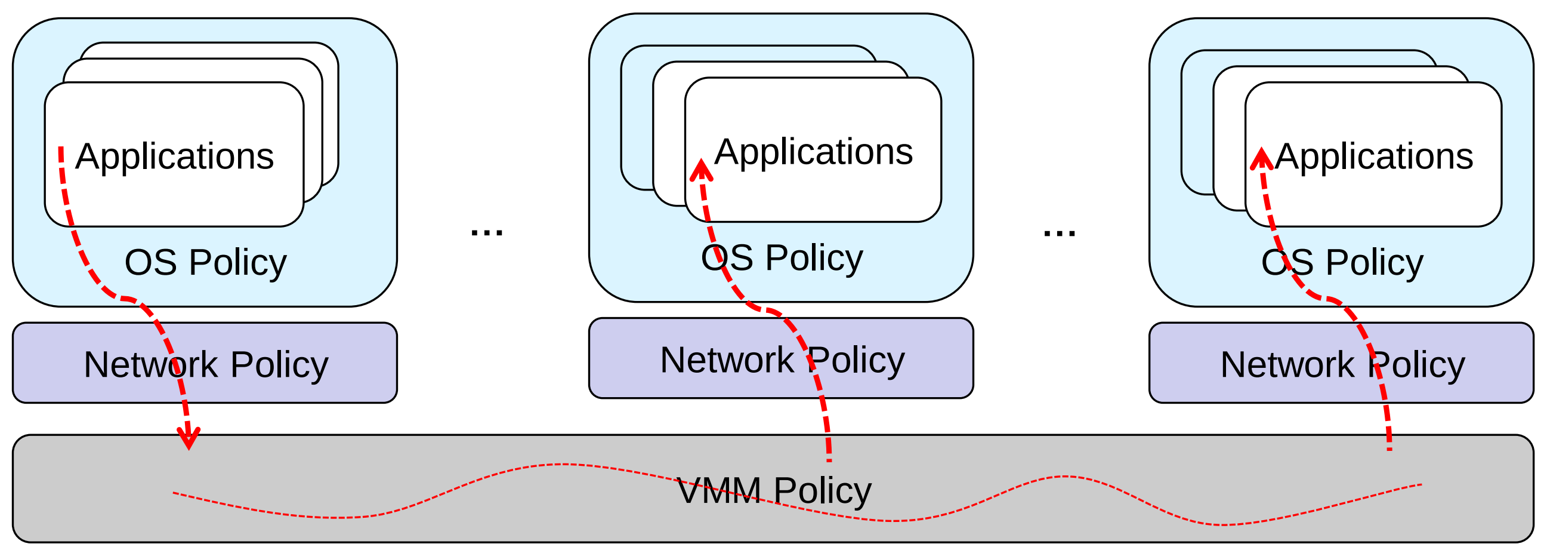
Evaluating Security Policy Compliance in Virtual Machine Environments

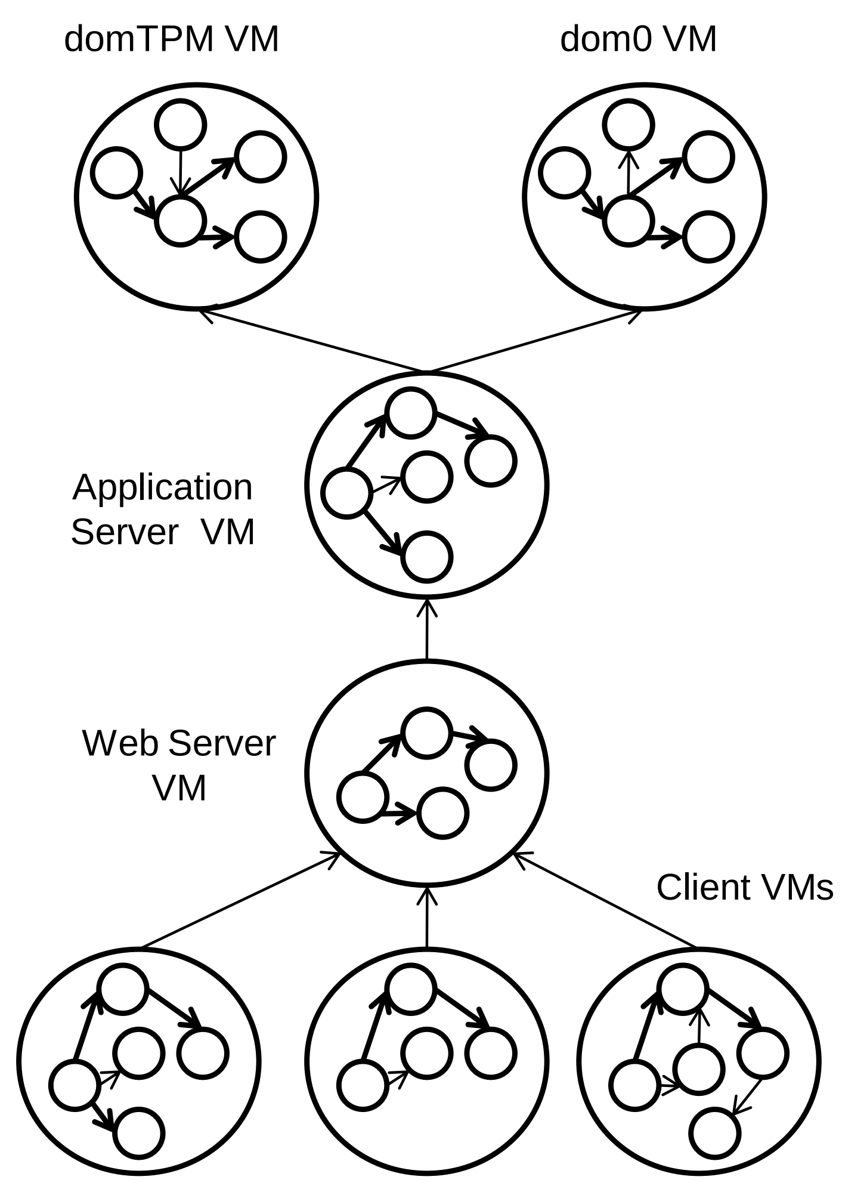
Sandra Rueda, Hayawardh Vijayakumar, Trent Jaeger, Patrick McDaniel

- A reference monitor for virtual machine monitors (VMM) was recently developed. When combined with MAC enforcement present in the virtual machines (VMs), we can have comprehensive MAC enforcement.
- *Since these policies are independently developed and enforced, there is no guarantee about their compliance, as a whole, with a global security goal.*
- We developed a formal definition of compliance based on information flows, and a tool to evaluate, whether this kind of system, as a whole, is compliant with a global security goal or not.

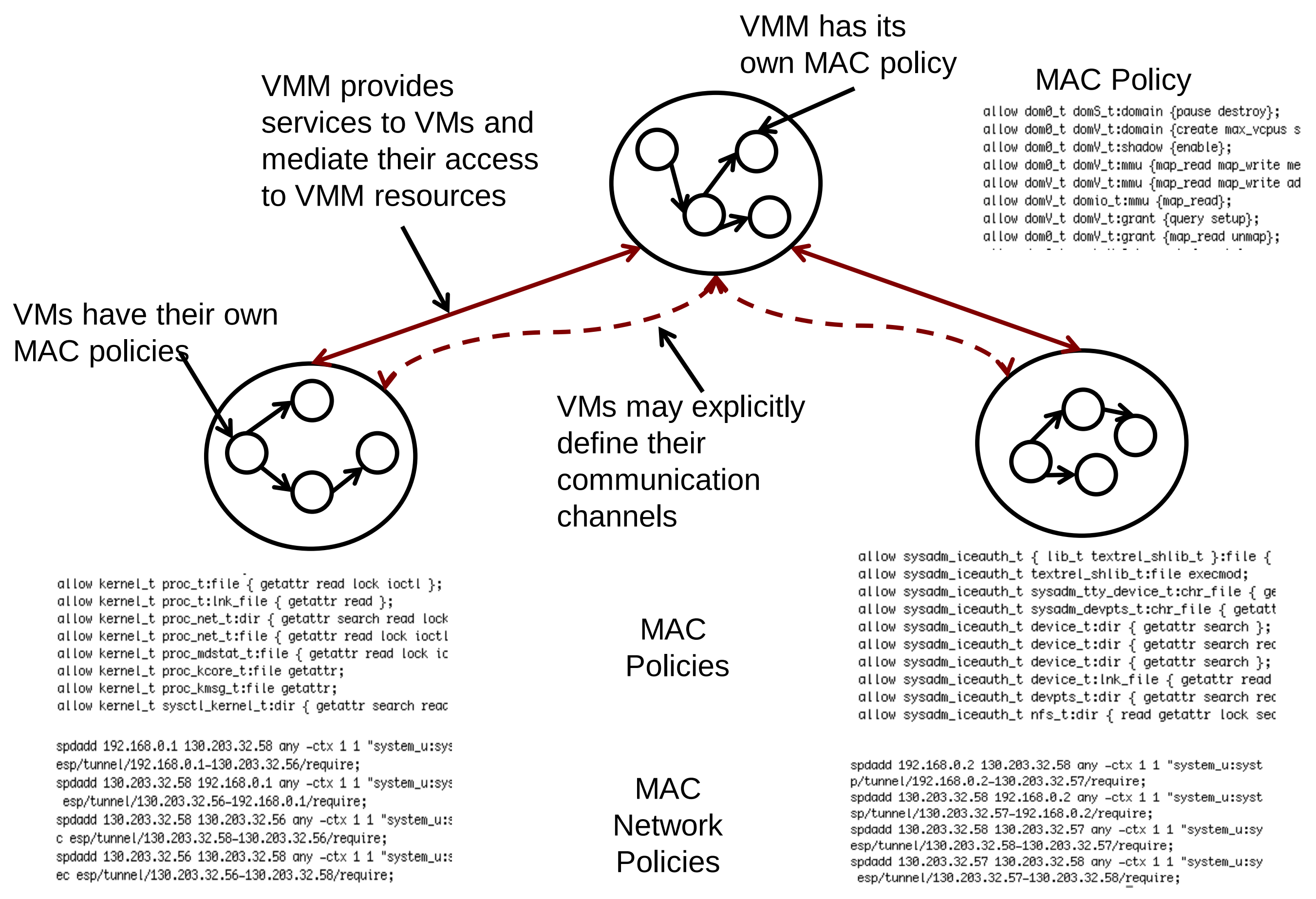


VM-Systems

Configuration



VM-systems have a single VMM policy and several OS and network MAC policies.


VMM has its own MAC policy
VMM provides services to VMs and mediate their access to VMM resources
MAC Policy
VMs have their own MAC policies
VMs may explicitly define their communication channels
MAC Policies
MAC Network Policies

```
allow dom0_t domS_t:domain {pause destroy};
allow dom0_t domV_t:domain {create max_vcpus s
allow dom0_t domV_t:shadow {enable};
allow dom0_t domV_t:mmu {map_read map_write me
allow domV_t domV_t:mmu {map_read map_write ad
allow domV_t domio_t:mmu {map_read};
allow domV_t domV_t:grant {query setup};
allow dom0_t domV_t:grant {map_read unmap};
```

```
allow kernel_t proc_t:file { getattr read lock ioctl };
allow kernel_t proc_t:lnk_file { getattr read };
allow kernel_t proc_net_t:dir { getattr search read lock
allow kernel_t proc_net_t:file { getattr read lock ioctl
allow kernel_t proc_mdstat_t:file { getattr read lock ic
allow kernel_t proc_kcore_t:file getattr;
allow kernel_t proc_kmsg_t:file getattr;
allow kernel_t sysctl_kernel_t:dir { getattr search reac
```

```
spdadd 192.168.0.1 130.203.32.58 any -ctx 1 1 "system_u:sys
esp/tunnel/192.168.0.1-130.203.32.56/require;
spdadd 130.203.32.58 192.168.0.1 any -ctx 1 1 "system_u:sys
 esp/tunnel/130.203.32.56-192.168.0.1/require;
spdadd 130.203.32.58 130.203.32.56 any -ctx 1 1 "system_u:s
c esp/tunnel/130.203.32.58-130.203.32.56/require;
spdadd 130.203.32.56 130.203.32.58 any -ctx 1 1 "system_u:s
ec esp/tunnel/130.203.32.56-130.203.32.58/require;
```

```
allow sysadm_iceauth_t { lib_t textrel_shlib_t }:file {
allow sysadm_iceauth_t textrel_shlib_t:file execmod;
allow sysadm_iceauth_t sysadm_tty_device_t:chr_file { ge
allow sysadm_iceauth_t sysadm_devpts_t:chr_file { getatt
allow sysadm_iceauth_t device_t:dir { getattr search };
allow sysadm_iceauth_t device_t:dir { getattr search rec
allow sysadm_iceauth_t device_t:dir { getattr search };
allow sysadm_iceauth_t device_t:lnk_file { getattr read
allow sysadm_iceauth_t devpts_t:dir { getattr search rec
allow sysadm_iceauth_t nfs_t:dir { read getattr lock sec
```

```
spdadd 192.168.0.2 130.203.32.58 any -ctx 1 1 "system_u:syst
p/tunnel/192.168.0.2-130.203.32.57/require;
spdadd 130.203.32.58 192.168.0.2 any -ctx 1 1 "system_u:syst
sp/tunnel/130.203.32.57-192.168.0.2/require;
spdadd 130.203.32.58 130.203.32.57 any -ctx 1 1 "system_u:sy
esp/tunnel/130.203.32.58-130.203.32.57/require;
spdadd 130.203.32.57 130.203.32.58 any -ctx 1 1 "system_u:sy
 esp/tunnel/130.203.32.57-130.203.32.58/require;
```

Challenges

- Goals are ambiguous

  Administrators configure policies, not goals

  Classical information flow goals are too restrictive

  We need to relate policies to goals

  Complete manual specification is impractical
- VM-Systems are complex

  They involve multiple policies (VMM, OS, network)

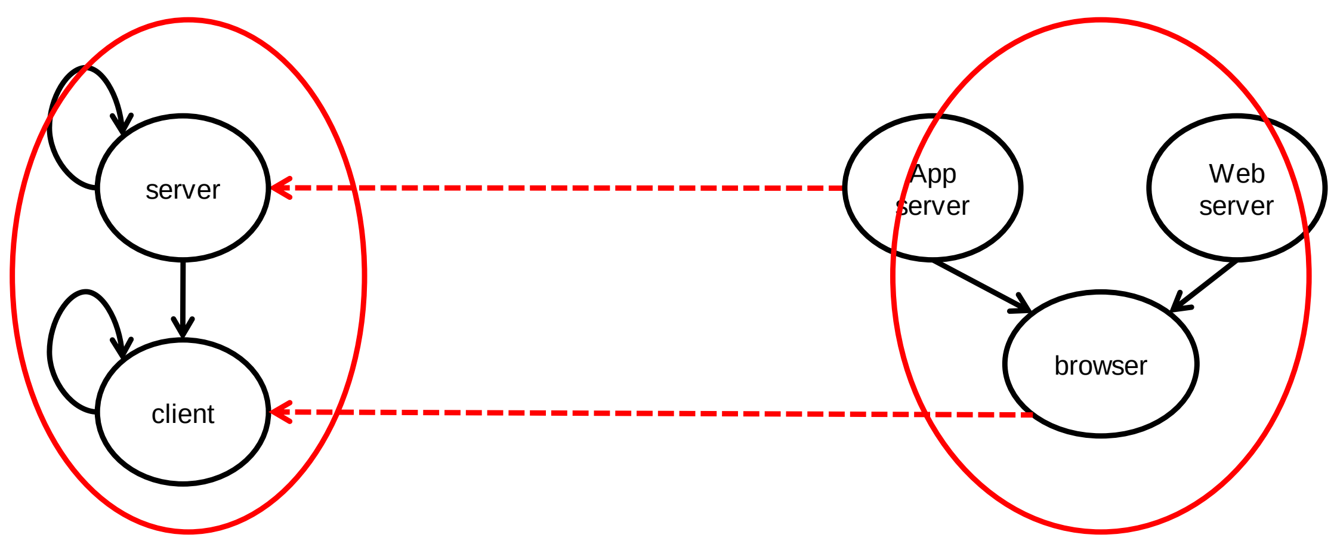
  Composing all policies into a single graph would prevent effective analysis

Compliance

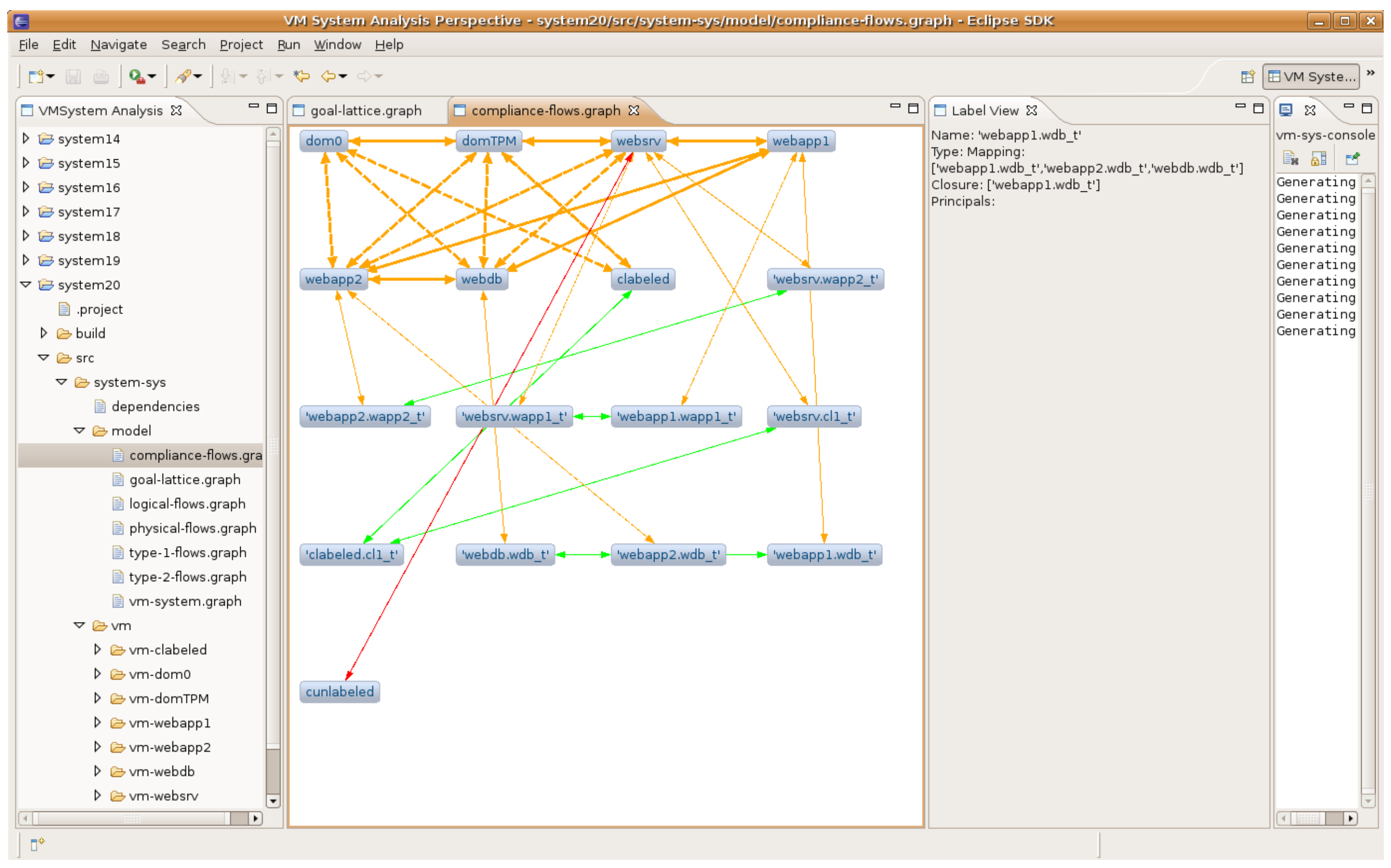
A VM-system is compliant with an information flow security goal, if all the flows enabled by the policies are authorized by the goal.

Integrity Goal - L
(Information Flow Security Goal)

Information Flow Graph - G
(MAC System Policy)



Analysis Tool



```
Name: 'webapp1.wdb_t'
Type: Mapping:
['webapp1.wdb_t','webapp2.wdb_t','webdb.wdb_t']
Closure: ['webapp1.wdb_t']
Principals:
```

```
Generating
Generating
Generating
Generating
Generating
Generating
Generating
Generating
Generating
Generating
```

We developed an Eclipse plug-in to define VM-systems, load their policies, evaluate compliance, display results, and suggest options to resolve conflicts. The purpose of the tool is to assess administrators in the configuration of a secure system.

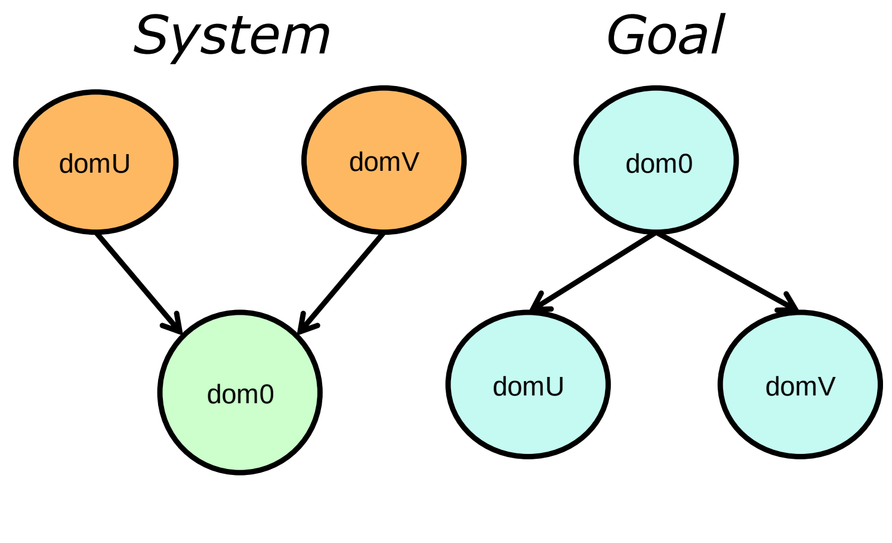
Approach

Problem: how to automatically deduce goals and map policy labels to goal labels.

- Use the VM-system configuration to obtain security goals. There is domain knowledge in the VM configuration that we can use to deduce default conservative goals. We also enable administrators to make refinements.
- Use top-down view of the system to evaluate information flow compliance. We do not need to integrate all the policies into a single information flow graph. Instead, we use inter-VM flows first to assess information flows before we assess flows within VMs.
- Analyze internal information flows if conflicts arise at the higher level view. We also define a comprehensive set of possible resolutions.

We use system domain knowledge and an iterative refinement to deduce goals, to automatically map policy labels to policy goals, and to evaluate compliance.

- A system configuration defines domain information that implies security requirements. For instance: if a VM domU depends on a VM dom0 then dom0's integrity must be higher than domU's integrity.



Publications

S. Rueda, H. Vijayakumar, T. Jaeger. Analysis of Virtual Machine System Policies. ACM SACMAT 2009.

S. Rueda, D. King, T. Jaeger. Verifying Compliance of Trusted Programs. USENIX Security Symposium 2008.

S. Rueda, Y. Sreenivasan, T. Jaeger. Flexible Security Configuration for Virtual Machines. ACM Computer Security Architecture Workshop. CSAW, 2008.

Sponsored by AFRL