CS 1653: Applied Cryptography and Network Security
Term Project, Phase 1

Lindsey Bieda leb35@pitt.edu Tucker Trainor tmt33@pitt.edu

January 19, 2012

Section 1: Group Information

Lindsey Bieda leb35@pitt.edu
Tucker Trainor tmt33@pitt.edu

Section 2: Security Requirements

System Assumptions & Threat Models

Threat Model 1

Preventing malicious users from accessing the system or authorized users from defacing existing data. Threats to data may come from outside the system or within. Unauthorized users should not be granted access to the systems, and methods involving login control and network security must be in place to prevent outside attacks to the system. Authorized users, whether intentional or not, should not be allowed to improperly access data to which they do not have rights to. Additionally, authorized users should not be able to unintentionally alter data via accidental overwrites or other common misfortunes.

- **Property 1: User verification.** User verification indicates that users accessing the the system need to be verified as the correct user they are logging in as. Through the use of simple techniques such as password based access. Additionally, IP-based identification of a user and SSH-key verification should be employed to further assist in verifying user access and identity. This is done in order to prevent unauthorized access to accounts and prevent multiple connections to the same account.
- **Property 2: User/group uniqueness.** User and group account names must be unique in order to ensure no collisions occur with respect to file permissions. In the case of a group or user name being created or modified in such a way that it is the same as another the system must respond in such a way that this does not happen.

- **Property 3: Strong login protocols.** Strong login protocols specifies that when logging into the system certain precautions must be taken to prevent brute forcing account access. This should be implemented by restricting failed authentication by a number of tries (eg: three) and timing out for a period of time before allowing more tries to take place.

- **Property 4: Connection restrictions.** Connection restrictions speaks to disallowing incoming connections from more than just what the users require in order to connect. Limiting port access prevents casual snooping or hacking and may mitigate Denial-of-Service attacks. The network should allow just as much connectivity as is necessary.

- **Property 5: Password integrity (and authentication).** Password integrity indicates that passwords shall be kept secure via never being exchanged or stored in clear text. Additionally, a strong password policy will be implemented such as requiring a minimal length, minimum character set, and expiration of passwords on a regular schedule.

- **Property 6: Filename uniqueness.** Filename uniqueness similar to group and user name uniqueness speaks to ensuring no issues arise from permissions and filename collisions. Users should be alerted if they are possibly overwriting their own owned file, however, if a user is potentially overwriting a file they do not own the action should be disallowed.

- **Property 7: Physical system protections.** The physical system protections speaks to ensuring that the physical location of the server should be secure. Additionally, the machine itself should have a RAID-5 setup in order to mitigate the potential for data loss. Any offsite backups of the server should also be maintained in a secure environment.

Threat Model 2

Privacy and isolation of files must be ensured if users are to trust the file sharing system. Protections must be in place in order to ensure that the files themselves are valid and not threats to the system. Other restrictions are utilized in order to prevent unauthorized access or actions on files and folders. File and system integrity must be guaranteed in order to maintain trust in the system.

- **Property 1: Correctness.** Correctness implies that if file f is shared with members of group g, only members of group g should be able to access f. The notion of "access" entails the creation, modification, and deletion of f, as well as the ability to see that f even exists. Without this requirement, any user could access any file, which is contrary to the notion of group-based file sharing.

- **Property 2: Permission levels.** Permission levels that each user or group has for a file or folder determines what a particular user is able to perform on a file, such as read, write, delete, or create within the file structure. Without proper handling of permissions, unauthorized access or modification to files can occur.

- **Property 3: File integrity.** File integrity refers to keeping a checksum or CRC to ensure modifications or corruptions in a file can be tracked or detected. Users may refer to these checksums to maintain integrity of the file and assurance of correctness, enhancing trust in the system.

- **Property 4: File restrictions.** File restrictions can be implemented in methods such as storage quotas to prevent overuse of system by overactive users. By maintaining control over quantity of files maintains quality of service for the common use of the server by all users.

- **Property 5: User roles.** User roles refers to the powers granted to users on the system. By enforcing a hierarchy of ownership and ability to grant permissions (e.g. administrator, group owner, etc.), unintended or malicious modification or deletion of files is diminished.

- **Property 6: Download integrity.** Download integrity refers to the prevention of malicious files even existing on the file sharing system. By implementing a proven virus and malware detection solution, the server maintains protection against corruption from within. Malicious content shall be quarantined and removed from the system, and system administrators shall be made aware of such activity.

- **Property 7: Open verified software.** Open verified software implies that the software installed on the system will be open source and heavily validated and supported by the community in order to ensure both correctness and security. Open verified software provides transparency in source code and by extension software, allowing enhanced security over black box software.