InfiniBand Primer and Basic Fabric Attacks

GIAC GCIH Gold Certification

Author: Aron Warren, aronwarren@gmail.com
Advisor: Tim Proffitt

Accepted: October 15th 2012

Abstract

For almost a decade InfiniBand Architecture (IBA) has become a networking choice amongst the Top 500 High Performance Computing Clusters of the world because of the lower cost, higher throughput, and lower latency which are better than Ethernet offerings. Some of the historical Ethernet security issues, such as Media Access Control (MAC) address spoofing, have been addressed in InfiniBand from the initial InfiniBand specifications. While MAC address spoofing is not directly replicable in InfiniBand, it is reinvented in this paper in the form of Global Unique ID (GUID) spoofing.

1. Introduction

InfiniBand™ is not a word used much in the hacking community. It is much like the phrase "Apple exploits" was to "Windows exploits" about 5 years ago or so. Apple was just not as attractive to hackers because Apple's market share did not justify not justify the effort spent. As will be shown, InfiniBand (IB) is quickly becoming a force to reckon with as it overtakes proprietary high speed interconnects like Myrinet. It may also offer, in some areas where low latency is desirable or critical, a replacement for Ethernet in the datacenter. To understand its' appeal and to understand how IB can be exploited, a bit of history and a common understanding of IB terminology is necessary.

1.1. InfiniBand Trade Association

The InfiniBand Trade Association is charged with maintaining and furthering the IB specification. The association was founded in 1999 and has a steering committee comprised of such High Performance Computing (HPC) heavyweights as HP, IBM, Intel, Mellanox and QLogic ("About the IBTA," 2010). The specification, spec, is currently matured at version 1.2.1 having last been updated in January 2008 ("Specification Download," 2010). The spec was written to provide a mechanism for achieving an alternative to the deficits of Ethernet and Myrinet, both of which have dominated the HPC markets for years.

1.2. InfiniBand Market

"InfiniBand is a low-latency, high-bandwidth interconnect which requires low processing overhead and is ideal to carry multiple traffic types (clustering, communications, storage, management) over a single connection" ("About InfiniBand," 2010). It operates in a purely switched environment and its' fabric is comprised solely of point-to-point links. IB is increasingly found in a growing percentage of the Top 500 supercomputers of the world ("Top 500 Statistics," 2012) currently eating away at Ethernet's and Myrinet's shares. The more common and ever growing datacenter is also seeing growth in IB adoption (TANEJA Group, 2012, p. 2). The consumer market is currently limited primarily to hobbyist but may even begin adopting IB technology as the price becomes more affordable and the desire for higher bandwidth becomes necessary (Harris, 2008).

1.3. Reasons for IB Adoption

1.3.1. Bandwidth

One of IB's offerings currently surpassing Ethernet is bandwidth. IB's data transfer rates are composed of the multiplication of two values. One value is the link width, also called lanes, comprised of the number of pairs of transmission and receiving lines. Common values are 1x, 4x and 12x. The second value is the signaling rate of which can be the more common Single Data Rate (SDR) at 2.5 Gb/s, Double Data Rate (DDR) at 5 Gigabits per second (Gb/s), or Quad Data Rate (QDR) at 10 Gb/s. The newer Fourteen Data Rate (FDR) is specified at 14 Gb/s while Enhanced Data Rate (EDR) offers a large 26Gb/s. Thusly a 4xQDR would be able to deliver a maximum of 4 times 10 Gb/s to equal 40 Gb/s. A 12xFDR would offer a theoretical 168 Gb/s ("InfiniBand Roadmap," 2010).

1.3.2. Latency

Another attractive offering of IB is that of latency which currently sits at sub 1µ (Mellanox Technologies, 2009, p. 4). Gigabit Ethernet, on the other hand, has on the scale of around 50 µ (QLogic Corporation, 2011, p. 4) while 10 Gigabit Ethernet is "on a par with InfiniBand" according to Cisco Systems literature (2009). 10 Gigabit Myrinet is around 3 to 25 µ (Vedantham, Nidoni, & Hussain, 2007, p. 107). While not all applications need the benefits of the lowest latency, there are quite a few applications such as communication sensitive high performance computing codes, financial trading transactions and enterprise databases that require it.

1.3.3. Fabric Simplicity

While Ethernet, in its' smallest networks, is a plug and play technology, IB is technically similar in implementation ease. While most operating system vendor releases include a plethora of Ethernet drivers for various vendor product lines, IB on the other hand typically requires the installation of additional software or drivers in order for the

Host Channel Adapter (HCA) to work. Currently drivers are limited to Windows, Linux, Solaris, VMWare and a handful of the community BSD OSs.

Unlike an Ethernet network, an IB network requires the use of a Subnet Manager (SM). The SM is in charge of the IB fabric that may consist of switches, routers and hosts. The SM directs the fabric initialization in addition to any changes to the fabric such as HCA addition, link failures and routing changes. Typically the SM is software running on a server connected to the fabric but the SM may also be found running inside a switch. The default install of the SM software is usually all that is required to get a simple IB network up and running.

2. How IB Compares to Other Current Technologies

2.1. Current Bandwidth Offerings

As mentioned before, bandwidth and latency are critical to some applications. IB still appears to be at the forefront of those two requirements. While our group found 40 Gigabit Ethernet adoption is still a relative newcomer with few vendor offerings (Warren, Kahlil & Hoehl, 2012), 40 Gigabit InfiniBand has been well established for years. The race to speeds over 100 Gigabit is pretty much over at this point with InfiniBand winning out (Morgan, 2010). It still remains to be seen if IB can gain enough market share to be a serious Ethernet competitor as 100GigE gains traction.

2.2. How the Fabric Works

The IB fabric is made up of a single or multiple switches that form a subnet. Optionally several subnets can be connected via an IB router. IB end nodes, or hosts, use HCAs that come in copper or optical flavors. The HCAs have a per-port programmable 64bit Globally Unique ID (GUID), which are the equivalent of an Ethernet MAC addresses. The 64bit GUID is the EUI-64 portion added to a 64bit IB subnet prefix to form a GID (Global Identifier) (Kashyap, 2006). The GID is what allows IB traffic to cross subnets. The GUID are hardcoded into silicon and cannot be changed while the host is online. Ethernet Media Access Controls (MACs) also are also hardcoded in chips but there are software options like Network Mapper (NMAP) that let you send spoofed MAC addresses. IB does not allow the OS to perform this trick in software. Local Identifications (LIDs) are created for each HCA port by the SM and are used to send traffic within a subnet. Packets are thusly routed from a Source LID (SLID) to a Destination LID (DLID). While not required, administrators can specify a GUID-to-LID static assignment (Oracle, 2012) otherwise the SM assigns a LID when the port becomes active. The SM collects the GUIDs from each HCA along with the GUIDs from the local switches to create a distance vector table. This is how the switches route the traffic from SLID to DLID most efficiently.

A final point on subnet managers is that the SM, if running as software on a server, is only supported by the industry to run on a Linux or Windows host. Software ports have made for the FreeBSD and Solaris operating systems at this point but are only supported by the port maintainers.

3. MAC Address Spoofing and the InfiniBand Equivalent

MAC Addresses are trivially forged these days. Under Linux all that is necessary is to run the command:

```
ifconfig eth0 hw ether 03:a0:04:d3:00:11   (Perrin, 2008)
```

Even some software packages have built in the ability to forge addresses. Take NMAP for example. All that is necessary to do is pass the "-- spoof-mac" as a command line option ("Options Summary", n.d.).

Under InfiniBand the process is a bit more difficult because the MAC address equivalent is stored in firmware. The first step in reprogramming a GUID is to use *lspci* to identify where the HCA is located on the PCI bus. In our example we will be using a Mellanox HCA.

```
# lspci | grep -i mellanox
07:00.0 InfiniBand: Mellanox Technologies MT25208 [InfiniHost III Ex] (rev a0)
```

07:00.0 is the PCI address of the Mellanox HCA being used. The next step is to use the mstflint firmware tool ("Mellanox Firmware Tools", 2012), in order to determine the current GUID assignments as well as the Parameter Set Identification (PSID) number. The PSID number will assist when downloading the product's correct firmware version. The *mstflint* command issued with the device and the query option, "q", is as follows:

```
# mstflint -d 07:00.0 q
Image type:      Failsafe
FW Version:      5.3.0
I.S. Version:    1
Device ID:       25218
Chip Revision:   A0
Description:     Node             Port1            Port2            Sys image
GUIDs:           0007be000004cda0 0007be000004cda1 0007be000004cda2
0007be000400d050
Board ID:        (MT_0140000001)
VSD:
PSID:            MT_0140000001
```

Shown above the HCA has a board GUID, which have been anonymized in this paper, of 0007be000004cda0, port 1 has a GUID of 0007be000004cda1 and port 2 has a GUID of 0007be000004cda2. Once the required firmware has been downloaded from the vendor's website the next step is to reprogram the same firmware version on the card while blanking the GUIDs. The example below shows the firmware filename matching the PSID and the "b" is the option to burn the firmware.

```
# mstflint -d 07:00.0 -blank_guids -i /usr/share/ib_firmware/MT_0140000001 b

Current FW version on flash:  5.3.0
New FW version:               N/A

Read and verify Invariant Sector            - OK
Read and verify PPS/SPS on flash            - OK
Burning second FW image without signatures  - 100%
Restoring second signature                  - OK
```

The last step is to program the desired GUIDs. The GUIDs are ordered by the node GUID, port 1, port 2 and finally the system image. The system image GUID is used by management software to identify components that part are larger device or are contained in an enclosure (Mellanox, 2012).

```
# mstflint -d 07:00.0 -guids 0x0007be000004cda0 0x0007be000004cda1
0x0007be000004cda2 0x0007be000400d050 sg
```

A system reboot is necessary for the newly reprogrammed GUIDs to become activated and discovered by the fabric.

4. The Attack

4.1. Reconnaissance

MAC spoofing is used essentially to impersonate another machine. The desired outcome of such an attack can be numerous but focused in the scenario below is the desire to assume the identity of another machine on the same IB subnet in order to obtain resources that are only allocated to that host. Those resources may be filesystems or databases exported only to select hosts inside the fabric.

The first step is to perform reconnaissance in order to obtain the GUIDs of the fabric. An assumption is made that it is already knows what and how the resources are being shared to the victim machine and that the attacker desires to have access to those resources. A further assumption made is that it is not possible to obtain direct access on the victim machine, or is not desired to obtain such access. Such reconnaissance of the IB fabric, including the victim's GUIDs, can be obtained by use of the ibnetdiscover command. Ibnetdiscover performs a sweep of the IB subnet and produces a report of what is discovered, roughly similar to NMAP. Below is the output of ibnetdiscover for the switch in our scenario that contains both our attacker and victim hosts. The GUID following the "H-" is the node GUID followed by the GUID of the port connected to the switch, port 1 on both the victim and attacker's hosts.

```
vendid=0x2c9
devid=0xb924
sysimgguid=0xc7cffff00513f
switchguid=0xc7cffff00513f(c7cffff00513f)
Switch  24 "S-000c7cffff00513f"         # "MT47396 Infiniscale-III Mellanox
Technologies" base port 0 lid 2 lmc 0
[1]   "H-0007be0200269464"[1](7be0200269465)  # "host0 HCA-1" lid 1 4xDDR
[2]   "H-0007be030000b6fc"[1](7be030000b6fd)    # "host1 HCA-1" lid 9 4xDDR
[3]   "H-0007be030000b5f4"[1](7be030000b5f5)    # "host2 HCA-1" lid 8 4xDDR
[4]   "H-0007be030000b63c"[1](7be030000b63d)  # "host3 HCA-1" lid 7 4xDDR
[5]   "H-0007be030000b5f8"[1](7be030000b5f9)   # "host4 HCA-1" lid 12 4xDDR
[8]   "H-0007be0200263008"[1](7be0200263009) # "host7 HCA-1" lid 16 4xDDR
[9]   "H-0007be02002632c8"[1](7be02002632c9)  # "victim HCA-1" lid 15 4xDDR
[10] "H-0007be000004cda0"[1](7be000004cda1) # "attacker HCA-1" lid 65 4xSDR
[11]  "H-0007be0200262e58"[1](7be0200262e59)  # "host10 HCA-1" lid 13 4xDDR
[13]    "S-001c7cffff0049fa"[24]               # "MT47396 Infiniscale-III Mellanox
Technologies" lid 36 4xDDR
```

It should be pointed out that this demonstration included the attacking machine utilizing a 4xSDR while the victim is a 4xDDR connection. This is only shown to highlight the different machines since an actual attack should not contain such obvious inconsistencies.

4.2. Attack

For this attack to attempt to go unnoticed the victim machine must be taken offline. The meaning of "offline" here might be differing states due to the implementation of the host OS, the IB HCA vendor or the IB fabric manufacturer. It is not obvious that when the operating system has been halted that the IB HCA will not still appear on the IB fabric. Currently it is being seen in some production environments that the host must be completely powered off, for example power removed from the HCA, for the HCA's transmitting and receiving signals to be lost.

Following, in italics, is how the victim machine looks according to the SM's log, in this example the Linux OpenSM log, when the machine has been taken offline:

The Fabric is stable, all switches and HCAs have been programmed and assigned LIDs:

```
Jul 07 10:47:19 635690 [B68EF940] 0x02 -> SUBNET UP
```

LID 2 is the Switch that detects a link state change:

```
Jul 07 10:47:20 344923 [ADCE1940] 0x01 -> log_trap_info: Received Generic Notice type:1 num:128 (Link state change) Producer:2 (Switch) from LID:2 TID:0x0000000000000018
```

A notice that the GID ending at 32c9, the victim, has gone out of service:

```
Jul 07 10:47:20 348165 [B68EF940] 0x02 -> log_notice: Reporting Generic Notice type:3 num:65 (GID out of service) from LID:1 GID:fe80::7:be02:26:32c9
```

The GUID 32c9 and LID 15 have been removed from the fabric:

```
Jul 07 10:47:20 348186 [B68EF940] 0x02 -> drop_mgr_remove_port: Removed port with GUID:0x0007be02002632c9 LID range [15, 15] of node:victim
```

The fabric is back up again after the state change:

```
Jul 07 10:47:20 349037 [B68EF940] 0x02 -> osm_ucast_mgr_process: minhop tables configured on all switches
Jul 07 10:47:20 354125 [B68EF940] 0x02 -> SUBNET UP
```

The attacker is now free to reprogram the attacking HCA with the GUIDs of the victim. Once that is done, as shown in section 3 above, the attacker's machine must be rebooted.

Following is the OpenSM log showing the attacker's machine being rebooted:

Switch reporting link state change:

```
Jul 07 11:18:15 553841 [AC8DF940] 0x01 -> log_trap_info: Received Generic Notice type:1 num:128 (Link state change) Producer:2 (Switch) from LID:2 TID:0x000000000000001f
Jul 07 11:18:15 553882 [AC8DF940] 0x02 -> log_notice: Reporting Generic Notice type:1 num:128 (Link state change) from LID:2 GID:fe80::c:7cff:ff00:513f
```

Attacking machine's GID taken out of service:

```
Jul 07 11:18:15 556824 [B68EF940] 0x02 -> log_notice: Reporting Generic Notice type:3 num:65 (GID out of service) from LID:1 GID:fe80::7:be00:4:cda1
```

Attacker's GUID taken out of service:

```
        Jul 07 11:18:15 556840 [B68EF940] 0x02 -> drop_mgr_remove_port: Removed port with
GUID:0x0007be000004cda1 LID range [65, 65] of node:attacker HCA-1
        Jul 07 11:18:15 557735 [B68EF940] 0x02 -> osm_ucast_mgr_process: minhop tables configured
on all switches
        Jul 07 11:18:15 562789 [B68EF940] 0x02 -> SUBNET UP
```

Now the attacker's machine is powering up. The switch is sending trap information reporting the link state change:

```
        Jul 07 11:19:35 511488 [B22E8940] 0x01 -> log_trap_info: Received Generic Notice type:1
num:128 (Link state change) Producer:2 (Switch) from LID:2 TID:0x0000000000000020
```

Switch log notice about link state change:

```
        Jul 07 11:19:35 511582 [B22E8940] 0x02 -> log_notice: Reporting Generic Notice type:1
num:128 (Link state change) from LID:2 GID:fe80::7:beff:ff00:513f
```

Switch tables being reconfigured:

```
        Jul 07 11:19:35 526654 [B68EF940] 0x02 -> osm_ucast_mgr_process: minhop tables configured
on all switches
```

Attacker machine with new GUID of 2632c9 is now in use;

```
        Jul 07 11:19:35 537297 [B68EF940] 0x02 -> log_notice: Reporting Generic Notice type:3
num:64 (GID in service) from LID:1 GID:fe80::7:be02:26:32c9
```

New state information being reported and then the fabric becomes stable:

```
        Jul 07 11:19:35 537309 [B68EF940] 0x02 -> state_mgr_report_new_ports: Discovered new port
with GUID:0x0002c902002632c9 LID range [15,15] of node: atacker HCA-1
Jul 07 11:19:35 537675 [B68EF940] 0x02 -> SUBNET UP
```

4.3. Finale

Once the attacker's machine has been rebooted the Subnet Manager's log file shows that a HCA's link state changed. The HCA GUID of the victim's machine, 632c9, is now found on the attacker's machine. On a busy network this state change may go unnoticed but it may be necessary for the attacker to create some noise on the fabric in order to hide the attacking machine's link state change during reboot. This could be achieved by forcing link state changes on other hosts. For example forcing reboots on other machines leading to misdirection from the intended event. Another mechanism of distraction is to cause link errors, known as symbol errors, on a random host which would cause flooding of the SM logs.

One event that must be avoided and will certainly be noticed by administrators is if the victim machine is still online when the attacker's machine comes online with the duplicate GUID:

```
Jul 07 11:25:50 903653 [AE6E2940] 0x01 -> log_trap_info: Received Generic Notice type:1 num:128
(Link state change) Producer:2 (Switch) from LID:2 TID:0x0000000000000021
Jul 07 11:25:50 903752 [AE6E2940] 0x02 -> log_notice: Reporting Generic Notice type:1 num:128 (Link
state change) from LID:2 GID:fe80::c:7cff:ff00:513f
Jul 07 11:25:51 107562 [ABEDE940] 0x01 -> report_duplicated_guid: ERR 0D01: Found duplicated node.
Jul 07 11:25:51 107667 [ABEDE940] 0x01 -> Directed Path Dump of 2 hop path: Path = 0,1,9
Jul 07 11:25:51 107685 [ABEDE940] 0x01 -> Directed Path Dump of 2 hop path: Path = 0,1,9
Jul 07 11:25:51 107767 [ABEDE940] 0x80 -> FATAL: duplicated guids or 12x lane reversal
Jul 07 11:25:51 112198 [ABEDE940] 0x01 -> report_duplicated_guid: ERR 0D01: Found duplicated node.
Jul 07 11:25:51 112214 [ABEDE940] 0x01 -> Directed Path Dump of 2 hop path: Path = 0,1,10
Jul 07 11:25:51 112230 [ABEDE940] 0x01 -> Directed Path Dump of 2 hop path: Path = 0,1,10
```

Notice the "ERR 0D01: Found duplicated node" as well as "FATAL: duplicated guids" indicating both hosts being online simultaneously.

It is also necessary that the administrators of the fabric not be made aware, or notice, the port change of the GUIDs from the switch port of the victim being moved to the switch port of the attacker. It does not appear that the idea of switchport-to-GUID monitoring is the topic of much discussion.

4.4. Remediation

Currently the best practice for preventing this type of attack is to have in place monitoring of the SM logs in such a way to notice GUIDs changing ports. While GUIDs are harder for humans to keep track of, another possibility is to assign GUIDs to hostnames and then to map the hostname to switch port. An initial baseline of the fabric would need to be created and then periodic updates performed as actual topology changes are made. An administrator could then run a monitoring program or script to detect any unplanned changes from the baseline to the current state of the fabric.

5. Conclusion

InfiniBand GUIDs are quite similar to Ethernet MAC addresses. While MAC addresses are easily spoofed in software on the operating system, InfiniBand GUIDs are much more difficult to spoof due to the design of the IB spec which require reprogramming the firmware. By eliminating the victim's presence on the IB fabric followed by a reprogramming the attacker's HCA GUIDs and rebooting the attacker is able to assume the network identity of the victim machine at the HCA level. From there if the circumstances are right the attacker can access resources that were previously offered to the victim machine. Such an attack may further be successful if the administrators of the network do not notice the link state changes of the victim machine or the attacker's GUID changing switch port locations.

By implementing a monitoring program to baseline the IB fabric topology at initialization alongside alerts issued for link state changes and GUID movement, this type of attack can be detected and mitigated. At this time it does not appear to be of much concern in the industry about this kind of attack given the lack of publicity of this attack vector. Therefore without other forms of host authentication present, we are again susceptible to this basic attack vector.

References

About InfiniBand. (2010). Retrieved from
http://www.infinibandta.org/content/pages.php?pg=about_us_infiniband

About the IBTA. (2010). Retrieved from http://www.infinibandta.org/content/pages.php?pg=about_us_overview

Cisco Systems Inc. (2009). Using 10 Gigabit Ethernet Interconnect for Computational Fluid Dynamics in Automotive Design and Engineering. Retrieved from http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9670/white_paper_c11-554431.pdf

Feldman, M. (2012). Mellanox Roars Through Second Quarter As InfiniBand Revenue Takes Off [Web log post]. Retrieved from http://www.hpcwire.com/hpcwire/2012-07-24/mellanox_roars_through_second_quarter_as_infiniband_revenue_takes_off.html?featured=top

Harris, R. (2008, January 31). Build a 10 Gbit home network for $1100 [Web log post]. Retrieved from http://www.zdnet.com/blog/storage/build-a-10-gbit-home-network-for-1100/284

InfiniBand Roadmap. (2010). Retrieved from http://www.infinibandta.org/content/pages.php?pg=technology_overview

Kashyap, V. (2006) IP over Infiniband. Retrieved from http://www.ietf.org/rfc/rfc4392.txt

Mellanox Firmware Tools. (2012). Retrieved from http://www.mellanox.com/content/pages.php?pg=management_tools&menu_section=34

Mellanox Technologies. (2009). InfiniBand Architecture Overview. Retrieved from http://www.hpcadvisorycouncil.com/events/switzerland_workshop/pdf/Presentations/Day%201/13_Mellanox%20InfiniBand%20Training.pdf

Mellanox Technologies. (2012). 648 Port InfiniBand FDR Switch Platform Hardware Installation Guide. Retrieved from http://www.mellanox.com/related-docs/user_manuals/SX6536_Installation_Guide.pdf

Morgan, T. P. (2010). InfiniBand to outpace Ethernet's unstoppable force [Web log post]. Retrieved from http://www.theregister.co.uk/2010/06/29/infiniband_roadmap/

Options Summary. (n.d.) Retrieved from http://nmap.org/book/man-briefoptions.html

Oracle. (2012). Oracle Exalogic Machine Owner's Guide. Retrieved from http://docs.oracle.com/cd/E18476_01/doc.220/e18478/fabric.htm#CIHJHAHJ

Perrin, C. (2008). How to spoof a MAC address [Web log post]. Retrieved from http://www.techrepublic.com/blog/security/how-to-spoof-a-mac-address/395

QLogic Corporation. (2011). Introduction to Ethernet Latency: An Explanation of Latency and Latency Measurement (Report No. Sn0330915-00 rev. A 08/11). Retrieved from http://www.qlogic.com/Resources/Documents/TechnologyBriefs/Adapters/Tech_Brief_Introduction_to_Ethernet_Latency.pdf

Specification Download. (2010). Retrieved from http://www.infinibandta.org/content/pages.php?pg=technology_download

TANEJA Group. (2012). Infiniband's Data Center March. Retrieved from http://members.infinibandta.org/kwspub/Taneja_Analyst_Report_-_InfiniBands_Data_Center_-_July_2012.pdf

Top 500 Statistics. (2012). Retrieved from http://i.top500.org/overtime

Vedantham, S., Nidoni, S., & Hussain, M. (2007, November). Evaluating the Myrinet-10g Interconnect on Dell Poweredge Servers. Retrieved from http://i.dell.com/sites/content/business/solutions/power/en/Documents/ps4q07-20070554-Vendantham.pdf

Warren, A., Kahlil, G., & Hoehl, M. (2012). Implementing and Automating Critical Control 19: Secure Network Engineering for Next Generation Data Center Networks. Retrieved from http://www.sans.edu/student-files/projects/jwp-whitepaper-hoehl-khalil.doc