Introducing a Database Security Course at the University of Virginia

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Bevan Charles

Spring 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Technical Advisor: Rosanne Vrugtman, Department of Computer Science

Introducing a Database Security Course at the University of Virginia

CS 4991 Capstone, Fall 2021

Bevan Charles
Computer Science and Entrepreneurship
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
bac5ta@virginia.edu

Abstract

In the context of university curriculums, we use databases in order to learn how to store and use information, but securing this information can often take a backseat. Though existing courses may briefly touch on database security and give a student the notion that it's important, students may sometimes take shortcuts to get fully functional, but not secure, banks of data, since security is not a primary focus of the course. To ensure that database security is not being overlooked, creating a course on it is a necessity to prepare a software engineer who knows how to protect consumer data and create secure software that is less likely to be compromised. One of the focuses of this proposed course would be discussions revolving around the various methods through which databases are breached. Students would then learn how to protect databases and complete hands-on coding assignments to demonstrate their understanding of the material. This course will create not only a student who can secure a database, but one that will keep security and user privacy at the forefront of innovation.

1 Introduction

Software companies are nothing without two things: data and consumers. For any software-related endeavor, a data lake is mandatory to build anything meaningful. Above all else, there is a need for a database security course because of the sheer number of annual data breaches. As more software rolls out each year, it is inevitable that hackers see unsecured data on the web as a growing market, along with more potential for vulnerability.

2 Related Works

Though many smaller-scale attacks go unreported, there were 3950 confirmed data breaches during 2020 [1]. While that number may seem smaller than expected, the number of people affected in each of these breaches is astronomical. For example, in January 2020, a Microsoft database was left unprotected and revealed the personal information of more than 280 million users [1]. However, the information can get way more personal than that. From 2017 to 2019, the number of people affected by a data breach increased by an astonishing 80% [1]. Healthcare information, especially, includes some of a person's most private information and healthcare databases can leak information that can be used against patients. For example, health insurance companies can use consumer information to increase insurance rates and put a financial toll on patients. Though consumers are "protected" by the Affordable Care Act which disallows medical companies from denying service to patients, these companies use consumer information to hike up certain plan's prices and assume how risky it is to add an individual to one of your plans [2].

Additionally, though data can be leaked or accessed in many different ways, hacking and the number of people prying around for unsecured data have increased exponentially. A study of medical data breaches from 2005 to 2019 showed that a total of 249.09 million people's medical records were unauthorizedly accessed. What is more interesting, however, is the breakdown of the number of these records that were released through hacking, as opposed to other means, over the years. From 2005 to 2009, only 0.6 million records were breached by hacking. From 2010 to 2014, 14.7 million records were breached by hacking. Finally, from 2015 to 2019, 145.75 million health records were exposed through hacking [3]. This shows that over the years, the willingness by hackers to go out and search for databases with weak protection has only increased.

Although words like "data" and "software" are often used when discussing this subject, it makes it seem that big tech companies are the only places being affected, when in reality, sectors like healthcare suffer just as much. Though this is clearly a daunting problem within the software and tech industry, it may not be a huge priority for university students until they actually get to the industry and experience it.

Comparitech, an informative website for consumers that prioritizes studies on cybersecurity and user/data privacy, put a research team in charge of an experiment to see just how prevalent database attacks really were [4]. Bob Diachenko, a cybersecurity expert who was in charge of the research team, had an idea to deploy an unsecured database, full of fake data, to the cloud, and see how many unauthorized requests to access database information would come through. Over a ten-day period, roughly 175 attacks were made on this database, for an average of around 18 attacks per day [5]. Additionally, a week after the experiment ended, the database was still deployed and vulnerable for attacks. A hacker with a ransomware bot at their disposal deleted all the data from the server and left behind contact information along with a blackmail request stating that they would have to pay the hacker a hefty sum of money in order to recover their data. The attack which deleted the data and left a ransom request lasted all of five seconds, and was perfectly calculated. Though, in this case, the attack was futile since the database was full of phony data, imagine if that data was instead full of people's personal information like home addresses, social security numbers, taxpayer identification information, and even healthcare information.

This experiment perfectly demonstrates the extent to which unsecured databases can easily be penetrated and the reason database security needs to be prioritized in a university setting in order to show rising software engineers just how vital and common this issue is. However, there's a huge shortcoming in that department – according to the Washington Post, a 2016 study found that out of the top 10 computer science programs in the country, not a single one of them required their CS majors to complete even one cybersecurity elective [6]. Introducing a database security course would be a great way to allow computer science majors to get accustomed to the idea of database attacks and how often and easily they can occur.

3 Project Need

Having taken both Database Systems and Introduction to Cybersecurity at the University of Virginia, I learned the curriculum's shortcomings in terms of educating its software engineers to be more fluent in database security. Having the chance to intern at three companies in the past – a financial tech startup, a government defense contractor, and a large, corporate bank – it has become obvious that database security is not a primary focus for internships. Since the duration of most summer internships is only around eight to twelve weeks, companies are instead using this time to evaluate their interns more for team-fit and technical aptitude. Rather than having you focus on the security principles of coding, they focus on your raw coding ability in order to determine if you have the technical skills to join their company. Additionally, companies recognize that students are not professional software engineers yet, and so they leave the cybersecurity and privacy of their data to be handled by full-time employees.

3.1 Course Proposal

The proposed course, Computer Science 4755: Database Security, would integrate material taught in both Computer Science 3710: Introduction to Cybersecurity and Computer Science 4750: Database Systems. Introduction to Cybersecurity does a great job of raising general awareness of technical security among computer science majors and does an even better job of emphasizing current events to show the gravity and backlash of cyberattacks. Similarly, CS 4750 does a great job of teaching database fundamentals and necessary background that any student will need to have in order to take a class on database security. It also very briefly touches on database security, though it is not an emphasis, nor the purpose of the class.

It would be highly recommended, but not mandatory, that students interested in enrolling in the database security course complete the database systems course as a pre- or co-requisite with database security. Introduction to Cybersecurity would not be required since the direct concepts taught in it are not necessary to know anything about database security. Taking it would help students keep security as a priority while coding, but taking a class on database security will instill the same objective into a student's practices.

3.2 Course Details

While the main purpose of this database security course is to be introduced at the University of Virginia, it will be flexible to fit at any school where database security is lacking. While most schools have a class on database systems, the extent to which database security is covered in all of them is considerably different. For example, the University of Virginia covers database security for only one week of the semester, so while students gain a general understanding, they are never learning any security principles at a deep level. The proposed class would meet Tuesday and Thursday for seventy-five minutes each. Assuming a fifteen-week semester, the course would be broken down so that students would have a basic reminder on database materials, but still learn plenty about database security fundamentals.

3.3 Course Breakdown

The first couple of weeks of the class would be a refresher for students on database materials and cybersecurity principles. The next several weeks and the majority of the semester, would consist of learning of different types of cyberattacks; current and historical events; examples of specific types of attacks; and finally, how to actually counter and prevent these attacks. For each attack, theoretical material would be taught first, later supplemented with labs and hands-on activities that require students to use learned security principles to actually practice securing databases.

3.3.1 Revisiting Database System Material

As previously mentioned, weeks one and two of the course would be dedicated to re-teaching database fundamentals taught in database systems. Though most students will be familiar with this material, it would still be helpful to them to relearn the material, as it ensures that students who have previously taken database systems in two or more semesters will still have a strong background. It also provides a system of support for students who are interested in jumping into database security before taking database systems, since it is not a mandatory pre-requisite. Though these students will likely have had some database experience, either through an internship, school project or just self-learning, it will be a nice way for them to have at least some concrete background in databases so they don't feel overwhelmed or that they are falling behind.

3.3.2 Learning Database Vulnerabilities

Weeks three through six would take a look at several kinds of engineered database vulnerabilities and how these become opportunities for hackers. An example of a database vulnerability includes database configuration errors when being deployed to the cloud. For example, deploying software to the cloud usually follows one of three models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [7]. All three of the following services are used for different reasons, but all still deal with databases and provide different levels of data vulnerability [8].

For example, a Denial of Service (DoS) attack can affect any of these three cloud models. Denial of service attacks can happen to a database when too many queries are allowed to be executed at a given time, and this large amount of traffic can overwhelm a data server and make it unable to provide any type of service to a consumer and ultimately, cause it to crash [9].

Another database vulnerability to be examined is a lack of protecting database integrity and accessibility. This entails data systems leaving no access trails behind in the event that a malicious actor attempts to access unauthorized data [10]. It would also ensure that users don't have direct ownership access of database tables and also that they can only deal with the tables that they should have access to.

The most popular type of database attack, the SQL injection, which occurs when users are allowed to make dynamic queries to a database and potentially uncover all information hidden in one, will also be covered [11]. The surprising thing is that even though there are many simple ways to protect against this attack, it still happens so often. Imperva, a cyber security company, did some research and identified statistics from how many SQL injections were attempted on their firewall [12]. Out of all the types of database attacks, they found SQL injections to be the most popular with around 20% of all database attacks coming from SQL injections [13]. SQL injections are one of the easiest types of database attacks to prevent against, and so this is a strong piece of evidence showing that the importance database security is not being emphasized enough.

3.3.3 Secure Database Practices

Weeks seven through eleven would then go into detail on good practices for securing databases and how to be pro-active against becoming the victim of a database attack. One of the important concepts that will be introduced to students is role-based access control (RBAC). RBAC entails securely assigning each user within a database a 'role.' Each role has certain permissions and eligibilities to perform only certain operations on a database [14]. One of the most pressing issues with database security is giving out excess privileges to certain people, and this prevents that by limiting data access. Another important task to teach students is validating data inputs before allowing queries to be run and also encouraging the active use of prepared statements or stored procedures [15]. This is an act of using an intermediary query parametrization to ensure that malicious code is never inserted into or executed on a database. This is how to stop an SQL injection and students will be taught that following simple procedures can stop an expensive database attack. Another important practice is learning to create backups of SQL databases in the case of employee error, physical hardware failure, or even a ransomware attack that leaves you with lost or corrupt data. Cloud services like Amazon Web Services often offer tools for backup databases, but these are often costly, so the course will cover how to automate the database backup process using SQL [16]. This automation process will not only encrypt the data, but also create a schedule to create a new backup every day or week.

3.3.4 Database Auditing

Weeks twelve through fourteen will focus on two prime ideas: database auditing and how to recover from a database attack. Database auditing is the practice of monitoring actions performed on a database to ensure accountability and prevention of unexpected actions. The database security course will look at some tools that do a great job of database auditing, like Oracle's Database 12c, IBM's Db2, and the MySQL Enterprise Audit [17]. Students will receive hands-on experience working with at least one of these tools and learn how to conduct database auditing. Additionally, students will learn how to recover from a database attack. Tools like ApexSQL Log and ApexSQL Recover will be used to teach students that acting swift and efficient after a data breach still gives them a chance to recover their data and find the responsible attackers [18].

3.3.5 Course Assignments and Assessments

Weeks one through six will not be too assignment heavy, as database refresher material and learning about database attacks offers limited opportunities for hands-on activities. The Thursday lecture for the week two through six lectures will include a brief, 20-minute quiz at the beginning of class that tests basic understanding of what was taught in class the week prior. These quizzes will not be overly difficult, but will just ensure that students are understanding the material. Weeks seven through fifteen, however, will be more assignment heavy and feature less assessments, since they will actually be learning how to secure databases. Students will be expected to show their understanding of good database practices by using some of the tools mentioned in this paper as well as demonstrating clear understanding of concepts like prepared and procedure statement as well as implementing role-based access control through an online server like Oracle or phpMyAdmin.

4 Conclusion

Creating this course would be a big step in the right direction for universities who feel that their current curriculums are not putting nearly as much emphasis on security as they should be. As the cybersecurity begins to see exponential growth over the next few years, the number of security-related careers will be plentiful. Database security, particularly, is such an important aspect of being a software engineer because working at any company will show that software cannot be built without data, but software cannot be maintained without database security.

The existing database systems and introduction to cybersecurity courses will continue to do a great job of teaching database fundamentals and introducing general security principles, respectively. This proposed class on database security will stand to serve as a gateway for students to learn about database security and how to deal with it once they begin their professional endeavors.

References

[11] Sobers, R. U. (2021, April 16). *98 must-know data breach statistics for 2021: Varonis*. Inside Out Security. Retrieved October 16, 2021, from https://www.varonis.com/blog/data-breach-statistics/.

[1] Allen, M. (2018, July 17). *Health insurers are vacuuming up consumer data that could be used to raise rates*. HealthLeaders Media. Retrieved October 16, 2021, from https://www.healthleadersmedia.com/finance/health-insurers-are-vacuuming-consumer-data-could-be-used-raise-rates.

[10] Seh, Adil H., Mohammad Zarour, Mamdouh Alenezi, Amal K. Sarkar, Alka Agrawal, Rajeev Kumar, and Raees Ahmad Khan. 2020. "Healthcare Data Breaches: Insights and Implications" *Healthcare* 8, no. 2: 133. https://doi.org/10.3390/healthcare8020133

[18] Zorz, Z. (2020, October 14). *With database attacks on the rise, how can companies protect themselves?* Help Net Security. Retrieved October 22, 2021, from https://www.helpnetsecurity.com/2020/10/14/securing-exposed-databases/.

[3] Bischoff, P. (2020, September 29). *We setup a honeypot to see how long for hackers find unsecured database*. Comparitech. Retrieved October 22, 2021, from https://www.comparitech.com/blog/information-security/unsecured-database-honeypot/.

[7] Peterson, A. (2019, April 8). *Universities aren't doing enough to train the Cyberdefenders America desperately needs*. The Washington Post. Retrieved October 16, 2021, from https://www.washingtonpost.com/news/the-switch/wp/2016/04/11/universities-arent-doing-enough-to-train-the-cyberdefenders-america-desperately-needs/.

[8] Rani, D., & Ranjan, R. K. (2014). A Comparative Study of SaaS, PaaS, and IaaS in Cloud Computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, *4*(6), 458–461. https://doi.org/10.23956/ijarcsse

[5] Mozumder, D. P., Mahi, J. N., & Whaiduzzaman, M. (2017). Cloud Computing Security Breaches and Threats Analysis. *International Journal of Scientific & Engineering Research*, *8*(1), 1287–1297. https://doi.org/10.14299/000000

[4] Fruhlinger, J. (2021, February 12). *DDoS explained: How distributed denial of service attacks are evolving*. CSO Online. Retrieved October 22, 2021, from https://www.csoonline.com/article/3222095/ddos-explained-how-denial-of-service-attacks-are-evolving.html.

[15] UTC CS Department. (2020). *CPSC 4670/5670: Database Security and Auditing*. CPSC 4670/5670: Database Security and Auditing | University of Tennessee at Chattanooga. Retrieved October 26, 2021, from https://www.utc.edu/engineering-and-computer-science/caecd/course-listing/cpsc-4670.

[13] *Top 10 database attacks*. BCS. (2021, August 16). Retrieved October 22, 2021, from https://www.bcs.org/articles-opinion-and-research/top-ten-database-attacks/.

[6] Nakar, O., & Azaria, J. (2019, June 13). *SQL injection attacks: So old, but still so relevant. here's why (charts): Imperva*. Imperva. Retrieved October 22, 2021, from https://www.imperva.com/blog/sql-injection-attacks-so-old-but-still-so-relevant-heres-why-charts/.

[2] Avital, N. (2019, January 19). *The State of Web Application Vulnerabilities in 2018*. Imperva. Retrieved October 22, 2021, from https://www.imperva.com/blog/the-state-of-web-application-vulnerabilities-in-2018/.

[16] *What is role-based access control (RBAC) | core security*. CoreSecurity. (n.d.). Retrieved October 26, 2021, from https://www.coresecurity.com/identity-governance-and-administration/role-based-access-control.

[9] Rubens, P. (2021, March 11). *How To Prevent SQL injection attack*. eSecurityPlanet. Retrieved October 22, 2021, from https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks/.

[14] Upadhyay, N. (2021, February 12). *Automate SQL database backups using maintenance plans*. SQLShack. Retrieved October 22, 2021, from https://www.sqlshack.com/automate-sql-database-backups-using-maintenance-plans/.

[17] Yehuda, Y. (2021, February 24). *Database Auditing: Why You Need them and What Tools to Use*. DBmaestro. Retrieved October 22, 2021, from https://www3.dbmaestro.com/blog/database-audits-why-you-need-them-what-tools-to-use.

[12] Solution Cener. (2018, September 19). *How to recover data that is missing or damaged as a result of a SQL injection attack*. ApexSQL by Quest. Retrieved October 22, 2021, from https://solutioncenter.apexsql.com/recover-damaged-missing-data-due-to-sql-injection-attack/.