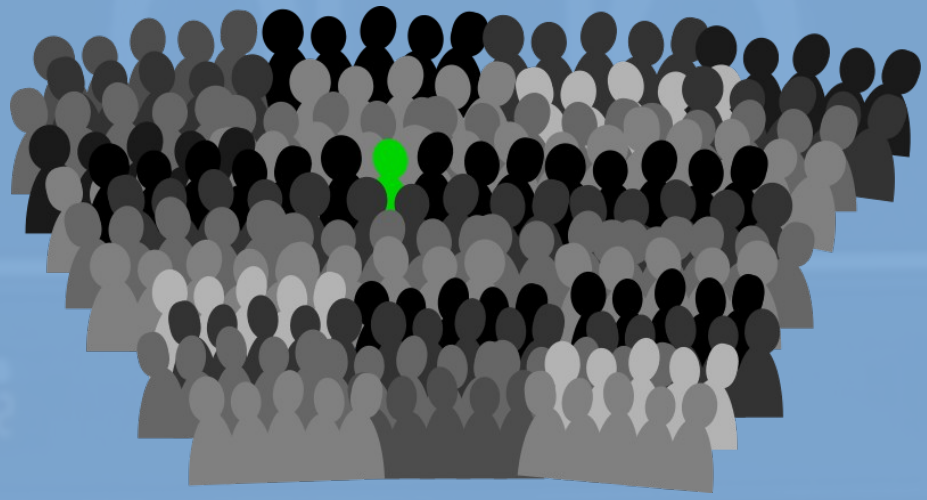
Maciej Lasyk

Ganymede

OWASP Poland, 2013-10-17

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

Zgłoszenie	Problem: spora liczba kandydatów (>100)
1 etap – proste zadanie	Cel: odsianie ludzi z innej bajki (80% odrzuceń!)
2 etap – tel/social.eng.	Cel: poznanie, manipulacja
3 etap – challenge	Global Thermonuclear War ;)

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)

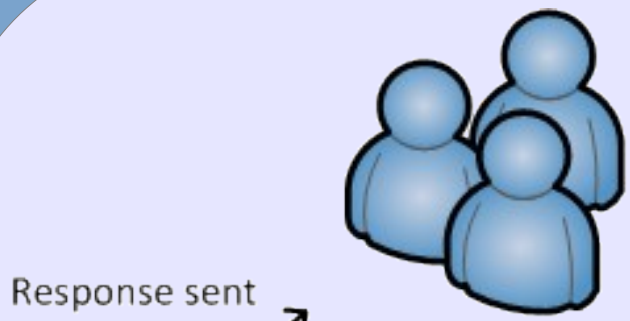
GPG us ur CV using http://..../gpg.asc

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



Clients connect to
Node's HTTP server

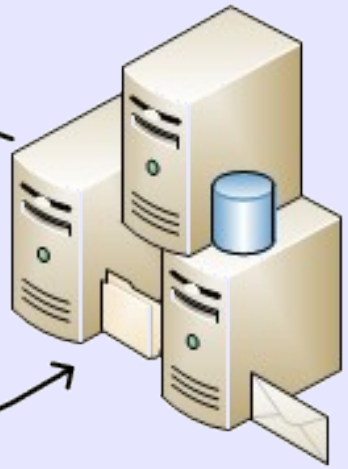
- Event driven
- Event loop
- Callbacks
- SPA, async, REST, Json

Response sent
To clients

Node's
Event Loop
Mono-threaded

Responses via
callback

Register callback



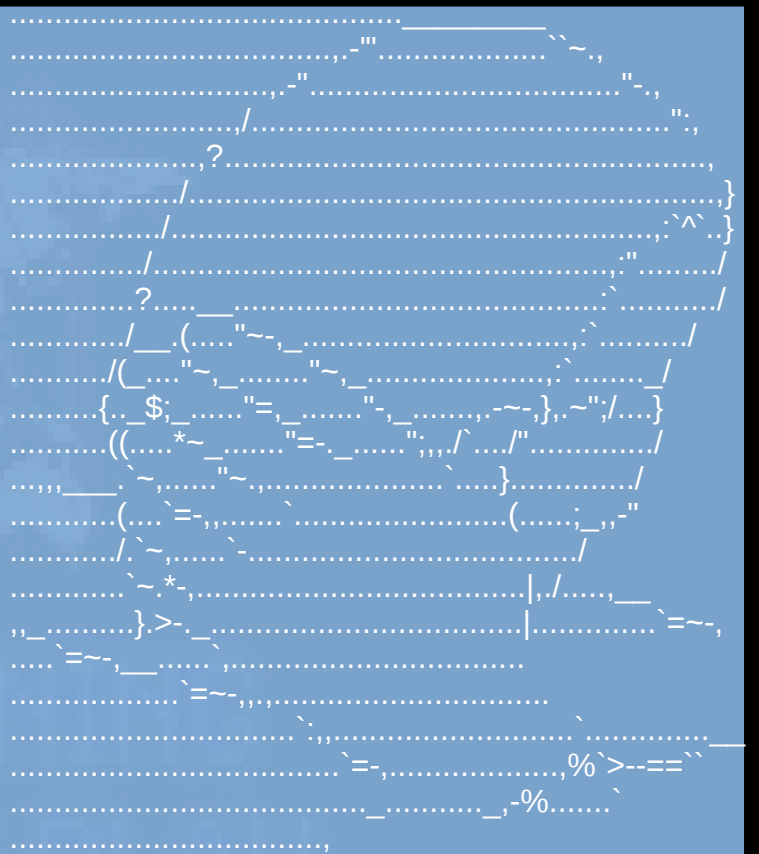
Long operations are
deffered to workers
threads

Threads are handled by
Node internally

http://magnetik.github.io/

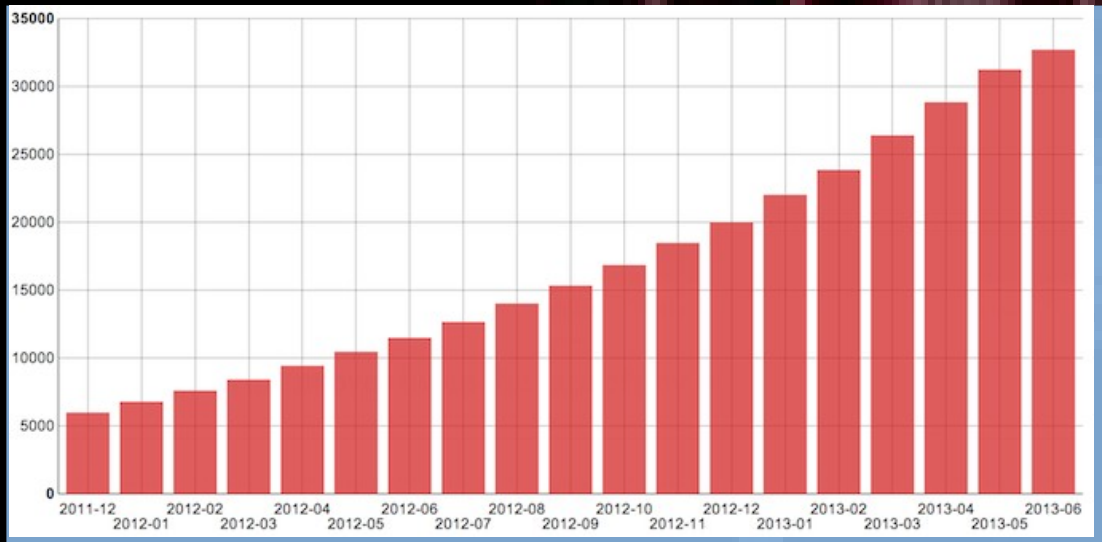
Node.js - zagrożenia

- Brak logów.
- Brak obsługi błędów
- Brak konfiguracji – jednak to może “+”?
- Brak filtrów sprawdzających user-input
- Evil eval(). Server-side XSS
- Moduły npm – kto i jak je tworzy?



https://media.blackhat.com/bh-us-11/Sullivan/BH_US_11_Sullivan_Server_Side_WP.pdf

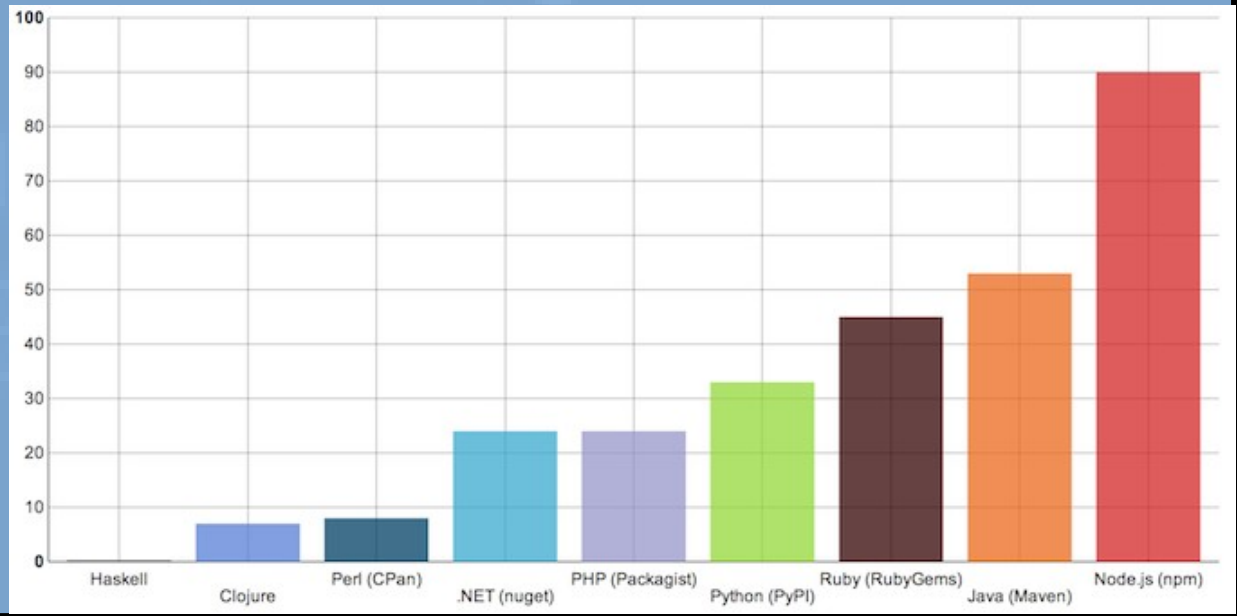
Node.js - npm



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

Przyrost modułów npm

Porównanie ilości modułów Node.js i innych technologii



Node.js – jak żyć?

- Używajmy frameworków: https://npmjs.org/ (express.js)
- Sprawdzajmy moduły npm: https://nodesecurity.io/
- Edukacja: https://github.com/toolness/security-adventure
- 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 >>>

Node.js – SELinux sandbox

Etap 2 – social engineering

Etap 3 - wirtualizacja

Etap 3 – bezpieczeństwo sieci

Etap 3 – proces bootowania, VNC

Etap 3 – restricted shells

Etap 3 – control groups

Etap 3 – webaplikacja

Etap 3 – nagrywanie sesji

Etap 3 – bezpieczeństwo danych

Podsumowanie

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

Ganymede

OWASP Poland, 2013-10-17

THE ONLY WINNING MOVE IS NOT TO PLAY