

# Securing NFSv3 with IPsec

Nikos Gkorogiannis

21/2/2007

NFS

IPsec

Putting the two together

Please, be considerate to others in relation to what you learn in this talk, ahem.

## Case study

- ▶ CS department with embedded computing Linux labs
- ▶ Networked embedded-computing boards running Linux
- ▶ Linux server with user files
- ▶ No budget :)

# NFS

# NFS

# What is NFS

- ▶ Network File System (built on top Sun's RPC)
- ▶ for Unix-like Operating Systems (virtually all \*nices)
- ▶ *Stateless*
- ▶ History:

Never released NFSv1

Early 1980s NFSv2 – basic, popular and compatible

Early 1990s NFSv3 – more functionality, by now ubiquitous

? NFSv4 – all the above plus security! but when?

# NFS services

Services comprising NFSv3 (separate RPC daemons):

- ▶ **MOUNT**
- ▶ **NFS**
- ▶ STAT
- ▶ LOCK
- ▶ QUOTA
- ▶ Only MOUNT and NFS are required.

# How does it work?

A typical message sequence:

CLIENT

EXPORT →

MNT /home →

READ /home/email →

SERVER

← /home, /usr

← FSID=0xFA...

← DATA='From...'

# NFSv3 authentication

Two main modes:

- ▶ AUTH\_UNIX, credentials=UID, GID
- ▶ AUTH\_DES, AUTH\_KERB, credentials=crypto hash
- ▶ For v3, only AUTH\_UNIX is popular, others are non-standard.  
(proprietary extensions)

Constrast CIFS,

- ▶ Credentials are a Kerberos ticket.
- ▶ Cf mounting filesystems as a user.

## A typical attack on NFSv3

- ▶ Attacker connects a Linux laptop to the server's network.
- ▶ Discovers what mountpoints are exported by the NFS server.
- ▶ Mounts one on the laptop.
- ▶ Issues `su` and assumes the identity of a valid user.
- ▶ Freely reads and writes files belonging to that user.

Can we fix this?

# Why use it at all?

“You mean people are still using this?”

- ▶ Ubiquitous (standard).
- ▶ Fast.
- ▶ Simple!
- ▶ Good for private networks.

# IPsec

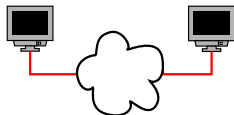
# IPsec

# What is IPsec

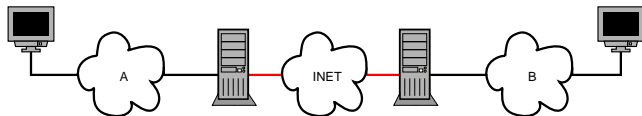
- ▶ An IP-based protocol (in fact, two).  
ICMP, TCP, UDP, **IPsec:ESP**, **IPsec:AH**, etc
  - ▶ Therefore, **not** user level (SSL) but IP stack/OS-level.
- ▶ With two main modes, Tunnel and Transport
- ▶ And loads of crypto methods:  
MD5, SHA-1, DES, 3DES, AES etc
- ▶ Internet Key Exchange (IKE)

## IPsec modes

- ▶ Transport mode (end-to-end, no encapsulation)

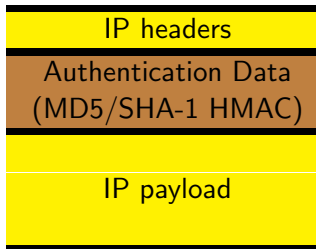


- ▶ Tunnel mode (think VPN, most popular)

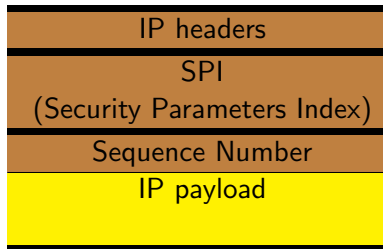


# AH vs ESP

## AH (transport)



## ESP (transport)



Hashed or encrypted

**Not** hashed or encrypted

# Mechanics

## Security Policy DB

- ▶ **Src & dst IP address**
- ▶ **Ipssec proto (AH and/or ESP)**
- ▶ **AH type and info**
- ▶ **ESP type and info**
- ▶ tunnel/transport mode flags
- ▶ and others!

## Security Association DB

- ▶ **Everything in SPD**
- ▶ **Key info**
- ▶ Sequence counter
- ▶ Lifetime info
- ▶ and others!

# Putting the two together

## Putting the two together

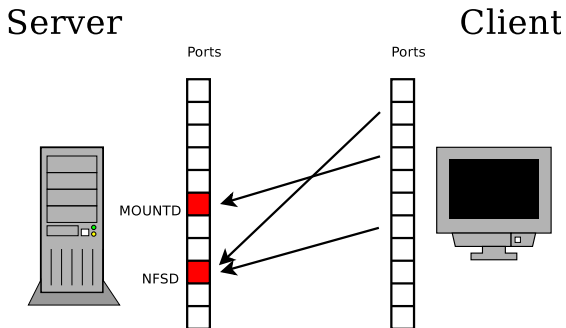
## Defining the problem

- ▶ NFSv3 offers no better security than host-based, ie the server trusts that the client authenticates the user.
- ▶ We could write software integrating IPsec at the user level, but that would become a security nightmare.
- ▶ ⇒ Ensure that the client is a trusted one, as far as client-generated NFSv3 traffic is concerned.

## How? (1)

- ▶ How can we know that NFSv3 packets received by the server are coming from authorised clients?
  - ▶ Use a shared secret (key) between server and client and AH on NFSv3 packets from client to server.
- ▶ ⇒ Create rules that cover packets **to** the server's MOUNTD or NFSD ports, forcing the use of AH and discard non-AH packets.

## How? (2)



# The nitty-gritty

- ▶ Individual keys for each client?
- ▶ Key management/revocation?
- ▶ Throughput?
- ▶ Phasing-in

## References

### NFS

- ▶ <http://nfs.sourceforge.net/> including “NFS HOWTO”
- ▶ <http://www.unix.org.ua/oreilly/networking/puis/>  
Chapter 20 on NFS from “Practical UNIX & Internet security”

### IPsec

- ▶ <http://www.commsdesign.com/showArticle.jhtml?articleID=192200444>
- ▶ <http://www.netbsd.org/Documentation/network/ipsec/>
- ▶ <http://www.unixwiz.net/techtips/iguide-ipsec.html>