Marques Soares

Final Report

1. Team introduction

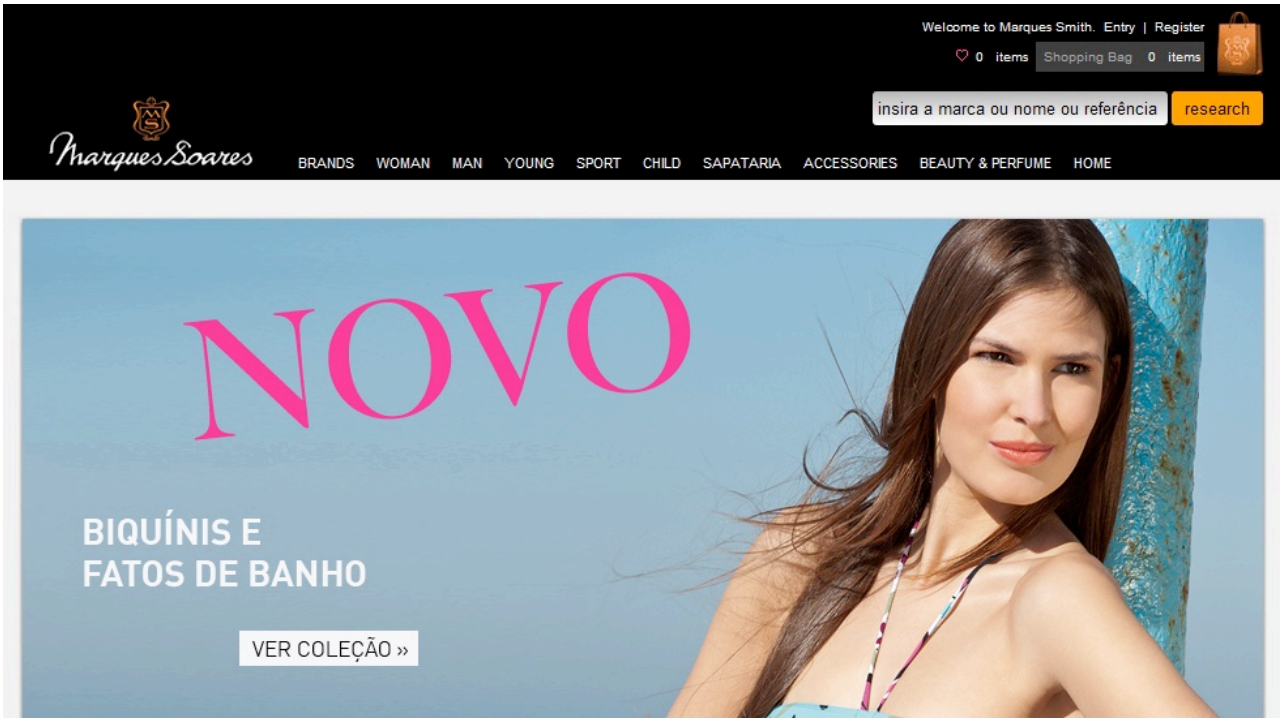
Our team consists of five members each from a different country. The team members are Irena Adamovič, Jonathan Lamy, Niko Koskimaa, Sadoht Fernandez, Hugo van der Ven.. Hugo is the only one with education in business. The others are programmers.

Niko is from Finland and he studies computer science at the Metropolia University of Applied Sciences.
Sadoht is from Spain. He studies computer sciences at University of Vigo.
Jonathan is from Belgium. He studies programming at University College of Namur - Liege - Luxembourg.
Hugo is from the Netherlands and is studying Business Informatics on the University of Applied science.
Irena is from Lithuania and is studying at Vilnius university of applied science.

2. Customer description

Marques Soares opened its door 50 years ago with an area of 150 m2 and 10 employees, the sale was intended only to clothing fabrics, knitwear and hosiery. Today has an area of 14,000 m2 and 370 employees, divided between Porto, Braga, Aveiro, Santarém, Beja, Évora and Vila Real.
In 1995 the company was awarded by the European Foundation For Entrepreneurship Research, as one of the 500 most dynamic companies in Europe.

Right now they have three distribution channels (stores, catalog sales and e-commerce), the main focus for this research was to improve sales for the e-commerce distribution channel. This is done with an website which can be reached on the following URL:
http://www.marquessoares.pt/v2/


Welcome to Marques Smith. Entry | Register
0 items
Shopping Bag 0 items
Marques Soares
insira a marca ou nome ou referência
research
BRANDS WOMAN MAN YOUNG SPORT CHILD SAPATARIA ACCESSORIES BEAUTY & PERFUME HOME
NOVO
BIQUÍNIS E
FATOS DE BANHO
VER COLEÇÃO »

3. Case data

During the interviews made with the representatives of the marketing and business-IT sectors we found out that the 90% of the sales are made in their stores. From the rest 10%, only the 10% are made using their web site. In the other sales customers use the catalog to check the products and then they order them by email, phone, etc.

In their goals list was, at least, to double the web site revenue. The following goals were stated as well by the company:

1. We want to be the best in Portugal on the internet sales;
2. At least double our web site sales;
3. Attract new customers;
4. Keep good relationship with customers;
5. We only offer discounts to loyal clients;
    a. 5% discount paying by cash and only through catalog or internet;
6. We don't need paypal because 90% of our customers pay with Marques Soares card;
7. Catalog costs 5€, but they give back the money for the catalog if they buy something.

Marques Soares gave us their online sales data. It had almost 13 000 rows of purchase data. Each row contained information about the customer, the product ordered and the order itself. Unfortunately we did not receive any documentation describing the dataset. Another unfortunate thing was that we didn't receive the sales data from the real shops. If we had got the shops' sales data we could have compared that data with the webshop data and achieve better results.

We also got the web server's log files, but we didn't use them at all, because we got the Google Analytics account for the web site. We could have done some analysis with the log files that we could not do with the Google Analytics, but we thought that analysing the logs would take too much time we couldn't spare.

Example of the given Excel sales data

The following data was given in the format of an excel file containing 12762 rows from the period september 2009 until march 2012.

Customer number	Number of an specific customer. Example: 201245
Date of birth	Birthday of the customer. Formatting: year-month-day 19511024
Postal code	Postal code of the customer. 8005-421
City/Locality	City of the customer. Example: Porto
Customer Honorific Title	Customer title. Example: MSc

Customer admission date (yyyymmdd)	Admission date of the customer. Formatting: year-month-day 19511024
Payment Method:	1-Credit card; 3 and 8-Bank account debit; 5-Payment on postal receipt; 7-Cash; 26 and 27 - ATM payment. Example: number
Product code	Ordered product number. Example: 93939393
Color	Color of the ordered product in short codes. Example: UNI
Size	Size of the ordered product, can be numbers or letters. Example: 38 or XL
Product description	Description of the ordered product. Example: Pyjamas
Fall	Catalog period. Example: year-year 2011-2012
Fall description	Season of the catalog. Example: Outono-Inverno
Brand	Brand of the ordered product. Example: Camel
Order year	Year of the order. Example: 2011
Order month	Month of the order. Example: 11
Order day	Day of the order. Example 1
Order Number	Unique code for defining the customer's order within specific time interval. Example: 83838
Number of months considered for payment:	1-Payment of the whole amount; >1-Payment divided in parts. Example: 10
Product reference in the catalogue	Reference number from the catalog. Example: 93933
Store department:	SAP-Shoes; HOM-Men; SRA-Women; GAN-Youngsters; REL-Watches; CRI-Children; LIN-Lingerie; DSP-Sports; EST-Home textiles; ELE-Electrical appliances; MRQ-Leather goods; PER-Perfumes; PRA-Beachwear; ACS-Accessories. Example: SAP
Unit price (Euros)	Product price in euros including tax. Example: 160
Customer Gender	Gender of the customer. Example: F
Customer Profession	Profession of the customer. Example: PROFESSORA - ESC.D.MARIA II
Customer Type:	7-Cash Customers (customers who pay the whole amount of the order); 1, 0 and 9- Customers who use credit for payment. Example: 1

4. Processing and analysis of data

We mainly analyzed the sales data using Excel and QlikView. We gathered useful information by aggregating data and making graphs. !!!!!!!!!!!!data mining

4.1 Data analysis

Exel

Using Excel we were able to organize the data and make some statistic charts showing the current situation of the company.

Dashboard created in QlikView

We created dashboard, which allowed us to better analyze the data and make some ear future predictions.

Website analysis

After the analysis of the given sales data we also did an check on the security of the website and its validation. The following section mainly contains problems we found inside the current webshop (http://www.marquessoares.pt/v2/) combined with possible improvements.

4.2 Boolean Based SQL Injection

SQL Injection occurs when data input for example “by a user” is interpreted as a SQL command rather than normal data by the backend database. This is an extremely common vulnerability and its successful exploitation can have critical implications. We confirmed the vulnerability by executing a test SQL Query on the back-end database. In these tests, SQL Injection was not obvious but the different responses from the page based on the injection test allowed us to identify and confirm the SQL Injection.

Summary

Severity :	Critical
Confirmation :	Confirmed
Vulnerable URL :	http://www.marquessoares.pt/v2/search.php?text=-1 OR 17-7=10
Parameter Name:	text
Parameter Type:	Querystring
Attack Pattern:	-1 OR 17-7=10

Impact

Depending on the backend database, the database connection settings and the operating system, an attacker can mount one or more of the following type of attacks successfully:

- Reading, Updating and Deleting arbitrary data from the database;

- Executing commands on the underlying operating system;
- Reading, Updating and Deleting arbitrary tables from the database.

Actions to Take

1. See the remedy for solution.
2. If you are not using a database access layer (DAL), consider using one. This will help you to centralise the issue. You can also use an ORM (*object relational mapping*). Most of the ORM systems use only parameterised queries and this can solve the whole SQL Injection problem.
3. Locate all of the dynamically generated SQL queries and convert them to parameterised queries. (*If you decide to use a DAL/ORM change all legacy code to use these new libraries*)
4. Use your weblogs and application logs to see if there was any previous but undetected attack to this resource.

Remedy

The best way to protect your code against SQL Injections is using parameterised queries (*prepared statements*). Almost all modern languages provide built in libraries for this. Wherever possible do not create dynamic SQL queries or SQL queries with string concatenation.

Required Skills for Successful Exploitation

There are numerous freely available tools to exploit SQL Injection vulnerabilities. This is a complex area with many dependencies, however it should be noted that the numerous resources available in this area have raised both attacker awareness of the issues and their ability to discover and leverage them.

4.2 Password Transmitted Over HTTP

Summary

Severity :	Important
Confirmation :	Confirmed
Vulnerable URL :	http://www.marquessoares.pt/v2/cart.php
Form target action:	login.php

Impact

If an attacker can intercept network traffic he/she can steal user credentials.

Actions to Take

1. See the remedy for solution.
2. Move all of your critical forms and pages to HTTPS and do not serve them over HTTP.

Remedy

All sensitive data should be transferred over HTTPS rather than HTTP. Forms should be served over HTTPS. All aspects of the application that accept user input starting from the login process should only be served over HTTPS.

4.3 Cross-site Scripting

XSS (Cross-site Scripting) allows an attacker to execute a dynamic script (*Javascript, VbScript*) in the context of the application. This allows several different attack opportunities, mostly hijacking the current session of the user or changing the look of the page by changing the HTML on the fly to steal the user's credentials. This happens because the input entered by a user has been interpreted as HTML/Javascript/VbScript by the browser.
XSS targets the users of the application instead of the server. Although this is a limitation, since it allows attackers to hijack other users' session, an attacker might attack an administrator to gain full control over the application.

Summary

Severity :	Important
Vulnerable URL :	http://www.marquessoares.pt/v2/search.php?text=<script>ns(0x00009B)</script>
Parameter Name:	text
Parameter Type:	Querystring
Attack Pattern:	<script>ns(0x00009B)</script>

Impact

There are many different attacks that can be leveraged through the use of XSS, including:

- Hijacking users' active session.
- Changing the look of the page within the victim's browser.
- Mounting a successful phishing attack.
- Intercept data and perform man-in-the-middle attacks.

Remedy

The issue occurs because the browser interprets the input as active HTML, Javascript or VbScript. To avoid this, all input and output from the application should be filtered. Output should be filtered according to the output format and location. Typically the output location is HTML. Where the output is HTML ensure that all active content is removed prior to its presentation to the server.
Prior to sanitizing user input, ensure you have a pre-defined list of both expected and acceptable characters with which you populate a white-list. This list needs only be defined once and should be used to sanitize and validate all subsequent input.
There are a number of predefined, well structured white-list libraries available for many different environments, good examples of these include, OWASP Reform and Microsoft Anti Cross-site Scripting libraries are good examples.

4.4 Database User Has Admin Privileges

We identified that the target web site is connecting to the backend database by using a user that has administrative privileges. This issue has been **confirmed** by checking the connection privileges via an identified SQL Injection vulnerability in the application.

Summary

Severity :	Important
Confirmation :	Confirmed
Vulnerable URL :	http://www.marquessoares.pt/v2/search.php?text=-1 OR 17-7=10
Parameter Name:	text
Parameter Type:	Querystring
Attack Pattern:	-1 OR 17-7=10

Impact

This can allow an attacker to gain extra privileges via SQL Injection attacks. Here is the list of attacks that the attacker might carry out:

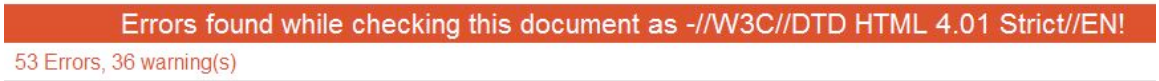
- Gain full access to the database server.
- Gain a reverse shell to the database server and execute commands on the underlying operating system.
- Access the database with full permissions. Where it may be possible to read, update or delete arbitrary data from the database.
- Furthermore, depending on the platform and the database system user an attacker might carry out a privilege escalation attack to gain administrator access to the target system.

Remedy

Create a database user with the least possible permissions for your application and connect to the database with that user. Always follow the principle of providing the least privileges for all users and applications.

4.5 Site validation

The current webshop is not comforting the current web standards as stated by the W3C group, these problems can be easily fixed by using the free to use HTML and CSS validators.



HTML validator: http://validator.w3.org/
CSS validator: http://jigsaw.w3.org/css-validator/

The combination of improved metatags and validated site will greatly increase the findability of the site on Google and other search-engines.

5. Conclusions and suggestions

The company should put more keywords for search in their page to increase probability of finding their page through internet search. They also should concentrating on specific directions and define concrete goals, which can improve their performance. They should use web mining and other new technologies .