Threat Modeling Report

Created on 11/15/2020 5:16:08 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	0
Not Applicable	15
Needs Investigation	0
Mitigation Implemented	13
Total	28
Total Migrated	0

Diagram: Diagram 1

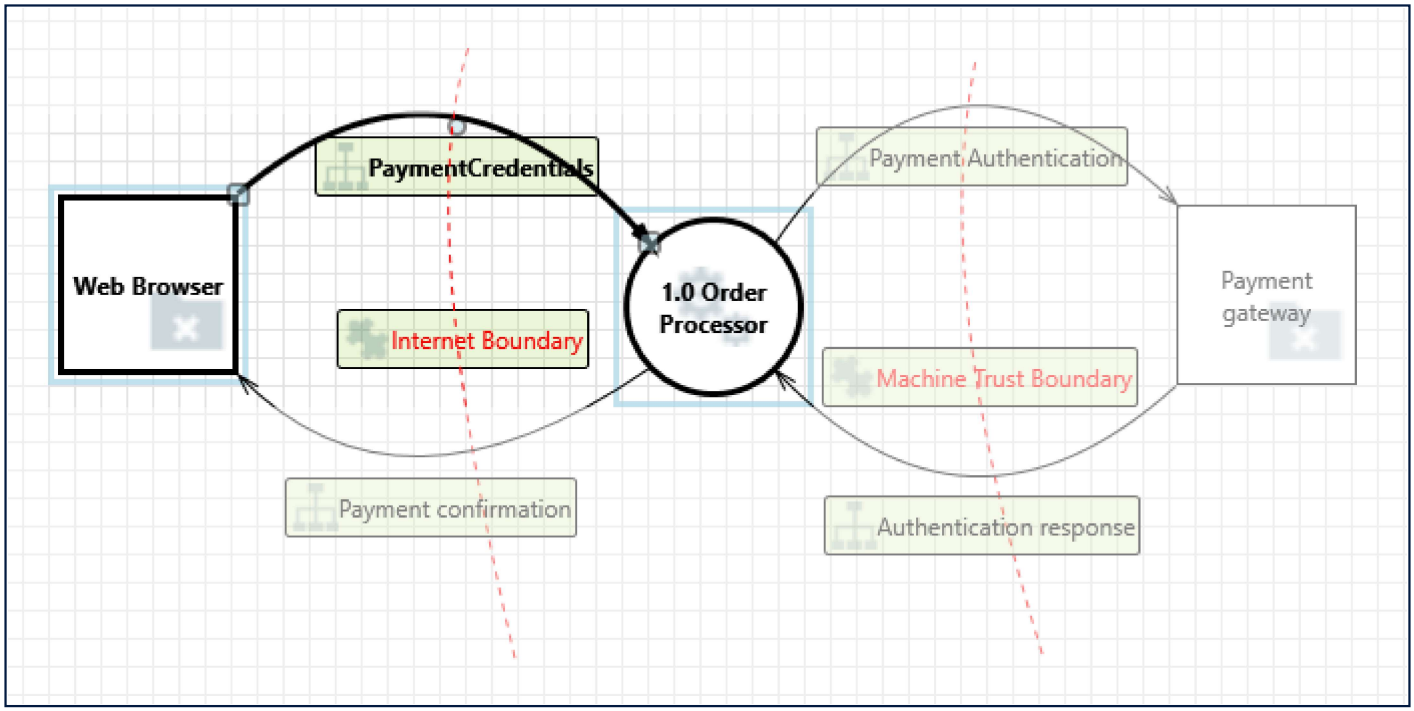
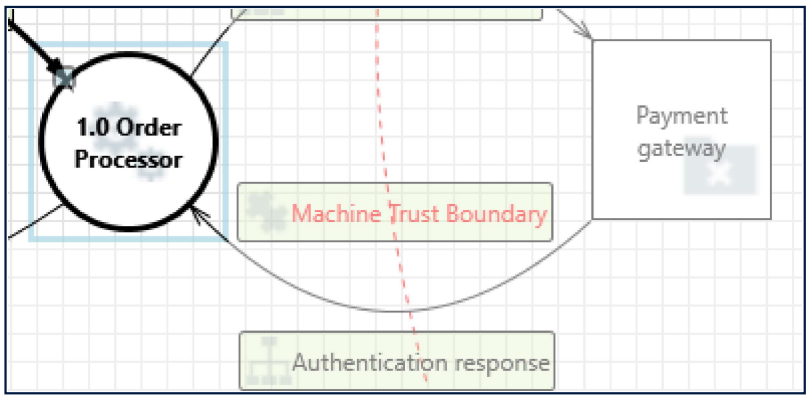


Diagram 1 Diagram Summary:

Not Started	0
Not Applicable	15
Needs Investigation	0
Mitigation Implemented	13
Total	28
Total Migrated	0

Interaction: Authentication response



1. Spoofing the Payment gateway External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Payment gateway may be spoofed by an attacker and this may lead to unauthorized access to 1.0 Order Processor. Consider using a standard authentication mechanism to identify the

external entity.

Justification: Magento allows only valid payment gateways like Paypal.

2. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: 1.0 Order Processor may be able to impersonate the context of Payment gateway in order to gain additional privilege.

Justification: Magento uses robust encryption algorithms and SSL certificates to secure the sever and data so it is difficult to impersonate.

3. Spoofing the 1.0 Order Processor Process [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: 1.0 Order Processor may be spoofed by an attacker and this may lead to information disclosure by Payment gateway. Consider using a standard authentication mechanism to identify the destination process.

Justification: Not applicable to Magento. Each Payment gateway has its own authorization mechnism.

4. Potential Lack of Input Validation for 1.0 Order Processor [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Authentication response may be tampered with by an attacker. This may lead to a denial of service attack against 1.0 Order Processor or an elevation of privilege attack against 1.0 Order Processor or an information disclosure by 1.0 Order Processor. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.

Justification: The Order Processor has a Response Validator component that validates the responses from payment gateway.

5. Potential Data Repudiation by 1.0 Order Processor [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: 1.0 Order Processor claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: Magento has a logging application integrated with the Order Processor to keep a track of all the requests and responses.

6. Data Flow Sniffing [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure

Description: Data flowing across Authentication response may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.

Justification: Magento uses Advanced Encrytion Standard to encrypt all sensitive data. So the data flowing across would be encrypted and protected.

7. Potential Process Crash or Stop for 1.0 Order Processor [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service

Description: 1.0 Order Processor crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Web Server is seperatly handled in the magento architecture , thus it can handle data well without any kind of crashes.

8. Data Flow Authentication response Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: Not in the scope of the payment module, it is handled by the SSL module.

9. 1.0 Order Processor May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Payment gateway may be able to remotely execute code for 1.0 Order Processor.

Justification: Not in the scope of payment module , handled by the admin module.

10. Elevation by Changing the Execution Flow in 1.0 Order Processor [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: An attacker may pass data into 1.0 Order Processor in order to change the flow of program execution within 1.0 Order Processor to the attacker's choosing.

Justification: Not in the scope of the payment module, it is handled by the SSL team.

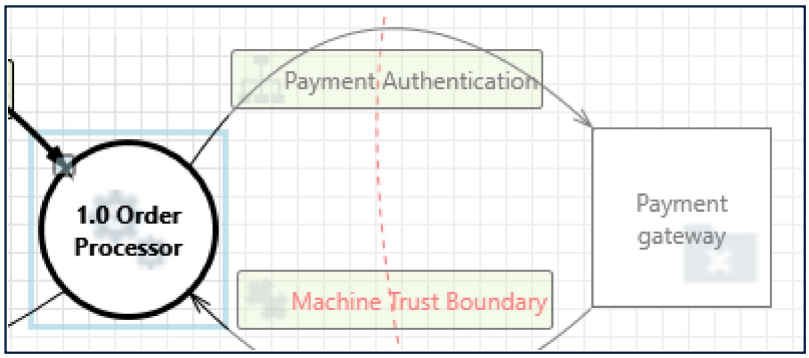
11. Cross Site Request Forgery [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The other browses to web site B. Web site B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g. by session cookie, integrated authentication, IP whitelisting. The attack can be carried out in many ways such as by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable web site that the victim will visit. The issue can only be resolved on the server side by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) which is known only to the legitimate web site and the browser and which is protected in transit through SSL/TLS. See the Forgery Protection property on the flow stencil for a list of mitigations.

Justification: It is covered in the SSL threat module. The payment module is secured under SSL.

Interaction: Payment Authentication



12. Spoofing of the Payment gateway External Destination Entity [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: Payment gateway may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Payment gateway. Consider using a standard authentication mechanism to identify the external entity.

Justification: It is beyond the scope of payment module, it is taken care by the admin team.

13. External Entity Payment gateway Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Payment gateway claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: Magento follows PCI complaince , so it ensures to keep a track of all the payment records.

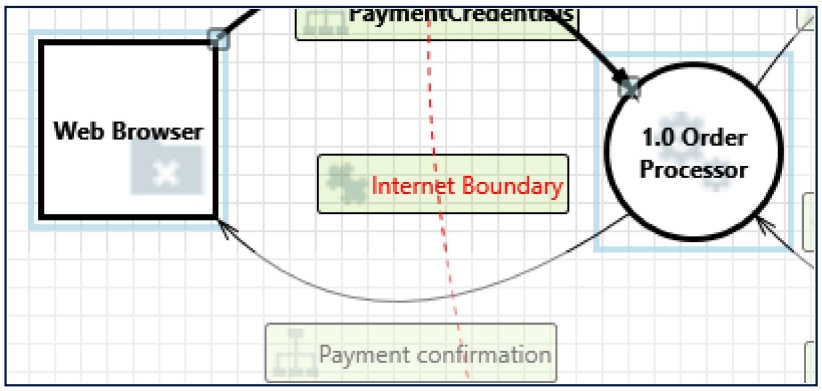
14. Data Flow Payment Authentication Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: Not in the scope of the payment module

Interaction: Payment confirmation



15. Spoofing of the Web Browser External Destination Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Web Browser may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Web Browser. Consider using a standard authentication mechanism to identify the external entity.

Justification: The Order Processor validates its users when it gets the requests and then it maintains the session tokens and transfer data only during the valid sessions.

16. External Entity Web Browser Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category: Repudiation

Description: Web Browser claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: Order Processor logs all the transaction details sent and recieved from the browser.

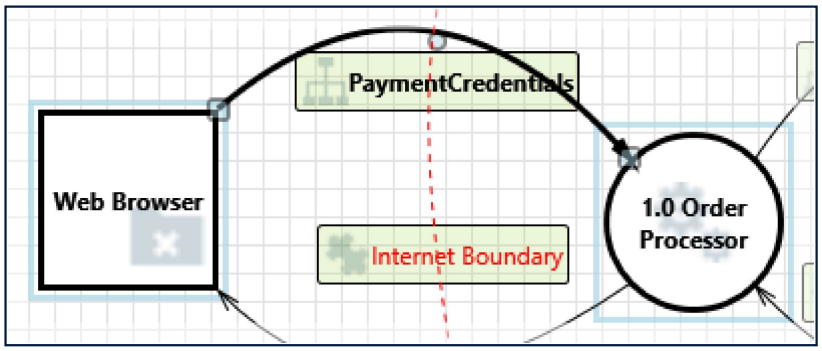
17. Data Flow Payment confirmation Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: Not in the scope of the payment module, it is handled in SSL module.

Interaction: PaymentCredentials



18. Spoofing the Web Browser External Entity [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: Web Browser may be spoofed by an attacker and this may lead to unauthorized access to 1.0 Order Processor. Consider using a standard authentication mechanism to identify the external entity.

Justification: Magento authorizes all its users before continuting the transactions. It has been handled in the Login module.

19. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: 1.0 Order Processor may be able to impersonate the context of Web Browser in order to gain additional privilege.

Justification: Only legit payment gateways are present in Magento for integration. Payment gateways validate the user payment information with help of OTP.

20. Spoofing the 1.0 Order Processor Process [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: 1.0 Order Processor may be spoofed by an attacker and this may lead to information disclosure by Web Browser. Consider using a standard authentication mechanism to identify the destination process.

Justification: The complete data that comes from the Order process is encrypted, the attacker cannot take leverage of the data.

21. Potential Lack of Input Validation for 1.0 Order Processor [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across PaymentCredentials may be tampered with by an attacker. This may lead to a denial of service attack against 1.0 Order Processor or an elevation of privilege attack against 1.0 Order Processor or an information disclosure by 1.0 Order Processor. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.

Justification: Magento has in built input validation system that handles the user data. Based on this validation it will be passed to the later stages.

22. Potential Data Repudiation by 1.0 Order Processor [State: Not Applicable] [Priority: High]

Category: Repudiation

Description: 1.0 Order Processor claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: It is handled by the login module of threat modelling

23. Data Flow Sniffing [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure

Description: Data flowing across PaymentCredentials may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.

Justification: Magento uses robust encryption algorithms and SSL certificates to secure the sever and data so it is difficult to impersonate.

24. Potential Process Crash or Stop for 1.0 Order Processor [State: Not Applicable] [Priority: High]

Category: Denial Of Service

Description: 1.0 Order Processor crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: Handled in the Admin module, not in the scope of payment module.

25. Data Flow PaymentCredentials Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: It has been handled in the other threat module, not in the scope of payment module.

26. 1.0 Order Processor May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Web Browser may be able to remotely execute code for 1.0 Order Processor.

Justification: Not in the scope of Payment module, it is handled in the login module

27. Elevation by Changing the Execution Flow in 1.0 Order Processor [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: An attacker may pass data into 1.0 Order Processor in order to change the flow of program execution within 1.0 Order Processor to the attacker's choosing.

Justification: Magento implements intrusion detection systems to monitor any kind of anamalous activity.

28. Cross Site Request Forgery [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege

Description: Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The other browses to web site B. Web site B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g. by session cookie, integrated authentication, IP whitelisting. The attack can be carried out in many ways such as by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable web site that the victim will visit. The issue can only be resolved on the server side by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) which is known only to the legitimate web site and the browser and which is protected in transit through SSL/TLS. See the Forgery Protection property on the flow stencil for a list of mitigations.

Justification: It is covered in the SSL threat module. The payment module is secured under SSL.