SVGrid: A Secure Virtual Environment for Untrusted Grid Applications

Xin Zhao, Kevin Borders, and Atul Prakash
University of Michigan
{zhaoxin,kborders,aprakash}@umich.edu

Motivation

- Grid computing requires the sharing of geographically distributed "autonomous" resources such as CPU time, disk spaces and network bandwidth.
- However, these resources could be abused by malicious users.

Threat Model

- Trust is not necessarily transitive. Access via Grid software opens up another channel for NetBots or other forms of attack.
- OS or applications may have vulnerabilities that can be exploited.
  - intruders wish to modify critical system files or maliciously access network from the victim host
    - Steal confidential information (e.g., passwords)
    - Taint system logs to hide intrusion trails,
    - Create back doors and Trojan horses to hide their presence and retain access to the machine
    - Mount further attacks against other online grid computers.

We assume that attackers do not have physical access to target machines but have remote access.

Properties of SVGrid

- Transparency.
  - Resources protected by SVGrid (filesystem and network) can be transparently accessed by grid applications without modification on application source code.
- Complete mediation.
  - All accesses to protected resources must go through the monitor virtual machine and are subject to security checking.
- Attack resistance.
  - The protection mechanism itself is resistant to attacks and hard to be disabled by malicious grid applications

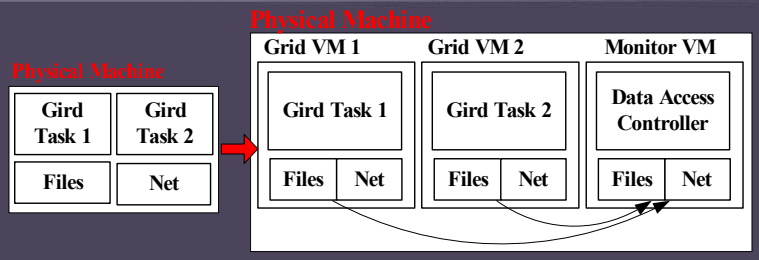
Can Traditional Protection Mechanisms Achieve This Goal? --Sensitive File Protection

- Most of them run in the target operating systems, which themselves are susceptible to network attacks.
  - E.g., Mandatory Access Control (e.g., SELinux)
  - Intrusion Detection/Prevention Systems (e.g., Snort, ISS RealSecure Server Sensor)
    - Can be disabled if the OS kernel is compromised;
    - Passive protection; do not prevent malicious accesses to sensitive data.
  - File Integrity Checker (e.g., virus scanner, Tripwire)
    - Can be bypassed if OS is compromised;
    - Passive protection; do not prevent malicious accesses to sensitive data.

Can Traditional Protection Mechanisms Achieve This Goal? --Network Protection

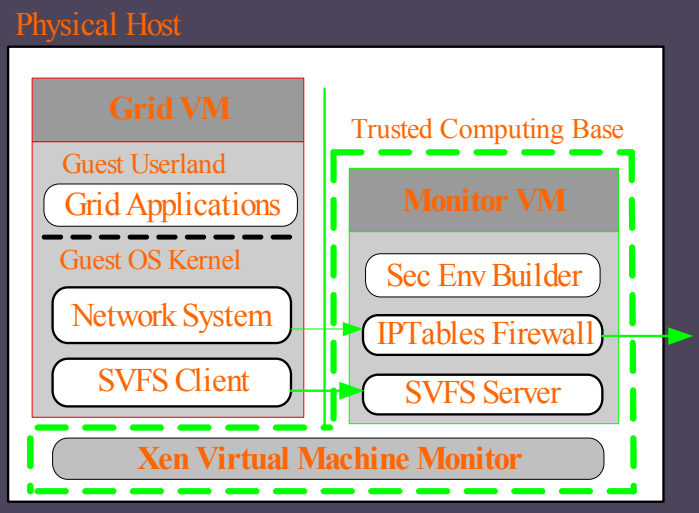
- Firewall is the common solution, However…
- If deployed in the target host,
  - it can be turned off after an attacker gains system privileges
- If deployed in a standalone device such as router,
  - Hard to apply different policies for different grid tasks running on the same physical host.

Our Solution--Basic Idea



- Leverage VM Technology to isolate grid tasks into multiple virtual machines
- Move data and network resources to a dedicated monitor VM and enforce security policy independently in the monitor VM
- Achieve non-bypassable, security control on file accesses; Even successful intruders in standard VMs cannot disable data access controller.

SVGrid Architecture



Secure Environment Builder

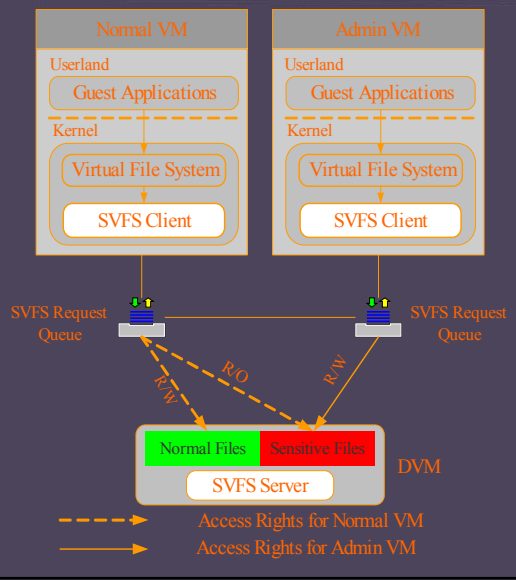
- Exporting appropriate filesystem to the new GVM
- Loading security policies into the SVFS server and network firewall

A Sample Policy

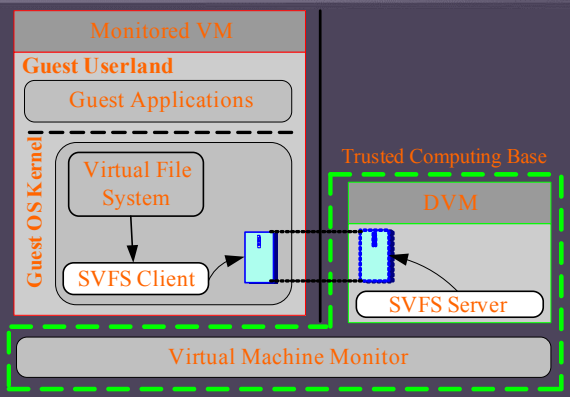
```
# Sample Customized Policies.  Shows the options
user {abc}              # the policy is customized for grid user "abc"
#Network Description
IP {
  a.b.c.d/netmask       # abc is allowed to access host/subnet a.b.c.d/netmask
}
bind {
  80                                # app can bind to port 80,
  443                               # and port 443
}

#File Access Description
open_rw {                           # app can read these files
  /path/to/file1
  /path/to/file2
 /path/to/dir/*
}
open_a  { }                         # app can append to these files
unlink  { }                         # app can unlink these files
```

Detailed Design of SVFS



VM based RPC (VRPC)



- RPC eases inter-VM cooperation
- Shared memory based communication reduces the cost of cross domain data exchange

Network Access Control

- IPtables is deployed in the monitor VM
- Monitor VM knows information (MAC, IP) of the network interface of Grid VM.
- Monitor VM maps the source VM to (MAC, IP). Source VM ID is identified by VMM and cannot be forged, preventing IP spoofing

Evaluation on Prototype

- Security Evaluation
  - Successfully prevent multiple malicious softwares such as Lrk5 (malicious file access), mworm (malicious network access)
- Performance Evaluation
  - Time used for linux kernel build
    - on SVGrid, 284.65 seconds
    - On a native Linux box, 262.80 seconds
    - Performance penalty is 8.3%

Questions?

- Contact info:
  - zhaoxin@edu.umich.edu
  - aprakash@edu.umich.edu

Xen based Virtual Machine Environment