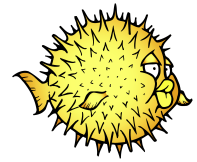
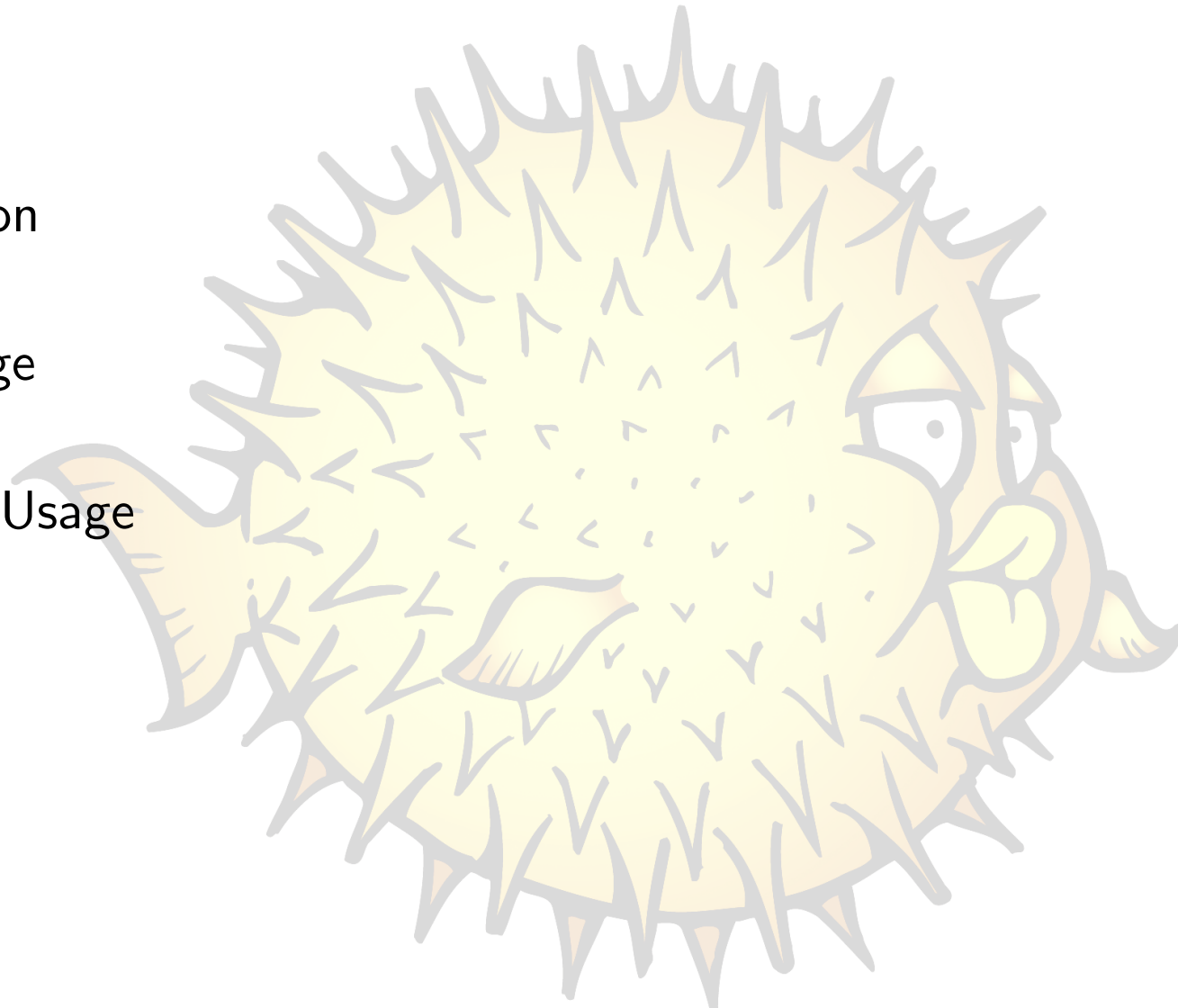
OpenSSH - Secure Shell and beyond

Elmar Hoffmann <elho@elho.net>

Linux User Group Mönchengladbach
November 14, 2006

Overview





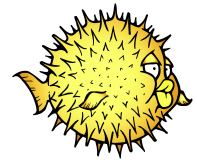
Introduction

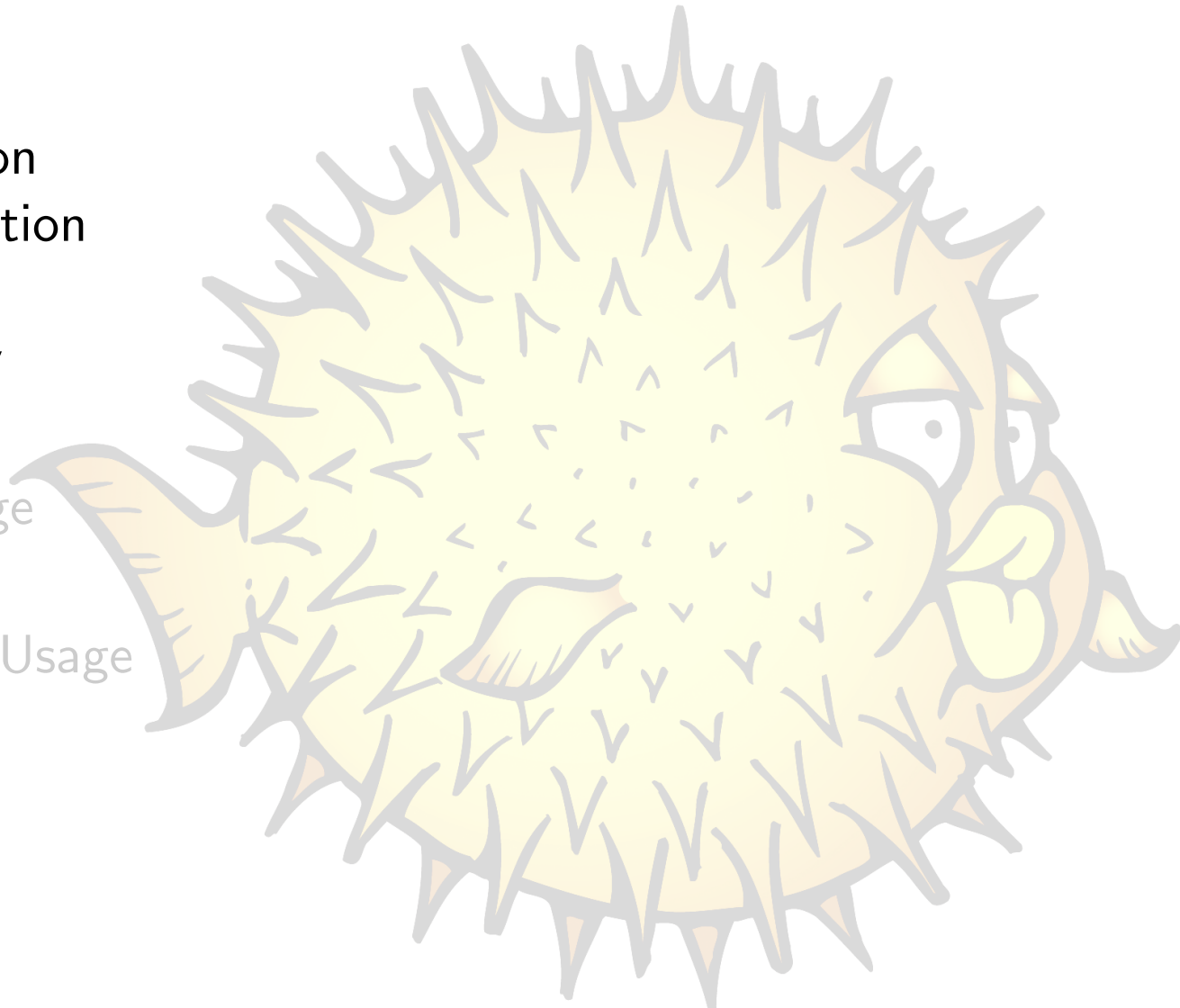
Basic Usage

Advanced Usage

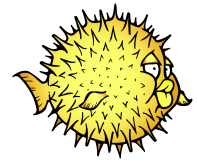
The End

Overview





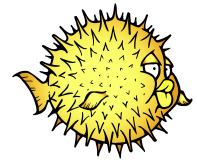
Motivation



- Use encryption and authentication when:
  - working remotely
    - telnet and rlogin replacement
  - transfering files
    - ftp alternative

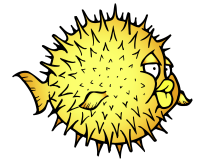
  ...

Basics



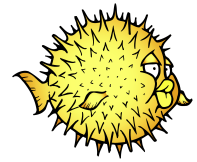
- OpenSSH flavour
  - OpenSSH
  - Portable OpenSSH
- SSH protocol version
  - SSH 1: 1.3, 1.5
    - **Vulnerable to man-in-the-middle attacks due to design flaws!**
  - SSH 2: 2.0

History

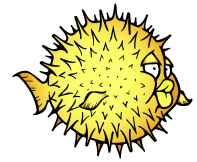


- 1999: OpenSSH 1.2.2
- 2000: OpenSSH 2.0
  - SSH 2 protocol
- 2001: OpenSSH 3.0
- 2002: OpenSSH 3.2
  - Privilege separation
- 2005: OpenSSH 4.0
- 2006: OpenSSH 4.3
  - (experimental) VPN

Overview

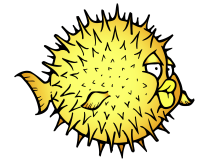


Remote Shell



- Log into remote system `host.example.com` with username `user` and execute interactive shell or given command:
  - `ssh user@host.example.com`
  - `ssh user@host.example.com -- command options`

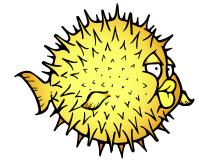
Remote Shell



- Log into remote system `host.example.com` with username `user` and execute interactive shell or given command:
  - `ssh user@host.example.com`
  - `ssh user@host.example.com -- command options`
- Authentication of remote host via host key
- Known good host keys are stored in known hosts file
  - Site: `/etc/ssh/ssh_known_hosts`
  - User: `~/.ssh/known_hosts`

```
user@foo:~$ ssh user@bar.example.com
The authenticity of host 'bar.example.com (192.168.1.1)' can't be established.
RSA key fingerprint is 24:24:e8:72:d4:3a:a0:dd:4c:20:35:29:c4:3f:88:8c.
Are you sure you want to continue connecting (yes/no)?
```

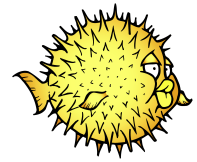
Remote Shell



- Log into remote system `host.example.com` with username `user` and execute interactive shell or given command:
  - `ssh user@host.example.com`
  - `ssh user@host.example.com -- command options`
- Authentication of remote host via host key
- Known good host keys are stored in known hosts file
  - Site: `/etc/ssh/ssh_known_hosts`
  - User: `~/.ssh/known_hosts`

```
user@foo:~$ ssh user@bar.example.com
The authenticity of host 'bar.example.com (192.168.1.1)' can't be established.
RSA key fingerprint is 24:24:e8:72:d4:3a:a0:dd:4c:20:35:29:c4:3f:88:8c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'bar.example.com' (RSA) to the list of known hosts.
Password:
```

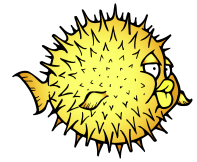
Remote Shell



- Log into remote system `host.example.com` with username `user` and execute interactive shell or given command:
  - `ssh user@host.example.com`
  - `ssh user@host.example.com -- command options`
- Authentication of remote host via host key
- Known good host keys are stored in known hosts file
  - Site: `/etc/ssh/ssh_known_hosts`
  - User: `~/.ssh/known_hosts`

```
user@foo:~$ ssh user@bar.example.com
The authenticity of host 'bar.example.com (192.168.1.1)' can't be established.
RSA key fingerprint is 24:24:e8:72:d4:3a:a0:dd:4c:20:35:29:c4:3f:88:8c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'bar.example.com' (RSA) to the list of known hosts.
Password:
user@bar:~$
```

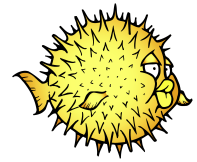
Hashed known hosts files



Hostnames and addresses in `known_hosts` files are hashed:

- Avoids disclosure of potential targets for an attacker
- Makes manual removal and updates of entries impossible
- Solution: Use `ssh-keygen(1)` to edit `known_hosts` files
  - Search hostname:
    `ssh-keygen -F hostname`
  - Hash `known_hosts` file:
    `ssh-keygen -H`
  - Remove all keys belonging to hostname:
    `ssh-keygen -R hostname`
- Introduced in OpenSSH 4.0

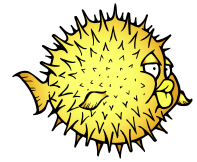
Public Key Authentication



- Allows for either:
  - Two factor authentication
  - Passwordless non-interactive authentication
    (beware of the implications!)
- Keypairs are created using `ssh-keygen(1)`:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
```

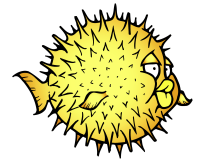
Public Key Authentication



- Allows for either:
  - Two factor authentication
  - Passwordless non-interactive authentication
    (beware of the implications!)
- Keypairs are created using `ssh-keygen(1)`:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

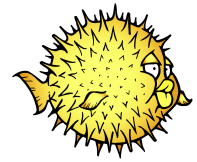
Public Key Authentication



- Allows for either:
  - Two factor authentication
  - Passwordless non-interactive authentication
    (beware of the implications!)
- Keypairs are created using `ssh-keygen(1)`:

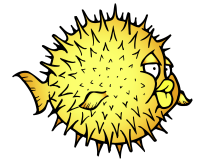
```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
42:dc:9f:7d:d7:c2:37:4a:9e:27:e0:da:ba:03:4d:71 user@host
```

Public Key Authentication II



- Authorised keys file lists public keys that are allowed to log in `~/.ssh/authorized_keys`
- To grant yourself access, append the contents of your `~/.ssh/id_rsa.pub` to it

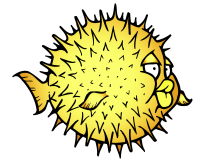
Configuration files



- Client configuration file
  - Site: `/etc/ssh/ssh_config`
  - User: `~/.ssh/config`
- Allows global and per host settings:

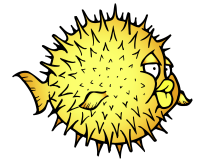
```
Compression yes
CompressionLevel 9

Host *.lan.example.com
    Compression no
```

X11 Forwarding



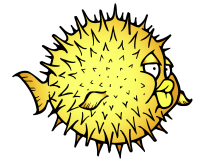
- Allow remote usage of applications with a graphical user interface
- Has to be enabled, **only enable for trusted hosts!**
  - `ssh -X host.example.com`
  - `ForwardX11 yes` configuration option

Secure File Transfer

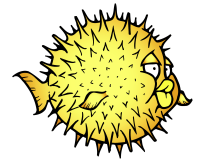


- Secure Copy
  `scp file host.example.com:path/`
- Secure File Transfer Program
  `sftp host.example.com`
- rsync
  `rsync --archive dir host.example.com:path/`

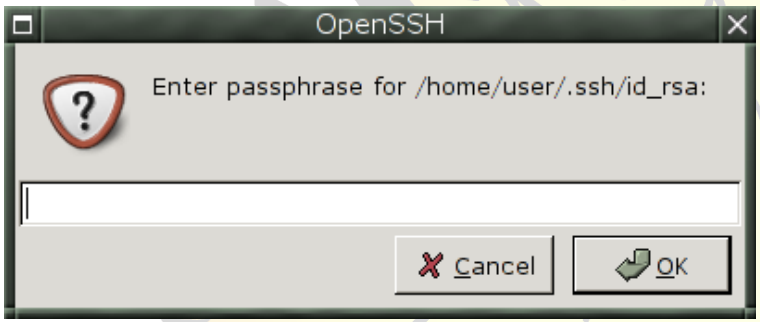
Overview



SSH-Agent



- Cache decrypted private keys
- Graphical passphrase dialog via `ssh-askpass(1)`
  - ssh-askpass
  - ssh-askpass-gnome

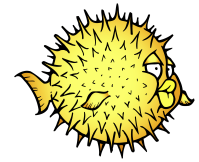


Automatically start ssh-agent on session login:

- From Xsession
- From `~/.bash_profile` or similar:
  `eval `ssh-agent``
- Using keychain

  `http://www.gentoo.org/projects/keychain/`

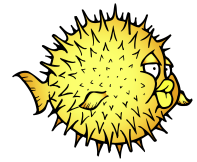
SSH-Agent II



- Key management using `ssh-add(1)`:
  - Add key to the agent:
    `ssh-add keyfile`
  - List keys in the agent:
    `ssh-add -l`
  - Remove all keys from the agent:
    `ssh-add -D`

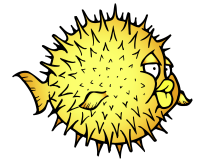
```
$ ssh-add
Enter passphrase for /home/user/.ssh/id_rsa:
Identity added: /home/user/.ssh/id_rsa (/home/user/.ssh/id_rsa)
$ ssh-add -l
2048 42:dc:9f:7d:d7:c2:37:4a:9e:27:e0:da:ba:03:4d:71 /home/user/.ssh/id_rsa (RSA)
```

SSH-Agent Forwarding



- Allow use of local agent on remote host
- Has to be enabled, **only enable for trusted hosts!**
  - `ssh -A host.example.com`
  - `ForwardAgent yes` configuration option

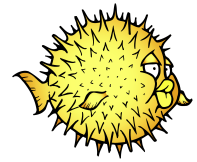
Sending Environment Variables



Allows automatic setting of environment variables on the remote host to the same values as on the local host.

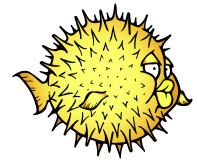
- Useful for locale environment variables
- Configuration options:
  - Server: `AcceptEnv`
  - Client: `SendEnv`
- Introduced in OpenSSH 3.8

Port Forwarding



- Static Port Forwarding
  - Forward local to remote port:
    `ssh -L port:host:hostport host.example.com`
    `LocalForward` configuration option
  - Forward remote to local port:
    `ssh -R port:host:hostport host.example.com`
    `RemoteForward` configuration option
- Dynamic Port Forwarding
  - SOCKS server listening on local port:
    `ssh -D port host.example.com`
  - Introduced in OpenSSH 3.0

Proxy Commands



- External command used to establish connection to server
- `ProxyCommand` configuration option
  - Connection through HTTP proxy
    - corkscrew

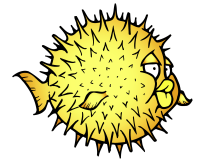
      `http://www.agroman.net/corkscrew/`

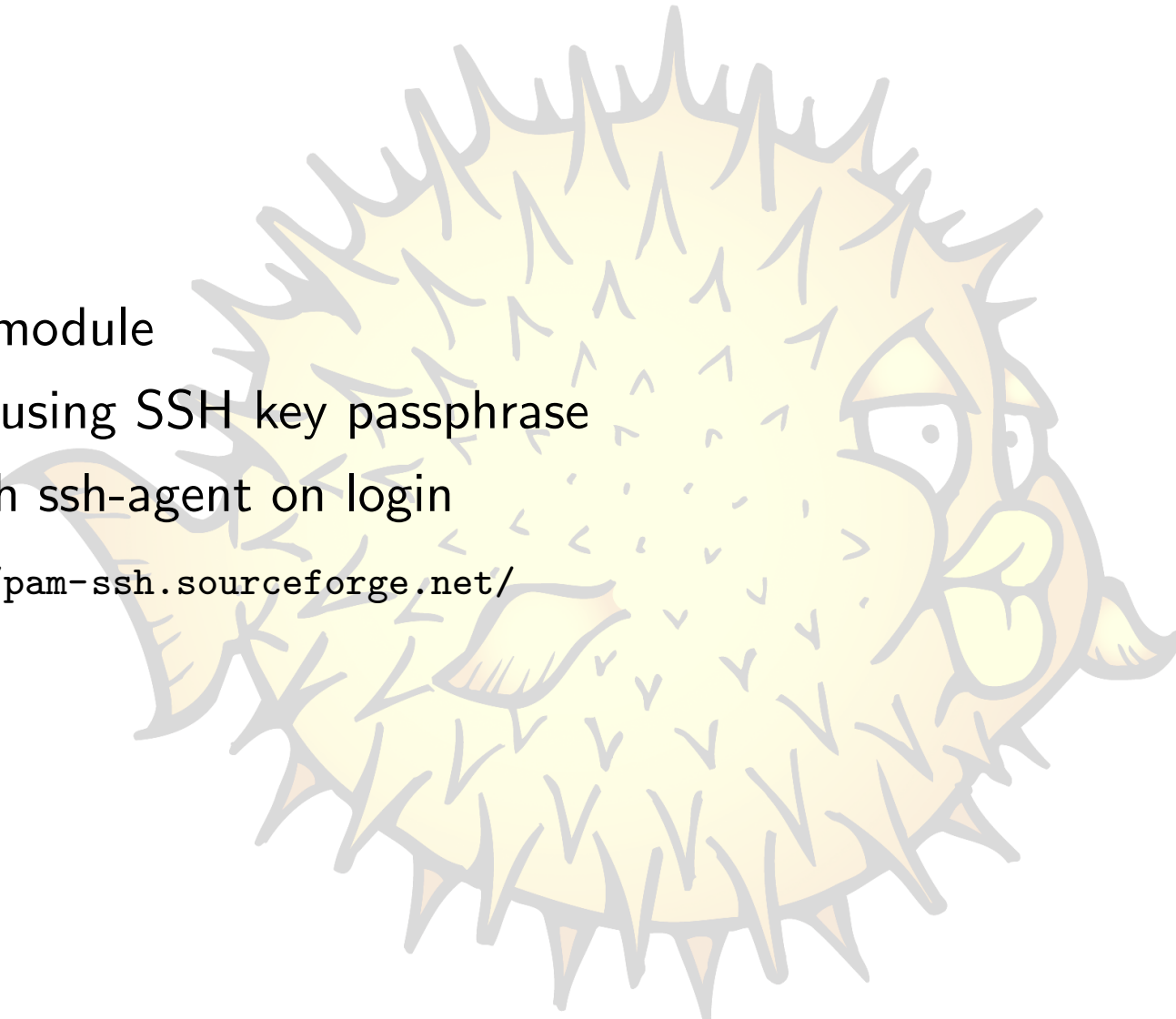
    ```
    ProxyCommand /usr/bin/corkscrew proxy.example.com 8080 %h %p
    ```

  - Connection through SSH gateway

    ```
    Host internal.example.com
        ProxyCommand /usr/bin/ssh user@gateway.example.com nc -q 0 %h %p
    ```

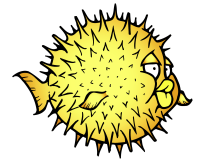
Local Shell

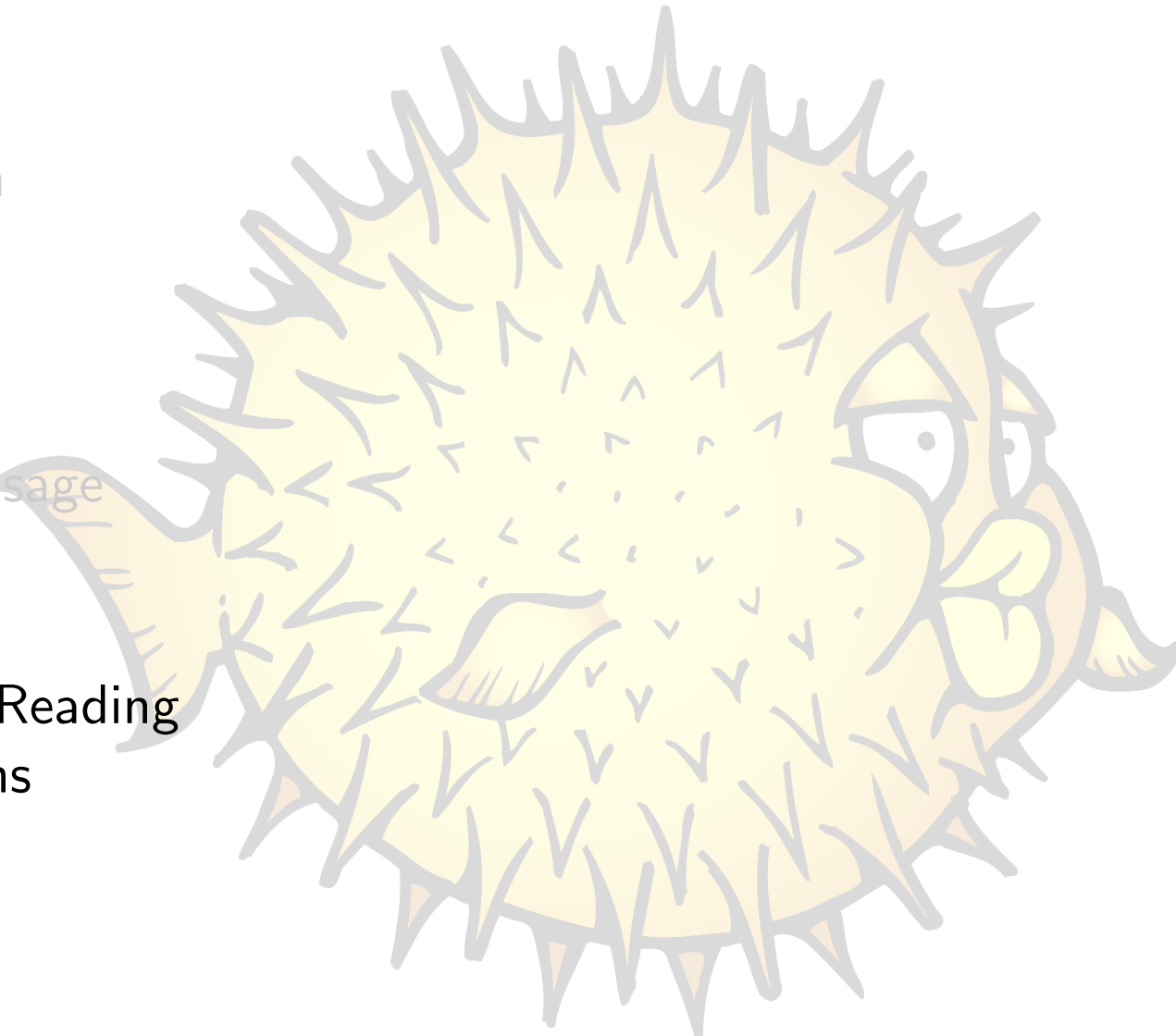




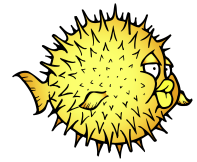
- PAM module
- Login using SSH key passphrase
- Launch ssh-agent on login
- `http://pam-ssh.sourceforge.net/`

Overview





Further Reading



- OpenSSH Manuals

  `http://www.openssh.com/manual.html`
- OpenSSH FAQ

  `http://www.openssh.com/faq.html`
- OpenSSH key management

  `http://www.ibm.com/developerworks/library/l-keyc.html`

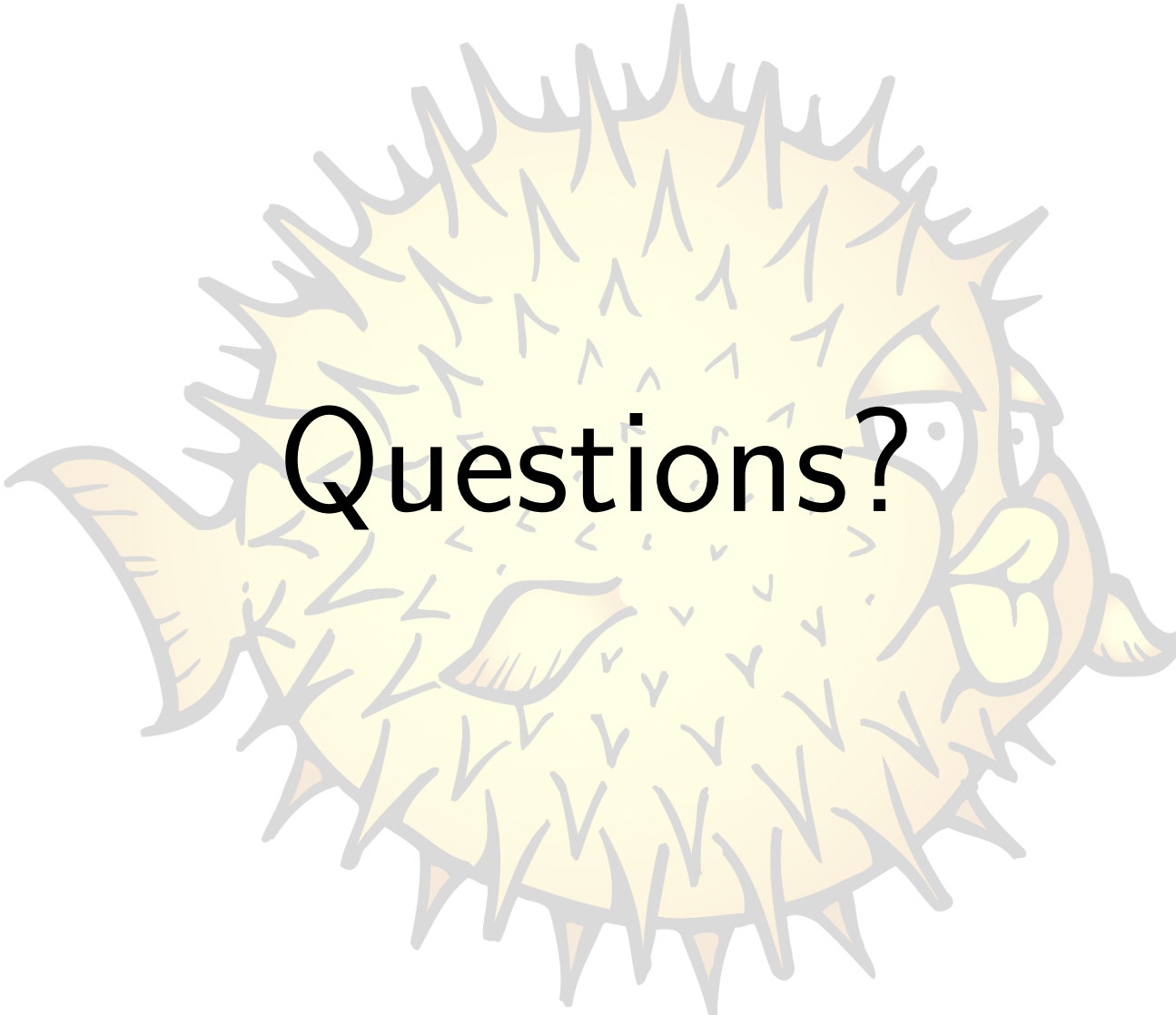
  `http://www.ibm.com/developerworks/library/l-keyc2/`

  `http://www.ibm.com/developerworks/library/l-keyc3/`

- OpenSSH - Secure Shell and beyond

  `http://www.elho.net/pub/`

Questions



Questions?