Problem Set 1 Solutions

Problem 1-1. Distributing Crack (Grant Emery, Brian Perrin, Emil Sit)

In order to evaluate the two parallelization schemes we look at some general considerations:

- **Goals** — The goal of a traditional password cracker is to discover weak passwords in a password file, to allow system administrators to correct the situation. A malicious user generally needs to discover only one password which allows remote login in order to break into the target machine — once given a shell, many paths usually exist to a root compromise. For both of these types of users, efficiency is a primary concern. In general, we can try to optimize space and time efficiency. Generally this represents a tradeoff. Of course, these days, both computation and storage is cheap, so we can choose our optimization. We believe that the above goals would lead to a preference for time efficiency.
- **Effect of Salt** — Space efficiency is made a more difficult problem by the presence of a salt. The salt increases the number of possible encryptions for a given password. For example, the two byte salt used in the Unix password system uses alphanumeric characters, which give 64 possibilities for each byte of salt, increasing the password space by a factor of 4096. This directly affects the amount of space we would need to mount a straight dictionary look-up attack.
- **Straight Numbers** — Observe that both splitting algorithms do not affect the basic worst case analysis. Suppose we are using a w word dictionary over k machines to analyze p passwords having s distinct salts. If you split the password file, each machine has roughly s/k salts, and so it must perform ws/k encryptions in the worst case. If you split the dictionary, each machine now has s salts but w/k words. So you still need to calculate sw/k things. Both schemes are thus equally efficient in the worst case.

0.1 Splitting the Dictionary

If each machine has a section of the dictionary, it will be able to keep more versions of encryptions cached for each word. Scheme A offers us increased efficiency if we're going to be cracking large numbers of passwords, since the higher the number of passwords, the likelier it will be that two will have the same salt, and we'll be able to reuse cached results. The different permutations that modern cracking programs compute for each word doesn't significantly affect the problem, as it basically just makes the dictionary larger. The only disadvantage to splitting the dictionaries is that some overhead (coordination) is required amongst the machines so that when one successfully cracks a password, the others stop working on it.

0.2 Splitting the Password File

Not all words in the dictionary are equally likely to be used as passwords, so that on a split of the dictionary, there could be a machine that is checking words that all have a small chance of being used. If we split the password file instead, it is possible to associate the words in the dictionary with a probability so that the cracker program would check all of the more likely words first, thus having a higher chance of getting a crack quickly. It is possible to optimize a dictionary split so that this advantage is applied to that scheme as well, but it takes much more setup time to prepare. Intelligent splitting of the password file can group all passwords with the same salt onto the same computer, so that the computation and caching benefit from splitting the dictionary is largely negated.

0.3 Conclusion

Splitting the password file has an advantage if someone is looking to have a high chance of getting a fast crack, without worrying about completeness. This would be advantageous to someone like a hacker. However, splitting the dictionary is better for complete coverage, if you wanted to check every entry in a password file, as a sys admin might.

Problem 1-2. Motor Control Biometrics (Nick Estrada, Nick Feamster, Michael Freedman)

Alice's claim that Bob's path and timing as he drags the mouse over ten points on the screen is sufficient information for biometric authentication is questionable for a number of reasons.

First, Bob's motion of the mouse may not be all that distinguishable from the way anyone else might move the mouse. More or less, a typical user will move the mouse between points in essentially a straight line fashion (if we allow for the fact that a user might purposefully insert non-straight line type motion, then the motion is being used as a type of password, as opposed to just biometric authentication). As such, the range for possible timing and path templates is extremely small, and the crossover between false positives and false negatives could potentially be quite high.

Second, let's assume that Bob will want to authenticate himself to more than one machine using this system. Bob's timing may vary if the tracking on the mouse differs across machines, or if the graphic which contains the ten points exhibits a different scale on different machines. This problem makes it rather difficult to make Θ constrained enough to reduce false positives while still ensuring that the false negative rate remains tolerably low. As such, we can expect the failure crossover rate for Alice's scheme to be intolerably high.

Third, even on the same machine, Bob's mouse usage can vary significantly, depending on his state when using the computer: how tired, how sweaty, how nervous or relaxed, how greasy his hands, if he always uses the same chair/work terminal that results in a specific posture, how dirty the mouse ball. These are only some of the factors that affect his consistency between login attempts. Similarly to the previous point, an reasonable threshold could result in an unacceptably high false positive rate.

Given the similar fashions in which users are likely to connect *ordered* points (requiring strict template matching) and Bob's inconsistencies (requiring less string template matching), the crossover rate is unacceptably high.

Alternatives and Improvements

An improvement on Alice's method is to display N dots that are **not** numbered. Users will generally trace the dots in the same pattern time and time again. Still, there will be a probabilistic distribution that makes some ordering more likely than others. So, while an improvement, we might face still-high crossover rates. This scheme obviously breaks down if an attacker sees the pattern his victim generally uses.

An alternative scheme involving only standard computer hardware might involve testing reflexes (the terminal displays a random sequence of keys one-at-a-time (such as "arrow" combinations) and through some basic learning algorithms, figures out what a user's normal response rate is. Requiring users to type phrases or words is also viable: a user will "memorize" how to type the same login phrase; depending upon the typing of the user, response rates to different characters alone or in sequence may vary dramatically.

A possible method to eliminate false positives, without the user even knowing of such a scheme, is to detect the method by which they initially access the terminal (i.e. to get rid of the "screen saver" to bring up the login prompt, does a user click the mouse, hit a key, move the mouse, etc.).

Note: With the continued spread of microphones and/or cameras as hardware comprising the normal composition of computer systems, we can begin to plan on voice recognition, face recognition, or similar technologies, as biometric authentication. The cross-system availability of these superior techniques, however, is not yet certain.

More Improvements: Persistence of Authentication

To enable persistence of authentication, we also suggest the following metrics:

- **Work patterns:** how often the user switches applications, time between rest breaks, etc.
- **Typing patterns:** the speed of between keystrokes, how quickly a user corrects mistakes and the shortcuts used to delete words.
- **Usage patterns:** what is performed at initial login, how is the mouse or keyboard used (For example, does the user most often type `xterm &` to open a new window, or does the user do this by clicking on the root window and selecting a "Make New Window" option?) While such a persistence test requires machine learning for accuracy, a user profile (and thus the validity of such information) strengthens over time.

In short, biometric authentication utilizing normal hardware cannot adequately ensure both low false positives and negatives. For acceptable results, specific biometric authentication hardware (fingerprint card, iris scanner, voice recognizer, etc) is a necessity.

Problem 1-3. eBay Security Policy (Chiuyi Chan, Ying Lee, Ding Meng, Tairan Wang)

eBay is an on-line auction market place which allows sellers to post their goods on the page, and interested buyers can bid on the goods. eBay then charges the sellers a fee on posting their items, and if the transaction is completed, a Final Value charge is imposed on sellers based on the amount sold.

- What security policies do buyers/sellers (users) want?
  - Privacy: eBay should not track buyers' buying habits or users' selling habits. Also, personal information should not be disclosed to third parties. (e.g. Users' email and address information should not be distributed for advertising purposes.)
  - Security: A good authentication system is required so that no attackers can impersonate a user. All the information users submit through the web should be well protected. Since sellers send out their credit card information for payment, a system to encrypt this information is essential.
  - Fair auction: Sellers have to sell to the bidder offering the highest price. All bidders should bid with the sole interest of acquiring that particular item, instead of pushing up the item's price. Therefore, it is very important to make sure that nobody can have multiple accounts, and that sellers are prohibited from bidding on goods they posted. Of course, we have no way of monitoring communication between users. Similarly, bidders have to pay what they bid for.
  - Right transaction: like most of the other on-line transactions, buyers should get the right goods before they are supposed to remit payment. On the other hand, buyers should be certain to pay once they receive the right goods.
  - Integrity of other members of the community: Ideally, users want eBay to monitor other users' behavior and make note of users not conforming to the rules. (e.g. forbid them from using the site if they commit serious fraud)
- What security policies does eBay want?
  - Security: Auction site has to be able to authenticate users. Normal security concerns for any computer system: protecting database from attack by outsiders or registered users who already log into the system, maintaining availability, and protecting important information transmitted through the web, such as credit card information.
  - Transaction records: transaction records of all users should be kept to facilitate monitoring.
  - Identification of users: in order to maintain the reputation of the trading community, it is important for the auction site to have access to crucial information about their users. However, currently one can start auctioning on eBay by making up all the information during registration. This may be because eBay wants to stay away from the responsibility of enforcing good transactions. Obtaining accurate identification of users also help to prevent multiple accounts of one user.
  - Monitoring users' behavior: find out which users are good and which are bad. Keep users informed of their own risk of trading with users on the other end. Currently, a rating system is being used for users to rate each other. By doing so, eBay can again stay away from taking up the responsibilities of bad transactions. The rating system should be well protected from break-ins and bogus entries. The number of votes a particular user can cast on another user should be strictly correlated to their transaction records in the past.
  - Reliable backup mechanism: in case of site crashes, power interruption or other situations which lead to temporary shut down of the auction site, they should be able to fix the problem and restore everything within a reasonable short period of time.