Connectathon ‘99

WebNFS Security Negotiation

Alex Chiu

3/8/99

WebNFS Security Negotiation

- Why WebNFS security negotiation?
- Requirements
- WebNFS Security Negotiation
- References

Why WebNFS Security Negotiation?

- **Existing security negotiation for NFS**
  - **NFS V3/MOUNT V3 [RFC 1813]**

Client **Server**

---------- MOUNT port ? ----------> Portmapper
<---------- Port=984 ----------

------- Filehandle for /export/foo ? ----> Mountd @ port 984
<----- Filehandle=0xf82455ce0, RPC auth flavors ------

(MOUNTPROC3_MNT procedure)

Why WebNFS Security Negotiation?

- NFS V2/MOUNT V2 [RFC1094]

  MOUNTPROC_MNT only returns NFS V2 filehandles

  No RPC procedures available for security negotiation

- WebNFS

  - Some Highlights [RFC's 2054 (client), 2055 (server)]

    Eliminates overhead of PORTMAP and MOUNT

    Eases firewall transit

Why WebNFS Security Negotiation?

Multi-component LOOKUP: reduces the number of LOOKUP requests

WebNFS Client		WebNFS Server
LOOKUP FH=0x0, "a/b/c"	---->	
	<----	FH=0x3

- **WebNFS client initially assumes WebNFS support on server, i.e., not using MOUNT**
- **WebNFS client falls back to MOUNT if the server does not support WebNFS**

Why WebNFS Security Negotiation?

- **Problem**

  Server: # share -o sec=dh,public /export

  Client: What security mechanism should I use to access server?

- **Solution: WebNFS security negotiation**

Requirements

- **Must work seamlessly with NFS V2 and V3, and the WebNFS protocols**
- **Must be backward compatible with servers that do not support this negotiation**
- **Minimum number of network turnarounds (latency)**

WebNFS Security Negotiation

- **WebNFS Multi-component LOOKUP [RFC2054]**

WebNFS Client		WebNFS Server
LOOKUP FH=0x0, "path"	---->	
	<----	FH=0x3

- **WebNFS Security Negotiation Multi-component LOOKUP (SNEGO-MCL)**

WebNFS Client		WebNFS Server
LOOKUP FH=0x0, *0x81*, <*sec-index*>, "path"		
	---------->	
	<----------	
		FH: *length, status, {sec_1 sec_2 ... sec_n}*

WebNFS Security Negotiation

0x81 **(1 octet, non-ascii): client's indication to negotiate security mechanisms**

sec-index **(1 octet): stores the current value of the index into the array of security mechanisms to be returned from the server;** ***sec-index*** **begins with one.**

status **(1 octet): indicates whether there are more security mechanisms (1 means yes, 0 means no) that require the client to perform another SNEGO-MCL to get them**

length**: describes the number of valid octets that follow**

{sec_1 sec_2 ... sec_n}**: array of security mechanisms. Each security mechanism is represented by a four-octet integer.**

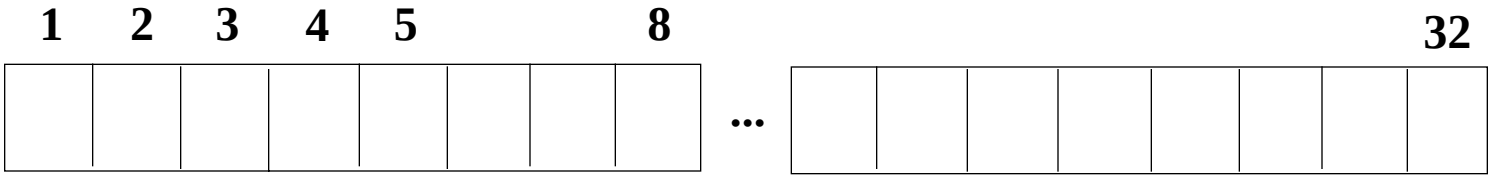
WebNFS Security Negotiation

- **Overloaded Filehandle**

NFS V2 and V3 filehandles are “overloaded” to return server’s security info: *length, status, {sec_1 sec_2 ... sec_n}*

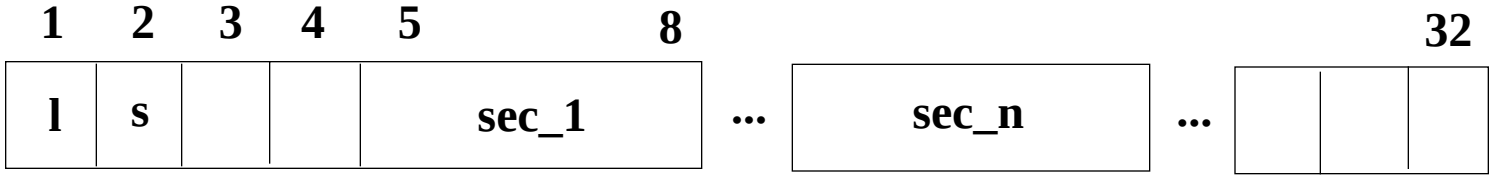
- **NFS V2**

- NFS V2 Filehandle [RFC1094]



WebNFS Security Negotiation

- Overloaded NFS V2 filehandle



first four octets: length octet, l = 4 * n, (n: size of the array of the security mechanisms),
status octet,
two padded octets to make XDR four-octet aligned

{sec_1, sec_2, ..., sec_n}: array of up to 7 security mechanisms sent in the current overloaded filehandle

WebNFS Security Negotiation

- Overloaded NFS V2 filehandle Example

If an NFS V2 server shares /export with 10 security mechanisms, two SNEGO-MCL requests would be needed for the client to get the complete security information:

LOOKUP FH=0x0, *0x81, 0x01,* "/export"

---------->

<----------

0x1c, 0x01, {0x3900 0x3901 0x3902 0x3903 0x3904 0x3905 0x3906}

LOOKUP FH=0x0, *0x81, 0x08,* "/export"

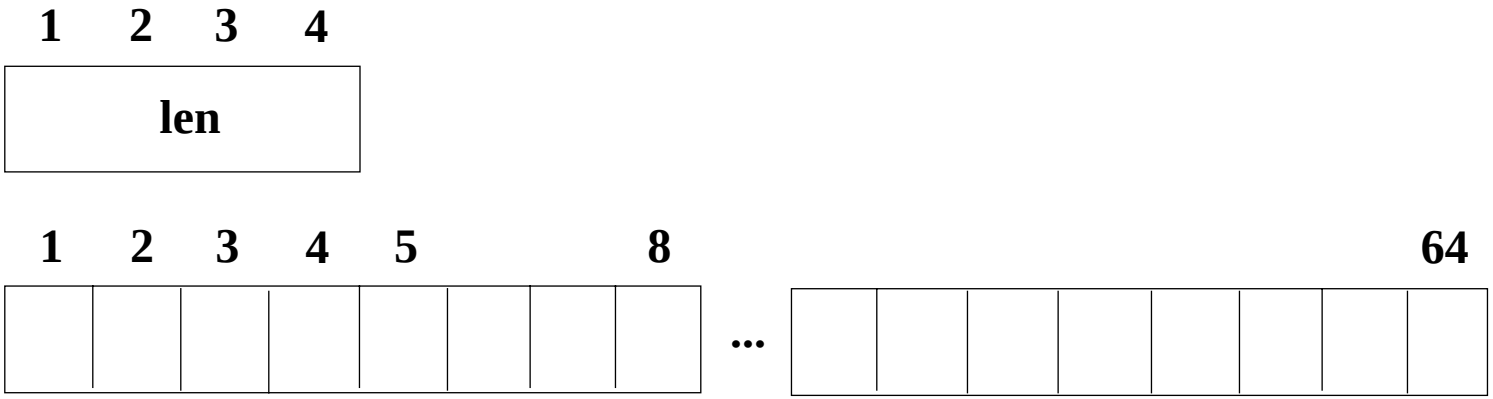
---------->

<----------

0x0c, 0x00, {0x3907 0x3908 0x3909}

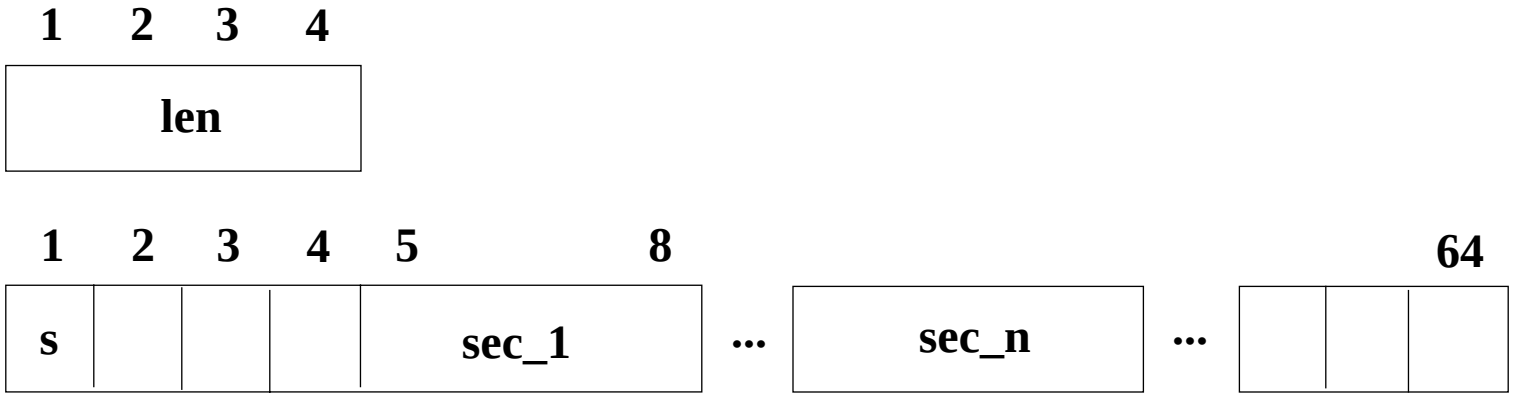
WebNFS Security Negotiation

- **NFS V3**
  - **NFS V3 Filehandle [RFC1813]**



WebNFS Security Negotiation

- Overloaded NFS V3 filehandle



len = 4 * (n + 1)

first four octets of filehandle: status octet, three padded octets to make XDR four-octet aligned

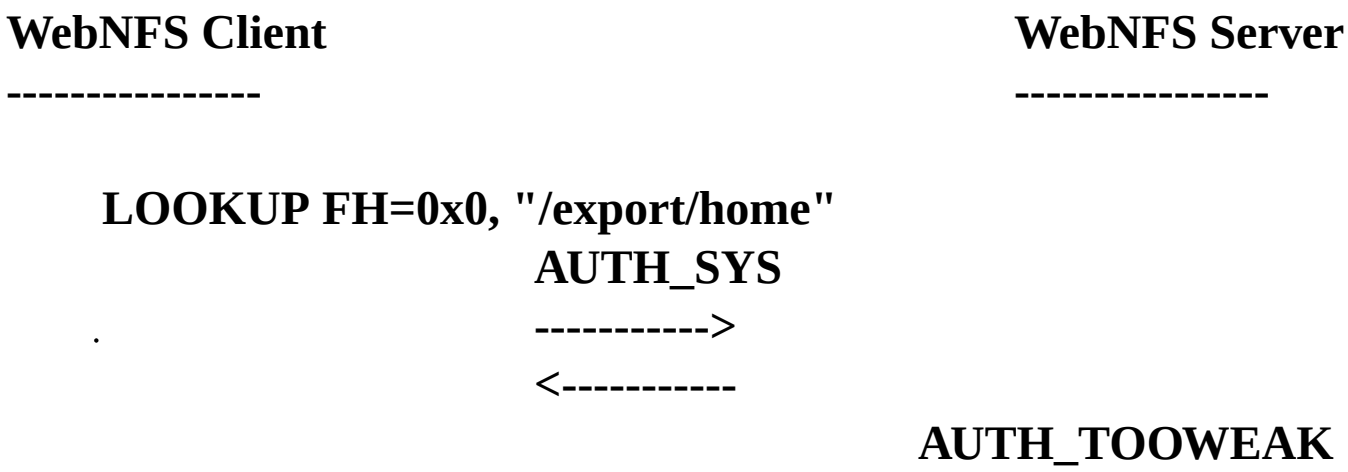
{sec_1, sec_2, ..., sec_n}: array of up to 15 security mechanisms sent in the current overloaded filehandle

WebNFS Security Negotiation

WebNFS Security Negotiation Example

Server: share -o sec=sec_1:sec_2:sec_3,public /export/home

Assume AUTH_SYS is the default security mechanism and is not one of {sec_1, sec_2, sec_3}.



WebNFS Security Negotiation

LOOKUP FH=0x0, 0x81, 0x01, "/export/home"

AUTH_SYS

---------->

<----------

overloaded FH: length, status, {sec_1 sec_2 sec_3}

LOOKUP FH=0x0, "/export/home"

sec_n

---------->

<----------

FH = 0x01

NFS request with FH=0x01

sec_n

---------->

<----------

...

References

[RFC1094] Sun Microsystems, Inc., "Network Filesystem Specification", RFC 1094, March 1989. NFS version 2 protocol specification.
http://www.internic.net/rfc/rfc1094.txt

[RFC1813] Sun Microsystems, Inc., "NFS Version 3 Protocol Specification", RFC 1813, June 1995. NFS version 3 protocol specification.
http://www.internic.net/rfc/rfc1813.txt

[RFC2054] Callaghan, B., "WebNFS Client Specification", RFC 2054 October 1996.
http://www.internic.net/rfc/rfc2054.txt

[RFC2055] Callaghan, B., "WebNFS Server Specification", RFC 2055, October 1996.
http://www.internic.net/rfc/rfc2055.txt