Modern SSL Pinning

in iOS system & applications

Dima Kovalenko
Dec 2, 2017
OWASP Kyiv 2017

Agenda

1. Seasons in the sun
2. Smelly breath of SSL
3. SSL pinning versus SSLKillSwitch
4. Modern techniques to sniff/prevent sniffing SSL traffic
5. Summary

Seasons in the sun

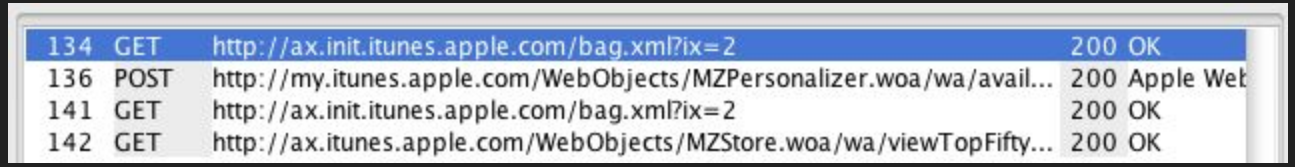
From the beginning of the iPhone era to 2010:

- HTTP everywhere
- HTTPs is a very rare beast
- Any HTTP sniffer can see applications' traffic

Life is good!

Seasons in the sun

Apple AppStore traffic in 2009



134	GET	http://ax.init.itunes.apple.com/bag.xml?ix=2	200	OK
136	POST	http://my.itunes.apple.com/WebObjects/MZPersonalizer.woa/wa/avail...	200	Apple Web
141	GET	http://ax.init.itunes.apple.com/bag.xml?ix=2	200	OK
142	GET	http://ax.itunes.apple.com/WebObjects/MZStore.woa/wa/viewTopFifty...	200	OK

Seasons in the sun

Apple AppStore traffic in 2009

```
HTTP/1.1 200 OK
Content-Length: 40469
Content-Type: text/xml; charset=UTF-8
x-apple-application-site: NWK
X-Apple-Partner: origin.0
X-NS-MZS:
x-webobjects-loadaverage: 0
x-apple-application-instance: 20208
x-apple-request-store-front: 143441-1,2
x-apple-aka-ttl: Generated Sat Jun 13 03:24:39 PDT 2009, Expires Sat Jun 13 09:04:39 PDT 2009, TT
L 20400s
Cache-Control: max-age=3935
Date: Sat, 13 Jun 2009 14:58:09 GMT
Connection: close
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/Prop
ertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>signature</key><data>etF4WWeoWi69kTjbhNssW6aaHAoMDg58DfA87Pcjk1GL5O7y+6Vt+Pi
QZMe7sA1MVw7RO0PNSq7rXsUpP1mRS0ZKlczQYR/SUxKneXU4tXfbyAg2CDVcKSjU/ANUnMFWeM698X
```

Smelly breath of SSL

Starting from 2010, more and more iOS apps use SSL. However:

- HTTP protocol is still widely used (now over SSL)
- iOS applications trust system certificate storage

It looks like SSL is used mostly to prevent MitM-attacks (stealing passwords, cookies etc) that prevent sniffing traffic from your own device.

Smelly breath of SSL

In 2010, the way to bypass SSL is simple:

1. Generate an SSL certificate
2. Add the certificate to iOS system storage
3. Use the certificate in your sniffer

Smelly breath of SSL

Numerous instructions how to do it

SSL pinning versus SSLKillSwitch

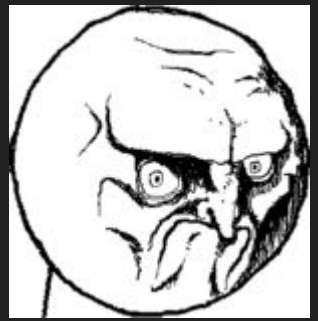
SSL certificate pinning is widely used since about 2012.

1. HTTP is still the core protocol for many iOS apps, but...
2. ...the apps do not trust system certificate storage anymore!

SSL pinning versus SSLKillSwitch

So

1. Any app has it’s own “per-app” certificate storage.
2. There is no common implementation of the “per-app” storages (iOS apps hardcode certificates, keep certificates in external files, request certificates on first start and save to app bundle settings etc).
3. There is no common way to sniff SSL traffic anymore!



SSL pinning versus SSLKillSwitch

In July 2012, Alban “nabla” Diquet saves all!

His research shows that

1. Most of iOS apps (and even iOS itself) use the same system function to check certificate
2. The functions can be hooked/patched to make any certificate valid

SSL pinning versus SSLKillSwitch

The nabla's tool, called SSLKillSwitch, is a MobileSubstrate extension.

It hooks 3 important iOS SSL stack functions:

- SSLSetSessionOption(...)
- SSLCreateContext(...)
- SSLHandshake(...)

```
// Immediately set the kSSLSessionOptionBreakOnServerAuth option in order to disable cert validation
original_SSLSetSessionOption(sslContext, kSSLSessionOptionBreakOnServerAuth, true);
```

Of course, SSLKillSwitch is not the only tool of this kind, but I believe it's first and most used.

SSL pinning versus SSLKillSwitch

<!-- DEMO1: SSLKillSwitch against YouTube -->

Modern techniques to sniff/prevent sniffing SSL traffic

In 2016, iOS app developers start to implement custom SSL validation techniques. The techniques include numerous features, e.g.

1. Pinning public keys (SubjectPublicKeyInfo (SPKI)) vs. certificate pinning
2. Client-side certificates
3. iOS SSL stack functions integrity check…
4. ...and so on

SSLKillSwitch and similar tools are not the absolute weapon against SSL pinning anymore!

Modern techniques to sniff/prevent sniffing SSL traffic

<!-- DEMO2: hook SSLRead/SSLWrite and sniff Apple Push traffic -->

Modern techniques to sniff/prevent sniffing SSL traffic

<!-- DEMO3: patch Instagram openssl-based embedded SSL framework and sniff the traffic -->

Summary

Everything is bad!

QUESTIONS?

Twitter: @kov4l3nko
Mail: kov4l3nko@gmail.com
Blog: https://kov4l3nko.github.io