Maciej Lasyk

Ganymede

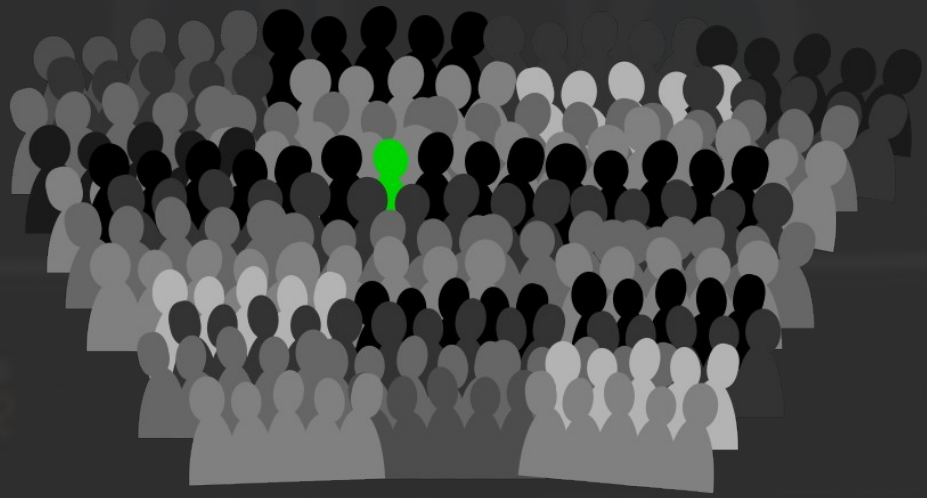
OWASP Poland, 2013-10-17

Recruitment process @OWASP?

- Because this system is web application (partially)
- Because we based (100%) on FOSS (open-source)
- Because security matters
- Because OWASP people cares about security and can affect recruitment processes (hopefully) ;)

Recruitment

- Lot of recruitment agencies / services
- Huge number of potential candidates
- Whole team is involved in recruitment
- Candidate evaluation takes really lot of time



SysAdmin / Operations

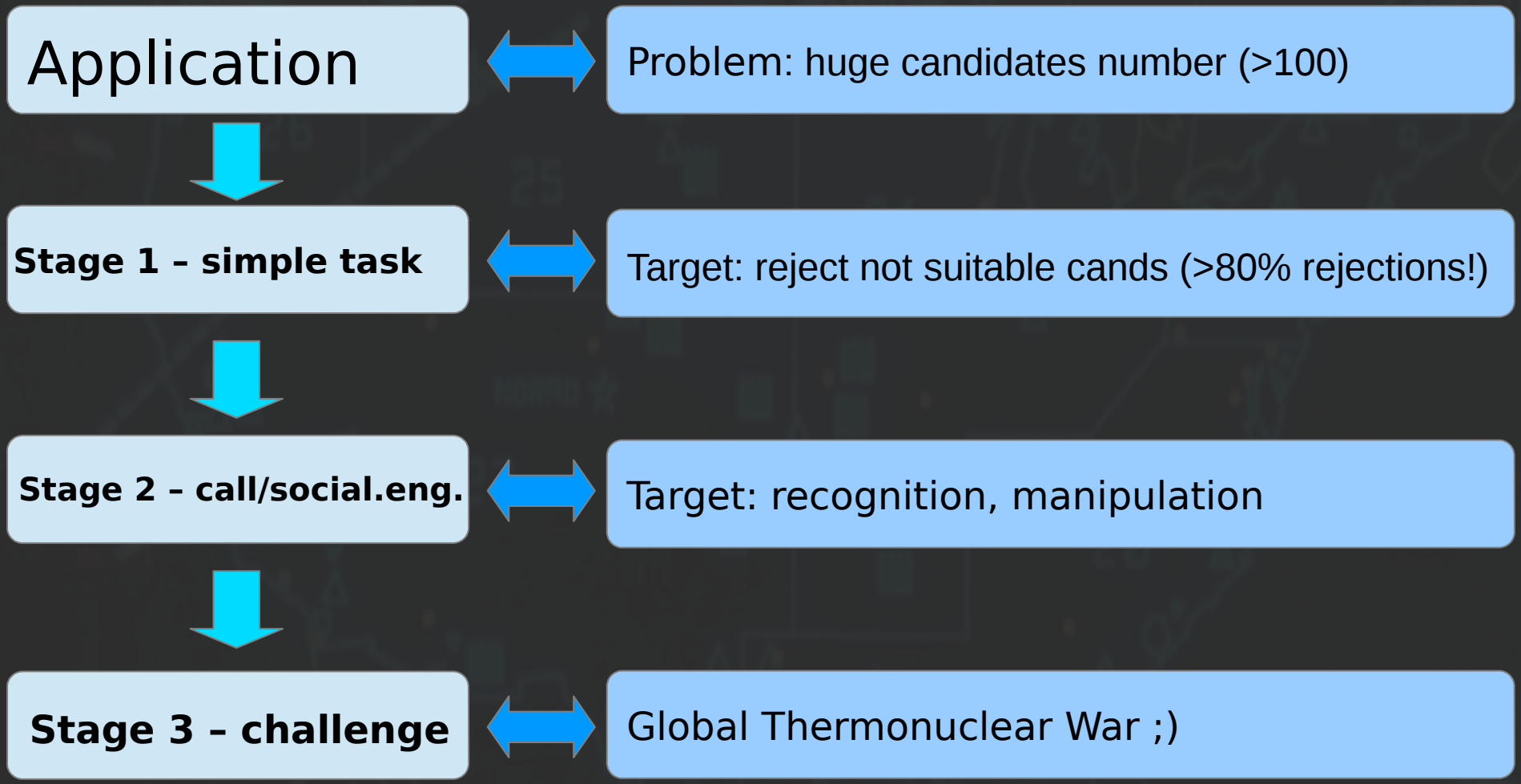
- He is sysop, developer, QA and network specialist
- Also great for performance tuning
- Responsible for critical data (all data)
- Easy handles moving UPSes between racks ;)

- Anytime day / night understands what you’re talking to him
- Everything he does respects high security standards
- Loves playing games (do you know sysop that doesn’t play)? ;)

Let’s play then

- Any idea? Not Quake / Diablo / Warcraft ;)
- pythonchallenge.com, wechall.net – CTFs are great!
- trueability.com – event for sysops
- So maybe CTF / challenge?
- Such system would have to fulfill some requirements:
  - Optimization of recruitment process time
  - Minimisation of the risk of rejecting good candidate
  - Draw attention as very interesting (you like mindfscks?)

Let's start the ball rolling



Stage 1 – telnet / SMTP

So - are you in? If so - please follow the white rabbit **@comegetsome.ganymede.eu** using **1130 TCP** port. And... say hello in the SMTP way to resolve this one

RFC-821/1869:
HELO/EHLO ??.....??

GPG us ur CV using
http://..../gpg.asc

Lack of GPG knowledge :(
RTFM!

Stage 1 – telnet / SMTP

So - are you in? If so - please follow the white rabbit **@comegetsome.ganymede.eu** using **1130 TCP** port. And... say hello in the SMTP way to resolve this one 

RFC-821/1869:
HELO/EHLO my.hostname

1 trap – not server's hostname but client's (90% catched)

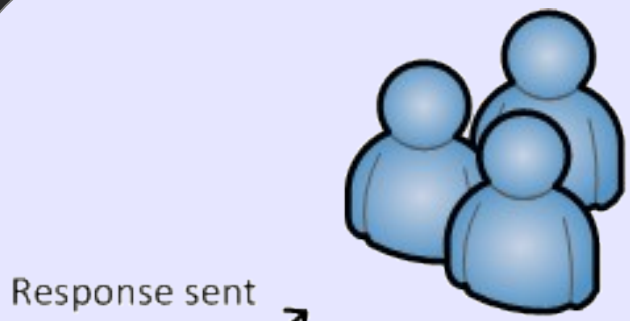
GPG us ur CV using http://..../gpg.asc

Lack of GPG knowledge :(RTFM!

Stage 1 – node.js

- At the beginning – pure C server. After 3am.. Node.js (simplicity) ;)
- What's wrong with node.js?
  - http://seclists.org/bugtraq/ - 0 hits
  - http://osvdb.org/ - 2 hits
  - http://1337day.com/, http://www.exploit-db.com/ - 1 hit
  - https://nodesecurity.io/advisories - 4 hits
- Does it mean that node.js is safe & secure?

Node.js – how it works?



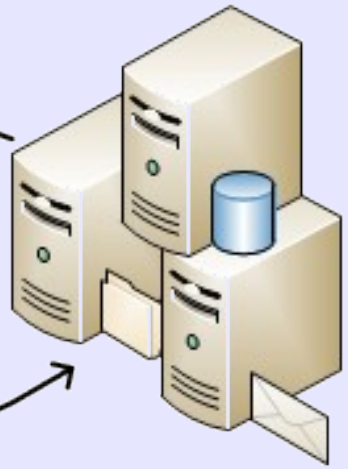
Clients connect to
Node's HTTP server

- Event driven
- Event loop
- Callbacks
- SPA, async, REST, Json

Response sent
To clients

Node's
Event Loop
Mono-threaded

Responses via
callback



Long operations are
deffered to workers
threads

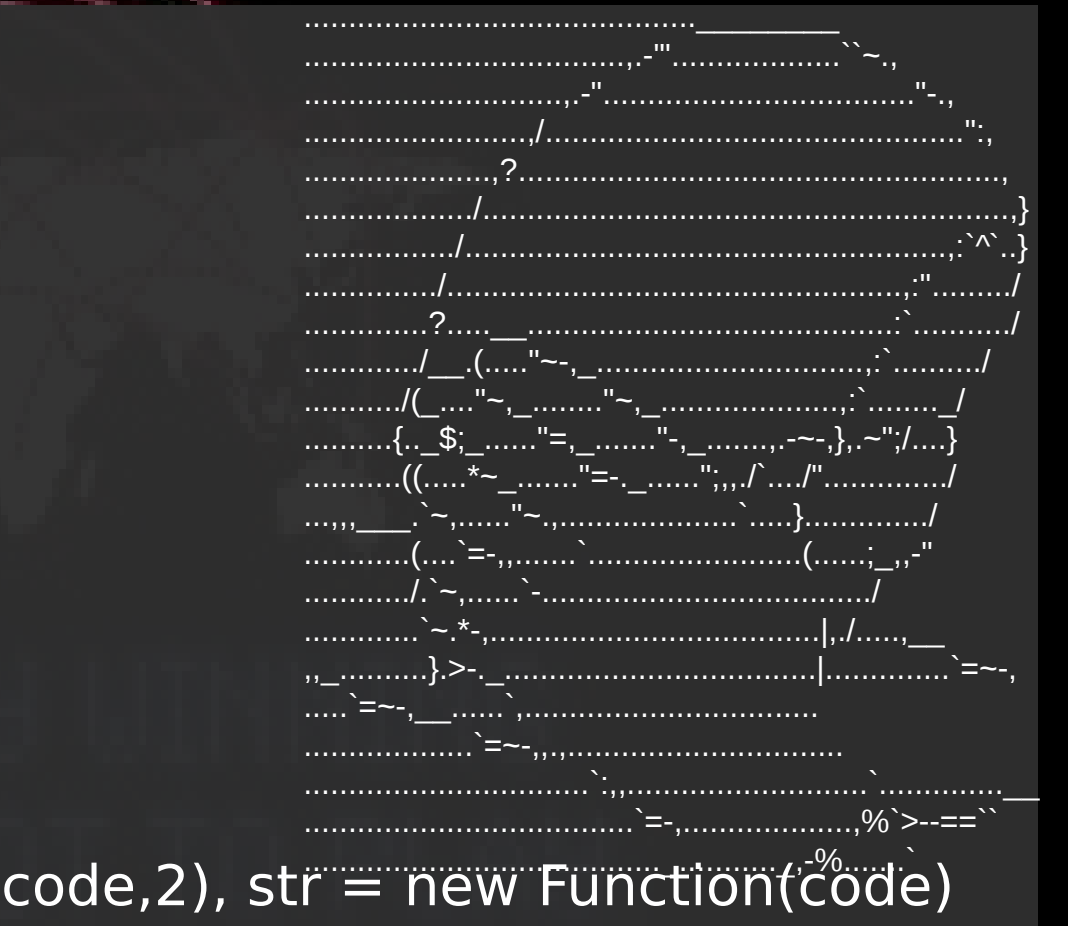
Threads are handled by
Node internally

Register callback

http://magnetik.github.io/

Node.js - threats

- no logging
- No error handling - DoS
- No configuration – “+” or “-”?
- No filters checking user-input
- JS: function as a variable
- Evil eval(code). Server-side XSS
- setInterval(code,2), setTimeout(code,2), str = new Function(code)
- Moduły npm – who creates those?



Node.js – evil eval()

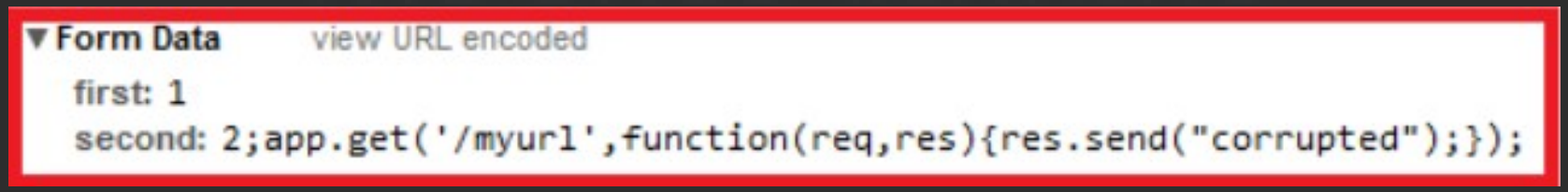
```
// Show the form to client
app.get("/sum",function(req,res){
    res.send("<form method='POST'>"+
    "<input name='first' /><input name='second' />"+
    "<input type='submit' value='submit' />");
});
// Process the form

app.post("/sum",function(req,res){
    var sum = eval(req.body.first +"+"+req.body.second);
    res.send("the answer is "+sum);
});
```

Node.js – evil eval()

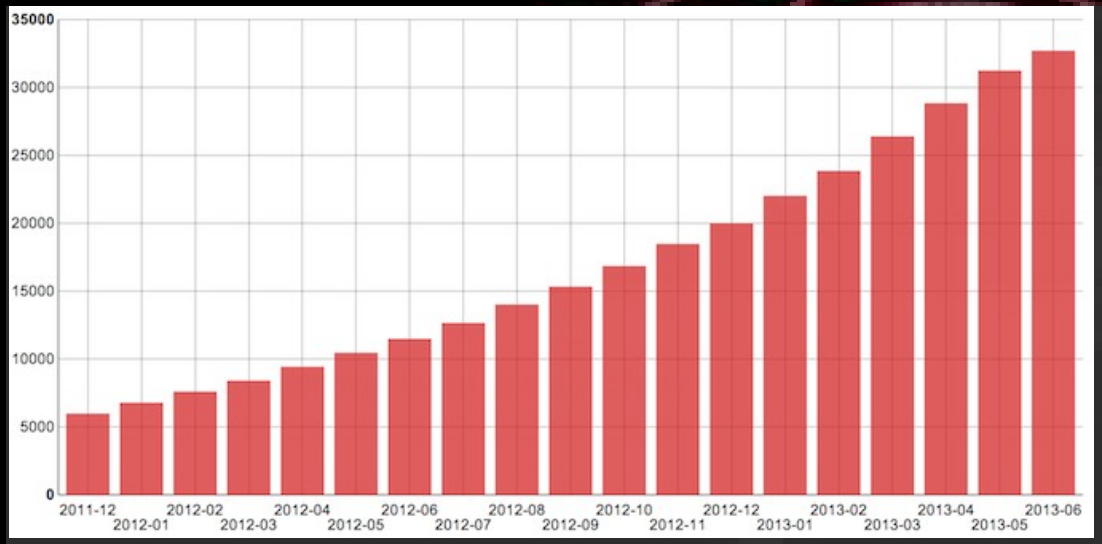
```
// Show the form to client
app.get("/sum",function(req,res){
    res.send("<form method='POST'>"+
    "<input name='first' /><input name='second' />"+
    "<input type='submit' value='submit' />");
});
// Process the form

app.post("/sum",function(req,res){
    var sum = eval(req.body.first +"+"+req.body.second);
    res.send("the answer is "+sum);
});
```



This way we added new functionality to the server during runtime!
http://node.js/myurl

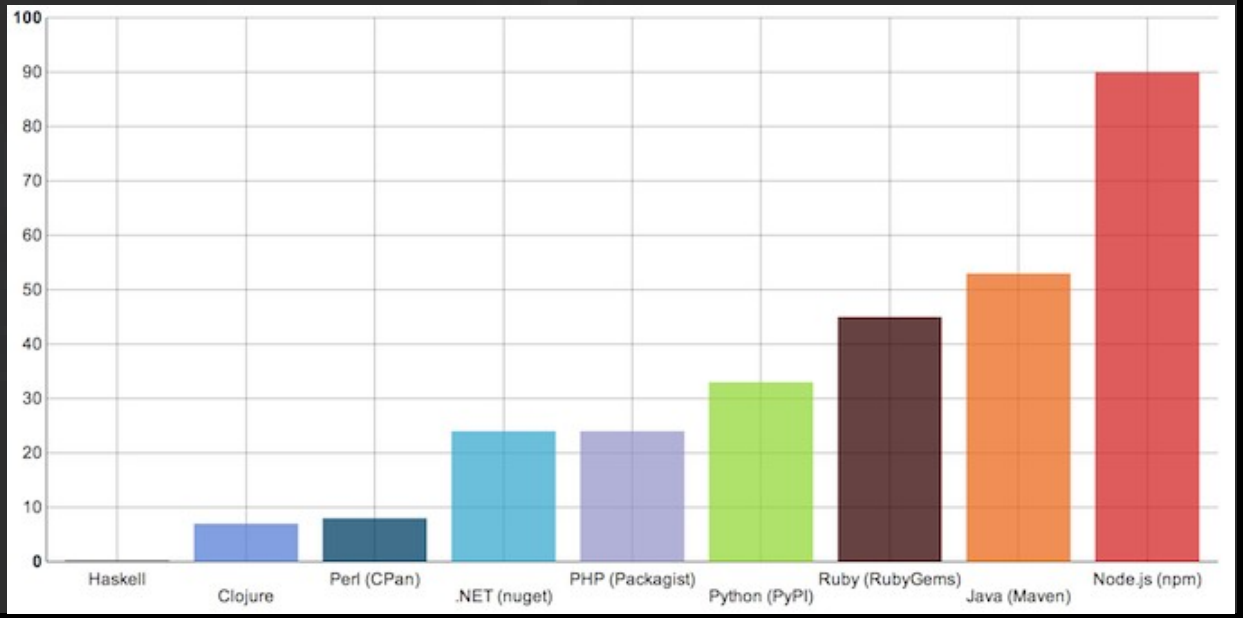
Node.js - npm



https://blog.nodejitsu.com/npm-innovation-through-modularity

Amount of npm modules in the time

Amount of npm-mods/day comparison to node.js and others



Node.js – how can?

- Use frameworks: https://npmjs.org/ - carefully
- Npm modules are not validated! Check those: https://nodesecurity.io
- Watch module dependencies!
- must have: your own error handling & logging
- This is server – we need proper server security solutions:
  - Monitoring – think how to monitor your app
  - Control-groups – set limits for resources
  - SELinux sandbox

Node.js – SELinux sandbox

- 'home_dir' and 'tmp_dir'
- App can r/w from std(in|out) + only defined FDs
- No network access
- No access to foreign processes / files
- We can easily connect sandbox with cgroups :
- Helpful: semodule -DB (no dontaudit)
- grep XXX /var/log/audit/audit.log | audit2allow -M node.sandbox
- semodule -i node.sandbox.pp

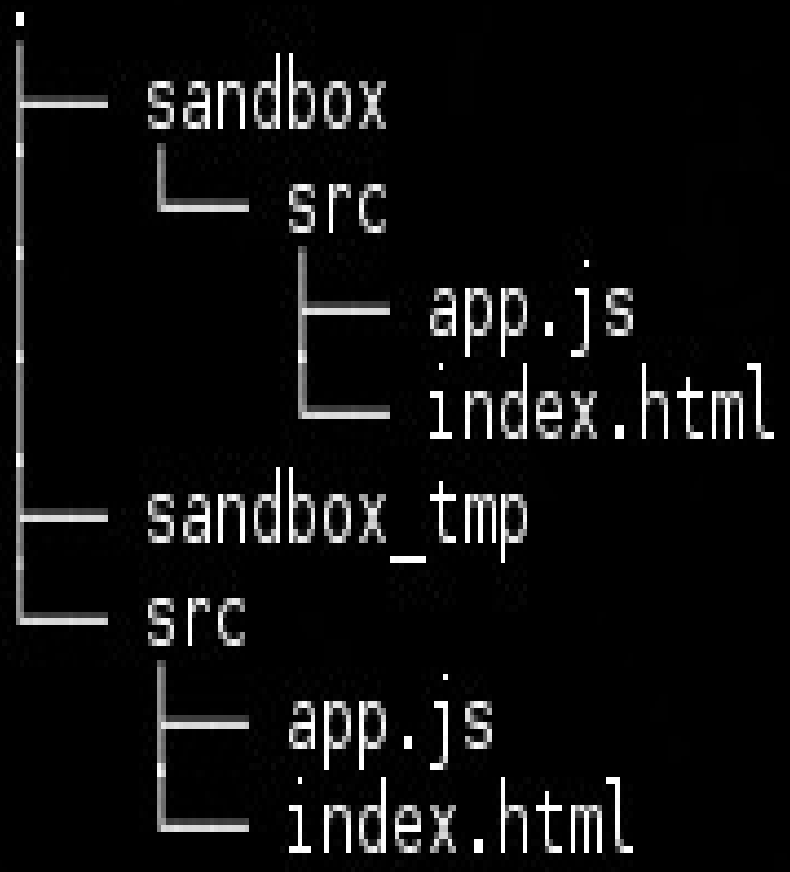
Node.js – SELinux sandbox

```
module node.js 1.0;

require {
    type user_devpts_t
    type anon_inodefs_t;
    type http_cache_port_t;
    type sandbox_t;
    type sandbox_net_t;
    class process execmem;
    class tcp_socket name_bind;
    class tcp_socket { name_bind listen };
    class chr_file { read append };
    class file write;
}

#============= sandbox_net_t ==============
allow sandbox_net_t anon_inodefs_t:file write;
allow sandbox_net_t http_cache_port_t:tcp_socket name_bind;
allow sandbox_net_t self:tcp_socket { accept listen };
allow sandbox_net_t self:tcp_socket listen;
allow sandbox_net_t user_devpts_t:chr_file { ioctl getattr };
allow sandbox_net_t user_devpts_t:chr_file { read append };

#============= sandbox_t ==============
allow sandbox_t self:process execmem;
```




```
sandbox -C -M -i src/index.html -H sandbox -T sandbox_tmp/ -t sandbox_net_t /usr/bin/node src/app.js
```

Node.js – how can #2

- Freeze node.js version per project?
- Let’s read & learn:
  - https://media.blackhat.com/bh-us-11/Sullivan/BH_US_11_Sullivan_Server_Side_WP.pdf
  - http://lab.cs.ttu.ee/dl91
  - https://github.com/toolness/security-adventure
- Pseudo–configuration – set limits in your code (e.g. POST size)
- try...catch ftw
- use strict; - helps even with eval case (partially)
- Bunyan / dtrace: https://npmjs.org/package/bunyan
- node.js OS? Oh and use / build node.js packages (fpm or whatever)

Stage 2 – social engineering

- Stage’s target is to verify & check candidate’s security awareness
- Christopher Hadnagy – SE framework (2k10):
  - http://www.social-engineer.org/framework/Social_Engineering_Framework
- Everyone can act as recruiter and call anyone
- Building network / connections on Linkedin is very easy
- Trust (lingo, easiness in some env: research)
- Sysop knows really much about env – he’s good target
- So one has to only get sysop’s trust and decrease his carefulness

Stage 3 - virtualization

- Our needs?
  - Boot process supervision
  - Console access
  - Resource management
  - Redundant storage
  - Rescue mode for VMs
  - Security by default

> AWS

> KVM/libvirt

> XEN/libvirt

> LXC

Stage 3 - virtualization

	boot	console	resources mgmt.	redundant storage	rescue VM	security
AWS	✗	✗	✓	✓	🔧	✓
KVM	✓	✓	✓	🔧	✓	✓
	✓	✓	✓	🔧	✓	✓
	✗	✓	🔧	🔧	✓	

Stage 3 - virtualization

vs

Performance XEN/HVM or KVM?

Stage 3 - virtualization

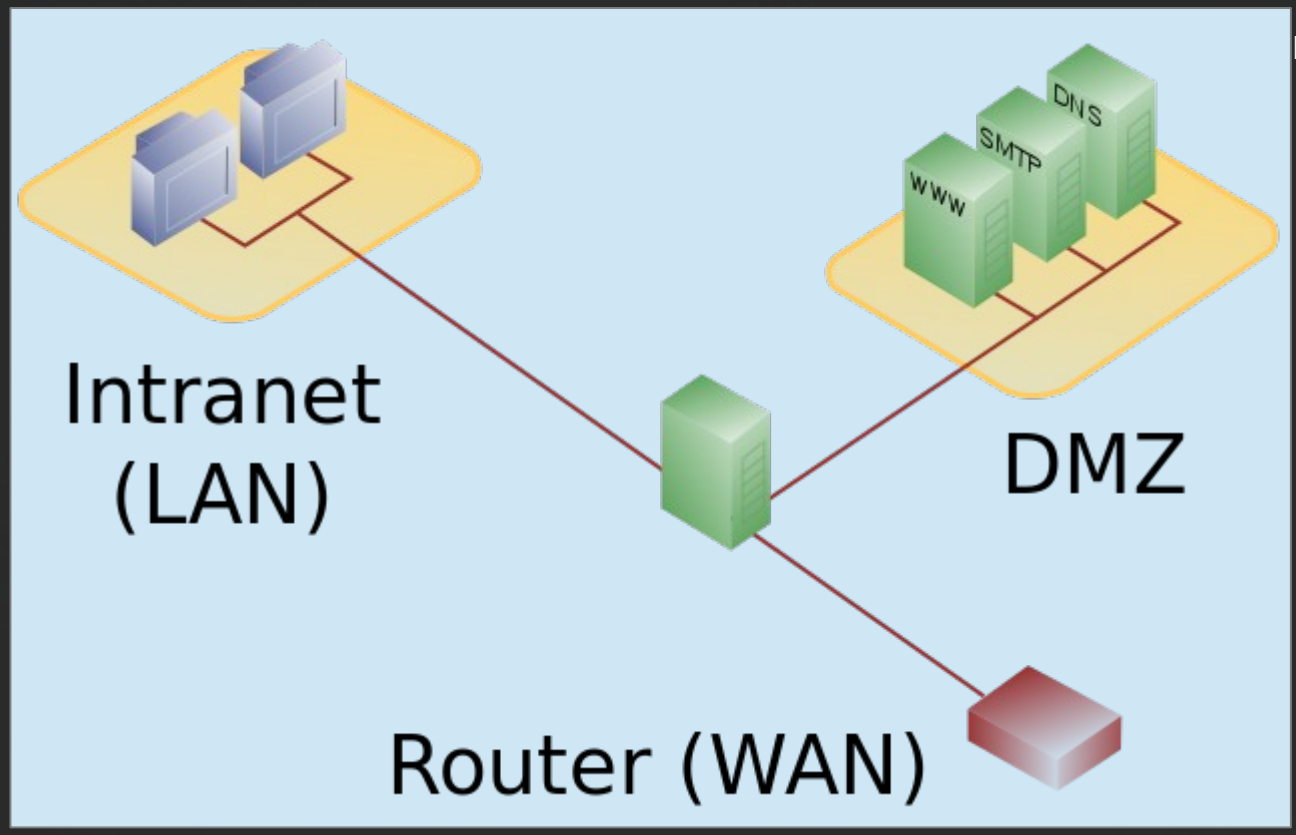
vs

Performance XEN/HVM or KVM?

We had great performance issues with XEN/HVM

The winner is „hat in the red” and its PV
(but with the cgroups help – under heavy load KVM
not that stable)

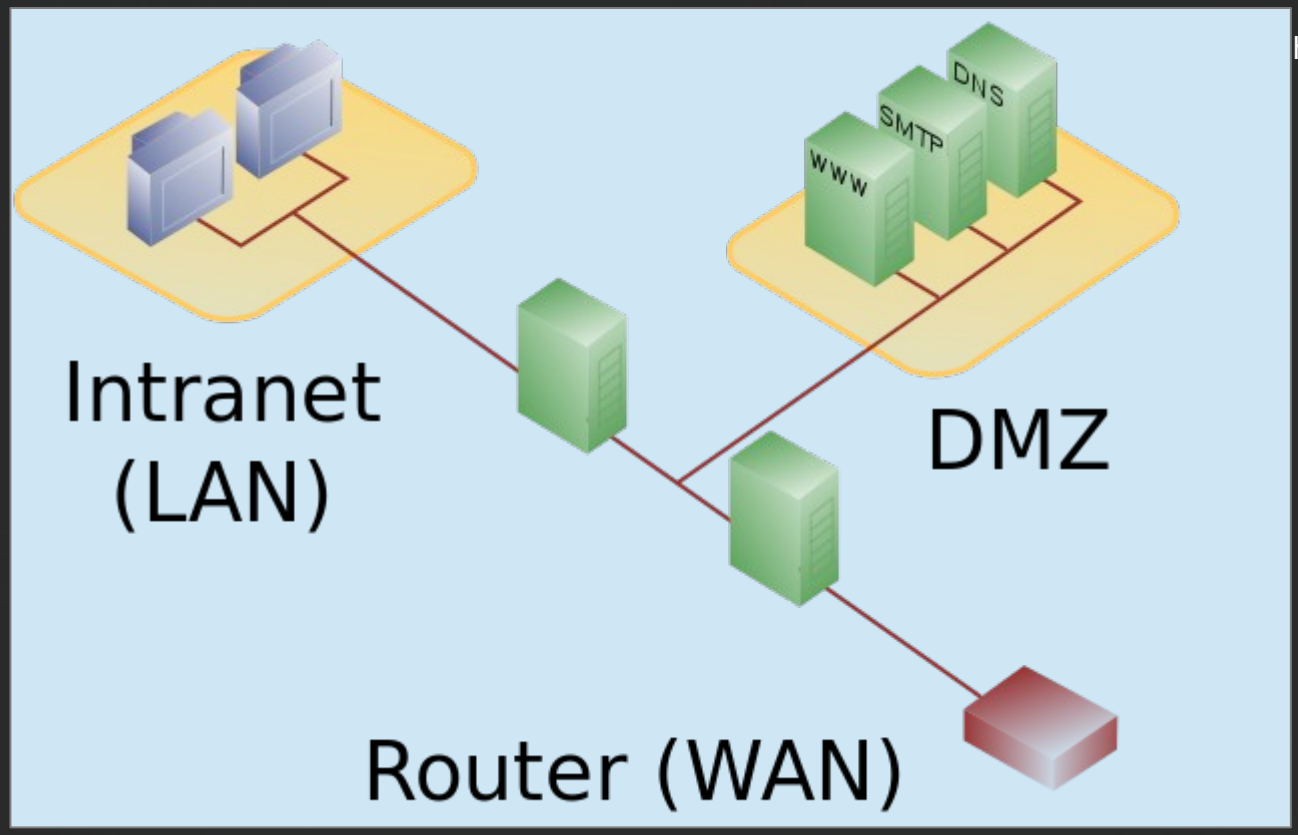
Stage 3 – network security



https://en.wikipedia.org

DMZ (Demilitarized Zone) – logical or physical partition

Stage 3 – network security



https://en.wikipedia.org

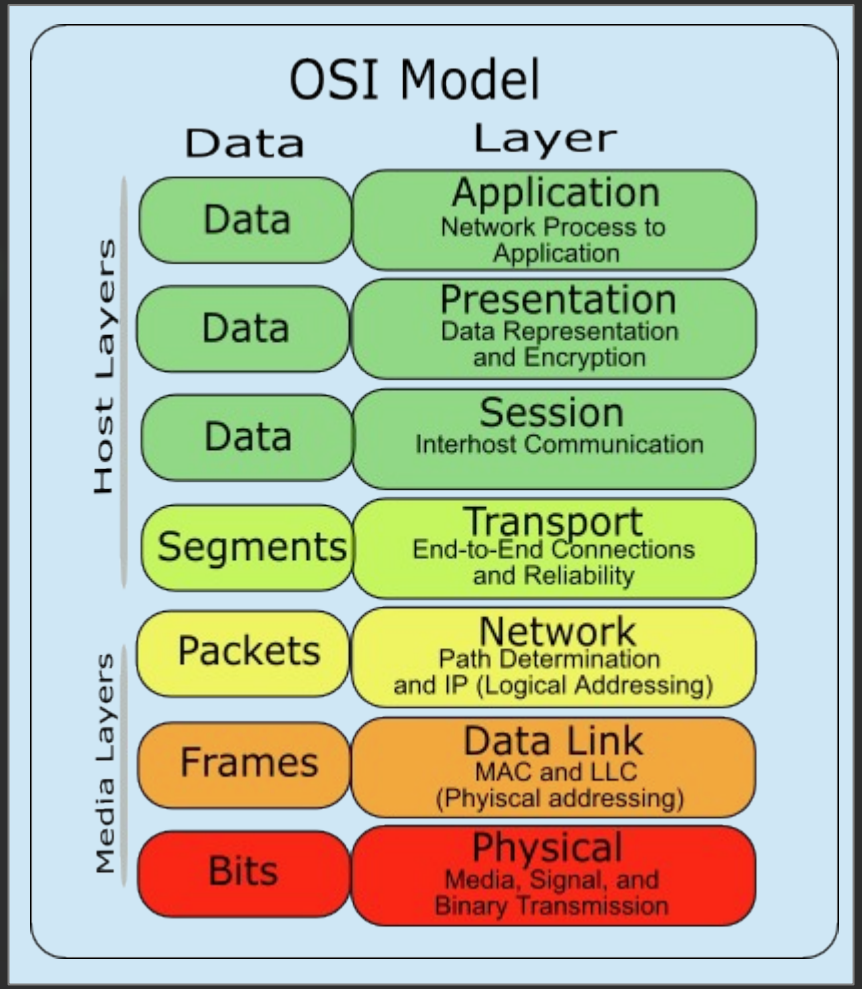
DMZ (Demilitarized Zone) – logical or physical partition

Stage 3 – network security

- Separated, dedicated DMZ (VLAN?) for host
- No routing / communication from this DMZ with other segments
- Low – cost solutions?
  - OpenWRT / DDWRT way || Pure Linux server
  - 802.1Q – VLANs

Stage 3 – network security

- Network isolation on KVM host:
  - Host/network bridge: L2 switch
  - netfilter / nwfilter (IBM)
  - By default there's no packets isolation in the bridged network - ebtables null, no filtering
  - ebtables – filtering l2– so we gain isolation
  - Or virsh nwfilter-list
    - allow-arp,dhcp,dhcp-server,clean-traffic, no-arp-ip-spoofing, no-arp-mac-spoofing, no-arp-spoofing, no-ip-multicast, no-ip-spoofing, no-mac-broadcast, no-mac-spoofing, no-other-l2-traffic
  - L2 filtering? /proc/sys/net/bridge



https://www.redhat.com/archives/libvir-list/2010-June/msg00762.html
http://pic.dhe.ibm.com/infocenter/lnxinfo/v3r0m0/topic/liaat/liaatsecurity_pdf.pdf

Stage 3 –boot process, VNC

- Accessing boot process – VNC
- VNC security? SSL? Complications..
- Maybe VNC over SSH tunnel?
  - Encryption
  - No certificates issues
  - Every admin can easily use VNC

Stage 3 – restricted shells

- SSH tunneling requires SSH access (thank You Captain Obvious!)
- SSH access is a threat per se
- Let’s limit this SSH / shell access – use restricted shells

Restricted shells by. Google ;) =>

Stage 3 – restricted shells

- Restricted shells are threat by default – unless we know how to use those!
- Under some circumstances one could escape the rshell:

```
~$ vi
:set shell=/bin/sh
:shell
```

```
~$ rbash
~$ cd /
rbash: cd: restricted
~$ bash
~$ cd /
/$
```

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

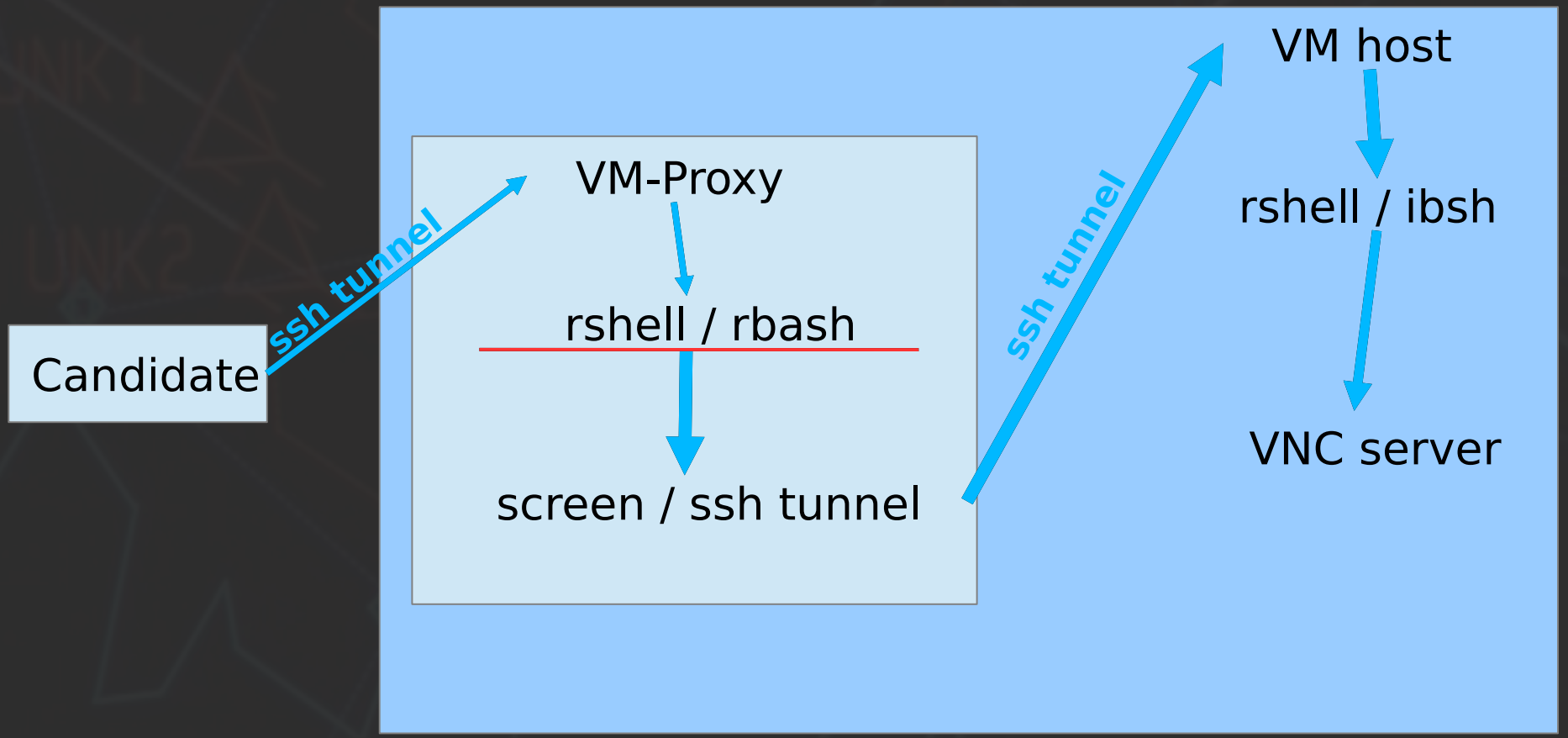
Stage 3 – restricted shells

- Rbash:
  - CentOSie / RHEL approved / friendly / legit ;)
  - Protects from directory traversal
  - Prohibits access to files via direct path
  - Prohibits setting PATH or other shell env variables
  - No commands output redirection
  - PATH=$HOME/bin – and reconsider 2x what to put into this „bin”

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

Stage 3 – SSH tunnel / VNC

- We must go deeper!

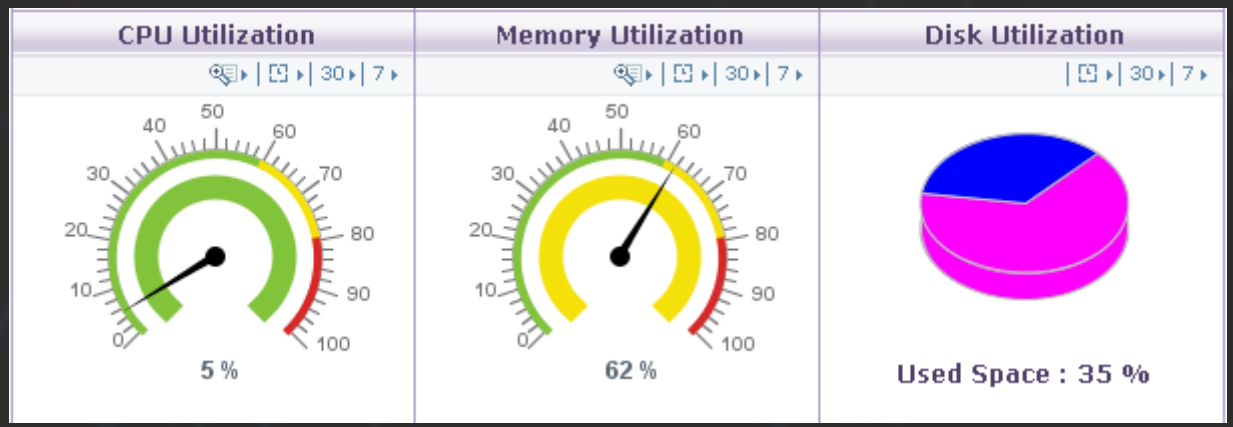


Stage 3 – restricted shells

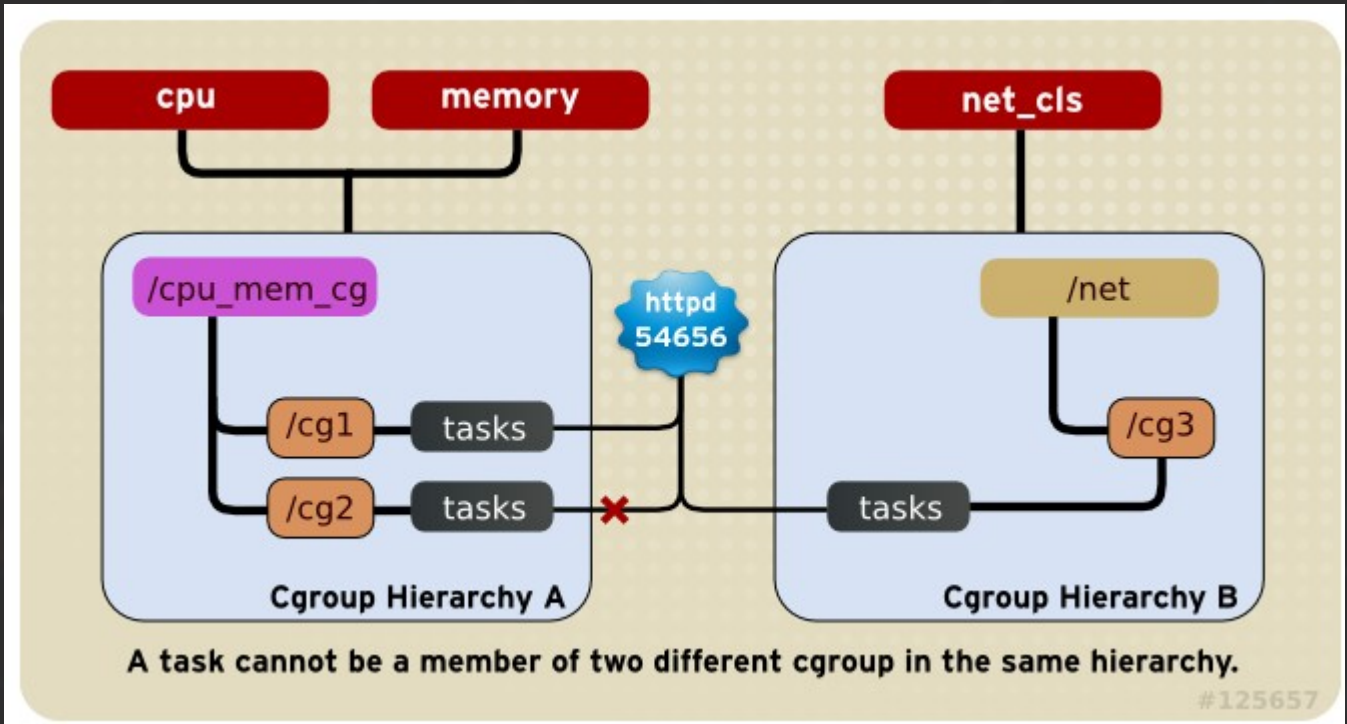
- Other restricted shells:
  - rssh – allows scp, sftp, rsync
  - sudosh - http://sourceforge.net/projects/sudosh
    - Allows saving whole user session and replay it
    - One can define allowed operations for user
    - Little outdated – better use sudosh3
  - Ibsh (small, fast, secure): http://sourceforge.net/projects/ibsh/

Stage 3 – control groups

- resource management in a simple way (ulimits, nice, limits.conf).. but..
- Could you set 50 IOPS for defined process?
- What about 100Kbp/s limit for particular user?
- issues with memory–leaks in Java?



Stage 3 – control groups

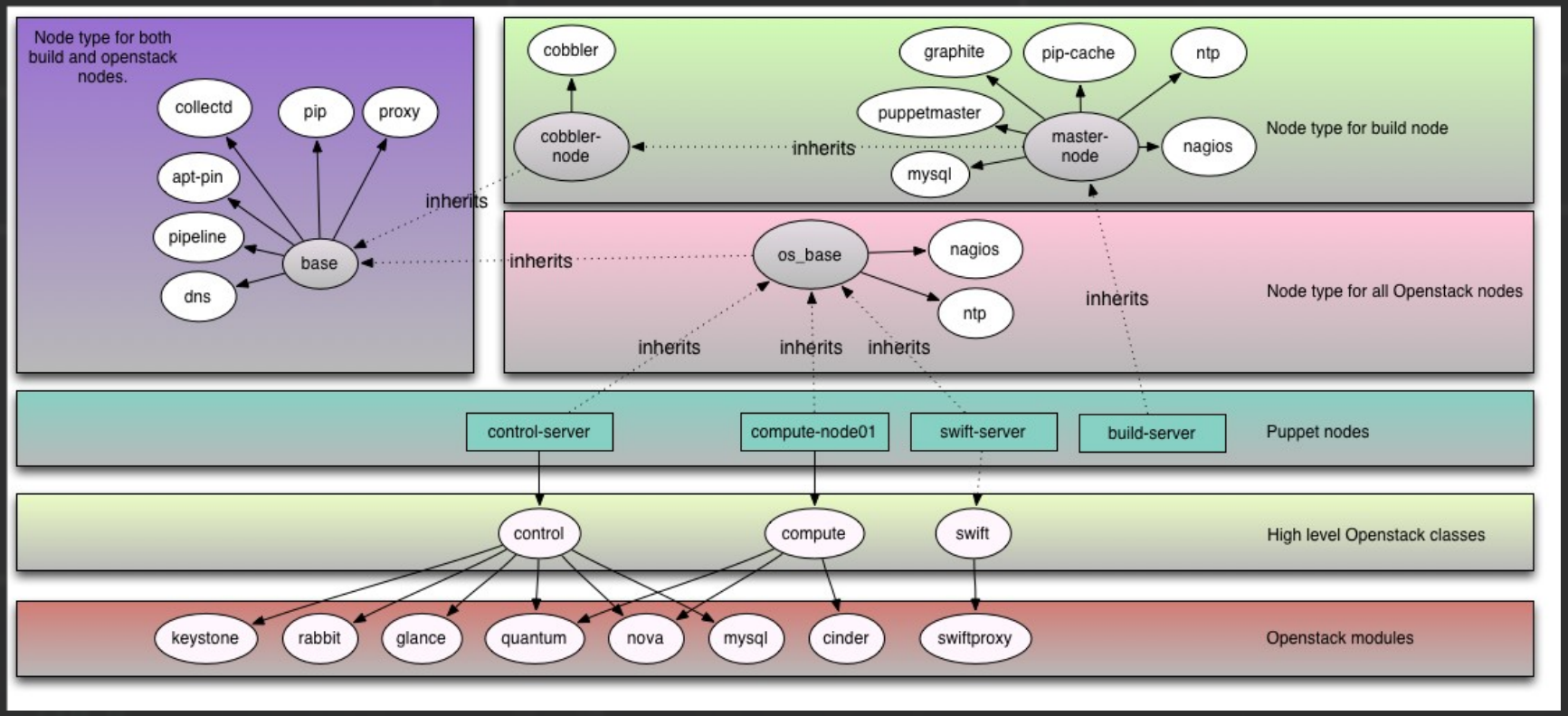


https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Resource_Management_Guide/ch01.html

- Debian & RHEL friendly
- Running apps in cgroup context
- Setting cgroup context for process during runtime

Stage 3 – web application

- OpenStack?



„Couple” of compliations ;) “Out of the box” – yup – I’ve heard about that ;) Could you deploy it in a few hours – securely?

Stage 3 – web application

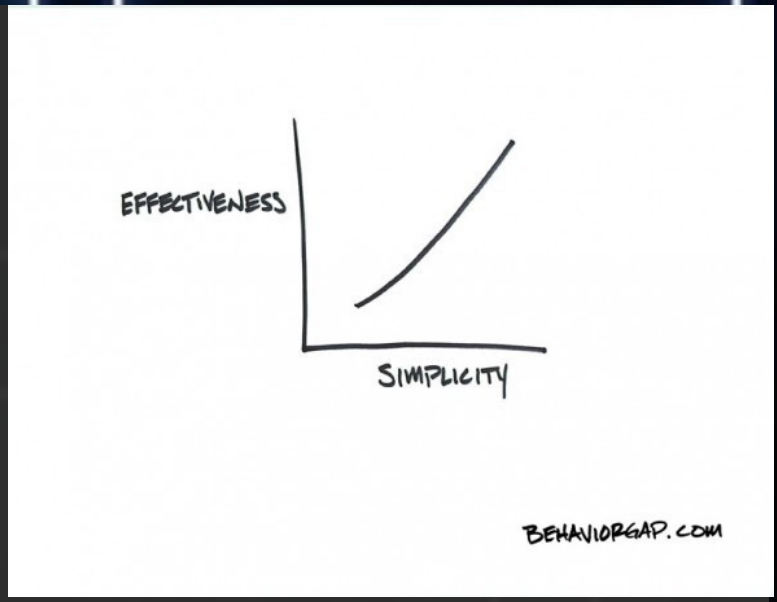
Commodore OS ???

Stage 3 – web application

Commodore OS Vision FTW!

Stage 3 – web application

- Apache + mod_security
- mod_security + OWASP rules
- PHP & Python :)
- Simplicity!
- VM management with simple daemon + screen:
  - while(1) do: manage_VMs();
- And this just works!

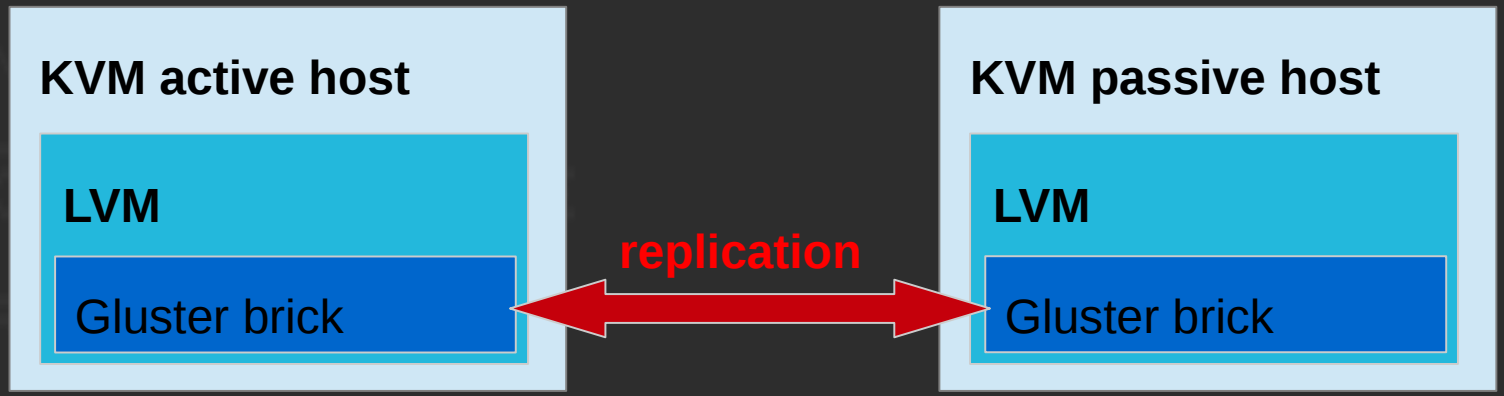


Stage 3 – recording SSH sessions

- We have to record all sessions – also those under „screen”
- Real time recording
- sudosh3 (sudosh fork) – kinda proxy shell – great ;)
- auditd – lov-lewel tool for recording syscalls
- Asciinema (ascii.io, Marcin Kulik) – great one, but not for audit purposes
- Ttyrec – outdated: http://0xcc.net/ttyrec/index.html.en
- Ssh logging patch - outdated: http://www.kdvelectronics.eu/ssh-logging/ssh-logging.html

Stage 3 – data security

- What if we loose any of the VMs...? Brrr....
- Risk assesement – what would be enough for us?
  - RAID1 / Mirror – “usually” is enough for a 3 – month time
  - Backups – useful ;) RAID / replication are not backups...
  - GlusterFS / DRBD – if you have enough resources – try it :)




Pytania?

Maciej Lasyk
http://maciek.lasyk.info
maciek@lasyk.info
Twitter: @docent_net

Ganymede

THE ONLY WINNING MOVE IS NOT TO PLAY

OWASP Poland, 2013-10-17