3-22-2019

A Framework for Cyber Vulnerability Assessments of InfiniBand Networks

Daryl W. Schmitt

A Framework for Cyber Vulnerability Assessments of InfiniBand Networks

THESIS

Daryl W. Schmitt, Capt, USAF

AFIT-ENG-MS-19-M-054

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-19-M-054

A FRAMEWORK FOR CYBER VULNERABILITY ASSESSMENTS OF INFINIBAND NETWORKS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Science

Daryl W. Schmitt, BS

Capt, USAF

March 2019

AFIT-ENG-MS-19-M-054

A FRAMEWORK FOR CYBER VULNERABILITY ASSESSMENTS OF INFINIBAND NETWORKS

THESIS

Daryl W. Schmitt, BS
Capt, USAF

Committee Membership:

Scott R. Graham, Ph.D.
Chair

Lt Col Patrick J. Sweeney, Ph.D.
Member

Robert F. Mills, Ph.D.
Member

AFIT-ENG-MS-19-M-054

Abstract

InfiniBand is a popular Input/Output interconnect technology used in High Performance Computing clusters. It is employed in over a quarter of the world's 500 fastest computer systems. Although it was created to provide extremely low network latency with a high Quality of Service, the cybersecurity aspects of InfiniBand have yet to be thoroughly investigated. The InfiniBand Architecture was designed as a data center technology, logically separated from the Internet, so defensive mechanisms such as packet encryption were not implemented. Cyber communities do not appear to have taken an interest in InfiniBand, but that is likely to change as attackers branch out from traditional computing devices. This thesis considers the security implications of InfiniBand features and constructs a framework for conducting Cyber Vulnerability Assessments. Several attack primitives are tested and analyzed. Finally, new cyber tools and security devices for InfiniBand are proposed, and changes to existing products are recommended.

Acknowledgements

I wish to thank my advisor, Dr. Scott Graham, for the enormous amount of patience he had with me. I appreciate the time he carved out of his busy schedule to meet with me regularly, even when I did not have a prearranged meeting time.

Thanks to my committee members, Lt Col Patrick Sweeney and Dr. Robert Mills, for mentoring me during this process and for editing various drafts.

Ryan Gordon deserves a special thanks for assisting me with equipment setup and software troubleshooting during his internship.

I would also like to thank Steven Dunlap for troubleshooting help and for brainstorming with me.

Additionally, I am grateful to my sponsors at the Air Force Research Laboratory for allowing me to choose this thesis topic and for providing invaluable technical guidance early on in my research.

Daryl W. Schmitt

Table of Contents

Page

List of Figures

List of Tables

List of Abbreviations

AAA	Authentication, Authorization, and Accounting
ACL	Access Control List
ACM	Association for Computing Machinery
AI	Artificial Intelligence
API	application programming interface
APT	Advanced Persistent Threat
ARP	Address Resolution Protocol
ASIC	Application Specific Integrated Circuit
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
BASH	Bourne-Again Shell
BMA	Baseboard Management Agent
C2	Command and Control
CAN	Controller Area Network
CCM	Cloud Controls Matrix
CIA	Confidentiality, Integrity, and Availability
CLI	command line interface
CPU	Central Processing Unit
CQ	Completion Queue
CRC	Cyclic Redundancy Check
CVA	Cyber Vulnerability Assessment
CVSS	Common Vulnerability Scoring System
DDoS	distributed denial of service
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized Zone

DoS	Denial of Service
DSCP	differentiated services code point
DTN	data transfer node
eIPoIB	Ethernet Tunneling over IPoIB
EoIB	Ethernet over InfiniBand
FFIEC	Federal Financial Institutions Examination Council
FTP	File Transfer Protocol
GID	Global IDentifier
GUI	graphical user interface
GUID	Global Unique IDentifier
HCA	Host Channel Adapter
HMI	Human-Machine Interface
HPC	high-performance computing
HVAC	heating, ventilation, and air conditioning
IANA	Internet Assigned Numbers Authority
IBA	InfiniBand Architecture
IBTA	InfiniBand Trade Association
ICRC	Invariant Cyclic Redundancy Check
ICS	Industrial Control System
IDS/IPS	Intrusion Detection/Prevention System
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
IPoIB	Internet Protocol over InfiniBand
IPsec	Internet Protocol Security
IT	Information Technology

iWARP Internet Wide Area RDMA Protocol

LAN Local Area Network

LDAP Lightweight Directory Access Protocol

LID Local IDentifier

MAC Media Access Control

MAD Management Datagram

MPI Message Passing Interface

MTU maximum transmission unit

NIC Network Interface Card

NIST National Institute of Standards and Technology

NTP Network Time Protocol

NVD National Vulnerability Database

NVMe Non-Volatile Memory Express

NVMHCIS Non-Volatile Memory Host Controller Interface Specification

OEM original equipment manufacturer

OFED OpenFabrics Enterprise Distribution

OS operating system

OSI Open Systems Interconnection

PDU Protocol Data Unit

PPS ports, protocols, and services

PSN Packet Sequence Number

QoS quality of service

QP Queue Pair

QPN Queue Pair Number

RADIUS Remote Authentication Dial-In User Service

RC Reliable Connection

RD	Reliable Datagram
RDMA	Remote Direct Memory Access
RoCE	RDMA over Converged Ethernet
SDN	Software Defined Networking
SFP	Small Form-factor Pluggable
SL	Service Level
SM	Subnet Manager
SMA	Subnet Management Agent
SMB	Server Message Block
SMP	Subnet Management Packet
SNMP	Simple Network Management Protocol
SPAN	Switched Port Analyzer
SQL	Structured Query Language
SR-IOV	Single Root I/O Virtualization
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UC	Unreliable Connection
UD	Unreliable Datagram
UDP	User Datagram Protocol
UFM®	Unified Fabric Manager
ULP	Upper Layer Protocol
USB	Universal Serial Bus
V2I	vehicle-to-infrastructure
V2V	vehicle-to-vehicle
VANETs	vehicular ad hoc networks

VCRC Variant Cyclic Redundancy Check

VL Virtual Lane

VLAN Virtual Local Area Network

VM virtual machine

VPI Virtual Protocol Interconnect

VPN Virtual Private Network

WQ Work Queue

WR Work Request

WRR weighted round robin

XSS Cross-Site Scripting

A FRAMEWORK FOR CYBER VULNERABILITY
ASSESSMENTS OF INFINIBAND NETWORKS

I. Introduction

The cyber threat landscape is continually evolving as attackers target new types of networks, devices, and applications. According to Symantec's 2018 Internet Security Threat Report, the number of new mobile malware variants increased by 54 percent in 2017, as compared to 2016 [1]. Much more alarming was the 600% increase in attacks against Internet of Things (IoT) devices. These numbers illuminate recent cyber trends, but primarily include less sophisticated actors targeting individuals and smaller businesses. In 1998, Post and Kagan concluded that anti-virus software was generally "effective at preventing the spread of viruses" [2], but modern experts agree that it usually only catches older and less stealthy malicious files [3].

Relatively little is known about the activities of more advanced cyber threats such as larger criminal organizations, hacktivists, and nation state supported intelligence and hacking groups. Top tier actors are using traditional malware less and less, because repeated use of the same or similar strains cannot be trusted to work if a victim or outside security researcher locates any code fragments left behind. Furthermore, the most glaring vulnerabilities in commercially available operating systems and applications have already been patched. This is why secret knowledge about a previously undiscovered vulnerability, a so-called zero day exploit, is so valuable. These can be sold on the black market for hundreds of thousands of dollars each, but more importantly they can give a well-financed buyer a weapon for which there is probably no defense [4].

Despite the lack of documentation on advanced cyber actors, it must be assumed that they are building capabilities against networks designated by the United States Department of Homeland Security as belonging to one of the 16 critical infrastructure sectors [5]. Information Technology (IT) professionals and cyber defenders have often relied on signature-based detection methods to notify them about anomalous and potentially malicious activities within their networks, but this approach cedes the initiative to the attacker and relegates the defender to a reactive position. Symantec's findings suggest the need for more proactive measures throughout the computer industry as a whole. Cybersecurity experts must explore and evaluate all of their computing equipment in novel ways, including from an outsider's perspective.

It is unreasonable to expect network engineers and programmers to have the talent, time, or funding to compete with elite computer hackers, especially those with the backing of a nation state or a large criminal organization. As a result, much of the onus falls on the research community to investigate and test the cyber hardening and resiliency of computing processing entities which were not previously thought of as such. Examples of these include mobile devices, the IoT, Industrial Control System (ICS) networks, and embedded devices communicating over vehicular networks. Non-traditional computing devices such as these do not usually have active traffic monitoring in place, much less an actual human periodically viewing logs and alerts. These types of equipment were not designed with cyber security as a primary feature. Instead, they were built for user convenience, durability (availability of services), and profitability. Despite these inherent challenges, a small amount of cyber hardening can greatly increase the cost to the attacker and reduce the overall likelihood of a successful cyber attack.

This thesis will focus on a niche computing market, specifically the portion of the high-performance computing (HPC) community using InfiniBand, which is an

Input/Output interconnect technology. As with the systems mentioned earlier, InfiniBand equipment has not been subjected to thorough, external security testing, because it has not been considered a likely target of hackers. InfiniBand designers recognized the need for hardening and resiliency, and they wanted to improve upon some of the fundamental weaknesses of Ethernet. High-speed networking hardware is very expensive, so InfiniBand customers rightfully expect that their equipment will not be easily compromised by attacks of relatively low sophistication (so called "script kiddy" level). According to the November 2018 update to the TOP500 list, InfiniBand equipment empowers 27% of the 500 most powerful computer systems in the world but accounts for 37.4% of their total performance (see Figures 1 and 2 below) [6]. So, InfiniBand manufacturers have much to lose in terms of reputation and finances should a newsworthy cyber attack on an InfiniBand network occur.

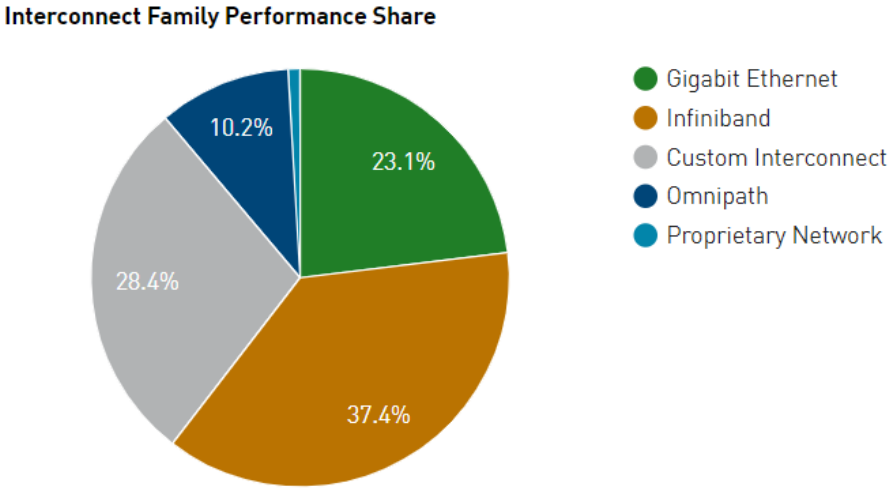
Figure 1. TOP500 Computer Systems

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)
1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,397,824	143,500.0	200,794.9
2	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0
3	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9

The desire to put such concerns to rest can be seen in a Mellanox white paper entitled "Security in Mellanox Technologies InfiniBand Fabrics" [7]. It is part of the motivation behind this thesis. The document provides a security review of InfiniBand protocols and also highlights certain Mellanox Technologies products offerings. A

company forum is necessarily biased, so it is worthwhile to provide an alternate viewpoint. This thesis seeks to explore InfiniBand security features and implications even further.

Figure 2. TOP500 List Statistics



Little published research tackles the question of how to defend HPC systems in general. This has not been a priority, because these networks have not been considered to be at significant risk due to being in isolated enclaves. No model exists for evaluating the cyber posture of an InfiniBand network. This thesis describes essential elements of a meaningful Cyber Vulnerability Assessment (CVA) of InfiniBand and how one could or should be performed. Basic attack primitives are tested and analyzed, and threats to InfiniBand systems are considered. Lastly, this thesis proposes creating certain cyber tools and devices as well as improving current ones in order to bolster network defenses.

These topics were generated as a result of the following research questions:

- What, if any, cybersecurity concerns are inherent to the InfiniBand specification?
- What security features does (the) InfiniBand (protocol, architecture, hardware,

software) have? Are they effective?

- How should one defend an InfiniBand network differently from how traditional IT networks are defended?
- How could the overall security of an InfiniBand network be evaluated and then improved upon? What skillsets and equipment would be needed to perform this evaluation?

The structure of the thesis as follows: the background is discussed in Chapter 2; Chapter 3 covers the methodology behind an InfiniBand CVA and of various attack primitives; the CVA framework is detailed in Chapter 4 along with the results of individual attacks; Chapter 5 proposes cyber tools and mitigations for the InfiniBand Architecture; and, finally, the conclusion and future work round out Chapter 6.

II. Background

This chapter will describe the InfiniBand Architecture and how its components interact with each other. Current and potential employment of InfiniBand equipment will be explored to understand what information these networks are processing and what about them might attract cyber attackers. The next section will delve into security aspects of InfiniBand, which is the focus of this thesis. This will lead into a discussion about how cybersecurity evaluations are performed.

2.1 InfiniBand

InfiniBand is a network protocol comparable to Ethernet. It is extremely lightweight, designed to minimize latency. In the late 1990s, the computer industry recognized that it was facing a tremendous hurdle. Processor speeds were continuing to increase according to Moore's Law, but memory latency and network bandwidth limitations were continually nullifying processor performance gains. For the average user, this was not much of a problem. But for high-end servers, especially those operating in clusters, a faster networking solution was needed. The networking (Gigabit Ethernet) and storage (Fibre Channel) cards in a single system were pushing bandwidth limits of motherboard buses and networking cables [8]. The InfiniBand Trade Association (IBTA) was formed to find solutions to such obstacles.

Over 180 companies assembled in August 1999 to develop the InfiniBand Architecture (IBA). IBM and Intel were the co-chairs of the IBTA, and the Steering Committee included influential companies such as Dell, Compaq, HP, Microsoft, and Sun. With differing needs from many contributors, the IBTA had to design a flexible system. Their specification had to "scale down to cost-effective small server systems as well as scaling up to large, highly robust, enterprise-class facilities" and had to allow

for "new invention and vendor differentiation" [9]. The Association was not striving to construct the most secure networks but rather to ensure the lowest latency and highest application performance [10].

Unlike InfiniBand, Ethernet and the Internet Protocol (IP) suite was not built with such a singular purpose. Its creator, the Internet Engineering Task Force (IETF), realized the need for a layered protocol stack that could support many types of traffic. As a result, hundreds of subordinate protocols now run over Transmission Control Protocol (TCP) and/or User Datagram Protocol (UDP). Many of these services have been officially assigned virtual port numbers by the Internet Assigned Numbers Authority, with ports 0 through 49151 utilized as either system or user ports [11]. This traffic diversification, in addition to dynamic packet routing, makes the IP suite perfect for the volatile World Wide Web, but these features add complexity and limit performance. InfiniBand, on the other hand, is optimized for high bandwidth and low latency. The IBA simply considers data encapsulated within InfiniBand packets as Upper Layer Protocols. InfiniBand is not a service-oriented interconnect technology, so application layer bits are completely ignored during transit. This would not be the case if deep packet inspection took place somewhere along the route, but InfiniBand does not currently implement this capability.

Specialized equipment is needed to process and empower InfiniBand. Modern InfiniBand hardware consists of individual copper or fiber cables capable of up to 200 Gb/s full bi-directional bandwidth, but the version 1.0 release was primarily based on 2.5 Gigabit copper [12]. The basic copper link has four wires, comprising a differential signaling pair for each direction. The original specification called for several data rates: 1x, 4x, or 12x copper and 1x fiber [13]. Table 1 below shows the current InfiniBand standards, highlighting two decades of growth.

While individuals may enjoy 200 Gigabit networking equipment, InfiniBand hard-

Table 1. InfiniBand Bandwidth Specifications

InfiniBand Standard	Acronym	Line Rate	Lines	Total
Quad Data Rate	QDR	10 Gb/s	4	40 Gb/s
Fourteen Data Rate	FDR	14 Gb/s	4	56 Gb/s
Enhanced Data Rate	EDR	25 Gb/s	4	100 Gb/s
High Data Rate	HDR	50 Gb/s	4	200 Gb/s

ware was not intended for the general population. It is too specialized and is cost prohibitive for individuals and smaller companies. InfiniBand was built primarily for high-performance computing (HPC) clusters, logically isolated from the open Internet. InfiniBand nodes are capable of communicating across the web, but the Internet backbone could not likely run on InfiniBand due to features such as predetermined, static routing. It is clear the IBTA was aware of this, because the "present IBA Router specification does not cover the routing protocol nor the messages exchanged between routers." True routers are optional in InfiniBand networks, as InfiniBand has been successfully utilized without routers in a production environment between two distant clusters [14]. Nonetheless, the fundamental nature of InfiniBand is that of a data center technology not typically deployed in a network's Demilitarized Zone unless firewalls or other similar access controlled layers are placed in front of the InfiniBand fabric [7].

This logical segregation provides a fair amount of security, but it is not hard to imagine that driven cyber actors would want to gain access to the valuable data stored within. However, communicating on an InfiniBand network is not a trivial task, because most Windows and Linux applications were not written to be able to run over InfiniBand, or at least to take advantage of its superior latencies. That begs the question as to what types of customers the IBTA envisioned when it designed the InfiniBand specification. Since this paper seeks to improve the defenses of InfiniBand systems, it is important to understand what types of networks employ InfiniBand

equipment. One cannot understand an attacker's motives without appreciating the value of that which is being protected.

2.1.1 Existing Deployment of InfiniBand Equipment.

The TOP500 list referenced in the Introduction gives us a decent sampling of industries, government agencies, and other organizations that are currently using InfiniBand equipment. The fastest systems on the list are predominantly government-sponsored research organizations, but there are a surprising number of energy corporations and weather services (see Table 2) as well.

Table 2. TOP500 InfiniBand Usage by Customer Categories

Category	Examples
Research/Academia	Department of Energy National Labs (US) Forschungszentrum Juelich (Germany) CEA/TGCC-GENCI (Curie supercomputer - France) SciNet/University of Toronto (Canada)
Oil & Gas	Eni S.p.A. (Italy) Total Exploration Production (France) Saudi Aramco (Saudi Arabia)
Weather	China Meteorological Administration Meteo France National Oceanic and Atmospheric Administration (US)
HPC, AI, & Cloud Computing	National Center for HPC (Taiwan) Atos (France) Inspur (China)
Social Media	Facebook (US)
Health Care	National Institutes of Health (US)
Manufacturing	NVIDIA (US)

The common thread in these networks is the need for massive data processing. Most of the research supercomputering clusters are closed systems, but that is not likely the case with the energy and weather networks, for example. The latter require real-time inputs from atmospheric sensors, satellite imagery, and more, so these cannot be air-gapped systems. As for the private companies, they probably setup

their InfiniBand network segment(s) using the Industrial Control System (ICS) model where the corporate (enterprise) network is connected to the control system network through firewalls and/or Virtual Private Network (VPN) connections [15]. Several reasons for this include the need for billing and usage monitoring, remote supervision, as well as engineering and 3rd party access.

The level of remote connectivity to a computer system is critical from a cybersecurity perspective, because it determines how an attacker might gain access to the system. Almost no computer is truly isolated, since new software and updates must get installed somehow. Nonetheless, closed networks may necessitate physical access in the form of a compact disk or flash memory device brought in by a malicious insider or unknowing victim, as evidenced by the now infamous Stuxnet worm that made the public aware of this type of attack vector. Another option is to merely connect to the Internet occasionally when updates or other periodic transmissions are needed.

The next question to consider from a defender's standpoint is what data may be processed in these InfiniBand networks that might be of interest to hackers and criminal organizations. Research and development advances and trade secrets may be valuable, so espionage is a definite threat to the confidentiality of the data. Google, Facebook, and financial institutions require ultra low latency for search queries and transactions, but the retrieved data is usually far less valuable than the algorithms that enable these systems. Another possibility is that cyber actors might want to commandeer a HPC network in order to harness its computational power, such as for digital currency mining, other cryptographic attacks, or for a complex algorithm requiring parallel computing. Use as simple attack bots seems very unlikely due to the level of effort and sophistication required to gain access to these supercomputing networks. Lastly, data integrity should be a concern, because competing companies and adversarial nations might want to tamper or corrupt data for the sake of sabotage.

2.1.2 Future Use Cases of InfiniBand.

The process of understanding stationary, terrestrial networks has been taking place for many years now, but as we approach the year 2020, mobile platforms must be considered and contemplated for all types of computing devices. The term *vehicular* here is not being used to refer merely to automobiles, but rather to include all manned and unmanned ground vehicles, naval vessels (surface and subsurface), aircraft, and spacecraft as well. Most vehicular networks today employ basic bus structures with bandwidths in the Megabit per second range. Two major examples are Controller Area Network (CAN) bus and MIL-STD-1553. These network standards are collision domains, meaning that only one node on the bus may transmit at any one time. Besides the fact that this is an inefficient communications model, there is also limited addressing available and a lack of private conversations since network traffic is seen by all devices on the bus.

The reason these bus structures still exist in large numbers (other than cost and simplicity of design) is that vehicles have not required a high bandwidth and low latency intranetwork. However, an increasing number of technological devices, each demanding more bandwidth, is being added to vehicles for the sake of safety and improved comfort. Modern automobiles are also being equipped with communications gear to connect them with vehicular ad hoc networks (VANETs), enabling vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications supported by wireless access protocols such as IEEE 802.11p [16].

It is improbable that a relatively slow moving vehicle such as an automobile would have the need for an ultra low latency, high bandwidth network inside. Alternatively, a jet aircraft or a large satellite, travelling at high speeds, might. One reason for this is that extremely accurate geolocation of signals requires precision timing. Signals travelling at the speed of light move approximately 0.98 feet per nanosecond, so clock

skew of even a few microseconds could cause significant inaccuracies. Furthermore, taking more frequent data points usually results in a smaller Circular or Spherical Error Probable, the two or three dimensional spatial region in which over 50% of the points fall. Faster moving vehicles are able to capture distinct lines of bearing much more often than their slower counterparts, but they require equipment capable of performing the calculations.

Air and space vehicles increasingly have advanced sensors capable of receiving radio and cellular communications, full motion video, thermal information, meteorological data, radar signals and returns, airborne data links, performing terrain mapping, tracking other moving objects, and more. The fusion and processing of thousands of inputs to produce a real-time, situational awareness display could necessitate a networking solution such as InfiniBand. At a macro level, aircraft worldwide could serve as a mesh collection network relaying through satellites, ground stations, and fellow aircraft. This airborne network would be far grander and require much higher bandwidth than VANETs.

A high-speed, vehicular network must be nearly autonomous, as configuration changes or updates during operation are difficult. An exception might be satellites, because they cannot be grounded for maintenance after launch and are easier to reach via a directional antenna. The problem with granting remote access to a network engineer or programmer is that this would also open the possibility for unintended access. Wireless environments are prone to periods of poor reception as well, whether it be due to a high noise floor, signal blockage, or jamming. Nonetheless, the primary threats to a mobile, HPC network would be insider personnel and supply chain tampering. These topics will be discussed later.

The reasons an attacker might attempt to gain access to a high-speed, vehicular network are similar to those listed in the previous section. Some differences are that

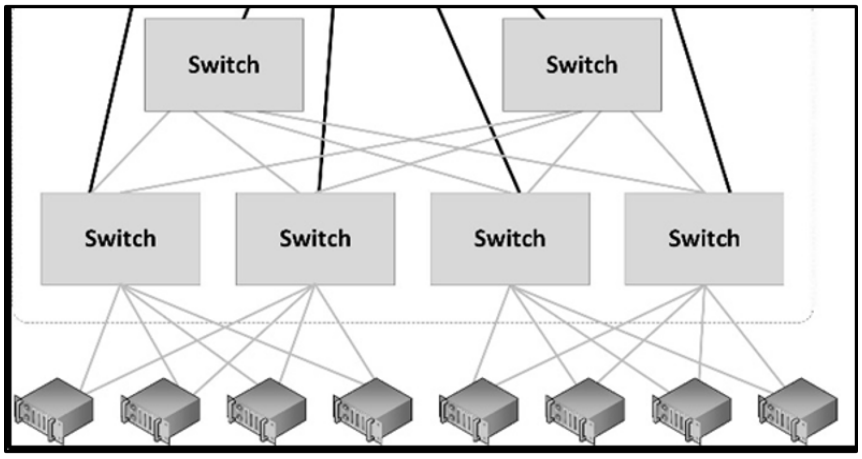
vehicular networks are relatively small compared to data warehouses and enterprise level systems, they do not have 24/7 availability, and an attacker would not likely know the vehicle's exact schedule, route, or orbit. The potential to hijack the vehicle while in transit is worth mentioning, but the navigational controls should be logically, if not physically, separated from the mission systems. These vehicles may contain a treasure trove of data, but it is probably very perishable information. Governmental systems are the most likely targets, because the opportunity to obtain classified government materials provides a large incentive for opposing military leaders.

2.1.3 InfiniBand Fundamentals.

At its core, InfiniBand is a computer networking protocol, but the term InfiniBand has become overloaded. The InfiniBand Trade Association had to design a network infrastructure around it in order to maximize the performance of InfiniBand systems. The group decided to improve upon the typical hierarchical structure of switches and routers used by Ethernet-based networks. InfiniBand employs a switched fabric topology (see Figure 3), which is a partial mesh providing connection reliability for Inter-Processor Communication systems by allowing multiple paths between systems. Scalability is supported with fully hot swappable connections managed by a single unit - the Subnet Manager [8]. InfiniBand hosts have network cards, equivalent to Ethernet Network Interface Cards, called Host Channel Adapters (HCAs). HCAs usually have at least two physical ports so that the node can be connected to two or more InfiniBand switches simultaneously. It would be impractical to create a full mesh by establishing direct links between all devices, but switched fabrics provide a nice compromise by improving redundancy, load balancing, and routing speeds.

Ethernet networks use dynamic Address Resolution Protocol (ARP) and routing tables to determine how and where to send traffic based on link speeds and congestion.

Figure 3. Switched Fabric Topology



InfiniBand switches do not make routing decisions themselves. Instead, all shortest paths are calculated by the Subnet Manager (SM) during network initialization and after configuration changes. The SM then pushes out forwarding tables to every device in the subnet, including all compute nodes. There can be multiple Subnet Managers running simultaneously but only one acting as the master. Each Channel Adapter and switch has a Subnet Management Agent (SMA) that communicates with the SM. The SM sets up and maintains every link within a subnet, performing network discovery periodically.

As for the InfiniBand protocol, it is based on the 7-layer Open Systems Interconnection (OSI) model, just as the Internet Protocol stack is [17]. Layer 2 addressing is done via a Local IDentifier (LID) that is dynamically assigned by the SM [14]. The LID is a 16-bit value, so a single subnet can support up to 65k hosts. Media Access Control (MAC) addresses used in Ethernet, by contrast, are burned into the network card by the manufacturer. However, these can be easily changed in software. A simple command like **ifconfig eth0 hw ether 02:01:02:03:04:08** will accomplish this task on many Linux distributions. This is significant, because InfiniBand addresses cannot be easily modified in such a manner. At Layer 3, InfiniBand locators are called Global IDentifiers (GIDs). These are valid IPv6 addresses for the most part.

The first half (that is, 64 of the 128 bits) of each GID is called the Global Unique IDentifier (GUID). GUIDs are embedded in the HCA, although there is not just one of them per network card as with MAC addresses; each HCA port has its own GUID. Distinct port addressing helps enforce the static routing discussed previously.

InfiniBand does not use sockets or virtual ports like Ethernet networks do. Instead, InfiniBand connections are established between two endpoints via entities called Queue Pairs (QPs). Each QP consists of a Send Queue and a Receive Queue, and each QP represents one end of a channel. If an application requires more than one connection, more QPs are created. A Send Queue and Receive Queue together are referred to as a Work Queue (WQ). Work Queues may put the results of completed Work Requests in an associated Completion Queue. This includes both successfully completed requests and unsuccessfully completed requests. Completion Queues notify applications about information regarding ended Work Requests (status, opcode, size, source) [18]. The user may insert a completion notification routine to be invoked when a new entry is added to a Completion Queue. This nomenclature lends itself to viewing InfiniBand as a messenging service.

Keeping applications informed of network activity is especially vital considering that InfiniBand has a major speed enhancing feature called Remote Direct Memory Access (RDMA). RDMA permits transferring of data without interrupting either processor. In order to avoid involving the Operating System, the applications at each end of the channel must have instant access to QPs. This is accomplished by mapping the QPs directly into each application's virtual address space. Thus, the application at each end of the connection has direct, virtual access to the channel connecting it to the application (or storage) at the other end of the channel. This is the notion of Channel Input/Output [19]. The fact that there is no extra copying of data (such as to various levels of cache) is referred to as "zero copy" networking.

Table 3. InfiniBand Transport Services

Class of Service	Acronym	State	Responses Sent?
Reliable Connection	RC	Connection oriented	Acknowledged
Reliable Datagram	RD	Multiplexed	Acknowledged
Unreliable Connection	UC	Connection oriented	Unacknowledged
Unreliable Datagram	UD	Connectionless	Unacknowledged
Raw Datagram		Connectionless	Unacknowledged

InfiniBand offers stateful and stateless connection types similar to TCP and UDP, but these are not as critical in InfiniBand. They are called Reliable Connection (RC) and Unreliable Datagram (UD). The Unreliable Connection (UC), Reliable Datagram (RD), Raw IPv6 Datagram, and Raw Ethertype Datagram transport mediums exist as well, although these are not as mainstream. Table 3 summarizes the transport options. In Ethernet networks, most higher level protocols run over TCP to guarantee 100% packet delivery. Only trivial traffic that can be easily resent (e.g. domain name queries) or that requires high speeds (i.e. streaming video) runs over UDP. Conversely, UD is extremely common in InfiniBand. InfiniBand is more efficient at avoiding congestion due to its Priority-based Flow Control. The Ethernet Pause frame "stops all traffic indiscriminately" whereas InfiniBand "strictly avoids packet loss by employing link-by-link flow control, which prevents a data packet from being sent from one end of a link if there is insufficient space to receive the packet at the other end of that link" [14]. More importantly, the InfiniBand receiver HCA drops all out-of-order packets as that is an error condition as far as the InfiniBand receiver is concerned. (This is a setting that can be changed, but the default is to disallow out-of-order delivery.)

The 'reliable' connection types (RC and RD) send acknowledgements after every transmission. Possible responses are either a positive Acknowledge (Ack) or a Negative Acknowledge (Nak). The latter message can be triggered under three con-

ditions: a temporary Receiver Not Ready (RNR Nak), a Packet Sequence Number (PSN) Sequence Error Nak, or a fatal Nak error code. The RC, UC, and RD classes support RDMA and require unique Queue Pair Numbers (QPNs), meaning that no two connections can share the same QPN simultaneously. Since there are no virtual ports in InfiniBand, this is the primary way that connections can be distinguished from each other. Conversely, UD QPs use the same QPN, because they can send to and receive messages from any other UD QP using either unicast (one to one) or multicast (one to many). In addition to RDMA reads and writes, atomic extensions to the RDMA operations also exist. These are essentially a combined write and read RDMA, carrying the data involved as immediate data [9]. See Table 4 for a detailed listing of available QP operations by connection type.

Table 4. Queue Pair Operations

Operation	UD	UC	RD	RC
Send (with immediate)	X	X	X	X
Receive	X	X	X	X
RDMA Write (with immediate)		X	X	X
RDMA Read			X	X
Atomic: Fetch and Add			X	X
Atomic: Compare and Swap			X	X
Maximum message size	MTU	1GB	1GB	1GB

RDMA has become popular due to InfiniBand, but it has been used in other I/O interconnects. The reason for its success is that RDMA enables high-throughput, low-latency networking with low CPU utilization. These advantages make it especially useful in massively parallel compute clusters. RDMA over Converged Ethernet (RoCE) and the Internet Wide Area RDMA Protocol (iWARP) now bring a similar capability to networks employing Ethernet-based software. The main difference between the two is that iWARP uses a "complex mix of layers, including DDP (Direct Data Placement), a tweak known as MPA (Marker PDU Aligned framing), and a

separate RDMA protocol to deliver RDMA services over TCP/IP," whereas RoCE operates "over standard Layer 2 and Layer 3 Ethernet switches" [20]. RoCE's superior performance metrics compared to iWARP have made it the market front runner.

Likewise, InfiniBand products (by Mellanox, at least) support Ethernet applications by offering Internet Protocol over InfiniBand (IPoIB) and Ethernet over InfiniBand (EoIB) services. IPoIB uses an Upper Layer Protocol (that is, application layer) driver that allows it to encapsulate IP datagrams over an InfiniBand Connected or Datagram transport service. EoIB is akin to IPoIB except that it includes the (Layer 2) Ethernet header and only runs over UD. EoIB performs an address translation from Ethernet layer 2 MAC addresses (48 bits long) to InfiniBand Layer 2 addresses made of LID/GID and QPN, whereas IPoIB exposes a 20 byte hardware address to the OS [21]. Because of this, EoIB requires additional equipment, specifically a BridgeX gateway that can connect an InfiniBand fabric to its 'external' side, an Ethernet network segment. This may be why EoIB is apparently being phased out, as evidenced by the fact that it is not even mentioned in the latest version of the Mellanox OpenFabrics Enterprise Distribution (OFED) Linux User's Manual (to be explained later). The Ethernet Tunneling over IPoIB (eIPoIB) driver seems to have replaced this functionality.

2.2 InfiniBand Security Features

The InfiniBand Architecture provides isolation and protection services using keys. Keys are "values assigned by an administrative entity that are used in messages in order to authenticate that the initiator of a request is an authorized requester and that the initiator has the appropriate privileges for the request being made" [22]. There are five types of InfiniBand keys: Management Keys (M_Key), Baseboard Management Keys (B_Key), Partition Keys (P_Key), Queue Keys (Q_Key), and Memory Keys

(L_Key and R_Key). A Partition Key designates a network partition to which a Channel Adapter port belongs. Each port is assigned at least one P_Key by the Subnet Manager, and these values point to entries within the port's partition key table. InfiniBand partitions are equivalent to Ethernet Virtual Local Area Networks (VLANs), so P_Keys are like VLAN tags. Memory keys are needed for RDMA operations and come in the form of local keys, L_Keys, and remote keys, R_Keys. System memory is registered to bestow access to local and remote CAs. Registration returns a pair of keys, which each have associated access permissions (i.e. read-only versus read-write) [23]. The same memory buffer can be registered several times, even with different permissions, and every registration results in a different set of keys.

Queue Keys are preshared keys that are used in the datagram connection types (RD and UD). Prior to establishing a connection, CAs exchange Q_Keys between QPs. Receipt of a packet with a different Q_Key than the one provided to the remote QP indicates that the packet is not valid, hence it is rejected. Management and Baseboard Management keys enforce control of the master Subnet Manager and of the subnet Baseboard Manager, respectively. The Baseboard Manager component communicates with nodes to provide an in-band mechanism for managing chassis [22]. The Baseboard Manager's purview covers topics such as retrieval of vital product data (e.g. serial number, manufacturing information), environmental data, and adjusting power and cooling resources [24]. The Baseboard Manager communicates with a Baseboard Management Agent (BMA) on each node, just as the SM does so with every SMA. Every CA port and each switch have an M_Key and a B_Key. These values do not need to be identical across all devices, but they must match what the destination is expecting in order to verify that the source of a management packet is the correct one.

InfiniBand also provides integrity and quality of service (QoS). Integrity is ensured

by two checksums called Cyclic Redundancy Checks (CRCs). As the name implies, the 16-bit Variant Cyclic Redundancy Check (VCRC) is recalculated at each hop. The 32-bit Invariant Cyclic Redundancy Check (ICRC) complements the VCRC by protecting the fields that do not change along the communications pathway. Each packet has a VCRC and an ICRC, and per block CRCs exist as well for each memory block sent in the payload. As for QoS, packets are assigned a priority value between 0 to 15, called their Service Level (SL). (This is equivalent to the differentiated services code point (DSCP) field in Ethernet.) Its SL factors into which Virtual Lane (VL) a packet will transit through. Each physical link can support up to 16 VLs, with VL 15 solely for management packets. Service Levels stay constant throughout a subnet, but a packet can change Virtual Lanes at each hop depending on SL to VL mapping tables [25].

InfiniBand management is performed in-band, using Management Datagrams (MADs), which are UDs with a maximum transmission unit (MTU) as low as 256 bytes. Some MADs are called Subnet Management Packets (SMPs); these are unique in several ways. In addition to transiting in VL 15, they are always sent and received on Queue Pair 0 of each port and can use directed routing [9]. Directed routing occurs when a Subnet Management Packet tells a switch which ports to send it on. This is necessary when forwarding tables have not yet been initialized.

InfiniBand software derives from the OpenFabrics Enterprise Distribution (OFED) suite from the OpenFabrics Alliance, a collaborative development by major high performance I/O vendors. Mellanox augmented this package in order to create its own version of OFED. It supports both InfiniBand and Ethernet (technically, RoCE), although many network cards cannot process both interconnect types [26]. OFED includes custom diagnostic tools for network users and maintainers to ascertain the status of the fabric. Two such utilities are ibstat and ibdump, which are analogous

to the traditional Linux ifconfig and tcpdump, respectively. These OFED tools will be covered individually in a later chapter. OFED also includes open-source software called OpenSM, which brings Subnet Manager functionality.

As for writing InfiniBand-supported applications, that is done via a series of functions called 'verbs.' The InfiniBand Architecture "contains no APIs, defined registers, etc. Instead it is specified as a collection of *verbs* – abstract representations of the function that must be present, but may be implemented with any combination and organization of hardware, firmware, and software" [9]. For instance, the IBA standard does not specify how a queue should be implemented internally in the HCA hardware. Each manufacturer must provide a driver as part of the OFA Verbs API whose input will be function calls and data structures that are defined in great detail by the API. Due to latency requirements, Mellanox programming is accomplished in the C language according to their "RDMA Aware Networks Programming User Manual." A few example verbs/functions are **ibv_get_device_list()**, **ibv_reg_mr()** to register a memory region, and **ibv_create_qp()** to create a queue pair. Table 5 contrasts some features of Ethernet and InfiniBand.

Table 5. Ethernet vs InfiniBand

Feature	Ethernet	InfiniBand
Host adapter	Network Interface Card (NIC)	Host Channel Adapter (HCA)
Programming model	Sockets	Verbs
Layer 2 addressing	Media Access Control (MAC) address statically assigned by NIC manufacturer	Local IDentifier (LID) dynamically assigned by Subnet Manager (SM)
Layer 3 addressing	Internet Protocol (IP) address	Global IDentifier (GID): 64-bit subnet ID assigned by SM plus 64-bit Global Unique IDentifier (GUID) assigned by HCA manufacturer
Forwarding tables	Distributed control; each switch discovers neighbors independently	Centralized control via SM
Packet capture	Use standard operating system capture tools, such as Wireshark or tcpdump	Use a vendor-specific tool such as ibdump from Mellanox

A less InfiniBand-specific way to write network applications is through the Message Passing Interface (MPI). MPI is a library specification that enables the development of parallel software to utilize parallel computers, clusters, and heterogeneous networks [21]. Mellanox's version of OFED includes Open MPI, which is an open source implementation by The Open MPI Project. Open MPI supports numerous interconnect types including InfiniBand/RoCE/iWARP verbs, TCP sockets, Shared memory, and more. As a result, it provides HPC programmers and users an easy way to migrate software between varied networks. Programming can be done in C or Fortran, although native Microsoft Windows support has been removed; use of Cygwin is required [27].

2.3 Standard Information Technology Protections

The process of hardening and defending a new type of system is often challenging, so it is wise to consider current protective measures for similar systems. Computer network defense of IP-based networks has been evolving for nearly three decades, so it is an established discipline. Before attempting to come up with innovative solutions, this thesis seeks to evaluate whether traditional Information Technology (IT) security tenets and devices can be applied to InfiniBand networks.

The most fundamental of IT security principles is the Confidentiality, Integrity, and Availability (CIA) triad. Confidentiality revolves around the concept of 'least privilege,' which asserts that access to information and assets should only be granted on a need to know basis so that information that is only available to some should not be accessible by everyone. This is seemingly at odds with availability, but it is not. Availability calls for systems and services to be running and operating properly when needed. This implies that disasters should degrade system performance and maintenance actions should interfere with uptime to the least extent possible. Networks

should be designed to be fault tolerant and resilient. This can be achieved through redundant systems such as software backups, power generators, hot/warm/cold sites, etc. Lastly, integrity ensures that information in transit or at rest (i.e. in memory) is not altered without permission of an owner or authorized user. An example of integrity checking is a one way cryptographic hash of a file or data block that can be calculated and then compared against at a later time [28].

Extending the notion of access control are the related principles of Authentication, Authorization, and Accounting (AAA) plus non-repudiation. Authentication is the act of confirming one's identity, ascertaining whether or not someone is a legitimate user. Authorization determines if an authenticated user has the necessary permissions or privileges to perform a certain action. Accounting (also known as Accountability) is associated with auditing and logging. It gives administrators the ability to track the activities that users performed. Completing the circle is non-repudiation, which proves that an event or action has taken place so that it can't be repudiated (that is, denied) later [29]. Digital signatures and timestamps are two pieces of information that can be used as evidence, should the need arise.

These security principles are put into effect by cyber defense software and devices. At the host level, the most common entity is anti-virus software. Anti-virus software checks the local file system for malware signatures obtained by anti-virus companies and security researchers. This is an effective way to share the findings of cybersecurity experts worldwide, but the newest, most advanced malware strains usually avoid detection. Part of the reason for this is that signature updates only occur periodically (e.g. weekly), and APT-level actors tend not to release their best material for risk of it getting caught. In addition, polymorphic code exists, mutating the source code of a program while keeping the original algorithm intact. The desired effect is to prevent exact signature matches once the code is recompiled.

Other host level security measures include file verification (attestation) and account management. File verification compares the hashes of critical files against known good versions. This ensures data integrity and can work in tandem with file backups. An offshoot of checking known good files is to look for newly created files that were not in the last system image. This can reveal unwanted material, although operating systems and applications do generate legitimate temporary files, so false positives could be numerous. Account management is the constant process of updating basic user and administrator accounts so that users have the correct privileges established by policy and/or management. At the host level, this has to do with a user's ability to read, write, and execute individual files and folders. At the network level, this entails allowing the user (or service) to interact with other devices.

Network security is often more difficult than that at the host level, especially in an enterprise-scale network with widely heterogeneous data and/or with a large number of approved ports, protocols, and services (PPS). The strictest way to limit network traffic is with firewalls and Access Control Lists (ACLs). Application layer filtering is possible with more modern firewalls, but traditional firewalls and ACL rules block incoming or outgoing traffic on specific burbs/interfaces (physical ports) of a firewall, switch, or router. Traffic is typically blocked or allowed based on the destination or source address, port, protocol, or service. Similarly, proxy servers or proxies act as application-level gateways between a local network and a larger-scale network such as the Internet. They usually filter limited types of traffic, such as web pages or e-mail content [30].

The network equivalent to anti-virus software is an Intrusion Detection/Prevention System (IDS/IPS). These devices look for signatures in network traffic. The difference between the two is that an IDS will only send alerts based on signature matches whereas an IPS will drop or reject individual packets or conversations. Anti-intrusion

systems also tend not to be super effective, because hackers can use compression and packing techniques to obfuscate their code over the wire. Network access control is handled by a AAA server running an authentication service such as RADIUS or TACACS+ [31]. Microsoft Windows Server utilizes Active Directory which runs LDAP and Kerberos, which are newer, more secure services [32].

A strong network defense combines multiple of the elements discussed in this subchapter. Just as castles of old did not rely solely on moats or castle walls to repel invaders, computer networks should not depend on only one defense mechanism. Network architects used to believe that a firewall alone could prevent all unwanted traffic from entering, but more recently the accepted train of thought is that talented attackers can find a way in somehow and that defenders must utilize layered defenses in order to make an adversary's job as difficult as possible. This strategy is referred to as Defense in Depth. The goal is that even if a hacker or piece of malware were to gain access, that entity would not be able to gain administrative privileges or be able to move around the internal network unchecked.

2.4 Previous InfiniBand Security Research

Warren from the SANS Institute presents a GUID spoofing attack, accomplished by altering the values in firmware [33]. This is a significant contribution, but with limited realism because the 'victim' machine, to which the actual GUID belonged, was taken offline prior to the attack. An attacker who compromises a single host would not be able to shut down another host without first gaining access to it, thus negating the benefit of spoofing its address. (The exception might be launching a successful denial of service attack.) Nonetheless, this precaution led to a more straightforward proof of concept experiment. In Ethernet networks, duplicate MAC addresses within the same subnet can cause network instability. It is unclear how the Subnet Manager

would react to a GUID change on a live network, since GUIDs are not supposed to change, unlike LIDs or even GIDs.

In a white paper titled "Security in Mellanox Technologies InfiniBand Fabrics", Mellanox asserted that the InfiniBand Architecture "targets one of the main concerns in such environments [a data center LAN] which is Security, and has many built in mandatory features that enable much better isolation and security than current networks and other cluster interconnects" [7]. The paper emphasizes that InfiniBand is a Layer 2 protocol much like Ethernet, so almost all Layer 3-7 security mechanisms will work the same way with InfiniBand. Switch administration is done out-of-band via management ports as opposed to many switches which can be remotely configured. Their switches support Remote Authentication Dial-In User Service (RADIUS) authentication, although it uses MD5 hashes which are now considered to be insecure. Mellanox contends that hardware-based features such as packet construction and GUID addressing significantly improve security by preventing "software applications to gain control over them and maliciously change those attributes." The paper claims that standard Layer 2 attacks such as MAC floods, gratuitous ARP, and VLAN hopping are not possible in InfiniBand.

In contrast, Lee et al. from Texas A&M and the Intel Corporation believe that the IBA specification has omitted security, resulting in potential vulnerabilities that could be exploited with moderate effort [34]. The authors did not orchestrate a particular attack themselves but base their argument on the lack of encryption in InfiniBand. Lee et al. are concerned by the fact that powerful keys used for authentication and management are sent over the wire in plaintext. The thought is that these keys could be easily obtained from a traffic sniffer then spoofed to powerful effect. The authors infer that having infiltrated an InfiniBand network, a hacker could abuse the extensive computational power and massive storage capacity of the cluster, which could be "used

for another attack and as a repository for illegal contents." Even though their focus is on data confidentiality, Lee et al. proposed a security enhancement to protect against Denial of Service (DoS) attacks. They built a simulation testbed with a stateful partition enforcement mechanism in switches using trap messages. They wanted to filter all types of packets with an invalid Partition Key.

2.5 Cyber Vulnerability Assessment

A Cyber Vulnerability Assessment (CVA) is an integral part of a good security program. It is the process of identifying and analyzing security vulnerabilities that might exist in a computer system. The term 'system' here usually refers to a network or enterprise but can be an individual device or component. Vulnerability assessments are typically conducted through "network-based or host-based methods, using automated scanning tools to conduct discovery, testing, analysis and reporting of systems and vulnerabilities. Manual techniques can also be used to identify technical, physical and governance-based vulnerabilities" [35]. The reader may also be familiar with related types of 'assessments' such as ethical hacking and penetration testing. The finer distinctions between these will not be made in this thesis, but it is significant to note that vulnerability assessments are not exploitative by nature, meaning that "practitioners (or the tools they employ) will not typically exploit vulnerabilities they find .. a suspected vulnerability does not typically need to be exploited to identify its existence and apply a fix" (if known and available) [35].

Conducting a CVA consists of two main objectives, the planning and performing of the vulnerability assessment. The planning portion is extremely important, because it entails "gathering all relevant information, defining the scope of activities, defining roles and responsibilities," and more [36]. A CVA of a production network includes interviewing system administrators and reviewing appropriate policies and procedures

relating to the systems being assessed. These and other recommended actions will be described later in this thesis, because our test network merely consists of a few Linux hosts connected to a single switch. As a result, this is a technical assessment only.

The process of defining the assessment scope is almost always up to the customer or network owner. It determines what entities are in play, but the execution strategy is usually up to the assessor. Many cyber experts believe in looking from an attacker's perspective by employing the so called "hacker methodology." This progression lists the stages of a cyber attack from reconnaissance and enumeration to covering tracks and exfiltration of data. Physical attacks are carried out in much the same manner, and of course each type of attack does not necessarily incorporate every stage of the progression. Some of the codified models include Lockheed Martin's "Cyber Kill Chain" [37], the MITRE Corporation's Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) Matrix [38], and the STRIDE model by Praerit Garg and Loren Kohnfelder at Microsoft [39]. (MITRE is a not-for-profit company that operates multiple federally funded research and development centers.)

Lockheed's Kill Chain started with a 2011 company white paper and is focused on the concept of the Advanced Persistent Threat (APT). The authors define APTs as "well-resourced and trained adversaries that conduct multi-year intrusion campaigns targeting highly sensitive economic, proprietary, or national security information" [37]. The acronym is further broken down such that Advanced denotes these actors are targeted, coordinated, and purposeful, and Threat specifies person(s) with intent, opportunity, and capability. The crux of Lockheed Martin's approach is intelligence-driven computer network defense, a risk management strategy that addresses the threat component of risk, incorporating analysis of adversaries, their capabilities, objectives, doctrine and limitations. The Cyber Kill Chain stages are as follows: reconnaissance, weaponization, delivery, exploitation, installation, command and control

(C2), and actions on objectives.

As an aside, the phrase APT is also used by FireEye Inc. (formerly Mandiant) to designate major cyber groups suspected of receiving direction and support from an established nation state. Cyber attribution, the process of tracking, identifying and laying blame on the perpetrator of a cyberattack, is extremely difficult to do with complete accuracy due to attacker obfuscation techniques making the long and painstaking process of performing computer forensics even more challenging. Nonetheless, obtained code fragments and chatter within the hacking community can provide valuable clues. FireEye has assigned names to various groups, such that APT 1 is associated with China, APT 29 aligned with Russia, and APT 33 with Iran, for example [40].

The STRIDE model highlights types of cyber attacks, rather than their stages. It is a mnemonic for security 'threats' in six categories:

Spoofing (of user identity)
Tampering
Repudiation
Information disclosure (privacy breach or data leak)
Denial of service
Elevation of privilege

This STRIDE model is a good albeit generic way to classify attacks, but it does not delve into the progression of attacks nor does it investigate individual attacker techniques.

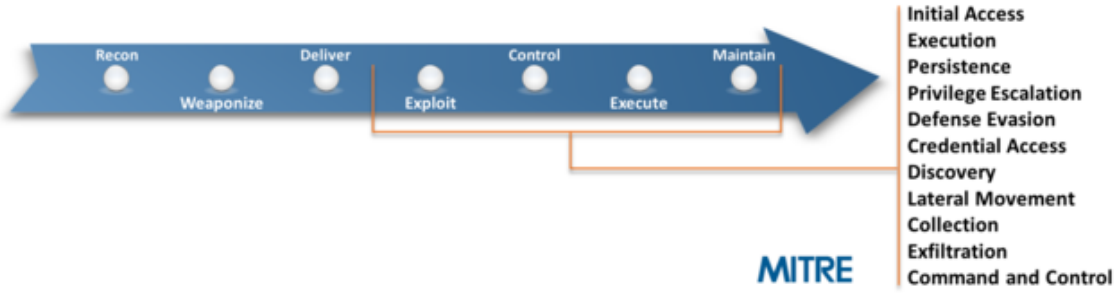
That is the strength of MITRE's framework. In addition, the ATT&CK Matrix has been widely accepted within US governmental cyber communities, so we have

chosen to adopt it in this thesis. The list begins with gaining initial access on a device. All adversarial actions taken prior to establishing a foothold in the network are covered by the PRE-ATT&CK Matrix. For the sake of our testing, the assumption was made that a host or other device on a generic InfiniBand network could be compromised somehow. The method or means by which that unintended access might be acquired was not considered, because our setup is a closed network. However, Chapter 3 will cover PRE-ATT&CK considerations.

The full ATT&CK Matrix includes techniques spanning Windows, macOS, and Linux platforms. Many of these are OS dependent. InfiniBand is supported by newer Windows distributions, but we will focus on Linux-style attacks, because Linux is more prevalent in high-performance networks. No cyber attack model explicitly incorporates InfiniBand, so this thesis seeks to discover and document specific techniques using the ATT&CK tactics as guidelines.

The 11 ATT&CK Matrix tactic categories, shown in Figure 4, are the column headings of the table. They are as follows: Initial Access, Execution, Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement, Collection, Exfiltration, and Command and Control (C2) [38].

Figure 4. MITRE Corporation Enterprise Tactics



These functions are typically performed in that order chronologically, although attackers have their own tradecraft and preferences. The available time on target

and required stealthiness can also influence the sequence of events. Execution is the means by which the cyber effect is produced. Common options include a command line interface (CLI), a graphical user interface (GUI), a script, or a compiled binary. Persistence allows an attacker to quickly and/or easily regain access to that device should the connection be severed for whatever reason. Privilege escalation means increasing one's level of access to files, directories, and programs. Ideally the attacker would have full administrative rights such as being able to modify, add, or delete anything on the file system. In Linux the default administrator is the 'root' account.

Defense evasion is bypassing security measures (i.e. anti-virus software, firewalls) and avoiding detection. Credential access is the process of harvesting usernames, passwords, Personal Identification Numbers, and even cryptographic keys. Discovery is how the attacker learns what other systems are on the internal network. After that is accomplished, the decision may be made to attempt to pivot to another device. This is called Lateral Movement. Collection is assembling and staging the victim's data so that it can later be sent to a location of the attacker's choosing. Lastly, C2 is how the attacker communicates with his or her malicious beacons and implants.

Cyber methodologies such the ATT&CK Matrix describe how a hacker perpetrates a cyber attack. The next chapter of this thesis will cover how to harden a high-performance network against attacks, specifically how to plan a CVA. InfiniBand is a high bandwidth, low latency interconnect with many functions performed in hardware instead of software. This presents additional challenges for both attackers and defenders.

III. Methodology

This chapter will lay the foundation for conducting a Cyber Vulnerability Assessment (CVA) of InfiniBand. It will consider the motivations behind performing an assessment, how one should be properly scoped, and what equipment and personnel would be necessary to evaluate a live, production InfiniBand network. Possible sources of attack will be explored so that likely ingress points can be hardened prior to the cyber defense team's arrival. In parallel with this notional assessment, planned attack primitives, which will be tested later on our small, laboratory network, will be described at the end of the chapter.

3.1 Assessment Purpose and Scope

Establishing the purpose and scope of a CVA is not a trivial task, yet it is a critical one. Performing vulnerability assessments on complex, high-performance computing (HPC) systems is a difficult and costly operation, not only due to the equipment involved but also for the necessary experience and expertise amongst members of the assessment team. The customer's motivation in hiring the team may drive much of the planning effort. Sometimes organizational policy dictates having external (as in not accomplished by employees), periodic assessments done, and other times leadership wants to be reassured that their valuable systems are 'safe.' Ideally, impromptu assessments would be initiated based on credible threat intelligence indicating that cyber threat actors are targeting specific industries, companies, or types of networks or that an intrusion was detected on a similar system. The reality is that this exists only to a very generic extent, and larger organizations are often afraid to publicly admit that their network was breached. Government agencies do not want to lose their citizens' trust and to embolden their enemies, whereas companies fear plummeting

stock prices.

This lack of information sharing between organizations is harmful to the collective. The power of anti-virus software, for example, is that a malware sample discovered by Kaspersky researchers in Russia will likely result in the generation of virus definitions that could protect a laptop computer in Brazil days later. Regardless, cyber threat intelligence is not always accurate or up-to-date anyway. Peer network owners are often not even aware that they were victims of an attack for months after the fact, because more advanced adversaries are far better at avoiding detection than their less sophisticated counterparts. So, cyber threat intelligence is an important resource, but complete reliance on it is unwise.

Once the need to perform a CVA has been established, discussions between the client and representatives from the assessment team will begin. At the macro level, a management decision must be made as to whether to cover policy and procedures only, to conduct a technical assessment, or both. A major factor in this decision is whether the InfiniBand network is live and must have 24/7 availability. The network owner may not allow assessors to bring in or connect their own equipment, but this is preferred for several reasons. First, production machines have programs and processes actively running on them. These processes may be using much of the system's processor and memory resources, so an assessor working on a live node could incidentally cause a crash while performing his or her duties. Secondly, cyber teams occasionally get contacted because of a suspected network intrusion or anomalous system behavior. It is beneficial to both parties that one does not accidentally infect the other, and the team must be fully confident that their tools have not been compromised. Lastly, the team should (at least initially) be assigned their own subnet, partition or Virtual Lane (VL) for logical segregation in order to limit network disruption and to allow for pseudo-external testing.

Penetration testing starting from outside the network's perimeter or from a Demilitarized Zone (DMZ) is also a possibility, but this is usually not included in a CVA. A penetration tester's inability to gain access to an internal network within a limited amount of time does not prove that it could not be done eventually or that it has not been done previously. Keep in mind that Advanced Persistent Threats (APTs) commonly perform reconnaissance on a target for months or years at a time.

Prioritization of effort in a CVA should be based on risk. Identify the assets and define the risk and critical value for each device, based on the client's input [41]. A master Subnet Manager and a Windows Domain Controller are two examples of critical nodes, part of what the military cyber community refers to as key cyber terrain. Non-essential nodes may be omitted from the assessment, time permitting, but it is worth noting that attackers often choose such locations to establish persistence, precisely because they are not heavily defended or well monitored. Ethernet networks commonly make use of automated vulnerability scanners such as Nessus by Tenable Inc. [42]. It uses plugins to perform configuration audits, patch management, find (known) software vulnerabilities, etc. A vulnerability scanner for InfiniBand does not exist, but an IP-based scanner could be run if Internet Protocol over InfiniBand (IPoIB) is enabled on the network.

Policy assessments review topics such as an organization's asset management, access control, user awareness and training, data security, maintenance procedures, as well as response and recovery planning. This thesis is not an authoritative source on Information Technology (IT) policy-related issues. Instead, it is recommended that assessors create their own checklists based on the National Institute of Standards and Technology (NIST) cybersecurity framework or another industry accepted standard [43]. Many items will not apply, since these frameworks were written for Ethernet networks, but they are solid models. Also be aware of industry-specific

security guidelines and regulations such as the Cloud Controls Matrix (CCM) for cloud service providers and the Federal Financial Institutions Examination Council (FFIEC) IT Examination Handbook for the financial sector [35]. Sean Peisert wrote a very insightful article for Communications of the ACM, detailing unique security considerations for HPC environments [44]. These topics will be covered in Section 3.5.

Focus should always be on the customer's mission. This must be well understood so that the cyber team can tailor a solution in terms of the assessment duration, facility access, work schedule, etc. The team should request as much information about the network as possible ahead of time in order to identify potential network susceptibilities and to formulate an assessment strategy. If the client consents to giving the team copies of software used on their network, the team can setup a small, prototype network on which to perform initial testing.

3.2 Equipment Setup

A prototype network was what the author used for this thesis. It is minimal but allows for a technical CVA through experimentation with basic attack primitives (see Section 3.6). The network consists of three desktop computers and one Mellanox SX6012 switch. The switch has twelve ports, each capable of 56 Gb/s full bi-directional bandwidth. It also has Ethernet, RS232, and Mini-USB management ports for out-of-band maintenance [45]. The computers are high-performance desktop machines capable of handling the requirements of this assessment. They have identical hardware and software. The only significant difference between the nodes is that host #3 was chosen to be the Subnet Manager. As a result, it is running the OpenSM program in master mode. In addition, host #3 is connected to the switch via an RJ-45 to DB9 serial cable. This does not affect the normal in-band traffic,

although it could provide an attacker a way to access the switch. Figure 5 shows the actual equipment used.

Figure 5. AFIT InfiniBand Network

A production InfiniBand network will probably have dozens to thousands of nodes. As previously mentioned, a cyber defense team should bring some of their own equipment to the assessment. The customer may employ Windows, Linux, or a mix of operating systems, so the team should be prepared for all eventualities. Likewise, the Channel Adapters may be solely running InfiniBand or a combination of InfiniBand and Ethernet (RoCE or iWARP), such as the Virtual Protocol Interconnect (VPI) cards by Mellanox [18]. It is suggested that the team acquire CAs from a couple different manufacturers for the sake of familiarity and in case of compatibility issues.

Programs written using Message Passing Interface (MPI) should run on any InfiniBand platform, but the verbs may be somewhat vendor-specific. InfiniBand verbs are supplied by the Host Channel Adapter (HCA) driver [46].

As for the cyber team's mobile kit, it should have at least two high-performance PCs (depending on the size of the network), preferably mounted with removable solid state drives. Several copper and fiber cables should be included, ideally EDR quality or better. A forensics pack will be needed, containing spare, blank drives, write blockers, a disk cloner, as well as various connectors and adapters (e.g. NVMe/NVMHCIS, SATA-to-IDE, FireWire).

Vehicular networks may have additional requirements. If cannon plugs or other non-standard connectors are required to connect to these systems, they should be provided by maintenance personnel. This topic must discussed in detail prior to arriving at the vehicle's location. Personal protective equipment will likely need to worn while working on or near the vehicle. Additionally, physical access to individual components may not be possible or might take much longer it would with traditional server racks. These are but some of the planning considerations that should be addressed for vehicular systems.

3.3 Team Composition/Required Skillsets

Assembling a cyber team with varied knowledge of HPC systems and advanced cyber techniques will likely be difficult. Many disciplines are involved. Innovation is key, because true cyber tools have not been written for InfiniBand. Popular information security mainstays such as Nmap, Scapy, Metasploit, Nikto, Maltego, and Ettercap do not support InfiniBand. All of them rely on crafting packets in software and/or communicating with virtual ports. So, an assessment team would need to build their own tools and scripts or at least develop capabilities to demonstrate system secu-

rity weaknesses. Creativity is needed to find network susceptibilities using techniques that are outside of traditional engineering methods and thought processes. Important technical skills include: a deep and low-level understanding of hardware and software design, ability to reverse engineer portions of hardware and software, an understanding of various methods and strategies for exploiting systems, an understanding of parallel and distributed systems, and an ability to create interfaces/simulations to monitor and stimulate communications.

No single individual will possess all of these traits, but a talented group can. A team of people from varied backgrounds should benefit from different thought processes, skills, and experiences. A word of caution is warranted, though; InfiniBand networking (and HPC clusters in general) is so fundamentally different from standard Ethernet that all members should be familiar with the InfiniBand protocol and architecture, as well as types of HPC applications. Hiring a cybersecurity professional with no background in non-traditional computer systems could be disastrous, or at least necessitate weeks to months of training and exposures before he or she is useful to the team. As for programming skills, Remote Direct Memory Access (RDMA) and MPI programming can be accomplished in C, so experts in that language are extra valuable. Python, BASH, and Powershell scripting knowledge along with experience using Windows and Linux operating systems are a must as well. (Linux is more prevalent in HPC systems for obvious reasons.)

The OpenFabrics Enterprise Distribution (OFED) software suite should be common to all InfiniBand equipment, but vendors make some of their own diagnostic tools and all of their own hardware. Experience with equipment from different manufacturers is useful, but this can be acquired on the job. InfiniBand HCA and switch Application Specific Integrated Circuits (ASICs) are custom chips, so knowledge of chip design and testing is needed. More important, though, is having a team mem-

ber(s) with experience writing or modifying firmware.

3.4 Sources of Attacks

Defense in Depth is always the preferred method of protecting computer networks. However, HPC systems tend to be flatter networks, making Defense in Depth more difficult to implement. Adding more network hops and performing deep packet inspection increases latencies, a consequence that is deemed to be counterproductive. Regardless, most HPC clusters are fairly closed networks, so the most critical step for a defender is to prevent hackers from gaining initial access.

E-mail phishing attacks are the easiest and most common ways to gain a foothold in traditional networks. Network engineers will be targeted by cyber actors, especially if they leave their resume and contact information on hiring websites (e.g. Indeed.com, LinkedIn, ZipRecruiter, CareerBuilder). Second to phishing-style attacks are vulnerable websites that can be exploited by SQL injection, Cross-Site Scripting, input validation, etc. Corporate networks will definitely be susceptible to these avenues, but research LANs should not. Despite this, many supercomputer systems can be reached remotely via an internal web interface (i.e. from within the campus network) or through a Virtual Private Network (VPN) Concentrator. Vehicular networks have to consider a multitude of wireless signals and protocols.

Each network is unique, so there is no catch all solution. Network maintainers must focus on how information gets in and out. The concept of "connected versus connectable" applies to well segregated networks. Windows Update, anti-virus suites, and Linux software installers (apt, yum, rpm, dpkg) make it easy for users to acquire and update their operating system and programs, but these require an Internet connection. Manual package downloading and installation are possible as well, but it is imperative that the files are obtained from a proper website such that of the original

equipment manufacturer or a trusted source like the OpenFabrics Alliance. Shared repositories (i.e. GitHub) are very popular and widely used but are not ideal from a security standpoint.

If files are downloaded manually, all corresponding cryptographic hashes (e.g. MD5, SHA-1, SHA-256) posted on the download page should be verified before any files are installed. This is not a guarantee that the contents are safe as the website could be compromised, and collision attacks have been successfully demonstrated on the older algorithms, MD5 [47] and SHA-1 [48]. Removable media arriving through the mail have similar concerns.

No network is completely closed off, so even janitorial staff, HVAC technicians, and vehicle maintainers are potential threat vectors. Customers of expensive equipment often pay for the service of having a representative from the manufacturer visit regularly. All employees and visitors must be vetted (and preferably given badges) prior to admittance in to the facility. Physical access is always the most powerful type, especially because HPC nodes, as with Industrial Control System (ICS) hosts, are often left logged in when unattended or with a blank or default password. USB ports can (and should) be disabled when not in use. Fortunately, autorun functionality is not enabled by default anymore.

3.5 Assumptions & General Security Evaluation

Most of what has been discussed thus far in this chapter is applicable to many types of computer systems and CVAs. But this thesis would not be a valuable contribution to academia if it did not really emphasize why and how security concerns and potential solutions for them are drastically different in HPC networks from those in typical networks. The intent here is to dispel any assumptions the reader may have [44]:

1. HPC systems are optimized for high **performance**.

This sounds like a feature that many networks should desire. However, security is often at odds with performance, because the former wishes to add more checks, restrictions, segregation, and complexity. The latter wants to improve efficiency and remove all unnecessary delays.

2. HPC networks are usually designed to fulfill a **singular purpose**.

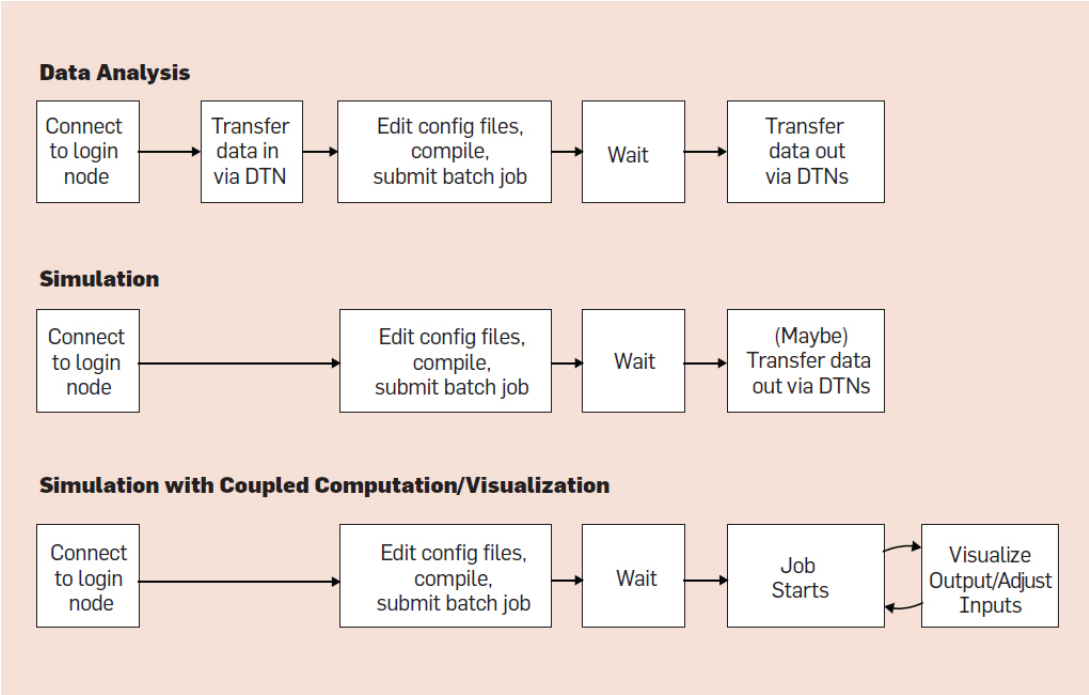
   They are typically used for automated computation, commonly performing some set of mathematical operations or running non-polynomial time algorithms. Historically, the purpose has been to do modeling and simulation, but data analysis is becoming a more frequent task. Processing time on these systems is costly, and jobs are scheduled in a queue. Downtime is seen as lost revenue or as missed opportunities for research. By contrast, most traditional networks are general-use. They provide various services. A user may want to access network storage via SMB or FTP, send and receive e-mail traffic over SMTP/POP3/IMAP, connect to another machine using RDP, SSH, or Telnet, or communicate with a website using DNS and HTTP(S).

3. Users have a **distinct interaction** with HPC networks.

   Most computer applications require constant, or at least regular, human interaction. They are often graphical user interfaces where aesthetics are valued. HPC applications rarely involve extensive graphics. They run millions of scientific calculations faster than a human can respond, so this is why scripts are written and given a series of inputs and/or parameters. Programs usually do not run on the local machine; they are initiated on a login node and sent to compute nodes for execution. The user then waits for minutes to days for job completion, after which time results can be retrieved through a data transfer node (DTN). Figure 6 shows high-level workflow diagrams for different types of

scientific computing.

Figure 6. Common HPC Workflows



HPC networks have other traits that distinguish them from traditional ones, but this thesis is primarily concerned with the InfiniBand interconnect. Having clarified some assumptions about HPC network usage, we will now investigate the security implications of InfiniBand as a system. A general evaluation of InfiniBand security must take a holistic approach, looking at its protocol, physical equipment (hardware), supporting software, and network architecture.

3.5.1 InfiniBand Hardware Attacks.

Of these, hardware vulnerabilities are the most challenging for defenders, because they are by far the most difficult to detect. InfiniBand switches and HCAs use custom ASICs that are currently capable of sending and receiving data at rates up to 200 Gb/s per port. The newest Mellanox product lines are Quantum [49] and ConnectX-6

[50], respectively. InfiniBand hardware is cost prohibitive for individuals and smaller businesses, so these chipsets have not been externally tested or publicly vetted to the same extent that more general purpose hardware (e.g. Intel i7 CPU) has been. In addition, there has not been any published assembly language or microcode, and, as a result, programmers and users must use vendor-specific tools and application programming interfaces (APIs). So, if hardware vulnerabilities or backdoors were to exist, they would be nearly impossible to come upon without any insider knowledge. Reverse engineering a microchip begins with a very expensive and time consuming physical process called delidding, progressively stripping off layers of the chip. Then one can take photos of chip cross-sections using a scanning electron microscope [51]. Modern ASICs have over a billion transistors, so delidding one of these chips would take a considerable amount of time. A far more likely threat to InfiniBand hardware is supply chain tampering.

NIST lists cyber supply chain risks as "insertion of counterfeits, unauthorized production, tampering, theft, insertion of malicious software and hardware, as well as poor manufacturing and development practices in the cyber supply chain" [52]. In the case of InfiniBand or other HPC products, this would be an extremely sophisticated attack, probably necessitating nation state level support. Malicious variants of the ICs could be produced with an embedded rootkit or logic bomb. These could then be substituted in for the original chips while the devices were in transit between the manufacturer and the customer. The hope would be that the compromised devices would find their way into facilities or onto networks otherwise out of reach of the attacker. Testing for hardware vulnerabilities is such a complex operation that it will not be discussed further in this thesis. However, firmware modification is much more feasible, so it will be covered as an attack primitive later in this chapter.

A major security advantage of InfiniBand hardware is that packet crafting is done

in the HCA, not in the CPU. Malformed packets cannot be tailored using software with the way that HCAs are currently configured. Arbitrary metadata fields of a packet cannot be altered without going through the vendor's API. Possible exceptions to this are installing malicious firmware to reprogram the HCA and/or using a different driver with a special set of verbs. Legitimate verbs check the validity of inputs before creating a protection domain, registering a memory region, using a hardware device, etc. A side effect of hardware packet crafting is that traffic sniffing requires a custom tool (e.g. ibdump) instead of standard programs like tcpdump and Wireshark. Without the ability to manipulate packets, attackers would not be able to perform *active* man-in-the-middle attacks. Even if it were possible, software packet crafting is much slower than doing so in hardware, so this might cause noticeable network delays and congestion.

3.5.2 InfiniBand Architecture Evaluation.

The next portion of InfiniBand networking to evaluate is the architecture. The switched fabric topology is not impervious to attack, but it does have advantages over traditional switched networks. Having more than one physical connection to the rest of the subnet provides redundancy and resiliency. If one link were to go down, that endpoint should still be able to communicate through its other Channel Adapter port. Espionage is more difficult to carry out and generally less successful when potential routes between two endpoints do not share at least one common intermediate node. Valuable intelligence can be gathered when a cyber actor performs traffic sniffing from a Switched Port Analyzer (SPAN) port on an Ethernet switch; all (or selected) traffic traversing that switch is mirrored out a different port to the host the listener is on.

InfiniBand nodes are not always connected to just a single switch, so in theory only a portion of the packets headed to or from a specific node would transit through the

compromised switch. This depends on the shortest paths calculated by the Subnet Manager (SM), resulting in forwarding tables unique to each node. Switched fabrics cut down the attack surface by taking away the ability for switches to perform dynamic routing. This is not really a security enhancement of the architecture itself but rather the instantiation by InfiniBand. Ethernet attacks such as ARP cache poisoning and routing table overflows are not possible because of InfiniBand predetermined routes. Table 6 below summarizes some of the advantages of switched fabrics [8]. These benefits are more pronounced when fabrics are compared to bus structures still used in most vehicular networks.

Table 6. Switched Fabric vs. Bus Topology

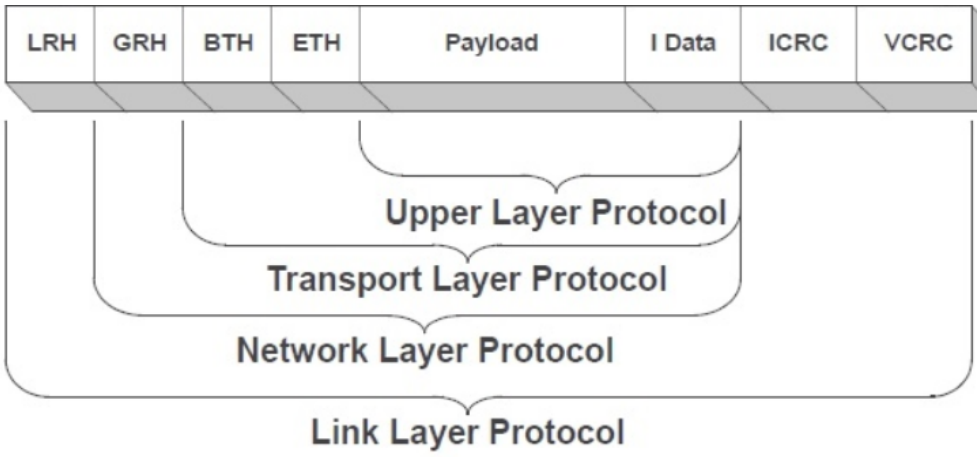
Feature	Fabric	Bus
Topology	Switched	Shared Bus
Pin Count	Low	High
Number of Endpoints	Many	Few
Max Signal Length	Kilometers	Inches
Reliability	High	Low
Scalability	High	Very Limited
Fault Tolerance	High	Low

3.5.3 InfiniBand Protocol Features.

The InfiniBand protocol follows the Open Systems Interconnection (OSI) model as previously discussed in Section 2.1. See Figure 7 for a visualization showing the layers of encapsulation. The Local Route Header constitutes Layer 2, the Global Route Header Layer 3, and Layer 4 consists of both the Base Transport Header and the Extended Transport Header. The Local IDentifier (LID) as well as Service Level and Virtual Lane information are in the Local Route Header. The Global Route Header has the IP version and the Global IDentifier (GID), but this header is omitted entirely during local (within a subnet) transmissions. The Packet Sequence Number (PSN)

as well as queue, partition, and memory keys are all in the Layer 4 headers. The fields in the Extended Transport Header differ based on the Base Transport Header operation or the Local Route Header next header. The Invariant and Variant Cyclic Redundancy Checks (ICRC and VCRC) are the respective checksums for bits that do not change during the transmission and those that are recalculated at each hop. Breaking the checksum up into two parts has the added benefit of making packets slightly harder to spoof, were that easily accomplished. This also decreases the work that switches have to do, thereby lowering transmission delay times.

Figure 7. IBA Data Packet Format



** Graphic courtesy of the InfiniBand Trade Association.*

An obvious security concern of the InfiniBand protocol is the omission of data encryption. InfiniBand packets are always sent in cleartext natively. Encrypting the payload and possibly some of the metadata contained in the higher levels of encapsulation (Protocol Data Unit headers) could improve data confidentiality. The IP stack offers encryption down to Layer 3, in the form of Internet Protocol Security (IPsec), but other common protocols like Transport Layer Security (TLS) and Secure Shell (SSH) operate at Layer 5 or above. The InfiniBand Trade Association (IBTA) chose not to implement encryption, because it adds system complexity and is computationally expensive, significantly increasing network latencies. Technically, SSH is still

available if one enables IP over InfiniBand (see Section 2.1), but even this does not hide the memory keys and other important keys since it only encrypts a portion of the data (that is, the IP payload within the InfiniBand payload). Another dangerous, albeit necessary, feature is forced routing. This could allow an attacker to ignore forwarding tables and send a packet along any pathway they desired. Positive security attributes include having Virtual Lanes, various keys (even if sent in the clear), and unique Queue Pair Numbers.

3.5.4 InfiniBand Software Susceptibilities.

Lastly, we examine InfiniBand supporting software. OpenMPI and the OFED suite are primary means for network users and programmers to interact with the fabric. Both are open-source software projects, a fact that improves collaboration and development within the HPC community. On the other hand, this aids would-be attackers as well. OFED code could be modified relatively easily and recompiled to create a new cyber weapon. A malicious variant might be extremely stealthy if it maintained all or most of the functionality of the original code. OpenSM is one such application ripe for exploitation, because it is a powerful tool that determines all traffic routing within a subnet. Manufacturer scripts on the file system provide another way to get acquainted with the inter workings of InfiniBand, especially considering that many hackers are not as adept in lower-level programming languages like C and Fortran. (Knowledge of assembly language is critical for software reverse engineering, so that is a different story.) InfiniBand diagnostic tools comprise the majority of the OFED binaries. Binaries can be overwritten, fuzzed for input validation vulnerabilities, and/or brute forced by testing all command line (or graphical) options.

3.6 Planned Attack Primitives

This section will explain the types of cyber attacks that will be attempted later. Some are feasible on Ethernet networks, so the intent here is to launch equivalent attacks in InfiniBand. The particular vectors mentioned in this thesis have been selected based on the authors' previous experience and research, using the MITRE Corporation's ATT&CK Matrix as a guide. Of the 11 tactic categories from the matrix, not all of them pertain to InfiniBand. Specifically, Initial Access, Persistence, Privilege Escalation, and Defense Evasion involve methods particular to the Operating System being employed.

3.6.1 Category: Execution.

Security researchers and the hacking community have created a multitude of cyber tools for Ethernet networks. Very few if any of these would function in InfiniBand due to hardware packet crafting, the lack of virtual ports, etc., without activating IPoIB or EoIB. Using these Upper Layer Protocols is a legitimate technique, but this paper does not consider InfiniBand "running in Ethernet mode."

- **OFED Diagnostic Tools**

  The diagnostic utilities in the OFED suite provide means to debug the connectivity and status of InfiniBand devices in a fabric. Due to the lack of custom cybersecurity tools in InfiniBand, these utilities could provide the building blocks for future cyber weapons, enabling an attacker to manipulate settings and network traffic in unintended ways. Hackers with remote access are more frequently "living off the land" instead of relying on malware to bring the desired effect. Running standard operating system commands and using provided diagnostic tools are much stealthier techniques compared to transferring and

executing files not native to that environment. OFED tool usage should not set off any alarms, nor does it put the attacker's code at risk of being quarantined or found.

All of the OFED diagnostic tools will be studied and tested. Individual packet captures will be taken for each one to find out what network traffic is generated during execution. Not every possible combination of parameters will be run. Instead, a few options that appear to have more dangerous or powerful ramifications will be chosen and tested (e.g. ibping with the flood option).

- **RDMA Programming**

  RDMA programming for IB, RoCE, and iWARP is accomplished via the verbs API. Mellanox states that their architecture "permits direct user mode access to the hardware" through a "dynamically loaded library" [18]. Networking experts can program with verbs in order to customize and optimize the RDMA network or to enable destructive actions. Anyone writing software for an InfiniBand network will access the verbs or a cross-technology solution such as MPI. Custom software will not be written for this thesis, but this potential attack vector is still worth mentioning.

- **Malicious Firmware Installation**

  Firmware is embedded code that programs a particular hardware device. In this case, it is HCA (and possibly switch) firmware that is of interest. Reprogramming an InfiniBand HCA could allow an attacker to modify outgoing and intercept incoming packets. Firmware is chipset-dependent, so no code modification would be guaranteed to work on all InfiniBand HCAs. This research does not include firmware development but shows how malicious firmware could be burned into a device.

3.6.2 Category: Credential Access.

InfiniBand does not use usernames and passwords for authentication, nor does it require access tokens or tickets like Kerberos. Usernames and passwords are OS mechanisms meant for human users, but HPC clusters usually run automated processes. Instead, matching source addresses and keys grant communications access between nodes. Administrative privileges are needed to run most InfiniBand tools.

- **Address Spoofing**

  Firewalls and Intrusion Detection/Prevention Systems (IDS/IPS) often block traffic or generate alerts based on the source (MAC and/or IP) address. Modifying one's source can allow an attacker to bypass these middleware devices. Spoofing can cause the destination computer to grant elevated permissions if no other authentication/authorization is in use. Address spoofing can also enable 'reflected' attacks, since any responses would be sent to the true owner of the spoofed address. Popular 'hacking' tools such as Nmap and Scapy can easily accomplish this attack:

  nmap -S $IP_Address

  ifconfig eth0 hw ether $MAC_Address

  nmap –spoof-mac $MAC_Address

  Address spoofing is an attack on data confidentiality, because it can allow an unintended user or computer to gain access to otherwise denied resources. This thesis will attempt to duplicate the GUID spoofing accomplished by Aron Warren in 2012 [33]. LID spoofing will be investigated as well.

3.6.3 Category: Discovery.

A cyber actor can learn the addresses and structure of an internal network in different ways. Typically, discovery is performed passively through traffic analysis and/or actively through scanning. The former utilizes programs such as tcpdump and Wireshark, whereas the latter involves tools like traceroute, Nmap, SolarWinds, and many more.

- **Network Traffic Sniffing**

  Traffic sniffing can give an attacker valuable situational awareness about a network. It may not be thought of as an attack in and of itself, but it is an attack on the confidentiality of the data. Just like in many forms of communication, monitoring network traffic requires the perpetrator to be positioned between the sender and the recipient unless he or she is only interested in conversations involving one particular participant. The reason for this is that messages are not always sent out to every device, unless the topology of the network is a shared bus, or a network hub is used instead of a switch or router. Broadcast and multicast messages exist in both Ethernet and InfiniBand, but these are not 'private' messages. Therefore, switches tend to be the preferred target on which to perform sniffing. Methods to perform InfiniBand traffic sniffing will be explored in the next chapter.

- **Network Mapping**

  The ideal byproduct of the discovery step is a complete and accurate network map. Not every node communicates regularly, so passive monitoring may not be sufficient; active scanning may be necessary to identify all connected devices. Even during relatively idle times, Ethernet networks produce a lot of noise in the form of Address Resolution Protocol (ARP), Network Time Protocol (NTP),

and Simple Network Management Protocol (SNMP) traffic, among others. InfiniBand, on the other hand, has very little overhead. A network mapping tool, especially one with a visual display, could provide InfiniBand users and maintainers with valuable situational awareness about their network. A few of the OFED diagnostic tools deliver this functionality in text-only output form, so this thesis will attempt to augment one or more of these tools with a graphical interface.

3.6.4 Category: Lateral Movement.

Traditional pivoting is not necessary in an InfiniBand network, because remote interactive logons are not used (again, excluding SSH via IPoIB). HPC clusters automate work using scripts, and nodes share resources. In a sense, hosts are extensions of each other due to RDMA, far more so than file sharing via Server Message Block (SMB) or File Transfer Protocol (FTP).

- **Subnet Manager Hijacking**

  As previously discussed, the Subnet Manager is an extremely powerful entity in InfiniBand networks, somewhat analogous to a Windows Domain Controller. There must be one master Subnet Manager, but several other nodes can be running in slave or standby mode. These backups perform some form of polling (vendor specific) to ensure the master is operational, and failover to one of them when it is not [9]. The OpenSM software is open source, so an attacker could download the source code, modify and recompile it [53] [54]. Having done so, the next step would be running a weaponized version of OpenSM on the compromised machine and performing some kind of Denial of Service so that the master Subnet Manager was unable to communicate, thereby getting replaced as master.

Subnet Manager Hijacking could affect the integrity, confidentiality, and/or availability of the network depending on the design and operation of a malicious Subnet Manager. This thesis will not create a weaponized version of OpenSM but will demonstrate how to cause the master Subnet Manager to fail remotely so that another instance could take over.

- **Out-of-Band Switch Access**

  In-band switch management is not possible in pure InfiniBand; switches do not have a service that is listening for connection requests. Multi-interconnect systems such as Mellanox's Virtual Protocol Interconnect (VPI) do not follow this rule, because they can run Ethernet and InfiniBand simultaneously. As such, they can allow SSH and HTTP(S) logins. The intended method of managing an InfiniBand switch is by connecting a serial cable from a host to the switch's management or console port. Then one can use HyperTerminal, minicom, PuTTY, or another serial terminal program to receive a login prompt. One should follow the configuration settings (e.g. baud rate) in the switch's user manual, a copy of which should be accessible online.

  Unless an attacker has physical access to the switch, he or she cannot simply bring in a laptop and connect it. A more plausible scenario would be that a network administrator left the switch physically connected to a nearby node, probably to some kind of management terminal. It is very possible that the default credentials (username and password), which can also be found in the user manual, have never been changed. So, if an attacker could gain access to this connected node, pivoting to the switch might be easy. The following Linux command checks for existing serial devices:

  $$dmesg \mid grep\ ttyS$$

The default is usually **/dev/ttyS0**. The result of an attacker gaining console access on an InfiniBand switch could be just as damaging as that of the Subnet Manager Hijacking discussed previously. The difference might be the level of impact in that there is only one Subnet Manager per subnet whereas there can be multiple switches. Nonetheless, commands run on a switch can have an impact on the entire fabric. This thesis will explore connecting to a switch directly and remotely.

3.6.5 Category: Collection.

Collection is the method by which an attacker obtains the information they were seeking.

- **Falsified Memory Keys**

  Unintended acquisition of RDMA memory keys could allow an attacker to read from or write to a remote memory region. This attack cannot be attempted unless a way to manually craft or manipulate packets is discovered. Even if cleartext memory keys were obtained from packets transiting the network, these would not be usable as is.

3.6.6 Category: Exfiltration.

This tactic category is being more loosely interpreted as creating the desired effect, instead of merely exfiltrating data from a victim device or network. The reason for this change was that Denial of Service (DoS) attacks do not fit nicely into any MITRE ATT&CK Matrix category.

- **Denial of Service Attacks**

Denial of Service attacks attempt to disable a node or program by various means, including consuming all its resources or shutting it down entirely. This is most often a means to an end, such as allowing an attacker to thwart defenses or migrating to a failover service or situation that may be more advantageous. A common DoS attack is a ping flood. The goal is to saturate a victim machine or network link with so many ping packets that legitimate traffic is dropped or severely stalled. The ibping command will be run in one (command line) terminal on one machine, on multiple terminals on one machine, and finally on multiple machines.

IV. Analysis

This chapter will describe the process of conducting a Cyber Vulnerability Assessment (CVA) on an InfiniBand network once planning and preparation have been completed, and the assessment team is on site. At a high level, this framework is not significantly different from how other CVAs are conducted. However, the details of the assessment relating to unique considerations of InfiniBand make this a distinct process. Finally, results and analysis from testing the attack primitives from section 3.6 will be discussed.

4.1 Assessment Execution Process

A CVA is usually a multi-phased process. More than one site visit is often required due to budgetary constraints, scheduling considerations, customer availability, and team learning time. Networks can vary significantly in their design, usage, and collection of software applications. This information is not always communicated clearly (if at all) over e-mail or during meetings prior to the assessment team's initial visit. Therefore, interviewing network administrators, programmers, maintainers, and users is extremely beneficial for giving different perspectives on how the system functions.

4.1.1 Phase: Information Gathering.

This information gathering is the first stage in a CVA. Network and building maps are essential, but the client often has outdated or inaccurate versions of these, so the team should verify the products given to them. InfiniBand Local IDentifiers (LIDs) are dynamically assigned, so they can change, but Global Unique IDentifiers (GUIDs) should not. Hostnames, physical equipment locations, and details regarding segre-

gation elements (e.g. subnets, partitions, Virtual Lanes) should also be confirmed. InfiniBand diagnostic utilities can obtain much of the pertinent data, but their outputs are merely textual and are not very user-friendly. Verification of authorized software and users can be a longer task in enterprise networks, but HPC clusters tend to be more homogeneous. It may be worthwhile to search the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) for known vulnerabilities in any publicly available software [55]. However, this is unlikely to be fruitful given the preponderance of custom applications in the HPC world.

Checking for improper physical connections is much more challenging. Wiring diagrams may be needed, especially for larger facilities and any vehicular systems. Teams must be on the lookout for rogue equipment. Physical attacks are more challenging when fiber cables are used. Copper cables can be tapped, and splitters can be installed to mirror traffic. Optical splitters are available now as well [56]. Security testers and criminals alike have been known to unplug the cable from a networked computer or printer and connect their laptop instead, for instance. Ideally, attacks such as this would not be possible, but proper countermeasures (e.g. port security) are not always activated. Teams should also look for unlabeled or improperly marked removable media (e.g. CDs, USB drives). Organizational policy should dictate what types of media and/or which individually registered devices are allowed inside the facility. Ideally these would be logged and scanned (for contents, not necessarily viruses) when the owner enters and exits.

The goal of the information gathering phase is to identify and understand the way the organization operates so that a network baseline can be established [57]. Highly sophisticated cyber actors, so called Advanced Persistent Threats (APTs), who may have the capability to gain access to a high-performance computing (HPC) cluster, will not likely leave obvious traces behind. Instead of searching for blatantly mali-

cious files, processes and network traffic, the team should attempt to determine what normal activity is in order to differentiate it from what is abnormal. Unauthorized software such as computer games may not be an actual security problem but rather an indication of larger concerns.

Once the assessment team has established a network baseline, it can determine the key terrain (critical nodes) and document initial findings. This is not meant to be an evaluation or a condemnation of particular employees or groups. Instead, it is an opportunity to highlight security weaknesses and oversights. Not all issues can be fixed; some can only be mitigated. An important consideration in this determination is cost of any additional equipment and software and of their implementation. This is the crux of the next CVA phase, which is the secure or hardening stage. Information gathering never ceases completely, but the focus shifts to finding solutions or ways to improve the current security situation.

4.1.2 Phase: Hardening.

Solutions do not need to be expensive or complex. Examples include changing access control methods (e.g. rule-based, role-based), requiring individual user accounts, strengthening password complexity/length/age, disabling Internet Protocol over InfiniBand (IPoIB), if possible, and increasing auditing/logging levels. Stopping attacks at the network level is difficult, since security middleware devices do not exist for InfiniBand currently. However, resiliency can be improved by adding or upgrading backup systems and electrical power sources, for instance. In lieu of multi-layered defense in depth, special attention should be paid to locking down remote access. Accounts not in use should be deleted or at least disabled. Multi-factor authentication is preferred, if available, requiring use of a physical token or biometric signature(s).

Much of the hardening stage involves making policy changes. Upgrading software

and firmware is a susceptible but important event, so a regular schedule should be made for doing so. Hot patches should be applied as soon as possible, and all updates should be validated as to their source and cryptographic hash (or file size and name at the very least). Periodic validation (attestation) of critical software and firmware should be performed as well to ensure that no tampering or accidental manipulation has occurred. The assessment team should take more of a supportive role during this stage, assisting the local employees in implementing recommended improvements.

4.1.3 Phase: Monitoring.

The next phase in a CVA is the protect or monitoring stage. Network hardening is a constant process, but once a satisfactory level has been attained, it is time to evaluate whether the improvements made were effective and to clear the network of any unwanted or potentially dangerous code. The desired end state is a clean network defended by well equipped maintainers, if not dedicated cybersecurity personnel. All executables, scripts, and compressed files on every network node should be reviewed. Text and data files will probably be too numerous and large to be worth looking into, but the team can confirm that these have the correct file header. This is also known as a file's "magic number." A Windows .exe file should start with the ASCII characters "MZ" (0x4D5A in hexadecimal), for instance, so if a .txt file has that file header, something is seriously wrong. The *file* command in Linux checks these headers based on the **/etc/magic** database.

The most advanced cyber threats leave behind little or no evidence on disk. This is why monitoring running processes is so important. The key is to look for outliers, because it is very unlikely a skilled hacker would have instances of their malicious code running on a majority of the nodes. This would be less stealthy and would incrementally increase the attacker's likelihood of being detected. Live memory forensics is

very challenging but possible. It involves taking snapshots of memory, so it can only see what is currently running. Nonetheless, memory forensics can be very effective, because malicious code cannot be obfuscated in memory unlike on disk or while in transit where compression, packing, encryption, encoding, and polymorphism can all be employed.

Another necessary place to look is autorun locations such as registry keys, scheduled tasks, cron jobs, and services. These can be an exception to the standard of not leaving traces on disk, because attackers want persistence in case they lose access for whatever reason. Gaining access to a closed network requires significant resources, so losing access would be costly. Once again, comparison across all nodes can reveal oddities.

4.1.4 Phase: Follow-up.

The last stage of a CVA is performing follow-up and/or returning when a major equipment upgrade or software transition takes place. Given the relatively short duration and high cost of most CVAs, not every piece of data will be analyzed while the team is on site. This is especially true for InfiniBand systems, because these are high bandwidth networks operating around the clock. Furthermore, the client sometimes forgets to inform the team about some aspect of their network or is unaware of it at the time, so future consulting can be valuable.

Performing a Cyber Vulnerability Assessment is a rigorous task, because assessment team members must become system experts within a short period of time. It is unlikely that the team will be given months to study the network (and facilities) beforehand. Even if initial planning takes place well in advance, the team probably has other taskings occupying its schedule, so the key is to be efficient. The team should develop a few questionnaires tailored to different types of InfiniBand networks.

These questions should cover all administrative issues as well as basic details about the network and users so that the team can ask follow-up questions and discuss more advanced topics when on site. The client should be made aware that an incomplete response to the questionnaire could prolong the duration or lower the quality of the CVA.

4.2 Likely Threats to InfiniBand

Very little, if any, information has been published about confirmed attacks on InfiniBand networks. This is not entirely surprising considering the level of sophistication required to gain access to high-performance systems. Nonetheless, in order to remain proactive, it is imperative that the HPC community and security researchers consider the most probable threats to this portion of the computing market.

The first step is to consider the concept of risk. There are many formulas for determining risk, but this one is widely accepted [58]:

$$Risk = Threat\ x\ Vulnerability\ x\ Impact$$

This and similar formulas are not intended to be literal mathematical products but instead a method for breaking down risk into component factors. Impact, also called consequence, is the set of possible outcomes if a successful attack were to occur. Usually the worst case scenario is assumed. If that scenario only involved a couple more hours of work for network administrators, then the impact would not be very severe. Conversely, a network breach could lead to the loss of valuable research or trade secrets and ruin the organization's reputation, causing irreparable damage and great financial harm. Vulnerability is the level of exposure, how difficult the system's weaknesses are to exploit. It is relatively easy to enter a building through an open

window as opposed to trying to defeat a retinal and hand print scanner monitored by trained, armed guards.

The last component of risk is the threat, which can be further broken down into capability times intent. Many people may have the desire to rob a bank, but few have the ability to do so without getting caught. It is the combination of capability and intent that makes an attacker truly dangerous. Some refer to threat and vulnerability together as the likelihood of (successful) attack. Returning to the risk equation, it is safe to say to that the impact of a cyber attack on a high-performance system could be quite high, whereas the vulnerability is probably low. The unknown factor is the threat.

Previous sections covered cyber threat intelligence and the existence of APTs. FireEye, Inc. is one company that tracks the attribution and activities of known APT groups [40]. According to their list of current threats, some APTs have targeted governmental and industry sectors that contain InfiniBand users. Table 7 shows several groups who may be involved in attempts to penetrate InfiniBand networks.

Table 7. Threat Groups Possibly Targeting InfiniBand Networks

APT #	Targeted Sectors/Networks
1	IT, Scientific Research, Energy
3	High Tech
5	High-Tech Manufacturing
17	U.S. Government, IT Companies
18	High Tech
30	Air-gapped Networks
33	Energy
37	Electronics, Manufacturing, Healthcare
38	Financial Institutions

This list is not exhaustive by any means. It is merely pointing out that even the most advanced cyber actors are not usually complete mysteries. Their code and behavior often follow recognizable patterns or trends. Organizations should be cognizant of

who may be targeting their networks and should participate in cyber intelligence sharing.

4.3 Individual Attacks/Findings

Section 4.1 discussed how a notional assessment should be performed. In this section, actual tests were run in attempt to exploit weaknesses in core InfiniBand products.

4.3.1 OFED Diagnostic Tools.

These diagnostic tools are categorized in Table 8. An affirmative in the third column indicates that the authors have used or believe these tools can be used in a malicious manner. This includes possible usage as part of a larger cyber weapon. Some of these OFED tools require the user to be running as root or another local administrator account. Nonetheless, these commands can be run from any node and will affect the entire fabric just the same. The result is that every node becomes a critical one. Windows-based networks differentiate between local and domain accounts such that a local user cannot make changes on a remote node without submitting credentials acceptable to the distant end. This issue will be discussed further in Chapter 5.

A particularly susceptible tool is ibccconfig. The manual page for this command gives the following message: "WARNING −− You should understand what you are doing before using this tool. Misuse of this tool could result in a broken fabric" [59]. InfiniBand quality of service (QoS) is a robust system, but a key point is that every packet is assigned a Service Level (SL). Each port's SL to Virtual Lane (VL) mapping table determines the VL on which the packet will be sent. The InfiniBand Architecture (IBA) specifies a dual priority weighted round robin (WRR) scheme. In

Table 8. OFED Diagnostic Tools

Command(s)	Man(ual) page?	Exploitable?	Function
ibaddr	yes	no	simple address resolver
ibdev2netdev	no	no	device/port status checker
ibdiagnet, iblinkinfo, ibnetdiscover, ibnodes, ibswitches	yes	yes	fabric scanners
ibdiagpath, ibtracert	no, yes	no	route tracers
ibdump	yes	yes	traffic sniffer
ibnetsplit	yes	no	new subnet creator
ibping, ibsysstat	yes	yes	connectivity verifiers
ibportstate	yes	yes	port state querier/modifier
ibqueryerrors, perfquery	yes	no	reports port errors
ibroute, dump_fts	yes, no	yes	display forwarding table(s)
ibstat, ibstatus	yes	no	ifconfig/ipconfig equivalent
ibtopodiff	yes	no	topology difference checker
mstflint	yes	yes	firmware burner
saquery, sminfo, smpdump, smpquery, smparquery	yes (x4), no	maybe	issue subnet admin queries
ibcacheedit	yes	maybe	edit ibnetdiscover output
ibccconfig, ibccquery	yes	yes, no	congestion control

this scheme, "each VL is assigned a priority (high or low) and a weight. Within a given priority, data is transmitted from virtual lanes in approximate proportion to their assigned weights (excluding, of course, virtual lanes that have no data to be transmitted)" [25].

The ibccconfig tool can abused in several ways. Mapping all the SLs to the same VL will essentially eliminate priorities. However, if an attacker wanted to subtly (or not!!) impede certain traffic, he or she could manipulate the VL weights so that the targeted VL was allocated a lower percentage of the total bandwidth. Another option is to significantly increment the HighPriCounter so that all of the low priority lanes rarely got their turn. The effect of changing maximum transmission unit (MTU) sizes

is minimal.

Some of the other OFED diagnostic tools will be covered in later sections.

4.3.2 Malicious Firmware Installation.

How this attack is carried out depends on the location of the malicious firmware. Mellanox provides an automatic updater tool called *mlxfwmanager* for Internet-connected devices. The normal syntax is:

mlxfwmanager −−online -u -d $device

Manual firmware installation can be accomplished as follows:

mlxfwmanager_pci -i fw_file.bin

The mlxfwmanager binary could be replaced with a malicious version that would download firmware from a location of the attacker's choosing. In this way an authorized user could unknowingly install dangerous code, although this would not work if the file hash of the correct firmware was verified from its true source. Alternatively, the attacker could pull a copy of the malicious firmware through his or her covert communications channel (beacon) and install manually. Covering tracks afterwards might be an issue, although the old version of the firmware could be re-installed later.

4.3.3 Address Spoofing.

As stated previously, LIDs and GUIDs/GIDs are the Layer 2 and 3 addresses in InfiniBand. Mellanox asserts that "a node does not determine what the LID should be," because "LIDs are assigned by the subnet manager and not the node itself" [7]. Several commands can be issued to print out a host's LID(s): ibaddr, ibdiagnet, ibnodes, ibstat, ibnetdiscover, ibv_devices, and ibv_devinfo. Since these addresses are assigned dynamically, it is probable that they are stored as a file instead of burned

into the firmware (as GUIDs are). The authoritative location for LIDs appears to be at **/sys/class/infiniband/$device/ports/$port/lid**, where $device and $port are those of the HCA. (The /sys/class/infiniband/ directory was ascertained from a Mellanox script elsewhere on the file system.) For the PC tested, these variables were 'mlx5_0' and 'mlx5_1', respectively. The permissions on that file were initially read-only for everyone (-r–r–r–), and the owner was root. The following commands were run to grant full read-write access:

sudo chmod +w /sys/class/infiniband/mlx5_0/ports/1/lid

sudo chmod 777 /sys/class/infiniband/mlx5_0/ports/1/lid

The file permissions were changed, but the contents (0x2, the LID for that device/-port) were not after these attempts to edit them:

sudo gedit /sys/class/infiniband/mlx5_0/ports/1/lid

sudo echo 0x9 >> /sys/class/infiniband/mlx5_0/ports/1/lid

It is possible that this LID file may be file locked from an application or exists in firmware. As for GUIDs, [33] detailed how to alter the GUIDs in the HCA firmware. This command was used initially to blank the GUIDs:

mstflint –d $PSID -blank_guids –i /usr/share/ib_firmware/...

mstflint –d $PSID -guids fake_GUID_1 fake_GUID_2 ..

The second command was run to spoof the GUIDs. (The Parameter-Set IDentification, or PSID, is a unique identifier for the configuration of the firmware [60].) The SANS paper was written in 2012 using an old version of OFED, and since then the firmware location has changed. There is no **/usr/share/ib_firmware/** directory, and a system wide search did not reveal an obvious replacement. Nonetheless, the mstflint command still supports GUID changing options.

Another method attempted to accomplish address spoofing was through verbs programming. Infiniband connections are established via Queue Pairs (QPs). MacArthur et al. make the analogy that QPs are somewhat "comparable to the port number in TCP and UDP, and make it possible to multiplex many independent flows to the same destination HCA" [14]. The **union ibv_gid *gid** output parameter of the **ibv_query_gid()** function or the return value of the **ibv_get_device_guid()** method could be changed afterwards. Likewise, the **struct ibv_port_attr *port_attr** parameter of **ibv_query_port()** contains the LID of that port (**lid**) as well as the Subnet Manager's LID (**sm_lid**). Unfortunately, the **ibv_create_qp()** function fails to create a queue pair when given incorrect inputs, because validation occurs when resources are attempted to be used [18].

4.3.4 Network Traffic Sniffing.

Infiniband HCAs usually have two physical ports in order to maximize fabric effectiveness. If a host is connected to more than one networking device, an attacker who had command line access on one of them would most likely not be able to hear all the traffic to or from that particular host. Ethernet and InfiniBand switches alike are typically configured through an out-of-band connection to the management or console port. However, many network administrators prefer to have the ability to make changes remotely. As a result, Ethernet switches commonly allow access via telnet (port 23, unencrypted), SSH (port 22, encrypted), or even through a web browser over HTTP/HTTPS (port 80, unencrypted or port 443, encrypted) [61]. These services are not run on InfiniBand switches without enabling IPoIB.

Nonetheless, the ibdump tool does allow a user to monitor HCA traffic. The traditional tcpdump does not function since packets are crafted in hardware, not in software.

ibdump -d $device -w $filename.pcap

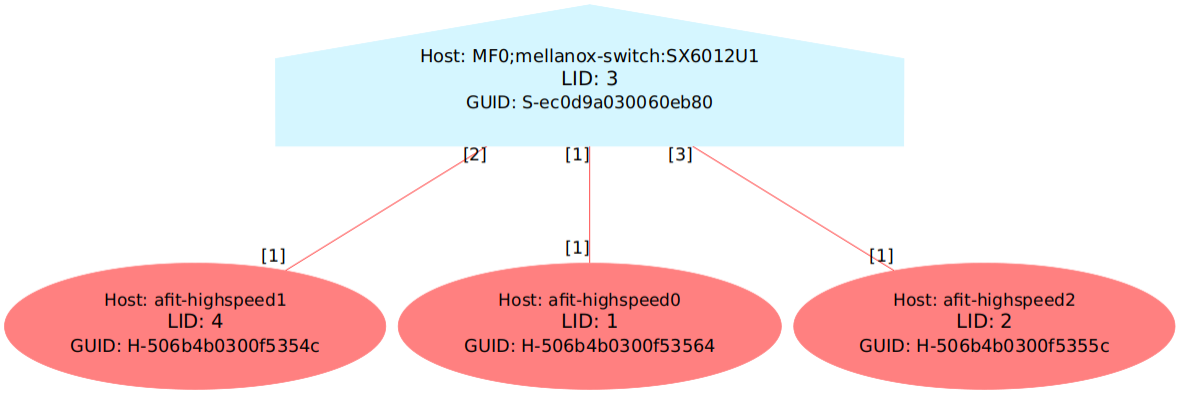
After running the above command, the packet capture (.pcap) file can be loaded into Wireshark, which has an InfiniBand plugin. The plugin can decipher all bits in the Layer 2-4 headers that are not reserved for vendor-specific fields.

4.3.5 Network Mapping.

Reconnaissance is an important step for attackers and defenders alike. Network administrators need to discover or confirm what is on their network, and intruders survey the landscape in order to find potential targets. Passive mapping, in the form of traffic sniffing, is not a very easy option in InfiniBand, since, to be effective, it requires putting a network tap on a switch, installing a hardware splitter, or altering forwarding tables to mirror traffic. On the other hand, several of the OFED diagnostic tools were designed to perform discovery, and use of them should not raise any alarms. However, there are two minor limitations with tools such as ibdiagnet, iblinkinfo, and ibnetdiscover. First of all, the outputs are all text, which is not very user friendly or intuitive. Secondly, they list devices one at a time, not necessarily in the order in which they are connected to each other.

There is no graphical mapping tool for InfiniBand. The open source ZenMap python code could be adapted to InfiniBand, but this may be overkill considering the lack of hosted services and virtual ports in InfiniBand. Furthermore, the OFED tools provide all the necessary network information without needing to allow users the ability to fully customize the dozens of parameter options that make text parsing far more complicated. A relatively simple mapping program based on ibnetdiscover was found on GitHub and was enhanced in order to display hostnames, LID and port details in addition to the GUIDs [62]. Figure 8 shows the output of the modified program:

Figure 8. AFIT InfiniBand Network



This visual representation aids in human understanding of a network's layout. It also hastens the assessment team's ability to identify potential issues such as traffic bottlenecks/load balancing concerns and to make recommendations for improved physical or logical network segmentation.

4.3.6 Subnet Manager Hijacking.

The method chosen to replace the master Subnet Manager (SM) is with the ibportstate tool. The ibstat command informs the user of the SM's LID, so an attacker could then run a tool such as ibnetdiscover to determine where the SM's host is connected to a switch or switches. Disabling the corresponding switch port(s) will shut off all communication to that device, causing it to fail during polling:

ibportstate 3 1 disable

(This command tells LID 3 to disable its first port.) The malicious version of OpenSM, waiting in slave mode, will take over as master assuming there are no other backups. This process appears to require at least 10 seconds, but afterwards the switch port(s) can be enabled to restore normal traffic flow. This attack has some limitations, including if the SM is being run on a switch. Disabling all of a switch's ports may be irreversible without physical access.

4.3.7 Out-of-Band Switch Access.

This attack was unsuccessful but not for an anticipated reason. Testing revealed that the switch's console port was shut off or disabled about 10 minutes after switch power up. Attempts to establish a serial connection after that time went unanswered. This could very well be a Mellanox power saving or security feature. If so, it appears to be very effective.

Regardless, this is still a legitimate attack vector. Syntax for the MLNX-OS switch operating system is very similar to that of Cisco IOS. This makes the transition easy for people with network administration experience.

4.3.8 Falsified Memory Keys.

This attack vector was not attempted, because no way to manipulate packet fields has been discovered thus far. A secondary reason was that the desired memory keys were not able to be captured by an unintended observer. The keys are sent in plaintext and can be seen in Wireshark after the ibdump tool is run [63]. However, unless the malicious user is on one of the endpoint nodes or the traffic is mirrored to the attacker's location, he or she will not be able to view the necessary packets. If packet manipulation were possible, then the attacker could attempt to brute force or make educated guesses on active memory keys. However, memory keys are at least 4 bytes in length, so there are billions ($2^{32} > 10^9$) of possible combinations.

4.3.9 Denial of Service Attacks.

There are other ways to produce a denial of service, but this thesis only investigated use of the ibping command invoking the flood option:

ibping -f 4

The above command sends echo request packets to and receives echo reply packets from LID 4, the victim computer, back to back without delay. During testing, a single instance sent 265k packets in 5 seconds, or 53k per second. Running three terminals simultaneously produced 1.6M packets in 8.5 seconds, or 217k per second when the ends of the capture were removed to account for starting and stopping the commands manually. Four terminals generated 290k packets per second adjusted. Lastly, five terminals each were run on two different hosts for a total of ten simultaneous ibping instances. In this case, 4.5M packets were sent in 9.8 seconds, or 582k per second adjusted. The volume seemed to scale fairly linearly according to limited sampling. No major packet loss or other harmful effects were observed, but it is possible that hundreds of instances could create a distributed denial of service (DDoS).

V. Proposed Tools & Mitigations

This chapter makes recommendations for cyber software and hardware solutions that could be adapted for the InfiniBand Architecture (IBA). Possible security flaws in existing tools are also explored.

5.1 InfiniBand Cyber Tools

There is an argument to be made that InfiniBand is safer without cybersecurity testing software. In Ethernet networks, many of the same "hacking tools" are used by both attackers and defenders. As it stands with InfiniBand, there is some degree of security by obscurity taking place. Nonetheless, dedicated cyber actors will inevitably create custom InfiniBand tools, especially if none exist already. The benefit of having industry standard tools is that they are known and can be tailored so as to not have malicious or overly powerful options. Nmap, for example, offers parameters for address spoofing and performing zombie scans [64]. These options appear to have little or no legitimate uses.

The graphical network mapping tool shown in Chapter 4 is a nice prototype, but there is much room for improvement. Other fields may be useful to administrators and defenders such as port states, chassis information, logged on user(s), and more. This idea taken to the extreme would be a near real-time monitoring program similar to a Human-Machine Interface (HMI) used in the Industrial Control System (ICS) world. The difference is that this tool should not allow remote changes, lest it become too powerful.

The next proposed tool would be for file integrity checking. Currently, if an attacker replaced the binary for OpenSM or for an OFED diagnostic tool with a malicious executable (with similar functionality or not), it might be difficult to de-

termine this had happened without manual verification such as taking a file hash of the binary and comparing it to a known good copy. That process is inefficient, but a local (or network) script could do it periodically. Ideally, a rogue Subnet Manager would not be allowed to participate in polling so that it would never become the master. A couple minor hurdles with this approach would be performing software upgrades and if different versions of the same programs were allowed on the network simultaneously.

The last proposed tool deals with a fundamental shortcoming of Linux and InfiniBand, that being the lack of network or domain user accounts. It is true that Lightweight Directory Access Protocol (LDAP) solutions such as OpenLDAP exist, but these are not widely used [65]. The problem is that a user who gains root access on a host, either directly through the root account or inclusion in the **/etc/sudoers** file, would have nearly unrestricted access on the network. This might be acceptable if the sudoer and **/etc/group** files were tightly managed and always up-to-date, but this is not a trivial task. Non-administrative users may require temporary root access to install software or to run certain diagnostic tools or other programs so that their scripts and applications run correctly. There are additional ways to restrict network access such as creating more subnets and partitions, but these mechanisms shrink the effective size of the network instead of limiting what users can do on it. A purely InfiniBand variation of OpenLDAP could be very useful to network administrators.

5.2 Defense in Depth

As previously mentioned, implementing Defense in Depth on InfiniBand systems is difficult due the flat hierarchy of switched fabrics. In addition, the lack of virtual port usage means that a traditional (internal) firewall would not be effective. (Having a firewall at the network perimeter is as critical as always.) Only an application layer

firewall would be of any use at all inside an InfiniBand fabric, and even then it would increase network latency significantly, assuming such a device could actually keep up with the traffic volume. Adding security features to a system comes at a cost in terms of decreased performance, more electrical power, increased design complexity, and/or additional monetary or manpower expenses. Sometimes this trade off is favorable, but in this case network owners would not (and should not) tolerate such delays considering that throughput and latency are two of the primary reasons to purchase and employ InfiniBand equipment.

The idea of instituting InfiniBand traffic monitoring is still worth considering, however. Traffic could be mirrored to a secondary location, either through physical splitters, network taps, and/or Switched Port Analyzer (SPAN) ports. After that, the question is what to do with the data. The passive option would be to store it for future review and/or analytics. This is similar to the function of a data historian in ICS networks. (It should be noted that the duration of storage might be very limited unless Petabyte storage is available or filters are able to significantly reduce redundant information.) The active option would be something akin to an Intrusion Detection/Prevention System (IDS/IPS), although those devices rely on user-created and community signatures. It is possible that signatures could be developed, but defenders would have to know what potentially malicious traffic looks like. Furthermore, high-performance networks often employ custom software, so the organization might not get much external help in building up this device.

The most viable traffic monitoring alternative appears to be an IDS device run by an Artificial Intelligence (AI) program. An AI could learn normal traffic patterns and could alert a human cyber defender of abnormal activities. There is a chance for high false positive rates, but high-performance networks tend to be more homogeneous, and the program could be tuned over time to decrease the amount of false alerts.

5.3 Modifications to Existing Products

Beyond inventing new security tools and devices, it is important to consider how security aspects of current products might be improved. A positive step that Oracle recently took was introducing signed firmware [66]. This is a more secure and faster way of validating the authenticity of firmware. Manual verification is not foolproof, and posting the cryptographic hash of the real firmware on a public website could give an Advanced Persistent Threat (APT) time to find a hash collision and apparent legitimacy afterwards. Other InfiniBand manufacturers should consider mandating signed firmware as well.

A major security concern in InfiniBand systems will always be the Subnet Manager, because it is a powerful network entity. The fact that OpenSM is open source exacerbates this concern, because anyone with the proper expertise could study the code to learn how it is implemented and to identify any weaknesses or critical dependencies it might have. This thesis has not exhaustively studied OpenSM, but there does not appear to be way to disallow new hosts from joining the network. In Ethernet, enabling port security and disabling Dynamic Host Configuration Protocol (DHCP) are two methods to prevent an unauthorized device on the network. A malicious insider should not be able to receive a dynamically assigned address by connecting their own device to a switch or by removing the Small Form-factor Pluggable (SFP) optical transceiver from an existing node and using that cable instead.

Another potential Subnet Manager issue is that Mellanox's switch operating system, MLNX-OS, has a command called **ib sm ignore-other-sm** that "ignores all the rules governing SM elections and attempts to manage the fabric" [67]. That sounds like a dangerous command that only exists for user convenience. Given console access to other nodes, a network administrator should be able to stop any unwanted Subnet Manager instances. This discussion makes an inquisitive mind wonder how this com-

mand works and whether or not it usually succeeds. Mellanox would probably not use a destructive technique such as the one used in this thesis, but the description for the **show ib sm sm-priority** command says that if two or more active SMs have the same highest priority (between 0 and 15), the one with the lowest port Global Unique IDentifier (GUID) will manage the fabric.

Armed with this knowledge, an APT could manufacture counterfeit Channel Adapters with extremely low GUIDs burned into them. These network cards could have a logic bomb or hardware backdoor that occasionally attempts to beacon home through the Internet or that listens for covert communications channels (i.e. port knocking). They might also allow an informed user to craft malformed packets via a custom driver, as well as having an advantage in Subnet Manager elections. This amount of speculation may frustrate some readers, but it is exactly the creativity needed to anticipate the most grave threats.

A more abstract idea for improving the security of fabric management is moving to a Software Defined Networking (SDN) approach. SDN separates a network's control and data planes, so management occurs out-of-band [68]. Currently, Subnet Management Packets are sent in-band on Virtual Lane (VL) 15. Another benefit of SDN is rapid network reconfigurability. Part of the initiative for developing SDN was the fact that updating the rules and configuration files on traditional networking equipment is very time- and skill-intensive. This is not suitable for the needs of modern, adaptable networks such as clouds. Then again, the whole concept of SDN is somewhat at odds with InfiniBand, which shifts a lot of functionality from software to hardware. Furthermore, high-performance networks are often designed and tuned to perform a specific mission very well. Despite this, Mellanox realized the need for SDN capabilities. The company's Unified Fabric Manager (UFM®) claims to have SDN features but is proprietary software, so it was not tested in this thesis [69].

Another way to improve network security and adaptability is through compatibility with virtualization technologies such as Linux containers. Mellanox has started this process as well through its Single Root I/O Virtualization (SR-IOV) and Docker support [21]. Virtualization is the polar opposite of shared memory, in that virtual machines (VMs) and containers do not have any external dependencies and usually operate as self-contained units. Virtualization also improves resiliency, because a compromised VM or container can easily be returned to a known good snapshot or state. Common container usage involves starting up a new instance when one is needed and tearing it down upon task completion or when it is no longer necessary. The primary security concern would be if the repository were to get infected.

The proposals in this chapter are not meant to be an exhaustive list. The intent is to initiate discussions and innovation regarding InfiniBand and high-performance computing (HPC) security. If multiple vulnerabilities are found and published, they should probably be included in the National Vulnerability Database (NVD) using the Common Vulnerability Scoring System (CVSS) [55].

VI. Conclusion & Future Work

Although the InfiniBand Trade Association did not make cybersecurity its top priority when it designed and developed the InfiniBand Architecture, it is evident that many features of InfiniBand are inherently resistant to tampering and attack. Hardware packet crafting, predetermined routing, and the redundant pathways of a switched fabric topology contribute to the very low network latency and high availability that the high-performance computing (HPC) industry needs. InfiniBand does somewhat rely on external defense mechanisms such as firewalls and other forms of network segregation. However, HPC clusters were intended to be located in data warehouses behind Demilitarized Zones, not providing services as Internet facing servers. This is part of the justification as to why the association decided not to incorporate packet encryption or provide support to protocols running over TCP/IP.

Nonetheless, some minor security upgrades could make the InfiniBand Architecture more difficult to exploit without significantly degrading network performance. The Subnet Manager is such a critical component, that it should have some protections in place should the master fail. A possible solution could be file verification, whereby a node could not become the master if its OpenSM file hashes did not match those of the other standby nodes. Also, simple denial of service attacks such as disabling switch ports from any host or invoking massive ping floods should not be allowed, even if the user is operating with elevated system privileges. In addition to improving existing software, new cyber tools and security devices could be created to bolster InfiniBand defenses. A graphical mapping tool, network account management software, and an InfiniBand variant of an Intrusion Detection System are a few examples of these.

This thesis presented a framework for evaluating the cybersecurity posture of an InfiniBand network. The most important step is performing risk analysis and

considering serious threats to the system. Before conducting a Cyber Vulnerability Assessment (CVA), however, an assessment team must be assembled with the proper skillsets and equipment. A plan and scope for the assessment must be developed with guidance from the client. Once on site, the team should work hand-in-hand with local system owners, maintainers, and users to learn about the network and establish a baseline. At this point, the team can assist personnel with implementing any recommended changes and upgrades. The network must be cleared of unwanted or potentially dangerous code, and local maintainers must be confident about their ability to defend the network after the team's departure.

The big picture concepts in this framework are not different from how CVAs are conducted on other networks. What makes this research contribution unique is in its distinctions between HPC and traditional Information Technology (IT) networks, in the terminology, usage of, and threats to both, as well as between the InfiniBand and Internet Protocol (IP) stacks. This thesis also performed a holistic security evaluation of InfiniBand, including its protocol, hardware, supporting software, and network architecture. The focus was on a generic network and core software, but current and future InfiniBand use cases were explored.

There are many possibilities for follow-on research of this topic. Any of the proposals from Chapter 5 might make a good candidate, but some of these would be suitable for a research group instead of a single individual. A student with more extensive hardware knowledge might consider attempting to develop a custom Channel Adapter driver or firmware. A more software-intensive selection could involve construction of cyber tool, possibly for parsing, storing, and cataloging InfiniBand network traffic.

Appendix A. Graphical Mapping Tool

```
#!/usr/bin/env python
#
# Copyright (C) 2016 Vangelis Tasoulas <vangelis@tasoulas.net>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/>.

# **This version has been updated by Daryl Schmitt and Lucas Mireles.**

from __future__ import print_function
import os
import sys
import re
import argparse
import logging
import time
import pygraphviz as pgv
import xml.etree.ElementTree as et
import xml.dom.minidom as md
from collections import OrderedDict

__all__ = [
    'quick_regexp', 'print_',
    'hex_to_rgb', 'LOG'
]

PROGRAM_NAME = 'InfiniBand-Graphviz-ualization'
VERSION = '0.2.1'
AUTHOR = 'Vangelis Tasoulas, Daryl Schmitt, and Lucas Mireles'

LOG = logging.getLogger('default.' + __name__)

##################################################
################ HELPER FUNCTIONS ################
##################################################

#-----------------------------------------------------------------------
def error_and_exit(message):
    """
    Prints the "message" and exits with status 1
    """
    print("\nFATAL ERROR:\n" + message + "\n")
```

```
    exit(1)

#-----------------------------------------------------------------------
def print_(value_to_be_printed, print_indent=0, spaces_per_indent=4, endl="\n"):
    """
    This function, among anything else, it will print dictionaries (even nested ones) in a good looking way

    # value_to_be_printed: The only needed argument and it is the
                           text/number/dictionary to be printed
    # print_indent: indentation for the printed text (it is used for
                    nice looking dictionary prints) (default is 0)
    # spaces_per_indent: Defines the number of spaces per indent (default is 4)
    # endl: Defines the end of line character (default is \n)

    More info here:
    http://stackoverflow.com/questions/19473085/create-a-nested-dictionary-for-a-word-python?answertab=active#tab-top
    """

    if(isinstance(value_to_be_printed, dict)):
        for key, value in value_to_be_printed.iteritems():
            if(isinstance(value, dict)):
                print_('{0}{1!r}:'.format(print_indent * spaces_per_indent * ' ', key))
                print_(value, print_indent + 1)
            else:
                print_('{0}{1!r}: {2}'.format(print_indent * spaces_per_indent * ' ', key, value))
    else:
        string = ('{0}{1}{2}'.format(print_indent * spaces_per_indent * ' ', value_to_be_printed, endl))
        sys.stdout.write(string)

#-----------------------------------------------------------------------
def hex_to_rgb(value):
    value = value.lstrip('#')
    lv = len(value)
    return tuple(int(value[i:i + lv // 3], 16) for i in range(0, lv, lv // 3))

#-----------------------------------------------------------------------
class quick_regexp(object):
    """
    Quick regular expression class, which can be used directly in if() statements in a perl-like fashion.

    #### Sample code ####
    r = quick_regexp()
    if(r.search('pattern (test) (123)', string)):
        print(r.groups[0]) # Prints 'test'
        print(r.groups[1]) # Prints '123'
    """
    def __init__(self):
        self.groups = None
        self.matched = False

```

```
    def search(self, pattern, string, flags=0):
        match = re.search(pattern, string, flags)
        if match:
            self.matched = True
            if(match.groups()):
                self.groups = re.search(pattern, string, flags).groups()
            else:
                self.groups = True
        else:
            self.matched = False
            self.groups = None

        return self.matched

#--------------------------------------------------------------------

##########################################
###### Configure logging behavior ######
##########################################
# No need to change anything here

def _configureLogging(loglevel):
    """
    Configures the default logger.

    If the log level is set to NOTSET (0), the
    logging is disabled

    # More info here: https://docs.python.org/2/howto/logging.html
    """
    numeric_log_level = getattr(logging, loglevel.upper(), None)
    try:
        if not isinstance(numeric_log_level, int):
            raise ValueError()
    except ValueError:
        error_and_exit('Invalid log level: %s\n'
        '\tLog level must be set to one of the following:\n'
        '\t    CRITICAL <- Least verbose\n'
        '\t    ERROR\n'
        '\t    WARNING\n'
        '\t    INFO\n'
        '\t    DEBUG    <- Most verbose' % loglevel)

    defaultLogger = logging.getLogger('default')

    # If numeric_log_level == 0 (NOTSET), disable logging.
    if(not numeric_log_level):
        numeric_log_level = 1000
    defaultLogger.setLevel(numeric_log_level)

    logFormatter = logging.Formatter()

    defaultHandler = logging.StreamHandler()
    defaultHandler.setFormatter(logFormatter)
```

```
158     defaultLogger.addHandler(defaultHandler)

160 ############################################################
    ###### Add command line options in this function ######
162 ############################################################
    # Add the user defined command line arguments in this function
164
    def _command_Line_Options():
166     """
        Define the accepted command line arguments in this function
168
        Read the documentation of argparse for more advanced command line
170     argument parsing examples
        http://docs.python.org/2/library/argparse.html
172     """

174     parser = argparse.ArgumentParser(description=PROGRAM_NAME + " version " + VERSION)

176     parser.add_argument("-v", "--version",
                            action="version", default=argparse.SUPPRESS,
178                         version=VERSION,
                            help="show program's version number and exit")
180
        loggingGroupOpts = parser.add_argument_group('Logging Options', 'List of optional logging
        options')
182     loggingGroupOpts.add_argument("-q", "--quiet",
                                      action="store_true",
184                                   default=False,
                                      dest="isQuiet",
186                                   help="Disable logging in the console. Nothing will be printed.")
        loggingGroupOpts.add_argument("-l", "--loglevel",
188                                   action="store",
                                      default="INFO",
190                                   dest="loglevel",
                                      metavar="LOG_LEVEL",
192                                   help="LOG_LEVEL might be set to: CRITICAL, ERROR, WARNING, INFO,
        DEBUG. (Default: INFO)")

194     parser.add_argument("-f", "--topology-file",
                            action="store",
196                         required=True,
                            dest="topo_file",
198                         metavar="TOPOLOGY",
                            help="Topology file to load the data from.")
200     parser.add_argument("-d", "--detailed-topo",
                            action="store_true",
202                         default=False,
                            dest="detailed_topo",
204                         help="If the user needs a detailed topology, use 'record' shapes and draw
        individual ports on each shape. Note that a detailed topology might be good for visualization,
        but it is not supported by Gephi.\n"
                                 "Default: False (use 'rectangle' shapes with multiple connections).")
206     parser.add_argument("-c", "--use-clusters",
```

```
                        action="store_true",
208                     default=False,
                        dest="use_clusters",
210                     help="If enabled, HCAs (nodes) connected on the same switches are grouped
    in the same cluster.\n"
                             "Unfortunately only 'dot' supports clustering at the moment, so
    clustering is disabled by default.\n"
212                          "Default: False")
    parser.add_argument("-o", "--optimized-for-black-bg",
214                     action="store_true",
                        default=False,
216                     dest="optimize_black_bg",
                        help="Gephi can make really good looking graphs on black background. If
    you are about to plot a final graph on black background then use this option to optimize the
    colors for it.\n"
218                     "Default: False")
    parser.add_argument("-r", "--render-file",
220                     action="store_true",
                        default=False,
222                     dest="render_file",
                        help="If enabled, the output will be rendered with the 'neato' layout, and
     saved in a PDF file.\n"
224                     "Default: False")
    parser.add_argument("-e", "--export-gexf",
226                     action="store_true",
                        default=False,
228                     dest="export_gexf",
                        help="Support for DOT files in Gephi is really bad. Use this option to
    export a gexf file if you want to play with the graph in Gephi.\n"
230                     "Default: False")

232     opts = parser.parse_args()

234     if(opts.isQuiet):
            opts.loglevel = "NOTSET"
236
        return opts
238
#########################################################
240 ################ WRITE MAIN PROGRAM #################
#########################################################
242
if __name__ == '__main__':
244     """
        Write the main program here
246     """
        # Parse the command line options
248     options = _command_Line_Options()
        # Configure logging
250     _configureLogging(options.loglevel)

252     LOG.info("{} v{} is running...\n".format(PROGRAM_NAME, VERSION))

254     ########################################
```

```
    #LOG.critical("CRITICAL messages are printed")
    #LOG.error("ERROR messages are printed")
    #LOG.warning("WARNING messages are printed")
    #LOG.info("INFO message are printed")
    #LOG.debug("DEBUG messages are printed")

    # If the user needs a detailed topology, use "record" shapes and draw individual ports on each
      shape.
    # Otherwise, use "rectangle" shapes with multiple connections.
    #
    # Unfortunately, a detailed topology is not supported by Gephi.
    # Disabling by default.
    detailed = options.detailed_topo

    # If enabled, HCAs (nodes) connected on the same switches are grouped in the same cluster.
    # Unfortunately only 'dot' supports clustering at the moment, so I disable it by default.
    useClusters = options.use_clusters

    exportGexf = options.export_gexf

    # Read the topology and build the graph in an OrderedDict
    topology_file = options.topo_file
    topology = OrderedDict()
    num_of_switches = 0
    num_of_hcas = 0
    current_node = ""
    with open(topology_file, mode='r', buffering=1) as f:
        isinstance(f, file)
        for line in f:
            line = line.strip()
            isinstance(line, str)
            if line:
                #added DWS 8 Jan 19
                lid_idx = line.find('lid') # Find the last occurrence
                if(lid_idx > 0):
                    lid = int(line.split('lid')[1].strip().split(' ')[0])

                r = quick_regexp()
                # This regexp will read the name of nodes and the number of ports (Switches or
    HCAs)
                if (r.search("^(\w+)\s+(\d+)\s+\"(.+?)\"\s+#\s+\"(.+?)\"", line)):
                    #print_(r.groups)
                    current_node = r.groups[2]
                    #print(current_node)
                    topology[current_node] = OrderedDict()
                    # Added by DWS - 8 Jan 19
                    topology[current_node]['hostname'] = r.groups[3].replace('/', '\\').split(' ')
    [0]
                    topology[current_node]['number_of_ports'] = int(r.groups[1])

                    if r.groups[0] == 'Switch':
                        topology[current_node]['node_type'] = 'switch'
                        # Added by DWS - 8 Jan 19
                        topology[current_node]['lid'] = lid
```

```
306                         num_of_switches = num_of_switches + 1
                        else:
308                         topology[current_node]['node_type'] = 'hca'
                            # Added by DWS - 8 Jan 19
310                         topology[current_node]['lid'] = 0
                            num_of_hcas = num_of_hcas + 1
312
                    # This regexp will read the port lines from a dump
314                 if (r.search("^\[(\d+)\].*?\"(.+?)\"\[(\d+)\]", line)):
                        # Added by DWS - 8 Jan 19 .. this only works if HCA is just using one port
316                     if (topology[current_node]['node_type'] == 'hca') and (topology[current_node][
    'number_of_ports'] == 1):
                            topology[current_node]['lid'] = lid
318                     local_port = int(r.groups[0])
                        connected_to_remote_host = r.groups[1]
320                     connected_to_remote_port = int(r.groups[2])
                        topology[current_node][local_port] = {connected_to_remote_host:
    connected_to_remote_port}
322
    #if(len(topology) > 1000 and detailed):
324     #LOG.warn("The provided network contains {} nodes (too many) and you have chosen to draw a
      detailed topology.\n"
                #"If the drawing state takes much longer than anticipated, please run the program
      again with the detailed topology switch turned off.".format(len(topology)))
326
    G = pgv.AGraph(name = "Fat-tree", strict = False)
328
    ##################
330 # Graphviz attribute list!
    #    www.graphviz.org/doc/info/attrs.html
332 ##################

334 # Graph attributes
    G.graph_attr['rankdir'] = 'TB' # TB, BT, LR, RL
336 G.graph_attr['ranksep'] = 1.0
    #G.graph_attr['nodesep'] = 0.0
338 # Type of the edges: (line, false), (spline, true), (none, ""), curved, polyline, ortho
    G.graph_attr['splines'] = 'line'
340 # Do not allow nodes to overlap. Scale makes the compilation very fast but spreads the graph!
    G.graph_attr['overlap'] = 'scale'
342 # The size of the output image in inches. Use a multiple of 7.75 and 10.25
    # http://stackoverflow.com/questions/3489451/how-to-set-the-width-and-heigth-of-the-ouput-
     image-in-pygraphviz
344 G.graph_attr['size'] = "{},{}!".format(7.75 * 12, 10.25 * 12)
    if options.optimize_black_bg:
346     G.graph_attr['bgcolor'] = '#000000'

348 # Colors
    if options.optimize_black_bg:
350     HCA_Color = '#cccccc'
        HCA_Edge_Color = '#ff0000'
352     Switch_Color = '#ffffff'
        Switch_Edge_Color = '#a0a0a0'
354 else:
```

```
        HCA_Color = '#ff8080'
        HCA_Edge_Color = '#ff0000'
        Switch_Color = '#d5f6ff'
        Switch_Edge_Color = '#000000'

    Cluster_Color = "yellow"

    # Shapes      # Update by LEM - 1/9/19
    HCA_Shape = "ellipse"
    Switch_Shape = "house"

    G.add_nodes_from(topology)

    if exportGexf:
        #gephi_edge_id = G.number_of_nodes()
        gephi_edge_id = 0
        root_node = et.Element('gexf')
        root_node.attrib['xmlns'] = "http://www.gexf.net/1.3"
        root_node.attrib['version'] = "1.3"
        root_node.attrib['xmlns:viz'] = "http://www.gexf.net/1.3/viz"
        root_node.attrib['xmlns:xsi'] = "http://www.w3.org/2001/XMLSchema-instance"
        root_node.attrib['xsi:schemaLocation'] = "http://www.gexf.net/1.3 http://www.gexf.net/1.3/gexf.xsd"
        meta_node = et.SubElement(root_node, 'meta', attrib = {'lastmodifieddate': time.strftime("%Y-%m-%d")})
        creator_node = et.SubElement(meta_node, 'creator')
        creator_node.text = PROGRAM_NAME
        description_node = et.SubElement(meta_node, 'description')
        description_node.text = "Graph generated from file '{}'".format(os.path.realpath(topology_file))
        graph_node = et.SubElement(root_node, 'graph')
        graph_node.attrib['defaultedgetype'] = "undirected"
        graph_node.attrib['mode'] = "static"
        nodes_node = et.SubElement(graph_node, 'nodes')
        edges_node = et.SubElement(graph_node, 'edges')

    # Cluster id for the subgraphs
    if useClusters:
        global_cluster_id = 0

    # Add ports and edges
    for node in G.nodes():
#Label      # Update by LEM - 1/9/19
        label = "Host: {0}\nLID: {1}\nGUID: {2}".format(topology[node.name]['hostname'],topology[node.name]['lid'],node.name)
        node.attr['label'] = label

# Node Attributes # Update by LEM - 1/10/19
node.attr['style'] = 'filled'
node.attr['margin'] = 0.2
node.attr['fontsize'] = 30
node.attr['fontcolor'] = 'black'

        if useClusters:
```

```
            Subgraph_For_Current_Switch = None
406
        numPorts = topology[node.name]['number_of_ports']
408         if topology[node.name]['node_type'] == 'switch':
      node.attr['shape'] = Switch_Shape # Update by LEM - 1/9/19
410             node.attr['fillcolor'] = Switch_Color
            node.attr['color'] = Switch_Color
412         elif topology[node.name]['node_type'] == 'hca':
      node.attr['shape'] = HCA_Shape  # Update by LEM - 1/9/19
414             node.attr['fillcolor'] = HCA_Color
            node.attr['color'] = HCA_Color
416
        if exportGexf:
418             node_node = et.SubElement(nodes_node, 'node', attrib = {'id': node.name, 'label': node
    .name})
            r, g, b = hex_to_rgb(node.attr['color'])
420             et.SubElement(node_node, 'viz:color', attrib = {'r': str(r), 'g': str(g), 'b': str(b),
     'a': "0.0"})

422         if(numPorts > 0):
            for port in range(1, numPorts + 1):
424                 if(detailed):
                    if port == 1:
426                         separator = ""
                    else:
428                         separator = "|"
                    label = "{}|<{}> {}".format(label, port, port)
430
                if port in topology[node.name].keys():
432                     remote_port = topology[node.name][port].values()[0]
                    remote_host = topology[node.name][port].keys()[0]
434                     try:
                        # Check if there is already a link established...
436                         # If not, the 'get_edge' command will raise an exception and a new edge
    will be added
                        G.get_edge(remote_host, node.name, key="{}-{}".format(port, remote_port))
438                     except:
                        if(detailed):
440                             G.add_edge(node.name, remote_host, key="{}-{}".format(remote_port,
    port), headport = remote_port, tailport = port)
                        else:
442                             G.add_edge(node.name, remote_host, key="{}-{}".format(remote_port,
    port))

444                         edge = G.get_edge(node.name, remote_host, key="{}-{}".format(remote_port,
    port))

446      # Edge Attributes # Update by LEM - 1/10/19
      edge.attr['penwidth'] = 1
448      edge.attr['fontsize'] = 30
      edge.attr['headlabel'] = "[{0}]".format(remote_port)  #port num that is leaving HCA
450      edge.attr['taillabel'] = "[{0}]".format(port)   #port num that is entering switch
      edge.attr['minlen'] = 2
452
```

```
                        if topology[remote_host]['node_type'] == 'hca':
454                         edge.attr['color'] = HCA_Edge_Color
                            if useClusters:
456                             if (Subgraph_For_Current_Switch is None):
                                    Subgraph_For_Current_Switch = G.subgraph(remote_host, 'cluster
    {}'.format(global_cluster_id))
458                                 Subgraph_For_Current_Switch.graph_attr['fillcolor'] =
    Cluster_Color
                                    Subgraph_For_Current_Switch.graph_attr['style'] = 'filled'
460                                 global_cluster_id = global_cluster_id + 1
                                else:
462                                 Subgraph_For_Current_Switch.add_node(remote_host)
                        else: # Else the node connects two switches.
464                         edge.attr['color'] = Switch_Edge_Color

466                     if exportGexf:
                            gephi_edge_id += 1
468                         edge_node = et.SubElement(edges_node, 'edge',
                                                      attrib = {'id': str(gephi_edge_id),
470                                                             'source': node.name,
                                                               'target': remote_host,
472                                                            'label': "{} -- {}".format(node.
    name, remote_host)})
                            r, g, b = hex_to_rgb(edge.attr['color'])
474                         et.SubElement(edge_node, 'viz:color', attrib = {'r': str(r), 'g': str(
    g), 'b': str(b)})

476 #print(G.string())
    LOG.info("Total number of nodes: {}\n"
478          "Total number of Switches: {}\n"
             "Total number of HCAs: {}\n"
480          "Total number of Edges: {}".format(num_of_hcas + num_of_switches, num_of_switches,
    num_of_hcas, len(G.edges())))

482 output_filename = os.path.basename(topology_file)
    dot_filename = "{}.dot".format(output_filename)
484 G.write(dot_filename)

486 LOG.info("The dot file has been saved in the file '{0}'.".format(dot_filename))

488 if options.render_file:
        # twopi, gvcolor, wc, ccomps, tred, sccmap, fdp, circo, neato, acyclic, nop, gvpr, dot,
    sfdp
490         drawing_args=''
            G.layout(prog='neato', args=drawing_args)
492         # 'canon', 'cmap', 'cmapx', 'cmapx_np', 'dia', 'dot',
            # 'fig', 'gd', 'gd2', 'gif', 'hpgl', 'imap', 'imap_np',
494         # 'ismap', 'jpe', 'jpeg', 'jpg', 'mif', 'mp', 'pcl', 'pdf',
            # 'pic', 'plain', 'plain-ext', 'png', 'ps', 'ps2', 'svg',
496         # 'svgz', 'vml', 'vmlz', 'vrml', 'vtx', 'wbmp', 'xdot', 'xlib'
            render_filename = "{}.pdf".format(output_filename)
498         G.draw(render_filename)

500         LOG.info("The rendered file has been saved in the file '{0}'.".format(render_filename))
```

```
502         # Use command line to debug long executing drawings:
            #dot -Tpdf -Gnslimit=1000 k-18-n-3.topology.dot -v -O
504
        if exportGexf:
506         gexf_filename = "{}.gexf".format(output_filename)
            xml_ugly = et.tostring(root_node, encoding='UTF-8', method='xml')
508         xml = md.parseString(xml_ugly)
            with open(gexf_filename, "w") as f:
510             f.write(xml.toprettyxml(indent="  ", newl="\n", encoding='UTF-8'))

512         LOG.info("The gexf file has been saved in the file '{0}'.".format(gexf_filename))
            LOG.info("\nIf you want to generate a beautiful graph with Gephi, follow the following
        steps:\n"
514                  "  1. Load the file '{0}' in Gephi.\n"
                     "  2. Go to the 'Overview' tab and choose a placement layout (I like the results
        of 'ForceAtlas 2' algorithm).\n"
516                  "  3. Tune as needed and run the layout until you are satisfied with the
        placement and press stop.\n"
                     "  4. Go to the 'Preview' tab and press the 'Refresh' button.\n"
518                  "  5. Under the 'Edges' group, untick the 'Curved' tick box and change the 'Color
        ' of the edges from 'mixed' to 'original'\n"
                     "  6. Choose a 'black' background if you used the '--optimized-for-black-bg (-o)'
         option, and press a final refresh.\n"
520                  "  7. Once you are satisfied with the result, press the 'Export SVG/PDF/PNG'
        button to save the layout.\n".format(gexf_filename))
```

Bibliography

1. G. Cleary, M. Corpin, O. Cox, H. Lau, B. Nahorney, D. O'Brien, B. O'Gorman, J.-P. Power, S. Wallace, P. Wood, and C. Wueest, "Internet Security Threat Report (ISTR) Volume 23," Symantec Corporation, Mountain View, CA, Tech. Rep., 2018.

2. G. Post and A. Kagan, "The use and effectiveness of anti-virus software," *Computers and Security*, vol. 17, no. 7, pp. 589–599, 1998.

3. S. Anthony, "It might be time to stop using antivirus," 2017. [Online]. Available: https://arstechnica.com/information-technology/2017/01/antivirus-is-bad/

4. P. H. O'Neill, "Zero day exploits are rarer and more expensive than ever, researchers say," 2017. [Online]. Available: https://www.cyberscoop.com/zero-day-vulns-are-rarer-and-more-expensive-than-ever/

5. U.S. Department of Homeland Security, "Critical Infrastructure Sectors," 2018. [Online]. Available: https://www.dhs.gov/cisa/critical-infrastructure-sectors

6. E. Strohmaier, J. Dongarra, H. Simon, and M. Meuer, "TOP500 List Statistics," 2018. [Online]. Available: https://www.top500.org/statistics/list/

7. Mellanox Technologies, "Security in Mellanox Technologies InfiniBand Fabrics," 2012.

8. ——, "Introduction to InfiniBand™," 2003.

9. G. F. Pfister, "An Introduction to the InfiniBand™ Architecture," in *High Performance Mass Storage and Parallel I/O: Technologies and Applications*, 1st ed., Rajkumar Buyya and Toni Cortes, Eds. New York, NY: John Wiley & Sons, Inc., 2001, ch. 42, pp. 617–632.

10. InfiniBand Trade Association, "About InfiniBand™," 2018. [Online]. Available: https://www.InfiniBandta.org/about-InfiniBand/

11. B. Volz, E. Cain, E. Kohler, J. Postel, M. Lottor, M. Accetta, R. Stewart, R. Adams, R. Nethaniel, R. Thomas, and R. Ullmann, "Service Name and Transport Protocol Port Number Registry," 2019. [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

12. Sarah Rubenoff, "HDR 200G InfiniBand: Empowering Next Generation Data Centers," 2018. [Online]. Available: https://insidehpc.com/2018/02/hdr-200g-infiniband-data-enters/

13. Brad Smith, "Mellanox Launches 200Gb/s HDR InfiniBand AOC and DAC LinkX® Cables," 2016. [Online]. Available: http://www.mellanox.com/blog/2016/11/mellanox-launches-200gbs-hdr-infiniband-aoc-and-dac-linkx-cables/

14. P. MacArthur, Q. Liu, R. D. Russell, F. Mizero, M. Veeraraghavan, and J. M. Dennis, "An Integrated Tutorial on InfiniBand, Verbs, and MPI," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2894 – 2926, 2017.

15. Trend Micro Inc., "Industrial Control System - Definition," 2018. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/definition/industrial-control-system

16. E. C. Eze, S. Zhang, and E. Liu, "Vehicular ad hoc networks (VANETs): Current state, challenges, potentials and way forward," in *ICAC 2014 - Proceedings of the 20th International Conference on Automation and Computing: Future Automation, Computing and Manufacturing*. Cranfield, UK: IEEE, 2014.

17. V. Beal, "The 7 Layers of the OSI Model," 2018. [Online]. Available: https://www.webopedia.com/quick_ref/OSI_Layers.asp

18. Mellanox Technologies, "RDMA Aware Networks Programming User Manual rev. 1.7," Mellanox Technologies, Sunnyvale, CA, Tech. Rep., 2015. [Online]. Available: www.mellanox.com

19. Paul Grun, "Introduction to InfiniBand ™ for End Users: Industry-Standard Value and Performance for High Performance Computing and the Enterprise," *White paper, InfiniBand Trade Association*, 2010.

20. Mellanox Technologies, "RoCE vs. iWARP Competitive Analysis," 2017.

21. ——, "Mellanox OFED for Linux User Manual rev. 4.4," Mellanox Technologies, Sunnyvale, CA, Tech. Rep., 2018.

22. Oracle Corporation, "Delivering Application Performance with Oracle's InfiniBand Technology: A Standards-Based Interconnect for Application Scalability and Network Consolidation," 2012.

23. D. A. Deming, "InfiniBand Software Architecture and RDMA," in *Storage Developer Conference (SDC)*, Storage Networking Industry Association (SNIA), Ed., Santa Clara, CA, 2013. [Online]. Available: https://www.snia.org/

24. QLogic Corporation, "Fabric Manager Users Guide v6.0, rev. A," QLogic Corporation, Aliso Viejo, CA, Tech. Rep., 2010.

25. D. Crupnicoff, S. Das, and E. Zahavi, "Deploying Quality of Service and Congestion Control in InfiniBand-based Data Center Networks," Mellanox Technologies Inc., Santa Clara, CA, Tech. Rep., 2005. [Online]. Available: http://www.mellanox.com

26. "Mellanox OpenFabrics Enterprise Distribution for Linux (MLNX_OFED)," 2018. [Online]. Available: http://www.mellanox.com/page/products_dyn?product_family=26

27. The Open MPI Project, "Open MPI Documentation," 2019. [Online]. Available: https://www.open-mpi.org/doc/https://www.open-mpi.org/faq/

28. Security Ninja, "CIA Triad," 2018. [Online]. Available: https://resources.infosecinstitute.com/cia-triad/

29. M. Chapple, D. L. Shinder, and E. Tittel, *TICSA Certification: Information Security Basics*, 1st ed. Hoboken, NJ: Pearson IT Certification, 2003. [Online]. Available: http://www.pearsonitcertification.com/articles/article.aspx?p=30077&seqNum=5

30. Indiana University, "About proxy servers," 2018. [Online]. Available: https://kb.iu.edu/d/ahoo

31. InfoSec Resources, "Security+: Authentication Services (RADIUS, TACACS+, LDAP, etc.) (SY0-401)," 2017. [Online]. Available: https://resources.infosecinstitute.com/security-plus-authentication-services-radius-tacacs-ldap-etc-sy0-401/

32. T. Heron, "Difference between Kerberos and LDAP in Active Directory," 2016. [Online]. Available: https://social.technet.microsoft.com/Forums/ie/en-US/cb6b05cc-3162-456d-a987-d113821fbdd6/difference-between-kerberos-and-ldap-in-active-directory?forum=winserverDS

33. A. Warren, "InfiniBand Fabric and Userland Attacks," Ph.D. dissertation, SANS Institute, 2012.

34. M. Lee, E. J. Kim, and M. Yousif, "Security Enhancement in InfiniBand Architecture," in *Proceedings - 19th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2005*, 2005.

35. ISACA, "Vulnerability Assessment," 2017.

36. R. Boyce, "Vulnerability Assessments: The Pro-active Steps to Secure Your Organization," Ph.D. dissertation, SANS Institute, 2001.

37. E. Hutchins, M. Cloppert, and R. Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," *Proceedings of the International Conference on Information Warfare & Security*, 2011.

38. MITRE Corporation, "ATTaCK Matrix for Enterprise," 2018. [Online]. Available: https://attack.mitre.org/

39. A. Shostack, *Threat Modeling: Designing for Security*, 1st ed., Carol Long, Victoria Swider, Tom Dinse, Chris Wysopal, Christine Mugnolo, and Luann Rouff, Eds. Indianapolis, IN: John Wiley & Sons, 2014.

40. FireEye Inc., "Advanced Persistent Threat Groups," 2019. [Online]. Available: https://www.fireeye.com/current-threats/apt-groups.html

41. K. Gonzalez, "A Step-By-Step Guide to Vulnerability Assessment," 2018. [Online]. Available: https://securityintelligence.com/a-step-by-step-guide-to-vulnerability-assessment/

42. Tenable Inc., "Nessus Professional," 2019. [Online]. Available: https://www.tenable.com/products/nessus/nessus-professional

43. National Institute of Standards and Technology, "Cybersecurity Framework," 2019. [Online]. Available: https://www.nist.gov/cyberframework

44. S. Peisert, "Security in High-Performance Computing Environments," *Communications of the ACM, Vol. 60 No. 9*, pp. 72–80, 9 2017. [Online]. Available: https://cacm.acm.org/magazines/2017/9/220422-security-in-high-performance-computing-environments/fulltext

45. Mellanox Technologies, "SX6012 Switch Product Brief," 2018. [Online]. Available: http://www.mellanox.com/related-docs/prod_ib_switch_systems/PB_SX6012.pdf

46. ——, "InfiniBand Software and Protocols Enable Seamless Off-the-shelf Applications Deployment," 2007.

47. M. M. J. Stevens, "On Collisions for MD5," Ph.D. dissertation, Eindhoven University of Technology, 2007.

48. B. Schneier, "Cryptanalysis of SHA-1," 2005. [Online]. Available: https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html

49. Mellanox Technologies, "Mellanox Quantum™," 2018. [Online]. Available: http://www.mellanox.com/page/products_dyn?product_family=264&mtag=quantum

50. ——, "200Gb/s ConnectX-6 Ethernet Single/Dual-Port Adapter IC," 2019. [Online]. Available: http://www.mellanox.com/page/products_dyn?product_family=268&mtag=connectx_6_en_ic

51. J. Grand, "Hardware Reverse Engineering: Access, Analyze, and Defeat," in *Black Hat Workshop*, Washington D.C., 2011. [Online]. Available: https://media.blackhat.com/bh-dc-11/Grand/BlackHat_DC_2011_Grand-Workshop.pdf

52. NIST, "Cyber Supply Chain Risk Management," 2018. [Online]. Available: https://csrc.nist.gov/projects/supply-chain-risk-management

53. OpenFabrics Alliance, "Index of /downloads/management," 2018. [Online]. Available: https://www.openfabrics.org/downloads/management/

54. GitHub Inc., "linux-rdma/opensm," 2018. [Online]. Available: https://github.com/linux-rdma/opensm/tree/master/opensm

55. National Institute of Standards and Technology, "National Vulnerability Database," 2019. [Online]. Available: https://nvd.nist.gov/

56. Mellanox Technologies, "LinkX® Infiniband DAC Splitter Cables," 2019. [Online]. Available: http://www.mellanox.com/products/interconnect/infiniband-copper-splitter.php

57. SolarWinds MSP, "Network Vulnerability Assessment | 6 Vital Steps," 2019. [Online]. Available: https://www.solarwindsmsp.com/content/network-vulnerability-assessment-steps

58. K. Van Impe, "Simplifying Risk Management," 2017. [Online]. Available: https://securityintelligence.com/simplifying-risk-management/

59. A. Chu, "Ibccconfig manual page," 2018. [Online]. Available: http://manpages.ubuntu.com/manpages/xenial/man8/ibccconfig.8.html

60. Mellanox Technologies, "Firmware Downloads," 2018. [Online]. Available: http://www.mellanox.com/page/firmware_download

61. Dell Technologies, "How to enable HTTPS/SSH and disable HTTP/Telnet for switch management on PowerConnect 5500 series switches," 2018. [Online]. Available: https://www.dell.com/support/article/us/en/04/how10442/how-to-enable-https-ssh-and-disable-http-telnet-for-switch-management-on-powerconnect-5500-se lang=en

62. cyberang3l, "InfiniBand-Graphviz-ualization," 2016. [Online]. Available: https://github.com/cyberang3l/InfiniBand-Graphviz-ualization

63. Wireshark Foundation, "Wireshark InfiniBand Filters," 2019. [Online]. Available: https://www.wireshark.org/docs/dfref/i/infiniband.html

64. G. F. Lyon, "nmap(1) - Linux man page," 2011. [Online]. Available: https://linux.die.net/man/1/nmap

65. OpenLDAP Foundation, "OpenLDAP," 2018. [Online]. Available: https://www.openldap.org/

66. Oracle Corporation, "When Downgrading Infiniband Switch from 2.2.8 or higher to 2.2.7 or lower: Error: Signature verification failed! (Doc ID 2405979.1)," 2018. [Online]. Available: https://support.oracle.com/knowledge/Sun%20Microsystems/2405979_1.html

67. Mellanox Technologies, "Mellanox MLNX-OS User Manual," Mellanox Technologies, Sunnyvale, CA, Tech. Rep., 2018. [Online]. Available: http://www.mellanox.com/

68. D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

69. Mellanox Technologies, "Unified Fabric Manager (UFM®) Software for Data Center Management," 2019. [Online]. Available: http://www.mellanox.com/page/products_dyn?product_family=100&mtag=unified_fabric_manager

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
03–01–2019	Master's Thesis	Sept 2017 — Mar 2019

4. TITLE AND SUBTITLE

A Framework for Cyber Vulnerability Assessments of InfiniBand Networks

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Capt Daryl W. Schmitt

5d. PROJECT NUMBER

18G230

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-19-M-054

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
2241 Avionics Circle
WPAFB, OH 45433-7765
Attn: Steven Stokes
COMM 937-528-8035, Email: steven.stokes@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RYWA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

InfiniBand is a popular Input/Output interconnect technology used in High Performance Computing clusters. It is employed in over a quarter of the world's 500 fastest computer systems. Although it was created to provide extremely low network latency with a high Quality of Service, the cybersecurity aspects of InfiniBand have yet to be thoroughly investigated. The InfiniBand Architecture was designed as a data center technology, logically separated from the Internet, so defensive mechanisms such as packet encryption were not implemented. Cyber communities do not appear to have taken an interest in InfiniBand, but that is likely to change as attackers branch out from traditional computing devices. This thesis considers the security implications of InfiniBand features and constructs a framework for conducting Cyber Vulnerability Assessments. Several attack primitives are tested and analyzed. Finally, new cyber tools and security devices for InfiniBand are proposed, and changes to existing products are recommended.

15. SUBJECT TERMS

InfiniBand, Networking, Cybersecurity, Cyber Vulnerability Assessment

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Scott Graham, AFIT/ENG
U	U	U	UU	111	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4581; scott.graham@afit.edu