End to End Defense against Rootkits in Cloud Environment

Sachin Shetty

Associate Professor
Electrical and Computer Engineering
Director, Cybersecurity Laboratory
Tennessee State University

Tennessee State University
Cyber Security Research Lab

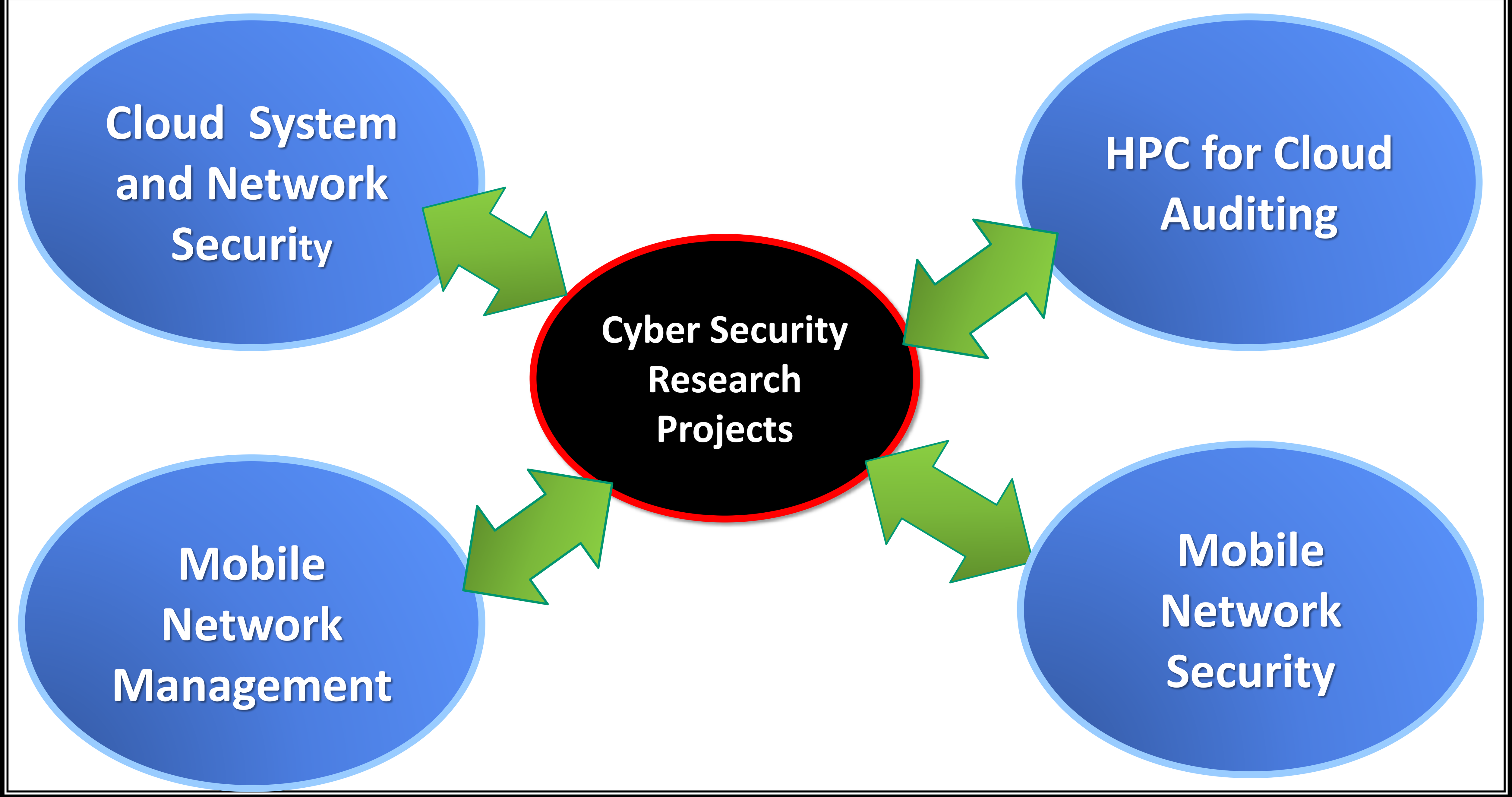
Mission

- Development of secure algorithms, security protocols and visual analytic tools to protect next generation Internet, cloud and mobile systems and networks from emerging cyber threats
- Support Ph.D. dissertations, Master Theses and Undergraduate research projects
- Infuse Cybersecurity based concepts and laboratory modules in undergraduate and graduate curriculum

Technologies

- **Cloud Security:** Emerging Cloud Threat Analysis, Cloud System and Network Traffic Collection Techniques, Monitoring and Characterization for Cloud Auditing
- **High Performance Computing for Cloud Auditing:** Hadoop based Distributed Computing and Real time computing on Spark and Storm
- **Mobile Network Management and Security :** Dynamic Spectrum Access, Secure Resource Allocation in Cognitive Radio Network, and Cloud centric security solution to protect smartphones

Research Areas



Collaboration/Funding/Achievements

- **Collaboration:** Federal Research Labs (AFRL and NSWC Crane), Academia (Georgia Tech, IUPUI, PSU, RIT, UTSA, TAMU, UIUC), DHS CoE (VACCINE and CCICADA)
- **Centers of Excellence -** DoD Center of Excellence in Cyber Security and DHS Center of Excellence in Infrastructure Resilience
- **Funding:** Grants and contracts from AFOSR, AFRL, NSF, DHS, DOE, ONR, Boeing and Amazon
- **Publications:** Over 80 articles in conferences, workshops, journals, and books published

POC : Dr. Sachin Shetty, Email: sshetty@tnstate.edu, Web: http://cyberviz.tnstate.edu/sachin

Tennessee State University Cyber Security Research Lab

Research Team

- **Team:** Faculty members from Electrical and Computer Engineering and Computer Science. 10 PhD, 5 Master and 5 undergraduate students
- **Director:** Sachin Shetty
  - PhD Old Dominion University, Modeling and Simulation
  - Associate Professor, Department of Electrical Computer Engineering
  - Director, Cyber Security Laboratory
  - Associate Director, TIGER Institute
  - Engineer, Naval Surface Warfare Center

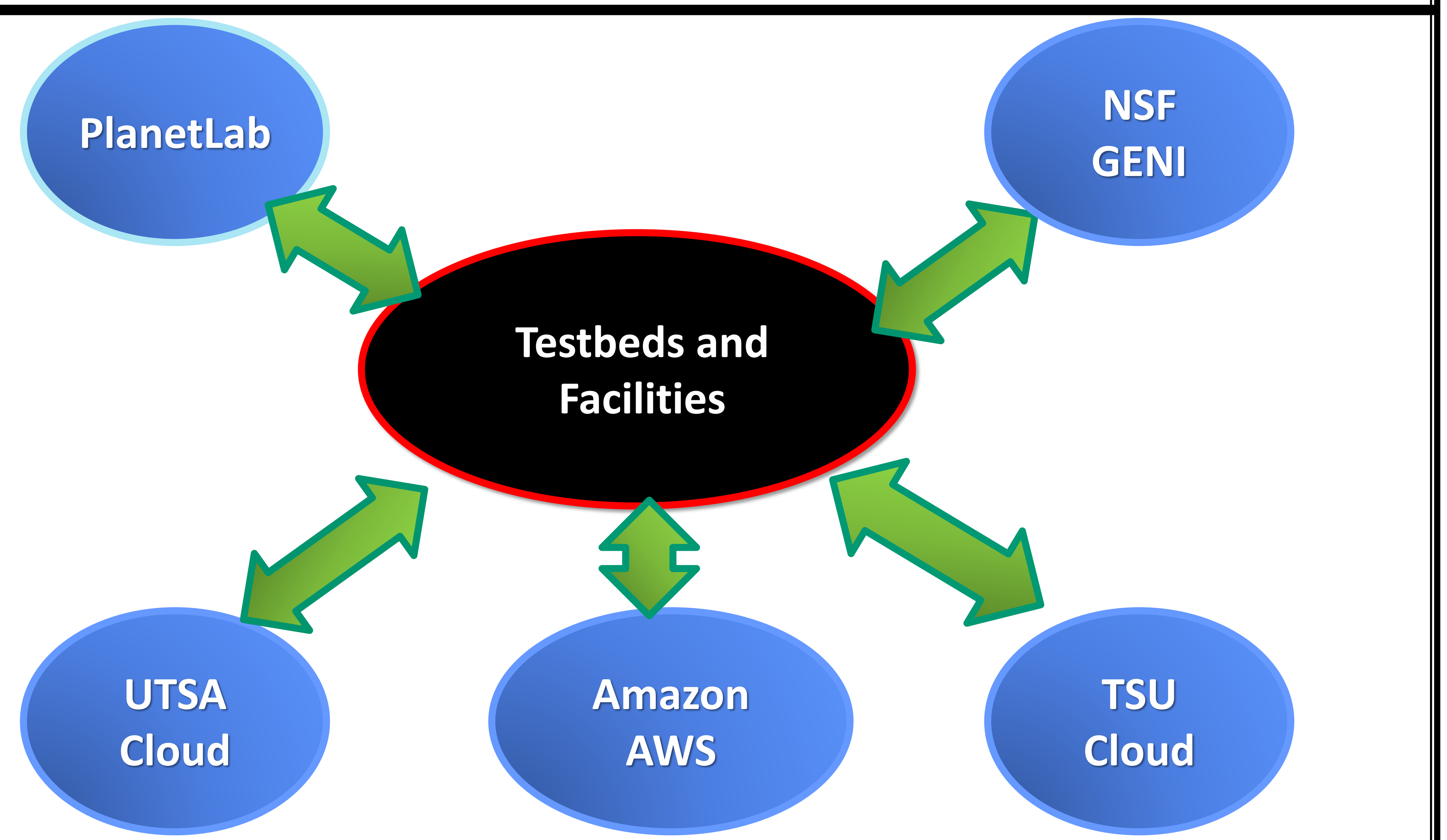
Accomplishments

- Developed IP geolocation technique to accurately locate data in cloud data centers
- Developed technique to assess the security of carrier network in commercial cloud computing environments
- Designed control-theoretic and machine learning based anomaly detection technique to improve accuracy of IDS
- Developed game-theoretic approach to allocate spectrum on cognitive radio cloud network
- Designed cloud centric solution to detect search engine poisoning attacks on smartphones

Funded Research Projects

- **Moving Target Defense:** Heterogeneous VM replication technique for cloud IDS (AFOSR)
- **Security Metrics:** Understanding and quantifying security risks associated with outsourcing cloud data (DHS, AFRL and NSF)
- **Network Anomaly Detection:** Statistical models for anomaly detection of cloud data center traffic (AFRL and DHS)
- **Smartphone Security:** Cloud centric machine learning classifier to detect search engine poisoning attacks (AFRL)
- **Cognitive Radio Cloud Network:** Secure spectrum allocation in cloud computing environment (NSF)
- **Mobile Cyber Physical System Security** : Robust and resilient architecture to protect mobile embedded systems (Boeing)

Resources



POC : Dr. Sachin Shetty, Email: sshetty@tnstate.edu, Web: http://cyberviz.tnstate.edu/sachin

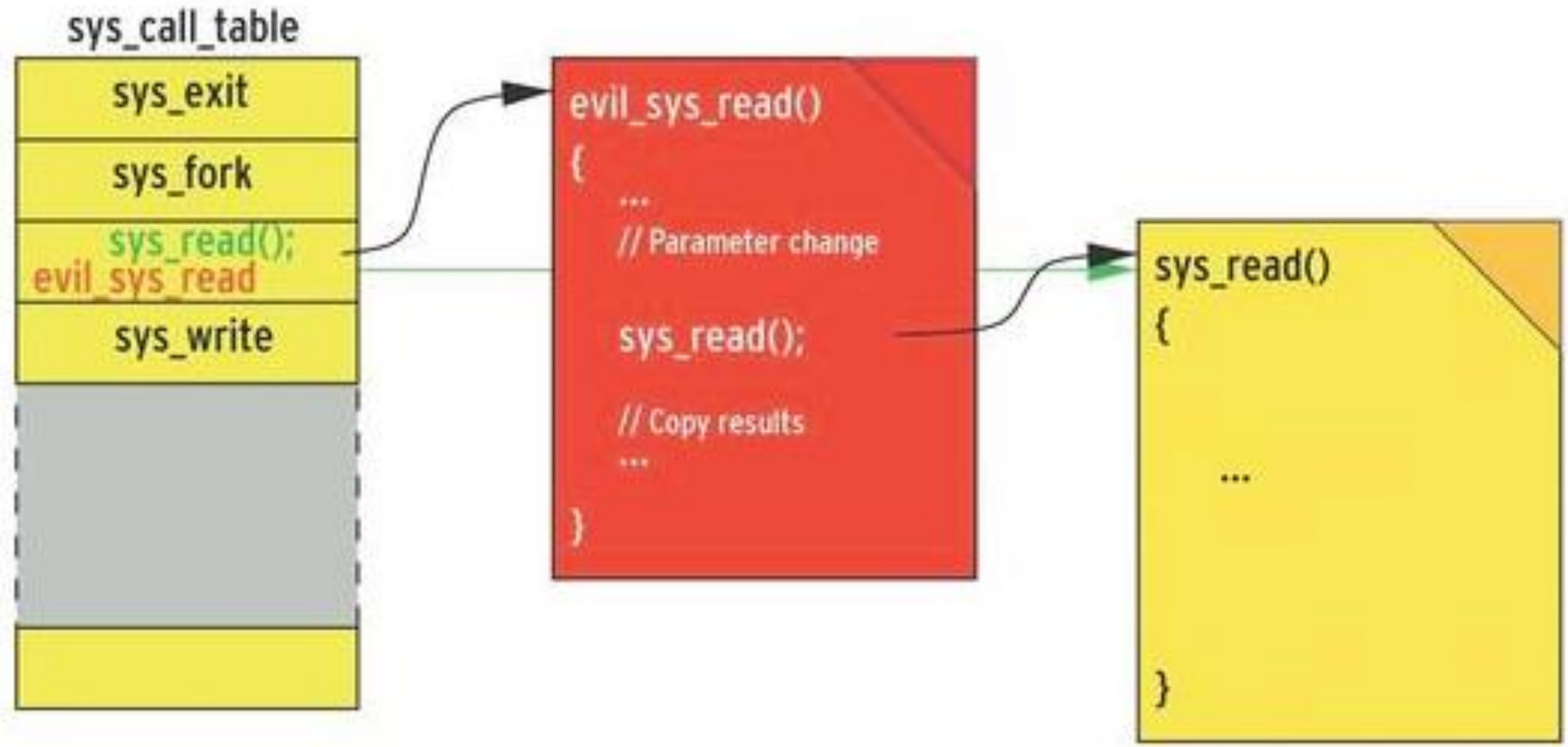
Roadmap

- Introduction
- Background
- Threat Model and Assumptions
- Design of RootkitDet
- Implementation
- Evaluation

Introduction: Kernel Rootkits

- Rootkits allow adversaries to completely control computer
  - Gaining access to Linux kernel
- Why should I care about kernel rootkits?
  - Kernel rootkits can decrypt PGP messages or SSH connections !!
- Rootkits are software which provides attacker camauflaged access
  - User rootkits modify application
  - Kernel rootkits modify kernel data structures and code

System Call Hijacking



Source: Linux Magazine

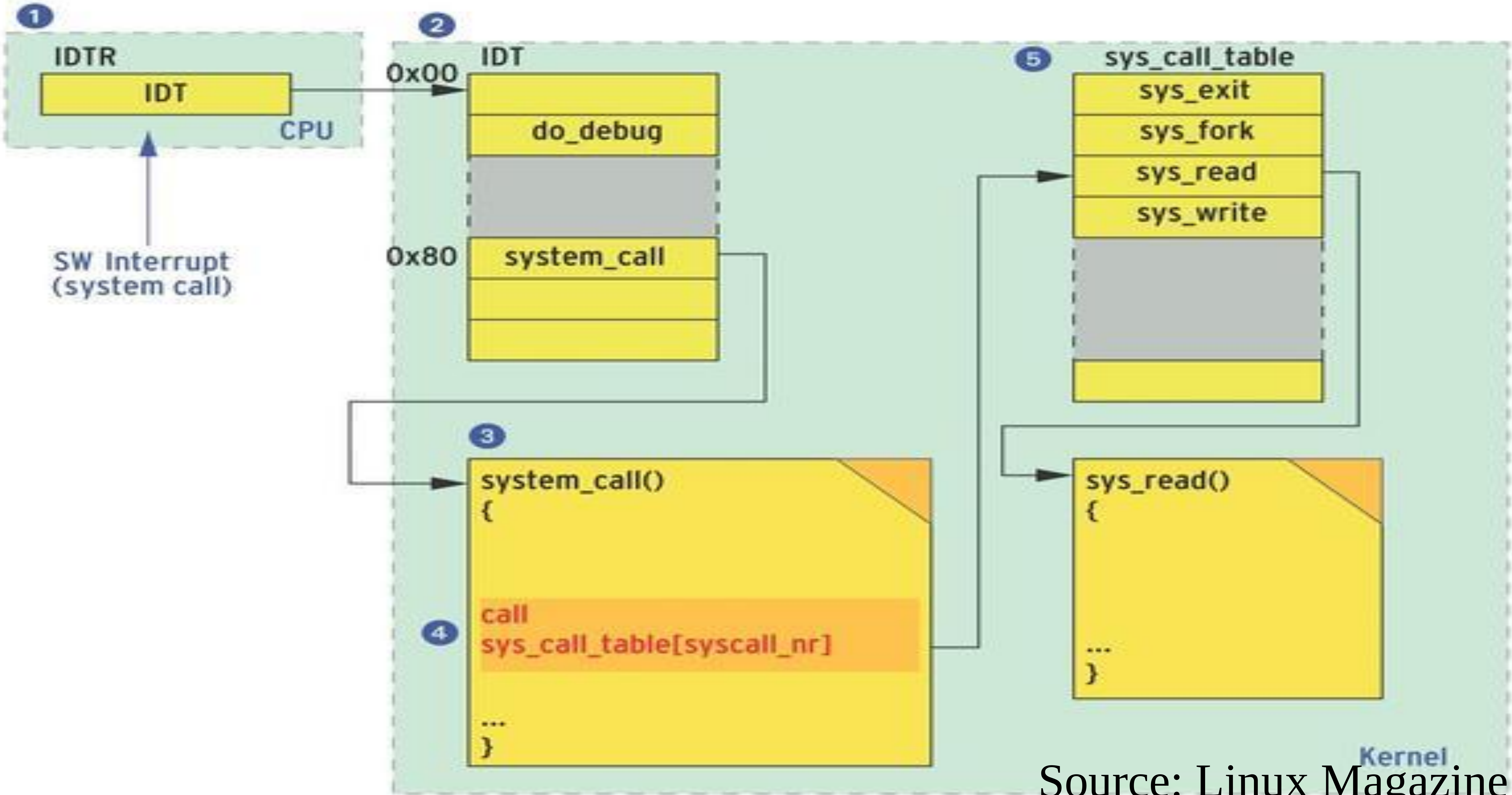
Example of attack code

```
void disable_write_protection_cr0( void )
{
    unsigned long value;

    asm volatile("mov %%cr0,%0":"=r"
        (value));
    if( value & 0x00010000 ) {
        value &= ~0x00010000;
        asm volatile("mov %0,%%cr0"::"r"
                    (value));
    }
}
```

Source: Linux Magazine

In the “unholy” pursuit of the sys call table



Source: Linux Magazine

Kernel Rootkits and Cloud Computing

- Cloud Computing enable ubiquitous access to data and applications
- Cloud security issues have gained traction due to vulnerabilities in low level operating systems and virtual machine implementations resulting in novel denial-of-service attacks [Liu, CCSW 2010]
- Kernel-Level Rootkits has the potential to inflict maximum damage and can launch stealth attacks, which can be difficult to detect or eliminate by the administrator.
- Need for an efficient, scalable and easy to deploy Kernel-rootkits defense system in the cloud

Kernel Rootkits and Cloud Computing

- Kernel rootkit achieves various goals, especially hiding certain malicious processes from security monitoring, anti-virus software, intrusion detection, and VMI (virtual machine introspection).
- Kernel rootkit achieve its goals by modifying certain kernel data structures.
- During the past 10 years, kernel-level rootkits have been emerging as a major security threat.
  - MacAfee AvertLabs reported recently the number of rootkits had increased 600 percent.
  - Rootkits have been leveraged by criminals to conduct bank fraud

Kernel Rootkits and Cloud Computing

- Defending against kernel rootkits in Infrastructure as a Service (IaaS),Platform as a Service (PaaS), and Software as a Service (SaaS)
- In IaaS, kernel rootkits may enable the criminal to keep a backdoor in a VM (virtual machine) for the attacker to gain whole control of the guest operating system.
- They may also hide some other malware which may inflict serious damage or launch stealthy attacks.
- Due to the hiding, this malware can become difficult to detect or eliminate by the administrator.

Kernel Rootkits and Cloud Computing

- In a cloud environment, cloud providers are responsible for countering kernel rootkits in tenant VMs as they can fully leverage the security features of the underneath hypervisors.

- We focus on cloud environments since besides standard requirements such as effectiveness and efficiency, cloud environments have several unique requirements regarding how kernel rootkits should be countered.

Kernel Rootkits and Cloud Computing

- (R1) End-to-end defense.
  - Cloud administrators need to quickly reverse the malicious modifications made by the rootkit to its target VM.
  - Manual diagnose and reverse the malicious modifications would render ineffective availability and business continuity loss
- (R2) Scalable defense.
  - The total defense costshould be linear (if not sublinear) in the number of VMs being simultaneously protected.
  - The defense should also facilitate dynamic addition and deletion of VMs.
- (R3) Adoptable defense.
  - The defense should be compatible with existing commercial (and open source) cloud platforms.

Kernel Rootkits and Cloud Computing

- Existing kernel rootkit defenses are limited. Following is a categorization of current defenses
  - (1A) Detecting modified control or non-control data or violations of invariants .
  - (1B) Preventing installation of kernel rootkits by performing analysis on the code being loaded into the kernel space.
  - (1C) Defending kernel rootkits by cooperating with anti-malware software.
  - (1D) Protecting the kernel by restoring infected kernels to healthy state.

Kernel Rootkits and Cloud Computing

- We may briefly summarize the limitations of these defenses in terms of the requirements as follows.
- (a) Defenses in Class 1A and 1C are not end-to-end or focus only on control data.
- (b) Defenses in Class 1B and 1D might be defeated by the rootkits leveraging novel techniques or kernel vulnerabilities and some of them are not very easy to be adopted because they are designed based on special hypervisors.
- (c) Defenses in Class 1C are not very scalable because they have to create multiple instances of anti-malware software to monitor multiple VMs.

Kernel Rootkits and Cloud Computing

- Class 1A efforts
  - *Petroni, CCS '07* detects persistent kernel control-flow attacks by identifying function pointers in kernel data structures to the kernel and modules' code.
  - *Wang, CCS '11* protects thousands of kernel hooks in a guest OS from being hijacked.
  - *Baliga et. al., DSC '11* detects kernel rootkits by identifying data invariants in the kernel.
  - *Petroni*, *Usenix '06* focuses on semantic integrity violations in kernel dynamic data.
  - While these works focus on control/non-control data in the kernel, our system focuses on the code inserted into kernel space and attempts to perform recovery.

Kernel Rootkits and Cloud Computing

- Class 1B efforts
  - *C. Kruegel et al ACSCAC '07* performs static analysis on the module that is being loaded and prevents it if it resembles the behavior of rootkits.
  - *Garfinekel, NDSS '03* protects the guest OS kernel code and critical data structures from being modified.
  - *Seshadri, SIGOPS '07*, presented a tiny hypervisor that enforces page-level protection of the memory used by the code of the kernel and modules, prevents the installation of the kernel rootkits by ensuring the code integrity
  - *Riley, RAID '08* protects the code integrity in the guest OS kernel by routing guest kernel instruction fetches to shadow memory which contains authenticated code and is protected from writeaccess.
  - However, they are not easy to be adopted in the cloud platforms as the design considerations are tightly wedded to the hypervisor.

Kernel Rootkits and Cloud Computing

- Class 1C efforts
  - *Payne, SP '08* detects malware by providing semantic view of the guest OS to anti-malware software
  - *Jiang, CCS '07* presents an architecture that gives the security tools the ability to do active monitoring.
  - While they are cooperated with external security tools or anti-malware software, our system can defend rootkits alone and monitor more VMs with less effort.
- Class 1D efforts
  - *Fraser, ACSAC '08* applies different repair techniques to restore the infected kernel to healthy state after detection
  - However, it can be defeated by rootkits that insert new control-data in the kernel space

Kernel Rootkits and Cloud Computing

- RootkitDet, an end-to-end defense against kernel rootkits in a cloud environment.
  - Detects the kernel rootkits by looking for suspicious code in the kernel space of the guest OSes.
  - Diagnosis to precisely identify kernel data structures that were maliciously modified by the rootkit.
  - Attempts to reverse the modifications.
  - Registration procedure can be leveraged to enable separation between legitimate code and rootkit code in the kernel space.
  - RootkitDet attempts to eliminate the effect of the rootkit.
  - RootkitDet first performs static analysis on the suspicious code to collect certain characteristic information of the rootkit.

Kernel Rootkits and Cloud Computing

- Categorize the rootkit heuristically according to the collected characteristic information.
- If the rootkit can be categorized, RootKitDet would be able to identify the kernel data structures that were malicious modified by the rootkit.
- Finally, RootkitDet reverses the modifications as follows: it restores the modified control data with pre-known values, and recovers the broken links between the modified non-control data and other data structures.

Kernel Rootkits and Cloud Computing

- Problem Setting
  - User accesses service provided by cloud user
  - Vulnerabilities exist in application service and/or guest OS
  - Attacker may gain root privilege, install a kernel-level rootkit to launch stealth attack on VM
- Requirements of defense system
  - End-to-end defense
  - Light-weight / low overhead
  - Scalable
  - Easy to use

Agenda

- Introduction
- Threat Model and Assumptions
- Background
- Design of RootkitDet
- Implementation
- Evaluation

Threat Model and Assumptions

- Threat model
  - External attacker installs rootkit in the OS kernel managing VMs by exploiting zero-day vulnerabilities in kernel and application software in VMs
  - Attacker gains control over multiple VMs to steal confidential data, modify memory, etc
- Assumptions
  - CPU supports NX-bit(Non-executable) feature, and Linux kernel utilizes this feature for memory protection
  - Kernel-level rootkits insert code into kernel space
  - VMM is not affected by rootkits