Handout 2
Articles for Oral Reports in CompSci 725

v1.0 of 24 July 2014

Clark Thomborson
Andrew Colarik

July 24, 2014

A significant fraction (15%) of your marks in CompSci 725 will be awarded for your performance of an oral presentation during a lecture period, during weeks 6 to 12 of the semester. Your presentation will be based on your careful reading and analysis of a professional publication that appears on the list at the end of this handout.

Midnight on Monday 28 July. All students should send an email to Andrew (`mailto:a.colarik@auckland.ac.nz`), indicating their first, second, and third choice of article for their oral presentation. Technical articles can (usually) be uniquely identified by the surname of the first author and the year of publication. You should use this format when transmitting your preference list to Andrew. For example, your preference list might be "Forrest 1996, Burrows 1990, Birrell 1985".

Andrew will digest your emails on Tuesday, starting from the ones he received closest to midnight the previous evening.

No article will be presented more than three times by students in this course. Andrew will communicate with you by email if your preferred articles are unavailable.

Before lecture on Wednesday 30 July, Andrew will webpost the first draft of an "Articles to be Presented" handout. This handout will fix the date of your oral presentation. If you have a problem with this date, please let him know as soon as possible.

References

[1] A. Birrell, "Secure communication using remote procedure calls," *ACM Trans. Comput. Syst.*, pp. 1–14, 1985. [Online]. Available: http://dx.doi.org.exproxy.auckland.ac.nz/10.1145/214451.214452

Abstract. Research on encryption-based secure communication protocols has reached a stage where it is feasible to construct end-to-end secure protocols. The design of such a protocol, built as part of a remote procedure call package, is described. The security abstraction presented to users of the package, the authentication mechanisms, and the protocol for encrypting and verifying remote calls are also described.

[2] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, Feb. 1990. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1145/77648.77649

Abstract. Authentication protocols are the basis of security in many distributed systems, and it is therefore essential to ensure that these protocols function correctly. Unfortunately, their design has been extremely error prone. Most of the protocols found in the literature contain redundancies or security flaws. A simple logic has allowed us to describe the beliefs of trustworthy parties involved in authentication protocols and the evolution of these beliefs as a consequence of communication. We have been able to explain a variety of authentication protocols formally, to discover subtleties and errors in them, and to suggest improvements. In this paper we present the logic and then give the results of our analysis of four published protocols, chosen either because of their practical importance or because they serve to illustrate our method.

[3] D. E. Denning, "A lattice model of secure information flow," *Commun. ACM*, vol. 19, no. 5, pp. 236–243, May 1976. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1145/360051.360056

Abstract. This paper investigates mechanisms that guarantee secure information flow in a computer system. These mechanisms are examined within a mathematical framework suitable for formulating the requirements of secure information flow among security classes. The central component of the model is a lattice structure derived from the security classes and justified by the semantics of information flow. The lattice properties permit concise formulations of the security requirements of different existing systems and facilitate the construction of mechanisms that enforce security. The model provides a unifying view of all systems that restrict information flow, enables a classification of them according to security objectives, and suggests some new approaches. It also leads to the

construction of automatic program certification mechanisms for verifying the secure flow of information through a program.

[4] D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls," *CoRR*, vol. abs/0903.2171, 2009. [Online]. Available: http://arxiv.org/abs/0903.2171

Abstract. While Mandatory Access Controls (MAC) are appropriate for multilevel secure military applications, Discretionary Access Controls (DAC) are often perceived as meeting the security processing needs of industry and civilian government. This paper argues that reliance on DAC as the principal method of access control is unfounded and inappropriate for many commercial and civilian government organizations. The paper describes a type of non-discretionary access control - role-based access control (RBAC) - that is more central to the secure processing needs of non-military systems than DAC.

Reprinted to arXiv by author Kuhn from *Proc. 15th NIST-NSA National (USA) Computer Security Conf.*, 1992, pp. 554-63.

[5] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff, "A sense of self for Unix processes," in *Proc. 1996 IEEE Symp. on Security and Privacy*, May 1996, pp. 120–128. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1109/SECPRI.1996.502675

Abstract. A method for anomaly detection is introduced in which "normal" is defined by short-range correlations in a process' system calls. Initial experiments suggest that the definition is stable during normal behaviour for standard UNIX programs. Further; it is able to detect several common intrusions involving `sendmail` and `lpr`. This work is part of a research program aimed at building computer security systems that incorporate the mechanisms and algorithms used by natural immune systems.

[6] L. Kagal, T. Finin, and A. Joshi, "Trust-based security in pervasive computing environments," *Computer*, vol. 34, no. 12, pp. 154–157, Dec 2001. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1109/2.970591

Abstract. Traditionally, stand-alone computers and small networks rely on user authentication and access control to provide security. These physical methods use system-based

controls to verify the identity of a person or process, explicitly enabling or restricting the ability to use, change, or view a computer resource. However, these strategies are inadequate for the increased flexibility that distributed networks such as the Internet and pervasive computing environments require because such systems lack central control and their users are not all predetermined. Mobile users expect to access locally hosted resources and services anytime and anywhere, leading to serious security risks and access control problems. We propose a solution based on trust management that involves developing a security policy, assigning credentials to entities, verifying that the credentials fulfill the policy, delegating trust to third parties, and reasoning about users' access rights. This architecture is generally applicable to distributed systems but geared toward pervasive computing environments.

[7] B. W. Lampson, "Protection," *SIGOPS Oper. Syst. Rev.*, vol. 8, no. 1, pp. 18–24, Jan. 1974. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1145/775265.775268

Abstract. Abstract models are given which reflect the properties of most existing mechanisms for enforcing protection or access control, together with some possible implementations. The properties of existing systems are explicated in terms of the model and implementations.

[8] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, "A taxonomy of computer program security flaws," *ACM Comput. Surv.*, vol. 26, no. 3, pp. 211–254, Sept. 1994. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1145/185403.185412

Abstract. An organized record of actual flaws can be useful to computer system designers, programmers, analysts, administrators, and users. This survey provides a taxonomy for computer program security flaws, with an Appendix that documents 50 actual security flaws. These flaws have all been described previously in the open literature, but in widely separated places. For those new to the field of computer security, they provide a good introduction to the characteristics of security flaws and how they can arise. Because these flaws were not randomly selected from a valid statistical sample of such flaws, we make no strong claims concerning the likely distribution of actual security flaws within the taxonomy. However, this method of organizing security flaw data can help those

who have custody of more representative samples to organize them and to focus their efforts to remove and, eventually, to prevent the introduction of security flaws.

[9] D. Mazières, M. Kaminsky, M. F. Kaashoek, and E. Witchel, "Separating key management from file system security," *SIGOPS Oper. Syst. Rev.*, vol. 33, no. 5, pp. 124–139, Dec. 1999. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1145/319344.319160

Abstract. No secure network file system has ever grown to span the Internet. Existing systems all lack adequate key management for security at a global scale. Given the diversity of the Internet, any particular mechanism a file system employs to manage keys will fail to support many types of use.We propose separating key management from file system security, letting the world share a single global file system no matter how individuals manage keys. We present SFS, a secure file system that avoids internal key management. While other file systems need key management to map file names to encryption keys, SFS file names effectively contain public keys, making them self-certifying pathnames. Key management in SFS occurs outside of the file system, in whatever procedure users choose to generate file names. Self-certifying pathnames free SFS clients from any notion of administrative realm, making inter-realm file sharing trivial. They let users authenticate servers through a number of different techniques. The file namespace doubles as a key certification namespace, so that people can realize many key management schemes using only standard file utilities. Finally, with self-certifying pathnames, people can bootstrap one key management mechanism using another. These properties make SFS more versatile than any file system with built-in key management.

[10] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Commun. ACM*, vol. 21, no. 12, pp. 993–999, Dec. 1978. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1145/359657.359659

Abstract. Use of encryption to achieve authenticated communication in computer networks is discussed. Example protocols are presented for the establishment of authenticated connections, for the management of authenticated mail, and for signature verification and document integrity guarantee.

Both conventional and public-key encryption algorithms are considered as the basis for protocols.

[11] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978. [Online]. Available: http://dx.doi.org.ezproxy.auckland.ac.nz/10.1145/359340.359342

Abstract. An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences: (1) Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intented recipient. Only he can decipher the message, since only he knows the corresponding decryption key. (2) A message can be "signed" using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in "electronic mail" and "electronic funds transfer" systems. A message is encrypted by representing it as a number M, raising M to a publicly specified power e, and then taking the remainder when the result is divided by the publicly specified product, n, of two large secret primer numbers p and q. Decryption is similar; only a different, secret, power d is used, where $e * d \equiv 1 \pmod{(p-1)*(q-1)}$. The security of the system rests in part on the difficulty of factoring the published divisor, n.

[12] J. G. Steiner, B. C. Neuman, and J. I. Schiller, "Kerberos: An authentication service for open network systems," in *Proc. Winter 1988 Usenix Conference*, Feb. 1988. [Online]. Available: ftp://athena-dist.mit.edu/pub/kerberos/doc/usenix.PS

Abstract. In an open network computing environment, a workstation cannot be trusted to identify its users correctly to network services. *Kerberos* provides an alternative approach whereby a trusted third-party authentication service is used to verify users' identities. This paper gives an overview of the *Kerberos* authentication model as implemented for MIT's Project Athena. It describes the protocols used by clients, servers, and *Kerberos* to achieve authentication. It also describes the management and replication of the database required. The views of *Kerberos* as seen by the user, program-

mer, and administrator are described. Finally, the role of *Kerberos* in the larger Athena picture is given, along with a list of applications that presently use *Kerberos* for user authentication. We describe the addition of *Kerberos* authentication to the Sun Network File System as a case study for integrating *Kerberos* with an existing application.

Note: a PDF version of this article was available on 24 July 2014 at http://research.cs.wisc.edu/areas/os/Qual/papers/kerberos.pdf.