NETWORK-BOUND DISK ENCRYPTION

NYRHUG Monthly Meeting - February 2020

Patrick Ladd
Technical Account Manager - FSI
pladd@redhat.com
http://people.redhat.com/pladd
Booting...

Disk Password: 

HOW DO WE AUTOMATE?
Shh... I'm Secret!
Encryption Key

Shh... I'm Secret!
Encryption Key

Key Encryption Key

Shh... I'm Secret!
Encryption Key

Key Encryption Key

"correct battery horse staple"

Shh... I'm Secret!
K e y  E n c r y p t i o n  K e y

E n c r y p t i o n  K e y

S h h . . . I ' m  S e c r e t !

"c o r r e c t  b a t t e r y  h o r s e  s t a p l e"

S T A N D A R D  P A S S W O R D  M O D E L

Encryption Key

Key Encryption Key

Shh... I'm Secret!
STANDARD ESCROW MODEL?

Encryption Key

Key Encryption Key

"d41d8cd9...ecf8427e"

Shh... I'm Secret!

Escrow
S T A N D A R D   E S C R O W   M O D E L ?

Encryption Key

Key Encryption Key

"d41d8cd9...ecf8427e"

Shh... I'm Secret!

Escrow

TLS / GSSAPI
STANDARD ESCROW MODEL?

Encryption Key

Key Encryption Key

"d41d8cd9...ecf8427e"

Shh... I'm Secret!

TLS / GSSAPI

Escrow
STANDARD ESCROW MODEL?

Encryption Key

Key Encryption Key

"d41d8cd9...ecf8427e"

Shh... I'm Secret!

TLS / GSSAPI

Escrow
K e y   E n c r y p t i o n   K e y

S h h . . .   I ' m   S e c r e t !

"d41d8cd9...ecf8427e"

S T A N D A R D   E S C R O W   M O D E L ?

Encryption Key

Key Encryption Key

Shh... I'm Secret!

KDC/CA

Escrow

TLS / GSSAPI
STANDARD ESCROW MODEL

Encryption Key

Key Encryption Key

"d41d8cd9...ecf8427e"

Shh... I'm Secret!

KDC/CA

Escrow

Backups

TLS / GSSAPI
STANDARD ESCROW MODEL

Encryption Key

Key Encryption Key
"d41d8cd9...ecf8427e"

Shh... I'm Secret!

KDC/CA

Escrow

Backups

TLS / GSSAPI

HEARTBLEED
LESSONS LEARNED

- Presuming TLS will protect key transfer is dangerous
- Complexity increases attack surface
- Escrows are difficult to deploy
- X.509 is hard to get right
ASYMMETRIC CRYPTO?
BRACE YOURSELVES,

MATH IS COMING
\( C \in_R [1, p - 1] \)
\( c = g^C \)
\( K = g^{SC} = sC \)

\( S \in_R [1, p - 1] \)
\( s = g^S \)
\( K = g^{CS} = cS \)
**BINDING WITH ECDH (INSECURE)**

**PROVISIONING**

\[
S \in_R [1, p - 1] \\
S = gS \\
\leftarrow s
\]

\[
C \in_R [1, p - 1] \\
c = gC \\
K = gSC = sC
\]

Discard: \( K, C \)

Retain: \( s, c \)

**RECOVERY**

\[
c \rightarrow \\
K = xS \\
\leftarrow K
\]

Weaknesses:

1. K is revealed to a passive attacker.
2. With c, the passive attacker can get K.
3. Server learns c and therefore K.

Resolved: c **MUST** be private
MCCALLUM-RELYEA KEY EXCHANGE

**PROVISIONING**

\[ S \in R[1, p-1] \]
\[ s = gS \]
\[ \leftarrow s \]

\[ C \in R[1, p-1] \]
\[ c = gC \]
\[ K = gSC = sC \]

Discard: K, C

Retain: s, c

**RECOVERY**

\[ E \in R[1, p-1] \]
\[ e = gE \]
\[ x = c + e \]
\[ \rightarrow x \]

\[ y = xS \]
\[ \leftarrow y \]

\[ K = y - sE \]

Because: \[ K = gCS + gES - gSE \]

To keep c private, e & E MUST be private.
Shh... I'm Secret!

Key Encryption Key

MR Exchange

Server
Encryption Key

Key Encryption Key

Shh... I'm Secret!

MR Exchange

Server

Crypto HW
<table>
<thead>
<tr>
<th>Property</th>
<th>Escrow</th>
<th>MR Exchange</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server presence during provisioning</td>
<td>Required</td>
<td>Optional</td>
</tr>
<tr>
<td>Server presence during recovery</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>Server knowledge of keys</td>
<td>Required</td>
<td>None</td>
</tr>
<tr>
<td>Key transfer</td>
<td>Required</td>
<td>None</td>
</tr>
<tr>
<td>Client authentication</td>
<td>Required</td>
<td>Optional</td>
</tr>
<tr>
<td>Transport encryption</td>
<td>Required</td>
<td>Optional</td>
</tr>
<tr>
<td>End-to-end Encryption</td>
<td>Difficult</td>
<td>Unneeded</td>
</tr>
</tbody>
</table>
TANG

- https://github.com/latchset/tang
- Server-side daemon
- Simple: HTTP + JOSE
- Fast (>2k req/sec)
- Extremely small
- Minimal dependencies
- RHEL 7.4
INSTALLING A TANG SERVER

$ sudo yum install tang
$ sudo systemctl enable --now tangd.socket
CLEVIS

- https://github.com/latchset/clevis/
- Decryption automation and policy framework
- Minimal dependencies
- Early boot integration
- GNOME integration
- RHEL 7.4
$ yum install clevis

$ echo redhat | clevis encrypt tang '{"url":"http://localhost"}' > mydata.jwe

The advertisement is signed with the following keys:

haD7Y­8VkAyJo6­vdZMrGQXCSfI

Do you wish to trust the advertisement? [yN] y

$ cat mydata.jwe

{"ciphertext":"­O59czAqybvxHdme2t3I5A", ...}

$ clevis decrypt < mydata.jwe

redhat

$ systemctl stop tangd.socket

$ clevis decrypt < mydata.jwe

$ echo $? 1

BASIC ENCRYPTION WITH TANG / LUKSV1
DISK BINDING WITH TANG

$ clevis bind luks -d /dev/sda1 tang '{"url":"http://tang.srv"}'
The advertisement is signed with the following keys:
   haD7Y-8VkAyJo6-vdZMrGQXCSfI

Do you wish to trust the advertisement? [yN] y
Enter passphrase for /dev/sda1:

$ luksmeta show -d /dev/sda1
0  active empty
1  active cb6e8904-81ff-40da-a84a-07ab9ab5715e
2  inactive empty
3  inactive empty
...

# For root volume unlocking at boot:
$ yum install clevis-dracut
$ dracut -f
$ reboot

# For removable storage GNOME unlocking:
$ yum install clevis-udisks2
**KNOWN ISSUES**

**dracut / NetworkManager integration problems**

- Active BZ with fix pending
- Shouldn’t hurt in most cases, but it looks bad

```bash
[root@nbde7 ~] $ ip route
default via 192.168.122.1 dev eth0
default via 192.168.122.1 dev eth0 proto dhcp metric 100
192.168.122.0/24 dev eth0 proto kernel scope link src 192.168.122.170
192.168.122.0/24 dev eth0 proto kernel scope link src 192.168.122.170 metric 100
```
FROM AUTOMATION TO POLICY
SHAMIR SECRET SHARING

threshold = ?
SHAMIR SECRET SHARING

threshold = ?

threshold = ?
SIMPLE LAPTOP

unlock?

threshold = 1

Admin Password

User Password

Use Admin Password

Use User Password

threshold = 1
AUTOMATED LAPTOP

unlock?

threshold = 1

Admin Password

Tang

User Password
HIGH SECURITY SYSTEM

unlock?

threshold = 2

User Password

User Password

User Password
COMPLEX LAPTOP POLICY

unlock? → QR Code
  thresh. = 1

SSS → TPM
  thresh. = 2

SSS → Password
  thresh. = 2

  → Yubikey
  → Tang
  → Bluetooth
BASIC SHAMIR'S WITH TANG

$ echo PT | clevis encrypt sss
\n{"pins": {"tang": [{"url": "http://a.tang.srv"}, {"url": "http://b.tang.srv"}], "t": 1} \n> out.jwe

The advertisement is signed with the following keys:
    haD7Y­8VKAyJo6­vdZMrGQXCSfI

Do you wish to trust the advertisement? [yN] y

The advertisement is signed with the following keys:
    Edp­ESShUx4_95kGt­DTsCBbPag

Do you wish to trust the advertisement? [yN] y

$ clevis decrypt < out.jwe
PT

# Bring Down Tang Server A
$ clevis decrypt < out.jwe
PT

# Bring Down Tang Server B
$ clevis decrypt < out.jwe
$ echo $?
1
EXPLORING THE ECOSYSTEM
**DEPENDENCY: JOSÉ**

- https://github.com/latchset/jose
- JSON Object Signing and Encryption
- C Library & Command Line Utility
- Bottom Line: User-Friendly, Standards Compliant Crypto

```
$ jose jwk gen -i '{"alg": "A128GCM"}' -o oct.jwk
$ jose jwk gen -i '{"alg": "RSA1_5"}' -o rsa.jwk
$ jose jwk gen -i '{"alg": "ES256"}' -o ec.jwk

$ echo hi | jose jwe enc -i -k rsa.pub.jwk -o msg.jwe
$ jose jwe dec -i msg.jwe -k rsa.jwk
hi
$ jose jwe dec -i msg.jwe -k oct.jwk
Decryption failed!

$ echo hi | jose jws sig -i -k ec.jwk -o msg.jws
$ jose jws ver -i msg.jws -k ec.pub.jwk
hi
$ jose jws ver -i msg.jws -k oct.jwk
No signatures validated!
```
DEPENDENCY: LUKSMETA

- https://github.com/latchset/luksmeta
- Store metadata in LUKSv1 header gap
- C library & Command Line Utility

```
$ echo hi | luksmeta save -d /dev/sdc1 -s 2 -u EC998562-B60D-47F0-A579-DCA8C12F5BF6

$ luksmeta load -d /dev/sdc1 -s 2 -u EC998562-B60D-47F0-A579-DCA8C12F5BF6
hi

$ luksmeta load -d /dev/sdc1 -s 2 -u 12618962-A1E5-48F1-B327-D7C60E20FC02
Slot contains different UUID
```
José

- PKCS#11 Support
- Python Bindings
- Additional crypto backends
- Additional algorithms

FUTURE FEATURES

Clevis

- Password Pin
- PKCS#11 Pin (including, in the future, TPM)
- Ext4 encryption support

Tang

- Binding IDs (Optional; sacrifices anonymity)
- Revocation (requires Binding IDs)
Device Selection
Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu’s “Begin Installation” button.

Local Standard Disks
20 GiB

0x1af4
vda / 20 GiB free

Disks left unselected here will not be touched.

Specialized & Network Disks

Add a disk...

Disks left unselected here will not be touched.

Storage Configuration
- Automatic
- Custom

I would like to make additional space available.

Encryption
- Encrypt my data. You'll set a passphrase next.

Full disk summary and boot loader...
DISK ENCRYPTION PASSPHRASE

You have chosen to encrypt some of your data. You will need to create a passphrase that you will use to access your data when you start your computer.

Passphrase: 

- No password supplied

Confirm:

Warning: You won’t be able to switch between keyboard layouts (from the default one) when you decrypt your disks after install.
RESOURCES

- RHEL 7:

- RHEL 8:

- Multi-device setup
  - https://access.redhat.com/articles/4500491
QUESTIONS?

Clevis
Pin
TANG