Justin Anthony Timberlake
Midterm Exam
October 11th, 2017

Fall 2017 CS 436 Midterm (Group 1)

1. Customer Checkout Threat Model

System Description – A customer checkout line is an ongoing queue of customers who need to purchase goods with any given form of payment to include cash transactions, credit card usage, or the use of checks to pay for their groceries. In this system, a singular cashier is available at each PoS terminal, in front of which, the customers line up to be processed one by one. In the case of large grocery stores, multiple grocery items are placed on a conveyor, which cycles the goods towards the cashier, who scans the items as they reach him. After enough items have been scanned, the goods are placed in bags, which are then handled by the customer and placed in the shopping cart once more. Lastly, the intended payment method is used to pay for the groceries that have been processed by the cashier. A receipt is printed and the customer leaves with the bagged groceries he has paid for.

Assets and their importance – the cash register with cash inside (high), the groceries (high), the digital PoS scanner for bar code scanning (high), the payment method of the customer (such as checks / cards / cash)(high), computer database for groceries in the PoS terminal (high), the monitor read out for the cashier's use (medium), the credit card scanner readout (medium), the conveyor for the groceries (low), grocery bags (low)

Entry Points – the customer side of the counter, the cashier side of the counter, the cash register, the grocery conveyor belt (where groceries can be stolen or mishandled), the computer system (local network infiltration)

Attackers – External : robbers, hackers
Internal : malicious customers,, malicious cashiers, malicious staff members with access to the databases and computer systems of the store
Capabilities : Robbers can either attack customers for money or groceries either before or after check out by taking the shopping cart itself possibly. Robbers can also attack the cashier to retrieve cash from the register. Hackers can infiltrate the store's database or transaction logs, possibly stealing information from employees and customers alike. Internally, malicious customers can attempt to short cashiers some money owed (with cash) or write faulty checks. Customers can also (perhaps upon realizing they could not afford their goods) simply take their groceries right past the cashier without checking out, thus stealing the groceries. Malicious cashiers can take money from the register or give the customer faulty information. Other malicious staff members can also disrupt the store's local network and records database just as a hacker could.
Motivations : financial, information theft, malice

Vulnerabilities and Threats – theft of goods (by robbers or customers)(high), theft of funds (high), hacker infiltration to system (high), miscalculation of funds (high), PoS terminal breakdown (move to another register) (medium), error in pricing database

(medium), bar code scanner not functioning (medium), damage to PoS external hardware (such as monitors, keypad) (low), conveyor belt breakdown (low), etc.

Mitigation Strategies
Spoofing Identity – example : breaking into register to gain same access as cashier
Mitigation : Require cashier access log in and a system of locks on the register in particular.
Tampering with data – example : Altering data within the pricing databases
Mitigation : Require authorization before changes are made to the databases. Make regular checks and routine scans on all terminals for possible malware or other malicious software within the store's local network of machines.
Repudiation – example : denial of a particular transaction performed by the customer
Mitigation : Require a digital signature transactions, such as a PIN for card users at the PoS terminal. Print out receipts and keep a back up of all transactions within system logs so that there can be no mistake when a particular transaction was made.
Information Disclosure – example : someone stealing the cashier's log in password or information related to logging into sensitive information within the local network
Mitigation : Do not write down any cashier or system administrator passwords. Do not leave accounts with sensitive information logged on for long periods of time or if anyone steps away from his terminal. Encrypt all forms of authentication such as passwords and other sensitive data used to access the system database and transaction logs. Keep all sensitive data thoroughly hidden and inaccessible to any person who is not absolutely necessary to share with. Make all cashier and administrator access strictly on a need-to-know basis.
Denial of Service – example : PoS terminal software shutdown
Mitigation : Require authentication and authorization of all major changes in the sytem settings and run regular diagnostic, defragmentation, and virus scans on all machines with access to the store's local network connection. Disallow foreign pieces of software to run from external locations on the terminals and system administrator computers themselves.

2.
A) Shifting all characters by 8 positions in the alphabet,
where A … Z = 1 … 26 :
A => I, B => J, C => K, D => L, E => M, F => N, G => O, H => P, I => Q, J => R, K => S, L => T, M => U, N => V, O => W, P => X, Q => Y, R => Z, S => A, T => B, U => C, V => D, W => E, X => F, Y => G, Z => H
Therefore, the string reads :
Nwcz akwzm ivl amdmv gmiza iow wcz nibpmza jzwcopb nwzbp wv bpqa kwvbqvmvb i vme vibqwv kwvkmqdml qv Tqjmzbg ivl lmlqkibml bw bpm xzwxwaqbqwv bpib itt umv izm kzmibml mycit.

B) k = 5 intuitively because only 2 words of a single letter exist in the English language (A or I) and k would have to be greater than 10 or less than 0 to shift from "I" to "F".

Therefore, to decrypt, we shift the characters back by 5 positions in the alphabet, where A … Z = 1 … 26 :
A => V, B => W, C => X, D => Y, E => Z, F => A, G => B, H => C, I => D, J => E, K => F, L => G, M => H, N => I, O => J, P => K, Q => L, R => M, S => N, T => O, U => P, V => Q, W => R, X => S, Y => T, Z => U
Therefore, the string reads:
A stitch in time saves nine.

C) The one time pad reads :
kghnrtllhytkgvgxbxmlrykiyteirumgmbnbxjvd
The ASCII values of the one time pad are :
k (107) g (103) h (104) n (110) r (114) t (116) l (108) l (108) h (104) y (121) t (116) k (107) g (103) v (118) g (103) x (120) b (98) x (120) m (109) l (108) r (114) y (121) k (107) i (105) y (121) t (116) e (101) i (105) r (114) u (117) m (109) g (103) m (109) b (98) n (110) b (98) x (120) j (106) v (118) d (100)

The ASCII values of the given input string are :
p (112) l (108) e (101) a (97) s (115) e (101) c (99) a (97) l (108) l (108) a (97) t (116) f (102) i (105) v (118) e (101) o (111) c (99) l (108) o (111) c (99) k (107)

Add these together at each corresponding position in each string and take the remainder of their sum divided by 26 (modulus 26) and the resulting ASCII values are out of the range of letters on the ASCII chart.

(onetimepad + string)%26 =
(11), (3), (23),(25), (21), (9), (25), (23), (4), (21), (5), (15), (23), (15), (13), (13), (1), (11), (9), (11), (5), (20)

Finally, add 97 to the final result for each since all the characters in each string are in the lower case range of letters (97 is the ASCII value where the lower case alphabet starts, so ‘a’ = 97).

+97 =
l (108) d (100) x (120) z (122) v (118) j (106) z (122) x (120) e (101) v (118) f (102) p (112) x (120) p (112) n (110) n (110) b (98) l (108) j (106) l (108) f (102) u (117)
Therefore, the final string is : ldxzvjzxevfpxpnnbljlfu

3.
A) Justin Anthony Timberlake
md5 hash : a1bf97514812bacd49ec85ecf0cd76bc

B) i) Since Alice has access to the symmetric key, she can therefore use a means of encryption which will generate a Message Authentication Code (MAC) as well if and only if the encryption key is properly secured. However, any given regular symmetric

encryption key will not necessarily generate the MAC, so without one present, it is possible for Trudy to send Bob a message which he will decrypt and accept on his end, thinking it to be from Alice. This could be advantageous to Trudy, even if she does not know what the message will decrypt to on Bob's end, sadly.
ii) No, he cannot prove that. Even using the MAC from Alice, there is no way for he as the receiver to prove that it was in fact Alice who signed instead of himself. This is because in symmetric cryptography, the receiver has the same secret key. So, while the message can be authenticated, if and only if secure encryption and a MAC are used, symmetric encryption will never allow non-repudiation even with the usage of MACs.

4.
A) i) distributed denial of service attack
ii) Prevention strategy : She can use bandwidth management services and hardware to prevent malicious traffic from reaching the servers. The hardware itself will filter malicious traffic if it is built to be smart and dynamic. She can also block all unnecessary port access, including pings. Finally, she can use egress and ingress filtering, which prevents packet traffic between her and fake sources. Ingress filtering is enabled at the ISP level.

B) i) This is an attack, as it is not humanly possible to send so many queries so quickly as a regular user.
ii) To slow down this attack, Goofy can throttle traffic to some given source which is sending too many requests. This will immediately slow down the traffic, however, Goofy may also need to drop all traffic to the attack hosts to a more securely monitored node for further investigation and set up a firewall to monitor for denial of service attacks, which can blacklist ranges of IP addresses whenever an attack occurs. Finally, Goofy can require that users validate their "humanity" by having them solve some sort of puzzle such as CAPTCHA or some kind of picture selection puzzle, which would not theoretically be possible to complete as a botnet controlled computer for example, but would be intuitively obvious to an actual human user. Using said puzzles will make DDoS attacks not only more difficult to succeed, but also significantly more costly to the malicious attackers trying to use them.

5.
A) This is a terrible biometric based on the properties of good biometric systems :
Universality – Typically, people pull cords with their arms or legs. However, not everyone has all of their limbs still and not everyone has a way to pull a cord in the exact same way, which could be read on the same necessary scale as any other given employee. There would be no standard as to exactly how an employee could pull the cord without excluding some employees, which invalidates the universality of this biometric.
Distinctiveness – There is nothing distinct about the way someone pulls a cord a particular time. Two people can be evenly matched in strength, so many employees will not be able to be identified uniquely, thus invalidating the distinctiveness of this biometric.

Permanence – Every employee has days where they are not feeling as well or they are tired from a long weekend, for example. On these days, employees may not be at optimum strength and considering that an individual's strength changes over time (perhaps as an employee ages or maybe breaks an arm in an accident), the permanence of this biometric is thus invalidated.
Collectability – While it will be easy for most employees to pull a cord each morning when authenticating, it is not optimum to ask this of every employee every morning because pulling a cord with max effort is also tiring and uses energy which should be applied to working throughout the day instead of just signing in. This makes it hard to get good authentication readings for everyone, thus invalidating the collectability of this biometric.
Other issues – There could be employees who are too strong and break the given cord and also employees who find some other way to generate enough force, pulling on the cord by means of some mechanism for example, to spoof the identity of someone else's strength. Also, if an otherwise very weak person were to begin working out with weights, not only would they be able to eventually spoof the identity of several other stronger employees, but his own biometric would need to constantly be updated, making for an incredibly inefficient process for the entirety of the authentication system.

B) i) This password contains only lowercase letters and is short, therefore very easy to guess (length 4, so at most $2^4 = 16$ guesses to determine what the password is, likely fewer since there is no variation in types of characters and it is a very common sequence). There are also no special symbols, uppercase letters, or symbols, which can all add to the entropy of a password.
ii) Similar to the last password, while it does contain at least numbers and lower case letters, it is based on two common sequences. There is not enough seemingly random material in either of these first passwords as neither make the use of uppercase letters or special symbols.
iii) While this password looks good at first as it uses uppercase and lowercase letters along with a number, there are still no symbols present and what is worse, two words from the dictionary comprise most of the password, "Hello" and "World", which may be found out even faster based on pattern matching.
iv) This password has an excellent length, making it take longer to brute force and even uses a mix of uppercase, lowercase and numbers. While it could be further improved with the use of symbols, this password is very strong and difficult to guess. However, the main problem with this password is that it is also almost impossible for anybody to remember, thus will likely be written down somewhere, thus invalidating whatever security it does provide for the user.