Aufgabe 1: Rechnersicherheit

(a) Zugangskontrolle: Ein System überprüft für einen bestimmten Nutzer, ob dieser für die Benutzung von Betriebsmitteln des Systems authentisiert ist, also z.B. durch die Eingabe von Benutzername und Passwort oder auch durch eine Chipkarte erkannt werden. Damit soll verhindert werden,dass unautorisierte Personen mit dem System interagieren können.

Zugriffskontrolle: Wenn nun eine Person Zugang auf ein System erlangt hat, heißt es nicht automatisch, dass dieser auch alle Operationen darauf ausführen kann. Vor der Ausführung wird immer überprüft ob der Nutzer auch die Rechte besitzt eine bestimmte Operation auszuführen.Dabei gibt es verschiedene Modelle um den Zugriff zu kontrollieren, z.B. die Mandatory Access Control (MAC) wo jeder Nutzer einer bestimmten Schutzebenezugewiesen, diese hierarchisch angeordnet ist.

(b) Nein es ist nicht sinnvoll ein System mit Zugangskontrolle aber ohne Zugriffskontrolle zu gestalten, da nicht jeder Nutzer Zugriff auf alle Daten und Operationen haben soll. Ein Beispiel wäre wenn ein Kunde bei einem Bankautomaten Geld abheben will. Er benutzt seine Chipkarte um sich zu identifizieren und gibt eine PIN ein um bestätigen, dass ihm diese Karte auch gehört. Hätte das System nun keine Zugriffskontrollen implementiert, könnte der Nutzer z.B. sich die Daten von anderen Kunden ansehen und bearbeiten oder sein Konto mit Geldauffüllen, obwohl dieser nichts eingezahlt hat

(c) Wenn sich bei der Zugangskontrolle kein Nutzer identifiziert, kann das System bei der Zugriffskontrolle auch nicht bestimmen,welche Rechte dieser Nutzer nun hat, da dieser sich anonym angemeldet hat.

(d) Mit dem Share-this-Folder-Link wurden implizit die Rechte (Leserechte, aber keine Schreibrechte) eines Nutzers festgelegt. Damit weiß das System, wenn jemand auf den Link zugreift, dass dieser Nutzer eben nur diese Rechte haben soll.

Aufgabe 2: Timing-Attack

(1) Diese Methode ist anfällig für Timing-Angriffe, da die Bearbeitungszeit mit der Passwortlänge korreliert. Eine triviale Möglichkeit wäre einen zufallsgenerierten Integer zu erzeugen für jeden Schleifendurchlauf (also jedes Zeichen im Passwort) und dann wait() mit dem eben erzeugten Zufallsinteger auszuführen. Dadurch wird bei jedem Schleifendurchlauf eine zufällige Zeiteinheit lang gewartet, wodurch ein Timing-Angriff abgewehrt werden könnte.

(2) Einen Passwortüberprüfungsalgorithmus angenommen vergleicht der Angreifer die Bearbeitungszeit für die Überprüfung mit einer selbsterstellten Heuristik. Er wird dann Passwörter genau der Länge zuerst ausprobieren, die als aufgrund der Bearbeitungszeit als wahrscheinlich gilt.