



## Από την ανάπτυξη στην εγκατάσταση (ESaaS §12.1)



1



## Επισκόπηση θεμάτων

- Διαθεσιμότητα & αποκρισιμότητα
- Ardex
- Παρακολούθηση
- Αναβαθμίσεις & σημαίες χαρακτηριστικών
- Ανακούφιση βάσης δεδομένων: ευρετήρια & καταχρηστικά ερωτήματα
- Ανακούφιση βάσης δεδομένων: κρυφή αποθήκευση
- Προστασία δεδομένων πελατών

2



## Ανάπτυξη και εγκατάσταση

---

Ανάπτυξη:

- Έλεγχος για να διασφαλιστεί ότι η εφαρμογή σας λειτουργεί όπως σχεδιάστηκε

Εγκατάσταση:

- Έλεγχος για να διασφαλιστεί ότι η εφαρμογή σας λειτουργεί όταν χρησιμοποιείται με τρόπους για τους οποίους δεν σχεδιάστηκε για να χρησιμοποιείται

---

3



## Τα άσχημα νέα

---

- «Οι χρήστες είναι άσχημο πράγμα»
- μερικά σφάλματα εμφανίζονται μόνο υπό πίεση
- περιβάλλον παραγωγής != περιβάλλον ανάπτυξης
- ο κόσμος είναι γεμάτος σκοτεινές δυνάμεις
- και ηλίθιους

---

4



## Τα καλά νέα: το PaaS κάνει την εγκατάσταση πολύ ευκολότερη

- πάρε έναν Εικονικό Ιδιωτικό Διακομιστή (Virtual Private Server, VPS), ίσως στο νέφος
- εγκατέστησε και ρύθμισε τα Linux, Rails, Apache, *mysqld*, *openssl*, *sshd*, *ipchains*, *squid*, *qmail*, *logrotate*...
- διόρθωσε σχεδόν κάθε εβδομάδα ευπάθειες ασφάλειας
- αντιμετώπισε την *Κόλαση Βιβλιοθηκών* (Library Hell)
- βελτιστοποίησε όλα τα απαραίτητα στοιχεία για να έχει το μέγιστο δυνατό όφελος
- Μάθε πώς να αυτοματοποιήσεις την οριζόντια κλιμάκωση

5



## Ο στόχος μας: δουλεύουμε με PaaS!

Το PaaS χειρίζεται...	Εμείς χειριζόμαστε...
Τα «εύκολα» επίπεδα οριζόντιας κλιμάκωσης	Την ελαχιστοποίηση του φόρτου της βάσης δεδομένων
Τη βελτιστοποίηση της απόδοσης σε επίπεδο συστατικών μερών	Τη ρύθμιση της απόδοσης σε επίπεδο εφαρμογής (π.χ. κρυφή αποθήκευση)
Την ασφάλεια σε επίπεδο υποδομής	Την Ασφάλεια σε επίπεδο εφαρμογής

Είναι πράγματι εφικτό;

- Το Pivotal Tracker & το Basecamp εκτελούνται το καθένα σε μια μοναδική ΒΔ (128GB «συσκευασία» μαζικής παραγωγής <10 χιλ. δολ.)
- Πολλές εφαρμογές SaaS δεν επικοινωνούν άμεσα με τον έξω κόσμο (εσωτερικές ή περιορισμένου ενδιαφέροντος)

6



## Ορισμός απόδοσης & ασφάλειας

- Διαθεσιμότητα ή χρόνος λειτουργίας (Uptime)  
*Σε ποιο % του χρόνου ο ιστότοπος λειτουργεί & είναι προσπελάσιμος;*
- Αποκρισιμότητα
  - Πόση ώρα μετά το «κλικ» λαμβάνει ο χρήστης απόκριση;
- Δυνατότητα κλιμάκωσης
  - Καθώς αυξάνει το πλήθος χρηστών, μπορείς να διατηρήσεις την αποκρισιμότητα χωρίς να αυξήσεις το κόστος/χρήστη;
- Απόρρητο
  - Περιορίζεται η προσπέλαση των δεδομένων στους κατάλληλους χρήστες;
- Έλεγχος αυθεντικότητας
  - Μπορούμε να εμπιστευτούμε ότι ο χρήστης είναι αυτός που ισχυρίζεται ότι είναι;
- Ακεραιότητα δεδομένων
  - Είναι εμφανής η τροποποίηση των ευαίσθητων δεδομένων των χρηστών;

Απόδοση  
Σταθερότητα

Ασφάλεια

7

Έστω R = διαθεσιμότητα εφαρμογής RottenPotatoes

H = διαθεσιμότητα του Heroku

C = διαθεσιμότητα σύνδεσης Διαδικτύου

P = αντίληψη του Armando για τη διαθεσιμότητα της RP

Ποια σχέση ισχύει μεταξύ αυτών των ποσοτήτων;

☐  $P \leq C \leq H \leq R$

☐  $P \geq \min(C, H, R)$

☐  $P \leq C \leq \min(H, R)$

☐ Δεν γνωρίζουμε χωρίς επιπλέον πληροφορίες



8

8



## Σημείωση για τον έλεγχο/κάλυψη

```
def index
  @admins = Admin.all
  @doctors = Doctor.all
  @shifts = Shift.all
  @users_awaiting_approver = User.where(:type =>nil)
end
```

9



# ΚΟΨΤΕ

10

10



## Ποσοτικοποίηση της διαθεσιμότητας & αποκρισιμότητας (ESaaS §§12.2, 12.5)

© 2013 Armando Fox & David Patterson, all rights reserved

11



## Είναι σημαντικός ο χρόνος απόκρισης;

- Πόσο σημαντικός είναι ο χρόνος απόκρισης;\*
  - Amazon: +100ms => 1% πτώση στις πωλήσεις
  - Yahoo!: +400ms => 5-9% πτώση στην κυκλοφορία
  - Google: +500ms => 20% λιγότερες αναζητήσεις
- Κλασικές μελέτες (Miller 1968, Bhatti 2000)
  - <100 ms είναι «στιγμιαίο»
  - >7 sec είναι χρόνος εγκατάλειψης

Jeff Dean,  
Google Fellow



«Η ταχύτητα είναι  
χαρακτηριστικό»

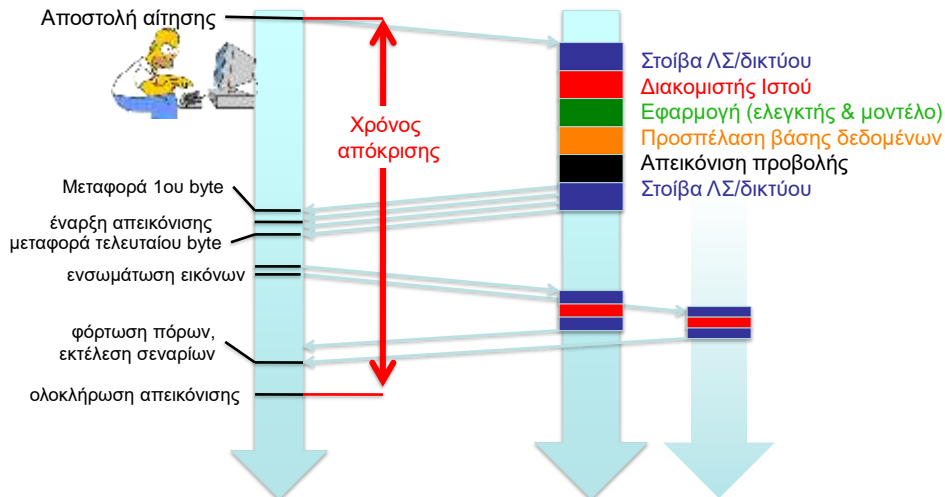
- <http://code.google.com/speed>

Πηγή: Nicole Sullivan (Yahoo! Inc.), *Design Fast Websites*, <http://www.slideshare.net/stubbornella/designing-fast-websites-presentation> 12

12



## Πού πηγαίνει ο χρόνος; (διακομιστής/δίκτυο)



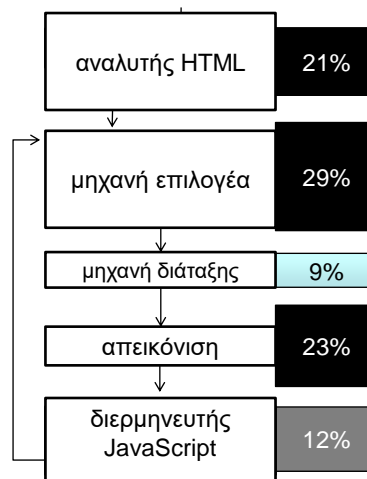
13

13



## Πού πηγαίνει ο χρόνος; (πελάτης)

- Επιλογείς CSS + διερμηνευτής JavaScript = 41% συνολικού χρόνου απεικόνισης (rendering) στον πελάτη
  - Ειδικά επιλογείς που απαιτούν διάσχιση του δέντρου DOM, π.χ. `div > li`
- Οι φυλλομετρητές ανταγωνίζονται ως προς την ταχύτητα του διερμηνευτή JavaScript => η απόδοση επιλογέα/αναλυτή αποδεικνύεται όλο και πιο συχνά η αιτία επιβάρυνσης!
- Δουλειά στο UC Berkeley (Prof. Ras Bodík's group):
  - Παραλληλοποίηση συντακτικής ανάλυσης & επιλογέων CSS
  - Αυξανόμενη χρήση της GPU για απεικόνιση



Courtesy Leo Meyerovich, UC Berkeley

14

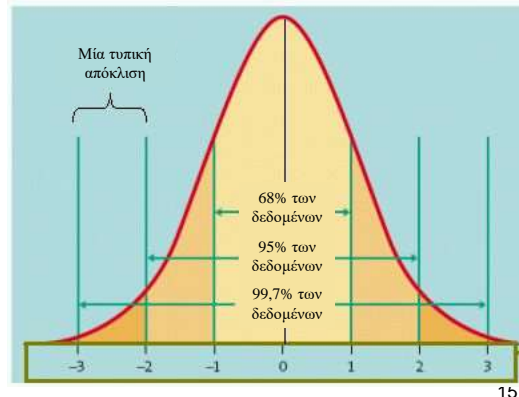


## Απλοποιημένη (& εσφαλμένη) άποψη του χρόνου απόκρισης

- Για τυπική κανονική κατανομή χρόνου απόκρισης γύρω από τη μέση τιμή:  $\pm 2$  τυπικές αποκλίσεις γύρω από τη μέση τιμή είναι το 95% διάστημα εμπιστοσύνης

• Μέσος χρόνος απόκρισης  $T$  σημαίνει:

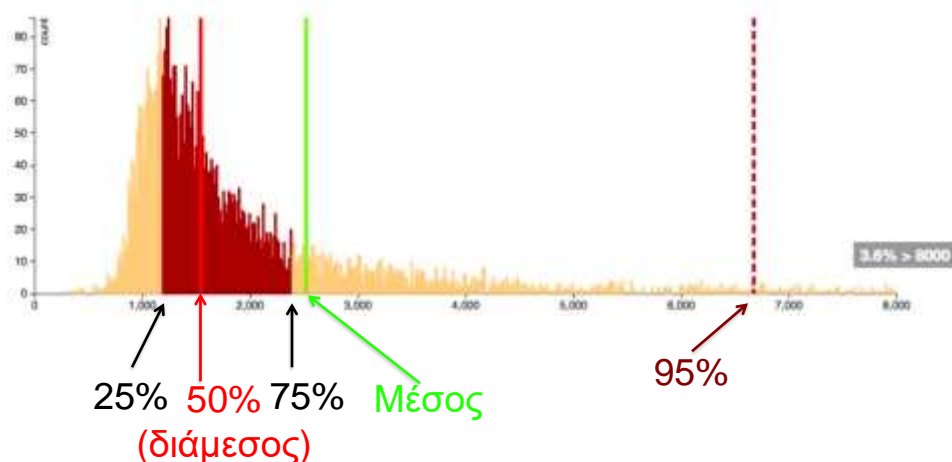
- 95% χρηστών λαμβάνουν  $T+2\sigma$
- 99,7% χρηστών λαμβάνουν  $T+3\sigma$



15



## Ένα πραγματικό παράδειγμα



Ευγενική χορηγεία του Bill Kayser, Distinguished Engineer, New Relic. <http://blog.newrelic.com/breaking-down-apdex>

Χρησιμοποιείται κατόπιν άδειας του δημιουργού.

16



Ποιο από τα ακόλουθα είναι το ΛΙΓΟΤΕΡΟ πιθανό, στις περισσότερες περιπτώσεις, να βελτιώσει τον χρόνο απεικόνισης/χρόνο παρουσίασης για μια εφαρμογή SaaS όπως η RottenPotatoes;



- ☐ Χρήση απλών επιλογέων, όπως ο `#id`, αντί ένθετων, όπως ο `p > span`
- ☐ Χρήση τεχνικών εφαρμογής στυλ με CSS αντί JavaScript για «πλοήγηση με καρτέλες» και εφέ κίνησης με την αιώρηση του δείκτη του ποντικιού (hovering)
- ☐ Αύξηση του αποτελεσματικού εύρους ζώνης μεταξύ του διακομιστή και των πελατών
- ☐ Χρήση διοχέτευσης αγαθών του Rails για την απομείωση (minify) της JavaScript και τη συγκέντρωση όλου του κώδικα JavaScript σε ένα μόνο αρχείο

17

17



# Κόψτε

18

18



# SLOs, Apdex, και κλιμακωσιμότητα (ESaaS §12.2)

© 2013 Armando Fox & David Patterson, all rights reserved

19



## Κλιμακωσιμότητα

- Ένας «παραφορτωμένος» και καταχρασμένος όρος – σημαίνει διαφορετικά πράγματα σε διαφορετικά πεδία
- Για το SaaS: *καθώς αυξάνεται το πλήθος των χρηστών, ο χρόνος απόκρισης παραμένει ίδιος για τον χρήστη*
  - *Ο χρόνος απόκρισης είναι σημαντικό μετρικό του διαδραστικού SaaS*
- Επιπλέον ιδανικά θέλουμε το εξής: *καθώς το πλήθος των χρηστών αυξάνεται, το κόστος εξυπηρέτησης του κάθε χρήστη παραμένει ίδιο (ή μειώνεται)*
  - Ένα πιθανό μετρικό: *χρήστες ανά διακομιστή ανά \$*
  - Αποτυπώνει τις επιδράσεις της *διεκπεραιωτικότητας* (ή «εύρους ζώνης»)

20

20



## Στόχος Επιπέδου Υπηρεσίας (SLO)

- Χρόνος ικανοποίησης αίτησης χρήστη («καθυστέρηση» ή «χρόνος απόκρισης»)
- SLO: Αντί για χειρότερη περίπτωση ή μέσο όρο: ποιο % χρηστών παίρνει αποδεκτή απόδοση
- Προσδιορίζει εκατοστημόριο, επιθυμητό χρόνο απόκρισης, χρονικό περιθώριο
  - π.χ.,  $99\% < 1$  δευτερόλεπτο, σε περιθώριο 5 λεπτών
  - γιατί το χρονικό περιθώριο είναι σημαντικό;
- *Συμφωνητικό* επιπέδου υπηρεσίας (service level agreement, SLA) είναι ένα SLO με το οποίο δεσμεύεται με συμβόλαιο ο πάροχος

21

21



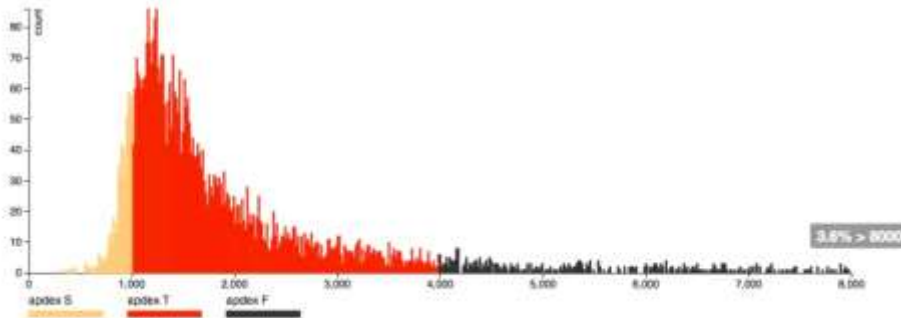
## Arpdex: απλοποιημένο SLO

- Με δεδομένο ένα κατώφλι καθυστέρησης  $T$  για την ικανοποιητική εμπειρία χρήστη:
  - *Ικανοποιητικές* αιτήσεις διαρκούν  $t \leq T$
  - *Ανεκτές* αιτήσεις διαρκούν  $T \leq t \leq 4T$
  - $\text{Arpdex} = (\# \text{ικανοποιητικών} + 0.5(\# \text{ανεκτών})) / \# \text{αιτήσεων}$
  - 0,85 μέχρι 0,93 γενικά «καλό»
- **Προειδοποίηση!** Μπορεί να κρύψει *συστηματικές* έκτοπες τιμές (outliers) αν δεν χρησιμοποιηθεί σωστά!
  - π.χ., η κρίσιμη ενέργεια συμβαίνει μία φορά κάθε 15 «κλικ» αλλά διαρκεί  $10x \Rightarrow (14+0)/15 > 0.9$

22



## Οπτικοποίηση Apdex

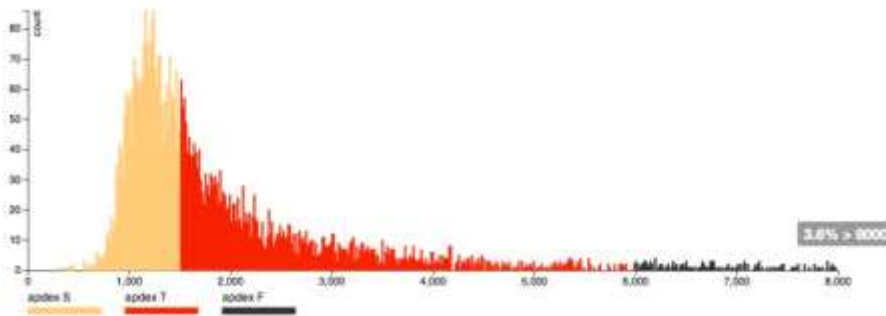


$T=1000\text{ms}$ ,  $\text{Apdex} = 0.49$

23



## Οπτικοποίηση Apdex



$T=1500\text{ms}$ ,  $\text{Apdex} = 0.7$

24



## Τι πρέπει να κάνουμε αν ο ιστότοπος είναι αργός;

- Μικρός ιστότοπος: υπερπαροχή (overprovision)
  - εφαρμόζεται στα επίπεδα παρουσίασης & λογικής
  - πριν την υπολογιστική νέφους, ήταν οδυνηρό
  - σήμερα, γίνεται σχεδόν αυτόματα (π.χ. Rightscale)
- Μεγάλος ιστότοπος: πρόβλημα
  - Παροχή 10% σε ιστότοπο με 10.000 υπολογιστές  
→ 1000 αδρανείς υπολογιστές
- *Ιδέα: τα ίδια προβλήματα που μας βγάζουν έξω από το επίπεδο που είναι φιλικό στο PaaS είναι εκείνα που μας ταλαιπωρούν σε μεγαλύτερους ιστότοπους!*

25

Ο στόχος λειτουργίας της RottenPotatoes είναι 99,9%. Χθες υπήρχε μια ώρα διακοπή. Ποια πρόταση είναι σωστή:



- ☐ Εξαιτίας της διακοπής, η RottenPotatoes δεν έχει ελπίδα να πετύχει τον στόχο λειτουργίας αυτό το έτος
- ☐ Η RottenPotatoes μπορεί να πετύχει τον στόχο λειτουργίας αν δεν υπάρχουν άλλες διακοπές αυτό το έτος
- ☐ Αν δεν υπήρξαν χρήστες που προσπάθησαν να επισκεφτούν τον ιστότοπο κατά τη διακοπή, δεν επηρεάστηκε ο χρόνος λειτουργίας
- ☐ Δεν υπάρχουν επαρκείς πληροφορίες για να καθοριστεί αν η RottenPotatoes μπορεί να πετύχει τον χρόνο λειτουργίας που αντιλαμβάνεται ο χρήστης

26

26



# ΚΟΨΤΕ

27

27

Ο Armando  
φωτογραφία με ένα:

θαλάσσιο\_\_\_\_\_

- a) RSpec
- b) Cucumber
- c) Jasmine

στο Tunich Reef,  
Cozumel, MX



28

28



# ΚΟΨΤΕ

29

29



Συνεχής ολοκλήρωση &  
συνεχής εγκατάσταση  
(ESaaS §12.3)

© 2013 Armando Fox & David Patterson, all rights reserved

30



## Κυκλοφορίες τότε και τώρα: Πάρτι κυκλοφορίας των Windows 95



31



## Κυκλοφορίες τότε και τώρα

- Facebook: η κύρια διακλάδωση ωθούνταν μία φορά την εβδομάδα, με στόχο μία φορά την ημέρα (Bobby Johnson, Διευθυντής Μηχανικών, προς το τέλος του 2011)
- Amazon: αρκετές εγκαταστάσεις ανά εβδομάδα
- StackOverflow: πολλές εγκαταστάσεις ανά ημέρα (Jeff Atwood, συνιδρυτής)
- GitHub: δεκάδες εγκαταστάσεις ανά ημέρα (Zach Holman)
- **Λογική: κίνδυνος == πλήθος ωρών μηχανικών που επενδύθηκαν στο προϊόν από την τελευταία εγκατάσταση!**

*Όπως η ανάπτυξη και η εισαγωγή χαρακτηριστικών, η εγκατάσταση θα πρέπει να είναι ένα **ασήμαντο γεγονός** που συμβαίνει όλη των ώρα*

32





## Επιτυχημένη εγκατάσταση

- **Αυτοματοποίηση:** συνεπής διαδικασία εγκατάστασης
  - Γίνεται ήδη από ιστότοπους PaaS όπως οι Heroku, CloudFoundry
  - Χρήση εργαλείων όπως το Capistrano για αυτο-φιλοξενούμενους ιστότοπους
- **Συνεχής ολοκλήρωση:** έλεγχος ολοκλήρωσης της εφαρμογής πέρα από αυτό που κάνει ο κάθε προγραμματιστής
  - Η εισαγωγή κώδικα πριν την κυκλοφορία πυροδοτεί CI (continuous integration)
  - Επειδή οι εισαγωγές είναι συχνές, λειτουργεί πάντα η CI
  - Συνηθισμένη στρατηγική: ολοκλήρωση με το GitHub



[https://github.com/saasbook/hw2\\_rottenpotatoes/admin/hooks](https://github.com/saasbook/hw2_rottenpotatoes/admin/hooks)

33



## Γιατί CI;

- Διαφορές μεταξύ περιβαλλόντων ανάπτυξης & παραγωγής
- Έλεγχος σε διαφορετικούς φυλλομετρητές ή εκδόσεις
- Έλεγχος ολοκλήρωσης υπηρεσιοστρεφών αρχιτεκτονικών όταν οι απομακρυσμένες υπηρεσίες λειτουργούν προβληματικά
- Θωράκιση: προστασία από επιθέσεις
- Έλεγχος καταπόνησης/διάρκειας νέων χαρακτηριστικών/μονοπατιών κώδικα
- Παράδειγμα: η CI του Salesforce εκτελεί 150K+ ελέγχους και δημιουργεί αυτόματα αναφορά σφάλματος όταν ο έλεγχος αποτυγχάνει

34



## Συνεχής εγκατάσταση

- Ωθηση => CI => εγκατάσταση *πολλές φορές ανά ημέρα*
  - η εγκατάσταση μπορεί να ολοκληρωθεί αυτόματα με εκτελέσεις CI
- Επομένως δεν έχουν νόημα οι κυκλοφορίες λογισμικού;
  - Εξακολουθούν να είναι χρήσιμες ως ορόσημα ορατά στον πελάτη
  - Προστίθενται ετικέτες (tag) σε συγκεκριμένες επικυρώσεις με τα ονόματα κυκλοφορίας
 

```
git tag 'happy-hippo' HEAD
git push --tags
```
  - Ή προσδιορισμός κυκλοφορίας με το αναγνωριστικό επικύρωσης Git (commit ID)

35

Η RottenPotatoes μόλις έλαβε μερικά νέα χαρακτηριστικά AJAX. Πού είναι λογικό να ελεγχθούν αυτά τα χαρακτηριστικά;



- ☐ Χρήση *autotest με RSpec+Cucumber*
- ☐ Στη CI
- ☐ Στο περιβάλλον προσωρινής εγκατάστασης
- ☐ Σε όλα τα παραπάνω

36

36



# ΚΟΨΤΕ

37

37



Αναβαθμίσεις & σημαίες  
χαρακτηριστικών  
(ESaaS §12.4)

Armando Fox

© 2013 Armando Fox & David Patterson, all rights reserved

38



## Το πρόβλημα με τις αναβαθμίσεις

- Τι γίνεται αν ο αναβαθμισμένος κώδικας διατίθεται σε πολλούς διακομιστές;
  - Κατά τη διάθεση, μερικοί θα έχουν την έκδοση  $n$  και άλλοι θα έχουν την έκδοση  $n+1$ ...θα δουλέψει αυτό;
- Τι γίνεται αν ο αναβαθμισμένος κώδικας συνοδεύεται από μια μετάβαση (migration) σχήματος;
  - Η έκδοση σχήματος  $n+1$  χαλάει τον τρέχοντα κώδικα
  - Ο νέος κώδικας δεν θα δουλέψει με το τρέχον σχήμα

39



## Απλοϊκή ενημέρωση

1. Η υπηρεσία τίθεται εκτός λειτουργίας
  2. Εφαρμόζεται καταστρεπτική μετάβαση, με αντιγραφή δεδομένων
  3. Εγκατάσταση του νέου κώδικα
  4. Η υπηρεσία τίθεται ξανά σε λειτουργία
- Μπορεί να οδηγήσει σε απαράδεκτο χρόνο εκτός λειτουργίας

<http://pastebin.com/5dj9k1cj>

40



## Αυξητικές αναβαθμίσεις με σημαίες χαρακτηριστικών

1. Μη καταστρεπτική μετάβαση <http://pastebin.com/TYx5qaSB>
2. Η μέθοδος εγκατάστασης προστατεύεται με σημαία χαρακτηριστικού <http://pastebin.com/qqrLfuQh>
3. Αλλαγή της τιμής της σημαίας – αν υπάρξει πρόβλημα, επανέρχεται η προηγούμενη τιμή
4. Όταν μετακινηθούν όλες οι εγγραφές, εγκατάσταση του νέου κώδικα χωρίς τη σημαία χαρακτηριστικού
5. Εφαρμογή της μετάβασης για αφαίρεση παλιών στηλών

41



## «Αναίρεση» αναβάθμισης

- Χτυπά η καταστροφή... χρησιμοποιούμε αντίστροφη μετάβαση;
  - είναι εξονυχιστικά ελεγμένα;
  - είναι η μετάβαση αντιστρέψιμη;
  - είμαστε βέβαιοι ότι δεν εφάρμοσε κάποιος άλλος μια μη αντιστρέψιμη μετάβαση;
- Χρήση σημαιών χαρακτηριστικών
  - οι αντίστροφες μεταβάσεις είναι κυρίως για την ανάπτυξη

42



## Άλλες χρήσεις των σημαιών χαρακτηριστικών

- Προκαταρκτικός έλεγχος: βαθμιαία διάθεση ενός χαρακτηριστικού σε αυξανόμενο πλήθος χρηστών
  - π.χ., για έλεγχο προβλημάτων απόδοσης
- Έλεγχος A/B
- Πολύπλοκο χαρακτηριστικό που καλύπτει πολλές εγκαταστάσεις
- το `gem rollout` καλύπτει αυτές και άλλες περιπτώσεις

43

Ποια από τις παρακάτω είναι ΚΑΚΗ θέση , (αν υπάρχει) για την αποθήκευση της τιμής (π.χ., `true/false`) μιας σημαίας χαρακτηριστικού;



- ☐ Ένα αρχείο `YAML` στον κατάλογο `config/` της εφαρμογής
- ☐ Μια στήλη σε έναν υπάρχοντα πίνακα βάσης δεδομένων
- ☐ Ένας ξεχωριστός πίνακας βάσης δεδομένων
- ☐ Αυτές είναι όλες καλές θέσεις για την αποθήκευση τιμών που αντιστοιχούν σε σημαίες χαρακτηριστικών

44

44



# ΚΟΨΤΕ

45

45



Παρακολούθηση  
(ESaaS §12.6)

Armando Fox

© 2013 Armando Fox & David Patterson, all rights reserved

46



## Είδη παρακολούθησης

- «Αν δεν το παρακολουθείς, μάλλον έχει χαλάσει»
- Κατά τον χρόνο ανάπτυξης (*δυναμική ανάλυση*)
  - Προσδιορίζονται πιθανά προβλήματα απόδοσης/σταθερότητας *πριν* εμφανιστούν στην παραγωγή
- Στην παραγωγή
  - Εσωτερικά: ενσώχληστρωση ενσωματωμένη στην εφαρμογή ή το πλαίσιο εργασίας (Rails, Rack, κλπ.)
  - Εξωτερικά: ενεργή διερεύνηση (probing) από άλλους ιστότοπους.

47



## Γιατί εξωτερική παρακολούθηση;

- Εντοπίζεται αν ο ιστότοπος είναι εκτός λειτουργίας
- Εντοπίζεται αν ο ιστότοπος είναι αργός για λόγους έξω από τα όρια μέτρησης της εσωτερικής παρακολούθησης
- Λαμβάνεται η άποψη του χρήστη από πολλές διαφορετικές θέσεις στο Διαδίκτυο
- Παράδειγμα: Pingdom

48

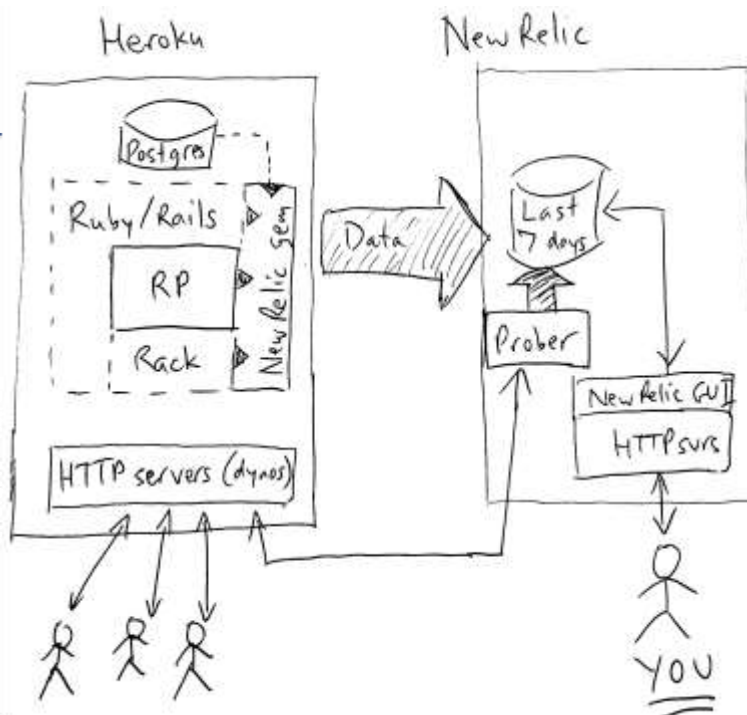




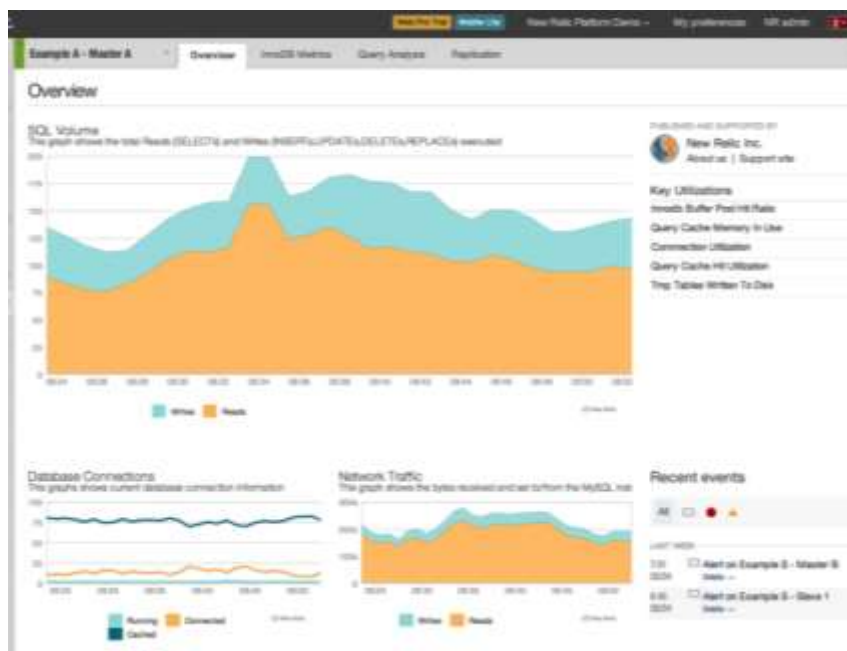
## Εσωτερική παρακολούθηση

- Πριν το SaaS/PaaS: *τοπικά*
  - Τοπική συλλογή & αποθήκευση πληροφοριών, π.χ. Nagios
- Σήμερα: *φιλοξενούμενη (hosted)*
  - Οι πληροφορίες συλλέγονται από την εφαρμογή σας αλλά αποθηκεύονται σε κεντρική θέση
  - Οι πληροφορίες είναι διαθέσιμες ακόμη και όταν η εφαρμογή είναι εκτός λειτουργίας
- Παράδειγμα: New Relic
  - είναι βολικό που έχει και κατάσταση ανάπτυξης και κατάσταση παραγωγής
  - το βασικό επίπεδο υπηρεσίας είναι δωρεάν για εφαρμογές Heroku

49



50



51



## Δείγματα εργαλείων παρακολούθησης

Τι παρακολουθείται	Επίπεδο	Παράδειγμα εργαλείου	Φιλοξενούμενο
Διαθεσιμότητα	ιστότοπος	pingdom.com	Ναι
Μη χειριζόμενες εξαιρέσεις	ιστότοπος	airbrake.com	Ναι
Αργές ενέργειες ελεγκτή ή ερωτήματα ΒΔ	εφαρμογή	newrelic.com (έχει επίσης κατάσταση ανάπτυξης/dev mode)	Ναι
«Κλικ», χρόνοι σκέψης	εφαρμογή	Google Analytics	Ναι
Ακεραιότητα διεργασίας & τηλεμετρία (διακομιστής MySQL, Apache, κ.λπ.)	διεργασία	god, monit, nagios	Όχι

- Ενδιαφέρον: Χαρακτηριστικά παρακολούθησης που μπορούν να διαβαστούν από τον πελάτη με το cucumber-newrelic

<http://pastebin.com/TaecHfND>

52



## Τι μετράμε;

- Έλεγχος καταπόνησης ή φορτίου: πόσο μπορεί να ζοριστεί το σύστημά μου...
  - ...πριν η απόδοση γίνει μη αποδεκτή;
  - ...πριν «κρεμάσει»;
- Συνήθως, ένα συστατικό θα είναι σημείο *συμφόρησης*
  - μια συγκεκριμένη προβολή, ενέργεια, ερώτημα, ...
- Οι ελεγκτές φορτίου μπορούν να είναι απλοί ή προηγμένοι
  - χρήση ενός μοναδικού URI ξανά και ξανά
  - χρήση μιας σταθερής ακολουθίας URI επαναληπτικά
  - αναπαραγωγή αρχείου καταγραφής

53

53



## Σφάλματα μακροβιότητας

- Η διαρροή πόρων (RAM, buffer αρχείων, πίνακας συνεδριών) είναι ένα κλασικό παράδειγμα
- Κάποιο λογισμικό υποδομής όπως το Apache κάνει ήδη *αναζωογόνηση*
  - με άλλα λόγια «κυλιόμενη επανεκκίνηση»
- Σχετικό: εξάντληση συνεδριών
  - Λύση: αποθήκευση ολόκληρου του `session[]` σε cookie (γίνεται εξ ορισμού στο Rails 3)

54



Ποιο από τα παρακάτω μάλλον δεν είναι μετρικό ιδιαίτερου ενδιαφέροντος για εσάς, τον χειριστή της εφαρμογής;

- ☐ Πιο αργά ερωτήματα
- ☐ Μέγιστη χρήση της ΚΜΕ
- ☐ Εκατοστημόριο 99% του χρόνου απόκρισης
- ☐ Χρόνος απεικόνισης των 3 πιο αργών προβολών

55

55



# ΚΟΨΤΕ

56

56



## Κρυφή αποθήκευση: Βελτίωση του χρόνου απεικόνισης & της απόδοσης της βάσης δεδομένων (ESaaS §12.6)

© 2013 Armando Fox & David Patterson, all rights reserved

57



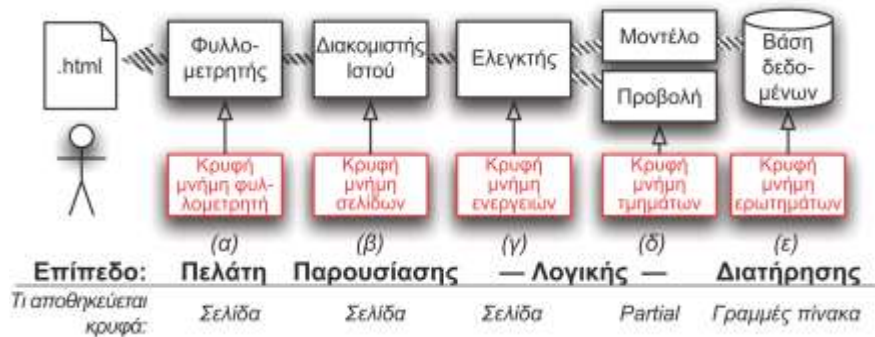
## Η πιο γρήγορη βάση δεδομένων είναι αυτή που δεν χρησιμοποιεί

- **Κρυφή αποθήκευση:** Αποφυγή χρήσης της βάσης δεδομένων αν το αποτέλεσμα ενός ερωτήματος δεν έχει αλλάξει
  1. Προσδιορίζεται τι θα αποθηκεύεται
    - πλήρης προβολή: αποθήκευση σελίδας & ενέργειας
    - τμήματα της προβολής: κρυφή αποθήκευση τμημάτων με επαναχρησιμοποιήσιμα τμήματα (partials)
  2. Ακύρωση (απόρριψη) *ανεπίκαιρων* κρυφά αποθηκευμένων εκδόσεων όταν αλλάζει η υποκείμενη ΒΔ

58

58

## Ροή κρυφής αποθήκευσης



59

## Κρυφή αποθήκευση σελίδων & ενεργειών

- Πότε: η έξοδος ολόκληρης της ενέργειας μπορεί να αποθηκευτεί κρυφά
  - Η κρυφή αποθήκευση σελίδας παρακάμπτει την ενέργεια ελεγκτή  
`cache_page :index`
  - Η κρυφή αποθήκευση ενέργειας εκτελεί πρώτα τα φίλτρα
- **Προειδοποίηση:** η κρυφή αποθήκευση βασίζεται στο URL σελίδας χωρίς τις προαιρετικές παραμέτρους "?"!
  - `/movies/index?rating=PG` = `movies/index`
  - `/movies/index/rating/PG` ≠ `movies/index`
- **Παγίδα:** μην αναμιγνύετε μονοπάτια κώδικα που ανήκει σε φίλτρο & κώδικα που δεν ανήκει σε φίλτρο στην ίδια ενέργεια!

60

60



## Παράδειγμα

- Κακό:

```

cache :page :index
def index
  if logged_in?
    ...
  else
    redirect_to login_path
  end
end

```

- Καλύτερο:

```

cache :page :public_index
cache :action :logged_in_index
before_filter :check_logged_in,
  :only => 'logged_in_index'
def public_index
  ...
end

def logged_in_index
  ...
end

```

61

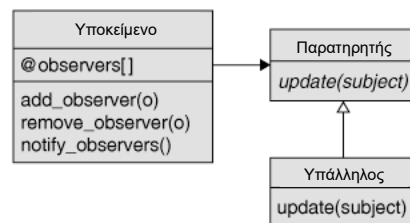
61



## Κρυφή αποθήκευση τμήματος για προβολές

- Αποθηκεύει κρυφά την HTML που προκύπτει από την απεικόνιση τμήματος μιας σελίδας (π.χ. partial)
- `cache "movies_with_ratings" do`  
`= render :collection => @movies`
- Πώς ανιχνεύουμε πότε οι κρυφά αποθηκευμένες εκδόσεις δεν συμφωνούν με τη βάση δεδομένων;
- Οι *Sweepers* χρησιμοποιούν το σχεδιαστικό υπόδειγμα Παρατηρητή για να διαχωρίσουν τη λογική χρόνου λήξης από την υπόλοιπη εφαρμογή

<http://pastebin.com/fCZJSimS>



62

62



## Πόσο βοηθάει η κρυφή αποθήκευση;

- Με ~1K ταινίες και ~100 κριτικές/ταινία στη RottenPotatoes στο Heroku, η **heroku logs** δείχνει:



- Μπορεί να εξυπηρετήσει 8x με 21x περισσότερους χρήστες με το ίδιο πλήθος διακομιστών αν χρησιμοποιείται κρυφή αποθήκευση

63

Επισκέπτες κάτω των 17 ετών στη RottenPotatoes δεν θα πρέπει να βλέπουν ταινίες με καταλληλότητα NC-17 σε καμία λίστα. Υπάρχει ένα φίλτρο ελεγκτή που μπορεί να αποφασίσει αν ένας χρήστης είναι κάτω των 17. Ποια είδη κρυφής αποθήκευσης θα ήταν κατάλληλα:



i) Σελίδας ii) Ενεργειών iii) Τμήματος

- ☐ (i) & (iii)
- ☐ (ii) & (iii)
- ☐ (iii) μόνο
- ☐ (i), (ii) και (iii)

64

64





# ΚΟΨΤΕ

65

65



Αποφυγή καταχρηστικών  
ερωτημάτων  
(ESaaS §12.8)

© 2013 Armando Fox & David Patterson, all rights reserved

66




## Να είστε ευγενικοί με τη βάση δεδομένων

- Η υπερβολική διόγκωση της βάσης δεδομένων σε ένα μηχάνημα == μεγάλη επένδυση: θρυμματισμός (sharding), αναπαραγωγή (replication), κ.λπ.
- Εναλλακτική: βρείτε τρόπους να μειώσετε το φορτίο της βάσης δεδομένων ώστε να παραμένει σε επίπεδο φιλικό προς το PaaS
  1. Χρήση **κρυφής αποθήκευσης** για μείωση του πλήθους των προσπελάσεων στη βάση δεδομένων
  2. Αποφυγή **προβλήματος «n+1 ερωτημάτων»** στις Συσχετίσεις
  3. Συνετή χρήση **ευρετηρίων**

67



## Πρόβλημα n+1 ερωτημάτων

- **Πρόβλημα:** εκτελείς n+1 ερωτήματα για να διασχίσεις μια συσχέτιση, αντί για 1 ερώτημα <http://pastebin.com/QKxqcbhk>
- **Λύση:** το gem `bullet` μπορεί να σε βοηθήσει να τα βρεις 
- **Μάθημα:** όλες οι αφαιρέσεις τελικά έχουν διαρροές!

68



## Πρόθυμη φόρτωση

- Απλοϊκός τρόπος:

```
@movie = movie.where( ... )
reviews = @movie.reviews
```

- Μπορεί να είναι γρηγορότερος:

```
@movie = movie.where( ... ).include(:reviews)
@movie.reviews.each do |review|
  # οι κριτικές μπορεί να έχουν φορτωθεί ήδη!
```

69



## Ευρετήρια

- Επιταχύνουν την προσπέλαση κατά την αναζήτηση πίνακα της ΒΔ κατά στήλη αντί με βάση το πρωτεύον κλειδί
  - π.χ., `Movie.where("rating = 'PG'")`
- Παρόμοιο με τη χρήση πίνακα κατακερματισμού
  - το εναλλακτικό είναι η *σάρωση πίνακα* – κακό!
  - ακόμη μεγαλύτερο όφελος αν η ιδιότητα έχει μοναδική τιμή
- Γιατί όχι ευρετήριο για κάθε στήλη;
  - καταλαμβάνει χώρο
  - θα πρέπει να ενημερώνονται όλα τα ευρετήρια όταν ενημερώνεται ο πίνακας

70



## Ποια ευρετήρια πρέπει να δημιουργήσετε;

- Στήλες ξένου κλειδιών, π.χ. το πεδίο `movie_id` στον πίνακα `Reviews`
  - γιατί;
- Στήλες που εμφανίζονται στις εντολές `where()` των ερωτημάτων ActiveRecord
- Στήλες με βάση τις οποίες γίνεται ταξινόμηση
- Χρησιμοποίησε το gem `rails_indexes` (στο GitHub) για προσδιορισμό ευρετηρίων που λείπουν (και αυτών που είναι περιττά!)



71



## Πόσο βοηθούν τα ευρετήρια;

- Χρόνος σε δευτερόλεπτα για ανάγνωση 100 κριτικών

Πλήθος κριτικών:	2000	20.000	200.000
Ανάγνωση 100, καθόλου ευρετήρια	0,94	1,33	5,28
Ανάγνωση 100, ευρετήρια FK	0,57	0,63	0,65
Απόδοση	166%	212%	208%

Επιβάρυνση απόδοσης ερωτημάτων δημιουργίας λόγω ευρετηρίων:

	200.00
Δημιουργία 1000, καθόλου ευρετήρια	9,69
Δημιουργία 1000, όλα τα ευρετήρια	11,30
Απόδοση	-17%

72

Έστω ότι η Movie έχει πολλούς Moviegoer μέσω των Reviews. Ποιο ευρετήριο ξένου κλειδιού θα βοηθούσε ΠΕΡΙΣΣΟΤΕΡΟ στην επιτάχυνση του ερωτήματος



`fans = @movie.moviegoers`

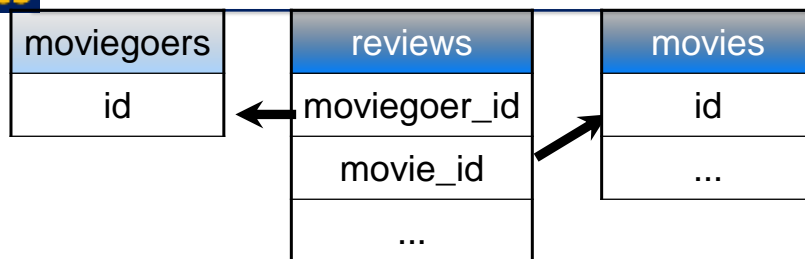
- `movies.review_id`
- `reviews.movie_id`
- `reviews.moviegoer_id`
- `moviegoers.review_id`

73

73



## has\_many :through



- `moviegoer: has_many :reviews`  
`has_many :movies, :through => :reviews`
- `movie: has_many :reviews`  
`has_many :moviegoers, :through => :reviews`
- `reviews: belongs_to :moviegoer`  
`belongs_to :movie`

74



# ΚΟΨΤΕ

75

75



Προστασία των δεδομένων  
των πελατών  
(ESaaS §12.9)

© 2013 Armando Fox & David Patterson, all rights reserved

76



## Συχνές επιθέσεις σε εφαρμογές

1. Υποκλοπή & SSL
  2. Επίθεση ενδιάμεσου/κατάληψη συνεδρίας
  3. Έγχυση SQL
  4. Πλαστογραφία αίτησης μεταξύ ιστότοπων (Cross-site request forgery, CSRF)
  5. Εκτέλεση σεναρίων μεταξύ ιστότοπων (Cross-site scripting, XSS)
  6. Μαζική ανάθεση τιμών σε ευαίσθητες ιδιότητες
- ...περισσότερα μπορείτε να διαβάσετε στο βιβλίο

77



## Επίπεδο Ασφαλών Υποδοχών (Secure Sockets Layer, SSL)

- Ιδέα: *κρυπτογράφηση* της κυκλοφορίας HTTP για αποφυγή υποκλοπών
- Πρόβλημα: για τη δημιουργία *ασφαλούς καναλιού*, δύο μέρη πρέπει πρώτα να *μοιραστούν ένα μυστικό*
- Αλλά στον Ιστό, τα δύο μέρη δεν γνωρίζονται μεταξύ τους
- Λύση: *κρυπτογραφία δημόσιου κλειδιού* (Rivest, Shamir, & Adelman, Βραβείο Τούρινγκ 2002)

78



## Τι κάνει και τι δεν κάνει το SSL

- Κάθε εντολέας έχει ένα *κλειδί* με δύο μέρη που ταιριάζουν
  - δημόσιο μέρος: όλοι μπορούν να το γνωρίζουν
  - ιδιωτικό μέρος: ο εντολέας το κρατάει κρυφό
  - όταν είναι γνωστό το ένα μέρος, δεν μπορεί να βρεθεί το άλλο
- Βασικός μηχανισμός: η *κρυπτογράφηση* με το ένα κλειδί απαιτεί *αποκρυπτογράφηση* από το άλλο
  - Αν ένα μήνυμα μπορεί να αποκρυπτογραφηθεί με το δημόσιο κλειδί του Bob, τότε πρέπει να το δημιούργησε ο Bob
  - Αν χρησιμοποιήσω το δημόσιο κλειδί του Bob για να δημιουργήσω ένα μήνυμα, μόνο ο Bob θα μπορεί να το διαβάσει

79



## Πώς δουλεύει το SSL (απλοποιημένο)

1. Ο ιστότοπος Bob.com αποδεικνύει την ταυτότητά του στην CA (Certificate Authority – Αρχή Έκδοσης Πιστοποιητικών)
  2. Η CA χρησιμοποιεί το *ιδιωτικό* της κλειδί για να δημιουργήσει ένα πιστοποιητικό που συσχετίζει αυτήν την ταυτότητα με το όνομα τομέα «bob.com»
  3. Το πιστοποιητικό εγκαθίσταται στον διακομιστή του Bob.com
  4. Ο φυλλομετρητής επισκέπτεται το <http://bob.com>
  5. Τα δημόσια κλειδιά της CA *ενσωματώνονται στον φυλλομετρητή*, έτσι μπορεί να ελέγξει αν ένα πιστοποιητικό ταιριάζει με ένα όνομα υπολογιστή υπηρεσίας (hostname)
  6. Η *ανταλλαγή κλειδιών Diffie-Hellman* χρησιμοποιείται για την εγκαθίδρυση ενός κρυπτογραφημένου καναλιού για περαιτέρω επικοινωνία
- Χρήση της μεθόδου `force_ssl` για την επιβολή χρήσης SSL από μερικές ή όλες τις ενέργειες

80





## Τι κάνει και δεν κάνει

- ✓ Διασφαλίζει ότι ο bob.com είναι νόμιμος
- ✓ Εμποδίζει τους υποκλοπείς να διαβάσουν (ή να αλλοιώσουν) την κυκλοφορία μεταξύ του φυλλομετρητή & του & bob.com
- ✓ Επιβαρύνει τον διακομιστή με πρόσθετη δουλειά!

### ΔΕΝ:

- ✗ Διασφαλίζει στον διακομιστή ποιος είναι ο χρήστης
- ✗ Λέει τίποτα για το τι συμβαίνει στα ευαίσθητα δεδομένα αφού φτάσουν στον διακομιστή
- ✗ Λέει τίποτε σχετικά με το αν ο διακομιστής είναι ευπαθής σε άλλες επιθέσεις διακομιστή
- ✗ Προστατεύει τον φυλλομετρητή από κακόβουλο λογισμικό αν ο διακομιστής είναι κακόβουλος

81



## Έγχυση SQL

- Προβολή: `= text_field_tag 'name'`
- Εφαρμογή: `Movigoer.where("name=#{params[:name]}")`
- Ο κακόβουλος χρήστης συμπληρώνει:  
`BOB'); DROP TABLE movigoers; --`
- `SELECT * FROM movigoers WHERE (name='BOB'); DROP TABLE movigoers; --`
- Λύση: `Movigoer.where("name=?", params[:name])`


[xkcd.com/327](http://xkcd.com/327)

82



## Επίθεση με έγχυση SQL;



83



## Πλαστογραφία αίτησης μεταξύ ΙΣΤΟΤΟΠΩΝ

1. Η Alice εισέρχεται στον bank.com, τώρα έχει cookie
2. Η Alice πηγαίνει στο blog.evil.com
3. Η σελίδα περιέχει:  
``
4. Ο evil.com συλλέγει τις προσωπικές πληροφορίες της Alice

Λύσεις:

- (αδύναμη) έλεγχος του πεδίου Referer στην κεφαλίδα HTTP
- (ισχυρή) συμπερίληψη *συνεδρίας nonce* σε κάθε αίτηση
  - `csrf_meta_tags` στο `layouts/application.html.haml`
  - `protect_from_forgery` στο `ApplicationController`
  - Οι βοηθητικές συναρτήσεις φορμών του Rails περιλαμβάνουν αυτόματα τη συνεδρία nonce στις φόρμες

84

Αν ένας ιστότοπος έχει ένα έγκυρο πιστοποιητικό SSL από μια αξιόπιστη CA, ποιο από τα ακόλουθα είναι σωστό:



- i) Ο ιστότοπος πιθανόν δεν είναι κακόβουλος που «υποδύεται» τον πραγματικό ιστότοπο
- ii) Οι επιθέσεις CSRF + έγχυσης SQL είναι δυσκολότερες εναντίον του
- iii) Τα δεδομένα είναι ασφαλή όταν φτάσουν στον ιστότοπο

- ☐ (i) μόνο
- ☐ (i) & (ii) μόνο
- ☐ (ii) & (iii) μόνο
- ☐ (i), (ii) & (iii)

85

85

Σωστό ή λάθος:



Αν η Baidu χρησιμοποιούσε SSL για όλη την κυκλοφορία, θα ήταν πιο δύσκολο για το Great Cannon να εισαγάγει κακόβουλη JavaScript στις αποκρίσεις της Baidu.

- ☐ Σωστό
- ☐ Λάθος

86

86



# ΚΟΨΤΕ

87

87



Η προοπτική σχεδιασμού και  
τεκμηρίωσης στην απόδοση, τις  
κυκλοφορίες, την αξιοπιστία, και την  
ασφάλεια

(*Engineering Software as a Service §12.10*)

David Patterson

© 2013 Armando Fox & David Patterson, all rights reserved

88

88



## Ο Σ&Τ στα 4 ζητήματα;

- Αντιμετωπίζει ο σχεδιασμός και τεκμηρίωση ζητήματα απόδοσης;
- Υπάρχει κάτι ιδιαίτερο στις κυκλοφορίες λογισμικού στο μοντέλο σχεδιασμού και τεκμηρίωσης;
- Υπάρχουν πρότυπα ποιότητας στον Σ&Τ;
- Είναι οι προκλήσεις αξιοπιστίας και ασφάλειας παρόμοιες στον σχεδιασμό και τεκμηρίωση;
- Μπορεί η έλλειψη αξιοπιστίας να μειώσει την ασφάλεια;

89

89



## Σ&Τ και απόδοση

- Όπως η αξιοπιστία και η ασφάλεια, η απόδοση θεωρείται μια *μη λειτουργική* απαίτηση
  - Μπορεί να είναι μέρος των ελέγχων αποδοχής
- Οι κύκλοι ζωής σχεδιασμού και τεκμηρίωσης αγνοούν την απόδοση επειδή
  - Οι βελτιστοποιήσεις απόδοσης συχνά δικαιολογούν τις κακές πρακτικές στην τεχνολογία λογισμικού
  - Καλύπτεται σε άλλα βιβλία/μαθήματα



90

90



## Σ&Τ και διαχείριση κυκλοφορίας λογισμικού

- Ειδική περίπτωση διαχείρισης διευθέτησης
- Οι κυκλοφορίες στον Σ&Τ περιλαμβάνουν τα πάντα: κώδικα, αρχεία διευθέτησης, δεδομένα, & τεκμηρίωση
- Σχήμα αρίθμησης κυκλοφοριών Σ&Τ π.χ., έκδοση Rails 3.2.12
  - .12 είναι μικρή κυκλοφορία
  - .2 είναι μεγάλη κυκλοφορία
  - 3 τόσο μεγάλη κυκλοφορία που μπορεί να χαλάσει τα API, απαιτεί εκ νέου μεταφορά του κώδικα της εφαρμογής



91

91



## Σ&Τ και αξιοπιστία

- Αξιοπιστία μέσω πλεονασμού
  - Οδηγία: να μην υπάρχει μοναδικό σημείο αποτυχίας
- Πόσο πλεονασμό μπορεί να αντέξει οικονομικά ο πελάτης;
- *Μέσος Χρόνος Αποτυχίας (MTTF)* περιλαμβάνει λογισμικό & χειριστές καθώς και υλικό
- Μη διαθεσιμότητα  $\approx$  *Μέσος Χρόνος Επισκευής (Time To Repair/MTTR)*
  - Η βελτίωση του MTTR μπορεί να είναι ευκολότερη από ό,τι του MTTF, αλλά μπορούμε να προσπαθήσουμε να βελτιώσουμε και τα δύο



92

92



## Σ&Τ και διαδικασίες για τη βελτίωση λογισμικού

- Η παραδοχή του Σ&Τ είναι ότι μπορεί να βελτιώσει τη διαδικασία ανάπτυξης λογισμικού της εταιρείας  
=> Πιο αξιόπιστο προϊόν λογισμικού
  - Καταγράφει όλες τις πτυχές του έργου για να δει τι μπορεί να βελτιώσει
- Μια εταιρεία λαμβάνει πρότυπο ISO 9001 αν έχει
  1. Καθιερωμένη διαδικασία
  2. Μέθοδο για να δει αν ακολουθείται η διαδικασία
  3. Καταγράφει αποτελέσματα για να βελτιώσει τη διαδικασία
    - Αποδοχή για τη **διαδικασία**, όχι την ποιότητα του παραγόμενου κώδικα



93

93



## ISO 9001



94

94



## Σ&Τ και ασφάλεια

- Η αξιοπιστία βασίζεται στις πιθανότητες, αλλά η ασφάλεια πρέπει να παρέχει άμυνα απέναντι σε έξυπνους αντιπάλους
  - Η βάση δεδομένων Κοινών Ευπαθειών και Εκθέσεων (Common Vulnerabilities and Exposures) περιλαμβάνει συχνές επιθέσεις: [cvedetails.com](http://cvedetails.com)
- Μερικές τεχνικές βελτίωσης της αξιοπιστίας εμποδίζουν τις επιθέσεις:
  - Υπερχείλιση ενδιάμεσης μνήμης, αριθμητική υπερχείλιση, ανταγωνισμός δεδομένα (data races)
- Έλεγχοι διείσδυσης μέσω της ομάδας *τίγρεων* μπορούν να ελέγχουν την ασφάλεια



95

95



## 3 αρχές ασφάλειας

1. **Ελάχιστο προνόμιο:** ένας χρήστης ή ένα συστατικό στοιχείο μιας εφαρμογής δεν θα πρέπει να έχει περισσότερα προνόμια – δηλαδή, όχι περισσότερες πληροφορίες πρόσβασης και περισσότερους πόρους– από τα αναγκαία για να εκτελέσει την εργασία που του έχει ανατεθεί
  - Η αρχή «χρειάζεται να ξέρει» για διαβαθμισμένες πληροφορίες

96

96





## 3 αρχές ασφάλειας

2. **Ασφαλείς προεπιλογές**: αν δεν παρέχεται ρητή άδειας προσπέλασης ενός αντικειμένου σε έναν χρήστη ή συστατικό στοιχείο μιας εφαρμογής, δεν θα πρέπει να επιτρέπεται η προσπέλαση του αντικειμένου
  - Η άρνηση πρόσβασης πρέπει να είναι η προεπιλογή
3. Ο μηχανισμός προστασίας της **ψυχολογικής αποδεκτότητας** δεν θα πρέπει να κάνει πιο δύσκολη τη χρήση της εφαρμογής σε σχέση με την περίπτωση που δεν υπάρχει προστασία
  - Χρειάζεται να είναι εύχρηστη ώστε να τηρούνται συνεχώς οι μηχανισμοί ασφάλειας

97

97

Ποια πρόταση σχετικά με την ασφάλεια είναι ΛΑΘΟΣ;



1. Η ακατάλληλη αρχικοποίηση δεδομένων θα μπορούσε να παραβιάσει την αρχή ασφάλειας των ασφαλών προεπιλογών
2. Η απουσία ελέγχου των ορίων ενδιάμεσης μνήμης θα μπορούσε να παραβιάσει την αρχή ασφάλειας του ελάχιστου προνομίου
3. Η μη απομάκρυνση ανταγωνισμών για δεδομένα θα μπορούσε να παραβιάσει την αρχή ασφάλειας της ψυχολογικής αποδεκτότητας
4. Κανένα δεν είναι λάθος – όλα είναι σωστά

98

98



# ΤΕΛΟΣ

99

99



Πλάνες, παγίδες & τελικές  
παρατηρήσεις  
(ESaaS §12.11-12.13)

© 2013 Armando Fox & David Patterson, all rights reserved

100



## Πρώιμη βελτιστοποίηση ή χωρίς μετρήσεις

- Η ταχύτητα είναι ένα χαρακτηριστικό που αναμένουν οι χρήστες
  - 99% εκατοστημόριο (π.χ.), όχι «μέσος όρος»
- *Οριζόντια κλιμάκωση* >> απόδοση ανά μηχανήμα, αλλά υπάρχουν πολλές αιτίες πιθανών καθυστερήσεων
- Η παρακολούθηση βοηθά: μέτρησε δύο φορές, όχι μία
- Περισσότερα: [railslab.newrelic.com/scaling-rails](http://railslab.newrelic.com/scaling-rails)

101

101



## «Η δική μου είναι εφαρμογή 3 επιπέδων σε υπολογιστική νέφους, θα κλιμακωθεί»

- Η βάση δεδομένων κλιμακώνεται ιδιαίτερα δύσκολα
  - Ακόμη και αν το καταφέρεις, θα θες και πάλι να τοποθετήσεις τις «δαπανηρές» λειτουργίες εκτός του SLO
- Μια βοήθεια: κρυφή αποθήκευση σε πολλά επίπεδα
  - ολόκληρη σελίδα, τμήμα, ερώτημα
  - η *λήξη* της κρυφής μνήμης είναι διατομεακό ζήτημα
  - η υποστήριξη της Ruby-on-Rails για διατομεακά ζητήματα επιτρέπει τον δηλωτικό καθορισμό τους
- Χρησιμοποίησε PaaS για όσο μπορείς

102

102



## «Ο μικρός μου ιστότοπος δεν είναι στόχος»

- Οι χάκερ μπορεί να στοχεύουν τους χρήστες σας, όχι τα δεδομένα σας
- Όπως η απόδοση, η ασφάλεια είναι *διατομεακό ζήτημα* – δύσκολα προστίθεται εκ των υστέρων
- Χρησιμοποιήστε υπάρχουσες βέλτιστες πρακτικές και εργαλεία – είναι απίθανο να τα καταφέρετε καλύτερα δημιουργώντας τα δικά σας
- Προετοιμαστείτε για την καταστροφή: να παίρνετε συχνά εφεδρικά αντίγραφα του ιστότοπου και της βάσης δεδομένων

103

Οι χρήστες σας περιστασιακά παραπονιούνται ότι ο ιστότοπός σας είναι αργός, αλλά το New Relic αναφέρει χαμηλά επίπεδα κυκλοφορίας και μικρή χρήση της ΚΜΕ. Ποια είναι η πιθανή αιτία;



- ☐ Δεν υπάρχουν αρκετά dyno του Heroku, έτσι οι αιτήσεις συχνά τίθενται σε αναμονή
- ☐ Μερικά ερωτήματα είναι ασυνήθιστα αργά επειδή μοιράζεστε τη ΒΔ με άλλες εφαρμογές
- ☐ Μερικές προβολές χρειάζονται ασυνήθιστα πολύ χρόνο για να απεικονιστούν σε συγκεκριμένους φυλλομετρητές (εξαιτίας της JavaScript)
- ☐ Θα μπορούσε να είναι οποιοδήποτε από αυτά/Δεν υπάρχουν επαρκείς πληροφορίες

104

104



105

105



## Ποιος επιτέθηκε στο GitHub και πώς; (A CS169 Special Report)

Ορισμένες εικόνες & κείμενο © 2015  
από την NETRESEC, χρήση κατόπιν  
άδειας. Αρχική δημοσίευση:  
<http://netres.ec/?b=153DB4E>

© 2015 Armando Fox & David Patterson, all rights reserved

106



## Διαθεσιμότητα αντί αποκρισιμότητας

- Διαθεσιμότητα: πόσο πιθανό είναι ότι ένας τυχαίος χρήστης να εξυπηρετηθεί με επιτυχία;
- Χρυσό πρότυπο: το δημόσιο τηλεφωνικό σύστημα των ΗΠΑ, 99,999% χρόνος λειτουργίας («πέντε εννιάρια»)
  - Γενικός κανόνας: 5 εννιάρια ~ 5 λεπτά/χρόνο
  - Κάθε εννιάρι είναι μια τάξη μεγέθους => 4 εννιάρια ~ 50 λεπτά/χρόνο, κ.λπ.
  - Οι καλές υπηρεσίες Διαδικτύου έχουν 3-4 εννιάρια
  - Ο ιστότοπος του ACA αρχικά είχε «δύο τεσσάρια» (~44%)
- Αποκρισιμότητα: πόσος χρόνος μέχρι την απάντηση;
  - Συνήθως δεν καθορίζεται από το εύρος ζώνης!

107



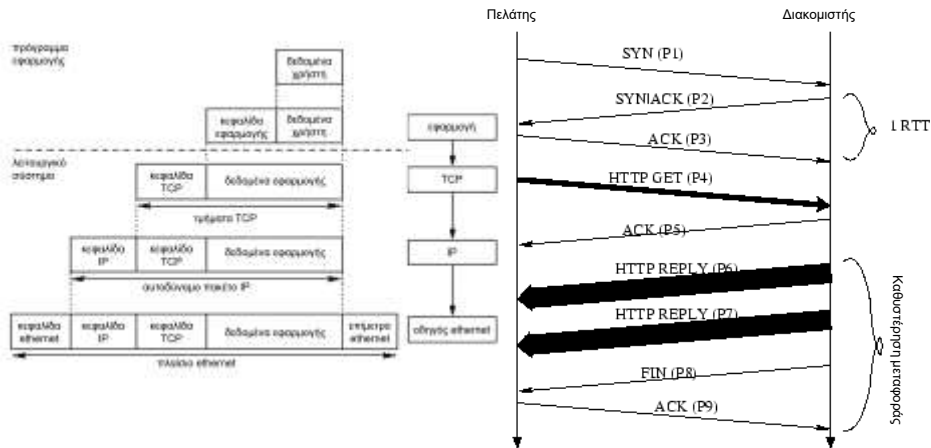
## Επιθέσεις άρνησης υπηρεσιών

- Στόχος: εμποδίζουν τους νόμιμους χρήστες να εξυπηρετούνται σωστά (να λαμβάνουν καλές υπηρεσίες)
- Συνήθως: αυξάνουν τον χρόνο απόκρισης ώστε αυτός ο ιστότοπος βασικά να γίνει άχρηστος
  - Λήξη χρόνου αναμονής → μειώνουν τη διαθεσιμότητα
  - Οι χρήστες εγκαταλείπουν → πρακτικά το ίδιο πράγμα
- *Κατανεμημένη άρνηση υπηρεσιών (Distributed denial of service, DDoS)*
  - Πολλοί χρήστες (μερικές φορές ακούσια) επισκέπτονται μαζί έναν ιστότοπο
  - Μπορεί να είναι δύσκολο να περιοριστεί ανάλογα με το πλήθος των χρηστών

108



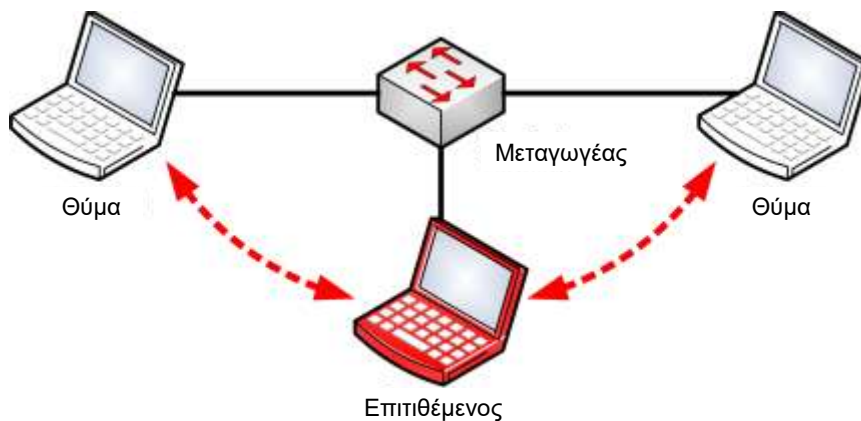
## Ανασκόπηση των TCP/IP + HTTP



109



## Επίθεση μεσολαβητή στην άκρη (Man-on-the-side, MotS)



110



## Η επίθεση

1. Ο χρήστης επισκέπτεται έναν ιστότοπο που χρησιμοποιεί το Baidu Analytics
  - Όπως το Google Analytics, ο ιστότοπος περιέχει  
`<script src="http://hm.baidu.com/h.js?0d3b1ae...">`
2. Ο S προσπαθεί να συνδεθεί με το Baidu.com για να ανακτήσει το JS
3. Η επίθεση MotS εισάγει *διαφορετική μπερδεμένη JS*
4. Η JavaScript «ξεμπερδεύεται» και εκτελείται η `eval()`
5. Ο κώδικας JavaScript προκαλεί επαναλαμβανόμενες αιτήσεις σε δύο συγκεκριμένα αποθετήρια στο GitHub
6. Η διαθεσιμότητα του GitHub σύντομα πέφτει σε λιγότερο από 2 εννιάρια

ΔΙΑΘΕΣΙΜΟΤΗΤΑ ΔΙΑΚΟΜΙΣΤΗ ΕΦΑΡΜΟΓΗΣ

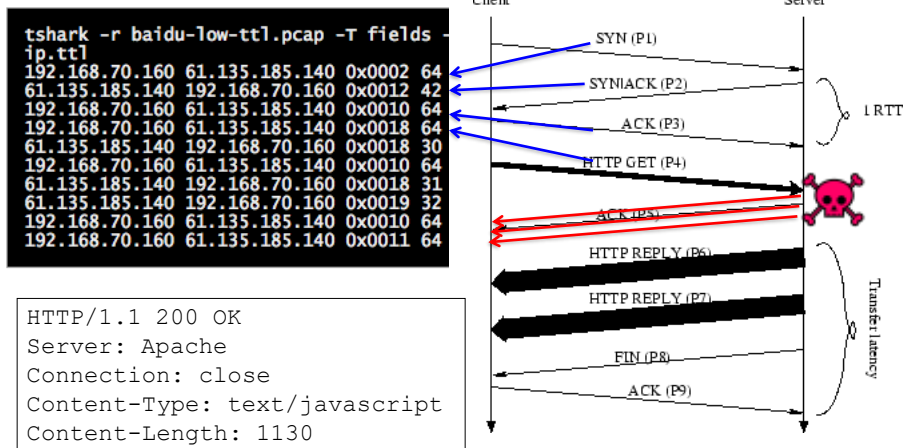
99.9816%



111



## Εισαγωγή πακέτων (packet injection)



112





## Πειστήρια για την εισαγωγή πακέτων

- Πεδίο Time-to-live (TTL) στο IP
  - κάθε οντότητα που χειρίζεται & προωθεί το πακέτο μειώνει το TTL κατά 1
  - όταν φτάσει στο μηδέν, το πακέτο απορρίπτεται
  - σκοπός είναι να εμποδίζονται οι ατέρμονες βρόχοι δρομολόγησης
- Κατά την επίθεση, τα πεδία TTL των πακέτων που εισάγονται έχουν διαφορετικές/τυχαίες τιμές, αλλά τα πεδία TTL των SYN+ACK του Baidu είναι συνεχώς 42

```
tshark -r baidu-low-ttl.pcap -T fields -e ip.src -e ip.dst -e tcp.flags -e ip.ttl
192.168.70.160 61.135.185.140 0x0002 64 <- SYN (client)
61.135.185.140 192.168.70.160 0x0012 42 <- SYN+ACK (server)
192.168.70.160 61.135.185.140 0x0010 64 <- ACK (client)
192.168.70.160 61.135.185.140 0x0018 64 <- HTTP GET (client)
61.135.185.140 192.168.70.160 0x0018 30 <- Injected packet 1 (injector)
192.168.70.160 61.135.185.140 0x0010 64
61.135.185.140 192.168.70.160 0x0018 31 <- Injected packet 2 (injector)
61.135.185.140 192.168.70.160 0x0019 32 <- Injected packet 3 (injector)
192.168.70.160 61.135.185.140 0x0010 64
192.168.70.160 61.135.185.140 0x0011 64
```

113



## Τι εισάγεται;

Τρία πακέτα IP

το #1 περιέχει κεφαλίδες HTTP:

```
HTTP/1.1 200 OK
Server: Apache
Connection: close
Content-Type: text/javascript
Content-Length: 1130
```

τα #2, #3 περιέχουν τα δύο τμήματα της μπερδεμένης JS →



114



## Συσκευαστές και απομειωτές

- Απομειωτής: μετονομάζει μεταβλητές, αποκόπτει κενά διαστήματα, κ.λπ.
  - ο παραγόμενος κώδικας παραμένει εκτελέσιμος «ως έχει»
- Συσκευαστής/περιπλέκτης: *κωδικοποιεί* το κείμενο με κάποιο τρόπο
  - Κατά τον χρόνο εκτέλεσης, ο αποκωδικοποιητής αντιστρέφει αυτή τη διαδικασία
  - Μετά καλεί την `eval()` για να εκτελέσει τον αποσυσκευασμένο κώδικα
- Εδώ χρησιμοποιείται ο συσκευαστής «Dean Edwards packer» (Google): απλός, δεν προορίζεται για χρήση ως πραγματικός περιπλέκτης
  - Σχετικά δημοφιλής μεταξύ σωστών προγραμματιστών
  - Έτσι, δεν αποκλείεται (μπλοκάρεται) από τα εργαλεία ασφάλειας

115

**Φόρτωση της jQuery αν δεν είναι ήδη φορτωμένη από αυτή τη σελίδα**

**Τα URL στόχοι του DDoS**

**Αναμονή για πάντα για επιτυχή κλήση AJAX**

**Όταν ολοκληρωθεί, αποστολή ξανά, και αυτό συνεχίζεται για 365 δευτερόλεπτα (~3 μέρες)**

```
document.write("<script  
src='http://libs.baidu.com/jquery/2.0.0/jquery.min.js'></script>");  
!window.jQuery && document.write("<script  
src='http://code.jquery.com/jquery-latest.js'></script>");  
starttime = (new Date).getTime();  
var count = 0;  
  
function unixtime() {  
    var a = new Date;  
    return Date.UTC(a.getFullYear(), a.getMonth(), a.getDay(), a.  
        getHours(), a.getMinutes(), a.getSeconds()) / 1000;  
}  
  
url_array = ["https://github.com/greentfse",  
    "https://github.com/sm-nytimes"];  
NUM = url_array.length;  
  
function r_send() {  
    var a = unixtime() * NUM;  
    get(url_array[a])  
}  
  
function get(a) {  
    var b;  
    $.ajax({  
        url: a,  
        dataType: "script",  
        timeout: 1000,  
        cache: false,  
        beforeSend: function() {  
            requestTime = (new Date).getTime()  
        },  
        complete: function() {  
            responseTime = (new Date).getTime();  
            b = Math.floor(responseTime - requestTime);  
            1000 > responseTime - starttime && (r_send(b), count += 1)  
        }  
    });  
}
```

116



## Άλλα URL των οποίων η έγκυρη JS αντικαταστάθηκε με εισαγμένο κώδικα

hm.baidu.com/h.js  
 cbjs.baidu.com/js/o.js  
 dup.baidustatic.com/tpl/wh.js  
 dup.baidustatic.com/tpl/ac.js  
 dup.baidustatic.com/painter/clb/fixed7o.js  
 dup.baidustatic.com/painter/clb/fixed7o.js  
 eclick.baidu.com/fp.htm?br= ...  
 pos.baidu.com/acom?adn= ...  
 cpro.baidu.com/cpro/ui/uijs.php?tu=...  
 pos.baidu.com/sync\_pos.htm?cpoid=...

117



## Περισσότερα πειστήρια για την εισαγωγή πακέτων

- Το ~99% των αιτήσεων τρίτων μερών προς το Baidu Analytics επιστρέφουν σωστή/έγκυρη JavaScript
  - Η επίθεση MotS βασίζεται στον χρονισμό ώστε οι επιτιθέμενοι να λαμβάνουν πρώτοι τα πακέτα σας
- Ποιος κάνει την εισαγωγή;
  - Βασίζεται στην προσπέλαση σχετικά πρωτογενών υποδομών Διαδικτύου
  - Σίγουρα προέρχεται από το «Μεγάλο Τείχος Προστασίας της Κίνας» (Great Firewall of China)
  - Η Baidu ισχυρίζεται ότι είναι αθώα και δεν έχει βρει διαρροές/περίεργη δραστηριότητα στα δικά της συστήματα

118



## Συνέντευξη τύπου, 30 Μαρτίου

Ερ: «...η αναφορά λέει ότι ένα ιστότοπος των ΗΠΑ βρέθηκε υπό επίθεση χάκερ, και η πηγή της επίθεσης προερχόταν από την Κίνα. Πώς απαντάτε;»

Α: «Είναι κάπως περίεργο το ότι κάθε φορά που ένας ιστότοπος στις ΗΠΑ ή άλλη χώρα βρίσκεται υπό επίθεση, υπάρχουν υποψίες ότι πίσω από αυτό βρίσκονται Κινέζοι χάκερ.»

«...Η Κίνα είναι ένα από τα μεγάλα θύματα κυβερνοεπιθέσεων. Έχουμε τονίσει ότι η Κίνα ελπίζει να συνεργαστεί με τη διεθνή κοινότητα προκειμένου ... να συμβάλουμε σε έναν ειρηνικό, ασφαλή, ανοιχτό και συνεργατικό κυβερνοχώρο.»

«Ευελπιστούμε ότι όλα τα μέρη μπορούν να δουλέψουν μαζί για να αντιμετωπίσουν τις επιθέσεις των χάκερ με ένα θετικό και εποικοδομητικό τρόπο.»



Εκπρόσωπος Υπουργείου Εξωτερικών Υποθέσεων  
Hua Chunying, 3/30/15

119



## Μαθήματα

- Κίνδυνοι μιας ενσωματωμένης γλώσσας, ειδικά μιας που επιτρέπει χρήση αποτίμησης (*eval*)
- Μπορείς να κάνεις πραγματική ζημιά αν έχεις πρόσβαση σε υποδομή Διαδικτύου μεταξύ τελικών σημείων
- Κρύψου από την κοινή θέα (Dean Edwards packer)
- Μην εμπιστεύεσαι κανέναν: χρησιμοποίησε κρυπτογράφηση από άκρο σε άκρο (SSL/TLS)
- Δώσε ιδιαίτερη προσοχή σε αναδυόμενα παράθυρα με προειδοποιήσεις για πιστοποιητικά
  - Το Gogo Inflight Wifi πλαστογραφεί πιστοποιητικά SSL για την Google και άλλους ιστότοπους

120



121