Competence Centers for
Excellent Technologies

TRANSPORT LAYER SECURITY BEYOND CRYPTO – NOTARIES AND PINNING TO THE RESCUE?

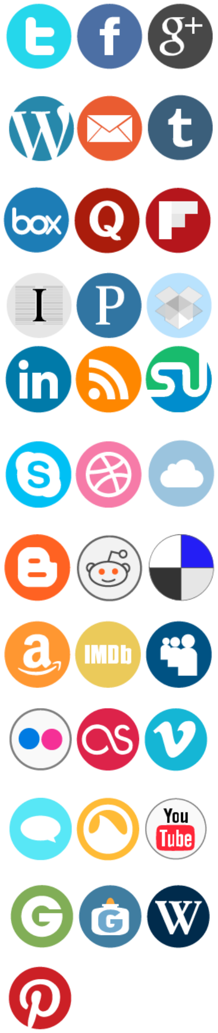
Artemios G. Voyiatzis

“If you think cryptography is the answer to your problem, then you don't know what your problem is.”

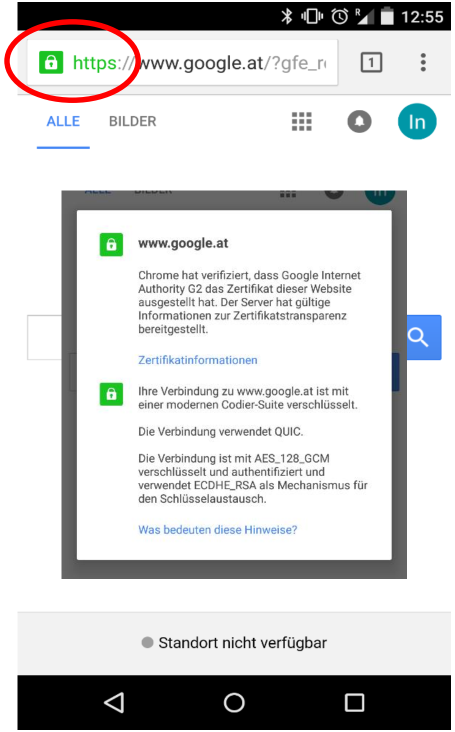
Who is the weakest link in security?

The app landscape

- >2 billion smartphones
- >2.2 million smartphone applications (apps)
- Capture and process sensitive user information
- Transfer information to/from remote sites



Is my communication secure?



www.google.at

Chrome hat verifiziert, dass Google Internet Authority G2 das Zertifikat dieser Website ausgestellt hat. Der Server hat gültige Informationen zur Zertifikatstransparenz bereitgestellt.

Zertifikatinformationen

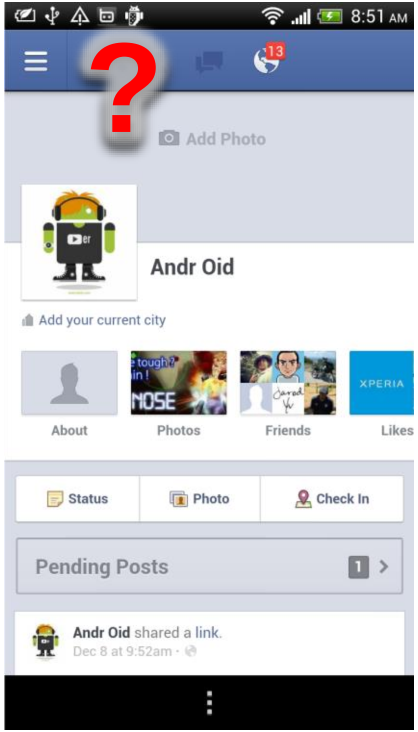
Ihre Verbindung zu www.google.at ist mit einer modernen Codier-Suite verschlüsselt.

Die Verbindung verwendet QUIC.

Die Verbindung ist mit AES_128_GCM verschlüsselt und authentifiziert und verwendet ECDHE_RSA als Mechanismus für den Schlüsselaustausch.

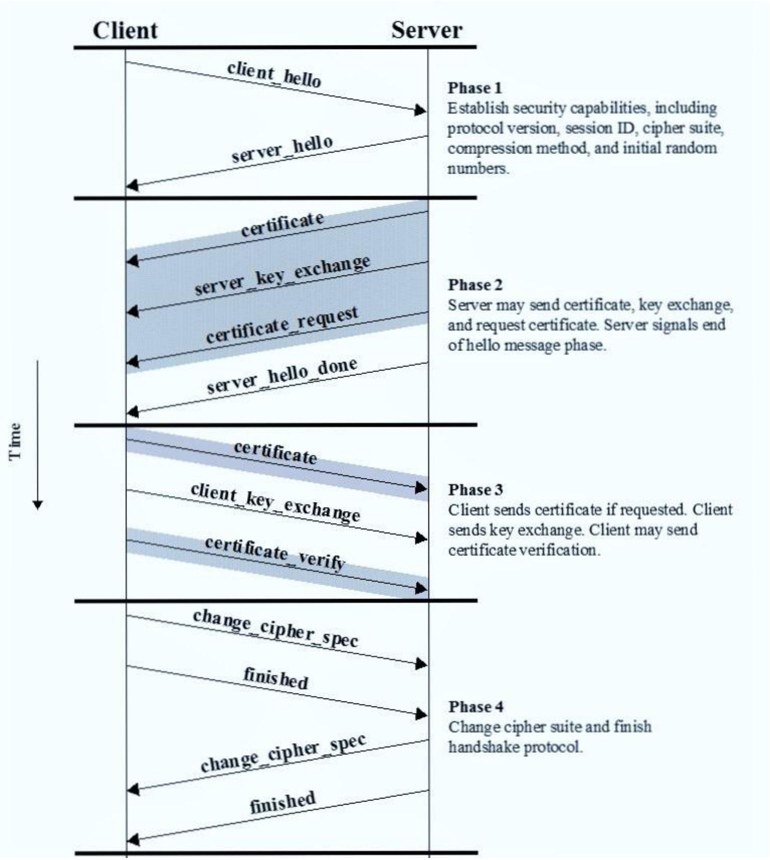
Was bedeuten diese Hinweise?

Standort nicht verfügbar



TLS connection setup

- Handshake protocol
- Four phases
- In phase 2:
  - Server sends a certificate


Client
Server
client_hello
server_hello
certificate
server_key_exchange
certificate_request
server_hello_done
certificate
client_key_exchange
certificate_verify
change_cipher_spec
finished
change_cipher_spec
finished
Time
Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.
Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.
Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.
Phase 4
Change cipher suite and finish handshake protocol.

Certificate validation before crypto

- Use the CA information of the trust store
- The client checks the validity of the server certificate
- Is the certificate authentic?
  - Is it signed by a trusted Certificate Authority?
  - Does the hostname matches the subjectAltname or CN?
  - Is it expired or still valid?
  - Is the certificate revoked?

SSL/TLS & Android apps

- Default HTTPS API in Android implements proper certificate validation

What could go wrong?

The central role of CAs

Tom Espiner

November 02, 2012

DigiNotar governm

Share this content:

Data Centre

How a cl

GlobalSi

Bug

Security

Chinese CA han

GitHub, Florida

Man-in-the-middle

29 Aug 2016 at 07:57, Darren Pauli

Security

utch

top websites' HTTPS

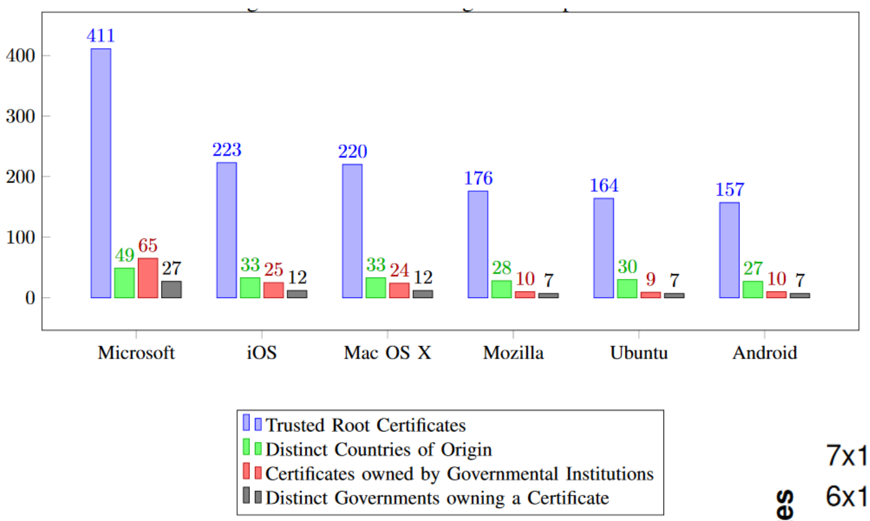
187

certificate authority have gone
bility to support usual protocols. The
small have had their HTTPS certificates
hat for some people their browsers no longer trust

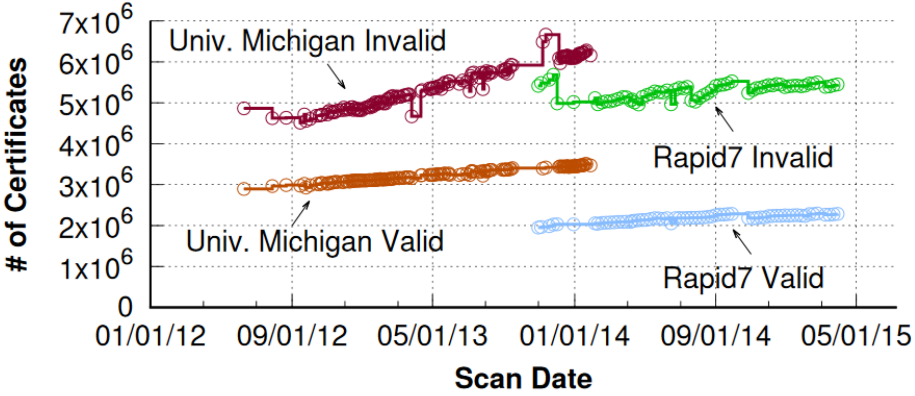
hris Williams, 18 days

26

Who signed these certificates?

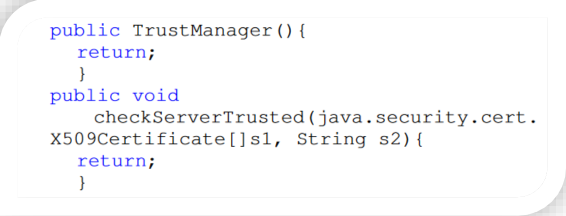


Fadai et al., Trust me, I'm a Root CA! Analyzing SSL Root CAs in Modern Browsers and Operating Systems. ARES 2015, August 24-28, 2015, France.



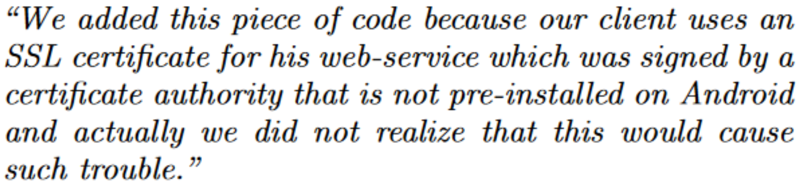
Chung et al., Measuring and Applying Invalid SSL Certificates: The Silent Majority, IMC 2016, November 14-16, 2016, USA.

Custom validation

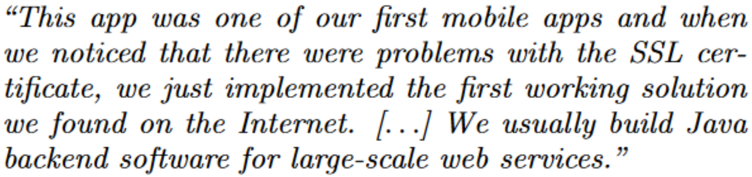


```
public TrustManager(){
   return;
   }
public void
     checkServerTrusted(java.security.cert.
X509Certificate[]s1, String s2){
   return;
   }
```

- Fahl et al. (2012): Tested 13,000 apps
  - A 1,000 of them improperly handled validation
- In 2013, they asked the developers



"We added this piece of code because our client uses an SSL certificate for his web-service which was signed by a certificate authority that is not pre-installed on Android and actually we did not realize that this would cause such trouble."



"This app was one of our first mobile apps and when we noticed that there were problems with the SSL certificate, we just implemented the first working solution we found on the Internet. [...] We usually build Java backend software for large-scale web services."

[1] S. Fahl, M. Harbach, T. Muders, L. Baumgärtner, B. Freisleben, and M. Smith, "Why Eve and Mallory love Android: An analysis of Android SSL (in)security," in ACM CCS 2012.

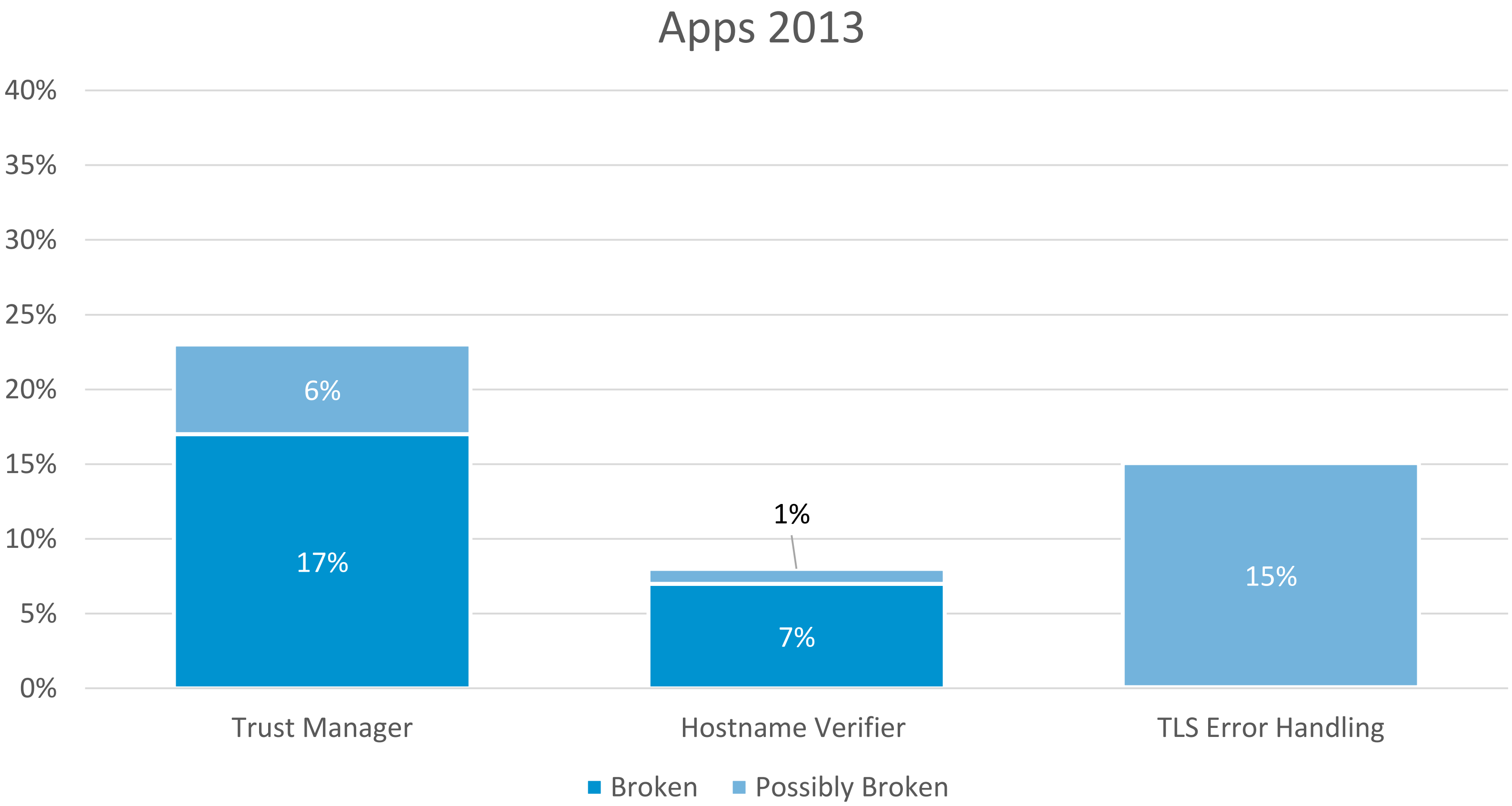
[2] S. Fahl, M. Harbach, H. Perl, M. Koetter, and M. Smith, "Rethinking SSL development in an appified world," in ACM CCS 2013.

But things improve, don’t they?

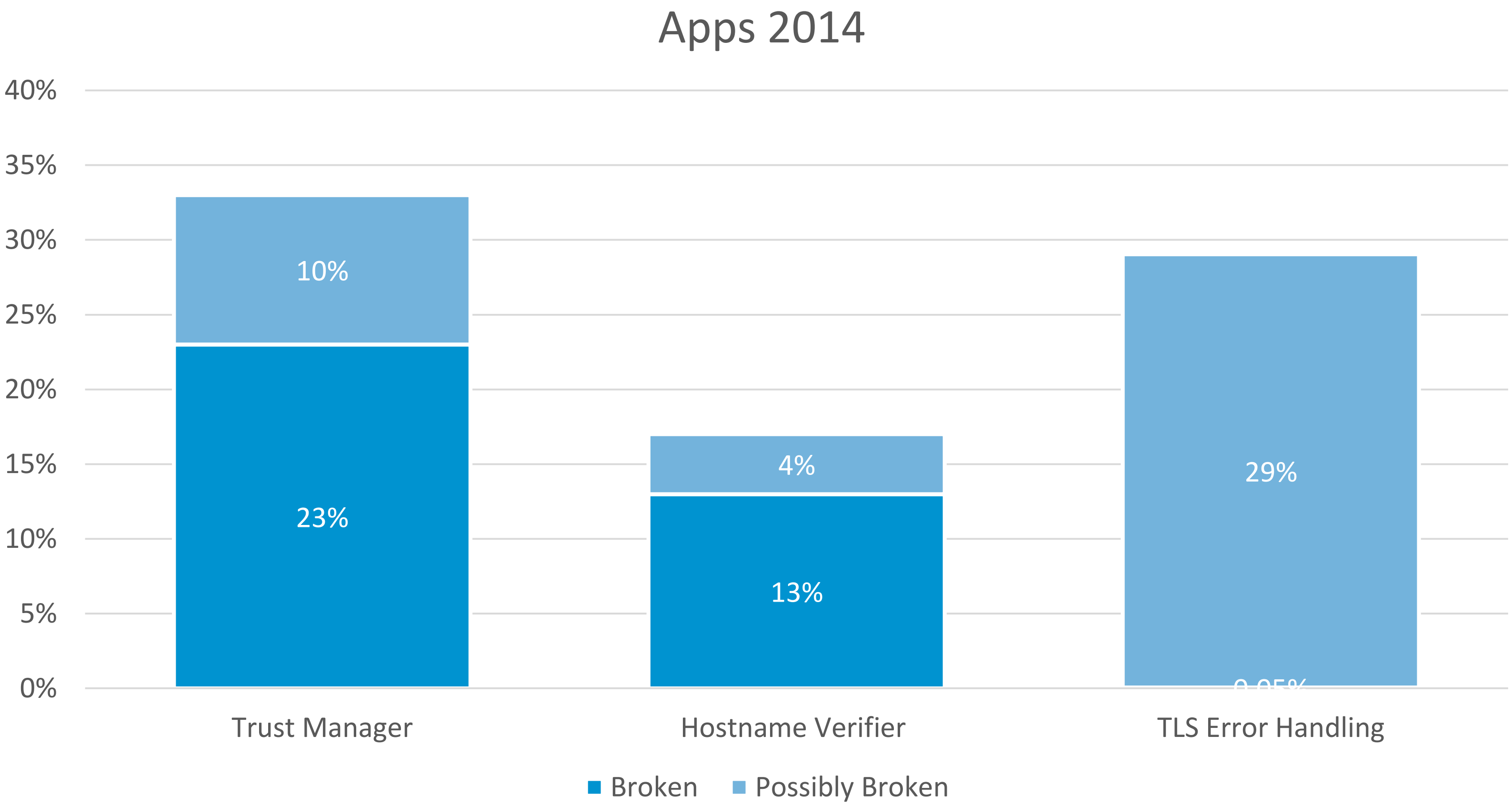
- Experiment on 50,000 Android apps[3]
  - Top 25,000 from Q4/2013
  - Top 25,000 from Q4/2014
- Test using Mallodroid script
- Focus explicitly on **custom TrustManager** implementations

[3] D. Buhov, M. Huber, G. Merzdovnik, E.R. Weippl, "Pin It! Improving Android Network Security At Runtime," in IFIP Networking 2016, Austria, 2016.

Results


Apps 2013
40%
35%
30%
25%
20%
15%
10%
5%
0%
6%
17%
1%
7%
15%
Trust Manager
Hostname Verifier
TLS Error Handling
Broken
Possibly Broken

Results


Apps 2014
40%
35%
30%
25%
20%
15%
10%
5%
0%
10%
23%
4%
13%
29%
0.05%
Trust Manager
Hostname Verifier
TLS Error Handling
Broken
Possibly Broken

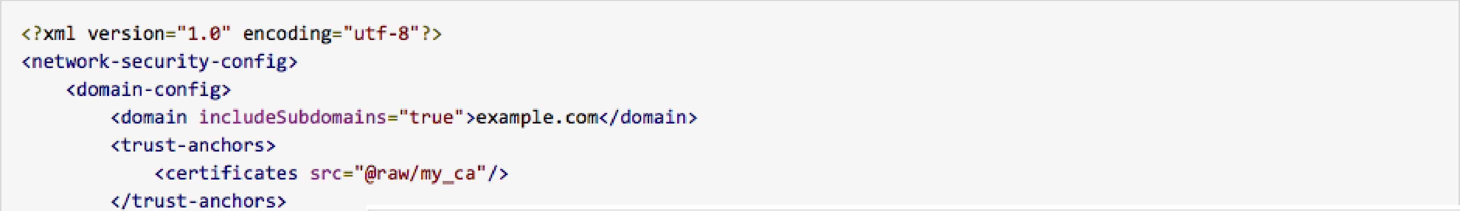
How can we fix this for user?

- PinningTrustManager PoC code on github
- Device/OS-based rather than app-based (no hope)
- Defend against developer errors in cert. handling
- Combines dynamic instrumentation techniques and cert. pinning
- User is alerted if cert. changes (e.g., injected)
  - Still chance of TOFU pinning

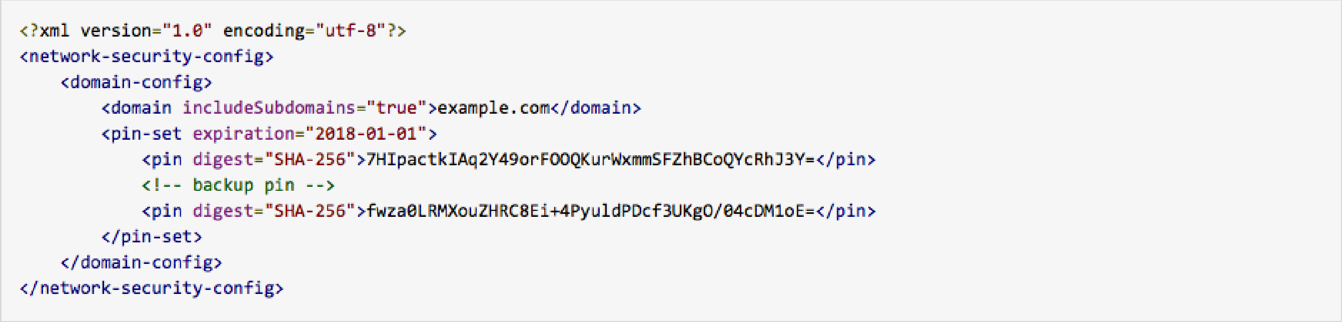
[3] D. Buhov, M. Huber, G. Merzdovnik, E.R. Weippl, "Pin It! Improving Android Network Security At Runtime," in IFIP Networking 2016, Austria, 2016.

Android 7.0 Nougat and pinning

- New approach – config file
- Much easier implementation/integration



```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config>
        <domain includeSubdomains="true">example.com</domain>
        <trust-anchors>
            <certificates src="@raw/my_ca"/>
        </trust-anchors>
    </domain-config>
</network-security-config>
```

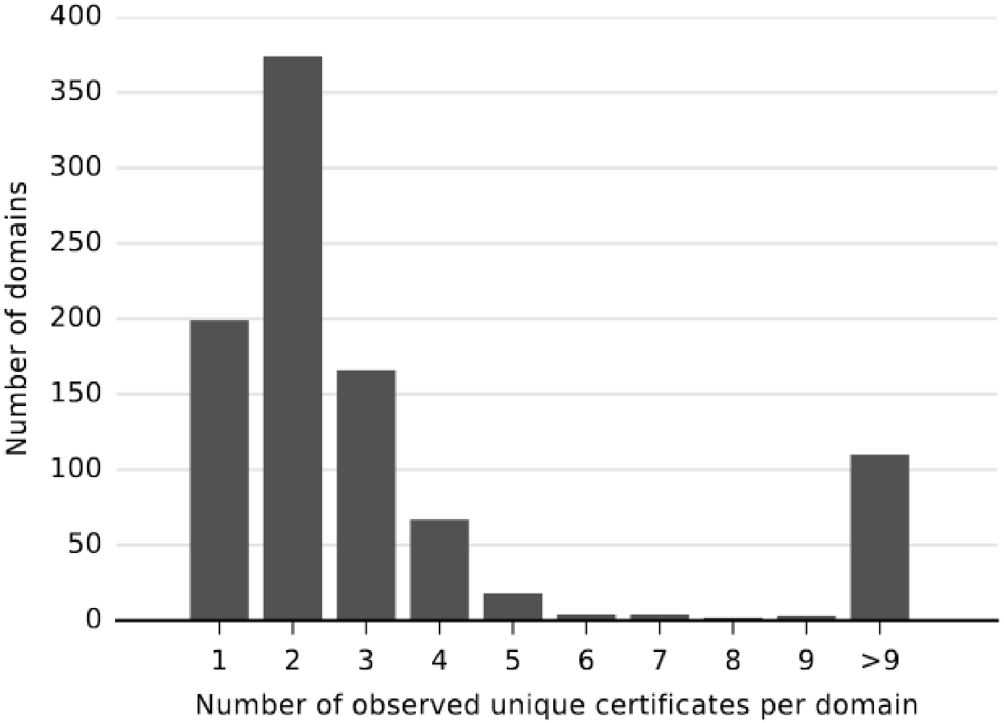


```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config>
        <domain includeSubdomains="true">example.com</domain>
        <pin-set expiration="2018-01-01">
            <pin digest="SHA-256">7HIpactkIAq2Y49orFOOQKurWxmmSFZhBCoQYcRhJ3Y=</pin>
            <!-- backup pin -->
            <pin digest="SHA-256">fwza0LRMXouZHRC8Ei+4PyuldPDcf3UKgO/04cDM1oE=</pin>
        </pin-set>
    </domain-config>
</network-security-config>
```

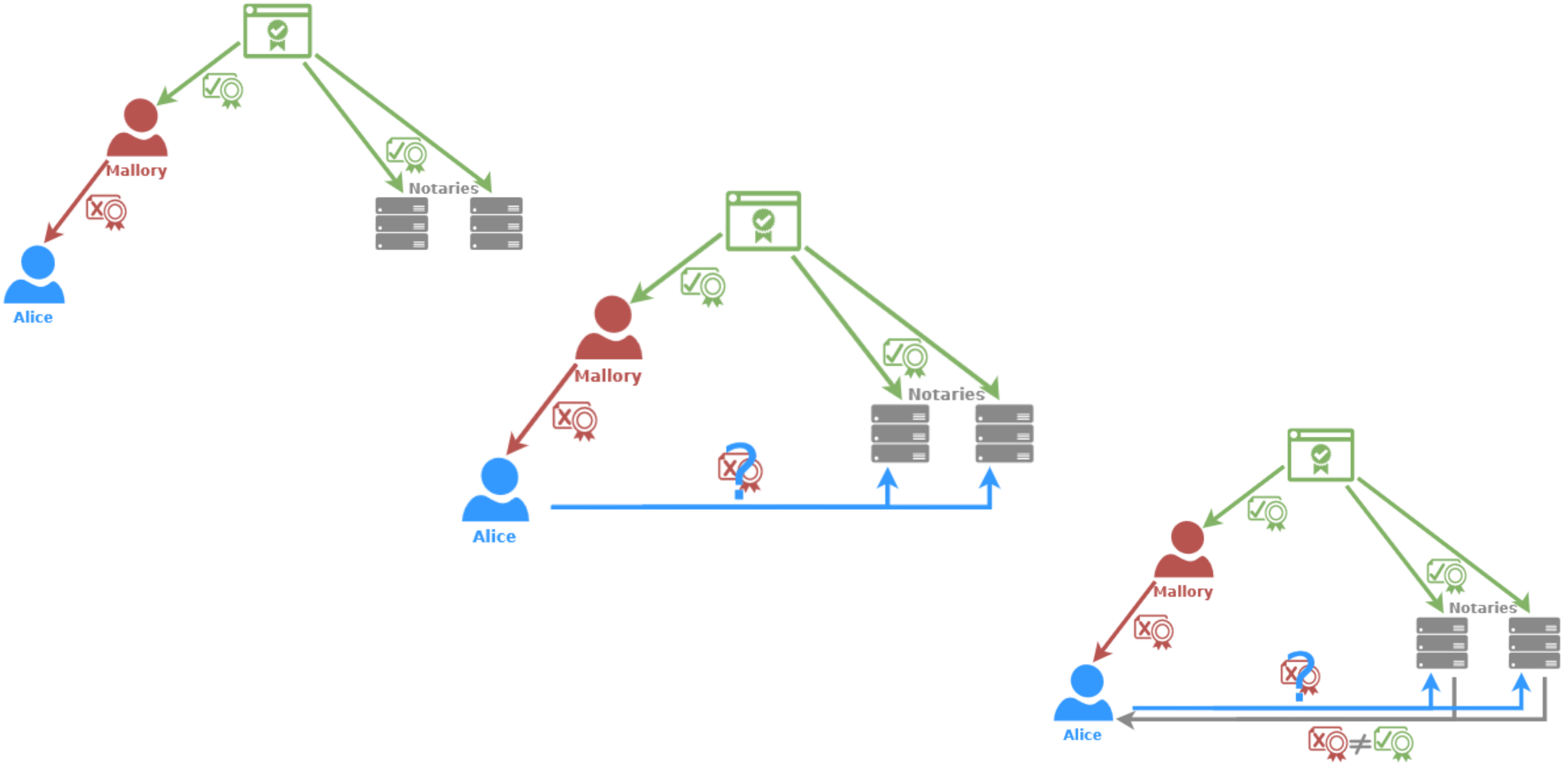
Happy developers (?)

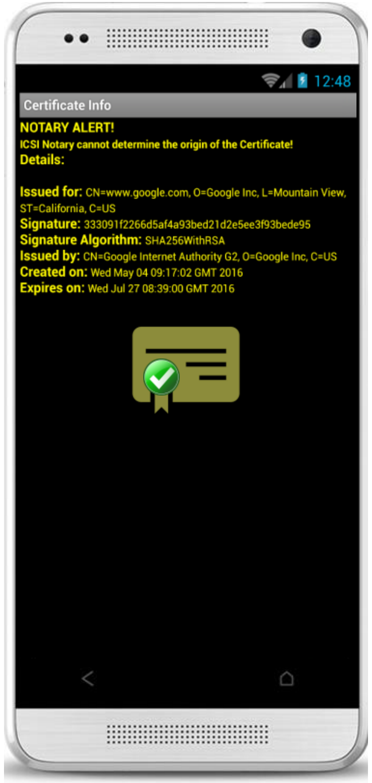
- No need for custom code
- But need to maintain two versions
- What happens when the cert. expires?
  - Recent case with Mozilla plug-ings
- How do you update apps with new files?
  - How do you force to update?

Is pinning enough?



TLS Notary Service





Certificate Info

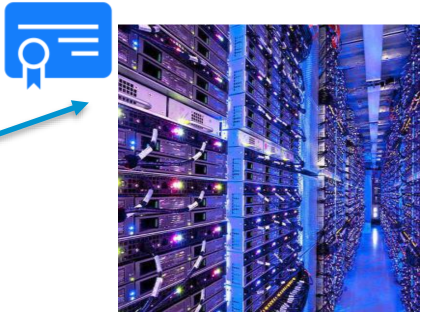
NOTARY ALERT!

ICSI Notary cannot determine the origin of the Certificate!

Details:

Issued for: CN=www.google.com, O=Google Inc, L=Mountain View, ST=California, C=US
Signature: 333091f2266d5af4a93bed21d2e5ee3f93bede95
Signature Algorithm: SHA256WithRSA
Issued by: CN=Google Internet Authority G2, O=Google Inc, C=US
Created on: Wed May 04 09:17:02 GMT 2016
Expires on: Wed Jul 27 08:39:00 GMT 2016

App/Web Server



ICSI Certificate Notary

Happy users!

- Certificate pinned on first use
  - Or even deployed with app ;)
- Feed Notary before app deployment
- No user involvement in decision
  - Only if TOFU && !Notary
- Better usability **and** better security
- PoC code also on github
  - Require rooted device (Thanks Google)
  - Would love see it integrated in next Android OS ☺

Conclusion

Protocol security

App security

User security

Credits

u'smile

Thank you!

Artemios G. Voyiatzis

avoyiatzis@sba-research.org

@a_voyages