University Of Washington, CSE 590P – Computer Security - Homework 2

Tadayoshi Kohno, John Manferdelli

Due: 4:30pm January 25, 2007.

See the course website (http://www.cs.washington.edu/education/courses/csep590b/07wi/) for instructions on how to submit your homework via the UW Catalyst Tools. For this assignment, you should submit a PDF file named 'YourLastName-YourFirstInitial-HW2.pdf. Please write your name on the first page. The entire assignment is worth 26 points and there is an opportunity for extra credit.

1. Authentication (10, subparts have roughly equal value): Passwords are the only authentication mechanism we've discussed in detail so far. Recently, Bruce Schneier reported on password usage patterns in connection with the MySpace phishing attacks (http://www.schneier.com/crypto-gram-0701.html#9). Estimate the following:
    a. The number of 8 letter passwords if all characters are lower case letters.
    b. The number of 8 letter passwords if all characters are upper or lower case letters.
    c. The number of 8 letter passwords if all 90 characters directly accessible on my HP Keyboard can be used.
    d. The number of passwords consisting of one of 100,000 common words and names with three extra characters (any of the 90 available on the standard keyboard) appended.
    e. Schneier estimates that a single machine can test 200,000 passwords/second. How long would an online attack take for each of (a)-(d) above.
    f. Suppose an offline attack against a password is possible by pre-computing all possible hashed passwords and doing a binary search (suppose passwords can be compared against a cryptographic hash of a known design whose length is 160 bits). How big a "lookup table" is needed in each of the above cases? [Hint: you are comparing hashes against hashes; the only point of the "known hash" algorithm is that it in known to the off-line attacker]
    g. Recalculate the cost of an offline "dictionary" against password files from many sources now that a random "seed" of 32 bits is used in the "hashed" password.
2. Authorization (10, subparts have roughly equal value): This problem involves resources on a single shared computer at PreviousGen.com.
    a. You login is "dmr" and you are a member of the following groups: cryptographers, system-programmers, bell-alumni. Consider the following ACLs and security ids listed Appendix A. Files you can open directly for reading and writing by compliantly calling the access control system and abiding by the result (online only). The Access control system is pretty "vanilla": (username, password) pairs are stored in a file (as salted cryptographic hashes) called /etc/passwd and ACL's are accessible only by the OS. What files can you access when logged in as "dmr" and why?
    b. Now assume that members of "system-programmers" can remove themselves from any group that they are in and can add people to any group. Can you read any additional files now? If so, why?
    c. I'm a system administrator and accidentally delete ken's read ACE on the file "News-from-Bell-Labs" at 03:00 and restore it at 11:00. This is a sort of revocation. Draw several alternative "timelines" (beginning at 2:00 and ending at 12:00) showing ken's possible access to this file in light of my accidental revocation. Unlike dmr, ken will abide by access control decisions. How would this timeline change if we used capabilities instead of ACLs?
    d. Once again you are dmr. You are insanely jealous of ken who is about to send off his paper on how to break AES to Eurocrypt 2008. All the copies of ken's paper are on the computer system and are mentioned in the above ACLs. Can you stop ken? (The real ken would have been smart enough to store copies of the paper elsewhere). What conclusions can you draw?
    e. Dmr finds out that ken won't send the paper to Eurocrypt until tomorrow. Suggest a way for dmr to get a copy of the paper and pass it off as his own work.

f. Extra credit: Feel free to comment on the security policy adopted by the users of this computer

3. Access Control (6, each worth 3):
    a. Bell Labs is a bureaucratic place. They require all pre-publication papers be stored on an NFS file server (consult Appendix B for how access control decisions are made on NFS) and a "trusted" process takes a cryptographic hash of all these files regularly and stores the hashes to insure researchers don't do the things dmr did in 3d and 3e. Obviously, researchers don't want even management to see advanced copies of their papers so ONLY the cryptographic hashes are stored and the associated user ids. Ken and dmr have their own machines which are connected via an Ethernet connection to the NFS server. Discuss dmr's chances of mounting a denial of service attack on poor ken (ken only keeps copies of his papers on the NFS server) or "stealing credit" for the paper.
    b. You work for Microsoft and you need to be certain that, despite his excellent reputation, you must revoke all of jmanfer's modify permissions to the Windows source files on the server "rocksolid" by 7 PM. How can you do it?
4. Covert Channels (Extra Credit): John's friend Brian wants to transmit key material to John. Unfortunately he can only communicate with John on the super-secure computer and any process of Brian's with access to keying material is monitored so that any bits from the keys written to a file or sent via any output mechanism from those processes is recorded. How can Brian transmit keys to John without getting caught? The penalty for transmitting keying material is 10 years in prison so Brian is well motivated to be careful.

Reading

Gollmann: Chapters 3, 4, 5. If you read them before you attempt the problems, it may make motivate the HW and make it a little easier to answer some of the questions.

Appendix A – ACL Problem Data

Login	SID	Login	SID
Root	S-1-5-18	Everyone	S-1-1-0
Bell-Alumni	S-1-5-182	Backup-operators	S-1-5-116
Cryptographers	S-1-5-183	jlm	S-1-5-21
ken	S-1-5-19	pjw	S-1-5-122
dmr	S-1-5-20	jmg	S-1-5-123
System-programmers	S-1-5-111	bal	S-1-5-27

Group Membership

Bell-Alumni : ken, dmr, jlm, pjw, jmg, bal

Cryptographers: jlm, pjw, bal

System-programmers: ken, dmr, jlm, bal

Backup-operators: jmg.

Everyone group includes any user subject that can "log-in".

Acls (the arrows are to remind you they are access control *lists*)

/etc/password. Owner: S-1-5-18

Read: **S-1-5-111** → S-1-5-116

Write: DENY, **S-1-5-111** → S-1-5-116

/papers/Eurocrypt07.pdf. Owner: S-1-5-20

Read: S-1-5-20 → S-1-5-21 → S-1-5-27

Write: S-1-5-20

/dmr/myaddresses.txt. Owner: S-1-5-20

Read: S-1-5-20 → S-1-1-0

Write: S-1-5-20

/dmr/Todo.txt. Owner: S-1-5-20

Read: S-1-5-20 → S-1-1-0

Write: S-1-5-20

/ken/todo. Owner: S-1-5-19

Read: S-1-5-19

Write: S-1-5-19

/jlm/p590/homeworkAnswers.pdf. Owner: S-1-5-21

Read: S-1-1-0

Write: S-1-5-21

/bal/SecretAgreement. Owner: S-1-5-27

Read: S-1-5-27

Write: S-1-5-27

/pjw/Fortran77project.pdf. Owner: S-1-5-122

Read: S-1-5-122

Write: S-1-5-122

/jmg/bestparserideas.pdf. Owner: S-1-5-27

Read: S-1-1-0

Write: : S-1-5-27

/bal/aes.c. Owner: S-1-5-27

Read: S-1-1-0

Write: S-1-5-27

/jlm/dmriswrongaboutaes.pdf. Owner: S-1-5-21

Read: S-1-1-0

Write: S-1-5-21

/ken/myaeswork.pdf. Owner: S-1-5-19

Read: S-1-5-19

Write: S-1-5-19

/AlumniNewsFromBell/September2006.pdf. Owner: S-1-5-122

Read: S-1-5-19 → S-1-5-21→ S-1-5-27→ S-1-5-122

Write: S-1-5-122

Appendix B - Brief description of NFS file sharing (idealized)

Each Client machine maintains a map from NFS mounted file systems including a map to the NFS server. A daemon uses the mapping information to send open request to the server. These requests consist of the client id, user-id as supplied by the client, and marshaled parameters. If the open is successful, the NFS Server returns a handle that serves as a parameter to future read, write and close calls. Administrators on each client and each server must create files naming (in the case of the clients) the servers it is willing to trust for NFS services and (in the case of servers) which clients it is willing to serve.

Access control decisions by the server are handled as follows: the NFS daemon forwards the request package to the access control system which uses the parameters (user-id, filename, etc) to perform the same access check as if the user was local. The underlying UID system and access control system is not modified at all.

Once a handle is created for reading or writing, subsequent read or write calls from approved clients are forwarded. In the case of read, data read is read into the server memory, transmitted by the daemon to the client daemon and copied into the user address space as a normal read would.

NFS was one of the first networked file systems for “microcomputers” and is still used today (with some modifications).