**UAT Checkpoints for Checkout**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Job** | UAT Prerequisite | **Date:** |  | |
| **PGW API** | Checkout | **Version:** | 1.2.0 | |
| **Merchant Name:** | Nagadhat Bangladesh Limited | **Merchant Wallet:** | 01765000009 | |
|  | | | | |
| **Check Points** | **Normal Scenario** | **Abnormal Scenario** | | **Checked (Yes/No)** |
| Environment follows the requirement of UAT | 1. Using bKash PGW Production credentials. 2. Site/App Environment –  * If Staging environment, then Staging environment shows exact replication of Production environment. * If Production Environment, then bKash payment is available to a specific test user account.  1. General customers do not have any access to bKash payment system. | 1. Using bKash PGW Sandbox credentials. 2. Staging environment does not replicate production environment properly. 3. General customers do have access to bKash payment system. | | **Yes** |
| Payment Start Page (Merchant Site) | Pay with bKash button must be “Pay with bKash” logo provided by KAM following branding guideline. | 1. Proper bKash logo is not placed. 2. Low resolution Logo place. | | **Yes** |
| Success case, Create Payment: | 1. API Version is 1.2.0 2. API request Parameter should be passed as guided in dev portal. 3. Should be called once for normal scenario. | 1. API Version is not 1.2.0 2. Extra parameter found in API request call. 3. Multiple API call found without any reason. | | **Yes** |
| Success case, Execute Payment: | Should be called once for normal scenario after wallet validation confirmation from bKash side. | 1. Multiple call found without any reason. 2. Without any reason Query Payment API called. 3. Without any reason Search Transaction API called. | | **Yes** |
| Return Page (Merchant Site) after successful payment | Showing a message like “Payment is successful”.  For popup message, popup should show at least 6sec or it has a close button for user interaction. | 1. Not showing any success message. 2. Showing “Payment is processing**”** type of message. 3. Customer cannot track this payment. | | **Yes** |
| Return Page (Merchant Site) if user manually Close the bKash window. | It should go back to payment page to re-initiate the payment. | 1. Return to a page where payment buttons are there but inactive or not clickable. 2. Without going to home page or different page customer cannot do anything. 3. Without any reason Execute Payment API called. 4. Without any reason Query Payment API called. 5. Without any reason Search Transaction API called. | | **Yes** |
| Query Payment API Implementation | Query Payment API implemented properly -  If there is no response from Execute Payment API, after **30sec** API timeout, calling Query Payment API to know the status of the pending payment.   * If the status is "**Initiated**", trying from Create Payment again. * If the status is "**Completed**", then providing success response to the customer. |  | | **Yes** |
| Search Transaction | Implemented and getting proper response in Search Transaction API. | 1. Not implemented 2. Not getting proper response in Search Transaction API. | | **Yes** |
| API Response Timeout | **30sec** API timeout applied for each bKash PGW API. Means waiting for each API responses for **30sec**. | 1. API Timeout not applied. 2. Wrong API timeout applied. | | **Yes** |
| Error Message Implementation  (To verify, test Insufficient Balance as example) | 1. Implemented all the error codes in production so customers are getting a clear message of failed payment. 2. For popup message, message should show at least 6 second or it should have close option. 3. Customer should see Payment Failed message as well. | 1. Not showing proper error message. 2. Showing error code as well with Error Message. 3. Not showing Payment Failed message. | | **Yes** |
| Implemented everything according to the Solution Document. | Process flow is maintained Accordingly. | Not followed. | | **Yes** |

**Security Check points:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Check Points** | **Normal Scenario** | **Abnormal Scenario** | **Checked (Yes/No)** |
| Site has a valid **SSL** certificate or app calls the API via **HTTPS**. | **HTTPS** | **HTTP** | **Yes** |
| Validating data between frontend request and backend system, before calling Create Payment API. | 1. Unique Invoice ID must be generated. 2. Validate the amount and Invoice ID from backend before calling Create Payment API. | 1. Duplicate Invoice ID generated. 2. Invoice ID and Amount can be changed via MITM attack. | **Yes** |
| Payment API require authentication (managed by cookie or token) | managed by cookie or token | No Authentication applied. | Cookie or Token? = Token |
| If you use Cookie or Token, what is the life time of the Cookie or Token? | Cookie or Token have a Life time. It should be as low as possible according to your business needs | Life time of a Cookie or token is not optimized. | Lifetime =?  1 Hour |
| Cookie or Token will expire after logout. | Cookie or Token will expire after logout. | Cookie or Token will not expire after logout. | **Yes** |
| Cookie or Token will expire after getting new Cookie or Token. | Cookie or Token will expire after getting new Cookie or Token. | Cookie or Token will not expire after getting new Cookie or Token. | **Yes** |
| Payment API is not accessible by external clients. | Payment API is not accessible by external clients. | Payment API is accessible by external clients. If accessible then what is the authentication procedure? | **Yes** |
| Website/Mobile APP is not vulnerable to the Owasp top 10 vulnerabilities (https://owasp.org/www-project-top-ten/) like SQL injection, XSS(Cross site scripting), etc. Please mention which tests you have performed. | Tested and found **Not Vulnerable** | 1. Tested and found **Vulnerable** 2. Not performed any test | **Not Vulnerable**  = **(Yes)** |
| **Tests Performed?** |
| 1. APP\_KEY  2. APP\_SECRET  3. USERNAME  4. PASSWORD  5. ID\_TOKEN  6. REFRESH\_TOKEN  Above parameters –   * used for calling bKash PGW APIs only. * not used for any other purpose or in any business logic. * stored securely. * not exposed in the front end (Browser or Mobile APP).   Note: If above parameter exposed, then as a merchant your reputation/revenue can both be hampered. | Mentioned parameters –   * used for calling bKash PGW APIs only. * not used for any other purpose or in any business logic. * stored securely. * not exposed in the front end (Browser or Mobile APP). | Mentioned parameters –   * used for other purpose or in any business logic. * Not stored securely. * exposed in the front end (Browser or Mobile APP).   Note: If mentioned parameter exposed, then as a merchant your reputation/revenue can both be hampered. | **Yes** |

**List of Owasp top 10 vulnerabilities (https://owasp.org/www-project-top-ten/):**

|  |  |  |
| --- | --- | --- |
| A01:2021 | Broken Access Control | Tested and found **not vulnerable** |
| A02:2021 | Cryptographic Failures | Not performed any test |
| A03:2021 | [Injection](https://owasp.org/Top10/A03_2021-Injection/) | Tested and found **not vulnerable** |
| A05:2021 | [Security Misconfiguration](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/) | Tested and found **not vulnerable** |
| A07:2021 | [Identification and Authentication Failures](https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/) | Tested and found **not vulnerable** |
| A08:2021 | [Software and Data Integrity Failures](https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/) | Not performed any test |
| A09:2021 | [Security Logging and Monitoring Failures](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/) | Tested and found **not vulnerable** |
| A10:2021 | [Server-Side Request Forgery](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/) | Tested and found **not vulnerable** |

Nagadhat staging link: <https://staging.nagadhat.com.bd/>

User: 01725361208  
Password: 01725361208