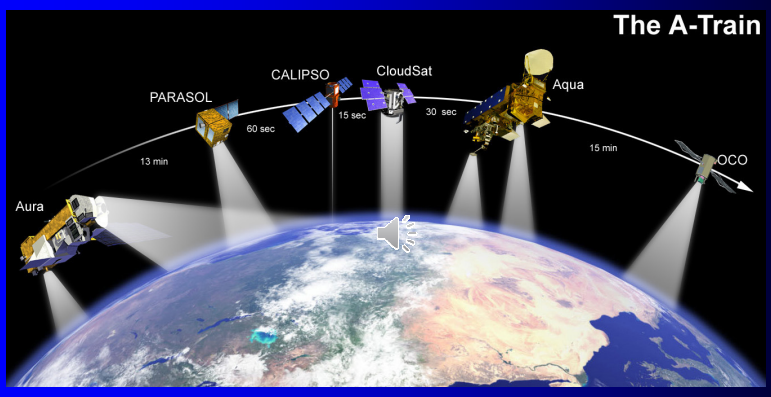
Cyber Defense Research Laboratory

laying the groundwork

David Morgan

Jet Propulsion Laboratory
August 16, 2013

They're talking!



http://www.nasa.gov/topics/earth/features/atrain_climate.html

… but who to ??

Ground Data Systems (GDS)

- receivers of satellites' transmissions
- they're computers
  - cpu
  - memory
  - software
  - storage
  - interfaces

- ➔ vulnerable !!

Department objectives – long-term

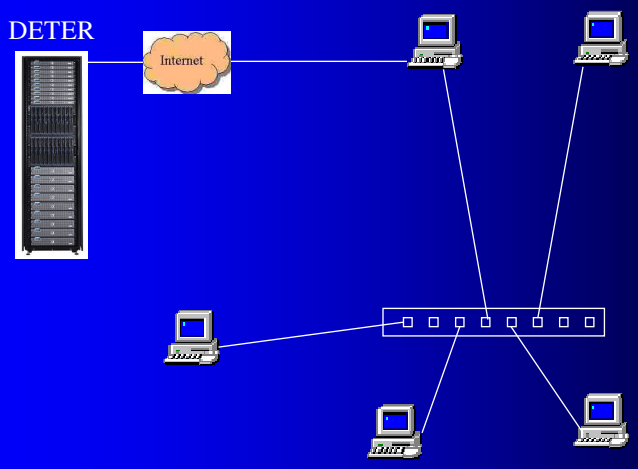
- *build* a testbed lab
  - physical and virtual machines
- *isolate* it
- *provision* it with tools
  - traffic generators
  - performance/vulnerability analyzers
  - sniffers/spoofers/crackers
- *replicate* target systems locally
- *test* them

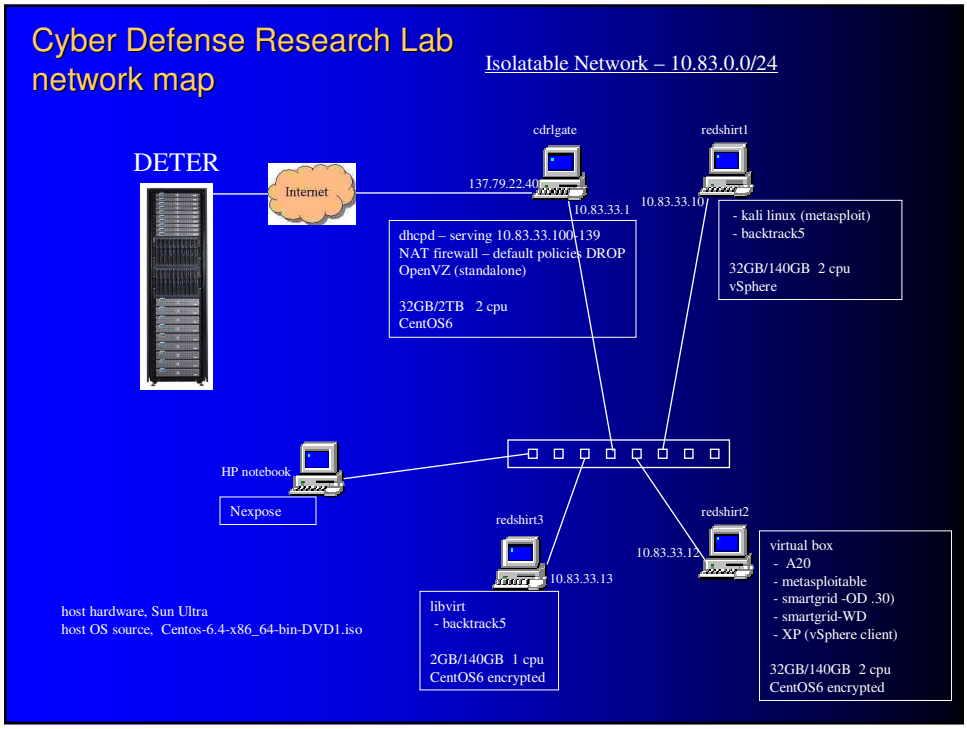
Summer project objectives – short-term

- build a springboard foundation
  - as much of the grand vision as feasible
  - refining start plan as we gain experience over 10 wks
  - build in capabilities for future scale-up
- operate its components

Cyber Defense Research Lab network map

Isolatable Network – 10.83.0.0/24





Basics - dhcp

- dispensing addresses within 10.83.0.0/16
  - enough to scale to as many IPs as desired
- PM addressing static
- VM addressing
  - from dhcp if configured for "bridged" (in its VM product)
  - independent if configured for "NAT"

Basics - gateway firewall

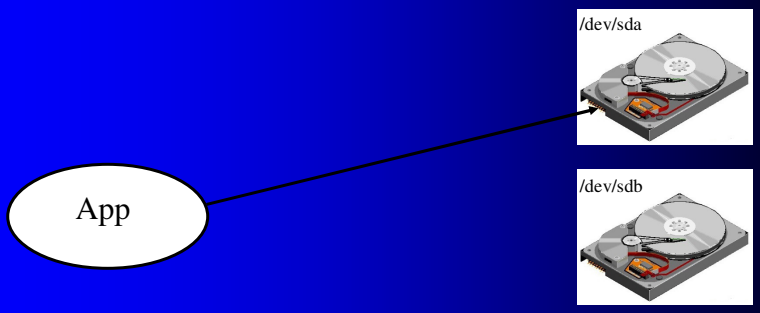
"You can do nothing unless I say so."

- a short script using "iptables"
- default policy "allow no packet passage"
  - no outbound ← unusual
  - no inbound
  - no pass-thru
- selective exceptions
- contain and control data

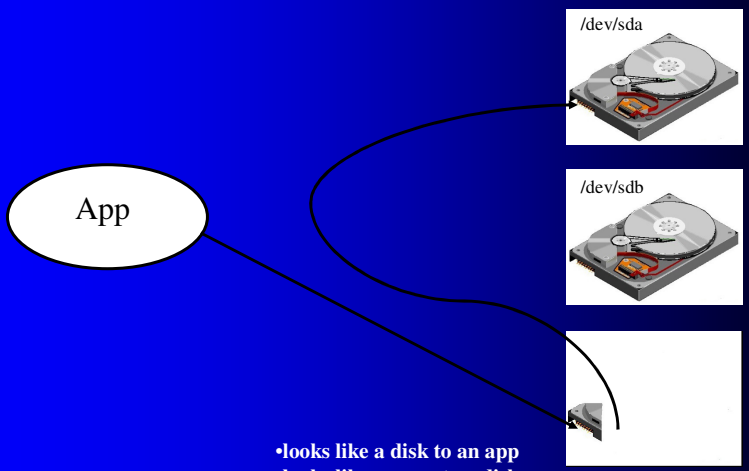
Data confidentiality

- encrypt data not in use
- when is data not in use?
  - on disk – "data at rest"
  - in transit – "data in motion"
- what are the techniques?
  - disk encryption
  - encrypted tunneling

Using hardware disk devices



Using software disk devices



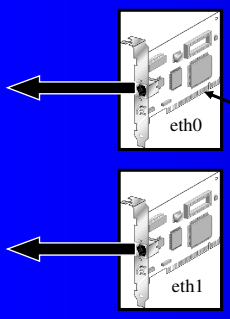
- looks like a disk to an app
- looks like an app to a disk
- gets to massage (e.g. encrypt) traffic passing through

Concretely: linux's dm_crypt

App

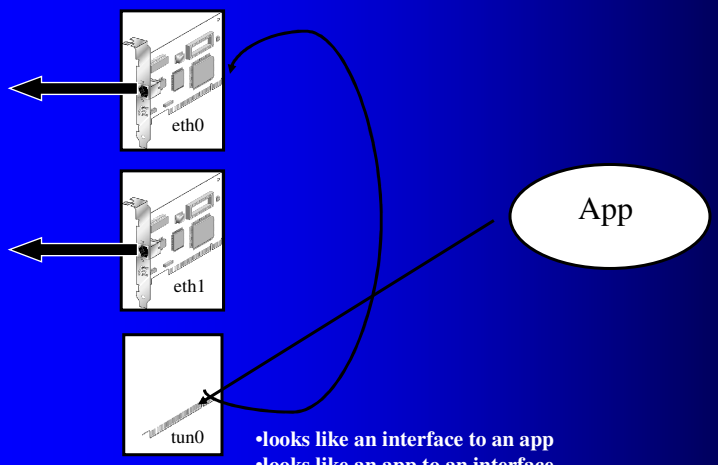
```
root@cdrlgate:~
File  Edit  View  Search  Terminal  Help
[root@cdrlgate ~]# ls /dev/sd*
/dev/sda  /dev/sda1  /dev/sda2  /dev/sdb  /dev/sdc  /dev/sdd  /dev/sde
[root@cdrlgate ~]# ls /dev/mapper/sd*
ls: cannot access /dev/mapper/sd*: No such file or directory   inserted USB drive
[root@cdrlgate ~]# ls /dev/sd*
/dev/sda  /dev/sda1  /dev/sda2  /dev/sdb  /dev/sdc  /dev/sdd  /dev/sde  /dev/sdf
                                                                          New!
[root@cdrlgate ~]# ls /dev/mapper/sd*
ls: cannot access /dev/mapper/sd*: No such file or directory
[root@cdrlgate ~]# ls /dev/dm*
/dev/dm-0  /dev/dm-1  /dev/dm-2  /dev/dm-3
[root@cdrlgate ~]#                                  gave it a "shadow" name
[root@cdrlgate ~]# cryptsetup create sdfs_encrypted_name /dev/sdf   for encrypted I/O
Enter passphrase:
[root@cdrlgate ~]# ls /dev/sd*
/dev/sda  /dev/sda1  /dev/sda2  /dev/sdb  /dev/sdc  /dev/sdd  /dev/sde  /dev/sdf
[root@cdrlgate ~]# ls -l /dev/mapper/sd*
lrwxrwxrwx 1 root root 7 Aug 15 11:15 /dev/mapper/sdfs_encrypted_name -> ../dm-4
[root@cdrlgate ~]# ls /dev/dm*
/dev/dm-0  /dev/dm-1  /dev/dm-2  /dev/dm-3  /dev/dm-4
                                             New!
[root@cdrlgate ~]#
[root@cdrlgate ~]# cryptsetup status sdfs_encrypted_name
/dev/mapper/sdfs_encrypted_name is active.
  type:  PLAIN
  cipher:  aes-cbc-essiv:sha256
  keysize: 256 bits
  device:  /dev/sdf
  offset:  0 sectors
  size:    15826944 sectors
  mode:    read/write
[root@cdrlgate ~]# cryptsetup remove sdfs_encrypted_name
[root@cdrlgate ~]# cryptsetup status sdfs_encrypted_name
/dev/mapper/sdfs_encrypted_name is inactive.
[root@cdrlgate ~]# ls /dev/dm*
/dev/dm-0  /dev/dm-1  /dev/dm-2  /dev/dm-3
                                  Gone!
[root@cdrlgate ~]#
```

Using hardware interfaces



App

Using software interfaces



- **looks like an interface to an app**
- **looks like an app to an interface**
- **gets to massage (e.g. encrypt) traffic passing through**

Our implementation

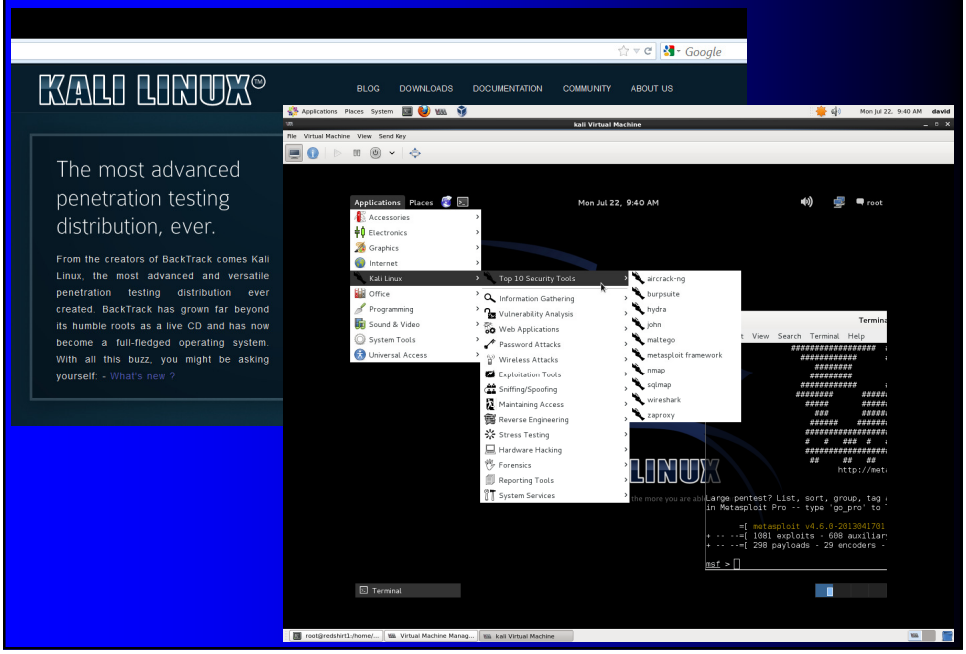
- dm-crypt for data-at-rest
  - near whole-disk encryption
  - on server machines (by extension their virtual ones)
- OpenVPN, ssh for data-in-motion
  - encrypted tunnel to DETER

Provisioned tools

- Kali linux – metasploit
- backtrack

- wireshark – protocol analyzer/sniffer
- nmap – port scanner
- nessus, nexpose – vulnerability scanners/analyzers

Many, for the price of one



Leveraging the node population
with virtual machine products

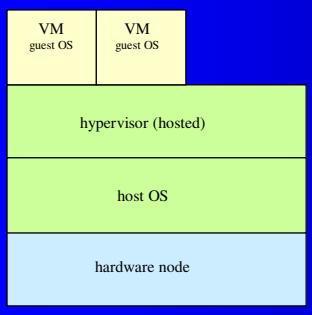
- containers
  - OpenVZ standalone
- hosted hypervisor
  - KVM, qemu via linux Virtualization Library
- bare-metal hypervisor
  - VMware vSphere, dedicated machine

OpenVZ containers

- lightweight
- created 1000
- scripted quick start/stop to obtain large numbers (e.g. for denial-of-service gang-ups)

Hosted vs bare-metal hypervisors

Hosted

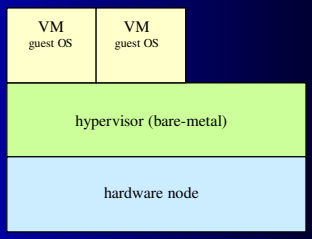


hypervisor is in immediate contact below with an OS. OS is in contact with hardware.

hypervisor is in immediate contact above with virtual machines

VMware example: VMware Player

Bare-metal

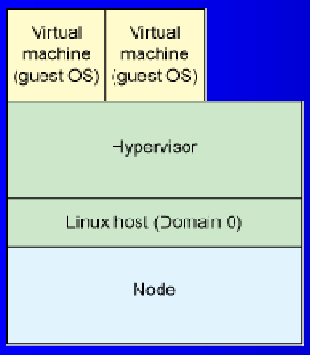


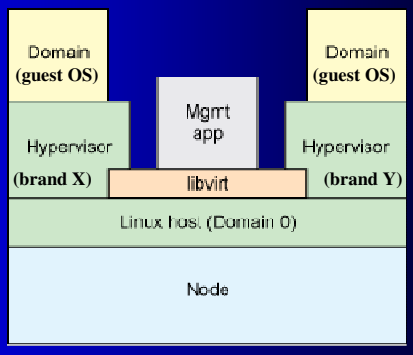
hypervisor is in immediate contact below with hardware. It is an OS.

hypervisor is in immediate contact above with virtual machines

VMware example: vSphere ESXi

role of libvirt virtualization library

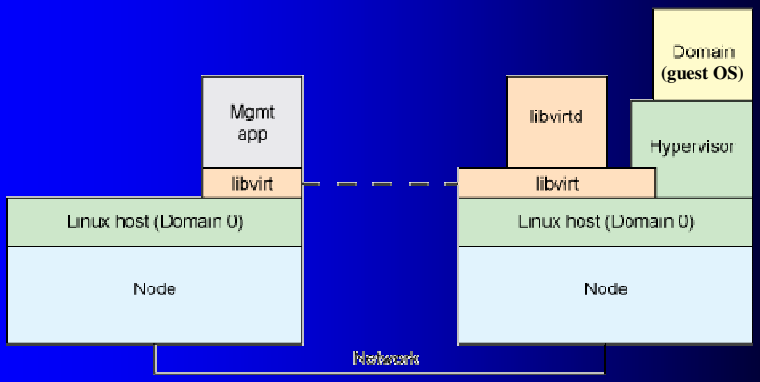




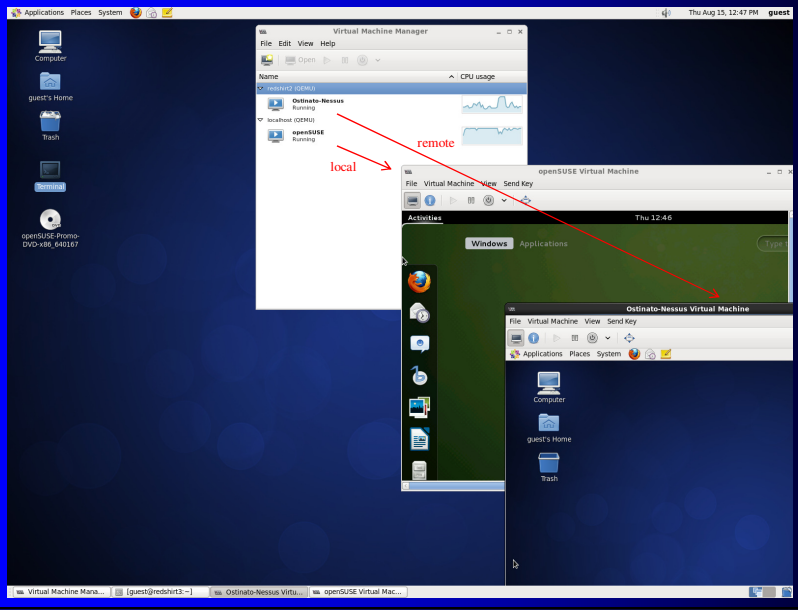
"…support[s] extensibility over a wide variety of hypervisors, …which allows a common API to service a large number of underlying hypervisors in a common fashion."

"Anatomy of the libvirt virtualization library," IBM Developer Works
http://www.ibm.com/developerworks/library/l-libvirt/

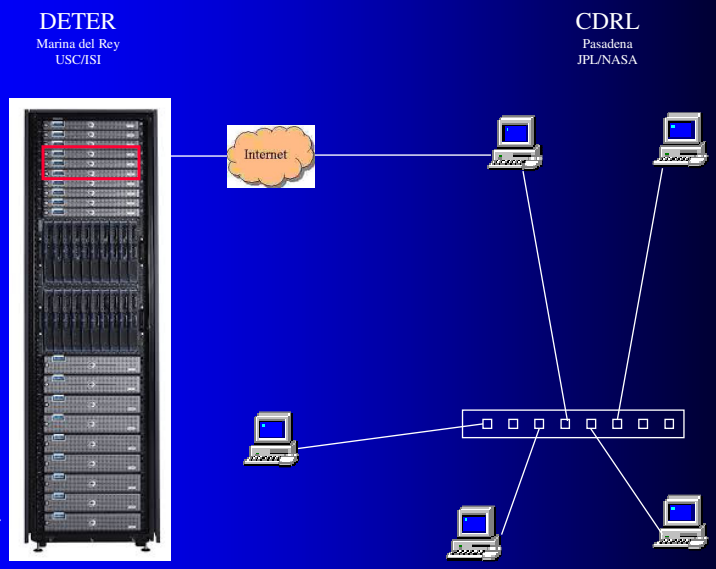
Remote control



Mgmt app interface to libvirt



Encompass DETER “rent-a-net” networks



Is it like cloud?

No, cloud is like it.

Yes, it’s remote.

No, PMs not VMs.

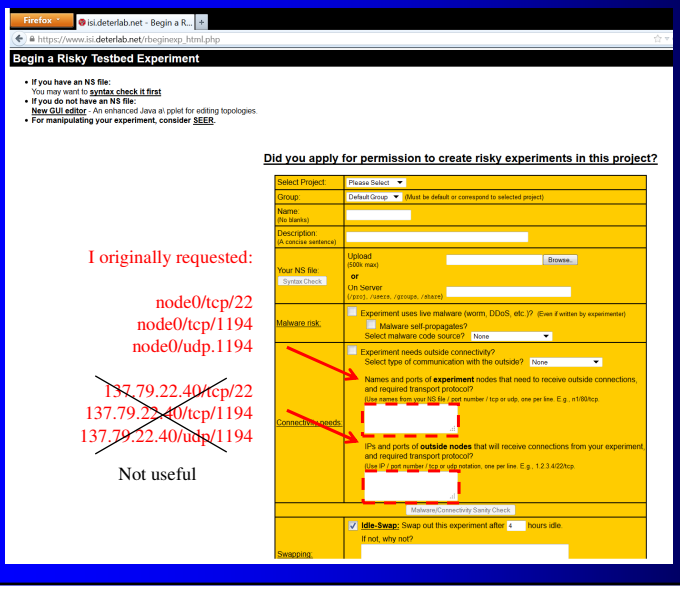
Yes, use as needed.

No, …

Two DETER integration formalizations

- “Risky experiment”
  - benefit: outisde net can ***talk*** to DETER net
  - requirement: get permission
  - timeframe: shorter-term (we did it)
- Federation
  - benefit: outside net can ***join*** DETER net
  - requirement: write an access controller (driver)
  - timeframe: longer-term (maybe we’ll do it)

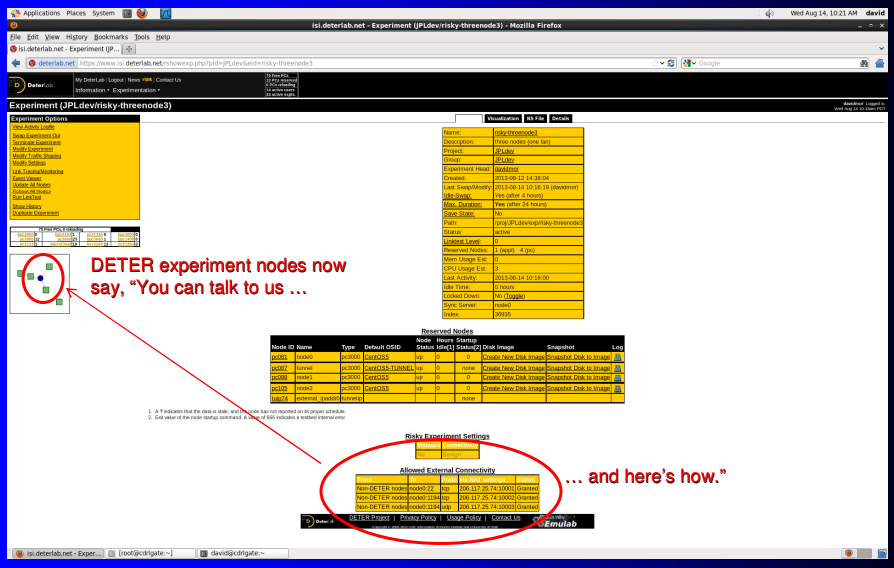
“risky” experiment admits outlanders



But… JPL firewall issue

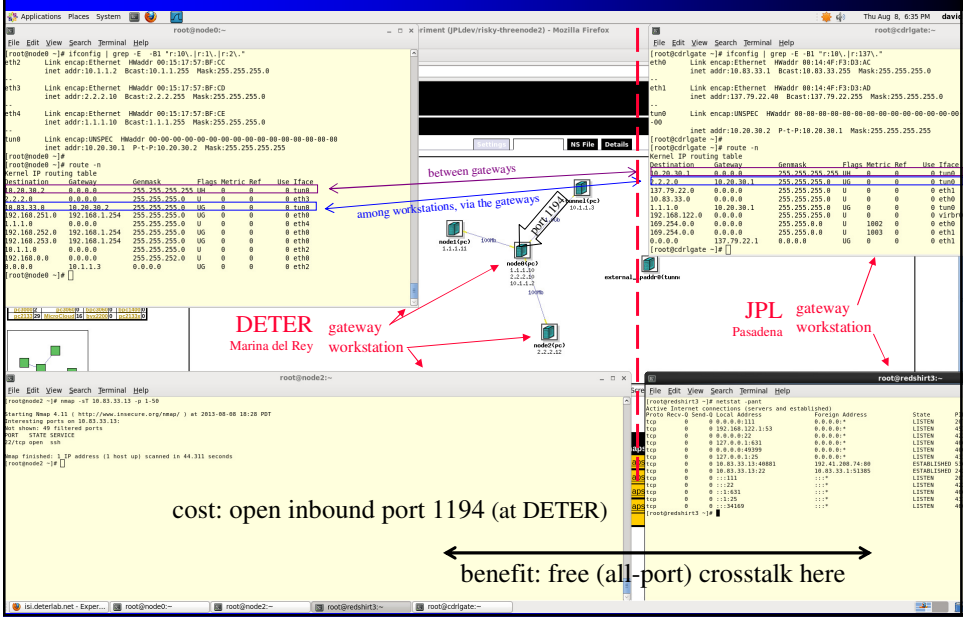
- JPL perimeter drops unsolicited inbound
- in vain, currently, asking DETER to let leave what JPL won’t let arrive
- directionality: JPL → DETER only for unsolicited traffic
- therefore TCP only, for desired OpenVPN app
  - OpenVPN defaults to 2-way unsolicited UDP handshake
  - drawback: implies tcp over tcp (potential bad adaptive timeout interactions)

The resulting benefit



What we did over summer vacation

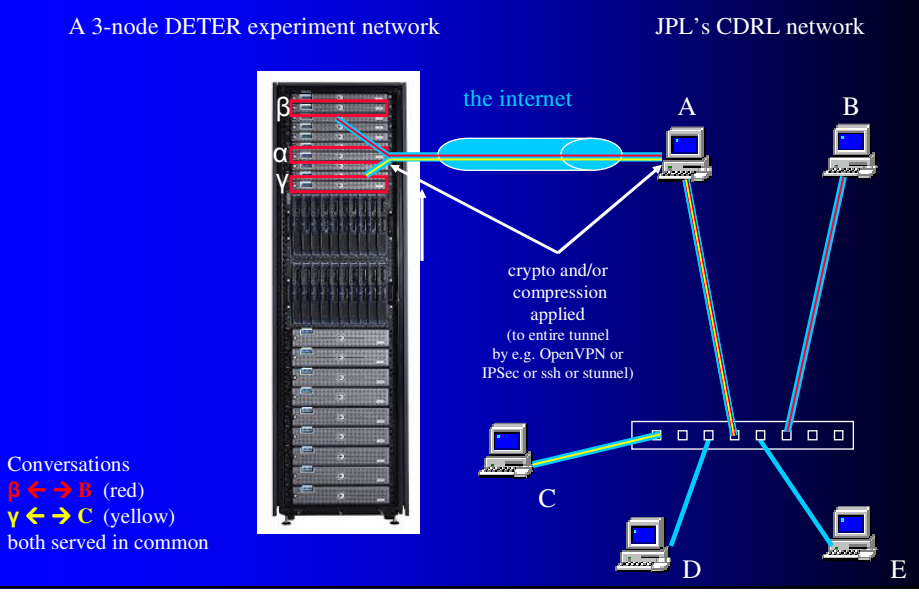
with port 1194 access at DETER (namely, OpenVPN)



Virtues of OpenVPN

- multiplexes all ports/protocols over a single port and protocol (ask DETER for just that one)
- applies common encryption service to all machines in the 2 networks

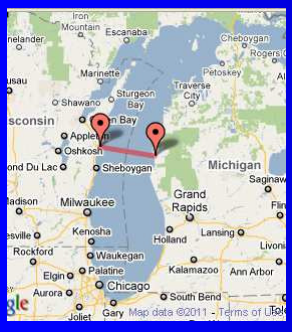
Applying common services



A VPN…

- Conveys an IP packet between machines
  - … not as a packet
  - … but as cargo in another packet
- Destination shucks carrier packet, releases cargo as packet into local networking machinery
- "Tunnel" since one packet "passes through" another
- Implemented in linux by module ipip.o

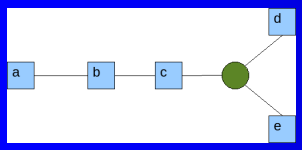
S.S. Badger



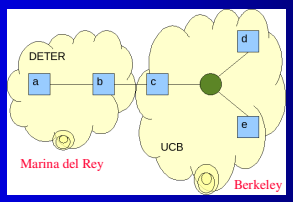
- Conveys a car between states
  - … not as a car/motor-vehicle
  - … but as cargo in a boat
- Destination throws away boat, releases car as a motor vehicle onto local roadways
- "Tunnel" since one vehicle "passes through" another
- Implemented by Lake Michigan Carferry Service

2nd DETER integration approach: experiment across federated testbeds

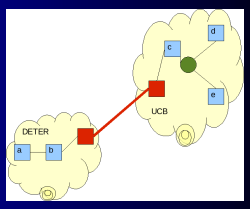
user's view (consolidated)



geographical view (distributed)



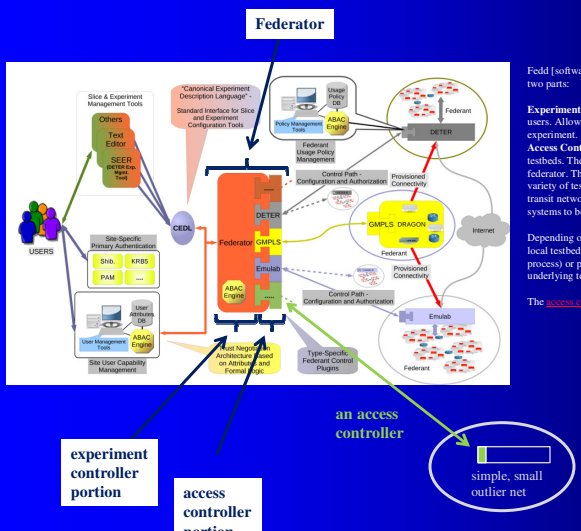
implementation



If federated, JPL would be to DETER as UCB is to DETER

How to federate an ordinary net?

- write an access controller -



Fedd [software] plays the role of the Federator in the diagram…. It is divided into two parts:

Experiment controller Presents the abstraction of a federated experiment to the users. Allows allocation, configuration, and release of resources to the federated experiment.
Access Controller Negotaties access and allocation of resources from federant testbeds. The access controller is depicted by the regular plug-in shapes in the federator. These are standard plug-in interfaces that allow standard access to a variety of testbed types. Currently emulab style testbeds and DRAGON style transit networks are supported. The regular system interface to fedd allows new systems to be added easily.

Depending on th nature of the access controller it may be implemented on the local testbed, on the same host as the experiment controller (even in the same process) or partially in both places. This depends on the access mechanisms of the underlying testbed and the policies of the administrators. …

The access controller/experiment controller interface is documented.

Thank you …

DJ, Chris, Bryan, Kymie et al
(393G – Cyber Defense and Information Architecture group)

Rich Alvidrez (JPL) and Jinan Darwiche (SMC)

Jet Propulsion Lab

… for this special summer opportunity