Crypto-Policies

Was ist das?

Susanne Schütze
ratten@buecherratten.in-berlin.de

Chemnitzer Linux Tage 2024

17. März 2024

Bild von Pete Linforth auf Pixabay

Inhalt

weitere Informationen:

Inhalt

- Abfrage: Wer von euch musste letztens die Probleme von den Kollegen lösen?

Die Fehler Beschreibung

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.

Ich schau es mir an . . .

Aber deine Lösung muss idempotent sein und mit Ansible umsetzbar

. . .

whoami

- Susanne Schütze
- 40 Jahre
- Pronomen: sie
- Berufsbezeichnung: Fachinformatikerin für Systemintegration
- Berufliches Themenfeld: Automatisierung mit Ansible
- Linuxerin seit Kernel 2.6.24 (2009)
- Zugehörigkeiten zu: Haecksen, LinuxWorks!, BeLUG, FSFE

URL zu Folien und Handout:

`http://git.tuxteam.de/gitweb/?p=susannes-git/Crypto-Policy-Vortrag.git;a=tree`

Kryptographie im System bisheriger Stand

Abfrage

Wer von euch:

- hat den Überblick über alle Crypto-Konfigurationsmöglichkeiten?
- ist der Meinung, das die Konfigurations-Optionen bezüglich Crypto einheitlich sind?
- findet es easy mal eben Systemweit zB SHA1 auszuschalten?

weitere Informationen:

Kryptographie im System bisheriger Stand

Abfrage

- Bitte um Handzeichen

Kryptographie im System bisheriger Stand

- jedes Tool hat eigene Crypto-Regeln
- Crypto-Regeln in der Konfigurationsdatei des Tools definieren

Beispiel: SSH /etc/ssh/sshd.conf

```
1 Ciphers aes128-ctr,aes192-ctr,aes256-ctr
2 HostKeyAlgorithms
  ↪  ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-s
3 KexAlgorithms
  ↪  ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2
4 MACs hmac-sha2-256,hmac-sha2-512,hmac-sha1
```

weitere Informationen:

Kryptographie im System bisheriger Stand

- Wenn zB. ssh auf SHA1-Algorithmen verzichtet, bedeutet das nicht das OpenSSL, davon weiß und auch darauf verzichtet

Debugging des SSH-Fehlers

Was läuft hier?

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

```
[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
           \-669 /usr/sbin/sshd -D
           ↪  -oCiphers=aes256-gcm@openssh.com,aes256-ct
```

Debugging des SSH-Fehlers

die mysteriöse Variable

- die 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/opensshse
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
```

weitere Informationen:

Debugging des SSH-Fehlers
die mysteriöse Variable

- Realisiert werden Policies darüber, das die Variabel CRYPTOPOLICY beim Aufruf der Systemd Unit gefüllt werden

Einführung Crypto-Policies

Idee: ein Ort für Systemweite Crypto-Einstellungen

- alle Crypto über ein systemweites Tool eingestellen
- 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 Sicherheitsstandardisierungen entsprechen sollen
- bisher in Fedora, RHEL, CentOS, OpenSuse, Oracle Linux, Ubuntu, Debian-sid(testing), ...
- Entwicklung:
  `https://gitlab.com/redhat-crypto/fedora-crypto-policies`
- Eigenentwicklung von RedHat, Idee findet jedoch Konsens in Community

weitere Informationen:

Einführung Crypto-Policies

Idee: ein Ort für Systemweite Crypto-Einstellungen

- Cryptopolicies werden über Python umgesetzt

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

Grob Überblick

LEGACY	kompatibel mit RHEL 5
FUTURE	Vorhersage zu zukünftigen Bedrohungen *1
BSI	nach BSI Standardisierung TR-02102-2 (bisher erst in Fedora39) *2
FIPS	genügt FIPS 140 Anforderungen *3
DEFAULT	
EMPTY	für Debugging deaktiviert alle Crypto

*1 fun fact: Die RedHat Customer Portal API kann zur Zeit noch nicht mit Future.

*2 `https://gitlab.com/redhat-crypto/fedora-crypto-policies/-/blob/master/policies/BSI.pol`

weitere Informationen:

Arten von Policies

Grob Überblick

- *[3]Amerikanische Zertifizierung für Kryptographie bezogen auf Kryptographische Teile des Produkts. Gegensatz zu CC (Common Criteria for Information Technology Security Evaluation) bezogen auf Sicherheitsbezogene Themen (ISO 15408)

Welche Policies gibt es?

LEGACY

LEGACY

- maximale Kompatibilität mit älteren Systemen
- weniger sicher
- Support für TLS 1.0, TLS 1.1, und SSH2
- erlaubt DSA und 3DES
- RSA and Diffie-Hellman Parameter $>$ 1024 Bit
- mindestens 64-bit Sicherheit

MACs:	all HMAC with SHA-1 or $>$ + all modern MACs (Poly1305 etc.)
Curves:	all prime $>=$ 255 bits (including Bernstein curves)
Signature algorithms:	with SHA1 hash or better (DSA allowed)
TLS Ciphers:	all $>=$ 112-bit key, $>=$ 128-bit block (incl. 3DES, no RC4)
Non-TLS Ciphers:	same as TLS ciphers with added Camellia
Key exchange:	ECDHE, RSA, DHE
DH params size:	$>=$ 1023
RSA keys size:	$>=$ 1023
DSA params size:	$>=$ 1023
TLS protocols:	TLS $>=$ 1.0, DTLS $>=$ 1.0

Welche Policies gibt es?

DEFAULT

DEFAULT

- ist heutiger Standard
- erlaubt TLS 1.2, TLS 1.3, IKEv2 und SSH2
- akzeptiert Diffie-Hellman Parameter $>$ 2048 Bits
- mindestens 112-Bit Sicherheit
- Ausnahmsweise sind SHA-1 Signaturen in DNSSec erlaubt

MACs:	all HMAC with SHA-1 or better + all modern MACs (Poly1305 etc.)
Curves:	all prime >= 255 bits (including Bernstein curves)
Signature algorithms:	with SHA-1 hash or better (no DSA)
TLS Ciphers:	>= 128-bit key, >= 128-bit block (AES, ChaCha20, including AES-CBC)
non-TLS Ciphers:	as TLS Ciphers with added Camellia
key exchange:	ECDHE, RSA, DHE (no DHE-DSS)
DH params size:	>= 2048
RSA keys size:	>= 2048
TLS protocols:	TLS >= 1.2, DTLS >= 1.2

Welche Policies gibt es?
FUTURE

FUTURE

- Konservative Policy
- Vermutung, das sie zukünftigen Angriffen widersteht, auf Kosten der Kompatibilität
- es kann Kommunikation mit vielen Systemen verhindern, die schwächere Kryptographie verwenden
- es sind keine SHA-1 in Signaturen erlaubt
- es wird eine (unvollständige) Post-Quanten Kryptographie unterstützt
- Unterstützung von 256-Bit symmetrischer Verschlüsselung
- RSA und Diffie-Hellman Parameter länger als 3071 Bits
- mindestens 128-Bit Sicherheit

MACs:	all HMAC with SHA-256 or $>$ + all modern MACs (Poly1305 etc.)
Curves:	all prime $>=$ 255 bits (including Bernstein curves)
Signature algorithms:	with SHA-256 hash or better (no DSA)
TLS Ciphers:	$>=$ 256-bit key, $>=$ 128-bit block, only Authenticated Encryption (AE) ciphers, no CBC ciphers
non-TLS Ciphers:	same as TLS ciphers with added non AE ciphers, CBC ones enabled only in Kerberos
key exchange:	ECDHE, DHE (no DHE-DSS, no RSA)
DH params size:	$>=$ 3072
RSA keys size:	$>=$ 3072
TLS protocols:	TLS $>=$ 1.2, DTLS $>=$ 1.2

Welche Policies gibt es?
FIPS

FIPS

- Kompatible zu den FIPS 140-2 Voraussetzungen
- wird für fips-mode-setup verwendet
- bietet mindestens 112-Bit Sicherheit

MACs:	all HMAC with SHA1 or better
Curves:	all prime >= 256 bits
Signature algorithms:	with SHA-256 hash or better (no DSA)
TLS Ciphers:	>= 128-bit key, >= 128-bit block (AES, including AES-CBC)
non-TLS Ciphers:	same as TLS Ciphers
key exchange:	ECDHE, DHE (no DHE-DSS, no RSA)
DH params size:	>= 2048
RSA params size:	>= 2048
TLS protocols:	TLS >= 1.2, DTLS >= 1.2

Die BSI-Policy

BSI

- Author: Marcus Meissner von OpenSuse
- Grundlage des BSI Standards TR 02102 `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.html`
- Empfehlungen des BSI werden regelmäßig geupdated
- unvollständige Unterstützung für Post-Quanten Kryptographie
- 128 Bit Sicherheit (außer RSA)
- erlaubt kein SHA1 (außer DNSSEC und RPM)
- Beachtet Chacha20 und Camellia werden nicht vom BSI empfohlen

MACs:	all HMAC with SHA-256 or better + all modern MACs
Curves:	all prime >= 255 bits (including Bernstein curves)
Signature algorithms:	with SHA-256 hash or better (no DSA)
TLS Ciphers:	>= 256-bit key, >= 128-bit block, only Authenticated
Encryption	(AE) ciphers
non-TLS Ciphers:	same as TLS ciphers with added non AE ciphers
key exchange:	ECDHE, DHE (no DHE-DSS, no RSA)
DH params size:	>= 3072
RSA keys size:	>= 2048 (until end of 2023, then it will switch to 3072)
TLS protocols:	TLS >= 1.2, DTLS >= 1.2

Welche Policies gibt es?

NEXT und EMPTY

NEXT

nur für Fedora, ähnlich wie RHEL-8 Default, daher Alias für Default

EMPTY

Alle Kryptographischen Algorithmen sind deaktiviert, soll nur für Debugging genutzt werden

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
  - sind symbolische links von /etc/crypto-policies/back-ends nach /usr/share/crypto-policies.
  - generiert Backend Konfigurations-Dateien
- deaktivieren
  - Bei SSH über eine Variable in der Konfigurationsdatei (sshd.config) 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

das Backend bearbeiten

- `update-crypto-policies --show`
- FIPS
- 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,aes25
  ↪ 6-ctr,aes256-cbc,aes128-gcm@openssh.com,aes128-ct
  ↪ r,aes128-cbc
  ↪ -oMACs=hmac-sha2-256-etm@openssh.com,hmac-sha1-et
  ↪ m@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-
  ↪ sha2-256,hmac-sha1,hmac-sha2-512
  ↪ -oGSSAPIKeyExchange=no -oKexAlgorithms=ecdh-sha2-
  ↪ nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,di
  ↪ ffie-hellman-group-exchange-sha256,diffie-hellman
  ↪ -group14-sha256,diffie-hellman-group16-sha512,dif
  ↪ fie-hellman-group18-sha512
  ↪ -oHostKeyAlgorithms=ecdsa-sha2-nistp256,ecdsa-sha
  ↪ 2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp3
```

weitere Informationen:

erste Lösung

das Backend bearbeiten

- Einstellungen im Backend halten bis zum Reboot / Sinnvoll wäre gewesen in die Manpage zu schauen!

Umfrage

Eurer Meinung nach...

Eine Man Page

- 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

weitere Informationen:

Umfrage

Eurer Meinung nach...

- Bitte um Handzeichen

Crypto-Policy Man-Page

```
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 . . .

ssh-key-size

unter alma8

```
sibille@Libelle:~$ ssh-copy-id -i ~/.ssh/crypt_1024rsa.pub root@crypt-arbeit8
Number of key(s) added: 1
sibille@Libelle:~$ ssh-copy-id -i ~/.ssh/crypt_2048rsa.pub root@crypt-arbeit8
Number of key(s) added: 1
sibille@Libelle:~$ ssh root@crypt-arbeit8
root@crypt-arbeit8:# update-crypto-policies --show
LEGACY
root@crypt-arbeit8:# update-crypto-policies --set Default
 Setting system policy to DEFAULT
root@crypt-arbeit8:# reboot
sibille@Libelle:~$ ssh -i .ssh/crypt_1024rsa root@crypt-arbeit8 -v
debug1: Local version string SSH-2.0-OpenSSH_8.4p1 Debian-5+deb11u1
debug1: Remote protocol version 2.0, remote software version OpenSSH_8.0
debug1: Next authentication method: publickey
debug1: Offering public key: .ssh/crypt_1024rsa RSA
  ↪  SHA256:hkpFBRW/y76PZlG903lf1POqZ90DQfFoRfpqFqD/BwY explicit
debug1: Server accepts key: .ssh/crypt_1024rsa RSA SHA256:hkpFBRW/y76PZlG903lf1POqZ90DQfFoR
debug1: Authentication succeeded (publickey).
Authenticated to crypt-arbeit8 ([192.168.2.38]:22).
root@crypt-arbeit8:~[root@crypt-arbeit8 ~]# exit
logout
```

ssh-key-size

unter alma9

```
sibille@Libelle:~$ ssh-copy-id -i ~/.ssh/crypt_1024rsa.pub root@crypt-arbeit9
Number of key(s) added: 1
sibille@Libelle:~$ ssh-copy-id -i ~/.ssh/crypt_2048rsa.pub root@crypt-arbeit9
Number of key(s) added: 1
sibille@Libelle:~$ ssh root@crypt-arbeit9
root@crypt-arbeit9:# update-crypto-policies --showypto-policies --show
LEGACY
root@crypt-arbeit9:# update-crypto-policies --set Defaultlicies --set Default
 Setting system policy to DEFAULT
root@crypt-arbeit9:# reboot
sibille@Libelle:~$ ssh -i .ssh/crypt_1024rsa root@crypt-arbeit9 -v
debug1: Local version string SSH-2.0-OpenSSH_8.4p1 Debian-5+deb11u1
debug1: Offering public key: .ssh/crypt_1024rsa RSA
  ↪  SHA256:hkpFBRW/y76PZlG903lf1POqZ90DQfFoRfpqFqD/BwY explicit
debug1: Authentications that can continue:
  ↪  publickey,gssapi-keyex,gssapi-with-mic,password
 debug1: Next authentication method: password
root@crypt-arbeit9's password:
sibille@Libelle:~$ ssh -i .ssh/crypt_2048rsa root@crypt-arbeit9 -v
debug1: Offering public key: .ssh/crypt_2048rsa RSA
  ↪  SHA256:g5pZruCXK0Ng2/xW37JDXqQrMCo/XLL4jfETuZNcMQs explicit
 debug1: Server accepts key: .ssh/crypt_2048rsa RSA SHA256:g5pZruCXK0Ng2/xW37JDXqQrMCo/XLL4j
debug1: Authentication succeeded (publickey).
Authenticated to crypt-arbeit9 ([192.168.2.107]:22).
```

Versprechung oder Werbung?

ssh-key-size

Beweis.

AlmaLinux8 hat noch eine ältere SSH Version, diese kennt die Option RequiredRSASize noch nicht.

Auszug aus der SSH manpage AlmaLinux9

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.

Dadurch sind die Optionen min_rsa_size in der Crypto-Policy für OpenSSH in AlmaLinux8 wirkungslos.

Nach Aussage von Red Hat, haben sie OpenSSH auf RHEL8 extra gepatched. . . Ich hab einen Patch in Fedora 37 gefunden, in den Paketen von AlmaLinux und Rocky Linux jedoch nicht.

weitere Informationen:

Versprechung oder Werbung?

ssh-key-size

- In Alma- und Rocky Linux tritt der gleiche Fehler auf, jedoch nicht in Fedora 37

übersicht über die Libaries und ihrer Scopes

- libssh SSH2 protocol implementation (scopes: libssh, SSH)
- sequoia PGP, outside of rpm-sequoia (scopes: sequoia)
- rpm-sequoia PGP backend (scopes: rpm, rpm-sequoia)
- BIND DNS (scopes: BIND, DNSSec)
- GnuTLS (scopes: GnuTLS, SSL, TLS)
- Kerberos 5 (scopes: krb5, Kerberos)
- Libreswan IPsec and IKE protocol implementation (scopes: libreswan, IPSec, IKE)
- NSS TLS library (scopes: NSS, SSL, TLS)
- OpenJDK runtime environment (scopes: java-tls, SSL, TLS)
- OpenSSH SSH2 (scopes: OpenSSH, SSH)
- OpenSSL TLS library (scopes: OpenSSL, SSL, TLS)

Deaktivierte Cipher suites

- DH with parameters $<$ 1024 bits
- RSA with key size $<$ 1024 bits
- Camellia
- RC4
- ARIA
- SEED
- IDEA
- TLS CBC mode ciphersuites using SHA-384 HMAC
- AES-CCM8
- all ECC curves incompatible with TLS 1.3, including secp256k1
- IKEv1

diese können jedoch aktiviert werden

Konfigurations-Dateien

/etc/crypto-policies/back-ends Back-End Config-Dateien, verlinkt zur Package Crypto-Policies, es sei den local.d -Konfiguration wurde hinzugefügt

/etc/crypto-policies/config Beinhaltet Namen der aktiven Crypto-Policies

/etc/crypto-policies/local.d weitere Konfiguration, entweder Package basierend, oder vom Admin erstellt. Die back-end.config wird angehängt, wie ausgeliefert, vom Packet.

/usr/share/crypto-policies/policies Policy Definitions-Datei

/usr/share/crypto-policies/policies/modules Unter Definitions-Dateien

/etc/crypto-policies/policies Änderungen der Policy-Definitionen des System Admins

/etc/crypto-policies/policies/modules Änderungen der Modul-Definitionen des System Admins

/usr/share/crypto-policies/<POLICYNAME> generierte Datei des back-ends für die Policy POLICYNAME

Policy Definition hand made

- im Ordner /etc/crypto-policies/policies oder /usr/share/crypto-policies/policies
- Dateiname muss GROSS geschrieben werden
- Dateiextension ist .pol

Hand made Crypto Policy

Konfigurations-Parameter

möglich Parameter innerhalb von Konfigurationsdateien:

- Liste erlaubter:

mac:	MAC Algorithmen
group:	Gruppen oder elliptic curves für key exchanges
hash:	cryptographic hash (message digest)
sign:	signature
cipher:	symmetric encryption Algorithmen(inkl. modes)
key_exchange:	key exchange algorithms
protocol:	TLS, DTLS and IKE Protokoll Versions. einige Backends erlauben kein selektives deaktivieren von Protokoll Versionen

- minimale Anzahl der Bits für:

min_dh_size:	parameters for DH key exchange
min_dsa_size:	DSA keys
min_rsa_size:	RSA keys

- binärer Werte:

sha1_in_certs:	1 SHA1 erlaubt in certificate signatures
arbitrary_dh_groups:	1 arbitrary group in Diffie-Hellman erlaubt
ssh_certs:	1 OpenSSH certificate authentication erlaubt,

Konfigurations-Parameter

auf Bereiche eingrenzen

Konfiguration auf bestimmte Backends (scopes) eingrenzen

- option@scope1,scope2 = ...
  - `cipher@SSH = -*-CBC`
- Negierung mit option@!scope
- scope ist case-insensitive
- scopes sind bevorzugte Schreibweise
- die Reihenfolge der Crypto ist wichtig, was zuerst steht wird priorisiert
- Reihenfolge der Optionen ist vorgegeben

wichtige scopes für SSH:

- OpenSSH SSH2 (scopes: OpenSSH, SSH)
- libssh SSH2 protocol implementation (scopes: libssh, SSH)

policy from scratch

Beispiel FIPS (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
```

Mit Modulen bestehende Policies erweitern

- im Ordner /etc/crypto-policies/policies/modules oder /usr/share/crypto-policies/policies/modules
- Dateiname muss GROSS geschrieben werden
- Dateiextension ist .pmod
- mit - (minus) Parameter entfernen
- * ist Wildcard
- Konfiguration erlaubt (noch) vollständiges Überschreiben der key-exchange Parameter (von ECDHE keys)

Policy-Module

Beispiel AES-128-Module

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

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 vorhanden Module

crypto_policies_reload ob direkt nach dem Setzen der Policy die Services neu gestartet werden

crypto_policies_reboot_ok ob das System neu gestartet wird

crypto_policies_reboot_required wird von der Rolle gesetzt wenn Neustart des Systems erforderlich

kann in RHEL über rpm-Paket installiert werden ()

Was die Rolle noch nicht kann:

- Customized Module erstellen
- Customized Policies erstellen

Ansible-Role: Crypto-Policies

Beispiel-Playbook

```
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:

```
1 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/`

Ende

Danke fürs zuhören und mitmachen!
Gibt es noch Fragen?

Bücherratten
ratten@buecherratten.in-berlin.de

URL zu Folien und Handout:

`http://git.tuxteam.de/gitweb/?p=susannes-git/Crypto-Policy-Vortrag.git;a=tree`