"SSHARK!"

Centralized SSH key expiration and revocation without server support

Anatole Shaw
ash@greenhost.nl

SSH key-based authentication

- SSH `authorized_keys` files
- They list public keys for authentication
- Lists are normally static & unmanaged
- Keys have no expiration date
- Because they are public keys, people like to have a small number of of them (often 1) which appear in `authorized_keys` files on a large number of servers

What's the problem?

- A compromised SSH private key may provide access to a large number of servers
- Disabling the access rights of the compromised key requires modifying the `authorized_keys` file on every server where it appears
- What is the lifetime of your key? What is your procedure for removing it from use?

OpenSSH Project's solution

- Extend the `authorized_keys` spec
  - Add a `@revoked` "marker" for keys
  - Add a `@cert-authority` "marker" that designates the key as an authority which certifies other keys
- Why this is not yet a solution
  - Not standardized or well documented
  - No centralized revocation, no expiration
  - How many servers run the latest OpenSSH?
  - All of the servers where your key is?

What we realized

- The `command="..."` option in `authorized_keys` is capable of running a key-granular gatekeeper that verifies key validity
- This option is supported for a long time by OpenSSH and DropBear SSH servers

What we realized

- OpenSSL can take in SSH keys...

```
openssl rsa -in id_rsa -text >id_rsa.pem
```

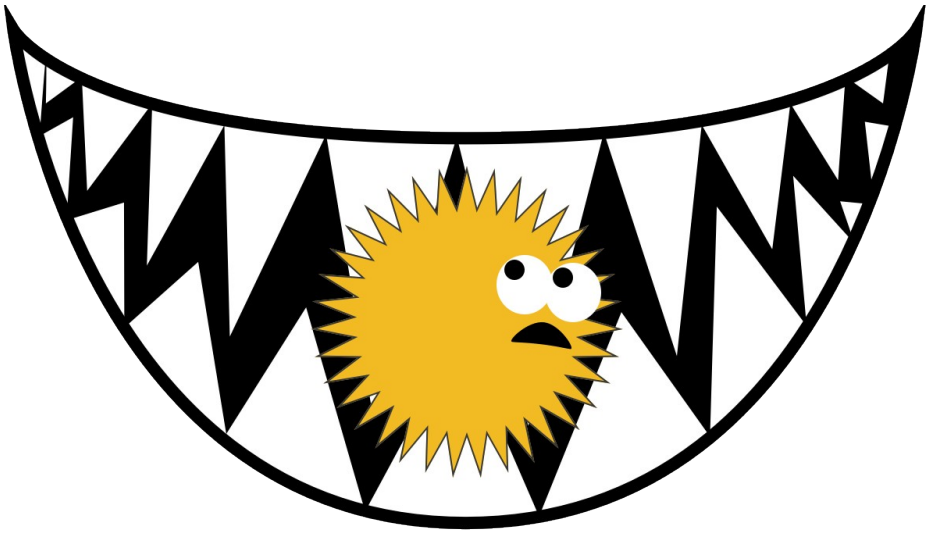
- Then they can be used for signing...

```
openssl rsautl -sign -inkey id_rsa.pem
  -keyform PEM -in message.txt
    >message_signed
```

Introducing SSHARK

"Authorization and Revocation of Keys"

Key expiration and revocation data in DNS, signed by the subject SSH key itself



What is SSHARK?

- The “sshark-gen” program
  - Generates signed messages using your SSH private key
  - Messages indicate validity period, or revocation
  - Output is suitable for inclusion into a DNS zone

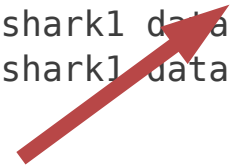
What is SSHARK?

- The “sshark” program
  - Run from your `~/.ssh/authorized_keys` file
  - Uses the key type, fingerprint, and comment field to determine the DNS zone for lookups
  - Looks up TXT records for SSHARK data on key validity
  - If the key is valid, things proceed as normal
  - If the key is revoked or expired, access is denied

How to use SSHARK

- Generate SSHARK TXT records for DNS
- Specify seconds into the future, or “revoke”

```
% sshark-gen ~/.ssh/id_rsa revoke
ssh-rsa-7c34c56a….ash._sshark.greenhost.nl.
  TXT "sshark1 serial 1354024367 expiry 0"
s1354024367.ssh-rsa-7c34c56a….ash._sshark.greenhost.nl. TXT "sshark1 data HdO4lxSG…
s1354024367.ssh-rsa-7c34c56a….ash._sshark.greenhost.nl. TXT "sshark1 data xg4fez2J…
s1354024367.ssh-rsa-7c34c56a….ash._sshark.greenhost.nl. TXT "sshark1 data hL6V6RAj…
s1354024367.ssh-rsa-7c34c56a….ash._sshark.greenhost.nl. TXT "sshark1 data B9JvdnNE…
```



How to use SSHARK

- Upload the *sshark* executable to the server
- Invoke it in your `authorized_keys` file on the key you want to control

```
command="~/bin/sshark -t ssh-rsa
-l 7c:34:c5:6a:90:df:2c:5d:5b:7c:6b:df…"
ssh-rsa AAAAB3NzaC1ycyokQroQi0QFtdbghg…
ash@greenhost.nl
```

SSHARK in action

```
ash@gonzo:~$ ssh fozzy
Enter passphrase for key '/home/ash/.ssh/id_rsa':
[sshark] Your key is valid until Sat Dec  8 03:28:30 2012
Linux fozzy 3.2.0-4-amd64 #1 SMP Debian 3.2.32-1 x86_64
Last login: Tue Nov 27 14:16:21 UTC 2012 from 144.51.73.129
ash@fozzy:~$
```

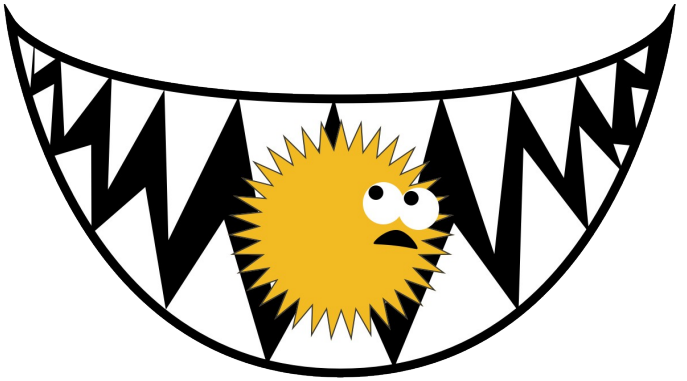
```
ash@gonzo:~$ ssh fozzy
Enter passphrase for key '/home/ash/.ssh/id_rsa':
==========================[sshark]==========================
Sorry, your SSH key expired on Sat Dec  8 03:28:30 2012
Key: ssh-rsa 7c:34:c5:6a:90:df:2c:5d:5b:7c:6b:df:ec:18:48:44
Zone: ash._sshark.greenhost.nl
============================================================
Connection to fozzy closed.
```

The future

- Extend to work with key types other than RSA
- Rewrite in C (currently in Perl) so it can run reliably on the largest variety of systems
- Remove dependency on `ssh-keygen` by importing some code from that program
- Community input and participation!

For more information...

www.sshark.org



www.greenhost.nl

greenhost
duurzame webhosting &
digitale veiligheid

Anatole Shaw
ash@greenhost.nl