HealthNet

Group 3

Andrew Hill
Catilina Self
Amy Stewart
Lance Fisher
Patrick Taylor

Revision	Date	Author(s)	Remarks
1.0	29 March 2016	Group 3	Initial Writing
2.0	26 April 2016	Group 3	Revisions
3.0	28 April 2016	Catilina, Amy, Andrew	Revised and finalized for submission

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Design Overview

1.1.1 Description of Problem

Create a secure software for creating, retrieving, and storing the medical records of patients for the client hospital.

1.1.2 Technologies Used

The Health Net Software will communicate with input devices designed for healthcare professionals. The required devices include a keyboard, mouse, display monitor, and computer.

The target platform will be Rose Checkers Linux Operating System and the development environment is C programming.

1.2 Requirements Traceability Matrix

Requirement	UC	SD
Authenticating User Requirements	2	4.1.2.1
Generating/adding to a Parameterized Report	3	4.1.2.2
Collection of Patient Demographics	5	4.1.2.2
Active Medication List	6	4.1.2.2
Protect Electronic Health Information	7	5.3.6
Creating Patient Accounts	8	4.1.2.2
Patient Release of Lab Test Results	10	4.1.2.2

2 SYSTEM ARCHITECTURAL DESIGN

2.1 Chosen System Architecture

Basic Architecture includes application based in C programing operating on a Rose Checkers Linux operating system.

Health Net consists of the following components.

1. User Interface - A command line prompt with menu options and the ability for users to enter data.
2. Security Shell - All text fields are filtered through a security wrapper that verifies all input from the user. Additionally users are initially authenticated with a username and password to verify authentication. User permissions are based on authentication. Each action the user performs is verified against the user's permissions.
3. File I/O Data Storage - All data is encrypted and stored in a file. Data is retrieved from the file based on user's identification and permissions.

2.2 Discussion of Alternative Designs

Some argument was made for using a GUI driven interface that used the button listeners and text boxes as a line of defense, but this idea was scrapped in favor of the more secure command line model.

2.3 System Interface Description

The Programs primary interface is a command line text-based program. The user is able to enter basic information, such as; filename, numerical options, etc. All inputs are verified at through the Security Shell. User actions include logging in, updating a record, viewing a record, creating records and logging out.

3 DETAILED DESCRIPTION OF COMPONENTS

3.1 Component Overview

Health Net is composed of the following components.

1. Database
2. Class Library
3. Client Software
4. Security Shell

3.2 Database

We will be using file IO as for our database. Data is segregated based on user accounts and permissions. Users must have specific permissions to view files. The file IO also contains information various application data.

3.3 Class Library

The Class Library is a package of classes common to file IO implementation. The file software will use these class definitions to know how to store the objects in the database. The client software will use the class definitions in order to ask for and display the correct information about each class type.

3.4 Client Software

The Client Software will reside on a desktop computer. Its purpose is to present data to the user as requested and provide an interface so that the user can easily update application information. It will handle conflict resolution automatically when possible and otherwise allow the user to choose which version of an item to keep.

3.5 Security Shell

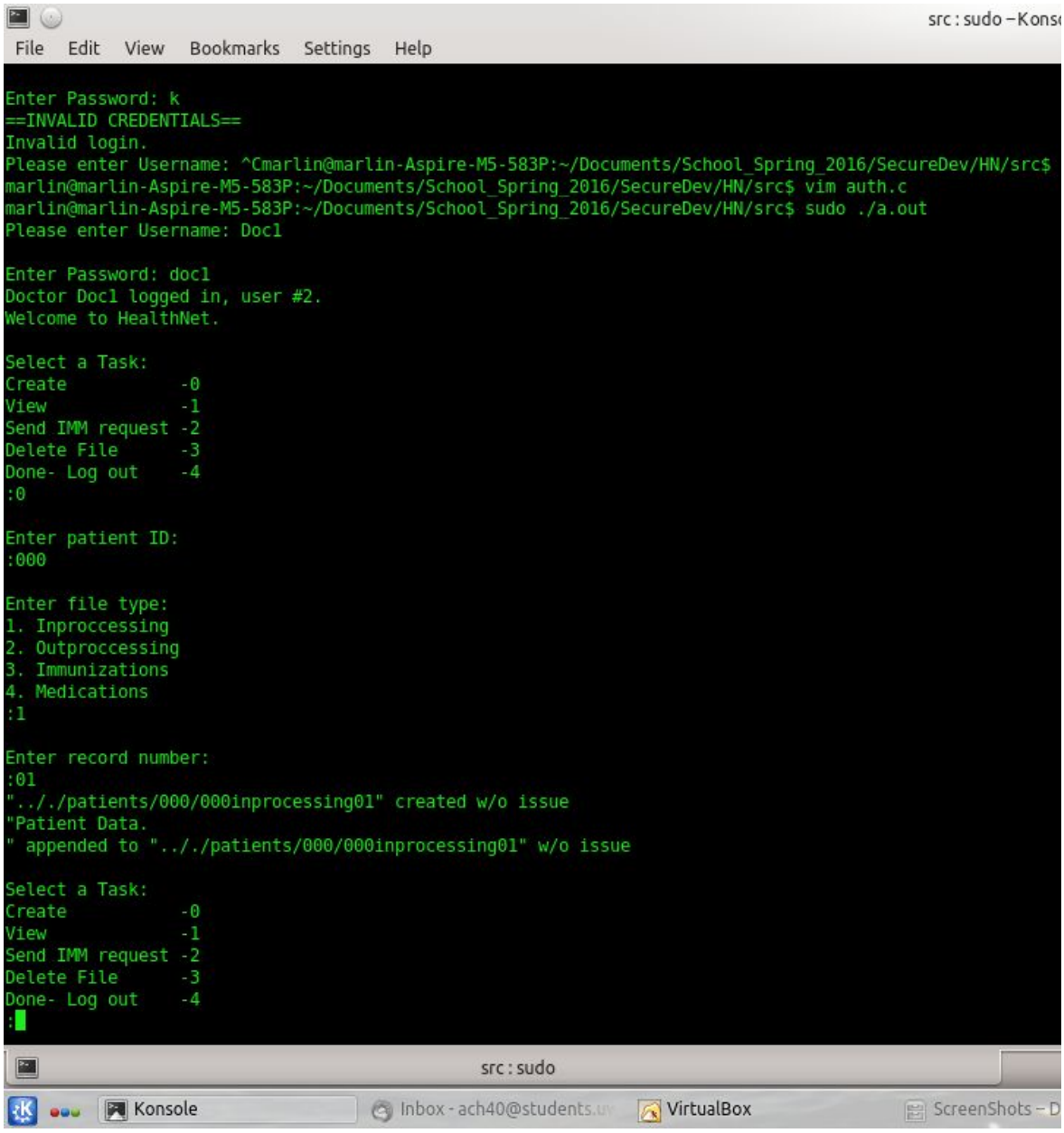
The Security Shell screens all input from the user in addition to verifying the user's identity through a username and password. Logins are given access or denied access to the client software based on correct username and passwords. The username and password create a unique key that is compared against the user's permission set. The version of client software presented to the user is based on the user's permission set. Each action the user selects is verified against the user's permissions. Actions are permitted or denied based on the permission set.

4 USER INTERFACE DESIGN

4.1 Description of the User Interface

User interface is plain text via terminal. Health Net prompts the user for information such as a username and password. HealthNet then displays the output and further input requests based on input validation algorithms.

4.1.1 Screen Images



4.1.2 Objects and Actions

4.1.2.1 Authentication

Step 1: Start the Application
Step 2: Enter username when prompted. Press [Enter].
Step 3: Enter password when prompted. Press [Enter]. Success message is shown upon successful login. Denied Access message appears for an incorrect username and password. Repeat Steps 1-3.
Step 4: For additional questions or concerns when using the Health Net software please contact our customer service representative at 1800-NOA-NSER.

4.1.2.2 Menu Options

Complete section 4.1.2.1 prior to begining this seciton.
Users can conduct the following actions based on permissions.

1. View Records
   Step 1: Enter menu option to perform. Press [Enter]. Actions are based on user permissions.
   Step 2: Results are displayed on the terminal window. Menu is displayed.
   Step 3: Repeat Steps 4-6 as needed. Follow the prompts.
2. Create Records
   Step 1: Enter menu option to Create Records. Press [Enter]. Actions are based on user permissions.
   Step 2: Software prompts for information. Enter the specified information. Press [Enter].
   Step 3: Continue Step 2 until Health Net has completed the form. A menu option appears when the form is complete.
   Step 4: Repeat Steps 2-3 as needed. Follow the prompts.
3. Edit Records
   Step 1: Enter menu option to Edit Records. Press [Enter]. Actions are based on user permissions.
   Step 2: Software prompts for information. Enter the specified information. Press [Enter].
   Step 3: Continue Step 2 until Health Net has completed the form. A menu option appears when the form is complete.
   Step 4: Repeat Steps 2-3 as needed. Follow the prompts.
4. Logout
   Step 1: Enter menu option to Logout. Press [Enter]. A message is displayed that states a successful Logout.

5 ADDITIONAL MATERIAL

5.1 Establish Design Requirements

Name:	Fail-Safe Defaults
Summary	Base access decisions on permission rather than exclusion.
Application	Access is only granted to the Health Net software through approved credentials.

Name:	Complete Mediation
Summary	Every access to every object must be checked for authority.
Application	At every new action selected, the user's credentials are checked for permission before access is granted.

Name:	Least Privilege
Summary	Every program and every user of the system should operate using the least set of privileges necessary to complete the job.
Application	Every user is only granted privileges according to their account type.

Name:	Least Common Mechanism
Summary	Minimize the amount of mechanisms common to more than one user and depended on by all users.
Application	

Name:	Psychological Acceptability
Summary	It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly.
Application	The security guards are built into this software so that the user is required to do virtually nothing in terms of ensuring safety of the information in the Health Net file system. This was to make the software as user friendly as possible and ensure that the users would be able to effortlessly and continuously use the product without any inconvenience. This facilitates a

	likelihood for continued reuse of the software and low training required for use.

5.2 Attack Surface Analysis/Reduction
The following site was used as a base for HeathNet Attack Surface Analysis/Reduction, https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet.

5.2.1 Attack Surface Analysis
The Health Net attack surface is broken into three areas.

Targets - the processes or data resources an attacker aims to control. Health Net targets are permissions of the highest privileged user (Doctor) and having access to the hash keys used to locate specific user data, and access to the File IO.

Channels - method of communicating information. Health Net uses a command line interface for communication with a user.

Access rights - the resources associated privileges. Health Net associates the following access rights with user permissions; user actions, data input, and data retrieval.

5.2.1 Attack Surface Reduction
Targets

User Permissions - implement a multiple level verification of the user.
hash keys - ensure the data storage of Hash keys is encrypted and can only be accessed by the security shell.
File IO - ensure File IO can only be accessed through the Health Net application. Verify all user permissions when accessing File IO.

Channels

Command line interface - verify all user input using a security shell. Implement whitelisting to verify all allowed characters.

Access rights

User actions - verify the user has the correct set of permissions before completing the requested action.
Data input - verify all data input and ensure whitelisting is implemented.
Data retrieval - store data according to unique hash key. Ensure user is verified prior to data retrieval.

5.3 Threat Modeling

The following site was used as a base for HeathNet Threat Modeling,
https://msdn.microsoft.com/en-us/library/ff648644.aspx.

5.3.1 Key Terms

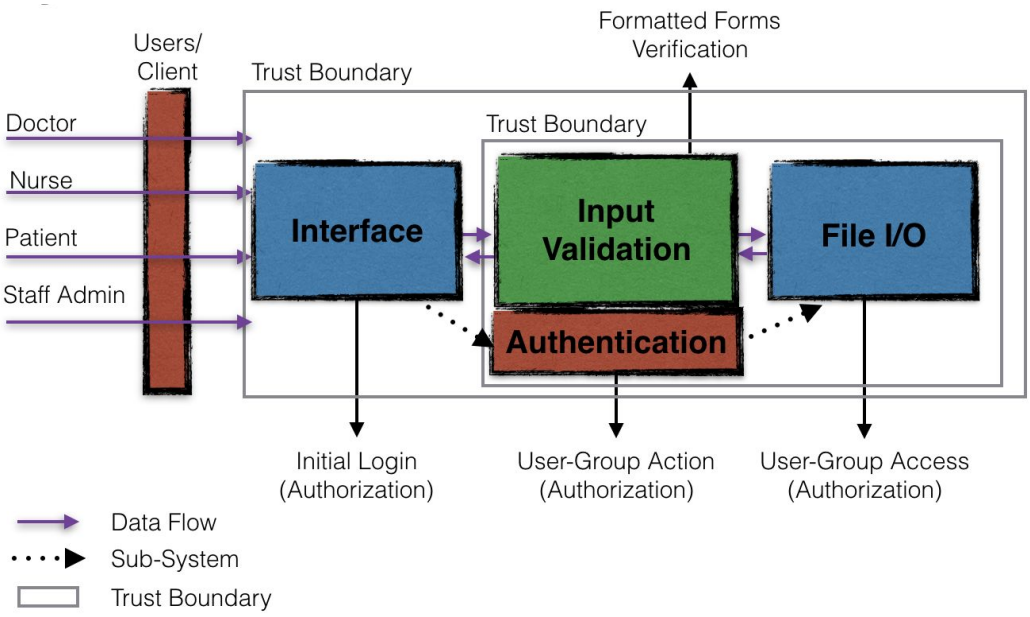
Asset. A resource of value, such as the data in a database or on the file system. A system resource.
Threat. A potential occurrence, malicious or otherwise, that might damage or compromise your assets.
Vulnerability. A weakness in some aspect or feature of a system that makes a threat possible.
Vulnerabilities might exist at the network, host, or application levels.
Attack (or exploit). An action taken by someone or something that harms an asset. This could be someone following through on a threat or exploiting a vulnerability.
Countermeasure. A safeguard that addresses a threat and mitigates risk.

5.3.2 Identified Assets

The Application must protect the following assets.

1. Hash Keys
2. Patient Information
3. Healthcare Professional Accounts
4. Patient Accounts
5. System Admin Account
6. Log Files
7. File I/O Availability

5.3.3 Architecture Diagram



5.3.4 Implementation Technologies

Technology/Platform	Implementation Details
Linux Rose Checkers Operating System	Includes user accounts and user-group permissions.
File Input and File Output	Used to store user data, provide views and store account information.
Command Line Access Software	Stored procedures.
Encryption Library (Any suggestions on what to use?)	Used to encrypt usernames and passwords, and sensitive account data.

5.3.5 Application Decomposition

Security Profile		Trust Boundaries
Input Validation	Exception Management	Data Flow
Authentication	Auditing and Logging	Entry Points
Sensitive Data	Profile Management	Privileged Code
Authorization	Hash Key Creation	Access to Records
		User Group Permissions

5.3.6 Security Profile

Category	Considerations
Input Validation	Is all input data validated? Could an attacker inject commands or malicious data into the application? Is data validated as it is passed between separate trust boundaries (by the recipient entry point)? Can data in the database be trusted?
Authentication	Are credentials secured if they are passed between functions? Are strong account requirements used? Are strong passwords enforced? Are unique usernames enforced? Are password verifiers (using one-way hashes) used for user passwords?
Sensitive Data	What sensitive data is handled by the application? How is it secured when stored? Can the file be opened without the application? How is access to data verified with user authorization and authentication? What type of encryption is used and how are encryption keys secured?
Authorization	What gatekeepers are used at the entry points of the application? How is authorization enforced with the File I/O? Is a defense in depth strategy used? Do you fail securely and only allow access upon successful confirmation of credentials? Are application functions verified against the user's authorization to conduct actions?
Hash Key Creation	Is the hash key created unique? Are the hash keys stored securely?
Exception Management	How does the application handle error conditions? Are exceptions ever allowed to propagate back to the client? Are generic error messages that do not contain exploitable information used?
Auditing and Logging	Does the application log actions conducted by the user? How are log files secured? Who can access the log files?
Profile Management	Can users have multiple profiles? How are user profiles enforced? How are user profiles verified with the user’s authorization and authentication?

5.3.7 Identified Attack Pattern Threats with Rating

Threat	Countermeasures	Threat Level
Tampering with data	Use data hashing and signing. Use digital signatures. Use strong authorization. Use tamper-resistant protocols across communication links. Secure communication links with protocols that provide message integrity.	Umm
Repudiation	Create secure audit trails. Use digital signatures. Ensure applications failover functionality. Do not need to consider high volume traffic due to scope of project. Limited to one user.	Meh
Information disclosure	Use strong authorization. Use strong encryption. Do not store secrets (for example, passwords) in plaintext. Do not store sensitive patient data in plaintext.	WTF
Password Cracking	Limit default accounts and do not use “administrator”. Apply lockout policies to end-user accounts to limit the number of retry attempts that can be used to guess. Log failed logins for patterns of password hacking attempts.	Meh
Arbitrary Code Execution	Validate input for all fields. Protect against buffer overflows for the stack and heap.	WTF
Unauthorized Access	Lock down file access based on permissions of user account. Verify all user authorization prior to completing application functions.	WTF
Input Validation	Buffer overflow; code injection; canonicalization	WTF
Authentication	Brute force attacks; dictionary attacks	Meh
Cryptography	Poor key generation or key management; weak or custom encryption	Umm
Parameter Manipulation	Query string manipulation; form field manipulation	Meh

Exception Management	Information disclosure; denial of service	Meh
Auditing and Logging	User denies performing an operation; attacker exploits an application without trace; attacker covers his or her tracks	Meh