Secure Authentication
A Brief Overview

PacNOG I Workshop

June 22, 2005
Nadi, Fiji

Hervey Allen

What are we talking about?

Any service you run that authenticates *should not* do so in the clear. This includes:

- pop
- imap
- shell login
- file transfer
- web login (think webmail)
- sending (think smtp)

Some replacements

- POP ==> POPS with ssl cert (port 110 vs. 995)
- imap ==> imaps with ssl cert (port 143 vs. 993)
- smtp authed with TLS (port 465/other vs. 25)
- telnet ==> ssh
- ftp ==> sftp or scp
- http login via https with ssl cert
- http upload is harder
- anonymous ftp is OK. Watch uploads

Avoiding the ssh tunnel

SSH tunneling is cool and powerful, but can circumvent some secure practices and is hard for most users.

You can use pops, imaps, and smtp with tls to remove the need for most ssh tunnels.

This can avoid the need for users doing this.

```
ssh -C -f username@host.domain -L 1100:localhost:110 sleep 10000
```

It can be painful...

Windows has no built-in ssh/sftp/scp client. This can make secure shell login requirements painful.

For secure web login simply force the login page to be https. Most scripting and programatic interfaces make this easy.

In PHP:

```
if ($_SERVER["HTTPS"] != 'on')
    {
      header("Location: https://" .$_SERVER['SERVER_NAME'] \
  .$_SERVER['PHP_SELF']."?referrer=$referrer");
    }
```

But, it's worth it

Start to get your user community used to the idea of “no passwords in the clear”

Has the potential to steer your organization clear of potential liability issues in the future.

You'll sleep better at night... ;-)