SVS Bachelor-Projekt Network Security

Blatt 3: Datenkommunikation

Louis Kobras
6658699

Utz Pöhlmann
6663579

1 HTTP

1.1

- hat nicht auf Pings reagiert (VM)
- RealOS: Ping an IP 134.100.56.130 erfolgreich, `telnet` fehlgeschlagen

2 SMPT (Mail Spoofing)

2.1

zwei Sätze zum Topic
Input Protokoll
Kommentar erste Mail vs. echte Mail
Modifikation bei zweiter Mail

3 License Server (DNS-Spoofing)

3.1

Protokoll:

1. Key als User-Input
2. Übermitteln des Keys an den Server
3. Rückgabe vom Server, ob Key gültig oder nicht (`SERIAL_VALID=0` bzw. `SERIAL_VALID=1`)

4a Wenn gültig, Dank für Kauf

4b Wenn nicht gültig, FBI ist unterwegs

3.2

- Verhindern der Kommunikation der Software mit dem echten Auth-Server
- Geschehen durch Erweitern des Hosts um `127.0.0.1 license-server.svslab` in `/etc/hosts`
- Herunterladen der Java-Klasse `TCPClient.java`[1]
- Manipulieren des Servers: `ServerSocket` auf svslab-Port (1337) gesetzt
- Manipulieren des Servers: Rückgabe des Servers auf statisch `‘‘SERIAL_VALID=1’’` gesetzt
- $\Longrightarrow$ alle Keys gültig, unabhängig von Eingabe

[1] https://systembash.com/a-simple-java-tcp-server-and-tcp-client/

3.3

Es gibt zwei anmerkbare Mängel.

1. Es sollte nicht angegeben werden, ob die Serial-Länge korrekt ist.
2. Es könnte mithilfe einer eindeutigen Signatur o.Ä. eine Abfrage an den Server eingebunden werden, ob er "echt" ist (gehasht).

4 License Server (Brute-Force-Angriff)

4.1

Das Programm funktioniert an sich, wenn man aber an den Server sendet, kriegt man (scheinbar nach Zufall) entweder "invalid command" oder "invalid length" zurück, bei Eingabe von `serial=abcdefgh` ($a, b, c, d, e, f, g, h \in \{0, 1, ..., 9\}$).

Wir baten zwei Gruppen neben uns um Hilfe, jedoch konnten diese uns auch nicht weiterhelfen bzw. haben keinen Fehler in unserem Programm gefunden.

Als Ausgangspunkt wurde die Java-Klasse `TCPClient.java` von [todo:link] genommen.

gültige Keys:

- 90877300
- 31337000
- 21935900
- 62674000

4.2

Möglichkeiten, sich zu verteidigen, enthalten, sind jedoch nicht beschränkt auf:

- Sperren des Absenders der Auth-Anfrage nach n Fehlversuchen (Unterbrechen von Brute-Force-Attacken)[2]
- Prüfung der IP bzw. Prüfsumme, ob Empfänger und Absender korrekt sind (Zurechenbarkeit)
- Limitieren der Eingabe auf k pro Minute (Verlangsamen von Brute-Force-Attacken)

4.3

5 Implementieren eines TCP-Chats

5.1

5.2

5.3

5.4

5.5

[2] Je nach Art der Sperrung ist dies lediglich eine Bremse; wird z.B. nur die IP gesperrt, kann diese resettet werden, um wieder Zugang zu erlangen.