EXTERIOR: Using A Dual-VM Based External Shell for Guest-OS Introspection, Configuration, and Recovery

Yangchun Fu, Zhiqiang Lin

Department of Computer Science
The University of Texas at Dallas

March 17th, 2013

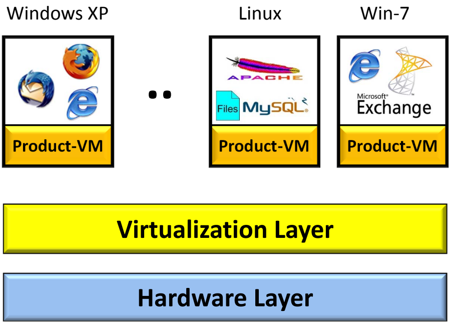
Outline

1. Overview
2. Our Approach
3. Evaluations
4. Conclusion

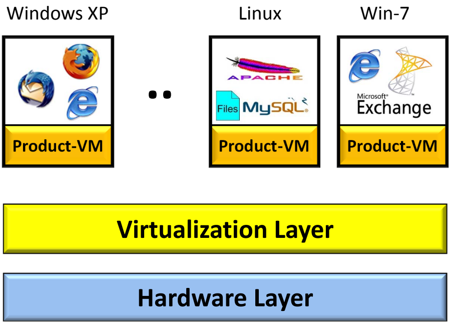
Outline

1. Overview
2. Our Approach
3. Evaluations
4. Conclusion

Virtualization

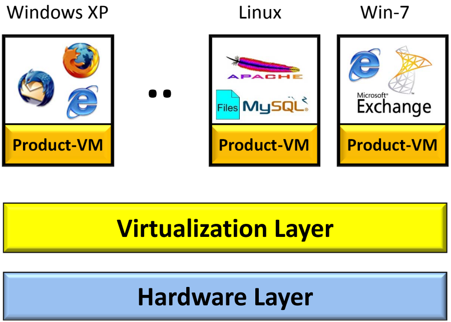


Virtualization



Virtualization (i.e., hypervisor) [Popek and Goldberg, 1974] has pushed our computing paradigm from **multi-tasking** to **multi-OS**.

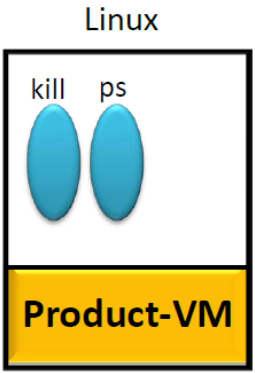
Virtualization



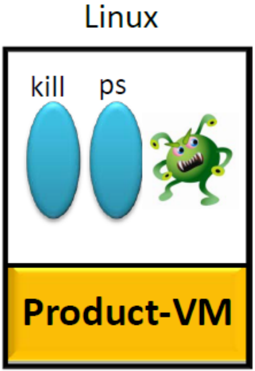
Virtualization (i.e., hypervisor) [Popek and Goldberg, 1974] has pushed our computing paradigm from **multi-tasking** to **multi-OS**.

Consolidation, Migration, Isolation ...

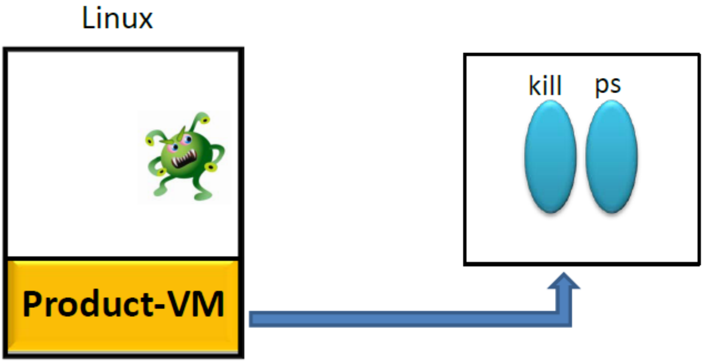
Execution Mode



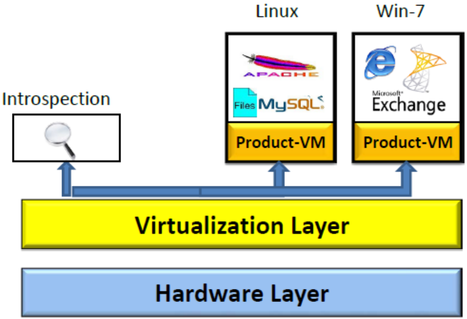
Execution Mode



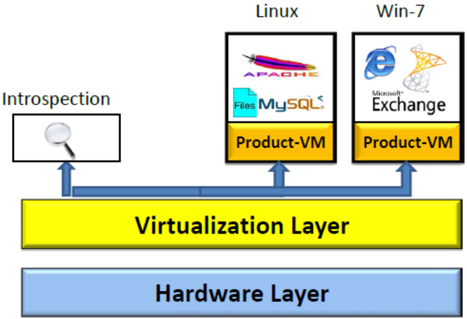
Execution Mode



Virtual Machine Introspection (VMI) [Garfinkel et al, NDSS'03]

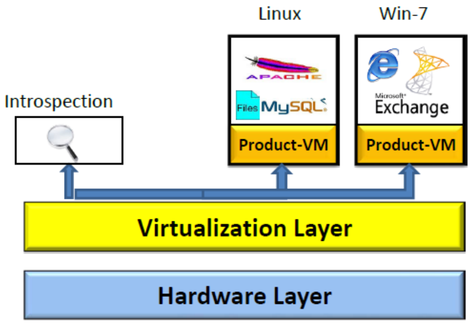


Virtual Machine Introspection (VMI) [Garfinkel et al, NDSS'03]



Using a trusted, dedicated virtualization layer program to monitor the running VMs

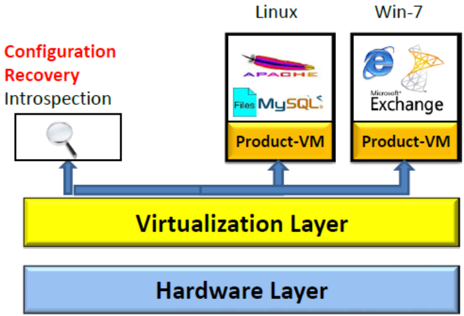
Virtual Machine Introspection (VMI) [Garfinkel et al, NDSS'03]



Using a trusted, dedicated virtualization layer program to monitor the running VMs

- Intrusion Detection
- Malware Analysis
- Memory Forensics

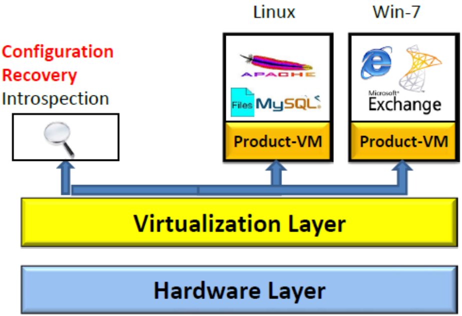
Virtual Machine Introspection (VMI)



Using a trusted, dedicated virtualization layer program to monitor the running VMs

- Intrusion Detection
- Malware Analysis
- Memory Forensics

Virtual Machine Introspection (VMI)



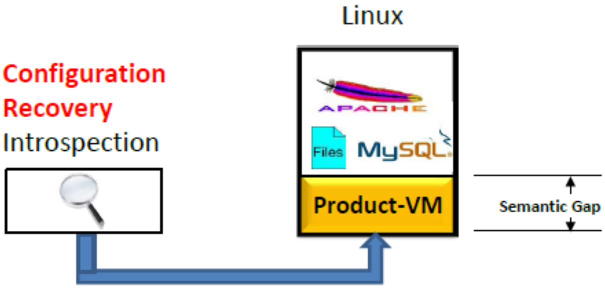
Using a trusted, dedicated virtualization layer program to monitor the running VMs

- Intrusion Detection
- Malware Analysis
- Memory Forensics

EXTERIOR

- Execute trusted utilities in SVM for timely Guest-OS introspection, (re)configuration and recovery.

The Semantic Gap in VMI ([Chen and Noble HotOS'01])



- View exposed by Virtual Machine Monitor is at low-level
- There is no abstraction and no APIs
- Need to reconstruct the guest-OS abstraction

Outline

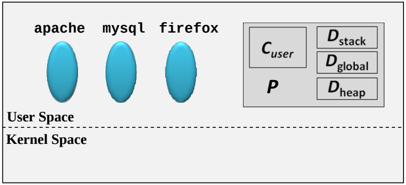
1. Overview

2. Our Approach

3. Evaluations

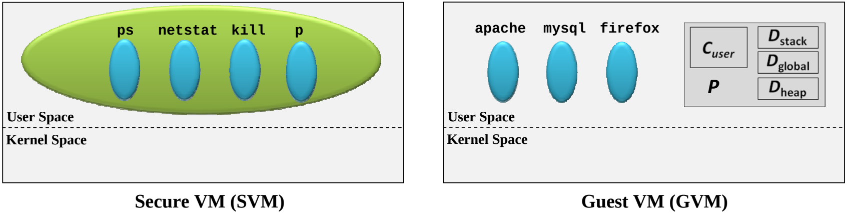
4. Conclusion

Using a Dual-VM Architecture

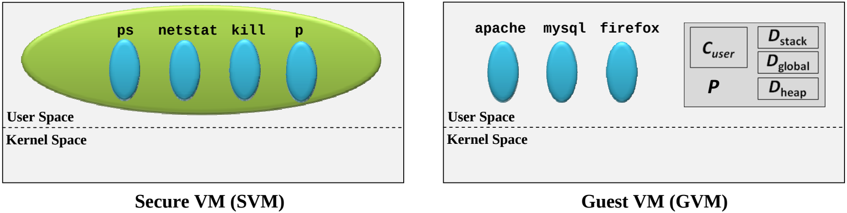


Guest VM (GVM)

Using a Dual-VM Architecture

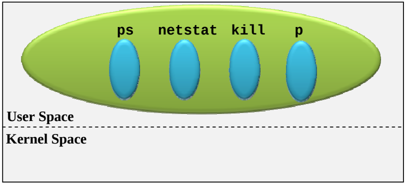


Using a Dual-VM Architecture

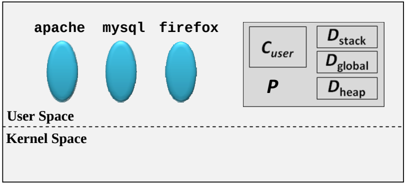


- Virtual Machine Introspection
- Virtual Machine Configuration
- Intrusion Detection, Prevention (Recovery)

Advantages



Secure VM (SVM)



Guest VM (GVM)

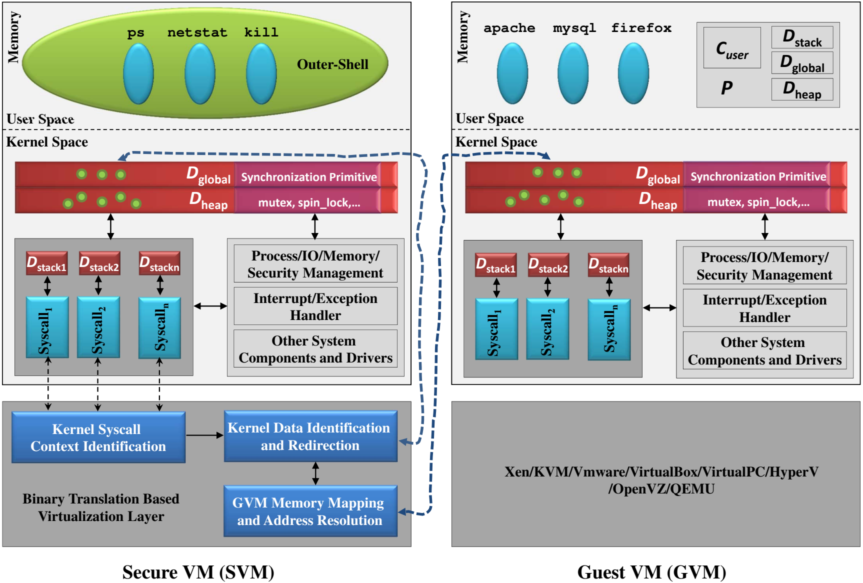
- **Isolation** (SVM and GVM are isolated)
- **Trustworthiness** (trust code is running in secure VM)
- **Automation** (no need to develop introspection utilities)
- **Security** (enabling malware analysis, forensics...)
- **Transparency** (programmers write native program in SVM)

Observation

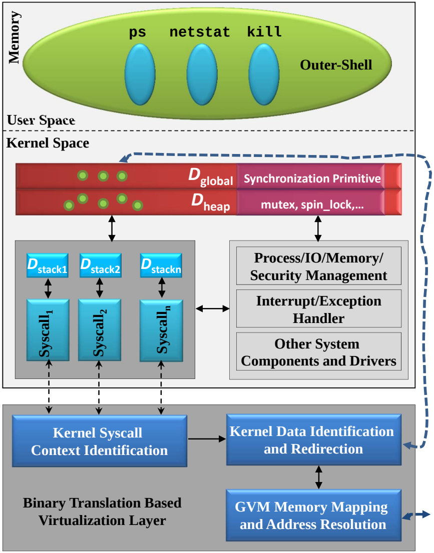
```
 1 execve("/sbin/sysctl",["sysctl", "-w","kernel..=1"],...) = 0
 2 brk(0)                                           = 0x604000
 3 access("/etc/ld.so.nohwcap",F_OK)               = -1 ENOENT
 4 mmap(NULL, 8192, PROT_READ|.., -1,0) = 0x7f07b1749000
 5 access("/etc/ld.so.preload",R_OK)               = -1 ENOENT
 6 open("/etc/ld.so.cache", O_RDONLY)               = 3
 ...
47 open("/proc/sys/kernel/randomize_va_space",O_WRONLY|...) = 3
48 fstat(3, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
49 mmap(NULL, 4096, PROT_READ|.., -1, 0) = 0x7f07b1748000
50 write(3, "1\n", 2)                               = 2
51 close(3)                                         = 0
...
57 exit_group(0)                                    = ?
```

Syscall trace of running `sysctl -w` to turn on the address space randomization in Linux kernel 2.6.32

Architecture Overview of EXTERIOR

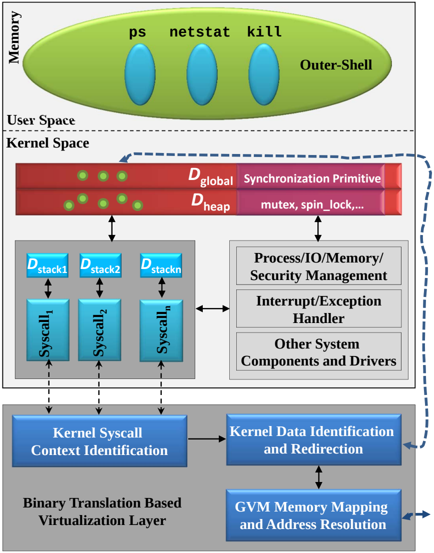

Memory
ps
netstat
kill
Outer-Shell
User Space
Kernel Space
D_{global}
Synchronization Primitive
D_{heap}
mutex, spin_lock,...
D_{stack1}
D_{stack2}
D_{stackn}
$Syscall_1$
$Syscall_2$
$Syscall_n$
Process/IO/Memory/ Security Management
Interrupt/Exception Handler
Other System Components and Drivers
Kernel Syscall Context Identification
Kernel Data Identification and Redirection
GVM Memory Mapping and Address Resolution
Binary Translation Based Virtualization Layer
Secure VM (SVM)
Memory
apache
mysql
firefox
C_{user}
P
D_{stack}
D_{global}
D_{heap}
User Space
Kernel Space
D_{global}
Synchronization Primitive
D_{heap}
mutex, spin_lock,...
D_{stack1}
D_{stack2}
D_{stackn}
$Syscall_1$
$Syscall_2$
$Syscall_n$
Process/IO/Memory/ Security Management
Interrupt/Exception Handler
Other System Components and Drivers
Xen/KVM/Vmware/VirtualBox/VirtualPC/HyperV /OpenVZ/QEMU
Guest VM (GVM)

The algorithms


Memory
ps
netstat
kill
Outer-Shell
User Space
Kernel Space
D_{global}
Synchronization Primitive
D_{heap}
mutex, spin_lock,...
D_{stack1}
D_{stack2}
D_{stackn}
$Syscall_1$
$Syscall_2$
$Syscall_n$
Process/IO/Memory/ Security Management
Interrupt/Exception Handler
Other System Components and Drivers
Kernel Syscall Context Identification
Kernel Data Identification and Redirection
Binary Translation Based Virtualization Layer
GVM Memory Mapping and Address Resolution

Secure VM (SVM)

The algorithms

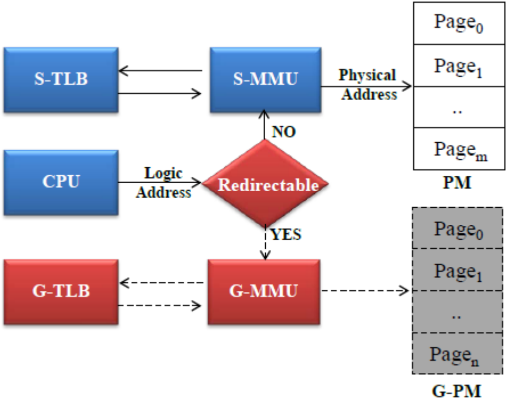

Memory
ps
netstat
kill
Outer-Shell
User Space
Kernel Space
D_{global}
Synchronization Primitive
D_{heap}
mutex, spin_lock,...
D_{stack1}
D_{stack2}
D_{stackn}
$Syscall_1$
$Syscall_2$
$Syscall_n$
Process/IO/Memory/ Security Management
Interrupt/Exception Handler
Other System Components and Drivers
Kernel Syscall Context Identification
Kernel Data Identification and Redirection
Binary Translation Based Virtualization Layer
GVM Memory Mapping and Address Resolution

Secure VM (SVM)

The Algorithm

```
DynamicBinaryInstrumentation($i$):
  if SysCallExecContext($s$):
    if SysCallRedirectable($s$):
      RedirectableDataTracking($i$);
    for $\alpha$ in MemoryAddress($i$):
      if DataRead($\alpha$):
        $PA(\alpha) \leftarrow$ V2P($\alpha$)
        Load($PA(\alpha)$)
      else:
        if Configuration:
          Store($PA(\alpha)$)
        else: //Introspection
          COW-Store($PA(\alpha)$)
```

Mapping the GVM Memory Address

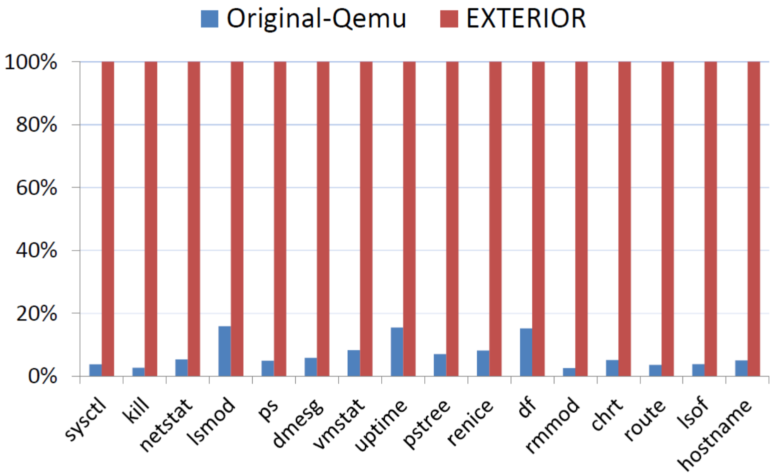


Outline

Effectiveness

Category	Utility	Effective?	
		Syntactics	Semantics
Introspection	ps (1)	✗	✓
	pstree (1)	✗	✓
	lsmod (8)	✓	✓
	dmesg (1)	✓	✓
	vmstat (8)	✗	✓
	netstat (8)	✓	✓
	lsof (8)	✗	✓
	uptime (1)	✗	✓
	df (1)	✗	✓
Configuration	sysctl (8)	✓	✓
	route (8)	✓	✓
	hostname (1)	✓	✓
	chrt (1)	✓	✓
	renice (1)	✓	✓
Recovery	kill (1)	✓	✓
	rmmod (8)	✓	✓

Performence Overhead



Recovery

Rootkit	Targeted Function Pointer	Succeed?
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	✓

OS-Agnostic Testing

Linux Distribution	Kernel Version	Release Date	Transparent?
Debian 4.0	2.6.26	2007-04-06	✓
Debian 5.0	2.6.28	2009-02-12	✓
Debian 6.0	2.6.32	2010-01-22	✓
Fedora-8	2.6.23	2007-11-08	✓
Fedora-10	2.6.27	2008-11-25	✓
Fedora-12	2.6.31	2009-11-17	✓
Fedora-14	2.6.35	2010-11-02	✓
Fedora-16	3.1.0	2011-11-08	✓
OpenSUSE-10.3	2.6.22	2007-10-04	✓
OpenSUSE-11.0	2.6.25	2008-06-19	✓
OpenSUSE-11.1	2.6.27	2008-12-18	✓
OpenSUSE-11.2	2.6.31	2009-11-12	✓
OpenSUSE-11.3	2.6.34	2010-07-15	✓
OpenSUSE-12.1	3.1.0	2011-11-16	✓
Ubuntu-8.04	2.6.24	2008-04-24	✓
Ubuntu-8.10	2.6.27	2008-10-30	✓
Ubuntu-9.04	2.6.28	2009-04-23	✓
Ubuntu-9.10	2.6.31	2009-10-29	✓
Ubuntu-10.04	2.6.32	2010-04-29	✓
Ubuntu-10.10	2.6.35	2010-10-10	✓
Ubuntu-11.04	2.6.38	2011-04-28	✓
Ubuntu-11.10	3.0.4	2011-10-13	✓

Limitations and Future Work

Limitations

- Can handle kernel ASLR
- Need an identical trusted kernel
- Need to stop the guest VM

Limitations and Future Work

Limitations

- Can handle kernel ASLR
- Need an identical trusted kernel
- Need to stop the guest VM

Future Work

- Derandomize the kernel address space
- Port to Windows OS

Outline

Conclusion

- EXTERIOR is a novel dual-VM based external shell for trusted, native, out-of-VM program execution.

Conclusion

- EXTERIOR is a novel dual-VM based external shell for trusted, native, out-of-VM program execution.

- It can be used for (automatic) introspection, (re)configuration of the guest-OS state (in the cloud), and can perform a timely response such as recovery from a kernel malware intrusion.

Conclusion

- EXTERIOR is a novel dual-VM based external shell for trusted, native, out-of-VM program execution.

- It can be used for (automatic) introspection, (re)configuration of the guest-OS state (in the cloud), and can perform a timely response such as recovery from a kernel malware intrusion.

- EXTERIOR has demonstrated a new program execution model on top of virtualization.

Conclusion

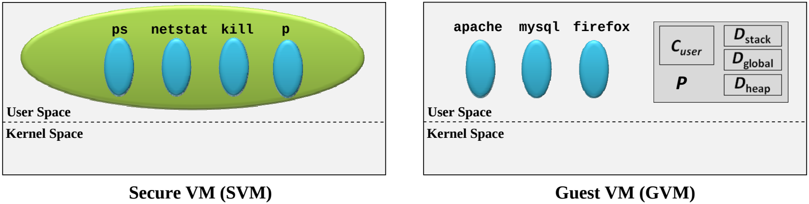
- EXTERIOR is a novel dual-VM based external shell for trusted, native, out-of-VM program execution.

- It can be used for (automatic) introspection, (re)configuration of the guest-OS state (in the cloud), and can perform a timely response such as recovery from a kernel malware intrusion.

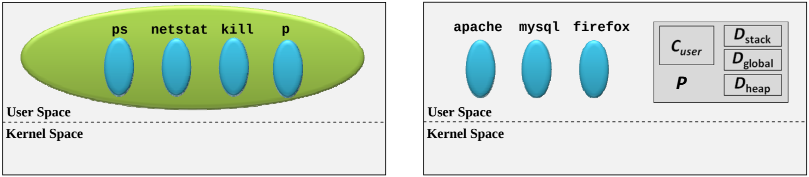
- EXTERIOR has demonstrated a new program execution model on top of virtualization.

- (We believe) It will open new opportunities for system administration and security.

Thank you !


ps
netstat
kill
p
User Space
Kernel Space
Secure VM (SVM)
apache
mysql
firefox
C_{user}
D_{stack}
D_{global}
P
D_{heap}
User Space
Kernel Space
Guest VM (GVM)

Thank you !



Secure VM (SVM)

Guest VM (GVM)

Contact us via. {yangchun.fu,zhiqiang.lin}@utdallas.edu for any questions