Übungsaufgaben zur Vorlesung

IT-Sicherheit Grundlagen

Wolf Müller

Abgabetermin: Montag, 12.06.2023, 11:00 Uhr per Moodle
HU Berlin, Sommersemester 2023

Lab 2: Blockchiffren und das Padding Oracle (11 + 5 Punkte)

Aufgabe 2.1: Blockchiffren (3 Punkte)

Seit Monaten hat Agent Thurgood den Verdacht, dass sich ein Maulwurf in seinen eigenen Reihen versteckt, der regelmäßig vertrauliche Informationen an den Feind weitergibt. Er hatte es fast geschafft, den Verräter bei der letzten Übergabe von geheimen Dokumenten auf frischer Tat zu ertappen. Zumindest kannte er jetzt den geheimen Treffpunkt und wartete nur auf die nächste Gelegenheit, ihn dort zu ergreifen. Doch auch der Gegner hatte davon Wind bekommen und versuchte seinen Spitzel zu warnen, jedoch gelang es Thurgood's Überwachungsexperten die folgende geheime Nachricht abzufangen:

```
f67172d4cec9ef92a6c66527b5e22893 5a4de9f47483a1e9a302fd949f9d8dc2
```

Aus vertrauenswürdigen Quellen weiß Thurgood, dass die Nachricht mit AES-256 im CBC-Modus verschlüsselt ist. Die ersten 16 Byte sind der Initialisierungsvektor, die zweiten 16 Byte der eigentliche Chiffretext. Außerdem weiß er, dass der Klartext der übermittelten geheimen Nachricht folgender Text in ASCII-Codierung (ohne die Anführungszeichen) ist: „`Run,we're blown!`“. Helfen Sie Thurgood, die Nachricht so verändern, dass beim Spitzel die Nachricht „`Meet you tonight`“ entschlüsselt wird. Erklären Sie kurz, wie der Angriff funktioniert und wie man ihn verhindern kann.

Aufgabe 2.2: Padding Oracle (8 + 5 Punkte)

In dieser Aufgabe sollen Sie einen Angriff auf einen Webservice durchführen, der einen Chiffretext empfängt, der mit AES-256-CBC verschlüsselt wurde. Damit die verschlüsselte Nachricht ein Vielfaches der Blockgröße ist, wird ein PKCS#7-Padding eingeführt, wie hier zu sehen:
`https://en.wikipedia.org/wiki/Padding_%28cryptography%29#PKCS#5_and_PKCS#7`

Sollte der Server beim Entschlüsseln feststellen, dass dieses Padding falsch ist, so kann man das am Fehlercode erkennen. Der Server hat die folgende Adresse:

`http://gruenau2.informatik.hu-berlin.de:8888/`

Dort gibt es einen Link zu:

`http://gruenau2.informatik.hu-berlin.de:8888/store_secret/`

oder

`http://gruenau2.informatik.hu-berlin.de:8888/store_secret/?secret=`

Wenn Sie hinter dem Schrägstrich einen Chiffretext angeben, der erst Base64- und dann URL-codiert ist, erhalten Sie eine Antwort.

Aufgabe 2.2.A: Vertraulichkeit (8 Punkte)

Der Angriff den Sie implementieren können, wurde bereits 2002 auf der EuroCrypt von Sergey Vaudenay vorgestellt. Unter dem Link oder auch in dieser pdf-Datei finden Sie eine vereinfachte Beschreibung eines Angriffs, mit dem Sie den Klartext folgender Nachricht mit Hilfe des Servers finden können:

`2QicDQHnGmRuZys0M5JcwCSTeFNXvVm%2FSsG1vaEkIZU1OiGgpLJTdbRO2beA831a0xsatfOy01N38W1RidzrXA%3D%3D`

Alternativ können Sie auch die Variante mit dem Query-Parameter `secret`:

`/store_secret/?secret=KLQ%2B...%3D%3D`

nutzen.

Schreiben Sie ein Programm, das diesen Angriff implementiert und geben Sie den Klartext ab!

Aufgabe 2.2.B: Integrität (5 Zusatzpunkte)

Ändern Sie den gegebenen Chiffretext so, dass er vom Server zu

```
'); DROP TABLE Vorname;-- not the data you need
```

entschlüsselt wird. Dabei ersetzen Sie das Wort Vorname bitte mit einem Vornamen Ihrer Wahl, beispielsweise Ihrem eigenen. Um Ihr Ergebnis zu überprüfen, können Sie Ihre Lösung aus Teil 2.2.A verwenden. Geben Sie den geänderten Chiffretext ab!

Hinweise

- Die Abgabe erfolgt über Moodle.
- Sie können dafür eine Programmiersprache ihrer Wahl verwenden. Weiterhin können Sie beliebige Bibliotheken verwenden (z. B. Cryptography, PyCryptodome, Crypto++, OpenSSL ...).

- Die Ausgabe von Aufgabe 2.2 Vertraulichkeit soll lediglich den Klartext mit einem nachfolgendem Newline Character beinhalten.
- Um die Korrektur zu erleichtern schreiben Sie bitte ein Bash-Skript namens „`padding.sh`“, welches Ihren Quelltext ggfs. kompiliert und ausführt. Packen Sie alle Quelltextdateien, das Bash-Skript und die **pdf**-Dateien mit den Antworten und Anmerkungen in ein tar-Archiv. Kompilierte Dateien brauchen Sie nicht mit abzugeben, da das Compilieren ihr Skript übernehmen soll.
- Das Referenzsystem ist `gruenau2`.
- Das Orakel kann auf den Rechnern `gruenau[1-5].informatik.hu-berlin.de` auf dem Port `8888` erreicht werden und es können Anfragen aus dem Netz der HU gestellt werden.
- In das HU-Netz gelangen Sie, indem Sie sich auf einen Rechner dort einloggen oder OpenVPN verwenden.
- Alternativ können Sie auch *SSH port forwarding* verwenden, um den Port `8888` von `gruenau2.informatik.hu-berlin.de` auf Ihren lokalen Rechner `localhost` umzuleiten. Dies gelingt z.B. mit dem Kommando:
  `ssh -L 8888:localhost:8888 USER@gruenau2.informatik.hu-berlin.de`,
  wobei Sie natürlich `USER` durch Ihren eigenen Loginnamen ersetzen müssen. Solange das *SSH port forwarding* aktiv ist, können Sie statt dem Domainnamen `gruenau*` einfach `localhost` in den URLs verwenden.
- Folgender Golem–Artikel zeigt, dass Abwandlungen des Padding Oracles auch heute immer noch für Sicherheitslücken verantwortlich sind.