

Interesting hash collisions for X.509 certificates

(a.k.a. how to hop over an extension)

Ralf-Philipp Weinmann
Technische Universität Darmstadt

<ralf@coderpunks.org>



Overview

- Quick intro to ASN.1, DER, X.509 and X509v3 certificate extensions
- Hash collisions: Using length fields as “conditional jumps”
- Hiding and hopping over extensions
- Impact: Become your own root CA without spending big \$\$\$
- Conditions on the hash collisions
- Conclusions



ASN.1 / DER encoding

- Abstract Syntax Notation 1, defined in ITU X.680 (formerly X.208)
- defines notation, but not encoding
- different encodings
 - Basic Encoding Rules (BER, X.690)
 - Distinguished Encoding Rules (DER, X.690)
 - Packed Encoding Rules (PER, X.691)



More ASN.1

- different data types:
 - BOOLEAN, BIT STRING, INTEGER, OCTET STRING, SEQUENCE, SET, CHOICE, IA5String, OBJECT IDENTIFIER, etc.
- TLV encoding: Tag - Length - Value
- DER encoding: unique representation



Reading and writing ASN.1 by hand

- Just kidding, we don't actually have to do that:
- Neat tool by Matasano: Blackbag
 - converts DER encoded ASN.1 data into shell scripts
 - executing the shell script gives ASN.1 data again
- Still, helps to know what we're dealing with



Dissecting an ASN.1 structure

- `0x30820240 ...`
 - `0x30`: SEQUENCE type
 - `0x82`: MSB of byte indicates lower 7 bits give length of length field (`0x82 & 0x80 = 2` [bytes])
 - `0x0240`: SEQUENCE structure is 576 bytes long



X.509v3 cert extensions

```
Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension
```

```
Extension ::= SEQUENCE {  
    extnID          OBJECT IDENTIFIER,  
    critical        BOOLEAN DEFAULT FALSE,  
    extnValue       OCTET STRING  
}
```



A typical X.509 root CA certificate

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 0 (0x0)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=US, O=The Go Daddy Group, Inc., OU=Go Daddy Class 2 Certification Authority

Validity

Not Before: Jun 29 17:06:20 2004 GMT

Not After : Jun 29 17:06:20 2034 GMT

Subject: C=US, O=The Go Daddy Group, Inc., OU=Go Daddy Class 2 Certification Authority

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (2048 bit)

Modulus (2048 bit):

xx:xx:xx:xx;xx:xx:xx:xx:xx:xx:xx:xx;xx:xx:xx:

...

Exponent: 3 (0x3)

X509v3 extensions:

X509v3 Subject Key Identifier:

D2:C4:B0:D2:91:D4:4C:11:71:B3:61:CB:3D:A1:FE:DD:A8:6A:D4:E3

X509v3 Authority Key Identifier:

keyid:D2:C4:B0:D2:91:D4:4C:11:71:B3:61:CB:3D:A1:FE:DD:A8:6A:D4:E3

DirName:/C=US/O=The Go Daddy Group, Inc./OU=Go Daddy Class 2 Certification Authority

serial:00

X509v3 Basic Constraints:

CA:TRUE

Signature Algorithm: sha1WithRSAEncryption

xx:xx:xx:xx;xx:xx:xx:xx:xx:xx:xx:xx;xx:xx:xx:xx:xx:xx:

...



A typical X.509 root CA certificate

Certificate:

Data:

Version: 3 (0x2)
Serial Number: 0 (0x0)
Signature Algorithm: sha1WithRSAEncryption
Issuer: C=US, O=The Go Daddy Group, Inc., OU=Go Daddy Class 2 Certification Authority
Validity
Not Before: Jun 29 17:06:20 2004 GMT
Not After : Jun 29 17:06:20 2034 GMT
Subject: C=US, O=The Go Daddy Group, Inc., OU=Go Daddy Class 2 Certification Authority
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (2048 bit)
Modulus (2048 bit):
 xx:xx:xx:xx;xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:
 ...
Exponent: 3 (0x3)

X509v3 extensions:
X509v3 Subject Key Identifier:
 D2:C4:B0:D2:91:D4:4C:11:71:B3:61:CB:3D:A1:FE:DD:A8:6A:D4:E3
X509v3 Authority Key Identifier:
 keyid:D2:C4:B0:D2:91:D4:4C:11:71:B3:61:CB:3D:A1:FE:DD:A8:6A:D4:E3
 DirName:/C=US/O=The Go Daddy Group, Inc./OU=Go Daddy Class 2 Certification Authority
 serial:00
X509v3 Basic Constraints:
 CA:TRUE

Signature Algorithm: sha1WithRSAEncryption

xx:xx:xx:xx;xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:
...



A typical X.509 root CA certificate

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 0 (0x0)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=US, O=The Go Daddy Group, Inc., OU=Go Daddy Class 2 Certification Authority

Validity

Not Before: Jun 29 17:06:20 2004 GMT

Not After : Jun 29 17:06:20 2034 GMT

Subject: C=US, O=The Go Daddy Group, Inc., OU=Go Daddy Class 2 Certification Authority

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (2048 bit)

Modulus (2048 bit):

xx:xx:xx:xx;xx:xx:xx:xx:xx:xx:xx:xx;xx:xx:xx:

...

Exponent: 3 (0x3)

X509v3 extensions:

X509v3 Subject Key Identifier:

D2:C4:B0:D2:91:D4:4C:11:71:B3:61:CB:3D:A1:FE:DD:A8:6A:D4:E3

X509v3 Authority Key Identifier:

keyid:D2:C4:B0:D2:91:D4:4C:11:71:B3:61:CB:3D:A1:FE:DD:A8:6A:D4:E3

DirName:/C=US/O=The Go Daddy Group, Inc./OU=Go Daddy Class 2 Certification Authority

serial:00

X509v3 Basic Constraints:

CA:TRUE

Signature Algorithm: sha1WithRSAEncryption

xx:xx:xx:xx;xx:xx:xx:xx:xx:xx:xx:xx;xx:xx:xx:xx:xx:xx:

...



The idea of the attack

- High degree of freedom in first block of MD5 hash collisions
- Exploit this degree of freedom:
 - Place difference(s) on length field(s)
- Gain: “conditional jump”
- Allows to hide extension to signer in the certificate request



An Example (1)

- X.509 Basic Constraints extension: certificate can be marked as CA cert
- 14 bytes: 30 0c 06 03 55 1d 13 04 05 30 03 01 01 ff
- OID 2.5.29.19, OCTET STRING 30 03 01 01 ff
- OCTET STRING encodes:

```
SEQUENCE {  
    BOOLEAN TRUE  
}
```



An Example (2)

- Goal: hide these 14 bytes inside another extension
- pad with another bogus extension of 114 bytes



Inflatable extensions

- 30 82 02 XX+ Δ 06 08 2B 06 01 04 01 C0 6D 25 04 82 02 XX+ Δ

```
SEQUENCE { // total length 512 + XX + delta bytes
  OBJECT IDENTIFIER '1 3 6 1 4 1 8301 37'
  OCTET STRING // total length 512 + YY bytes
  ...
}
```

- Δ is appearing twice, 14 bytes apart
- distance can be trivially changed by changing OID
- fixed bytes: marked red
- complexity of MD5 collision search increases



Inflatable extensions

- `30 82 02` $XX+\Delta$ `06 08` 2B 06 01 04 01 C0 6D 25 04 `82 02` $XX+\Delta$

```
SEQUENCE { // total length 512 + XX + delta bytes
  OBJECT IDENTIFIER '1 3 6 1 4 1 8301 37'
  OCTET STRING // total length 512 + YY bytes
  ...
}
```

- Δ is appearing twice, 14 bytes apart
- distance can be trivially changed by changing OID
- fixed bytes: marked red
- complexity of MD5 collision search increases



Inflatable extensions

- `30 82 02 XX+Δ 06 08 2B 06 01 04 01 C0 6D 25 04 82 02 XX+Δ`

```
SEQUENCE { // total length 512 + XX + delta bytes
  OBJECT IDENTIFIER '1 3 6 1 4 1 8301 37'
  OCTET STRING // total length 512 + YY bytes
  ...
}
```

- Δ is appearing twice, 14 bytes apart
- distance can be trivially changed by changing OID
- fixed bytes: marked red
- complexity of MD5 collision search increases



Impact

- Root CAs can be tricked into unwittingly creating intermediate CA certificates
- Apply attack for **ANY** root CA in the list of trusted roots in Firefox, Internet Explorer, Safari and/or Opera
 - result: mass SSL breakage
- Much harder to do with SHA1 than with MD5
 - less degrees of freedom (according to Christian)



The last slide



The last slide

Questions?

Comments?

