Using SQLMAP to Detect SQLI Vulnerabilities

Waad Almadhy[1†] and Amal Alruwaili[2††] and Saloua Hendaoui[3†††]

Department of Computer Science, College of Computer and Information Sciences, Jouf University,
Jouf, Skaka, Saudi Arabia

Summary
One of the most discussed topics is cyber security when it comes to web application and how to protect it and protect databases. One of the most widely used and widespread techniques is SQLI, and it is used by hackers and hackers. In this research, we touched on the concept of SQLI and what are its different types, and then we detected a SQLI vulnerability in a website using SQLMAP. Finally, we mentioned different ways to avoid and protect against SQLI.

Key words:
cybersecurity, SQLMAP, SQLI, Vulnerability, Web Application Attacks.

1. Introduction

More than ever, companies are taking the security of their databases very seriously. Database security is always a priority for companies, but it has become an even higher priority because not only are attacks on databases more prevalent, but they are also more effective. Individuals who have the training to understand how to use vulnerabilities in these systems can break into them with relative ease. These databases are used in the network, using the Web, databases are used unintentionally.[11] One of the most common attacks is SQLI. In this paper, Kali Linux has been used to perform a smooth and efficient penetration test using SQLMAP tool. All databases were obtained on the website with their tables. Finally, the IDs and passwords for the administrators were obtained.

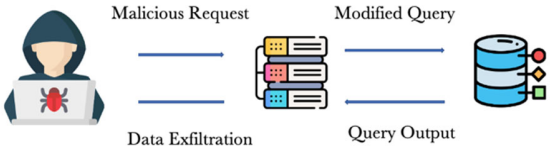

Fig. 1. SQLI attack

1.1 What is SQLI

Due to the wide sweep of digitization in the world, the adoption of applications and websites by government agencies, the military, health, and others have increased, for ease of use and quick access to data. With the use of websites, databases are used to deal with data in them. For example, it determines whether the student is accepted or not on the university website, and the status of his acceptance is saved in the database. In the health sector, the patient's name and his medical history are saved through the website, and his condition is retrieved from the databases. But is the confidentiality and security of the data in the database maintained?

SQL injection is the most common and widely used method for hacking databases by attackers, in which vulnerabilities in the database server or the web are exploited. The created threat query segments are injected to change the intended effect; thus, the attacker can gain unauthorized access to the databases by changing, modifying, or adding data, or preventing users from accessing the data and others. SQL is a query language that gives the possibility to the web using specific offices to communicate with databases. Based on the query, the results are returned by the server. The injection is only done when the user or the system accepts input queries [12].

1.2 SQL type

- First-order Injection

In this type, by providing user input with the design of HTTP GET or POST, SQL statements are injected by attackers or by using cookies, or also server variables that consist of HTTP and others. For example, to execute a second statement, UNIONS is added to an existing statement [13].

- Second-order injection

This type occurs when the data is stored by the application provided by the user and then in an insecure manner is combined into SQL queries For example, when the web fails before storing in the database in the process of validating the entry. The password is subsequently modified by the attacker using the following: UPDATE nametable SET password = “‘+ newPassword +”’ WHERE username = “” + username +”’ AND password = “’ + oldPassword + “””. In this example, the admin is the name of the logged-in attacker. Since "-" is a comment operator for SQL, anything after it will be ignored by the SQL engine. The result in this example will be that "admin", which is the name of the administrator user, will be changed to the value specified by the attacker [13].

- Illegal/Logically Incorrect Queries

In this type, invalid or illegal SQL is entered, which by way of default error pages are returned, which in turn reveals information that can be injected by collecting certain information about the back-end database and its structure for the web application. An example is checking for the column name. Input (username): "Sara" SQL: SELECT *

Manuscript received December 5, 2021
Manuscript revised December 20, 2021
https://doi.org/10.22937/IJCSNS.2022.22.1.32

FROM employee WHERE username = "Sara"' AND password =. Result:" wrong syntax at 'Sara. Unclosed quotation mark after the character string "AND Password= "ccc" [13].

- Tautologies

Here to extract the data or to determine what is injectable or also to be able to authenticate trades, a query is injected that evaluates to true for the entries entered in the database. For example, Input (username): lana' or '1'='1–. SQL: SELECT * FROM employees WHERE username = 'lana' or '1'='1' – AND password =. Result: All employees are retrieved [13].

- Union Query

UNION SELECT is injected to trick the application into returning data from another table than the intended table. For example, normal SQL statement + "semi-colon" + UNION SELECT < the rest of injected query> [13].

- PiggyBacked Queries

It is used to extract data, modify it, or perform a denial of service. In the original query, additional queries are injected. In this type, the goal is to add new queries that depend on the original query and not modify it. When this type is used, multiple SQL queries are received by the DBMS. The first query, which will be executed normally, is the normal and normal query, while to satisfy the attack, subsequent queries will be executed. A common example: normal SQL statement + ";" + (INSERT or UPDATE, DROP, DELETE) <rest of injected query> [13].

- Stored Procedures

When creating a suspicious SQL statement insert, for example, as an alternative to a normal stored procedure, a different store procedure can be done by the attacker, to elevate the privilege for example, or to implement a denial of service. as an example, using a query delimiter (;) with "SHUTDOWN" store procedure: normal SQL statement + "; SHUTDOWN;" <rest of injected query> [13] .

- Error-based

A type of injection in which error messages are relied on from the database server, which consists of information about the structure of the database[9].

- Time-Based Blind

In this type, a specific page will be shown in a time-based injection vulnerability. The technique adopted here consists of executing several inquiries in which the function and the delay rule are included. In this type, by observing the response time, the attacker can infer the information [10].

2. Lecture Review

In [14] The researchers mentioned that with the frequent gaps in most web applications, attackers and hackers can gain access to sensitive data. They also mentioned the danger of SQL injection on web applications and that it is one of the most common threats. In order not to filter the input made by the user, these attackers can exploit these errors. In their research paper, the researchers reviewed PHP techniques and other techniques to protect against SQL injection. They also mentioned the various ways to detect SQL injection attacks, their types, and the most important causes. Finally, they discussed the purification of SQL injection vulnerabilities.

In [15] to address high-risk vulnerabilities in NoSQL, researchers designed Kerberos. It was also designed to validate the Data-Centric data encryption security model. This module aids in securing NoSQL databases by designing and increasing the appropriate security mechanism. In Kerberos, powerful network encryption tools are provided to help secure data across organizations.

In [1] the researchers compared SQLI vulnerabilities on content management systems and used vulnerability scanners Nikto, SQLMAP on WordPress, Drupal, and Joomla web pages installed on a LAMP server. The results of their research were that CMS responded to SQLI attacks but got warnings about various vulnerabilities that could be exploited. Finally, practices that can be implemented to prevent SQLI are suggested.

In [2] SQLI attack methods were analyzed, and they also provided the best defense mechanisms to detect and prevent these attacks. The researchers simulated the SQLI attack process using Kali Linux. Finally, an analysis of best practices was presented to counteract this type of attack

In [3] The researchers discussed different types of SQLI attacks and what are the different ways to deal with this type of attack. The researchers also included preventive methods and examples of them. The researchers focused on countering this type of attack using stored procedures.

In [11] The SQL attack was dealt with, and then a new system was proposed that consists of three levels to detect and mitigate SQLI attacks. The approach is included in static as well as dynamic and run-time related detection and prevention mechanisms. Illegal queries are also removed, and the system is prepared for a secure environment.

In [5] The researchers proposed SQLi-labs, a program that is used for training and teaching and contains many weaknesses in SQLI. The teacher can perform SQL attacks for students using this software, which helps students to refine and train their skills.

In [6] For the SQLIV vulnerability, a black box test was proposed. It is working on SQLIV automation in SQLI. The researchers also mentioned that recent studies showed the need to improve the effectiveness of SQLIV to reduce the cost of manual vulnerability checking. The focus of this paper is to improve and increase the effectiveness of SQLIV by suggesting an object-oriented approach to help reduce false positives and to provide space for the ability to

improve the proposed scanner. Using different vulnerable applications, evaluations showed that the proposed scanner could analyze the response of the page that has been attacked using four different techniques.

In [7] It was mentioned that the proposed algorithm works fast and offers a great solution against SQLI attacks. The researchers also mentioned that the proposed algorithm is great in examining its simple detection process against SQLI attacks. Using multiple detection methods, the researchers analyzed the paperwork, which results in the ability to use the proposed algorithm in any applications that interact with the database, and not only use it on web applications.

In [8] To detect complex SQLI attacks, an adaptive method is proposed that is based on the deep forest. The researchers optimize the structure of the deep forest, by means of the first feature vector and average the previous outputs. The inputs will be sequenced at each layer. Experiments showed that the proposed method in this paper effectively solves the problem of feature degradation of deep forest which occurs with the increase of layers. Then the researchers introduced the deep forest model which is based on the AdaBoost algorithm, and which updates the feature weights in each layer by using the error rate. In the training process, there are multiple features with weights that are not the same, based on their impact on the result. Based on the results, it was shown that the performance of the method proposed in this research paper is better than the traditional methods of machine learning and deep learning methods.

Table 1: Compression table

NO	Search Title	Publication Year	Technique Used
1.	A novel three-tier SQLi detection and mitigation scheme for cloud environments	2017	Three-tier method
2.	Analysis on Database Security Model Against NOSQL Injection	2017	Kerberos
3.	Vulnerability Analysis of Content Management Systems to SQL Injection Using SQLMAP	2018	Nikto, SQLMAP
4.	An algorithm for detecting SQL injection vulnerability using black-box testing	2019	SQLIVS algorithm
5.	A SQL Injection Detection Method Based on Adaptive Deep Forest	2019	Adaptive Deep Forest Method
6.	Overview of SQL Injection Defense Mechanisms	2020	Kali- Linux
7.	A System for Prevention of SQLi Attacks	2020	stored procedure method
8.	SQL Injection Teaching Based on SQLi-labs	2020	filtering method
9.	Detection and Prevention of SQL Injection	2020	Algorithm SQLIAD
10.	SQL Injection Attacks Prevention System Technology: Review	2021	Nikto, SQLMAP neural network SQLA, DIAVA, PHP system

3. Methodology

In this part, we will discuss how to search for sites that can be infected with SQLI vulnerabilities and how to exploit them using SQLMAP. Initially to set up the system, Kali Linux will be installed on a Virtual Machine and through it SQLMAP will be used for penetration testing and exploits. Most of the infected sites are related to PHP, containing an ID identifier. Therefore, First We will filter sites using inurl:"php?id=" Google Search Operator and then determine whether they are infected or not, as shown in Figure 2.

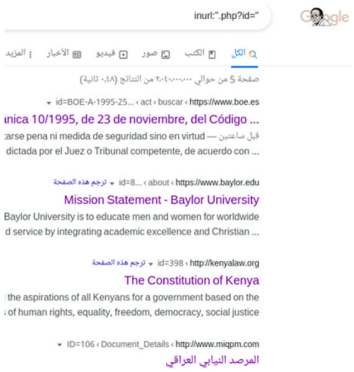

Fig. 2. inurl:"php?id=" Google Search Operator

After that, we will test the search results one by one by adding (') at the end of the website link as shown in Figure 3. If a message appears containing an SQL error, it is likely that the site contains gaps, and if a similar message does not appear, it is likely that the site is secured from this type of vulnerability.

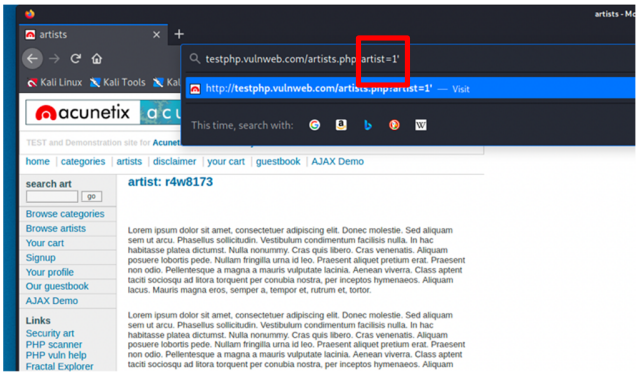

Fig. 3. adding (') to the link

As shown in Figure 4, we have an infected site. Now using SQLMAP, we will take advantage of this vulnerability by extracting the databases and their tables and the data they contain.

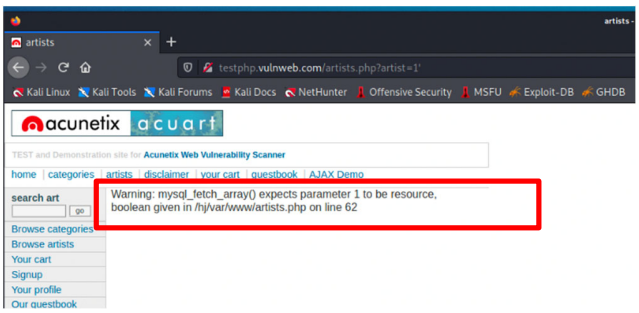

Fig. 4. SQL error

With a SQL error message, is it sure you can exploit SQLI? No, because of the different levels of the vulnerabillity. If the vulnerability is of a high level, there is the ability to exploit it, but if it is of a medium level, it is possible to succeed or fail to exploit it. But if it is of a weak level, it is difficult to exploit this vulnerability.

Using sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --dbs in kali-Linux we have shown the databases as shown in Figure 5.

Fig. 5. SQLMAP command.

As shown in Figure 6, There are 3 types of SQLI found which are error-based SQLI, time-based blind, and UNION query. There are two available database.

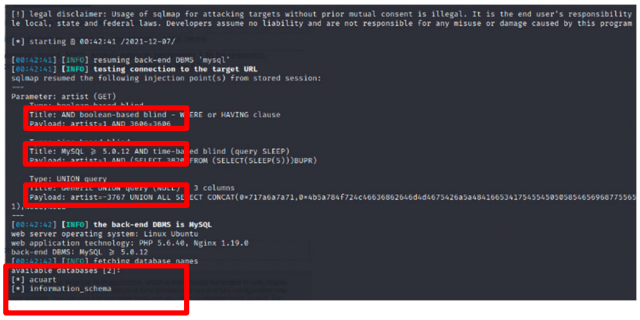

Fig. 6. SQLIs have been found.

In this part As shown in Figure 7,8 we tried to extract information about database tables using sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart --tables

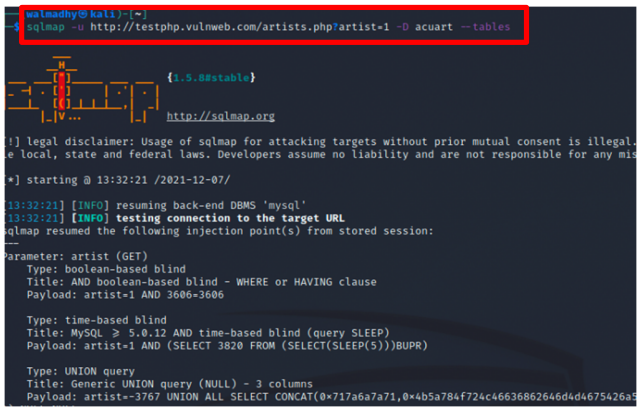

Fig. 7. SQLMAP command.

In this step, as shown in Figure 8, we extracted the columns in the acuart database in an attempt to extract sensitive data sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart --T users --columns

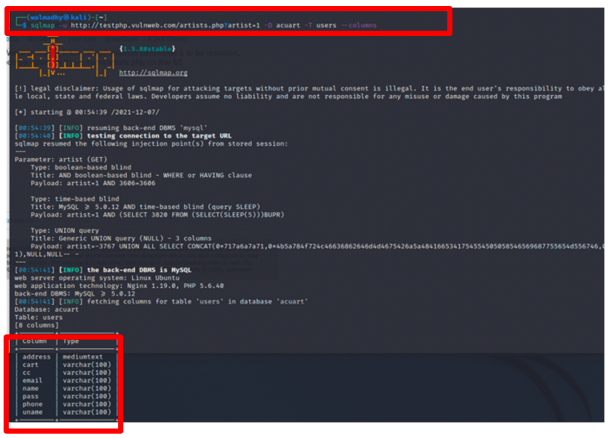

Fig. 8. Database Tables

Now as shown in Figure 9 we will extract the data in the columns of the users table in order to extract the usernames and passwords using sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart --T users -C uname,pass -dump

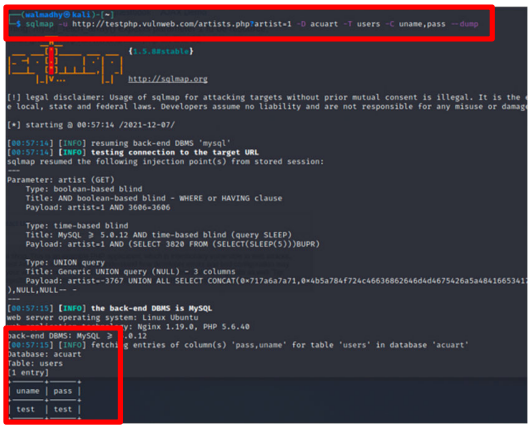

Fig. 9. SQLMAP command.

Now to be able to enter the site using the data that was extracted, we will need to show the site's control panel, so we will use a tool developed by Abdullrahman Wasfi as shown in figure 10.

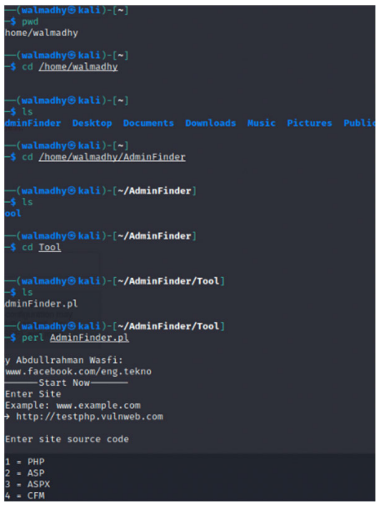

Fig. 10. AdminFinder.pl tool.

Now using the specified link in figure 11, which is the control panel link, we will be able to enter the site.

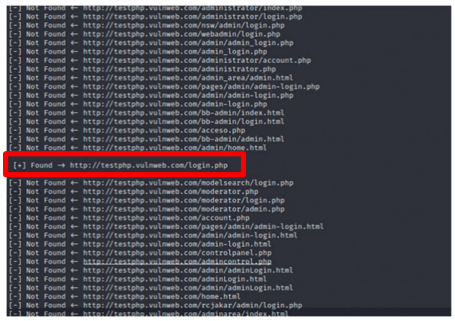

Fig. 11. Control panel link

As shown in Figures 12, 13, using the data that was extracted and the control panel, we were able to enter the site and it is possible to use all the user's privileges.

Fig. 12. Control panel

Fig. 13. The site has been successfully entered.

4. Preventive Measures for SQL Injection Attacks

4.1 Prepared Statement

One of the best ways to prevent SQLI is instead of using a statement, a prepared statement is used. One of the basic problems of SQLI is the use of user-entered inputs as an SQL statement. Instead of inserting values into the statement, the SQL statement uses a parameter using a prepared statement to insert a value into the database. Thus, the back-end database is prevented from running malicious SQL queries that will harm the database [16].

4.2 Using Stored Procedures

In contrast to using prepared data, an additional layer of security is added to the database through stored procedures. The difference between a stored procedure and a prepared statement is that in the procedure the SQL code is stored and configured in the database server and then called from the web application. It is not necessary to give permission access to the user if the user is able to access any database at any time through stored procedure strategies. In this case, the database is still secure [16].

4.3 Validating User Input

In the verification process after checking the input provided by the user must be used. In this case, first, the user input value must be checked whether it is of an acceptable type or not [16].

4.4 Limiting Privileges

The meaning here is to restrict user access to resources. When you do not need administrator access, do not connect to the database using this type of access. An account with limited access is required [16].

4.5 Encrypting Data

There is great harm in storing data in an unencrypted form, which may cause theft of this data if the permission is lost, or a malicious entry allows users to view the data. If the stored data is encrypted, the attacker will be prevented from reading the data on the databases [16].

5. Conclusion

SQLI is one of the most widespread threats to web applications, so we discussed in this research the concept of SQLI and its most famous types, we exploited the SQLI vulnerability using SQLMAP for a site that was introduced to help ethical hackers apply their skills. Finally, we mentioned different ways to protect against SQLI vulnerabilities.

Acknowledgments

The authors would like to thank the Deanship of Graduate Studies at Jouf University for funding and supporting this research through the initiative of DGS, Graduate Students Research Support (GSR) at Jouf University, Saudi Arabia.

References

[1] Ojagbule, O., Wimmer, H., & Haddad, R. J. (2018, April). Vulnerability Analysis of Content Management Systems to SQL Injection Using SQLMAP. In SoutheastCon 2018 (pp. 1-7). IEEE.

[2] Tasevski, I., & Jakimoski, K. (2020, November). Overview of SQL Injection Defense Mechanisms. In 2020 28th Telecommunications Forum (TELFOR) (pp. 1-4). IEEE.

[3] Patel, D., Dhamdhere, N., Choudhary, P., & Pawar, M. (2020, September). A System for Prevention of SQLi Attacks. In 2020 International Conference on Smart Electronics and Communication (ICOSEC) (pp. 750-753). IEEE.

[4] Rajeh, W., & Abed, A. (2017, August). A novel three-Tier SQLi detection and mitigation scheme for cloud environments. In 2017 International Conference on Electrical Engineering and Computer Science (ICECOS) (pp. 33-37). IEEE.

[5] Ping, C., Jinshuang, W., Lanjuan, Y., & Lin, P. (2020, September). SQL Injection Teaching Based on SQLi-labs. In 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE) (pp. 191-195). IEEE.

[6] Aliero, M. S., Ghani, I., Qureshi, K. N., & Rohani, M. F. A. (2020). An algorithm for detecting SQL injection vulnerability using black-box testing. Journal of Ambient Intelligence and Humanized Computing, 11(1), 249-266.

[7] Singh, S., & Kumar, A. (2020). Detection and prevention of sql injection. International Journal of Scientific Research & Engineering Trends, 6(3), 1642-1645.

[8] Li, Q., Li, W., Wang, J., & Cheng, M. (2019). A SQL injection detection method based on adaptive deep forest. IEEE Access, 7, 145385-145394.

[9] Voitovych, O. P., Yuvkovetskyi, O. S., & Kupershtein, L. M. (2016, September). SQL injection prevention system. In 2016 International Conference Radio Electronics & Info Communications (UkrMiCo) (pp. 1-4). IEEE.

[10] Hu, J., Zhao, W., & Cui, Y. (2020, February). A Survey on SQL Injection Attacks, Detection and Prevention. In Proceedings of the 2020 12th International Conference on Machine Learning and Computing (pp. 483-488).

[11] Malik, M., & Patel, T. (2016). Database security attacks and control methods. International Journal of Information, 6(1/2), 175-183.

[12] H. Alsobhi and R. Alshareef, "SQL Injection Countermeasures Methods," 2020 International Conference on Computing and Information Technology (ICCIT-1441), 2020, pp. 1-4, doi: 10.1109/ICCIT-144147971.2020.9213748.

[13] B. Appiah, E. Opoku-Mensah and Z. Qin, "SQL injection attack detection using fingerprints and pattern matching technique," 2017 8th IEEE International Conference on

Software Engineering and Service Science (ICSESS), 2017, pp. 583-587, doi: 10.1109/ICSESS.2017.8342983.
[14] Kareem, F. Q., Ameen, S. Y., Salih, A. A., Ahmed, D. M., Kak, S. F., Yasin, H. M., ... & Omar, N. (2021). SQL injection attacks prevention system technology. Asian Journal of Research in Computer Science, 13, 32.
[15] Priyadharshini, S., & Rajmohan, R. (2017). Analysis on database security model against NOSQL injection. Int. J. Sci. Res. Comput. Sci., Eng. Inf. Technol, 2(2), 168-171.
[16] Hlaing, Z. C. S. S., & Khaing, M. (2020, February). A detection and prevention technique on sql injection attacks. In 2020 IEEE Conference on Computer Applications (ICCA) (pp. 1-6). IEEE.

Waad Almadhy: Master student in jouf university received the B.E.. degrees, from jouf Univ

Amal Alruwaili: Master student in jouf university received the B.E.. degrees, from jouf Univ

Saloua Hendaoui received the B.E. and M.E. degrees, from tunis Univ. in 2011 and 2009, respectively. She received the Dr.. degree from Cartage Univ. in 2017. Working as a assistant professor (from 2018) in the Dept. of computer Science Jouf University