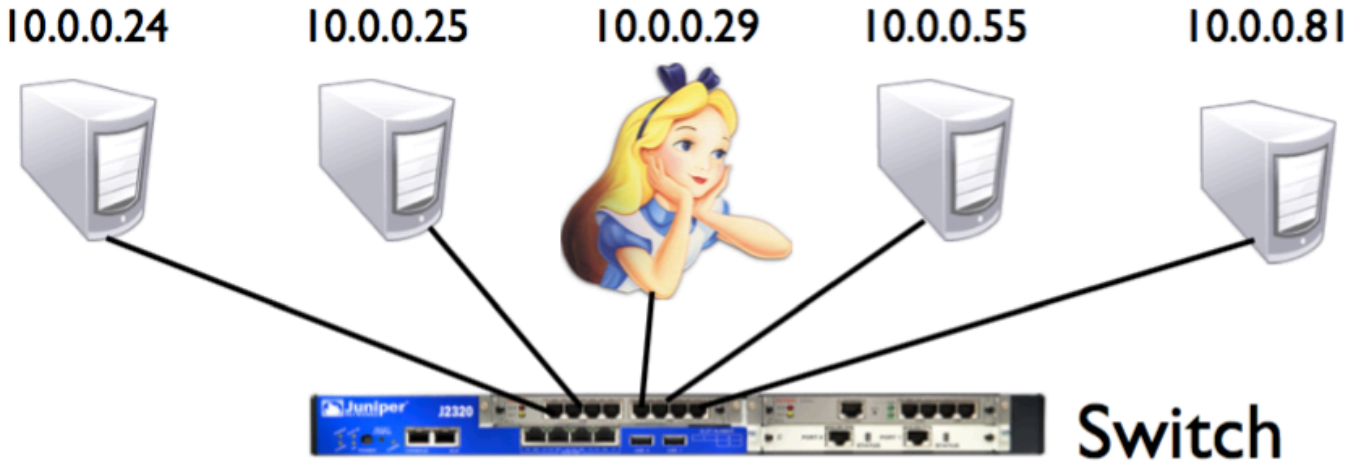
# February 15, 2021
# Intrusion Protection


- Today
  - Software Defined Networking
  - SNORT Rules
  - Honeypots
- Assignments
  - Lab 1: Due Monday, Feb 22
  - Project
    - Topic Due: Monday, Feb 22

# Routing Problem:
# How do Alice's messages get to Bob?


MAC 6 BYTES
↳ INTERFACE
IP IPV4 4 BYTES
IPV6 128 BITS

# Routing within the local network

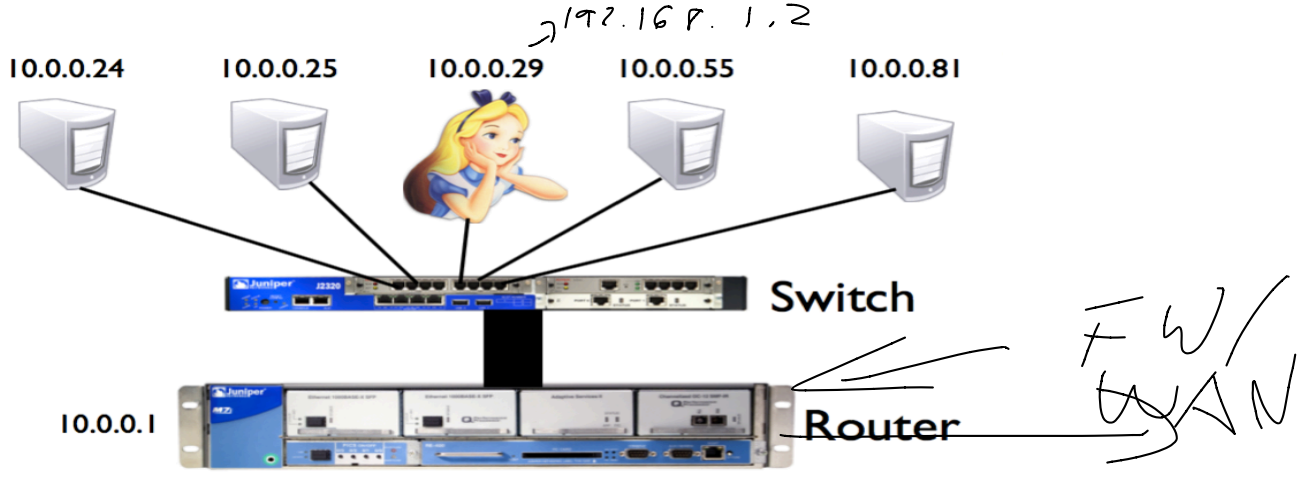


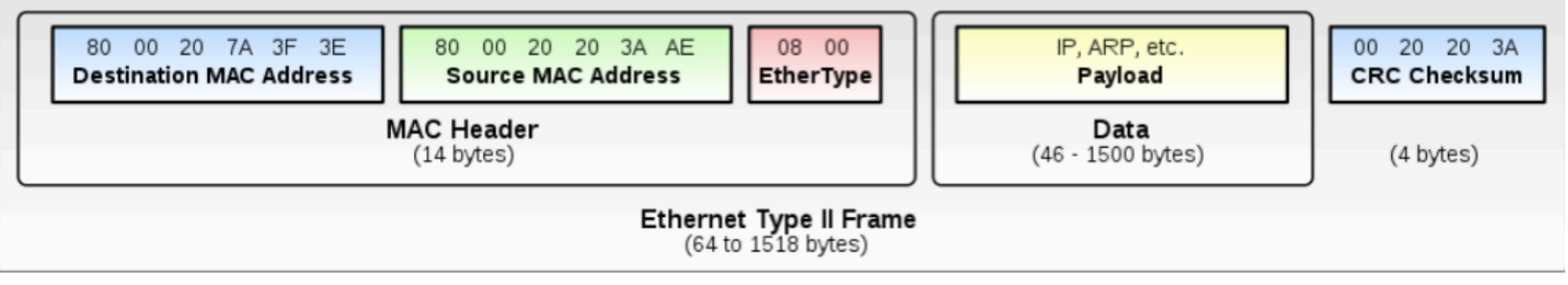

Each host knows the network prefix of the local network

- All nodes within the local network are reachable within 1 hop
- **CIDR Notation**: BaseAddress/Prefix_Size
  - e.g., 10.0.0.0/24:

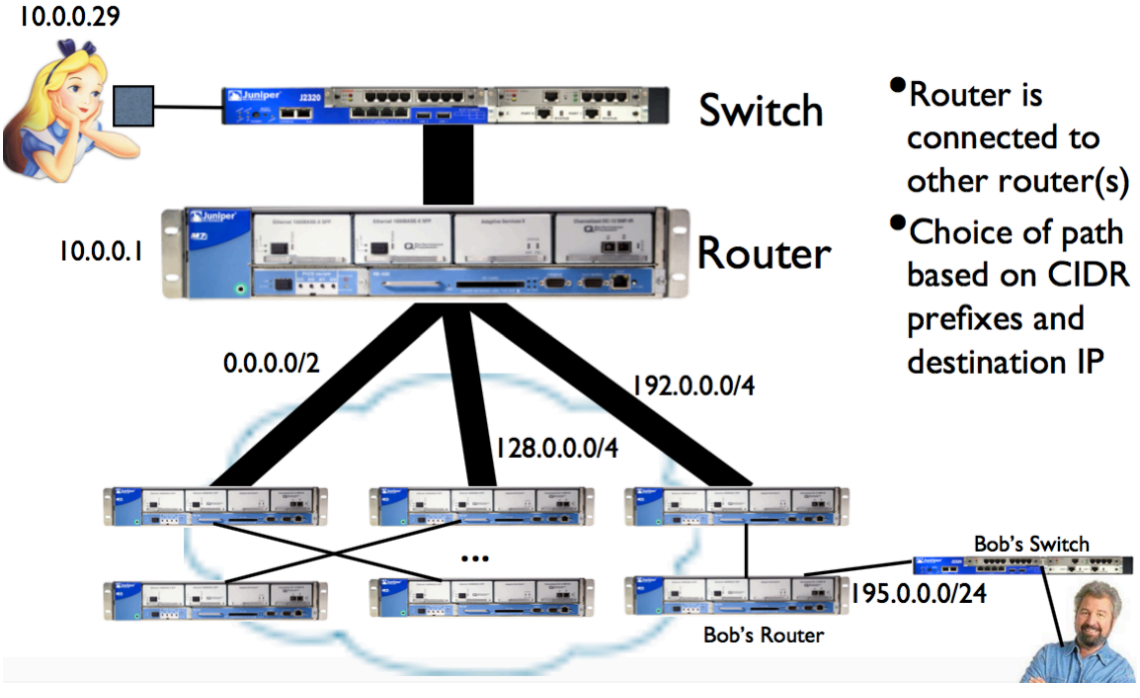
ARP
IP → MAC

# Routing outside the local subnet



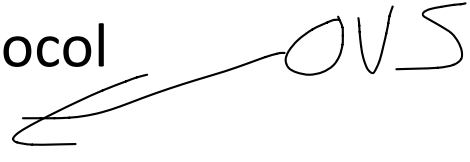
# Routing outside the local subnet

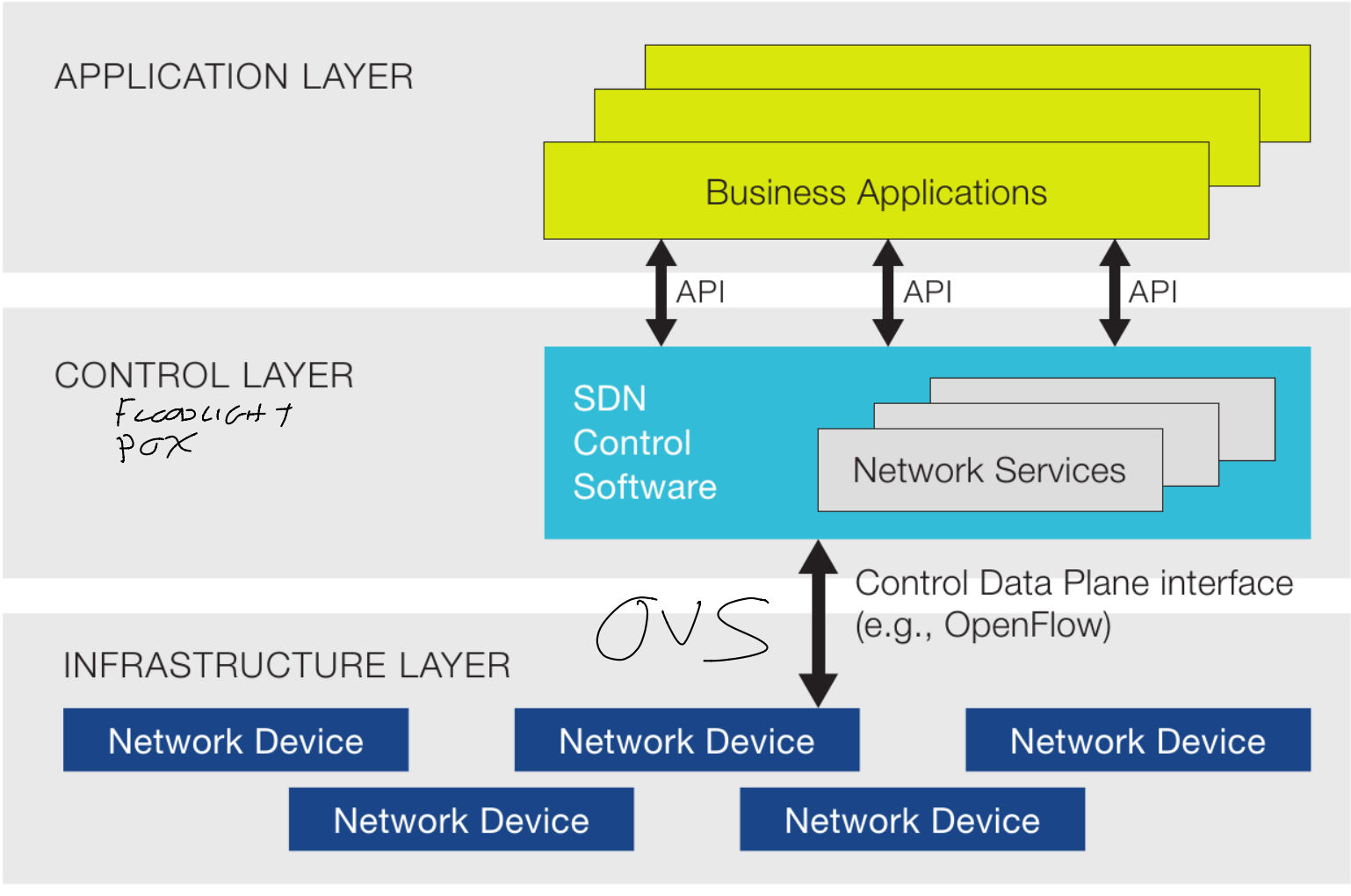


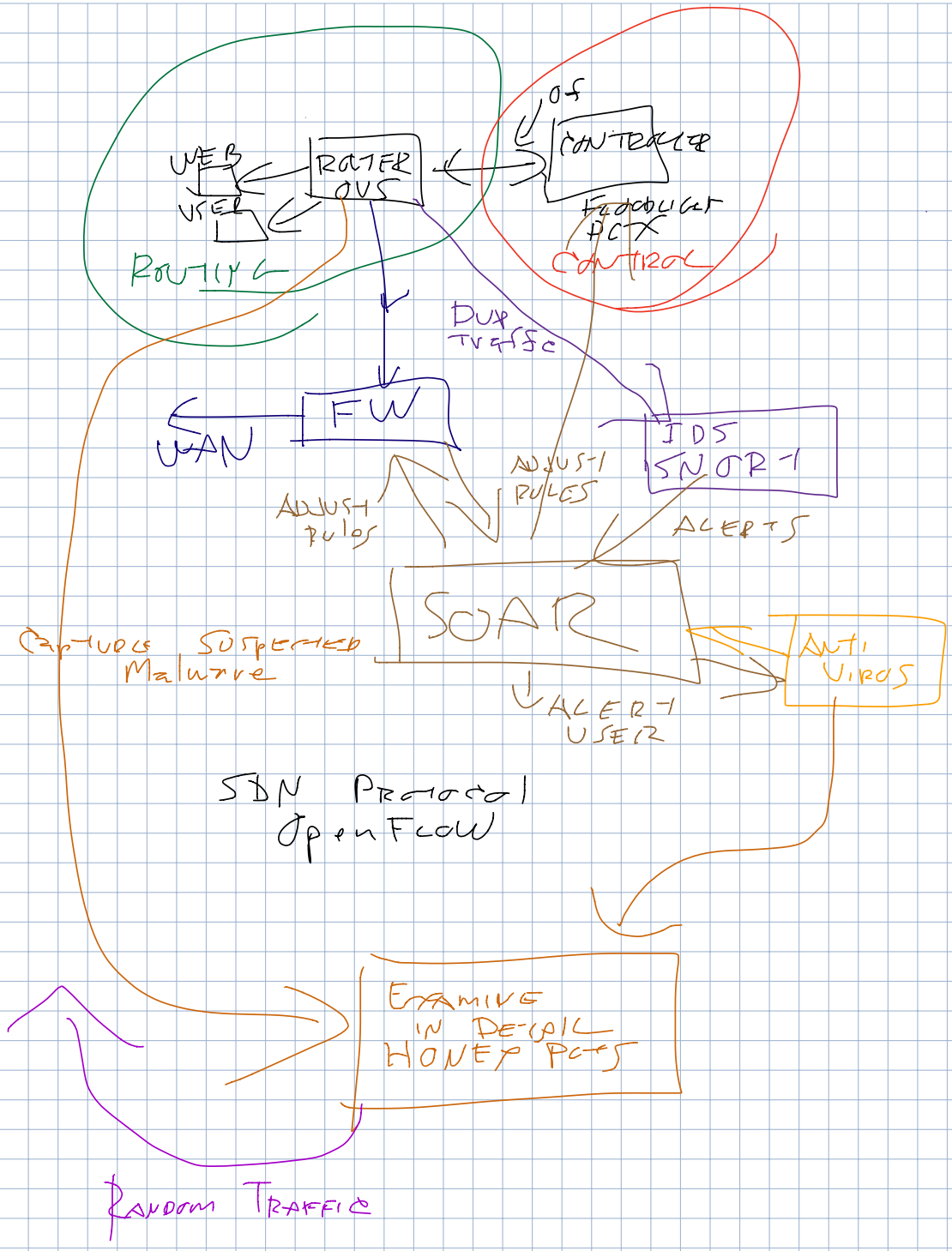
# Software Defined Networking (SDN)

- Two packet activities
  - Routing – Path for packets
    - Control plane
  - Forwarding – Actual movement of the packet
    - Data plane

- Separation of Concerns
- OpenFlow protocol
- Open vSwitch
  - Controllers (floodlight, pox, etc.)




APPLICATION LAYER
Business Applications
API
API
API
CONTROL LAYER
Floodlight
POX
SDN
Control
Software
Network Services
OVS
Control Data Plane interface
(e.g., OpenFlow)
INFRASTRUCTURE LAYER
Network Device
Network Device
Network Device
Network Device
Network Device


OF
CONTROLLER
WEB
VIEW
ROUTER
OVS
FLOODLIGHT
REST
CONTROL
ROUTING
DUP
TRAFFIC
FW
WAN
IDS
SNORT
ADJUST
RULES
ADJUST
RULES
ALERTS
SOAR
CAPTURE SUSPECTED
MALWARE
ANTI
VIRUS
ALERT
USER
SDN PROTOCOL
OpenFlow
EXAMINE
IN DETAIL
HONEYPOTS
RANDOM TRAFFIC

# Snort


- Open source IDS
- Signature detection
- Lots of available rulesets

ACTION PROTOCOL IP/PORT ie. 10.0.0.0/16 ← HEADER

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306
(msg:"MYSQL root login attempt";
 flow:to_server,established;
 content:"|0A 00 00 01 85 04 00 00 80|root|00|";
 classtype:protocol-command-decode;
 sid:1775;
 rev:2;)
```

OPTIONS ← Remainder of match

IDS Approaches
- Rules
- Machine Learning

Rules
→ Match
→ Actions
↳ Alert
↳ Drop, Pass, Reject
↳ Packet Rewritting

More than raw packets
- Reconstitute protocol layer
- HTTP Reply multiple packet
- Combine into 1

# How can we study attacks?

- Honeypots!
  - collection of decoy services (fake mail, web, ftp, etc.)
  - decoys often mimic behavior of unpatched and vulnerable services

# Honeypots

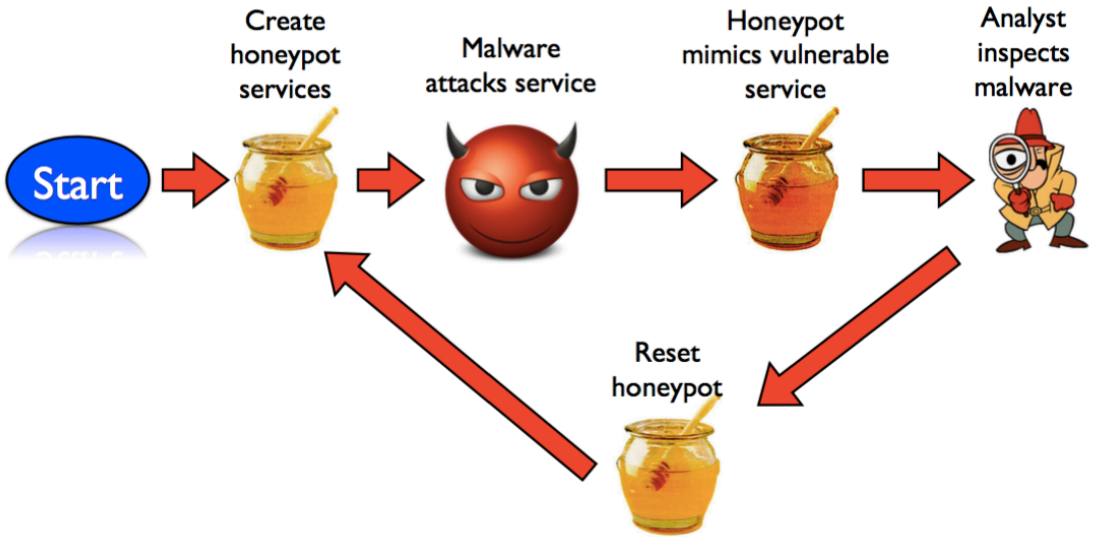
- Three main uses:
  - **forensic analysis**: better understand how malware works; collect evidence for future legal proceedings
  - **risk mitigation**:
    - provide "low-hanging fruit" to distract attacker while safeguarding the actually important services
    - tarpits: provide very slow service to slow down the attacker
  - **malware detection**: examine behavior of incoming request in order to classify it as benign or malicious

# Honeypots

- Low Interaction: emulated services
  - inexpensive
  - may be easier to detect

- High Interaction: noemulation;honeypot maintained inside of real OS
  - Expensive
  - good realism

# Example Honeypot Workflow



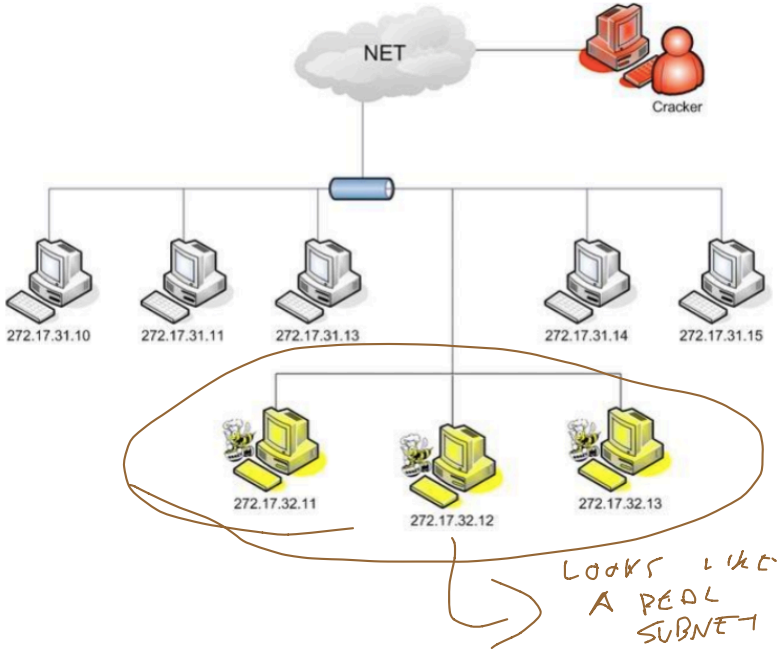
# Challenges

- Honeypot must resemble actual machine
  - simulate actual services (Apache, MySQL, etc.)
  - but not too much... bad form to actually help propagate the worm (legal risks!)
- Some worms do a reasonably good job of detecting honeypots

# Honeynets

- **Honeynet:** also called **honeyfarms**
  - Collection of honeypots that simulate a network; or
  - Single honeypot that emulates services on multiple emulated “machines” (that is, on a network)

# Example Deployment



# honeyd

- Open-source virtual honeynet
  - creates virtual hosts on network
  - services actually run on a single host
  - scriptable services

# Internet Background Radiation

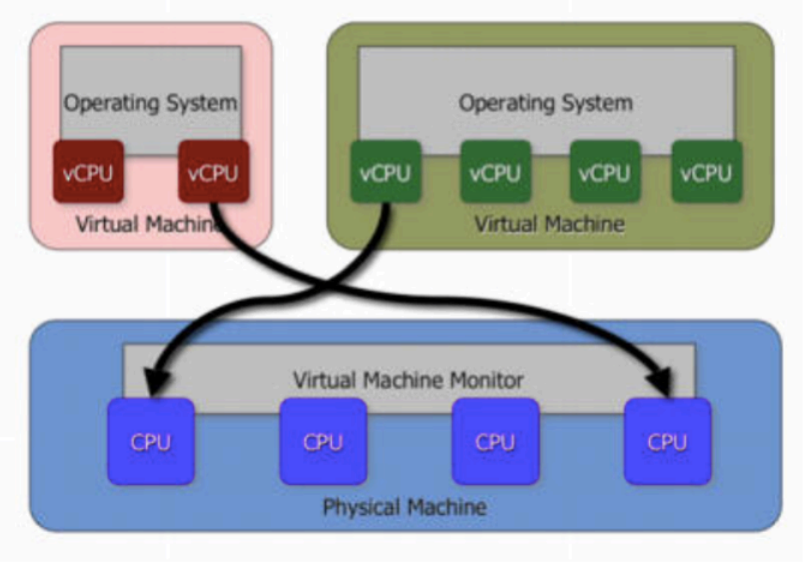
- **Internet Background Radiation** or **Backscatter:** Traffic that is sent to addresses on which no device is set up (these unused portions of the Internet are called **darknets**)
  - Backscatter primarily originates from spam, worms, and port scans
  - Estimated at 5.5Gbps
  - Estimated that 70% of background radiation due to Conficker Worm

ATTACKERS LOOK FOR PATTERNS
- RANDOMNESS (ENTROPY) IS YOUR FRIEND
- ENCRYPTED DATA
  ↳ ALMOST INDISTINGUISHABLE FROM RANDOM BYTES
- RANDOM PACKETS
  ↳ ALMOST IMPOSSIBLE TO FIND THE "REAL" DATA

# Virtual Machines

- **Virtual machine:** isolated virtual hardware running within a single operating system
  - i.e., a software implementation of hardware
  - usually provides emulated hardware which runs OS and other applications
  - i.e., a computer inside of a computer
- What's the point?
  - extreme software isolation -- programs can't easily interfere with one another if they run on separate machines
  - much better hardware utilization than with separate machines
  - power savings
  - easy migration -- no downtime for hardware repairs/ improvements

# Virtual Machines



# Honeypots and VMs

- Most virtual machines provide checkpointing features
  - **Checkpoint** (also called **snapshot**) consists of all VM state (disk, memory, etc.)
  - In normal VM usage, user periodically creates snapshots before making major changes
  - Rolling back ("restoring") to snapshot is fairly inexpensive
- **Checkpointing features are very useful for honeypots**
  - Let malware do its damage
  - Pause VM and safely inspect damage from virtual machine monitor
  - To reset state, simply restore back to the checkpoint

# Detecting VMs

- Lots of research into detecting when you're in a virtual machine
  - examine hardware drivers
  - time certain operations
  - look at ISA support
- Malware does this too!
  - if not in VM, wreak havoc
  - if in VM, self-destruct