Valerie Angulo

Practical Computer Security Assignment 2

1. Explain the approaches used to gain unauthorized access to the system. Also, identify three vulnerabilities that you can take advantage of to gain unauthorized access to the system.

My approach is to use a web proxy (Burp Suite) to do intruder attacks on the user/login by putting obvious username pairs with a random password. Once the username is verified by “password not recognized”, passwords can be tested with the correct username. A login vulnerability that I can exploit is that wrong credentials are specified, so I can see if a username is correct or not. Once I know the proper username, I can then utilize Burp Suite to test various inputs for passwords until one works. Three vulnerabilities that I could take advantage of to gain access to the system are the ability to set cookies, modify HTML forms and the design flaws of the failed logins. If I had the cookie of someone who successfully logged in, I could change my cookie to theirs and login as them without knowing their username or password. There is also an auto login function, so once I get the session ID cookie of someone, it will automatically log in unless they logged out. I could modify the GET and POST headers to manipulate what I want the server to receive. I could also set the number of failed logins to a higher number and attempt various usernames/passwords and the website will tell me whether I got the username incorrect or the password incorrect.

2. For each vulnerability identified in Q1, explain how you would remove such vulnerability and improve the security of the system.

To remove the vulnerability of being able to set cookies, I would I would have to pay to have the website over the more secure HTTPS domain so that the headers of the cookies would be encrypted. I would also not have an auto login function, or I could have the session terminate after a certain amount of time so that if the cookie session ID was obtained you would still need to know the username/password. To remove the vulnerability of manipulating the GET and POST HTTP requests, I would also have to have the website over the more secure HTTPS domain. To remove the vulnerability of flawed design, I would say “Invalid username/password” instead of revealing which was wrong.

3. After you have successfully logged in, go to the ‘search’ page. The search page queries a list of employee names based on the inputted keywords. Identify a possible SQL statement used for this query.

SELECT Employee_Name FROM Search_Table WHERE Employee_Name = ‘keyword’

4. The next task is to use SQL injection to display all the names of the employees in the web application, how would you achieve this? Explain how you achieve this with step-by-step screenshots.

I would inject into the keyword input **’ OR 1=1 - -** so that the query will be true no matter what, returning all employee names.

Ex: Employee_Name = ‘ **’ OR 1=1 - -** ’

5. Your task is now to get hold of ALL employees’ social security numbers. The social security numbers are stored as a 9 numerical digits. How would you do this? Explain how you achieve this with step-by-step screenshots.

I would do an SQL injection attack where I would try to change the query to include the column containing an employees SSN. I would try to see what other columns are in the database so that I could join columns by injecting more queries into user input fields.

If query is something like: SELECT Employee_Name FROM Search_Table WHERE Employee_Name = ‘keyword’ then I could submit a query in the keyword field such as:

’ OR 1=1; JOIN SELECT SSN FROM Search_Table WHERE SSN = ‘’ OR 1 = 1; - -

I can test to see if the query is SELECT * FROM Search_Table WHERE Employee_Name = ‘keyword’ instead, which would give me access to all rows of an employee, I would just have to figure out the headers for the SSN row.

6. You would also like to get hold of the username and hashed passwords of the user and admin accounts of the web application. How would you accomplish this task? What are the username and hashed passwords for all the accounts? Explain with step-by-step screenshots.

’ OR 1=1; JOIN SELECT username, password FROM Search_Table WHERE username = ‘’ OR 1 = 1; - -

In keyword, I could try to access admin information by inputting **admin’--** into the keyword input.

7. If logged on as a regular user (using an account identified in Q6 in a ‘user’ group), can you find way(s) to gain elevated access to a resource that is protected from users, i.e. to access an administrative function? For this, you must identify a page that only an admin can perform the administrative function. Explain in detail how the privilege can be escalated. You should be presented with a pop-up window that says ‘SUCCESS’, if you can successfully perform the administrative function as an admin.

To gain elevated privilege you need the sessionID cookie value of an administrator. You can then modify your cookie to be the administrators cookie and gain control of their session. You can also see what cookies are given to a regular user and try to put in values for this ID such as 1 to see if you gain admin access.

8. Identify two entry points that are vulnerable to a cross-site scripting attack. Explain in detail on how the vulnerability is detected.

Two entry points that are vulnerable to a cross-site scripting attack are unvalidated input coming into web applications and output to the browser that is not HTML encoded. This vulnerability is exploited through client-side scripting languages such as HTML and JavaScript and HTTP headers in HTTP requests. For example, scripts can be hidden in HTTP headers or embedded in user input. The vulnerability of client-side input can be detected by logging everything on your servers to check for anything out of the ordinary. You can also do XXS tests for detection or testing methods such as checking for unsanitized user input being utilized in a script, open brackets, “”, non-numerical or alphabet characters and other weird input that can be entered by a user to make sure that your code isn’t vulnerable to XXS, using whitespacing to limit what a user can input as appropriate data.

9. Out of the two entry points that you identified in Q8, which one would you choose to use as part of a social engineering attack. Explain.

I would use output to the browser as part of a social engineering attack because I could craft a link that looks like its coming from a certain site and send it in an email to a user. If they click on the link, they will be directed to my attacker version of the site, where any input or credentials they put in will be able to be seen and used by me.

10. From Q9, craft a link so that when it is clicked by a victim, he/she will be presented with a page that asks for the site’s user login and passwords. The page should not be redirected to somewhere else. The URL shown in the browser needs to begin with ‘www.myvulnerablesite.com/learn/’. When the login credentials are entered or submitted, they will be sent to the attacker’s side. The attacker’s side must be able to receive requests sent from the page. You can implement this part by any means and in any web programming language you prefer. You also need to create a web page that can display the IP of the victim and the inputs captured. For this task you may need to pay some (small) fee to host your web, if you don’t already have one. You need to submit the link to be submitted to a victim, and the link on how to access the attacker’s web page. Full points will be given when login credentials (from the victim browser) can be displayed on the attacker’s website that you provide.

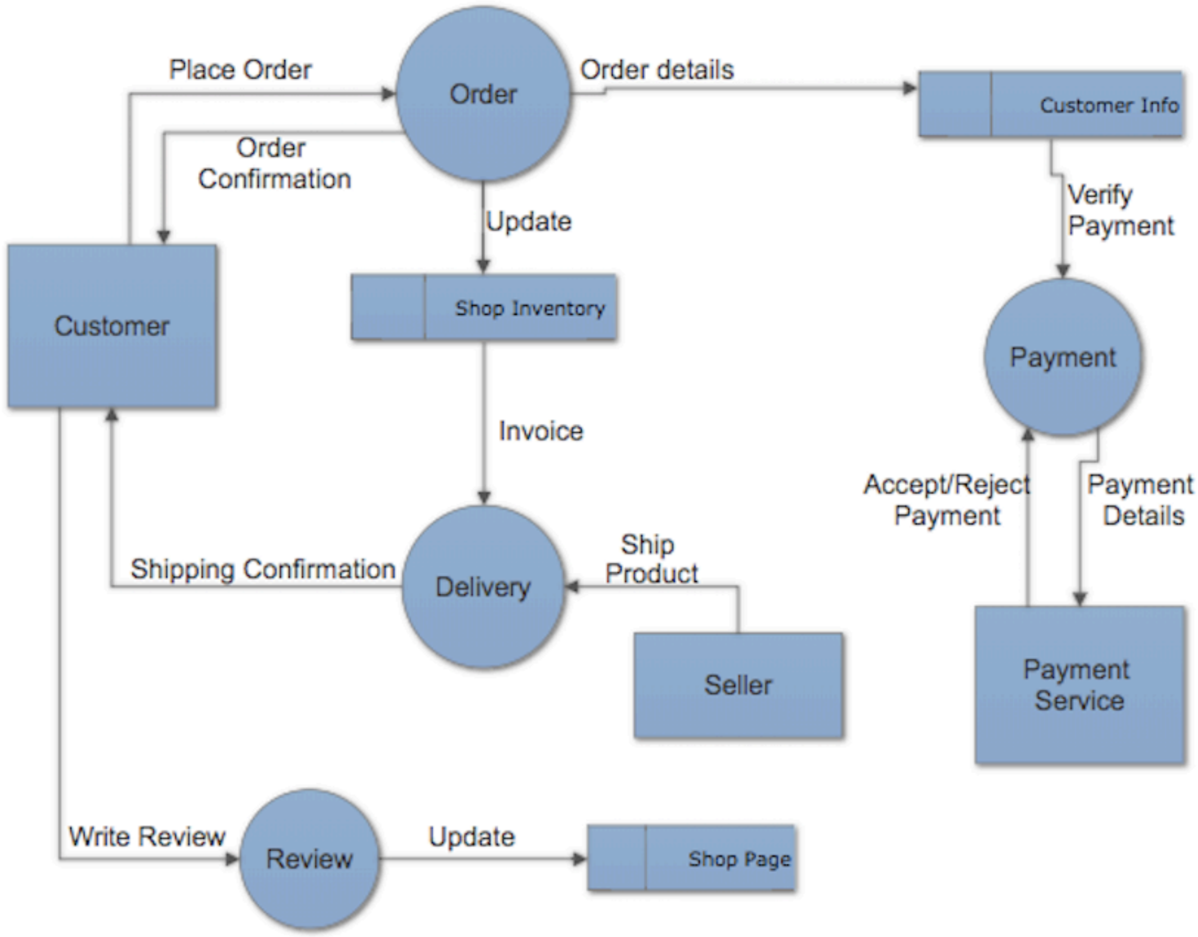
I don’t know any web programming languages or how to make a web page but this is the site I would have someone sent to if they clicked my link:

www.myvulnerablesite.com/learn/login+%3Cscript%3Eattack_script()%3C/script%3E

11. For the threats/vulnerabilities that you identify in Q8-10, explain in detail how this can be prevented by a developer and by a user of the website.

Countermeasures for this type of attack is input validation where a developer should never trust any input coming in from the client side, should always check/sanitize the input and not build forms that take in unvalidated input or use client side input for other functions without sanitizing it. Code such as onfocus=“alert(document.cookie)” could be helpful as well, so that the user is notified if their cookie is being compromised, as a warning of potential unwanted data retrieval and they can end their session and log out so that the cookie cannot be used to login without credentials. Developers can make sure their cookies and tokens are encrypted and don't hold any important data, and aren't linked to any particular user. To prevent reflected XXS attacks, the user must be vigilant with the source of links sent to them in e-mails, especially links sending them to a page requesting very serious data or a website that would result in huge loss of important personal info if login credentials were compromised.

12. Perform a threat modeling for Etsy. Limit your analysis to the following features: Selling & buying of goods / providing reviews / processing of payment . The deliverables include: a level-1[1] Data Flow Diagram (DFD) of the system and list of threats identified using STRIDE-per-element.



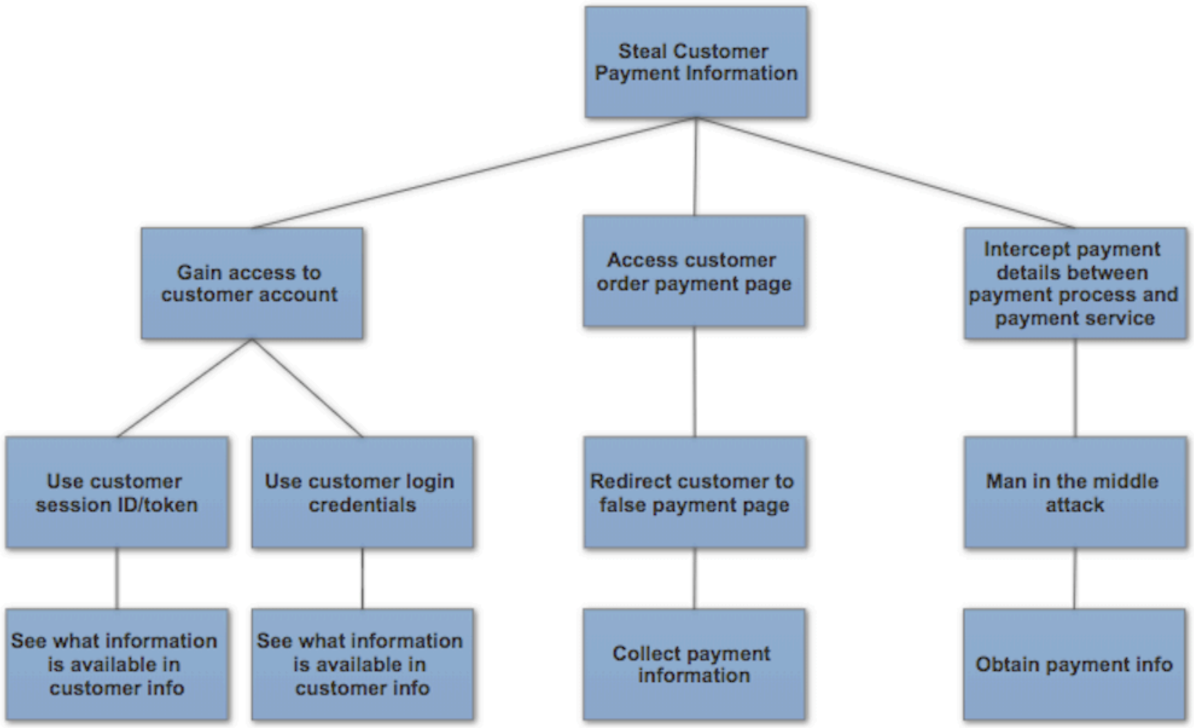
Element	S	T	R	I	D	E
Customer	-attacker account -attacker with customer login credentials -attacker with customer session token/ID	-attacker modifies customer account -attacker deletes customer account	-customer says they aren't who they are	-customer discloses info not on the site	-customer cannot access account -site is down	-customer gains special customer privileges -premium account upgrade
Seller	-attacker account -attacker with seller account credentials -attacker with seller session token/ID	-attacker modifies seller account -attacker deletes seller account	-seller says they aren't who they are	-seller discloses info not on the site	-seller cannot access account -site is down	-seller gains special seller account privileges -premium account upgrade
Payment Service	-attacker poses as payment service -unsafe payment service -attacker poses as customer to payment service -man in the middle attack between payment and payment process	-attacker modifies payment info on service -attacker inputs bad input onto payment service	-customer says they never received refund -customer says they weren't charged correct amount -seller says payment info is not valid	-attacker has access to customers payment service account -attacker exposes customers payment service account info	-payment service is down -customer cannot access service -seller cannot access service -seller cannot be paid	X

Element	**S**	**T**	**R**	**I**	**D**	**E**
Order	-order page is attacker page -order page contains malevolent script	-attacker modifies order -attacker deletes order -attacker makes more orders	-customer says they never ordered item -customer says they returned item -seller saying they didn't receive return -seller says they sent order	-order invoices are disclosed -attackers can see who is a good target based on how pricey their orders are	-site is down -customer cannot access site -seller cannot access site -customer cannot make order -seller cannot receive order	-customer gains eligible multiple order privileges
Payment	-attacker script on payment service -attacker redirects customer to attacker payment page -man in the middle attack between payment and payment process	-charging customer more than what was said -selling item for less than what was said	-customer says they never received refund -customer says they weren't charged correct amount -seller says they weren't paid correct amount -customer says they paid but were never sent confirmation	-payment information is collected by attacker -payment information is sold by attacker -payment information is used by attacker	-site is down -customer cannot access site -seller cannot access site -customer cannot make payment -seller cannot receive payment	-Customer gains eligible discount privileges
Delivery	-site isn't safe -attacker is in control of delivery	-delivery methods modified -delivery process halted -delivery process terminated	-customer says they never received item -seller says they never received return	-delivery info exposed -shipping/ billing info disclosed	-customer cannot receive item -seller cannot ship item -invoice not received -site is down	-customer gains shipping privileges

Element	S	T	R	I	D	E
Review	-attacker writes review with attacker account -attacker writes/ deletes review through customer account	-attacker writes/ deletes review through customer account	-customer says they did not write review	-customer can disclose sensitive info over review -attacker can disclose sensitive info over review	-site is down -customer cannot write review -other shoppers cannot read reviews -seller cannot respond/ view review	-customer can submit review without having bought item
Customer Info	-attacker puts their info to receive products	-deleting customer info -modifying customer info -deleting customer account	-customer denies inputing info -customer says info was changed	-exposing customer data	-customer cannot access their info -site is down	X
Shop Inventory	-sellers shop inventory is actually another sellers inventory	-removing items from inventory -adding items to inventory -deleting inventory	-seller says they never put certain item in shop inventory -seller saying they didnt have item in shop	-inventory items info disclosed	-seller cannot access shop -site is down	-seller gains shop privileges -customer gains access to shop inventory
Shop Page	-shop page is redirected to attacker page -shop page contains malevolent script	-taking down all sellers items in shop -deleting seller info -modifying seller info -deleting seller account	-seller says they did not put info that is on shop page	-seller information disclosed on page to public	-customer cannot view site page -seller cannot access site page -other buyers cannot access site page	-seller gains shop privileges -customer gains access to a shop

13. Draw an attack tree[2] for one threat identified in Q12 that you think that has the highest risk. Explain also why you think so. Your attack tree should show different ways to accomplish the attack goal.

I think the threat with the highest risk for Etsy is having an attacker steal customers payment information because an attacker would be able to access the payment information from many customers through Etsy, ruining the sites reputation and disclosing and using the information for various nefarious purposes outside of the site. It would harm both the business and the customers involved.


Steal Customer Payment Information
Gain access to customer account
Access customer order payment page
Intercept payment details between payment process and payment service
Use customer session ID/token
Use customer login credentials
Redirect customer to false payment page
Man in the middle attack
See what information is available in customer info
See what information is available in customer info
Collect payment information
Obtain payment info