Title

Conflict or Review — Two Approaches to an Information Security Laboratory[1]

Author Information

Michael Näf	David Basin
Information Security Group	Information Security Group
Computer Science Department	Computer Science Department
Swiss Federal Institute of Technology	Swiss Federal Institute of Technology
michael.naef@inf.ethz.ch	basin@inf.ethz.ch

Postal Address
Michael Näf
Information Security Group
Haldeneggsteig 4 / Weinbergstrasse
ETH-Zentrum, IFW C47.2
CH-8092 Zürich
Phone: +41 44 632 68 76
E-Mail: michael.naef@inf.ethz.ch

Keywords
Applied Information Security, Laboratory, Education

Subject Classification
(According to the ACM Computing Classification System)
C.0, C.2.0, C.2.4, D.4.6, H.2.0, H.2.7, K.3.2, K.6.5

[1] Based in part on M. Näf and D. Basin, "Konflikt oder Review — zwei Ansätze für Labors in angewandter Informationssicherheit", which appeared in *Informatik Spektrum*, 5 (28), Oct 2005, Springer. Permission for reuse has been kindly granted by the publisher.

Hands-on course work plays an important role in Information Security education. Through such course work, students can apply, deepen, and extend the conceptual knowledge taught in more theory-oriented lectures. Moreover, they can acquire experience with, and an understanding of, realistic systems, which is particularly important in Information Security. Hands-on experience shows how often a system's security depends on fine, seemingly innocuous details. Such details can be the source of serious security weaknesses and it is difficult to grasp them only in theory.

The importance of laboratory-based courses in general is also emphasized in curriculum development [1, 6]. The advantages are manifold: labs foster teamwork, enhance the ability to deal with concrete systems, and train students to develop practical solutions for realistic problems while accounting for constraints like usability, cost, performance, and security.

How can a laboratory-based course in Information Security be designed? We will present two approaches, one based on conflict, the other on review. The Conflict-Based Approach is popular and well-documented and here we just briefly recall its main features. The Review-Based Approach is less common and only partially documented. We have designed and held a review-based course that we present in detail as an exemplary realization of this approach. Finally, we compare and contrast both approaches, as well as the skills they convey.

1 The Conflict-Based Approach

Over the past decade, various institutions have developed and offered so-called hacker labs. These labs emphasize real-time attack and defense activities (e.g. [2, 5, 3]). We call this the *Conflict-Based Approach.* It is typically structured into four phases:

1. **Knowledge acquisition:** The students acquire the necessary offensive and defensive skills and understanding. They learn how to attack third-party IT systems and how to protect their own. Protecting a system often includes deploying preventive measures, system monitoring, and collecting evidence using forensic techniques. Knowledge acquisition is usually based on some combination of lectures, laboratory experiments, and reading material.

2. **Design and implementation:** Given a laboratory environment and associated "rules of the game", student teams design and implement their own IT systems, including infrastructure for applying offensive and defensive strategies in the conflict phase.

3. **Conflict:** Each team tries to protect its own system and simultaneously attack other systems. In some variants, the teams concentrate fully on protecting their systems against attacks by professional security experts, so-called Red or Tiger Teams.

4. **Wrap-up:** The teams reflect, discuss, and summarize their experiences in a presentation or written report.

The most important characteristic of this approach is the conflict, which is carried out in real-time. One of the first labs of this type was, not surprisingly, designed by a US Military Academy [5, 7]. Warfare is used as an analogy and one speaks of weapons, battles, and battlefields. Tactics are based on the insights of military strategists like Sun Tzu and Julius Caesar.

The real-time nature of such information warfare puts the student teams under pressure. They must be able to quickly identify the cause of problems and security breaches and devise and implement effective countermeasures. This requires intense work and the students acquire considerable knowledge and understanding of technical interrelationships in a short amount of time. They also learn to organize themselves within their teams.

2 The Review-Based Approach

Based on the understanding of the Conflict-Based Approach described above, we have designed the *Applied Security Laboratory* course at the Computer Science Department of ETH Zurich. The course is positioned as an optional part of the major program in Information Security. Basic knowledge of Information Security, networking, and operating systems — acquired either from previous courses or self study — is a prerequisite to taking the course. We have structured the course similar to the Conflict-Based Approach, but with a different emphasis:

1. **Knowledge acquisition:** The students prepare by working through self-study material in groups of two. (This phase represents about 30% of a 14-week semester, with 3 hours of laboratory work and 3 or more hours of additional individual work per week.)
2. **Design and implementation:** The students form teams of four. Each team receives the same assignment and is expected to design and implement a system that fulfills the requirements specified. (30%)
3. **Review:** The teams swap the systems they implemented and each conducts a thorough technical and conceptual review of another team's system. (20%)
4. **Wrap-up:** The teams finish their documentation and discuss results. The students' learning is assessed in a final examination. (20%)

Before describing these phases in more detail, we first address the question of technical infrastructure.

2.1 Technical infrastructure

A security laboratory requires a technical infrastructure which the participants use for their experiments in the different course phases. The requirements on such an infrastructure reflect both the needs of the students carrying out experiments and the needs of the instructor for building experiments and administrating the infrastructure. We summarize here the central requirements that guided the design of our infrastructure.

- **Accessibility:** The students must be able to work efficiently and independently. Moreover, they require privileged access to the experiment setups.
- **Maintenance:** The time and effort for the initial implementation and continuous operation must be reduced to a minimum by designing a very simple infrastructure.
- **Security:** Third-party systems in external networks, including the Internet, must not be inadvertently affected by laboratory activities.
- **Stability and extensibility:** Existing experiments must be repeatable and new experiments should be easy to implement and deploy.

These requirements can be met in different ways. We employ a straightforward architecture: A dedicated network within a lab room connects ten workstations and an infrastructure server. The infrastructure server in turn provides central services like user authentication, storage, and Internet access. A firewall filters traffic to and from external networks.

A central ingredient in ensuring security and stability is virtualization. In our case, we equip each workstation with a VMware installation, which virtualizes an x86 architecture. This virtualization makes it possible to run several operating systems, each in its own virtual

Integrity Check

A standard way of detecting manipulated files is to compare current file attributes with previously recorded ones. Here is a simple way of doing this:

```
# find / -xdev -type f -print0 | xargs -0 md5sum > /tmp/new.md5
# diff -1 -u /etc/old.md5 /tmp/new.md5
```

Question 12: This solution associates each file with a cryptographic checksum. What modifications remain unnoticed when you use this technique?

There are a number of tools, like Tripwire or AIDE, that are substantially more comprehensive than the solution described above. We are going to study AIDE, an open source tool, in more detail. [...]

Figure 1: Excerpt 1 from the self-study material

machine but on the same workstation. The virtual machines can communicate with each other using virtual network interfaces and devices. As we will see shortly, virtualization also facilitates review by allowing the teams to exchange images containing their, possibly networked, systems.

We provide preconfigured virtual machines for the different experiments. The students download these machines to their laboratory workstations and can then work with the resulting self-contained setup consisting of one or more networked machines.

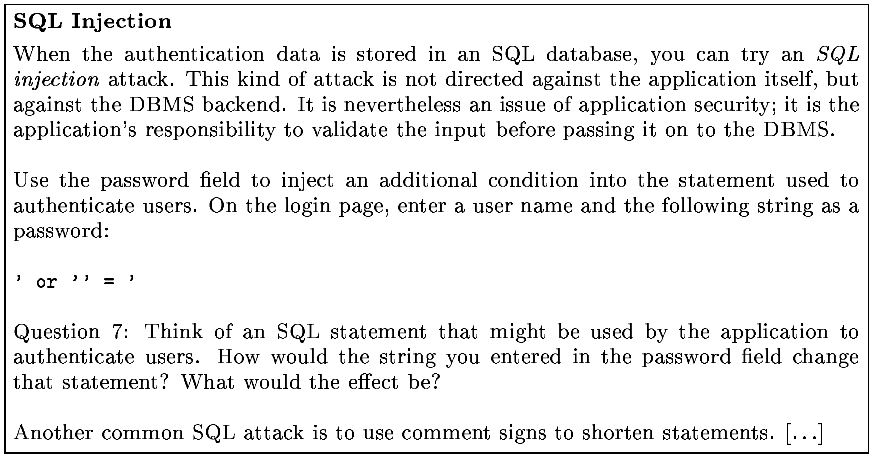
2.2 Four course phases

We now expand on our course structure.

Knowledge acquisition. The first phase is centered on self study using course material that we have designed for this purpose. The focus of our current material is on operating system and web application security. It combines background information (including references to additional reading material), hands-on experiments, and short exercises and questions that encourage the students to reflect on what they observed during their experiments. Figures 1 and 2 show two excerpts from this material.

We also hold a series of short lectures that complement the course material. The lectures address topics such as the role of standards in Information Security, how to conduct a risk analysis, or ethical and legal issues. However, the students spend most of their time at the laboratory workstations, working through the experiments and associated questions. The students work in groups of two, mainly due to the limited number of workstations. But the teamwork also gives them the opportunity to discuss the material and learn from each other. Instructors are available at scheduled times to provide additional support.

Design and implementation. We initiate the design and implementation phase by giving the students a description of a project that is realistic but manageable within the course's time frame. The description specifies the project's goal as well as the functional and the security-related requirements (Fig. 3). The students now work in teams of four. This allows us to scale up the project and allows the students to practice teamwork. The implementation part encourages the students to deepen and extend the knowledge and skills they acquired before.



SQL Injection

When the authentication data is stored in an SQL database, you can try an *SQL injection* attack. This kind of attack is not directed against the application itself, but against the DBMS backend. It is nevertheless an issue of application security; it is the application’s responsibility to validate the input before passing it on to the DBMS.

Use the password field to inject an additional condition into the statement used to authenticate users. On the login page, enter a user name and the following string as a password:

```
’ or ’’ = ’
```

Question 7: Think of an SQL statement that might be used by the application to authenticate users. How would the string you entered in the password field change that statement? What would the effect be?

Another common SQL attack is to use comment signs to shorten statements. [...]

Figure 2: Excerpt 2 from the self-study material

Review. The review phase follows. Thanks to the use of virtual machines, the students can easily swap the systems they have implemented, including all the necessary access credentials. Each team reviews another team’s system, identifies as many security-relevant weaknesses as possible, and suggests appropriate fixes. This analysis is based on all the options available. This includes a hands-on examination of the running system to discover and even exploit weaknesses as well as carefully studying the design documentation (e.g. of the system architecture and application logic) and the implementation itself (e.g. performing a code review). The students are also required to compare the system they reviewed with their own solution and to highlight particularly novel or elegant aspects of the other team’s system.

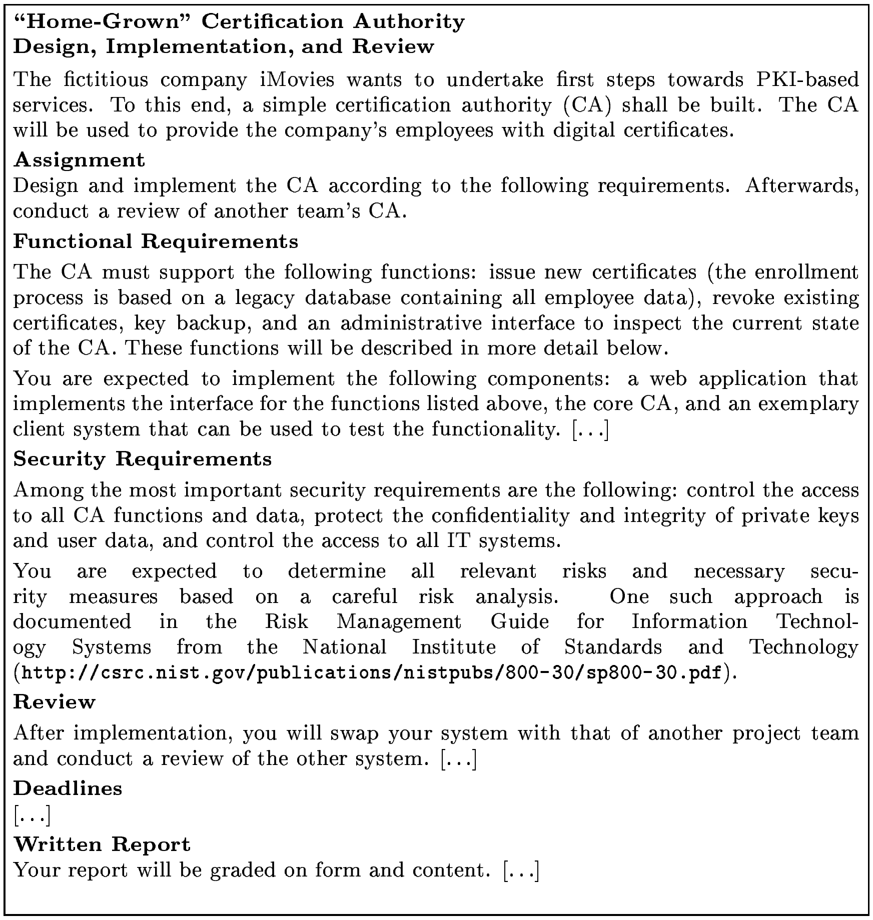
Wrap-up. Towards the end of the course, each team submits a written report that documents its own system design, including the risk analysis and the review results. All lab participants receive copies of all reports and are thus provided with comprehensive documentation of the project work. Each team also presents and discusses in class its most noteworthy or surprising conclusions, either from the design and implementation or from the review phase.

We are aware that group work can have disadvantages. Difficulties can arise, for example, when group members commit substantially different amounts of time and effort into the project or even have very different skills. Still, we consider group work to be an important part of the project and a valuable skill, and we assess the team effort: The report is graded and each team member receives the same grade.

Finally, a written examination is used to assess each student’s individual learning. It covers the first three phases, knowledge acquisition, design and implementation, and review. The final grade combines the grades from the written exam (60%) and report (40%).

2.3 Results

We have held the Applied Security Laboratory three times to date. In this section, we summarize our impressions as well as those of the students based on their responses to a survey conducted at the end of the last course. Of the 28 participants, 24 provided feedback.



“Home-Grown” Certification Authority
Design, Implementation, and Review

The fictitious company iMovies wants to undertake first steps towards PKI-based services. To this end, a simple certification authority (CA) shall be built. The CA will be used to provide the company’s employees with digital certificates.

Assignment

Design and implement the CA according to the following requirements. Afterwards, conduct a review of another team’s CA.

Functional Requirements

The CA must support the following functions: issue new certificates (the enrollment process is based on a legacy database containing all employee data), revoke existing certificates, key backup, and an administrative interface to inspect the current state of the CA. These functions will be described in more detail below.

You are expected to implement the following components: a web application that implements the interface for the functions listed above, the core CA, and an exemplary client system that can be used to test the functionality. [...]

Security Requirements

Among the most important security requirements are the following: control the access to all CA functions and data, protect the confidentiality and integrity of private keys and user data, and control the access to all IT systems.

You are expected to determine all relevant risks and necessary security measures based on a careful risk analysis. One such approach is documented in the Risk Management Guide for Information Technology Systems from the National Institute of Standards and Technology (`http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf`).

Review

After implementation, you will swap your system with that of another project team and conduct a review of the other system. [...]

Deadlines

[...]

Written Report

Your report will be graded on form and content. [...]

Figure 3: Excerpt from the project assignment

The overall impression was very positive. The students liked, in particular, the hands-on work in the knowledge acquisition phase and the realistic project work in the design and implementation and the review phase. One reason often articulated was that the project assignment was open in many respects. This encouraged lively discussions during the design phase. Furthermore, the large degree of freedom gave rise to a variety of different solutions, and the students gained additional insights by examining these design variants.

The technical infrastructure we provided was suitable for the laboratory activities. It worked well from the administrative perspective, and the students were able to work efficiently. However, some teams would have preferred to have direct Internet access from within the virtual machines, which would simplify installing software and system patches. With the current setup, emphasizing security over convenience, the students had to use secondary storage to transfer software to the virtual machines.

The most frequent criticism concerned the time and effort required to complete the lab. Many students felt that the effort needed exceeded the allotted 6 hours per week. This can be compensated in the future either by reducing the scope of the assignment or providing the students more credit (and thereby a larger time allotment) for the course.

3 Comparison

The Conflict-Based and the Review-Based Approaches are similarly structured, as depicted in Fig. 4. Moreover, they both motivate the students by providing a realistic assignment and a competitive challenge. In addition, both foster teamwork. But apart from these similarities, we can identify two fundamental differences:

- **Time pressure.** The students are under substantially more time pressure in the Conflict-Based Approach. Related to this is a stronger emphasis on using short-term attack and defense mechanisms. The students must be able to quickly select and utilize different tools. Moreover, they must be able to quickly understand problems, for example narrowing down the weaknesses exploited and the damage done during attacks.

  In the Review-Based Approach, the students can proceed at a more relaxed pace in detecting weaknesses, including those weaknesses that cannot immediately be exploited. From this point of view, the analysis is more comprehensive. An example is a brute force attack against an encryption mechanism with insufficient key length. Such an attack is feasible in theory, but requires considerable time or special hardware in practice.

- **Perspective.** In the Conflict-Based Approach, the student teams examine each other's systems in the role of an attacker and therefore primarily from the outside. Only after a successful break-in are they also in the position to analyze the system from the inside. At the same time, some students play the role of defenders or perhaps forensic specialists and analyze their own team's system with regard to its suitability to resist or detect attacks.

  There is only one role in the Review-Based Approach: the students act as reviewers and receive full access to all components of the target system. This is essential, because there are many potential weaknesses that are easier to spot with complete system access and the possibility to inspect the application logic. Examples include lax file access permissions, implementation flaws in privileged system commands, inadequate backup mechanisms, or race conditions within the application.

The two approaches emphasize different activities and skills. Both approaches have proved themselves, as the references and our experience at ETH Zurich demonstrate. An educational institution must choose between the approaches based on its educational goals. It is difficult to define exact criteria for this choice. Still, the following may serve as a rough guideline.

The Conflict-Based Approach stresses the selection and application of attack and defense strategies and techniques under time pressure; in other words, it emphasizes and develops the skills needed for reacting to an ongoing attack with real-time decision demands. This knowledge and the corresponding skills are particularly useful for IT professionals working in an operational environment (e.g. a service provider), but it can also serve as an important awareness-building measure for future managers. The Review-Based Approach emphasizes the design of secure systems, the thorough study of design variants, and the non time-critical analysis of implementations based on their designs. It is therefore especially appropriate for future developers and system architects.

An interesting option is to use both approaches together in a complementary way. The overall structure would follow the four phases shown in Fig. 4. During the analysis phase, the students would first carry out the conflict. Afterwards, they would swap the systems they tested during the conflict and conduct a more comprehensive examination during the

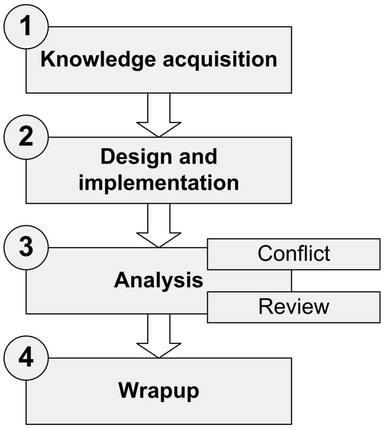


Figure 4: Common structure of the Conflict-Based and Review-Based Approaches

review. This order is sensible because the conflicts are more realistic when the opposing teams do not know what offensive and defensive measures to expect.

A combination of the two approaches clearly requires that the necessary resources — time and support — are available. Given this, the advantages of the two approaches can be combined. However, the combination is an option, not a necessity, and the decision taken should be determined by the educational goals for the course.

References

[1] P. J. Denning, D. Comer, D. Gries, M. C. Mulder, A. B. Tucker, A. J. Turner, and P. R. Young, “Computing as a discipline,” *Communications of the ACM*, vol. 32, no. 1, 1989, pp. 9–23.

[2] J. M. D. Hill, A. C. Carver Jr., J. W. Humphries, and U. W. Pooch, “Using an isolated network laboratory to teach advanced networks and security,” *Proceedings of the 32nd SIGCSE technical symposium on computer science education*, 2001, pp. 36–40.

[3] M. Micco and H. Rossman, “Building a Cyberwar Lab: Lessons Learned Teaching Cybersecurity Principles to Undergraduates,” *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education*, 2002, pp. 23–27.

[4] D. Ragsdale, D. Welch, and R. Dodge, “Information Assurance the West Point Way,” *IEEE Security & Privacy*, vol. 1, no. 5, 2003, pp. 64–67.

[5] J. Schafer, D. J. Ragsdale, J. R. Surdu, and C. A. Carver, “The IWAR range: a laboratory for undergraduate information assurance education,” *Proceedings of the 6th annual CCSC northeastern conference on computing in small colleges*, 2001, pp. 223–232.

[6] A. B. Tucker, “Computing curricula 1991,” *Communications of the ACM*, vol. 34, no. 6, 1991, pp. 68–84.

[7] D. Welch, D. Ragsdale, and W. Schepens, “Training for information assurance,” *Computer*, vol. 35, no. 4, 2002, pp. 30–37.