Crypto-Policies – Was ist das?

Kieler Open Source Linux Tage 2024

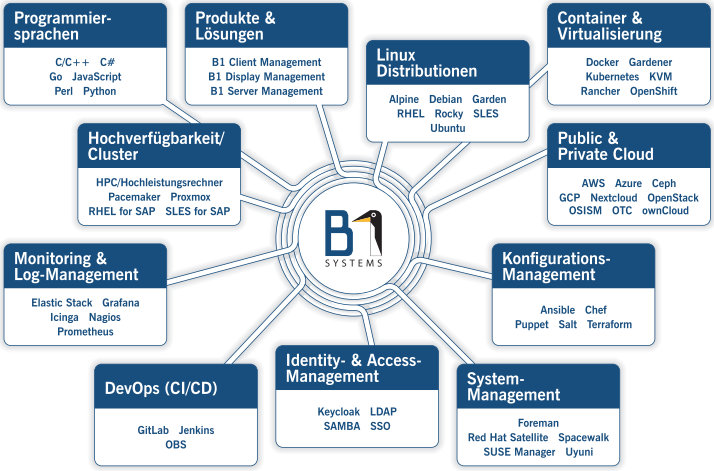
20. September 2024

Susanne Schütze
B1 Systems GmbH
schuetze@b1-systems.de

Vorstellung B1 Systems

- gegründet 2004
- spezialisiert auf Linux/Open Source-Themen
- national & international tätig
- ca. 150 Mitarbeiter:innen
- unabhängig von Soft- & Hardware-Herstellern
- Leistungsangebot:
  - Managed Service & Betrieb
  - Beratung & Consulting
  - Support
  - Training
  - Lösungen & Entwicklung
- Standorte in Rockolding, Köln, Berlin, Dresden & Jena

Schwerpunkte



whoami

- Susanne Schütze
- 40 Jahre
- Fachinformatikerin für Systemintegration
- bei B1 Systems GmbH seit Juli 2024
- berufliche Themen: Linux Client Management, Development, Ansible, Salt, etc

Frage zu Beginn

Wer von euch musste letztens Probleme von Kolleg:innen lösen?

Die Fehlerbeschreibung

Kollege

Ich kann mich nicht mehr mit meinem SSH-Key mit den RHEL8-Servern verbinden, kannst du mal rauskriegen, woran das liegt? Du magst doch SSH.

...

Die Fehlerbeschreibung

Kollege

Ich kann mich nicht mehr mit meinem SSH-Key mit den RHEL8-Servern verbinden, kannst du mal rauskriegen, woran das liegt? Du magst doch SSH.

Susanne

Klar
Was hast du für ‘nen SSH-Key?
Wie lautet die Fehlermeldung?

. . .

Die Fehlerbeschreibung

Kollege

Ich kann mich nicht mehr mit meinem SSH-Key mit den RHEL8-Servern verbinden, kannst du mal rauskriegen, woran das liegt? Du magst doch SSH.

Susanne

Klar
Was hast du für ‘nen SSH-Key?
Wie lautet die Fehlermeldung?

Kollege

Ich hab ‘nen normalen RSA SSH-Key und ich hab keine Fehlermeldung, es kommt nur die Passwort-Abfrage von dem Server

. . .

Die Fehlerbeschreibung

Kollege

Ich hab ‘nen normalen RSA SSH-Key und ich hab keine Fehlermeldung, es kommt nur die Passwort-Abfrage von dem Server

Susanne

Wie viele Bits hat denn dein RSA?
Hast du mal `ssh -v user@Server` oder
`ssh -oKexAlgorithms=ecdh-sha2-nistp256 user@Server` versucht?

...

Die Fehlerbeschreibung

Kollege

Ich hab ‘nen normalen RSA SSH-Key und ich hab keine Fehlermeldung, es kommt nur die Passwort-Abfrage von dem Server

Susanne

Wie viele Bits hat denn dein RSA?
Hast du mal `ssh -v user@Server` oder
`ssh -oKexAlgorithms=ecdh-sha2-nistp256 user@Server` versucht?

Kollege

Natürlich * Bits und ich benutze Putty

...

Die Fehlerbeschreibung

Susanne

Wie viele Bits hat denn dein RSA?
Hast du mal `ssh -v user@Server` oder
`ssh -oKexAlgorithms=ecdh-sha2-nistp256 user@Server` versucht?

Kollege

Natürlich * Bits und ich benutze Putty

Susanne

Ich schau es mir an . . .

. . .

Die Fehlerbeschreibung

Kollege

Natürlich * Bits und ich benutze Putty

Susanne

Ich schau es mir an . . .

Kollege

Aber deine Lösung muss idempotent sein und mit Ansible umsetzbar

. . .

Kryptographie im System – aktueller Stand

Wer von euch:

- hat den Überblick über alle Krypto-Konfigurationsmöglichkeiten?
- ist der Meinung, dass die Konfigurations-Optionen bezüglich Kryptographie einheitlich sind?
- findet es easy, mal eben systemweit z. B. SHA1 auszuschalten?

Kryptographie im System – aktueller Stand

Wer von euch:

- hat den Überblick über alle Krypto-Konfigurationsmöglichkeiten?
- ist der Meinung, dass die Konfigurations-Optionen bezüglich Kryptographie einheitlich sind?
- findet es easy, mal eben systemweit z. B. SHA1 auszuschalten?

Kryptographie im System – aktueller Stand

Wer von euch:

- hat den Überblick über alle Krypto-Konfigurationsmöglichkeiten?
- ist der Meinung, dass die Konfigurations-Optionen bezüglich Kryptographie einheitlich sind?
- findet es easy, mal eben systemweit z. B. SHA1 auszuschalten?

Kryptographie im System – aktueller Stand

- jedes Tool hat eigene Krypto-Regeln
- Krypto-Regeln werden in der Konfigurationsdatei des Tools definiert

Listing: Auszug aus Datei `/etc/ssh/sshd.conf`

```
1 Ciphers aes128-ctr,aes192-ctr,aes256-ctr
2 HostKeyAlgorithms
  ↪  ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521
3 KexAlgorithms ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp52⌋
  ↪  1,diffie-hellman-group14-sha1,diffie-hellman-group-exchange-sha256
4 MACs hmac-sha2-256,hmac-sha2-512,hmac-sha1
```

Kryptographie im System – aktueller Stand

- jedes Tool hat eigene Krypto-Regeln
- Krypto-Regeln werden in der Konfigurationsdatei des Tools definiert

Listing: Auszug aus Datei `/etc/ssh/sshd.conf`

```
1 Ciphers aes128-ctr,aes192-ctr,aes256-ctr
2 HostKeyAlgorithms
  ↪  ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521
3 KexAlgorithms ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp52⌋
  ↪  1,diffie-hellman-group14-sha1,diffie-hellman-group-exchange-sha256
4 MACs hmac-sha2-256,hmac-sha2-512,hmac-sha1
```

Debugging des SSH-Fehlers

- `ssh -oKexAlgorithms=ecdh-sha2-nistp256` wird benötigt
- Algorithmus-Änderungen in `/etc/ssh/sshd_config` ohne Effekt
- sshd-Unit?

Listing: Terminal Ausgabe

```
[root@crypt-arbeit8 ~]# systemctl status sshd
* sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service)
   Active: active (running) since 6min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 669 (sshd)
    Tasks: 1 (limit: 11160)
   Memory: 6.4M
   CGroup: /system.slice/sshd.service
↪  -oCiphers=aes256-gcm@openssh.com,aes256-ctr,aes256-cbc,aes128-gcm@openssh
↪  .com,aes128-ctr,aes128-cbc -oMACs=hmac-sha2-256-etm@openssh.com,...
```

Debugging des SSH-Fehlers

- `ssh -oKexAlgorithms=ecdh-sha2-nistp256` wird benötigt
- Algorithmus-Änderungen in `/etc/ssh/sshd_config` ohne Effekt
- sshd-Unit?

Listing: Terminal Ausgabe

```
[root@crypt-arbeit8 ~]# systemctl status sshd
* sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service)
   Active: active (running) since 6min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 669 (sshd)
    Tasks: 1 (limit: 11160)
   Memory: 6.4M
   CGroup: /system.slice/sshd.service
↪  -oCiphers=aes256-gcm@openssh.com,aes256-ctr,aes256-cbc,aes128-gcm@openssh
↪  .com,aes128-ctr,aes128-cbc -oMACs=hmac-sha2-256-etm@openssh.com,...
```

Debugging des SSH-Fehlers

- `ssh -oKexAlgorithms=ecdh-sha2-nistp256` wird benötigt
- Algorithmus-Änderungen in `/etc/ssh/sshd_config` ohne Effekt
- sshd-Unit?

Listing: Terminal Ausgabe

```
[root@crypt-arbeit8 ~]# systemctl status sshd
* sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service)
   Active: active (running) since 6min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 669 (sshd)
    Tasks: 1 (limit: 11160)
   Memory: 6.4M
   CGroup: /system.slice/sshd.service
↪  -oCiphers=aes256-gcm@openssh.com,aes256-ctr,aes256-cbc,aes128-gcm@openssh
↪  .com,aes128-ctr,aes128-cbc -oMACs=hmac-sha2-256-etm@openssh.com,...
```

Debugging des SSH-Fehlers

Listing: Service Unit /usr/lib/systemd/system/sshd.service

```
1  [Unit]
2  Description=OpenSSH server daemon
3  Documentation=man:sshd(8) man:sshd_config(5)
4  After=network.target sshd-keygen.target
5  Wants=sshd-keygen.target
6  [Service]
7  Type=notify
8  EnvironmentFile= -/etc/crypto-policies/back-ends/opensshserver.config
9  EnvironmentFile=-/etc/sysconfig/sshd
10 ExecStart=/usr/sbin/sshd -D $OPTIONS $CRYPTO_POLICY
11 ExecReload=/bin/kill -HUP $MAINPID
12 KillMode=process
13 Restart=on-failure
14 RestartSec=42s
15 [Install]
16 WantedBy=multi-user.target
```

Einführung Crypto-Policies

- alle kryptographischen Einstellungen über ein systemweites Tool konfigurieren
- die Cipher Suite an einem Ort konfigurieren, überschreibt Tool-Konfiguration
- Crypto-Einstellungen in Konfigurations-Dateien werden wirkungslos
- leichter zu maintainen, zu updaten, anzupassen
- Vorteil, wenn Systeme bestimmten Sicherheitsstandards entsprechen sollen
- bisher in Fedora, RHEL, CentOS, openSUSE, Oracle Linux, Ubuntu, Debian Sid (unstable), ...
- Entwicklung:

`https://gitlab.com/redhat-crypto/fedora-crypto-policies`

- Eigenentwicklung von Red Hat, Idee findet jedoch Konsens in Community

One tool to rule them all

Wofür können Crypto-Policies momentan verwendet werden?

- libssh SSH2 protocol implementation
- sequoia PGP, outside of rpm-sequoia
- rpm-sequoia PGP backend
- BIND DNS
- GnuTLS
- Kerberos 5
- Libreswan IPsec and IKE protocol implementation
- NSS TLS library
- OpenJDK runtime environment
- OpenSSH SSH2
- OpenSSL TLS library

weitere Libraries sind in aktiver Entwicklung

Arten von Policies

LEGACY	kompatibel mit RHEL 5
FUTURE	Vorhersage zukünftiger Bedrohungen
BSI	nach BSI-Richtlinie RT-02102-2 (bisher erst in Fedora39) [1]
FIPS	genügt FIPS 140 Anforderungen
DEFAULT	in der Regel aktiviert
EMPTY	ausschließlich für Debugging; deaktiviert alle Kryptographischen Algorithmen

[1] https://gitlab.com/redhat-crypto/fedora-crypto-policies/-/blob/master/policies/BSI.pol

Arten von Policies

LEGACY	kompatibel mit RHEL 5
FUTURE	Vorhersage zukünftiger Bedrohungen
BSI	nach BSI-Richtlinie RT-02102-2 (bisher erst in Fedora39) [1]
FIPS	genügt FIPS 140 Anforderungen
DEFAULT	in der Regel aktiviert
EMPTY	ausschließlich für Debugging; deaktiviert alle Kryptographischen Algorithmen

[1] https://gitlab.com/redhat-crypto/fedora-crypto-policies/-/blob/master/policies/BSI.pol

Policies anzeigen und setzen

- anzeigen: `update-crypto-policies --show`
- ändern: `update-crypto-policies --set FUTURE:NO-SHA1`
  - setzt die systemweiten Policies auf „Future“ mit dem Module „no-sha1“
  - unabhängig davon, welche vorher aktiv war
  - Module dazu laden: mit Doppelpunkt trennen, auch mehrfach
  - symbolische Links von `/etc/crypto-policies/back-ends` nach `/usr/share/crypto-policies`
  - generiert Backend Konfigurations-Dateien

Policies deaktivieren

- deaktivieren
  - bei SSH über eine Variable in der Konfigurationsdatei (`/etc/ssh/sshd.conf`) als opt-out
  - über die CLI mit Cipher-Optionen
- nach Änderungen an den Policies wird ein Neustart empfohlen, weil evtl. viele Services betroffen sind

Erste Lösung

Listing: Terminal Ausgabe

```
[root@arbeit ~]# update-crypto-policies --show
FIPS
```

- manuell bearbeitete Dateien:
  - /etc/crypto-policies/back-ends/opensshserver.config
  - /etc/crypto-policies/back-ends/libssh.config
  - /etc/crypto-policies/back-ends/openssh.config

```
1 CRYPTO_POLICY='-oCiphers=aes256-gcm@openssh.com,aes256-ctr,aes256-cbc,aes128-g
  ↪ cm@openssh.com,aes128-ctr,aes128-cbc
  ↪ -oMACs=hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-5
  ↪ 12-etm@openssh.com,hmac-sha2-256,hmac-sha1,hmac-sha2-512
  ↪ -oGSSAPIKeyExchange=no -oKexAlgorithms=ecdh-sha2-nistp256,ecdh-sha2-nistp3
  ↪ 84,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-
  ↪ group14-sha256,diffie-hellman-group16-sha512,...
```

Erste Lösung

Listing: Terminal Ausgabe

```
[root@arbeit ~]# update-crypto-policies --show
FIPS
```

- manuell bearbeitete Dateien:
  - /etc/crypto-policies/back-ends/opensshserver.config
  - /etc/crypto-policies/back-ends/libssh.config
  - /etc/crypto-policies/back-ends/openssh.config

```
CRYPTO_POLICY='-oCiphers=aes256-gcm@openssh.com,aes256-ctr,aes256-cbc,aes128-g
↪ cm@openssh.com,aes128-ctr,aes128-cbc
↪ -oMACs=hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-5
↪ 12-etm@openssh.com,hmac-sha2-256,hmac-sha1,hmac-sha2-512
↪ -oGSSAPIKeyExchange=no -oKexAlgorithms=ecdh-sha2-nistp256,ecdh-sha2-nistp3
↪ 84,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-
↪ group14-sha256,diffie-hellman-group16-sha512,...
```

Erste Lösung

Listing: Terminal Ausgabe

```
[root@arbeit ~]# update-crypto-policies --show
FIPS
```

- manuell bearbeitete Dateien:
  - /etc/crypto-policies/back-ends/opensshserver.config
  - /etc/crypto-policies/back-ends/libssh.config
  - /etc/crypto-policies/back-ends/openssh.config

```
1 CRYPTO_POLICY='-oCiphers=aes256-gcm@openssh.com,aes256-ctr,aes256-cbc,aes128-g
  ↪ cm@openssh.com,aes128-ctr,aes128-cbc
  ↪ -oMACs=hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-5
  ↪ 12-etm@openssh.com,hmac-sha2-256,hmac-sha1,hmac-sha2-512
  ↪ -oGSSAPIKeyExchange=no -oKexAlgorithms=ecdh-sha2-nistp256,ecdh-sha2-nistp3
  ↪ 84,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-
  ↪ group14-sha256,diffie-hellman-group16-sha512,...
```

Meinungsumfage

Eine Manpage ...

- ... ist die *Single Source of Truth* um zu wissen, was ein Programm kann, bzw. nicht kann
- stellt die ideale Funktionsweise einen Programms dar
- ist der perfekte Ort um CLI-Nerds (RTFM) mit Werbung für ein Programm zu versorgen
- enthält Programm-Features, die noch nicht im Programm integriert sind
- ist ein fabelhaftes Versprechen, was ein Programm alles für Funktionen hat
- ist keine Anleitung zu dem Programm

Meinungsumfage

Eine Manpage ...

- ... ist die *Single Source of Truth* um zu wissen, was ein Programm kann, bzw. nicht kann
- stellt die ideale Funktionsweise einen Programms dar
- ist der perfekte Ort um CLI-Nerds (RTFM) mit Werbung für ein Programm zu versorgen
- enthält Programm-Features, die noch nicht im Programm integriert sind
- ist ein fabelhaftes Versprechen, was ein Programm alles für Funktionen hat
- ist keine Anleitung zu dem Programm

Meinungsumfage

Eine Manpage ...

- ... ist die *Single Source of Truth* um zu wissen, was ein Programm kann, bzw. nicht kann
- stellt die ideale Funktionsweise einen Programms dar
- ist der perfekte Ort um CLI-Nerds (RTFM) mit Werbung für ein Programm zu versorgen
- enthält Programm-Features, die noch nicht im Programm integriert sind
- ist ein fabelhaftes Versprechen, was ein Programm alles für Funktionen hat
- ist keine Anleitung zu dem Programm

Meinungsumfage

Eine Manpage ...

- ... ist die *Single Source of Truth* um zu wissen, was ein Programm kann, bzw. nicht kann
- stellt die ideale Funktionsweise einen Programms dar
- ist der perfekte Ort um CLI-Nerds (RTFM) mit Werbung für ein Programm zu versorgen
- enthält Programm-Features, die noch nicht im Programm integriert sind
- ist ein fabelhaftes Versprechen, was ein Programm alles für Funktionen hat
- ist keine Anleitung zu dem Programm

Meinungsumfage

Eine Manpage ...

- ... ist die *Single Source of Truth* um zu wissen, was ein Programm kann, bzw. nicht kann
- stellt die ideale Funktionsweise einen Programms dar
- ist der perfekte Ort um CLI-Nerds (RTFM) mit Werbung für ein Programm zu versorgen
- enthält Programm-Features, die noch nicht im Programm integriert sind
- ist ein fabelhaftes Versprechen, was ein Programm alles für Funktionen hat
- ist keine Anleitung zu dem Programm

Meinungsumfage

Eine Manpage ...

- ... ist die *Single Source of Truth* um zu wissen, was ein Programm kann, bzw. nicht kann
- stellt die ideale Funktionsweise einen Programms dar
- ist der perfekte Ort um CLI-Nerds (RTFM) mit Werbung für ein Programm zu versorgen
- enthält Programm-Features, die noch nicht im Programm integriert sind
- ist ein fabelhaftes Versprechen, was ein Programm alles für Funktionen hat
- ist keine Anleitung zu dem Programm

crypto-policy Manpage

Listing: Auszug aus der Manpage crypto-policies

```
1   PROVIDED POLICIES
2      DEFAULT
3           The DEFAULT policy is a reasonable default policy for today's standards. It allows
            ↪  the TLS 1.2 and TLS 1.3 protocols, as well as IKEv2 and SSH2. The RSA and
            ↪  Diffie-Hellman parameters are accepted if larger than 2047 bits.
4           The level provides at least 112-bit security with the exception of SHA-1 signatures
            ↪  needed for DNSSec and other still prevalent legacy use of SHA-1 signatures.
5           - MACs: all HMAC with SHA-1 or better + all modern MACs (Poly1305 etc.)
6           - Curves: all prime >= 255 bits (including Bernstein curves)
7           - Signature algorithms: with SHA-1 hash or better (no DSA)
8           - TLS Ciphers: >= 128-bit key, >= 128-bit block (AES, ChaCha20, including AES-CBC)
9           - non-TLS Ciphers: as TLS Ciphers with added Camellia
10          - key exchange: ECDHE, RSA, DHE (no DHE-DSS)
11          - DH params size: >= 2048
12          - RSA keys size: >= 2048
13          - TLS protocols: TLS >= 1.2, DTLS >= 1.2
```

Überprüfen wir das mal . . .

Versprechen oder Werbung?

Beweis:

AlmaLinux8 hat noch eine ältere SSH Version. Diese kennt die Option `RequiredRSASize` noch nicht.
Die Option `min_rsa_size` in der Crypto-Policy für OpenSSH bezieht sich auf die SSH Option `RequiredRSASize` und ist in AlmaLinux8 somit wirkungslos.

Listing: Auszug aus der SSH Manpage

```
RequiredRSASize
Specifies the minimum RSA key size (in bits) that sshd(8) will accept. User and host-based
authentication keys smaller than this limit will be refused. The default is 1024 bits. Note
that this limit may only be raised from the default.
```

Policy-Definition „from Scratch“

- im Ordner `/etc/crypto-policies/policies` oder `/usr/share/crypto-policies/policies`
- Dateiextension ist `.pol`
- Dateiname in Großbuchstaben, z.B. `EXAMPLE.pol`

Policy „from Scratch“

Listing: Beispiel FIPS Policy (stark gekürzt)

```
mac = AEAD HMAC-SHA2-256 HMAC-SHA2-384
group = SECP256R1 SECP384R1 SECP521R1 FFDHE-2048
hash = SHA2-256 SHA2-384 SHA2-512 SHA2-224
sign = ECDSA-SHA3-256 ECDSA-SHA2-256
cipher = AES-256-GCM AES-256-CCM AES-256-CTR
cipher@TLS = AES-256-GCM AES-256-CCM AES-128-GCM
# Kerberos is an exception,
#allow CBC CTS ciphers no other options
cipher@Kerberos = AES-256-CBC AES-128-CBC
key_exchange = ECDHE DHE DHE-RSA PSK DHE-PSK
protocol@TLS = TLS1.3 TLS1.2 DTLS1.2
protocol@IKE = IKEv2
# Parameter sizes
min_dh_size = 2048
min_dsa_size = 2048  # DSA is disabled
min_rsa_size = 2048
# GnuTLS only for now
sha1_in_certs = 0
arbitrary_dh_groups = 1
ssh_certs = 1
ssh_etm = 1
```

Crypto-Policy „from Scratch“ – Konfigurations-Parameter

Listing: Beispiel FIPS Policy (stark gekürzt)

```
1  mac = AEAD HMAC-SHA2-256 HMAC-SHA2-384
2  group = SECP256R1 SECP384R1 SECP521R1 FFDHE-2048
3  hash = SHA2-256 SHA2-384 SHA2-512 SHA2-224
4  sign = ECDSA-SHA3-256 ECDSA-SHA2-256
5  cipher = AES-256-GCM AES-256-CCM AES-256-CTR
6  cipher@TLS = AES-256-GCM AES-256-CCM AES-128-GCM
7  cipher@Kerberos = AES-256-CBC AES-128-CBC
8  key_exchange = ECDHE DHE DHE-RSA PSK DHE-PSK
9  protocol@TLS = TLS1.3 TLS1.2 DTLS1.2
10 protocol@IKE = IKEv2
11 min_dh_size = 2048
12 min_dsa_size = 2048  # DSA is disabled
13 min_rsa_size = 2048
14 sha1_in_certs = 0
15 arbitrary_dh_groups = 1
16 ssh_certs = 1
17 etm@SSH = ANY
```

`mac`: MAC-Algorithmen

`group`: Gruppen oder Elliptische Kurven für Schlüsseltausch

`hash`: kryptographische Hashes (Message Digest)

`sign`: Signaturen

`cipher`: symmetrische Verschlüsselungsalgorithmen (inkl. Modes)

`key_exchange`: Algorithmen für Schlüsseltausch

`protocol`: TLS, DTLS und IKE Protokoll Versionen; abhängig von Backends

Crypto-Policy „from Scratch“ – Konfigurations-Parameter

Listing: Beispiel FIPS Policy (stark gekürzt)

```
1  mac = AEAD HMAC-SHA2-256 HMAC-SHA2-384
2  group = SECP256R1 SECP384R1 SECP521R1 FFDHE-2048
3  hash = SHA2-256 SHA2-384 SHA2-512 SHA2-224
4  sign = ECDSA-SHA3-256 ECDSA-SHA2-256
5  cipher = AES-256-GCM AES-256-CCM AES-256-CTR
6  cipher@TLS = AES-256-GCM AES-256-CCM AES-128-GCM
7  cipher@Kerberos = AES-256-CBC AES-128-CBC
8  key_exchange = ECDHE DHE DHE-RSA PSK DHE-PSK
9  protocol@TLS = TLS1.3 TLS1.2 DTLS1.2
10 protocol@IKE = IKEv2
11 min_dh_size = 2048
12 min_dsa_size = 2048  # DSA is disabled
13 min_rsa_size = 2048
14 sha1_in_certs = 0
15 arbitrary_dh_groups = 1
16 ssh_certs = 1
17 etm@SSH = ANY
```

`min_dh_size`: minimale DH-Schlüsselgröße

`min_dsa_size`: minimale DSA-Schlüsselgröße

`min_rsa_size`: minimale RSA-Schlüsselgröße

Crypto-Policy „from Scratch“ – Konfigurations-Parameter

Listing: Beispiel FIPS Policy (stark gekürzt)

```
1  mac = AEAD HMAC-SHA2-256 HMAC-SHA2-384
2  group = SECP256R1 SECP384R1 SECP521R1 FFDHE-2048
3  hash = SHA2-256 SHA2-384 SHA2-512 SHA2-224
4  sign = ECDSA-SHA3-256 ECDSA-SHA2-256
5  cipher = AES-256-GCM AES-256-CCM AES-256-CTR
6  cipher@TLS = AES-256-GCM AES-256-CCM AES-128-GCM
7  cipher@Kerberos = AES-256-CBC AES-128-CBC
8  key_exchange = ECDHE DHE DHE-RSA PSK DHE-PSK
9  protocol@TLS = TLS1.3 TLS1.2 DTLS1.2
10 protocol@IKE = IKEv2
11 min_dh_size = 2048
12 min_dsa_size = 2048  # DSA is disabled
13 min_rsa_size = 2048
14 sha1_in_certs = 0
15 arbitrary_dh_groups = 1
16 ssh_certs = 1
17 etm@SSH = ANY
```

- binäre Werte (Default:1):

  `sha1_in_certs`: SHA1 erlaubt in Zertifikats-Signaturen

  `arbitrary_dh_groups`: beliebige Gruppe in Diffie-Hellman erlaubt

  `ssh_certs`: OpenSSH Zertifikatsauthentifizierung erlaubt

Crypto-Policy „from Scratch“ – Konfigurations-Parameter

Listing: Beispiel FIPS Policy (stark gekürzt)

```
1  mac = AEAD HMAC-SHA2-256 HMAC-SHA2-384
2  group = SECP256R1 SECP384R1 SECP521R1 FFDHE-2048
3  hash = SHA2-256 SHA2-384 SHA2-512 SHA2-224
4  sign = ECDSA-SHA3-256 ECDSA-SHA2-256
5  cipher = AES-256-GCM AES-256-CCM AES-256-CTR
6  cipher@TLS = AES-256-GCM AES-256-CCM AES-128-GCM
7  cipher@Kerberos = AES-256-CBC AES-128-CBC
8  key_exchange = ECDHE DHE DHE-RSA PSK DHE-PSK
9  protocol@TLS = TLS1.3 TLS1.2 DTLS1.2
10 protocol@IKE = IKEv2
11 min_dh_size = 2048
12 min_dsa_size = 2048  # DSA is disabled
13 min_rsa_size = 2048
14 sha1_in_certs = 0
15 arbitrary_dh_groups = 1
16 ssh_certs = 1
17 etm@SSH = ANY
```

`etm@SSH`

`ANY/DISABLE_ETM/DISABLE_NON_`

nur im Kontext von OpenSSH(Scope): Encrypt-then-Mac und Encrypt-and-Mac

Crypto-Policy „from Scratch“ – Konfigurations-Parameter

Listing: Beispiel FIPS Policy (stark gekürzt)

```
mac = AEAD HMAC-SHA2-256 HMAC-SHA2-384
group = SECP256R1 SECP384R1 SECP521R1 FFDHE-2048
hash = SHA2-256 SHA2-384 SHA2-512 SHA2-224
sign = ECDSA-SHA3-256 ECDSA-SHA2-256
cipher = AES-256-GCM AES-256-CCM AES-256-CTR
cipher@TLS = AES-256-GCM AES-256-CCM AES-128-GCM
cipher@Kerberos = AES-256-CBC AES-128-CBC
key_exchange = ECDHE DHE DHE-RSA PSK DHE-PSK
protocol@TLS = TLS1.3 TLS1.2 DTLS1.2
protocol@IKE = IKEv2
min_dh_size = 2048
min_dsa_size = 2048  # DSA is disabled
min_rsa_size = 2048
sha1_in_certs = 0
arbitrary_dh_groups = 1
ssh_certs = 1
etm@SSH = ANY
```

Scopes:

- `option@scope`
- Eingrenzen der Backends
- bevorzugte Schreibweise
- case-insensitive
- Negierung mit `option@!scope`

Mit Modulen bestehende Policies erweitern

- im Ordner `/etc/crypto-policies/policies/modules` oder `/usr/share/crypto-policies/policies/modules`
- Dateiname in Großbuchstaben, z.B. `NO-SHA1.pmod`
- Dateiextension ist `.pmod`
- Konfiguration erlaubt (noch) vollständiges Überschreiben der Key-Exchange Parameter (von ECDHE-Keys)

Policy-Module

Listing: Beispiel AES-128-Module

```
1  # Disable the AES-128 cipher, all modes
2  cipher = -AES-128-*
3  # Disable CHACHA20-POLY1305 for the TLS protocol (OpenSSL, GnuTLS, NSS, and OpenJDK)
4  cipher@TLS = -CHACHA20-POLY1305
5  # Allow using the FFDHE-1024 group with the SSH protocol (libssh and OpenSSH)
6  group@SSH = FFDHE-1024+
7  # Disable all CBC mode ciphers for the SSH protocol (libssh and OpenSSH)
8  cipher@SSH = -*-CBC
9  # Allow the AES-256-CBC cipher in applications using libssh
10 cipher@libssh = AES-256-CBC+
```

- * ist Wildcard
- + fügt Parameter hinzu
- - (Minus) Parameter entfernen

Policy-Module

Listing: Beispiel AES-128-Module

```
1  # Disable the AES-128 cipher, all modes
2  cipher = -AES-128-*
3  # Disable CHACHA20-POLY1305 for the TLS protocol (OpenSSL, GnuTLS, NSS, and OpenJDK)
4  cipher@TLS = -CHACHA20-POLY1305
5  # Allow using the FFDHE-1024 group with the SSH protocol (libssh and OpenSSH)
6  group@SSH = FFDHE-1024+
7  # Disable all CBC mode ciphers for the SSH protocol (libssh and OpenSSH)
8  cipher@SSH = -*-CBC
9  # Allow the AES-256-CBC cipher in applications using libssh
10 cipher@libssh = AES-256-CBC+
```

- * ist Wildcard
- + fügt Parameter hinzu
- - (Minus) Parameter entfernen

Policy-Module

Listing: Beispiel AES-128-Module

```
1  # Disable the AES-128 cipher, all modes
2  cipher = -AES-128-*
3  # Disable CHACHA20-POLY1305 for the TLS protocol (OpenSSL, GnuTLS, NSS, and OpenJDK)
4  cipher@TLS = -CHACHA20-POLY1305
5  # Allow using the FFDHE-1024 group with the SSH protocol (libssh and OpenSSH)
6  group@SSH = FFDHE-1024+
7  # Disable all CBC mode ciphers for the SSH protocol (libssh and OpenSSH)
8  cipher@SSH = -*-CBC
9  # Allow the AES-256-CBC cipher in applications using libssh
10 cipher@libssh = AES-256-CBC+
```

- * ist Wildcard
- + fügt Parameter hinzu
- - (Minus) Parameter entfernen

Ansible-Role: Crypto-Policies

Rolle: `linux-system-roles.crypto_policies`

`https://galaxy.ansible.com/linux-system-roles/crypto_policies`

Variablen:

`crypto_policies_policy` Spezifizierung der Policy und Module

`crypto_policies_available_policies` Liste der vorhandenen Policies

`crypto_policies_available_subpolicies` Liste der vorhandenen Module

`crypto_policies_reload` Direkt nach dem Setzen der Policy die Services neu starten?

`crypto_policies_reboot_ok` System neu starten?

`crypto_policies_reboot_required` wird von der Rolle gesetzt wenn Neustart des Systems erforderlich

Ansible-Role: Crypto-Policies

- kann in RHEL über rpm-Paket `rhel-system-roles` installiert werden
- was die Rolle noch nicht kann:
  - Customized Module erstellen
  - Customized Policies erstellen

Ansible-Role: Crypto-Policies

Listing: Playbook zum Setzen der Policy inklusive Module

```
1 - name: Manage crypto policies
2   hosts: all
3   roles:
4     role: linux-system-roles.crypto_policies
5     vars:
6       crypto_policies_policy: "DEFAULT:NO-SHA1"
7       crypto_policies_reload: false
```

Ist das gleiche wie:

Listing: Befehl zum Setzen der Policy inklusive Module

```
update-crypto-policies --set DEFAULT:NO-SHA1
```

Ressourcen zum Recherchieren

- `man crypto-policies`
- `man update-crypto-policies`
- Vorträge:
  - `https://www.youtube.com/watch?v=NLSm8Kqd5N0`
  - `https://ftp.belnet.be/mirror/FOSDEM/video/2020/UA2.114/security_custom_crypto_policies.webm`
- interaktives Lab:

`https://www.redhat.com/en/interactive-labs/customize-system-wide-cryptographic-policy`

- Entwicklungs-Repo:

`https://gitlab.com/redhat-crypto/fedora-crypto-policies/`

Vielen Dank für Ihre Aufmerksamkeit!

Bei weiteren Fragen wenden Sie sich bitte an info@b1-systems.de oder +49 (0)8457 - 931096