Proceedings of the 14th Colloquium for Information Systems Security Education
Baltimore Marriott Inner Harbor
Baltimore, Maryland June 7 - 9, 2010

A Tool for Teaching Web Application Security

Li-Chiou Chen, *Seidenberg School of Computer Science and Information Systems, Pace University,*
Lixin Tao, *Seidenberg School of Computer Science and Information Systems, Pace University,*
Xiangdong Li, *New York City College of Technology, City University of New York,*
Chienting Lin, *Seidenberg School of Computer Science and Information Systems, Pace University*

Abstract – Web application security has been an emerging topic while an increasing number of commercial applications are web-based. We are developing a new secure web development teaching tool, called SWEET (Secure WEb dEvelopment Teaching), to teach the students about web application security based on the life cycle of the application development. This paper describes the development of SWEET and provides an example of laboratory exercises on secure web communications. Experiences of incorporating SWEET in Information Assurance courses are also discussed.

Index terms – web security, virtual machine, Information Assurance curriculum, and hands-on exercise.

I. Introduction

We are developing a new secure web development teaching tool, called SWEET (Secure WEb dEvelopment Teaching), for undergraduate and graduate computing courses. The purpose of the project is to enhance the students' learning experience in computing through a standardized computing environment and teaching modules in secure web development. Web application security has been an emerging topic while an increasing number of commercial applications are designed based on Extensible Makeup Language (XML) and using Hypertext Transfer Protocol (HTTP) for communications. For example, recently, social networking software has been used intensively, especially among the college students, and integrated with various marketing or gaming software. The security of new web applications has been widely discussed but it has not been fully resolved [1, 2]. The teaching materials in this area are also very limited. There is a need for developing new teaching materials that can address the emerging security issues in web application development. The materials should also be able to attract the students' interests and provide them with the hands-on experience in learning these new concepts.

Author affiliation information goes here in 9-point, italic, Times New Roman. This footnote should use a non-printing symbol as its reference so that it does not interfere with the numbering of regular footnotes.

ISBN 1-933510-99-4 / 2010 CISSE

SWEET features virtualized web servers and a development platform that allows instructors to teach the security issues in web application development using regular computer laboratories. It includes eight three-hour teaching modules that are composed of the lecture materials, hands-on exercises and project ideas.

The goals of developing SWEET is to 1) train a new generation of computing professionals who would understand and be able to solve security problems occurring in web development, 2) introduce a new knowledge area in information assurance, 3) enrich the current computing curriculum and attract more students studying in computing, and 4) bridge the gap between the current computing curriculum and the industry expectation by introducing an emerging knowledge area, secure web development.

II. Background

A. Narrow the gap between the demand and supply in IT professionals

The enrollment of the undergraduate computing programs in the US has dropped significantly in the recent years. According to Computer Research Association, the enrollment in Computer Science (CS) has declined since its peak in 2000 and had dropped 18 percent between 2005/2006 and 2006/2007. In the meantime, the demand for the IT professionals is still increasing and they have earned more on average than other professions. Bureau of Labor Statistics predicted that the professional-level IT positions would be paid higher salaries and more than twice the growth rate of the overall workforce from 2006 to 2016, with a growth rate of 24.1% [3]. In the 2006 National Survey of Recent College Graduates (NSRCG), National Science Foundation [4] reported that CS tied for second with health majors for the highest median salary ($45,000) at the baccalaureate level.

These statistics have shown that the US education at the undergraduate level will not produce enough IT professionals to meet the demand of the job market if the trend continues. There are many reasons that contribute to the gap between the demand for IT professionals and the

enrollment of computing undergraduates. One of them is the lag between the knowledge scope of our current computing curricula and the expectations of the IT industry. It is critical to broaden the scope of student knowledge by using the teaching materials that would provide them with both the solid background in computing principles and the hands-on experiences in practice.

B. Broaden the knowledge scope of computing undergraduates

SWEET covers two emerging knowledge areas in computing: web application development and information security. As a generalization of the web technologies, the Internet business services are typically implemented by integrating existing services. The XML technology is the foundation of data integration across heterogeneous IT systems. The web service is a particular implementation technology of the Internet business services, and service-oriented architecture (SOA) specifies the software architecture based on the service integration. The information security has become an importance issue for Internet business services in different disciplines, such as banking, finance, and telecommunications. The annual CSI/FBI computer crime and security survey [5] had shown that information security has continuously been a top priority for many organizations. This trend brings a great demand for the qualified Information Assurance (IA) professionals. Frost & Sullivan [6] estimated that the number of information security professionals worldwide in 2007 were approximately 1.66 million. This figure was expected to increase to almost 2.7 million professionals by 2012. This demand has provided a great opportunity for the computing programs. To meet the current trend, we need innovative courseware to train qualified security professionals.

C. Provide innovative and practical teaching materials in secure web development

The teaching materials in secure web development are very limited. While teaching information assurance courses, we found that it is difficult to combine appropriate laboratory exercises using the current available courseware to cover both the web technologies and the security topics. The existing courseware often needs to be specially re-designed in order to meet the specific learning objectives designed for the students in computing. Motivated by the lack of appropriate courseware to meet our demand, we decided to design our own hands-on teaching tool in secure web development.

Many information security educators had designed courseware with hands-on laboratory exercises for information assurance courses [7] but none of them focused specifically on secure web development. The textbooks in web security that are suitable for undergraduate courses are also very limited. Most of the textbooks in computer security published in recent five years only keep a chapter or a section in web security with a limited overview of Secure Socket Layer (SSL) and certificate authority. While there are many books in web application vulnerabilities [8-13] and secure programming [14], but they are designed for the practitioners, not for the undergraduate students.

III. SECURE WEB DEVELOPMENT TEACHING MODULES

A. Design Criteria

Portability and flexibility for easy adoption: To achieve the flexibility of adoption, we have designed SWEET teaching modules as self-contained units so that they can be taught together as a single web security course or adopted separately in the appropriate courses that cover some aspects of secure web development, such as web development courses or system analysis and design courses.

Simplicity for students with limited background: The SWEET teaching modules target at the sophomore to junior undergraduate students who have only taken an introductory level of programming as well as some computer networking concepts. The modules are also suitable for the Information Systems/Information Technology Masters' students who have only an introductory background in computing. The laboratory exercises have been designed to illustrate the principles taught in lectures. For example, when discussing web application vulnerabilities, the lecture materials introduce different types of attacks against the web applications, such as the SQL injection or Cross-Site Scripting. The hands-on exercises also contain a vulnerable web server so that the students can follow simple step-by-step instructions to see how these attacks are carried out and how the web developers can avoid these vulnerabilities or apply the ramifications.

Structured hands-on laboratory exercises to attract students' interests: Each SWEET teaching module is composed of the lecture materials and hands-on exercises that focus on the topics discussed in the lecture materials. These exercises will allow the students to review the contents of the lecture and to apply what they have learned to solve well-structured problems in a pre-configured virtualized environment.

Semi-structured project ideas for building problem solving skills: Course projects can stimulate students to form new ideas and to develop their problem solving skills. SWEET incorporates the descriptions of potential

ISBN 1-933510-99-4 / 2010 CISSE

course projects that focus on the web security problems faced by the practitioners. These projects aim to encourage the students to think across all secure web development topics and to collect the relevant information to solve problems.

Secure web development life cycle: We have incorporated the software assurance paradigm [15] in SWEET. The current network level defenses are not enough to solve the security problems embedded in the web application development. For example, a packet-level firewall does not examine the contents of web level traffic and therefore cannot identify any potential threat embedded in the application layer traffic. The fundamental solution to address the web application security is to identify the security vulnerabilities right from the development stage of the web application. Software assurance ensures the web applications to be as they are designed by examining each stage in the life cycle of the web application development. The Software Assurance Initiative promoted by the Department of Defense defines software assurance as " the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software" [16]. We have applied this paradigm specifically on the web application development during the analysis, design, implementation and deployment. SWEET teaching modules are formatted based on the web application vulnerabilities in each phase of the life cycle of the web application development.

B. Virtualization Platform

SWEET utilizes the virtualization technology to configure a computing environment needed for the hands-on exercises. All exercises are built upon a pre-configured virtualized platform that has a virtualization layer and an application layer. The virtualization layer contains two pre-configured VMware virtual machines (VM) that allow the students to conduct the hands-on exercises on both the Windows and Linux environments. The application layer includes a web server, a database that connects to the web server, web applications, a proxy server for monitoring the web traffic, and other web security and programming tools. To run the virtual machine, the students only need to download the free VMware Player. The pre-configured virtual machines and the host machine can run concurrently.

The virtual machines which support the SWEET include the following applications pre-installed and pre-configured:

- Web and application servers: IIS, Tomcat, Apache, GlassFish (Sun's Java EE 5 server reference implementation),
- Web Proxy: Paros, Web Scarab ,
- Web Security testing: Web Goat , .Net Security Toolkits,
- Programming/scripting languages: Java, C#, C/C++, VB .NET , Perl, PHP, Ruby,
- Programming IDEs: Eclipse, NetBeans, Visual Studio, Java Development Toolkit (JDK),
- Tutorials and documentation: MSDN library, Java EE 5 tutorial, Google Web Toolkit ; Web service and XML tutorials and Linux tutorials.

C. Web Security Teaching Modules

Each SWEET[1] teaching module will include the lecture materials and hands-on exercises. We have developed four modules and are currently developing another four modules. The eight teaching modules are described as below:

1) Web application development overview: The lecture will cover HTML form and its various supported GUI components; URL structure and URL rewrite; HTTP basic requests; the four-tiered web architecture and web server architecture and configuration; session management with cookies, hidden fields, and server session objects; CGI vs. Java servlet/JSP web applications. The laboratory exercises will guide the students to setup a web server and observe HTTP traffic via a web proxy.

2) Service oriented architecture overview: The lecture will cover the service-oriented computing and architecture; web service for integrating heterogeneous information systems across the networks; service interface methods and method invocation data with XML dialects WSDL and SOAP. The laboratory exercises will guide the students to configure and secure a simple web service application.

3) Secure Web Communications: The lecture will cover Secure Socket Layer (SSL) protocols; public key infrastructure, certificate authority and X.509; digital certificates; certification validation and revocation; online certification status protocol. The laboratory exercises will guide the students to configure SSL on a web server and to create and sign server certificates.

4) Security issues and countermeasures in the analysis and design phases of web development: The lecture will cover the life cycle of web application development; abuse cases analysis; risk analysis, secure requirements; and Secure UML. The laboratory exercises will guide the students to design a secure requirement plan and conduct the risk analysis for a web service application.

[1] SWEET teaching modules and virtual machines can be downloaded from http://csis.pace.edu/~lchen/sweet/.

ISBN 1-933510-99-4 / 2010 CISSE

5) Security issues and countermeasures in the implementation phase of web development: The lecture will cover the attacks exploiting vulnerabilities occurred during the implementation, such as buffer overflow, SQL injection, and poor authentication. Code review and risk-based testing will be discussed. The laboratory exercises will guide the students to understand the various vulnerabilities and countermeasures via a pre-configured vulnerable web server.

6) Security issues and countermeasures in the deployment phase of web development: The lecture will cover the attacks exploiting vulnerabilities occurred during the deployment, such as cross-site scripting (XSS) and e-shoplifting. The architectural risk analysis will be introduced, which include the attack resistance analysis, ambiguity analysis and weakness analysis. The laboratory exercises will guide the students to understand XSS and e-shoplifting and countermeasures via a pre-configured vulnerable web server.

7) Web application stress testing: The lecture will cover the application penetration testing; web server load balancing; and distributed denial of service attacks. The laboratory exercises will guide the students to conduct a penetration testing to a pre-configured vulnerable web server.

8) Securing Ajax application: The lecture will cover the client-side sandbox security; Java security policy management; Ajax technology overview; securing Ajax applications; tradeoff of client-side and server-side computing with Ajax. The laboratory exercises will guide the students to study the security vulnerabilities of a sample Ajax application.

IV. An Example of Hands-on Exercises: Secure Web Communications

Each exercise contains a step-by-step instruction to show the students how to accomplish a laboratory assignment. All software tools needed and the laboratory instructions are pre-configured on the SWEET virtual machines. Figure 1 shows a part of the laboratory exercises for introducing the secure web communications. These exercises guide the students to examine the root certificates and web server certificates in the browsers and to create a self-signed certificate for a web server. The web server for this exercise is pre-configured in a VMware virtual machine. Ubuntu[2] Linux, Apache[3], and OpenSSL[4] are pre-installed in a virtual machine although the students are provided with a clean slate Ubuntu virtual machine and the instructions to install Apache and OpenSSL from scratch.

For creating a self-signed web server certificate, the exercise guides the students to create a public and private key pair, a Secure Socket Layer (SSL) certificate and a certificate signing request (CSR). The students will then play the role of a Certificate Authority (CA) to sign the certificate signing request for the pre-configured virtual web server. In the lecture materials, we have introduced the concept of public key encryption, digital certificates, and vulnerability in SSL implementation. We have also cautioned the students of the risk in using self-signed certificates and explained why a commercial server would ask a trusted third party, such as VeriSign or RSA, to sign the certificate.

The step-by-step instructions guide the students to create a certificate for a pre-installed virtual web server using OpenSSL, to run the server and to communicate with the server using HTTPS. The students are also asked to examine the HTTPS handshaking transactions. We have also designed review questions in the step-by-step instructions to verify the students' progress and to ensure the students' understanding of the results.

The virtualization technology provides various advantages of running the exercises. First, the exercises are portable regardless the underlying operating systems so that the students can run it either in a general-purpose computer laboratory or on their home computers. Second, the exercises can be repeated multiple times as needed. As long as the students start from the virtual machine provided for the exercise, they are able to practice the same configuration steps as many times as they would like to. Third, the experimental environment is confined within a virtual machine without interrupting the actual computing environment. Finally, a vulnerable virtual web server can be configured, investigated, exploited and fixed within the virtual machine without exposing the vulnerabilities to the actual networking environment.

[2] Ubuntu is a variant of Linux which is available at http://www.ubuntu.com/.

[3] Apache is an open source web server program which is available at http://www.apache.org/.

[4] OpenSSL is a set of open source SSL libraries which are available at http://www.openssl.org/.

ISBN 1-933510-99-4 / 2010 CISSE

Laboratory Exercises
Topic: Secure Web Communications

I. Certificate Management in a Web Browser

This exercise guides the students to examine the root certificates pre-installed in browsers and web server certificates sent by HTTPS sessions.
…….(instructions) ……..

II. Create certificates using OpenSSL on Linux Virtual Machine

We have established a web site called Pace Bank on a pre-configured Linux virtual machine and the web site is not secure. We will secure it by creating a SSL certificate for the web server before we can run the server securely with HTTPS.

1. Access the Terminal window by navigating to Applications > Accessories > Terminal.

2. Point the terminal shell to the ssl directory by running command:

cd /etc/apache2/ssl

The ssl directory is where all the private keys, certificate signing requests and certificates are stored.

3. To generate the Certificate Signing Request (CSR), you need to create your own private/public key pairs first. You will create a key by the name of server.key.

The following command generates a RSA private key that is 1024 bit long. Run the command from terminal to create the key:

sudo openssl genrsa –des3 –out server.key 1024

where the parameters are

- **genrsa** indicates to OpenSSL that you want to generate a key pair.
- **des3** indicates that the private key should be encrypted and protected by a passphrase.
- **out** indicates the file name in which to store the results.
- **1024** indicates the number of bits of the generated key.

When prompted for the [sudo] password, it is the same that is used to login. You are prompted for a pass phrase, once you type in the initial pass phrase, you will be asked to verify this pass phrase.

Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:

4. Next you create a certificate signing request with the private/public key you have just created. This command will prompt for a series of things: When prompted enter the values as follows:

Country Name: US
State or Province Name: New York
…….(a set of values for the certificate)

To create the Certificate Signing Request, run the following command at the terminal prompt, making sure you are still in the /etc/apache2/ssl directory.

sudo openssl req –new –key server.key –out server.csr

You will be prompted to enter your private key passphrase and to enter the values which were addressed above.

5. Now you will create your self-signed certificate.

The certificate signing request (CSR) has to be signed by a Certificate Authority (CA). For testing purpose, you will not ask a commercial CA (such as Verisign) to sign the certificate but sign the CSR by yourselves, which is called self-signing. In this case, you are your own CA.

The following command takes the certificate signing request and your private key in order to create your self-signed certificate. The certificate will expire in 365 days.

sudo openssl x509 –req –days 365 –in server.csr –signkey server.key –out server.crt

6. Run command *ls* in the /etc/apache2/ssl directory and you will see the self-signed server certificate (server.crt), the certificate signing request (server.csr), and the server private key (server.key).

III. Running a Secure Web Server

This exercise guides the students to run a web server through HTTPS using the certificate they have just created in II. …….(instructions) ……..

IV. Capturing TLS Handshakes

This exercise guides the students to capture TLS handshaking transactions between the Pace Bank virtual server and a browser using OpenSSL.
…….(instructions) ……..

Figure 1. An example of laboratory exercise instructions

ISBN 1-933510-99-4 / 2010 CISSE

V. Evaluation and Discussions

We have currently incorporated some of the SWEET modules in two courses at Pace University: Overview of Computer Security, an introductory computer security course for the undergraduate students, and Web Security, an IA graduate elective course. We collected the students' feedback on the SWEET modules adopted in the two classes offered in Fall 2009. A web-based survey using 5-point Likert scale was conducted at the end of the semester. The survey included the questions to elicit their feedback on the lecture materials, laboratory exercises, the mapping between the lecture and the lab and the overall impact of these modules on their learning experience. Table 1 shows a summary of the demographics and Table 2 shows a list of summarized results.

Our results show that the students had invested significant amount of time (2-4 hours per week on average) in completing their hands-on exercises. However, they generally agreed that the course materials were well planned (average 4.1 for the lecture category), the exercises had drawn their interests (average 4.1 for the lab exercise category), the exercises had helped them in learning the course materials (average 4.1 for the mapping between labs and lecture), and they would be interested in pursuing further in IA (average 3.9 for the overall category).

Our results also show that the teaching modules have achieved the similar results for the different groups of students. After a multiple regression analysis, we found that there is not enough evidence to support significant difference between the graduate and undergraduate students, between the male and female students, and between the full-time working professionals and full-time students.

The SWEET teaching modules will be adopted by New York City College of Technology, which is a minority university. Part of the SWEET projects will be incorporated in two undergraduate courses at New York City College of Technology: Web Design and Information Security. We believe that this collaboration will broaden the participation of underrepresented students. Furthermore, the SWEET teaching modules will be posted on a project web site of both institutions to help other institutions to adopt or incorporate it into their Web/Security courses and to train more qualified IT professionals to meet the demand of the workforce.

Number of Participants	45 in total Graduate students: 31 (69%) Undergraduate students: 14 (31%)
Average age	Graduate students: 31 Undergraduate students: 24
Sex	62% : Male 38% : Female
Average years of working experience	38% working full time (an average of 5 years working experience and an average of 1.5 years of IA experience)
Average hours of studying outside the classroom	Graduate students: 10 hours Undergraduate students: 7 hours
Current use of security related computer applications or tools	11% : Never 11% : about once a month 31% : about once a week 16% : about once or twice a day 18% : three to five times a day 13% : more than five times a day
Current general computer use	2% : about once or twice a day 22% : about three to five times a day 76% : more than five times a day
Availability of computer access	4% : at home only 96% : both at home and at office/school

Table 1: Summary of demographics

Survey Item / Average / (Standard Deviation)	**Category average /Standard deviation**
Q1. The contents of the lectures improve my knowledge in information security/computer network security. 4.0 (1.0)	Lecture: 4.1 (1.0)
Q2. Each lecture has a well-designed theme in information security/computer network security. 4.0 (1.1)	
Q3. Each lecture has sufficient supporting course materials, such as handouts, slides, textbook materials, for me to understand and review the weekly topic. 4.0 (1.1)	
Q4. The contents of the lectures covers topics that I would like to learn in information security/computer network security. 4.0 (1.0)	
Q5 Please estimates the number of hours you spent on a laboratory exercise. graduate students: 4.3 (3.7) undergraduate students: 2.3 (2.6)	

ISBN 1-933510-99-4 / 2010 CISSE

Q6. Each lab exercise has a theme in the area of information security or computer network security. 4.2 (1.0)	Lab/ Homework Exercises: 4.1 (1.0)
Q7. Lab exercises stimulate my further interests in learning other security software or technology. 4.1 (1.1)	
Q8. The lab equipments are sufficient for running the required lab exercises. 4.0 (1.0)	
Q9. I have better understanding regarding the weekly topic after finishing the corresponding lab exercises. 4.0 (0.9)	Mapping between Lab/ Homework Exercises and Lectures: 4.1 (0.9)
Q10. I know better about putting the technologies or concepts/theories being taught in practice after finishing the related lab exercises. 4.0 (1.0)	
Q11. The combination of the lectures and lab exercises makes the class more interesting and informative than a class with only lectures. 4.2 (1.0)	
Q12. Lab exercises stimulate my further interests in learning the technology and theories/concepts behind the security software or security problems. 4.2 (1.0)	
Q13. After taking the class, I am even more interested in the information security/computer network security area that I did. 4.0 (1.0)	Overall Assessment: 3.9 (0.9)
Q14. After taking the class, I am even more interested in having a career in the information security/computer network security area than I did. 3.7 (1.0)	
Q15. This class improves my knowledge and skills in the area of information security/computer network security. 4.1 (1.0)	
Q16. I will be interested in taking other security classes that blend in lab exercises with lectures. 4.0 (1.0)	

Table 2: Summary of Survey Results

VI. Conclusions

We had described our effort in developing the teaching modules for the secure web development. A laboratory exercise example on secure web communications was presented and our experience of utilizing the example was discussed. We had also discussed our experience of incorporating the modules in our IA courses.

The secure web development is an important topic in assuring the confidentiality, integrity and availability of the web-based systems. It is necessary for the security professionals to understand the security issues in the life cycle of developing a web-based system. Our secure web development teaching modules (SWEET) provides the flexible teaching materials for the Information Assurance educators to incorporate this topic in their courses and to educate the next generation of security professionals using hands-on exercises and examples.

VII. Acknowledgement

This material is based upon work supported in part by the National Science Foundation under Grant No. 0837549 and the Department of Defense under the Information Assurance Scholarship Program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, the Department of Defense, or the US government.

VIII. References

[1] M. A. Davidson and E. Yoran, “Enterprise Security for Web 2.0,” IT System Perspectives, November 2007.

[2] G. Lawton, “Web 2.0 Creates Security Challenges,” IEEE Computer, October 2007.

[3] Bureau of Labor Services, “Occupational employment projections to 2016,” Monthly Labor Review, November 2007, http://www.bls.gov/opub/mlr/2007/11/contents.htm

[4] National Science Foundation, Division of Science Resources Statistics, “An Overview of Science, Engineering, and Health Graduates: 2006,” Arlington, VA (NSF 08-304), 2006.

[5] Computer Security Institute. "CSI Computer Crime and Security Survey," 2009.

[6] The International Information Systems Security Certification Consortium, Inc. “The 2008 (ISC)^2 Global Information Security Workforce Study, ” 2008, Frost & Sullivan White Paper.

[7] D. Ulmer and L. Tao. "Architectural solutions to agent-enabling e-commerce portals with pull/push abilities", *WSEAS Transactions on Computers*, Issue 5, Volume 5, May 2006. pp.1026-1033.

[8] R. Allen, K Qian, L. Tao, and X. Fu. "Web Development with JavaScript and AJAX Illustrated", *Jones and Bartlett*, to be published in December 2008

[9] M. Andrews and J. A. Whittaker. How to Break Web Software: Functional and Security Testing of Web Applications and Web Services. February 2006, Addison-Wesley

[10] M. Fisher. Developer's Guide to Web Application Security, July 2006, Syngress

[11] S. Garfinkel. Web Security, Privacy and Commerce, 2nd Edition, 2002, O'Reilly

[12] S. Shah. Web 2.0 Security - Defending AJAX, RIA, AND SOA, December 2007, Charles River

[13] D. Stuttard and M. Pinto. The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws, October 2007, Wiley

[14] M. G. Graff and K. R. van Wyk. Secure Coding: Principles & Practices, June 2003, O'Reilly

[15] G. McGraw, "Software Security: Building Security In," Addison-Wesley, 2006.

[16] M. Komaroff (ASD/NII) and Kristin Baldwin (OSD/AT&L), DoD Software Assurance Initiative, September 13, 2005.

ISBN 1-933510-99-4 / 2010 CISSE