TLS 1.3 for Penetration Testers

Richard J. Moore
Principal Security Consultant

Who am I?

Richard J. Moore, MWR InfoSecurity

Principal Security Consultant

- Security
- Software Engineering
- Lots of SSL/TLS

Agenda

MWR InfoSecurity

- Why TLS 1.3
- Changes in TLS 1.3 Basics
- Testing the Basics
- Changes in TLS 1.3
- Deployment
- Bleeding edge features


MWR
INFOSECURITY
Do we need TLS 1.3?

Security Whack-a-mole

MWR INFOSECURITY

- Heartbleed
- BEAST
- Padding oracle attacks
- CRIME, BREACH (compression)
- PKI compromises (Comodohacker, Lavabit)
- Protocol flaws (renegotiation)
- Poor use of Diffie-Hellman (lack of support, configuration)

Many issues were identified years ago

- Issues with TLS CBC cipher suites identified in 2004
  - Of course they weren't thought to be practical…
- Weak ciphers
  - But no one would leave these enabled in production…
- RSA key exchange
  - Allows passive eavesdropping if you know the server key

And so TLS 1.3 was born

- Encrypt as much as possible
- Address known weaknesses
- Make privacy an aim

- Removing features, not just adding
- Input from cryptographers

Changes in TLS 1.3 Basics

MWR InfoSecurity

Removal of legacy (ciphers)

MWR INFOSECURITY

- Bad or outdated ciphers
  - NULL
  - RC4, 3DES
  - Export ciphers
  - All mention of legacy RC2, single-DES etc.
- Unusual ciphers
  - ARIA, Camellia, Seed,…

what does this mean for me?

MWR INFOSECURITY

- No more “Weak ciphers” vulnerabilities ☺
  - We can finally concentrate on more interesting issues
  - And no more padding pentest reports with noise…
  - Sadly we’ll still have this for things supporting older versions too

Ciphers Suites in TLS 1.3

MWR INFOSECURITY

- Cipher suites have been redefined
  - No longer include certificate type (e.g. RSA)
  - No longer include key exchange
- Only 5 options
- More on this later

Removal of legacy (cipher modes)

- All CBC mode ciphersuites
- Always uses Authenticated Encryption with Additional Data (AEAD)

What does this mean for me?

MWR INFOSECURITY

- No more padding oracle flaws
  - POODLE
  - TLS POODLE
  - BEAST
  - Zombie POODLE
  - GOLDENDOODLE

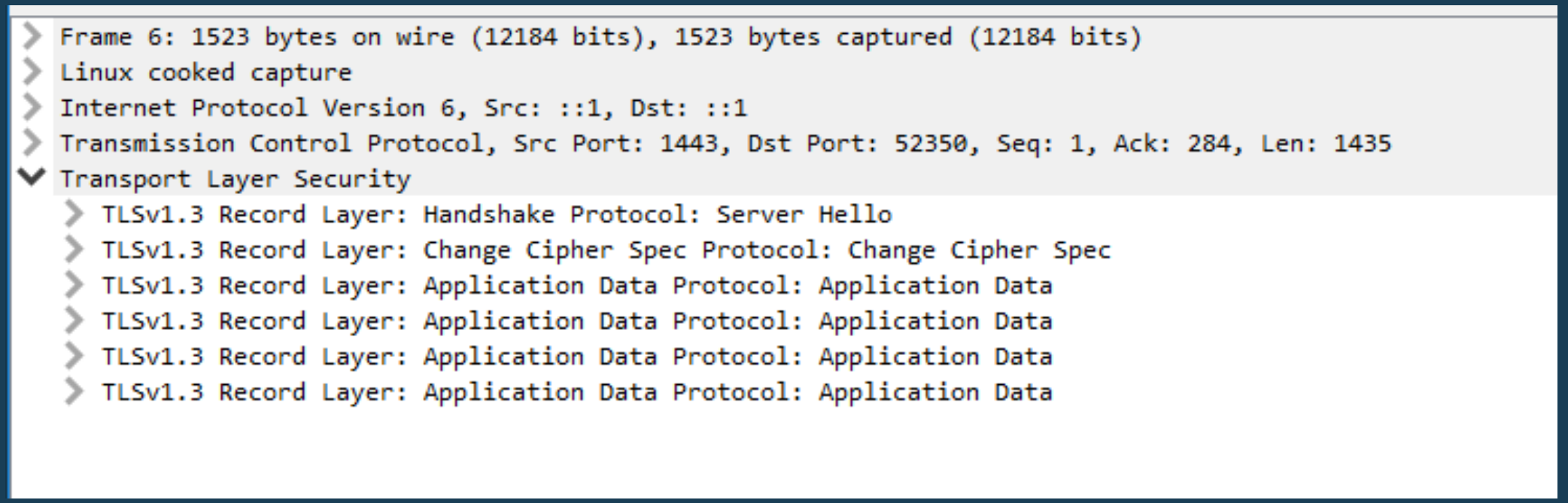
Encrypted Handshake

- Handshake now encrypts as much as possible
- Even ‘public’ information like the server certificate
- TLS Extensions

what does this mean for me?

MWR INFOSECURITY

- Most of the handshake is no longer visible in wireshark – “Application Data”



Testing TLS 1.3 Basics

MWR InfoSecurity

Demo

Tooling Support

MWR INFOSECURITY

Tool	TLS 1.3 Support
Openssl (1.1.1or newer)	Yes
sslscan	No
testssl (latest kali)	No
testssl (latest github)	Yes (buggy)
sslyze	Yes
BurpSuite Pro	Yes (2.1 only)

Openssl Examples

- TLS 1.3 only server
- TLS 1.2/1.3 server
- List supported versions of a server
- List supported ciphers TLS 1.3 of a server

Changes in TLS 1.3

MWR InfoSecurity

Removal of legacy (key exchange)

- Removal of RSA key exchange
  - Currently widely used
  - Lacks forward secrecy
- Removal of DSA certificates and key exchange
  - Meh (no one used these anyway)
- Now always uses Diffie Hellman
  - Client key is sent in ClientHello as an extension

what does this mean for me?

MWR INFOSECURITY

- Forward secrecy is always enforced
- You can't just give tools the private key and decrypt traffic
  - Wireshark can't decrypt
  - WAFs and inspecting proxies must terminate and re-encrypt
  - No longer possible to log traffic and decrypt it later

How can I decrypt traffic?

MWR INFOSECURITY

- But I need to…
  - Thick client tests
  - Useful for protocol analysis, debugging etc.
- Log the key by setting the SSLKEYLOGFILE environment variable
  - Firefox (and anything else that uses NSS)
  - Chrome
  - GnuTLS
  - Does not work for OpenSSL

How can I decrypt traffic?

MWR INFOSECURITY

- OpenSSL
  - Provides a callback that can be used to get the keys
  - Need to either use a debugger or LD_PRELOAD hooking
  - Peter Wu from Wireshark has already written a tool for this
    - https://git.lekensteyn.nl/peter/wireshark-notes/tree/src/sslkeylog.c
  - Will not work for applications that statically link ☹

Demo

Other Changes

Version Negotiation

MWR INFOSECURITY

- TLS always had a version negotiation
  - Record layer version number
  - Client hello version number
  - Server picks the result (usually the highest)
- Or that was the theory…
- Too many broken implementations
- TLS 1.3 introduces a new version negotiation extension

Early Data – 0 RTT

MWR INFOSECURITY

- TLS has allowed sessions to be stored and resumed for years
- Makes for fast connections to servers
- TLS 1.3 extends to let clients send data in the hello message
- Allows server to start sending responses quickly
- Risky
  - Early data can be replayed
  - Applications allowing early data must protect against this

Demo

Deployment

Current Support

MWR INFOSECURITY

- OpenSSL 1.1.1
- GnuTLS 3.6.4
- Chrome
- Firefox
- Java 11
- Cloudflare
- F5 BigIP 14.1.0.1
- Android Q

Deployment Scenarios

MWR INFOSECURITY

- Websites
  - Likely to support both TLS 1.2 and 1.3 for the foreseeable future
- Internal services
  - If you control both endpoints then why not?
  - TLS 1.3 is faster, so deployment is likely
  - No need for TLS 1.2 in this situation
- Mozilla TLS config generator agrees
  - Intermediate is 1.2 + 1.3
  - Modern is 1.3 only

Summary

- TLS 1.3 is coming
- You should be aware of it as a Pentester
  - It will impact your testing
  - Tooling support is uneven
  - Cross-checking the results is easy

Questions?

MWR INFOSECURITY