Hardware backed SSH keys: ssh-tpm-agent

Morten Linderud

FOSDEM 2025

$ whoami

- Morten Linderud
  - Foxboron
- F/OSS developer since ~2013
- Arch Linux Developer since ~2016
- Hackeriet
- Devops at NRK

Hardware backed keys?

Key Compromises

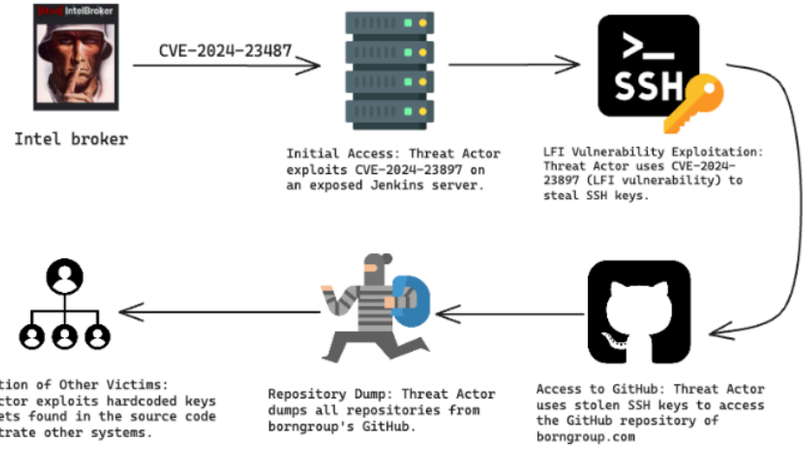
Key Compromises

- Not limited to SSH keys
  - Cookies, access tokens etc...
- Impersonation
- Priviledge escalation

Born Group breach (2024)

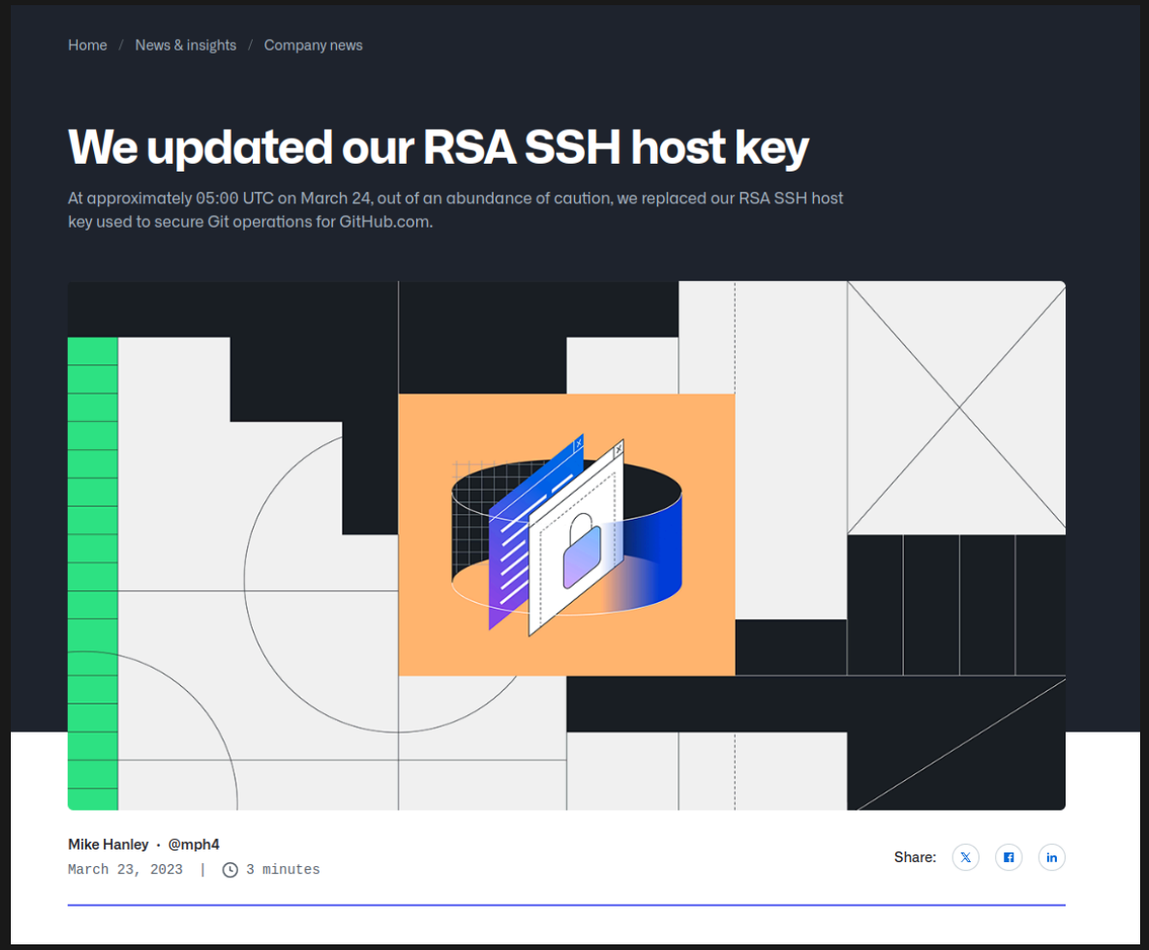
Sophisticated Attack on Born Group

Initial Access: Threat Actor exploits CVE-2024-23897 on an exposed Jenkins server.
LFI Vulnerability Exploitation: Threat Actor uses CVE-2024-23897 (LFI vulnerability) to steal SSH keys.
Access to GitHub: Threat Actor uses stolen SSH keys to access the GitHub repository of borngroup.com.
Repository Dump: Threat Actor dumps all repositories from BORN Group's GitHub.
Infiltration of Other Victims: Threat Actor exploits hardcoded keys and secrets found in the source code to infiltrate other systems.



BORN Group Supply Chain Breach: In-Depth Analysis of Intelbroker's Jenkins Exploitation

Github RSA ssh host key leak (2023)



Github Blog - We updated our RSA SSH host key

How do we solve this?

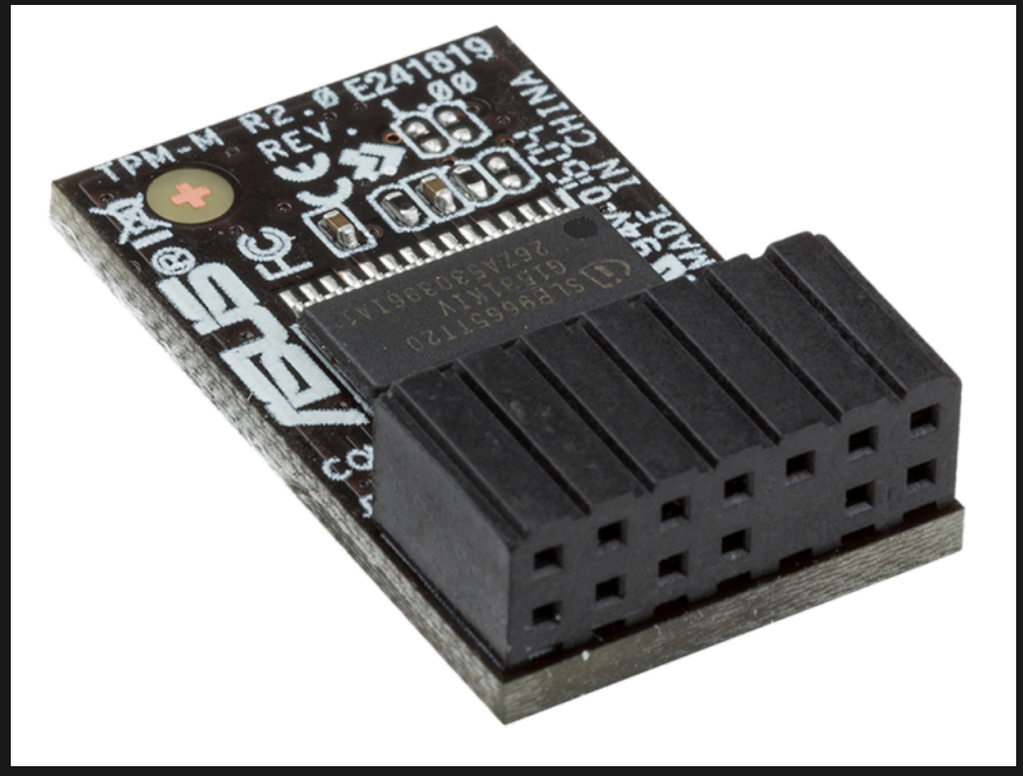
Yubikeys!

YES,

BUT

1 item	€50 EUR
Shipping & handling learn more	€0 EUR
VAT	€12.50 EUR
Grand total	**€62.50 EUR**

Trusted Platform Module (TPM)



What are they?

- Secure crypto processor
- Implementation:
  - Discrete TPM (dTPM)
  - Firmware TPM (fTPM)

Features

- Platform Integrity
- Hierarchies and keys
- Attestation

Platform Integrity

```
λ ~ » tpm2_pcrread
  sha256:
    0 : 0x6262A7D70F099FBE6A6B26BEA7B610D49C91FE218E83CEDF45605DAF8D5FB875
    1 : 0x878EDB17F96149EF540C9FA00912944B177DE77CFDD9D21AAD0325856D824275
    2 : 0x3D458CFE55CC03EA1F443F1562BEEC8DF51C75E14A9FCF9A7234A13F198E7969
    3 : 0x3D458CFE55CC03EA1F443F1562BEEC8DF51C75E14A9FCF9A7234A13F198E7969
    4 : 0x3C20C88A2D161B48FBE093DDBB46E6F5D76A893884704A8F6237065E0C974E66
    5 : 0x3BF951C4937BD85CFBBF5D00ECD3F83069E7696939CB7BDB8089AB6DA71338DE
    6 : 0x3D458CFE55CC03EA1F443F1562BEEC8DF51C75E14A9FCF9A7234A13F198E7969
    7 : 0x576DEE5AA15CC918AB56E3CB50091618388AA86F2D43252D7B9D31072538AE07
    8 : 0x0000000000000000000000000000000000000000000000000000000000000000
    9 : 0xBC2BF2C68444550684B86D50CF23639C93A51A68BD8681FBED181BBF35FD76AA
    10: 0x0000000000000000000000000000000000000000000000000000000000000000
    11: 0x9FDC3908055743F6B68FCF0D268B8E6627D2DAFF34C77323FDEC6D2AA062162C
    12: 0x54A5CF3DC7AABD28AD7AA66896FF25FAC2856D66D8982FFB2B346F3CB22FCA68
```

UAPI Group Specification - 🔏 Linux TPM PCR Registry 🗒️

Hierarchies and keys

- Embedded seeds to create keys
- Hierarchy seeds
  - Null
  - Owner
  - Endorsement

Hierarchies and keys

- Null
  - Session key
- Owner
  - Device owner key
- Endorsement
  - Lifetime of the device
  - Chains back to OEM

More on keys...

- Shielded keys!
- KDF under one of the hierarchy seeds
- Exported from the TPM and encrypted

Key creation and signing

```
$ tpm2_createprimary -C e -c primary.ctx
$ tpm2_create -G rsa -u rsa.pub -r rsa.priv -C primary.ctx
$ tpm2_load -C primary.ctx -u rsa.pub -r rsa.priv -c rsa.ctx
$ echo "my message" > message.dat
$ tpm2_sign -c rsa.ctx -g sha256 -o sig.rssa message.dat
$ tpm2_verifysignature -c rsa.ctx -g sha256 -s sig.rssa -m message.dat
```

Key creation and signing

```
$ tpm2_createprimary -C e -c primary.ctx
$ tpm2_create -G rsa -u rsa.pub -r rsa.priv -C primary.ctx
$ tpm2_load -C primary.ctx -u rsa.pub -r rsa.priv -c rsa.ctx
$ echo "my message" > message.dat
$ tpm2_sign -c rsa.ctx -g sha256 -o sig.rssa message.dat
$ tpm2_verifysignature -c rsa.ctx -g sha256 -s sig.rssa -m message.dat
```

Key creation and signing

```
$ tpm2_createprimary -C e -c primary.ctx
$ tpm2_create -G rsa -u rsa.pub -r rsa.priv -C primary.ctx
$ tpm2_load -C primary.ctx -u rsa.pub -r rsa.priv -c rsa.ctx
$ echo "my message" > message.dat
$ tpm2_sign -c rsa.ctx -g sha256 -o sig.rssa message.dat
$ tpm2_verifysignature -c rsa.ctx -g sha256 -s sig.rssa -m message.dat
```

Key creation and signing

```
$ tpm2_createprimary -C e -c primary.ctx
$ tpm2_create -G rsa -u rsa.pub -r rsa.priv -C primary.ctx
$ tpm2_load -C primary.ctx -u rsa.pub -r rsa.priv -c rsa.ctx
$ echo "my message" > message.dat
$ tpm2_sign -c rsa.ctx -g sha256 -o sig.rssa message.dat
$ tpm2_verifysignature -c rsa.ctx -g sha256 -s sig.rssa -m message.dat
```

Key creation and signing

```
1 $ tpm2_createprimary -C e -c primary.ctx
2 $ tpm2_create -G rsa -u rsa.pub -r rsa.priv -C primary.ctx
3 $ tpm2_load -C primary.ctx -u rsa.pub -r rsa.priv -c rsa.ctx
4 $ echo "my message" > message.dat
5 $ tpm2_sign -c rsa.ctx -g sha256 -o sig.rssa message.dat
6 $ tpm2_verifysignature -c rsa.ctx -g sha256 -s sig.rssa -m message.dat
```

Key creation and signing

```
$ tpm2_createprimary -C e -c primary.ctx
$ tpm2_create -G rsa -u rsa.pub -r rsa.priv -C primary.ctx
$ tpm2_load -C primary.ctx -u rsa.pub -r rsa.priv -c rsa.ctx
$ echo "my message" > message.dat
$ tpm2_sign -c rsa.ctx -g sha256 -o sig.rssa message.dat
$ tpm2_verifysignature -c rsa.ctx -g sha256 -s sig.rssa -m message.dat
```

Key creation and signing

```
$ tpm2_createprimary -C e -c primary.ctx
$ tpm2_create -G rsa -u rsa.pub -r rsa.priv -C primary.ctx
$ tpm2_load -C primary.ctx -u rsa.pub -r rsa.priv -c rsa.ctx
$ echo "my message" > message.dat
$ tpm2_sign -c rsa.ctx -g sha256 -o sig.rssa message.dat
$ tpm2_verifysignature -c rsa.ctx -g sha256 -s sig.rssa -m message.dat
```

Key creation and signing

```
$ tpm2_createprimary -C e -c primary.ctx
$ tpm2_create -G rsa -u rsa.pub -r rsa.priv -C primary.ctx
$ tpm2_load -C primary.ctx -u rsa.pub -r rsa.priv -c rsa.ctx
$ echo "my message" > message.dat
$ tpm2_sign -c rsa.ctx -g sha256 -o sig.rssa message.dat
$ tpm2_verifysignature -c rsa.ctx -g sha256 -s sig.rssa -m message.dat
```

TPM Policies

- Restrict usage
- Include system state
- Signed policies
- See `systemd-pcrlock(8)`

Caveats

- Not an HSM(!)
  - Slow devices
- Limited cryptography
  - RSA2048
  - NIST P-256/384
  - SHA256/SHA384
- Needs user-friendly tooling
- Not supported by openssh

ssh-agent

ssh-agent

- Hold private keys for ssh
- Communicated over a UNIX socket
- Caches passwords
- Can offload key operations(!)

ssh-agent

```
$ eval $(ssh-agent)

$ ssh-add .ssh/id_ed25519
Identity added: .ssh/id_ed25519 (localhost)

$ ssh-add -l
256 SHA256:zCgUHuvA2vSr06RulqTNSk7z2eCAMXqf6LuzYihrB+k localhost (ED25519)
```

ssh-agent

```
$ eval $(ssh-agent)

$ ssh-add .ssh/id_ed25519
Identity added: .ssh/id_ed25519 (localhost)

$ ssh-add -l
256 SHA256:zCgUHuvA2vSr06RulqTNSk7z2eCAMXqf6LuzYihrB+k localhost (ED25519)
```

ssh-agent

```
$ eval $(ssh-agent)

$ ssh-add .ssh/id_ed25519
Identity added: .ssh/id_ed25519 (localhost)

$ ssh-add -l
256 SHA256:zCgUHuvA2vSr06RulqTNSk7z2eCAMXqf6LuzYihrB+k localhost (ED25519)
```

ssh-agent

```
$ eval $(ssh-agent)

$ ssh-add .ssh/id_ed25519
Identity added: .ssh/id_ed25519 (localhost)

$ ssh-add -l
256 SHA256:zCgUHuvA2vSr06RulqTNSk7z2eCAMXqf6LuzYihrB+k localhost (ED25519)
```

ssh-agent

```
$ cat ~/.ssh/config
Host localhost
    IdentityFile ~/.ssh/id_ed25519.pub

$ ssh -i ~/.ssh/id_ed25519.pub root@localhost
```

ssh-agent

```
$ cat ~/.ssh/config
Host localhost
    IdentityFile ~/.ssh/id_ed25519.pub

$ ssh -i ~/.ssh/id_ed25519.pub root@localhost
```

ssh-agent

```
1 $ cat ~/.ssh/config
2 Host localhost
3     IdentityFile ~/.ssh/id_ed25519.pub
4
5 $ ssh -i ~/.ssh/id_ed25519.pub root@localhost
```

ssh-agent

```
$ cat ~/.ssh/config
Host localhost
    IdentityFile ~/.ssh/id_ed25519.pub

$ ssh -i ~/.ssh/id_ed25519.pub root@localhost
```

ssh-agent

`golang.org/x/crypto/ssh/agent`

ssh-tpm-agent

https://github.com/foxboron/ssh-tpm-agent

ssh-tpm-agent

- ssh-agent supporting TPM keys
- Support key creation
  - RSA 2048
  - NIST P256/P358
- openssh key import
- `github.com/google/go-tpm`

Start an agent

```
$ export SSH_TPM_AUTH_SOCK="/run/user/1000/ssh-tpm-agent.sock"
$ ssh-tpm-agent &
```

Key creation

```
$ ssh-tpm-keygen
Generating a sealed public/private ecdsa key pair.
Enter file in which to save the key (/home/fox/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/fox/.ssh/id_ecdsa.tpm
Your public key has been saved in /home/fox/.ssh/id_ecdsa.pub
The key fingerprint is:
SHA256:NCMJJ2La+q5tGcngQUQvEOJP3gPH8bMP98wJOEMV564
The key's randomart image is the color of television, tuned to a dead channel.
```

Add key to agent

```
$ ssh-tpm-add /home/fox/.ssh/id_ecdsa.tpm
Identity added: id_ecdsa.tpm

$ export SSH_AUTH_SOCK="/run/user/1000/ssh-tpm-agent.sock"
$ ssh-add -l
256 SHA256:bHnFOGJ/vJetVxa1ncwBu6yoX6Kpj/WgmGu/cP8ZCH0  (ECDSA)
```

Add key to agent

```
$ ssh-tpm-add /home/fox/.ssh/id_ecdsa.tpm
Identity added: id_ecdsa.tpm

$ export SSH_AUTH_SOCK="/run/user/1000/ssh-tpm-agent.sock"
$ ssh-add -l
256 SHA256:bHnFOGJ/vJetVxa1ncwBu6yoX6Kpj/WgmGu/cP8ZCH0  (ECDSA)
```

Add key to agent

```
$ ssh-tpm-add /home/fox/.ssh/id_ecdsa.tpm
Identity added: id_ecdsa.tpm

$ export SSH_AUTH_SOCK="/run/user/1000/ssh-tpm-agent.sock"
$ ssh-add -l
256 SHA256:bHnFOGJ/vJetVxa1ncwBu6yoX6Kpj/WgmGu/cP8ZCH0  (ECDSA)
```

Add key to agent

```
$ ssh-tpm-add /home/fox/.ssh/id_ecdsa.tpm
Identity added: id_ecdsa.tpm

$ export SSH_AUTH_SOCK="/run/user/1000/ssh-tpm-agent.sock"
$ ssh-add -l
256 SHA256:bHnFOGJ/vJetVxa1ncwBu6yoX6Kpj/WgmGu/cP8ZCH0  (ECDSA)
```

Add key to agent

```
$ ssh-tpm-add /home/fox/.ssh/id_ecdsa.tpm
Identity added: id_ecdsa.tpm

$ export SSH_AUTH_SOCK="/run/user/1000/ssh-tpm-agent.sock"
$ ssh-add -l
256 SHA256:bHnFOGJ/vJetVxa1ncwBu6yoX6Kpj/WgmGu/cP8ZCH0  (ECDSA)
```

Key import

```
$ ssh-keygen -t ecdsa -f id_ecdsa
[...]

$ ssh-tpm-keygen --import id_ecdsa
Sealing an existing public/private ecdsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_ecdsa.tpm
The key fingerprint is:
SHA256:bDn2EpX6XRX5ADXQSuTq+uUyia/eV3Z6MW+UtxjnXvU
The key's randomart image is the color of television, tuned to a dead channel.
```

Key import

```
$ ssh-keygen -t ecdsa -f id_ecdsa
[...]

$ ssh-tpm-keygen --import id_ecdsa
Sealing an existing public/private ecdsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_ecdsa.tpm
The key fingerprint is:
SHA256:bDn2EpX6XRX5ADXQSuTq+uUyia/eV3Z6MW+UtxjnXvU
The key's randomart image is the color of television, tuned to a dead channel.
```

Key import

```
$ ssh-keygen -t ecdsa -f id_ecdsa
[...]

$ ssh-tpm-keygen --import id_ecdsa
Sealing an existing public/private ecdsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_ecdsa.tpm
The key fingerprint is:
SHA256:bDn2EpX6XRX5ADXQSuTq+uUyia/eV3Z6MW+UtxjnXvU
The key's randomart image is the color of television, tuned to a dead channel.
```

Key import

```
$ ssh-keygen -t ecdsa -f id_ecdsa
[...]

$ ssh-tpm-keygen --import id_ecdsa
Sealing an existing public/private ecdsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_ecdsa.tpm
The key fingerprint is:
SHA256:bDn2EpX6XRX5ADXQSuTq+uUyia/eV3Z6MW+UtxjnXvU
The key's randomart image is the color of television, tuned to a dead channel.
```

Demo!

Hierarchy Keys

```
$ export SSH_AUTH_SOCK="/run/user/1000/ssh-tpm-agent.sock"
$ ssh-tpm-agent --hierarchy owner --no-load &
$ ssh-add -l
2048 SHA256:yt7A20tcRnzgaD2ATgAXSNWy9sP6wznysp3SkoK3Gj8 Owner hierarchy key (RSA)
256 SHA256:PmEsMeh/DwFP04iUaWLNeX4maMR6r1vfqw1BbbdFjIg Owner hierarchy key (ECDSA
```

Hierarchy Keys

```
$ export SSH_AUTH_SOCK="/run/user/1000/ssh-tpm-agent.sock"
$ ssh-tpm-agent --hierarchy owner --no-load &
$ ssh-add -l
2048 SHA256:yt7A20tcRnzgaD2ATgAXSNWy9sP6wznysp3SkoK3Gj8 Owner hierarchy key (RSA)
256 SHA256:PmEsMeh/DwFP04iUaWLNeX4maMR6r1vfqw1BbbdFjIg Owner hierarchy key (ECDSA
```

Hierarchy Keys

```
$ export SSH_AUTH_SOCK="/run/user/1000/ssh-tpm-agent.sock"
$ ssh-tpm-agent --hierarchy owner --no-load &
$ ssh-add -l
2048 SHA256:yt7A20tcRnzgaD2ATgAXSNWy9sP6wznysp3SkoK3Gj8 Owner hierarchy key (RSA)
256 SHA256:PmEsMeh/DwFP04iUaWLNeX4maMR6r1vfqw1BbbdFjIg Owner hierarchy key (ECDSA
```

Hierarchy Keys

```
$ export SSH_AUTH_SOCK="/run/user/1000/ssh-tpm-agent.sock"
$ ssh-tpm-agent --hierarchy owner --no-load &
$ ssh-add -l
2048 SHA256:yt7A20tcRnzgaD2ATgAXSNWy9sP6wznysp3SkoK3Gj8 Owner hierarchy key (RSA)
256 SHA256:PmEsMeh/DwFP04iUaWLNeX4maMR6r1vfqw1BbbdFjIg Owner hierarchy key (ECDSA
```

Remote key wrapping

On the client

```
$ tpm2_createprimary -C o -G ecc -g sha256 -c prim.ctx \
  -a 'restricted|decrypt|fixedtpm|fixedparent|sensitivedataorigin|userwithauth|nc
  -f pem -o srk.pem
```

On the provisoning remote

```
1 $ ssh-keygen -t ecdsa -b 256 -N "" -f ./ecdsa.key
2 $ ssh-tpm-keygen --wrap-with srk.pub --wrap ecdsa.key -f wrapped_id_ecdsa
```

On the provisoning remote

```
1 $ ssh-keygen -t ecdsa -b 256 -N "" -f ./ecdsa.key
2 $ ssh-tpm-keygen --wrap-with srk.pub --wrap ecdsa.key -f wrapped_id_ecdsa
```

On the provisoning remote

```
1 $ ssh-keygen -t ecdsa -b 256 -N "" -f ./ecdsa.key
2 $ ssh-tpm-keygen --wrap-with srk.pub --wrap ecdsa.key -f wrapped_id_ecdsa
```

On the provisoning remote

```
1 $ ssh-keygen -t ecdsa -b 256 -N "" -f ./ecdsa.key
2 $ ssh-tpm-keygen --wrap-with srk.pub --wrap ecdsa.key -f wrapped_id_ecdsa
```

On the client

```
1 $ ssh-tpm-keygen --import ./wrapped_id_ecdsa.tpm -f id_ecdsa.tpm
2 $ ssh-tpm-add id_ecdsa.tpm
```

On the client

```
1 $ ssh-tpm-keygen --import ./wrapped_id_ecdsa.tpm -f id_ecdsa.tpm
2 $ ssh-tpm-add id_ecdsa.tpm
```

On the client

```
1 $ ssh-tpm-keygen --import ./wrapped_id_ecdsa.tpm -f id_ecdsa.tpm
2 $ ssh-tpm-add id_ecdsa.tpm
```

On the client

```
1 $ ssh-tpm-keygen --import ./wrapped_id_ecdsa.tpm -f id_ecdsa.tpm
2 $ ssh-tpm-add id_ecdsa.tpm
```

Hostkeys

```
$ sudo ssh-tpm-keygen -A
2023/09/03 17:03:08 INFO Generating new ECDSA host key
2023/09/03 17:03:08 INFO Wrote /etc/ssh/ssh_tpm_host_ecdsa_key.tpm
2023/09/03 17:03:08 INFO Generating new RSA host key
2023/09/03 17:03:15 INFO Wrote /etc/ssh/ssh_tpm_host_rsa_key.tpm

$ sudo ssh-tpm-hostkeys --install-system-units
Installed /usr/lib/systemd/system/ssh-tpm-agent.service
Installed /usr/lib/systemd/system/ssh-tpm-agent.socket
Installed /usr/lib/systemd/system/ssh-tpm-genkeys.service
Enable with: systemctl enable --now ssh-tpm-agent.socket

$ sudo ssh-tpm-hostkeys --install-sshd-config
Installed /etc/ssh/sshd_config.d/10-ssh-tpm-agent.conf
Restart sshd: systemd restart sshd
```

Hostkeys

```
$ sudo ssh-tpm-keygen -A
2023/09/03 17:03:08 INFO Generating new ECDSA host key
2023/09/03 17:03:08 INFO Wrote /etc/ssh/ssh_tpm_host_ecdsa_key.tpm
2023/09/03 17:03:08 INFO Generating new RSA host key
2023/09/03 17:03:15 INFO Wrote /etc/ssh/ssh_tpm_host_rsa_key.tpm

$ sudo ssh-tpm-hostkeys --install-system-units
Installed /usr/lib/systemd/system/ssh-tpm-agent.service
Installed /usr/lib/systemd/system/ssh-tpm-agent.socket
Installed /usr/lib/systemd/system/ssh-tpm-genkeys.service
Enable with: systemctl enable --now ssh-tpm-agent.socket

$ sudo ssh-tpm-hostkeys --install-sshd-config
Installed /etc/ssh/sshd_config.d/10-ssh-tpm-agent.conf
Restart sshd: systemd restart sshd
```

Hostkeys

```
2023/09/03 17:03:08 INFO Generating new ECDSA host key
2023/09/03 17:03:08 INFO Wrote /etc/ssh/ssh_tpm_host_ecdsa_key.tpm
2023/09/03 17:03:08 INFO Generating new RSA host key
2023/09/03 17:03:15 INFO Wrote /etc/ssh/ssh_tpm_host_rsa_key.tpm

$ sudo ssh-tpm-hostkeys --install-system-units
Installed /usr/lib/systemd/system/ssh-tpm-agent.service
Installed /usr/lib/systemd/system/ssh-tpm-agent.socket
Installed /usr/lib/systemd/system/ssh-tpm-genkeys.service
Enable with: systemctl enable --now ssh-tpm-agent.socket

$ sudo ssh-tpm-hostkeys --install-sshd-config
Installed /etc/ssh/sshd_config.d/10-ssh-tpm-agent.conf
Restart sshd: systemd restart sshd

```

Hostkeys

```
$ sudo ssh-tpm-hostkeys --install-system-units
Installed /usr/lib/systemd/system/ssh-tpm-agent.service
Installed /usr/lib/systemd/system/ssh-tpm-agent.socket
Installed /usr/lib/systemd/system/ssh-tpm-genkeys.service
Enable with: systemctl enable --now ssh-tpm-agent.socket

$ sudo ssh-tpm-hostkeys --install-sshd-config
Installed /etc/ssh/sshd_config.d/10-ssh-tpm-agent.conf
Restart sshd: systemd restart sshd

$ systemctl enable --now ssh-tpm-agent.socket
$ systemd restart sshd

$ sudo ssh-tpm-hostkeys
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCLDH2xM
```

Hostkeys

```
Installed /usr/lib/systemd/system/ssh-tpm-genkeys.service
Enable with: systemctl enable --now ssh-tpm-agent.socket

$ sudo ssh-tpm-hostkeys --install-sshd-config
Installed /etc/ssh/sshd_config.d/10-ssh-tpm-agent.conf
Restart sshd: systemd restart sshd

$ systemctl enable --now ssh-tpm-agent.socket
$ systemd restart sshd

$ sudo ssh-tpm-hostkeys
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCLDH2xM
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDAoMPsv5tEpTDFw34ltkF45dTHAPl4aLu6HigBkNnI

$ ssh-keyscan -t ecdsa localhost
# localhost:22 SSH-2.0-OpenSSH_9.4
```

Hostkeys

```
$ sudo ssh-tpm-hostkeys --install-sshd-config
Installed /etc/ssh/sshd_config.d/10-ssh-tpm-agent.conf
Restart sshd: systemd restart sshd

$ systemctl enable --now ssh-tpm-agent.socket
$ systemd restart sshd

$ sudo ssh-tpm-hostkeys
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCLDH2xM
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDAoMPsv5tEpTDFw34ltkF45dTHAPl4aLu6HigBkNnI

$ ssh-keyscan -t ecdsa localhost
# localhost:22 SSH-2.0-OpenSSH_9.4
localhost ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAA
```

Hostkeys

```
12
13 $ sudo ssh-tpm-hostkeys --install-sshd-config
14 Installed /etc/ssh/sshd_config.d/10-ssh-tpm-agent.conf
15 Restart sshd: systemd restart sshd
16
17 $ systemctl enable --now ssh-tpm-agent.socket
18 $ systemd restart sshd
19
20 $ sudo ssh-tpm-hostkeys
21 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCLDH2xM
22 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDAoMPsv5tEpTDFw34ltkF45dTHAPl4aLu6HigBkNnI
23
24 $ ssh-keyscan -t ecdsa localhost
25 # localhost:22 SSH-2.0-OpenSSH_9.4
26 localhost ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAA
```

Hostkeys

```
$ sudo ssh-tpm-keygen -A
2023/09/03 17:03:08 INFO Generating new ECDSA host key
2023/09/03 17:03:08 INFO Wrote /etc/ssh/ssh_tpm_host_ecdsa_key.tpm
2023/09/03 17:03:08 INFO Generating new RSA host key
2023/09/03 17:03:15 INFO Wrote /etc/ssh/ssh_tpm_host_rsa_key.tpm

$ sudo ssh-tpm-hostkeys --install-system-units
Installed /usr/lib/systemd/system/ssh-tpm-agent.service
Installed /usr/lib/systemd/system/ssh-tpm-agent.socket
Installed /usr/lib/systemd/system/ssh-tpm-genkeys.service
Enable with: systemctl enable --now ssh-tpm-agent.socket

$ sudo ssh-tpm-hostkeys --install-sshd-config
Installed /etc/ssh/sshd_config.d/10-ssh-tpm-agent.conf
Restart sshd: systemd restart sshd
```

TPM 2.0 Key Files

Workgroup:	Network Working Group
Internet-Draft:	draft-bottomley-tpm-keys-00
Published:	23 July 2024
Intended Status:	Informational
Expires:	24 January 2025
Author:	J. Bottomley
	Linux Kernel

ASN.1 Specification for TPM 2.0 Key Files

Abstract

This specification is designed to be an extension to the ASN.1 (defined in [X.680]) specification of PKCS #1 [RFC8017] to define the file format of private keys that need to be loaded into a TPM 2 device to operate.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 24 January 2025.

Copyright Notice

https://www.hansenpartnership.com/draft-bottomley-tpm2-keys.html

TPM 2.0 Key Files

- ASN.1 format for TPM keys
- openssl `tpm2` provider
- Linux keyring support

TPM 2.0 Key Files

- Support different key types
  - Loadable Keys
  - Importable Keys
  - Sealed Keys

TPM 2.0 Key Files

https://github.com/Foxboron/go-tpm-keyfiles

openssl key creation

```
$ openssl genpkey -provider tpm2 \
    -algorithm EC -pkeyopt group:P-256 \
    -pkeyopt user-auth:1234 \
    -out id_ecdsa.tpm

$ ssh-tpm-keygen --print-pubkey ./id_ecdsa.tpm
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHA[...]Tkw=
```

openssl key creation

```
$ openssl genpkey -provider tpm2 \
    -algorithm EC -pkeyopt group:P-256 \
    -pkeyopt user-auth:1234 \
    -out id_ecdsa.tpm

$ ssh-tpm-keygen --print-pubkey ./id_ecdsa.tpm
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHA[...]Tkw=
```

openssl key creation

```
$ openssl genpkey -provider tpm2 \
    -algorithm EC -pkeyopt group:P-256 \
    -pkeyopt user-auth:1234 \
    -out id_ecdsa.tpm

$ ssh-tpm-keygen --print-pubkey ./id_ecdsa.tpm
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHA[...]Tkw=
```

openssl key creation

```
$ openssl genpkey -provider tpm2 \
    -algorithm EC -pkeyopt group:P-256 \
    -pkeyopt user-auth:1234 \
    -out id_ecdsa.tpm

$ ssh-tpm-keygen --print-pubkey ./id_ecdsa.tpm
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHA[...]Tkw=
```

But is it secure?

SSH

TPM Key

SHA256:GfOTIDD7wpDrDPRR4rPdP1dhjzqssYAAKPtYW9drNI4

Added on Aug 31, 2024

Never used — Read/write

Delete

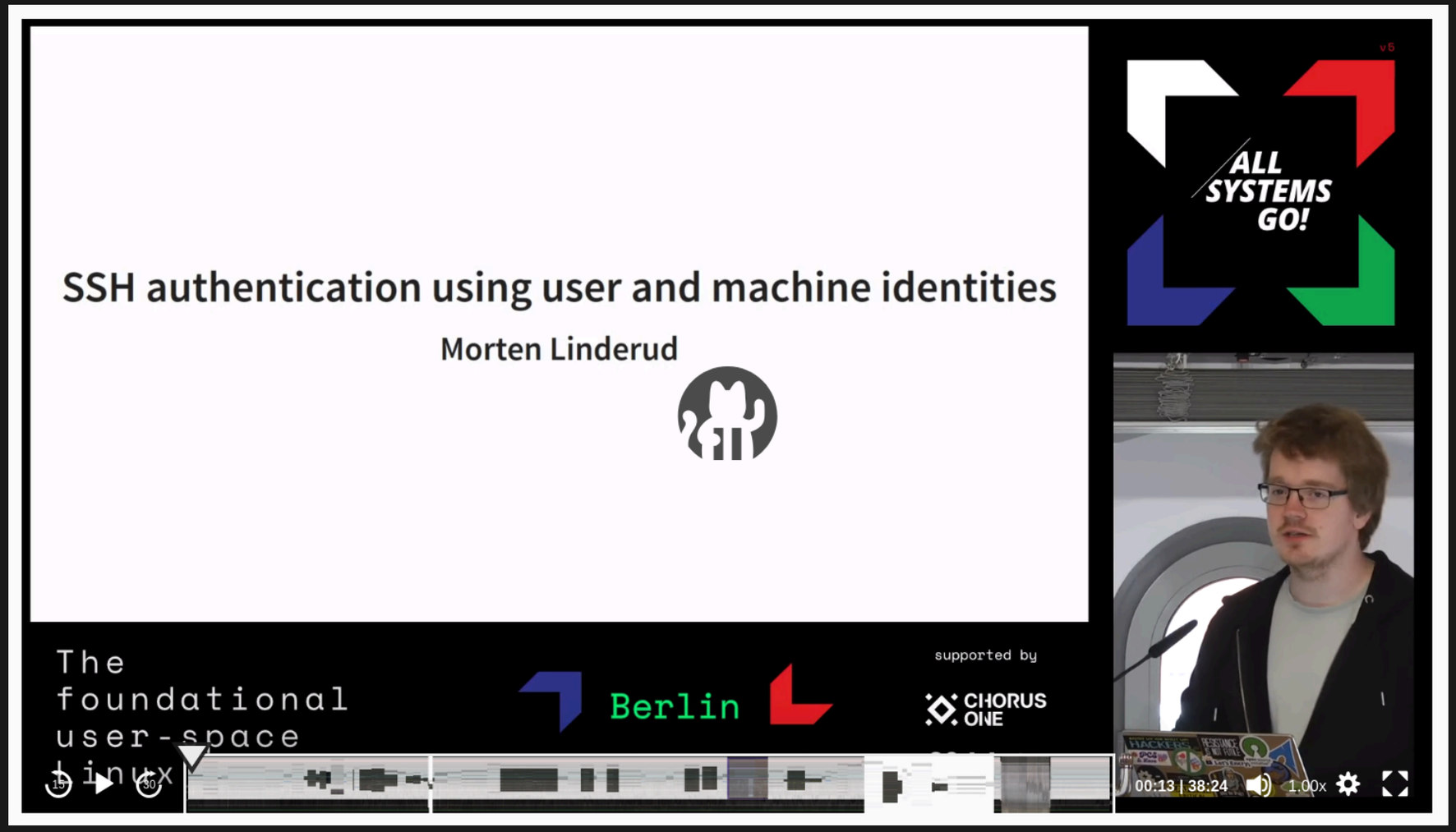
Github ssh key

```
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLXM/
KDMRNt84G78CE0I0TBws2gfF65fA94YBmB57kYs0ZxiHQxykSoxEE6zaPyfgw5IegpkqPz9jOdEH
qqt/bg= fox@framework
```

PASSWORD: 1234

Improvements

- TPM Policy support
  - `systemd-pcrlock`
- Landlock support
- An `1.0.0` release!
- Feature requests welcome!



https://media.ccc.de/v/all-systems-go-2024-320-ssh-authentication-using-user-and-machine-identities

Thanks!

- Github: Foxboron
- https://linderud.dev
- Email: morten@linderud.pw
- Mastodon: https://chaos.social/@Foxboron