DFG/Geni Workshop

Disposable Virtual Machines
Security cluster

Hyojoon Kim
Ilker Ozcelik
Alexander Marold
Oliver Hanka

The Research Problem

- Today's network suffers from variety of attacks and malicious behaviors
  - DDoS
  - Password cracking
  - SQL injection
  - …

- Can we find an architectural solution which can fundamentally solve the problem in the future Internet?

Why Important?

- We want the Internet to be:
  - Secure
  - Reliable
  - Resilient
  - Available

  ..

  However, current architecture of the Internet is not able to fully support this!

Different Type of Attacks

- Single-source attacks
  - User is the attacker
  - Attacks with specific (recognizable) pattern
  - Possible to detect at the target (e.g. pattern matching)

- Distributed attacks
  - User is unaware that his/her machine is performing attacks
  - Machine is part of a botnet (e.g. DDoS)
  - Machine may only send one or two packets
  - Could affect user experience
  - “Hard” to detect at the target (malicious or legit traffic?)

Our Approach

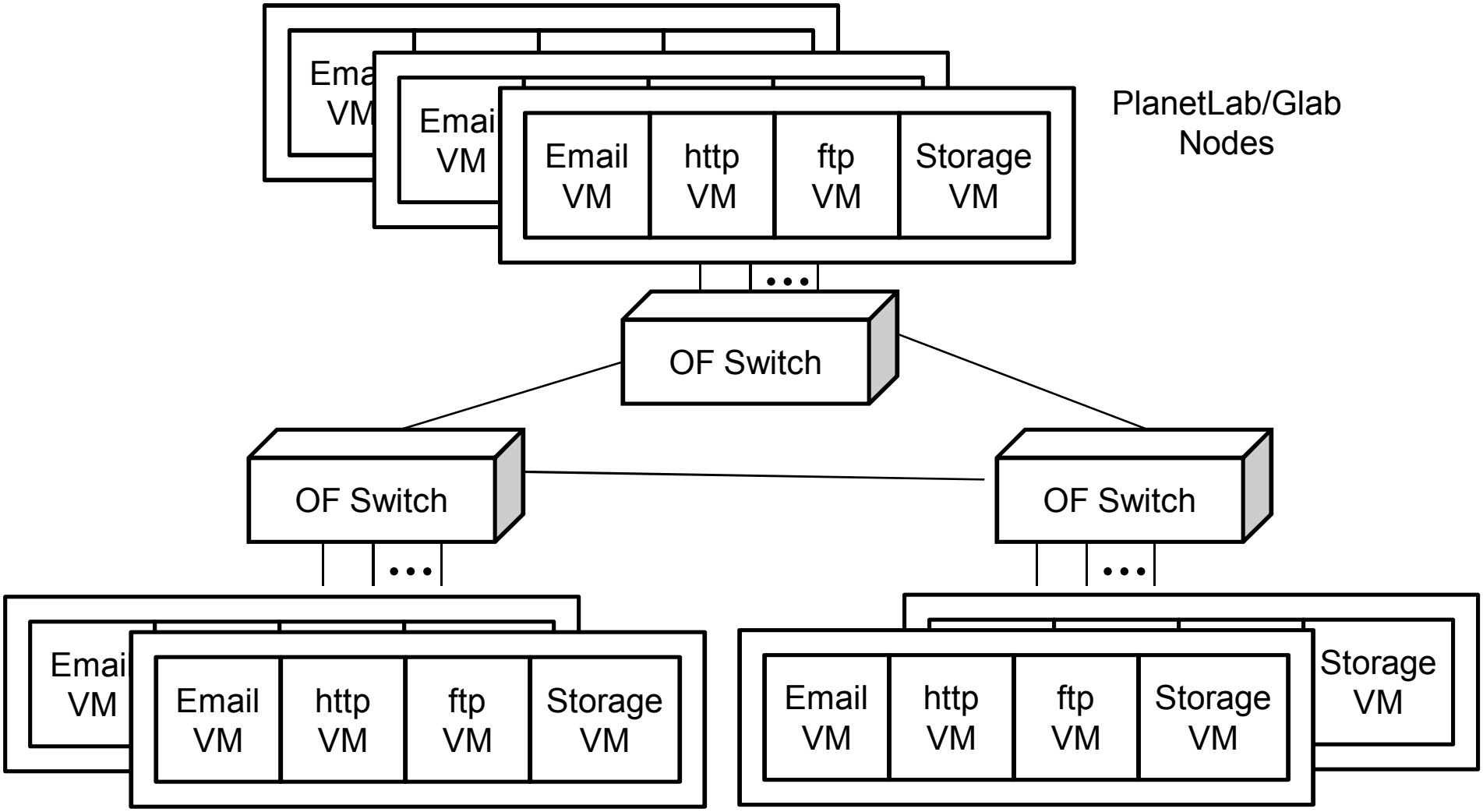
- Address the problem at the source
- Disposable VM approach
  - Each network application runs in its own disposable virtual machine
  - After using the program/machine, it gets destroyed and a new instance is created
- Benefits
  - Deletes downloaded malicious code
  - Reduces malicious traffic generation in the Internet
  - Possible to block a single malicious VM without affecting other applications
  - Possible to have your application “on the go” (e.g. firewall, browser, video chat)

Experiment Requirements & How GENI/Glab can Help

- **Experiment requirements**
  - Novel addressing scheme
    - Locator/Identifier split
    - Accountable identifier
  - Possibility to run “lots” of VMs

- **GENI/Glab resources**
  - PlanetLab nodes & Glab nodes
    - End host with multiple VMs running
  - OpenFlow switches
    - To use different address schemes other than IP
    - To interconnect end hosts

Experiment Setup



Experiments

- Evaluate effectiveness of disposable VMs
  - Inject a variety of attacks, and compare the “current system” and “disposable VM system”
  - Find the appropriate lifetime of each VM
    - Measure “damage” in the network (e.g. SPAM messages send)
    - Increasing time of VM lifetime (5min, 30min, 1h, …)
  - Requirement: as many nodes as possible

- User experience (Prototype)
  - Average load time of the VMs
  - Customization of the VMs
  - Requirement: a few nodes

Challenges of Disposable VMs

- Performance
- Personalization
  - How about browser customization?
  - Plugin download
  - Is the repository safe? / Plugin not malicious?
  - Possible solution: Monitoring the behavior of each new piece of software
- Storage of data
  - Possible solutions:
    - VM for data storage
    - Data are scanned when passed from one VM to another
- User friendly
  - Should ***not*** drastically change how people do every computing

Got funding? :-)