

Team BluSh3ll

Lockheed Martin

Hunter Rowlette, Mitchell Palmer, Andrew Chapin, Andre Herrera, Carl Bai

October 4, 2019

V0.1

Table of Contents

[1.0 Executive Summary 2](#_Toc21114895)

[2.0 Description of the Problem 3](#_Toc21114896)

[3.0 Objectives 3](#_Toc21114897)

[4.0 Technical Approach 4](#_Toc21114898)

[5.0 Equipment and Facilities 7](#_Toc21114899)

[6.0 Deliverables 7](#_Toc21114900)

[7.0 Cost and Management Proposal 8](#_Toc21114901)

[7.1 Organization Chart and Qualifications Per Example in the RFP. 8](#_Toc21114902)

[7.2 WBS identifying tasks detail, labor hours and non-labor costs Per Example in the RFP. 10](#_Toc21114903)

[7.3 Project Schedule per Example in the RFP. 10](#_Toc21114904)

[7.4 Project Cost per example in the RFP. 10](#_Toc21114905)

# 1.0 Executive Summary

Lockheed Martin (LM) Space, the Sponsor, provides large scale and complex systems to the US Government and other customers. Delivery of various files and executables to remote systems is an area of concern for Cybersecurity risks to LM and their customers. This project facilitates the exploration and development of methods to securely deliver files, executables, updates and data to remote systems. To that end, the Sponsor desires a packer/loader tool capable of performing compression and encryption capabilities on binary executables.

# 2.0 Description of the Problem

Delivery of various files and executables to remote systems is an area of concern for Cybersecurity risks to LM and its customers. LM desires a method to securely deliver files, executables, updates and data to remote systems through a packer/loader tool. The packer is to be a program on the local system that will compress and encrypt data. The encryption should be password based. The packer must support binary executables. The loader is a process that will be run on the remote system, listening for incoming data. The loader should automatically decrypt and decompress incoming data. If the data is detected to be an executable file, the file should be run automatically completely in memory.

# 3.0 Objectives

LM has described serval required objectives in their RFP that must be included into the final deliverable.

* Ability to compress and encrypt a Windows based binary executable
* Encryption should be password based
* Portable Executable (PE) file format binary executables must be supported
* Compression/Encryption (Packer) capability must output to a single file
* Packer must be able to support dynamic linking against libraries
* Loader must be able to take Packer output and decrypt it
* Loader must be able to output Packer file contents based on user defined, configurable parameters
* Loader must be able to identify when Packer file contents are executables and run them
* Running of executables by the Loader capability must be configurable, allowing user to toggle between an automatic run and a prompt during decryption/decompression
* Loader must run executables in memory as a separate thread/process

Along with these they have several objectives that they have listed as desired.

* Compression and encryption of multiple files at a time to a single file
* PE32+ file format binary executables supported
* Loader never touches disk (e.g. everything happens in memory: decryption, decompression, and execution)
* Packer supports both static and dynamic linking against libraries
* Ability to also run in Linux/compress and encrypt a Linux based binary executable
* ELF file format binary executables supported
* Ability to also run in macOS/compress and encrypt a macOS based binary executable
* Mach-O file format binary executables supported

# 4.0 Technical Approach

**Requirements**

* A packer and a loader, two separate pieces of software
* The packer shall compress and encrypt a Windows based binary executable
* The loader shall listen for incoming packed data and decrypts it
* The loader shall be able to detect executables
* The loader shall be able to run these in memory as a separate thread/process
* The loader shall toggle between an automatic run and a prompt during decryption/decompression
* The packer and loader shall support PE format (with PE32+, ELF, Mach-O desired)
* The encryption shall be AES-256 and password based

**Risks**

* Network Restrictions
  + If the network architecture of the sponsor contains firewall rules or networking segmentation that we do not account for, then the packer will not be able to establish a connection with the loader and we will not be able to meet our project requirements. As such, additional time will be required to allow the packer to communicate in more restricted networks, and we will need to adjust the schedule accordingly.
* System Requirements
  + If the machine where the loader is deployed does not meet our code’s expected requirements, then it will not be able to function properly. Additional time would be required to redesign parts of the loader to get around these restrictions.

**Design Constraints**

* Our combined knowledge and skillsets
  + While the team has extensive experience in software development, some of the particulars of creating a packer and loader combined with network file distribution are outside of members’ area of expertise.
* Limited to open source software
  + Without adding a budget to the project to purchase expensive proprietary software, the team can only use free open source software. Additionally, with free closed source software, the team would be unable to determine its inner workings and as such would be unable to guarantee the software does what it claims to do. Similarly, the team would be unable to guarantee that the software only does what it claims to do.
* Time allocated to the project
  + With the hard time constraints imposed by completing this project as a senior capstone, any requirements needing more time than the two combined semesters will be impossible to complete.

**CONOPS Diagram**

From a user’s perspective, they will start by selecting a data block to be packaged. From there, the data block will be copied to the packer’s server, where it will be packed and sent through the network to the selected receiver. Once the receiving computer has completed receiving the encrypted data, the packer will decrypt the data block and will either run it or make it available on the system.

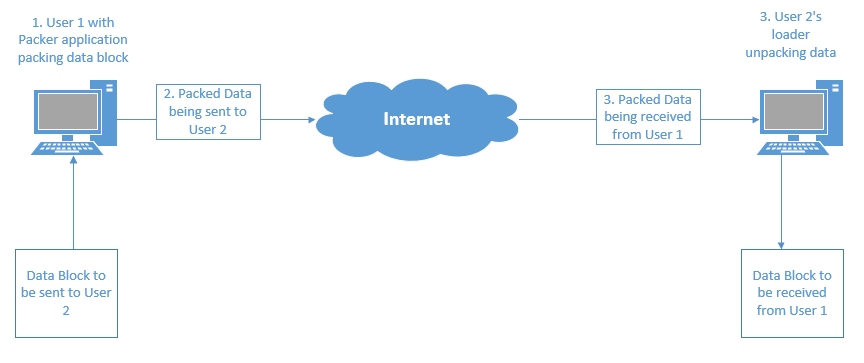
****

Figure - CONOPS Diagram

**Architecture**

The project has two major components: a server containing the packer application and client nodes containing a loader. The server and nodes will be connected via a network, allowing for bi-directional data transition. The nodes have a heartbeat with the server to inform it that they are available for data to be pushed or the status of the data received. The data sent to the nodes is transmitted over an AES encrypted tunnel on the network. The AES cryptographic operations are handled in-software on both ends, thus allowing the network traffic to be standard packets.

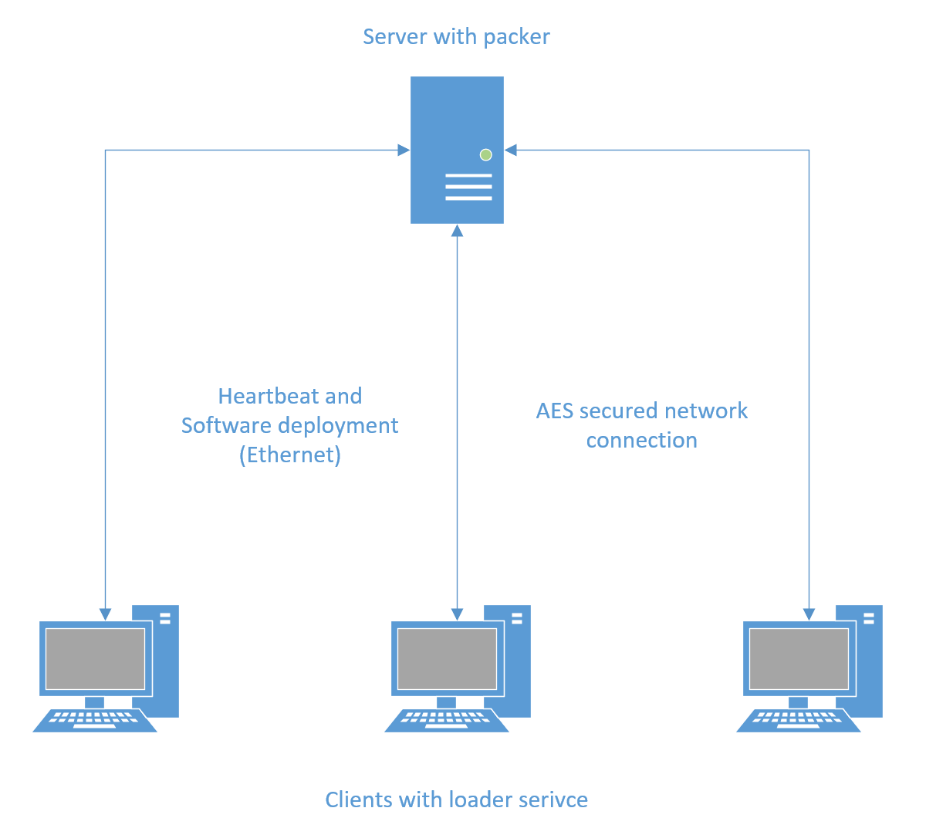


Figure 2 - Architecture Diagram

**Data Model**

Input data to our project will be sent to the packer server, where the packer application will package it by compressing and encrypting the data. When a package needs to be pushed to a client node, the packer server will transmit this encrypted data over an arbitrary network to the node. When the node receives the data, it will be loaded into memory where the loader stored on the hard drive will decrypt it into memory. This decrypted data will then be checked to see if it is an executable. If it is an executable, it will be run in memory. This program will then be able to write files to the hard drive. If the data pack is not a program, it will be written to the hard drive.

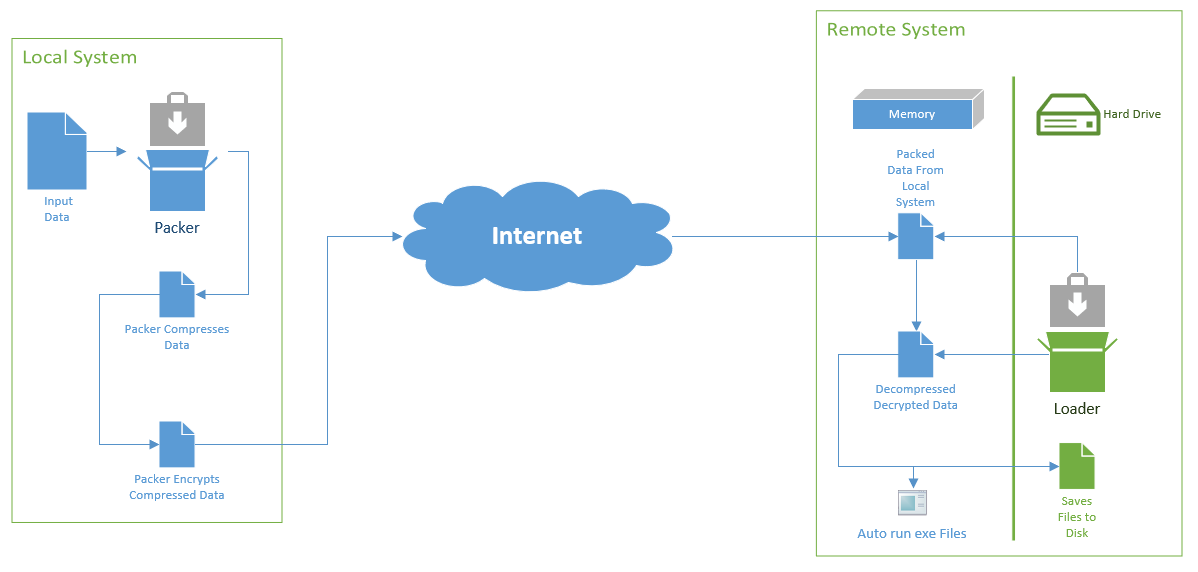


Figure - Data Model Diagram

**Verification Method**

Quality assurance through integration and unit tests of the packer and loader software products. Final confirmation from the sponsor of the functionality of the product. Customer testing in the environment which will provide user confirmation of the function of the product. Alterations to be made as requested. Final packaging to meet sponsor and customer requests.

**Tasks**

* Requirements
* Research & Design
  + Existing Tools
    - Statement on why competitors fail to meet customer needs
  + General Architecture
    - Refinement of features
    - Defining minimal viable product at each development stage
  + Design
* Implementation/Coding
  + Packer initial development
    - Implementation of basic modules
    - Unit testing
    - Initial integration testing
  + Loader initial development
    - Implementation of basic modules
    - Unit testing
    - Initial integration testing
  + Packer loader networking development
    - Implementation of network modules
    - Integration testing
  + Packer / Loader networking integration
    - integration testing
  + Deployment mechanism development
    - Implementation of basic modules
    - integration testing
* Testing & Quality Assurance (Q.A.)
  + Packer Quality Assurance
    - final confirmation of function
  + Loader Quality Assurance
    - final confirmation of function
  + networking Integration Quality Assurance
    - final confirmation of function
  + customer testing in environment
    - user confirmation of function
    - alterations made as requested
  + final developed for packaging and workflow to meet customer requests

# 5.0 Equipment and Facilities

There are no unique hardware, software or data needs for this project. All software that is needed to support development activities is available and open source.

# 6.0 Deliverables

* CLIN-1 (Contract Line Item Number 1) Customer Reporting “Quad -Pack”
* CLIN-2 Weekly Activity/Time Sheet
* CLIN-3 Color Team Briefing
* CLIN-4 Proposal (as a response to this RFP)
* CLIN-5 Design Review Briefing
* CLIN-6 Poster Paper
* CLIN-7 Encryption and Compression Design and Techniques Report
* CLIN-8 Final Report and Team Presentation
* CLIN-9 Product Specifications
* CLIN-10 Packer/Loader Source Code and Completed/Compiled Tool

# 7.0 Cost and Management Proposal

## 7.1 Organization Chart and Qualifications Per Example in the RFP.

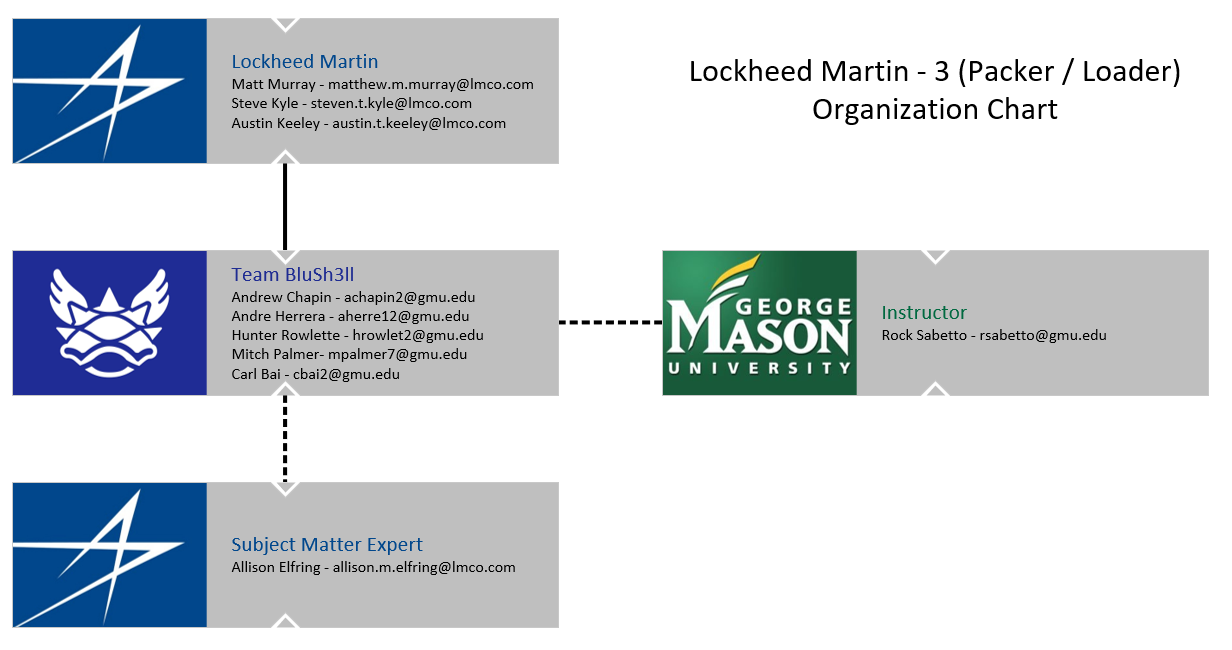


Figure - Organization Chart

Hunter Rowlette is a Cybersecurity Major at George Mason University. Hunter is currently employed at the Card Access Security Office where he works to maintain and upkeep several GMU’s systems to include card access, CCTV, and POS terminals for all 5 GMU campuses located in Virginia. Through this job and his education, he has acquired experience with networking, operating systems, and coding particularly with C and Python. You can reach him at [hrowlet2@gmu.edu](mailto:hrowlet2@gmu.edu).

Mitchell Palmer is a Cybersecurity Engineering Intern at MITRE where he maintains a digital forensics toolkit and performs malware reverse engineering. Before starting at MITRE, Mitchell spent two years with the Department of Justice where he analyzed data for criminal and national security investigations on a cyber task force. He also has experience with networking, software development, coding in C, and operating systems. At school, he is active as a tour guide and with club ultimate frisbee. You can reach him at [mpalmer7@gmu.edu](mailto:mpalmer7@gmu.edu).

Andre Herrera is a Cybersecurity Engineering Major at George Mason University. Andre Interned at M&T bank last summer where he worked under the Cyber Operations department in network defense. There he worked with their SIEM tool including running searches to gather data for review and creating new rules that generate offenses when the proper criteria is reached for the cyber analysts in the SOC to investigate. Andre has networking experience routing, switching, dynamic routing protocols, IP Addressing, subnetting, ACLs, port security, VLANs, DHCP, trunking, and NAT. He has experience with both Linux and Windows OS as well as programming knowledge in C and Python and Java. You can reach him at [aherre12@gmu.edu](mailto:aherre12@gmu.edu).

Carl Bai has worked at the Federal Reserve Bank of Boston in Directory Services, as a Help Desk Analyst at the New England Center for Children, as a Network and Information Security Consultant for JoeBai Dot Com, and in development operations at Continuum Managed Services LLC. Additionally, Carl has worked as a freelance software developer on a variety of professional and public projects. He has experience with networked systems both in engineering and administration, including Cisco, Extreme, and Ubiquti networking products and Windows and Linux server and desktop configuration, deployment, and support. Additional, enterprise technology experience includes IBM mainframes and cloud technologies, including Amazon Web Services and Microsoft Azure development and administration. Carl has also supported mixed client environments including Windows, Apple, and Linux/Unix systems. As part of development, He has extensive experience with Python, C, C++, C#, PHP, Basic, JAVA, JavaScript, RUBY development and has some experience in many more. You can reach him at [cbai2@masonlive.gmu.edu](mailto:cbai2@masonlive.gmu.edu).

Andrew Chapin is a senior at George Mason University studying Cyber Security Engineering. When he isn’t attending classes or working tech support at the Volgenau Engineering building, you can find him playing CTFs and HackTheBox. Andrew has helped develop multiple educational resources surrounding cyber security topics for various CYSE classes at GMU. Working with Proff. Jim Jones, Andrew helped lead a team of penetration testers to research network intrusion detection and prevention systems. You can learn more about Andrew’s work by checking out his website ([www.achapin.com](http://www.achapin.com)) or emailing him at [achapin2@gmu.edu](mailto:achapin2@gmu.edu).

## 7.2 WBS identifying tasks detail, labor hours and non-labor costs Per Example in the RFP.

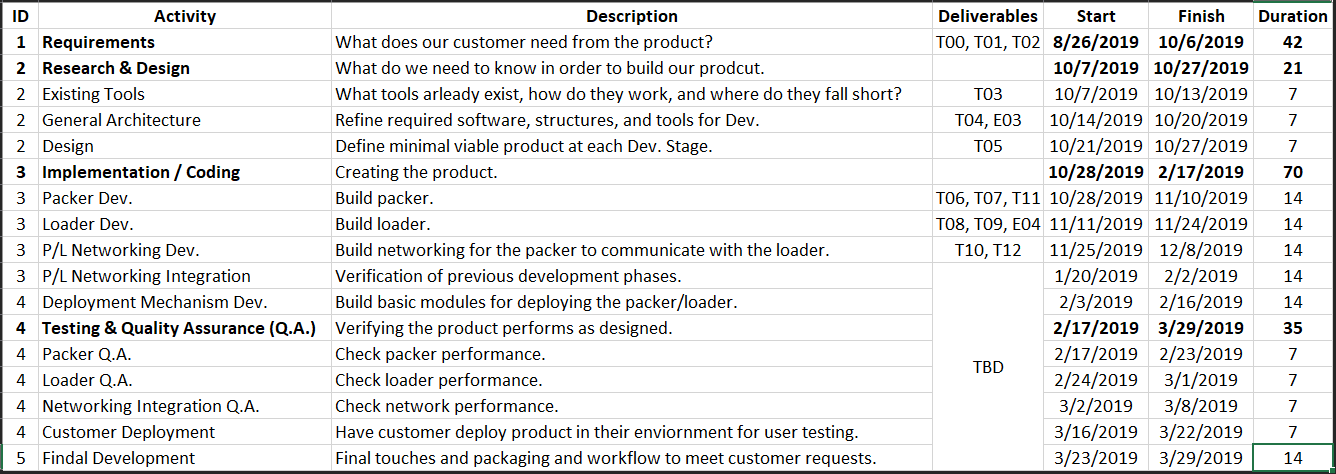


Figure - Work Breakdown Structure

## 7.3 Project Schedule per Example in the RFP.

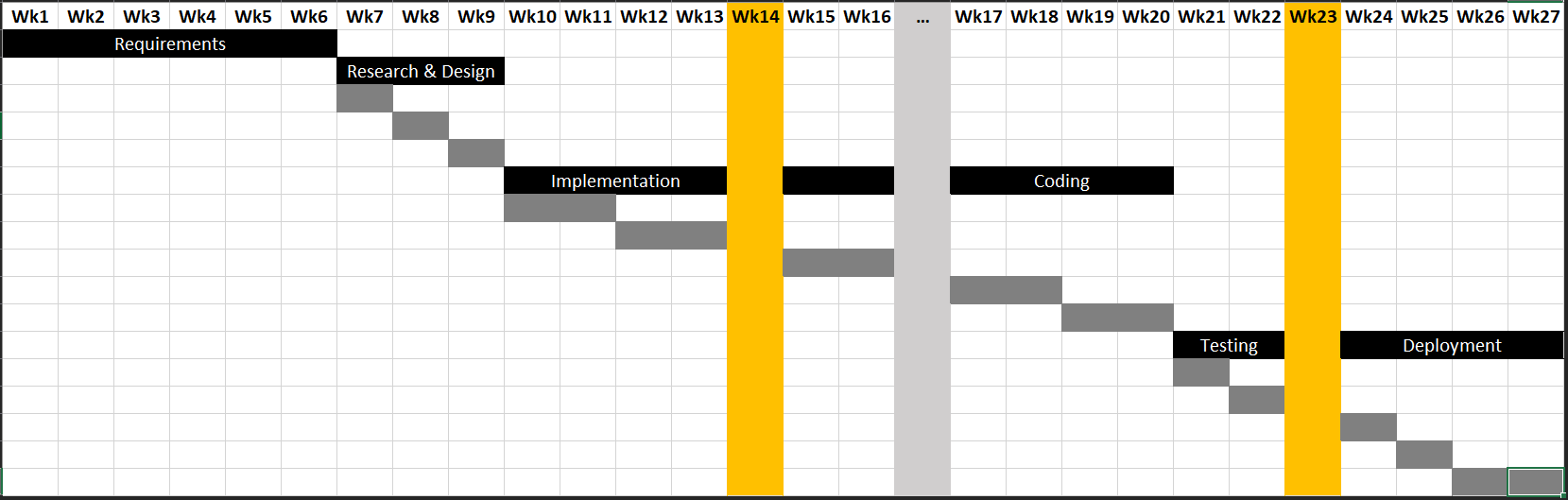


Figure - Project Schedule

## 7.4 Project Cost per example in the RFP.

No costs are attributed to this project outside of labor. Labor costs are outlined in *Figure 7* with hours worked by all group members each week of the projects span. The two weeks with no hours worked correspond to holiday breaks (shown by the yellow columns in *Figure 6*).

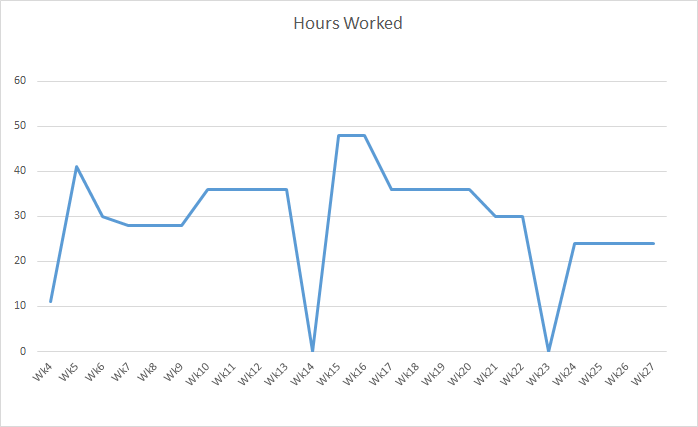


Figure 7 - Project Cost in Labor Hours