RCHAIN

COOPERATIVE

Hosting Your Own Data Center from Home

Hosting Your Own Secure Computer Infrastructure From Home

Goals

Expanding performant compute infrastructure to the edge

Build with:

- CIA - Confidentiality, Integrity, Availability
- Proper scaling architecture
- Security
- Ease of management and maintenance

Choice of software is important!!!

Use and support open projects whenever possible. This means financially too.

ISC2 Code of Ethics - https://www.isc2.org/Ethics

There are only four mandatory canons in the Code. By necessity, such high-level guidance is not intended to be a substitute for the ethical judgment of the professional.

Code of Ethics Preamble:
The safety and welfare of society and the common good, duty to our principles, and to each other, requires that we adhere, and be seen to adhere, to the highest ethical standards of behavior.
Therefore, strict adherence to this Code is a condition of certification.

Code of Ethics Canons:
Protect society, the common good, necessary public trust and confidence, and the infrastructure.
Act honorably, honestly, justly, responsibly, and legally.
Provide diligent and competent service to principles.
Advance and protect the profession.

Certifications on the Path to Linux or Security

- Linux
  - https://www.lpi.org/our-certifications/lpic-1-overview
  - http://www.lpi.org/our-certifications/lpic-2-overview
- Security
  - https://www.isc2.org/Certifications/CISSP
  - https://www.sans.org/
- Networking
  - https://learningnetwork.cisco.com/community/certifications/ccna
  - https://www.juniper.net/us/en/training/certification/
  - https://cumulusnetworks.com/support/networking-training/
  - Etc ...

Why?

- Fun
- Educational
- For ownership, control, decentralization and distributed compute.
  - With the cloud, you don't own anything. You already signed it away. - Steve Wozniak
  - Nice for getting started but bad for large entities. Easy but lazy. Ancestry.com
  - Rights to YOUR data get different in cloud infrastructure.
- For decentralized storage and distributed compute.
  - The future is a mix of localized and centralized compute. A good example is using your phone. I do compute locally and for voice translation I send my voice command to a network API that will use a specialized compute platform to perform its function in the most effective way possible. The API will then give your device a response back.
  - Localized storage caches are important. CDNs make a huge difference.

Cloud Computing vs Localized Home Computing

Who is hosting on the public cloud right now?

Who is hosting some service on their private cloud right now? From home?

Is any service or data from your home publically available on the world wide web?

- NGINX and static website
- Blog/Vlog
- Documentation - Sphynx, Mediawiki, Dokuwiki, Confluence
- Content Management System CMS -Wordpress, Joomla
- Code/Ticketing/Project Management - GitLab, Redmine, Request Tracker
- Simple hardened bastion host or limited host with ssh access

Networks are Changing - Bandwidth is Changing

Centralized compute has given us a lot of benefits.

Problem of the pendulum.

Centralized->decentralized->centralized->decentralized. Why?

Optimization technology for decentralized compute and data.

This doesn’t have to necessarily take place on RChain or blockchain network.

It could. My preference is decentralized compute in general. **Power corrupts ...**

Fiber Optic Technology to the Rescue

Optic vs Copper. Light vs Current. Why is fiber so much better?

Wave Division Multiplexing

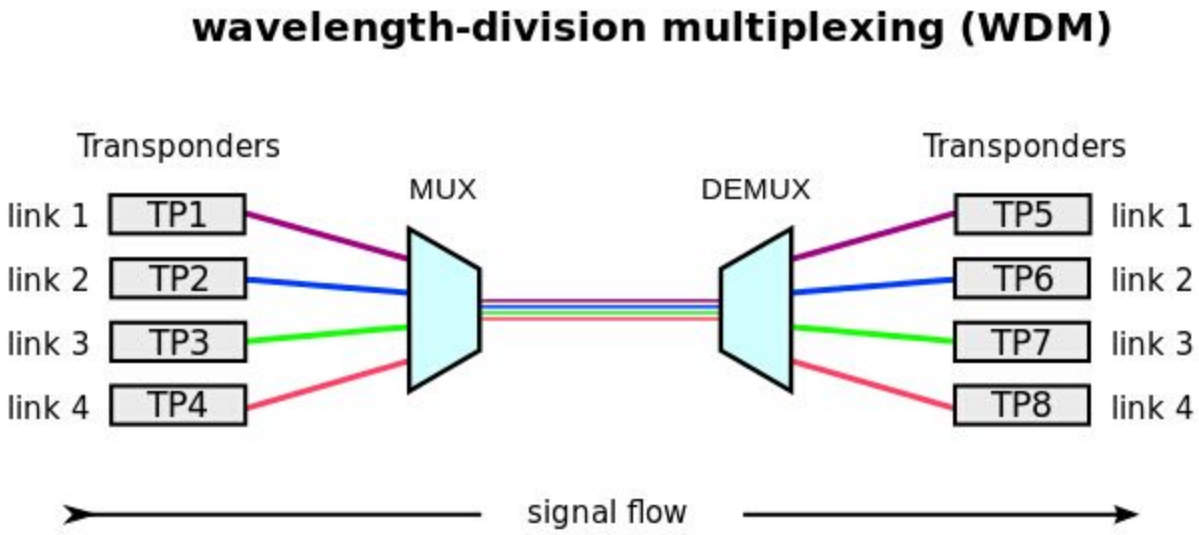
WDM - https://en.wikipedia.org/wiki/Wavelength-division_multiplexing

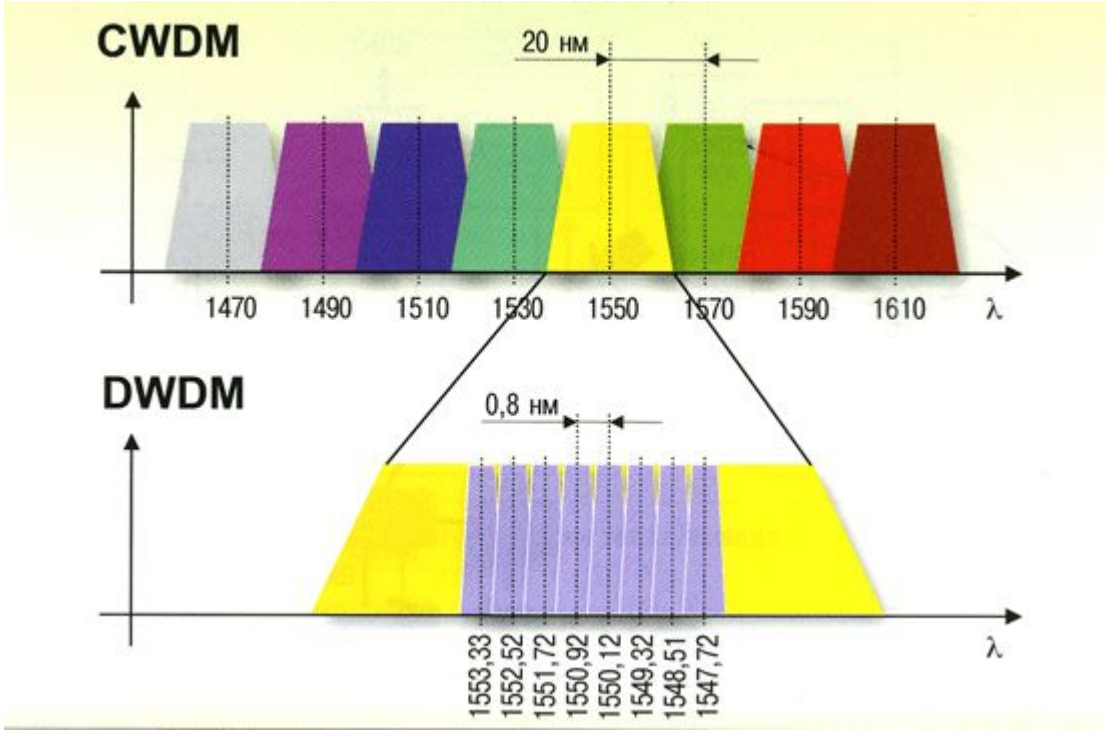
Coarse WDM - older tech

Dense WDM - newer tech, more light waves, more bandwidth

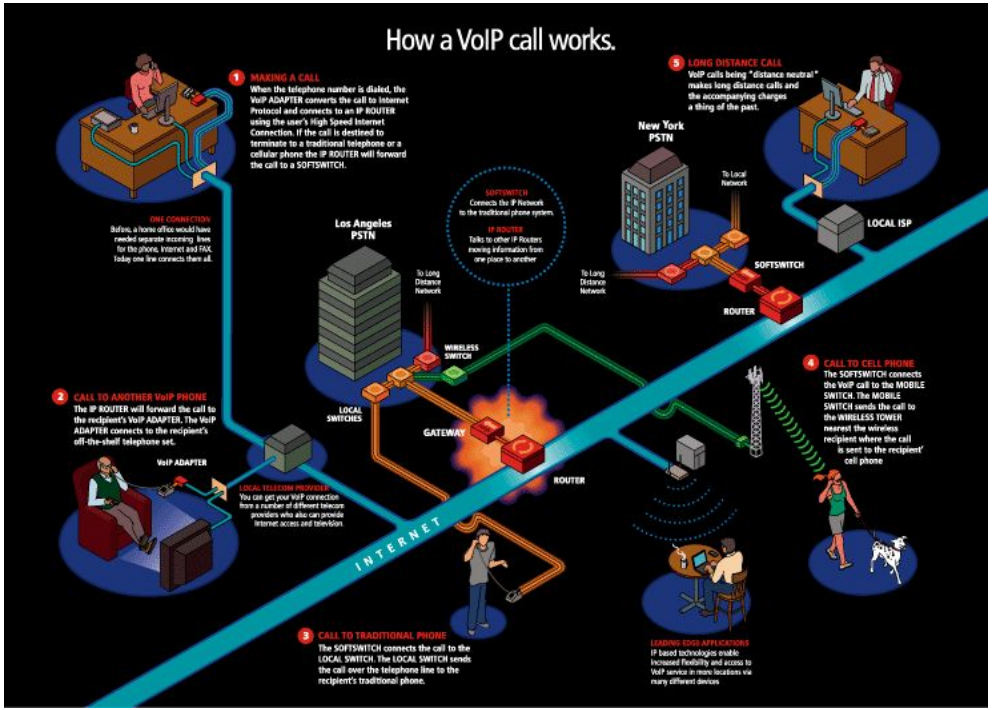
Manufacturing Improvements. Economies of scale at work.

Corning in NY - China and other parts of the world are helping drive costs down.


wavelength-division multiplexing (WDM)
Transponders
MUX
DEMUX
Transponders
link 1
TP1
link 2
TP2
link 3
TP3
link 4
TP4
TP5
link 1
TP6
link 2
TP7
link 3
TP8
link 4
signal flow


CWDM
20 нм
1470
1490
1510
1530
1550
1570
1590
1610
λ
DWDM
0,8 нм
1553,33
1552,52
1551,72
1550,92
1550,12
1549,32
1548,51
1547,72
λ

Communication Has Become Easier/Cheaper



Voice and SMS Have Become Easy/Cheap

Google Voice, Magic Jack, Skype, OpenSIPS, Kamailio, Asterisk

Twilio - https://www.twilio.com/

Nexmo - https://www.nexmo.com/

Freedom Voice - https://www.freedomvoice.com/ - Godaddy

$1

Nice libraries to mix into your applications

WebRTC - https://en.wikipedia.org/wiki/WebRTC - https://github.com/webrtc

Decentralized Networks Demand Bandwidth

Netflix, Youtube and Facebook were a big driver for residential bandwidth.

- Municipal ftth(fiber-to-the-home) networks
- Google Fiber
- ATT, Verizon, Comcast and other large and small ISPs still struggle to bring good service and bandwidth.
- Mom and cable internet experience. Why is it so hard?
- Demand is high but why is demand under served?
- Utopia and different financing and construction models.
- Funding from cryptocurrency?
- SpaceX

SpaceX & the Rise of Satellite Internet

Wireless always will be slow. Wireless & wired will always be needed.

Wireless has gotten better.

Google balloons, Facebook solar plane, low Earth orbital satelites.

SpaceX recently launched two statistical collection satellites. Microsats.

These two stat collection satelites will gather data to test and demonstrate the viability of SpaceX building a constellation of 4,425 Ka/Ku band low Earth orbit satellites.

Encourage Network Bandwidth

I really feel that private sector should build and maintain the networks the they should be operated and owned by citizens.

Politics get in the way of progress.

As we work on blockchain and other networked technologies we really need people working on and promoting solutions to support the infrastructure and improve performance.

Fiber everywhere? 100Gig everywhere?

Understanding How Networking Works

Understanding is the first step to defending.

CCNA BootCamp

Linux has a lot of tools

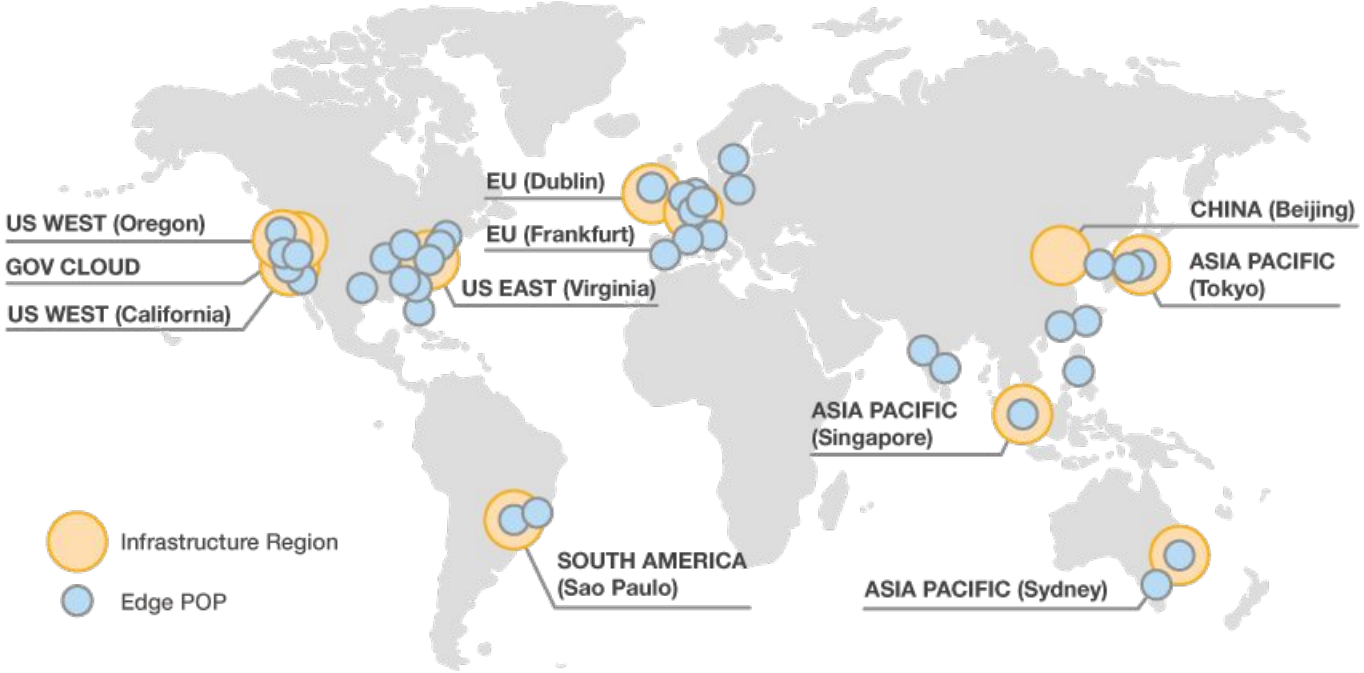
ip, brctl, iptables, ebtables, suricata, docker, lxd - script your own

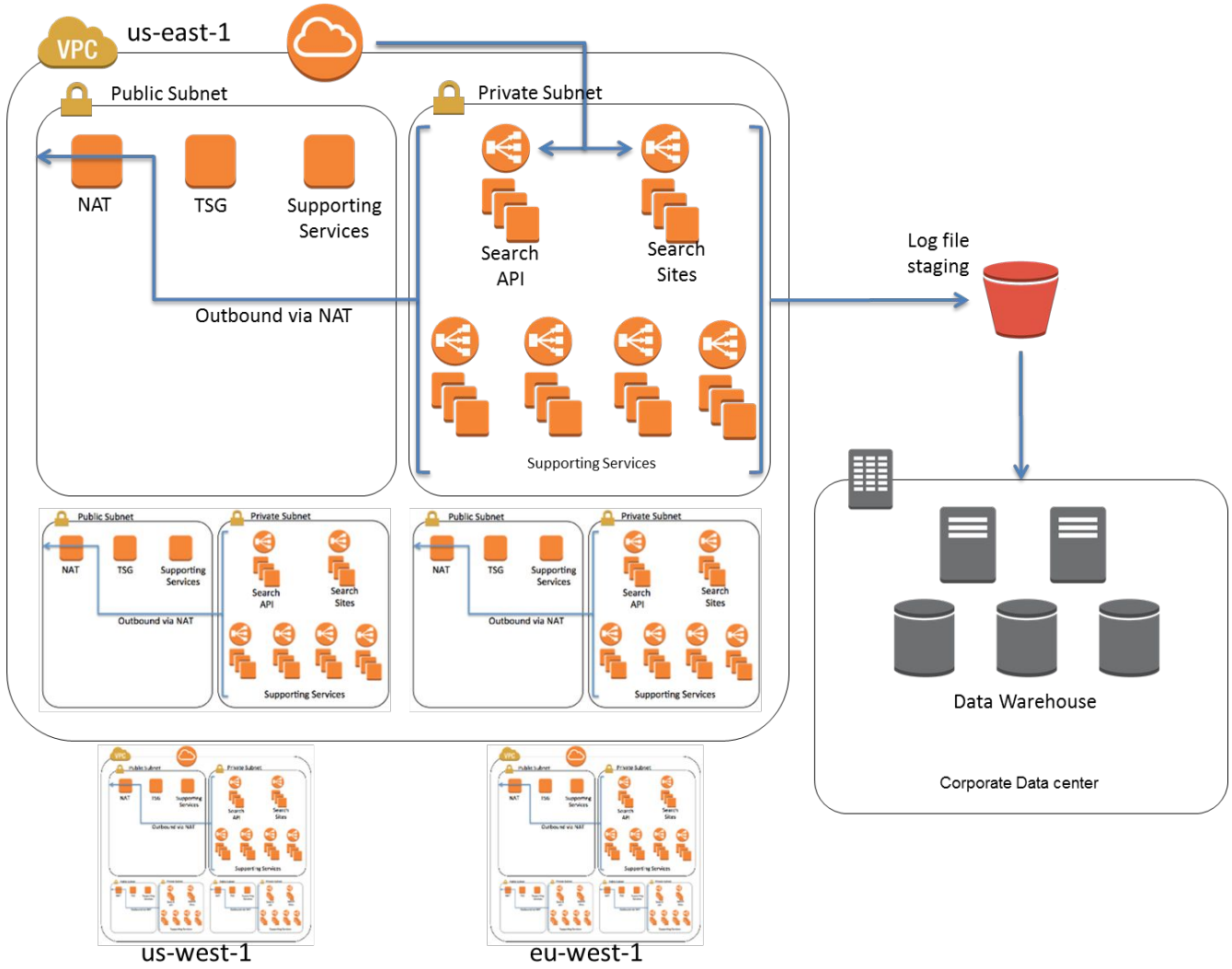
Mininet - http://mininet.org/

Eve-NG -http://www.eve-ng.net/ - better than GNS3 - web accessible

http://www.boson.com/netsim-cisco-network-simulator

Learn from AWS or GCE - They've been around ...



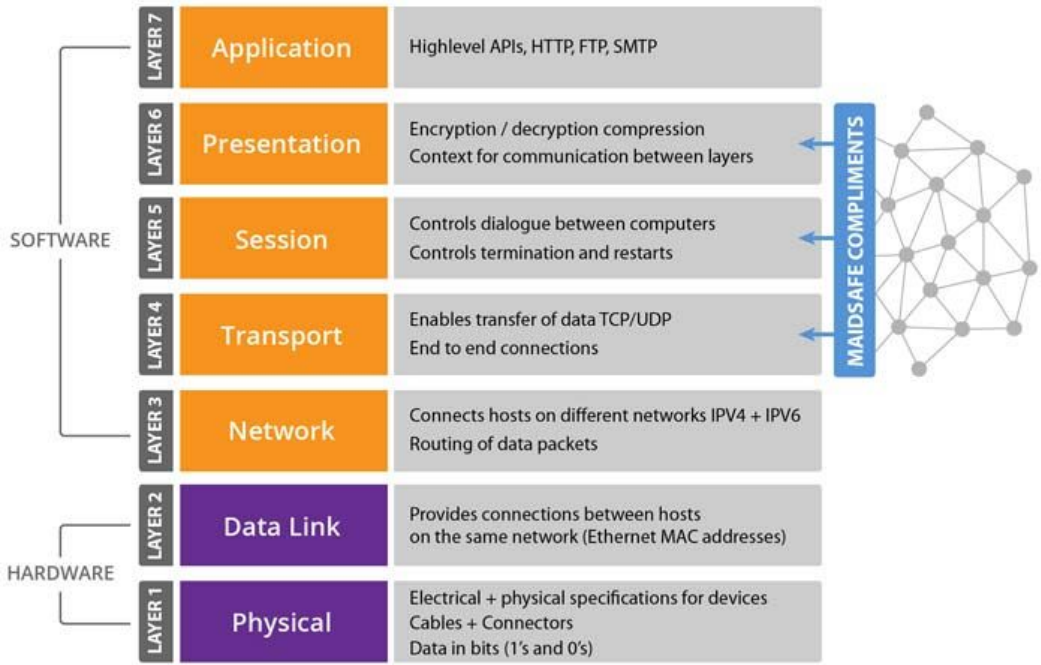

VPC
us-east-1
Public Subnet
Private Subnet
NAT
TSG
Supporting Services
Search API
Search Sites
Outbound via NAT
Supporting Services
Log file staging
Data Warehouse
Corporate Data center
us-west-1
eu-west-1

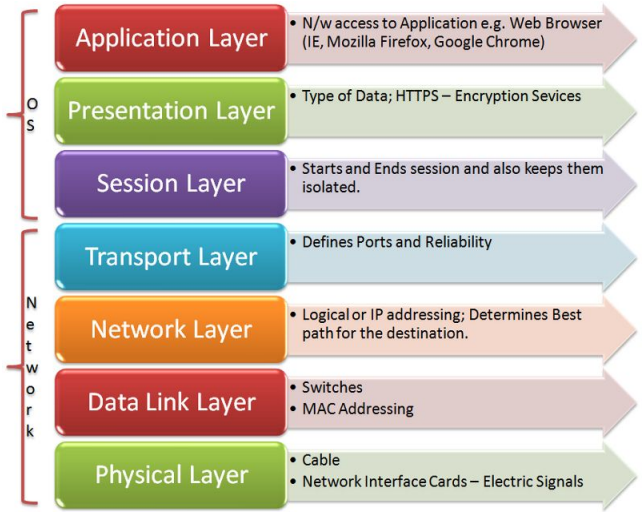
CDNs and the Emergence of Micro Data Centers

https://en.wikipedia.org/wiki/Micro_Data_Center

- Content Delivery Networks
- Carrier Data Centers
- Service Provider Data Centers
- Micro Data Centers for Edge
- Homes could be part of this. Even faster cache. P2P networks could be doing this soon, with or without blockchain.

OSI Model - Sooner is ALWAYS Better than Later





Starting with the Physical Layer

https://www.ebay.com/itm/Build-Your-Own-HP-Proliant-DL380-G7-Server-2x-X5650-2-66GHz-12-Hex-Core-DVD-RPS/172744071499?hash=item283859594b:m:m-sv6eVQt-Q5Ro-wFcgKbXw

Amazon, NewEgg, Best Buy, Costco, Walmart, Alibaba Express

Ebay, Amazon Used and many other liquidation sites or vendors.

http://www.acmemicro.com/

https://www.fs.com/

Managing Soft & Hard Network Resources

Netbox - https://github.com/digitalocean/netbox - DCIM

Racktables - https://www.racktables.org/ - DCIM

NocProject - https://github.com/nocproject/noc - IPAM+

Infrastructure as Code - https://www.terraform.io/ - declarative and reviewable

SaltStack - https://saltstack.com/ - Puppet, Chef, Ansible …

Monitoring - Zabbix, ZenOSS, OpenNMS, Nagios, NFDUMP, Prometheus, scripts

Logs - Elastic Stack - https://www.elastic.co, Splunk Free tier

Monitoring - Active vs Passive vs Simple

Install monitoring agent on host if possible. This launched Nagios.

Active Checks - Host -> Monitoring Server - push method - great for firewalled

Passive Checks - Monitoring Server -> Host - pull method

Simple Checks - Agent is not loaded on host and is ran from monitor Server

Use proxies liberally to scale. They collect, aggregate and forward to central storage.

Use data partitioning to scale as well for time series data. There are scripts out there for different databases and monitoring systems.

Metrics are Simple

Item ID: primary key - this is mainly tied to a host and a metric type - i.e. CPU

Value: decimal - this is the actual value of metric at a point in time

Timestamp: timestamp/datetime - time collected or time received maybe both

Many times, timestamp will be stored in nanoseconds int for ease of consume

Visualization

Grafana - this pairs well with Prometheus and often is

D3.js, C3.js or HighCharts - https://www.highcharts.com/ - json to pretty graph

Hardware is Cheap

Some good resources for infrastructure building

- http://www.acmemicro.com/
- https://www.fs.com/
- https://www.newegg.com/
- https://www.amazon.com/
- https://www.ebay.com/
- https://www.alibaba.com/
- Costco, Walmart, Bestbuy, Dell, Lenovo, HP, ASUS or local computer store

You local electrical supply store can helpful as well.

Hardware is Powerful

NVME is fast. At $199 EVO is a great economical way to get started

- https://www.amazon.com/Samsung-960-EVO-Internal-MZ-V6E250BW/dp/B01M20VBU7?th=1
- Storage Capacity: 500 GB.
  Form Factor: NVMe M.2.
  Sequential Read: 3,200MB/s.
  Sequential Write: 1,800MB/s.
  Interface: PCIe 3.0 x4, NVMe 1.2.

Getting IP Space & Having it Advertised

ARIN

https://www.arin.net/fees/fee_schedule.html

https://www.arin.net/knowledge/getaddresses.pdf

https://www.youtube.com/watch?v=yjITQGEb6gg&feature=youtu.be

https://www.arin.net/resources/request.html#submit

Getting DNS - OpenSRS, AWS, Google, GoDaddy, NameCheap

Storage Rule #1 - Use ZFS - Ceph

https://www.thegeekdiary.com/solaris-zfs-command-line-reference-cheat-sheet/

http://www.freenas.org/

Ceph - https://en.wikipedia.org/wiki/Ceph_(software) - https://ceph.com/

Moose/Lizard FS but I would probably stick with ceph

FreeNAS for Shared Storage

ZFS

Snapshots

Enable jumbo frames. In GUI and go to Network. Then choose your network device and add mtu 9000 for Jumbo Frames of 9000 Byte.

jumbo frame is an Ethernet frame with a payload greater than the standard maximum transmission unit (MTU) of 1,500 bytes.

Jumbo frames are used on local area networks that support at least 1 Gbps and can be as large as 9,000 bytes. Because jumbo frames are not defined in the IEEE 802.3 specifications for Ethernet, vendor support for jumbo frames and their maximum transmission units may vary.
You usually only set jumbo frames on your storage switching infrastructure.

Using a Firewall DMZ to Mitigate Risks

Beware

- Beware many Linksys and other home routers. This may change in the future.
  - Linksys DMZ is not true DMZ just Destination NAT.
  - It does not isolate your host from the rest of your network, just the internet ips.
  - https://www.linksys.com/us/support-article?articleNum=140747
  - A true DMZ is completely isolated from the rest of the network on its own ip range or isolated layer2 as well via vlans. Ebtables
- Alternatives
  - Linux - iptables/ebtables
  - BSD Open/Free - ipfw - OpenBSD probably is still considered most secure.

Build Your Own Wifi Router Using pfSense or Linux

Small firewall/router or wifi firewall/router using pfSense or pure Linux

https://www.amazon.com/gp/product/B01KX9OU58

https://www.amazon.com/Q190G4N-S07-Industrial-Gateway-Firewall-pfSense/dp/B01CSCGD58

dd-wrt - https://www.dd-wrt.com

More packages from community

Rasberry pie - slower but works

Linux Security Tools

- iptables - layer 3/4 filtering
- ebtables - layer 2 filtering. Great for logical isolation of hosts on same network.
- If on lxdbr0 or could use Docker0
  apt install ebtables
  <my-bridge-name> is bridge you want to isolate all host traffic on.
  ebtables -A FORWARD --logical-in <my-bridge-name> -j DROP
  ebtables -A FORWARD --logical-out <my-bridge-name> -j DROP
- https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Setting_up_IPSinline_for_Linux
-
- sudo iptables -I FORWARD -j NFQUEUE
  In this case, all forwarded traffic goes to Suricata.
  In case of the host situation, these are the two most simple iptable rules;
  sudo iptables -I INPUT -j NFQUEUE
  sudo iptables -I OUTPUT -j NFQUEUE
- AF_PACKET
- Nmap
- OpenVAS
- Scapy

Multi Tenancy Architecture

- Multi tenancy applications and architecture is not new
- Logical partitioning has been around for ages
- Mainframe, server cluster, database
- Protective Isolation Methods
  - Physical, logical, authenticated, authorized, cryptographic etc.
- Cryptography is special. Algorithms will always change as compute power changes.

Fail2Ban

Very easy to setup. Nice for website hosting.

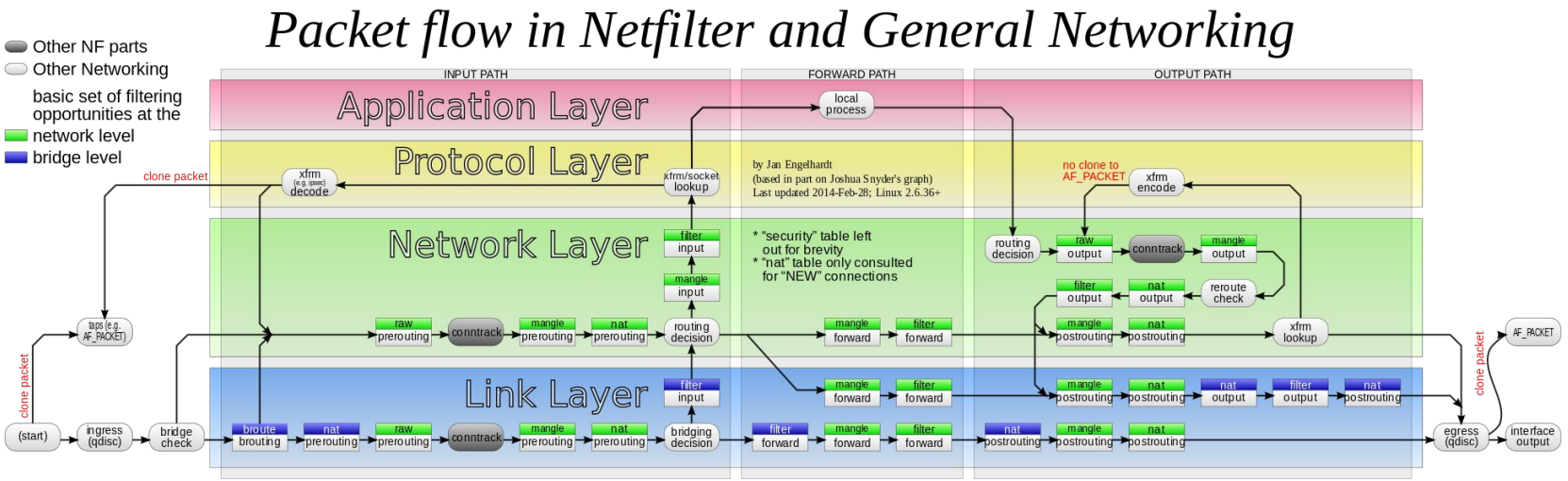
Fail2ban scans log files (e.g. /var/log/apache/error_log) and bans IPs that show the malicious signs -- too many password failures, seeking for exploits, etc. Generally Fail2Ban is then used to update firewall rules to reject the IP addresses for a specified amount of time, although any arbitrary other action (e.g. sending an email) could also be configured. Out of the box Fail2Ban comes with filters for various services (apache, courier, ssh, etc).

Fail2Ban is able to reduce the rate of incorrect authentications attempts however it cannot eliminate the risk that weak authentication presents. Configure services to use only two factor or public/private authentication mechanisms if you really want to protect services.

You can easily write your own blocking by tailing apache, nginx or application logs and blocking ip depending on behavior. Python is nice for this.

Netfilter Packet Flow

This is a common flow for any network filter stack.



Using Suricata or Snort to do IDS/IPS

You can always write rules within your application.

- Rules - these define how to treat traffic depending on network source or destination and what to do when certain traffic patterns or content are matched.
- Events->Triggers->Actions
- Actions - notifications, api calls, or immediate prevention by dropping packets or disabling ip.

Writing Suricata and Snort Rules
http://www.vorant.com/files/EZ_Snort_Rules.pdf
https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricata_Rules

Snort vs Suricata

Suricata is newer and so was built better from learning from the past.

Suri is multithreaded and context/protocol aware.

Snort rules say "this rule can fire on traffic on port 80,8080,8081". Suricata rules say "this rule fires on HTTP traffic". So it catches stuff on unusual ports, or unusual stuff on normal ports. Snort 3.0 is currently in alpha. Go with Suricata for now.

Suricata and NFQ

https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Setting_up_IPSinline_for_Linux

NFQUEUE is an iptables and ip6tables target which delegate the decision on packets to a userspace software. For example, the following rule will ask for a decision to a listening userpsace program for all packet going to the box:

iptables -A INPUT -j NFQUEUE --queue-num 0

suricata --build-info
and examine if you have NFQ between the features.

To run suricata with the NFQ mode, you have to make use of the -q option. This option tells Suricata which of the queue numbers it should use.

sudo suricata -c /etc/suricata/suricata.yaml -q 0

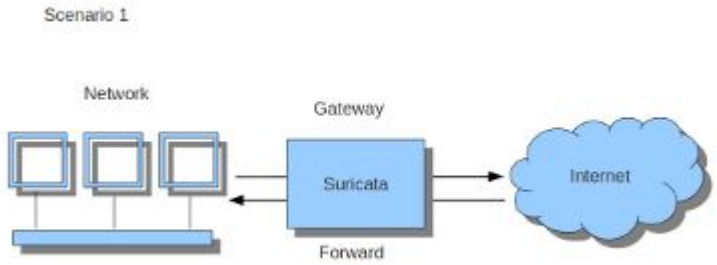
AFPACKET vs NFQ - http://sublimerobots.com/2017/06/snort-ips-with-nfq-routing-on-ubuntu/

You can run Snort two different ways in inline mode, with AFPACKET or with NFQ. AFPACKET is simpler to setup but only lets you bridge sets of paired interfaces.

Using NFQUEUE to do InLine Filtering

sudo iptables -I FORWARD -j NFQUEUE

In this case, all forwarded traffic goes to Suricata.



In case of the host situation, these are the two most simple iptable rules;

sudo iptables -I INPUT -j NFQUEUE

sudo iptables -I OUTPUT -j NFQUEUE

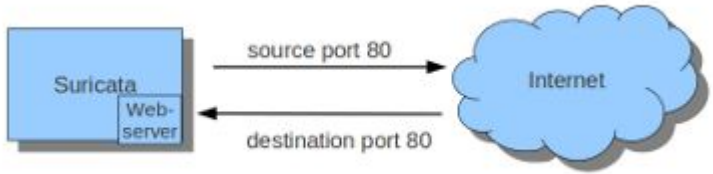
It is possible to set a queue number. If you do not, the queue number will be 0 by default.

TCP or TCP Port 80

Imagine you want Suricata to check TCP traffic or all incoming traffic on TCP port 80. How about all traffic on destination-port 80.

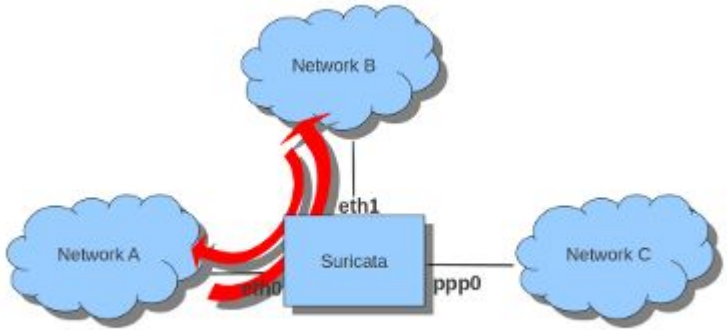
sudo iptables -I INPUT -p tcp -j NFQUEUE
sudo iptables -I OUTPUT -p tcp -j NFQUEUE
In this case, Suricata checks just TCP traffic.

sudo iptables -I INPUT -p tcp --sport 80 -j NFQUEUE
sudo iptables -I OUTPUT -p tcp --dport 80 -j NFQUEUE



Suricata on Specific Interfaces

sudo iptables -I FORWARD -i eth0 -o eth1 -j NFQUEUE
sudo iptables -I FORWARD -i eth1 -o eth0 -j NFQUEUE



ADC - NGINX or OpenResty(NGINX+LUA)

LUA is fast. Faster than TCL which is what F5 uses. Most ADCs will use LUA or TCL.

A.I. - Torch uses LUA

With OpenResty you can look into the HTTP Headers and Payload (after decrypt) and add/delete/modify headers. Look for malicious intent or modify payload, like json, so it can’t do bad things to your data store.

REST is handy in that it is a defined protocol.

Use an Application Delivery Controller ADC

HAProxy - http://www.haproxy.org/

NGINX

OpenResty (NGINX+LUA) - Cloudfare supported and used for quite a while.

PostgREST - https://postgrest.com

PostGraphile - https://www.graphile.org/postgraphile/introduction/

I like NGINX or OpenResty depending what I’m doing. I like PostgreSQL for db. It now performs row level security.

Network Flows Using nfdump

https://github.com/phaag/nfdump

There are lots of flow collectors, pmacct, ntop and others.

You can write your own.

I find NFDUMP to be the best Open Source project. C, very fast. Used for security.

Commercial cloud and

https://www.manageengine.com/products/netflow/

Default Docker Firewall and Other Containers

For security reasons, Docker configures the `iptables` rules to prevent containers from forwarding traffic from outside the host machine, on Linux hosts. Docker sets the default policy of the `FORWARD` chain to `DROP`.

By default, always drop inbound connections. Only allow ports in you want to. UPnP can help dynamically open up ports on use.

Stateful firewalls, which are most firewalls these days, allow you to track NEW, ESTABLISHED, RELATED sessions and dynamically allow inbound traffic based on outbound.

Host Operating Systems

Use what you want but not all are created equal.

Linux, Windows, OSX, Free/Open BSD ...

I personally would use Linux or FreeBSD. OpenBSD if you wanted even more security. I think there are a lot of eyes on Linux these days.

For Linux choose a distro. Again, choose what you want but not all distros are created equal. It usually ends up being Debian/Ubuntu vs CentOS/Redhat. I feel either is a pretty safe choice. Both have their strengths and weaknesses. Redhat dominates Financial and sensitive governmental functions in many areas.

SELinux vs AppAmor - context security

Using Virtualization for Logical Isolation

- Use virtualization. Performance loss is usually minimal compared to gains in management. Security is pretty good and in many ways better because of forced containment.
  - Overhead: Numbers are all over but Hypervisor %25, Containers %5 but this varies widely. It is getting better.
- Everyone is doing this. Cloud platforms rely heavily on virtual environments.
- For Hypervisor Proxmox VE, VMWare KVM
- For containers I would use LXD, LXC or Docker.
- Use ZFS
  - BTRFS still has a long way to go.

Hypervisors

KVM vs Docker vs LXD vs BHyve

KVM, Kernel-based Virtual Machine, is a hypervisor built into the Linux kernel. It is similar to Xen in purpose but much simpler to get running. Unlike native QEMU, which uses emulation, KVM is a special operating mode of QEMU that uses CPU extensions (HVM) for virtualization via a kernel module.

Using KVM, one can run multiple virtual machines running unmodified GNU/Linux, Windows, or any other operating system. (See Guest Support Status for more information.) Each virtual machine has private virtualized hardware: a network card, disk, graphics card, etc.

Differences between KVM and Xen, VMware, or QEMU can be found at the KVM FAQ.

ProxMox VE

https://www.proxmox.com/en/proxmox-ve

https://www.proxmox.com/en/proxmox-ve/comparison

- Open Source
- Debian based
- Commercial support available if wanted
- Not libvirt API - https://libvirt.org/
  - Libvrit supports KVM, QEMU, Xen, Virtuozzo, VMWare ESX, LXC, BHyve
- https://pve.proxmox.com/wiki/Proxmox_VE_API

https://www.vmware.com/products/esxi-and-esx.html

Linux Containers Anatomy

LXC tries to create an environment as close as possible as a standard Linux installation but without the need for a separate kernel. LXC uses these Linux kernel features:

- Kernel namespaces (ipc, uts, mount, pid, network, and user)
  - Provide basic isolation, making sure that each container cannot “see” or affect other containers.
- Control groups (cgroups)
  - Used to allocate resources (memory, CPU, I/O...) between containers.
- AppArmor profiles
  - Straight forward and easy to verify but limited compared to selinux.
- SELinux profiles
  - Incredibly complex but with this complexity you have more control over how processes are isolated. Hard to verify.
- Secure computing mode (seccomp) policies
  - Linux kernel feature. You can use it to restrict the actions available within the container.
- Chroots (using pivot_root)
- Kernel capabilities

LXC relies on two significant feature sets of the kernel: kernel namespaces, and control groups. Kernel namespaces provide basic isolation, making sure that each container cannot “see” or affect other containers.

Control groups are used to allocate resources (memory, CPU, I/O...) between containers.

Full Virtualization & Hypervisors

What if I want to run different operating systems? Well, you can split workload into different container hosts based on operating system. This will be many of the environments. Hypervisors aren’t going away. There will be a use for these. Use the right tools for specific needs.

The one thing that hypervisors can do that containers can’t, is to use different operating systems or kernels. For example, you can use VMware vSphere to run instances of Linux and Windows at the same time. With LXC, all containers must use the same operating system and kernel. In short, you can’t mix and match containers the way you can VMs.

More tools

- OSSEC, Tripwire
- Nmap
- OpenVAS, MetaSploit - Nessus
- pfSense
- Palo Alto
- Scapy - http://www.secdev.org/projects/scapy/
- Kali Linux - https://www.kali.org/
- https://www.trustedsource.org/
- https://www.spamhaus.org/

Fin

Questions?

TCP vs UDP

Transmission Control Protocol is a connection-oriented protocol, which means that it requires handshaking to set up end-to-end communications. Once a connection is set up, user data may be sent bi-directionally over the connection.

Reliable – Strictly only at transport layer, TCP manages message acknowledgment, retransmission and timeout. Multiple attempts to deliver the message are made. If it gets lost along the way, the server will re-request the lost part. In TCP, there's either no missing data, or, in case of multiple timeouts, the connection is dropped. (This reliability however does not cover application layer, at which a separate acknowledgement flow control is still necessary)
Ordered – If two messages are sent over a connection in sequence, the first message will reach the receiving application first. When data segments arrive in the wrong order, TCP buffers delay the out-of-order data until all data can be properly re-ordered and delivered to the application.
Heavyweight – TCP requires three packets to set up a socket connection, before any user data can be sent. TCP handles reliability and congestion control.
Streaming – Data is read as a byte stream, no distinguishing indications are transmitted to signal message (segment) boundaries.
User Datagram Protocol is a simpler message-based connectionless protocol. Connectionless protocols do not set up a dedicated end-to-end connection. Communication is achieved by transmitting information in one direction from source to destination without verifying the readiness or state of the receiver.

Unreliable – When a UDP message is sent, it cannot be known if it will reach its destination; it could get lost along the way. There is no concept of acknowledgment, retransmission, or timeout.
Not ordered – If two messages are sent to the same recipient, the order in which they arrive cannot be predicted.
Lightweight – There is no ordering of messages, no tracking connections, etc. It is a small transport layer designed on top of IP.
Datagrams – Packets are sent individually and are checked for integrity only if they arrive. Packets have definite boundaries which are honored upon receipt, meaning a read operation at the receiver socket will yield an entire message as it was originally sent.
No congestion control – UDP itself does not avoid congestion. Congestion control measures must be implemented at the application level.
Broadcasts - being connectionless, UDP can broadcast - sent packets can be addressed to be receivable by all devices on the subnet.

https://pve.proxmox.com/wiki/Cluster_Manager

Pveam update
pveam available
https://pve.proxmox.com/wiki/Command_line_tools

https://pve.proxmox.com/wiki/Linux_Container