Securing Internet Communication

CS 161: Computer Security

Prof. Vern Paxson

TAs: Devdatta Akhawae, Mobin Javed & Matthias Vallentin

http://inst.eecs.berkeley.edu/~cs161/

March 31, 2011
Today’s Lecture

• Applying crypto technology in practice
• Goal #1: overview of the most prominent Internet security protocols
  – **SSL/TLS**: transport-level (process-to-process) on top of TCP
  – *(DNSSEC*: securing domain name lookups)*
  – Issues that arise in securing these
• Goal #2: cement understanding of crypto building blocks & how they’re used together
Building Secure End-to-End Channels

- *End-to-end* = communication protections achieved all the way from originating client to intended server
  - With no need to trust intermediaries
- Dealing with threats:
  - Eavesdropping?
    - Encryption (including session keys)
  - Manipulation (injection, MITM)?
    - Integrity (use of a MAC); *replay protection*
  - Impersonation?
    - Signatures

(What’s missing? Availability …)
Building A Secure End-to-End Channel: SSL/TLS

- SSL = *Secure Sockets Layer* (predecessor)
- TLS = *Transport Layer Security* (standard)
  - Both terms used interchangeably
- Notion: provide means to secure *any* application that uses TCP
SSL/TLS In Network Layering

7 Application
4 Transport
3 (Inter)Network
2 Link
1 Physical

7 Application
4 SSL / TLS
3 Transport (TCP)
2 (Inter)Network
1 Link

1 Physical
Building A Secure End-to-End Channel: SSL/TLS

- SSL = Secure Sockets Layer (predecessor)
- TLS = Transport Layer Security (standard)
  - Both terms used interchangeably
- Notion: provide means to secure *any* application that uses TCP
  - Secure = encryption/confidentiality + integrity + authentication (of server, but *not* of client)
  - E.g., puts the ‘s’ in “https”
Regular web surfing - http://URL

But if we click here ...
Web surfing with TLS/SSL - https: URL

Note: all of these images, etc., are now also fetched via https: URLs.

Doing so gives the web page full integrity, in keeping with end-to-end security.
Building A Secure End-to-End Channel: SSL / TLS

- SSL = Secure Sockets Layer (predecessor)
- TLS = Transport Layer Security (standard)
  - Both terms used interchangeably
- Notion: provide means to secure any application that uses TCP
  - Secure = encryption/confidentiality + integrity + authentication (of server, but not of client)
  - E.g., puts the ‘s’ in “https”
- API similar to “socket” interface used for regular network programming
  - Fairly easy to convert an app to be secured
HTTPS Connection (SSL / TLS)

- Browser (client) connects via TCP to Amazon’s HTTPS server
- Client sends over list of crypto protocols it supports
- Server picks protocols to use for this session
- Server sends over its certificate
- (all of this is in the clear)
- **Client now validates cert**

Browser sends SYN → Amazon

Amazon responds with SYN ACK → Browser

Browser responds with ACK → Amazon

Amazon sends "Hello. I support (TLS+RSA+AES128+SHA1) or (SSL+RSA+3DES+MD5) or …"

Let’s use TLS+RSA+AES128+SHA1

Here’s my cert

~2-3 KB of data
HTTPS Connection (SSL / TLS), con’t

• For RSA, browser constructs a long (2048 bits) random string R
• Browser sends R encrypted using Amazon’s public RSA key $K_A$
• From R browser & server derive pairs of symm. cipher keys $(C_B, C_S)$ and MAC integrity keys $(I_B, I_S)$
  – One pair to use in each direction
• Browser & server exchange MACs computed over entire dialog so far
• If good MAC, Browser displays 🪜
• All subsequent communication encrypted w/ symmetric cipher (e.g., AES128) cipher keys, MACs
  – Messages also numbered to thwart replay attacks

Browser

Amazon

Here’s my cert

~2–3 KB of data

$R$

$\{R\}_{K_A}$

$\text{MAC}(\text{dialog, } I_B)$

$\text{MAC}(\text{dialog, } I_S)$

$\{M_1, \text{MAC}(M_1, I_B)\}_{C_B}$

$\{M_2, \text{MAC}(M_2, I_S)\}_{C_S}$
Inside the Server’s Certificate

- **Domain name** associated w/ cert
  - e.g., www.amazon.com

- Amazon’s **public key** (e.g., 2048 bits for RSA)

- A bunch of auxiliary info (physical address, type of cert, expiration time)

- Name of certificate’s **issuer** (e.g., Verisign)

- Optional URL to *revocation center* to check for revoked certs

- A public-key **signature** of a hash (SHA-1) of all this
  - Constructed using the issuer’s private RSA key
  - Call this signature $S$
Validating Amazon’s Identity

• Browser compares domain name in cert w/ URL
  – Note: this provides an end-to-end property
    (as opposed to say a cert associated with an IP address)

• Browser accesses separate cert belonging to the issuer
  – These are hardwired into the browser - trusted!

• Browser applies issuer’s public key to invert signature S, obtaining hash of what issuer signed
  – Compares with its own SHA-1 hash of Amazon’s cert

• Assuming hashes match, now have high confidence it’s indeed Amazon …
  – assuming signatory is trustworthy

= assuming didn’t lose private key; assuming didn’t sign thoughtlessly
End-to-End ⇒ Powerful Protections

• Attacker runs a sniffer to capture our WiFi session?
  – (maybe by breaking crummy WEP security)
  – Encrypted communication is unreadable
    • No problem!

• DNS cache poisoning?
  – Client goes to wrong server
  – Detects impersonation
    • No problem!

• Attacker hijacks our connection, injects new traffic
  – Data receiver rejects it due to failed integrity check
    • No problem!
Powerful Protections, con’t

• DHCP spoofing?
  – Client goes to wrong server
  – Detects impersonation
    • No problem!

• Attacker manipulates routing to run us by an eavesdropper or take us to the wrong server?
  – They can’t read; we detect impersonation
    • No problem!

• Attacker slips in as a Man In The Middle?
  – They can’t read, they can’t inject
  – They can’t even replay previous encrypted traffic
  – No problem!
Validating Amazon’s Identity, con’t

• Browser retrieves cert belonging to the **issuer**
  – These are hardwired into the browser - **trusted**!

• What if browser can’t find a cert for the issuer?
This Connection is Untrusted

You have asked Firefox to connect securely to www.mikestoolbox.org, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

Get me out of here!

Technical Details

www.mikestoolbox.org uses an invalid security certificate.

The certificate is not trusted because the issuer certificate is not trusted.

(Error code: sec_error_untrusted_issuer)

I Understand the Risks

Safari can't verify the identity of the website “www.mikestoolbox.org”.

The certificate for this website was signed by an unknown certifying authority. You might be connecting to a website that is pretending to be “www.mikestoolbox.org”, which could put your confidential information at risk. Would you like to connect to the website anyway?

Show Certificate  Cancel  Continue
Validating Amazon’s Identity, con’t

• Browser retrieves cert belonging to the **issuer**
  – These are hardwired into the browser - **trusted!**

• What if browser can’t find a cert for the issuer?

• If it can’t find the cert, then warns the user that site has not been verified
  – Note, can still proceed, just **without authentication**

• Q: Which end-to-end security properties do we lose if we incorrectly trust that the site is whom we think?

• A: **All of them!**
  – Goodbye confidentiality, integrity, authentication
  – Attacker can read everything, modify, impersonate
SSL / TLS Limitations

- Properly used, SSL / TLS provides powerful end-to-end protections
- So why not use it for *everything*??
- Issues:
  - Cost of public-key crypto
    - Can buy hardware to accelerate, but $$
    - Note: *symmetric* key crypto on modern hardware is non-issue
  - Hassle of buying/maintaining certs (fairly minor)
<table>
<thead>
<tr>
<th>Vendor</th>
<th>Market Share</th>
<th>2-year Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>VeriSign/Thawte/ GeoTrust</td>
<td>59.6%</td>
<td>$695/$249/$399</td>
</tr>
<tr>
<td>Comodo</td>
<td>8.3%</td>
<td>$178</td>
</tr>
<tr>
<td>GoDaddy</td>
<td>5.3%</td>
<td>$54</td>
</tr>
<tr>
<td>DigiCert</td>
<td>2.1%</td>
<td>$256</td>
</tr>
<tr>
<td>Entrust</td>
<td>1.3%</td>
<td>$299</td>
</tr>
<tr>
<td>Network Solutions</td>
<td>1.1%</td>
<td>$229</td>
</tr>
</tbody>
</table>

(Circa April 2008)
SSL / TLS Limitations

- Properly used, SSL / TLS provides powerful end-to-end protections
- So why not use it for *everything*??

- Issues:
  - Cost of public-key crypto
    - Can buy hardware to accelerate, but $$
    - Note: *symmetric* key crypto on modern hardware is non-issue
  - Hassle of buying/maintaining certs (fairly minor)
  - DoS amplification
    - Client can force server to undertake public key operations
    - But: requires established TCP connection, and given that, there are other juicy targets like back-end databases
  - Integrating with other sites that don’t use HTTPS
  - **Latency**: extra round trips ⇒ pages take longer to load
SSL / TLS Limitations, con’t

• Problems that SSL / TLS does not take care of?
  • TCP-level denial of service
    – SYN flooding
    – RST injection
      o (but does protect against data injection!)
  • SQL injection / XSS / server-side coding/logic flaws
  • Browser coding/logic flaws
  • User flaws
    – Weak passwords
    – Phishing
  • Issues of trust …
TLS/SSL Trust Issues

- User has to make correct trust decisions …
Welcome to eBay

Ready to bid and buy? Register here

Join the millions of people who are already a part of the eBay family. Don't worry, we have room for one more.

Register as an eBay Member and enjoy privileges including:

- Bid, buy and find bargains from all over the world
- Shop with confidence with PayPal Buyer Protection
- Connect with the eBay community and more!

Sign in to your account

Back for more fun? Sign in now to buy, bid and sell, or to manage your account.

User ID: joebob
I forgot my user ID

Password: **********
I forgot my password

☐ Keep me signed in for today. Don't check this box if you're at a public or shared computer.

Sign in

Having problems with signing in? Get help.

Protect your account: Create a unique password by using a combination of letters and numbers that are not...
Welcome to eBay

Ready to bid and buy? Register here

Join the millions of people who are already a part of the eBay community.

Register as an eBay Member and enjoy privileges including:

- Bid, buy, and find bargains from all over the world
- Shop with confidence with PayPal Buyer Protection
- Connect with the eBay community and more!

Internet Explorer

When you send information to the Internet, it might be possible for others to see that information. Do you want to continue?

- In the future, do not show this message.

Yes  No

Having problems with signing in? Get help.

Protect your account: Create a unique password by using a combination of letters and numbers that are not
Internet Explorer

When you send information to the Internet, it might be possible for others to see that information. Do you want to continue?

☑️ In the future, do not show this message.

Yes  No
Information you exchange with this site cannot be viewed or changed by others. However, there is a problem with the site's security certificate.

⚠️ The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

✅ The security certificate date is valid.

✅ The security certificate has a valid name matching the name of the page you are trying to view.

Do you want to proceed?

Yes  No  View Certificate
Please confirm your identity.

Please answer security questions:

Select your secret question:

Answer the secret question you provided.

What is your other eBay user ID or another email used to be associated with this account?

Have you ever sold something on eBay?

- No
- Yes

This certificate is intended for the following purpose(s):
- Ensures the identity of a remote computer

- Refer to the certification authority's statement for details.

- Issued to: rover.ebay.com

- Issued by: VeriSign Class 3 Secure Server CA - G3

- Valid from: 10/22/2010 to 12/1/2012
Please answer security question.

Select your secret question:

Answer the secret question you provided.

What is your other eBay user ID or another contact you used to create your eBay account?

What email used to be associated with this account?

Have you ever sold something on eBay?

○ No
○ Yes
Please confirm your identity.

Please answer security questions:

- Select your secret question...
- Answer the secret question you provided.
- What is your other eBay user ID or another account?
- What email used to be associated with this account?
- Have you ever sold something on eBay?
  - No
  - Yes

Certificate:

Field | Value
--- | ---
Subject Alternative Name | DNS Name=ever.ebay.com, ... Subject Type=End Entity, Public Key
Basic Constraints | Digital Signature, Key Encryption
Key Usage | Server Authentication (1.3.6...
CRL Distribution Points | [1] CRL Distribution Point: Distrib...
Enhanced Key Usage | Server Authentication (1.3.6...
Authority Key Identifier | KeyID=6d 5c 53 31 8c 8f 18 5b...
Authority Information Access | [1] Authority Info Access: Access...

Edit Properties... Copy to File... OK
Please confirm your identity.

Please answer security questions:

Select your secret question...
Answer the secret question you provided.

What is your other eBay user ID or another email?

What email used to be associated with this account?

Have you ever sold something on eBay?
- No
- Yes

Certificate status:
This certificate is OK.
The equivalent as seen by most Internet users:

*(note: an actual Windows error message!)*
Certificate Errors

What should you do if you see a SSL certificate error?

• Continue on to the site and ignore the error?
• Forget about visiting the site?

What if you learned that 62% of SSL-enabled websites have invalid certs?
TLS/SSL Trust Issues, con’t

• “Commercial certificate authorities protect you from anyone from whom they are unwilling to take money”
  – Matt Blaze, circa 2001

• So how many CAs do we have to worry about, anyway?
### A-Trust-Qual-02

Root certificate authority  
Expires: Tuesday, December 2, 2014 3:00:00 PM PT  

<table>
<thead>
<tr>
<th>Name</th>
<th>Kind</th>
<th>Expires</th>
<th>Keychain</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-Trust-nQual-01</td>
<td>certificate</td>
<td>Nov 30, 2014 3:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>A-Trust-nQual-03</td>
<td>certificate</td>
<td>Aug 17, 2015 3:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>A-Trust-Qual-01</td>
<td>certificate</td>
<td>Nov 30, 2014 3:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>A-Trust-Qual-02</td>
<td>certificate</td>
<td>Dec 2, 2014 3:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AAA Certificate Services</td>
<td>certificate</td>
<td>Dec 31, 2028 3:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AC Raiz Certicámara S.A.</td>
<td>certificate</td>
<td>Apr 2, 2030 2:42:02 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AddTrust Class 1 CA Root</td>
<td>certificate</td>
<td>May 30, 2020 3:38:31 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AddTrust External CA Root</td>
<td>certificate</td>
<td>May 30, 2020 3:48:38 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AddTrust Public CA Root</td>
<td>certificate</td>
<td>May 30, 2020 3:41:50 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AddTrust Qualified CA Root</td>
<td>certificate</td>
<td>May 30, 2020 3:44:50 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Admin-Root-CA</td>
<td>certificate</td>
<td>Nov 9, 2021 11:51:07 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AdminCA-CD-T01</td>
<td>certificate</td>
<td>Jan 25, 2016 4:36:19 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AffirmTrust Commercial</td>
<td>certificate</td>
<td>Dec 31, 2030 6:06:06 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AffirmTrust Networking</td>
<td>certificate</td>
<td>Dec 31, 2030 6:08:24 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AffirmTrust Premium</td>
<td>certificate</td>
<td>Dec 31, 2040 6:10:36 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AffirmTrust Premium ECC</td>
<td>certificate</td>
<td>Dec 31, 2040 6:20:24 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>America Onli...ation Authority 1</td>
<td>certificate</td>
<td>Nov 19, 2037 12:43:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>America Onli...ation Authority 2</td>
<td>certificate</td>
<td>Sep 29, 2037 7:08:00 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AOL Time W...cation Authority 1</td>
<td>certificate</td>
<td>Nov 20, 2037 7:03:00 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>AOL Time W...cation Authority 2</td>
<td>certificate</td>
<td>Sep 28, 2037 4:43:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Apple Root CA</td>
<td>certificate</td>
<td>Feb 9, 2035 1:40:36 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Apple Root Certificate Authority</td>
<td>certificate</td>
<td>Feb 9, 2025 4:18:14 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Application CA G2</td>
<td>certificate</td>
<td>Mar 31, 2016 7:59:59 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>ApplicationCA</td>
<td>certificate</td>
<td>Dec 12, 2017 7:00:00 AM</td>
<td>System Roots</td>
</tr>
</tbody>
</table>

167 items
TLS/SSL Trust Issues

• “Commercial certificate authorities protect you from anyone from whom they are unwilling to take money”
  – Matt Blaze, circa 2001
• So how many CAs do we have to worry about, anyway?
• Of course, it’s not just their greed that matters …
Computerworld - A solo Iranian hacker on Saturday claimed responsibility for stealing multiple SSL certificates belonging to some of the Web's biggest sites, including Google, Microsoft, Skype and Yahoo.

Early reaction from security experts was mixed, with some believing the hacker's claim, while others were dubious.

Last week, conjecture had focused on a state-sponsored attack, perhaps funded or conducted by the Iranian government, that hacked a certificate reseller affiliated with U.S.-based Comodo.

On March 23, Comodo acknowledged the attack, saying that eight days earlier, hackers had obtained nine bogus certificates for the log-on sites of Microsoft's Hotmail, Google's Gmail, the Internet phone and chat service Skype and Yahoo Mail. A certificate for Mozilla's Firefox add-on site was also acquired.
Solo Iranian hacker takes credit for Comodo certificate attack

Security researchers split on whether 'ComodoHacker' is the real deal

By Gregg Keizer
March 27, 2011 08:39 PM ET

Where did you learn about cryptography and hacking. Are there books in Persian? English books? Or are you self-taught, learning from the Internet?

d) I'm self taught, books in Persian and English, but mostly papers in internet, short papers from experts like Bruce Schneier, RSA people (Ron, Adi and Leonard) and specially David Wagner. I learned programming in Qbasic when I was 9, I started learning cryptography when I was 13

funded or conducted by the Iranian government, that hacked a certificate reseller affiliated with U.S.-based Comodo.

On March 23, Comodo acknowledged the attack, saying that eight days earlier, hackers had obtained nine bogus certificates for the log-on sites of Microsoft's Hotmail, Google's Gmail, the Internet phone and chat service Skype and Yahoo Mail. A certificate for Mozilla's Firefox add-on site was also acquired.
TLS/SSL Trust Issues

• “Commercial certificate authorities protect you from anyone from whom they are unwilling to take money”
  – Matt Blaze, circa 2001

• So how many CAs do we have to worry about, anyway?

• Of course, it’s not just their greed that matters …

• … and it’s not just their diligence & security that matters …

  – “A decade ago, I observed that commercial certificate authorities protect you from anyone from whom they are unwilling to take money. That turns out to be wrong; they don't even do that much.” - Matt Blaze, circa 2010
Law Enforcement Appliance Subverts SSL

By Ryan Singel  March 24, 2010  1:55 pm  Categories: Surveillance, Threats

That little lock on your browser window indicating you are communicating securely with your bank or e-mail account may not always mean what you think its means.

Normally when a user visits a secure website, such as Bank of America, Gmail, PayPal or eBay, the browser examines the website’s certificate to verify its authenticity.

At a recent wiretapping convention, however, security researcher Chris Soghoian discovered that a small company was marketing internet spying boxes to the feds. The boxes were designed to intercept those communications — without breaking the encryption — by using forged security certificates, instead of the real ones that websites use to verify secure connections. To use the appliance, the government would need to acquire a forged certificate from any one of more than 100 trusted Certificate Authorities.
Security Warning: Do you trust the Russian government?

Firefox has detected that your connection to this website is probably not secure. If you are attempting to access or transmit sensitive data, you should stop this task, and try again using a different Internet connection.

Firefox has detected a potential security problem while trying to access www.bankofamerica.com, a website visited at least 131 times in the past by persons using this computer.

In these previous browsing sessions, www.bankofamerica.com provided a security certificate verified by a company in the United States.

However, this website is now presenting a different security certificate verified by a company based in Russia.

If you do not trust the government of Russia with your private data, or think it unlikely that Bank of America would obtain a security certificate from a company based there, this could be a sign that someone is attempting to intercept your secure communications.

Click here to learn more about security certificates and this potentially risky situation.

If you trust the government of Russia and companies located there to protect your privacy and security, click here to accept this new certificate and continue with your visit to the site.

Get me out of here!
<table>
<thead>
<tr>
<th>Name</th>
<th>Kind</th>
<th>Expires</th>
<th>Keychain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1 Public...ication Authority</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 1 Public...ication Authority</td>
<td>certificate</td>
<td>Aug 2, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 1 Public...on Authority – G2</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 2 Primary CA</td>
<td>certificate</td>
<td>Jul 6, 2019 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 2 Public...ication Authority</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 2 Public...ication Authority</td>
<td>certificate</td>
<td>Aug 2, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 3 Public...ication Authority</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 3 Public...ication Authority</td>
<td>certificate</td>
<td>Aug 2, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 4 Public...ication Authority</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>CNNIC ROOT</td>
<td>certificate</td>
<td>Apr 16, 2027 12:09:14 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Common Policy</td>
<td>certificate</td>
<td>Oct 15, 2027 9:08:00 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>COMODO Certification Authority</td>
<td>certificate</td>
<td>Dec 31, 2029 3:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Deutsche Telekom Root CA 2</td>
<td>certificate</td>
<td>Jul 9, 2019 4:59:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>DigiCert Assured ID Root CA</td>
<td>certificate</td>
<td>Nov 9, 2031 4:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>DigiCert Global Root CA</td>
<td>certificate</td>
<td>Nov 9, 2031 4:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>DigiCert High...ance EV Root CA</td>
<td>certificate</td>
<td>Nov 9, 2031 4:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>DigiNotar Root CA</td>
<td>certificate</td>
<td>Mar 31, 2025 11:19:21 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>DoD CLASS 3 Root CA</td>
<td>certificate</td>
<td>May 14, 2020 6:13:00 AM</td>
<td>System Roots</td>
</tr>
</tbody>
</table>
**CNNIC ROOT**

Root certificate authority

Expires: Friday, April 16, 2027 12:09:14 AM PT

- This certificate is valid

### Trust

### Details

<table>
<thead>
<tr>
<th>Subject Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>CN</td>
</tr>
<tr>
<td>Organization</td>
<td>CNNIC</td>
</tr>
<tr>
<td>Common Name</td>
<td>CNNIC ROOT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issuer Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>CN</td>
</tr>
<tr>
<td>Organization</td>
<td>CNNIC</td>
</tr>
<tr>
<td>Common Name</td>
<td>CNNIC ROOT</td>
</tr>
</tbody>
</table>

| Serial Number | 1228079105 |
| Version       | 3           |
| Signature Algorithm | SHA-1 with RSA Encryption (1 2 840 113549 1 1 5) |
| Parameters    | none        |

| Not Valid Before | Monday, April 16, 2007 12:09:14 AM PT |
CNNIC ROOT

Root certificate authority
Expires: Friday, April 16, 2027 12:09:14 AM PT

This certificate is valid

<table>
<thead>
<tr>
<th>Feature</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>When using this certificate</td>
<td>Use System Defaults</td>
</tr>
<tr>
<td>Secure Sockets Layer (SSL)</td>
<td>no value specified</td>
</tr>
<tr>
<td>Secure Mail (S/MIME)</td>
<td>no value specified</td>
</tr>
<tr>
<td>Extensible Authentication (EAP)</td>
<td>no value specified</td>
</tr>
<tr>
<td>IP Security (IPsec)</td>
<td>no value specified</td>
</tr>
<tr>
<td>iChat Security</td>
<td>no value specified</td>
</tr>
<tr>
<td>Kerberos Client</td>
<td>no value specified</td>
</tr>
<tr>
<td>Kerberos Server</td>
<td>no value specified</td>
</tr>
<tr>
<td>Code Signing</td>
<td>no value specified</td>
</tr>
<tr>
<td>( 1 2 840 113635 100 1 19 )</td>
<td>no value specified</td>
</tr>
</tbody>
</table>
CNNIC ROOT

Certificate

Root certificate authority
Expires: Friday, April 16, 2027 12:09:14 AM PT
✓ This certificate is valid

Trust

When using this certificate:
✔ Use System Defaults

Secure Sockets Layer (SSL)
Always Trust
Never Trust
no value specified

Secure Mail (S/MIME)
no value specified

Extensible Authentication (EAP)
no value specified

IP Security (IPsec)
no value specified

iChat Security
no value specified

Kerberos Client
no value specified

Kerberos Server
no value specified

Code Signing
no value specified

(1 2 840 113635 100 1 19)
no value specified
CNNIC ROOT
Root certificate authority
Expires: Friday, April 16, 2027 12:09:14 AM PT
✔ This certificate is valid

▼ Trust

When using this certificate: Never Trust

Secure Sockets Layer (SSL) Never Trust
Secure Mail (S/MIME) Never Trust
Extensible Authentication (EAP) Never Trust
IP Security (IPsec) Never Trust
iChat Security Never Trust
Kerberos Client Never Trust
Kerberos Server Never Trust
Code Signing Never Trust

(12 840 113635 100 1 19) Never Trust
## CNNIC ROOT

Root certificate authority
Expires: Friday, April 16, 2027 12:09:14 AM PT

⚠️ This certificate is marked as not trusted for all users

<table>
<thead>
<tr>
<th>Name</th>
<th>Kind</th>
<th>Expires</th>
<th>Keychain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1 Publication Authority</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 1 Publication Authority</td>
<td>certificate</td>
<td>Aug 2, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 1 Publication Authority – G2</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 2 Primary CA</td>
<td>certificate</td>
<td>Jul 6, 2019 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 2 Publication Authority</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 2 Publication Authority – G2</td>
<td>certificate</td>
<td>Aug 2, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 3 Publication Authority</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 3 Publication Authority – G2</td>
<td>certificate</td>
<td>Aug 2, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 4 Publication Authority</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Class 4 Publication Authority – G2</td>
<td>certificate</td>
<td>Aug 1, 2028 4:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>CNNIC ROOT</td>
<td>certificate</td>
<td>Apr 16, 2027 12:09:14 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Common Policy</td>
<td>certificate</td>
<td>Oct 15, 2027 9:08:00 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>COMODO Certification Authority</td>
<td>certificate</td>
<td>Dec 31, 2029 3:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Deutsche Telekom Root CA 2</td>
<td>certificate</td>
<td>Jul 9, 2019 4:59:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>DigiCert Assured ID Root CA</td>
<td>certificate</td>
<td>Nov 9, 2031 4:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>DigiCert Global Root CA</td>
<td>certificate</td>
<td>Nov 9, 2031 4:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>DigiCert High…rance EV Root CA</td>
<td>certificate</td>
<td>Nov 9, 2031 4:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>DigiNotar Root CA</td>
<td>certificate</td>
<td>Mar 31, 2025 11:19:21 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>DoD CLASS 3 Root CA</td>
<td>certificate</td>
<td>May 14, 2020 6:13:00 AM</td>
<td>System Roots</td>
</tr>
</tbody>
</table>

167 items
Securing DNS Lookups

• How can we ensure that when clients look up names with DNS, they can trust the answers they receive?

• Idea #1: do DNS lookups over TLS
  – (assuming either we run DNS over TCP, or we use “Datagram TLS”)
Securing DNS using SSL / TLS

Host at \texttt{xyz.poly.edu} wants IP address for \texttt{gaia.cs.umass.edu}

local DNS server (resolver) \texttt{dns.poly.edu}

Idea: connections \{1,8\}, \{2,3\}, \{4,5\} and \{6,7\} all run over SSL / TLS

requesting host \texttt{xyz.poly.edu}

root DNS server ('\.')

TLD DNS server ('.edu')

authoritative DNS server ('umass.edu', 'cs.umass.edu') \texttt{dns.cs.umass.edu}

\texttt{gaia.cs.umass.edu}
Securing DNS Lookups

• How can we ensure that when clients look up names with DNS, they can trust the answers they receive?

• Idea #1: do DNS lookups over TLS
  – (assuming either we run DNS over TCP, or we use “Datagram TLS”)
  – Issues?
    • Performance: DNS is very lightweight. TLS is not.
    • Caching: crucial for DNS scaling. But then how do we keep authentication assurances?