Maciej Lasyk

Ganymede

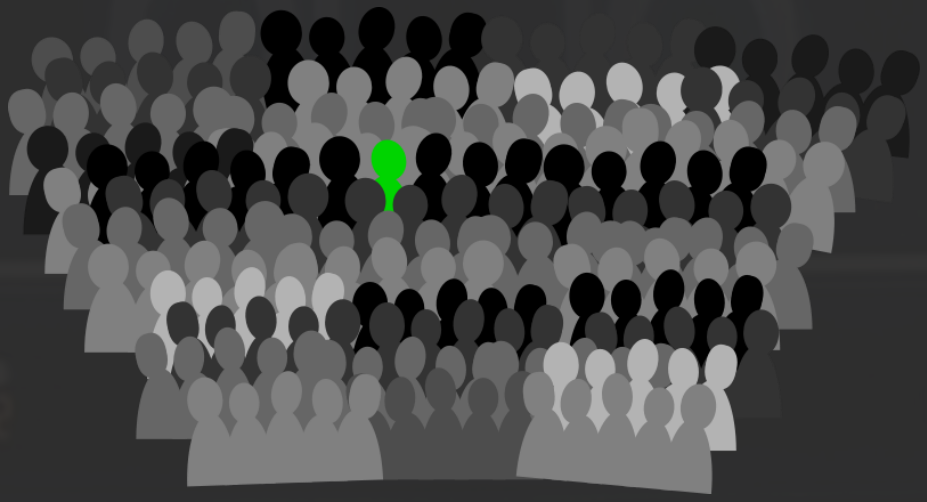
OWASP Poland, 2013-10-17

Czemu rekrutacja na OWASPie?

- Ponieważ ten system to webaplikacja
- Ponieważ bazuje w 100% na FOSS (open-source)
- Ponieważ security było dla nas bardzo istotne
- Bo na OWASPie spotykają się ludzie, którzy mają wpływ na procesy rekrutacji w innych firmach (hopefully) ;)

Rekrutacja

- Sporo agencji / serwisów świadczących usługi rekrutacyjne
- Potencjalnie ogromna ilość kandydatów
- Procesowaniem kandydatów zajmuje się cały zespół
- Rozmowy i testy pochłaniają wiele czasu



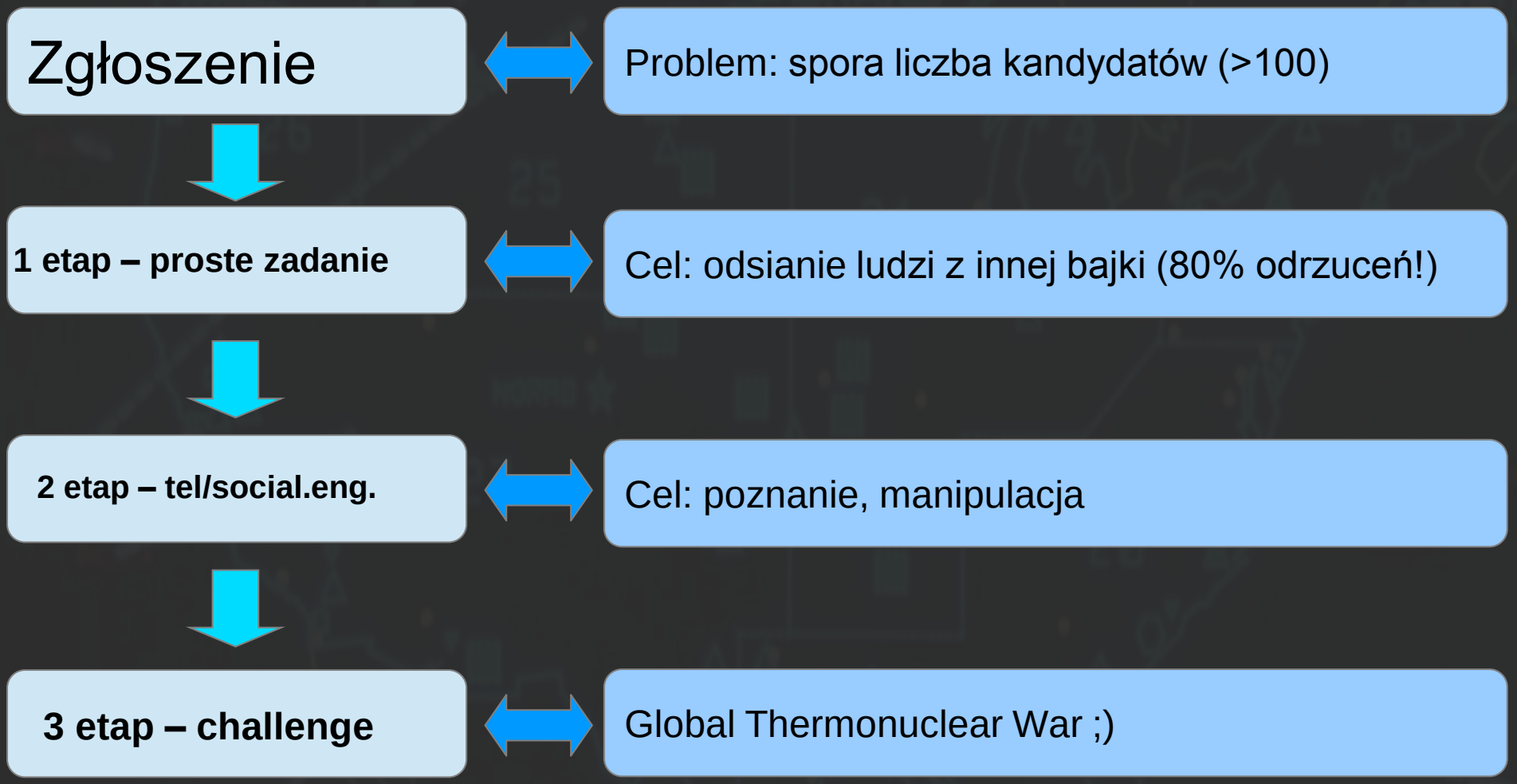
SysAdmin / Operations

- Jest administratorem, programistą, testerem i sieciowcem
- Performance tuning również nie jest mu obcy
- Odpowiada za krytyczne (wszystkie) dane
- Potrafi przenosić UPSy ;)
- O 4 nad ranem w niedzielę rozumie kolegów z innych krajów ;)
- Wszystko to robi w kontekście wysokiego bezpieczeństwa
- Lubi grać (znacie admina, który nie gra?) ;)

Zagrajmy więc

- Pomysł na grę? Nie Quake / Diablo / Warcraft ;)
- pythonchallenge.com, wechall.net – CTFy są świetną formą
- trueability.com – taki event dla adminów
- Może więc coś w rodzaju CTFa / challenge?
- Taki system musiałby spełniać kilka wymogów:
  - Optymalizacja czasu rekrutacji
  - Zminimalizować ryzyko odrzucenia dobrego kandydata
  - Być na tyle ciekawy aby przyciągał (kto nie lubi mindfscków)

Let's start the ball rolling



1 etap – telnetem przez 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

Sporo nieznajomości GPG :(
RTFM!

1 etap – telnetem przez 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 haczyk – hostname nie serwera a klienta (90% się złapała)

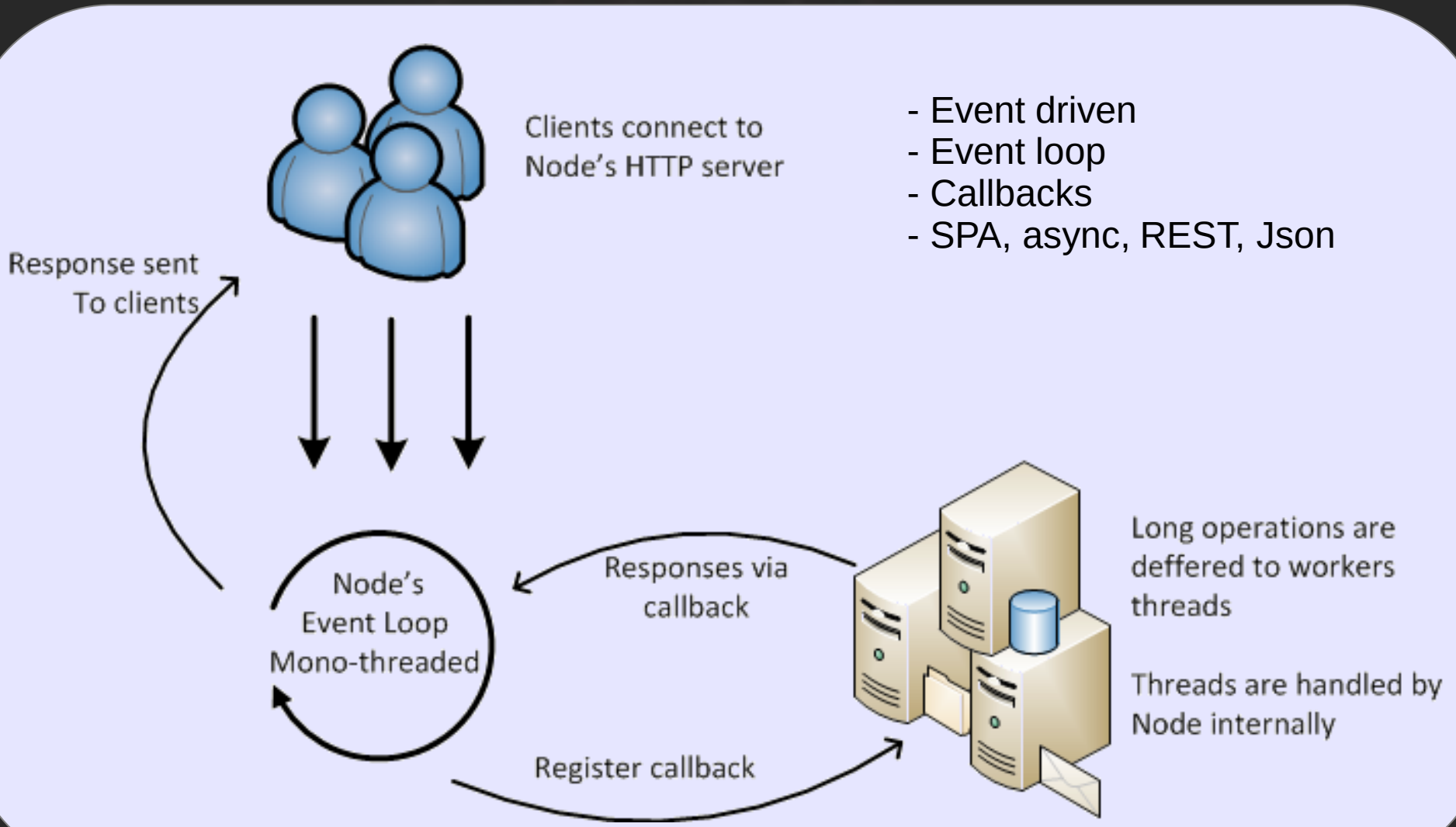
GPG us ur CV using http://..../gpg.asc

Sporo nieznajomości GPG :(RTFM!

1 etap – node.js

- Początkowo serwer w C. Po 3 w nocy jednak node.js ;)
- Co jest nie tak z node.js?
  - http://seclists.org/bugtraq/ - 0 trafień
  - http://osvdb.org/ - 2 trafienia
  - http://1337day.com/, http://www.exploit-db.com/ - 1 trafienie
  - https://nodesecurity.io/advisories - 4 trafienia
- Czy to oznacza, że node.js jest “bezpieczny”?

Node.js – model działania

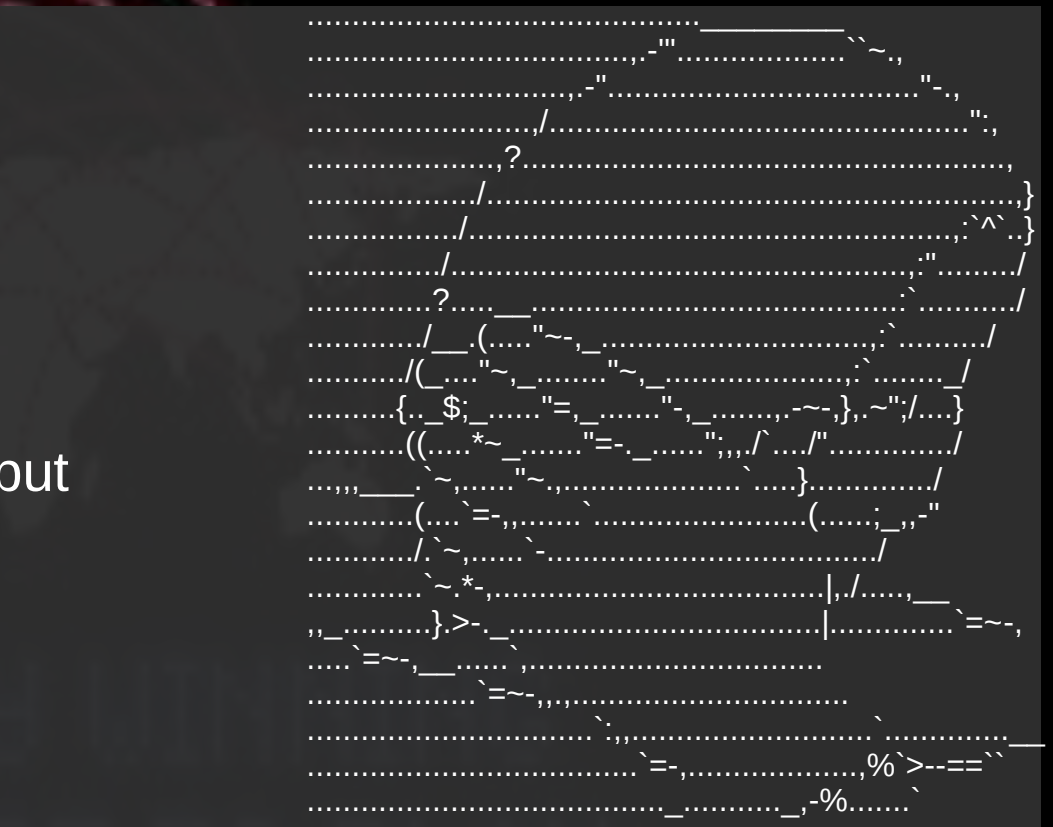


- Event driven
- Event loop
- Callbacks
- SPA, async, REST, Json

http://magnetik.github.io/

Node.js - zagrożenia

- Brak logów
- Brak obsługi błędów - DoS
- Brak konfiguracji – “+” czy “-”?
- Brak filtrów sprawdzających 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 – kto i jak je tworzy?



Node.js – evil eval()

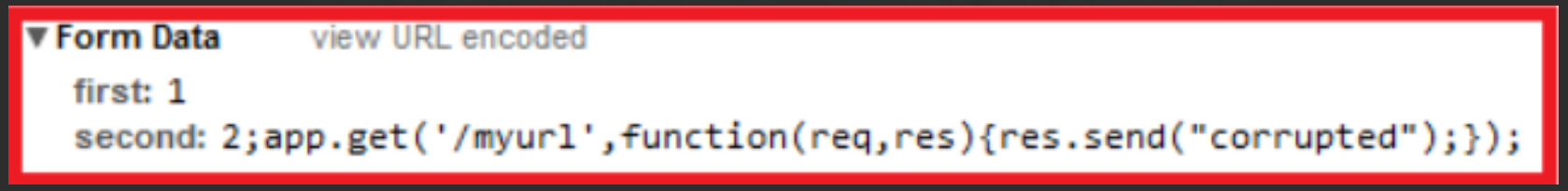
```
// Show the form to client
app.get("/sum",function(req,res){
    res.send("<form method='POST'>"+
    "<input name='first' /><input name='second' />"+
    "<input type='submit' value='submit' />");
});
// Process the form

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()

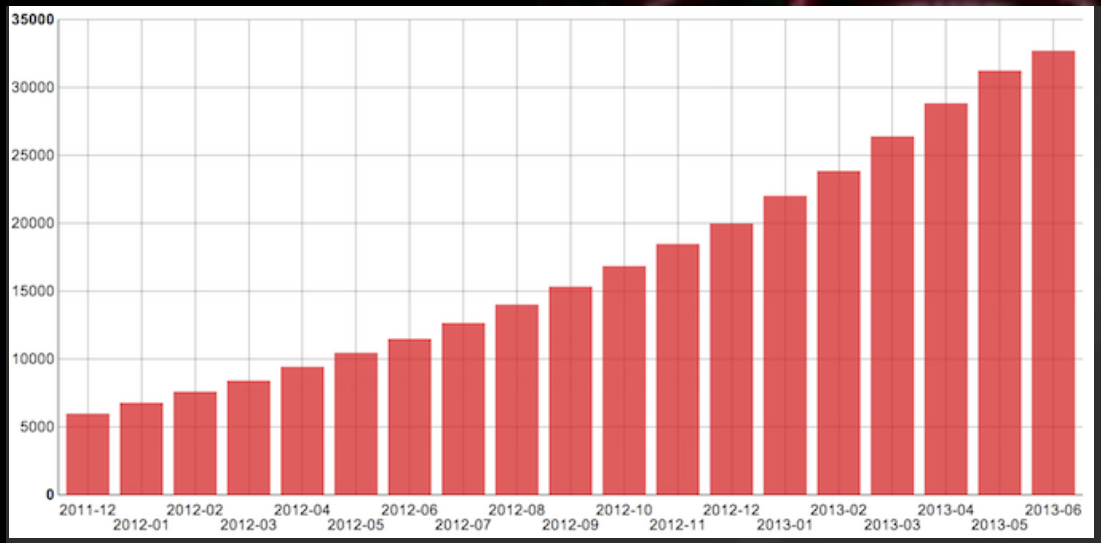
```
// Show the form to client
app.get("/sum",function(req,res){
    res.send("<form method='POST'>"+
    "<input name='first' /><input name='second' />"+
    "<input type='submit' value='submit' />");
});
// Process the form

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



W ten oto sposób rozszerzyliśmy funkcjonalność o http://node.js/myurl

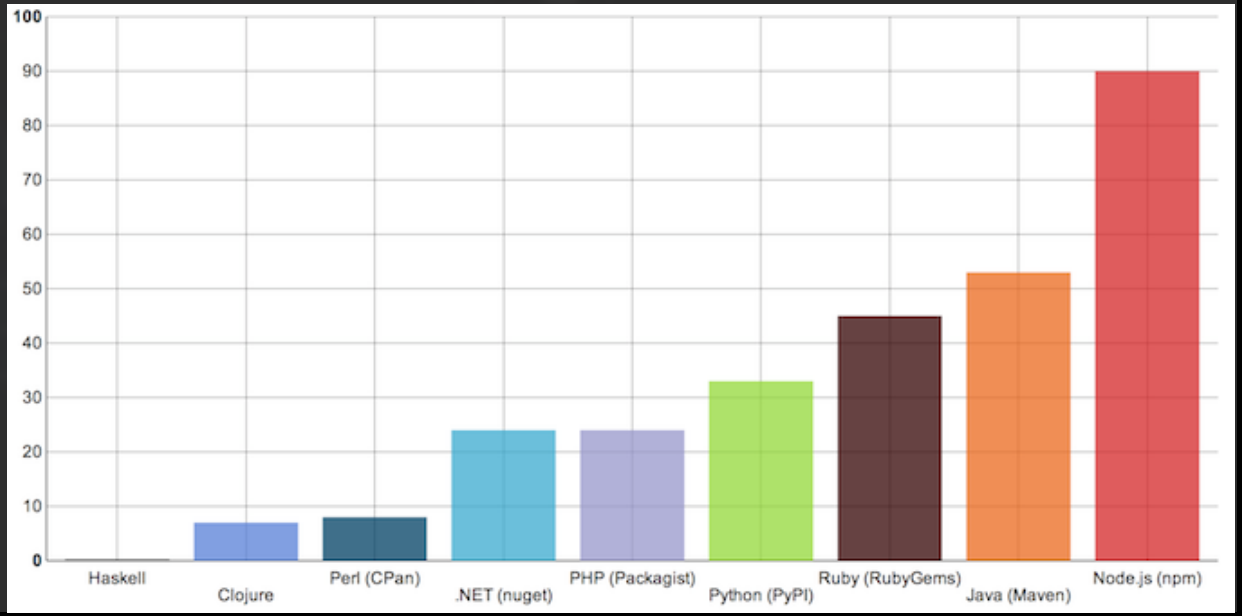
Node.js - npm



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

Przyrost ilości modułów npm

Porównanie ilości modułów/day Node.js i innych technologii



Node.js – jak żyć?

- Używajmy frameworków: https://npmjs.org/ - uważnie
- Moduły npm nie są validowane! Sprawdzajmy je: https://nodesecurity.io
- Uważajmy na zależności między modułami
- Własna obsługa logów i błędów – takie “must have”
- Skoro to jest serwer to potrzeba nam rozwiązań sec-server-side:
  - Monitoring – twórzmy aplikacje myśląc o tym jak je monitorować
  - Control-groups – ustalmy limity zasobów! (o tym zaraz...)
  - SELinux sandbox (o tym zaraz)

Node.js – SELinux sandbox

- Ustawiamy aplikacji 'home_dir' i 'tmp_dir'
- Aplikacja może domyślnie r/w z std(in|out) + defined FDs
- Domyślnie – brak dostępu do sieci
- Brak dostępu do nie swoich procesów / plików
- Sandboxa możemy spiąć z cgroupsami :)
- Pomocne: semodule -DB (no dontaudit)
- grep XXX /var/log/audit/audit.log | audit2allow -M node.sandbox
- semodule -i node.sandbox.pp

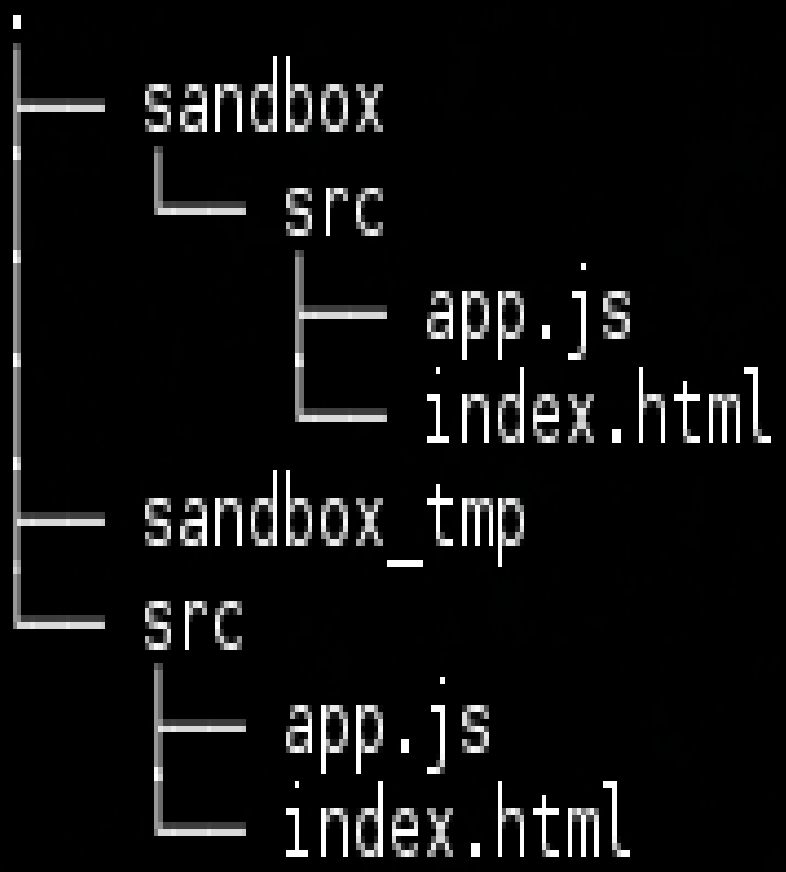
Node.js – SELinux sandbox

```
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 – jak żyć #2

- Freeze node.js version per project?
- Czytamy:
  - 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–konfiguracja – ustalajmy limity w kodzie (np. POST size)
- try...catch ftw
- use strict; - pomaga nawet w sprawie evala (częściowo)
- Bunyan / dtrace: https://npmjs.org/package/bunyan
- A node.js OS? No i instalujmy node.js z paczek (fpm choćby)

Etap 2 – social engineering

- Celem tego etapu jest weryfikacja i poznanie kandydata
- Christopher Hadnagy – SE framework (2k10):
  - http://www.social-engineer.org/framework/Social_Engineering_Framework
- Każdy może się podać za rekrutera i zadzwonić do kandydata
- Budowanie profilu/znajomości na Linkedinie jest bardzo proste
- Zaufanie (lingo, znajomość tematu / środowiska: research)
- Administrator jest osobą ze sporą wiedzą – jest dobrym celem
- Wystarczy tylko zdobyć zaufanie i obniżyć poziom ostrożności

Etap 3 - wirtualizacja

- Co nam potrzeba?
  - Nadzór nad procesem bootowania
  - Możliwość wpięcia się w konsolę
  - Sterowanie zasobami
  - Redundantny storage
  - Tryb rescue dla obrazu VM
  - Security by default

> AWS

> KVM/libvirt

> XEN/libvirt

> LXC

Etap 3 - wirtualizacja

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

Etap 3 - wirtualizacja

vs

Wydajność XEN/HVM czy KVM?

Etap 3 - wirtualizacja

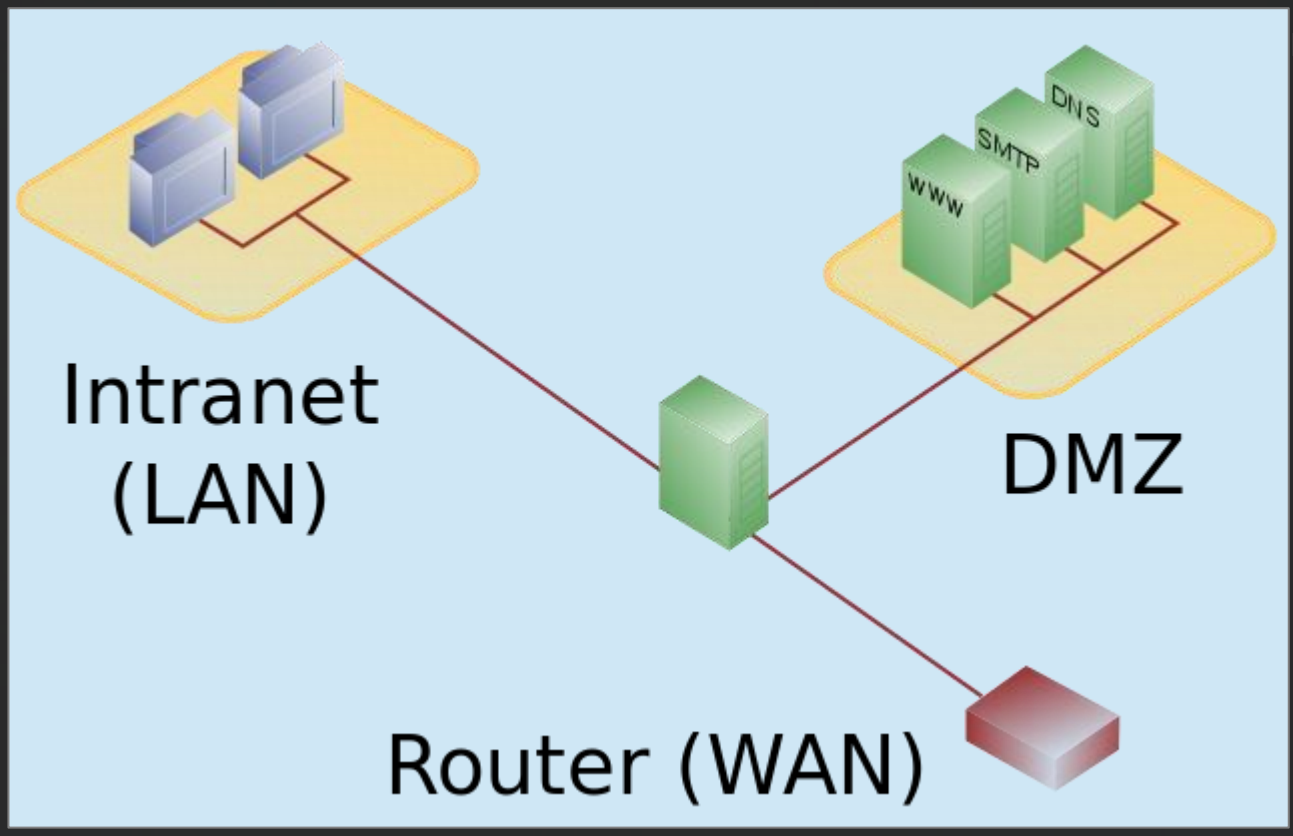
vs

Wydajność XEN/HVM czy KVM?

Z XEN/HVM mieliśmy problemy wydajnościowe – spore.

Tutaj wygrywa czerwony kapelusz i jego PV
(jednak z pomocą cgroups – KVM przy
sporym obciążeniu zachowuje się niestabilnie)

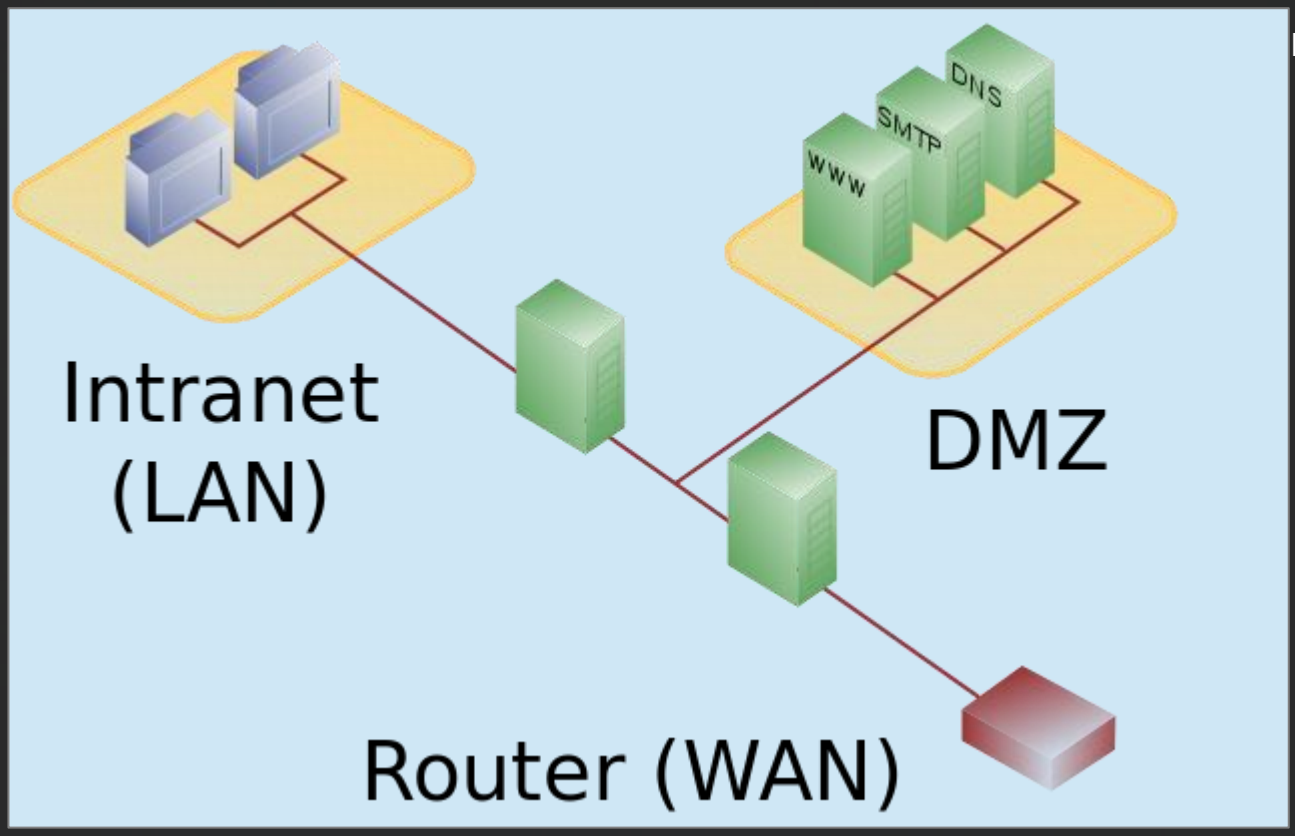
Etap 3 – bezpieczeństwo sieci



https://en.wikipedia.org

DMZ (Demilitarized Zone) – wydzielona część sieci zawierająca usługi, które są wystawione bezpośrednio na świat i są też najbardziej zagrożone atakami. Podział logiczny (VLANy) lub fizyczny

Etap 3 – bezpieczeństwo sieci



https://en.wikipedia.org

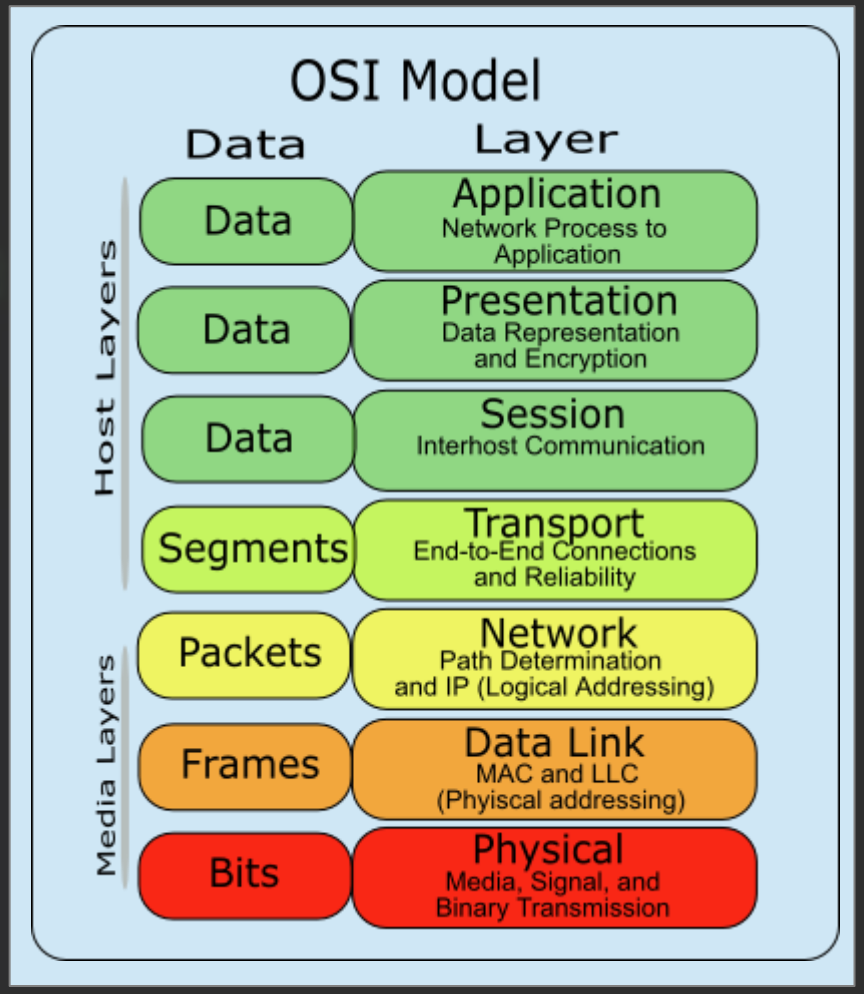
DMZ (Demilitarized Zone) – wydzielona część sieci zawierająca usługi, które są wystawione bezpośrednio na świat i są też najbardziej zagrożone atakami. Podział logiczny (VLANy) lub fizyczny

Etap 3 – bezpieczeństwo sieci

- Wydzielony DMZ (VLAN?) dla hosta
- Brak routingu / komunikacji tej DMZ z resztą
- Tanie rozwiązania?
  - OpenWRT / DDWRT way || Pure Linux server
  - 802.1Q – VLANy

Etap 3 – bezpieczeństwo sieci

- Izolacja sieci na hoście KVM:
  - Host/network bridge: switch warstwy 2
  - netfilter vs nefilter (IBM)
  - domyślnie nie mamy izolacji pakietów dla VM w bridge'u – ebtables null, no filtering
  - ebtables – filtrowanie warstwy 2 – w ten sposób załatwiamy izolację
  - Albo 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

Etap 3 – proces bootowania, VNC

- Dostęp do procesu bootowania – VNC
- Bezpieczeństwo VNC? SSL? Komplikacje..
- A gdyby tak VNC over SSH tunnel?
  - Jest szyfrowanie
  - Żadnej zabawy z certami
  - Każdy z nas to już robił...

Etap 3 – restricted shells

- Tunel SSH wymaga konta SSH (thank You Captain Obvious!)
- A za pomocą konta shellowego można sporo nabroić...
- Ograniczmy więc możliwe akcje użytkownika – restricted shells

Restricted shells wg. Google ;) =>

Etap 3 – restricted shells

- Restricted shells są groźne – trzeba rozumieć jak działają!
- W sprzyjających warunkach można uciec z rshella:

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

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

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

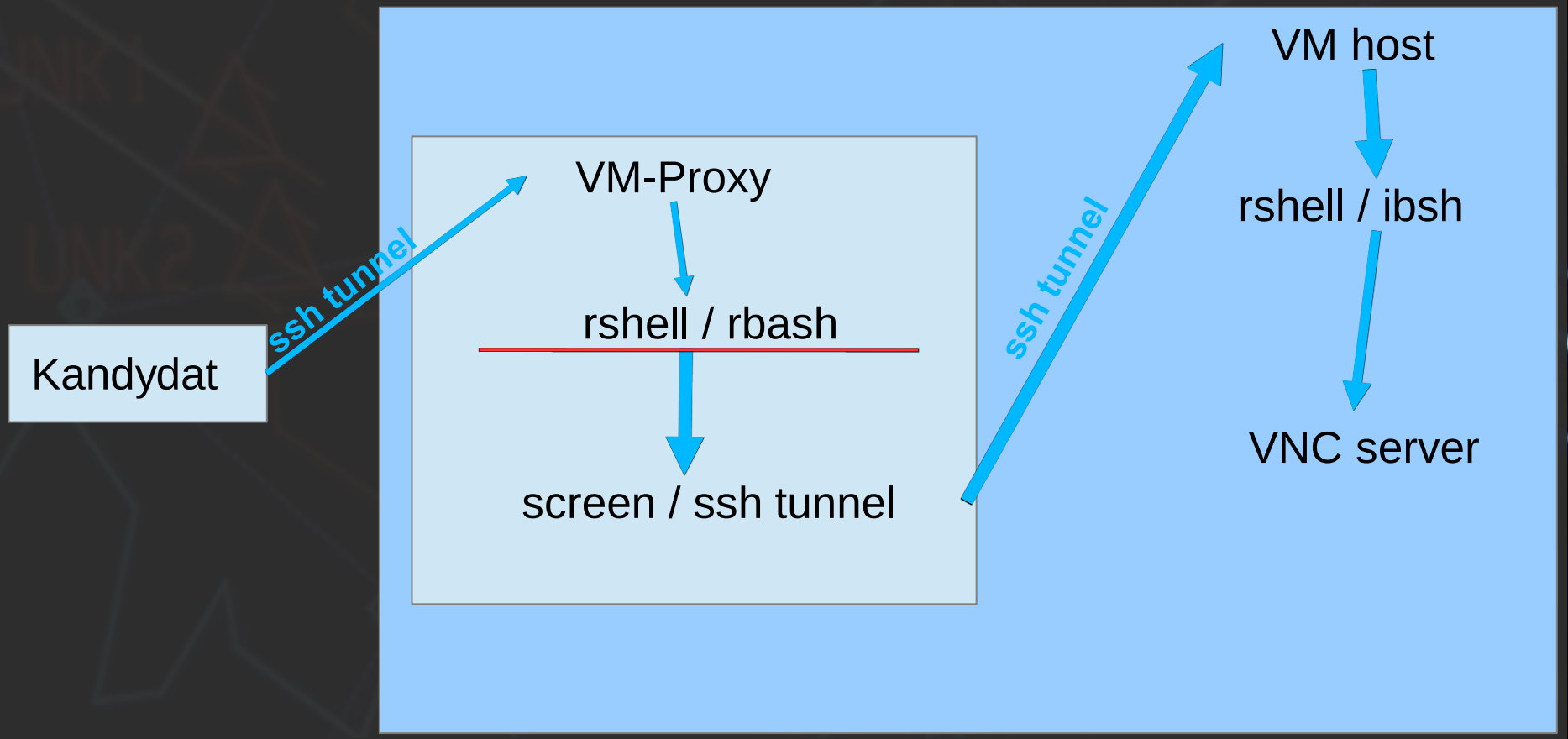
Etap 3 – restricted shells

- Rbash:
  - Jest w CentOSie / RHEL jako legit ;)
  - Zabrania trawersować katalogi
  - Zabrania używac bezpośrednich ścieżek do plików / katalogów (takich z '/')
  - Zabrania ustawiać PATH czy inną zmienną powłoki
  - Zabrania przekierowywać output komend
  - PATH=$HOME/bin – i niech tam będzie prawie pusto!

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

Etap 3 – SSH tunnel / VNC

- We must go deeper!

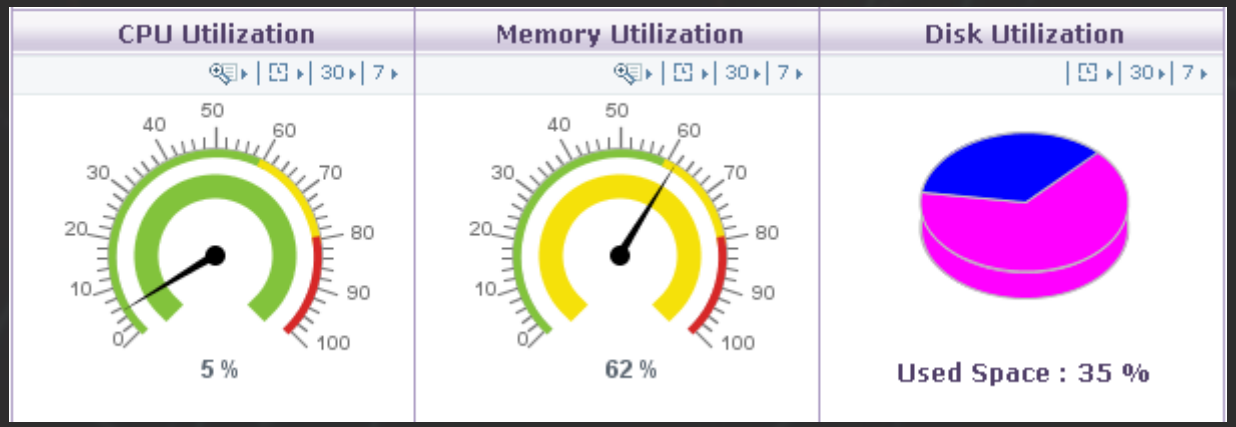


Etap 3 – restricted shells

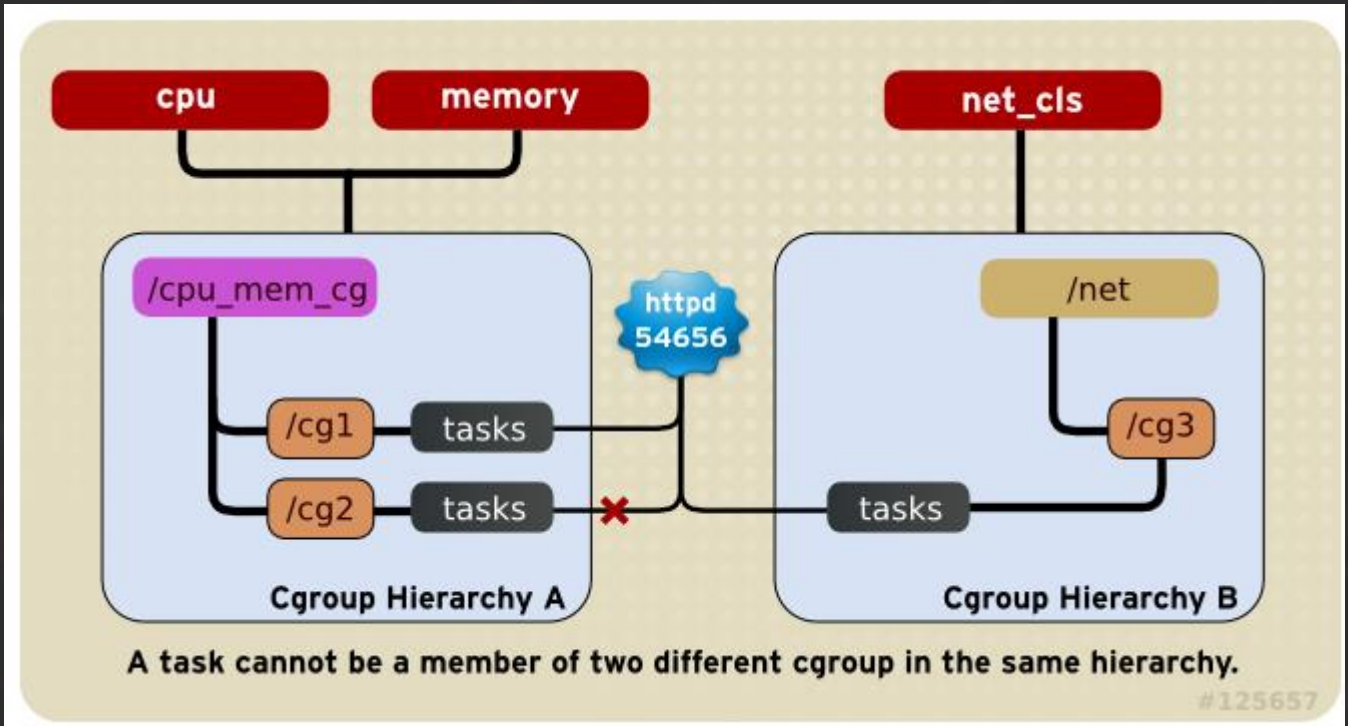
- Inne shelle:
  - rssh – zezwala na wykonywanie scp, sftp, rsync itd
  - sudosh - http://sourceforge.net/projects/sudosh
    - Pozwala zapisywać całą sesję użytkownika i ją odtwarzać
    - Pozwala określać jakie operacja są dozwolone dla usera
    - Trochę outdated – lepiej sudosh3
  - lbsh (small, fast, secure): http://sourceforge.net/projects/ibsh/

Etap 3 – control groups

- zarządzanie zasobami w prosty sposób (ulimits, nice, limits.conf)
- Potraficie przypisać 50 IOPS dla dowolnego procesu?
- A może limit 100Kbp/s dla wybranego usera?
- Problemy z memory–leakami w Javie?



Etap 3 – control groups

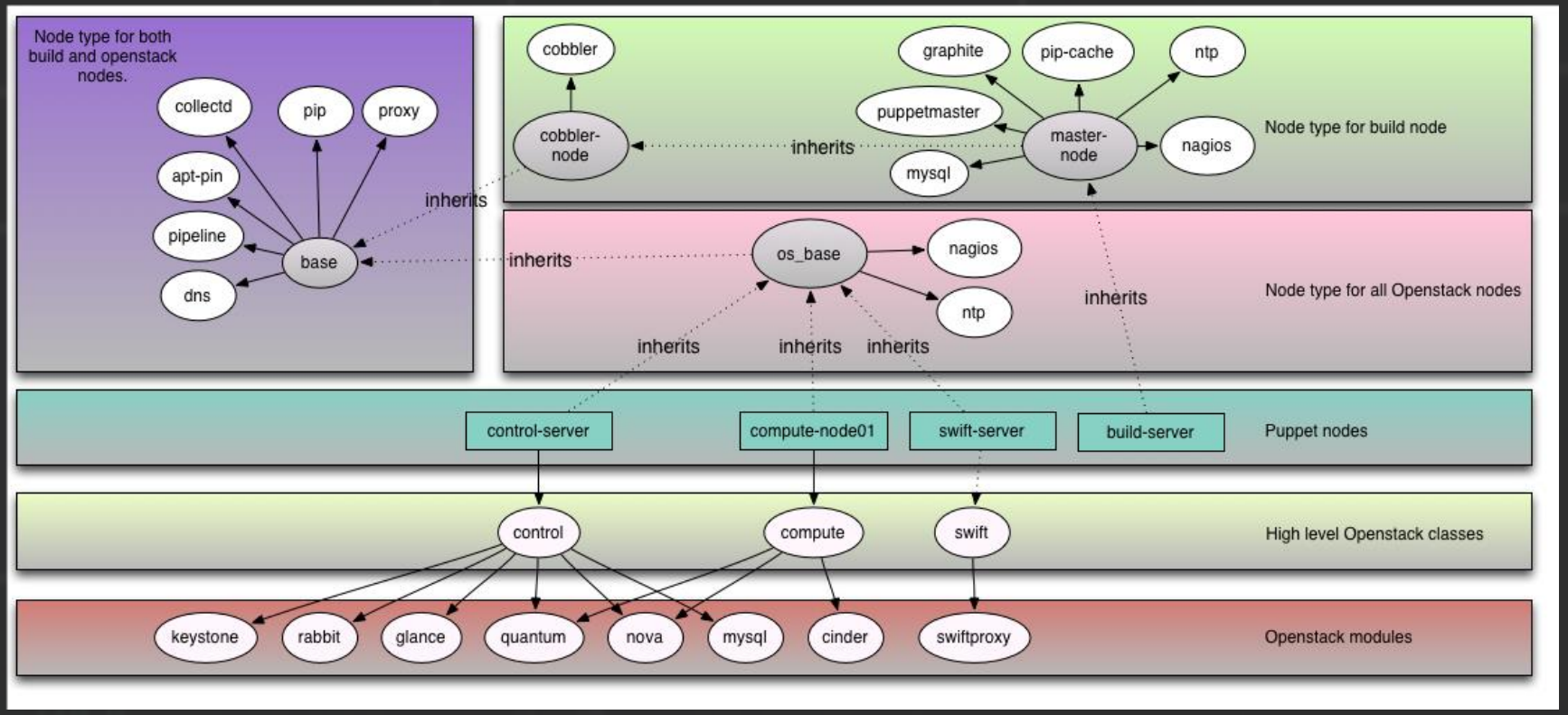


https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Resource_Management_Guide/ch01.html

- Debian & RHEL friendly
- Można uruchamiać aplikacje w kontekście cgroupy
- Można żyjący proces przypisać do konkretnej cgroupy

Etap 3 – webaplikacja

- OpenStack?



Trochę komplikacji. “Out of the box” tylko w internetach widzieli.
A zrób to wydajnie i bezpiecznie w kilka godzin ;)

Etap 3 – webaplikacja

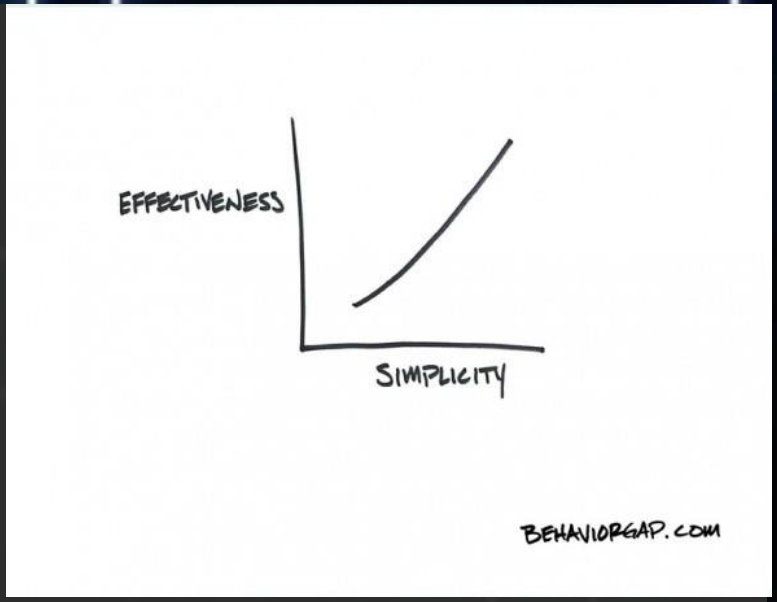
Commodore OS ???

Etap 3 – webaplikacja

Commodore OS Vision FTW!

Etap 3 – webaplikacja

- Apache + mod_security
- mod_security + OWASP rules
- PHP & Python :)
- Prostota!
- Sterowanie VM za pomocą screena i prostego pseudo – demona:
  - while(1) do: zarządzaj_VMkami();
- I to po prostu działa!

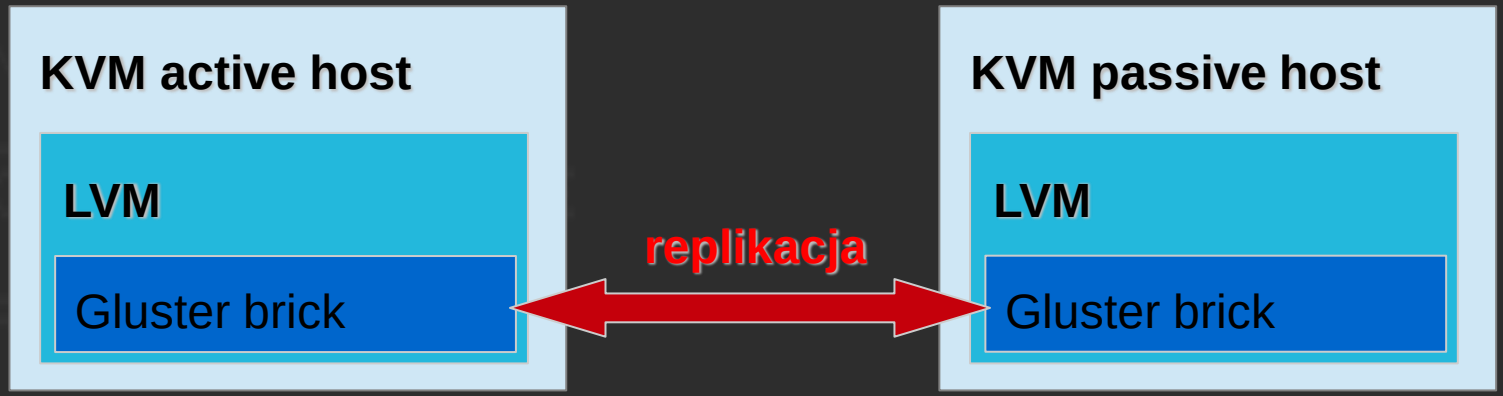


Etap 3 – nagrywanie sesji SSH

- Nagrywanie każdej sesji (tak, również screen)
- Nagrywanie real-time (a nie flush co jakiś czas)
- sudosh3 (fork sudosh) – powłoka pośrednicząca
- auditd – niskopoziomowe narzędzie logujące wybrane syscalle
- Asciinema (ascii.io, Marcin Kulik) – super, ale nie jako “audit”
- Ttyrec – outdated: http://0xcc.net/ttyrec/index.html.en
- Ssh logging patch - outdated: http://www.kdvelectronics.eu/ssh-logging/ssh-logging.html

Etap 3 – bezpieczeństwo danych

- Gdyby tak stracić choć jedną VMkę... brr.
- Ocena ryzyka – co nam wystarcza?
  - RAID1 / Mirror – “zazwyczaj” w skali kwartału wystarcza
  - Backupy – przydatne ;) RAID / replikacja to nie backup...
  - GlusterFS / DRBD – jeśli tylko mamy na to zasoby to polecam :)




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