SSH - targeted MitM attacks

One **RFC** to rule them all,
one **RFC** to find them,
one **RFC** to bring them all,
and in the darkness bind them

Manfred Kaiser

Österreichisches Bundesheer
Direktion IKT und Cyber

! IMPORTANT !

The described attack assumes that the SSH keys are protected by Fido2 tokens or SSH-Askpass.

If this is not the case, a MitM attack is much easier!

SSH 2 Protokoll

SSH Host Key

🔑 RFC-4251 - SSH Host Keys

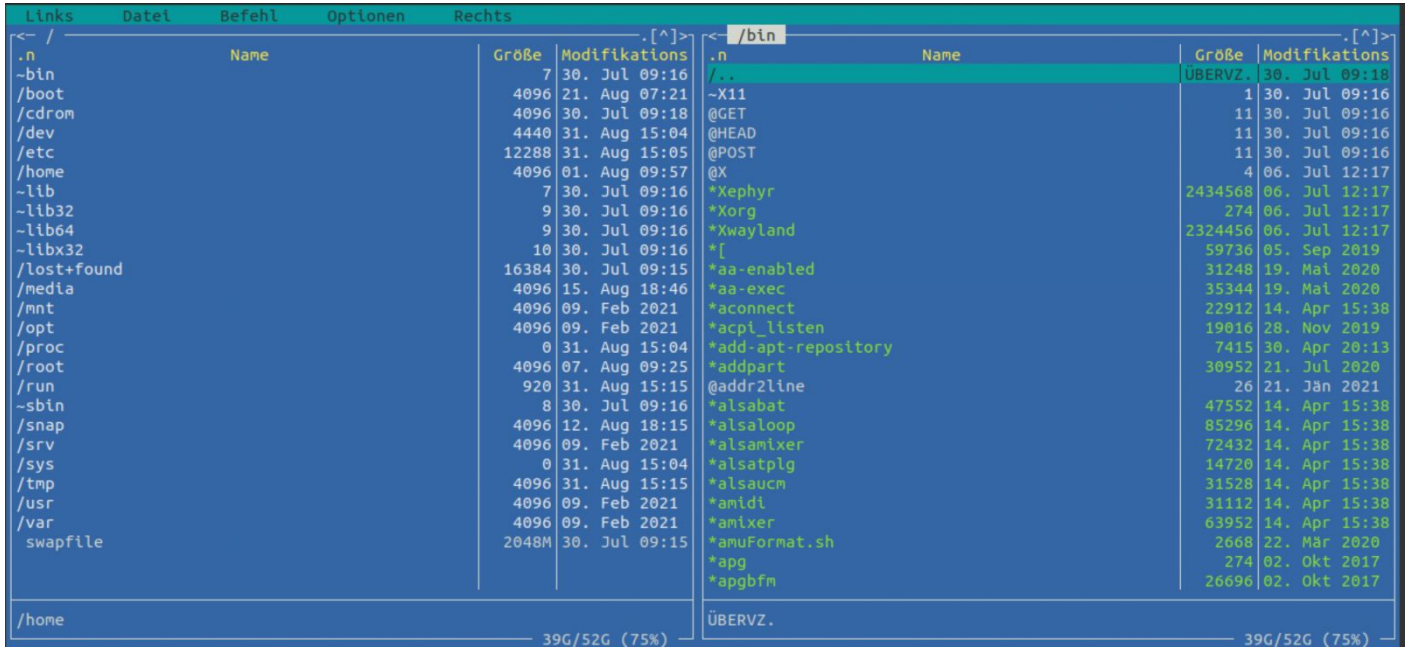
```
The authenticity of host 'github.com (140.82.121.3)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

➔ **Trust on first use**

➔ Primary security feature of SSH

Midnight Commander

CVE-2021-36370: server fingerprint isn't verified *(discovered by AUT-milCERT during an audit of open source software)*



Midnight Commander

```
    /* At this point we havn't yet authenticated.  The first thing to do
     * is check the hostkey's fingerprint against our known hosts Your app
     * may have it hard coded, may go to a file, may present it to the
     * user, that's your call
     */
    sftpfs_super->fingerprint =
        libssh2_hostkey_hash (sftpfs_super->session, LIBSSH2_HOSTKEY_HASH_SHA1);

    if (!sftpfs_recognize_auth_types (super))
```

SSH 2 Protokoll

Authentication

🔒 RFC-4252 - SSH Authentication Protocol

```
The server drives the authentication by telling the client which
authentication methods can be used...
```

```
The only REQUIRED authentication 'method name' is "publickey"
authentication.  All implementations MUST support this method; ...
```

Offered methods:

- **publickey**,password

🔒 RFC-4252 - SSH Authentication Protocol

```
The client has the freedom to try the methods listed by
the server in any order.
```

Most common order for SSH clients:

1. none
2. publickey
3. keyboad-interactive
4. password

🔒 RFC-4252 - "none" authentication

```
A client may request a list of authentication 'method name' values
that may continue by using the "none" authentication 'method name'.

If no authentication is needed for the user, the server MUST return
SSH_MSG_USERAUTH_SUCCESS.  Otherwise, the server MUST return
SSH_MSG_USERAUTH_FAILURE and MAY return with it a list of methods
that may continue in its 'authentications that can continue' value.
```

🔒 RFC-4252 - publickey authentication

```
The only REQUIRED authentication 'method name' is "publickey"
authentication.  All implementations MUST support this
method;
```

It is possible not to offer "publickey" authentication, but not RFC compliant!

🔒 RFC-4256 - keyboard-interactive

```
The server may send as many requests as are necessary to authenticate
the client; the client MUST be prepared to handle multiple exchanges.
```

- Similar to password authentication
- Supports multiple inputs
- Mostly used for 2 factor authentication

🔒 RFC-4256 - keyboard-interactive

```
int         num-prompts
string      prompt[1] (ISO-10646 UTF-8)
boolean     echo[1]
```

```
The num-prompts field may be `0', in which case there will be no
prompt/echo fields in the message
```

If no prompts are sent, no input is required on the client side.
input has to be made on the client side.

🔒 RFC-4252 - password authentication

- Security depends on password
- Password is transmitted in clear text
- Can be reused in case of MitM attacks

Password authentication should not be used!

draft-ietf-secsh-agent-02

SSH Agent Forwarding

```
Anyone with access to the authentication agent
can perform private key operations with the agent.
...
The authentication agent should not be run or
forwarded to machine whose integrity is not trusted...
```

- **OpenSSH 8.4 (2020-09-27)**
  - scp(1), sftp(1): allow the -A flag to explicitly enable agent forwarding in scp and sftp.

Fido2 Token

? 2 factor authentication for SSH ?

👇 Fido2 Token

- Hardware token for secure login
- Supposed to provide protection against phishing
- Support since OpenSSH 8.2
- Provides protection against misuse

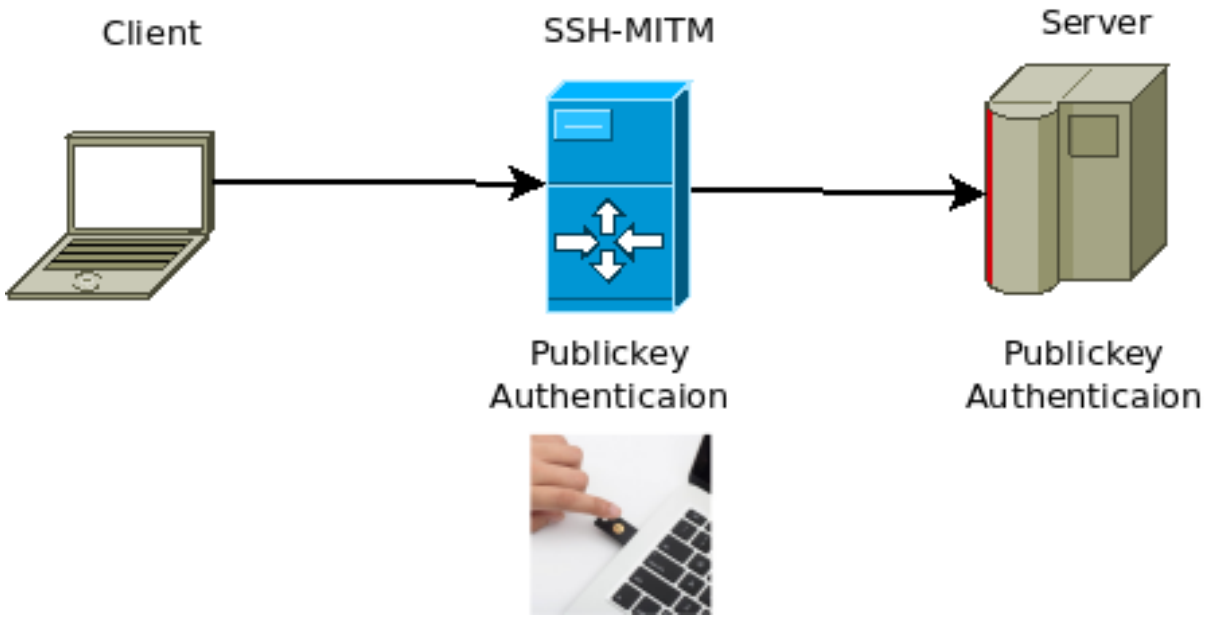
👇 Fido2 Token - "*Presence*" Erkennung

```
$ ssh USER@34.212.121.12 -p 22
Confirm user presence for key ECDSA-SK SHA256:esvq6KPZ5FGtt...
[Tab your YubiKey U2F Security Key now]
```

Man in the Middle

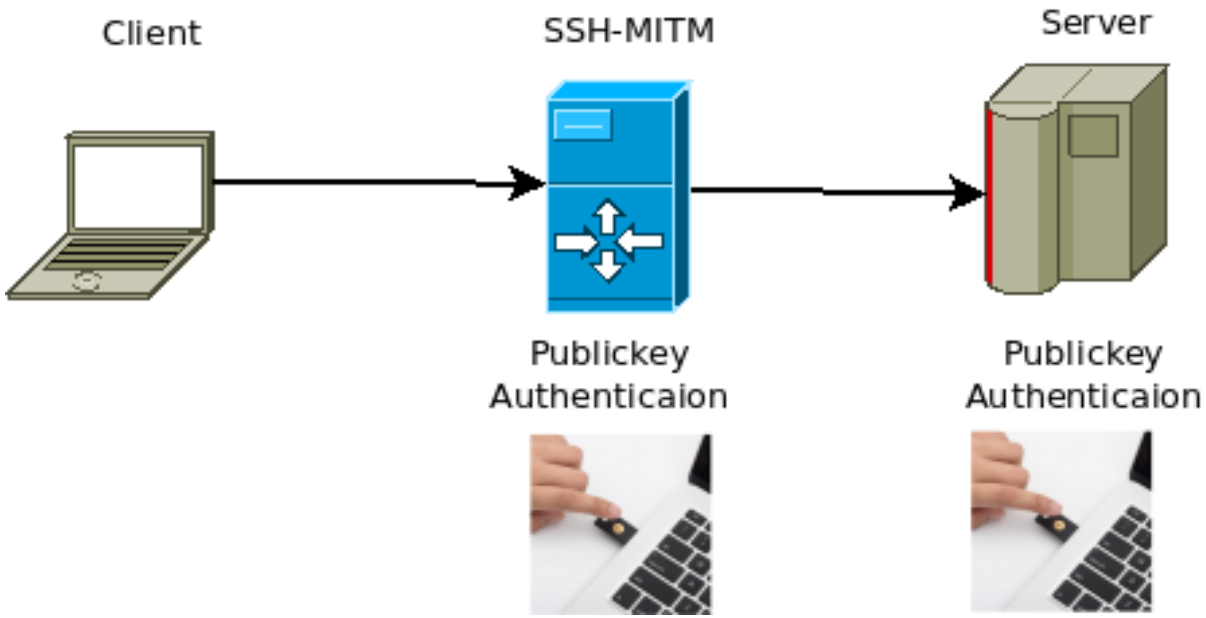
Fido2 Token & Agent Forwarding

🔨 Classic MitM Attack



When logging in to the MitM server, the token must be must be pressed

🔨 Classic MitM Attack



A login on the target server requires a 2.
Confirmation of the token -> MitM attack detected!

Trivial Success Authentication

Spoofing FIDO2 Tokens and SSH-Askpass

🗡️ Trivial authentication

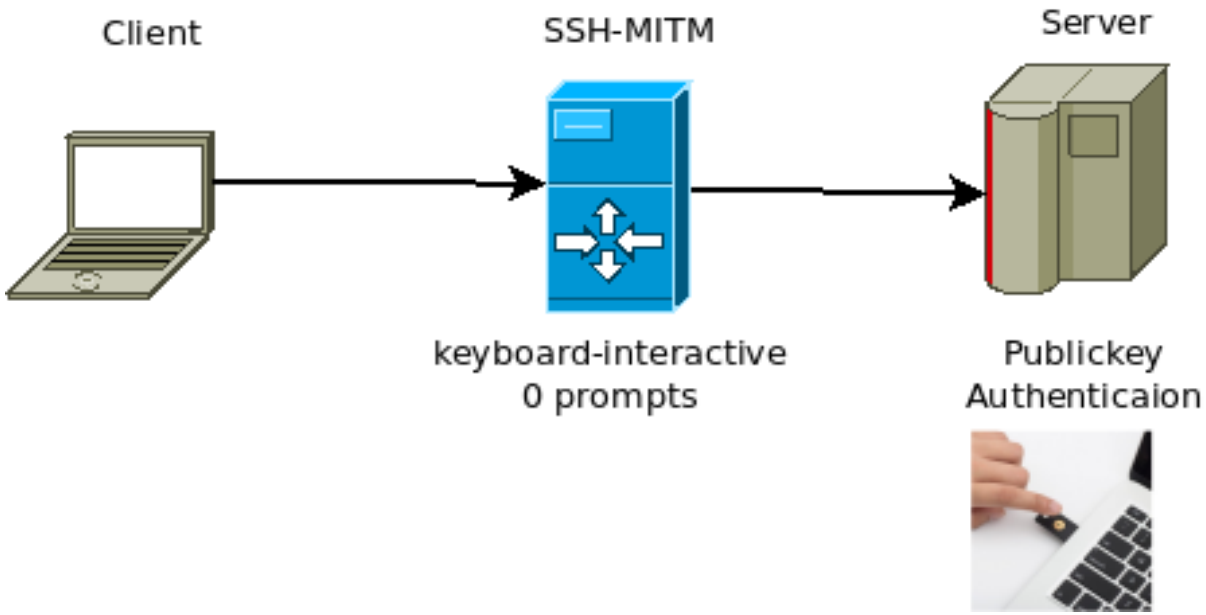
- Client accepts the login
- Client is not prompted to perform authentication
- No user interaction necessary
- Agent forwarding recommended

🗡️ Trivial authentication - Methods

- none -> no login necessary
- keyboard-intreactive -> **only with 0 prompts**

Client is forced to login

🗡 Trivial Success Authentication - Attack



The token must be confirmed only 1 time!

🗡️ Trivial Success Authentication - Exploit

```
# Install SSH-MITM
pip install ssh-mitm

# start server
ssh-mitm --remote-host TARGET \
  --disallow-publickey-auth \
  --enable-keyboard-interactive-auth \
  --disable-keyboard-interactive-prompts
```

🗡️ Trivial Success Authentication - CVE

- **PuTTY:** CVE-2021-36367
- **OpenSSH:** CVE-2021-36368
- **Dropbear:** CVE-2021-36369

🗡️ Trivial Success Authentication - Patches

Patches were developed by **AUT-milCERT** in collaboration with **Simon Tatham (PuTTY)** and **Matt Johnston (Dropbear)**.

- PuTTY: >= 0.76
- Dropbear has already merged the patch into the master.

Note PuTTY:
Trust Sigils were introduced in 0.71 to allow the user to detect spoofing attacks, but they are not automatically prevented.

🗡️ Trivial Success Authentication - OpenSSH

manfred-kaiser commented on 19 Aug — Author

AUT-milCERT want's to release information about "trivial success authentication" in the next weeks.

Do you want to merge our patch or are you planning to implement some other mitigation approaches in the next release?

djmdjm commented on 20 Aug — Contributor

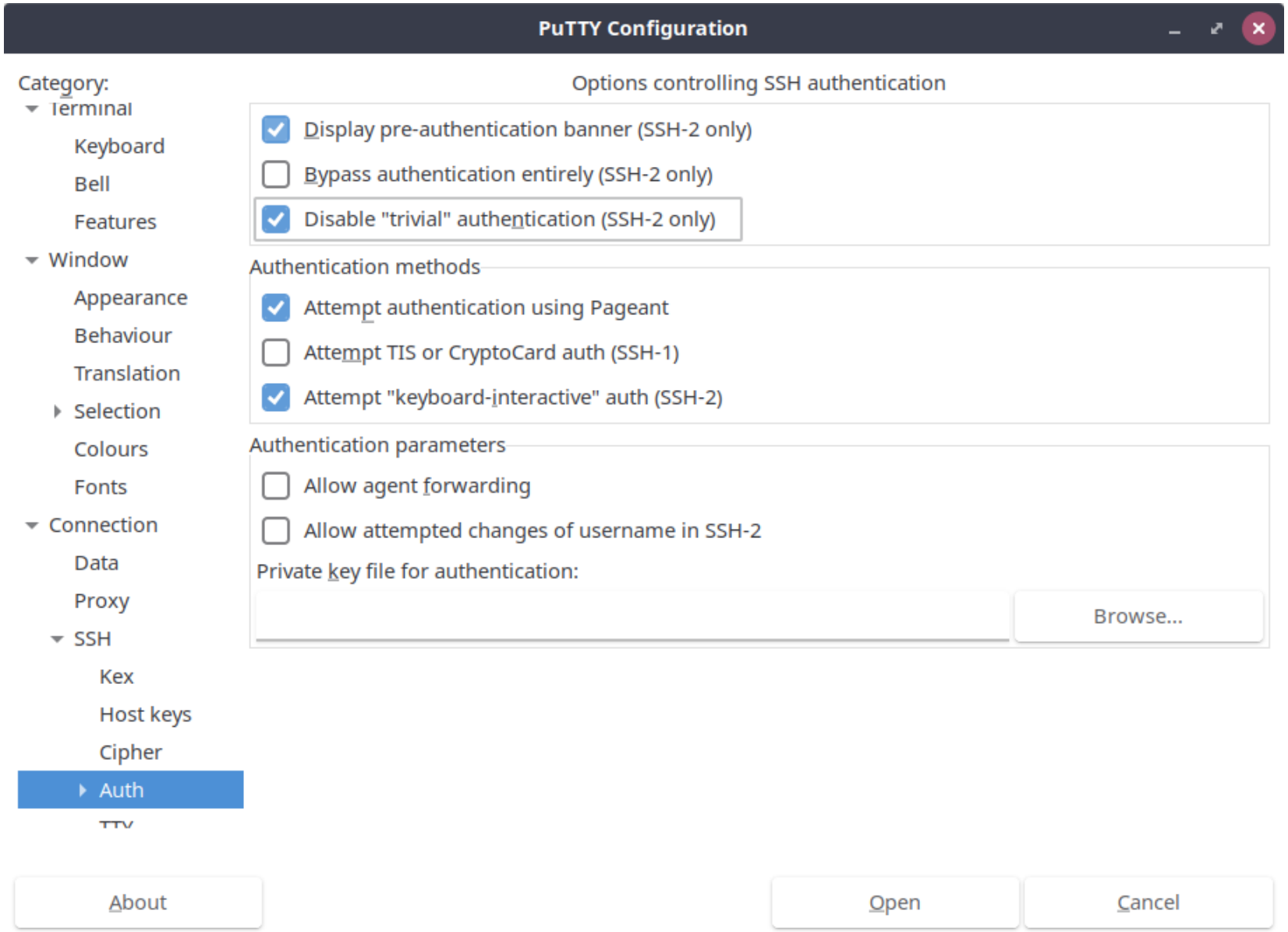
No, we do not plan to merge this patch.

🗡️ Trivial Success Authentication - OpenSSH

Argumentation:

- spoofing attacks are not a security vulnerability
- none Authentication is not a security risk
  - Verbose logging has been adjusted to detect a successful login using "none".

Problem -> nobody reads 100 log lines every time a connection is established.


PuTTY Configuration
Category:
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Fonts
Connection
Data
Proxy
SSH
Kex
Host keys
Cipher
Auth
TTY
Options controlling SSH authentication
Display pre-authentication banner (SSH-2 only)
Bypass authentication entirely (SSH-2 only)
Disable "trivial" authentication (SSH-2 only)
Authentication methods
Attempt authentication using Pageant
Attempt TIS or CryptoCard auth (SSH-1)
Attempt "keyboard-interactive" auth (SSH-2)
Authentication parameters
Allow agent forwarding
Allow attempted changes of username in SSH-2
Private key file for authentication:
Browse...
About
Open
Cancel

SSH-MITM - ssh audits made simple

ssh man-in-the-middle (ssh-mitm) server for security audits supporting **publickkey authentication**, **session hijacking** and **file manipulation**

- **Webseite:** https://www.ssh-mitm.at
- **Github:** https://github.com/ssh-mitm/ssh-mitm
- **Lizenz:** LGPL-3.0-or-later