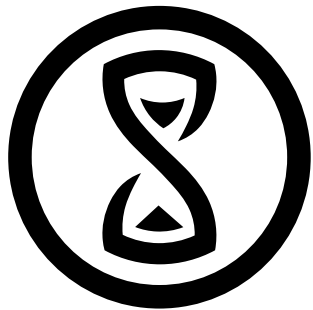
Hash Extension Attacks

Wie hashbasierte Signaturen nicht sein sollten (und warum)



Kasalehlia

14. November 2015

Kryptologische Hashfunktionen

> *Eine [kryptoligische] Hashfunktion ist eine Funktion, die eine Zeichenfolge beliebiger Länge auf eine Zeichenfolge mit fester Länge abbildet. Mathematisch ist diese Funktion nicht injektiv (linkseindeutig) und nicht notwendigerweise surjektiv (rechtstotal).*
>
> Wikipedia

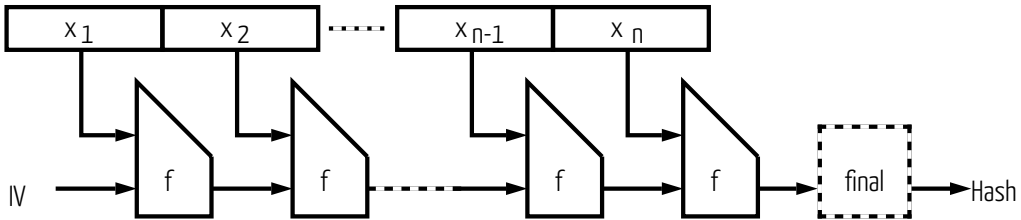
z.B. MD4, MD5, SHA-1, SHA-2, Whirlpool, SHA-3/Keccak

Merkle-Damgård Konstruktion

1. Auffüllen auf Vielfaches der Blockgröße (*Padding*)
2. Rekursive Anwendung einer Kompressionsfunktion (mit einem Initialisierungsvektor)
3. *Optional* Anwendung einer Finalisierungsfunktion

Merkle-Damgård Konstruktion

Ausgehend von Blockgröße *a*, Hashlänge *b* und Kompressionsfunktion $f : \{0,1\}^{a+b} \rightarrow \{0,1\}^{b}$



verwendet von: MD5, SHA-1, SHA-2, ...

MD5

a = 512bit, b = 128bit

MD5

a = 512bit, *b* = 128bit

Padding

- Anhängen von 0x80
- Mit 0x00 Auffüllen bis Länge = n**a* - 64bit
- Länge der Ausgansnachricht als 64bit little-endian Integer anhängen

MD5

a = 512bit, *b* = 128bit

Padding

- Anhängen von 0x80
- Mit 0x00 Auffüllen bis Länge = n**a* - 64bit
- Länge der Ausgansnachricht als 64bit little-endian Integer anhängen

Kompression

640bit auf 128bit

MD5

a = 512bit, *b* = 128bit

Padding

- Anhängen von 0x80
- Mit 0x00 Auffüllen bis Länge = n**a* - 64bit
- Länge der Ausgansnachricht als 64bit little-endian Integer anhängen

Kompression

640bit auf 128bit

Finalisierung

keine

Attacke

Beispiel: Webseitensession (stateless server)
Es werden zwei Cookies gesetzt:

Attacke

Beispiel: Webseitensession (stateless server)
Es werden zwei Cookies gesetzt:

session name=user&admin=false

Attacke

Beispiel: Webseitensession (stateless server)
Es werden zwei Cookies gesetzt:

session name=user&admin=false

signature ea7b6086a667265f2c38f49925441f26

Attacke

Beispiel: Webseitensession (stateless server)
Es werden zwei Cookies gesetzt:

session name=user&admin=false

signature ea7b6086a667265f2c38f49925441f26

Die Sigantur ist MD5(<8 byte secret><session>)

Extension

Nun kann ein Angreifer den Klartext erweitern ohne den Anfang zu kennen:

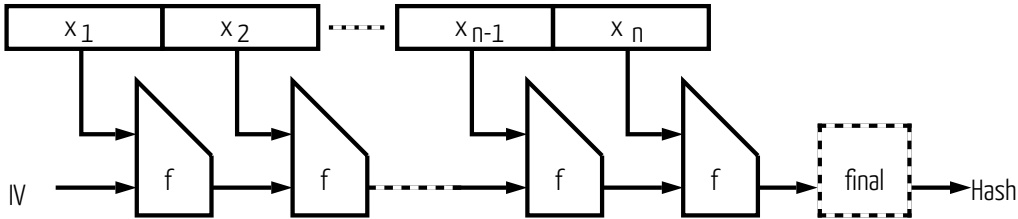
Extension

Nun kann ein Angreifer den Klartext erweitern ohne den Anfang zu kennen:

Neuer Klartext: name=user&admin=false<*Padding*>&admin=true

Merkle-Damgård Konstruktion (recall)

Ausgehend von Blockgröße a, Hashlänge b und Kompressionsfunktion $f : \{0,1\}^{a+b} \rightarrow \{0,1\}^{b}$



verwendet von: MD5, SHA-1, SHA-2, ...

HashPump

https://github.com/bwall/HashPump

HashPump

https://github.com/bwall/HashPump

> hashpump -s <Signatur> -d <Klartext> -a <Erweiterung> -k <Länge des Secret>

HashPump

https://github.com/bwall/HashPump

> hashpump -s <Signatur> -d <Klartext> -a <Erweiterung> -k <Länge des Secret>

> hashpump -s 'ea7b6086a667265f2c38f49925441f26' -d 'name=user&admin=false' -a '&admin=true' -k 8

HashPump

https://github.com/bwall/HashPump

> hashpump -s <Signatur> -d <Klartext> -a <Erweiterung> -k <Länge des Secret>

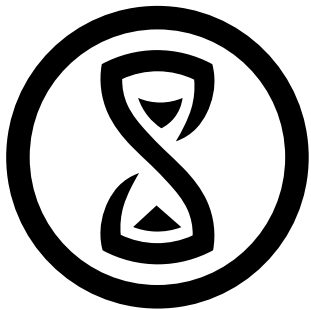
> hashpump -s 'ea7b6086a667265f2c38f49925441f26' -d 'name=user&admin=false' -a '&admin=true' -k 8

4f546b6a3b849af27ff930a7b9ea6d3d
name=user&admin=false\x80\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\xe8\x00\x00\x00\x00\x00\x00\x00&admin=true

Gegenmaßnahmen

- andere Hashverfahren nutzen
- Secret mehrfach einfließen lassen
- Datenstruktur verwenden die nicht erweiterbar ist (z.B. JSON)

Kasalehlia
kasa@lehlia.de



Stratum 0 e. V. Braunschweig
https://stratum0.org/