INFORMATION ASSURANCE FOR COMPUTER-MEDIATED CHAT IN AN EDUCATION AND RESEARCH ENVIRONMENT

John Carden[1], Kaila Billie[1], Jung Hee Kim[1], Michael Glass[2]

[1]North Carolina A&T State University (UNITED STATES)
[2]Valparaiso University (UNITED STATES)

Abstract

This paper describes information assurance issues raised by a computer-supported collaboration tool used in an education plus research environment. COMPS (COmputer-Mediated Problem Solving) is a web-based chat application designed for small-group problem-solving exercises. The student conversations are covered by Educational Rights and Privacy Act (FERPA), a U.S. law governing privacy of student records. A chat application on the public internet presents more issues, partly because the chat could be intercepted or non-students could find and join the conversations. Engaging in research raises more issues, as the files leave control of the original application and are distributed to researchers. Research publication and data management requirements of institutions and funding agencies expand the potential problems even further, since chat is now exposed to the world and preserved long-term. This paper focuses on privacy and security. The main danger is leaking. The main culprit is policies and practices that do not address leakage points.

Keywords: FERPA, Information Assurance, Computer-Mediated Chat, Collaborative Learning Environments

1 INTRODUCTION

COMPS (COmputer-Mediated Problem Solving) is a web-based chat application designed for small-group problem-solving exercises. The student conversations are covered by Educational Rights and Privacy Act (FERPA), a U.S. law governing privacy of student records [1]. Because COMPS is on the public web, it should be protected against information being intercepted, non-students finding and join the conversations, and outsiders finding its data. Engaging in research raises more issues, as the files leave control of the original application and are distributed to researchers. Research publication and data management requirements of institutions and funding agencies expand the potential problems even further, since chat transcripts are exposed to the world and preserved long-term.

The three primary information assurance goals of an application are effectively and reliably gathering data, being free of crashes while in use, and preventing the leaking of data through accidental or intentional actions. This paper focuses on privacy and security. The main danger is leaking. The main culprits are a) practices that do not conform to policy, and b) technology that was not written with the goal of addressing leakage.

A primary information leakage issue with chat is the identities of the participants. In the educational environment participants must be identified and linked to other class records. Because of its use in the research environment, chat records must be linked to many associated records for the same student: assessments, survey instruments, the identities of other students in the same conversations, and class records. Identities are therefore captured by the computer and part of the structured log data. These can be anonymized by normal means. However, within the discussion itself, students also refer to each other by names, nicknames, and misspelled names, and potentially say things to each other that should not be revealed. Furthermore, their conversations are arguably educational records covered by FERPA, and thus cannot be revealed in an identifiable way. The fact that a student participated is an educational record which should not be leaked, as it could be equivalent to publishing class rosters. Further, since chat text is used for training the COMPS text mining applications that are used for automated assessment of conversations, student identifying information from inside the conversation could accidentally end up as part of a text mining model.

COMPS information assurance issues were not analyzed completely in isolation. The authors looked briefly at the practices of Massive Open Online Courses (MOOCs) edX and Coursera and the Piazza group-collaboration web site [2] [3] [4]. All three of these sites provide educational experiences with computer-based discussion. Privacy policies and security measures taken on these sites inform proposed changes to COMPS privacy policy and security measures.

The result of this study is an analysis of the information security issues and a proposed collection of changes to address them. The information security problems of the COMPS project should be of general interest to online educational research that is embedded in real educational settings.

The analysis in this paper is in two parts: testing the COMPS software for technological attacks, and checking how COMPS project architecture can be improved to better support the research policies.

2 BACKGROUND

2.1 COMPS Computer-Mediated Problem-Solving

COMPS is a product of a research project in computer-supported collaborative learning. It allows students to solve problems together in a computer-chat environment [5] [6]. The exercises are oriented to exploratory learning, utilizing concepts and analytical skills. An example is a COMPS problem for a second level Java programming class, where students must figure out what a tricky bit of code would produce [7]. This type of exercise is in contrast to hands-on writing and testing of code. An exercise in a math education class asks students to determine the winning strategy for a nim-like game. Figure 1 shows the COMPS chat application. In addition to the chat window, it contains a window for students to write their agreed-upon answers to parts of the problem. This makes the answers and student agreement explicit, and is a convenient place for instructors to review the student work and provide help if needed. The application has been proven to be successful at helping underperforming students [6] [7] and can be a more engaging format than a typical classroom lecture.

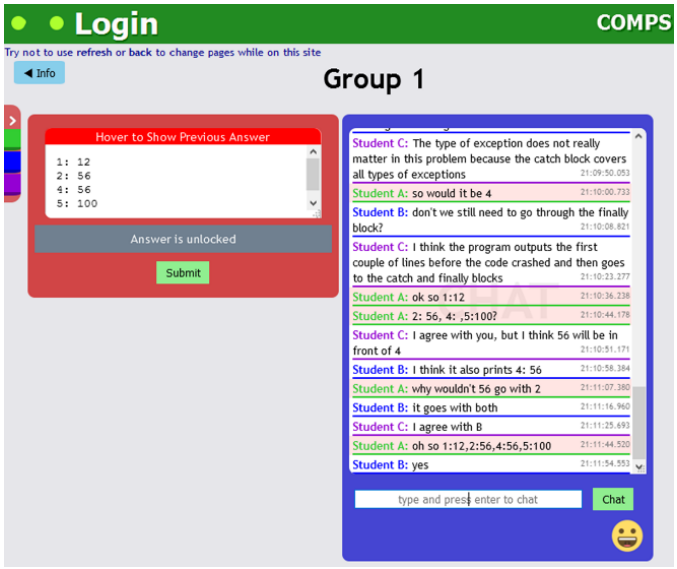

Figure 1. COMPS user session. [8]

2.2 Information Assurance Vulnerabilities

Of the three main goals of information assurance (gather data effectively, not crash, not leak data) data leaks are the focus of this paper

When students use this application, their information accumulates. Data from COMPS lab sessions is funnelled into research to help further development of COMPS and the learning exercises that COMPS employs, as well as better understanding of educational phenomena. This data includes, e.g., student names, conversations within the application, group scores on problem solving and engagement, student identification numbers, and other personally identifiable information that could violate FERPA if released. For this reason, COMPS employs a data management plan which sets rules for how user data should used, stored and transmitted.

The evolution of the product has greatly increased the possibility that data could be transmitted outside of the application and intercepted. Earlier versions of the chat software used custom TCP ports and a cross-domain policy server that would not let it run on the public web, now it will run in any web browser. An administrative web application has been added that integrates COMPS exercises with student rosters and assessments associated

with the chat exercises. Logs of conversations in process are now transmitted to another site for real-time text analysis. The universe of researchers who have access to the data has grown.

To summarize, the practices that result in COMPS conversation vulnerabilities are:

1. Conversations are transmitted in real time over the net,
2. Recorded,
3. Shared for research purposes,
4. Shared for development purposes,
5. Saved for long periods of time due to requirements of government-funded research. [8]

2.3 Existing Policy Protections

The existing protections are administrative in nature. They are governed by the data management plan required by the funding agency [9] and the human subjects research plan filed with the university's Institutional Review Board. Students that use the application sign a consent form that details how their information is to be used. Researchers who use the data pass human subject research training. The policies state that data from the experiments is published or archived only in anonymized form with personal identifying information removed. Records with identifiers are kept by investigators in password-protected computers. Data for analysis is partially kept in password protected accounts on servers where analysis is performed. It is up to the experimenters and educators in this project to follow these procedures.

3 TESTING FOR TECHNOLOGICAL ATTACKS

Much of COMPS technology was written without attention to preventing data leaks. This study was derived from common industry techniques used for security testing [10] [11]. In this section we describe the security testing method, then the results and recommendations. We finish with observation of the practices of the alternative web education web sites.

3.1 Application Map

An initial step in testing the application is to create an application map that details possible attack surfaces [10]. The application map for the COMPS chat application was derived by using a web spider tool to manually or automatically access all the links in a web page and then save the results in the tool [10]. The web spider used in this study was part Burp Suite, often used for penetration testing. The web spidering technique results in a layout of the web application and provides a map of starting points for tests. Figure 2 is a map of the COMPS application.

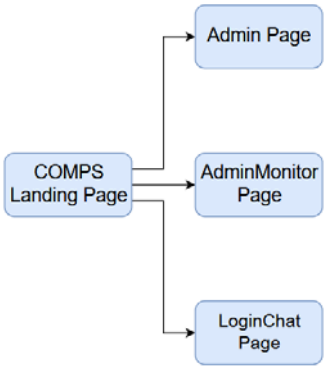

Figure 2. A simple representation of the COMPS application map. [8]

Each of these links was then subject to security testing.

In addition to this map, the web spidering also revealed several JavaScript, PHP, and image files in the application, that were not directly available when simply viewing the website in a browser.

3.2 Black-box Testing

Black-box testing methods involve probing the application from a position outside the application. We started with the URLs discovered in the application map. We used Burp Suite to listen for web traffic and capture WebSocket packets. This information is used to create simulated malicious input to the COMPS application, for example

simulated HTML form input. The black-box testing portion of this study was completed first, to be conducted with as little knowledge of the inner workings of the application as possible.

3.3 White-box Testing

White-box testing looks at the code, with both active and static code analysis. Active analysis involves reviewing the code directly to look for possible logic issues like “fail open” security measures or unsafe input validation. This is done with full access to the source code and any other system artifacts that are available. Static analysis is done using a scanning tool that scans the code and looks for known methods, practices, or libraries that are known to be vulnerable to certain types of attacks. In this case the CHECKMARX tool was used [12].

3.4 Recommendations from Technological Attack Testing

Several flaws and vulnerabilities were discovered in the process of testing the application. The most important of these are detailed here. In addition to detailing what flaws and vulnerabilities have been found, this section also details the results of examining the MOOC sites.

3.4.1 Admin Page Access

The admin page for this application is accessible by simply clicking on a link to the page. This is not a big problem, because the only part of COMPS that is accessed this way is the mechanism to start and stop the application session. Although probably not an information leakage vulnerability, it conflicts with information assurance goals because anyone can access the page and stop sessions midway through. This will be fixed by requiring authentication to access the admin page.

3.4.2 Intercepting, Altering, and Blocking Packets

One of the most alarming vulnerabilities that was found is that packets are easily intercepted, altered and blocked using Burp Suite. This allows the attacker to passively collect COMPS information, violating the third information security goal. But it also permits an attacker to interfere, violating the goal of effective and reliable information recording. The traffic that was intercepted was in the form of WebSockets messages, which contain all chat text and metadata such as names. The authors were able to influence the application by changing the contents of these packets and were able force users to seem like they were saying things that they were not. The potential for misuse in this case is very high. If the attackers are able to figure out what types of messages to send to the server, they could disconnect sessions, alter logs, destroy data, or possibly take control of the server. The best way to combat this would be implement WebSockets Secure and Secure Socket Layer technologies in the application. This would give the application end-to-end encryption and prevent third parties from capturing and altering data.

3.4.3 CHECKMARX Static Analysis

The static analysis performed by CHECKMARX found several vulnerabilities in the currently used source code. There was one judged to have high severity, which constituted 1.75% of the total discovered vulnerabilities. Thirteen vulnerabilities were judged medium, and forty-three were judged low. To be succinct, medium and low security vulnerabilities have not been addressed in this report but are covered in an internal project report [8]. The high security vulnerability uses data encoded into the URL to open a WebSocket connection to the server. A user could input a crafted URL and force the application to create a connection with malicious parameters. This could result in the application performing unpredictably, loss of data, data injection, and other problems. This high severity issue should be addressed by validating the URL-data.

3.5 Comparison with Other Web Sites

All three of the other web sites that the authors have analyzed promote collaborative learning online. The first thing that the authors noticed is that all these websites employ HTTPS for almost all communication to provide end-to-end encryption between the server and the client. Packet sniffing or putting a proxy in the middle of the communication would not be helpful to an attacker, as it is with COMPS. Another security method in use is the implementation of appropriate authorization authentication procedures. All users sign up for an account with a username and password and must login to access their account. These MOOCs offer administrative accounts for instructors so that classes can be organized and only class instructors can access information on the entire class and make changes and updates to materials and programs.

4 SUPPORT FOR COMPS DATA MANAGEMENT POLICIES

In this section we see how COMPS data management policies could be better supported with better project procedures. The most important question is whether handling of the log files supports the policy. We look at some of the privacy policies of the alternate web sites. We finish with recommendations that should strengthen the project against accidentally leaking data.

The written COMPS project policies were vetted by the Institutional Review Boards of the host institutions before the project could collect data in classes and before the institutions could receive grant contracts. These policies are re-vetted by the IRBs when they change. The funding agency additionally requires and vets the data management plan. The written policies are not tested as part of this project.

4.1 Conversation Logs

Log files of COMPS chat sessions contain all keystrokes. From the logs transcript files are derived, containing conversation text and metadata extracted from the log files. Transcript files are typically spreadsheets with one row per dialogue turn, and often contain manually or automatically generated annotations attached to each row. As noted in the background, these files are stored, transmitted over the network during lab sessions, and processed by researchers. They are saved archivally, and extracts are published. Transcript files with annotations added are inputs to further analysis and text analytic software.

Policy is that data will not escape the project (e.g. by publication or accidental transfer) with identifying information in place. After reviewing the log files in COMPS, it is apparent that the logs are being saved with the real names of users and not with pseudonyms as detailed in the data management plan. These logs are also stored in plain text, so if the server was ever compromised, the attacker would be able to view these logs without any authorization.

A second observation is that there is no tracking of who sees or downloads log files. They are kept in special-purpose password-protected accounts on a server, where most log analyses are performed. However some people have downloaded these files to their own computers to more conveniently process them. Derived data is then uploaded to the server. This is permitted, and the researchers must have passed human subjects training. However the lack of auditability is a problem. And there are no controls to enforce deletion of unneeded transcript spreadsheets from researchers' computers.

A third observation is that conversation transcripts sometimes contain identifiable or personal information. This must be manually scrubbed before the text can be published or presented.

A final observation of current practice is that the researcher responsible for publication scrubs the transcripts. This includes both scrubbing the text for individually identifying information, and removing names from conversation turns and metadata.

4.2 Other Web Site Comparisons

A main point of difference between Coursera and edX on the one hand, and Piazza on the other, is that Piazza activities are usually delivered in the context of regular classes from an educational institution. The two MOOCs offer classes independently of other educational institutions that the public can register for. In this respect COMPS exercises are more like Piazza.

All three learning platforms contain privacy policies that the user consents to. COMPS users can sign a consent form to have their data used for research purposes, as dictated by human subjects guidelines and vetted by the Institutional Review Board. Consenting to participate in the experiment is not necessary for participating in COMPS class discussions, which are expected of all students. Piazza, congruent with being offered within institutional contexts, promises to follow FERPA. Coursera, a private company with no US government education funding, does not. The edX MOOC, a non-profit corporation, says it follows FERPA.

In Piazza classes student registration is controlled by the professor. The professor uploads a roster and invites the students to join. This is comparable to the current COMPS practice, where a front-end application permits the professor to do similarly as an administrative convenience, but COMPS currently permits anybody to join a conversation.

We note Piazza students may be required by their instructor to participate in the Piazza extension of the course activities, but human subjects research guidelines dictate that the students be able to opt out of research.

4.3 Recommendations for Better Support of Data Management Policy

A first recommendation is the application be updated to use a hash value derived from the student name at the moment the data is collected during the conversation. This hash value will be used to put a pseudonym in the logs. This will prevent the main source of personally identifiable information from being leaked in all subsequent uses of the data.

A second recommendation is that the COMPS project establish a repository for log and transcript files. Having a repository can strongly aid the policies for preserving privacy:

- The logs and transcripts can be stored with names removed from their metadata
- The text of the transcripts can be scrubbed once by hand, the scrubbed texts will be the ones that are normatively distributed to researchers.
- Access to the data can be recorded. It will be known who has checked out which files.

A third recommendation is that COMPS consider protection mechanisms for spreadsheets, such as password protections or encryption.

A final recommendation is that COMPS support Piazza-style enrollment in classes and conversations, where participation in FERPA-compliant activities is controlled by the instructor of the class.

5 CONCLUSIONS

COMPS was examined for information security needs, flaws, and potential fixes. The biggest threat is leakage of personally identifiable information from educational records. The software was probed for malicious attacks, and the procedures were examined for negligent leaks.

The main conclusion is that negligent leaks are the first problem. Still a mainly in research and development stage, it is used for only about 4 exercises per semester, involving about 30 conversations. It online for about 15 hours per semester, presenting limited opportunity for attack. Log and transcript files, on the other hand, are continuously circulating among personnel. Much of the opportunities to leak information could be obviated with a few changes to procedures: anonymizing logs at the source, keeping them in a central repository, monitoring the logs as they are checked out and checked back in, protecting them when they are on researchers' computers.

A second conclusion is that a few changes to the software will significantly reduce the possibility of both passive monitoring and malicious interference.

Although the information security analysis in this study necessarily examined the particulars of the COMPS application and project, the vulnerabilities are far more general. Computer supported collaborative learning, and computer-based education, are active areas of research. As research projects move toward implementation and dissemination, the policies, procedures, and software that served well for initial stages wlll encounter many of the same issues we discovered in COMPS.

ACKNOWLEDGEMENTS

Partial support for this work was provided by the National Science Foundation's Improving Undergraduate STEM Education (IUSE) program under Award No. 1504918. 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.

6 REFERENCES

[1] "Family Educational Rights and Privacy Act," US Department of Education, [Online]. Available: https://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html. [Accessed 10 5 2018].

[2] Coursera, "Coursera," [Online]. Available: https://www.coursera.org/. [Accessed 8 May 2018].

[3] edX, "edX: Top Online Courses," [Online]. Available: https://www.edx.org/. [Accessed 8 May 2018].

[4] Piazza, "Piazza: The incredibly easy, completely free Q&A platform," [Online]. Available: https://piazza.com/. [Accessed 8 May 2018].

[5] A. Willis, A. Evans, J. H. Kim, K. Bryant, Y. Jagvaral and M. Glass, "Identifying Domain Reasoning to Support Computer Monitoring in Typed-Chat Problem-Solving Dialogues," *Journal of Computing Sciences in Colleges,* vol. 33, no. 2, pp. 11-19, 2017.

[6] J. H. Kim, M. Glass, T. Kim and K. Bryant, "27th Modern Artificial Intelligence and Cognitive Science Conference (MAICS-16)," in *Student Understanding and Engagement in a Class Employing COMPS Computer Mediated Problem Solving: A First Look*, Dayton, 2016.

[7] J. H. Kim, T. Kim and M. Glass, "Early Experiences with Computer Supported Collaborative Dialogue for a Second Semester Java Class," *Journal of Computing Sciences in Colleges,* vol. 32, no. 2, 2016.

[8] J. T. Carden, "Testing and Securely Updating COMPS, a Collaborative Learning Web Application (Master's Project Report)," Computer Science Dept, North Carolina A&T State University, Greensboro, 2018.

[9] J. H. Kim and M. Glass, "Deploying Computer Monitored Problem-Solving Discussions for Student Conceptual Understanding in Java Programming Classes: Data Management Plan," 2016.

[10] G. McGraw, Software security: Building security in, Addison Wesley, 2013.

[11] C. Wysopal, L. Nelson, D. D. Zovi and E. Dustin, The Art of Software Security Testing, Upper Saddle River: Pearson Education, Inc., 2006.

[12] "Securing Uncompiled Code with CxSAST," CHECKMARX, [Online]. Available: https://www.checkmarx.com/products/static-application-security-testing/. [Accessed 10 5 2018].