LINUX IN THE ENTERPRISE

OBJECTIVES

- Understand what Linux is
- Understand devices that Linux is built into
- Understand why we need to care about Linux in the enterprise

WHAT IS LINUX?

- Linux is an operating system
- It's used for a lot of technology, such as cars, TV's, thermostats, and particularly servers
- Linux runs most of the internet in some, size, shape or form
  - Estimated over 95% of the internet runs on Linux
- 99% of supercomputers run Linux
- Android, which powers over 2/3 of phones in the world, is Linux

A BRIEF HISTORY ON LINUX

- GNU Project started in 1983 by Richard Stallman
  - Goal was to create a “complete Unix-compatible software system” composed of free software
- 1985 Stallman started Free Software Foundation and wrote the GNU GPL
  - GNU stands for “GNU’s Not Unix” – a recursive acronym
  - GPL stands for General Public License which allows users to run, study, share and modify the software
- UNIX like operating system that started development by Linus Torvalds in 1991
  - Started to take off in the mid 90’s when it was adopted by larger corporations

TODAY

- Linux keeps evolving
- Linus Torvalds is still the primary contributor to the kernel and still completely involved
- There are 100's of distributions and variations of Linux

WHY DO WE CARE?

- Someone at least has something they own that runs Linux
- Enterprises run at least something on Linux
  - Industrial control systems
  - Web servers
  - Files servers
  - DNS
  - Big Data systems

WHAT IS LINUX GOOD AT?

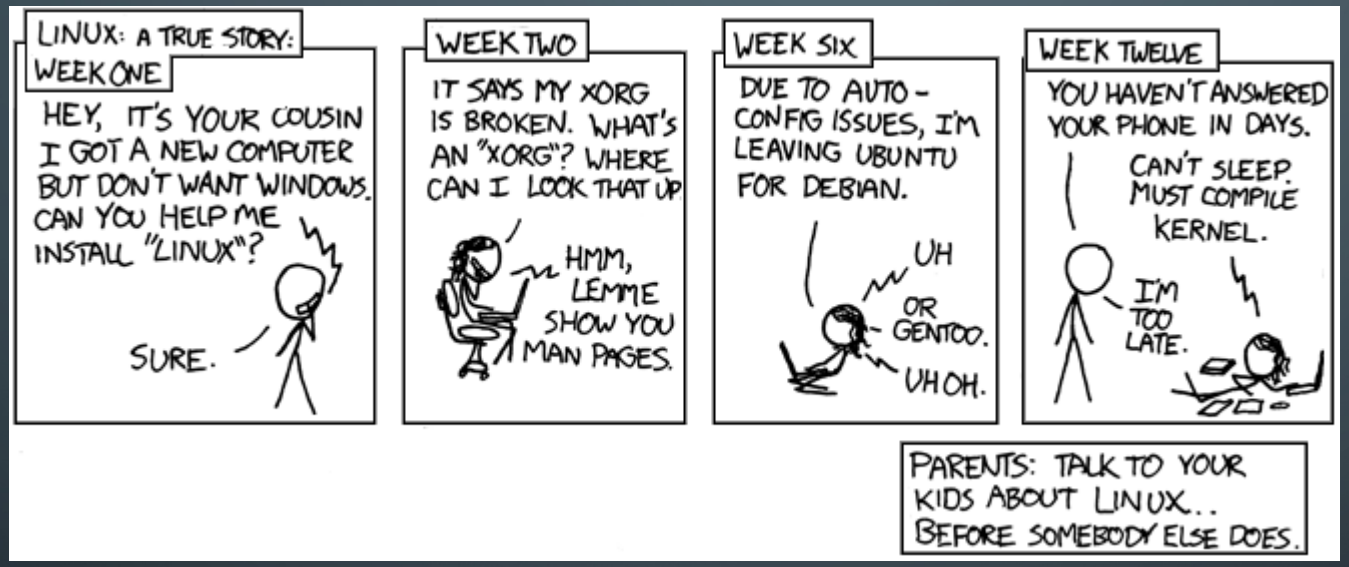
- You name it – there is something that it is good at
- Just takes time to figure out and program
- Some of the pitfalls
  - You have to understand it
  - You constantly need to know how it works and keep up with the technology
  - It's a different way of thinking

CONCLUSION

- A system administrator will encounter Linux at some point in their career
- Well versed Linux administrators will always have work
- Since Linux runs quite a bit of the internet and technology, it's best to learn it even if you are not going to be a Linux admin

ABOUT LINUX AND DISTRIBUTIONS

THE BIRDS AND THE BEES



- Source: xkcd.com

LINUX DISTRIBUTIONS

- In order to be called “Linux”, a distribution must have the Linux kernel
- Each usually has some kind of package management system in order to install software
- There are 1000’s of distributions out there
- You can create your own, if you would like - http://www.linuxfromscratch.org/
- Distributions may include, including the kernel:
  - Open source tools and libraries
  - Window Manager
  - Desktop environment
- While most Linux distributions are open source, some are paid for due to support and proprietary software

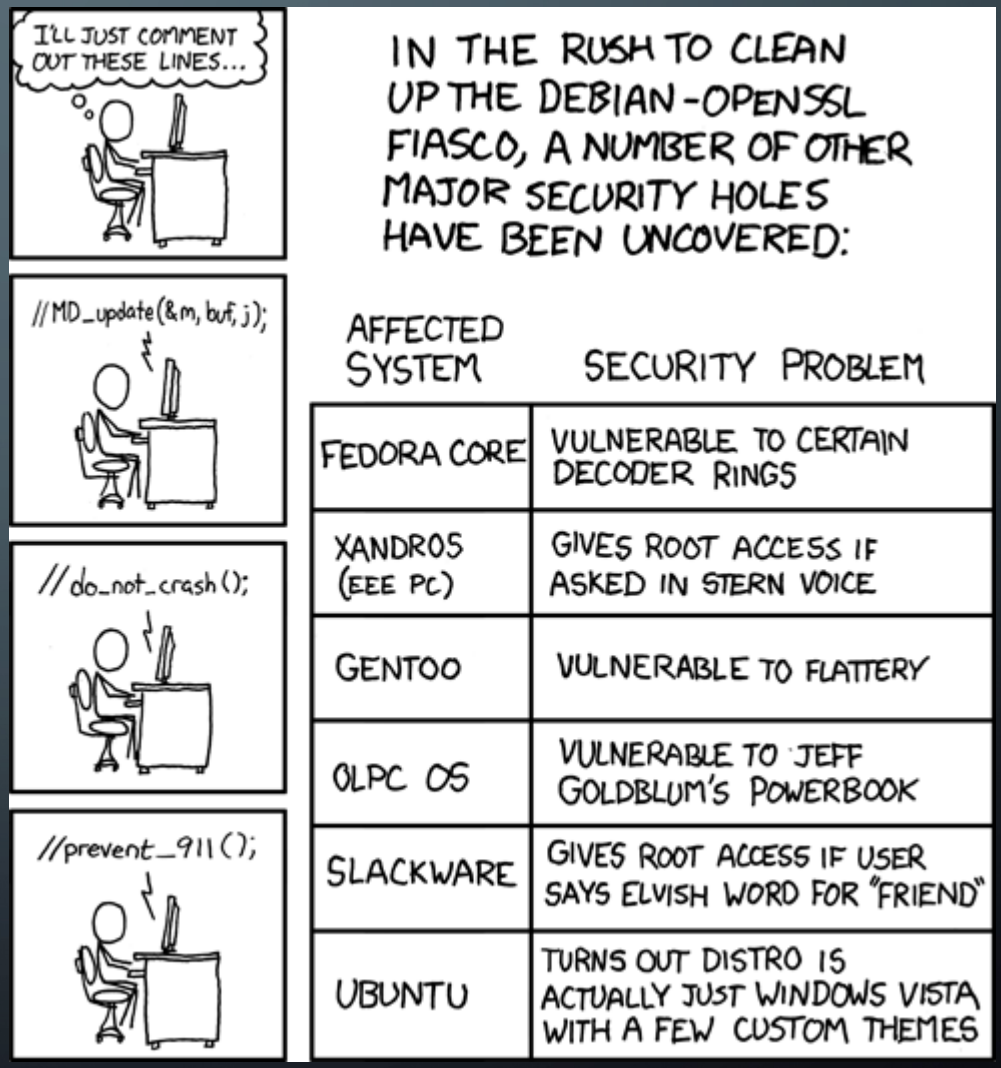
LINUX DISTRIBUTIONS – HOW TO DECIDE

- What Linux distribution is best?
  - Really, which Linux distro is best for me?
- Does the distribution have support?
  - Is there paid support?
  - Is there a large user support community?
- Does the distribution stay up to date on packages?
  - Do they stay up to date on security patches?
- How difficult is the distribution to use?
  - Will my coworkers hate me for picking it?

LINUX DISTRIBUTIONS – MAJOR

- Distrowatch.com has a lot of information about distribution
- Major distribution include:
  - Redhat
    - Fedora,CentOS,Scientific
  - Debian
    - Ubuntu, Knoppix, Mint
  - openSUSE
  - Arch
  - Gentoo
  - Slackware

DISTRIBUTIONS COMIC



LINUX DISTRIBUTIONS – CUSTOM EXAMPLES

- Phones - Android
- Routers - ddWRT
- Security - Kali
- IoT – Nest, Raspberry Pi
- Privacy - Tails

CONCLUSION

- Linux distributions have all kinds of different capabilities
- Some are large, some are small
- Some provide great support
- It's really up to the system administrator to decide what distro is right for which purpose

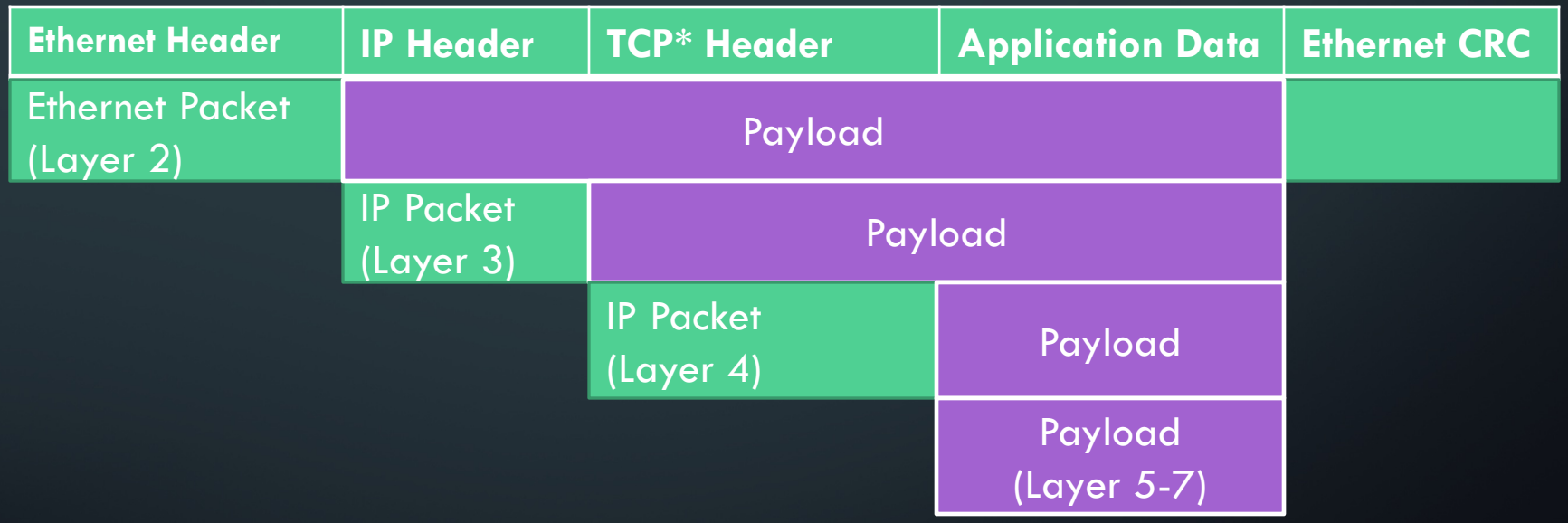
LINUX NETWORKING

PREVIOUSLY COVERED

- What is an IP?
- What is a subnet?
- CIDR notation

- Focus on IPv4

A TYPICAL PACKET

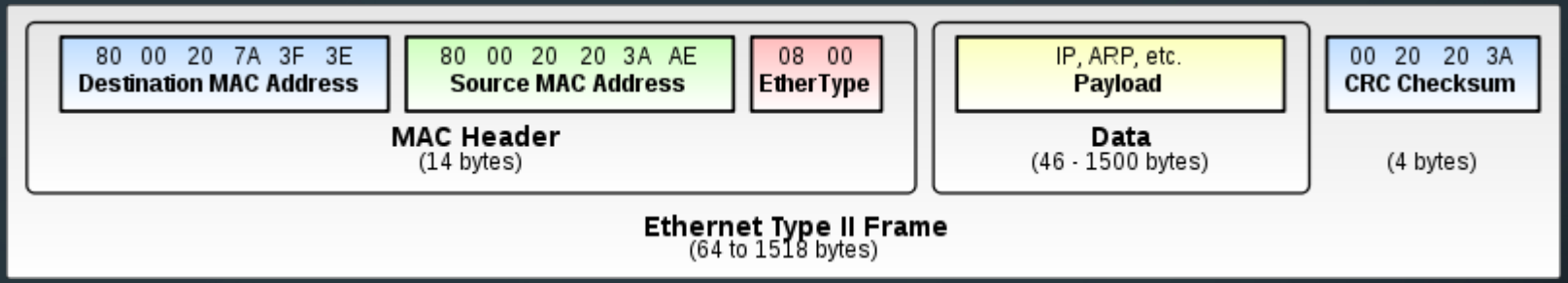


* Could be TCP, UDP, ICMP, or other protocols that ride on IP.

MTU

- Maximum Transmission Unit
  - Maximum size a layer can pass forward without having to break up the packet (fragmentation)
    - Ethernet is 1500bytes
    - 802.11 is 2272bytes
    - Jumbo Frames is 1500-9000bytes
- Ethernet Efficiency
  - Efficiency = $\frac{Payload\_Size}{Frame\_Size}$ $\frac{1500}{1538} = 97.53\%$ or 97.5Mbps on a 100Mbps connection

DATA LINK - LAYER 2 - ETHERNET



- Layer 2
- Typically 14 byte header
  - 6 byte destination address
  - 6 byte source address
  - 2 bytes for type
    - IP, IPv6, ARP, etc.
- Addresses must be unique
  - First 3 bytes represent manufacture
  - Burnt in during manufacturing – can be overridden (or spoofed)
- Special Addresses
  - Broadcast Address – FF:FF:FF:FF:FF:FF

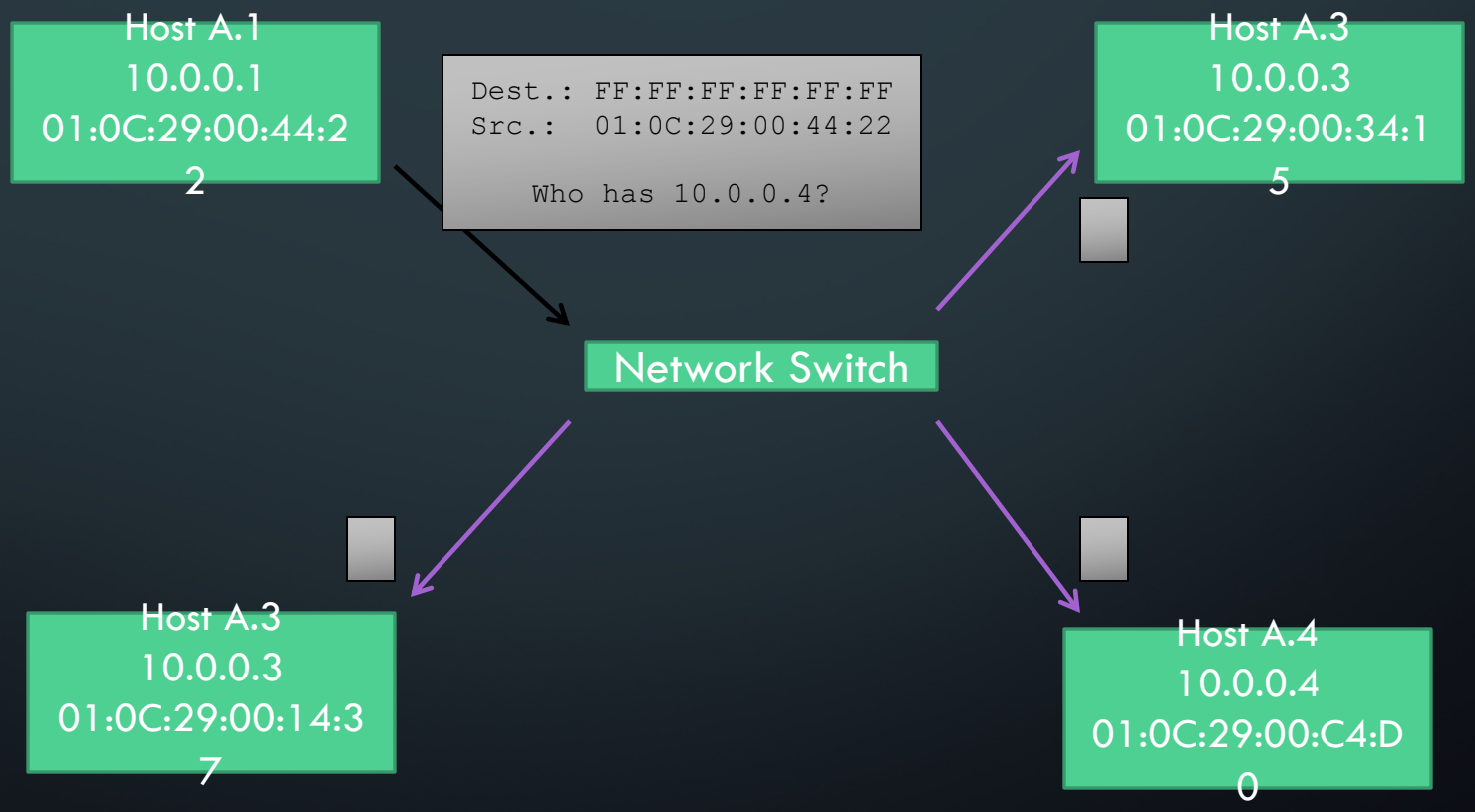
DATA LINK HARDWARE – HUBS AND SWITCHES

- Hubs
  - Send all packets to everybody
    - Not very secure
  - Shared Bandwidth
- Switches are smart hubs
  - Maintain MAC address list for each port
  - Dedicated port bandwidth

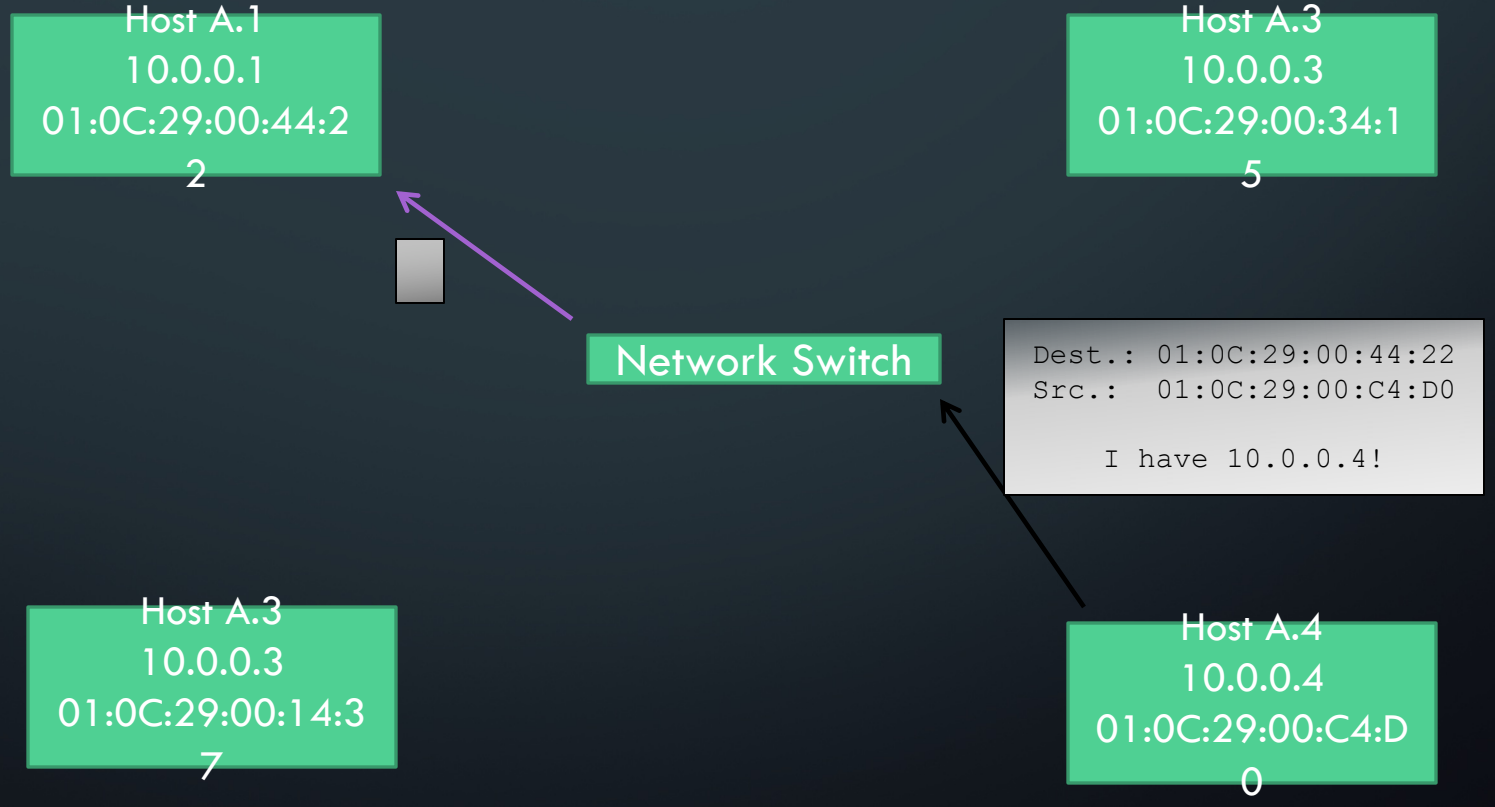
ARP

- Address Resolution Protocol
  - Resolve IP addresses to MAC addresses
    - Broadcasts who has IP to network
    - IP holder responds via senders MAC address
- Hubs and switches can only route MAC addresses
  - No knowledge of IP

ARP EXAMPLE - REQUEST

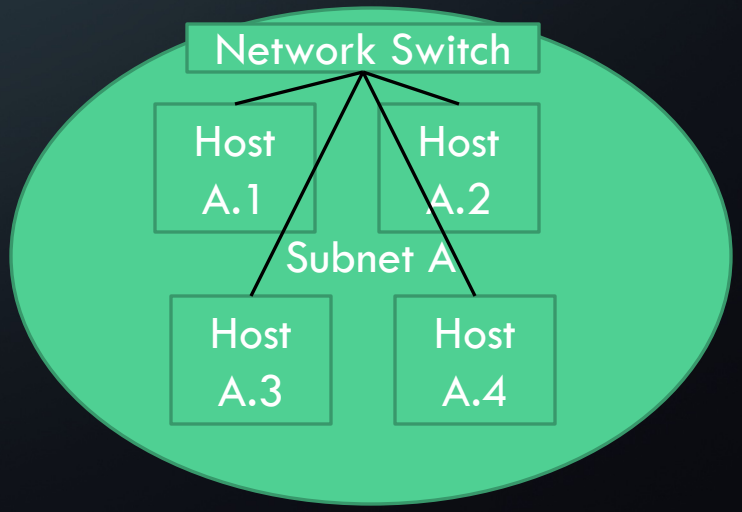


ARP EXAMPLE - RESPONSE



ARP/MAC

- Traditional Networks (ARP) Address Resolution Protocol
  - A.1 wants to talk to A.2
  - A.1 asks all hosts/everyone (**broadcasts**) what MAC is A.2?
  - A.2 **Broadcasts** back answer
  - A.1 sends packet to A.2 with A.2's MAC address (otherwise switch wouldn't know which network port to send it to)



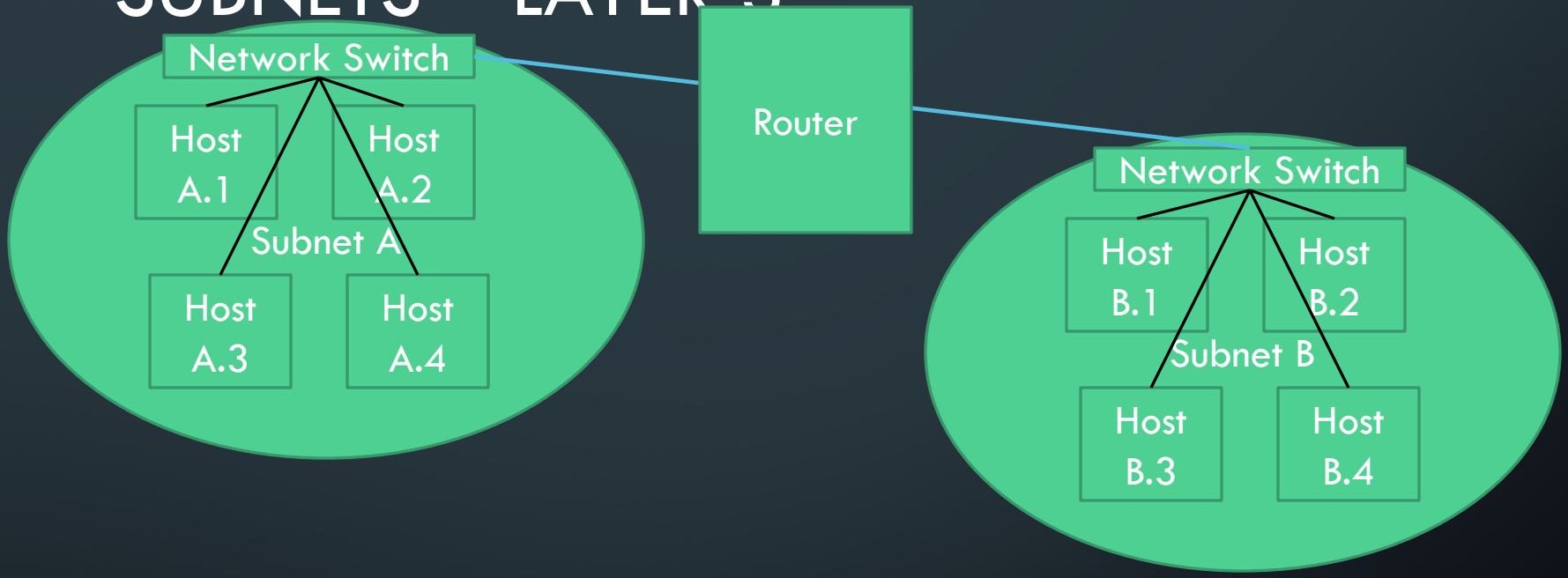
ARP RESPONSE - SECURITY

- Nothing to prevent other hosts from answering
  - First to respond wins
  - Can create Man in the Middle
- Switches can only remember finite number of MAC addresses (4k?)
  - If too many, switch can failsafe revert to hubs
  - MAC flood to create this situation
- Advanced switches can prevent this
  - $50 12 port switch vs. a $5k one.

ARP

- Works good, but what about large networks?
- Each host receives broadcasts
  - Must check if message is meant for host
- More hosts means more broadcast
  - Eventually run out of host system resources
- Need a way to segment networks

SUBNETS – LAYER 3



- A.1 wants to talk to B.1
  - A.1 sees B.1's IP is not on local subnet
  - A.1 sends data packet to default Router to route it
  - Router received packet and ARP process begins on B subnet

** Some additional ARPing may initially occur between A.1 and router (assuming cached)

SUBNETTING

- A subnet is a sub-network
  - A range of IP addresses
  - Defined by a subnet

- 10.0.20.0/24 – Subnet is 256 hosts
- 10.0.20.0/23 – Subnet is 512 hosts

WHAT IS A ROUTER?

- Router – Forwards data between compute networks beyond directly connected devices.
  - Connects multiple subnets together
  - (Slide 15)

- Devices are directly connected when data is forwarded using network switches.

ROUTER [GATEWAY IS A ROUTER]

- Routes traffic
  - Can be static routes (this is what we'll use)
  - Can dynamically build routes
    - Self healing, load balancing, scalable, etc.

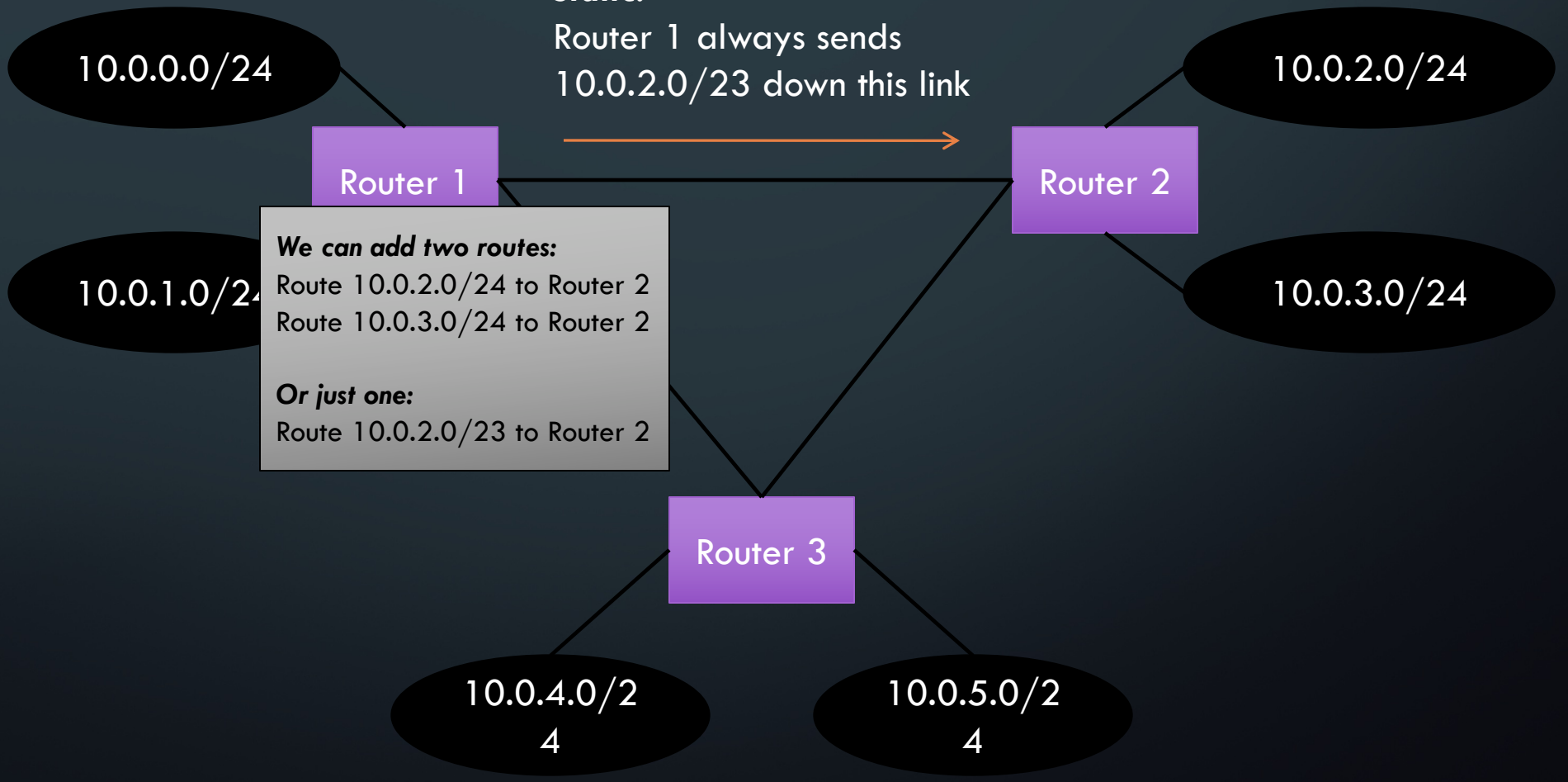
- If you want to send to an IP address not on your subnet (defined by subnet mask) you will need a router to send it for you
  - Can have a default router (only one)
  - Can have static routes to override
  - netstat -rn ← shows default routing table
    - Or: ip route show

ROUTING TABLE: NETSTAT -RN

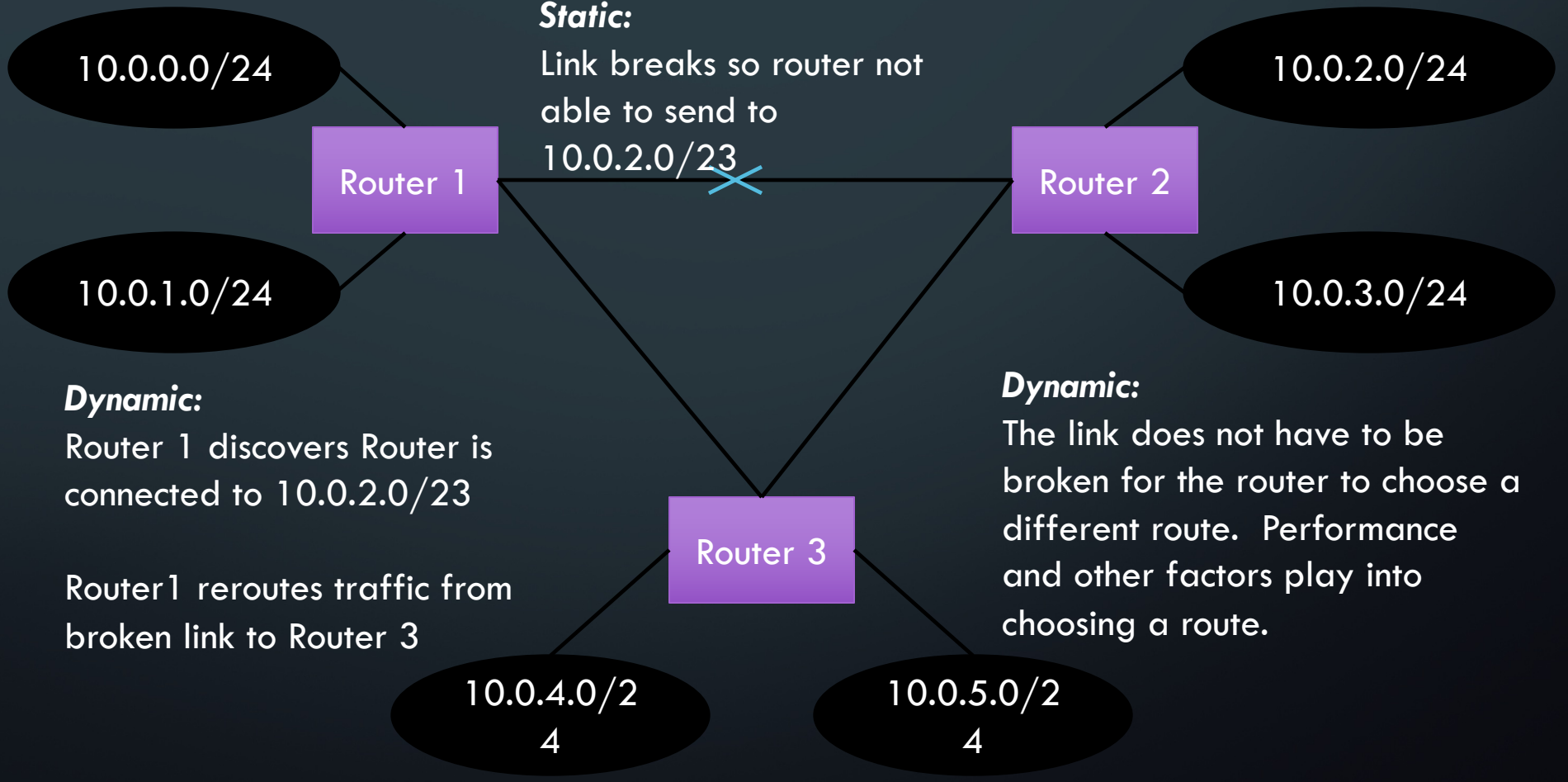
```
user@router:~$ netstat -rn

Kernel IP routing table

Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface

128.198.50.16   0.0.0.0         255.255.255.248 U         0 0          0 eth0

10.0.5.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1

10.0.7.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1

10.0.0.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1

10.0.3.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1

10.0.9.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1

10.0.11.0       0.0.0.0         255.255.255.0   U         0 0          0 eth1

10.0.12.0       10.0.0.106      255.255.254.0   UG        0 0          0 eth1

0.0.0.0         128.198.50.17   0.0.0.0         UG        0 0          0 eth0
```

- Routes are processed in order – default route last
- Mask 0.0.0.0 routes everything, but it is the last to be checked
- Almost all hosts have at least one route
  - Usually just a default route

STATIC VS. DYNAMIC



STATIC VS. DYNAMIC



Static:
Link breaks so router not able to send to 10.0.2.0/23

Dynamic:
Router 1 discovers Router is connected to 10.0.2.0/23

Router1 reroutes traffic from broken link to Router 3

Dynamic:
The link does not have to be broken for the router to choose a different route. Performance and other factors play into choosing a route.

RIP (Routing Information Protocol) is a way for the Routers to dynamically exchange route information.

STATIC VS. DYNAMIC

- So why choose Static?

STATIC VS. DYNAMIC

- So why choose Static?
  - It's quick/easier.
  - It's constant.

CREATING STATIC ROUTES

- Any traffic sent to us for a given subnet, we forward to a given IP address

- 1st: Need a subnet to route
- 2nd: Need a destination to route it to

- Examples:
  - ISP gave us 128.198.0.0/22
  - We have 6 routers and have to use one for the incoming connection.

MANUALLY ADDING ROUTES

- Default routes (gateways)
  - `route add default gw 10.0.0.1`
  - Statically routes subnet mask 0.0.0.0 or /0 to 10.0.0.1
- Static routes
  - `route add -net 10.0.12.0 netmask 255.255.254.0 gw 10.0.0.106 dev eth0`

PERSISTING STATIC ROUTES

- Edit vi /etc/network/interface and add:
  - `up route add –net 10.0.13.0/24 gw 10.0.12.137`

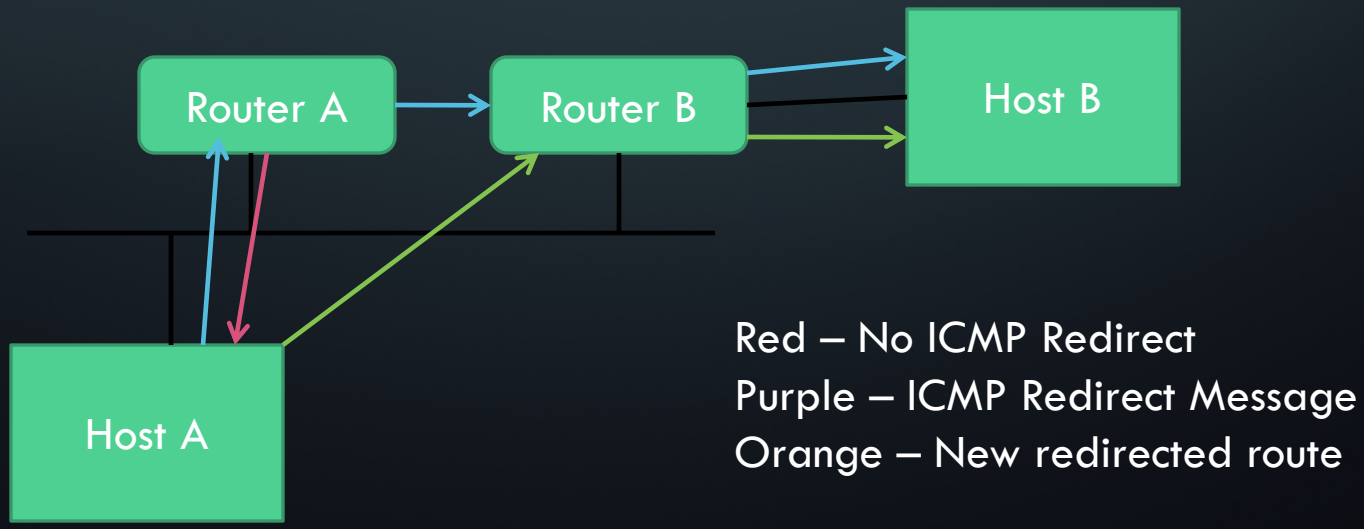
- This will apply a static route and route all 10.0.13.0 traffic the router sees to 10.0.12.137

ICMP

- Internet Control Messaging Protocol
  - Intended to complement IP
  - Not used to send data but rather host status and error messages
- Ping
  - ICMP command that queries if a host is online
  - If hosts receives a ping ‘echo request’, the host, if online, should respond with a ‘echo reply’
  - Useful for determining what hosts are online

ICMP REDIRECTS

- If a router received a packet and determines the host can route it more efficiently it sends an ICMP redirect
- Prevents excess router hops



HOW TO CHECK ROUTES

- tracert
  - Is able to determine the routers between it and a given destination.

- To install:
  - apt-get install traceroute
    - apt-get will be covered in later slides – this is just a reference.

HOW TO SETUP NETWORKING ON UBUNTU

- Must have a network interface

- Use lsmod to list modules inserted into the kernel
  - /etc/modules – file containing modules at boot time
  - /etc/modprobe.d – config files for modules

HOW TO SETUP NETWORKING ON UBUNTU

- Where is the network interface?
  - Can use `dmesg` to help determine interface names and link availability
  - `ifconfig -a`
  - Looking in /dev for stuff that looks right

HOW TO ADD A MACHINE TO A NETWORK?

- Assign a unique IP
- Configure host to boot up with ip address
- Add default routes
  - Allows it access to the internet
- Add a DNS server's IP to the host
  - vi edit the /etc/resolv.conf
    - nameserver 10.0.0.1

MANUALLY ASSIGNING IP ADDRESS

- Quickly configuring IP
  - `ifconfig eth0 10.0.0.2 netmask 255.255.255.0 up`
  - `route add default gw 10.0.0.1`

- Route command adds a default static route
  - Routes subnet mask 0.0.0.0 to 10.0.0.1

PERMANENTLY ADDING IP ADDRESS

- Edit /etc/network/interface and add:

```
auto eth0
iface eth0 inet static # can be static or dhcp
address 10.0.0.2
netmask 255.255.255.0  # this is a /24
gateway 10.0.0.1       # default gateway (optional)
```

- gateway is a static route for 0.0.0.0
- gateway must exist on your local subnet

BRING INTERFACE ONLINE

- /etc/init.d/networking restart
  - Restarting networking can cause all adapters to restart
    - Consider using nohup is connected remotely
    - sudo nohup /etc/init.d/networking restart
- ifup eth0

HOW DO WE CREATE A ROUTER IN UBUNTU?

- Easy!

- Add two network interfaces
- Configure them
- Enable Routing

CREATING UBUNTU ROUTER

- Edit /etc/network/interface and add:

```
auto eth0
iface eth0 inet static
address 10.0.0.106
netmask 255.255.255.0
gateway 10.0.0.1

auto eth1
iface eth1 inet static
address 10.0.12.1
netmask 255.255.255.0
```

- Edit /etc/sysctl.conf and uncomment line to:
  - `net.ipv4.ip_forward=1`

WORKING WITH OTHER SYSTEMS

OBJECTIVE

- Discuss what Linux systems may connect to
- Discuss what purpose system administrators may have for supporting other systems
- Understand why Linux is an integral part of the entire enterprise

IT'S NOT ALL ABOUT LINUX

- You can have just Linux in your environment
- Traditionally, you won't be able to just have Linux however
- Linux does not do well at everything
  - Desktop environment in an enterprise
  - Windows applications

AUTHENTICATION

- Linux may be joined to an Active Directory domain
- This serves the following purposes:
  - Security
  - File Sharing
  - Trust

LINUX AND AD

- Linux can be authenticated to the domain in three different ways:
  - LDAP
  - AD to LDAP
  - AD to LDAP and Kerberos
- Installation may be complicated
  - SAMBA
  - PAM
  - WinBind

WHY DO THIS?

- In an AD environment, authentication is necessary
- Auditing is necessary
- Local accounts should never be used

LDAP FOR CUSTOM SYSTEMS

- Linux installations may use third party open source for software
- Software may require authentication
- LDAP is typically used

CONCLUSION

- While Linux can stand on its own, in an enterprise, integration is necessary
- Security is a major consideration

SELINUX

OBJECTIVES

- Describe what SELINUX is
- Understand why we don’t just disable it
- Discuss how SELINUX provides Least Privilege

PROBLEMS WITH DAC

- Administrators cannot control every user
- File permissions and ACLs cant protect against everything
- Processes can change permissions and other security properties
- Compromised software may have access to other parts of the system

SELINUX

- Stands for Security-Enhanced Linux
- Originally developed by the NSA to show the value of Mandatory Access Control
- Built into the Linux kernel as of kernel 2.6
- Primarily used in RedHat and related distributions – Fedora, CentOS, and Scientific Linux

SELINUX BREAKDOWN

- Deny by default
- Log everything
- Only allow exceptions one at a time

SELINUX MODES

- Enforcing – Default mode which will enforce the SELinux security policy on the system
- Permissive – SELinux is enabled, but does not enforce. Also logs. Good for troubleshooting
- Disabled – SELinux is disabled. Never use this.

TYPES OF ENFORCEMENT

- Type Enforcement – Primary control which uses the “targeted” policy
- Role-Based Access Control – Enforcement based on a users role
- Multi-Level Security – Used in the “targeted” policy, but generally hidden and not often used
- Multi-Category Security – Extension of Multi-Level, but

FEATURES AND BENEFITS

- Policies are separate from enforcement
- Great logging
- Controls much of the OS such as files, processes, network, process execution

DOWNFALLS

- Can be time consuming to understand
- Complex
- While everything is logged, finding what may have been denied can be difficult
- If you make a change, you have to submit the new context

TYPICAL USAGE

- A web server might use SELinux to block a user from creating content within a directory
- Users might only be able to access certain files
- Secure systems might employ SELinux for least privilege across many areas

CONCLUSION

- SELinux provides powerful control over a system
- Use it if you want to have a very secure system
- Time and patience may be needed to configure it

LINUX AND SECURITY

OBJECTIVES

- Understand what threats Linux faces
- Discuss what items administrators need to be concerned with
- Understand there is no such thing as a secure computer

IS LINUX SECURE?

- Linux can be configured in a way that it is as secure as every other OS out there
- NO COMPUTER OR OS IS COMPLETELY SECURE!
- Linux can be configured nearly any which way you want it to be configured
- This also provides pitfalls if we provide secure measures incorrectly

BENEFITS OF LINUX FOR SECURITY

- Most software on Linux is open source – so you have communities of developers helping to secure it
- The Linux kernel itself is relatively secure
- Software usually has less privileges
- Size is smaller, so software, not kernel is sought after

HOW LINUX MAY BE COMPROMISED

- Software vulnerabilities
- Configurations errors
- Social Engineering or Users in general
- Rootkits, Viruses, and Trojans

SOFTWARE VULNERABILITIES

- Buffer overflows are still the number one software vulnerability
- Linux software is not infallible
- “Linus Law” states – “given enough eyeballs, all bugs are shallow” – meaning the more people you have looking at software the more it is secure and bugs are patched, however some bugs still make it through
- Developers of custom software may not have luxury of testing software
- Software may not be patched

CONFIGURATION ERRORS

- Since software needs to be installed in a certain, typically, in Linux, configuration pages may have enhanced privildges
- It's very easy to do something in Linux
- It's very hard to undo something in Linux
- Forgetting to close ports or remove configuration pages is a common issue

SOCIAL ENGINEERING AND USERS

- Even though it is the job of the system administrator to make sure systems are secure, humans are not 100% infallible
- Users may make mistakes
- Amazon and other larger corporations have been taken down because of a simple, inadvertant command

ROOTKITS, VIRUSES, AND TROJANS

- There are still rootkits, viruses and Trojans that are developed for Linux, but not in the same ballpark as other OS's
- Moris Worm in 1988 was the first Linux worm
- Linux has open source AV however
  - chrootkit
  - Rkhunter
  - ClamAV

CONCLUSION

- Administrators need to be diligent in securing Linux systems
- While Linux does not contract viruses in a traditional sense typically, they are a target for hackers who wish to exploit software vulnerabilities