CS396: Security, Privacy and Society (Spring 2024)
Homework #3

Instructions

Please carefully read the following guidelines on how to complete and submit your solutions.

1. The homework is **due on Monday, April 22, 2024, at 11:59pm**. Late submissions are accepted subject with a penalty of 15 points per 12 hours late (rounded down). Starting early always helps!
2. Solutions are accepted only via Canvas, where all relevant files should be submitted **as a single .zip archive**. This should include your typed answers **as a .pdf file** and **the source code** of any programming possibly used in your solutions.
3. If asked, you should be able to explain details in your source code (e.g., related to the design of your program and its implementation).
4. You are bound by the Stevens Honor System. You may use any sources related to course materials, but **information from external sources must be properly cited**. Your submission acknowledges that you have abided by this policy.

Problem 1: Password Cracking! (30%)

Hooray! You have compromised an online server and stolen their list of stored passwords along with the associated usernames. Time to break into those accounts. One problem though: once you open the password file, you discover that only the password *hashes* were stored. Guess more security controls were in place than you thought. Let's see if we can recover any passwords anyway!

(1) You figure that some passwords are more common than others. Possibly there are some users who chose very common passwords. And luckily for you, there are many password lists available. The RockYou wordlist is one such password list, consisting of passwords leaked in a data breach. The entire wordlist is huge; over 14 million unique passwords. Let's start by just downloading the top 2000 words. You can find them in the file rockyou2000.txt under the Canvas course module labelled Password Cracking Code.

(a) You don't know what hash function the target passwords were hashed with, but you're guessing it's a common one. First, hash every password of rockyou2000.txt using md5 (an insecure hash, but perhaps the server used it anyway). Submit the results in a .txt file rockyou2000md5.txt in the form of **plaintext password: resulting hash in hex format** with one such pair per line.

Also submit the code or script you wrote to do this. Note: you don't need to write the md5 hash function itself; many popular programming languages have their own built-in cryptographic libraries that can do this for you (e.g. hashlib in Python).

(b) Do the same thing using the sha256 hash function. Again submit both the resulting file rockyou2000sha256.txt and the corresponding code or script.

(2) Time to see if any of our precomputed hashes are a match! Download the file of leaked password hashes: leakedhashes.txt in the Canvas module. Check if any of your precomputed hashes matches the hashes of the users. Did you get any hits? Submit a file listing the **user: plaintext password** pairs of all passwords you cracked using this method (note: it will not be all passwords).

(3) What security controls could have been implemented to help mitigate this attack? Name and describe the potential mitigation impact of at least two distinct controls.

(4) There are many password cracking tools available to level up our skills.Take a look at Hashcat here and read this article. Then list and describe the different kinds of attacks given in the article.

Problem 2: Understanding NSEC5 (35%)

Introduced by Goldberg *et al.* and currently under IETF's consideration for standardization, *NSEC5* comprises a replacement of the NSEC3 protocol for removing an identified vulnerability. Read the Introduction section from the original paper, and answer the following questions.

(1) What *type of leakage* does NSEC3 allow for and what is the *technical reason* behind such vulnerability?

(2) What are the core *technical features* of the new cryptographic hash function used by NSEC5 that allows for correct verification of non-existing domain names, yet preventing the type of leakage NSEC3 allows?

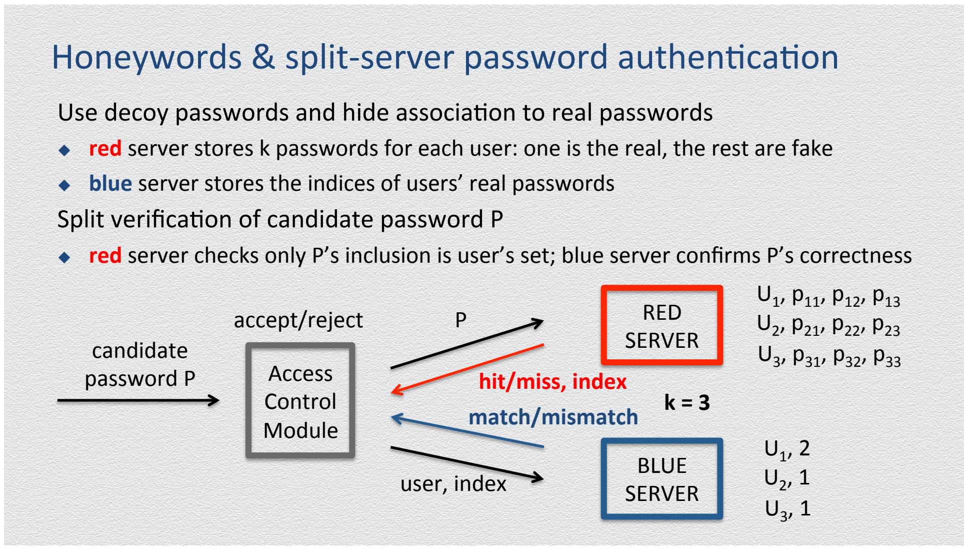
(3) NSEC5 requires distributing a *signing key* to any secondary DNS resolver, who by definition is assumed to be *untrusted* with respect to validity of the provided answers for non-existing domain names. How is this technical choice justified?

(4) In terms of *secure system design*, what is the lesson learnt in view of what justified the development of protocols DNSSEC, NSEC, NSEC3 and NSEC5?

Problem 3: Intrusion detection (35%)

Introduced by Juels and Rivest, *Honeywords* comprise a non-cryptographic method for hardening password security against stolen password files after successful breaches into authentication servers. The idea is to employ *decoy* passwords so that any user's account is associated with not only one (the real) password, but also with many fake ones, and then distribute password verification *across two servers*, say one red and one blue, each storing and verifying "half" of the credentials needed to be verified in order to successfully authenticate a user (see also Figure 1). Read about the Honeywords authentication system from the original paper, and answer the following questions.

(1) How does this *split architecture* improve security? Consider the two cases where an attacker compromises only one of the servers.



Honeywords & split-server password authentication

Use decoy passwords and hide association to real passwords

- red server stores k passwords for each user: one is the real, the rest are fake
- blue server stores the indices of users' real passwords

Split verification of candidate password P

- red server checks only P's inclusion is user's set; blue server confirms P's correctness

Figure 1: Hardening password security by employing decoy passwords in a split-server architecture.

(2) How does this system make password cracking *detectable*.

(3) What constitutes a *good* honeyword for a user whose real password is `Bo$tonRedSox76`, if honeywords are generated by tweaking real passwords? Provide a few examples.

(4) Stevens employs the Honeywords authentication system, and you just stole the passwords list, `0000000000000000000000000`, `itWb!%s45_3gMoI00286!*mooewTi409##21jUi`, `7304ASpaceOdyssey`, and `2001ASpaceOdyssey`, of an employee in the Office of the Registrar. If you have *only one* chance to impersonate them (and try to increase your GPA), which password will you choose and why?