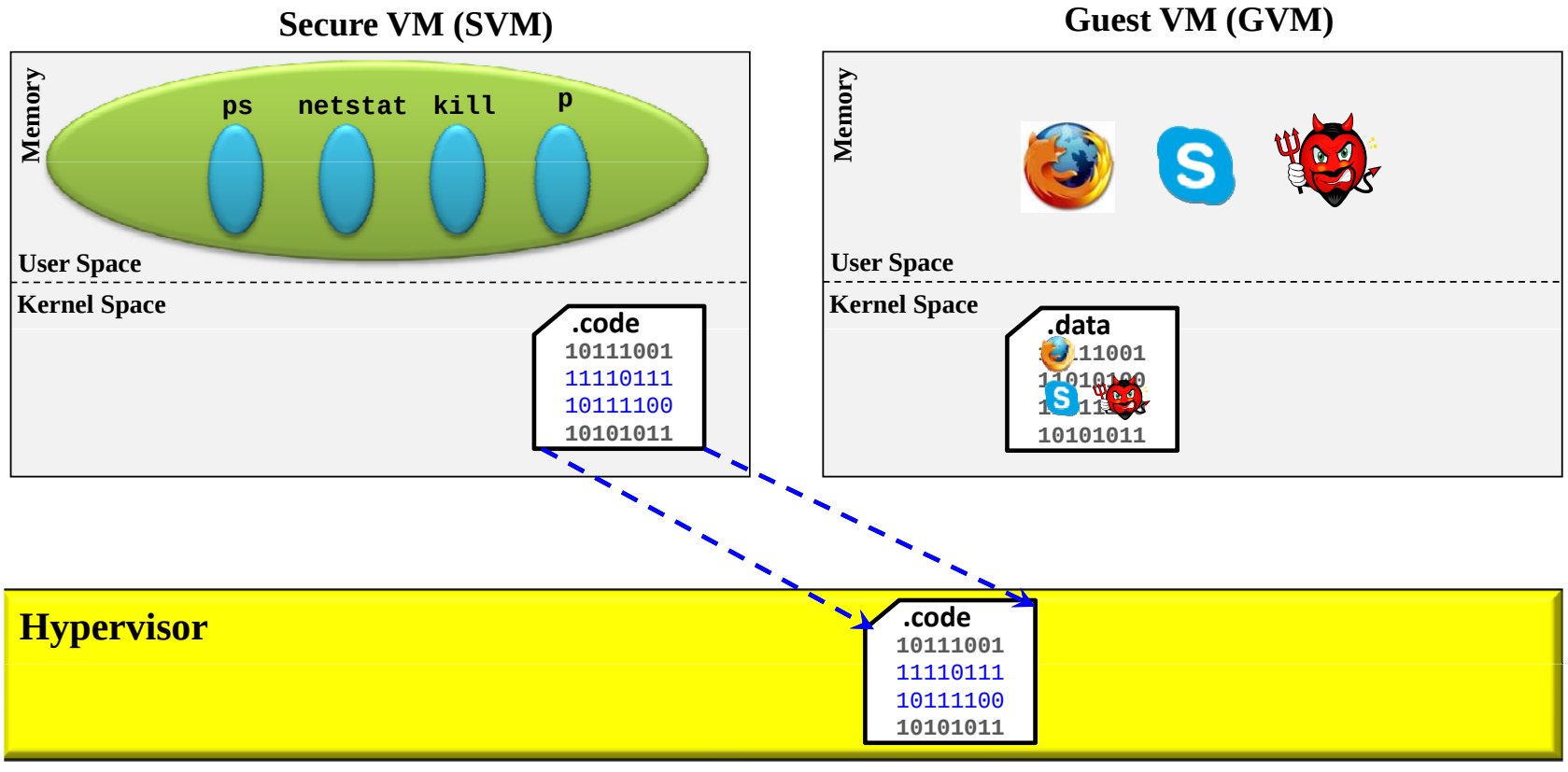
Using Code Implanting [DSN'10, ATC'14, DSN'14]



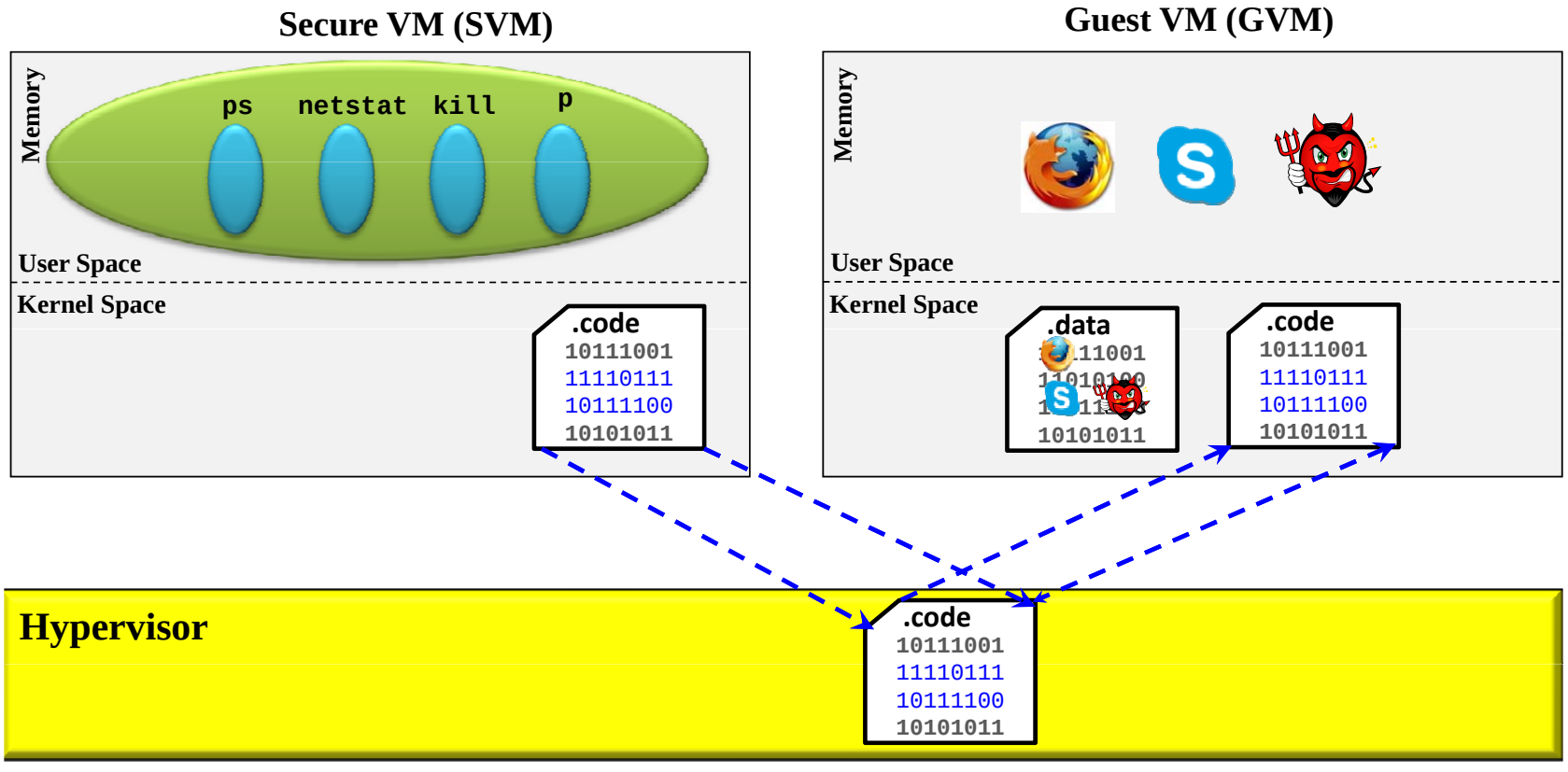
Using Code Implanting [DSN'10, ATC'14, DSN'14]



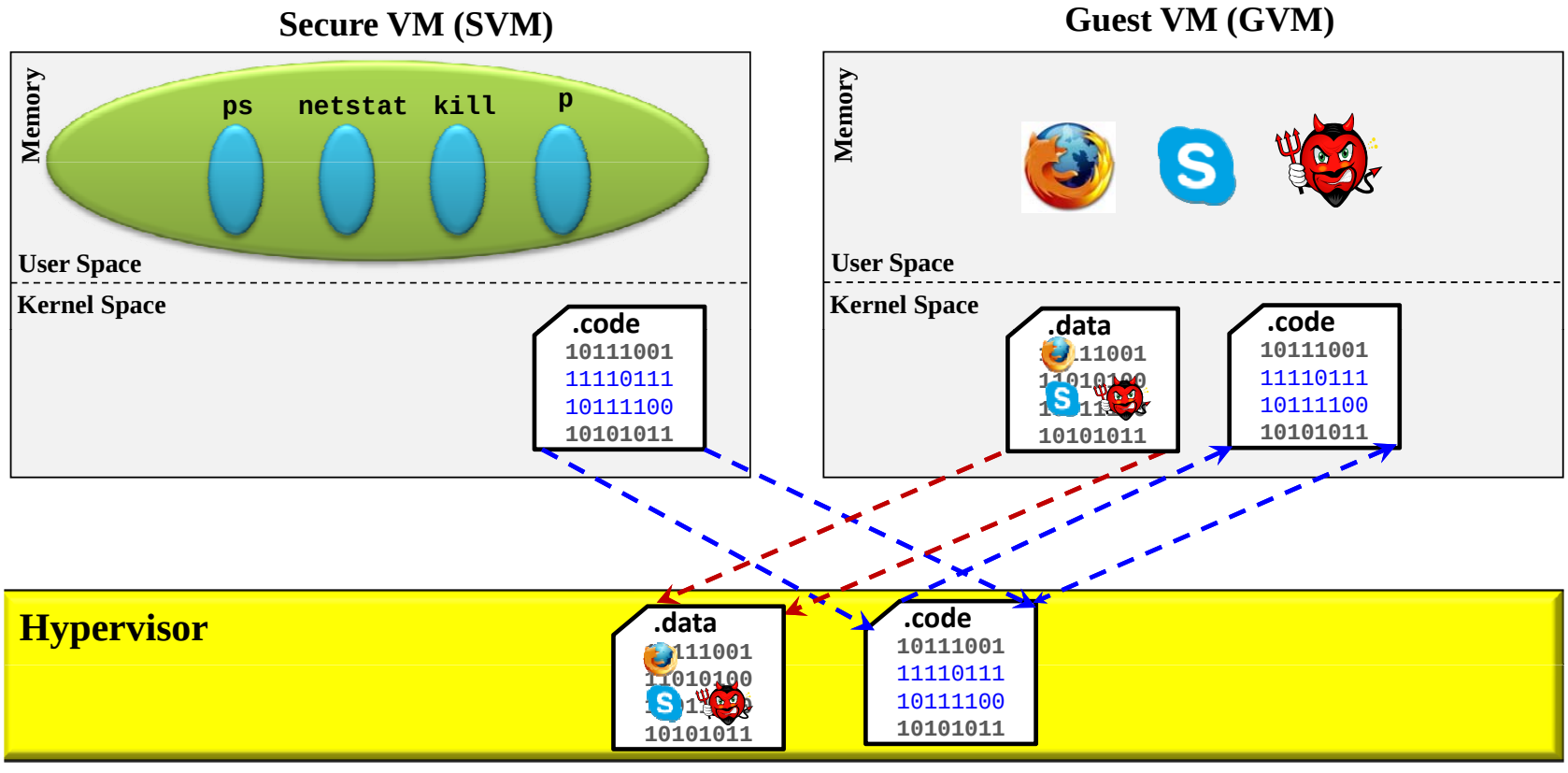
Using Code Implanting [DSN'10, ATC'14, DSN'14]



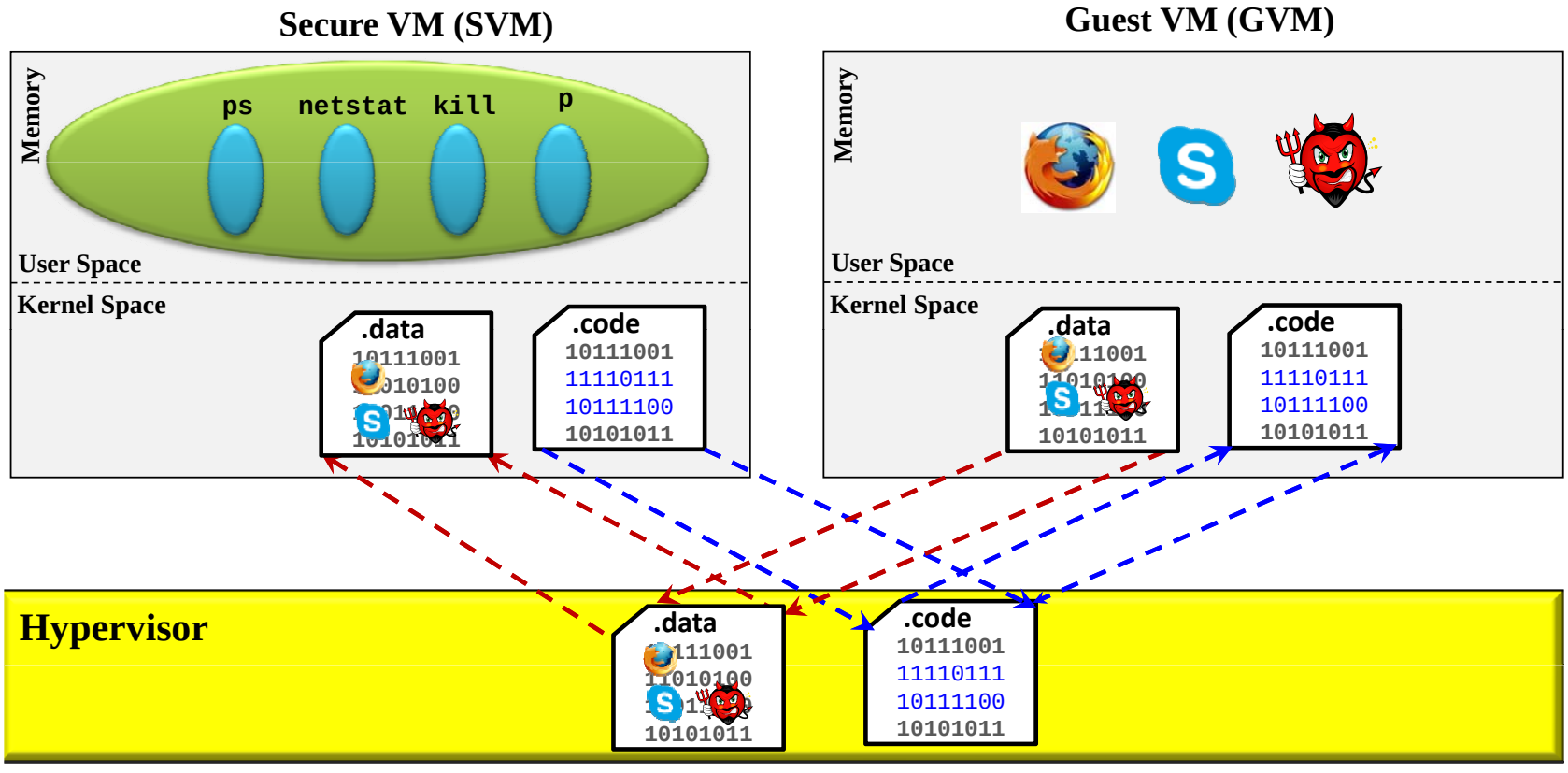
Using Code Implanting [DSN'10, ATC'14, DSN'14]



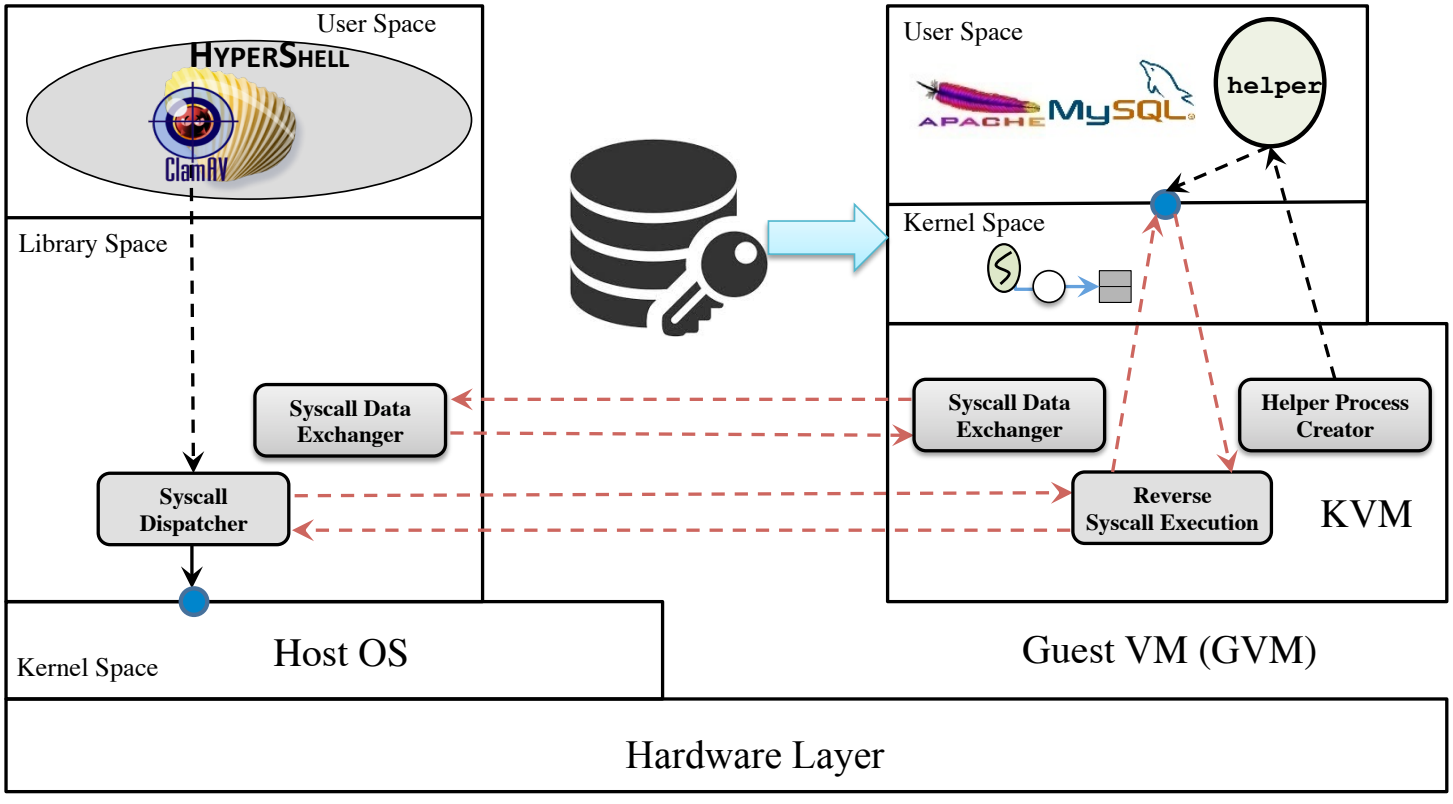
Using Code Implanting [DSN'10, ATC'14, DSN'14]



Using Code Implanting [DSN'10, ATC'14, DSN'14]



Disk Introspection: FDE disk virus scanning [ATC'14]



Disk Introspection: FDE disk virus scanning [ATC'14]

1. Encypted by dm-crypt
2. 101,415 files
3. 1336.09 megabytes in size

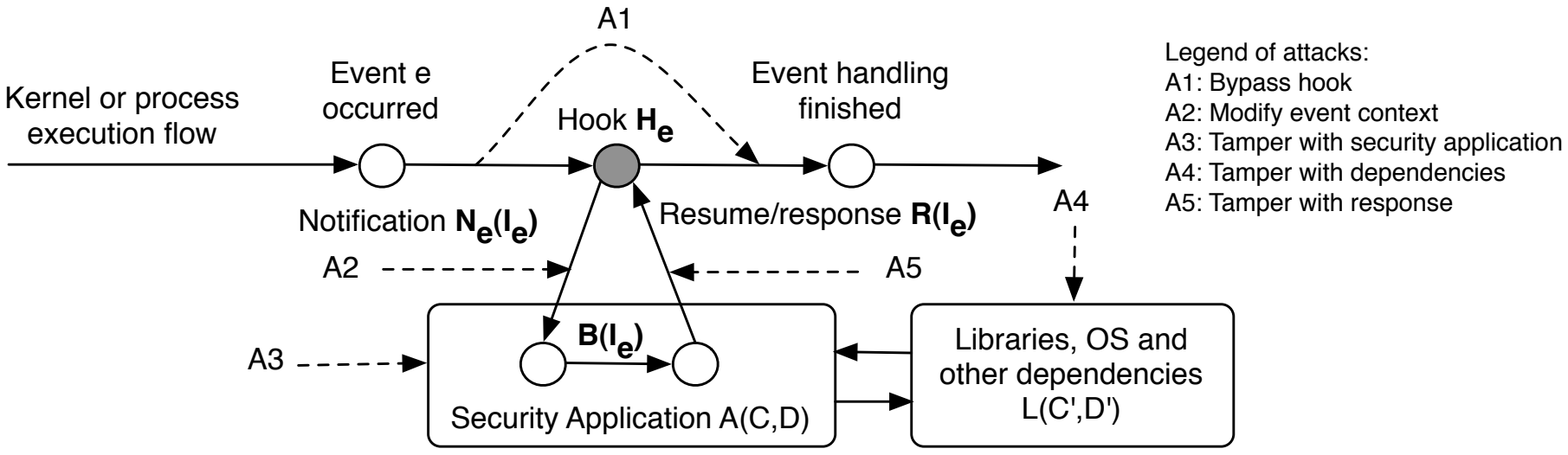
Disk Introspection: FDE disk virus scanning [ATC'14]

1. Encypted by dm-crypt
2. 101,415 files
3. 1336.09 megabytes in size

Clamav successfully detect two viruses!!

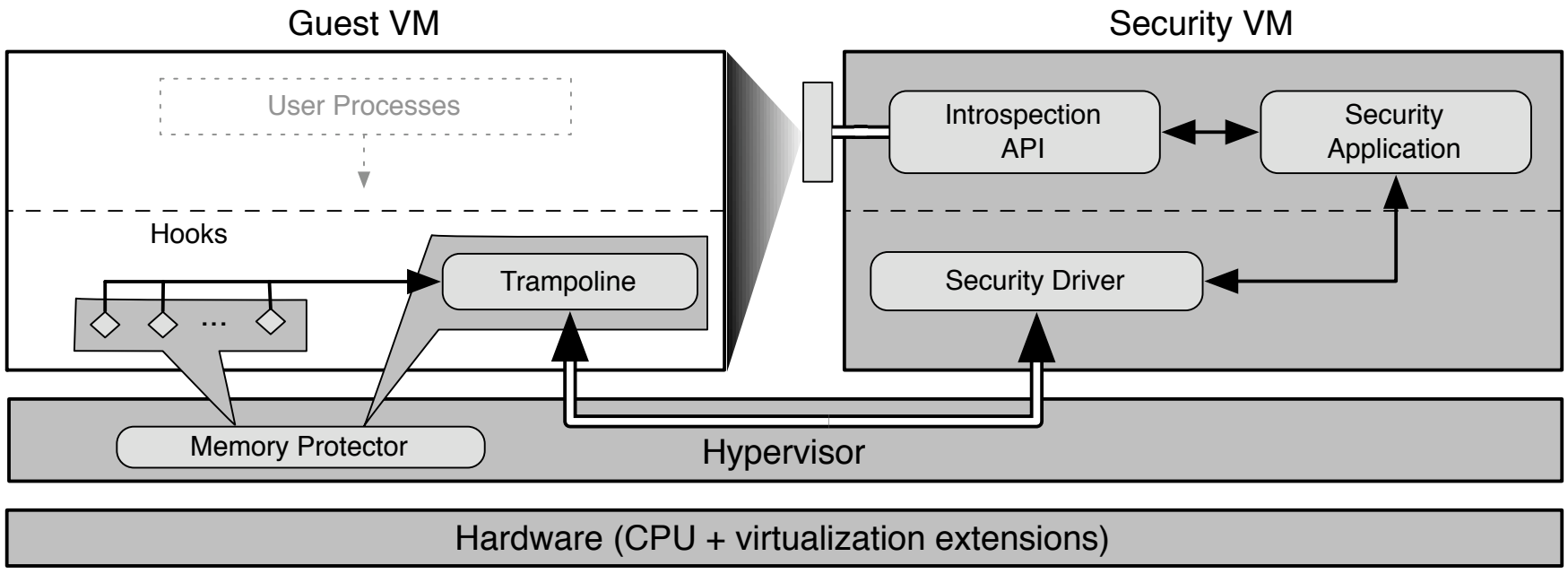
Outline

Continues Monitoring [Payen et al. SP'08]

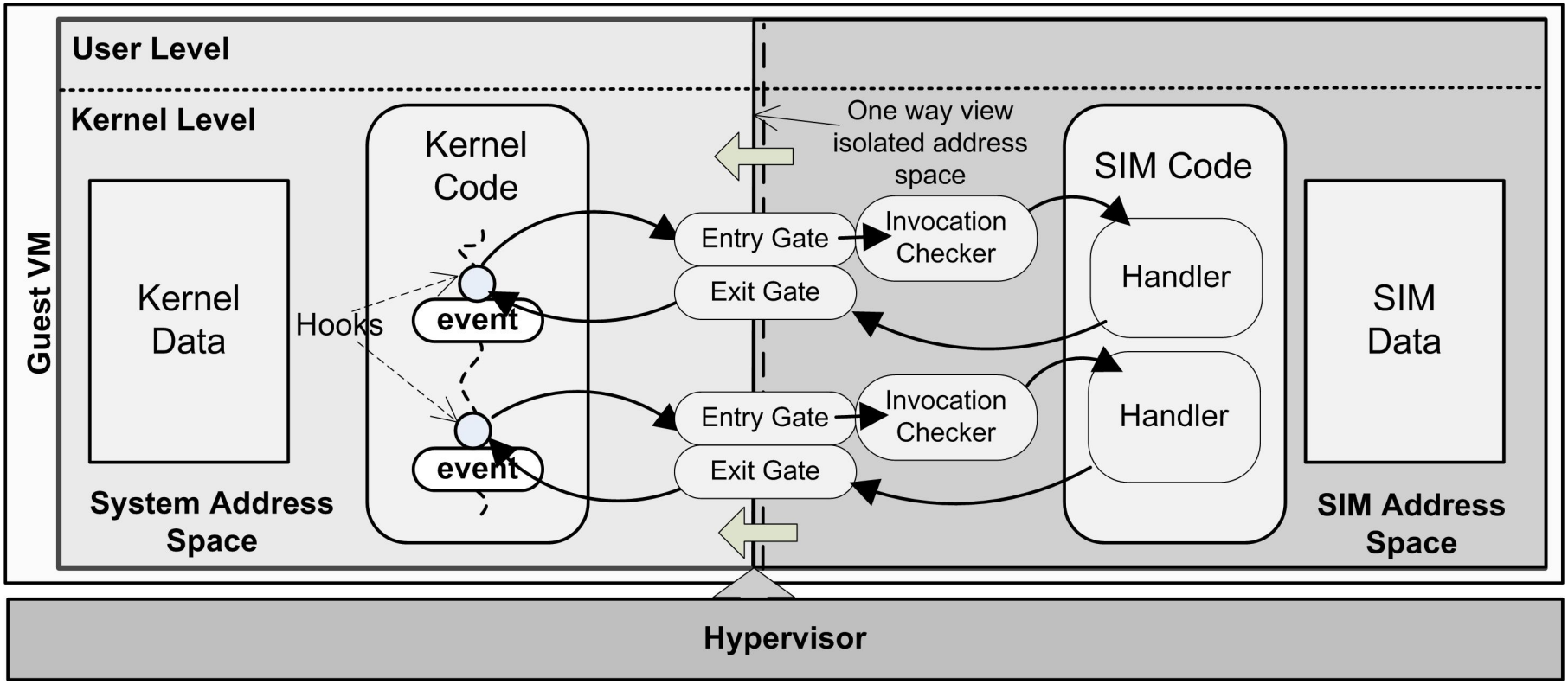


Formal model of secure active monitoring shown with potential attacks.

Continues Monitoring [Payen et al. SP'08]

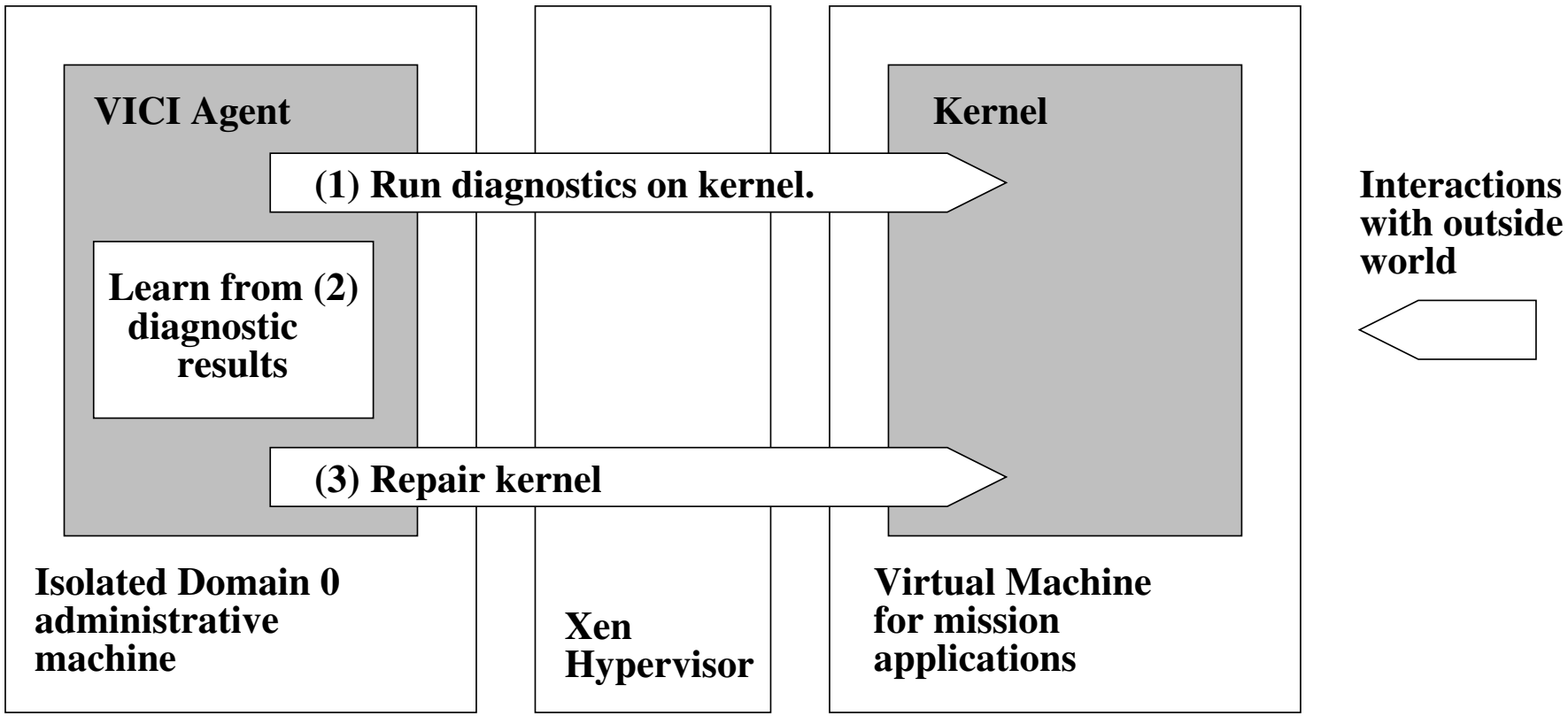


In-VM Monitoring w/ Hardware[Sharif et al. CCS'09]



Outline

VMI based Attack Repair [Fraser et al. ACSAC'08]



Out-of-Box Attack Recovery, Repair

Rootkit	Targeted Function Pointer	Repaired?
adore-2.6	kernel global, heap object	✗
hookswrite	IDT table	✓
int3backdoor	IDT table	✓
kbdv3	syscall table	✓
kbeast-v1	syscall table, tcp4_seq_show	✓
mood-nt-2.3	syscall table	✓
override	syscall table	✓
phalanx-b6	syscall table, tcp4_seq_show	✓
rkit-1.01	syscall table	✓
rial	syscall table	✓
suckit-2	IDT table	✓
synapsys-0.4	syscall table	✓

Table : Rootkit Repairing with An Exterior [Fu and Lin, VEE'13] Tool.

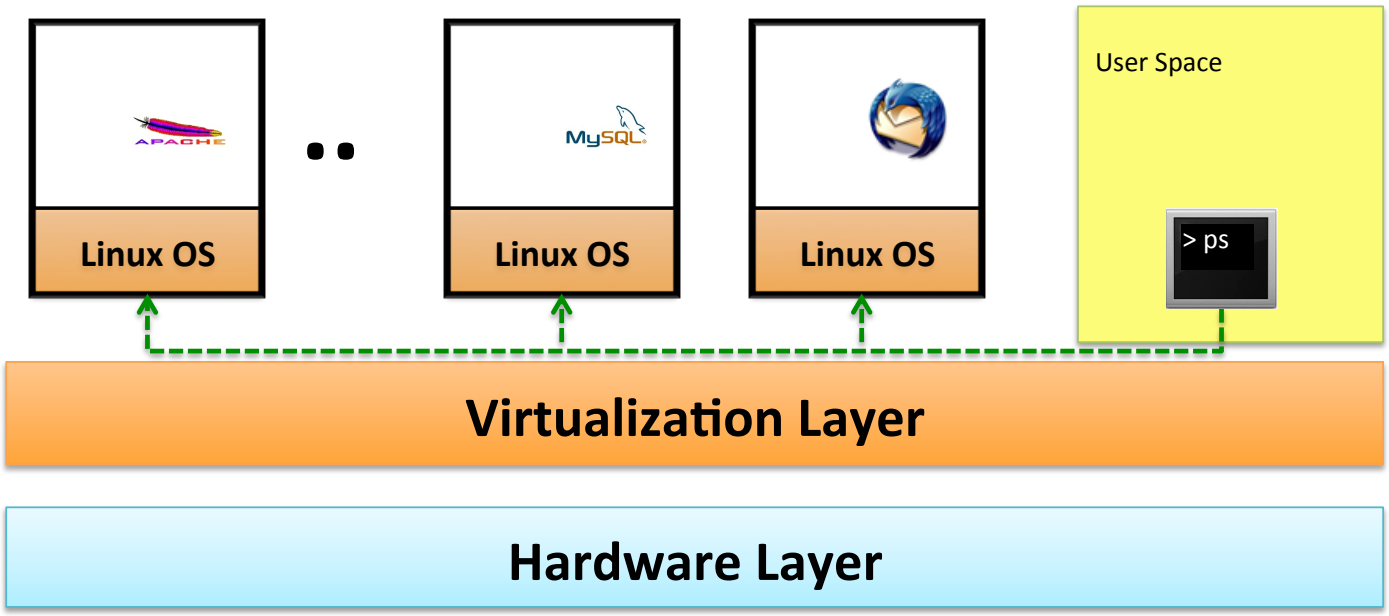
Outline

Non Applications

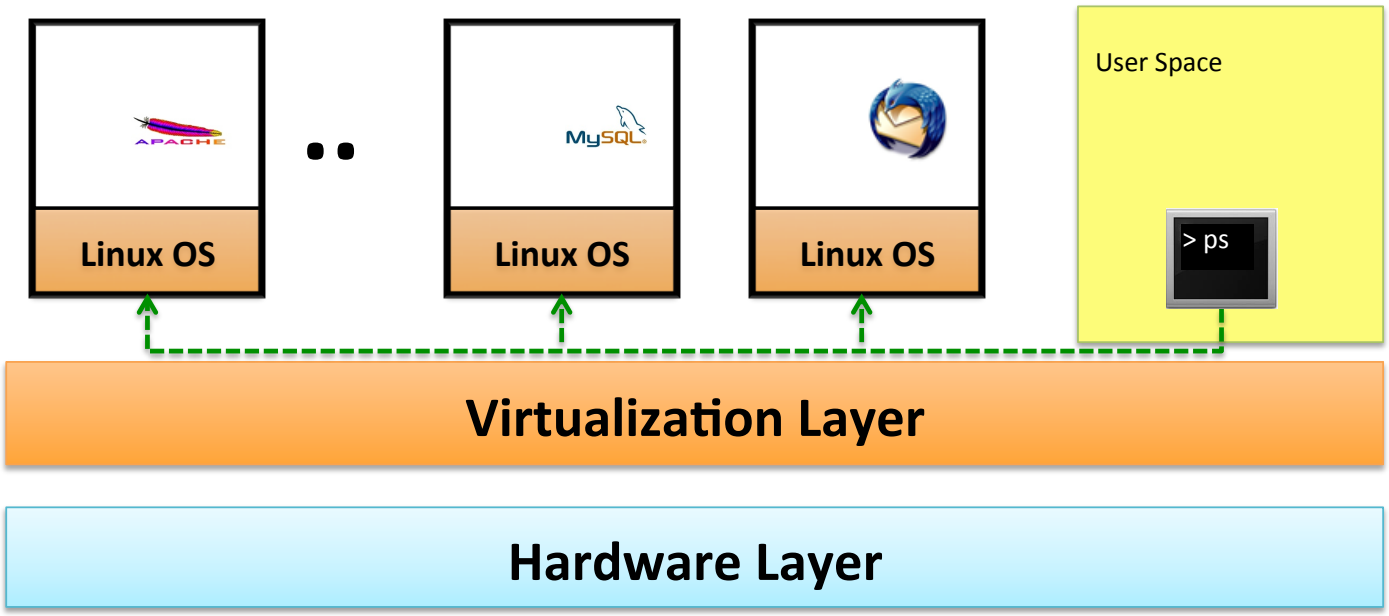
Non Security

1. Virtual machine management
2. Process management
3. High performance computing
4. Autonomous computing
5. ...

Out-of-VM Management: Writable VMI [ATC'14]



Out-of-VM Management: Writable VMI [ATC'14]



Advantages

- Only install the management utilities at hypervisor layer.
- Automated, uniformed, and centralized management.

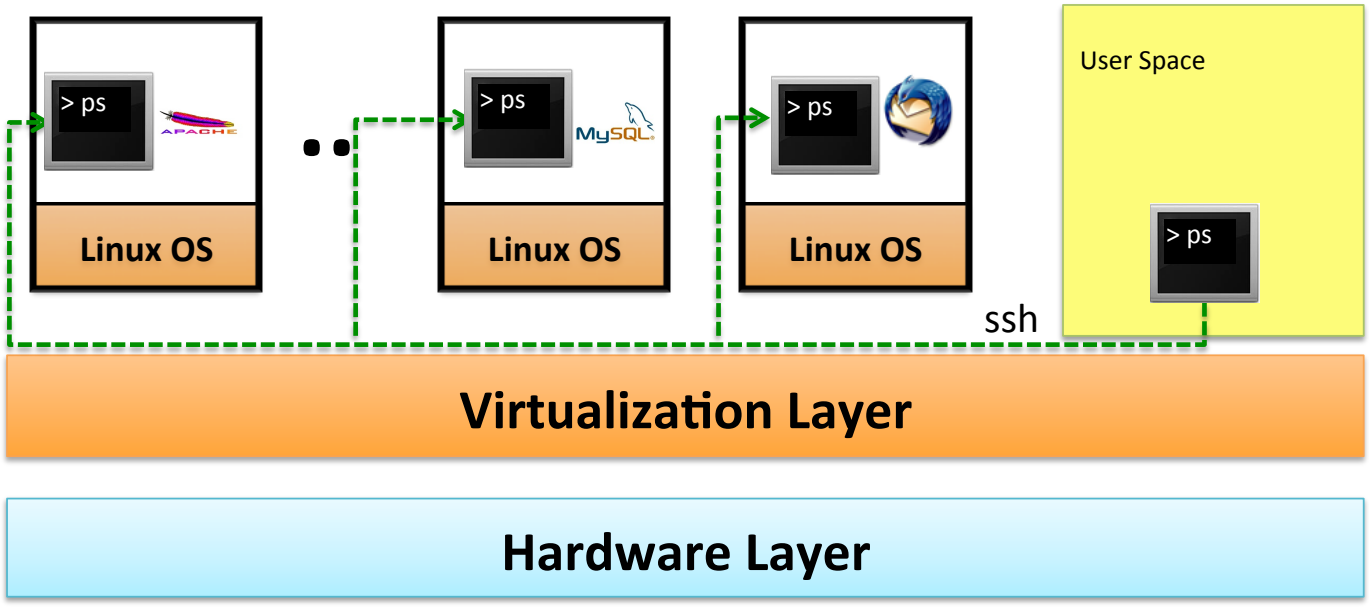
In-VM Management: Existing Approaches

In-VM Management: Existing Approaches

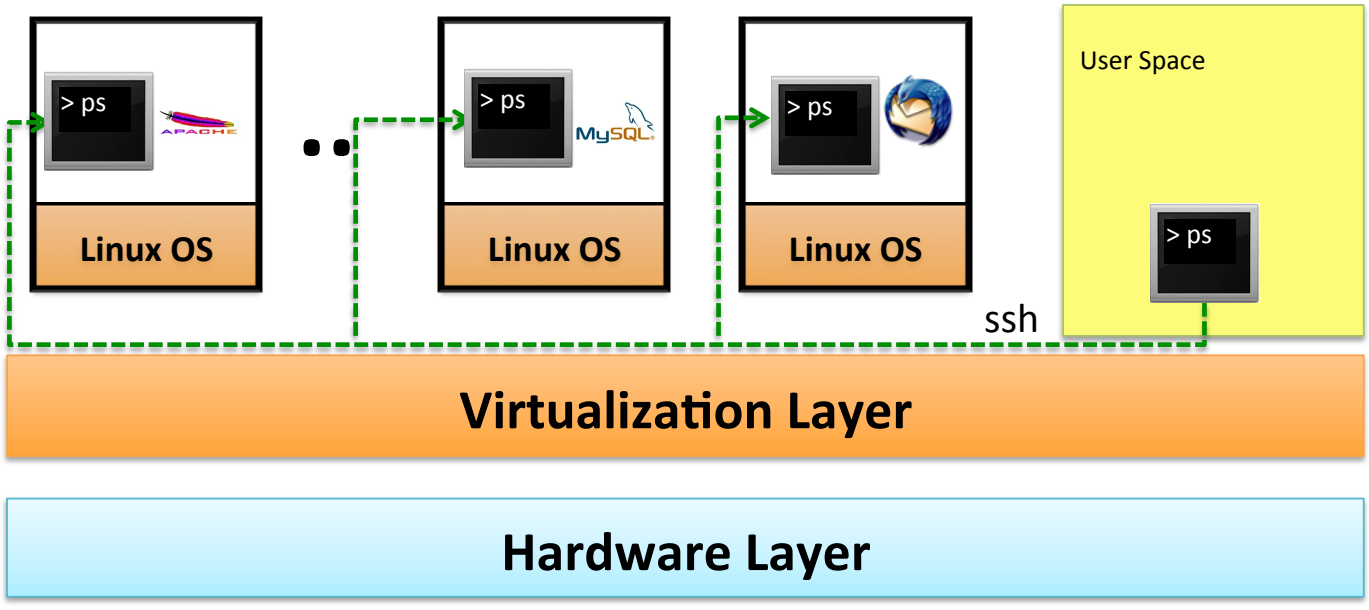
Disadvantages

- Scattered, distributed
- Install, update, and execute in each VM

In-VM Management: Existing Approaches



In-VM Management: Existing Approaches



Disadvantages

- Requiring the (admin) login password.
- Requiring install the management utilities in each VM.

Performance Impact: HyperShell [ATC'14]

Process	**S**	$B(ms)$	$D(ms)$	$T(X)$
ps	✗	1.33	5.42	4.08
pidstat	✗	1.95	7.56	3.88
nice	✓	0.07	0.11	1.57
getpid	✓	0.01	0.02	2.00
mpstat	✗	0.29	0.66	2.28
pstree	✗	0.69	6.03	8.74
chrt	✓	0.11	0.16	1.45
renice	✓	0.11	0.18	1.64
top	✗	504.92	510.85	1.01
nproc	✓	0.07	0.26	3.71
sleep	✓	1.27	1.28	1.01
pgrep	✓	0.89	4.72	5.30
pkill	✓	0.87	4.33	4.98
snice	✓	0.17	0.65	3.82
echo	✓	0.07	0.09	1.29
pwdx	✓	0.05	0.07	1.40
pmap	✓	0.16	0.36	2.25
kill	✓	0.01	0.04	4.00
killall	✓	0.62	3.03	4.89
Memory	**S**	$B(ms)$	$D(ms)$	$T(X)$
free	✗	0.04	0.08	2.00
vmstat	✗	0.19	0.33	1.74
slabtop	✗	0.22	0.36	1.64
Modules	**S**	$B(ms)$	$D(ms)$	$T(X)$
rmmod	✓	0.51	3.14	6.16
modinfo	✓	0.48	1.54	3.21
lsmod	✓	0.10	0.17	1.70
Environment	**S**	$B(ms)$	$D(ms)$	$T(X)$
who	✓	0.14	0.72	5.14
env	✓	0.07	0.11	1.57
printenv	✓	0.07	0.1	1.43
whoami	✓	0.19	0.45	2.37
stty	✓	0.11	0.46	4.18
users	✓	0.09	0.53	5.89
uname	✓	0.09	0.11	1.22
id	✓	0.26	0.85	3.27
date	✗	0.11	0.12	1.09
w	✗	0.95	6.62	6.97
hostname	✓	0.04	0.06	1.50
groups	✓	0.21	0.62	2.95
hostid	✓	0.16	0.56	3.50
locale	✓	0.09	0.17	1.89
getconf	✓	0.09	0.34	3.78
System Utils	**S**	$B(ms)$	$D(ms)$	$T(X)$
uptime	✗	0.07	0.47	6.71
sysctl	✓	8.5	42.72	5.03
arch	✓	0.07	0.11	1.57
dmesg	✓	0.38	0.51	1.34
lscpu	✓	0.26	1.21	4.65
mcookie	✗	0.29	0.49	1.69
Disk/Devices	**S**	$B(ms)$	$D(ms)$	$T(X)$
blkid	✓	0.14	0.61	4.36
badblocks	✓	0.35	0.44	1.26
lspci	✓	31.40	36.52	1.16
iostat	✓	0.45	1.04	2.31
du	✓	0.11	0.53	4.82
df	✓	0.16	0.35	2.19
Filesystem	**S**	$B(ms)$	$D(ms)$	$T(X)$
sync	✓	8.07	6.53	0.81
getcap	✓	0.04	0.08	2.00
lsof	✓	3.31	6.12	1.85
pwd	✓	0.07	0.11	1.57
Files	**S**	$B(ms)$	$D(ms)$	$T(X)$
chgrp	✓	0.19	0.47	2.47
chmod	✓	0.07	0.14	2.00
chown	✓	0.19	0.47	2.47
cp	✓	0.11	0.27	2.45
uniq	✓	0.09	0.35	3.89
file	✓	0.87	1.72	1.98
find	✓	0.20	0.58	2.90
grep	✓	0.35	2.14	6.11
ln	✓	0.08	0.14	1.75
ls	✓	0.14	0.27	1.93
mkdir	✓	0.10	0.19	1.90
mkfifo	✓	0.10	0.19	1.90
mknod	✓	0.10	0.19	1.90
mv	✓	0.15	0.31	2.07
rm	✓	0.08	0.15	1.88
od	✓	0.12	0.35	2.92
cat	✓	0.07	0.18	2.57
link	✓	0.07	0.13	1.86
comm	✓	0.08	0.22	2.75
shred	✗	0.72	0.92	1.28
truncate	✓	0.07	0.26	3.71
head	✓	0.07	0.15	2.14
vdir	✓	0.63	3.95	6.27
nl	✓	0.08	0.17	2.13
tail	✓	0.08	0.20	2.50
namei	✓	0.07	0.13	1.86
whereis	✓	2.05	4.86	2.37
stat	✓	0.27	0.78	2.89
readlink	✓	0.07	0.12	1.71
unlink	✓	0.07	0.13	1.86
cut	✓	0.08	0.17	2.13
dir	✓	0.07	0.20	2.86
mktemp	✓	0.09	0.18	2.00
rmdir	✓	0.07	0.13	1.86
ptx	✓	0.12	0.45	3.75
chcon	✓	0.06	0.12	2.00
Network	**S**	$B(ms)$	$D(ms)$	$T(X)$
ifconfig	✗	0.32	1.15	3.59
ip	✓	0.10	0.20	2.00
route	✓	138.65	150.32	1.08
ipmaddr	✓	0.13	0.34	2.62
iptunnel	✓	0.09	0.29	3.22
nameif	✓	0.10	0.21	2.10
netstat	✗	0.25	0.37	1.48
arp	✓	0.14	0.24	1.71
ping	✗	15.02	18.2	1.21
Avg.	-	7.27	8.45	2.73

Outline

Deployment

1. Bare metal (e.g., Xen, vSphere, Hyper-V)
2. Hosted, Native Hypervisor (e.g., KVM)
3. Hosted, Emulation Hypervisor (e.g., QEMU)
4. Extra Hardware (e.g., PCI device in Copilot)

Deployment Comparison

Metric	Definition
Flexibility	How many constraints are imposed on the monitor
Security	How well the deployment type provides for security coverage
Invisibility	How difficult the presence of the monitor is to detect from within the VM
Speed	How much system slowdown occurs compared to no monitor running
Space	How much storage capability the deployment type possesses

Table : Definitions of the metrics used to compare out-of-VM monitor deployment types.

Deployment Comparison

Approaches	Flexibility	Security	Invisibility	Speed	Space
Bare Metal	●	◐	◐	●	●
Hosted, Native Hypervisor	●	◐	◐	●	●
Hosted, Emulation Hypervsior	●	◐	◐	○	●
Extra Hardware	○	●	●	●	○

Table : Comparison between different out-of-VM monitor deployment types. Note that symbol ○ denotes a low degree for that comparison item, ◐ denotes a medium degree, and ● denotes a high degree.