Hardest Attackers

Hanne Riis Nielson[1] Flemming Nielson[2]

Abstract. Most interesting properties of computer systems are inherently undecidable yet static analysis allows to automatically validate systems against such properties by making sure to always "err on the safe side". We take this idea one step further by identifying "hardest attackers" in such a way that if the static analysis is able to demonstrate protection against the "hardest attacker" then the system is manifestly protected against all of the infintely many attackers.

The overall approach. Consider a computer system that is to be protected against hostile elements in its execution environment. This may be done by restricting access to key components to agents knowing a set of agreed passwords or to agents knowing a set of secret keys (assuming classical symmetric cryptography). Determining whether or not the system fulfills its intentions is usually an undecidable property — intuitively because *(i)* there is an infinity of attackers that do not know the passwords or secret keys, and *(ii)* even in the presence of a single attacker the success of penetration somehow involves the halting problem and therefore is likely to be undecidable.

Concerning *(ii)*, the only way to "by-pass" undecidability is to settle for an approximative answer that always "errs on the safe side" and that can be computed effectively. This is the avenue successfully taken by a number of approaches to the static analysis of programs: type systems, data flow analysis, constraint based analysis etc. Semantic soundness ensures that when a certain behaviour is guarded against according to the analysis then indeed it is guarded against according to the semantics. It goes without saying that there is no notion of semantic completeness; but as demonstrated by the widespread use of type systems and data flow analysis in compilers this

[1] On leave from Aarhus University, Denmark. Visiting *Universität des Saarlandes, FR 6.2 Informatik, Im Stadtwald, Bau 45 (Zimmer 417), Postfach 15 11 50, D-66041 Saarbrücken, Germany.* E-mail: `hrn@daimi.au.dk`.

[2] On leave from Aarhus University, Denmark. Visiting *Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85 (Zimmer 605), D-66123 Saarbrcken, Germany.* E-mail: `fn@daimi.au.dk`.

needs not be a hindrance if the static analyses offer "good" results on "well-behaved" programs.

Concerning *(i)*, to overcome the problem of the infinity of attackers we propose a novel use of the intrinsic operation of the static analysis so as to identify a "hardest attacker" that is as hard to protect against (from the point of view of the analysis) as any of the infinity of attackers. Thus if the analysis is able to demonstrate that the "hardest attacker" cannot penetrate then it is guaranteed that none of the infinitely many attackers will be able to penetrate either. This approach is in analogy with identifying problems that are hard for a given complexity class; indeed the term "hardest attacker" is more appropriate than "complete attacker" because the "hardest attacker" is identified with respect to the static analysis rather than the semantics and thus might not be a successful attacker with respect to the semantics; this is just another facet of static analysis always "erring on the safe side".

A worked example. To illustrate the viability of the approach suggested above we have validated the protectiveness of a firewall expressed using the communication-free fragment of the mobile ambients calculus [4].

The ambient calculus is based on the π-calculus [7] but allows more directly to model the mobility of computation; as in the spi-calculus [1] it allows to model passwords and secret keys that cannot be guessed by brute force attack (modelled using the restriction operator, ν). As an example, a message M, encrypted under the key K, is represented simply as the ambient $K[M]$ and the attempt at decrypting such a message is represented by the ambient $\mathsf{open}\,K$. If K is a secret only known to the principals $P_1, \cdots, P_n$ the entire system is represented as $(\nu\, K)(P_1 \mid \cdots \mid P_n)$ where each P_i may contain $K[M]$ as well as $\mathsf{open}\,K$.

The firewall introduced in [4] is a system intended only to allow agents knowing a set of agreed passwords to enter. Indeed, it is shown in [4] that all agents using the passwords in a specific manner will indeed succeed in entering. Our aims in [10] were complementary: to validate the "protectiveness" of the firewall against all attackers not knowing any of the required passwords.

To this end we first developed a static analysis suitable for the mobile ambients; to be more specific, this amounted to transferring the notion of control flow analysis for object-oriented and functional programs (see e.g. [9]) to the mobile ambients. This was somewhat more complex than the developments performed for the π-calculus [2] and was facilitated by the use of the flow logic style of specification [8] (also used in [2]); naturally we established the by now standard semantic correctness and model intersection results.

The novel aspect of the development was to inspect the intrinsic operation of the control flow analysis and to develop a formula for the "hardest attacker". The result can be compared with the approaches of the "Dolev-Yao tradition" [5], including [3, 11, 6], although the latter intend to operate at the semantic level rather than at the level of the analysis. To be more specific, the analyses of the "Dolev-Yao tradition" amount to an "informal" analysis of the capabilities of the attacker, leading to an inductively defined behaviour. However, such an analysis is not nearly as straightforward when the computational capabilities are modified, such as when adding the novel notion of mobility found in the mobile ambients. The advantage of our approach therefore is two-fold. On the one hand, it allows us to make the analysis in a "formal" manner, because we can prove theorems with respect to the dynamic semantics (as mitigated by the analysis). On the other hand, the techniques used to implement the analysis provide insights on how to define the capabilities of the hardest attacker.

Finally, to make this practical we developed a polynomial time test validating the protectiveness of the firewall against the "hardest attacker". The efficient operation of this test hinges on the efficient implementation of the static analysis and profits from the general techniques for the design and implementation of static analysis.

We are currently extending the development to deal with the communications present in the full calculus of mobile ambients.

Conclusion. In summary, we have illustrated a novel approach to the validation of safety and security properties of software systems. We are confident that it will scale up to other calculi and to web-based languages with explicit cryptographic primitives. By considering flow logics that express more powerful analyses it is likely that one can capture attackers in a more precise manner; perhaps one can show that the firewall protects againsts attackers knowing only some of the secret passwords or even against attackers knowing all the secret passwords but being unable to use them in the appropriate manner (as might be the case for an agent representing a "courier" service).

References

1. M. Abadi. Secrecy by typing in security protocols. In *Proceedings of Theoretical Aspects of Computer Software 1997, LNCS 1281*, pages 611–638. Springer, 1997.
2. C. Bodei, P. Degano, F. Nielson, and H. Riis Nielson. Static analysis for the π-calculus with applications to security. *Information and Computation*, (to appear), 2000.
3. D. Bolignano. An approach to the formal verification of cryptographic protocols. In *Proc. 3rd ACM Conf. on Computer and Communications Security*, pages 106–118. ACM Press, 1996.
4. L. Cardelli and A. D. Gordon. Mobile ambients. In *Proceedings of FoSSaCS'98, LNCS 1378*, pages 140–155. Springer, 1998.
5. D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, IT-29(12):198–208, 1983.
6. F. J. T. Fábrega, J. C. Herzog, and J. D. Guttman. Strand Spaces: Why is a security protocol correct? In *Proceedings of 1998 IEEE Symposium on Security and Privacy*, pages 160–171. IEEE Press, 1998.
7. R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes (I and II). *Information and Computation*, 100(1):1–77, 1992.
8. F. Nielson and H. Riis Nielson. Flow logics and operational semantics. *Electronic Notes of Theoretical Computer Science*, 10, 1998.
9. F. Nielson, H. Riis Nielson, and C. Hankin. *Principles of Program Analysis*. Springer, 1999.
10. F. Nielson, H. Riis Nielson, R. R. Hansen, and J. G. Jensen. Validating firewalls in mobile ambients. In *Proceedings of Concur'99, LNCS 1664*, pages 463–477. Springer, 1999.
11. L. C. Paulson. Proving properties of security protocols by induction. In *Proc. 10th Computer Security Foundations Workshop*, pages 70–83. IEEE, 1997.