BSides Canberra 2016

Hacking Fibre Channel Networks

Kylie
Network Engineer
Cisco Systems Australia

Intro to Kylie

- Kylie Peak @kylieengineer

Work:

- Currently working for Cisco Australia in Advanced Services
- Previously worked for Government and Telstra
- 15 years of working on/building networks

Education:

- ANU Engineering, IT Masters CSU

Hobbies:

- BSides Canberra organiser
- Monthly Security meetups at ANU
- Ruxcon host for the past 2 years

This talk comes from building networks and noticing a lack of security

My comments are my own & not those of my employer

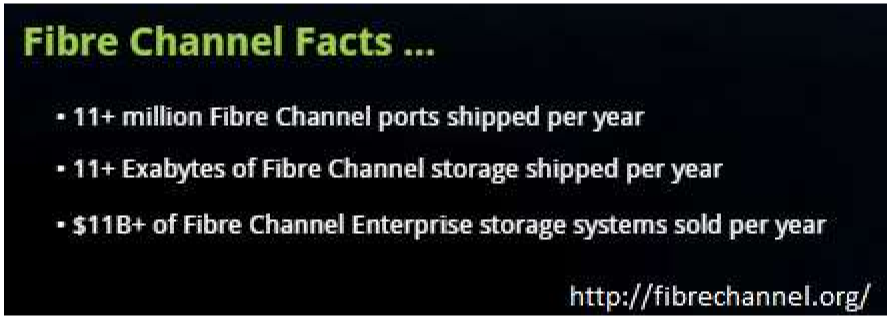
I also love gifs!!

Talk Outline

- General Introduction to Storage Networks & the FC protocol
- How to build a storage network for Research
- Security Features & weaknesses of FC
- Demonstration of Attacks
- Mitigations
- Questions

What is FC? What is it Used For?

- A SAN (Storage Area Network) is a term that defines all the Hardware and Software infrastructure that allows a computer to access storage that is not directly attached to it
- SCSI is a common block level storage protocol implemented in most OS'
- FC encapsulates the SCSI protocol for transport across a fabric

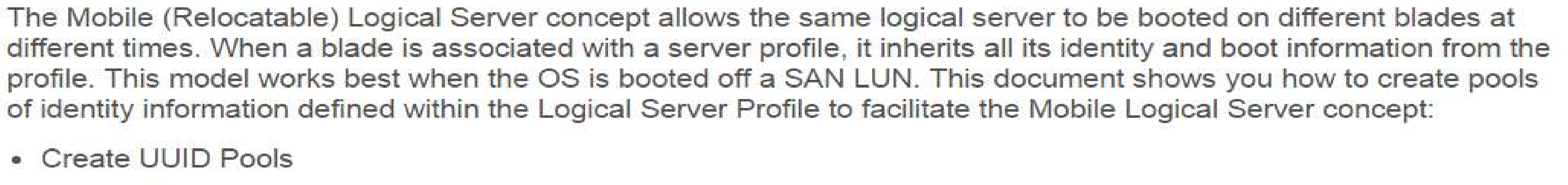


Fibre Channel Facts ...

- 11+ million Fibre Channel ports shipped per year
- 11+ Exabytes of Fibre Channel storage shipped per year
- $11B+ of Fibre Channel Enterprise storage systems sold per year

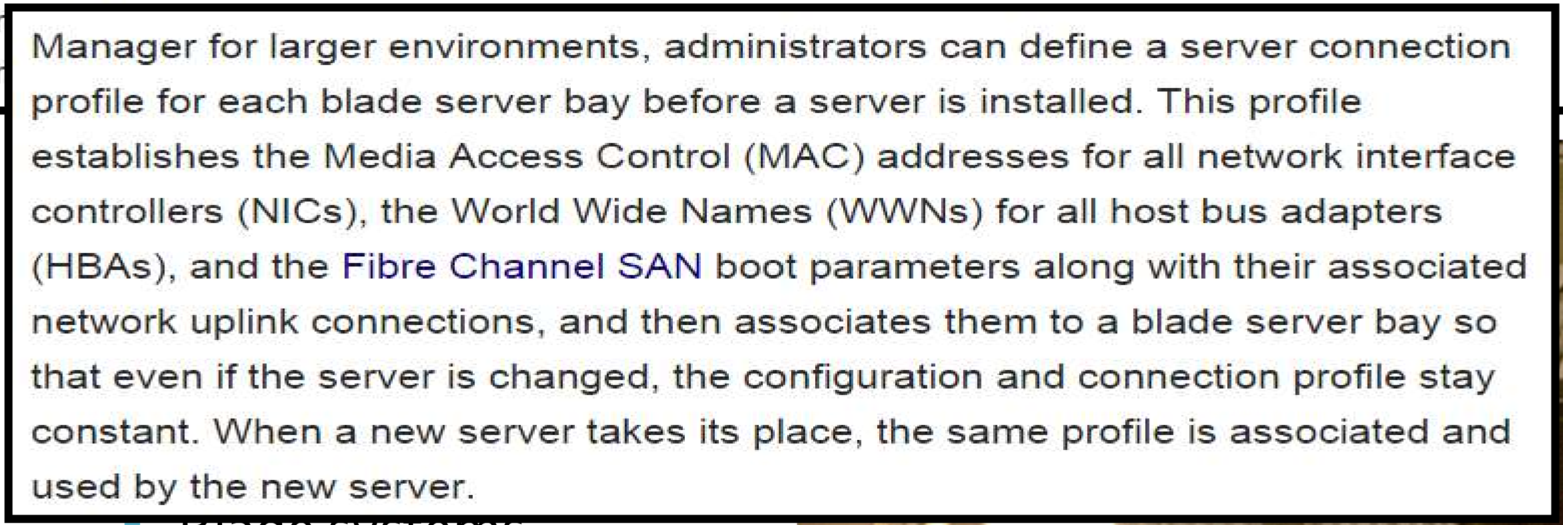
http://fibrechannel.org/

Why look at Fibre Channel (FC) security?



The Mobile (Relocatable) Logical Server concept allows the same logical server to be booted on different blades at different times. When a blade is associated with a server profile, it inherits all its identity and boot information from the profile. This model works best when the OS is booted off a SAN LUN. This document shows you how to create pools of identity information defined within the Logical Server Profile to facilitate the Mobile Logical Server concept:

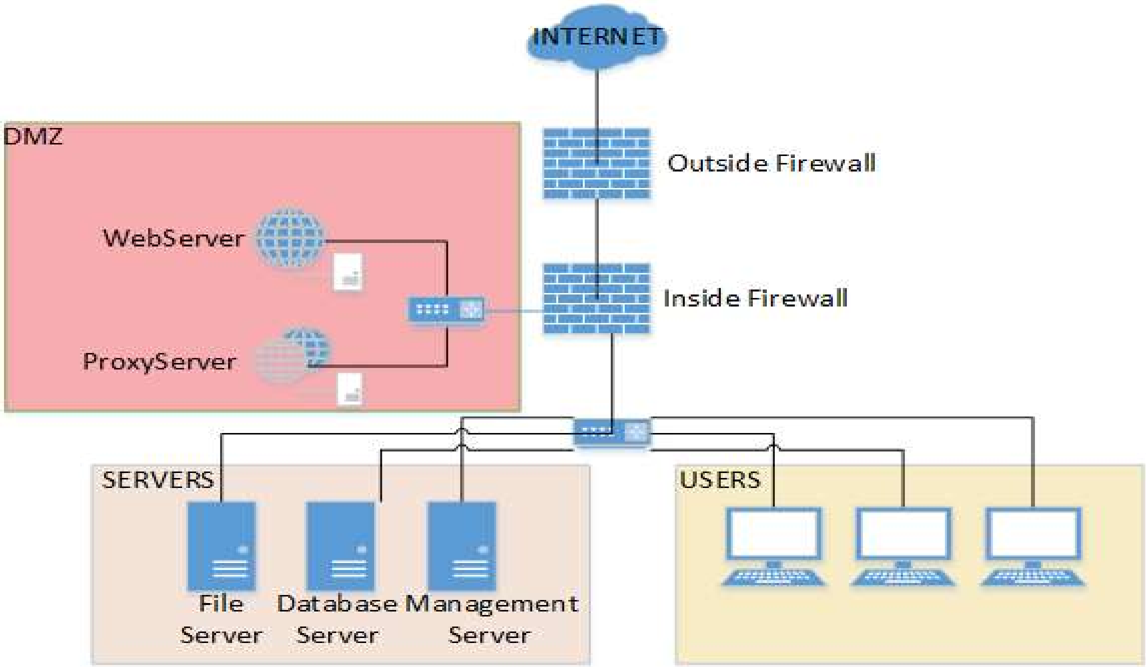
- Create UUID Pools
- Cr
- Cr



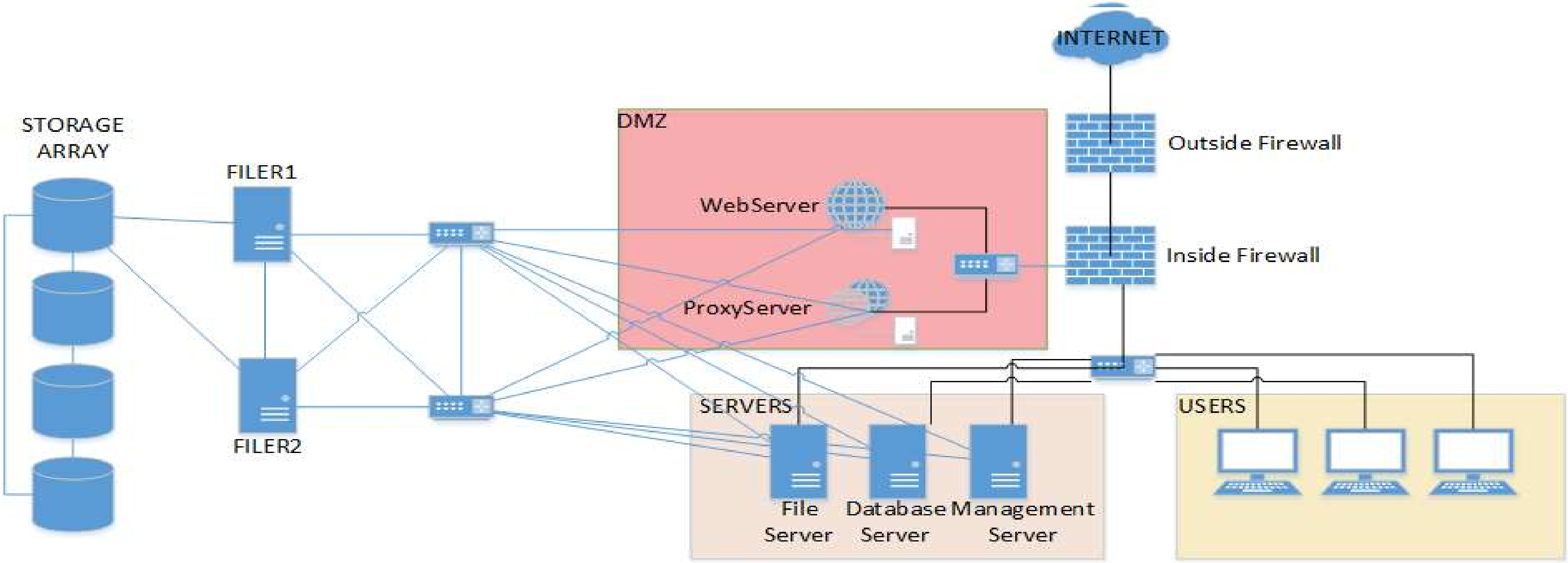
Manager for larger environments, administrators can define a server connection profile for each blade server bay before a server is installed. This profile establishes the Media Access Control (MAC) addresses for all network interface controllers (NICs), the World Wide Names (WWNs) for all host bus adapters (HBAs), and the Fibre Channel SAN boot parameters along with their associated network uplink connections, and then associates them to a blade server bay so that even if the server is changed, the configuration and connection profile stay constant. When a new server takes its place, the same profile is associated and used by the new server.

- Blade systems

What seems like a secure network…



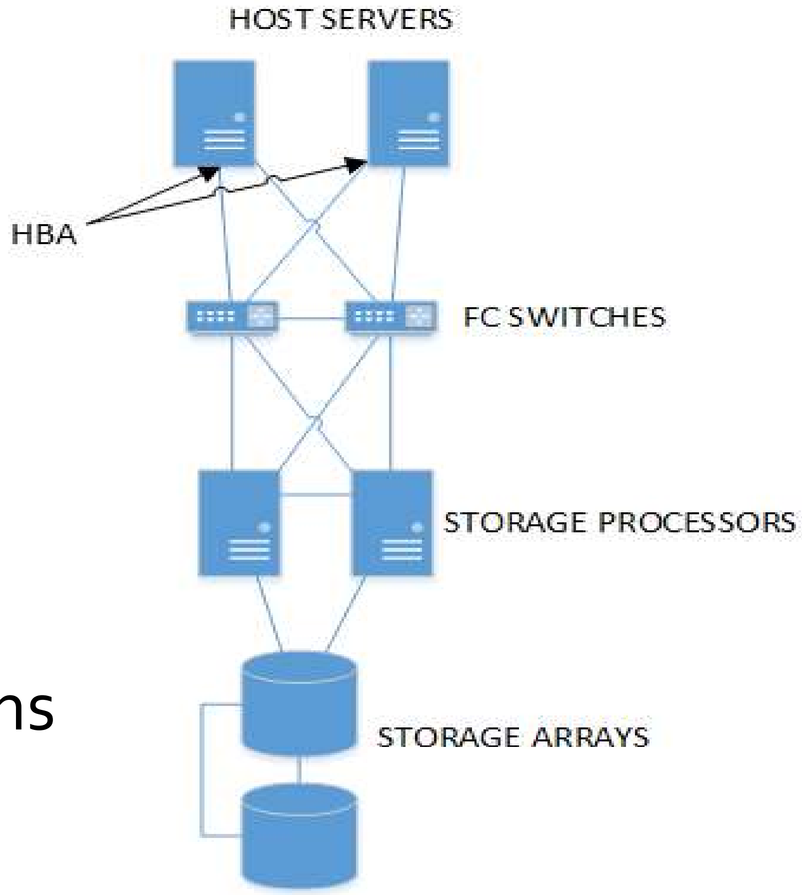
May not be…


INTERNET
Outside Firewall
Inside Firewall
DMZ
WebServer
ProxyServer
STORAGE ARRAY
FILER1
FILER2
SERVERS
File Server
Database Server
Management Server
USERS

FC Basics – SAN components

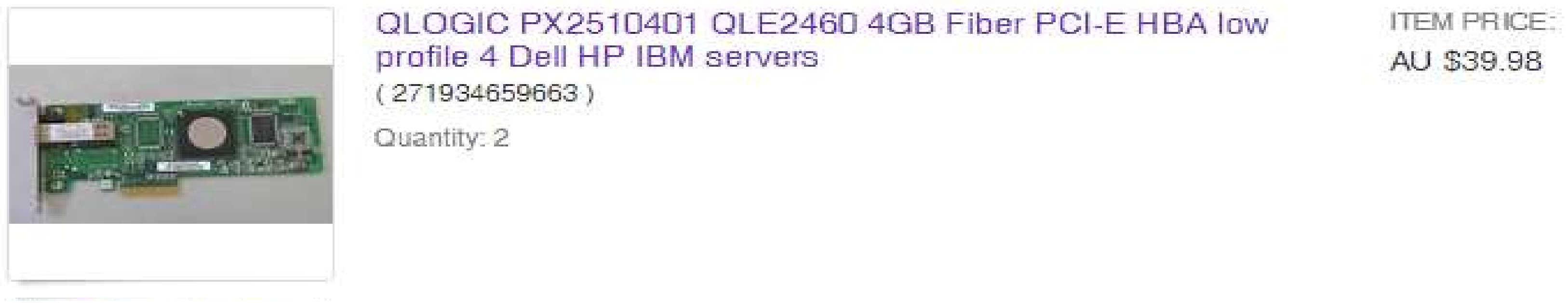
- host bus adapters (HBAs) in the host servers
- switches that help route storage traffic
- cables
- storage processors (SPs)
- storage disk arrays

A SAN topology with at least one switch present on the network forms a SAN fabric.



Buying these components

Lab Component	Cost
Server x 2	$598
FC switch	$50
SFPs x 2	$7
LC-LC Fibre x 2	$8.50
QLogic HBAs x 2	$39.98
GRAND TOTAL	$703.48



QLOGIC PX2510401 QLE2460 4GB Fiber PCI-E HBA low profile 4 Dell HP IBM servers

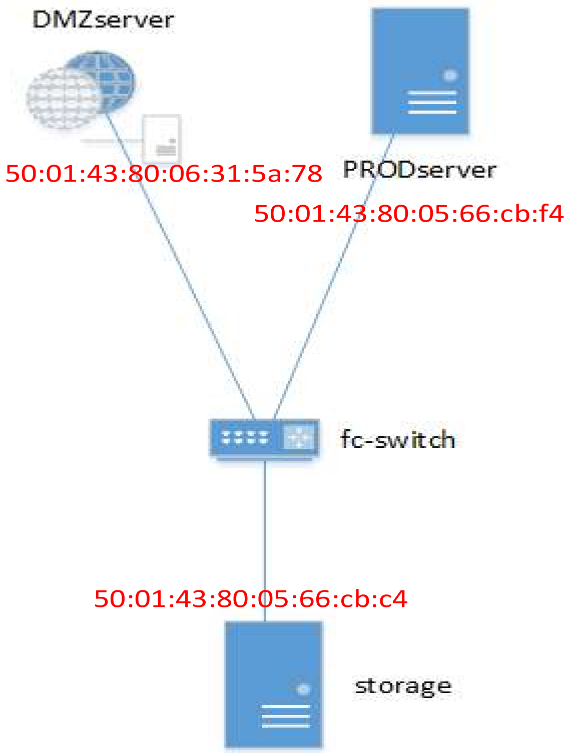
(271934659663)

Quantity: 2

ITEM PRICE:

AU $39.98

Putting it together



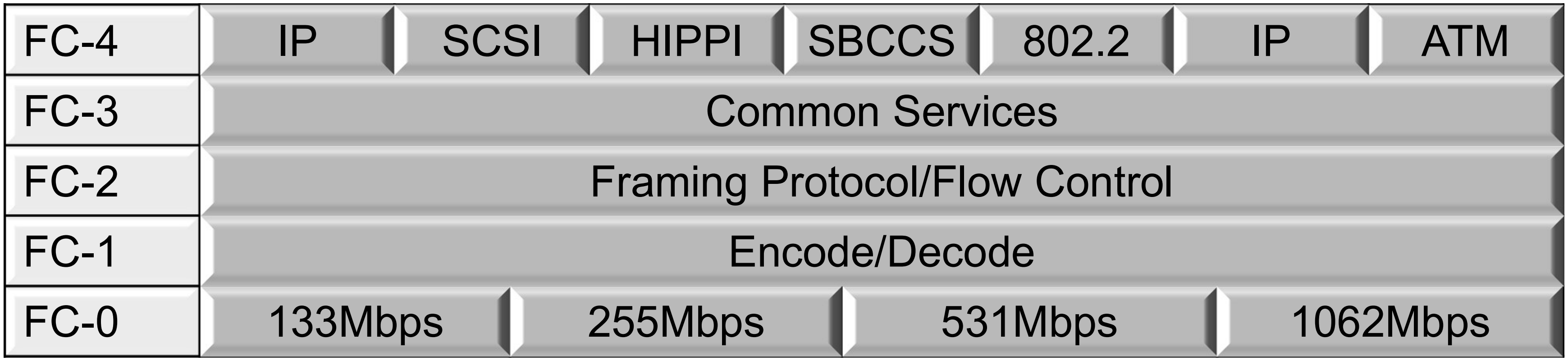
FC Security snapshot

Authentication	✗	In most cases doesn't exist
Authorization	✓	Provide through World Wide Names (WWN) and zoning
Encryption	✗	FC uses no encryption at any layer, all in the clear

Attack Investigation

- Name Server (NS) pollution
- Session hijacking – seq_id weakness
- E-port negotiation
- WWN spoofing – bypass wwn-based zoning & lun-masking

FC Protocol Layers

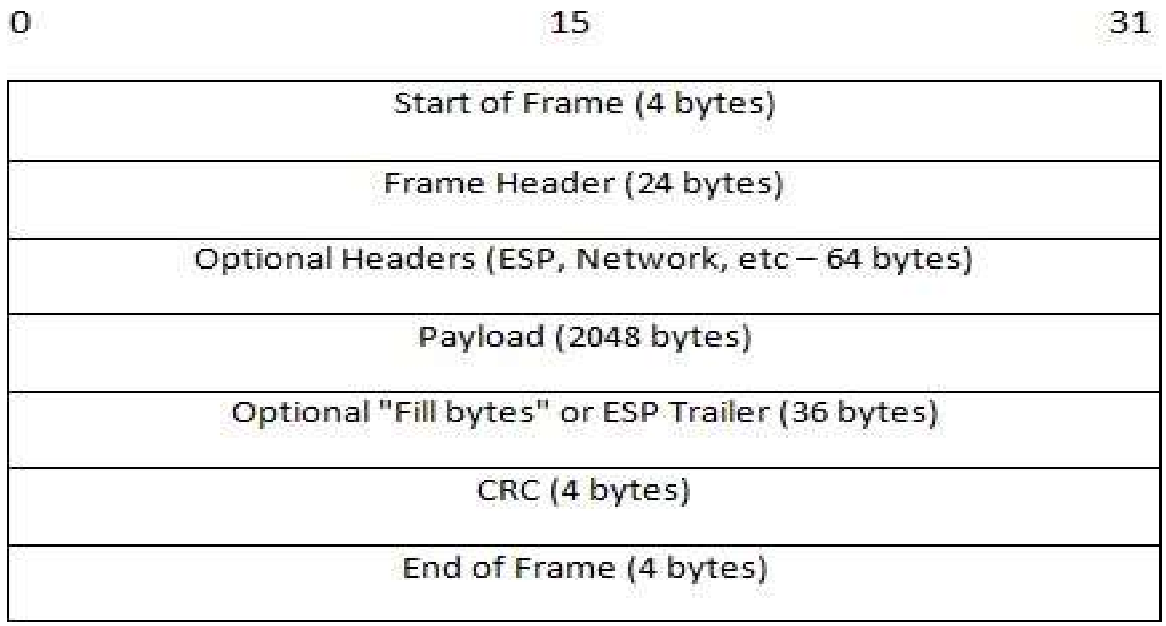


FC-4	IP	SCSI	HIPPI	SBCCS	802.2	IP	ATM
FC-3	Common Services						
FC-2	Framing Protocol/Flow Control						
FC-1	Encode/Decode						
FC-0	133Mbps	255Mbps	531Mbps	1062Mbps			

- To transfer traffic from host servers to shared storage, the SAN uses the Fibre Channel (FC) protocol that packages SCSI commands into Fibre Channel frames.

- FC-2 is where most of the weaknesses in FC are found

FC-2 Frame



0 15 31

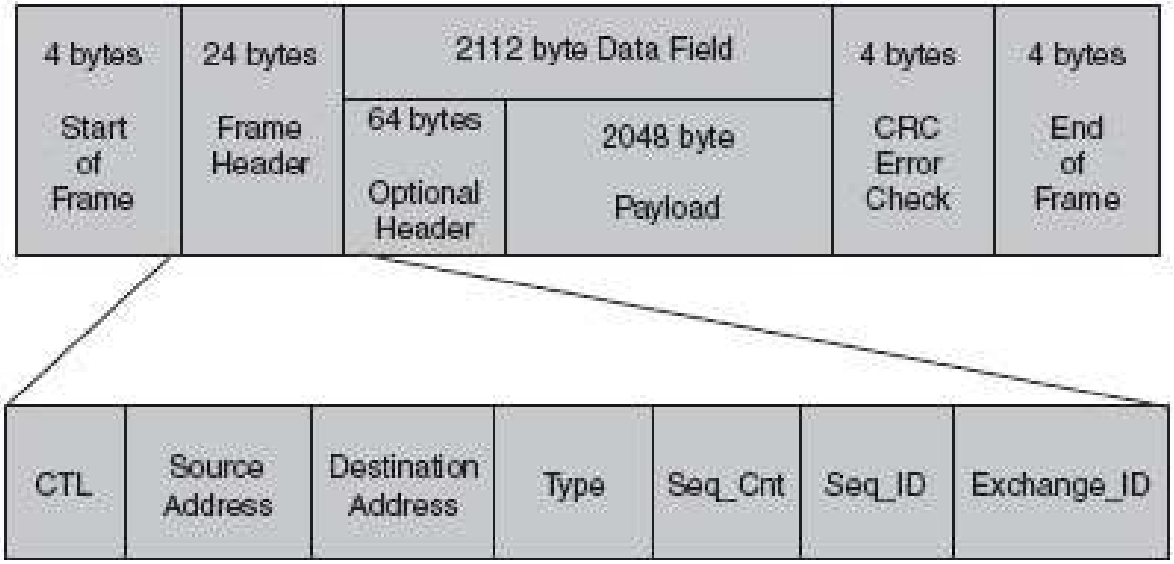
Start of Frame (4 bytes)
Frame Header (24 bytes)
Optional Headers (ESP, Network, etc – 64 bytes)
Payload (2048 bytes)
Optional "Fill bytes" or ESP Trailer (36 bytes)
CRC (4 bytes)
End of Frame (4 bytes)

The 24 byte frame header is included in the FC-2 frame

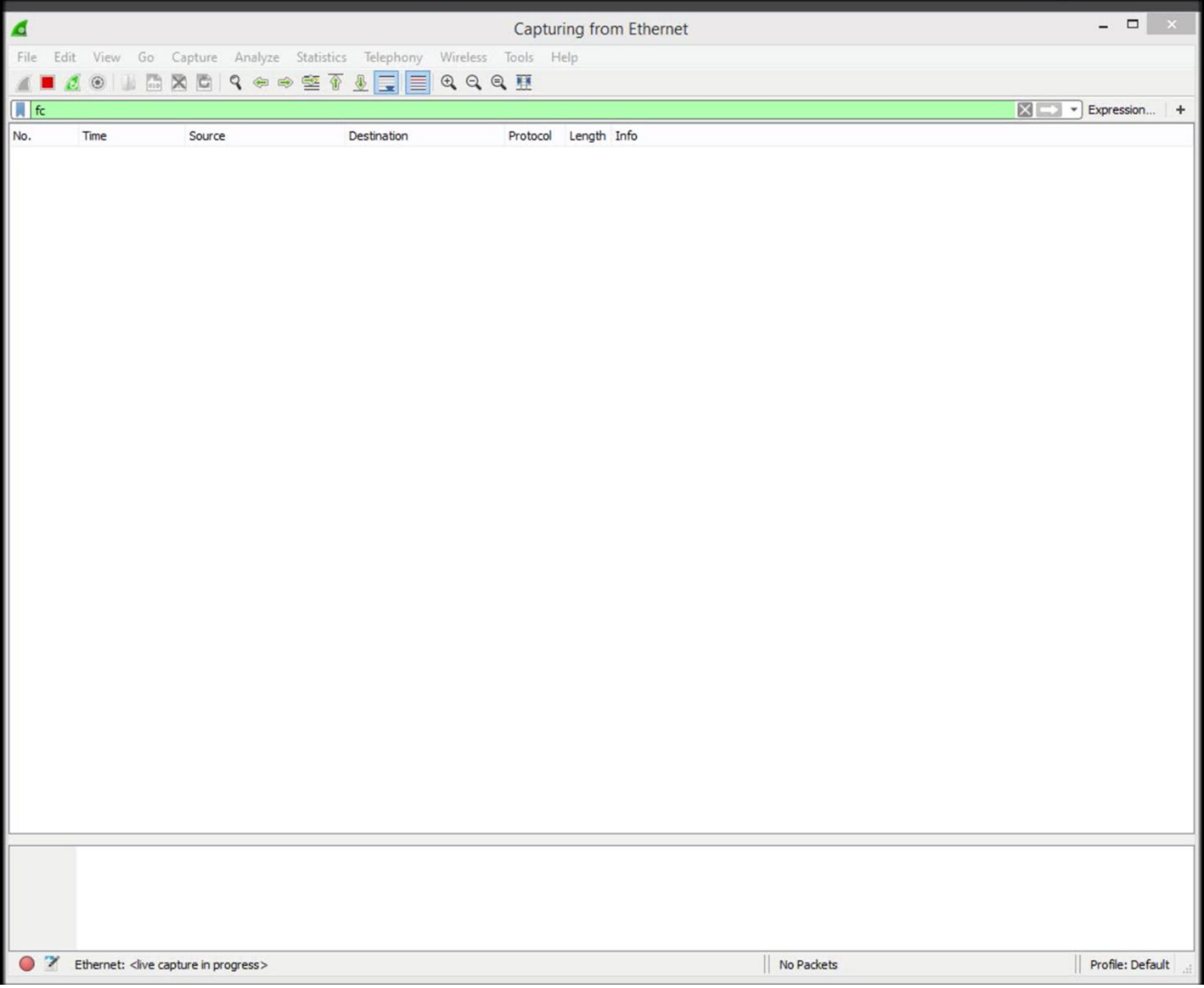
**http://www.enterprisenetworkingplanet.com/netsp/article.php/3690921/Storage-Networking-101--Understanding-the-Fibre-Channel-Protocol.htm

FC-2 Header Field

- Source & Dest address (24 bit)
- Seq_ID – identifies the session
- Seq_Cnt – identifies frames within a sequence
- Type – upper layer protocol byte section

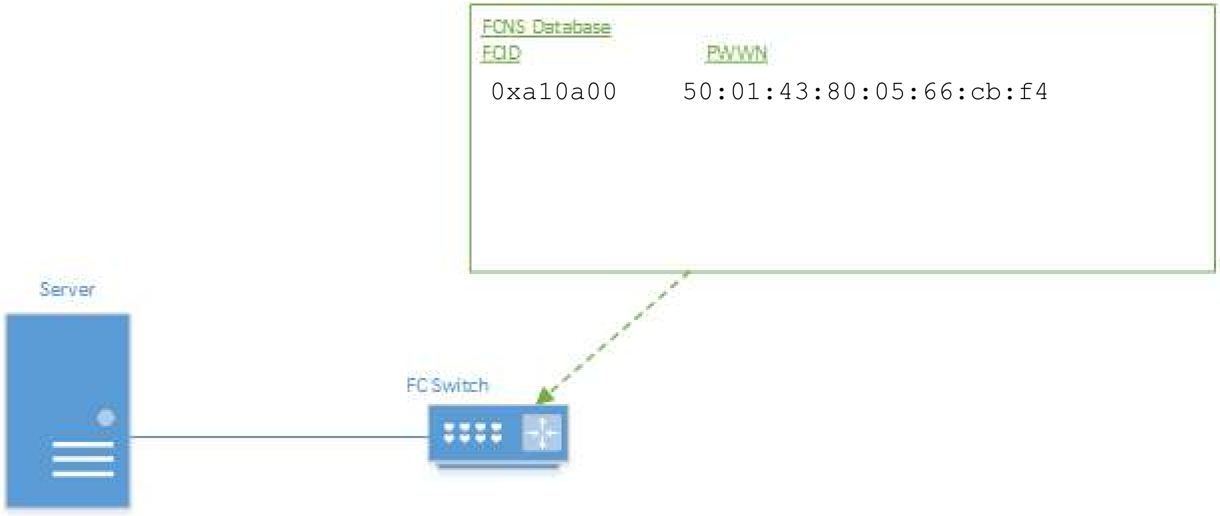


Protocol Analyzer – HBA Login



Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
fc
Expression...
+
No. Time Source Destination Protocol Length Info
Ethernet: <live capture in progress>
No Packets
Profile: Default

Breaking down NS login



FCNS Database

FCID	PWWN
0xa10a00	50:01:43:80:05:66:cb:f4

SRC_ID: 0x000000
DST_ID: 0xFFFFFE

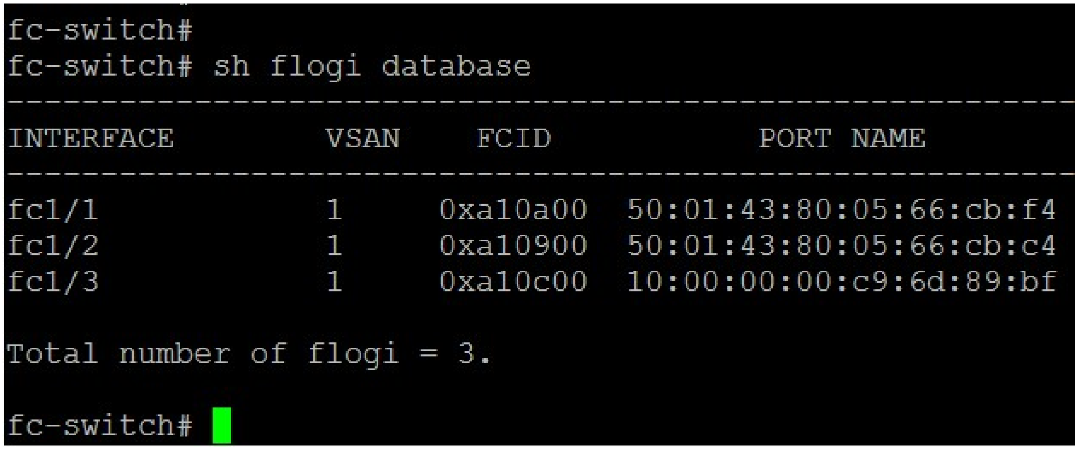
SRC_ID: 0xFFFFFE
DST_ID: 0xa10a00

SRC_ID: 0xFFFFFC
DST_ID: 0xa10a00

SRC_ID: 0xa10a00
DST_ID: 0xFFFFFC

FC Name Server Pollution

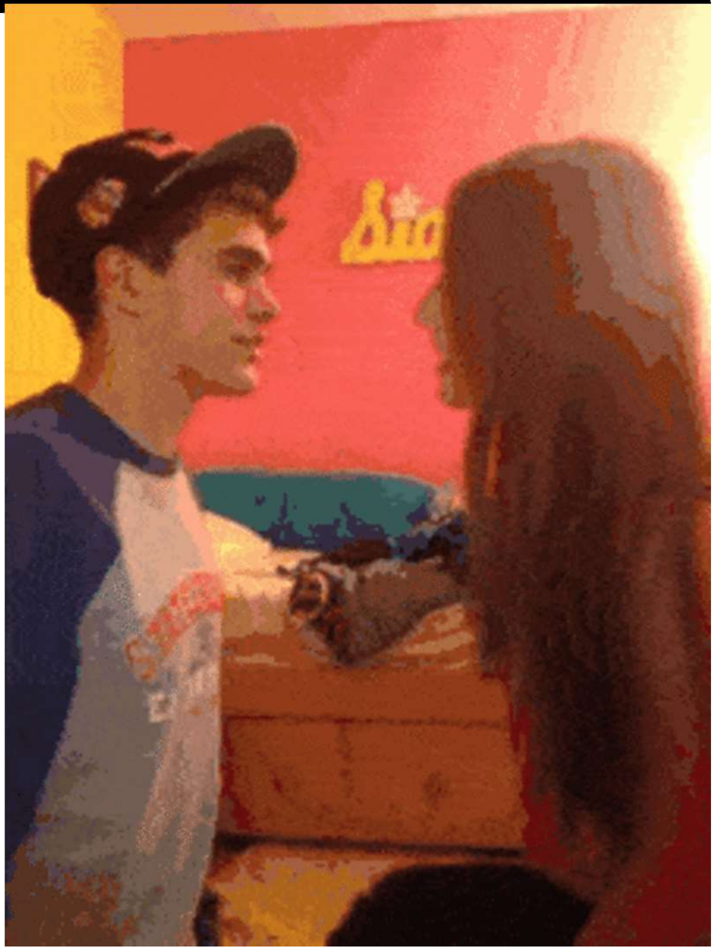
- The FC NS can be poisoned by sending a port-login to the well-known address of OxFFFFFC



```
fc-switch#
fc-switch# sh flogi database
---------------------------------------------------------------------------
INTERFACE         VSAN    FCID            PORT NAME
---------------------------------------------------------------------------
fc1/1             1     0xa10a00  50:01:43:80:05:66:cb:f4
fc1/2             1     0xa10900  50:01:43:80:05:66:cb:c4
fc1/3             1     0xa10c00  10:00:00:00:c9:6d:89:bf

Total number of flogi = 3.

fc-switch# 
```



Session Hijacking

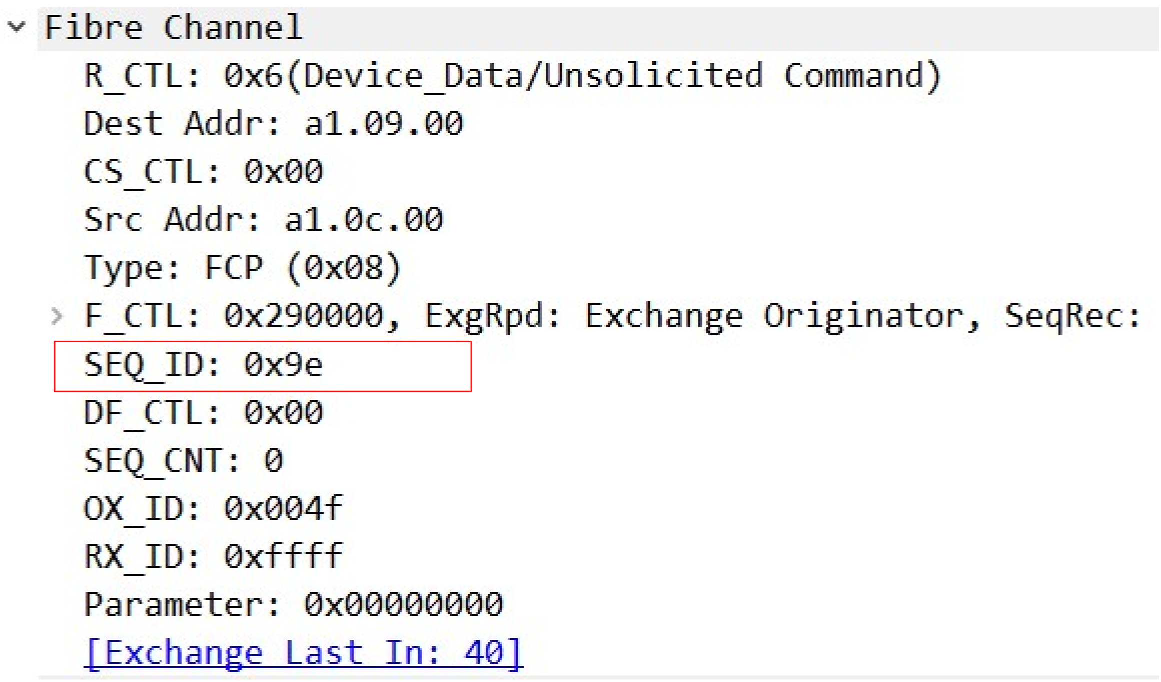
- **session hijacking** is the exploitation of a valid computer **session** to gain unauthorized access to information or services in a computer system

- Session hijacking in TCP is based on weak, predictable sequence numbers

Session Hijacking in FC

- FC sessions are controlled by the SEQ_CNT and SEQ_ID values in the frame header

- Observation of these suggest predictability in the SEQ numbers of sessions



```
Fibre Channel
  R_CTL: 0x6(Device_Data/Unsolicited Command)
  Dest Addr: a1.09.00
  CS_CTL: 0x00
  Src Addr: a1.0c.00
  Type: FCP (0x08)
> F_CTL: 0x290000, ExgRpd: Exchange Originator, SeqRec:
  SEQ_ID: 0x9e
  DF_CTL: 0x00
  SEQ_CNT: 0
  OX_ID: 0x004f
  RX_ID: 0xffff
  Parameter: 0x00000000
  [Exchange Last In: 40]
```

E-Port negotiation

- N-Port refers to a host login
- E-Port refers to a switch login – also known as an ISL link
- Full vSan and zone information transfer

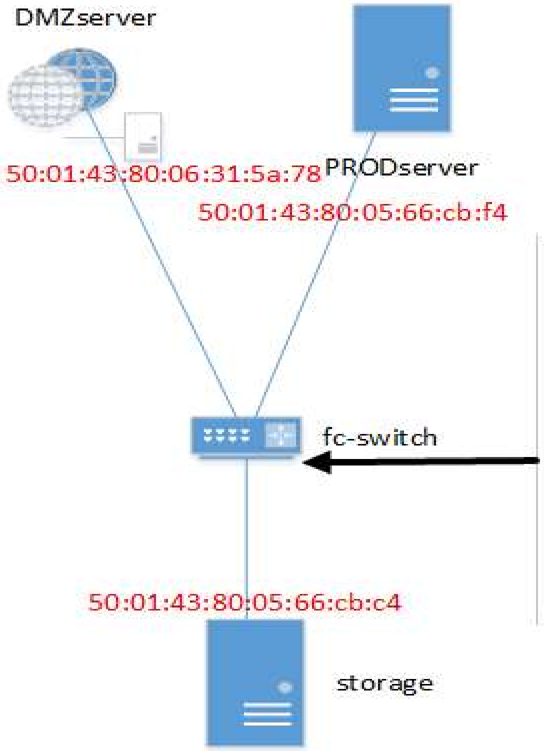
Authorisation

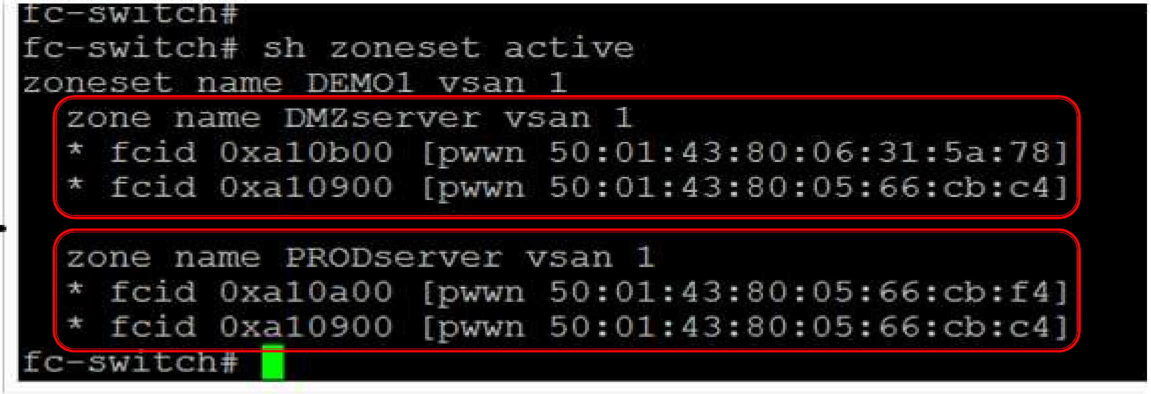
- Access authorisation from host to storage array through the switch is based on the WWN number associated to the HBA (aka FC NIC)
- This is done via zoning or lun masking

Authorisation:
FC Zoning & LUN Masking

- Used to restrict server access to storage arrays not allocated to that server
- Zones are based on grouping WWN numbers into servers that access a shared group of storage devices
- Devices outside a zone are not visible to the device inside the zone
- Similar to LUN masking, which can be implemented on endpoints.

Switch Zoning - example

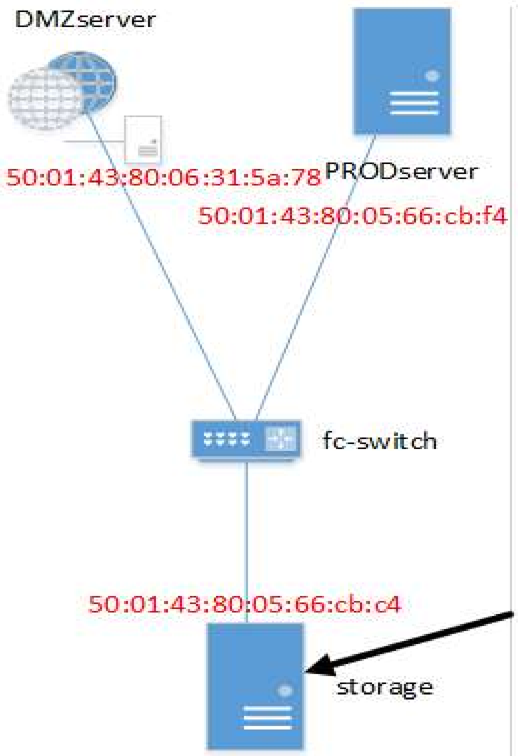


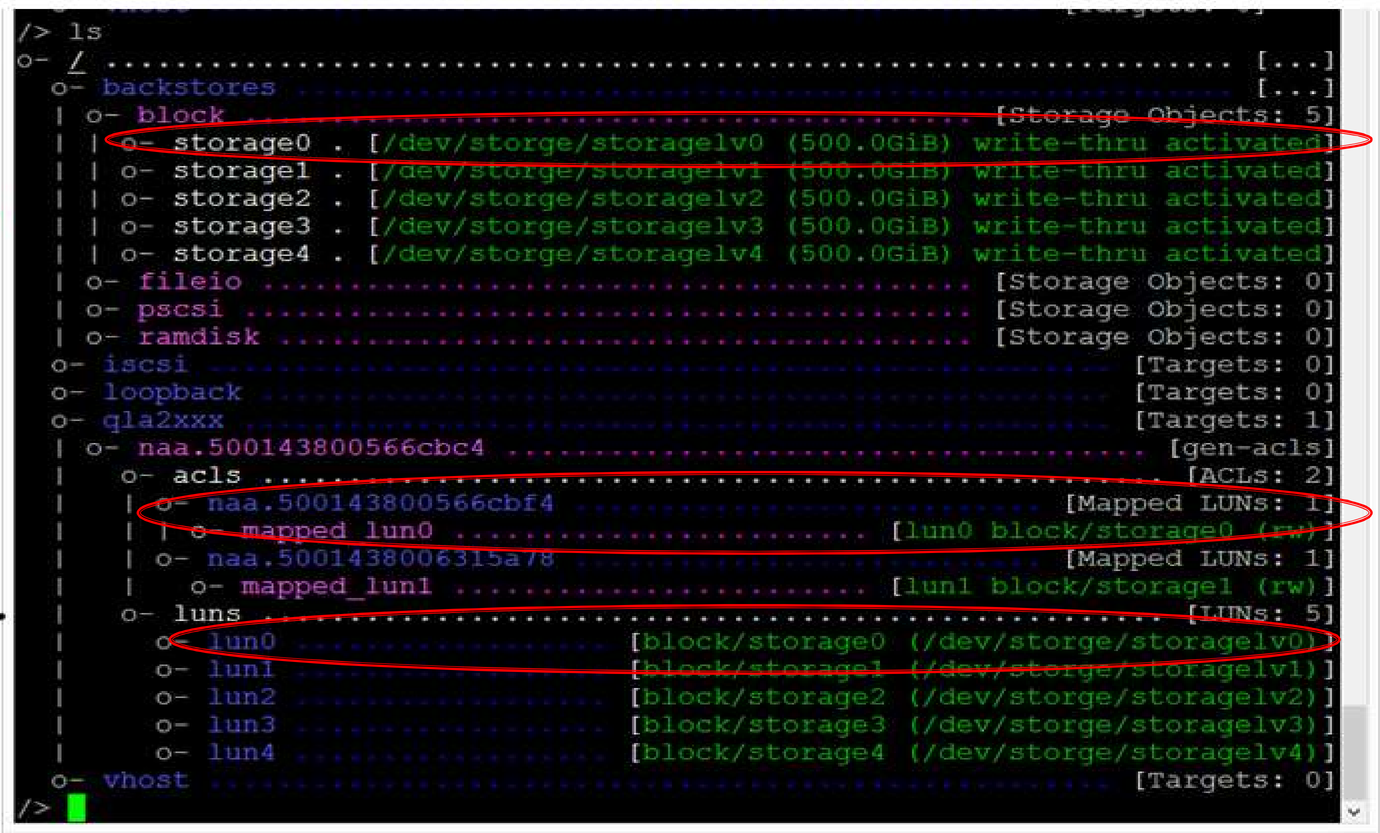


```
fc-switch#
fc-switch# sh zoneset active
zoneset name DEMO1 vsan 1
  zone name DMZserver vsan 1
  * fcid 0xa10b00 [pwwn 50:01:43:80:06:31:5a:78]
  * fcid 0xa10900 [pwwn 50:01:43:80:05:66:cb:c4]

  zone name PRODserver vsan 1
  * fcid 0xa10a00 [pwwn 50:01:43:80:05:66:cb:f4]
  * fcid 0xa10900 [pwwn 50:01:43:80:05:66:cb:c4]
fc-switch#
```

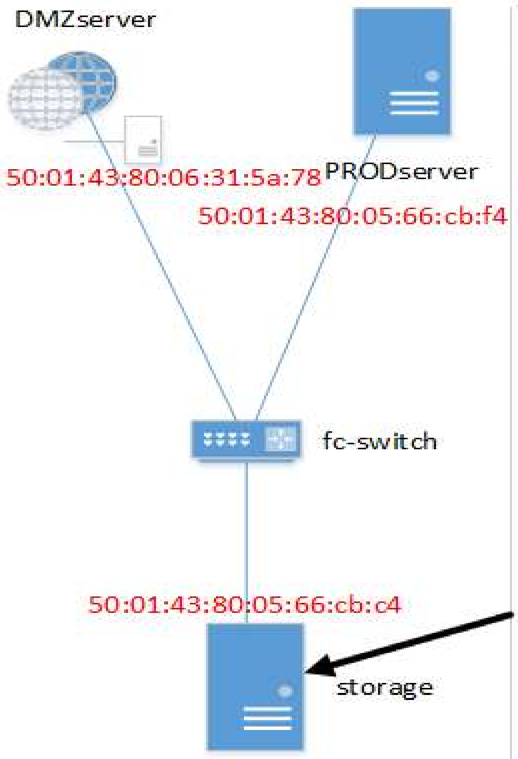
LUN Masking - example

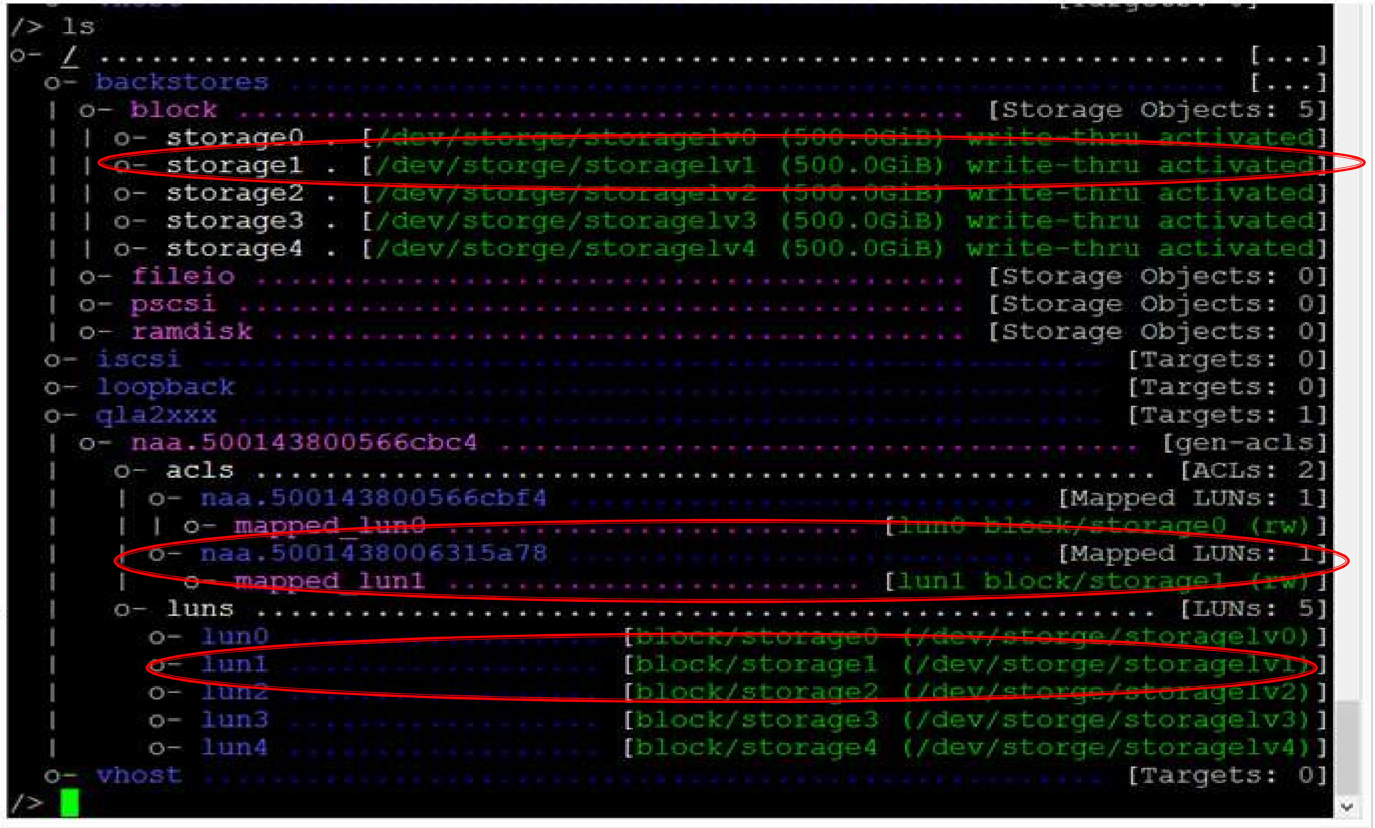




```
/> ls
o- / ......................................................................... [...]
  o- backstores .............................................................. [...]
  | o- block .................................................. [Storage Objects: 5]
  | | o- storage0 . [/dev/storge/storagelv0 (500.0GiB) write-thru activated]
  | | o- storage1 . [/dev/storge/storagelv1 (500.0GiB) write-thru activated]
  | | o- storage2 . [/dev/storge/storagelv2 (500.0GiB) write-thru activated]
  | | o- storage3 . [/dev/storge/storagelv3 (500.0GiB) write-thru activated]
  | | o- storage4 . [/dev/storge/storagelv4 (500.0GiB) write-thru activated]
  | o- fileio ................................................. [Storage Objects: 0]
  | o- pscsi .................................................. [Storage Objects: 0]
  | o- ramdisk ................................................ [Storage Objects: 0]
  o- iscsi ............................................................ [Targets: 0]
  o- loopback ......................................................... [Targets: 0]
  o- qla2xxx .......................................................... [Targets: 1]
  | o- naa.500143800566cbc4 ............................................ [gen-acls]
  |   o- acls ........................................................... [ACLs: 2]
  |   | o- naa.500143800566cbf4 ................................... [Mapped LUNs: 1]
  |   | | o- mapped_lun0 ........................... [lun0 block/storage0 (rw)]
  |   | o- naa.50014380006315a78 .................................. [Mapped LUNs: 1]
  |   |   o- mapped_lun1 ........................... [lun1 block/storage1 (rw)]
  |   o- luns ........................................................... [LUNs: 5]
  |     o- lun0 ..................... [block/storage0 (/dev/storge/storagelv0)]
  |     o- lun1 ..................... [block/storage1 (/dev/storge/storagelv1)]
  |     o- lun2 ..................... [block/storage2 (/dev/storge/storagelv2)]
  |     o- lun3 ..................... [block/storage3 (/dev/storge/storagelv3)]
  |     o- lun4 ..................... [block/storage4 (/dev/storge/storagelv4)]
  o- vhost ............................................................ [Targets: 0]
/>
```

LUN Masking -example





```
/> ls
o- / ............................................................ [...]
  o- backstores ................................................. [...]
  | o- block .................................... [Storage Objects: 5]
  | | o- storage0 . [/dev/storge/storagelv0 (500.0GiB) write-thru activated]
  | | o- storage1 . [/dev/storge/storagelv1 (500.0GiB) write-thru activated]
  | | o- storage2 . [/dev/storge/storagelv2 (500.0GiB) write-thru activated]
  | | o- storage3 . [/dev/storge/storagelv3 (500.0GiB) write-thru activated]
  | | o- storage4 . [/dev/storge/storagelv4 (500.0GiB) write-thru activated]
  | o- fileio ................................... [Storage Objects: 0]
  | o- pscsi .................................... [Storage Objects: 0]
  | o- ramdisk .................................. [Storage Objects: 0]
  o- iscsi .............................................. [Targets: 0]
  o- loopback ........................................... [Targets: 0]
  o- qla2xxx ............................................ [Targets: 1]
  | o- naa.500143800566cbc4 ............................... [gen-acls]
  |   o- acls ............................................... [ACLs: 2]
  |   | o- naa.500143800566cbf4 ....................... [Mapped LUNs: 1]
  |   | | o- mapped_lun0 ....................... [lun0 block/storage0 (rw)]
  |   | o- naa.5001438006315a78 ....................... [Mapped LUNs: 1]
  |   |   o- mapped_lun1 ....................... [lun1 block/storage1 (rw)]
  |   o- luns ............................................... [LUNs: 5]
  |     o- lun0 .................. [block/storage0 (/dev/storge/storagelv0)]
  |     o- lun1 .................. [block/storage1 (/dev/storge/storagelv1)]
  |     o- lun2 .................. [block/storage2 (/dev/storge/storagelv2)]
  |     o- lun3 .................. [block/storage3 (/dev/storge/storagelv3)]
  |     o- lun4 .................. [block/storage4 (/dev/storge/storagelv4)]
  o- vhost .............................................. [Targets: 0]
/>
```

Demo – SAN enumeration

```
[root@storage ~]# ^C
[root@storage ~]#
[root@storage ~]#
[root@storage ~]#
```

Storage Config

Demo – WWN Spoofing

Mitigations

Configuration Mitigations:

Zoning based on ports rather than WWNs
Port binding a WWN to a specific port
Lock port to specific type (eg. E-Port, F-Port)
Use FCAP, DH-CHAP and FC-SP for authentication

Design Mitigations:

Don’t share SAN resources with devices at a lower security level

Conclusion

- FC is a weak protocol by default
- FC default configuration is unsecure
- There are steps to harden it – but they are seldom done
- Security assessments should include Storage Area Networks (SANs)