Härtere Crypto für unsere Services

Michael F. Herbst
michael.herbst@iwr.uni-heidelberg.de
http://blog.mfhs.eu

Interdisziplinäres Zentrum für wissenschaftliches Rechnen
Ruprecht-Karls-Universität Heidelberg

29 Januar 2015

Disclaimer

Der Vortragende übernimmt für nichts Verantwortung, denn

Disclaimer

Der Vortragende übernimmt für nichts Verantwortung, denn

ich hab auch keine Ahnung von guter Crypto.

Übersicht

`ssh` Standardeinstellungen

HostKeyAlgorithms

```
ecdsa-sha2-nistp256-cert-v01@openssh.com, ecdsa-sha2-nistp384-cert-v01@openssh.com,
ecdsa-sha2-nistp521-cert-v01@openssh.com, ssh-ed25519-cert-v01@openssh.com,
ssh-rsa-cert-v01@openssh.com, ssh-dss-cert-v01@openssh.com,
ssh-rsa-cert-v00@openssh.com, ssh-dss-cert-v00@openssh.com, ecdsa-sha2-nistp256,
ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, ssh-ed25519, ssh-rsa, ssh-dss
```

KexAlgorithms

```
curve25519-sha256@libssh.org, ecdh-sha2-nistp256, ecdh-sha2-nistp384,
ecdh-sha2-nistp521, diffie-hellman-group-exchange-sha256,
diffie-hellman-group14-sha1, diffie-hellman-group-exchange-sha1,
diffie-hellman-group1-sha1
```

Ciphers

```
aes128-ctr, aes192-ctr, aes256-ctr, aes128-gcm@openssh.com,
aes256-gcm@openssh.com, chacha20-poly1305@openssh.com, arcfour256, arcfour128,
aes128-cbc, 3des-cbc, blowfish-cbc, cast128-cbc, aes192-cbc, aes256-cbc, arcfour
```

MACs

```
umac-64-etm@openssh.com, umac-128-etm@openssh.com, hmac-sha2-256-etm@openssh.com,
hmac-sha2-512-etm@openssh.com, umac-64@openssh.com, umac-128@openssh.com,
hmac-sha2-256, hmac-sha2-512, hmac-md5-etm@openssh.com,
hmac-sha1-etm@openssh.com, hmac-ripemd160-etm@openssh.com,
hmac-sha1-96-etm@openssh.com, hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1,
hmac-ripemd160, hmac-sha1-96, hmac-md5-96
```

`ssh` Standardeinstellungen

HostKeyAlgorithms

ecdsa-sha2-nistp256-cert-v01@openssh.com, ecdsa-sha2-nistp384-cert-v01@openssh.com,
ecdsa-sha2-nistp521-cert-v01@openssh.com, ssh-ed25519-cert-v01@openssh.com,
ssh-rsa-cert-v01@openssh.com, ssh-dss-cert-v01@openssh.com,
ssh-rsa-cert-v00@openssh.com, ssh-dss-cert-v00@openssh.com, ecdsa-sha2-nistp256,
ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, ssh-ed25519, ssh-rsa, ssh-dss

KexAlgorithms

curve25519-sha256@libssh.org, ecdh-sha2-nistp256, ecdh-sha2-nistp384,
ecdh-sha2-nistp521, diffie-hellman-group-exchange-sha256,
diffie-hellman-group14-sha1, diffie-hellman-group-exchange-sha1,
diffie-hellman-group1-sha1

Ciphers

aes128-ctr, aes192-ctr, aes256-ctr, aes128-gcm@openssh.com,
aes256-gcm@openssh.com, chacha20-poly1305@openssh.com, arcfour256, arcfour128,
aes128-cbc, 3des-cbc, blowfish-cbc, cast128-cbc, aes192-cbc, aes256-cbc, arcfour

MACs

umac-64-etm@openssh.com, umac-128-etm@openssh.com, hmac-sha2-256-etm@openssh.com,
hmac-sha2-512-etm@openssh.com, umac-64@openssh.com, umac-128@openssh.com,
hmac-sha2-256, hmac-sha2-512, hmac-md5-etm@openssh.com,
hmac-sha1-etm@openssh.com, hmac-ripemd160-etm@openssh.com,
hmac-sha1-96-etm@openssh.com, hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1,
hmac-ripemd160, hmac-sha1-96, hmac-md5-96

Table of Contents

Key exchange Algorithmen

DH	Diffie-Hellmann
DHE	ephemeral Diffie-Hellman (forward secrecy)
RSA	Rivest-Shamir-Adleman
PSK	pre-shared key

Key exchange Algorithmen

DH	Diffie-Hellmann
DHE	ephemeral Diffie-Hellman (forward secrecy)
RSA	Rivest-Shamir-Adleman
PSK	pre-shared key

Empfehlung

ganz klar: DHE

Ciphers

Liste

AES-CTR	Advanced Encryption Standard (Counter mode)
AES-CBC	AES (Cipher Block Chaining mode)
AES-GCM	AES(Galois Counter Mode)
RC4	Stromschiffre
Camellia	wenig benutzte Blockschiffre von Mitsubishi/NTT
3DES	Blockschiffre
ChaCha20	neue Stromschiffre von DJB

Ciphers

Empfehlung

- AES-CTR oder AES-GCM sind Standard
- Camellia ok, aber wenig getestet
- ChaCha20 ist ziemlich neu
- Blockgröße $\geq 128bit$

MACs

Liste

SHA	secure hash algorithm
SHA-1	verbesserter SHA
SHA-$n	SHA-2 Familie (224 bis 512)
SHA-3/Keccak	nicht weit verbreitet
MD5	Message Digest ver. 5
Poly1305	Neuer Digest von DJB
UMAC	“universal”: Zufällig gewählter Hash

MACs

Empfehlung

- MD5 und SHA/SHA-1 sind mind. verwundbar
- SHA-2 ist beste Wahl Verbreitung/Sicherheit
- Keygröße $\geq 128bit$
- Immer ETM (encrypt-than-MAC)!

Elliptische Kurven

- enorme Performancevorteile
- theoretisch keine Sicherheitseinbußen
- NIST Kurven problematisch
- DJB Kurve: ed25519

Table of Contents

openssl

- Erlaubte Cipher ermitteln:

  `openssl ciphers -V '$STRING'`

- Für Paranoide (openssl > `1.0.1e`):

  `EDH+aRSA+AES256:!SSLv3:@STRENGTH`

- Für Realisten:

  `EDH+aRSA:+AES256:+CAMELLIA128:+AES128:+SSLv3:!aNULL:!eNULL:`
  `!LOW:!3DES:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!ECDSA:!EECDH:`
  `CAMELLIA256-SHA:AES256-SHA:CAMELLIA128-SHA:AES128-SHA`

iceweasel und icedove

`about:config` settings

- security.ssl3.`$CIPHER` = true/false
- DEMO

iceweasel und icedove

`about:config` settings

- security.ssl3.`$CIPHER` = true/false
- DEMO

SSH Server

```
Protocol 2
#kein dsa oder ecdsa
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
PermitRootLogin no # oder 'without-password'
StrictModes yes
PermitEmptyPasswords no

KexAlgorithms curve25519-sha256@libssh.org,
    diffie-hellman-group-exchange-sha256
Ciphers chacha20-poly1305@openssh.com,
    aes256-gcm@openssh.com, aes128-gcm@openssh.com,
    aes256-ctr, aes192-ctr, aes128-ctr
MACs hmac-sha2-512-etm@openssh.com,
    hmac-sha2-256-etm@openssh.com,
    hmac-ripemd160-etm@openssh.com,
    umac-128-etm@openssh.com, hmac-sha2-512, hmac-sha2-256,
    hmac-ripemd160, umac-128@openssh.com, hmac-sha1
```

.ssh/config

Starke Ciphers als Default

```
...

Host *
    ServerAliveInterval 300
    Protocol 2

    HostKeyAlgorithms ssh-ed25519-cert-v01@openssh.com,
        ssh-rsa-cert-v01@openssh.com,
        ssh-rsa-cert-v00@openssh.com, ssh-ed25519,ssh-rsa

    KexAlgorithms curve25519-sha256@libssh.org,
        diffie-hellman-group-exchange-sha256

    Ciphers chacha20-poly1305@openssh.com,
        aes256-gcm@openssh.com, aes128-gcm@openssh.com,
        aes256-ctr, aes192-ctr, aes128-ctr

    MACs hmac-sha2-512-etm@openssh.com,
        hmac-sha2-256-etm@openssh.com,
        hmac-ripemd160-etm@openssh.com,
        umac-128-etm@openssh.com
```

.ssh/config

Schwächere Cipher

- Für viele Hosts muss man die MAC-Liste erweitern:

```
Host Host1 Host2 Host3
    MACs hmac-sha2-512-etm@openssh.com,
         hmac-sha2-256-etm@openssh.com,
         hmac-ripemd160-etm@openssh.com,
         umac-128-etm@openssh.com, hmac-sha2-512,
         hmac-sha2-256, hmac-ripemd160, umac-128@openssh.com,
         hmac-sha1

...

Host *
    (siehe vorherige Folie)
```

Table of Contents

Endergebnis

- Die defaults sind ok, aber nicht optimal
- Bessere Crypto möglich wenn Kompatibilität keine Rolle spielt

Links

- `https://bettercrypto.org/static/applied-crypto-hardening.pdf`
- `ach@lists.cert.at`
- `https://stribika.github.io/2015/01/04/secure-secure-shell.html`
- Die Folien findet ihr bald auf `http://blog.mfhs.eu`