

# Research Opportunities in Wireless Network Software Security

Profs. Dave Naumann and Susanne Wetzels are seeking two computer science undergraduate students to work on a case study in software security. The tasks are to design, specify, and verify compliance with (1) resource usage policies and (2) information flow policies. The case studies will be done in the context of CodeBlue, a collaborative music/dance performance application based on Bluetooth wireless networking [HCV<sup>+</sup>03]. The students will earn a stipend of \$3,500 plus \$1,350 subsistence, for working approximately 35 hours per week for 10 weeks during Summer 2005.

If you are interested in one of these positions, please read this document and contact Prof. Naumann by email as soon as possible. (naumann at cs.stevens.edu)

**The CodeBlue project.** The CodeBlue system originated as a Stevens senior design project and it has been developed mainly by undergraduate students over several years. A research paper describing the project was published in GLOBECOM. (A copy is at [www.cs.stevens.edu/naumann/publications/codeblue-new-3.pdf](http://www.cs.stevens.edu/naumann/publications/codeblue-new-3.pdf).) CodeBlue is being used as a testbed for research in software security and cryptography by Prof. Susanne Wetzels, a cryptographer whose research focuses on wireless network security, and Prof. Dave Naumann, whose research focuses on secure information flow and software specification and correctness verification. For more on us, see our web pages.

This summer, CodeBlue is being re-implemented by two Stevens Scholars, in order to integrate two outcomes from previous student research —Jared Cordasco’s solution to performance problems and Sabira Gupta’s runtime access controls and policy for untrusted plug-ins. Besides improving system performance and scalability to many input sensors, we want to enforce policies on use of resources by plug-ins accorded differing levels of trust. We also want to enforce policies that thwart attacks on the Bluetooth networking middleware.

**Task 1: resource usage policy.** We are seeking one student to work on (a) devising attack scenarios and defence policies, (b) specifying the policies as contracts for plug-ins, and (c) using the JML tool to enforce the policies.

The main role of plug-ins is to provide transformations of the MIDI data streams used to control music synthesis and lighting in response to input from wireless connected sensors worn by participants. Trusted plug-ins are allowed to generate large numbers of MIDI events, since this may be the creative effect desired in a particular musical performance and the participants are in a position to evaluate and control the behavior as desired. But untrusted ones should be held to strict quotas to prevent denial of service attacks and unpredictable behavior.

By focusing on resource policy within the application, we can study countermeasures relevant to attacks on Bluetooth, especially resource consumption attacks [JWY03], but without the need to obtain proprietary code for low levels of the Bluetooth stack. For the new implementation of CodeBlue, the student will write JML specifications for the API used by CodeBlue plug-ins, focusing on resource policy. For the plug-ins that we implement, she will use the ESC/Java2 tool<sup>1</sup> to check conformance with the specifications.<sup>2</sup> As

---

<sup>1</sup><http://www.sos.cs.ru.nl/research/escjava/main.html>

<sup>2</sup>As follow-on research in which we hope to involve this student, we will study load-time checking of untrusted downloads. We are in contact with researchers at DoCoMo labs, via Ravi Jain who was the

preparation for this summer project, our PhD student Qi Sun has developed small examples of resource policies tied to code-based access control. Naumann, Wetzel, and their PhD students will be available to help with the project.

**Task 2: information flow policy.** (Note: funding for this position has been informally approved by NSF but not yet officially confirmed.)

We are seeking one student to work on (a) coding simplified versions of some attacks on Bluetooth that have recently been in the news, (b) formulating information flow policies to counter these attack scenarios, and (c) experimenting with two information flow analysis tools to see whether the attacks could be prevented.

The student will begin by using the Jif tool which, while under active development, is fairly stable.<sup>3</sup> Wetzel will guide the student in adapting relevant Bluetooth code from two of her existing attack experiments; we only need an abstraction of the relevant features, not a working implementation of the full Bluetooth stack and programming APIs. With Naumann’s guidance, the student will formulate information flow policies (e.g., restricting the misuse of PINs in “bluejacking”) in the subset of Jif that corresponds roughly to the policy language in Banerjee, Naumann, and Sun’s [BN03a, BN03b, BN04, BN05, SBN04] work. Checking the code with Jif will serve to assess the effectiveness of Jif in particular and flow policy in general, and to add to our comparative study of Jif.

A complete exploit needs not only a vulnerability in Bluetooth but also a means of entry. Java prevents code insertion by means such as buffer overflows, but applications can be vulnerable due to untrustworthy plug-ins such as the music processing components that feature in the CodeBlue application. The student will implement malicious plug-ins for CodeBlue, based on the example coded by one of Wetzel’s research assistants last year, and check abstracted versions of them using Jif —because Jif does not support Java stack inspection which is the access control mechanism used in CodeBlue.

These activities should take about five weeks. At that point, our own tool SecJ, being implemented by Qi Sun as part of his dissertation research, should be sufficiently stable for the case studies to be ported to the Java subset that we support, which includes stack inspection. The remaining five weeks will be devoted to developing the case studies: porting from Jif to Java, writing the policies, running the inference tool, and iterating until we have convincing results. A positive outcome would be that the tool indeed catches misbehavior by CodeBlue plug-ins and by flawed Bluetooth components. An equally useful outcome would be to find ways in which our policies or inference tool fall short of detecting flaws or allowing correct code.

**Qualifications and requirements.** We need a undergraduate student with good programming skills and familiarity with Java and unix. You do not need to be familiar with the ESC/Java2, Jif, or with Java security; training will be provided by Profs. Naumann and Wetzel and Naumann’s PhD students. But preference will be given to candidates who have taken a course in software security.

The NSF requires that the student be a US citizen or have permanent residence status.<sup>4</sup>

---

original sponsor of CodeBlue while at Telcordia; we will follow their approach and use ESC/Java2 as in their case study [CEI<sup>+</sup>05].

<sup>3</sup><http://www.cs.cornell.edu/jif/>

<sup>4</sup>The student will be funded under the National Science Foundation (NSF) program Research Experiences for Undergraduates (REU). Funding is tied to Naumann’s NSF grant “Collaborative Research: Formal Methods for Behavioral Subclassing and Callbacks” (grant number CCF-0429894).

The student will be co-supervised by Naumann and by Prof. Susanne Wetzel at Stevens Institute of Technology.

**Miscellaneous information.** You can learn more about this general line of research by looking at the slides of the recent talk to CS undergraduates ([www.cs.stevens.edu/~naumann/UGtalk2005.pdf](http://www.cs.stevens.edu/~naumann/UGtalk2005.pdf)).

Naumann will be spending time at Microsoft Research, Redmond, this summer as a consultant on Microsoft's Spec# tool which is a competitor to the JML. If the student does a good job with Task 1, there may be an opportunity to visit Microsoft and present the research results.

Telcordia Technologies provided initial funding for CodeBlue and we will also present to the sponsoring researchers at Telcordia if the results are successful.

## References

- [BN02] Anindya Banerjee and David A. Naumann. Secure information flow and pointer confinement in a Java-like language. In *15th IEEE Computer Security Foundations Workshop (CSFW)*, pages 253–270, 2002.
- [BN03a] Anindya Banerjee and David A. Naumann. Stack-based access control for secure information flow. *Journal of Functional Programming*, 15(2):131–177, 2003. Special issue on Language Based Security, to appear.
- [BN03b] Anindya Banerjee and David A. Naumann. Using access control for secure information flow in a Java-like language. In *Computer Security Foundations Workshop (CSFW)*, pages 155–169. IEEE Computer Society Press, 2003.
- [BN04] Anindya Banerjee and David A. Naumann. History-based access control and secure information flow. In *Proceedings of the workshop on Construction and Analysis of Safe, Secure and Interoperable Smart Cards (CASSIS)*, 2004.
- [BN05] Anindya Banerjee and David A. Naumann. State based ownership, reentrance, and encapsulation. In *European Conference on Object-Oriented Programming (ECOOP)*, 2005. To appear.
- [CEI<sup>+</sup>05] Ajay Chander, David Espinosa, Nayeem Islam, Peter Lee, and George Necula. Enforcing resource bounds via static verification of dynamic checks. In *European Symposium on Programming (ESOP)*, 2005. To appear.
- [HCV<sup>+</sup>03] Dennis Hromin, Michael Chladil, Natalie Vanatta, David Naumann, Susanne Wetzel, Farooq Anjum, and Ravi Jain. CodeBlue: a Bluetooth interactive dance club system. In *IEEE Globecom*, 2003.
- [JWY03] Markus Jakobsson, Susanne Wetzel, and Bulent Yener. Stealth attacks on ad-hoc wireless networks. In *IEEE Vehicular Technical Conference*, 2003.
- [SBN04] Qi Sun, Anindya Banerjee, and David A. Naumann. Modular and constraint-based information flow inference for an object-oriented language. In Roberto Giacobazzi, editor, *Static Analysis Symposium (SAS)*, volume 3148 of *LNCS*, pages 84–99. Springer-Verlag, 2004.