Trivial Authentication

The “trivial authentication” phishing attack was found by **Manfred Kaiser (AUT-milCERT)** in cooperation with **Simon Tatham (PuTTY)** and **Matt Johnston (Dropbear)** during an security audit.

Trivial authentication is a special form of phishing attack that exploits authentication methods to force a client to log in.

Trivial authentication methods are those that do not require any interaction from the client. “none” is an example of a trivial authentication method because it allows an immediate login to a server and in most cases the client simply accepts it if the login was successful.

SSH keys can be additionally protected with a FIDO2 token as of OpenSSH 8.2, thus protecting against misuse should the private key be compromised.

In this way, it is possible to bypass confirmation of the FIDO2 token for login to the Man in the Middle server. If SSH Agent Forwarding is enabled, an attacker can use the FIDO2 protected key to login to another server. The phishing attack is not noticeable in this case because the user only has to confirm the key once. For which server the confirmation is done is not shown, which is why the user has no way to check anything.

From the user’s point of view, the user expects that exactly one confirmation of the key must take place. However, the login on the Man in the Middle server was done with an authentication method that did not require access to the private key. Only the abusive login of the attacker requires a confirmation. Thus, the user’s expectation is met and in most cases access to the private key is allowed.

If the client has phishing detection, as it has since PuTTY 0.71, a user can detect that the confirmation is for another server, provided that the trust sigils are appropriately controlled.

Assigned CVEs

- PuTTY: CVE-2021-36367
- OpenSSH: CVE-2021-36368

- Dropbear: CVE-2021-36369

Authentication methods

none authentication

The easiest way for a phishing attack is to use “none” authentication. This is used to grant a client access to a server without a login. For a Man in the Middle attacker, this is interesting in that no other authentication methods are executed.

However, using “none” authentication has several disadvantages. The client is forced to log in immediately. As a result, an attacker loses the opportunity to test other authentication methods.

It could be that a user wants to connect to a server that requires “publickey” authentication only for selected users and only a password is sufficient for all others. In such a case, the attacker would no longer have the possibility to ask for the password and a login would no longer be possible for the attacker.

Publickey authentication

Publickey authentication is a non-trivial authentication method. The reason is that the client creates a signature with the private key, which should be validated by the server.

However, an attacker cannot make a client bypass the signature process. The attacker only has the option to fully perform or reject an authentication using Publickey.

Nevertheless, publickey authentication is an essential part of a phishing attack on FIDO2/SSH-Askpass protected keys.

As described in the Man in the Middle attack on public key authentication chapter, it is necessary to check all public keys against the actual target server. This is necessary to find out which key would have been used for the login.

Once the key is known, potentially better targets can be searched for.

After the key is known, the publickey authentication of the client can be terminated. The actual phishing attack then takes place using a different authentication method.

It is essential that an authentication using publickey authentication against the Man in the Middle server must never be successful. Otherwise, there is a risk that the phishing attack will be noticed.

“keyboard-interactive” authentication

The actual phishing attack takes place in the “keyboard-interactive” authentication method. With “keyboard-interactive” any number of prompts can be sent to the client. The number of prompts must therefore be >=0.

If 0 (zero) prompts are sent to the client, no client interaction is necessary. However, if at least 1 prompt is sent to the client, the user must make an input.

If 0 prompts are sent to the client, “keyboard-interactive” is a trivial authentication. Only if at least 1 prompt is sent, it is no longer considered trivial.

For the phishing attack it is necessary that 0 (zero) prompts are sent to the client to force a login on the server. Once the session is established, the attacker can access the passed agent and connect to its target server.

SSH-MITM - PoC to phish FIDO2 tokens

When using SSH-Askpass and/or a FIDO2 key, it is necessary that the usage of a private key always gets manually approved. If a MitM attack occurs that requires two authentifications (one on the MitM server and one on the remote server), the user has to confirm the usage of his private key twice.

However, the user knows that there should only be one confirmation process, as he is connecting to only one server. Therefore it is very likely, that he accepts the first confirmation and declines the second one. If that happens, the MitM server can‘t authenticate itself to the remote server.

Note

It’s recommended to use Agent forwarding, when trying out the bypass!

If you need to handle clients without a forwarded agent, you can configure a fallback host.

Phishing FIDO2 tokens / SSH-Askpass

Since version 1.0.0 SSH-MITM has full support for phishing FIDO2 tokens.

```
$ ssh-mitm server --remote-host TARGET --enable-trivial-auth
```

Connect the client to SSH-MITM with agent forwarding:

```
$ ssh -A -p 10022 localhost
```

Verification without bypass

To verify the default behavior for a login using publickey authentication on the MitM server and on the remote server, SSH-MITM can be started with following parameters:

```
$ ssh-mitm server --remote-host TARGET
```

Connect the client to SSH-MITM with agent forwarding:

```
$ ssh -A -p 10022 localhost
```

In this case, the client must authenticate to the SSH-MITM server using “publickey”, which requires a confirmation.

After the user has successfully logged in to the MitM server, the agent is requested and logging in to the remote host is started, which requires a 2nd confirmation by ssh-