Web Application Hacking/Security 103

CIS 5930/4930
Offensive Computer Security
Spring 2014

Outline

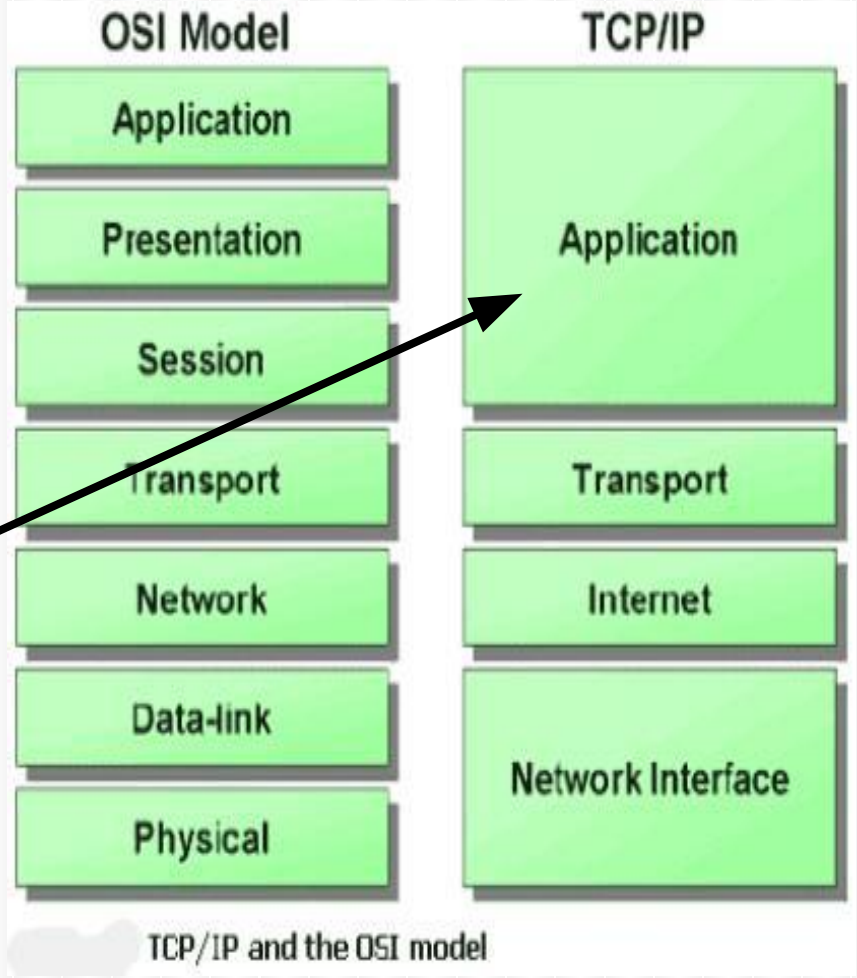
- SSL / TLS and the Certificate Authority infrastructure
  - the basics
  - the history, the story
  - the flaws
  - important CA attacks
  - lessons learned (ignored)
- SSL / TLS attacks
  - sslstrip
  - sslsniff
  - crypto attacks
    - BEAST (Browser Exploit Against SSL/TLS)
    - CRIME

What is SSL?

Secure Sockets Layer
developed by *Netscape*

- predecessor to **TLS**

- a cryptographic protocol that provides secure communication over the internet
- Encryption @ the application layer
  - asymmetric cryptography for key exchange
  - symmetric encryption for confidentiality
  - message authentication codes for message integrity



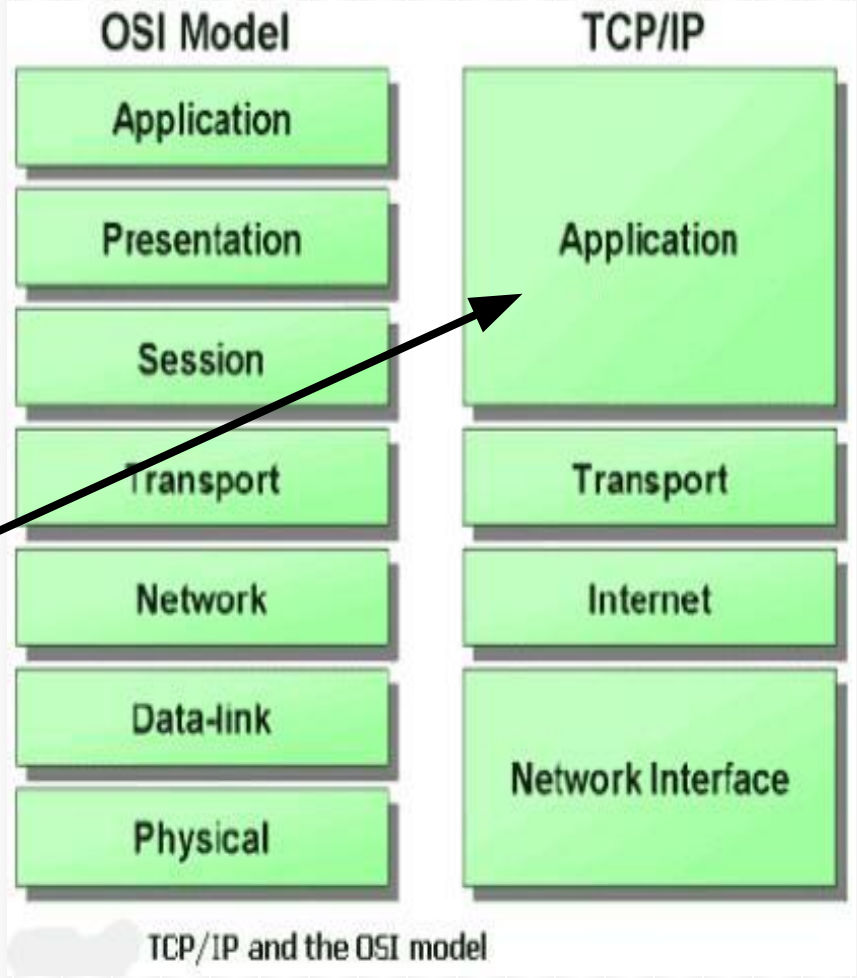
What is TLS?

Transport Layer Security

RFC 5246

- successor to **SSL**
  - but is derived from an early version of **SSL**!
- a cryptographic protocol that provides secure communication over the internet
- Encryption @ the application layer
  - asymmetric cryptography for key exchange
  - symmetric encryption for confidentiality
  - message authentication codes for message integrity

(At a high level SSL and TLS are about the same)



TCP/IP and the OSI model

SSL / TLS uses

- web browsing (HTTPS)
- email
- internet faxing (still exists???)
- instant messaging
- VOIP
- etc...

The history

- In the early 90's, at the dawn ~~of time~~ the World Wide Web
  - Engineers at Netscape developed a protocol for making secure HTTP requests
    - Very scarce body of knowledge about how to secure protocols
    - + intense pressure to get the job done
      - Deadlines man!!!
        - *4AM decisions*
    - gave us **SSL**!
      - amazing it lasted this long
- Today, the fundamental system engineered back in the 90's now faces serious problems with authenticity
  - Diminishing trust
  - Hackers are smarter
  - We now know more about how to secure things

A Secure Protocol

- Secrecy
- Integrity
- Authenticity

SSL / TLS Handshake

Client

Server

The following is sent in plaintext:

Client's SSL version #, Cipher Settings, Session-data, etc..

Server's SSL version #, Cipher Settings, Session-data, etc..
+ Server's certificate

Client uses certificate info to authenticate the server

If authentication fails, the user is warned that an encrypted and authenticated connection cannot be established

Now encrypted communication begins...

Supported by the asymmetric key encryption

Client and server establish a master secret to generate the session key

- Session key = a symmetric key used to encrypt and decrypt the session
- Details depend on cipher settings

Now encrypted communication begins with the symmetric key encryption

"Begin encrypted session", Encrypted (with symmetric key) final client handshake message

"Begin encrypted session", Encrypted (with symmetric key) final server handshake message

so HTTPS uses Certificates

Certificate Authorities (CAs) say

"This key belongs to mail.live.com"

(Browser trusts the CAs)

Handshake notes

- Implementation details can vary
- Details can vary over versions

- **Client authenticates server based off of server's certificate.....**
  - Automated to not involve the user
    - unless the certificate is of unknown origin....

The issue of "Trust"

- Trust is too hard for the normal user to think about
- Browser vendors decide the trust for you!
  - how nice of them ^-^
- Browsers ship with trusted root certificate authorities
  - Trusted root certificate authorities:
    - about 40 (chrome)
  - Intermediate certificate authorities
    - about 25 (chrome)
  - Users can add certificates

I don't even know 40 people that I trust!!

Certificates

- Certificates
  - composed of a Public key, and a Private key
- Public key certificate (aka "certificate"):
  - digitally signed statement that binds the value of a public key to the identity of the person, device, or service that holds the corresponding private key
    - usually X.509 standard certificates
      - designed in 1980
        - messy
        - overly flexible and general

Certificates

- Signing *(Digital Signatures)*
  - The private key of a certificate can be used to sign a message
    - which is then decrypted by the public key
      - nonrepudiation
      - authenticity
  - Private key can be used to sign other user's public keys
    - establish a trust relationship
      - Verisign signs the certificate for Microsoft
        - (Verisign trusts Microsoft)
    - foundation to the public key infrastructure **(PKI)**

Public Key Infrastructure

A public-key infrastructure (PKI)

- set of hardware, software, people, policies, and procedures needed for digital certificate:
  - creation,
  - management,
  - distribution,
  - use,
  - storage,
  - and revokation
    - certificate black-lists exist.
      - and they work!

Public Key Infrastructure

Certificate Authorities (CA)

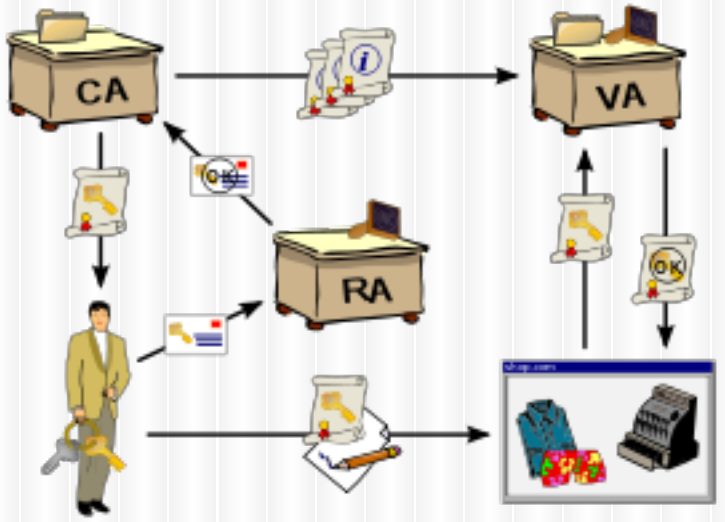
- binds the public keys with the respective user identities
  - user identities MUST be unique
- CA uses it's own private key to sign user's public key

Validation Authority (VA)

- 3rd party exists to provide and vouch for user information
- involved in *registration* and *issuance* process

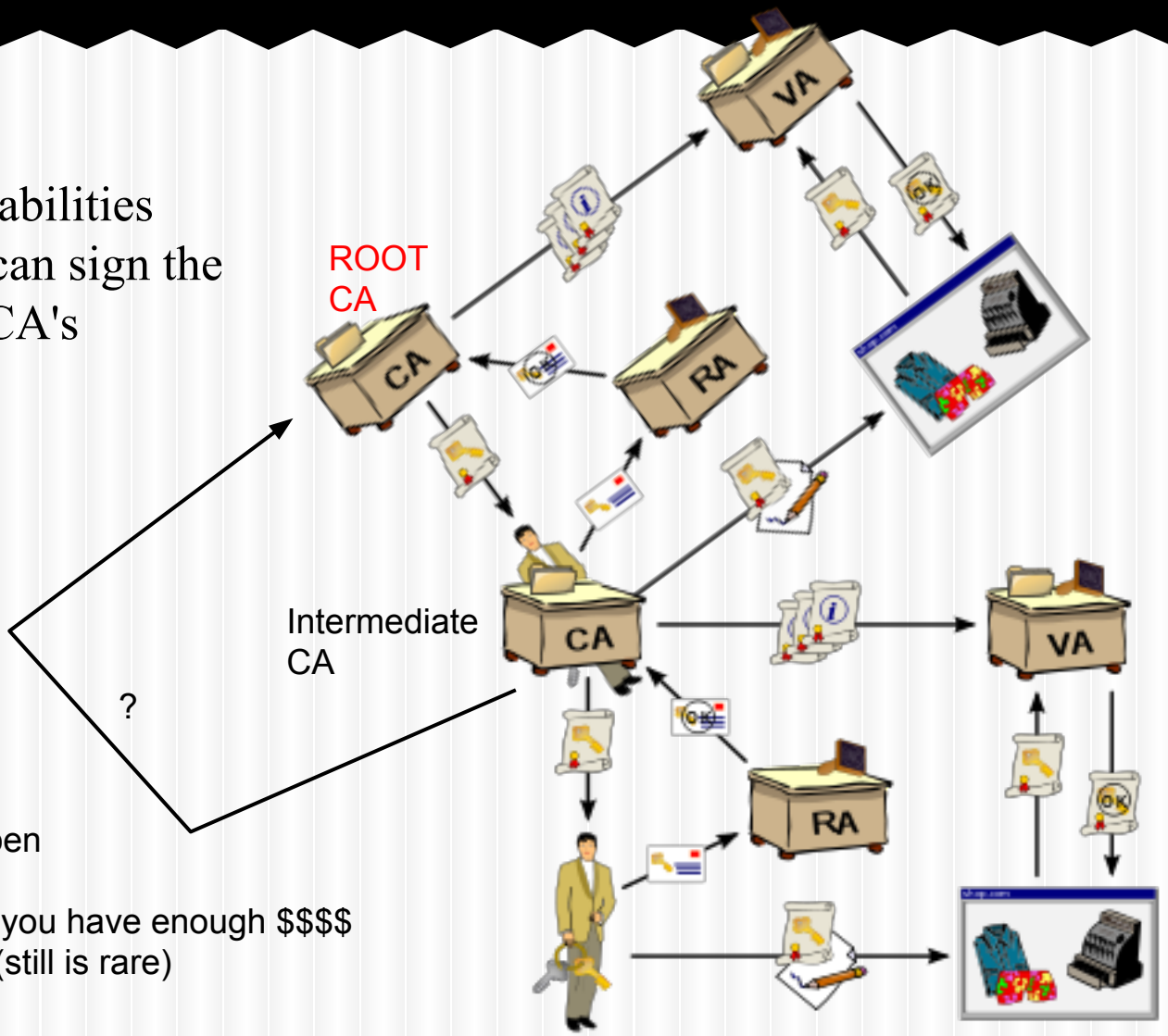
Registration Authority (RA)

- Exists to ensure that the public key is bound to the individual to which it is assigned
  - to ensure **non-repudiation**



CA chains

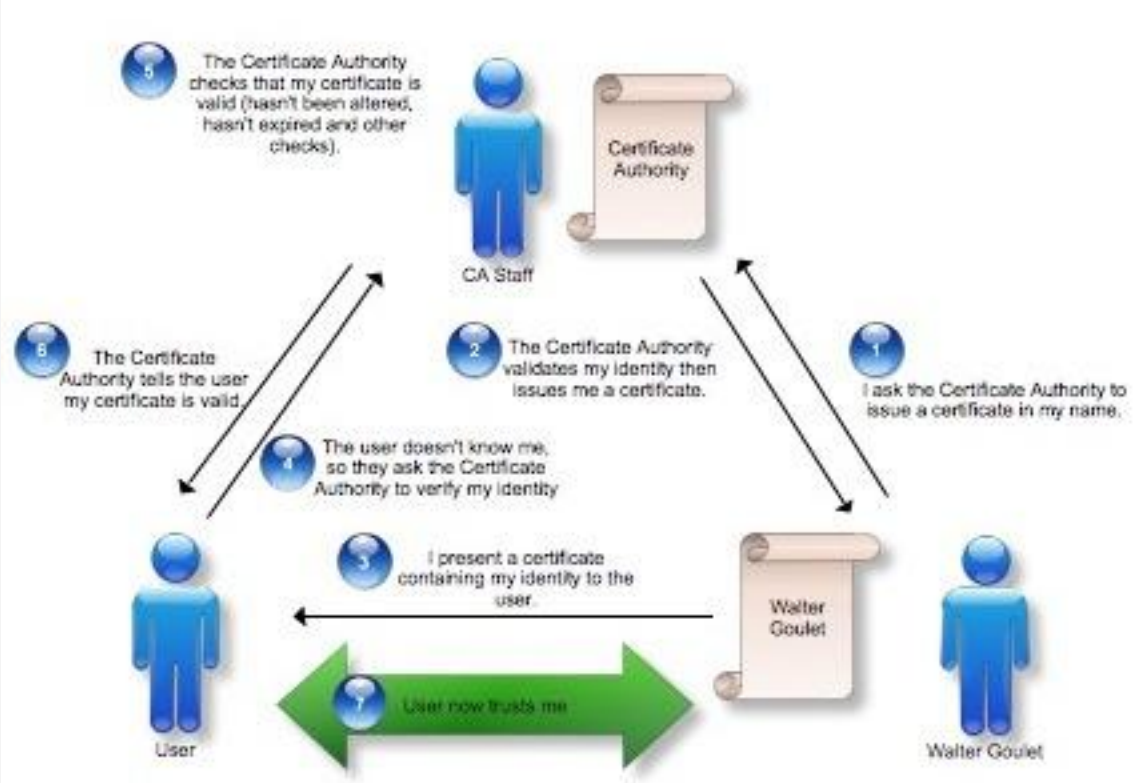
- Certificates have permissions / capabilities
  - ROOT CA's can sign the key of other CA's



Should NOT happen

But is an option if you have enough $$$$
(still is rare)

Getting a certificate for your website



img source: http://blog.securism.com/2009/01/summarizing-pki-certificate-validation/

Getting a certificate for your website

SSL Provider	Product Name	Minimum Price per Year ($)*	High/Low Assurance
COMODO CA	PositiveSSL	$49.00	Low
COMODO	Intranet SSL	$31.00	High
COMODO	InstantSSL	$64.95	High
GeoTrust	QuickSSL Premium	$118.00	Low
Thawte	SSL 123	$129.80	Low
COMODO CA	PositiveSSL Wildcard	$149.00	Low
GeoTrust	True BusinessID	$159.20	High
COMODO	InstantSSL Pro	$89.95	High
COMODO	EnterpriseSSL Elite	$179.80	High
Go Daddy ®	Standard Wildcard	$179.99	Low
Thawte	Web server cert	$219.80	High
COMODO	PremiumSSL	$116.95	High

Getting a cert means:

- forking over $
- provide identifying info about yourself
- promising to play nice
  - **<u>you are buying their trust</u>**

Verisign	Managed PKI for SSL Std	$234.00	High
COMODO	EnterpriseSSL Gold	$239.80	High
Entrust	Standard SSL Certificates	$132.00	99%
COMODO	EnterpriseSSL Platinum	$311.80	High
Verisign	Secure Site Cert	$331.67	High
GeoTrust	True BusinessID Wildcard	$399.20	High
Verisign	Managed PKI for SSL prem	$570.00	High
COMODO	PremiumSSL Wildcard	$334.95	High

Getting a certificate for your website

SSL Provider	Product Name	Minimum Price per Year ($)*	High/Low Assurance
COMODO CA	PositiveSSL	$49.00	Low
COMODO	Intranet SSL	$31.00	High
COMODO	InstantSSL	$64.95	High
GeoTrust	QuickSSL Premium	$118.00	Low
Thawte	SSL 123	$129.80	
COMODO CA	PositiveSSL Wildcard	$149.00	
GeoTrust	True BusinessID	$159.20	
COMODO	InstantSSL Pro	$89.95	
COMODO	EnterpriseSSL Elite	$179.80	
Go Daddy ®	Standard Wildcard	$179.99	
Thawte	Web server cert	$219.80	
COMODO	PremiumSSL	$116.95	

But how do website owners decide?

- They all do the same thing right?
  - why not buy the cheapest
- After all, they're all "trustworthy"!

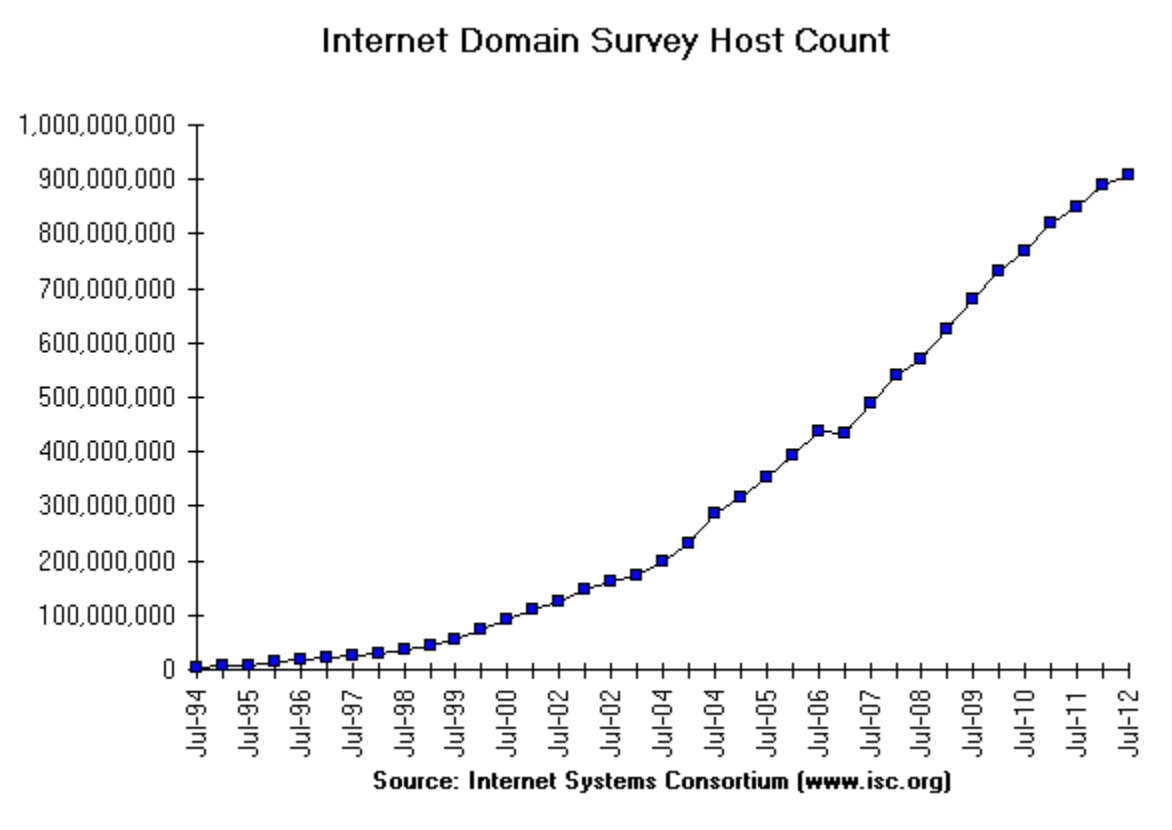
Managed PKI for SSL Std	$234.00	High
EnterpriseSSL Gold	$239.80	High
Standard SSL Certificates	$132.00	99%
EnterpriseSSL Platinum	$311.80	High
Secure Site Cert	$331.67	High
True BusinessID Wildcard	$399.20	High
Managed PKI for SSL prem	$570.00	High
PremiumSSL Wildcard	$334.95	High

Who can become a CA?

- You
- Me
- Anyone really..

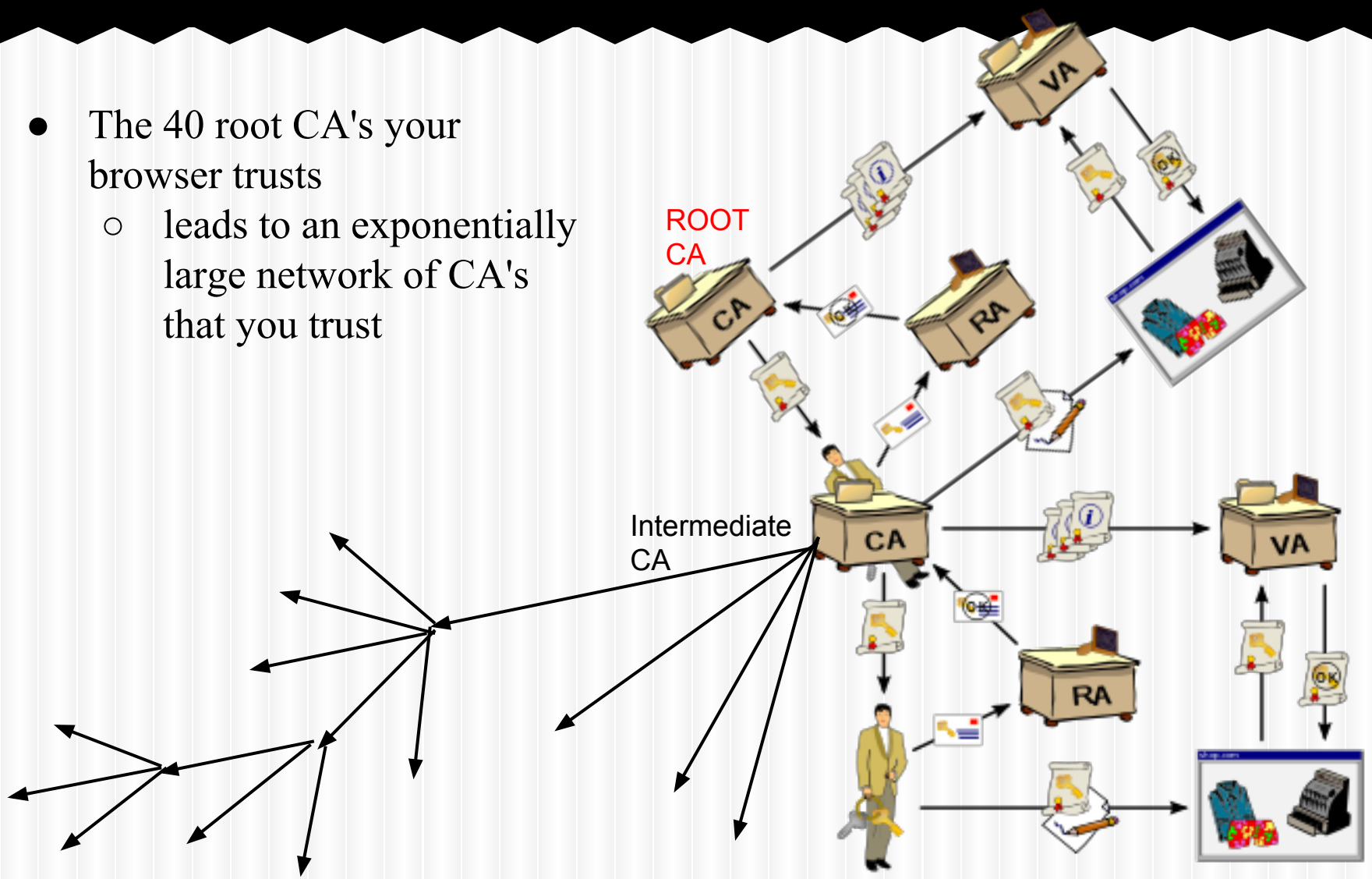
But you have to get someone to trust you

Securing the Internet



CA chains

- The 40 root CA's your browser trusts
  - leads to an exponentially large network of CA's that you trust



IN PRACTICE

Certificate Authorities (CA)

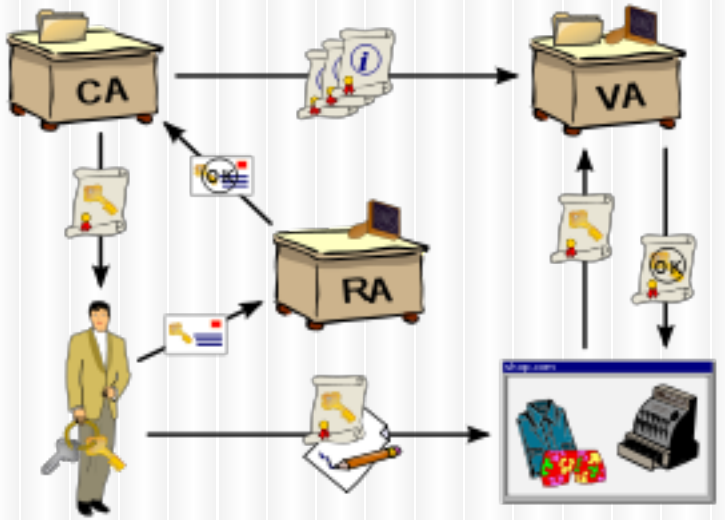
- Are the single point of failure
  - top target for hackers
- Look bad if they get hacked
  - meh, brush it under the rug!

Validation Authority (VA)

- who?
- usually don't even exist
- involved in *registration* and *issuance* process
  - issuance-smishuance
  - registration-smegistration

Registration Authority (RA)

- Probably an intern at the CA
  - supposed to be a 3rd party
- keeps an "eye out" for "bad things"



DOING EVERYTHING RIGHT COSTS $$$

- Competing with the lowest bidder
- pricing is all artificial anyways
- **ZERO** consequences when **EVERYTHING** goes wrong

CA blues

- 1980s
  - x.509 designed
    - (pro and con) flexible and general
    - ugly as hell
    - long history of implementation vulnerabilities
- 1990's
  - SSL conceived
- 2009
  - Three major vulnerabilities affected the world, just due to CA mistakes
    - whoops I published my private key in my public_html directory
- 2010
  - growing evidence of governments compelling CAs to do their bidding
    - https://www.eff.org/files/ccc2010.pdf
    - https://www.eff.org/observatory

Intermission

Let us peruse the SSL observatory

https://www.eff.org/files/colour_map_of_CAs.pdf
A map of the organizations that can function as CAs that are trusted (directly or indirectly) by Mozilla or Microsoft

(total of 650 CAs in that list)

Noteworthy CAs on that list

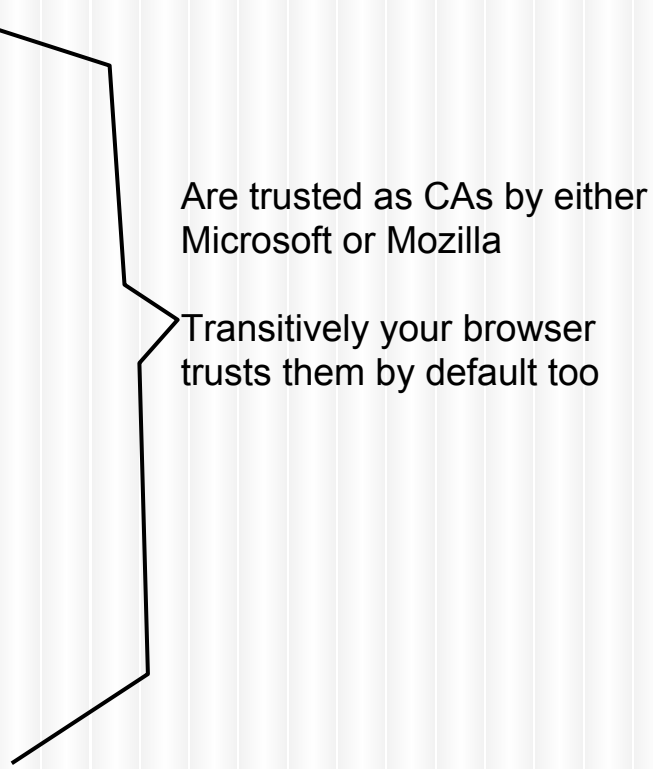
U.S. Department of Homeland Security

U.S. Defence Contractors

CNNIC, 2007
(China Internet Network Information Center)

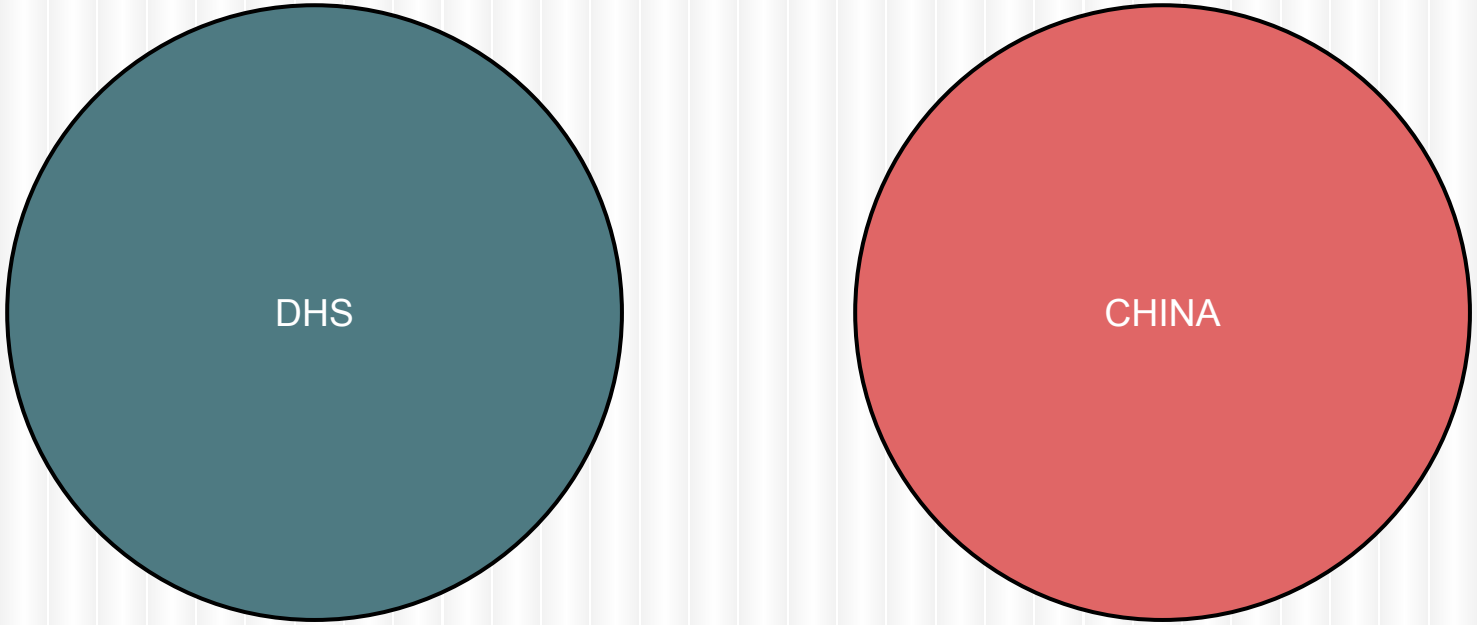
Etisalat
(Emeriates Telecommunications Corporation)

The Island of Berumuda



Scoping Issues

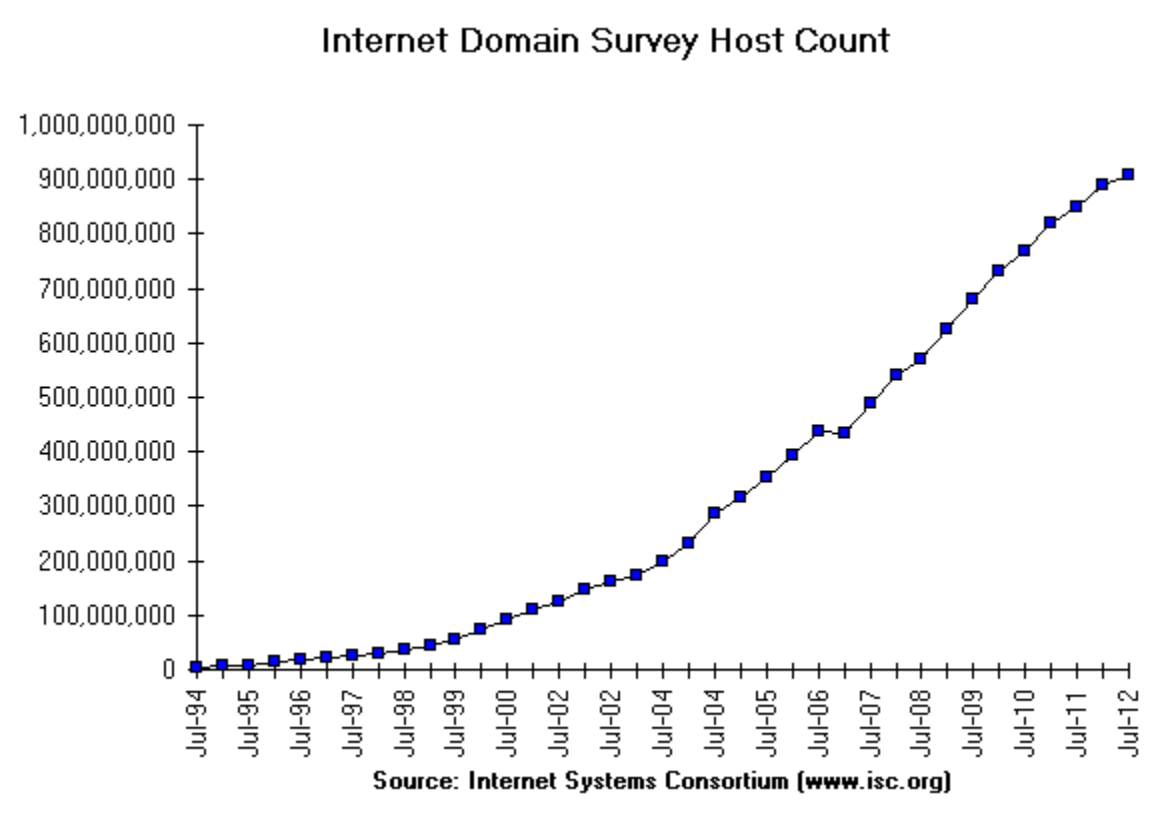
Maybe DHS should just sign for sites in the USA
and likewise Chinese state run CA's should just sign for sites in China



But naturally there are americans who would never trust DHS as a CA

and chinese citizens who would never trust any chinese CAs

Securing the Internet



Important CA "attacks"

- The following people were just able to obtain a major certificate without hacking
  - Mike Zussman obtained login.live.com
    - simply just asked for it
    - to do his security research
  - Eddy Nigg obtained mozilla.com
    - no Validation Authority (VA) stopped him
    - he was investigating unethical CA practices
      - didn't have to try very hard to hit a jackpot
    - http://www.sslshopper.com/article-ssl-certificate-for-mozilla.com-issued-without-validation.html
  - Verisign issued a code signing cert for "Microsoft Coporation" to unknown hackers
    - could sign KMDs, Windows Updates, Applications, etc..

APT

"cyber"

Important CA attacks

- RSA (2010)
  - SecureID program compromised
    - Not really SSL, but similar in concept
    - Massive hack hit 760 companies
      - google
      - facebook
      - microsoft
      - 20% of the Fortune 100 list
        - many more in the Fortune 500
    - http://money.cnn.com/2011/10/27/technology/rsa_hack_widespread/index.htm
    -

Important CA attacks

- **Comodo**
  - *hacked in **March 2011***

TECHNOLOGY | March 24, 2011

Web Firm Suspects Iran Hacked Into It

Internet-Security Company Says It Was Tricked Into Authenticating Fake Sites, Opening Access to Data, Not Money

  - Attacker IP address: 212.95.136.18
  - GPS Longitude & latitude: 35.696111 51.423056
  - attacker made off with some important certs
    - mail.google.com
    - www.google.com
    - login.yahoo.com
    - login.skype.com
    - addons.mozilla.org
    - login.live.com
    - ...

Important CA attacks

- Comodo (continued)
  - Immediately after the attack, the CEO issued the following statement:
    - "This [attack] was extremely sophisticated and critically executed… it was a very well orchestrated, very clinical attack, and the attacker knew exactly what they needed to do and how fast they had to operate"
    - Also claimed that all of the IP addrs involved in the attack were from Iran.
      - *sparked debate on* "cyber war"
        - "All of the above leads us to one conclusion only: that this was likely to be a state-driven attack."

drama ++

quotes cited from: http://privacy-pc.com/articles/ssl-and-the-future-of-authenticity-comodo-hack-and-secure-protocol-components.html

Important CA attacks

- Comodo (continued)
  - It turned out that the hacker was an amateur
    - script kiddie who talks a really BIG game
      - its hilarious
      - see his/her ramblings: http://pastebin.com/85WV10EL
  - Comodo secures 25% - 20% of all sites on the internet
    - Surely there must be consequences! A amateur breaking 25% of the internet is preposterous!
  - Got hacked 3 times later that year

And no one cared

Nothing happened to Comodo!
Comodo CEO named entrepreneur of the year at RSA 2011

Important CA attacks

- DigiNotar (2011)
  - DigiNotar issued a rouge *.google.com certificate
    - presented a number of Internet users in Iran
    - noticed by DigiNotar, and then quickly revoked
      - Hackers!
  - Important because the ENTIRE DUTCH GOVERNMENT runs off of DigiNotar certificates
    - Thus the hackers were able the hack the ENTIRE DUTCH GOVERNMENT (in a sense)
      - http://www.f-secure.com/weblog/archives/00002228.html
        - Hacker's "proof" http://pastebin.com/jhz20PqJ
        - DigiNotar got dropped by most browser vendors

Important CA attacks

- DigiNotar (continued)
  - Dutch government took over the company afterwards
    - That same month, the company was declared bankrupt
      - http://en.wikipedia.org/wiki/DigiNotar
      - *Lesson learned:* you are not too big to fail, if you cause someone bigger than you to fail, when you fail
      - Total Compromise of CA servers: http://threatpost.com/en_us/blogs/final-report-diginotar-hack-shows-total-compromise-ca-servers-103112
- StartCom (2011)
  - Israeli CA
  - simply just rumors of a breach
- GlobalSign (2011)
  - possibly compromised by same hacker that got DigiNotar
    - reports concluded there was no evidence of any breach

It is important to note how rumors of breaches affect "trust"

Important CA attacks

- Verisign repeatedly hacked (2010-2011)
  - revealed hacks in a quarterly SEC (Securities and Exchange Commission) filing in October 2011
    - only because new SEC guidelines required reporting security breaches to investors
      - these new guidelines have resulted in an EXPLOSION of reports/filings disclosing security breaches and breach risks http://blogs.reuters.com/financial-regulatory-forum/2012/04/06/disclosures-2012-level-of-cyber-security-risk-disclosures-varies-after-new-sec-guidance/
  - secures over 50% of the internet
    - .com .net .gov
      - allows hackers to impersonate any company on the net
  - not sure as to the extent of the hack
    - Verisign's DNS system processes 50+ billion queries per day
    - report implies APT / nation-state attack
    - SSL hit? we don't know...
      - Verisign sold its SSL business to Symantec, summer 2010

A Secure Protocol.....

- Secrecy
  - everyone's a CA now a days
- Integrity
  - no accountability for CAs
- Authenticity
  - CAs can give your cert away to others by accident

"the security of HTTPS is only as strong as the practices of the least trustworthy/competent CA,"

EFF SSL Observatory
https://www.eff.org/observatory

Countries than can intercept secure communications (SSL)

[source eff ssl observatory]

The flaw

- **We are locked into these trust relationships**
  - Market forces reward the cheapest "Trust" vendors
    - only natural to see so many hacks
      - imagine the ones that don't get detected / reported
  - The browser vendors could have dropped Comodo after their ridiculous hacks
    - Would have broken 25% of the internet
      - Convenience won out, and we still trust Comodo to this day
- **There is no "agility" in the current model**
  - we cannot adapt to disturbances in trust
    - trust becomes forever

Defending against the Broken CA system

skim: http://csrc.nist.gov/publications/nistbul/july-2012_itl-bulletin.pdf

in summary:
...Its complicated...

Alternatives to the broken CA system

- http://convergence.io/
  - Moxie Marlinspike
  - Trust Agility
  - Firefox plugin
    - its simple and it works

Attacking SSL (without targeting the CA system)

SSL / TLS Handshake

Client

Server

The following is sent in plaintext:

Client's SSL version #, Cipher Settings, Session-data, etc..

Server's SSL version #, Cipher Settings, Session-data, etc..
+ Server's certificate

Client uses certificate info to authenticate the server

If authentication fails, the user is warned that an encrypted and authenticated connection cannot be established

Now encrypted communication begins...

Supported by the asymmetric key encryption

Client and server establish a master secret to generate the session key

- Session key = a symmetric key used to encrypt and decrypt the session
- Details depend on cipher settings

Now encrypted communication begins with the symmetric key encryption

"Begin encrypted session", Encrypted (with symmetric key) final client handshake message

"Begin encrypted session", Encrypted (with symmetric key) final server handshake message

Tools for breaking SSL/TSL

sslstrip
- uses ARP poisoning to MITM attack unwitting users

sslsniff
- provided a CA cert, can decrypt all SSL/TLS traffic

BEAST
- Juliano Rizzo and Thai Duong (2011)
- attacks TLS 1.0, 1.1

CRIME
- Juliano Rizzo and Thai Duong (again 2012)
- attacks all versions of TLS/SSL
- details unknown

Or the slew of SSL bugs that have recently come out
- Linux GNUUtils
- iOS “goto fail” bug
- and the many others over history.

sslstrip

- need to be on same network as your victim
  - ARP spoof the victim
    - impersonate the gateway
      - all traffic routes through you then
- Once you intercept all traffic:
  - replace all GET HTTPS / POST HTTPS with
    - GET HTTP
    - POST HTTP
      - simply just replace HTTPS with HTTP

SSLSTRIP tutorial

View: http://www.youtube.com/watch?v=Q1hnHbBb_bA

defeating sslstrip

- Have your website run **strict transport security**
  - ssl on all pages
    - MANY websites do not do this
      - **due to incompentence**
        - there is NO other excuse now a days
  - A HTTP request to your server will always respond with a HTTPS response.
    - no more downgrading from HTTPS to HTTP
    - plain HTTP not permitted

sslsniff

- requires being able to monitor the communication of your intended victim
- requires also having the certificates to decrypt the SSL/TLS traffic
  - Bad guys do this after hacking a CA
- cannot be defended against

BEAST

- Crypto attack tools
  - can strip HTTPS cookies from a session
    - < 10 minutes
  - exploits vulnerabilities in TLS 1.0, 1.1
- Defense is to use latest TLS / SSL versions

Illustrated Guide to BEAST

http://commandlinefanatic.com/cgi-bin/showarticle.cgi?article=art027

CRIME

- Similar to BEAST
  - details not fully disclosed yet
  - supposedly affects all versions of TLS / SSL

Required Reading:

1. *"SSL and the future of Authenticity": https://www.youtube.com/watch?v=Z7Wl2FW2TcA*
2. *Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL http://files.cloudprivacy.net/ssl-mitm.pdf*
3. *Read Chapter 10 in WAHH*

The end

questions?